CS159 Lecture 5: LMPC and Model Learning

Ugo Rosolia

Caltech

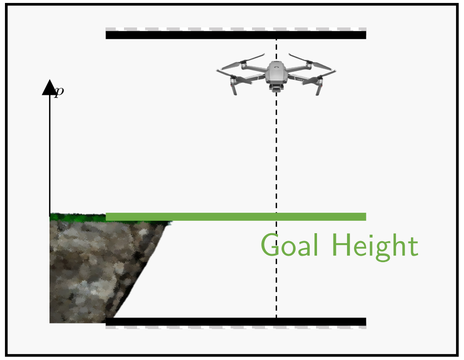
Spring 2021

Today's Class

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

Recap of Lecture #4

Why did we focus on feasibility?

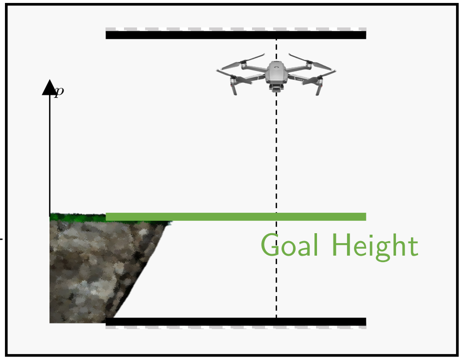


Recap of Lecture #4

Why did we focus on feasibility?

Solve a Finite Time Optimal Control Problem (FTOCP)

Measured state $x(t)$

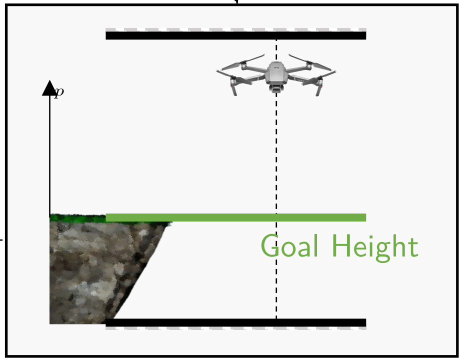


Recap of Lecture #4

Why did we focus on feasibility?

Solve a Finite Time Optimal Control Problem (FTOCP)

Control action $u(t)$



Measured state $x(t)$

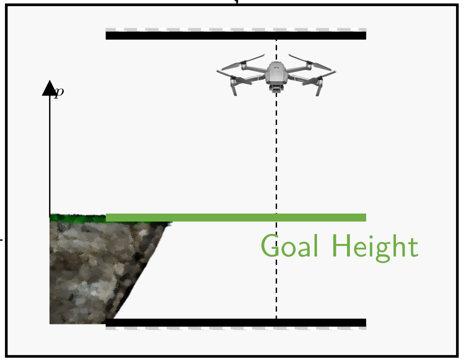
Recap of Lecture #4

Why did we focus on feasibility?

Time Loop: Perform the control task

Solve a Finite Time Optimal Control Problem (FTOCP)

Control action $u(t)$



Measured state $x(t)$

Recap of Lecture #4

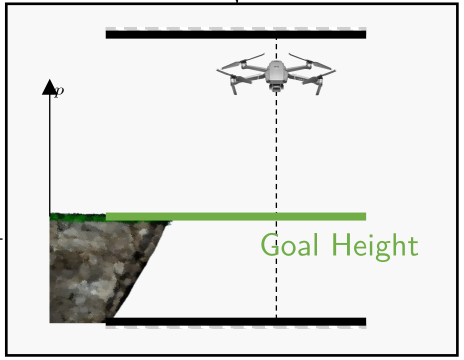
Why did we focus on feasibility?

Time Loop: Perform the control task

Solve a Finite Time Optimal Control Problem (FTOCP) where we enforce

$$x(t) \in \mathcal{X} \text{ and } u(t) \in \mathcal{U}$$

Control action $u(t)$



Measured state $x(t)$

Recap of Lecture #4

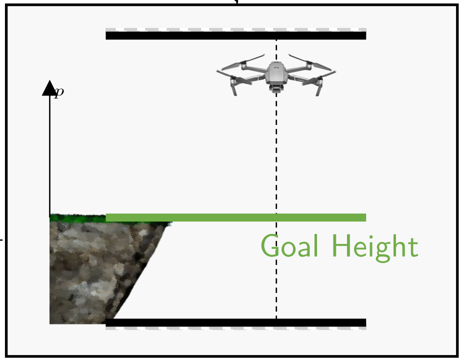
Why did we focus on feasibility?

Time Loop: Perform the control task

Solve a Finite Time Optimal Control Problem (FTOCP) where we enforce

$$x(t) \in \mathcal{X} \text{ and } u(t) \in \mathcal{U}$$

Control action $u(t)$



Measured state $x(t)$

Feasibility of the FTCOP at all times

Recap of Lecture #4

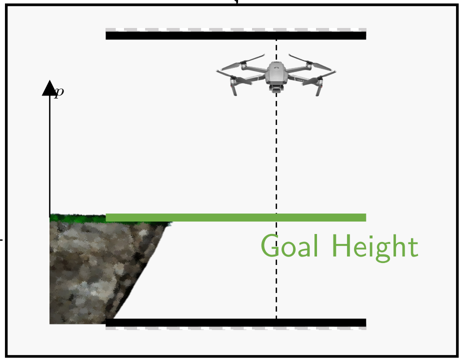
Why did we focus on feasibility?

Time Loop: Perform the control task

Solve a Finite Time Optimal Control Problem (FTOCP) where we enforce

$$x(t) \in \mathcal{X} \text{ and } u(t) \in \mathcal{U}$$

Control action $u(t)$



Measured state $x(t)$

Feasibility of the FTCOP at all times

⬇

Safety

Recap of Lecture #4

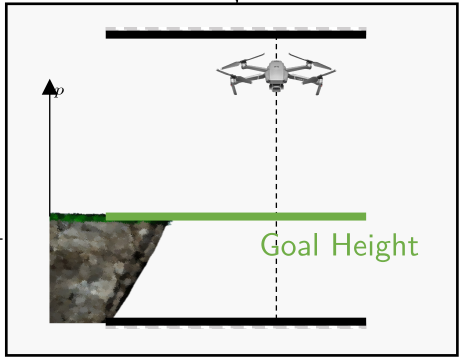
Why did we focus on feasibility?

Time Loop: Perform the control task

Solve a Finite Time Optimal Control Problem (FTOCP) where we enforce

$$x(t) \in \mathcal{X} \text{ and } u(t) \in \mathcal{U}$$

Control action $u(t)$



Measured state $x(t)$

Feasibility of the FTCOP at all times

~~$\Downarrow$~~ Not true when constraints are relaxed with slack variables!

Safety

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

$$J_t^*(x(t)) > J_{t+1}^*(x(t+1)),\ \forall x(t) \neq 0$$

Recap of Lecture #4

Stability Proof:

We have shown that

$$J_t^*(x(t)) > J_{t+1}^*(x(t+1)),\ \forall x(t) \neq 0$$

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

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

Recap of Lecture #4

Stability Proof:

We have shown that

$$\left.\begin{aligned} &J_t^*(x(t)) < J_{t+1}^*(x(t+1)),\ \forall x(t) \neq 0 \\ &J_t^*(x) > 0, \forall x \neq 0 \\ &J_t^*(0) = 0 \end{aligned}\right\} \lim_{t\to\infty} J_t^*(x(t)) = 0$$

Recap of Lecture #4

Stability Proof:

We have shown that

$$\left.\begin{aligned} &J_t^*(x(t)) > J_{t+1}^*(x(t+1)),\ \forall x(t) \neq 0 \\ &J_t^*(x) > 0, \forall x \neq 0 \\ &J_t^*(0) = 0 \end{aligned}\right\} \lim_{t\to\infty} J_t^*(x(t)) = 0$$

To show stability we need to that $J_t^*(x)$ is Lipchitz continuous

Recap of Lecture #4

Stability Proof:

We have shown that

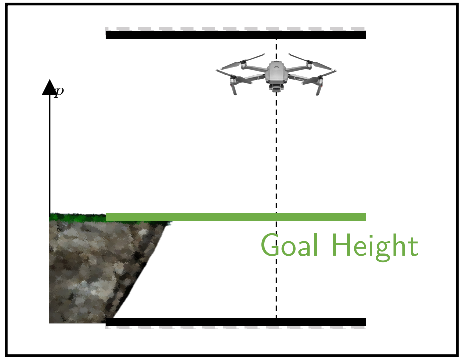
$$\left.\begin{aligned} &J_t^*(x(t)) > J_{t+1}^*(x(t+1)),\ \forall x(t) \neq 0 \\ &J_t^*(x) > 0, \forall x \neq 0 \\ &J_t^*(0) = 0 \end{aligned}\right\} \lim_{t\to\infty} J_t^*(x(t)) = 0$$

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

Lars Grüne, and Christopher M. Kellett. "ISS-Lyapunov functions for discontinuous discrete-time systems." IEEE Transactions on Automatic Control 59, no. 11 (2014): 3098-3103.

LMPC Recap #4

Time Loop: Perform the control task



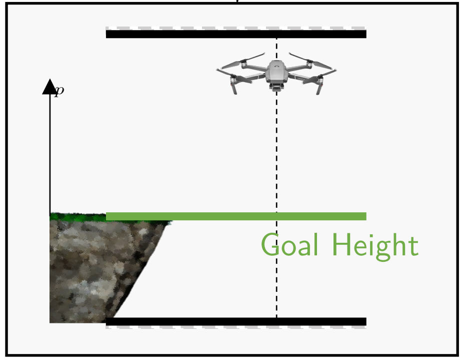
LMPC Recap #4

Time Loop: Perform the control task

Solve a Finite Time Optimal Control Problem (FTOCP) with:

$$\mathcal{X}_F = \mathcal{CS}^j \text{ and } V(x) = V^j(x)$$

Control action $u(t)$



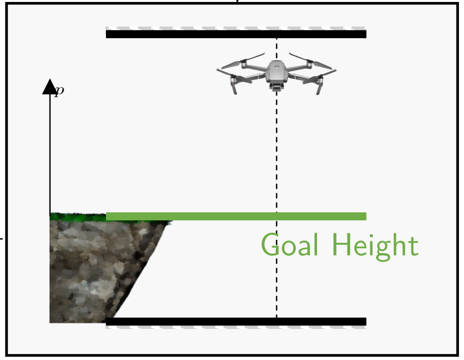
LMPC Recap #4

Time Loop: Perform the control task

Solve a Finite Time Optimal Control Problem (FTOCP) with:

$$\mathcal{X}_F = \mathcal{CS}^j \text{ and } V(x) = V^j(x)$$

Control action $u(t)$



Measured state $x(t)$

LMPC Recap #4

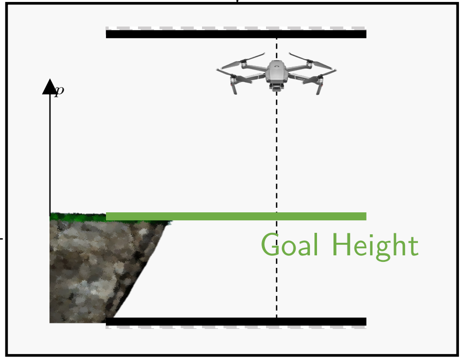
Iteration Loop: Update terminal components

Time Loop: Perform the control task

Solve a Finite Time Optimal Control Problem (FTOCP) with:

$$\mathcal{X}_F = \mathcal{CS}^j \text{ and } V(x) = V^j(x)$$

Control action $u(t)$



Measured state $x(t)$

LMPC Recap #4

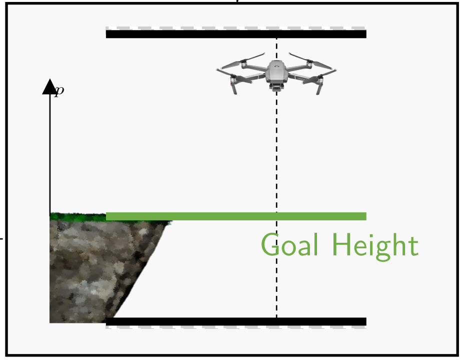
Iteration Loop: Update terminal components

Time Loop: Perform the control task

Solve a Finite Time Optimal Control Problem (FTOCP) with:

$$\mathcal{X}_F = \mathcal{CS}^j \text{ and } V(x) = V^j(x)$$

Control action $u(t)$



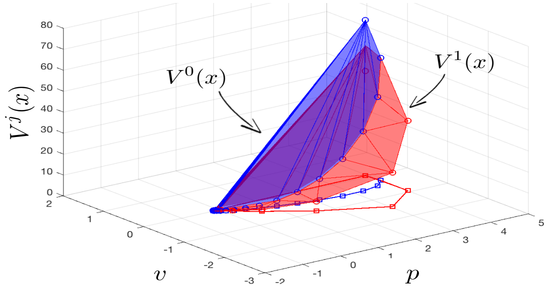
Measured state $x(t)$

Control task completed
Store closed-loop

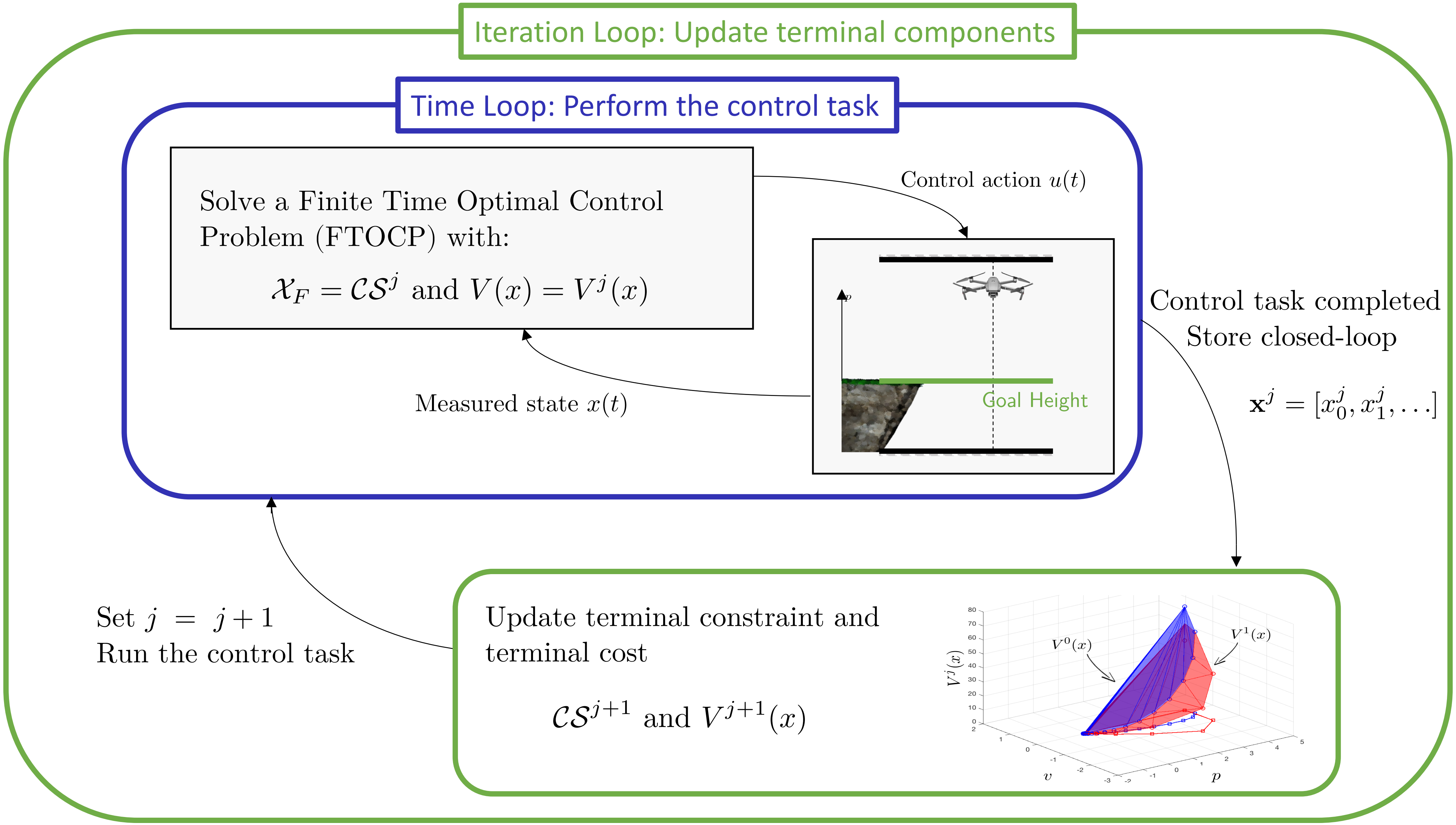
$$\mathbf{x}^j = [x_0^j, x_1^j, \dots]$$

Update terminal constraint and terminal cost

$$\mathcal{CS}^{j+1} \text{ and } V^{j+1}(x)$$



LMPC Recap #4



Iteration Loop: Update terminal components
Time Loop: Perform the control task
Solve a Finite Time Optimal Control Problem (FTOCP) with:
$\mathcal{X}_F = \mathcal{CS}^j$ and $V(x) = V^j(x)$
Control action $u(t)$
Goal Height
Measured state $x(t)$
Control task completed
Store closed-loop
$\mathbf{x}^j = [x_0^j, x_1^j, \ldots]$
Set $j = j+1$
Run the control task
Update terminal constraint and terminal cost
$\mathcal{CS}^{j+1}$ and $V^{j+1}(x)$
$V^0(x)$
$V^1(x)$
$V^j(x)$
v
p

Recap of Lecture #4

Theorem

Let $\mathbf{x}^j = [x_0^j, x_1^j, \ldots]$ 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 $J_{0\to\infty}^j(x_S)$ is non-increasing, i.e.,

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

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

Optimality Proof Sketch

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

Optimality Proof Sketch

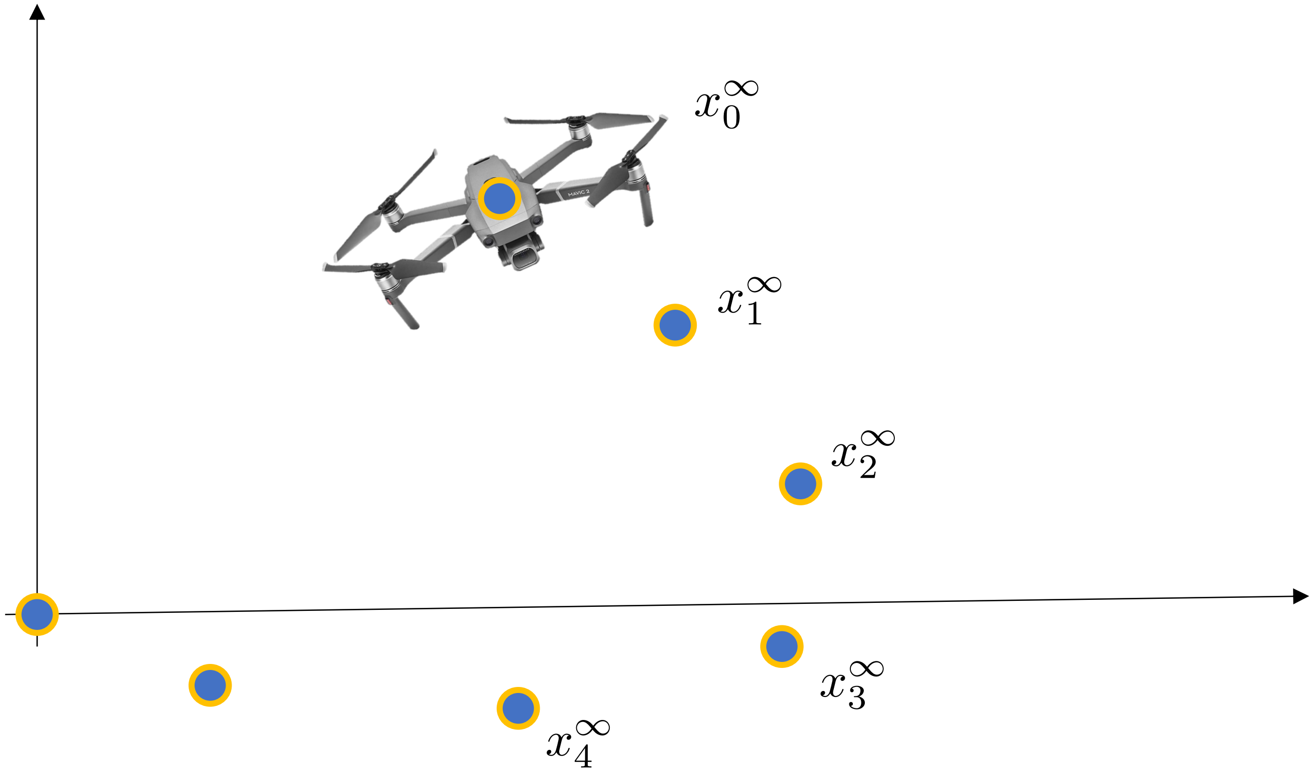
Define the trajectory at convergence $\mathbf{x}^\infty = [x_0^\infty, x_1^\infty, \ldots]$

Given an MPC horizon $N = 2$

Optimality Proof Sketch

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

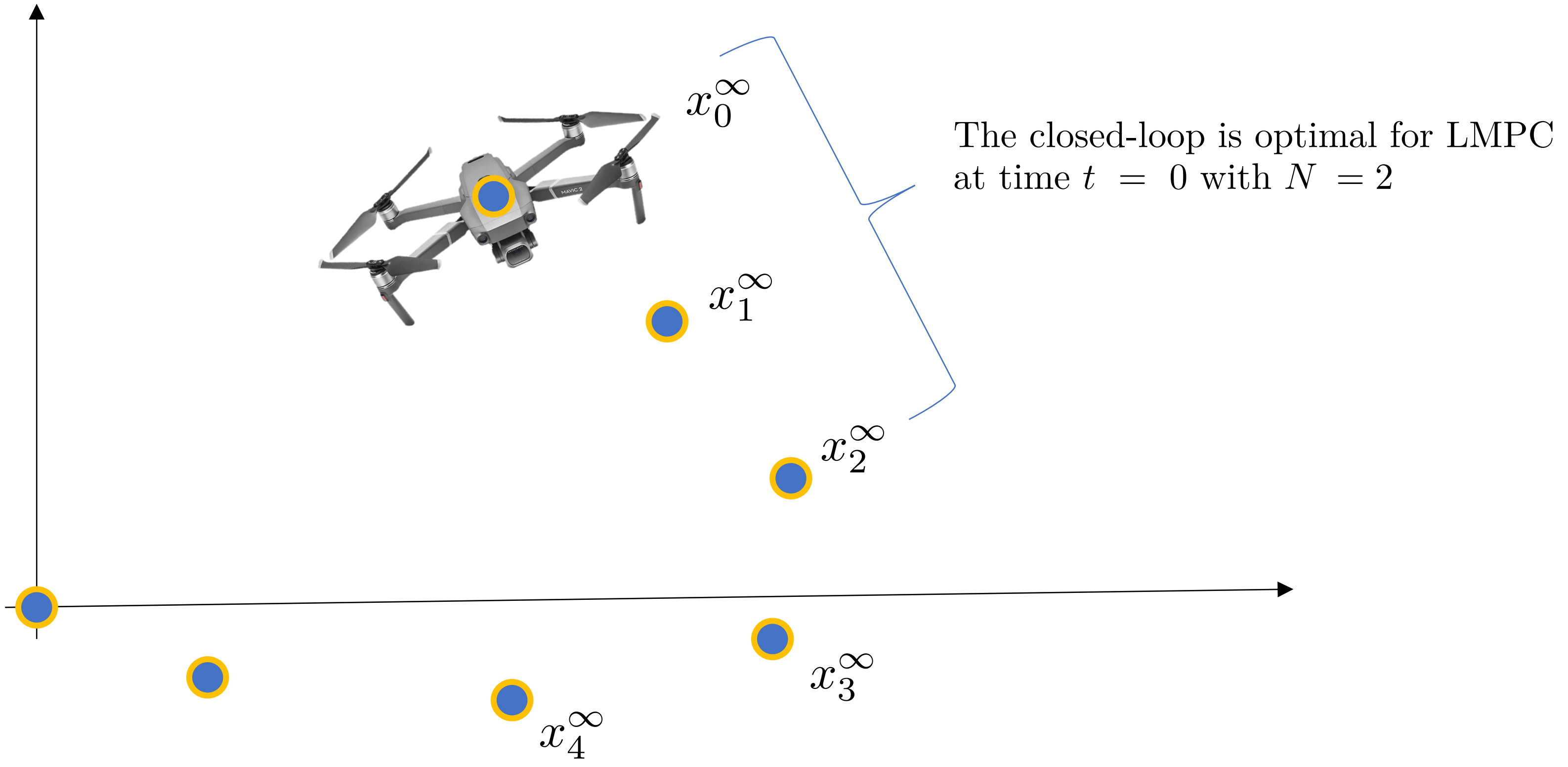
Given an MPC horizon $N = 2$



Optimality Proof Sketch

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

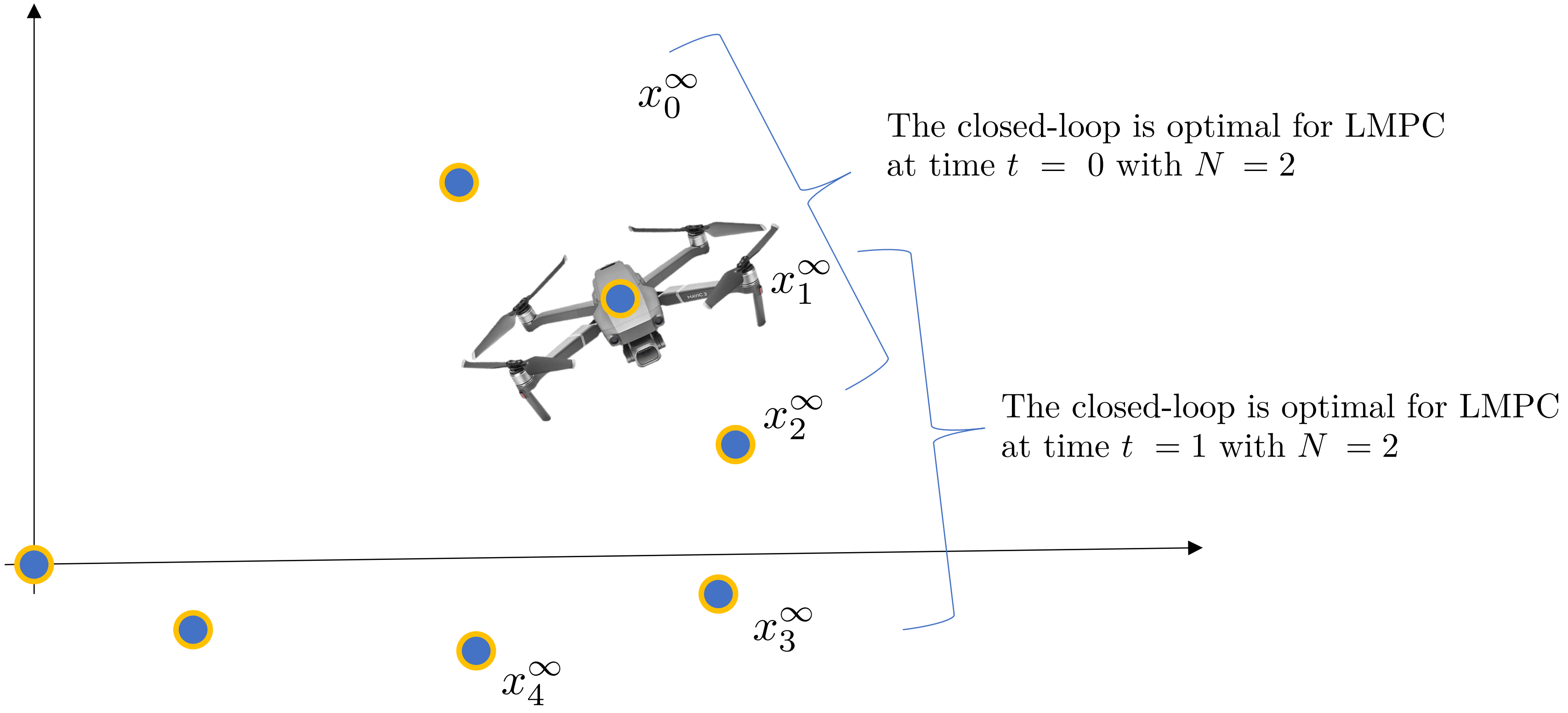
Given an MPC horizon $N = 2$



Optimality Proof Sketch

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

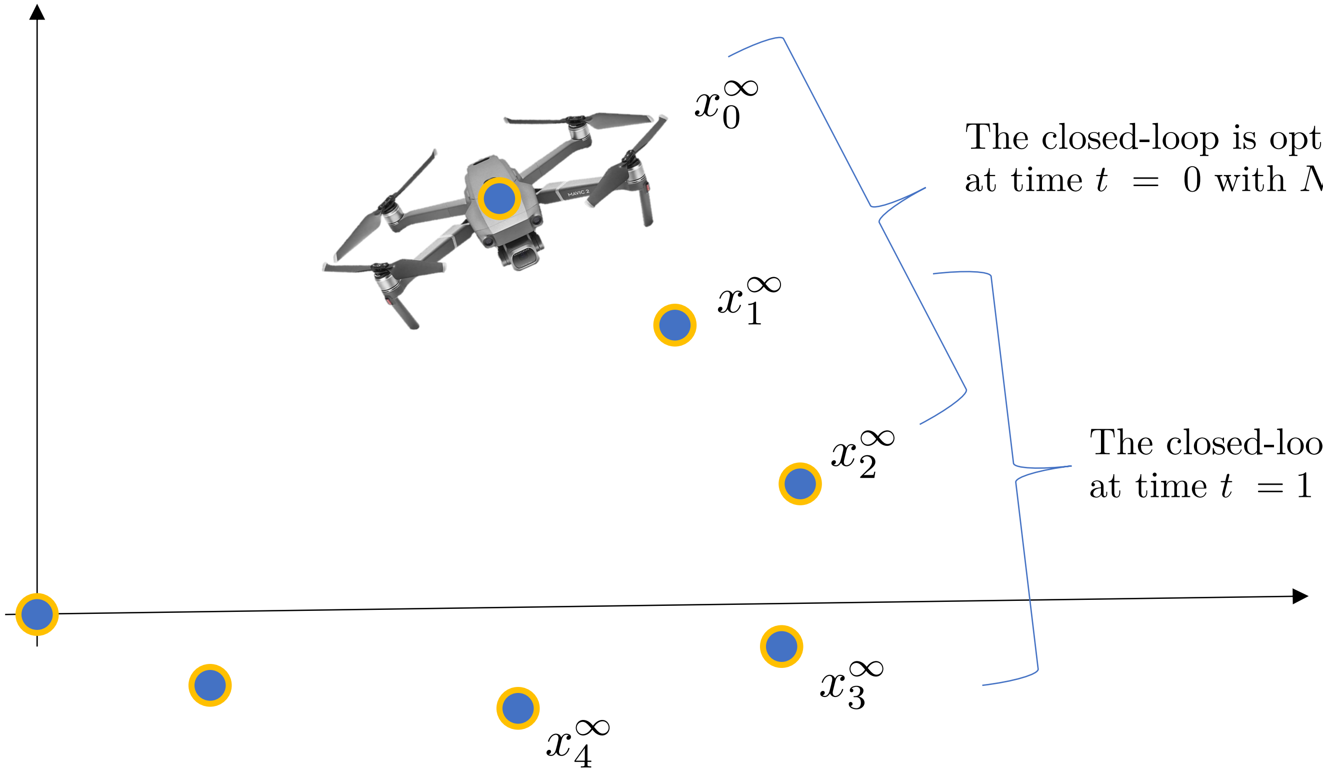
Given an MPC horizon $N = 2$



Optimality Proof Sketch

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

Given an MPC horizon $N = 2$



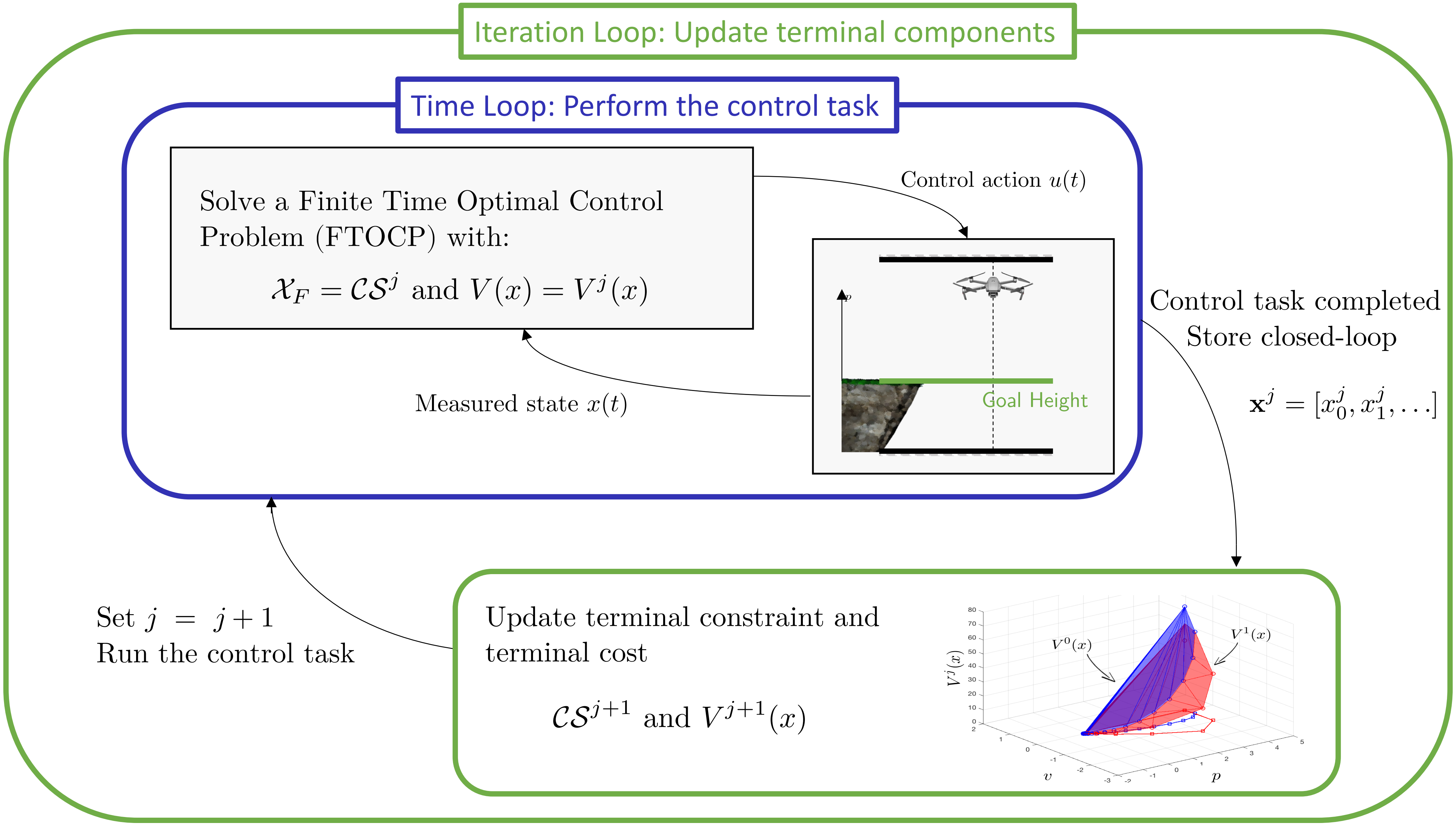
The closed-loop is optimal for LMPC at time $t = 0$ with $N = 2$

The closed-loop is optimal for LMPC at time $t = 1$ with $N = 2$

The Closed-loop is optimal for LMPC at time $t = 0$ with $N = 3$

Under some conditions associated the horizon length!

LMPC Recap #4


Iteration Loop: Update terminal components
Time Loop: Perform the control task
Solve a Finite Time Optimal Control Problem (FTOCP) with:
$\mathcal{X}_F = \mathcal{CS}^j$ and $V(x) = V^j(x)$
Control action $u(t)$
Goal Height
Measured state $x(t)$
Control task completed
Store closed-loop
$\mathbf{x}^j = [x_0^j, x_1^j, \dots]$
Set $j = j+1$
Run the control task
Update terminal constraint and terminal cost
$\mathcal{CS}^{j+1}$ and $V^{j+1}(x)$
$V^0(x)$
$V^1(x)$
$V^j(x)$
v
p

Linear(ized) LMPC

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

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

s.t.

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

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

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

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

Linear(ized) LMPC

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

$$J^{\text{LMPC},j}_{t\rightarrow t+N}(x^j_t) = \min_{\substack{u_{t|t},\dots,u_{t+N-1|t} \\ \boxed{\lambda^0_0,\dots,\lambda^{j-1}_{T_f}}}} \sum_{k=t}^{t+N-1} h(x_{k|t}, u_{k|t}) + V^{j-1}(x_{t+N|t})$$

s.t.

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

$$x_{t|t} = x^j_t,$$

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

$$\boxed{x_{t+N|t} = \sum_{i=0}^{j-1}\sum_{k} x^i_k \lambda^i_k,\ \ \sum_{i=0}^{j-1}\sum_{k} \lambda^i_k = 1,\ \lambda^i_k \geq 0.} \Longleftrightarrow \boxed{x_{t+N|t} \in \mathcal{CS}^{j-1}}$$

Linear(ized) LMPC

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

$$J^{\mathrm{LMPC},j}_{t\rightarrow t+N}(x_t^j) = \min_{\substack{u_{t|t},\ldots,u_{t+N-1|t} \\ \boxed{\lambda_0^0,\ldots,\lambda_{T_f}^{j-1}}}} \sum_{k=t}^{t+N-1} h(x_{k|t},u_{k|t}) + \boxed{\sum_{i=0}^{j-1}\sum_{k} J_k^i(x_k^i)\lambda_k^i}$$

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

s.t.

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

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

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

$$\boxed{x_{t+N|t} = \sum_{i=0}^{j-1}\sum_{k} x_k^i\lambda_k^i,\ \ \sum_{i=0}^{j-1}\sum_{k}\lambda_k^i = 1,\ \lambda_k^i \geq 0.} \Longleftrightarrow \boxed{x_{t+N|t} \in \mathcal{CS}^{j-1}}$$

Linear(ized) LMPC

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

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

$$\Updownarrow$$

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

s.t.

