Noncooperative Game Theory for Autonomous Racing

Dr. Alexander Liniger

Game-theoretical Motion Planning

ICRA 2021 - Tutorial

ETH zürich

CVL Computer Vision Lab

Automatic Control Laboratory IfA

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
  - IndyAutonomous
- **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
  - IndyAutonomous
- **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
  - IndyAutonomous
- **Structured** but **competitive** environment

Autonomous Racing Challenges

Driving at the handling limit

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

Staying safe inside the track

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

Interacting with other race cars

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

Autonomous Racing Challenges

Driving at the handling limit

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

Staying safe inside the track

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

Interacting with other race cars

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

Autonomous Racing Challenges

Driving at the handling limit

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

Staying safe inside the track

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

Interacting with other race cars

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

Autonomous Racing Challenges

Driving at the handling limit

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

Staying safe inside the track

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

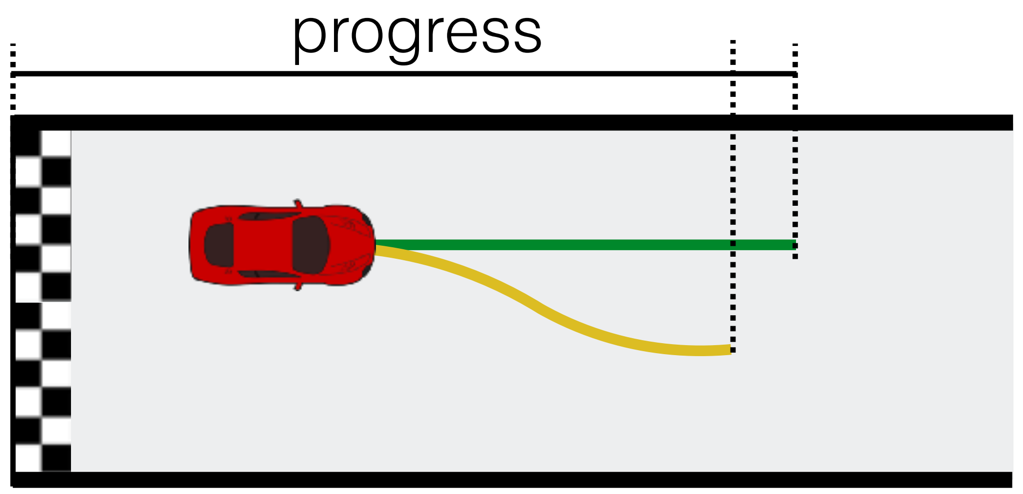
Interacting with other race cars

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

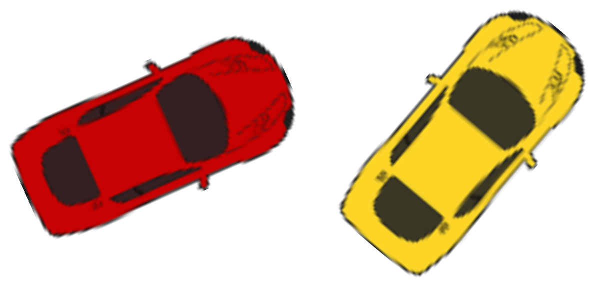
Racing Ingredients

Finish first

- Approximated by maximizing progress
- Generates racing trajectories



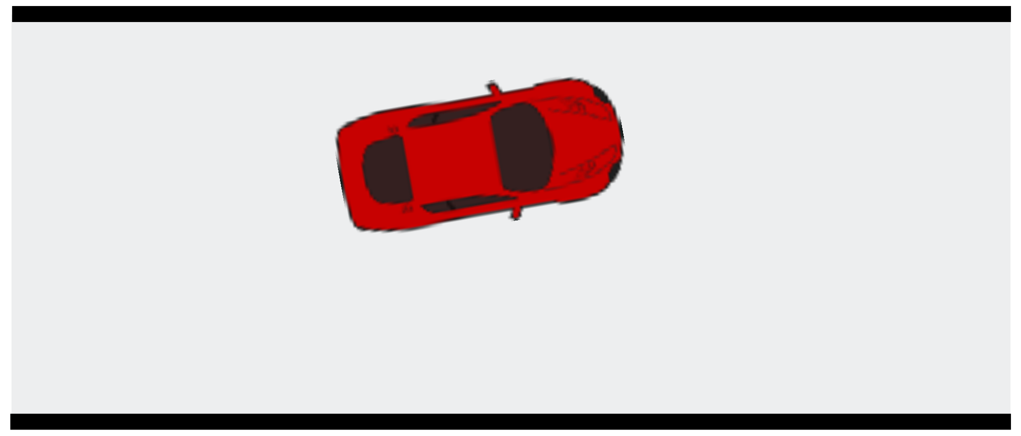
Do not collide with other cars



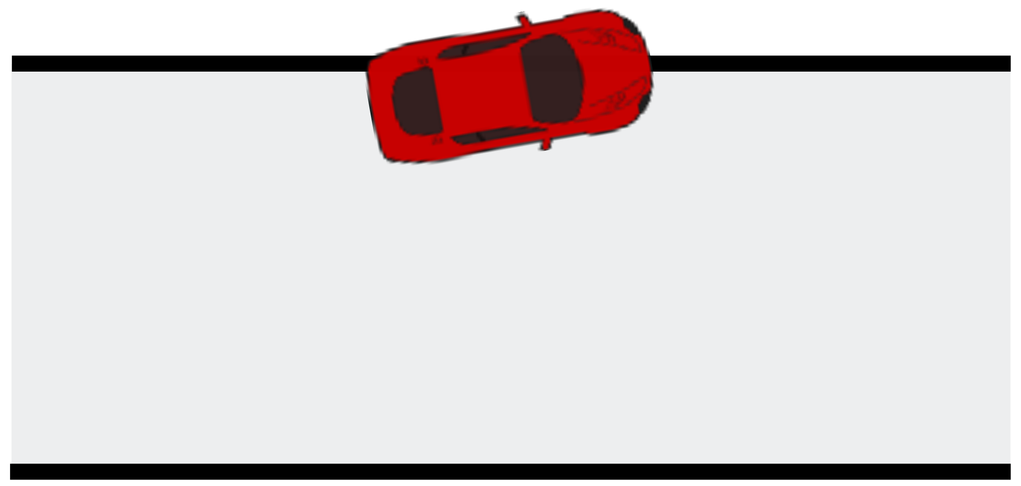
good

bad

Stay inside the track

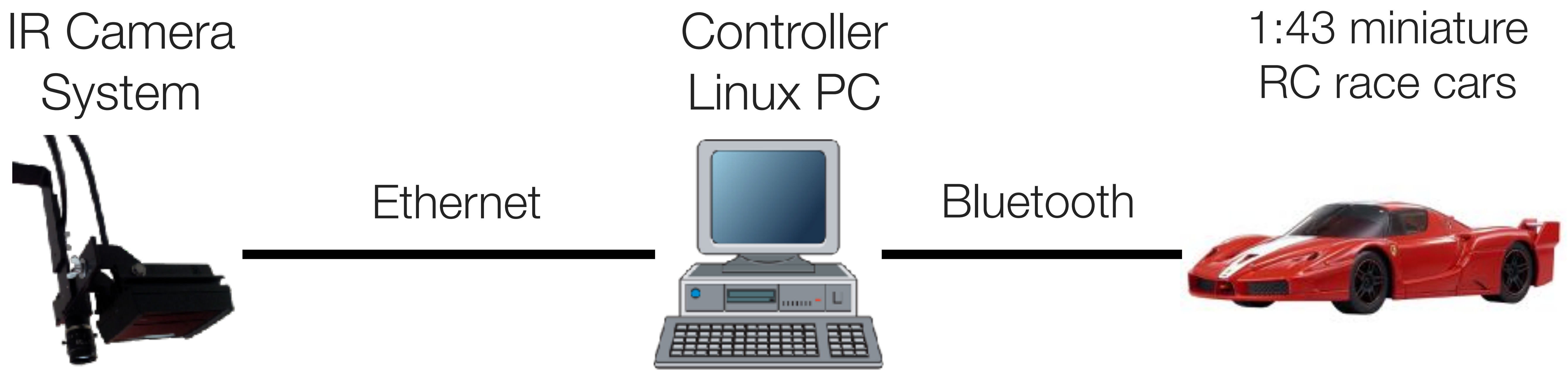


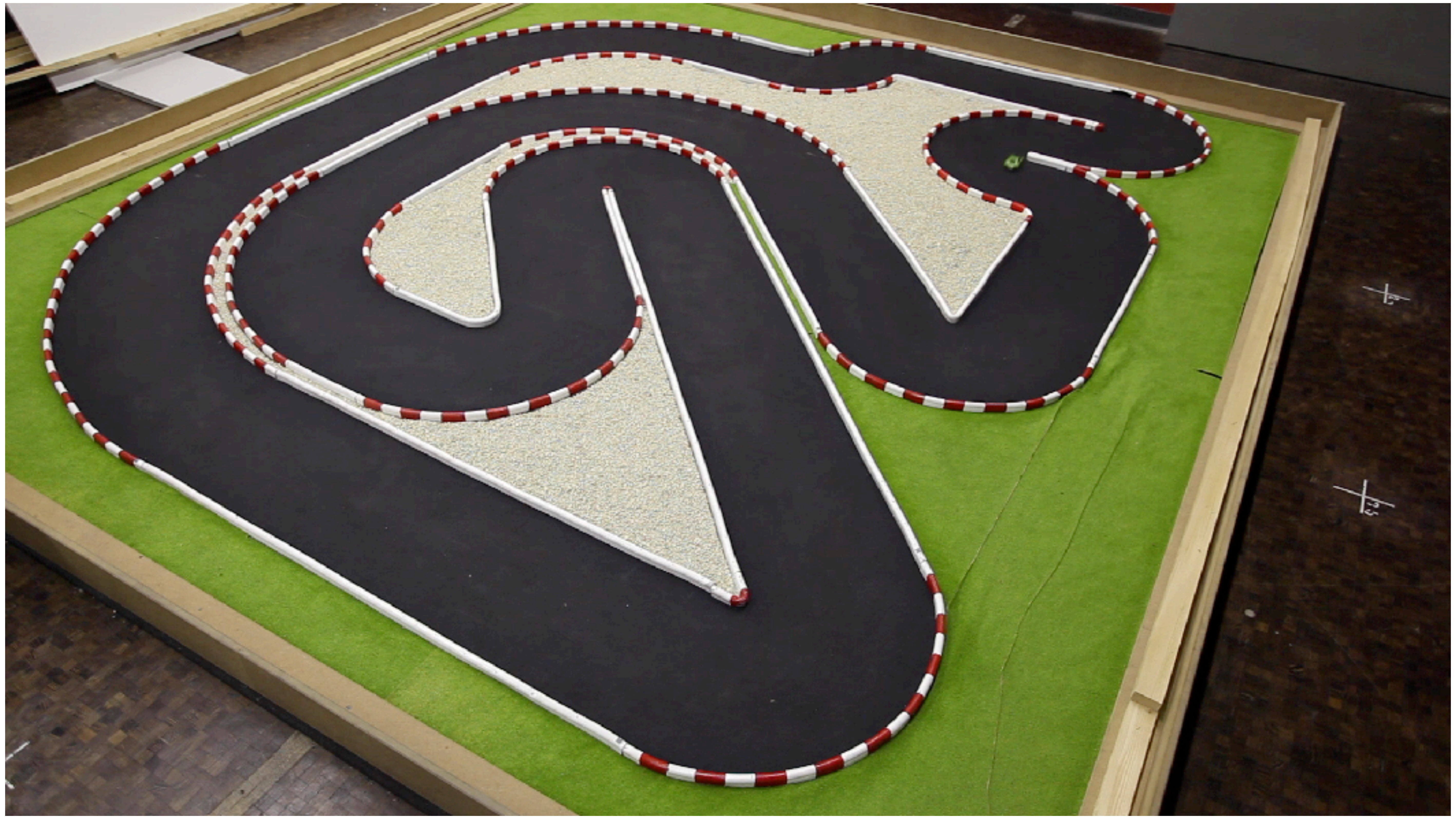
good



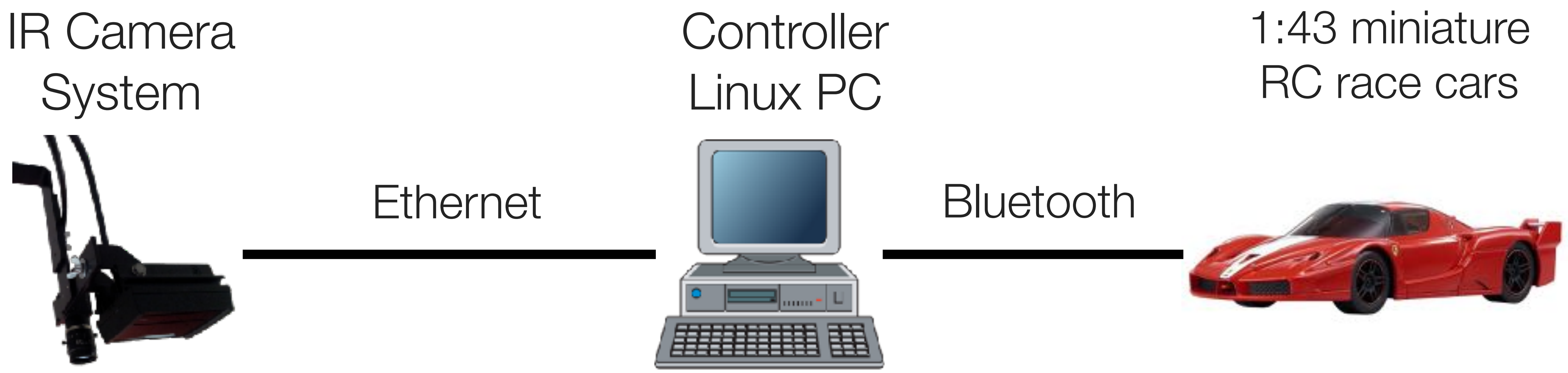
bad

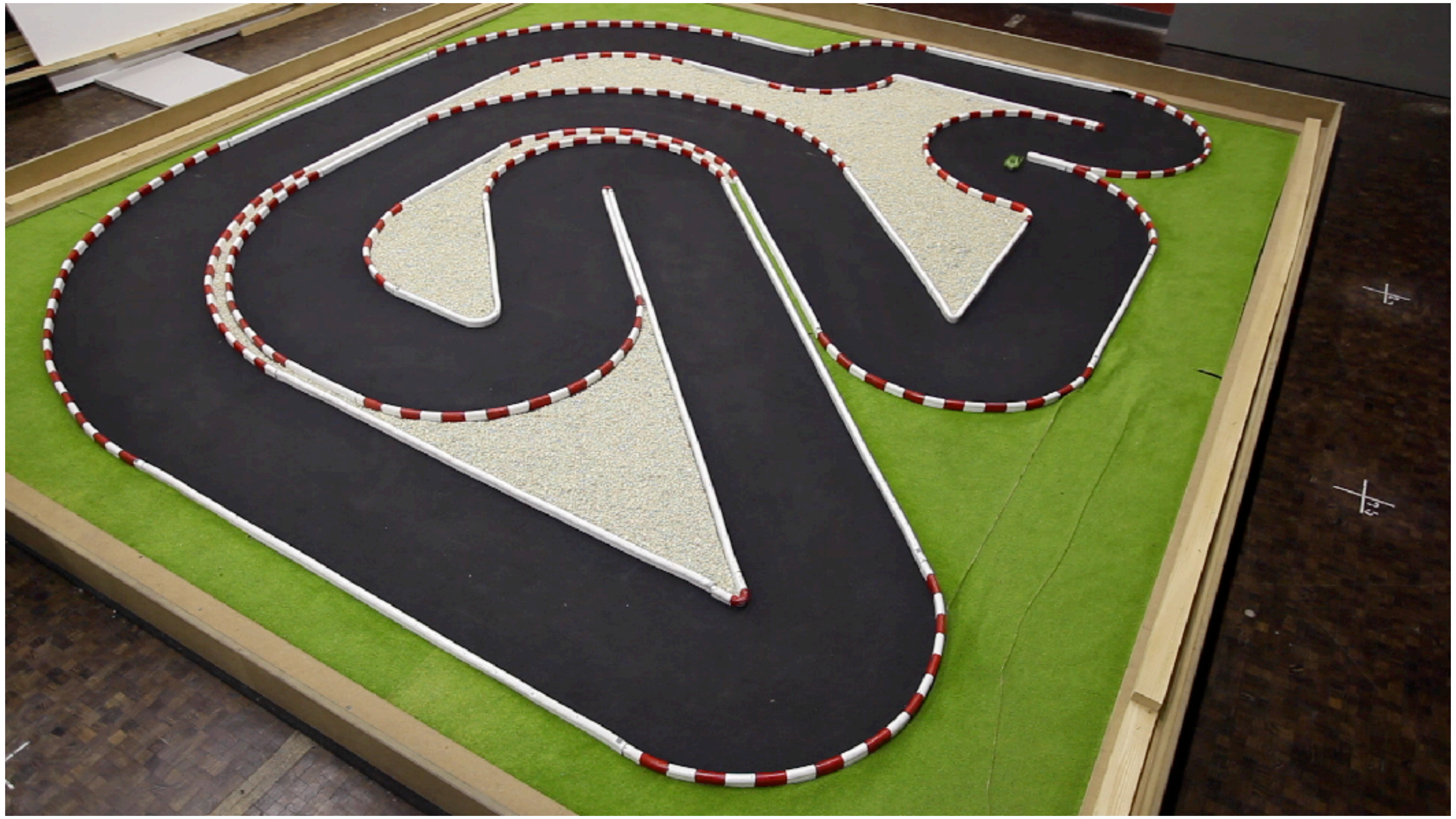
Experimental Set-Up





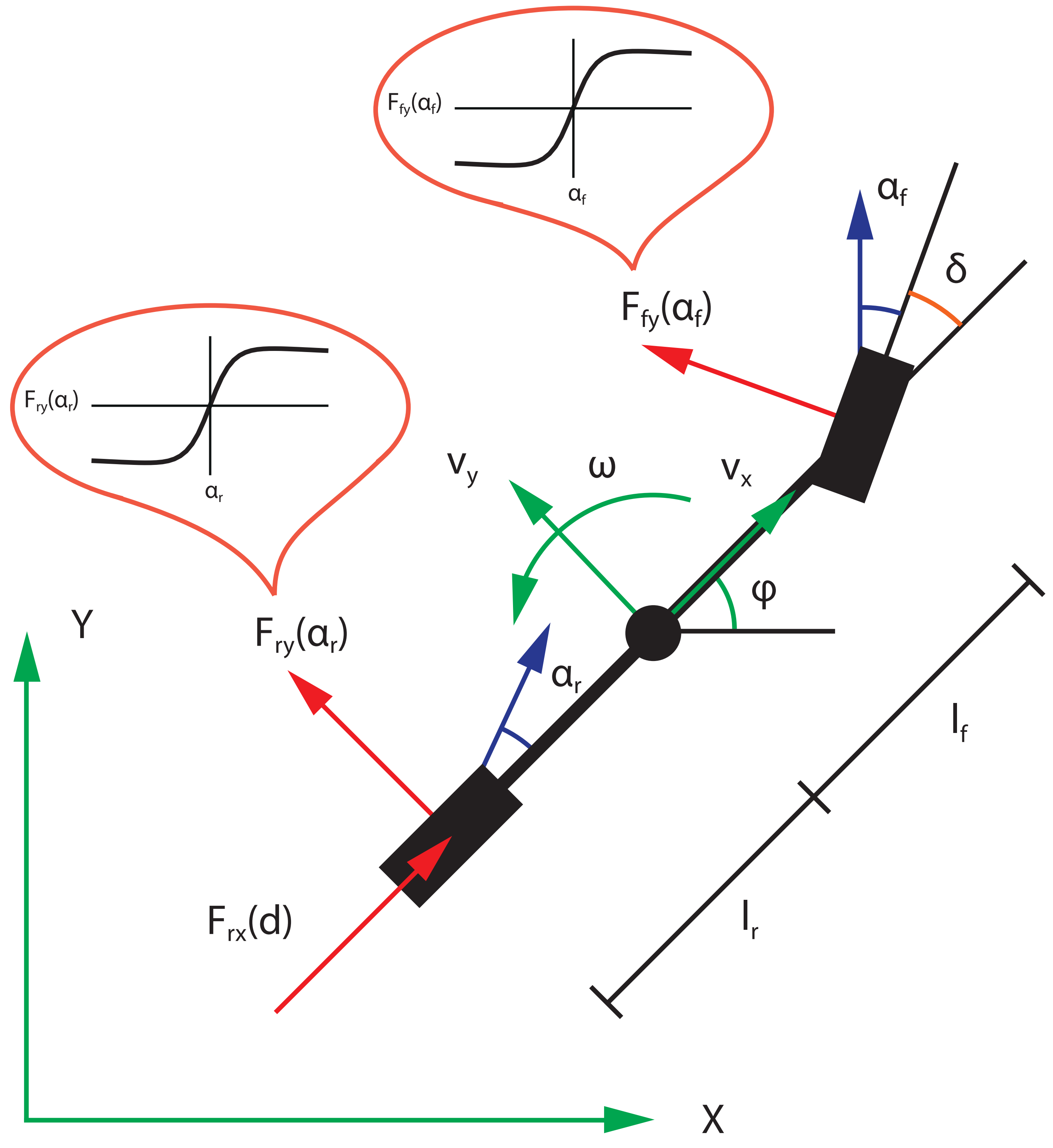
Experimental Set-Up





Car Model

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

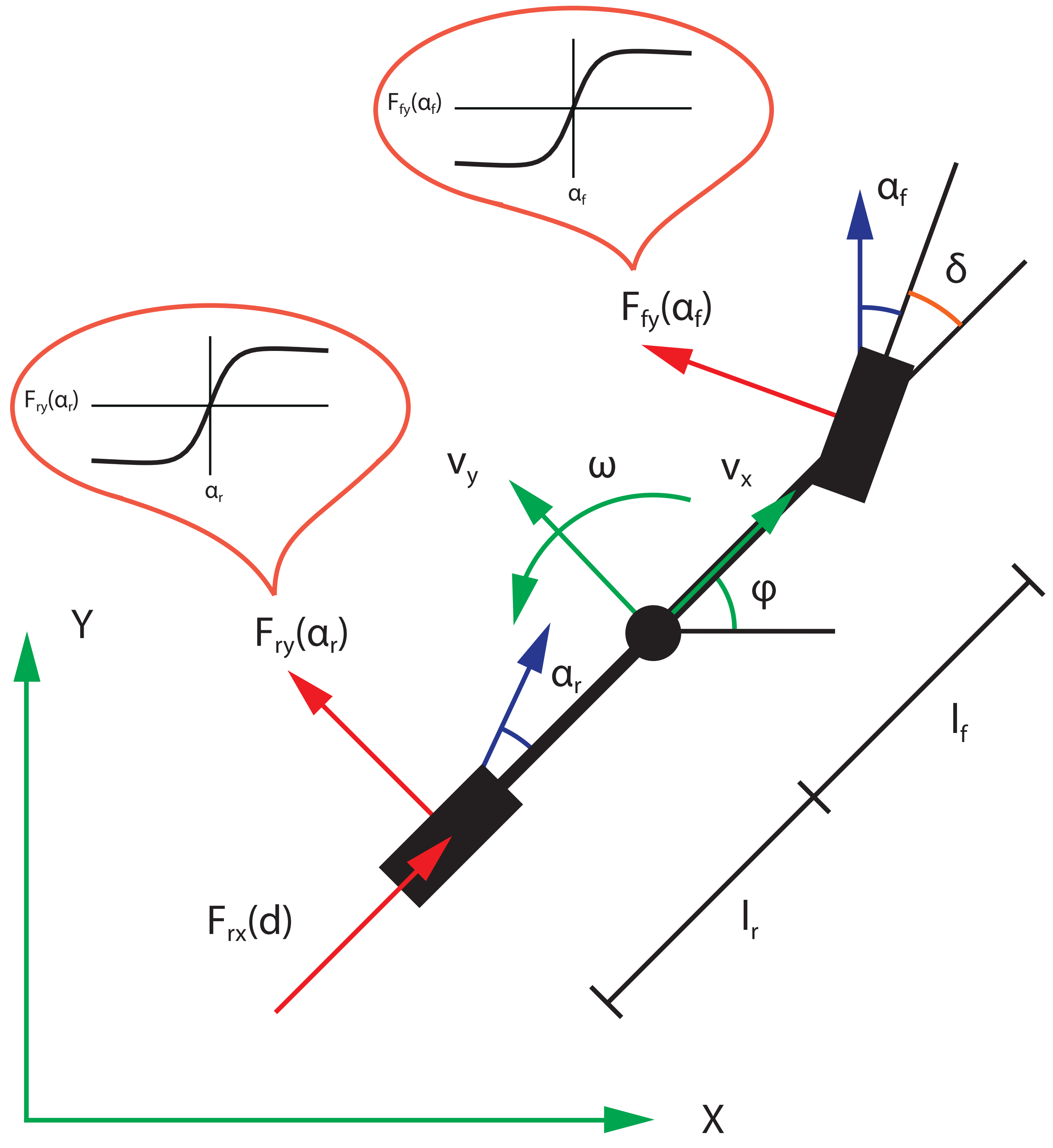


$$\dot{X} = v_x \cos(\varphi) - v_y \sin(\varphi)$$
$$\dot{Y} = v_x \sin(\varphi) + v_y \cos(\varphi)$$
$$\dot{\varphi} = \omega$$
$$\dot{v}_x = \frac{1}{m}(F_{r,x} - F_{f,y} \sin\delta + m v_y \omega)$$
$$\dot{v}_y = \frac{1}{m}(F_{r,y} + F_{f,y} \cos\delta - m v_x \omega)$$
$$\dot{\omega} = \frac{1}{I_z}(F_{f,y}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

- Highly nonlinear 6 dimensional system

Car Model

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

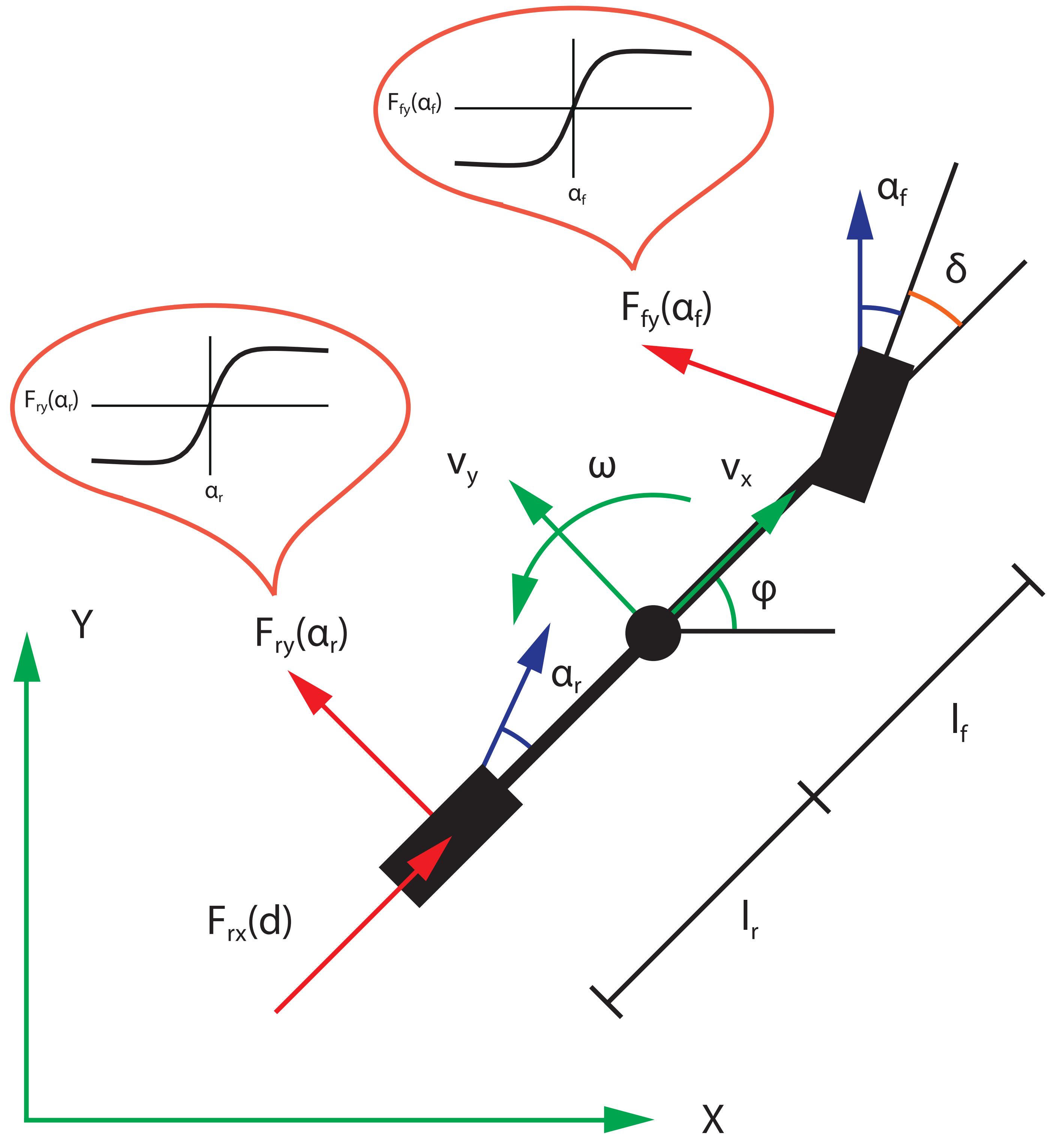


$$
\begin{aligned}
\dot{X} &= v_x \cos(\varphi) - v_y \sin(\varphi) \\
\dot{Y} &= v_x \sin(\varphi) + v_y \cos(\varphi) \\
\dot{\varphi} &= \omega \\
\dot{v}_x &= \frac{1}{m}(F_{r,x} - F_{f,y} \sin\delta + m v_y \omega) \\
\dot{v}_y &= \frac{1}{m}(F_{r,y} + F_{f,y} \cos\delta - m v_x \omega) \\
\dot{\omega}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

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

Car Model

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



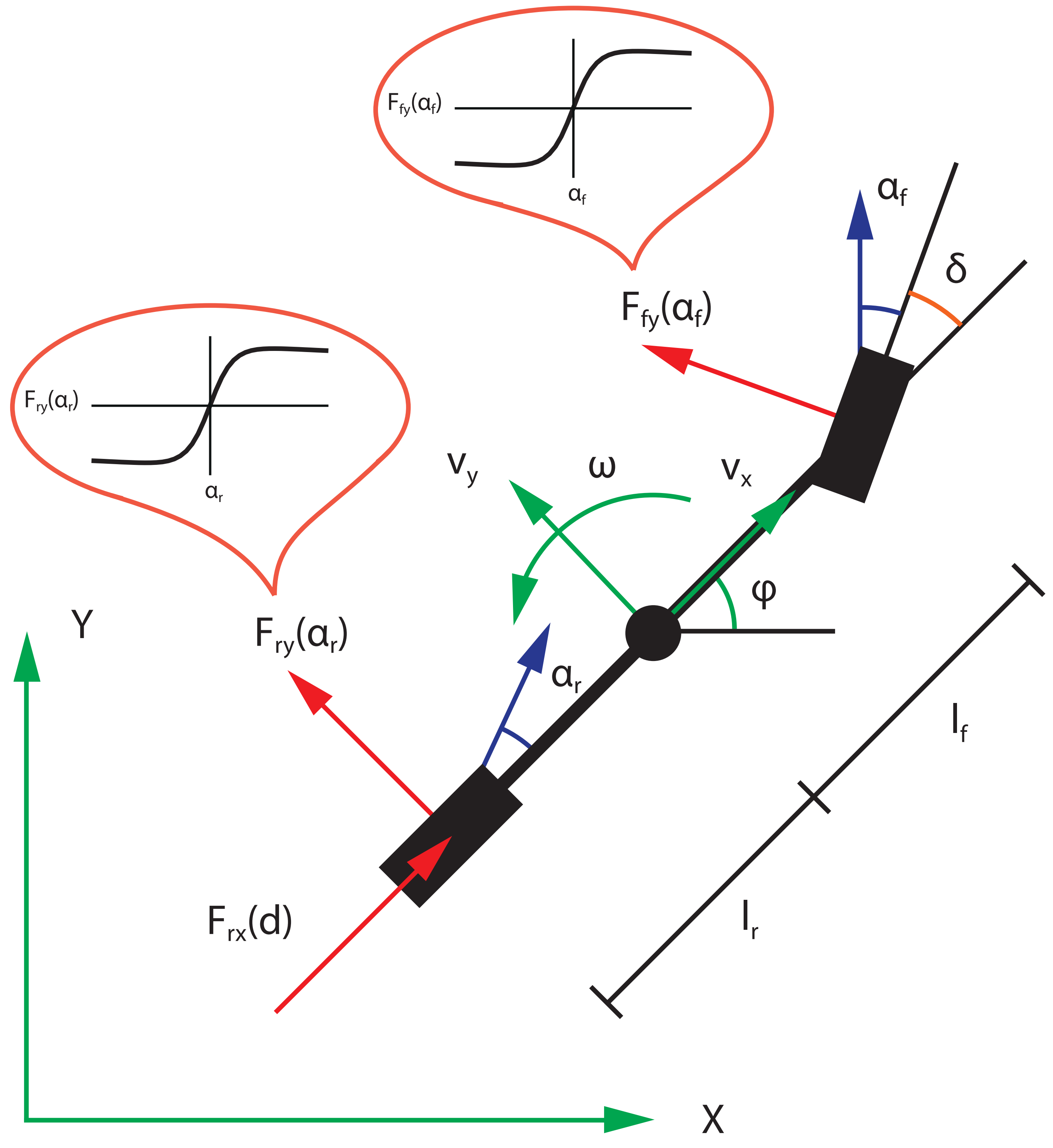
$$\dot{X} = v_x \cos(\varphi) - v_y \sin(\varphi)$$
$$\dot{Y} = v_x \sin(\varphi) + v_y \cos(\varphi)$$
$$\dot{\varphi} = \omega$$

