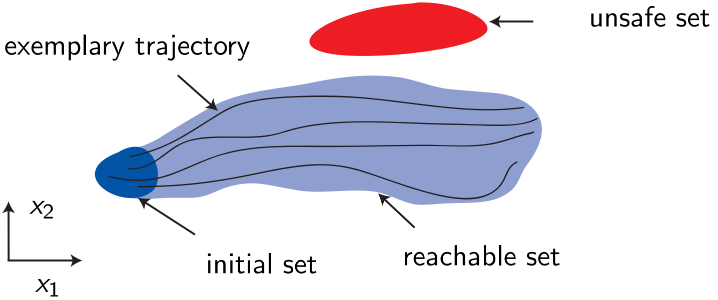
Advances in Reachability Analysis with Applications to Safety Verification of Vehicle Control Systems

Matthias Althoff, Colas Le Guernic, and Bruce H. Krogh

Carnegie Mellon University
New York University

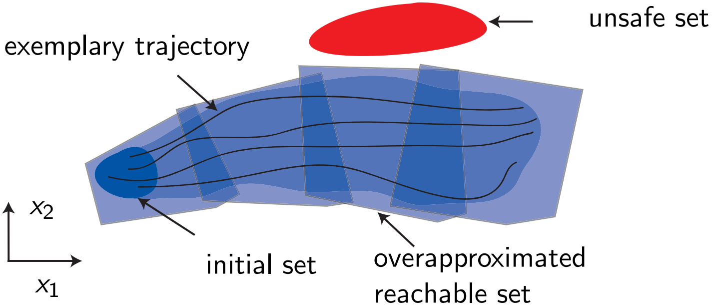
April 27, 2011

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 $\rightarrow$ real system is safe.

Main Innovations

Consideration of Time-Varying Parameters for Linear Systems

There is much work for linear time invariant (LTI) Systems; a wrapping-free algorithm exists [A. Girard, C. Le Guernic, O. Maler; HSCC 2006].

Here: The system matrix is uncertain and time-varying.

Novel Linearization Approach for Nonlinear Systems

Before: The linearization error is considered by an additional uncertain input.

Here: The linearization error is considered by adding parameter uncertainties.

Continuization of Hybrid Systems

Before: Hybrid dynamics requires intersection of reachable sets with guard sets.

Here: The intersection can be eliminated by temporarily enlarging the set of uncertain parameters.

Considered Class of Systems

Linear systems with uncertain time varying parameters

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

where $A : \mathbb{R}^+ \rightarrow \mathcal{A}$, $u : \mathbb{R}^+ \rightarrow \mathcal{U}$ are piecewise continuous, and $\mathcal{A} \subset \mathbb{R}^{n \times n}$, $\mathcal{U} \subset \mathbb{R}^n$. For reachability analysis, we consider all possible functions $A(t)$ and $u(t)$.

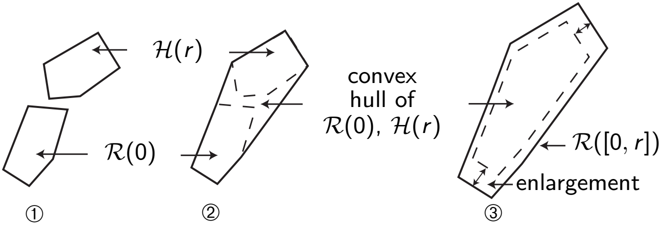
Example:

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

$$\mathcal{U} = \begin{pmatrix} 1 \\ 1 \end{pmatrix} [-0.05, 0.05]$$

Overview of Reachable Set Computation

1. Compute reachable set $\mathcal{H}(r)$ at time r when there is no input.
   Input not yet considered.
2. Obtain convex hull of initial set $\mathcal{R}(0)$ and $\mathcal{H}(r)$.
   Curvature of trajectories not yet considered.
3. Enlarge reachable set to account for (1) uncertain inputs, (2) curvature of trajectories.
4. Continue with further time intervals $[kr, (k+1)r]$, $k \in \mathbb{N}$.



Peano Baker Series

Superposition principle: First, consider only the initial state solution

$$x(t) = \Phi(A(\tau), t)x_0,$$

where $\Phi(A(\tau), t)$ is referred to as the Peano Baker Series.

