Interactive Motion Planning for Autonomous Racing

Dr. Alexander Liniger

UPenn mLab - October 2020

ETH zürich

CVL Computer Vision Lab

Institut für Automatik
Automatic Control Laboratory

Motivation

Autonomous driving

- Active research area since the 1980s
  - Research done in industry and academia
  - Waymo/Google: > 20 mio miles
- Take safety critical decisions in an uncertain environment

Motivation

Autonomous driving

- Active research area since the 1980s
  - Research done in industry and academia
  - Waymo/Google: > 20 mio miles
- Take safety critical decisions in an uncertain environment

Autonomous racing

- Drive as fast as possible around a track
  - Miniature race car set-up using RC cars
  - Formula Student Driverless
  - Roborace
- **Structured** but **competitive** environment

Motivation

Autonomous driving

- Active research area since the 1980s
  - Research done in industry and academia
  - Waymo/Google: > 20 mio miles
- Take safety critical decisions in an uncertain environment

Autonomous racing

- Drive as fast as possible around a track
  - Miniature race car set-up using RC cars
  - Formula Student Driverless
  - Roborace
- **Structured** but **competitive** environment

Motivation

Autonomous driving

- Active research area since the 1980s
  - Research done in industry and academia
  - Waymo/Google: > 20 mio miles
- Take safety critical decisions in an uncertain environment

Autonomous racing

- Drive as fast as possible around a track
  - Miniature race car set-up using RC cars
  - Formula Student Driverless
  - Roborace
- **Structured** but **competitive** environment

Autonomous Racing Challenges

Autonomous Racing Challenges

Driving at the handling limit

- If we do not drive at the limit we drive to slow
- Motion planning for a highly nonlinear system

Liniger, Domahidi & Morari OCAM 15, Liniger & Lygeros T-CST 17

Autonomous Racing Challenges

Driving at the handling limit

- If we do not drive at the limit we drive to slow
- Motion planning for a highly nonlinear system

Liniger, Domahidi & Morari OCAM 15, Liniger & Lygeros T-CST 17

Staying safe inside the track

- If we crash we lose!
- Infinite horizon constraint satisfaction

Liniger & Lygeros HSCC 15, Liniger & Lygeros T-CST 17

Autonomous Racing Challenges

Driving at the handling limit

- If we do not drive at the limit we drive to slow
- Motion planning for a highly nonlinear system

Liniger, Domahidi & Morari OCAM 15, Liniger & Lygeros T-CST 17

Staying safe inside the track

- If we crash we lose!
- Infinite horizon constraint satisfaction

Liniger & Lygeros HSCC 15, Liniger & Lygeros T-CST 17

Interact with other race cars

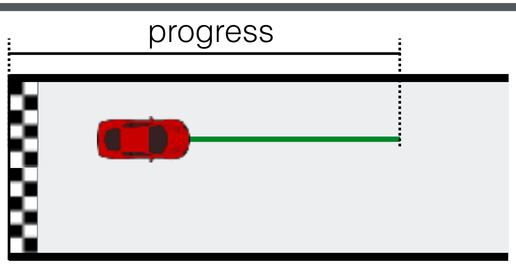
- The art of overtaking and interacting with other cars
- Decision making in a highly dynamical non-cooperative environment

Liniger & Lygeros T-CST 20

Racing Ingredients

Finish first

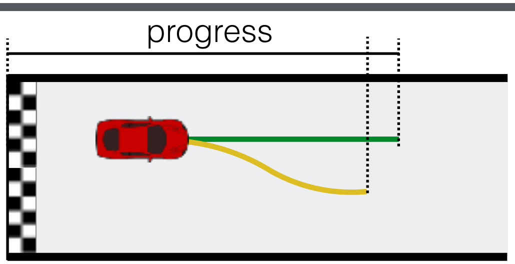
- Approximated by maximizing progress
- Generates racing trajectories



Racing Ingredients

Finish first

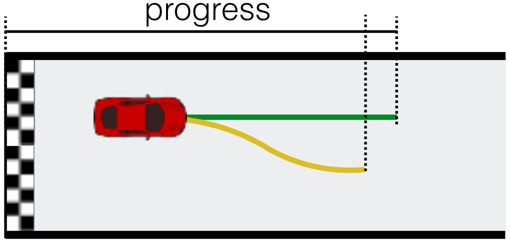
- Approximated by maximizing progress
- Generates racing trajectories



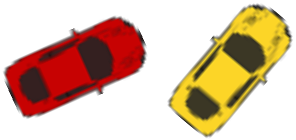
Racing Ingredients

Finish first

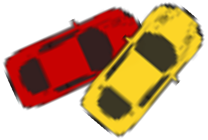
- Approximated by maximizing progress
- Generates racing trajectories



Do not collide with other cars



good

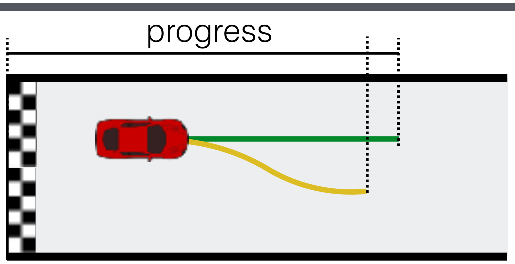


bad

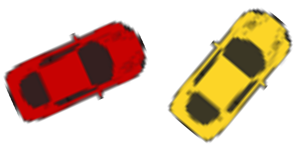
Racing Ingredients

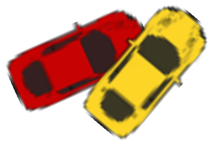
Finish first

- Approximated by maximizing progress
- Generates racing trajectories

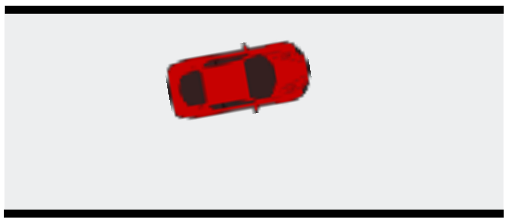


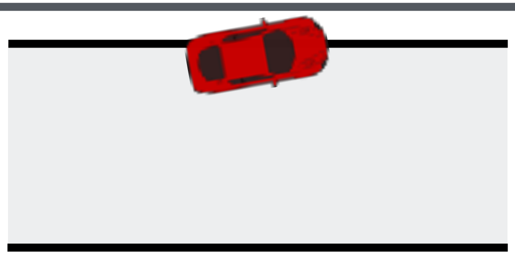
Do not collide with other cars


good


bad

Stay inside the track


good


bad

Experimental Set-Up

IR Camera
System

Ethernet

Controller
Linux PC

Bluetooth

1:43 miniature
RC race cars

Experimental Set-Up

IR Camera
System

Ethernet

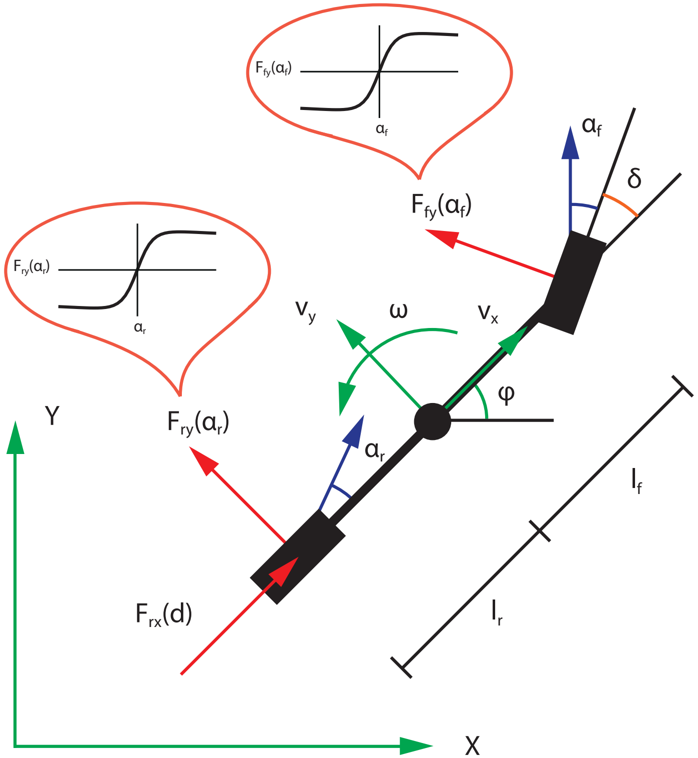
Controller
Linux PC

Bluetooth

1:43 miniature
RC race cars

Car Model

- Bicycle model, with nonlinear lateral tire forces (Pacejka)

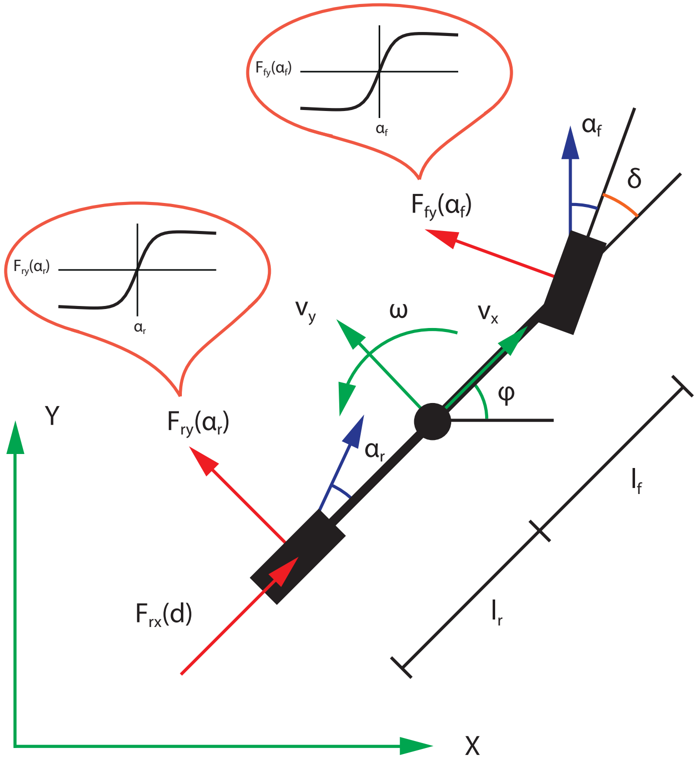


$$\dot{X} = v_x \cos(\varphi) - v_y \sin(\varphi)$$
$$\dot{Y} = v_x \sin(\varphi) + v_y \cos(\varphi)$$
$$\dot{\varphi} = \omega$$
$$\dot{v}_x = \frac{1}{m}(F_{r,x} - F_{f,y} \sin\delta + m v_y \omega)$$
$$\dot{v}_y = \frac{1}{m}(F_{r,y} + F_{f,y} \cos\delta - m v_x \omega)$$
$$\dot{\omega} = \frac{1}{I_z}(F_{f,y} l_f \cos\delta - F_{r,y} l_r)$$

- Highly nonlinear 6 dimensional system

Car Model

- Bicycle model, with nonlinear lateral tire forces (Pacejka)

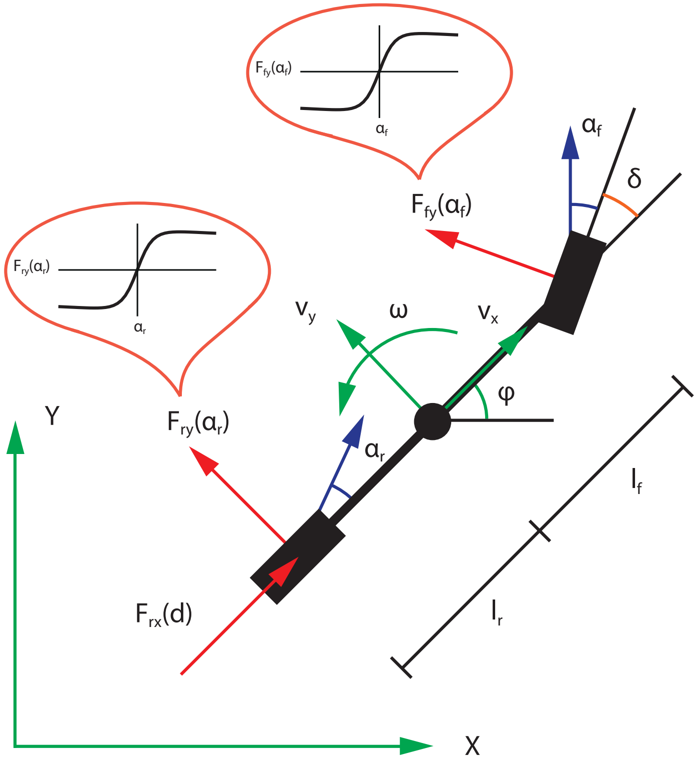


$$\dot{X} = v_x \cos(\varphi) - v_y \sin(\varphi)$$
$$\dot{Y} = v_x \sin(\varphi) + v_y \cos(\varphi)$$
$$\dot{\varphi} = \omega$$
$$\dot{v}_x = \frac{1}{m}(F_{r,x} - F_{f,y} \sin\delta + m v_y \omega)$$
$$\dot{v}_y = \frac{1}{m}(F_{r,y} + F_{f,y} \cos\delta - m v_x \omega)$$
$$\dot{\omega} = \frac{1}{I_z}(F_{f,y} l_f \cos\delta - F_{r,y} l_r)$$

- Highly nonlinear 6 dimensional system
- Separation is slow and fast dynamics

Car Model

- Bicycle model, with nonlinear lateral tire forces (Pacejka)



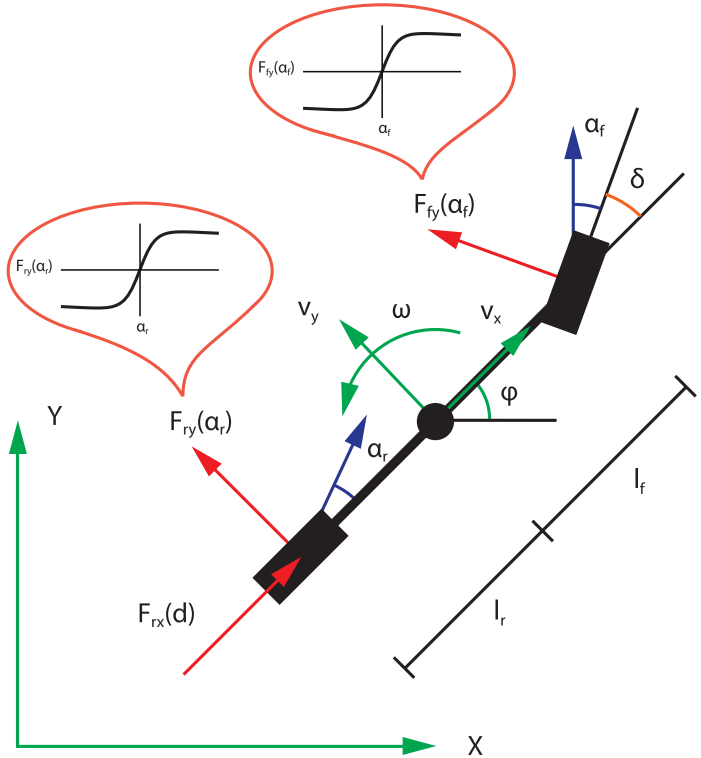
$$\dot{X} = v_x \cos(\varphi) - v_y \sin(\varphi)$$
$$\dot{Y} = v_x \sin(\varphi) + v_y \cos(\varphi)$$
$$\dot{\varphi} = \omega$$

$$\dot{v}_x = \frac{1}{m}(F_{r,x} - F_{f,y} \sin\delta + m v_y \omega)$$
$$\dot{v}_y = \frac{1}{m}(F_{r,y} + F_{f,y} \cos\delta - m v_x \omega)$$
$$\dot{\omega} = \frac{1}{I_z}(F_{f,y} l_f \cos\delta - F_{r,y} l_r)$$

- Highly nonlinear 6 dimensional system
- Separation is slow and fast dynamics

Car Model

- Bicycle model, with nonlinear lateral tire forces (Pacejka)



$$\dot{X} = v_x \cos(\varphi) - v_y \sin(\varphi)$$
$$\dot{Y} = v_x \sin(\varphi) + v_y \cos(\varphi)$$
$$\dot{\varphi} = \omega$$

$$\dot{v}_x = \frac{1}{m}(F_{r,x} - F_{f,y}\sin\delta + m v_y \omega)$$
$$\dot{v}_y = \frac{1}{m}(F_{r,y} + F_{f,y}\cos\delta - m v_x \omega)$$
$$\dot{\omega} = \frac{1}{I_z}(F_{f,y} l_f \cos\delta - F_{r,y} l_r)$$

- Highly nonlinear 6 dimensional system
- Separation is slow and fast dynamics

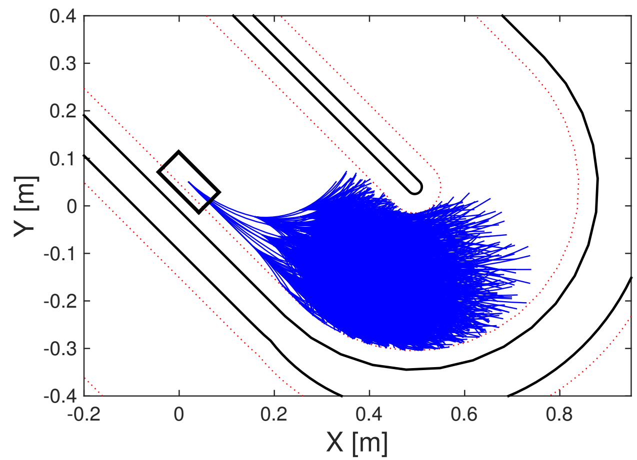
Hierarchical Control Structure

Path planning based on constant velocities primitives

- Plan for slow dynamics
- Reduced dimension
- Long discretization times

MPC-based trajectory tracking

- Considering full dynamical bicycle model
- Linearization points given by path planner



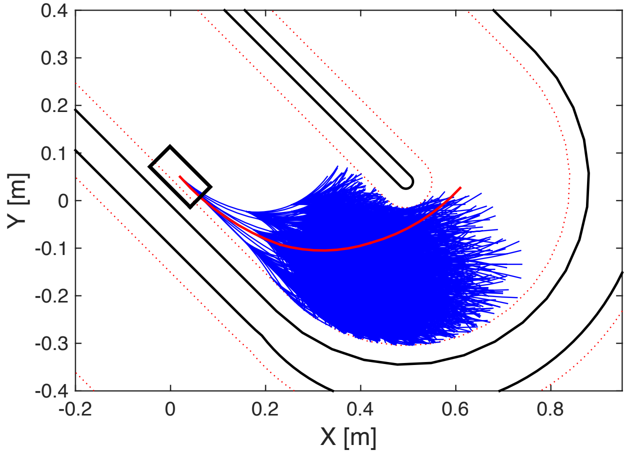
Hierarchical Control Structure

Path planning based on constant velocities primitives

- Plan for slow dynamics
- Reduced dimension
- Long discretization times

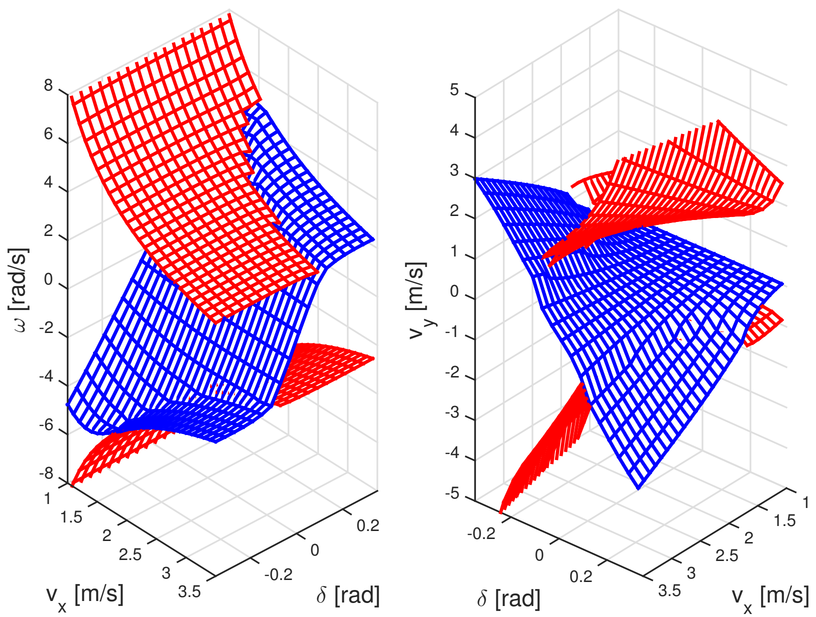
MPC-based trajectory tracking

- Considering full dynamical bicycle model
- Linearization points given by path planner



