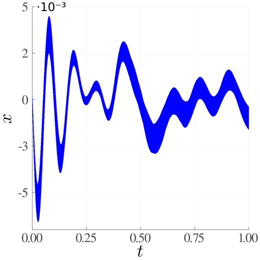
A gentle introduction to reachability analysis for dynamical systems

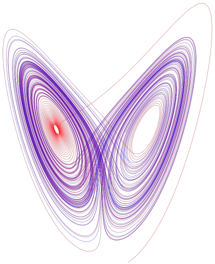


Christian Schilling

Aalborg University, Denmark

July 13, 2022

AALBORG UNIVERSITET



Dynamical systems

- **Continuous-time systems** modeled by **ordinary differential equations**

$$\dot{x}(t) = f(x(t)) \qquad (x \in \mathbb{R}^n)$$

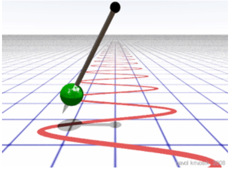
- **Initial-value problem**: Given an initial state $x_0 \in \mathbb{R}^n$, determine the **solution/trajectory** following f

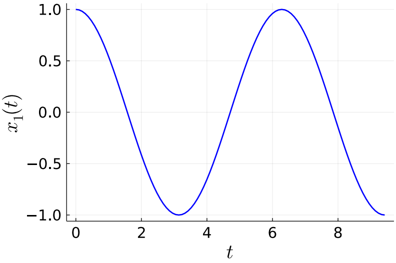
Simulation

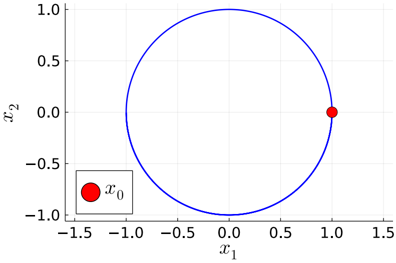
- Traditionally we use **simulations** to understand such systems

Example: **Harmonic oscillator**

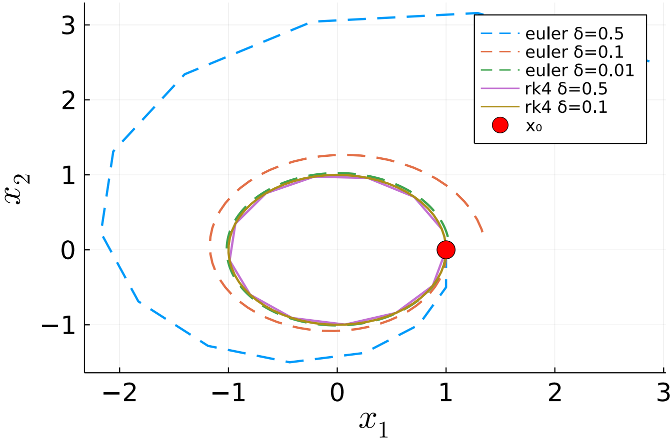
$$\dot{x} = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} x, \quad x_0 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$





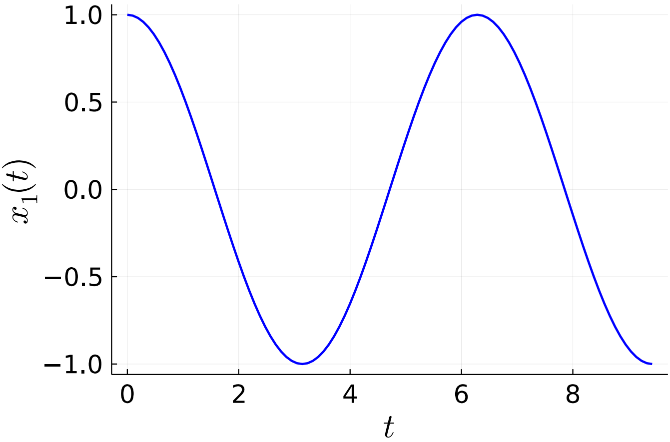


Problems with simulation (1): Precision



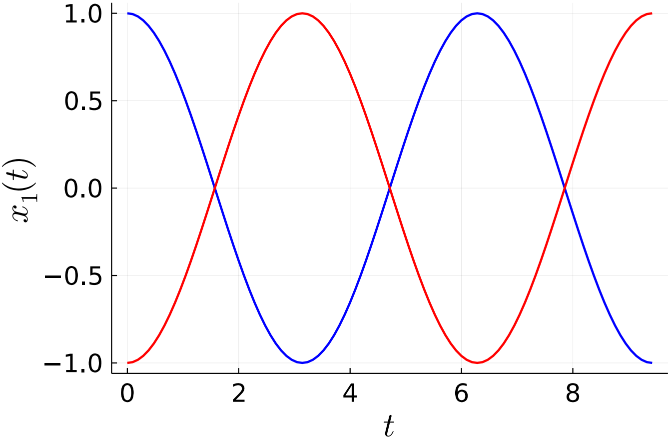
- Approaches: **reduce the time step**, **adaptive solvers**, ...

Problems with simulation (2): Coverage



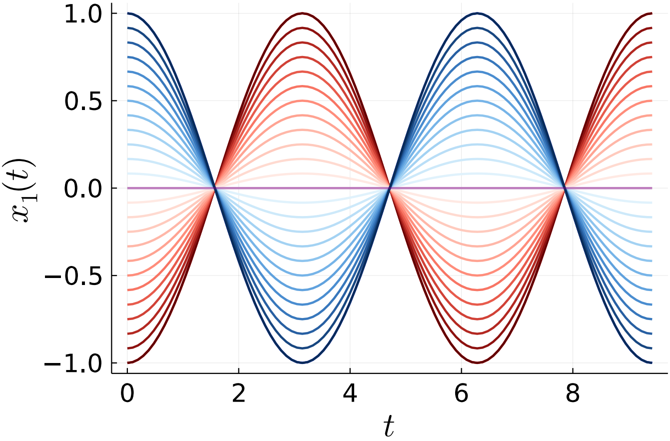
- Consider a **set of initial states** $\boldsymbol{x_1(0) \in [-1, 1]}$

Problems with simulation (2): Coverage



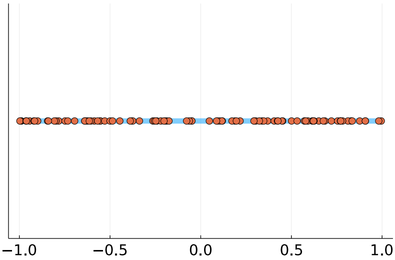
- Consider a **set of initial states** $\mathbf{x_1(0) \in [-1, 1]}$

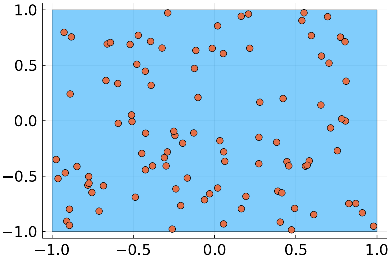
Problems with simulation (2): Coverage



- Approach: **sample the corners** (if the set has corners) (sufficient for **linear systems** only)

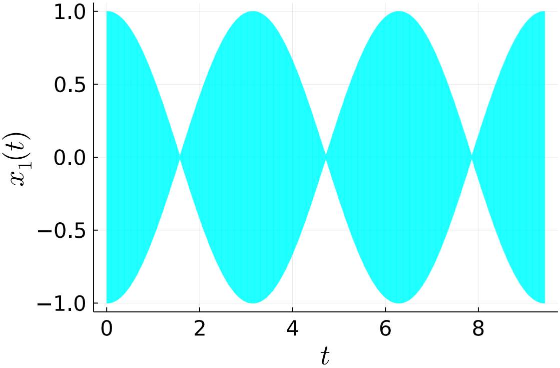
Problems with simulation (3): Dimensionality





