Using Taylor Models in the Reachability Analysis of Non-linear Hybrid Systems

Xin Chen[1] Erika Ábrahám[1] Sriram Sankaranarayanan[2]

[1]RWTH Aachen University, Germany

[2]University of Colorado, Boulder, CO.

SWIM 2012

Outline

1. Verification of hybrid systems
2. Taylor model method
3. Intersections of Taylor models and guards
4. Experimental results
5. Future work

Outline

Motivations

Given a hybrid system on which we would like to verifiy

1. **Reachability** – A given state set is reachable.
2. **Safety** – No unsafe state is reachable.
3. **Liveness** – Some good properties *keep happening*.
4. **...**

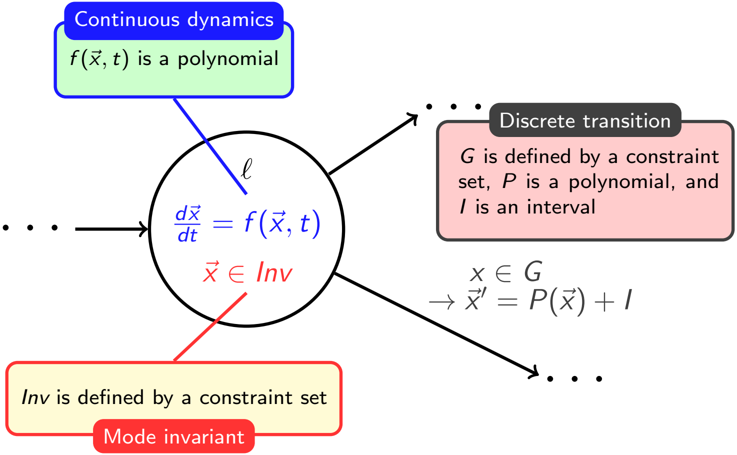
Motivations

Given a hybrid system on which we would like to verifiy

1. **Reachability** – A given state set is reachable.
2. **Safety** – No unsafe state is reachable.
3. **Liveness** – Some good properties *keep happening*.
4. **...**

Many important properties can be reduced to reachability.

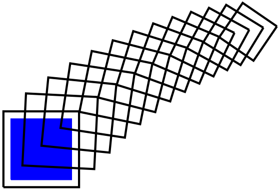
Hybrid automata



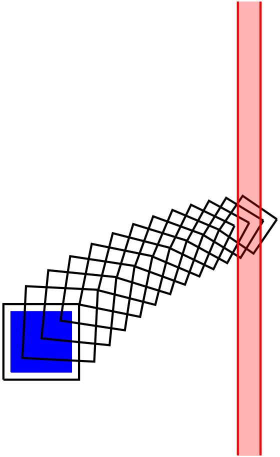
Reachability computation by flowpipe construction

Reachability computation by flowpipe construction

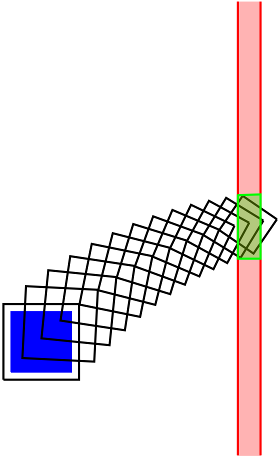
Reachability computation by flowpipe construction



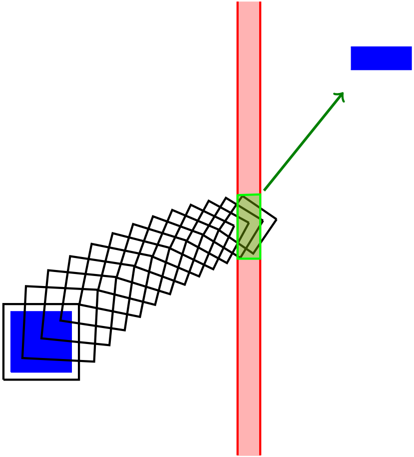
Reachability computation by flowpipe construction



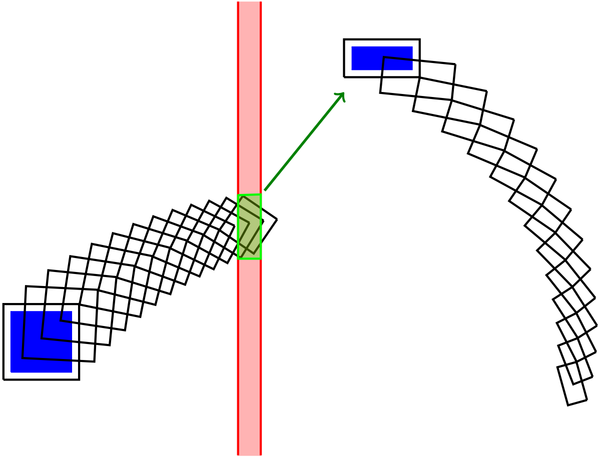
Reachability computation by flowpipe construction



Reachability computation by flowpipe construction



Reachability computation by flowpipe construction



Difficulties

Continuous part:

1. Overestimation is *heavily* accumulated.
2. Global accuracy improvement is inefficient.

Discrete part:

1. Flowpipe/guard and flowpipe/invariant intersections are difficult to compute.
2. Reducing the complexity of intersections is also hard.