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

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

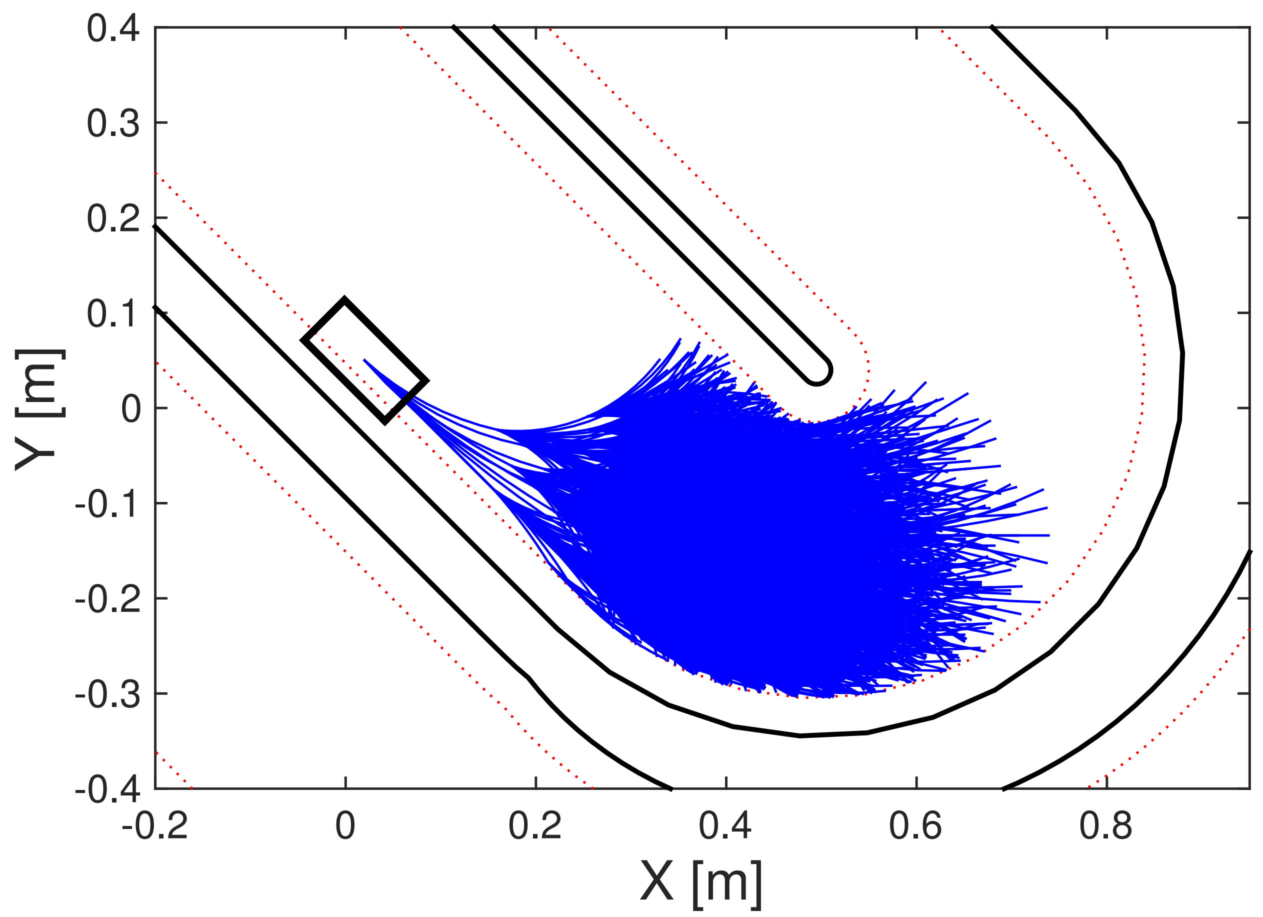
Hierarchical Control Structure

High-level motion planning based on constant velocities primitives

- Plan for slow dynamics
- Reduced dimension —> four dimensions instead of six
- Long discretization times —> 0.16s instead of 0.02s

MPC-based trajectory tracking

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



- 129 constant velocity motion primitives
- 3 prediction steps
- Lookahead of 0.48s

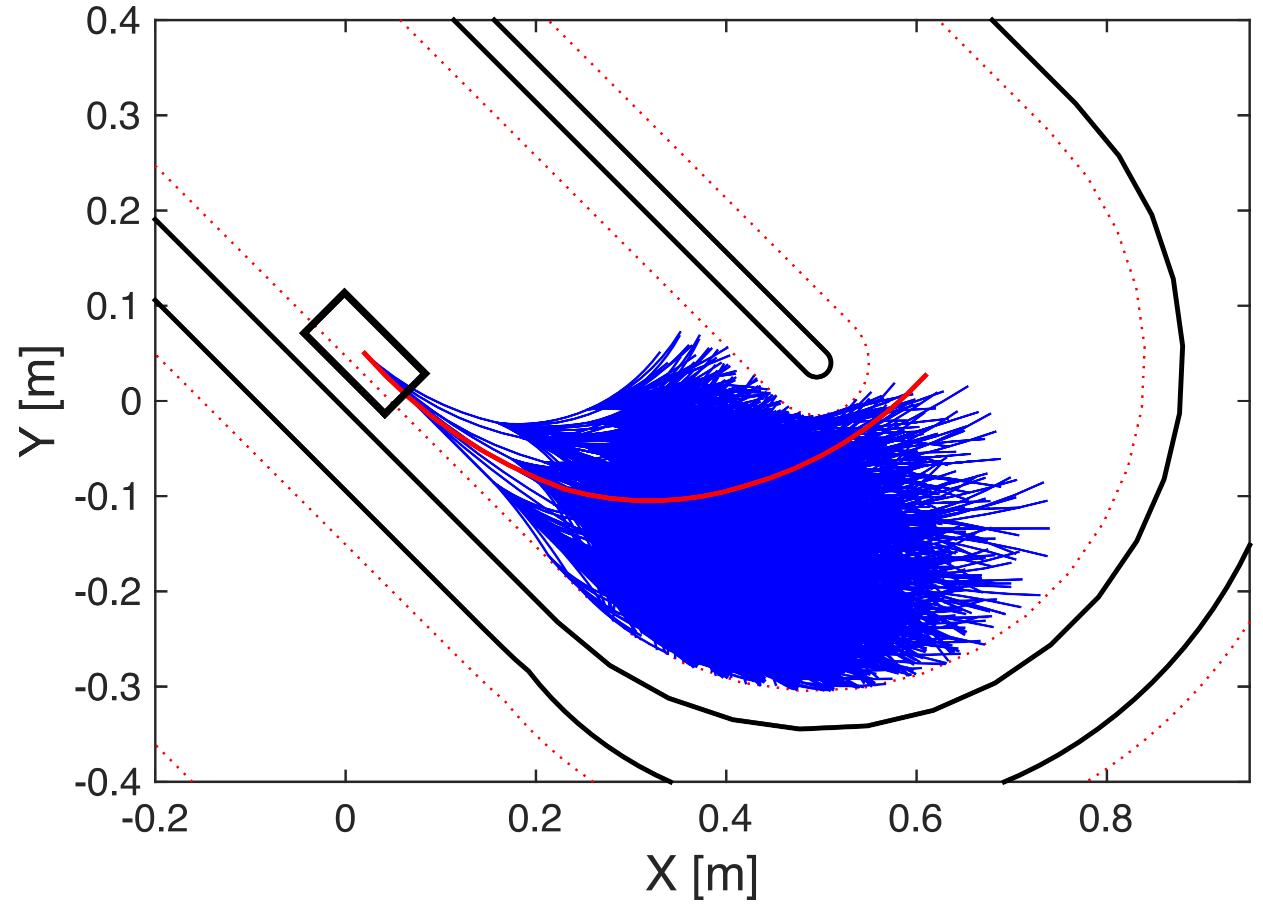
Hierarchical Control Structure

High-level motion planning based on constant velocities primitives

- Plan for slow dynamics
- Reduced dimension —> four dimensions instead of six
- Long discretization times —> 0.16s instead of 0.02s

MPC-based trajectory tracking

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



- 129 constant velocity motion primitives
- 3 prediction steps
- Lookahead of 0.48s

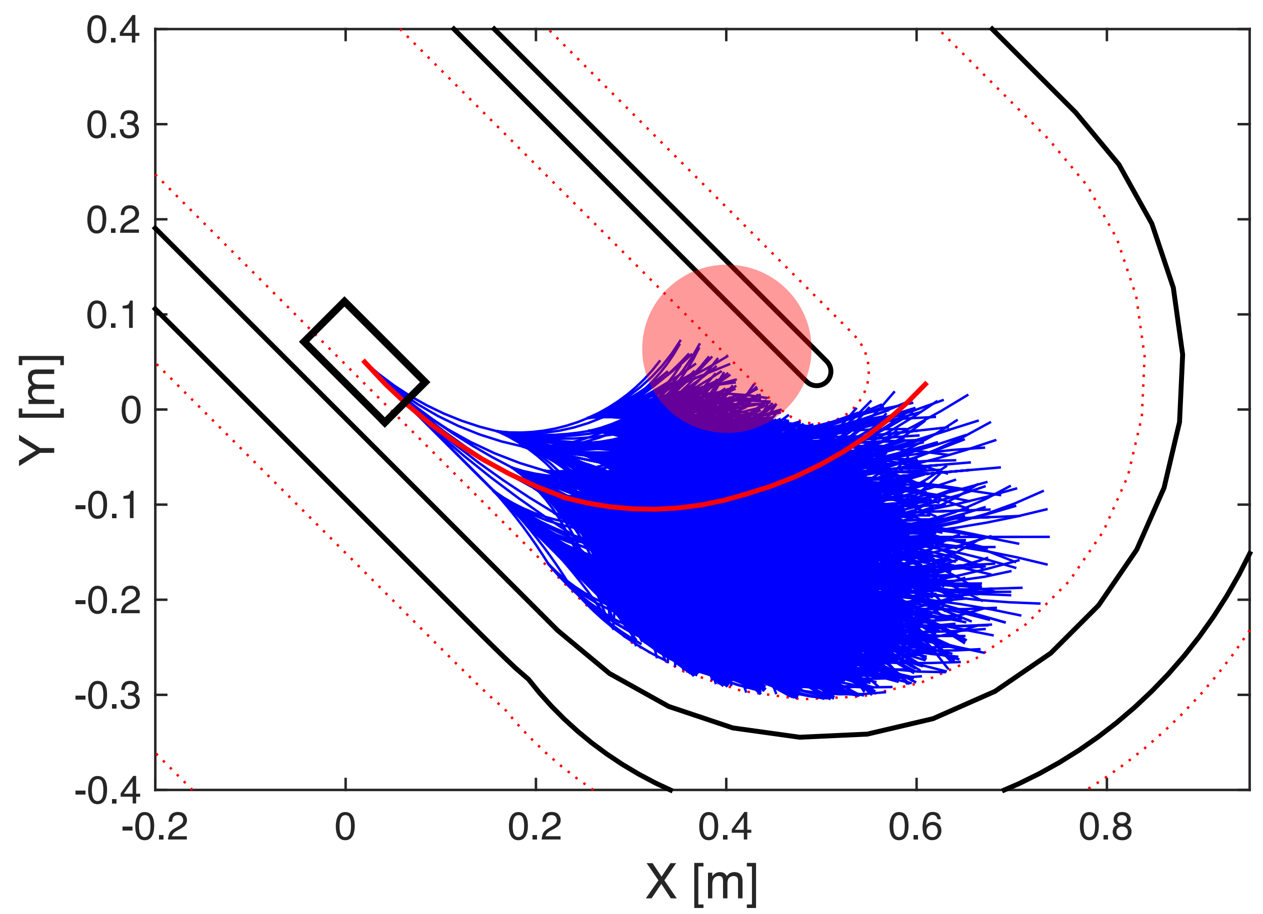
Hierarchical Control Structure

High-level motion planning based on constant velocities primitives

- Plan for slow dynamics
- Reduced dimension —> four dimensions instead of six
- Long discretization times —> 0.16s instead of 0.02s

MPC-based trajectory tracking

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



- 129 constant velocity motion primitives
- 3 prediction steps
- Lookahead of 0.48s

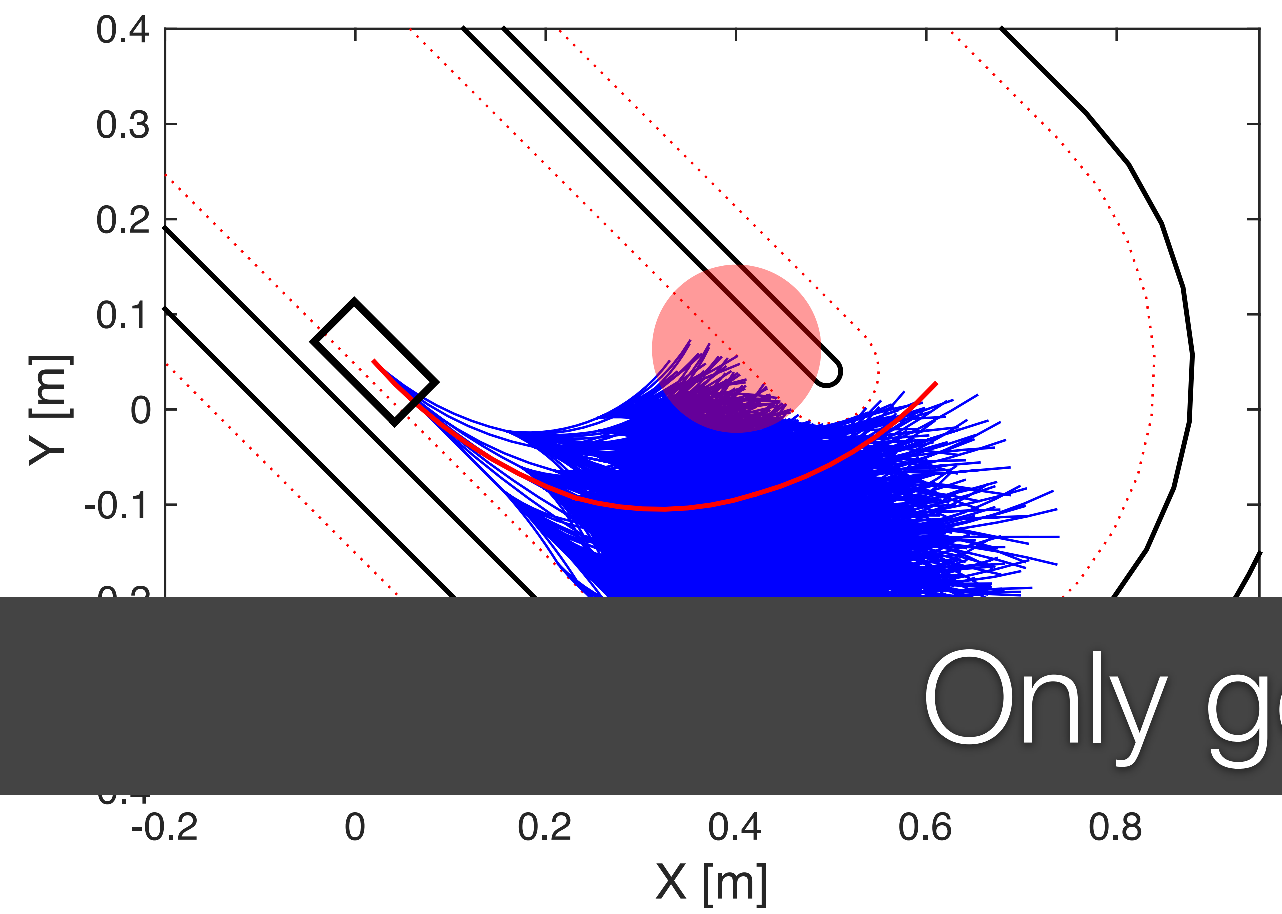
Hierarchical Control Structure

High-level motion planning based on constant velocities primitives

- Plan for slow dynamics
- Reduced dimension —> four dimensions instead of six
- Long discretization times —> 0.16s instead of 0.02s

MPC-based trajectory tracking

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



- 129 constant velocity motion primitives
- 3 prediction steps
- Lookahead of 0.48s

Only generate safe trajectories

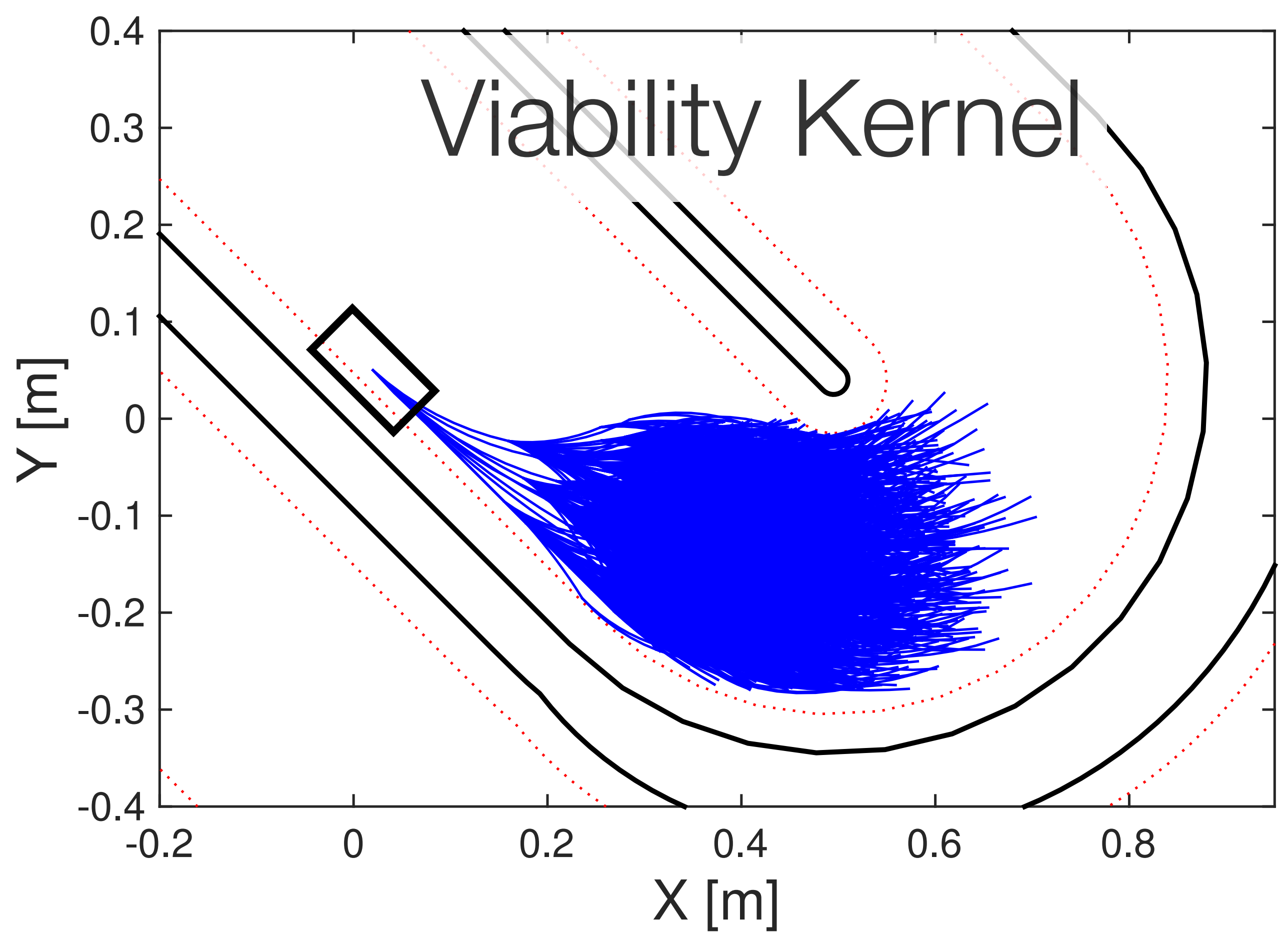
Hierarchical Control Structure

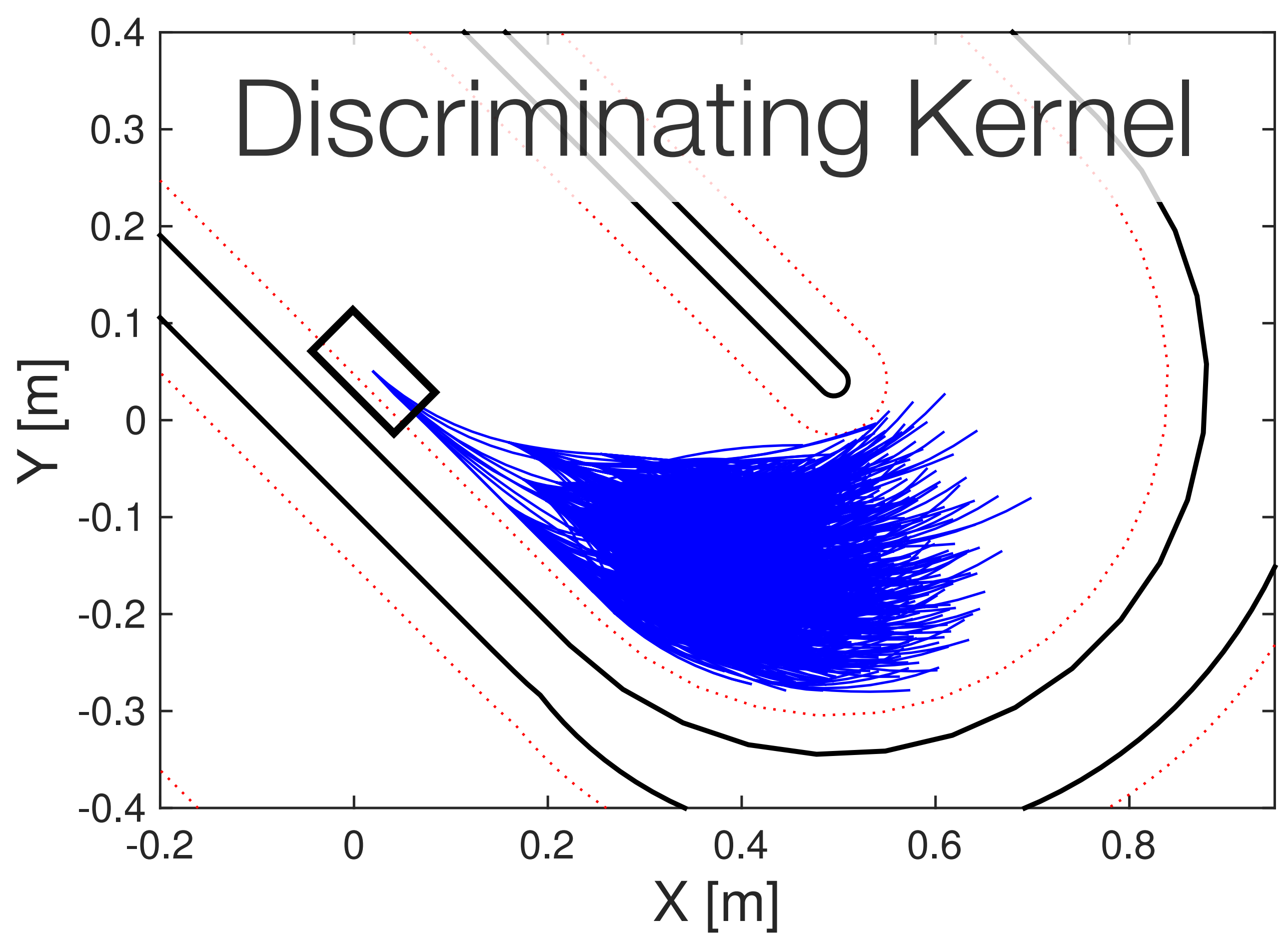
High-level motion planning based on constant velocities primitives

- Plan for slow dynamics
- Reduced dimension —> four dimensions instead of six
- Long discretization times —> 0.16s instead of 0.02s

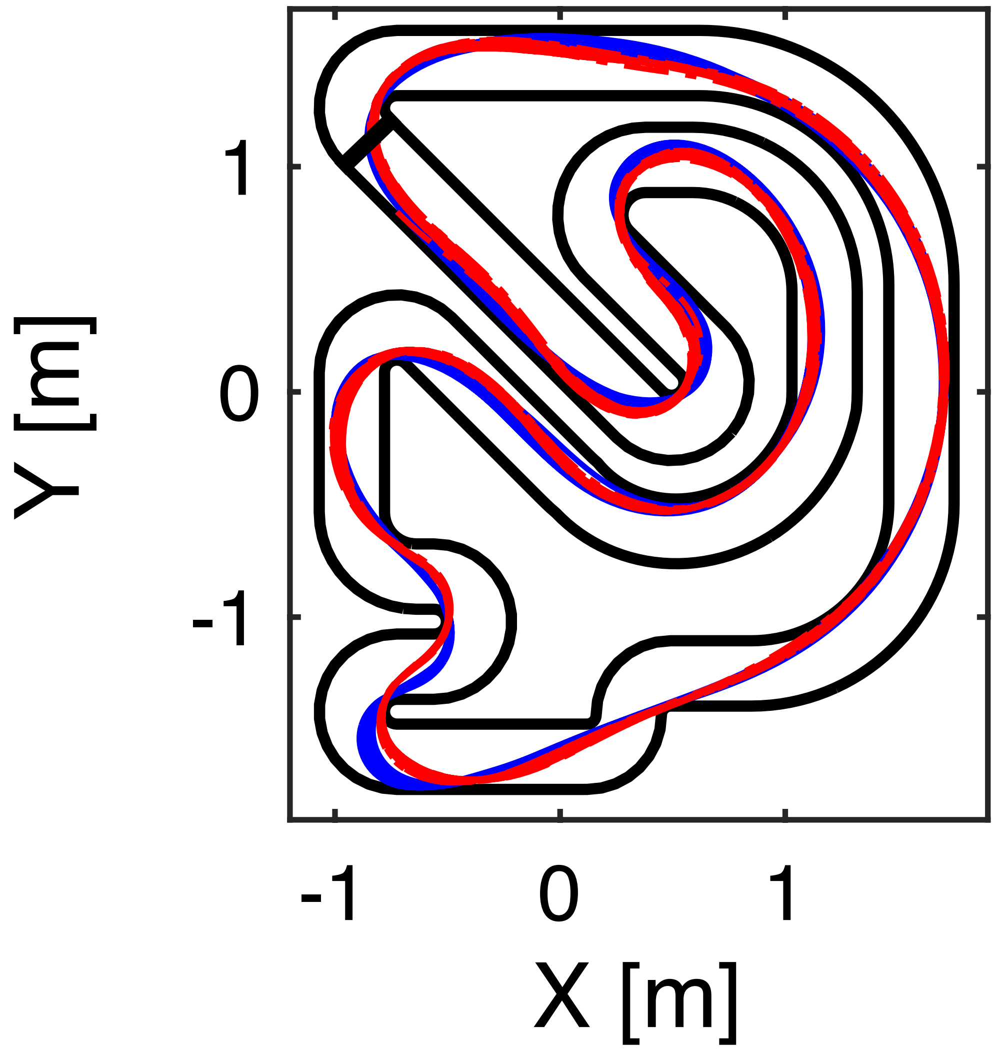
MPC-based trajectory tracking

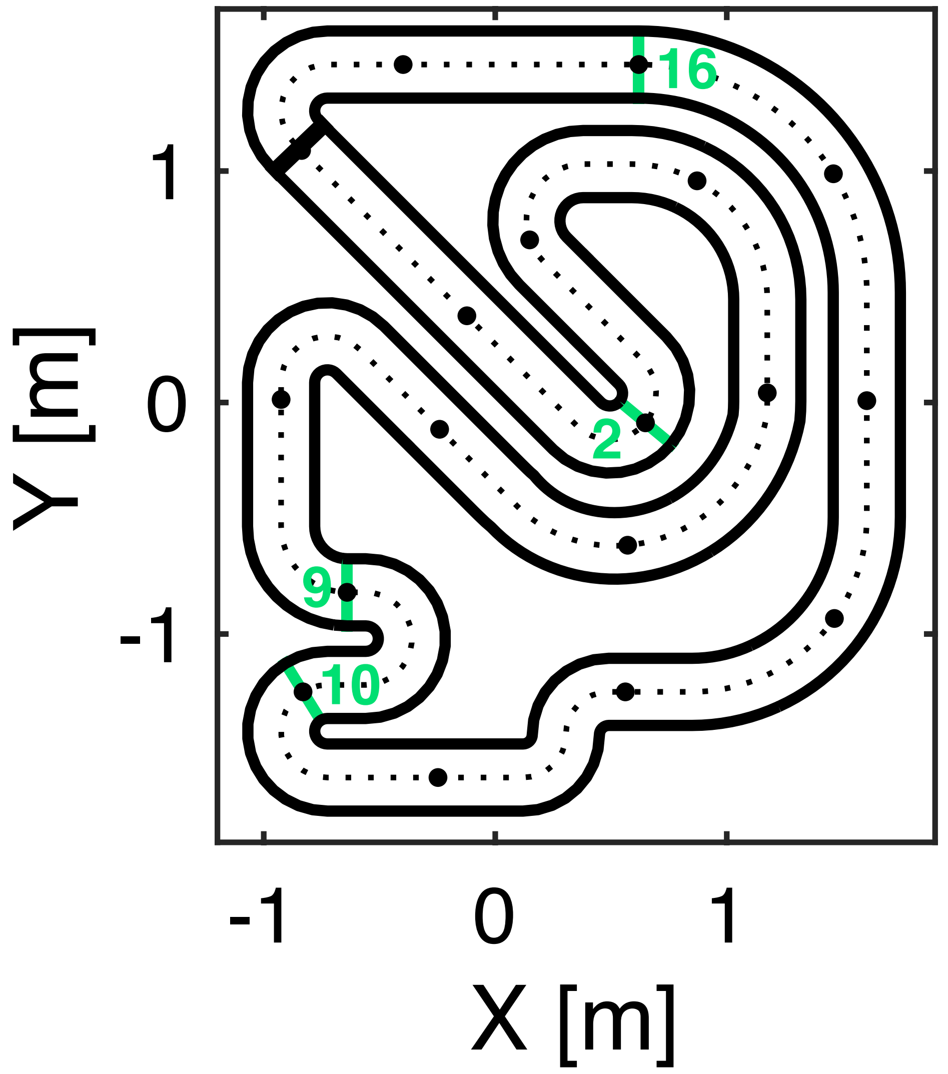
- Considering full dynamical bicycle model
- Linearization points given by path planner



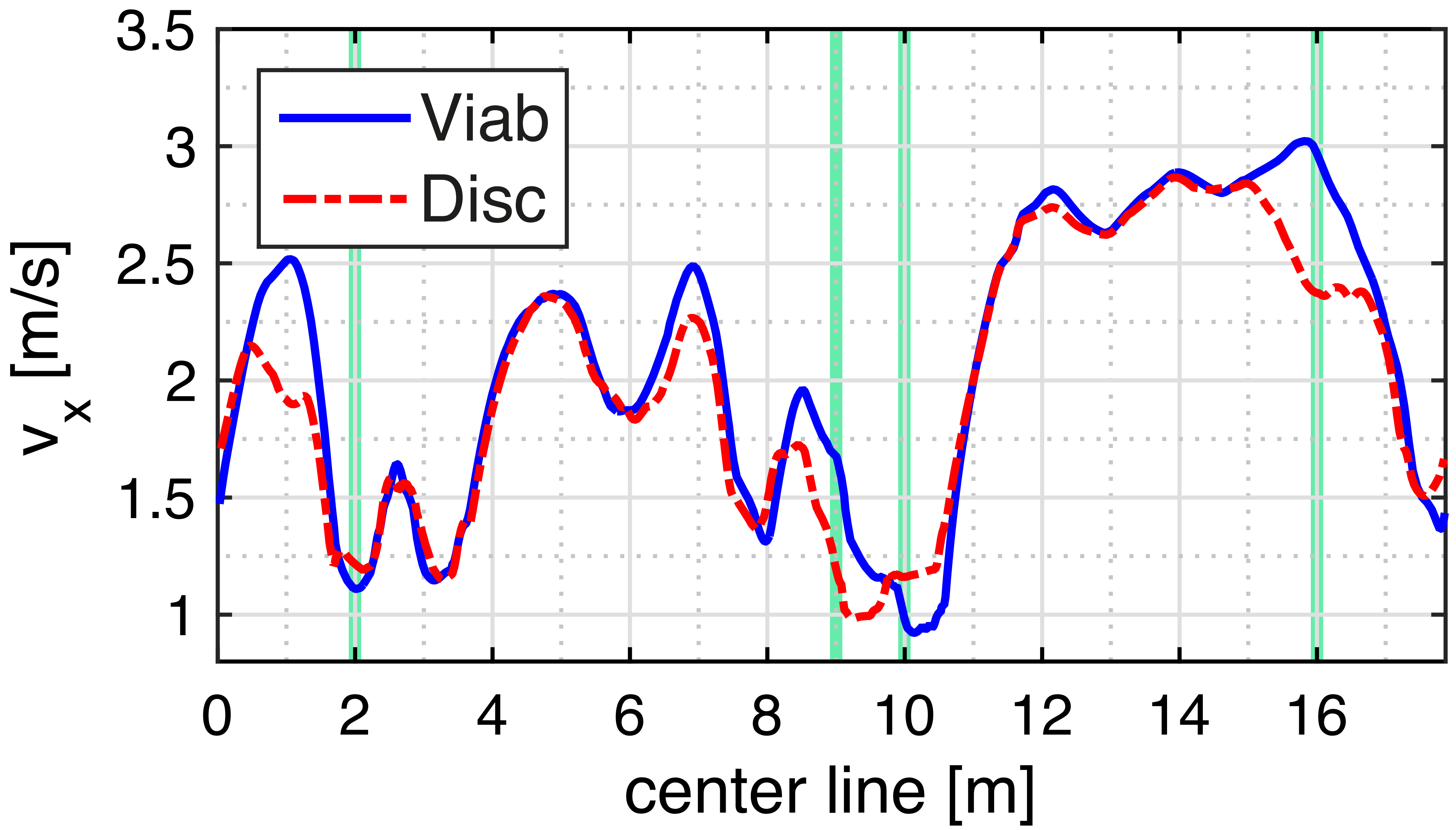


Simulation Results - Single Agent





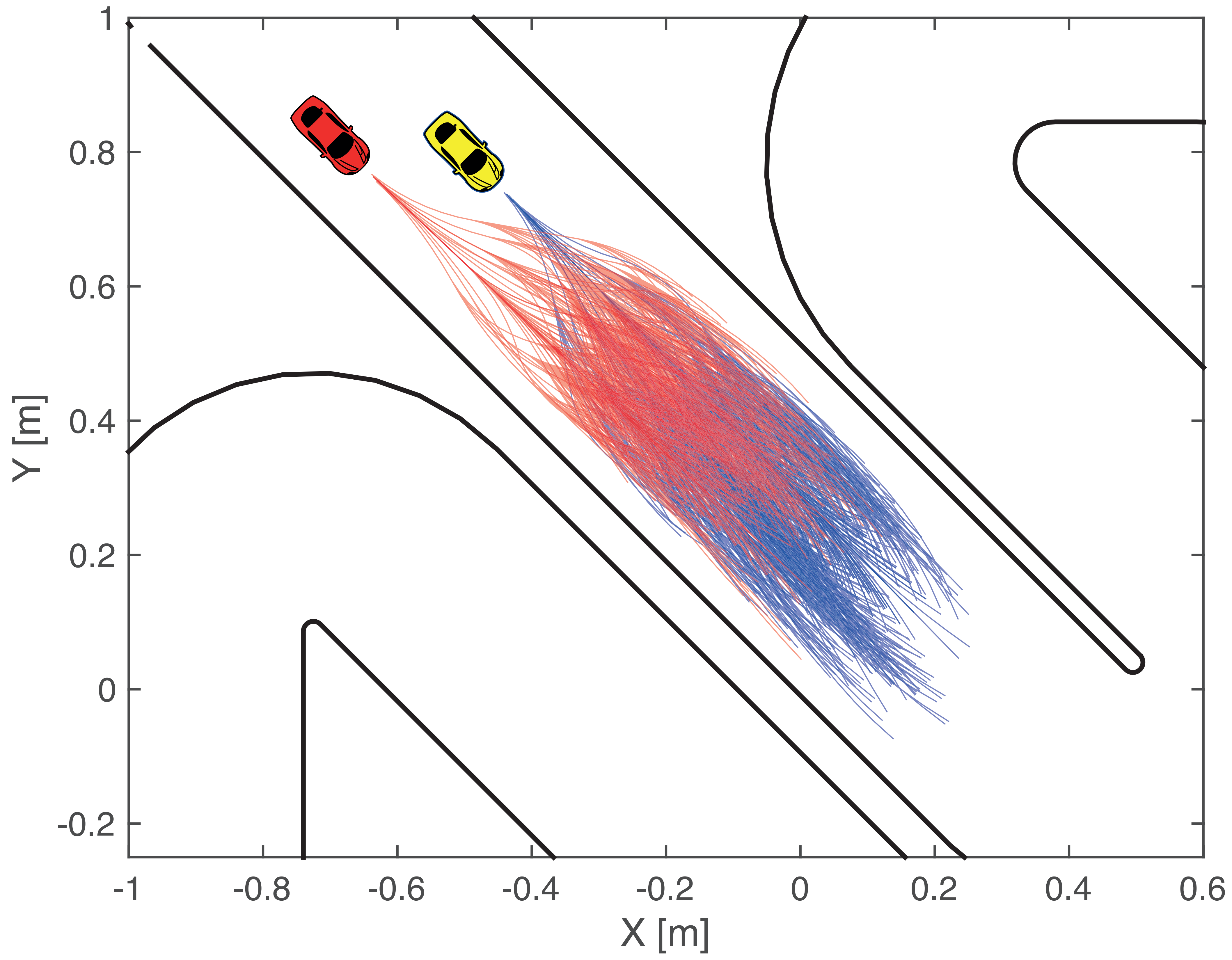
Kernel	mean lap time [s]	# constr. violations	median comp. time [ms]	max comp. time [ms]
No	8.76	4	32.26	247.7
Viab	8.57	0	0.904	7.968
Disc	8.60	1	0.870	7.533