Peano Baker Series

$$\begin{aligned}\Phi(A(\tau), t) = & \mathrm{I} + \int_0^t A(\sigma_1)d\sigma_1 + \int_0^t A(\sigma_1) \int_0^{\sigma_1} A(\sigma_2)\, d\sigma_2\, d\sigma_1 \\ & + \int_0^t A(\sigma_1) \int_0^{\sigma_1} A(\sigma_2) \int_0^{\sigma_2} A(\sigma_3)\, d\sigma_3\, d\sigma_2\, d\sigma_1 + \dots\end{aligned}$$

How to compute the set $\{\Phi(A(\tau), t) | A(\tau) \in \mathcal{A}\}$*?*

Overapproximation of the Peano Baker Series

1. Time discretization: $\int_0^t A(\sigma_i)d\sigma_i \approx \sum_{l_i=1}^{k} A(l_i\Delta)\Delta,\ t = k\Delta$ (Riemann integration).

Approximate $\Phi(A(\tau), t)$ iteratively as

$$\tilde{\Phi}_1(A(\tau), k, \Delta) = \mathtt{I} + \sum_{l_1=1}^{k} A(l_1\Delta)\Delta,$$

$$\tilde{\Phi}_i(A(\tau), k, \Delta) = \tilde{\Phi}_{i-1}(t, \Delta) + \sum_{l_i=1}^{k} \ldots \sum_{l_1=1}^{l_2} \left(\prod_{q=1}^{i} A(l_q\Delta) \right) \Delta^i,$$

Reminder: $$\Phi(A(\tau), t) = \underbrace{\overbrace{\mathtt{I} + \int_0^t A(\sigma_1)d\sigma_1}^{i=1} + \int_0^t A(\sigma_1) \int_0^{\sigma_1} A(\sigma_2)\, d\sigma_2\, d\sigma_1}_{i=2} + \ldots$$

Overapproximation of the Peano Baker Series

1. Time discretization: $\int_0^t A(\sigma_i)d\sigma_i \approx \sum_{l_i=1}^k A(l_i\Delta)\Delta$, $t = k\Delta$ (Riemann integration).
2. Replace concrete matrices by sets of matrices.

Approximate $\Phi(A(\tau), t)$ iteratively as

$$\tilde{\Phi}_1(A(\tau), k, \Delta) = \mathrm{I} + \underbrace{\sum_{l_1=1}^{k} A(l_1\Delta)\Delta}_{\in \bigoplus_{l_1=1}^{k} \mathcal{A}\Delta},$$

$$\tilde{\Phi}_i(A(\tau), k, \Delta) = \tilde{\Phi}_{i-1}(t, \Delta) + \underbrace{\sum_{l_i=1}^{k} \dots \sum_{l_1=1}^{l_2} \left(\prod_{q=1}^{i} A(l_q\Delta) \right) \Delta^i}_{\in \bigoplus_{l_i=1}^{k} \dots \bigoplus_{l_1=1}^{l_2} \mathcal{A}^i \Delta^i},$$

where $\oplus$ represents the Minkowski addition: $\mathcal{A} \oplus \mathcal{B} = \{A + B \mid A \in \mathcal{A}, B \in \mathcal{B}\}$.

Overapproximation of the Peano Baker Series

1. Time discretization: $\int_0^t A(\sigma_i)d\sigma_i \approx \sum_{l_i=1}^k A(l_i\Delta)\Delta$, $t = k\Delta$ (Riemann integration).
2. Replace concrete matrices by sets of matrices.
3. Apply distributivity of convex matrix sets: $a\mathcal{A} \oplus b\mathcal{A} = (a+b)\mathcal{A}$

Approximate $\Phi(A(\tau), t)$ iteratively as

$$\tilde{\Phi}_1(A(\tau), k, \Delta) \in \mathtt{I} \oplus \underbrace{\bigoplus_{l_1=1}^{k} \mathcal{A}\Delta}_{\subseteq \mathtt{CH}(\mathcal{A})t},$$

$$\tilde{\Phi}_i(A(\tau), k, \Delta) \in \tilde{\Phi}_{i-1}(t, \Delta) \oplus \underbrace{\bigoplus_{l_i=1}^{k} \cdots \bigoplus_{l_1=1}^{l_2} \mathcal{A}^i\Delta^i}_{\subseteq \frac{1}{i!}\mathtt{CH}(\mathcal{A}^i)t^i =: \overline{\mathcal{M}}_i(t)},$$

where $\mathtt{CH}()$ is the convex hull operator, which ensures that the distributivity law can be applied.