Difficulties

Continuous part:

1. Overestimation is *heavily* accumulated.
2. Global accuracy improvement is inefficient.

Discrete part:

1. Flowpipe/guard and flowpipe/invariant intersections are difficult to compute.
2. Reducing the complexity of intersections is also hard.

We try to find new trade-off between accuracy and efficiency.

Outline

High order approximations

Polynomial p is a *k-order approximation* of a smooth function $f : D \to \mathbb{R}$ iff

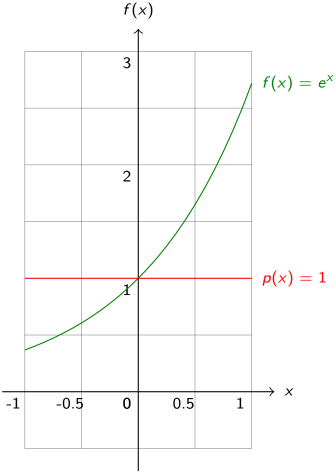
(a) $f \in C^k$

(b) $f(\vec{c}) = p(\vec{c})$ for the center point $\vec{c}$ of D and for each $0 < m \leq k$:

$$\left.\frac{\partial^m f}{\partial \vec{x}^m}\right|_{\vec{x}=\vec{c}} = \left.\frac{\partial^m p}{\partial \vec{x}^m}\right|_{\vec{x}=\vec{c}} .$$

Examples

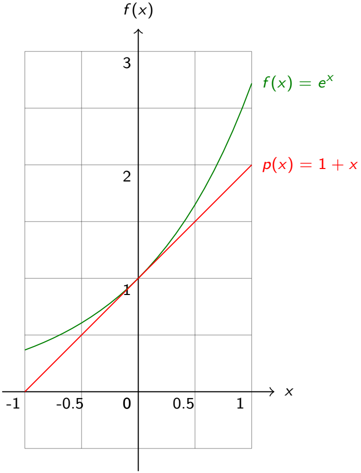
Several high order approximations of $f(x) = e^x$ with $x \in [-1, 1]$.



0-order approximation

Examples

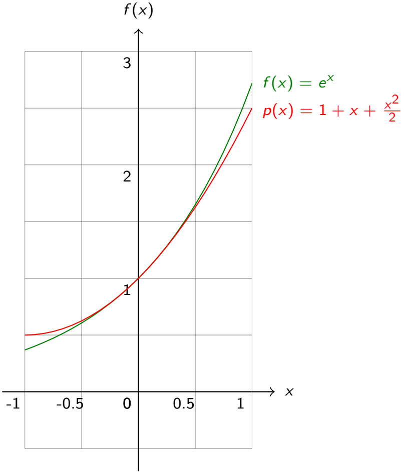
Several high order approximations of $f(x) = e^x$ with $x \in [-1, 1]$.



1-order approximation

Examples

Several high order approximations of $f(x) = e^x$ with $x \in [-1, 1]$.



2-order approximation

Taylor models

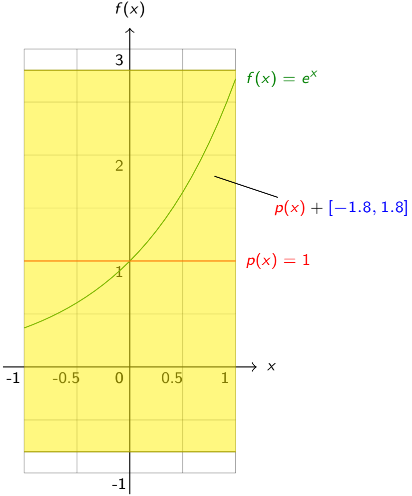
- Introduced by Berz and Makino.
- Taylor model: a pair (p, I) over an *interval* domain D. It defines the set

$$p + I \;=\; \{\vec{x} = p(\vec{x}_0) + \vec{y} \mid \vec{x}_0 \in D \wedge \vec{y} \in I\}$$

- Closed under many basic operations.

