

CS159 Lecture 5: LMPC and Model Learning

Ugo Rosolia

Caltech

Spring 2021

This week

- ▶ Recap of Lecture #4
- ▶ LMPC implementation
- ▶ Autonomous Racing Experiments
- ▶ Model learning in MPC (a brief summary)
- ▶ Planning under uncertainty
- ▶ LMPC for uncertain systems

Recap of Lecture #4

Stability Proof:

Recap of Lecture #4

Stability Proof:

We have shown that

Recap of Lecture #4

Stability Proof:

We have shown that

$$J_t^*(x) > 0, \forall x \neq 0$$

$$J_t^*(0) = 0$$

Recap of Lecture #4

Stability Proof:

We have shown that

$$J_t^*(x) > 0, \forall x \neq 0$$

$$J_t^*(0) = 0$$

$$J_t^*(x(t)) < J_t^*(x(t+1)), \forall x(t) \neq 0$$

Recap of Lecture #4

Stability Proof:

We have shown that

$$\left. \begin{array}{l} J_t^*(x) > 0, \forall x \neq 0 \\ J_t^*(0) = 0 \\ J_t^*(x(t)) < J_t^*(x(t+1)), \forall x(t) \neq 0 \end{array} \right\} \lim_{t \rightarrow \infty} J_t^*(x(t)) = 0$$

Recap of Lecture #4

Stability Proof:

We have shown that

$$\left. \begin{array}{l} J_t^*(x) > 0, \forall x \neq 0 \\ J_t^*(0) = 0 \\ J_t^*(x(t)) < J_t^*(x(t+1)), \forall x(t) \neq 0 \end{array} \right\} \lim_{t \rightarrow \infty} J_t^*(x(t)) = 0$$

To show stability we need to that $J_t^*(x)$ is Lipchitz continuous (can be relaxed to continuity around the origin).

Recap of Lecture #4

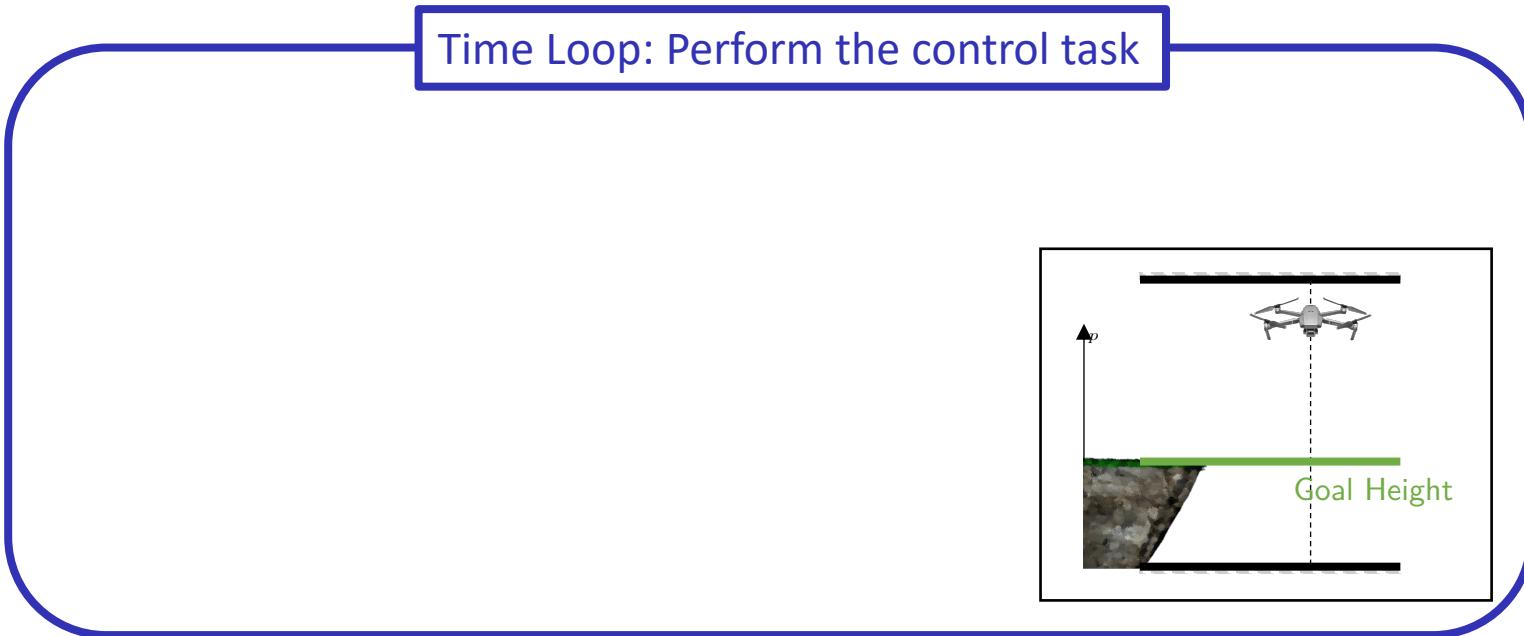
Stability Proof:

We have shown that

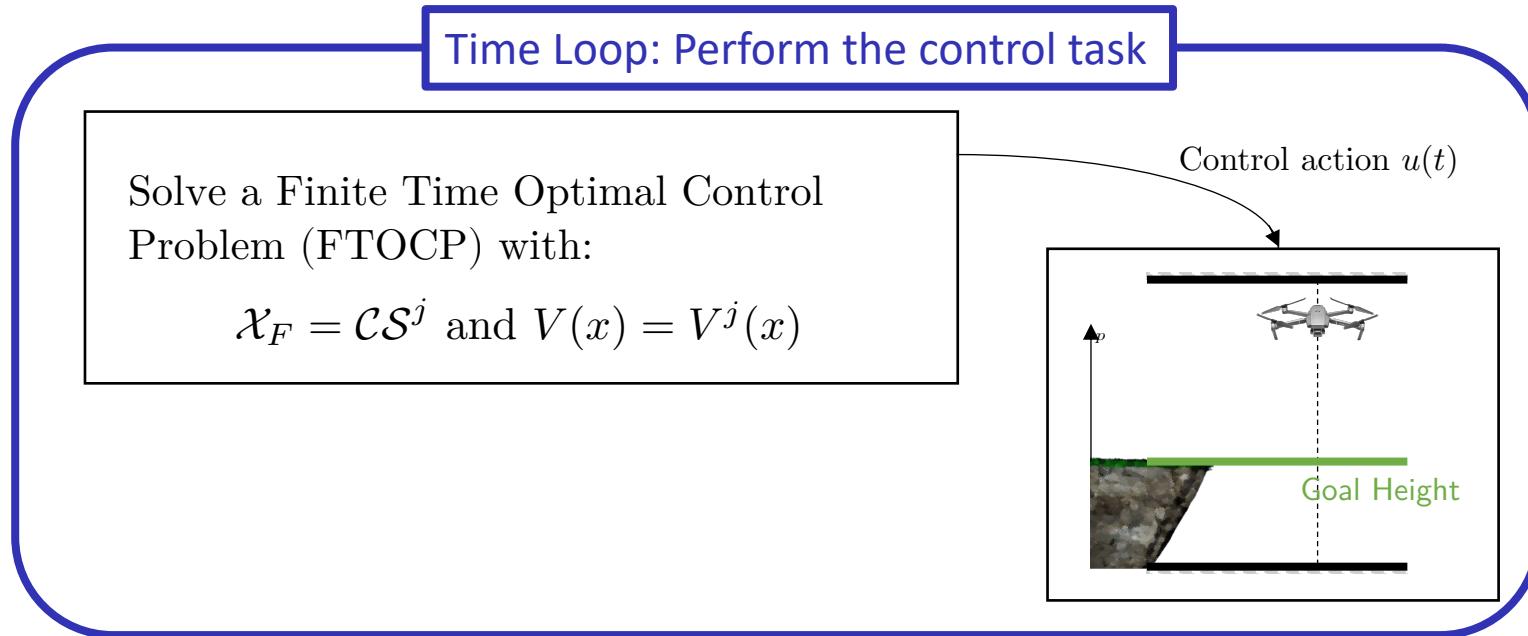
$$\left. \begin{array}{l} J_t^*(x) > 0, \forall x \neq 0 \\ J_t^*(0) = 0 \\ J_t^*(x(t)) < J_t^*(x(t+1)), \forall x(t) \neq 0 \end{array} \right\} \lim_{t \rightarrow \infty} J_t^*(x(t)) = 0$$

To show stability we need to that $J_t^*(x)$ is Lipchitz continuous (can be relaxed to continuity around the origin).

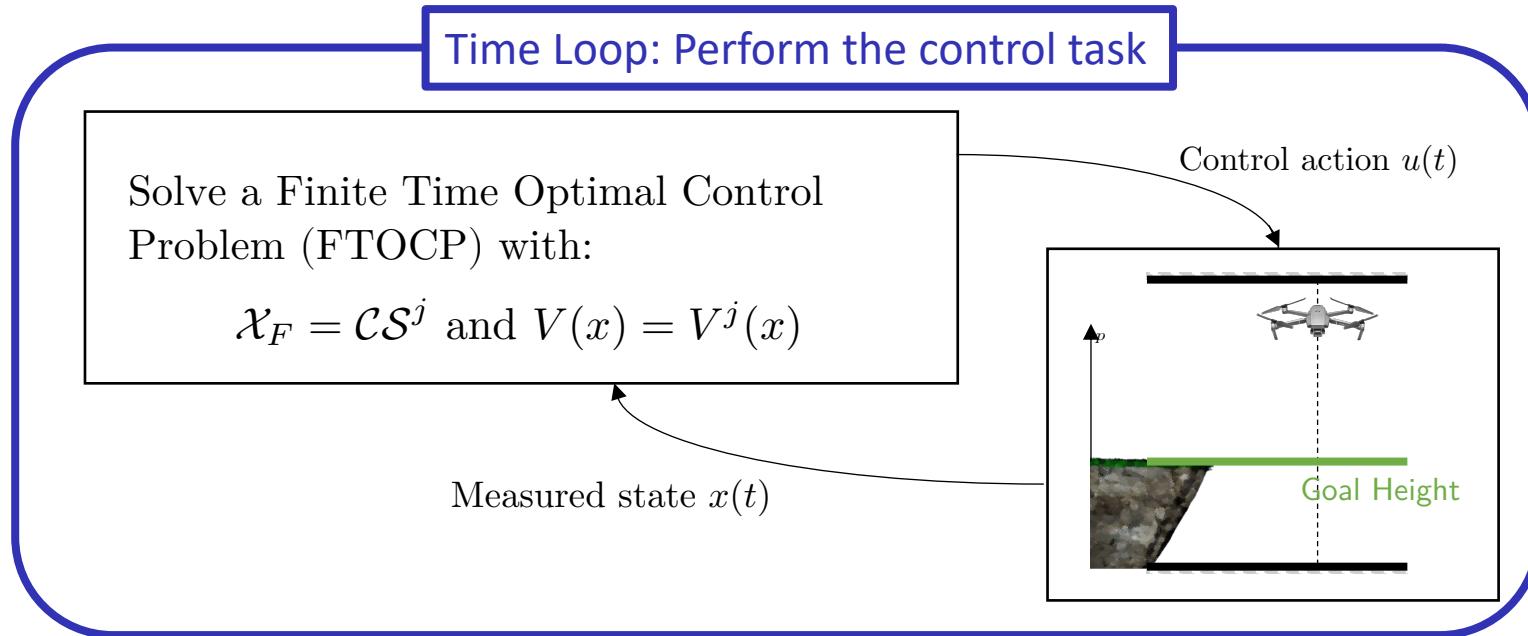
Recap of Lecture #4



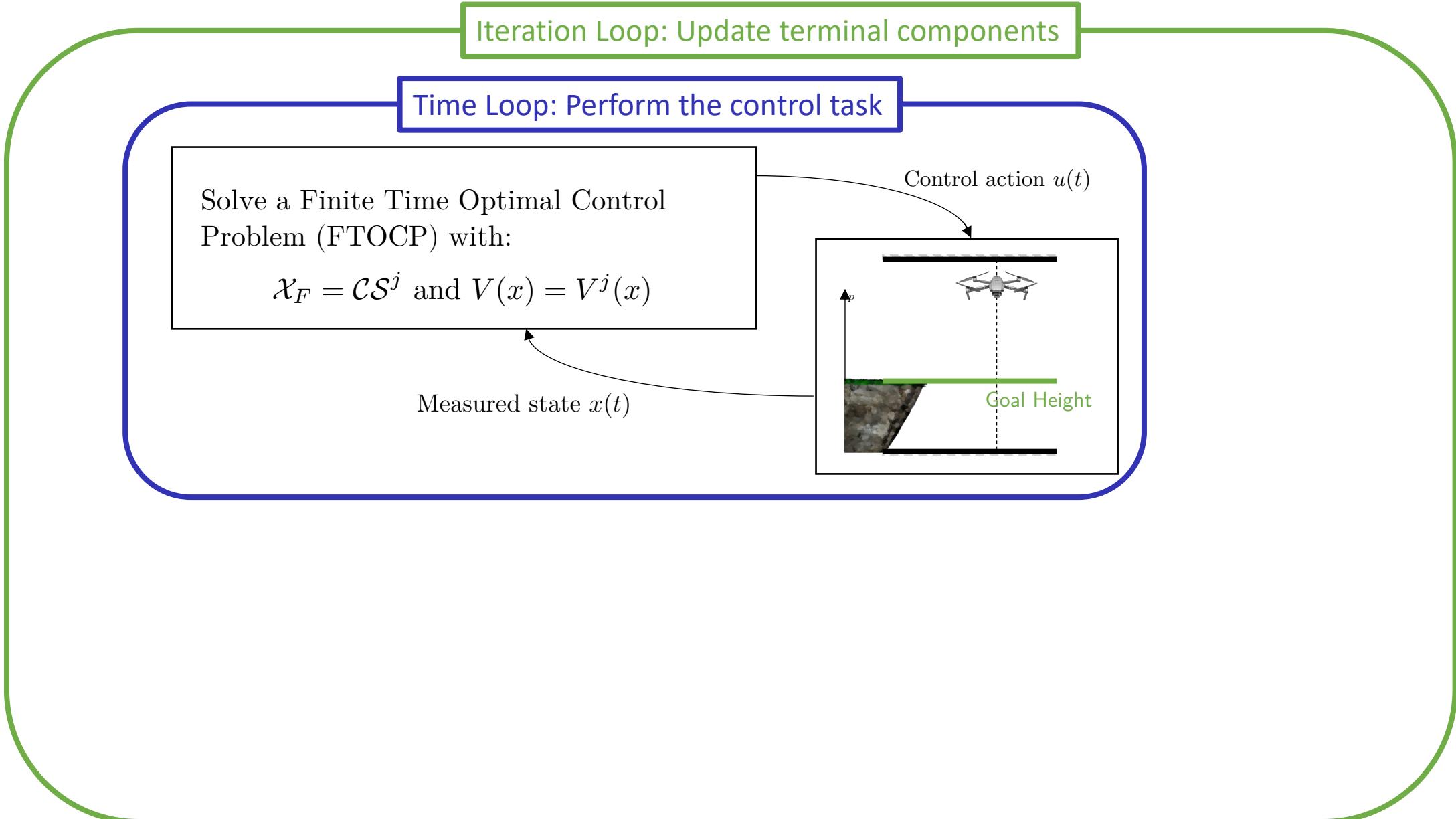
Recap of Lecture #4



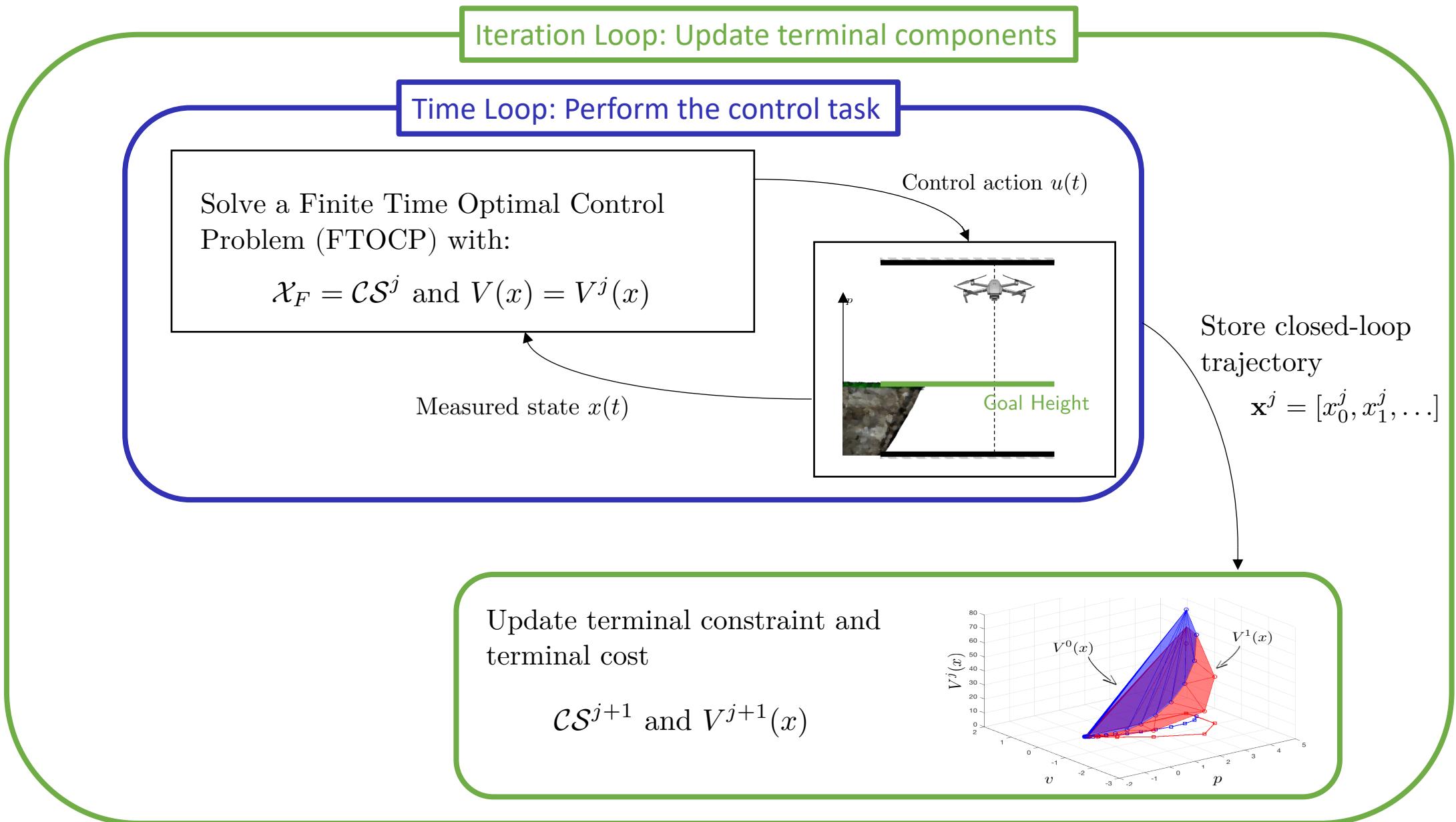
Recap of Lecture #4



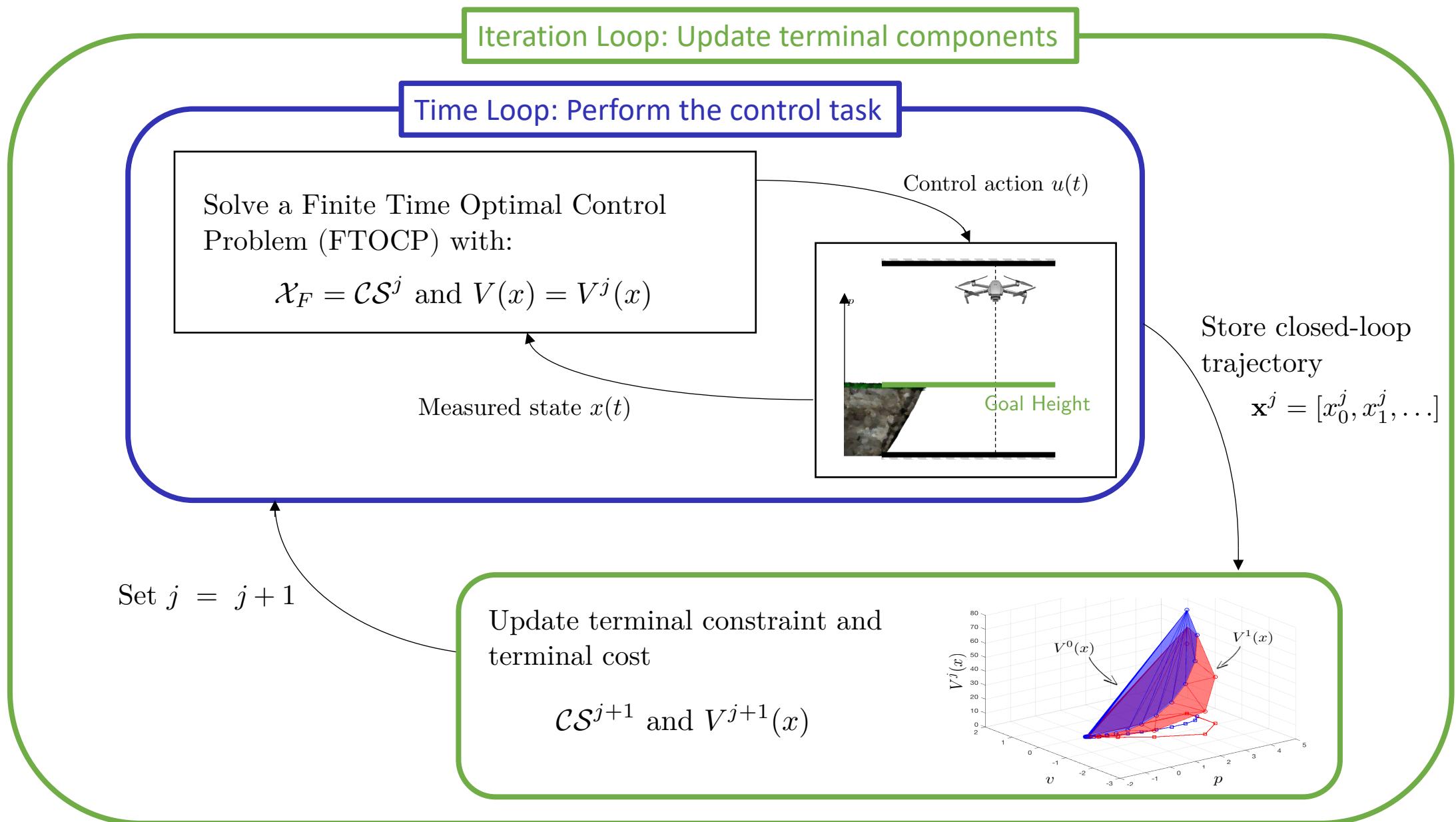
Recap of Lecture #4



Recap of Lecture #4



Recap of Lecture #4



Recap of Lecture #4

Theorem

Let $\mathbf{x}^j = [x_0^j, x_1^j, \dots]$ be the closed-loop trajectory from the starting state x_S at iteration j . Consider sequence $\{\mathbf{x}^j\}$ of closed-loop trajectories and assume that for $c < \infty$ we have that

$$\mathbf{x}^c = \mathbf{x}^{c+1}$$

Then we have that

- ▶ At each iteration state and input constraints are satisfied.
- ▶ The closed-loop cost $J_{0 \rightarrow \infty}^j(x_S)$ is non-increasing, i.e.,

$$J_{0 \rightarrow \infty}^{j+1}(x_S) = \sum_{t=0}^{\infty} h(x_t^{j+1}, u_t^{j+1}) \leq \sum_{t=0}^{\infty} h(x_t^j, u_t^j) = J_{0 \rightarrow \infty}^j(x_S)$$

- ▶ $\mathbf{x}^c = \mathbf{x}^*$, under mild conditions (LICQ holds at each time t)

Optimality Proof Sketch

Define the trajectory at convergence $\mathbf{x}^\infty = [x_0^\infty, x_1^\infty, \dots]$

Optimality Proof Sketch

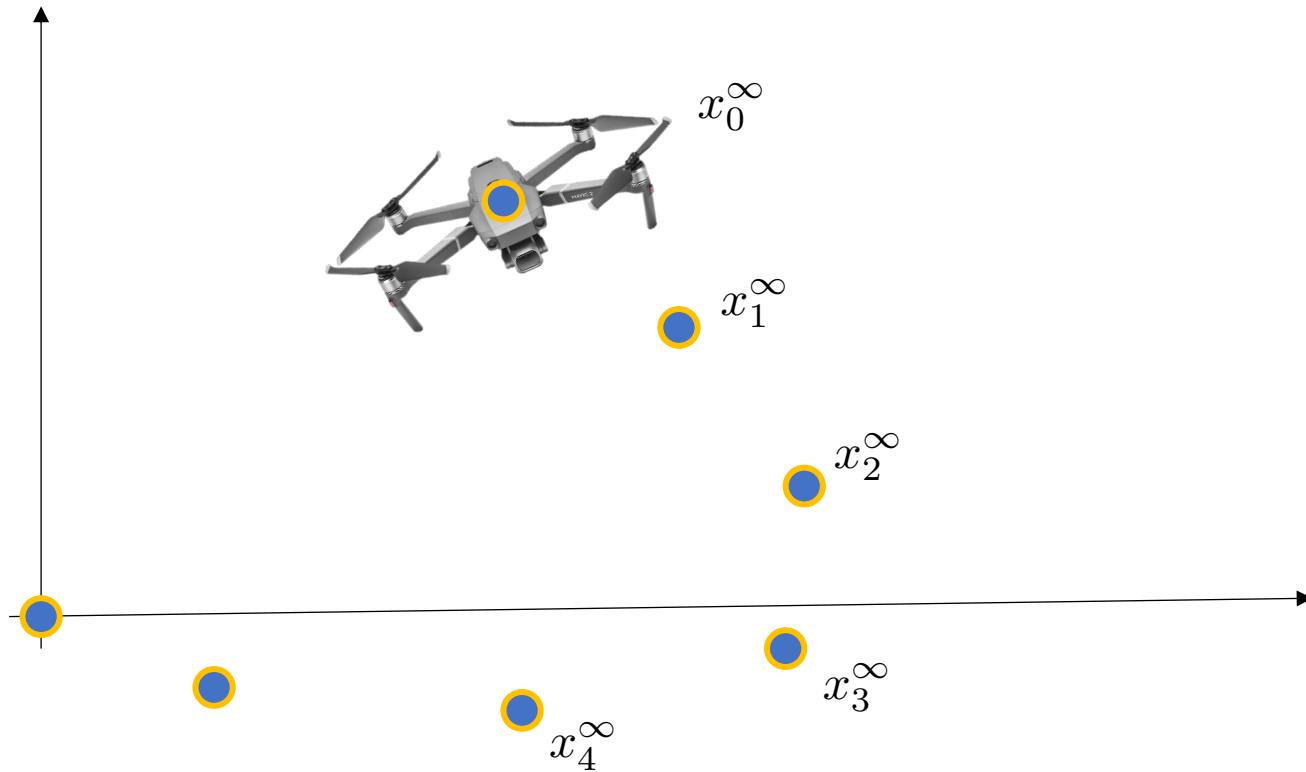
Define the trajectory at convergence $\mathbf{x}^\infty = [x_0^\infty, x_1^\infty, \dots]$

Given an MPC horizon $N = 2$

Optimality Proof Sketch

Define the trajectory at convergence $\mathbf{x}^\infty = [x_0^\infty, x_1^\infty, \dots]$

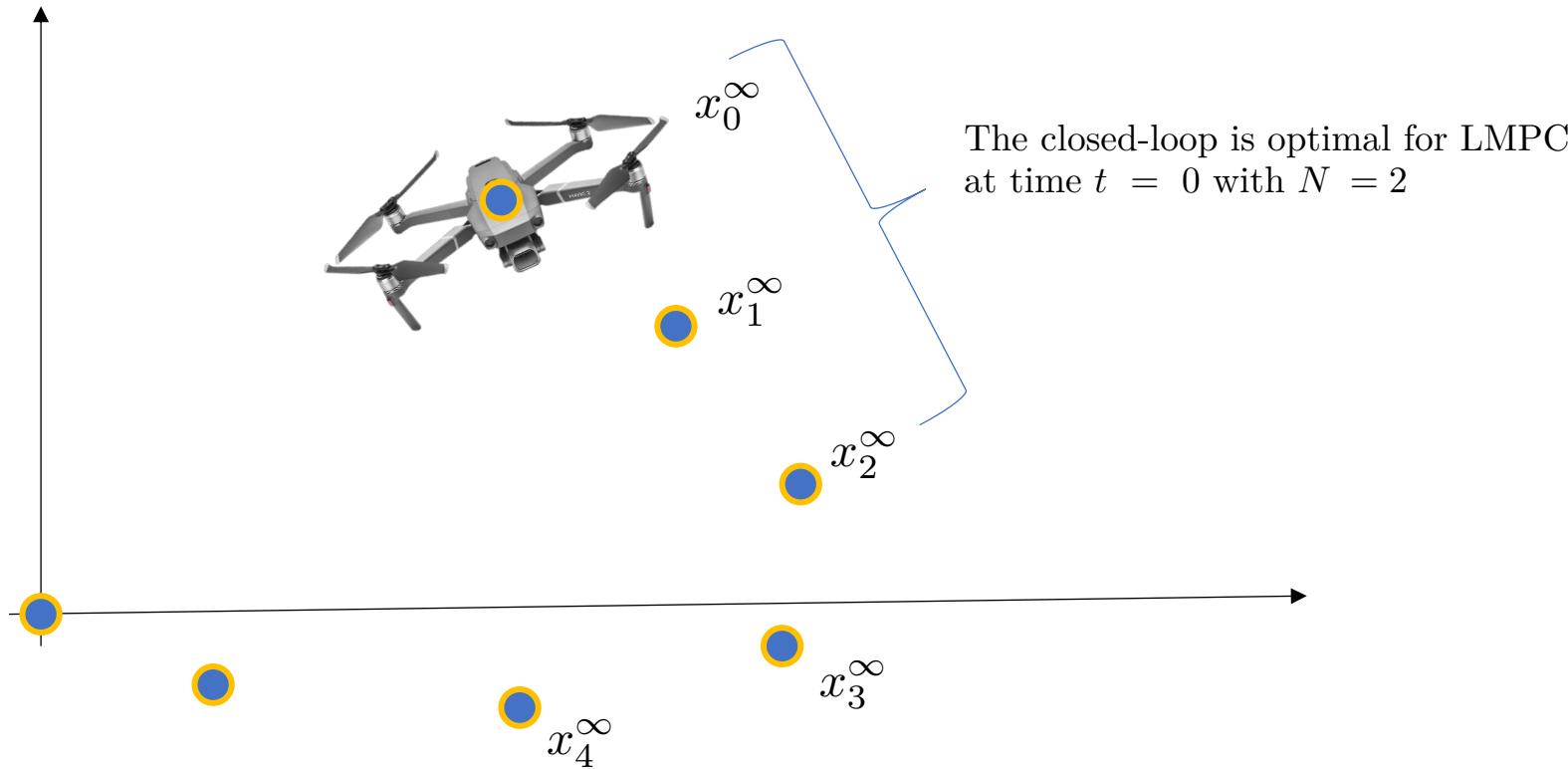
Given an MPC horizon $N = 2$



Optimality Proof Sketch

Define the trajectory at convergence $\mathbf{x}^\infty = [x_0^\infty, x_1^\infty, \dots]$

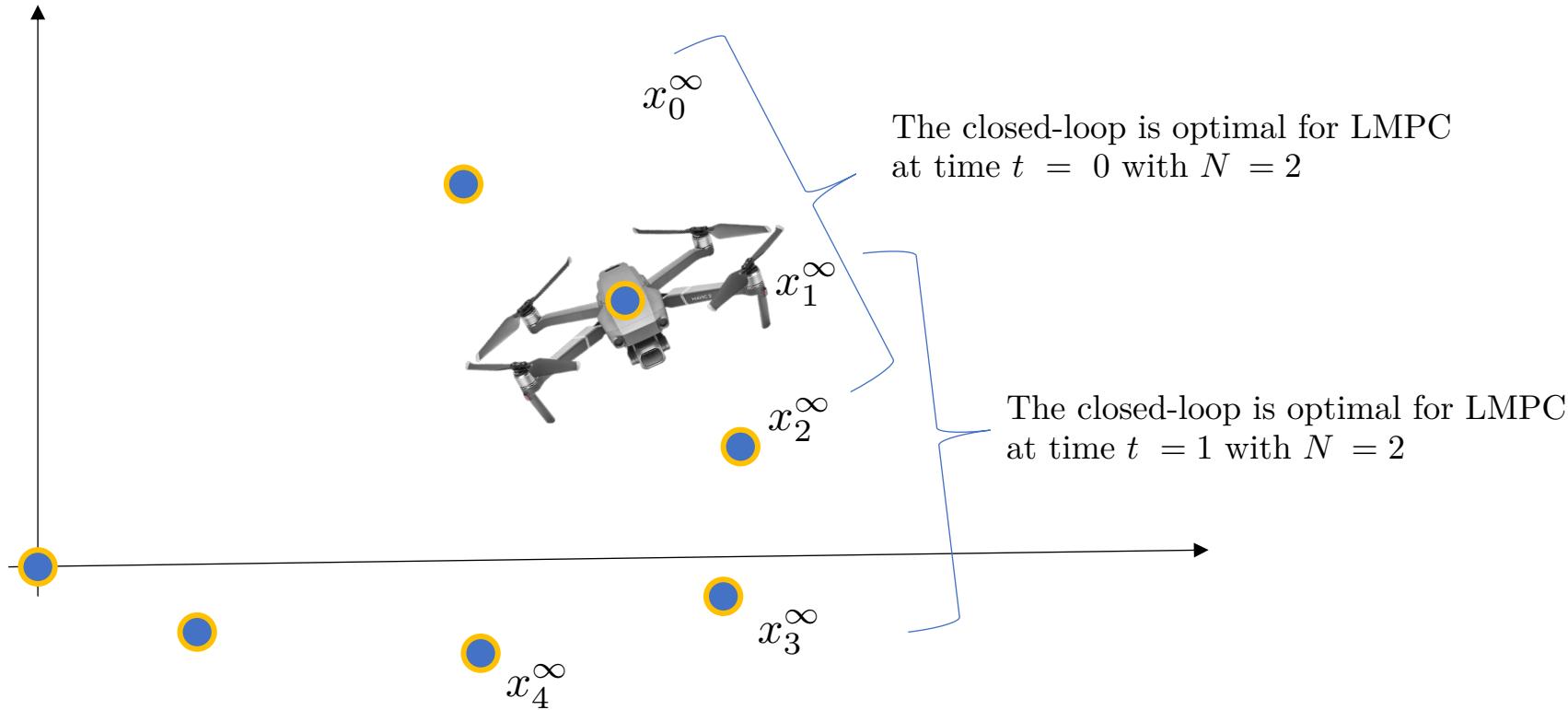
Given an MPC horizon $N = 2$



Optimality Proof Sketch

Define the trajectory at convergence $\mathbf{x}^\infty = [x_0^\infty, x_1^\infty, \dots]$

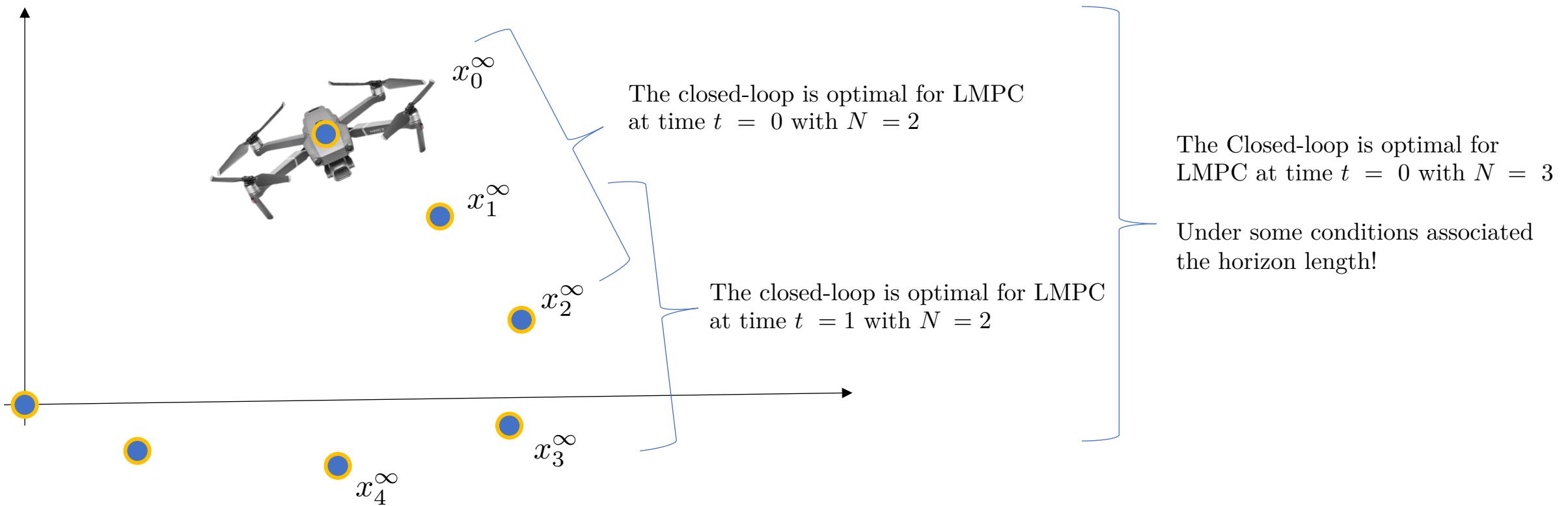
Given an MPC horizon $N = 2$



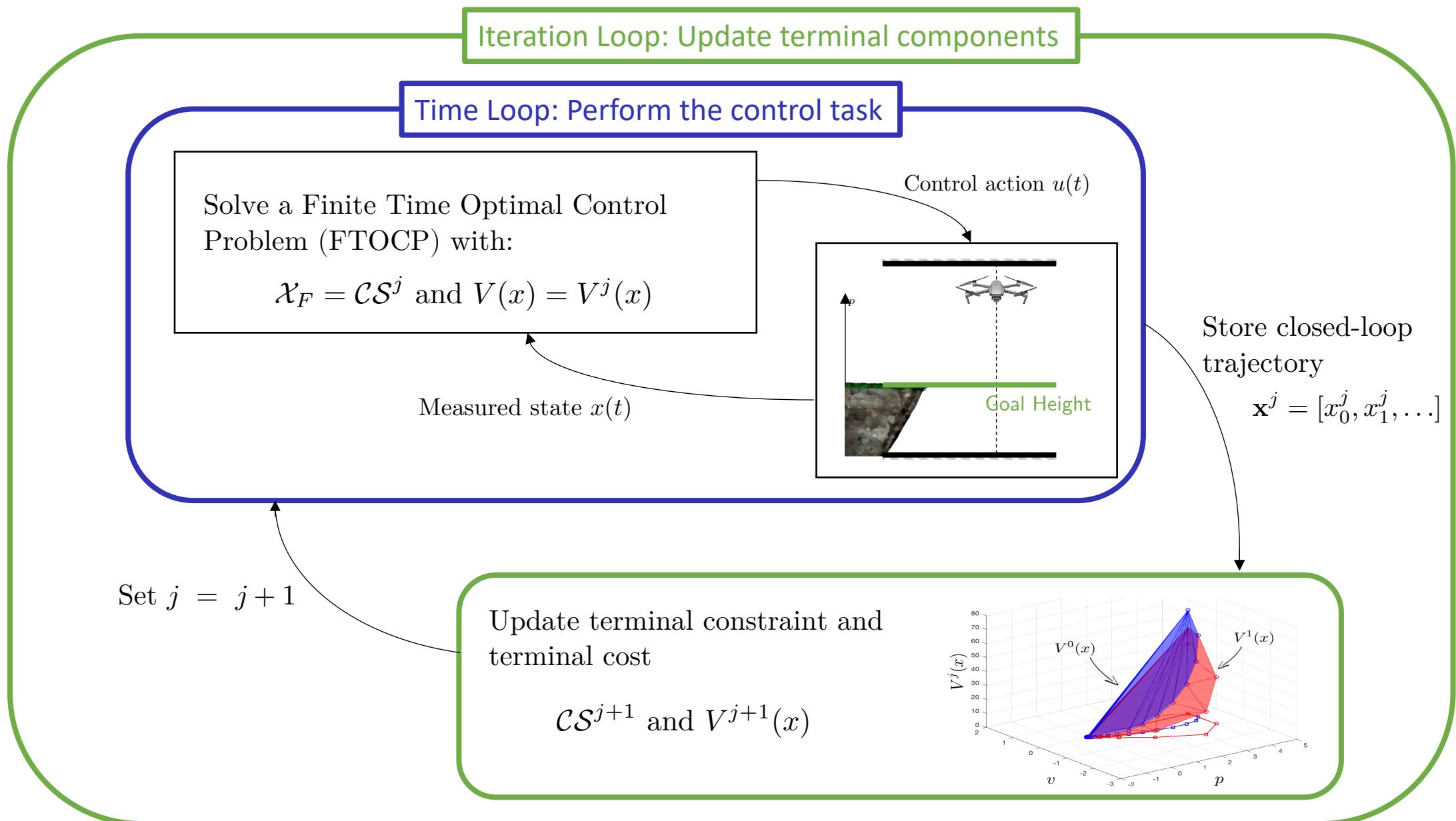
Optimality Proof Sketch

Define the trajectory at convergence $\mathbf{x}^\infty = [x_0^\infty, x_1^\infty, \dots]$

Given an MPC horizon $N = 2$



How to implement the LMPC?



Linear(ized) LMPC

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

$$J_{t \rightarrow t+N}^{\text{LMPC},j}(x_t^j) = \min_{u_{t|t}, \dots, u_{t+N-1|t}} \sum_{k=t}^{t+N-1} h(x_{k|t}, u_{k|t}) + V^{j-1}(x_{t+N|t})$$

s.t.

$$x_{k+1|t} = Ax_{k|t} + Bu_{k|t}, \quad \forall k \in [t, \dots, t+N-1]$$

$$x_{t|t} = x_t^j,$$

$$x_{k|t} \in \mathcal{X}, \quad u_{k|t} \in \mathcal{U}, \quad \forall k \in [t, \dots, t+N-1]$$

$$x_{t+N|t} \in \mathcal{CS}^{j-1}$$

Linear(ized) LMPC

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

$$J_{t \rightarrow t+N}^{\text{LMPC},j}(x_t^j) = \min_{\substack{u_{t|t}, \dots, u_{t+N-1|t} \\ \lambda_0^0, \dots, \lambda_{T_f}^{j-1}}} \sum_{k=t}^{t+N-1} h(x_{k|t}, u_{k|t}) + V^{j-1}(x_{t+N|t})$$

s.t.

$$x_{k+1|t} = Ax_{k|t} + Bu_{k|t}, \quad \forall k \in [t, \dots, t+N-1]$$

$$x_{t|t} = x_t^j,$$

$$x_{k|t} \in \mathcal{X}, \quad u_{k|t} \in \mathcal{U}, \quad \forall k \in [t, \dots, t+N-1]$$

$$x_{t+N|t} = \sum_{i=0}^{j-1} \sum_k x_k^i \lambda_k^i, \quad \sum_{i=0}^{j-1} \sum_k \lambda_k^i = 1, \quad \lambda_k^i \geq 0.$$

$$x_{t+N|t} \in \mathcal{CS}^{j-1}$$

Linear(ized) LMPC

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

$$J_{t \rightarrow t+N}^{\text{LMPC},j}(x_t^j) = \min_{\substack{u_{t|t}, \dots, u_{t+N-1|t} \\ \lambda_0^0, \dots, \lambda_{T_f}^{j-1}}} \sum_{k=t}^{t+N-1} h(x_{k|t}, u_{k|t}) + \sum_{i=0}^{j-1} \sum_k J_k^i(x_k^i) \lambda_k^i$$

s.t.

$$x_{k+1|t} = Ax_{k|t} + Bu_{k|t}, \quad \forall k \in [t, \dots, t+N-1]$$

$$x_{t|t} = x_t^j,$$

$$x_{k|t} \in \mathcal{X}, \quad u_{k|t} \in \mathcal{U}, \quad \forall k \in [t, \dots, t+N-1]$$



$$V^{j-1}(x_{t+N|t})$$

$$x_{t+N|t} = \sum_{i=0}^{j-1} \sum_k x_k^i \lambda_k^i, \quad \sum_{i=0}^{j-1} \sum_k \lambda_k^i = 1, \quad \lambda_k^i \geq 0.$$

$$x_{t+N|t} \in \mathcal{CS}^{j-1}$$

Linear(ized) LMPC

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

$$J_{t \rightarrow t+N}^{\text{LMPC},j}(x_t^j) = \min_{\substack{u_{t|t}, \dots, u_{t+N-1|t} \\ \lambda_0^0, \dots, \lambda_{T_f}^{j-1}}} \sum_{k=t}^{t+N-1} h(x_{k|t}, u_{k|t}) + \sum_{i=0}^{j-1} \sum_k J_k^i(x_k^i) \lambda_k^i$$

s.t.

$$x_{k+1|t} = Ax_{k|t} + Bu_{k|t}, \quad \forall k \in [t, \dots, t+N-1]$$

$$x_{t|t} = x_t^j,$$

$$x_{k|t} \in \mathcal{X}, \quad u_{k|t} \in \mathcal{U}, \quad \forall k \in [t, \dots, t+N-1]$$



$$V^{j-1}(x_{t+N|t})$$

$$x_{t+N|t} = \sum_{i=0}^{j-1} \sum_k x_k^i \lambda_k^i, \quad \sum_{i=0}^{j-1} \sum_k \lambda_k^i = 1, \quad \lambda_k^i \geq 0.$$

$$x_{t+N|t} \in \mathcal{CS}^{j-1}$$

- ▶ Convex optimization problem over inputs and lambdas
- ▶ Safety and performance improvement guarantees still hold (simple proofs as before)
- ▶ Converges to global optimal solution (LICQ required)

Learning MPC for Autonomous Racing

Real-time implementation on the Berkeley Autonomous Race Car (BARC)

Problem Formulation

Minimum Time Control Problem

$$\min_{T, \mathbf{u}} \quad T \quad \text{Control objective}$$

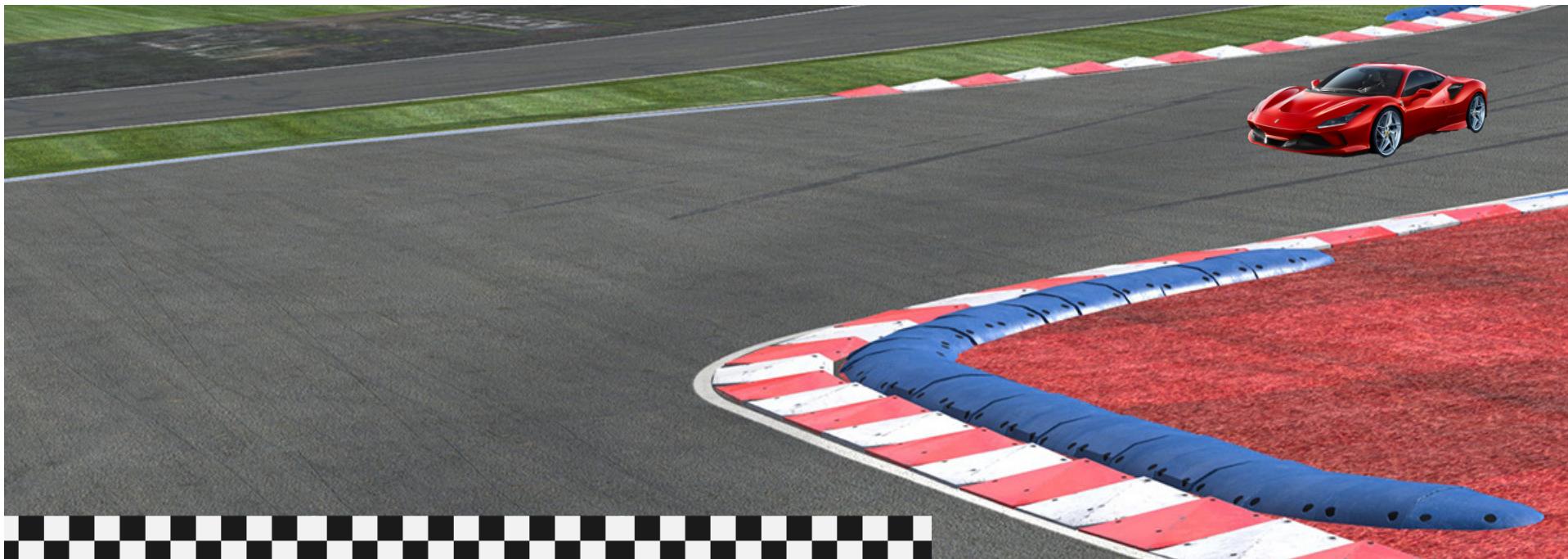
$$x_0 = x_s, \quad x_T = \mathcal{X}_F \quad \text{Start & end position}$$

System dynamics
System constraints

Safety constraints

$$x_{k+1} = f(x_k, u_k), \quad \forall k \in \{0, \dots, T-1\}$$

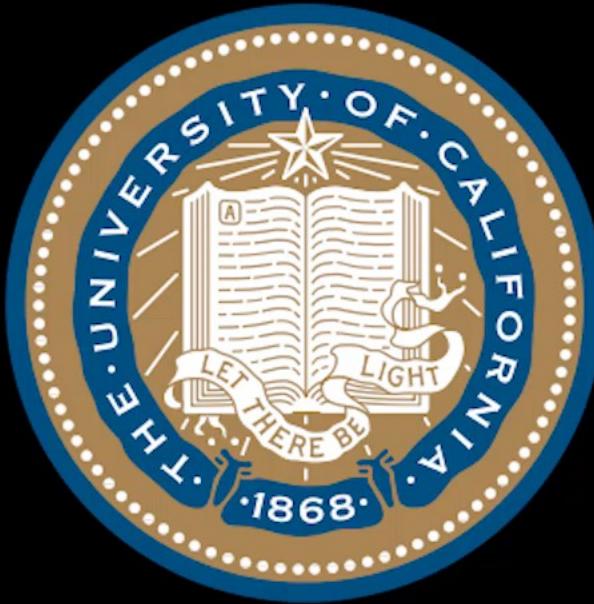
$$x_k \in \mathcal{X}, \quad u_k \in \mathcal{U}, \quad \forall k \in \{0, \dots, T-1\}$$





Learning Model Predictive Controller full-size vehicle experiments

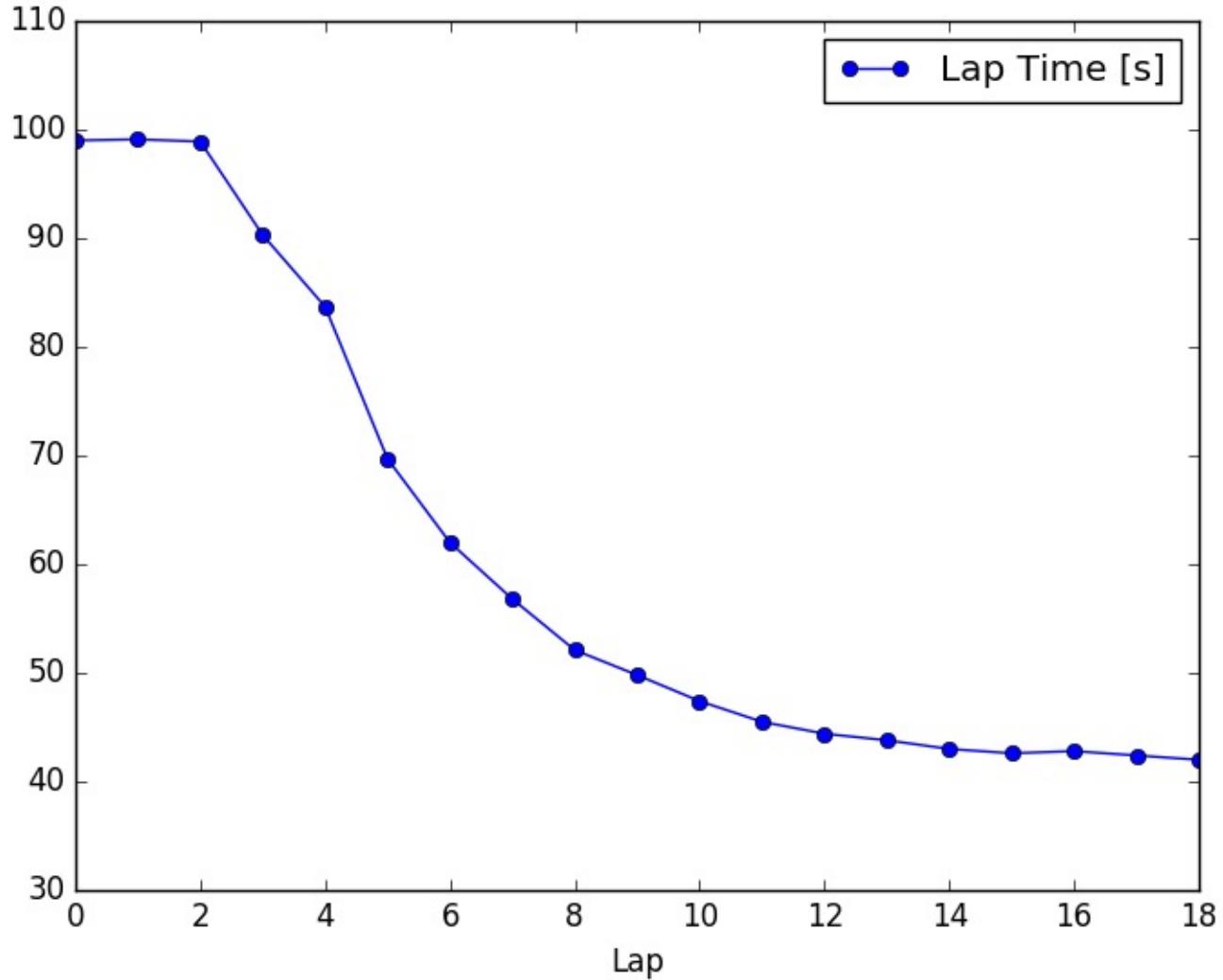
Credits: Siddharth Nair, Nitin Kapania and Ugo Rosolia

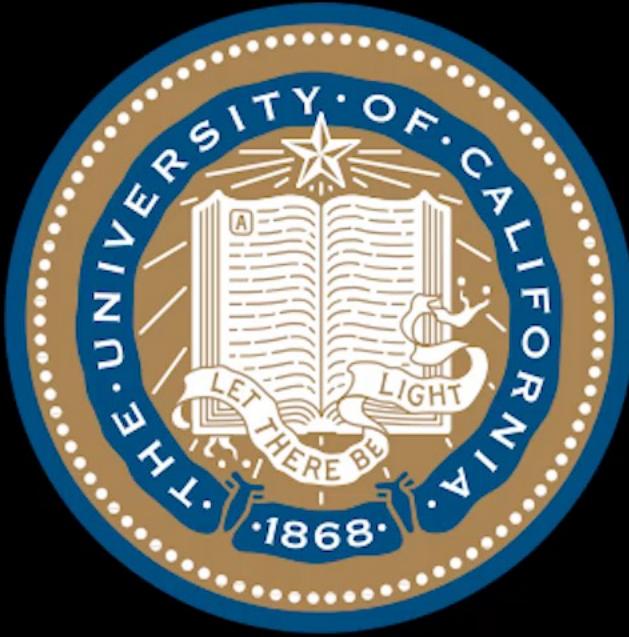


Learning Model Predictive Controller full-size vehicle experiments

Credits: Siddharth Nair, Nitin Kapania and Ugo Rosolia

Lap Time

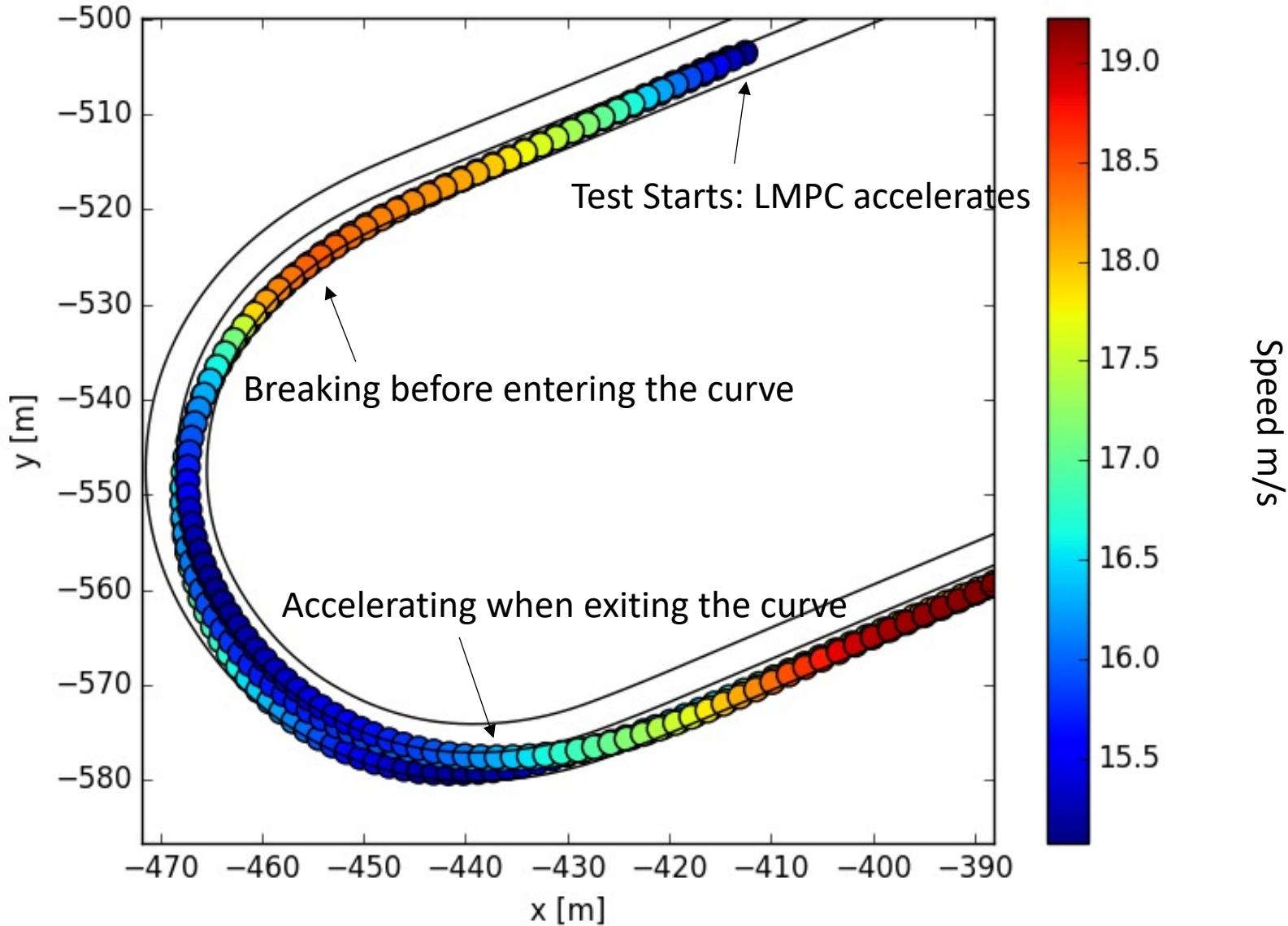




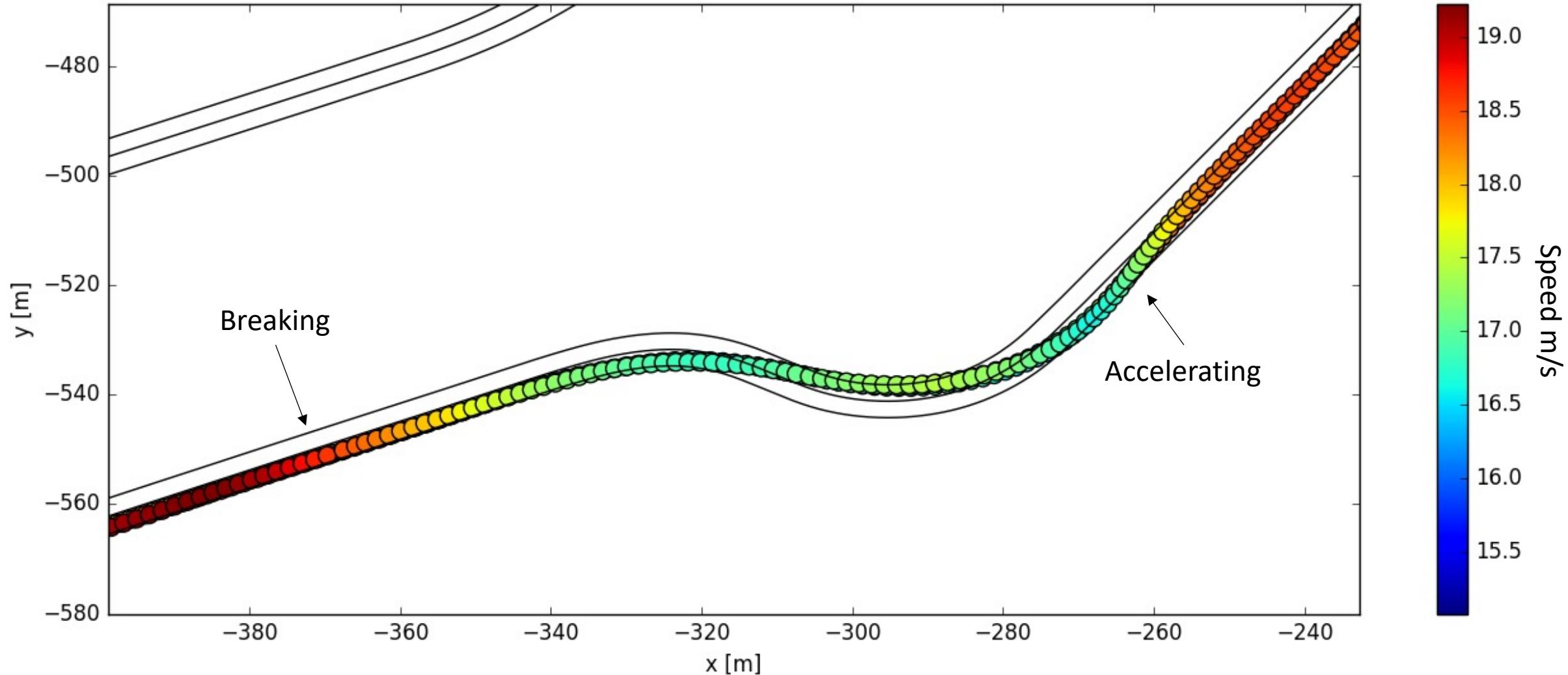
Learning Model Predictive Controller full-size vehicle experiments

Credits: Siddharth Nair, Nitin Kapania and Ugo Rosolia

Velocity Profile at Convergence (Curve 1)



Velocity Profile at Convergence (Chicane)



Problem Formulation

Minimum Time Control Problem

$$\min_{T, \mathbf{u}} \quad T \quad \text{Control objective}$$

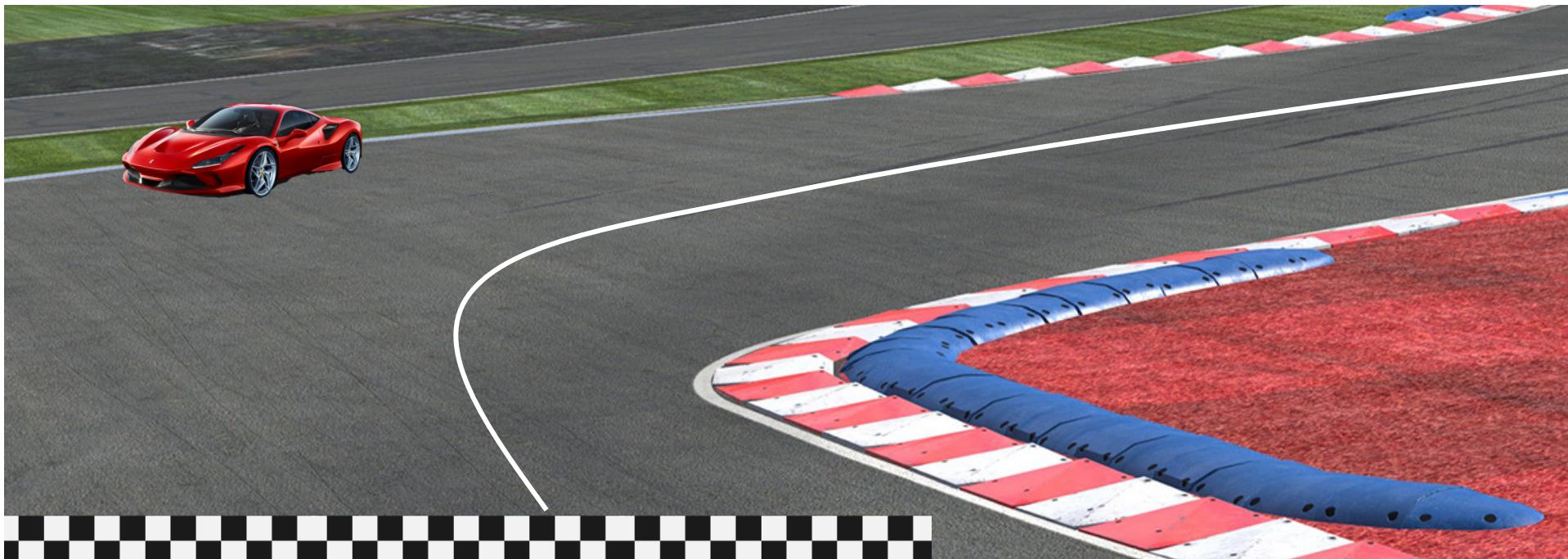
$$x_0 = x_s, \quad x_T = \mathcal{X}_F \quad \text{Start & end position}$$

