Learning how to autonomously race a car: a predictive control approach

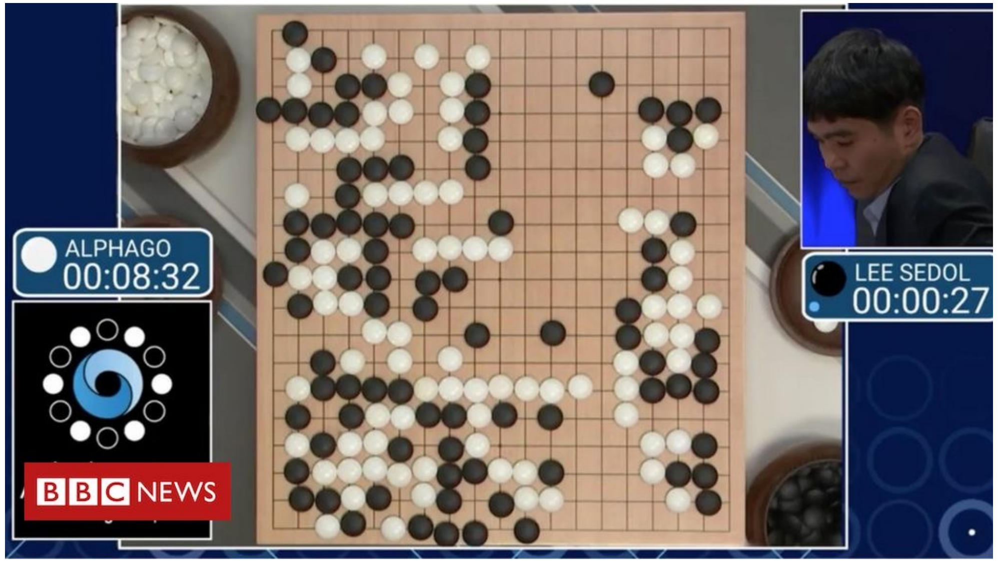
Ugo Rosolia

AMBER Lab
California Institute of Technology

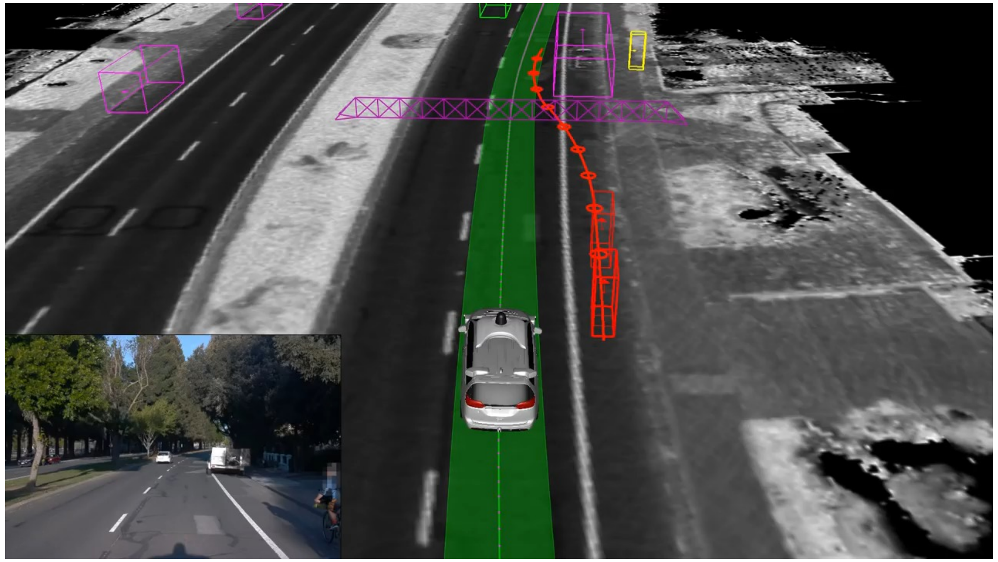
June, 2021

Success Stories from AI

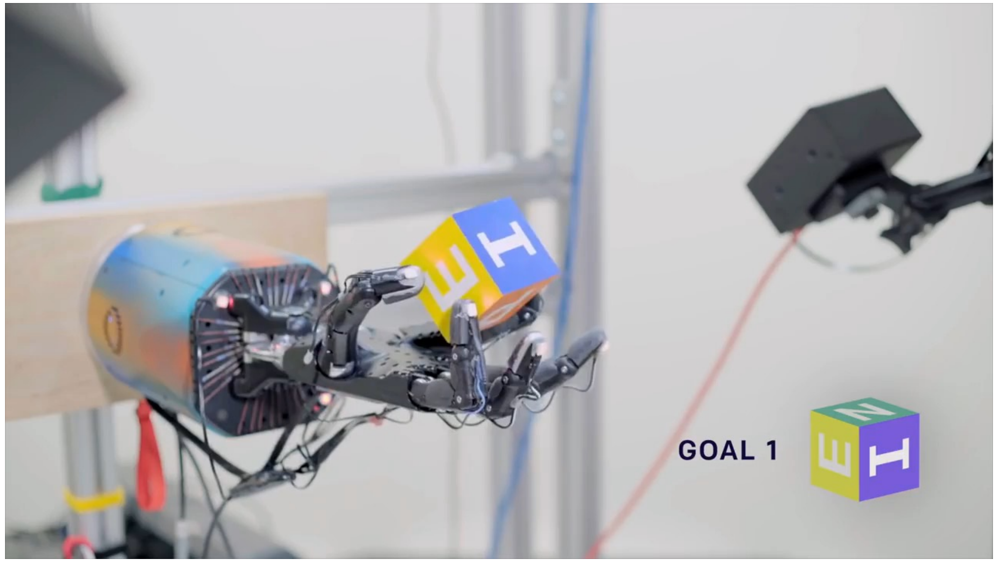
Alpha GO



Waymo's Perception Module



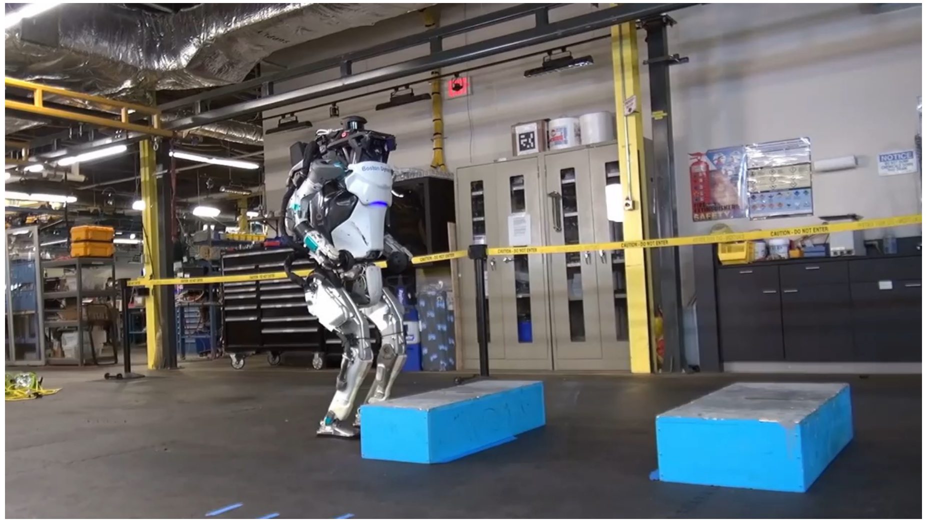
OpenAI



Google

Success Stories from Control Theory

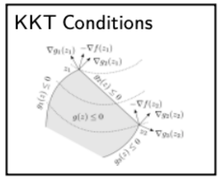
Boston Dynamics



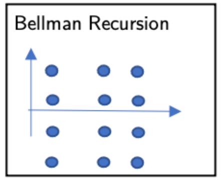
Stanford Dynamic Design Lab

Standard Control Pipeline

Optimal Trajectory

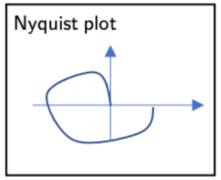


Optimization

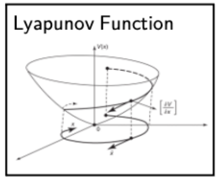


Dynamic Programming

Trajectory Tracking

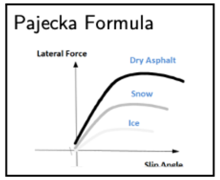


Frequency Domain

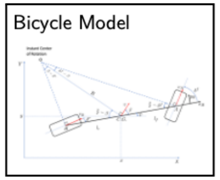


Nonlinear Control

System Identification

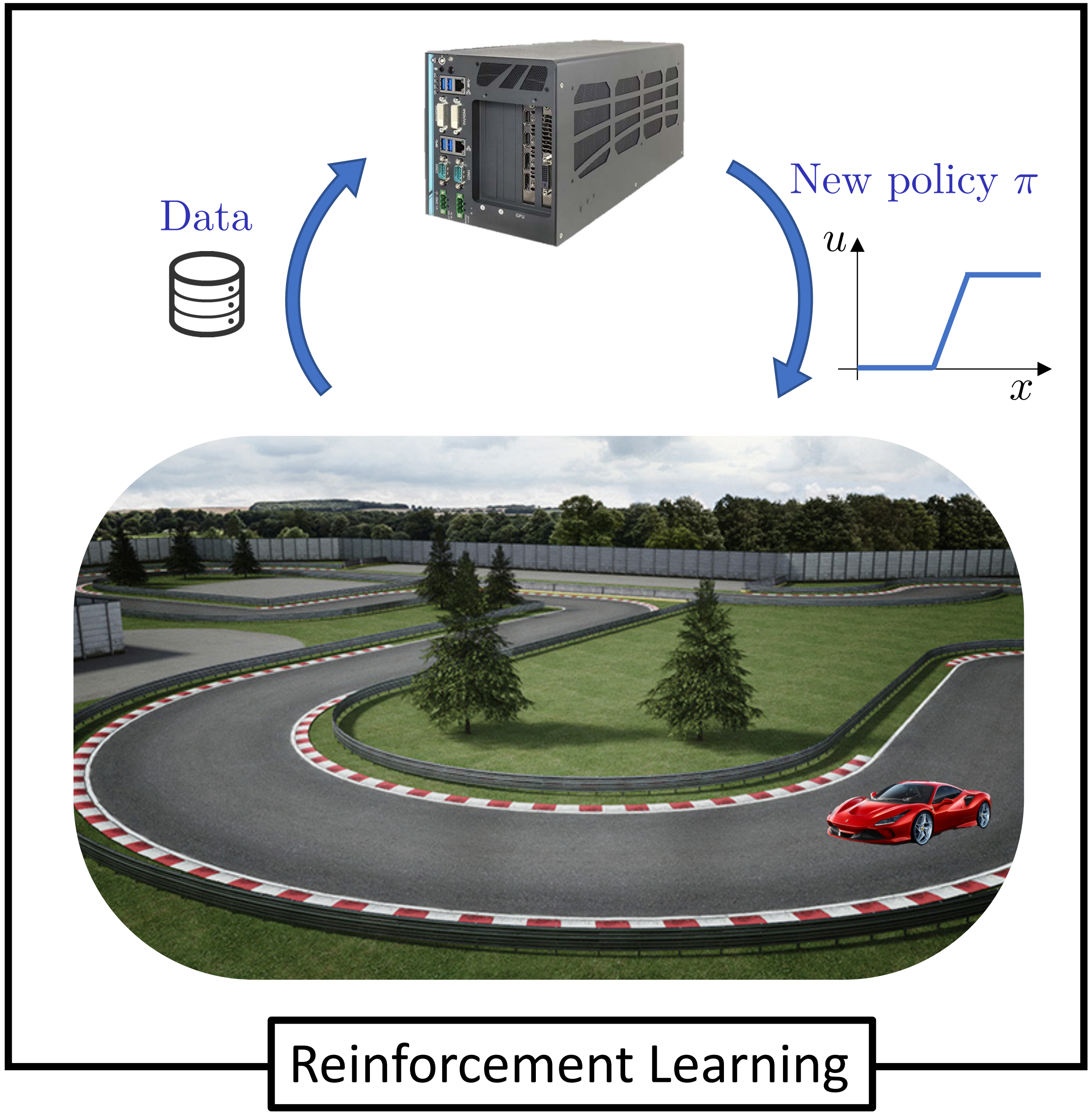


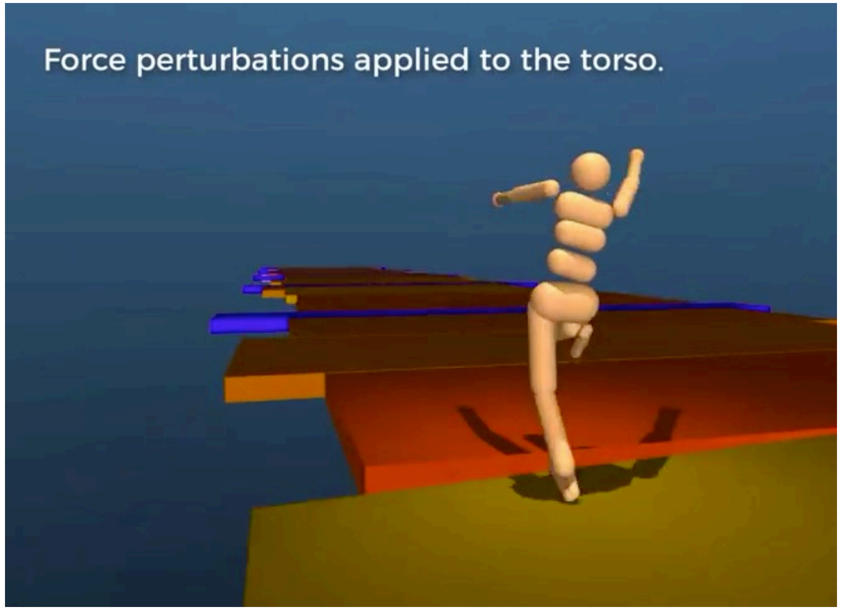
Tire Dynamics



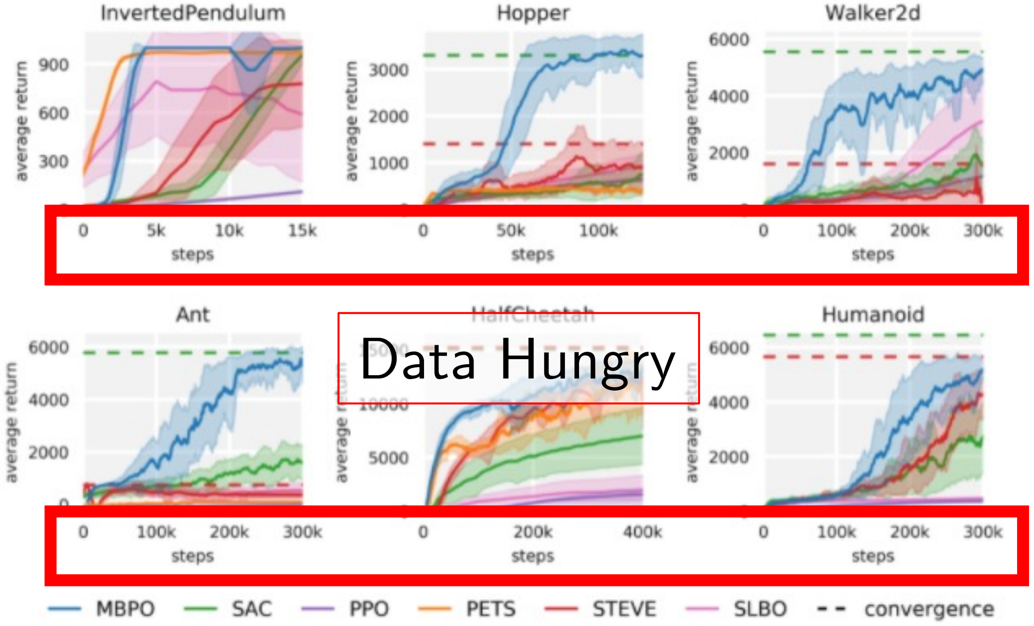
Vehicle Dynamics

Can we simplify the control design?



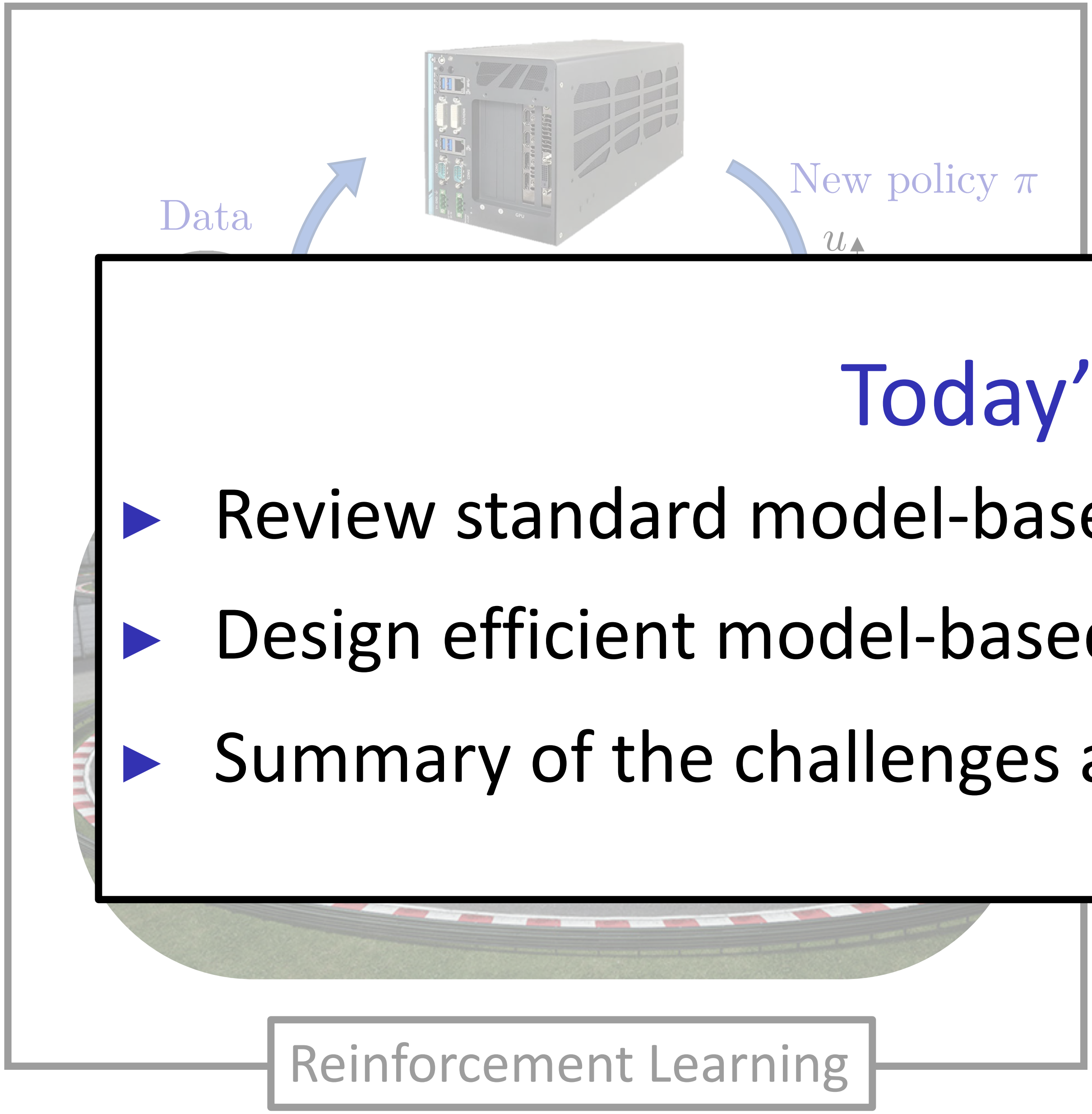


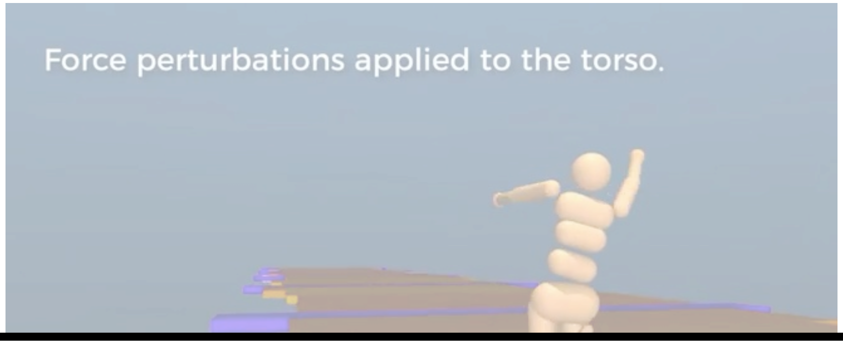
DeepMind



M. Janner, J. Fu, M. Zhang, and S. Levine. "When to trust your model: Model-based policy optimization." arXiv preprint arXiv:1906.08253 (2019).

Can we simplify the control design?

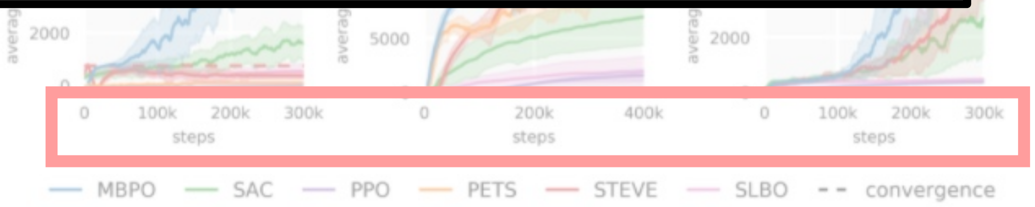




DeepMind

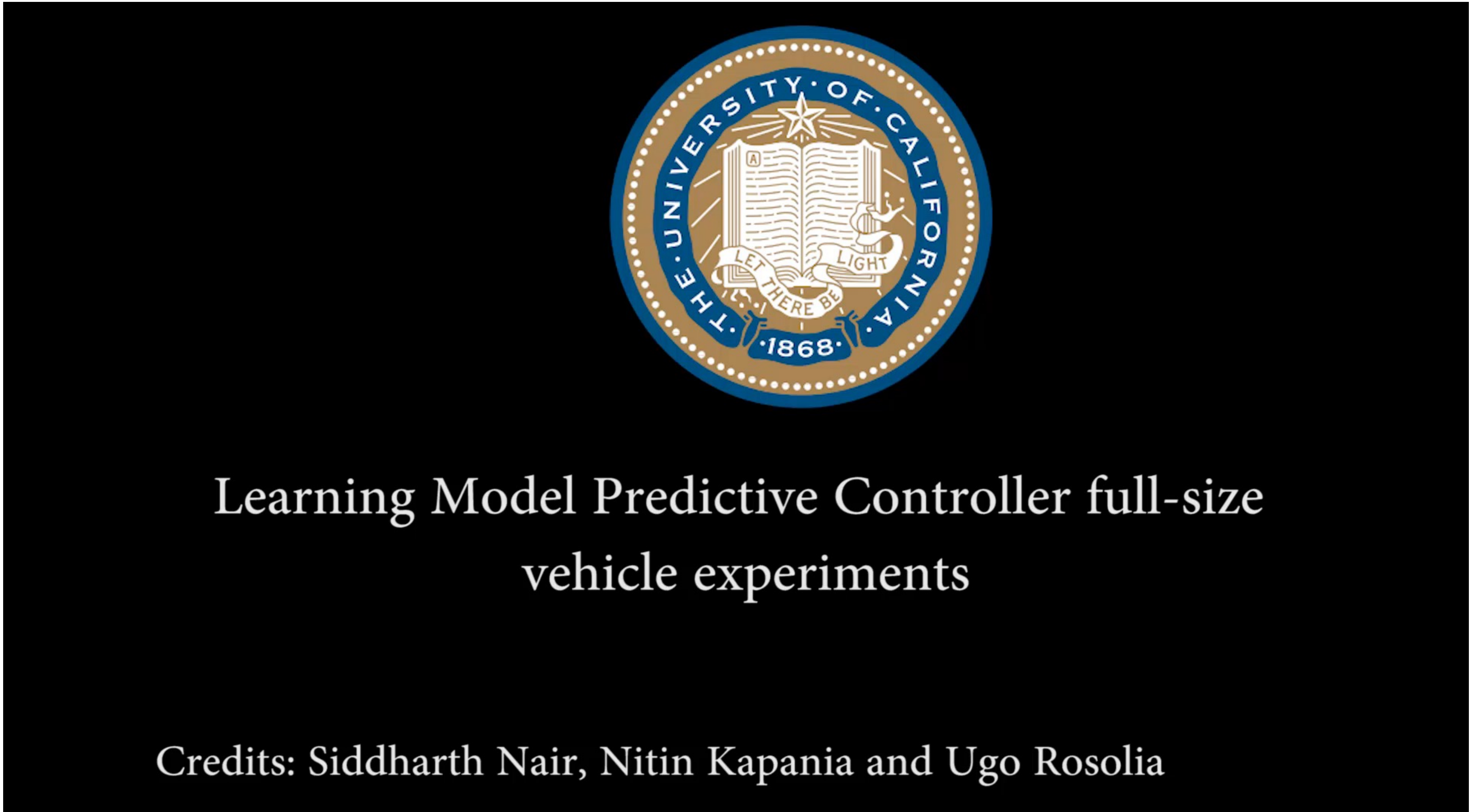
Today's goals:

- Review standard model-based and model-free RL strategies
- Design efficient model-based RL framework
- Summary of the challenges ahead of us

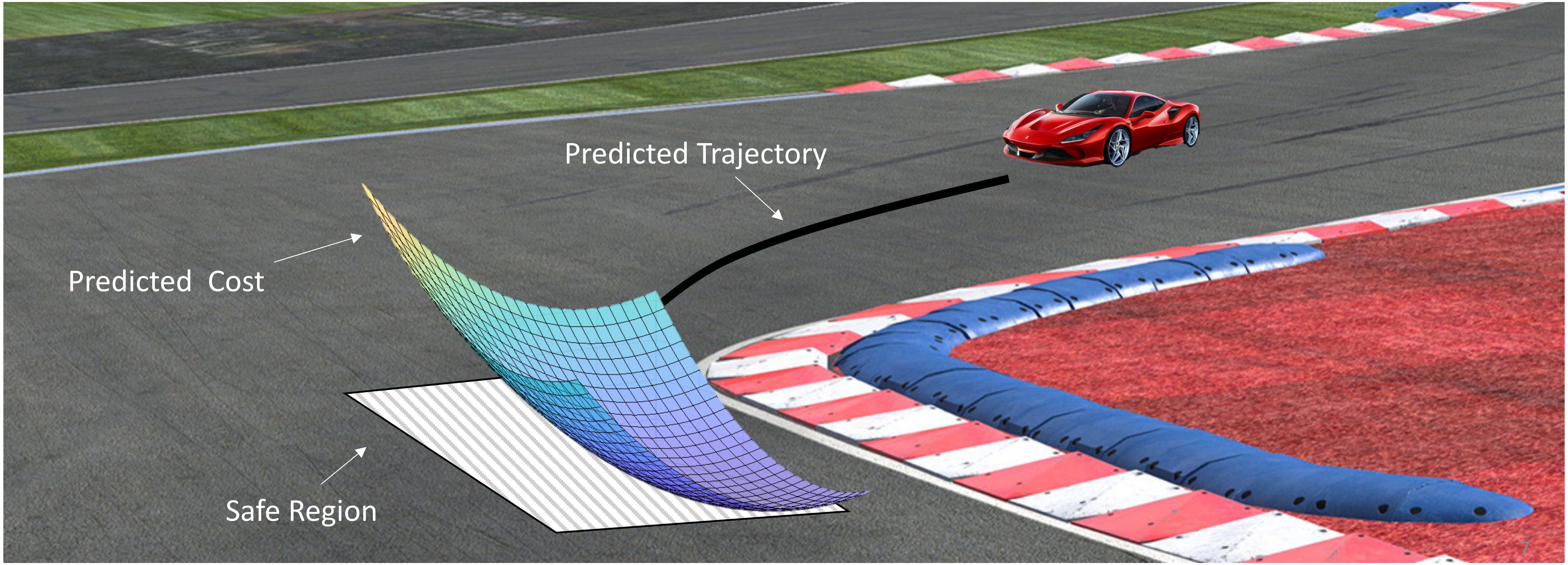


M. Janner, J. Fu, M. Zhang, and S. Levine. "When to trust your model: Model-based policy optimization." arXiv preprint arXiv:1906.08253 (2019).