Constant Velocities

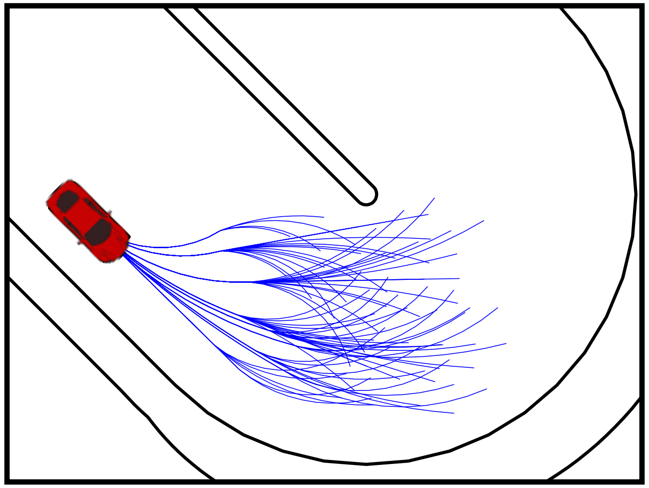
- Velocities (v_x,v_y,ω) “always” at steady state
- Find points where (v_x,v_y,ω) are constant



- Gridding stationary velocity points
- Library of possible movements (Motion Primitives)
- Low dimensional grid (~100) can capture the whole system

Path Planning Model

- Library of constant velocity “primitives”
- Assumptions:
  - New constant velocity can be reached immediately
  - Stay at the constant velocity for a fix time period T_{pp}
  - Transition between constant velocity are restricted $u_k \in \mathcal{U}(q_k)$



$$X_{k+1} = X_k + \int_0^{T_{pp}} \bar{v}_x(u_k)\cos(\varphi) - \bar{v}_y(u_k)\sin(\varphi)dt$$

$$Y_{k+1} = Y_k + \int_0^{T_{pp}} \bar{v}_x(u_k)\sin(\varphi) + \bar{v}_y(u_k)\cos(\varphi)dt$$

$$\varphi_{k+1} = \varphi_k + \int_0^{T_{pp}} \bar{\omega}(u_k)dt$$

$$q_{k+1} = u_k$$

- Discrete time dynamical system: $x_{k+1} = f(x_k, u_k) \quad u_k \in \mathcal{U}(q_k)$

Path Planning Algorithm

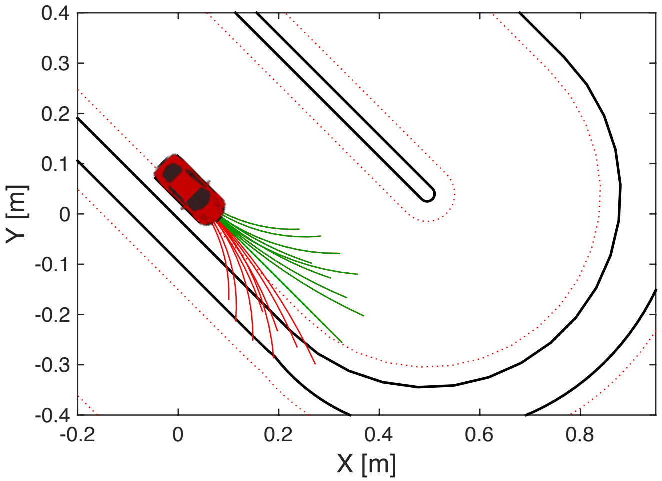
Breath-First Path Generation

$$\max_{u,x} \quad p(x_N)$$

$$\text{s.t.} \quad x_0 = x$$

$$x_{k+1} = f(x_k, u_k)\,, \qquad u_k \in \mathcal{U}(x_k)$$

$$x_k \in K, \qquad k = 1, \ldots, N$$



Path Planning Algorithm

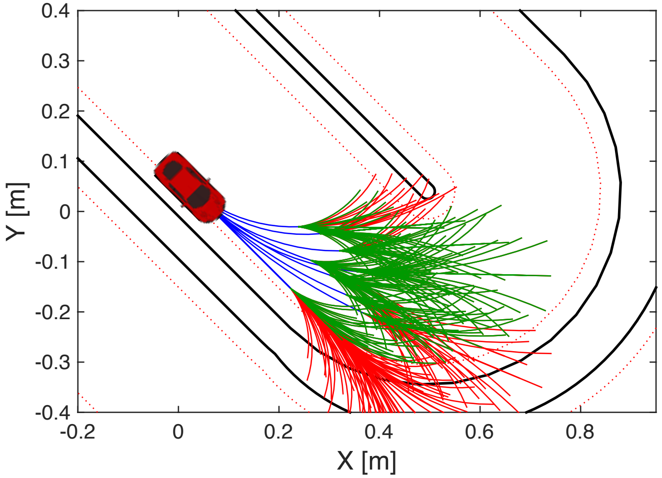
Breath-First Path Generation

$$\max_{u,x} \quad p(x_N)$$

$$\text{s.t.} \quad x_0 = x$$

$$x_{k+1} = f(x_k, u_k), \qquad u_k \in \mathcal{U}(x_k)$$

$$x_k \in K, \qquad k = 1, \ldots, N$$



Path Planning Algorithm

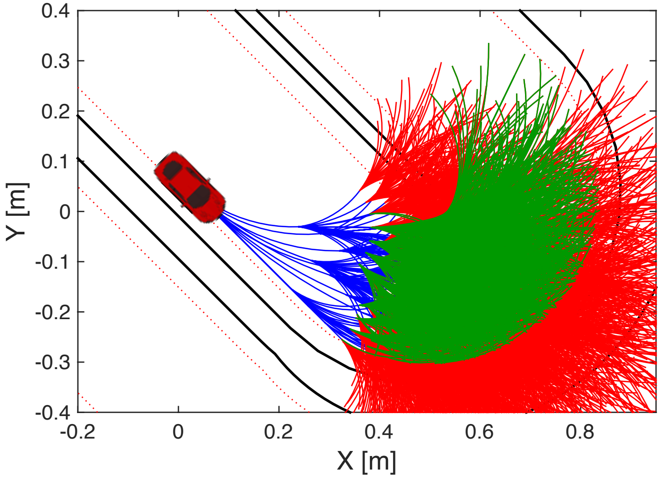
Breath-First Path Generation

$$\max_{u,x} \quad p(x_N)$$

$$\text{s.t.} \quad x_0 = x$$

$$x_{k+1} = f(x_k, u_k), \quad u_k \in \mathcal{U}(x_k)$$

$$x_k \in K, \quad k = 1, \ldots, N$$



Path Planning Algorithm

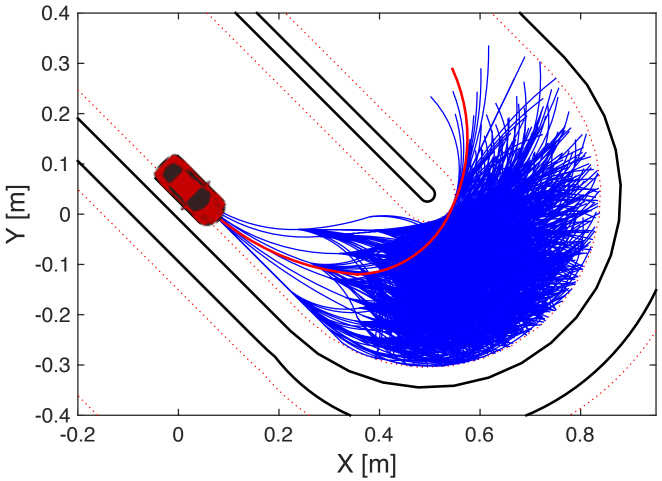
Breath-First Path Generation

$$\max_{u,x} \quad p(x_N)$$

$$\text{s.t.} \quad x_0 = x$$

$$x_{k+1} = f(x_k, u_k), \quad u_k \in \mathcal{U}(x_k)$$

$$x_k \in K, \quad k = 1, \ldots, N$$



Path Planning Algorithm

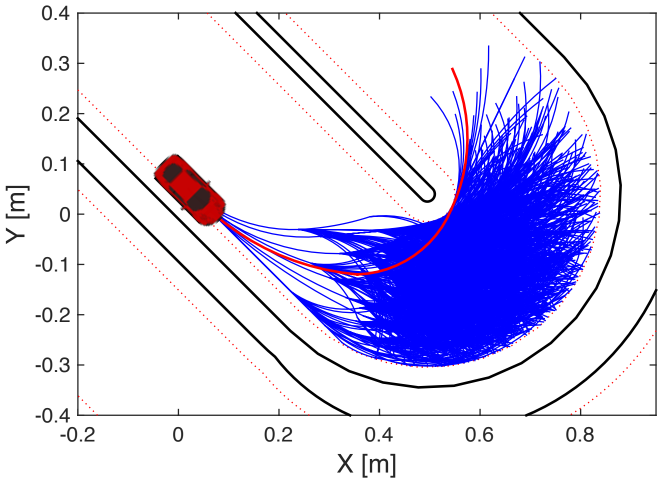
Breath-First Path Generation

$$\max_{u,x} \quad p(x_N)$$

$$\text{s.t.} \quad x_0 = x$$

$$x_{k+1} = f(x_k, u_k)\,, \qquad u_k \in \mathcal{U}(x_k)$$

$$x_k \in K, \qquad\qquad k = 1, \ldots, N$$

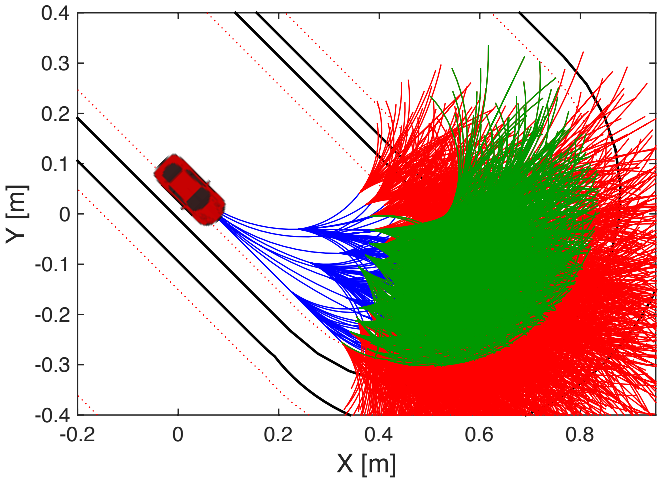


- Tree grows exponentially in the horizon
- Time to check track constraints is the bottle neck
- Optimal trajectory often not recursive feasible/viable

Path Planning Algorithm

Breath-First Path Generation

$$
\begin{aligned}
\max_{u,x} \quad & p(x_N) \\
\text{s.t.} \quad & x_0 = x \\
& x_{k+1} = f(x_k, u_k), \qquad u_k \in \mathcal{U}(x_k) \\
& x_k \in K, \qquad\qquad\quad k = 1, \dots, N
\end{aligned}
$$



- Tree grows exponentially in the horizon
- Time to check track constraints is the bottle neck
- Optimal trajectory often not recursive feasible/viable

Path Planning Algorithm

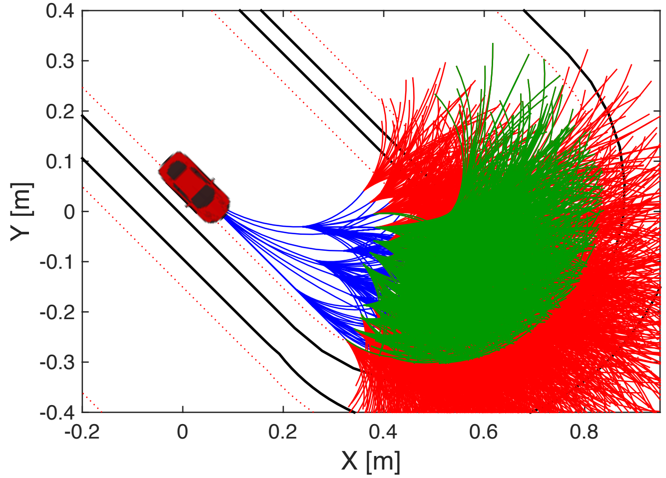
Breath-First Path Generation

$$\max_{u,x} \quad p(x_N)$$

$$\text{s.t.} \quad x_0 = x$$

$$x_{k+1} = f(x_k, u_k)\,, \qquad u_k \in \mathcal{U}(x_k)$$

$$x_k \in K, \qquad k = 1, \dots, N$$



- Tree grows exponentially in the horizon
- Time to check track constraints is the bottle neck
- Optimal trajectory often not recursive feasible/viable

Can we only generate safe trajectories

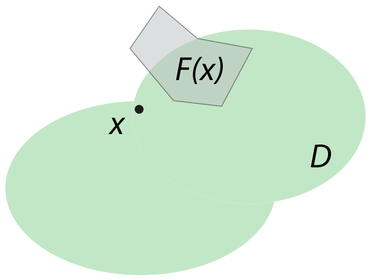
Viability Theory

- Given:
  - A difference inclusion $x_{k+1} \in F(x_k) = \{f(x_k, u_k) \mid u_k \in \mathcal{U}\}$
  - $K \subset \mathbb{R}^n$ is a compact set
- A solution is viable if: $$\begin{cases} x_{k+1} \in F(x_k)\,, & \forall k \geq 0 \\ x_0 \in K & \\ x_k \in K\,, & \forall k \geq 0 \end{cases}$$

Definition [Saint-Pierre 94]

Let $F : \mathbb{R}^n \to \mathbb{R}^n$ be a set valued map. Then a closed subset $D \subset \mathbb{R}^n$ is a viability domain of F if:

$$\forall x \in D, \quad F(x) \cap D \neq \emptyset$$



- The viability kernel $Viab_F(K)$, is the largest closed viability domain contained in K

Viability Theory

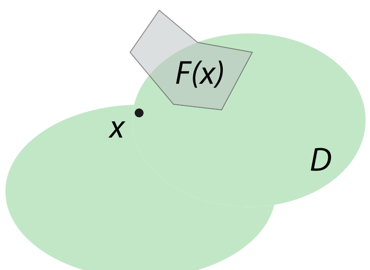
- Given:
  - A difference inclusion $x_{k+1} \in F(x_k) = \{f(x_k, u_k) \mid u_k \in \mathcal{U}\}$
  - $K \subset \mathbb{R}^n$ is a compact set
- A solution is viable if:

$$\begin{cases} x_{k+1} \in F(x_k)\,, & \forall k \geq 0 \\ x_0 \in K & \\ x_k \in K\,, & \forall k \geq 0 \end{cases}$$

Definition [Saint-Pierre 94]

Let $F : \mathbb{R}^n \to \mathbb{R}^n$ be a set valued map. Then a closed subset $D \subset \mathbb{R}^n$ is a viability domain of F if:

$$\forall x \in D, \quad F(x) \cap D \neq \emptyset$$



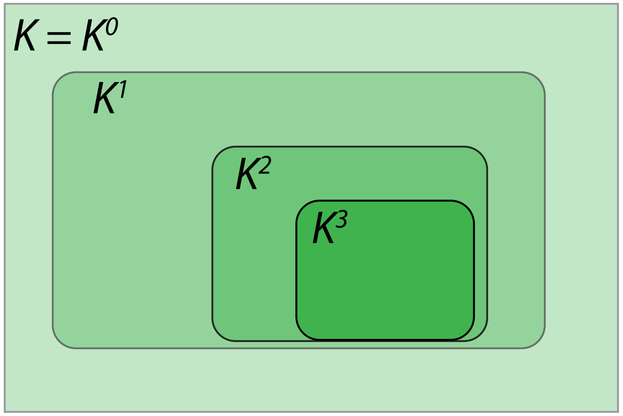
A viable trajectory is a safe trajectory

contained in K

Viability Kernel Algorithm

- Given:
  - A discrete difference inclusion $x_{k+1} \in F(x_k)$
  - $K \subset \mathbb{R}^n$ is a compact set
- Construction of *Viab*$_F$*(K)*:
  - Sequence of nested subsets

$$K^0 = K$$
$$K^{n+1} = \{x \in K^n \mid F(x) \cap K^n \neq \emptyset\}$$



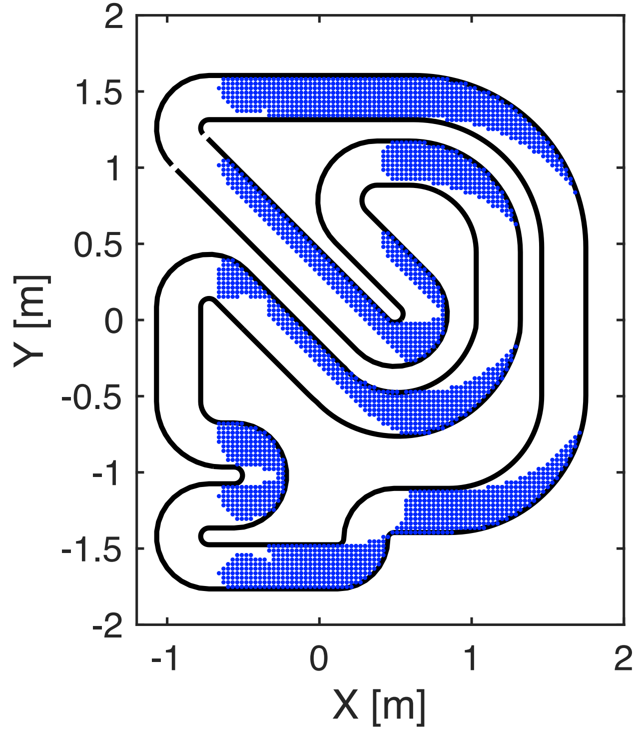
Proposition [Saint-Pierre 94]

Let $F : \mathbb{R}^n \to \mathbb{R}^n$ be a upper-semicontinuous set-valued map with closed values and let K be a compact subset of *Dom(F)*

$$Viab_F(K) = \bigcap_{n=0}^{\infty} K^n$$

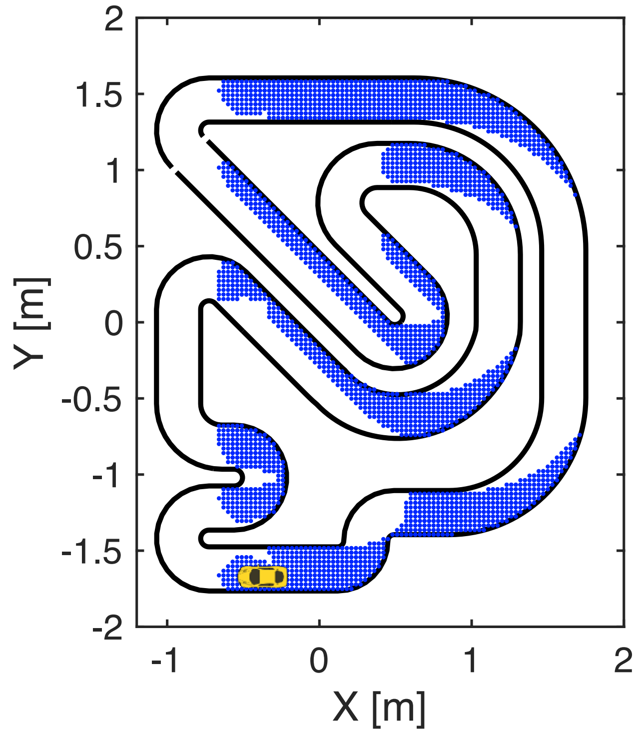
Viability Kernel

- Viability kernel can be computed by discretizing the state-space
- $F(x_k)$ given by the path planning model
  - Sample-data system viability kernel algorithm
- K given by the track constraints



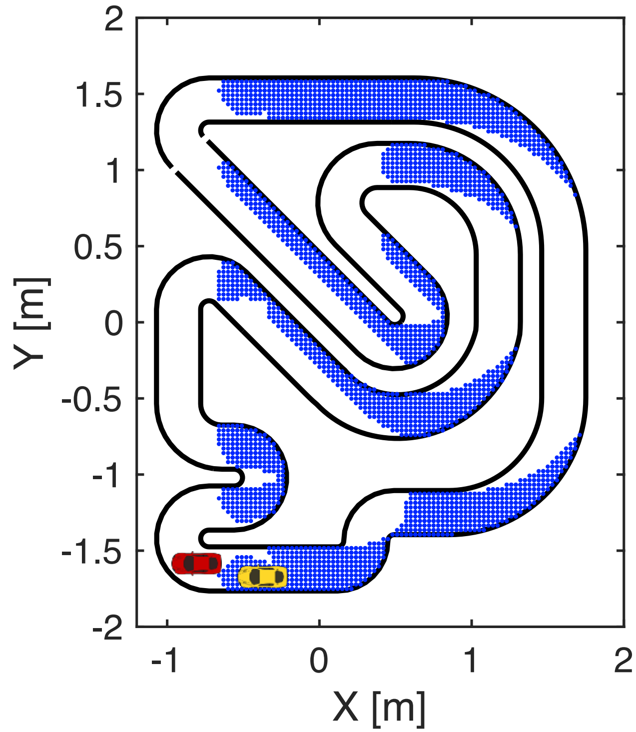
Viability Kernel

- Viability kernel can be computed by discretizing the state-space
- $F(x_k)$ given by the path planning model
  - Sample-data system viability kernel algorithm
- K given by the track constraints



