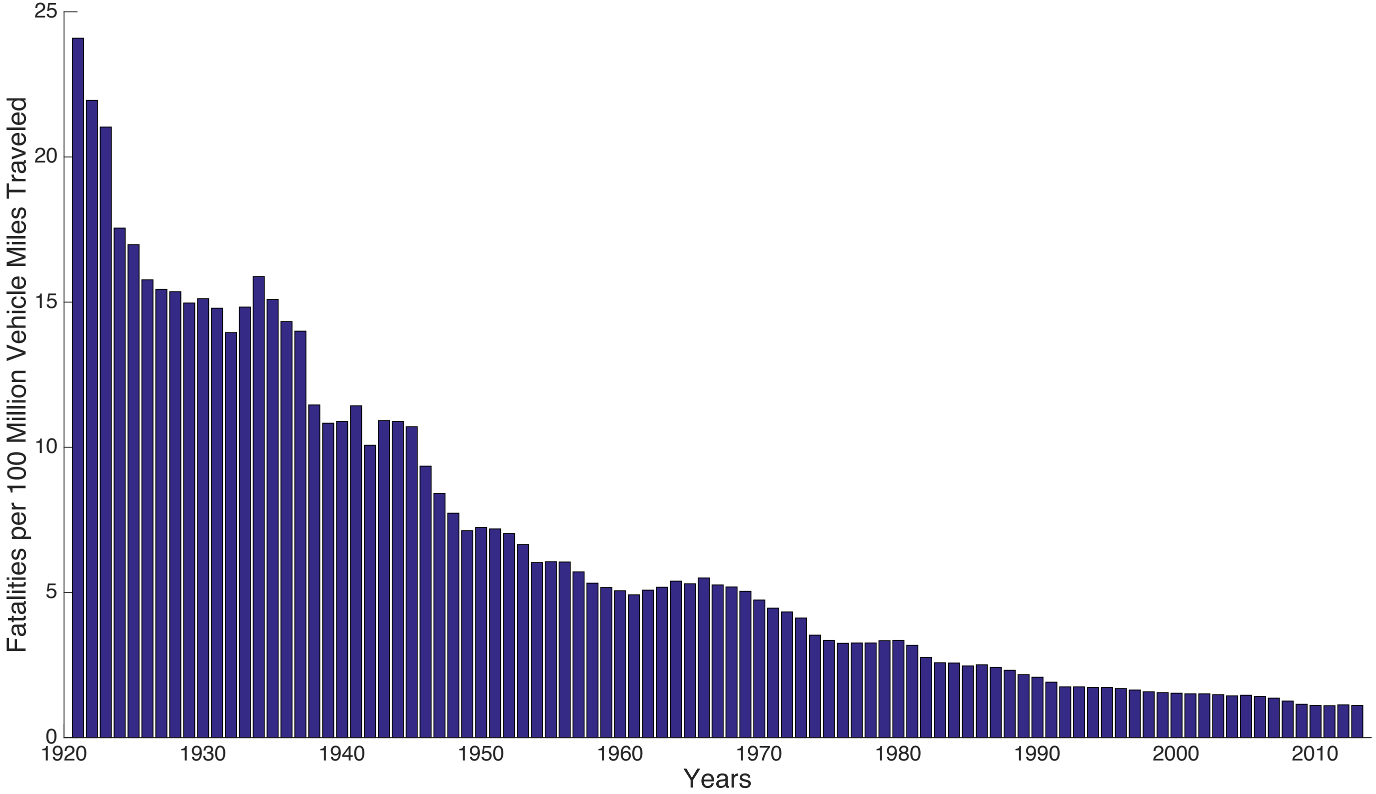
Bridging the Gap Between Safety and Real-Time Performance for Autonomous Vehicle Control

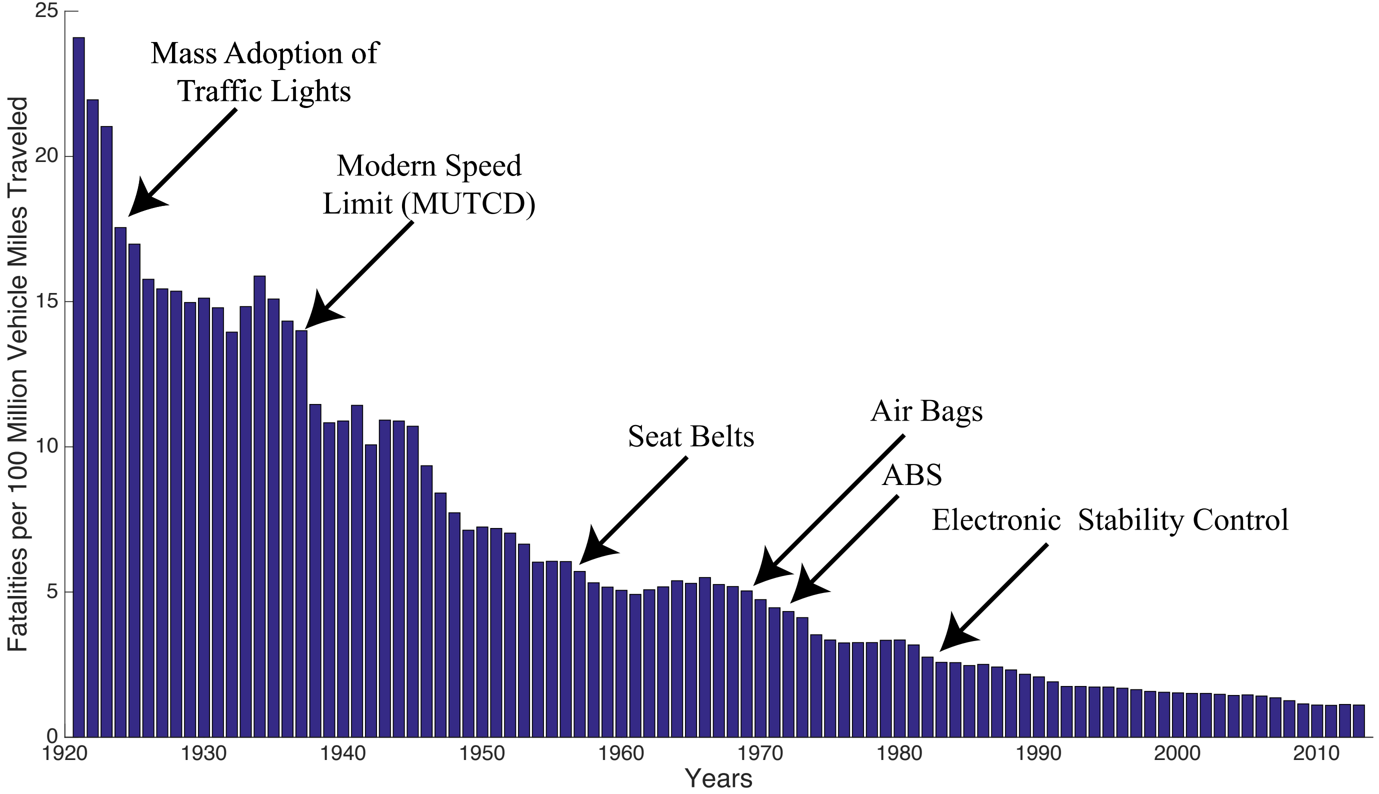
Ram Vasudevan

Mechanical Engineering
Robotics Institute
University of Michigan

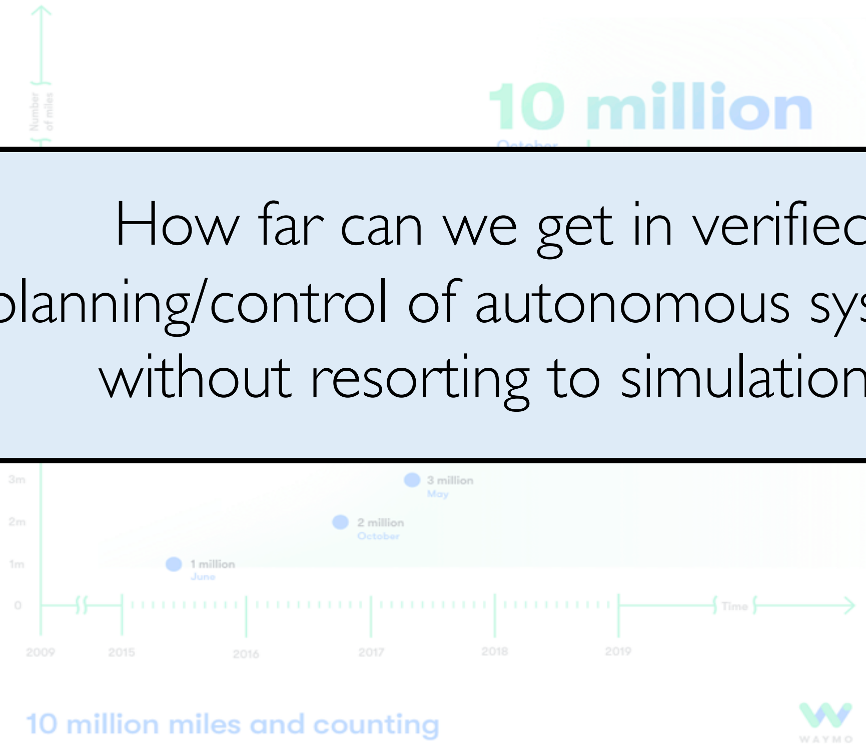
Improving Vehicle Safety



Improving Vehicle Safety

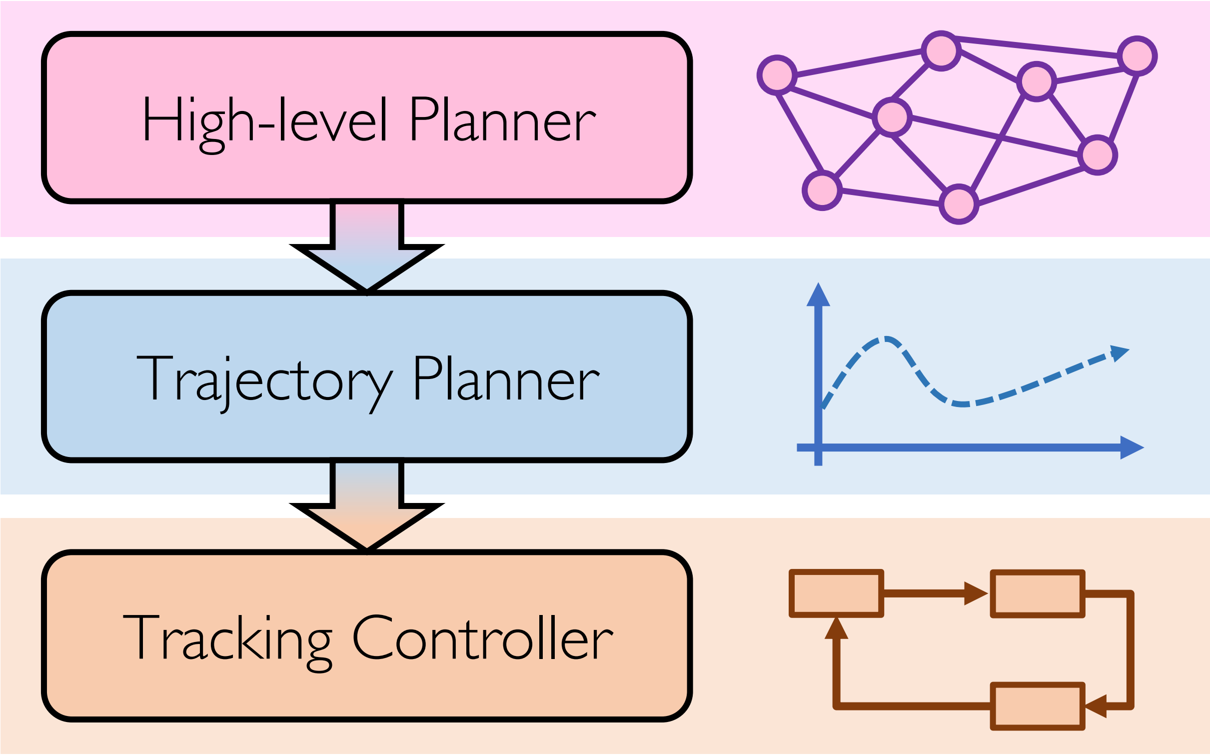


How Much AV Driving Has Happened?



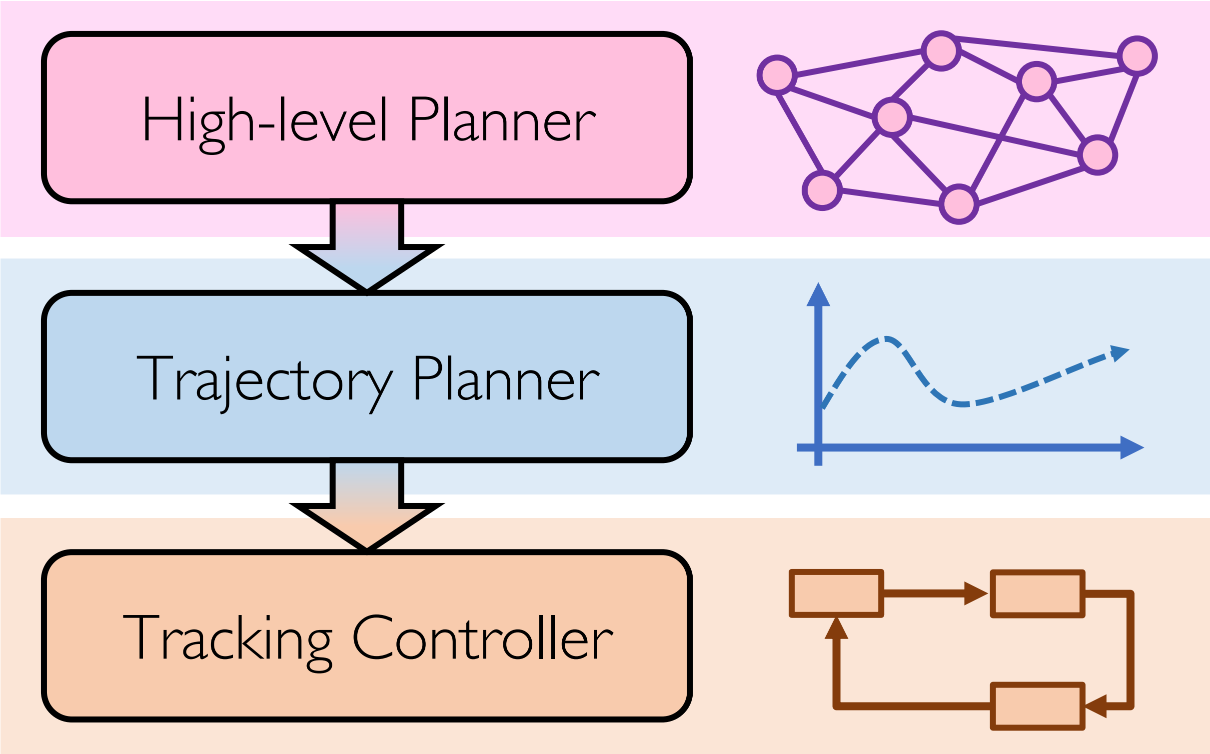
How far can we get in verified planning/control of autonomous systems without resorting to simulation?

Planning and Control Hierarchy



Why Apply A Planning Hierarchy?

