Provably Safe Maneuvers of Automated Vehicles

Matthias Althoff

Technische Universität München

October 07, 2015

EU Project UnCoVerCPS: Partners

Unifying Control and Verification of Cyber-Physical Systems
(UnCoVerCPS)

Funding: 4.9 mio Euro

Participant organisation name	Country
Technische Universität München (TUM)	Germany
Université Joseph Fourier Grenoble 1 (UJF)	France
Universität Kassel (UKS)	Germany
Politecnico di Milano (PoliMi)	Italy
GE Global Research Europe (GE)	Germany
Robert Bosch GmbH (Bosch)	Germany
Esterel Technologies (ET)	France
Deutsches Zentrum für Luft- und Raumfahrt (DLR)	Germany
Tecnalia (Tec)	Spain
R.U.Robots Limited (RUR)	United Kingdom

EU Project UnCoVerCPS: Main objectives

- Novel on-the-fly control and verification concepts.

EU Project UnCoVerCPS: Main objectives

- Novel on-the-fly control and verification concepts.
- Unifying control and verification to quickly react to changing environments.

EU Project UnCoVerCPS: Main objectives

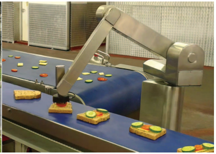
- Novel on-the-fly control and verification concepts.
- Unifying control and verification to quickly react to changing environments.
- A unique tool chain that makes it possible to integrate modeling, control design, formal verification, and automatic code generation.

EU Project UnCoVerCPS: Main objectives

- Novel on-the-fly control and verification concepts.
- Unifying control and verification to quickly react to changing environments.
- A unique tool chain that makes it possible to integrate modeling, control design, formal verification, and automatic code generation.
- Prototypical realizations for automated vehicles, human-robot collaborative manufacturing, wind turbines and smart grids.

EU Project UnCoVerCPS: Main objectives

- Novel on-the-fly control and verification concepts.
- Unifying control and verification to quickly react to changing environments.
- A unique tool chain that makes it possible to integrate modeling, control design, formal verification, and automatic code generation.
- Prototypical realizations for automated vehicles, human-robot collaborative manufacturing, wind turbines and smart grids.
- A new development process that reduces development time and costs for critical cyber-physical systems.

Focus Of This Talk: Online Verification of Automated Cars

Thought experiment

How many possible situations is an automated car facing?

Focus Of This Talk: Online Verification of Automated Cars

Thought experiment

How many possible situations is an automated car facing?

- Infinitely many!

Focus Of This Talk: Online Verification of Automated Cars

Thought experiment

How many possible situations is an automated car facing?

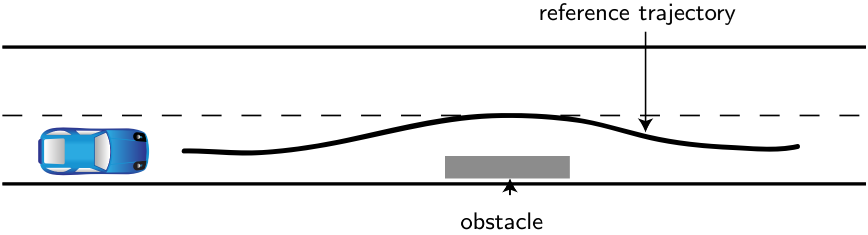
- Infinitely many!
- O.k., let's discretize (see details below): At least 10^{81} (outnumbers the estimated amount of atoms in the universe)

Problem: How should this be verified upfront?
Solution: We have to verify the vehicle while it is in operation
$\rightarrow$ online verification.

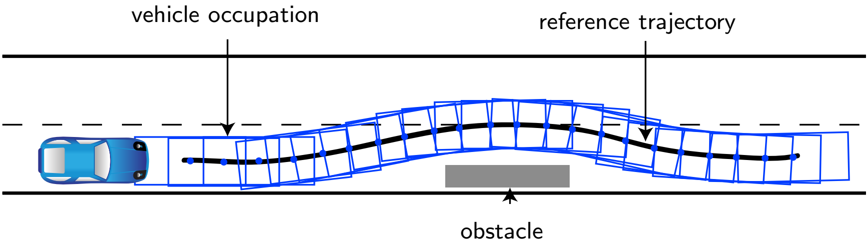
Discretization of the problem:

- Each surrounding vehicle: position (x- and y-coordinate), velocity and orientation (4 variables).
- Each lane: width, curvature, and change of curvature (3 variables).
- Own vehicle: x- and y-coordinate, velocity, orientation, yaw rate, steering angle (6 state variables) and tire-road friction, current loading (2 variables).
- Bounds on numbers of variable values and objects: 20 values per variable, maximum 10 surrounding vehicles, 5 lanes.
- Result: $(20^4)^{10} \cdot (20^3)^5 \cdot 20^6 \cdot 20^2 \approx 10^{81}$.

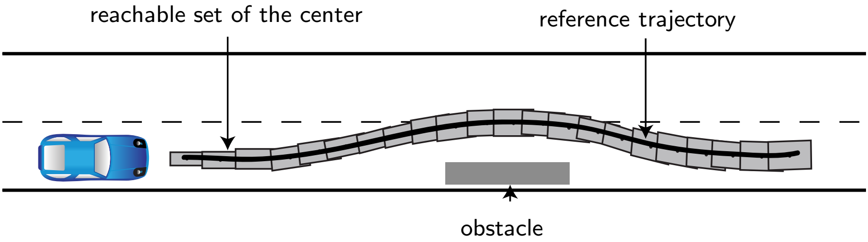
Trajectory Verification: Situation



Trajectory Verification: Standard Approach



Trajectory Verification: Considering Uncertainties



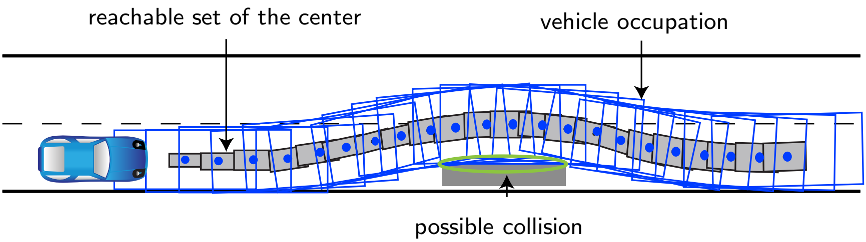
Robust Safety Problem

Is the planned maneuver of the autonomous vehicle still safe under

- uncertain initial states,
- uncertain measurements,
- and disturbances?

Objective: Guarantee safety when bounds on uncertainties are known.

Trajectory Verification: Formal Verification Reveals Problems



Robust Safety Problem

Is the planned maneuver of the autonomous vehicle still safe under

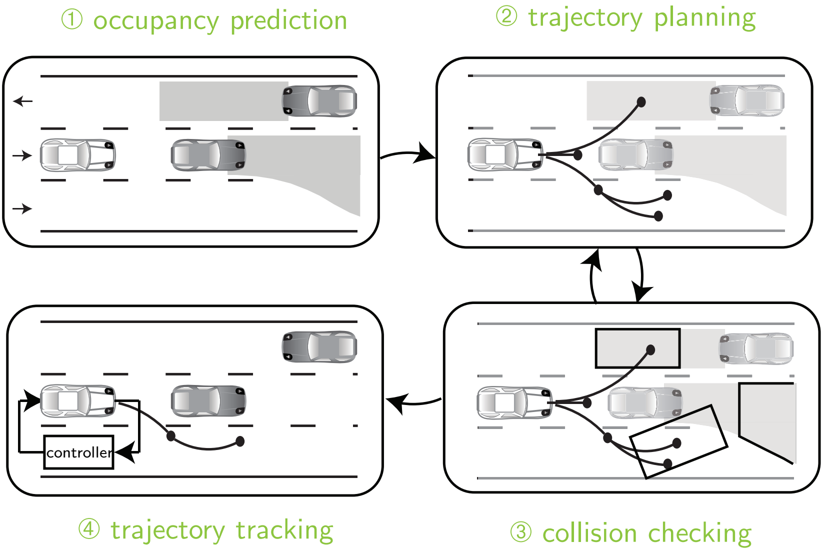
- uncertain initial states,
- uncertain measurements,
- and disturbances?