Overapproximation of the State Transition Matrix

The expressions $\overline{\mathcal{M}}_i(t)$ are independent of Δ. For $\lim_{\Delta \to 0}$ we have that

Overapproximation of the state transition matrix

$$\Phi(A(\tau), t) \in \bigoplus_{i=0}^{\infty} \overline{\mathcal{M}}_i(t), \qquad \overline{\mathcal{M}}_i(t) = \frac{t^i}{i!} \mathtt{CH}(\mathcal{A}^i).$$

Overapproximation of the state transition matrix: time invariant case

$$\Phi(A, t) \in \left\{ \sum_{i=0}^{\infty} \frac{t^i}{i!} A^i \middle| A \in \mathcal{A} \right\}.$$

Overapproximation of the State Transition Matrix

The expressions $\overline{\mathcal{M}}_i(t)$ are independent of Δ. For $\lim_{\Delta \to 0}$ we have that

Overapproximation of the state transition matrix

$$\Phi(A(\tau), t) \in \bigoplus_{i=0}^{\infty} \overline{\mathcal{M}}_i(t), \qquad \overline{\mathcal{M}}_i(t) = \frac{t^i}{i!}\mathtt{CH}(\mathcal{A}^i).$$

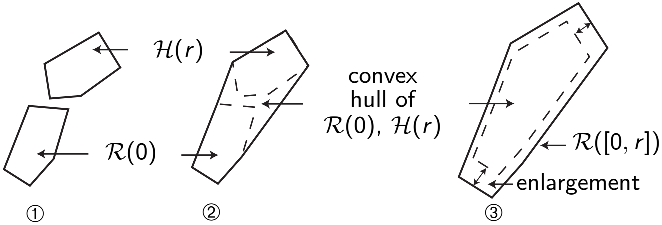
Overapproximation of the state transition matrix by a finite sum

$$\Phi_i(A(\tau), t) \in \bigoplus_{i=0}^{\eta} \overline{\mathcal{M}}_i(t) \oplus [-W(t), W(t)],$$

$W(t)$: remainder bound

Overview of Reachable Set Computation

1. Compute reachable set $\mathcal{H}(r)$ at time r when there is no input.
   done
2. Obtain convex hull of initial set $\mathcal{R}(0)$ and $\mathcal{H}(r)$.
   trivial
3. Enlarge reachable set to account for (1) uncertain inputs (*next slide*), (2) curvature of trajectories (*skipped*).
4. Continue with further time intervals $[kr, (k+1)r]$, $k \in \mathbb{N}$.



Input Solution

Removing the input

The differential equation $\dot{x}(t) = A(t)x(t) + u(t)$ can be rewritten as

$$\frac{d}{dt}\begin{pmatrix} x(t) \\ 1 \end{pmatrix} = \underbrace{\begin{pmatrix} A(t) & u(t) \\ 0 & 0 \end{pmatrix}}_{A_u(t)} \begin{pmatrix} x(t) \\ 1 \end{pmatrix}$$

...analogous proofs ...

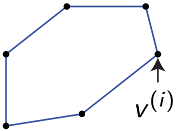
Reachable set due to the input

$$\mathcal{P}(t) = \bigoplus_{i=0}^{\eta} \left(\frac{t^{i+1}}{(i+1)!} \mathtt{CH}(\mathcal{A}^i \mathcal{U}) \right) \oplus \frac{t}{\eta+2} [-W(t), W(t)] \{|\mathcal{U}|\} .$$

Typical Types of Sets for Reachability Analysis

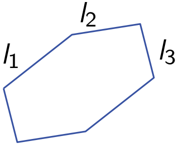
Polytopes: Convex hull of vertices

$$\left\{ \sum_{i=1}^{r_A} \alpha_i v^{(i)} \middle| v^{(i)} \in \mathbb{R}^n, \alpha_i \geq 0, \sum_i \alpha_i = 1 \right\}$$



Zonotopes: Minkowski sum of line segments $l_i = [-1, 1] g^{(i)}$

$$\left\{ g^{(0)} + \sum_{i=1}^{\kappa_A} p_i g^{(i)} \middle| g^{(i)} \in \mathbb{R}^n, p_i \in [-1, 1] \right\}$$



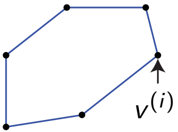
Interval Vector

$$[\underline{a}, \overline{a}], \quad \forall i : \underline{a}_i \leq \overline{a}_i, \quad \underline{a}, \overline{a} \in \mathbb{R}^n.$$

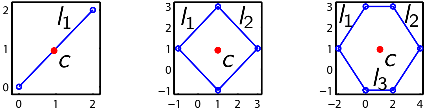
Typical Types of Sets for Reachability Analysis

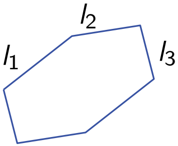
Polytopes: Convex hull of vertices

$$\left\{ \sum_{i=1}^{r_A} \alpha_i v^{(i)} \middle| v^{(i)} \in \mathbb{R}^n, \alpha_i \geq 0, \sum_i \alpha_i = 1 \right\}$$