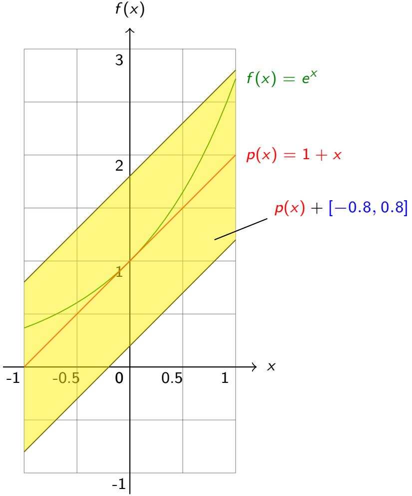
Berz. **Modern Map Methods in Particle Beam Physics.** ser. Advances in Imaging and Electron Physics. Academic Press, 1999, vol. 108.

High order over-approximations by Taylor models



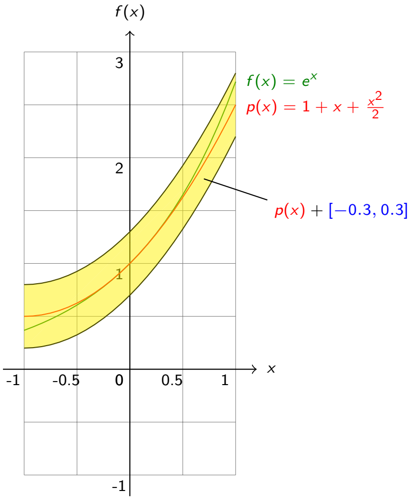
0-order over-approximation

High order over-approximations by Taylor models



1-order over-approximation

High order over-approximations by Taylor models



2-order over-approximation

Taylor model arithmetic

Addition:

$$(p_1, I_1) + (p_2, I_2) = (p_1 + p_2, I_1 \oplus I_2)$$

Multiplication:

$$(p_1, I_1) * (p_2, I_2) = (p_1 * p_2, (\mathtt{B}(p_1) \otimes I_2) \oplus (I_1 \otimes \mathtt{B}(p_2)) \oplus (I_1 \otimes I_2))$$

Truncation:

$$\mathtt{Trunc}_k((p_n, I)) = (p_k, I \oplus \mathtt{B}(p_n - p_k))$$

More operations: anti-derivation, Lie derivation, ...

Makino and Berz. **Taylor models and other validated functional inclusion methods.** *J. Pure and Applied Mathematics*, vol. 4, no. 4, 2003.

Verified integration by Taylor models

Assume the ODE is $\frac{d\vec{x}}{dt} = f(\vec{x}, t)$ and the initial set is given by a Taylor model X_0 over domain D_0. In the i^{th} integration step,

Berz and Makino. **Verified integration of ODEs and flows using differential algebraic methods on high-order Taylor models.** *Reliable Computing*, vol. 4, 1998.
Berz and Makino. **Rigorous integration of flows and ODEs using Taylor models.** *In Proc. of SNC'09.*

Verified integration by Taylor models

Assume the ODE is $\frac{d\vec{x}}{dt} = f(\vec{x}, t)$ and the initial set is given by a Taylor model X_0 over domain D_0. In the i^{th} integration step,

1. Compute a k-order approximation $p_k(\vec{x}_0, t)$ over $\vec{x}_0 \in X_{i-1}, t \in [0, t_i - t_{i-1}]$ for the flow starting from the Taylor model X_{i-1}.

Berz and Makino. **Verified integration of ODEs and flows using differential algebraic methods on high-order Taylor models.** *Reliable Computing*, vol. 4, 1998.
Berz and Makino. **Rigorous integration of flows and ODEs using Taylor models.** *In Proc. of SNC'09.*

Verified integration by Taylor models

Assume the ODE is $\frac{d\vec{x}}{dt} = f(\vec{x}, t)$ and the initial set is given by a Taylor model X_0 over domain D_0. In the i^{th} integration step,

1. Compute a k-order approximation $p_k(\vec{x}_0, t)$ over $\vec{x}_0 \in X_{i-1}, t \in [0, t_i - t_{i-1}]$ for the flow starting from the Taylor model X_{i-1}.
2. Evaluate a proper remainder I_k such that $(p_k(\vec{x}_0, t), I_k)$ is an over-approximation of the flow in $[0, t_i - t_{i-1}]$.

Berz and Makino. **Verified integration of ODEs and flows using differential algebraic methods on high-order Taylor models.** *Reliable Computing*, vol. 4, 1998.
Berz and Makino. **Rigorous integration of flows and ODEs using Taylor models.** *In Proc. of SNC'09.*

Verified integration by Taylor models

Assume the ODE is $\frac{d\vec{x}}{dt} = f(\vec{x}, t)$ and the initial set is given by a Taylor model X_0 over domain D_0. In the i^{th} integration step,

1. Compute a k-order approximation $p_k(\vec{x}_0, t)$ over $\vec{x}_0 \in X_{i-1}, t \in [0, t_i - t_{i-1}]$ for the flow starting from the Taylor model X_{i-1}.
2. Evaluate a proper remainder I_k such that $(p_k(\vec{x}_0, t), I_k)$ is an over-approximation of the flow in $[0, t_i - t_{i-1}]$.
3. Compute the i^{th} flowpipe $X_{[t_{i-1}, t_i]} = (p_k(X_{i-1}, t), I_k)$, and $X_i = (p_k(X_{i-1}, t_i - t_{i-1}), I_k)$.

