HYBRIDIZATION METHODS OF DYNAMICAL SYSTEMS

Pradyot Prakash
IST Austria
June 30th, 2015
OVERVIEW
GOAL

- Large number of systems governed by continuous dynamics
- Input: A set of initial states, forbidden states
- Output: Set of reachable states and bad states
- Focus on dynamics given by differential equations of the form

\[ \dot{x} = f(x) \]
Reachable states

- Set of states that a system passes through over time

Image from: www.mathworks.com
Bad states

- Reachable states $\cap$ Forbidden states
HYBRIDIZATION APPROACH

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

- \( f(x) \) non-linear in many of the cases
- Non-linear systems difficult to analyze computationally
- Need to find simpler dynamics (possibly affine)
- Need to partition the state space into multiple modes with simpler dynamics in each mode
Hybrid Automata

- State A: \(0 \leq x \leq 1, 1 \leq y \leq 2\) with Flow \(\text{Flow}_A\)
- State B: \(1 \leq x \leq 2, 1 \leq y \leq 2\) with Flow \(\text{Flow}_B\)
- State C: \(0 \leq x \leq 1, 0 \leq y \leq 1\) with Flow \(\text{Flow}_C\)
- State D: \(1 \leq x \leq 2, 0 \leq y \leq 1\) with Flow \(\text{Flow}_D\)

Transitions:
- \(\text{guard}_{AB}\) from A to B
- \(\text{guard}_{BA}\) from B to A
- \(\text{guard}_{CA}\) from C to A
- \(\text{guard}_{AC}\) from A to C
- \(\text{guard}_{DB}\) from D to B
- \(\text{guard}_{BD}\) from B to D
- \(\text{guard}_{CD}\) from C to D
- \(\text{guard}_{DC}\) from D to C
FLOWS
\[ \dot{x} = f(x) \]
\[ x \in P \]

How to define P?

- Initially the original domain: possibly a very large polytope
- Need to partition the state space into smaller hyper-boxes
- Need to ensure an over approximation of the dynamics
- Estimate bounds on the dynamics
- Have simpler dynamics for each partition P

Focus on systems with a single state initially
Image from: https://buffy.eecs.berkeley.edu
Suppose we have a 2D system:

\[
\begin{align*}
\dot{x} &= f_1(x, y) \\
\dot{y} &= f_2(x, y)
\end{align*}
\]

And we get,

\[
\begin{align*}
x &\in [x_a, x_b] \\
y &\in [y_a, y_b]
\end{align*}
\]
Suppose we have a 2D system:

\[
\begin{align*}
\dot{x} &= f_1(x, y) \\
\dot{y} &= f_2(x, y)
\end{align*}
\]

And we get,

\[
\begin{align*}
x &\in [x_a, x_b] \\
y &\in [y_a, y_b]
\end{align*}
\]
Suppose we have a 2D system:

$$\dot{x} = f_1(x, y)$$
$$\dot{y} = f_2(x, y)$$

And we get,

$$x \in [x_a, x_b]$$
$$y \in [y_a, y_b]$$

We hybridize the dynamics of the initial system to the domain in orange!
\[ \dot{x} = f(x) = [f_1(x) \ldots f_n(x)]^T \]

We find:

\[ \delta_{i_{\text{min}}} = \min_{x \in \mathbb{P}} f_i(x) \]
\[ \delta_{i_{\text{max}}} = \max_{x \in \mathbb{P}} f_i(x) \]

Replace:

\[ \delta_{i_{\text{min}}} \leq \dot{x}_i \leq \delta_{i_{\text{max}}} \quad \forall i \]

This is called differential inclusion. This defines a linear hybrid automata.

Reachability analysis is rather efficient for this class of systems.
RESULTS I
\[ \dot{x} = 1 + x^2y - 2.5x \]
\[ \dot{y} = 1.5x - x^2y \]

\( x, y : \text{molarity} \)

**Figure: Flow**
\[
\dot{x} = 1 + x^2 y - 2.5x \\
\dot{y} = 1.5x - x^2 y \\
x, y : molarity
\]

**Figure:** SpaceEx: 30x40 grid size
\[ \dot{x} = y \]
\[ \dot{y} = (1 - x^2)y - x \]

\( x \): position

**Figure: Flow**
Van der Pol Oscillator

Figure: SpaceEx: 30x30 grid size

\[
\begin{align*}
\dot{x} &= y \\
\dot{y} &= (1 - x^2)y - x \\
x & : \text{position}
\end{align*}
\]
Can we improve these results?
Can we improve these results?

