Verification of non-linear systems using hybridization

Eugene Asarin (LIAFA, Paris), Thao Dang and Antoine Girard (VERIMAG, Grenoble)

Reachability analysis

Reachable set computations are useful for

- **Verification**
  problems such as proving that the system does not reach a 'bad' state
- **Controller synthesis**
  problems such as determining the set of states from which it is possible to reach a target set while avoiding a forbidden set

Many existing methods and tools (see the next slide)

Reachability analysis

Direct methods

- Track the evolution of the reachable set under the flow of the system. Various set representations: *e.g.* polyhedra, ellipsoids, level sets
- Exact results, or accurate approximations with error bounds. Using symbolic or numerical computations
- Tools: Coho, CheckMate, d/dt, HysDel, VeriShift, Vertdict, Requiem, Level-set toolbox, ..

Indirect methods

- Abstraction methods: reducing to a simpler system that preserves the property (*e.g.* Tiwari & Khanna 02; Alur et al. 02; Clarke et al. 03)
- Prove the property without computing reachable sets: *e.g.* Barrier certificates Prajna & Jadbabaie04, polynomial invariants Tiwari & Khanna04.

⋆ **Scalability** is still challenging (complexity and size of real-life systems)

Plan

Hybridization methods for non-linear systems

- Principle
- Construction of approximate systems
  - Piecewise affine approximation
  - Piecewise multi-affine approximation
- Reachability computations using approximate systems

Plan

Hybridization methods for non-linear systems

- **Principle**
- Construction of approximate systems
  - Piecewise affine approximation
  - Piecewise multi-affine approximation
- Reachability computation methods for approximate systems

Principle

System Δ : $\dot{x} = f(x), \quad x \in \mathcal{X}$, f is Lipschitz

Main Idea:

- **Step 1**. Approximate the 'complex' system Δ by a system which is "piecewise less complex" (complex: difficult to analyze). The approximate system is thus **hybrid**
- **Step 2**. Using the approximate system to yield approximate analysis results for Δ

Construction of the approximate system: partition the state space $\mathcal{X}$ into disjoint regions and assign to each region an approximate vector field (which is simpler to analyze)

Important questions :

- Error between the approximate and the original systems
- Classes of approximate vector fields to use

Construction of the approximate system

System Δ : $\dot{x} = f(x), \quad x \in \mathcal{X}, f$ is Lipschitz

- Partition the state space $\mathcal{X}$ into disjoint regions of size $\mathbf{h}$ and assign to each region an approximate vector field
- $\mathbf{h}$: space discretization size
- $f_{\mathbf{h}}$: composite vector field over the whole state space $\mathcal{X}$
- Hybridization error $\boldsymbol{\pi}(\mathbf{h}) = \sup_{x \in \mathcal{X}} ||\mathbf{f}(\mathbf{x}) - \mathbf{f_h}(\mathbf{x})||$
- Conservative approximate system

$$\text{System } \Delta_{\mathbf{h}} : \quad \dot{x} = f_{\mathbf{h}}(x) + u$$

$u(\cdot)$: disturbance taking values in $Ball(\boldsymbol{\pi}(\mathbf{h}))$

Approximation Properties

Conservativeness: The set of trajectories of Δ is included in the set of trajectories of $\Delta_{\mathbf{h}}$

Convergence

- Distance between trajectories: $x(0) = x_h(0)$, L is Lipschitz cst. of f.

$$\forall t \geq 0 : \; d\left(x(t), x_h(t)\right) \leq \frac{\boldsymbol{\pi}(\mathbf{h})}{L}(e^{Lt} - 1),$$

- Distance between the reachable sets up to time T

$$d\left(Reach_{[0,T]}(\Delta), Reach_{[0,T]}(\Delta_{\mathbf{h}})\right) \leq \frac{2\boldsymbol{\pi}(\mathbf{h})}{L}(e^{LT} - 1),$$

  that is $\mathcal{O}(\boldsymbol{\pi}(\mathbf{h}))$

- The reachable set of $\Delta_{\mathbf{h}}$ converges to the reachable sets of Δ with the same rate as $f_{\mathbf{h}}$ converges to f

Preservation of attractors

If Δ has an attractor A and B is its basin of attraction, and if

- $\forall x_0 \in B$ there exists a trajectory $x(t)$ of Δ such that $x(0) = x_0$
- $\exists \delta > 0 : \ \mathcal{N}(A, \delta) \subseteq B$.