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

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

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

$$\boxed{x_{t+N|t} = \sum_{i=0}^{j-1}\sum_{k} x_k^i\lambda_k^i,\quad \sum_{i=0}^{j-1}\sum_{k}\lambda_k^i = 1,\ \lambda_k^i \geq 0.} \Longleftrightarrow \boxed{x_{t+N|t} \in \mathcal{CS}^{j-1}}$$

- 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

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

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

System dynamics
System constraints

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

Safety constraints

$$\boxed{x_k \in \mathcal{X}, \;\; u_k \in \mathcal{U}, \qquad \forall k \in \{0, \ldots, T-1\}}$$

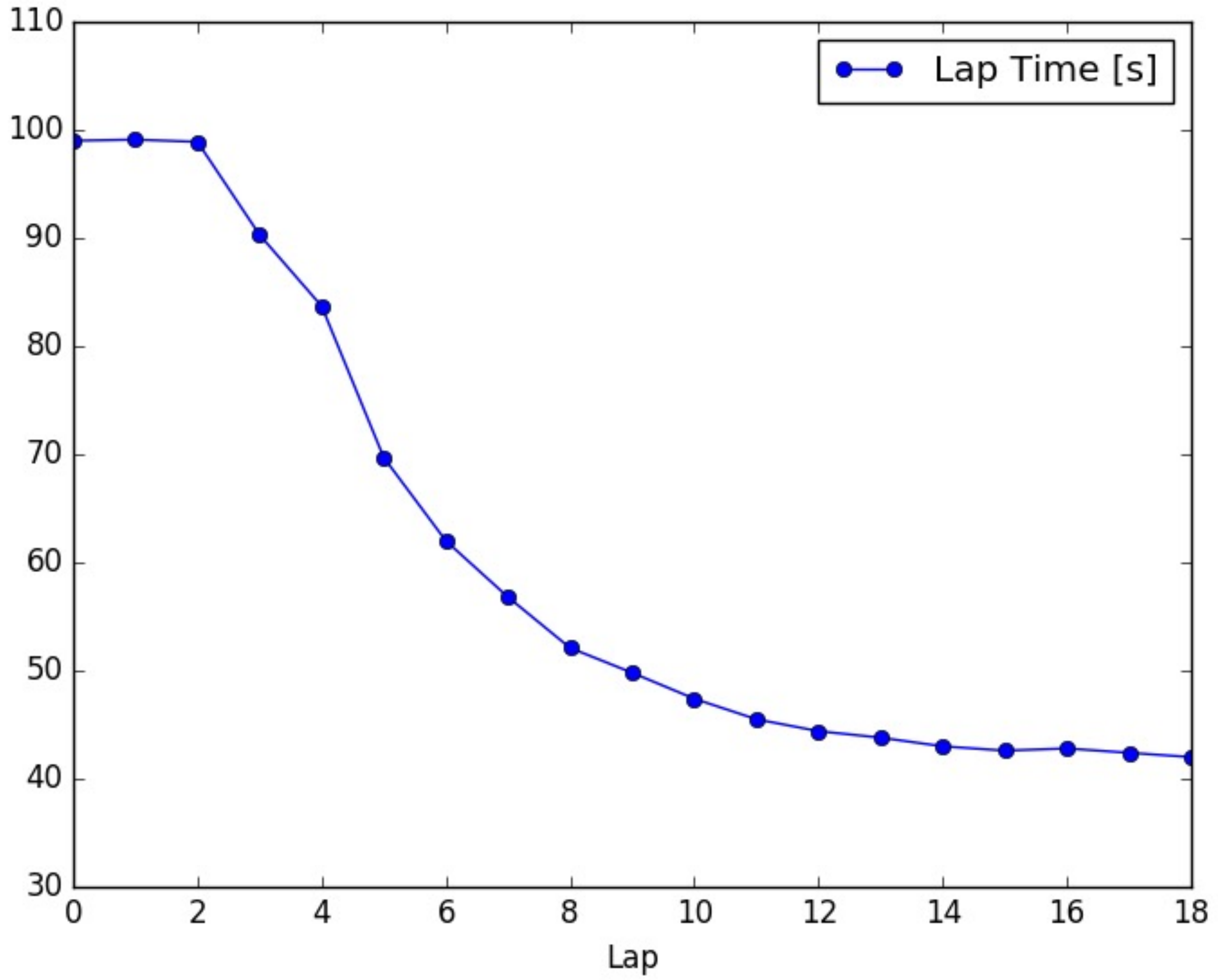
Learning Model Predictive Controller full-size vehicle experiments

Credits: Siddharth Nair, Nitin Kapania and Ugo Rosolia

Learning Model Predictive Controller full-size vehicle experiments

Credits: Siddharth Nair, Nitin Kapania and Ugo Rosolia

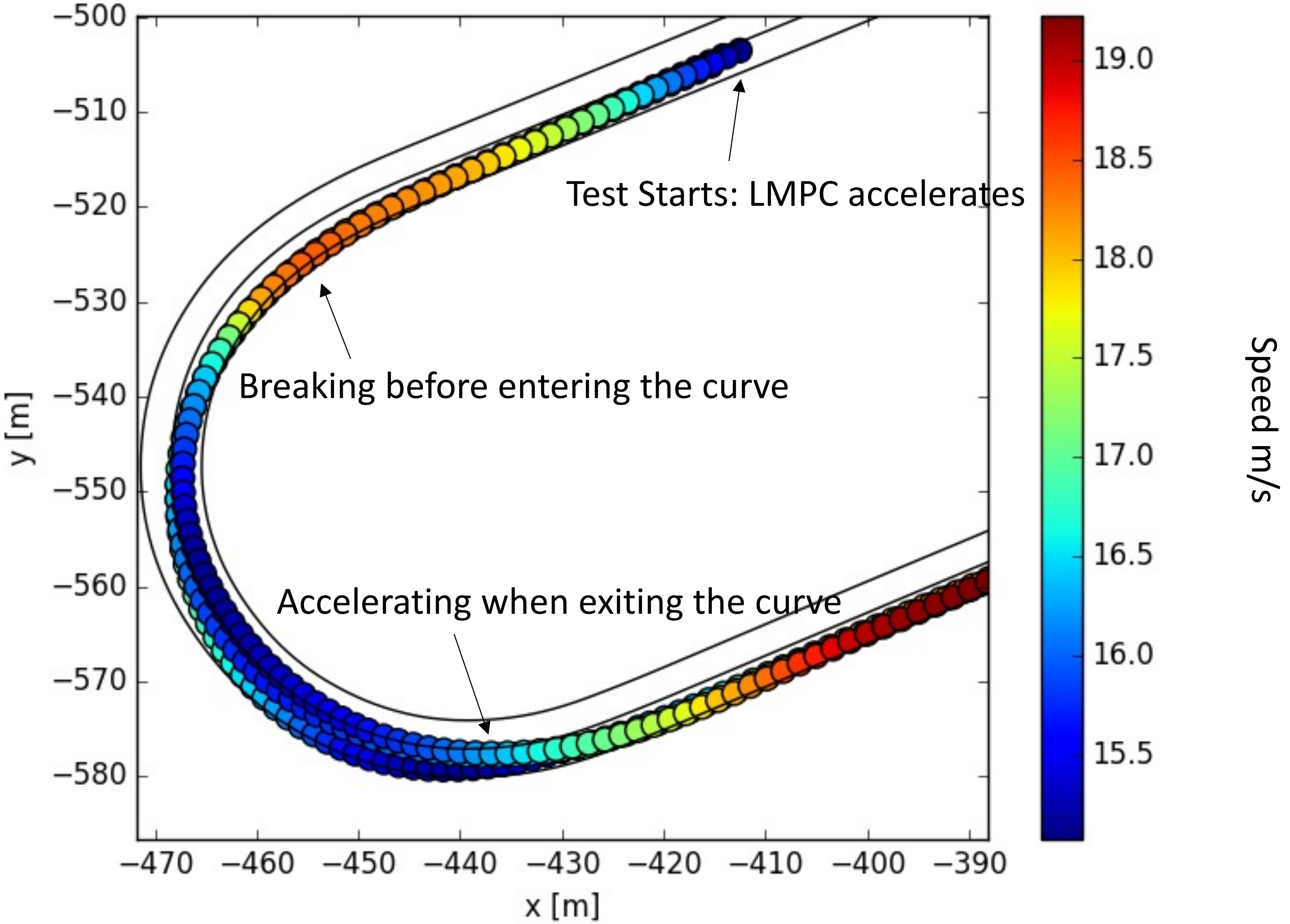
Lap Time



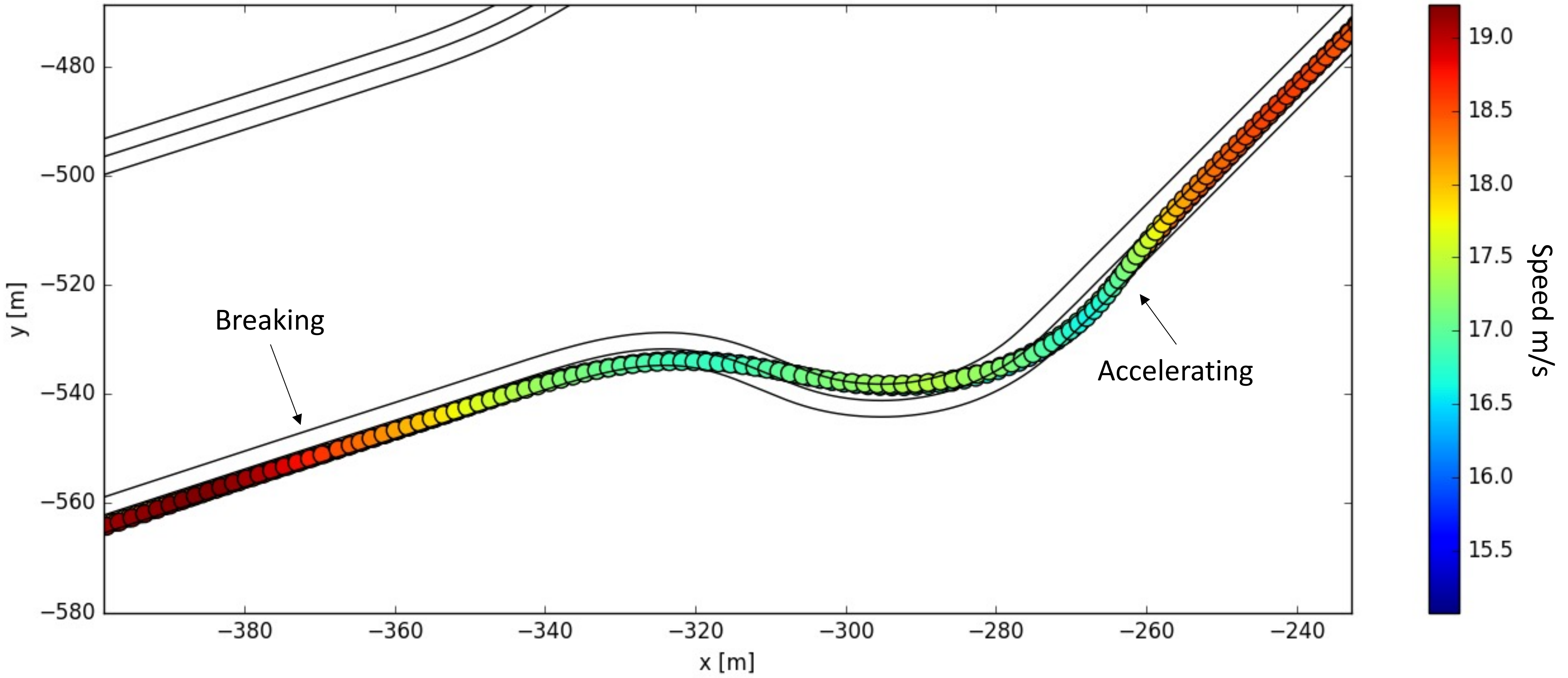
Learning Model Predictive Controller full-size vehicle experiments

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

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

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

System dynamics
System constraints

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

Safety constraints

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

Problem Formulation

Minimum Time Control Problem

$$\min_{T,\mathbf{u}} \boxed{T}$$ Control objective

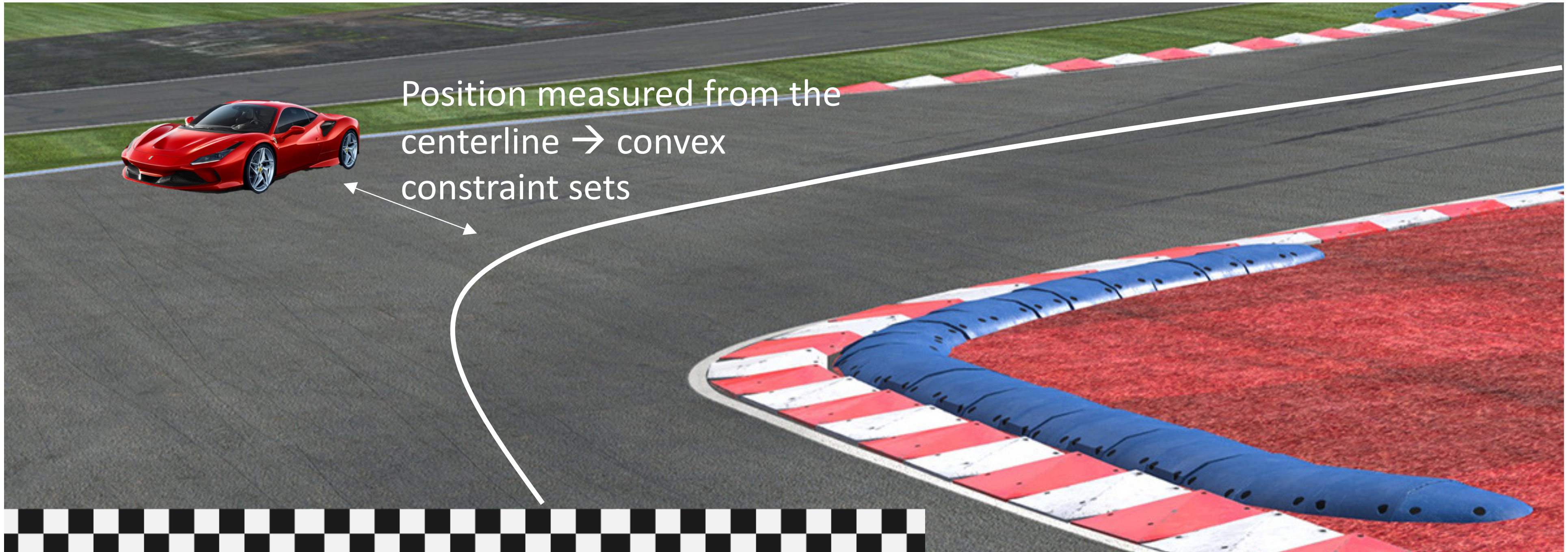
$$\boxed{x_0 = x_s,\ x_T = \mathcal{X}_F}$$ Start & end position

System dynamics
System constraints

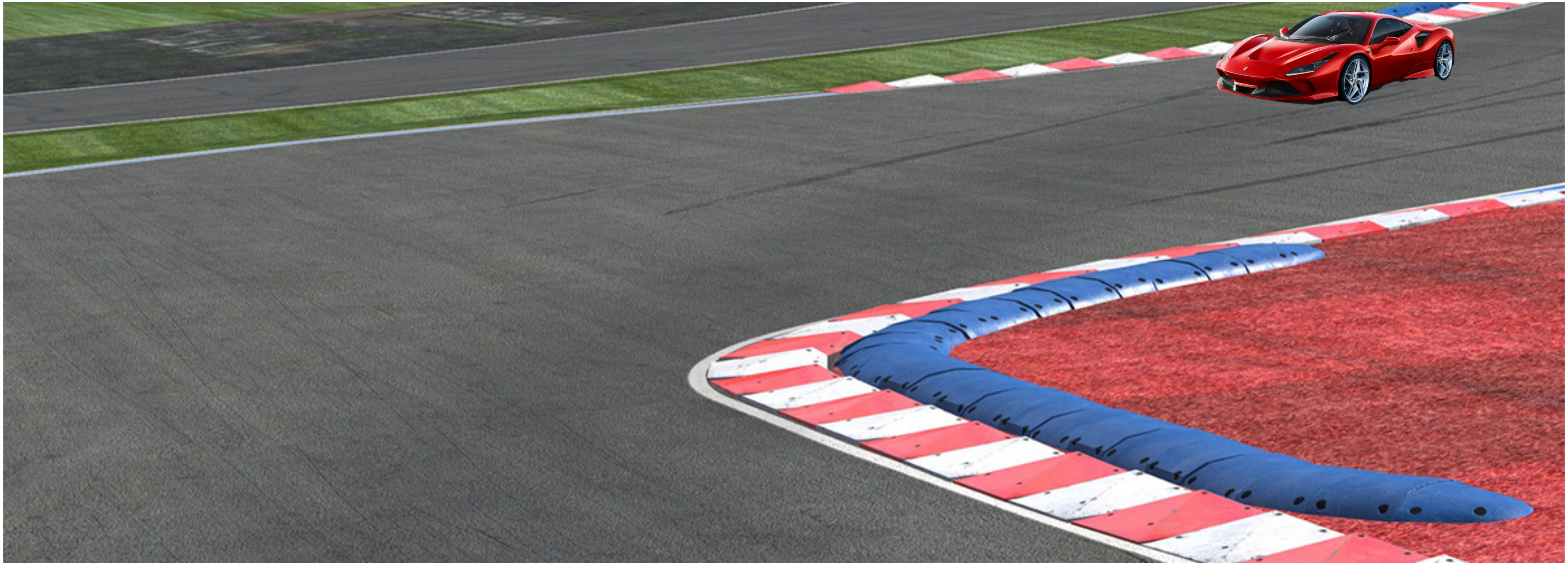
$$\boxed{x_{k+1} = f(x_k, u_k), \quad \forall k \in \{0, \ldots, T-1\}}$$

Safety constraints

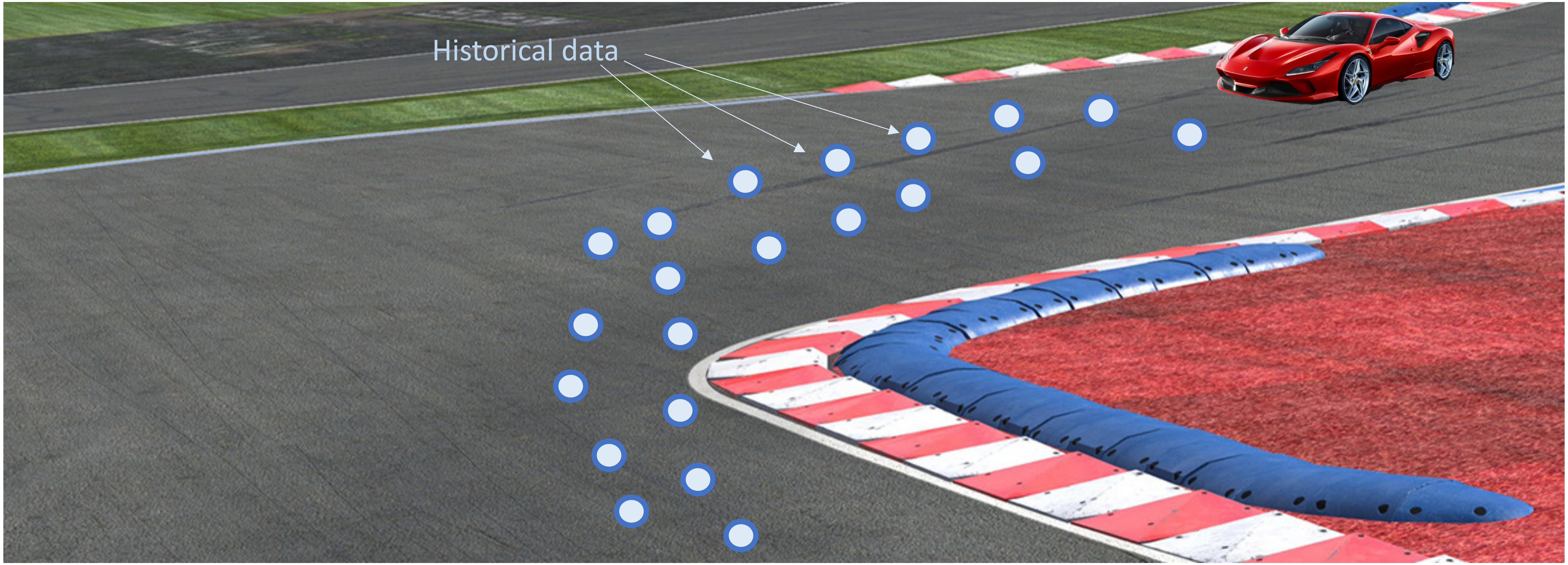
$$\boxed{x_k \in \mathcal{X},\ u_k \in \mathcal{U}, \quad \forall k \in \{0, \ldots, T-1\}}$$



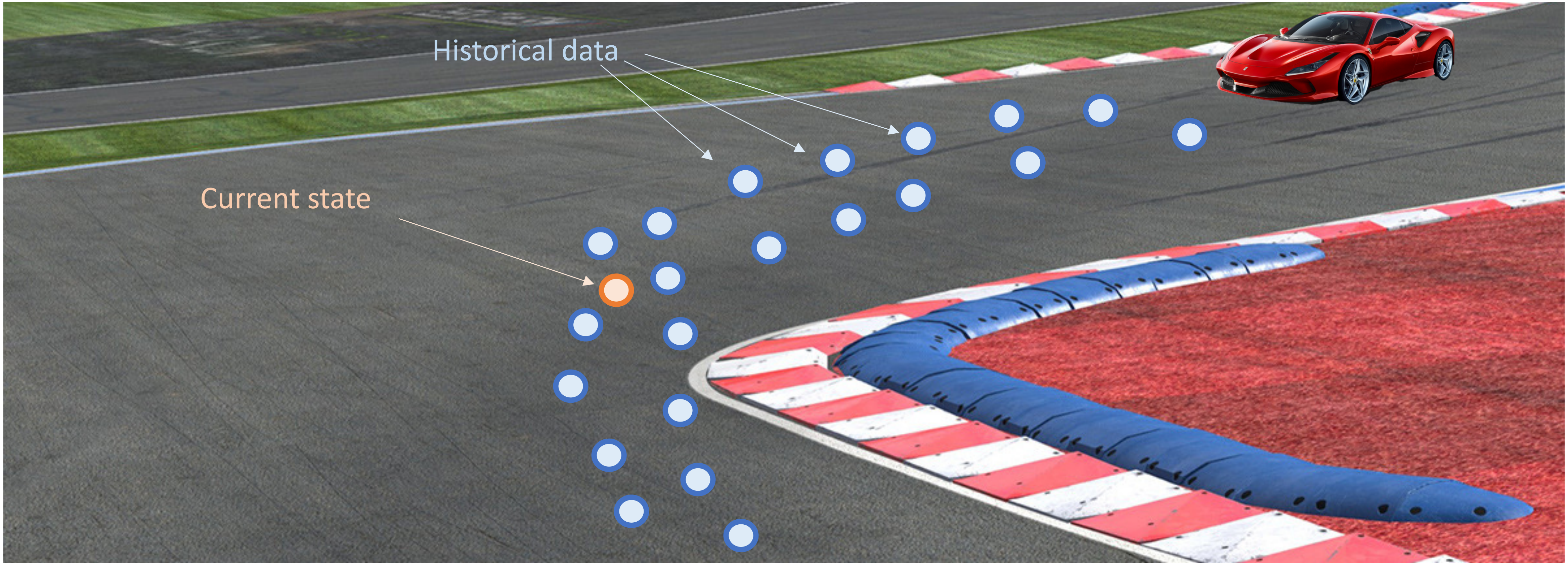
Local Approximations



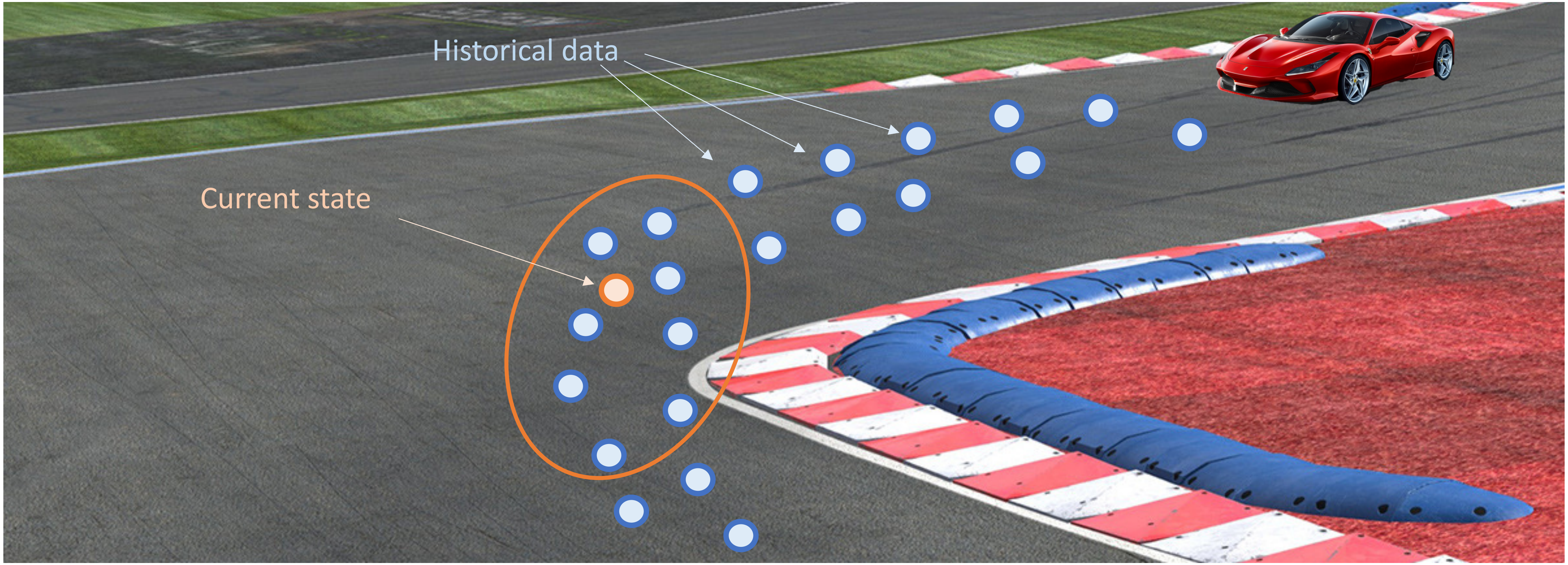
Local Approximations



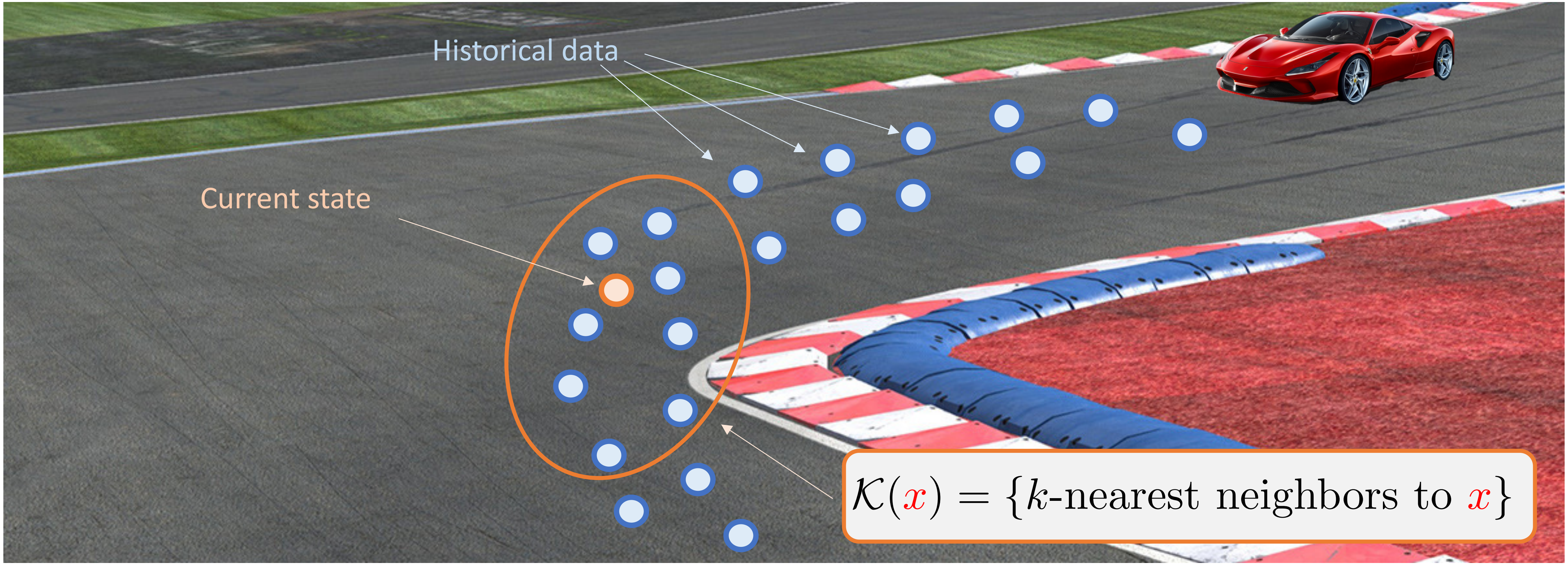
Local Approximations



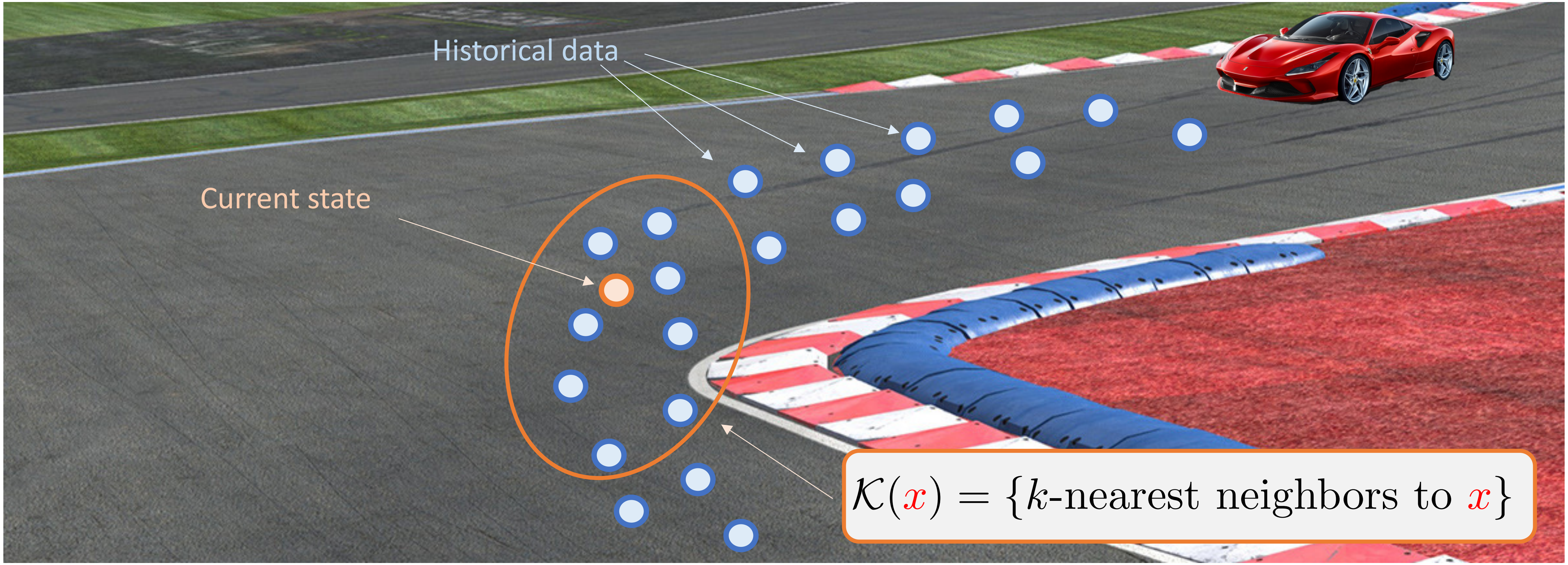
Local Approximations



Local Approximations



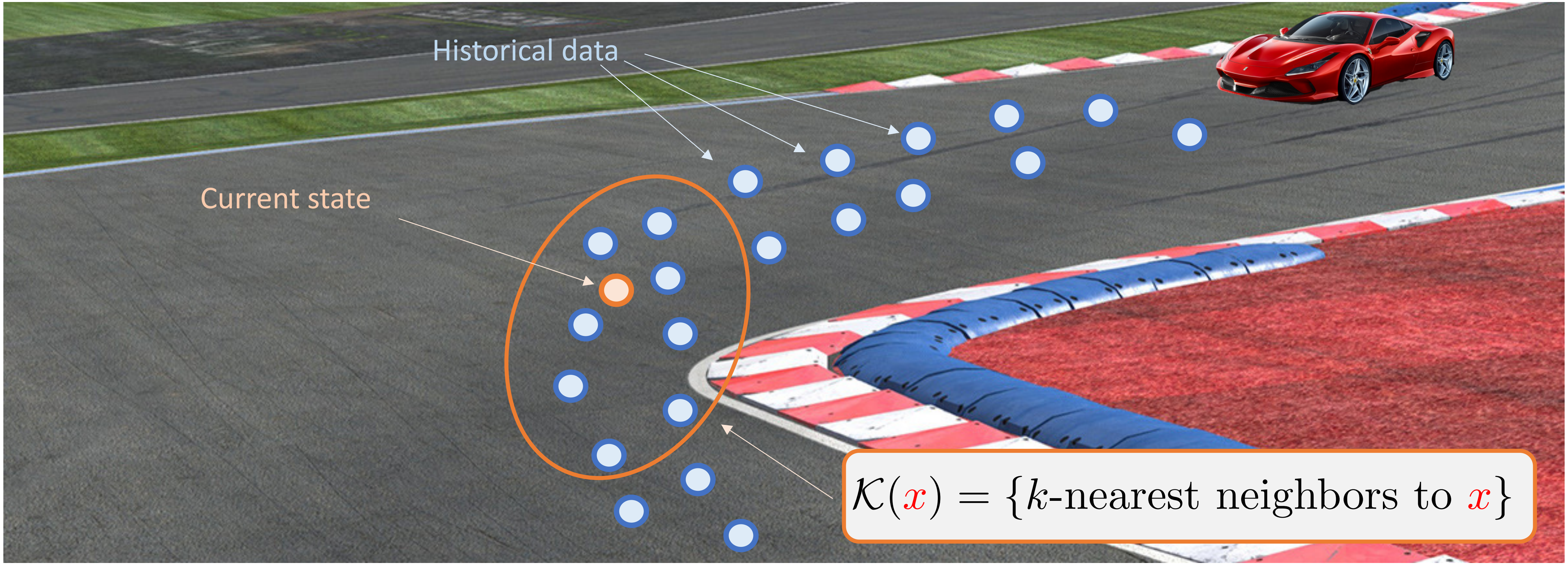
Local Approximations



Local convex safe set approximation:

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

Local Approximations



Local value function approximation:

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

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

Learning Model Predictive Controller

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

$$J^{\mathrm{LMPC},j}_{t\rightarrow t+N}(x^j_t) = \min_{u^j_{t|t},\dots,u^j_{t+N-1|t}} \sum_{k=t}^{t+N-1} h(x^j_{k|t}, u^j_{k|t}) + V^{j-1}(x^j_{t+N|t}, x^+_{t+N})$$

s.t.

$$x^j_{k+1|t} = A^j_{k|t} x^j_{k|t} + B^j_{k|t} u^j_{k|t} + C^j_{k|t}$$

$$x^j_{t|t} = x^j_t,$$

$$x^j_{k|t} \in \mathcal{X},\ u^j_{k|t} \in \mathcal{U},\ \forall k \in [t, \cdots, t+N-1]$$

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

Learning Model Predictive Controller

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

$$J^{\mathrm{LMPC},j}_{t\to t+N}(x^j_t) = \min_{u^j_{t|t},\ldots,u^j_{t+N-1|t}} \sum_{k=t}^{t+N-1} h(x^j_{k|t}, u^j_{k|t}) + \boxed{V^{j-1}(x^j_{t+N|t}, x^+_{t+N})}$$

s.t.

$$x^j_{k+1|t} = A^j_{k|t} x^j_{k|t} + B^j_{k|t} u^j_{k|t} + C^j_{k|t}$$

$$x^j_{t|t} = x^j_t,$$

$$x^j_{k|t} \in \mathcal{X},\ \ u^j_{k|t} \in \mathcal{U},\ \ \forall k \in [t, \cdots, t+N-1]$$

$$x^j_{t+N|t} \in \boxed{\mathcal{CS}^{j-1}(x^+_{t+N})},$$

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)

$$J^{\mathrm{LMPC},j}_{t\to t+N}(x^j_t) = \min_{u^j_{t|t},\ldots,u^j_{t+N-1|t}} \sum_{k=t}^{t+N-1} h(x^j_{k|t}, u^j_{k|t}) + \boxed{V^{j-1}(x^j_{t+N|t}, x^+_{t+N})}$$

s.t.

$$x^j_{k+1|t} = A^j_{k|t} x^j_{k|t} + B^j_{k|t} u^j_{k|t} + C^j_{k|t}$$

$$x^j_{t|t} = x^j_t,$$

$$x^j_{k|t} \in \mathcal{X},\ u^j_{k|t} \in \mathcal{U},\ \forall k \in [t, \cdots, t+N-1]$$

$$x^j_{t+N|t} \in \boxed{\mathcal{CS}^{j-1}(x^+_{t+N})},$$

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)

$$J^{\mathrm{LMPC},j}_{t\rightarrow t+N}(x^j_t) = \min_{u^j_{t|t},\ldots,u^j_{t+N-1|t}} \sum_{k=t}^{t+N-1} h(x^j_{k|t}, u^j_{k|t}) + \boxed{V^{j-1}(x^j_{t+N|t}, x^+_{t+N})}$$

s.t.

$$\boxed{x^j_{k+1|t} = A^j_{k|t} x^j_{k|t} + B^j_{k|t} u^j_{k|t} + C^j_{k|t}}$$

$$x^j_{t|t} = x^j_t,$$

$$x^j_{k|t} \in \mathcal{X},\ u^j_{k|t} \in \mathcal{U},\ \forall k \in [t, \cdots, t+N-1]$$

$$x^j_{t+N|t} \in \boxed{\mathcal{CS}^{j-1}(x^+_{t+N})},$$

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)

$$J^{\mathrm{LMPC},j}_{t\rightarrow t+N}(x^j_t) = \min_{u^j_{t|t},\ldots,u^j_{t+N-1|t}} \sum_{k=t}^{t+N-1} h(x^j_{k|t}, u^j_{k|t}) + \boxed{V^{j-1}(x^j_{t+N|t}, x^+_{t+N})}$$

s.t.

$$\boxed{x^j_{k+1|t} = A^j_{k|t} x^j_{k|t} + B^j_{k|t} u^j_{k|t} + C^j_{k|t}}$$

$$x^j_{t|t} = x^j_t,$$

$$x^j_{k|t} \in \mathcal{X},\ \ u^j_{k|t} \in \mathcal{U},\ \ \forall k \in [t, \cdots, t+N-1]$$

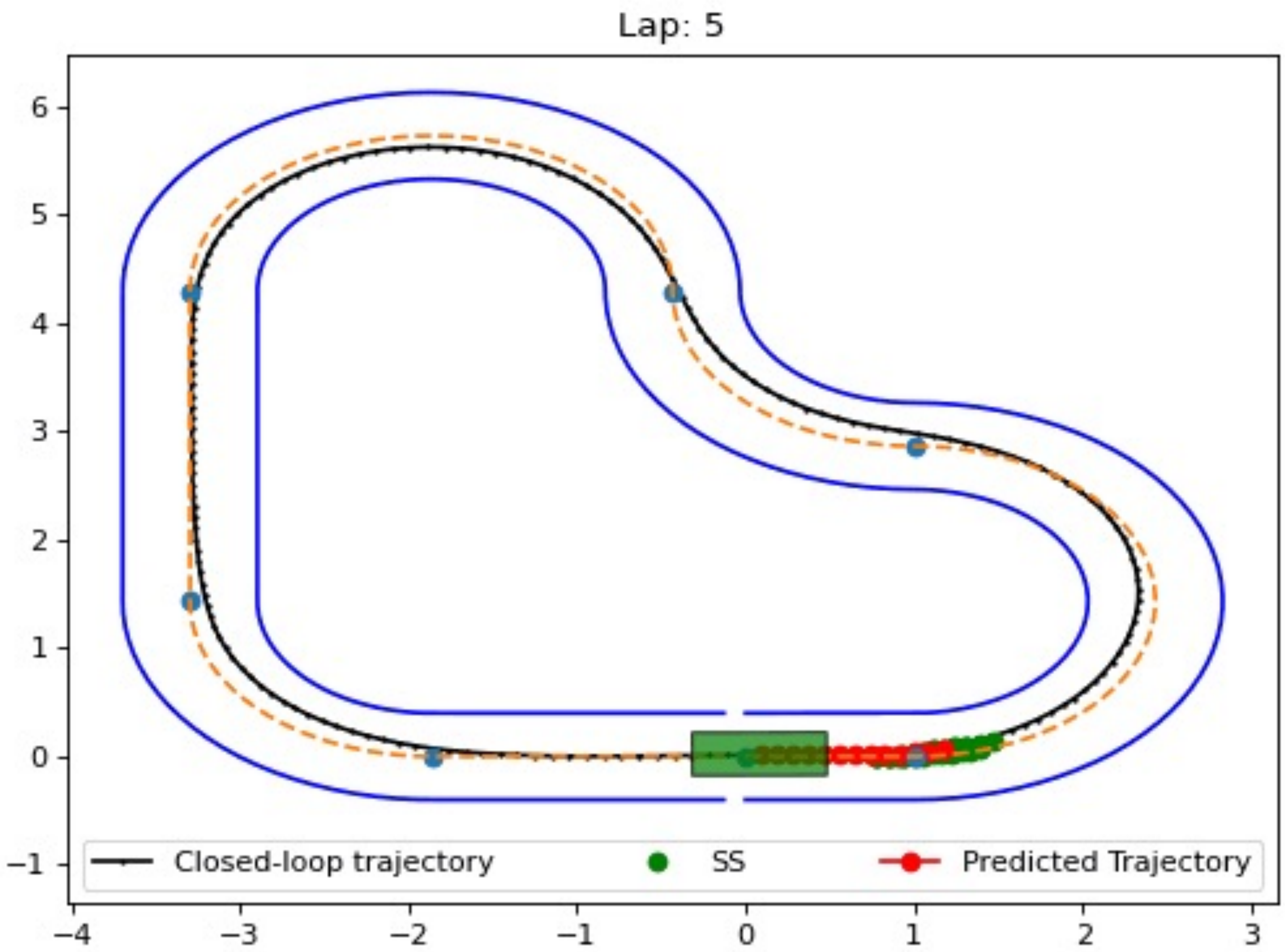
$$x^j_{t+N|t} \in \boxed{\mathcal{CS}^{j-1}(x^+_{t+N})},$$

Computed online

Computed online using LMPC strategy

Then apply to the system the control input $u^j_t = u^{*,j}_{t|t}$

Safe Set and Value Function Approximations



System ID in Autonomous Racing

- Nonlinear Dynamical System,

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

System ID in Autonomous Racing

- Nonlinear Dynamical System,

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

Kinematic Equations

System ID in Autonomous Racing

- Nonlinear Dynamical System,

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

Kinematic Equations

- Identifying the Dynamical System

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

Linearization around predicted trajectory

System ID in Autonomous Racing

- Nonlinear Dynamical System,

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

Dynamic Equations

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

Kinematic Equations

- Identifying the Dynamical System

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

Linearization around predicted trajectory

System ID in Autonomous Racing

- Nonlinear Dynamical System,

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

Dynamic Equations

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

Kinematic Equations

- Identifying the Dynamical System

Local Linear Regression

$$x^j_{k+1|t} = \begin{bmatrix} \dot{x}_{k+1|t} \\ \dot{y}_{k+1|t} \\ \ddot{\psi} \\ \psi_{k+1|t} \\ X_{k+1|t} \\ Y_{k+1|t} \end{bmatrix} = \begin{bmatrix} \multicolumn{3}{c}{\boxed{\arg\min \sum_{i,s} K(x^j_{k|t} - x^i_s)||\Lambda_y \begin{bmatrix} x^j_{k|t} \\ u^j_{k|t} \\ 1 \end{bmatrix} - y^i_{s+1}||, \forall y \in \{\dot{x}, \dot{y}, \ddot{\psi}\}}} \\ \boxed{\begin{matrix} \text{Linearized Kinematics} \\ \text{Linearized Kinematics} \\ \text{Linearized Kinematics} \end{matrix}} & x^j_{k|t} + & \boxed{\begin{matrix} \text{Linearized Kinematics} \\ \text{Linearized Kinematics} \\ \text{Linearized Kinematics} \end{matrix}} \end{bmatrix} \begin{bmatrix} u^j_{k|t} \\ 1 \end{bmatrix}$$

Linearization around predicted trajectory

System Identification – Design Steps

Identifying the Dynamical System

Local Linear Regression

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

Linearization around predicted trajectory

Implementation Details

System Identification – Design Steps

Identifying the Dynamical System

Local Linear Regression

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

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

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

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

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

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

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

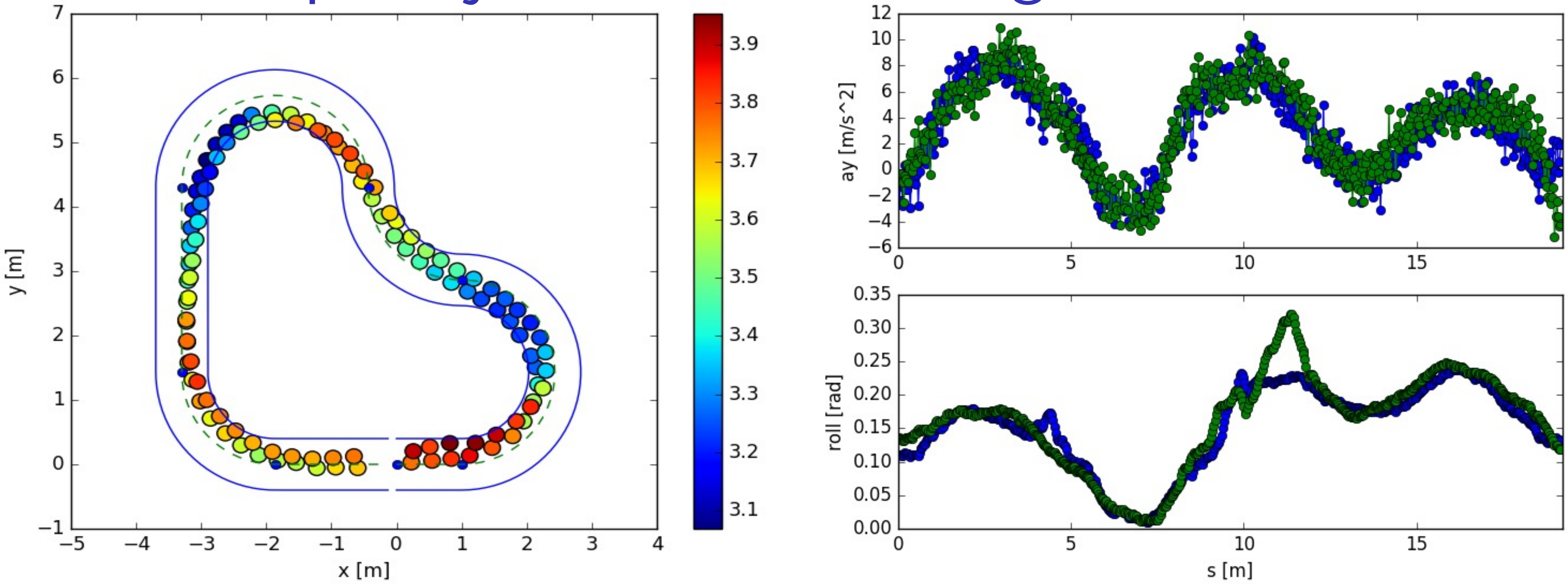
Linearization around predicted trajectory

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 Predictictive Control for Autonomous Racing