Today's Example

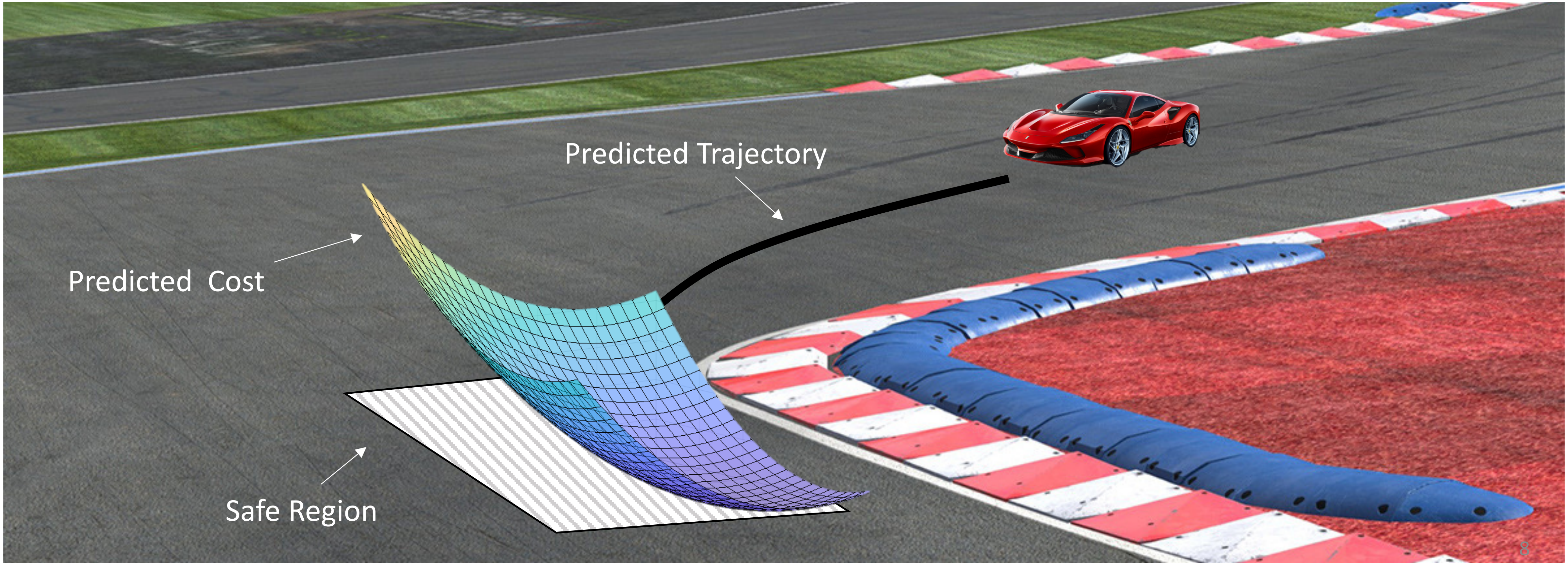


Lessons from Model Predictive Control (MPC)



- Predicted trajectory given by Prediction Model
- Safe region estimated by the Safe Set
- Predicted cost estimated by Value Function

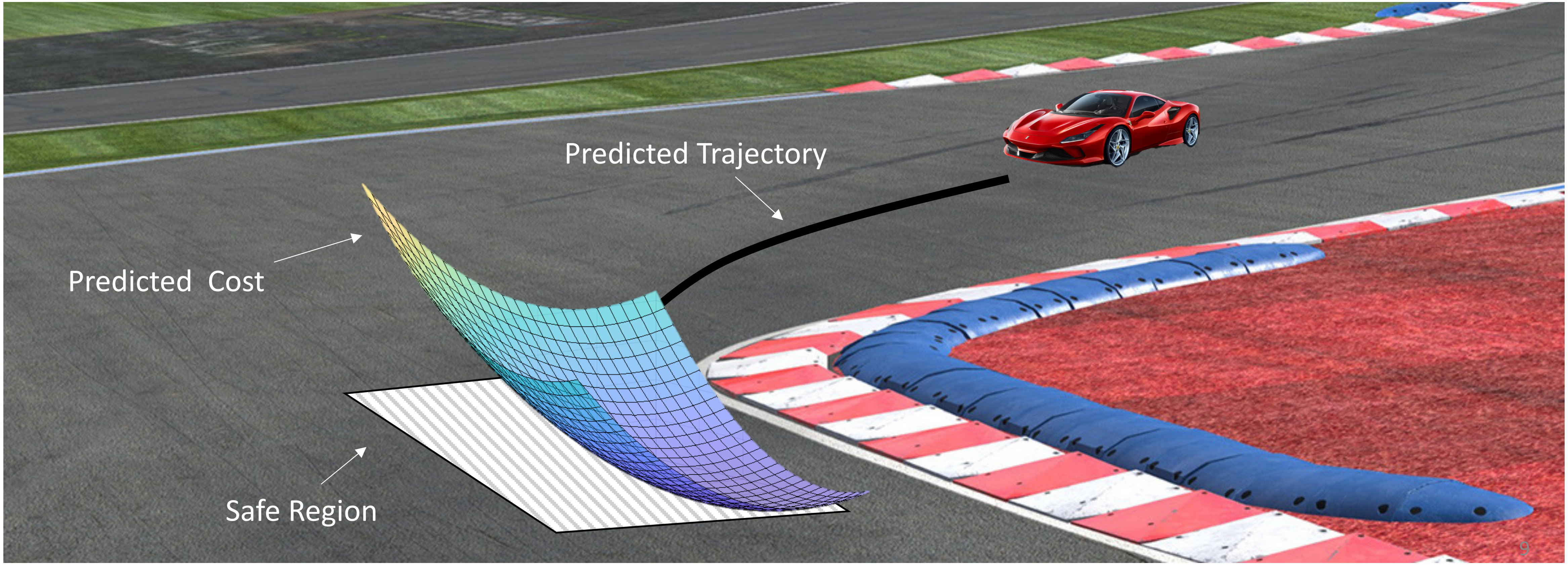
Lessons from Model Predictive Control (MPC)



- Predicted trajectory given by Prediction Model
- Safe region estimated by the Safe Set
- Predicted cost estimated by Value Function

Identified from historical data

Lessons from Model Predictive Control (MPC)



- Predicted trajectory given by Prediction Model
- Safe region estimated by the Safe Set
- Predicted cost estimated by Value Function