Berz and Makino. **Verified integration of ODEs and flows using differential algebraic methods on high-order Taylor models.** *Reliable Computing*, vol. 4, 1998.
Berz and Makino. **Rigorous integration of flows and ODEs using Taylor models.** *In Proc. of SNC'09.*

Performance

Performance

Complexity: If p is a polynomial with n variables and k degree, then it could have $\binom{n+k}{k}$ monomials in the worst case.

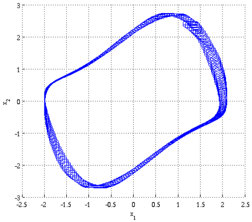
Performance

Complexity: If p is a polynomial with n variables and k degree, then it could have $\binom{n+k}{k}$ monomials in the worst case.

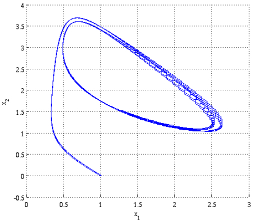
Accuracy:

- The initial set in every step is represented by a *high order model*.
- By Taylor model arithmetic, *overestimation is effectively restricted*.
- Auxiliary methods can be used to further improve the accuracy, such as shrink wrapping, preconditioning, ...

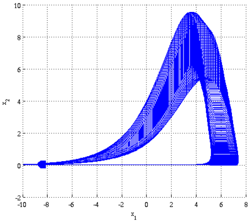
Continuous systems:



Van-der-Pol oscillator

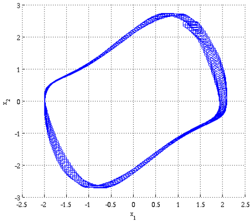


Brusselator

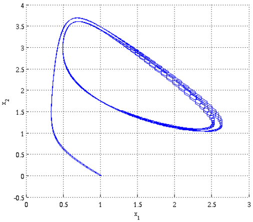


Rössler attractor

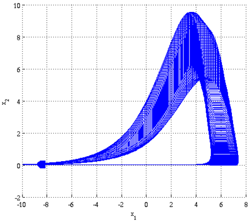
Continuous systems:



Van-der-Pol oscillator



Brusselator



Rössler attractor

How can we apply Taylor models to hybrid systems?

Outline

Taylor model flowpipe construction for hybrid systems

- In a mode:
  - Verified integration by using Taylor models.
  - Compute flowpipe/invariant intersections.
- For a discrete transition:
  - Compute flowpipe/guard intersections.
  - Compute the image of the reset mapping.

Taylor model flowpipe construction for hybrid systems

- In a mode:
  - Verified integration by using Taylor models.
  - Compute flowpipe/invariant intersections.
- For a discrete transition:
  - Compute flowpipe/guard intersections.
  - Compute the image of the reset mapping.

Over-approximate an intersection

Intersection of a Taylor model (p, I) and a guard:

$$\underbrace{\vec{x} = p(\vec{x}_0, t) + \vec{y} \ \wedge \ \underbrace{\vec{x}_0 \in D_0 \wedge t \in [0, \Delta]}_{\text{Taylor model domain}} \ \wedge \ \vec{y} \in I}_{\text{Taylor model}} \ \wedge \ \underbrace{\gamma(\vec{x})}_{\text{guard predicate}}$$

Over-approximate an intersection

We propose the following techniques to over-approximate the intersection $(p, I) \cap G$.

- **Domain contraction** -
  Contract the domain of p such that (p, I) with the new domain is the over-approximation.
- **Range over-approximation** -
  Over-approximate the Taylor model (p, I) by a convex set S, then compute $S \cap G$.
- **Template method** -
  Compute a Taylor model over-approximation $(p^*, 0)$ based on a given *template*.

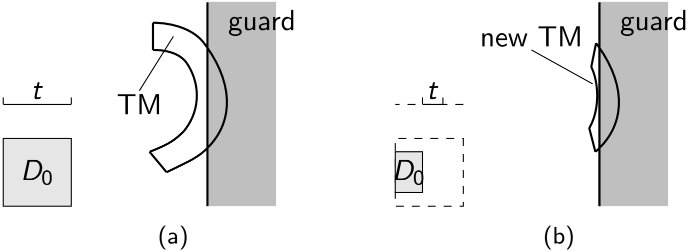
Over-approximate an intersection

We propose the following techniques to over-approximate the intersection $(p, I) \cap G$.