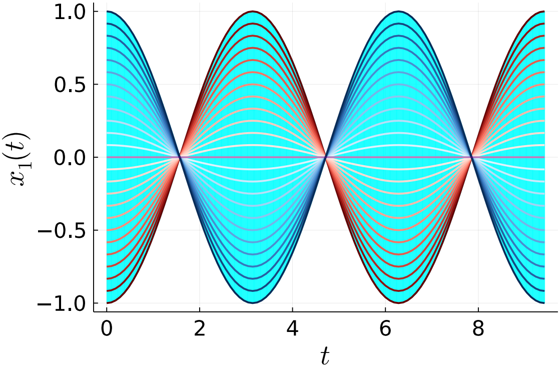
- Sampling coverage is low for higher dimensions
- Vertex sampling: n-dimensional hyperrectangle has 2^n vertices

Reachability analysis



- **Enclose** the **reachable states** $\{x(t) : x(0) \in \mathcal{X}_0, t \geq 0\}$
- **Set-based simulations** (same intuition)
- **Rigorous proof method** (captures **all solutions**)
- **Fast** (linear systems with thousands of dimensions)

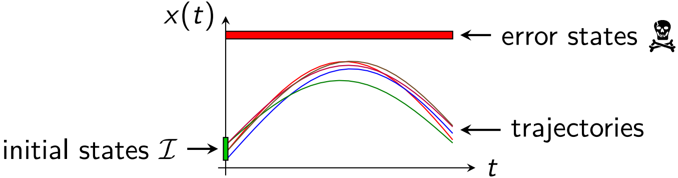
Reachability analysis



- **Enclose** the **reachable states $\{x(t) : x(0) \in \mathcal{X}_0, t \geq 0\}$**
- **Set-based simulations** (same intuition)
- **Rigorous proof method** (captures **all solutions**)
- **Fast** (linear systems with thousands of dimensions)

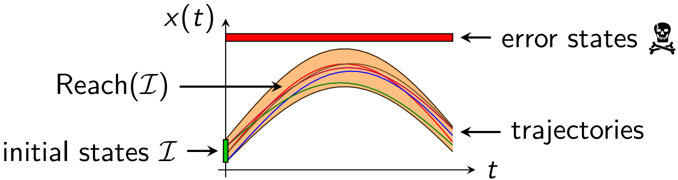
Safety verification

- Task: Verify that **no trajectory** leads to an **error state**



Safety verification

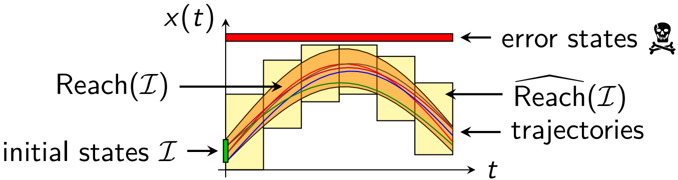
- Task: Verify that **no trajectory** leads to an **error state**
- Equivalent to showing **Reach**$(\mathcal{I}) \cap$ ☠ $= \emptyset$
- Only **decidable** under strong restrictions



Safety verification

- Task: Verify that **no trajectory** leads to an **error state**
- Equivalent to showing **Reach**$(\mathcal{I}) \cap$ ☠ $= \emptyset$
- Only **decidable** under strong restrictions
- Showing $\widehat{\textbf{Reach}}(\mathcal{I}) \cap$ ☠ $= \emptyset$ is sufficient

  ↑ overapproximation of Reach$(\mathcal{I})$

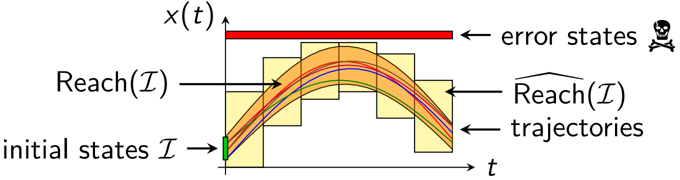


Safety verification

- Task: Verify that **no trajectory** leads to an **error state**
- Equivalent to showing $\textbf{Reach}(\mathcal{I}) \cap$ ☠ $= \emptyset$
- Only **decidable** under strong restrictions
- Showing $\widehat{\textbf{Reach}}(\mathcal{I}) \cap$ ☠ $= \emptyset$ is sufficient

overapproximation of $\text{Reach}(\mathcal{I})$

restriction: bounded time



JuliaReach[1]

- Open-source **reachability toolbox**
  https://github.com/JuliaReach
- Joint work with **Marcelo Forets** and many others
- Won ARCH-COMP friendly competition 2018 and 2020

Linear systems (times in seconds)

tool	**BLDC01**	**CBF01**	**PLAD04-42**	**BRKDC01**
dimension	48	200	9	4
CORA	2.9	30	1.4	12
HyDRA	0.426	–	1.83	–
JuliaReach	0.0096	12	0.031	0.82
SpaceEx	1.6	319	0.36	21

[1] S. Bogomolov, M. Forets, G. Frehse, K. Potomkin, and C. Schilling. *HSCC.* 2019.

JuliaReach[1]

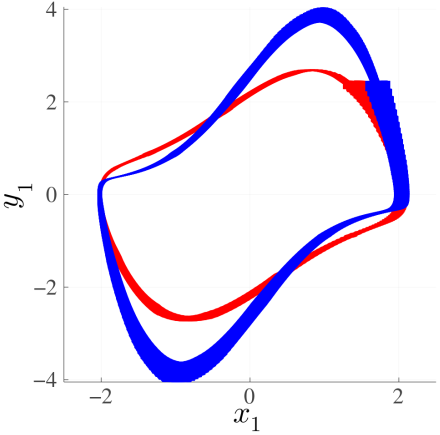
- Open-source **reachability toolbox**
  https://github.com/JuliaReach
- Joint work with **Marcelo Forets** and many others
- Won ARCH-COMP friendly competition 2018 and 2020

Nonlinear systems (times in seconds)

tool	**CVDP20**	**LALO20-W0.1**	**LOVO21**	**SPRE21**
dimension	4	7	2	4
Ariadne	11	31	8	–
CORA	7.7	38	23	26
DynIbex	510	1,851	75	144
JuliaReach	1.5	6.4	3.4	24

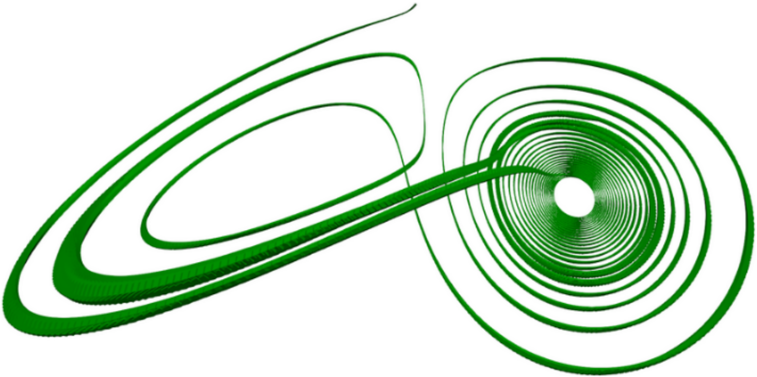
[1] S. Bogomolov, M. Forets, G. Frehse, K. Potomkin, and C. Schilling. *HSCC*. 2019.