Identified from historical data

Estimate these components to simplify the design

Three key components to learn

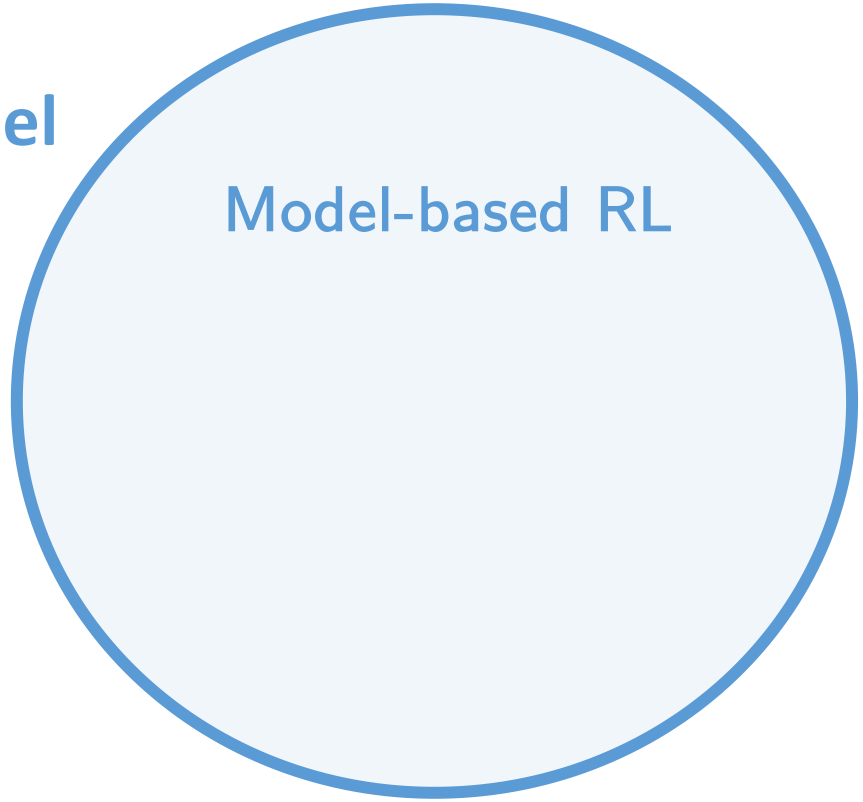
Prediction Model

Value Function

Safe Set

Three key components to learn

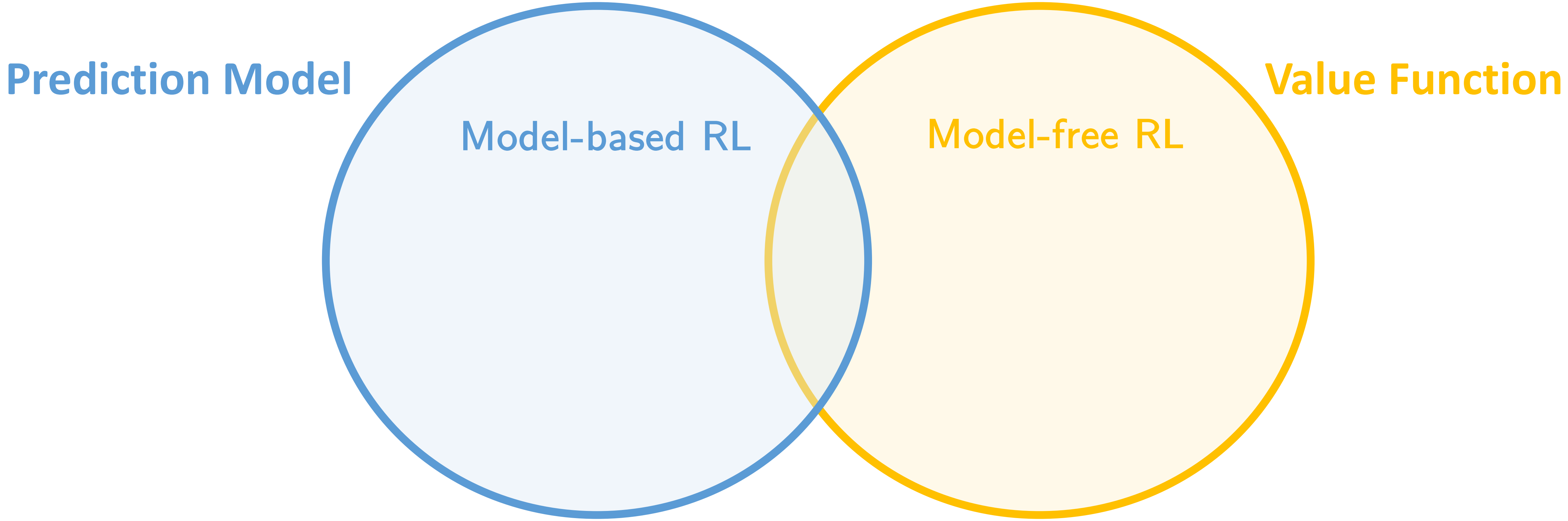
Prediction Model



Value Function

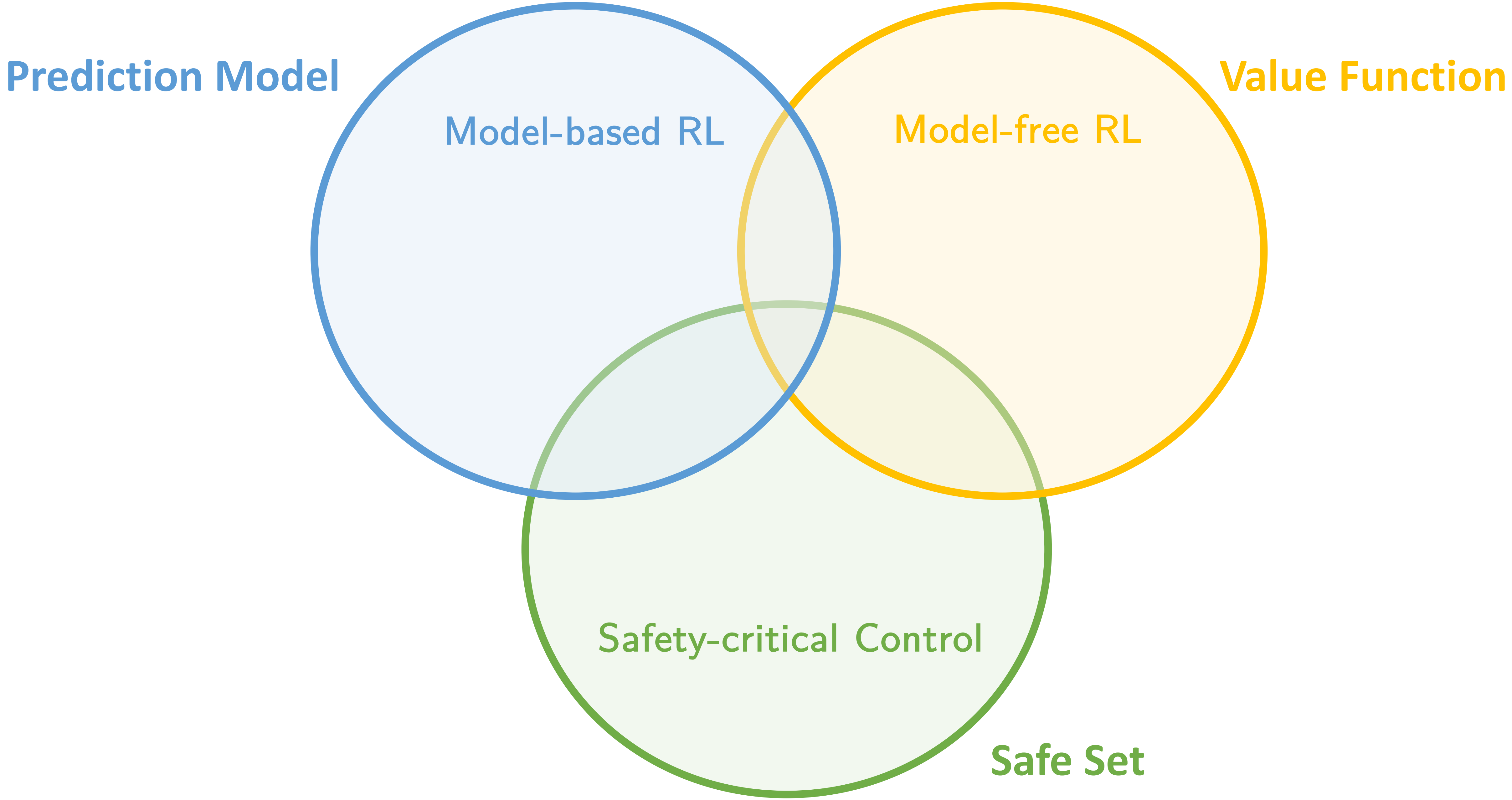
Safe Set

Three key components to learn

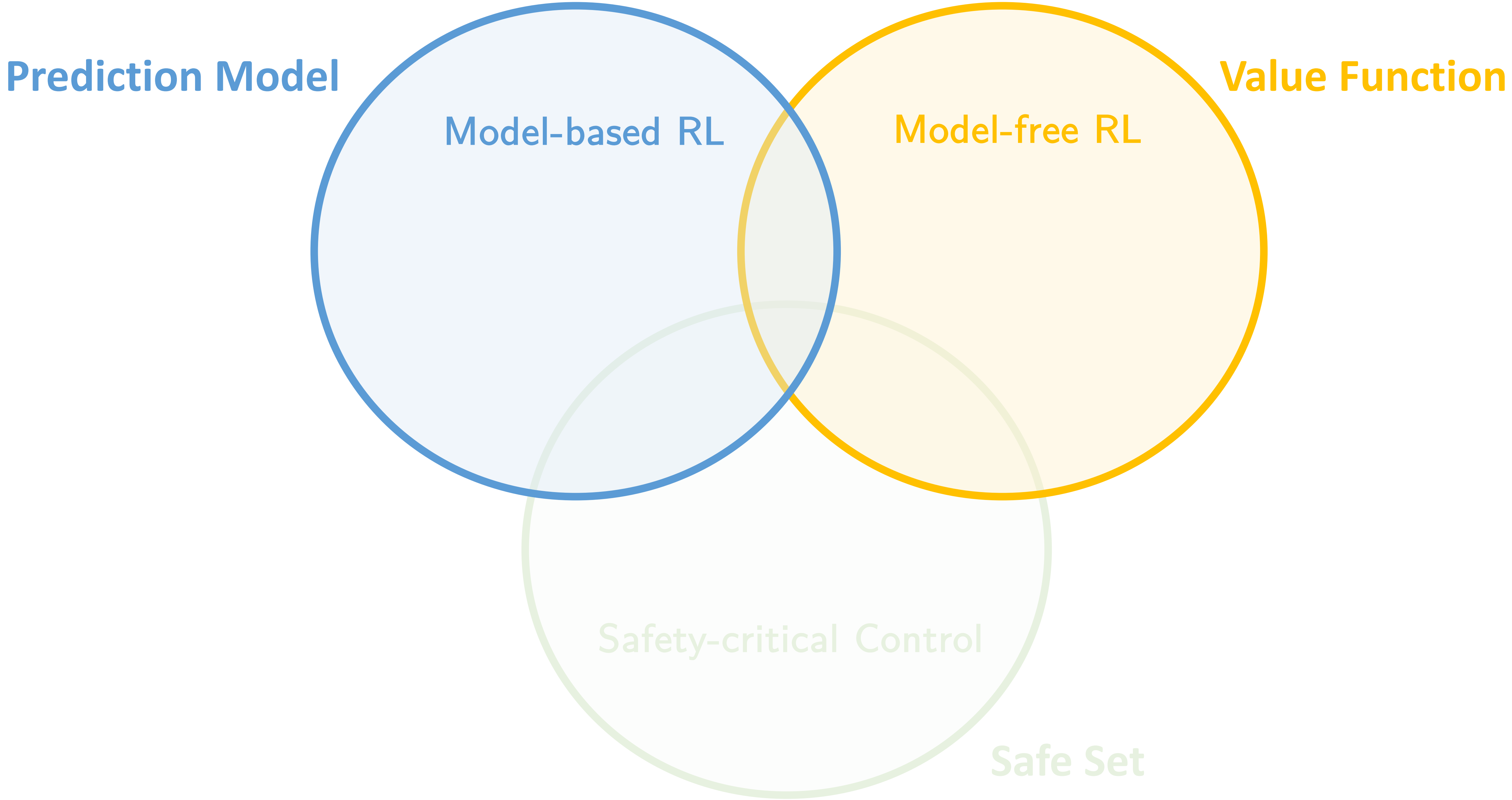


Safe Set

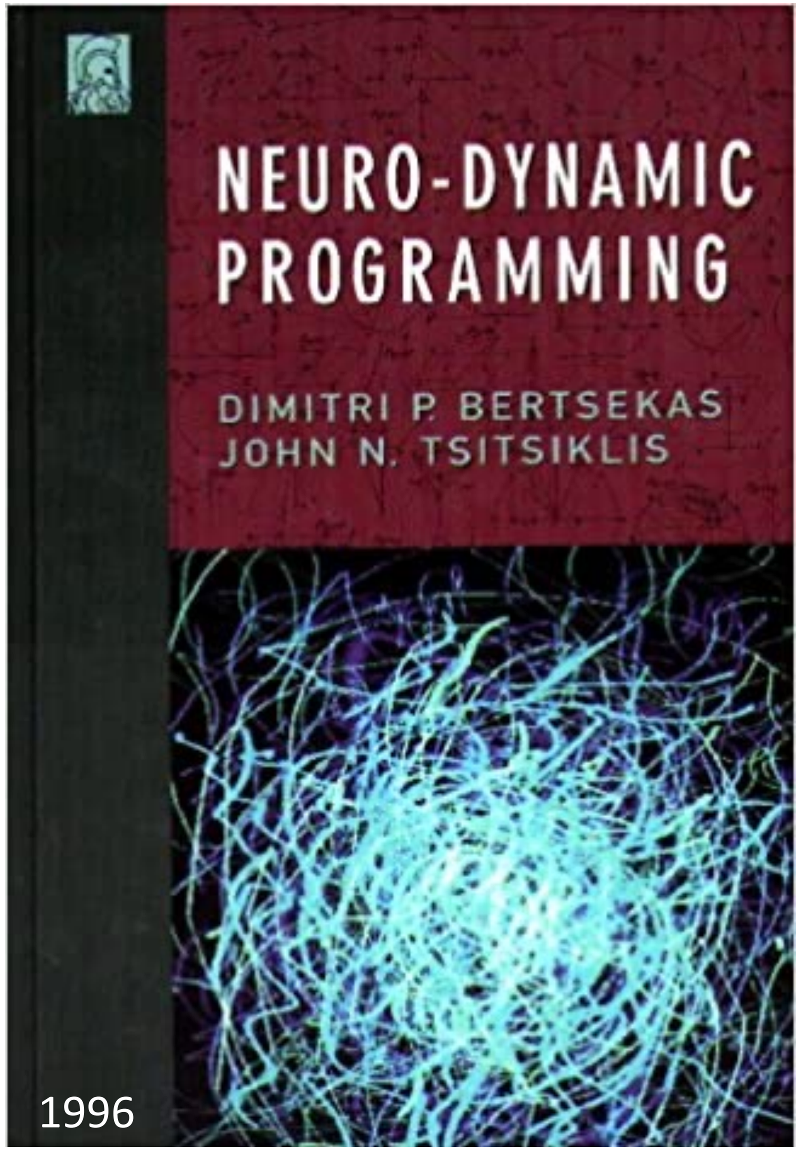
Three key components to learn

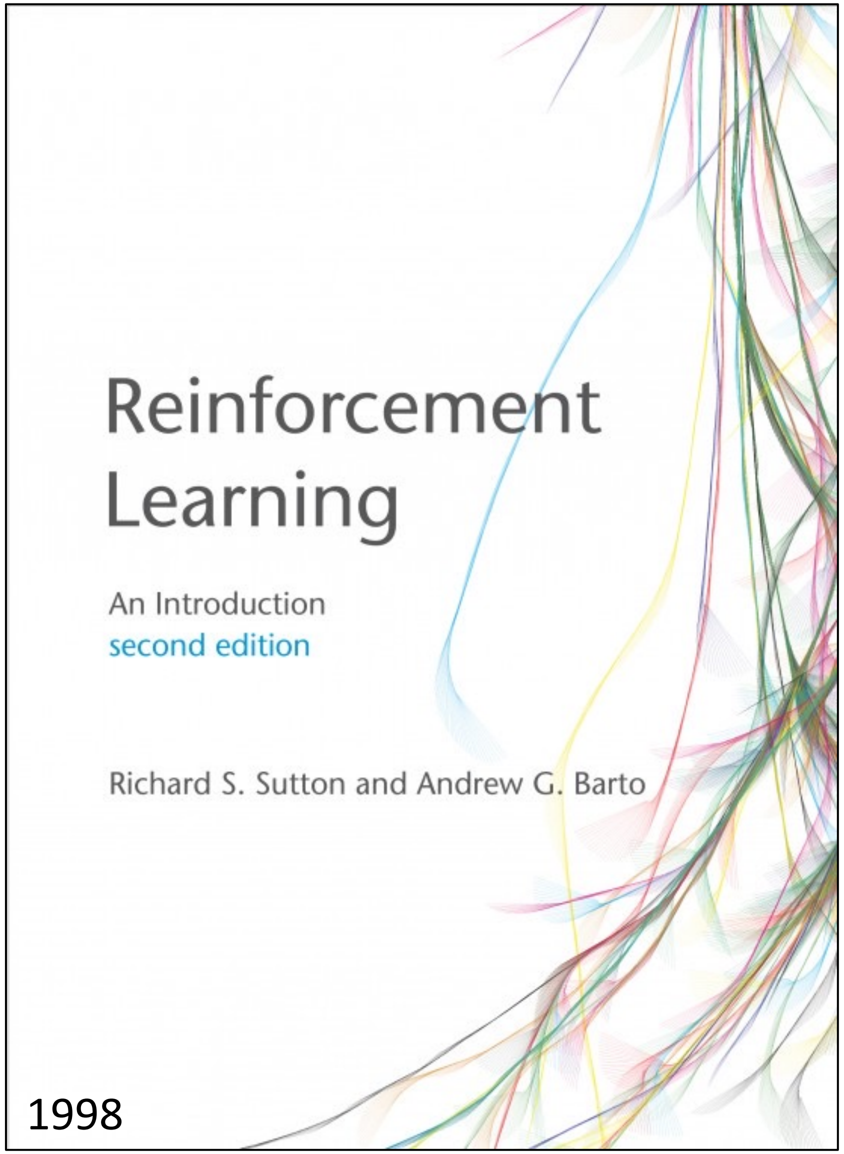


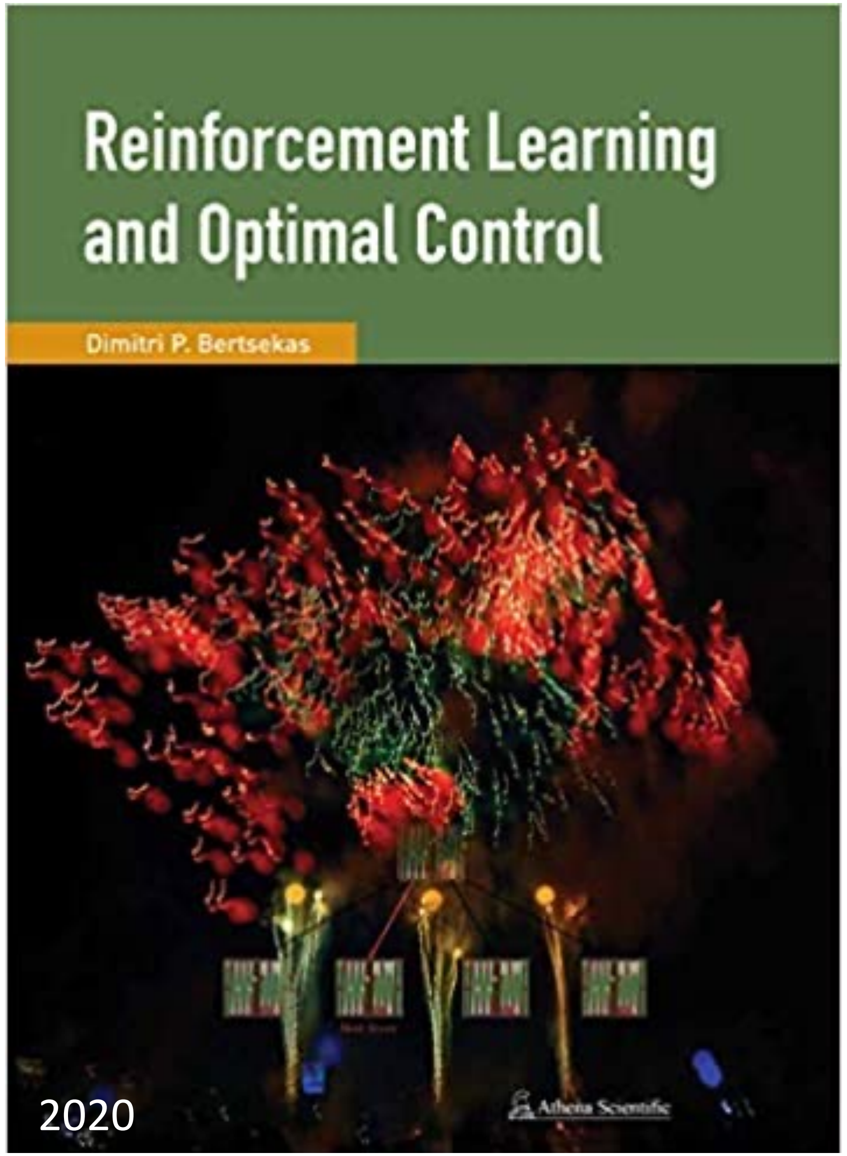
Three key components to learn



Theoretical Foundations of Reinforcement Learning







Theoretical Foundations of Reinforcement Learning

Principle of Optimality:

$$u^* = \arg\min_{u \in U} \left[h(x,u) + \mathbb{E}[V^*(x^+)|x,u] \right]$$

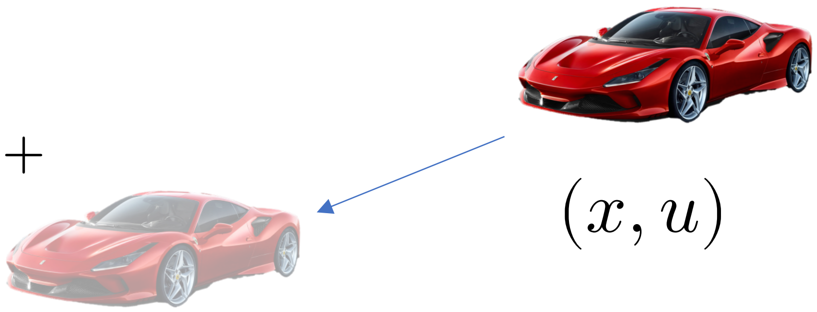
- u^*: Optimal Control Action
- $h(x,u)$: Instantaneous cost
- V^*: Future cost
- x^+: Future state

Value function to compute future cost

Map from all possible board configurations to the cost!

+

Prediction Model to compute x^+



Deep Reinforcement Learning

Principle of Optimality:

$$u^* = \arg\min_{u \in U} \left[h(x,u) + \mathbb{E}[V^*(x^+)|x,u] \right]$$

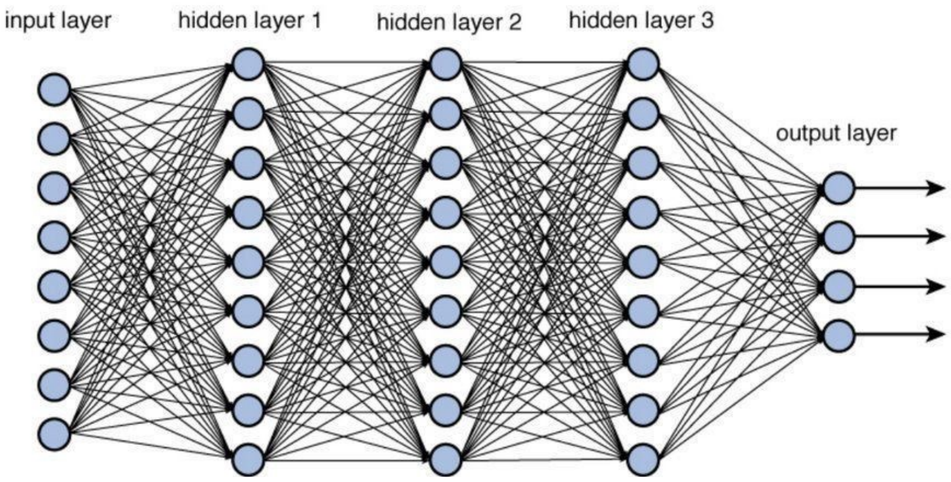
Optimal Control Action

Instantaneous cost

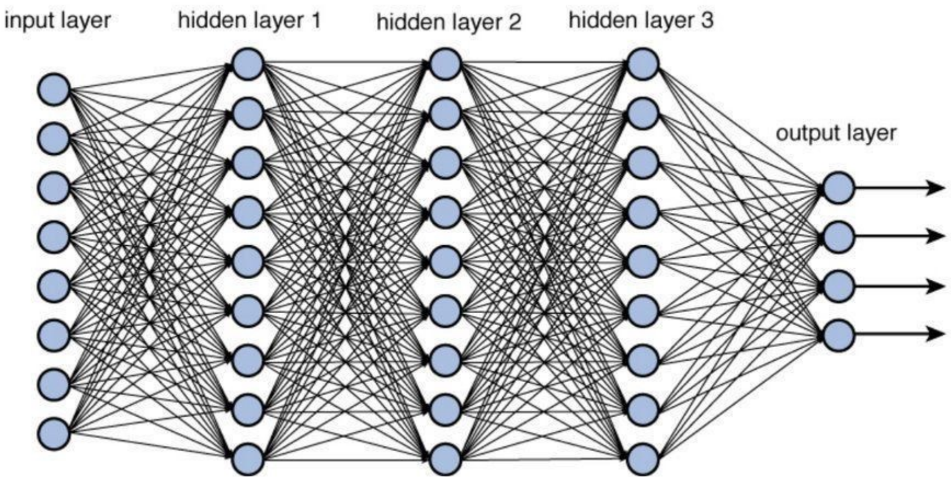
Future cost

Future state

Value function to compute future cost



Prediction Model to compute x^+



Model-Based vs Model-Free

Principle of Optimality:

$$u^* = \arg\min_{u \in U} \left[h(x,u) + \mathbb{E}[V^*(x^+) | x, u] \right]$$

Optimal Control Action

Instantaneous cost

Future cost

Future state

Model-Based vs Model-Free

Principle of Optimality:

$$u^* = \arg\min_{u \in U} \left[h(x,u) + \mathbb{E}[V^*(x^+) | x, u] \right]$$

Optimal Control Action

Instantaneous cost

Future cost

Future state

Model-Based RL

$$[u^*, u_1, \ldots, u_{N-1}] = \arg\min_{u_k \in U} \mathbb{E}\Big[\sum_{k=0}^{N-1} h(x_k, u_k) + V^*(x_N) \Big]$$

Model-Based vs Model-Free

Principle of Optimality:

$$u^* = \arg\min_{u \in U} \left[h(x,u) + \mathbb{E}[V^*(x^+)|x,u] \right]$$

Optimal Control Action

Instantaneous cost

Future cost

Future state

Model-Based RL

$$[u^*, u_1, \ldots, u_{N-1}] = \arg\min_{u_k \in U} \mathbb{E}\Big[\sum_{k=0}^{N-1} h(x_k, u_k) + V^*(x_N) \Big]$$

The above, for long horizon N is approximated as

$$[u^*, u_1, \ldots, u_{N-1}] = \arg\min_{u_k \in U} \mathbb{E}\Big[\sum_{k=0}^{N-1} h(x_k, u_k) \Big]$$

Model-Based vs Model-Free

Principle of Optimality:

$$u^* = \arg\min_{u \in U} \left[h(x,u) + \mathbb{E}[V^*(x^+)|x,u] \right]$$

Optimal Control Action

Instantaneous cost

Future cost

Future state

Model-Based RL

$$[u^*, u_1, \ldots, u_{N-1}] = \arg\min_{u_k \in U} \mathbb{E}\Big[\sum_{k=0}^{N-1} h(x_k, u_k) + V^*(x_N)\Big]$$

The above, for long horizon N is approximated as

$$[u^*, u_1, \ldots, u_{N-1}] = \arg\min_{u_k \in U} \mathbb{E}\Big[\sum_{k=0}^{N-1} h(x_k, u_k)\Big]$$

Model-Free RL

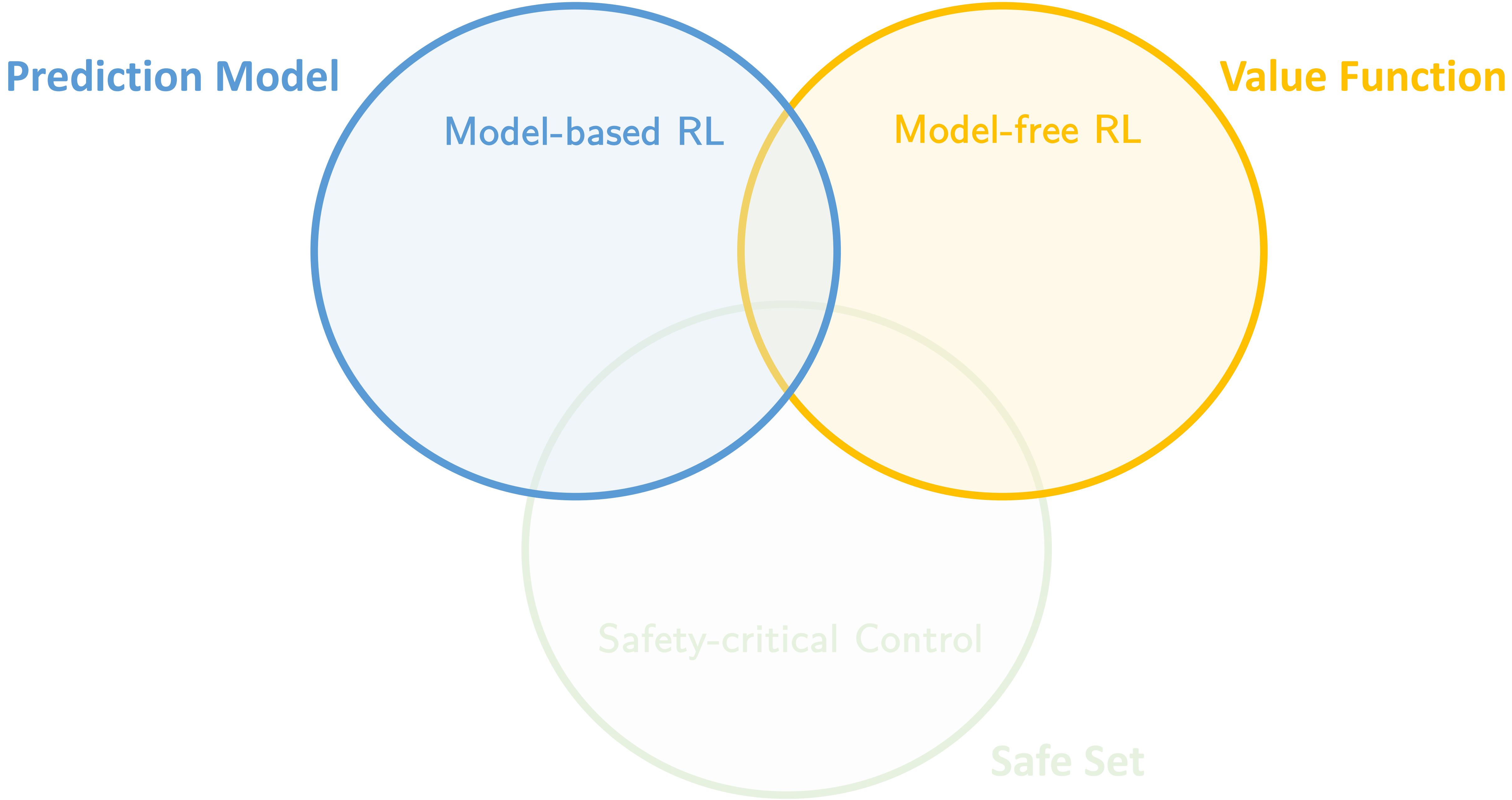
Define the Q-factor:

$$Q^*(x,u) = h(x,u) + \mathbb{E}[V^*(x^+)|x,u]$$

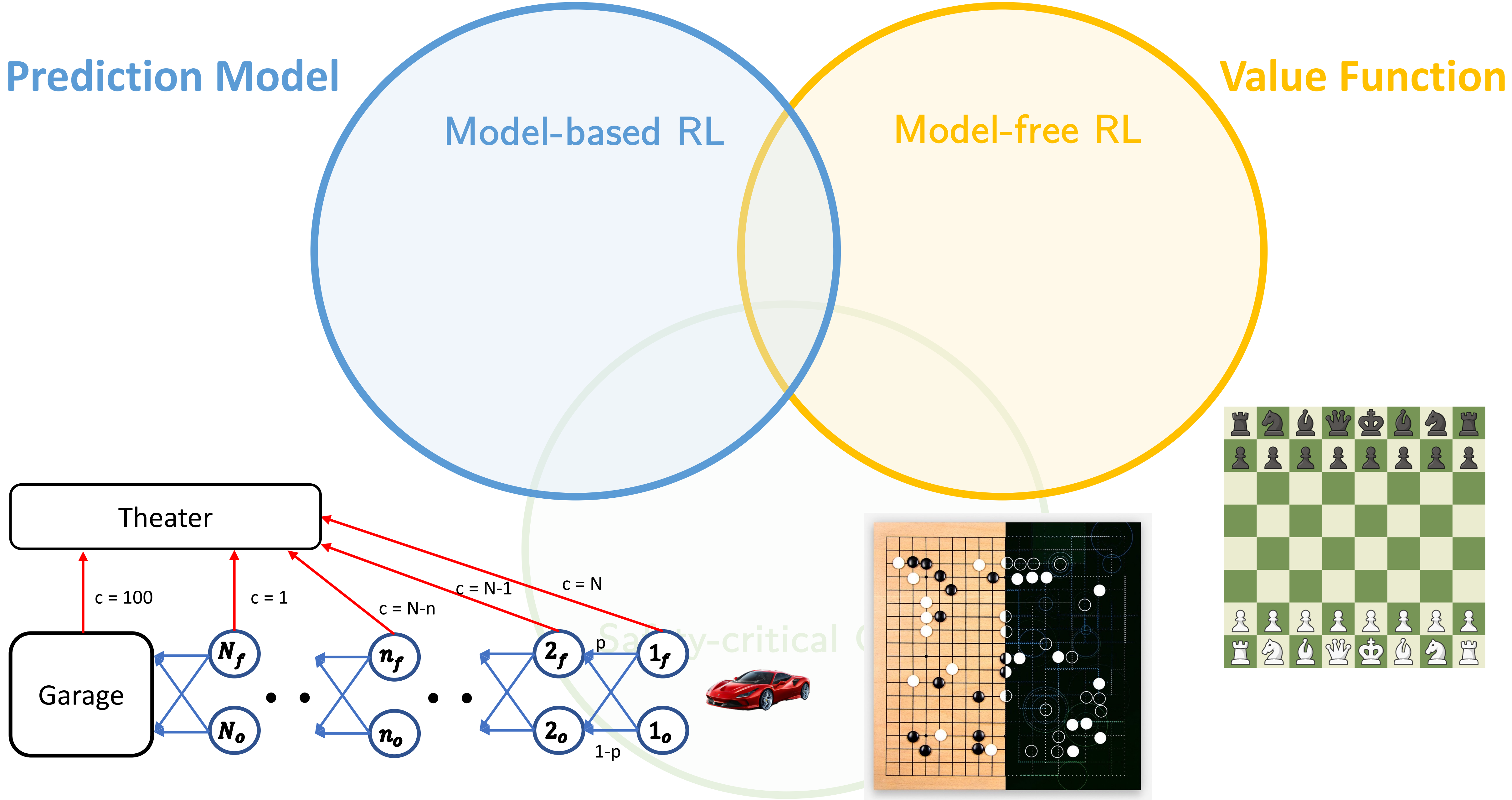
Then the optimal action is

$$u^* = \arg\min_{u \in \mathcal{U}} Q^*(x,u)$$

Three key components to learn



Three key components to learn



Three key components to learn

Prediction Model

Value Function

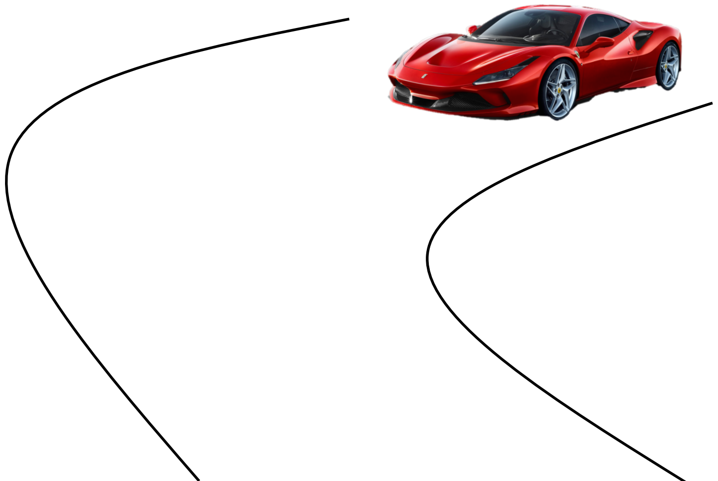
Model-based RL

Model-free RL

$$\min_{\{u_k\}_{k=0}^{N}} \mathbb{E}\left[\sum_{t=0}^{N} h(x_t, u_t)\right] + \min_{u \in \mathcal{U}} Q^*(x, u)$$

Safety-critical Control

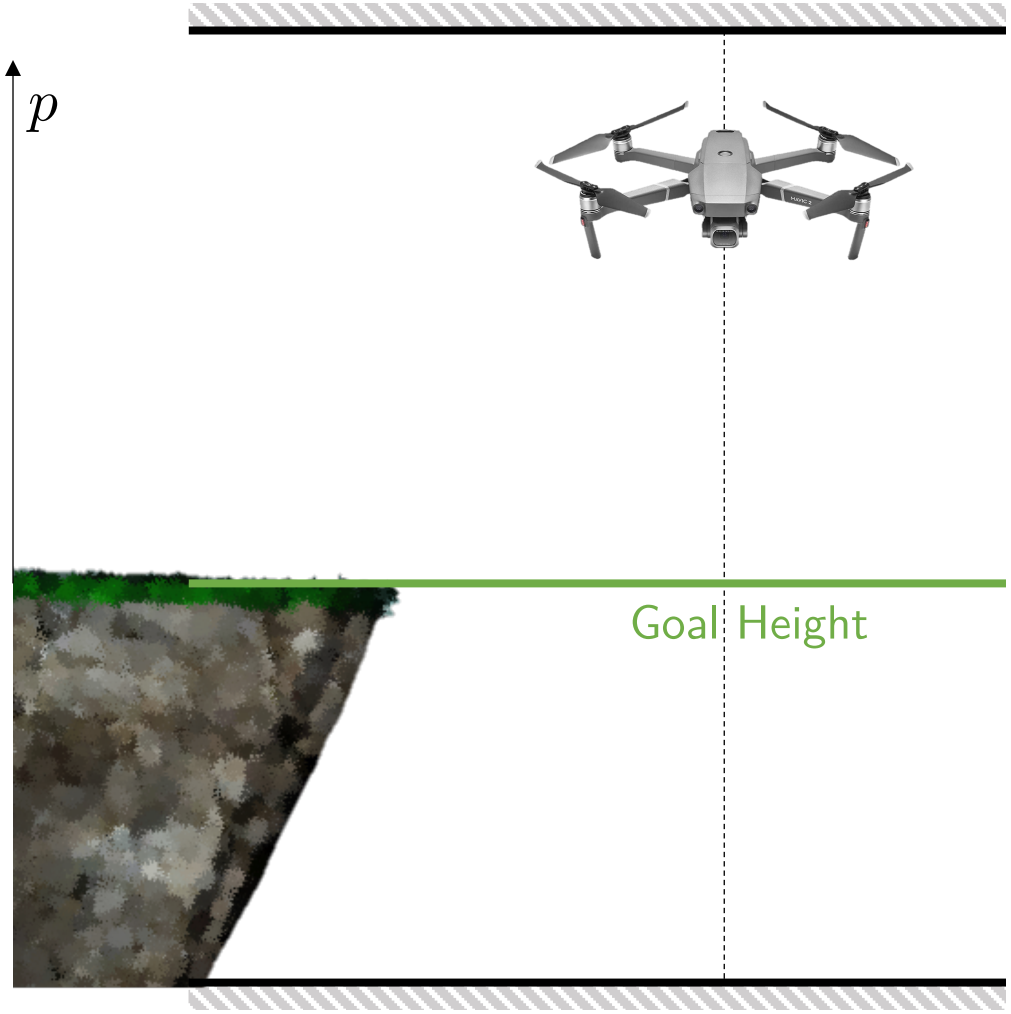
Safe Set



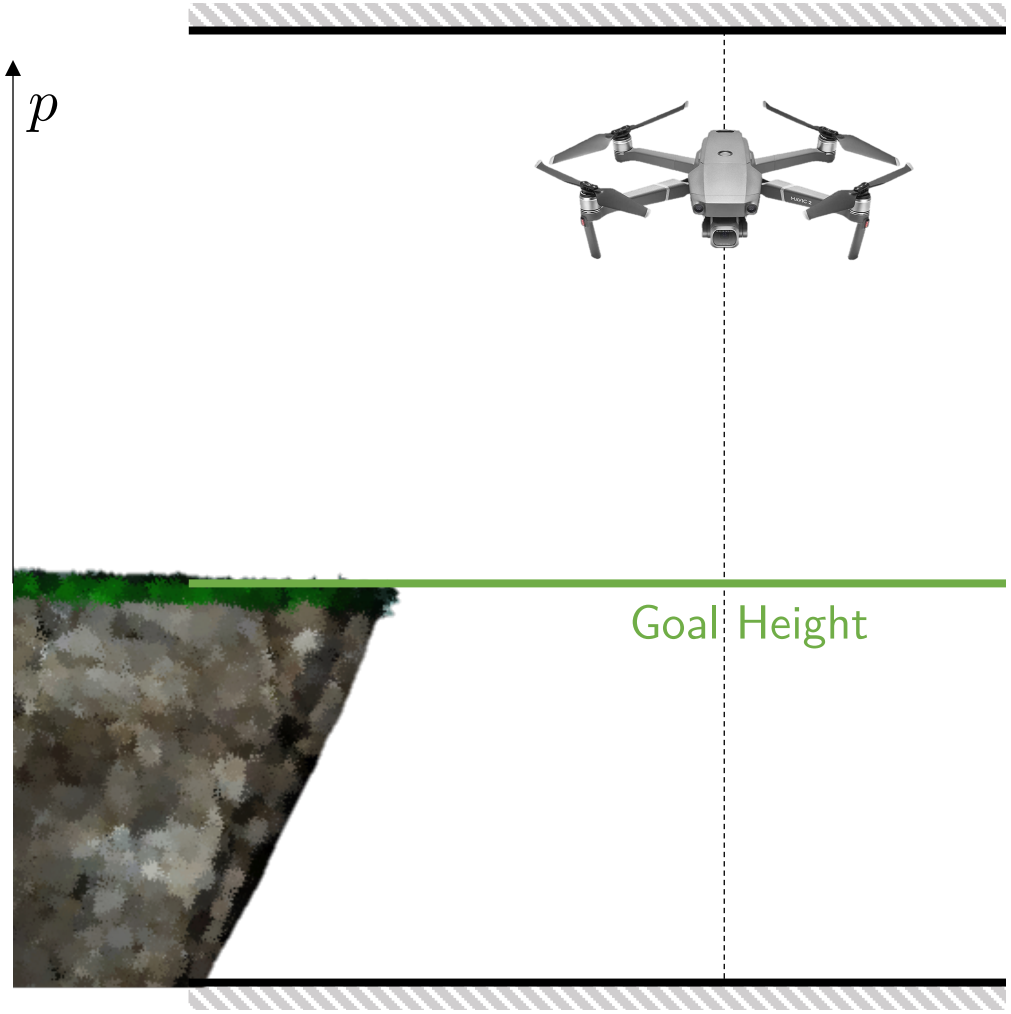
What is different in safety-critical systems?