- Both methods have the same lap time
- But use different driving styles
- Difference allows for interesting racing

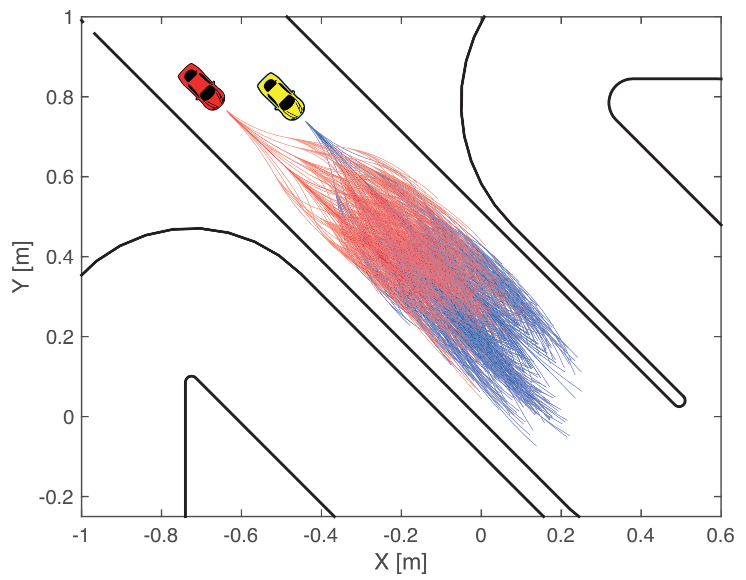
Bimatrix Racing Games

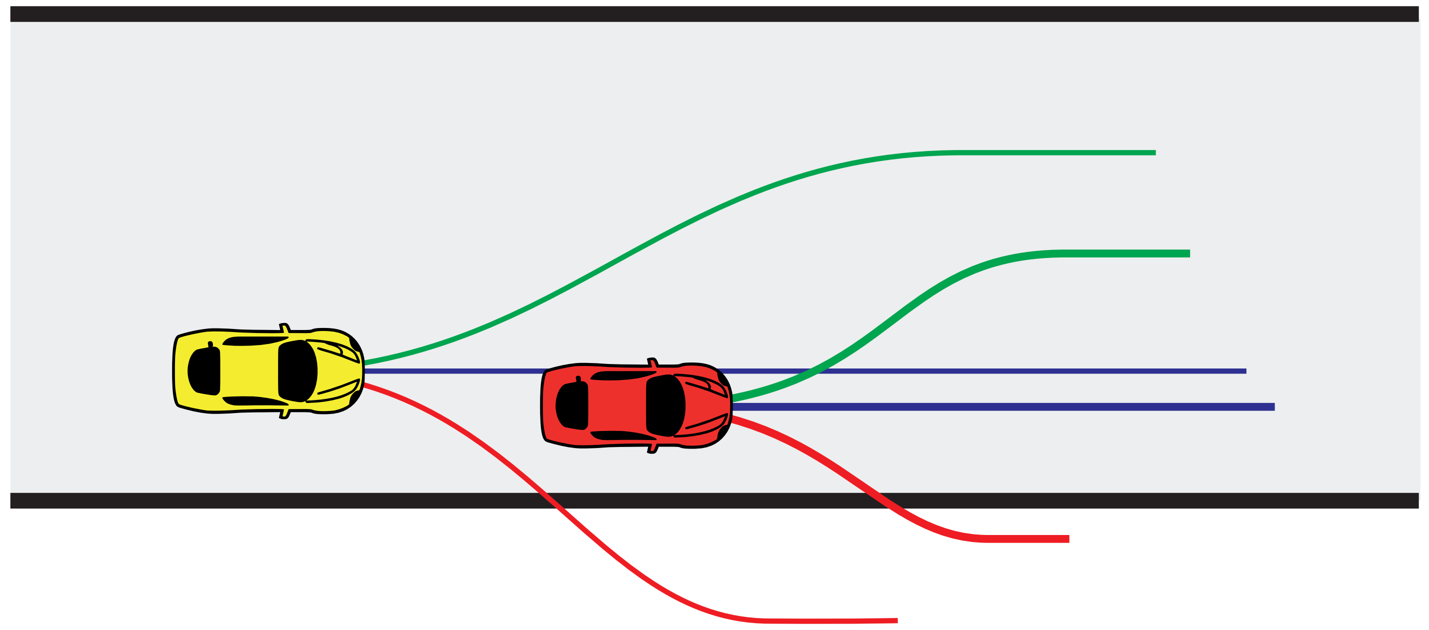
- Every trajectory is an action of a car
  - Each trajectory has a payoff
  - Payoff depends on actions of both cars



Bimatrix Racing Games

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

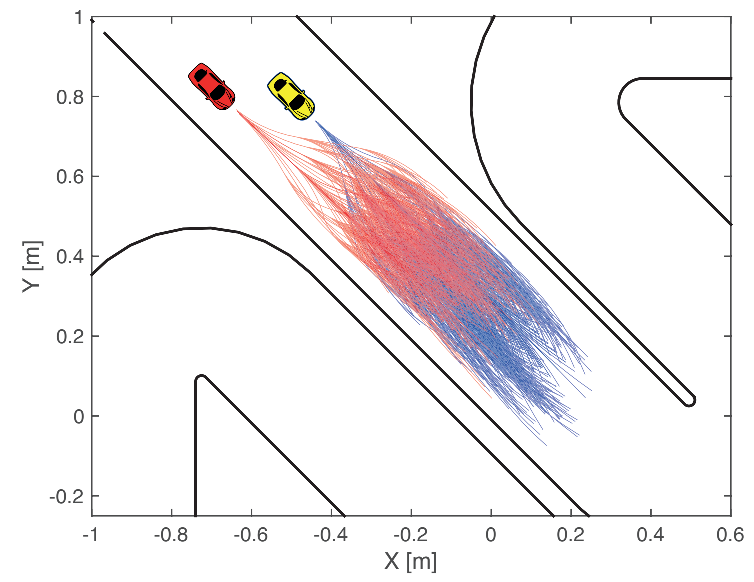


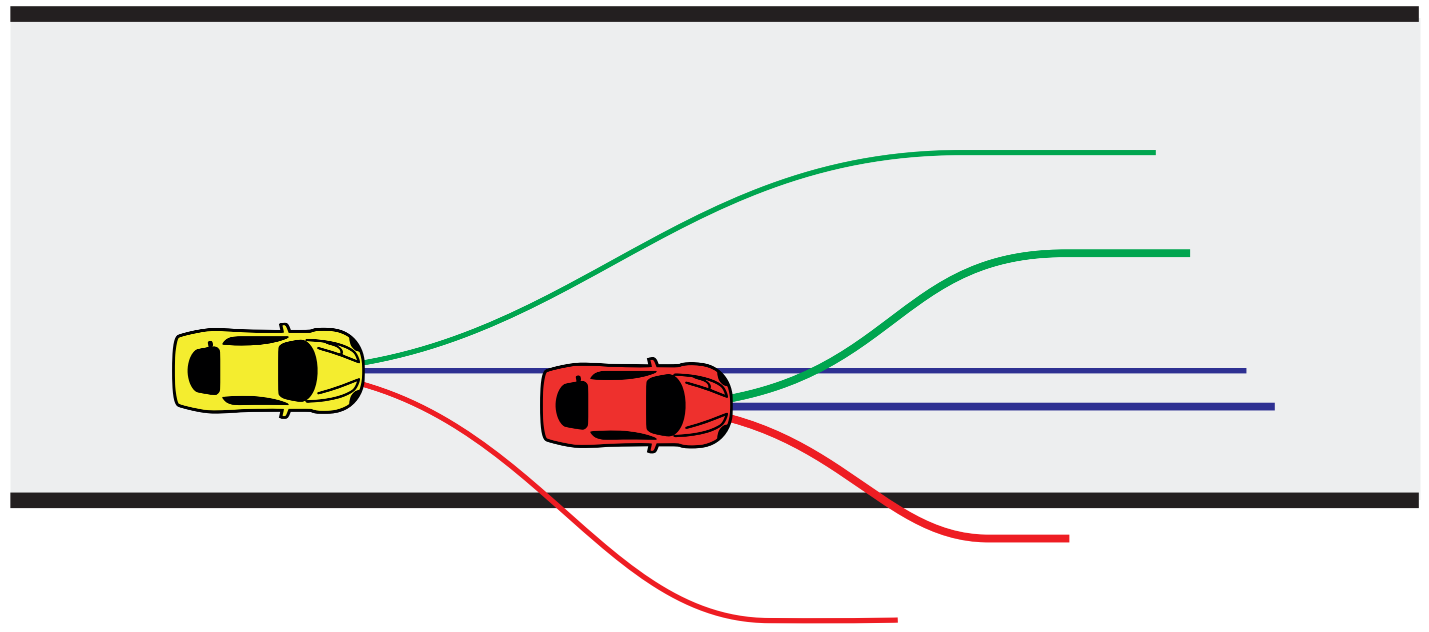


$$A = \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,1} & a_{2,2} & a_{2,3} \\ a_{3,1} & a_{3,2} & a_{3,3} \end{bmatrix} \quad B = \begin{bmatrix} b_{1,1} & b_{1,2} & b_{1,3} \\ b_{2,1} & b_{2,2} & b_{2,3} \\ b_{3,1} & b_{3,2} & b_{3,3} \end{bmatrix}$$

Bimatrix Racing Games

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



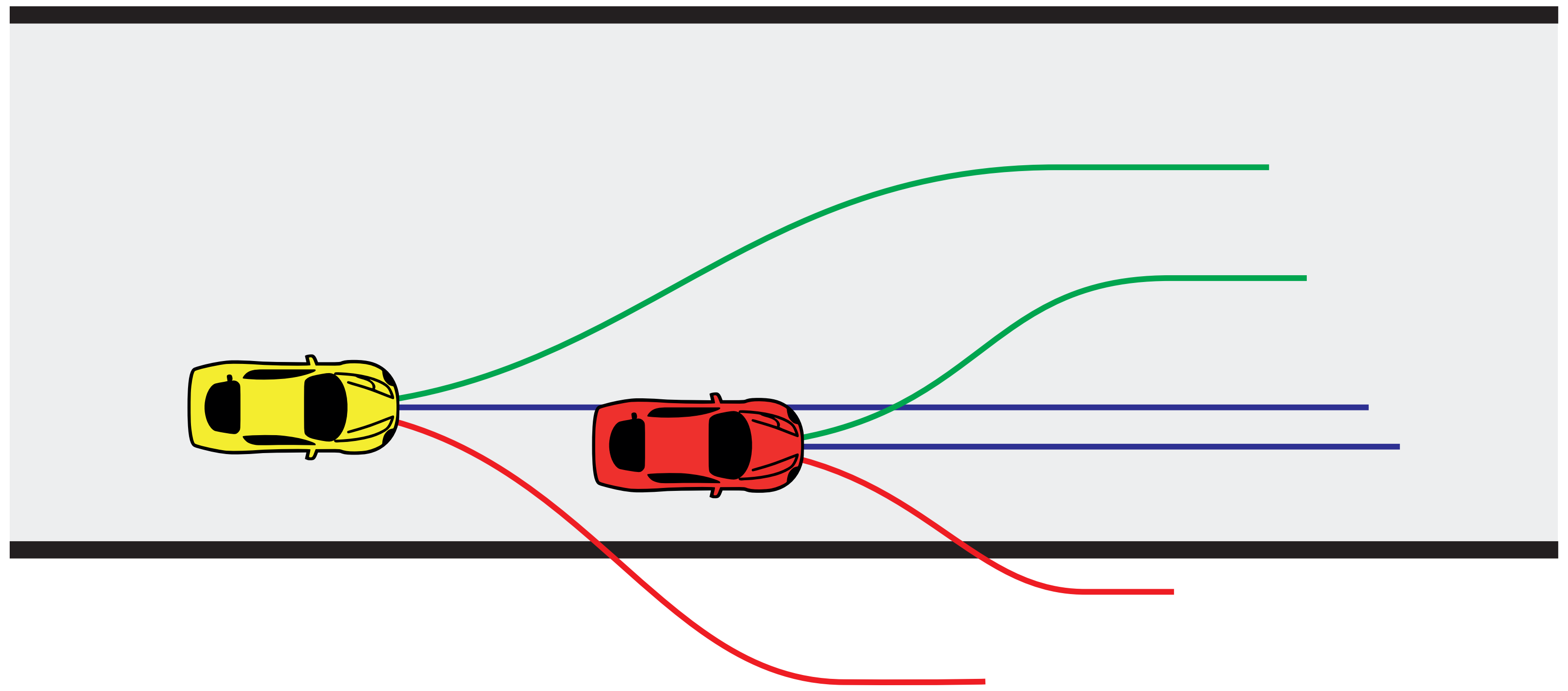


$$A = \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,1} & a_{2,2} & a_{2,3} \\ a_{3,1} & a_{3,2} & a_{3,3} \end{bmatrix} \quad B = \begin{bmatrix} b_{1,1} & b_{1,2} & b_{1,3} \\ b_{2,1} & b_{2,2} & b_{2,3} \\ b_{3,1} & b_{3,2} & b_{3,3} \end{bmatrix}$$

- 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



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

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

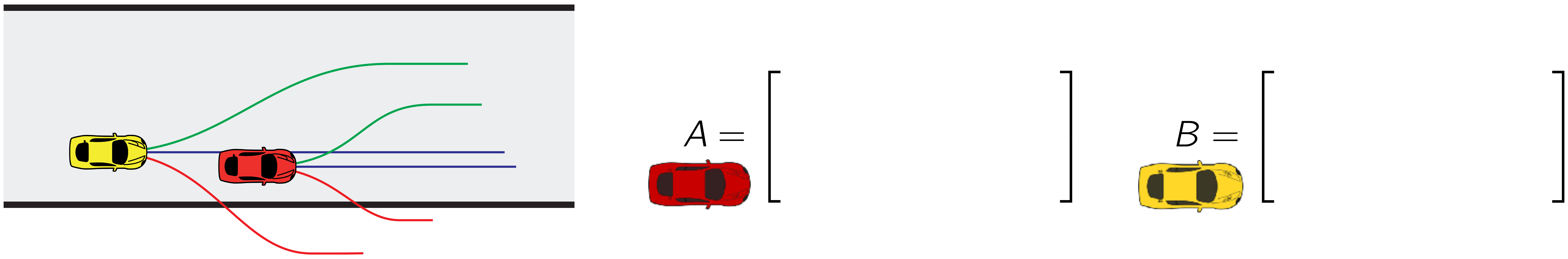
Three Racing Games

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



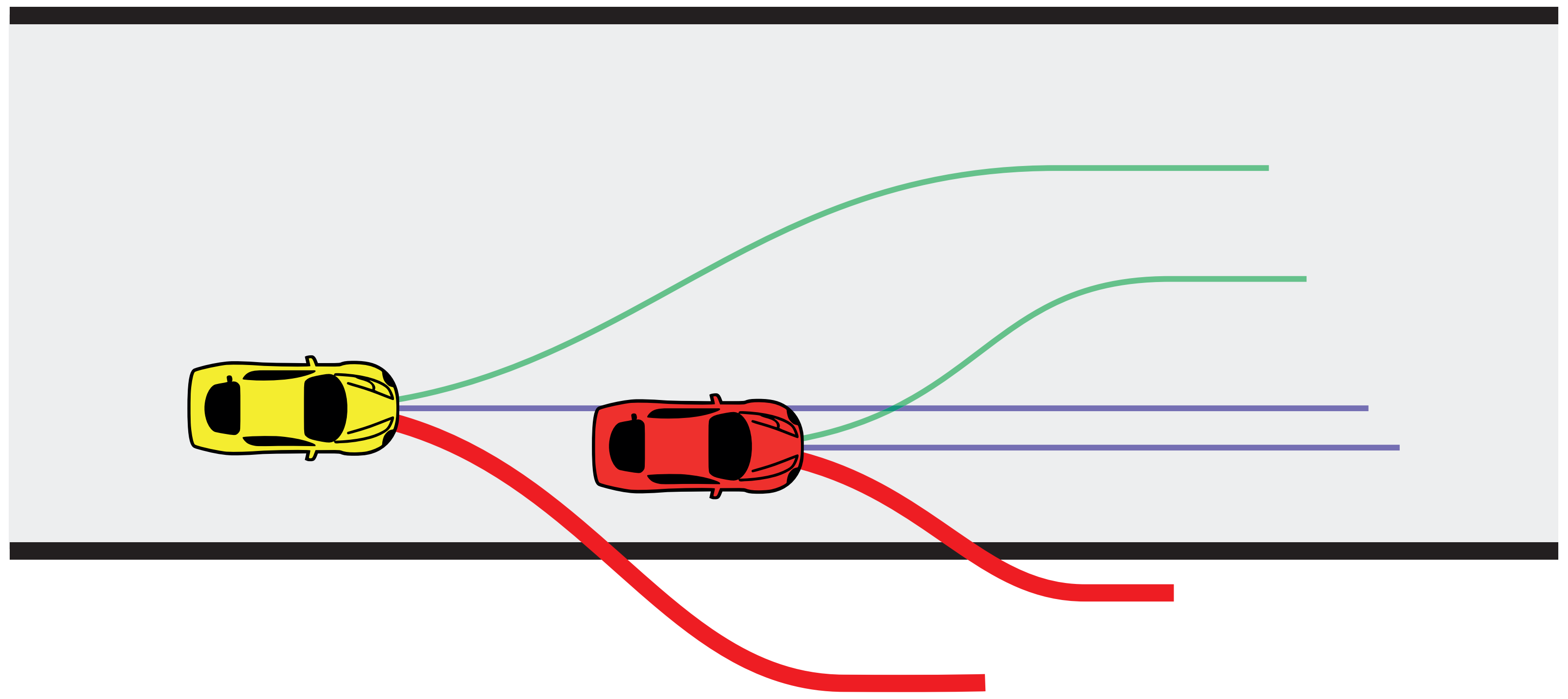
Three Racing Games

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



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

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

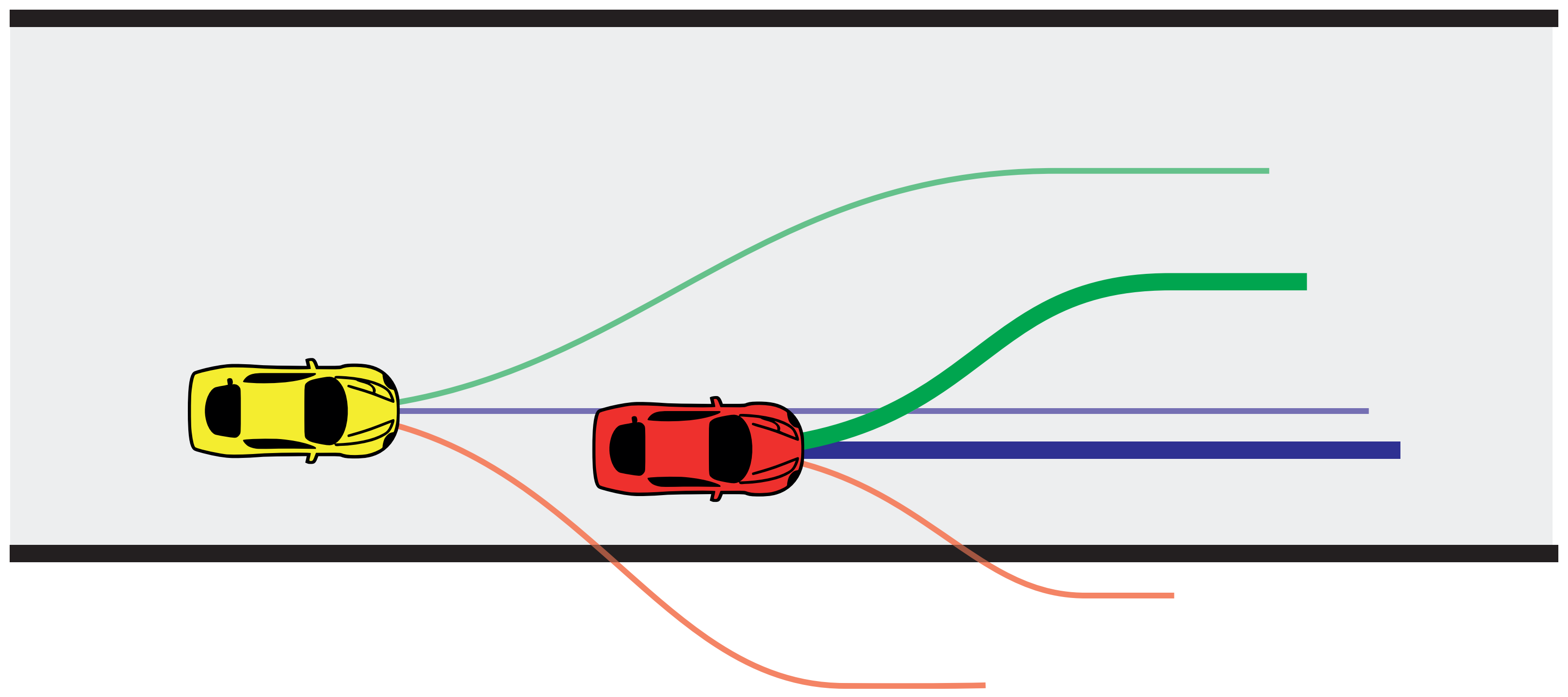
Three Racing Games

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



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

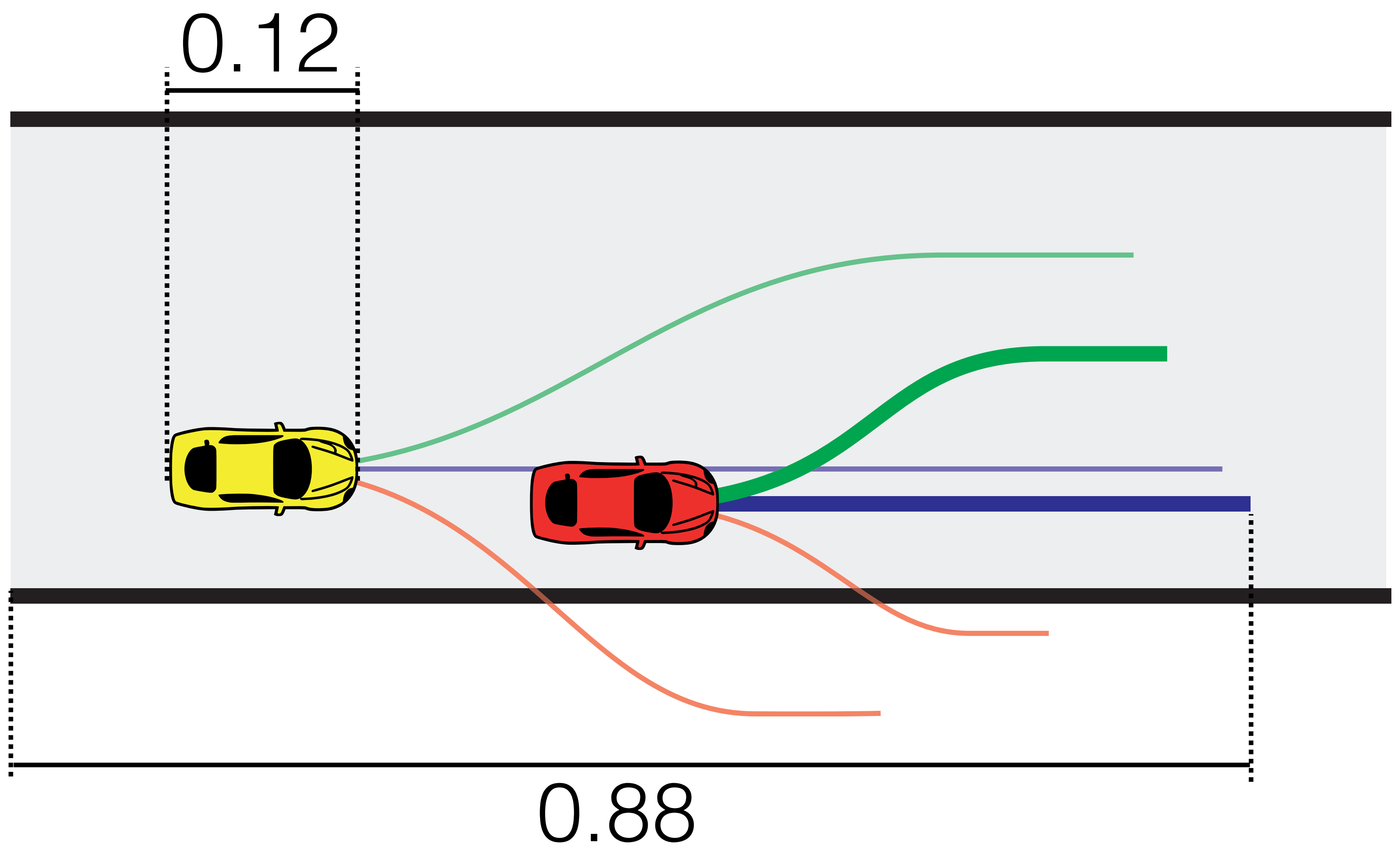
Three Racing Games

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



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

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

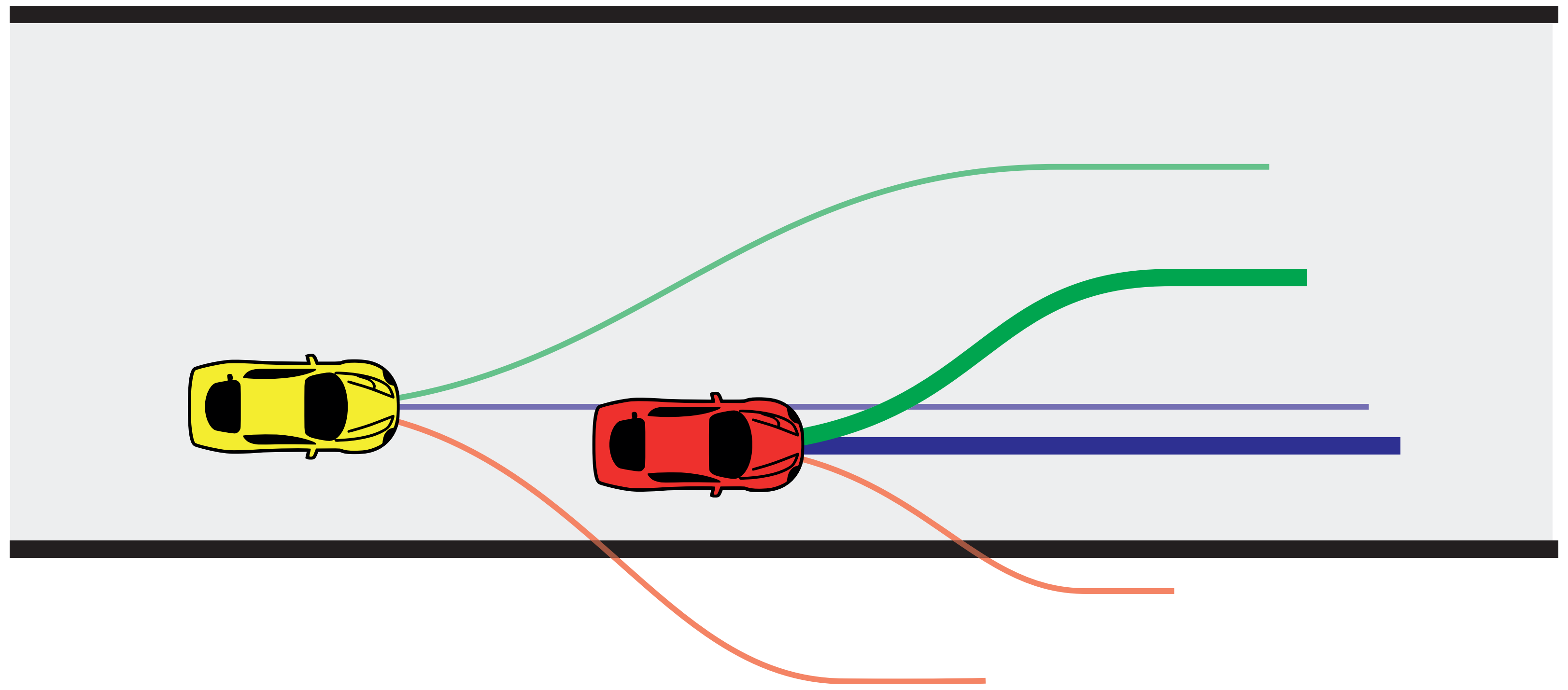
Three Racing Games

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



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

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

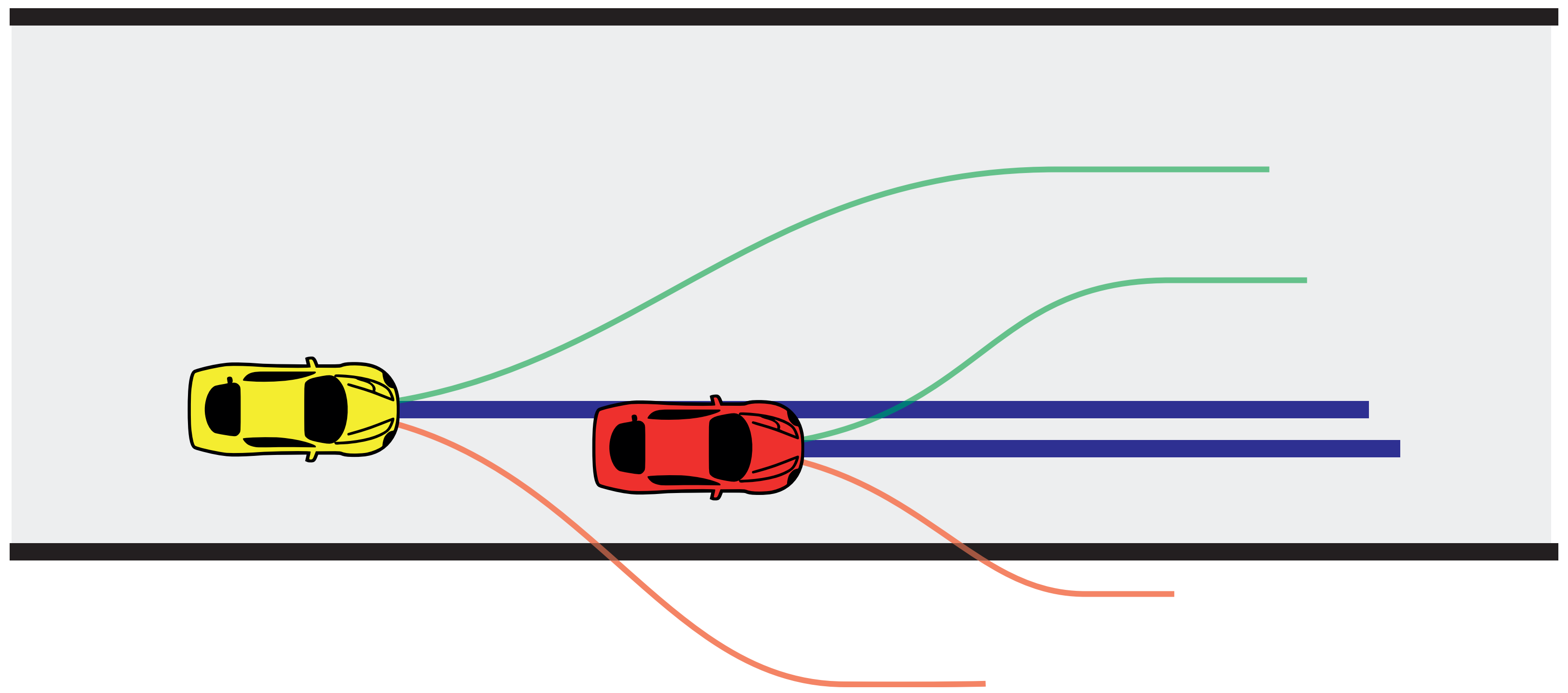
Three Racing Games

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



$$A = \begin{bmatrix} 0.83 & 0.83 & 0.83 \\ 0.88 & 0.88 & 0.88 \\ -10 & -10 & -10 \end{bmatrix} \qquad B = \begin{bmatrix} 0.81 & 0.86 & -10 \\ 0.81 & \boxed{-1} & -10 \\ 0.81 & 0.86 & -10 \end{bmatrix}$$