Examples



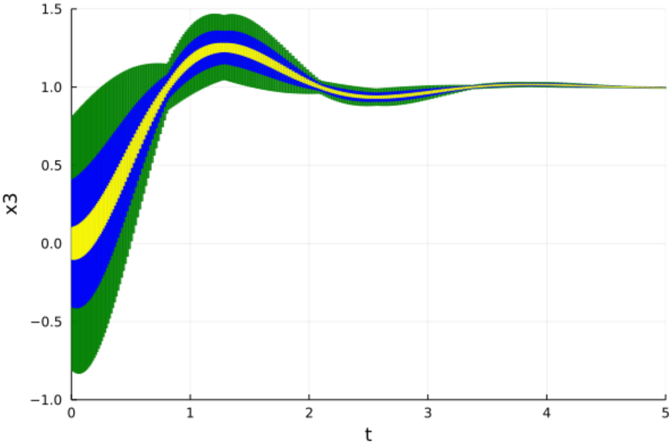
Van der Pol oscillator (limit cycle)

Examples



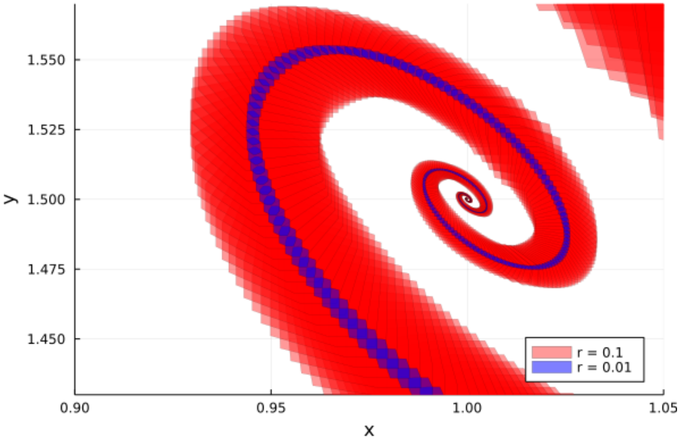
Lorenz system (chaotic behavior)

Examples



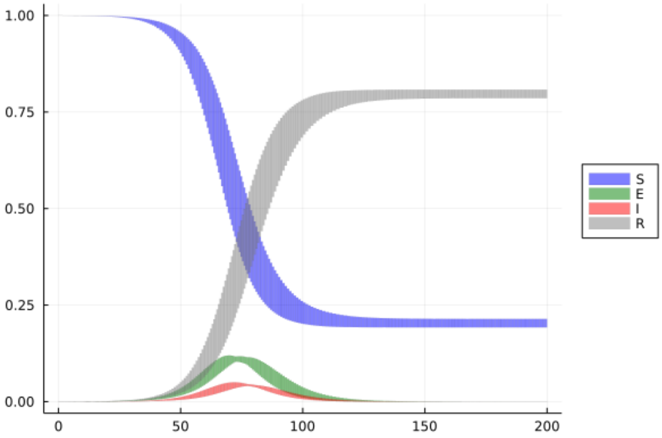
Quadrotor (robotics)

Examples



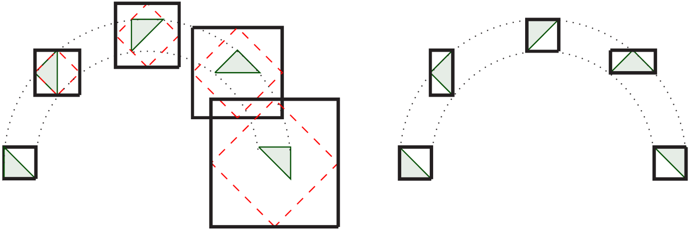
Brusselator (chemical reaction)

Examples



SEIR model (epidemiology)

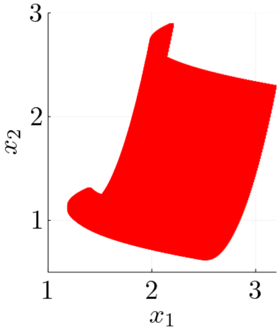
Reachability for nonlinear systems



- **Reachtube construction** computes a **sequence of sets** until a time horizon
- Checking whether a state is reachable is **undecidable**
  Hence the true reachable states are **not computable**
- **Overapproximation** or **underapproximation**
- **Wrapping effect**
- Alternative approaches: **invariant generation**, **abstraction**

Taylor models

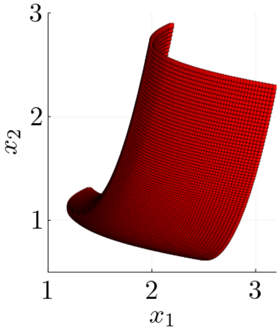
- **Truncated polynomials with interval remainder**
- **Rigorous arithmetic**



$$p_1(x) = 1.7 - 0.5x_1 + 0.4x_2 + 0.6x_1^2 + [-0.001, 0.001]$$
$$p_2(x) = 1.2 + 0.3x_1 + 0.8x_2 + 0.6x_2^2 + [-0.001, 0.001]$$
$$x \in [-1, 1]^2$$

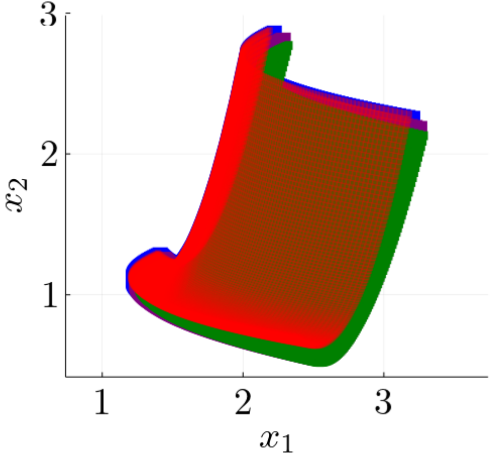
Taylor models

- **Truncated polynomials with interval remainder**
- **Rigorous arithmetic**



$$p_1(x) = 1.7 - 0.5x_1 + 0.4x_2 + 0.6x_1^2 + [-0.001, 0.001]$$
$$p_2(x) = 1.2 + 0.3x_1 + 0.8x_2 + 0.6x_2^2 + [-0.001, 0.001]$$
$$x \in [-1, 1]^2$$

Taylor models for reachability



- Wrap in another **Taylor model over time** t
- **Forward computation** (here: two steps)

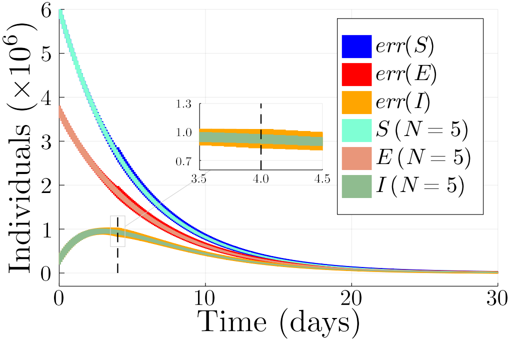
Taylor models for reachability

$$1.7 - 0.5\,x_1 + 0.4\,x_2 + 0.6\,x_1^2 + (\,1.2 + 0.3\,x_1 + 0.8\,x_2 + 0.6\,x_2^2)\,t + (\,-\,0.85 + 0.25\,x_1 - 0.2\,x_2 - 0.3\,x_1^2)\,t^2 + (\,-\,0.19999999999999998 - 0.049999999999999996\,x_1 - 0.13333333333333333\,x_2 - 0.09999999999999999\,x_2^2)\,t^3 + (\,0.07083333333333333 - 0.020833333333333332\,x_1 + 0.016666666666666666\,x_2 + 0.024999999999999998\,x_1^2)\,t^4 + (\,0.009999999999999998 + 0.0024999999999999996\,x_1 + 0.006666666666666666\,x_2 + 0.004999999999999999\,x_2^2)\,t^5 + (\,-\,0.002361111111111111 + 0.0006944444444444445\,x_1 - 0.0005555555555555556\,x_2 - 0.0008333333333333332\,x_1^2)\,t^6 + (\,-\,0.00023809523809523804 - 5.952380952380951e{-5}\,x_1 - 0.00015873015873015873\,x_2 - 0.00011904761904761902\,x_2^2)\,t^7 + (\,4.2162698412698416e{-5} - 1.240079365079365e{-5}\,x_1 + 9.92063492063492e{-6}\,x_2 + 1.4880952380952378e{-5}\,x_1^2)\,t^8 + [-1.00001e{-10},\ 1.00001e{-10}]$$