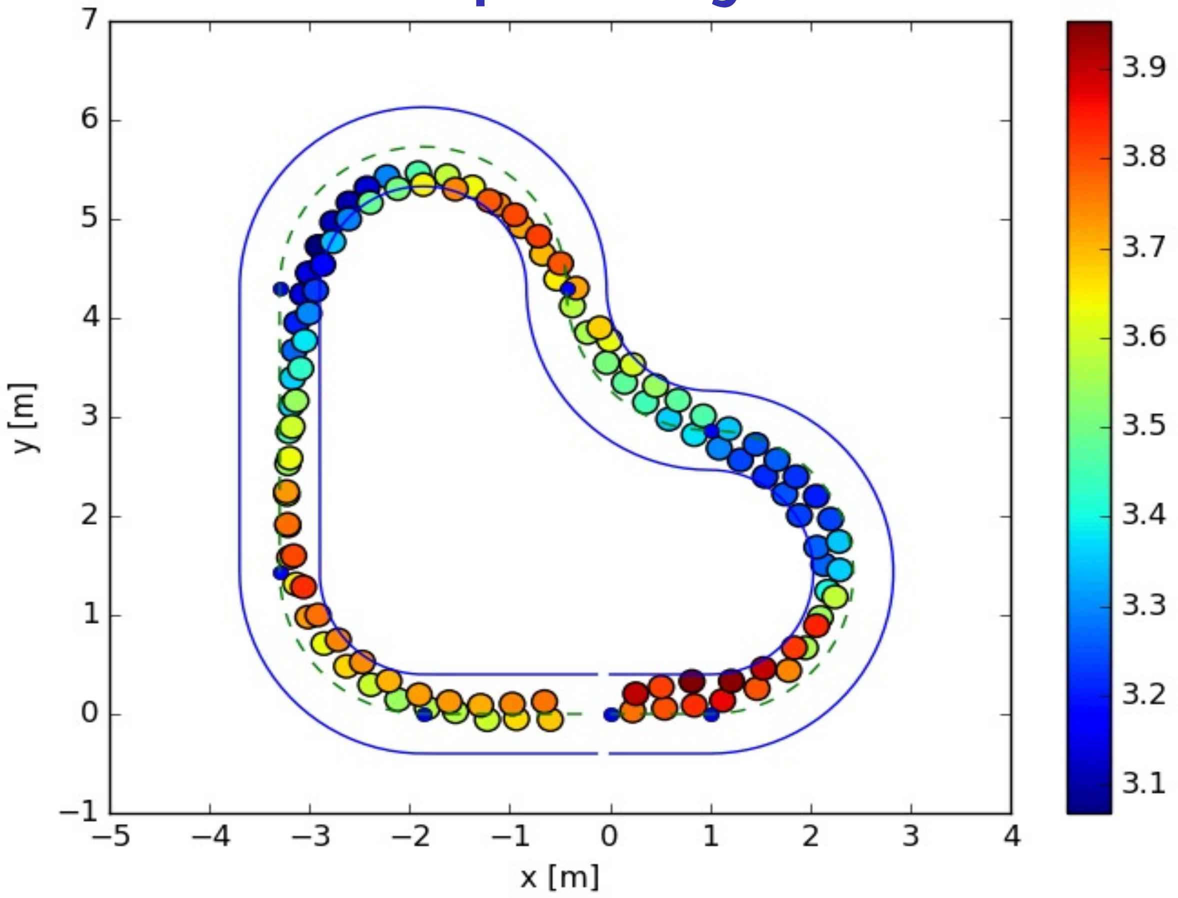
Closed-loop Trajectories at Convergence

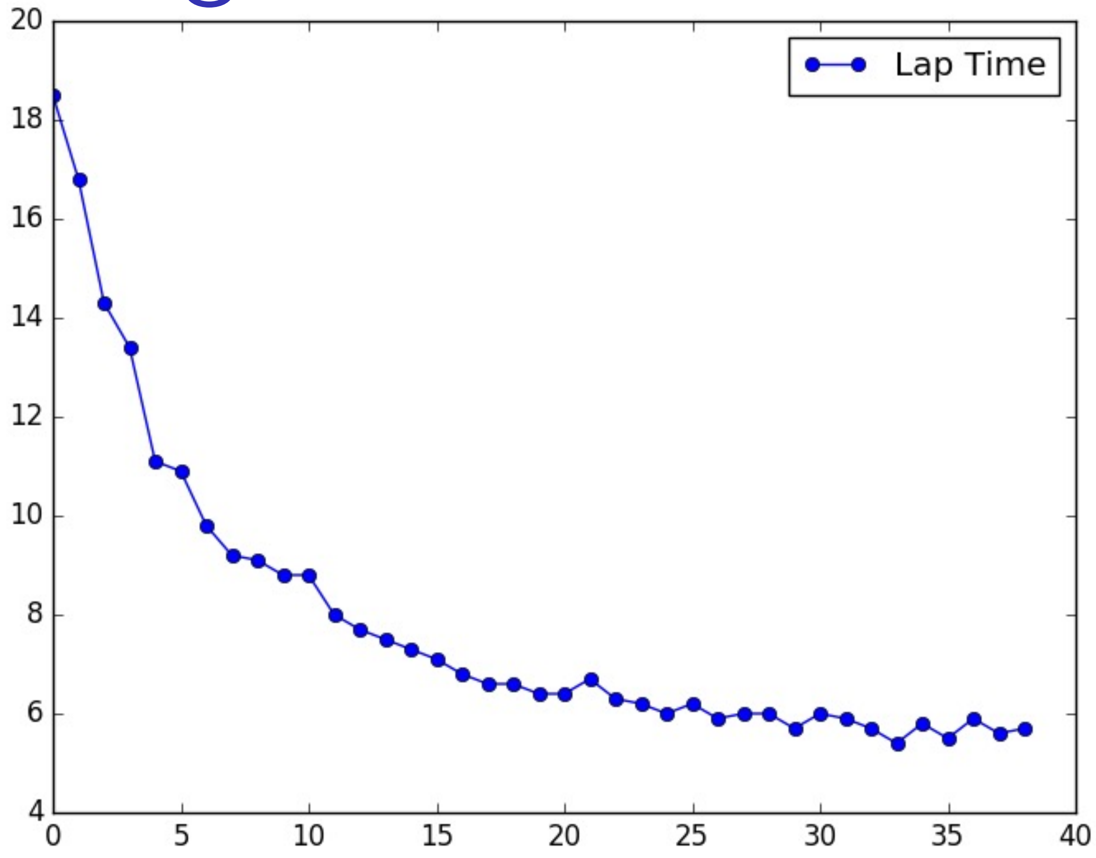


Remarks

- 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

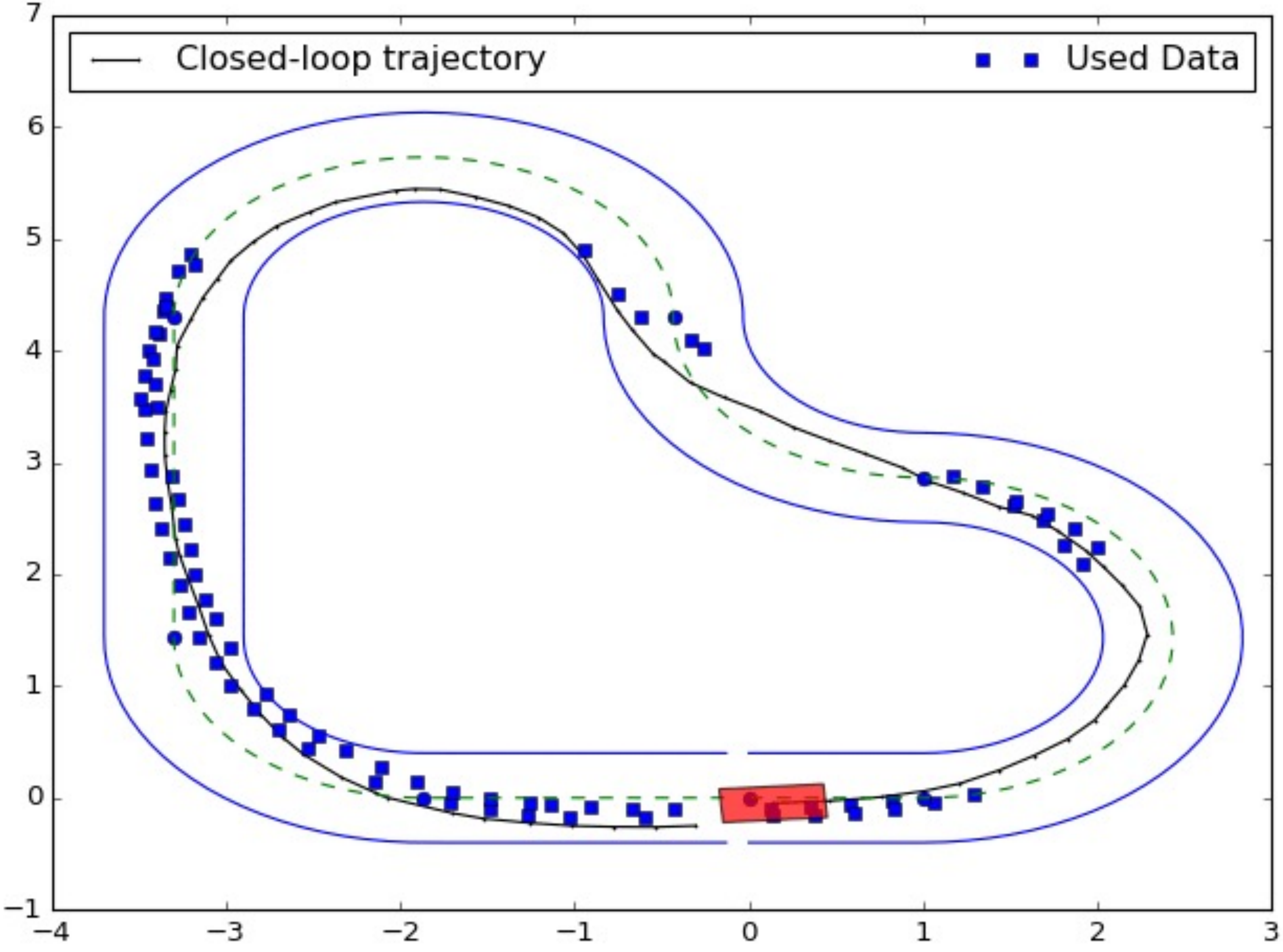




Remarks

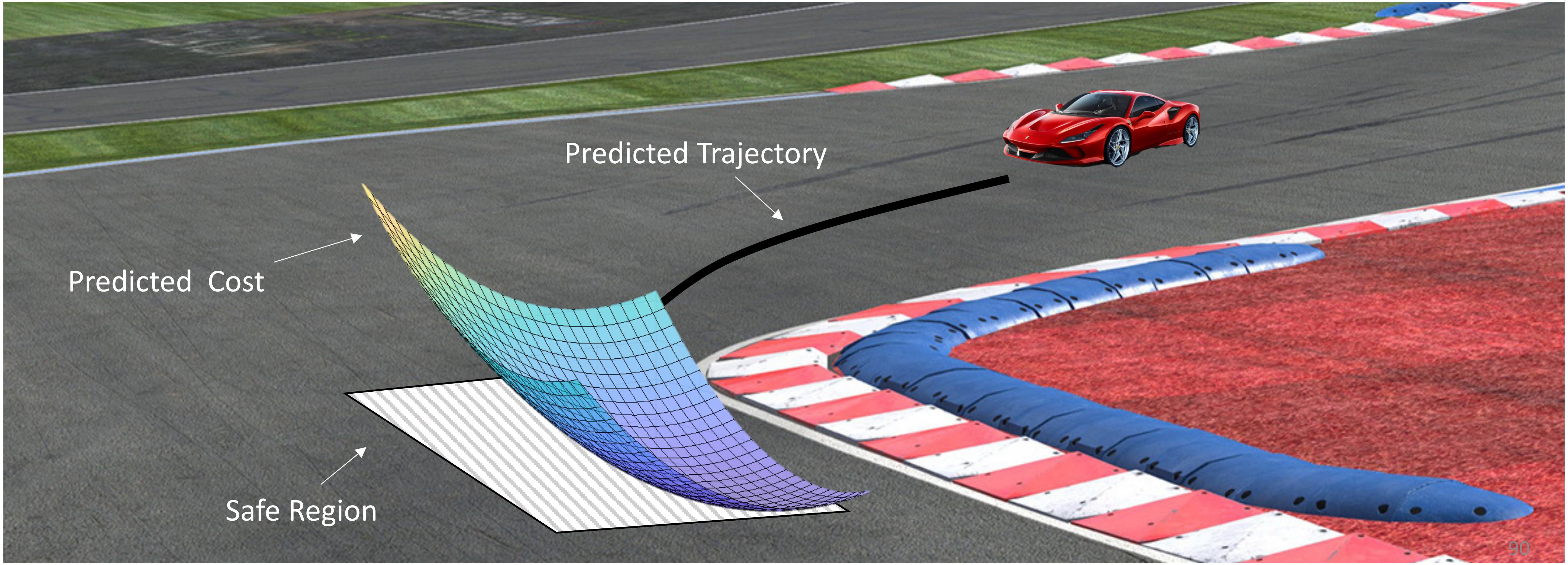
- A **reference trajectory is not needed** for the controller implementation
- The controller converges after few laps: the learning process is data efficient
- The **controller safely explores the state space** iteratively improving the lap time

Data Point Selection



The key components

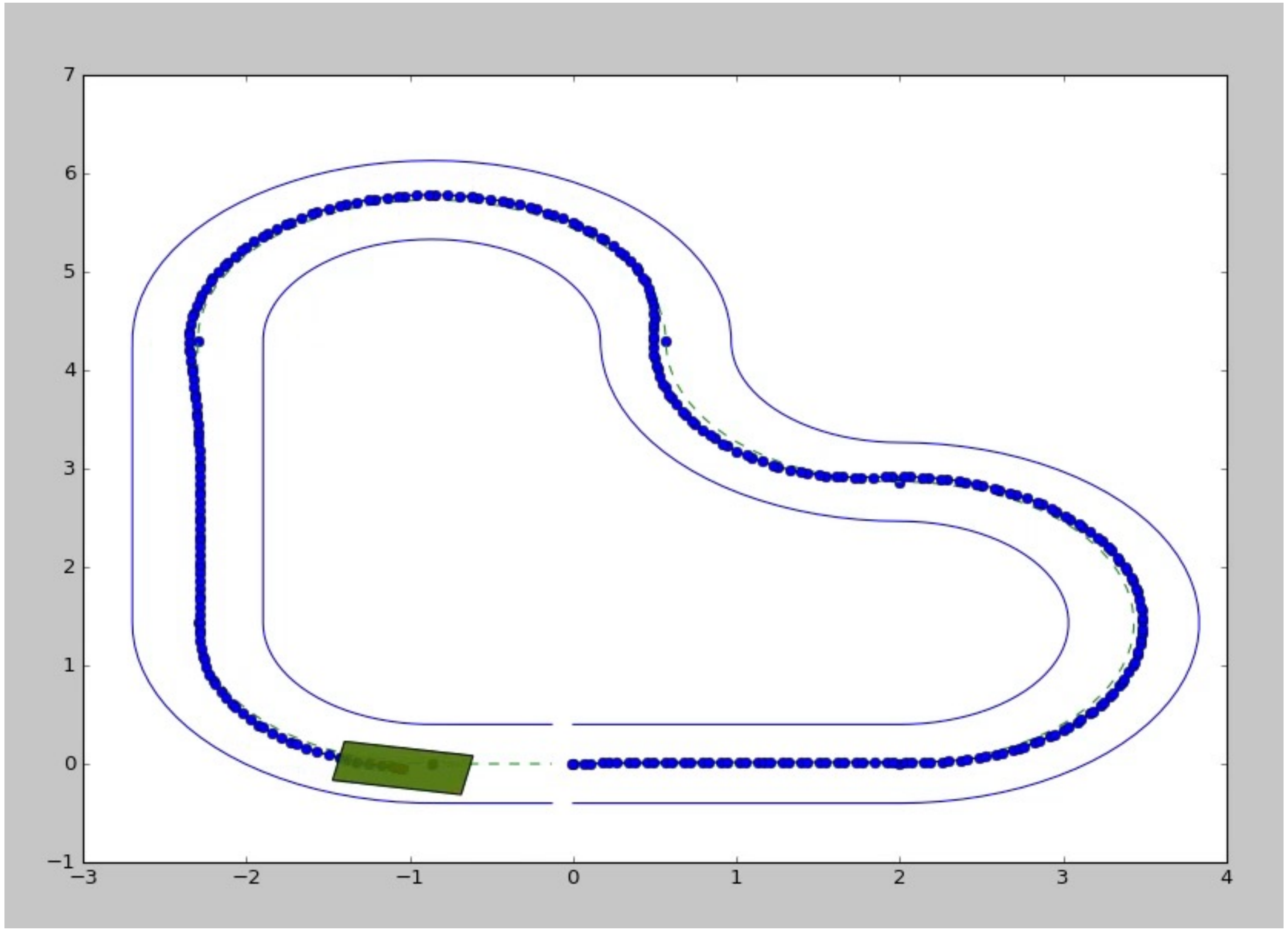
- Predicted trajectory given by prediction model
- Predicted cost estimated by value function approximation
- Safe region estimated by the safe set



Do you need the safe set? – Yes

LMPC without Invariant Set

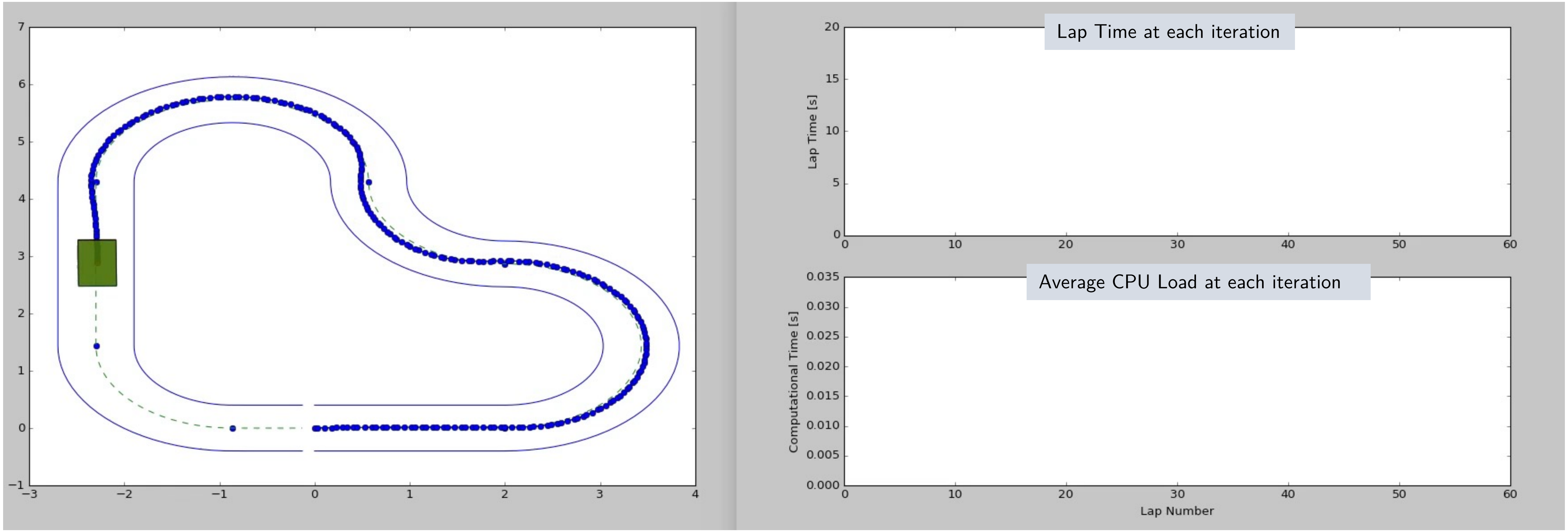
The controller extrapolates the value function on the Vx dimension



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



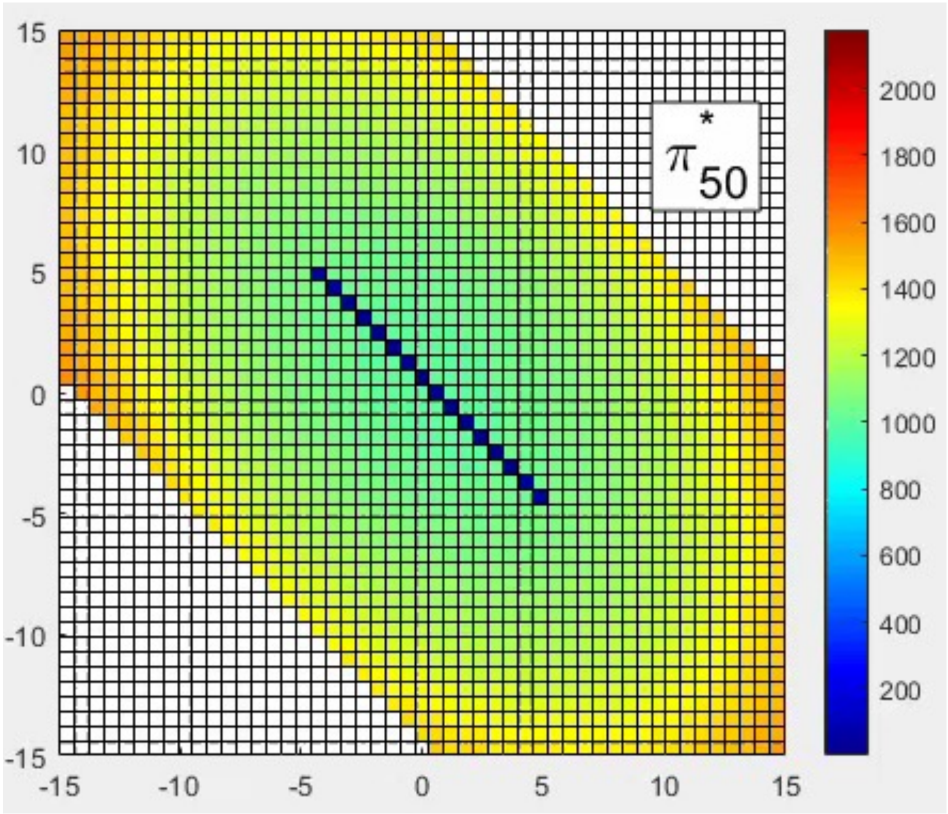
Comparison with Approximate DP (aka RL)

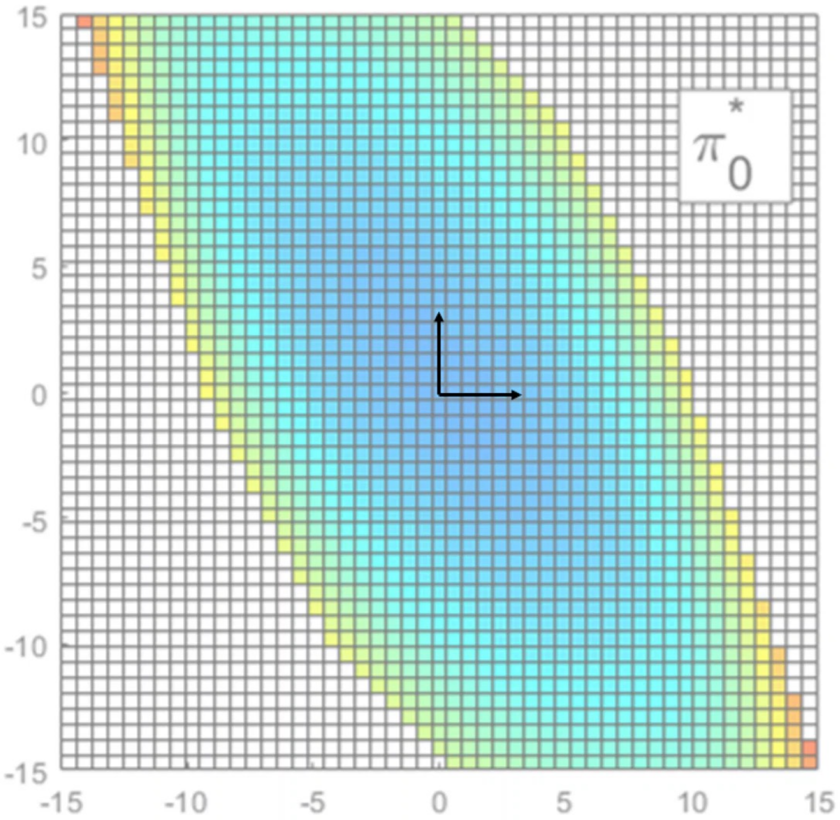
- Some references:
  - Bertsekas paper connecting MPC and ADP [1], books on RL and OC [2,3]
  - Lewis and Vrabie survey [4]
  - Recht survey [5]

- 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:

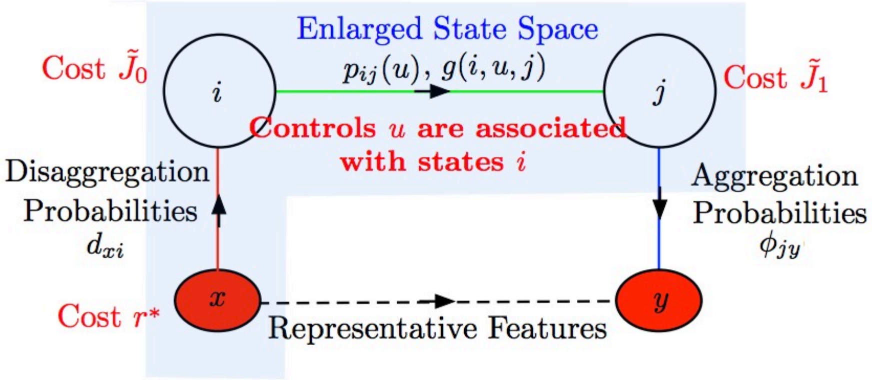
- Gridding, global properties
- Backward, one-step iteration

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

$$r_x^* = \sum_{i=1}^{n} d_{xi}\tilde{J}_0(i), \qquad x \in \mathcal{A},$$

$$\tilde{J}_0(i) = \min_{u \in U(i)} \sum_{j=1}^{n} p_{ij}(u)\big(g(i,u,j) + \alpha \tilde{J}_1(j)\big), \qquad i = 1,\dots,n,$$

$$\tilde{J}_1(j) = \sum_{y \in \mathcal{A}} \phi_{jy} r_y^*, \qquad j = 1,\dots,n$$

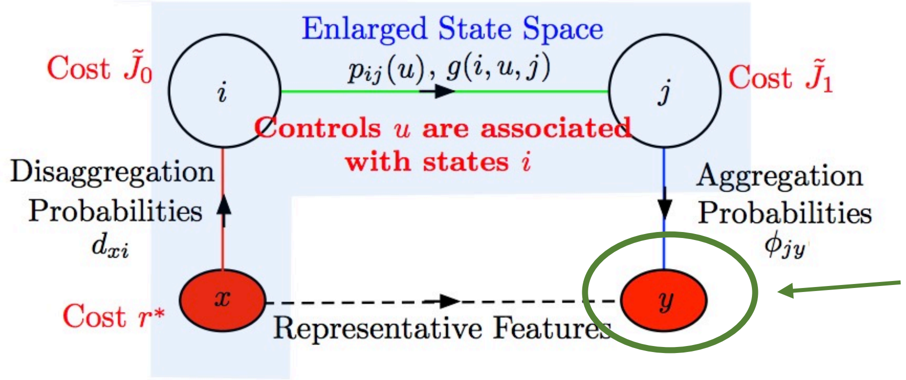
r^* solves uniquely the composite Bellman equation $r^* = Hr^*$:

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

Slide from Prof. D. Bertsekas class on aggregation, available at http://web.mit.edu/dimitrib/www/Slides_Lecture12_RLOC.pdf
Details on aggregation in Chapter 6 of “Reinforcement Learning and Optimal Control” by Prof. D. Bertsekas

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

$$r_x^* = \sum_{i=1}^{n} d_{xi} \tilde{J}_0(i), \qquad x \in \mathcal{A},$$

$$\tilde{J}_0(i) = \min_{u \in U(i)} \sum_{j=1}^{n} p_{ij}(u)\big(g(i,u,j) + \alpha \tilde{J}_1(j)\big), \qquad i = 1, \ldots, n,$$

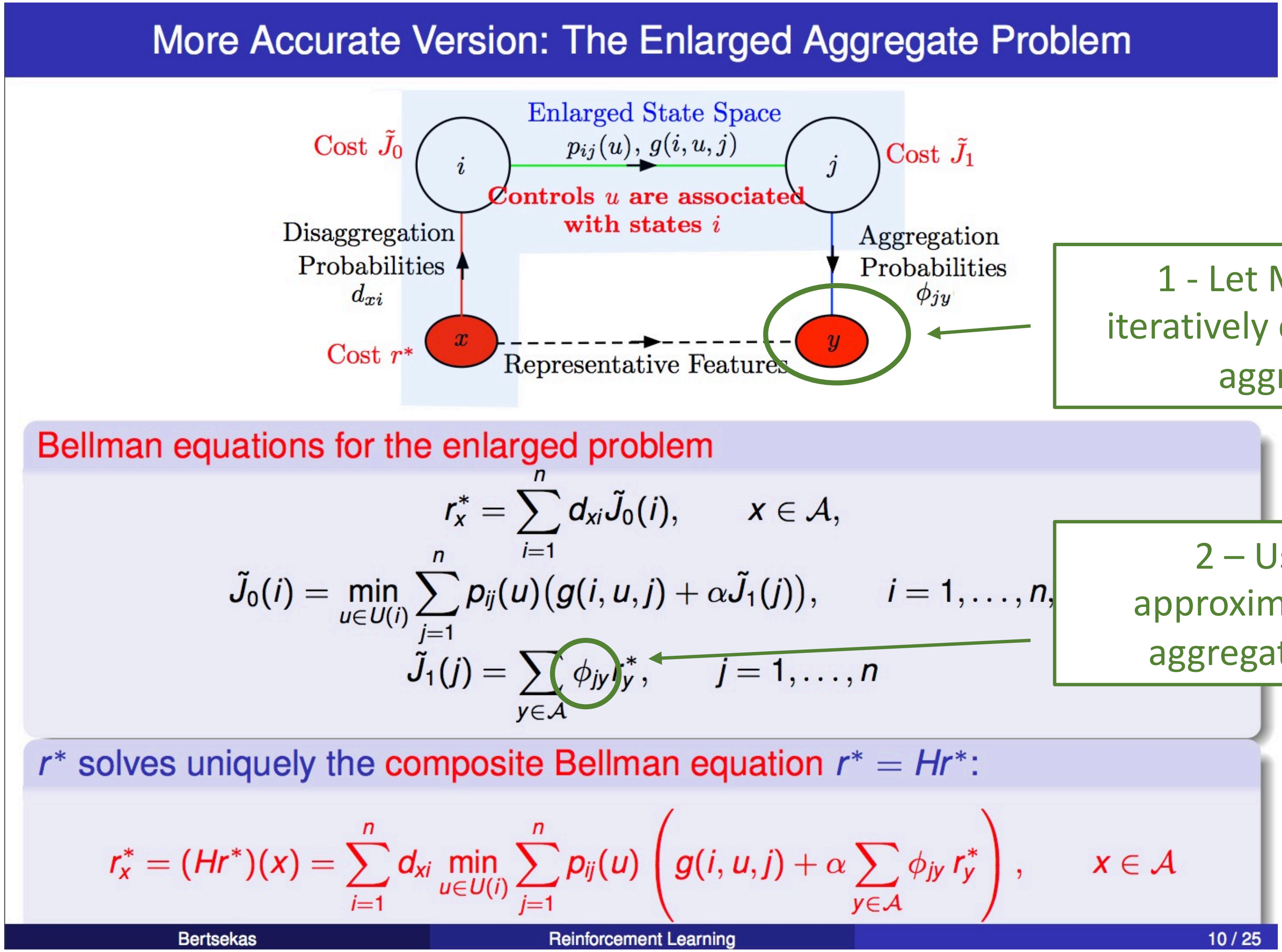
$$\tilde{J}_1(j) = \sum_{y \in \mathcal{A}} \phi_{jy} r_y^*, \qquad j = 1, \ldots, n$$

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

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

Slide from Prof. D. Bertsekas class on aggregation, available at http://web.mit.edu/dimitrib/www/Slides_Lecture12_RLOC.pdf
Details on aggregation in Chapter 6 of "Reinforcement Learning and Optimal Control" by Prof. D. Bertsekas

LMPC as Efficient and Safe Aggregated ADP



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

2 – Use barycentric approximation to compute aggregation probabilities

Slide from Prof. D. Bertsekas class on aggregation, available at http://web.mit.edu/dimitrib/www/Slides_Lecture12_RLOC.pdf
Details on aggregation in Chapter 6 of "Reinforcement Learning and Optimal Control" by Prof. D. Bertsekas

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

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

Gray Box

known

unknown

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

The complexity of the prediction model

Linear

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

Gray Box

known

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

known

unknown

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

Black Box

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

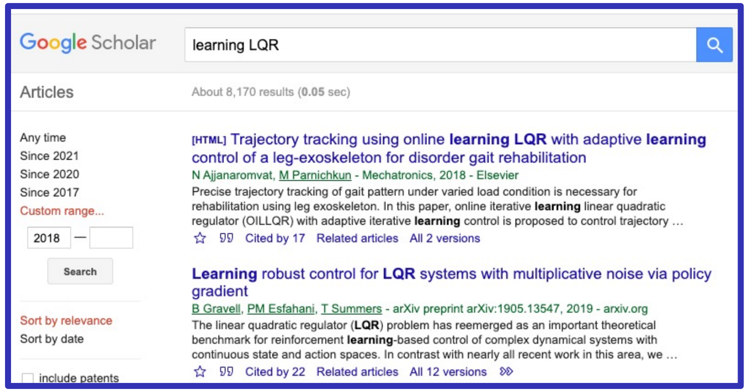
The complexity of the prediction model

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

known

unknown

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

Black Box

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

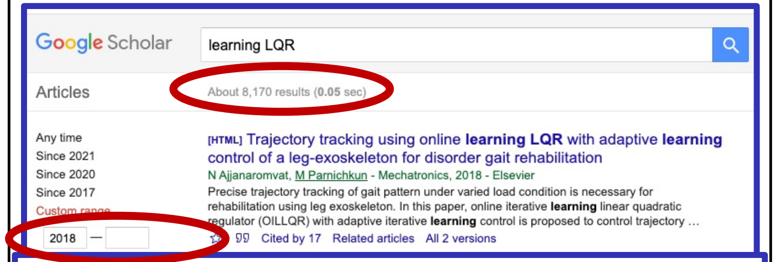
The complexity of the prediction model

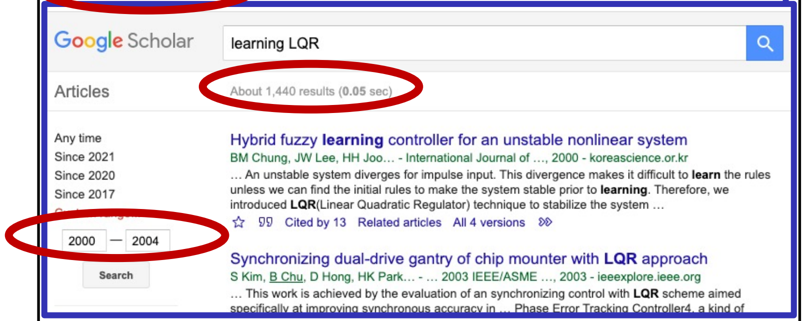
Linear

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

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

known

unknown

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

Black Box

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

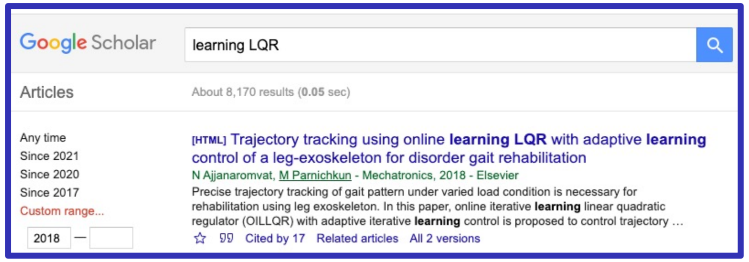
The complexity of the prediction model

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.



Enrico Terzi, Lorenzo Fagiano, Marcello Farina, Riccardo Scattolini. Learning multi-step prediction models for receding horizon control. In2018 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

known unknown

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

Black Box

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

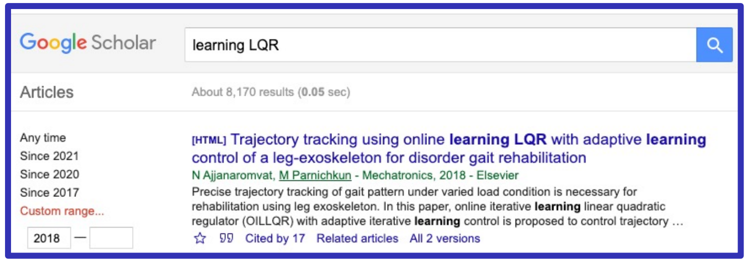
The complexity of the prediction model

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.



Enrico Terzi, Lorenzo Fagiano, Marcello Farina, Riccardo Scattolini. Learning multi-step prediction models for receding horizon control. In2018 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

known 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

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

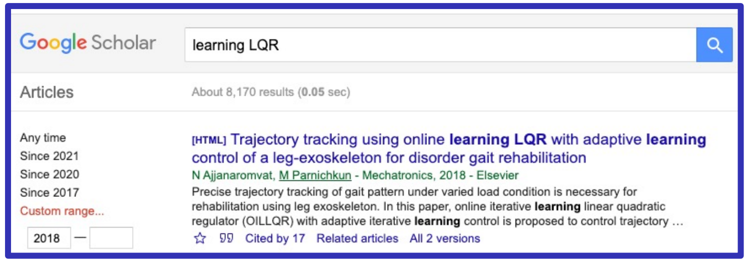
The complexity of the prediction model

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. In2018 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

known 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).

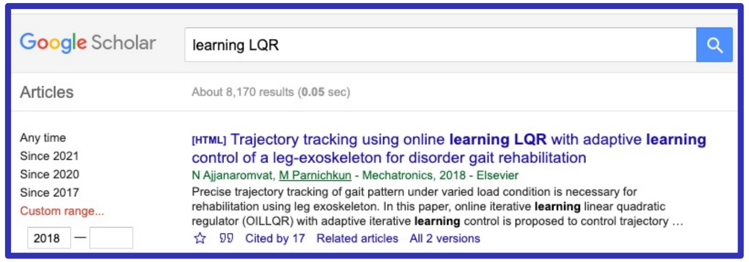
The complexity of the prediction model

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. In2018 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

known 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).

Learning Linear Models: One Step Prediction

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

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

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

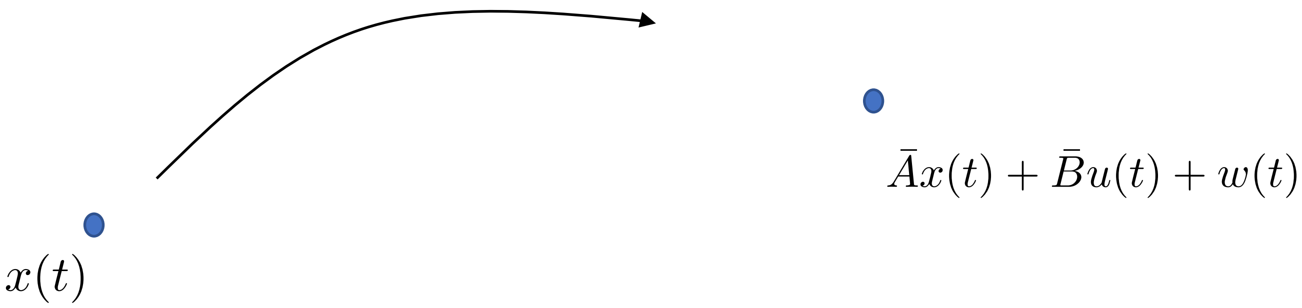
$x(t)$

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

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

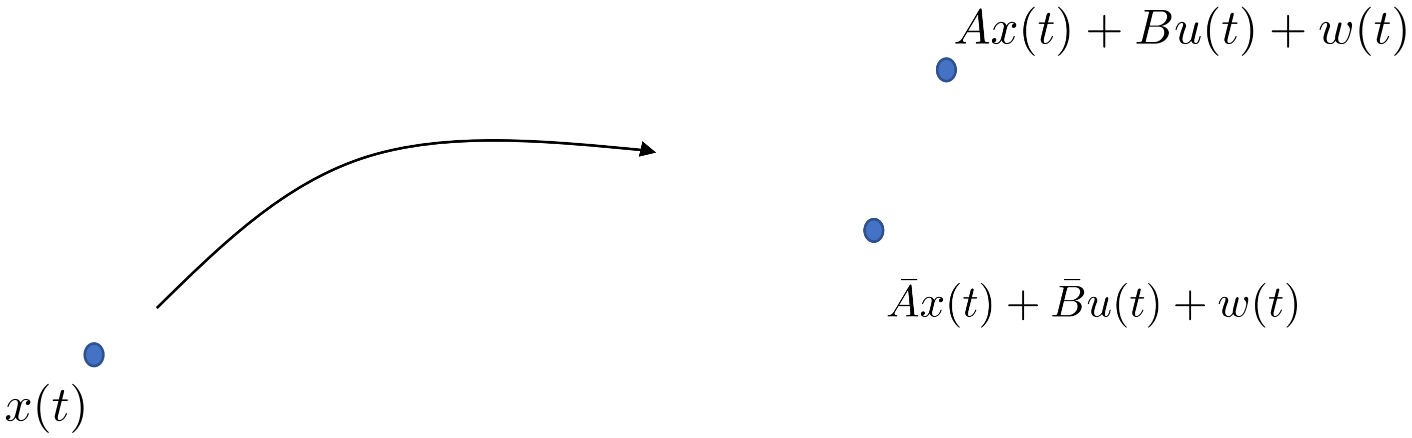


Marco C., Campi, and Erik Weyer. "Finite sample properties of system identification methods." *IEEE Transactions on Automatic Control* 47.8 (2002): 1329-1334.

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

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

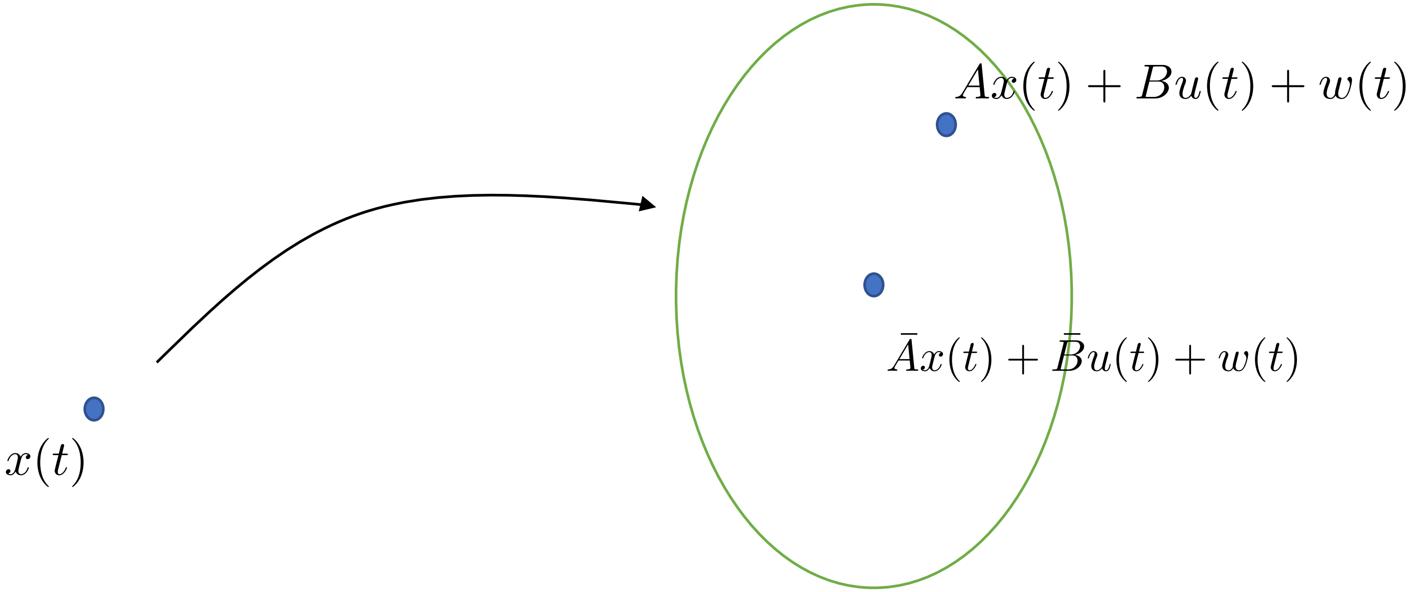


Marco C., Campi, and Erik Weyer. "Finite sample properties of system identification methods." *IEEE Transactions on Automatic Control* 47.8 (2002): 1329-1334.

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

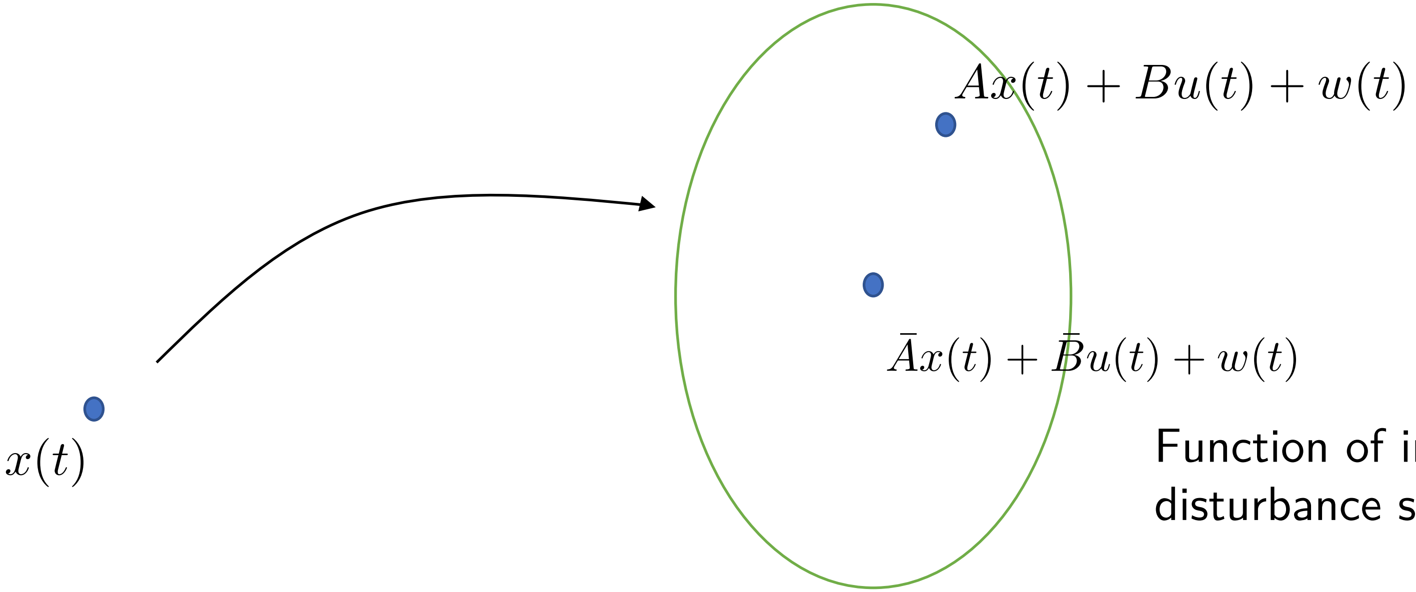
$$\bar{A}, \bar{B} = \arg\min_{A,B} \sum_{k=0}^{T-1} \sum_{i=1}^{M} ||x_{k+1}^{(i)} - Ax_k^{(i)} - Bu_k^{(i)}||_2^2$$



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

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



Function of initial condition, applied input, disturbance support and the identification error:

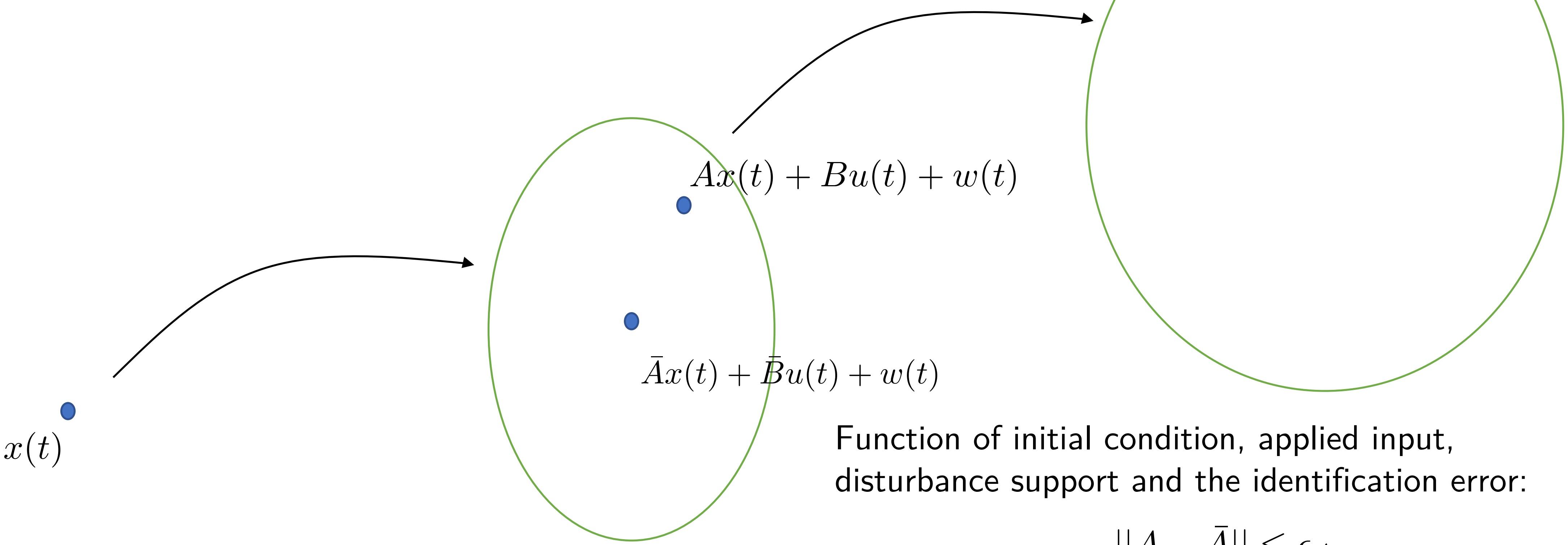
$$||A - \bar{A}|| \leq \epsilon_A$$
$$||B - \bar{B}|| \leq \epsilon_B$$

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

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



Function of initial condition, applied input, disturbance support and the identification error:

$$||A - \bar{A}|| \leq \epsilon_A$$
$$||B - \bar{B}|| \leq \epsilon_B$$

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

Enrico Terzi, Lorenzo Fagiano, Marcello Farina, Riccardo Scattolini. Learning multi-step prediction models for receding horizon control. In2018 European Control Conference (ECC) 2018 Jun 12 (pp. 1335-1340). IEEE.