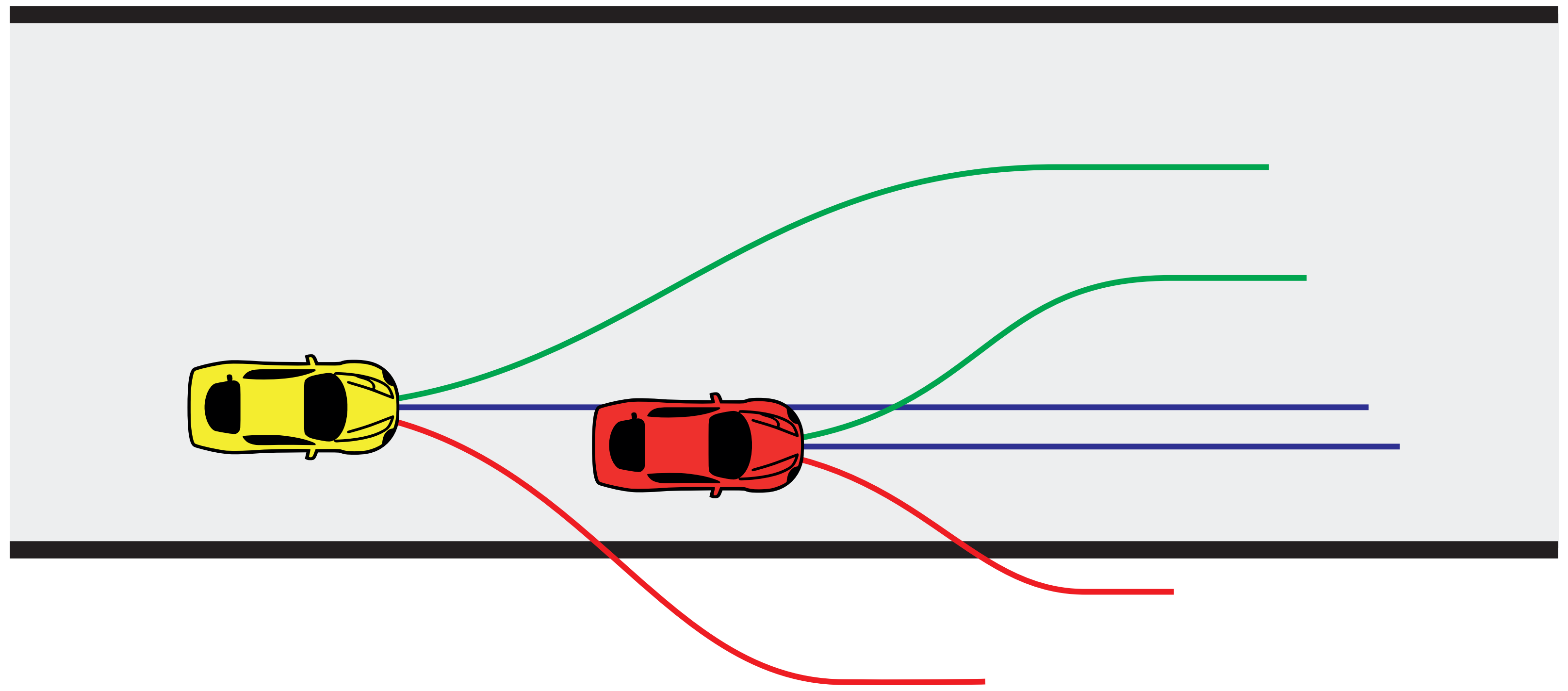
Three Racing Games

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



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

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

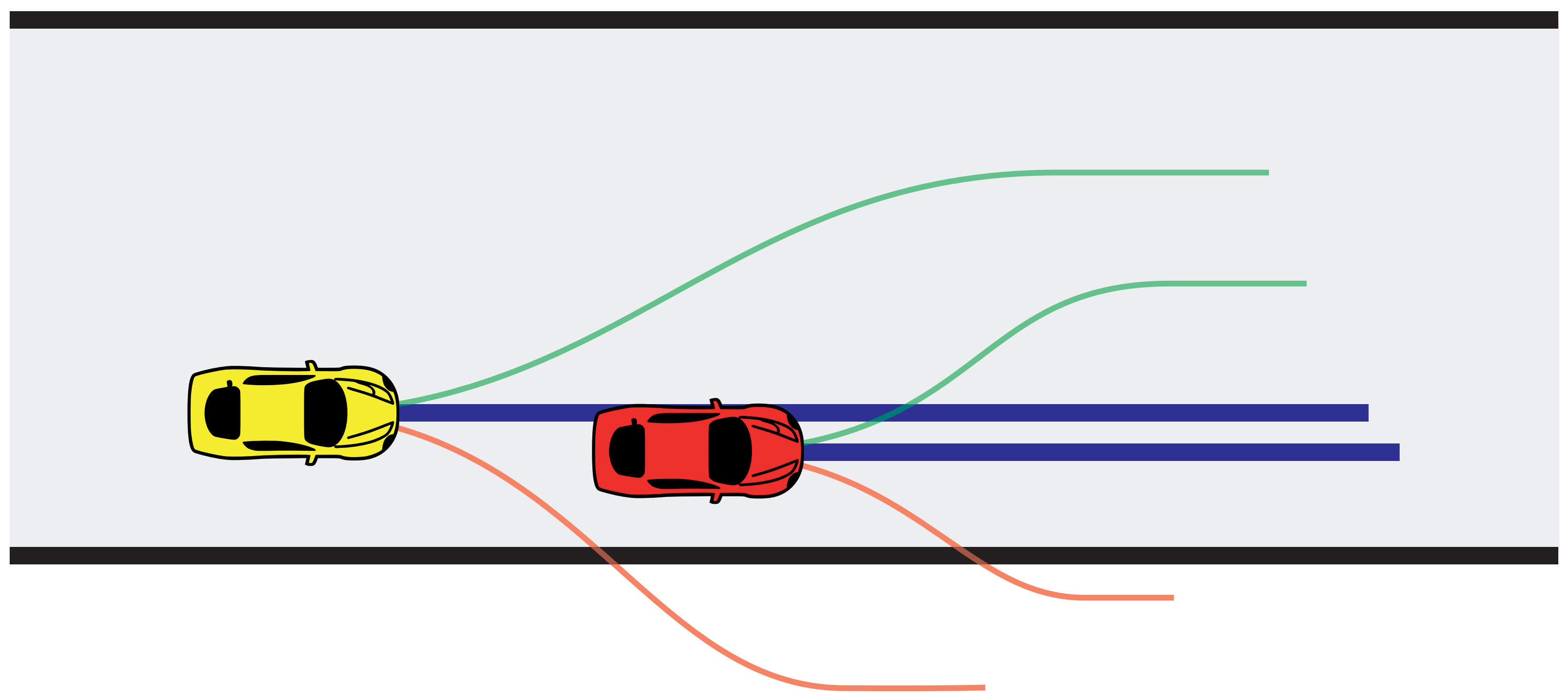
Three Racing Games

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



$$A = \begin{bmatrix} 0.83 & 0.83 & 0.83 \\ 0.88 & \boxed{-1} & 0.88 \\ -10 & -10 & -10 \end{bmatrix} \qquad B = \begin{bmatrix} 0.81 & 0.86 & -10 \\ 0.81 & -1 & -10 \\ 0.81 & 0.86 & -10 \end{bmatrix}$$