$$1.2 + 0.3\,x_1 + 0.8\,x_2 + 0.6\,x_2^2 + (\,-\,1.7 + 0.5\,x_1 - 0.4\,x_2 - 0.6\,x_1^2)\,t + (\,-\,0.6 - 0.15\,x_1 - 0.4\,x_2 - 0.3\,x_2^2)\,t^2 + (\,0.2833333333333333 - 0.08333333333333333\,x_1 + 0.06666666666666667\,x_2 + 0.09999999999999999\,x_1^2)\,t^3 + (\,0.049999999999999996 + 0.012499999999999999\,x_1 + 0.03333333333333333\,x_2 + 0.024999999999999998\,x_2^2)\,t^4 + (\,-\,0.014166666666666666 + 0.004166666666666667\,x_1 - 0.003333333333333333\,x_2 - 0.004999999999999999\,x_1^2)\,t^5 + (\,-\,0.0016666666666666663 - 0.0004166666666666666\,x_1 - 0.0011111111111111111\,x_2 - 0.0008333333333333332\,x_2^2)\,t^6 + (\,0.00033730158730158733 - 9.92063492063492e{-5}\,x_1 + 7.936507936507937e{-5}\,x_2 + 0.00011904761904761902\,x_1^2)\,t^7 + (\,2.976190476190476755e{-5} + 7.440476190476189e{-6}\,x_1 + 1.984126984126984e{-5}\,x_2 + 1.4880952380952378e{-5}\,x_2^2)\,t^8 + [-1.00001e{-10},\ 1.00001e{-10}]$$

- The first set (**blue**) in time interval **[0, 0.225497]**

Linearization

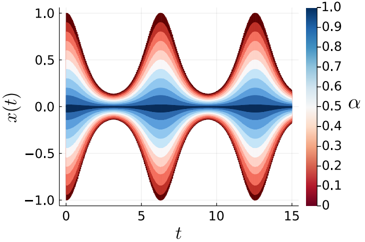


- **Carleman linearization** turns a **polynomial system** into an **infinite-dimensional linear system**
- **Truncation** leads to an **approximate system**
- Can **bound the approximation error** for **dissipative, weakly-nonlinear** systems $\rightsquigarrow$ **reachability algorithm**[1]

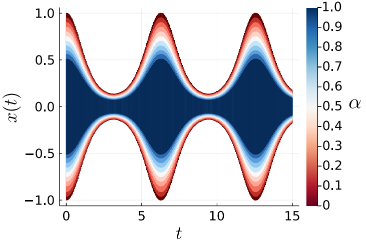
[1] M. Forets and C. Schilling. *RP*. 2021.

Work in progress: Probabilistic initial conditions

Harmonic oscillator



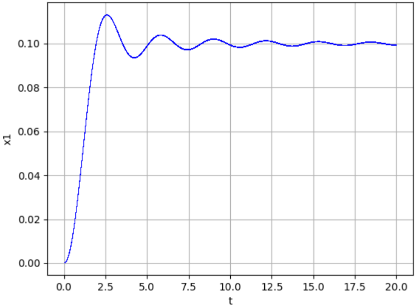
$\mathcal{X}_0 \sim U(-1, 1)$



$\mathcal{X}_0 \sim U([-1, 0], [0, 1])$

- Propagate **p-boxes** through **Taylor models**

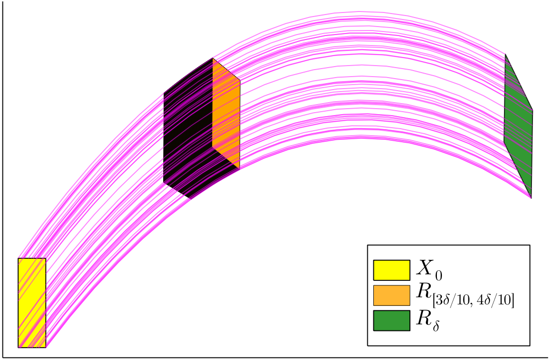
Linear systems



MNA5 (10,913 dimensions)

- **Linear systems**: $\dot{x}(t) = Ax(t)$
- **Reachability problem** still **not decidable**
- **Arbitrary precision** and **fast** (solution above: **90 sec**)

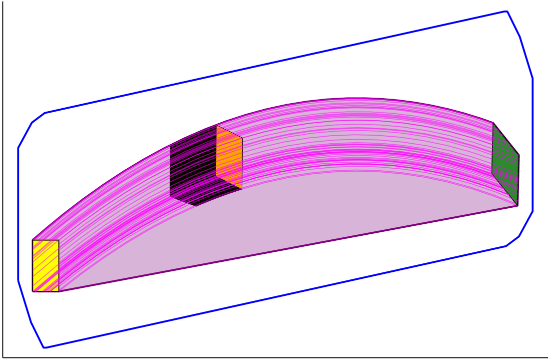
Time discretization



- **Linear systems** allow for **wrapping-free** algorithms based on **efficient set representations**[1]

[1] M. Forets and C. Schilling. *iFM*. 2022.

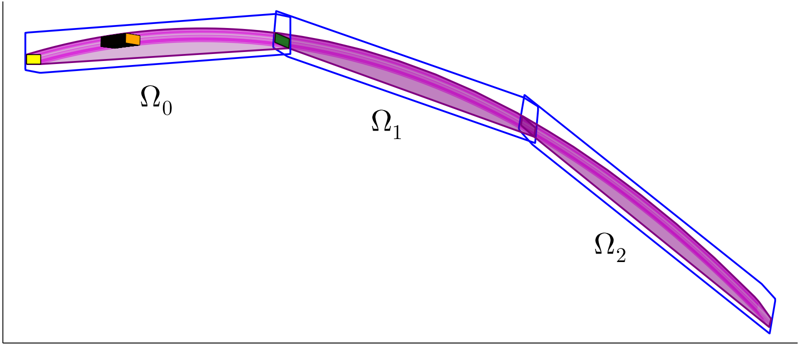
Time discretization



- **Linear systems** allow for **wrapping-free** algorithms based on **efficient set representations**[1]

[1] M. Forets and C. Schilling. *iFM*. 2022.

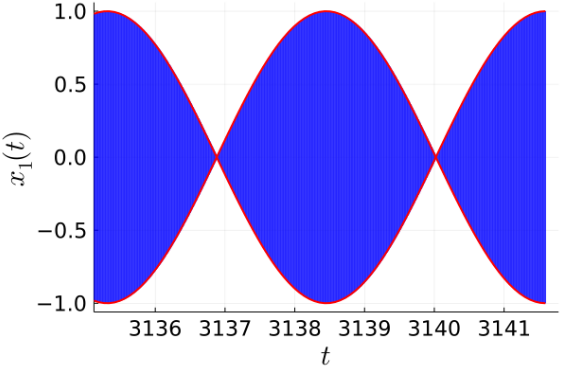
Time discretization



- **Linear systems** allow for **wrapping-free** algorithms based on **efficient set representations**[1]

[1] M. Forets and C. Schilling. *iFM*. 2022.