Objective: Guarantee safety when bounds on uncertainties are known.

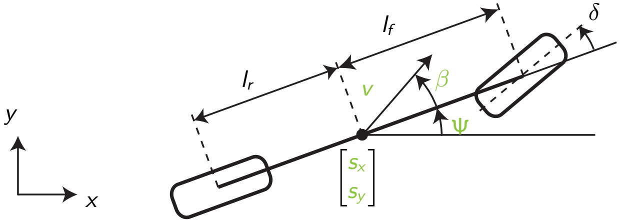
Outline

1. Overview of the approach
2. Models of the ego vehicle and other traffic participants
3. Verification procedure
4. Test results
5. Verification of high-fidelity vehicle models
6. Pre-computation using motion primitives
7. Adaptation for automated driving

Overview of the Approach



Model of the Uncontrolled Ego Vehicle



$$\dot{\beta} = \left(\frac{C_r l_r - C_f l_f}{m v^2} - 1\right)\dot{\Psi} + \frac{1}{m v}\left(C_f \delta - (C_f + C_r)\beta\right)$$
$$\dot{\Psi} = \dot{\Psi}$$
$$\ddot{\Psi} = \frac{1}{I_z}\left((l_r C_r - l_f C_f)\beta - (l_f^2 C_f + l_r^2 C_r)\frac{\dot{\Psi}}{v} + l_f C_f \delta\right)$$

yaw dynamics

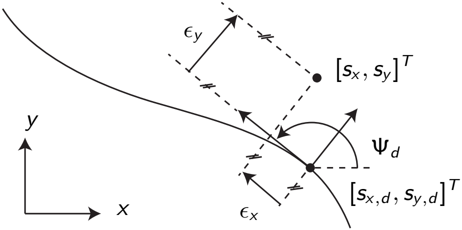
$$\dot{v} = a_x$$

longitudinal dynamics

$$\dot{s}_x = v \cos(\beta + \Psi)$$
$$\dot{s}_y = v \sin(\beta + \Psi)$$

position of the vehicle

Tracking Controller



$$\delta = k_1(\epsilon_y + u_{\epsilon_y}) + k_2(\Psi_d - \Psi - u_\Psi) + k_3(\dot{\Psi}_d - \dot{\Psi} - u_{\dot{\Psi}}) \quad \text{steering control}$$

$$a_x = k_4(\epsilon_x + u_{\epsilon_x}) + k_5(v_d - v - u_v) \quad \text{longitudinal control}$$

- Reference values: Ψ_d, $\dot{\Psi}_d$, v_d.
- Sensor noises: u_{ϵ_x}, u_{ϵ_y}, u_Ψ, $u_{\dot{\Psi}}$, u_v.
- Combining the vehicle model and the control laws yields the final model.

Constraints for Traffic Participants

Initially the following constraints are considered:

*C*1: positive longitudinal acceleration is stopped when a parameterized speed v_{max} is reached.

Constraints for Traffic Participants

Initially the following constraints are considered:

$C1$: positive longitudinal acceleration is stopped when a parameterized speed v_{max} is reached.

$C2$: driving backwards in a lane is not allowed.

Constraints for Traffic Participants

Initially the following constraints are considered:

- $C1$: positive longitudinal acceleration is stopped when a parameterized speed v_{max} is reached.
- $C2$: driving backwards in a lane is not allowed.
- $C3$: positive longitudinal acceleration is inversely proportional with speed above a parameterized speed v_S (modeling a maximum engine power).

Constraints for Traffic Participants

Initially the following constraints are considered:

$C1$: positive longitudinal acceleration is stopped when a parameterized speed v_{max} is reached.

$C2$: driving backwards in a lane is not allowed.

$C3$: positive longitudinal acceleration is inversely proportional with speed above a parameterized speed v_S (modeling a maximum engine power).

$C4$: maximum absolute acceleration is limited by a_{max}.

Constraints for Traffic Participants

Initially the following constraints are considered:

$C1$: positive longitudinal acceleration is stopped when a parameterized speed $v_{\max}$ is reached.

$C2$: driving backwards in a lane is not allowed.

$C3$: positive longitudinal acceleration is inversely proportional with speed above a parameterized speed v_S (modeling a maximum engine power).

$C4$: maximum absolute acceleration is limited by $a_{\max}$.

$C5$: actions that cause leaving the road/lane boundary are forbidden.

Constraints for Traffic Participants

Initially the following constraints are considered:

$C1$: positive longitudinal acceleration is stopped when a parameterized speed $v_{\max}$ is reached.

$C2$: driving backwards in a lane is not allowed.

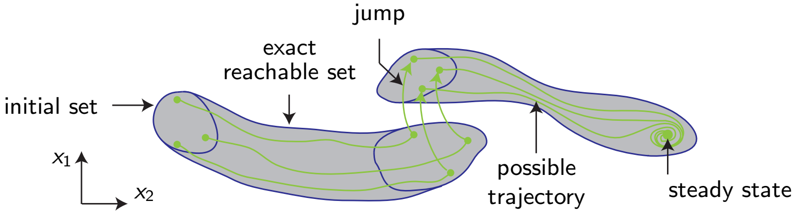
$C3$: positive longitudinal acceleration is inversely proportional with speed above a parameterized speed v_S (modeling a maximum engine power).

$C4$: maximum absolute acceleration is limited by $a_{\max}$.

$C5$: actions that cause leaving the road/lane boundary are forbidden.

When a violation of a constraint of a traffic participant is sensed, it is no longer considered in future predictions for that particular traffic participant.

Reachability Analysis

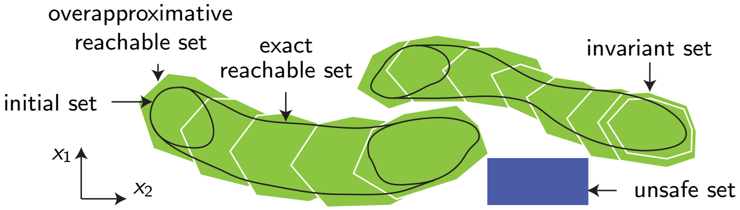


Informal Definition

A reachable set is the set of states that can be reached by a dynamical system in finite or infinite time for a

- set of initial states,
- uncertain inputs,
- and uncertain parameters.

Overapproximative Reachable Sets

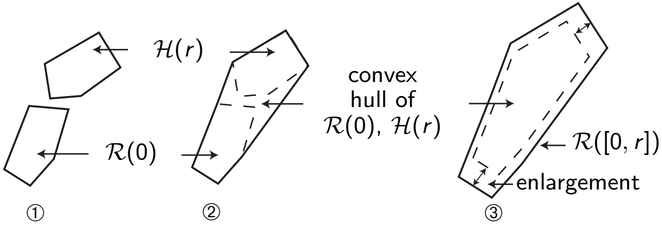


- Exact reachable set only for special classes computable
  $\rightarrow$ overapproximation computed for consecutive time intervals.
- Overapproximation might lead to spurious counterexamples.
- Simulation cannot prove correctness.