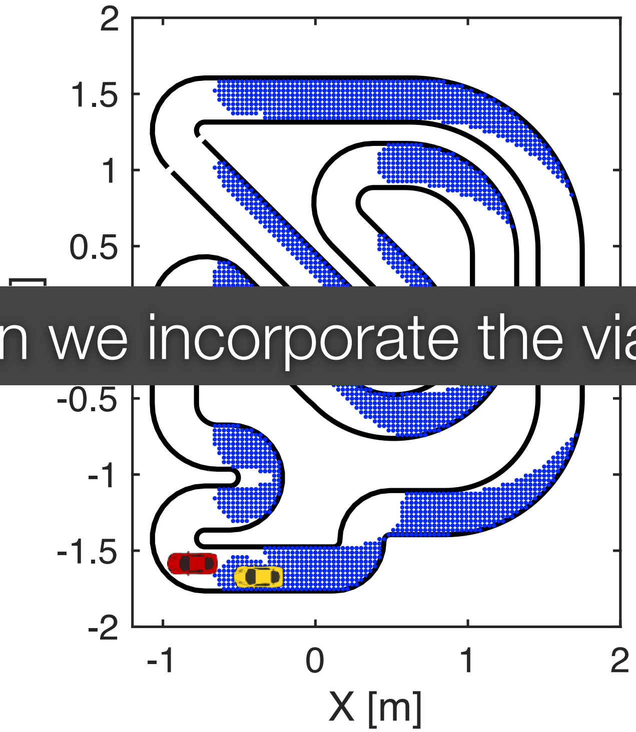
Viability Kernel

- Viability kernel can be computed by discretizing the state-space
- $F(x_k)$ given by the path planning model
  - Sample-data system viability kernel algorithm
- K given by the track constraints



Viability Kernel

- Viability kernel can be computed by discretizing the state-space
- $F(x_k)$ given by the path planning model
  - Sample-data system viability kernel algorithm
- K given by the track constraints

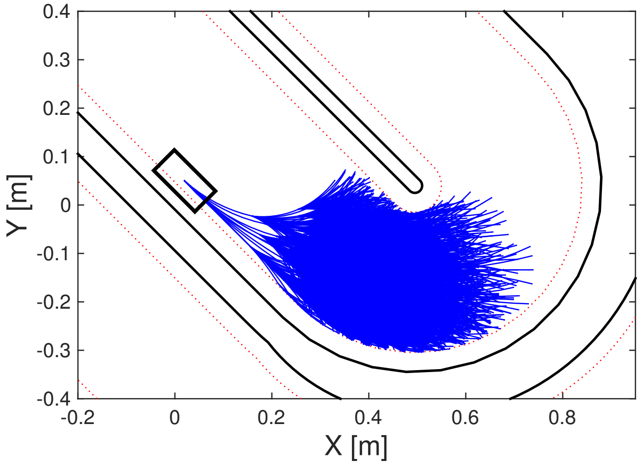


How can we incorporate the viability kernel

Viability Constraints

- Imposing viability constraints in the path planning problem

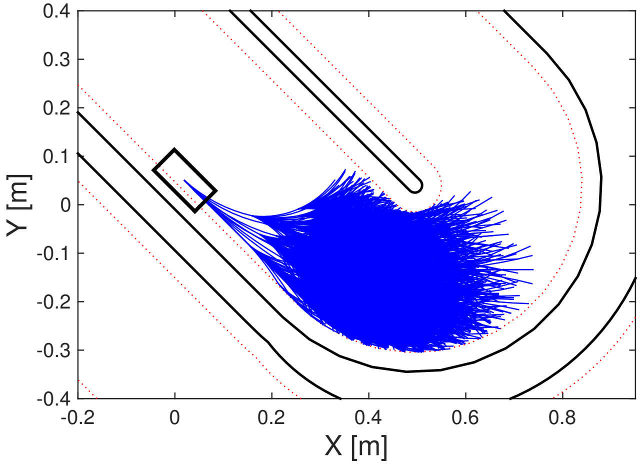
$$\max_{u,x} \quad p(x_N)$$
$$\text{s.t.} \quad x_0 = x$$
$$x_{k+1} = f(x_k, u_k), \quad u_k \in \mathcal{U}(x_k)$$
$$x_k \in K, \quad k = 1, \ldots, N$$



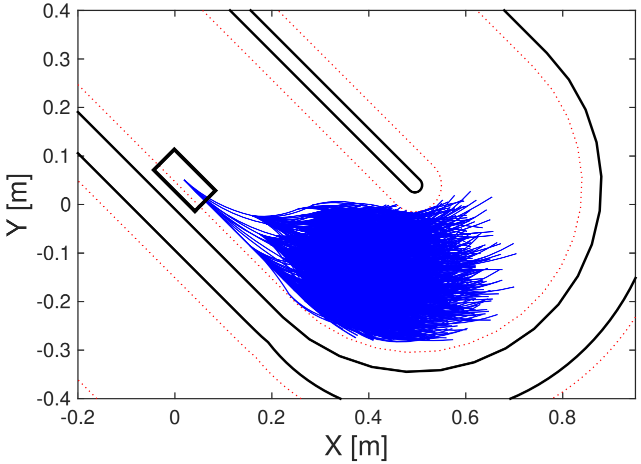
Viability Constraints

- Imposing viability constraints in the path planning problem

$$\begin{aligned} \max_{u,x} \quad & p(x_N) \\ \text{s.t.} \quad & x_0 = x \\ & x_{k+1} = f(x_k, u_k), \quad u_k \in \mathcal{U}(x_k) \\ & x_k \in K, \quad k = 1, \dots, N \end{aligned}$$

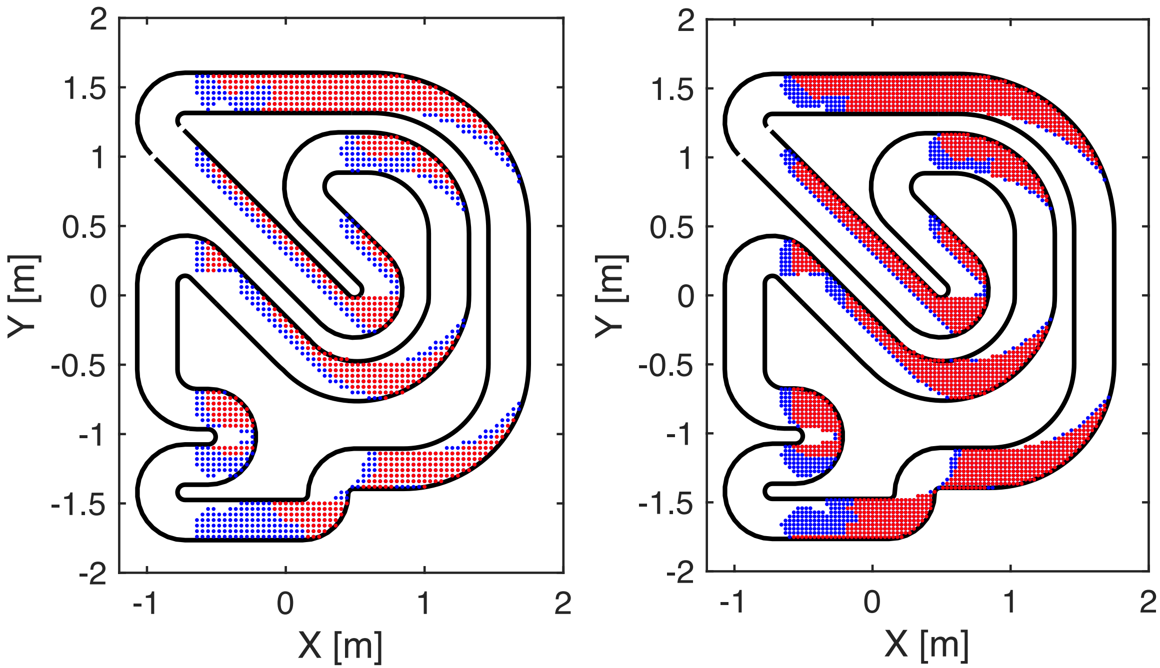


$$\begin{aligned} \max_{u,x} \quad & p(x_N) \\ \text{s.t.} \quad & x_0 = x \\ & x_{k+1} = f(x_k, u_k), \quad u_k \in \mathcal{U}(x_k) \\ & x_k \in Viab_F(K), \quad k = 1, \dots, N \end{aligned}$$



Discriminating Kernel

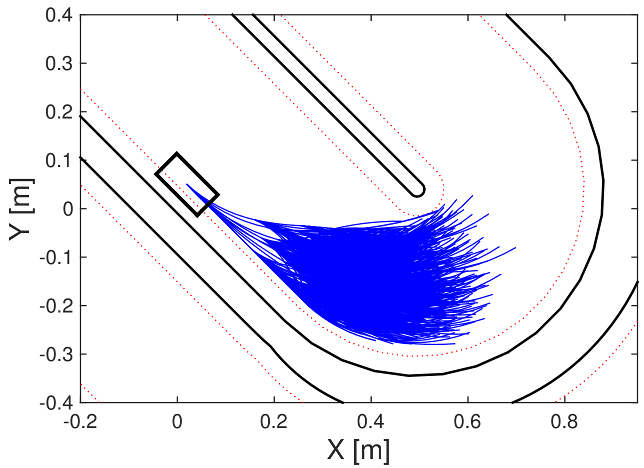
- Discretizing the state-space does introduce errors
- Errors can be modeled as an adversarial player
- Depending on grid size and Lipschitz constant
- Discriminating kernel —> gamified viability kernel



Discriminating Kernel

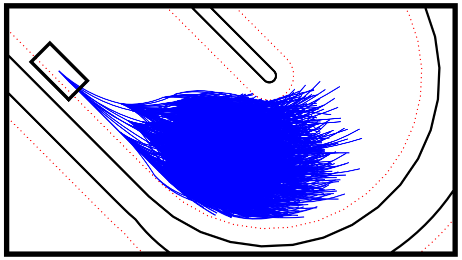
- Discretizing the state-space does introduce errors
- Errors can be modeled as an adversarial player
- Depending on grid size and Lipschitz constant
- Discriminating kernel —> gamified viability kernel

$$
\begin{aligned}
\max_{u,x} \quad & p(x_N) \\
\text{s.t.} \quad & x_0 = x \\
& x_{k+1} = f(x_k, u_k)\,, \quad u_k \in \mathcal{U}(x_k) \\
& f(x_k, u_k) \in Disc_G(K),\ k = 0, \ldots, N-1
\end{aligned}
$$



Simulation Results

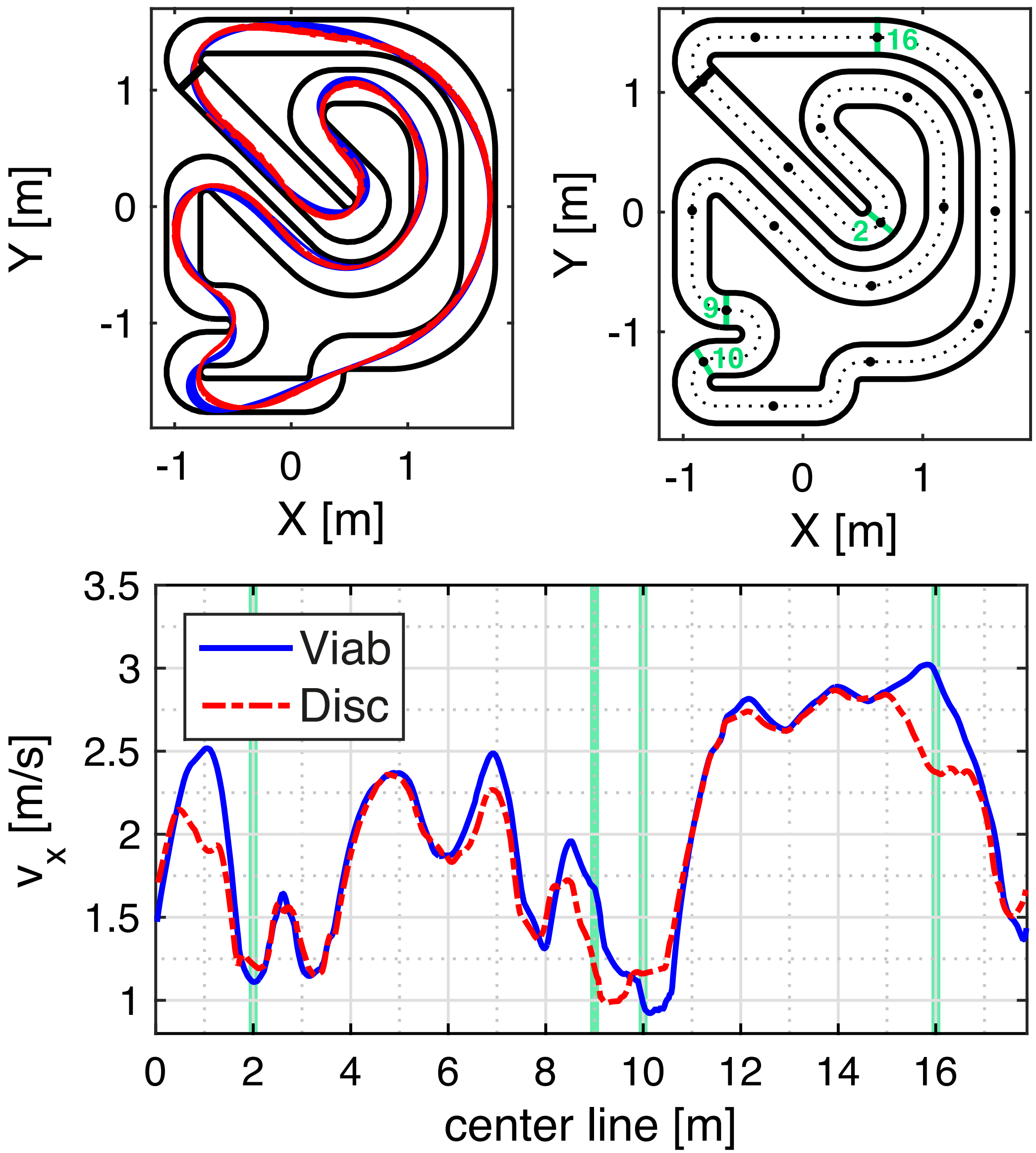
- Every 20 ms redo path planning and MPC step
- Simulation using full non-linear model
- Based on sensitivity study we determined
  - $T_{pp} = 0.16$ s
  - $N = 3$
  - $N_M = 129$



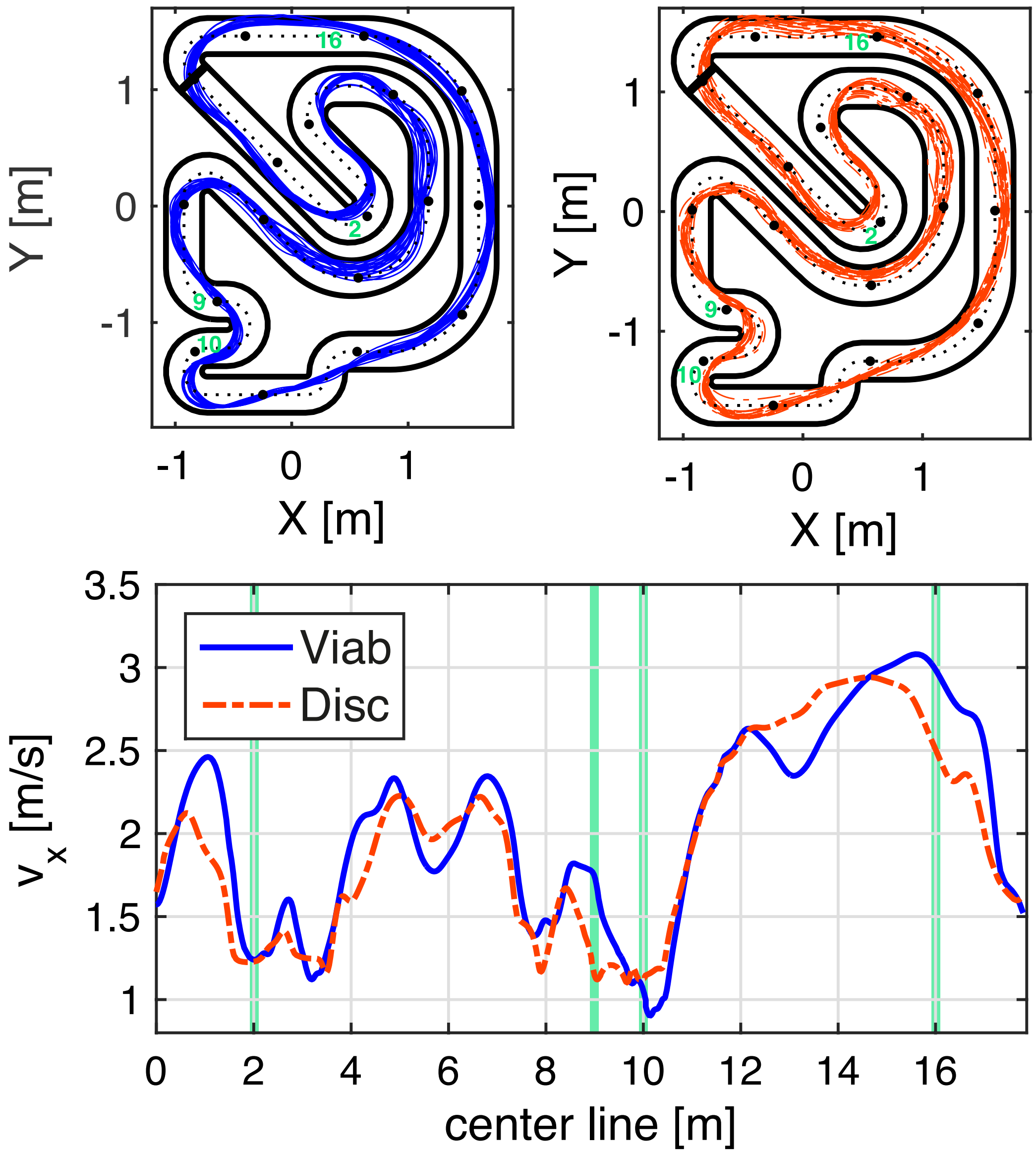
- Comparing: Viability vs Discrimination vs no kernel

Kernel	mean lap time [s]	# constr. violations	median comp. time [ms]	max comp. time [ms]
No	8.76	4	32.26	247.7
Viab	8.57	0	0.904	7.968
Disc	8.60	1	0.870	7.533

Simulation Results - Viab vs Disc



Experimental Results



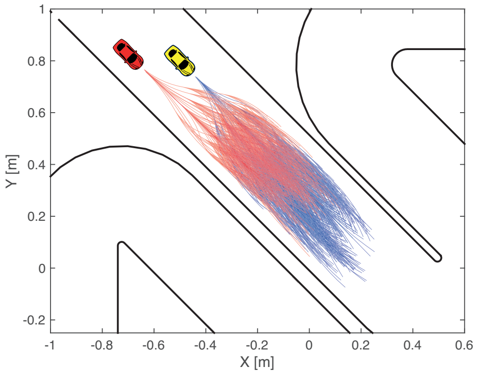
Experimental Results

Experimental Results

Experimental Results

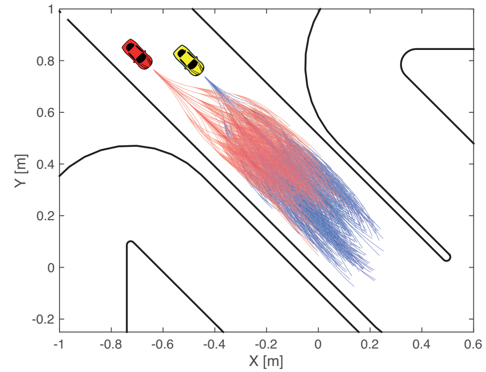
Bimatrix Racing Games

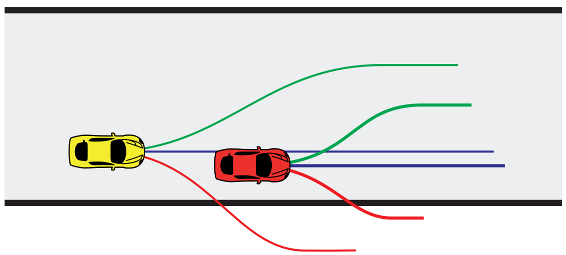
- Every trajectory is an action of a car
  - Each trajectory has a payoff
  - Payoff depends on actions of both cars



