Set-based Computation for Hybrid Systems Verification

Thao Dang

Special Acknowledgements. To Oded Maler and Eugene Asarin
Plan

1. Hybrid Systems: Motivations, Definition, Verification
2. Set-based Computation for
   - Reachability Computation
   - Invariant Computation
   - Stability Verification
3. Nonlinear extensions
Hybrid Systems: Motivations

Hybrid Dynamical Systems

A term that refers to a collection of mathematical models for systems with mixed continuous-discrete dynamics

Motivations

- **90s:** intensive use of computer in industrial control applications
  - Computer Science: discrete systems (e.g., automata, Petri nets)
  - Control Theory: purely discrete systems (e.g., automata, Petri nets, statechart) and purely continuous dynamics (diff. (alg.) equations/inclusions)
  - Lack of theoretical framework to deal with interaction between continuous and discrete dynamics

- **00s:** embedded/cyber-physical systems
  - A “legitimate” theme evolving into an active theme in Mathematical Modelling, CS, Control Theory, Design Automation

- **Beyond engineering applications:** Biological Modelling
Hybrid Systems Models

- Extension from timed automata [Alur and Dill 91] to hybrid automata [Alur et al. 95] (encouraged by the success of timed verification): enriching continuous dynamics

- Extension from differential equations with discontinuity to switched systems and hybrid control models [Branicky, Liberzon et al. 90s]: enriching switching dynamics (from sliding modes to transitions with guards and resets)

**Piecewise-Constant Derivative systems (PCD)** [Asarin, Maler, Pnueli, 93]
Example: Ball on String

- **dynamics** in *freefall* when $x \geq 0$, with mass $m$, $m\ddot{x} = F_g = -mg$.
- **dynamics** in *extension* when $x \leq 0$, with spring constant $k$, damping factor $d$, $m\ddot{x} = F_g + F_s = -mg - kx - d\dot{x}$.
- **transition** when $x = L$, collision factor $c \in [0, 1]$, $\dot{x}' = -c\dot{x}$.
Hybrid Automaton Model for Ball on String

(from SpaceEx Model Editor [Frehse et al. 11, http://spaceex.imag.fr])
Hybrid Automaton Behavior

---

Diagram showing the behavior of position $x$ and velocity $v$ over time $t$. The diagram illustrates the transition of states $x_0$ to $x_5$ and $v_0$ to $v_5$ with marked points at $t = 0, 0.5, 1, 1.5, 2, 2.5$. The graph depicts the changes in position and velocity over time, highlighting the dynamics of the hybrid automaton system.
Continuous Reachability Problem

Consider a simple linear differential equation:

\[ \dot{x} = Ax, \quad x(0) \in X_0 \]

Trajectory \( \xi(t) \) from an initial state \( x_0 \): \( \xi_{x_0}(t) = e^{At}x_0 \)

Reachable set from an initial set \( X_0 \): \( R = \{ e^{At}x_0 \mid x_0 \in X_0, t \geq 0 \} \)

Basic reachability problem: Is it possible to reach a bad set \( B \) from an initial set \( X_0 \)?

Challenges

- **Linear dynamics**: matrix exponentiation operator
- **Non-linear dynamics**: closed-form solutions are generally not known
- **Discrete dynamics**: non-termination
Decidability Boundaries

- Proof of undecidability for PCD (piece-wise constant derivative systems) in 3 dimensions [Asarin, Maler, Pnueli, 93]
- Decidability boundaries for hybrid systems [Henzinger et al. 00]
  - Decidability for hybrid automata where continuous dynamics are rectangular, or linear/affine of special eigenstructures and discrete dynamics are with non-deterministic resets
  - Undecidability (already!) for general linear hybrid automata [Henzinger et al., tool HyTech 97]
A change of point of view for researchers in verification

- In the continuous world, seeking exact answers may not be wise (since models are hardly exact w.r.t. reality)
- It is meaningful to seek approximate answers on more complex systems with non-trivial continuous dynamics

That is how emerged the continuous/hybrid reachability approximation problems...
Plan

1. Hybrid Systems: Motivations, Definition, Verification
2. Set-based Computation for
   - Reachability Approximation
   - Invariant Computation
   - Stability Verification
3. Nonlinear extensions
Continuous Reachability Computation