Then, for all $\varepsilon \in (0, \delta]$, there exists $\nu > 0$ such that if $\boldsymbol{\pi}(\mathbf{h}) \leq \nu$ then there exists a set $A_{\mathbf{h}}$, which is an attractor for the hybrid system $\Delta_{\mathbf{h}}$ and such that

$$A \subseteq A_{\mathbf{h}} \subseteq \mathcal{N}(A, \varepsilon).$$

In addition, B is in the basin of attraction of $A_{\mathbf{h}}$.

Approximate systems

We developed two methods for constructing approximate systems with good error bound $\boldsymbol{\pi}(\mathbf{h})$

- Piecewise affine systems
- Piecewise multi-affine systems

Plan

Hybridization methods for non-linear systems

- **Principle**
- Construction of approximate systems
  - **Piecewise affine approximation**
  - Piecewise multi-affine approximation
- Reachability computation methods for approximate systems

Piecewise affine approximation

- Using a simplicial mesh, each cell $C_{\mathbf{i}}$ is a simplex of size $\mathbf{h}$ (edge length)
- Define for each $C_{\mathbf{i}} = conv\{v_1, \ldots, v_{n+1}\}$ an affine function $l_{\mathbf{i}}$ interpolating f at the vertices. That is, for each vertex v_k of $C_{\mathbf{i}}$

$$l_{\mathbf{i}}(v_k) = f(v_k)$$

- Then, the approximate system is: $f_{\mathbf{h}}(x) = l_{\mathbf{i}}(x)$ if $x \in C_{\mathbf{i}}$.

Properties of the approximation:

- Given a mesh, $f_{\mathbf{h}}$ is uniquely defined. (An affine n-dimensional function is uniquely determined by its values at $(n+1)$ affinely independent points.)
- Piecewise affine function $f_{\mathbf{h}}$ is continuous over the state space

Hybridization error

- If f is L-Lipschitz, then

$$\boldsymbol{\pi}(\mathbf{h}) \leq \mathbf{h}\frac{2n\,L}{n+1}\,,$$

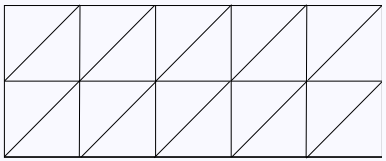
- If f is a C^2 function with a second derivative bound K, then

$$\boldsymbol{\pi}(\mathbf{h}) \leq \mathbf{h}^2\frac{n^2\,K}{2(n+1)^2}\,.$$

$\Rightarrow$ second-order approximation $\boldsymbol{\pi}(\mathbf{h}) = \mathcal{O}(\mathbf{h}^2)$

Mesh construction

- Define a rectangular mesh over $\mathcal{X}$
- Decompose each hyper-rectnagle into simplices: Let $\mathcal{P}$ be the set of permutations of $\{1, \ldots, n\}$.
  - Given $\mathbf{p} = (\mathbf{i_1}, \ldots, \mathbf{i_n}) \in \mathcal{P}$, the set $\mathbf{C_p} = \{0 \leq x_{\mathbf{i_1}} \leq \cdots \leq x_{\mathbf{i_n}} \leq 1\}$ is a simplex of $\mathbb{R}^n$.
  - $\{\mathbf{C_p}\}_{\mathbf{p} \in \mathcal{P}}$ defines a simplicial mesh of $[0, 1]^n$
  - Each hyper-rectangle is decomposed into $\mathbf{n}!$ simplices



Analysis of approximate systems

- Reachability computations for piecewise affine system $\Delta_{\mathbf{h}}$:
  Various existing techniques, such as **CheckMate**, **d/dt**
- Our implementation using reachability procedures of the tool **d/dt**

Plan

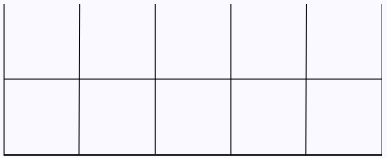
Hybridization methods for non-linear systems

- **Principle**
- Construction of approximate systems
  - Piecewise affine approximation
  - **Piecewise multi-affine approximation**
- Reachability computation methods for approximate systems

Piecewise multi-affine approximation

- Using a rectangular mesh, each cell C_i is a hypercube of size $\mathbf{h}$
- Define for each cell C_i a multi-affine function $f_{\mathbf{h}}$ interpolating f at its vertices
- Piecewise multi-linear function $f_{\mathbf{h}}$ is continuous over the state space

Approximation error: If f is C^2 on $\mathcal{X}$ with bounded second order derivatives $\Rightarrow$ **quadratic error**: $\boldsymbol{\pi}(\mathbf{h}) = \mathcal{O}(\mathbf{h}^2)$.