Bimatrix Racing Games

- Every trajectory is an action of a car
  - Each trajectory has a payoff
  - Payoff depends on actions of both cars

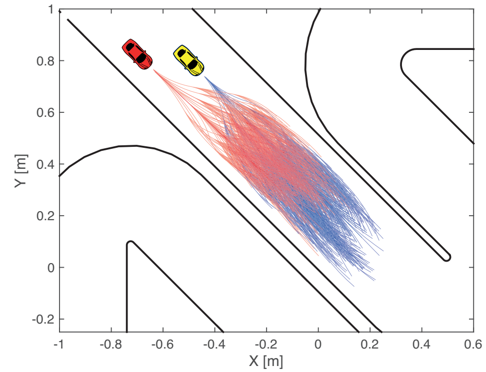


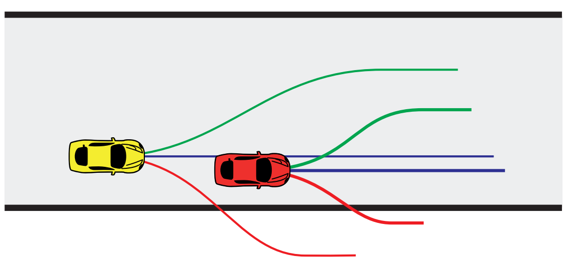


$$A = \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,1} & a_{2,2} & a_{2,3} \\ a_{3,1} & a_{3,2} & a_{3,3} \end{bmatrix} \quad B = \begin{bmatrix} b_{1,1} & b_{1,2} & b_{1,3} \\ b_{2,1} & b_{2,2} & b_{2,3} \\ b_{3,1} & b_{3,2} & b_{3,3} \end{bmatrix}$$

Bimatrix Racing Games

- Every trajectory is an action of a car
  - Each trajectory has a payoff
  - Payoff depends on actions of both cars



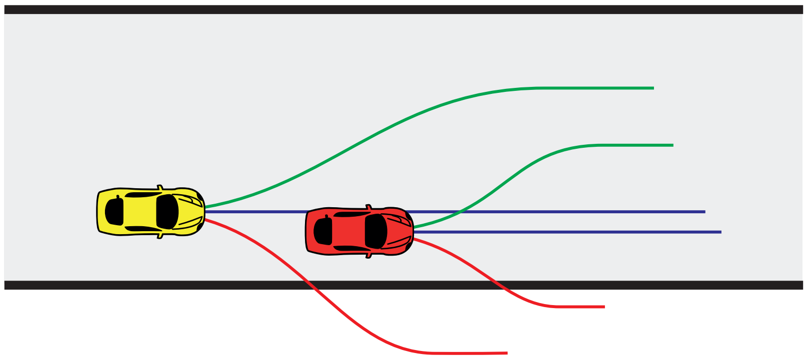


$$A = \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,1} & a_{2,2} & a_{2,3} \\ a_{3,1} & a_{3,2} & a_{3,3} \end{bmatrix} \quad B = \begin{bmatrix} b_{1,1} & b_{1,2} & b_{1,3} \\ b_{2,1} & b_{2,2} & b_{2,3} \\ b_{3,1} & b_{3,2} & b_{3,3} \end{bmatrix}$$

- The leader is always the car which is ahead at the beginning
- A trajectory pair is feasible if:
  - Trajectories stay inside the track and do not collide

Three Racing Games

- Sequential Game
- Cooperative Game
- Blocking Game



$$A = \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,1} & a_{2,2} & a_{2,3} \\ a_{3,1} & a_{3,2} & a_{3,3} \end{bmatrix}$$

$$B = \begin{bmatrix} b_{1,1} & b_{1,2} & b_{1,3} \\ b_{2,1} & b_{2,2} & b_{2,3} \\ b_{3,1} & b_{3,2} & b_{3,3} \end{bmatrix}$$

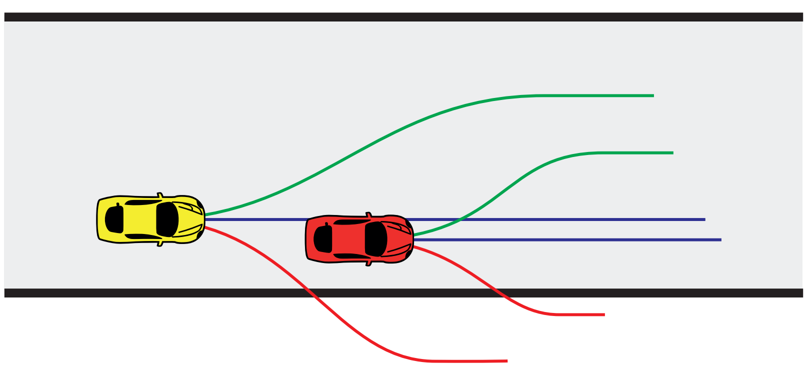
Three Racing Games

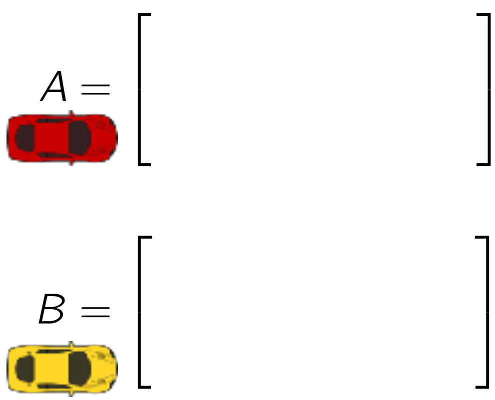
Sequential Game

- Exploiting the leader-follower structure
  - Low payoff if a trajectory leaves the track
  - Progress payoff if a trajectory is inside the track
  - Low payoff for the **follower** if trajectories collide

Cooperative Game

Blocking Game





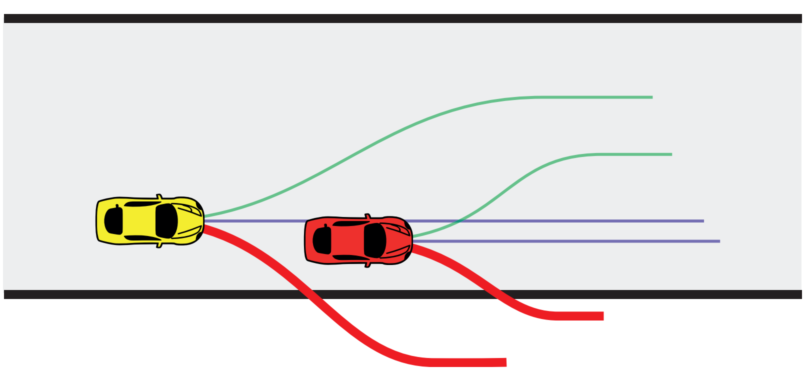
Three Racing Games

Sequential Game

- Exploiting the leader-follower structure
  - Low payoff if a trajectory leaves the track
  - Progress payoff if a trajectory is inside the track
  - Low payoff for the **follower** if trajectories collide

Cooperative Game

Blocking Game



$$A = \begin{bmatrix} & & \\ & & \\ -10 & -10 & -10 \end{bmatrix}$$

$$B = \begin{bmatrix} & & -10 \\ & & -10 \\ & & -10 \end{bmatrix}$$

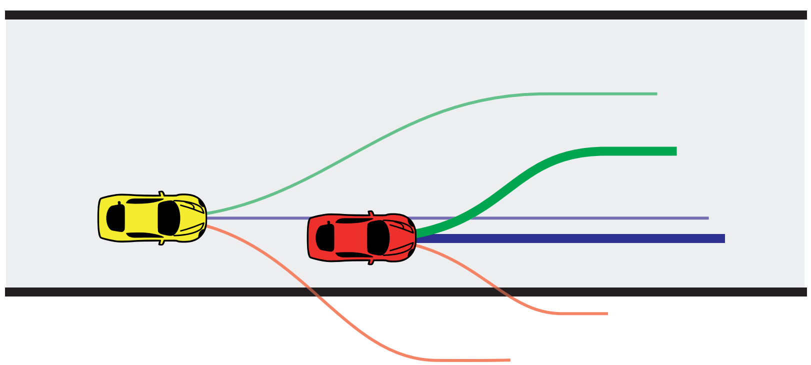
Three Racing Games

Sequential Game

- Exploiting the leader-follower structure
  - Low payoff if a trajectory leaves the track
  - Progress payoff if a trajectory is inside the track
  - Low payoff for the **follower** if trajectories collide

Cooperative Game

Blocking Game



$$A = \begin{bmatrix} 0.83 & 0.83 & 0.83 \\ 0.88 & 0.88 & 0.88 \\ -10 & -10 & -10 \end{bmatrix}$$

$$B = \begin{bmatrix} & & -10 \\ & & -10 \\ & & -10 \end{bmatrix}$$

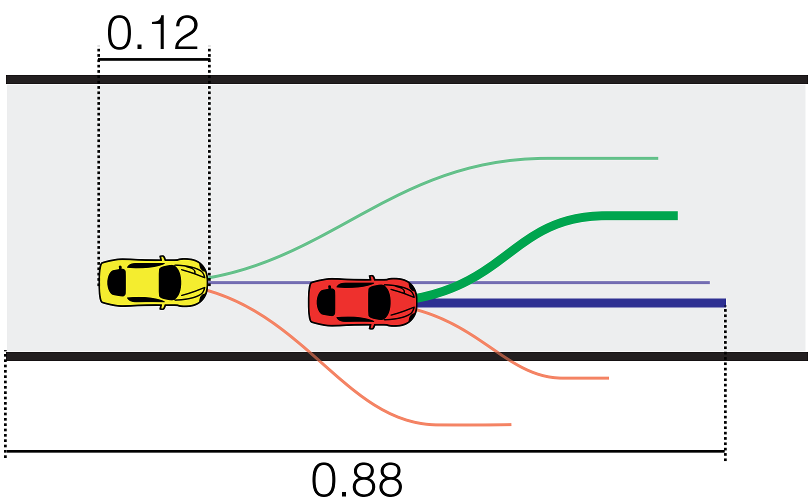
Three Racing Games

Sequential Game

- Exploiting the leader-follower structure
  - Low payoff if a trajectory leaves the track
  - Progress payoff if a trajectory is inside the track
  - Low payoff for the **follower** if trajectories collide

Cooperative Game

Blocking Game



$$A = \begin{bmatrix} 0.83 & 0.83 & 0.83 \\ 0.88 & 0.88 & 0.88 \\ -10 & -10 & -10 \end{bmatrix}$$

$$B = \begin{bmatrix} & & -10 \\ & & -10 \\ & & -10 \end{bmatrix}$$

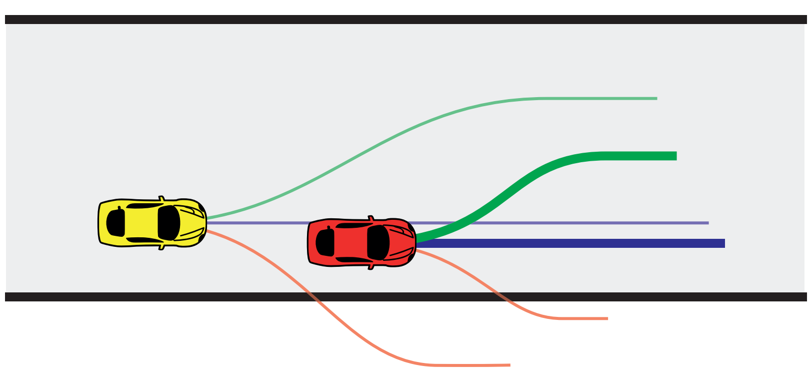
Three Racing Games

Sequential Game

- Exploiting the leader-follower structure
  - Low payoff if a trajectory leaves the track
  - Progress payoff if a trajectory is inside the track
  - Low payoff for the **follower** if trajectories collide

Cooperative Game

Blocking Game



$$A = \begin{bmatrix} 0.83 & 0.83 & 0.83 \\ 0.88 & 0.88 & 0.88 \\ -10 & -10 & -10 \end{bmatrix}$$

$$B = \begin{bmatrix} & & -10 \\ & & -10 \\ & & -10 \end{bmatrix}$$

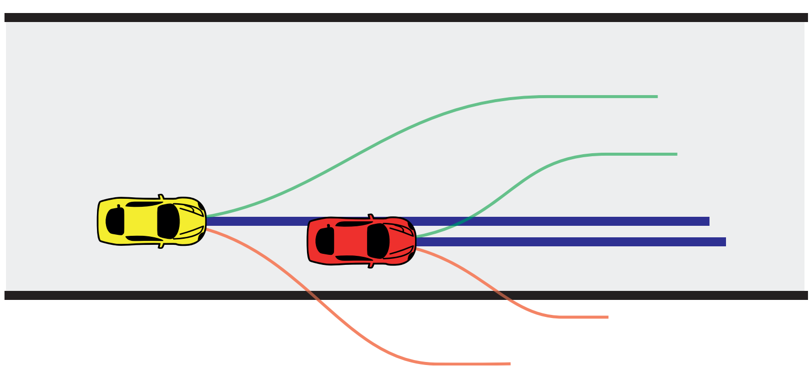
Three Racing Games

Sequential Game

- Exploiting the leader-follower structure
  - Low payoff if a trajectory leaves the track
  - Progress payoff if a trajectory is inside the track
  - Low payoff for the **follower** if trajectories collide

Cooperative Game

Blocking Game



$$A = \begin{bmatrix} 0.83 & 0.83 & 0.83 \\ 0.88 & 0.88 & 0.88 \\ -10 & -10 & -10 \end{bmatrix}$$

$$B = \begin{bmatrix} 0.81 & 0.86 & -10 \\ 0.81 & \boxed{-1} & -10 \\ 0.81 & 0.86 & -10 \end{bmatrix}$$

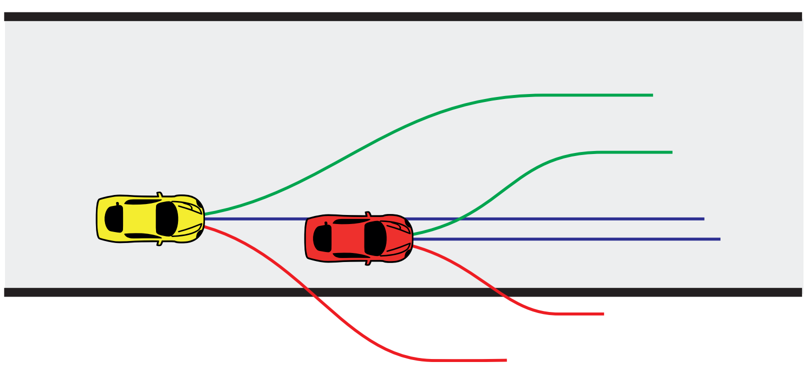
Three Racing Games

Sequential Game

Cooperative Game

- Both cars consider collisions
  - Low payoff if a trajectory leaves the track
  - Low payoff if the trajectories collide
  - Progress payoff if a trajectory is feasible

Blocking Game



$$A = \begin{bmatrix} 0.83 & 0.83 & 0.83 \\ 0.88 & -1 & 0.88 \\ -10 & -10 & -10 \end{bmatrix}$$

$$B = \begin{bmatrix} 0.81 & 0.86 & -10 \\ 0.81 & -1 & -10 \\ 0.81 & 0.86 & -10 \end{bmatrix}$$

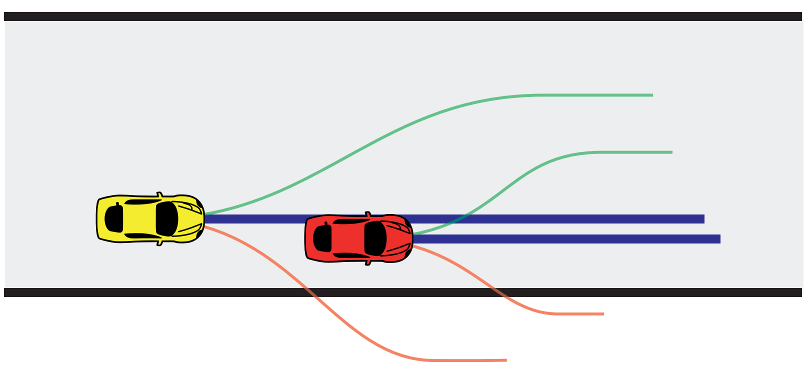
Three Racing Games

Sequential Game

Cooperative Game

- Both cars consider collisions
  - Low payoff if a trajectory leaves the track
  - Low payoff if the trajectories collide
  - Progress payoff if a trajectory is feasible

Blocking Game



$$A = \begin{bmatrix} 0.83 & 0.83 & 0.83 \\ 0.88 & \boxed{-1} & 0.88 \\ -10 & -10 & -10 \end{bmatrix}$$

$$B = \begin{bmatrix} 0.81 & 0.86 & -10 \\ 0.81 & -1 & -10 \\ 0.81 & 0.86 & -10 \end{bmatrix}$$

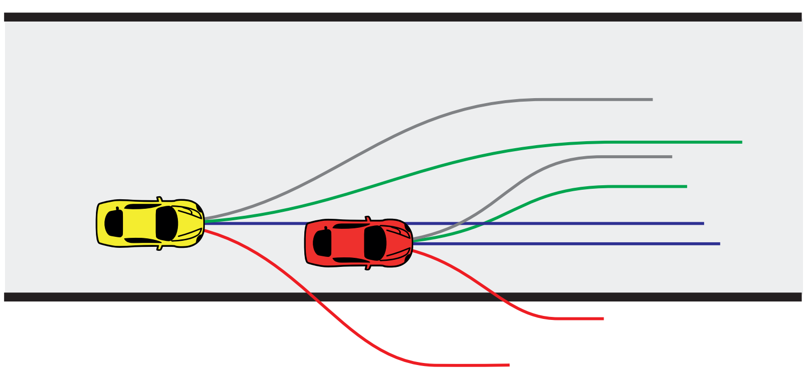
Three Racing Games

Sequential Game

Cooperative Game

Blocking Game

- Same collision structure as the cooperative game, **but:**
- Additional reward for staying **in front** at the end of the horizon



$A =$

$B =$

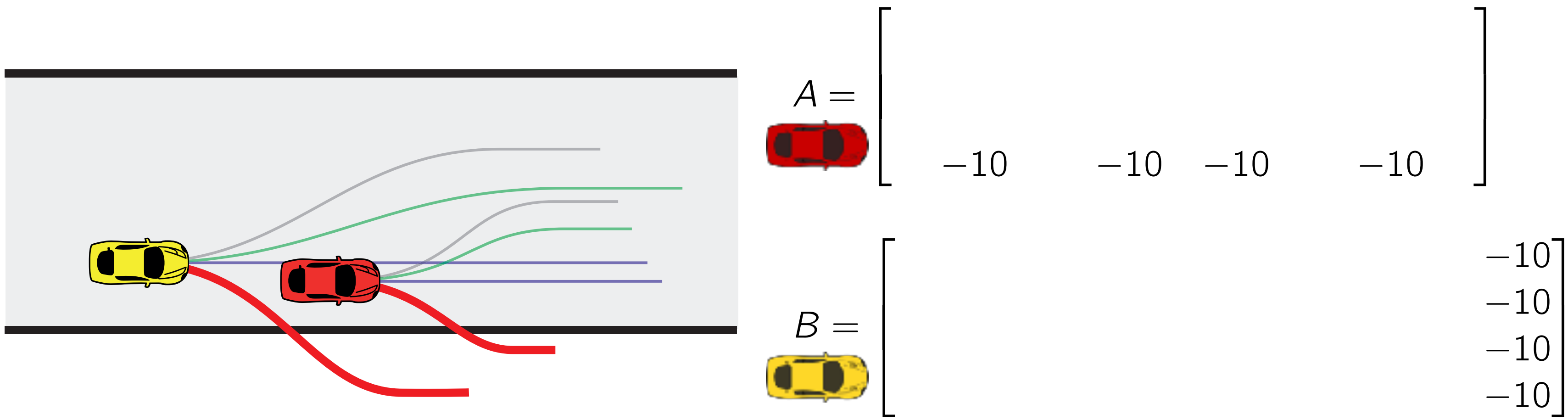
Three Racing Games

Sequential Game

Cooperative Game

Blocking Game

- Same collision structure as the cooperative game, **but:**
- Additional reward for staying **in front** at the end of the horizon