System dynamics
System constraints

Safety constraints

$$x_{k+1} = f(x_k, u_k), \quad \forall k \in \{0, \dots, T-1\}$$

$$x_k \in \mathcal{X}, \quad u_k \in \mathcal{U}, \quad \forall k \in \{0, \dots, T-1\}$$



Problem Formulation

Minimum Time Control Problem

$$\min_{T, \mathbf{u}} \quad T \quad \text{Control objective}$$

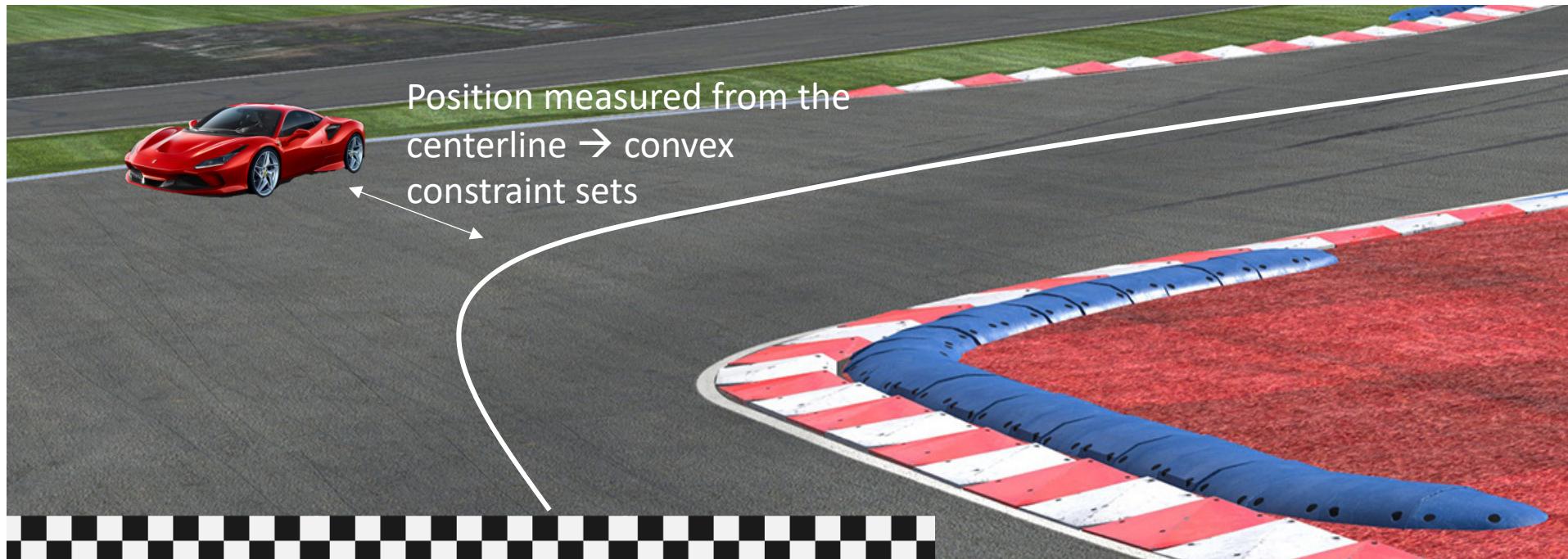
$$x_0 = x_s, \quad x_T = \mathcal{X}_F \quad \text{Start & end position}$$

System dynamics
System constraints

Safety constraints

$$x_{k+1} = f(x_k, u_k), \quad \forall k \in \{0, \dots, T-1\}$$

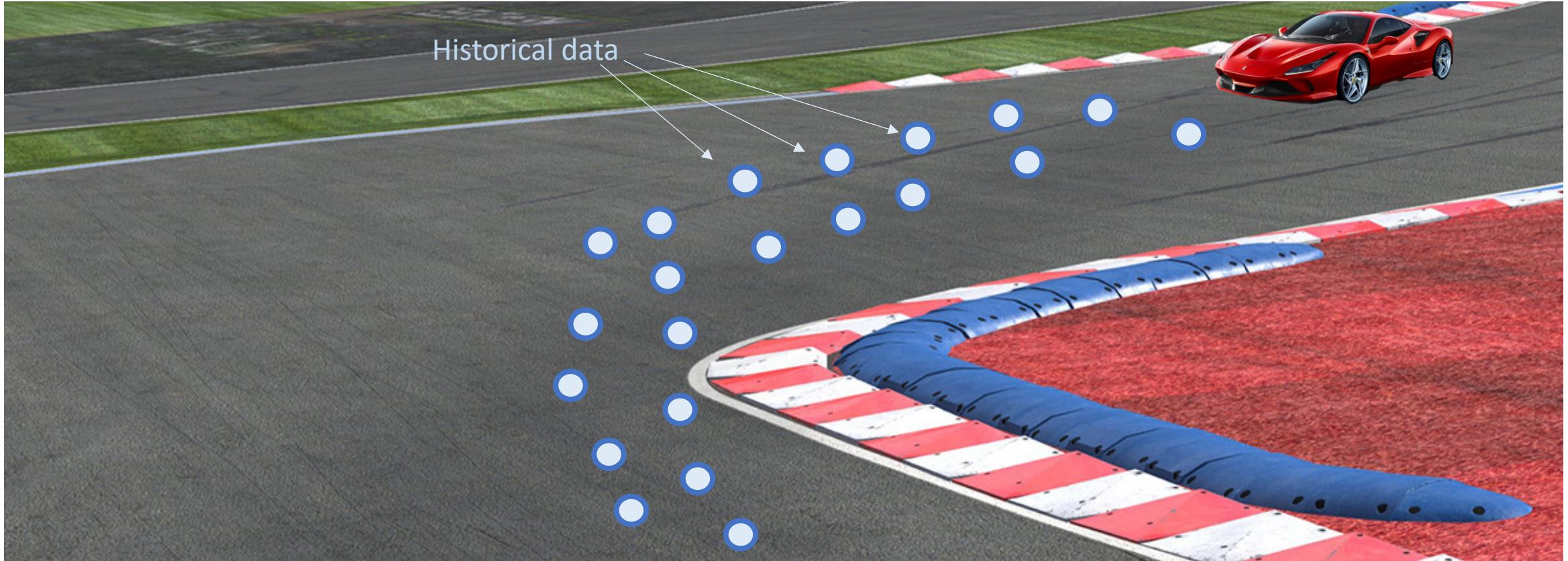
$$x_k \in \mathcal{X}, \quad u_k \in \mathcal{U}, \quad \forall k \in \{0, \dots, T-1\}$$



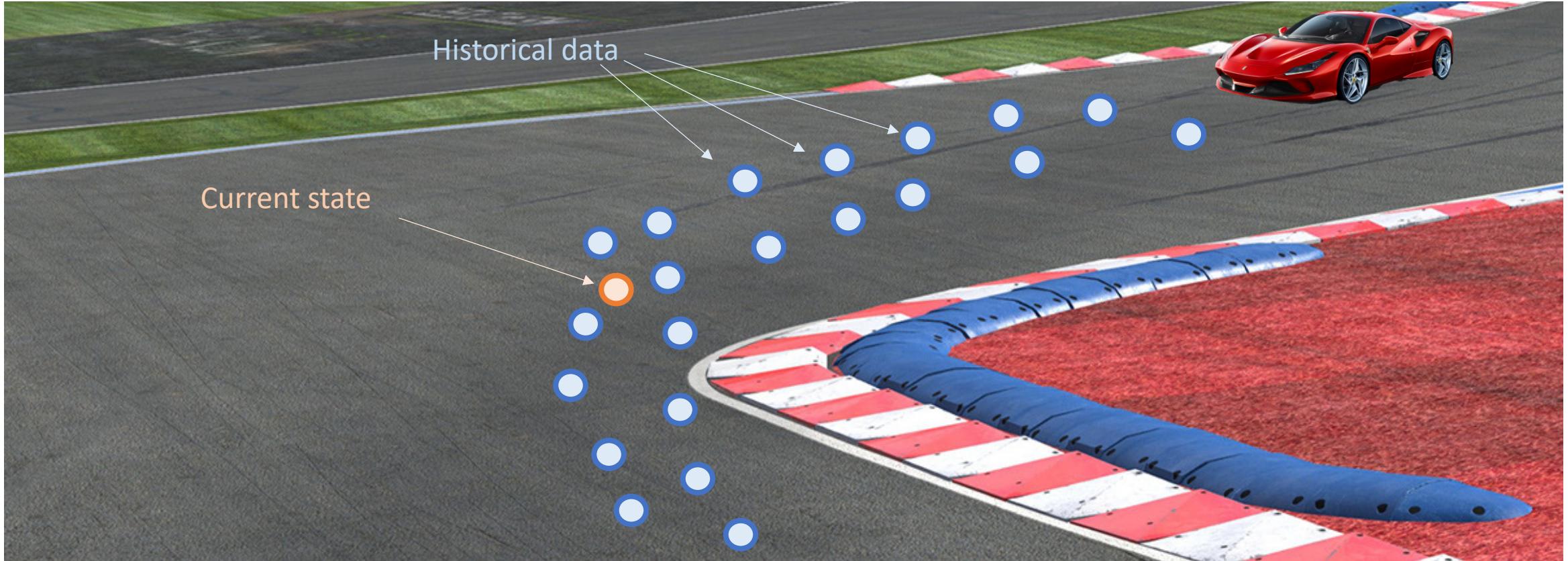
Local Approximations



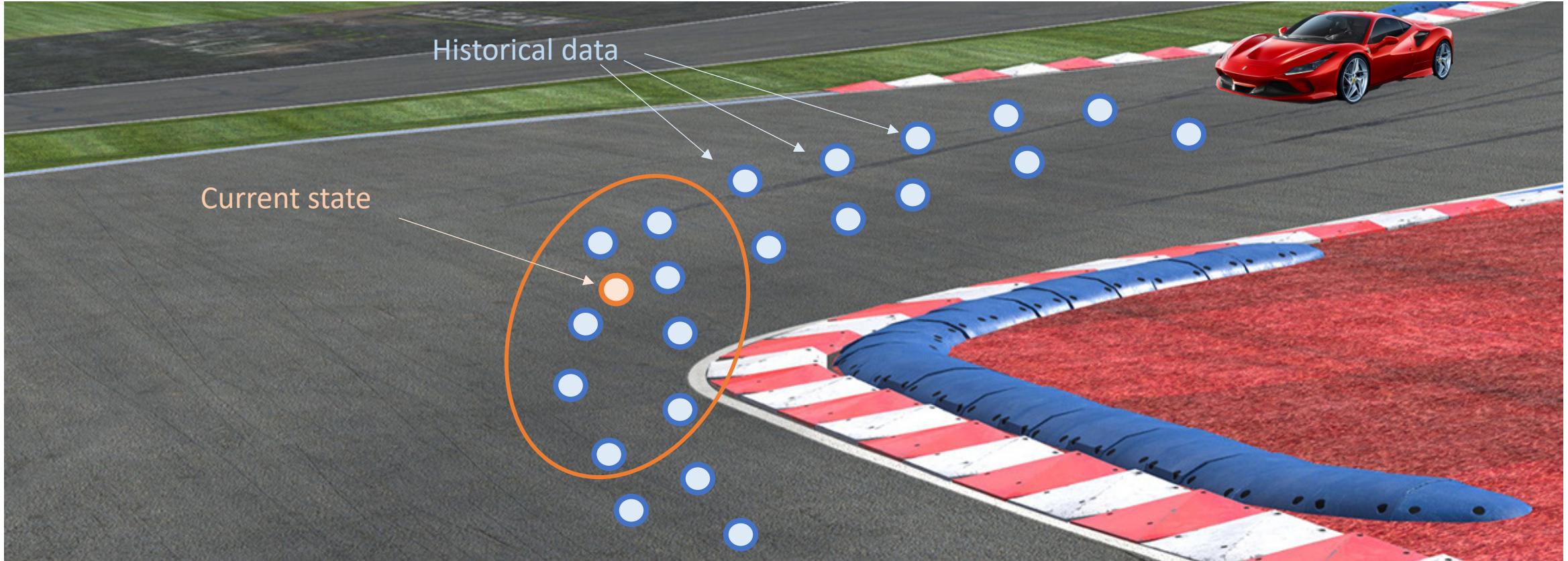
Local Approximations



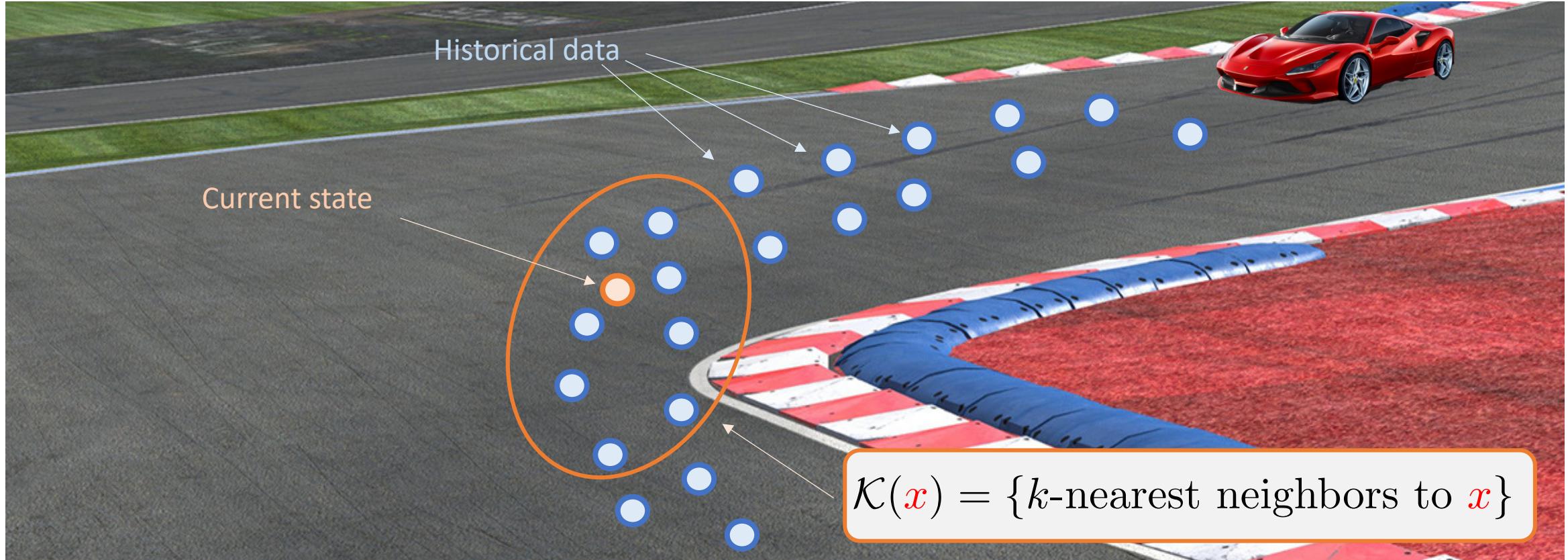
Local Approximations



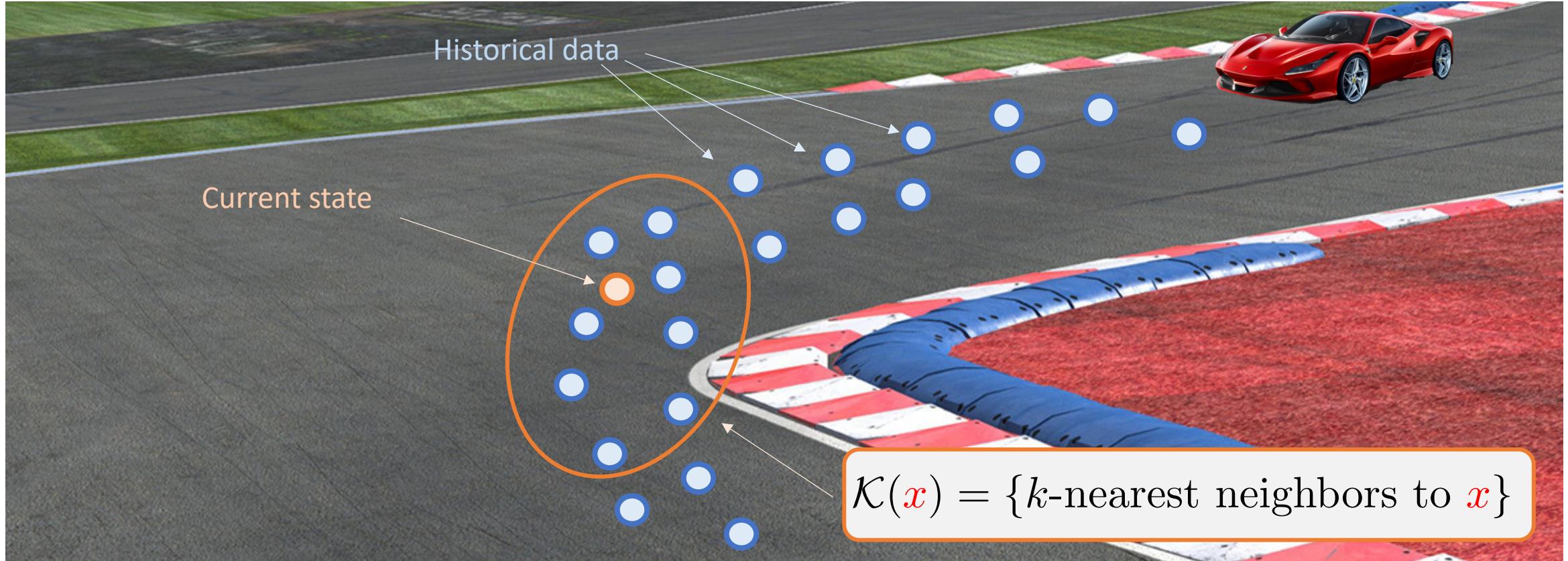
Local Approximations



Local Approximations



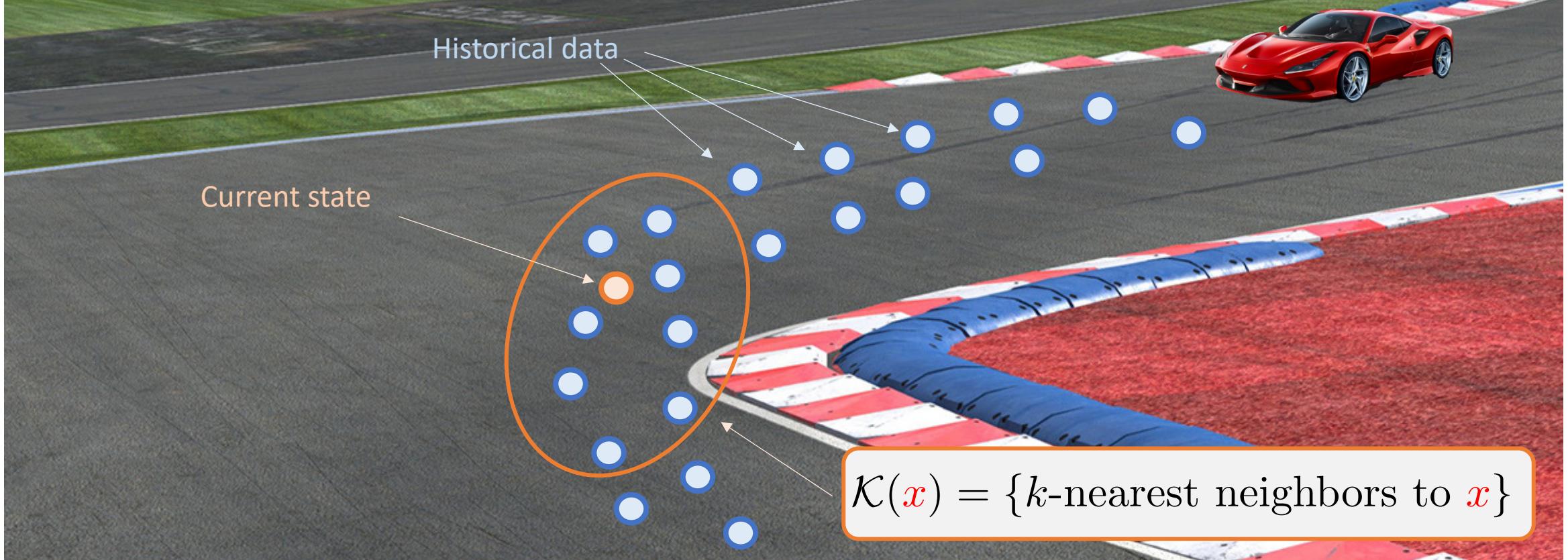
Local Approximations



Local convex safe set approximation:

$$\mathcal{CS}^j(\textcolor{red}{x}) = \text{conv} \left(\cup_{x_t^j \in \mathcal{K}(\textcolor{red}{x})} x_t^j \right)$$

Local Approximations



Local value function approximation:

$$V^j(\bar{x}, \textcolor{red}{x}) = \min_{\lambda_t^i \geq 0} \sum_{x_t^i \in \mathcal{K}^j(\textcolor{red}{x})} J_t^i(x_t^i) \lambda_t^i$$

subject to $\sum_{x_t^i \in \mathcal{K}^j(\textcolor{red}{x})} x_t^i \lambda_t^i = \bar{x}, \sum_i \sum_t \lambda_t^i = 1$

Learning Model Predictive Controller

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

$$J_{t \rightarrow t+N}^{\text{LMPC},j}(x_t^j) = \min_{u_{t|t}^j, \dots, u_{t+N-1|t}^j} \sum_{k=t}^{t+N-1} h(x_{k|t}^j, u_{k|t}^j) + V^{j-1}(x_{t+N|t}^j, x_{t+N}^+)$$

s.t.

$$x_{k+1|t}^j = A_{k|t}^j x_{k|t}^j + B_{k|t}^j u_{k|t}^j + C_{k|t}^j$$

$$x_{t|t}^j = x_t^j,$$

$$x_{k|t}^j \in \mathcal{X}, \quad u_{k|t}^j \in \mathcal{U}, \quad \forall k \in [t, \dots, t+N-1]$$

$$x_{t+N|t}^j \in \mathcal{CS}^{j-1}(x_{t+N}^+),$$

Learning Model Predictive Controller

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

$$J_{t \rightarrow t+N}^{\text{LMPC},j}(x_t^j) = \min_{u_{t|t}^j, \dots, u_{t+N-1|t}^j} \sum_{k=t}^{t+N-1} h(x_{k|t}^j, u_{k|t}^j) + V^{j-1}(x_{t+N|t}^j, x_{t+N}^+)$$

s.t.

$$x_{k+1|t}^j = A_{k|t}^j x_{k|t}^j + B_{k|t}^j u_{k|t}^j + C_{k|t}^j$$

$$x_{t|t}^j = x_t^j,$$

$$x_{k|t}^j \in \mathcal{X}, \quad u_{k|t}^j \in \mathcal{U}, \quad \forall k \in [t, \dots, t+N-1]$$

$$x_{t+N|t}^j \in \mathcal{CS}^{j-1}(x_{t+N}^+),$$

Computed online
using LMPC strategy

where $x_{t+N}^+ = x_{t+N-1|t-1}^*$

Learning Model Predictive Controller

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

$$J_{t \rightarrow t+N}^{\text{LMPC},j}(x_t^j) = \min_{u_{t|t}^j, \dots, u_{t+N-1|t}^j} \sum_{k=t}^{t+N-1} h(x_{k|t}^j, u_{k|t}^j) + V^{j-1}(x_{t+N|t}^j, x_{t+N}^+)$$

s.t.

$$x_{k+1|t}^j = A_{k|t}^j x_{k|t}^j + B_{k|t}^j u_{k|t}^j + C_{k|t}^j$$

$$x_{t|t}^j = x_t^j,$$

$$x_{k|t}^j \in \mathcal{X}, \quad u_{k|t}^j \in \mathcal{U}, \quad \forall k \in [t, \dots, t+N-1]$$

$$x_{t+N|t}^j \in \mathcal{CS}^{j-1}(x_{t+N}^+),$$

Computed online
using LMPC strategy

where $x_{t+N}^+ = x_{t+N-1|t-1}^*$ or $x_{t+N}^+ = \hat{f}(x_{t+N-1|t-1}^*, u_f)$

Learning Model Predictive Controller

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

$$J_{t \rightarrow t+N}^{\text{LMPC},j}(x_t^j) = \min_{u_{t|t}^j, \dots, u_{t+N-1|t}^j} \sum_{k=t}^{t+N-1} h(x_{k|t}^j, u_{k|t}^j) + V^{j-1}(x_{t+N|t}^j, x_{t+N}^+)$$

s.t.

$$x_{k+1|t}^j = A_{k|t}^j x_{k|t}^j + B_{k|t}^j u_{k|t}^j + C_{k|t}^j$$

$$x_{t|t}^j = x_t^j,$$

$$x_{k|t}^j \in \mathcal{X}, \quad u_{k|t}^j \in \mathcal{U}, \quad \forall k \in [t, \dots, t+N-1]$$

$$x_{t+N|t}^j \in \mathcal{CS}^{j-1}(x_{t+N}^+),$$

Computed
online

Computed online
using LMPC strategy

Learning Model Predictive Controller

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

$$J_{t \rightarrow t+N}^{\text{LMPC},j}(x_t^j) = \min_{u_{t|t}^j, \dots, u_{t+N-1|t}^j} \sum_{k=t}^{t+N-1} h(x_{k|t}^j, u_{k|t}^j) + V^{j-1}(x_{t+N|t}^j, x_{t+N}^+)$$

s.t.

$$x_{k+1|t}^j = A_{k|t}^j x_{k|t}^j + B_{k|t}^j u_{k|t}^j + C_{k|t}^j$$

$$x_{t|t}^j = x_t^j,$$

$$x_{k|t}^j \in \mathcal{X}, \quad u_{k|t}^j \in \mathcal{U}, \quad \forall k \in [t, \dots, t+N-1]$$

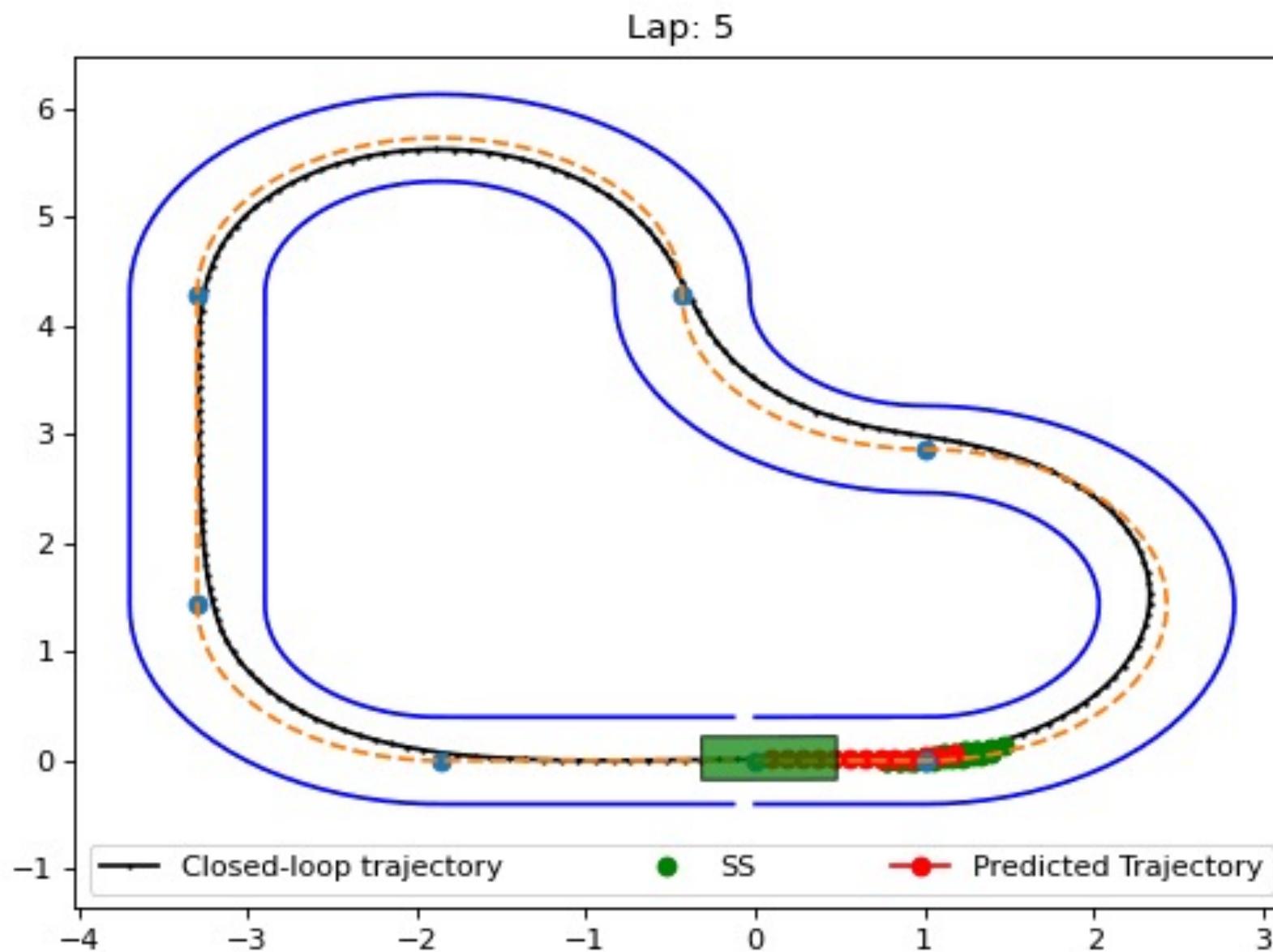
$$x_{t+N|t}^j \in \mathcal{CS}^{j-1}(x_{t+N}^+),$$

Computed
online

Computed online
using LMPC strategy

Then apply to the system the control input $u_t^j = u_{t|t}^{*,j}$

Safe Set and Value Function Approximations



System ID in Autonomous Racing

- Nonlinear Dynamical System,

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

System ID in Autonomous Racing

- Nonlinear Dynamical System,

$$\ddot{x} = \dot{y}\dot{\psi} + \frac{1}{m} \sum_i F_{x_i}$$

$$\ddot{y} = -\dot{x}\dot{\psi} + \frac{1}{m} \sum_i F_{y_i}$$

$$\ddot{\psi} = \frac{1}{I_z} (a(F_{y_{1,2}}) - b(F_{y_{2,3}}) + c(-F_{x_{1,3}} + F_{x_{2,4}}))$$

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

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

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

Linearization around predicted trajectory

System ID in Autonomous Racing

- ▶ Nonlinear Dynamical System,

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

Dynamic Equations
Kinematic Equations

- ▶ Identifying the Dynamical System

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

Linearization around predicted trajectory

System ID in Autonomous Racing

- Nonlinear Dynamical System,

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

Dynamic Equations
Kinematic Equations

- Identifying the Dynamical System

Local Linear Regression

$$x_{k+1|t}^j = \begin{bmatrix} \dot{x}_{k+1|t} \\ \dot{y}_{k+1|t} \\ \ddot{\psi} \\ \psi_{k+1|t} \\ X_{k+1|t} \\ Y_{k+1|t} \end{bmatrix} = \left[\begin{array}{c} \boxed{\arg \min \sum_{i,s} K(x_{k|t}^j - x_s^i) || \Lambda_y \begin{bmatrix} x_{k|t}^j \\ u_{k|t}^j \\ 1 \end{bmatrix} - y_{s+1}^i ||}, \forall y \in \{\dot{x}, \dot{y}, \ddot{\psi}\} \\ \boxed{\text{Linearized Kinematics}} \\ \boxed{\text{Linearized Kinematics}} \\ \boxed{\text{Linearized Kinematics}} \end{array} \right] x_{k|t}^j + \left[\begin{array}{c} \boxed{\text{Linearized Kinematics}} \\ \boxed{\text{Linearized Kinematics}} \\ \boxed{\text{Linearized Kinematics}} \\ \boxed{\text{Linearized Kinematics}} \end{array} \right] \begin{bmatrix} u_{k|t}^j \\ 1 \end{bmatrix}$$

Linearization around predicted trajectory

System Identification – Design Steps

Identifying the Dynamical System

Local Linear Regression

$$x_{k+1|t}^j = \begin{bmatrix} \dot{x}_{k+1|t} \\ \dot{y}_{k+1|t} \\ \ddot{\psi} \\ \psi_{k+1|t} \\ X_{k+1|t} \\ Y_{k+1|t} \end{bmatrix} = \left[\begin{array}{c} \arg \min \sum_{i,s} K(x_{k|t}^j - x_s^i) || \Lambda_y \begin{bmatrix} x_{k|t}^j \\ u_{k|t}^j \\ 1 \end{bmatrix} - y_{s+1}^i ||, \forall y \in \{\dot{x}, \dot{y}, \ddot{\psi}\} \\ \hline \text{Linearized Kinematics} \\ \text{Linearized Kinematics} \\ \text{Linearized Kinematics} \end{array} \right] x_{k|t}^j + \left[\begin{array}{c} \hline \text{Linearized Kinematics} \\ \text{Linearized Kinematics} \\ \text{Linearized Kinematics} \\ \text{Linearized Kinematics} \end{array} \right] \begin{bmatrix} u_{k|t}^j \\ 1 \end{bmatrix}$$

Linearization around predicted trajectory

Implementation Details

System Identification – Design Steps

Identifying the Dynamical System

Local Linear Regression

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

Linearization around predicted trajectory

Implementation Details

- ▶ Compute trajectory to linearize around from previous optimal inputs and stored data

System Identification – Design Steps

Identifying the Dynamical System

Local Linear Regression

$$x_{k+1|t}^j = \begin{bmatrix} \dot{x}_{k+1|t} \\ \dot{y}_{k+1|t} \\ \ddot{\psi} \\ \psi_{k+1|t} \\ X_{k+1|t} \\ Y_{k+1|t} \end{bmatrix} = \left[\begin{array}{c} \arg \min \sum_{i,s} K(x_{k|t}^j - x_s^i) || \Lambda_y \begin{bmatrix} x_{k|t}^j \\ u_{k|t}^j \\ 1 \end{bmatrix} - y_{s+1}^i ||, \forall y \in \{\dot{x}, \dot{y}, \ddot{\psi}\} \\ \hline \text{Linearized Kinematics} \\ \text{Linearized Kinematics} \\ \text{Linearized Kinematics} \end{array} \right] x_{k|t}^j + \left[\begin{array}{c} \hline \text{Linearized Kinematics} \\ \text{Linearized Kinematics} \\ \text{Linearized Kinematics} \\ \text{Linearized Kinematics} \end{array} \right] \begin{bmatrix} u_{k|t}^j \\ 1 \end{bmatrix}$$

Linearization around predicted trajectory

Implementation Details

- ▶ Compute trajectory to linearize around from previous optimal inputs and stored data
- ▶ Enforce model-based sparsity in local linear regression

System Identification – Design Steps

Identifying the Dynamical System

Local Linear Regression

$$x_{k+1|t}^j = \begin{bmatrix} \dot{x}_{k+1|t} \\ \dot{y}_{k+1|t} \\ \ddot{\psi} \\ \psi_{k+1|t} \\ X_{k+1|t} \\ Y_{k+1|t} \end{bmatrix} = \left[\begin{array}{c} \arg \min \sum_{i,s} K(x_{k|t}^j - x_s^i) || \Lambda_y \begin{bmatrix} x_{k|t}^j \\ u_{k|t}^j \\ 1 \end{bmatrix} - y_{s+1}^i ||, \forall y \in \{\dot{x}, \dot{y}, \ddot{\psi}\} \\ \hline \text{Linearized Kinematics} \\ \text{Linearized Kinematics} \\ \text{Linearized Kinematics} \end{array} \right] x_{k|t}^j + \left[\begin{array}{c} \hline \text{Linearized Kinematics} \\ \text{Linearized Kinematics} \\ \text{Linearized Kinematics} \\ \text{Linearized Kinematics} \end{array} \right] \begin{bmatrix} u_{k|t}^j \\ 1 \end{bmatrix}$$

Linearization around predicted trajectory

Implementation Details

- ▶ Compute trajectory to linearize around from previous optimal inputs and stored data
- ▶ Enforce model-based sparsity in local linear regression
- ▶ Perform local linear regression at the linearization points using a subset of the stored data

System Identification – Design Steps

Identifying the Dynamical System

Local Linear Regression

$$x_{k+1|t}^j = \begin{bmatrix} \dot{x}_{k+1|t} \\ \dot{y}_{k+1|t} \\ \ddot{\psi} \\ \psi_{k+1|t} \\ X_{k+1|t} \\ Y_{k+1|t} \end{bmatrix} = \left[\begin{array}{c} \arg \min \sum_{i,s} K(x_{k|t}^j - x_s^i) || \Lambda_y \begin{bmatrix} x_{k|t}^j \\ u_{k|t}^j \\ 1 \end{bmatrix} - y_{s+1}^i ||, \forall y \in \{\dot{x}, \dot{y}, \ddot{\psi}\} \\ \hline \text{Linearized Kinematics} \\ \text{Linearized Kinematics} \\ \text{Linearized Kinematics} \end{array} \right] x_{k|t}^j + \left[\begin{array}{c} \hline \text{Linearized Kinematics} \\ \text{Linearized Kinematics} \\ \text{Linearized Kinematics} \\ \text{Linearized Kinematics} \end{array} \right] \begin{bmatrix} u_{k|t}^j \\ 1 \end{bmatrix}$$

Linearization around predicted trajectory

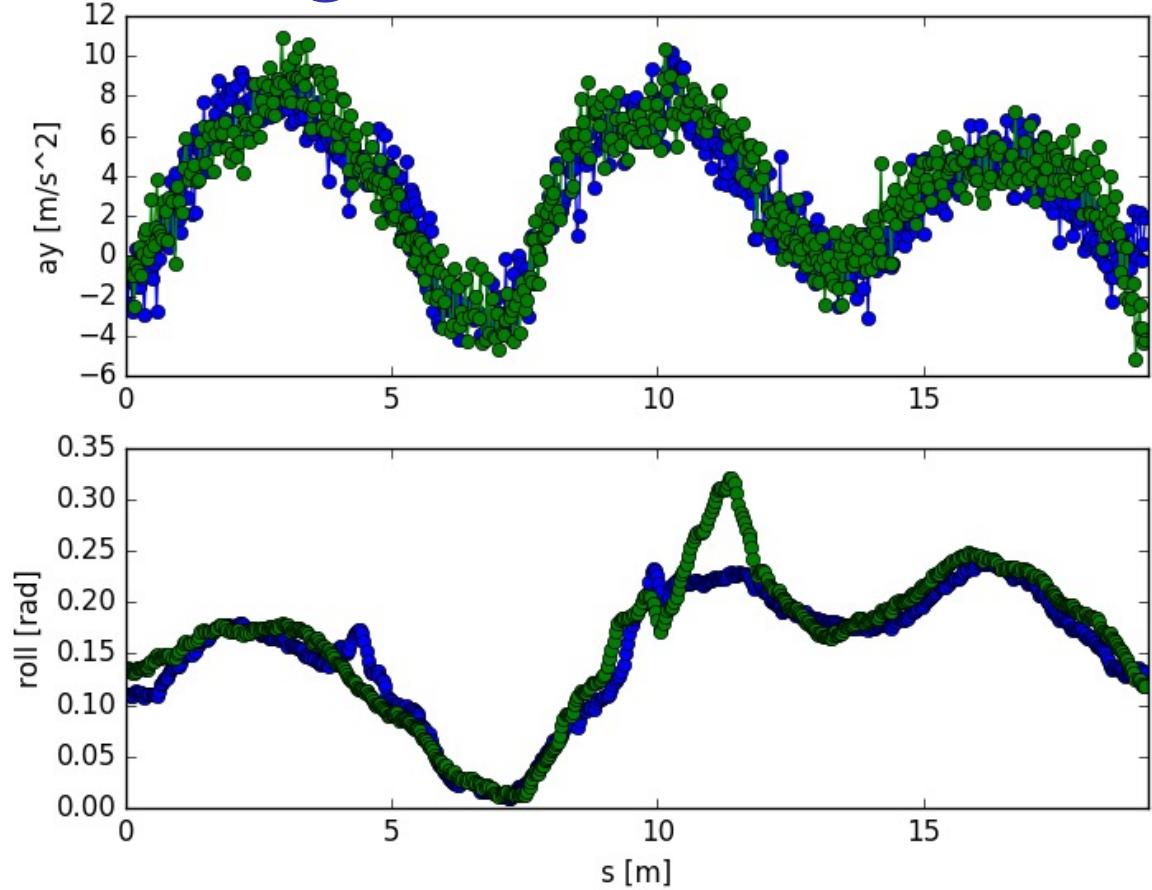
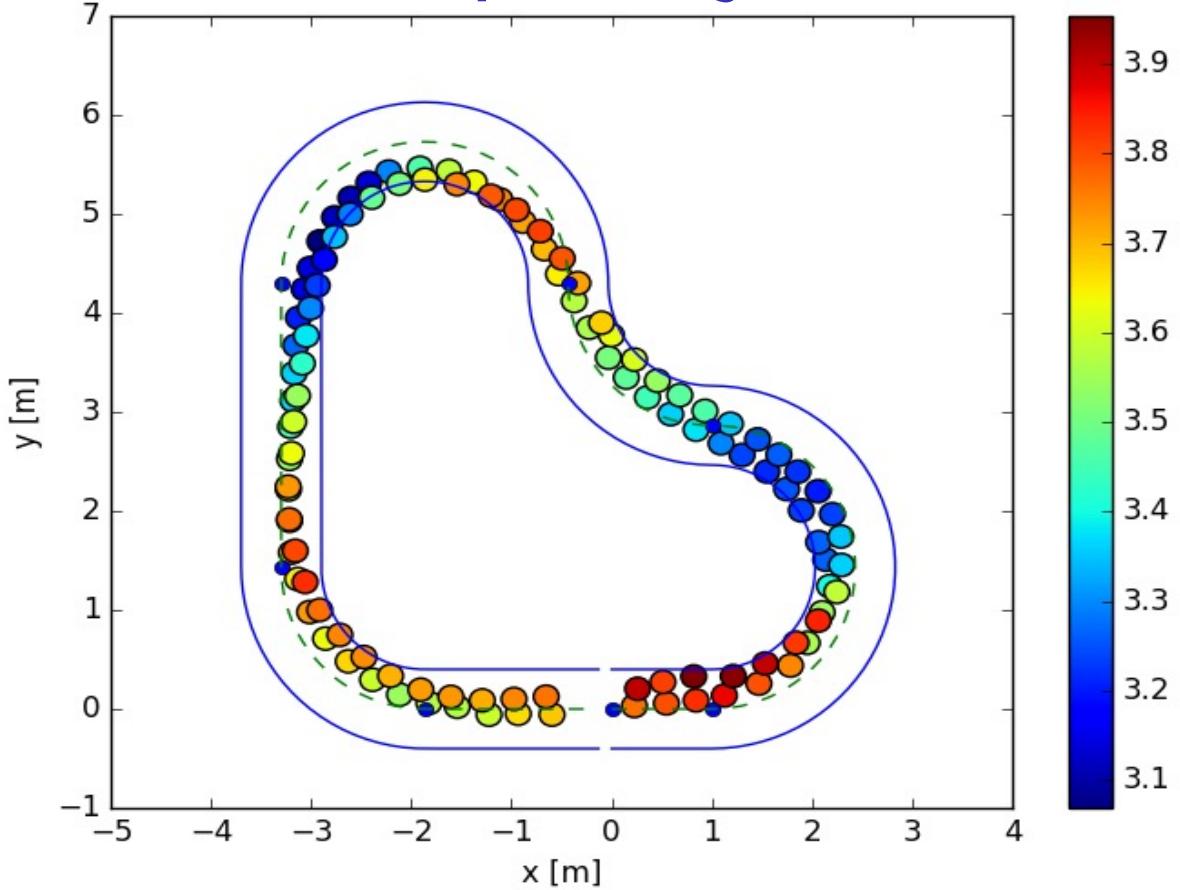
Implementation Details

- ▶ Compute trajectory to linearize around from previous optimal inputs and stored data
- ▶ Enforce model-based sparsity in local linear regression
- ▶ Perform local linear regression at the linearization points using a subset of the stored data
- ▶ Use kernel $K()$ to weight differently data as a function of distance to the linearization trajectory



Learning Model Predictive Control for Autonomous Racing

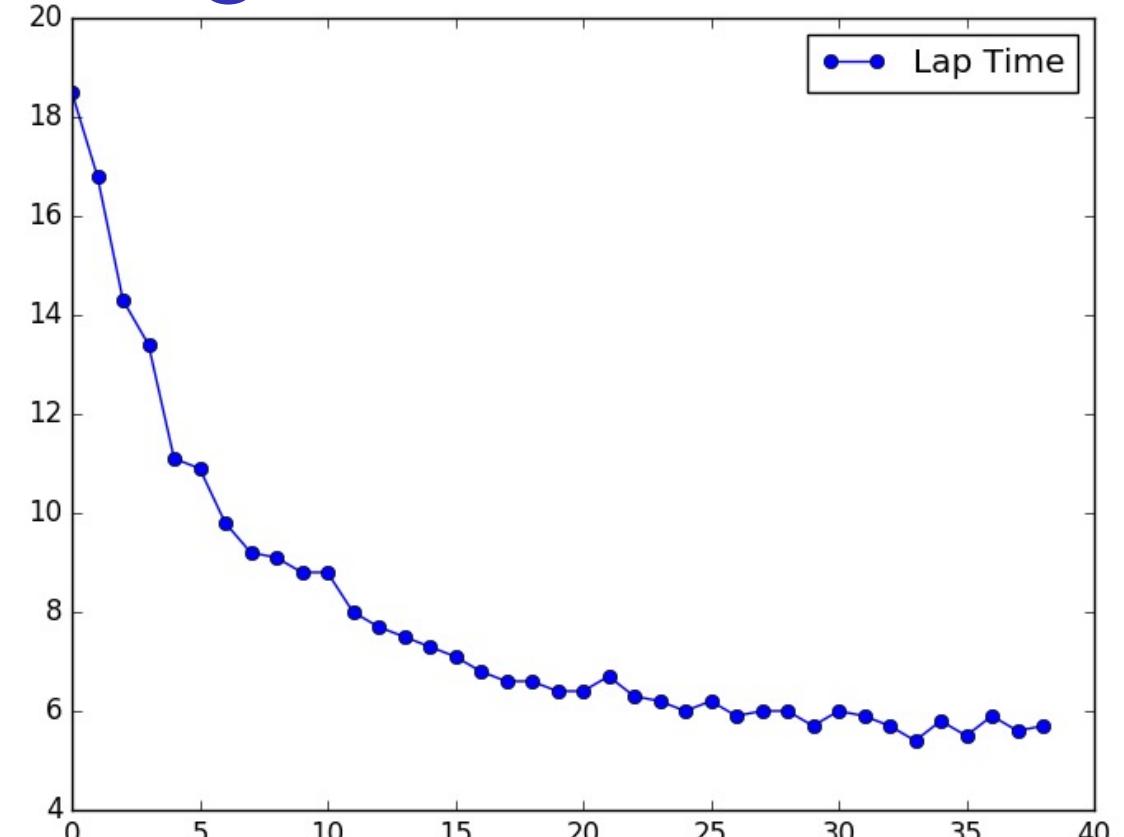
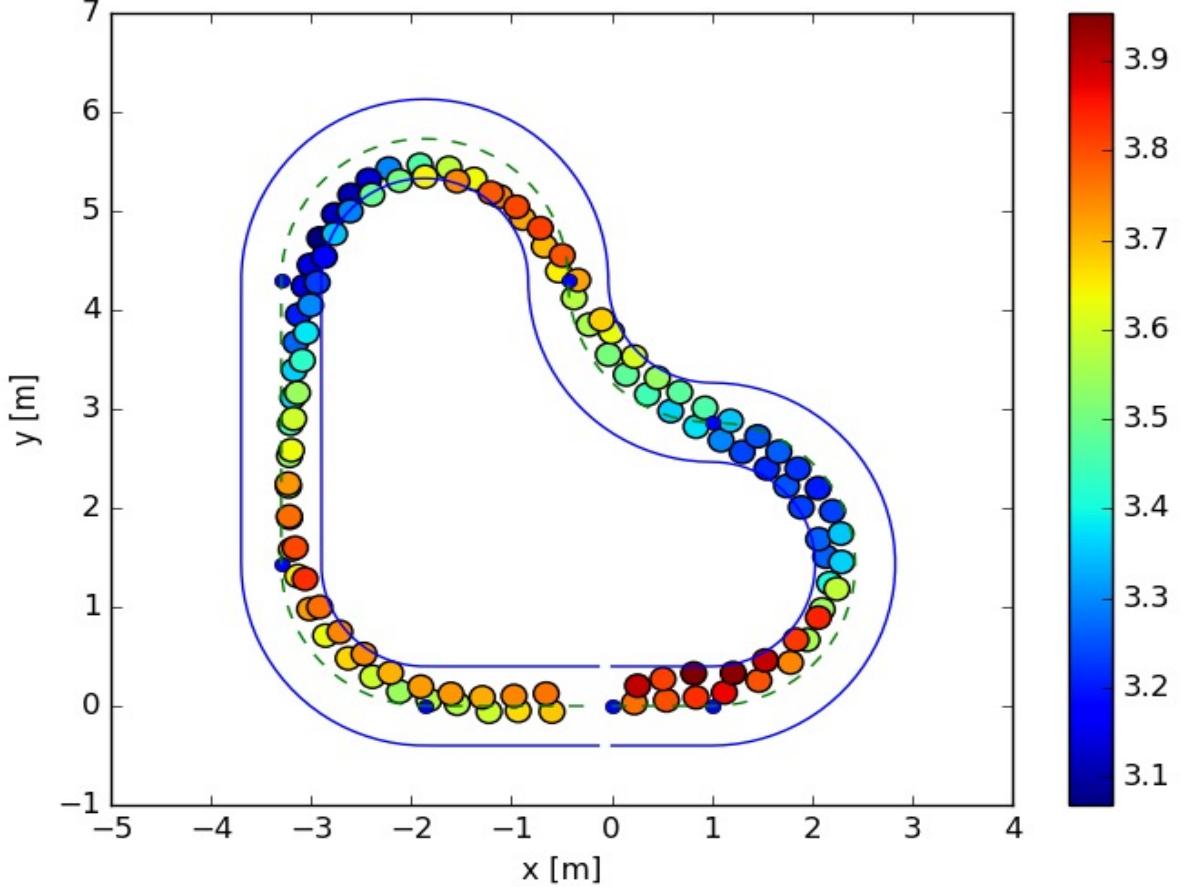
Closed-loop Trajectories at Convergence



Remarks

- ▶ A **reference trajectory is not needed** for the controller implementation
- ▶ The controller converges after few laps: the learning process is data efficient
- ▶ The **controller safely explores the state space** iteratively improving the lap time

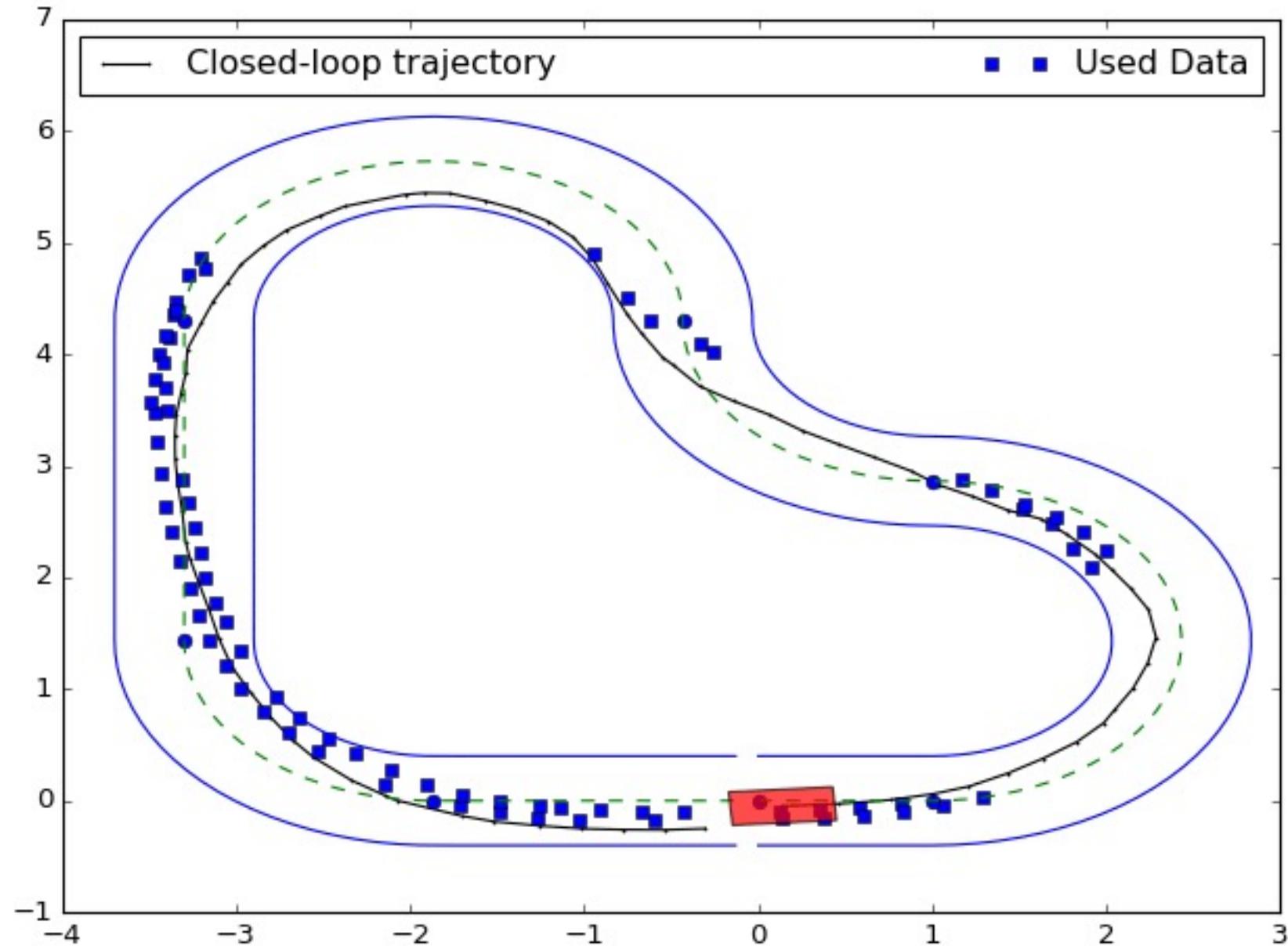
Closed-loop Trajectories at Convergence



Remarks

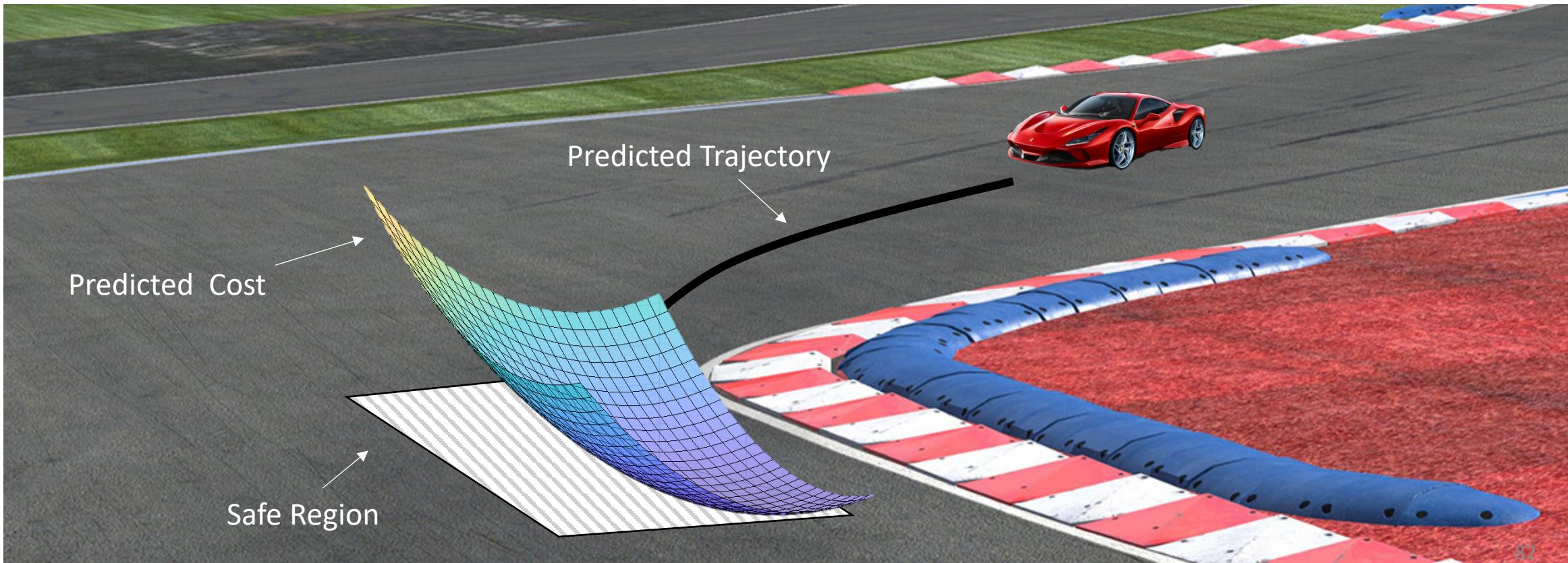
- ▶ A **reference trajectory is not needed** for the controller implementation
- ▶ The controller converges after few laps: the learning process is data efficient
- ▶ The **controller safely explores the state space** iteratively improving the lap time

Data Point Selection



The key components

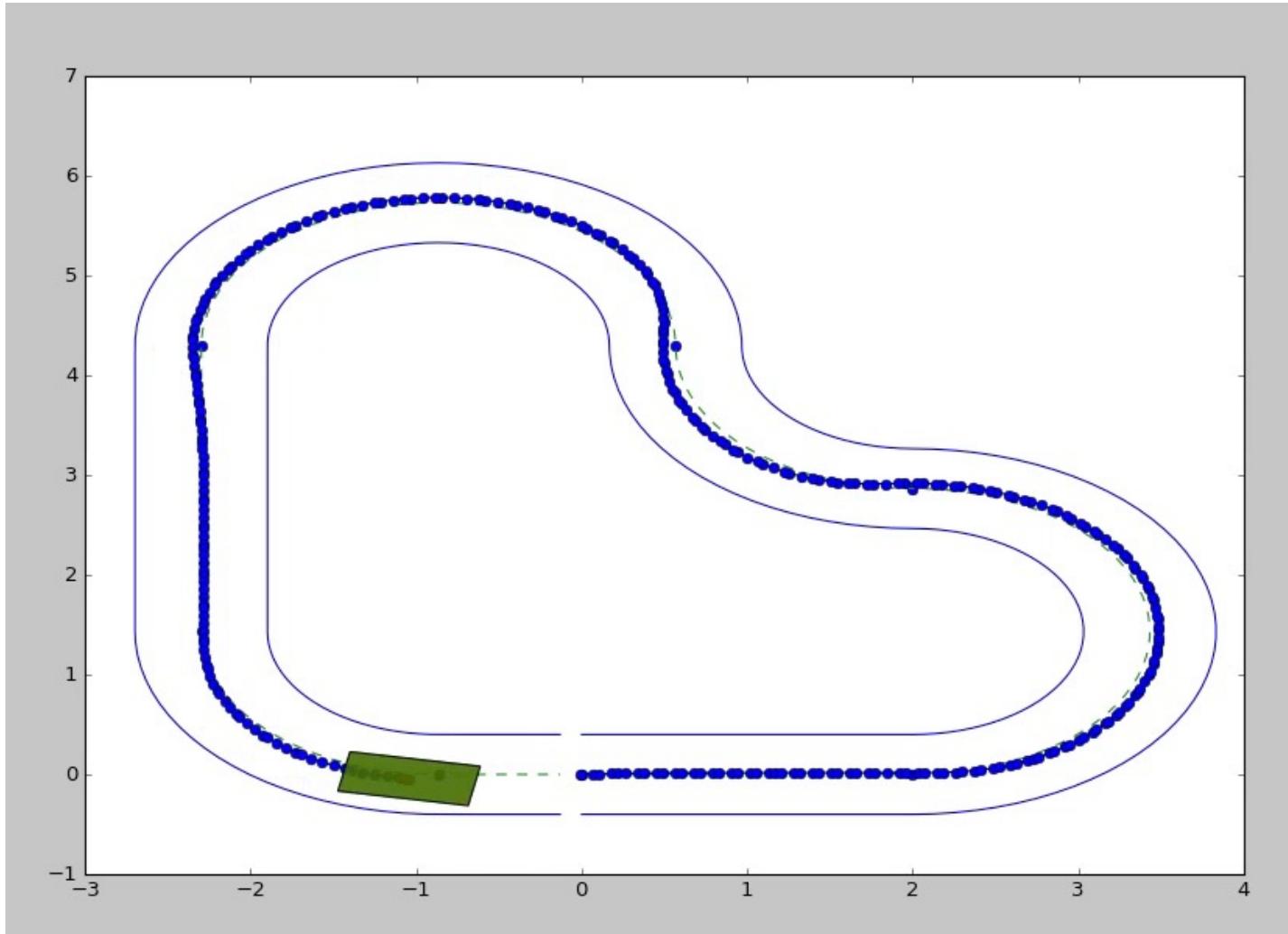
- ▶ Predicted trajectory given by prediction model
- ▶ Predicted cost estimated by value function approximation
- ▶ Safe region estimated by the safe set



Do you need the safe set? – Yes

LMPC without Invariant Set

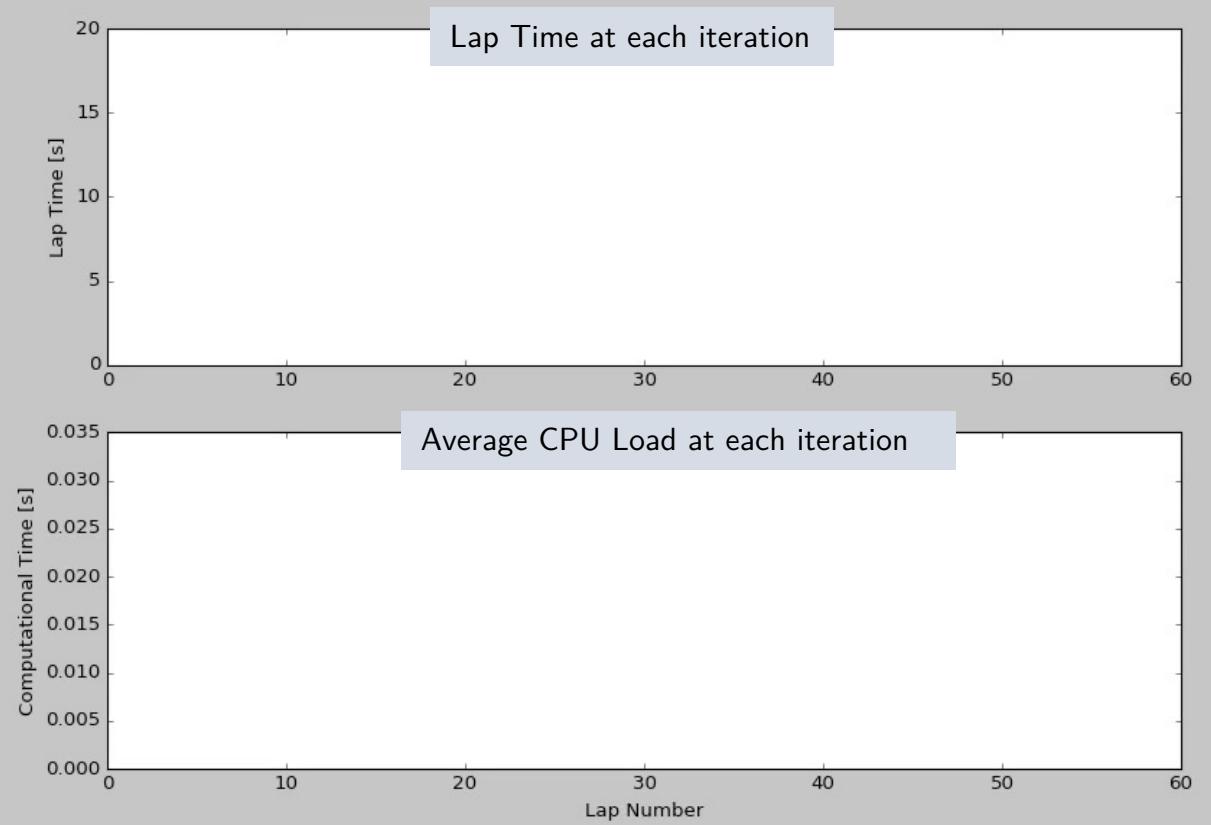
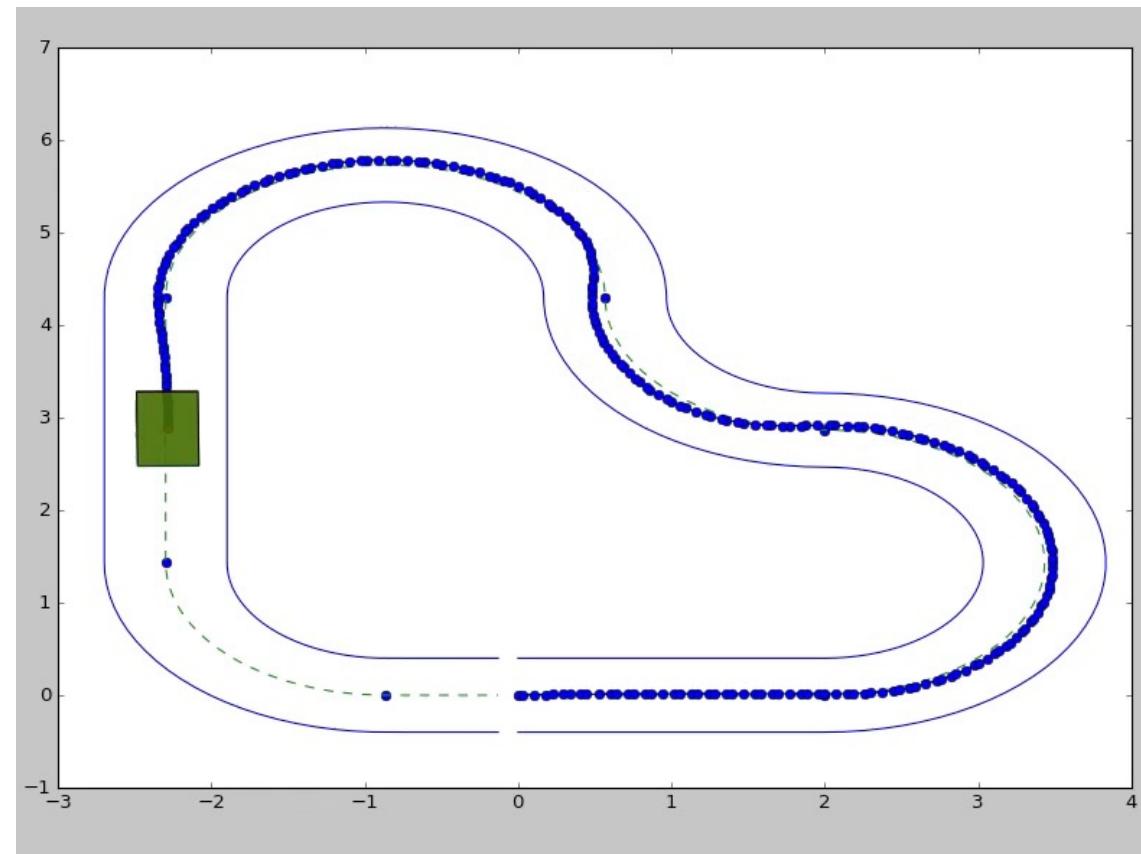
The controller extrapolates the value function on the Vx dimension



Do you need to Predict to Learn? Yes

When the LMPC horizon is $N = 1$ the controller

- ▶ solves the Bellman equation using the value function approximation
- ▶ does not explore the state space as it cannot plan outside the safe set



Comparison with Approximate DP (aka RL)

- ▶ Some references:
 - ❖ Bertsekas paper connecting MPC and ADP [1], books on RL and OC [2,3]
 - ❖ Lewis and Vrabie survey [4]
 - ❖ Recht survey [5]

- ▶ LMPC highlights
 - ❖ Continuous state and action formulation
 - ❖ Constraints satisfaction and Sampled Safe Sets
 - ❖ Q-function constructed locally based on cost/model driven exploration
 - ❖ Q-function at stored state is “exact” and upperbounds at intermediate points

[1] D. Bertsekas, “Dynamic programming and suboptimal control: A survey from ADP to MPC.” European Journal of Control 11.4-5 (2005)

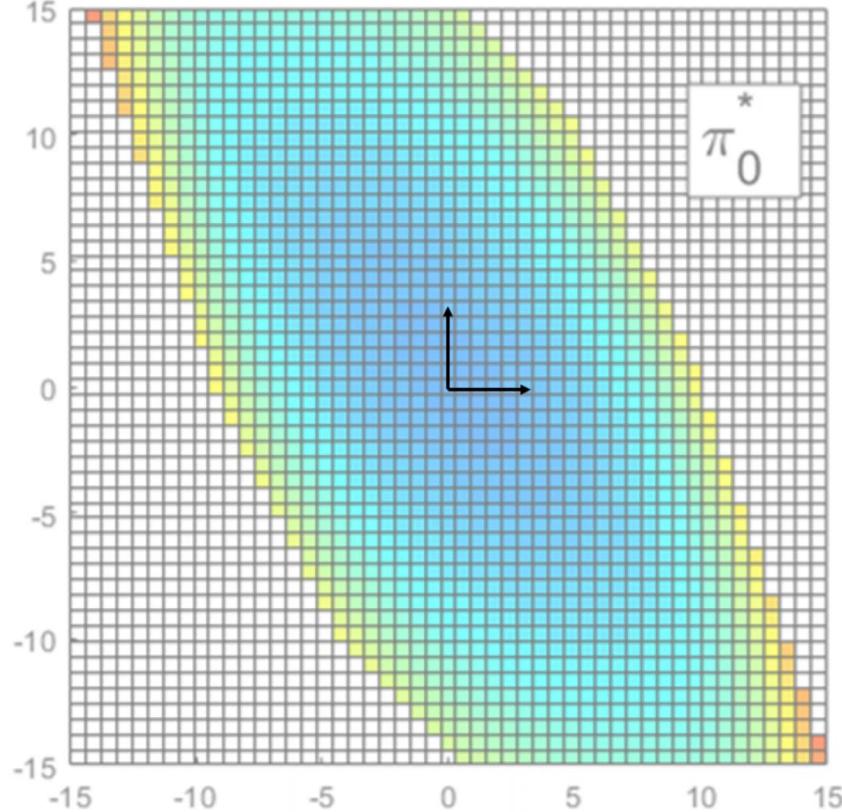
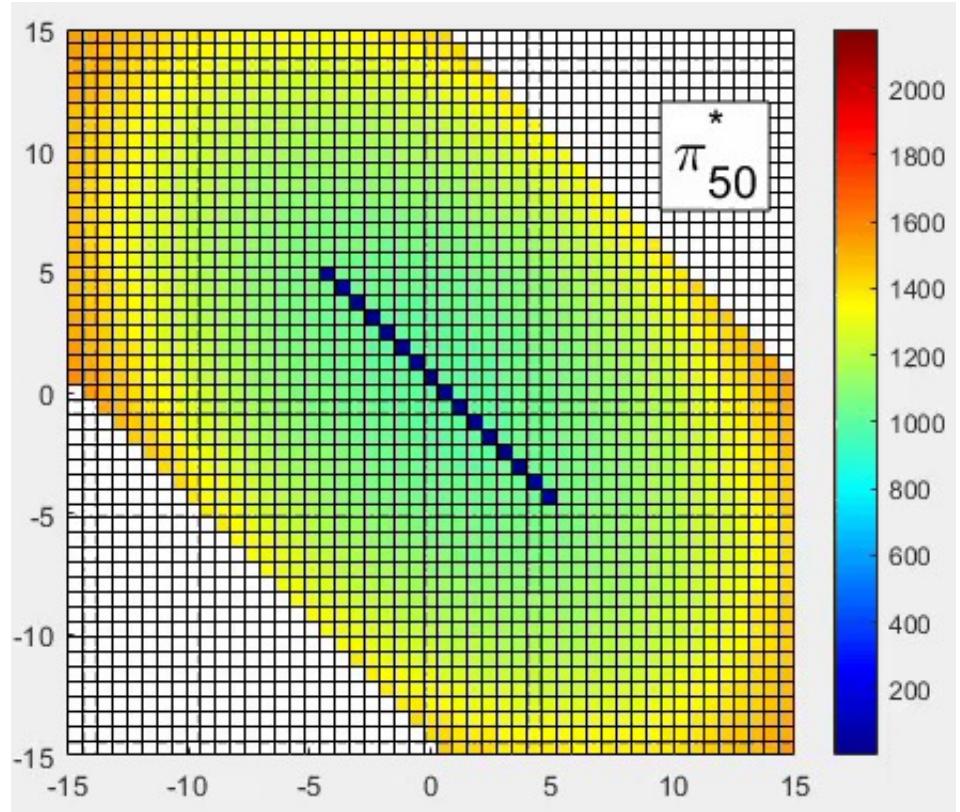
[2] D. Bertsekas, “Reinforcement learning and optimal control.” Athena Scientific, 2019.