Zonotopes: Minkowski sum of line segments
$l_i = [-1, 1] g^{(i)}$





Interval Vector

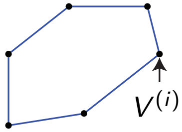
$$[\underline{a}, \overline{a}], \quad \forall i : \underline{a}_i \leq \overline{a}_i, \quad \underline{a}, \overline{a} \in \mathbb{R}^n.$$

Considered Matrix Sets for $\mathcal{A}$

Analogous definitions for matrix sets:

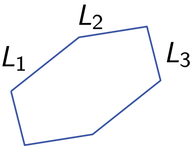
Matrix Polytopes: Convex hull of matrices

$$\left\{ \sum_{i=1}^{r_A} \alpha_i V^{(i)} \middle| V^{(i)} \in \mathbb{R}^{n\times n}, \alpha_i \geq 0, \sum_i \alpha_i = 1 \right\}$$



Matrix Zonotopes: Minkowski sum of "matrix line segments" $L_i = [-1,1]G^{(i)}$

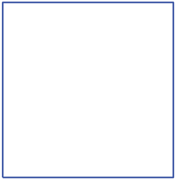
$$\left\{ G^{(0)} + \sum_{i=1}^{\kappa_A} p_i G^{(i)} \middle| G^{(i)} \in \mathbb{R}^{n\times n}, p_i \in [-1,1] \right\}$$



Interval Matrix

$$[\underline{A}, \overline{A}], \quad \forall i,j : \underline{A}_{ij} \leq \overline{A}_{ij}, \quad \underline{A}, \overline{A} \in \mathbb{R}^{n\times n}.$$

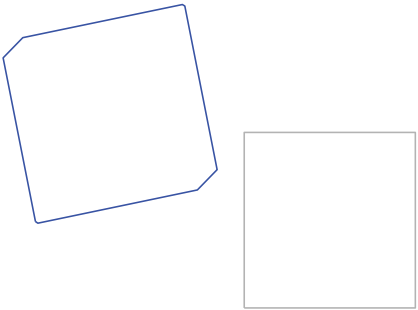
Reachability Algorithm



Compute $R([0, t_f])$

$$\mathcal{H}_0 = \mathtt{CH}(R(0) \cup \overline{\mathcal{M}}(r)R(0)) \oplus \mathcal{F}(r)R(0)$$
$$\mathcal{P}_0 = \mathcal{P}(r)$$
$$R_0 = \mathcal{H}_0 \oplus \mathcal{P}_0$$
for $k = 1 \dots t_f/r - 1$ **do**
$$R_k = \overline{\mathcal{M}}(r)R_{k-1} \oplus \mathcal{P}_0$$
end for
$$R([0, t_f]) = \bigcup_{k=1}^{t_f/r} R_{k-1}$$

Reachability Algorithm



Compute $R([0,t_f])$

$$\mathcal{H}_0 = \mathtt{CH}(R(0) \cup \overline{\mathcal{M}}(r)R(0)) \oplus \mathcal{F}(r)R(0)$$
$$\mathcal{P}_0 = \mathcal{P}(r)$$
$$R_0 = \mathcal{H}_0 \oplus \mathcal{P}_0$$

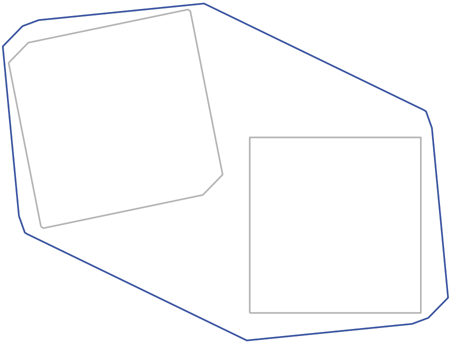
for $k = 1 \ldots t_f/r - 1$ **do**

$$R_k = \overline{\mathcal{M}}(r)R_{k-1} \oplus \mathcal{P}_0$$

end for

$$R([0,t_f]) = \bigcup_{k=1}^{t_f/r} R_{k-1}$$

Reachability Algorithm



Compute $R([0,t_f])$

$$\mathcal{H}_0 = \mathtt{CH}(R(0) \cup \overline{\mathcal{M}}(r)R(0)) \\ \oplus \mathcal{F}(r)R(0)$$

