Computing Interaction-aware Reachable sets of Automated Vehicles using Monte Carlo Tree Search

Xin Zhang
Final Presentation Master's Thesis
Supervisor: M.Sc. Edmond Irani Liu, M.Sc. Tommaso Benciolini

Chair of Automatic Control Engineering
Technical University of Munich

Motivation

Problem

Major challenges arise **in dense, dynamic situations** where the automated vehicles should **interact** with human-driven vehicles in motion planning.

Idea

- Reachable Sets $\longrightarrow$ Motion planning in dense, dynamic scenarios without interaction
- Monte Carlo Tree Search (MCTS) $\longrightarrow$ Interaction between multi agents

Goal

The project proposes to obtain interaction-aware reachable sets through MCTS. Interaction-aware reachable sets can express the driving strategy considering interaction with human-driven vehicles.

Motivation

Problem

Major challenges arise **in dense, dynamic situations** where the automated vehicles should **interact** with human-driven vehicles in motion planning.

Idea

- Reachable Sets $\longrightarrow$ Motion planning in dense, dynamic scenarios without interaction
- Monte Carlo Tree Search (MCTS) $\longrightarrow$ Interaction between multi agents

Goal

The project proposes to obtain interaction-aware reachable sets through MCTS. Interaction-aware reachable sets can express the driving strategy considering interaction with human-driven vehicles.

Motivation

Problem

Major challenges arise **in dense, dynamic situations** where the automated vehicles should **interact** with human-driven vehicles in motion planning.

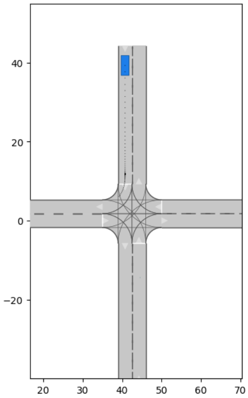
Idea

- Reachable Sets $\longrightarrow$ Motion planning in dense, dynamic scenarios without interaction
- Monte Carlo Tree Search (MCTS) $\longrightarrow$ Interaction between multi agents

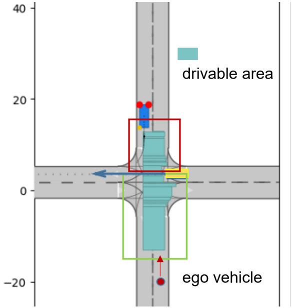
Goal

The project proposes to obtain interaction-aware reachable sets through MCTS. Interaction-aware reachable sets can express the driving strategy considering interaction with human-driven vehicles.

Motivation



Example for original reachable sets



Reachable set at $2s$

Ego vehicle passes through intersection before yellow vehicle.

Ego vehicle passes through intersection after yellow vehicle.

Related Work

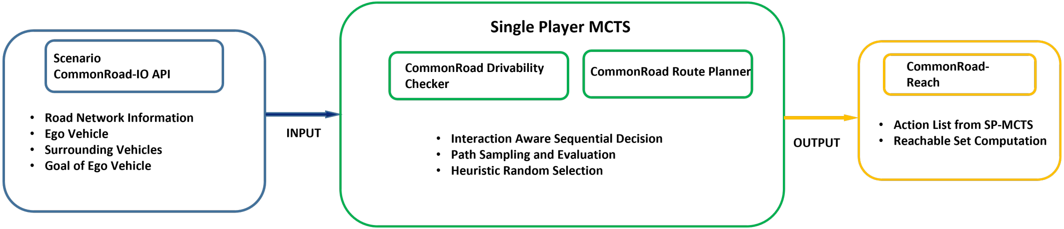
Monte Carlo Tree Search

- Tactical cooperative planning for autonomous highway driving using Monte-Carlo tree search [Lenz+ 2016]
- A survey of monte carlo tree search methods [Browne+ 2012]

Over-approximative Reachable Sets

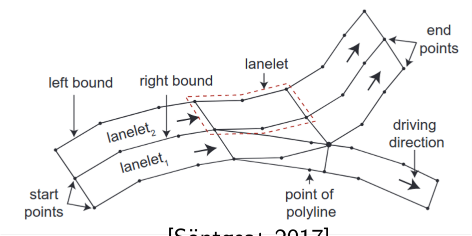
- CommonRoad-Reach: A Toolbox for Reachability Analysis of Automated Vehicles [Liu+ 2022]
- Computing the drivable area of autonomous road vehicles in dynamic road scenes [Söntges+ 2017]