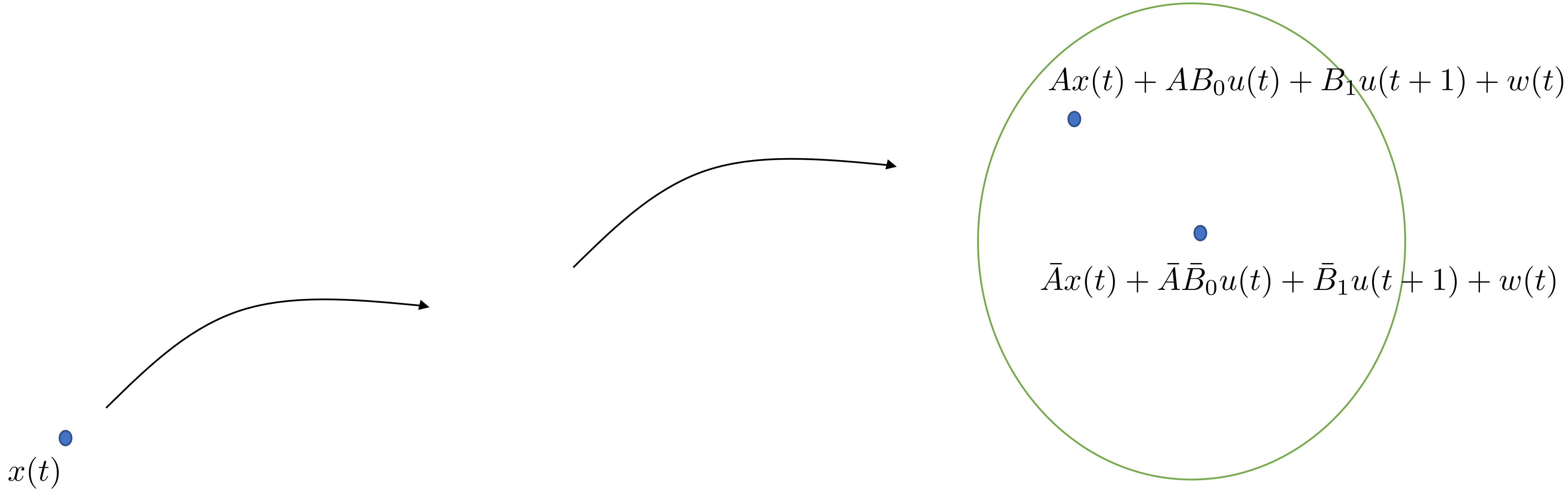
Learning Linear Models: Multi-Step Prediction

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

Enrico Terzi, Lorenzo Fagiano, Marcello Farina, Riccardo Scattolini. Learning multi-step prediction models for receding horizon control. In2018 European Control Conference (ECC) 2018 Jun 12 (pp. 1335-1340). IEEE.

Learning Linear Models: Multi-Step Prediction

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

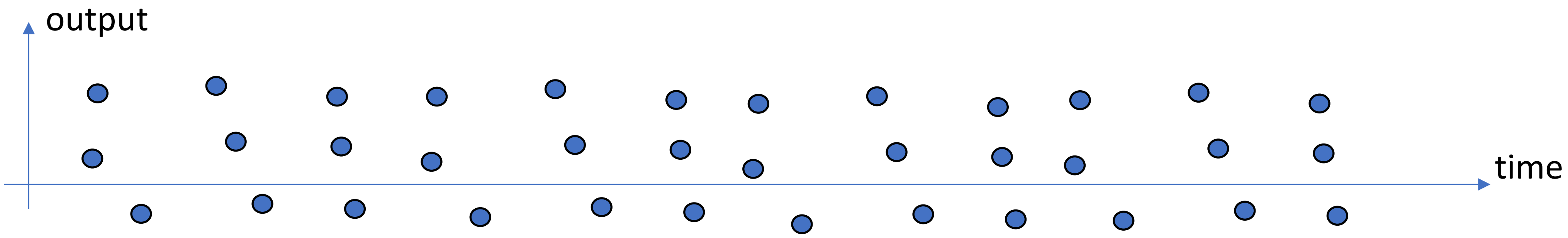


Enrico Terzi, Lorenzo Fagiano, Marcello Farina, Riccardo Scattolini. Learning multi-step prediction models for receding horizon control. In2018 European Control Conference (ECC) 2018 Jun 12 (pp. 1335-1340). IEEE.

Learning Linear Models: Data-Based Prediction

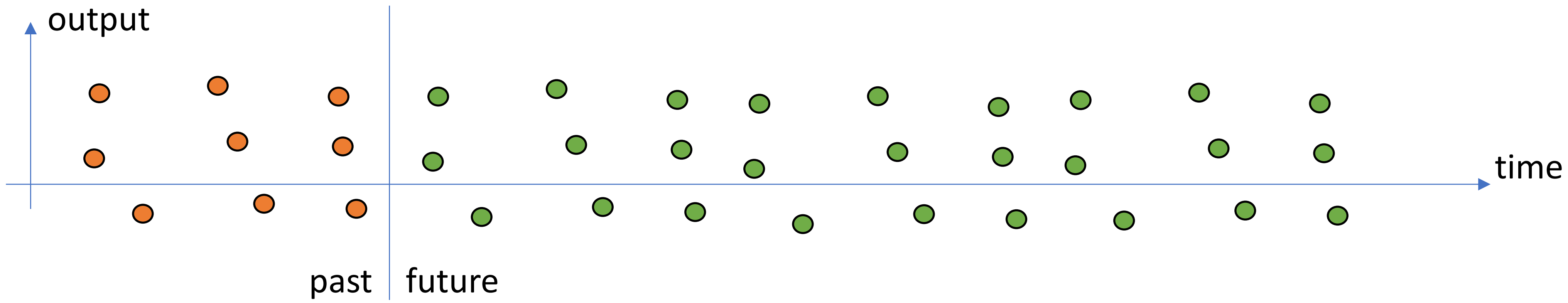
Jeremy Coulson, John Lygeros, and Florian Dörfler. "Data-enabled predictive control: In the shallows of the DeePC." *2019 18th European Control Conference (ECC)*. IEEE, 2019.

Learning Linear Models: Data-Based Prediction



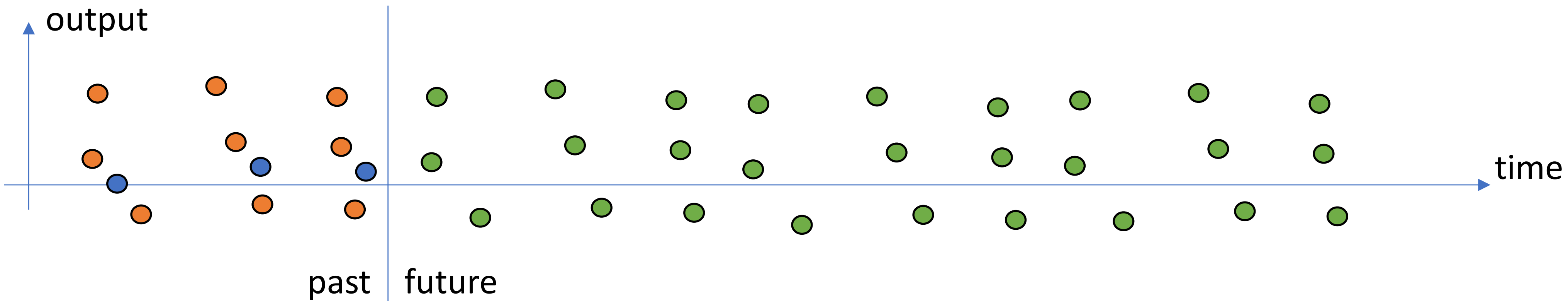
Jeremy Coulson, John Lygeros, and Florian Dörfler. "Data-enabled predictive control: In the shallows of the DeePC." *2019 18th European Control Conference (ECC)*. IEEE, 2019.

Learning Linear Models: Data-Based Prediction



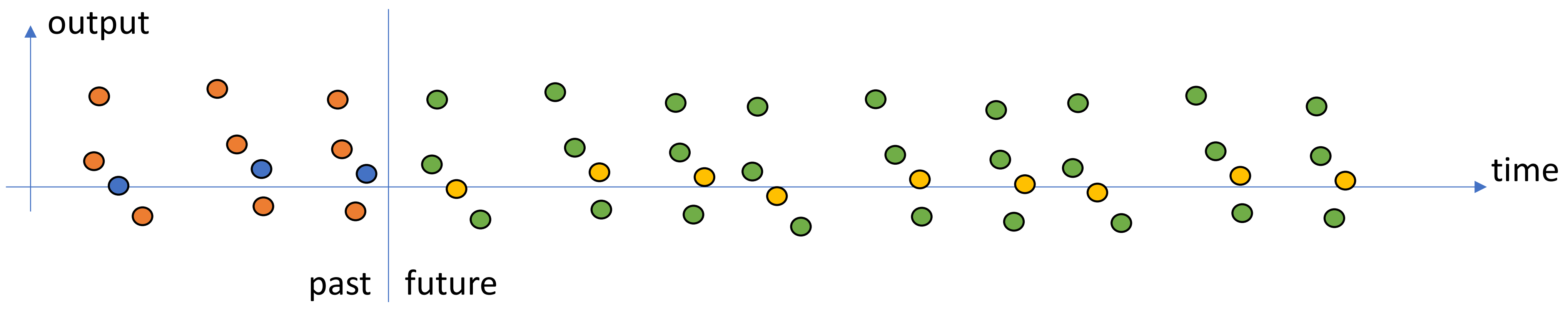
Jeremy Coulson, John Lygeros, and Florian Dörfler. "Data-enabled predictive control: In the shallows of the DeePC." *2019 18th European Control Conference (ECC)*. IEEE, 2019.

Learning Linear Models: Data-Based Prediction



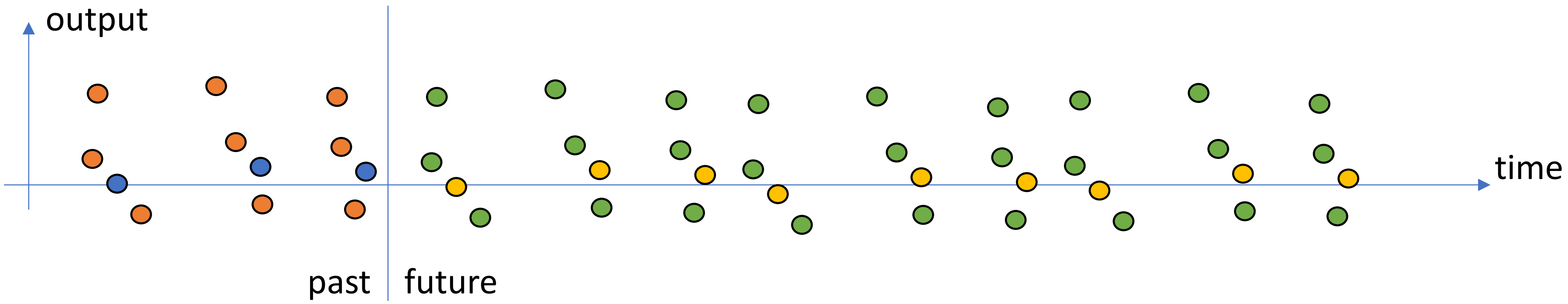
Jeremy Coulson, John Lygeros, and Florian Dörfler. "Data-enabled predictive control: In the shallows of the DeePC." *2019 18th European Control Conference (ECC)*. IEEE, 2019.

Learning Linear Models: Data-Based Prediction



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.

Learning Linear Models: Data-Based Prediction



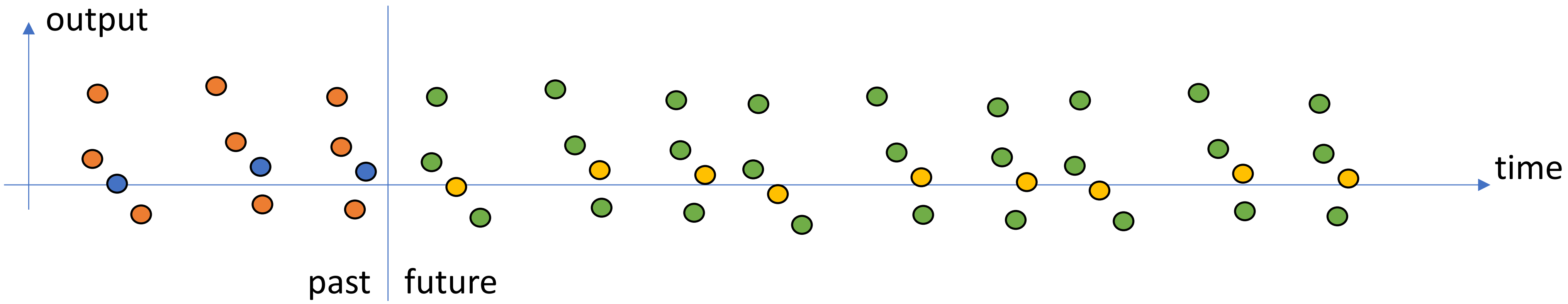
Construct the Hankel Matrix using Data

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

$$\begin{pmatrix} U_\mathrm{p} \\ U_\mathrm{f} \end{pmatrix} := \mathscr{H}_{T_\mathrm{ini}+N}(u^\mathrm{d}), \quad \begin{pmatrix} Y_\mathrm{p} \\ Y_\mathrm{f} \end{pmatrix} := \mathscr{H}_{T_\mathrm{ini}+N}(y^\mathrm{d}),$$

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.

Learning Linear Models: Data-Based Prediction



Construct the Hankel Matrix using Data

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

$$\begin{pmatrix} U_\mathrm{p} \\ U_\mathrm{f} \end{pmatrix} := \mathscr{H}_{T_\mathrm{ini}+N}(u^\mathrm{d}), \quad \begin{pmatrix} Y_\mathrm{p} \\ Y_\mathrm{f} \end{pmatrix} := \mathscr{H}_{T_\mathrm{ini}+N}(y^\mathrm{d}),$$

Replace the model

$$\begin{aligned} \underset{g,u,y}{\text{minimize}} \quad & \sum_{k=0}^{N-1} \left(\|y_k - r_{t+k}\|_Q^2 + \|u_k\|_R^2 \right) \\ \text{subject to} \quad & \begin{pmatrix} U_\mathrm{p} \\ Y_\mathrm{p} \\ U_\mathrm{f} \\ Y_\mathrm{f} \end{pmatrix} g = \begin{pmatrix} u_\mathrm{ini} \\ y_\mathrm{ini} \\ u \\ y \end{pmatrix}, \\ & u_k \in \mathcal{U}, \ \forall k \in \{0, \dots, N-1\}, \\ & y_k \in \mathcal{Y}, \ \forall k \in \{0, \dots, N-1\}. \end{aligned}$$

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.

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.

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. In2018 European Control Conference (ECC) 2018 Jun 12 (pp. 1335-1340). IEEE.

Gray Box

known

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

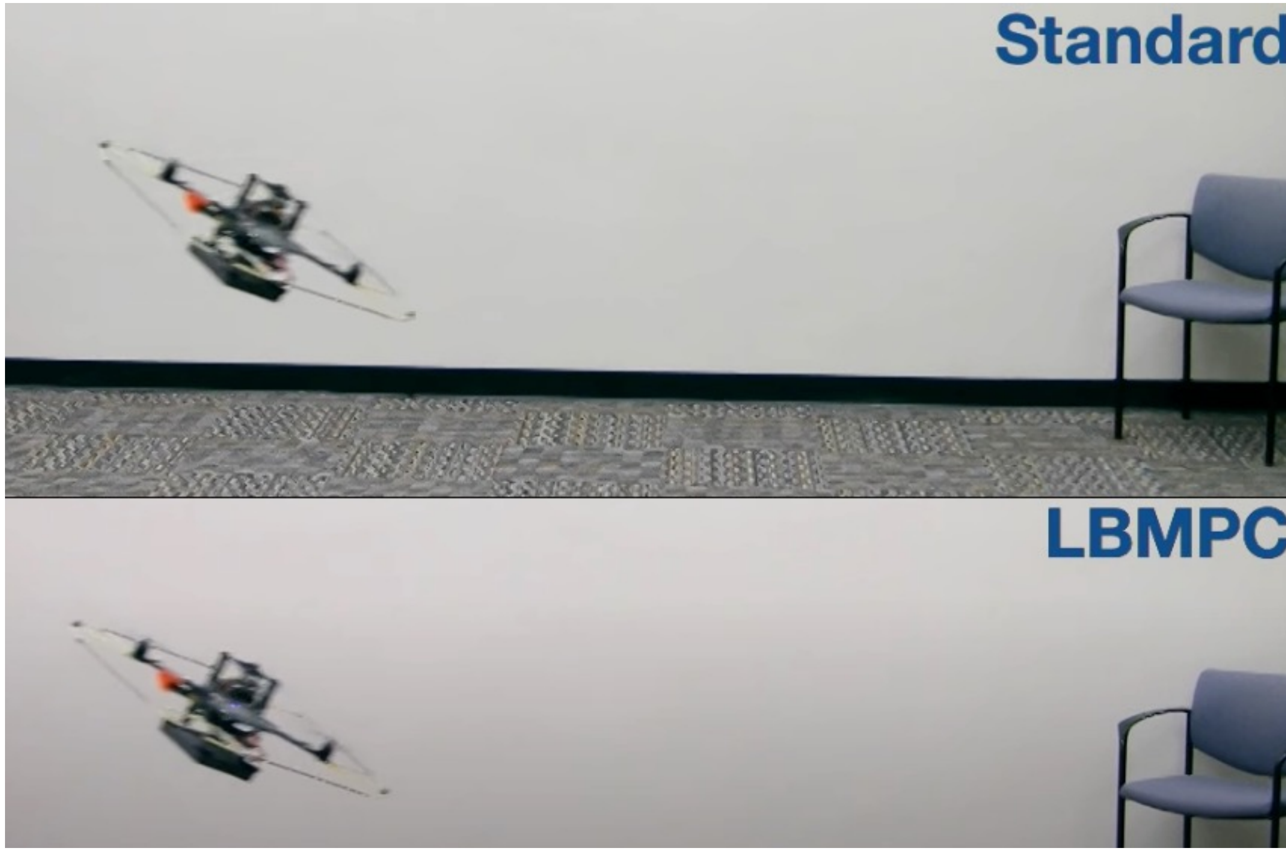
Most popular method is Gaussian Process

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

Gray Box Model

Most popular method is Gaussian Process

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



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.

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. In2018 European Control Conference (ECC) 2018 Jun 12 (pp. 1335-1340). IEEE.

Gray Box

known 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).

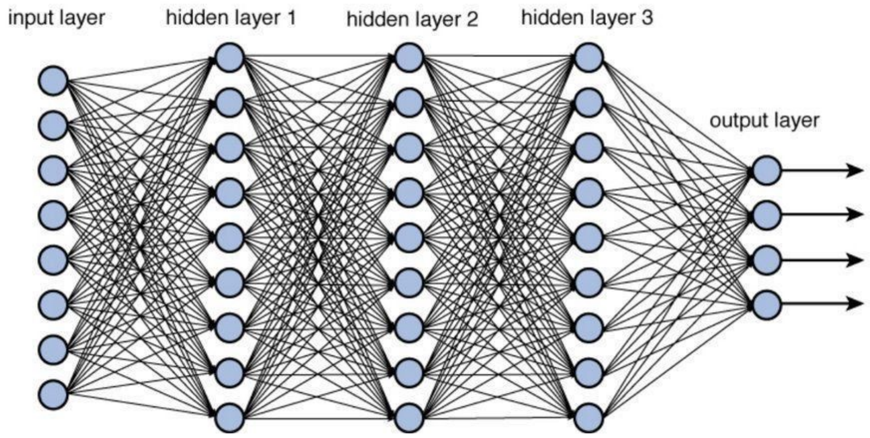
Anusha Nagabandi, Gregory Kahn, Ronald S. Fearing, and Sergey Levine. "Neural network dynamics for model-based deep reinforcement learning with model-free fine-tuning." In 2018 IEEE International Conference on Robotics and Automation (ICRA), pp. 7559-7566. IEEE, 2018.

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

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



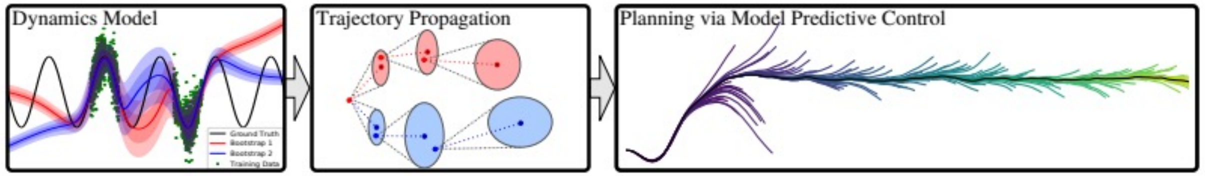
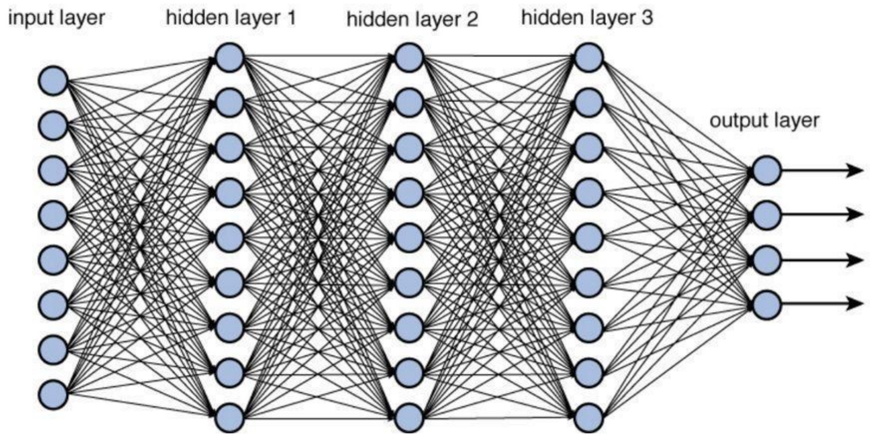


Figure 1: Our method (PE-TS): **Model**: Our probabilistic ensemble (PE) dynamics model is shown as an ensemble of two bootstraps (bootstrap disagreement far from data captures epistemic uncertainty: our subjective uncertainty due to a lack of data), each a probabilistic neural network that captures aleatoric uncertainty (inherent variance of the observed data). **Propagation**: Our trajectory sampling (TS) propagation technique uses our dynamics model to re-sample each particle (with associated bootstrap) according to its probabilistic prediction at each point in time, up until horizon T. **Planning**: At each time step, our MPC algorithm computes an optimal action sequence, applies the first action in the sequence, and repeats until the task-horizon.

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

Black Box Model



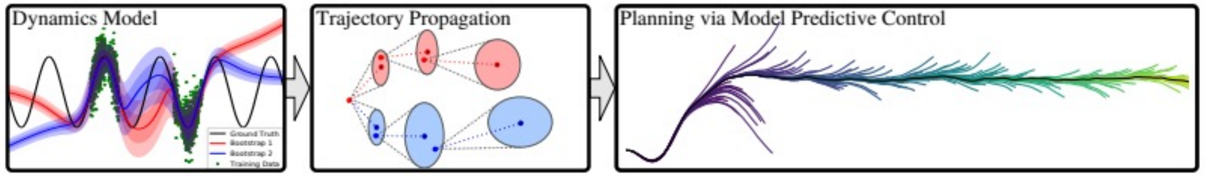
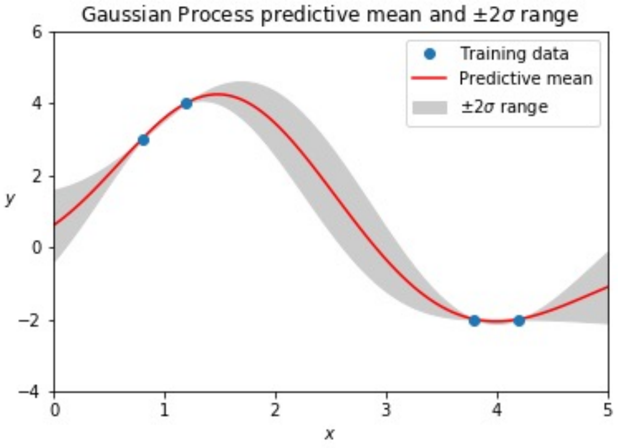
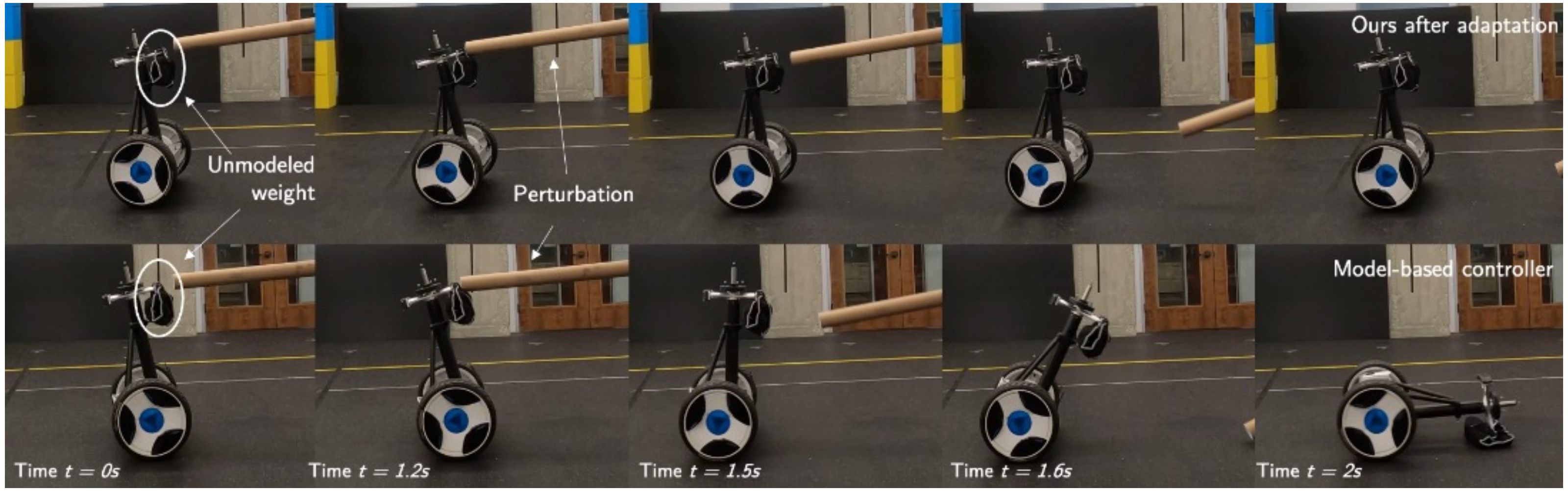


Figure 1: Our method (PE-TS): **Model**: Our probabilistic ensemble (PE) dynamics model is shown as an ensemble of two bootstraps (bootstrap disagreement far from data captures epistemic uncertainty: our subjective uncertainty due to a lack of data), each a probabilistic neural network that captures aleatoric uncertainty (inherent variance of the observed data). **Propagation**: Our trajectory sampling (TS) propagation technique uses our dynamics model to re-sample each particle (with associated bootstrap) according to its probabilistic prediction at each point in time, up until horizon T. **Planning**: At each time step, our MPC algorithm computes an optimal action sequence, applies the first action in the sequence, and repeats until the task-horizon.





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

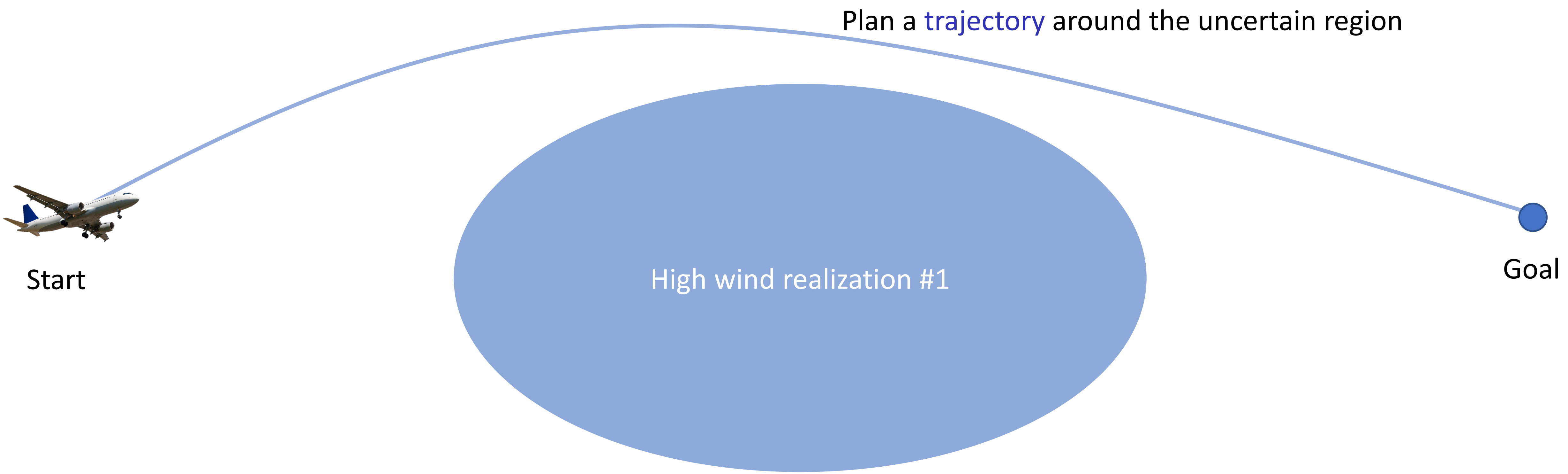
Ivan D. Jimenez, Rodriguez, Ugo Rosolia, Aaron D. Ames, and Yisong Yue. "Learning Unstable Dynamics with One Minute of Data: A Differentiation-based Gaussian Process Approach." *arXiv preprint arXiv:2103.04548* (2021).

Planning under uncertainty

Optimizing over control policies

Why is planning in uncertain environments harder?

Why is planning in uncertain environments harder?

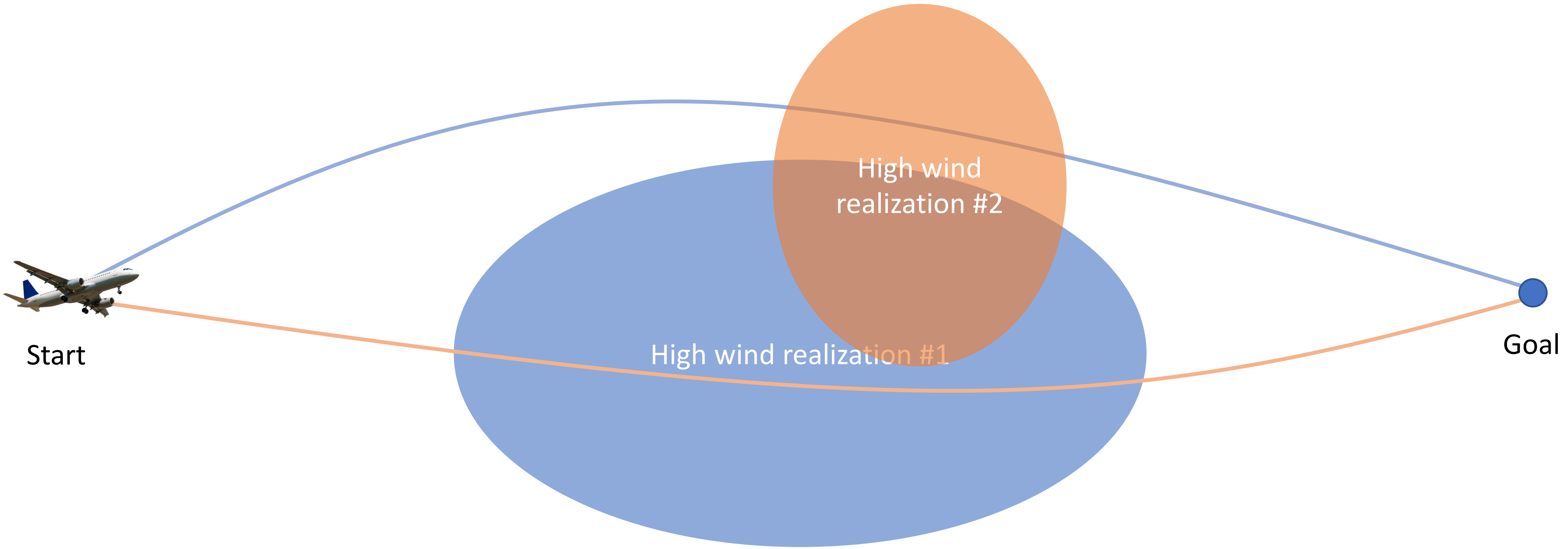


Why is planning in uncertain environments harder?

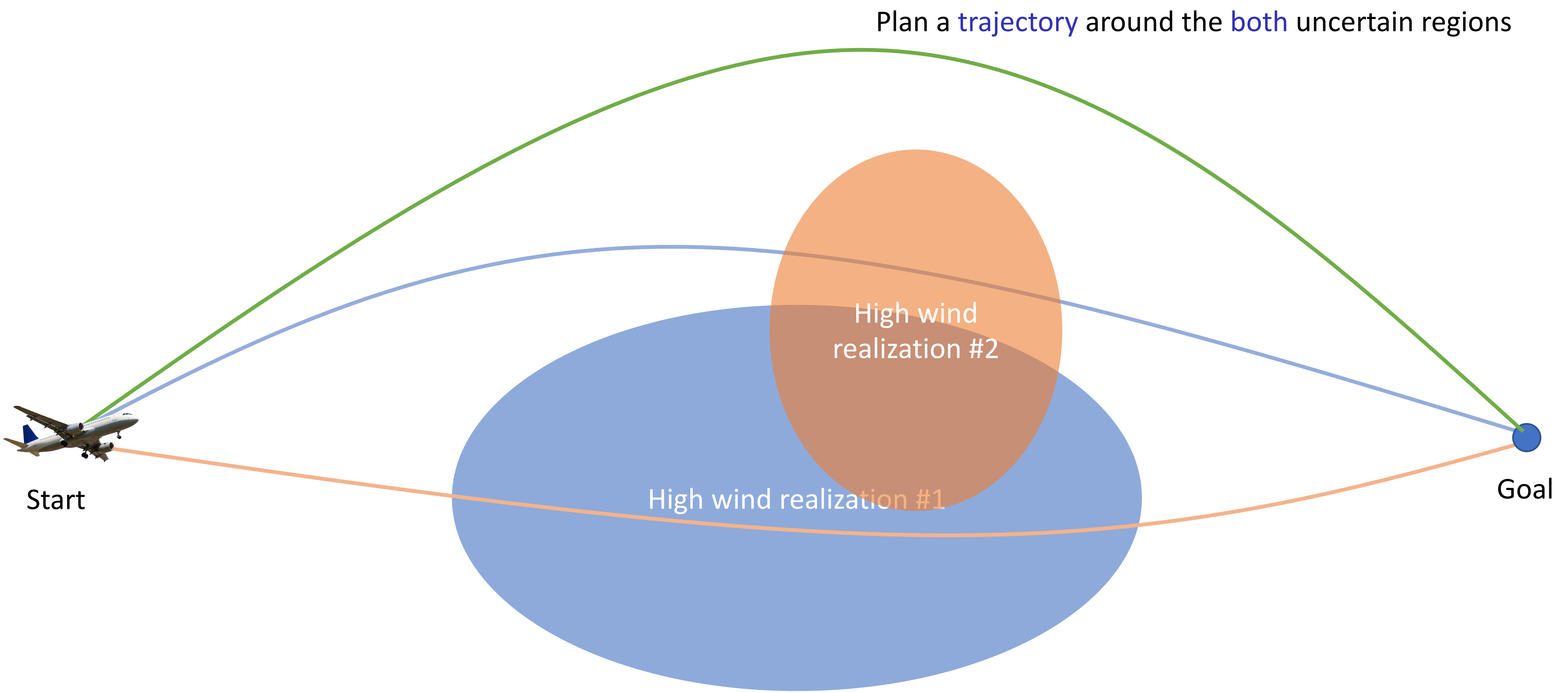
Why is planning in uncertain environments harder?

Plan a trajectory around the uncertain region

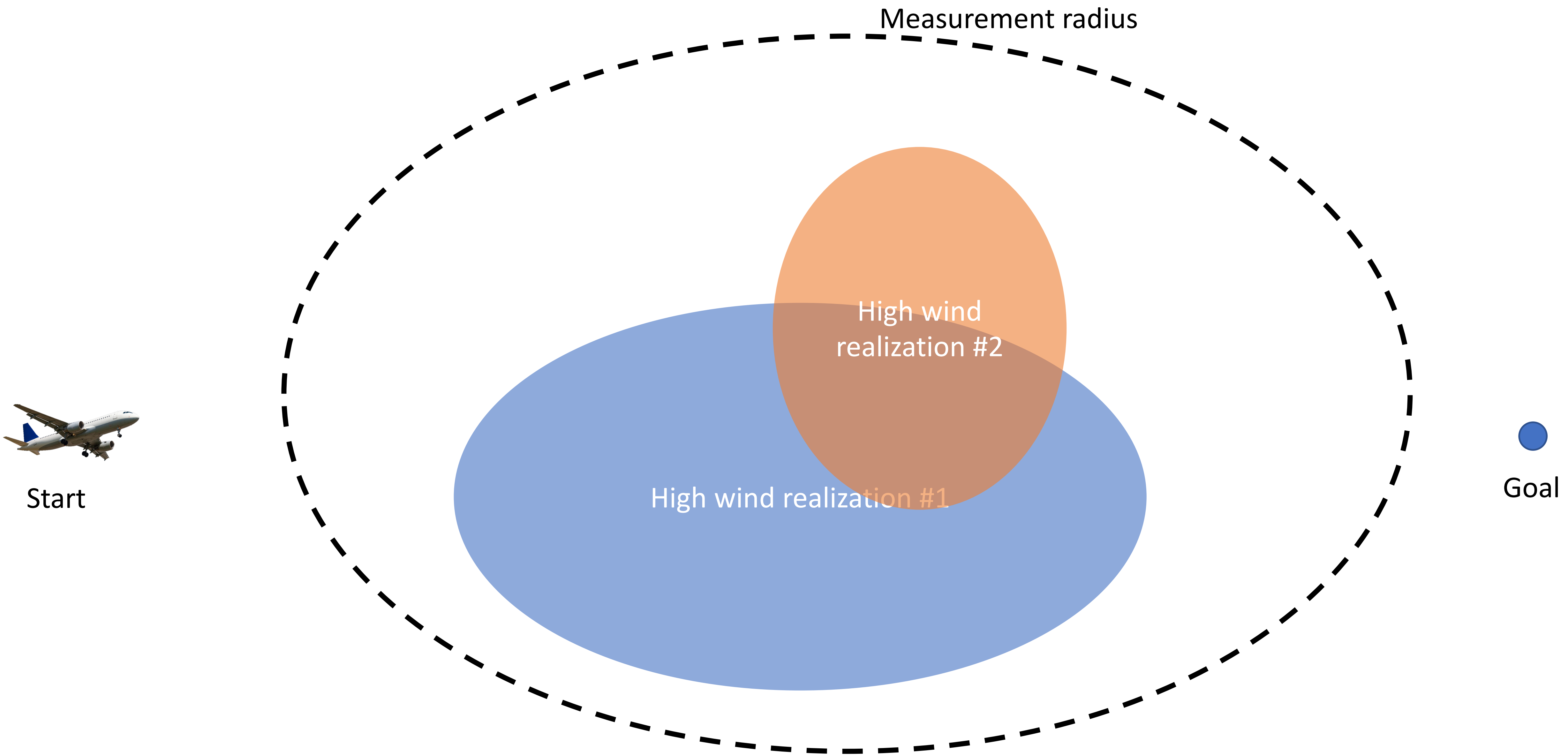
Why is planning in uncertain environments harder?



Why is planning in uncertain environments harder?

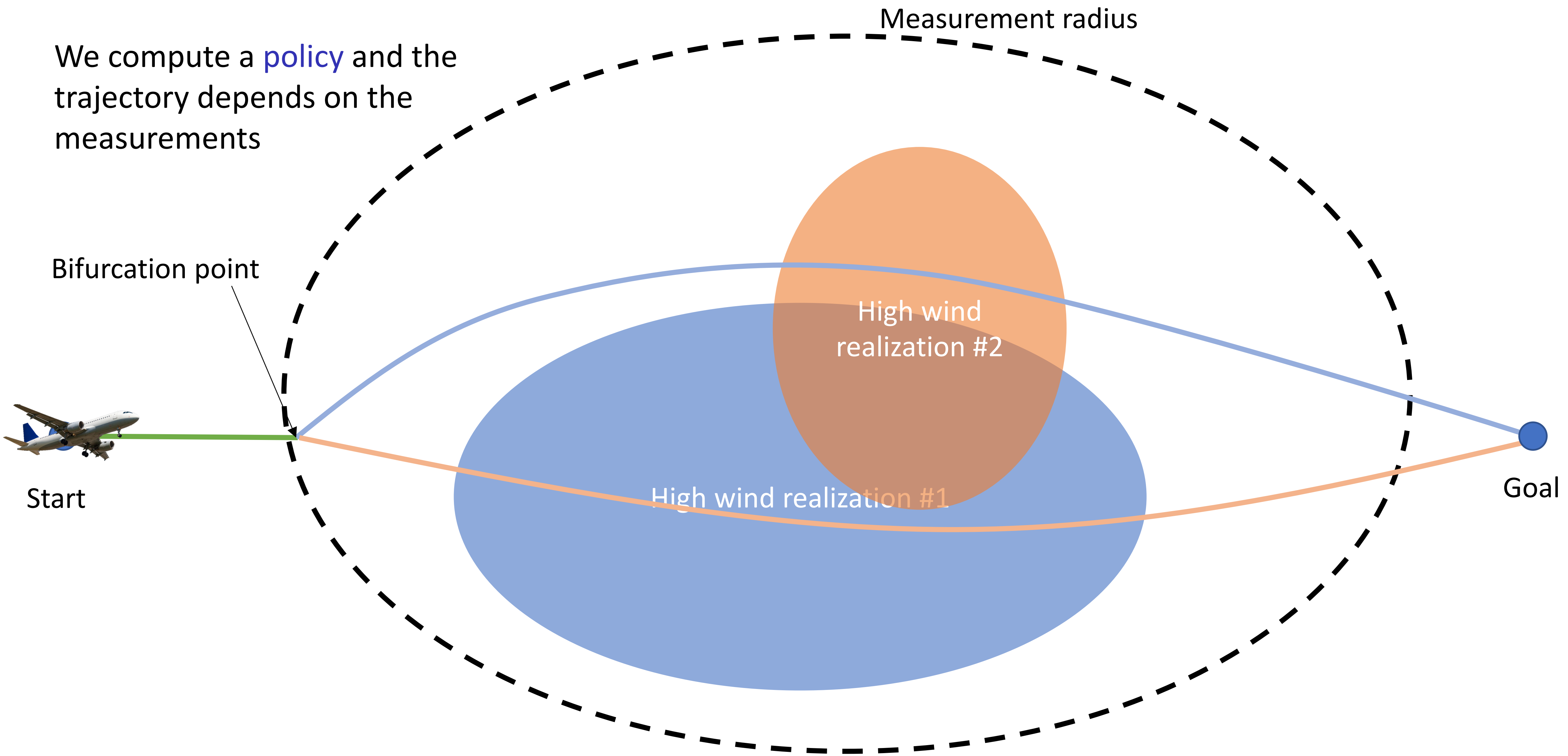


Why is planning in uncertain environments harder?