[3] D. Bertsekas, “Distributed Reinforcement Learning” http://web.mit.edu/dimitrib/www/RL_2_Rollout_&_PI.pdf

[4] F. Lewis, Frank, and D. Vrabie. "Reinforcement learning and adaptive dynamic programming for feedback control." IEEE circuits and systems magazine 9.3 (2009)

[5] R. Benjamin. "A tour of reinforcement learning: The view from continuous control." Annual Review of Control, Robotics, and Autonomous Systems 2 (2019)

Forward Value Iteration



Dynamic Programming:

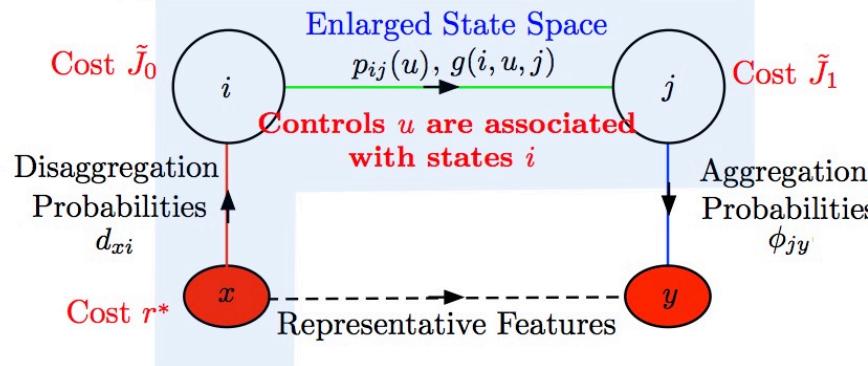
- ▶ Gridding, global properties
- ▶ Backward, one-step iteration

LMPC:

- ▶ No Gridding, local properties
- ▶ Forward, multi-step prediction
- ▶ LICQ required for optimality

LMPC as Efficient and Safe Aggregated ADP

More Accurate Version: The Enlarged Aggregate Problem



Bellman equations for the enlarged problem

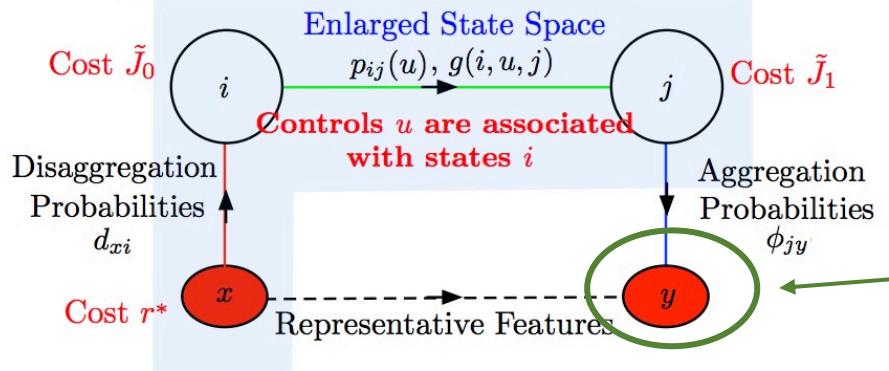
$$\begin{aligned} r_x^* &= \sum_{i=1}^n d_{xi} \tilde{J}_0(i), \quad x \in \mathcal{A}, \\ \tilde{J}_0(i) &= \min_{u \in U(i)} \sum_{j=1}^n p_{ij}(u) (g(i, u, j) + \alpha \tilde{J}_1(j)), \quad i = 1, \dots, n, \\ \tilde{J}_1(j) &= \sum_{y \in \mathcal{A}} \phi_{jy} r_y^*, \quad j = 1, \dots, n \end{aligned}$$

r^* solves uniquely the composite Bellman equation $r^* = Hr^*$:

$$r_x^* = (Hr^*)(x) = \sum_{i=1}^n d_{xi} \min_{u \in U(i)} \sum_{j=1}^n p_{ij}(u) \left(g(i, u, j) + \alpha \sum_{y \in \mathcal{A}} \phi_{jy} r_y^* \right), \quad x \in \mathcal{A}$$

LMPC as Efficient and Safe Aggregated ADP

More Accurate Version: The Enlarged Aggregate Problem



1 - Let MPC explore and iteratively construct the set of aggregate states

Bellman equations for the enlarged problem

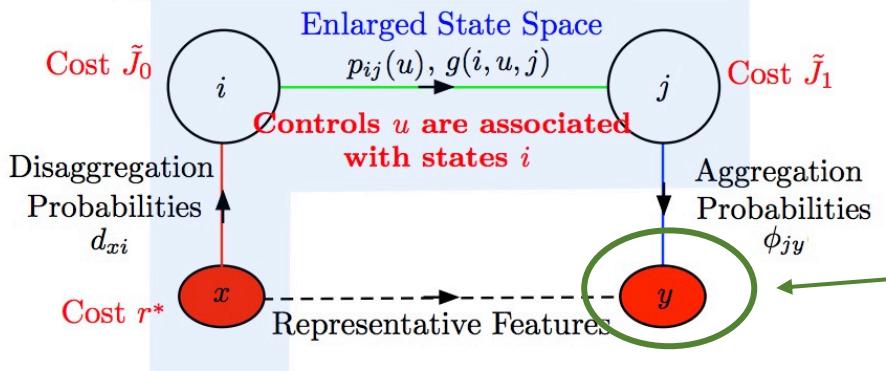
$$\begin{aligned} r_x^* &= \sum_{i=1}^n d_{xi} \tilde{J}_0(i), \quad x \in \mathcal{A}, \\ \tilde{J}_0(i) &= \min_{u \in U(i)} \sum_{j=1}^n p_{ij}(u) (g(i, u, j) + \alpha \tilde{J}_1(j)), \quad i = 1, \dots, n, \\ \tilde{J}_1(j) &= \sum_{y \in \mathcal{A}} \phi_{jy} r_y^*, \quad j = 1, \dots, n \end{aligned}$$

r^* solves uniquely the composite Bellman equation $r^* = Hr^*$:

$$r_x^* = (Hr^*)(x) = \sum_{i=1}^n d_{xi} \min_{u \in U(i)} \sum_{j=1}^n p_{ij}(u) \left(g(i, u, j) + \alpha \sum_{y \in \mathcal{A}} \phi_{jy} r_y^* \right), \quad x \in \mathcal{A}$$

LMPC as Efficient and Safe Aggregated ADP

More Accurate Version: The Enlarged Aggregate Problem



1 - Let MPC explore and iteratively construct the set of aggregate states

Bellman equations for the enlarged problem

$$r_x^* = \sum_{i=1}^n d_{xi} \tilde{J}_0(i), \quad x \in \mathcal{A},$$
$$\tilde{J}_0(i) = \min_{u \in U(i)} \sum_{j=1}^n p_{ij}(u) (g(i, u, j) + \alpha \tilde{J}_1(j)), \quad i = 1, \dots, n,$$
$$\tilde{J}_1(j) = \sum_{y \in \mathcal{A}} \phi_{jy} r_y^*, \quad j = 1, \dots, n$$

2 – Use barycentric approximation to compute aggregation probabilities

r^* solves uniquely the composite Bellman equation $r^* = Hr^*$:

$$r_x^* = (Hr^*)(x) = \sum_{i=1}^n d_{xi} \min_{u \in U(i)} \sum_{j=1}^n p_{ij}(u) \left(g(i, u, j) + \alpha \sum_{y \in \mathcal{A}} \phi_{jy} r_y^* \right), \quad x \in \mathcal{A}$$

Model Learning in MPC

A short and non-comprehensive summary

The complexity of the prediction model

The complexity of the prediction model

Linear

$$x_{k+1} = Ax_k + Bu_k + w_k$$

The complexity of the prediction model

Linear

$$x_{k+1} = Ax_k + Bu_k + w_k$$

known Gray Box unknown

$$x_{k+1} = f(x_k, u_k) + g(x_k, u_k) + w_k$$

The complexity of the prediction model

Linear

$$x_{k+1} = Ax_k + Bu_k + w_k$$

known Gray Box unknown

$$x_{k+1} = f(x_k, u_k) + g(x_k, u_k) + w_k$$

Black Box

$$x_{k+1} = x_k + g(x_k, u_k, w_k)$$

The complexity of the prediction model

Linear

$$x_{k+1} = Ax_k + Bu_k + w_k$$

Marco C., Campi, and Erik Weyer. "Finite sample properties of system identification methods." *IEEE Transactions on Automatic Control* 47.8 (2002): 1329-1334.

Sarah Dean, Horia Mania, Nikolai Matni, Benjamin Recht, and Stephen Tu. "On the sample complexity of the linear quadratic regulator." *Foundations of Computational Mathematics* (2019): 1-47.

Gray Box

$$x_{k+1} = f(x_k, u_k) + g(x_k, u_k) + w_k$$

known

unknown

Black Box

$$x_{k+1} = x_k + g(x_k, u_k, w_k)$$

The complexity of the prediction model

Linear

$$x_{k+1} = Ax_k + Bu_k + w_k$$

Marco C., Campi, and Erik Weyer. "Finite sample properties of system identification methods." *IEEE Transactions on Automatic Control* 47.8 (2002): 1329-1334.

Sarah Dean, Horia Mania, Nikolai Matni, Benjamin Recht, and Stephen Tu. "On the sample complexity of the linear quadratic regulator." *Foundations of Computational Mathematics* (2019): 1-47.

A screenshot of a Google Scholar search results page. The search query 'learning LQR' is entered in the search bar. The results section shows two articles:

- Trajectory tracking using online learning LQR with adaptive learning control of a leg-exoskeleton for disorder gait rehabilitation**
N Ajanaromvat, M Pamichkun - Mechatronics, 2018 - Elsevier
Precise trajectory tracking of gait pattern under varied load condition is necessary for rehabilitation using leg exoskeleton. In this paper, online iterative learning linear quadratic regulator (OILLR) with adaptive iterative learning control is proposed to control trajectory ...
☆ 99 Cited by 17 Related articles All 2 versions
- Learning robust control for LQR systems with multiplicative noise via policy gradient**
B Gravell, PM Esfahani, T Summers - arXiv preprint arXiv:1905.13547, 2019 - arxiv.org
The linear quadratic regulator (LQR) problem has reemerged as an important theoretical benchmark for reinforcement learning-based control of complex dynamical systems with continuous state and action spaces. In contrast with nearly all recent work in this area, we ...
☆ 99 Cited by 22 Related articles All 12 versions

Filtering options on the left include: Any time, Since 2021, Since 2020, Since 2017, Custom range..., Date range (2018), Sort by relevance, Sort by date, and include patents.

Gray Box

$$x_{k+1} = f(x_k, u_k) + g(x_k, u_k) + w_k$$

known unknown

Black Box

$$x_{k+1} = x_k + g(x_k, u_k, w_k)$$

The complexity of the prediction model

Linear

$$x_{k+1} = Ax_k + Bu_k + w_k$$

Marco C., Campi, and Erik Weyer. "Finite sample properties of system identification methods." *IEEE Transactions on Automatic Control* 47.8 (2002): 1329-1334.

Sarah Dean, Horia Mania, Nikolai Matni, Benjamin Recht, and Stephen Tu. "On the sample complexity of the linear quadratic regulator." *Foundations of Computational Mathematics* (2019): 1-47.

Google Scholar search results for "learning LQR". The search bar shows "learning LQR". The results section says "About 8,170 results (0.05 sec)". The first result is a paper titled "[HTML] Trajectory tracking using online learning LQR with adaptive learning control of a leg-exoskeleton for disorder gait rehabilitation" by N Ajanaromvat, M Pamichkun - Mechatronics, 2018 - Elsevier. The second result is a paper titled "Hybrid fuzzy learning controller for an unstable nonlinear system" by BM Chung, JW Lee, HH Joo... - International Journal of ... 2000 - koreascience.or.kr. Both results have a "Cited by 17" link. The search interface shows filters for "Any time", "Since 2021", "Since 2020", "Since 2017", and "Custom range" from 2018 to 2018. The bottom part of the screenshot shows another search for "learning LQR" with a custom range from 2000 to 2004, resulting in about 1,440 results. The results list includes a paper titled "Synchronizing dual-drive gantry of chip mounter with LQR approach" by S Kim, B Chu, D Hong, HK Park... - 2003 IEEE/ASME ... 2003 - ieeexplore.ieee.org.

Gray Box

$$x_{k+1} = f(x_k, u_k) + g(x_k, u_k) + w_k$$

known unknown

Black Box

$$x_{k+1} = x_k + g(x_k, u_k, w_k)$$

The complexity of the prediction model

Linear

$$x_{k+1} = Ax_k + Bu_k + w_k$$

Marco C., Campi, and Erik Weyer. "Finite sample properties of system identification methods." *IEEE Transactions on Automatic Control* 47.8 (2002): 1329-1334.

Sarah Dean, Horia Mania, Nikolai Matni, Benjamin Recht, and Stephen Tu. "On the sample complexity of the linear quadratic regulator." *Foundations of Computational Mathematics* (2019): 1-47.

A screenshot of a Google Scholar search results page. The search query 'learning LQR' is entered in the search bar. The results section shows a list of articles. The first result is highlighted with a blue border and shows the following details:

[HTML] Trajectory tracking using online learning LQR with adaptive learning control of a leg-exoskeleton for disorder gait rehabilitation
N Ajanaromvat, M Pamichkun - Mechatronics, 2018 - Elsevier
Precise trajectory tracking of gait pattern under varied load condition is necessary for rehabilitation using leg exoskeleton. In this paper, online iterative learning linear quadratic regulator (OILLQR) with adaptive iterative learning control is proposed to control trajectory ...
☆ 89 Cited by 17 Related articles All 2 versions

Enrico Terzi, Lorenzo Fagiano, Marcello Farina, Riccardo Scattolini. Learning multi-step prediction models for receding horizon control. In 2018 European Control Conference (ECC) 2018 Jun 12 (pp. 1335-1340). IEEE.

Jeremy Coulson, John Lygeros, and Florian Dörfler. "Data-enabled predictive control: In the shallows of the DeePC." 2019 18th European Control Conference (ECC). IEEE, 2019.

Gray Box

$$x_{k+1} = f(x_k, u_k) + g(x_k, u_k) + w_k$$

known unknown

Black Box

$$x_{k+1} = x_k + g(x_k, u_k, w_k)$$

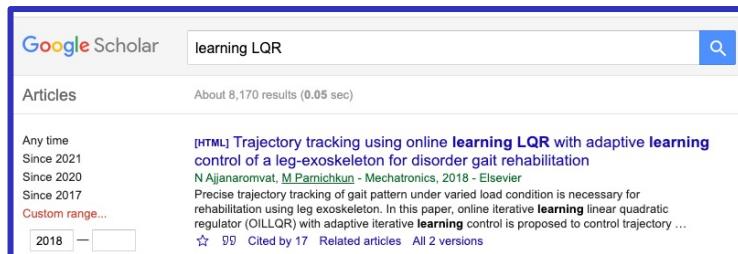
The complexity of the prediction model

Linear

$$x_{k+1} = Ax_k + Bu_k + w_k$$

Marco C., Campi, and Erik Weyer. "Finite sample properties of system identification methods." *IEEE Transactions on Automatic Control* 47.8 (2002): 1329-1334.

Sarah Dean, Horia Mania, Nikolai Matni, Benjamin Recht, and Stephen Tu. "On the sample complexity of the linear quadratic regulator." *Foundations of Computational Mathematics* (2019): 1-47.



Enrico Terzi, Lorenzo Fagiano, Marcello Farina, Riccardo Scattolini. Learning multi-step prediction models for receding horizon control. In 2018 European Control Conference (ECC) 2018 Jun 12 (pp. 1335-1340). IEEE.

Jeremy Coulson, John Lygeros, and Florian Dörfler. "Data-enabled predictive control: In the shallows of the DeePC." 2019 18th European Control Conference (ECC). IEEE, 2019.

Gray Box

$$x_{k+1} = f(x_k, u_k) + g(x_k, u_k) + w_k$$

Anil Aswani, Humberto Gonzalez, S. Shankar Sastry, and Claire Tomlin. "Provably safe and robust learning-based model predictive control." *Automatica* 49, no. 5 (2013)

Lukas Hewing, Kim P. Wabersich, Marcel Menner, and Melanie N. Zeilinger. "Learning-based model predictive control: Toward safe learning in control." *Annual Review of Control, Robotics, and Autonomous Systems* 3 (2020):

Lukas Hewing, Juraj Kabzan, and Melanie N. Zeilinger. "Cautious model predictive control using Gaussian process regression." *IEEE Transactions on Control Systems Technology* 28, no. 6 (2019): 2736-2743.

Milanese, Mario, and Carlo Novara. "Set membership identification of nonlinear systems." *Automatica* 40, no. 6 (2004): 957-975.

Marko Tanaskovic, Lorenzo Fagiano, Roy Smith, Paul Goulart, and Manfred Morari. "Adaptive model predictive control for constrained linear systems." In *2013 European Control Conference (ECC)*, pp. 382-387. IEEE, 2013.

Monimoy Bujarbaruah, Xiaojing Zhang, Ugo Rosolia, and Francesco Borrelli. "Adaptive MPC for iterative tasks." In *2018 IEEE Conference on Decision and Control (CDC)*, pp. 6322-6327. IEEE, 2018.

Black Box

$$x_{k+1} = x_k + g(x_k, u_k, w_k)$$

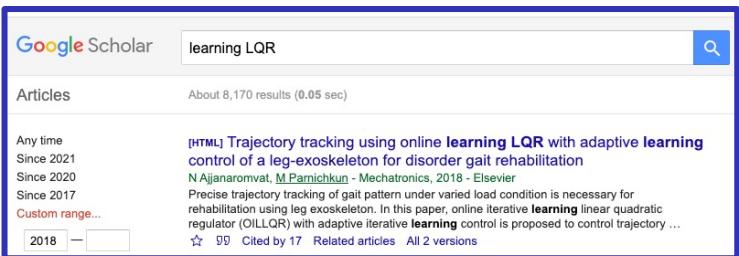
The complexity of the prediction model

Linear

$$x_{k+1} = Ax_k + Bu_k + w_k$$

Marco C., Campi, and Erik Weyer. "Finite sample properties of system identification methods." *IEEE Transactions on Automatic Control* 47.8 (2002): 1329-1334.

Sarah Dean, Horia Mania, Nikolai Matni, Benjamin Recht, and Stephen Tu. "On the sample complexity of the linear quadratic regulator." *Foundations of Computational Mathematics* (2019): 1-47.



Enrico Terzi, Lorenzo Fagiano, Marcello Farina, Riccardo Scattolini. Learning multi-step prediction models for receding horizon control. In 2018 European Control Conference (ECC) 2018 Jun 12 (pp. 1335-1340). IEEE.

Jeremy Coulson, John Lygeros, and Florian Dörfler. "Data-enabled predictive control: In the shallows of the DeePC." *2019 18th European Control Conference (ECC)*. IEEE, 2019.

Gray Box

$$x_{k+1} = f(x_k, u_k) + g(x_k, u_k) + w_k$$

Anil Aswani, Humberto Gonzalez, S. Shankar Sastry, and Claire Tomlin. "Provably safe and robust learning-based model predictive control." *Automatica* 49, no. 5 (2013)

Lukas Hewing, Kim P. Wabersich, Marcel Menner, and Melanie N. Zeilinger. "Learning-based model predictive control: Toward safe learning in control." *Annual Review of Control, Robotics, and Autonomous Systems* 3 (2020):

Lukas Hewing, Juraj Kabzan, and Melanie N. Zeilinger. "Cautious model predictive control using Gaussian process regression." *IEEE Transactions on Control Systems Technology* 28, no. 6 (2019): 2736-2743.

Milanese, Mario, and Carlo Novara. "Set membership identification of nonlinear systems." *Automatica* 40, no. 6 (2004): 957-975.

Marko Tanaskovic, Lorenzo Fagiano, Roy Smith, Paul Goulart, and Manfred Morari. "Adaptive model predictive control for constrained linear systems." In *2013 European Control Conference (ECC)*, pp. 382-387. IEEE, 2013.

Monimoy Bujarbaruah, Xiaojing Zhang, Ugo Rosolia, and Francesco Borrelli. "Adaptive MPC for iterative tasks." In *2018 IEEE Conference on Decision and Control (CDC)*, pp. 6322-6327. IEEE, 2018.

Black Box

$$x_{k+1} = x_k + g(x_k, u_k, w_k)$$

Chua, Kurtland, Roberto Calandra, Rowan McAllister, and Sergey Levine. "Deep reinforcement learning in a handful of trials using probabilistic dynamics models." *NeurIPS* (2018).

Anusha Nagabandi, Gregory Kahn, Ronald S. Fearing, and Sergey Levine. "Neural network dynamics for model-based deep reinforcement learning with model-free fine-tuning." In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 7559-7566. IEEE, 2018.

Lenz, Ian, Ross A. Knepper, and Ashutosh Saxena. "DeepMPC: Learning deep latent features for model predictive control." In *Robotics: Science and Systems*. 2015.

Kendall Lowrey, Aravind Rajeswaran, Sham Kakade, Emanuel Todorov, and Igor Mordatch. "Plan online, learn offline: Efficient learning and exploration via model-based control." In *Proc. Int. Conf. Mach. Learn.*, 2019.

Brijen Thananjeyan, Ashwin Balakrishna, Ugo Rosolia, Felix Li, Rowan McAllister, Joseph E. Gonzalez, Sergey Levine, Francesco Borrelli, and Ken Goldberg. "Safety augmented value estimation from demonstrations (saved): Safe deep model-based RL for sparse cost robotic tasks." *IEEE Robotics and Automation Letters* 5, no. 2 (2020).

The complexity of the prediction model

Linear

$$x_{k+1} = Ax_k + Bu_k + w_k$$

Marco C., Campi, and Erik Weyer. "Finite sample properties of system identification methods." *IEEE Transactions on Automatic Control* 47.8 (2002): 1329-1334.

Sarah Dean, Horia Mania, Nikolai Matni, Benjamin Recht, and Stephen Tu. "On the sample complexity of the linear quadratic regulator." *Foundations of Computational Mathematics* (2019): 1-47.

A screenshot of a Google Scholar search results page. The search query 'learning LQR' is entered in the search bar. The results section shows a list of articles. The first result is a paper by N. Ajanaromvat and M. Pamichkun titled 'Trajectory tracking using online learning LQR with adaptive learning control of a leg-exoskeleton for disorder gait rehabilitation'. The abstract discusses the use of online iterative learning linear quadratic regulator (OILLQR) with adaptive iterative learning control to control trajectory. The search results page also includes filters for 'Articles', 'Anytime', 'Custom range...', and date ranges from 2018 to 2021.

Enrico Terzi, Lorenzo Fagiano, Marcello Farina, Riccardo Scattolini. Learning multi-step prediction models for receding horizon control. In 2018 European Control Conference (ECC) 2018 Jun 12 (pp. 1335-1340). IEEE.

Jeremy Coulson, John Lygeros, and Florian Dörfler. "Data-enabled predictive control: In the shallows of the DeePC." 2019 18th European Control Conference (ECC). IEEE, 2019.

Gray Box

$$x_{k+1} = f(x_k, u_k) + g(x_k, u_k) + w_k$$

Anil Aswani, Humberto Gonzalez, S. Shankar Sastry, and Claire Tomlin. "Provably safe and robust learning-based model predictive control." *Automatica* 49, no. 5 (2013)

Lukas Hewing, Kim P. Wabersich, Marcel Menner, and Melanie N. Zeilinger. "Learning-based model predictive control: Toward safe learning in control." *Annual Review of Control, Robotics, and Autonomous Systems* 3 (2020):

Lukas Hewing, Juraj Kabzan, and Melanie N. Zeilinger. "Cautious model predictive control using Gaussian process regression." *IEEE Transactions on Control Systems Technology* 28, no. 6 (2019): 2736-2743.

Milanese, Mario, and Carlo Novara. "Set membership identification of nonlinear systems." *Automatica* 40, no. 6 (2004): 957-975.

Marko Tanaskovic, Lorenzo Fagiano, Roy Smith, Paul Goulart, and Manfred Morari. "Adaptive model predictive control for constrained linear systems." In *2013 European Control Conference (ECC)*, pp. 382-387. IEEE, 2013.

Monimoy Bujarbaruah, Xiaojing Zhang, Ugo Rosolia, and Francesco Borrelli. "Adaptive MPC for iterative tasks." In *2018 IEEE Conference on Decision and Control (CDC)*, pp. 6322-6327. IEEE, 2018.

Black Box

$$x_{k+1} = x_k + g(x_k, u_k, w_k)$$

Chua, Kurtland, Roberto Calandra, Rowan McAllister, and Sergey Levine. "Deep reinforcement learning in a handful of trials using probabilistic dynamics models." *NeurIPS* (2018).

Anusha Nagabandi, Gregory Kahn, Ronald S. Fearing, and Sergey Levine. "Neural network dynamics for model-based deep reinforcement learning with model-free fine-tuning." In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 7559-7566. IEEE, 2018.

Lenz, Ian, Ross A. Knepper, and Ashutosh Saxena. "DeepMPC: Learning deep latent features for model predictive control." In *Robotics: Science and Systems*. 2015.

Kendall Lowrey, Aravind Rajeswaran, Sham Kakade, Emanuel Todorov, and Igor Mordatch. "Plan online, learn offline: Efficient learning and exploration via model-based control." In *Proc. Int. Conf. Mach. Learn.*, 2019.

Brijen Thananjeyan, Ashwin Balakrishna, Ugo Rosolia, Felix Li, Rowan McAllister, Joseph E. Gonzalez, Sergey Levine, Francesco Borrelli, and Ken Goldberg. "Safety augmented value estimation from demonstrations (saved): Safe deep model-based RL for sparse cost robotic tasks." *IEEE Robotics and Automation Letters* 5, no. 2 (2020).

Learning Linear Models: One Step Prediction

$$\bar{A}, \bar{B} = \arg \min_{A, B} \sum_{k=0}^{T-1} \sum_{i=1}^M \|x_{k+1}^{(i)} - Ax_k^{(i)} - Bu_k^{(i)}\|_2^2$$

Marco C., Campi, and Erik Weyer. "Finite sample properties of system identification methods." *IEEE Transactions on Automatic Control* 47.8 (2002): 1329-1334.

Sarah Dean, Horia Mania, Nikolai Matni, Benjamin Recht, and Stephen Tu. "On the sample complexity of the linear quadratic regulator." *Foundations of Computational Mathematics* (2019): 1-47.

Learning Linear Models: One Step Prediction

$$\bar{A}, \bar{B} = \arg \min_{A, B} \sum_{k=0}^{T-1} \sum_{i=1}^M \|x_{k+1}^{(i)} - Ax_k^{(i)} - Bu_k^{(i)}\|_2^2$$

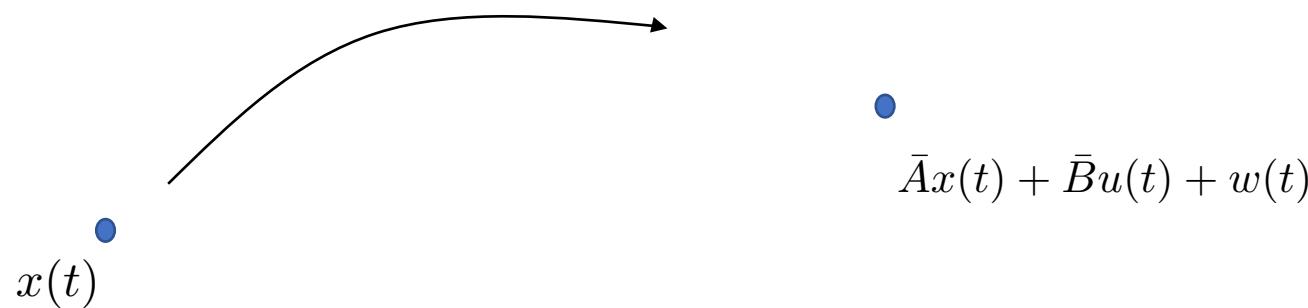
$x(t)$

Marco C., Campi, and Erik Weyer. "Finite sample properties of system identification methods." *IEEE Transactions on Automatic Control* 47.8 (2002): 1329-1334.

Sarah Dean, Horia Mania, Nikolai Matni, Benjamin Recht, and Stephen Tu. "On the sample complexity of the linear quadratic regulator." *Foundations of Computational Mathematics* (2019): 1-47.

Learning Linear Models: One Step Prediction

$$\bar{A}, \bar{B} = \arg \min_{A, B} \sum_{k=0}^{T-1} \sum_{i=1}^M \|x_{k+1}^{(i)} - Ax_k^{(i)} - Bu_k^{(i)}\|_2^2$$

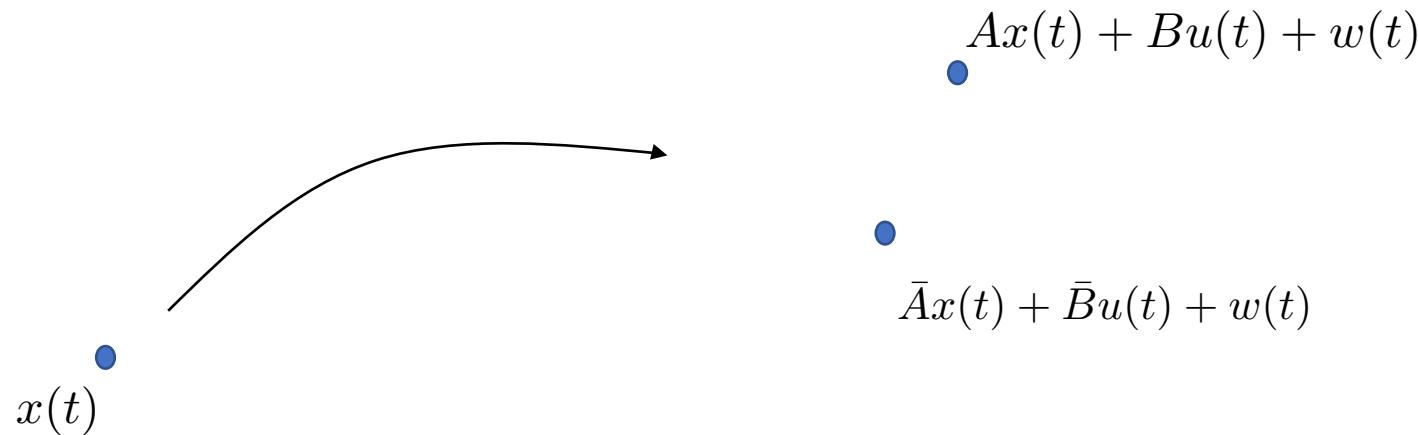


Marco C., Campi, and Erik Weyer. "Finite sample properties of system identification methods." *IEEE Transactions on Automatic Control* 47.8 (2002): 1329-1334.

Sarah Dean, Horia Mania, Nikolai Matni, Benjamin Recht, and Stephen Tu. "On the sample complexity of the linear quadratic regulator." *Foundations of Computational Mathematics* (2019): 1-47.

Learning Linear Models: One Step Prediction

$$\bar{A}, \bar{B} = \arg \min_{A, B} \sum_{k=0}^{T-1} \sum_{i=1}^M \|x_{k+1}^{(i)} - Ax_k^{(i)} - Bu_k^{(i)}\|_2^2$$

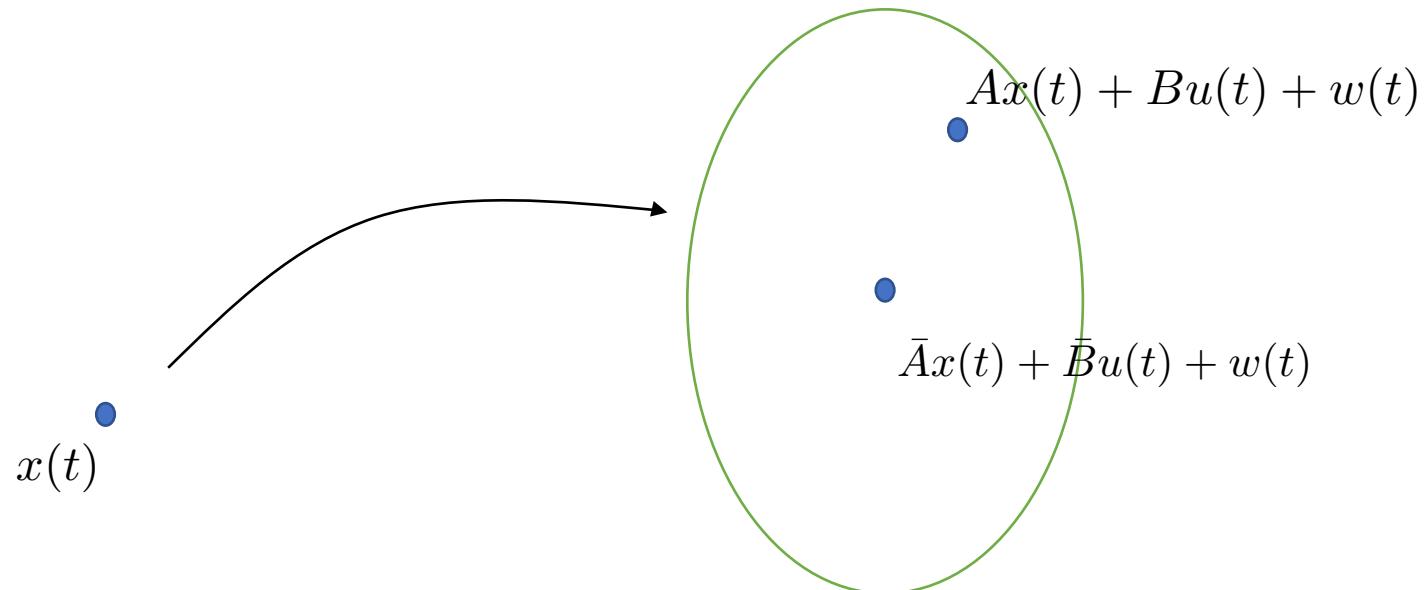


Marco C., Campi, and Erik Weyer. "Finite sample properties of system identification methods." *IEEE Transactions on Automatic Control* 47.8 (2002): 1329-1334.

Sarah Dean, Horia Mania, Nikolai Matni, Benjamin Recht, and Stephen Tu. "On the sample complexity of the linear quadratic regulator." *Foundations of Computational Mathematics* (2019): 1-47.

Learning Linear Models: One Step Prediction

$$\bar{A}, \bar{B} = \arg \min_{A, B} \sum_{k=0}^{T-1} \sum_{i=1}^M \|x_{k+1}^{(i)} - Ax_k^{(i)} - Bu_k^{(i)}\|_2^2$$

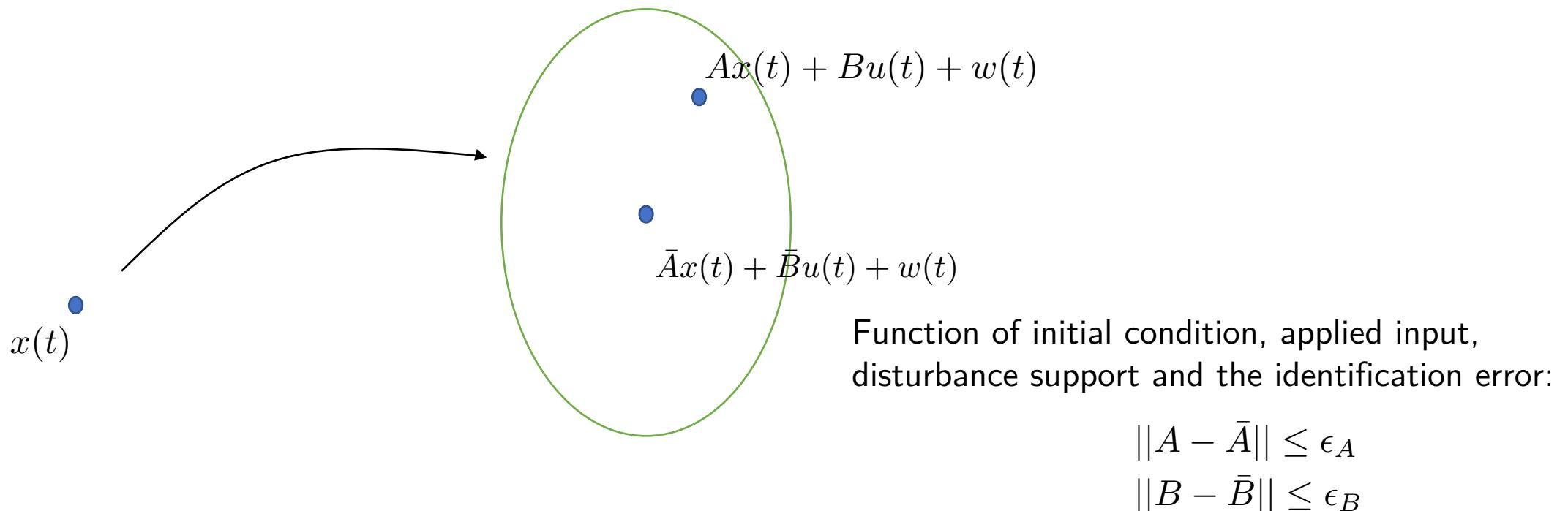


Marco C., Campi, and Erik Weyer. "Finite sample properties of system identification methods." *IEEE Transactions on Automatic Control* 47.8 (2002): 1329-1334.

Sarah Dean, Horia Mania, Nikolai Matni, Benjamin Recht, and Stephen Tu. "On the sample complexity of the linear quadratic regulator." *Foundations of Computational Mathematics* (2019): 1-47.

Learning Linear Models: One Step Prediction

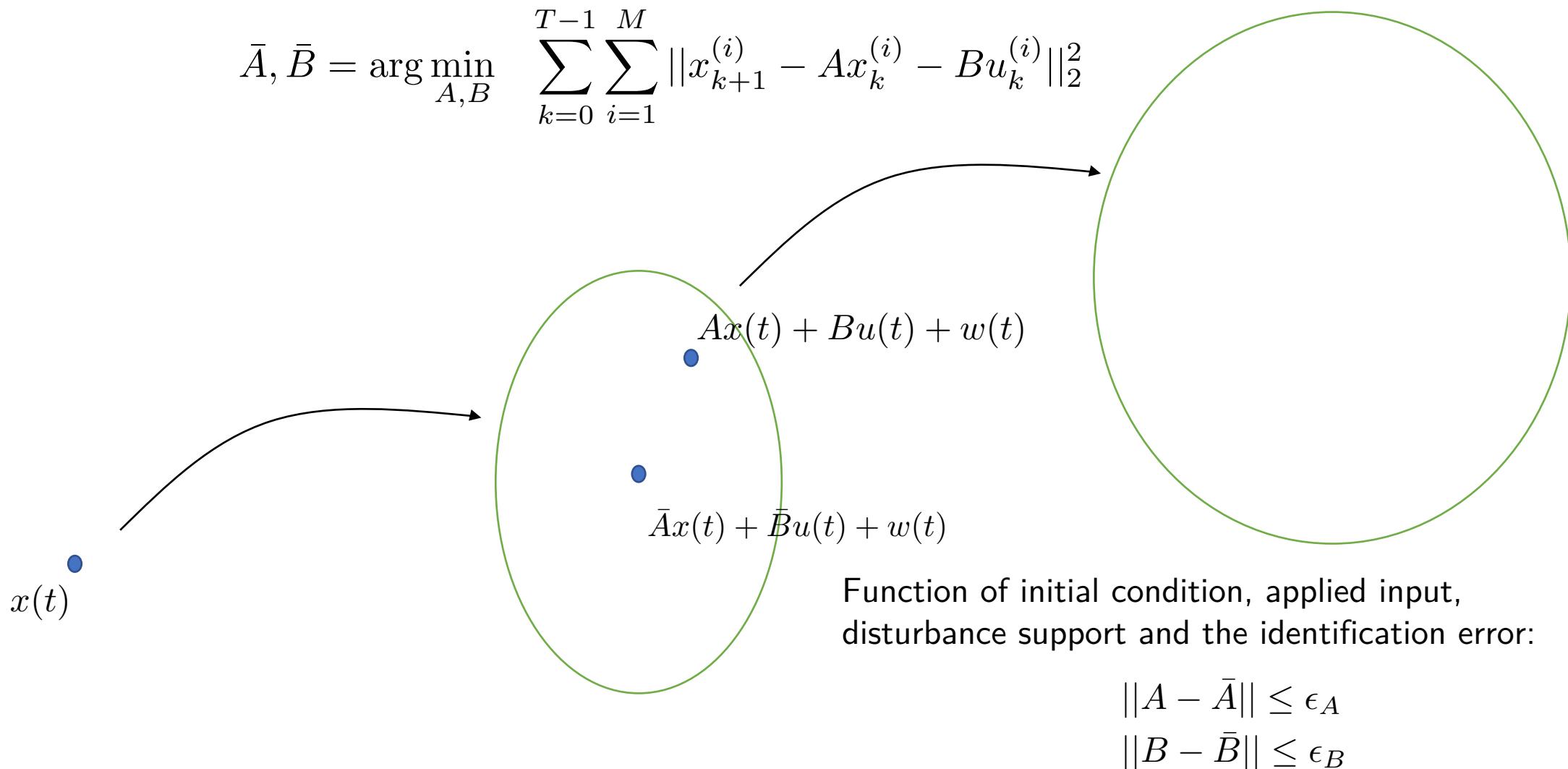
$$\bar{A}, \bar{B} = \arg \min_{A, B} \sum_{k=0}^{T-1} \sum_{i=1}^M \|x_{k+1}^{(i)} - Ax_k^{(i)} - Bu_k^{(i)}\|_2^2$$



Marco C., Campi, and Erik Weyer. "Finite sample properties of system identification methods." *IEEE Transactions on Automatic Control* 47.8 (2002): 1329-1334.

Sarah Dean, Horia Mania, Nikolai Matni, Benjamin Recht, and Stephen Tu. "On the sample complexity of the linear quadratic regulator." *Foundations of Computational Mathematics* (2019): 1-47.

Learning Linear Models: One Step Prediction



Marco C., Campi, and Erik Weyer. "Finite sample properties of system identification methods." *IEEE Transactions on Automatic Control* 47.8 (2002): 1329-1334.

Sarah Dean, Horia Mania, Nikolai Matni, Benjamin Recht, and Stephen Tu. "On the sample complexity of the linear quadratic regulator." *Foundations of Computational Mathematics* (2019): 1-47.

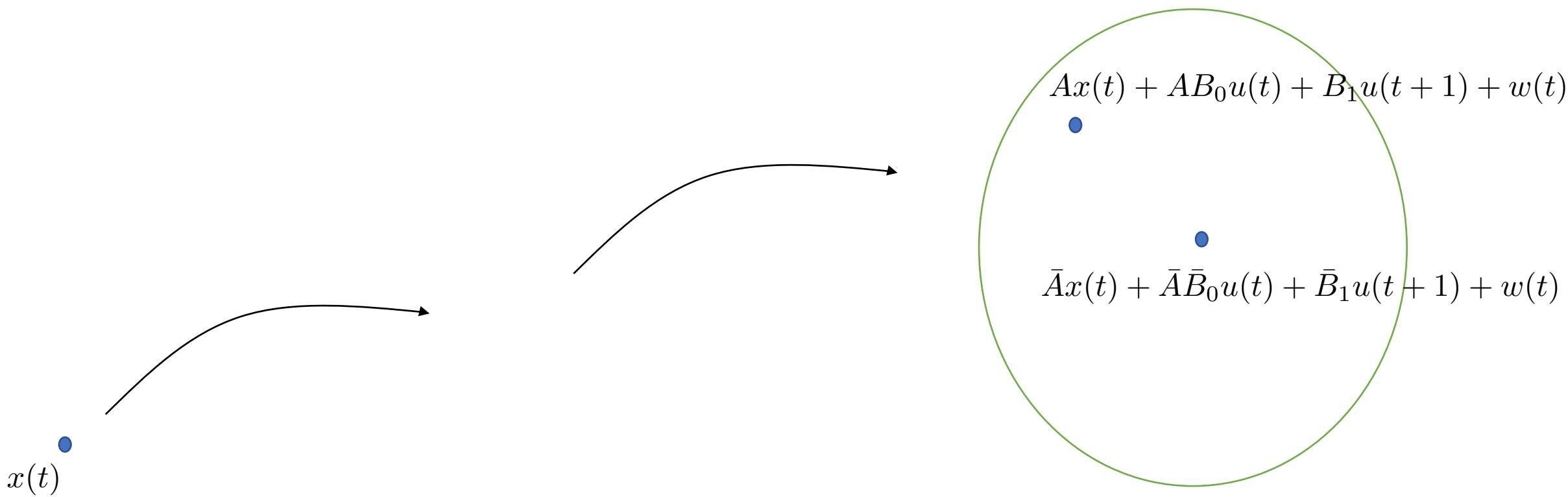
Learning Linear Models: Multi-Step Prediction

Learning Linear Models: Multi-Step Prediction

$$\bar{A}, \bar{B}_0, \dots, \bar{B}_N = \arg \min_{A, B_0, \dots, B_N} \sum_{k=0}^{T-1} \sum_{i=1}^M \|x_{k+N+1}^{(i)} - Ax_k^{(i)} - \sum_{j=0}^N B_j u_{k+j}^{(i)}\|_2^2$$

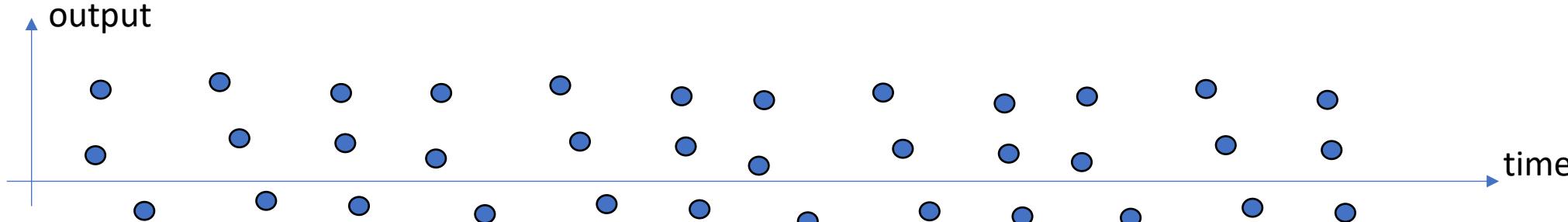
Learning Linear Models: Multi-Step Prediction

$$\bar{A}, \bar{B}_0, \dots, \bar{B}_N = \arg \min_{A, B_0, \dots, B_N} \sum_{k=0}^{T-1} \sum_{i=1}^M \|x_{k+N+1}^{(i)} - Ax_k^{(i)} - \sum_{j=0}^N B_j u_{k+j}^{(i)}\|_2^2$$

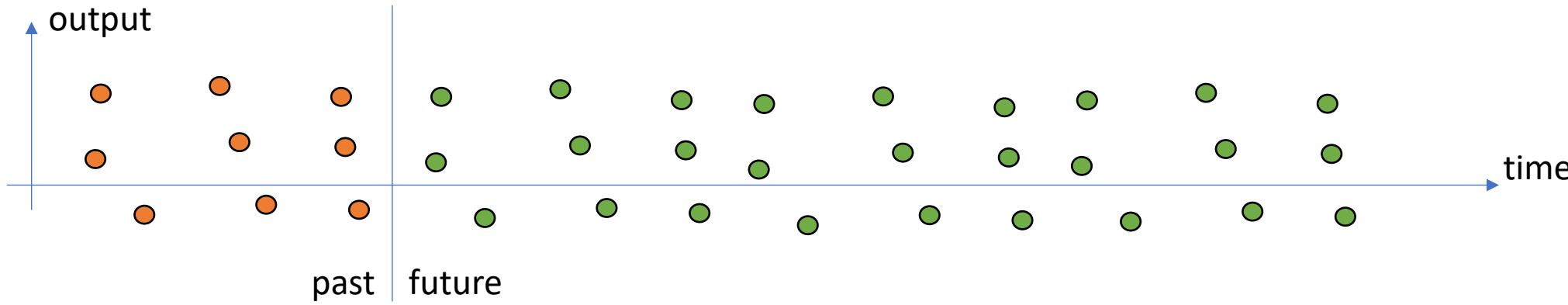


Learning Linear Models: Data-Based Prediction

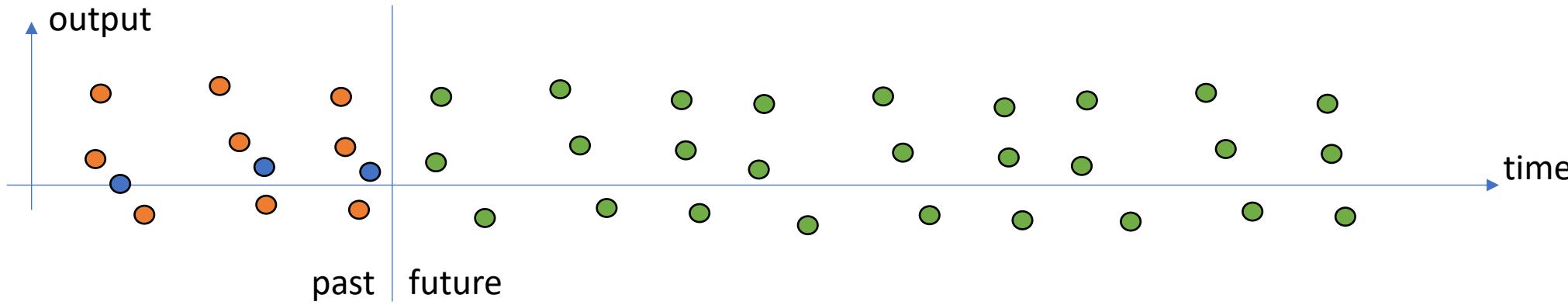
Learning Linear Models: Data-Based Prediction



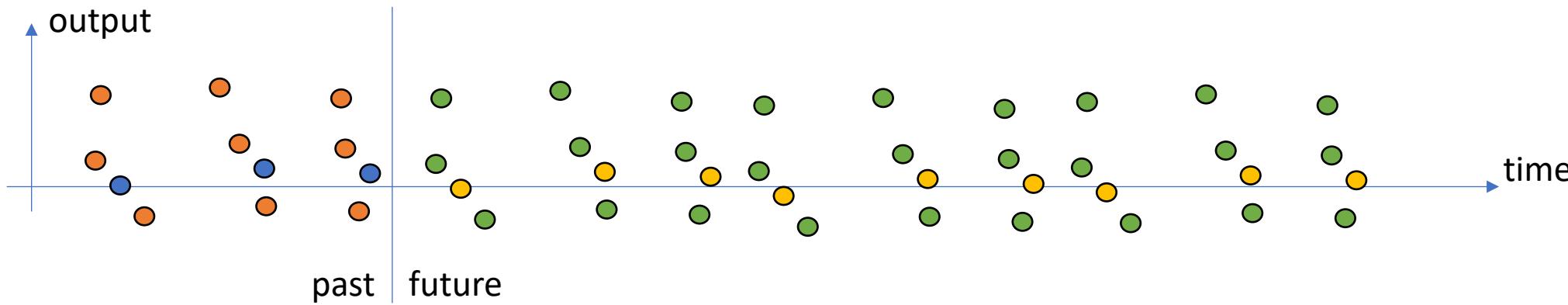
Learning Linear Models: Data-Based Prediction



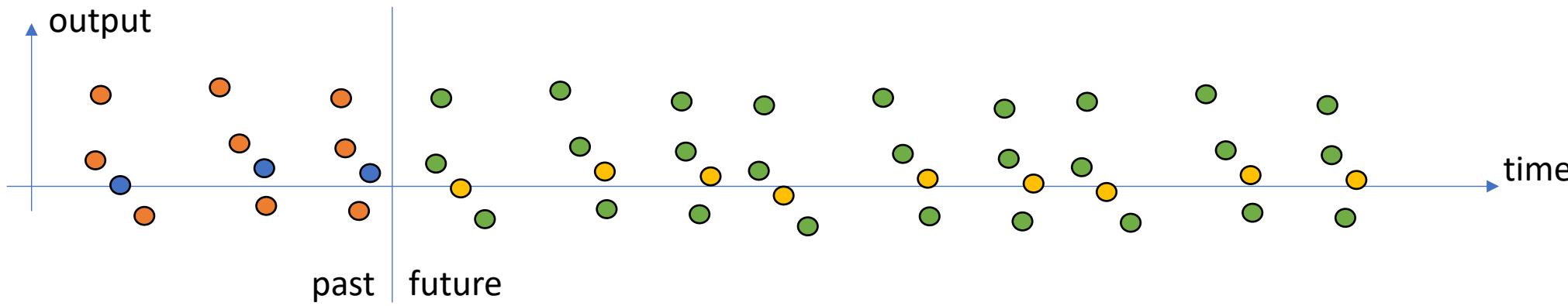
Learning Linear Models: Data-Based Prediction



Learning Linear Models: Data-Based Prediction



Learning Linear Models: Data-Based Prediction

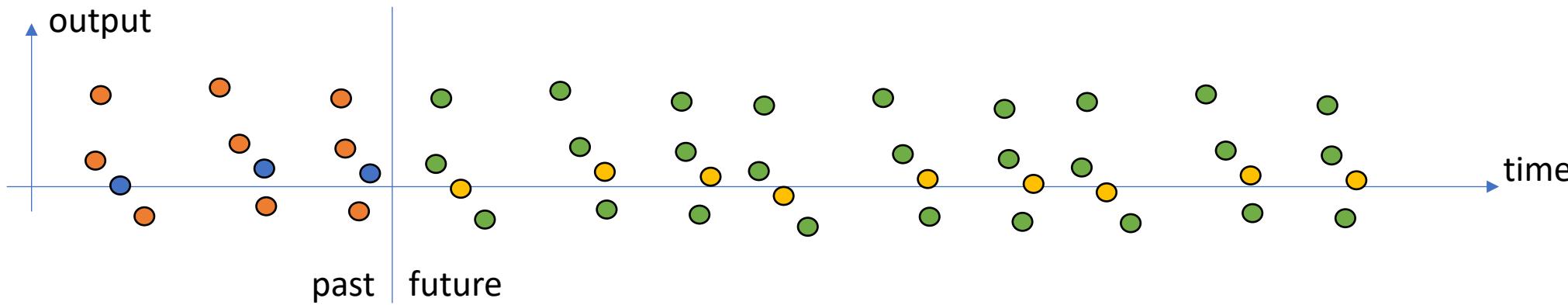


Construct the Hankel Matrix using Data

$$\mathcal{H}_L(u) := \begin{pmatrix} u_1 & \cdots & u_{T-L+1} \\ \vdots & \ddots & \vdots \\ u_L & \cdots & u_T \end{pmatrix}$$

$$\begin{pmatrix} U_p \\ U_f \end{pmatrix} := \mathcal{H}_{T_{\text{ini}}+N}(u^d), \quad \begin{pmatrix} Y_p \\ Y_f \end{pmatrix} := \mathcal{H}_{T_{\text{ini}}+N}(y^d),$$

Learning Linear Models: Data-Based Prediction



Construct the Hankel Matrix using Data

$$\mathcal{H}_L(u) := \begin{pmatrix} u_1 & \cdots & u_{T-L+1} \\ \vdots & \ddots & \vdots \\ u_L & \cdots & u_T \end{pmatrix}$$

$$\begin{pmatrix} U_p \\ U_f \end{pmatrix} := \mathcal{H}_{T_{\text{ini}}+N}(u^d), \quad \begin{pmatrix} Y_p \\ Y_f \end{pmatrix} := \mathcal{H}_{T_{\text{ini}}+N}(y^d),$$

Replace the model

$$\underset{g, u, y}{\text{minimize}} \quad \sum_{k=0}^{N-1} \left(\|y_k - r_{t+k}\|_Q^2 + \|u_k\|_R^2 \right)$$

$$\text{subject to} \quad \begin{pmatrix} U_p \\ Y_p \\ U_f \\ Y_f \end{pmatrix} g = \begin{pmatrix} u_{\text{ini}} \\ y_{\text{ini}} \\ u \\ y \end{pmatrix},$$

$$u_k \in \mathcal{U}, \forall k \in \{0, \dots, N-1\},$$

$$y_k \in \mathcal{Y}, \forall k \in \{0, \dots, N-1\}.$$

The complexity of the prediction model

Linear

$$x_{k+1} = Ax_k + Bu_k + w_k$$

Marco C., Campi, and Erik Weyer. "Finite sample properties of system identification methods." *IEEE Transactions on Automatic Control* 47.8 (2002): 1329-1334.

Sarah Dean, Horia Mania, Nikolai Matni, Benjamin Recht, and Stephen Tu. "On the sample complexity of the linear quadratic regulator." *Foundations of Computational Mathematics* (2019): 1-47.

Google Scholar search results for "learning LQR":
About 8,170 results (0.05 sec)

Articles: About 8,170 results (0.05 sec)

Any time: Trajectory tracking using online learning LQR with adaptive learning control of a leg-exoskeleton for disorder gait rehabilitation
N Aljanaee, M Pamichuk - Mechatronics, 2018 - Elsevier
Precise trajectory tracking of gait pattern under varied load condition is necessary for rehabilitation using leg exoskeleton. In this paper, online iterative learning linear quadratic regulator (OILLQR) with adaptive iterative learning control is proposed to control trajectory ...
52 99 Cited by 17 Related articles All 2 versions

Jeremy Coulson, John Lygeros, and Florian Dörfler. "Data-enabled predictive control: In the shallows of the DeePC." *2019 18th European Control Conference (ECC)*. IEEE, 2019.