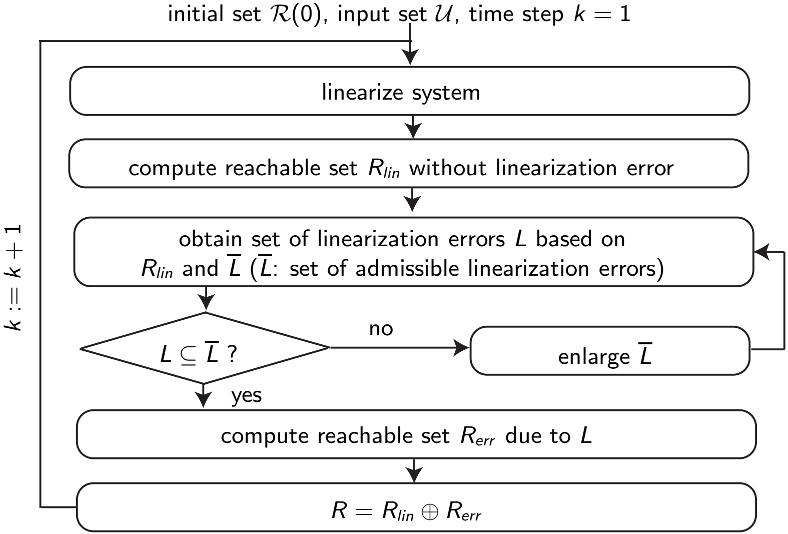
Linear Systems: Overview of Reachable Set Computation

$\dot{x}(t) = Ax(t) + u(t), \quad A \in \mathbb{R}^{n\times n}, \quad x(t) \in \mathbb{R}^n, \quad x(0) \in \mathcal{R}(0), \quad u(t) \in u_c \oplus \mathcal{U}$

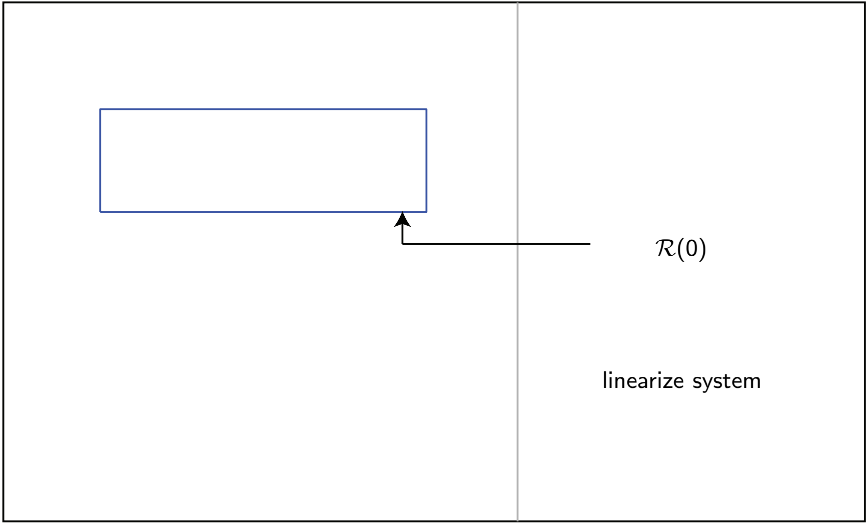
1. Compute reachable set $\mathcal{H}(r) = e^{Ar}\mathcal{R}(0) \oplus \int_{t=0}^{r} e^{A(r-t)}\mathrm{d}t\, u_c$ at time r neglecting the uncertain input ($\mathcal{C} \oplus \mathcal{D} := \{c + d | c \in \mathcal{C}, d \in \mathcal{D}\}$).
2. Obtain convex hull of initial set $\mathcal{R}(0)$ and $\mathcal{H}(r)$.
3. Enlarge reachable set to account for (1) uncertain inputs, (2) curvature of trajectories.
4. Continue with further time intervals $[kr, (k+1)r]$, $k \in \mathbb{N}$.



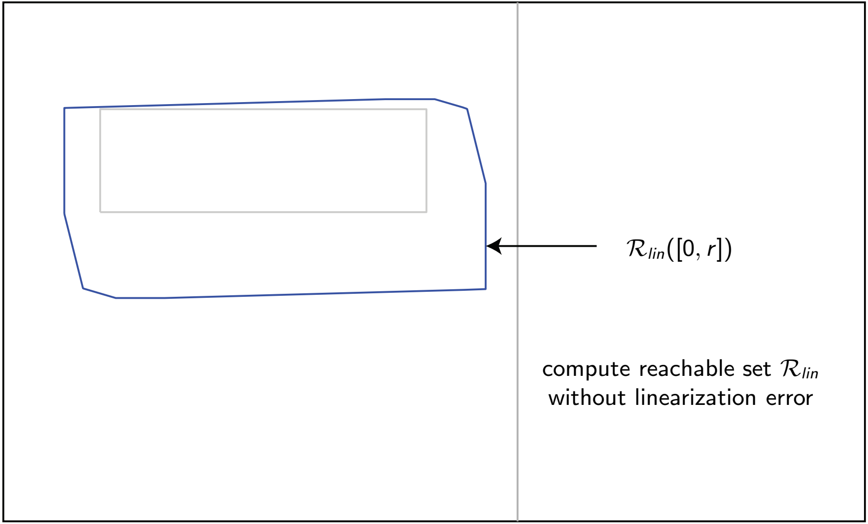
Nonlinear Reachability Analysis: Overall Algorithm



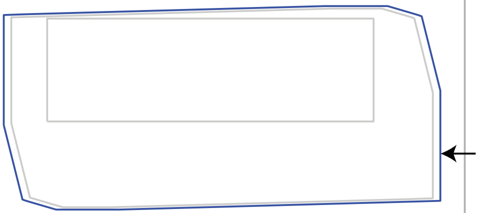
Overall Algorithm: Animation (I)



Overall Algorithm: Animation (II)



Overall Algorithm: Animation (III)



$\mathcal{R}_{lin}([0, r]) \oplus \overline{\mathcal{R}}_{err}([0, r])$

$\overline{\mathcal{R}}_{err}$: reachable set due to $\overline{\mathcal{L}}$

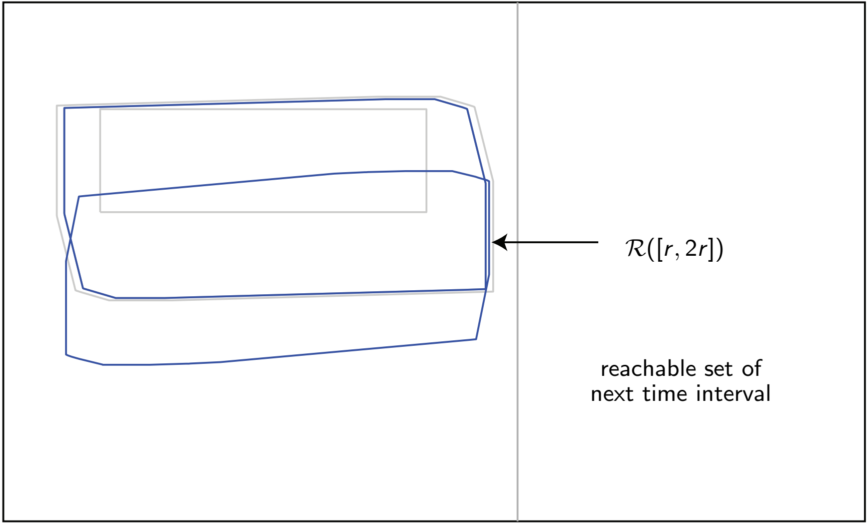
obtain set of linearization errors $\mathcal{L}$ based on $\mathcal{R}_{lin}([0, r]) \oplus \overline{\mathcal{R}}_{err}([0, r])$

Overall Algorithm: Animation (IV)

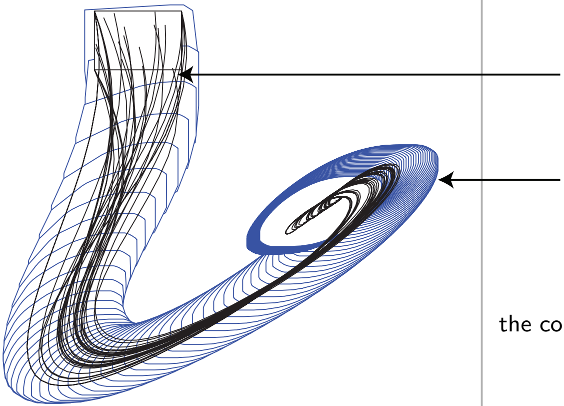
$$\mathcal{R}([0,r]) = \mathcal{R}_{lin}([0,r]) \oplus \mathcal{R}_{err}([0,r])$$

$$\mathcal{L} \subseteq \overline{\mathcal{L}} \text{ ?}$$

Overall Algorithm: Animation (V)



Overall Algorithm: Animation (VI)

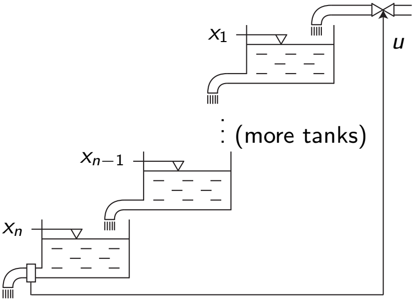


possible trajectories

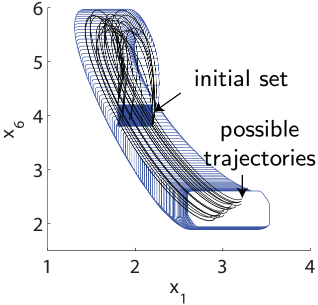
$\mathcal{R}([0, t_f])$

reachable set of the complete time horizon t_f

Scalability of the Linearization Approach



Water tank system.