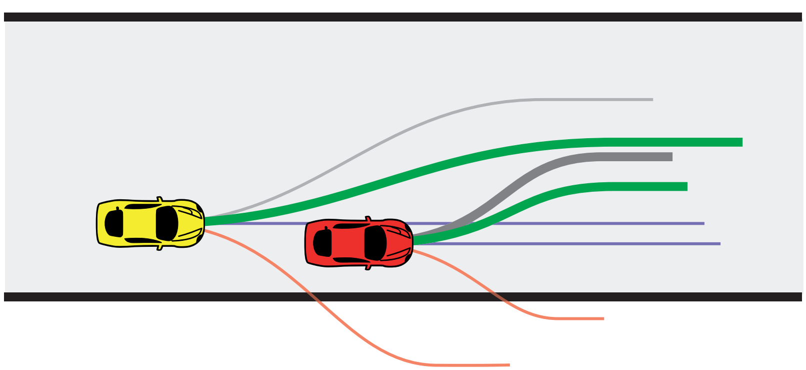
Three Racing Games

Sequential Game

Cooperative Game

Blocking Game

- Same collision structure as the cooperative game, **but:**
- Additional reward for staying **in front** at the end of the horizon



$$A = \begin{bmatrix} & -1 & & \\ & -1 & & \\ & & & \\ -10 & -10 & -10 & -10 \end{bmatrix}$$

$$B = \begin{bmatrix} & -1 & & -10 \\ & -1 & & -10 \\ & & & -10 \\ & & & -10 \end{bmatrix}$$

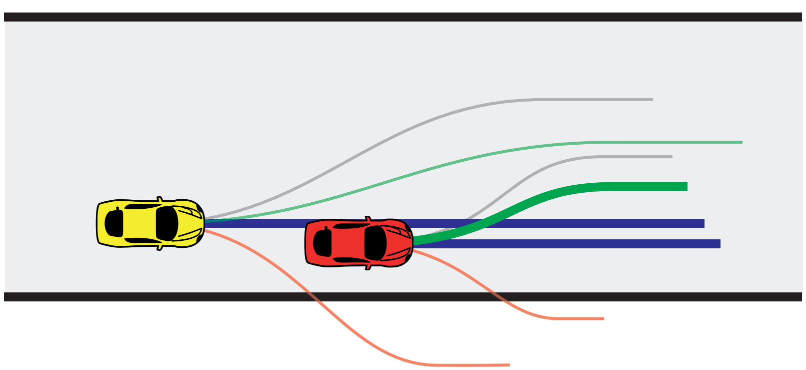
Three Racing Games

Sequential Game

Cooperative Game

Blocking Game

- Same collision structure as the cooperative game, **but:**
- Additional reward for staying **in front** at the end of the horizon



$$A = \begin{bmatrix} & -1 & & \\ & -1 & -1 & \\ & & -1 & \\ -10 & -10 & -10 & -10 \end{bmatrix}$$

$$B = \begin{bmatrix} -1 & & -10 \\ -1 & -1 & -10 \\ & -1 & -10 \\ & & -10 \end{bmatrix}$$

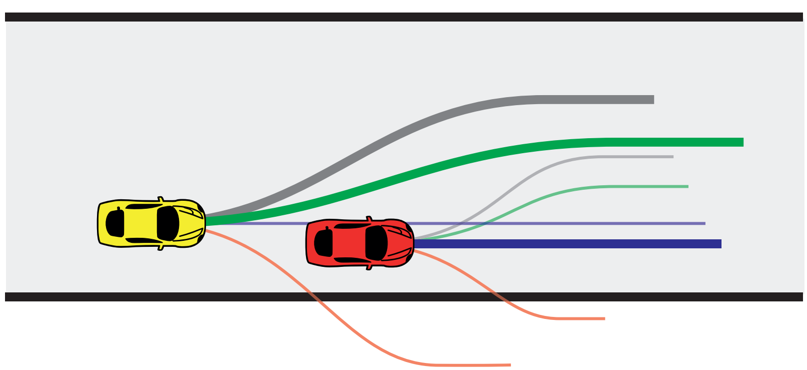
Three Racing Games

Sequential Game

Cooperative Game

Blocking Game

- Same collision structure as the cooperative game, **but:**
- Additional reward for staying **in front** at the end of the horizon



$$A = \begin{bmatrix} & & -1 & & & & \\ & & -1 & -1 & & & \\ 0.88 & & 0.88 & -1 & 0.88 & & \\ & -10 & -10 & -10 & & -10 & \end{bmatrix}$$

$$B = \begin{bmatrix} & & -1 & & -10 \\ & & -1 & -1 & -10 \\ 0.81 & 0.9 & & -1 & -10 \\ & & & & -10 \end{bmatrix}$$

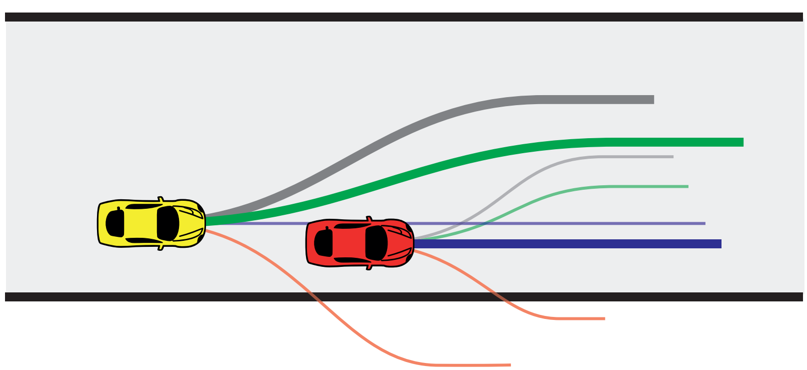
Three Racing Games

Sequential Game

Cooperative Game

Blocking Game

- Same collision structure as the cooperative game, **but:**
- Additional reward for staying **in front** at the end of the horizon



$$A = \begin{bmatrix} & -1 & & \\ & -1 & -1 & \\ 0.88 \boxed{+\,0.5} & 0.88 & -1 & 0.88+0.5 \\ -10 & -10 & -10 & -10 \end{bmatrix}$$

$$B = \begin{bmatrix} & -1 & & -10 \\ & -1 & -1 & -10 \\ 0.81 & 0.9 & -1 & -10 \\ & & & -10 \end{bmatrix}$$

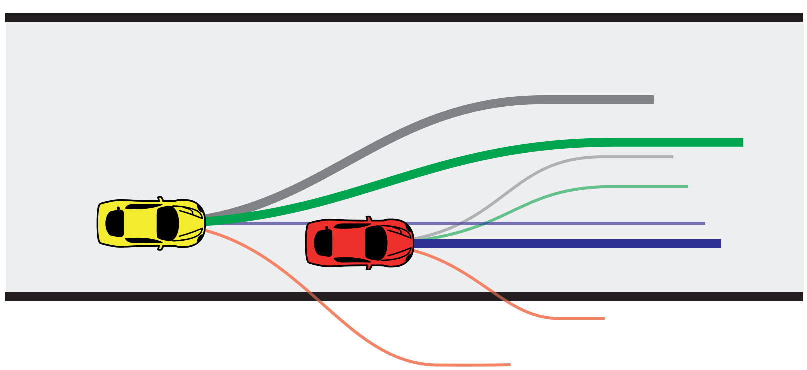
Three Racing Games

Sequential Game

Cooperative Game

Blocking Game

- Same collision structure as the cooperative game, **but:**
- Additional reward for staying **in front** at the end of the horizon



$$A = \begin{bmatrix} & -1 & & \\ & -1 & -1 & \\ 0.88 + 0.5 & 0.88 & -1 & 0.88 + 0.5 \\ -10 & -10 & -10 & -10 \end{bmatrix}$$

$$B = \begin{bmatrix} & -1 & & -10 \\ & -1 & -1 & -10 \\ 0.81 & 0.9 + 0.5 & -1 & -10 \\ & & & -10 \end{bmatrix}$$

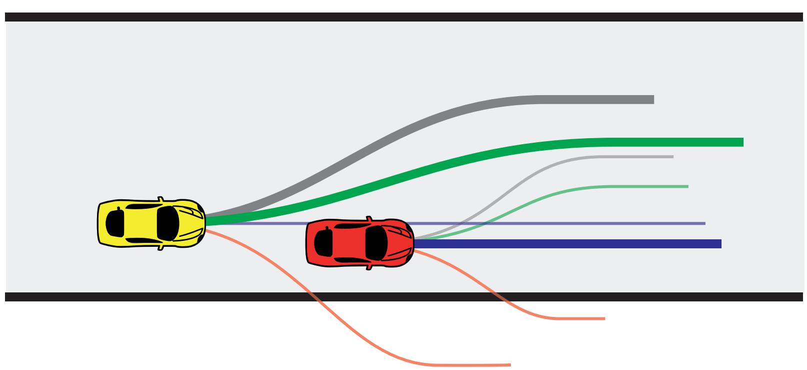
Three Racing Games

Sequential Game

Cooperative Game

Blocking Game

- Same collision structure as the cooperative game, **but:**
- Additional reward for staying **in front** at the end of the horizon



$$A = \begin{bmatrix} 0.83+0.5 & -1 & 0.83 & 0.83+0.5 \\ 0.85+0.5 & -1 & -1 & 0.85+0.5 \\ 0.88+0.5 & 0.88 & -1 & 0.88+0.5 \\ -10 & -10 & -10 & -10 \end{bmatrix}$$

$$B = \begin{bmatrix} 0.81 & -1 & 0.86+0.5 & -10 \\ 0.81 & -1 & -1 & -10 \\ 0.81 & 0.9+0.5 & -1 & -10 \\ 0.81+0.5 & 0.9+0.5 & 0.86+0.5 & -10 \end{bmatrix}$$

Three Racing Games

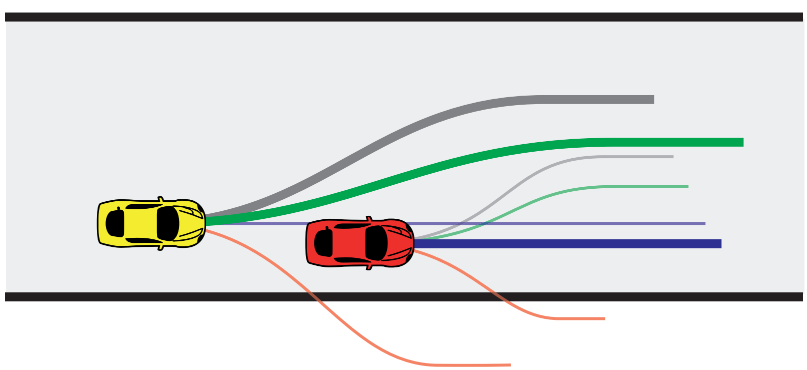
Sequential Game

Cooperative Game

Blocking Game

- Same collision structure as the cooperative game, **but:**
- Additional reward for staying **in front** at the end of the horizon

How should a car choose a trajectory?



$$A = \begin{bmatrix} 0.83+0.5 & -1 & 0.83 & 0.83+0.5 \\ 0.85+0.5 & -1 & -1 & 0.85+0.5 \\ 0.88+0.5 & 0.88 & -1 & 0.88+0.5 \\ -10 & -10 & -10 & -10 \end{bmatrix}$$

$$B = \begin{bmatrix} 0.81 & -1 & 0.86+0.5 & -10 \\ 0.81 & -1 & -1 & -10 \\ 0.81 & 0.9+0.5 & -1 & -10 \\ 0.81+0.5 & 0.9+0.5 & 0.86+0.5 & -10 \end{bmatrix}$$

Equilibria concepts

- Find an equilibrium trajectory pair of the bimatrix game
  - Pure strategies (no mixed strategies)
  - $(i^*, j^*) \in \Gamma^1 \times \Gamma^2$ is an equilibrium trajectory pair

Stackelberg Equilibria

- Game with leader-follower structure
  - Leader can enforce his trajectory on the follower
  - Follower plays the **best response:**

    $R(i) = \arg\max_{j \in \Gamma^2} b_{i,j}$

$$i^* = \arg\max_{i \in \Gamma^1} \min_{j \in R(i)} a_{i,j}$$

$$j^* = R(i^*)$$

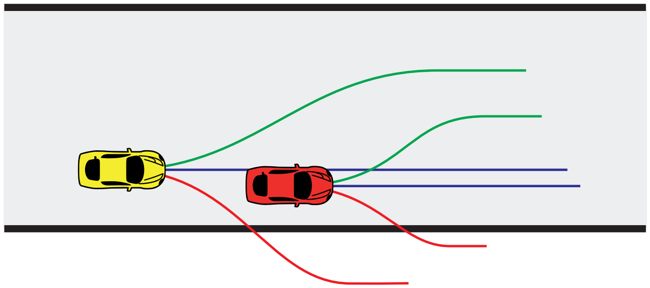
Nash Equilibria

- None of the players has a benefit from **unilaterally** changing the trajectory

$$a_{i^*,j^*} \geq a_{i,j^*} \quad \forall i \in \Gamma^1$$

$$b_{i^*,j^*} \geq b_{i^*,j} \quad \forall j \in \Gamma^2$$

Sequential and Cooperative Game



sequential game

$$A = \begin{bmatrix} 0.83 & 0.83 & 0.83 \\ 0.88 & 0.88 & 0.88 \\ -10 & -10 & -10 \end{bmatrix}$$

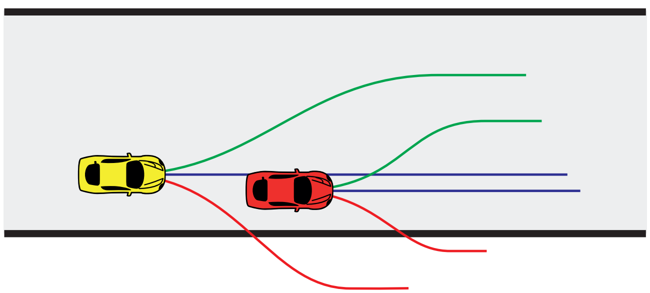
$$B = \begin{bmatrix} 0.81 & 0.86 & -10 \\ 0.81 & -1 & -10 \\ 0.81 & 0.86 & -10 \end{bmatrix}$$

cooperative game

$$A = \begin{bmatrix} 0.83 & 0.83 & 0.83 \\ 0.88 & -1 & 0.88 \\ -10 & -10 & -10 \end{bmatrix}$$

$$B = \begin{bmatrix} 0.81 & 0.86 & -10 \\ 0.81 & -1 & -10 \\ 0.81 & 0.86 & -10 \end{bmatrix}$$

Sequential and Cooperative Game



sequential game

$$A = \begin{bmatrix} 0.83 & 0.83 & 0.83 \\ 0.88 & 0.88 & 0.88 \\ -10 & -10 & -10 \end{bmatrix}$$

$$B = \begin{bmatrix} 0.81 & 0.86 & -10 \\ 0.81 & -1 & -10 \\ 0.81 & 0.86 & -10 \end{bmatrix}$$

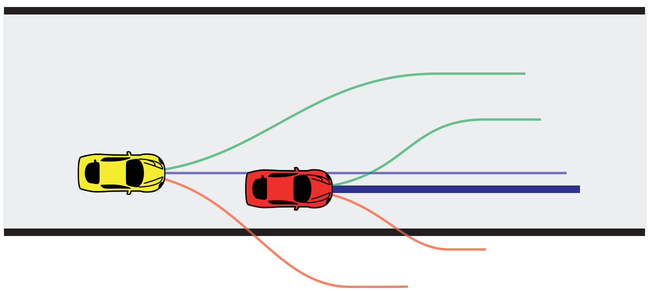
cooperative game

$$A = \begin{bmatrix} 0.83 & 0.83 & 0.83 \\ 0.88 & -1 & 0.88 \\ -10 & -10 & -10 \end{bmatrix}$$

$$B = \begin{bmatrix} 0.81 & 0.86 & -10 \\ 0.81 & -1 & -10 \\ 0.81 & 0.86 & -10 \end{bmatrix}$$

- The sequential game can be solved by sequential maximizing

Sequential and Cooperative Game



sequential game

$$A = \begin{bmatrix} 0.83 & 0.83 & 0.83 \\ 0.88 & 0.88 & 0.88 \\ -10 & -10 & -10 \end{bmatrix}$$

$$B = \begin{bmatrix} 0.81 & 0.86 & -10 \\ 0.81 & -1 & -10 \\ 0.81 & 0.86 & -10 \end{bmatrix}$$

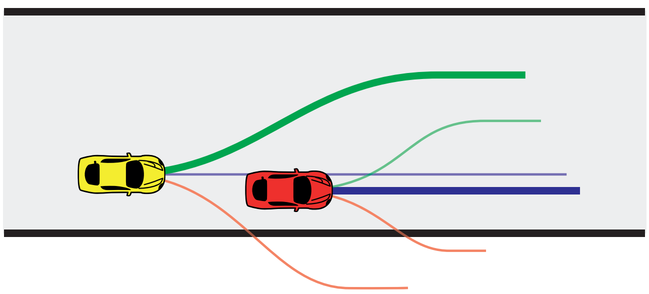
cooperative game

$$A = \begin{bmatrix} 0.83 & 0.83 & 0.83 \\ 0.88 & -1 & 0.88 \\ -10 & -10 & -10 \end{bmatrix}$$

$$B = \begin{bmatrix} 0.81 & 0.86 & -10 \\ 0.81 & -1 & -10 \\ 0.81 & 0.86 & -10 \end{bmatrix}$$

- The sequential game can be solved by sequential maximizing

Sequential and Cooperative Game



sequential game

$$A = \begin{bmatrix} 0.83 & 0.83 & 0.83 \\ 0.88 & 0.88 & 0.88 \\ -10 & -10 & -10 \end{bmatrix}$$

$$B = \begin{bmatrix} 0.81 & 0.86 & -10 \\ 0.81 & -1 & -10 \\ 0.81 & 0.86 & -10 \end{bmatrix}$$

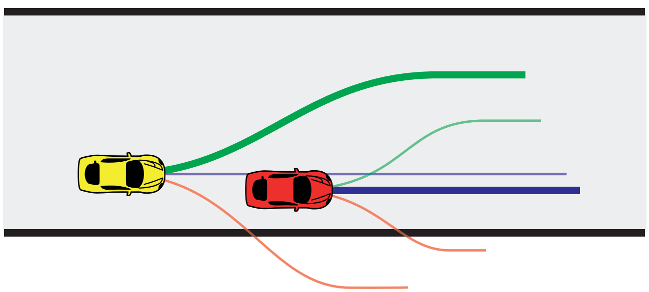
cooperative game

$$A = \begin{bmatrix} 0.83 & 0.83 & 0.83 \\ 0.88 & -1 & 0.88 \\ -10 & -10 & -10 \end{bmatrix}$$

$$B = \begin{bmatrix} 0.81 & 0.86 & -10 \\ 0.81 & -1 & -10 \\ 0.81 & 0.86 & -10 \end{bmatrix}$$

- The sequential game can be solved by sequential maximizing

Sequential and Cooperative Game



sequential game

$$A = \begin{bmatrix} 0.83 & 0.83 & 0.83 \\ 0.88 & 0.88 & 0.88 \\ -10 & -10 & -10 \end{bmatrix}$$

$$B = \begin{bmatrix} 0.81 & 0.86 & -10 \\ 0.81 & -1 & -10 \\ 0.81 & 0.86 & -10 \end{bmatrix}$$

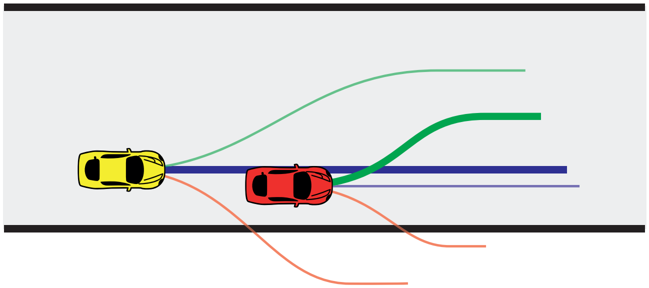
cooperative game

$$A = \begin{bmatrix} 0.83 & 0.83 & 0.83 \\ 0.88 & -1 & 0.88 \\ -10 & -10 & -10 \end{bmatrix}$$

$$B = \begin{bmatrix} 0.81 & 0.86 & -10 \\ 0.81 & -1 & -10 \\ 0.81 & 0.86 & -10 \end{bmatrix}$$

- The sequential game can be solved by sequential maximizing

Sequential and Cooperative Game



sequential game

$$A = \begin{bmatrix} 0.83 & 0.83 & 0.83 \\ \boxed{0.88} & 0.88 & 0.88 \\ -10 & -10 & -10 \end{bmatrix}$$

$$B = \begin{bmatrix} 0.81 & 0.86 & -10 \\ \boxed{0.81} & -1 & -10 \\ 0.81 & 0.86 & -10 \end{bmatrix}$$