- **Domain contraction** -
  Contract the domain of p such that (p, I) with the new domain is the over-approximation.
- **Range over-approximation** -
  Over-approximate the Taylor model (p, I) by a convex set S, then compute $S \cap G$.
- **Template method** -
  Compute a Taylor model over-approximation $(p^*, 0)$ based on a given *template*.

They can be used in a combination.

Domain contraction - An example



Domain contraction - Main procedure

Compute an *interval* contraction by *interval constraint propagation*:

1. Contract the interval in one dimension.
2. Propagate the result to the next dimension.

Domain contraction - Main procedure

Compute an *interval* contraction by *interval constraint propagation*:

1. Contract the interval in one dimension.
2. Propagate the result to the next dimension.

Accuracy can be further improved by

- Perform the contraction through all dimensions for several times.
- Determine an order on the dimensions and perform the contraction for one time.

Domain contraction - Contraction in one dimension

Compute the lower bound for the contracted domain in the i^{th} dimension.

while the size of $[lo, up]$ is larger than ϵ **do**
  Split $[lo, up]$ into $[lo, a]$ and $[a, up]$ wherein $a = \frac{lo+up}{2}$;
  if $(p(\vec{y}), I)$ with $\vec{y} \in D$ and $(\vec{y})_i \in [lo, a]$ intersects the guard **then**
    $up \leftarrow a$;
  else
    $lo \leftarrow a$;
  end if
end while
return lo;

Domain contraction - Contraction in one dimension

Compute the lower bound for the contracted domain in the i^{th} dimension.

while the size of $[lo, up]$ is larger than ϵ **do**
    Split $[lo, up]$ into $[lo, a]$ and $[a, up]$ wherein $a = \frac{lo+up}{2}$;
    if $(p(\vec{y}), I)$ with $\vec{y} \in D$ and $(\vec{y})_i \in [lo, a]$ intersects the guard **then**
        $up \leftarrow a$;
    else
        $lo \leftarrow a$;
    end if
end while
return lo;

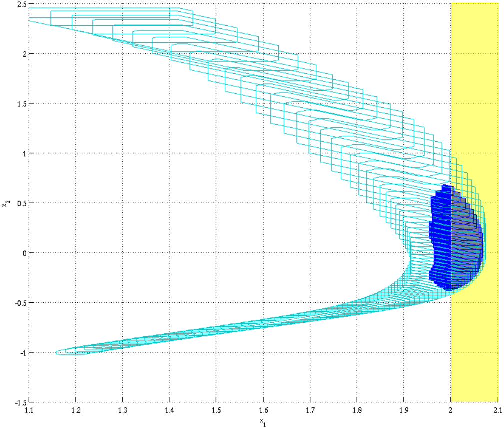
Domain contraction - Emptiness checking

Check the emptiness of

$$\{\vec{x} \in \mathbb{R}^n \mid \vec{x} = p(\vec{y}) + \vec{z} \ \wedge \ \vec{y} \in D \ \wedge \ \vec{z} \in I \ \wedge \ \gamma(\vec{x})\}$$

Approximation methods: *Conservative simplification* on p and γ.

Domain contraction - An example



Domain contraction

Advantages:

- Polynomial computation time.
- The result is still a Taylor model.

Disadvantages:

- The contracted domain is always an interval.
- Large overestimation might be introduced when either p or γ is of high degree.

Range over-approximation

1. Over-approximate (p, I) by a set S which can be a
   - **support function** -
     Conservative over-approximation of a polynomial.
   - **zonotope** -
     Zonotopes are order 1 *Taylor models and vise versa.*
2. Over-approximate $S \cap G$ by a Taylor model.

Sankaranarayanan, Dang and Ivancic. **Symbolic Model Checking of Hybrid Systems Using Template Polyhedra.** In *Proc. of TACAS'08.*
Le Guernic and Girard. **Reachability Analysis of Hybrid Systems Using Support Functions.** In *Proc. of CAV'09.*

Range over-approximation

Advantages:

- Available algorithms can be used.
- The orientation of the over-approximation can be chosen.

Disadvantages:

- Multi-time over-approximation.
- Best orientation is not easy to find.

Template method

We call $p^*(\vec{x}_0)$ a *template polynomial* if its domain is given by

$$\vec{x}_0 \in D_u = [\ell_1, u_1] \times [\ell_2, u_2] \times \cdots \times [\ell_n, u_n]$$

wherein $\ell_1, \ldots, \ell_n, u_1, \ldots, u_n$ are unknown parameters.

Sankaranarayanan, Sipma and Manna. **Constructing invariants for hybrid systems.** *Formal Methods in System Design.* 2008.

Gulwani and Tiwari. **Constraint-Based Approach for Analysis of Hybrid Systems.** In *Proc. of CAV'08.*

Template method

We call $p^*(\vec{x}_0)$ a *template polynomial* if its domain is given by

$$\vec{x}_0 \in D_u = [\ell_1, u_1] \times [\ell_2, u_2] \times \cdots \times [\ell_n, u_n]$$

wherein $\ell_1, \ldots, \ell_n, u_1, \ldots, u_n$ are unknown parameters.