Defining multi-affine approximations

For a 2-dimensional rectangle $[l_1, u_1] \times [l_2, u_2]$, we define a multi-affine function that interpolates f at the vertices of the rectangle:

For $x \in [l_1, u_1] \times [l_2, u_2]$,

$$\begin{aligned} f_{\mathbf{h}}(x) &= \frac{(u_2 - x_2)}{(u_2 - l_2)} \frac{(u_1 - x_1)}{(u_1 - l_1)} f(l_1, l_2) \\ &+ \frac{(u_2 - x_2)}{(u_2 - l_2)} \frac{(x_1 - l_1)}{(u_1 - l_1)} f(u_1, l_2) \\ &+ \frac{(x_2 - l_2)}{(u_2 - l_2)} \frac{(u_1 - x_1)}{(u_1 - l_1)} f(l_1, u_2) \\ &+ \frac{(x_2 - l_2)}{(u_2 - l_2)} \frac{(x_1 - l_1)}{(u_1 - l_1)} f(u_1, u_2) \end{aligned}$$

Defining multi-affine approximations

- Iteratively applying linear interpolation on each dimension

- Let $\mu_f(k; l_k, u_k)$ the function obtained by applying a linear interpolation of f on the k^{th} dimension at two points l_k and u_k. Let

$$\mathbf{p}(x) = \mu_f(1; l_1, u_1) \text{ and } \mathbf{q}(x) = \mu_f(2; l_2, u_2).$$

  Then, for $x \in [l_1, u_1] \times [l_2, u_2]$

$$f_{\mathbf{h}}(x) = \mu_{\mathbf{p}}(2; l_2, u_2) = \mu_{\mathbf{q}}(1; l_1, u_1).$$

Piecewise affine vs. Piecewise multi-affine

⋆ Advantage comparison

Simplicial meshes	Rectangular meshes
	smaller number of cells
	less complex geometric structure
available techniques for approximate systems	???

⋆ Reachability computations for piecewise multi-affine systems with uncertain input:
Abstraction by projection [Asarin Dang 2004]

- Use projection to obtain a uncertain bilinear control system
- Then, use our reachability technique for bilinear control systems

Method for polynomial systems [Dang 2006]

Plan

Hybridization methods for non-linear systems

- Principle
- Construction of approximate systems
  - Piecewise affine approximation
  - Piecewise multi-affine approximation
- **Reachability computation methods for polynomial systems**

Reachability computation method for polynomial systems

We consider a polynomial differential equation:

$$\dot{\mathbf{x}}(t) = g(\mathbf{x}(t))$$

Given a time step r, it is possible to find a numerical scheme (for detail see the HSCC workshop)

$$\mathbf{x}_{k+1} = Q(\mathbf{x}_k, r)$$

where $Q(\mathbf{x}_k)$ is a polynomial in $\mathbf{x}_k$.
We compute this map using Bézier techniques

The approximation error is $\mathcal{O}(r^2)$

Bézier simplices - definitions

A multi-index $\mathbf{i} = (\mathbf{i}[1], \ldots, \mathbf{i}[n+1])$ is a vector of non-negative integers. The norm $||\mathbf{i}|| = \sum_{j=1}^{n+1} \mathbf{i}[j]$

Δ: n-dimensional simplex with vertices $\{\mathbf{v}_1, \ldots, \mathbf{v}_{n+1}\}$. Given a point $\mathbf{x} \in \Delta$, $\lambda(\mathbf{x}) = (\lambda_1(\mathbf{x}), \ldots, \lambda_{n+1}(\mathbf{x}))$: barycentric coordinates of $\mathbf{x}$ w.r.t the vertices of Δ, that is, $\mathbf{x} = \sum_{k=1}^{n+1} \lambda_k(\mathbf{x})\mathbf{v}_k$ and $\sum_{k=1}^{n+1} \lambda_k(\mathbf{x}) = 1$.