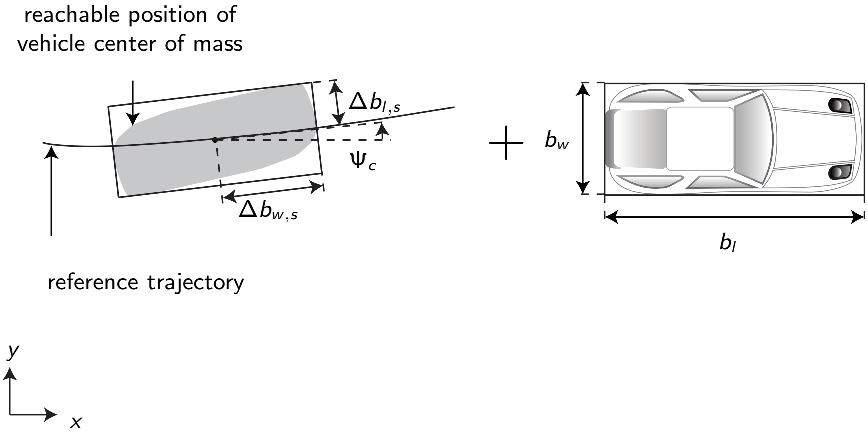


Projected reachable set ($n = 6$).

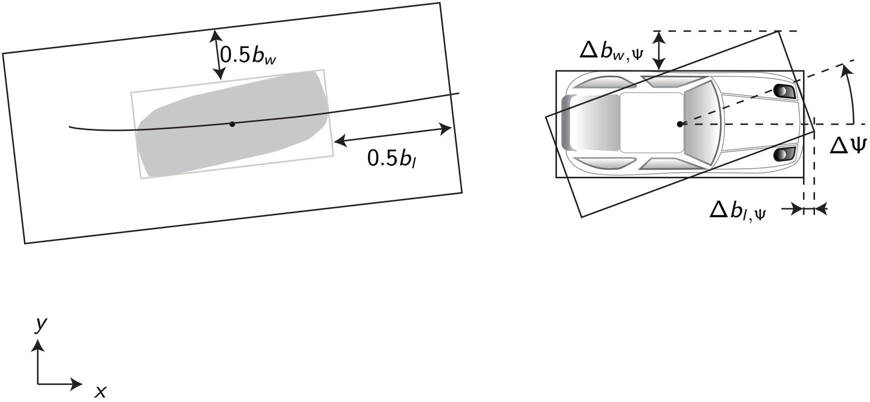
Complexity with respect to the number of continuous state variables n: $\mathcal{O}(n^3)$.

Dimension n	5	10	20	50	100
CPU-time [sec]	1.19	1.73	3.11	11.59	35.78

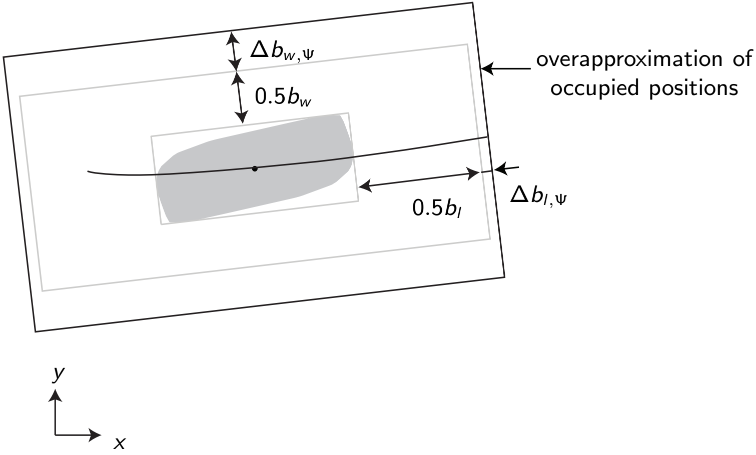
Occupied Positions: Step 1



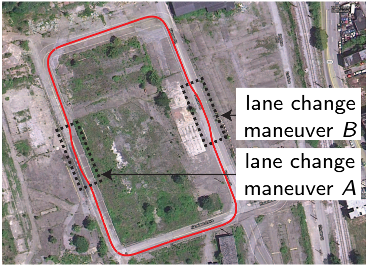
Occupied Positions: Step 2



Occupied Positions: Step 3

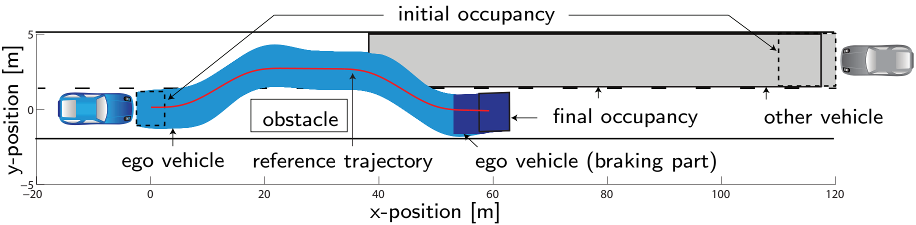


Online Verification Of Automated Driving

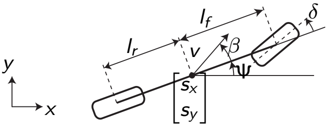


Test site

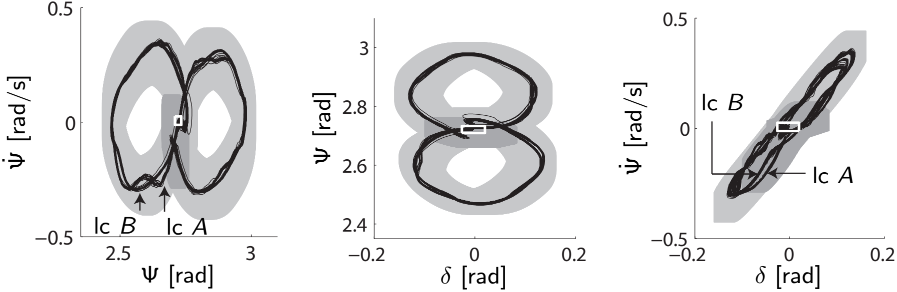
Test vehicle



Test Drive Results



s_x, s_y [m]	x- and y-position
Ψ [rad]	orientation
β [rad]	slip angle at center of mass
δ [rad]	front wheel angle
v [m/s]	velocity



computation time: $\approx$ 1.8 times faster than maneuver time (Intel i7, 1.6GHz)

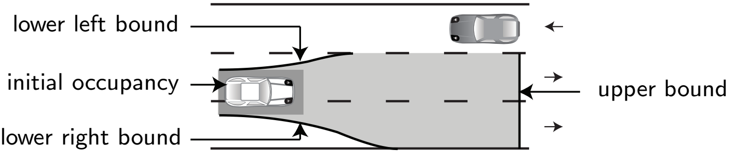
Abstraction Technique for Other Traffic Participants

Overapproximative Occupancy

Given are models M_i, $i = 1 \dots m$ which are abstractions of model M_0, i.e., $\mathrm{reach}(M_0) \subseteq \mathrm{reach}(M_i)$. The occupancy of the model M_0 can be overapproximated by

$$\mathrm{proj}\big(\mathrm{reach}(M_0)\big) \subseteq \bigcap_{i=1}^{m} \mathrm{proj}\big(\mathrm{reach}(M_i)\big). \qquad \square$$

Two models: Longitudinal dynamics along road boundaries (upper bound), lateral dynamics towards road boundaries (left/right bound).