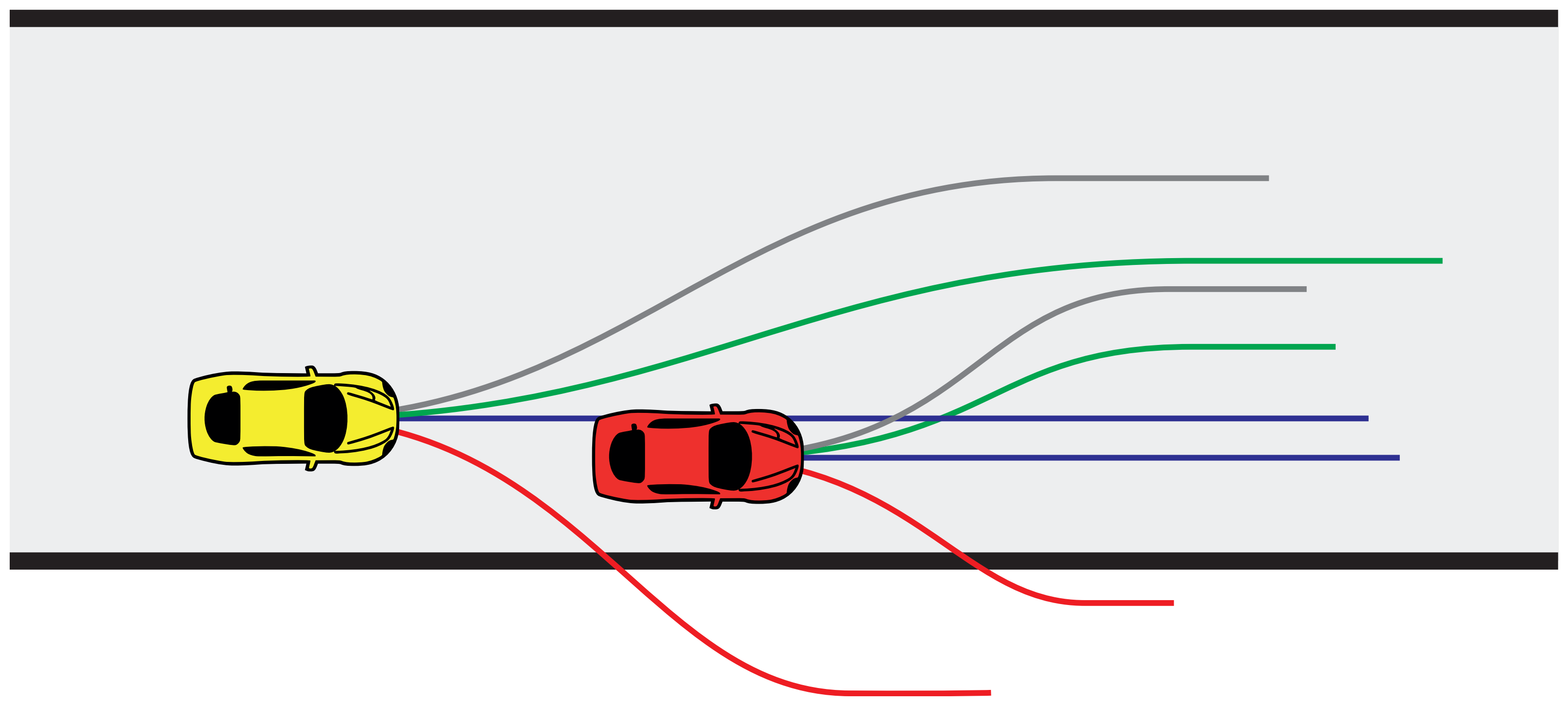
Three Racing Games

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



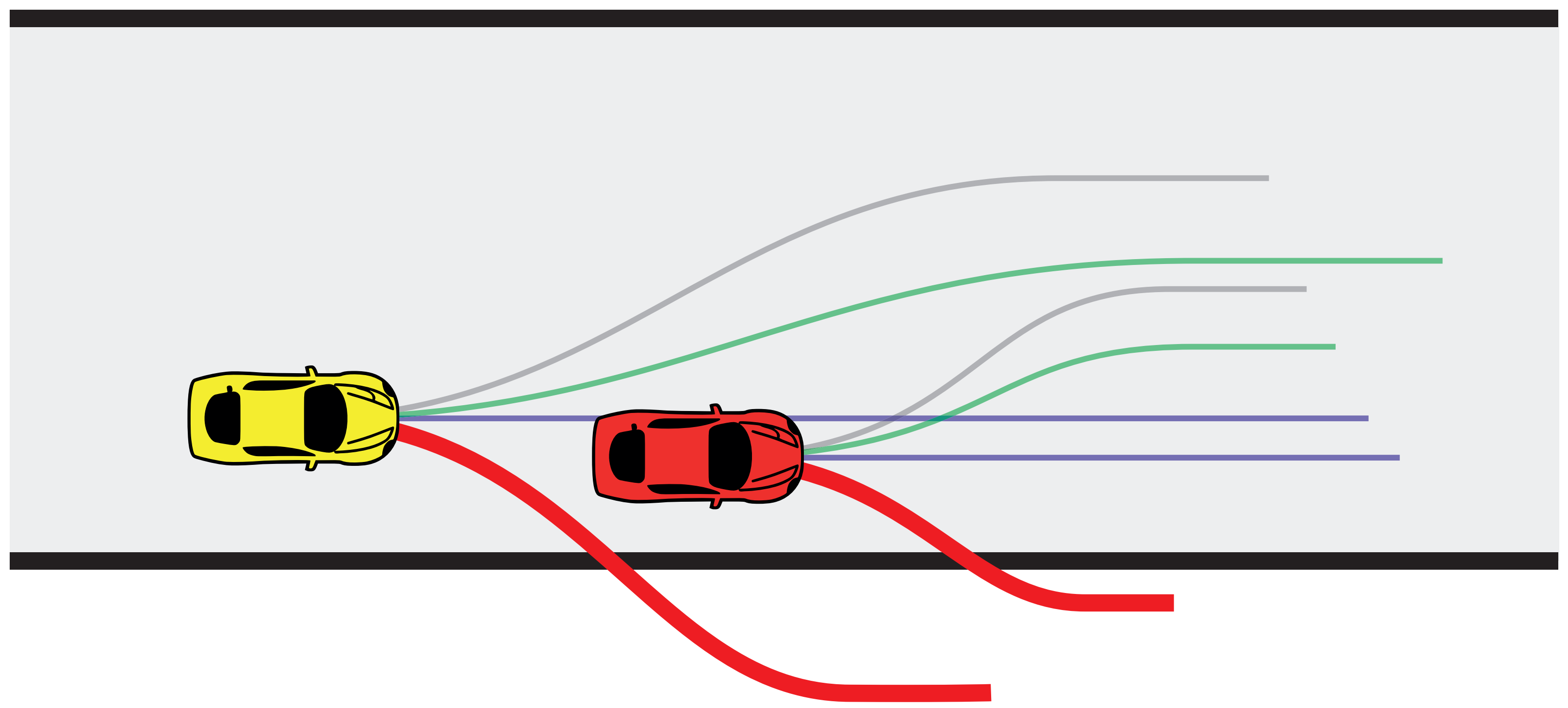
Three Racing Games

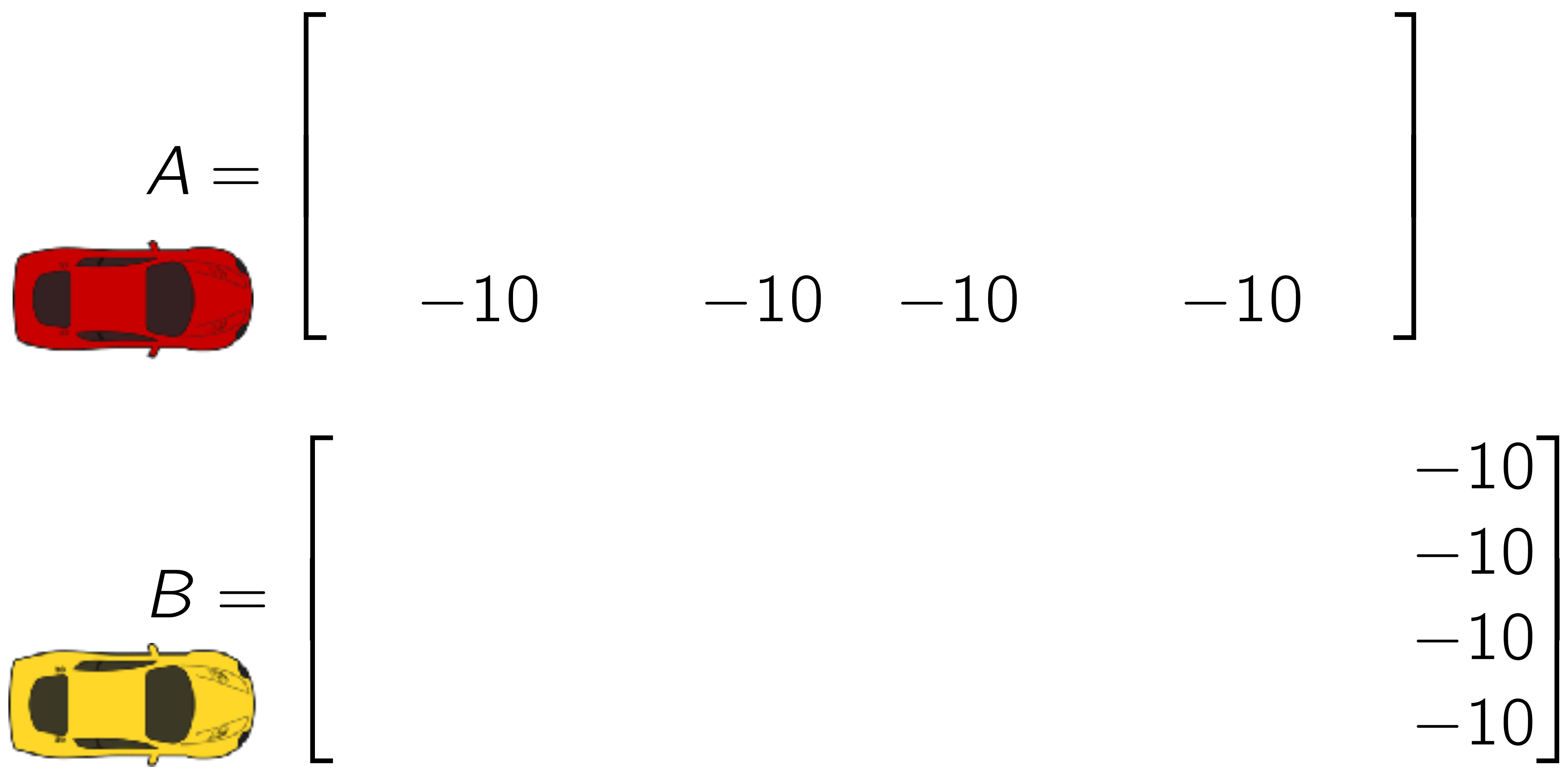
Sequential Game

Cooperative Game

Blocking Game

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





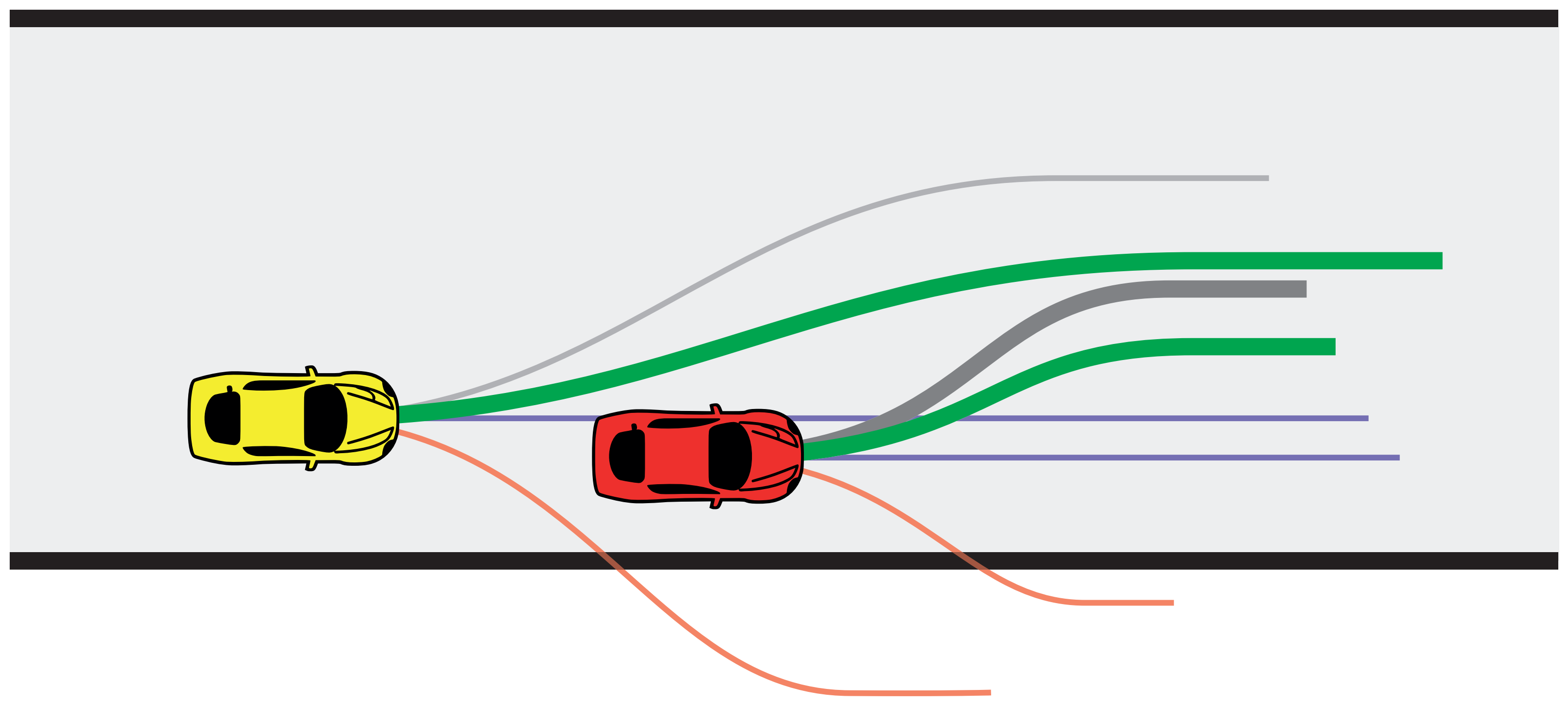
Three Racing Games

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



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

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

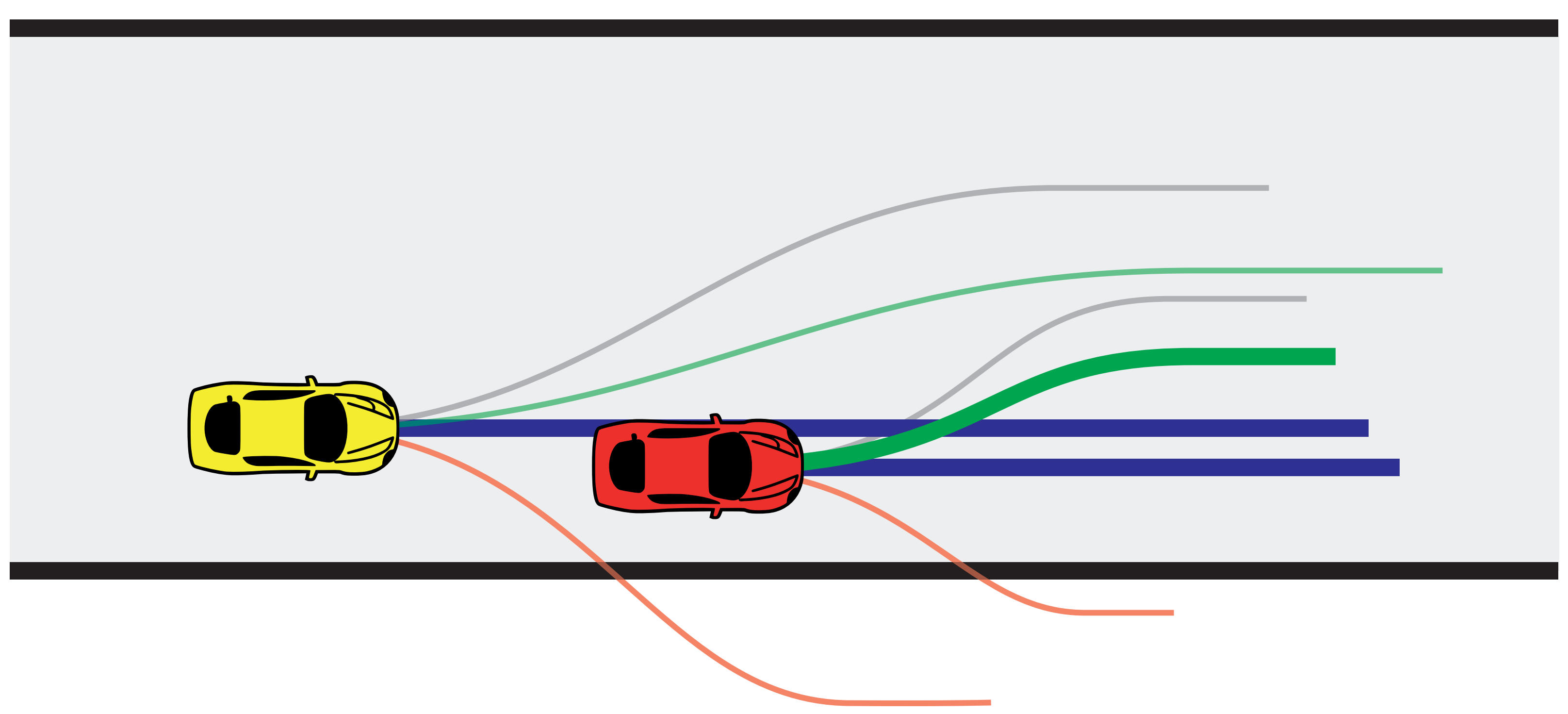
Three Racing Games

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



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

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

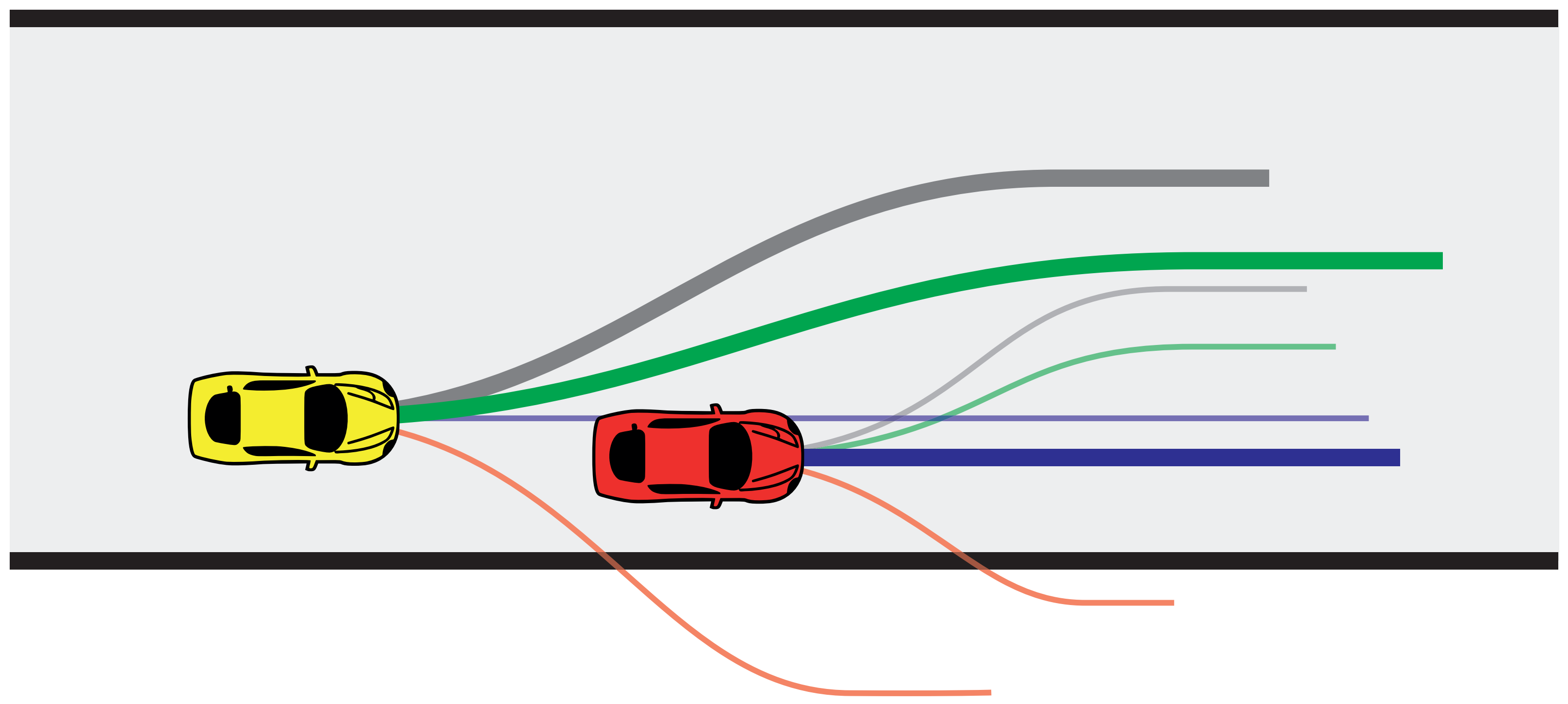
Three Racing Games

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



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

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

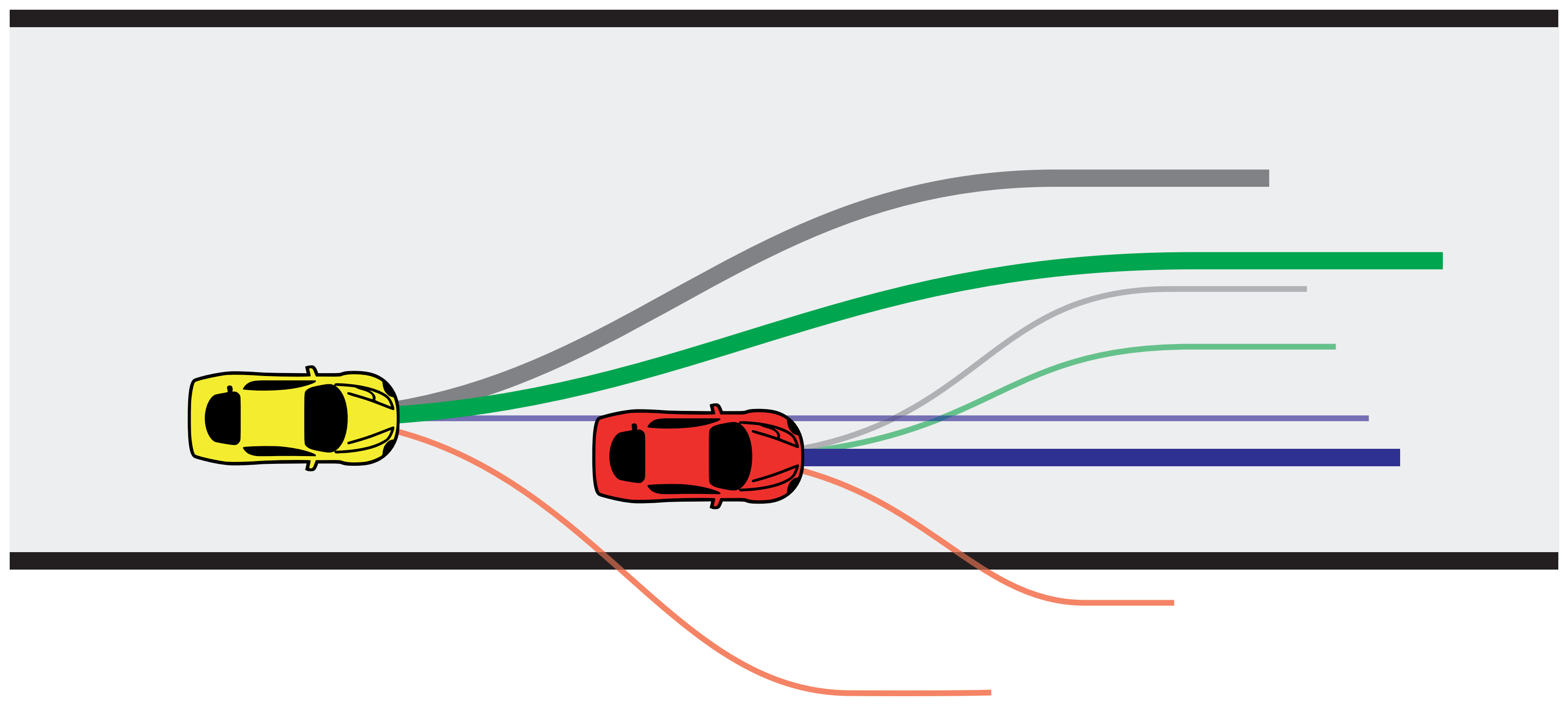
Three Racing Games

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



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

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

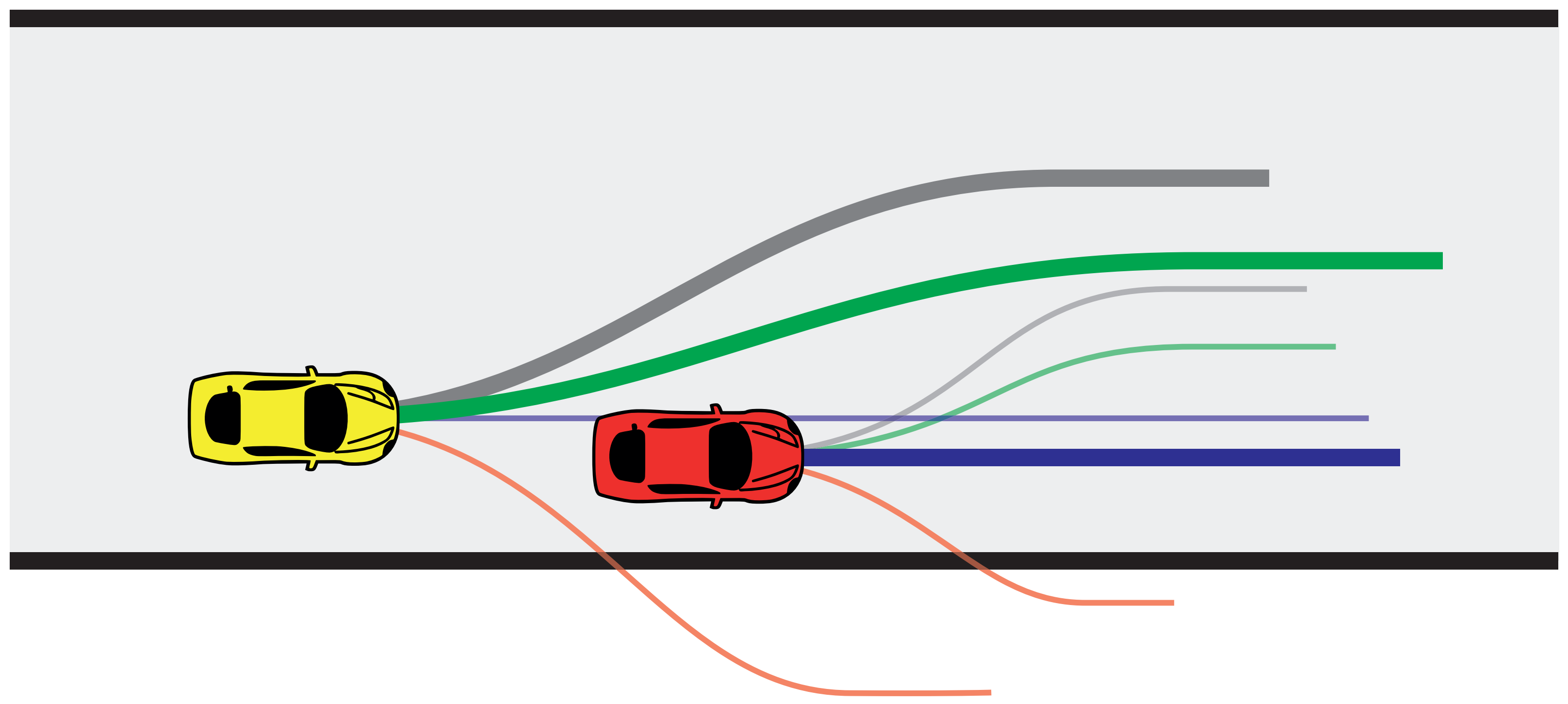
Three Racing Games

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



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

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

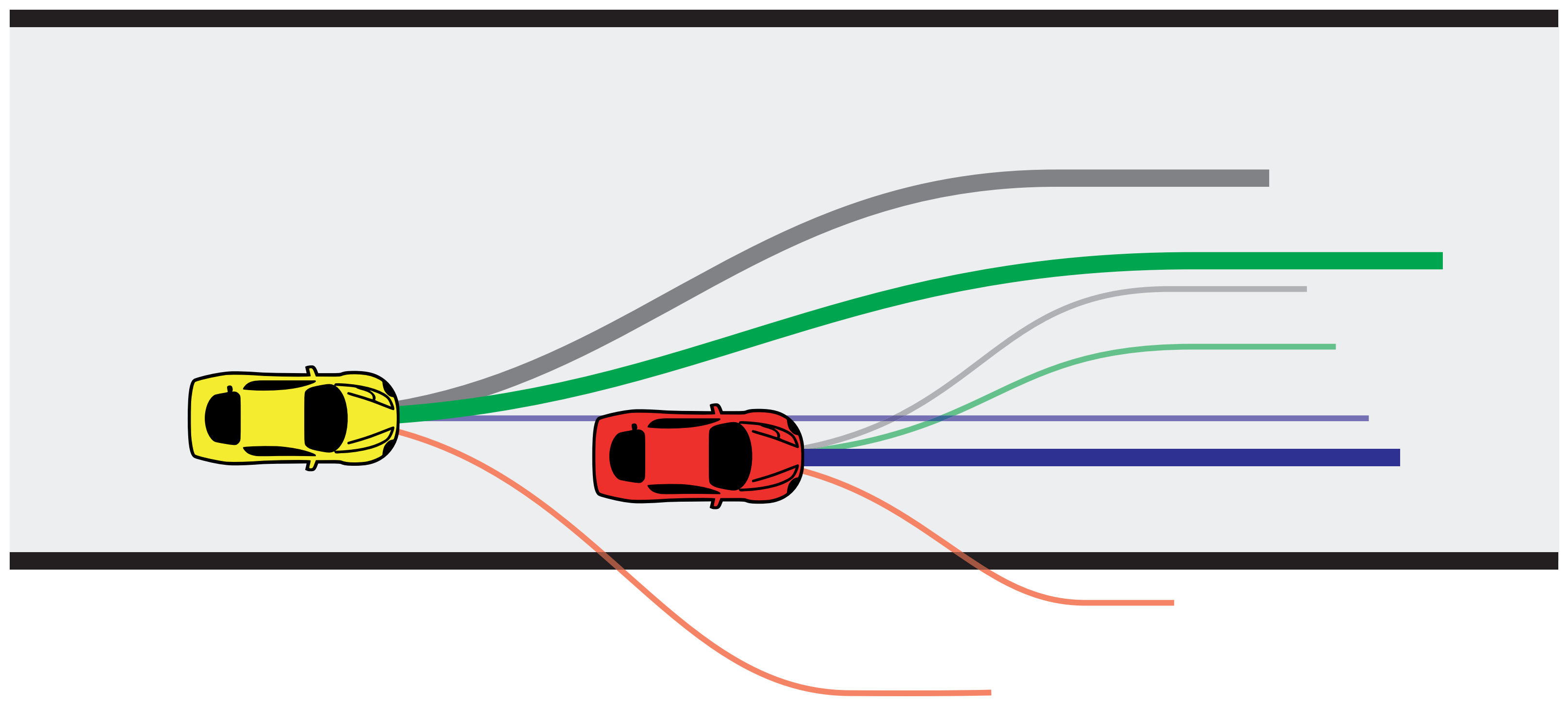
Three Racing Games

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

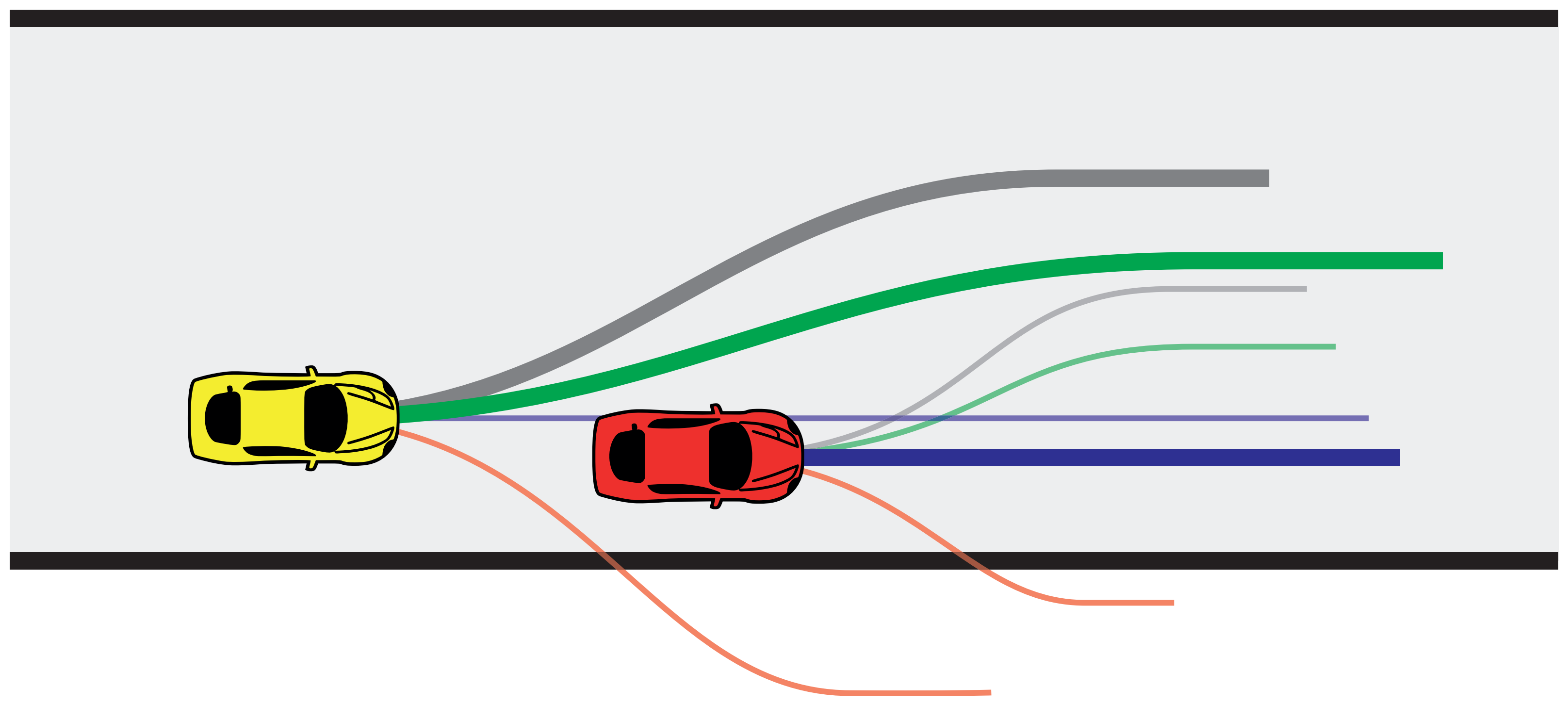
Sequential Game

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

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

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

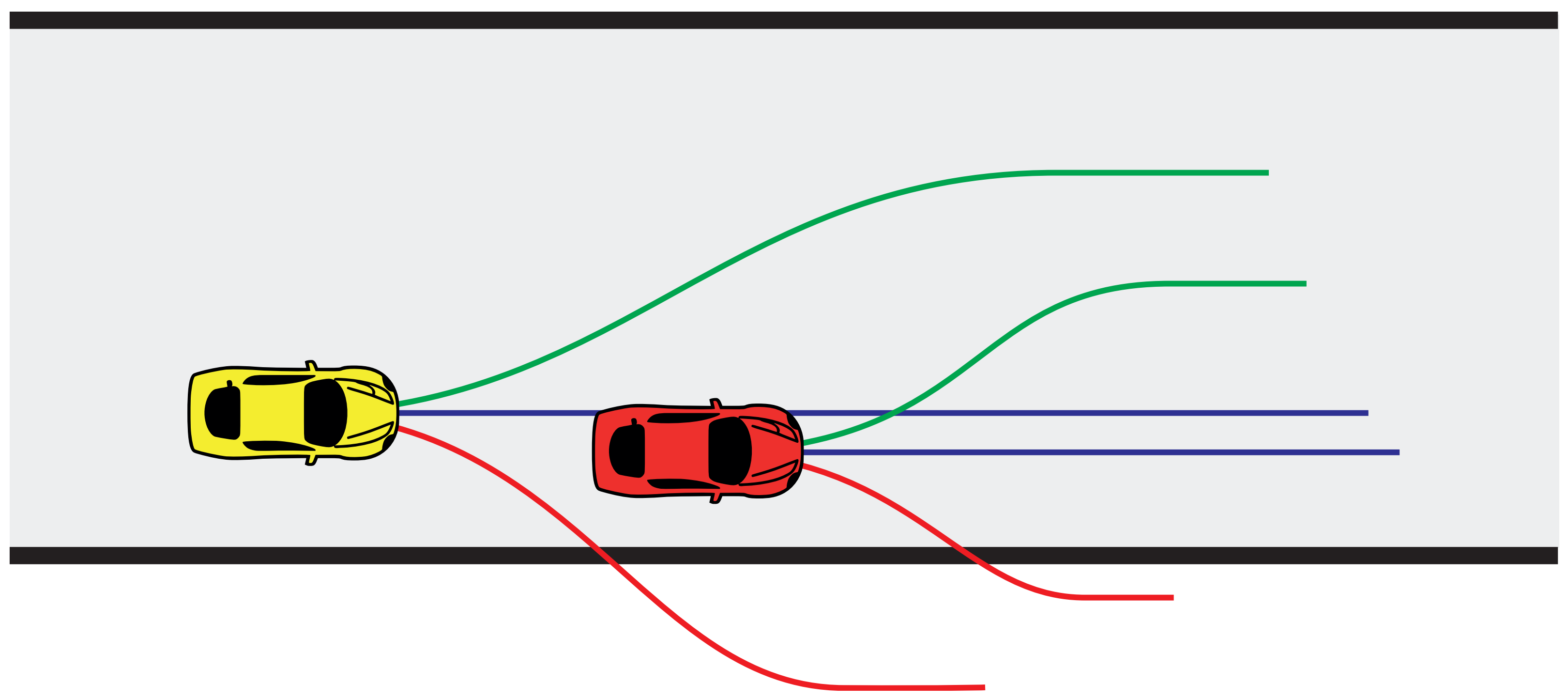
Nash Equilibria

- 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

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

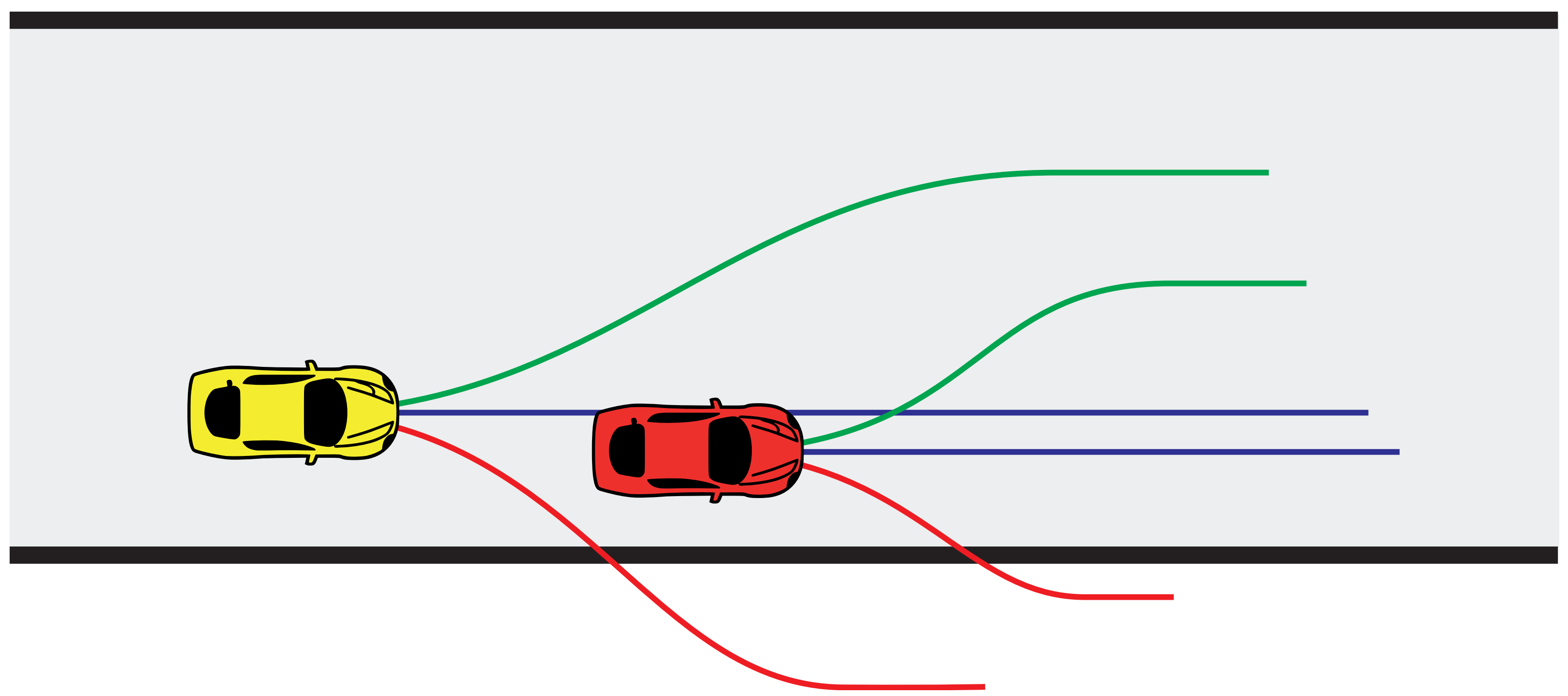
$$B = \begin{bmatrix} 0.81 & 0.86 & -10 \\ 0.81 & -1 & -10 \\ 0.81 & 0.86 & -10 \end{bmatrix}$$

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

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

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

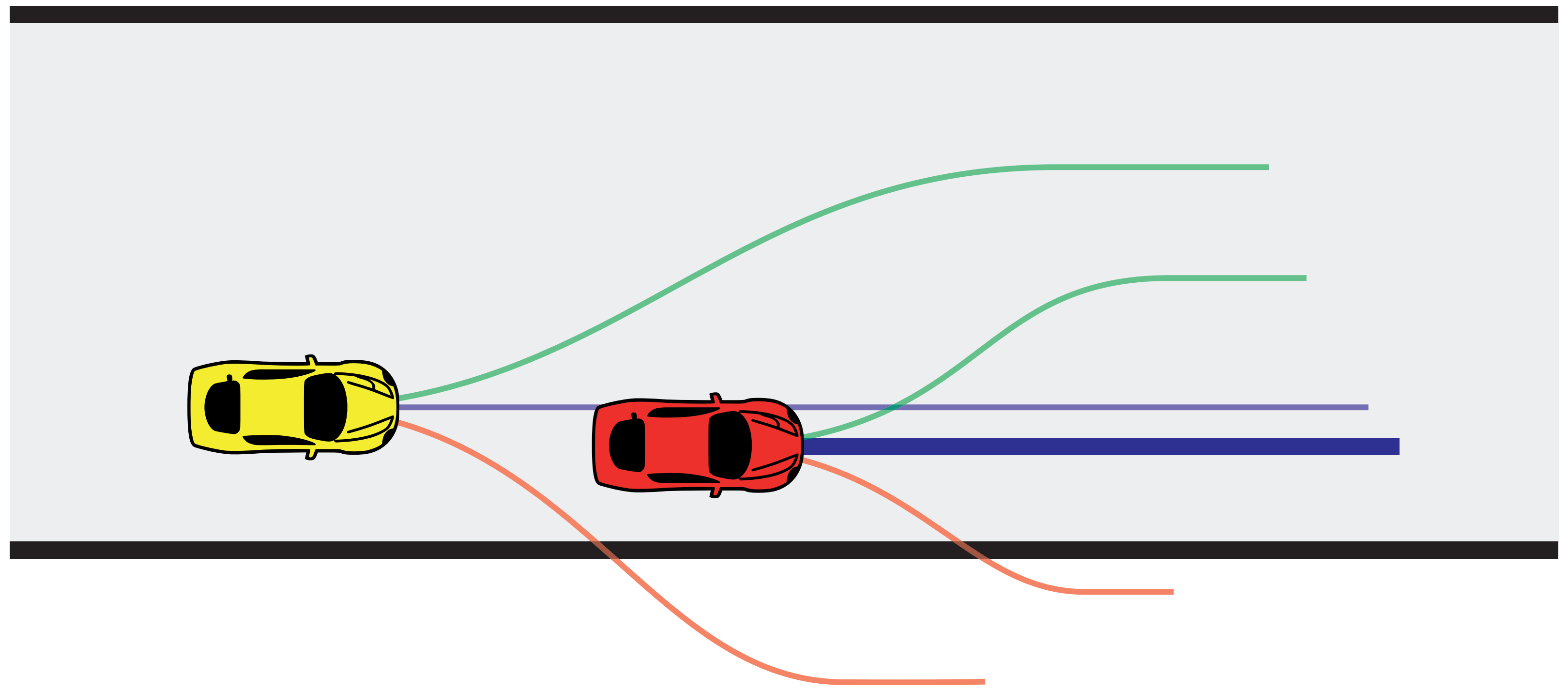
cooperative game

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

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

- The sequential game can be solved by sequential maximizing

Sequential and Cooperative Game



sequential game

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

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

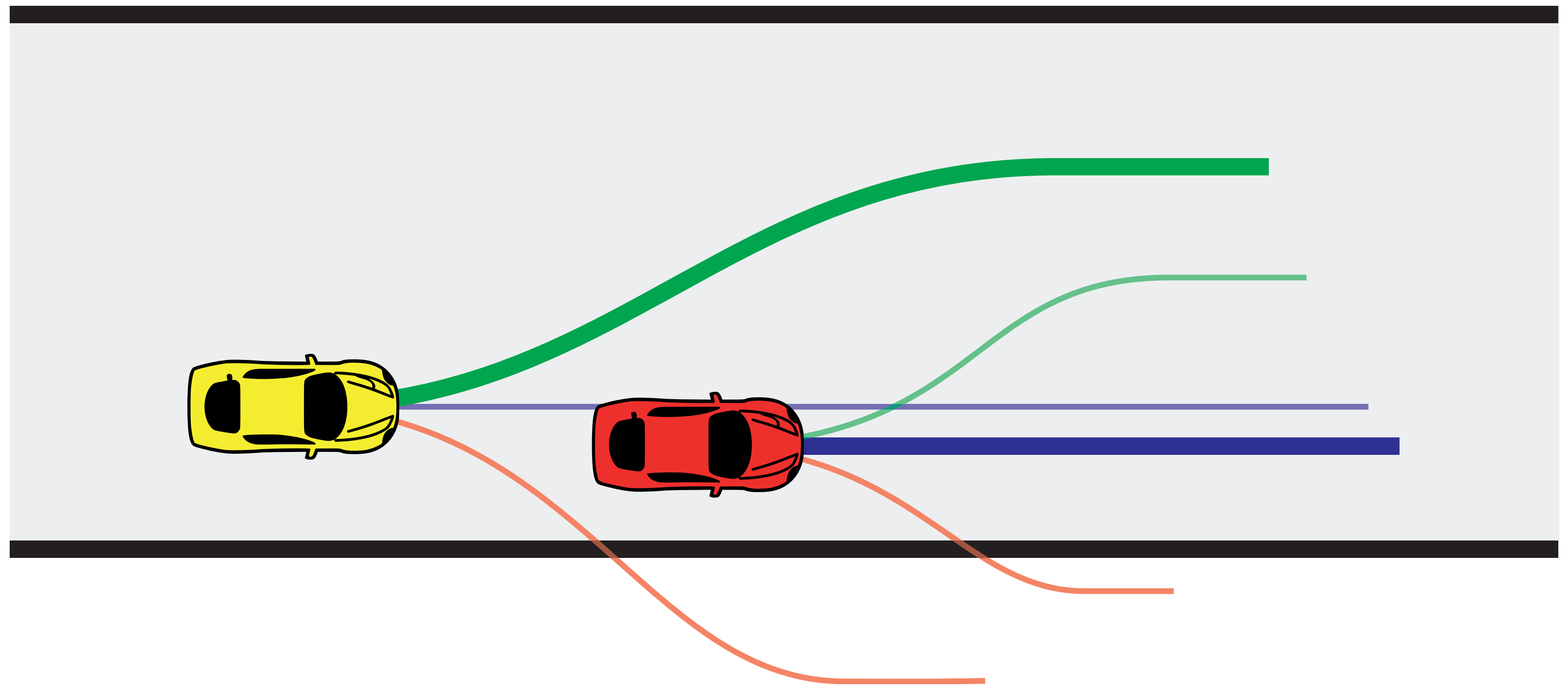
cooperative game

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

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

- The sequential game can be solved by sequential maximizing

Sequential and Cooperative Game



sequential game

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

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

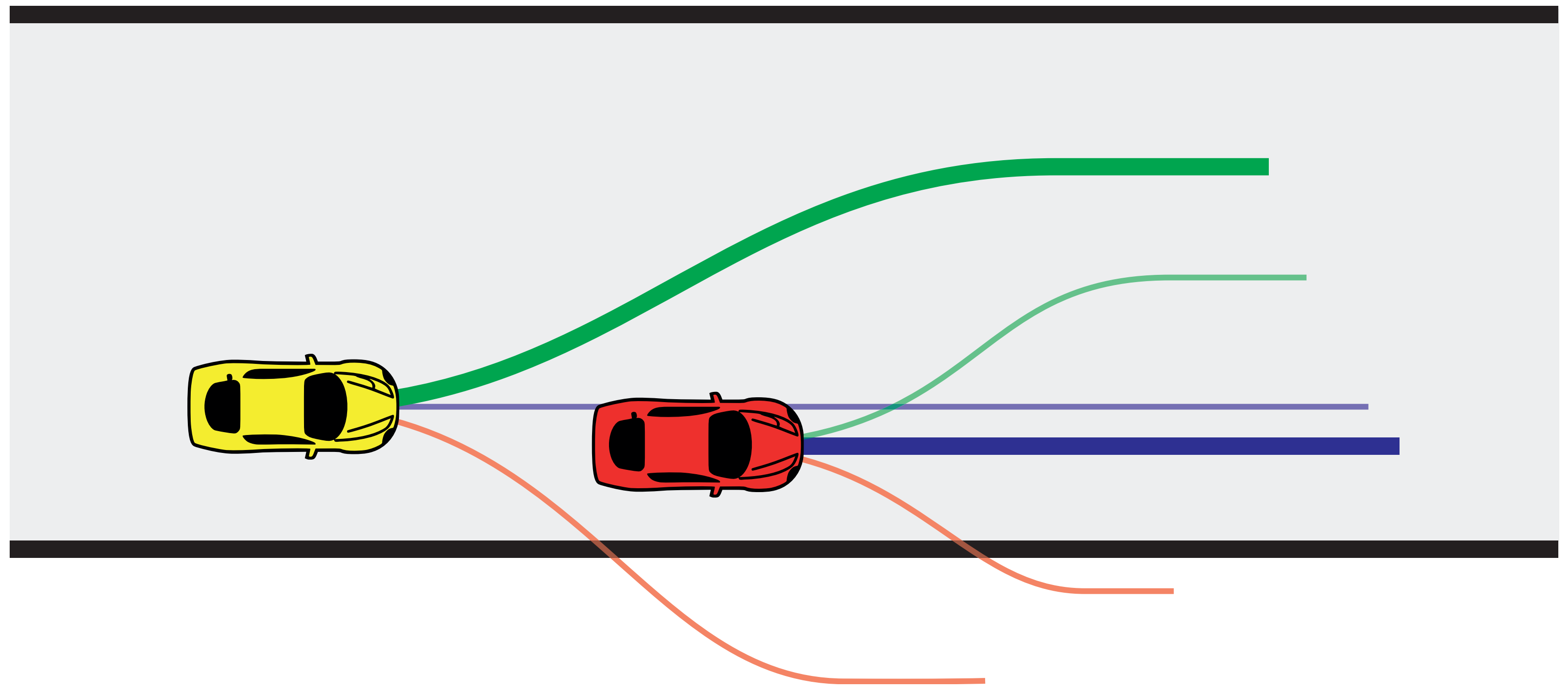
cooperative game

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

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

- The sequential game can be solved by sequential maximizing

Sequential and Cooperative Game



sequential game

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

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

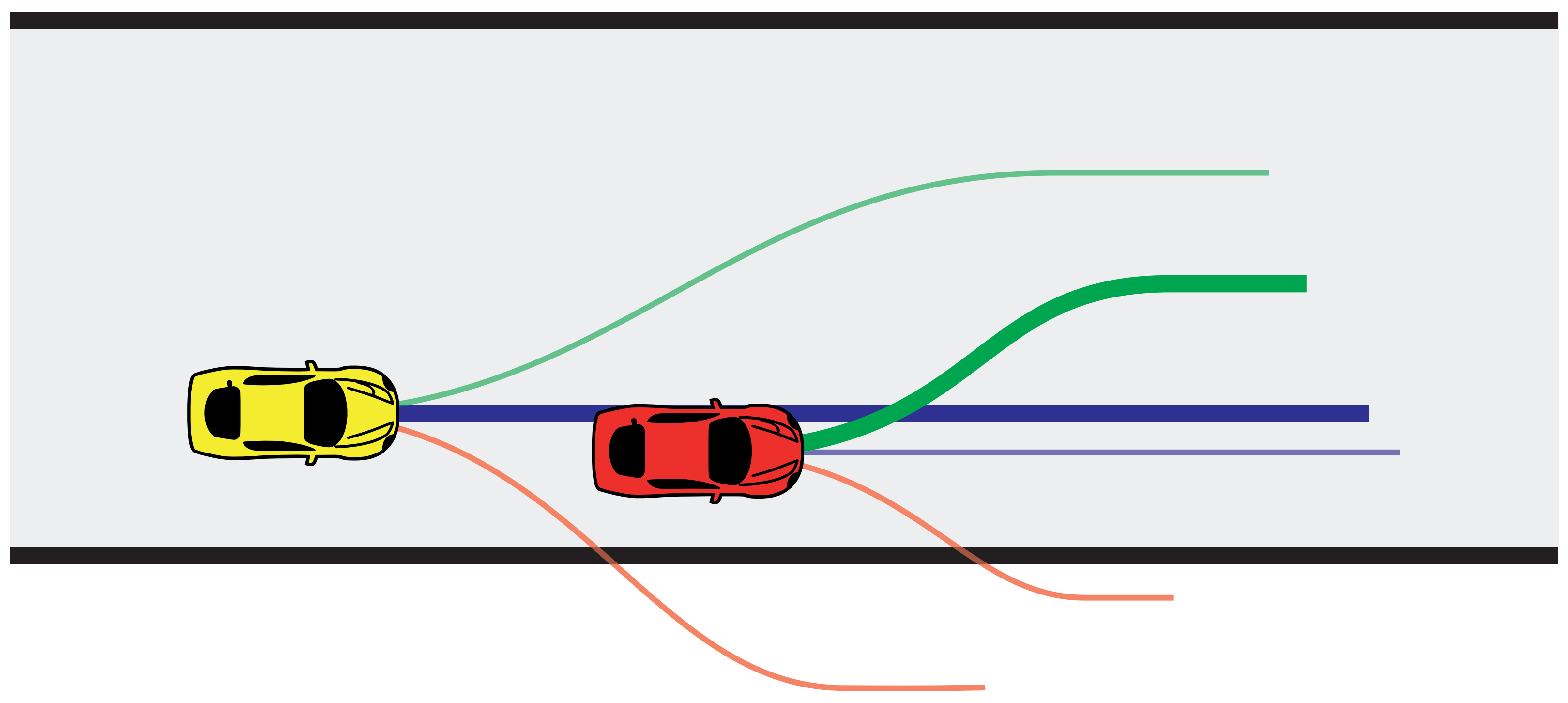
cooperative game

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

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

- The sequential game can be solved by sequential maximizing

Sequential and Cooperative Game



sequential game

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

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

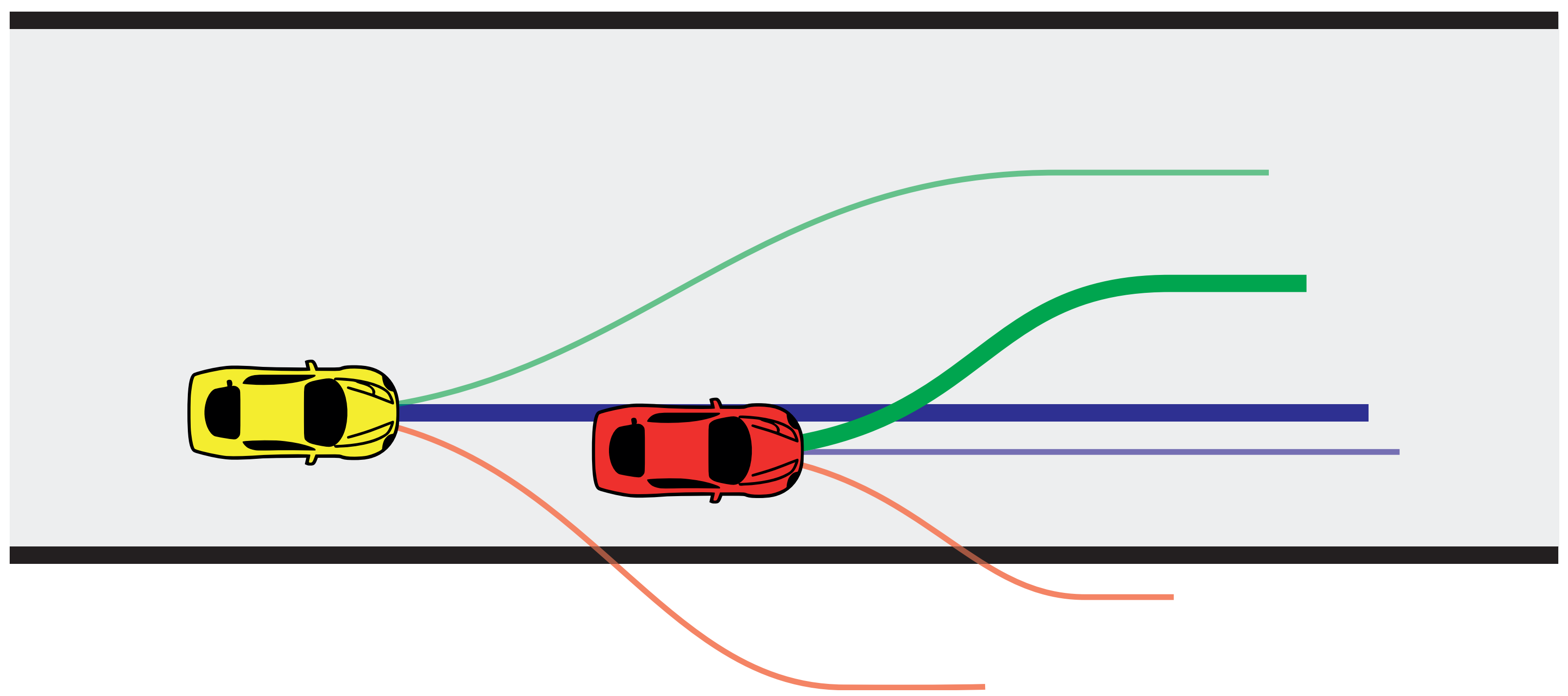
cooperative game

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

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

- The sequential game can be solved by sequential maximizing

Sequential and Cooperative Game



sequential game

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

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

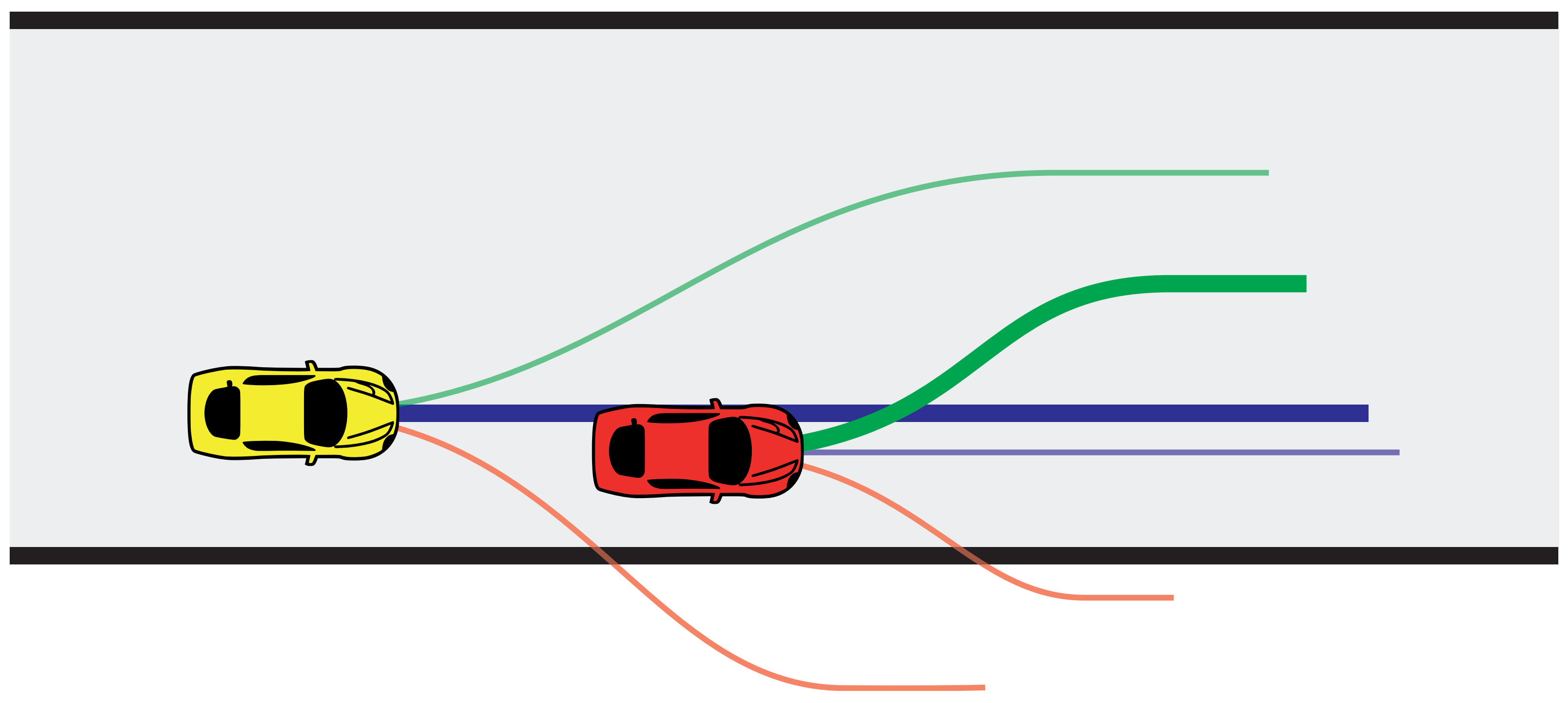
cooperative game

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

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

- 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 a Nash equilibrium

Sequential and Cooperative Game



sequential game

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

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

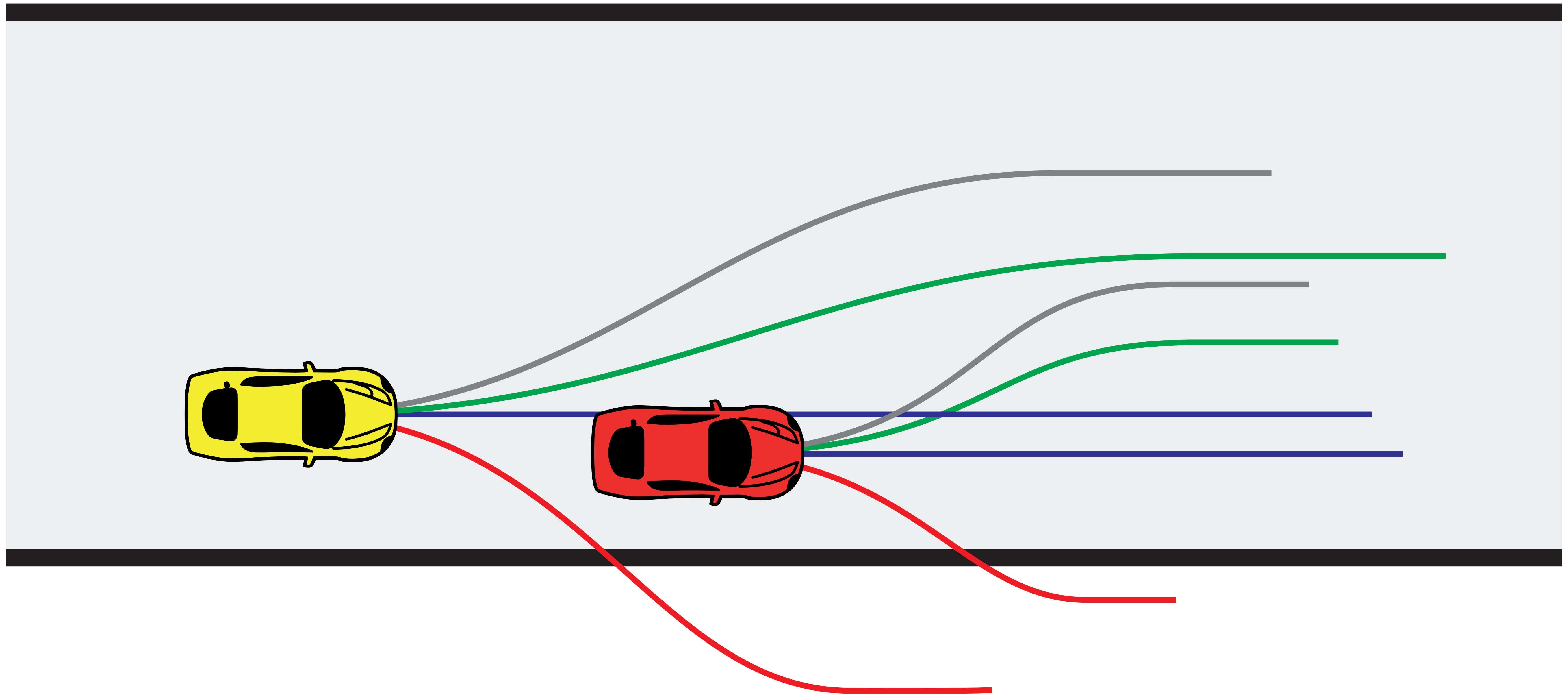
cooperative game

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

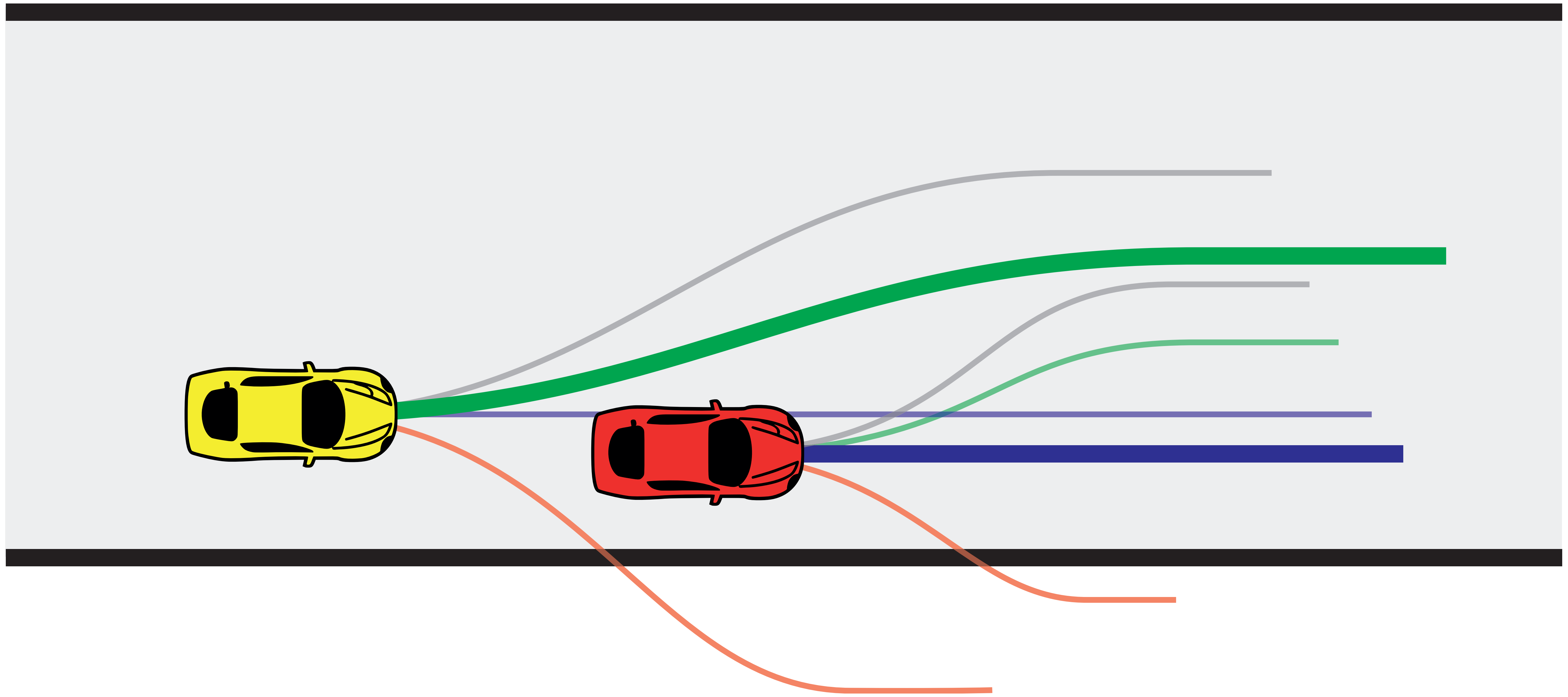
$$B = \begin{bmatrix} 0.81 & 0.86 & -10 \\ 0.81 & -1 & -10 \\ 0.81 & 0.86 & -10 \end{bmatrix}$$