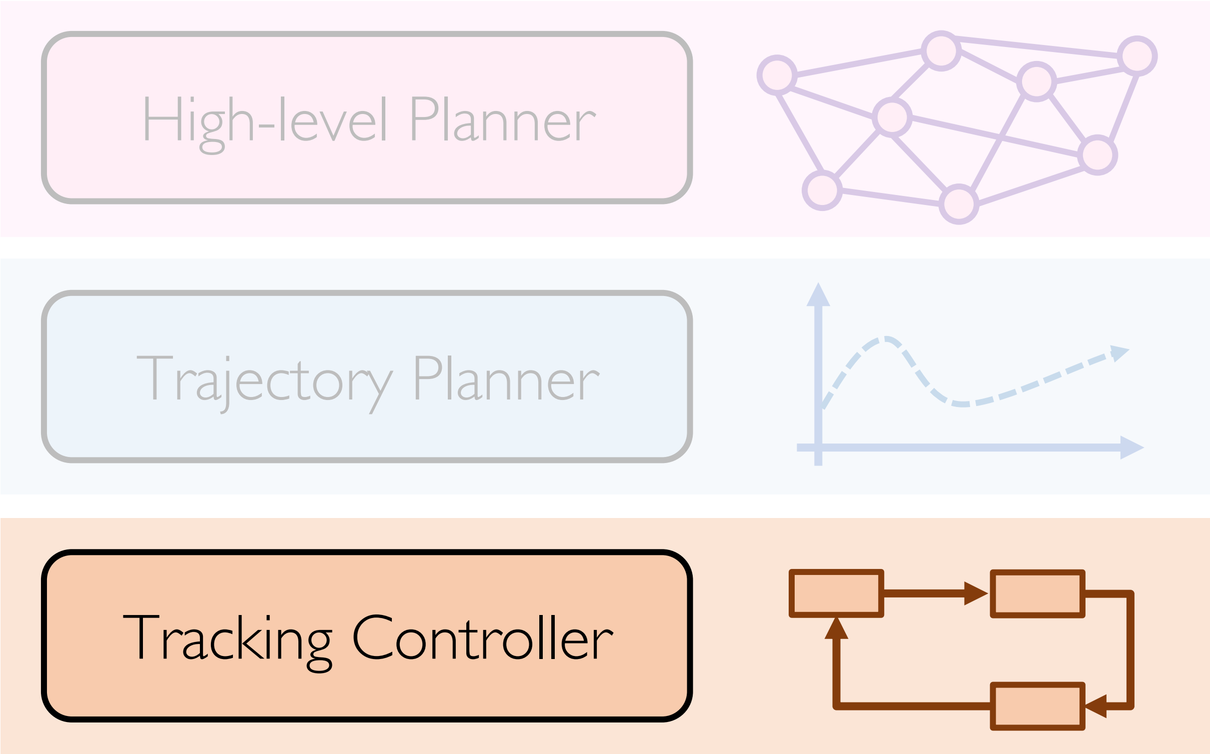
Planning and Control Hierarchy – How to Do It?



Path Generation + MPC

When Planning Goes Wrong

Bringing in Safety

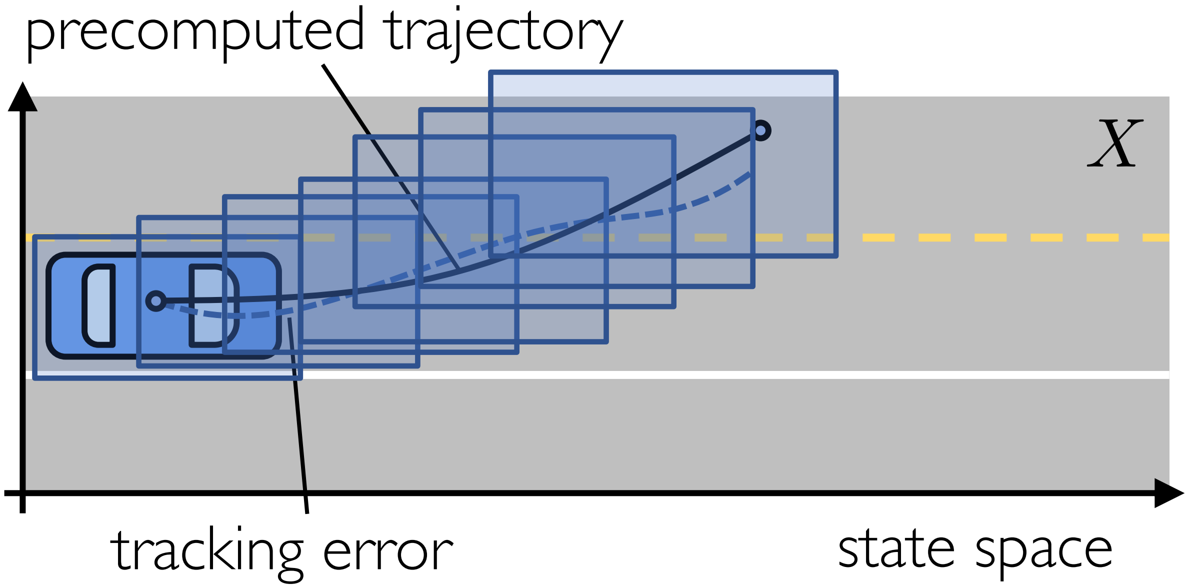


Overview of Validation + Path Planning Methods

Check - perform collision checking for precomputed trajectories

Zonotope Methods (Althoff, TRO 2014)

Verification via Check Methods - Zonotopes



Overview of Validation + Path Planning Methods

Check - perform collision checking for precomputed trajectories

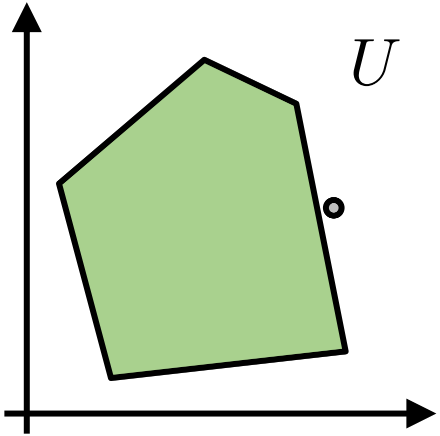
Zonotope Methods (Althoff, TRO 2014)

Correct – modify generated control inputs to ensure safety

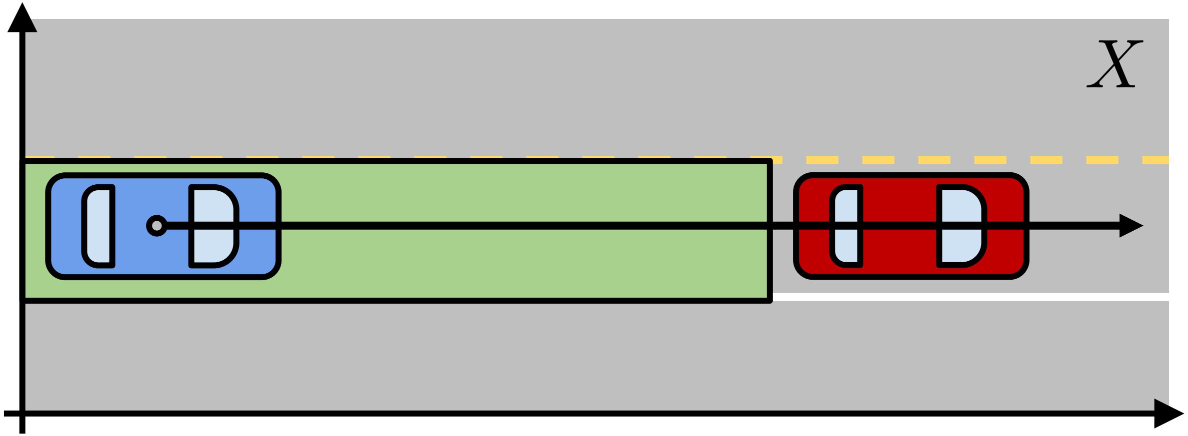
Control Barrier Functions (Ames et al., ECC 2019)

FaSTrack (Herbert, CDC 2017)

Verification via Correct Methods - CBFs

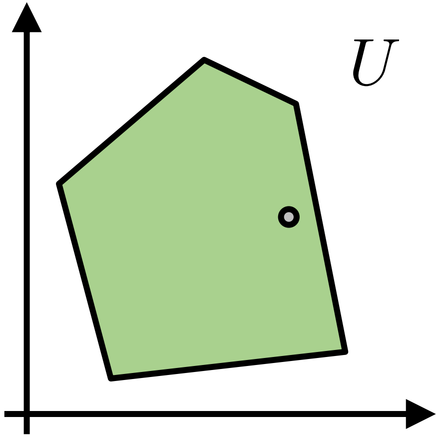


control inputs

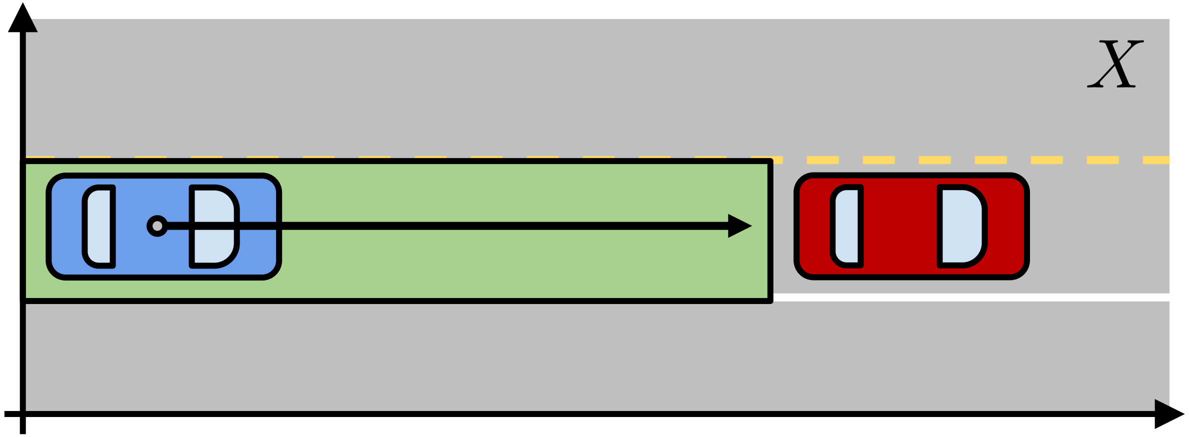


state space

Verification via Correct Methods - CBFs

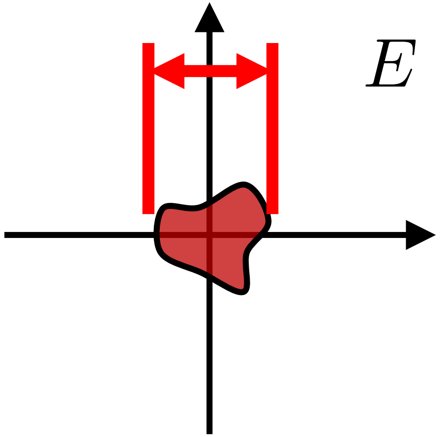


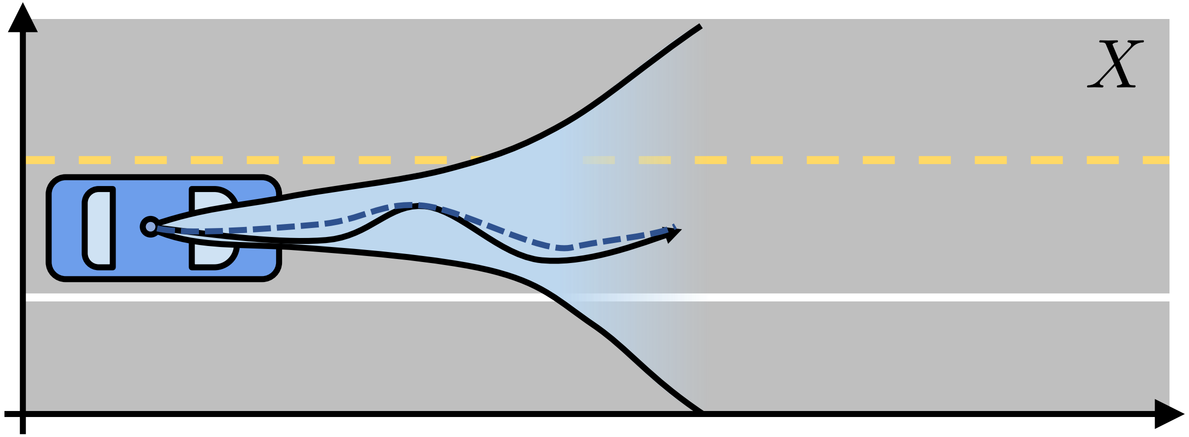
control inputs



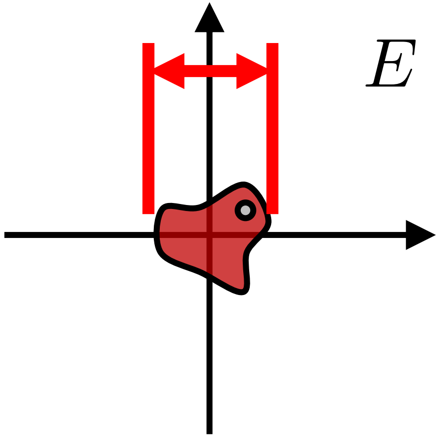
state space

Verification via Correct Methods - FaSTrack

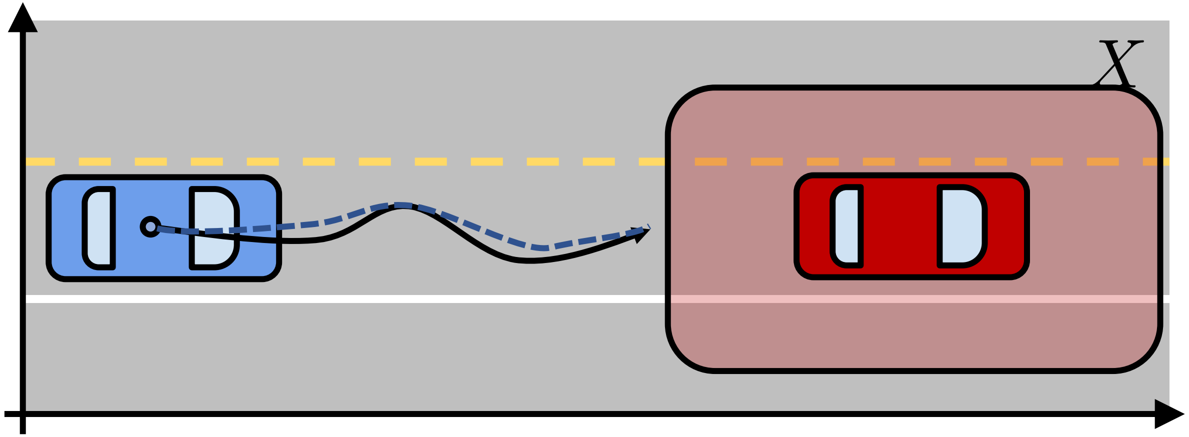




Verification via Correct Methods - FaSTrack

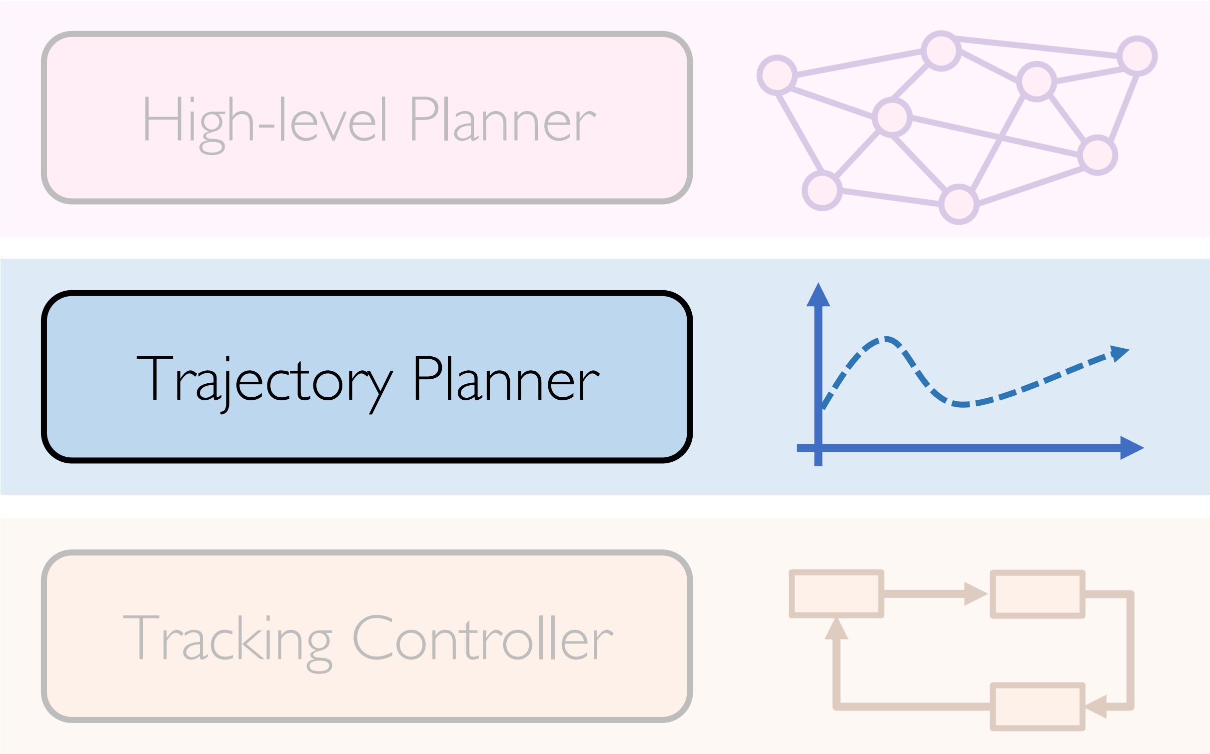


tracking error



state space

Bringing in Safety



Overview of Validation + Path Planning Methods

Check - perform collision checking for precomputed trajectories

Zonotope Methods (Althoff, TRO 2014)