Occupancy Along Road Boundaries

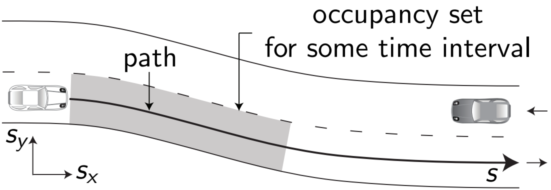
The dynamics becomes monotone when following a lane center.

Definition (Monotone dynamics)

For the initial state $x(0) \in \mathcal{R}(0)$ and inputs $u(t) \in \mathcal{U}$ the dynamics is monotone when the following holds for the solution $\chi(t, x(0), u(\cdot))$:

$$\text{if } \forall i, j, t \geq 0 : x_i(0) \leq \bar{x}_i(0), \quad u_j(t) \leq \bar{u}_j(t) \text{ then}$$
$$\forall i, t \geq 0 : \chi_i(t, x(0), u(\cdot)) \leq \chi_i(t, \bar{x}(0), \bar{u}(\cdot)). \qquad \square$$

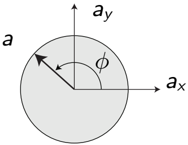
From this follows that e.g. the upper bound is provided by max. position, max. velocity, and max. acceleration:

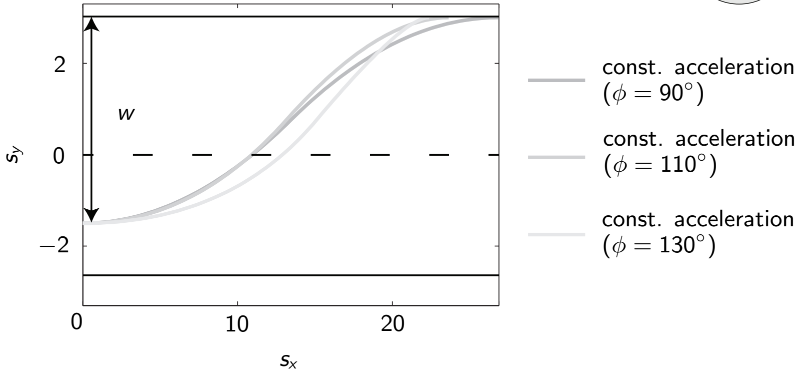


Occupancy Towards Road Boundaries

For lateral dynamics there exists no single combination of an initial state and an input trajectory determining the boundary.

Given the vehicle-fixed angle of the acceleration vector a, possible trajectories are:





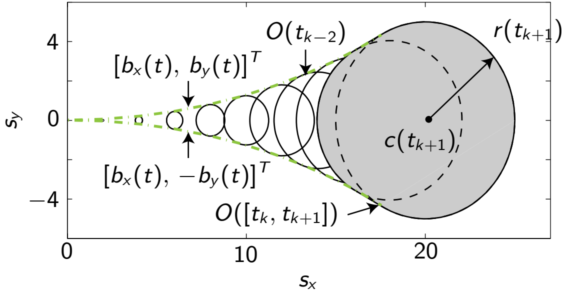
Occupancy Towards Road Boundaries: Method A

Using limit of absolute acceleration (constraint $C4$): Occupancies are circles with center $c(t)$ and radius $r(t)$:

$$c(t) = \begin{bmatrix} s_x(0) \\ s_y(0) \end{bmatrix} + \begin{bmatrix} v_x(0) \\ v_y(0) \end{bmatrix} t, \quad r(t) = \frac{1}{2} a_{\max} t^2.$$

From this follows the boundary of occupation:

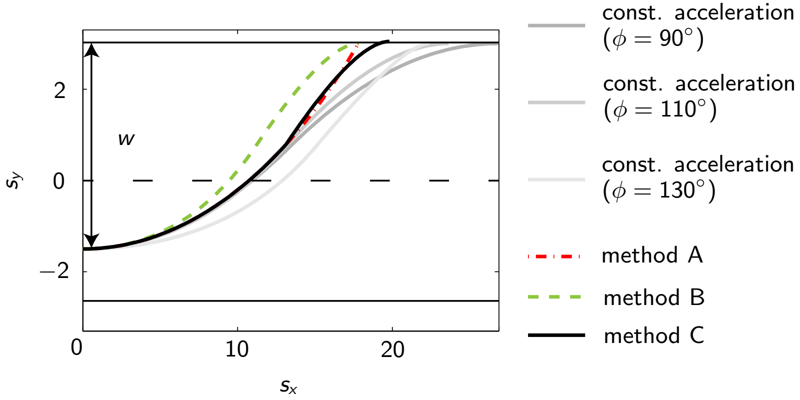
$$b_x(t) = v_0 t - \frac{a_{\max}^2 t^3}{2 v_0}, \quad b_y(t) = \sqrt{\frac{1}{4} a_{\max}^2 t^4 - \left(\frac{a_{\max}^2 t^3}{2 v_0} \right)^2}.$$



Occupancy Towards Road Boundaries: Method B and C

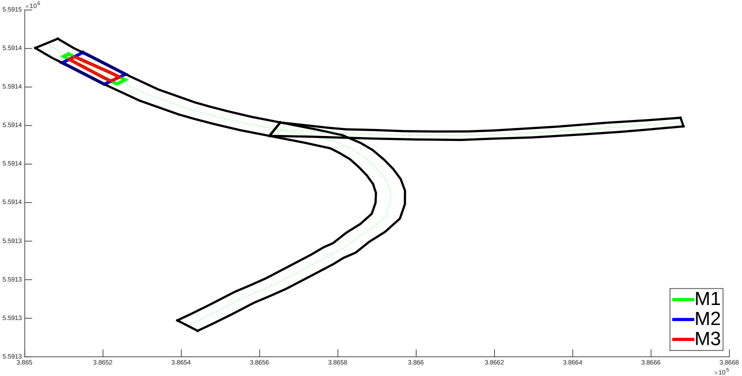
Method B: Assume independence of lateral and longitudinal acceleration $\rightarrow$ analytical solution.

Method C: Combination of method A and B.



Examples: Lane 1

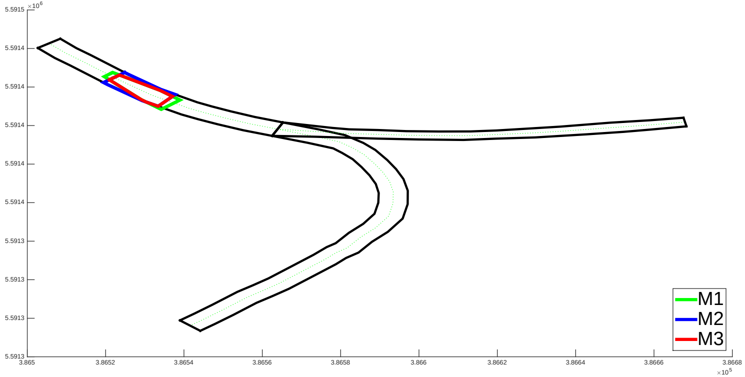
Step 1:



- M1: restricted absolute acceleration.
- M2: restricted acceleration and velocity in longitudinal direction.
- M3: staying within road boundaries.