Enrico Terzi, Lorenzo Fagiano, Marcello Farina, Riccardo Scattolini. Learning multi-step prediction models for receding horizon control. In *2018 European Control Conference (ECC)* 2018 Jun 12 (pp. 1335-1340). IEEE.

known Gray Box unknown

$$x_{k+1} = f(x_k, u_k) + g(x_k, u_k) + w_k$$

Anil Aswani, Humberto Gonzalez, S. Shankar Sastry, and Claire Tomlin. "Provably safe and robust learning-based model predictive control." *Automatica* 49, no. 5 (2013)

Lukas Hewing, Kim P. Wabersich, Marcel Menner, and Melanie N. Zeilinger. "Learning-based model predictive control: Toward safe learning in control." *Annual Review of Control, Robotics, and Autonomous Systems* 3 (2020):

Lukas Hewing, Juraj Kabzan, and Melanie N. Zeilinger. "Cautious model predictive control using Gaussian process regression." *IEEE Transactions on Control Systems Technology* 28, no. 6 (2019): 2736-2743.

Milanese, Mario, and Carlo Novara. "Set membership identification of nonlinear systems." *Automatica* 40, no. 6 (2004): 957-975.

Marko Tanaskovic, Lorenzo Fagiano, Roy Smith, Paul Goulart, and Manfred Morari. "Adaptive model predictive control for constrained linear systems." In *2013 European Control Conference (ECC)*, pp. 382-387. IEEE, 2013.

Monimoy Bujarbaruah, Xiaojing Zhang, Ugo Rosolia, and Francesco Borrelli. "Adaptive MPC for iterative tasks." In *2018 IEEE Conference on Decision and Control (CDC)*, pp. 6322-6327. IEEE, 2018.

Black Box

$$x_{k+1} = x_k + g(x_k, u_k, w_k)$$

Chua, Kurtland, Roberto Calandra, Rowan McAllister, and Sergey Levine. "Deep reinforcement learning in a handful of trials using probabilistic dynamics models." *NeurIPS* (2018).

Anusha Nagabandi, Gregory Kahn, Ronald S. Fearing, and Sergey Levine. "Neural network dynamics for model-based deep reinforcement learning with model-free fine-tuning." In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 7559-7566. IEEE, 2018.

Lenz, Ian, Ross A. Knepper, and Ashutosh Saxena. "DeepMPC: Learning deep latent features for model predictive control." In *Robotics: Science and Systems*. 2015.

Kendall Lowrey, Aravind Rajeswaran, Sham Kakade, Emanuel Todorov, and Igor Mordatch. "Plan online, learn offline: Efficient learning and exploration via model-based control." In *Proc. Int. Conf. Mach. Learn.*, 2019.

Brijen Thananjeyan, Ashwin Balakrishna, Ugo Rosolia, Felix Li, Rowan McAllister, Joseph E. Gonzalez, Sergey Levine, Francesco Borrelli, and Ken Goldberg. "Safety augmented value estimation from demonstrations (saved): Safe deep model-based RL for sparse cost robotic tasks." *IEEE Robotics and Automation Letters* 5, no. 2 (2020).

Gray Box Model

$$x_{k+1} = f(x_k, u_k) + g(x_k, u_k) + w_k$$

Gray Box Model

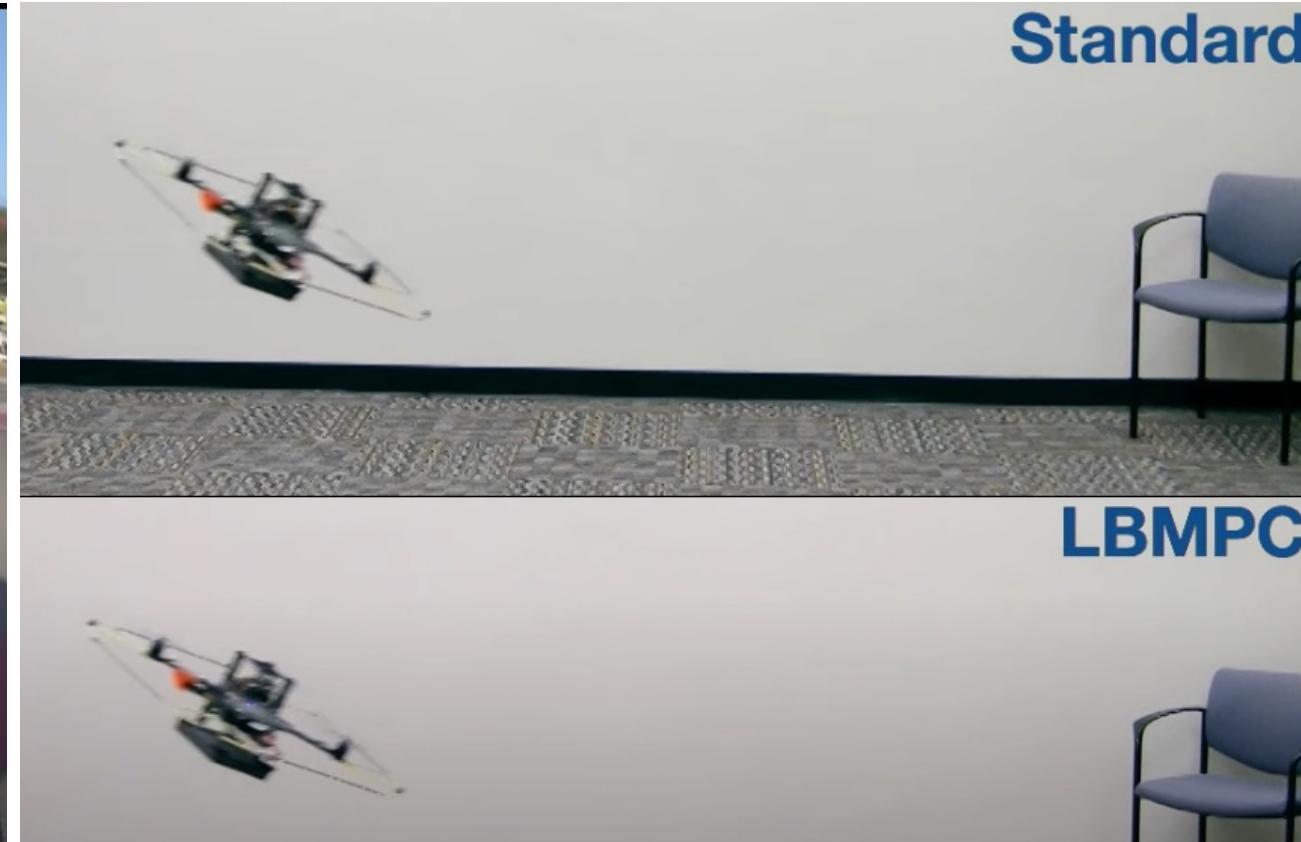
Most popular method is Gaussian Process

$$x_{k+1} = f(x_k, u_k) + g(x_k, u_k) + w_k$$

Gray Box Model

Most popular method is Gaussian Process

$$x_{k+1} = f(x_k, u_k) + g(x_k, u_k) + w_k$$



Juraj Kabzan,, Lukas Hewing, Alexander Liniger, and Melanie N. Zeilinger. "Learning-based model predictive control for autonomous racing." *IEEE Robotics and Automation Letters* 4, no. 4 (2019): 3363-3370.

Bouffard, Patrick, Anil Aswani, and Claire Tomlin. "Learning-based model predictive control on a quadrotor: Onboard implementation and experimental results." In *2012 IEEE International Conference on Robotics and Automation*, pp. 279-284. IEEE, 2012.

The complexity of the prediction model

Linear

$$x_{k+1} = Ax_k + Bu_k + w_k$$

Marco C., Campi, and Erik Weyer. "Finite sample properties of system identification methods." *IEEE Transactions on Automatic Control* 47.8 (2002): 1329-1334.

Sarah Dean, Horia Mania, Nikolai Matni, Benjamin Recht, and Stephen Tu. "On the sample complexity of the linear quadratic regulator." *Foundations of Computational Mathematics* (2019): 1-47.

Google Scholar search results for "learning LQR".
Articles: About 8,170 results (0.05 sec).
Filter: Any time, Since 2021.
Result 1: Trajectory tracking using online learning LQR with adaptive learning control of a leg-exoskeleton for disorder gait rehabilitation by N Aljanaromat, M Pamichuk - Mechantronics, 2018 - Elsevier.
Result 2: Precise trajectory tracking of gait pattern under varied load condition is necessary for rehabilitation using leg exoskeleton. In this paper, online iterative learning linear quadratic regulator (OILLQR) with adaptive iterative learning control is proposed to control trajectory ...
Cited by 17 Related articles All 2 versions

Jeremy Coulson, John Lygeros, and Florian Dörfler. "Data-enabled predictive control: In the shallows of the DeePC." *2019 18th European Control Conference (ECC)*. IEEE, 2019.

Enrico Terzi, Lorenzo Fagiano, Marcello Farina, Riccardo Scattolini. Learning multi-step prediction models for receding horizon control. In *2018 European Control Conference (ECC)* 2018 Jun 12 (pp. 1335-1340). IEEE.

Gray Box

$$x_{k+1} = f(x_k, u_k) + g(x_k, u_k) + w_k$$

Anil Aswani, Humberto Gonzalez, S. Shankar Sastry, and Claire Tomlin. "Provably safe and robust learning-based model predictive control." *Automatica* 49, no. 5 (2013)

Lukas Hewing, Kim P. Wabersich, Marcel Menner, and Melanie N. Zeilinger. "Learning-based model predictive control: Toward safe learning in control." *Annual Review of Control, Robotics, and Autonomous Systems* 3 (2020):

Lukas Hewing, Juraj Kabzan, and Melanie N. Zeilinger. "Cautious model predictive control using Gaussian process regression." *IEEE Transactions on Control Systems Technology* 28, no. 6 (2019): 2736-2743.

Milanese, Mario, and Carlo Novara. "Set membership identification of nonlinear systems." *Automatica* 40, no. 6 (2004): 957-975.

Marko Tanaskovic, Lorenzo Fagiano, Roy Smith, Paul Goulart, and Manfred Morari. "Adaptive model predictive control for constrained linear systems." In *2013 European Control Conference (ECC)*, pp. 382-387. IEEE, 2013.

Monimoy Bujarbaruah, Xiaojing Zhang, Ugo Rosolia, and Francesco Borrelli. "Adaptive MPC for iterative tasks." In *2018 IEEE Conference on Decision and Control (CDC)*, pp. 6322-6327. IEEE, 2018.

Black Box

$$x_{k+1} = x_k + g(x_k, u_k, w_k)$$

Chua, Kurtland, Roberto Calandra, Rowan McAllister, and Sergey Levine. "Deep reinforcement learning in a handful of trials using probabilistic dynamics models." *NeurIPS* (2018).

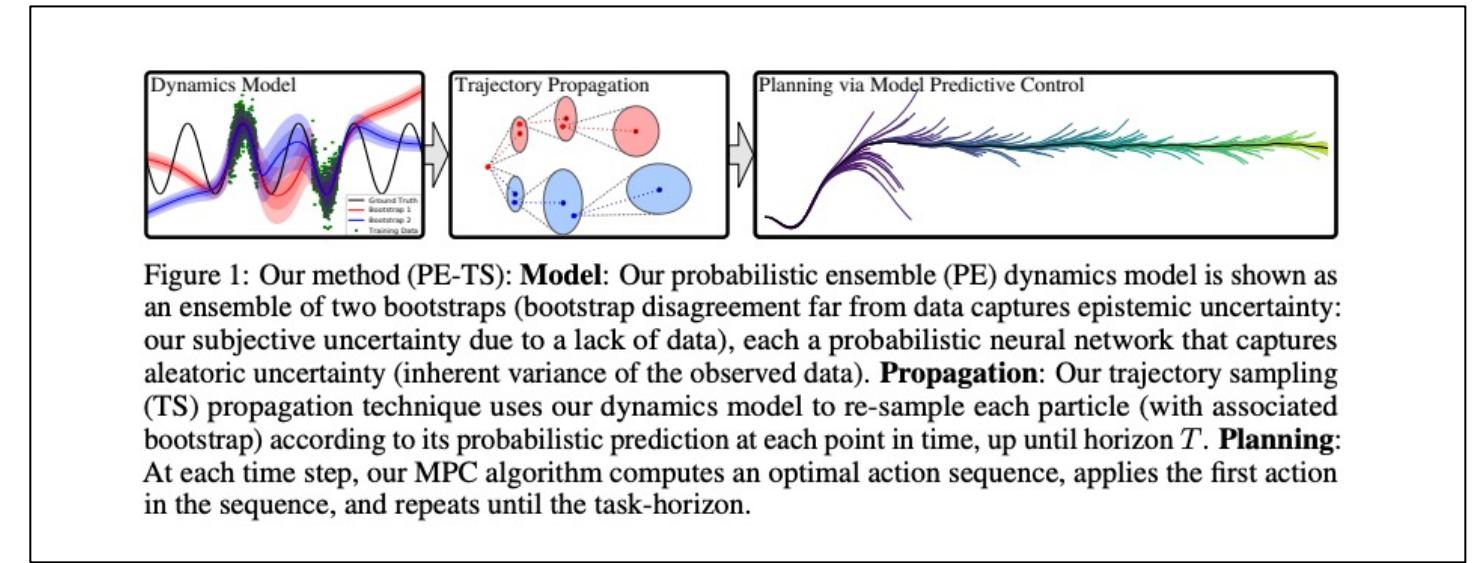
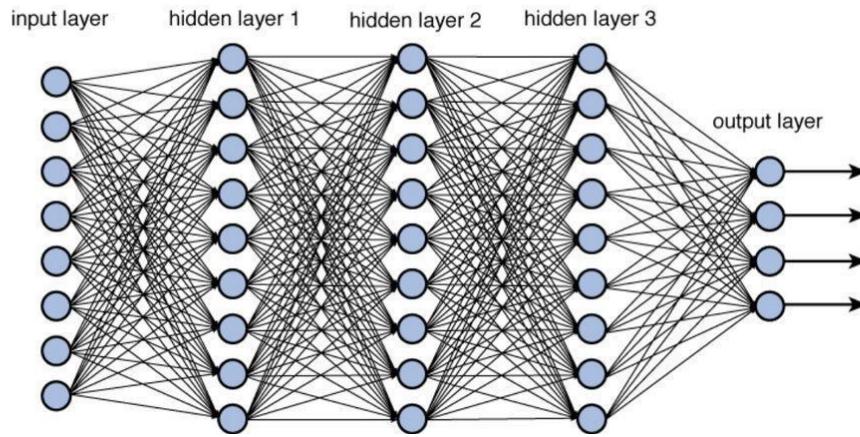
Anusha Nagabandi, Gregory Kahn, Ronald S. Fearing, and Sergey Levine. "Neural network dynamics for model-based deep reinforcement learning with model-free fine-tuning." In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 7559-7566. IEEE, 2018.

Lenz, Ian, Ross A. Knepper, and Ashutosh Saxena. "DeepMPC: Learning deep latent features for model predictive control." In *Robotics: Science and Systems*. 2015.

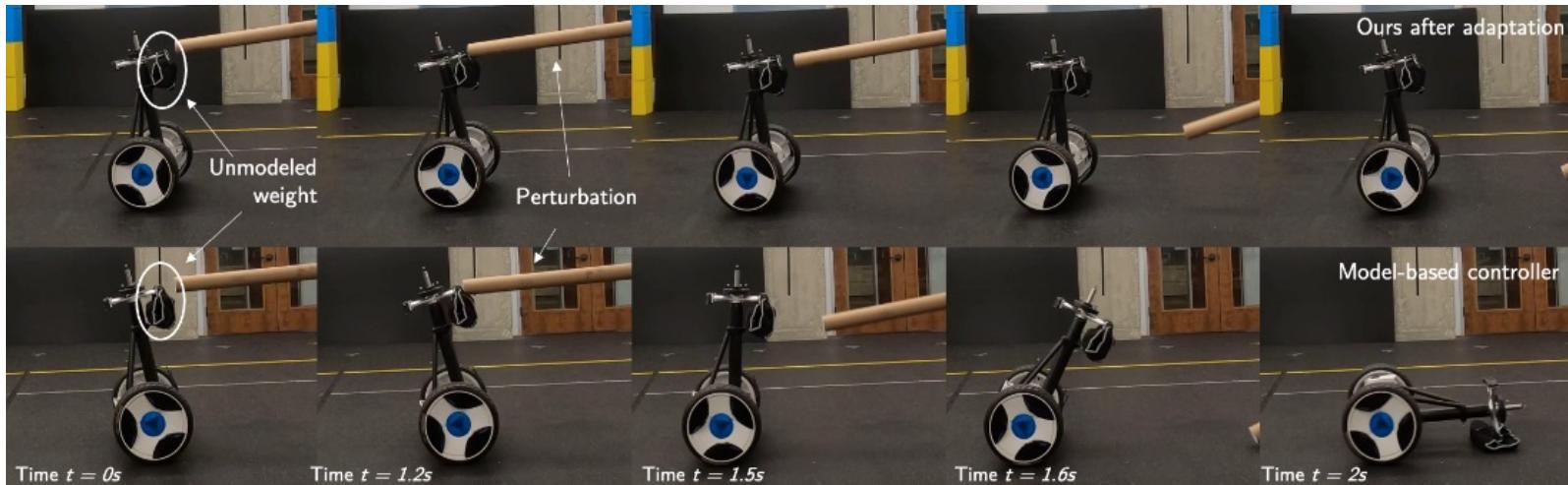
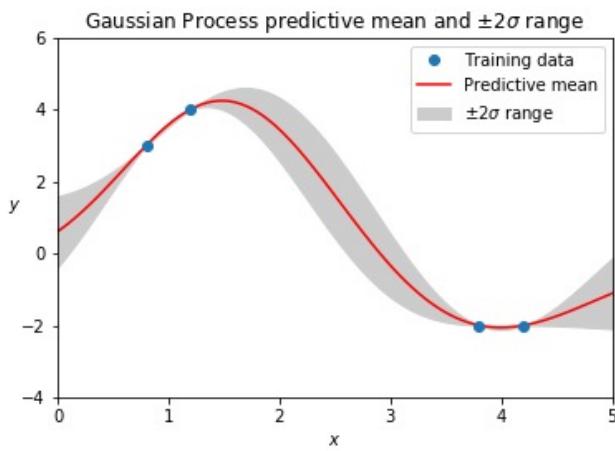
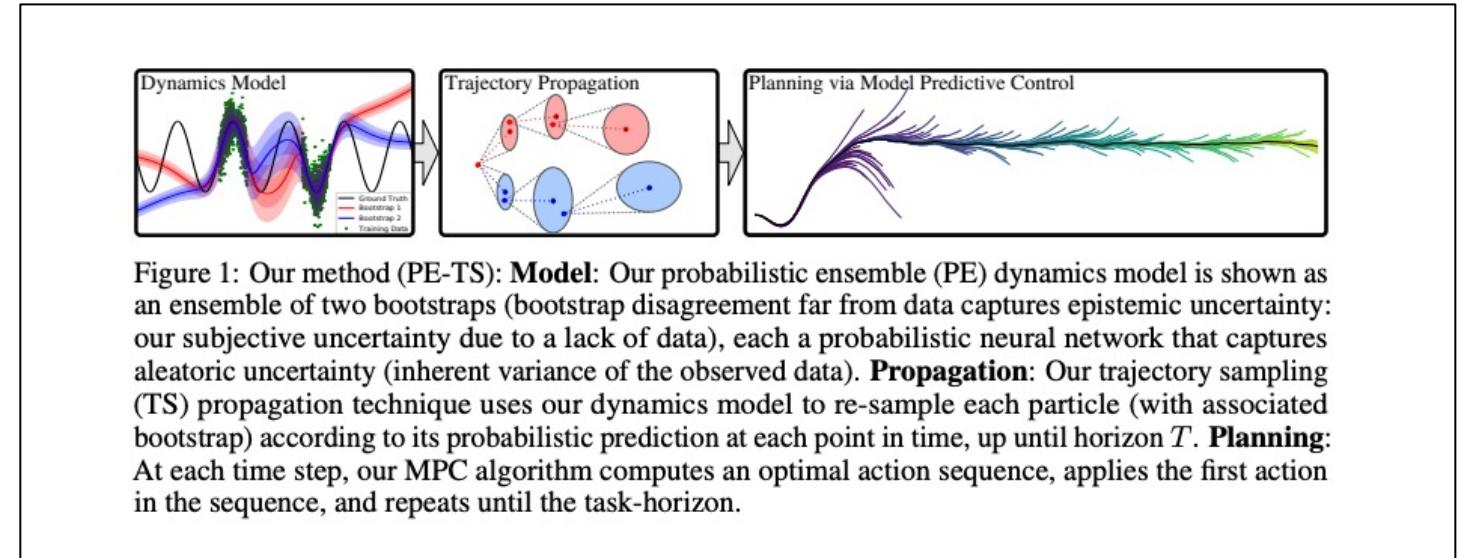
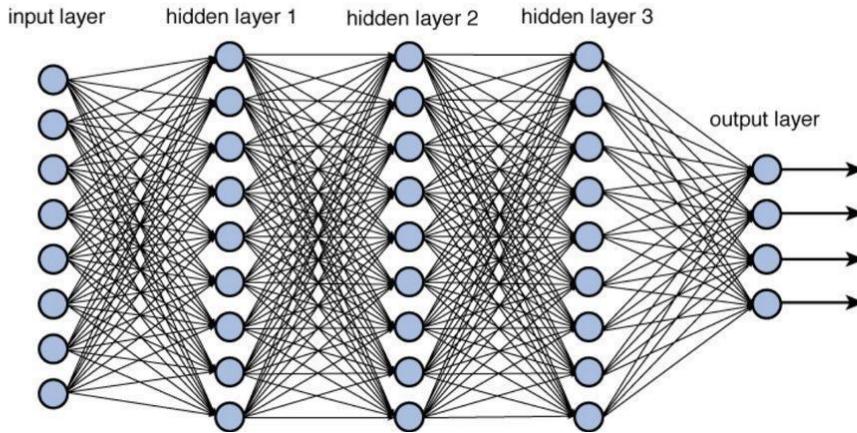
Deisenroth, Marc, and Carl E. Rasmussen. "PILCO: A model-based and data-efficient approach to policy search." In *Proceedings of the 28th International Conference on machine learning (ICML-11)*, pp. 465-472. 2011.

Brijen Thananjeyan, Ashwin Balakrishna, Ugo Rosolia, Felix Li, Rowan McAllister, Joseph E. Gonzalez, Sergey Levine, Francesco Borrelli, and Ken Goldberg. "Safety augmented value estimation from demonstrations (saved): Safe deep model-based RL for sparse cost robotic tasks." *IEEE Robotics and Automation Letters* 5, no. 2 (2020).

Black Box Model



Black Box Model



Chua, Kurtland, Roberto Calandra, Rowan McAllister, and Sergey Levine. "Deep reinforcement learning in a handful of trials using probabilistic dynamics models." NeurIPS (2018).

Ivan D. Jimenez, Rodriguez, Ugo Rosolia, Aaron D. Ames, and Yisong Yue. "Learning Unstable Dynamics with One Minute of Data: A Differentiation-based Gaussian Process Approach." *arXiv preprint arXiv:2103.04548* (2021).

Planning under uncertainty

Optimizing over control policies

Why is planning in uncertain environments harder?

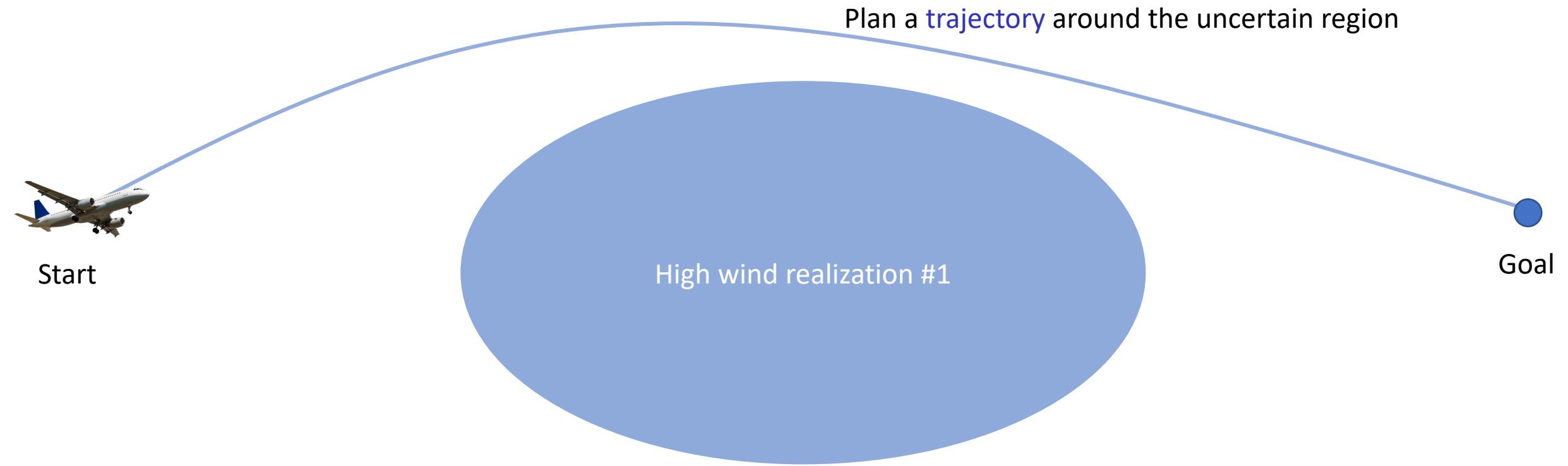


Start



Goal

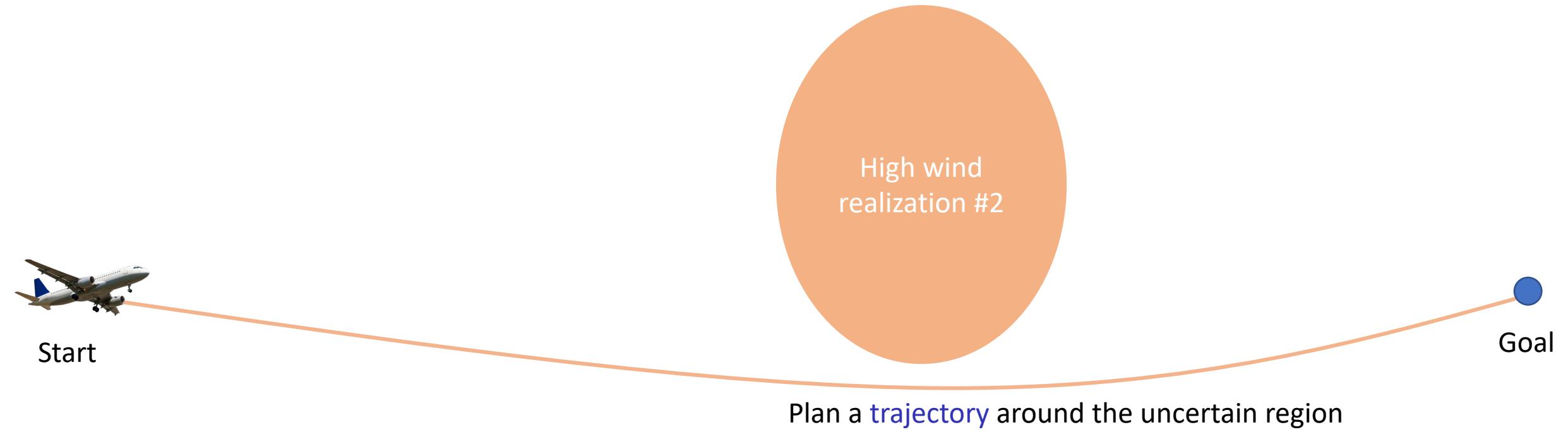
Why is planning in uncertain environments harder?



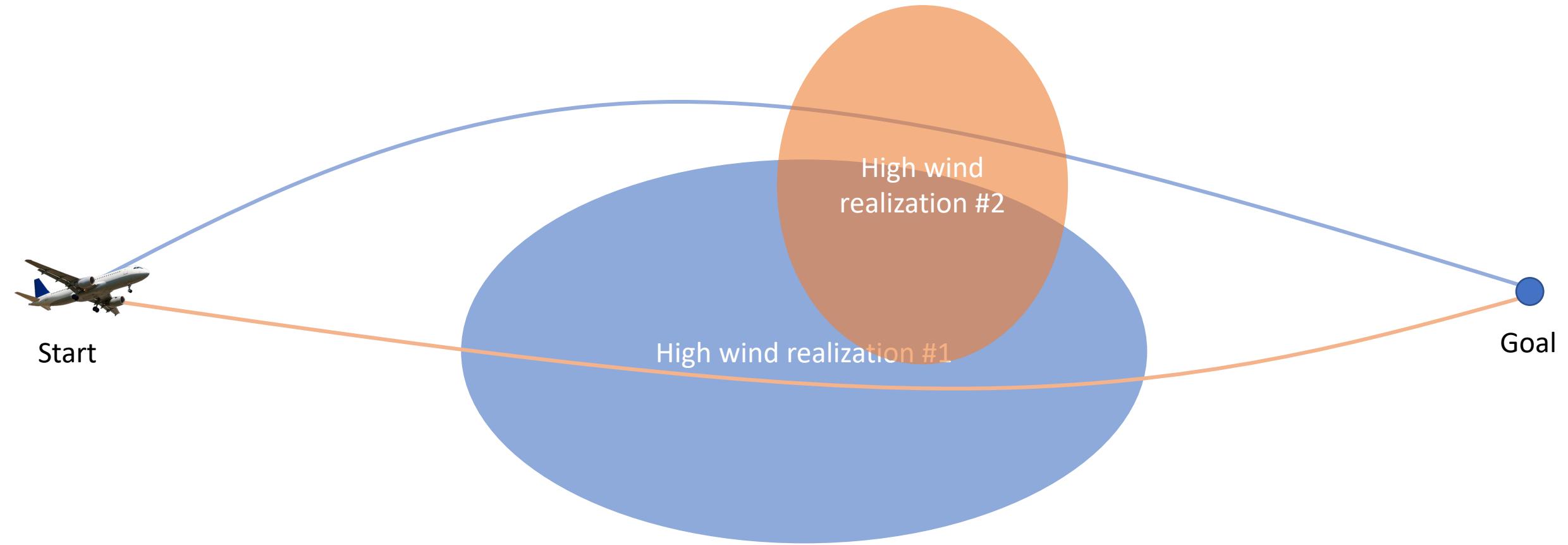
Why is planning in uncertain environments harder?



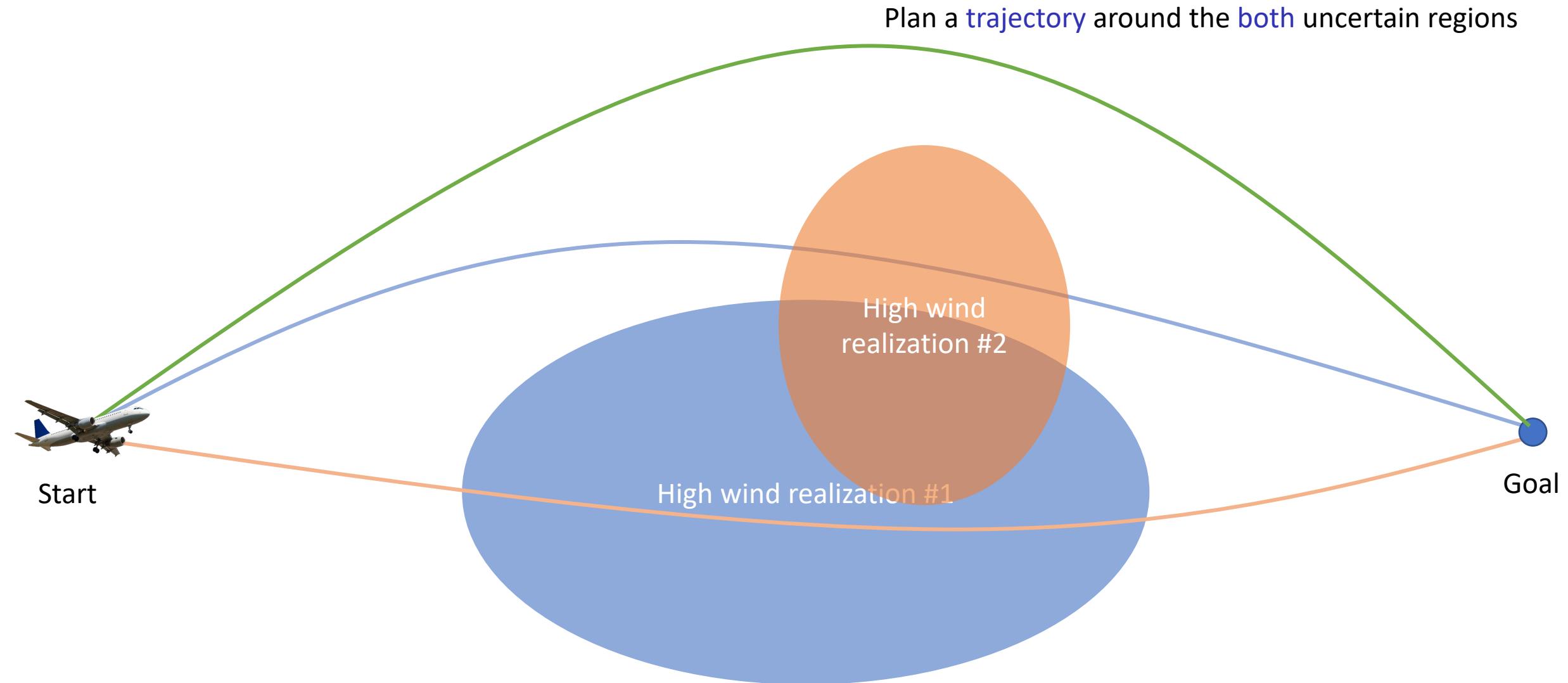
Why is planning in uncertain environments harder?



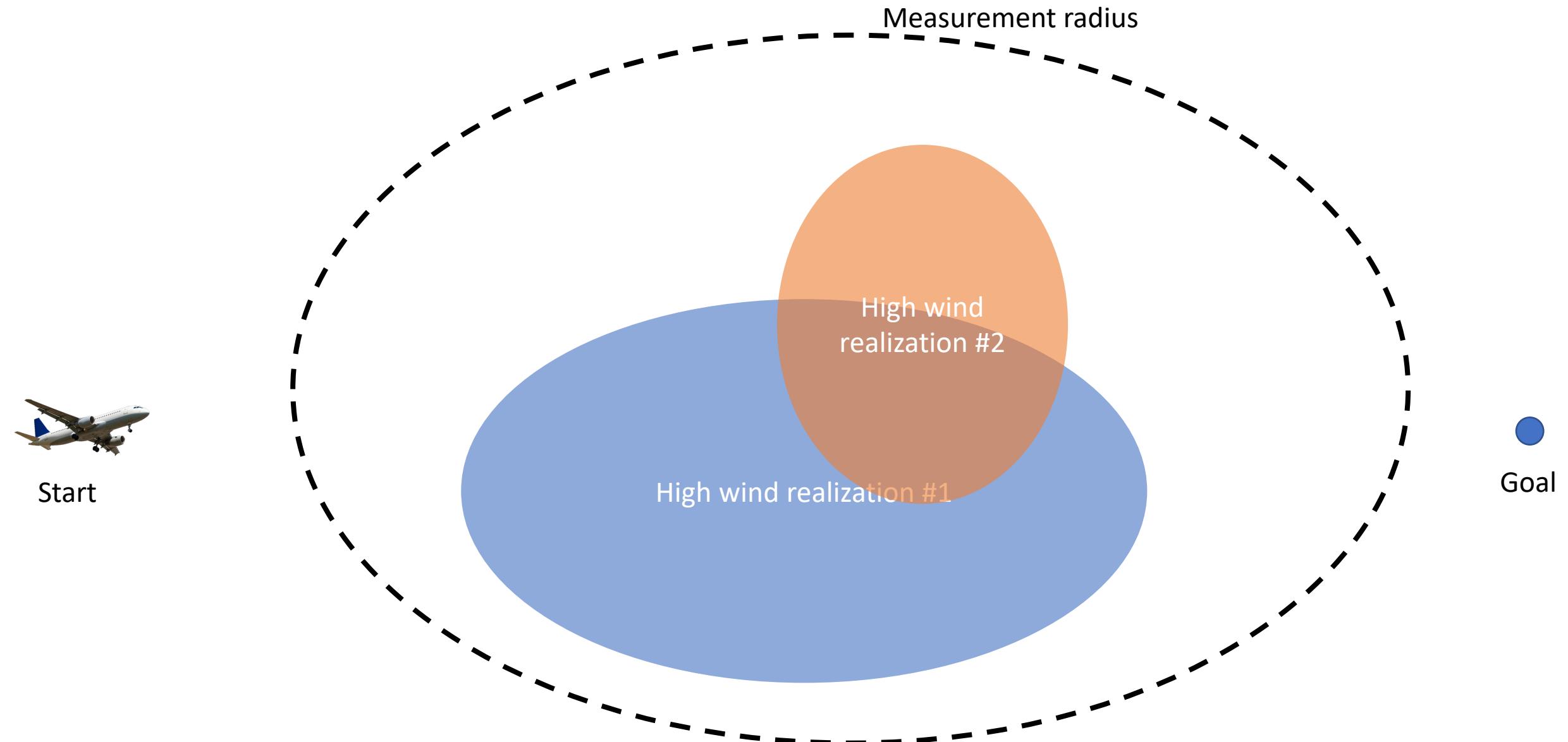
Why is planning in uncertain environments harder?



Why is planning in uncertain environments harder?

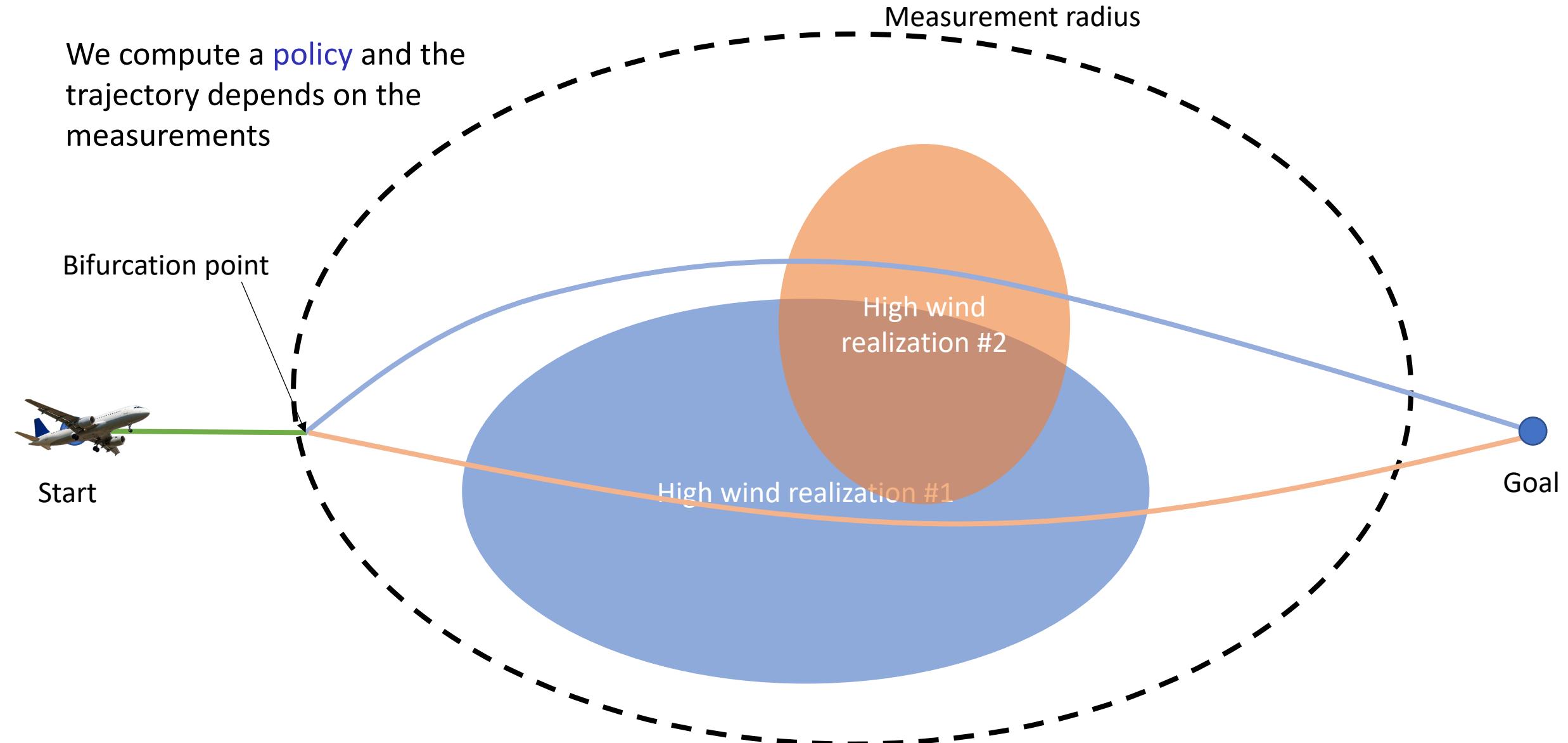


Why is planning in uncertain environments harder?



Why is planning in uncertain environments harder?

We compute a **policy** and the trajectory depends on the measurements



Problem Formulation

True system dynamics: $x_{k+1} = Ax_k + Bu_k + w_k$

Problem Formulation

True system dynamics: $x_{k+1} = Ax_k + Bu_k + w_k$

Nominal model: $\bar{x}_{k+1} = \bar{A}\bar{x}_k + \bar{B}\bar{u}_k + w_k$ where $\|A - \bar{A}\| \leq \epsilon_A, \|B - \bar{B}\| \leq \epsilon_B$

Problem Formulation

True system dynamics: $x_{k+1} = Ax_k + Bu_k + w_k$

Nominal model: $\bar{x}_{k+1} = \bar{A}\bar{x}_k + \bar{B}\bar{u}_k + w_k$ where $\|A - \bar{A}\| \leq \epsilon_A, \|B - \bar{B}\| \leq \epsilon_B$

Robust constraints:

Given a control policy $\pi : \mathbb{R}^n \rightarrow \mathbb{R}^d$ designed using the nominal model, we want to guarantee that the closed-loop system

$$x_{k+1} = Ax_k + B\pi(x_k) + w_k$$

satisfies $x_k \in \mathcal{X}, \pi(x_k) \in \mathcal{U}$ for all $w_k \in \mathcal{W}$.

Problem Formulation

True system dynamics: $x_{k+1} = Ax_k + Bu_k + w_k$

Nominal model: $\bar{x}_{k+1} = \bar{A}\bar{x}_k + \bar{B}\bar{u}_k + w_k$ where $\|A - \bar{A}\| \leq \epsilon_A, \|B - \bar{B}\| \leq \epsilon_B$

Robust constraints:

Given a control policy $\pi : \mathbb{R}^n \rightarrow \mathbb{R}^d$ designed using the nominal model, we want to guarantee that the closed-loop system

$$x_{k+1} = Ax_k + B\pi(x_k) + w_k$$

satisfies $x_k \in \mathcal{X}, \pi(x_k) \in \mathcal{U}$ for all $w_k \in \mathcal{W}$.

Can we leverage the MPC approach?

Problem Formulation

True system dynamics: $x_{k+1} = Ax_k + Bu_k + w_k$

Nominal model: $\bar{x}_{k+1} = \bar{A}\bar{x}_k + \bar{B}\bar{u}_k + w_k$ where $\|A - \bar{A}\| \leq \epsilon_A, \|B - \bar{B}\| \leq \epsilon_B$

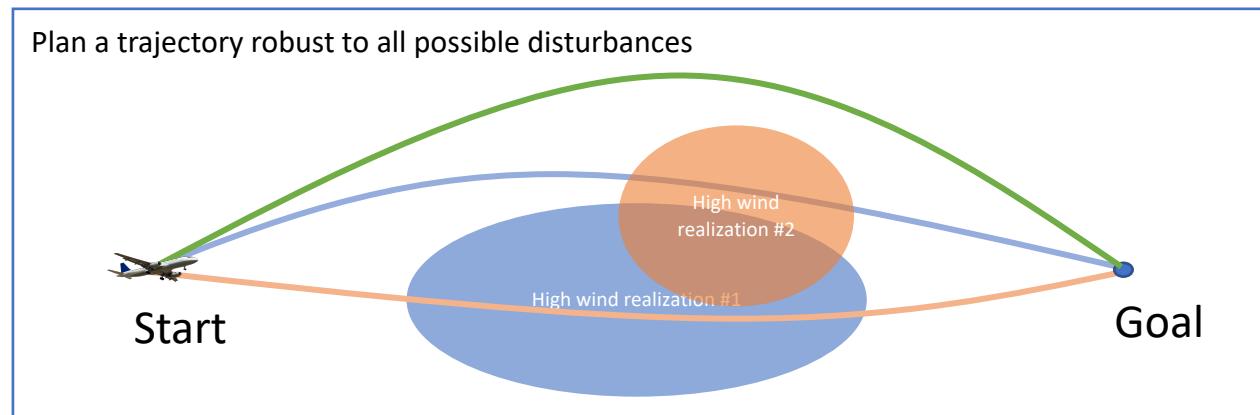
Robust constraints:

Given a control policy $\pi : \mathbb{R}^n \rightarrow \mathbb{R}^d$ designed using the nominal model, we want to guarantee that the closed-loop system

$$x_{k+1} = Ax_k + B\pi(x_k) + w_k$$

satisfies $x_k \in \mathcal{X}, \pi(x_k) \in \mathcal{U}$ for all $w_k \in \mathcal{W}$.

Can we leverage the MPC approach?



Problem Formulation

True system dynamics: $x_{k+1} = Ax_k + Bu_k + w_k$

Nominal model: $\bar{x}_{k+1} = \bar{A}\bar{x}_k + \bar{B}\bar{u}_k + w_k$ where $\|A - \bar{A}\| \leq \epsilon_A, \|B - \bar{B}\| \leq \epsilon_B$

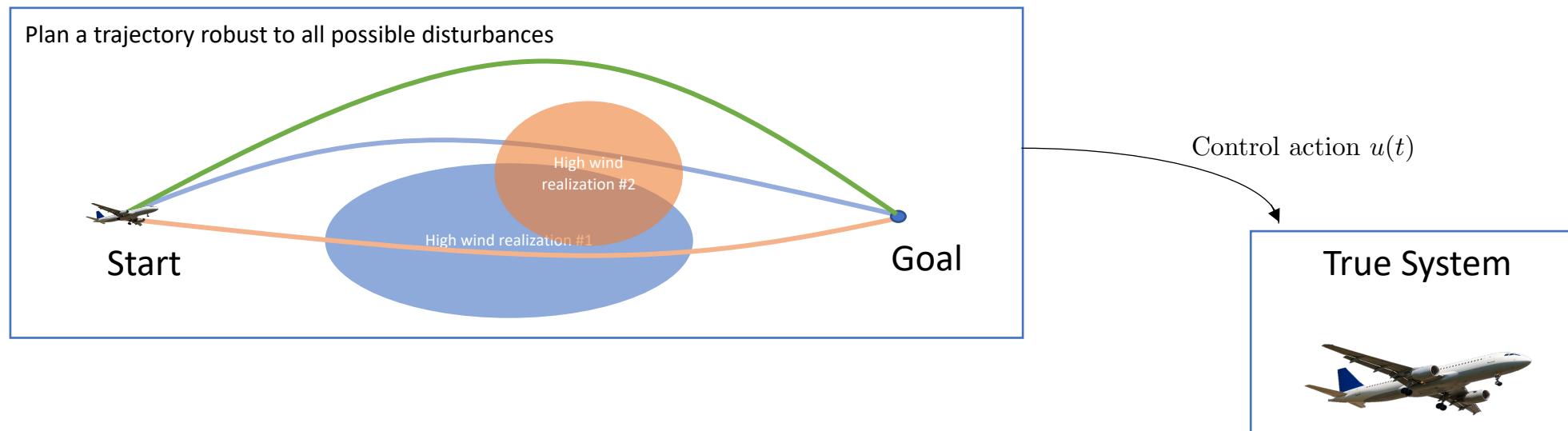
Robust constraints:

Given a control policy $\pi : \mathbb{R}^n \rightarrow \mathbb{R}^d$ designed using the nominal model, we want to guarantee that the closed-loop system

$$x_{k+1} = Ax_k + B\pi(x_k) + w_k$$

satisfies $x_k \in \mathcal{X}, \pi(x_k) \in \mathcal{U}$ for all $w_k \in \mathcal{W}$.

Can we leverage the MPC approach?



Problem Formulation

True system dynamics: $x_{k+1} = Ax_k + Bu_k + w_k$

Nominal model: $\bar{x}_{k+1} = \bar{A}\bar{x}_k + \bar{B}\bar{u}_k + w_k$ where $\|A - \bar{A}\| \leq \epsilon_A, \|B - \bar{B}\| \leq \epsilon_B$

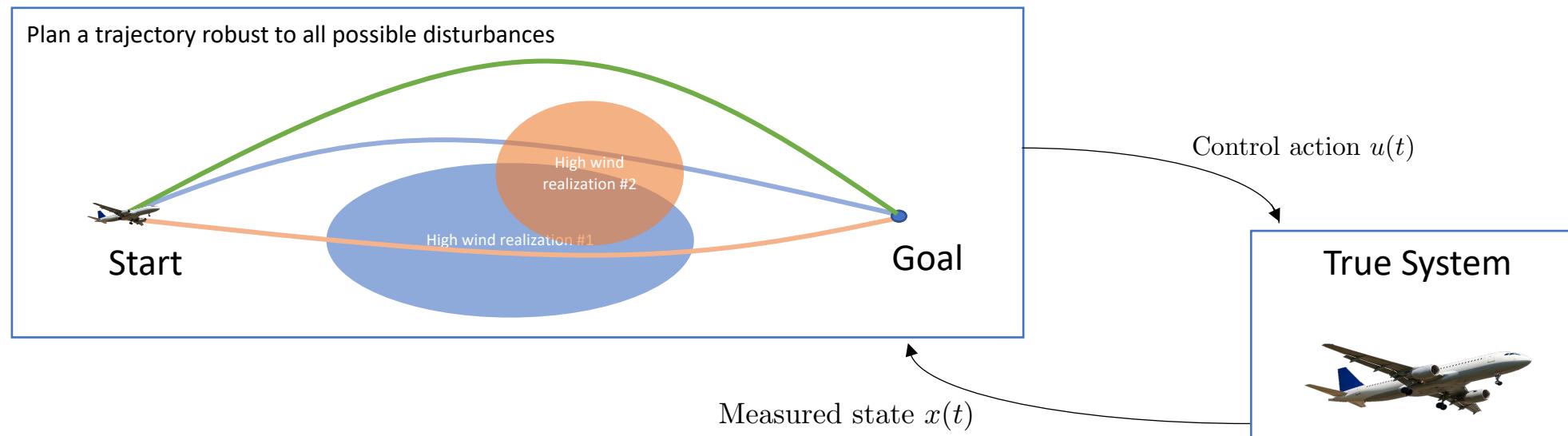
Robust constraints:

Given a control policy $\pi : \mathbb{R}^n \rightarrow \mathbb{R}^d$ designed using the nominal model, we want to guarantee that the closed-loop system

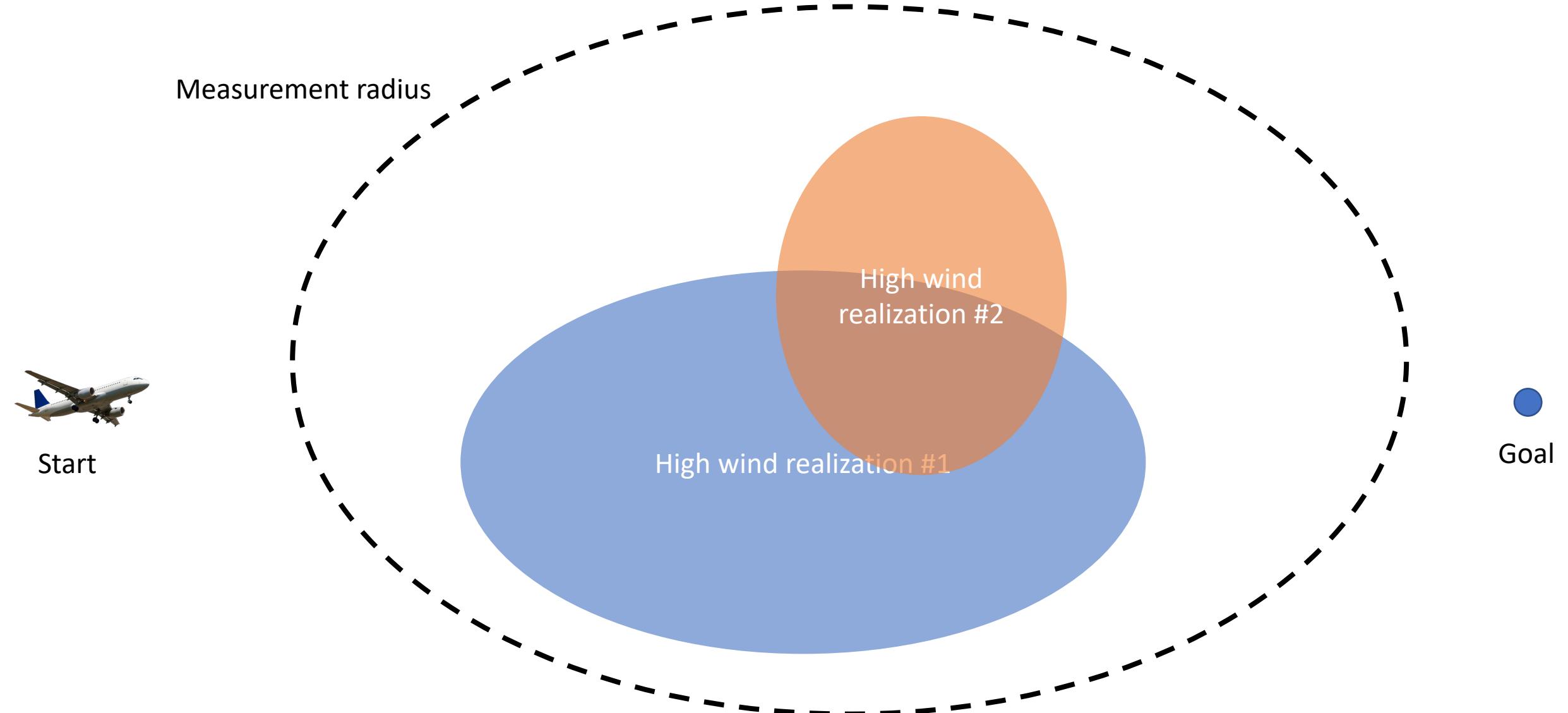
$$x_{k+1} = Ax_k + B\pi(x_k) + w_k$$

satisfies $x_k \in \mathcal{X}, \pi(x_k) \in \mathcal{U}$ for all $w_k \in \mathcal{W}$.

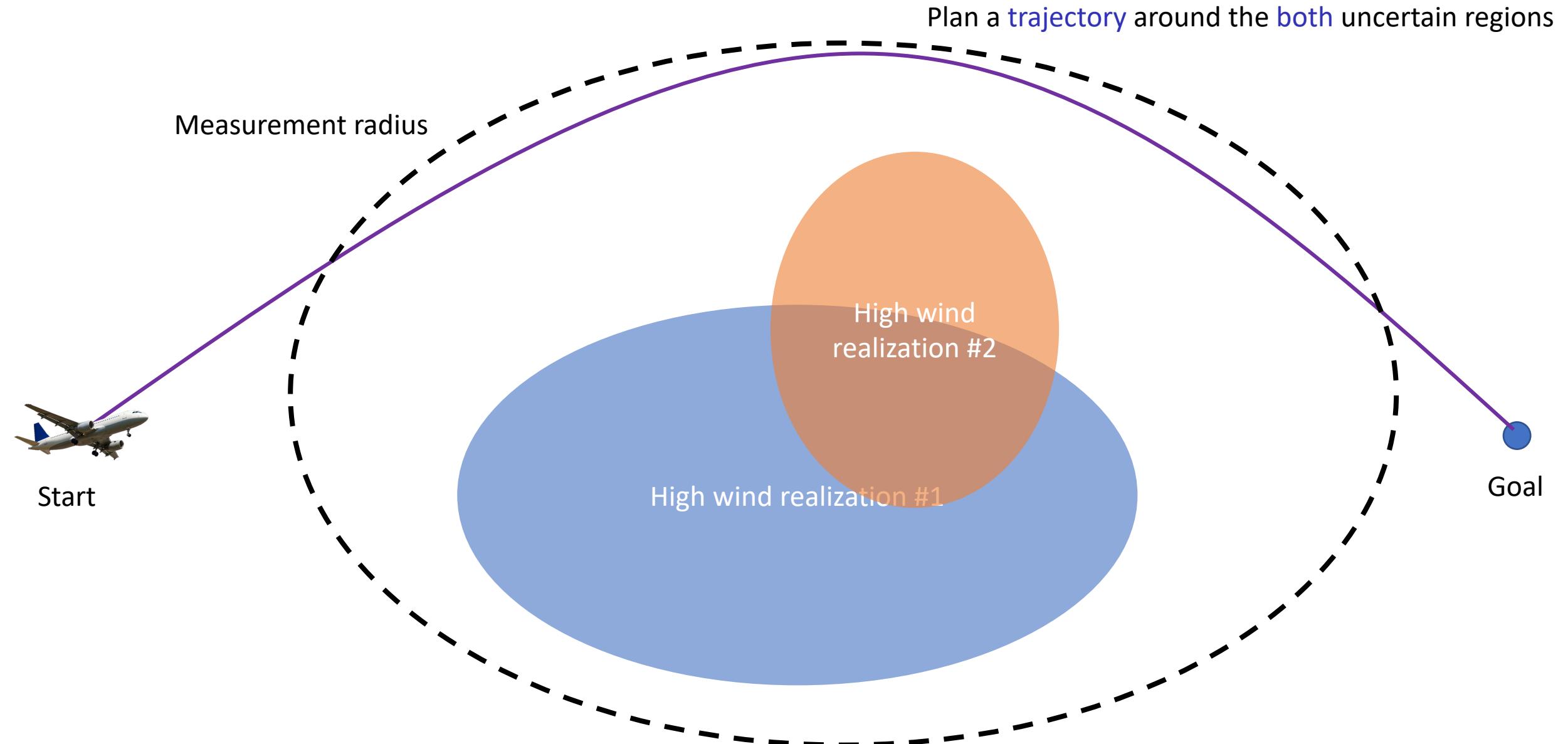
Can we leverage the MPC approach?



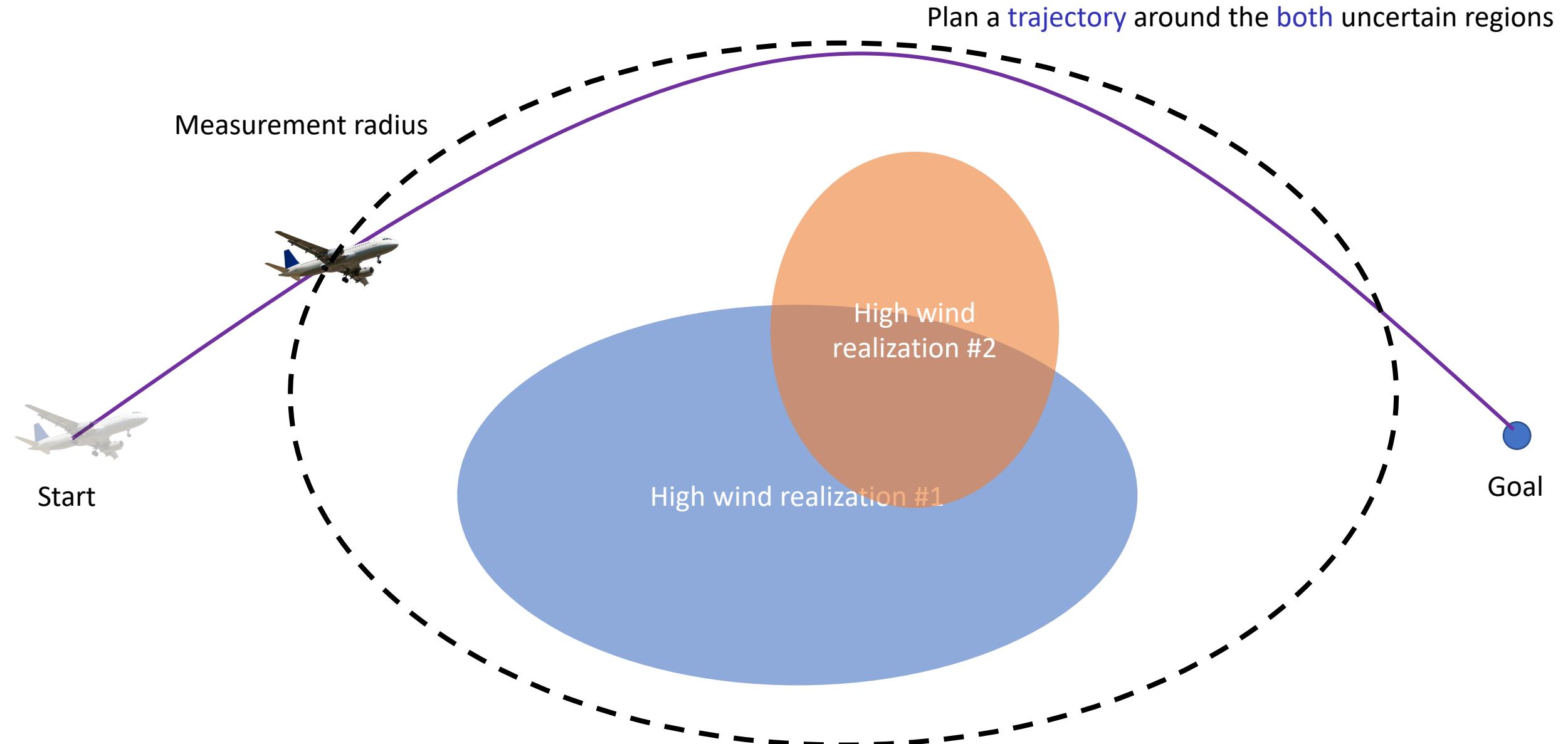
Can we leverage the MPC approach?



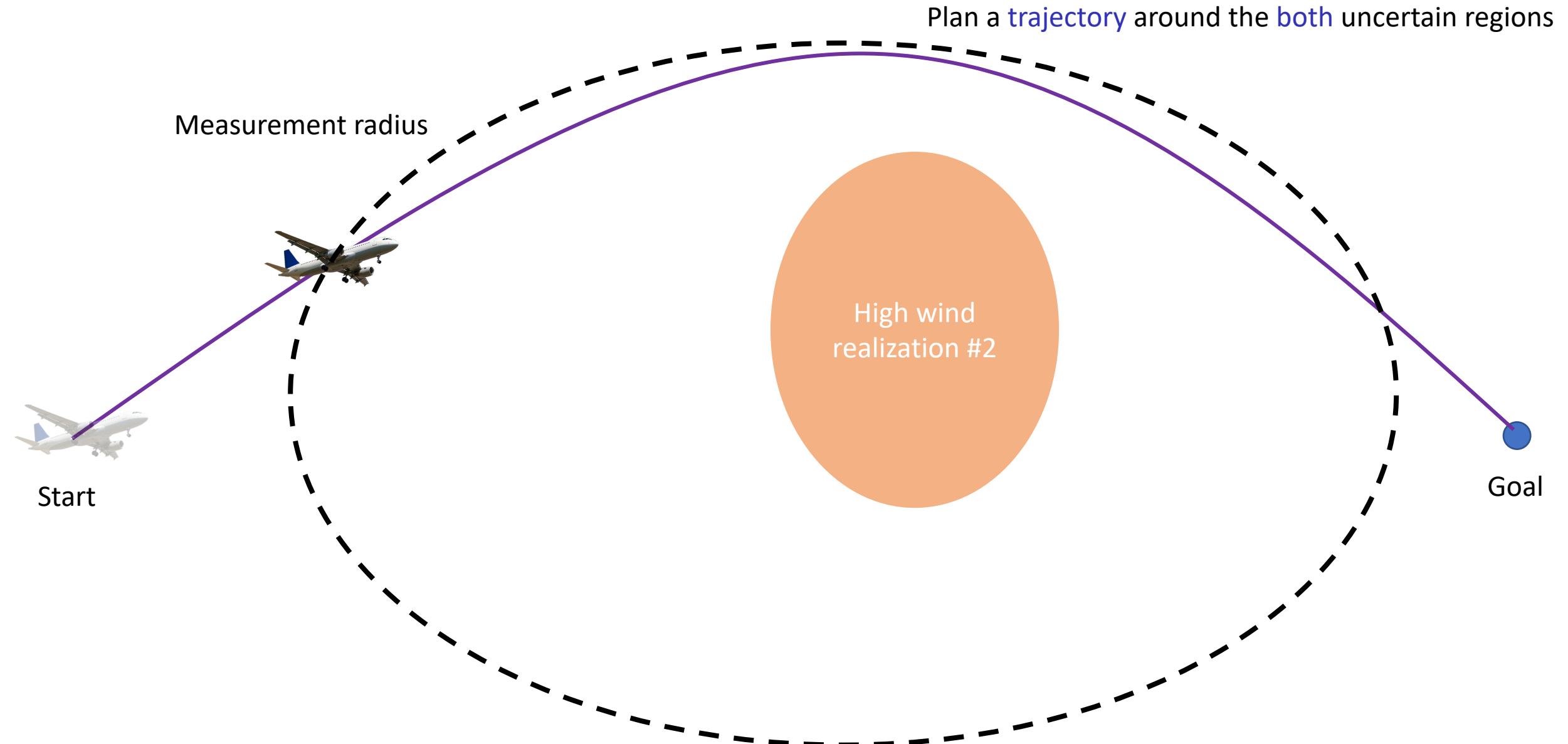
Can we leverage the MPC approach?



