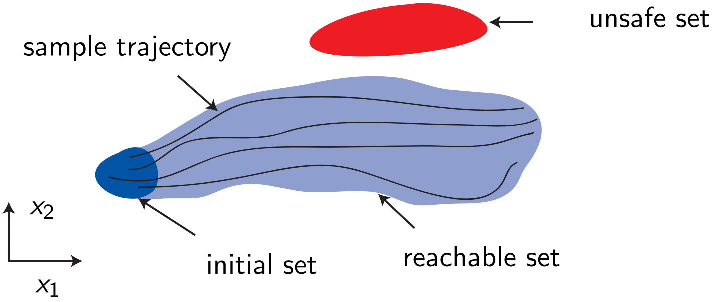
# Reachability Analysis: State of the Art for Various System Classes

Matthias Althoff

Carnegie Mellon University

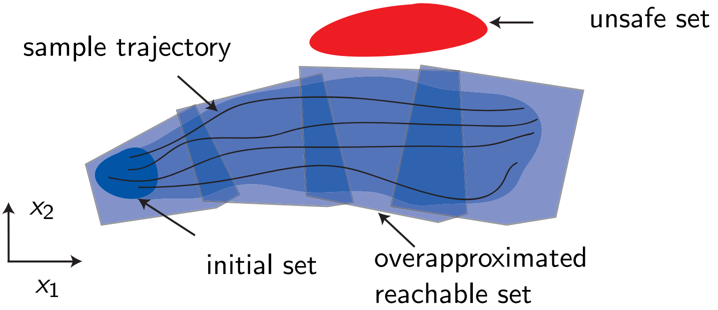
October 19, 2011

# 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.
- Challenge: Compute tight overapproximations while avoiding the curse of dimensionality.

# Overview of Important System Classes

For all system classes we consider

- uncertain initial states $x(0) \in \mathcal{X}$,
- uncertain inputs $u(t) \in \mathcal{U}$,
- finite or infinite time horizons (search for invariant set).

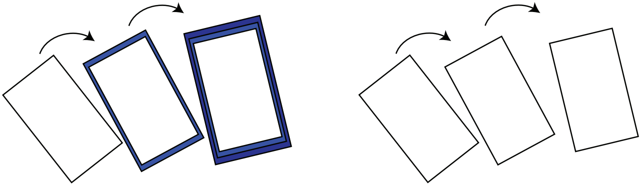
| System class | Dynamics | Cont. var. (best case) | Challenge |
|---|---|---|---|
| linear time invariant (LTI) | $\dot{x} = Ax(t) + Bu(t)$, | 1000 | none |
| LTI with unc. parameters | $\dot{x} = Ax(t) + Bu(t)$, $A \in \mathcal{A}$ | 100 | parameter dependencies |
| nonlinear | $\dot{x} = f(x(t), u(t), p)$, $p$: parameter vector | 100 | linearization errors |
| hybrid | hybrid automaton | 100 | guard intersection |

# Linear Time Invariant (LTI) Systems

Work of Colas Le Guernic and Antoine Girard (2006).

$$\dot{x} = A\,x(t) + Bu(t), \quad x(0) \in X_0,\ u(t) \in U$$

- Scalable ($\mathcal{O}(n^3)$; n: nr of cont. state variables) when using zonotopes or support functions as set representation. More than 1000 state variables in a few minutes.
- First wrapping-free algorithm for LTI-Systems; wrapping-effect: propagation of overapproximations through successive time steps.



with wrapping effect    without wrapping effect

# Linear Systems with Uncertain Parameters

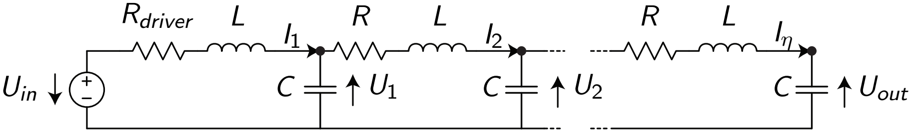
System matrix $A$ is uncertain in a set of matrices $\mathcal{A}$.