Examples: Lane 1

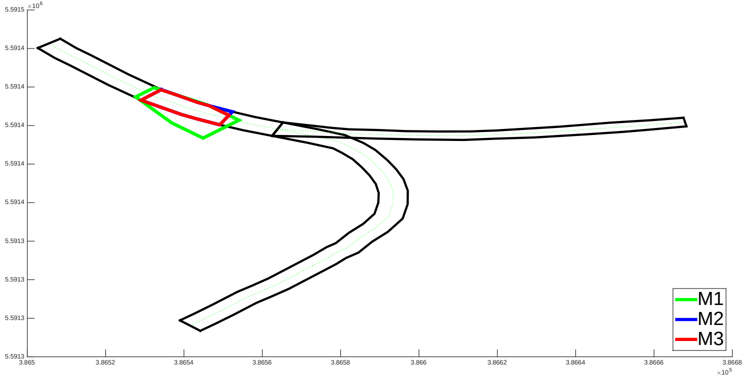
Step 2:



- M1: restricted absolute acceleration.
- M2: restricted acceleration and velocity in longitudinal direction.
- M3: staying within road boundaries.

Examples: Lane 1

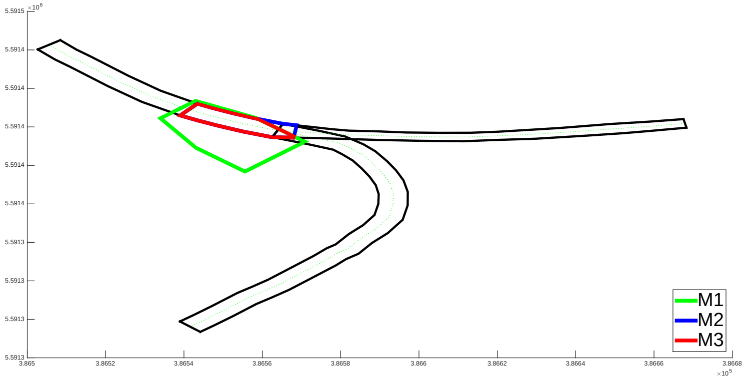
Step 3:



- M1: restricted absolute acceleration.
- M2: restricted acceleration and velocity in longitudinal direction.
- M3: staying within road boundaries.

Examples: Lane 1

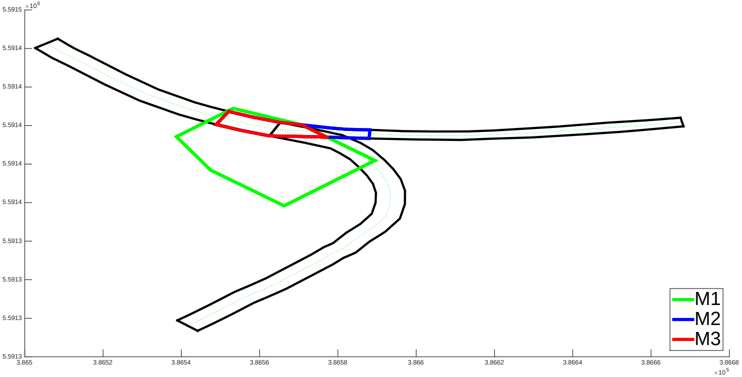
Step 4:



- M1: restricted absolute acceleration.
- M2: restricted acceleration and velocity in longitudinal direction.
- M3: staying within road boundaries.

Examples: Lane 1

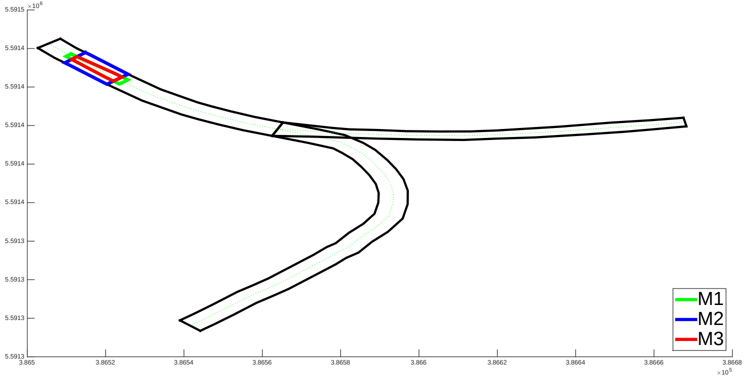
Step 5:



- M1: restricted absolute acceleration.
- M2: restricted acceleration and velocity in longitudinal direction.
- M3: staying within road boundaries.

Examples: Lane 2

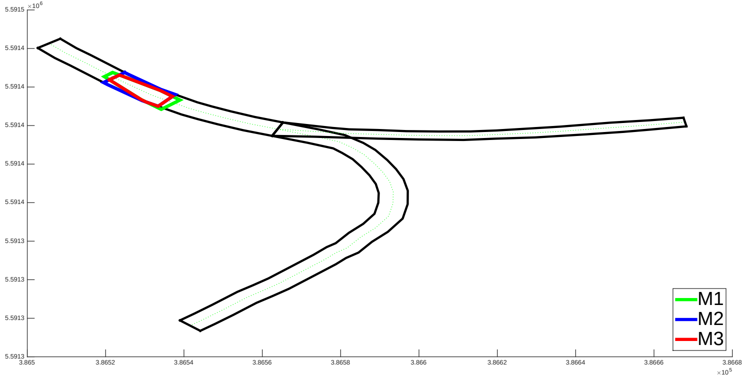
Step 1:



- M1: restricted absolute acceleration.
- M2: restricted acceleration and velocity in longitudinal direction.
- M3: staying within road boundaries.

Examples: Lane 2

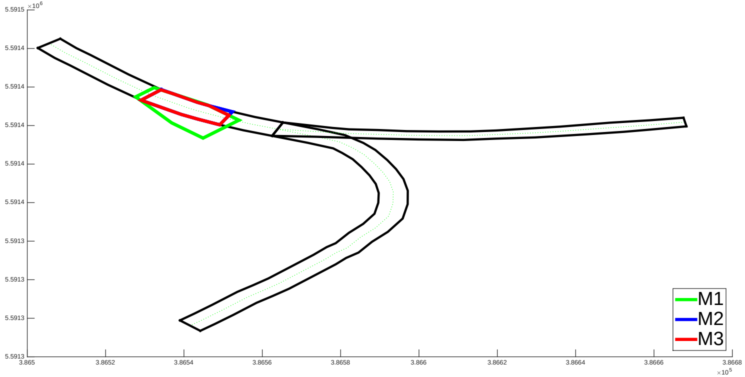
Step 2:



- M1: restricted absolute acceleration.
- M2: restricted acceleration and velocity in longitudinal direction.
- M3: staying within road boundaries.

Examples: Lane 2

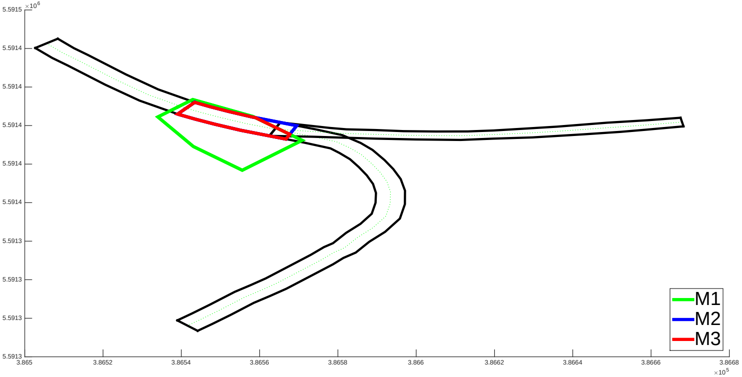
Step 3:



- M1: restricted absolute acceleration.
- M2: restricted acceleration and velocity in longitudinal direction.
- M3: staying within road boundaries.