We find the parameters such that the Taylor model $(p^*, 0)$ contains the intersection $(p, I) \cap G$:

$$\forall \vec{x}.((\vec{x} = p(\vec{y}) + \vec{z} \wedge \vec{y} \in D \wedge \vec{z} \in I \wedge \vec{x} \in G)$$
$$\rightarrow \exists \vec{x}_0.(\vec{x} = p^*(\vec{x}_0) \wedge \vec{x}_0 \in D_u))$$

Sankaranarayanan, Sipma and Manna. **Constructing invariants for hybrid systems.** *Formal Methods in System Design.* 2008.
Gulwani and Tiwari. **Constraint-Based Approach for Analysis of Hybrid Systems.** In *Proc. of CAV'08.*

Template method

We call $p^*(\vec{x}_0)$ a *template polynomial* if its domain is given by

$$\vec{x}_0 \in D_u = [\ell_1, u_1] \times [\ell_2, u_2] \times \cdots \times [\ell_n, u_n]$$

wherein $\ell_1, \ldots, \ell_n, u_1, \ldots, u_n$ are unknown parameters.

We find the parameters such that the Taylor model $(p^*, 0)$ contains the intersection $(p, I) \cap G$:

$$\forall \vec{x}.((\vec{x} = p(\vec{y}) + \vec{z} \wedge \vec{y} \in D \wedge \vec{z} \in I \wedge \vec{x} \in G) \\ \rightarrow \exists \vec{x}_0.(\vec{x} = p^*(\vec{x}_0) \wedge \vec{x}_0 \in D_u))$$

We may also add more constraints to limit the overestimation.

Sankaranarayanan, Sipma and Manna. **Constructing invariants for hybrid systems.** *Formal Methods in System Design.* 2008.
Gulwani and Tiwari. **Constraint-Based Approach for Analysis of Hybrid Systems.** In *Proc. of CAV'08.*

Template method

Advantages:

- Flexibility.
- The unknown parameters can be found by SMT solving.

Disadvantages:

- Bad scalability because of the quantifier elimination.
- Best template is not easy to find.

Outline

1. Verification of hybrid systems
2. Taylor model method
3. Intersections of Taylor models and guards
4. Experimental results
5. Future work

Glycemic Control in Diabetic Patients

$$\begin{cases} \frac{dG}{dt} & = & -p_1 G - X(G + G_B) + g(t) \\ \frac{dX}{dt} & = & -p_2 X + p_3 I \\ \frac{dI}{dt} & = & -n(I + I_b) + \frac{1}{V_I} i(t) \end{cases}$$

Typical parameters:

$$\begin{array}{llll} p_1 = 0.01, & p_2 = 0.025, & p_3 = 1.3 \times 10^{-5}, & V_I = 12, \\ n = 0.093, & G_B = 4.5, & I_b = 15. & \end{array}$$

G	plasma glucose concentration above the basal value G_B
I	plasma insulin concentration above the basal value I_B
X	insulin concentration in an interstitial chamber

Glycemic Control in Diabetic Patients

$g(t)$ infusion of glucose into the bloodstream
$i(t)$ infusion of insulin into the bloodstream
The infusion of glucose after a meal:

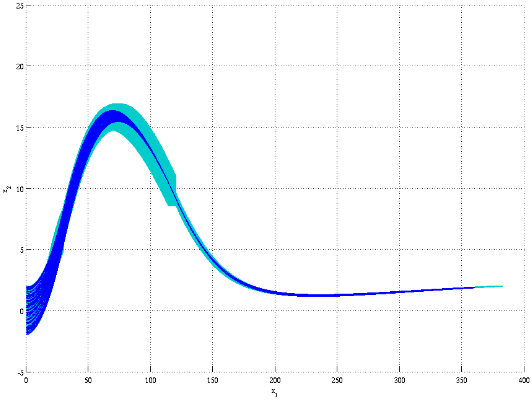
$$g(t) = \begin{cases} \frac{t}{60} & t \le 30 \\ \frac{120-t}{180} & t \in [30, 120] \\ 0 & t \ge 120 \end{cases}$$

Two control schemes:

$$i_1(t) = \begin{cases} \frac{25}{3} & G(t) \le 4 \\ \frac{25}{3}(G(t) - 3) & G(t) \in [4, 8] \\ \frac{125}{3} & G(t) \ge 8 \end{cases}$$

$$i_2(t) = \begin{cases} 1 + \frac{2G(t)}{9} & G(t) < 6 \\ \frac{50}{3} & G(t) \ge 6 \end{cases}$$

