# Reachability Analysis of Nonlinear and Hybrid Systems using Zonotopes

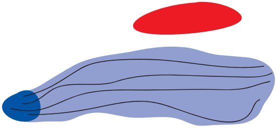
Matthias Althoff

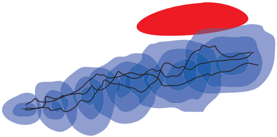
Carnegie Mellon Univ.

May 7, 2010

# Overview of the Talk

- **Main Talk: Reachability Analysis**
  - Linear Systems with Uncertain Parameters
  - Nonlinear Systems
  - Hybrid Systems



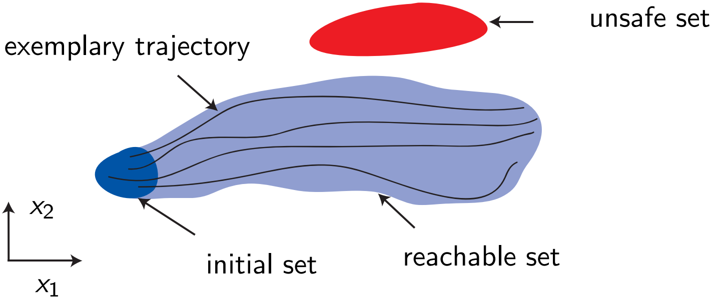


- **Stochastic Reachability Analysis of Linear Systems**
  - Basic Idea
  - Examples
- **Safety Assessment of Autonomous Cars**
  - Basic Idea
  - Examples

*Transregional Collaborative Research Center 28*

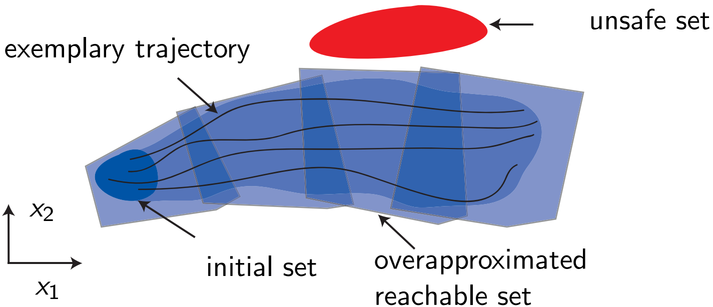
Cognitive Automobiles

# Safety Verification Using Reachable Sets



- System is safe, if no trajectory enters the unsafe set.

# Safety Verification Using Reachable Sets



- System is safe, if no trajectory enters the unsafe set.
- Overapproximated system is safe → real system is safe.

# Linear Systems with Uncertain Parameters

Reachability analysis is performed for the following system class:

## System Model

$$\dot{x} = A\,x + u(t),$$

$$x(0) \in X_0 \subset \mathbb{R}^n, \quad u(t) \in U \subset \mathbb{R}^n, \quad A \in \mathcal{A} \subset \mathbb{R}^{n\times n}$$

where $\mathcal{A}$ is a matrix of intervals and $U$ is a zonotope (specified later).

Example:

$$A \in \mathcal{A} = \begin{bmatrix} [-1.05, -0.95] & [-4.05, -3.95] \\ [3.95, 4.05] & [-1.05, -0.95] \end{bmatrix}$$

$$u(t) \in U = \begin{bmatrix} 1 \\ 1 \end{bmatrix} [-0.1, 0.1]$$

# Initial State Solution (Homogeneous Solution)

**Exact Solution (no uncertainties)**

$$x(r) = e^{A\,r} x(0).$$

**Exact Solution (uncertain system matrix)**

$$x(r) \in \left\{ e^{A\,r} x(0) \middle| A \in \mathcal{A} \right\}$$

- The set of exponential matrices is written in short as $e^{\mathcal{A}r}$.
- How to compute a tight over-approximation?

# Preliminaries: Interval Arithmetic

The interval matrix exponential $e^{\mathcal{A}r}$ is computed based on the addition and multiplication rule:

Given are the intervals $a = [\underline{a}, \overline{a}]$ and $b = [\underline{b}, \overline{b}]$:

$$\begin{aligned} a + b =& [\underline{a} + \underline{b}, \overline{a} + \overline{b}] \\ ab =& [\min(\underline{a}\underline{b}, \underline{a}\overline{b}, \overline{a}\underline{b}, \overline{a}\overline{b}), \max(\underline{a}\underline{b}, \underline{a}\overline{b}, \overline{a}\underline{b}, \overline{a}\overline{b})] \end{aligned}$$

Interval arithmetic is only exact for *single-use-expressions* (SUE). Example ($a = [-2, -1], b = [-1, 1]$):

$$\begin{aligned} c &= ab + a = [-4, 1], \quad \text{not SUE} \rightarrow \text{overapproximated} \\ c &= a(b + 1) = [-4, 0], \quad \text{SUE} \rightarrow \text{exact} \end{aligned}$$

# Interval Matrix Exponential

Taylor series of $e^{\mathcal{A}t}$

$$\begin{aligned} e^{\mathcal{A}t} &= I + \underbrace{\mathcal{A}t + \tfrac{1}{2!}(\mathcal{A}t)^2} + \underbrace{\tfrac{1}{3!}(\mathcal{A}t)^3 + \dots} \\ e^{\mathcal{A}t} &\subset I + W(t) + \textstyle\sum_{i=3}^{m} \tfrac{1}{i!}(\mathcal{A}t)^i + E(t) \end{aligned}$$

- $W$ is computed exactly: Interval arithmetic (SUE) & Analytical minimum and maximum for non-SUE elements.
- $\sum_{i=3}^{m} \frac{1}{i!}(\mathcal{A}t)^i$ is overapproximated with interval arithmetic (not SUE).
- $E(t)$ is a standard approximation for the matrix exponential remainder extended to interval matrices.

# Overview of Reachable Set Computation

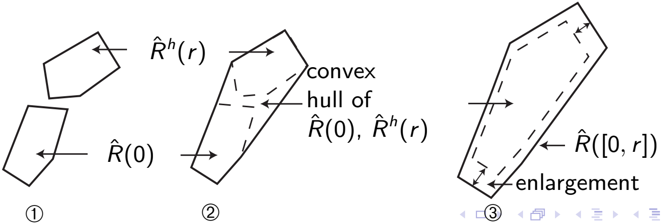
① Compute reachable set $\hat{R}^h(r)$ at time $r$ (without input).

② Obtain convex hull of $\hat{R}(0)$ and $\hat{R}^h(r)$.

③ Enlarge reachable set to guarantee enclosure of all trajectories.

$$\hat{R}([0,r]) = \underbrace{\mathcal{CH}(\hat{R}(0), \overbrace{e^{\mathcal{A}r}\hat{R}(0)}^{\textcircled{1}})}_{\textcircled{2}} + \underbrace{F\hat{R}(0) + \hat{R}^i([0,r])}_{\textcircled{3}}$$

$F$ : Error interval due to the curvature of trajectories within $t \in [0,r]$.
$\hat{R}^i([0,r])$ : Reachable set of the input (inhomogeneous solution).

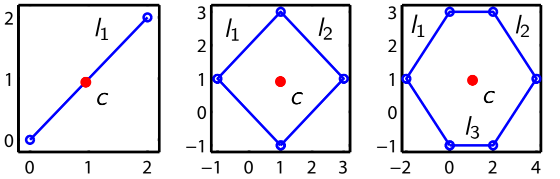


# Representation of Reachable Sets

## Definition of a zonotope $Z$

$$Z = \left\{ x \in \mathbb{R}^n \middle| x = c + \sum_{i=1}^{p} \beta_i g^{(i)}, \quad -1 \leq \beta_i \leq 1 \right\}, \quad c, g^{(i)} \in \mathbb{R}^n$$

- Interpretation: Minkowski sum of line segments $l_i = [-1, 1] g^{(i)}$.
- Zonotopes are centrally symmetric to $c$.
- Short notation: $Z = (c, g^{(1 \ldots p)})$.



(a) $c + l_1$ (b) $c + l_1 + l_2$ (c) $c + l_1 + l_2 + l_3$

# Operations on Zonotopes

Given are $Z_1 = (c_1, g^{(1\dots p)})$ and $Z_2 = (c_2, d^{(1\dots p)})$.

### Addition

$$Z_1 + Z_2 := \{x + y | x \in Z_1, y \in Z_2\} = (c_1 + c_2, g^{(1\dots p)}, d^{(1\dots u)})$$

### Matrix Multiplication

$$LZ_1 := \{Lx | x \in Z_1\} = (Lc, Lg^{(1\dots p)}), \quad L \in \mathbb{R}^{n\times n}$$

### Interval Matrix Multiplication

After defining $\hat{\mathcal{A}} = [-S, S]$ and $\tilde{A}, S \in \mathbb{R}^{n\times n}$, it follows that

$$\begin{aligned} \mathcal{A}Z_1 &= (\tilde{A} + \hat{\mathcal{A}})Z_1 \\ &\subseteq \tilde{A}Z_1 + \hat{\mathcal{A}}Z_1 \\ &\subseteq \tilde{A}Z_1 + \hat{\mathcal{A}}\mathrm{box}(Z_1), \quad \mathrm{box}() : \text{ generates over-appr. axis-aligned box.} \end{aligned}$$

# Operations on Zonotopes

Given are $Z_1 = (c_1, g^{(1\dots p)})$ and $Z_2 = (c_2, d^{(1\dots p)})$.