Overview of Algorithm



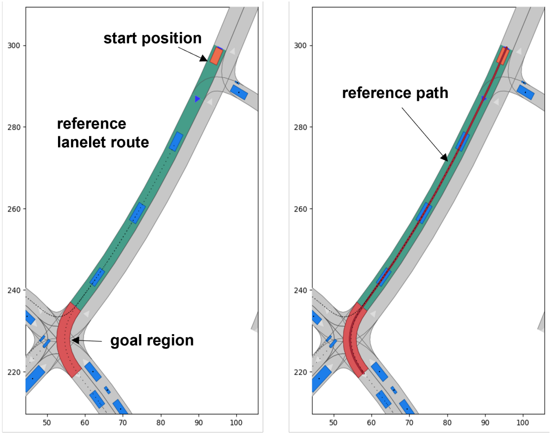
Framework for computation of interaction-aware reachable sets

Scenario: Road Network



[Söntges+ 2017]

Road Network with Lanelet



Route Planner

Scenario: Vehicle Model

System State Space Equation

$$\begin{pmatrix} \dot{x} \\ \ddot{x} \\ \dot{y} \\ \ddot{y} \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} x \\ \dot{x} \\ y \\ \dot{y} \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} u_x \\ u_y \end{pmatrix}$$

$$\begin{aligned} |u_x| &\leq a_{max,x} \\ |u_y| &\leq a_{max,y} \\ v_{min,x} \leq \dot{x} &\leq v_{max,x} \\ v_{min,y} \leq \dot{y} &\leq v_{max,y} \end{aligned}$$

x, y: Position
v: Velocity
a: Acceleration
u: Input

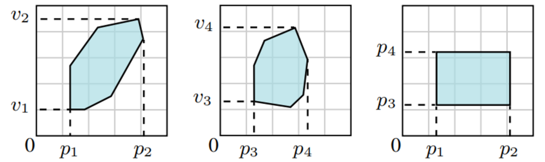
Reachable Sets

Definition of Reachable Set

The reachable set is the set of collision free states that can be reached over time starting from an initial set of states.

$$\mathcal{R}\left(t;\mathcal{X}_0\right)=\{s(t;u,s_0)\mid \exists u\in\mathcal{U},\exists s_0\in\mathcal{X}_0,\\ s\left(\tau;u,s_0\right)\notin\mathcal{F}\left(\tau\right)\,for\ \tau\in[t_0,t]\}$$

Base Set $\mathcal{R}_t^{(i)}=\mathcal{P}_{x,t}^{(i)}\times\mathcal{P}_{y,t}^{(i)}\Leftarrow f\left(a_{max,x},a_{min,x}\right)\times f\left(a_{max,y},a_{min,y}\right)$



$\mathcal{P}_{x,t}^{(i)}$ $\mathcal{P}_{y,t}^{(i)}$ $\mathcal{D}_t^{(i)}$

- s: State of trajectory
- s_o: Initial state
- $\mathcal{X}_0$: Set of initial state
- $\mathcal{F}$: Set of forbidden state
- $\mathcal{P}_{x,t}^{(i)}$: Polytope in x-direction
- $\mathcal{P}_{y,t}^{(i)}$: Polytope in y-direction
- $\mathcal{D}_t^{(i)}$: Drivable area

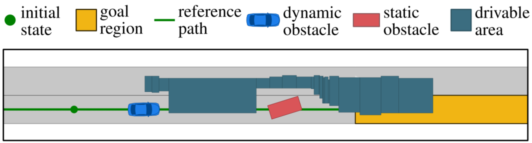
Reachable Sets

Definition of Reachable Set

The reachable set is the set of collision free states that can be reached over time starting from an initial set of states.