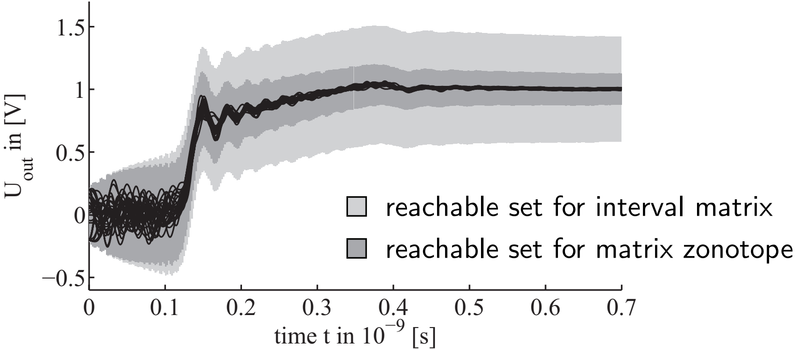
$$\dot{x} = A(t)\,x(t) + Bu(t), \quad x(0) \in X_0,\; u(t) \in U,\; A(t) \in \mathcal{A} \subset \mathbb{R}^{n\times n}$$

- Different algorithms for constant and time varying system matrix $A$.
- No wrapping-free implementation exists.
- Scalable ($\mathcal{O}(n^3)$) when using zonotopes as set representation.
- How to represent uncertainty in parameters?
    - Interval matrices $\mathcal{A} = [\underline{A}, \overline{A}]$,
    - matrix zonotopes $\mathcal{A} = \{C + \sum_{i=1}^{\kappa} \beta_i\, G_i | \beta_i \in [-1,1],\; C, G_i \in \mathbb{R}^{n\times n}\}$,
    - matrix polytopes $\mathcal{A} = \{\sum_{i=1}^{\kappa} \alpha_i V_i | V_i \in \mathbb{R}^{n\times n}, \alpha_i \geq 0, \sum_i \alpha_i = 1\}$.

# RLC circuit

Example: RLC circuit with 40 states.



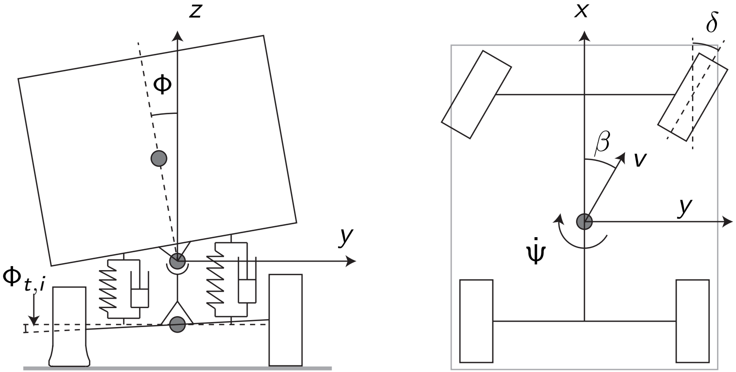
# Nonlinear Systems with Uncertain Parameters

General continuous dynamics described by a Lipschitz continuous function:

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

- Approach is based on linearizing the system dynamics while adding the linearization errors as an additional uncertain input.
- Scalable when using zonotopes.
- Two examples:
  - Rollover verification of a truck.
  - Online verification of autonomous car maneuvers.