Wrapping-free computation



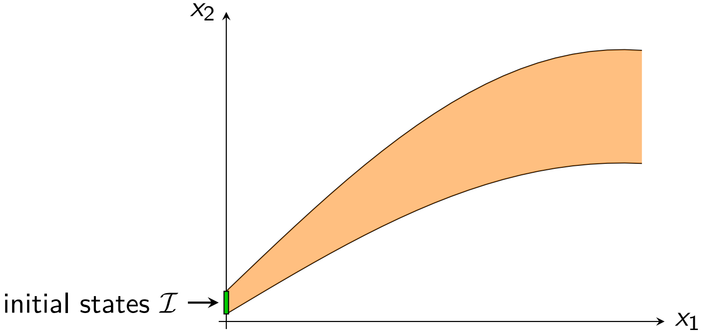
Harmonic oscillator after 500 periods

Time step: **0.01** $\rightsquigarrow$ **314,160** steps

Computation time: **0.33 seconds**

Decomposition approach[1,2]

Standard algorithm

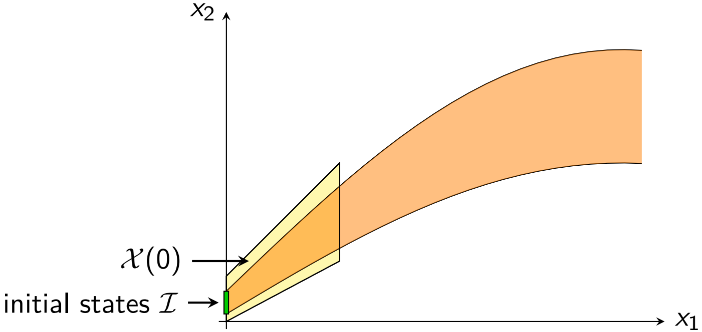


[1] S. Bogomolov, M. Forets, G. Frehse, F. Viry, A. Podelski, and C. Schilling. *HSCC.* 2018.

[2] S. Bogomolov, M. Forets, G. Frehse, A. Podelski, and C. Schilling. *Inf. Comput.* (2022).

Decomposition approach[1,2]

Standard algorithm



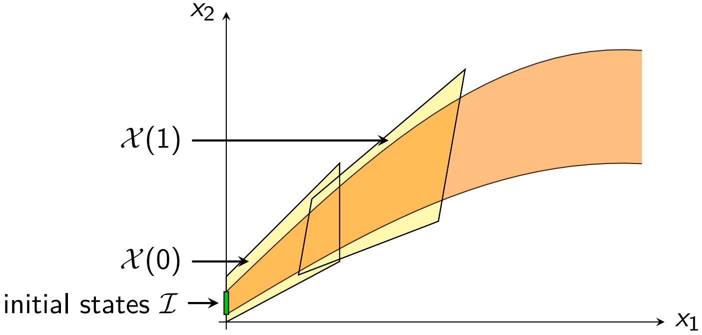
[1] S. Bogomolov, M. Forets, G. Frehse, F. Viry, A. Podelski, and C. Schilling. *HSCC*. 2018.

[2] S. Bogomolov, M. Forets, G. Frehse, A. Podelski, and C. Schilling. *Inf. Comput.* (2022).

Decomposition approach[1,2]

Standard algorithm

$\mathcal{X}(1) = \Phi \cdot \mathcal{X}(0)$



[1] S. Bogomolov, M. Forets, G. Frehse, F. Viry, A. Podelski, and C. Schilling. *HSCC.* 2018.

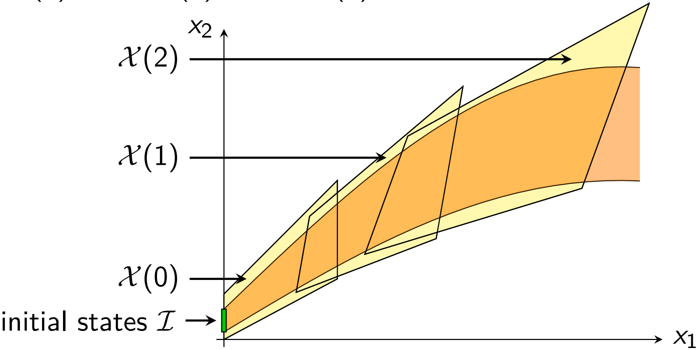
[2] S. Bogomolov, M. Forets, G. Frehse, A. Podelski, and C. Schilling. *Inf. Comput.* (2022).

Decomposition approach[1,2]

Standard algorithm

$\mathcal{X}(1) = \Phi \cdot \mathcal{X}(0)$

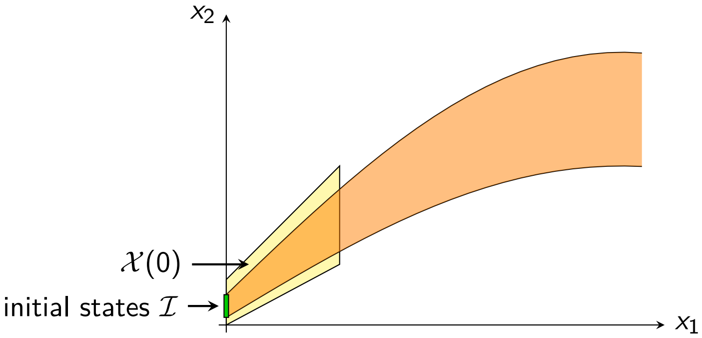
$\mathcal{X}(2) = \Phi \cdot \mathcal{X}(1) = \Phi^2 \cdot \mathcal{X}(0)$



[1] S. Bogomolov, M. Forets, G. Frehse, F. Viry, A. Podelski, and C. Schilling. *HSCC.* 2018.

[2] S. Bogomolov, M. Forets, G. Frehse, A. Podelski, and C. Schilling. *Inf. Comput.* (2022).

Decomposition approach[1,2]



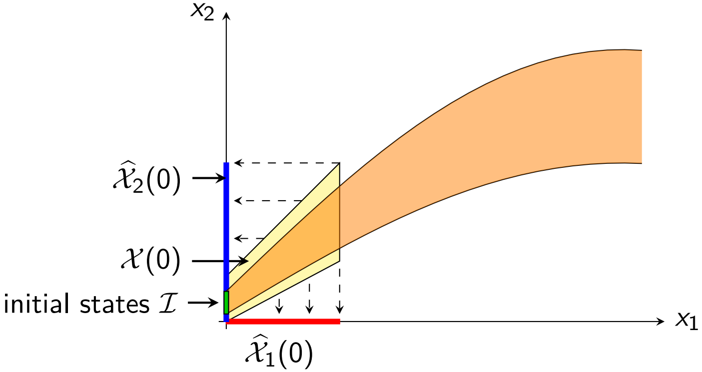
[1] S. Bogomolov, M. Forets, G. Frehse, F. Viry, A. Podelski, and C. Schilling. *HSCC*. 2018.

[2] S. Bogomolov, M. Forets, G. Frehse, A. Podelski, and C. Schilling. *Inf. Comput.* (2022).

Decomposition approach[1,2]

Decompose $\mathcal{X}(0)$ into **low-dimensional** sets $\widehat{\mathcal{X}}_1(0)$ and $\widehat{\mathcal{X}}_2(0)$

(Note: In general, we do not need to go down to 1D)

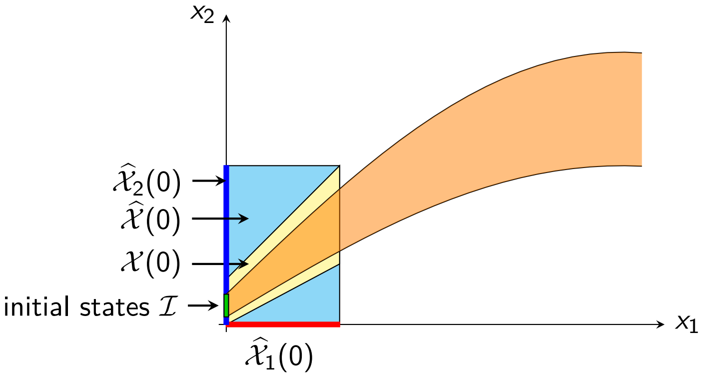


[1] S. Bogomolov, M. Forets, G. Frehse, F. Viry, A. Podelski, and C. Schilling. *HSCC*. 2018.