$$\mathcal{R}\left(t;\mathcal{X}_0\right)=\{s(t;u,s_0)\mid \exists u\in\mathcal{U},\exists s_0\in\mathcal{X}_0,$$
$$s\left(\tau;u,s_0\right)\notin\mathcal{F}\left(\tau\right) for\ \tau\in[t_0,t]\}$$

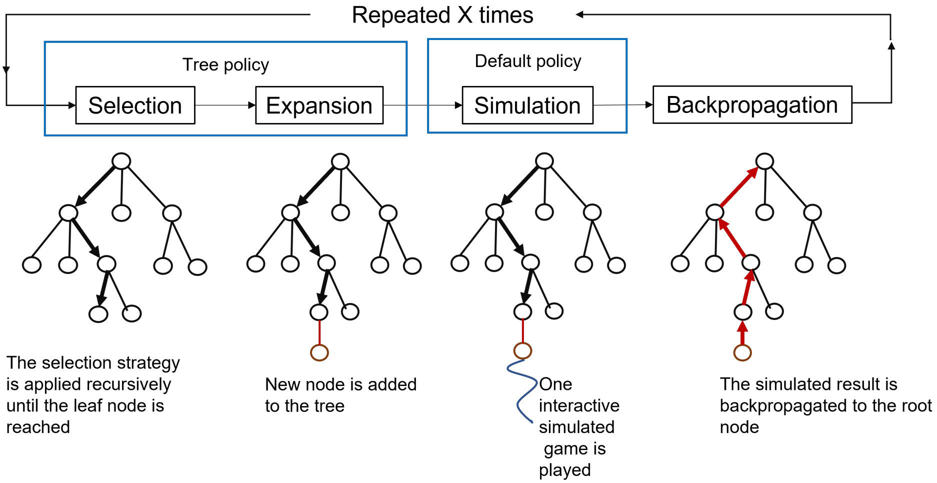
Base Set $\mathcal{R}_t^{(i)}=\mathcal{P}_{x,t}^{(i)}\times\mathcal{P}_{y,t}^{(i)}\Leftarrow f\left(a_{max,x},a_{min,x}\right)\times f\left(a_{max,y},a_{min,y}\right)$



A collision-free drivable area at time 2.7s [Liu+ 2022]

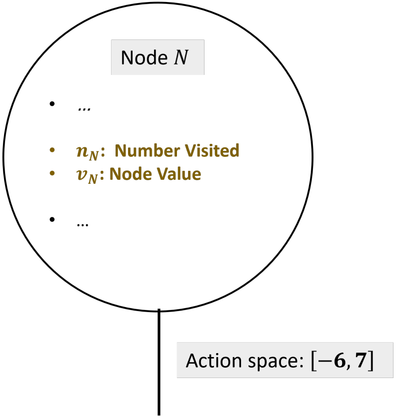
s: State of trajectory
s_o: Initial state
$\mathcal{X}_0$: Set of initial state
$\mathcal{F}$: Set of forbidden state
$\mathcal{P}_{x,t}^{(i)}$: Polytope in x-direction
$\mathcal{P}_{y,t}^{(i)}$: Polytope in y-direction
$\mathcal{D}_t^{(i)}$: Drivable area

Single-Player Monte Carlo Tree Search



Single-Player Monte Carlo Tree Search

MCTS Component: Node and Action Space



Action Definition

Action	Behavior	Method 1	Method 2
a_1	Braking	$[-6, -3]$	$[-6, -3]$
a_2	Deceleration	$[-3, -1]$	$[-4, -1]$
a_3	Constant speed	$[-1, 1]$	$[-2, 2]$
a_4	Acceleration	$[1, 4]$	$[1, 5]$
a_5	Rapid acceleration	$[4, 7]$	$[4, 7]$

$$\mathcal{A} = \{a_1, a_2, a_3, a_4, a_5\}$$

Tree Policy

Selection

- Control balance between exploitation and exploration
  - *Exploitation*: Select the best move
  - *Exploration*: Try the less promising moves

$$i = arg \max_{i \in \mathcal{I}_N} \left(\frac{v_i}{n_i} + C \sqrt{\frac{\ln n}{n_i}} + \sqrt{\frac{\sum v \; - n_i \, \bar{X}^2 + D}{n_i}} \right)$$