An **affine** differential equation (describing a **non-autonomous** system):

$$\dot{x} = Ax + Bu, \quad u \in \mathcal{U},$$

Trajectory $\xi(t)$ from $\xi(0) = x_0$ for given input signal $\zeta(t) \in \mathcal{U}$:

$$\xi_{x_0,\zeta}(t) = e^{At}x_0 + \int_0^t e^{A(t-s)}B\zeta(s)ds.$$

Reachable set at time $t$ from set $\mathcal{X}_0$ for any input signal:

$$\mathcal{X}_t = e^{At}\mathcal{X}_0 \oplus \mathcal{Y}_t,$$

$$\mathcal{Y}_t = \int_0^t e^{As}\mathcal{U}ds = e^{At}\mathcal{X}_0 \oplus \lim_{\delta \to 0} \bigoplus_{k=0}^{[t/\delta]} e^{A\delta k}\delta\mathcal{U}.$$  

Set operations:
- **matrix exponentiation** $e^{At}$
- **linear transformation** $e^{At}\mathcal{X}_0$
- **Minkowski sum** $\oplus$ (more generally to deal with non-determinism in continuous dynamics)
Consider a simple affine switched system in discrete time:

\[ x_{k+1} = A_i x_k + u_i, \quad i \in \{1, 2\}. \]

Union/Join operations

\[ 2^n \text{ convex sets} \]

2 convex sets

4 convex sets

n steps
Set representation: digital encoding of a possibly infinite set of states

**Polytope**

\[ \{ x : Tx \leq d, x \in \mathbb{R}^n \} \]

- \( T \): real matrix,
- \( d \): vector of bounds.

**Ellipsoid**

\[ \{ x : (x - c)^T P (x - c) \leq 1 \} \]

- \( P \): Positive semi-definite matrix,
- \( c \): real vector (center)
Requirements: efficient computation for typically required operations:

- **Linear transformation** (linear/linearized continuous dynamics)
- **Minkowski sum** (input disturbance/control)
- **Set intersection** (transition guards/switching conditions)
- **Convex hull** (approximate union of reachable segments)
- **Closure and low complexity** under required set operations.

Some common examples:

- **Polytopes closed under linear transformation, intersection with linear constraints.**
- **Ellipsoids closed under linear transformation, transition matrices of linear systems with uncertain input.**
- **Zonotopes closed under linear transformation and Minkowski sum.**
Flowpipe Construction for Continuous Reachability Approximation

Given $\dot{x} = Ax$, $x(0) \in \mathcal{X}_0$ (initial set, for example a rectangle), approximate the reachable set from $\mathcal{X}_0$ using time discretization:

- Reachable set at exactly time $k\delta$ (by semi-group property): $\mathcal{X}_{k\delta} = e^{A\delta} \mathcal{X}_{k-1}$ ⇒ $\mathcal{X}_{k\delta}$ are parallelotopes and can be exactly computed if no numeral integration error).
- Reachable set for time interval $[k\delta, (k+1)\delta]$: $\mathcal{R}_{[k\delta,(k+1)\delta]} \subseteq \Omega_k$ (over-approximated for example by zonotopes).
- Reachable set $\mathcal{R} \subseteq \bigcup_k \Omega_k$ ⇒ over-approximated for example by the convex hulls of reachable sets over several steps.
Discrete Reachability Approximation

- Intersection with guard sets
- Reset maps
- Union when the reachable set intersects a guard not at the same time

**Note:** although set computation for discrete transitions is standard, geometric complexity is often drastically increased by discrete dynamics!!!

(clip from SpaceEx output, for the Ball on String example)
Common Set Representations

<table>
<thead>
<tr>
<th>#ops</th>
<th>polyhedra</th>
<th>ellipsoids</th>
<th>zonotopes</th>
<th>support fct.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$m$ constr.</td>
<td>$k$ gen.</td>
<td>$n \times n$ matrix</td>
<td>$k$ generators</td>
</tr>
<tr>
<td>convex hull</td>
<td>exp</td>
<td>1</td>
<td>approx</td>
<td>$n^2k$ approx</td>
</tr>
<tr>
<td>Minkowski sum</td>
<td>exp</td>
<td>$nk^2$</td>
<td>approx</td>
<td>$n$</td>
</tr>
<tr>
<td>linear map</td>
<td>$n^2m$ / exp</td>
<td>$n^2k$</td>
<td>$n^3$</td>
<td>$n^2k$</td>
</tr>
<tr>
<td>intersection</td>
<td>1</td>
<td>exp</td>
<td>approx</td>
<td>approx</td>
</tr>
</tbody>
</table>
Plan