### Addition

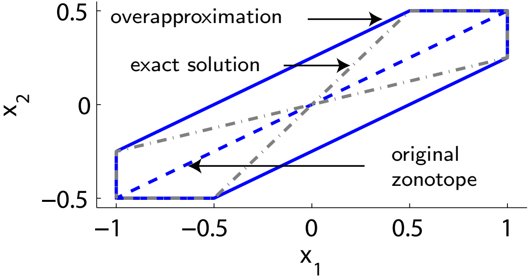
$$Z_1 + Z_2 := \{x + y | x \in Z_1, y \in Z_2\} = (c_1 + c_2, g^{(1\dots p)}, d^{(1\dots u)})$$

### Matrix Multiplication

$$LZ_1 := \{Lx | x \in Z_1\} = (Lc, Lg^{(1\dots p)}), \quad L \in \mathbb{R}^{n\times n}$$

Example:

Zonotope with a single generator:



# Input Solution (Inhomogeneous Solution)

### Exact Solution

The set of all input solution is:

$$x_p(r) \in \left\{ e^{A\,r} \int_0^r e^{-A\,t} u(t)\,dt \,\middle|\, A \in \mathcal{A}, u(t) \in \mathcal{U} \right\}$$

### Over-approximative Solution

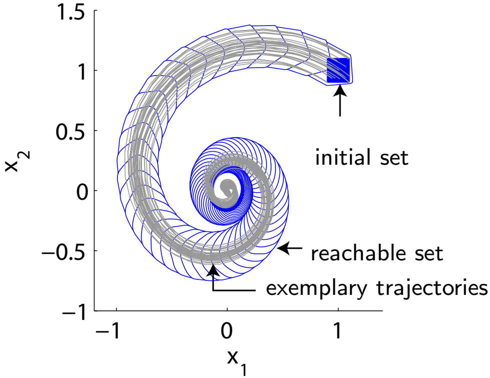
The integral can be over-approximated as follows:

$$\begin{aligned} \hat{R}^i([0,r]) &= \int_0^r e^{\mathcal{A}\tau}\,U\,d\tau \\ &\subseteq \sum_{i=0}^{m} \left( \frac{\mathcal{A}^i\,r^{i+1}}{(i+1)!}\,U \right) + E(r)\,r\,U. \end{aligned}$$

(proof omitted)

# Numerical Example (1)

$$\dot{x} = \underbrace{\begin{bmatrix} [-1.05, -0.95] & [-4.05, -3.95] \\ [3.95, 4.05] & [-1.05, -0.95] \end{bmatrix}}_{\mathcal{A}} x + \underbrace{\begin{bmatrix} 1 \\ 1 \end{bmatrix} [-0.1, 0.1]}_{U}$$



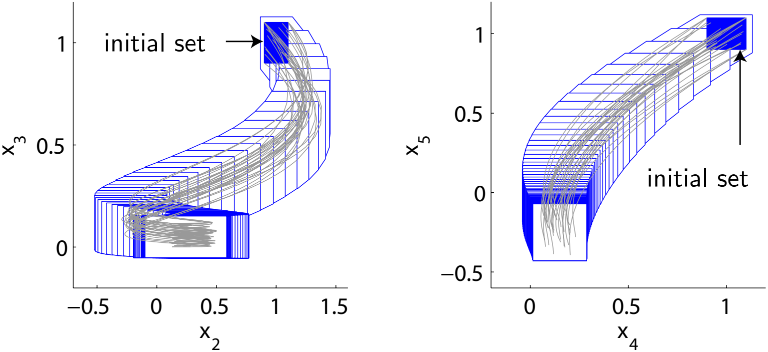
initial set

reachable set

exemplary trajectories

# Numerical Example (2)

Five dimensional example:



(a) Projection on $x_2$ and $x_3$

(b) Projection on $x_4$ and $x_5$

Computation with systems of higher dimensions for 125 time intervals:

| Dimension $n$ | 5 | 10 | 20 | 50 | 100 |
|---|---|---|---|---|---|
| CPU-time [sec] | 0.14 | 0.20 | 0.35 | 1.72 | 7.96 |

# Further Work

Compute with Matrix zonotopes instead of interval matrices:

Matrix Zonotope

$A(p) = \hat{A}^{(0)} + \sum_{i=1}^{\kappa} p^{(i)} \hat{A}^{(i)}, \quad \hat{p}^{(i)} \in [-1, 1].$

Example:

$$\dot{x} = \left( k \cdot \begin{bmatrix} -1.1 & -4.1 \\ 3.9 & -1.1 \end{bmatrix} + (1-k) \begin{bmatrix} -0.9 & -3.9 \\ 4.1 & -0.9 \end{bmatrix} \right) x + u(t),$$
$$k \in [0, 1].$$

Corresponding Interval Matrix:

$$\dot{x} = \begin{bmatrix} [-1.1, -0.9] & [-4.1, -3.9] \\ [3.9, 4.1] & [-1.1, -0.9] \end{bmatrix} x + u(t).$$

# Nonlinear Systems with Uncertain Parameters

Reachability analysis is performed for the following system class:

System Model

$$\dot{x} = f(x(t), u(t), p),$$
$$x(0) \in X_0 \subset \mathbb{R}^n, \quad u(t) \in U \subset \mathbb{R}^m, \quad p \in P \subset \mathcal{I}^o$$

and $u(t)$ is Lipschitz continuous.

Representations of the initial set $X_0$, the parameter set $P$ and the input set $U$:

- **Initial state set $X_0$, input set $U$:** represented by a zonotope.
- **Parameter set $P$:** represented by an o-dimensional interval ($\mathcal{I}$ is the set of real valued intervals).

# Basic Ideas, Properties

### ① Efficient Computation

Embed efficient computation of reachable sets for linear systems using zonotopes.

→ Linearize the system dynamics.

### ② Over-approximation

Compute linearization error bounds and add them to the set of uncertain inputs $U$.

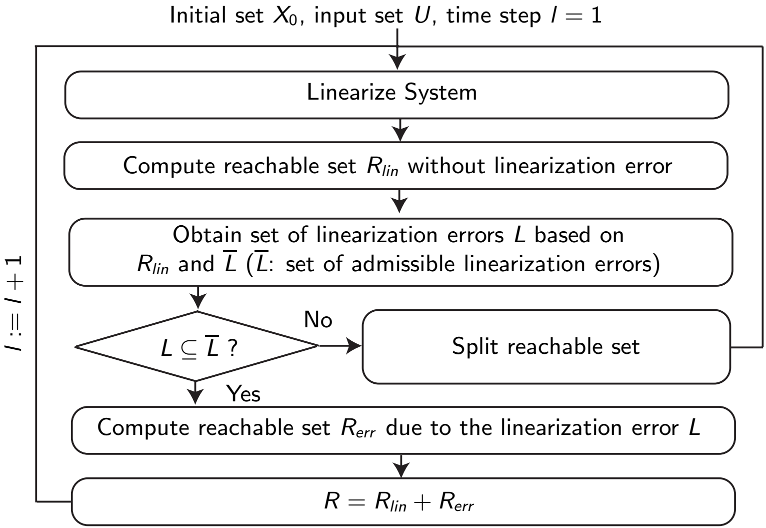
→ Reachable set of the nonlinear system is over-approximative.

### ③ Constrain Linearization Error

Control linearization error bounds by splitting reachable sets.

→ Allows a tradeoff between accuracy and efficiency.

# Overall Algorithm



Initial set $X_0$, input set $U$, time step $l = 1$

Linearize System

Compute reachable set $R_{lin}$ without linearization error

Obtain set of linearization errors $L$ based on $R_{lin}$ and $\overline{L}$ ($\overline{L}$: set of admissible linearization errors)

$L \subseteq \overline{L}$ ?