Yes!
We approximate:

\[
\dot{x} = Ax + u \quad A_{nxn}, u_{nx1}
\]

Original dynamics,

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

\[
\implies u = f(x) - Ax
\]

\(u\) can be treated as divergence from the actual value or simply the error in approximating.

**General Problem**

Find \(A\) s.t.

\[
\delta = \min_{A \in \mathbb{R}^{nxn}} \max_{x \in P} \|f(x) - Ax\|
\]
AFFINE DYNAMICS

- Not easy to solve!
- Alternative: Use Jacobian of $f$ to find $A$
- Equivalent to First Order Taylor Series approximation
- Compute Jacobian matrix at the center of the hyperbox
AFFINE DYNAMICS

- Not easy to solve!
- Alternative: Use Jacobian of \( f \) to find \( A \)
- Equivalent to First Order Taylor Series approximation
- Compute Jacobian matrix at the center of the hyperbox

Dropping \( \min_{A \in \mathbb{R}^{n \times n}} \), we find bounds for \( u \),

\[
\begin{align*}
    u_{\text{min}} &= \min_{x \in P} f(x) - Ax \\
    u_{\text{max}} &= \max_{x \in P} f(x) - Ax \\
    u &\in [u_{\text{min}}, u_{\text{max}}]
\end{align*}
\]
RESULTS II
\[ \dot{x} = 1 + x^2y - 2.5x \]
\[ \dot{y} = 1.5x - x^2y \]

\( x, y : \text{molarity} \)

**Figure: Flow**
\[ \dot{x} = 1 + x^2y - 2.5x \]

\[ \dot{y} = 1.5x - x^2y \]

\( x, y : \text{molarity} \)

**Figure:** SpaceEx: 30x40 grid size
Figure: SpaceEx: 10x10 grid size; 0.05 sampling time

\[ \dot{x} = 1 + x^2y - 2.5x \]
\[ \dot{y} = 1.5x - x^2y \]

\(x, y: \text{molarity}\)
\[ \dot{x} = 1 + x^2 y - 2.5x \]
\[ \dot{y} = 1.5x - x^2 y \]
\[ x, y : \text{molarity} \]

**Figure:** SpaceEx: 15x15 grid size; 0.05 sampling time
\[
\begin{align*}
\dot{x} &= 1 + x^2 y - 2.5x \\
\dot{y} &= 1.5x - x^2 y \\
x, y & : \text{molarity}
\end{align*}
\]

**Figure:** SpaceEx: 15x15 grid size; 0.01 sampling time
\[ \begin{align*} \dot{x} &= y \\
\dot{y} &= (1 - x^2)y - x \\
x &: \text{position} \end{align*} \]

Figure: Flow*
Van der Pol Oscillator

Figure: SpaceEx: 30x30 grid size

\[
\begin{align*}
\dot{x} &= y \\
\dot{y} &= (1 - x^2)y - x
\end{align*}
\]
Van der Pol Oscillator

Figure: SpaceEx: 10x10 grid size; 0.1 sampling time

\[
\begin{align*}
\dot{x} &= y \\
\dot{y} &= (1 - x^2)y - x
\end{align*}
\]

\( x : \text{position} \)
Van der Pol Oscillator

Figure: SpaceEx: 30x30 grid size; 0.1 sampling time

\[
\begin{align*}
\dot{x} &= y \\
\dot{y} &= (1 - x^2)y - x \\
x & : position
\end{align*}
\]
Van der Pol Oscillator

\[ \dot{x} = y \]
\[ \dot{y} = (1 - x^2)y - x \]

\(x\): position

**Figure:** SpaceEx: 30x30 grid size; 0.001 sampling time
Implementation Details

- Made improvements to HyST
- Added the Hybridization feature
- Takes in a single state SpaceEx system and hybridizes it a multiple mode system based on flags provided by the user
- Added layout feature to visualize the model
Figure: Before hybridization
Figure: After hybridization
CONCLUSION
FUTURE WORK

- Consider systems with more than one location
- Try different shapes for partitions (e.g. triangular)
- Partition state space based on simulation done on the initial model
Questions?