A Bézier simplex of degree d is a polynomial defined as:

$$\boldsymbol{\pi}(\mathbf{x}) = \sum_{||\mathbf{i}||=d} \mathbf{p}_\mathbf{i} B_{\mathbf{i},d}(\lambda_1(\mathbf{x}), \ldots, \lambda_{n+1}(\mathbf{x}))$$

where $B_{\mathbf{i},d}(y_1, \ldots, y_{n+1}) = \binom{d}{\mathbf{i}} y_1^{\mathbf{i}[1]} y_2^{\mathbf{i}[2]} \ldots y_{n+1}^{\mathbf{i}[n+1]}$ (Bernstein polynomials), with the multimonial coefficient $\binom{d}{\mathbf{i}} = \frac{d!}{\mathbf{i}[1]!\mathbf{i}[2]!\ldots\mathbf{i}[n+1]!}$.

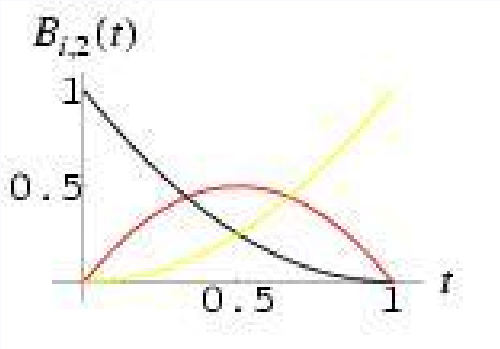
Each vector $\mathbf{p}_\mathbf{i} \in \mathbb{R}^n$ is called a Bézier control point and all such $\mathbf{p}_\mathbf{i}$ form the Bézier control net of $\boldsymbol{\pi}$ with respect to Δ.

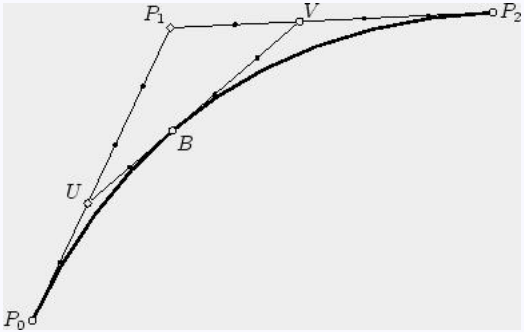
Example - Bézier curves

Dimension $n = 1$. With $d = 2$, multi-indices with $||\mathbf{i}|| = 2$: $\{(0,2),(1,1),(2,0)\}$.

A quadratic Bézier curve is: $\boldsymbol{\pi}(\mathbf{x}) = \sum_{\mathbf{i}} \mathbf{p}_{\mathbf{i}} B_{\mathbf{i},2}(\lambda_1(\mathbf{x}), \lambda_2(\mathbf{x}))$, for $\mathbf{x} \in [\mathbf{v}_1, \mathbf{v}_2]$.

There are 3 control points $\mathbf{p}_{\mathbf{i}}$.