Glycemic Control in Diabetic Patients

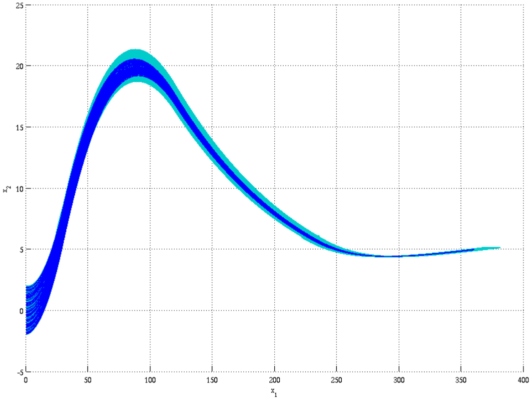


Our platform: CPU i7-860 2.8 GHz Memory: 4 GB

Order of the Taylor models: 9 Time step: 0.02 Time horizon: [0,360]

Total time: 2138 s Time of intersection: 663 s Memory: 430 MB

Glycemic Control in Diabetic Patients



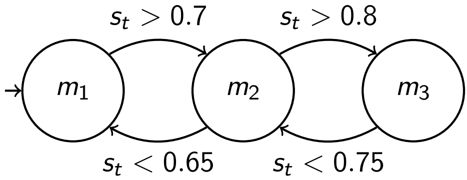
Order of the Taylor models: 9 Time step: 0.02 Time horizon: [0,360]

Total time: 1804 s Time of intersection: 443 s Memory: 410 MB

Vehicle model

The dynamics of a non-holonomic vehicle is given as follows,

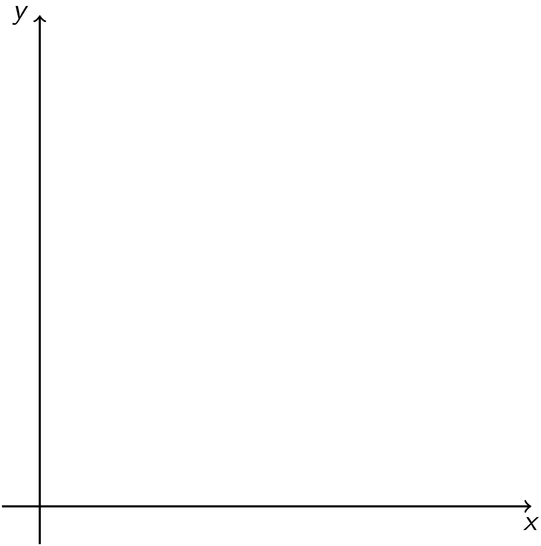
$$
\begin{array}{lll}
\frac{dx}{dt} = vc_t & \frac{dy}{dt} = vs_t & \frac{dv}{dt} = \color{red}{u_1} \\
\frac{dc_t}{dt} = \sigma v^2 s_t & \frac{ds_t}{dt} = -\sigma v^2 c_t & \frac{d\sigma}{dt} = \color{red}{u_2}
\end{array}
$$



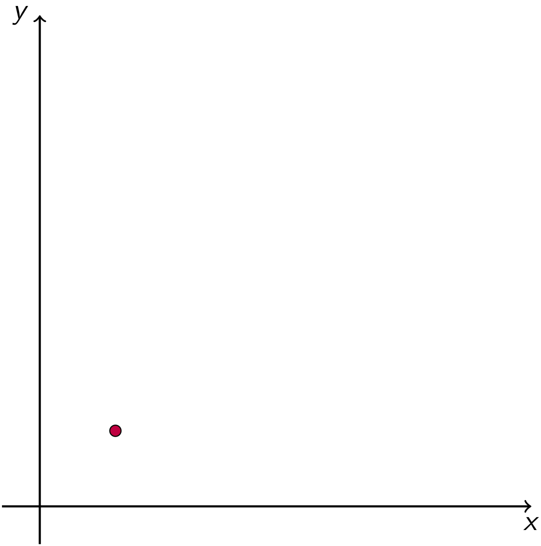
Initial set:

$$
\begin{array}{lll}
x \in [1, 1.2] & y \in [1, 1.2] & v \in [0.8, 0.81] \\
s_t \in [0.7, 0.71] & c_t \in [0.7, 0.71] & \sigma \in [0, 0.05] \,.
\end{array}
$$