Correct – modify generated control inputs to ensure safety

Control Barrier Functions (Ames et al., ECC 2019)

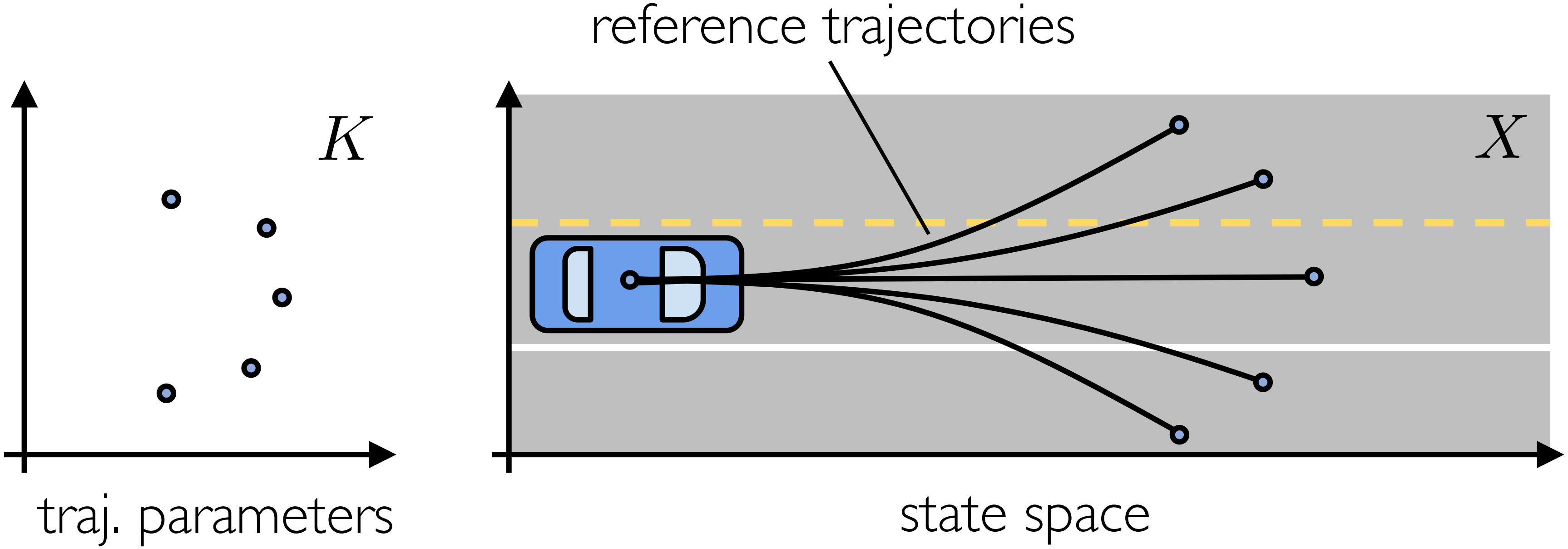
FaSTrack (Herbert, CDC 2017)

Select – select a not-at-fault trajectory at run-time

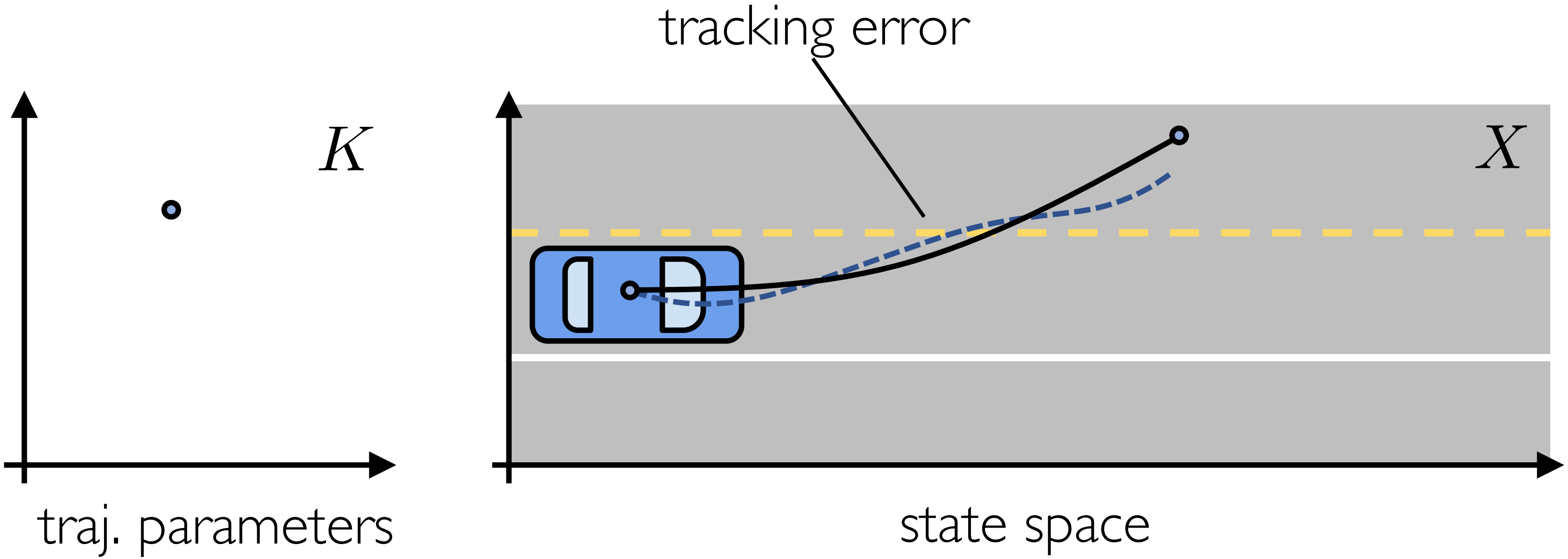
Funnel Library (Majumdar, IJRR 2017)

Reachability-based Trajectory Design (IJRR, in press)

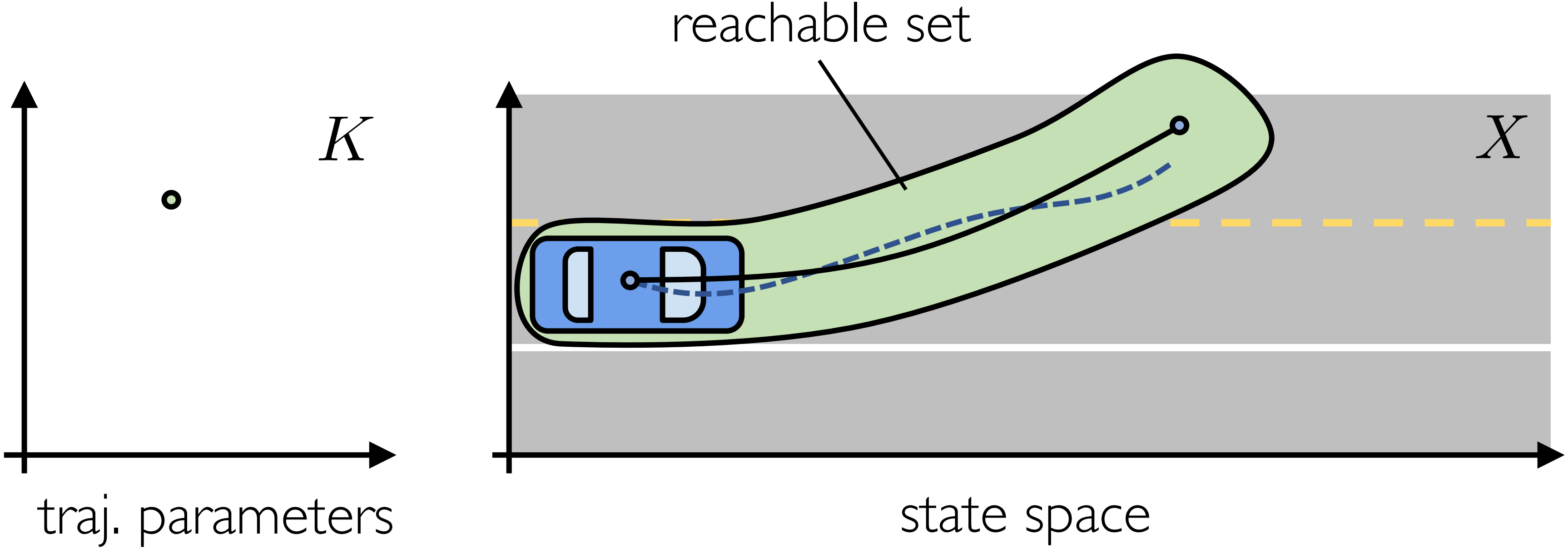
Verification via Select Methods - Funnel Library



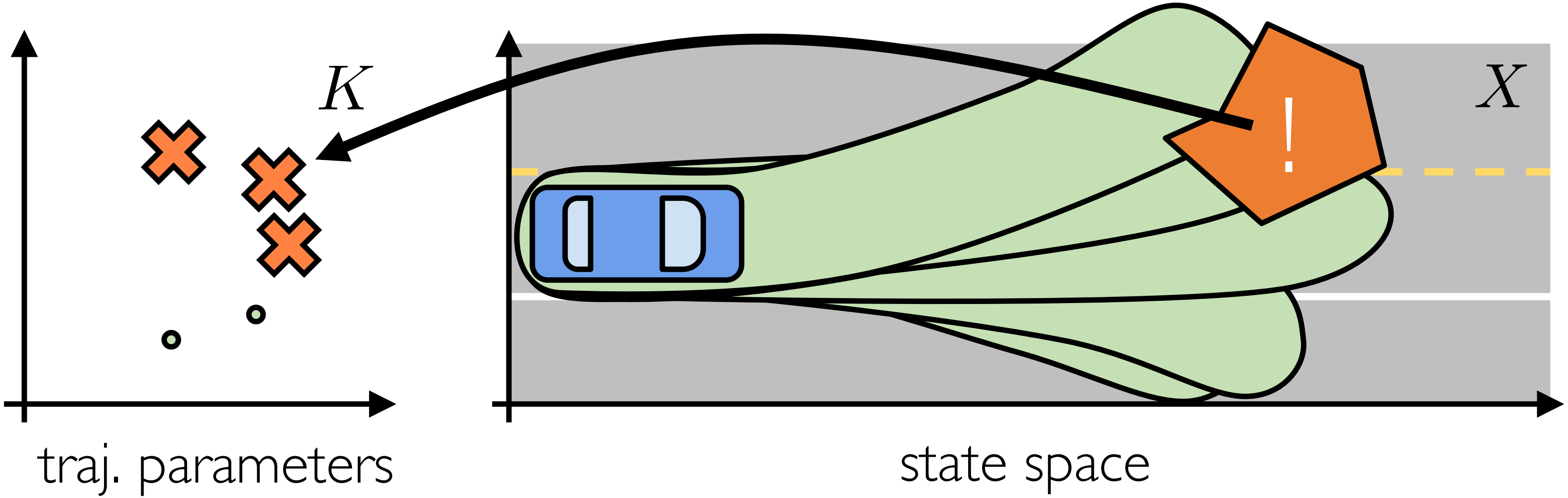
Verification via Select Methods - Funnel Library



Verification via Select Methods - Funnel Library



Verification via Select Methods - Funnel Library



Verification via Select Methods - Funnel Library

Collision check is not verified to operate safely

Have to pick finite library a priori

Reachability-based Trajectory Design

Reachability with traj-dependent error (offline)

Obstacle discretization for real-time validation (online)

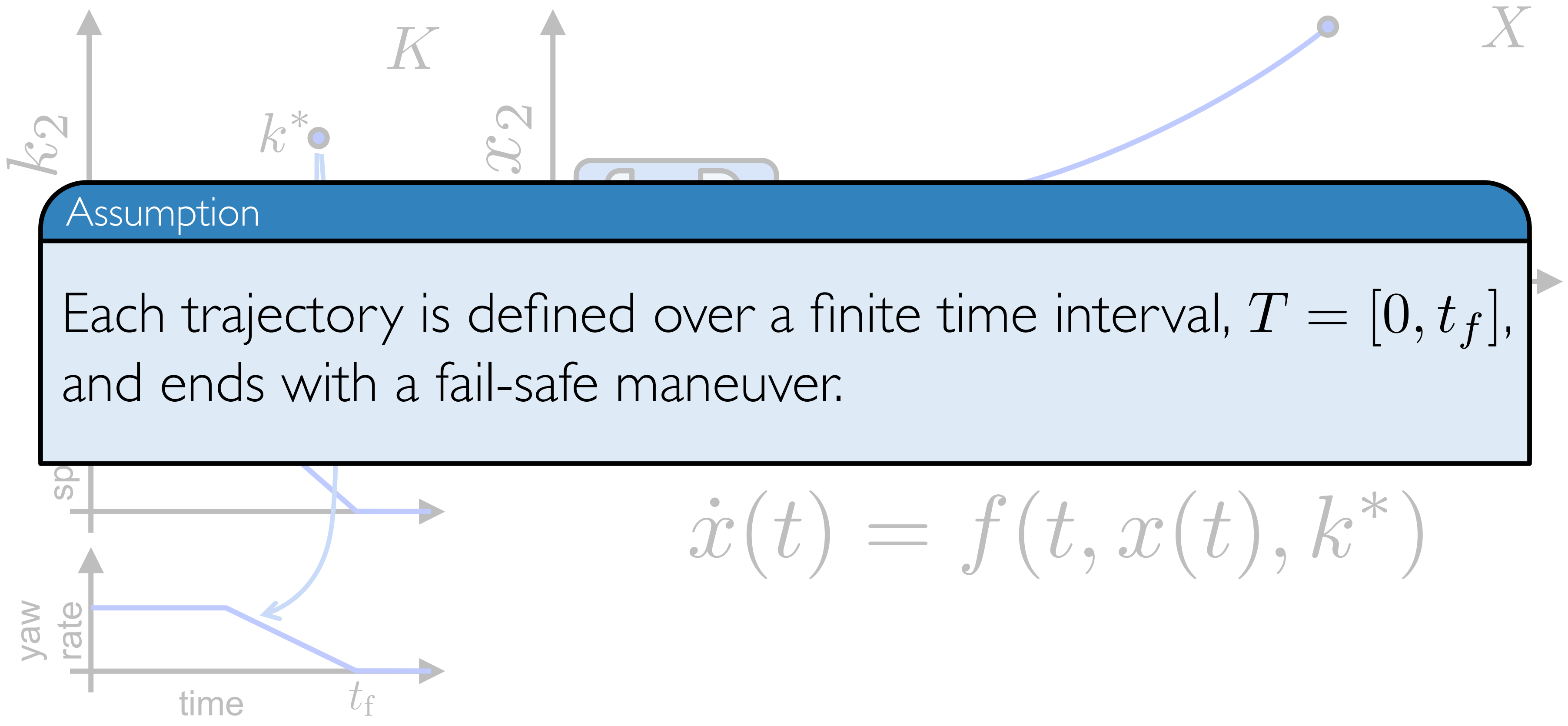
Outline

1. Reachability-based Trajectory Design for Provably Safe, Real-Time Planning for Autonomous Vehicles – STATIC
2. Reachability-based Trajectory Design for Provably Not-at-Fault, Real-Time Planning for Autonomous Vehicles – DYNAMIC
3. Conclusion

Outline

1. **Reachability-based Trajectory Design for Provably Safe, Real-Time Planning for Autonomous Vehicles – STATIC**
2. Reachability-based Trajectory Design for Provably Not-at-Fault, Real-Time Planning for Autonomous Vehicles – DYNAMIC
3. Conclusion