What is different in safety-critical systems? Constraints

What is different in safety-critical systems? Constraints



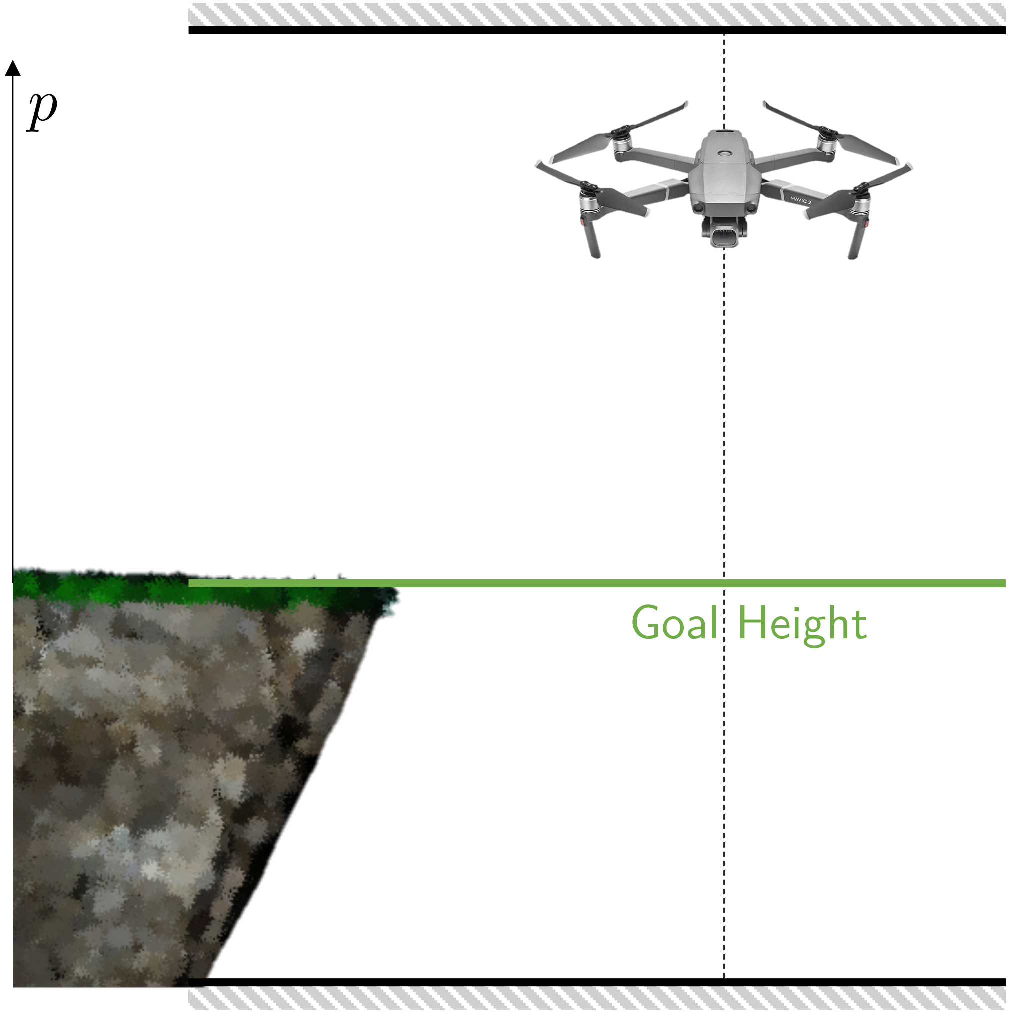
What is different in safety-critical systems? Constraints



- State

$$x = \begin{bmatrix} p \\ v \end{bmatrix} = \begin{bmatrix} \text{position} \\ \text{velocity} \end{bmatrix}$$

What is different in safety-critical systems? Constraints

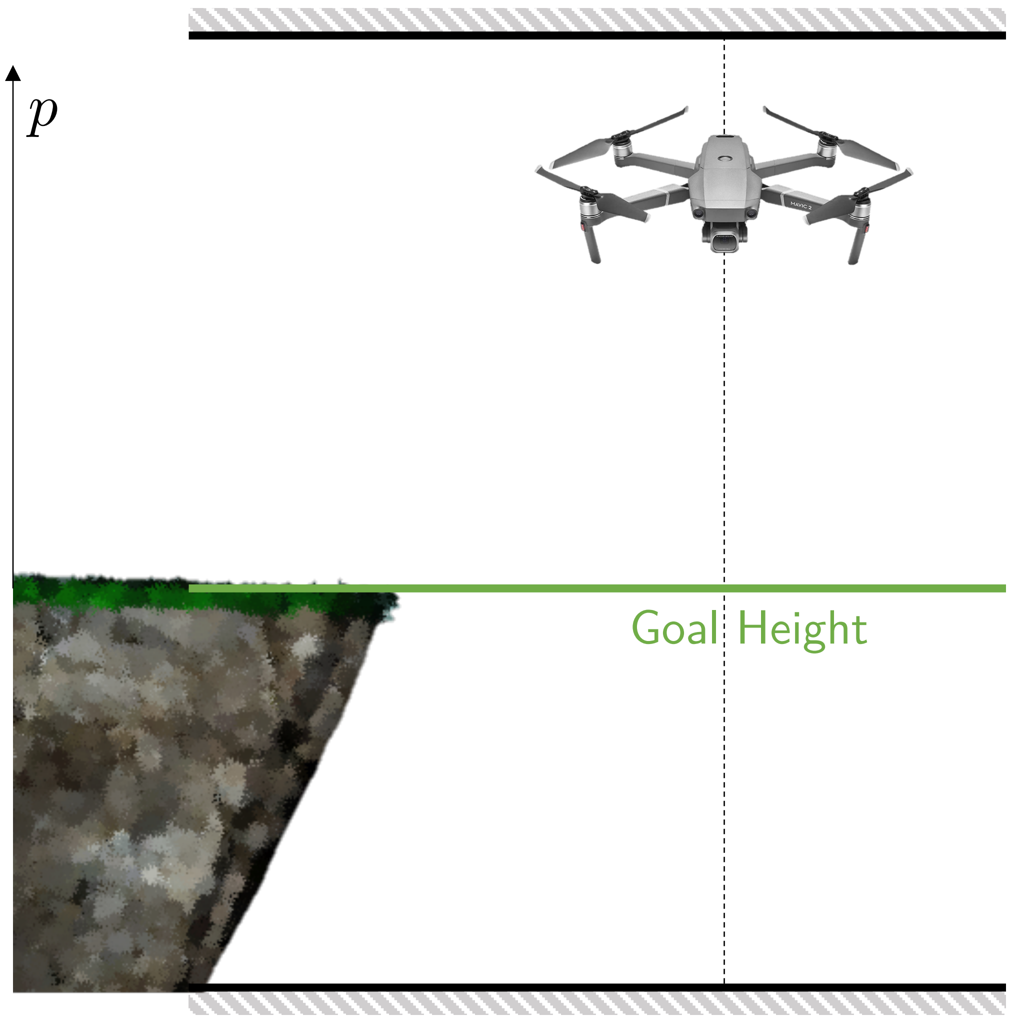


- State

$$x = \begin{bmatrix} p \\ v \end{bmatrix} = \begin{bmatrix} \text{position} \\ \text{velocity} \end{bmatrix}$$

- Input $u = a =$ acceleration

What is different in safety-critical systems? Constraints



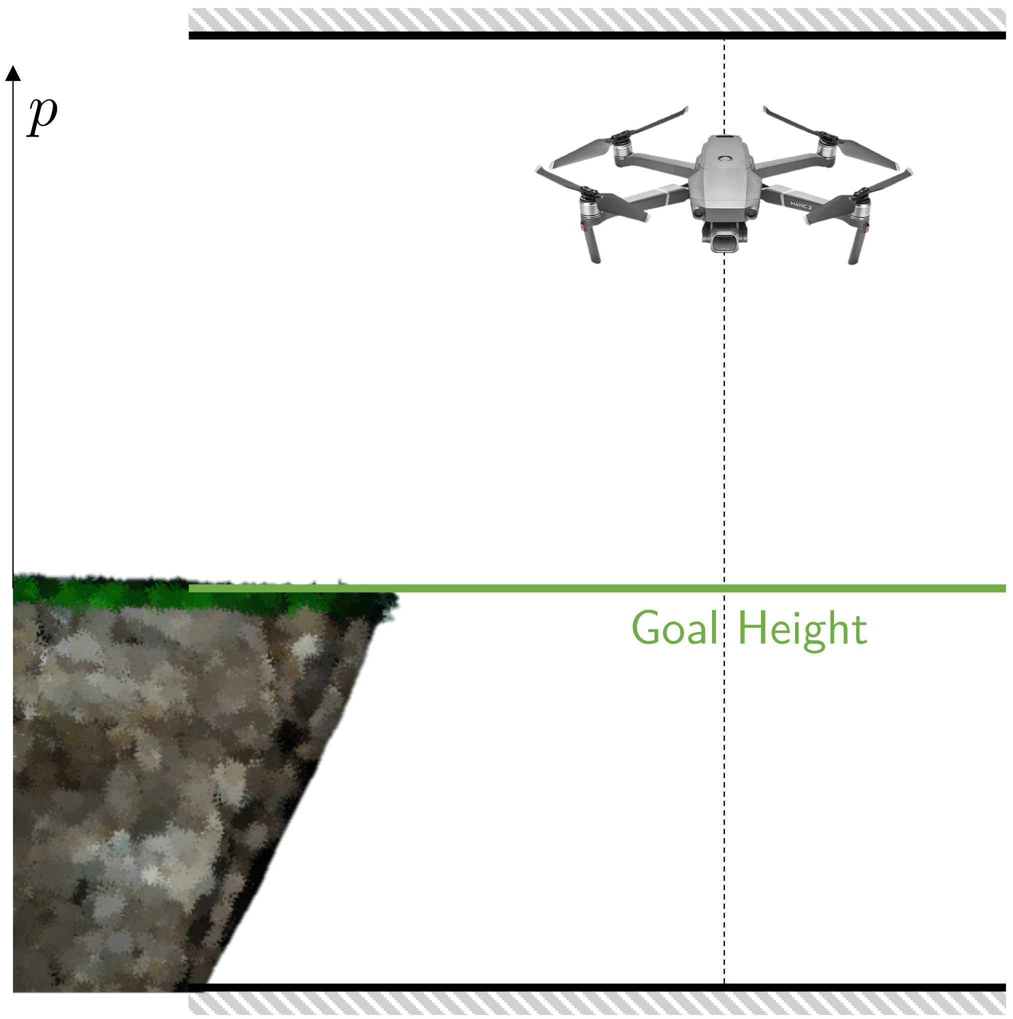
- State

$$x = \begin{bmatrix} p \\ v \end{bmatrix} = \begin{bmatrix} \text{position} \\ \text{velocity} \end{bmatrix}$$

- Input $u = a =$ acceleration
- Dynamics

$$\begin{bmatrix} p_{k+1} \\ v_{k+1} \end{bmatrix} = \begin{bmatrix} 1 & dt \\ 0 & 1 \end{bmatrix} \begin{bmatrix} p_k \\ v_k \end{bmatrix} + \begin{bmatrix} 0 \\ a_k \end{bmatrix}$$

What is different in safety-critical systems? Constraints



- State

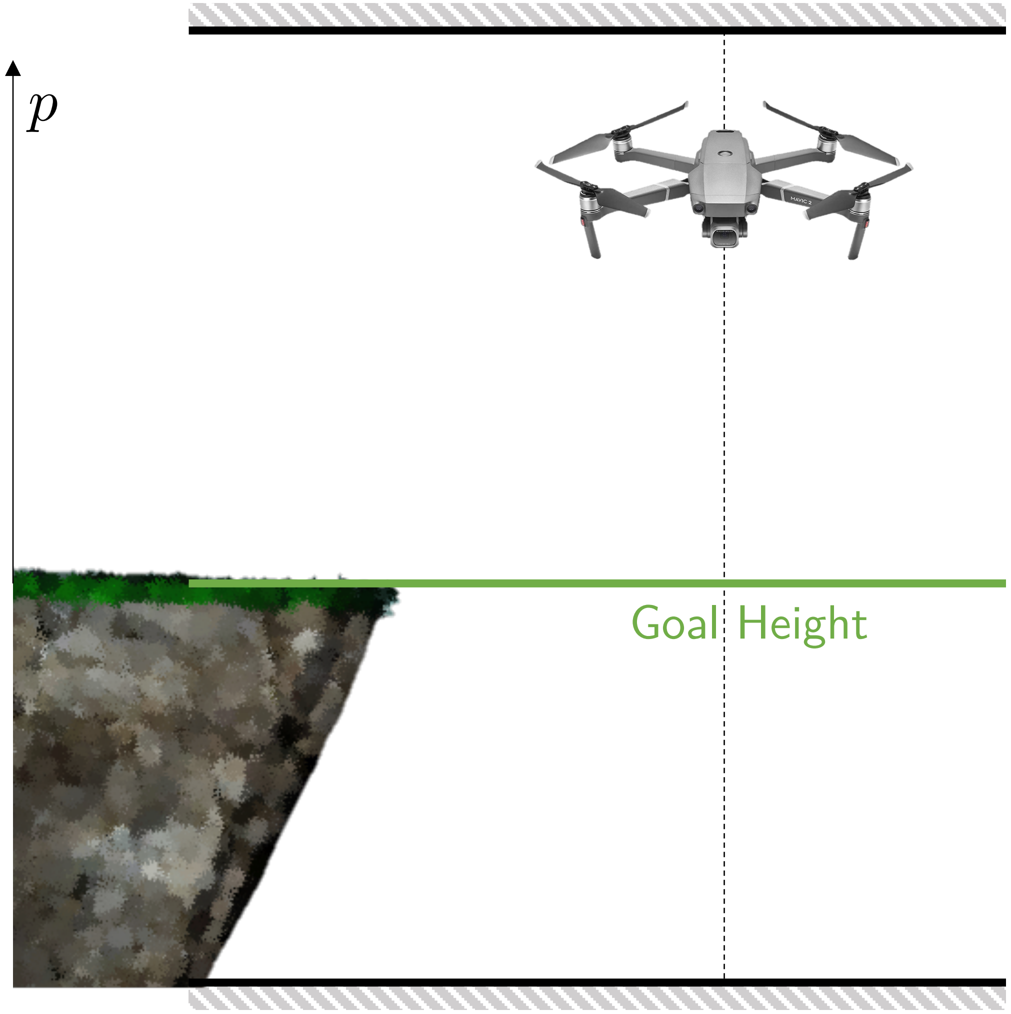
$$x = \begin{bmatrix} p \\ v \end{bmatrix} = \begin{bmatrix} \text{position} \\ \text{velocity} \end{bmatrix}$$

- Input $u = a =$ acceleration
- Dynamics

$$\begin{bmatrix} p_{k+1} \\ v_{k+1} \end{bmatrix} = \begin{bmatrix} 1 & dt \\ 0 & 1 \end{bmatrix} \begin{bmatrix} p_k \\ v_k \end{bmatrix} + \begin{bmatrix} 0 \\ a_k \end{bmatrix}$$

- Cost $x_k^\top Q x_k + u_k^\top R u_k$

What is different in safety-critical systems? Constraints



- State

$$x = \begin{bmatrix} p \\ v \end{bmatrix} = \begin{bmatrix} \text{position} \\ \text{velocity} \end{bmatrix}$$

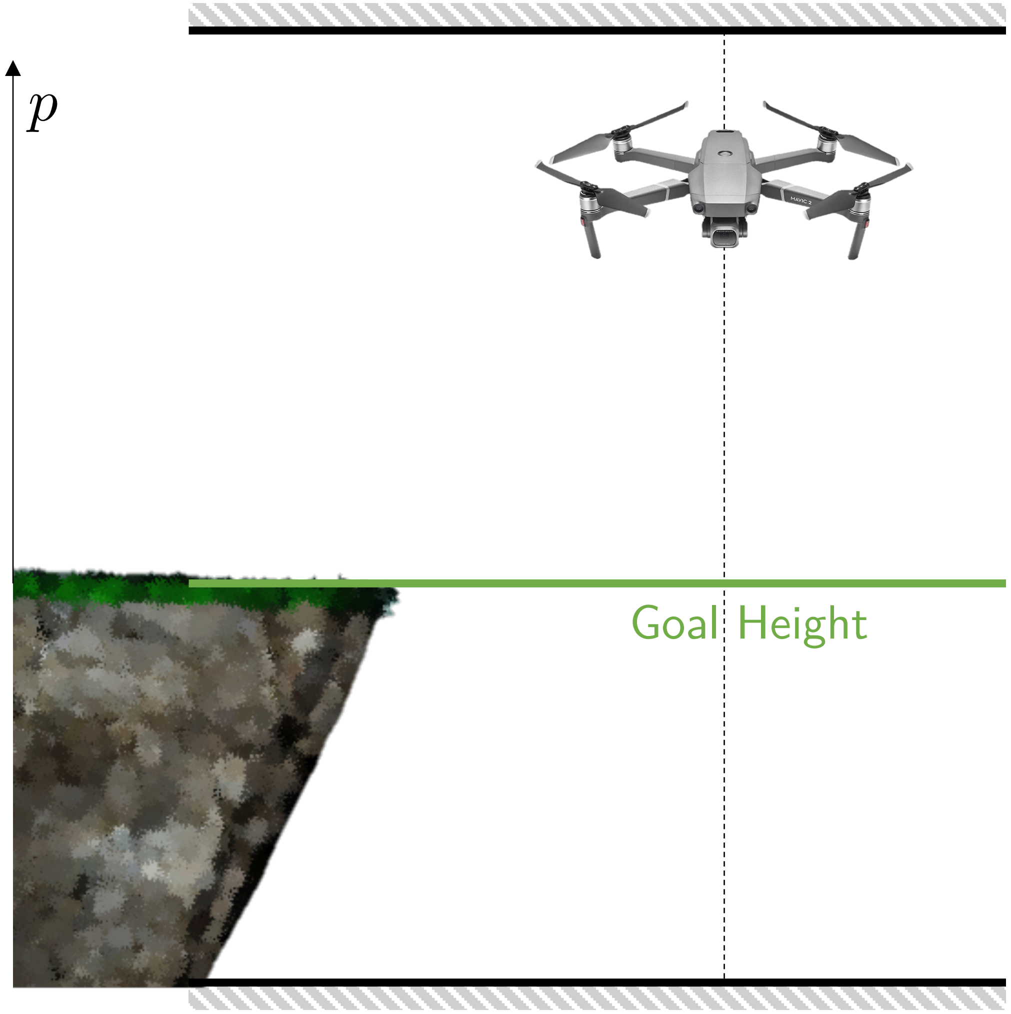
- Input $u = a =$ acceleration
- Dynamics

$$\begin{bmatrix} p_{k+1} \\ v_{k+1} \end{bmatrix} = \begin{bmatrix} 1 & dt \\ 0 & 1 \end{bmatrix} \begin{bmatrix} p_k \\ v_k \end{bmatrix} + \begin{bmatrix} 0 \\ a_k \end{bmatrix}$$

- Cost $x_k^\top Q x_k + u_k^\top R u_k$
- Constraints

$$\begin{bmatrix} -5 \\ -5 \\ -0.5 \end{bmatrix} \leq \begin{bmatrix} p_k \\ v_k \\ a_k \end{bmatrix} \leq \begin{bmatrix} 5 \\ 5 \\ 0.5 \end{bmatrix}$$

What is different in safety-critical systems? Constraints



- State

$$x = \begin{bmatrix} p \\ v \end{bmatrix} = \begin{bmatrix} \text{position} \\ \text{velocity} \end{bmatrix}$$

- Input $u = a =$ acceleration
- Dynamics

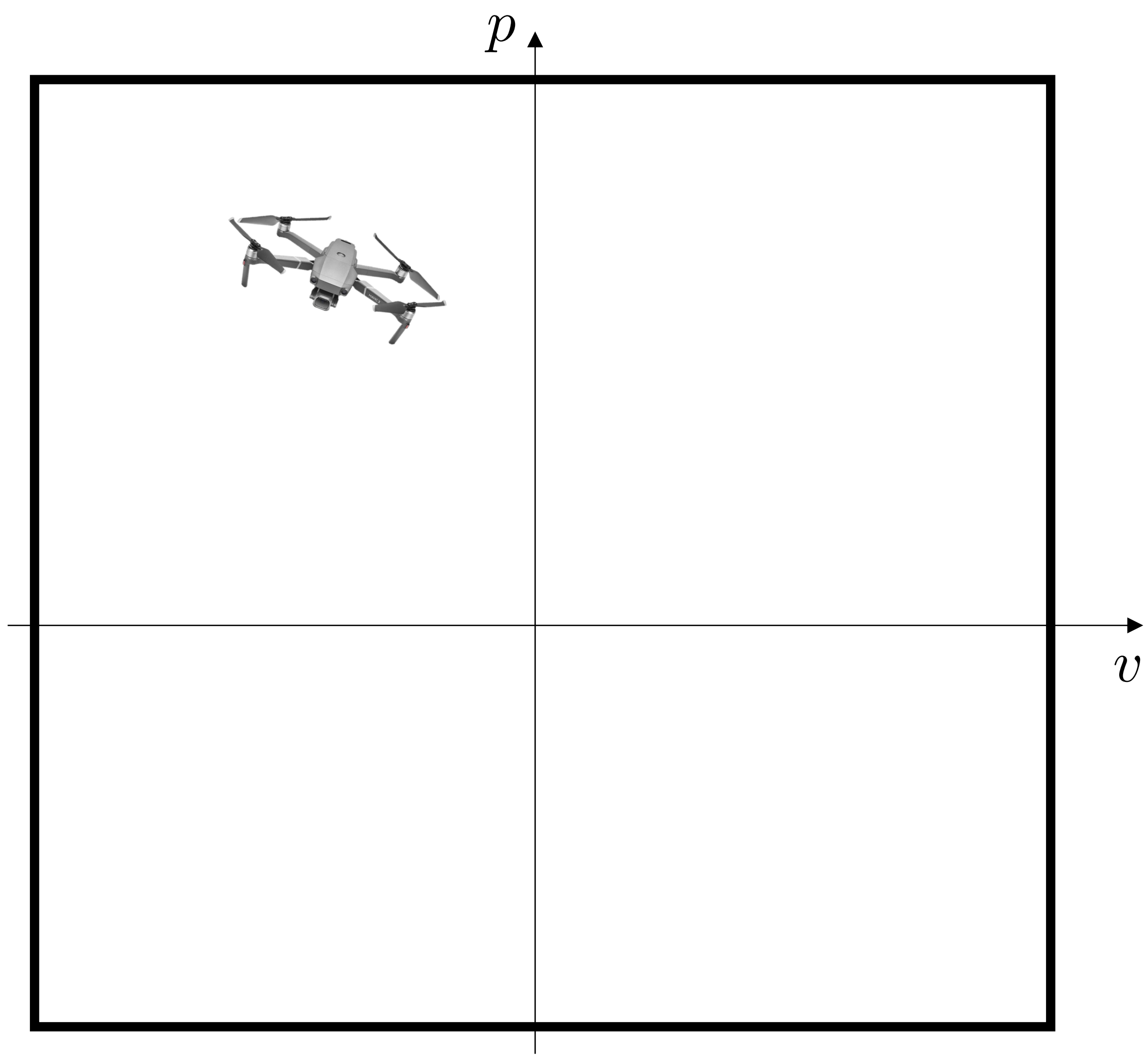
$$\begin{bmatrix} p_{k+1} \\ v_{k+1} \end{bmatrix} = \begin{bmatrix} 1 & dt \\ 0 & 1 \end{bmatrix} \begin{bmatrix} p_k \\ v_k \end{bmatrix} + \begin{bmatrix} 0 \\ a_k \end{bmatrix}$$

- Cost $x_k^\top Q x_k + u_k^\top R u_k$
- Constraints

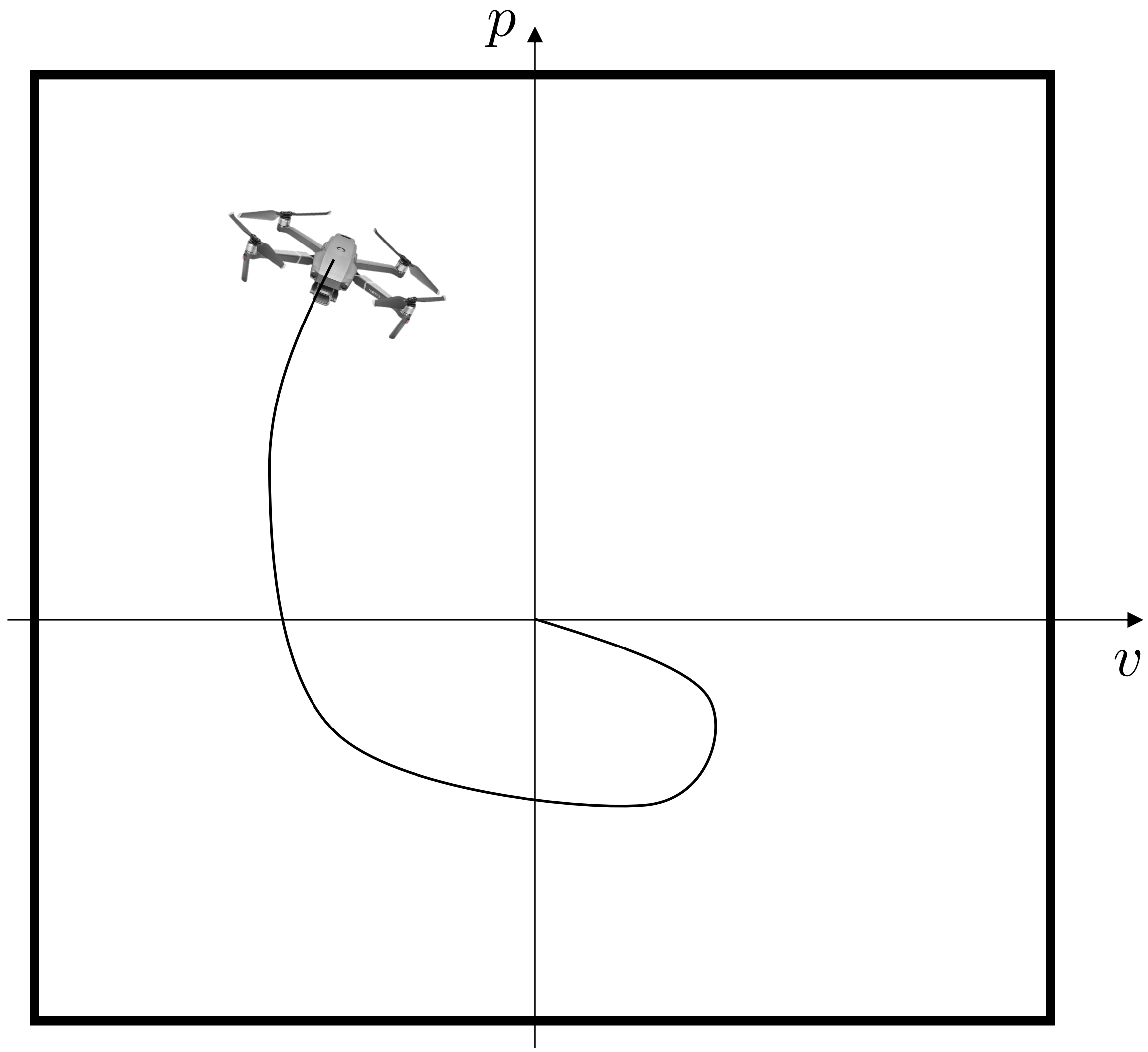
$$\begin{bmatrix} -5 \\ -5 \\ -0.5 \end{bmatrix} \leq \begin{bmatrix} p_k \\ v_k \\ a_k \end{bmatrix} \leq \begin{bmatrix} 5 \\ 5 \\ 0.5 \end{bmatrix}$$

Limited actuation!