No

Split reachable set

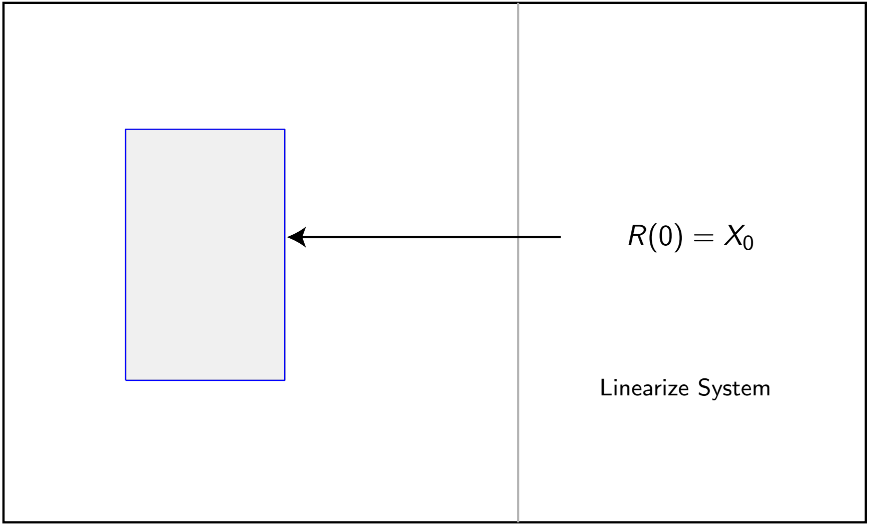
Yes

Compute reachable set $R_{err}$ due to the linearization error $L$

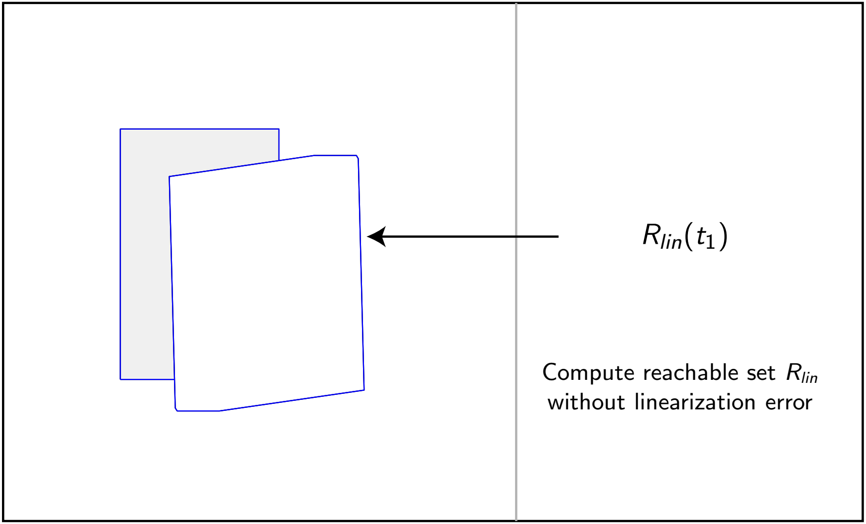
$R = R_{lin} + R_{err}$

$l := l + 1$

# Overall Algorithm: Animation

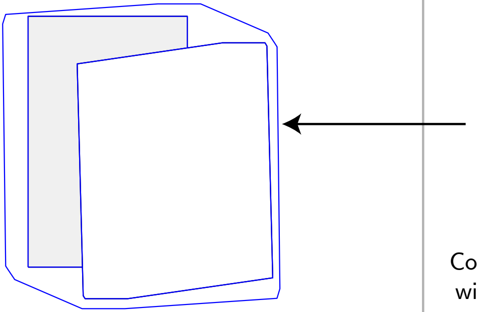


# Overall Algorithm: Animation



Compute reachable set $R_{lin}$ without linearization error

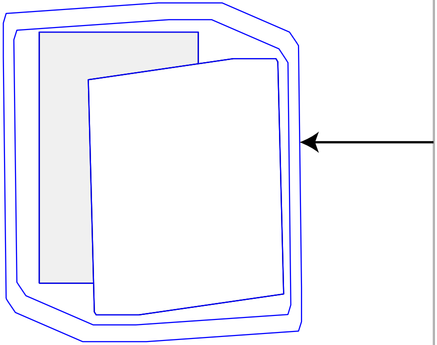
# Overall Algorithm: Animation



$R_{lin}([t_0, t_1])$

Compute reachable set $R_{lin}$ without linearization error

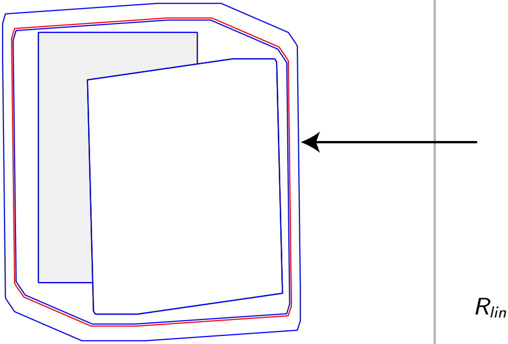
# Overall Algorithm: Animation



$R_{lin}([t_0, t_1]) + \overline{R}_{err}$,
$\overline{R}_{err}$: reachable set due to $\overline{L}$

Obtain set of linearization errors $L$ based on $R_{lin} + \overline{R}_{err}$

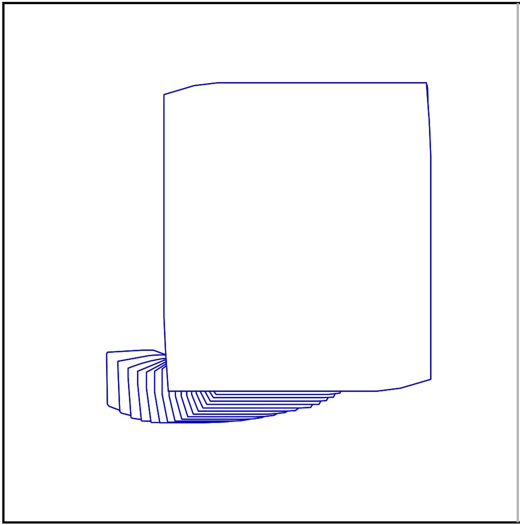
# Overall Algorithm: Animation



$R([t_0, t_1])$

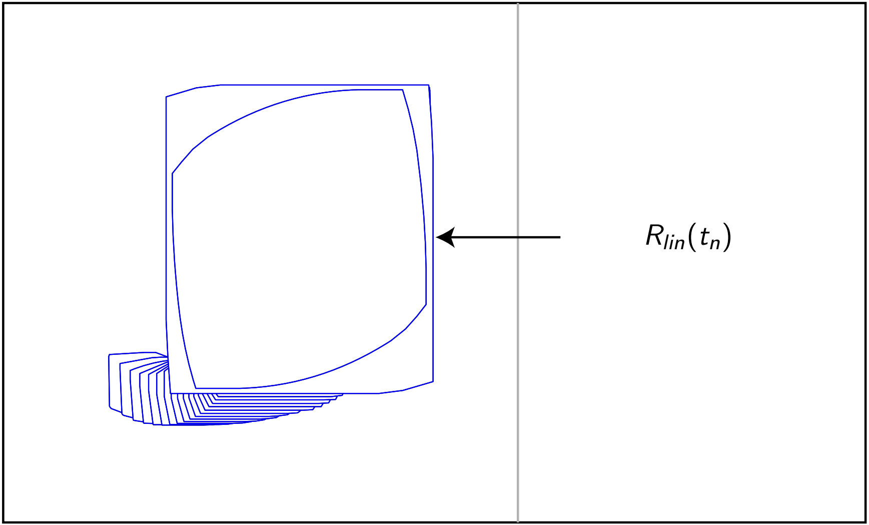
$$R([t_0, t_1]) = \\ R_{lin}([t_0, t_1]) + R_{err}([t_0, t_1])$$

# Overall Algorithm: Animation

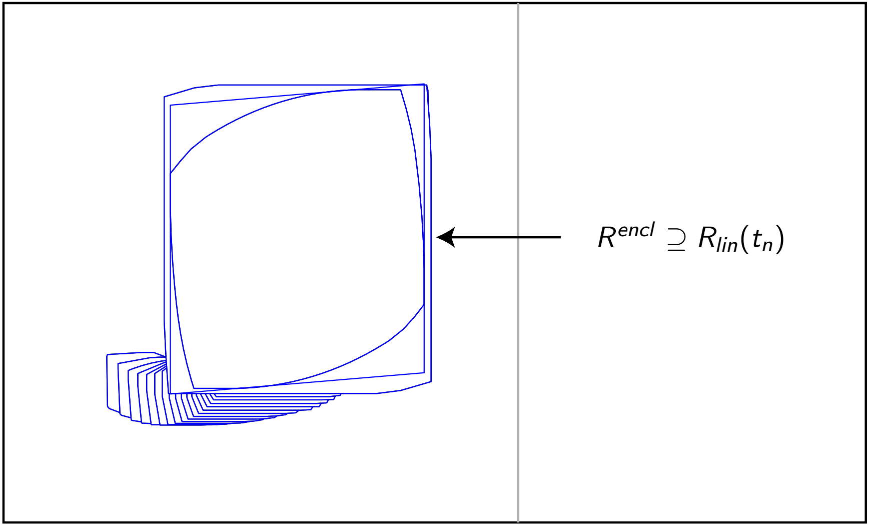


$L \not\subseteq \overline{L}$ !

# Overall Algorithm: Animation

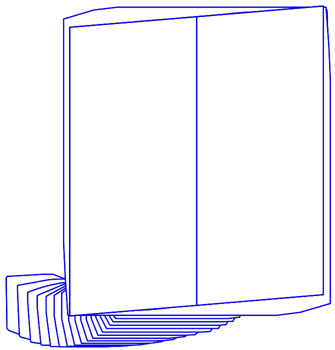


# Overall Algorithm: Animation



$R^{encl} \supseteq R_{lin}(t_n)$

# Overall Algorithm: Animation



Split reachable set

# Linearization and Lagrange Remainder

Original system: $\dot{x} = f(z(t), p(t))$, with $z^T := [x^T, u^T]$:

Taylor series

$$\dot{x}_i \in \underbrace{f_i(z^*, p) + \frac{\partial f_i(z,p)}{\partial z}\Big|_{z=z^*}(z - z^*)}_{1^{st}\text{ order Taylor series} \hat{=} A(p)x + B(p)u + f_i(z^*,p)} +$$

$$\underbrace{\frac{1}{2}(z - z^*)^T \frac{\partial^2 f_i(\xi, p)}{\partial z^2})\Big|_{z=z^*}(z - z^*)}_{\text{Lagrange remainder} L_i}, \quad \xi = z^* + [0,1](z - z^*)$$

- In case of parameter uncertainties: $A(p) \in \mathcal{A}$, $B(p) \in \mathcal{B}$ are bounded by interval matrices $\mathcal{A}$, $\mathcal{B}$.
- Linearization error is obtained from the Lagrange remainder using interval arithmetic → enclose reachable set by multidimensional interval.

# Control of the Linearization Error

Linearization error not enclosed by set of admissible linearization errors ($L \not\subseteq \overline{L}$) → Split reachable set (reduces search space for $L$).

1. Problem: How to split a zonotope, such that the resulting sets are zonotopes?

## Split of a zonotope

A zonotope $Z = (c, g^{(1\dots p)})$ is split into $Z_1$ and $Z_2$ such that $Z_1 \cup Z_2 = Z$, $Z_1 \cap Z_2 = Z^*$ by splitting a single generator:

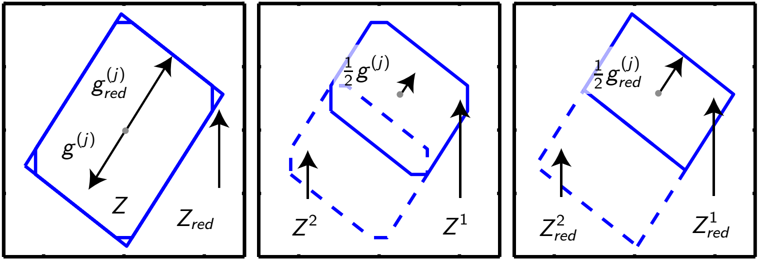
$$\begin{array}{ll} Z_1 & = (c - \frac{1}{2}g^{(j)}, \quad g^{(1\dots j-1)}, \quad \frac{1}{2}g^{(j)}, \quad g^{(j+1\dots p)}) \\ Z_2 & = (c + \frac{1}{2}g^{(j)}, \quad g^{(1\dots j-1)}, \quad \frac{1}{2}g^{(j)}, \quad g^{(j+1\dots p)}) \\ Z^* & = (c, \quad g^{(1\dots j-1)}, \quad g^{(j+1\dots p)}) \end{array}$$

# Splitting of Zonotopes: Overlapping vs. Overapproximation

2. Problem: Proposed split is not effective if zonotope is composed by many generators.

→ Zonotope is over-approximated by less generators.