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 $\pi : \mathbb{R}^n \to \mathbb{R}^d$ 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 $\pi : \mathbb{R}^n \to \mathbb{R}^d$ 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: $x_{k+1} = Ax_k + Bu_k + w_k$

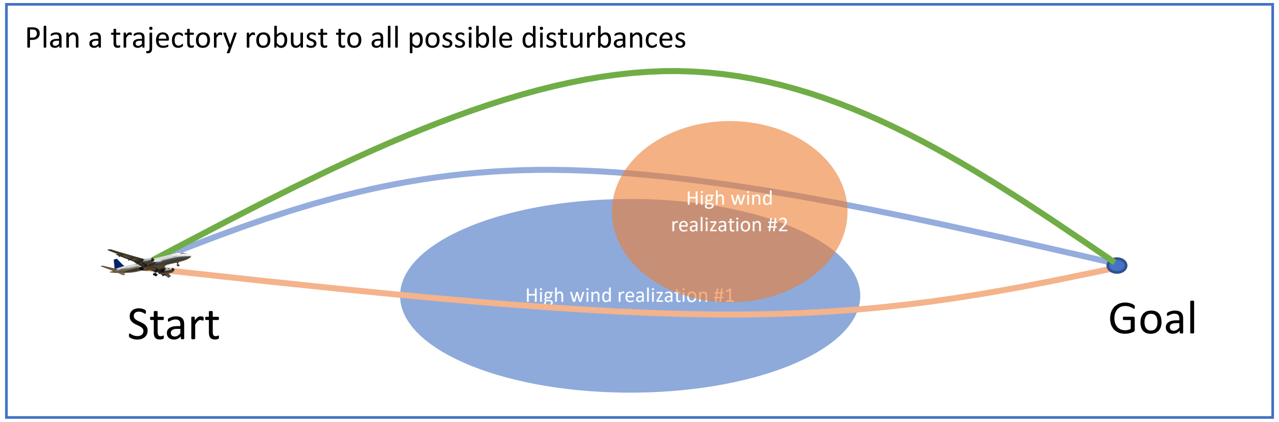
Nominal model: $\bar{x}_{k+1} = \bar{A}\bar{x}_k + \bar{B}\bar{u}_k + w_k$ where $||A - \bar{A}|| \leq \epsilon_A, ||B - \bar{B}|| \leq \epsilon_B$

Robust constraints:
Given a control policy $\pi : \mathbb{R}^n \to \mathbb{R}^d$ 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 $||A - \bar{A}|| \leq \epsilon_A, ||B - \bar{B}|| \leq \epsilon_B$

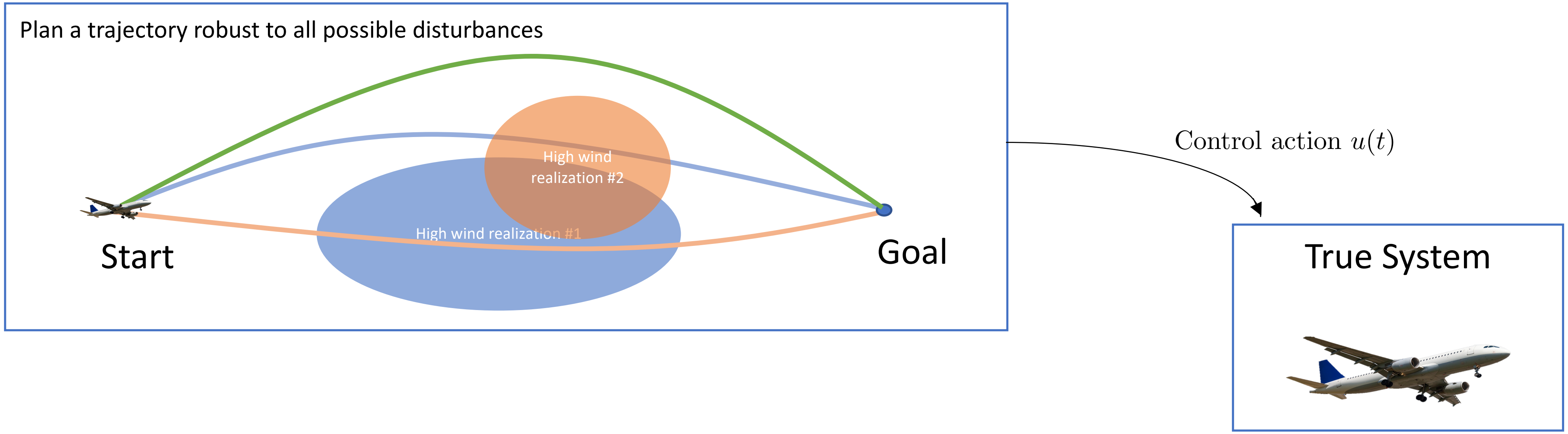
Robust constraints:

Given a control policy $\pi : \mathbb{R}^n \to \mathbb{R}^d$ 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 $||A - \bar{A}|| \leq \epsilon_A, ||B - \bar{B}|| \leq \epsilon_B$

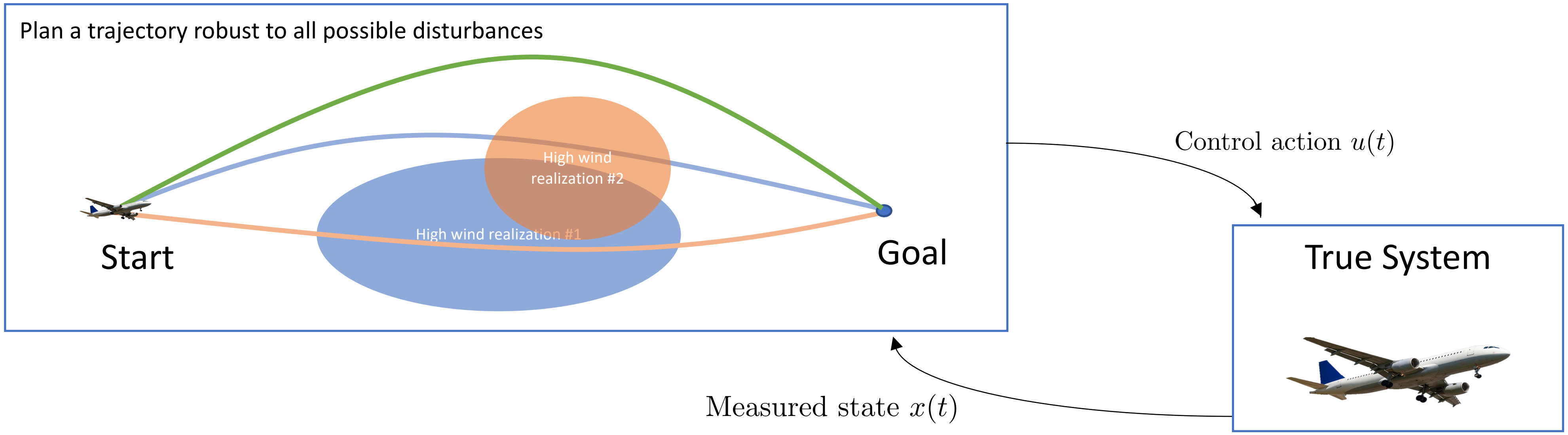
Robust constraints:

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

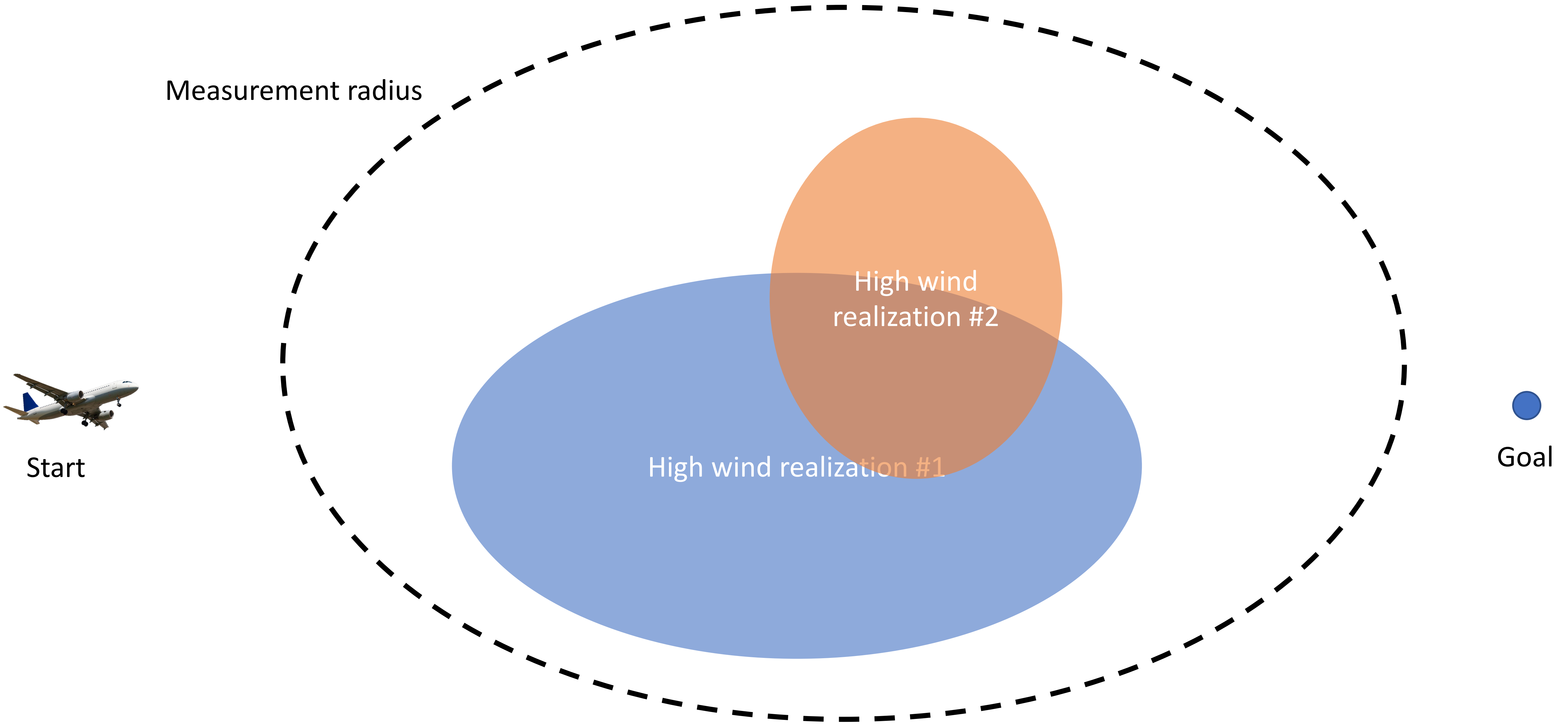
$$x_{k+1} = Ax_k + B\pi(x_k) + w_k$$

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

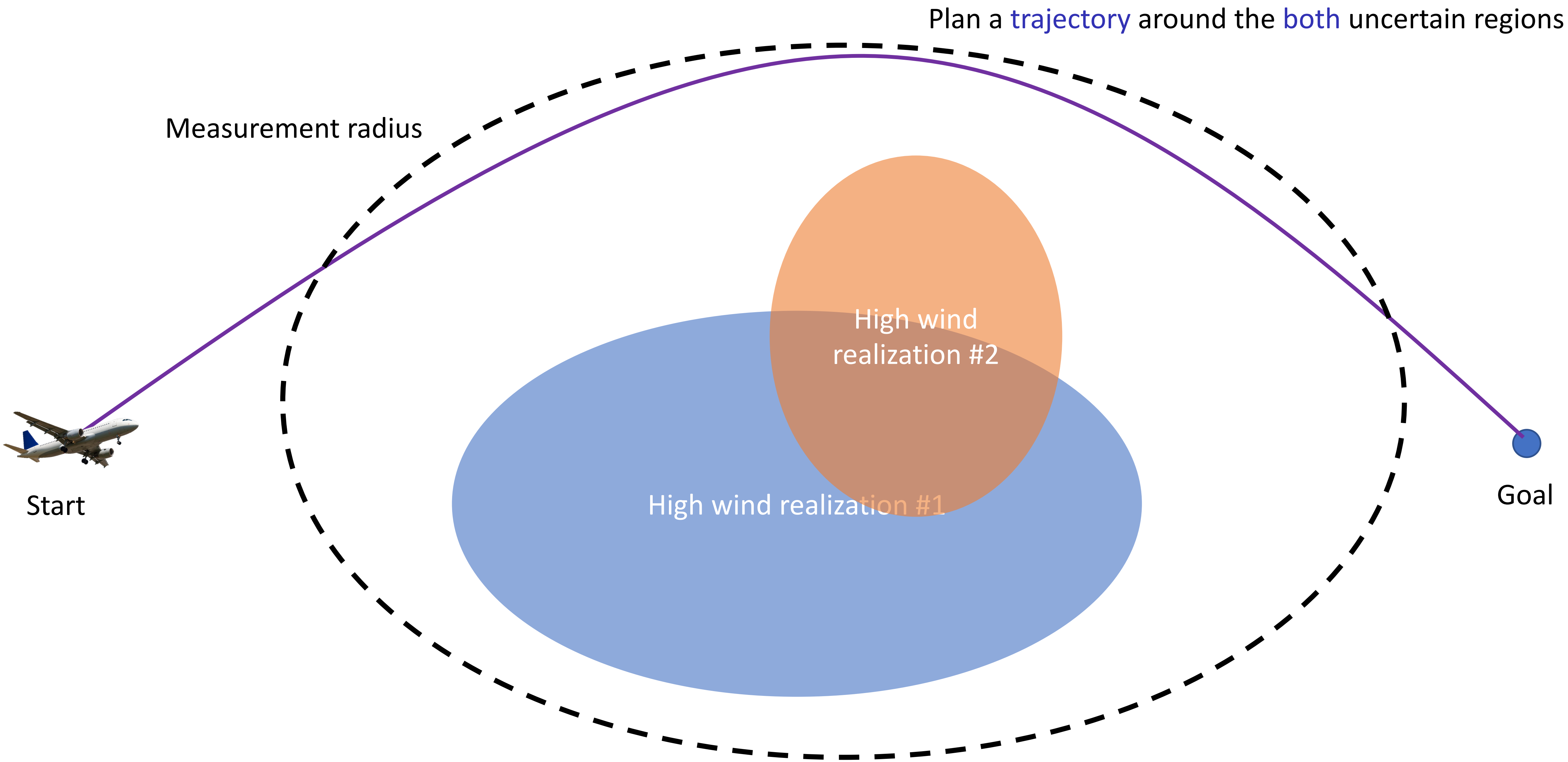
Can we leverage the MPC approach?



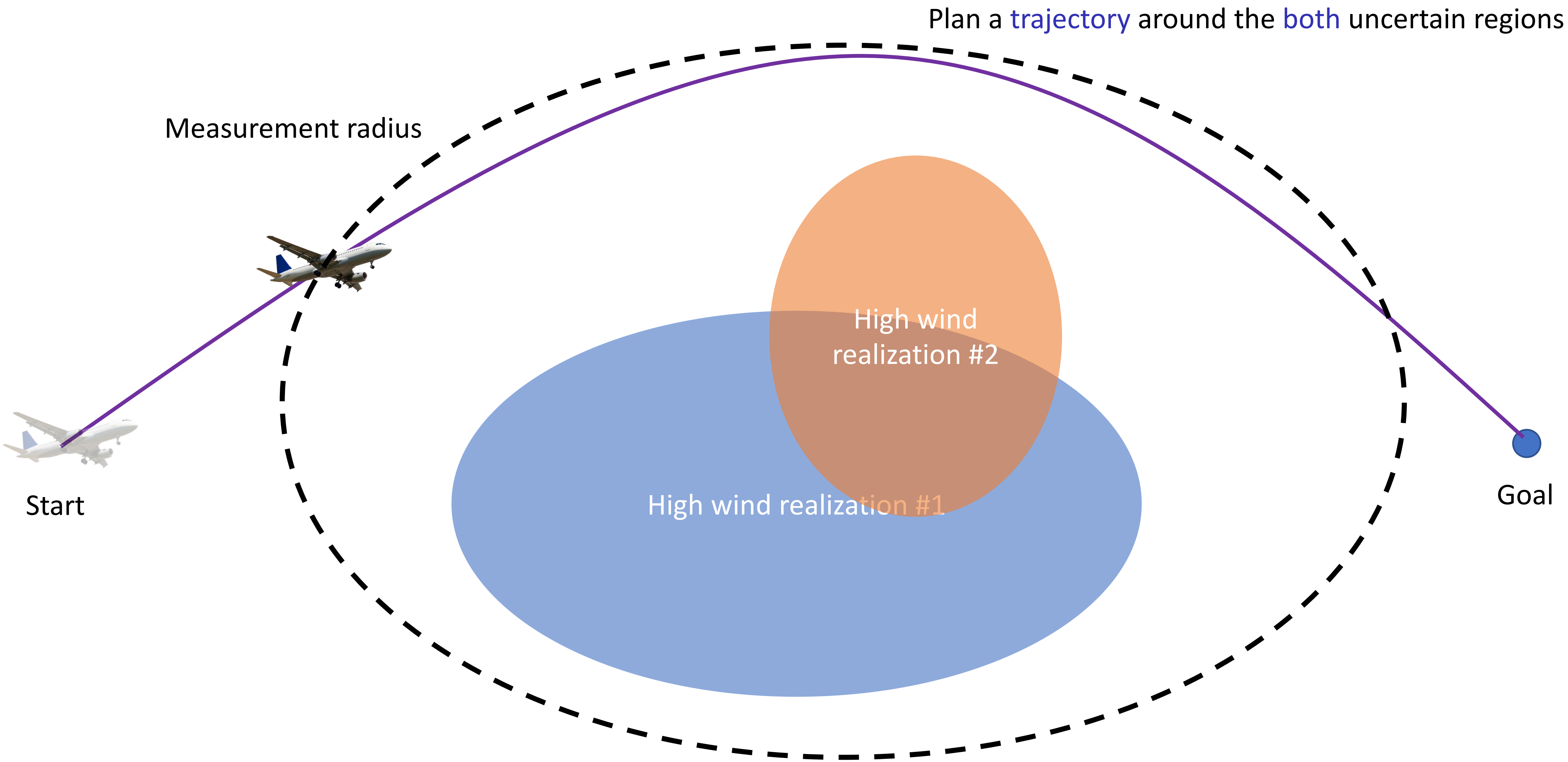
Can we leverage the MPC approach?



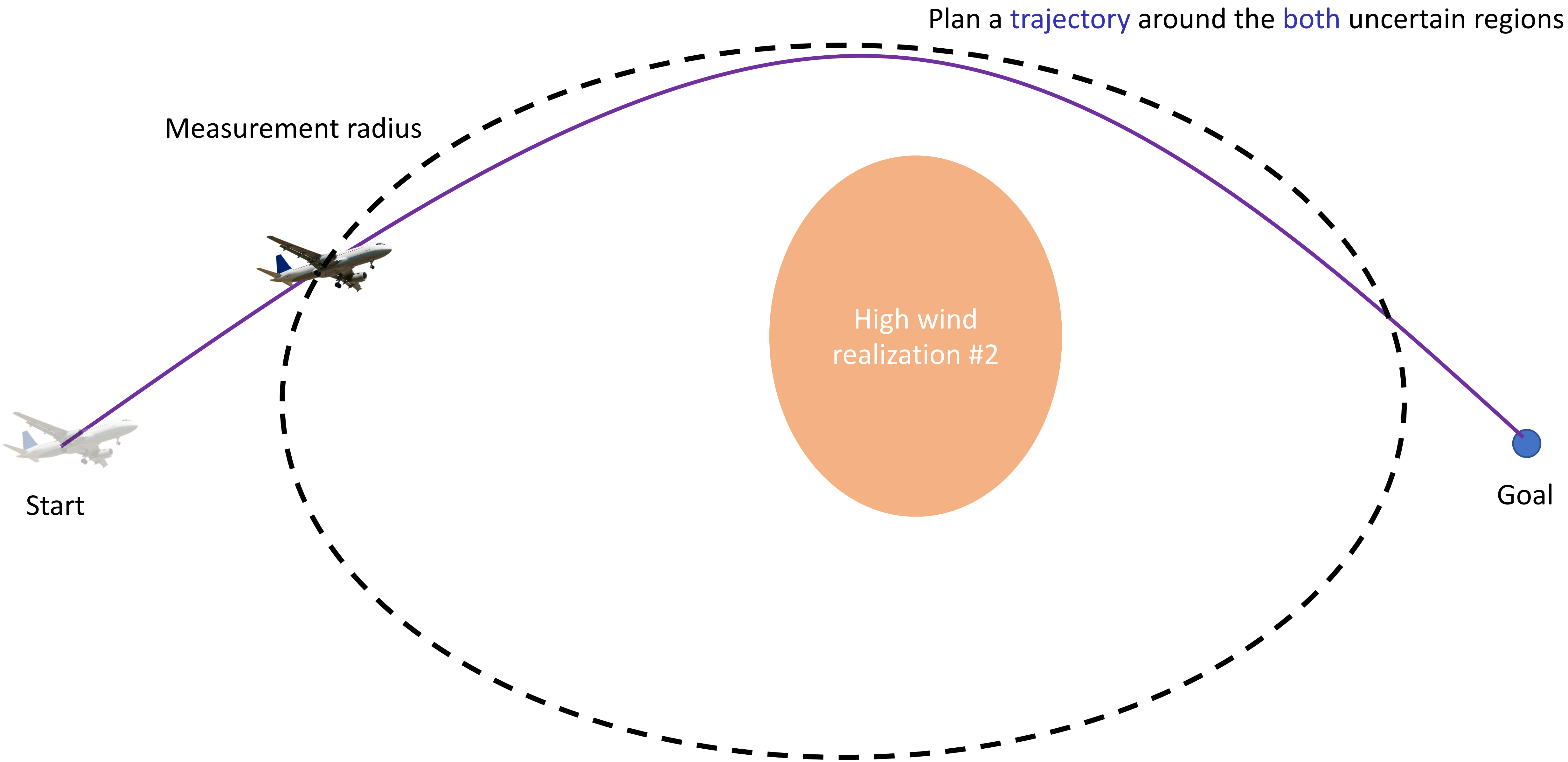
Can we leverage the MPC approach?



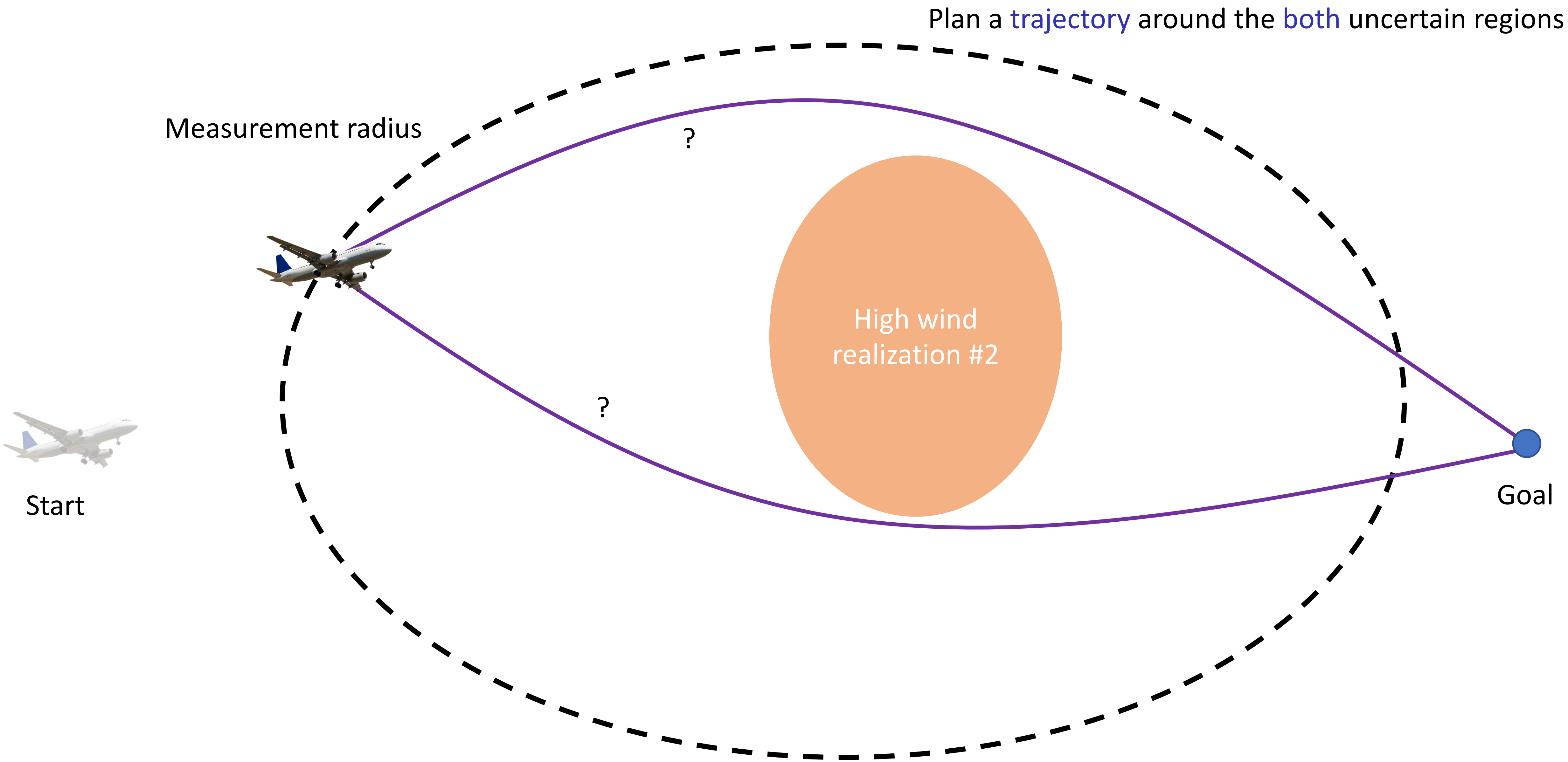
Can we leverage the MPC approach?



Can we leverage the MPC approach?

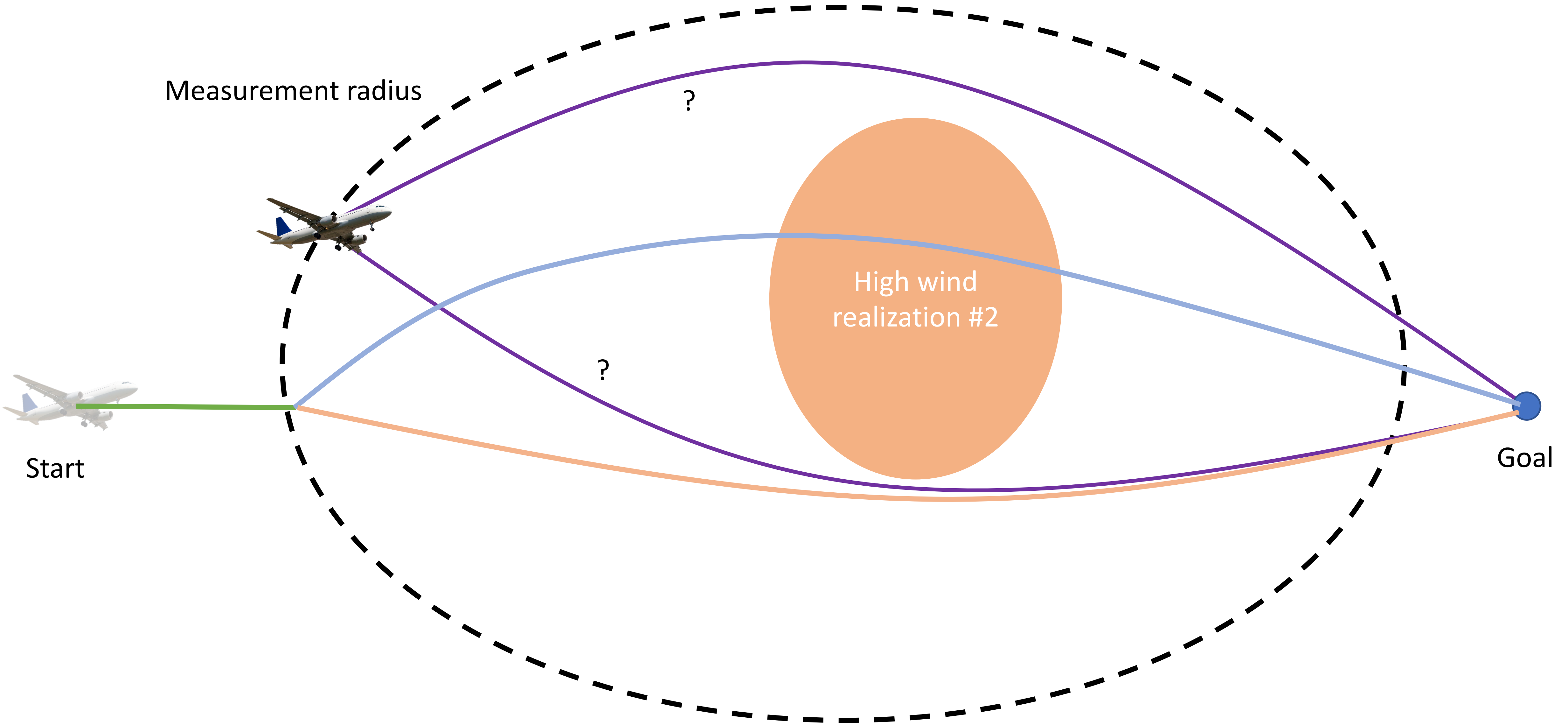


Can we leverage the MPC approach?



Can we leverage the MPC approach?

Plan a trajectory around the both uncertain regions



Why should I care?

Why should I care?

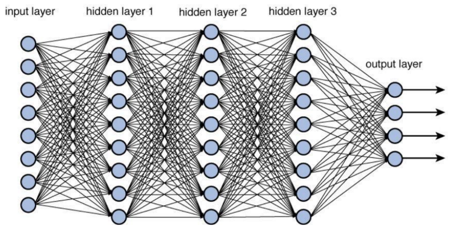
Model-based RL

Why should I care?

Model-based RL

Step 1: Learn a Model

(x_k, u_k)



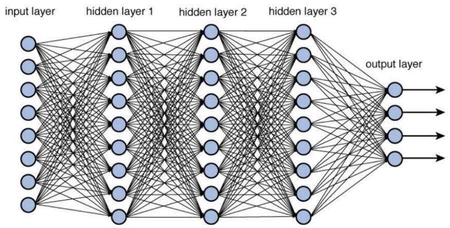
x_{k+1}

Why should I care?

Model-based RL

Step 1: Learn a Model

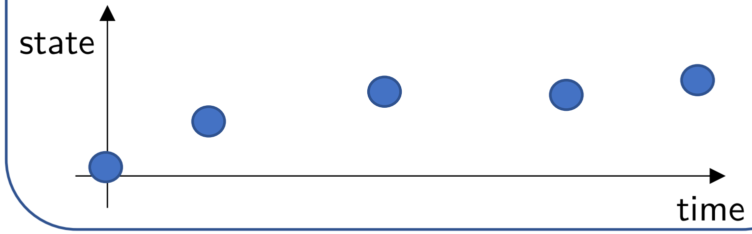
(x_k, u_k)



x_{k+1}

Step 2: Receding horizon planning

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

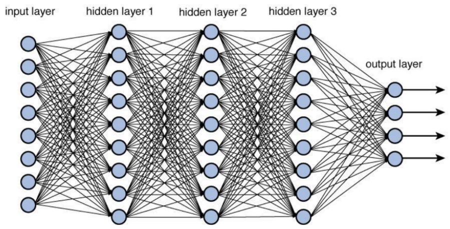


Why should I care?

Model-based RL

Step 1: Learn a Model

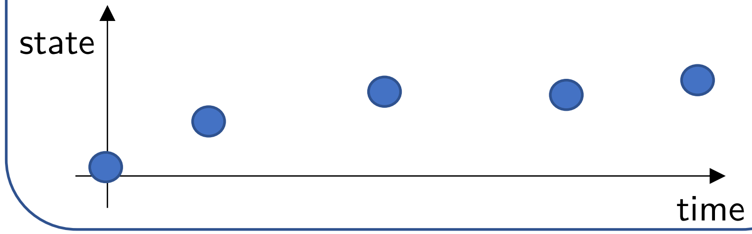
(x_k, u_k)



x_{k+1}

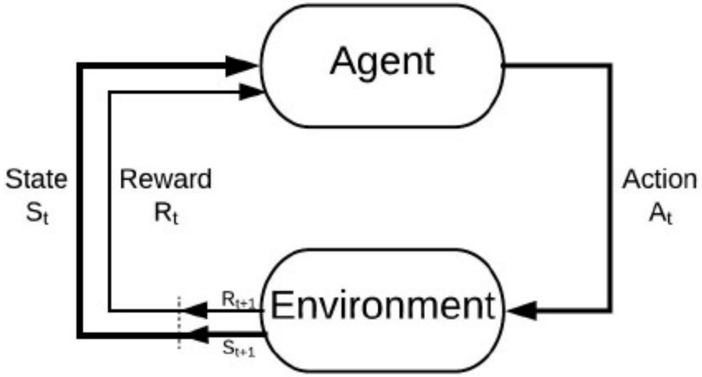
Step 2: Receding horizon planning

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

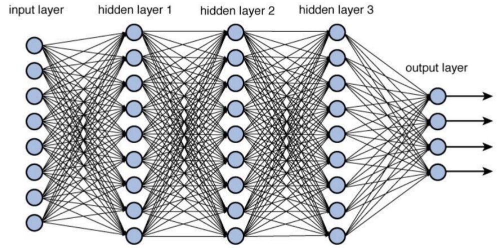


Model-free RL

Step 1: Learn a policy



x_k



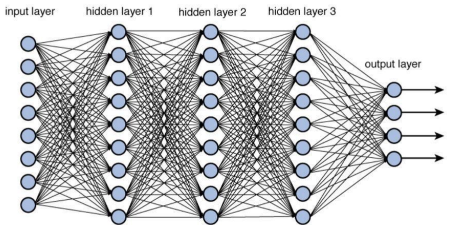
u_k

Why should I care?

Model-based RL

Step 1: Learn a Model

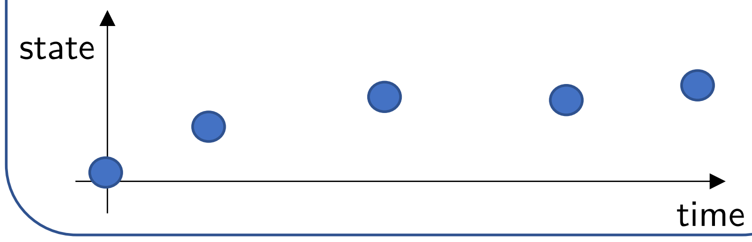
(x_k, u_k)



x_{k+1}

Step 2: Receding horizon planning

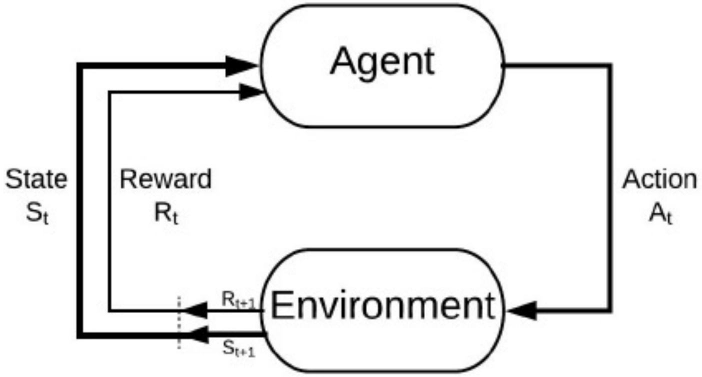
$$\min \mathbb{E}_{\substack{a_t \sim \pi \\ s_t \sim p}} [\textstyle\sum_{t=0}^{H} \gamma^t c(a_t, s_t)]$$



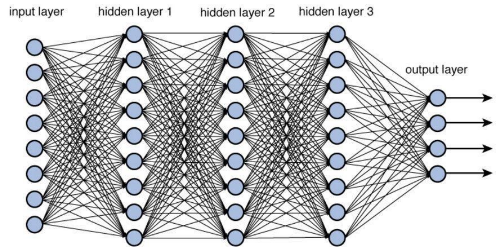
Which one has better asymptotic performance?

Model-free RL

Step 1: Learn a policy



x_k



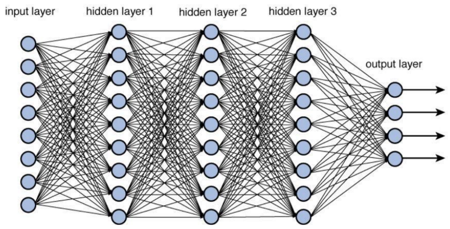
u_k

Why should I care?

Model-based RL

Step 1: Learn a Model

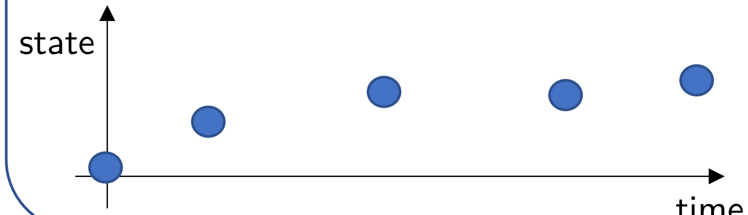
(x_k, u_k)



x_{k+1}

Step 2: Receding horizon planning

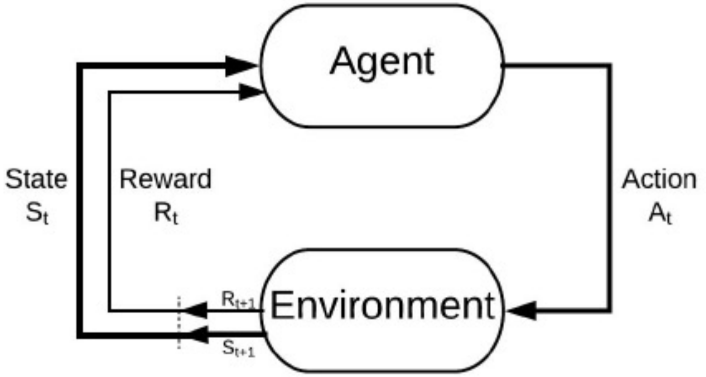
$$\min \mathbb{E}_{\substack{a_t \sim \pi \\ s_t \sim p}} [\textstyle\sum_{t=0}^{H} \gamma^t c(a_t, s_t)]$$



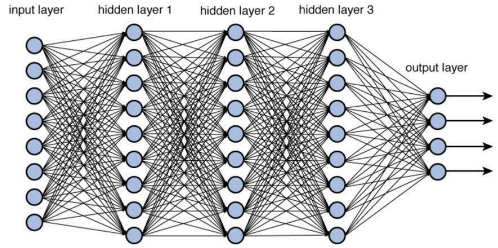
Which one has better asymptotic performance?

Model-free RL

Step 1: Learn a policy



x_k



u_k

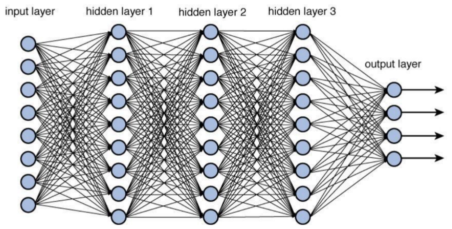
What can go wrong with model-based RL?

Why should I care?

Model-based RL

Step 1: Learn a Model

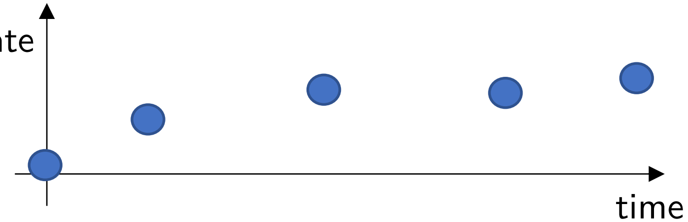
(x_k, u_k)



x_{k+1}

Step 2: Receding horizon planning

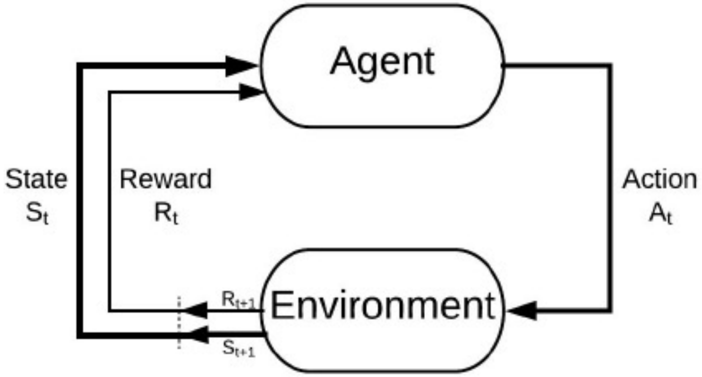
$$\min \mathbb{E}_{\substack{a_t \sim \pi \\ s_t \sim p}} [\sum_{t=0}^{H} \gamma^t c(a_t, s_t)]$$



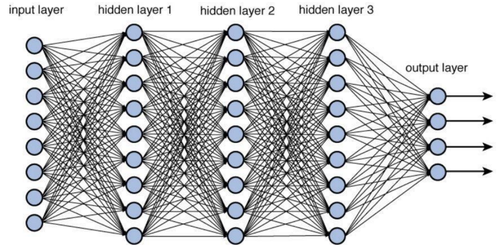
Which one has better asymptotic performance?

Model-free RL

Step 1: Learn a policy



x_k



u_k

What can go wrong with model-based RL?

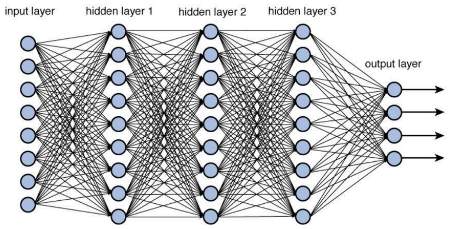
- The prediction horizon is short → need a value function and safe set

Why should I care?

Model-based RL

Step 1: Learn a Model

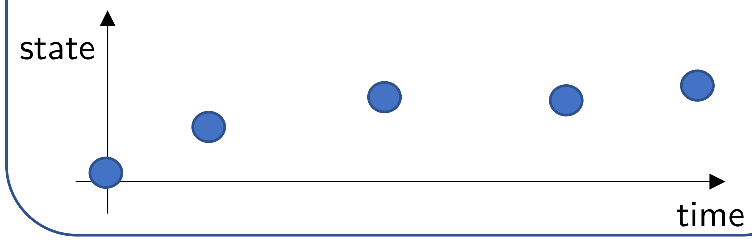
(x_k, u_k)



x_{k+1}

Step 2: Receding horizon planning

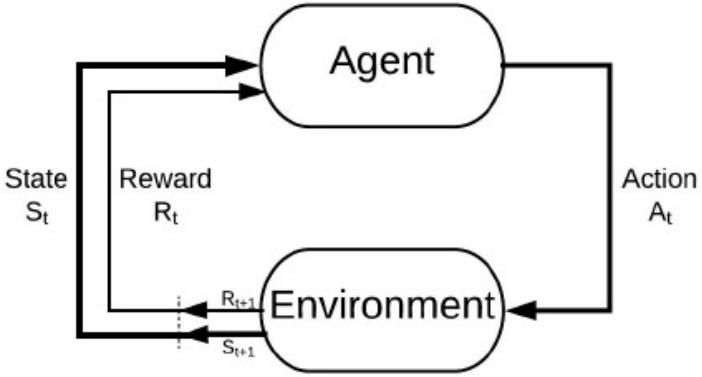
$$\min \mathbb{E}_{\substack{a_t \sim \pi \\ s_t \sim p}}\left[\sum_{t=0}^{H} \gamma^t c(a_t, s_t)\right]$$



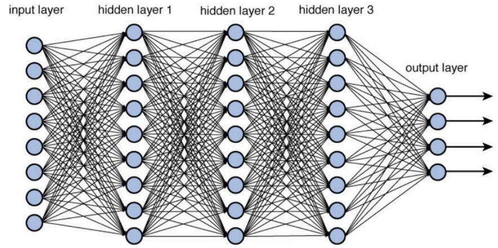
Which one has better asymptotic performance?