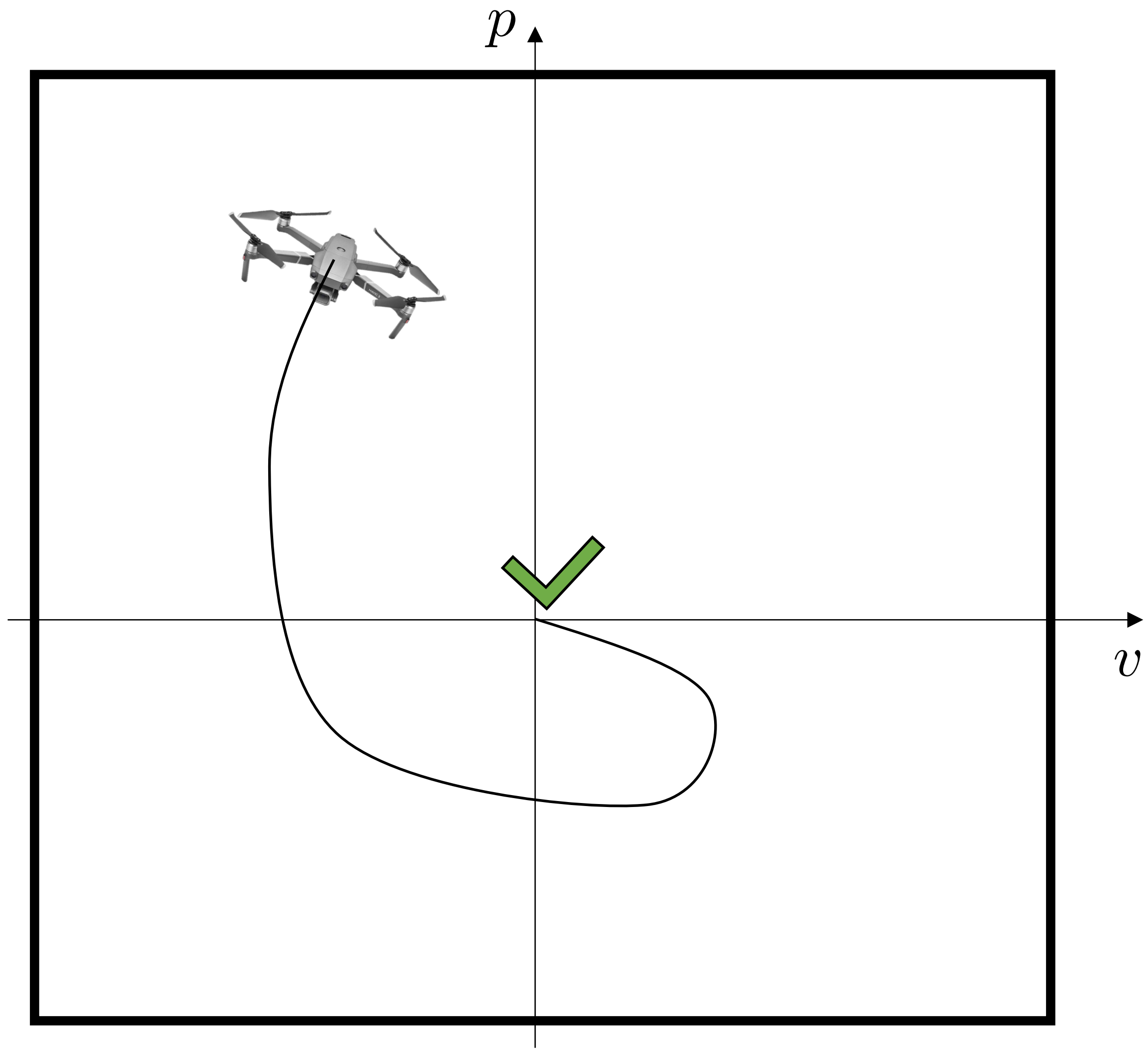
What is different in safety-critical systems? Constraints



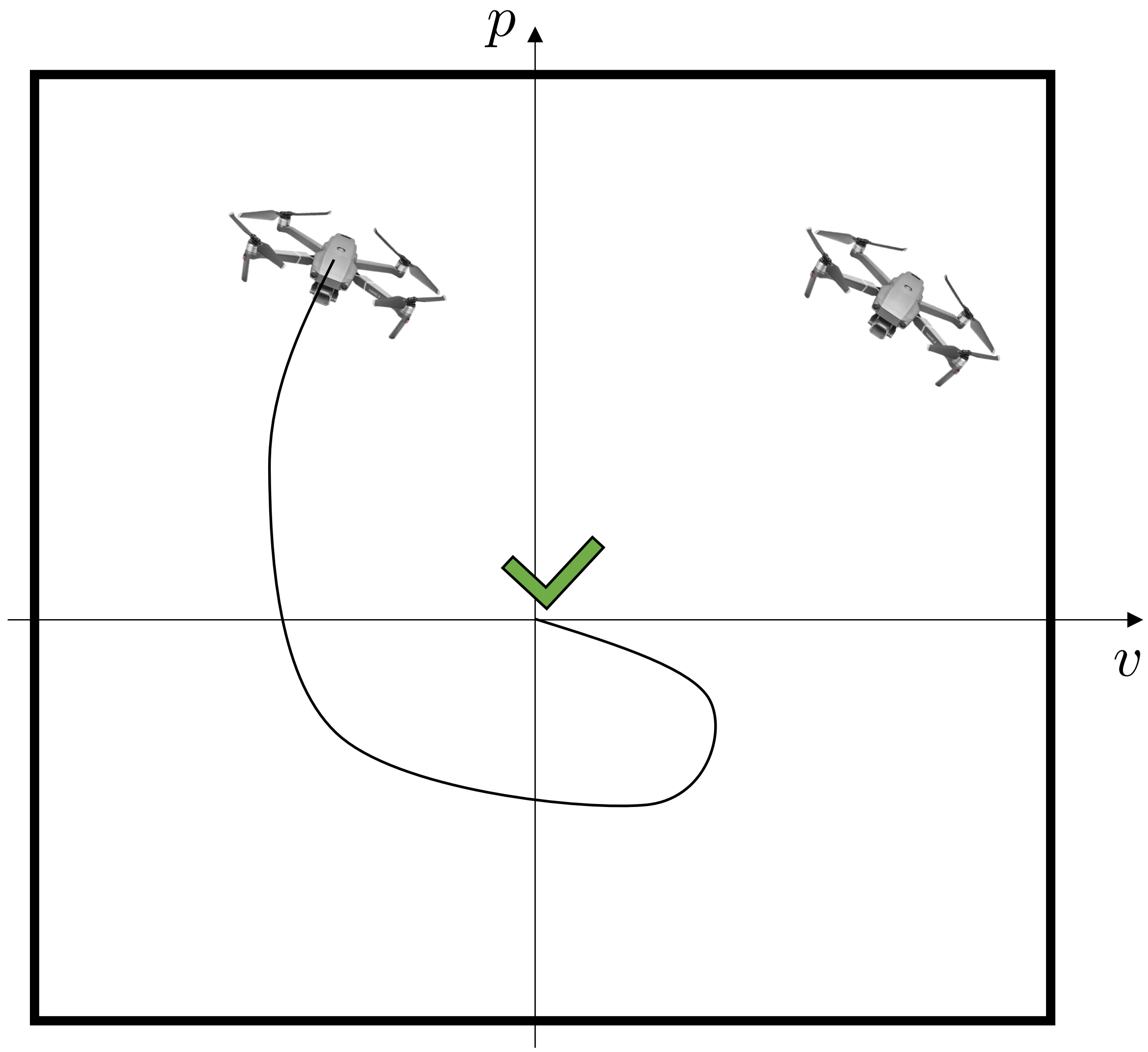
What is different in safety-critical systems? Constraints



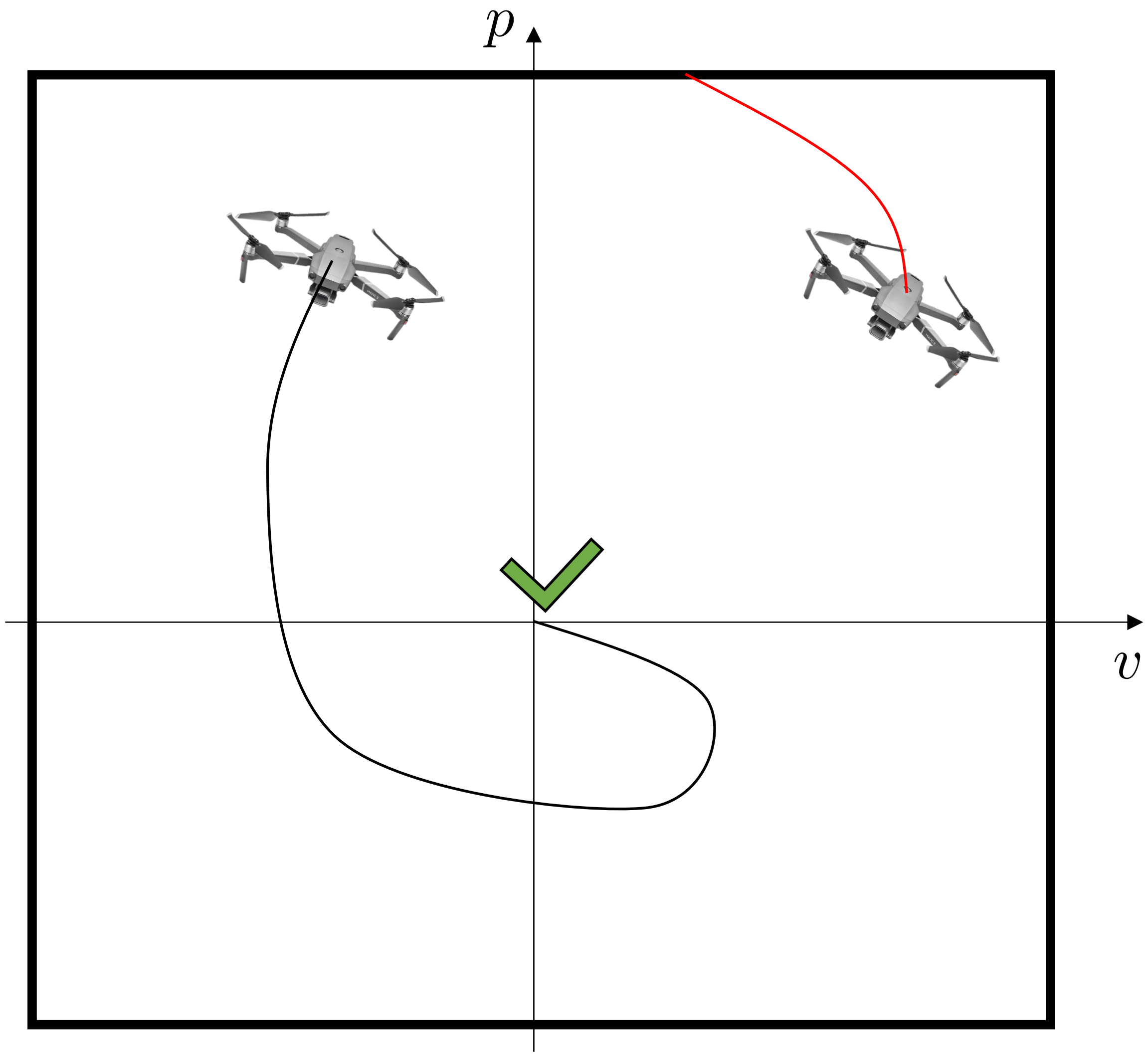
What is different in safety-critical systems? Constraints



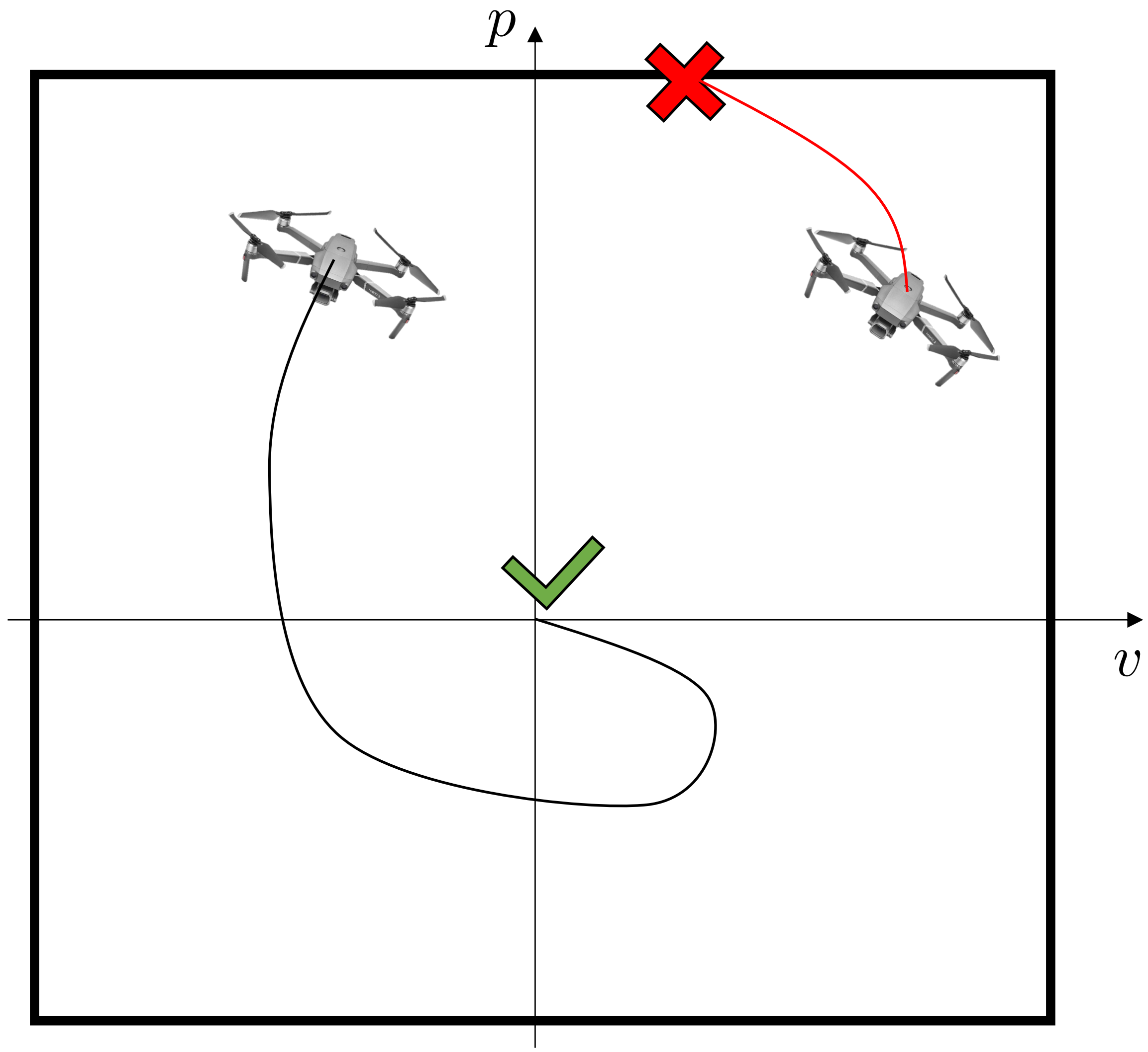
What is different in safety-critical systems? Constraints



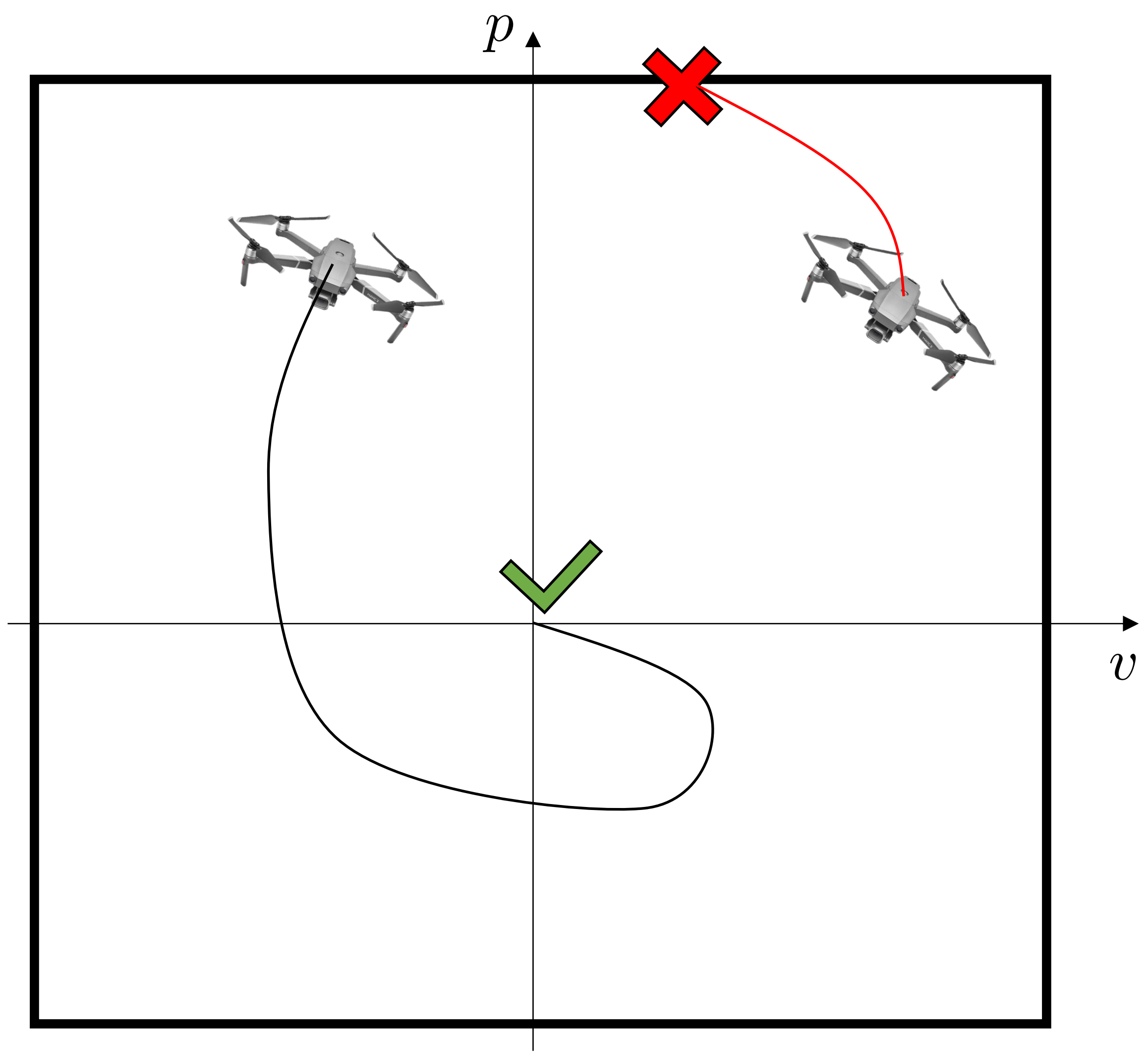
What is different in safety-critical systems? Constraints



What is different in safety-critical systems? Constraints

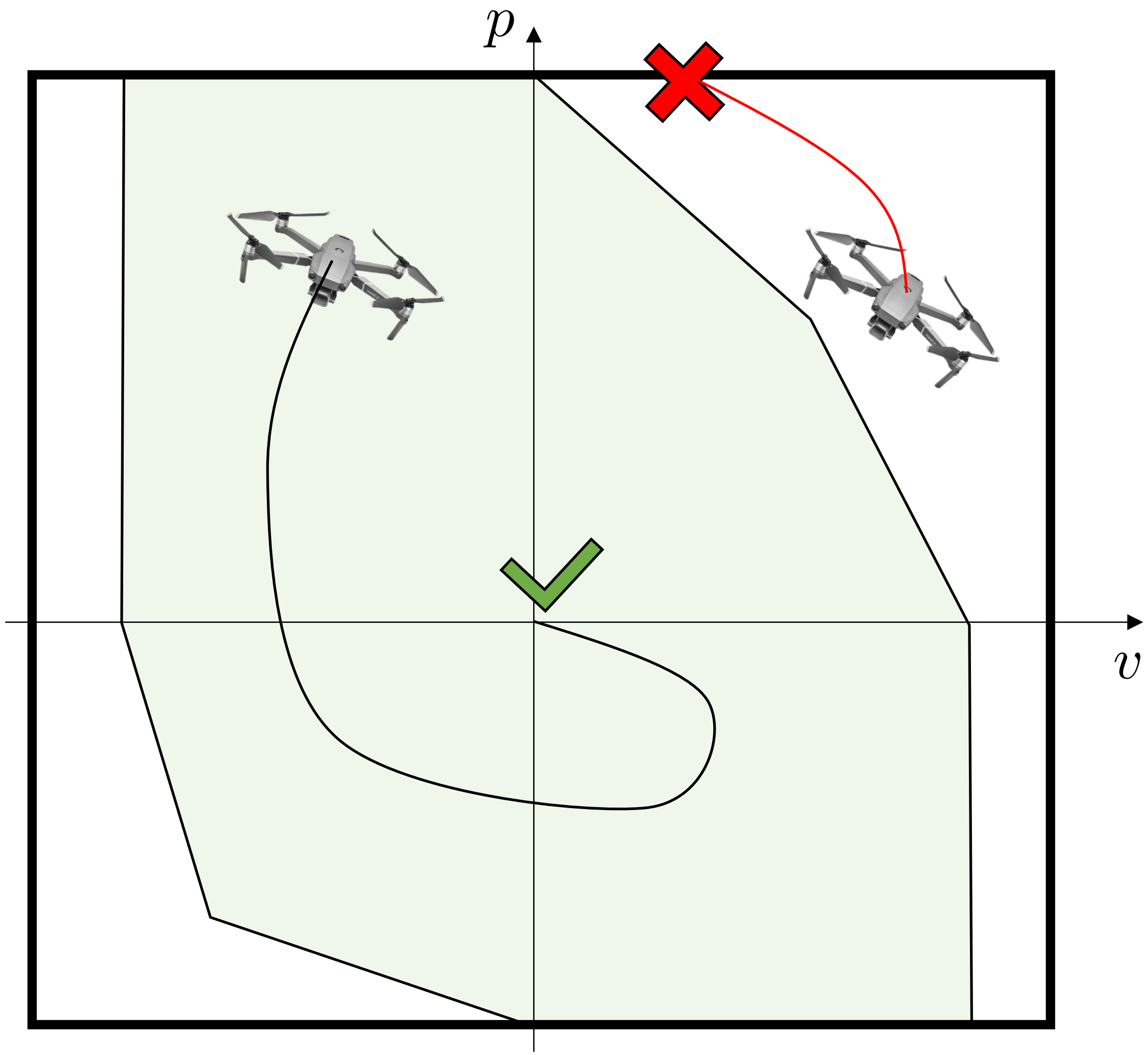


What is different in safety-critical systems? Constraints



Driving the drone to the origin is **impossible** due to inertia and input saturation