[2] S. Bogomolov, M. Forets, G. Frehse, A. Podelski, and C. Schilling. *Inf. Comput.* (2022).

Decomposition approach[1,2]

Define $\widehat{\mathcal{X}}(k) := \widehat{\mathcal{X}}_1(k) \times \widehat{\mathcal{X}}_2(k)$



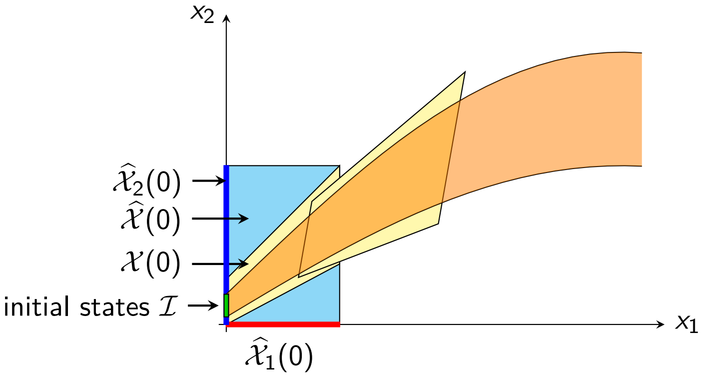
[1] S. Bogomolov, M. Forets, G. Frehse, F. Viry, A. Podelski, and C. Schilling. *HSCC*. 2018.

[2] S. Bogomolov, M. Forets, G. Frehse, A. Podelski, and C. Schilling. *Inf. Comput.* (2022).

Decomposition approach[1,2]

Define $\widehat{\mathcal{X}}(k) := \widehat{\mathcal{X}}_1(k) \times \widehat{\mathcal{X}}_2(k)$

Standard: $\mathcal{X}(k) = \Phi^k \cdot \mathcal{X}(0)$



[1] S. Bogomolov, M. Forets, G. Frehse, F. Viry, A. Podelski, and C. Schilling. *HSCC*. 2018.

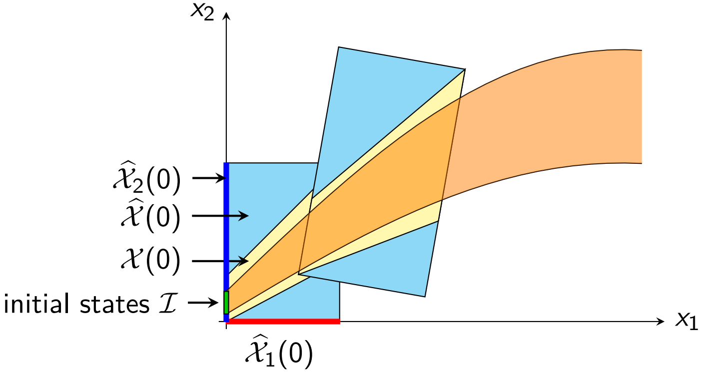
[2] S. Bogomolov, M. Forets, G. Frehse, A. Podelski, and C. Schilling. *Inf. Comput.* (2022).

Decomposition approach[1,2]

Define $\hat{\mathcal{X}}(k) := \hat{\mathcal{X}}_1(k) \times \hat{\mathcal{X}}_2(k)$

Standard: $\mathcal{X}(k) = \Phi^k \cdot \mathcal{X}(0)$

Decomposed: $\hat{\mathcal{X}}(k) = \Phi^k \cdot \hat{\mathcal{X}}(0)$?



[1] S. Bogomolov, M. Forets, G. Frehse, F. Viry, A. Podelski, and C. Schilling. *HSCC.* 2018.

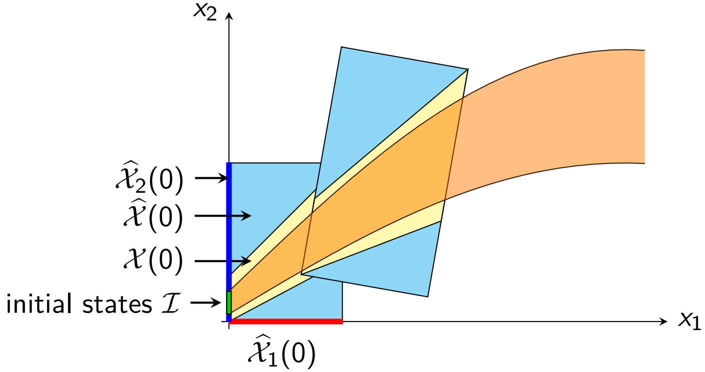
[2] S. Bogomolov, M. Forets, G. Frehse, A. Podelski, and C. Schilling. *Inf. Comput.* (2022).

Decomposition approach[1,2]

Define $\widehat{\mathcal{X}}(k) := \widehat{\mathcal{X}}_1(k) \times \widehat{\mathcal{X}}_2(k)$

Standard: $\mathcal{X}(k) = \Phi^k \cdot \mathcal{X}(0)$

Decomposed: $\widehat{\mathcal{X}}_i(k) = \bigoplus_j \Phi^k_{i,j} \cdot \widehat{\mathcal{X}}_j(0)$

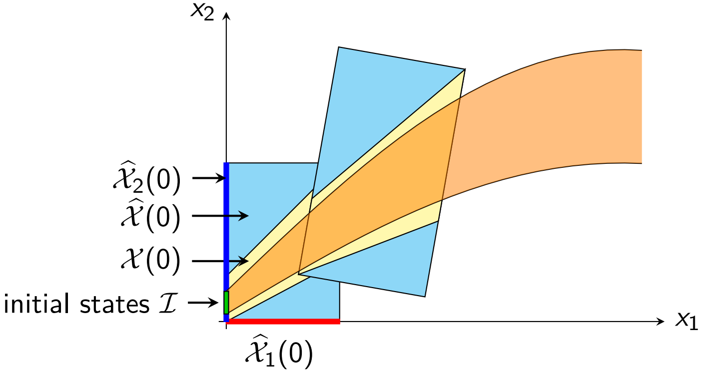


[1] S. Bogomolov, M. Forets, G. Frehse, F. Viry, A. Podelski, and C. Schilling. *HSCC*. 2018.

[2] S. Bogomolov, M. Forets, G. Frehse, A. Podelski, and C. Schilling. *Inf. Comput.* (2022).

Decomposition approach[1,2]

$$\widehat{\mathcal{X}}_i(k) = \bigoplus_j \Phi_{i,j}^k \cdot \widehat{\mathcal{X}}_j(0) \qquad \Phi = \left(\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right)$$



[1] S. Bogomolov, M. Forets, G. Frehse, F. Viry, A. Podelski, and C. Schilling. *HSCC.* 2018.

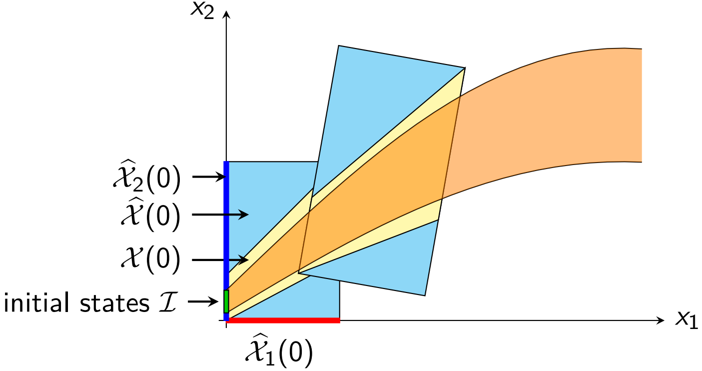
[2] S. Bogomolov, M. Forets, G. Frehse, A. Podelski, and C. Schilling. *Inf. Comput.* (2022).