Examples: Lane 2

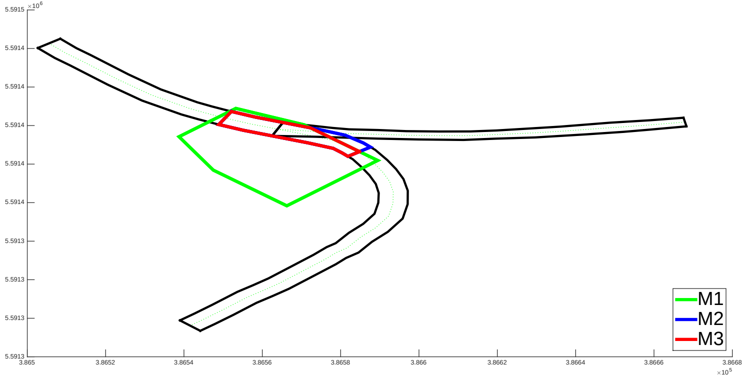
Step 4:



- M1: restricted absolute acceleration.
- M2: restricted acceleration and velocity in longitudinal direction.
- M3: staying within road boundaries.

Examples: Lane 2

Step 5:



- M1: restricted absolute acceleration.
- M2: restricted acceleration and velocity in longitudinal direction.
- M3: staying within road boundaries.

Methods for High-Order Models

Assumption of the previous example: Dynamics exactly described by a bicycle model.

Abstraction Challenges

- How large is the error between the bicycle model and a real vehicle/ high-fidelity model?

Methods for High-Order Models

Assumption of the previous example: Dynamics exactly described by a bicycle model.

Abstraction Challenges

- How large is the error between the bicycle model and a real vehicle/ high-fidelity model?
- Can one verify the real vehicle by a bicycle model plus uncertainty?

Methods for High-Order Models

Assumption of the previous example: Dynamics exactly described by a bicycle model.

Abstraction Challenges

- How large is the error between the bicycle model and a real vehicle/ high-fidelity model?
- Can one verify the real vehicle by a bicycle model plus uncertainty?
- What if parameters of the real vehicle are uncertain?

Methods for High-Order Models

Assumption of the previous example: Dynamics exactly described by a bicycle model.

Abstraction Challenges

- How large is the error between the bicycle model and a real vehicle/ high-fidelity model?
- Can one verify the real vehicle by a bicycle model plus uncertainty?
- What if parameters of the real vehicle are uncertain?

Error Bounding Challenge

Formally bounding $x(t) - z(t)$ (x: bicycle model, z: high-order model) is as hard as verifying the high-order model.

Methods for High-Order Models

Assumption of the previous example: Dynamics exactly described by a bicycle model.

Abstraction Challenges

- How large is the error between the bicycle model and a real vehicle/ high-fidelity model?
- Can one verify the real vehicle by a bicycle model plus uncertainty?
- What if parameters of the real vehicle are uncertain?

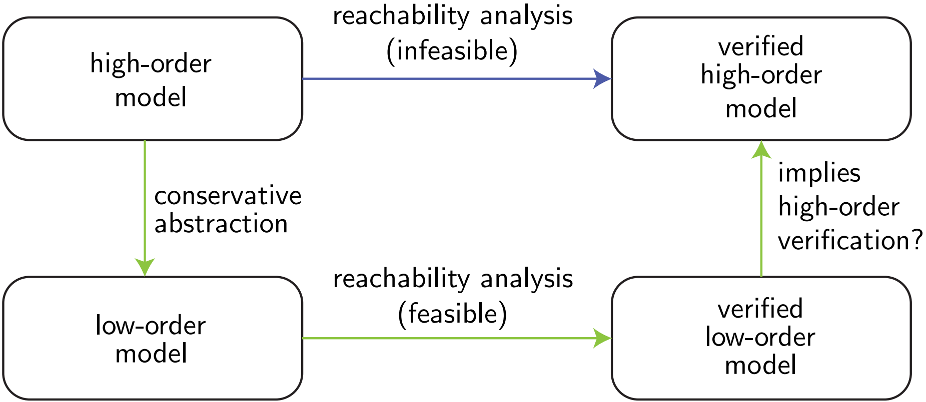
Error Bounding Challenge

Formally bounding $x(t) - z(t)$ (x: bicycle model, z: high-order model) is as hard as verifying the high-order model.

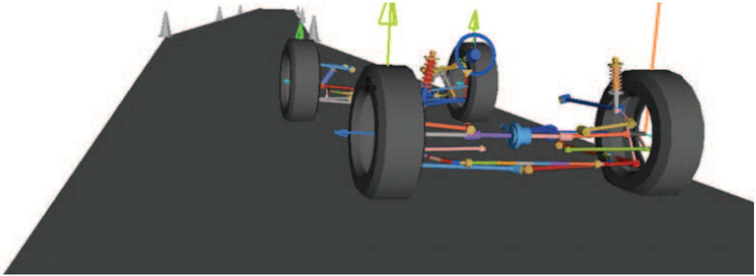
$\rightarrow$ Falsification of the low order model.

High-Order Model Verification Using Low-Order Models

Conservative abstraction (include uncertainties):



High-Order Vehicle Model



Source: `www.bremarauto.com`

Features:

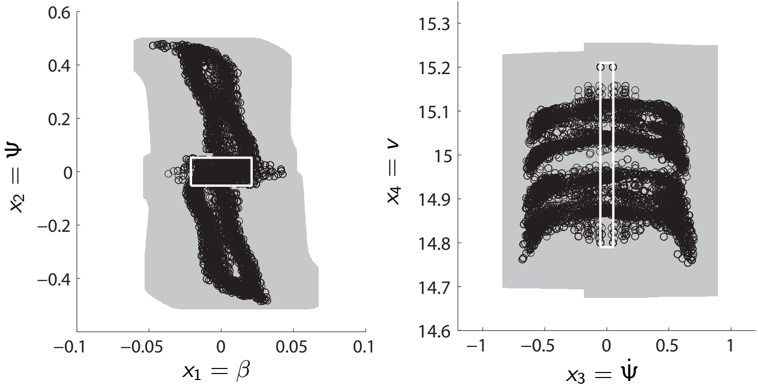
- Multi-body dynamics (28 state variables)
- Individual tire spin, slip, and camber angle.
- Nonlinear tire dynamics according to PAC2002 Magic-Formula.
- Suspension forces from springs, dampers, and anti-roll bars.

Falsification of Low-Order Models

Choose a finite number of maneuvers. For each maneuver:

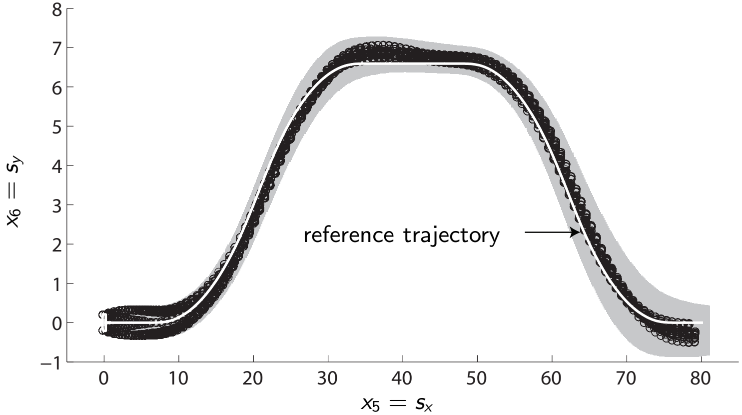
1. Compute the reachable set of the bicycle model using
$\dot{x} = f(x(t), x_d(t), u(t)) + v(t)$,
(x: state, x_d: desired state, u: sensor noise, v: additional disturbance).
2. Use rapidly-exploring random trees (RRTs) to guide the simulation of the high-fidelity model outside the reachable set.
3. In case of a violation, increase the uncertain input set $v(t) \in \mathcal{V}$ and go back to step 1.

Double Lane Change Test (1)



- The white set shows the set of initial states.
- Black circles show RRT nodes of the high-dimensional model.
- Gray area shows reachable set of the low-dimensional model.