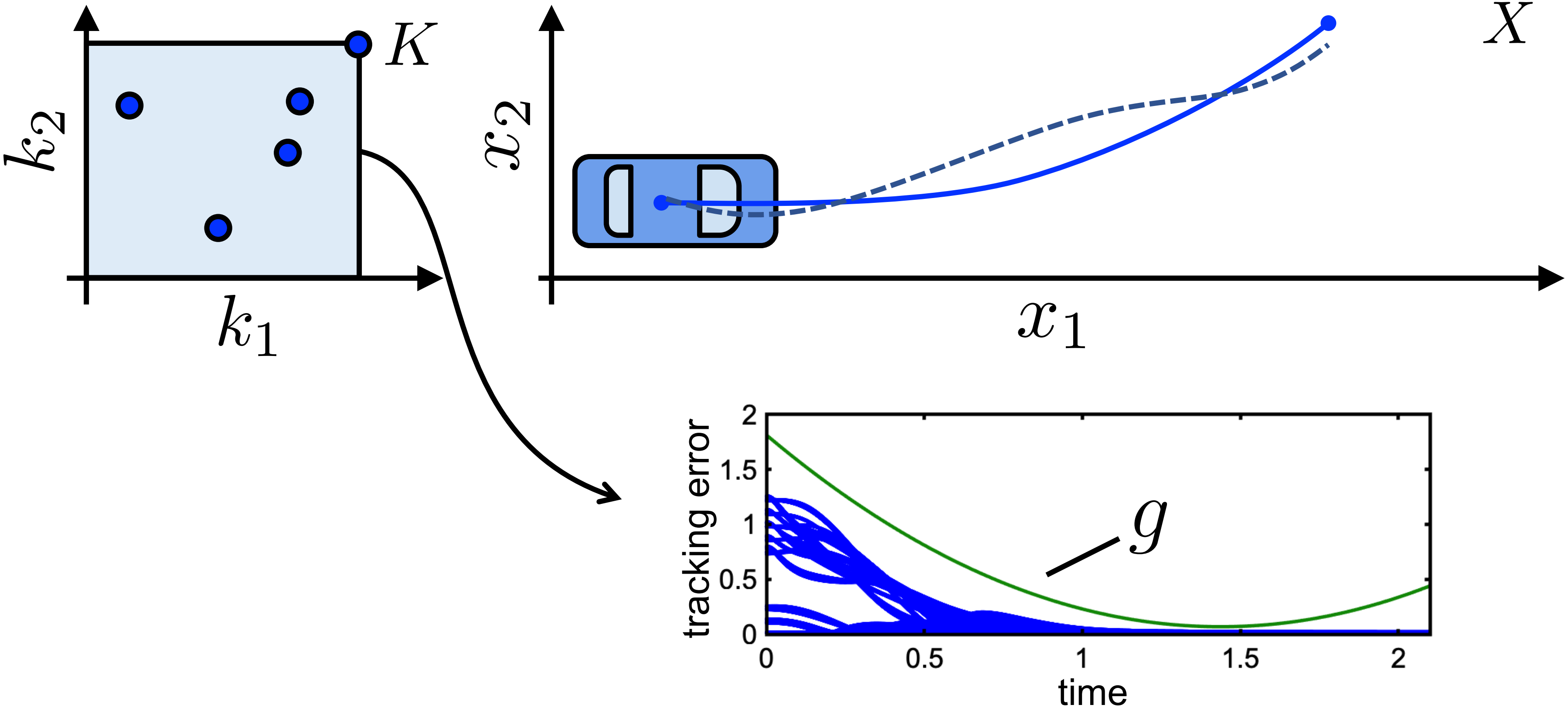
An Example of Trajectory Parameterization



Assumption

Each trajectory is defined over a finite time interval, $T = [0, t_f]$, and ends with a fail-safe maneuver.

An Example of Trajectory Parameterization



Bounding Tracking Error

Assumption (DSCC 2019, Smith et al. CDC 2019)

For each $j \in \{1, 2\}$, there exists g_j such that

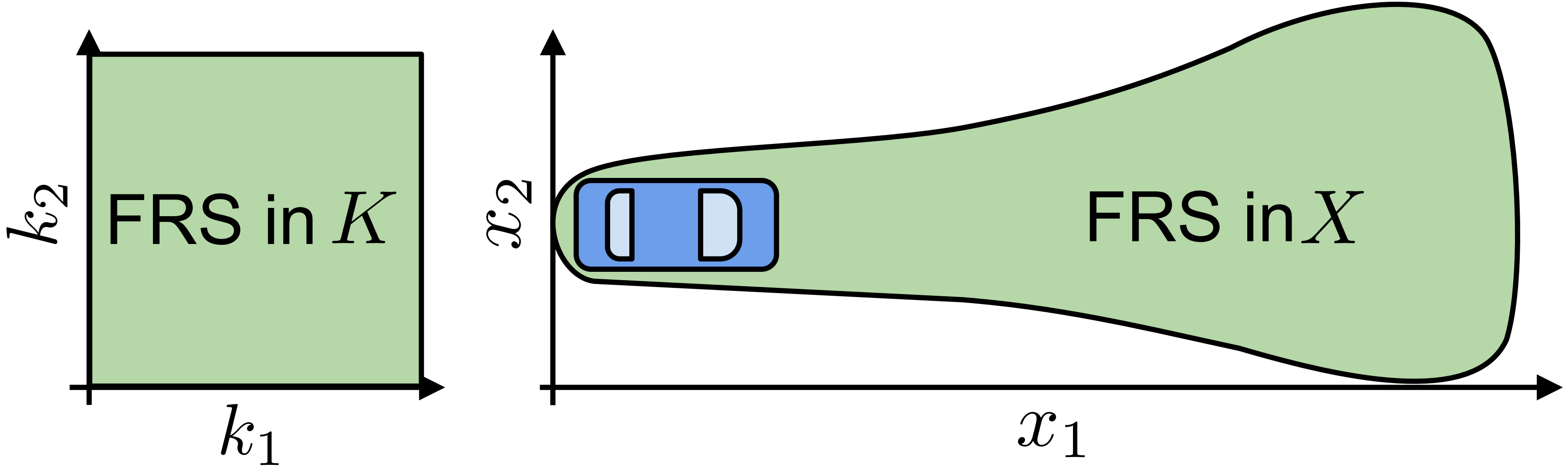
$$\left| x_{\mathrm{hi},j}(t;k) - x_j(t;k) \right| \leq$$

Vehicle Trajectory

Reference Trajectory

1 1.5 2
time

The Forward Reachable Set (FRS)



$$\dot{x}(t) = f(t, x(t), k) + g(t, k)d(t)$$

$$d(t) \in [-1, 1]$$

SOS Reachability (offline)

$$\inf_{\boxed{w,v,q}} \int_{X \times K} w(x,k)\, d\lambda_{X \times K}$$

$w : X \times K \to \mathbb{R}$ converges to indicator function on FRS

SOS Reachability (offline)

$$\inf_{w,v,q} \int_{X \times K} w(x,k)\, d\lambda_{X\times K}$$

$$\text{s.t.} \quad \boxed{\begin{aligned} &-\left(\frac{\partial v}{\partial t} + \frac{\partial v}{\partial x} f\right) \geq 0 \\ &-v(0,x,k) \geq 0 \end{aligned}}$$

$v : T \times X \times K \to \mathbb{R}$ is like a Lyapunov function

SOS Reachability (offline)

$$\inf_{w,v,q} \int_{X\times K} w(x,k)\,d\lambda_{X\times K}$$

$$\text{s.t.} \quad -\left(\frac{\partial v}{\partial t} + \frac{\partial v}{\partial x} f\right) - \boxed{q(t,x,k)} \geq 0$$

$$-\,v(0,x,k) \geq 0$$

$$\boxed{\begin{aligned} & q(t,x,k) \pm \left(\frac{\partial v}{\partial t} + \frac{\partial v}{\partial x} g\right) \geq 0 \\ & q(t,x,k) \geq 0 \end{aligned}}$$

$q : T \times X \times K \to \mathbb{R}$

incorporates uncertainty

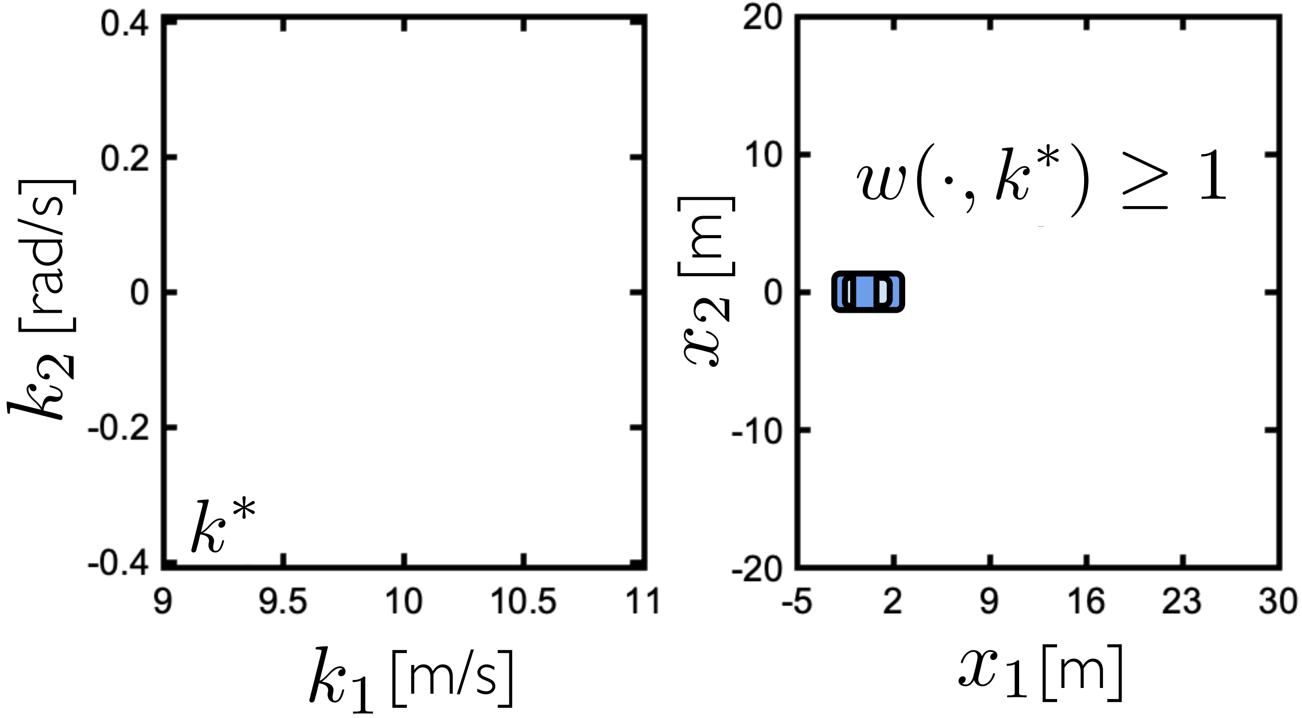
SOS Reachability (offline)

$$
\begin{aligned}
\inf_{w,v,q} \int_{X \times K} & w(x,k)\, d\lambda_{X \times K} \\
\text{s.t.} \quad & -\left(\frac{\partial v}{\partial t} + \frac{\partial v}{\partial x} f\right) - q(t,x,k) \geq 0 \\
& -v(0,x,k) \geq 0 \\
& q(t,x,k) \pm \left(\frac{\partial v}{\partial t} + \frac{\partial v}{\partial x} g\right) \geq 0 \\
& q(t,x,k) \geq 0 \\
& \boxed{\begin{aligned} & w(x,k) + v(t,x,k) - 1 \geq 0 \\ & w(x,k) \geq 0 \end{aligned}}
\end{aligned}
$$