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

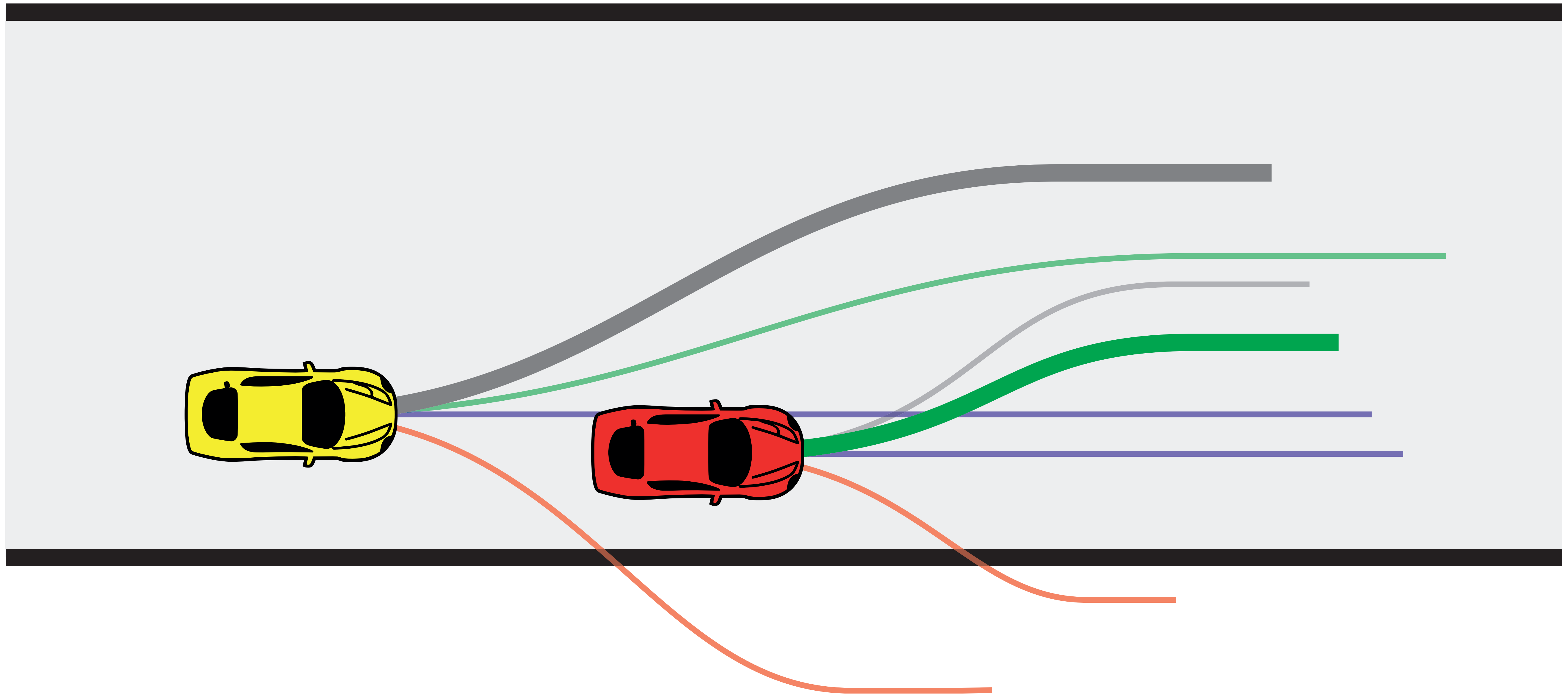
Blocking Trajectories



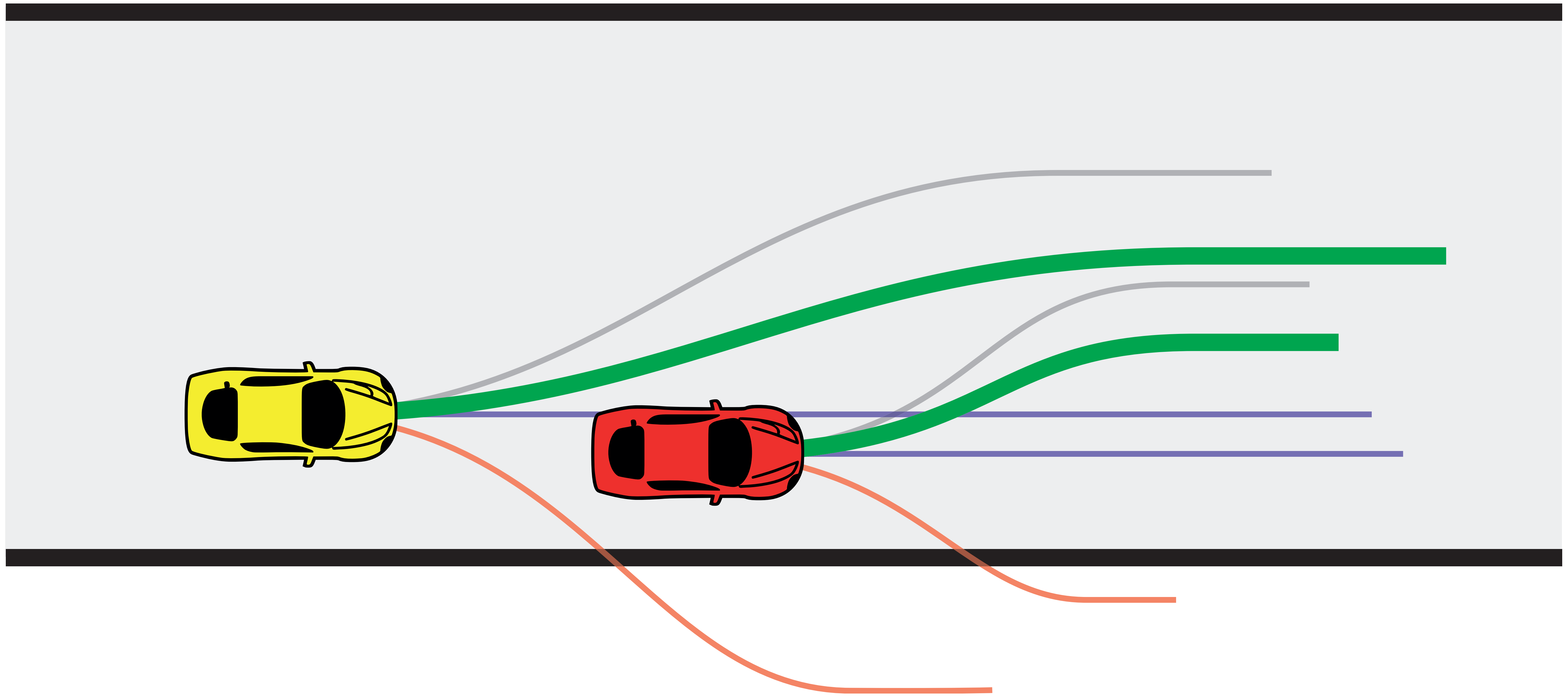
Blocking Trajectories



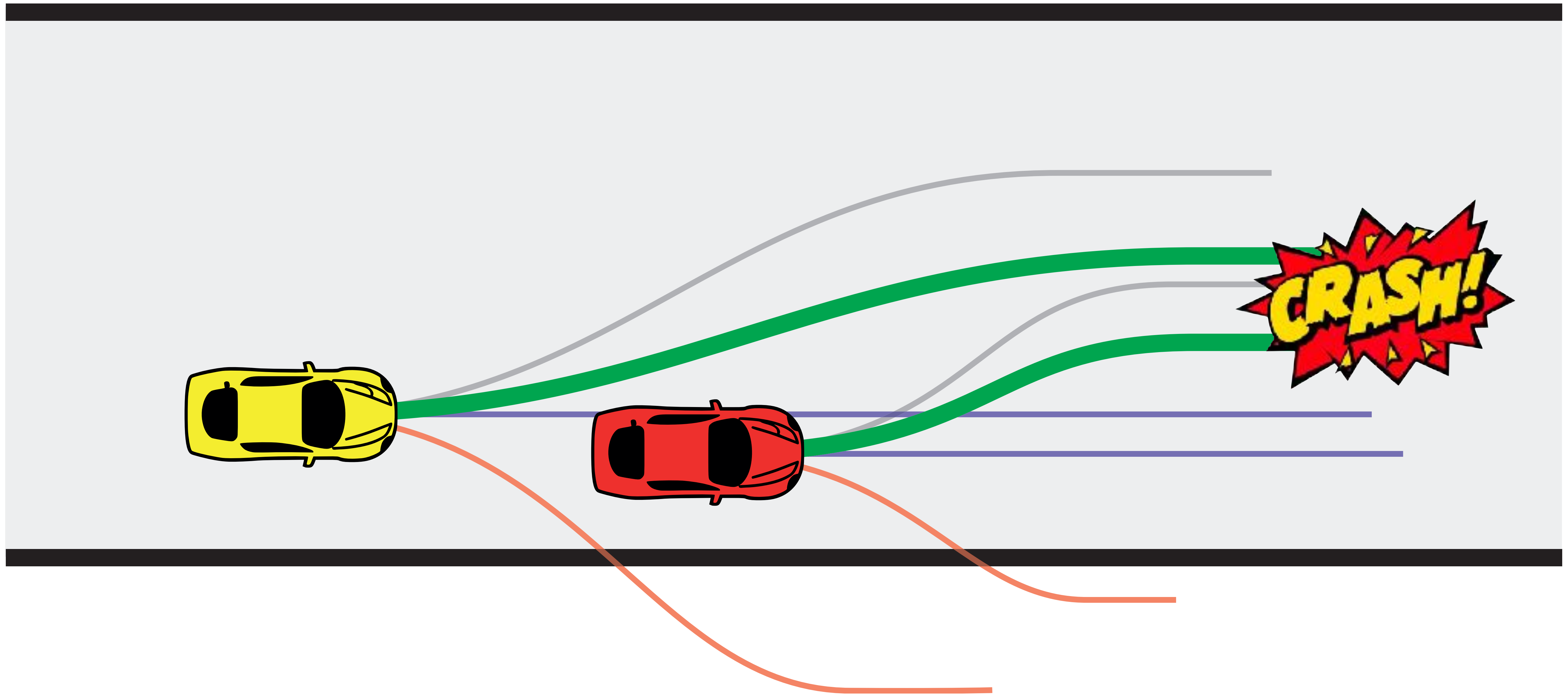
Blocking Trajectories



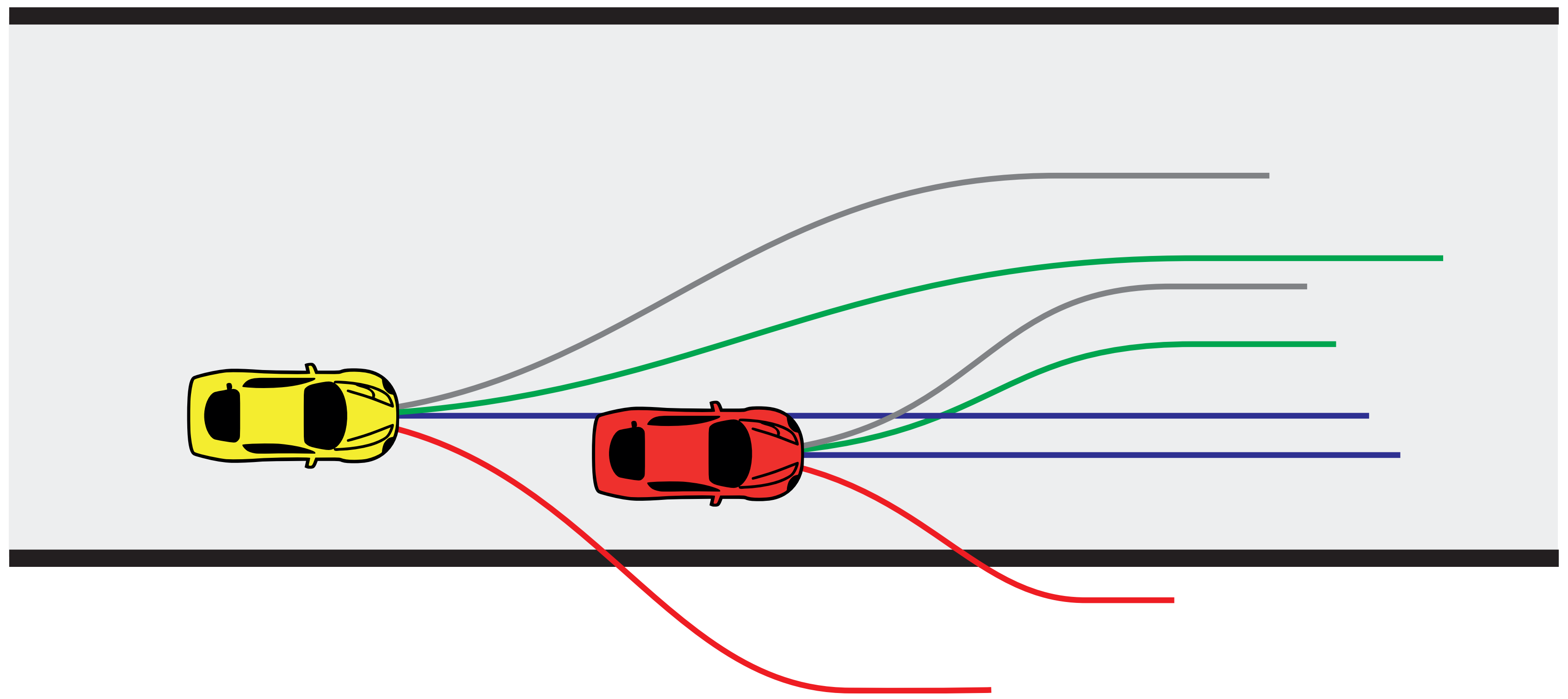
Blocking Trajectories



Blocking Trajectories



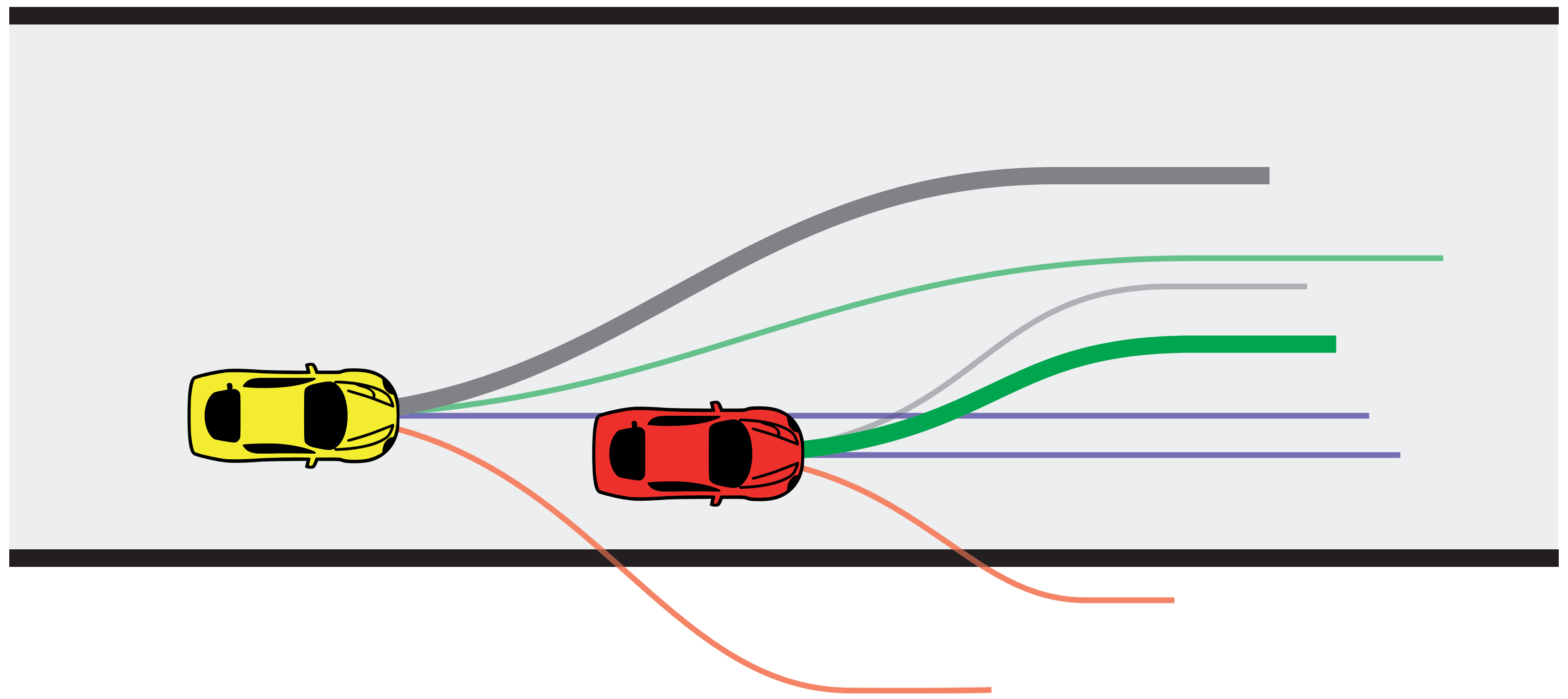
Blocking Trajectories



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

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

Blocking Trajectories

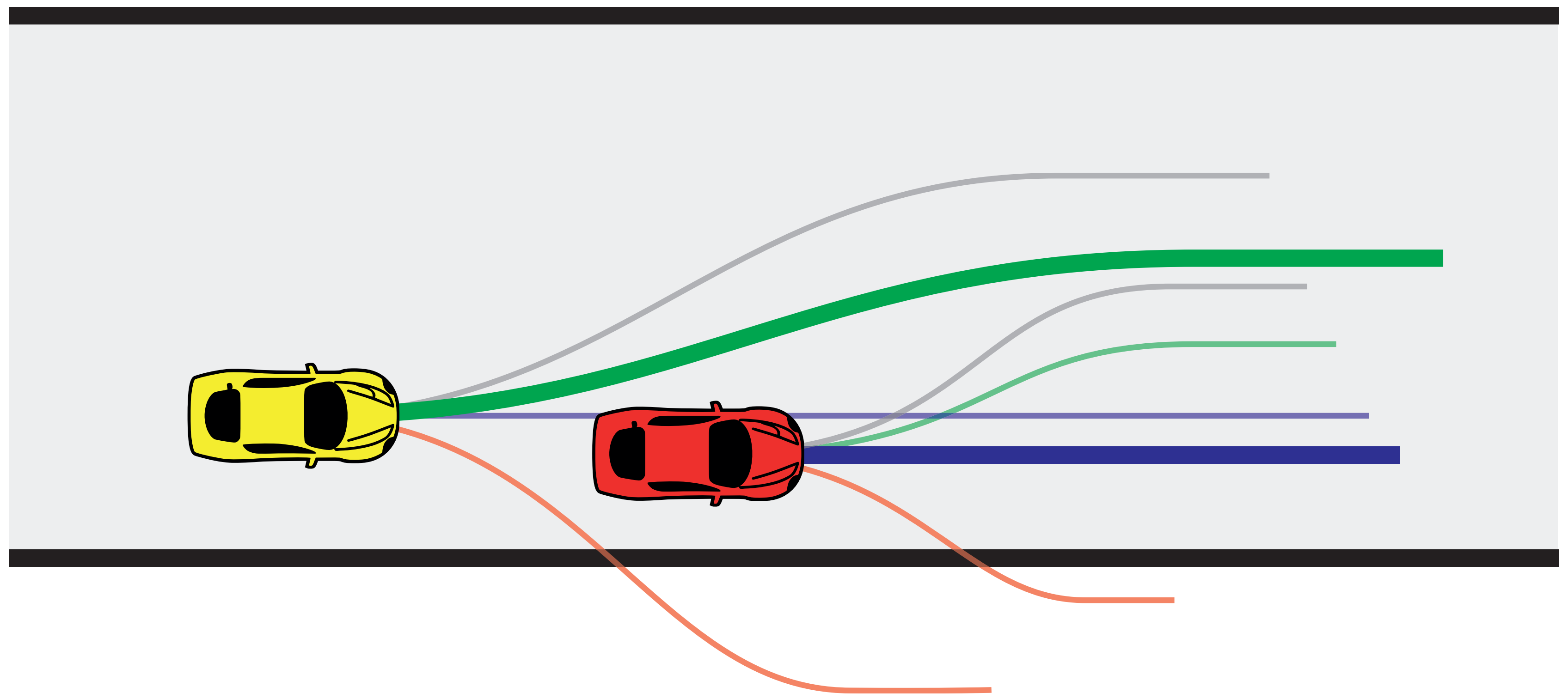


$$A = \begin{bmatrix} 0.83+0.5 & -1 & 0.83 & 0.83+0.5 \\ \boxed{0.85+0.5} & -1 & -1 & 0.85+0.5 \\ 0.88+0.5 & 0.88 & -1 & 0.88+0.5 \\ -10 & -10 & -10 & -10 \end{bmatrix}$$

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

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

Blocking Trajectories

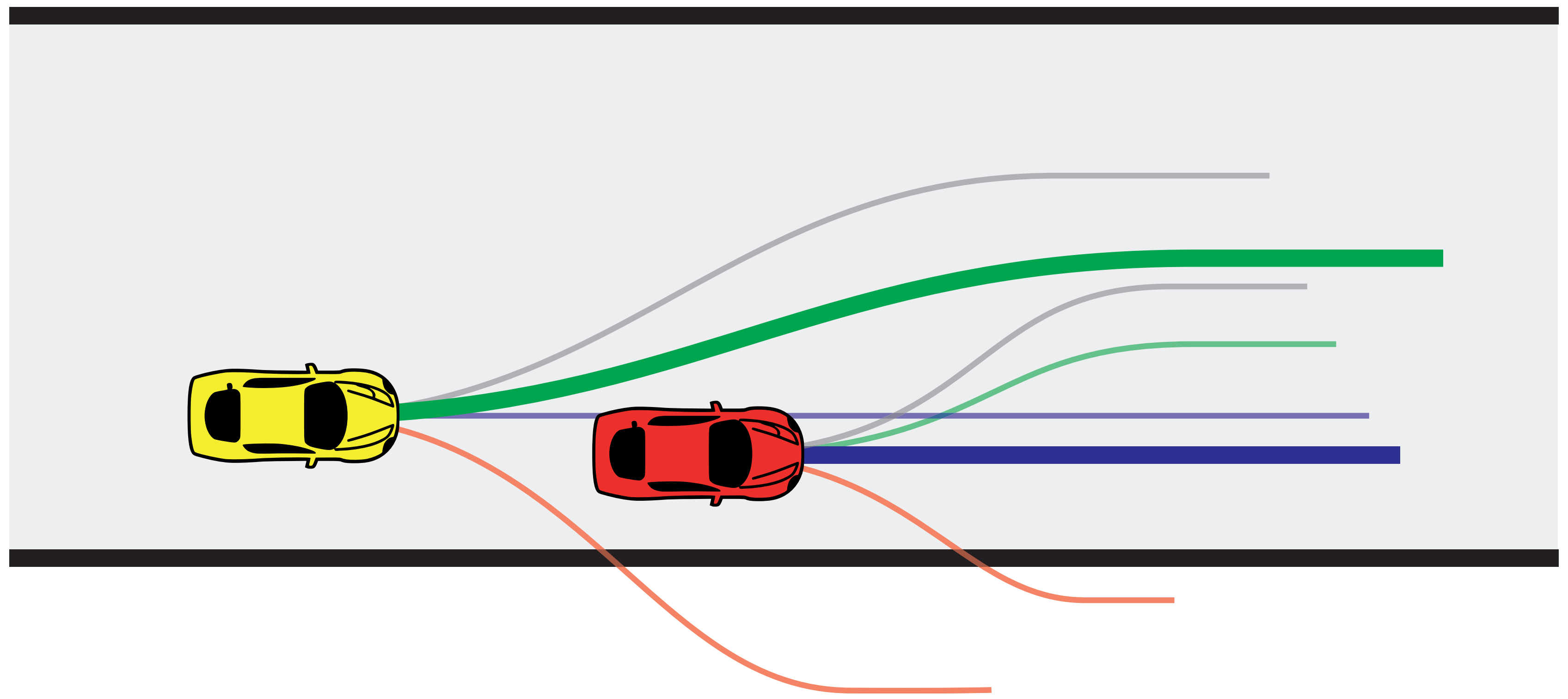


$$A = \begin{bmatrix} 0.83+0.5 & -1 & 0.83 & 0.83+0.5 \\ 0.85+0.5 & -1 & -1 & 0.85+0.5 \\ 0.88+0.5 & 0.88 & -1 & 0.88+0.5 \\ -10 & -10 & -10 & -10 \end{bmatrix}$$

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

- 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



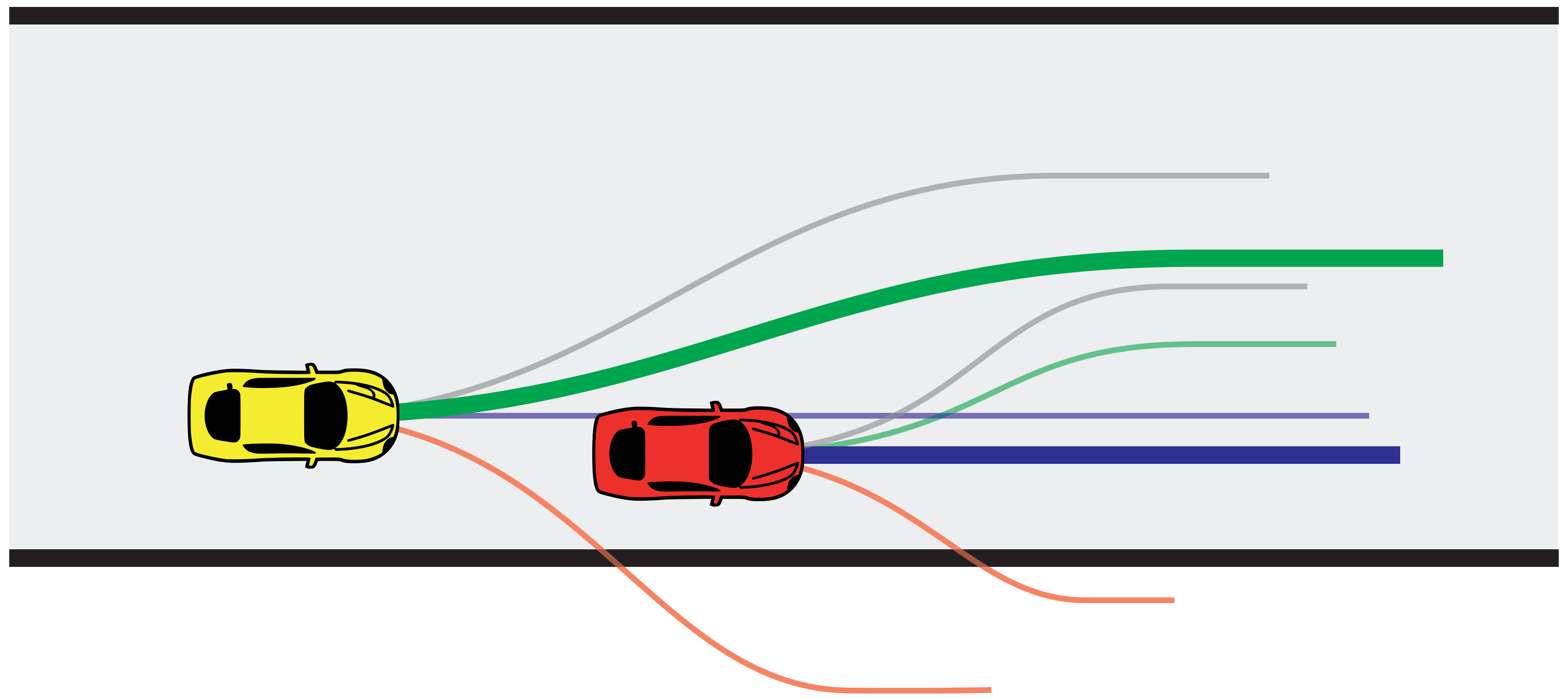
$$A = \begin{bmatrix} 0.83 + 0.5 & -1 & 0.83 & 0.83 + 0.5 \\ 0.85 + 0.5 & -1 & -1 & 0.85 + 0.5 \\ 0.88 + 0.5 & 0.88 & -1 & 0.88 + 0.5 \\ -10 & -10 & -10 & -10 \end{bmatrix}$$

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

- 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



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

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

- 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?