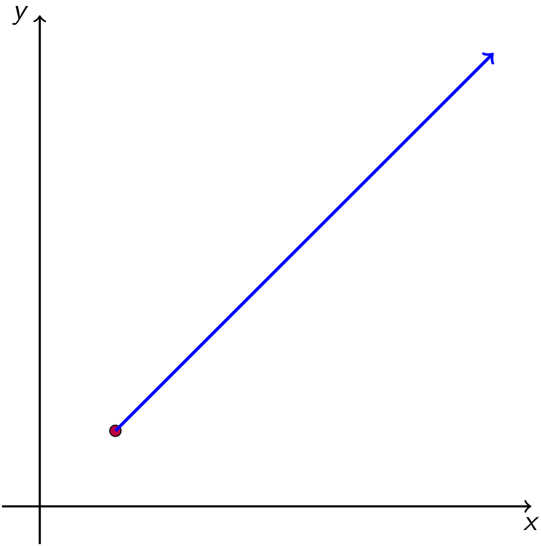
Vehicle model



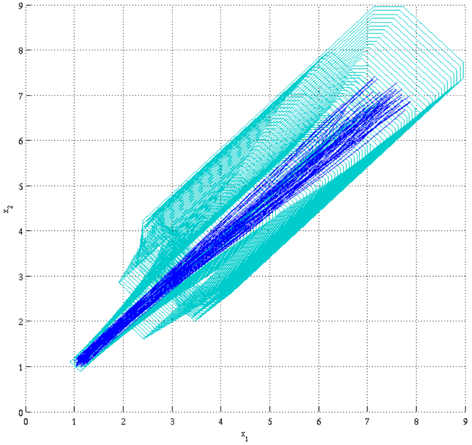
Vehicle model



Vehicle model



Vehicle model



Order of the Taylor models: 9 Time step: 0.1 Time horizon: [0,10]

Total time: 85 s Time of intersection: 40 s Memory: 4 MB

Scalability tests

D = 6							
deg = 2				deg = 4			
size	order	T	mem	size	order	T	mem
3	4	8	4	3	4	18	4
	9	211	12		9	449	12
5	4	13	4	5	4	27	4
	9	298	12		9	587	12
7	4	18	4	7	4	36	4
	9	373	12		9	687	12
D = 8							
deg = 2				deg = 4			
size	order	T	mem	size	order	T	mem
3	4	14	4	3	4	23	4
	9	1389	28		9	2430	28
5	4	18	4	5	4	27	4
	9	1433	28		9	2446	28
7	4	22	4	7	4	28	4
	9	2074	28		9	2474	28
D = 10							
deg = 2				deg = 4			
size	order	T	mem	size	order	T	mem
3	4	16	4	3	4	27	8
	9	t.o.	-		9	t.o.	-
5	4	22	8	5	4	46	8
	9	t.o.	-		9	t.o.	-
7	4	35	8	7	4	62	8
	9	t.o.	-		9	t.o.	-

Dang and Testylier. **Hybridization domain construction using curvature estimation.** In *Proc. of HSCC'11.*

More examples

				Our tool								Ariadne	
Benchmark	DEG	LOC	VAR	δ	T	ORD	T.T.	T.I.	MEM	D.C.	R.M.	T.T	MEM
Brusselator	3	1	2	0.05	[0,10]	4	77	0	4	-	-	34	12
Brusselator	3	1	2	0.03	[0,15]	4	152	0	8	-	-	EXC	-
Watertank	1	4	2	0.1	[0,80]	3	9	5	8	$\surd$	Z	7	24
Van-der-PolE	3	2	3	0.01	[0,6]	4	71	37	4	$\surd$	Z	EXC	-
Lotka-Volterra	2	1	3	0.01	[0,3]	4	14	0	8	-	-	52	8
Hallstah	3	1	2	0.01	[0,7]	3	19	0	≤ 1	-	-	0.6	≤ 1
B.B. no drag	1	1	4	0.02	[0,3]	3	0.8	0.3	≤ 1	$\surd$	-	0.2	≤ 1
B.B. const drag	1	2	4	0.02	[0,3]	3	2	0.8	≤ 1	$\surd$	-	0.6	≤ 1
B.B. Stokes-Einstein	2	2	4	0.02	[0,3]	3	8	2	≤ 1	$\surd$	-	EXC	-
Diabetic [16]	2	9	4	0.02	[0,360]	9	2138	663	430	$\surd$	Z	EXC	-
Diabetic [17]	2	6	4	0.02	[0,360]	9	1804	443	410	$\surd$	Z	EXC	-
Watt governor [39]	4	1	5	1e-4	[0,15]	5	NR	-	-	-	-	EXC	-
Vehicle	4	3	6	0.1	[0,10]	9	85	40	4	$\surd$	S.F.	EXC	-
Angiogenesis [38]	2	1	12	1e-8	[0,2e-6]	4	34	0	12	-	-	EXC	-
Coll-avoid-2 [40]	2	3	12	0.01	[0,10]	3	27	8	3	$\surd$	-	EXC	-

http://systems.cs.colorado.edu/research/cyberphysical/taylormodels/

Outline

Future work

- Taylor model flowpipe construction with varying orders and time steps.
- Split Taylor models according to equilibriums.
- Taylor model contraction for domains.
- Heuristics for choosing templates.

Thank you!

Questions?