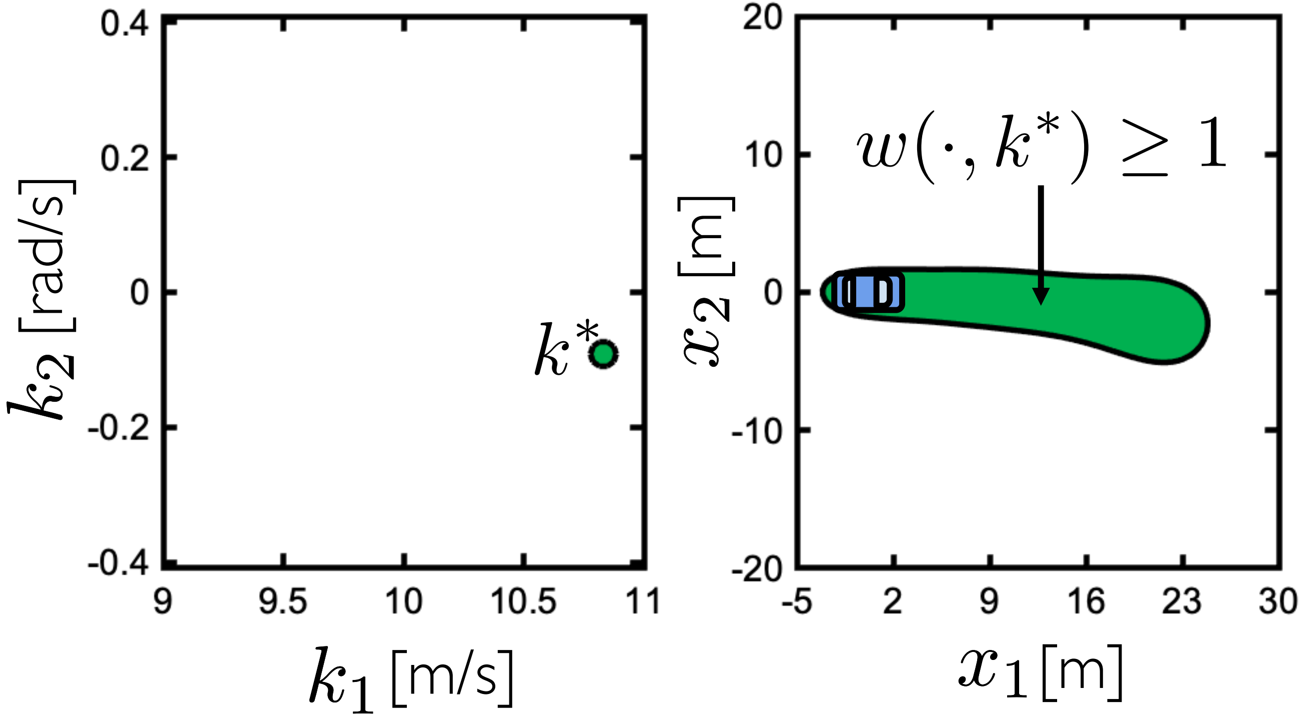
Theorem

$$(x,k) \in \mathrm{FRS} \implies w(x,k) \geq 1$$

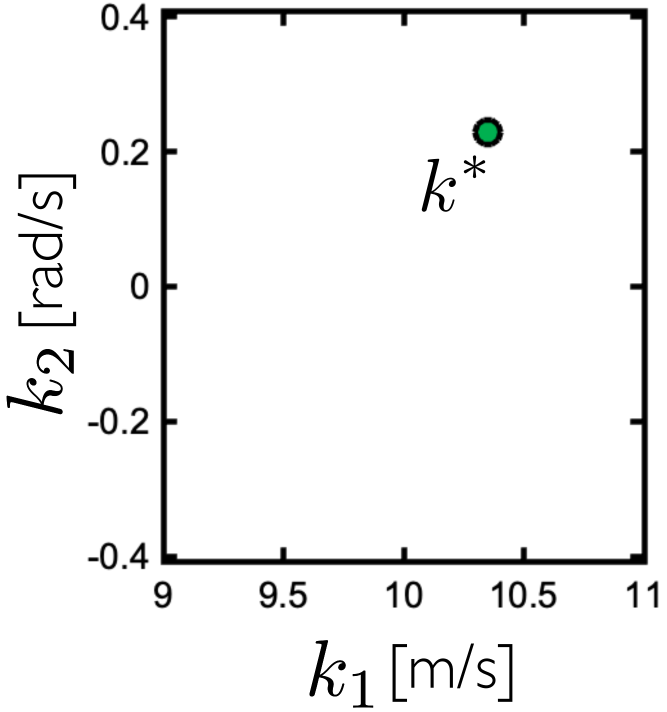
Projection from K to X

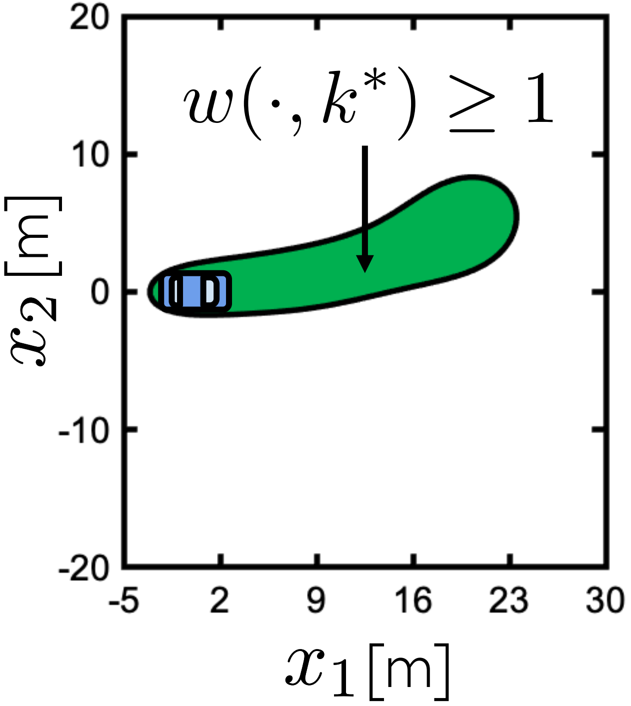


Projection from K to X

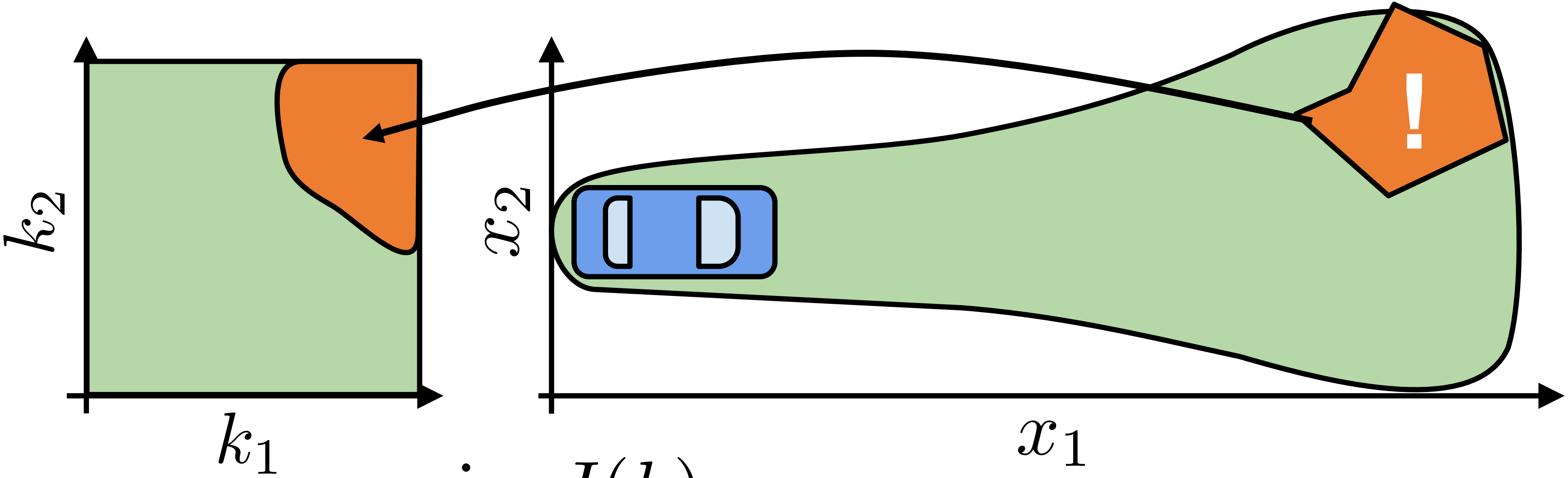


Projection from K to X





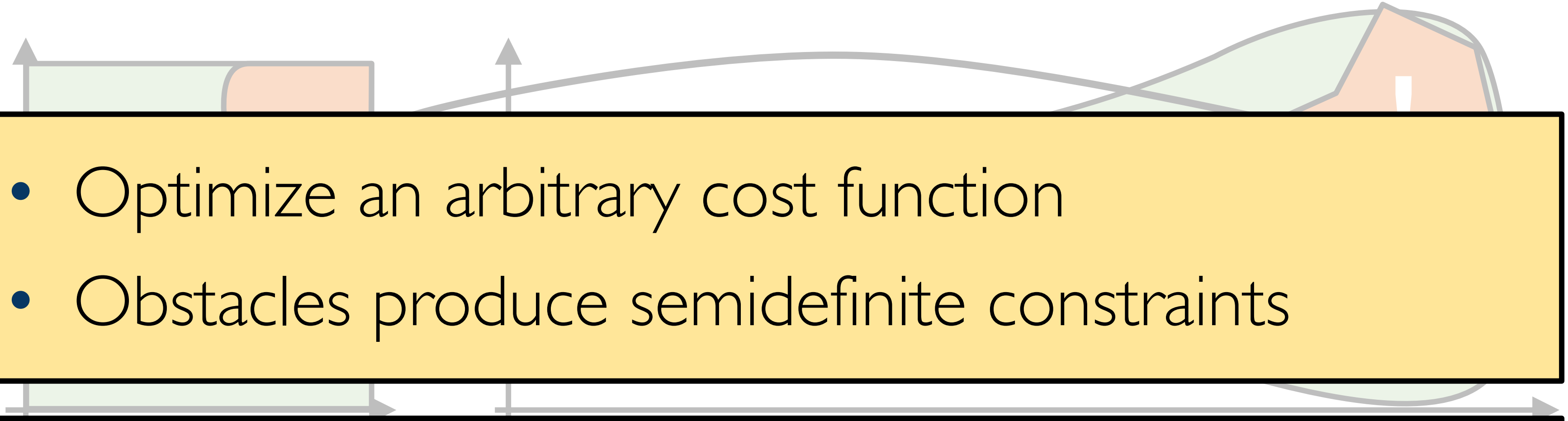
Goal for Online Computation



$$\min_{k \in K} J(k)$$

$$\text{s.t. } w(x,k) < 1, \ \forall \ x \in \text{obstacle}$$

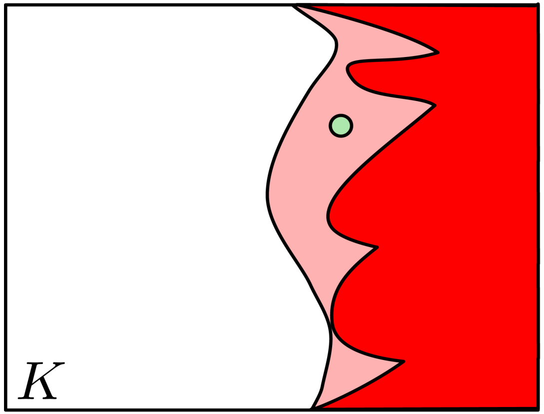
Goal for Online Computation

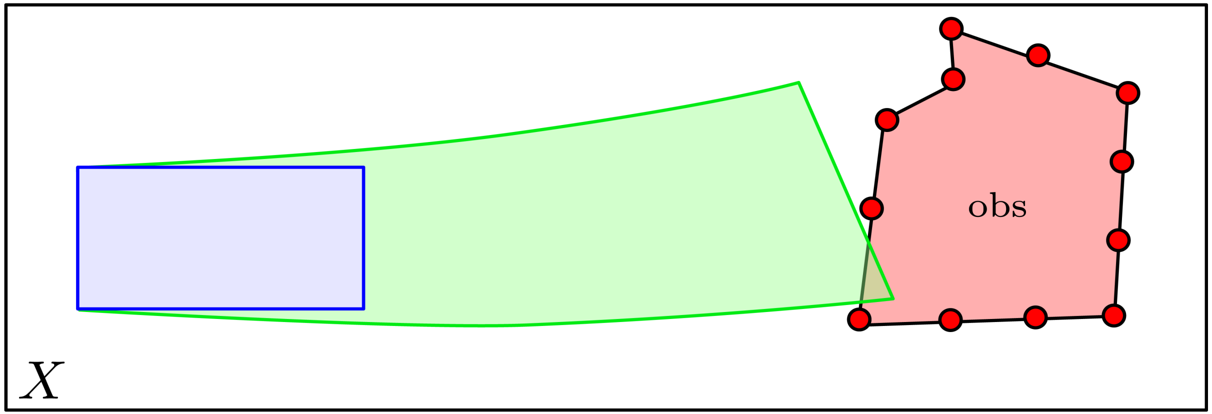


- Optimize an arbitrary cost function
- Obstacles produce semidefinite constraints

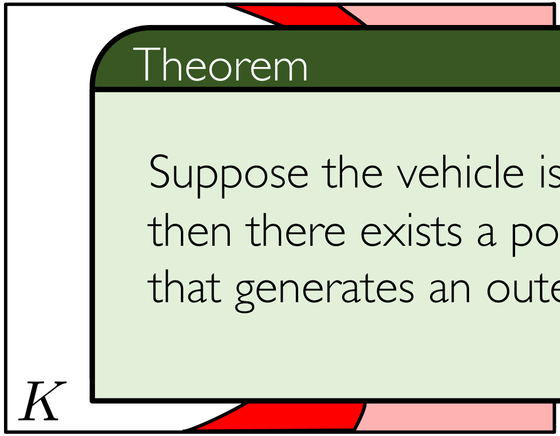
Obstacle Shape	Method	Mean Time [ms]	Std. Dev [ms]
Box	Set Intersection	17,800	1010
Line	Set Intersection	1050	73

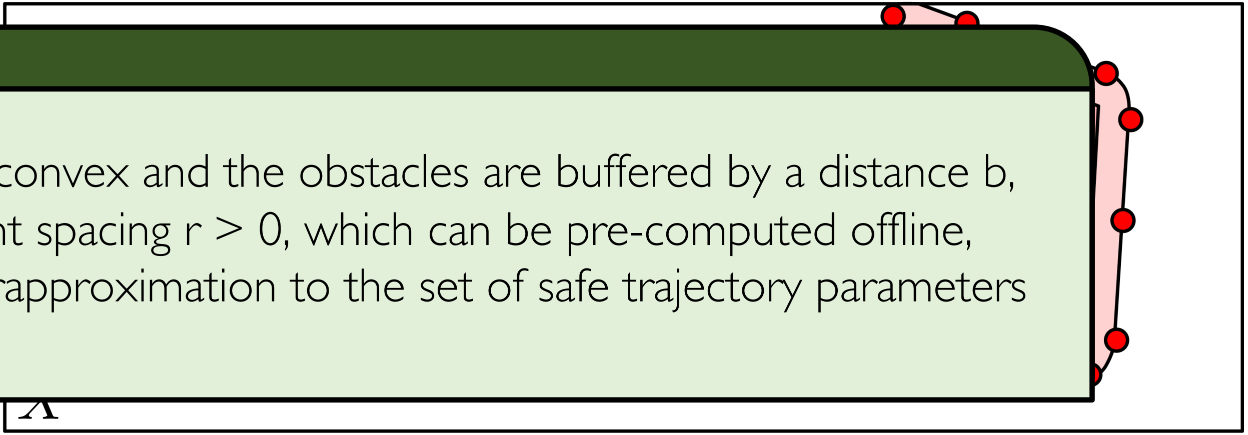
Obstacle Representations





Provably Safe, Obstacle Representations

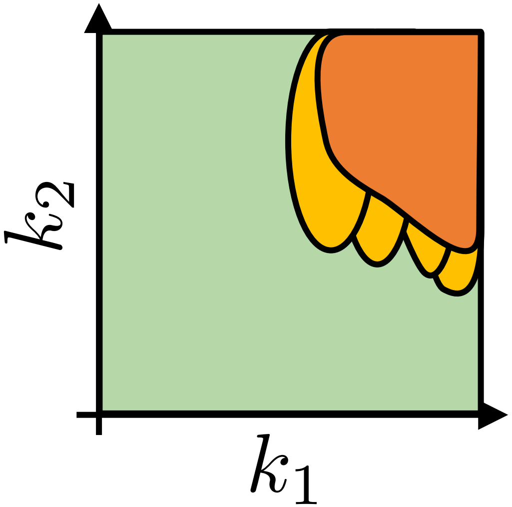




Theorem

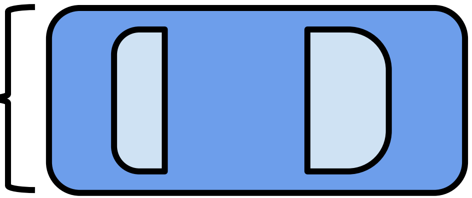
Suppose the vehicle is convex and the obstacles are buffered by a distance b, then there exists a point spacing r > 0, which can be pre-computed offline, that generates an outerapproximation to the set of safe trajectory parameters

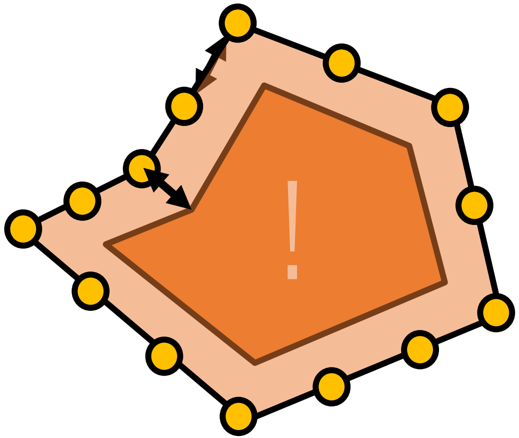
Provably Safe, Obstacle Representations for AVs