$$\mathcal{P}_0 = \mathcal{P}(r)$$

$$R_0 = \mathcal{H}_0 \oplus \mathcal{P}_0$$

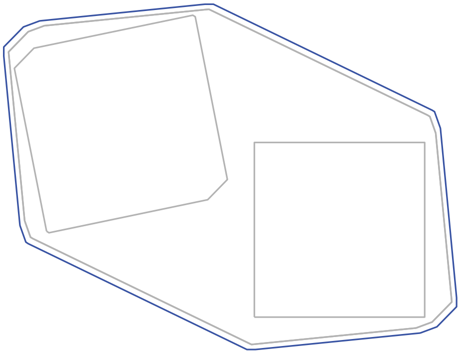
for $k = 1 \ldots t_f/r - 1$ **do**

$$R_k = \overline{\mathcal{M}}(r)R_{k-1} \oplus \mathcal{P}_0$$

end for

$$R([0,t_f]) = \bigcup_{k=1}^{t_f/r} R_{k-1}$$

Reachability Algorithm



Compute $R([0, t_f])$

$$\mathcal{H}_0 = \mathtt{CH}(R(0) \cup \overline{\mathcal{M}}(r)R(0)) \oplus \mathcal{F}(r)R(0)$$

$$\mathcal{P}_0 = \mathcal{P}(r)$$

$$R_0 = \mathcal{H}_0 \oplus \mathcal{P}_0$$

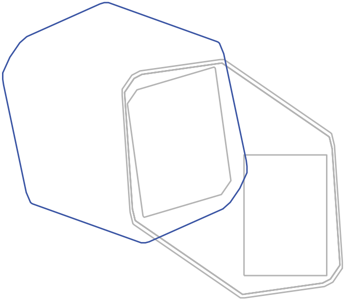
for $k = 1 \ldots t_f/r - 1$ **do**

$$R_k = \overline{\mathcal{M}}(r)R_{k-1} \oplus \mathcal{P}_0$$

end for

$$R([0, t_f]) = \bigcup_{k=1}^{t_f/r} R_{k-1}$$

Reachability Algorithm



Compute $R([0, t_f])$

$$\mathcal{H}_0 = \mathtt{CH}(R(0) \cup \overline{\mathcal{M}}(r)R(0)) \oplus \mathcal{F}(r)R(0)$$

$$\mathcal{P}_0 = \mathcal{P}(r)$$

$$R_0 = \mathcal{H}_0 \oplus \mathcal{P}_0$$

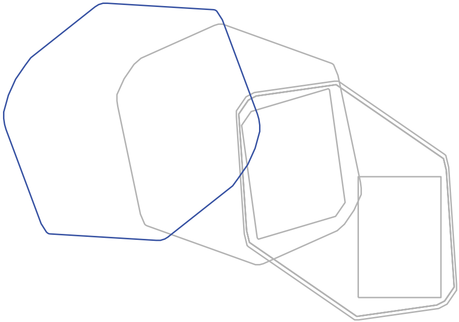
for $k = 1 \ldots t_f/r - 1$ **do**

$$R_k = \overline{\mathcal{M}}(r)R_{k-1} \oplus \mathcal{P}_0$$

end for

$$R([0, t_f]) = \bigcup_{k=1}^{t_f/r} R_{k-1}$$

Reachability Algorithm



Compute $R([0, t_f])$

$$\mathcal{H}_0 = \mathtt{CH}(R(0) \cup \overline{\mathcal{M}}(r)R(0)) \oplus \mathcal{F}(r)R(0)$$

$$\mathcal{P}_0 = \mathcal{P}(r)$$

$$R_0 = \mathcal{H}_0 \oplus \mathcal{P}_0$$

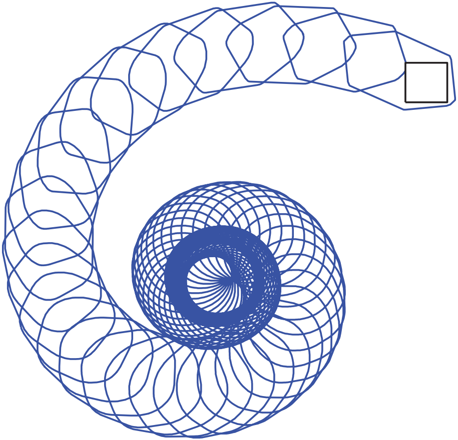
for $k = 1 \ldots t_f/r - 1$ **do**

$$R_k = \overline{\mathcal{M}}(r)R_{k-1} \oplus \mathcal{P}_0$$

end for

$$R([0, t_f]) = \bigcup_{k=1}^{t_f/r} R_{k-1}$$

Reachability Algorithm



Compute $R([0, t_f])$

$$\mathcal{H}_0 = \mathtt{CH}(R(0) \cup \overline{\mathcal{M}}(r)R(0)) \oplus \mathcal{F}(r)R(0)$$