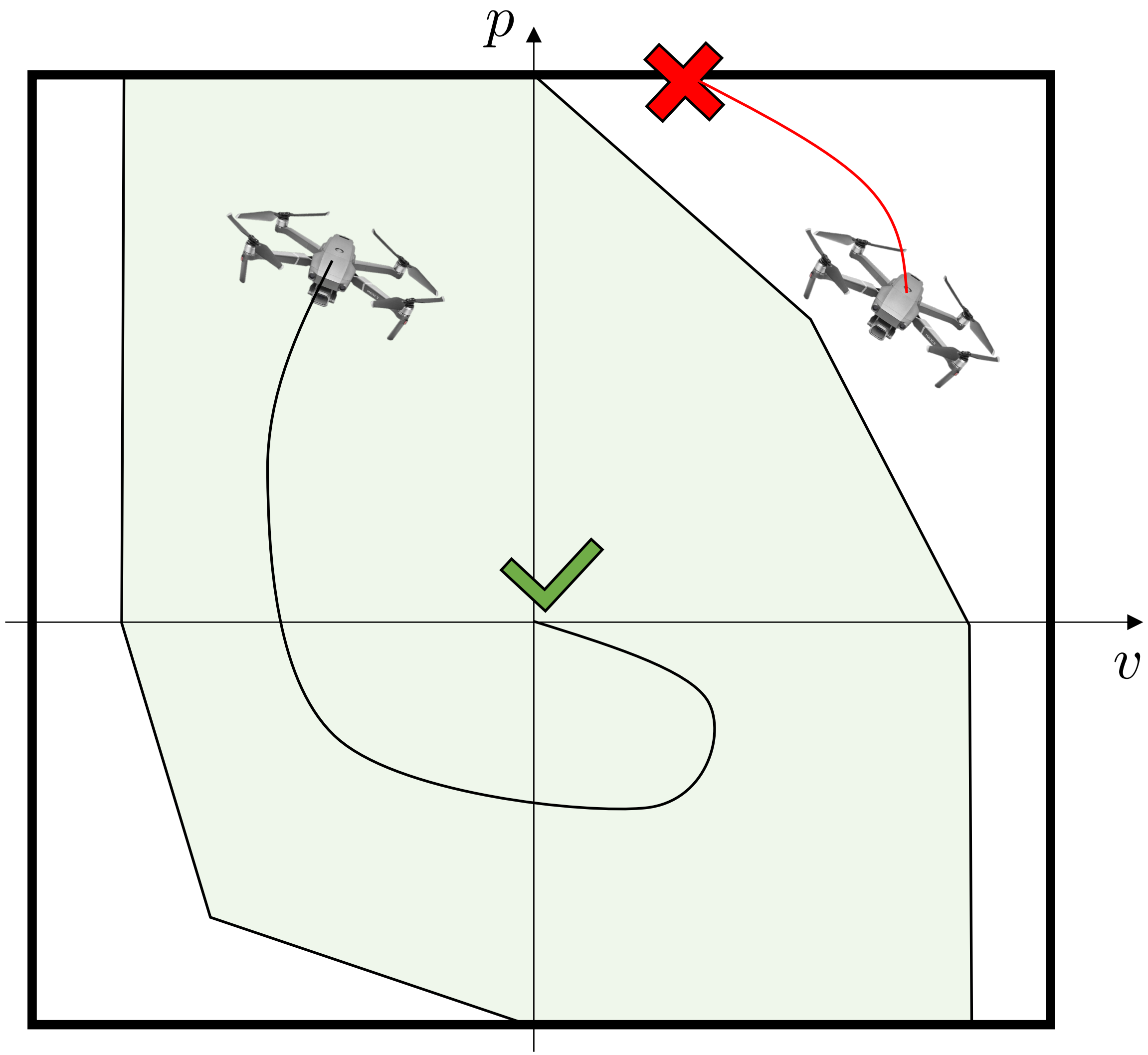
What is different in safety-critical systems? Constraints



Driving the drone to the origin is **impossible** due to inertia and input saturation

The drone can be driven to the origin only from a **subset** of the feasible set

What is different in safety-critical systems? Constraints

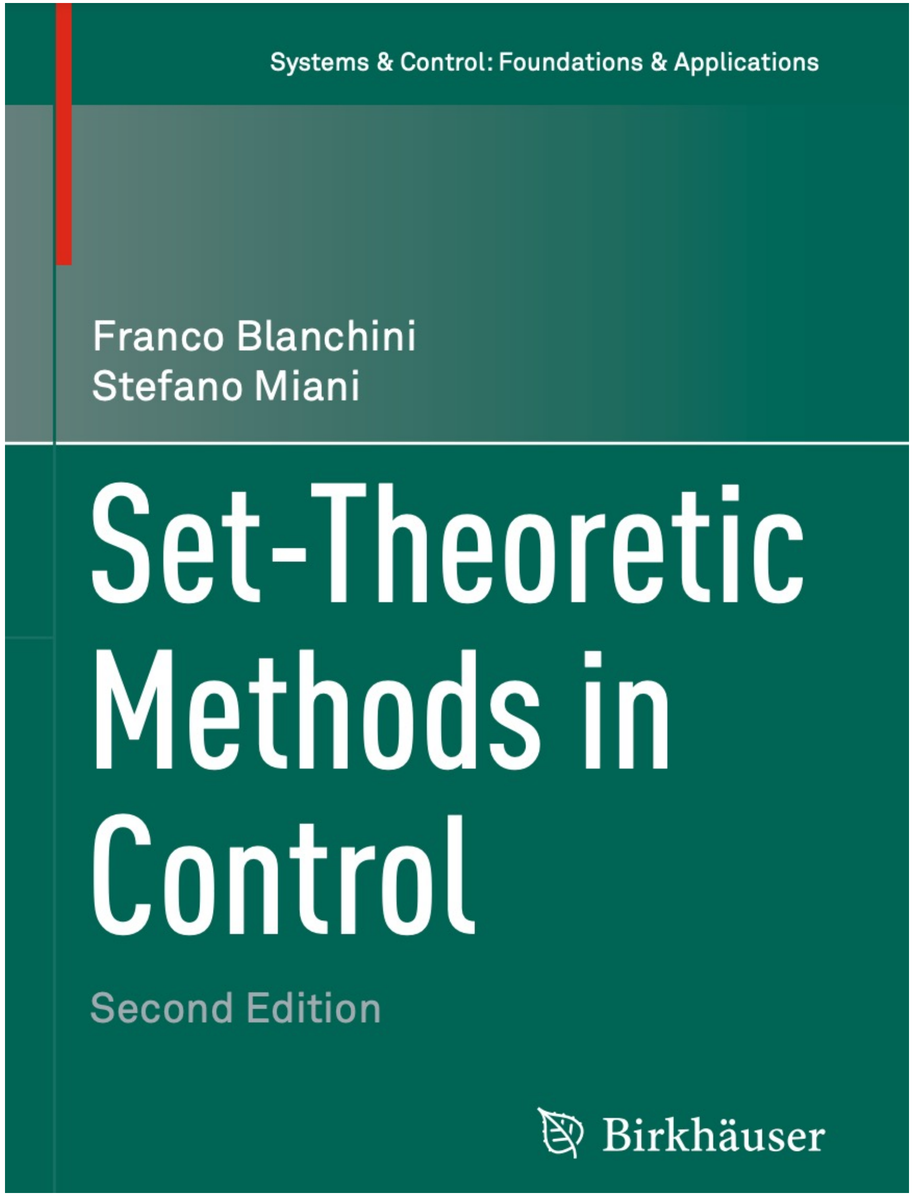


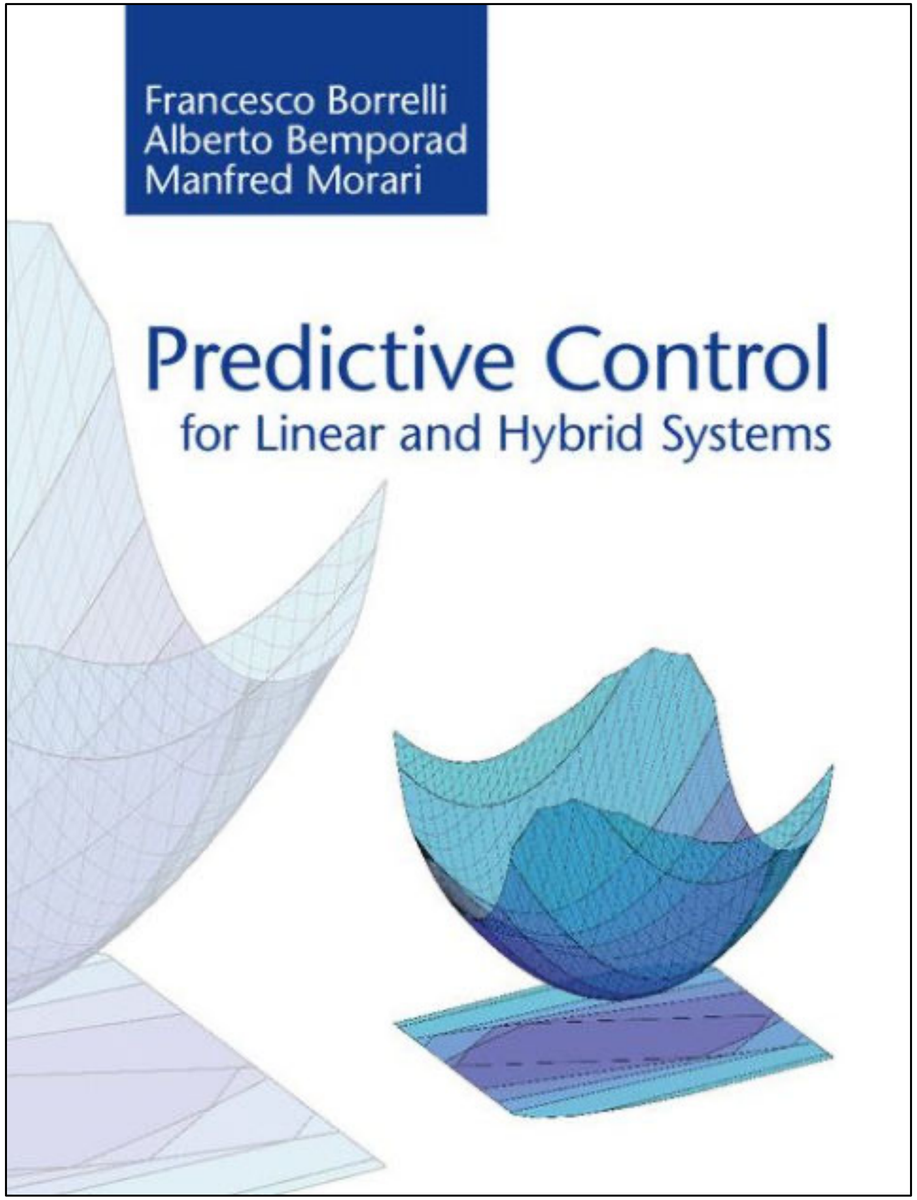
Driving the drone to the origin is **impossible** due to inertia and input saturation

The drone can be driven to the origin only from a **subset** of the feasible set

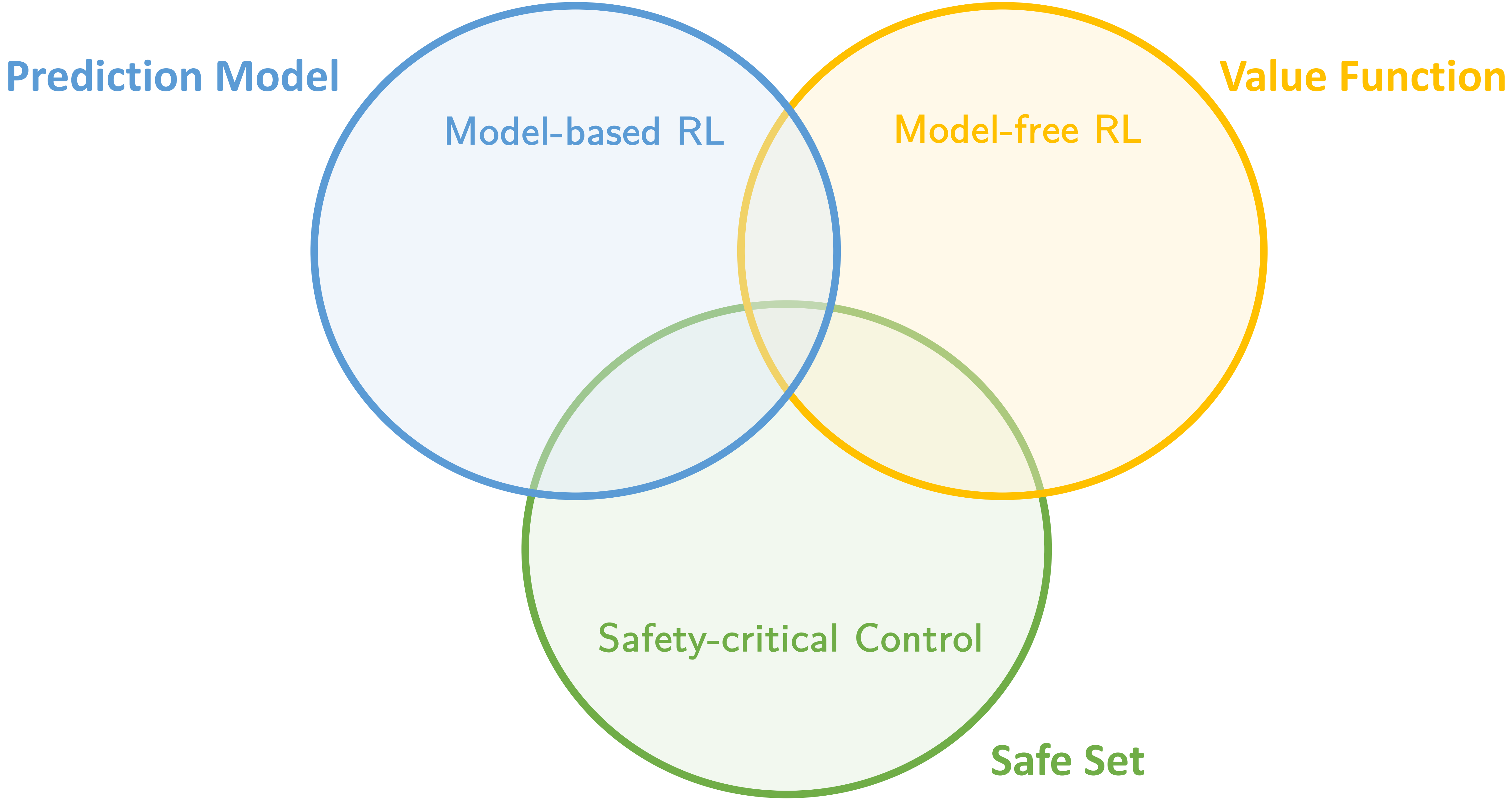
Key Message: We need to approximate the value function only over a subset of the feasible set

Computation of Safe Sets in the Control

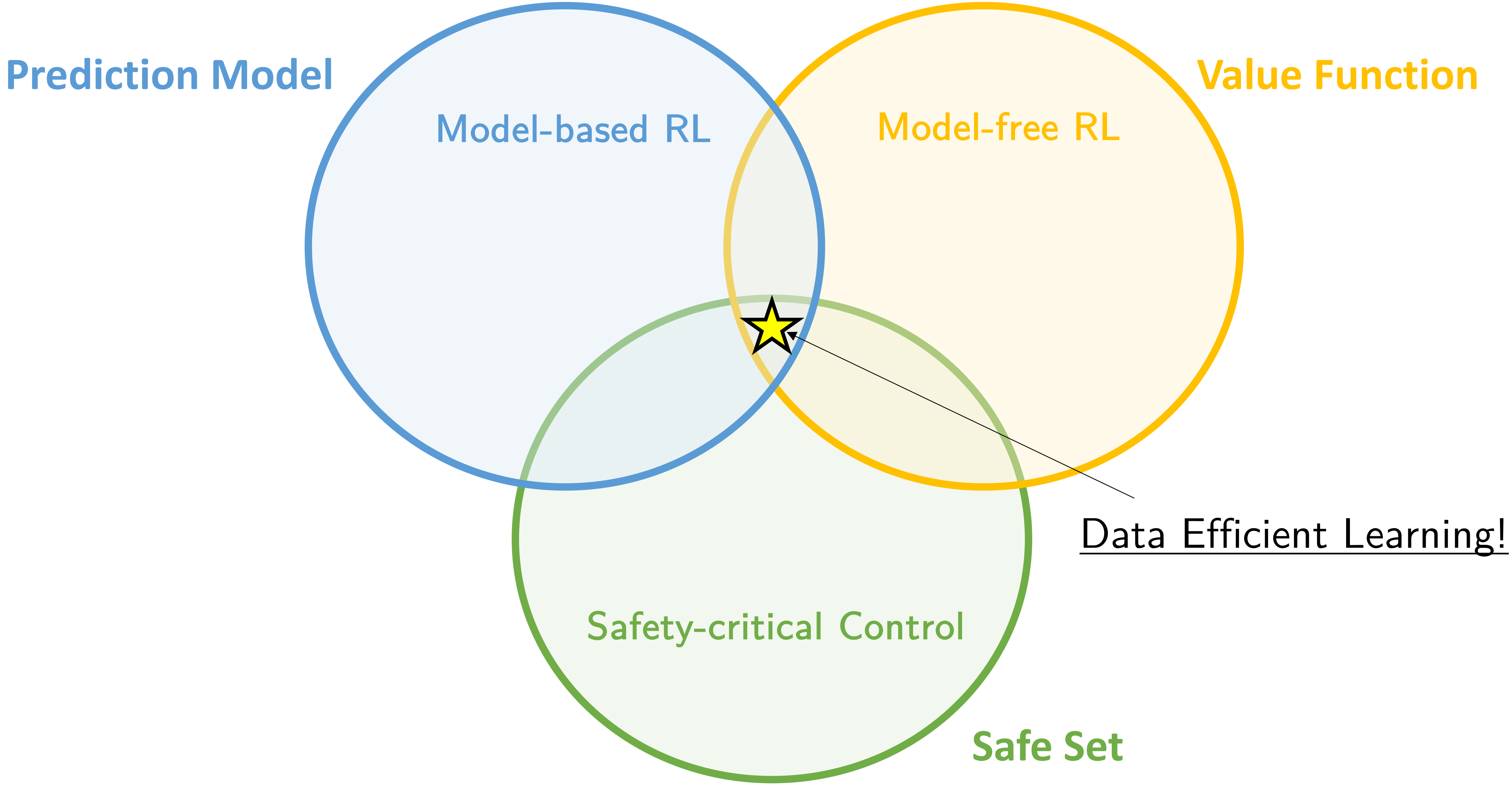




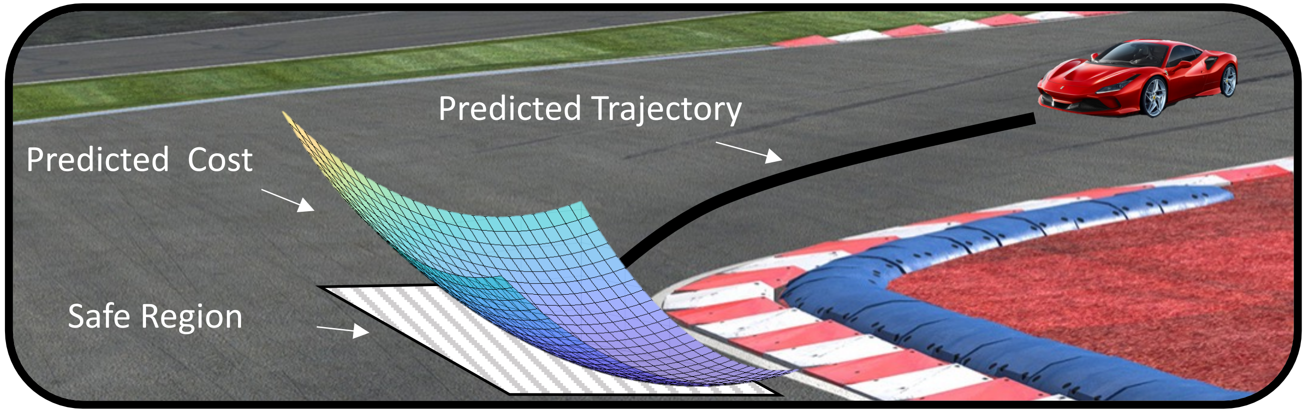
Three key components to learn



Three key components to learn

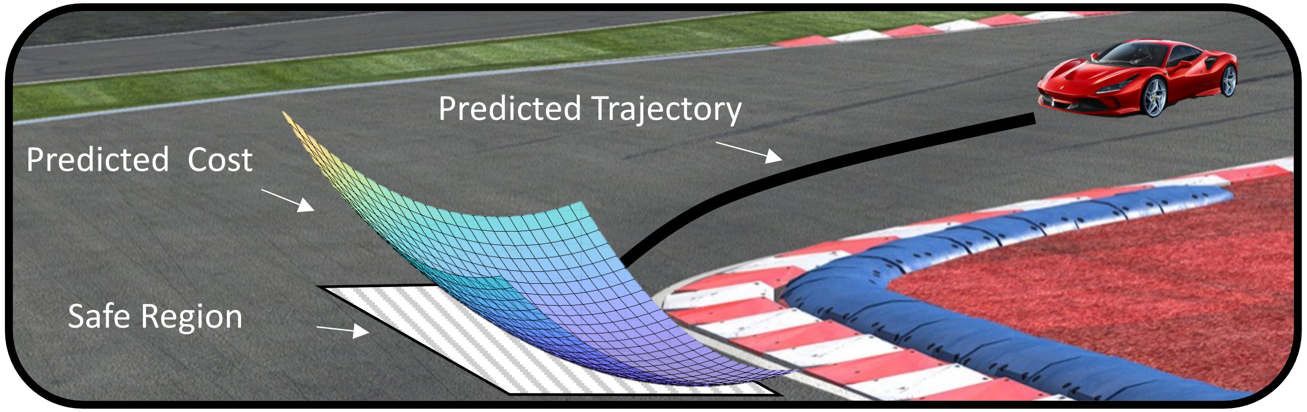


Learning Model Predictive Controller



Learning Model Predictive Controller

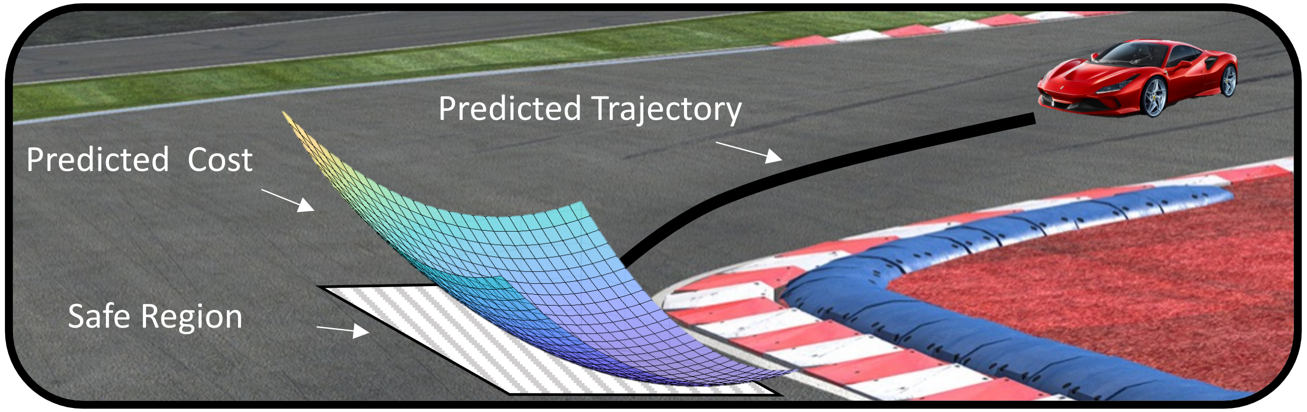
At time t of lap j solve the following Constrained Finite Time Optimal Control Problem (CFTOCP)



Learning Model Predictive Controller

At time t of lap j solve the following Constrained Finite Time Optimal Control Problem (CFTOCP)

$$J_{0\rightarrow N}^{\mathrm{LMPC},j}(x(t)) = \min_{u_t,\dots,u_{N-1}} \sum_{k=0}^{N-1} h(x_k, u_k) + V^{j-1}(x_N, x)$$

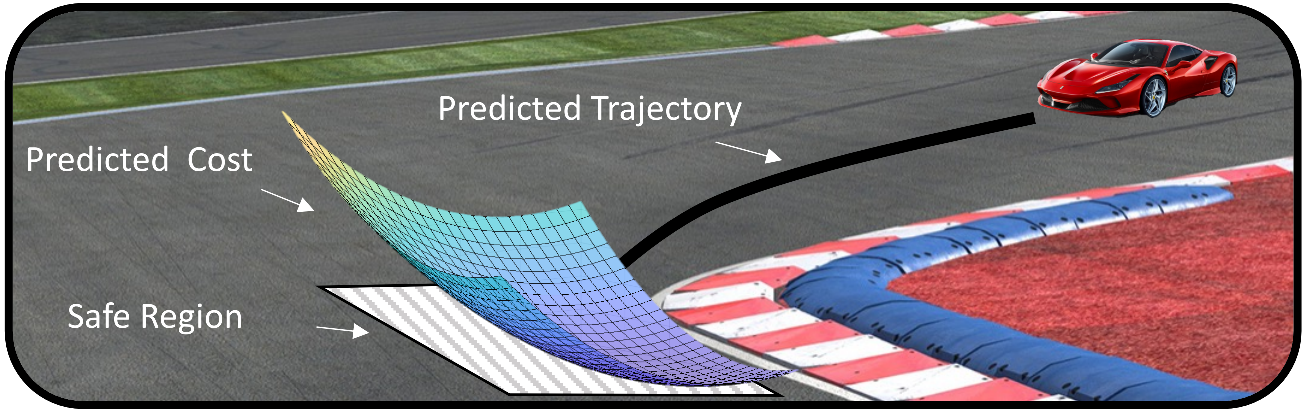


Learning Model Predictive Controller

At time t of lap j solve the following Constrained Finite Time Optimal Control Problem (CFTOCP)

$$J_{0\rightarrow N}^{\text{LMPC},j}(x(t)) = \min_{u_t,\ldots,u_{N-1}} \sum_{k=0}^{N-1} h(x_k, u_k) + V^{j-1}(x_N, x)$$

Value Function



Learning Model Predictive Controller

At time t of lap j solve the following Constrained Finite Time Optimal Control Problem (CFTOCP)

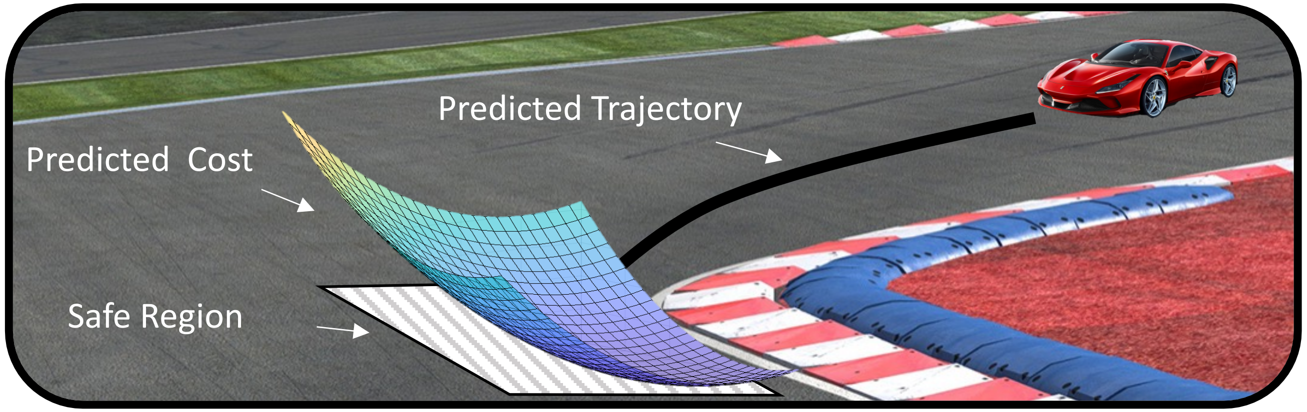
$$J_{0\rightarrow N}^{\mathrm{LMPC},j}(x(t)) = \min_{u_t,\ldots,u_{N-1}} \sum_{k=0}^{N-1} h(x_k, u_k) + V^{j-1}(x_N, x)$$

s.t.

$$x_{k+1} = A_k x_k + B_k u_k + C_k,$$

$$x_t = x(t),$$

Value Function



Learning Model Predictive Controller

At time t of lap j solve the following Constrained Finite Time Optimal Control Problem (CFTOCP)

$$J_{0\rightarrow N}^{\mathrm{LMPC},j}(x(t)) = \min_{u_t,\dots,u_{N-1}} \sum_{k=0}^{N-1} h(x_k, u_k) + V^{j-1}(x_N, x)$$

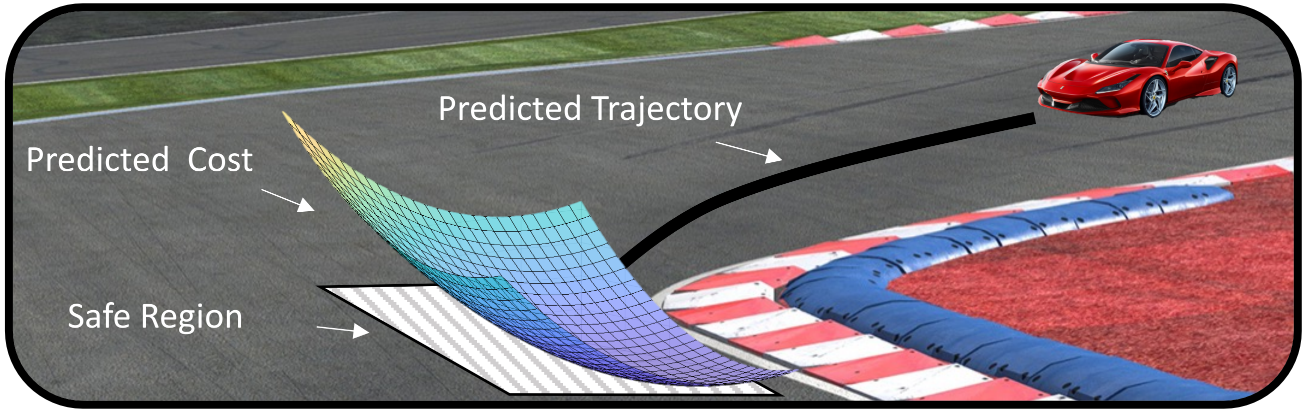
s.t.

$$x_{k+1} = A_k x_k + B_k u_k + C_k,$$

$$x_t = x(t),$$

Value Function

Prediction Model



Learning Model Predictive Controller

At time t of lap j solve the following Constrained Finite Time Optimal Control Problem (CFTOCP)

$$J_{0\to N}^{\text{LMPC},j}(x(t)) = \min_{u_t,\ldots,u_{N-1}} \sum_{k=0}^{N-1} h(x_k, u_k) + V^{j-1}(x_N, x)$$

s.t.