→ Tradeoff between over-approximation and effectiveness.



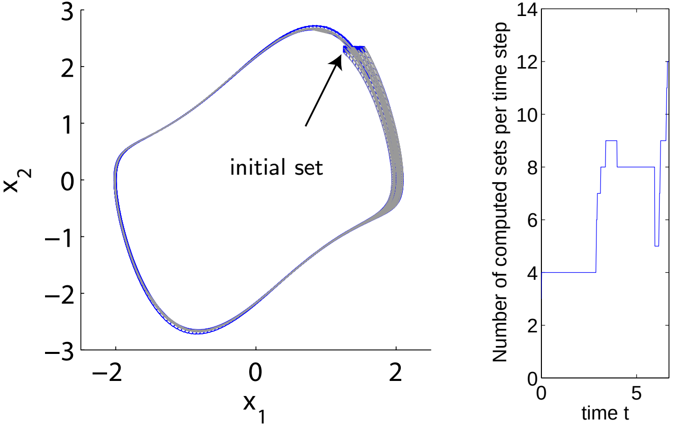
(a) $Z$, $Z_{red}$ (b) $Z^1$, $Z^2$ (c) $Z_{red}^1$, $Z_{red}^2$

3. Problem: Which generator should be split?

→ Brute force approach is applied.

## Van-der-Pol Oscillator

$$\dot{x}_1 = x_2, \quad \dot{x}_2 = (1 - x_1^2)x_2 - x_1$$



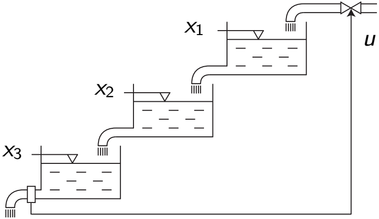
Computational time: 19 sec (Matlab, AMD Athlon64 3700+ processor).

# Water Tank System: Set Up

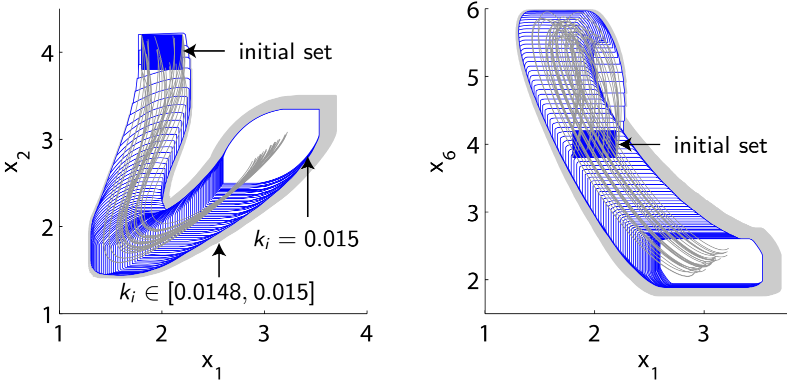
- The states $x_i$ are the water levels of each tank and $u$ is the water flow into the first tank.
- The differential equation for the i$^{th}$ tank is

$$\dot{x}_i = \frac{1}{A_i}(k_{i-1}\sqrt{2gx_{i-1}} - k_i\sqrt{2gx_i}).$$

- The uncertain parameters are $k_i \in [0.0148, 0.015]$ and the inflow disturbance is $v \in [-0.005, 0.005]$.



# Water Tank System: Reachable Sets

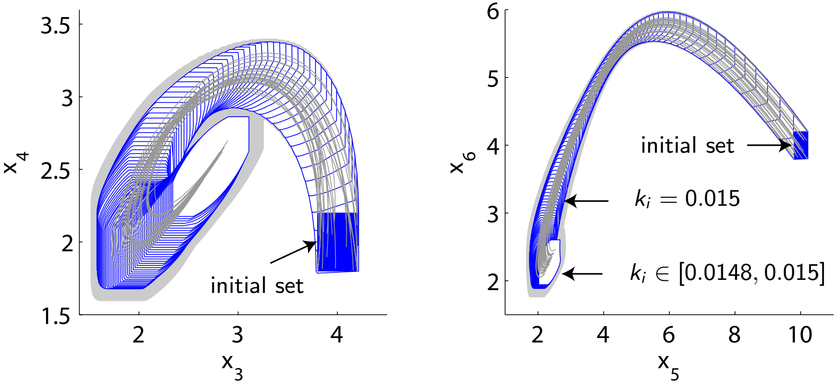


(a) Projection onto $x_1$, $x_2$.

(b) Projection onto $x_1$, $x_6$.

| Dimension $n$ | 5 | 10 | 20 | 50 | 100 |
|---|---|---|---|---|---|
| CPU-time [sec] | 1.19 | 1.73 | 3.11 | 11.59 | 35.78 |
| CPU-time [sec] (uncertain param.) | 6.83 | 12.92 | 28.94 | 119.58 | 523.56 |

# Water Tank System: Reachable Sets



(a) Projection onto $x_3$, $x_4$.

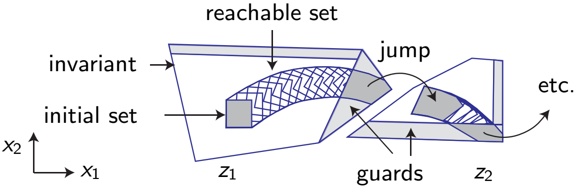
(b) Projection onto $x_5$, $x_6$.

| Dimension $n$ | 5 | 10 | 20 | 50 | 100 |
|---|---|---|---|---|---|
| CPU-time [sec] | 1.19 | 1.73 | 3.11 | 11.59 | 35.78 |
| CPU-time [sec] (uncertain param.) | 6.83 | 12.92 | 28.94 | 119.58 | 523.56 |

# Hybrid Systems

## Hybrid Automaton $HA = (Z, z_0, X, X_0, inv, T, g, j, flow)$

- the set of locations $Z$ with initial location $z_0$,
- the continuous state space $X \subset \mathbb{R}^n$ with initial state set $X_0$,
- the **invariant** *inv* and **guard sets** $g$ of each location $z$ which are modeled as **polytopes**,
- the set of discrete transitions $T \subseteq Z \times Z$,
- the **linear jump function** $j$ such that $x' = C_g x + d_g$ ($x'$: state after jump)
- the **linear flow function** $\dot{x} = A_z\, x + u(t)$

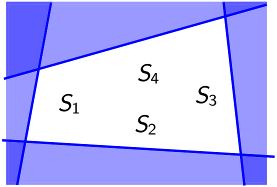


# Combined use of Zonotopes and Polytopes

- **Representation of continuous evolution by zonotopes:**
- **Representation of the intersection with guards, invariants by polytopes:**

$$P = \left\{ x \in \mathbb{R}^n \middle| Cx \leq d \right\}, \quad C \in \mathbb{R}^{q \times n}, d \in \mathbb{R}^q$$

Alternative definition: Intersection of halfspaces $S_i$.



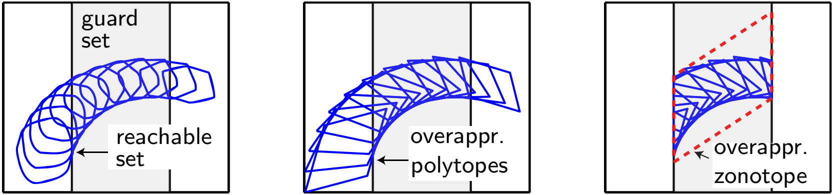
# Reachable Set Computation for Guard Set Intersection

① Compute reachable set using zonotopes.

② Transform zonotopes to overapproximated polytopes.

③ Intersect polytopes with the guard set.

④ Overapproximate intersected polytopes by a single zonotope → Continue computation within the invariant of the next discrete state.



(a) Step ①

(b) Step ②

(c) Step ③ and ④

Alternative for guards modeled by hyperplanes: A. Girard, C. Le Guernic (HSCC'08)

# Representation of Zonotopes by Halfspaces

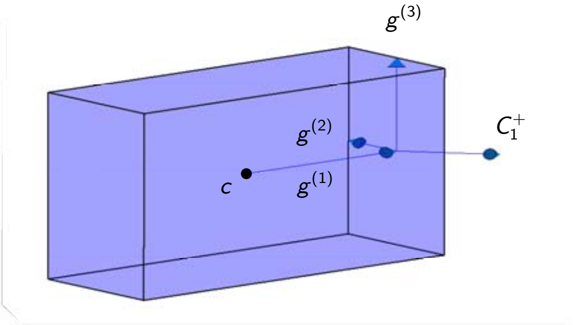
**Overview**:

- Zonotopes are a special case of polytopes:
  Exact Conversion
  - Conversion of parallelotopes
  - Conversion of zonotopes
- The change of representation is computationally expensive:
  Overapproximative Conversion
  - Overapproximate zonotopes by parallelotopes
  - Order reduction of zonotopes
  - Overapproximate zonotopes by several parallelotopes → intersection

# Preliminaries and Notations

1. Parallelotope is a zonotope of order 1: Number of generators $p$ equals the dimension $n$.
2. Facets are spanned by $n-1$ generators. Matrix of generators with the i$^{th}$ generator missing:

$$G^{\langle i \rangle} = [g^{(1)}, \dots, g^{(i-1)}, g^{(i+1)}, \dots, g^{(n)}].$$



# Preliminaries and Notations

① Parallelotope is a zonotope of order 1: Number of generators $p$ equals the dimension $n$.