# Sketch of the Truck



# Truck Dynamics

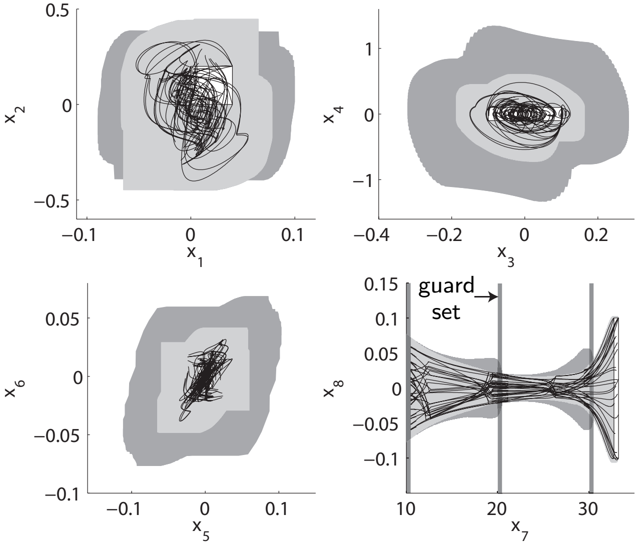
$$
\begin{aligned}
mv(\dot\beta + \dot\Psi) - m_S h\ddot\Phi &= Y_\beta \beta + Y_{\dot\psi}(v)\dot\Psi + Y_\delta \delta \\
-I_{xz}\ddot\Phi + I_{zz}\ddot\Psi &= N_\beta \beta + N_{\dot\psi}(v)\dot\Psi + N_\delta \delta \\
(I_{xx} + m_S h^2)\ddot\Phi - I_{xz}\ddot\Psi &= m_S g h \Phi + m_S v h(\dot\beta + \dot\Psi) - k_f(\Phi - \Phi_{t,f}) \\
&\quad -b_f(\dot\Phi - \dot\Phi_{t,f}) - k_r(\Phi - \Phi_{t,r}) - b_r(\dot\Phi - \dot\Phi_{t,r}) \\
-r(Y_{\beta,f}\beta + Y_{\dot\psi,f}\dot\Psi + Y_\delta\delta) &= m_{u,f} v(r - h_{u,f})(\dot\beta + \dot\Psi) + m_{u,f} g h_{u,f} \Phi_{t,f} \\
&\quad - k_{t,f}\Phi_{t,f} + k_f(\Phi - \Phi_{t,f}) + b_f(\dot\Phi - \dot\Phi_{t,f}) \\
-r(Y_{\beta,r}\beta + Y_{\dot\psi,r}\dot\Psi) &= m_{u,r} v(r - h_{u,r})(\dot\beta + \dot\Psi) - m_{u,r} g h_{u,r} \Phi_{t,r} \\
&\quad - k_{t,r}\Phi_{t,r} + k_r(\Phi - \Phi_{t,r}) + b_r(\dot\Phi - \dot\Phi_{t,r}) \\
\dot v &= a_x.
\end{aligned}
$$

yaw controller:

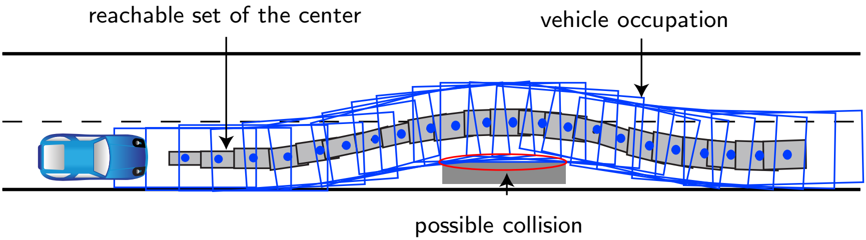
$$
\delta = k_1 e + k_2 \int e(t)\, dt, \quad e = \dot\Psi_d - \dot\Psi.
$$

| $v \in$ | $[10, 20]$ m/s | $[20, 30]$ m/s | $[30, \infty[$ m/s |
|---|---|---|---|
| controller gains | $k_1 = 0.4$ <br> $k_2 = 1.5$ | $k_1 = 0.5$ <br> $k_2 = 2$ | $k_1 = 0.6$ <br> $k_2 = 2.5$ |

# Reachable Set of the Truck



# Online Verification Of Autonomous Cars



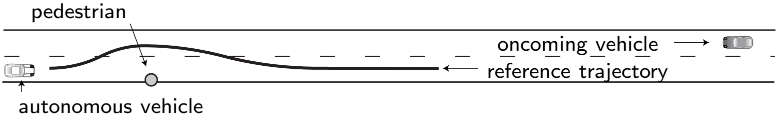

Autonomous vehicles cannot perfectly follow planned trajectories due to

- uncertain initial states,
- uncertain measurements,
- disturbances.

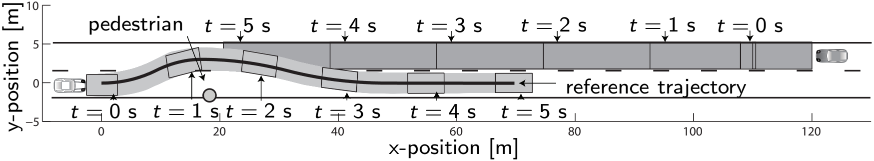
**Consequence:** Planned maneuver is safe under perfect conditions, but may become unsafe due to uncertainties.