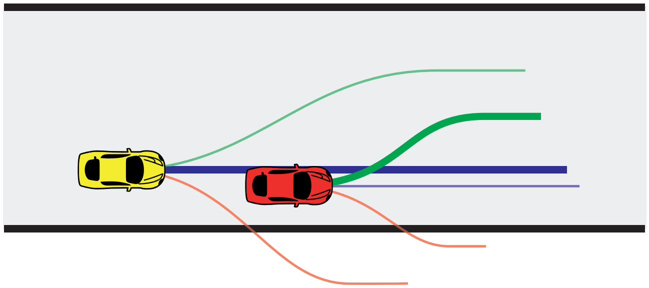
cooperative game

$$A = \begin{bmatrix} 0.83 & \boxed{0.83} & 0.83 \\ \boxed{0.88} & -1 & 0.88 \\ -10 & -10 & -10 \end{bmatrix}$$

$$B = \begin{bmatrix} 0.81 & \boxed{0.86} & -10 \\ \boxed{0.81} & -1 & -10 \\ 0.81 & 0.86 & -10 \end{bmatrix}$$

- The sequential game can be solved by sequential maximizing

Sequential and Cooperative Game



sequential game

$$A = \begin{bmatrix} 0.83 & 0.83 & 0.83 \\ \boxed{0.88} & 0.88 & 0.88 \\ -10 & -10 & -10 \end{bmatrix}$$

$$B = \begin{bmatrix} 0.81 & 0.86 & -10 \\ \boxed{0.81} & -1 & -10 \\ 0.81 & 0.86 & -10 \end{bmatrix}$$

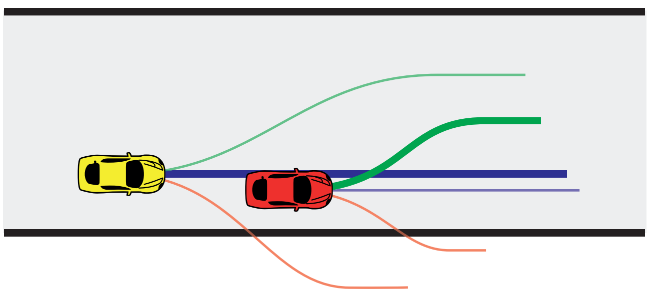
cooperative game

$$A = \begin{bmatrix} 0.83 & \boxed{0.83} & 0.83 \\ \boxed{0.88} & -1 & 0.88 \\ -10 & -10 & -10 \end{bmatrix}$$

$$B = \begin{bmatrix} 0.81 & \boxed{0.86} & -10 \\ \boxed{0.81} & -1 & -10 \\ 0.81 & 0.86 & -10 \end{bmatrix}$$

- The sequential game can be solved by sequential maximizing
- Sequential game feasible => equilibrium of the cooperative game
  - Predicting ideal behavior of other cars and play best response is Nash

Sequential and Cooperative Game



sequential game

$$A = \begin{bmatrix} 0.83 & 0.83 & 0.83 \\ \boxed{0.88} & 0.88 & 0.88 \\ -10 & -10 & -10 \end{bmatrix}$$

$$B = \begin{bmatrix} 0.81 & 0.86 & -10 \\ \boxed{0.81} & -1 & -10 \\ 0.81 & 0.86 & -10 \end{bmatrix}$$

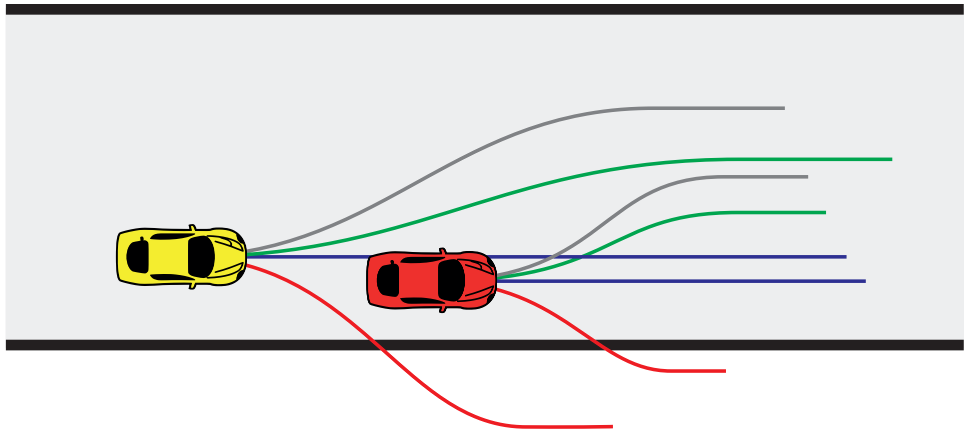
cooperative game

$$A = \begin{bmatrix} 0.83 & \boxed{0.83} & 0.83 \\ \boxed{0.88} & -1 & 0.88 \\ -10 & -10 & -10 \end{bmatrix}$$

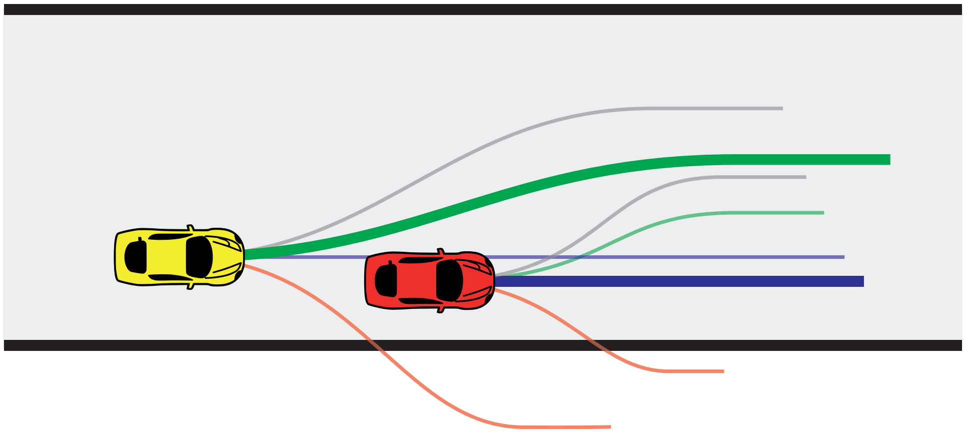
$$B = \begin{bmatrix} 0.81 & \boxed{0.86} & -10 \\ \boxed{0.81} & -1 & -10 \\ 0.81 & 0.86 & -10 \end{bmatrix}$$

- The sequential game can be solved by sequential maximizing
- Sequential game feasible => equilibrium of the cooperative game
  - Predicting ideal behavior of other cars and play best response is Nash
- Cooperative game is feasible if there exists a feasible trajectory pair

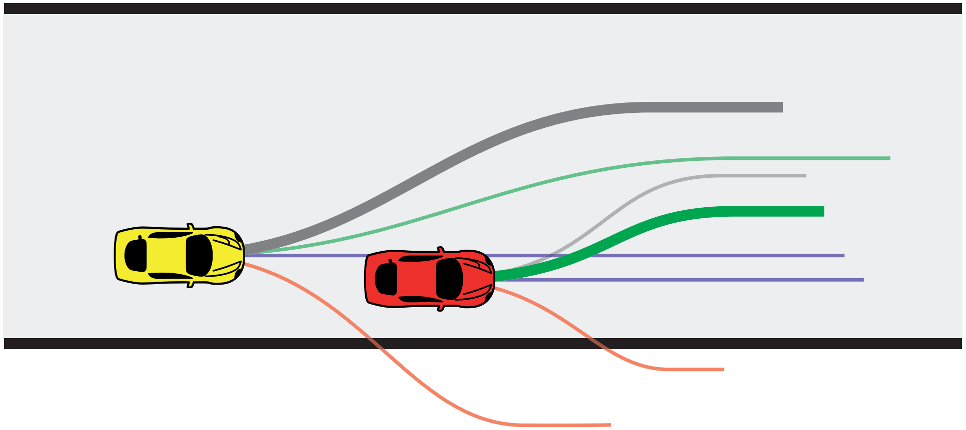
Blocking Trajectories



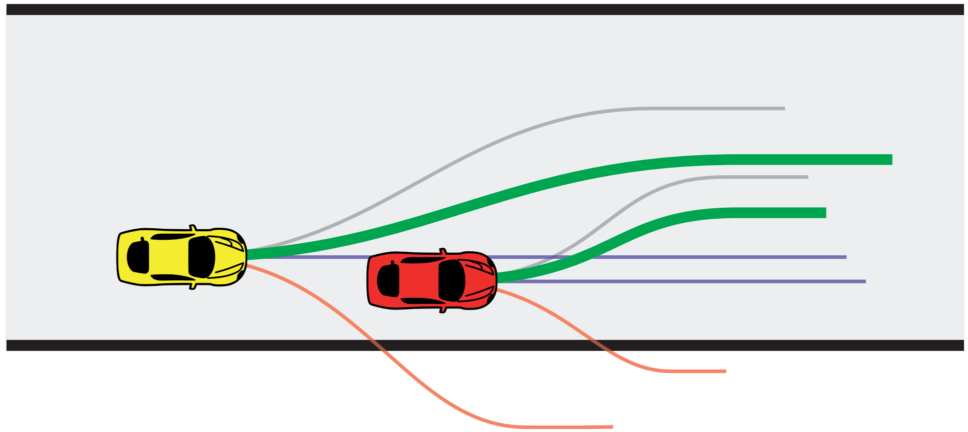
Blocking Trajectories



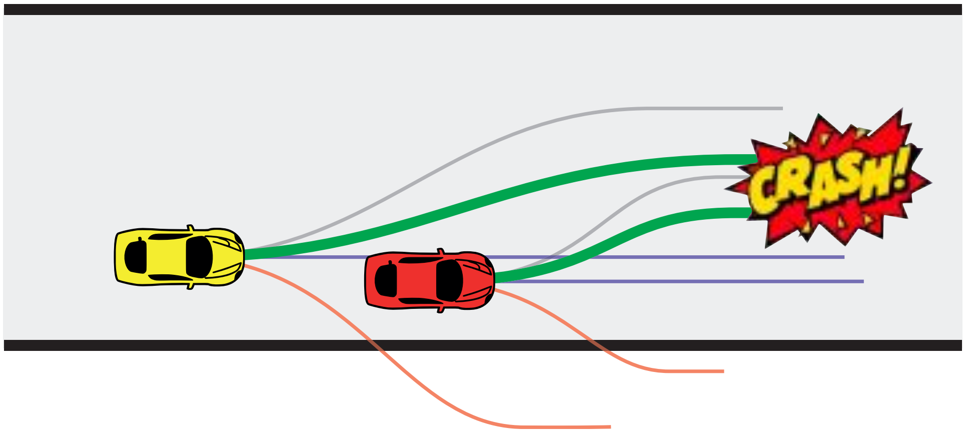
Blocking Trajectories



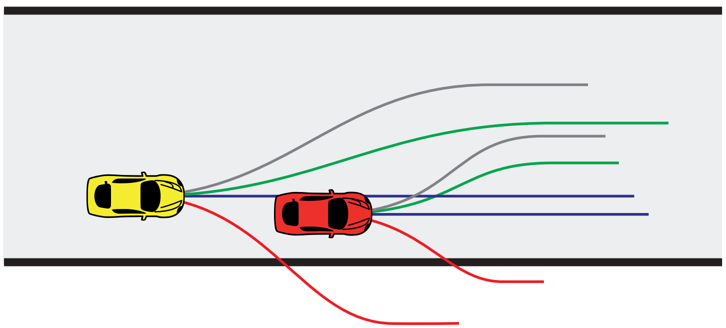
Blocking Trajectories



Blocking Trajectories



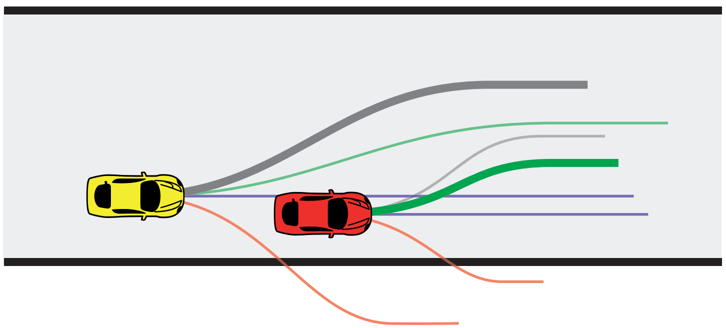
Blocking Trajectories



$$A = \begin{bmatrix} 0.83+0.5 & -1 & 0.83 & 0.83+0.5 \\ 0.85+0.5 & -1 & -1 & 0.85+0.5 \\ 0.88+0.5 & 0.88 & -1 & 0.88+0.5 \\ -10 & -10 & -10 & -10 \end{bmatrix}$$

$$B = \begin{bmatrix} 0.81 & -1 & 0.86+0.5 & -10 \\ 0.81 & -1 & -1 & -10 \\ 0.81 & 0.9+0.5 & -1 & -10 \\ 0.81+0.5 & 0.9+0.5 & 0.86+0.5 & -10 \end{bmatrix}$$

Blocking Trajectories

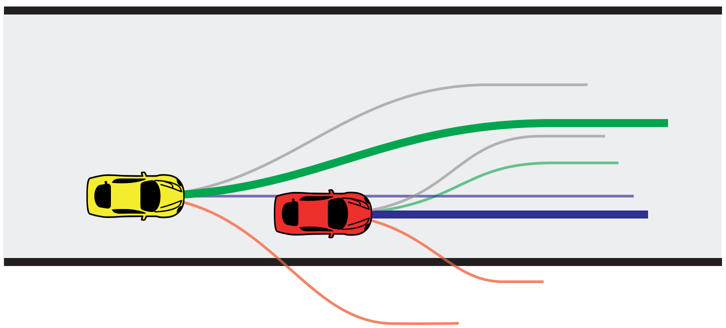


$$A = \begin{bmatrix} 0.83+0.5 & -1 & 0.83 & 0.83+0.5 \\ \boxed{0.85+0.5} & -1 & -1 & 0.85+0.5 \\ 0.88+0.5 & 0.88 & -1 & 0.88+0.5 \\ -10 & -10 & -10 & -10 \end{bmatrix}$$

$$B = \begin{bmatrix} 0.81 & -1 & 0.86+0.5 & -10 \\ \boxed{0.81} & -1 & -1 & -10 \\ 0.81 & 0.9+0.5 & -1 & -10 \\ 0.81+0.5 & 0.9+0.5 & 0.86+0.5 & -10 \end{bmatrix}$$

- If there exists a blocking trajectory and the **staying ahead reward** is big enough, the Stackelberg equilibrium is a blocking trajectory pair

Blocking Trajectories

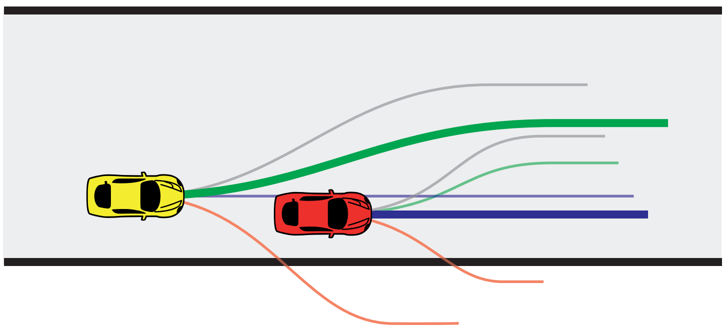


$$A = \begin{bmatrix} 0.83+0.5 & -1 & 0.83 & 0.83+0.5 \\ \boxed{0.85+0.5} & -1 & -1 & 0.85+0.5 \\ 0.88+0.5 & \boxed{0.88} & -1 & 0.88+0.5 \\ -10 & -10 & -10 & -10 \end{bmatrix}$$

$$B = \begin{bmatrix} 0.81 & -1 & 0.86+0.5 & -10 \\ \boxed{0.81} & -1 & -1 & -10 \\ 0.81 & \boxed{0.9+0.5} & -1 & -10 \\ 0.81+0.5 & 0.9+0.5 & 0.86+0.5 & -10 \end{bmatrix}$$

- If there exists a blocking trajectory and the **staying ahead reward** is big enough, the Stackelberg equilibrium is a blocking trajectory pair
- A blocking trajectory is not a Nash equilibrium (unless it is a Nash equilibrium of the cooperative game)

Blocking Trajectories



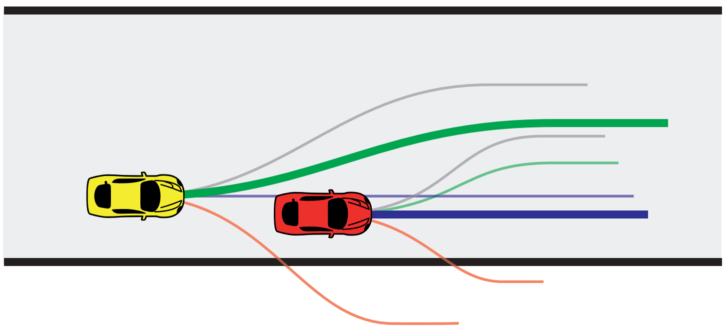
$$A = \begin{bmatrix} 0.83+0.5 & -1 & 0.83 & 0.83+0.5 \\ \boxed{0.85+0.5} & -1 & -1 & 0.85+0.5 \\ 0.88+0.5 & \boxed{0.88} & -1 & 0.88+0.5 \\ -10 & -10 & -10 & -10 \end{bmatrix}$$

$$B = \begin{bmatrix} 0.81 & -1 & 0.86+0.5 & -10 \\ \boxed{0.81} & -1 & -1 & -10 \\ 0.81 & \boxed{0.9+0.5} & -1 & -10 \\ 0.81+0.5 & 0.9+0.5 & 0.86+0.5 & -10 \end{bmatrix}$$

- If there exists a blocking trajectory and the **staying ahead reward** is big enough, the Stackelberg equilibrium is a blocking trajectory pair
- A blocking trajectory is not a Nash equilibrium (unless it is a Nash equilibrium of the cooperative game)

Stackelberg equilibrium seems best for all games

Blocking Trajectories



$$A = \begin{bmatrix} 0.83+0.5 & -1 & 0.83 & 0.83+0.5 \\ \boxed{0.85+0.5} & -1 & -1 & 0.85+0.5 \\ 0.88+0.5 & \boxed{0.88} & -1 & 0.88+0.5 \\ -10 & -10 & -10 & -10 \end{bmatrix}$$

$$B = \begin{bmatrix} 0.81 & -1 & 0.86+0.5 & -10 \\ \boxed{0.81} & -1 & -1 & -10 \\ 0.81 & \boxed{0.9+0.5} & -1 & -10 \\ 0.81+0.5 & 0.9+0.5 & 0.86+0.5 & -10 \end{bmatrix}$$

- If there exists a blocking trajectory and the **staying ahead reward** is big enough, the Stackelberg equilibrium is a blocking trajectory pair
- A blocking trajectory is not a Nash equilibrium (unless it is a Nash equilibrium of the cooperative game)

Stackelberg equilibrium seems best for all games

What is the resulting behavior of these games?