Point Spacing $\leq 2b$

Buffer $b \in (0, \frac{1}{2}W)$

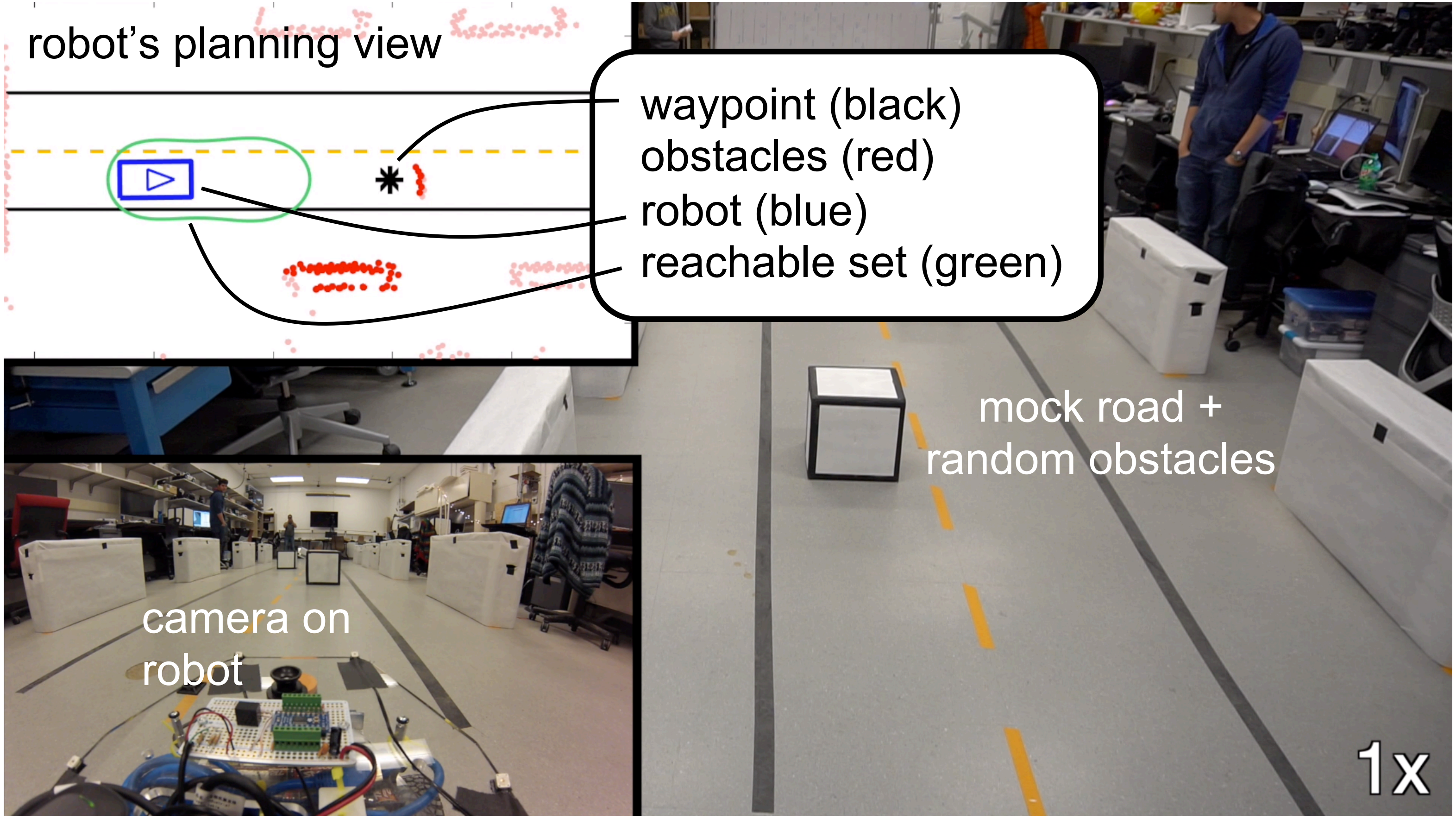




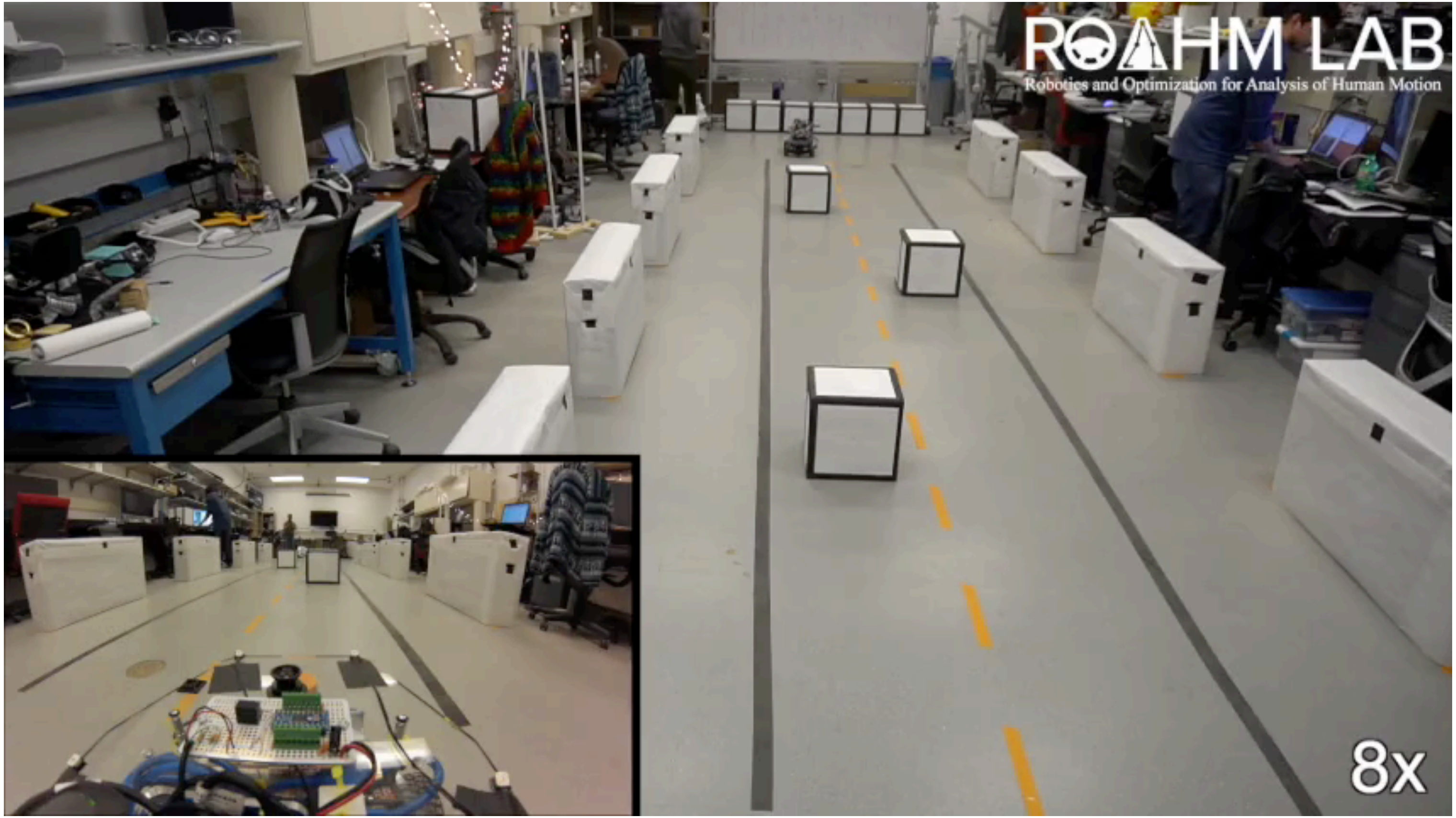
Theorem

If $\forall \circ \ w(\circ, k^*) < 1$, then AV avoids obstacle.

RTD Results – Rover



RTD Results – Rover

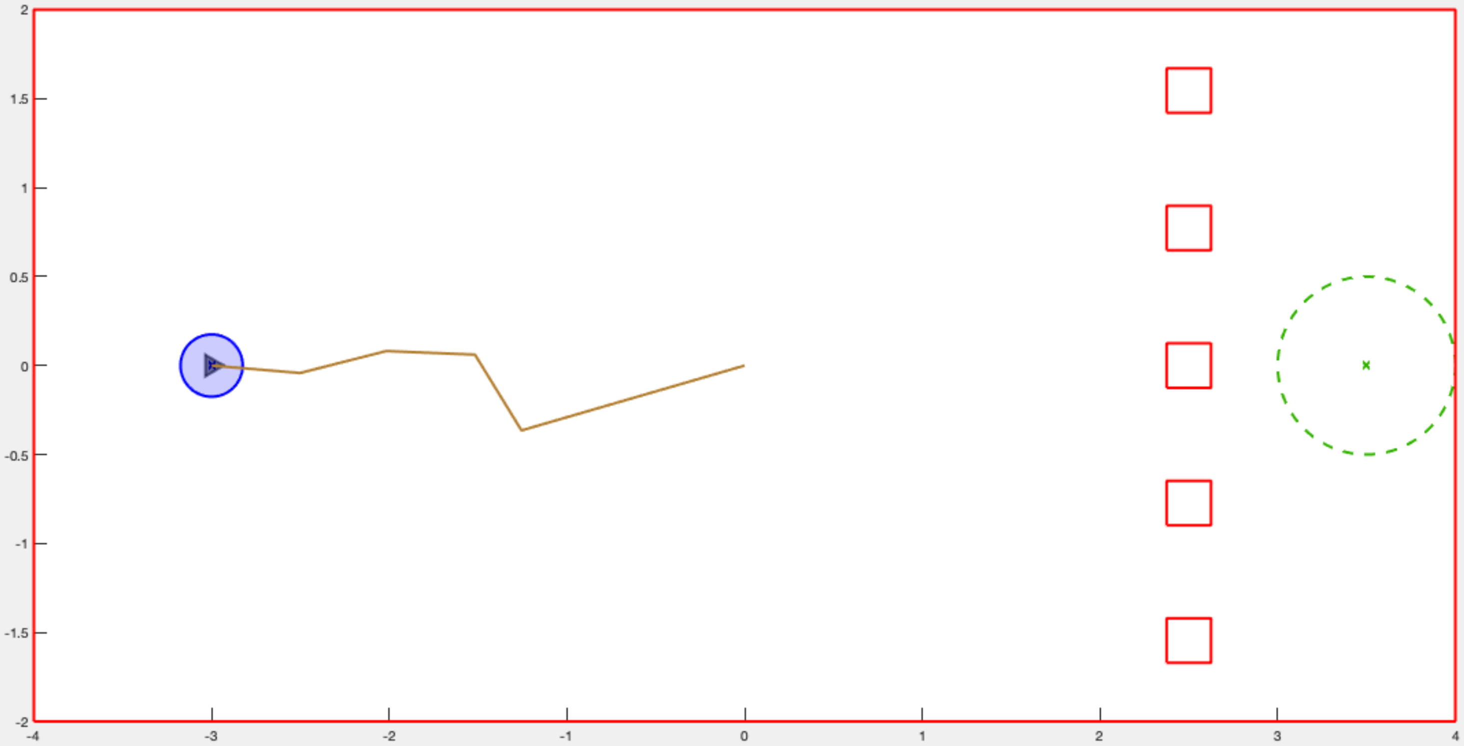


Comparison to Other Planners – Real-Time

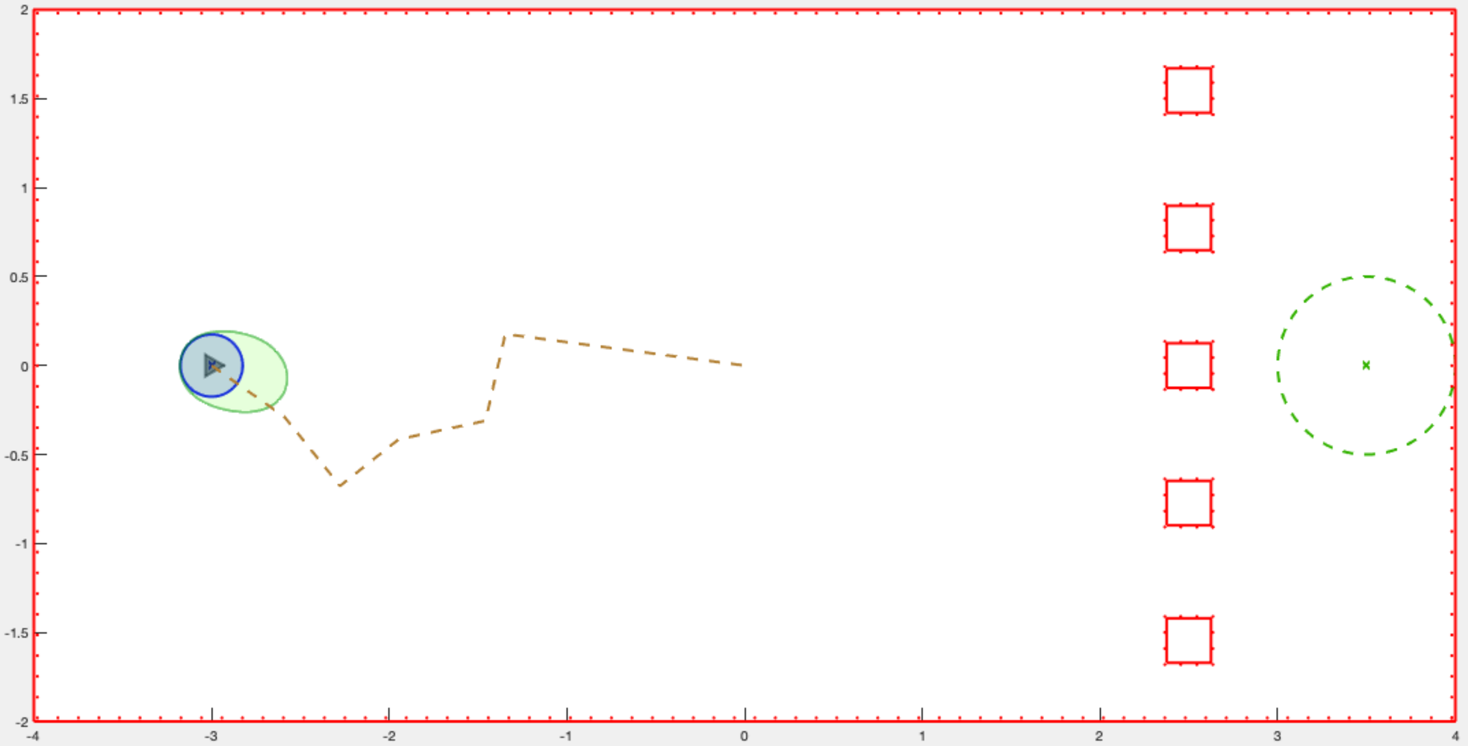
Evaluate real-time ability of planners to get to a goal and avoid randomly generated obstacles over 10k scenarios

τ_{plan} [s]	D_{sense} [m]	Planner	Goals [%]	Crashes [%]
0.5	4.0	RTD	**87.4**	**0.0**
		RRT	77.2	6.3
		NMPC	0.0	**0.0**

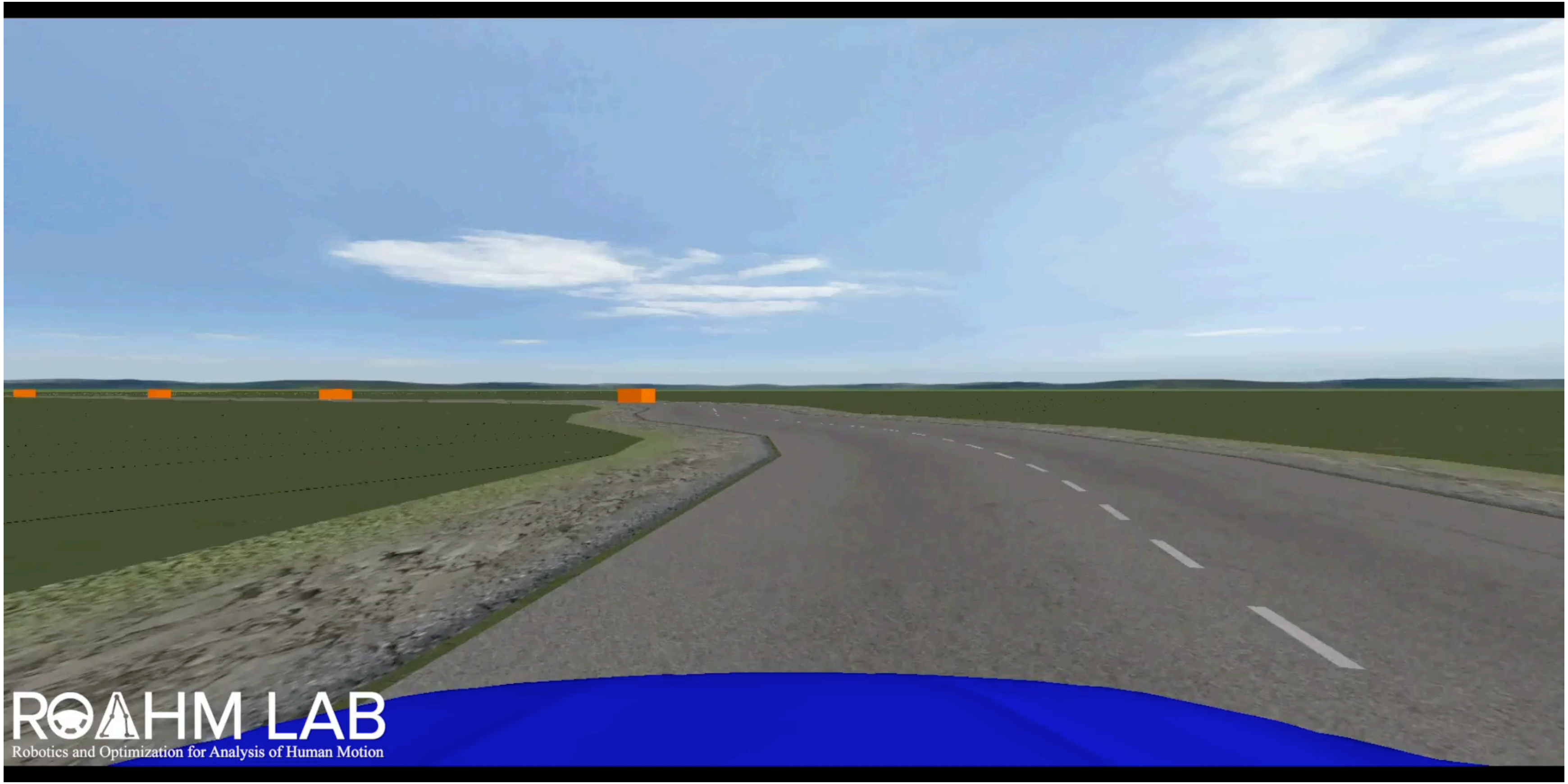
Comparison of FaSTrack to RTD



Comparison of FaSTrack to RTD



CarSim Full Powertrain Experiment



Comparison: 10 trials, $\tau_{\mathbf{plan}} = 0.5$s

Planner	Avg Planning Time	% of Track Complete		Crashes	Safe Stops
		Avg	Max		

[1] Y. Kuwata et al., TCST, 2009. [2] M. A. Patterson and A. V. Rao, ACM Trans. Math. Software, 2014.