Model-free RL

Step 1: Learn a policy



x_k



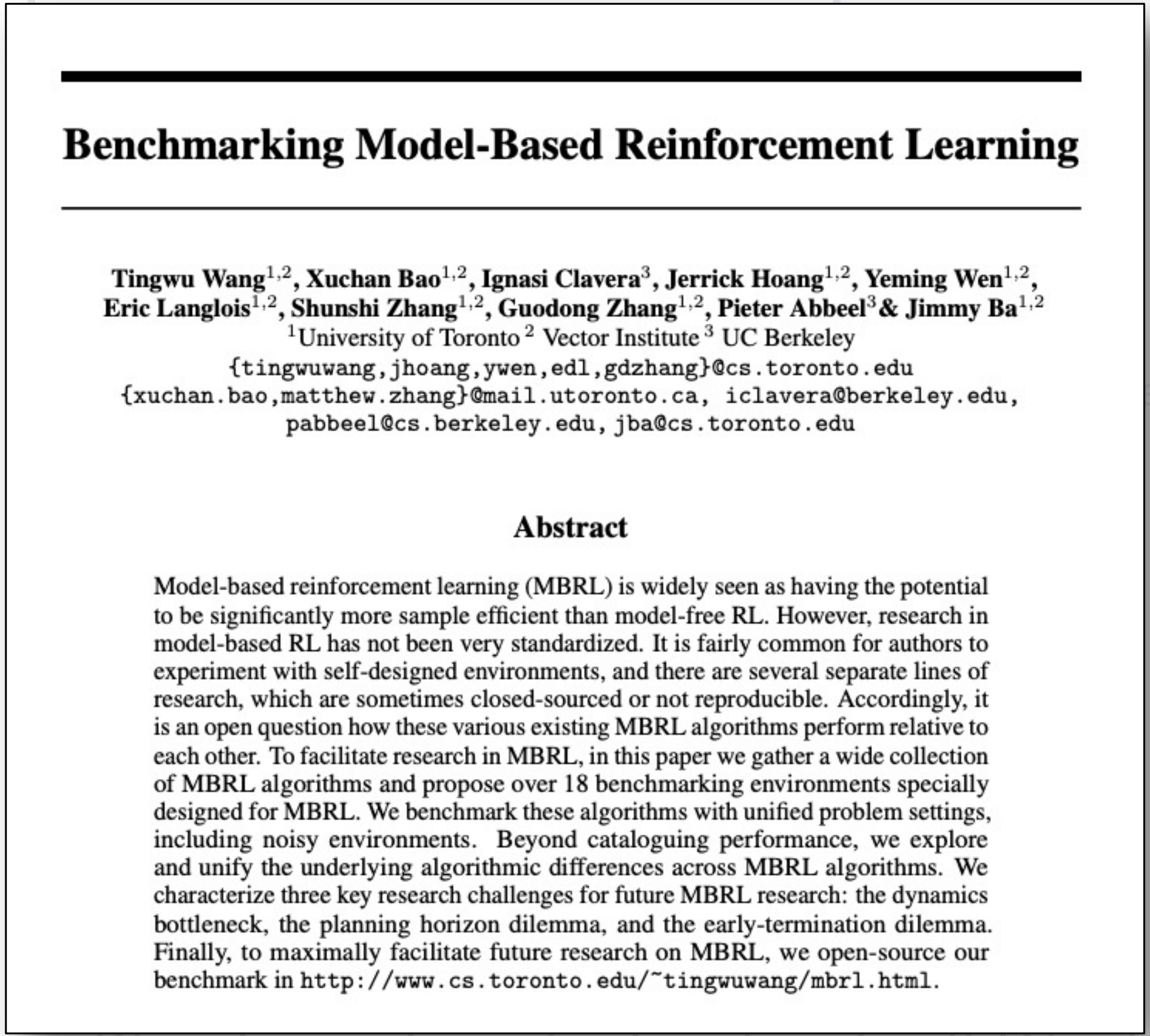
u_k

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 Clavera[3], Jerrick Hoang[1,2], Yeming Wen[1,2], Eric Langlois[1,2], Shunshi Zhang[1,2], Guodong Zhang[1,2], Pieter Abbeel[3] & Jimmy Ba[1,2]
[1]University 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
erformance?

Model-free RL

value function and safe set

▸ Planning trajectories is suboptimal! We should plan over policies

Why should I care?

Model-based RL

Model-free RL

Benchmarking Model-Based Reinforcement Learning

Tingwu Wang[1,2], Xuchan Bao[1,2], Ignasi Clavera[3], Jerrick Hoang[1,2], Yeming Wen[1,2],
Eric Langlois[1,2], Shunshi Zhang[1,2], Guodong Zhang[1,2], Pieter Abbeel[3] & Jimmy Ba[1,2]
[1]University 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-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

Model-free RL

Benchmarking Model-Based Reinforcement Learning

Tingwu Wang[1,2], Xuchan Bao[1,2], Ignasi Clavera[3], Jerrick Hoang[1,2], Yeming Wen[1,2], Eric Langlois[1,2], Shunshi Zhang[1,2], Guodong Zhang[1,2], Pieter Abbeel[3] & Jimmy Ba[1,2]
[1]University 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-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?

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

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

where

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

$$\Delta\vartheta(t) := B_{u0}^T P[B_{u0}(u_{comp}(t) - x^*(t))h + (K + A_0 C_0^+)\Delta y(t)], \quad \Delta y(t) = y(t) - y(t-h)$$

and this minimum is reachable for

$$\tilde{u}(t)dt = \tilde{u}^*(t)dt := -L^{-1}B_{u0}^T P[B_{u0}(u_{comp}(t) + x^*(t))dt + (K + A_0 C_0^+)dy(t)].$$

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

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

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

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

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

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

$$dV(e(t)) \overset{a.s.}{\le} [\varphi(t) + I(t)]dt + S^T(t)dw(t) - e^T(t)Q_c e(t)dt.$$

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*. Prinstone, 1969.
[3] T. Gard, *Introduction to Stochastic Differential Equations*. New York: Marcel Dekker, 1988.
[4] I. I. Gihman and A. V. Skorohod, *Stochastic Differential Equations*. New York: Springer-Verlag, 1972.
[5] M. Grimble, "Three and half DOF polynomial solution of the feedforward 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," *Applicationes Mathematicae*, vol. 24, no. 1, pp. 35–46, 1996.
[14] M. I. Taksar, A. S. Poznyak, and A. Iparraguirre, "Robust ... feedback control for linear stochastic systems in continuous tim... time-varying parameters," in *Proc. 34rd Conf. Decision and C...* New Orleans, LA, Dec. 1995, pp. 2179–2184.
[15] J. C. Willems, "Least squares stationary optimal control and al... Riccati equation," *IEEE Trans. Automat. Contr.*, vol. AC-16, no... 621–634, 1971.

Min–Max Feedback Model Predictive Control for Constrained Linear Systems

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

Abstract—**Min–max feedback formulations of model predictive ... are discussed, both in the fixed and variable horizon contexts. The ... schemes the authors discuss introduce, in the control optimizati... notion that feedback is present in the receding-horizon implemen... of the control. This leads to improved performance, compared to sta... model predictive control, and resolves the feasibility difficulties tha... with the min–max techniques that are documented in the lite... The stabilizing properties of the methods are discussed as well as... practical implementation details.**

Index Terms—**Feedback, min–max optimization, model predictiv... trol.**

I. INTRODUCTION

Model predictive control (MPC) is a control methodolog... is becoming mature. The basic aspects of the method are... well understood, and stabilizing formulations of the control la... documented in the literature, both for linear and nonlinear pro... [1], [2].

The MPC strategy optimizes an open-loop control sequen... each sample, to minimize a nominal cost function, subject to... state and input constraints. The optimization is usually bas... the assumption that the process model is exact and that ... disturbances are constant. Because the control law ignores the ... of possible future changes in disturbance and model mis... closed-loop performance can be poor [3], with likely violatio... the constraints, when disturbances or model mismatch are pres...

Some formulations of MPC have been proposed that address... issues [4], [5]. These methods rely on a min–max optimizat... predicted performance. However, these formulations optimize a... control profile over all possible disturbance (or model mis...

ELSEVIER

Available online at www.sciencedirect.com

SCIENCE DIRECT

automatica

Automatica 42 (2006) 523–533

www.elsevier.com/loc...

Optimization over state feedback policies for robust control with constr...

Paul J. Goulart[a,*], Eric C. Kerrigan[b], Jan M. Maciejowski[a]

[a] Department of Engineering, University of Cambridge, Trumpington Street, Cambridge CB2 1PZ, UK

[b] Department of Aeronautics and Department of Electrical and Electronic Engineering, Imperial College London, Exhibition Road, London S...

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 distu... mixed polytopic constraints on the state and input. It is shown that the class of admissible affine state feedback control policies wit... of 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 st... 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 th... 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 calcu... 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.

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 (M... 2002; Mayne, Rawlings, Rao, & Scokaert, 2000).

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

ELSEVIER

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]

[a] Department of Electrical and Electronic Engineering, Imperial College, London, SW72BT, UK

[b] School 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.

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 ...

$\{u_0, u_1, \ldots, u_{N-1}\}$ 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 $\{\mu_0(\cdot), \mu_1(\cdot), \ldots, \mu_{N-1}(\cdot)\}$ 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 con...

How do we fix this problem?

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

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

where

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

$$\Delta\vartheta(t) := B_{u0}^T P[B_{u0}(u_{\text{comp}}(t) - x^*(t))h + (K + A_0C_0^+)\Delta y(t)], \quad \Delta y(t) = y(t) - y(t-h)$$

and this minimum is reachable for

$$\bar{u}(t)\,dt = \bar{u}^*(t)\,dt := -L^{-1}B_{u0}^T P[B_{u0}(u_{\text{comp}}(t) + x^*(t))\,dt + (K + A_0C_0^+)\,dy(t)].$$

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

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

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

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

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

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

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*. Princetone, 1969.
[3] T. Gard, *Introduction to Stochastic Differential Equations*. New York: Marcel Dekker, 1988.
[4] I. I. Gihman and A. V. Skorohod, *Stochastic Differential Equations*. New York: Springer-Verlag, 1972.
[5] M. Grimble, "Three and half DOF polynomial solution of the feedforward 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," *Applicationes Mathematicae*, vol. 24, no. 1, pp. 35–46, 1996.
[14] M. I. Taksar, A. S. Poznyak, and A. Iparraguirre, "Robust ... feedback control for linear stochastic systems in continuous tim... time-varying parameters," in *Proc. 34rd Conf. Decision and C...* New Orleans, LA, Dec. 1995, pp. 2179–2184.
[15] J. C. Willems, "Least squares stationary optimal control and al... Riccati equation," *IEEE Trans. Automat. Contr.*, vol. AC-16, no... 621–634, 1971.

Min–Max Feedback Model Predictive Control for Constrained Linear Systems

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

Abstract—**Min–max feedback formulations of model predictive ... are discussed, both in the fixed and variable horizon contexts. The ... schemes the authors discuss introduce, in the control optimizati... notion that feedback is present in the receding-horizon implemen... of the control. This leads to improved performance, compared to sta... model predictive control, and resolves the feasibility difficulties tha... with the min–max techniques that are documented in the liter... The stabilizing properties of the methods are discussed as well a... practical implementation details.**

Index Terms—**Feedback, min–max optimization, model predictiv... trol.**

I. Introduction

Model predictive control (MPC) is a control methodolog... is becoming mature. The basic aspects of the method are well understood, and stabilizing formulations of the control la... documented in the literature, both for linear and nonlinear pro... [1], [2].

The MPC strategy optimizes an open-loop control sequen... each sample, to minimize a nominal cost function, subject to state and input constraints. The optimization is usually bas... the assumption that the process model is exact and that disturbances are constant. Because the control law ignores the ... of possible future changes in disturbance and model mis... closed-loop performance can be poor [3], with likely violatio... the constraints, when disturbances or model mismatch are pres...

Some formulations of MPC have been proposed that address... issues [4], [5]. These methods rely on a min–max optimizat... predicted performance. However, these formulations optimize a... control profile over all possible disturbance (or model mis...

ELSEVIER
Available online at www.sciencedirect.com
SCIENCE DIRECT
automatica
Automatica 42 (2006) 523–533
www.elsevier.com/loc...

Optimization over state feedback policies for robust control with constr...

Paul J. Goulart[a,*], Eric C. Kerrigan[b], Jan M. Maciejowski[a]
[a] Department of Engineering, University of Cambridge, Trumpington Street, Cambridge CB2 1PZ, UK
[b] Department of Aeronautics and Department of Electrical and Electronic Engineering, Imperial College London, Exhibition Road, London S...

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 distu... mixed polytopic constraints on the state and input. It is shown that the class of admissible affine state feedback control policies wi... of 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 st... 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 th... 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 calcu... 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.

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 (M... 2002; Mayne, Rawlings, Rao, & Scokaert, 2000).

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

ELSEVIER
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]
[a] Department of Electrical and Electronic Engineering, Imperial College, London, SW72BT, UK
[b] School 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.

Keywords: Robust model predictive control; Robustness; Bounded disturbances

1. Introduction

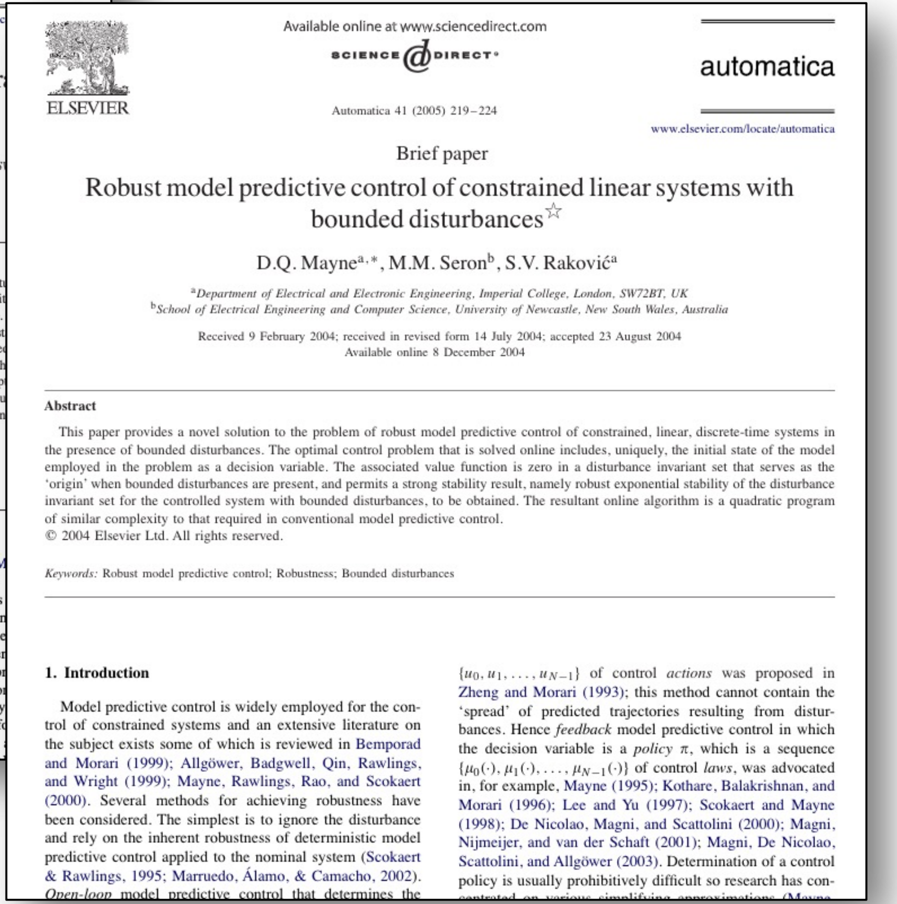
Model predictive control is widely employed for the control of constrained systems and an extensive literature on the subject exists some of which is reviewed in Bemporad and Morari (1999); Allgöwer, Badgwell, Qin, Rawlings, and Wright (1999); Mayne, Rawlings, Rao, and Scokaert (2000). Several methods 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 ...

$\{u_0, u_1, \ldots, u_{N-1}\}$ 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 $\{\mu_0(\cdot), \mu_1(\cdot), \ldots, \mu_{N-1}(\cdot)\}$ 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 con...

The problem is NP-hard!

How do we fix this problem?

The problem is NP-hard!

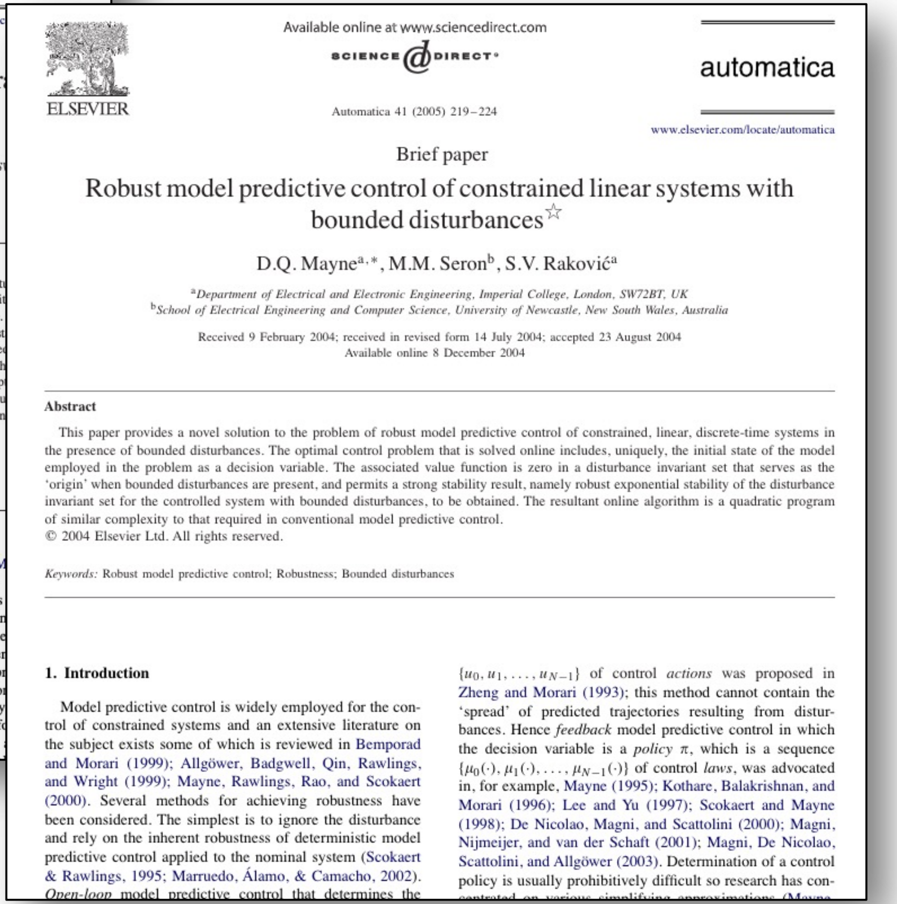


An elegant approximation

How do we fix this problem?

The problem is NP-hard!

An elegant approximation



What people do in practice

How do we fix this problem?

The problem is NP-hard!

An elegant approximation

What people do in practice

Optimizing over policies

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

Optimizing over policies

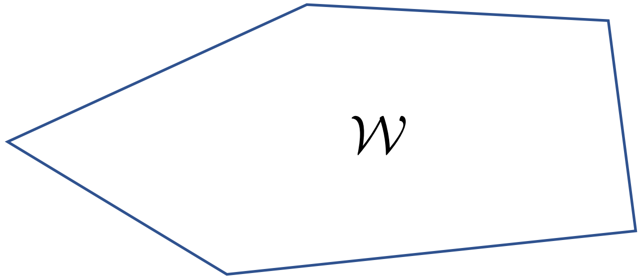
True system dynamics: $x_{k+1} = Ax_k + Bu_k + w_k$

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

Optimizing over policies

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

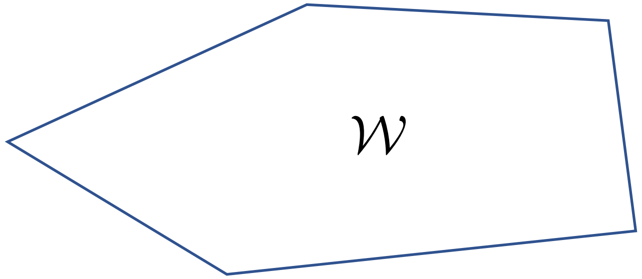
Assumption: the disturbance support $\mathcal{W} = \mathrm{conv}(w^{(0)}, \ldots, w^{(n_d)})$.



Optimizing over policies

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

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



The problem that we want to solve

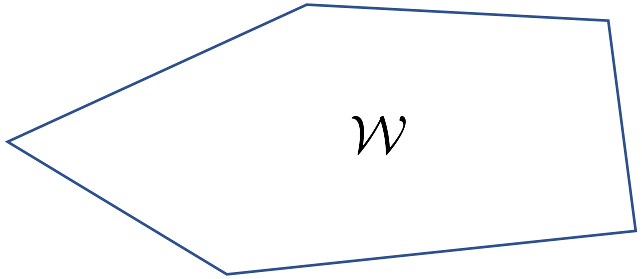
$$
\begin{aligned}
\min_{\pi} \quad & f(x(t), \pi, w_{t|t}, \ldots, w_{t+N|t}) \\
\text{subject to} \quad & x_{k+1|t} = Ax_{k|t} + Bu_{k|t} + w_{k|t} \\
& \text{robust constraint satisfaction} \\
& x_{t|t} = x(t) \\
& u_{k|t} = \pi(x_{k|t}),
\end{aligned}
$$

Optimization over policies

Optimizing over policies

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

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



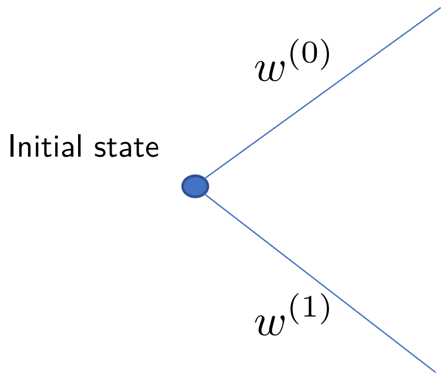
The problem that we want to solve

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

Optimization over policies

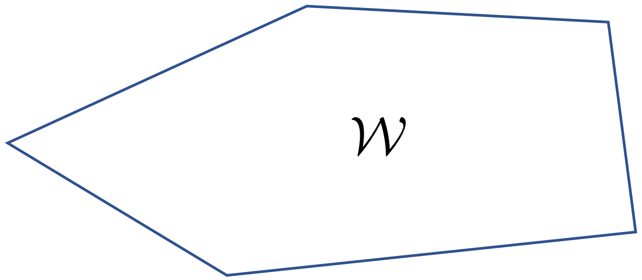
Finite dimensional reformulation

Optimize over a tree



Optimizing over policies

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



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

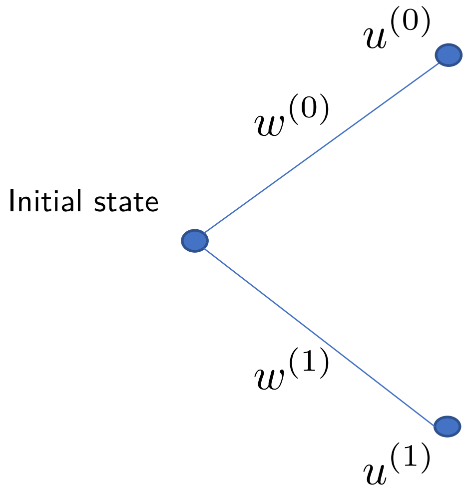
The problem that we want to solve

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

Optimization over policies

Finite dimensional reformulation

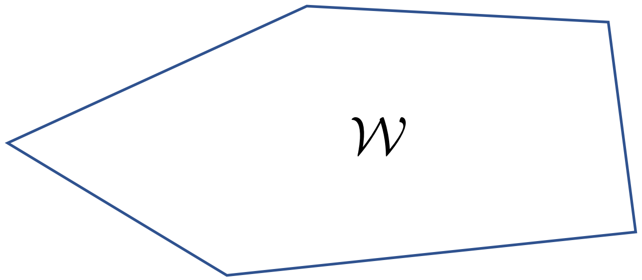
Optimize over a tree



Optimizing over policies

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

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



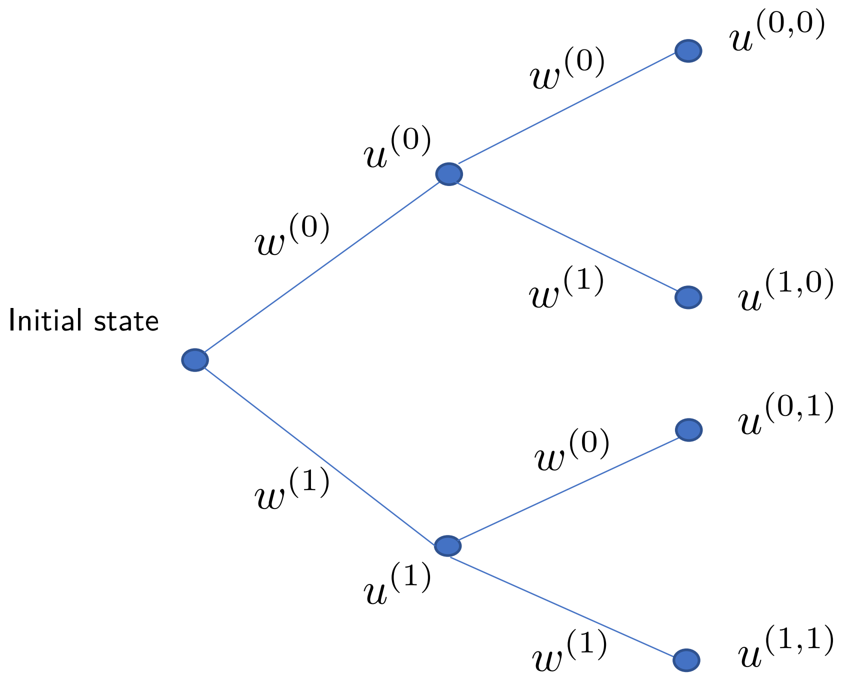
The problem that we want to solve

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

Optimization over policies

Finite dimensional reformulation

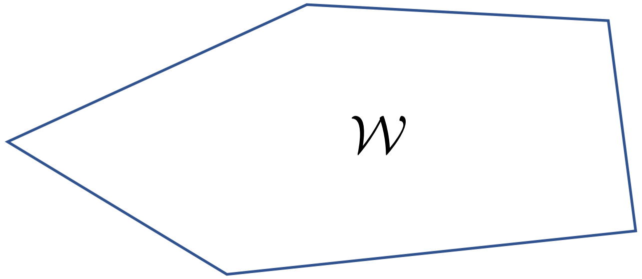
Optimize over a tree



Optimizing over policies

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

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



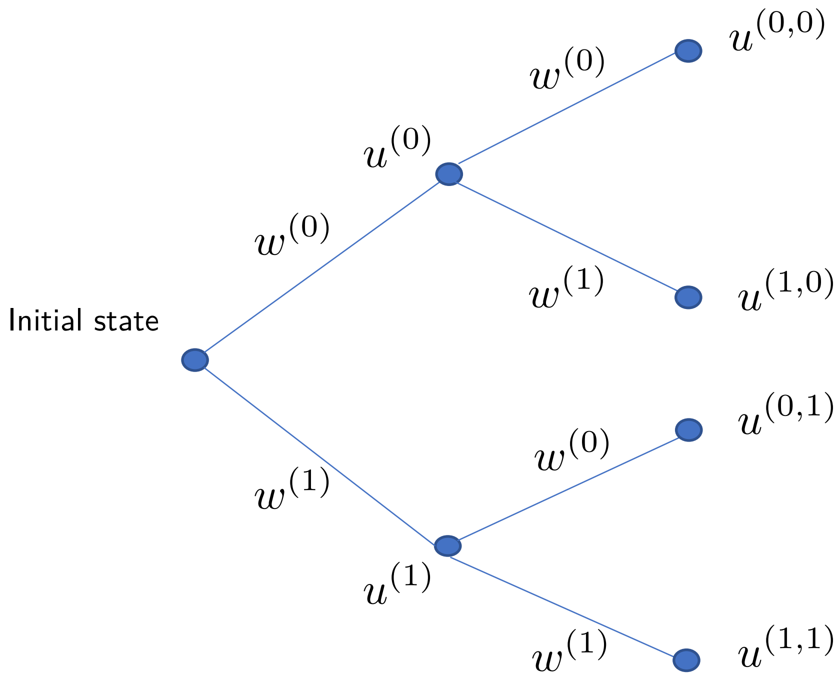
The problem that we want to solve

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

Optimization over policies

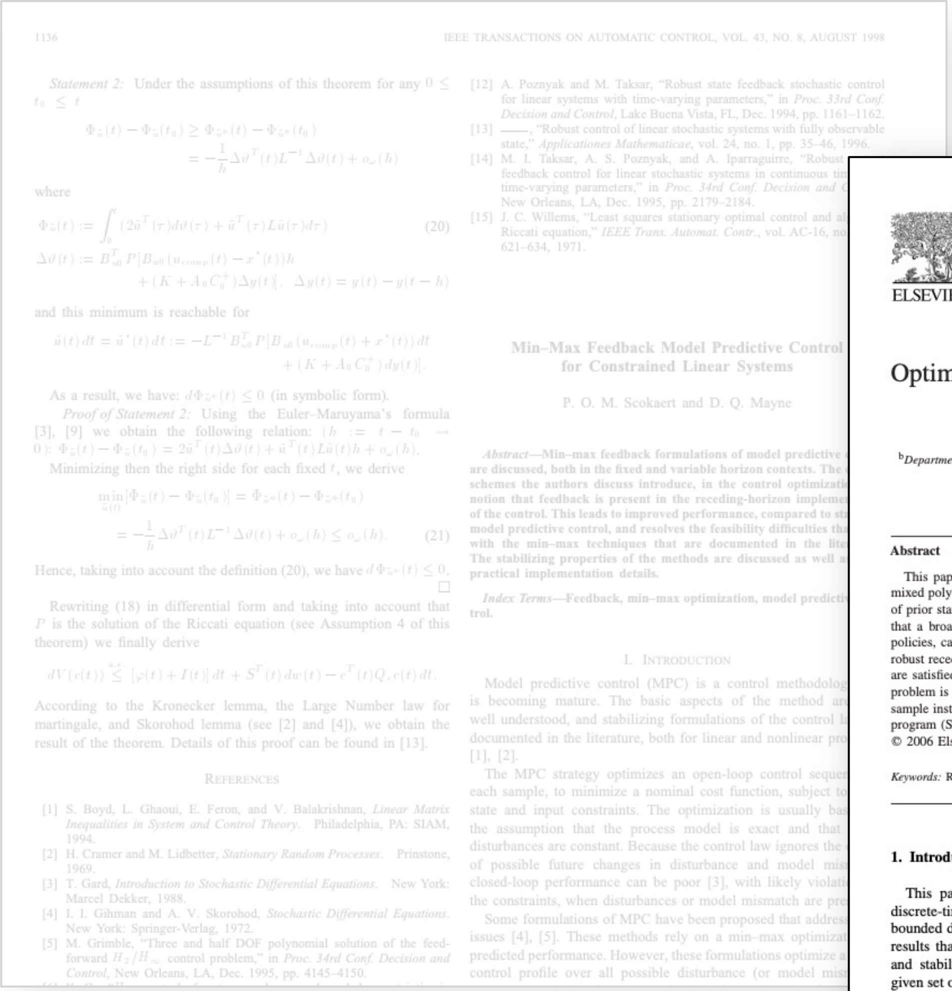
Finite dimensional reformulation

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?



The problem is NP-hard!

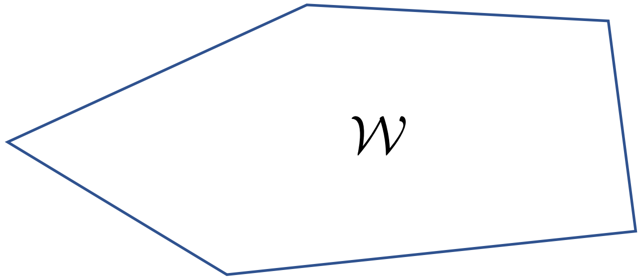
An elegant approximation

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 $\mathcal{W} = \mathrm{conv}(w^{(0)}, \dots, w^{(n_d)})$.



The problem that we want to solve

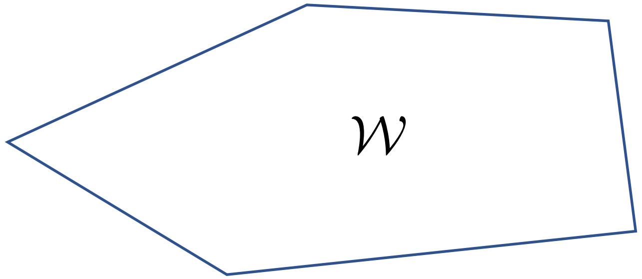
$$
\begin{aligned}
\min_{\pi} \quad & f(x(t), \pi, w_{t|t}, \dots, w_{t+N|t}) \\
\text{subject to} \quad & x_{k+1|t} = Ax_{k|t} + Bu_{k|t} + w_{k|t} \\
& \text{robust constraint satisfaction} \\
& x_{t|t} = x(t) \\
& u_{k|t} = \pi(x_{k|t}),
\end{aligned}
$$

Optimization over policies

Optimizing over disturbance feedback policies

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

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



The problem that we want to solve

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

Optimization over policies

Tractable approximation

Optimize over disturbance feedback policies

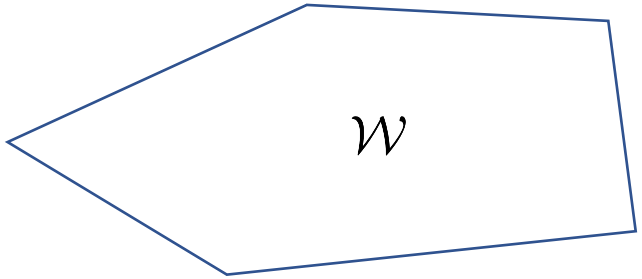
$$u_{k|t} = \pi(x(t), w_{t|t}, \ldots, w_{k-1|t})$$

Function of disturbances!

Optimizing over disturbance feedback policies

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

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



The problem that we want to solve

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

Optimization over policies

Tractable approximation

Optimize over disturbance feedback policies

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

Function of disturbances!

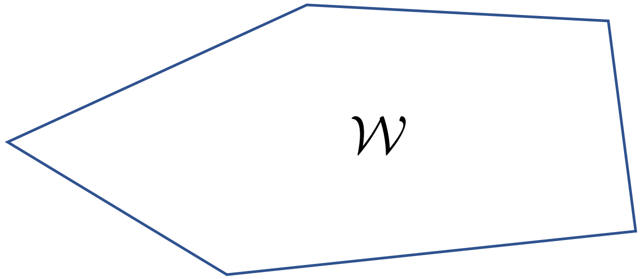
The problem is tractable when we pick:

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

Optimizing over disturbance feedback policies

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

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



The problem that we want to solve

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

Optimization over policies

Tractable approximation

Optimize over disturbance feedback policies

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

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

Min–Max Feedback Model Predictive Control for Constrained Linear Systems

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

The problem is NP-hard!

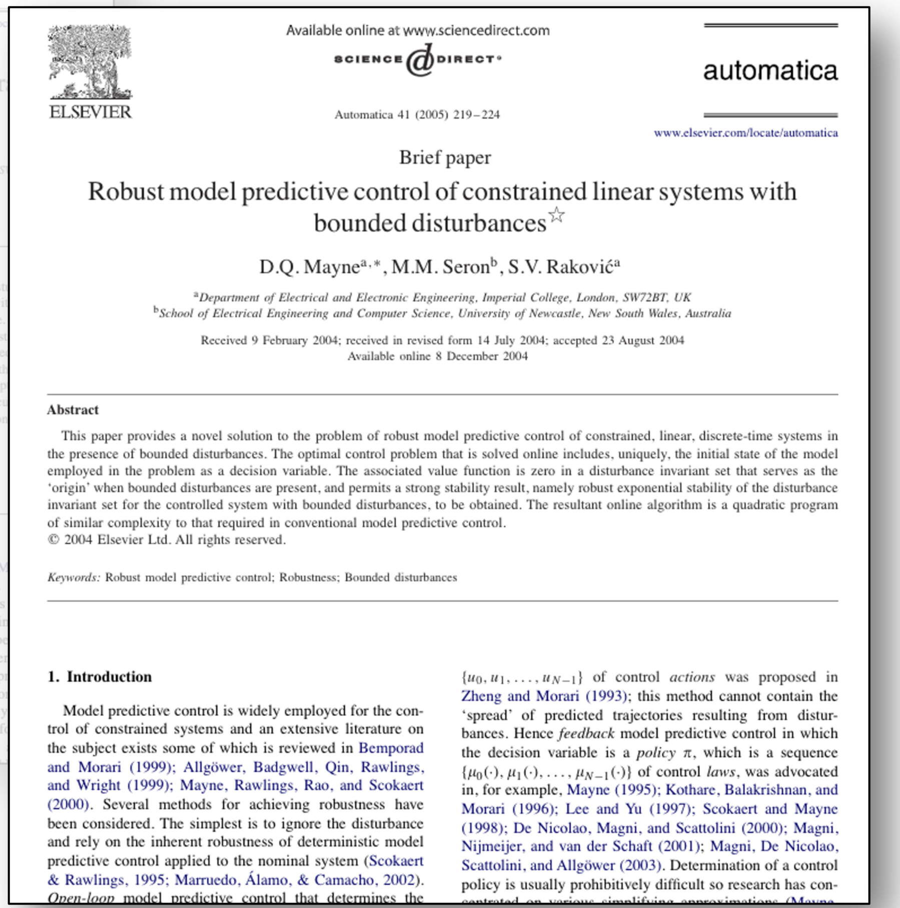

Available online at www.sciencedirect.com

Automatica 42 (2006) 523–533

Optimization over state feedback policies for robust control with constr…

Paul J. Goulart, Eric C. Kerrigan, Jan M. Maciejowski

An elegant approximation


Available online at www.sciencedirect.com

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]

[a] Department of Electrical and Electronic Engineering, Imperial College, London, SW72BT, UK

[b] School 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.

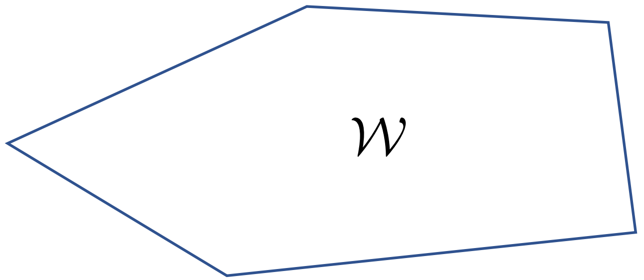
Keywords: Robust model predictive control; Robustness; Bounded disturbances

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 $\{u_0, u_1, \ldots, u_{N-1}\}$ 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 $\{\mu_0(\cdot), \mu_1(\cdot), \ldots, \mu_{N-1}(\cdot)\}$ 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 con…

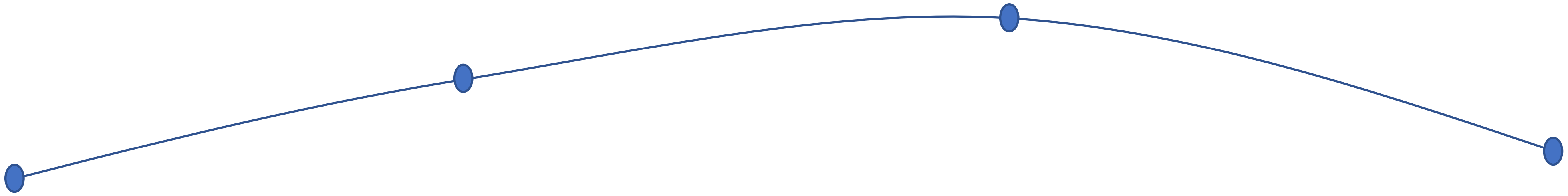
What people do in practice

Fixed tube robust MPC



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

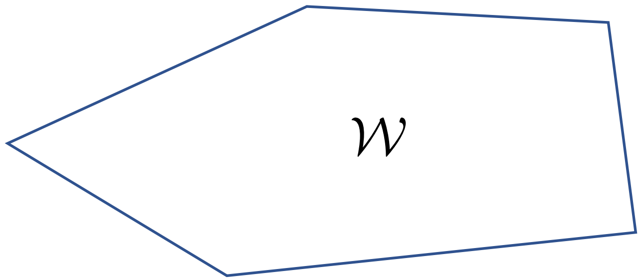
Assumption: the disturbance support $\mathcal{W} = \mathrm{conv}(w^{(0)}, \dots, w^{(n_d)})$.