$\mathcal{I}_N$: Set of child nodes of node N

i: Child node

v_i: Value of node i

n_i: Number visited of node i

$\bar{X}$: Average value of node i

n: Number visited of node N

C, D: Constant positive value

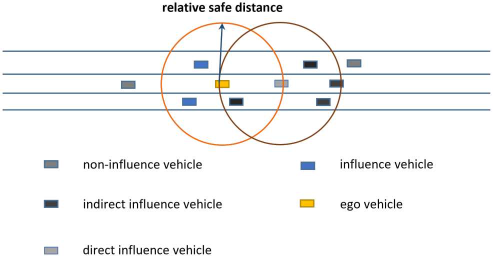
Expansion

- One node is added from untried actions per each expansion

Default Policy

In simulation stage, the ego vehicle will move under interaction with surrounding vehicles

Classification of Vehicles

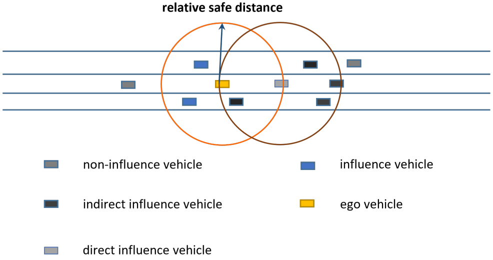


Relative safe distance $d_{rs} = \frac{v\ km/h}{2}$

Default Policy

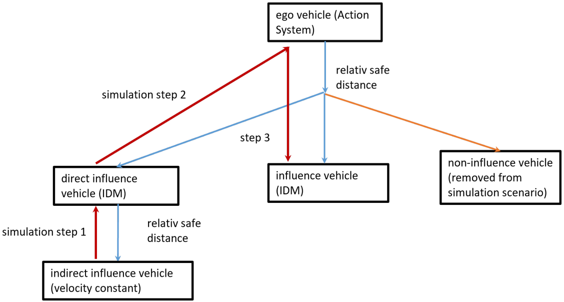
In simulation stage, the ego vehicle will move under interaction with surrounding vehicles

Classification of Vehicles



Relative safe distance $d_{rs} = \frac{v\ km/h}{2}$

Sequential Decision Chain



Default Policy

v_o: Desired velocity
δ: Free acceleration exponent
a: Maximum acceleration
b: Comfortable deceleration
T: Desired time gap
s_0: Minimum desired safe distance
Δv_α: Approaching rate

Car-Following Model: Intelligent Drive Model

$$a_{IDM} = \dot{v}_\alpha^{free} + \dot{v}_\alpha^{int}$$

$$\dot{v}_\alpha^{free} = a(1 - \left(\frac{v_\alpha}{v_0}\right)^\delta)$$

$$\dot{v}_\alpha^{int} = -a\left(\frac{s^*(v_\alpha, \Delta v_\alpha)}{s_\alpha}\right)^2 \; with \; s^*(v_\alpha, \Delta v_\alpha) = s_0 + v_\alpha T + \frac{v_\alpha \Delta v_\alpha}{2\sqrt{ab}}$$

Action System: Heuristic Random Selection

More simulation ← Search

Unrealistic ← Stochastic

Knowledge → Expensive computation

Deterministic → Too selective

Default Policy: Heuristic Random Selection

Road Information

If there is only ego vehicle on road, the velocity of ego vehicle v_e should refer to the desired velocity v_d.

Probability of action being selected

	$v_e > 1.5v_d$	$1.1v_d \leq v_e < 1.5v_d$	$0.9v_d \leq v_e < 1.1v_d$	$0.5v_d \leq v_e < 0.9v_d$	$v_e \leq 0.5v_d$
a_1	0.3	0.25	0.1	0.15	0.15
a_2	0.25	0.3	0.25	0.15	0.15
a_3	0.15	0.15	0.3	0.15	0.15
a_4	0.15	0.15	0.25	0.3	0.25
a_5	0.15	0.15	0.1	0.25	0.3

Default Policy: Heuristic Random Selection

Surrounding Vehicle

Carry out risk assessment R for each action [Glaser+ 2010] and rank them to determine the probability of being selected.