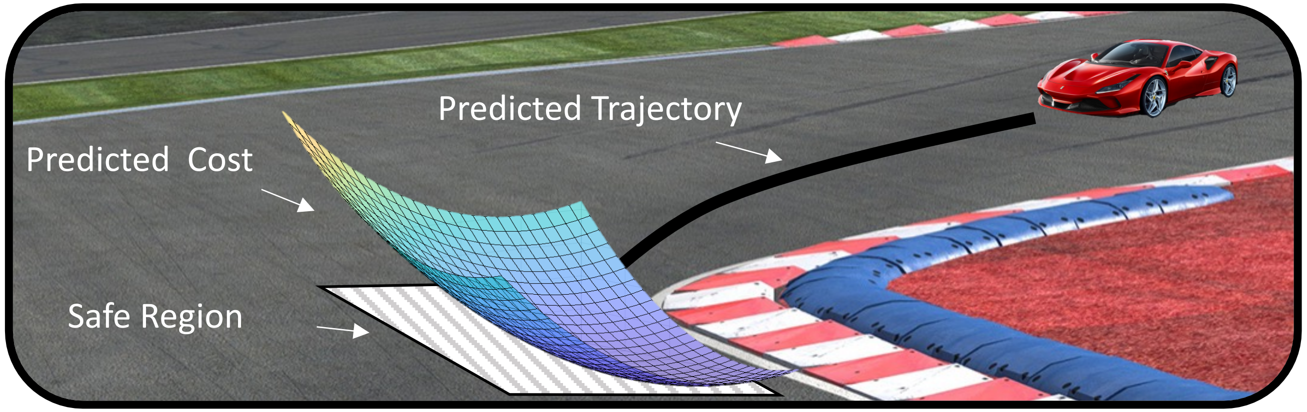
$$x_{k+1} = A_k x_k + B_k u_k + C_k,$$

$$x_t = x(t),$$

$$x_k \in \mathcal{X},\ u_k \in \mathcal{U},\ \forall k \in [0, \cdots, N-1]$$

Value Function

Prediction Model



Learning Model Predictive Controller

At time t of lap j solve the following Constrained Finite Time Optimal Control Problem (CFTOCP)

$$J^{\text{LMPC},j}_{0\rightarrow N}(x(t)) = \min_{u_t,\ldots,u_{N-1}} \sum_{k=0}^{N-1} h(x_k, u_k) + V^{j-1}(x_N, x)$$

s.t.

$$x_{k+1} = A_k x_k + B_k u_k + C_k,$$

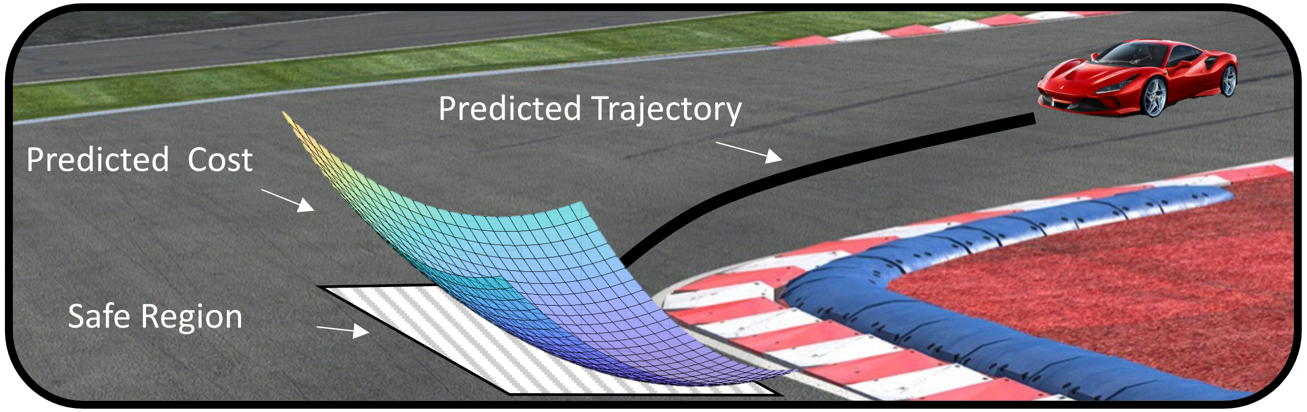
$$x_t = x(t),$$

$$x_k \in \mathcal{X},\ u_k \in \mathcal{U},\ \forall k \in [0, \cdots, N-1]$$

$$x_N \in \mathcal{CS}^{j-1}(x),$$

Value Function

Prediction Model



Learning Model Predictive Controller

At time t of lap j solve the following Constrained Finite Time Optimal Control Problem (CFTOCP)

$$J_{0\rightarrow N}^{\mathrm{LMPC},j}(x(t)) = \min_{u_t,\ldots,u_{N-1}} \sum_{k=0}^{N-1} h(x_k, u_k) + V^{j-1}(x_N, x)$$

s.t.

$$x_{k+1} = A_k x_k + B_k u_k + C_k,$$

$$x_t = x(t),$$

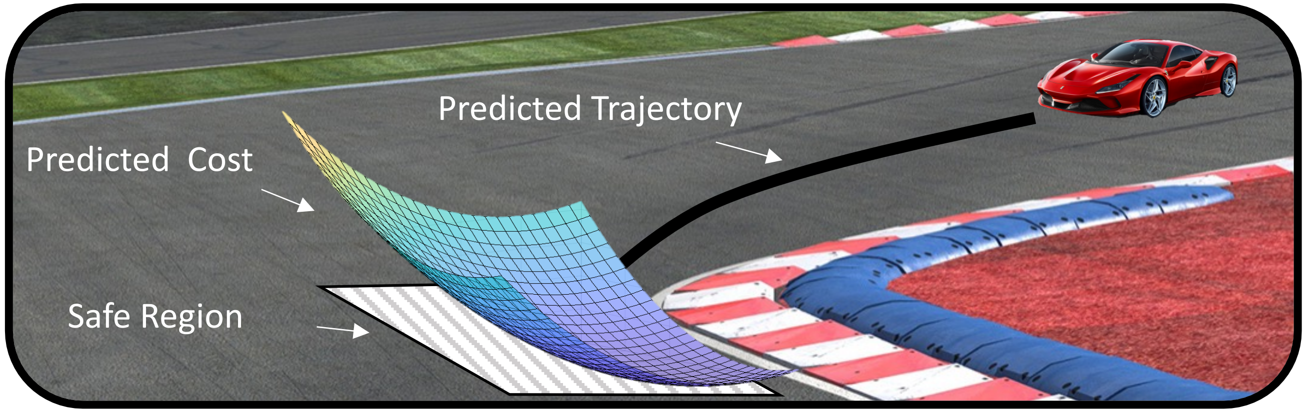
$$x_k \in \mathcal{X},\ u_k \in \mathcal{U},\ \forall k \in [0, \cdots, N-1]$$

$$x_N \in \mathcal{CS}^{j-1}(x),$$

Value Function

Prediction Model

Safe Region



Learning Model Predictive Controller

At time t of lap j solve the following Constrained Finite Time Optimal Control Problem (CFTOCP)

$$J_{0\to N}^{\mathrm{LMPC},j}(x(t)) = \min_{u_t,\dots,u_{N-1}} \sum_{k=0}^{N-1} h(x_k, u_k) + V^{j-1}(x_N, x)$$

s.t.

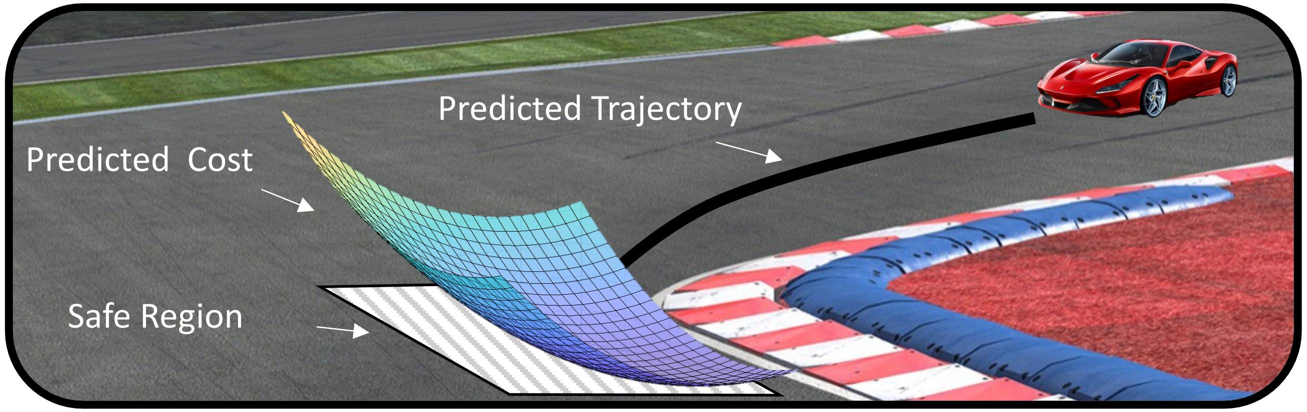
$$x_{k+1} = A_k x_k + B_k u_k + C_k,$$

$$x_t = x(t),$$

$$x_k \in \mathcal{X},\; u_k \in \mathcal{U},\; \forall k \in [0, \cdots, N-1]$$

$$x_N \in \mathcal{CS}^{j-1}(x),$$

Prediction Model



Learning Model Predictive Controller

At time t of lap j solve the following Constrained Finite Time Optimal Control Problem (CFTOCP)

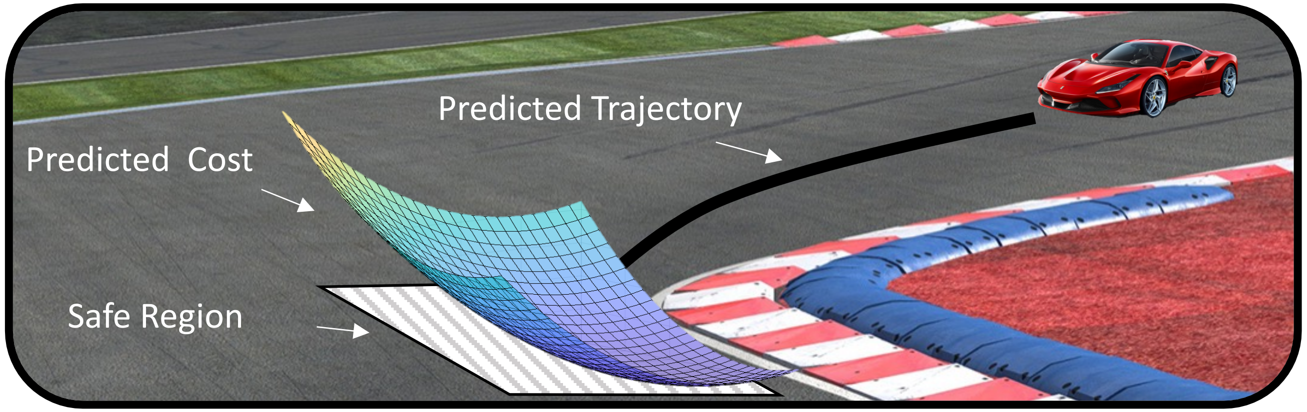
$$J_{0\rightarrow N}^{\mathrm{LMPC},j}(x(t)) = \min_{u_t,\ldots,u_{N-1}} \sum_{k=0}^{N-1} h(x_k, u_k) + V^{j-1}(x_N, x)$$

s.t.

$$x_{k+1} = A_k x_k + B_k u_k + C_k,$$
$$x_t = x(t),$$
$$x_k \in \mathcal{X},\ u_k \in \mathcal{U},\ \forall k \in [0, \cdots, N-1]$$
$$x_N \in \mathcal{CS}^{j-1}(x),$$

Prediction Model

In this topic area you will learn how to leverage DNN to estimate system dynamics



System ID in Autonomous Racing

- Nonlinear Dynamical System,

$$
\begin{aligned}
\ddot{x} &= \dot{y}\dot{\psi} + \tfrac{1}{m}\textstyle\sum_i F_{x_i} \\
\ddot{y} &= -\dot{x}\dot{\psi} + \tfrac{1}{m}\textstyle\sum_i F_{y_i} \\
\ddot{\psi} &= \tfrac{1}{I_z}(a(F_{y_{1,2}}) - b(F_{y_{2,3}}) + c(-F_{x_{1,3}} + F_{x_{2,4}}) \\
\dot{X} &= \dot{x}\cos\psi - \dot{y}\sin\psi, \quad \dot{Y} = \dot{x}\sin\psi + \dot{y}\cos\psi
\end{aligned}
$$

System ID in Autonomous Racing

- Nonlinear Dynamical System,

$$
\begin{aligned}
\ddot{x} &= \dot{y}\dot{\psi} + \frac{1}{m}\sum_i F_{x_i} \\
\ddot{y} &= -\dot{x}\dot{\psi} + \frac{1}{m}\sum_i F_{y_i} \\
\ddot{\psi} &= \frac{1}{I_z}(a(F_{y_{1,2}}) - b(F_{y_{2,3}}) + c(-F_{x_{1,3}} + F_{x_{2,4}}) \\
\dot{X} &= \dot{x}\cos\psi - \dot{y}\sin\psi, \quad \dot{Y} = \dot{x}\sin\psi + \dot{y}\cos\psi
\end{aligned}
$$

Kinematic Equations

System ID in Autonomous Racing

- Nonlinear Dynamical System,

$$
\begin{aligned}
\ddot{x} &= \dot{y}\dot{\psi} + \frac{1}{m}\sum_i F_{x_i} \\
\ddot{y} &= -\dot{x}\dot{\psi} + \frac{1}{m}\sum_i F_{y_i} \\
\ddot{\psi} &= \frac{1}{I_z}(a(F_{y_{1,2}}) - b(F_{y_{2,3}}) + c(-F_{x_{1,3}} + F_{x_{2,4}}) \\
\dot{X} &= \dot{x}\cos\psi - \dot{y}\sin\psi, \quad \dot{Y} = \dot{x}\sin\psi + \dot{y}\cos\psi
\end{aligned}
$$

Kinematic Equations

- Identifying the Dynamical System

$$
x^j_{k+1|t} = \begin{bmatrix} \dot{x}_{k+1|t} \\ \dot{y}_{k+1|t} \\ \ddot{\psi} \\ \psi_{k+1|t} \\ X_{k+1|t} \\ Y_{k+1|t} \end{bmatrix} = \begin{bmatrix} \\ \\ \\ \text{Linearized Kinematics} \\ \text{Linearized Kinematics} \\ \text{Linearized Kinematics} \end{bmatrix} x^j_{k|t} + \begin{bmatrix} \\ \\ \\ \text{Linearized Kinematics} \\ \text{Linearized Kinematics} \\ \text{Linearized Kinematics} \end{bmatrix} \begin{bmatrix} u^j_{k|t} \\ 1 \end{bmatrix}
$$

Linearization around predicted trajectory

System ID in Autonomous Racing

- Nonlinear Dynamical System,

$$\begin{aligned} \ddot{x} &= \dot{y}\dot{\psi} + \frac{1}{m}\sum_i F_{x_i} \\ \ddot{y} &= -\dot{x}\dot{\psi} + \frac{1}{m}\sum_i F_{y_i} \\ \ddot{\psi} &= \frac{1}{I_z}(a(F_{y_{1,2}}) - b(F_{y_{2,3}}) + c(-F_{x_{1,3}} + F_{x_{2,4}}) \end{aligned}$$

Dynamic Equations

$$\dot{X} = \dot{x}\cos\psi - \dot{y}\sin\psi, \quad \dot{Y} = \dot{x}\sin\psi + \dot{y}\cos\psi$$

Kinematic Equations

- Identifying the Dynamical System

$$x^j_{k+1|t} = \begin{bmatrix} \dot{x}_{k+1|t} \\ \dot{y}_{k+1|t} \\ \ddot{\psi} \\ \psi_{k+1|t} \\ X_{k+1|t} \\ Y_{k+1|t} \end{bmatrix} = \begin{bmatrix} \\ \\ \\ \text{Linearized Kinematics} \\ \text{Linearized Kinematics} \\ \text{Linearized Kinematics} \end{bmatrix} x^j_{k|t} + \begin{bmatrix} \\ \\ \\ \text{Linearized Kinematics} \\ \text{Linearized Kinematics} \\ \text{Linearized Kinematics} \end{bmatrix} \begin{bmatrix} u^j_{k|t} \\ 1 \end{bmatrix}$$

Linearization around predicted trajectory

System ID in Autonomous Racing

- Nonlinear Dynamical System,

$$\begin{aligned}\ddot{x} &= \dot{y}\dot{\psi} + \frac{1}{m}\sum_i F_{x_i}\\ \ddot{y} &= -\dot{x}\dot{\psi} + \frac{1}{m}\sum_i F_{y_i}\\ \ddot{\psi} &= \frac{1}{I_z}(a(F_{y_{1,2}}) - b(F_{y_{2,3}}) + c(-F_{x_{1,3}} + F_{x_{2,4}})\end{aligned}$$

Dynamic Equations

$$\dot{X} = \dot{x}\cos\psi - \dot{y}\sin\psi, \quad \dot{Y} = \dot{x}\sin\psi + \dot{y}\cos\psi$$

Kinematic Equations

- Identifying the Dynamical System

Local Linear Regression

$$x^j_{k+1|t} = \begin{bmatrix} \dot{x}_{k+1|t} \\ \dot{y}_{k+1|t} \\ \ddot{\psi} \\ \psi_{k+1|t} \\ X_{k+1|t} \\ Y_{k+1|t} \end{bmatrix} = \left[\begin{array}{c} \arg\min \sum_{i,s} K(x^j_{k|t} - x^i_s)||\Lambda_y \begin{bmatrix} x^j_{k|t} \\ u^j_{k|t} \\ 1 \end{bmatrix} - y^i_{s+1}||, \forall y \in \{\dot{x}, \dot{y}, \ddot{\psi}\} \\ \begin{array}{ccc} \begin{bmatrix} \text{Linearized Kinematics} \\ \text{Linearized Kinematics} \\ \text{Linearized Kinematics} \end{bmatrix} & x^j_{k|t} + & \begin{bmatrix} \text{Linearized Kinematics} \\ \text{Linearized Kinematics} \\ \text{Linearized Kinematics} \end{bmatrix} \end{array} \end{array} \right] \begin{bmatrix} u^j_{k|t} \\ 1 \end{bmatrix}$$

Linearization around predicted trajectory

Learning Model Predictive Controller

At time t of iteration j solve the following Constrained Finite Time Optimal Control Problem (CFTOCP)

$$J^{\text{LMPC},j}_{0\to N}(x_t) = \min_{u_t,\ldots,u_{N-1}} \sum_{k=0}^{N-1} h(x_k, u_k) + V^{j-1}(x_N, x)$$

s.t.

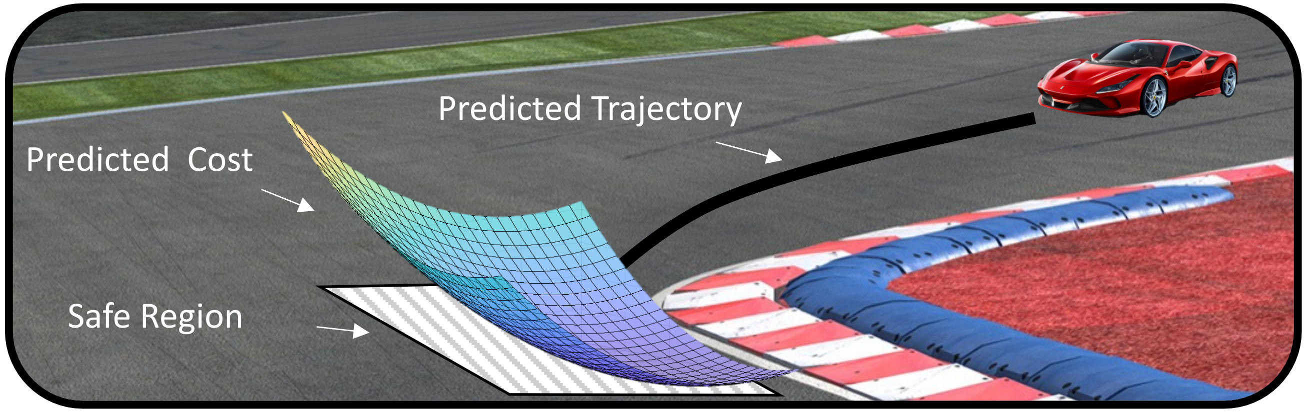
$$x_{k+1} = A_k x_k + B_k u_k + C_k,$$

$$x_t = x_t,$$

$$x_k \in \mathcal{X},\ u_k \in \mathcal{U},\ \forall k \in [0, \cdots, N-1]$$

$$x_N \in \ \mathcal{CS}^{j-1}(x),$$

Prediction Model



Learning Model Predictive Controller

At time t of iteration j solve the following Constrained Finite Time Optimal Control Problem (CFTOCP)

$$J_{0\to N}^{\mathrm{LMPC},j}(x_t) = \min_{u_t,\ldots,u_{N-1}} \sum_{k=0}^{N-1} h(x_k, u_k) + V^{j-1}(x_N, x)$$

s.t.

$$x_{k+1} = A_k x_k + B_k u_k + C_k,$$

$$x_t = x_t,$$

$$x_k \in \mathcal{X},\ u_k \in \mathcal{U},\ \forall k \in [0, \cdots, N-1]$$

$$x_N \in \mathcal{CS}^{j-1}(x),$$

Value Function

Safe Set

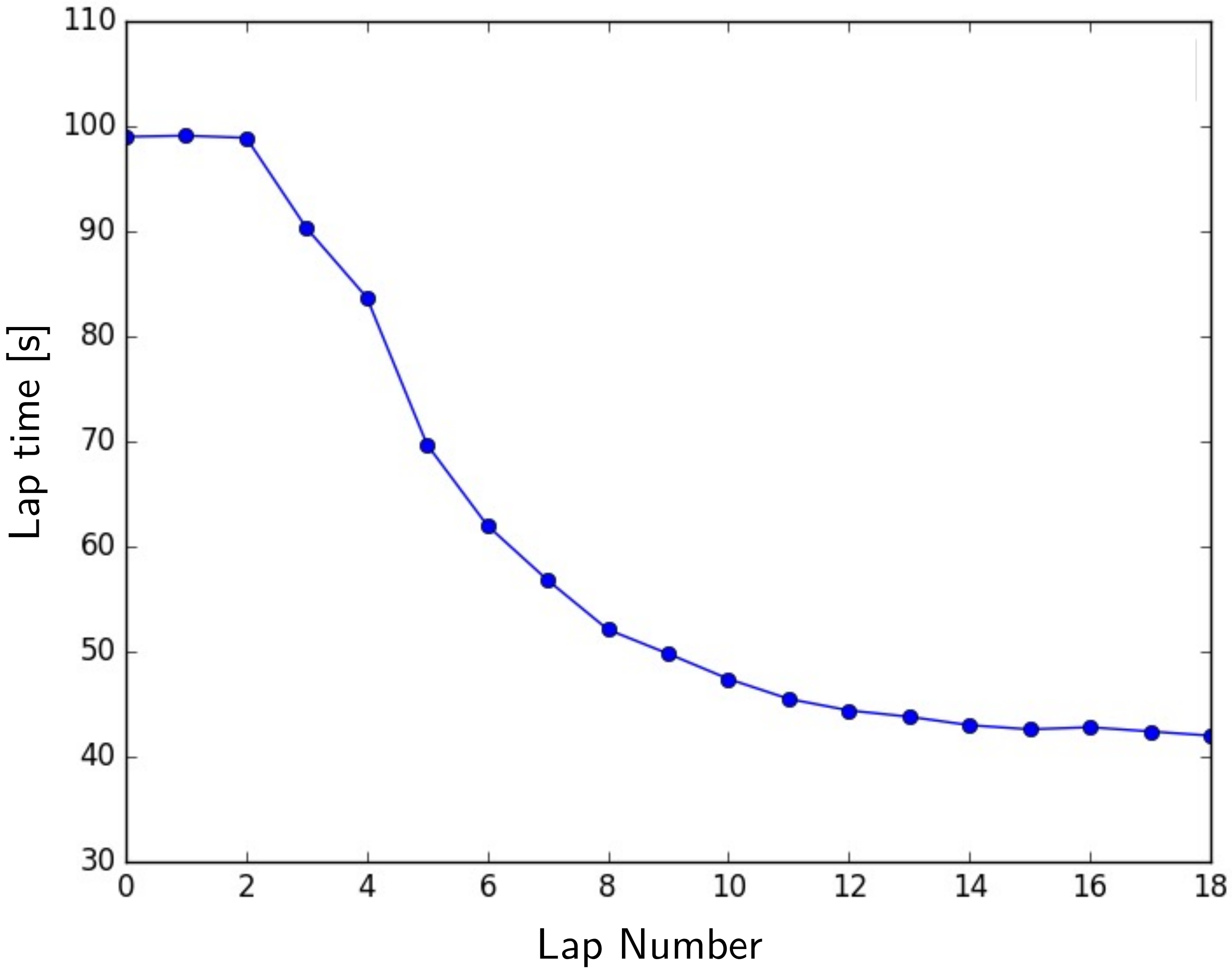
U. Rosolia, and F. Borrelli. "Learning model predictive control for iterative tasks. a data-driven control framework." *IEEE Transactions on Automatic Control* 63.7 (2017): 1883-1896.