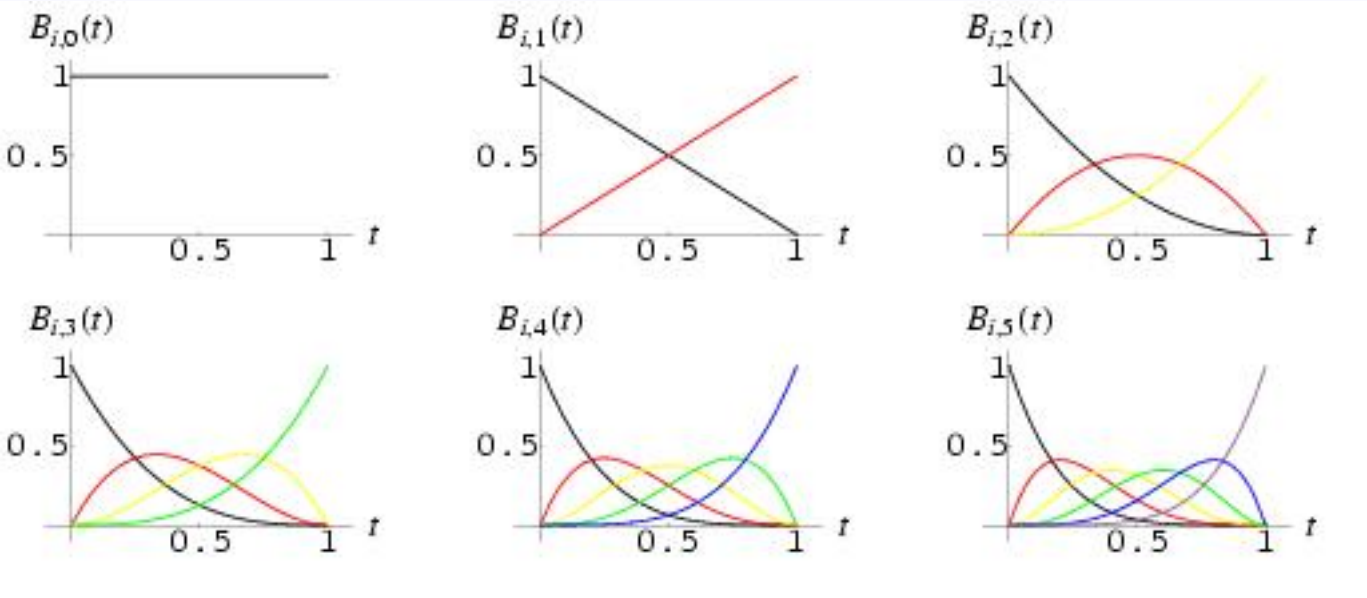


Bernstein polynomials

$$B_{\mathbf{i},2}(t) = B_{\mathbf{i},2}(y_1, y_2) = \binom{2}{\mathbf{i}} y_1^{\mathbf{i}[1]} y_2^{\mathbf{i}[2]}, y_1 = t \in [0,1], y_2 = 1 - t$$

Example - Bézier curves

Dimension $n = 1$.



Bernstein polynomials

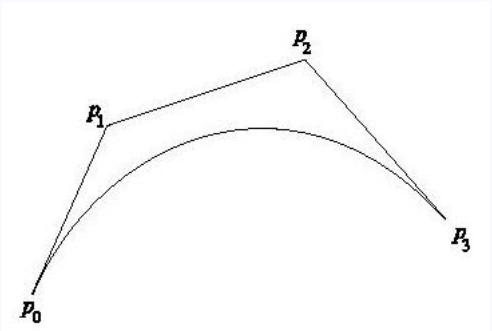
$$B_{\mathbf{i},d}(t) = B_{\mathbf{i},d}(y_1, y_2) = \binom{d}{\mathbf{i}} y_1^{\mathbf{i}[1]} y_2^{\mathbf{i}[2]}, y_1 = t \in [0,1], y_2 = 1 - t$$

Bézier simplices - Shape Properties

Any polynomial can be written in form of a Bézier simplex. Given an arbitrary point $\mathbf{x} \in \Delta$,

1. **Convex hull property**: the point $\boldsymbol{\pi}(\mathbf{x})$ lies inside the convex hull of the control net
2. **End-point interpolation property**: the polynomial $\boldsymbol{\pi}$ interpolates the control net at the corner control points.

Number of control points $\mathbf{p_i}$ is $\binom{d+n}{n} = \frac{(d+n)!}{d!\,n!}$.



$\Rightarrow$ Using the convex hull of the Bézier control net as a **tight over-approximation** of $\boldsymbol{\pi}(\Delta)$.

Computing the Bézier control points

Problem: For a polynomial $\boldsymbol{\pi}$ given in monomial form and a simplex $\Delta = conv\{\mathbf{v}_1, \ldots, \mathbf{v}_{n+1}\}$, we want to compute the control net of $\boldsymbol{\pi}$ with respect to Δ.

Blossoming principe For any polynomial $\boldsymbol{\pi} : \mathbb{R}^n \to \mathbb{R}^n$ of degree d, there is a unique symmetric d-affine map $\beta : \mathbb{R}^{nd} \to \mathbb{R}^n$ such that for all $\mathbf{x} \in \mathbb{R}^n$: $p(\mathbf{x}, \ldots, \mathbf{x}) = \boldsymbol{\pi}(\mathbf{x})$.
Recall: $q(\mathbf{x}_1, \ldots, \mathbf{x}_d)$ is *d-affine* if affine when all but one of its arguments are fixed; symmetric if independent of the ordering of the arguments.