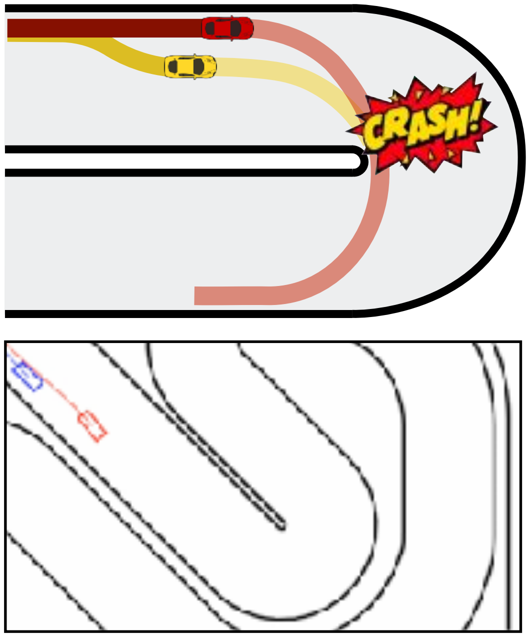
Simulation

- Play game in a receding horizon fashion
  - Solve game + MPC - apply first input - repeat
- Trajectory pruning based on viability and discriminating kernel
  - Viab -> aggressive driver / Disc -> cautious driver
- 500 different initial conditions, each run 4.5 laps
  - Both cars start close to each other

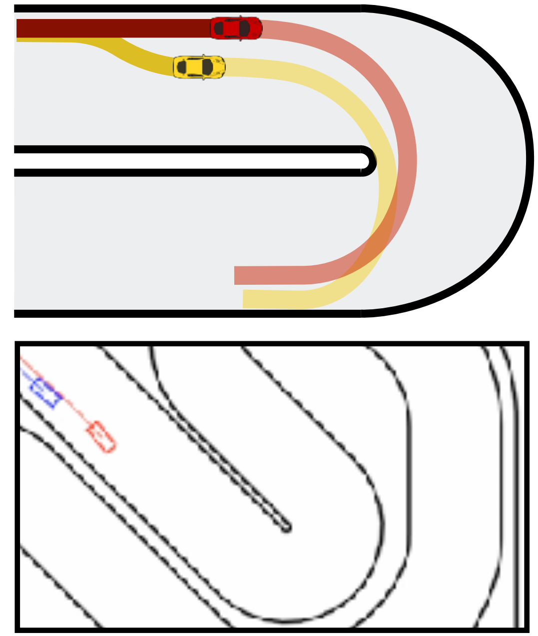
	sequential game	cooperative game	blocking game
# of overtaking maneuvers	113	857	414
colliding time steps per lap	2.4	2.0	2.3

Simulation

sequential game

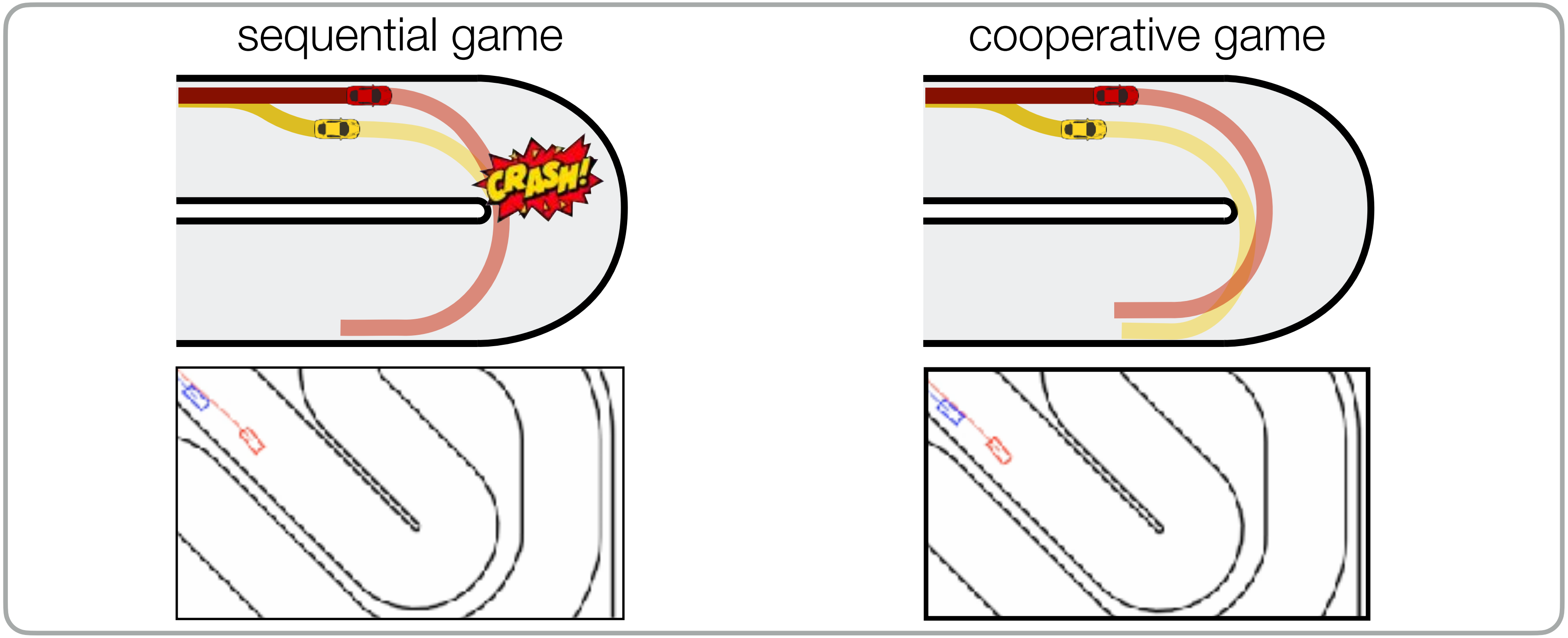


cooperative game



	sequential game	cooperative game	blocking game
# of overtaking maneuvers	113	857	414
colliding time steps per lap	2.4	2.0	2.3

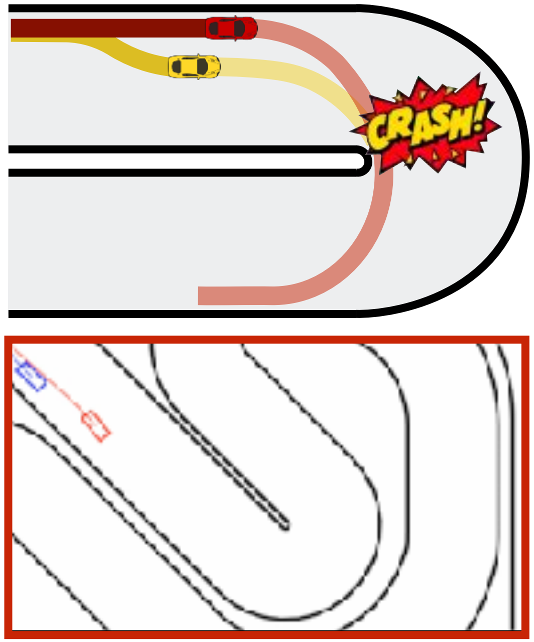
Simulation



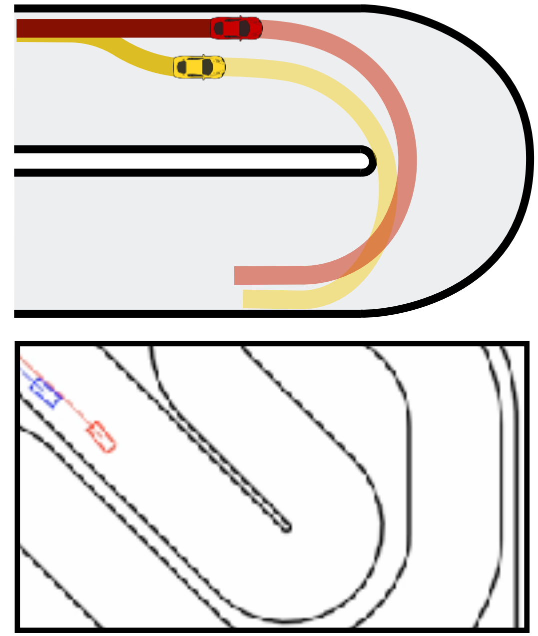
	sequential game	cooperative game	blocking game
# of overtaking maneuvers	113	857	414
colliding time steps per lap	2.4	2.0	2.3

Simulation

sequential game



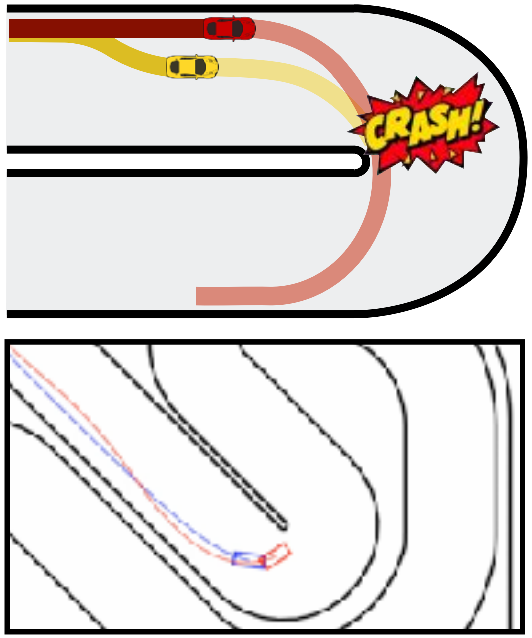
cooperative game



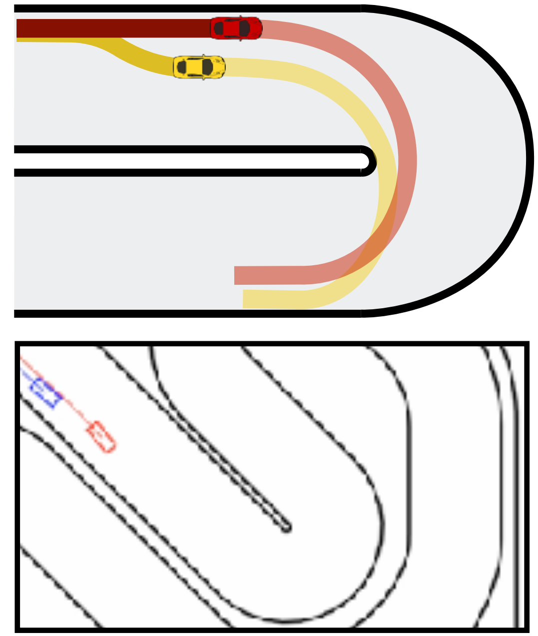
	sequential game	cooperative game	blocking game
# of overtaking maneuvers	113	857	414
colliding time steps per lap	2.4	2.0	2.3

Simulation

sequential game



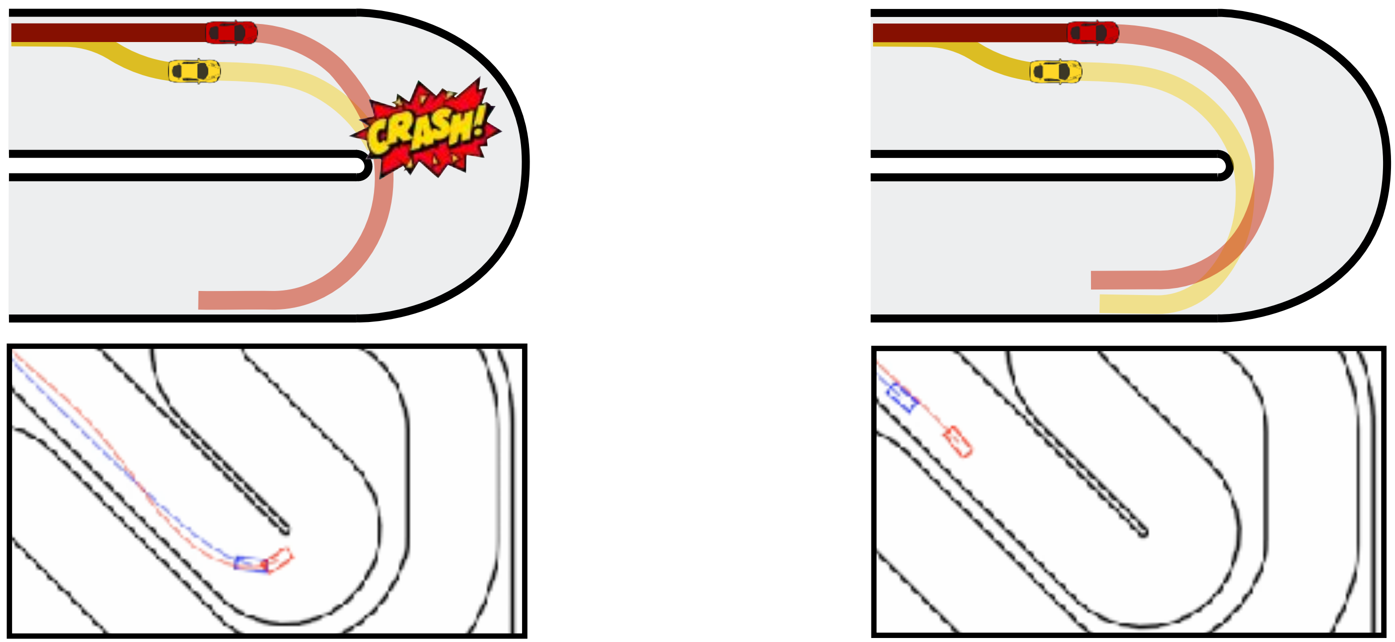
cooperative game



	sequential game	cooperative game	blocking game
# of overtaking maneuvers	113	857	414
colliding time steps per lap	2.4	2.0	2.3

Simulation

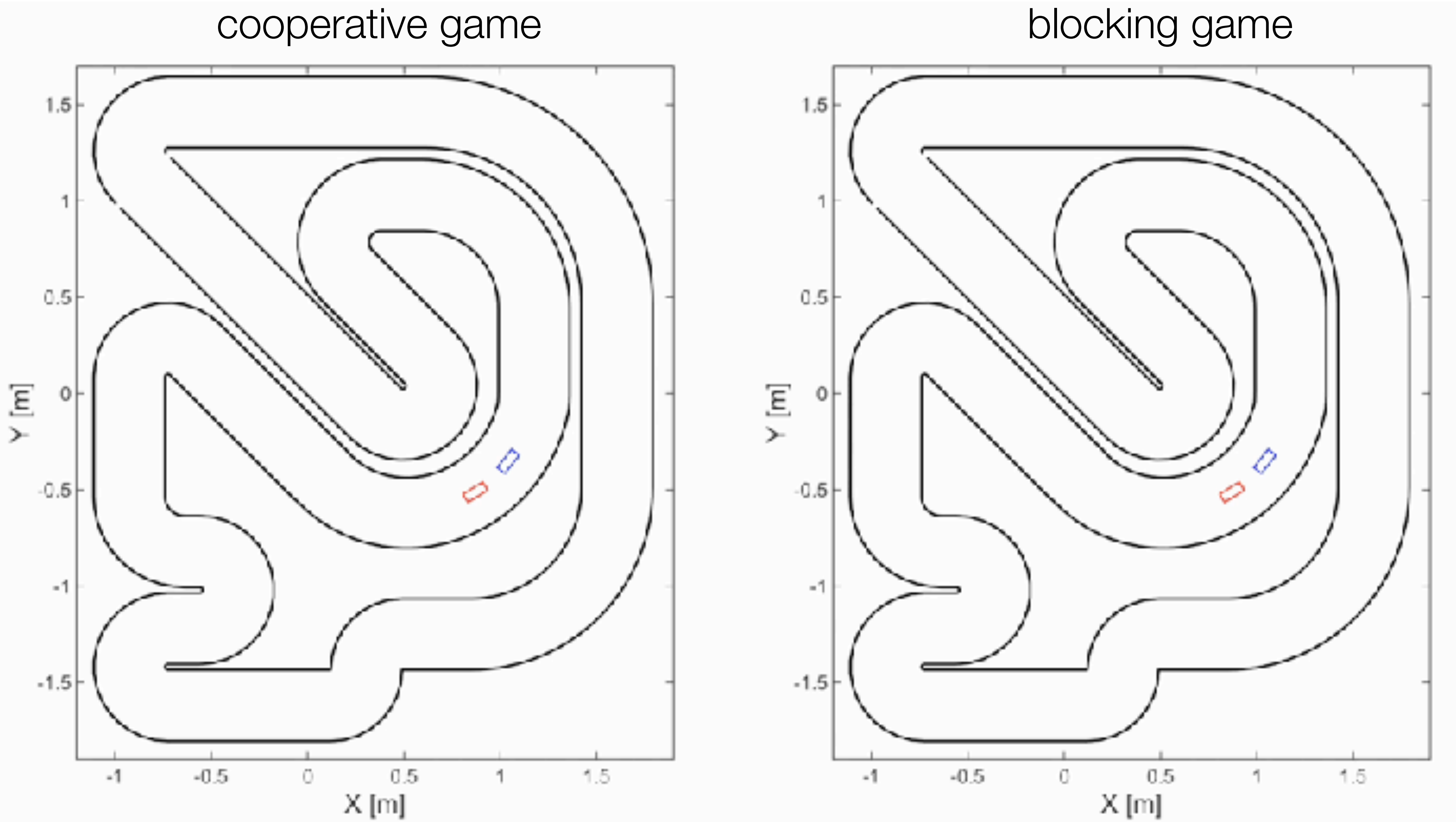
sequential game   cooperative game



	sequential game	cooperative game	blocking game
# of overtaking maneuvers	113	857	414
colliding time steps per lap	2.4	2.0	2.3

How do the cars drive?