② Facets are spanned by $n-1$ generators. Matrix of generators with the i$^{th}$ generator missing:

$$G^{\langle i\rangle} = [g^{(1)}, \dots, g^{(i-1)}, g^{(i+1)}, \dots, g^{(n)}].$$

③ Normal vector $C_i^+$ of the i$^{th}$ facet is perpendicular to all generators in $H := G^{\langle i\rangle}$:

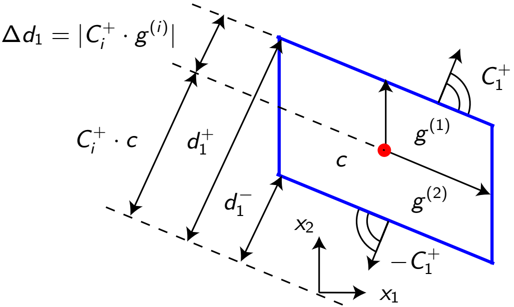
$$C_i^+ = nX(H) := [\dots, (-1)^{k+1}\det(H^{[k]}), \dots]^T,$$

where $H^{[k]}$ is the matrix $H$ whose k$^{th}$ row is removed.

# Conversion from Generator to Halfspace Representation of Parallelotopes

Halfspace representation $Cx \leq d$

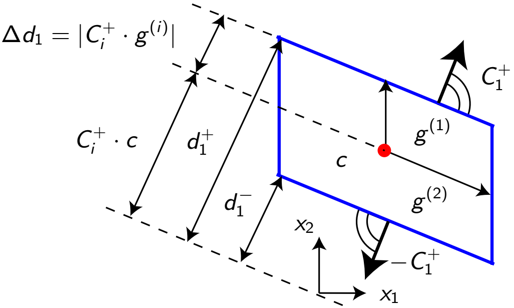
$$\begin{aligned} C &= \begin{bmatrix} C^+ & -C^+ \end{bmatrix}^T & C_i^+ &= nX(G^{\langle i \rangle})^T / \|nX(G^{\langle i \rangle})\|_2 \\ d &= \begin{bmatrix} d^+ & d^- \end{bmatrix}^T & d_i^+ &= C_i^+ \cdot c + \Delta d_i, \quad d_i^- = -C_i^+ \cdot c + \Delta d_i \\ & & \Delta d_i &= |C_i^+ \cdot g^{(i)}| \end{aligned}$$



# Conversion from Generator to Halfspace Representation of Parallelotopes

Halfspace representation $Cx \leq d$

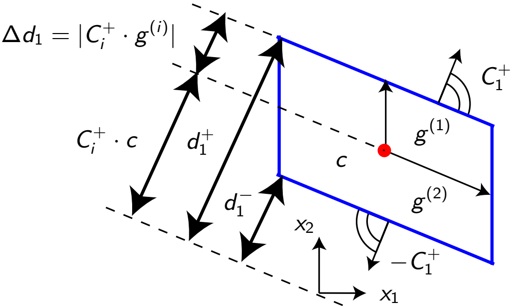
$$\begin{aligned} C &= \begin{bmatrix} C^+ & -C^+ \end{bmatrix}^T & \mathbf{C_i^+} &= \mathbf{nX(G^{\langle i\rangle})^T/\|nX(G^{\langle i\rangle})\|_2} \\ d &= \begin{bmatrix} d^+ & d^- \end{bmatrix}^T & d_i^+ &= C_i^+ \cdot c + \Delta d_i, \quad d_i^- = -C_i^+ \cdot c + \Delta d_i \\ & & \Delta d_i &= |C_i^+ \cdot g^{(i)}| \end{aligned}$$



# Conversion from Generator to Halfspace Representation of Parallelotopes

Halfspace representation $Cx \leq d$

$$
\begin{aligned}
C &= \begin{bmatrix} C^+ & -C^+ \end{bmatrix}^T & C_i^+ &= nX(G^{\langle i \rangle})^T / \| nX(G^{\langle i \rangle}) \|_2 \\
d &= \begin{bmatrix} d^+ & d^- \end{bmatrix}^T & \mathbf{d_i^+} &= \mathbf{C_i^+} \cdot \mathbf{c} + \mathbf{\Delta d_i}, \quad \mathbf{d_i^-} = -\mathbf{C_i^+} \cdot \mathbf{c} + \mathbf{\Delta d_i} \\
& & \mathbf{\Delta d_i} &= |\mathbf{C_i^+} \cdot \mathbf{g^{(i)}}|
\end{aligned}
$$



# Conversion from Generator to Halfspace Representation of Zonotopes

- Extension is straightforward: $n-1$ generators selected from $p$ generators for each non-parallel facet → $2\binom{p}{n-1}$ facets.
- Facet obtained by cancelling $p-n+1$ generators from the $G$-matrix which is denoted by $G^{\langle\gamma,\ldots,\eta\rangle}$.

### Halfspace representation $Cx \leq d$

$$\begin{aligned} C &= \begin{bmatrix} C^+ & -C^+ \end{bmatrix}^T & C_i^+ &= nX(G^{\langle\gamma,\ldots,\eta\rangle})^T / \|nX(G^{\langle\gamma,\ldots,\eta\rangle})\|_2 \\ d &= \begin{bmatrix} d^+ & d^- \end{bmatrix}^T & d_i^+ &= C_i^+ \cdot c + \Delta d_i, \quad d_i^- = -C_i^+ \cdot c + \Delta d_i \\ & & \Delta d_i &= \textstyle\sum_{\upsilon=1}^{p} |C_i^+ \cdot g^{(\upsilon)}| \end{aligned}$$

Complexity with respect to the number $p$ of generators is $\mathcal{O}(\binom{p}{n-1} \cdot p)$ → linear in the number of facets.

# Overapproximation of a Zonotope by a Parallelotope

## Basic Procedure

① Choose $n$ generators that should represent the parallelotope.

② Stretch chosen generators such that the zonotope is enclosed.

The overapproximated parallelotope $\Psi$ is generated as follows:

$$\Psi = \Gamma \cdot \mathrm{box}(\Gamma^{-1} Z)$$

where

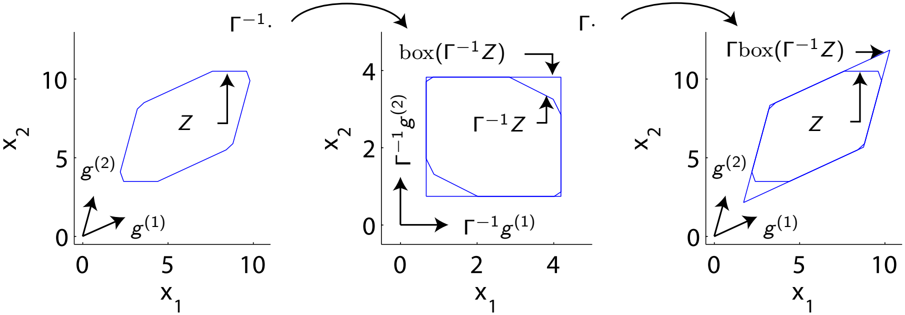
- $\Gamma \in \mathbb{R}^{n \times n}$ is the matrix of $n$ generators $g^{(i)}$ taken out of all $p$ generators,
- $\mathrm{box}(Z)$ returns the axis-oriented bounding box of a zonotope $Z$.

# Overapproximation of a Zonotope by a Parallelotope

## Basic Procedure

① Choose *n* generators that should represent the parallelotope.

② Stretch chosen generators such that the zonotope is enclosed.



Remaining question: How to choose Γ?

# Overapproximation of a Zonotope by a Parallelotope

Choose Γ such that a metric is minimized:

Proposed Metric

$$\Theta = (vol(W \cdot Z^{\mathrm{red}})/vol(W \cdot Z))^{1/n}, \quad W = \mathrm{diag}(w).$$

- $Z$: original zonotope, $Z^{\mathrm{red}}$: reduced zonotope, $W$: normalizes coordinate axes.
- $W = 1$: Determines the ratio of the edge length of two cubes, in which the volume of the reduced zonotope and the original zonotope fit.

Candidates for Γ have to pass two tests:

1. **Length test**: longest generators (2-norm) pass.
2. **Pseudo volume test**: generator combination spanning the largest volume $\tilde{\Theta} = |\det[g^{(i_1)}, \ldots, g^{(i_n)}]|^{-1}$ pass (no stretching considered).

For the remaining generator combinations, the best performance index Θ wins (sufficient to compute $vol(W \cdot Z^{\mathrm{red}})$).

# Overapproximation of a Zonotope by a Parallelotope

Choose $\Gamma$ such that a metric is minimized:

### Proposed Metric

$$\Theta = (vol(W \cdot Z^{\mathrm{red}})/vol(W \cdot Z))^{1/n}, \quad W = \mathrm{diag}(w).$$

- $Z$: original zonotope, $Z^{\mathrm{red}}$: reduced zonotope, $W$: normalizes coordinate axes.
- $W = 1$: Determines the ratio of the edge length of two cubes, in which the volume of the reduced zonotope and the original zonotope fit.

| dimension | order | mean of $t$ [sec]: | mean of $\Theta$: | [min,max] of $\Theta$: | variance of $\Theta$: |
|---|---|---|---|---|---|
| 2 | 2 | 0.0046 | 1.0582 | [1.0030, 1.1349] | 0.0011 |
| 2 | 6 | 0.0056 | 1.0908 | [1.0369, 1.1522] | 0.0005 |
| 4 | 2 | 0.0078 | 1.1560 | [1.0343, 1.2899] | 0.0024 |
| 4 | 6 | 0.0060 | 1.2967 | [1.2143, 1.3995] | 0.0015 |
| 6 | 2 | 0.0221 | 1.2574 | [1.0779, 1.4088] | 0.0039 |

# Overapproximation of a Zonotope by a Reduced Zonotope

### Basic Procedure

1. Split zonotope into a part $\check{Z}$ which is unchanged and a part $\tilde{Z}$ reduced to a parallelotope ($Z = \check{Z} + \tilde{Z}$).
2. Selected generators of unchanged part $\check{Z}$ are the longest generators (2-norm).