$$\mathcal{P}_0 = \mathcal{P}(r)$$

$$R_0 = \mathcal{H}_0 \oplus \mathcal{P}_0$$

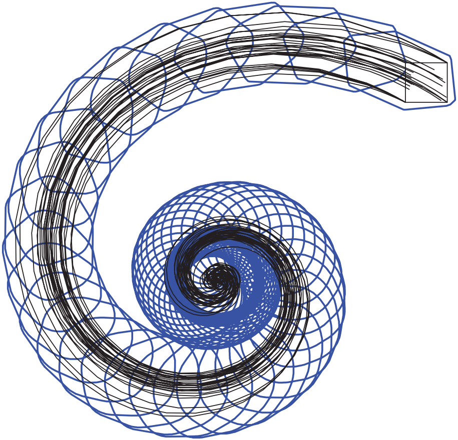
for $k = 1 \dots t_f/r - 1$ **do**

$$R_k = \overline{\mathcal{M}}(r)R_{k-1} \oplus \mathcal{P}_0$$

end for

$$R([0, t_f]) = \bigcup_{k=1}^{t_f/r} R_{k-1}$$

Reachability Algorithm



Compute $R([0, t_f])$

$$\mathcal{H}_0 = \mathtt{CH}(R(0) \cup \overline{\mathcal{M}}(r)R(0)) \oplus \mathcal{F}(r)R(0)$$

$$\mathcal{P}_0 = \mathcal{P}(r)$$

$$R_0 = \mathcal{H}_0 \oplus \mathcal{P}_0$$

for $k = 1 \ldots t_f/r - 1$ **do**

$$R_k = \overline{\mathcal{M}}(r)R_{k-1} \oplus \mathcal{P}_0$$

end for

$$R([0, t_f]) = \bigcup_{k=1}^{t_f/r} R_{k-1}$$

Computation Times of Random Examples

- Random examples of linear systems for 100 time intervals are computed.
- The random system matrices might be unstable; but does not change computation time.

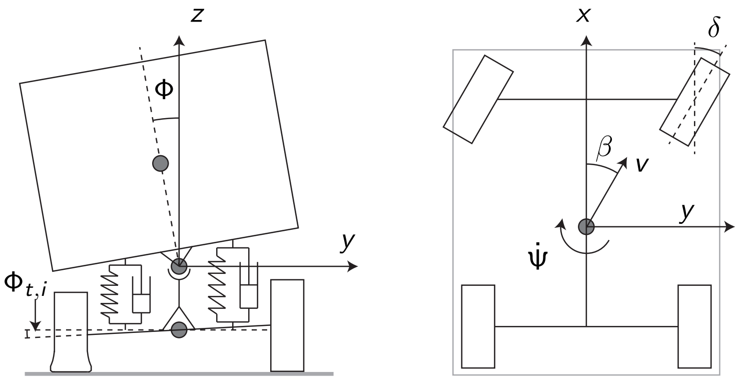
Table: Computation times in [s].

Dimension n	5	10	20	50	100
Interval matrix	0.10	0.12	0.33	0.82	3.64
Matrix zonotope $(\kappa = 1)$	0.13	0.17	0.60	2.65	8.72
Matrix zonotope $(\kappa = 2)$	0.18	0.30	1.13	4.73	18.77
Matrix zonotope $(\kappa = 4)$	0.34	0.68	2.60	18.07	98.70

κ: Number of generator matrices.

computed in MATLAB on an i7 Processor (1.6 GHz) and 6GB memory

Rollover Verification of a Truck



- Considered maneuver: Braking deceleration of $a_x = -0.7g$ (g: gravity constant); acceleration due to steering: $a_y \in [-0.4, 0.4]g$.
- Verification task: Can the vehicle roll over?
- state vector: $x = [\beta, \dot{\Psi}, \Phi, \dot{\Phi}, \Phi_{t,f}, \Phi_{t,r}, v, \int e(t)\, dt]^T$.