Outline

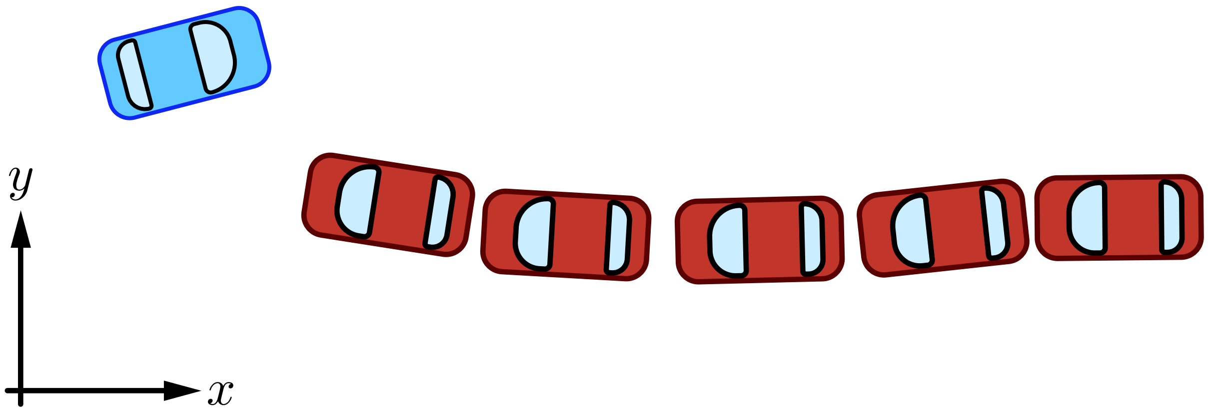
1. Reachability-based Trajectory Design for Provably Safe, Real-Time Planning for Autonomous Vehicles – STATIC

2. **Reachability-based Trajectory Design for Provably Not-at-Fault, Real-Time Planning for Autonomous Vehicles – DYNAMIC**

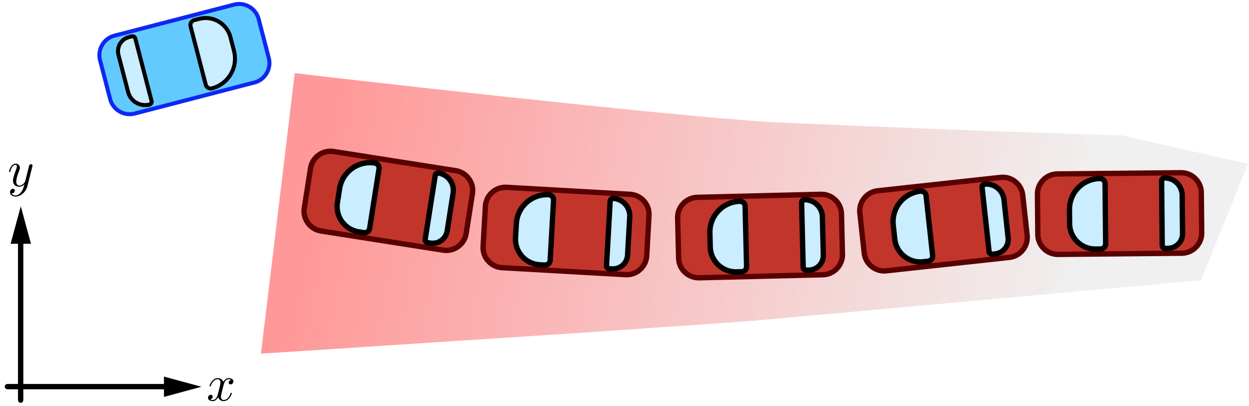
3. Conclusion

Moving Obstacles

Moving Obstacles



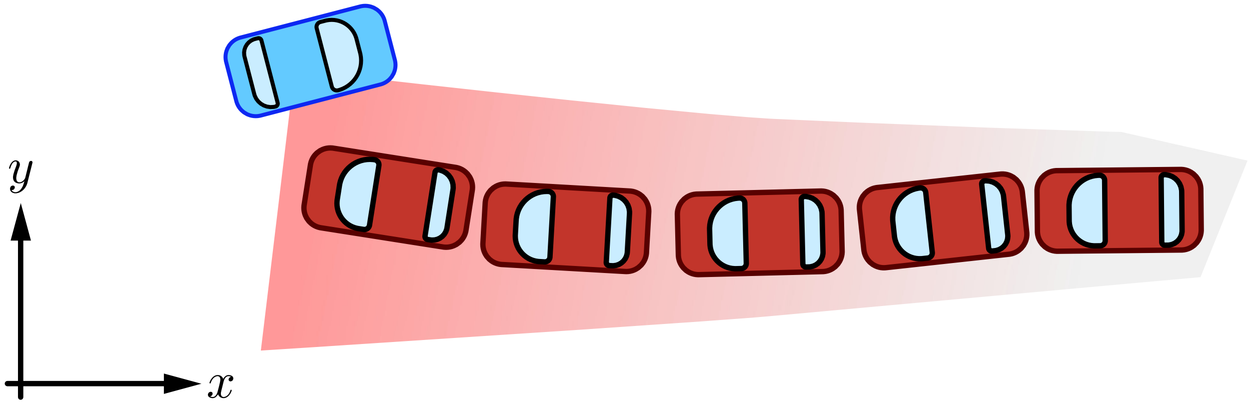
Moving Obstacles



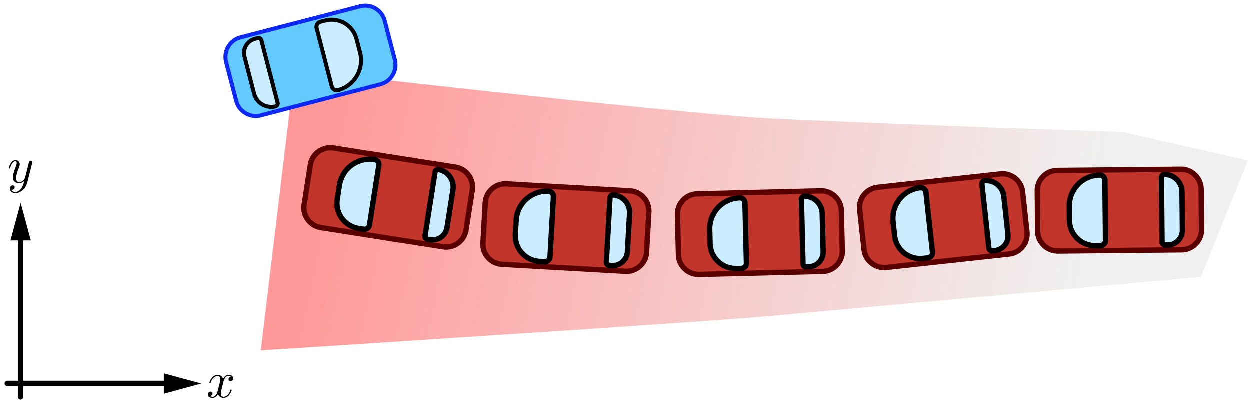
Assumption

The predictions are conservative and contain the true behavior of the obstacles.

Moving Obstacles



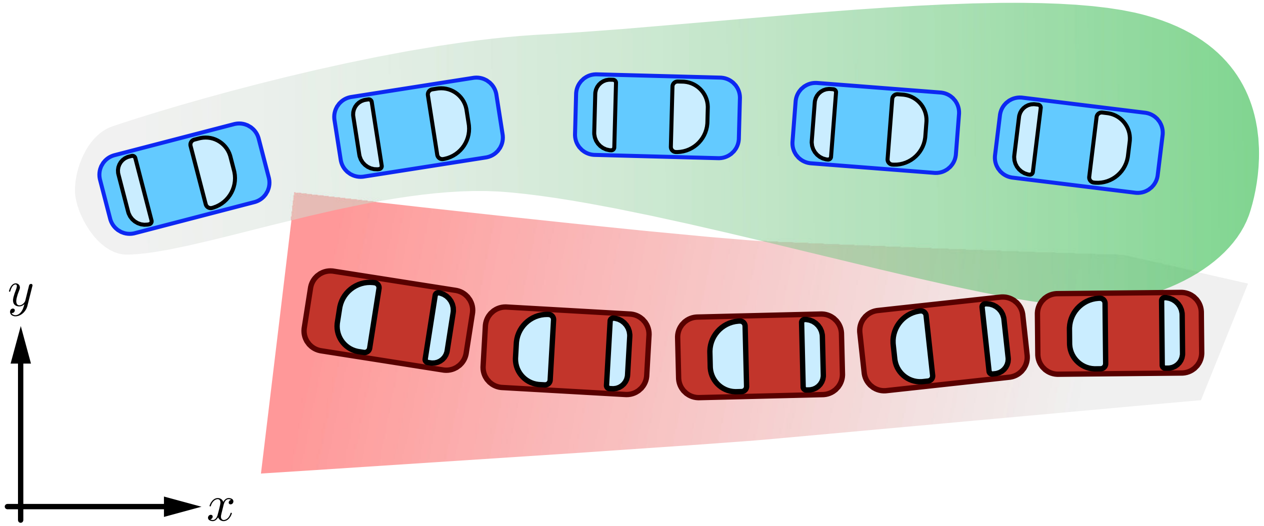
Moving Obstacles



Definition

The AV is not at fault at time t if it is stationary or not intersecting any obstacles.

Real-Time, Provably Not-at-Fault Planning



SOS Reachability (offline)

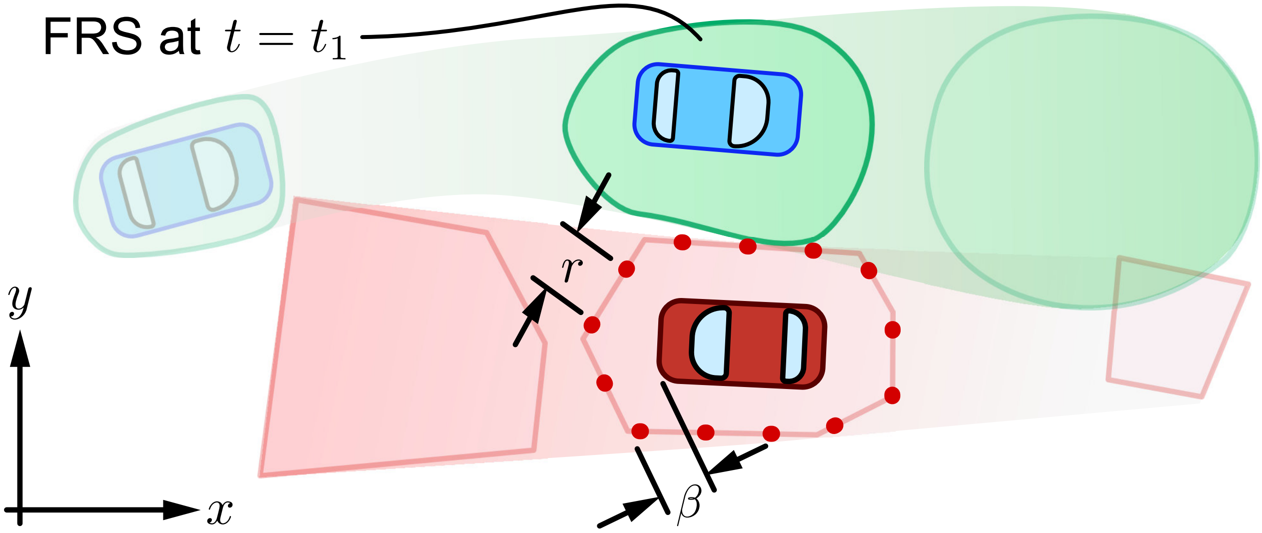
$$\inf_{w,v,q} \int_{X\times K} w(x,k)\,d\lambda_{X\times K}$$

$$\begin{aligned}
\text{s.t.}\quad & -\left(\frac{\partial v}{\partial t} + \frac{\partial v}{\partial x} f\right) - q(t,x,k) \geq 0 \\
& -v(0,x,k) \geq 0 \\
& q(t,x,k) \pm \left(\frac{\partial v}{\partial t} + \frac{\partial v}{\partial x} g\right) \geq 0 \\
& q(t,x,k) \geq 0 \\
& \boxed{\begin{aligned} & w(x,k) + v(t,x,k) - 1 \geq 0 \\ & w(x,k) \geq 0 \end{aligned}}
\end{aligned}$$

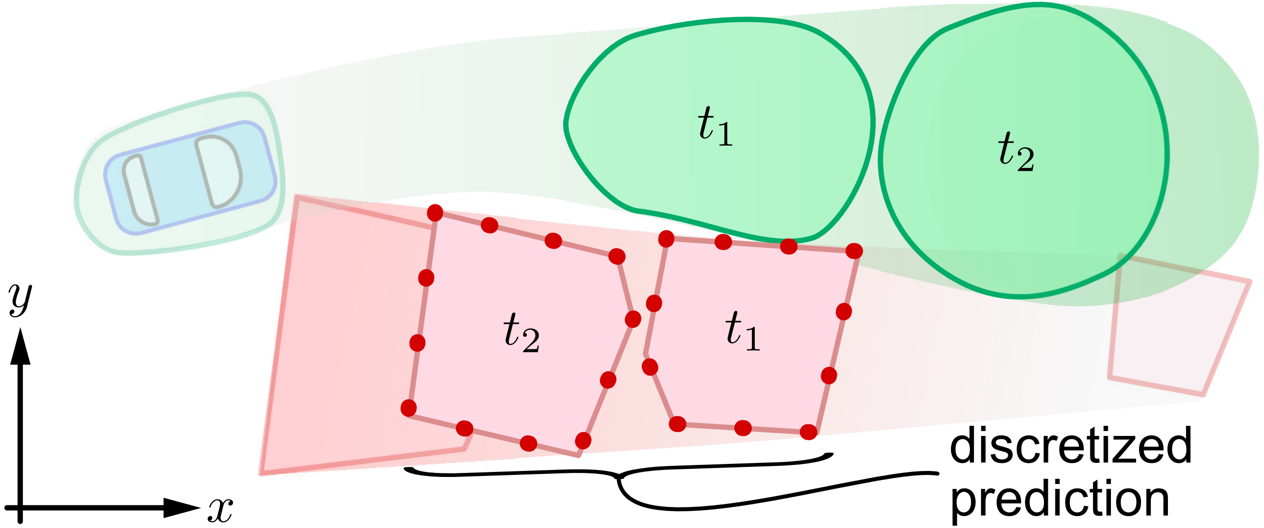
Theorem

$$(t,x,k) \in \mathrm{FRS} \implies v(t,x,k) \leq 0$$

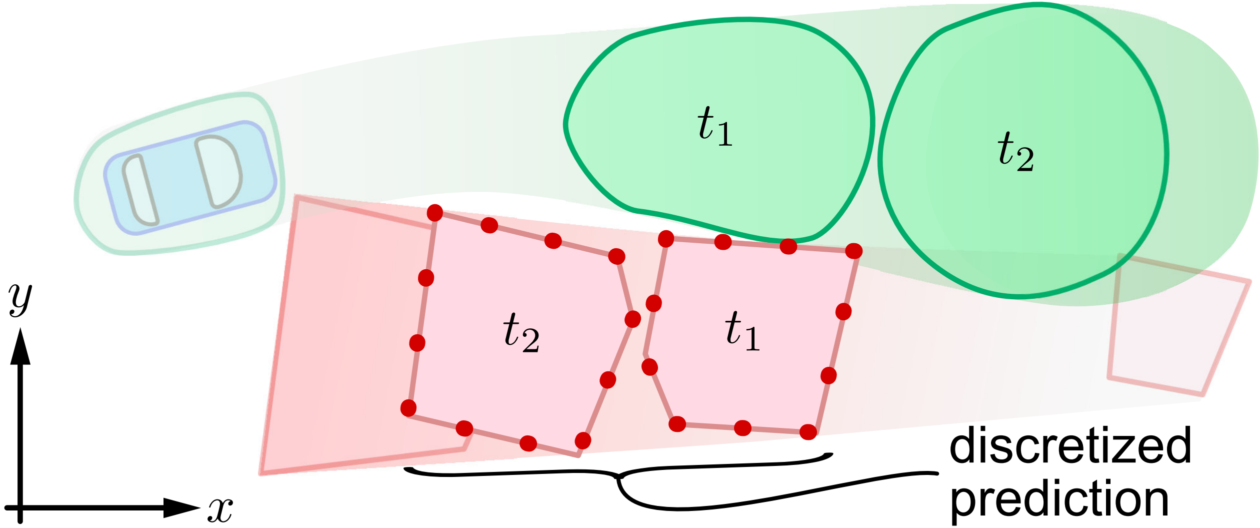
Real-Time, Provably Not-at-Fault Planning



Real-Time, Provably Not-at-Fault Planning



Real-Time, Provably Not-at-Fault Planning

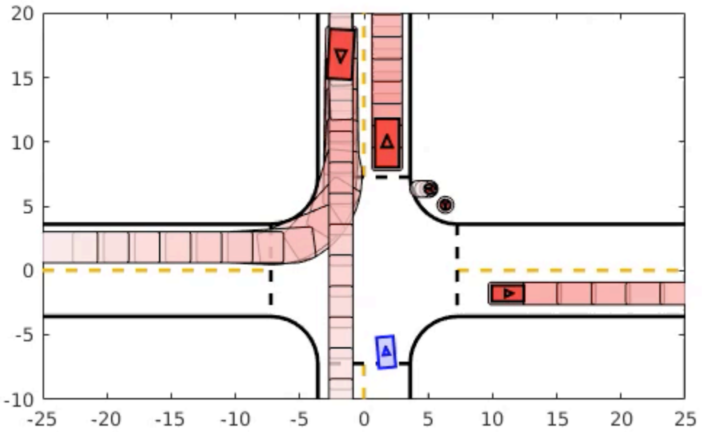


Theorem

Suppose the vehicle is convex, the maximum relative speed between any obstacle and the vehicle is bounded, then there exists a time discretization, which can be pre-computed offline, that generates an outerapproximation to the set of safe trajectory parameters.