The reduced zonotope $Z^{\mathrm{red}}$ is generated as follows:

$$Z^{\mathrm{red}} = \check{Z} + \Psi, \quad \Psi = \Gamma \mathrm{box}(\Gamma^{-1} \tilde{Z}).$$

Result: Improvements are marginal; computation time for halfspace conversion is drastically increased.

# Overapproximation of a Zonotope by Intersected Parallelotopes

## Basic Procedure

① Compute several enclosing parallelotopes with the best performance indices.

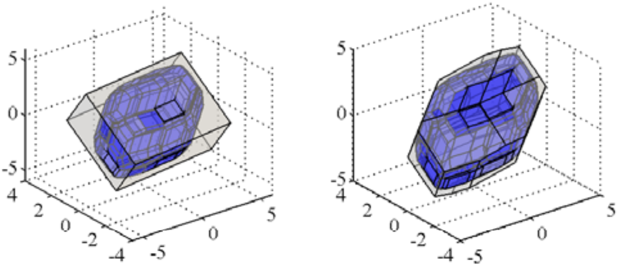
② Intersect obtained parallelotopes.

| dim. | order | inters. | mean of $t$ [sec]: | mean of $\Theta$: | [min,max] of $\Theta$: | variance of $\Theta$: |
|---|---|---|---|---|---|---|
| 4 | 2 | 1 | 0.0078 | 1.1560 | [1.0343, 1.2899] | 0.0024 |
| 4 | 2 | 4 | 0.0382 | 1.0288 | [1.0019, 1.0836] | 0.0003 |
| 4 | 6 | 1 | 0.0060 | 1.2967 | [1.2143, 1.3995] | 0.0015 |
| 4 | 6 | 4 | 0.0421 | 1.1383 | [1.0808, 1.2892] | 0.0010 |
| 6 | 2 | 1 | 0.0221 | 1.2574 | [1.0779, 1.4088] | 0.0039 |
| 6 | 2 | 4 | 0.0739 | 1.0964 | [1.0251, 1.1759] | 0.0010 |

# Overapproximation of a Zonotope by Intersected Parallelotopes

## Basic Procedure

1. Compute several enclosing parallelotopes with the best performance indices.
2. Intersect obtained parallelotopes.



(a) No intersection.

(b) 4 intersections.

# Overapproximation of a Set of Polytopes

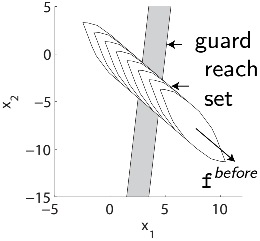
Polytopes have only a few facets → computation of vertices is feasible.

Task: Enclose points in $\mathbb{R}^n$

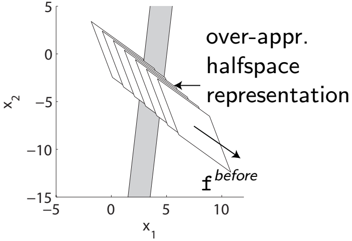
Possible methods are, e.g.:

1. Oriented rectangular hulls based on singular value decomposition (O. Stursberg, B. Krogh: HSCC 2003).
2. Compute axis-oriented box where one of these generators is replaced by the flow direction.
3. Compute in parallel with several enclosures.

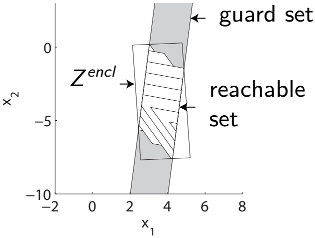
# Over-Approximation of a Set of Polytopes



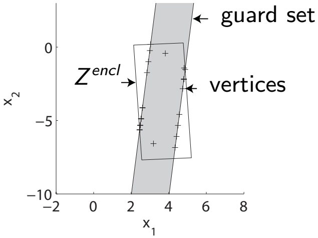
①



②



③



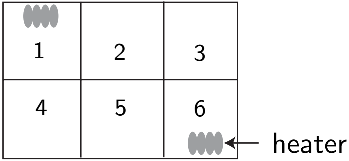
④

# Room Heating Example

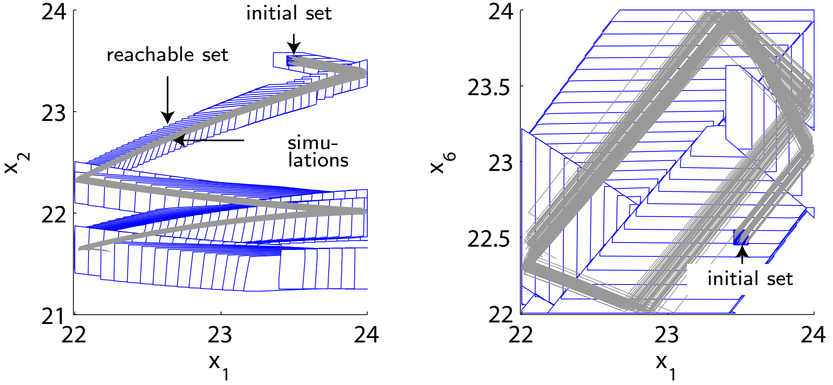
- 6 rooms with heaters in room 1 and 6 are considered.
- The heaters are switched on if the temperature is below 20 degree celsius and switched off when the temperature exceeds 24 degree.
- The temperature dynamics of room $i$ is:

$$\dot{x}_i = c_i h_i + b_i(u - x_i) + \sum_{i \neq j} a_{ij}(x_j - x_i)$$

with room specific constant parameters $a_{ij}$, $b_i$ and $c_i$.



# Reachable Sets

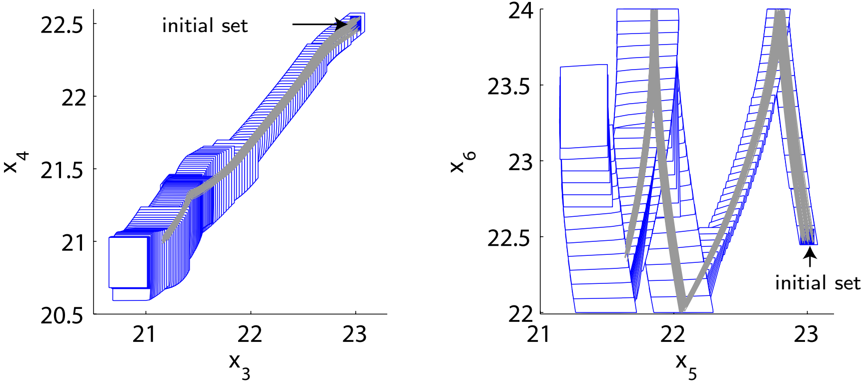


(a) Projection of $x_1$, $x_2$

(b) Projection of $x_1$, $x_6$

Computation time: 16.8 sec on an AMD Athlon64 3700+ processor (single core) in Matlab.

# Reachable Sets



(c) Projection of $x_3$, $x_4$

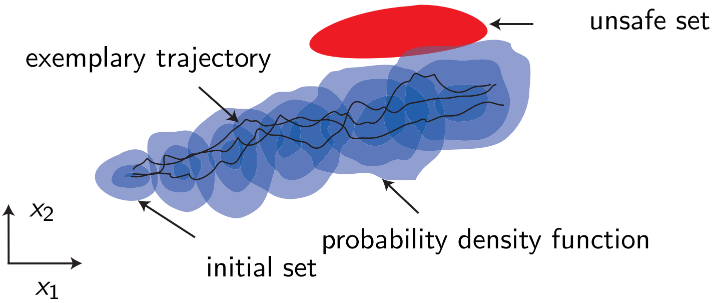
(d) Projection of $x_5$, $x_6$

Computation time: 16.8 sec on an AMD Athlon64 3700+ processor (single core) in Matlab.

# Conclusions

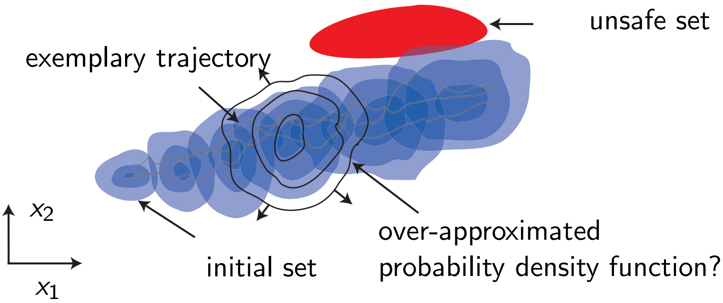
- Linear Systems:
  - Uncertain parameters within specified intervals.
  - No wrapping-free implementation as for LTI systems.
- Nonlinear Systems:
  - Efficient computation for high dimensional nonlinear systems with uncertain parameters.
  - Algorithm is best suited for systems with lower nonlinearity measure.
  - In case of highly nonlinear systems, the current implementation may get stuck due to numerical problems.
- Hybrid Systems:
  - Zonotopes allow efficient computations for the cont. evolution.
  - Efficient conversion from zonotopes to polytopes and vice versa possible; drawback: introduced overapproximation.
  - However: Reachable sets computed by polytopes also generate an overapproximation when intersected by guards.