1. Hybrid Systems: Motivations, Definition, Verification
2. Set-based Computation for
   - Reachability Computation
   - Invariant Computation
   - Stability Verification
3. Nonlinear extensions
Invariant Computation Problem

Linear discrete-time system $x_{k+1} = Ax_k$

**Positive Invariant:** Next reachable set contained inside the original set.

- Directions for convergence to an equilibrium are indicated by eigenvectors.
- Using zonotopes with eigenvectors as generators???
Usual Zonotope

A zonotope is a projection of higher dimensional hypercube onto lower dimensional space. For a real matrix of generators $\mathcal{V}$ and center $c$

$$Z(\mathcal{V}, c) := \{c + \mathcal{V}\zeta : \zeta \in [-1, 1]^m\}.$$ 

$\mathcal{V} = \begin{bmatrix} 1 & -1 & 0 & 0.5 \\ -1 & 0 & -1 & 0.5 \end{bmatrix}$

1. **Linear transformation and Minkowski sum**: Simple algebraic expressions

2. **Advantage** over polytope and ellipsoid for reachability problems

[Girard, Leguernic 07]
Drawback of Zonotopes for Computing Positive Invariants

- Eigenvectors can be complex valued.
- Usual zonotopes only have real-valued generators.

\[ a + \i b : \ a \neq 0 \ b \neq 0 \]

For a stable real matrix \( A \) with real eigenvalues \( \mu \) and real eigenvectors \( \mathcal{V} \),

\[ A (\mathcal{Z} (\mathcal{V}, 0)) \subseteq \mathcal{Z} (\mathcal{V}, 0) \]

* We cannot rely on the above proposition when eigenvectors are complex.
Extend simple zonotope to complex numbers in a way that can capture contraction along complex vectors.

Complex valued generators with complex combining coefficients whose absolute value $\leq 1$.

Geometrically, Minkowski sum of Ellipses and Line Segments.

For a complex matrix $\mathcal{V}$ and a real vector (center) $c$, 

$$
\mathcal{C}(\mathcal{V}, c) := \{ \mathcal{V} \zeta + c : \zeta \in \mathbb{C}^m, \|\zeta\|_\infty \leq 1 \}.
$$
Non-polytopic Projection of Complex Zonotope
For a stable matrix $A$ having complex eigenvalues $\mu$ and complex eigenvectors $V$,

$$A(C(V,0)) \subseteq C(V,0).$$
Template Complex Zonotope: Definition

Variable bounds on absolute values of combining coefficients.

\[ \mathcal{V} \in \mathbb{M}_{n \times m}(\mathbb{C}): \text{template, } s \in \mathbb{R}_+^m: \text{scaling factors}, \ c \in \mathbb{R}^n: \text{center}. \]

\[ \mathcal{T}(\mathcal{V}, c, s) = \{ \mathcal{V}\zeta + c : |\zeta_i| \leq s_i \ \forall i \in \{1, \ldots, m\} \}. \]

- Add more generators and adjust scaling factors to find better approximations.

\[ \mathcal{C}(\mathcal{V}, c) \]

\[ \mathcal{T}\left(\begin{bmatrix} \mathcal{V} & \mathcal{W} \end{bmatrix}, c, \begin{bmatrix} a \\ b \end{bmatrix}\right) \]

\[ \mathcal{C}\left(\begin{bmatrix} \mathcal{V} & \mathcal{W} \end{bmatrix}, s\right) \]
Basic Operations: Template Complex Zonotope

1. \( A\mathcal{T} (V, c, s) = \mathcal{T} (AV, Ac, s) \).

2. \( \mathcal{T} (V, c, s) \oplus \mathcal{T} (V', c', s') = \mathcal{T} \left( [V \ V'], c + c', \begin{bmatrix} s \\ s' \end{bmatrix} \right) \).

- Above are affine functions of center and scaling factors.
Checking Inclusion

Generally, checking inclusion between complex zonotopes requires non-convex optimization.