Connection with the Bézier control net:

$$\mathbf{p_i} = \beta(\underbrace{\mathbf{v}_1, \ldots, \mathbf{v}_1}_{i[1]}, \underbrace{\mathbf{v}_2, \ldots, \mathbf{v}_2}_{i[2]}, \ \ldots, \ \underbrace{\mathbf{v}_{n+1}, \ldots, \mathbf{v}_{n+1}}_{i[n+1]})$$

$\Rightarrow$ To compute $\mathbf{p_i}$ we can evaluate the blossom with some particular arguments

Evaluating blossom values

Illustrate the computation of blossom values of polynomial $(\mathbf{x}[i])^h(\mathbf{x}[j])^k$.

$$\beta^d_{h,k}(\mathbf{u}_1,\ldots,\mathbf{u}_d) = \frac{1}{\binom{d}{h}\binom{d-h}{k}} \sum_{\substack{I\cup J\subseteq\{1,\ldots,d\},\\ |I|=h,|J|=k,I\cap J=\emptyset}} \prod_{r\in I}\mathbf{u}_r[i]\prod_{s\in J}\mathbf{u}_s[j]$$

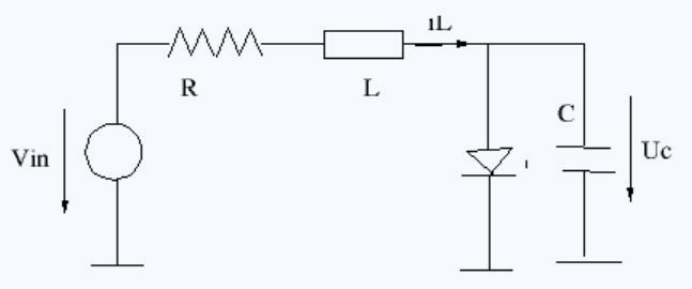
Denote $\sigma^d_{h,k} = \frac{1}{\binom{d}{h}\binom{d-h}{k}}\beta^d_{h,k}(\mathbf{u}_1,\mathbf{u}_2,\ldots,\mathbf{u}_d)$.

The symmetric function σ: choose h i^{th} coordinates of d argument points and k j^{th} coordinates and form their product, then sum these products over all possible choices.

$$\begin{cases} \sigma^d_{h,k} &= \sigma^{d-1}_{h,k} + \mathbf{u}_d[i]\sigma^{d-1}_{h-1,k} + \mathbf{u}_d[j]\sigma^{d-1}_{h,k-1} \text{ if } h,k\geq 0 \text{ and } h+k\geq 1,\\ \sigma^d_{0,0} &= 0 \end{cases}$$

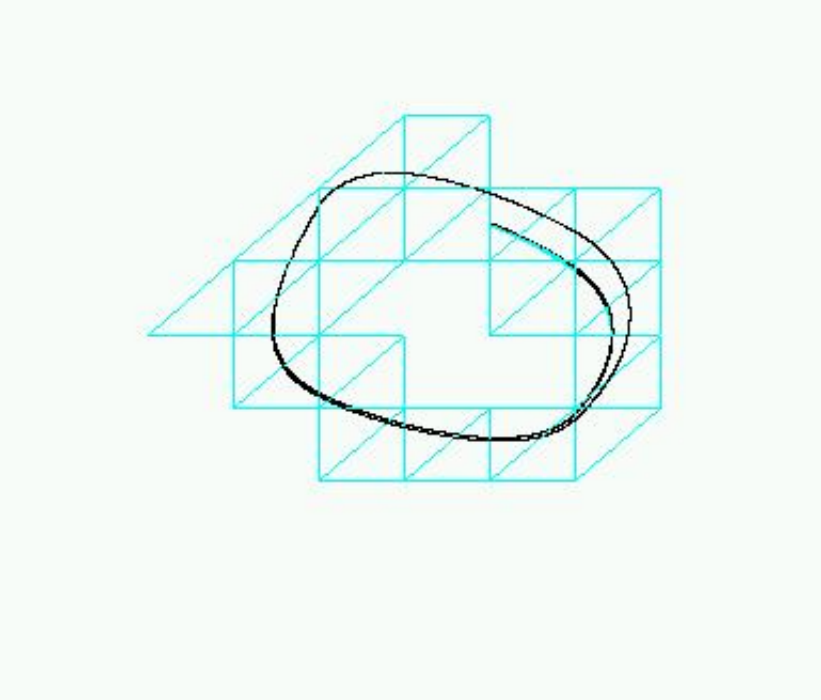
Complexity $O(d^3)$

Tunnel Diode Osccilator



$$
\begin{aligned}
-\frac{d}{dt}u_c &= \frac{1}{C}(0.43\,u_c^3 - 2.69\,u_c^2 + 4.56\,u_c + 1.4\,(1-\lambda)\,u_c) - \frac{1}{C}\,i_L \\
\frac{d}{dt}i_L &= -\frac{1}{L}\,u_c - \frac{R}{L}\,i_L + \frac{1}{L}\,V_{in}
\end{aligned}
$$