Double Lane Change Test (2)



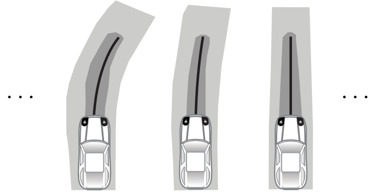
- The white set shows the set of initial states.
- Black circles show RRT nodes of the high-dimensional model.
- Gray area shows reachable set of the low-dimensional model.

Connecting Reachability Results of Motion Primitives

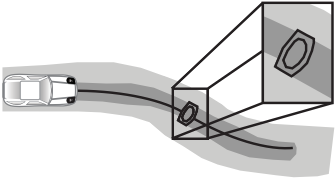
Basic idea: Pre-compute reachable sets of motion primitives.

Connection constraint: Results can be combined when final set is a subset of the initial set of the next motion primitive.

Result: Computational effort is shifted towards offline computation.



Reachable set and occupancy for all motion primitives



Connectivity check using reachability analysis

Motion Primitives

We use linear models for velocity v and curvature κ:

$$v^*(t) = p_1 + p_3\, t$$
$$\kappa^*(t) = p_2 + p_4\, t.$$

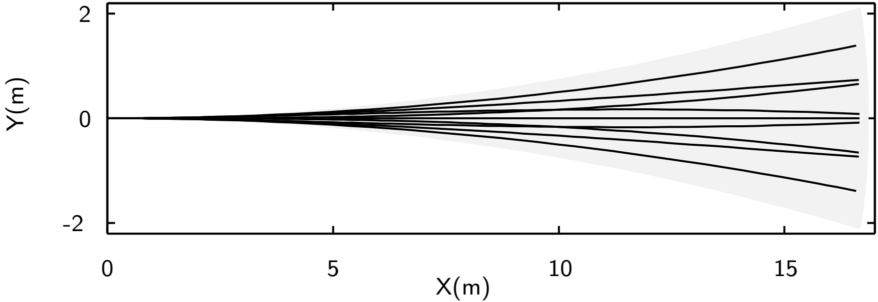
Other states are obtained using a unicycle model:

$$\dot{X}^* = \cos(\theta^*)\, v^*(t)$$
$$\dot{Y}^* = \sin(\theta^*)\, v^*(t)$$
$$\dot{\theta}^* = v^*(t)\, \kappa^*(t)$$

This choice results in spiral trajectories, specifically Euler spirals for the case of $\dot{v}^* = 0$.

Uncertain Motion Primitives

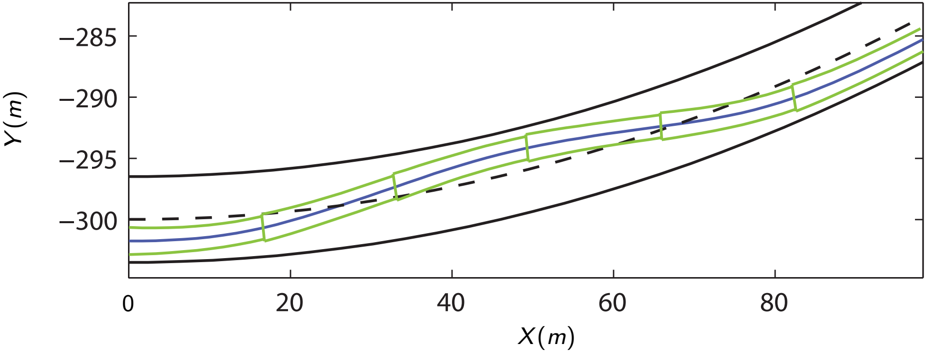
We allow uncertainties for each parameter ($p_i \in [\underline{p}_i, \overline{p}_i]$) so that we can represent infinitely many motion primitives (here: 9 nominal motion primitives):



Results of the pre-computed reachable sets are valid for any motion primitive, where $\forall i: \quad p_i \in [\underline{p}_i, \overline{p}_i]$

Example for Connecting Motion Primitives

- Motion primitives are connected according to the connection constraint.
- The reachability analysis is performed in almost no time due to the precomputation.



Interaction between Planning and Verification

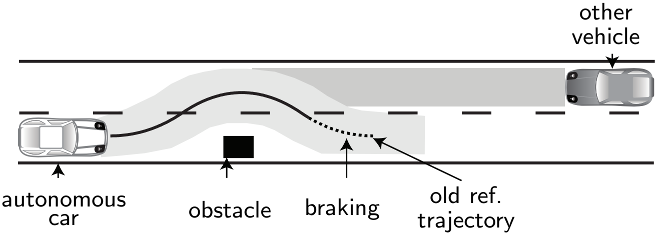
Planning Challenges

- What if the reference trajectory is unsafe?
- Is there enough time to re-plan and verify a new trajectory?
- What if a software bug or hardware failure occurs?

Planning Solution

- Plan maneuvers that are safe for all times (details later).
- Only change the previous plan if the new plan has already been verified.

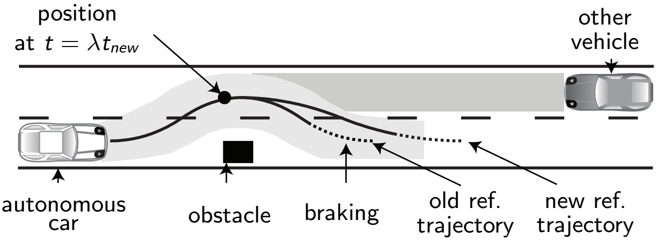
Maneuvers Verified for all Times



General Idea

- Add a braking maneuver to the end of the originally intended maneuver.
- The vehicle has to stop in a safe location (e.g. not on a railway crossing).
- The additional braking trajectory is only executed when no new safe plan is ready for execution.

Deviation from Previous Plan



General Idea

- Change maneuver only at a point from where the new reference trajectory has been verified.
- Verification time is linear in the time horizon t_f: $t_{ver} = \lambda t_f$.

$\rightarrow$ Change previous plan at $\lambda t_{f,\mathrm{new}}$.

Conclusions

- All possible behaviors can be computed.

Conclusions

- All possible behaviors can be computed.
- We consider uncertain initial states, disturbances and parameters.

Conclusions

- All possible behaviors can be computed.
- We consider uncertain initial states, disturbances and parameters.
- Assuming that the mathematical model captures all real behaviors (by adding uncertainty), we can prove safety.

Conclusions

- All possible behaviors can be computed.
- We consider uncertain initial states, disturbances and parameters.
- Assuming that the mathematical model captures all real behaviors (by adding uncertainty), we can prove safety.
- RRTs can be used to determine additional uncertainty for capturing the real behavior.

Conclusions

- All possible behaviors can be computed.
- We consider uncertain initial states, disturbances and parameters.
- Assuming that the mathematical model captures all real behaviors (by adding uncertainty), we can prove safety.
- RRTs can be used to determine additional uncertainty for capturing the real behavior.
- The computation is faster than the maneuver time.

Conclusions

- All possible behaviors can be computed.
- We consider uncertain initial states, disturbances and parameters.
- Assuming that the mathematical model captures all real behaviors (by adding uncertainty), we can prove safety.
- RRTs can be used to determine additional uncertainty for capturing the real behavior.
- The computation is faster than the maneuver time.
- Computation time can be drastically reduced with pre-computation techniques.

Conclusions

- All possible behaviors can be computed.
- We consider uncertain initial states, disturbances and parameters.
- Assuming that the mathematical model captures all real behaviors (by adding uncertainty), we can prove safety.
- RRTs can be used to determine additional uncertainty for capturing the real behavior.
- The computation is faster than the maneuver time.
- Computation time can be drastically reduced with pre-computation techniques.
- Implementation in C++ for deployment in the car.

Conclusions

- All possible behaviors can be computed.
- We consider uncertain initial states, disturbances and parameters.
- Assuming that the mathematical model captures all real behaviors (by adding uncertainty), we can prove safety.
- RRTs can be used to determine additional uncertainty for capturing the real behavior.
- The computation is faster than the maneuver time.
- Computation time can be drastically reduced with pre-computation techniques.
- Implementation in C++ for deployment in the car.
- Comparison of results with real world measurements.