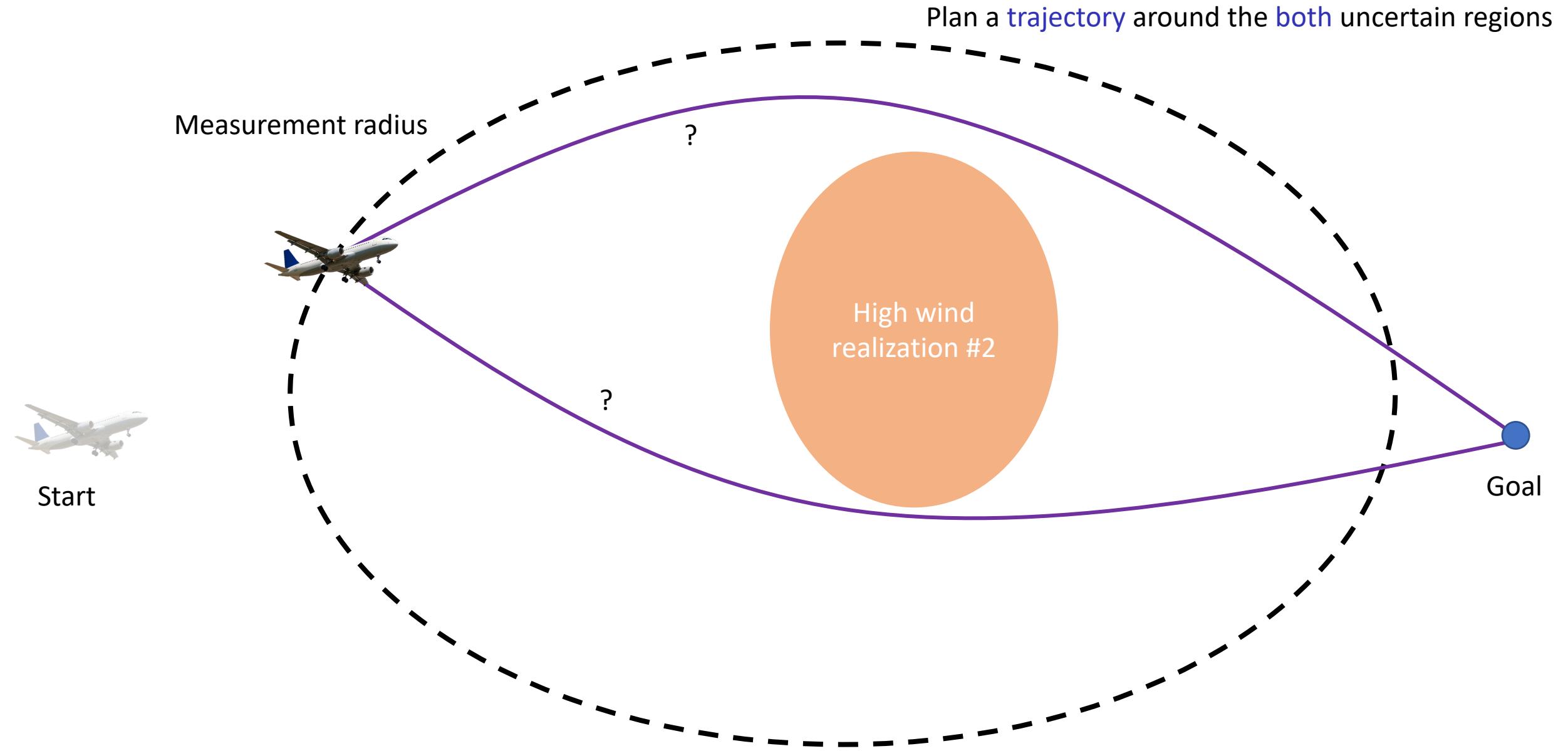
Can we leverage the MPC approach?



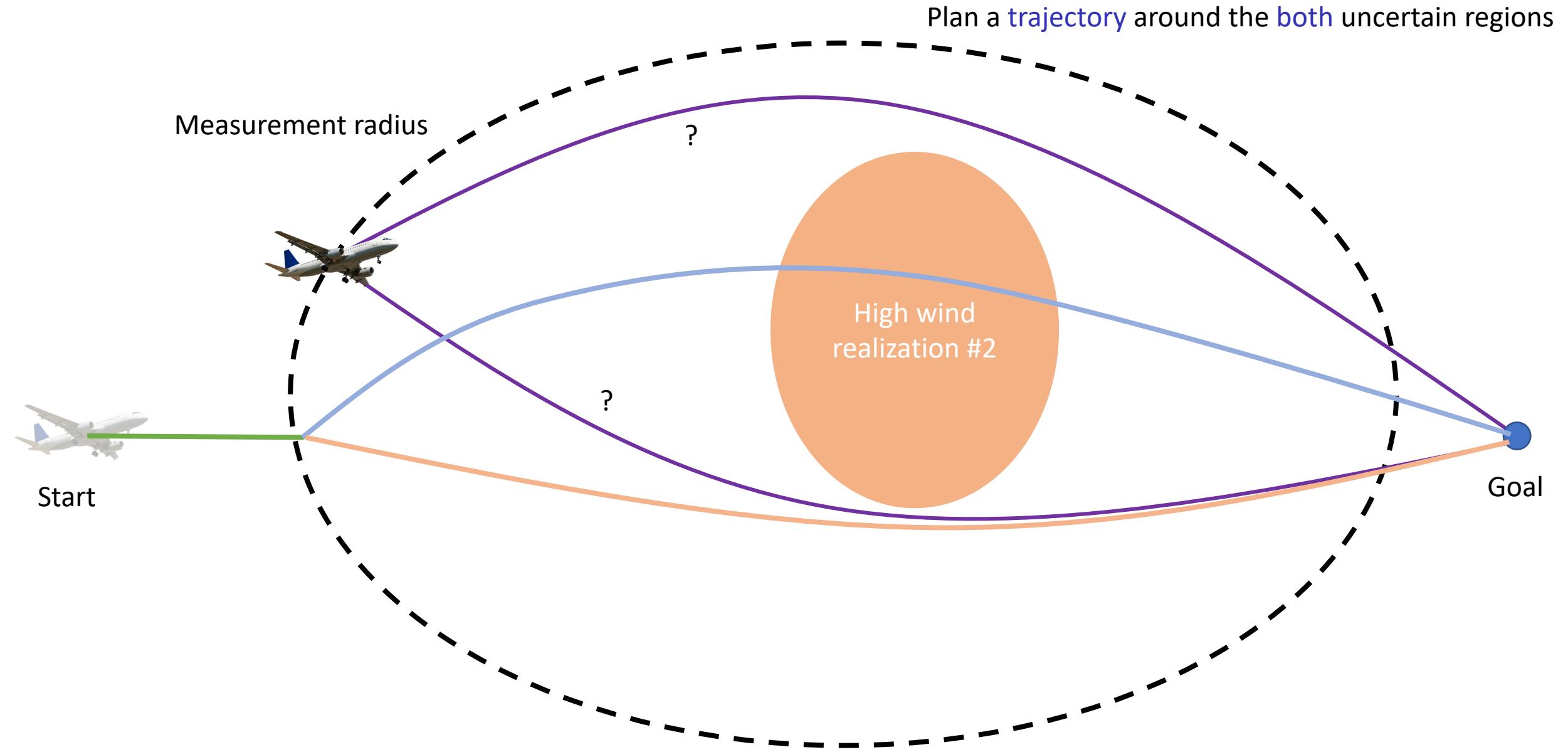
Can we leverage the MPC approach?



Can we leverage the MPC approach?



Can we leverage the MPC approach?



Why should I care?

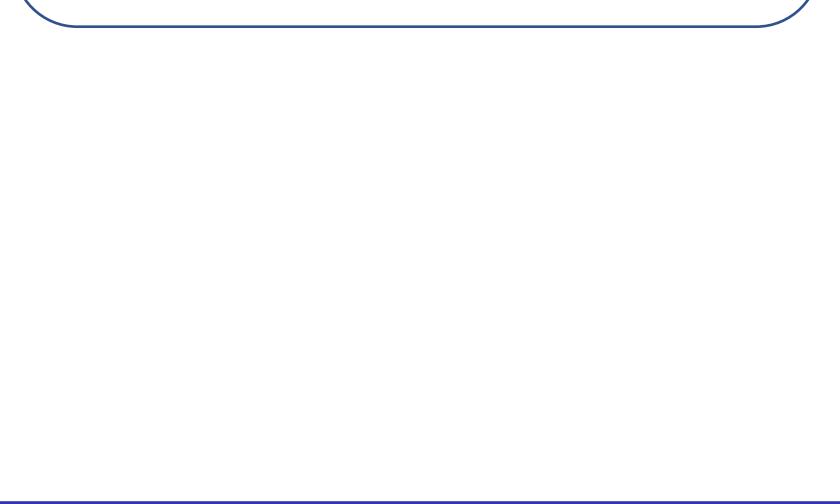
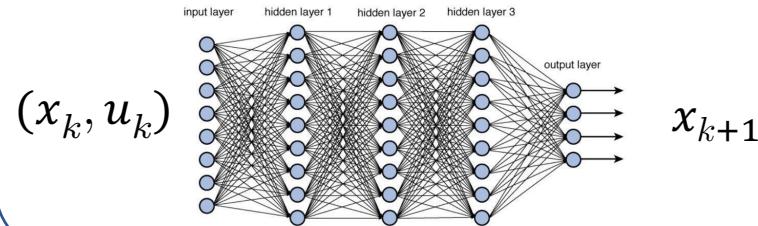
Why should I care?

Model-based RL

Why should I care?

Model-based RL

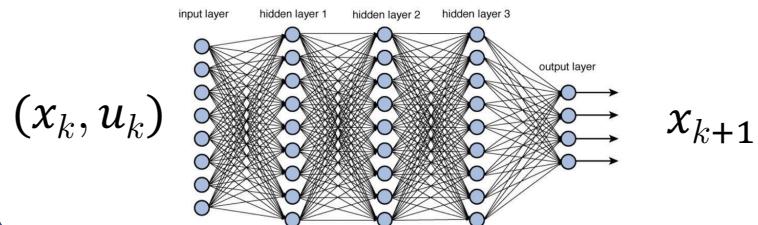
Step 1: Learn a Model



Why should I care?

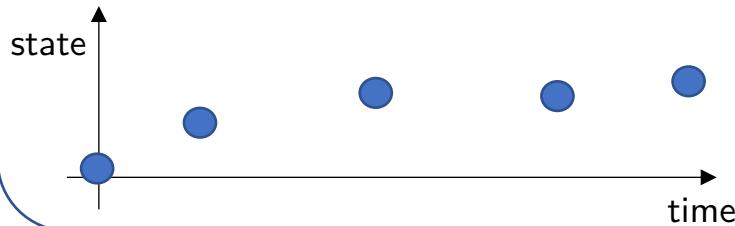
Model-based RL

Step 1: Learn a Model



Step 2: Receding horizon planning

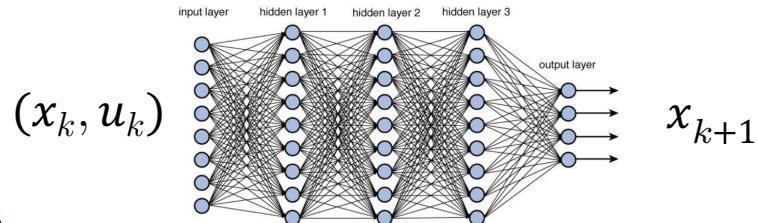
$$\min \mathbb{E}_{\substack{a_t \sim \pi \\ s_t \sim p}} [\sum_{t=0}^H \gamma^t c(a_t, s_t)]$$



Why should I care?

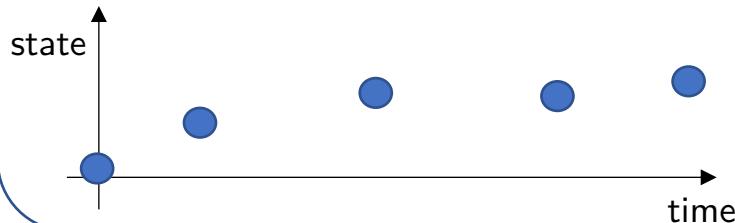
Model-based RL

Step 1: Learn a Model



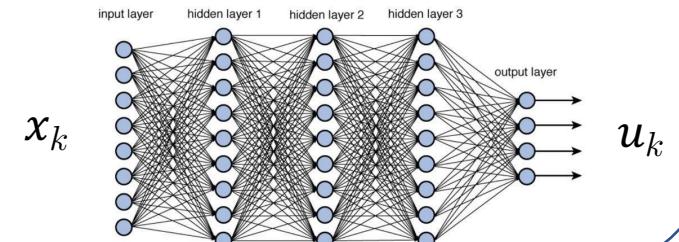
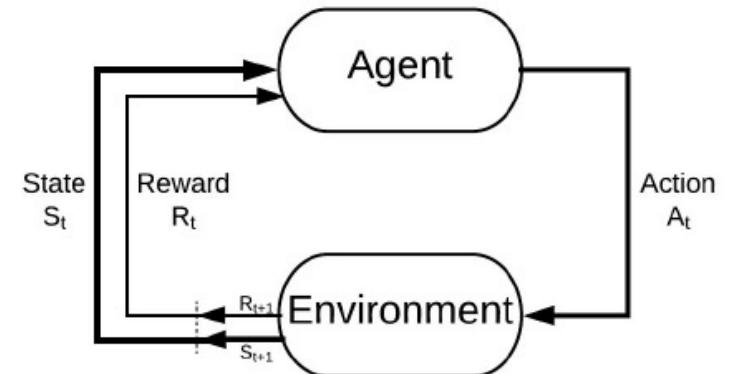
Step 2: Receding horizon planning

$$\min \mathbb{E}_{\substack{a_t \sim \pi \\ s_t \sim p}} [\sum_{t=0}^H \gamma^t c(a_t, s_t)]$$



Model-free RL

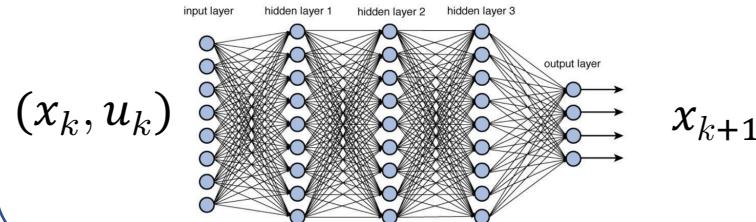
Step 1: Learn a policy



Why should I care?

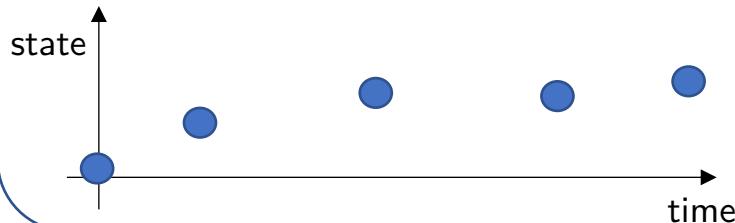
Model-based RL

Step 1: Learn a Model



Step 2: Receding horizon planning

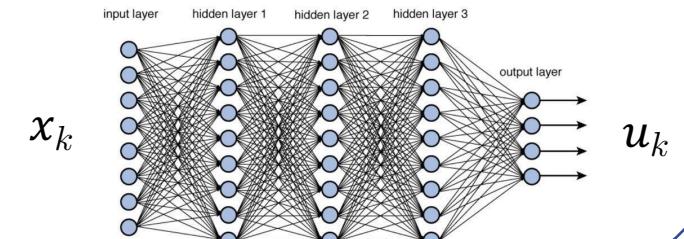
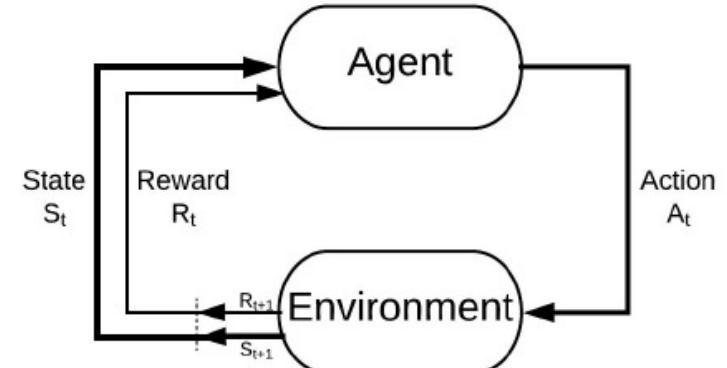
$$\min \mathbb{E}_{\substack{a_t \sim \pi \\ s_t \sim p}} [\sum_{t=0}^H \gamma^t c(a_t, s_t)]$$



Which one has better asymptotic performance?

Model-free RL

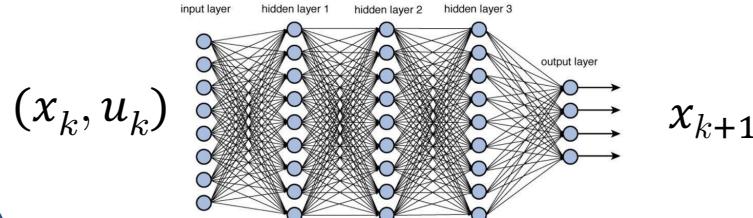
Step 1: Learn a policy



Why should I care?

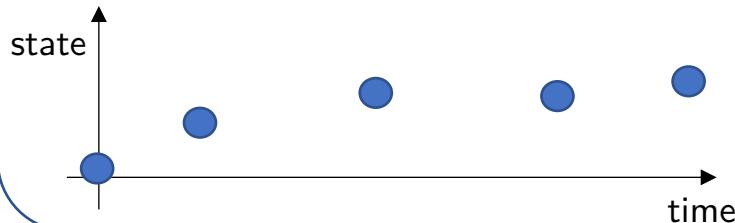
Model-based RL

Step 1: Learn a Model



Step 2: Receding horizon planning

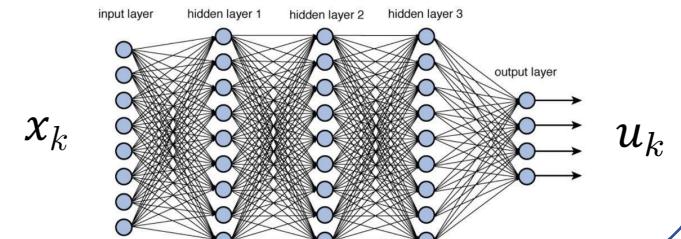
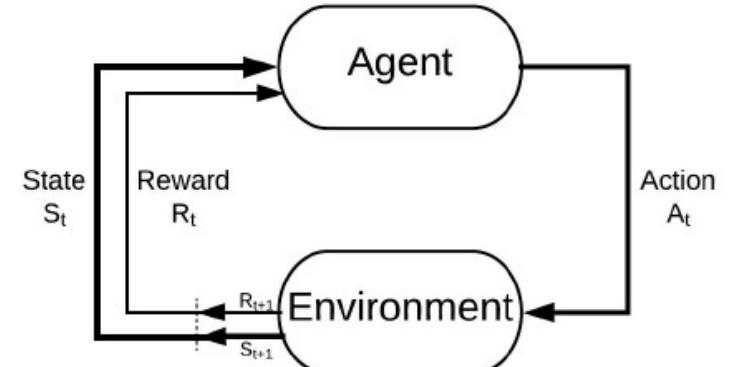
$$\min \mathbb{E}_{\substack{a_t \sim \pi \\ s_t \sim p}} [\sum_{t=0}^H \gamma^t c(a_t, s_t)]$$



Which one has better asymptotic performance?

Model-free RL

Step 1: Learn a policy

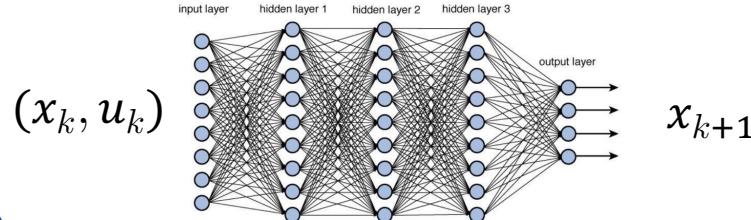


What can go wrong with model-based RL?

Why should I care?

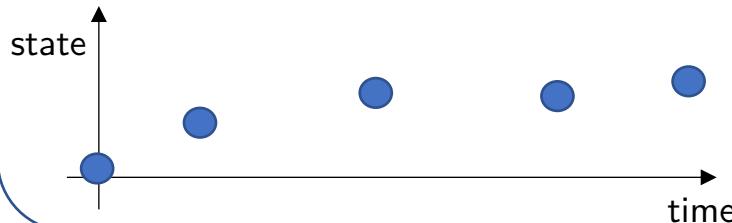
Model-based RL

Step 1: Learn a Model



Step 2: Receding horizon planning

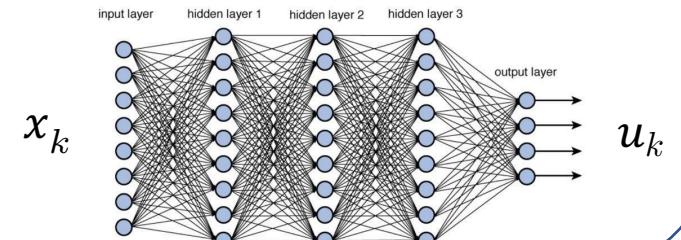
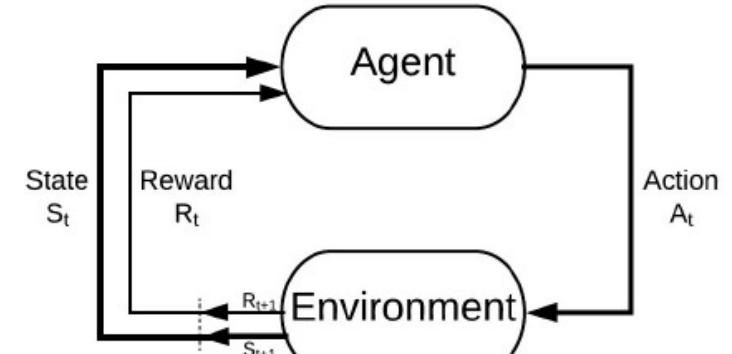
$$\min \mathbb{E}_{\substack{a_t \sim \pi \\ s_t \sim p}} [\sum_{t=0}^H \gamma^t c(a_t, s_t)]$$



Which one has better asymptotic performance?

Model-free RL

Step 1: Learn a policy



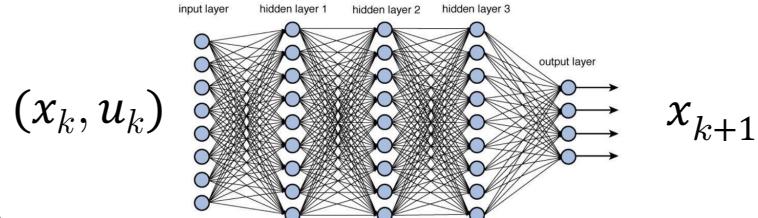
What can go wrong with model-based RL?

- ▶ The prediction horizon is short → need a value function and safe set

Why should I care?

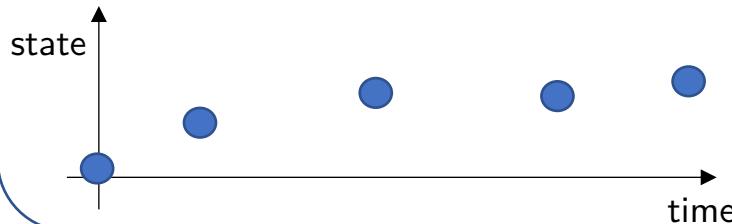
Model-based RL

Step 1: Learn a Model



Step 2: Receding horizon planning

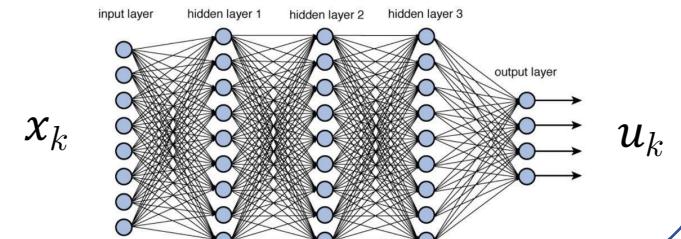
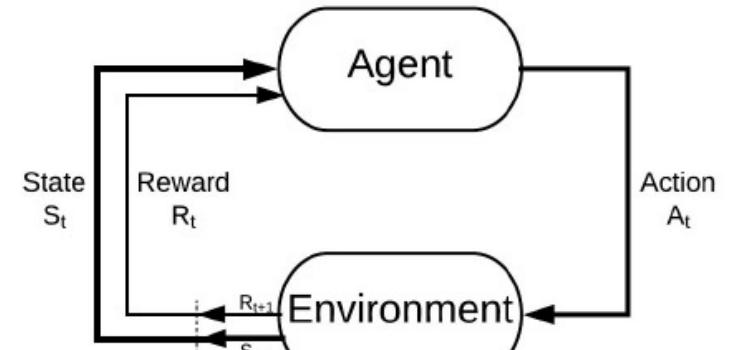
$$\min \mathbb{E}_{\substack{a_t \sim \pi \\ s_t \sim p}} [\sum_{t=0}^H \gamma^t c(a_t, s_t)]$$



Which one has better asymptotic performance?

Model-free RL

Step 1: Learn a policy



What can go wrong with model-based RL?

- ▶ The prediction horizon is short → need a value function and safe set
- ▶ Planning trajectories is suboptimal! We should plan over policies

Why should I care?

Model-based RL

Benchmarking Model-Based Reinforcement Learning

Tingwu Wang^{1,2}, Xuchan Bao^{1,2}, Ignasi Clavera³, Jerrick Hoang^{1,2}, Yeming Wen^{1,2}, Eric Langlois^{1,2}, Shunshi Zhang^{1,2}, Guodong Zhang^{1,2}, Pieter Abbeel³ & Jimmy Ba^{1,2}

¹University of Toronto ² Vector Institute ³ UC Berkeley
 $\{tingwuwang, jhoang, ywen, edl, gdzhang\}@cs.toronto.edu$
 $\{xuchan.bao, matthew.zhang\}@mail.utoronto.ca$, $iclavera@berkeley.edu$,
 $pabbeel@cs.berkeley.edu$, $jba@cs.toronto.edu$

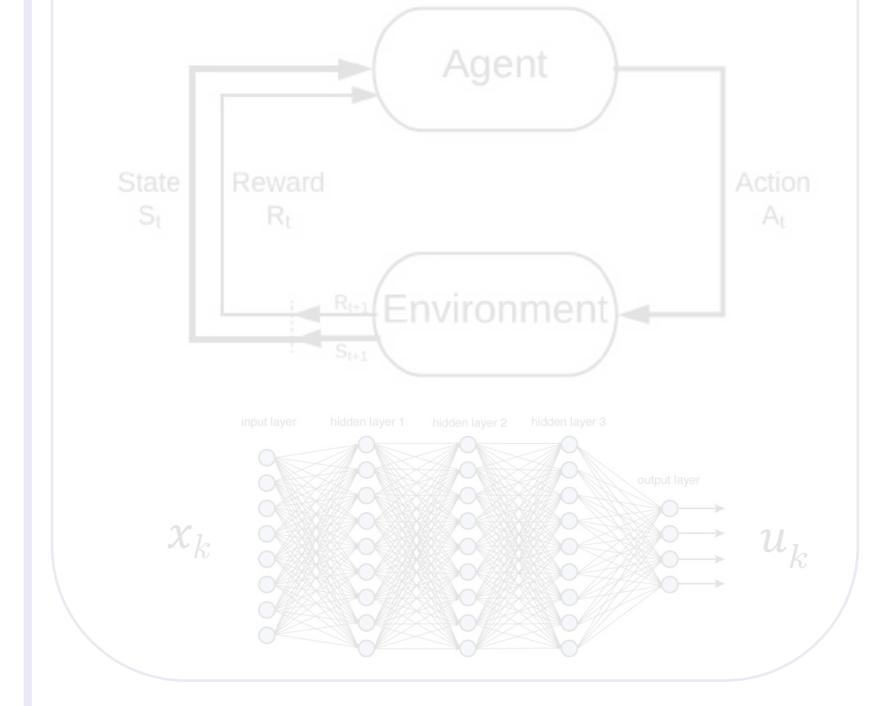
Abstract

Model-based reinforcement learning (MBRL) is widely seen as having the potential to be significantly more sample efficient than model-free RL. However, research in model-based RL has not been very standardized. It is fairly common for authors to experiment with self-designed environments, and there are several separate lines of research, which are sometimes closed-sourced or not reproducible. Accordingly, it is an open question how these various existing MBRL algorithms perform relative to each other. To facilitate research in MBRL, in this paper we gather a wide collection of MBRL algorithms and propose over 18 benchmarking environments specially designed for MBRL. We benchmark these algorithms with unified problem settings, including noisy environments. Beyond cataloguing performance, we explore and unify the underlying algorithmic differences across MBRL algorithms. We characterize three key research challenges for future MBRL research: the dynamics bottleneck, the planning horizon dilemma, and the early-termination dilemma. Finally, to maximally facilitate future research on MBRL, we open-source our benchmark in <http://www.cs.toronto.edu/~tingwuwang/mbrl.html>.

has better performance?

Model-free RL

Step 1: Learn a policy



value function and safe set
▶ Planning trajectories is suboptimal! We should plan over policies

Why should I care?

Model-based RL

Benchmarking Model-Based Reinforcement Learning

Tingwu Wang^{1,2}, Xuchan Bao^{1,2}, Ignasi Clavera³, Jerrick Hoang^{1,2}, Yeming Wen^{1,2}, Eric Langlois^{1,2}, Shunshi Zhang^{1,2}, Guodong Zhang^{1,2}, Pieter Abbeel³ & Jimmy Ba^{1,2}

¹University of Toronto ² Vector Institute ³ UC Berkeley

{tingwuwang, jhoang, ywen, edl, gdzhang}@cs.toronto.edu
{xuchan.bao, matthew.zhang}@mail.utoronto.ca, iclavera@berkeley.edu,
pabbeel@cs.berkeley.edu, jba@cs.toronto.edu

Abstract

Model-based reinforcement learning (MBRL) is widely seen as having the potential to be significantly more sample efficient than model-free RL. However, research in model-based RL has not been very standardized. It is fairly common for authors to experiment with self-designed environments, and there are several separate lines of research, which are sometimes closed-sourced or not reproducible. Accordingly, it is an open question how these various existing MBRL algorithms perform relative to each other. To facilitate research in MBRL, in this paper we gather a wide collection of MBRL algorithms and propose over 18 benchmarking environments specially designed for MBRL. We benchmark these algorithms with unified problem settings, including noisy environments. Beyond cataloguing performance, we explore and unify the underlying algorithmic differences across MBRL algorithms. We characterize three key research challenges for future MBRL research: the dynamics bottleneck, the planning horizon dilemma, and the early-termination dilemma. Finally, to maximally facilitate future research on MBRL, we open-source our benchmark in <http://www.cs.toronto.edu/~tingwuwang/mbrl.html>.

Model-free RL

Model-Based Reinforcement Learning via Meta-Policy Optimization

Ignasi Clavera*

UC Berkeley

iclavera@berkeley.edu

Jonas Rothfuss*

KIT, UC Berkeley

jonas.rothfuss@kit.edu

John Schulman

OpenAI

Yasuhiro Fujita

Preferred Networks

Tamim Asfour

Karlsruhe Inst. of Technology (KIT)

Pieter Abbeel

UC Berkeley, Covariant.AI

Abstract: Model-based reinforcement learning approaches carry the promise of being data efficient. However, due to challenges in learning dynamics models that sufficiently match the real-world dynamics, they struggle to achieve the same asymptotic performance as model-free methods. We propose Model-Based Meta-Policy-Optimization (MB-MPO), an approach that foregoes the strong reliance on accurate learned dynamics models. Using an ensemble of learned dynamic models, MB-MPO meta-learns a policy that can quickly adapt to any model in the ensemble with one policy gradient step. This steers the meta-policy towards internalizing consistent dynamics predictions among the ensemble while shifting the burden of behaving optimally w.r.t. the model discrepancies towards the adaptation step. Our experiments show that MB-MPO is more robust to model imperfections than previous model-based approaches. Finally, we demonstrate that our approach is able to match the asymptotic performance of model-free methods while requiring significantly less experience.

Keywords: Reinforcement Learning, Meta-Learning, Model-Based, Model-Free

► Planning trajectories is suboptimal! We should plan over policies

Why should I care?

Model-based RL

Benchmarking Model-Based Reinforcement Learning

Tingwu Wang^{1,2}, Xuchan Bao^{1,2}, Ignasi Clavera³, Jerrick Hoang^{1,2}, Yeming Wen^{1,2}, Eric Langlois^{1,2}, Shunshi Zhang^{1,2}, Guodong Zhang^{1,2}, Pieter Abbeel³ & Jimmy Ba^{1,2}

¹University of Toronto ² Vector Institute ³ UC Berkeley

{tingwuwang, jhoang, ywen, edl, gdzhang}@cs.toronto.edu
{xuchan.bao, matthew.zhang}@mail.utoronto.ca, iclavera@berkeley.edu,
pabbeel@cs.berkeley.edu, jba@cs.toronto.edu

Abstract

Model-based reinforcement learning (MBRL) is widely seen as having the potential to be significantly more sample efficient than model-free RL. However, research in model-based RL has not been very standardized. It is fairly common for authors to experiment with self-designed environments, and there are several separate lines of research, which are sometimes closed-sourced or not reproducible. Accordingly, it is an open question how these various existing MBRL algorithms perform relative to each other. To facilitate research in MBRL, in this paper we gather a wide collection of MBRL algorithms and propose over 18 benchmarking environments specially designed for MBRL. We benchmark these algorithms with unified problem settings, including noisy environments. Beyond cataloguing performance, we explore and unify the underlying algorithmic differences across MBRL algorithms. We characterize three key research challenges for future MBRL research: the dynamics bottleneck, the planning horizon dilemma, and the early-termination dilemma. Finally, to maximally facilitate future research on MBRL, we open-source our benchmark in <http://www.cs.toronto.edu/~tingwuwang/mbrl.html>.

Model-free RL

Model-Based Reinforcement Learning via Meta-Policy Optimization

Ignasi Clavera*

UC Berkeley

iclavera@berkeley.edu

Jonas Rothfuss*

KIT, UC Berkeley

jonas.rothfuss@kit.edu

John Schulman

OpenAI

Yasuhiro Fujita

Preferred Networks

Tamim Asfour

Karlsruhe Inst. of Technology (KIT)

Pieter Abbeel

UC Berkeley, Covariant.AI

Abstract: Model-based reinforcement learning approaches carry the promise of being data efficient. However, due to challenges in learning dynamics models that sufficiently match the real-world dynamics, they struggle to achieve the same asymptotic performance as model-free methods. We propose Model-Based Meta-Policy-Optimization (MB-MPO), an approach that foregoes the strong reliance on accurate learned dynamics models. Using an ensemble of learned dynamic models, MB-MPO meta-learns a policy that can quickly adapt to any model in the ensemble with one policy gradient step. This steers the meta-policy towards internalizing consistent dynamics predictions among the ensemble while shifting the burden of behaving optimally w.r.t. the model discrepancies towards the adaptation step. Our experiments show that MB-MPO is more robust to model imperfections than previous model-based approaches. Finally, we demonstrate that our approach is able to match the asymptotic performance of model-free methods while requiring significantly less experience.

Keywords: Reinforcement Learning, Meta-Learning, Model-Based, Model-Free

► Planning trajectories is suboptimal! We should plan over policies

How do we fix this problem?

How do we fix this problem?

1136

IEEE TRANSACTIONS ON AUTOMATIC CONTROL, VOL. 43, NO. 8, AUGUST 1998

Statement 2: Under the assumptions of this theorem for any $0 \leq t_0 \leq t$

$$\begin{aligned}\Phi_{\tilde{u}}(t) - \Phi_{\tilde{u}^*}(t_0) &\geq \Phi_{\tilde{u}^*}(t) - \Phi_{\tilde{u}^*}(t_0) \\ &= -\frac{1}{h} \Delta \vartheta^T(t) L^{-1} \Delta \vartheta(t) + o_\omega(h)\end{aligned}$$

where

$$\Phi_{\tilde{u}}(t) := \int_0^t (2\tilde{u}^T(\tau)d\vartheta(\tau) + \tilde{u}^T(\tau)L\tilde{u}(\tau)d\tau) \quad (20)$$

$$\begin{aligned}\Delta \vartheta(t) &:= B_{ab}^T P[B_{ab}(u_{comp}(t) - x^*(t))h \\ &+ (K + A_0 C_0^+) \Delta y(t)], \quad \Delta y(t) = y(t) - y(t-h)\end{aligned}$$

and this minimum is reachable for

$$\begin{aligned}\dot{\tilde{u}}(t) dt &= \tilde{u}^*(t) dt := -L^{-1} B_{ab}^T P[B_{ab}(u_{comp}(t) + x^*(t))dt \\ &+ (K + A_0 C_0^+) dy(t)].\end{aligned}$$

As a result, we have: $d\Phi_{\tilde{u}^*}(t) \leq 0$ (in symbolic form).

Proof of Statement 2: Using the Euler–Maruyama's formula [3], [9] we obtain the following relation: $(h := t - t_0 \rightarrow 0), \Phi_{\tilde{u}}(t) - \Phi_{\tilde{u}^*}(t_0) = 2\tilde{u}^T(t)\Delta \vartheta(t) + \tilde{u}^T(t)L\tilde{u}(t)h + o_\omega(h)$.

Minimizing then the right side for each fixed t , we derive

$$\begin{aligned}\min_{\tilde{u}(t)} [\Phi_{\tilde{u}}(t) - \Phi_{\tilde{u}^*}(t_0)] &= \Phi_{\tilde{u}^*}(t) - \Phi_{\tilde{u}^*}(t_0) \\ &= -\frac{1}{h} \Delta \vartheta^T(t) L^{-1} \Delta \vartheta(t) + o_\omega(h) \leq o_\omega(h).\end{aligned}\quad (21)$$

Hence, taking into account the definition (20), we have $d\Phi_{\tilde{u}^*}(t) \leq 0$. \square

Rewriting (18) in differential form and taking into account that P is the solution of the Riccati equation (see Assumption 4 of this theorem) we finally derive

$$dV(e(t)) \stackrel{a.s.}{\leq} [\varphi(t) + I(t)] dt + S^T(t) dw(t) - e^T(t)Q_e e(t) dt.$$

According to the Kronecker lemma, the Large Number law for martingale, and Skorohod lemma (see [2] and [4]), we obtain the result of the theorem. Details of this proof can be found in [13].

REFERENCES

- [1] S. Boyd, L. Ghaoui, E. Feron, and V. Balakrishnan, *Linear Matrix Inequalities in System and Control Theory*. Philadelphia, PA: SIAM, 1994.
- [2] H. Cramer and M. Lidbetter, *Stationary Random Processes*. Princeton, 1969.
- [3] T. Gard, *Introduction to Stochastic Differential Equations*. New York: Marcel Dekker, 1988.
- [4] I. I. Gilman and A. V. Skorohod, *Stochastic Differential Equations*. New York: Springer-Verlag, 1972.
- [5] M. Grimble, "Three and half DOF polynomial solution of the feed-forward H_2/H_∞ control problem," in Proc. 34rd Conf. Decision and Control, New Orleans, LA, Dec. 1995, pp. 4145–4150.

- [12] A. Poznyak and M. Taksar, "Robust state feedback stochastic control for linear systems with time-varying parameters," in Proc. 33rd Conf. Decision and Control, Lake Buena Vista, FL, Dec. 1994, pp. 1161–1162.
- [13] ———, "Robust control of linear stochastic systems with fully observable state," *Applications Mathematicae*, vol. 24, no. 1, pp. 35–46, 1996.
- [14] M. I. Taksar, A. S. Poznyak, and A. Ipparaguirre, "Robust feedback control for linear stochastic systems in continuous time with time-varying parameters," in Proc. 34rd Conf. Decision and Control, New Orleans, LA, Dec. 1995, pp. 2179–2184.
- [15] J. C. Willems, "Least squares stationary optimal control and the Riccati equation," *IEEE Trans. Automat. Contr.*, vol. AC-16, no. 6, pp. 621–634, 1971.



ELSEVIER

Available online at www.sciencedirect.com

SCIENCE @ DIRECT®

Automatica 42 (2006) 523–533

automatica

www.elsevier.com/locate/automatica



ELSEVIER

Available online at www.sciencedirect.com

SCIENCE @ DIRECT®

Automatica 41 (2005) 219–224

Brief paper

automatica

www.elsevier.com/locate/automatica

Optimization over state feedback policies for robust control with constraints

Paul J. Goulart^{a,*}, Eric C. Kerrigan^b, Jan M. Maciejowski^a^aDepartment of Engineering, University of Cambridge, Trumpington Street, Cambridge CB2 1FW, UK^bDepartment of Aeronautics and Department of Electrical and Electronic Engineering, Imperial College London, Exhibition Road, London SW7 2BY, UK

Received 15 March 2005; received in revised form 8 June 2005; accepted 25 August 2005

Abstract

This paper is concerned with the optimal control of linear discrete-time systems subject to unknown but bounded state dist mixed polytopic constraints on the state and input. It is shown that the class of admissible affine state feedback control policies with prior states is equivalent to the class of admissible feedback policies that are affine functions of the past disturbance sequence. that a broad class of constrained finite horizon robust and optimal control problems, where the optimization is over affine state policies, can be solved in a computationally efficient fashion using convex optimization methods. This equivalence result is used robust receding horizon control (RHC) state feedback policy such that the closed-loop system is input-to-state stable (ISS) and the are satisfied for all time and all allowable disturbance sequences. The cost to be minimized in the associated finite horizon op problem is quadratic in the disturbance-free state and input sequences. The value of the receding horizon control law can be calc sample instant using a single, tractable and convex quadratic program (QP) if the disturbance set is polytopic, or a tractable secon program (SOCP) if the disturbance set is given by a 2-norm bound. © 2006 Elsevier Ltd. All rights reserved.

Keywords: Robust control; Constraint satisfaction; Robust optimization; Predictive control; Optimal control

1. Introduction

This paper is concerned with the control of constrained discrete-time linear systems that are subject to additive, but bounded disturbances on the state. The main aim is to provide results that allow for the efficient computation of an optimal and stabilizing state feedback control policy that ensures a given set of state and input constraints are satisfied for all time, despite the presence of the disturbances. This is a problem that has been studied for some time now in the optimal control literature (Bertsekas & Rhodes, 1973) and a number of different

& Diaz-Bobillo, 1995) or predictive control (Mayne et al., 2002; Mayne, Rawlings, Rao, & Scokaert, 2000).

It is generally accepted that if disturbances accounted for in the formulation of a constraint control problem, then the optimization has to be admissible state feedback policies, rather than open sequences, otherwise infeasibility and instability may occur (Mayne et al., 2000). However, optimization (nonlinear) feedback policies is particularly constraints have to be satisfied. Current proposals for this using finite dimensional optimization, such

Abstract

This paper provides a novel solution to the problem of robust model predictive control of constrained, linear, discrete-time systems in the presence of bounded disturbances. The optimal control problem that is solved online includes, uniquely, the initial state of the model employed in the problem as a decision variable. The associated value function is zero in a disturbance invariant set that serves as the 'origin' when bounded disturbances are present, and permits a strong stability result, namely robust exponential stability of the disturbance invariant set for the controlled system with bounded disturbances, to be obtained. The resultant online algorithm is a quadratic program of similar complexity to that required in conventional model predictive control. © 2004 Elsevier Ltd. All rights reserved.

Keywords: Robust model predictive control; Robustness; Bounded disturbances

1. Introduction

Model predictive control is widely employed for the control of constrained systems and an extensive literature on the subject exists some of which is reviewed in Bemporad and Morari (1999); Allgöwer, Badgwell, Qin, Rawlings, and Wright (1999); Mayne, Rawlings, Rao, and Scokaert (2000). Several methods for achieving robustness have been considered. The simplest is to ignore the disturbance and rely on the inherent robustness of deterministic model predictive control applied to the nominal system (Scokaert & Rawlings, 1995; Marruedo, Álamo, & Camacho, 2002). *Open-loop* model predictive control that determines the

$\{u_0, u_1, \dots, u_{N-1}\}$ of control actions was proposed in Zheng and Morari (1993); this method cannot contain the 'spread' of predicted trajectories resulting from disturbances. Hence *feedback* model predictive control in which the decision variable is a *policy* π , which is a sequence $\{\mu_0(\cdot), \mu_1(\cdot), \dots, \mu_{N-1}(\cdot)\}$ of control laws, was advocated in, for example, Mayne (1995); Kothare, Balakrishnan, and Morari (1996); Lee and Yu (1997); Scokaert and Mayne (1998); De Nicalo, Magni, and Scattolini (2000); Magni, Nijmeijer, and van der Schaft (2001); Magni, De Nicolao, Scattolini, and Allgöwer (2003). Determination of a control policy is usually prohibitively difficult so research has concentrated on various simplifications and relaxations (Mayne

How do we fix this problem?

1136

IEEE TRANSACTIONS ON AUTOMATIC CONTROL, VOL. 43, NO. 8, AUGUST 1998

Statement 2: Under the assumptions of this theorem for any $0 \leq t_0 \leq t$

$$\Phi_{\tilde{u}}(t) - \Phi_{\tilde{u}^*}(t_0) \geq \Phi_{\tilde{u}^*}(t) - \Phi_{\tilde{u}^*}(t_0) \\ = -\frac{1}{h} \Delta \vartheta^T(t) L^{-1} \Delta \vartheta(t) + o_\omega(h)$$

where

$$\Phi_{\tilde{u}}(t) := \int_0^t (2\tilde{u}^T(\tau)d\vartheta(\tau) + \tilde{u}^T(\tau)L\tilde{u}(\tau)d\tau) \quad (20)$$

$$\Delta \vartheta(t) := B_{ab}^T P[B_{ab}(u_{comp}(t) - x^*(t))h \\ + (K + A_0 C_0^+) \Delta y(t)], \quad \Delta y(t) = y(t) - y(t-h)$$

and this minimum is reachable for

$$\dot{\tilde{u}}(t) dt = \tilde{u}^*(t) dt := -L^{-1} B_{ab}^T P[B_{ab}(u_{comp}(t) + x^*(t)) dt \\ + (K + A_0 C_0^+) dy(t)].$$

As a result, we have: $d\Phi_{\tilde{u}^*}(t) \leq 0$ (in symbolic form).

Proof of Statement 2: Using the Euler–Maruyama's formula [3], [9] we obtain the following relation: $(h := t - t_0 \rightarrow 0), \Phi_{\tilde{u}}(t) - \Phi_{\tilde{u}^*}(t_0) = 2\tilde{u}^T(t)\Delta \vartheta(t) + \tilde{u}^T(t)L\tilde{u}(t)h + o_\omega(h)$.

Minimizing then the right side for each fixed t , we derive

$$\min_{\tilde{u}(t)} [\Phi_{\tilde{u}}(t) - \Phi_{\tilde{u}^*}(t_0)] = \Phi_{\tilde{u}^*}(t) - \Phi_{\tilde{u}^*}(t_0) \\ = -\frac{1}{h} \Delta \vartheta^T(t) L^{-1} \Delta \vartheta(t) + o_\omega(h) \leq o_\omega(h). \quad (21)$$

Hence, taking into account the definition (20), we have $d\Phi_{\tilde{u}^*}(t) \leq 0$. \square

Rewriting (18) in differential form and taking into account that P is the solution of the Riccati equation (see Assumption 4 of this theorem) we finally derive

$$dV(e(t)) \stackrel{a.s.}{\leq} [\varphi(t) + I(t)] dt + S^T(t) dw(t) - e^T(t)Q_e e(t) dt.$$

According to the Kronecker lemma, the Large Number law for martingale, and Skorohod lemma (see [2] and [4]), we obtain the result of the theorem. Details of this proof can be found in [13].

REFERENCES

- [1] S. Boyd, L. Ghaoui, E. Feron, and V. Balakrishnan, *Linear Matrix Inequalities in System and Control Theory*. Philadelphia, PA: SIAM, 1994.
- [2] H. Cramer and M. Lidbetter, *Stationary Random Processes*. Princeton, 1969.
- [3] T. Gard, *Introduction to Stochastic Differential Equations*. New York: Marcel Dekker, 1988.
- [4] I. I. Gilman and A. V. Skorohod, *Stochastic Differential Equations*. New York: Springer-Verlag, 1972.
- [5] M. Grime, "Three and half DOF polynomial solution of the feed-forward H_2/H_∞ control problem," in Proc. 34rd Conf. Decision and Control, New Orleans, LA, Dec. 1995, pp. 4145–4150.

The problem is NP-hard!

- [12] A. Poznyak and M. Taksar, "Robust state feedback stochastic control for linear systems with time-varying parameters," in Proc. 33rd Conf. Decision and Control, Lake Buena Vista, FL, Dec. 1994, pp. 1161–1162.
- [13] ———, "Robust control of linear stochastic systems with fully observable state," *Applications Mathematicae*, vol. 24, no. 1, pp. 35–46, 1996.
- [14] M. I. Taksar, A. S. Poznyak, and A. Ipparaguirre, "Robust feedback control for linear stochastic systems in continuous time-varying parameters," in Proc. 34rd Conf. Decision and Control, New Orleans, LA, Dec. 1995, pp. 2179–2184.
- [15] J. C. Willems, "Least squares stationary optimal control and the Riccati equation," *IEEE Trans. Automat. Contr.*, vol. AC-16, no. 6, pp. 621–634, 1971.



Available online at www.sciencedirect.com

SCIENCE @ DIRECT®

Automatica 42 (2006) 523–533

automatica

www.elsevier.com/locate/automatica



Available online at www.sciencedirect.com

SCIENCE @ DIRECT®

Automatica 41 (2005) 219–224

Brief paper

automatica

www.elsevier.com/locate/automatica

Optimization over state feedback policies for robust control with constraints

Paul J. Goulart^{a,*}, Eric C. Kerrigan^b, Jan M. Maciejowski^a^aDepartment of Engineering, University of Cambridge, Trumpington Street, Cambridge CB2 1FW, UK^bDepartment of Aeronautics and Department of Electrical and Electronic Engineering, Imperial College London, Exhibition Road, London SW7 2BY, UK

Received 15 March 2005; received in revised form 8 June 2005; accepted 25 August 2005

Abstract

This paper is concerned with the optimal control of linear discrete-time systems subject to unknown but bounded state dist mixed polytopic constraints on the state and input. It is shown that the class of admissible affine state feedback control policies with prior states is equivalent to the class of admissible feedback policies that are affine functions of the past disturbance sequence. that a broad class of constrained finite horizon robust and optimal control problems, where the optimization is over affine state policies, can be solved in a computationally efficient fashion using convex optimization methods. This equivalence result is used robust receding horizon control (RHC) state feedback policy such that the closed-loop system is input-to-state stable (ISS) and the are satisfied for all time and all allowable disturbance sequences. The cost to be minimized in the associated finite horizon op problem is quadratic in the disturbance-free state and input sequences. The value of the receding horizon control law can be calc sample instant using a single, tractable and convex quadratic program (QP) if the disturbance set is polytopic, or a tractable secon program (SOCP) if the disturbance set is given by a 2-norm bound. © 2006 Elsevier Ltd. All rights reserved.

Keywords: Robust control; Constraint satisfaction; Robust optimization; Predictive control; Optimal control

1. Introduction

This paper is concerned with the control of constrained discrete-time linear systems that are subject to additive, but bounded disturbances on the state. The main aim is to provide results that allow for the efficient computation of an optimal and stabilizing state feedback control policy that ensures a given set of state and input constraints are satisfied for all time, despite the presence of the disturbances. This is a problem that has been studied for some time now in the optimal control literature (Bertsekas & Rhodes, 1973) and a number of different

& Diaz-Bobillo, 1995) or predictive control (Mayne et al., 2000; Mayne, Rawlings, Rao, & Scokaert, 2000).

It is generally accepted that if disturbances accounted for in the formulation of a constraint control problem, then the optimization has to be admissible state feedback policies, rather than open sequences, otherwise infeasibility and instability may occur (Mayne et al., 2000). However, optimization trary (nonlinear) feedback policies is particularly constraints have to be satisfied. Current proposals for this using finite dimensional optimization, such as

Abstract

This paper provides a novel solution to the problem of robust model predictive control of constrained, linear, discrete-time systems in the presence of bounded disturbances. The optimal control problem that is solved online includes, uniquely, the initial state of the model employed in the problem as a decision variable. The associated value function is zero in a disturbance invariant set that serves as the 'origin' when bounded disturbances are present, and permits a strong stability result, namely robust exponential stability of the disturbance invariant set for the controlled system with bounded disturbances, to be obtained. The resultant online algorithm is a quadratic program of similar complexity to that required in conventional model predictive control. © 2004 Elsevier Ltd. All rights reserved.

Keywords: Robust model predictive control; Robustness; Bounded disturbances

1. Introduction

Model predictive control is widely employed for the control of constrained systems and an extensive literature on the subject exists some of which is reviewed in Bemporad and Morari (1999); Allgöwer, Badgwell, Qin, Rawlings, and Wright (1999); Mayne, Rawlings, Rao, and Scokaert (2000). Several methods for achieving robustness have been considered. The simplest is to ignore the disturbance and rely on the inherent robustness of deterministic model predictive control applied to the nominal system (Scokaert & Rawlings, 1995; Marruedo, Álamo, & Camacho, 2002). *Open-loop* model predictive control that determines the

$\{u_0, u_1, \dots, u_{N-1}\}$ of control actions was proposed in Zheng and Morari (1993); this method cannot contain the 'spread' of predicted trajectories resulting from disturbances. Hence *feedback* model predictive control in which the decision variable is a *policy* π , which is a sequence $\{\mu_0(\cdot), \mu_1(\cdot), \dots, \mu_{N-1}(\cdot)\}$ of control laws, was advocated in, for example, Mayne (1995); Kothare, Balakrishnan, and Morari (1996); Lee and Yu (1997); Scokaert and Mayne (1998); De Nicalo, Magni, and Scattolini (2000); Magni, Nijmeijer, and van der Schaft (2001); Magni, De Nicolao, Scattolini, and Allgöwer (2003). Determination of a control policy is usually prohibitively difficult so research has concentrated on various simplifying assumptions (Mayne,

An elegant approximation

Optimizing over policies

True system dynamics: $x_{k+1} = Ax_k + Bu_k + w_k$

Optimizing over policies

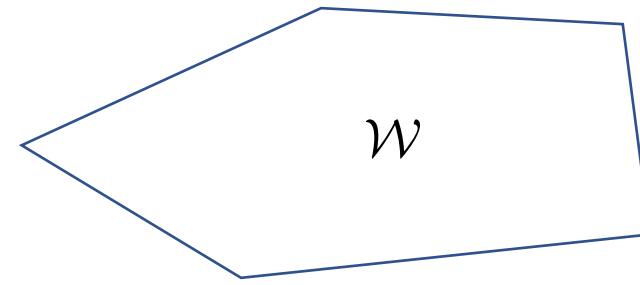
True system dynamics: $x_{k+1} = Ax_k + Bu_k + w_k$

Assumption: the disturbance support $\mathcal{W} = \text{conv}(w^{(0)}, \dots, w^{(n_d)})$.

Optimizing over policies

True system dynamics: $x_{k+1} = Ax_k + Bu_k + w_k$

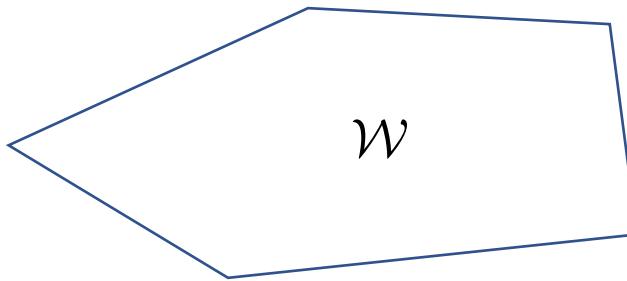
Assumption: the disturbance support $\mathcal{W} = \text{conv}(w^{(0)}, \dots, w^{(n_d)})$.



Optimizing over policies

True system dynamics: $x_{k+1} = Ax_k + Bu_k + w_k$

Assumption: the disturbance support $\mathcal{W} = \text{conv}(w^{(0)}, \dots, w^{(n_d)})$.



The problem that we want to solve

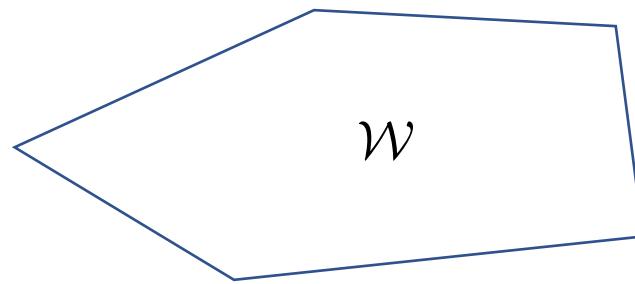
$$\begin{array}{ll}\min_{\pi} & f(x(t), \pi, w_{t|t}, \dots, w_{t+N|t}) \\ \text{subject to} & x_{k+1|t} = Ax_{k|t} + Bu_{k|t} + w_{k|t} \\ & \text{robust constraint satisfaction} \\ & x_{t|t} = x(t) \\ & u_{k|t} = \pi(x_{k|t}),\end{array}$$

Optimization over policies

Optimizing over policies

True system dynamics: $x_{k+1} = Ax_k + Bu_k + w_k$

Assumption: the disturbance support $\mathcal{W} = \text{conv}(w^{(0)}, \dots, w^{(n_d)})$.



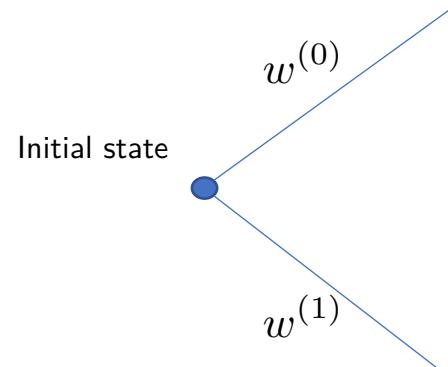
The problem that we want to solve

$$\begin{array}{ll}\min_{\pi} & f(x(t), \pi, w_{t|t}, \dots, w_{t+N|t}) \\ \text{subject to} & x_{k+1|t} = Ax_{k|t} + Bu_{k|t} + w_{k|t} \\ & \text{robust constraint satisfaction} \\ & x_{t|t} = x(t) \\ & u_{k|t} = \pi(x_{k|t}),\end{array}$$

Optimization over policies

Finite dimensional reformulation

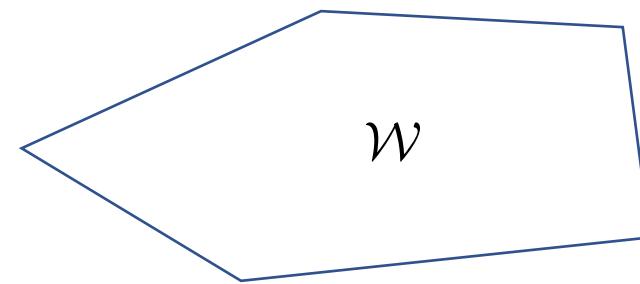
Optimize over a tree



Optimizing over policies

True system dynamics: $x_{k+1} = Ax_k + Bu_k + w_k$

Assumption: the disturbance support $\mathcal{W} = \text{conv}(w^{(0)}, \dots, w^{(n_d)})$.



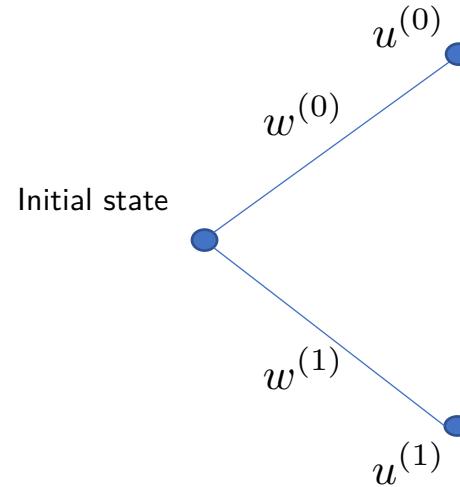
The problem that we want to solve

$$\begin{array}{ll}\min_{\pi} & f(x(t), \pi, w_{t|t}, \dots, w_{t+N|t}) \\ \text{subject to} & x_{k+1|t} = Ax_{k|t} + Bu_{k|t} + w_{k|t} \\ & \text{robust constraint satisfaction} \\ & x_{t|t} = x(t) \\ & u_{k|t} = \pi(x_{k|t}),\end{array}$$

Optimization over policies

Finite dimensional reformulation

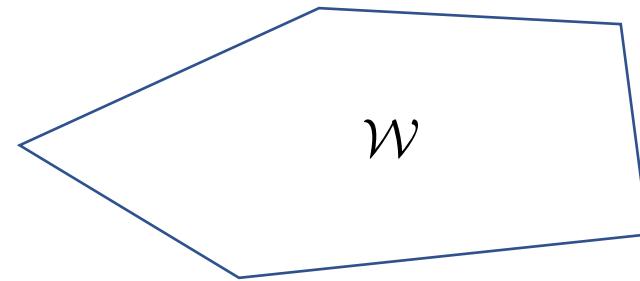
Optimize over a tree



Optimizing over policies

True system dynamics: $x_{k+1} = Ax_k + Bu_k + w_k$

Assumption: the disturbance support $\mathcal{W} = \text{conv}(w^{(0)}, \dots, w^{(n_d)})$.



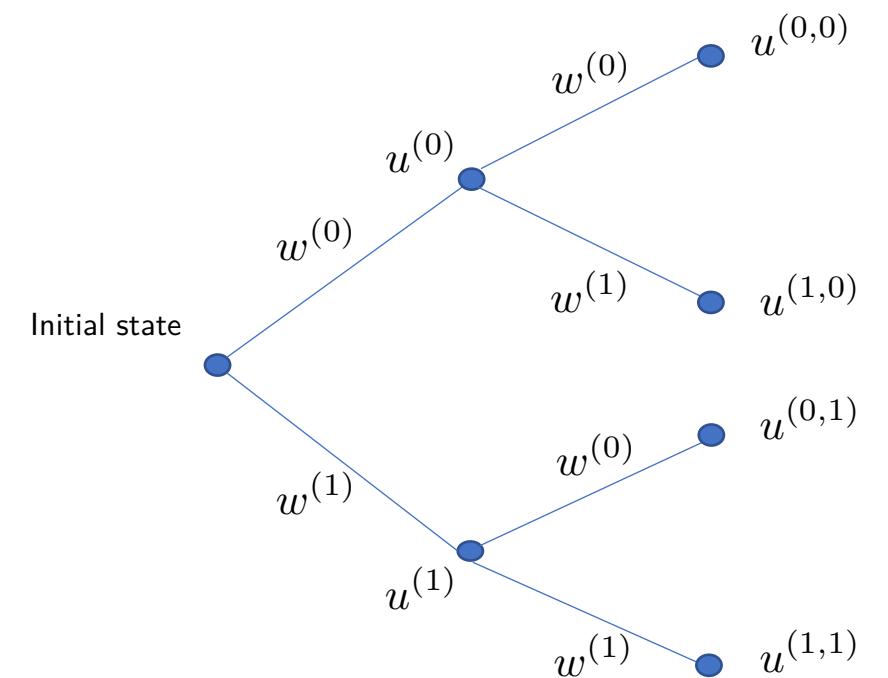
The problem that we want to solve

$$\begin{array}{ll}\min_{\pi} & f(x(t), \pi, w_{t|t}, \dots, w_{t+N|t}) \\ \text{subject to} & x_{k+1|t} = Ax_{k|t} + Bu_{k|t} + w_{k|t} \\ & \text{robust constraint satisfaction} \\ & x_{t|t} = x(t) \\ & u_{k|t} = \pi(x_{k|t}),\end{array}$$

Optimization over policies

Finite dimensional reformulation

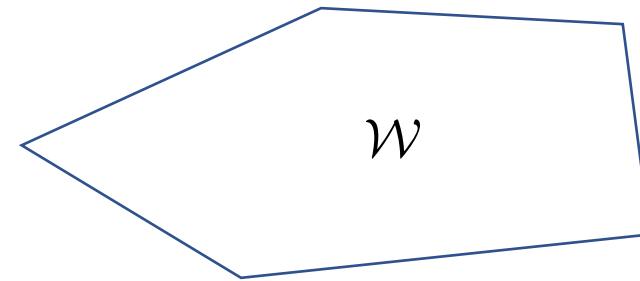
Optimize over a tree



Optimizing over policies

True system dynamics: $x_{k+1} = Ax_k + Bu_k + w_k$

Assumption: the disturbance support $\mathcal{W} = \text{conv}(w^{(0)}, \dots, w^{(n_d)})$.