# Verification Of Evasive Maneuver

Evasive maneuver due to a pedestrian stepping on the road:



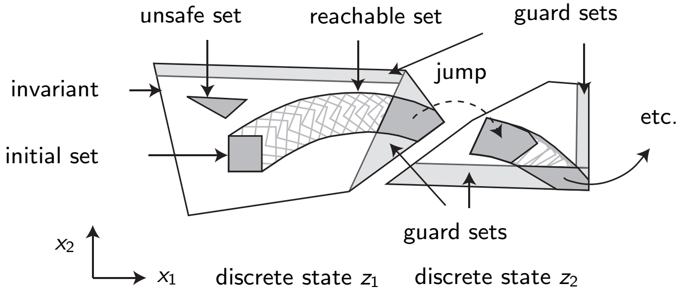

Road Occupancy after reachable set computation:




Computation time in MATLAB on an Intel i7 Processor with 1.6 GHz in 2.24 s $\rightarrow$ Around 2 times faster than maneuver time (5 s).

# Hybrid Systems

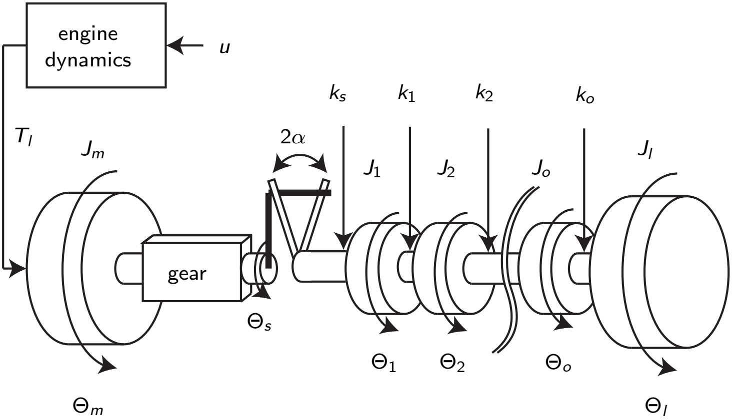
Graphical Description:




- In addition to continuous systems, the intersection with guard sets is required.
- Example: Reachability analysis of a powertrain (up to 100 cont. variables).

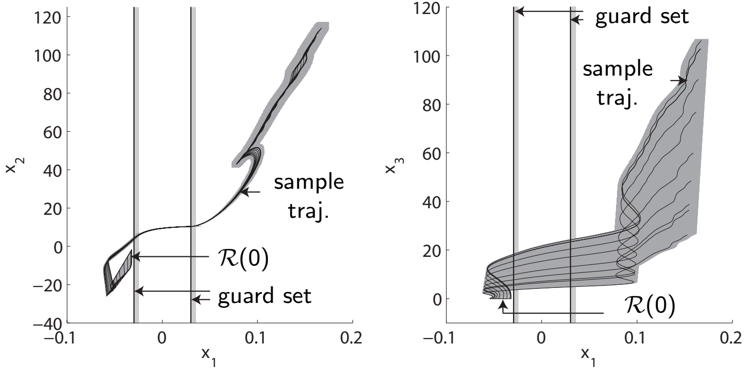
# Model of the Powertrain

Powertrain with arbitrary number of rotating masses:




The system is hybrid due to the consideration of backlash.

# Reachable Set of the Powertrain




Computation times in seconds:

| dim. $n$ | 11 | 21 | 31 | 41 | 51 | 101 |
|---|---|---|---|---|---|---|
| CPU time | 7.327 | 21.96 | 36.84 | 120.2 | 318.8 | 10079 |
| $1^{st}$ guard | 0.247 | 3.454 | 11.99 | 49.36 | 145.8 | 4609 |
| $2^{nd}$ guard | 0.259 | 3.494 | 12.61 | 51.57 | 148.1 | 4975 |

# Conclusions and Discussion

**Conclusions:**

- For all considered system classes (linear, nonlinear, hybrid) new techniques make it possible to consider systems beyond academic examples.
- However: Typical industry systems with several hundred state variables and complex dynamics (hybrid with nonlinear cont. dynamics) are still out of reach.

**Discussion to further improve scalability:**

- Consider verification in the design process:
    - What are subsystems and sub-specifications of the whole system?
    - Can the system design be slightly changed to the advantage of a much simpler verification?
- Can simple models represent complex models when adding uncertainty?