Inclusion of a point. Consider a point \( x \in \mathbb{C}^n \). Then \( x \in \mathcal{T}(\mathcal{V}, c, s) \subset \mathbb{C}^n \) if and only if all of the following is collectively true.

\[
\exists \zeta \in \mathbb{C}^m : \quad \forall \zeta = x - c \quad (1) \\
|\zeta| \leq s. \quad (2)
\]

Exact inclusion between template complex zonotopes. Consider \( \mathcal{V} \in \mathbb{M}_{n \times m}(\mathbb{C}) \) and \( \mathcal{V}' \in \mathbb{M}_{n \times r}(\mathbb{C}) \). The inclusion \( \mathcal{T}(\mathcal{V}', c', s') \subseteq \mathcal{T}(\mathcal{V}, c, s) \) holds if and only if

\[
\sup_{\{\zeta' \in \mathbb{C}^r : |\zeta'| \leq s'\}} \inf_{\{\zeta \in \mathbb{C}^m : \mathcal{V}\zeta = \mathcal{V}'\zeta' + c' - c\}} \sup_{i=1}^m (|\zeta_i| - s_i) \leq 0 \quad (3)
\]
Checking Inclusion

Generally, checking inclusion between complex zonotopes requires non-convex optimization. → we propose a convex relaxation that works well in practice.

Definition \((\mathcal{T}(\mathcal{V}', c', s') \subseteq \mathcal{T}(\mathcal{V}, c, s))\)

\[
\exists X \in M_{m \times r} (\mathbb{C}), y \in \mathbb{C}^m \text{ such that } \\
\forall X = \mathcal{V}' \text{ diag}(s'), \forall y = c' - c \\
\sup_{i=1}^{m} \left( |y_i| + \sum_{j=1}^{r} |X_{ij}| - s_i \right) \leq 0.
\]

Result: \(\subseteq \iff \subseteq\).

- \(\subseteq\): **Second order conic constraints** (convex constraints) on **center** and **scaling factors**.
Intersection between Complex Zonotopes

- Closed for common invertible template

Let us consider a invertible matrix $\mathcal{V}$

$$\mathcal{T}(\mathcal{V}, c, s) \cap \mathcal{T}(\mathcal{V}, c, s') = \mathcal{T}(\mathcal{V}, c, s \land s').$$

Non-closure for a non-invertible template.

Closure for an invertible template.
## Set Representation Comparisons

<table>
<thead>
<tr>
<th>Set representation</th>
<th>Linear transformation</th>
<th>Minkowski sum</th>
<th>Intersection with half-space</th>
<th>Positive Invariant (non-empty interior) stable invertible linear transformation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Convex polytope H-representation</td>
<td>Efficient only for invertible matrix</td>
<td>exp</td>
<td>Efficient</td>
<td>Maximum complexity of encoding not bounded</td>
</tr>
<tr>
<td>Zonotope</td>
<td>Efficient</td>
<td>Efficient</td>
<td>Not closed</td>
<td>May not exist</td>
</tr>
<tr>
<td>Ellipsoid</td>
<td>Efficient</td>
<td>Not closed</td>
<td>Not closed</td>
<td>Efficient encoding</td>
</tr>
<tr>
<td>Polynomial sub-level set</td>
<td>exp</td>
<td>exp</td>
<td>Efficient</td>
<td>Efficient encoding</td>
</tr>
<tr>
<td>Template Complex Zonotope</td>
<td>Efficient</td>
<td>Efficient</td>
<td>Not closed</td>
<td>Efficient encoding</td>
</tr>
</tbody>
</table>

Other set representations: augmented complex zonotopes [Adimoolam et al.], quadratic zonotopes [Adjé et al.], polynomial zonotopes [Althoff et al.], occupation measures [Laserre et al., Rocca et al.]
Plan

1. Hybrid Systems: Motivations, Definition, Verification
2. Set-based Computation for
   - Reachability Computation
   - Invariant Computation
   - Stability Verification
3. Nonlinear extensions
Stability Verification: Networked Control Systems

- **Networked control system** (plant and controller) with sensors (sampling) and actuators (impulse)
- Modeled as **Nearly periodic linear impulsive system**

1. **Linear impulse** (actuation) applied to sampled state of continuous time linear system. E.g.
   \[
   \dot{x} = A_c x, \quad \dot{\tau} = 1, \quad \tau \leq \tau_{\text{max}}
   \]
   If \( \tau \in [\tau_{\text{min}}, \tau_{\text{max}}] \)
   \( x := A_r x, \quad \tau := 0 \)