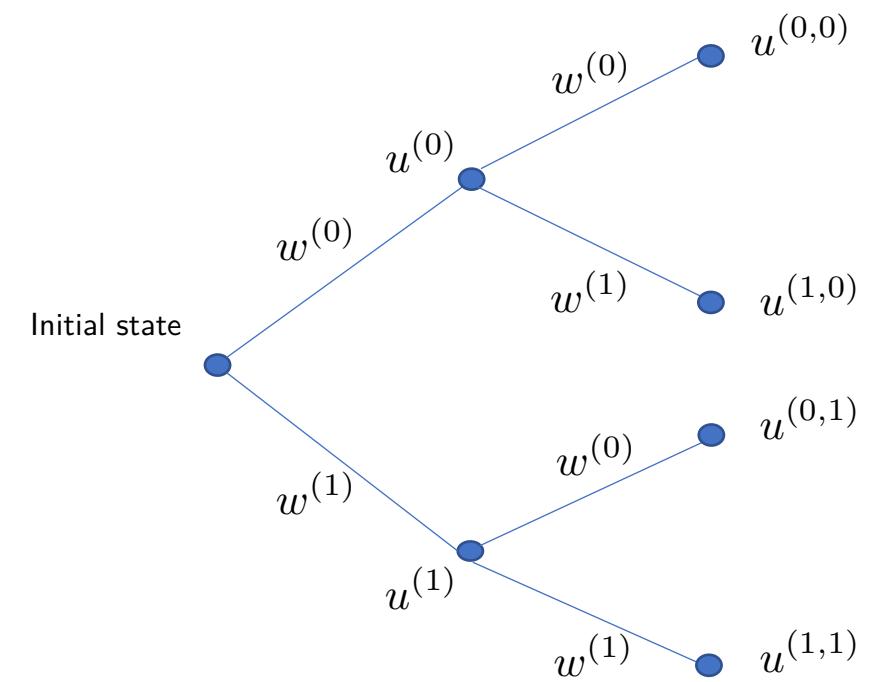
The problem that we want to solve

$$\begin{array}{ll}\min_{\pi} & f(x(t), \pi, w_{t|t}, \dots, w_{t+N|t}) \\ \text{subject to} & x_{k+1|t} = Ax_{k|t} + Bu_{k|t} + w_{k|t} \\ & \text{robust constraint satisfaction} \\ & x_{t|t} = x(t) \\ & u_{k|t} = \pi(x_{k|t}),\end{array}$$

Optimization over policies

Finite dimensional reformulation

Optimize over a tree



This approach can be used also when A and B are uncertain. For further details check:

Francesco Borrelli, Alberto Bemporad, and Manfred Morari. *Predictive control for linear and hybrid systems*. Cambridge University Press, 2017.

How do we fix this problem?

1136

IEEE TRANSACTIONS ON AUTOMATIC CONTROL, VOL. 43, NO. 8, AUGUST 1998

Statement 2: Under the assumptions of this theorem for any $0 \leq t_0 \leq t$

$$\begin{aligned}\Phi_{\omega}^*(t) - \Phi_{\omega}^*(t_0) &\geq \Phi_{\omega^*}(t) - \Phi_{\omega^*}(t_0) \\ &= -\frac{1}{h} \Delta \vartheta^T(t) L^{-1} \Delta \vartheta(t) + o_{\omega}(h)\end{aligned}$$

where

$$\Phi_{\omega}^*(t) := \int_0^t (2\bar{u}^T(\tau) d\vartheta(\tau) + \bar{u}^T(\tau) L \bar{u}(\tau) d\tau) \quad (20)$$

$$\begin{aligned}\Delta \vartheta(t) &:= B_{\omega}^T P[B_{\omega}(a_{\text{comp}}(t) - x^*(t)) h \\ &+ (K + A_3 C_b^+) \Delta y(t)], \quad \Delta y(t) = y(t) - y(t-h)\end{aligned}$$

and this minimum is reachable for

$$\begin{aligned}\bar{u}(t) dt &= \bar{u}^*(t) dt := -L^{-1} B_{\omega}^T P[B_{\omega}(a_{\text{comp}}(t) + x^*(t)) dt \\ &+ (K + A_3 C_b^+) dy(t)].\end{aligned}$$

As a result, we have: $d\Phi_{\omega^*}(t) \leq 0$ (in symbolic form).

Proof of Statement 2: Using the Euler-Maruyama's formula [3], [9] we obtain the following relation: $(h := t - t_0 \rightarrow 0), \Phi_{\omega}^*(t) - \Phi_{\omega}^*(t_0) = 2\bar{u}^T(t) \Delta \vartheta(t) + \bar{u}^T(t) L \bar{u}(t) h + o_{\omega}(h)$.

Minimizing then the right side for each fixed t , we derive

$$\begin{aligned}\min_{\omega(t)} [\Phi_{\omega}^*(t) - \Phi_{\omega}^*(t_0)] &= \Phi_{\omega^*}(t) - \Phi_{\omega^*}(t_0) \\ &= -\frac{1}{h} \Delta \vartheta^T(t) L^{-1} \Delta \vartheta(t) + o_{\omega}(h) \leq o_{\omega}(h).\end{aligned}\quad (21)$$

Hence, taking into account the definition (20), we have $d\Phi_{\omega^*}(t) \leq 0$. \square

Rewriting (18) in differential form and taking into account that P is the solution of the Riccati equation (see Assumption 4 of this theorem) we finally derive

$$dV(e(t)) \leq \|\varphi(t) + I(t)\| dt + S^T(t) dw(t) - e^T(t) Q e(t) dt.$$

According to the Kronecker lemma, the Large Number law for martingale, and Skorohod lemma (see [2] and [4]), we obtain the result of the theorem. Details of this proof can be found in [13].

REFERENCES

- [1] S. Boyd, L. Ghaoui, E. Feron, and V. Balakrishnan, *Linear Matrix Inequalities in System and Control Theory*. Philadelphia, PA: SIAM, 1994.
- [2] H. Cramer and M. Leadbetter, *Stationary Random Processes*. Princeton, 1969.
- [3] T. Gard, *Introduction to Stochastic Differential Equations*. New York: Marcel Dekker, 1988.
- [4] I. I. Gilman and A. V. Skorohod, *Stochastic Differential Equations*. New York: Springer-Verlag, 1972.
- [5] M. Grimble, "Three and half DOF polynomial solution of the feed-forward H_2/H_∞ control problem," in Proc. 34rd Conf. Decision and Control, New Orleans, LA, Dec. 1995, pp. 4145-4150.

The problem is NP-hard!

- [12] A. Poznyak and M. Taksar, "Robust state feedback stochastic control for linear systems with time-varying parameters," in Proc. 33rd Conf. Decision and Control, Lake Buena Vista, FL, Dec. 1994, pp. 1161-1162.
- [13] ———, "Robust control of linear stochastic systems with fully observable state," *Applications Mathematicae*, vol. 24, no. 1, pp. 35-46, 1996.
- [14] M. Taksar, A. S. Poznyak, and A. Ipparragirre, "Robust state feedback control for linear stochastic systems in continuous time with time-varying parameters," in Proc. 34rd Conf. Decision and Control, New Orleans, LA, Dec. 1995, pp. 2179-2184.
- [15] J. C. Willems, "Least squares stationary optimal control and a Riccati equation," *IEEE Trans. Automat. Contr.*, vol. AC-16, no. 6, pp. 621-634, 1971.



Available online at www.sciencedirect.com

SCIENCE @ DIRECT®

Automatica 42 (2006) 523-533

automatica

www.elsevier.com/locate/automatica

Min-Max Feedback Model Predictive Control for Constrained Linear Systems

P. O. M. Scokaert and D. Q. Mayne

Paul J. Goulart^{a,*}, Eric C. Kerrigan^b, Jan M. Maciejowski^a

^aDepartment of Engineering, University of Cambridge, Trumpington Street, Cambridge CB2 1FW, UK

^bDepartment of Aeronautics and Department of Electrical and Electronic Engineering, Imperial College London, Exhibition Road, London SW7 2BY, UK

Received 15 March 2005; received in revised form 8 June 2005; accepted 25 August 2005

Abstract

This paper is concerned with the optimal control of linear discrete-time systems subject to unknown but bounded state dist mixed polytopic constraints on the state and input. It is shown that the class of admissible affine state feedback control policies with prior states is equivalent to the class of admissible feedback policies that are affine functions of the past disturbance sequence, that a broad class of constrained finite horizon robust and optimal control problems, where the optimization is over affine policies, can be solved in a computationally efficient fashion using convex optimization methods. This equivalence result is used robust receding horizon control (RHC) state feedback policy such that the closed-loop system is input-to-state stable (ISS) and it are satisfied for all time and all allowable disturbance sequences. The cost to be minimized in the associated finite horizon op problem is quadratic in the disturbance-free state and input sequences. The value of the receding horizon control law can be calculated using a single, tractable and convex quadratic program (QP) if the disturbance set is polytopic, or a tractable secon program (SOCP) if the disturbance set is given by a 2-norm bound. © 2006 Elsevier Ltd. All rights reserved.

Keywords: Robust control; Constraint satisfaction; Robust optimization; Predictive control; Optimal control

1. Introduction

This paper is concerned with the control of constrained discrete-time linear systems that are subject to additive, but bounded disturbances on the state. The main aim is to provide results that allow for the efficient computation of an optimal and stabilizing state feedback control policy that ensures a given set of state and input constraints are satisfied for all time, despite the presence of the disturbances. This is a problem that has been studied for some time now in the optimal control literature (Bertsekas & Rhodes, 1973) and a number of different

& Diaz-Bobillo, 1995) or predictive control (Mayne et al., 2000; Mayne, Rawlings, Rao, & Scokaert, 2000).

It is generally accepted that if disturbances accounted for in the formulation of a constraint control problem, then the optimization has to be admissible state feedback policies, rather than open sequences, otherwise infeasibility and instability may occur (Mayne et al., 2000). However, optimization (nonlinear) feedback policies is particularly constraints have to be satisfied. Current proposals for this using finite dimensional optimization, such



Available online at www.sciencedirect.com

SCIENCE @ DIRECT®

Automatica 41 (2005) 219-224

Brief paper

Optimization over state feedback policies for robust control with constrained inputs

D.Q. Mayne^{a,*}, M.M. Seron^b, S.V. Raković^a

^aDepartment of Electrical and Electronic Engineering, Imperial College, London, SW7 2BT, UK

^bSchool of Electrical Engineering and Computer Science, University of Newcastle, New South Wales, Australia

Received 9 February 2004; received in revised form 14 July 2004; accepted 23 August 2004

Available online 8 December 2004

Abstract

This paper provides a novel solution to the problem of robust model predictive control of constrained, linear, discrete-time systems in the presence of bounded disturbances. The optimal control problem that is solved online includes, uniquely, the initial state of the model employed in the problem as a decision variable. The associated value function is zero in a disturbance invariant set that serves as the 'origin' when bounded disturbances are present, and permits a strong stability result, namely robust exponential stability of the disturbance invariant set for the controlled system with bounded disturbances, to be obtained. The resultant online algorithm is a quadratic program of similar complexity to that required in conventional model predictive control. © 2004 Elsevier Ltd. All rights reserved.

Keywords: Robust model predictive control; Robustness; Bounded disturbances

An elegant approximation

1. Introduction

Model predictive control is widely employed for the control of constrained systems and an extensive literature on the subject exists some of which is reviewed in Bemporad and Morari (1999); Allgöwer, Badgwell, Qin, Rawlings, and Wright (1999); Mayne, Rawlings, Rao, and Scokaert (2000). Several methods for achieving robustness have been considered. The simplest is to ignore the disturbance and rely on the inherent robustness of deterministic model predictive control applied to the nominal system (Scokaert & Rawlings, 1995; Marrucho, Álamo, & Camacho, 2002). *Open-loop* model predictive control that determines the

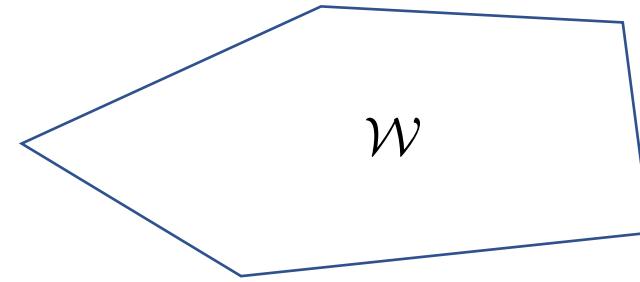
$\{u_0, u_1, \dots, u_{N-1}\}$ of control actions was proposed in Zheng and Morari (1993); this method cannot contain the 'spread' of predicted trajectories resulting from disturbances. Hence *feedback* model predictive control in which the decision variable is a *policy* π , which is a sequence $\{\mu_0(\cdot), \mu_1(\cdot), \dots, \mu_{N-1}(\cdot)\}$ of control laws, was advocated in, for example, Mayne (1995); Kothare, Balakrishnan, and Morari (1996); Lee and Yu (1997); Scokaert and Mayne (1998); De Nicolao, Magni, and Scattolini (2000); Magni, Nijmeijer, and van der Schaft (2001); Magni, De Nicolao, Scattolini, and Allgöwer (2003). Determination of a control policy is usually prohibitively difficult so research has concentrated on *receding-horizon* approximations (Mayne

What people do in practice

Optimizing over disturbance feedback policies

True system dynamics: $x_{k+1} = Ax_k + Bu_k + w_k$

Assumption: the disturbance support $\mathcal{W} = \text{conv}(w^{(0)}, \dots, w^{(n_d)})$.



The problem that we want to solve

$$\min_{\pi} f(x(t), \pi, w_{t|t}, \dots, w_{t+N|t})$$

subject to $x_{k+1|t} = Ax_{k|t} + Bu_{k|t} + w_{k|t}$
robust constraint satisfaction

$$x_{t|t} = x(t)$$

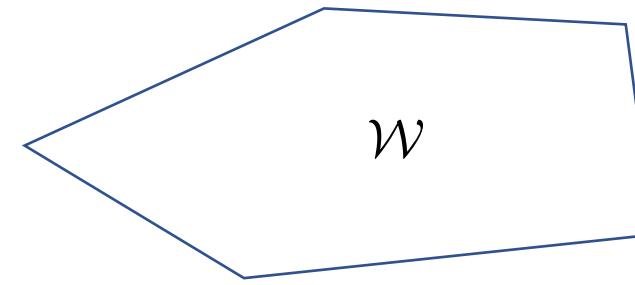
$$u_{k|t} = \pi(x_{k|t}),$$

Optimization over policies

Optimizing over disturbance feedback policies

True system dynamics: $x_{k+1} = Ax_k + Bu_k + w_k$

Assumption: the disturbance support $\mathcal{W} = \text{conv}(w^{(0)}, \dots, w^{(n_d)})$.



The problem that we want to solve

$$\begin{aligned} & \min_{\pi} \quad f(x(t), \pi, w_{t|t}, \dots, w_{t+N|t}) \\ \text{subject to} \quad & x_{k+1|t} = Ax_{k|t} + Bu_{k|t} + w_{k|t} \\ & \text{robust constraint satisfaction} \\ & x_{t|t} = x(t) \\ & u_{k|t} = \pi(x_{k|t}), \end{aligned}$$

Optimization over policies

Tractable
approximation

Optimize over disturbance feedback policies

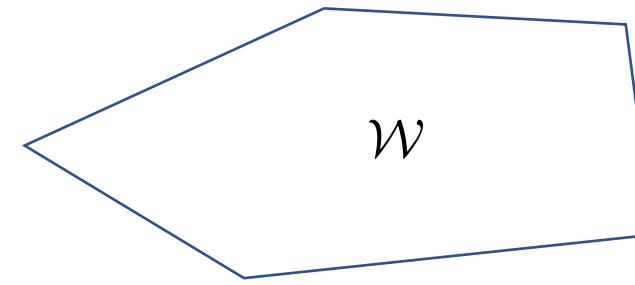
$$u_{k|t} = \pi(x(t), w_{t|t}, \dots, w_{k-1|t})$$

Function of disturbances!

Optimizing over disturbance feedback policies

True system dynamics: $x_{k+1} = Ax_k + Bu_k + w_k$

Assumption: the disturbance support $\mathcal{W} = \text{conv}(w^{(0)}, \dots, w^{(n_d)})$.



The problem that we want to solve

$$\begin{aligned} & \min_{\pi} \quad f(x(t), \pi, w_{t|t}, \dots, w_{t+N|t}) \\ \text{subject to} \quad & x_{k+1|t} = Ax_{k|t} + Bu_{k|t} + w_{k|t} \\ & \text{robust constraint satisfaction} \end{aligned}$$

$$x_{t|t} = x(t)$$
$$u_{k|t} = \pi(x_{k|t}),$$

Optimization over policies

Tractable
approximation

Optimize over disturbance feedback policies

$$u_{k|t} = \pi(x(t), w_{t|t}, \dots, w_{k-1|t})$$

Function of disturbances!

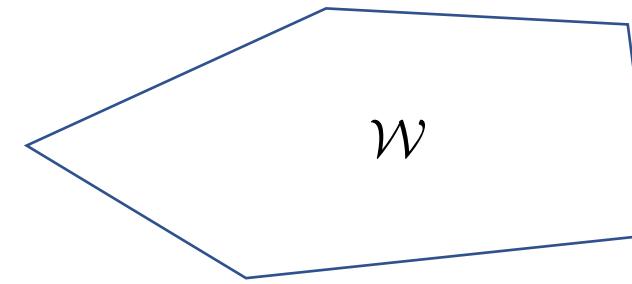
The problem is **tractable** when we pick:

$$u_{k|t} = M_{k|t} \begin{bmatrix} x(t) \\ w_{t|t} \\ \vdots \\ w_{k-1|t} \end{bmatrix}$$

Optimizing over disturbance feedback policies

True system dynamics: $x_{k+1} = Ax_k + Bu_k + w_k$

Assumption: the disturbance support $\mathcal{W} = \text{conv}(w^{(0)}, \dots, w^{(n_d)})$.



The problem that we want to solve

$$\begin{array}{ll} \min_{\pi} & f(x(t), \pi, w_{t|t}, \dots, w_{t+N|t}) \\ \text{subject to} & x_{k+1|t} = Ax_{k|t} + Bu_{k|t} + w_{k|t} \\ & \text{robust constraint satisfaction} \end{array}$$

$$x_{t|t} = x(t)$$

$$u_{k|t} = \pi(x_{k|t}),$$

Optimization over policies

Tractable
approximation

Optimize over disturbance feedback policies

$$u_{k|t} = \pi(x(t), w_{t|t}, \dots, w_{k-1|t})$$

Function of disturbances!

The problem is **tractable** when we pick:

$$u_{k|t} = M_{k|t} \begin{bmatrix} x(t) \\ w_{t|t} \\ \vdots \\ w_{k-1|t} \end{bmatrix}$$

This approach is **not tractable** when A and B are uncertain for more details check:

Aharon Ben-Tal, Alexander Goryashko, Elana Guslitzer, and Arkadi Nemirovski. "Adjustable robust solutions of uncertain linear programs." *Mathematical programming* 99, no. 2 (2004): 351-376.

How do we fix this problem?

1136

IEEE TRANSACTIONS ON AUTOMATIC CONTROL, VOL. 43, NO. 8, AUGUST 1998

Statement 2: Under the assumptions of this theorem for any $0 \leq t_0 \leq t$

$$\begin{aligned}\Phi_{\omega}(t) - \Phi_{\omega}(t_0) &\geq \Phi_{\omega^*}(t) - \Phi_{\omega^*}(t_0) \\ &= -\frac{1}{h} \Delta \vartheta^T(t) L^{-1} \Delta \vartheta(t) + o_\omega(h)\end{aligned}$$

where

$$\Phi_{\omega}(t) := \int_{t_0}^t (2\bar{u}^T(\tau) d\vartheta(\tau) + \bar{u}^T(\tau) L \bar{u}(\tau) d\tau) \quad (20)$$

$$\begin{aligned}\Delta \vartheta(t) &:= B_{\omega}^T P[B_{\omega}(a_{\text{comp}}(t) - x^*(t)) h \\ &+ (K + A_3 C_b^+) \Delta y(t)], \quad \Delta y(t) = y(t) - y(t-h)\end{aligned}$$

and this minimum is reachable for

$$\begin{aligned}\bar{u}(t) dt &= \bar{u}^*(t) dt := -L^{-1} B_{\omega}^T P[B_{\omega}(a_{\text{comp}}(t) + x^*(t)) dt \\ &+ (K + A_3 C_b^+) dy(t)].\end{aligned}$$

As a result, we have: $d\Phi_{\omega^*}(t) \leq 0$ (in symbolic form).

Proof of Statement 2: Using the Euler-Maruyama's formula [3], [9] we obtain the following relation: $(h := t - t_0 \rightarrow 0), \Phi_{\omega}(t) - \Phi_{\omega}(t_0) = 2\bar{u}^T(t) \Delta \vartheta(t) + \bar{u}^T(t) L \bar{u}(t) h + o_\omega(h)$.

Minimizing then the right side for each fixed t , we derive

$$\begin{aligned}\min_{\omega} [\Phi_{\omega}(t) - \Phi_{\omega}(t_0)] &= \Phi_{\omega^*}(t) - \Phi_{\omega^*}(t_0) \\ &= -\frac{1}{h} \Delta \vartheta^T(t) L^{-1} \Delta \vartheta(t) + o_\omega(h) \leq o_\omega(h).\end{aligned}\quad (21)$$

Hence, taking into account the definition (20), we have $d\Phi_{\omega^*}(t) \leq 0$. \square

Rewriting (18) in differential form and taking into account that P is the solution of the Riccati equation (see Assumption 4 of this theorem) we finally derive

$$dV(e(t)) \leq [\varphi(t) + I(t)] dt + S^T(t) dw(t) - e^T(t) Q e(t) dt.$$

According to the Kronecker lemma, the Large Number law for martingale, and Skorohod lemma (see [2] and [4]), we obtain the result of the theorem. Details of this proof can be found in [13].

REFERENCES

- [1] S. Boyd, L. Ghaoui, E. Feron, and V. Balakrishnan, *Linear Matrix Inequalities in System and Control Theory*. Philadelphia, PA: SIAM, 1994.
- [2] H. Cramer and M. Lidbetter, *Stationary Random Processes*. Princeton, 1969.
- [3] T. Gard, *Introduction to Stochastic Differential Equations*. New York: Marcel Dekker, 1988.
- [4] I. I. Gilman and A. V. Skorohod, *Stochastic Differential Equations*. New York: Springer-Verlag, 1972.
- [5] M. Grimble, "Three and half DOF polynomial solution of the feed-forward H_2/H_∞ control problem," in Proc. 34rd Conf. Decision and Control, New Orleans, LA, Dec. 1995, pp. 4145-4150.

The problem is NP-hard!

- [12] A. Poznyak and M. Taksar, "Robust state feedback stochastic control for linear systems with time-varying parameters," in Proc. 33rd Conf. Decision and Control, Lake Buena Vista, FL, Dec. 1994, pp. 1161-1162.
- [13] ———, "Robust control of linear stochastic systems with fully observable state," *Applications Mathematicae*, vol. 24, no. 1, pp. 35-46, 1996.
- [14] M. I. Taksar, A. S. Poznyak, and A. Ipparragirre, "Robust feedback control for linear stochastic systems in continuous time with time-varying parameters," in Proc. 34rd Conf. Decision and Control, New Orleans, LA, Dec. 1995, pp. 2179-2184.
- [15] J. C. Willems, "Least squares stationary optimal control and a Riccati equation," *IEEE Trans. Automat. Contr.*, vol. AC-16, no. 621-634, 1971.

- Available online at www.sciencedirect.com
SCIENCE @ DIRECT[®]
- automatica
- Automatica 42 (2006) 523-533
- www.elsevier.com/locate/automatica
- Available online at www.sciencedirect.com
SCIENCE @ DIRECT[®]
- automatica
- Automatica 41 (2005) 219-224
- www.elsevier.com/locate/automatica
- Brief paper
- Robust model predictive control of constrained linear systems with bounded disturbances[☆]
- D.Q. Mayne^{a,*}, M.M. Seron^b, S.V. Raković^a
- ^aDepartment of Engineering, University of Cambridge, Trumpington Street, Cambridge CB2 1FW, UK
^bDepartment of Aeronautics and Department of Electrical and Electronic Engineering, Imperial College London, Exhibition Road, London SW7 2BY, UK
- Received 15 March 2005; received in revised form 8 June 2005; accepted 25 August 2005
- Abstract
- This paper is concerned with the optimal control of linear discrete-time systems subject to unknown but bounded state disjunctive polytopic constraints on the state and input. It is shown that the class of admissible affine state feedback control policies with prior states is equivalent to the class of admissible feedback policies that are affine functions of the past disturbance sequence. That a broad class of constrained finite horizon robust and optimal control problems, where the optimization is over affine policies, can be solved in a computationally efficient fashion using convex optimization methods. This equivalence result is used to robust receding horizon control (RHC) state feedback policy such that the closed-loop system is input-to-state stable (ISS) and it is satisfied for all time and all allowable disturbance sequences. The cost to be minimized in the associated finite horizon optimization is quadratic in the disturbance-free state and input sequences. The value of the receding horizon control law can be calculated sample instant using a single, tractable and convex quadratic program (QP) if the disturbance set is polytopic, or a tractable second program (SOCP) if the disturbance set is given by a 2-norm bound.
- © 2006 Elsevier Ltd. All rights reserved.
- The MPC strategy optimizes an open-loop control sequence each sample, to minimize a nominal cost function, subject to state and input constraints. The optimization is usually based on the assumption that the process model is exact and that disturbances are constant. Because the control law ignores the possibility of future changes in disturbance and model mismatch, closed-loop performance can be poor [3], with likely violating the constraints, when disturbances or model mismatch are present.
- Some formulations of MPC have been proposed that address issues [4], [5]. These methods rely on a min-max optimization predicted performance. However, these formulations optimize a control profile over all possible disturbance (or model mismatch) sequences, otherwise infeasibility and instability may occur [Mayne et al., 2000]. However, optimization of trajectory (nonlinear) feedback policies is particularly constrained to satisfy the constraints. Current proposals for this use finite dimensional optimization, such as Diaz-Bobillo, 1995) or predictive control (Mayne et al., 2002; Mayne, Rawlings, Rao, & Scokaert, 2000).
- It is generally accepted that if disturbances are accounted for in the formulation of a constraint control problem, then the optimization has to be able to handle state feedback policies, rather than open-loop sequences, otherwise infeasibility and instability may occur [Mayne et al., 2000]. However, optimization of trajectory (nonlinear) feedback policies is particularly constrained to satisfy the constraints. Current proposals for this use finite dimensional optimization, such as Diaz-Bobillo, 1995) or predictive control (Mayne et al., 2002; Mayne, Rawlings, Rao, & Scokaert, 2000).
- Keywords: Robust control; Constraint satisfaction; Robust optimization; Predictive control; Optimal control
1. Introduction
- This paper is concerned with the control of constrained discrete-time linear systems that are subject to additive, but bounded disturbances on the state. The main aim is to provide results that allow for the efficient computation of an optimal and stabilizing state feedback control policy that ensures a given set of state and input constraints are satisfied for all time, despite the presence of the disturbances. This is a problem that has been studied for some time now in the optimal control literature (Bertsekas & Rhodes, 1973) and a number of different
- © 2004 Elsevier Ltd. All rights reserved.
- Keywords: Robust model predictive control; Robustness; Bounded disturbances
1. Introduction
- Model predictive control is widely employed for the control of constrained systems and an extensive literature on the subject exists some of which is reviewed in Bemporad and Morari (1999); Allgöwer, Badgwell, Qin, Rawlings, and Wright (1999); Mayne, Rawlings, Rao, and Scokaert (2000). Several methods for achieving robustness have been considered. The simplest is to ignore the disturbance and rely on the inherent robustness of deterministic model predictive control applied to the nominal system (Scokaert & Rawlings, 1995; Marruedo, Álamo, & Camacho, 2002). Open-loop model predictive control that determines the
- [☆]What people do in practice

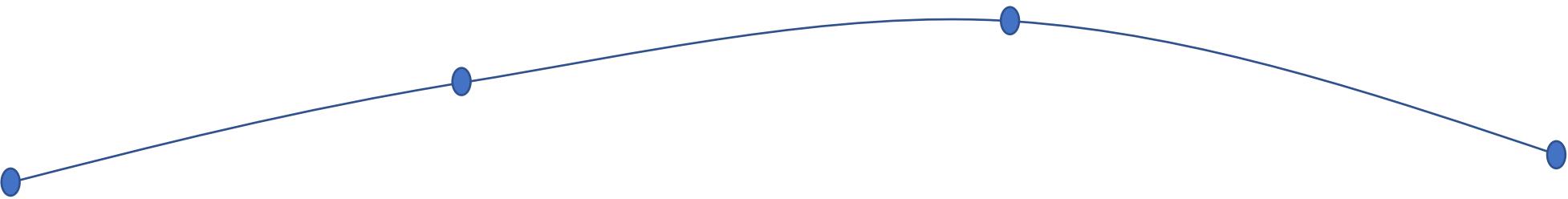
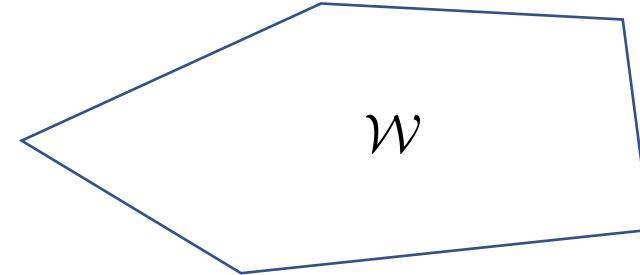
An elegant approximation

What people do in practice

Fixed tube robust MPC

True system dynamics: $x_{k+1} = Ax_k + Bu_k + w_k$

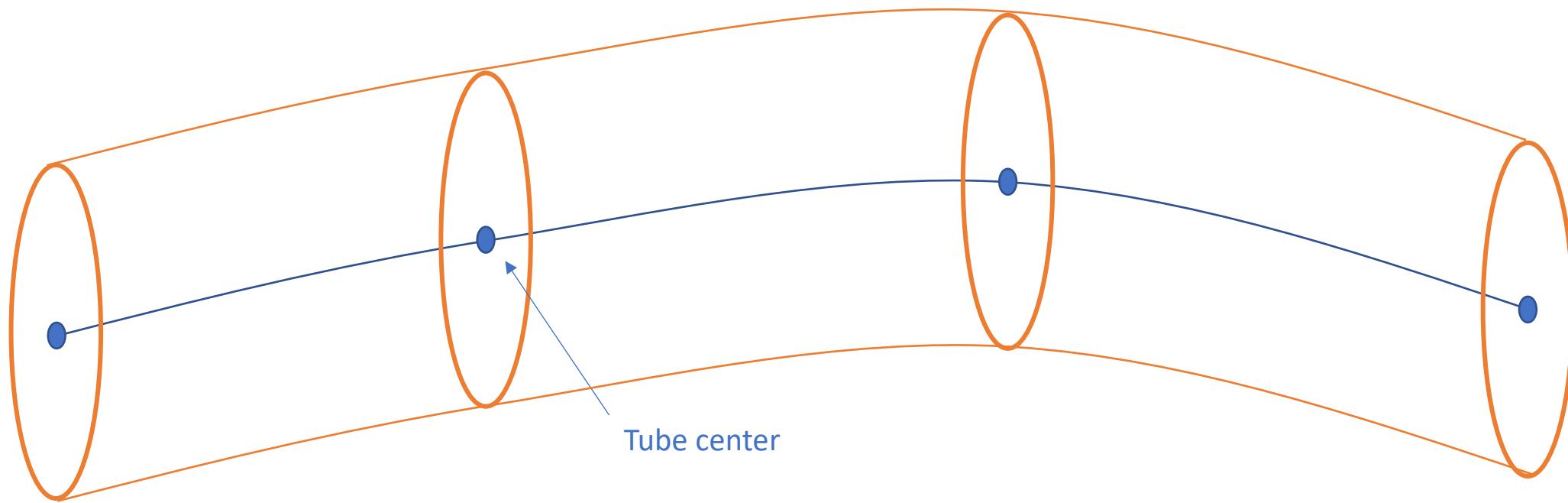
Assumption: the disturbance support $\mathcal{W} = \text{conv}(w^{(0)}, \dots, w^{(n_d)})$.



Fixed tube robust MPC

True system dynamics: $x_{k+1} = Ax_k + Bu_k + w_k$

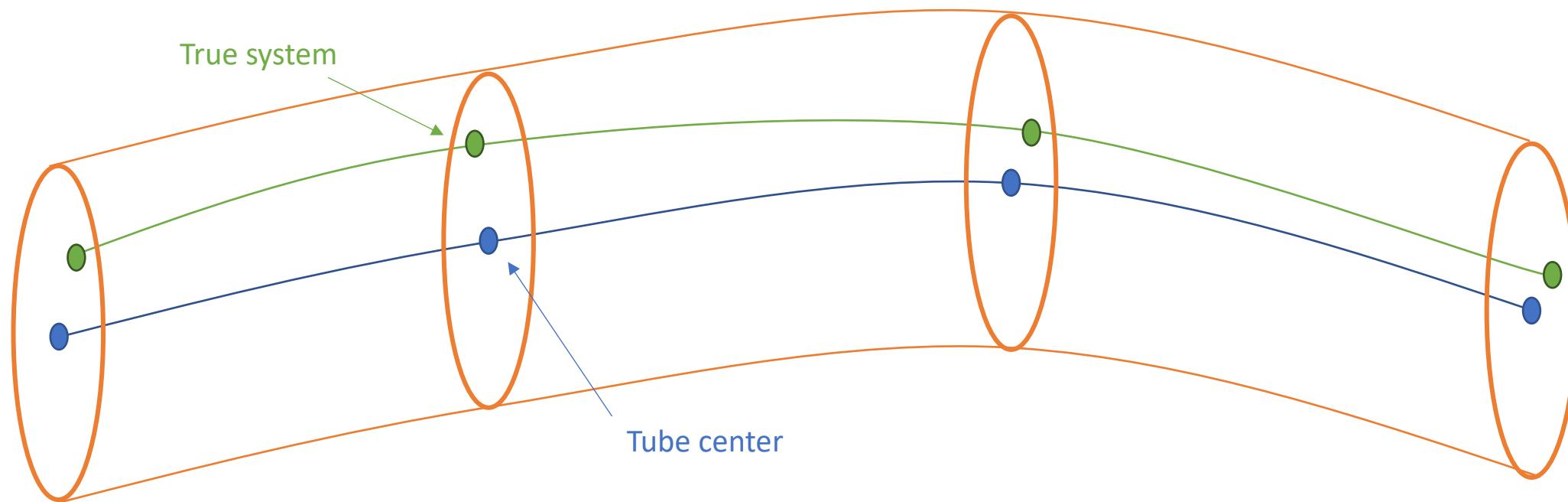
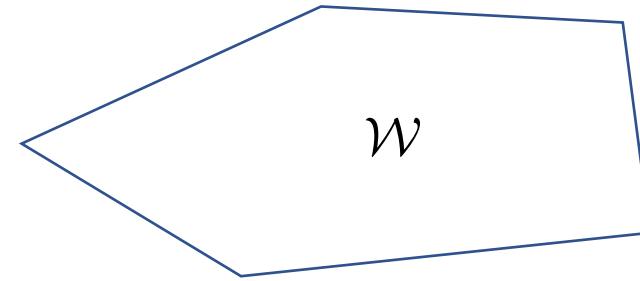
Assumption: the disturbance support $\mathcal{W} = \text{conv}(w^{(0)}, \dots, w^{(n_d)})$.



Fixed tube robust MPC

True system dynamics: $x_{k+1} = Ax_k + Bu_k + w_k$

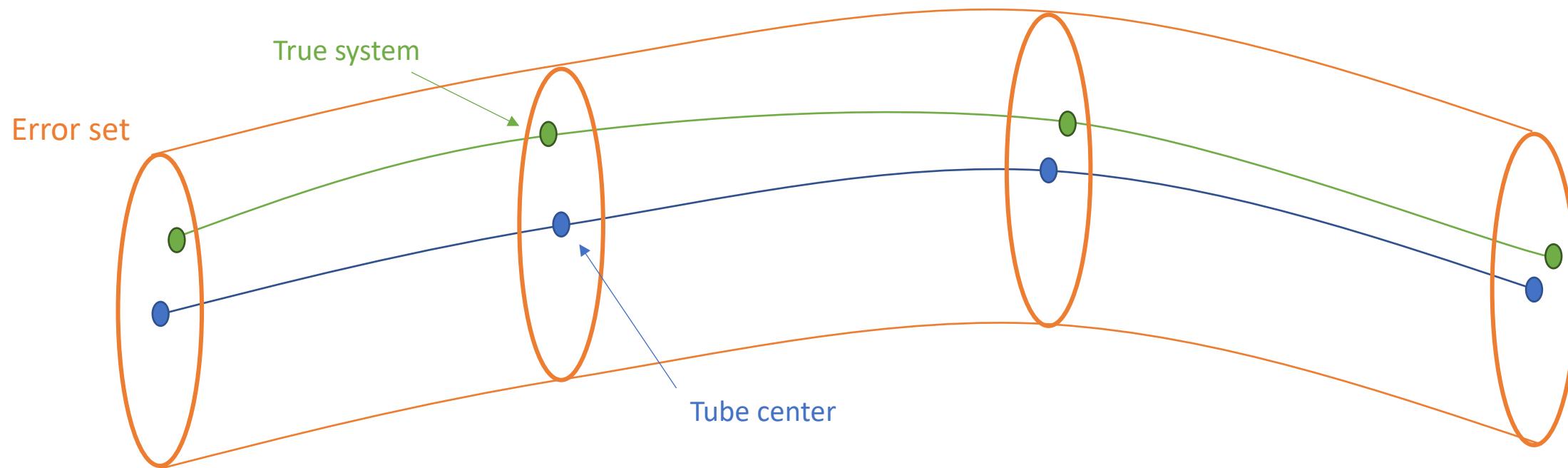
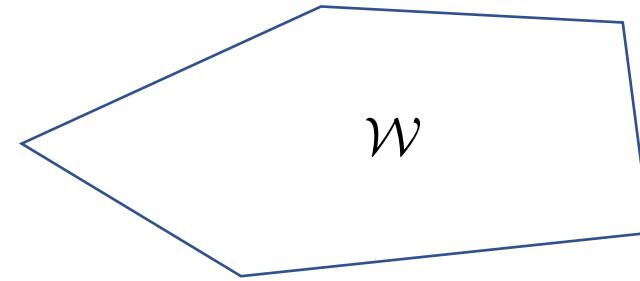
Assumption: the disturbance support $\mathcal{W} = \text{conv}(w^{(0)}, \dots, w^{(n_d)})$.



Fixed tube robust MPC

True system dynamics: $x_{k+1} = Ax_k + Bu_k + w_k$

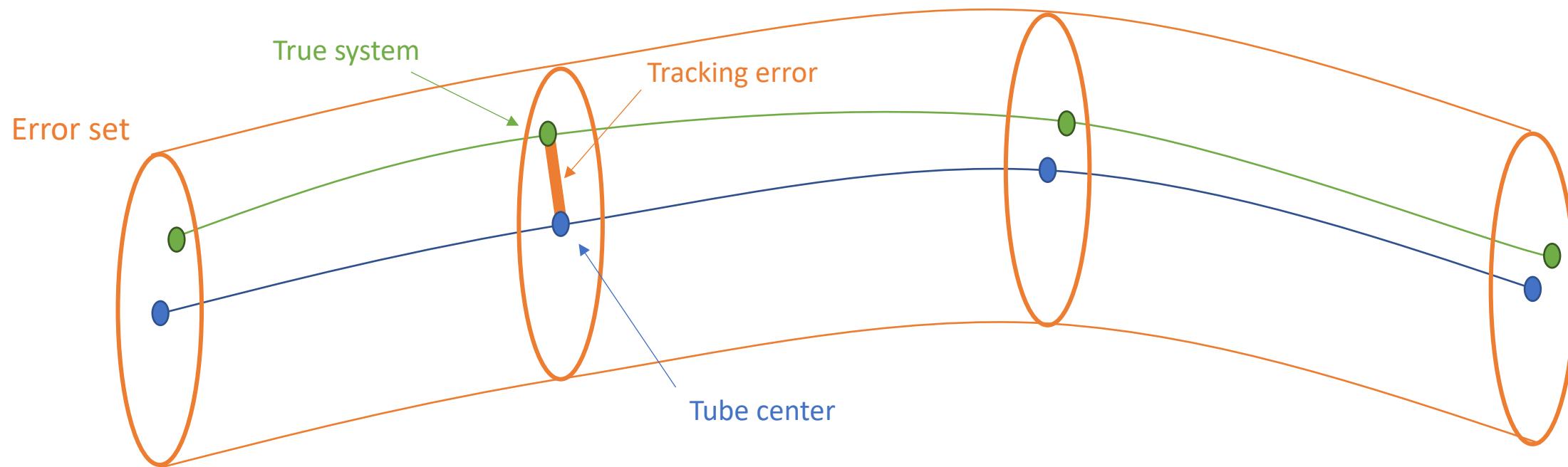
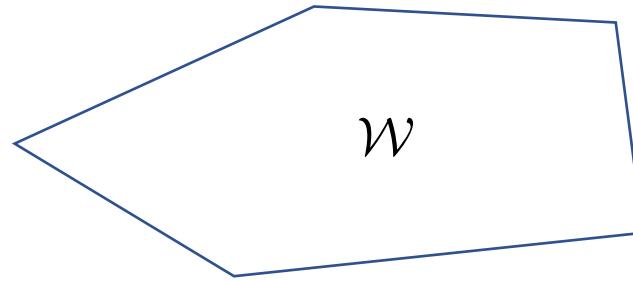
Assumption: the disturbance support $\mathcal{W} = \text{conv}(w^{(0)}, \dots, w^{(n_d)})$.



Fixed tube robust MPC

True system dynamics: $x_{k+1} = Ax_k + Bu_k + w_k$

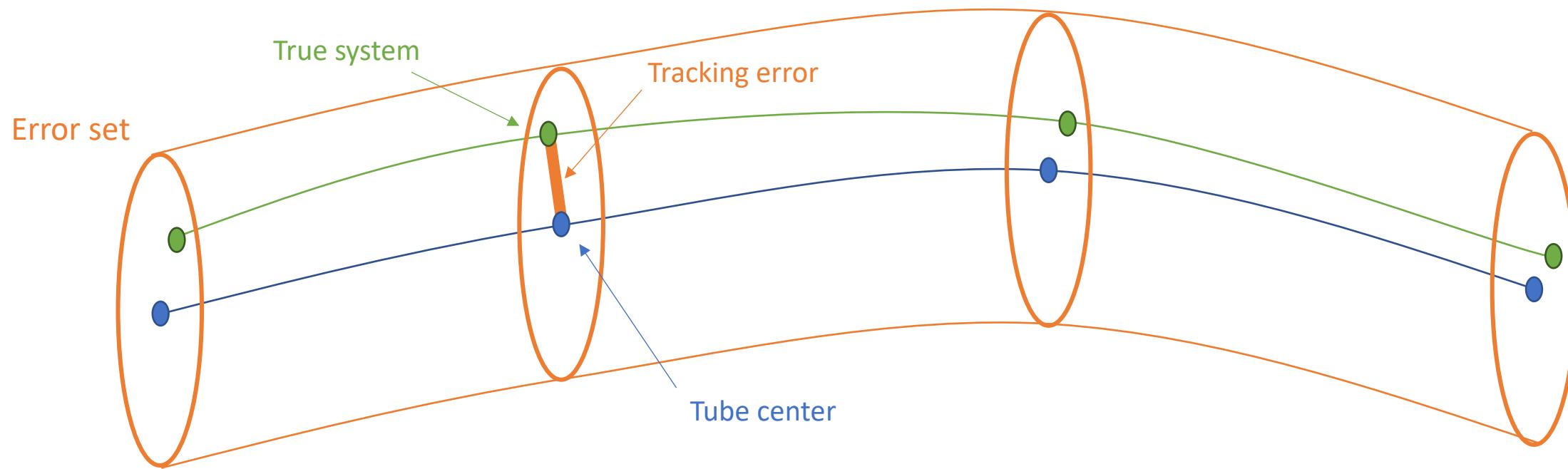
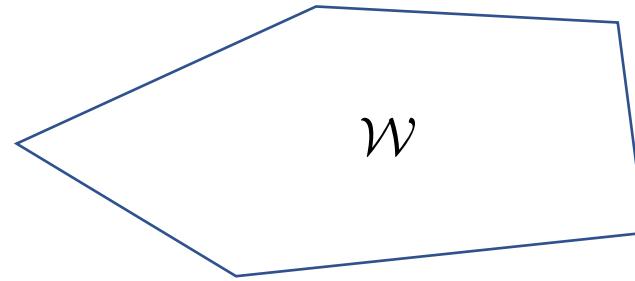
Assumption: the disturbance support $\mathcal{W} = \text{conv}(w^{(0)}, \dots, w^{(n_d)})$.



Fixed tube robust MPC

True system dynamics: $x_{k+1} = Ax_k + Bu_k + w_k$

Assumption: the disturbance support $\mathcal{W} = \text{conv}(w^{(0)}, \dots, w^{(n_d)})$.



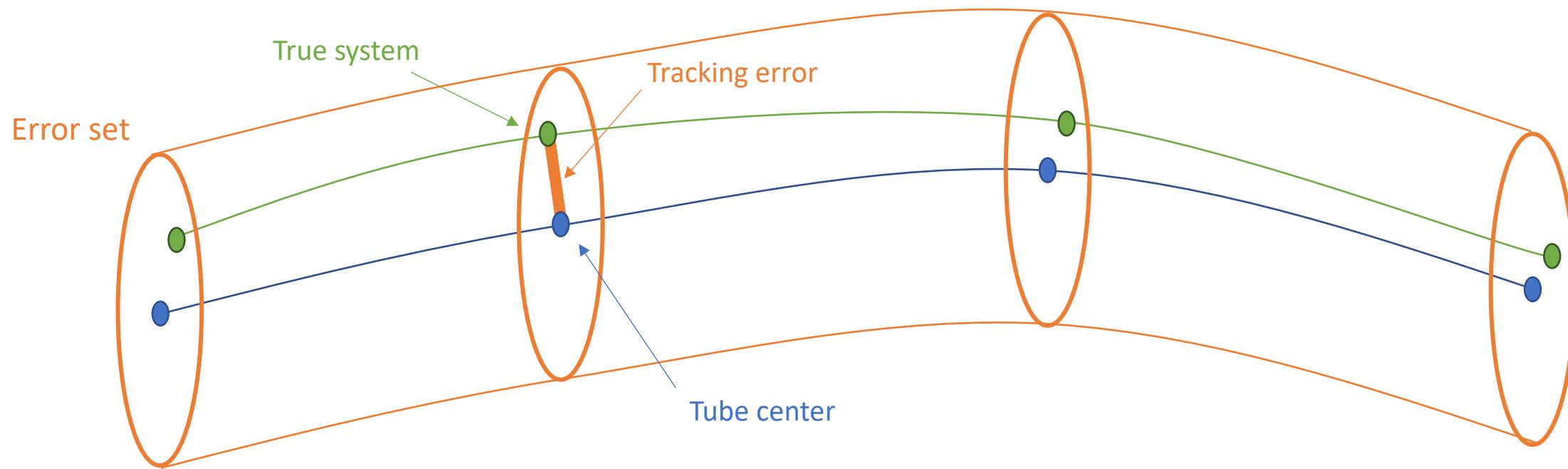
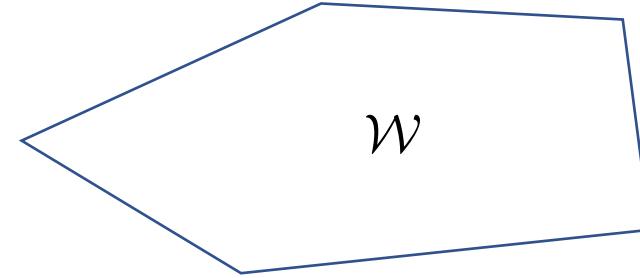
Key Ideas:

- ▶ Tube center planned via MPC
- ▶ Feedback to reduce tracking error

Fixed tube robust MPC

True system dynamics: $x_{k+1} = Ax_k + Bu_k + w_k$

Assumption: the disturbance support $\mathcal{W} = \text{conv}(w^{(0)}, \dots, w^{(n_d)})$.



Key Ideas:

- ▶ Tube center planned via MPC
- ▶ Feedback to reduce for tracking error

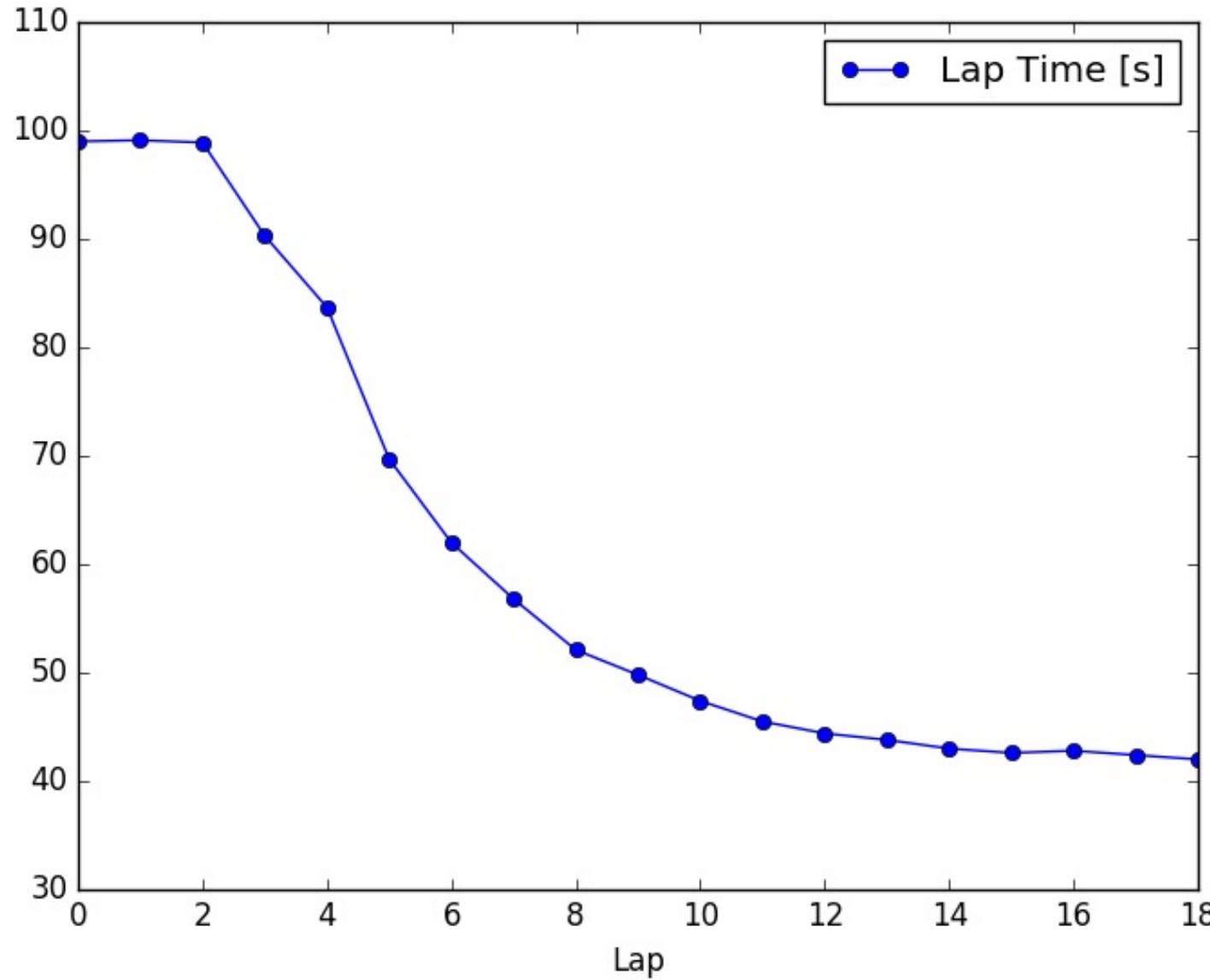


$$\pi(x) = \pi^{\text{MPC}}(x) + \pi^{\text{tracking}}(x, \text{tube center})$$

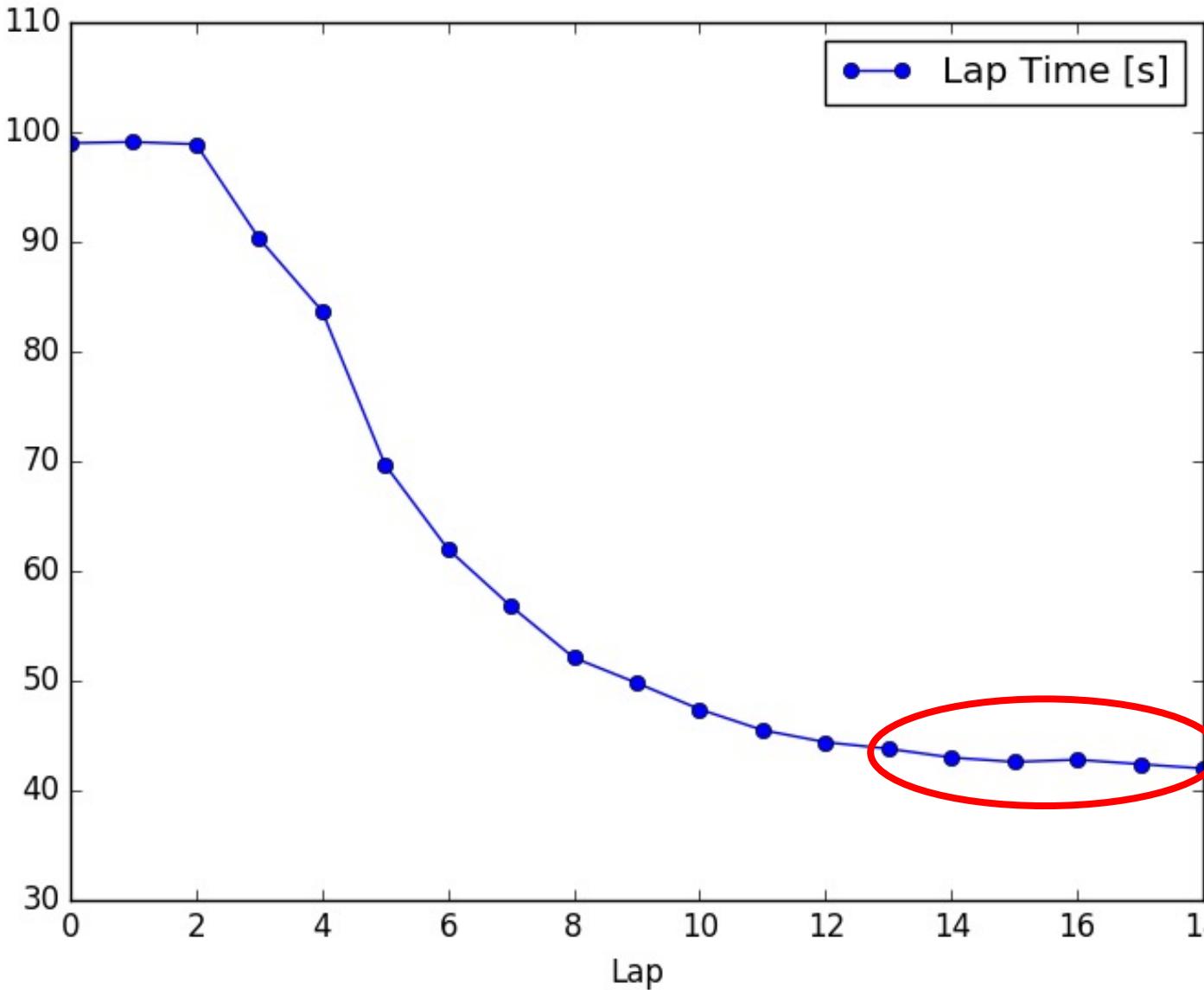
Model-free Control

How to reduce the computation load associated with MPC policies

Motivation: Lap Time Improvement



Goal: Reduce Computational Complexity



At convergence the
control policy does
not change

Learning as Computational Load Reduction

THINKING,
FAST AND SLOW

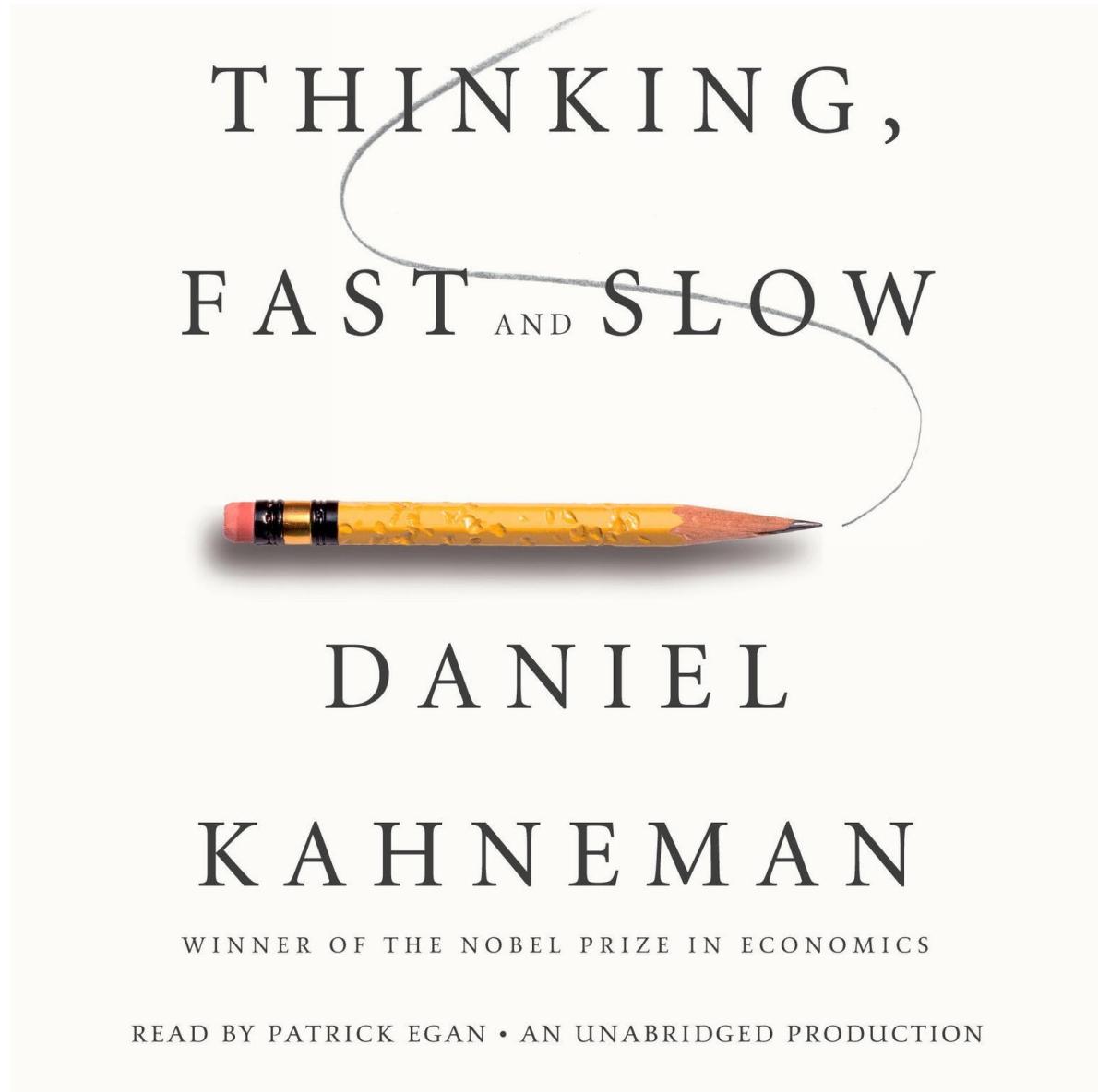


DANIEL
KAHNEMAN

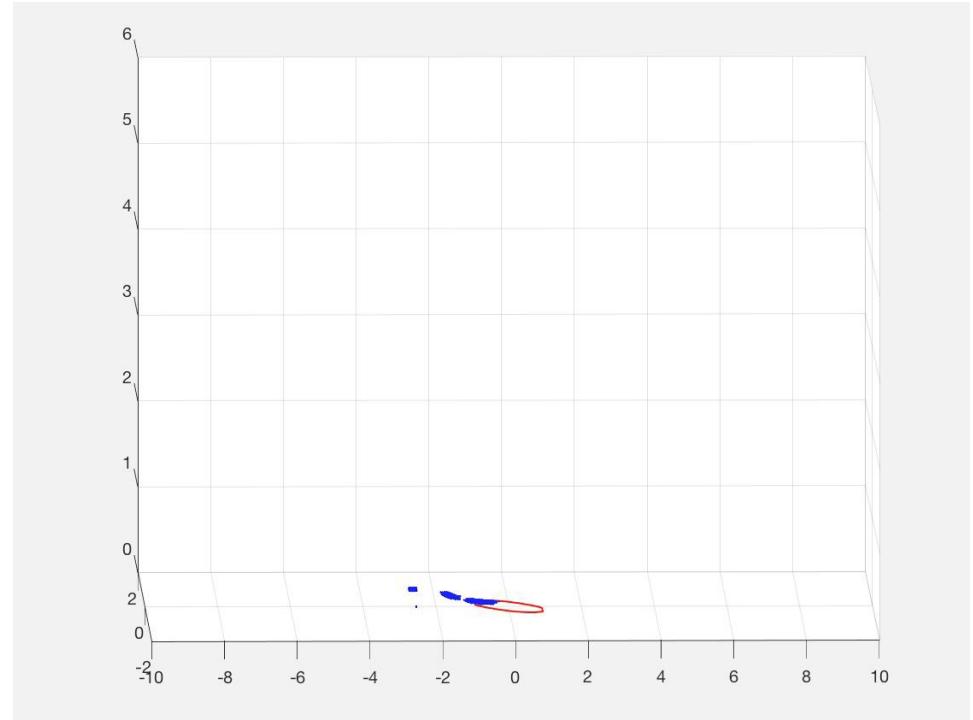
WINNER OF THE NOBEL PRIZE IN ECONOMICS

READ BY PATRICK EGAN • AN UNABRIDGED PRODUCTION

Learning as Computational Load Reduction

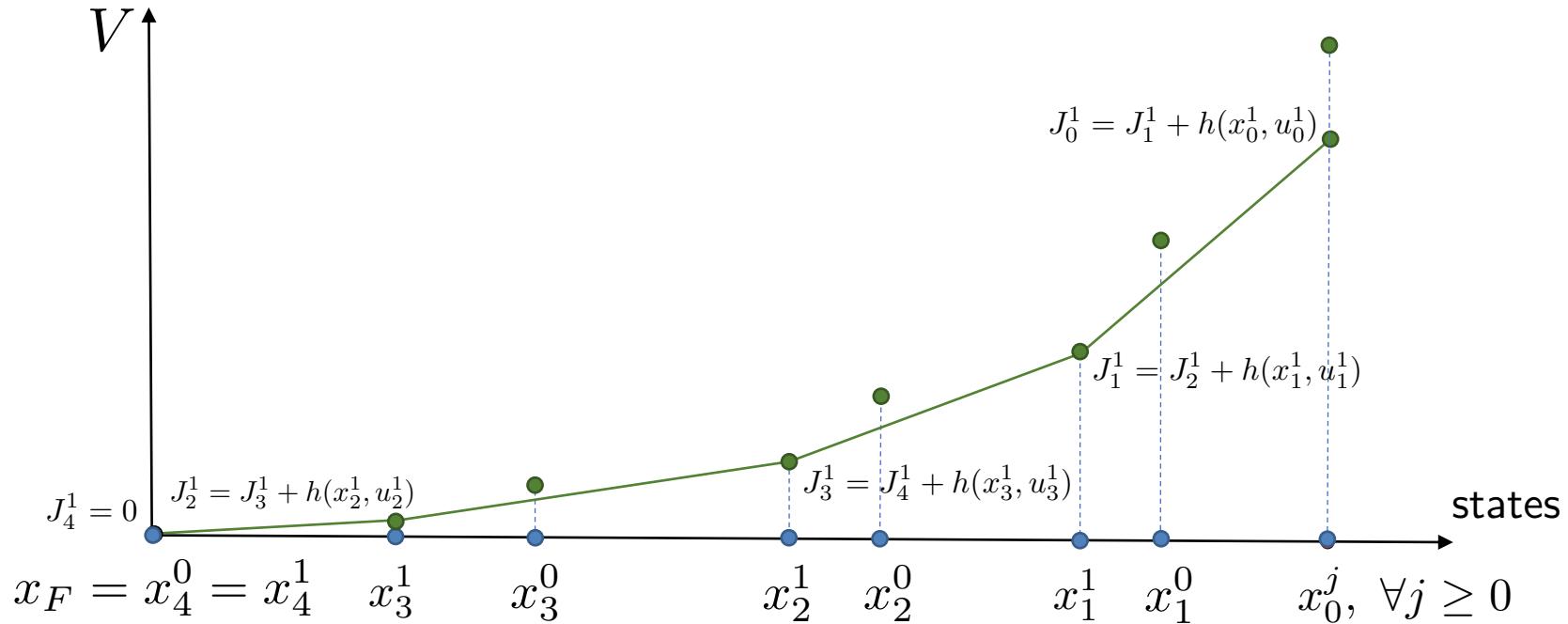


At convergence: the value function approximation does not change as we collect more data



Can we design a policy which used data and “plays back” the inputs that we computed with the expensive policy