Dynamics of the Closed Loop System

truck dynamics (blue variables are states, red ones are inputs):

$$
\begin{aligned}
m x_7(\dot{x}_1 + x_2) - m_S h \dot{x}_4 &= Y_\beta x_1 + Y_{\dot{\psi}}(x_7) x_2 + Y_\delta \delta \\
-I_{xz}\dot{x}_4 + I_{zz}\dot{x}_2 &= N_\beta x_1 + N_{\dot{\psi}}(x_7) x_2 + N_\delta \delta \\
(I_{xx} + m_S h^2)\dot{x}_4 - I_{xz}\dot{x}_2 &= m_S g h x_3 + m_S h x_7(\dot{x}_1 + x_2) - k_f(x_3 - x_5) \\
&\quad - b_f(x_4 - \dot{x}_5) - k_r(x_3 - x_6) - b_r(x_4 - \dot{x}_6) \\
-r(Y_{\beta,f} x_1 + Y_{\dot{\psi},f} x_2 + Y_\delta \delta) &= m_{u,f}(r - h_{u,f}) x_7(\dot{x}_1 + x_2) + m_{u,f} g h_{u,f} x_5 \\
&\quad - k_{t,f} x_5 + k_f(x_3 - x_5) + b_f(x_4 - \dot{x}_5) \\
-r(Y_{\beta,r} x_1 + Y_{\dot{\psi},r} x_2) &= m_{u,r}(r - h_{u,r}) x_7(\dot{x}_1 + x_2) - m_{u,r} g h_{u,r} x_6 \\
&\quad - k_{t,r} x_6 + k_r(x_3 - x_6) + b_r(x_4 - \dot{x}_6) \\
\dot{x}_7 &= a_x.
\end{aligned}
$$

yaw controller:

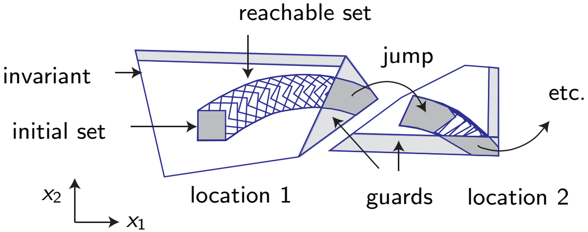
$$
\delta = k_1 e + k_2 \int e(t)\, dt, \quad e = \dot{\Psi}_d - \dot{\Psi} = \dot{\Psi}_d - x_2.
$$

velocity $x_7 \in$	$[10, 20]$ m/s	$[20, 30]$ m/s	$[30, \infty[$ m/s
controller gains	$k_1 = 0.4$ $k_2 = 1.5$	$k_1 = 0.5$ $k_2 = 2$	$k_1 = 0.6$ $k_2 = 2.5$

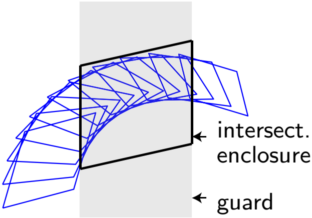
Standard Reachability Analysis of Hybrid Systems

Classical reachability analysis of hybrid systems

Reachable set computation is continued across discrete transitions using intersections with guard sets → Overapproximations due to intersections, overall complexity is not $\mathcal{O}(n^3)$ anymore.



(a) Reachable set of a hybrid system

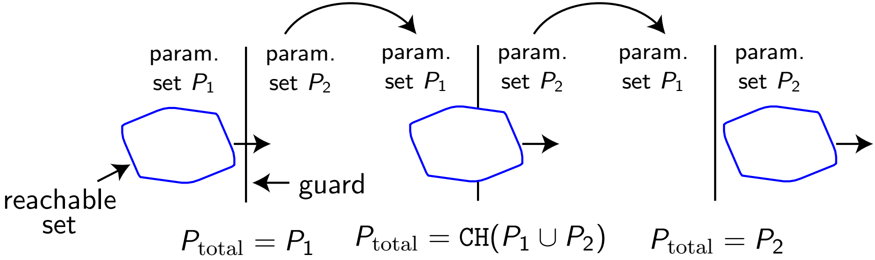


(b) Overapproximation due to guard intersection

Alternative Reachability Analysis of Hybrid Systems

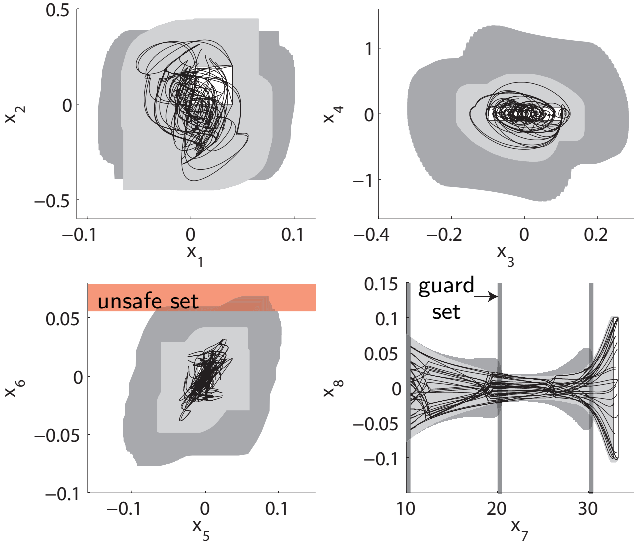
Reachability analysis using continuization

Reachable set is computed under a larger set of parameter uncertainties when intersecting several invariant sets.