Time to collision

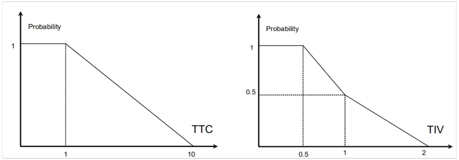
$$TTC = \frac{D_{ef}}{v_e - v_f}$$

Inter vehicular time

$$TIV = \frac{D_{ef}}{v_e}$$

Risk

$$R = P_{TTC}(v_e)(v_f - v_e) + P_{TIV}(v_e)max(v_f - v_e, v_f - \lambda v_f TIV - v_e)$$



TTC (left) and TIV (right) based probability of collision [Glaser+ 2010]

D_{ef}: Distance between ego vehicle and leader vehicle

v_e: Velocity of ego vehicle

v_f: Velocity of leader vehicle

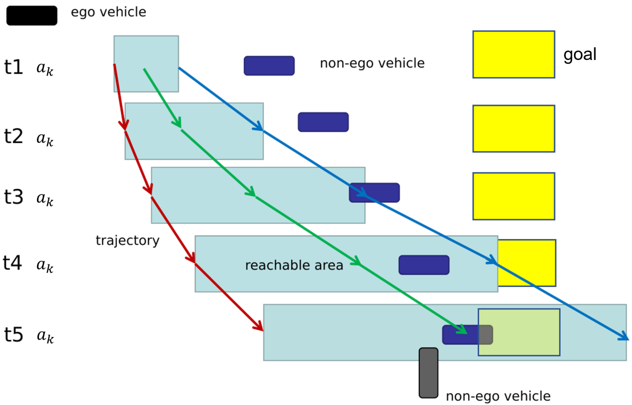
λ: Considered deceleration

Risk Order: $[1, 2, 3, 4, 5]$

Probability: $[\frac{1}{15}, \frac{2}{15}, \frac{1}{5}, \frac{4}{15}, \frac{1}{3}]$

Default Policy: Evaluation

Evaluation via Sampling Paths



Default Policy: Evaluation

Evaluation for one Sampling Path

- $\boldsymbol{j_r} = \sum \|\boldsymbol{p_i} - \boldsymbol{p_{r,i}}\|$: Cost for difference between simulated and reference path
- $\boldsymbol{j_v} = \|\boldsymbol{v_i} - \boldsymbol{v_d}\|$: Cost for difference between current velocity and expected velocity
- $\boldsymbol{j_d} = \sum \frac{1}{\boldsymbol{d_i}}$: Cost for distance to other vehicles
- $\boldsymbol{j_g} = \|\boldsymbol{p_i} - \boldsymbol{p_d}\|$: Cost for distance to the destination
- $\boldsymbol{j_t} = (\boldsymbol{t_N} - \boldsymbol{T_s})\boldsymbol{t_N}$: Cost for time
- $\boldsymbol{j_c}$: Cost for collision, if collision occurs

p_i: Position of simulated trajectory
$p_{r,i}$: Position of reference trajectory
d_i: Distance to other vehicles
p_d: Position of goal
t_N: Simulation time
T_s: Planning horizon time
w_i: Weight for evaluation item

$$\boldsymbol{J_{ego}} = \sum \boldsymbol{w_i j_i}$$

Default Policy: Evaluation

Evaluation for one Action

Collision Rate $\beta_c = \frac{K_1}{K}$

$$v_N = \begin{cases} \frac{\sum_{i \in (K-K_1)} -J_{ego,i}}{K - K_1}, & if\ \beta_c < \eta_c \\ \frac{\sum_{i \in K_1} -J_{ego,i}}{K_1}, & else \end{cases}$$

Reach Rate $\beta_r = \frac{K_2}{K}$

If $\beta_r < \eta_r$, simulation should **continue**;

else, simulation should be **stopped**

K: Number of sampling path

K_1: Number of sampling path where collision occurs

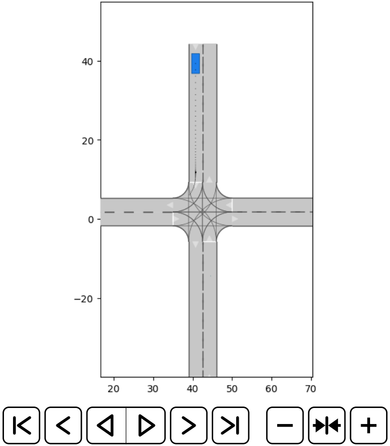
K_2: Number of sampling path to reach destination