# Stochastic Safety Verification



- Probability of being in an unsafe set can be computed from the probability density function (pdf).

# Stochastic Safety Verification



exemplary trajectory

unsafe set

initial set

over-approximated probability density function?

- Probability of being in an unsafe set can be computed from the probability density function (pdf).
- Is the exact probability density function computable? Is an over-approximation computable and how is it defined?

# Two Different Definitions

- Probability of **entering** an unsafe set $\mathcal{X}^{\mathrm{unsafe}}$:

$$P(\mathrm{reach}_{t_f}) = P(\exists t \in [0, t_f], x(t) \in \mathcal{X}^{\mathrm{unsafe}})$$
$$P(\mathrm{reach}_{\infty}) = P(\exists t \geq 0, x(t) \in \mathcal{X}^{\mathrm{unsafe}})$$

  Applied methods: Monte Carlo simulation, Markov chain abstraction, reformulation as stochastic optimal control problem, . . .

  Except for Monte Carlo simulation, methods suffer under the curse of dimensionality (usually exponential complexity in number of continuous state variables).

- Probability of **being** in an unsafe set $\mathcal{X}^{\mathrm{unsafe}}$:

$$P(x \in \mathcal{X}^{\mathrm{unsafe}}, t) = \int_{\mathcal{X}^{\mathrm{unsafe}}} f_X(x, t)\, dx.$$

  Equivalent to above definition when unsafe set is absorbing.

# Definition of the Considered System Class

### Considered System Class

$$\dot{X} = A\,X(t) + u(t) + C\xi(t),$$
$$X(0) \backsim f_X(x, t=0),\; u(t) \in U,\; \xi \hat{=} \text{white noise}$$

where $A$ and $C$ are matrices of proper dimension and $A$ has full rank. $X(t)$ is a stochastic process, $f_X(x,t)$ its probability density function.

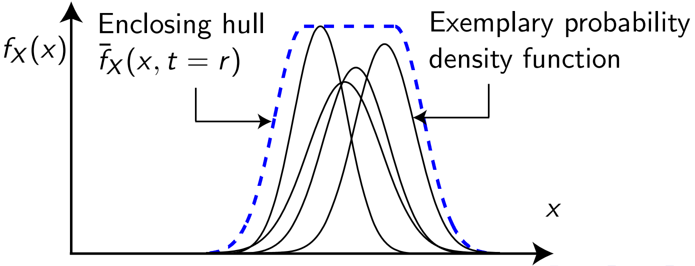
There are two kinds of inputs:

- $u(t)$: can take values from a set $U$; no stochastic information given.
- $C\xi(t)$: white noise input with multivariate Gaussian distribution.

# Enclosing Hull of Probability Distributions

- Input trajectory $u(t)$ is known:
  exact solution has Gaussian distribution: $X(t) \backsim \mathcal{N}(\mu(t), \Sigma(t))$ with mean value $\mu(t)$ and covariance matrix $\Sigma(t)$.
- Input trajectory $u(t)$ is unknown: → Enclosing hulls required:

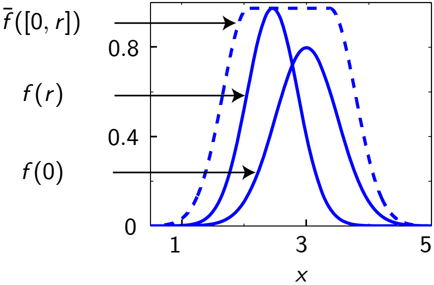
$$\bar{f}_X(x, t = r) = \sup\{f_X(x, t = r) | X(t) \text{ is a stochastic process,} \\ u(t) \in U,\ f_X(x, 0) = f_0\}$$



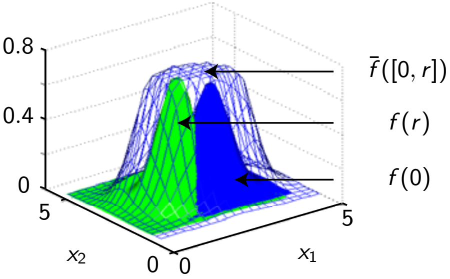
# Enclosing Hull for Time Intervals

Numerical examples:

- $f(0), f(r)$: probability distribution at time $t = 0$ and $t = r$,
- $\bar{f}([0,r])$: enclosing probabilistic hull for $t \in [0,r]$.



(e) One dimensional example.

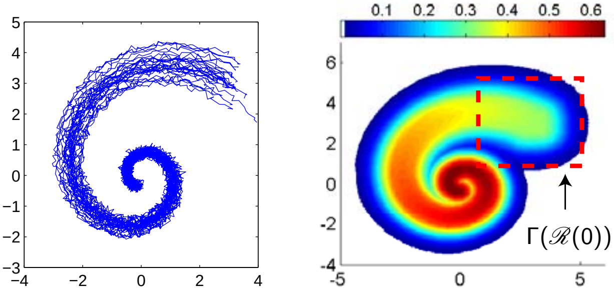


(f) Two dimensional example.

Uncertain mean is modeled by a zonotope → computational methods from previous slides can be applied.

# Two-dimensional example

$$\dot{x} = \begin{bmatrix} -1 & -4 \\ 4 & -1 \end{bmatrix} x + \begin{bmatrix} [-0.01, 0.01] \\ [-0.01, 0.01] \end{bmatrix} + \begin{bmatrix} 0.7 & 0 \\ 0 & 0.7 \end{bmatrix} \xi.$$



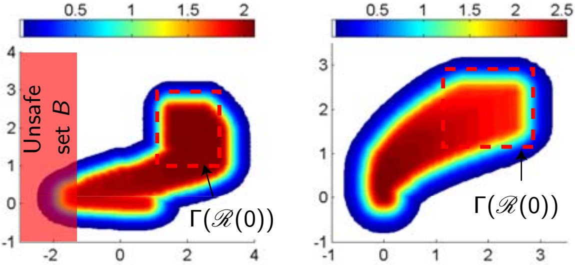
(a) Simulation examples.

(b) Enclosing probabilistic hulls.

# Five-dimensional example

$$\dot{x} = Ax + u + 0.5 \cdot I \cdot \xi,$$

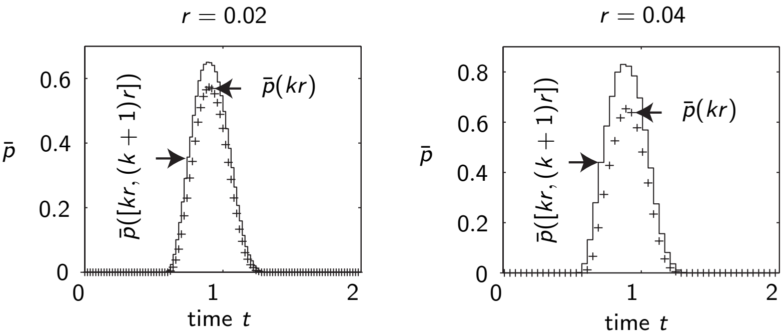
$$A = \begin{bmatrix} -1 & -4 & 0 & 0 & 0 \\ 4 & -1 & 0 & 0 & 0 \\ 0 & 0 & -3 & 1 & 0 \\ 0 & 0 & -1 & -3 & 0 \\ 0 & 0 & 0 & 0 & -2 \end{bmatrix}, \quad u \in U = \begin{bmatrix} [-0.1, 0.1] \\ \vdots \\ [-0.1, 0.1] \end{bmatrix}^T .$$



(a) Projection on $x_2$, $x_3$.

(b) Projection on $x_4$, $x_5$.

# Results and Computational Times



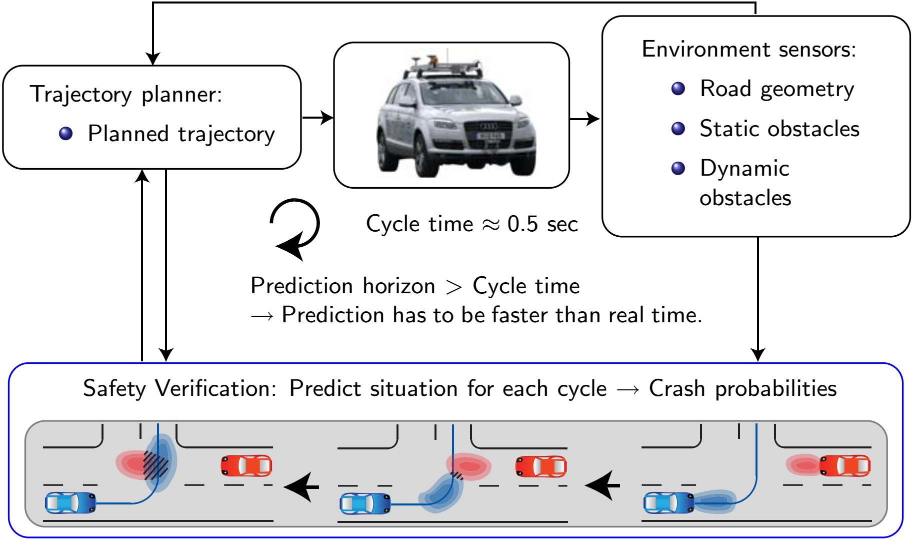
Computational times: Higher order systems with randomly generated matrices $A$, $C$ computed (Matlab + single core desktop computer (AMD Athlon64 3700)).

| Dimension $n$ | 5 | 10 | 20 | 50 | 100 |
|---|---|---|---|---|---|
| CPU-time [sec] | 0.72 | 1.29 | 2.61 | 8.97 | 29.1 |

# Conclusions

- Efficient computation of enclosing probabilistic hulls for high dimensional linear systems.
- Algorithm allows combining Gaussian white noise with disturbances of unknown stochastic properties.
- Over-approximative approach allows one to consider non-Gaussian noise.
- Possible integration in algorithms for the reachability analysis of nonlinear systems.