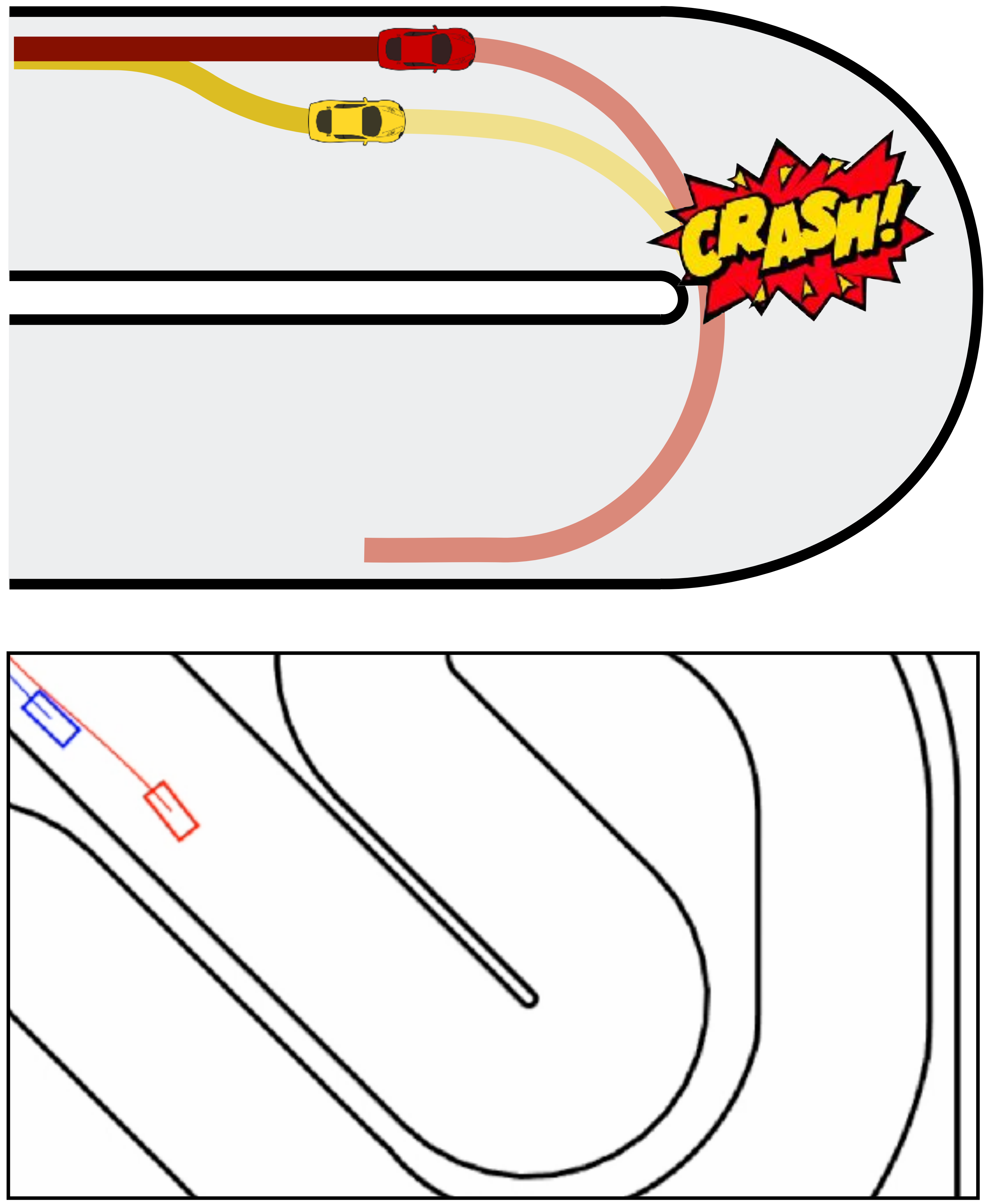
Simulation

- 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

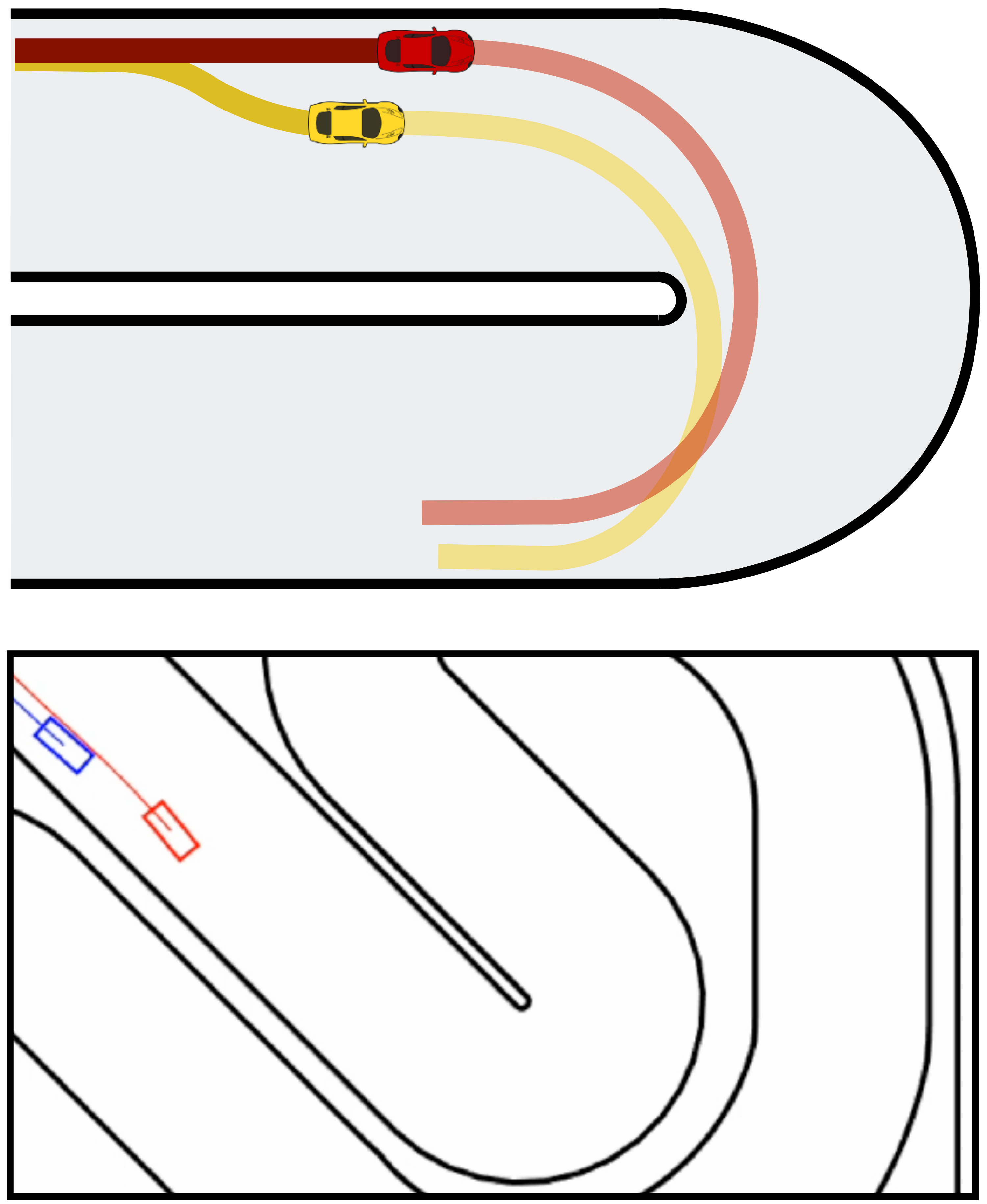
	sequential game	cooperative game	blocking game
# of overtaking maneuvers	113	857	414
colliding time steps per lap	2.4	2.0	2.3