η_c: Threshold of collision rate

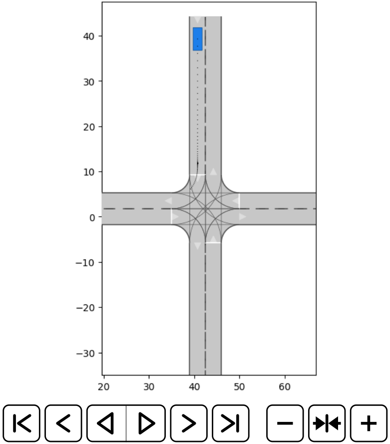
η_r: Threshold of reach rate

v_N: Node Value

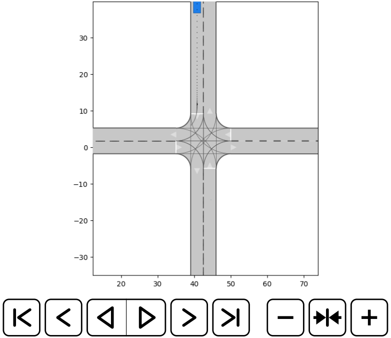
Result



Original reachable sets

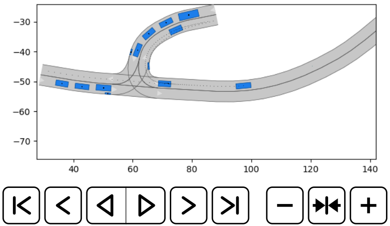


Interaction-aware reachable sets with $v_d = 15m/s$

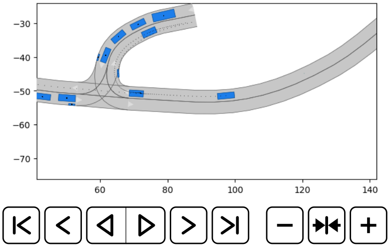


Interaction-aware reachable sets with $v_d = 6m/s$

Result



Interaction-aware reachable sets, desired velocity for direct influence vehicle is $15m/s$



Interaction-aware reachable sets, desired velocity for direct influence vehicle is $0m/s$

Conslusion

Conclusion

We successfully computed interaction-aware reachable sets using SP-MCTS. Interaction-aware reachable sets exhibit distinct driving strategies in the longitude direction.

Future Work

- Improve the sampling in simulation stage
- Try other vehicle behavior models
- Add actions in the lateral direction

References

Cameron B Browne, Edward Powley, Daniel Whitehouse, Simon M Lucas, Peter I Cowling, Philipp Rohlfshagen, Stephen Tavener, Diego Perez, Spyridon Samothrakis and Simon Colton. **A survey of monte carlo tree search methods**.
In: *IEEE Transactions on Computational Intelligence and AI in games* 4.1 (2012), pp. 1–43.

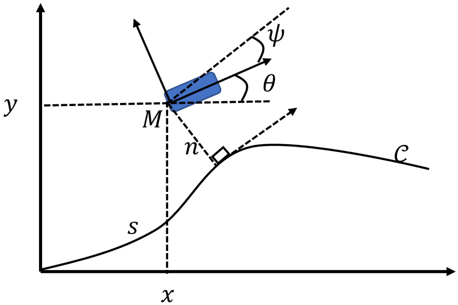
Sébastien Glaser, Benoit Vanholme, Saïd Mammar, Dominique Gruyer and Lydie Nouveliere.
Maneuver-based trajectory planning for highly autonomous vehicles on real road with traffic and driver interaction.
In: *IEEE Transactions on intelligent transportation systems* 11.3 (2010), pp. 589–606.

David Lenz, Tobias Kessler and Alois Knoll. **Tactical cooperative planning for autonomous highway driving using Monte-Carlo Tree Search**.
In: *2016 IEEE Intelligent Vehicles Symposium (IV)*. IEEE. 2016, pp. 447–453.