2. **Time between successive sampling is bounded.**

**Global exponential stability (GES)**

Exists \( \lambda \in [0, 1) \) and \( c > 0 \) such that \( \forall x_0 \in \mathbb{R}^n, \; k \in \mathbb{Z}_+ \)

\[
\|x_k\| \leq c\lambda^k\|x_0\|
\]
$H_T = e^{A_c \tau} A_r$. Contractive $C$-set [Fiacchini et al 2014]:

**Compact and convex set $\Psi$:**

1. **Origin is interior point.**
2. **Contraction:** $\exists \lambda \in [0, 1)$:
   $\forall \tau \in [\tau_{\text{min}}, \tau_{\text{max}}]$
   
   \[ H_T \Psi \subseteq \lambda \Psi. \]

Existence of **contractive $C$-set $\iff$ GES.**

- Find **contractive $C$-set to verify GES.**
2 stages:

1. **Synthesize a candidate complex zonotope.**
2. **Verify contraction** of candidate complex zonotope.
1. Collect **eigenvectors** of *K*-uniformly sampled reachability operators as template.

   Eigenvectors of $e^{A_c t} A_r$:
   
   $t$ is uniformly sampled

2. **Synthesize scaling factors** such that we get
   
   1. $\lambda$-contractive complex zonotope
   2. contains unit box containing origin (*C*-set)
1. Derive contraction bound in a small neighborhood of a reachability operator using known operations on complex zonotopes.
   - Requires convex optimization.

2. Dynamically divide sampling intervals into small sub-intervals while verifying contraction bound \(< 1\).

\[ e^{A_c t} A_r \]

\[ h_t = \text{Maximum interval at } t \text{ for which contraction bound is less than } 1. \]
Plan

1. Hybrid Systems: Motivations, Definition, Verification
2. Set-based Computation for
   - Reachability Computation
   - Invariant Computation
   - Stability Verification
3. Nonlinear extensions
\[ \dot{x} = f(x), \text{ with } f \text{ globally Lipschitz continuous.} \]

- choose a domain \( S \) (partition, sliding window)
- overapproximate in \( S \) with \( \dot{x} = Ax + u, u \in U \)
- linearizing \( f(x) \) around \( x_0 \in S \) gives

\[
\begin{align*}
    a_{ij} &= \left. \frac{\partial f_i}{\partial x_j} \right|_{x=x_0} \\
    b &= f(x_0) - Ax_0.
\end{align*}
\]

\[
U = \text{Appr} \{ f(x) - (Ax + b), x \in S \} \oplus b.
\]
Example: Van der Pol Oscillator

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

Hybridization: here triangular partition of size 0.05
 Bernstein expansions for polynomials
  - polyhedral approximation of successors [Dang et al. 12]

 Taylor models
  - polynomial approximations of Taylor expansion
  - represent sets with polynomials
  - Flow* verification tool [Chen et al. 12]

 Zonotopic Approximation
  - Linearization of nonlinear dynamics
  - represent sets with polynomial zonotope
  - CORA verification tool [Althof et al. 15]
Conclusions

- “More than 2 decades of research with a lot of optimism” (Eugene Asarin)
- The initial goal of “removing the word towards” (Oded Maler) from hybrid systems research papers and project proposals has been met!
- We are “there” indeed:
Appendix: Support Functions

(a) support function in direction $d$

(b) outer approximation

**support function** = linear optimization (efficient!)

$$\rho_\mathcal{P}(d) = \sup\{ d^T x \mid x \in \mathcal{P} \}. $$

computed values define polyhedral outer approximation

$$\left[\mathcal{P}\right]_D = \bigcap_{d \in D} \{ d^T x \leq \rho_\mathcal{P}(d) \}. $$
Appendix: Support Functions (2)

(a) support function in direction $d$

(b) outer approximation

- linear map: $\rho_{MX}(d) = \rho_X(M^Td), \mathcal{O}(mn)$,
- convex hull: $\rho_{\text{chull}(P \cup Q)}(d) = \sup\{\rho_P(d), \rho_Q(d)\}, \mathcal{O}(1)$,
- Minkowski sum: $\rho_{X \oplus Y}(d) = \rho_X(d) + \rho_Y(d), \mathcal{O}(1)$. 

\[ P \]

\[ \rho_P(d) \]

\[ d \]

\[ \rho_{\text{chull}(P \cup Q)} \]

\[ \sup \]

\[ Q \]

\[ \rho_Q(d) \]

\[ \mathcal{O}(1) \]