U. Rosolia, and F. Borrelli. "Learning how to autonomously race a car: a predictive control approach." *IEEE Transactions on Control Systems Technology* 28.6 (2019): 2713-2719.

Learning Model Predictive Controller full-size vehicle experiments

Credits: Siddharth Nair, Nitin Kapania and Ugo Rosolia

Lap Time



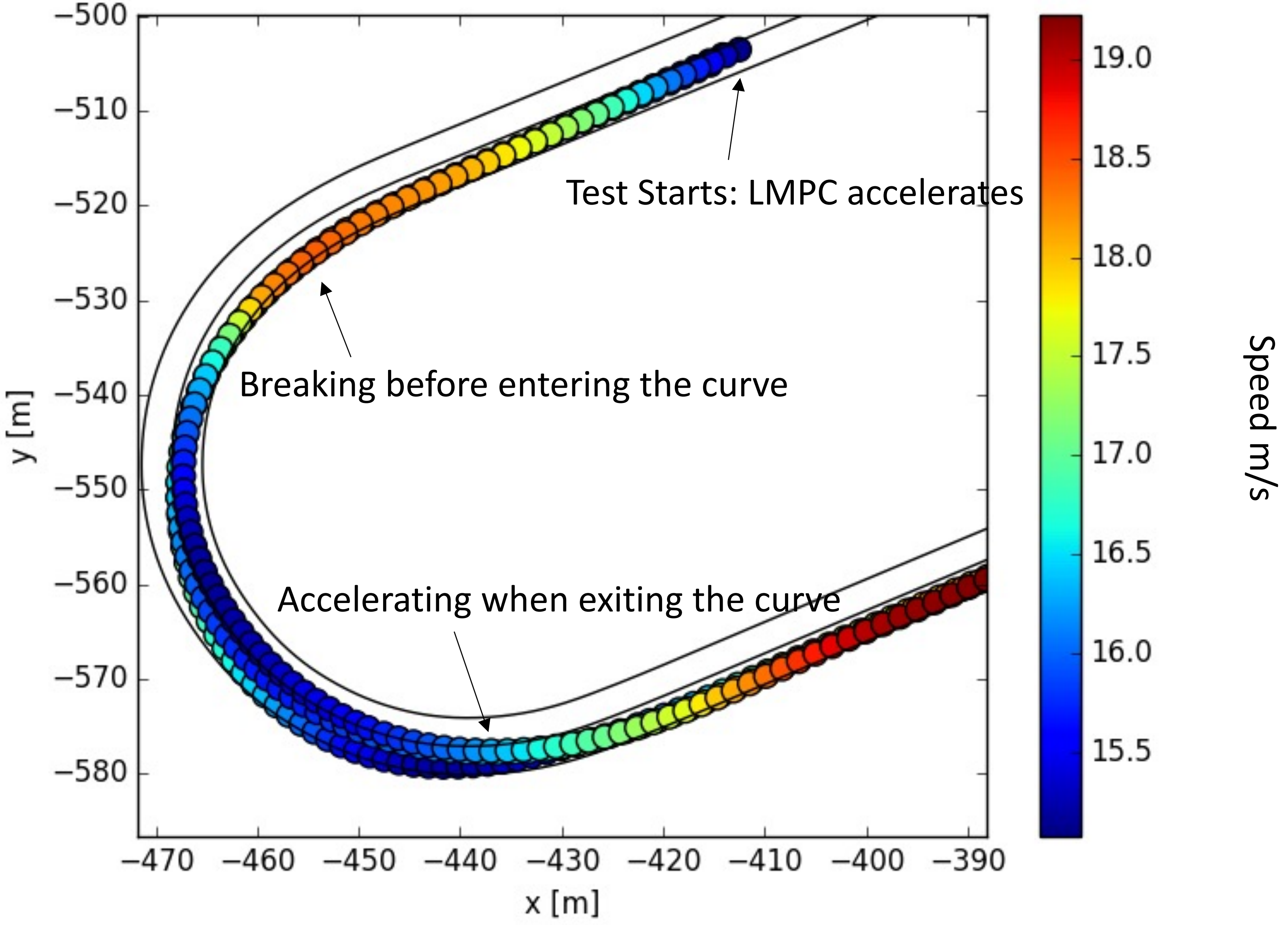
The control policy is constructed using ∼1k data points (last 2 laps)

The control action is computed using ∼100 data points

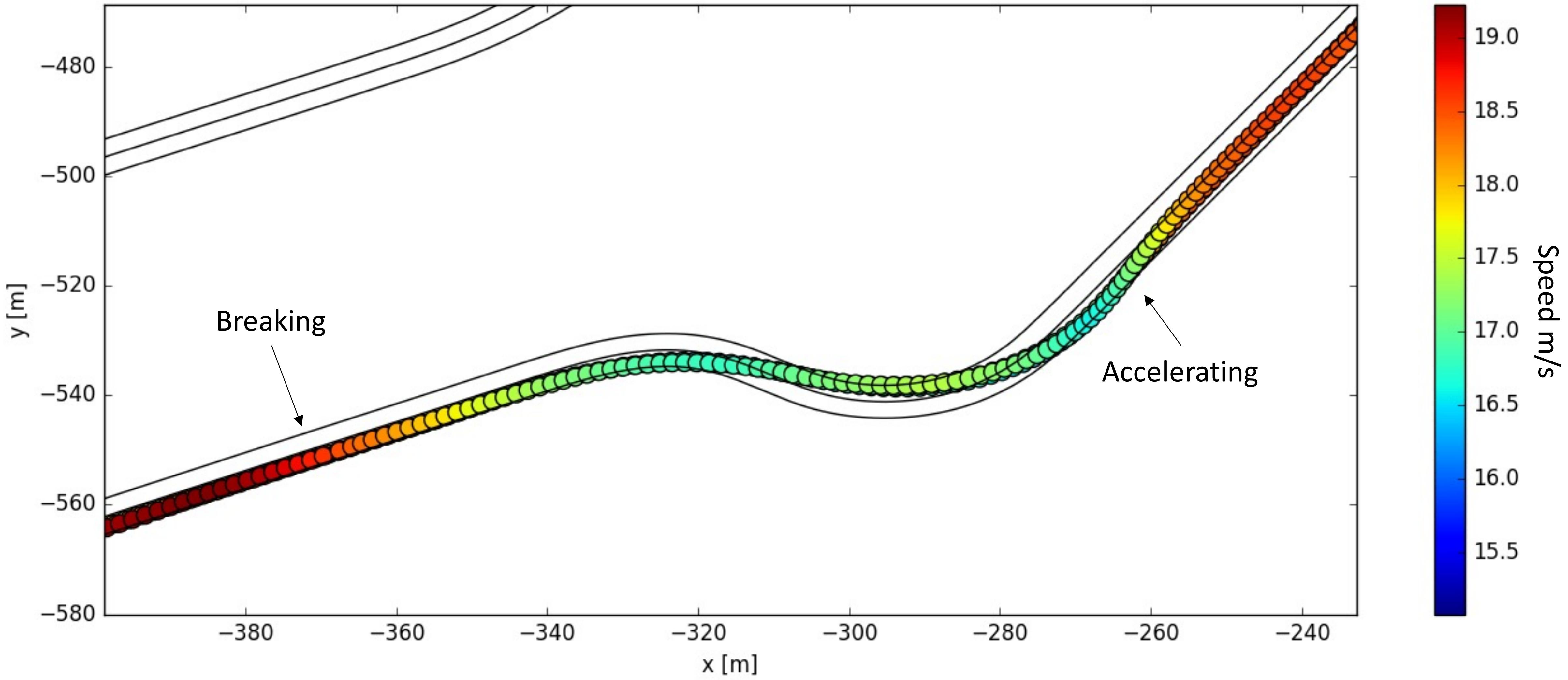
Learning Model Predictive Controller full-size vehicle experiments

Credits: Siddharth Nair, Nitin Kapania and Ugo Rosolia

Velocity Profile at Convergence (Curve 1)

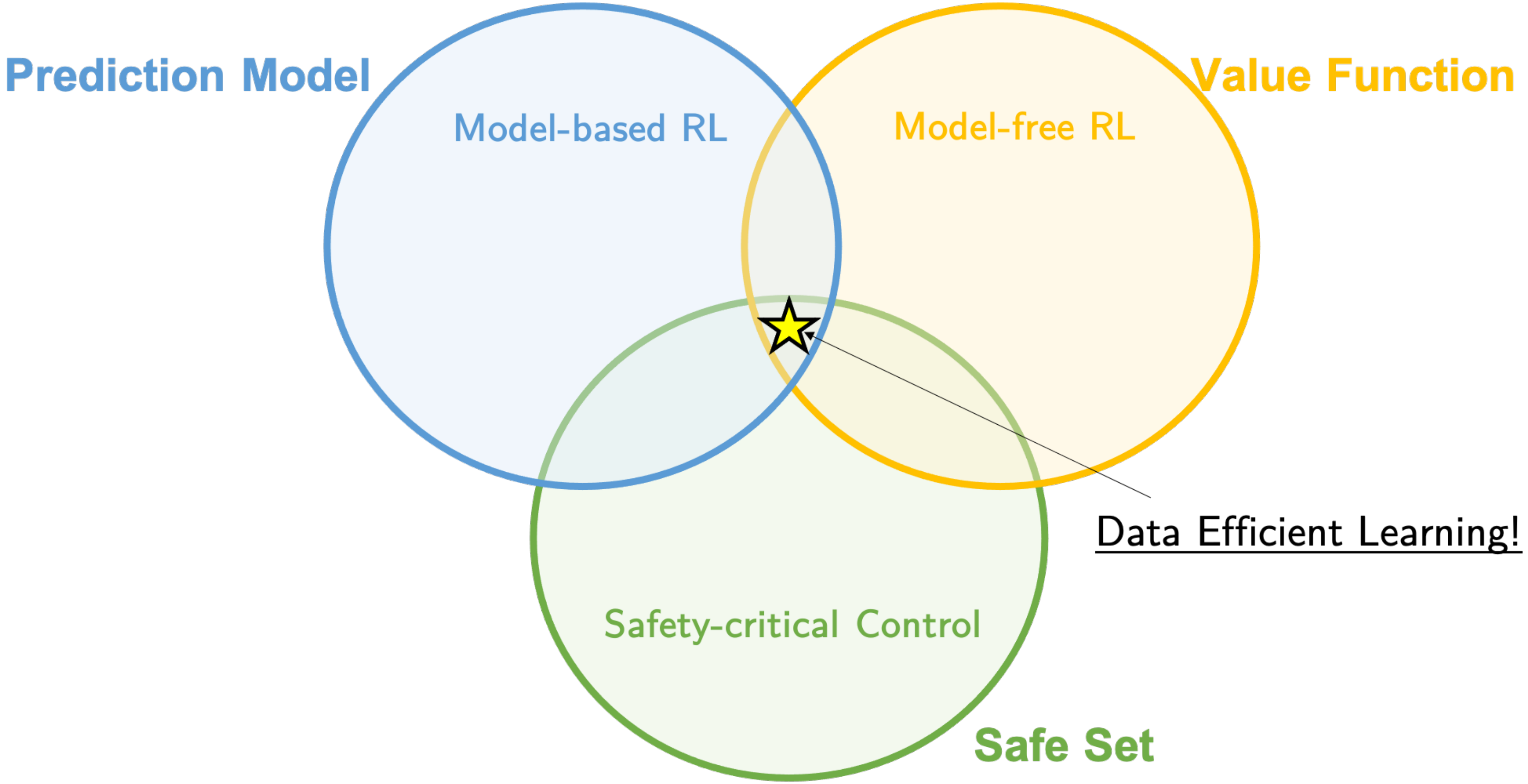


Velocity Profile at Convergence (Chicane)



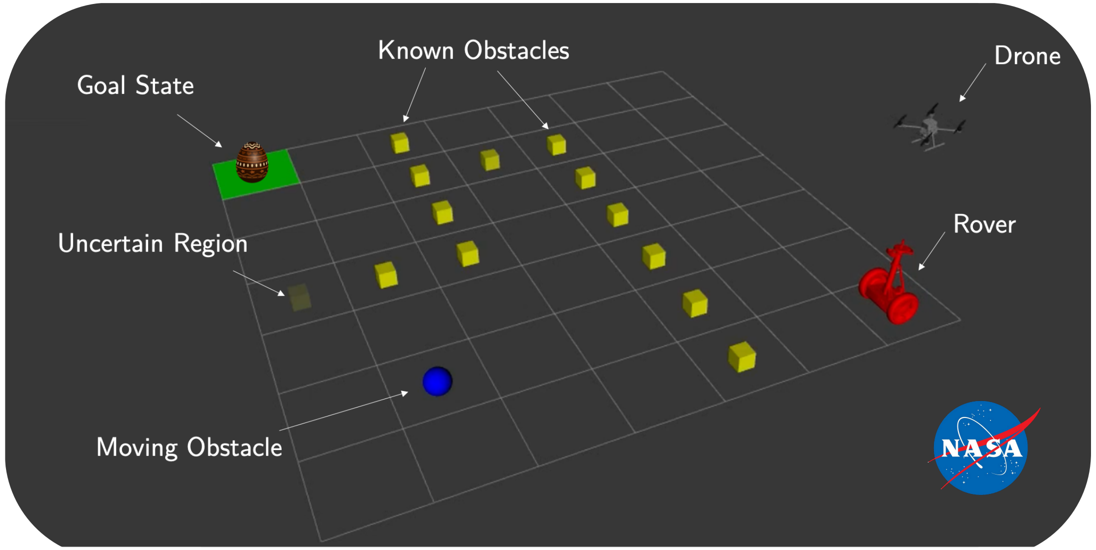
The key components

- Predicted trajectory given by prediction model
- Predicted cost estimated by value function
- Safe region estimated by the safe set



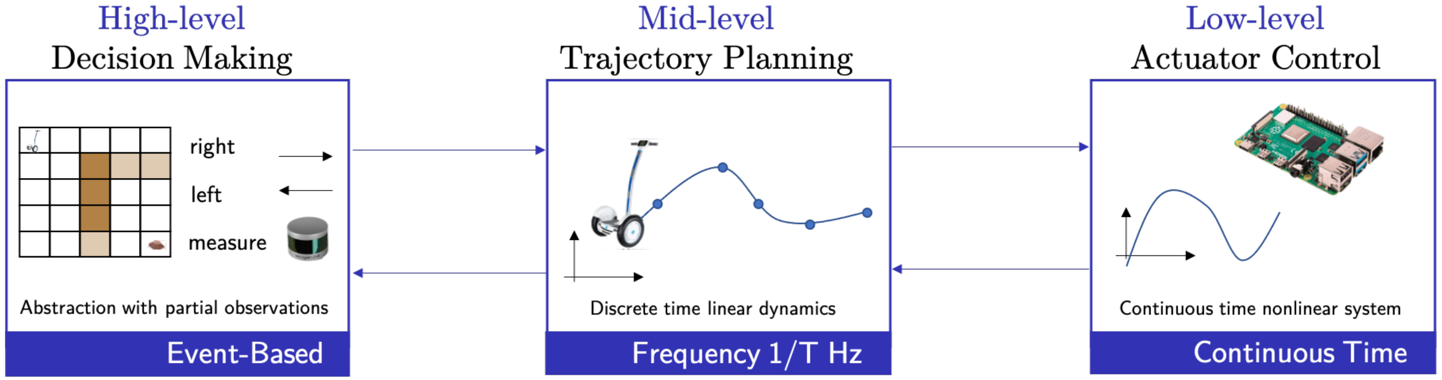
What is next?

- Partial Observability

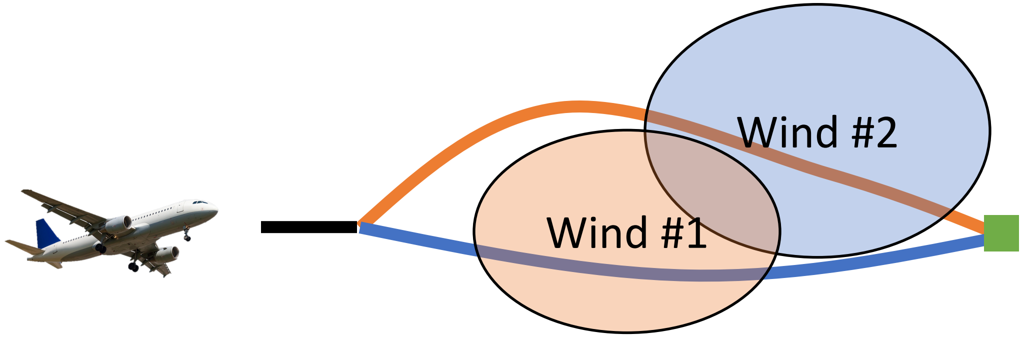


- Multi-agent systems

- Hierarchy + Learning

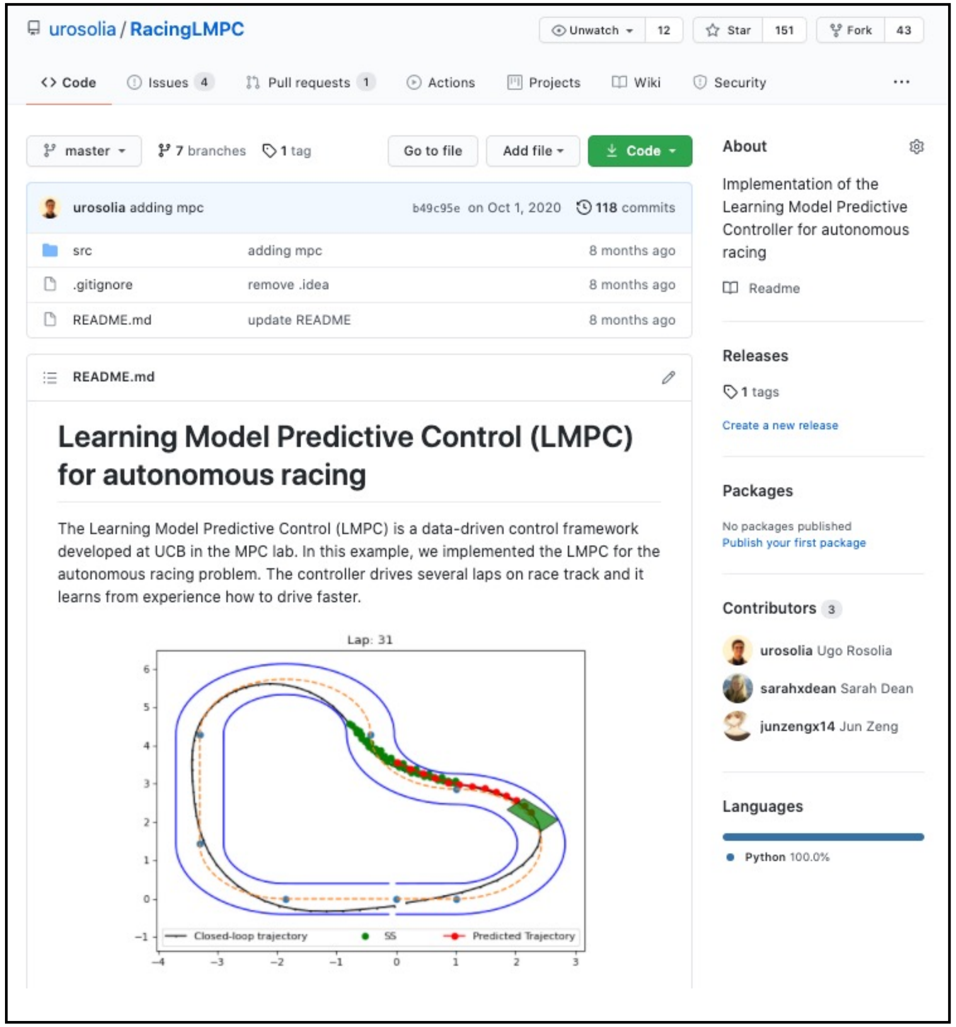


- Optimize over strategies, not trajectories

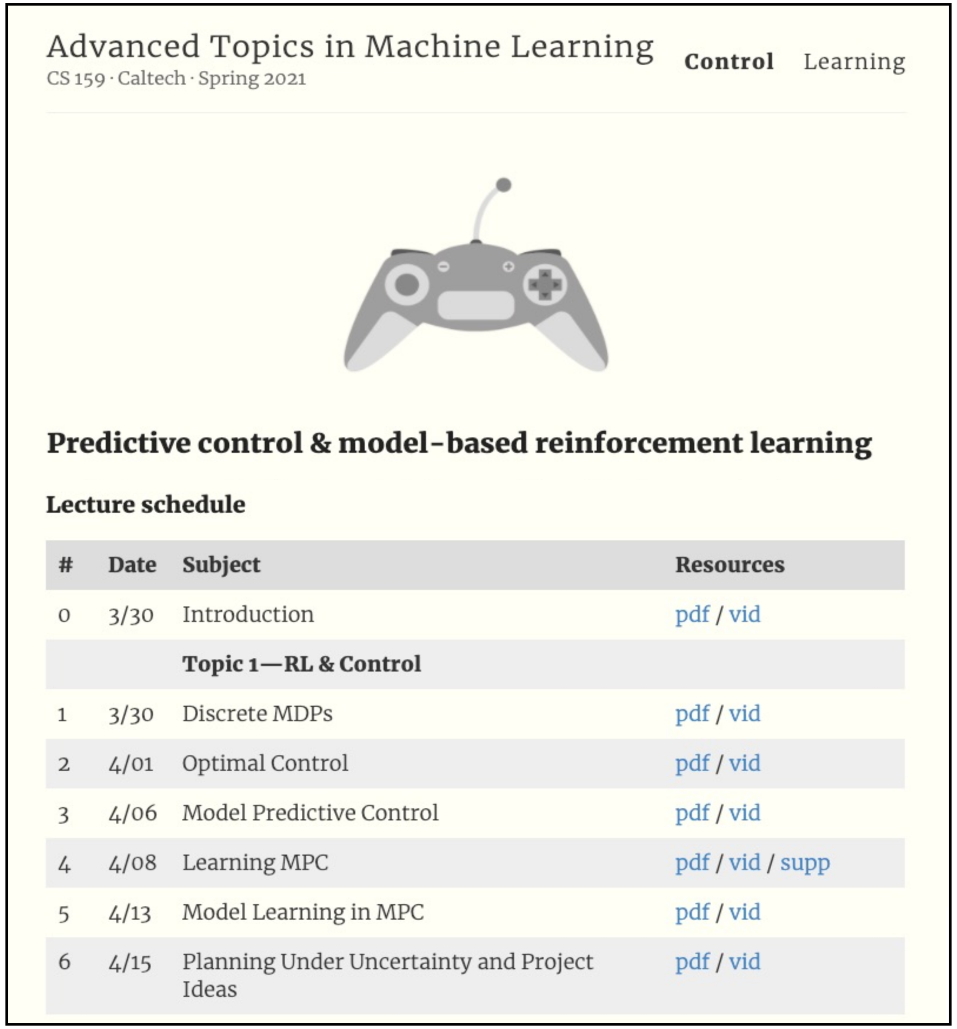


Thanks! Questions?

Code available online



Course material online



#	Date	Subject	Resources
0	3/30	Introduction	pdf / vid
		Topic 1—RL & Control	
1	3/30	Discrete MDPs	pdf / vid
2	4/01	Optimal Control	pdf / vid
3	4/06	Model Predictive Control	pdf / vid
4	4/08	Learning MPC	pdf / vid / supp
5	4/13	Model Learning in MPC	pdf / vid
6	4/15	Planning Under Uncertainty and Project Ideas	pdf / vid