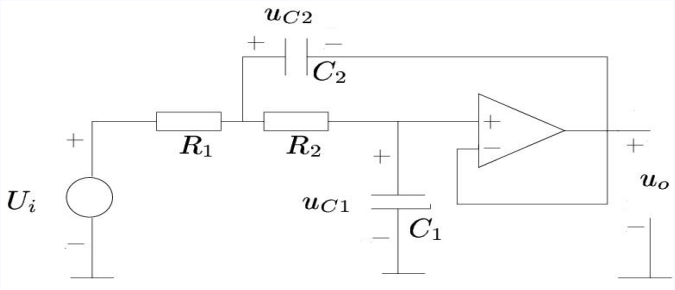
Tunnel Diode Osccilator



The result shows a limit cycle

Biquad low pass filter

[Hartong,Hedrich,Barke 2002]



$$\dot{u}_{C1} = \frac{u_{C2} + u_o - u_{C1}}{C_1 R_2} \qquad \dot{u}_{C2} = \frac{U_i - u_{C2} - u_o}{C_2 R_1} - \frac{u_{C2} + u_o - u_{C1}}{C_2 R_2}, \tag{1}$$

$$u_o - V_{max} \tanh(\frac{(u_{C2} - u_o)V_e}{V_{max}}) + U_{om} = 0, \tag{2}$$

$$U_{om} = \mathcal{V}(i_0), \quad i_o = -C_2\, \dot{u}_{C2}, \tag{3}$$

$$\mathcal{V}(i_o) = K_1 i_o + 0.5\sqrt{K_1 i_o^2 - 2K_2 i_o I_s + K_1 I_s^2 + K_2} - 0.5\,\sqrt{K_1 i_o^2 + 2K_2 i_o I_s + K_1 I_s^2 + K_2}. \tag{4}$$

Differentiating (2) $\Rightarrow$ nonlinear ODEs on a manifold with variables $z = (u_{C1}, u_{C2}, u_o)$

Biquad low pass filter: Approximation by hybridization

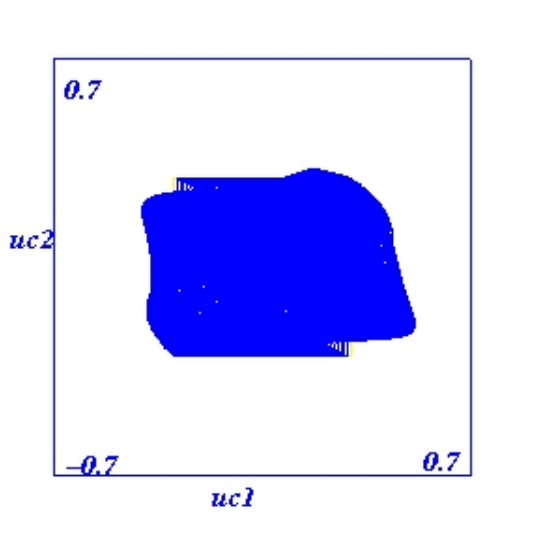
- *tanh* approximated by polynomial or piecewise affine ⇒ loss of important behaviors of the circuit
- the nonlinear characteristics $U_{om} = \mathcal{V}(i_o)$ can be approximated by a piecewise-affine function:

$$U_{om} = \begin{cases} 1e10\, i_o + 0.5e8 & \textbf{if } i_o \leq -i_s \\ 0 & \textbf{if } -i_s < i_o < i_s \\ 1e10\, i_o - 0.5e8 & \textbf{if } i_o \geq i_s \end{cases} \tag{5}$$

⇒ Hybrid automaton with 3 continuous modes

Biquad low pass filter: verification results

The property to verify is the ***absence of overshoots*** .



- $C_1 = 0.5e-8$, $C_2 = 2e-8$, and $R_1 = R_2 = 1e6$ (highly damped case)
- The initial set: $u_{C1} \in [-0.3, 0.3]$, $u_{C2} \in [-0.3, 0.3]$ and $u_o \in [-0.2, 0.2]$

Ongoing work

- Hybridization: hierarchical mesh construction
- Using higher degree approximants, such as piecewise quadratic
- Other new classes of properties preserved by hybridization