Simulation

sequential game

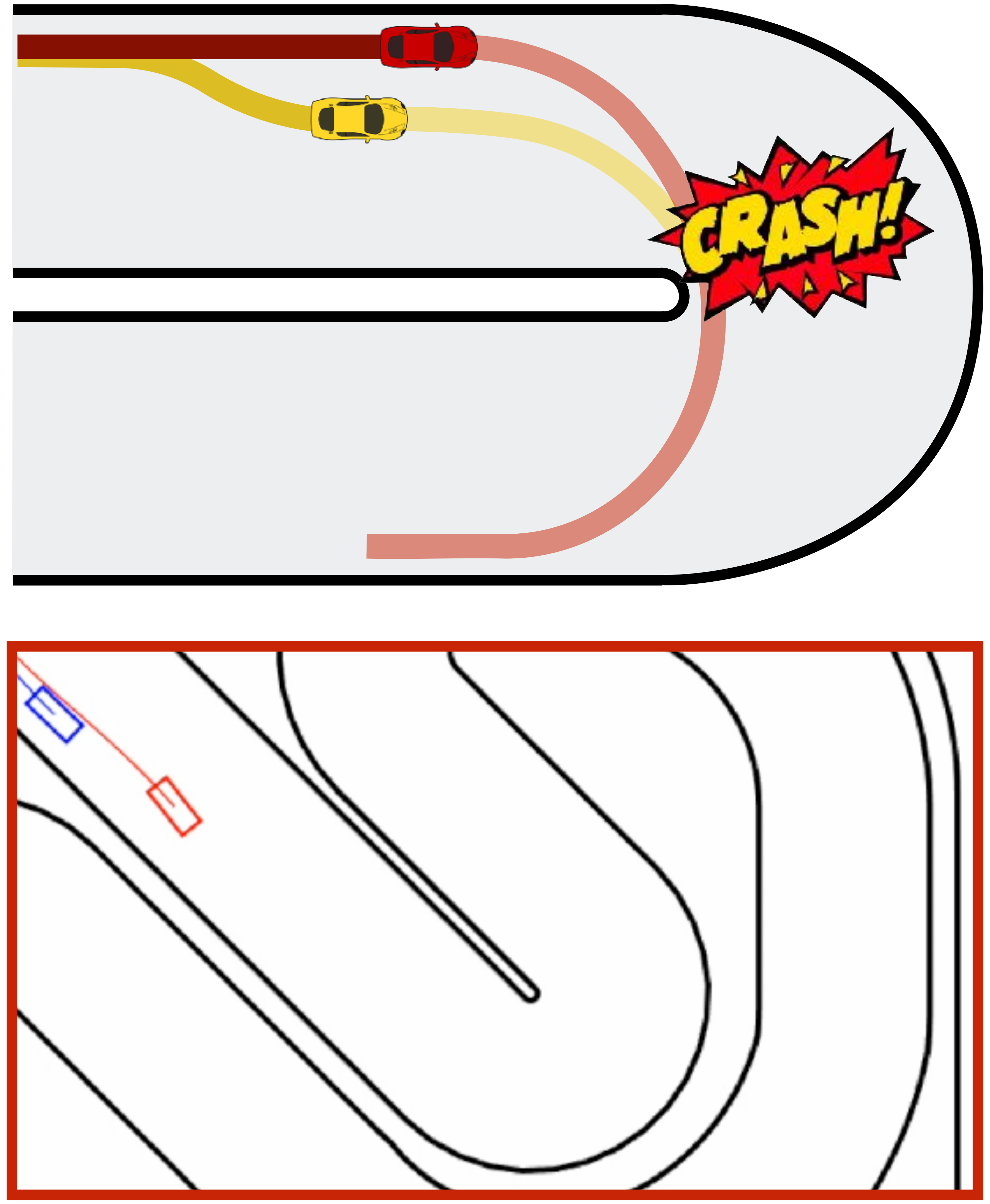


cooperative game

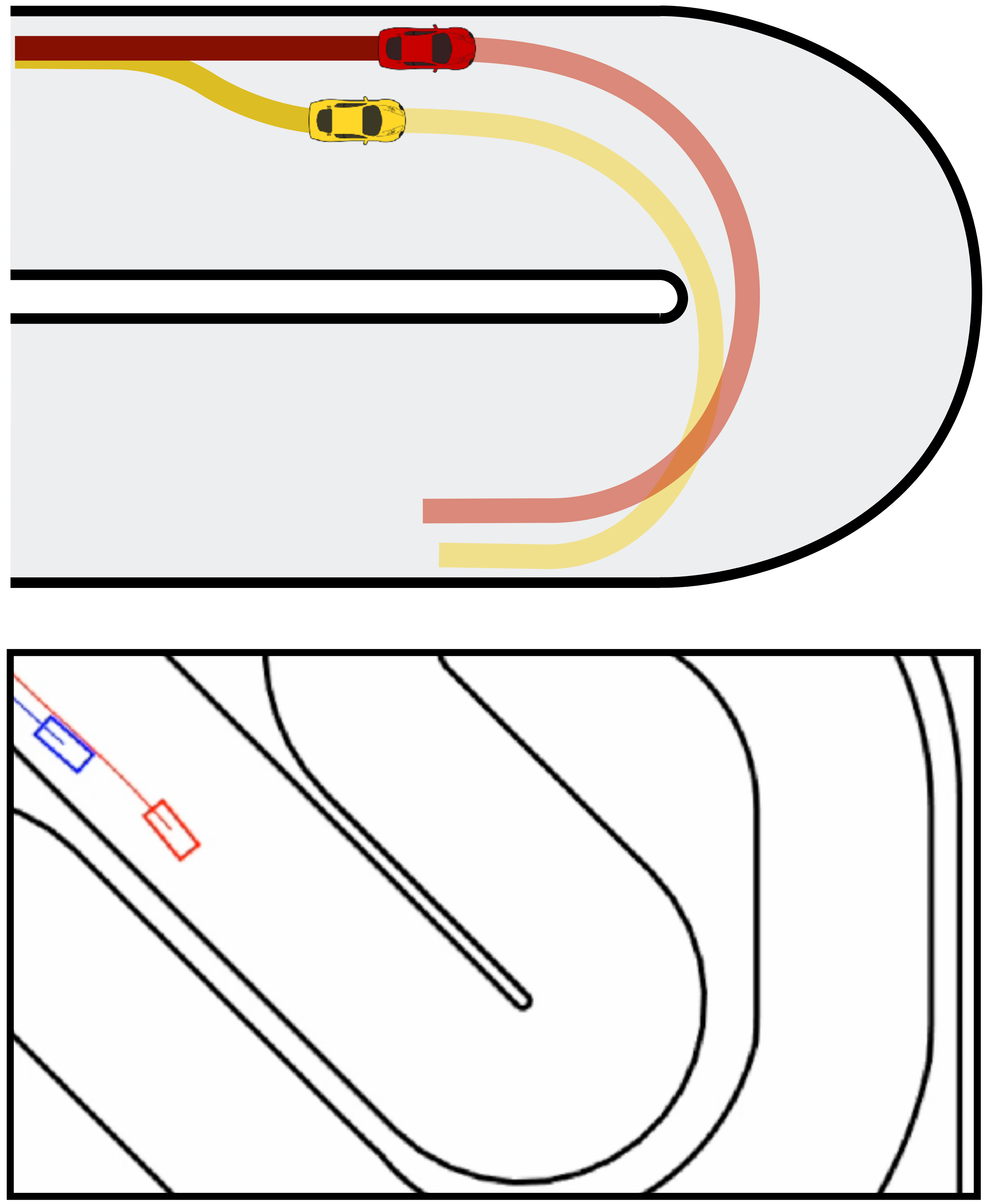


Simulation

sequential game

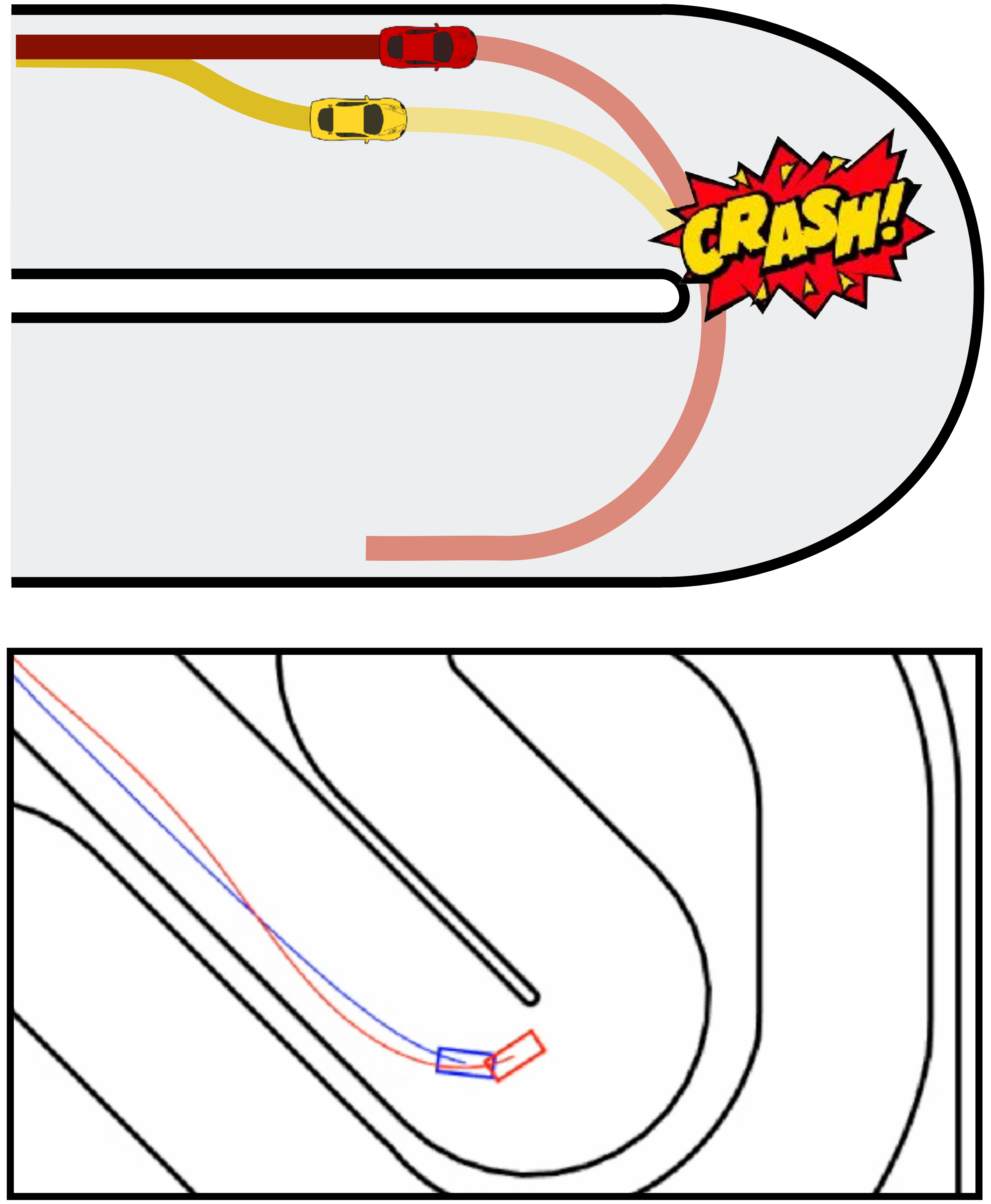


cooperative game

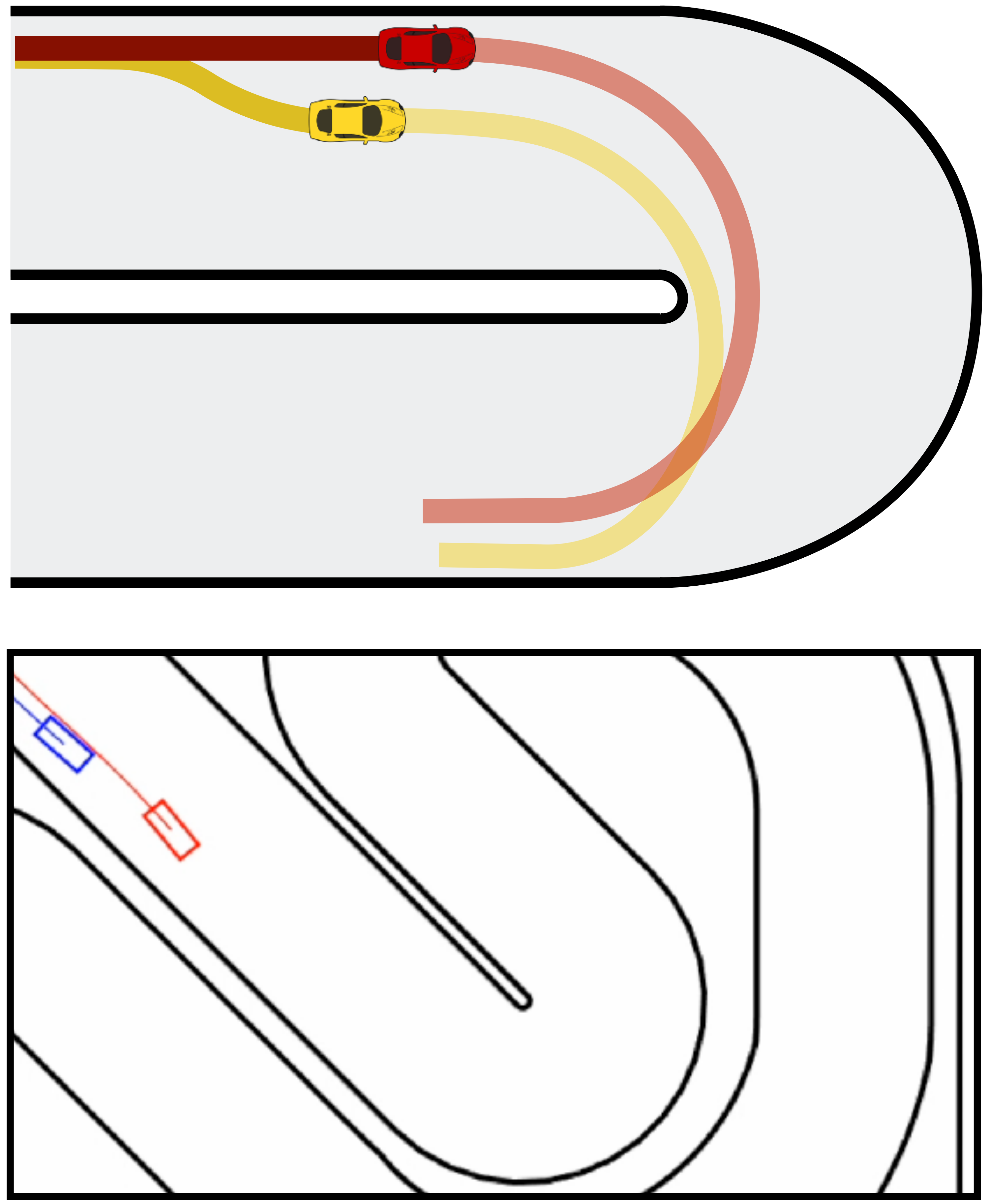


Simulation

sequential game

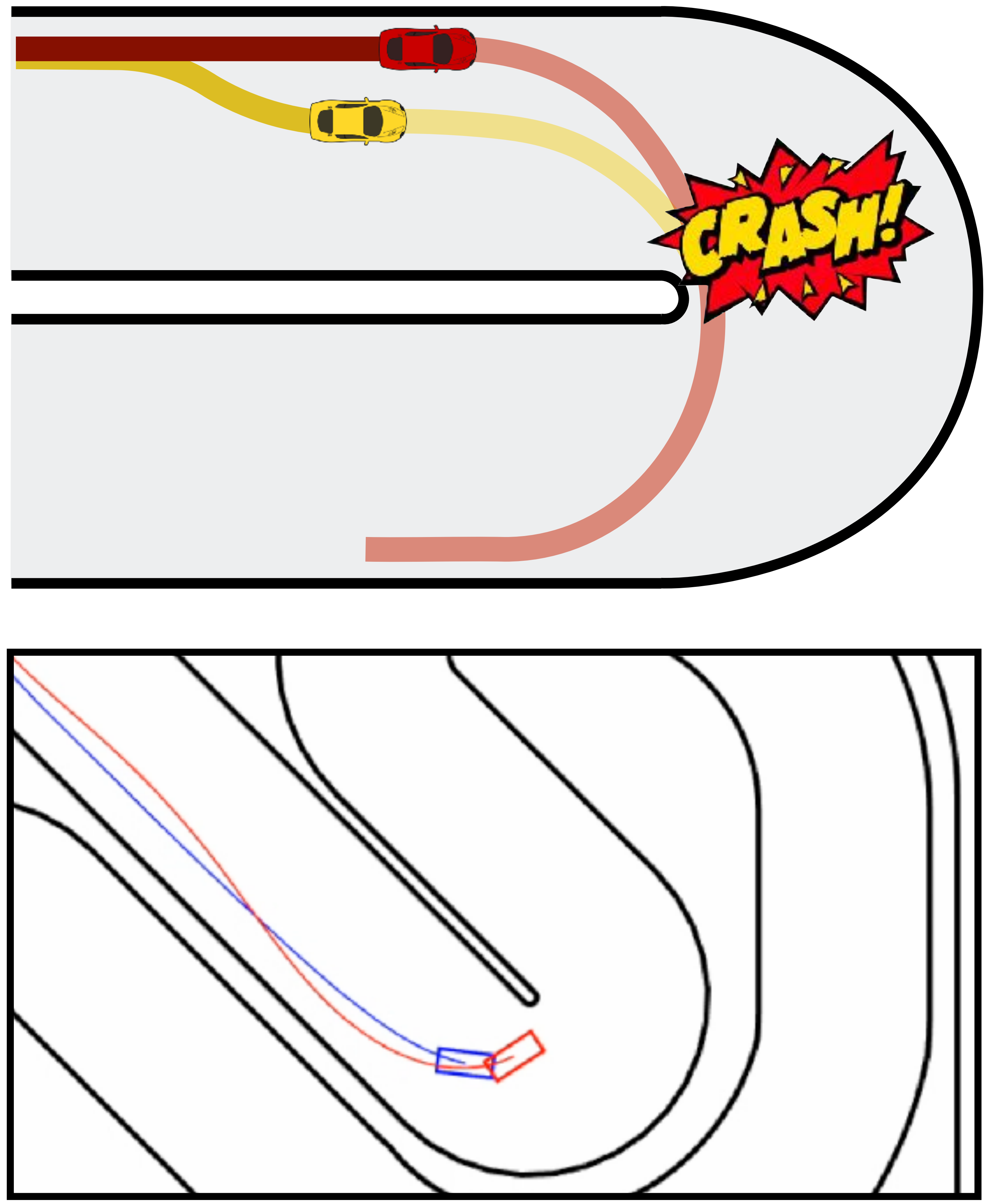


cooperative game

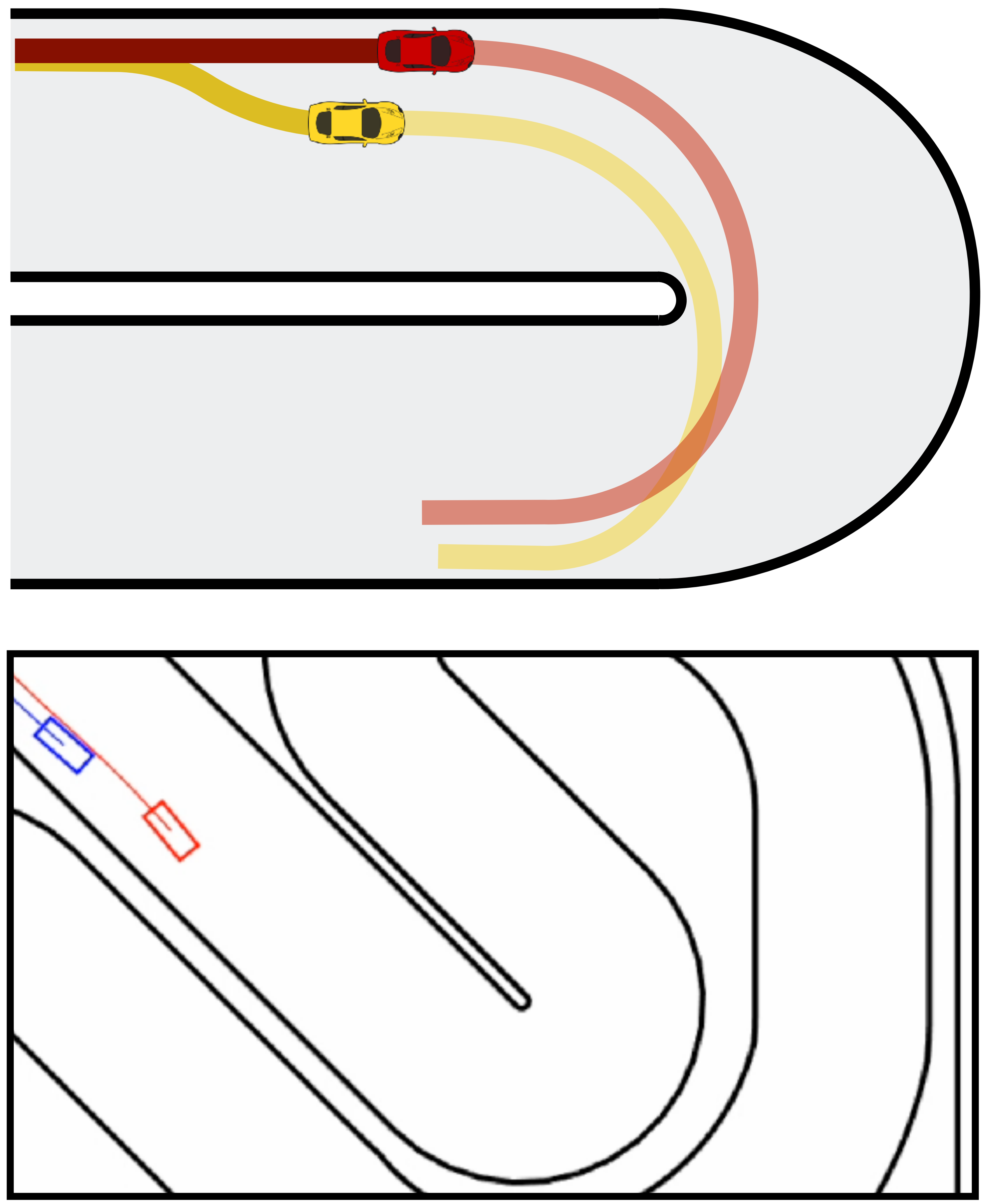


Simulation

sequential game

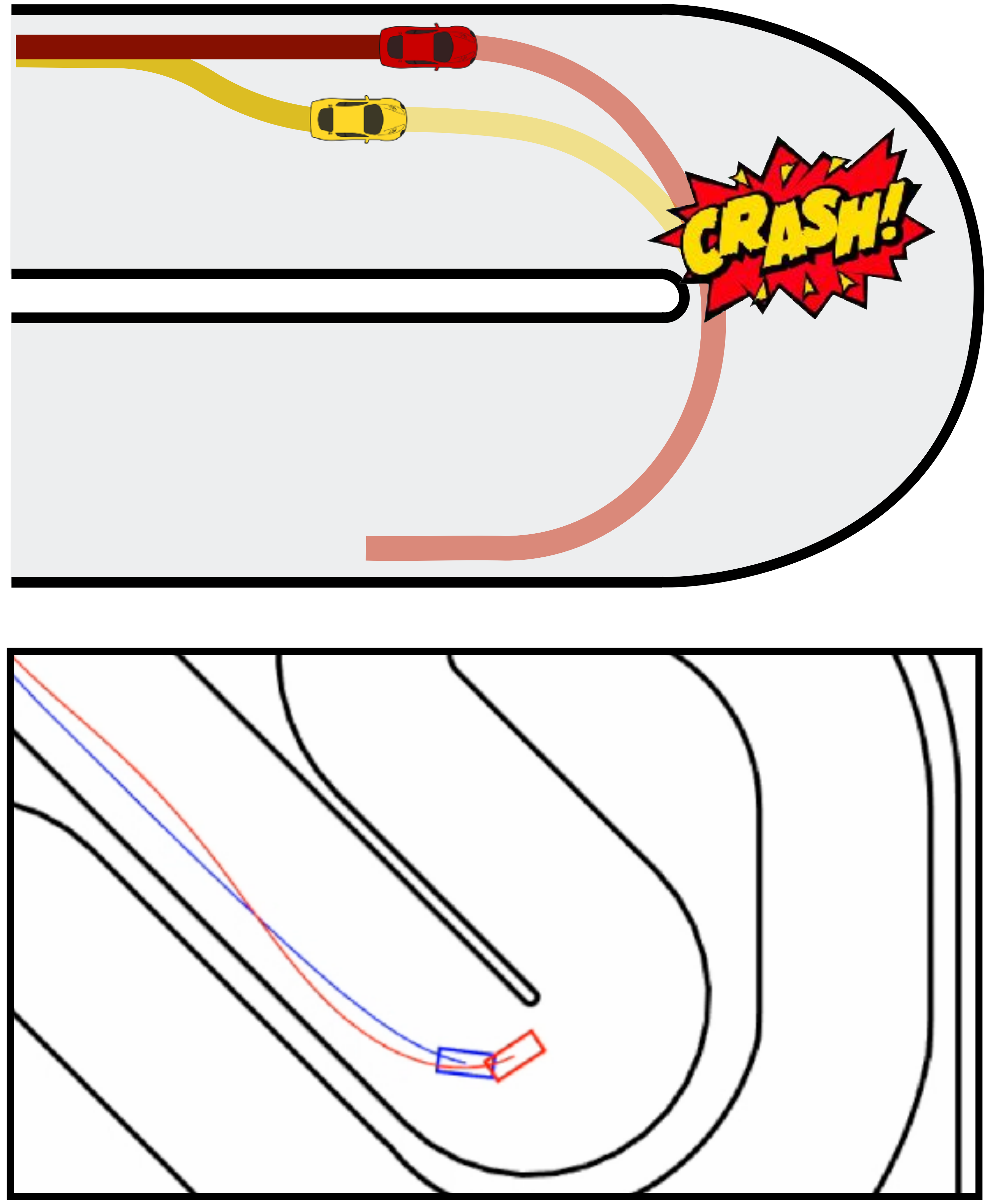


cooperative game

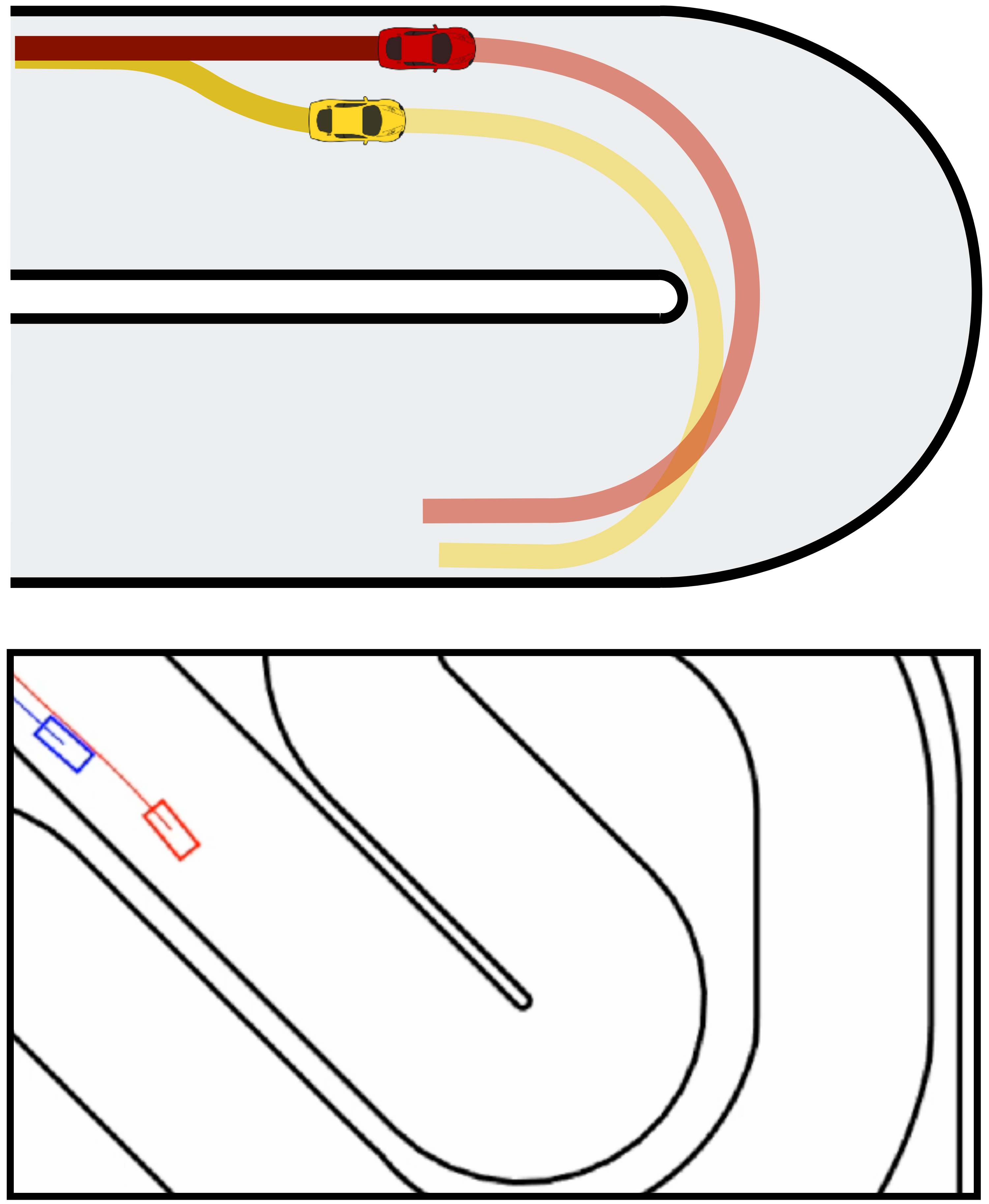


Simulation

sequential game

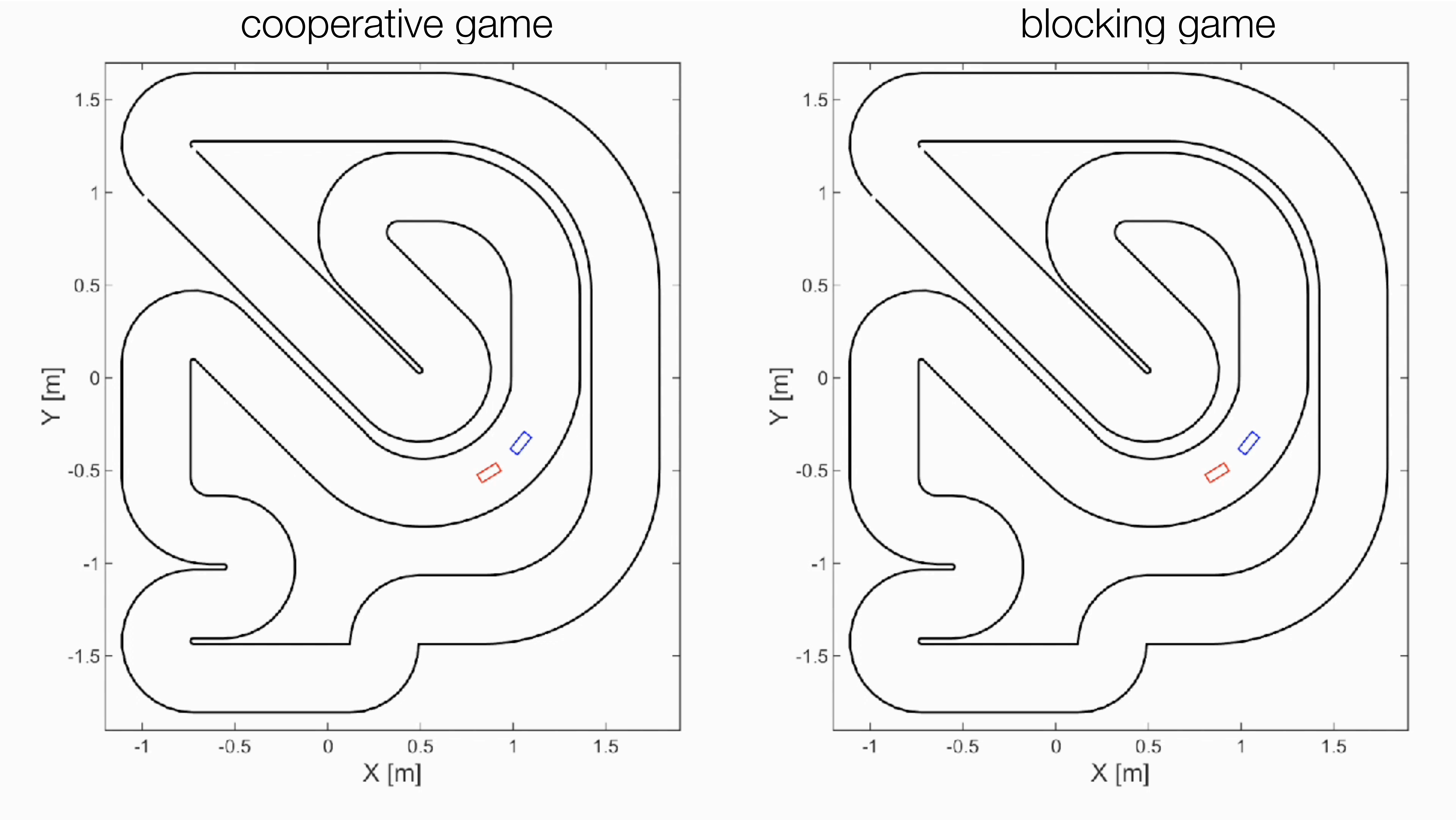


cooperative game



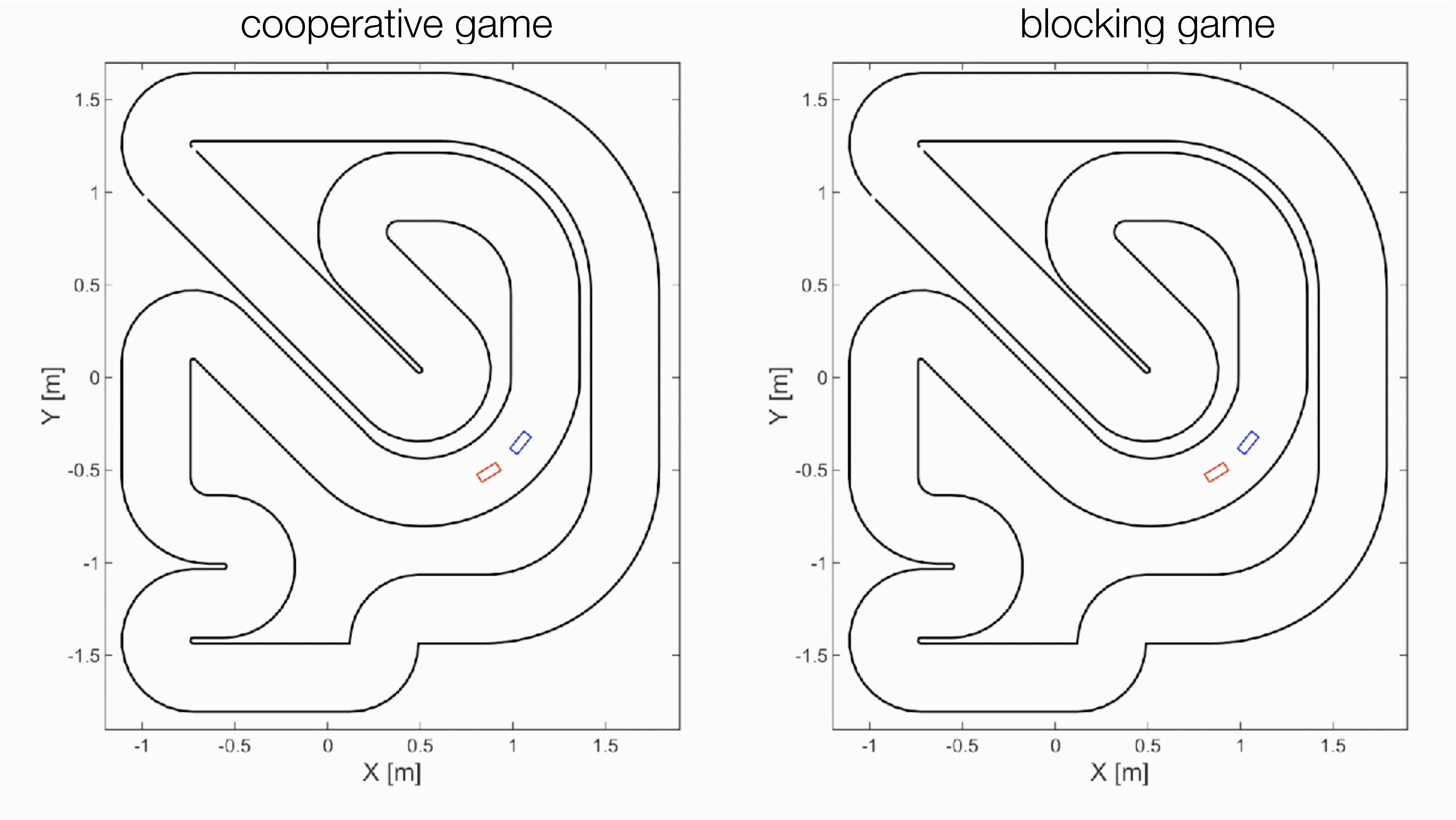
How do the cars drive?

Simulation



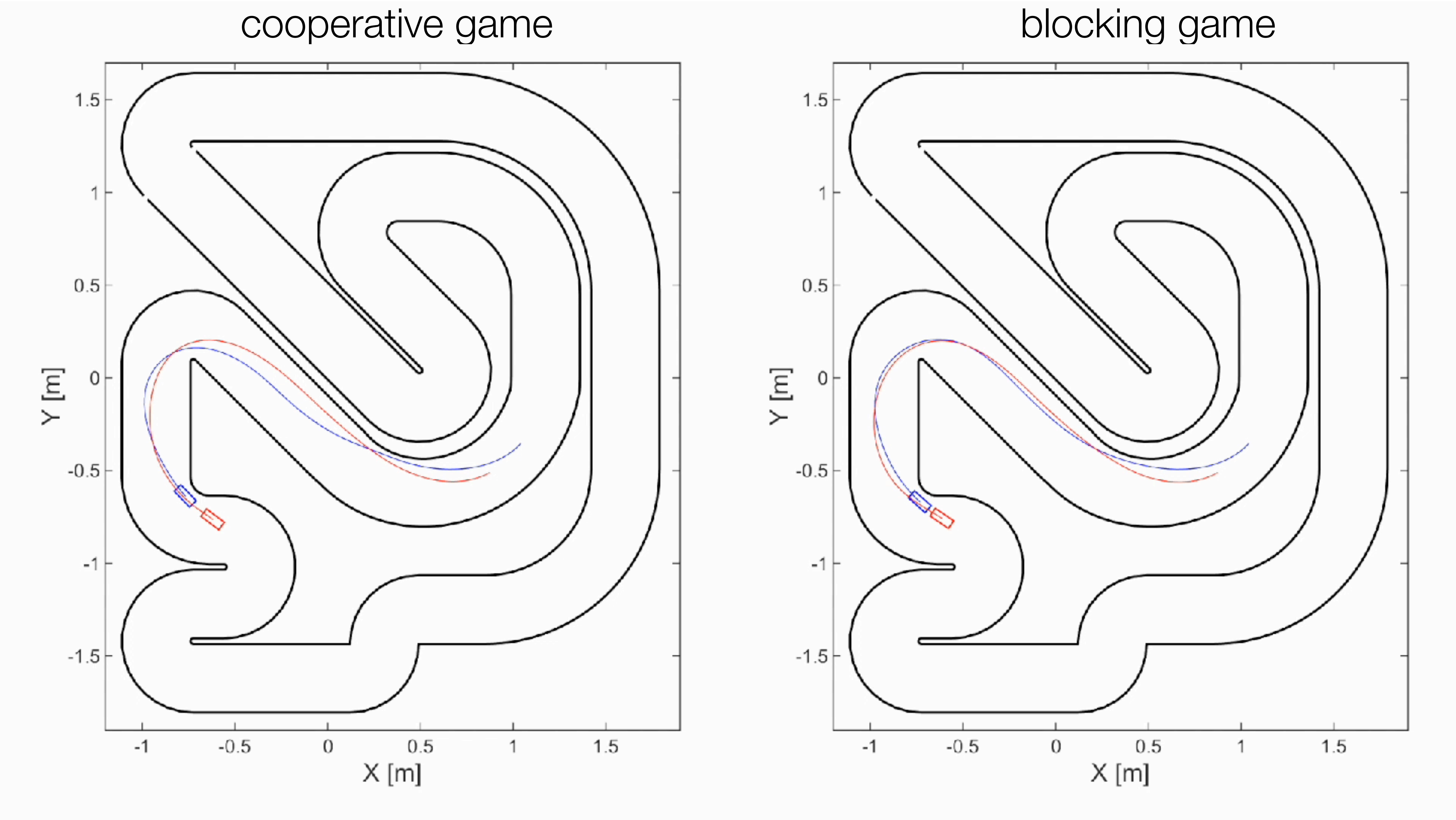
Viab -> aggressive driver   Disc -> cautious driver

Simulation



Viab -> aggressive driver   Disc -> cautious driver

Simulation

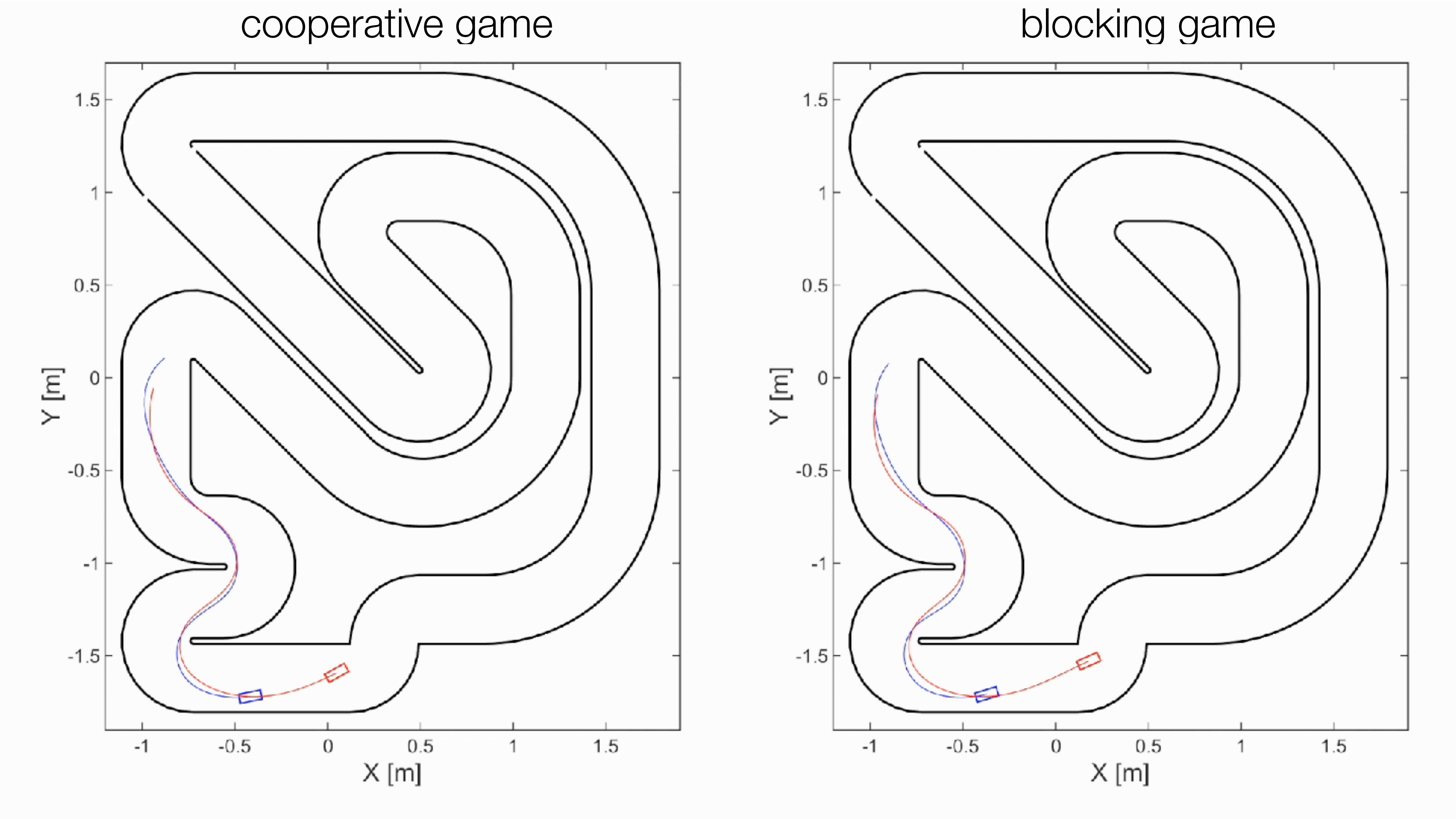


Viab -> aggressive driver

Disc -> cautious driver

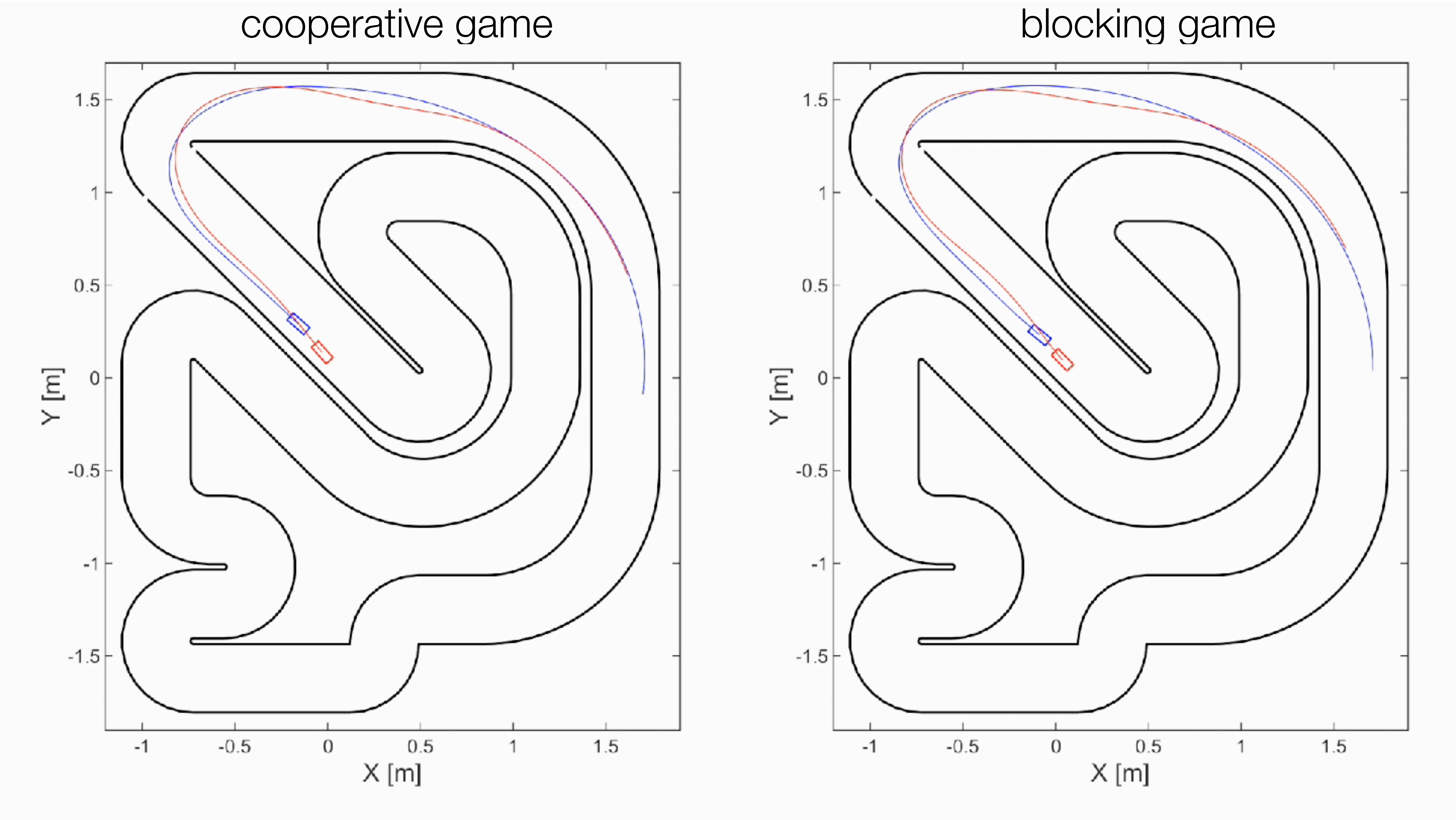
ETH zürich AUTOMATIC CONTROL LABORATORY IfA CVL

Simulation



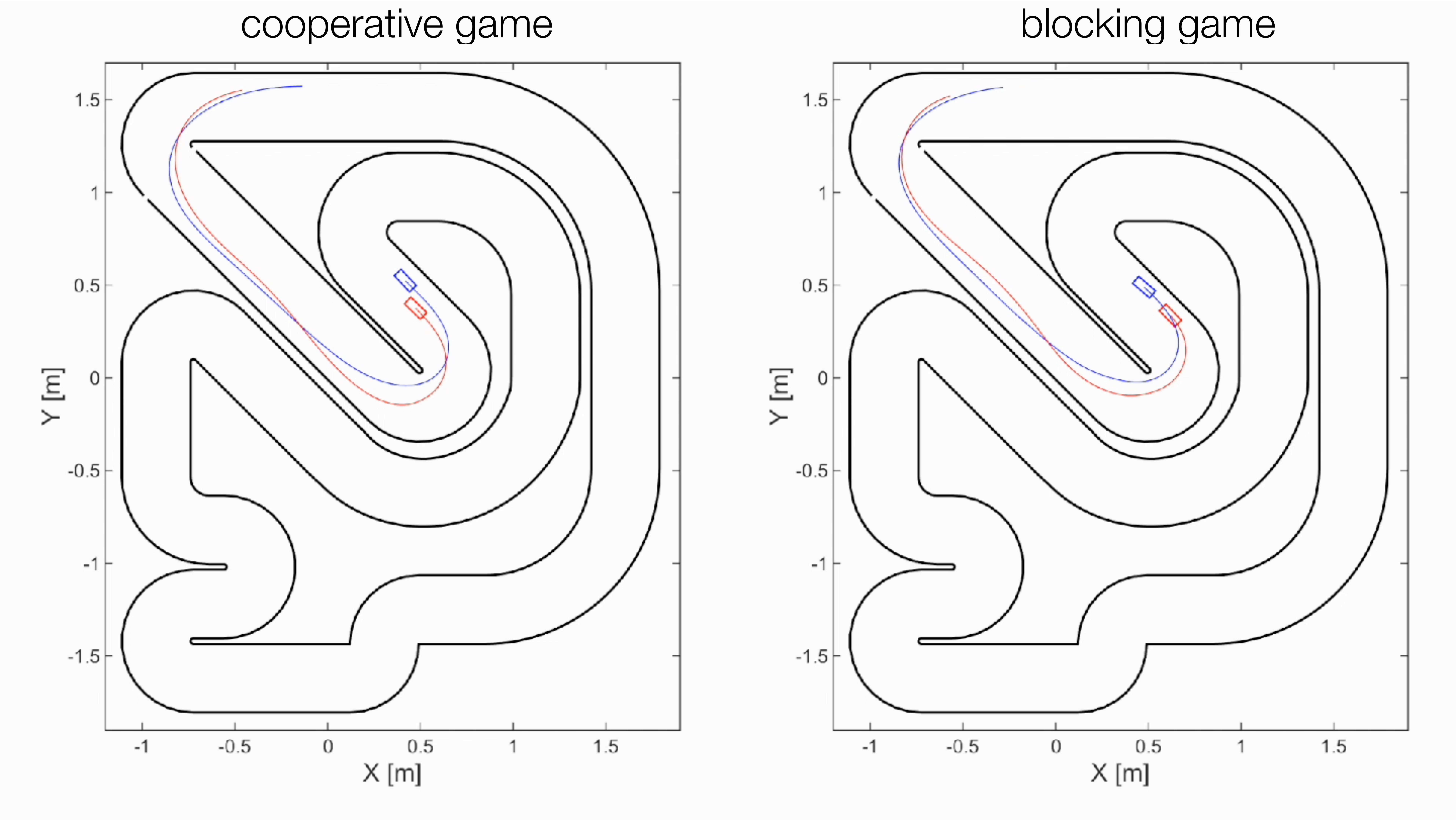
Viab -> aggressive driver   Disc -> cautious driver

Simulation



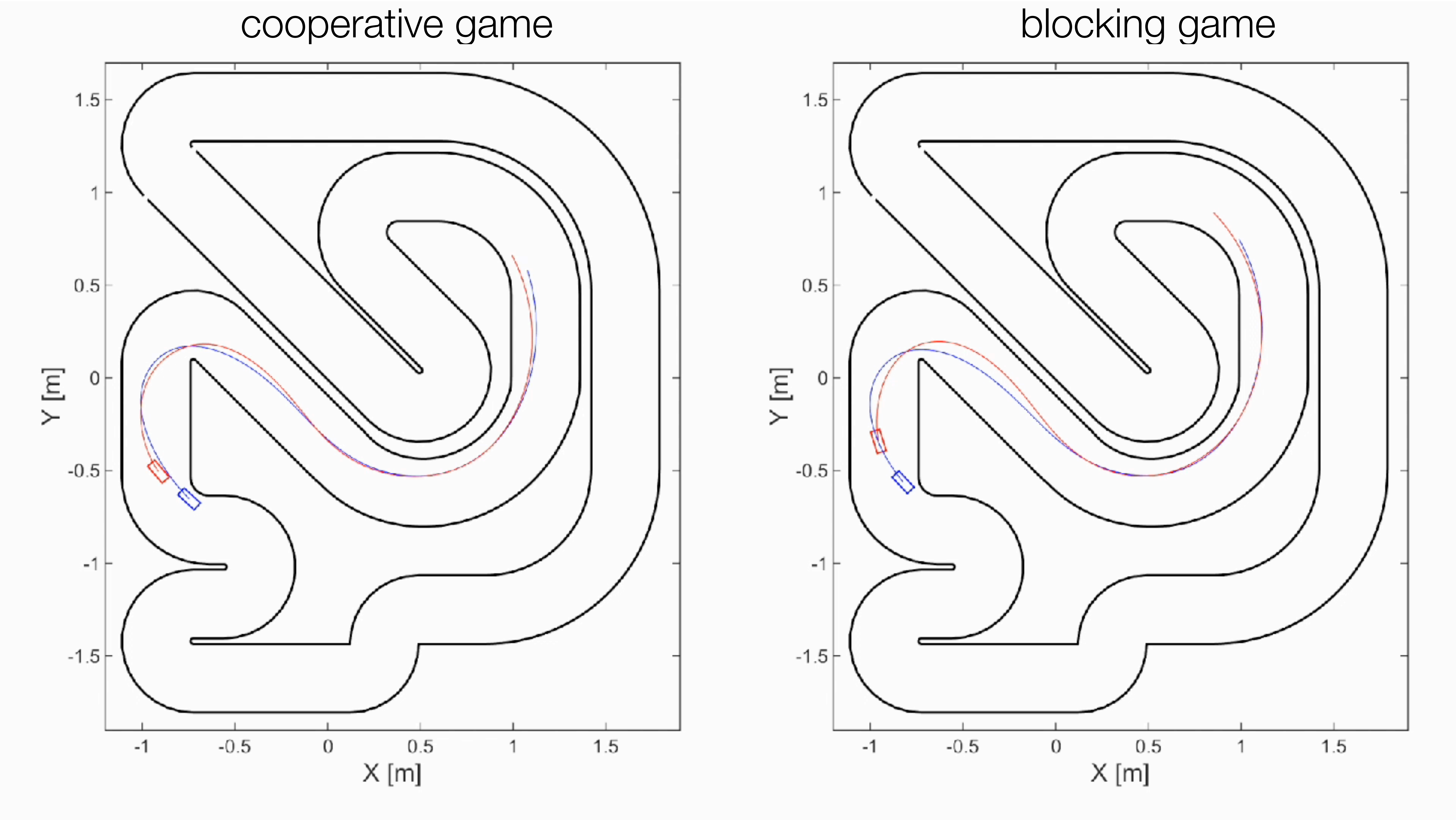
Viab -> aggressive driver   Disc -> cautious driver

Simulation



Viab -> aggressive driver
Disc -> cautious driver

Simulation

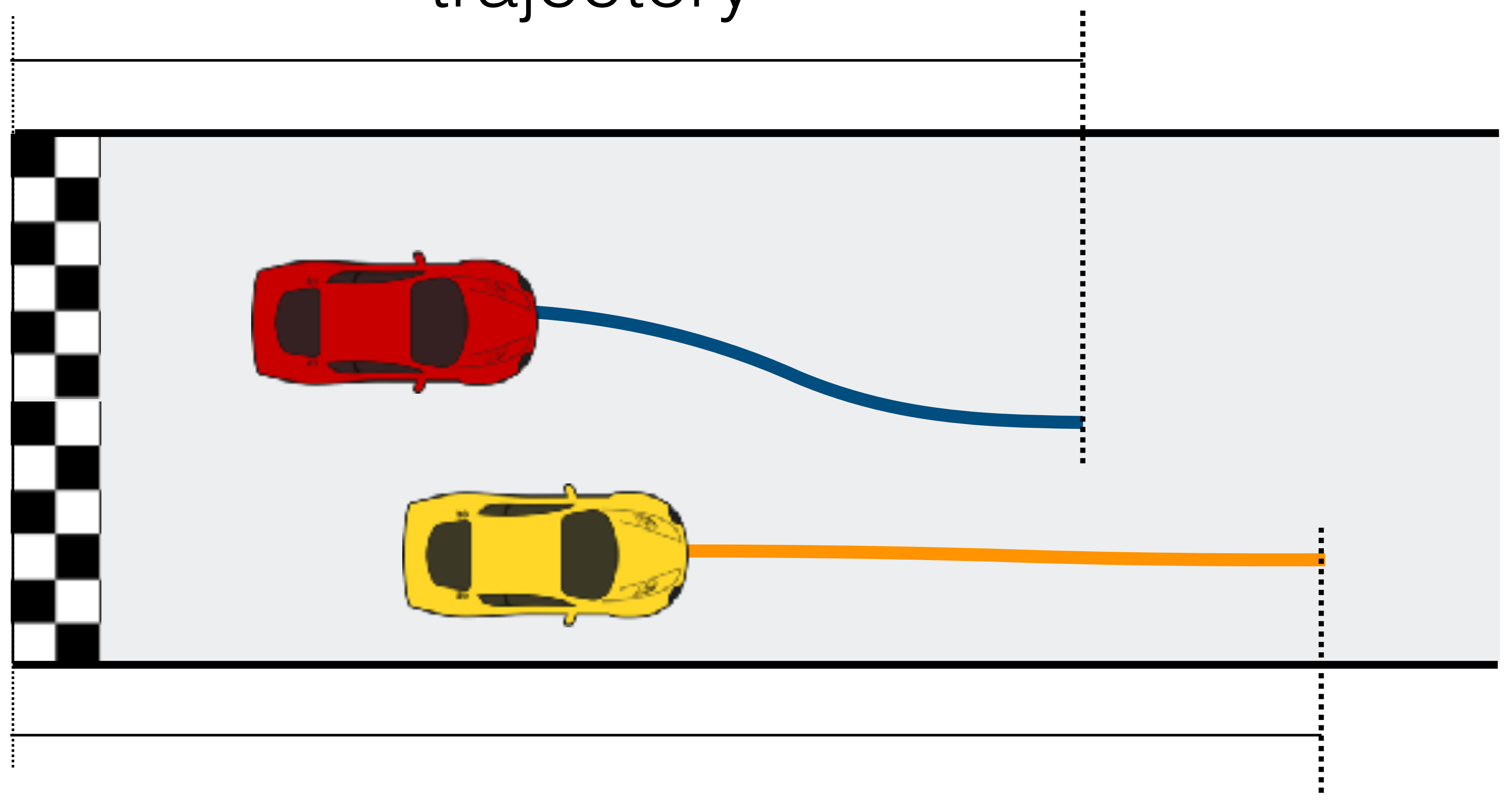


Viab -> aggressive driver
Disc -> cautious driver

Real-Time Implementation

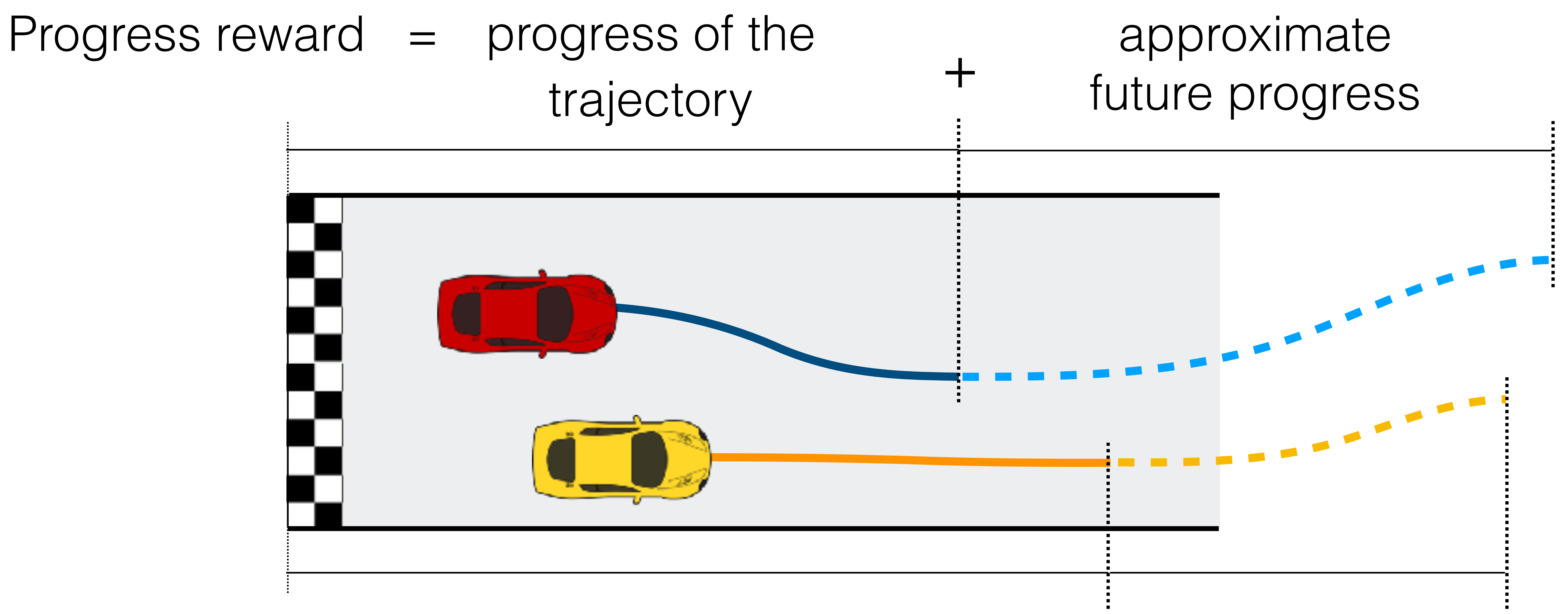
- Building the bimatrix game can be computationally expensive
  - Entries in the matrix grow quadratically with the number of trajectories
  - Collision checks become a bottle neck
  - 1,000 trajectories —> 32,000,000 collision checks (20ms discretization)
- Reduce prediction horizon from 3 to 2 steps (~200 instead of 3,000 trajectories)
- Only build bimatrix game for the 60 best trajectories (beam search)

Progress reward = progress of the trajectory



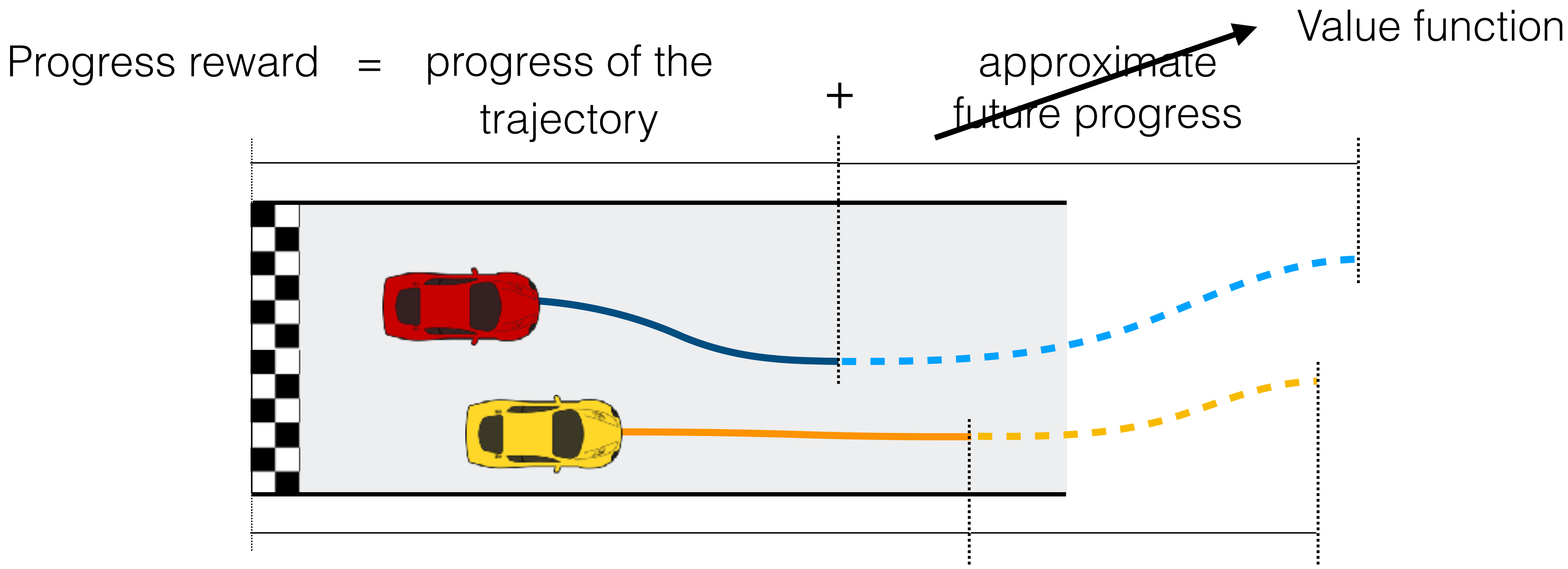
Real-Time Implementation

- Building the bimatrix game can be computationally expensive
  - Entries in the matrix grow quadratically with the number of trajectories
  - Collision checks become a bottle neck
  - 1,000 trajectories —> 32,000,000 collision checks (20ms discretization)
- Reduce prediction horizon from 3 to 2 steps (~200 instead of 3,000 trajectories)
- Only build bimatrix game for the 60 best trajectories (beam search)



Real-Time Implementation

- Building the bimatrix game can be computationally expensive
  - Entries in the matrix grow quadratically with the number of trajectories
  - Collision checks become a bottle neck
  - 1,000 trajectories —> 32,000,000 collision checks (20ms discretization)
- Reduce prediction horizon from 3 to 2 steps (~200 instead of 3,000 trajectories)
- Only build bimatrix game for the 60 best trajectories (beam search)



Experimental Results

Experimental Results

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 noncooperative 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

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

- Learn behavior of “opponent” —> urban traffic

Questions