- Only applicable if there are no jumps.
- Especially suited if the continuous dynamics does not change much.

Reachable Set of the Truck

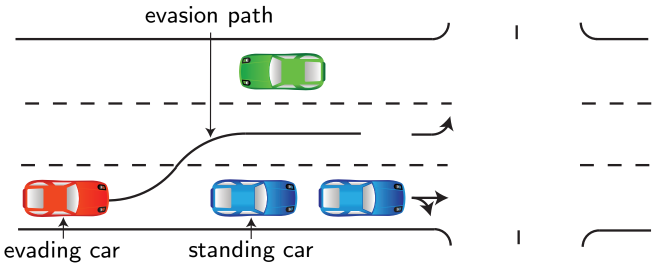


Verification of an Emergency Maneuver

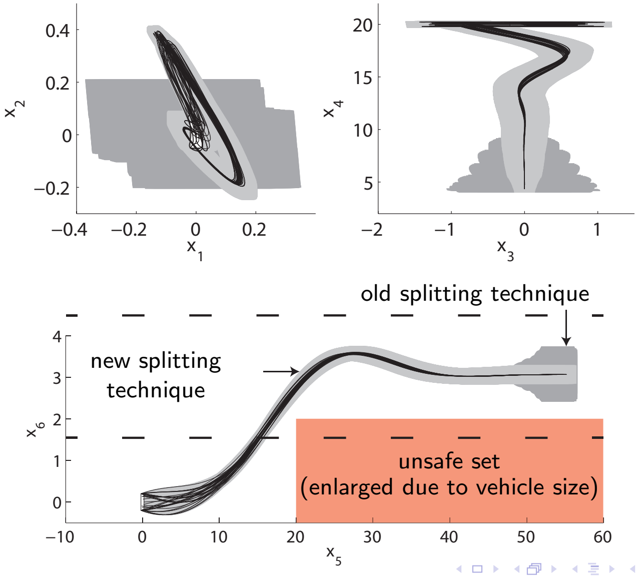
Motivation for automatic evasion maneuver

Crash is inevitable $\rightarrow$ vehicle automatically breaks, or steers, or does both. For velocities greater than $v = \sqrt{8a_{max}w}$, steering is more effective than braking.

a_{max}: maximum acceleration, w: width of the vehicle.



Reachable Set of the Evasive Maneuver



Next Step: Online Verification

Collaborator: Prof. John Dolan (Robotics Institute CMU)

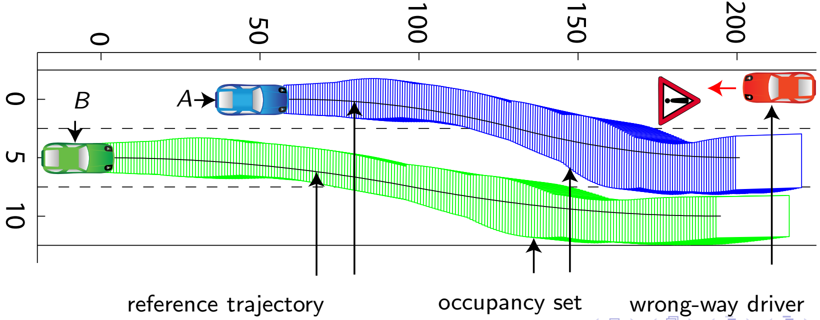
Case Study For Online Verification

Simplifications:

- constant velocity
- reference trajectory consists of arc segments

$\rightarrow$ System is linear.

Computation time including collision checks: 0.39 sec on desktop PC (AMD Athlon64 3700+) in MATLAB.



Conclusions

Reachability Analysis:

- Previous methods for the reachability analysis of LTI systems have been extended to uncertain linear time-varying systems.
- Approach scales well with the number of state variables ($\mathcal{O}(n^3)$).
- Continuization is promising for hybrid systems with similar continuous dynamics in adjacent locations.
- Result makes it possible to apply an alternative linearization approach $\rightarrow$ Further work required.

Automotive Applications:

- Cooperative intersection collision avoidance system (CICAS) with Toyota
- Verification of autonomous cars with the Robotics Institute at CMU.