Data-Based Policy

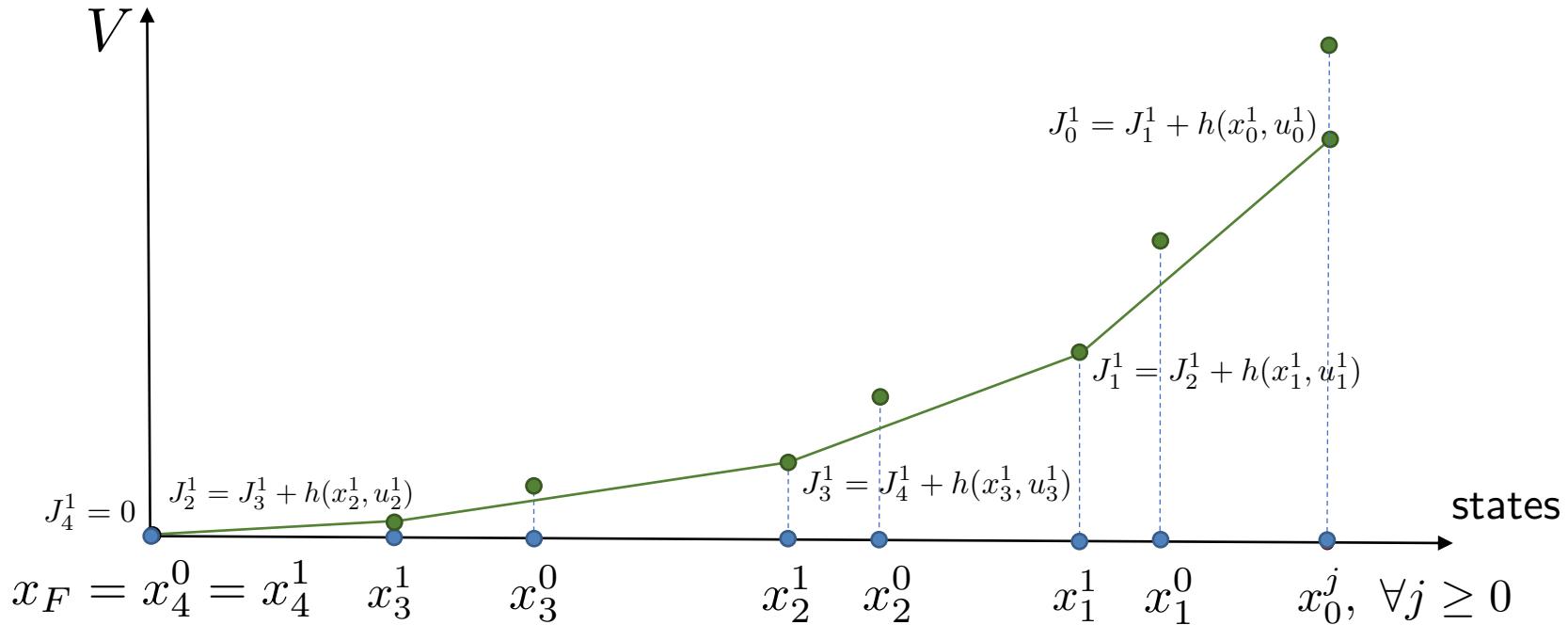


Multipliers Defining the Value Function Approximation

$$[\lambda_0^{0,*}, \dots, \lambda_{T^j}^{T^j,*}] = \arg \min_{\lambda_i^j \in [0,1]} \sum_i \sum_j J_i^j \lambda_i^j$$

$$\text{s.t} \quad \sum_i \sum_j x_i^j \lambda_i^j = \textcolor{red}{x}, \\ \sum_i \sum_j \lambda_i^j = 1$$

Data-Based Policy



Multipliers Defining the Value Function Approximation

$$[\lambda_0^{0,*}, \dots, \lambda_{T^j}^{T^j,*}] = \arg \min_{\lambda_i^j \in [0,1]} \sum_i \sum_j J_i^j \lambda_i^j$$

s.t

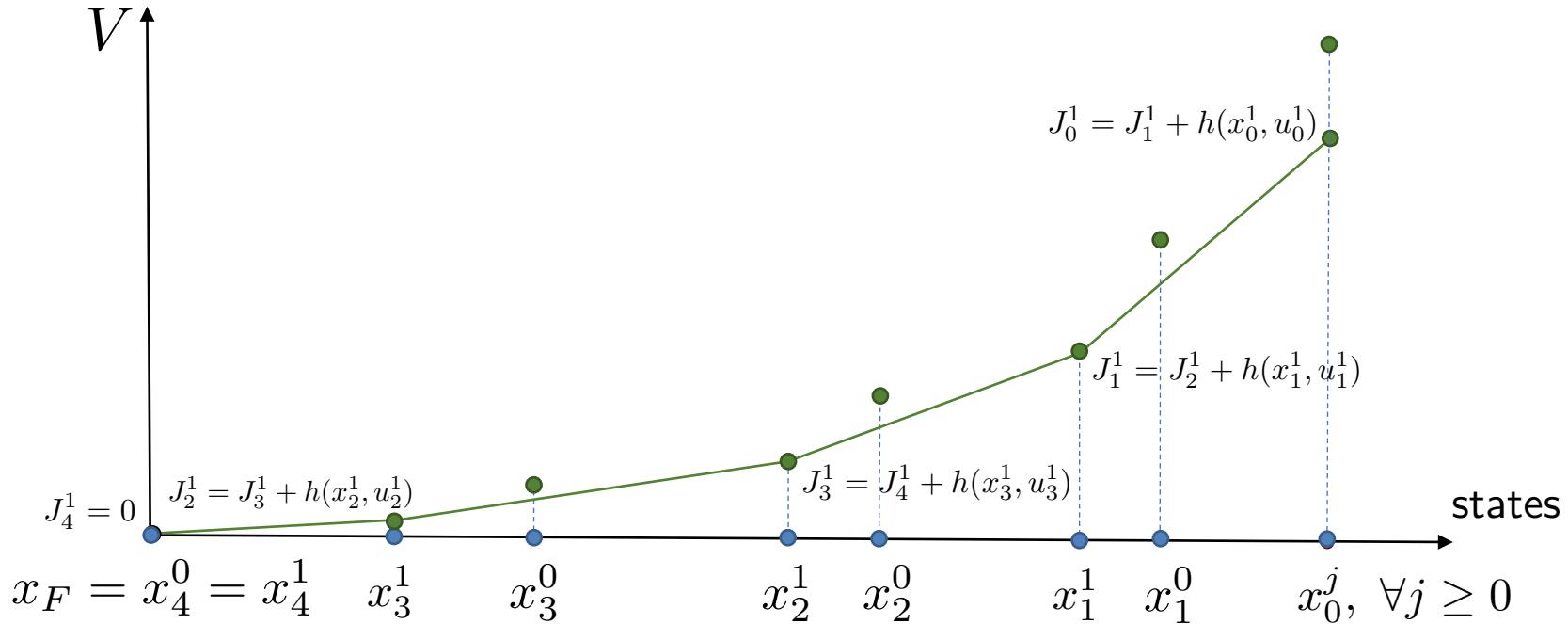
$$\sum_i \sum_j x_i^j \lambda_i^j = \textcolor{red}{x},$$

$$\sum_i \sum_j \lambda_i^j = 1$$

Data-Based Policy

$$\kappa^j(\textcolor{red}{x}) = \sum_i \sum_j u_i^j \lambda_i^{j,*}$$

Data-Based Policy



Multipliers Defining the Value Function Approximation

$$[\lambda_0^{0,*}, \dots, \lambda_{T^j}^{T^j,*}] = \arg \min_{\lambda_i^j \in [0,1]} \sum_i \sum_j J_i^j \lambda_i^j$$

s.t.

$$\sum_i \sum_j x_i^j \lambda_i^j = \textcolor{red}{x},$$

$$\sum_i \sum_j \lambda_i^j = 1$$

Data-Based Policy

$$\kappa^j(\textcolor{red}{x}) = \sum_i \sum_j u_i^j \lambda_i^{j,*}$$

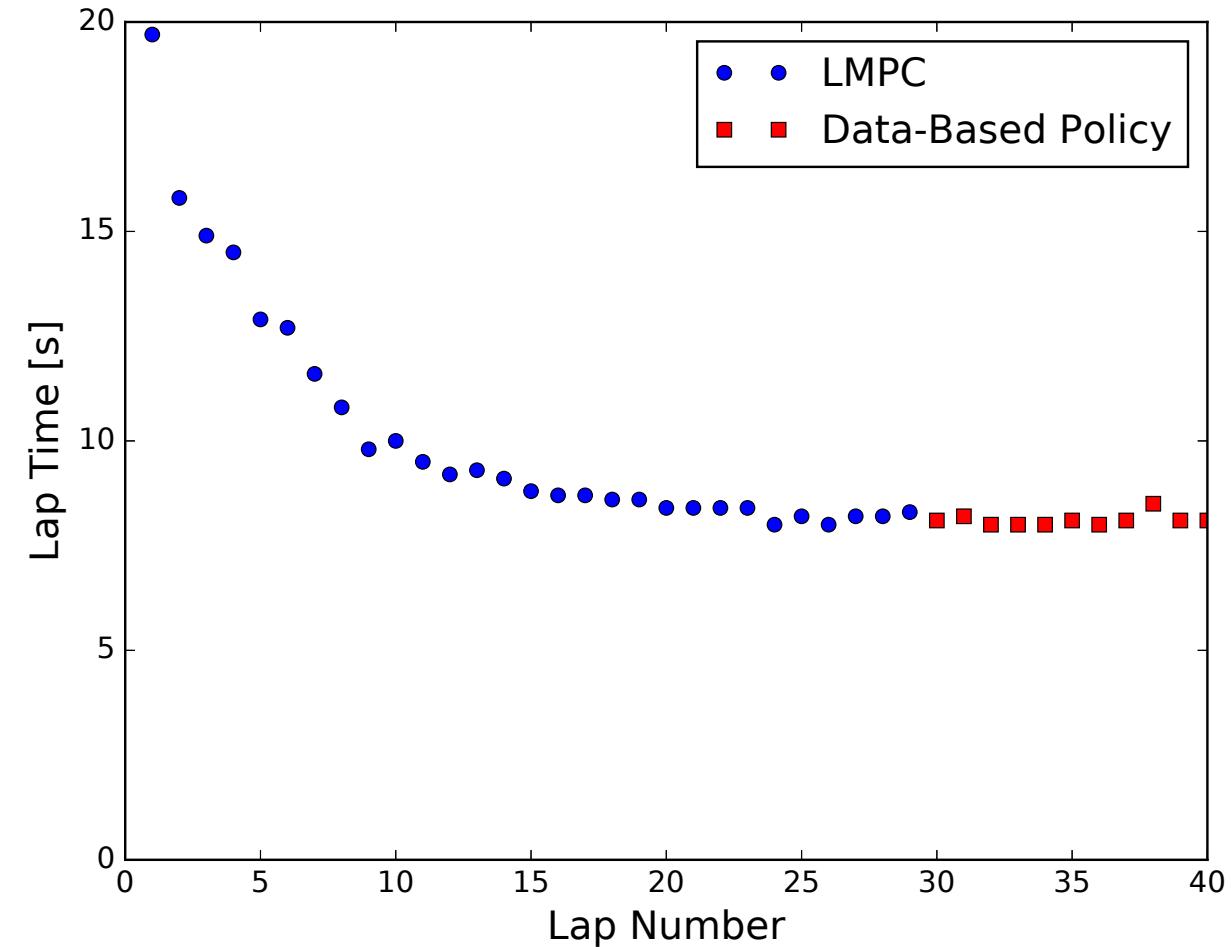
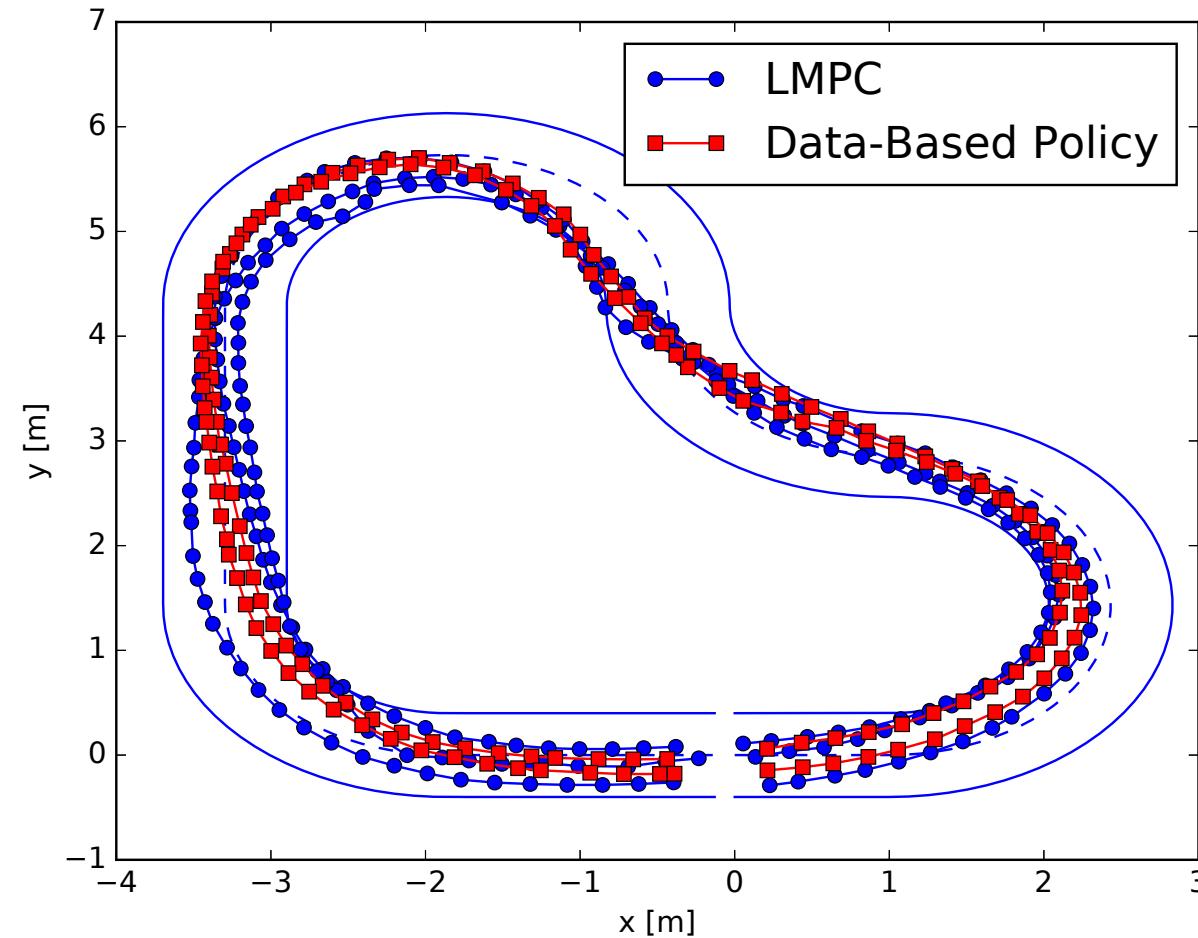
Historical Data

Berkeley Autonomous Race Car Implementation



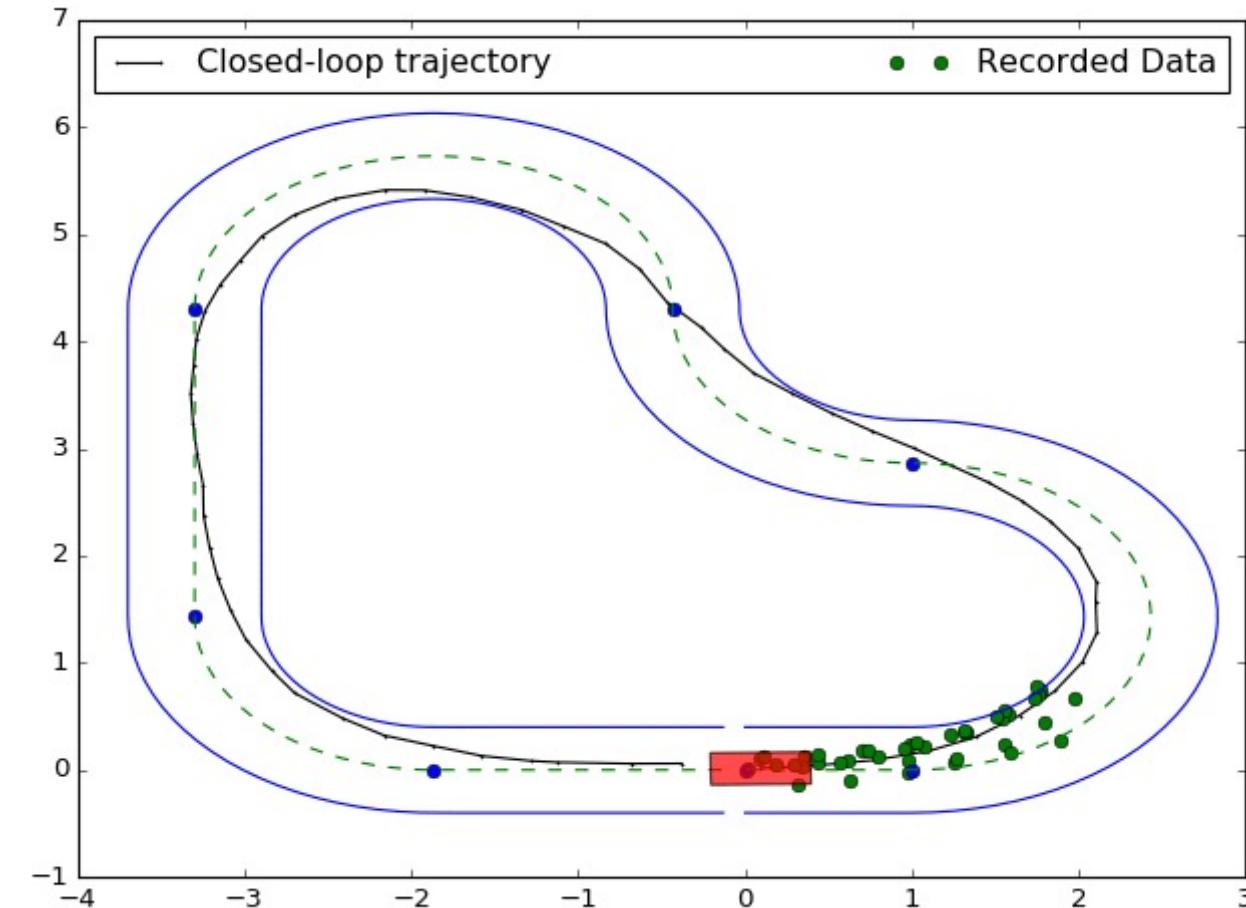
Learning Model Predictive Control
for Autonomous Racing

Berkeley Autonomous Race Car Implementation



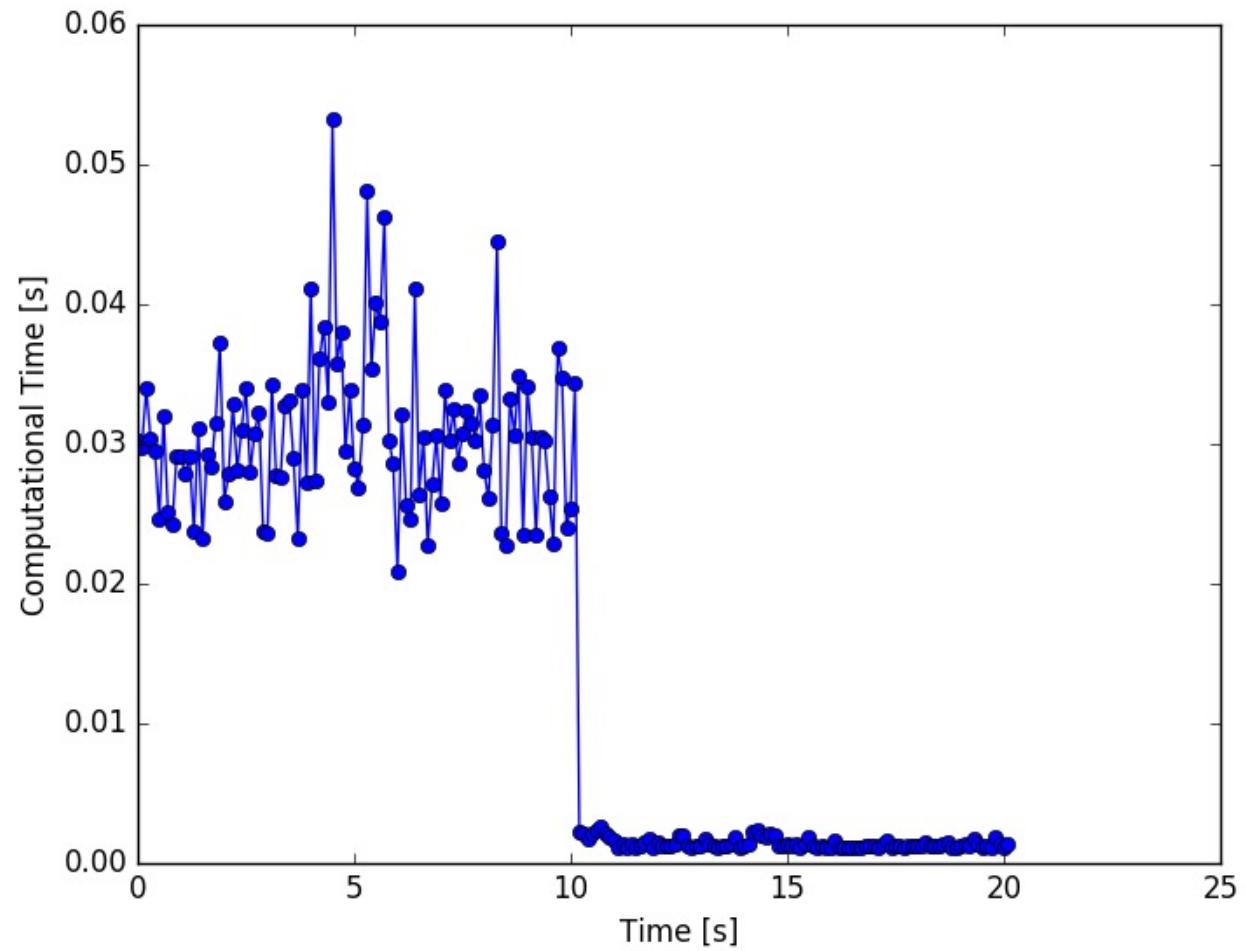
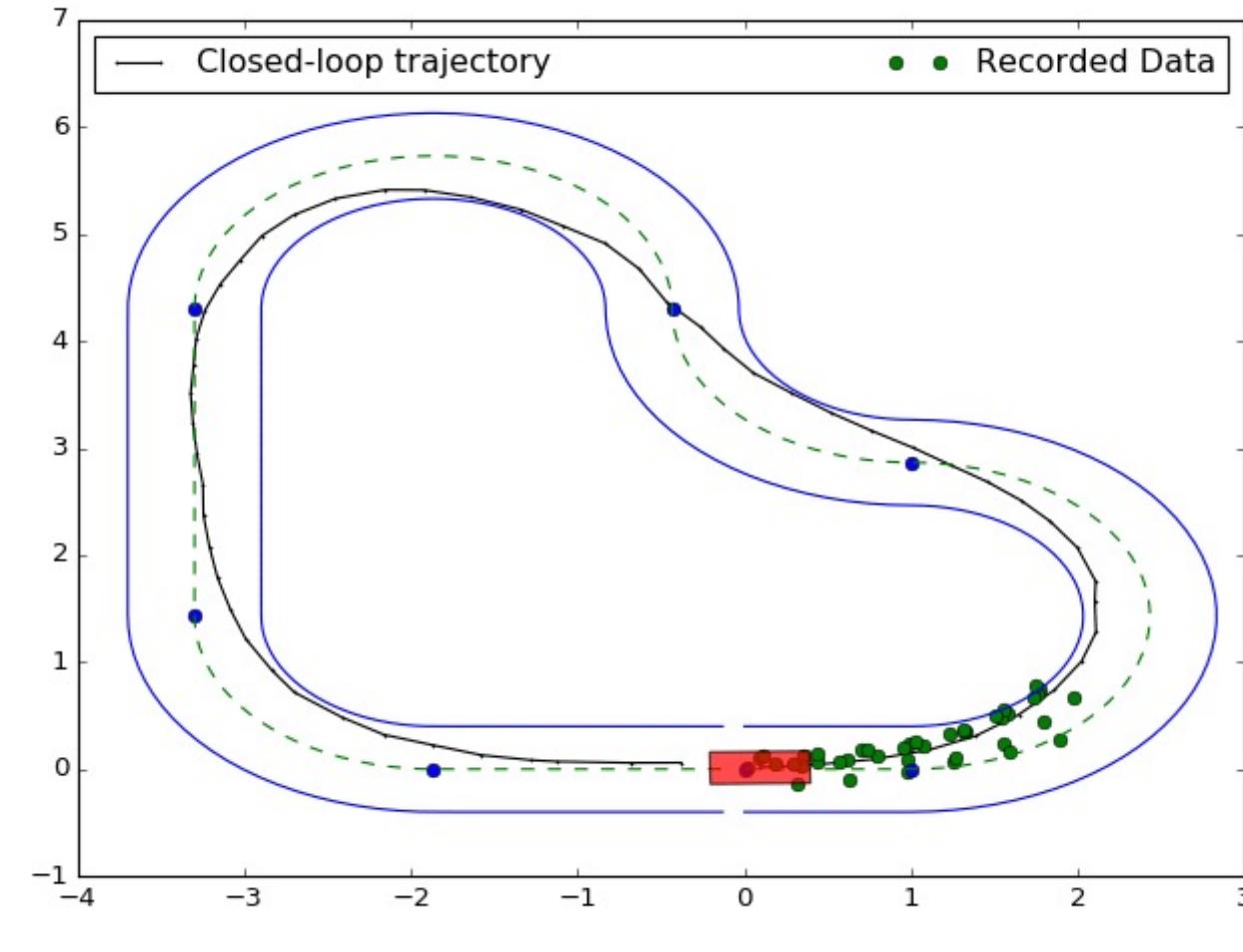
- ▶ A **reference trajectory** is **not needed** for the controller implementation
- ▶ The controller converges after few laps: the learning process is **data efficient**
- ▶ The controller **safely explores** the state space **iteratively improving** the lap time

Berkeley Autonomous Race Car Implementation



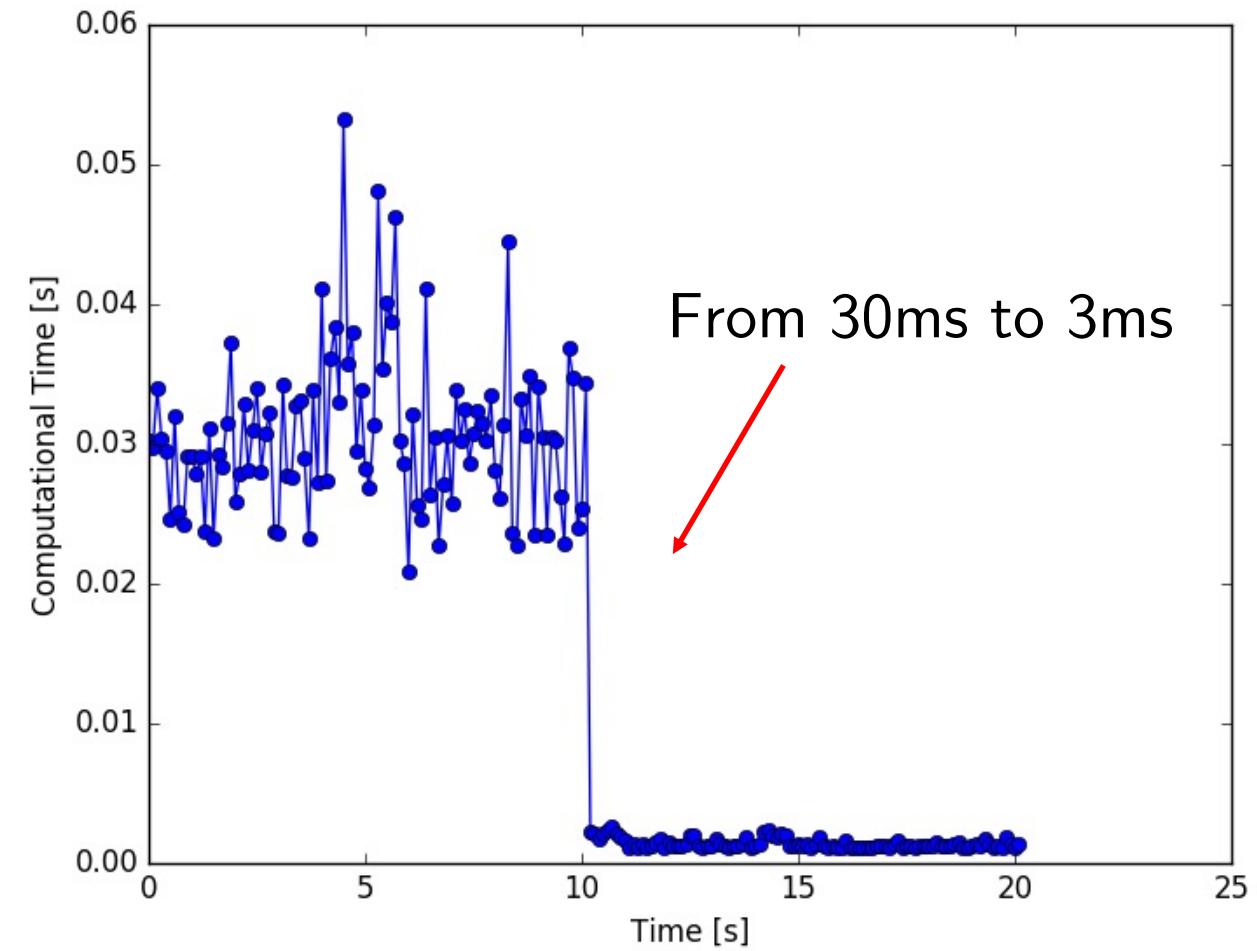
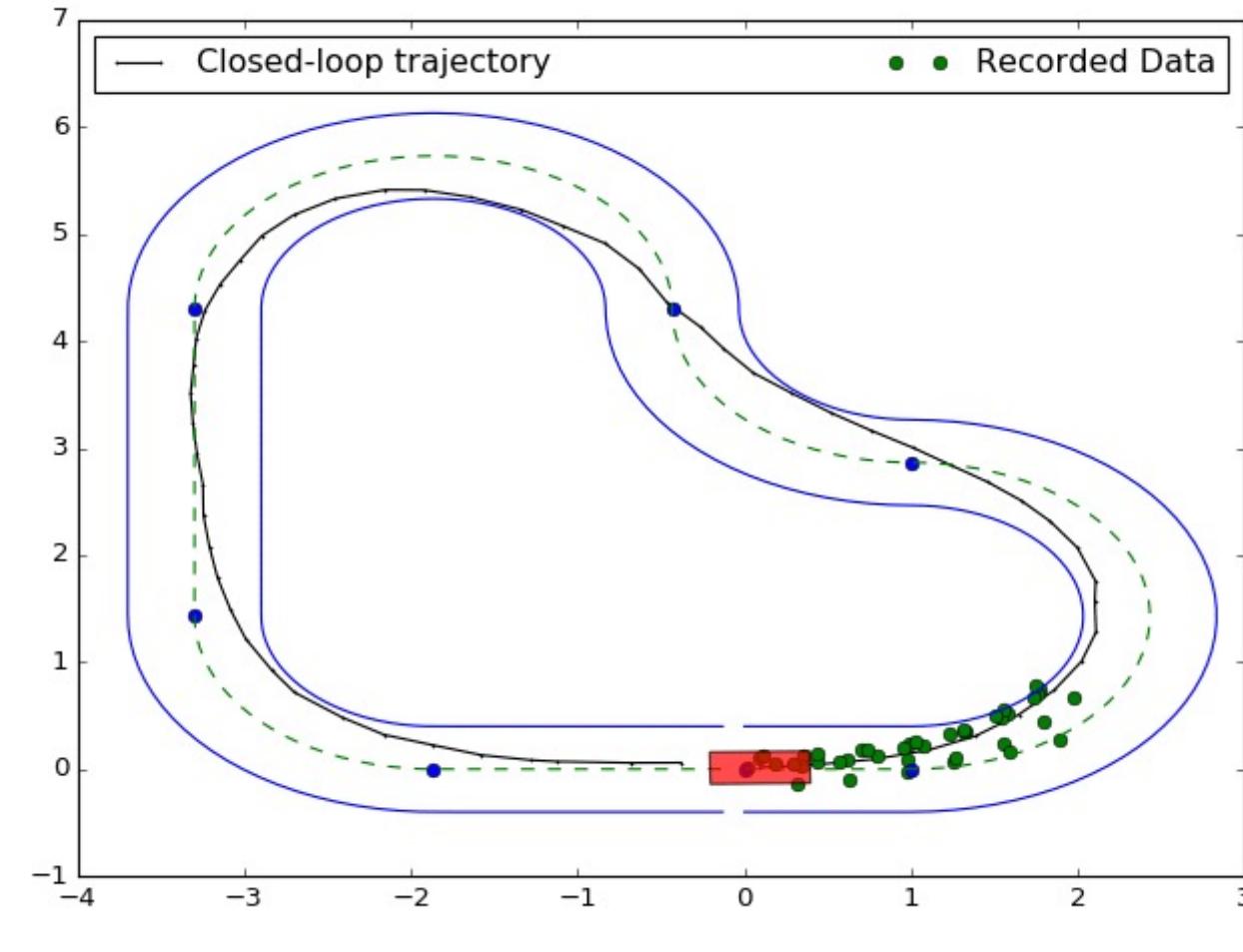
At each time t the data-based policy is computed with the K nearest neighbors (in green) to the current state. This allows us to account for the nonlinearity of the vehicle

Berkeley Autonomous Race Car Implementation



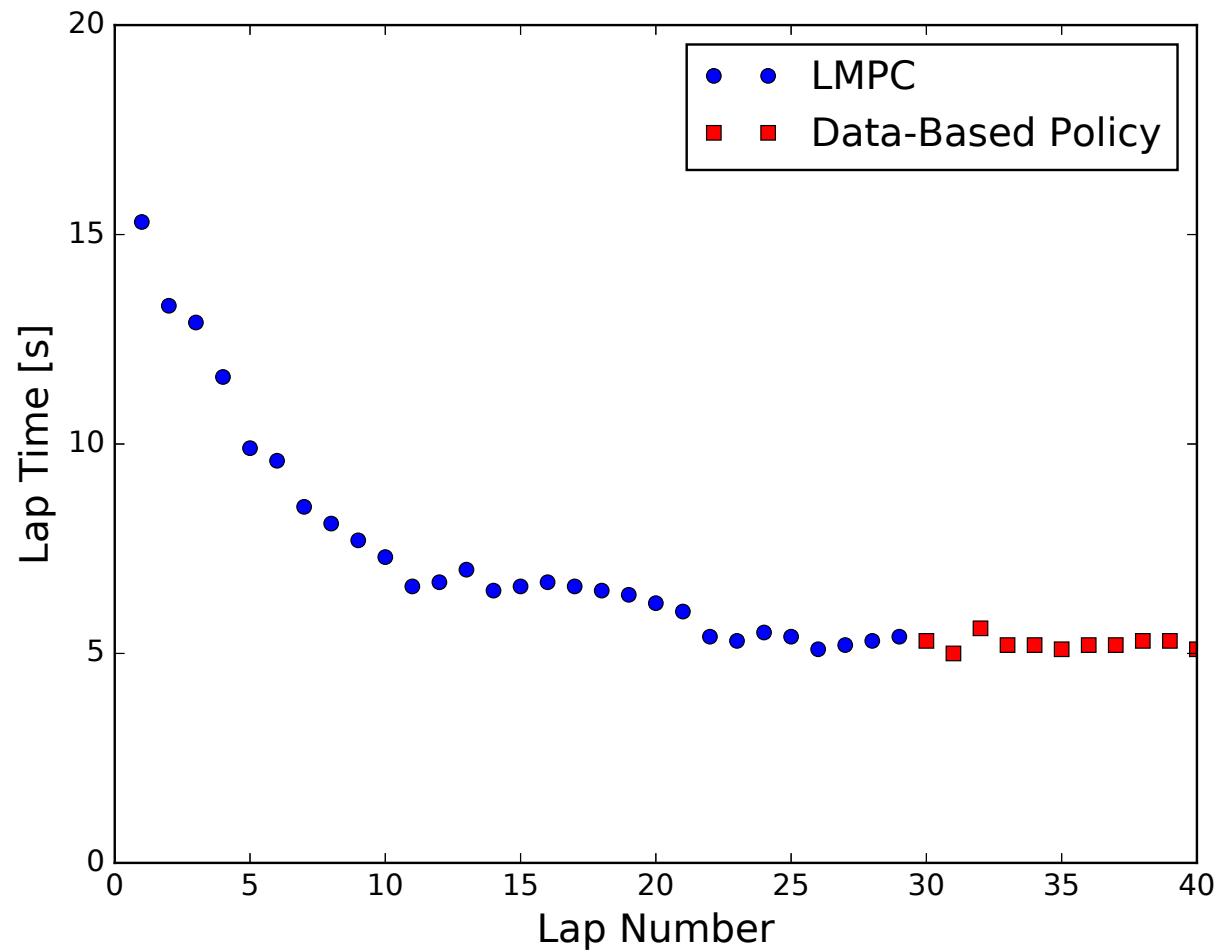
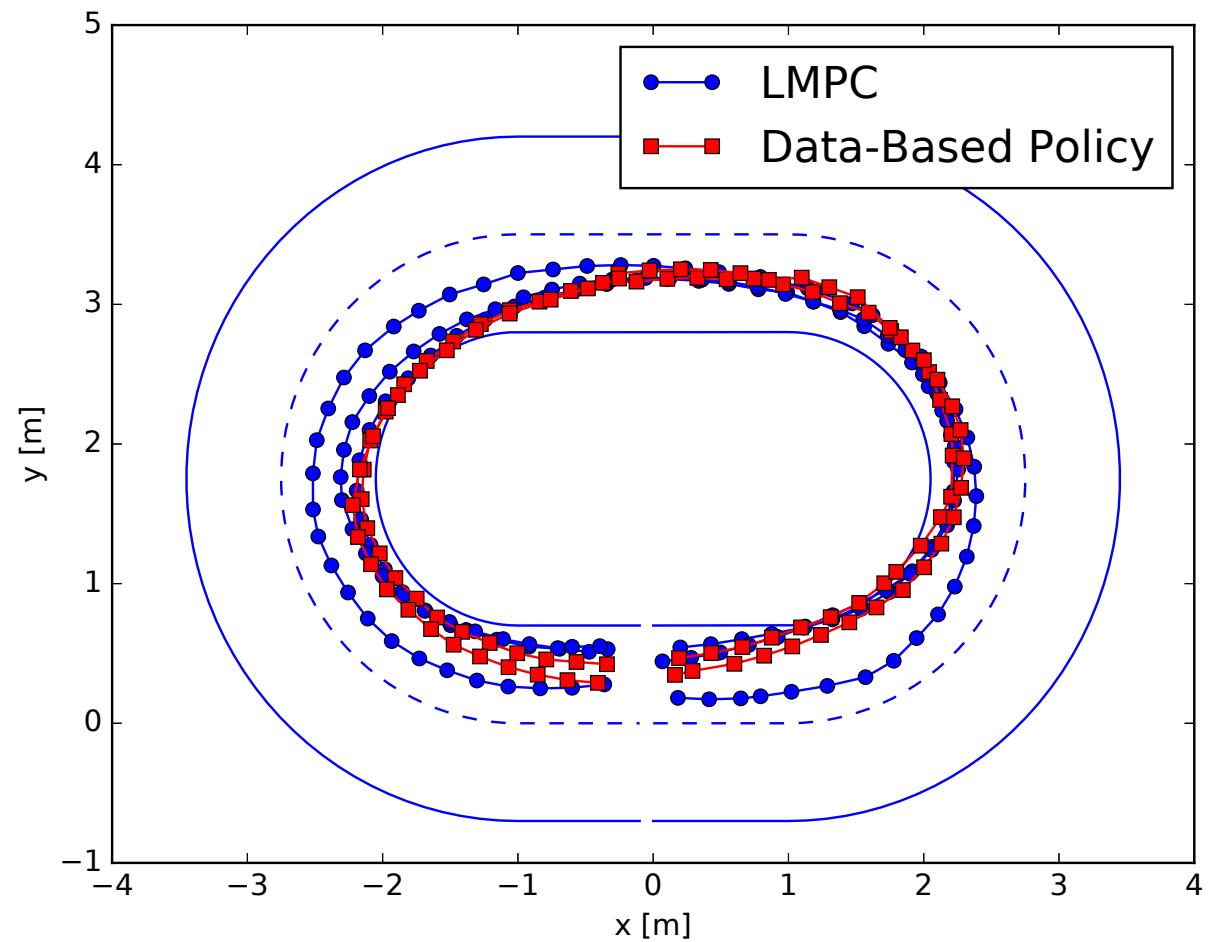
At each time t the data-based policy is computed with the K nearest neighbors (in green) to the current state. This allows us to account for the nonlinearity of the vehicle

Berkeley Autonomous Race Car Implementation

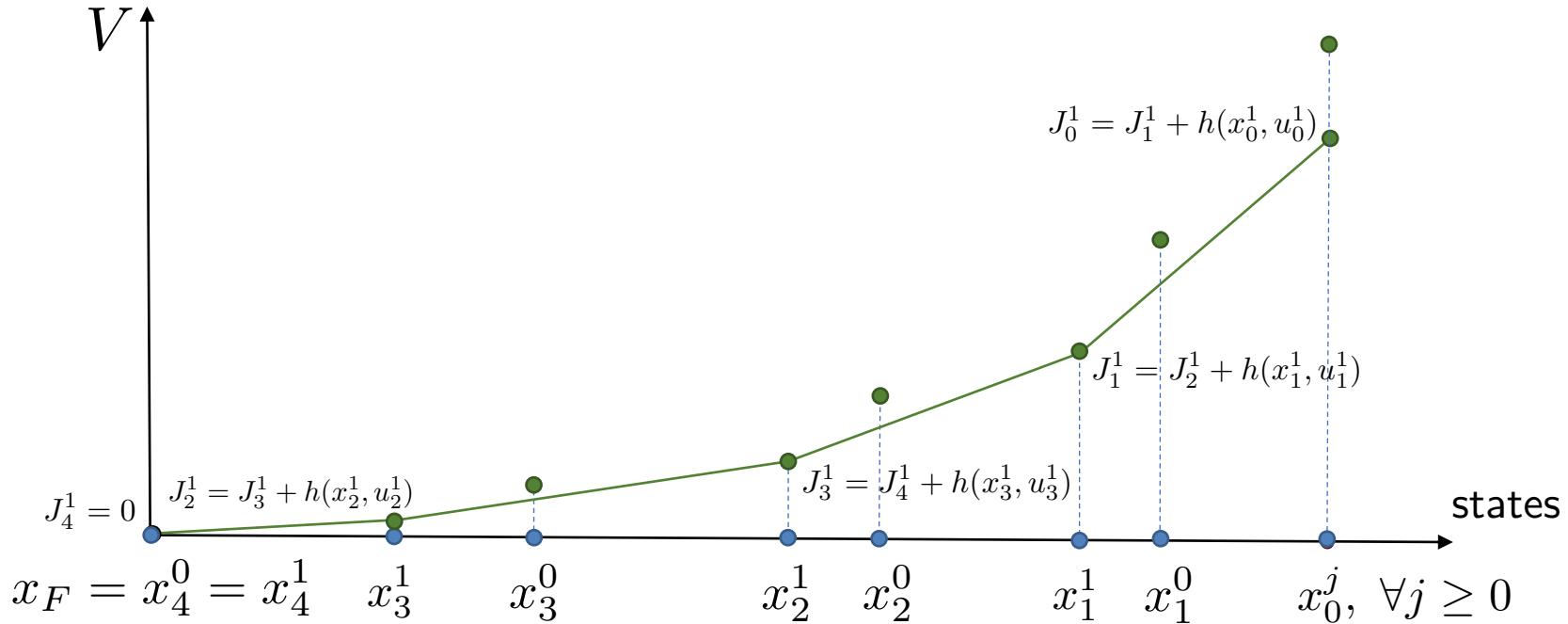


At each time t the data-based policy is computed with the K nearest neighbors (in green) to the current state. This allows us to account for the nonlinearity of the vehicle

Berkeley Autonomous Race Car Implementation



Data-Based Policy



Multipliers Defining the Value Function Approximation

$$[\lambda_0^{0,*}, \dots, \lambda_{T^j}^{T^j,*}] = \arg \min_{\lambda_i^j \in [0,1]} \sum_i \sum_j J_i^j \lambda_i^j$$

s.t.

$$\sum_i \sum_j x_i^j \lambda_i^j = \textcolor{red}{x},$$

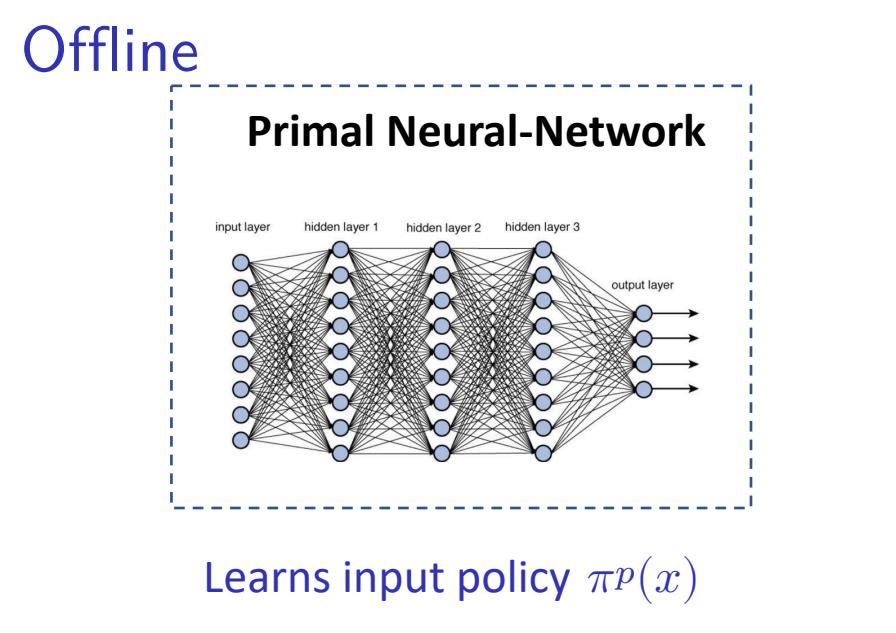
$$\sum_i \sum_j \lambda_i^j = 1$$

Data-Based Policy

$$\kappa^j(\textcolor{red}{x}) = \sum_i \sum_j u_i^j \lambda_i^{j,*}$$

Historical Data

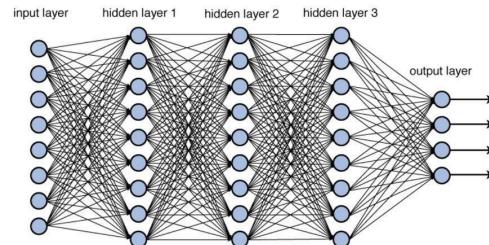
Speeding-up the solver



Speeding-up the solver

Offline

Primal Neural-Network



Learns input policy $\pi^p(x)$

Online

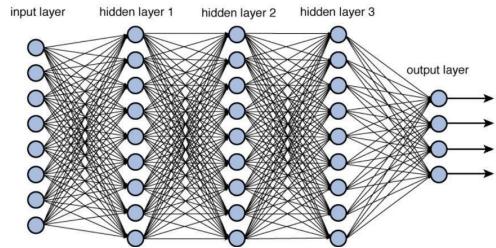
Apply the control policy $\pi^p(x)$

Primal-Dual Neural Network

Primal-Dual Neural Network

Offline

Primal Neural-Network

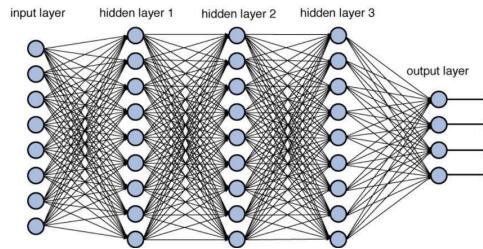


Learns input policy $\pi^p(x)$

Primal-Dual Neural Network

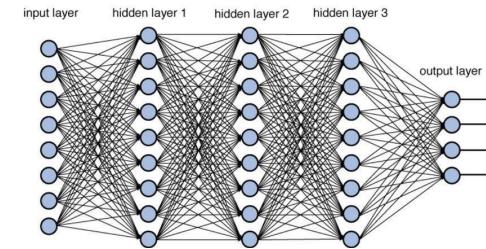
Offline

Primal Neural-Network



Learns input policy $\pi^p(x)$

Dual Neural-Network



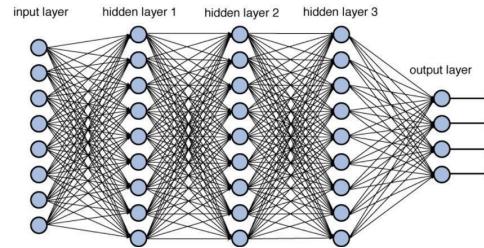
Learns optimality certificate $\lambda^d(x)$



Primal-Dual Neural Network

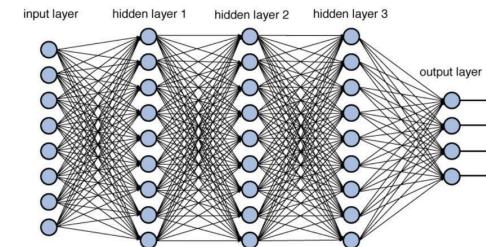
Offline

Primal Neural-Network



Learns input policy $\pi^p(x)$

Dual Neural-Network



Learns optimality certificate $\lambda^d(x)$



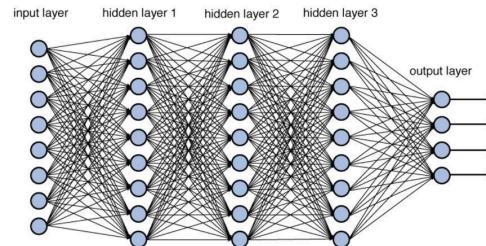
Offline Verification

Sample at random N states and check that $|J_N(\pi^p(x)) - J_N(\pi^p(x))| \leq \gamma$

Primal-Dual Neural Network

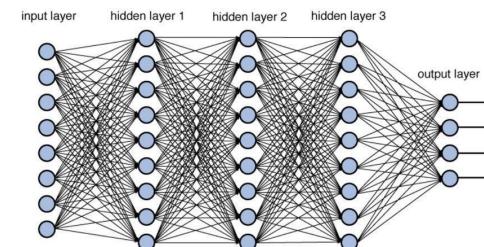
Offline

Primal Neural-Network



Learns input policy $\pi^p(x)$

Dual Neural-Network



Learns optimality certificate $\lambda^d(x)$



Offline Verification

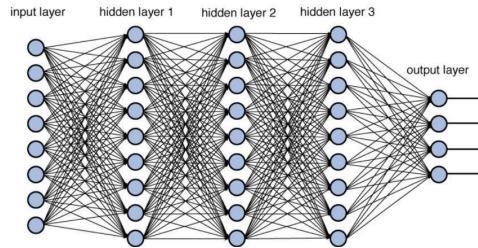
Sample at random N states and check that $|J_N(\pi^p(x)) - J_N^*| > \gamma/2$ w.p $\epsilon/2$ and $|J_N(\pi^p(x) - J_N^*)| > \gamma/2$ w.p $\epsilon/2$

If $N \geq \frac{\ln(\frac{1}{\beta/2})}{\ln(\frac{1}{1-\epsilon/2})}$ With probability at least $1 - \epsilon$, learned controllers will be at most γ -suboptimal

Primal-Dual Neural Network

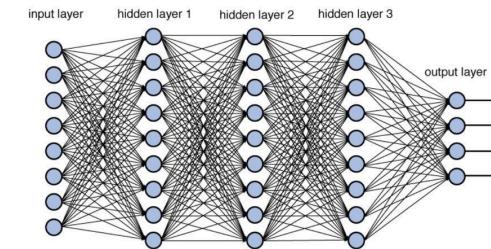
Offline

Primal Neural-Network



Learns input policy $\pi^p(x)$

Dual Neural-Network



Learns optimality certificate $\lambda^d(x)$

Online

LMPC for uncertain systems

A sample-based approach

LMPC for uncertain systems – Problem Formulation

LMPC for uncertain systems – Problem Formulation

- **Model Choice:** Linear system subject to bounded additive uncertainty,

$$x_{k+1} = Ax_k + Bu_k + w_k, \text{ with } w_k \in \mathcal{W} \quad \forall k \geq 0$$

LMPC for uncertain systems – Problem Formulation

- ▶ **Model Choice:** Linear system subject to bounded additive uncertainty,

$$x_{k+1} = Ax_k + Bu_k + w_k, \text{ with } w_k \in \mathcal{W} \quad \forall k \geq 0$$

- ▶ **Constraint Satisfaction:**

- **Robust,** $x_k \in \mathcal{X}, \forall w_k \in \mathcal{W} \quad k \geq 0.$

LMPC for uncertain systems – Problem Formulation

- ▶ **Model Choice:** Linear system subject to bounded additive uncertainty,

$$x_{k+1} = Ax_k + Bu_k + w_k, \text{ with } w_k \in \mathcal{W} \quad \forall k \geq 0$$

- ▶ **Constraint Satisfaction:**

- **Robust,** $x_k \in \mathcal{X}, \forall w_k \in \mathcal{W} \quad k \geq 0.$

- ▶ **Cost:** Worst Case.

LMPC for uncertain systems – Problem Formulation

- ▶ **Model Choice:** Linear system subject to bounded additive uncertainty,

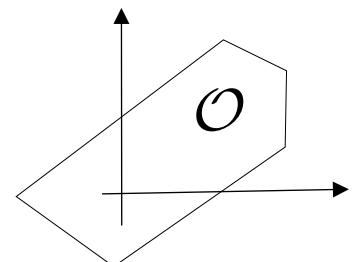
$$x_{k+1} = Ax_k + Bu_k + w_k, \text{ with } w_k \in \mathcal{W} \quad \forall k \geq 0$$

- ▶ **Constraint Satisfaction:**

- **Robust,** $x_k \in \mathcal{X}, \forall w_k \in \mathcal{W} \quad k \geq 0.$

- ▶ **Cost:** Worst Case.

- ▶ **Convergence:** Cannot regulate to a point but to a robust positive invariant . \mathcal{O}

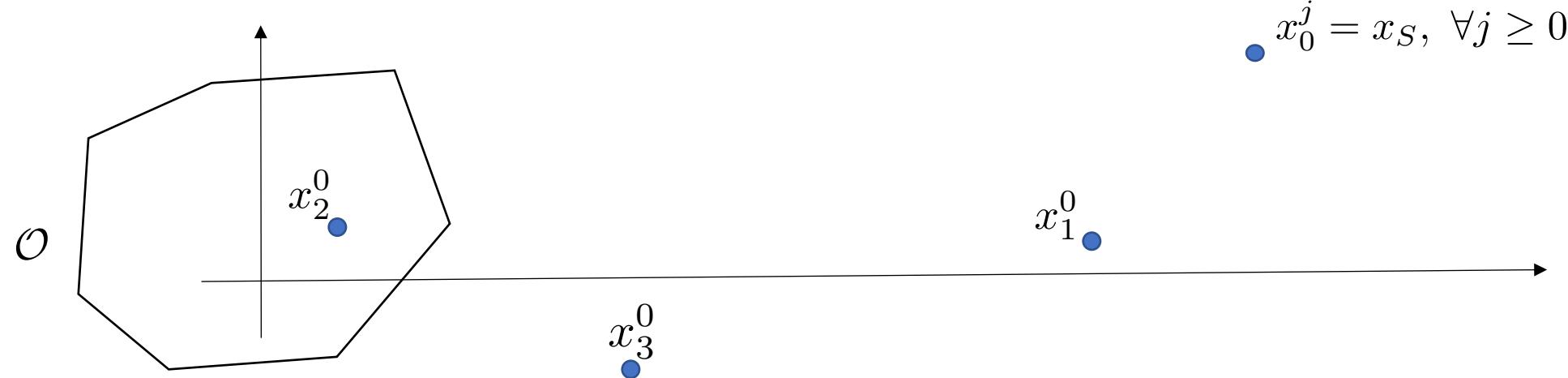


$$\forall x \in \mathcal{O} \rightarrow (A - KB)x + w \in \mathcal{O}, \quad \forall w \in \mathcal{W}$$

LMPC for uncertain systems – Differences w/ Nominal

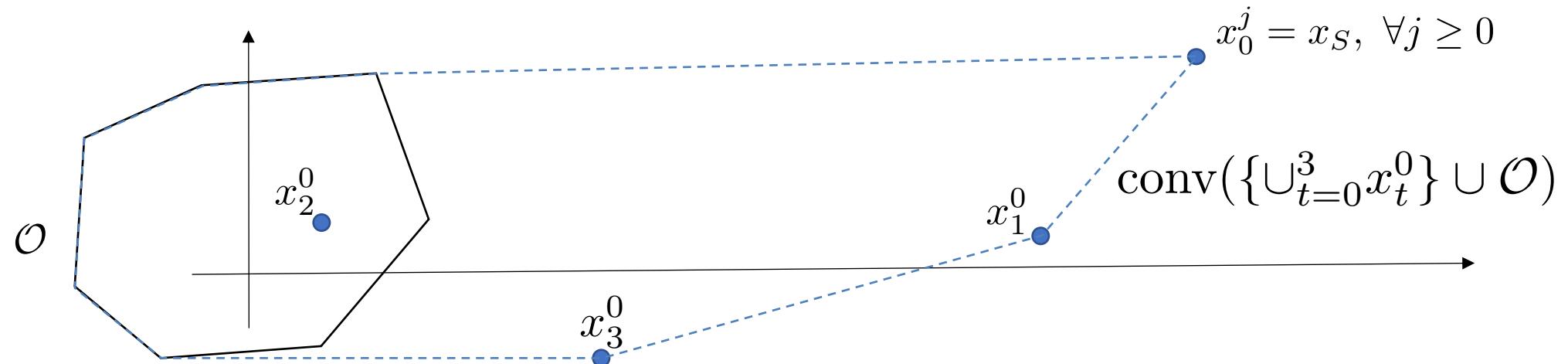
LMPC for uncertain systems – Differences w/ Nominal

Given a feasible trajectory which drives the system to the terminal robust invariant



LMPC for uncertain systems – Differences w/ Nominal

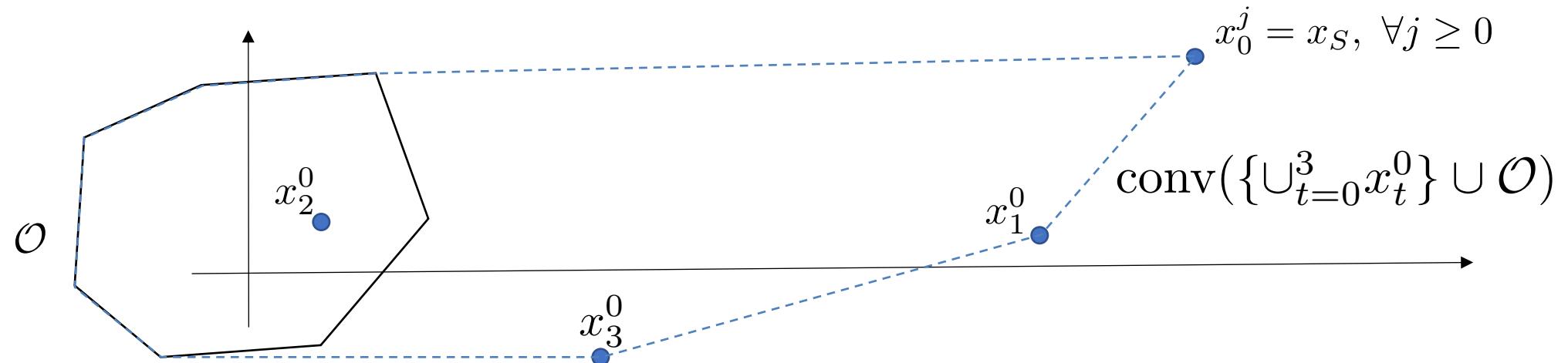
Given a feasible trajectory which drives the system to the terminal robust invariant



The convex hull of the stored data is **not a robust control invariant**.

LMPC for uncertain systems – Differences w/ Nominal

Given a feasible trajectory which drives the system to the terminal robust invariant



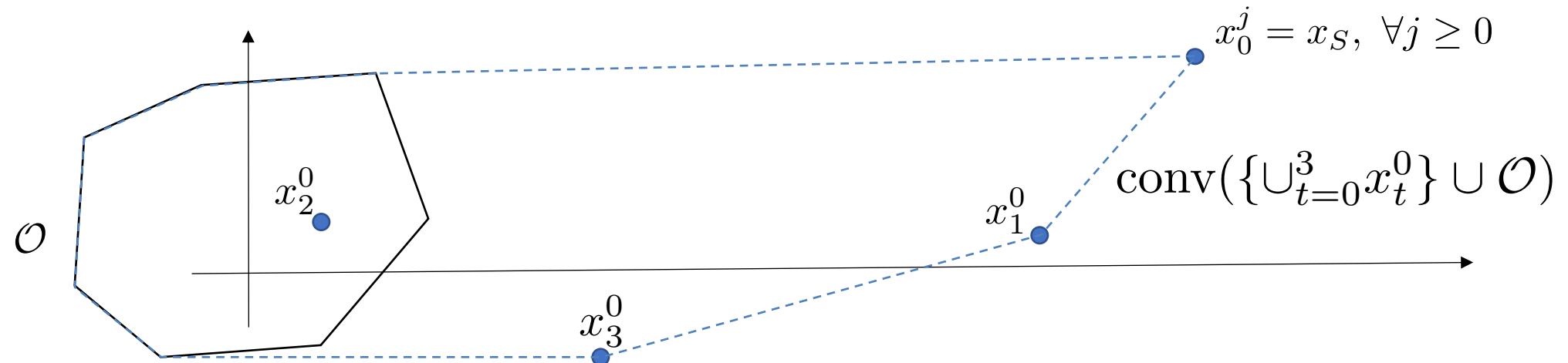
The convex hull of the stored data is **not a robust control invariant**.

For this reason,

- A. the Safe Set

LMPC for uncertain systems – Differences w/ Nominal

Given a feasible trajectory which drives the system to the terminal robust invariant



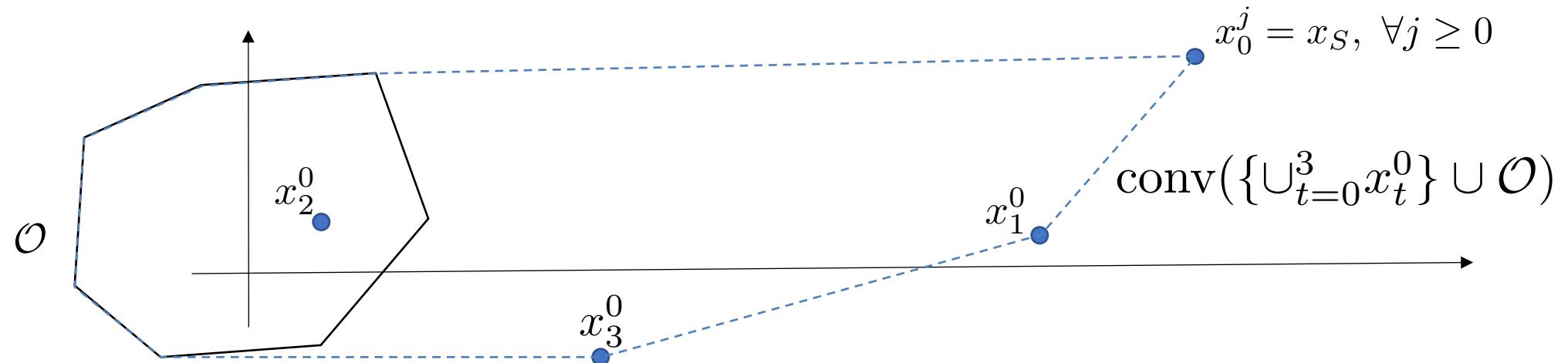
The convex hull of the stored data is **not a robust control invariant**.

For this reason,

- A. the Safe Set
- B. the Q-function

LMPC for uncertain systems – Differences w/ Nominal

Given a feasible trajectory which drives the system to the terminal robust invariant



The convex hull of the stored data is **not a robust control invariant**.

For this reason,

- A. the Safe Set
- B. the Q-function

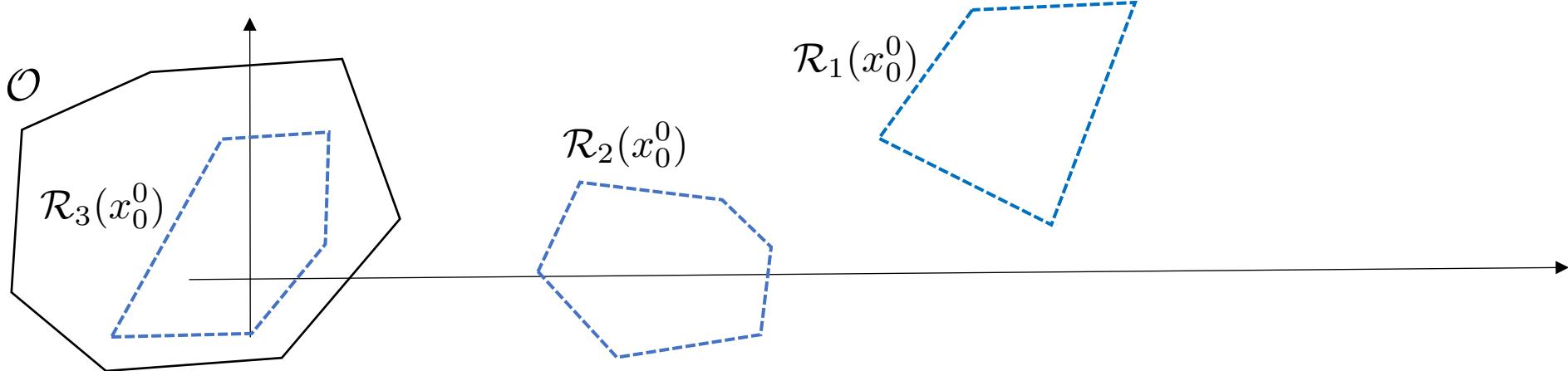
need to be designed differently in Robust LMPC.