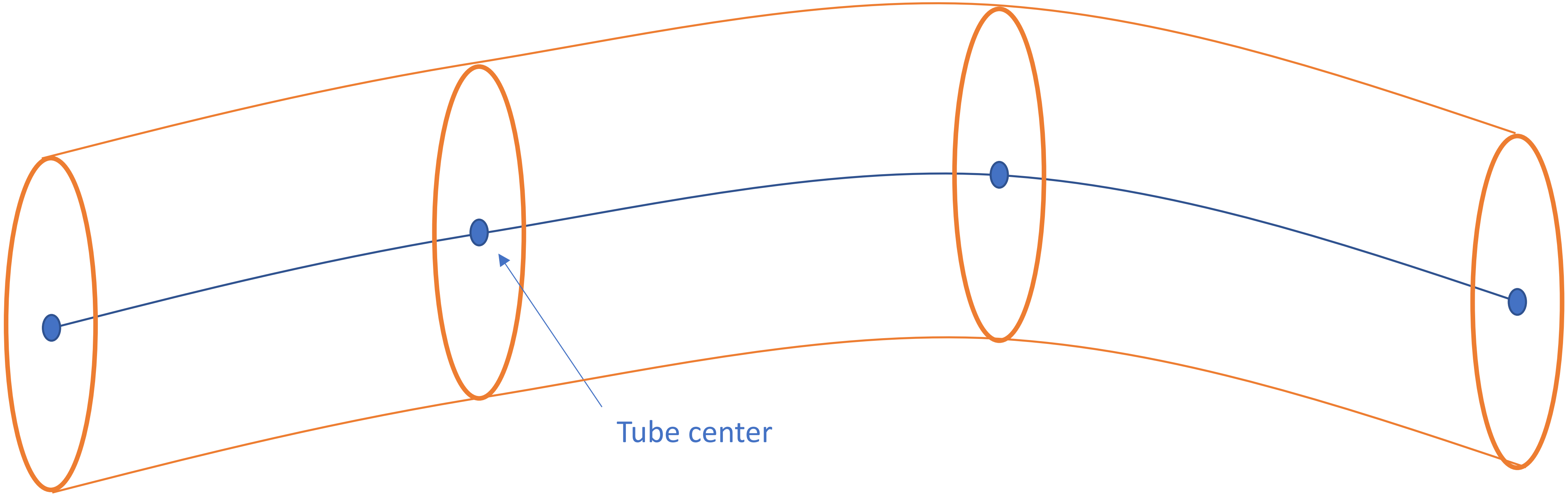


Fixed tube robust MPC

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

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

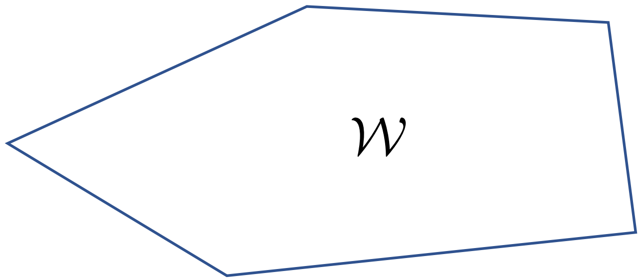


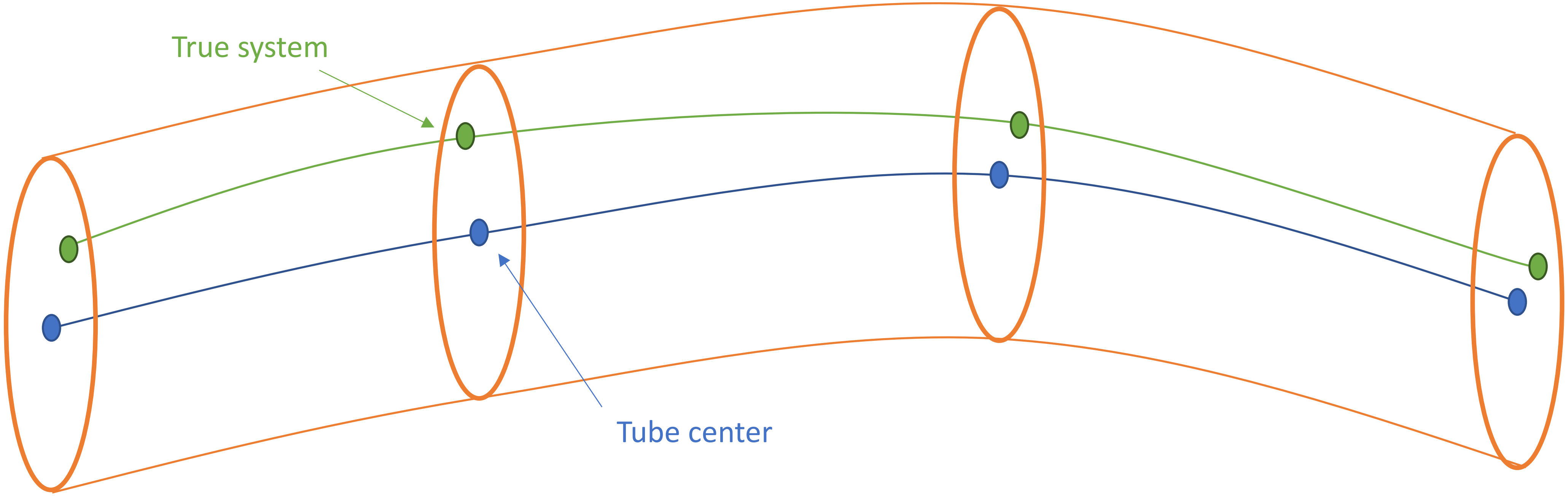


Fixed tube robust MPC

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

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

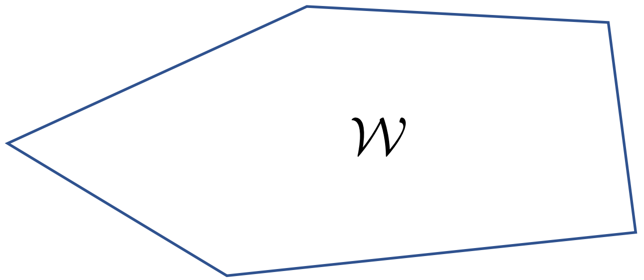


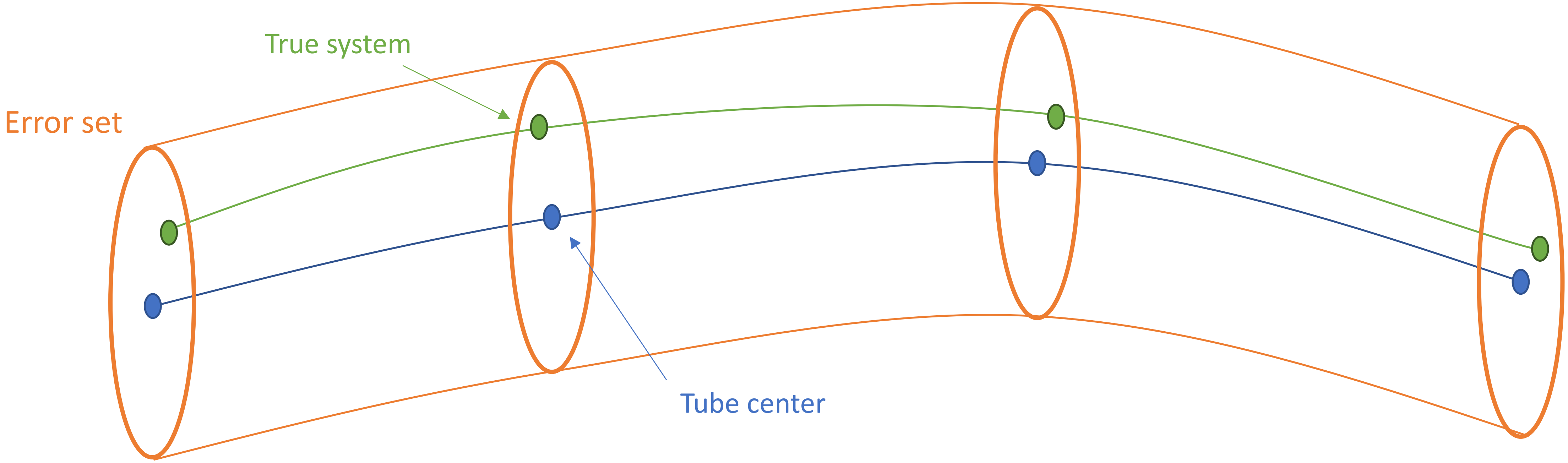


Fixed tube robust MPC

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

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

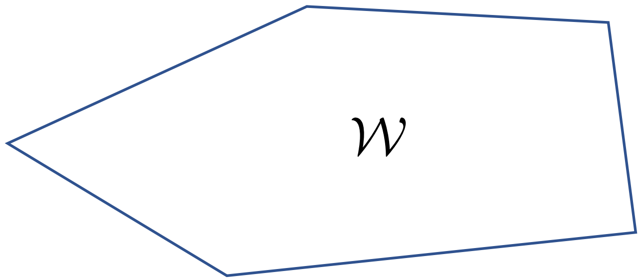


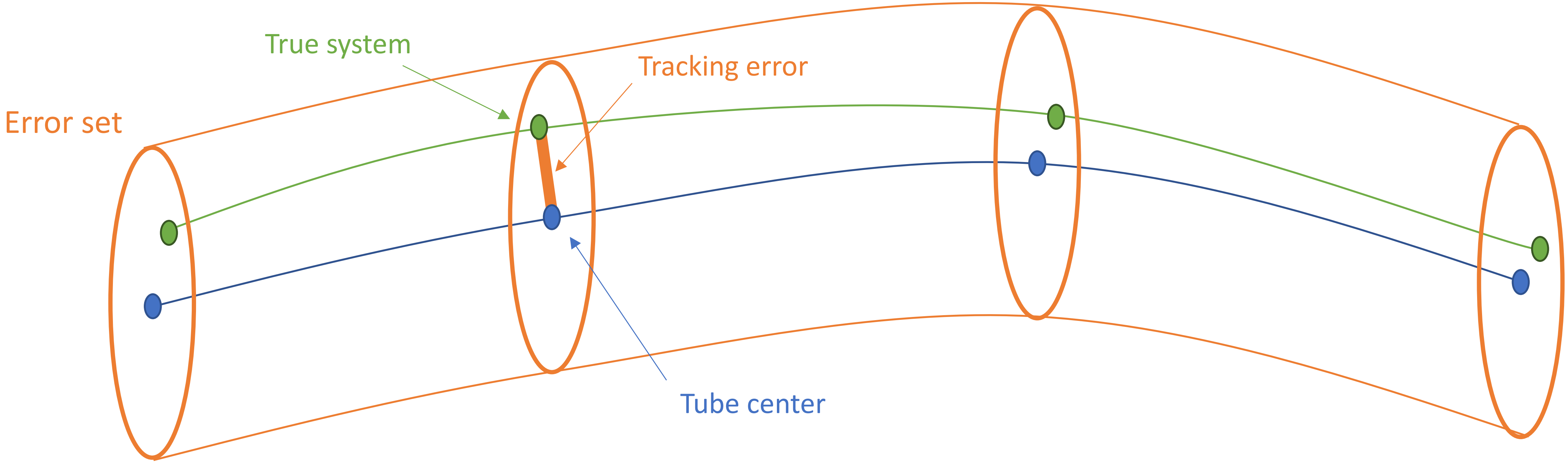


Fixed tube robust MPC

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

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

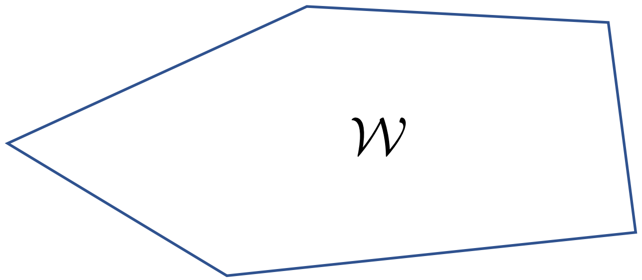


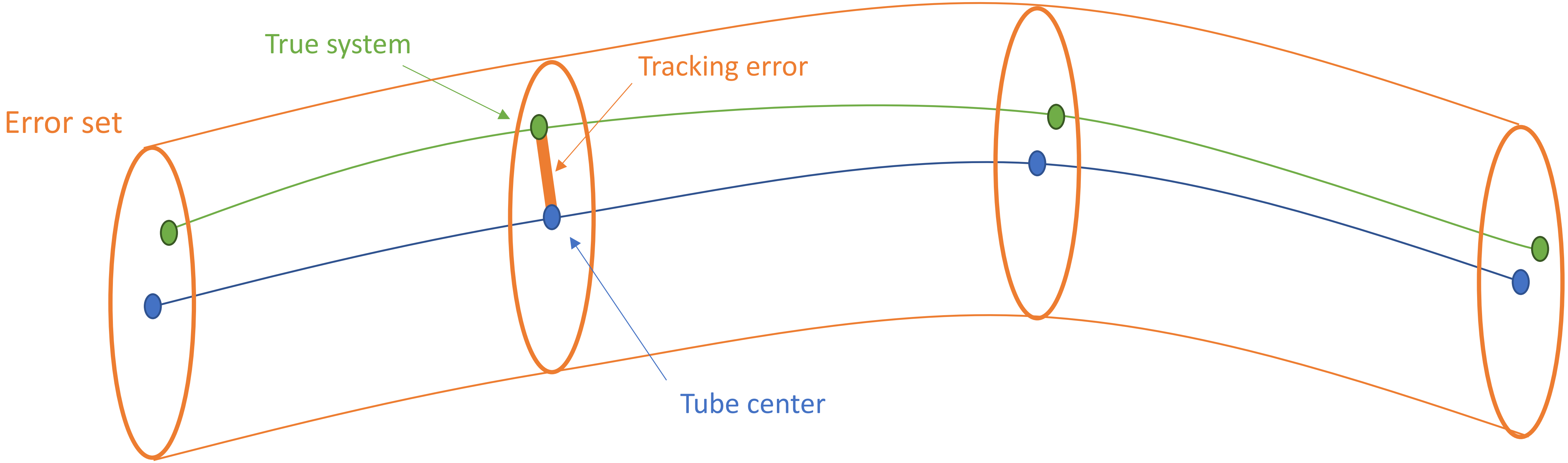


Fixed tube robust MPC

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

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





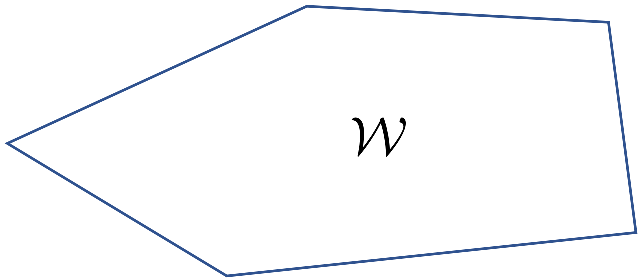
Key Ideas:

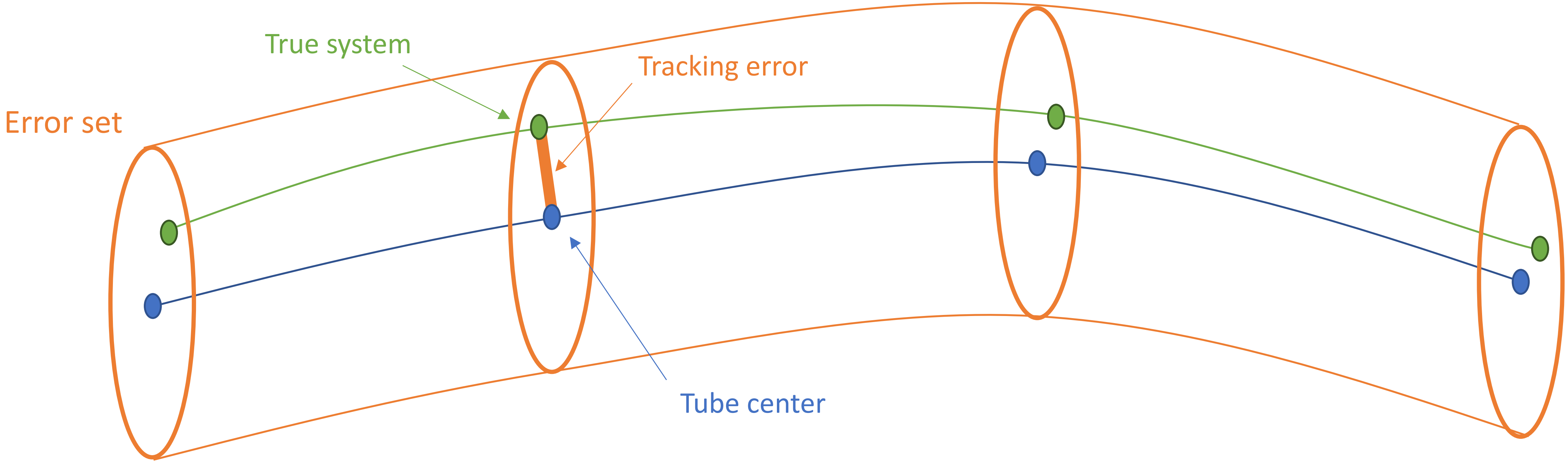
- Tube center planned via MPC
- Feedback to reduce for tracking error

Fixed tube robust MPC

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

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





Key Ideas:

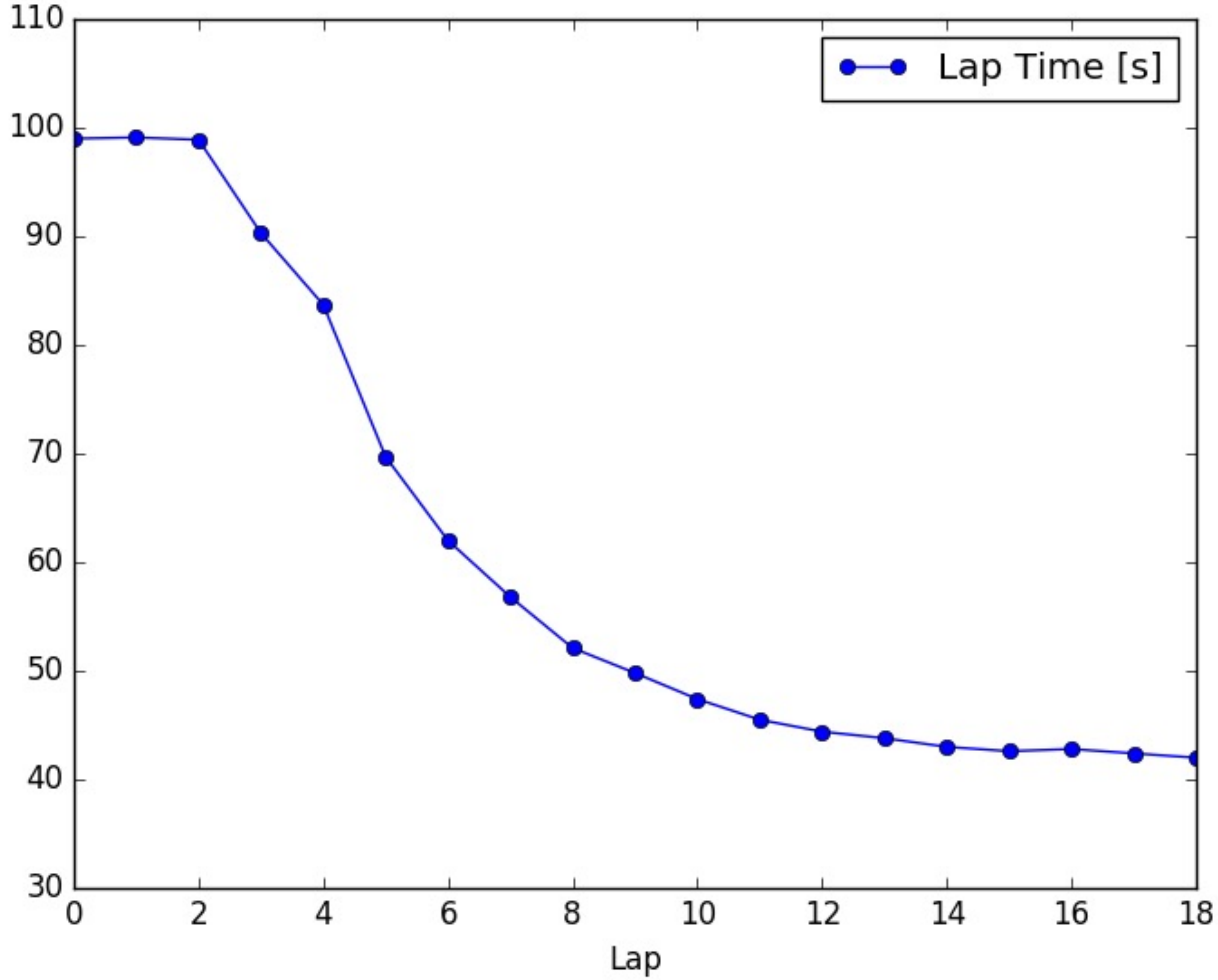
- Tube center planned via MPC
- Feedback to reduce for tracking error

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

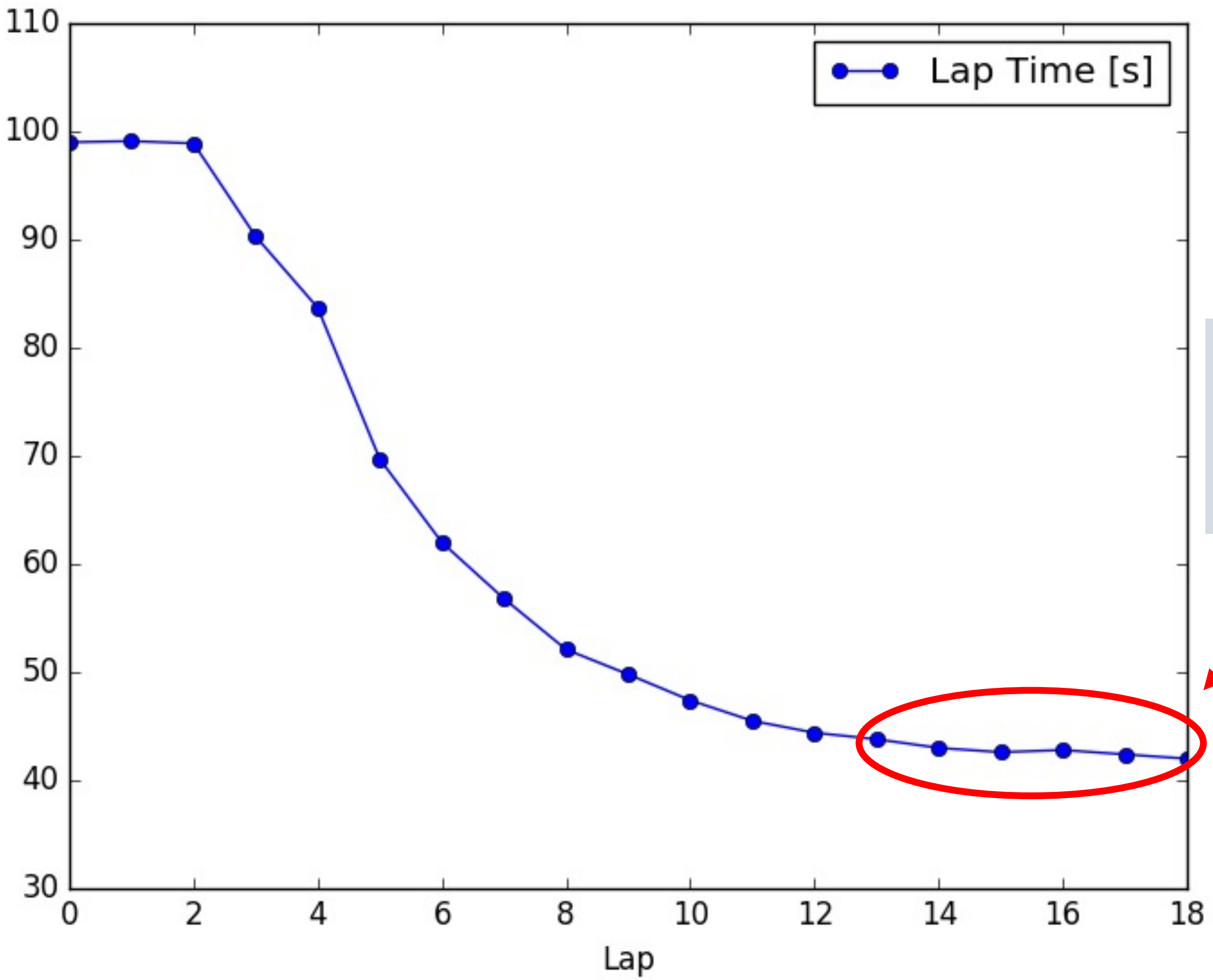
Model-free Control

How to reduce the computation load associated with MPC policies

Motivation: Lap Time Improvement

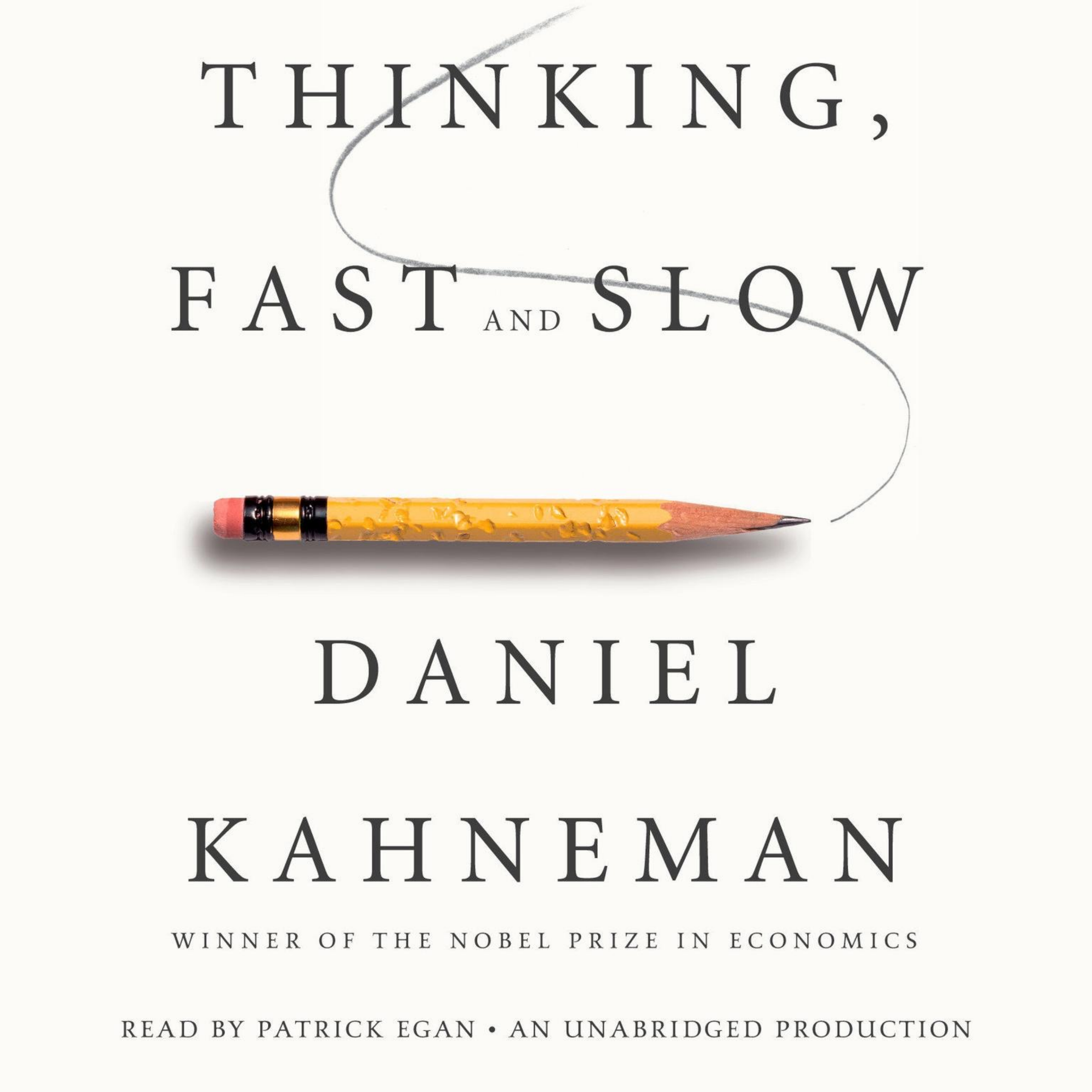


Goal: Reduce Computational Complexity

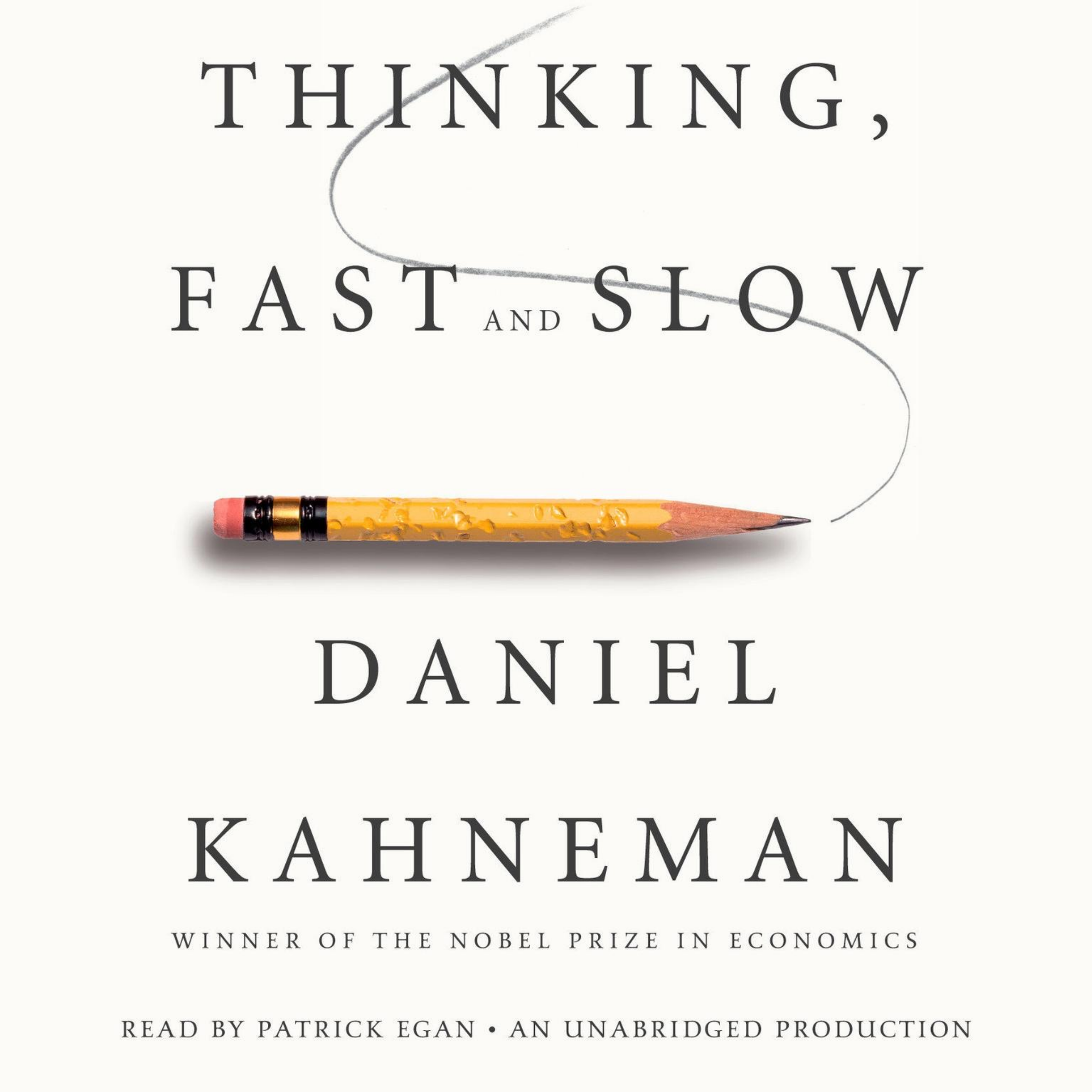


At convergence the **control policy** does not change

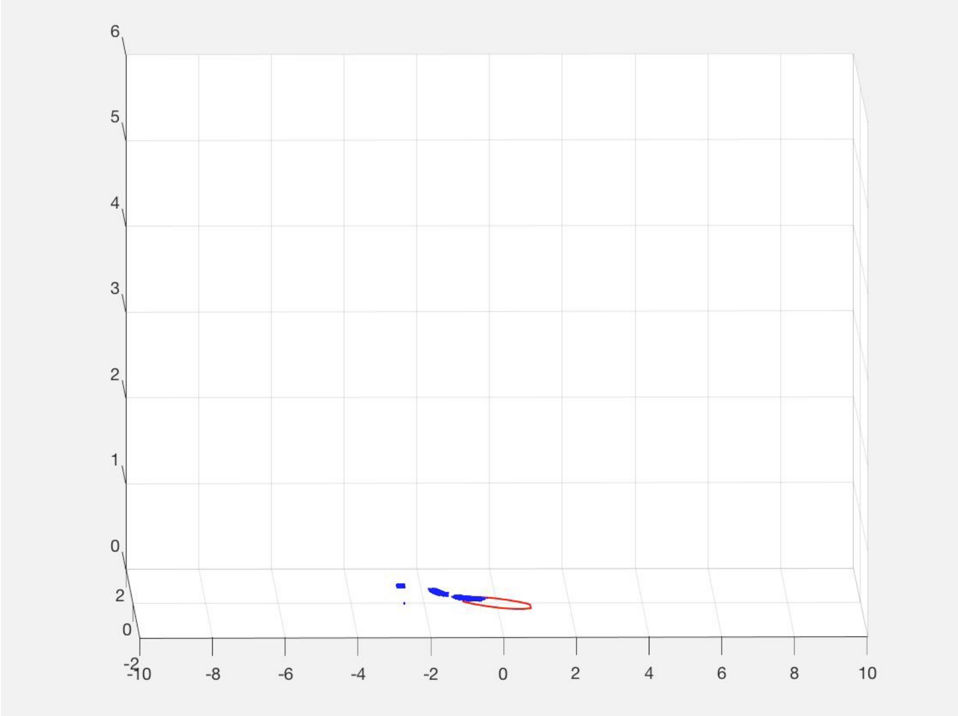
Learning as Computational Load Reduction



Learning as Computational Load Reduction

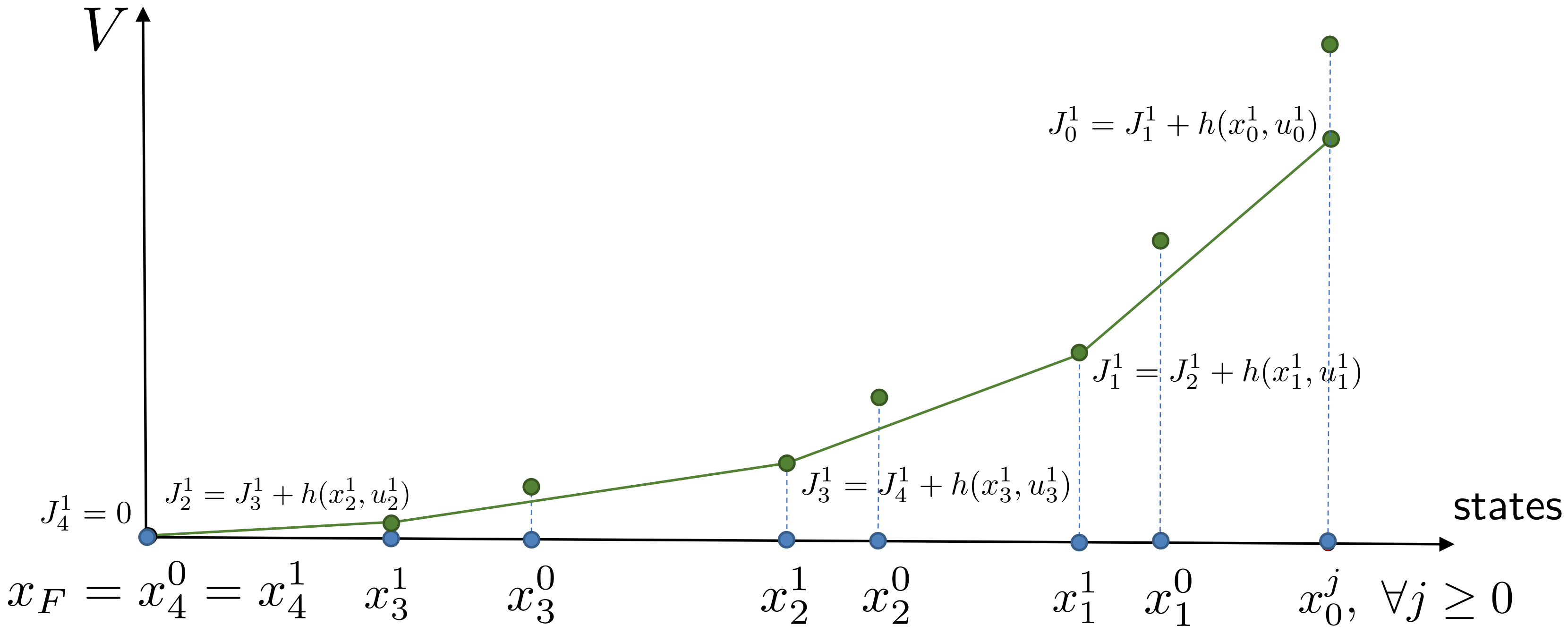


At convergence: the value function approximation does not change as we collect more data



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

Data-Based Policy



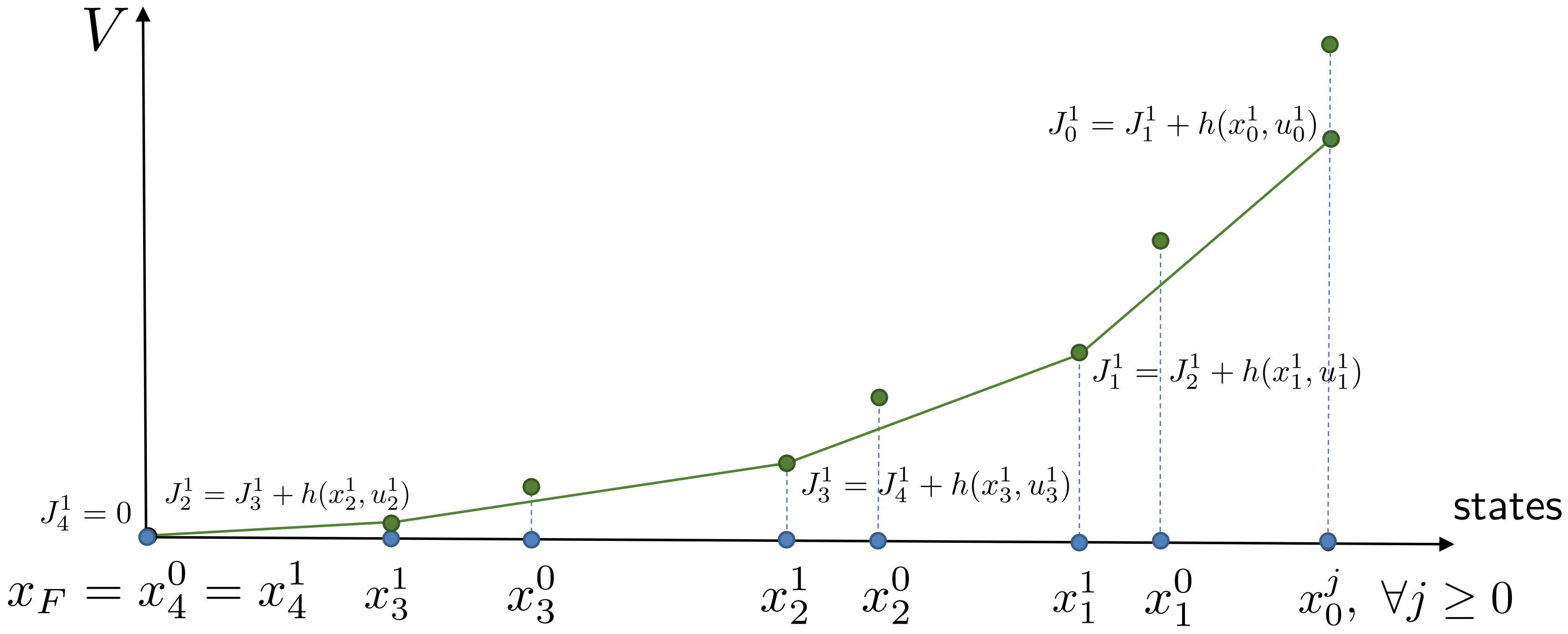
Multipliers Defining the Value Function Approximation

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

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

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

Data-Based Policy



Multipliers Defining the Value Function Approximation

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

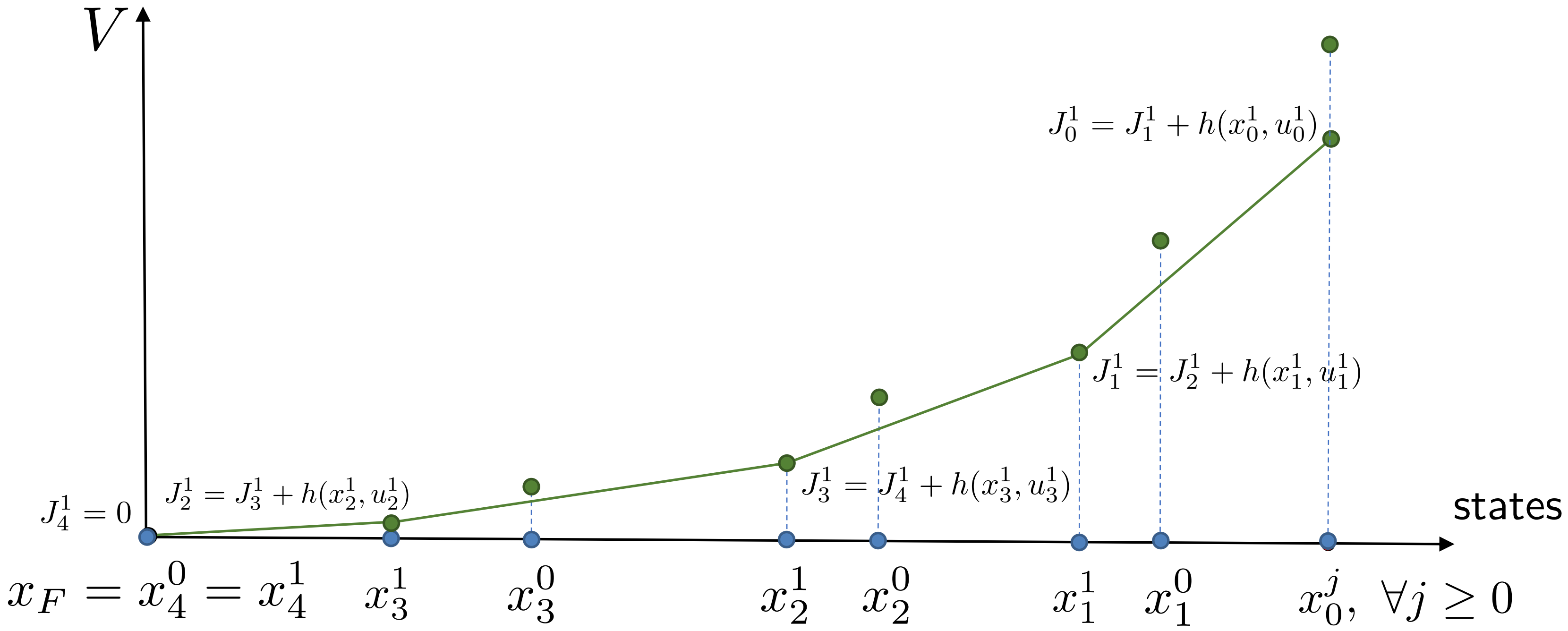
$$\text{s.t} \quad \sum_i \sum_j x_i^j \lambda_i^j = x,$$

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

Data-Based Policy

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

Data-Based Policy



Multipliers Defining the Value Function Approximation

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

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

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

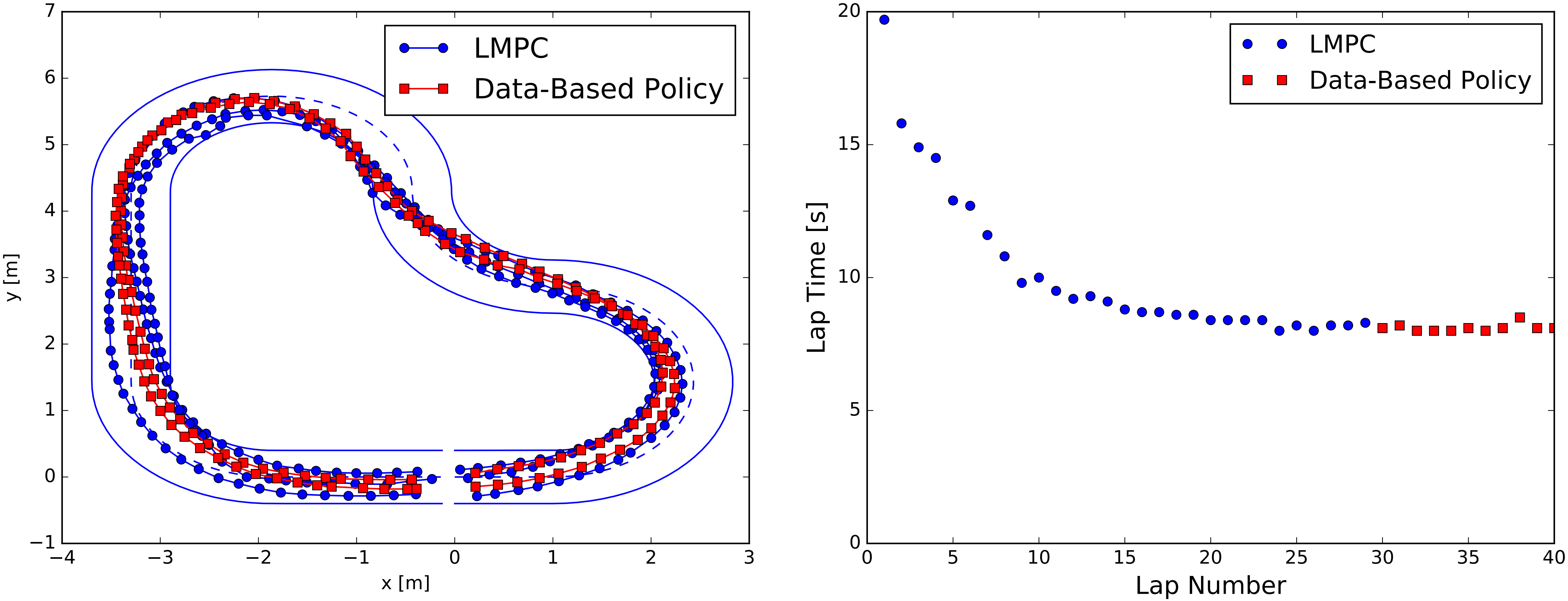
Data-Based Policy

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

Historical Data

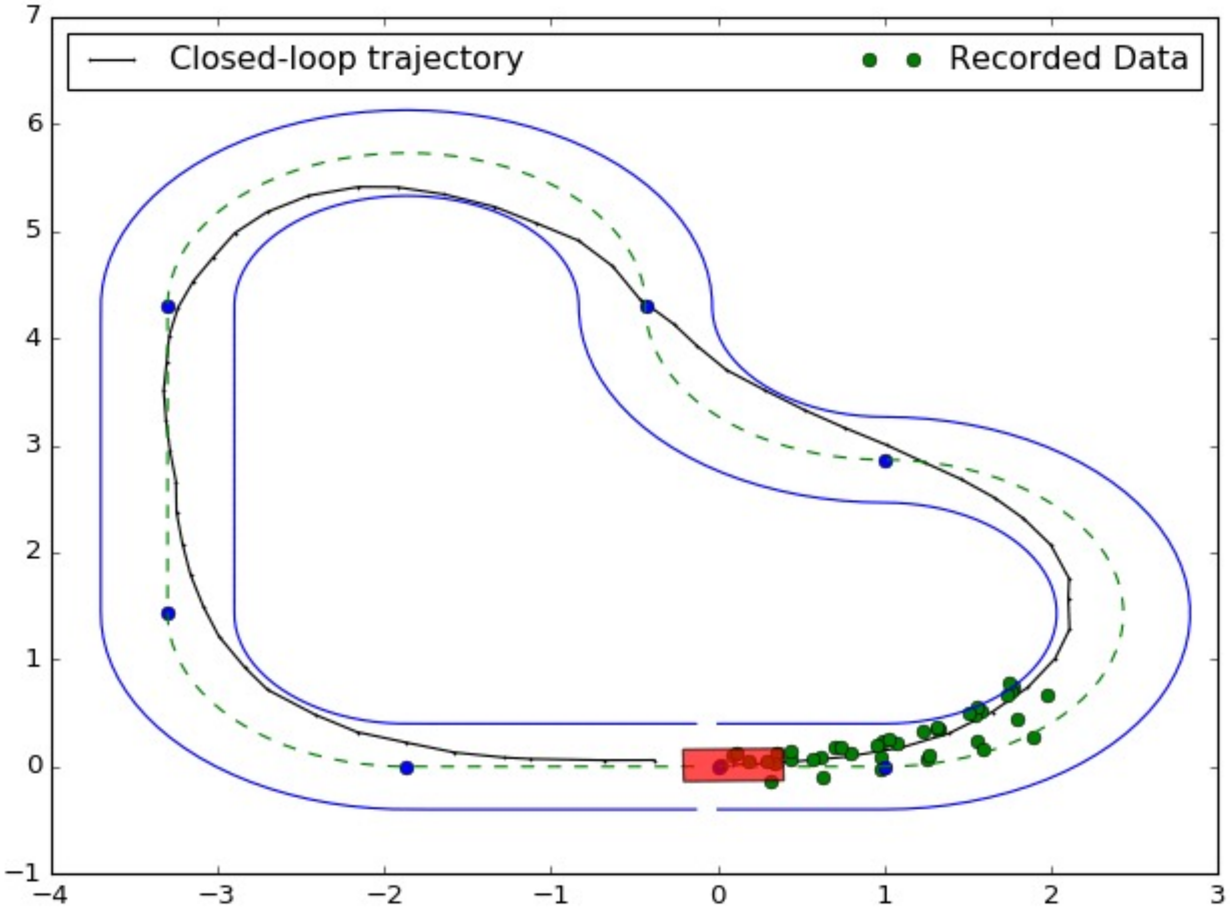
Berkeley Autonomous Race Car Implementation