Decomposition approach[1,2]

$$\widehat{\mathcal{X}}_i(k) = \bigoplus_j \Phi_{i,j}^k \cdot \widehat{\mathcal{X}}_j(0)$$

$$\widehat{\mathcal{X}}_1(1) = A \cdot \widehat{\mathcal{X}}_1(0)$$

$$\Phi = \left(\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right)$$



[1] S. Bogomolov, M. Forets, G. Frehse, F. Viry, A. Podelski, and C. Schilling. *HSCC*. 2018.

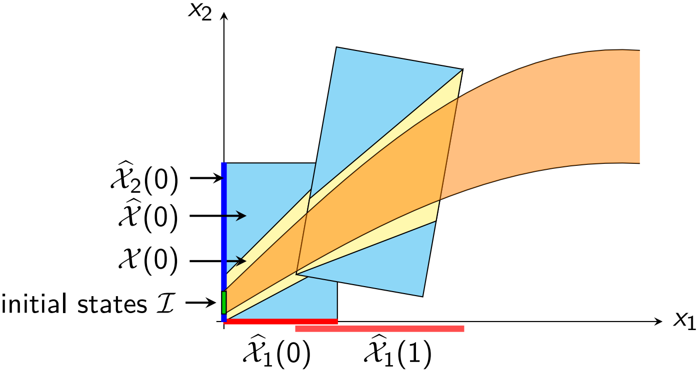
[2] S. Bogomolov, M. Forets, G. Frehse, A. Podelski, and C. Schilling. *Inf. Comput.* (2022).

Decomposition approach[1,2]

$$\widehat{\mathcal{X}}_i(k) = \bigoplus_j \Phi^k_{i,j} \cdot \widehat{\mathcal{X}}_j(0)$$

$$\widehat{\mathcal{X}}_1(1) = A \cdot \widehat{\mathcal{X}}_1(0) \oplus B \cdot \widehat{\mathcal{X}}_2(0)$$

$$\Phi = \left(\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right)$$



[1] S. Bogomolov, M. Forets, G. Frehse, F. Viry, A. Podelski, and C. Schilling. *HSCC.* 2018.

[2] S. Bogomolov, M. Forets, G. Frehse, A. Podelski, and C. Schilling. *Inf. Comput.* (2022).

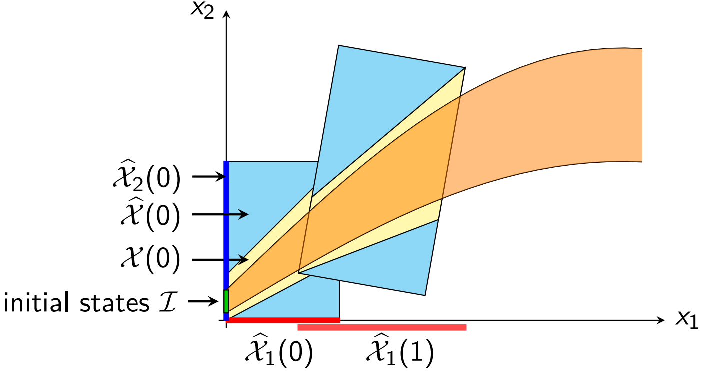
Decomposition approach[1,2]

$$\widehat{\mathcal{X}}_i(k) = \bigoplus_j \Phi^k_{i,j} \cdot \widehat{\mathcal{X}}_j(0)$$

$$\widehat{\mathcal{X}}_1(1) = A \cdot \widehat{\mathcal{X}}_1(0) \oplus B \cdot \widehat{\mathcal{X}}_2(0)$$

$$\widehat{\mathcal{X}}_2(1) = C \cdot \widehat{\mathcal{X}}_1(0)$$

$$\Phi = \left(\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right)$$



[1] S. Bogomolov, M. Forets, G. Frehse, F. Viry, A. Podelski, and C. Schilling. *HSCC.* 2018.

[2] S. Bogomolov, M. Forets, G. Frehse, A. Podelski, and C. Schilling. *Inf. Comput.* (2022).

Decomposition approach[1,2]

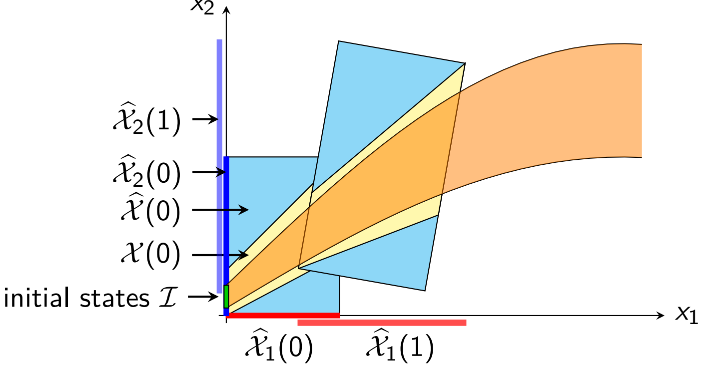
$$\hat{\mathcal{X}}_i(k) = \bigoplus_j \Phi^k_{i,j} \cdot \hat{\mathcal{X}}_j(0)$$

$$\hat{\mathcal{X}}_1(1) = A \cdot \hat{\mathcal{X}}_1(0) \oplus B \cdot \hat{\mathcal{X}}_2(0)$$

$$\hat{\mathcal{X}}_2(1) = C \cdot \hat{\mathcal{X}}_1(0) \oplus D \cdot \hat{\mathcal{X}}_2(0)$$

$$\Phi = \left(\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right)$$

$$\hat{\mathcal{X}}(1) = \hat{\mathcal{X}}_1(1) \times \hat{\mathcal{X}}_2(1)$$



[1] S. Bogomolov, M. Forets, G. Frehse, F. Viry, A. Podelski, and C. Schilling. *HSCC.* 2018.

[2] S. Bogomolov, M. Forets, G. Frehse, A. Podelski, and C. Schilling. *Inf. Comput.* (2022).

Decomposition approach[1,2]

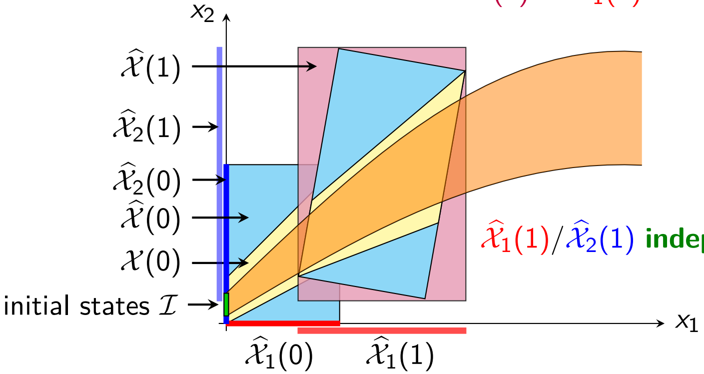
$$\widehat{\mathcal{X}}_i(k) = \bigoplus_j \Phi^k_{i,j} \cdot \widehat{\mathcal{X}}_j(0)$$

$$\widehat{\mathcal{X}}_1(1) = A \cdot \widehat{\mathcal{X}}_1(0) \oplus B \cdot \widehat{\mathcal{X}}_2(0)$$

$$\widehat{\mathcal{X}}_2(1) = C \cdot \widehat{\mathcal{X}}_1(0) \oplus D \cdot \widehat{\mathcal{X}}_2(0)$$

$$\Phi = \left(\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right)$$

$$\widehat{\mathcal{X}}(1) = \widehat{\mathcal{X}}_1(1) \times \widehat{\mathcal{X}}_2(1)$$



$\widehat{\mathcal{X}}_1(1)/\widehat{\mathcal{X}}_2(1)$ **independent**

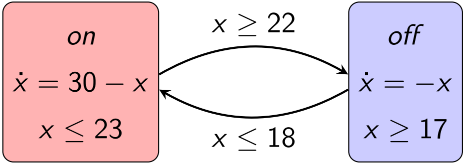
[1] S. Bogomolov, M. Forets, G. Frehse, F. Viry, A. Podelski, and C. Schilling. *HSCC.* 2018.