A. Robust Safe Set

A. Robust Safe Set

Assume that given control policy $\pi^0 : \mathbb{R}^n \rightarrow \mathbb{R}^d$ robustly steers, the closed-loop system $x_{k+1}^0 = Ax_k^0 + B\pi^0(x_k^0) + w_k^0$ converges to the terminal set \mathcal{O} .

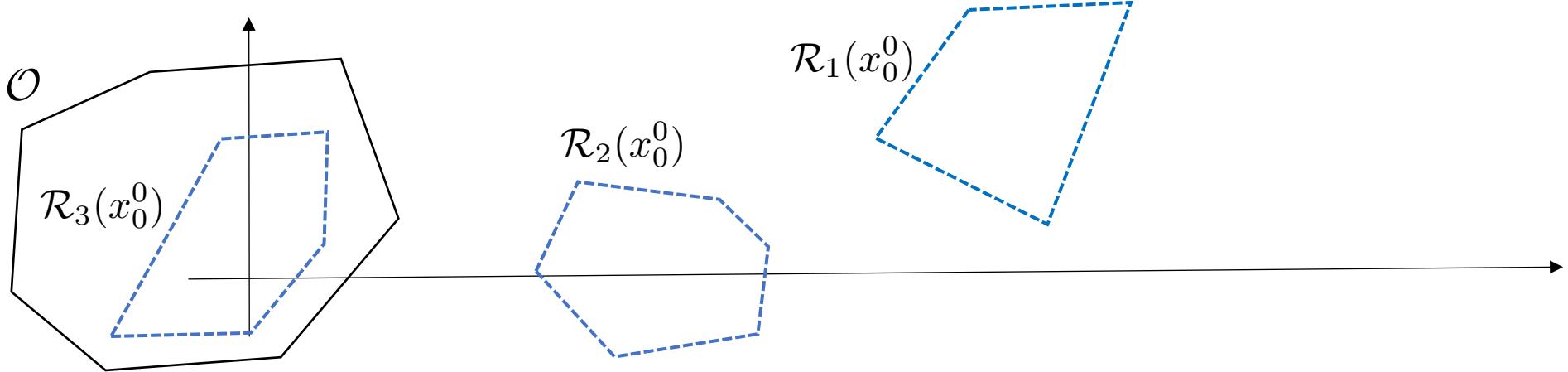
$$\bullet \quad x_0^j = x_S, \quad \forall j \geq 0$$



A. Robust Safe Set

Assume that given control policy $\pi^0 : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \rightarrow \mathbb{R}^{n_x}$ robustly steers, the closed-loop system converges to the terminal set $x_k^0 + w_k^0$

$$x_0^j = x_S, \forall j \geq 0$$

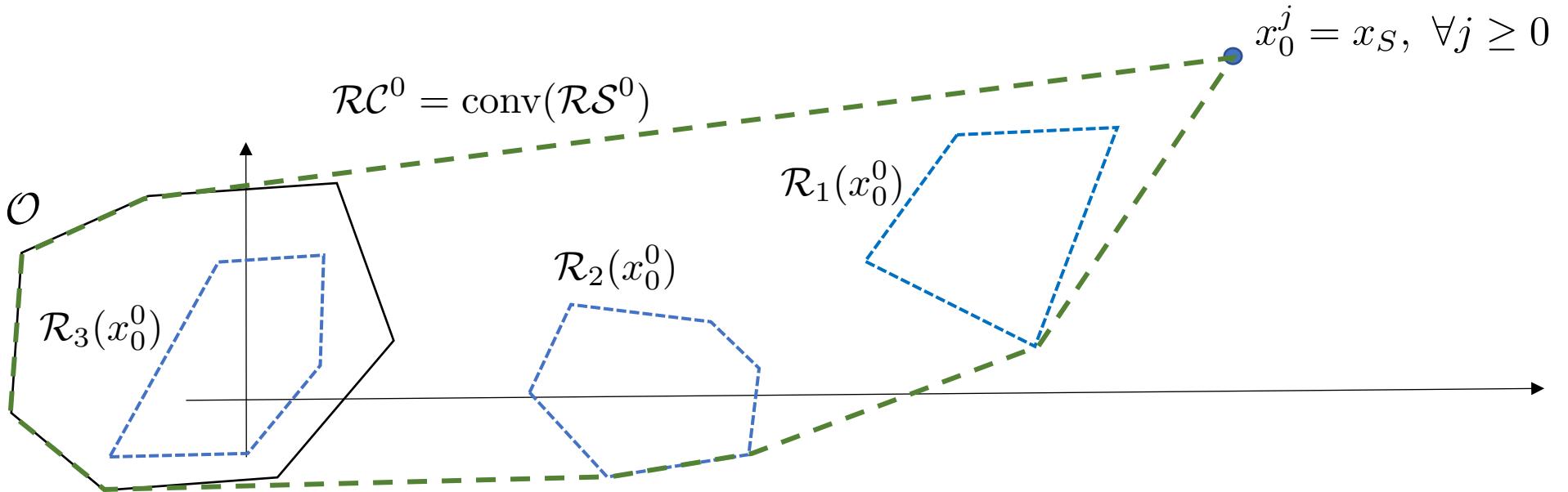


Definition: Robust Safe Set

The robust safe set $\mathcal{RS}^j = \{\cup_k \mathcal{R}_k(x_0^j)\}$ is a robust control invariant set for constrained uncertain linear systems.

A. Robust Convex Safe Set

Assume that given control policy $\pi^0 : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \rightarrow \mathbb{R}^{n_u}$ robustly steers, the closed-loop system converges to the terminal set $x_k^0 + w_k^0$



Definition: Robust Convex Safe Set

The robust convex safe set $\mathcal{RC}^j = \text{conv}(\mathcal{RS}^j)$ is a **robust control invariant** set for constrained uncertain linear systems.

B. Robust Q-Function

B. Robust Q-Function

Consider the Bellman recursion for the control policy , π^j

$$J_{\pi^j}^j(x) = \max_{w \in \mathcal{W}} [h(x, \pi^j(x)) + J_{\pi^j}^j(Ax + B\pi^j(x) + w)]$$

B. Robust Q-Function

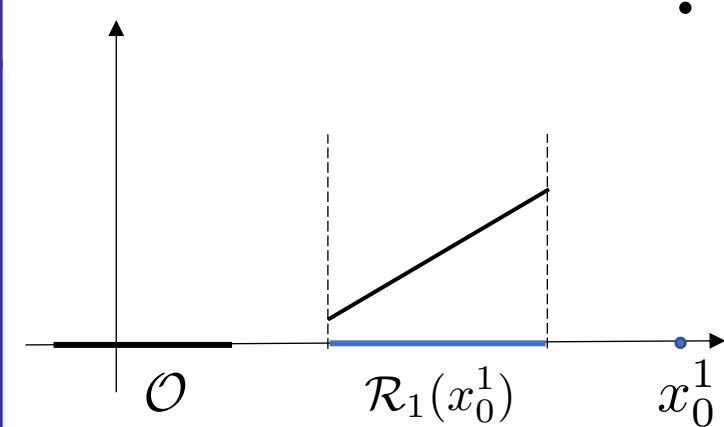
Consider the Bellman recursion for the control policy π^j ,

$$J_{\pi^j}^j(x) = \max_{w \in \mathcal{W}} [h(x, \pi^j(x)) + J_{\pi^j}^j(Ax + B\pi^j(x) + w)]$$

Definition: Robust Q-function

The robust Q-function, which represents the worst case cost-to-go over the robust safe set, is defined as

$$Q_r^j(x) = \begin{cases} \max_{w \in \mathcal{W}} [h(x, \pi^j(x)) + Q_r^j(Ax^j + Bu^j + w)] & \text{If } x \in \mathcal{RS}^j \\ +\infty & \text{If } x \notin \mathcal{RS}^j \end{cases}$$



B. Robust Q-Function

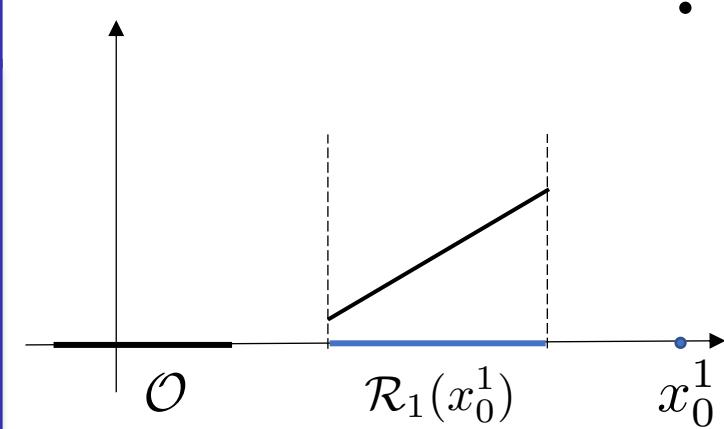
Consider the Bellman recursion for the control policy π^j ,

$$J_{\pi^j}^j(x) = \max_{w \in \mathcal{W}} [h(x, \pi^j(x)) + J_{\pi^j}^j(Ax + B\pi^j(x) + w)]$$

Definition: Robust Q-function

The robust Q-function, which represents the worst case cost-to-go over the robust safe set, is defined as

$$Q_r^j(x) = \begin{cases} \max_{w \in \mathcal{W}} [h(x, \pi^j(x)) + Q_r^j(Ax^j + Bu^j + w)] & \text{If } x \in \mathcal{RS}^j \\ +\infty & \text{If } x \notin \mathcal{RS}^j \end{cases}$$

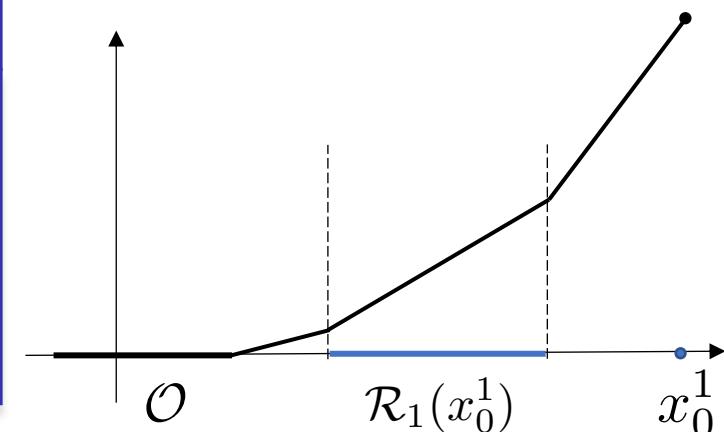


Definition: Robust Convex Q-function

The robust convex Q-function defined as

$$Q_{rc}^j(x) = \min_{\mu} \mu \mid (x, \mu) \in \text{conv}\left(\bigcup_{k=0}^j \text{epi}(Q_r^j(x))\right),$$

is a **robust control Lyapunov function**.



Robust Learning Model Predictive Control

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

$$\begin{aligned} J_{t \rightarrow t+N}^{\text{LMPC},j}(x_t^j) &= \min_{\pi_t^j(\cdot)} \max_{\bar{\mathbf{w}}_t^j} \left[\sum_{k=t}^{t+N-1} h(x_{k|t}^j, u_{k|t}^j) + Q_{rc}^{j-1}(x_{t+N|t}^j) \right] \\ x_{k+1|t}^j &= Ax_{k|t}^j + Bu_{k|t}^j + \bar{w}_{k|t}^j \\ u_{k|t}^j &= \pi_{k|t}^j(x_{k|t}^j) \\ x_{k|t}^j &\in \mathcal{X}, u_{k|t}^j \in \mathcal{U} \\ x_{t+N|t}^j &\in \mathcal{RC}^{j-1}, \quad \forall \bar{w}_{k|t}^j \in \mathcal{W}, k \in \{t, \dots, t+N\} \end{aligned}$$

for $\pi_t^j(\cdot) = [\pi_{t|t}^j(\cdot), \dots, \pi_{t+N|t}^j(\cdot)]$ $\bar{\mathbf{w}}_t^j = [\bar{w}_{t|t}^j, \dots, \bar{w}_{t+N|t}^j]$

Then apply to the system $u_t^j = \pi_{t|t}^{j,*}(x_t^j)$

Robust Learning Model Predictive Control

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

$$J_{t \rightarrow t+N}^{\text{LMPC},j}(x_t^j) = \min_{\pi_t^j(\cdot)} \max_{\bar{\mathbf{w}}_t^j} \left[\sum_{k=t}^{t+N-1} h(x_{k|t}^j, u_{k|t}^j) + Q_{rc}^{j-1}(x_{t+N|t}^j) \right]$$

$$x_{k+1|t}^j = Ax_{k|t}^j + Bu_{k|t}^j + \bar{w}_{k|t}^j$$

$$u_{k|t}^j = \pi_{k|t}^j(x_{k|t}^j)$$

$$x_{k|t}^j \in \mathcal{X}, u_{k|t}^j \in \mathcal{U}$$

$$x_{t+N|t}^j \in \mathcal{RC}^{j-1}, \quad \forall \bar{w}_{k|t}^j \in \mathcal{W}, k \in \{t, \dots, t+N\}$$

Iteratively
constructed using
the system model

for $\pi_t^j(\cdot) = [\pi_{t|t}^j(\cdot), \dots, \pi_{t+N|t}^j(\cdot)]$ $\bar{\mathbf{w}}_t^j = [\bar{w}_{t|t}^j, \dots, \bar{w}_{t+N|t}^j]$

Then apply to the system $u_t^j = \pi_{t|t}^{j,*}(x_t^j)$

Robust Learning Model Predictive Control – Remarks

At iteration j the robust convex safe set and the robust Q-function $Q_{rc}^j(\cdot)$ can be:

- ▶ Computed analytically (hard even for linear systems)

Robust Learning Model Predictive Control – Remarks

At iteration j the robust convex safe set and the robust Q-function $Q_{rc}^j(\cdot)$ can be:

- ▶ Computed analytically (hard even for linear systems)
- ▶ Computed propagating the vertices of the disturbance \mathcal{W} (explodes!)

Robust Learning Model Predictive Control – Remarks

At iteration j the robust convex safe set and the robust Q-function $Q_{rc}^j(\cdot)$ can be:

- ▶ Computed analytically (hard even for linear systems)
- ▶ Computed propagating the vertices of the disturbance w (explodes!)
- ▶ Approximated for a large number of rollout . Then, we could guarantee

$$\mathbb{P}(Ax + Bu + w \in \mathcal{CS}^j | x \in \mathcal{CS}^j) \geq 1 - \epsilon(N_{roll})$$

$$\mathbb{P}(Q_{rc}^j(Ax + B\pi^j(u) + w) - Q_{rc}^j(x) + h(x, \pi^j(u)) \leq 0 | x \in \mathcal{RS}^j) \geq 1 - \gamma(N_{roll})$$

where $\epsilon(N_{roll})$ and $\gamma(N_{roll})$ increase as the number of rollouts increases N_{roll}

Robust Learning Model Predictive Control – Remarks

At iteration j the robust convex safe set and the robust Q-function $Q_{rc}^j(\cdot)$ can be:

- ▶ Computed analytically (hard even for linear systems)
- ▶ Computed propagating the vertices of the disturbance w (explodes!)
- ▶ Approximated for a large number of rollout . Then, we could guarantee

$$\mathbb{P}(Ax + Bu + w \in \mathcal{CS}^j | x \in \mathcal{CS}^j) \geq 1 - \epsilon(N_{roll})$$

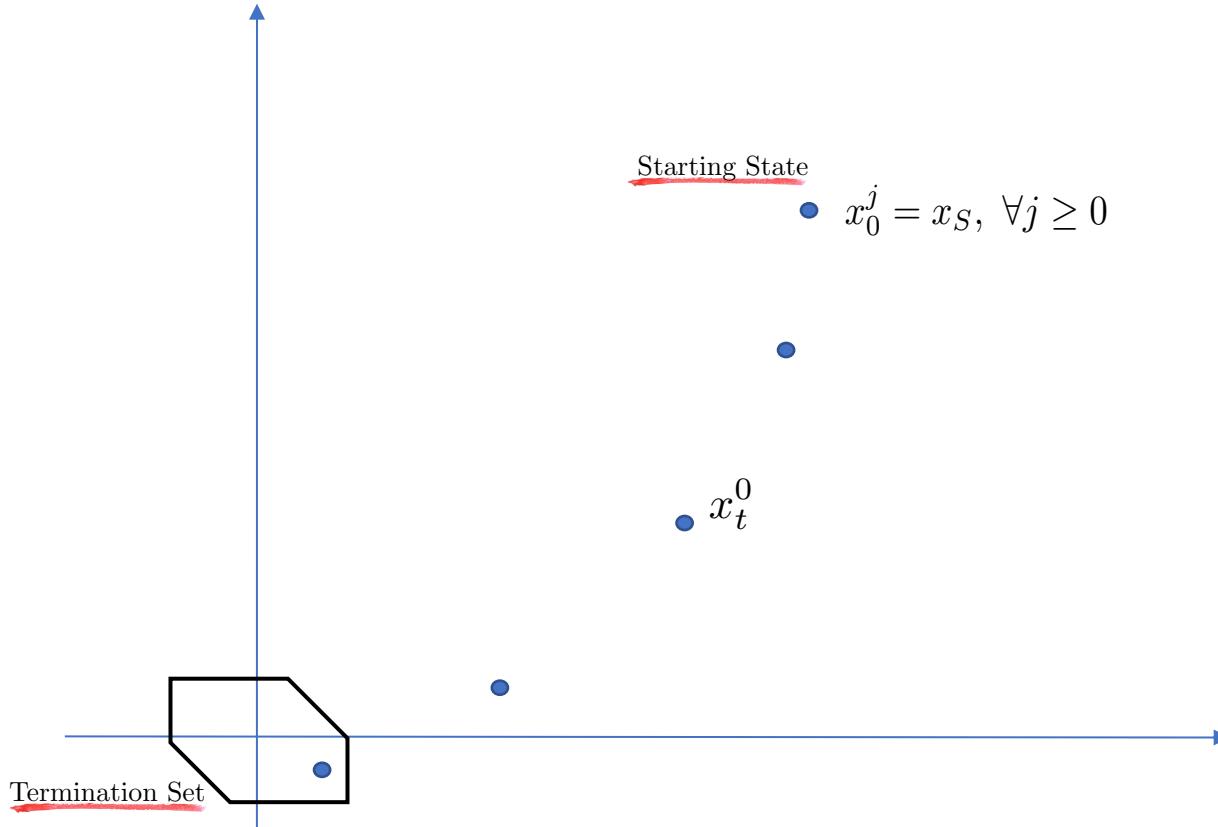
$$\mathbb{P}(Q_{rc}^j(Ax + B\pi^j(u) + w) - Q_{rc}^j(x) + h(x, \pi^j(u)) \leq 0 | x \in \mathcal{RS}^j) \geq 1 - \gamma(N_{roll})$$

where $\epsilon(N_{roll})$ and $\gamma(N_{roll})$ increase as the number of rollouts increases N_{roll}



Iterative approximation strategy
based on historical data

Example: Constrained LQR Uncertain System



Control Problem

Let

$$J_{\pi^j}^j(x) = \max_{w \in \mathcal{W}} [h(x, \pi^j(x)) + J_{\pi^j}^j(Ax + B\pi^j(x) + w)]$$

be the solution to the Bellman recursion for the control policy π^j . We want to solve the following finite time robust optimal control problem

$$J_{0 \rightarrow T^j}^{j,*}(x_0^j) = \min_{\pi^j(\cdot)} J_{\pi^j}^j(x_0^j)$$

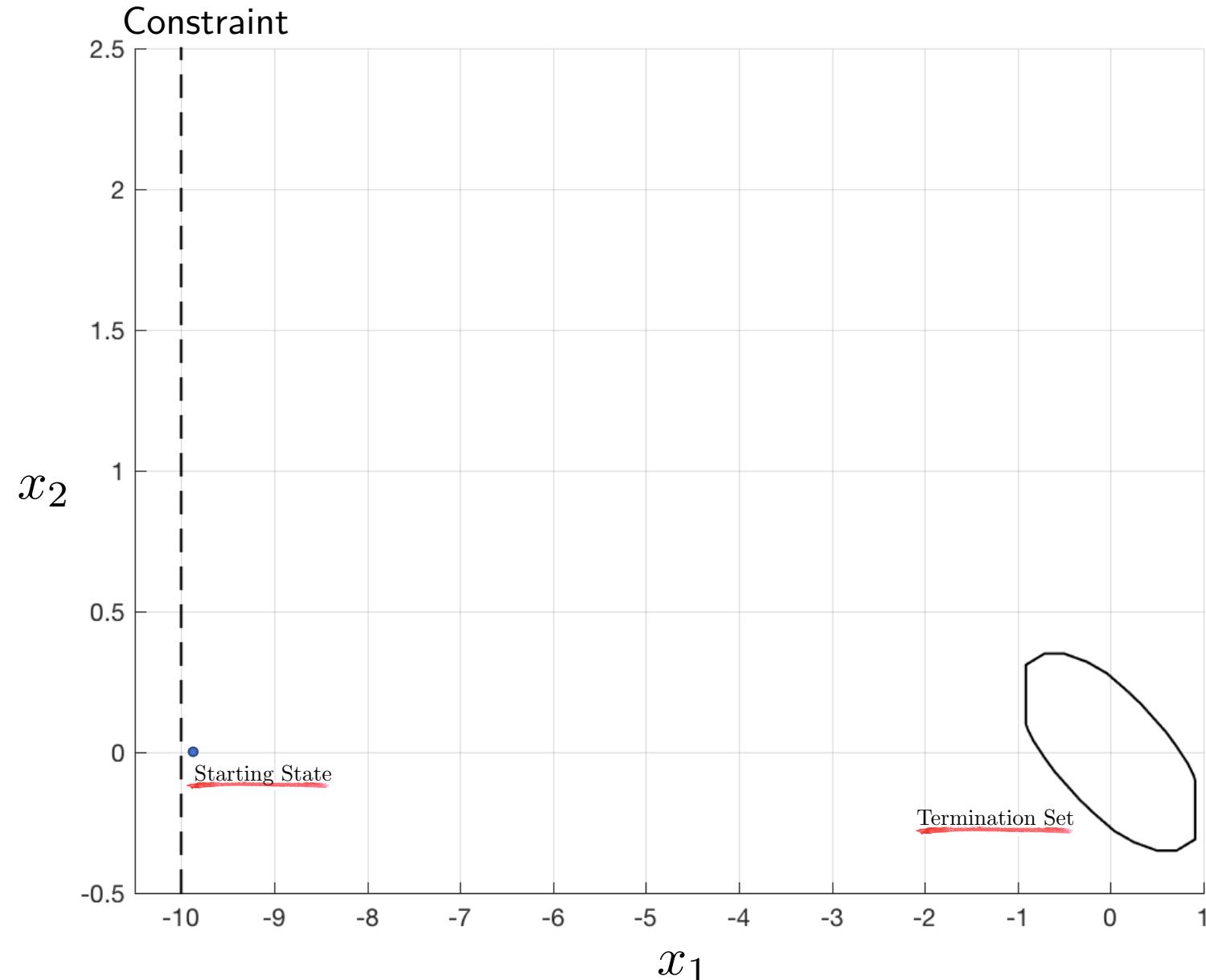
$$x_{k+1}^j = Ax_k^j + B\pi^j(x_k^j) + w_k^j$$

$$u_k^j = \pi^j(x_k^j)$$

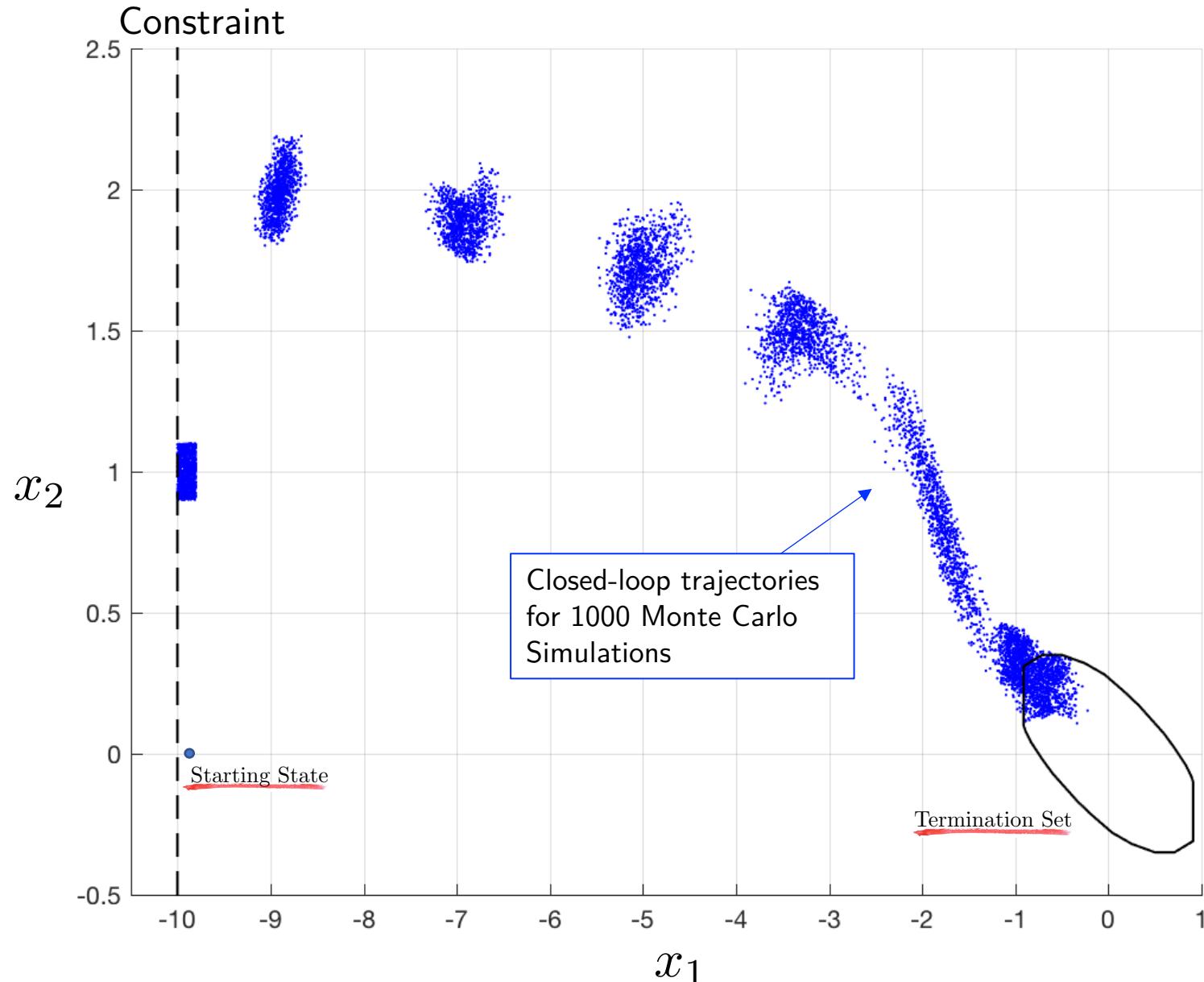
$$x_k^j \in \mathcal{X}, u_k^j \in \mathcal{U}, x_{T^j}^j \in \mathcal{O}$$

$$\forall w_k^j \in \mathcal{W}, k \in \{0, \dots, T^j\}$$

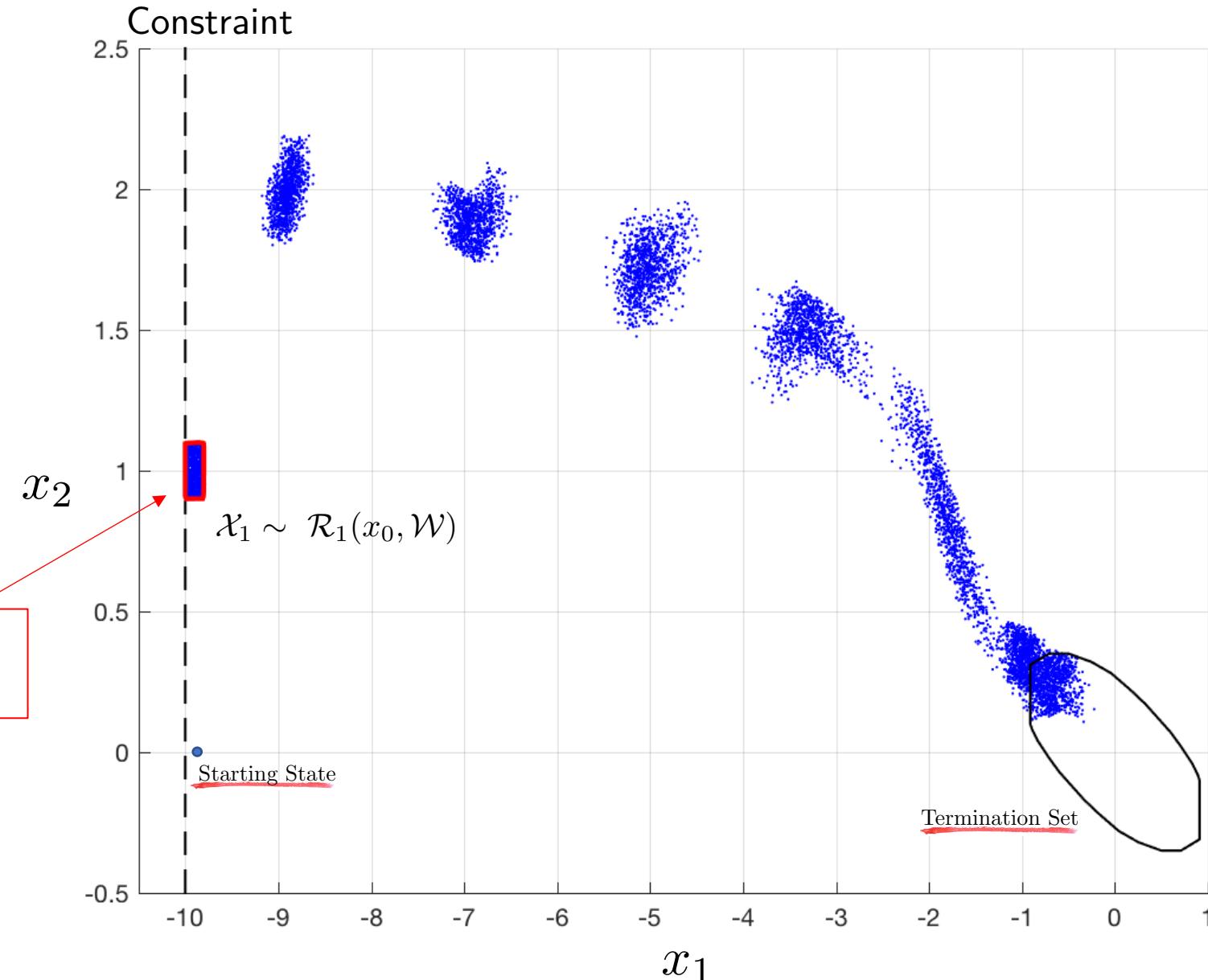
Example: Constrained LQR Uncertain System



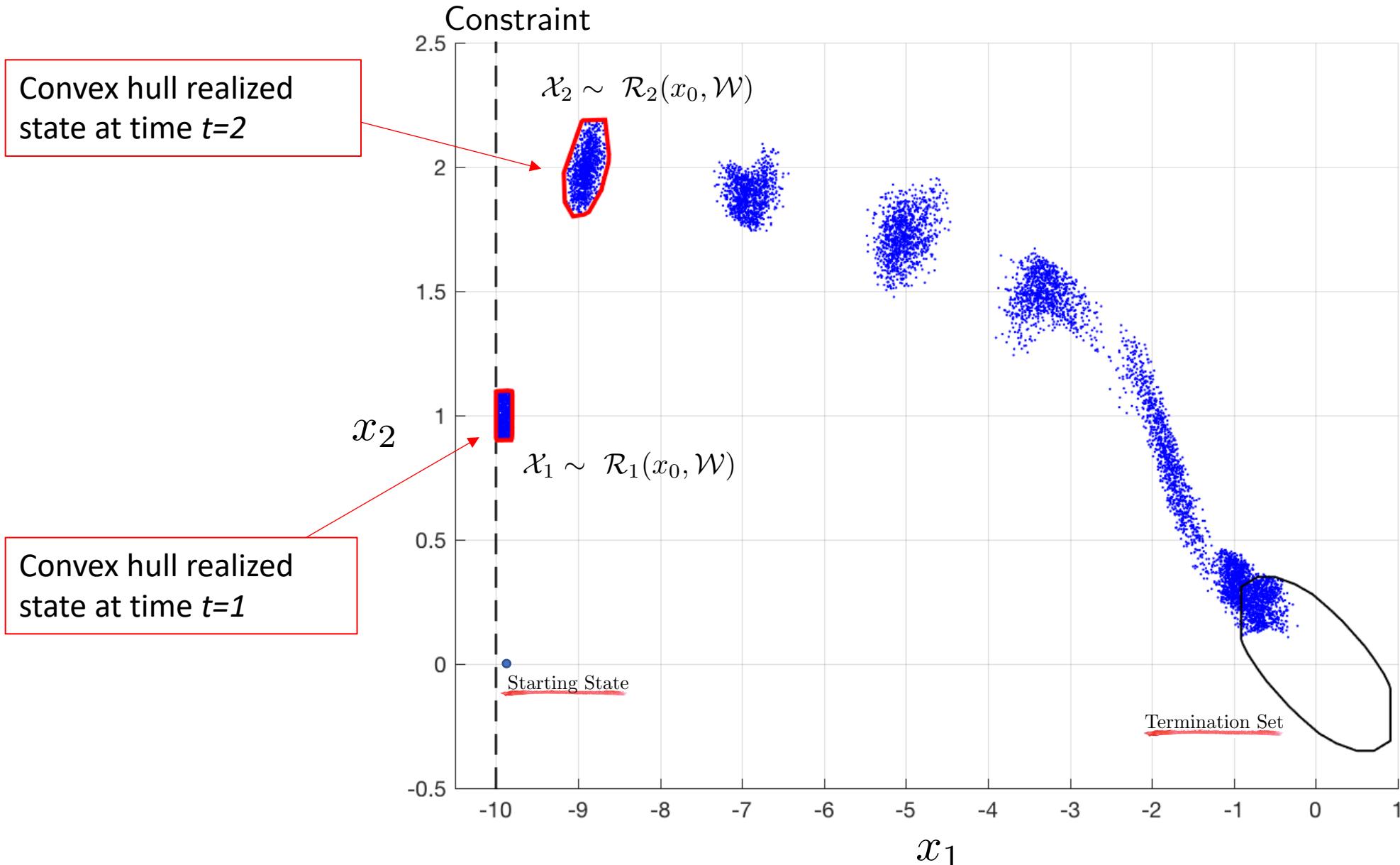
Example: Constrained LQR Uncertain System



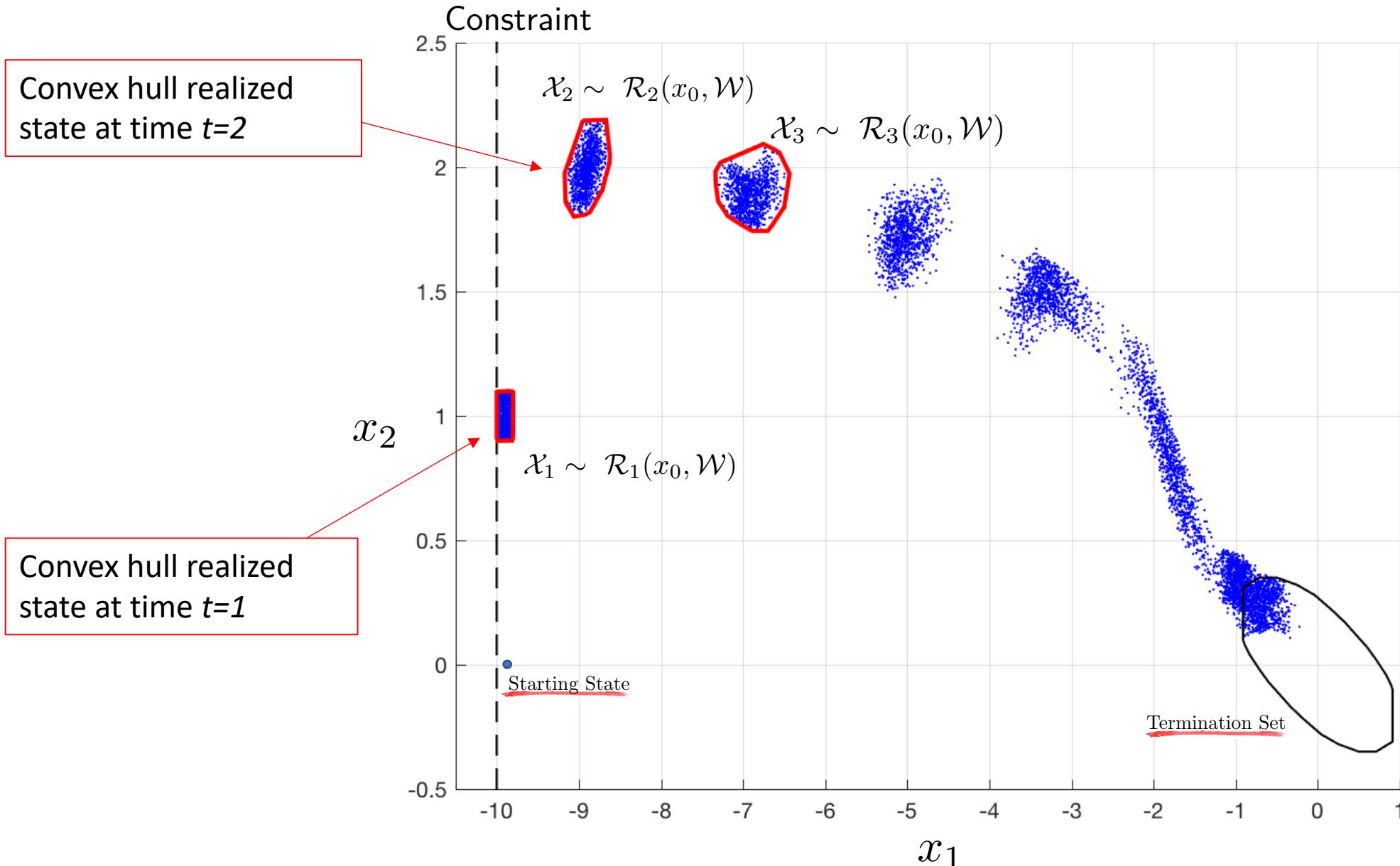
Example: Constrained LQR Uncertain System



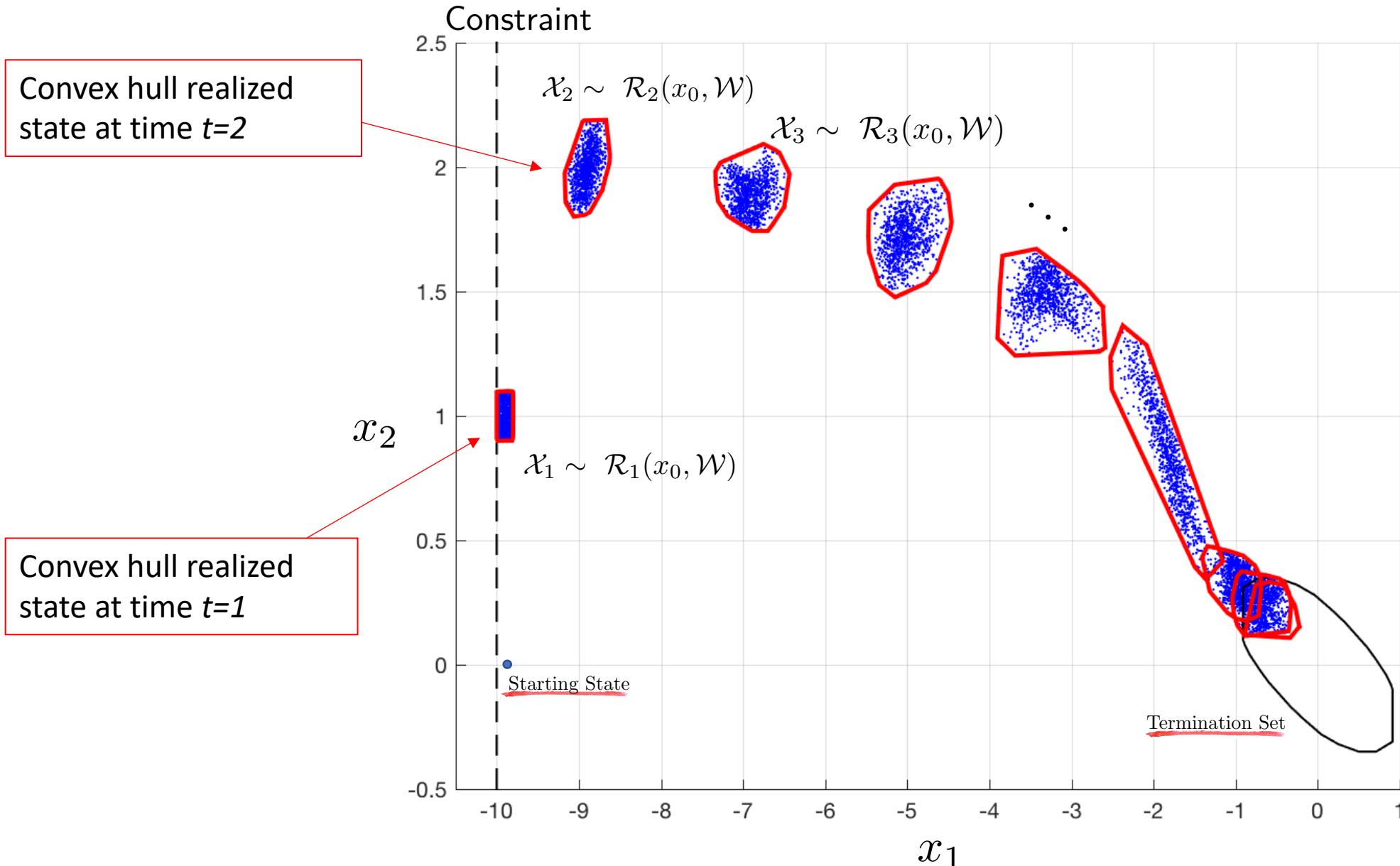
Example: Constrained LQR Uncertain System



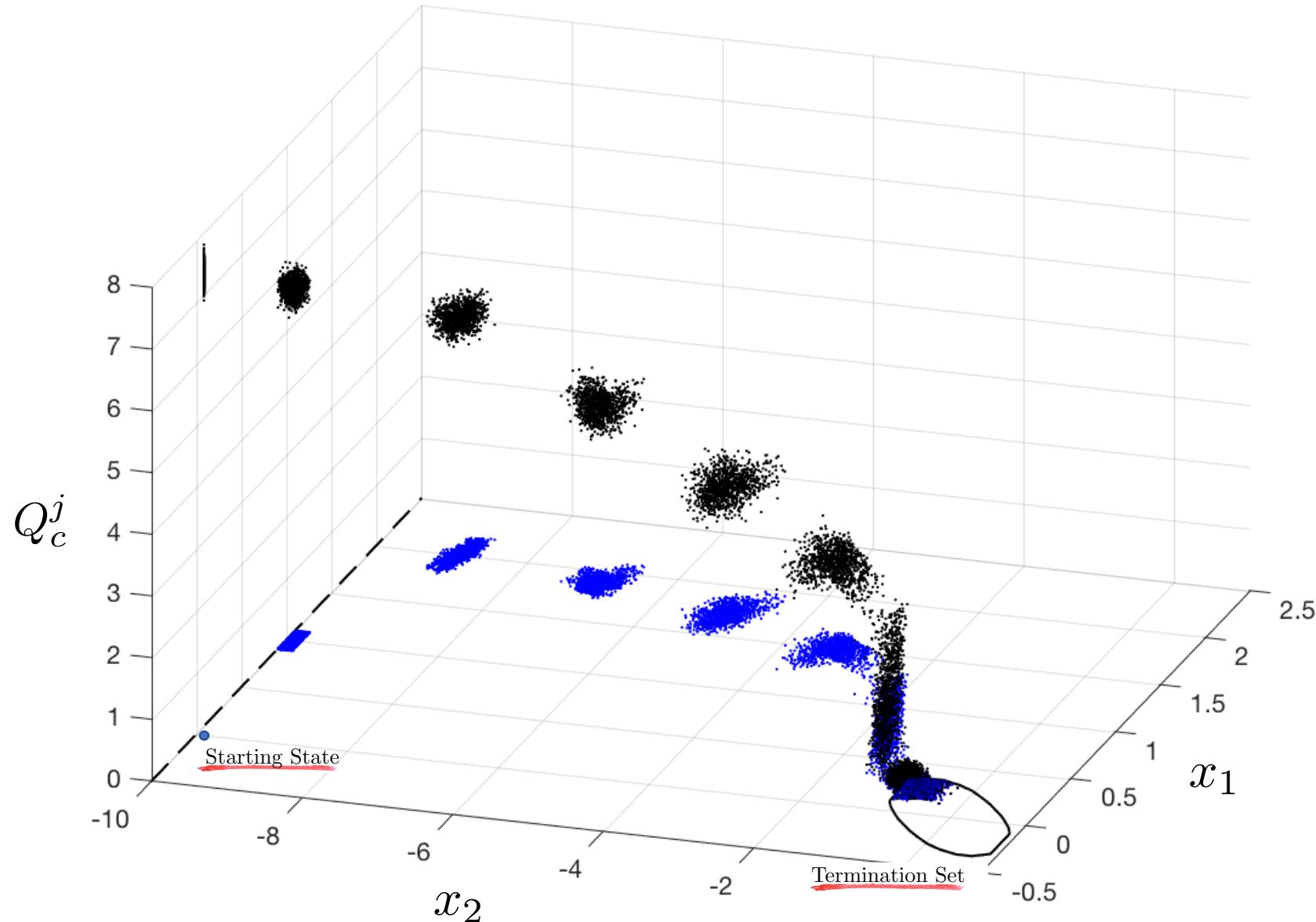
Example: Constrained LQR Uncertain System



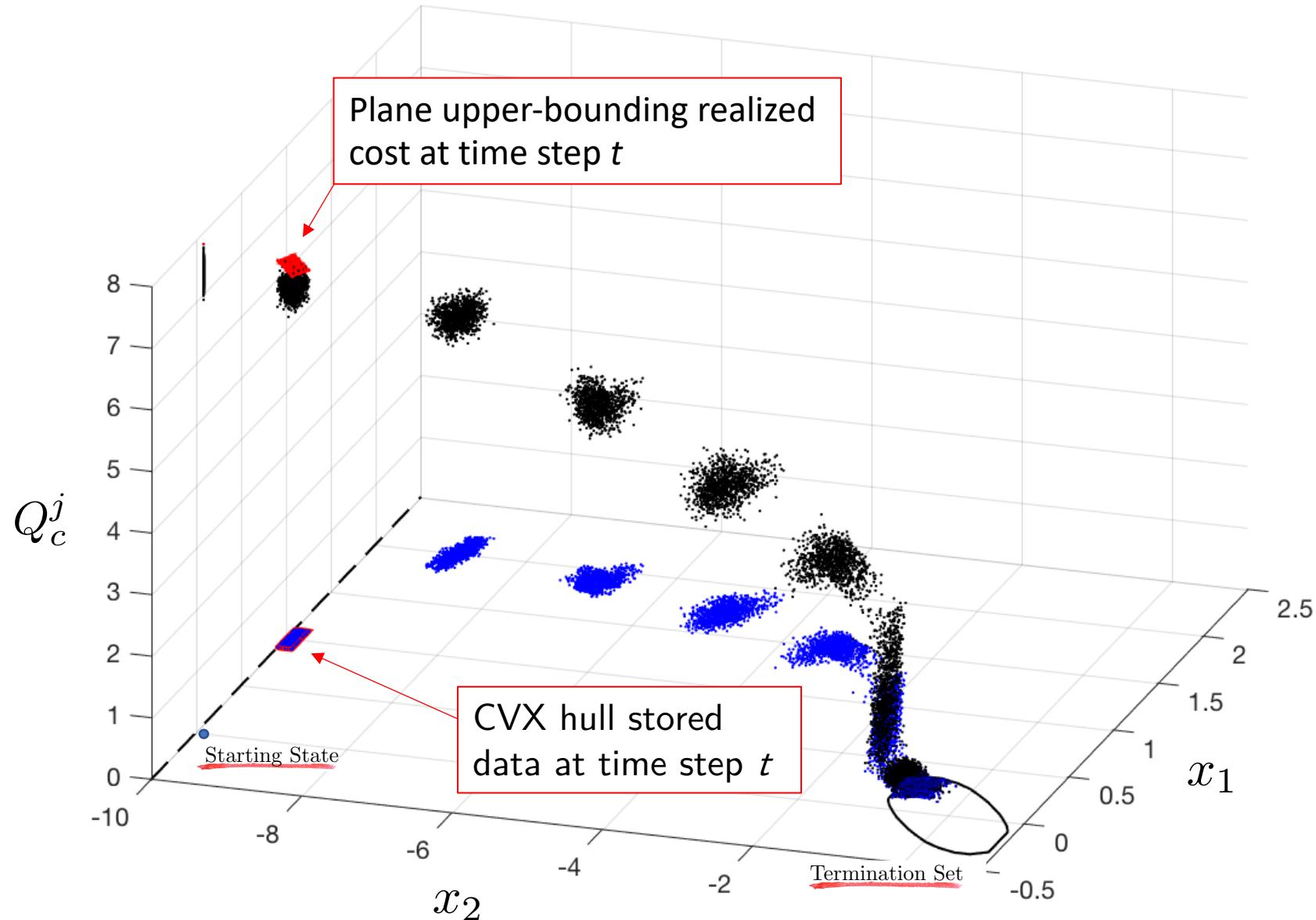
Example: Constrained LQR Uncertain System



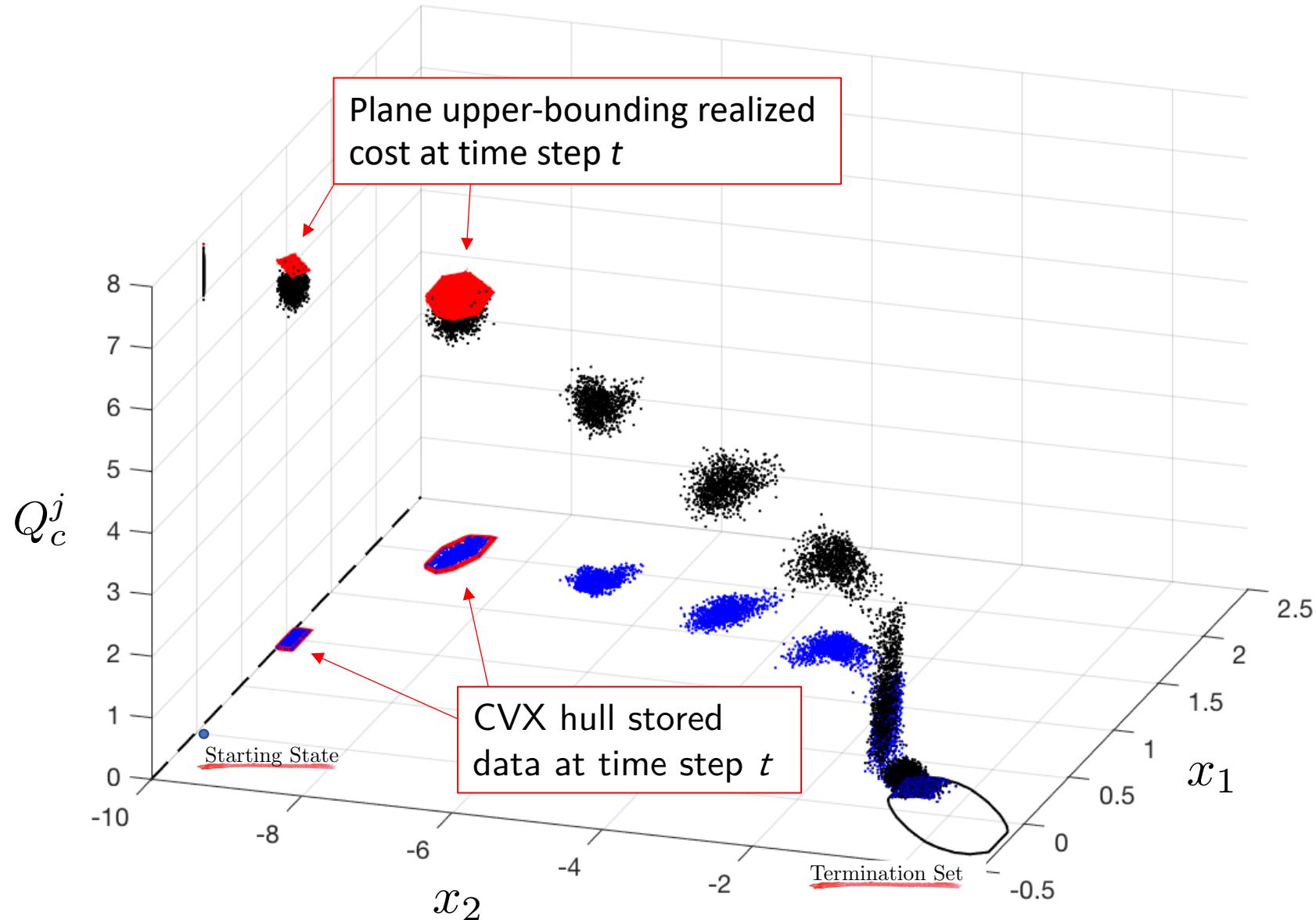
Example: Constrained LQR Uncertain System



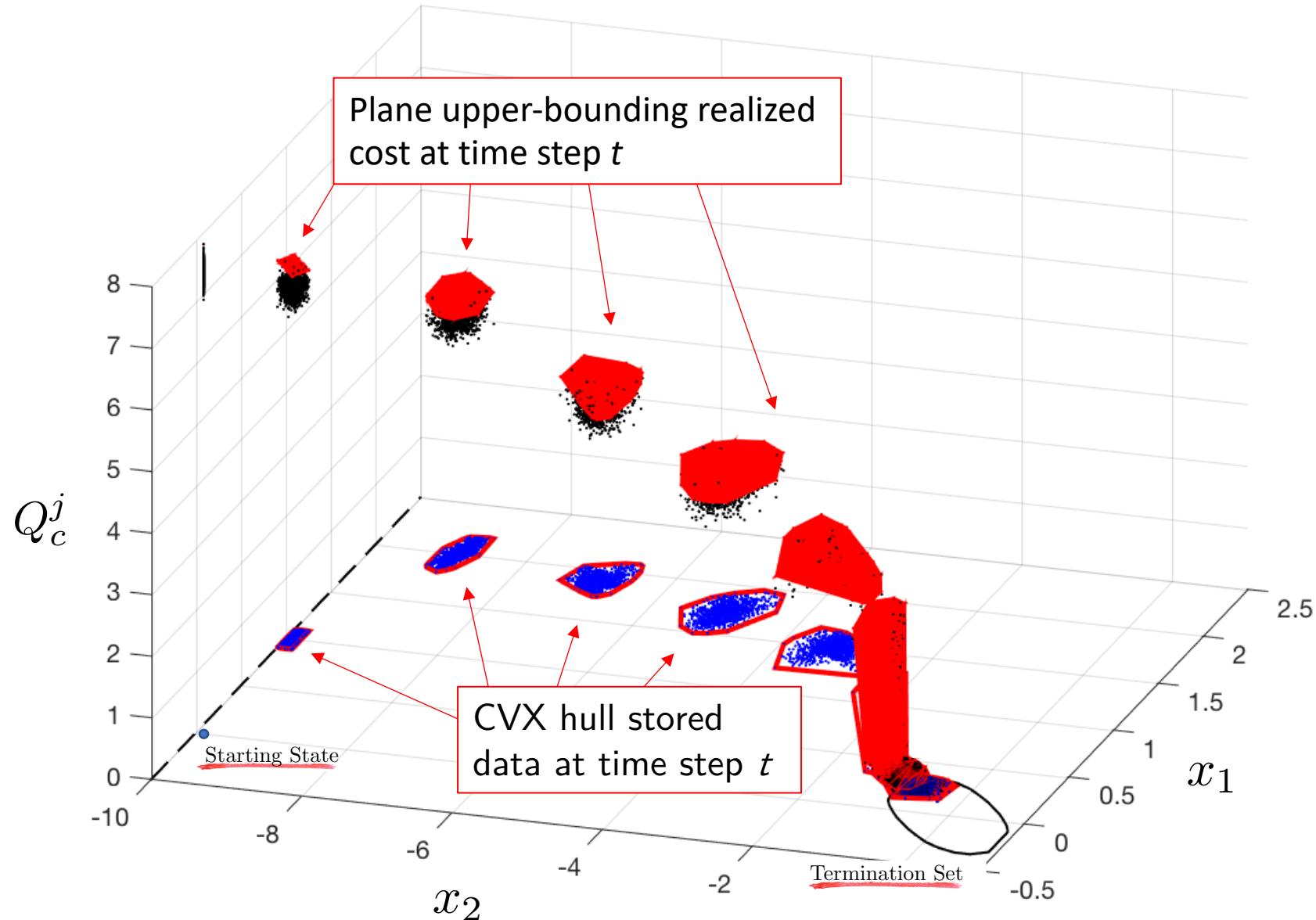
Example: Constrained LQR Uncertain System



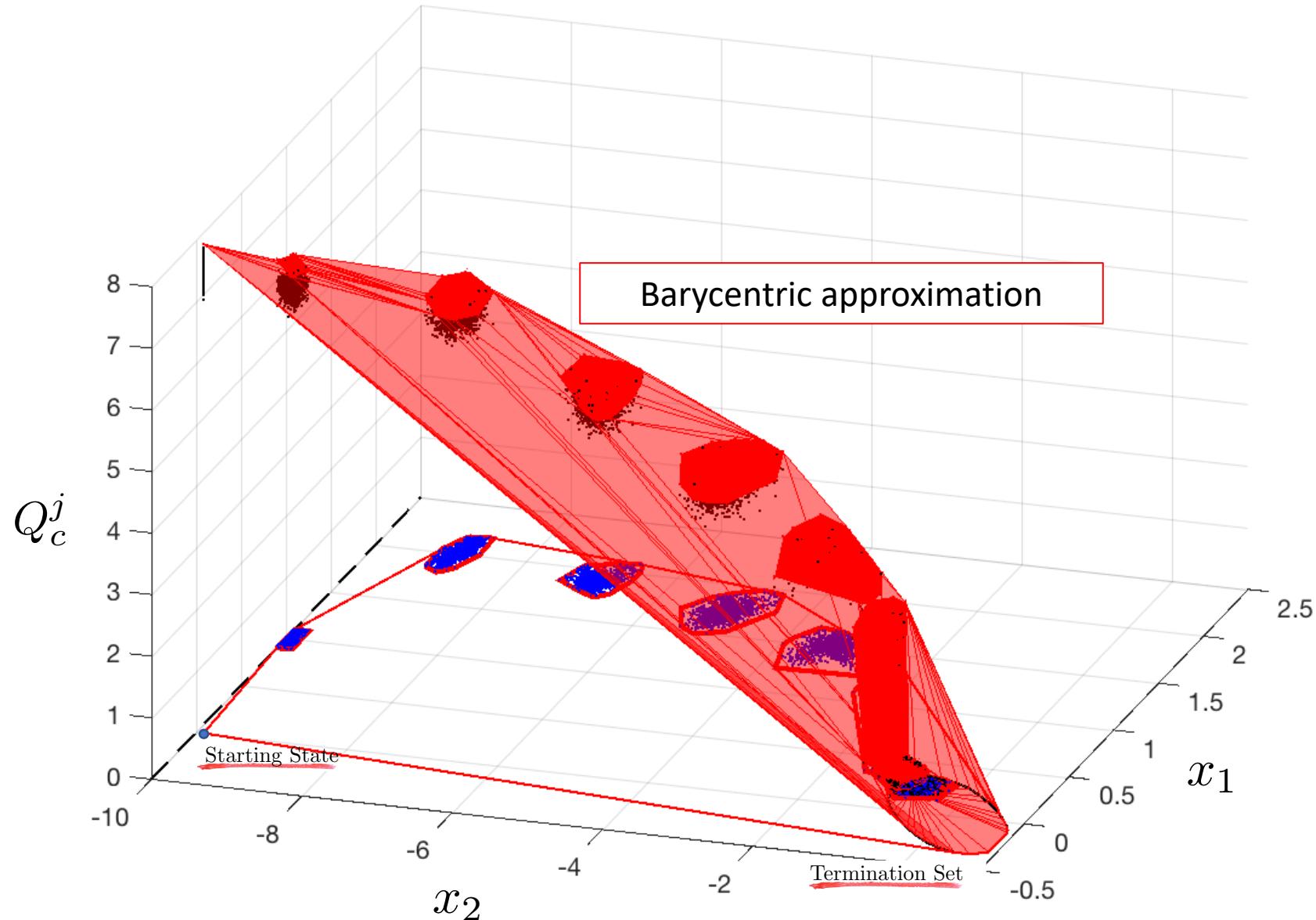
Example: Constrained LQR Uncertain System



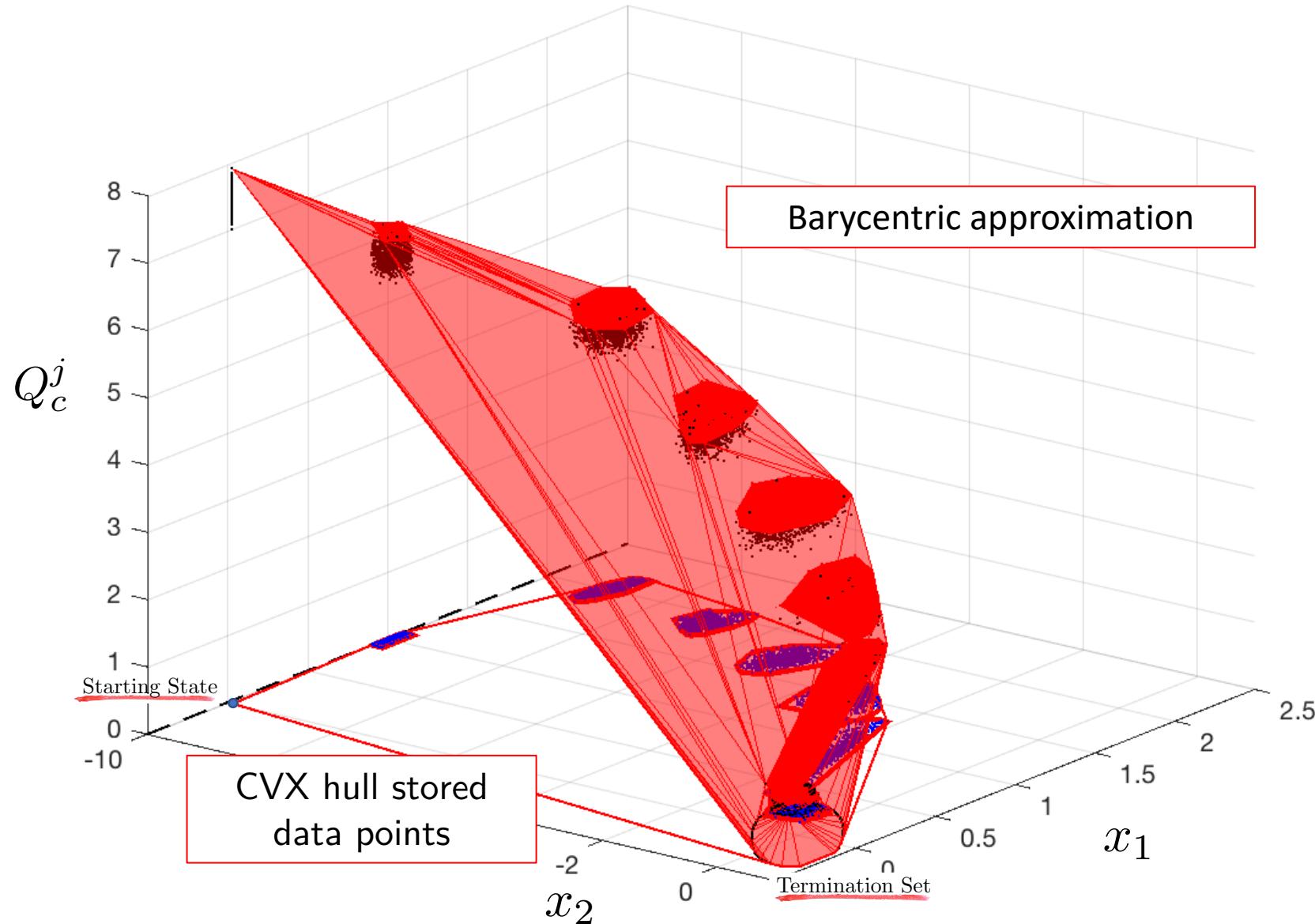
Example: Constrained LQR Uncertain System



Example: Constrained LQR Uncertain System



Example: Constrained LQR Uncertain System



Example: Constrained LQR Uncertain System