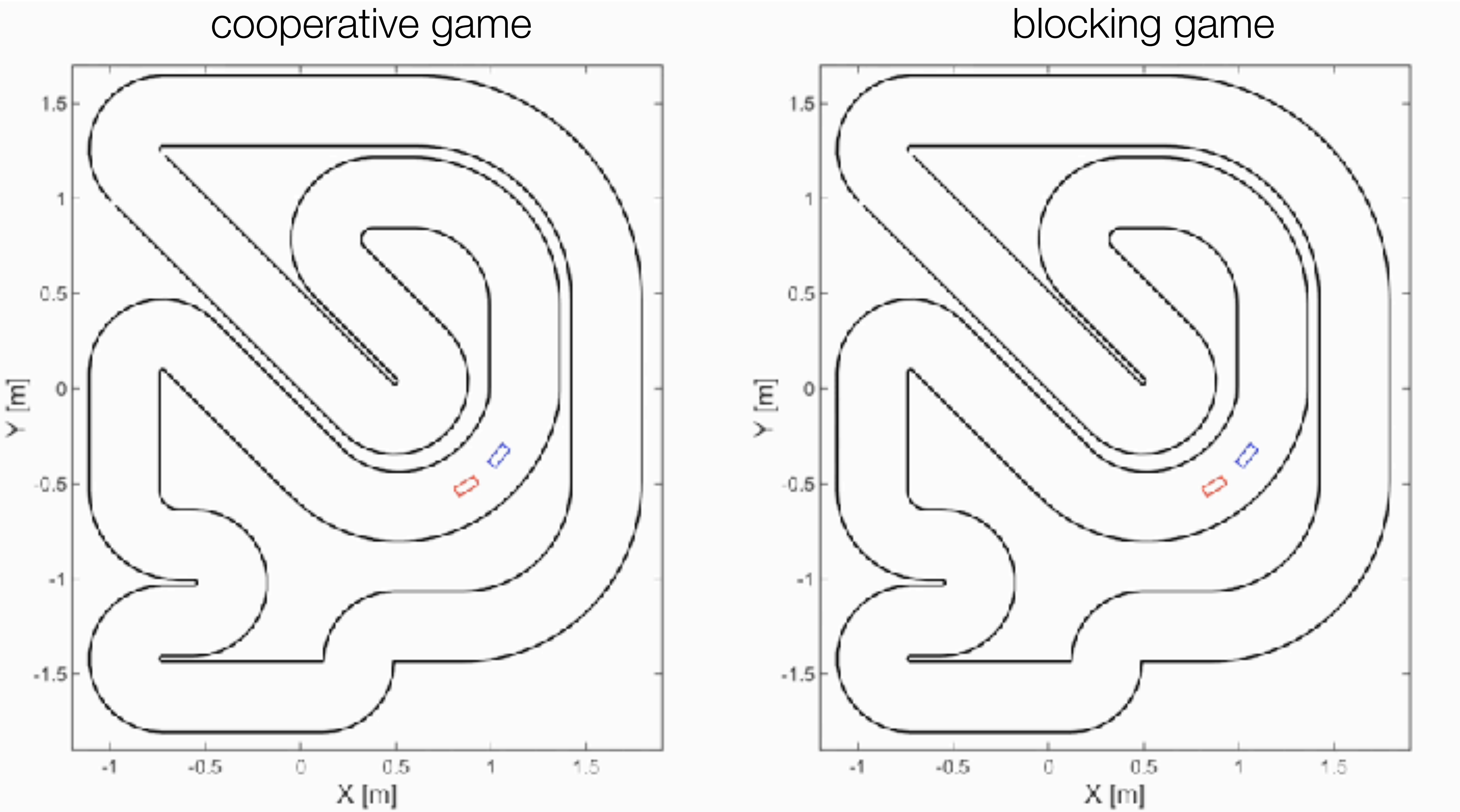
Simulation



Viab -> aggressive driver

Disc -> cautious driver

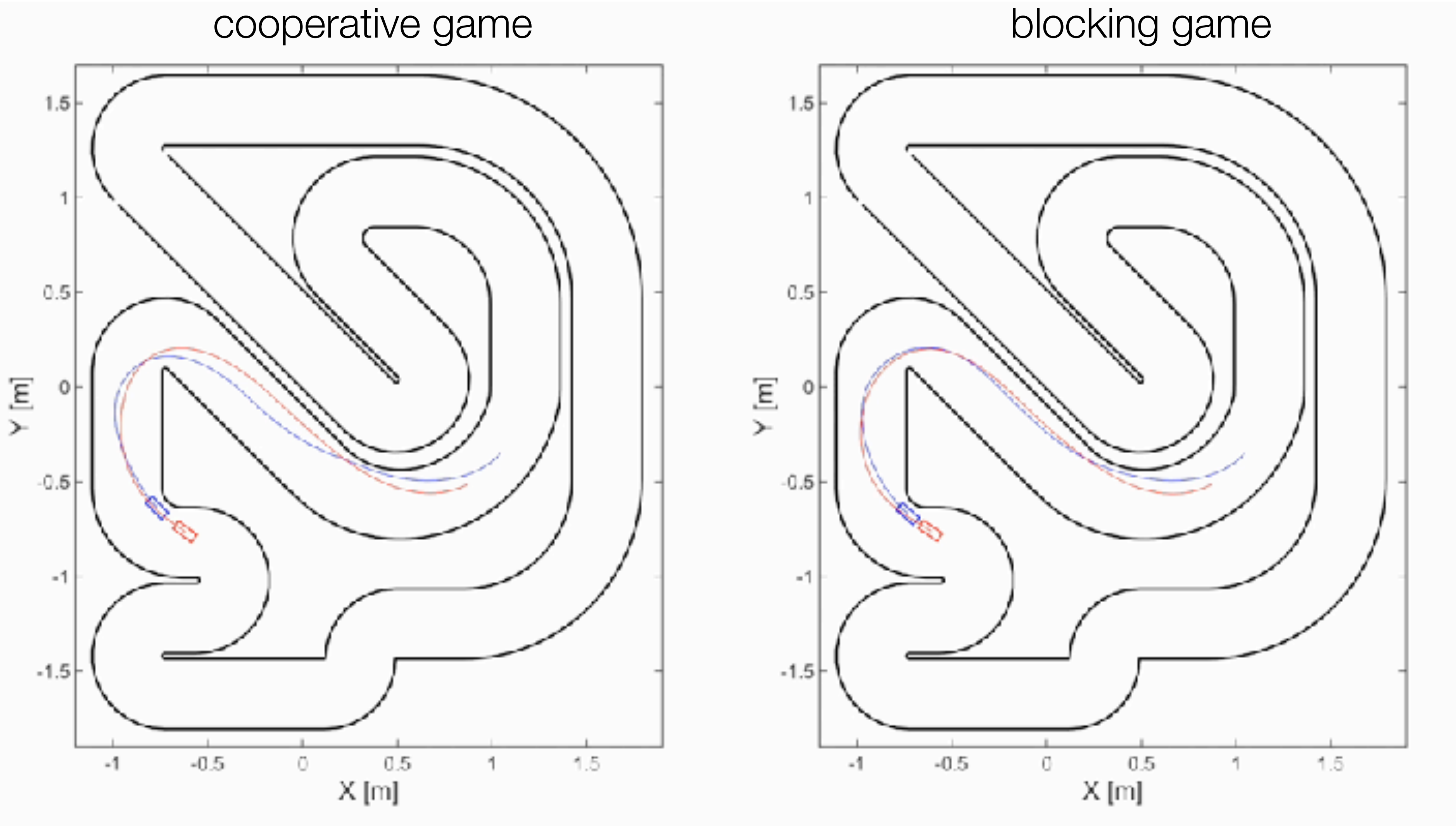
Simulation



Viab -> aggressive driver

Disc -> cautious driver

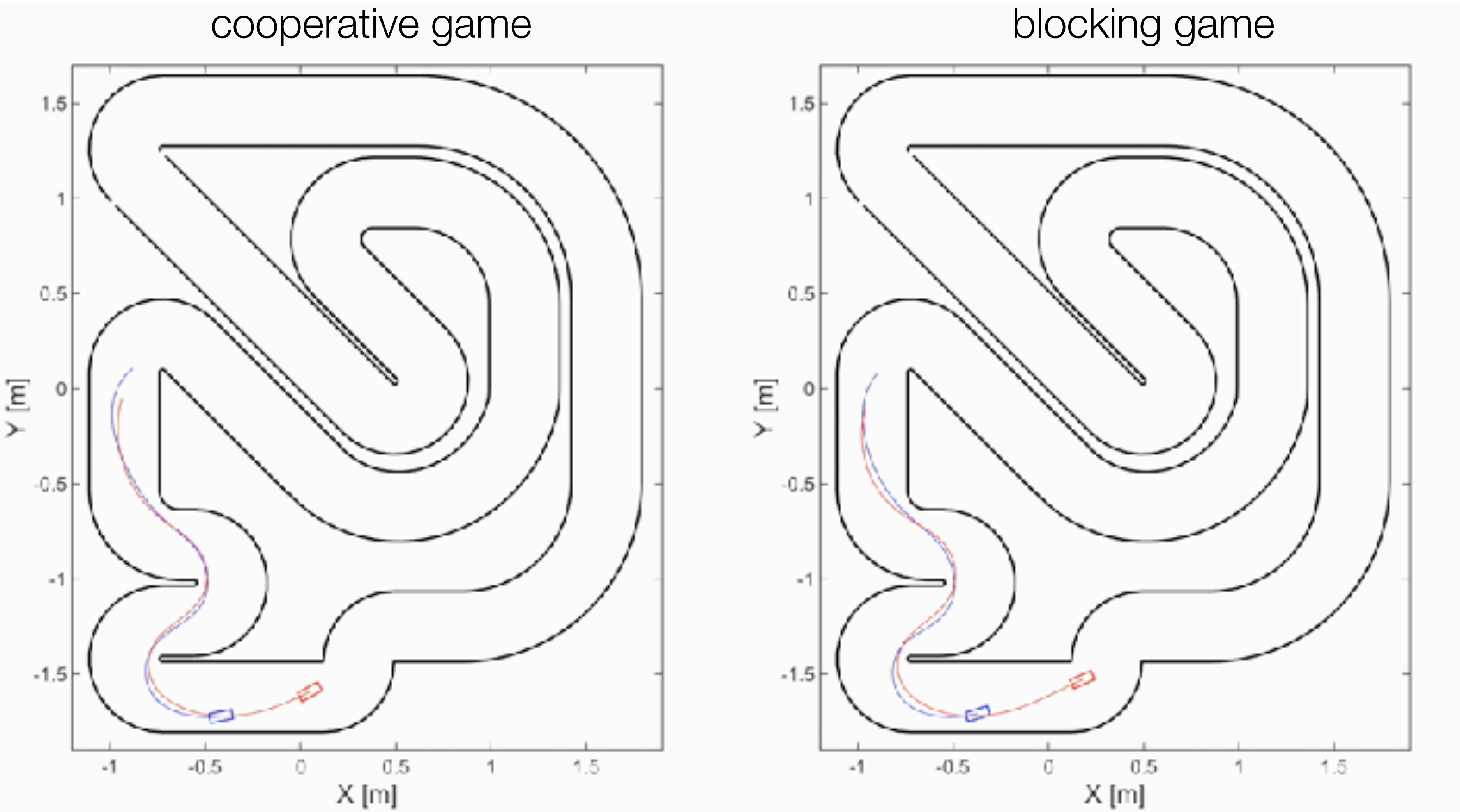
Simulation



Viab -> aggressive driver

Disc -> cautious driver

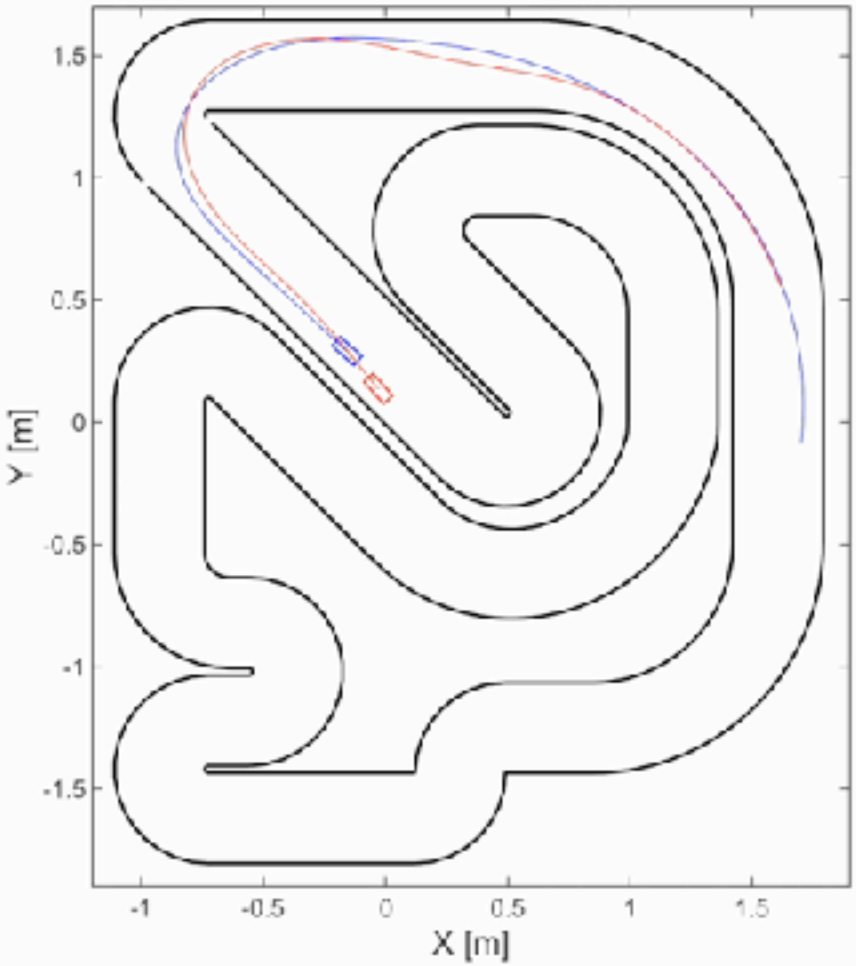
Simulation



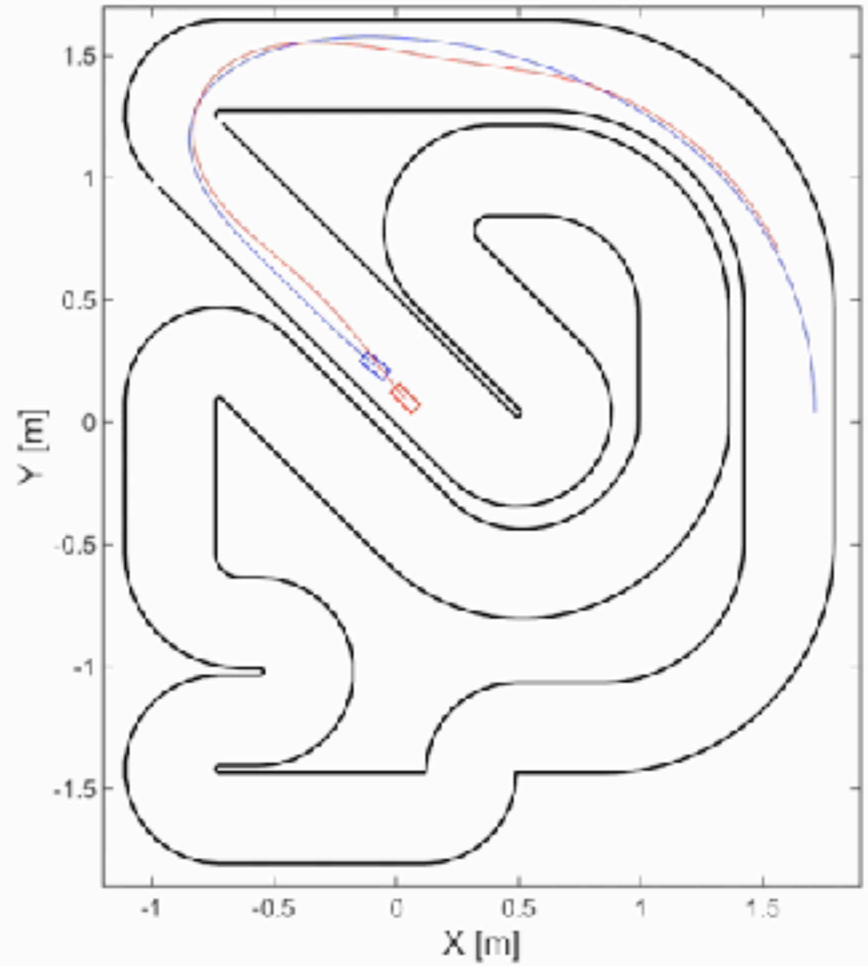
Viab -> aggressive driver   Disc -> cautious driver

Simulation

cooperative game



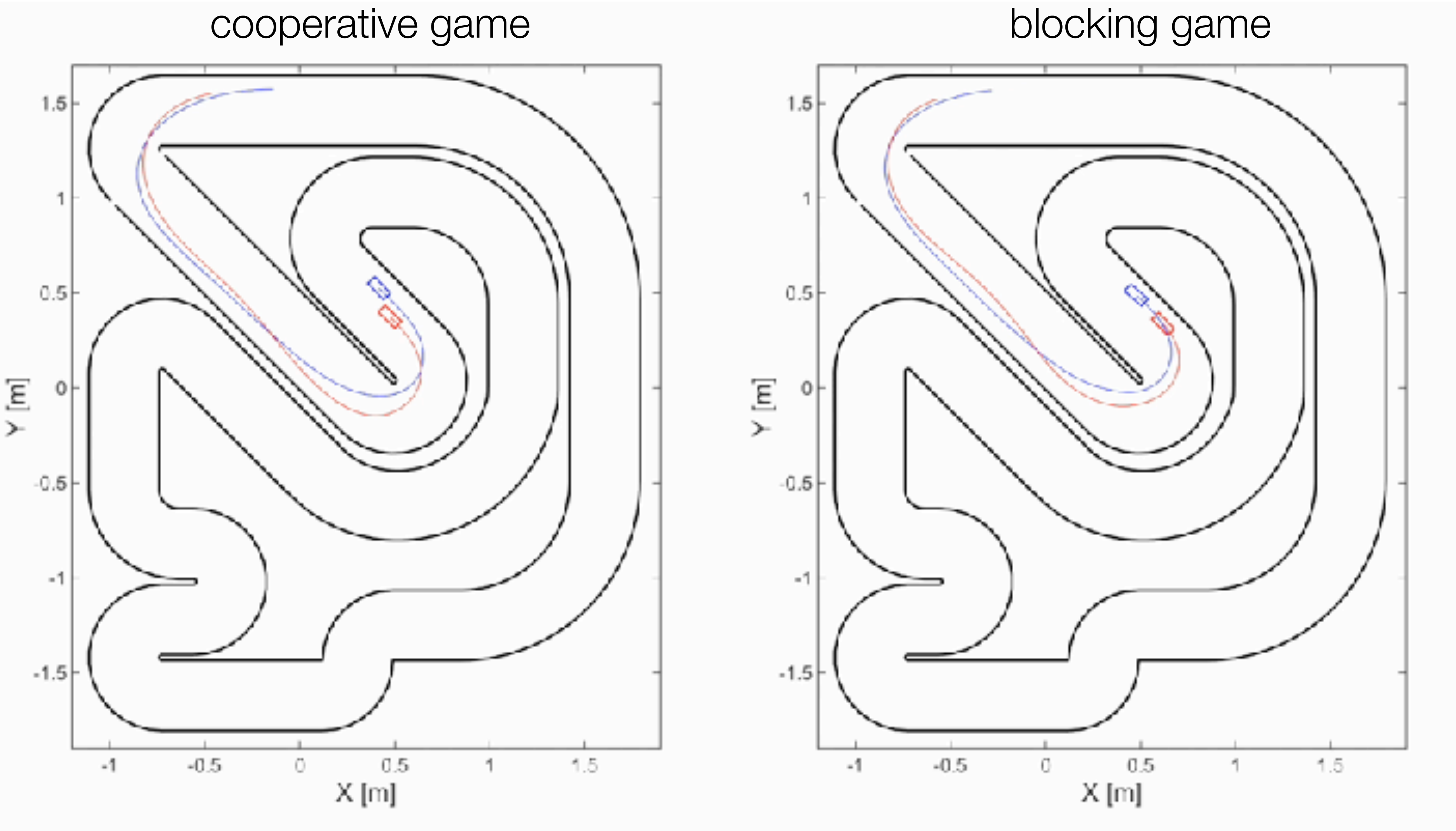
blocking game



Viab -> aggressive driver

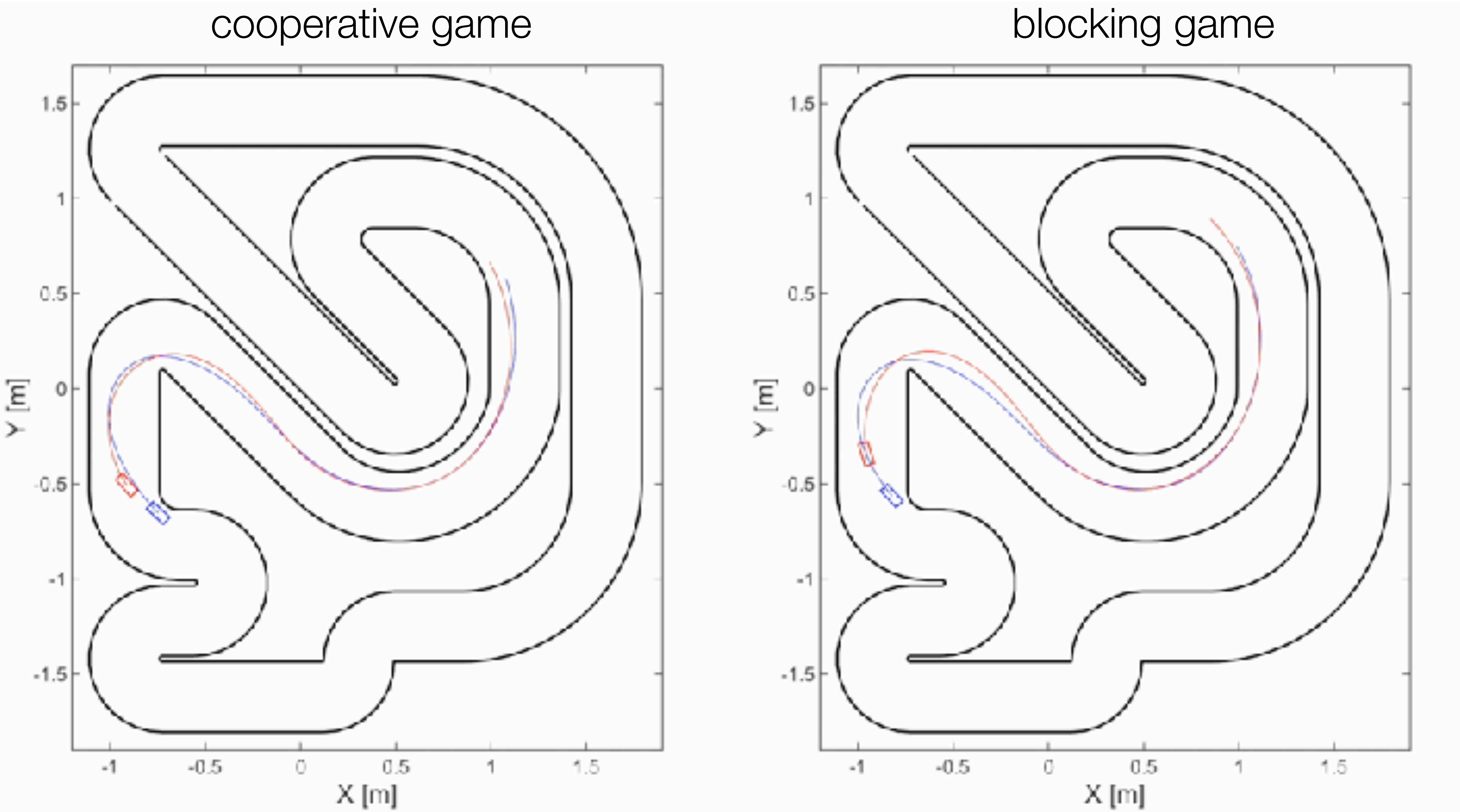
Disc -> cautious driver

Simulation



Viab -> aggressive driver
Disc -> cautious driver

Simulation



Viab -> aggressive driver

Disc -> cautious driver

Experimental Results

blocking game - $N_S = 2$

Experimental Results

blocking game - $N_S = 2$

Summary and Outlook

- We proposed a complet autonomous racing pipeline
  - Different behavior is seen for different viability kernels and games
- Interesting insights into behavior of non-cooperative decisions
  - Sequential maximization and leader-follower structure

Summary and Outlook

- We proposed a complet autonomous racing pipeline
  - Different behavior is seen for different viability kernels and games
- Interesting insights into behavior of non-cooperative decisions
  - Sequential maximization and leader-follower structure

Summary and Outlook

- We proposed a complet autonomous racing pipeline
  - Different behavior is seen for different viability kernels and games
- Interesting insights into behavior of non-cooperative decisions
  - Sequential maximization and leader-follower structure
- Reliable and real-time feasible games
  - Consider uncertainty in game formulation
  - Reinforcement learning-based terminal cost+constraints
  - High-performance implementation using GPU

Summary and Outlook

- We proposed a complet autonomous racing pipeline
  - Different behavior is seen for different viability kernels and games
- Interesting insights into behavior of non-cooperative decisions
  - Sequential maximization and leader-follower structure
- Reliable and real-time feasible games
  - Consider uncertainty in game formulation
  - Reinforcement learning-based terminal cost+constraints
  - High-performance implementation using GPU
- Model-learning for MPC
  - Learning model correction can be massively improve performance

$$x_{k+1} = f(x_k, u_k) + \mu_{\mathrm{GP}}(x_k, u_k)$$

Questions