# Safety Assessment of Autonomous Cars



Trajectory planner:

- Planned trajectory

Environment sensors:

- Road geometry
- Static obstacles
- Dynamic obstacles

Cycle time ≈ 0.5 sec

Prediction horizon > Cycle time
→ Prediction has to be faster than real time.

Safety Verification: Predict situation for each cycle → Crash probabilities

# Modeling of Other Traffic Participants

Structured Environments:

- Vehicles follow preferred paths such as traffic lanes.
- Used for: Normal behavior of traffic participants on a road network.

*Longitudinal dynamics*:

$$\dot{s} = v, \quad \dot{v} = f(v, u), \quad s : \text{position}, \quad v : \text{velocity}, \quad u : \text{input}.$$

$\longrightarrow$ Longitudinal probability distribution based on this model.

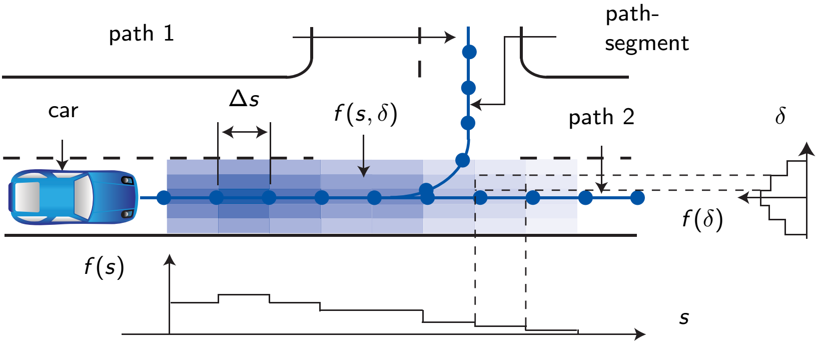
*Lateral dynamics*:

Difficult to model (e.g. driver model for lane keeping)
$\longrightarrow$ Static probability distribution.

# Modeling of Other Traffic Participants

## Structured Environments:

- Vehicles follow preferred paths such as traffic lanes.
- Used for: Normal behavior of traffic participants on a road network.

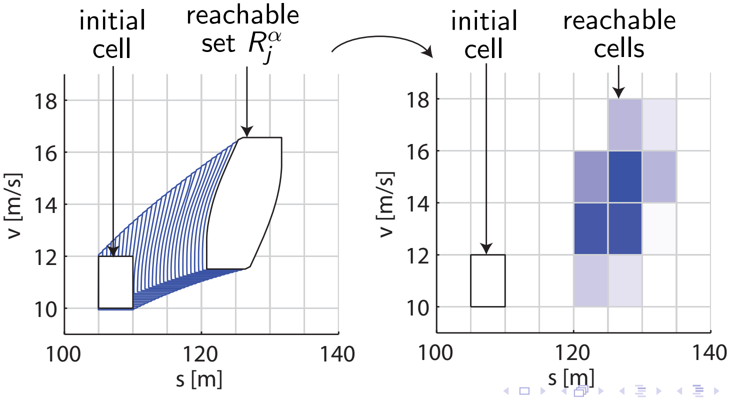


# Abstraction to Markov Chains via Reachable Sets

Geometric determination of the transition probabilities $\Phi_{ij}^{\alpha}$:

$$\Phi_{ij}^{\alpha} = \frac{V(R_j^{\alpha} \cap X_i)}{V(R_j^{\alpha})}, \text{ V(): Volume}$$

$j$: Initial cell, $i$: Cell after transition, *alpha*: Input cell

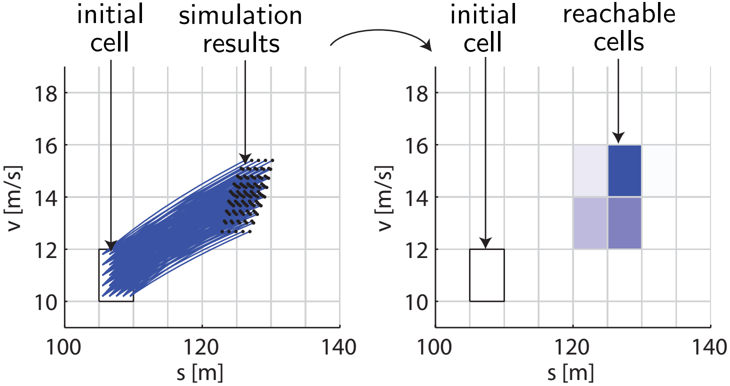


# Abstraction to Markov Chains via Simulations

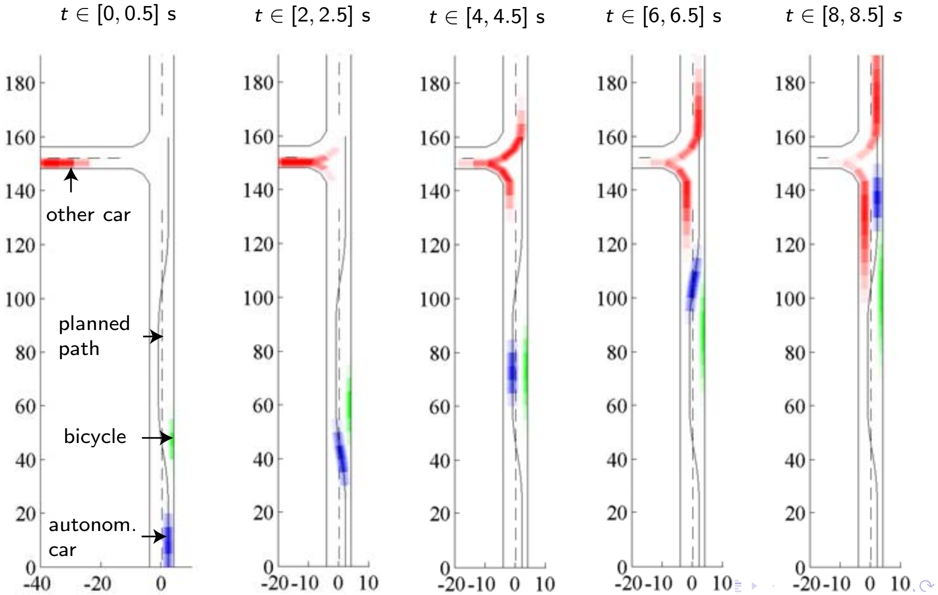
Counting the number $N$ of final states in cells:

$$\Phi_{ij}^{\alpha} = \frac{N_{ij}^{\alpha}}{\sum_i N_{ij}^{\alpha}}.$$

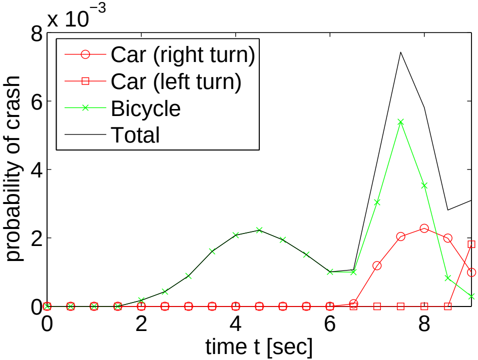
$j$: Initial cell, $i$: Cell after transition, *alpha*: Input cell



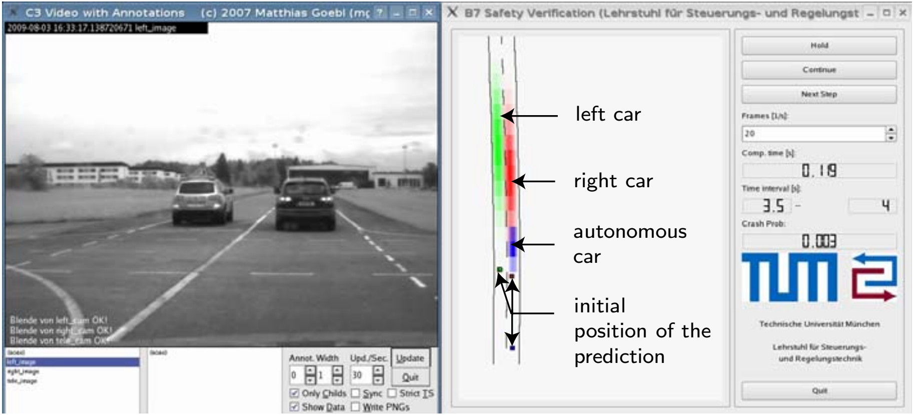
# Overtaking Scenario



# Overtaking Scenario



# Test Drive



C3 Video with Annotations (c) 2007 Matthias Goebl (m

2009-08-03 16:33:17.138720671 left_image

Blende von left_cam OK!
Blende von right_cam OK!
Blende von tele_cam OK!

Annot. Width: 0, 1

Upd./Sec.: 30

☑ Only Childs ☐ Sync ☐ Strict TS

☑ Show Data ☐ Write PNGs

B7 Safety Verification (Lehrstuhl für Steuerungs- und Regelungst

Frames [1/s]: 20

Comp. time [s]: 0.119

Time interval [s]: 3.5 – 4

Crash Prob: 0.003

Technische Universität München

Lehrstuhl für Steuerungs- und Regelungstechnik

left car

right car

autonomous car

initial position of the prediction

# Conclusions

- Traffic is inherently unsafe → Stochastic verification.
- Non-stochastic safety verification is only useful when all vehicles broadcast their plans.
- Separation into longitudinal & lateral dynamics saves computational time.

Things that have not been presented:

- Adaption of the longitudinal dynamics according to lane curvature, speed limit, interaction with traffic participants, lane changing.
- Comparison with Monte Carlo simulation:
    - Probability distribution: Markov chain abstraction is better.
    - Crash probability: Monte Carlo simulation is better.