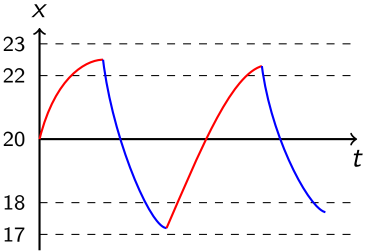
[2] S. Bogomolov, M. Forets, G. Frehse, A. Podelski, and C. Schilling. *Inf. Comput.* (2022).

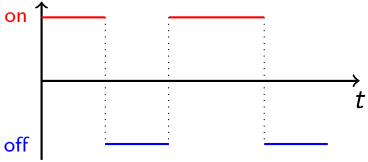
Hybrid and controlled systems

- Combine **continuous** and **discrete** behavior

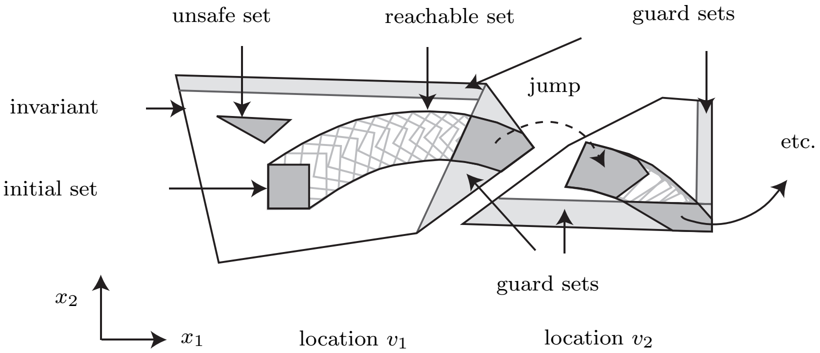
Example: **Thermostat controller**







Reachability analysis for hybrid systems

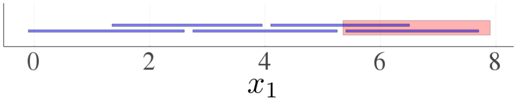


picture taken from[1]

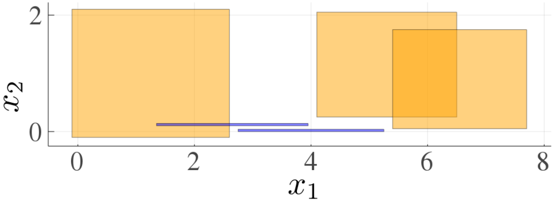
[1] M. Althoff. *An Introduction to CORA 2015. ARCH.* 2015.

Decomposition approach[1]

Low-dimensional check



High-dimensional sets



- **Check condition** for discrete transition in **low dimensions**
- Only compute **high-dimensional set** when necessary
- Allows to analyze a **1027-dimensional model** in 509 sec

[1]S. Bogomolov, M. Forets, G. Frehse, K. Potomkin, and C. Schilling. *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.* (2020).

Periodic controllers with clock jitter

Electro-mechanical brake

$$\begin{aligned}
\dot{I} &= \frac{1}{L}\cdot(K_P\cdot x_e + K_I\cdot x_c) - \frac{I}{L}\left(R + \frac{K^2}{d_{rot}}\right)\\
\dot{x} &= \frac{K}{i\cdot d_{rot}}\cdot I\\
\dot{x}_e &= 0\\
\dot{x}_c &= 0\\
\dot{t} &= 1
\end{aligned}$$

$$t \leq \tau + \iota$$

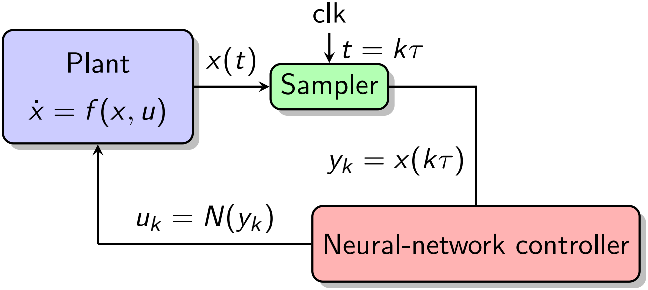
$$t \geq \tau - \iota$$

$$\begin{aligned}
x_e' &:= x_0 - x\\
x_c' &:= x_c + \tau\cdot(x_0 - x)\\
t' &:= t - \tau
\end{aligned}$$

- Analysis for **1,001 transitions: 9 sec**[1] (previous work: **13 h**)

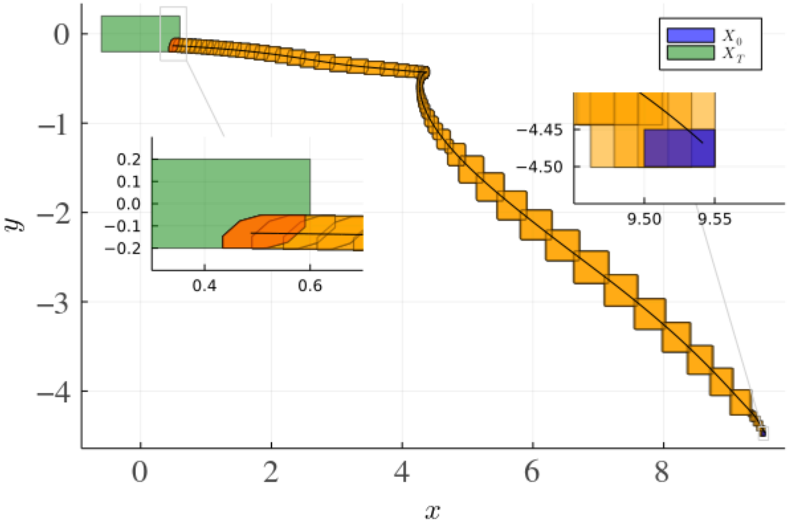
[1] M. Forets, D. Freire, and C. Schilling. *MEMOCODE*. 2020.

Neural-network controllers[1]



[1] C. Schilling, M. Forets, and S. Guadalupe. *AAAI*. 2022.

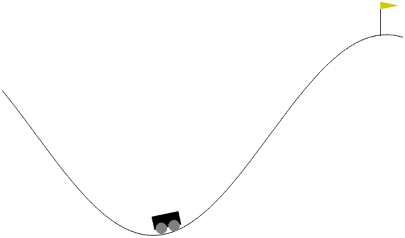
Neural-network controllers[1]

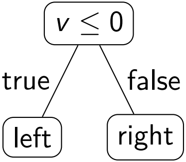


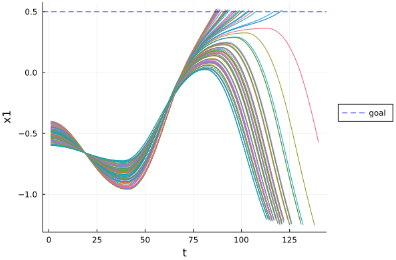
[1] C. Schilling, M. Forets, and S. Guadalupe. *AAAI*. 2022.

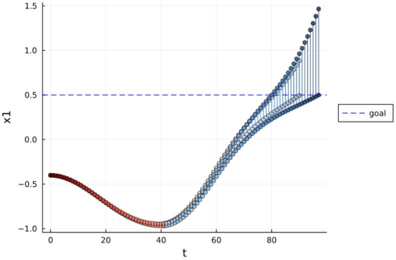
Work in progress: Decision-tree controllers

Mountain car









Summary

- **Reachability analysis** allows to reason about **sets of behaviors**
- Mature for **linear systems** (thousands of dimensions within seconds)
- Still hard for **nonlinear and hybrid systems**
- New challenges in **systems with learned controllers**
- **Exploiting structure** is key

- `https://github.com/JuliaReach`