Edmond Irani Liu, Gerald Würsching, Moritz Klischat and Matthias Althoff.
CommonRoad-Reach: A Toolbox for Reachability Analysis of Automated Vehicles.
In: *2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE. 2022, pp. 2313–2320.

Sebastian Söntges and Matthias Althoff. **Computing the drivable area of autonomous road vehicles in dynamic road scenes**.
In: *IEEE Transactions on Intelligent Transportation Systems* 19.6 (2017), pp. 1855–1866.

Scenario: Curvilinear Coordinate System



x, y: Position in Cartesian coordinates

θ: Orientation (heading)

s: Curvilinear abscissa

n: Signed lateral distance

ψ: Relative orientation

$\mathcal{C}$: Reference trajectory (curve)

Cartesian coordinates $[x\ y\ \theta]^T$ → Curvilinear coordinates $[s\ n\ \psi]^T$

Reachable Sets

Reachable set in X-direction

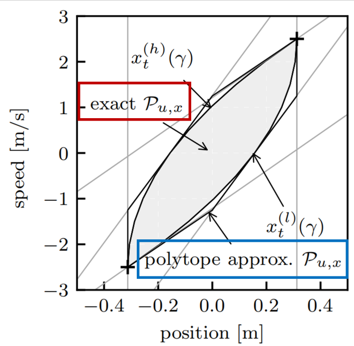
$$\begin{pmatrix}\dot{x}\\ \ddot{x}\end{pmatrix} = \mathrm{A}\begin{pmatrix}x\\ \dot{x}\end{pmatrix} + \mathrm{B}u_x, \text{ with } \mathrm{A} = \begin{pmatrix}0 & 1\\ 0 & 0\end{pmatrix} \text{ and } \mathrm{B} = \begin{pmatrix}0\\ 1\end{pmatrix}$$

$$|u_x| \leq a_{max,x}$$

Reformulated reachebable set:

$$\begin{aligned}\mathcal{R}(t;\mathcal{X}_0) &= \left\{s \middle| \exists u \in \mathcal{U}, s = e^{At}s_0 + \int_0^t e^{A(t-\tau)}Bu(\tau)\,d\tau\right\}\\ &= e^{At}s_0 \oplus \left\{s \middle| \exists u \in \mathcal{U}, s = \int_0^t e^{A(t-\tau)}Bu(\tau)\,d\tau\right\}\\ &= e^{At}s_0 \oplus \mathcal{P}_{u,x}(t) \text{ (}\oplus\text{: Minkowski sum)}\end{aligned}$$

approx. $\mathcal{P}_{u,x}(t)$



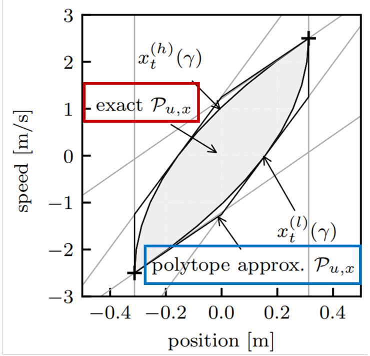
Reachable Sets

$approx.\ \mathcal{P}_{u,x}(t)$ for a specific time t is bounded by

Upper bound $\begin{cases} x_t^{(h)}(\gamma) = x_0 + \dot{x}_0 t + a_0 t^2(\frac{1}{2} - 2\gamma + \gamma^2) \\ \dot{x}_t^{(h)}(\gamma) = \dot{x}_0 + a_{max,x} t(1 - 2\gamma) \end{cases}$

Lower bound $\begin{cases} x_t^{(h)}(\gamma) = x_0 + \dot{x}_0 t - a_0 t^2(\frac{1}{2} - 2\gamma + \gamma^2) \\ \dot{x}_t^{(h)}(\gamma) = \dot{x}_0 - a_{max,x} t(1 - 2\gamma) \end{cases}$

via Pontryagin's Principle with switching time γt, $\gamma \in [0 \ldots 1]$



$approx.\ \mathcal{P}_{u,x}(t)$ depends on maximum und minimum acceleration.

γ = 0, 0.5, 1