Real-World, Car-Like Experiment

Comparison: EV Unprotected Left Turns



- 100 trials
- up to 4 cars, 2 pedestrians
- at-fault if stopped in intersection

Planner	At-fault Collisions (%)	Goals (%)	Average Time to Goal (s)	Average Speed (m/s)
Linear MPC	19.0	80.0	**14.9**	**3.35**
RTD	**0.00**	**99.0**	20.9	2.71

More Comparisons

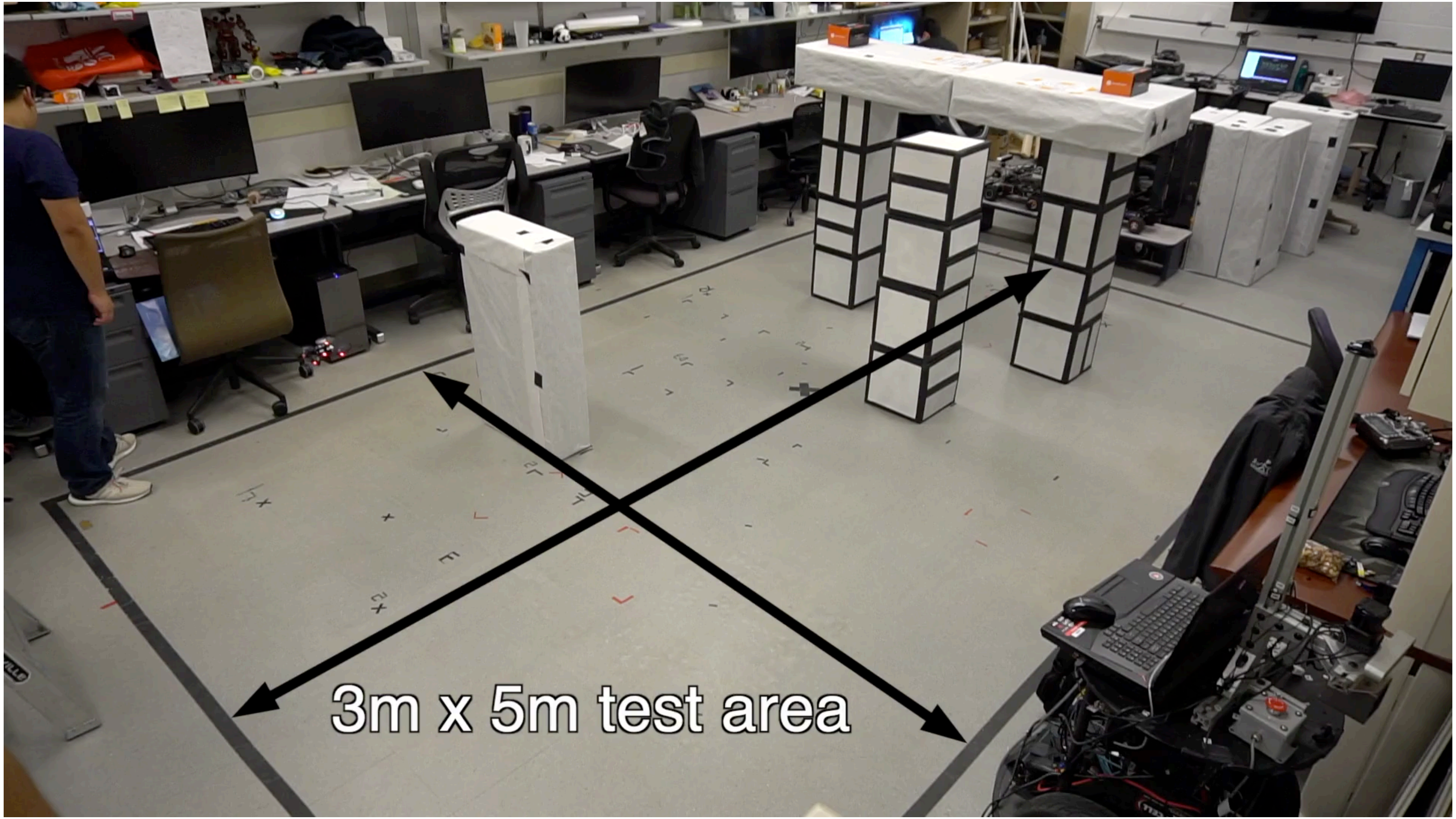
Other Method	Robot	Crash %	Goal %	RTD Goal %	
NMPC (GPOPS-II)	Segway	0.0	0.0	**87.4**	IJRR, in press
RRT (Kuwata et al., 2009)	Segway	6.3	77.2	**87.4**	
NMPC (GPOPS-II)	Rover	0.0	0.0	**93.1**	
RRT (Kuwata et al., 2009)	Rover	2.1	**97.4**	93.1	
NMPC (GPOPS-II)	Fusion	0.0	0.0	**93.1**	ACC ‘19
RRT (Kuwata et al., 2009)	Fusion	1.0	38.0	**100.0**	
State Lattice (McN., 2010)	Segway	7.9	92.4	**100.0**	RSS ‘19
State Lattice (McN., 2010)	EV	17.2	77.3	**90.7**	
Linear MPC	EV	19.0	80.0	**99.0**	ITSC ‘19
FasTrack (Herbert, 2017)	TurtleBot	0.0	85.3	**90.0**	
CHOMP (Zucker, 2013)	Fetch	18.0	82.0	**84.0**	RSS ‘20

Real-World, Ford Fusion

Outline

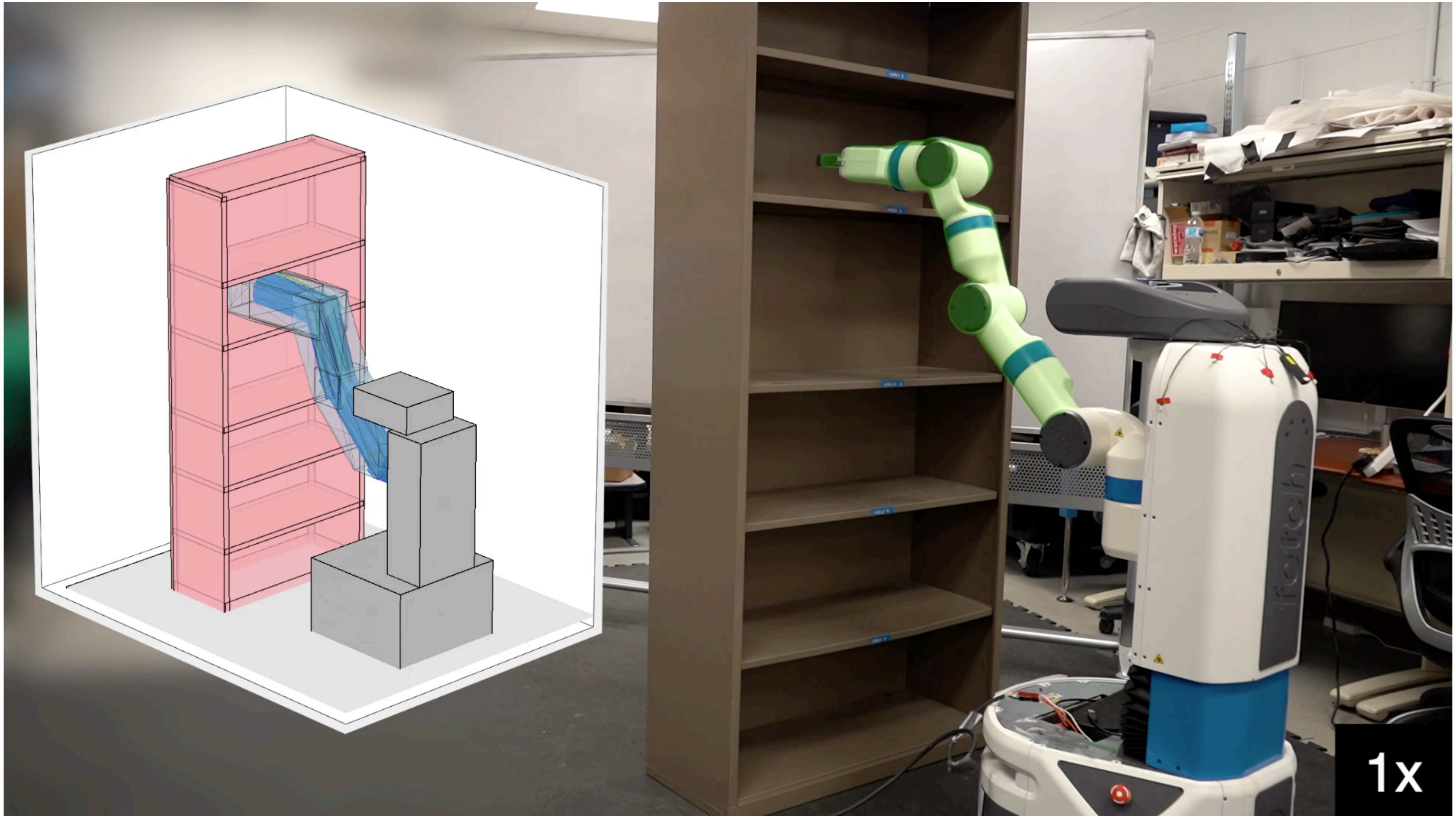
1. State of the Art in Verified AV Control
2. Reachability-based Trajectory Design for Provably Safe, Real-Time Planning for Autonomous Vehicles – STATIC
3. Reachability-based Trajectory Design for Provably Not-at-Fault, Real-Time Planning for Autonomous Vehicles – DYNAMIC
4. **Conclusion**

RTD – Quadrotor

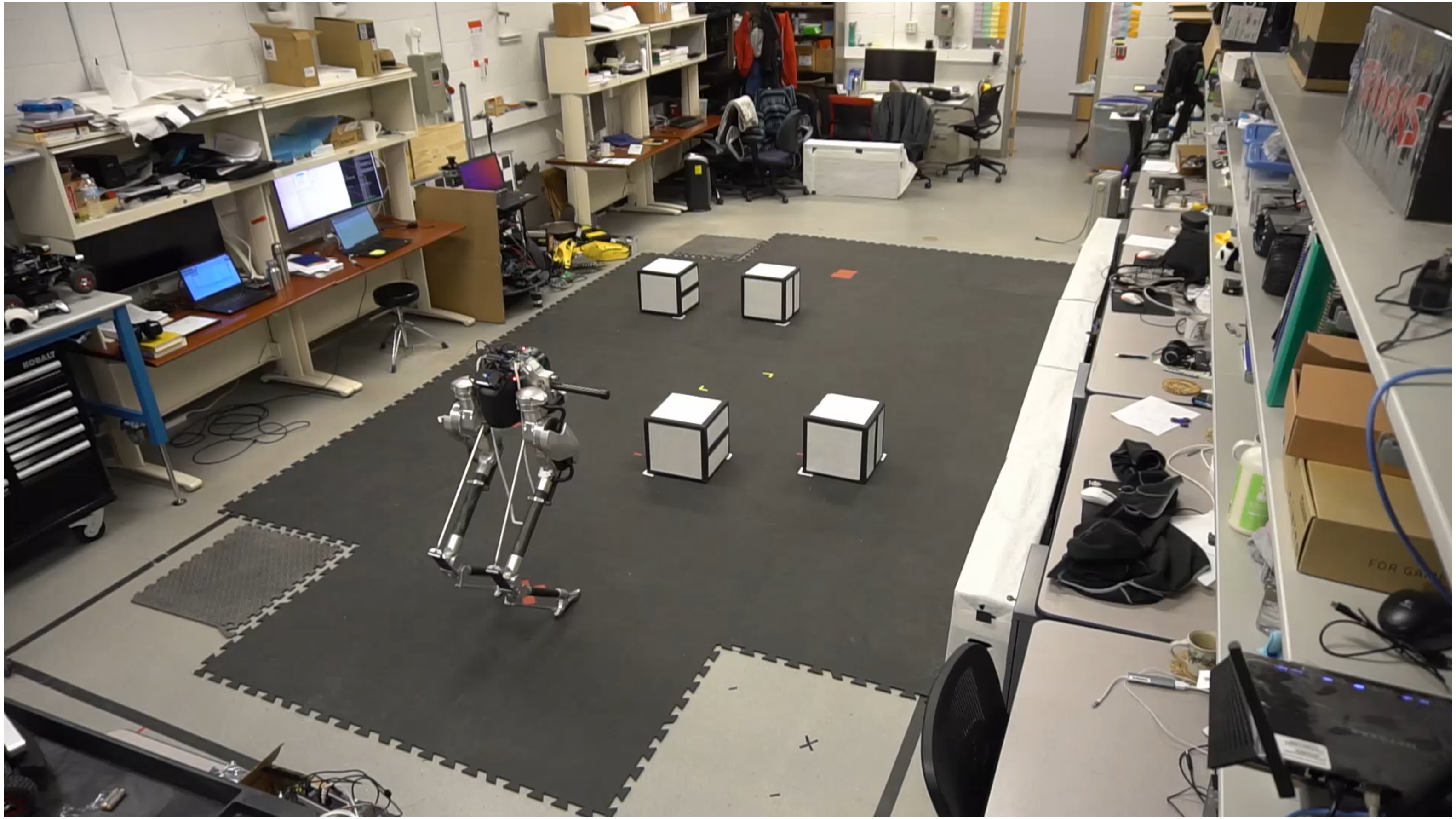


 https://youtu.be/1cldHVQK3Yw?t=94

RTD – High DOF Manipulator



RTD – Walking Robot



Takeaways and Challenges

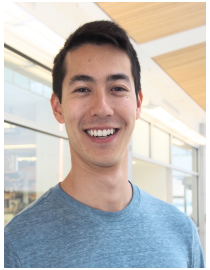
RTD enables collision-free/not-at-fault planning

Variety of hardware platforms and 10M+ simulations

How to verify perception and prediction algorithms?

How to incorporate models of uncertainty?

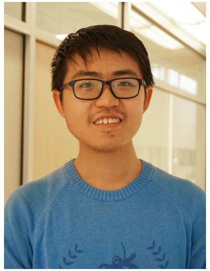
The Real Automators and Our Sponsors



Patrick Holmes

Shreyas Kousik

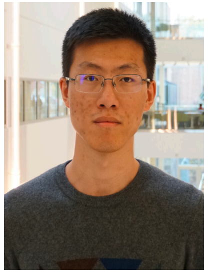
Hannah Larson



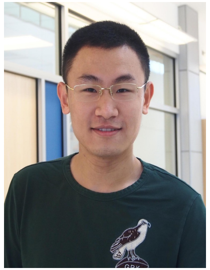
Jinsun Liu

Zehui Lu

Sean Vaskov



Bohao Zhang



Pengcheng Zhao

github.com/skousik/RTD_tutorial

github.com/ramvasudevan/arm_planning

github.com/skvaskov/RTD

Takeaways

RTD enables collision-free/not-at-fault planning

Variety of hardware platforms and 10M+ simulations

github.com/skousik/RTD_tutorial

github.com/ramvasudevan/arm_planning

github.com/skvaskov/RTD