Berkeley Autonomous Race Car Implementation



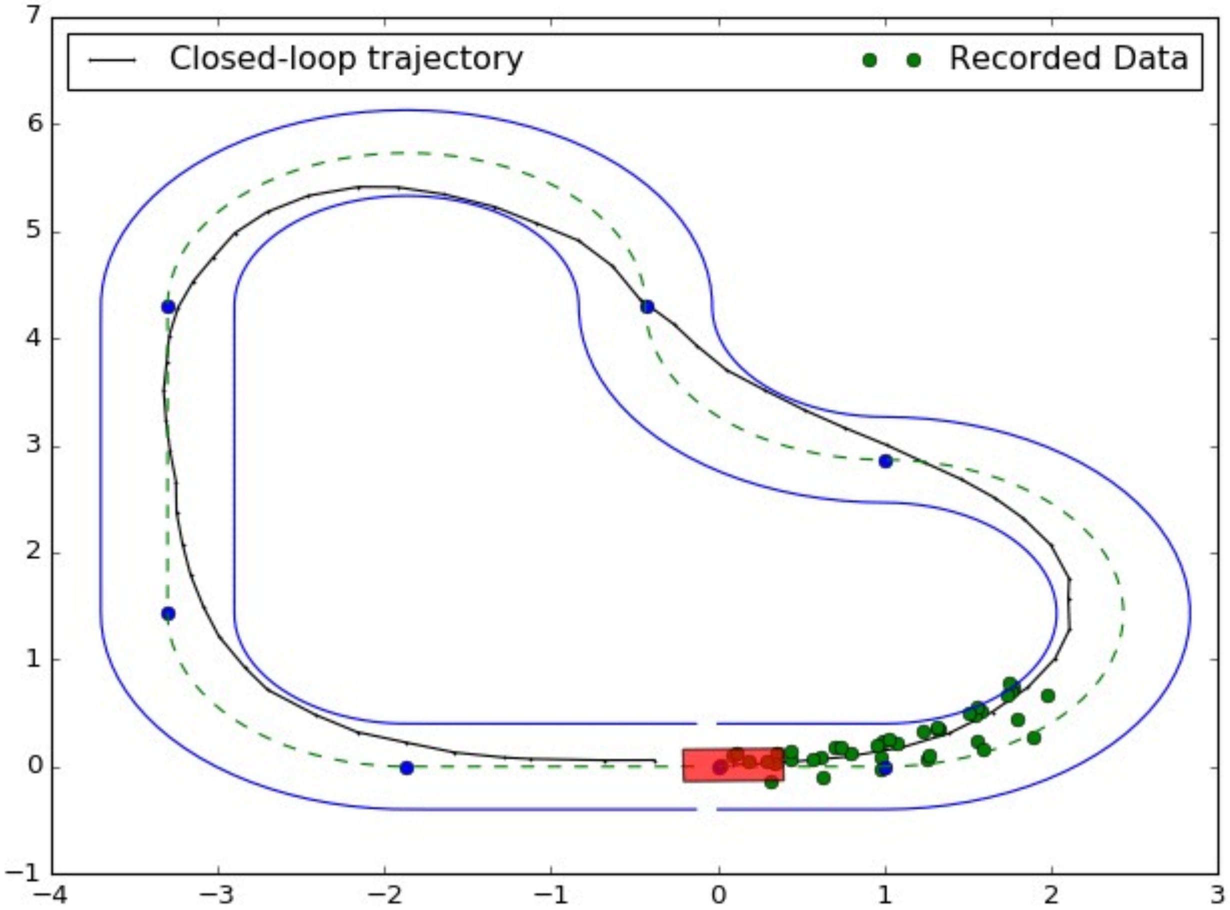
- A **reference trajectory** is **not needed** for the controller implementation
- The controller converges after few laps: the learning process is **data efficient**
- The controller **safely explores** the state space **iteratively improving** the **lap time**

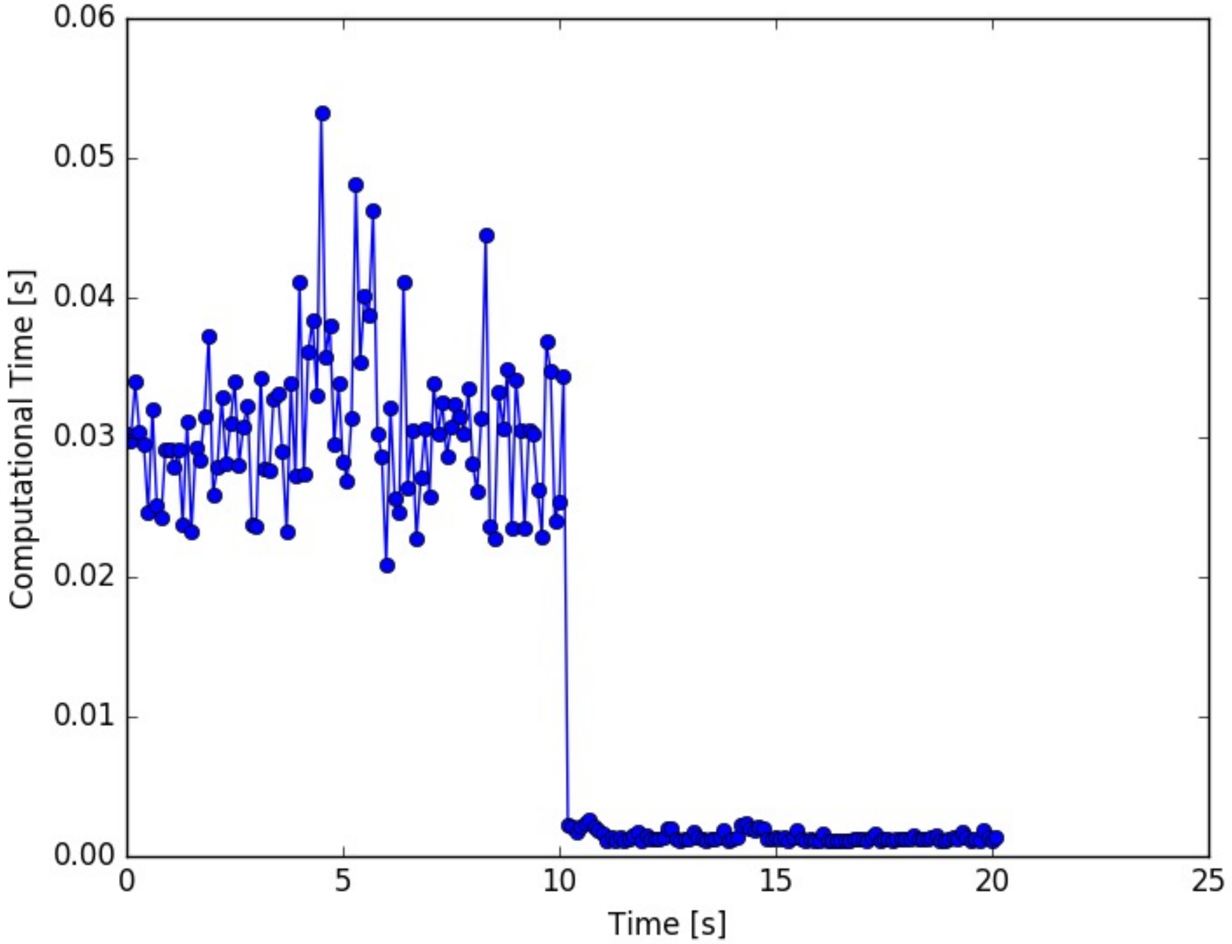
Berkeley Autonomous Race Car Implementation



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

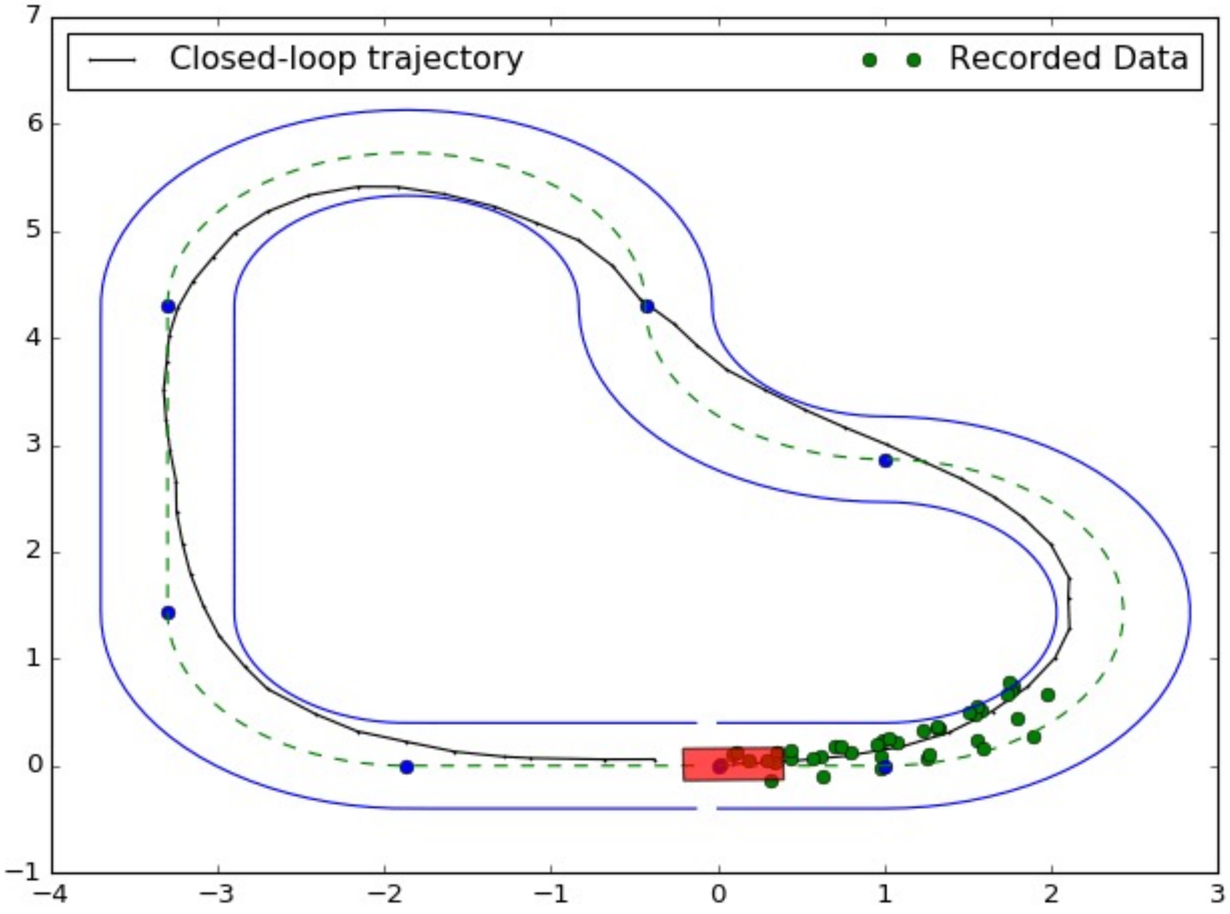
Berkeley Autonomous Race Car Implementation

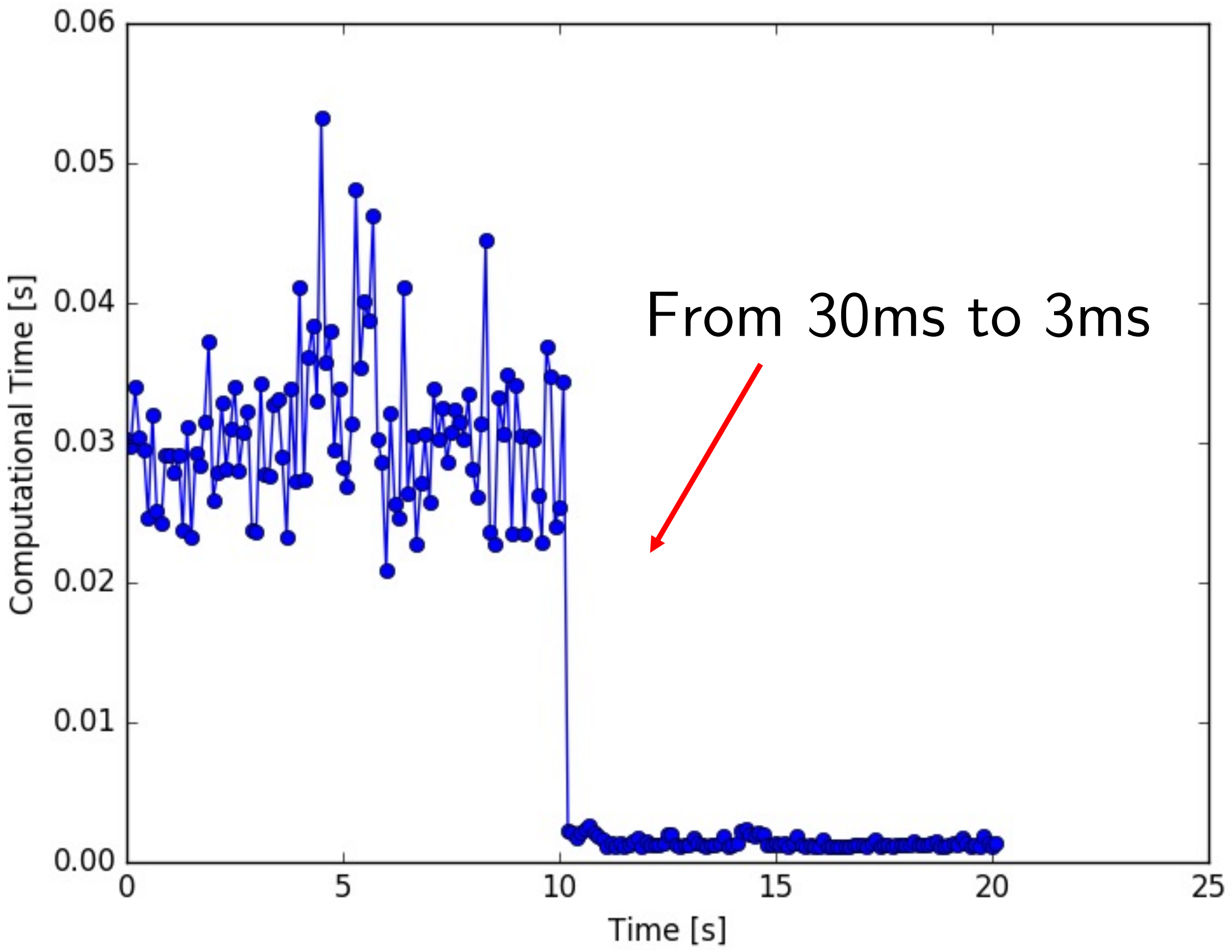




At each time t the data-based policy is computed with the K nearest neighbors (in green) to the current state. This allows us to account for the nonlinearity of the vehicle

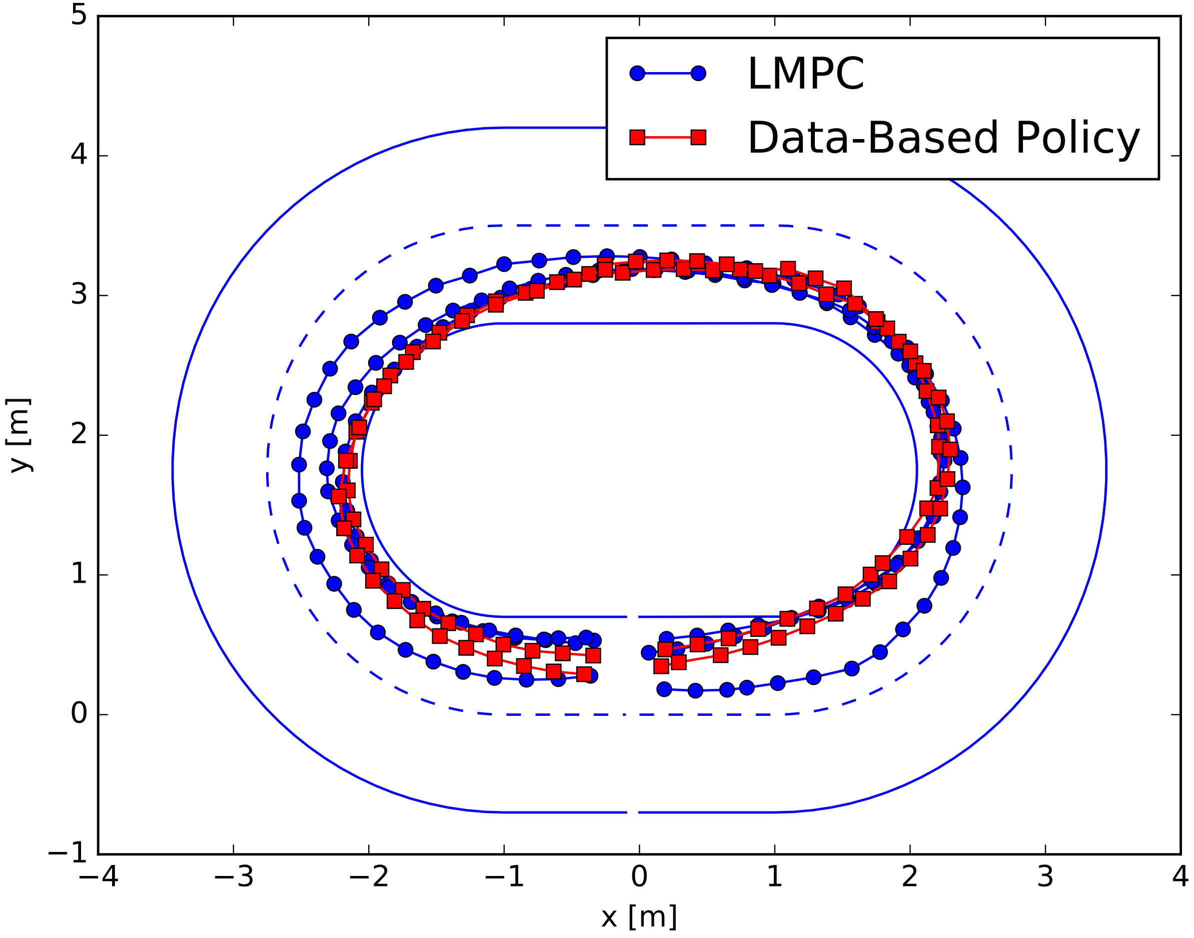
Berkeley Autonomous Race Car Implementation

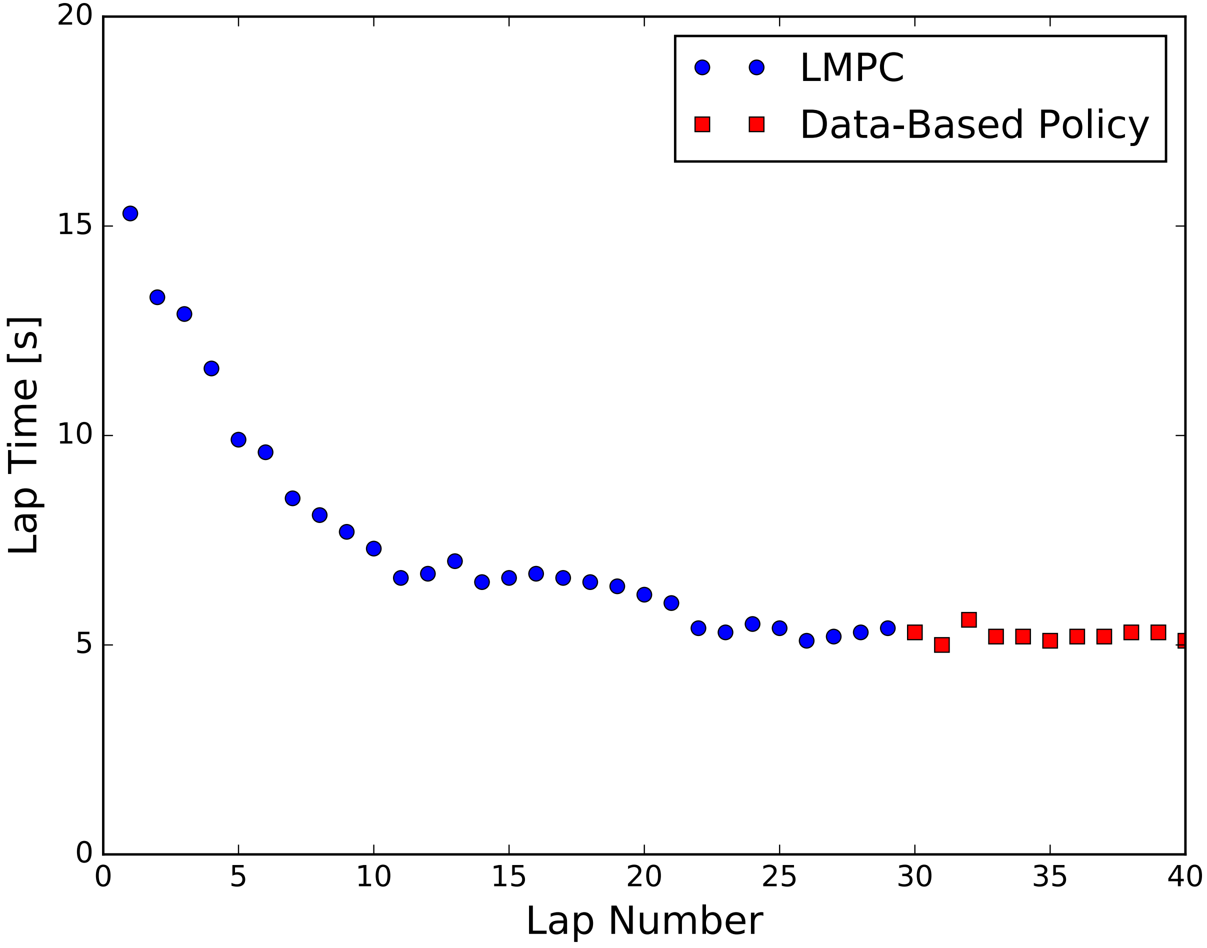




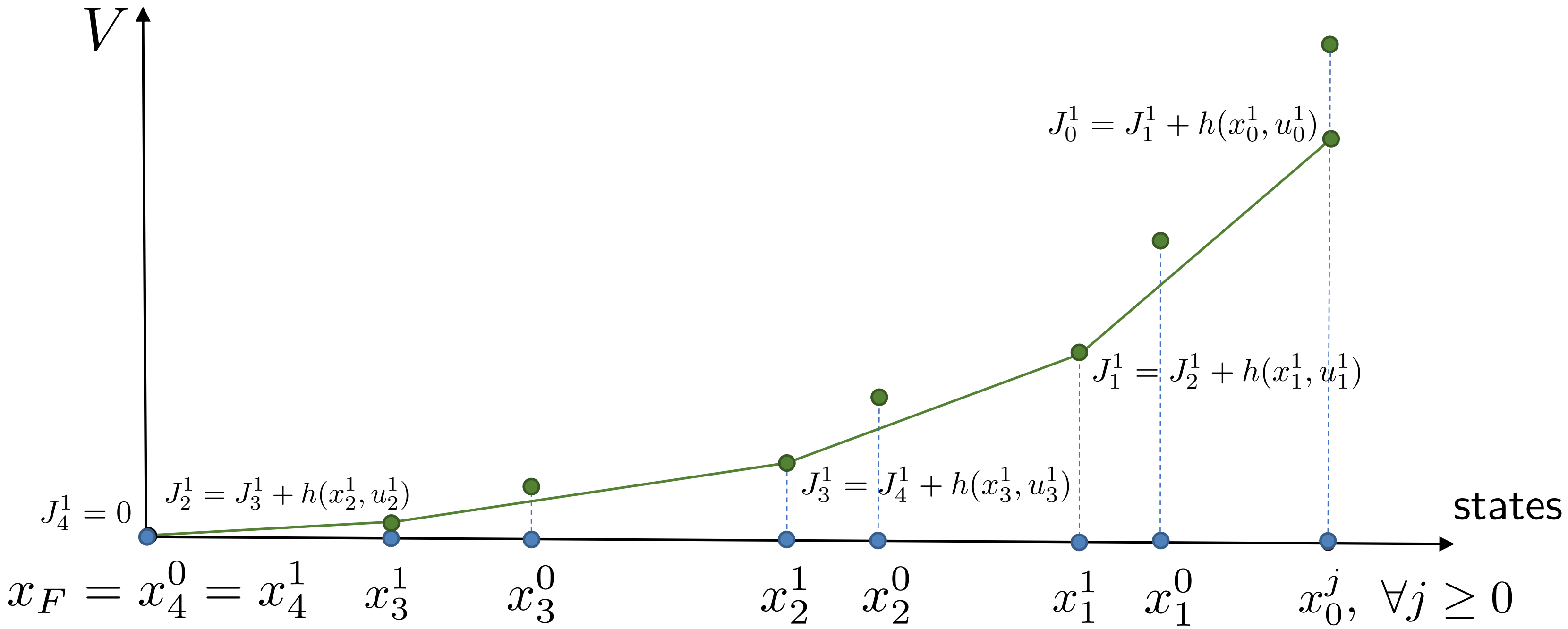
At each time t the data-based policy is computed with the K nearest neighbors (in green) to the current state. This allows us to account for the nonlinearity of the vehicle

Berkeley Autonomous Race Car Implementation





Data-Based Policy



Multipliers Defining the Value Function Approximation

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

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

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

Data-Based Policy

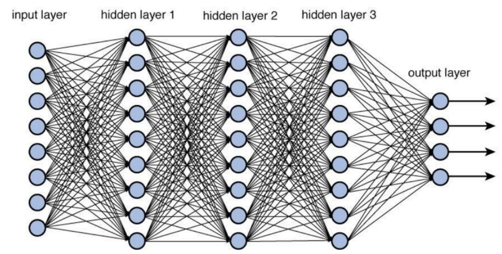
$$\kappa^j(x) = \sum_i \sum_j u_i^j \lambda_i^{j,*}$$

Historical Data

Speeding-up the solver

Offline

Primal Neural-Network

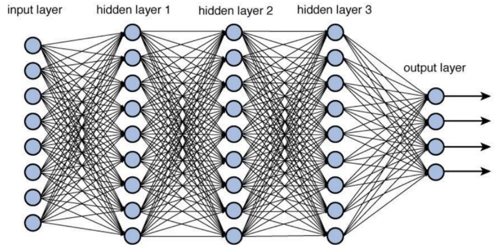


Learns input policy $\pi^p(x)$

Speeding-up the solver

Offline

Primal Neural-Network



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

Online

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

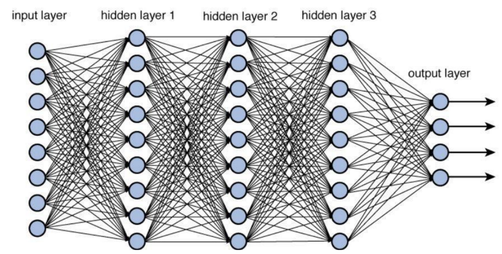
Steven Chen, Kelsey Saulnier, Nikolay Atanasov, Daniel D. Lee, Vijay Kumar, George J. Pappas, and Manfred Morari. "Approximating explicit model predictive control using constrained neural networks." In *2018 Annual American control conference (ACC)*

Primal-Dual Neural Network

Primal-Dual Neural Network

Offline

Primal Neural-Network



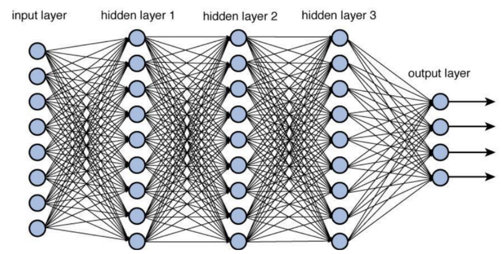
Learns input policy $\pi^p(x)$

Xiaojing Zhang, Monimoy Bujarbaruah, and Francesco Borrelli. "Safe and near-optimal policy learning for model predictive control using primal-dual neural networks". In *2019 American Control Conference (ACC) (2019)*
Xiaojing Zhang, Monimoy Bujarbaruah, and Francesco Borrelli. "Near-optimal rapid MPC using neural networks: A primal-dual policy learning framework." *IEEE Transactions on Control Systems Technology* (2020).

Primal-Dual Neural Network

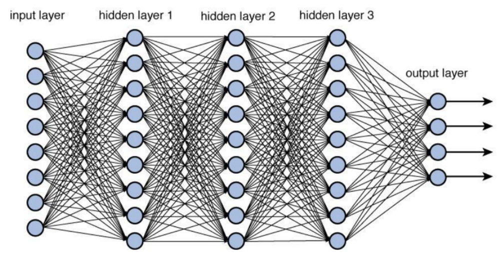
Offline

Primal Neural-Network



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

Dual Neural-Network



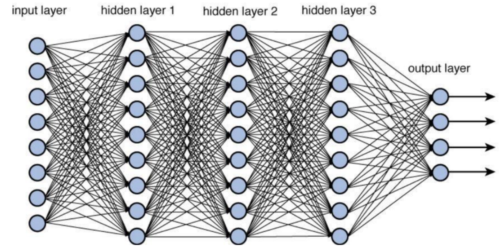
Learns optimality *certificate* $\lambda^d(x)$

Xiaojing Zhang, Monimoy Bujarbaruah, and Francesco Borrelli. "Safe and near-optimal policy learning for model predictive control using primal-dual neural networks". In *2019 American Control Conference (ACC) (2019)*
Xiaojing Zhang, Monimoy Bujarbaruah, and Francesco Borrelli. "Near-optimal rapid MPC using neural networks: A primal-dual policy learning framework." *IEEE Transactions on Control Systems Technology* (2020).

Primal-Dual Neural Network

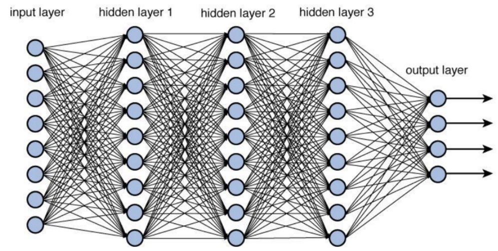
Offline

Primal Neural-Network



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

Dual Neural-Network



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

Offline Verification

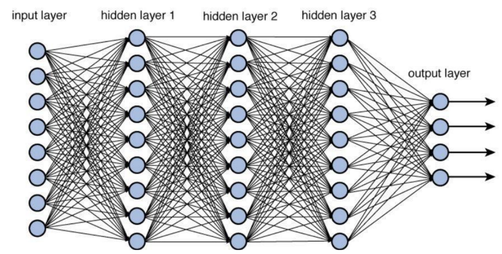
Sample at random N states and check that $|J_N(\pi^p(x) \; - J_N(\pi^p(x))| \leq \gamma$

Xiaojing Zhang, Monimoy Bujarbaruah, and Francesco Borrelli. "Safe and near-optimal policy learning for model predictive control using primal-dual neural networks". In *2019 American Control Conference (ACC) (2019)*
Xiaojing Zhang, Monimoy Bujarbaruah, and Francesco Borrelli. "Near-optimal rapid MPC using neural networks: A primal-dual policy learning framework." *IEEE Transactions on Control Systems Technology* (2020).

Primal-Dual Neural Network

Offline

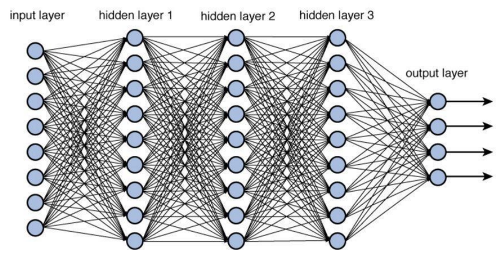
Primal Neural-Network



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 $\epsilon/2$

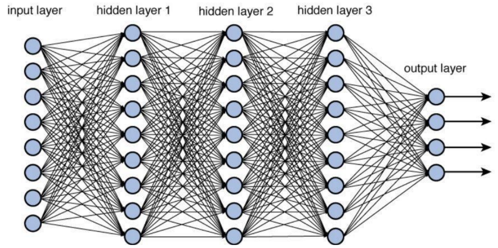
If $N \geq \frac{\ln(\frac{1}{\beta/2})}{\ln(\frac{1}{1-\epsilon/2})}$ $\Longrightarrow$ With probability at least $1-\epsilon$, learned controllers will be at most γ-suboptimal

Xiaojing Zhang., Monimoy Bujarbaruah, and Francesco Borrelli. "Safe and near-optimal policy learning for model predictive control using primal-dual neural networks". In *2019 American Control Conference (ACC) (2019)*
Xiaojing Zhang, Monimoy Bujarbaruah, and Francesco Borrelli. "Near-optimal rapid MPC using neural networks: A primal-dual policy learning framework." *IEEE Transactions on Control Systems Technology* (2020).
Tempo, Roberto, Er-Wei Bai, and Fabrizio Dabbene. "Probabilistic robustness analysis: Explicit bounds for the minimum number of samples." In *Proceedings of 35th IEEE Conference on Decision and Control*, (1996)

Primal-Dual Neural Network

Offline

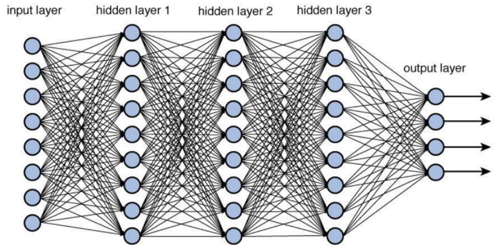
Primal Neural-Network



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

+

Dual Neural-Network



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

Online

Xiaojing Zhang., Monimoy Bujarbaruah, and Francesco Borrell. "Safe and near-optimal policy learning for model predictive control using primal-dual neural networks". In *2019 American Control Conference (ACC) (2019)*

Xiaojing Zhang, Monimoy Bujarbaruah, and Francesco Borrelli. "Near-optimal rapid MPC using neural networks: A primal-dual policy learning framework." *IEEE Transactions on Control Systems Technology* (2020).

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,

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

LMPC for uncertain systems – Problem Formulation

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

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

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

LMPC for uncertain systems – Problem Formulation

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

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

- Constraint Satisfaction:
  - Robust, $x_k \in \mathcal{X}, \; \forall w_k \in \mathcal{W} \; k \geq 0.$

- Cost: Worst Case.

LMPC for uncertain systems – Problem Formulation

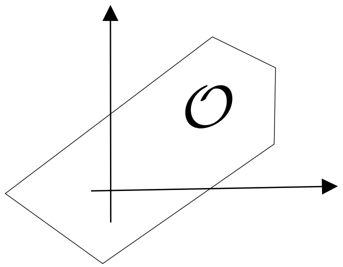
- Model Choice: Linear system subject to bounded additive uncertainty,

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

- Constraint Satisfaction:
  - Robust, $x_k \in \mathcal{X}, \; \forall w_k \in \mathcal{W} \; k \geq 0.$

- Cost: Worst Case.

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

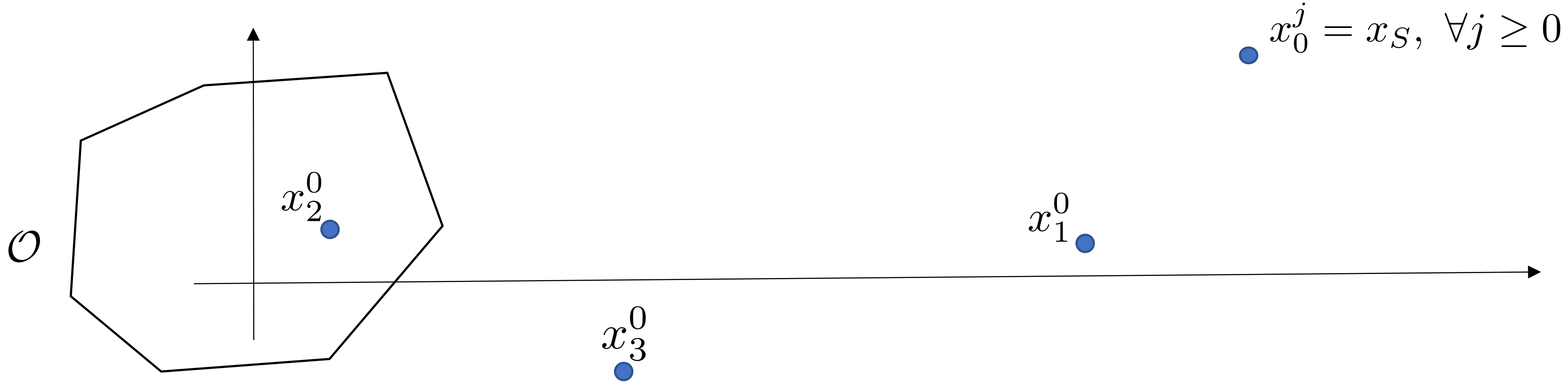


$$\forall x \in \mathcal{O} \rightarrow (A - KB)x + w \in \mathcal{O}, \; \forall w \in \mathcal{W}$$

LMPC for uncertain systems – Differences w/ Nominal

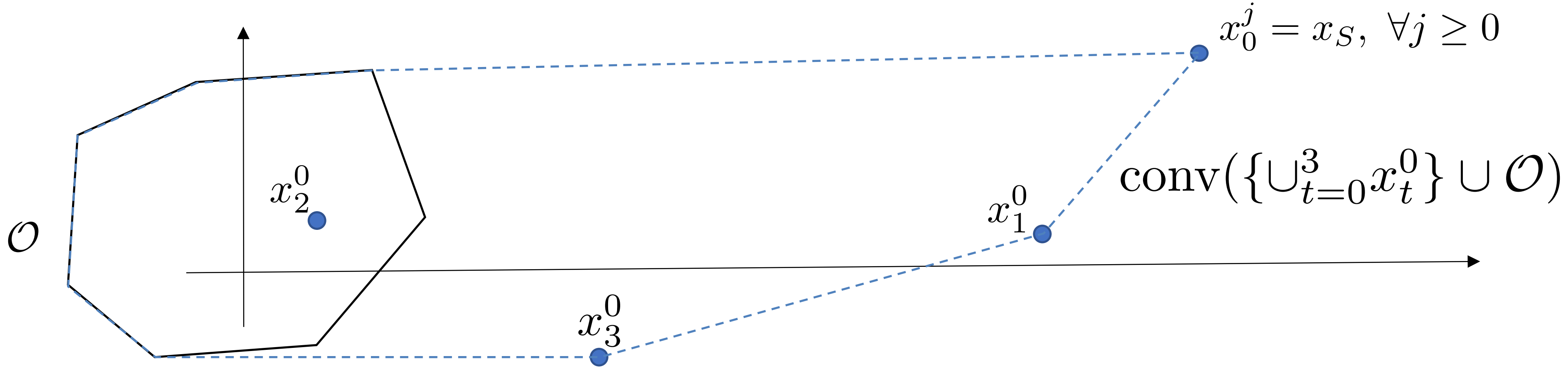
LMPC for uncertain systems – Differences w/ Nominal

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



LMPC for uncertain systems – Differences w/ Nominal

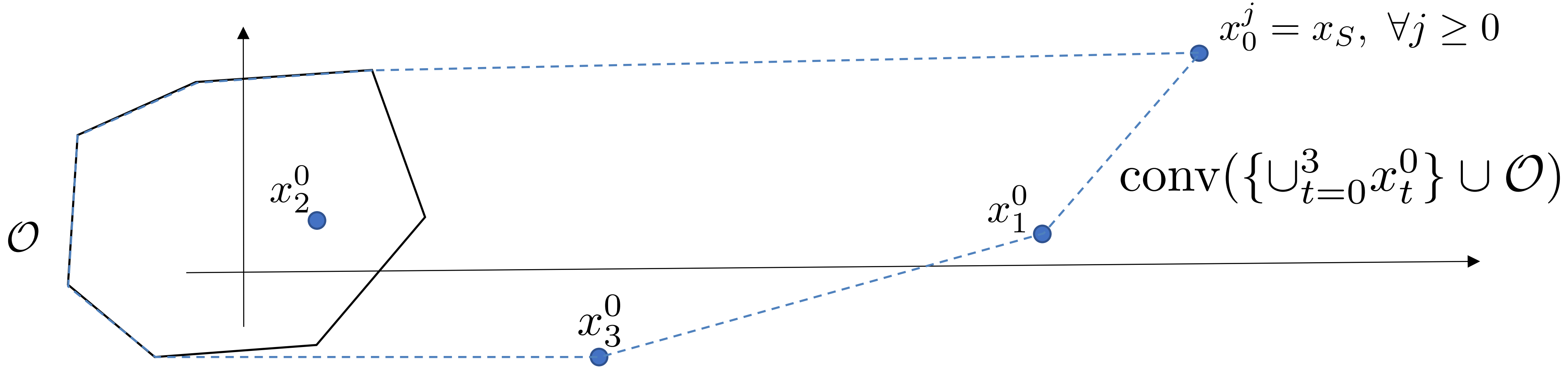
Given a feasible trajectory which drives the system to the terminal robust invariant



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



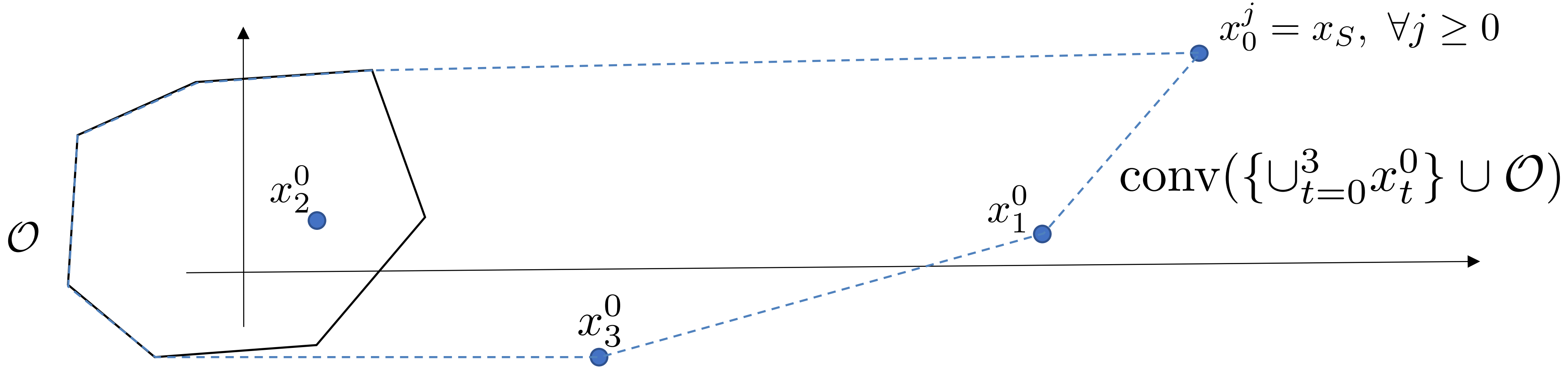
The convex hull of the stored data is **not a robust control invariant**.

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**.

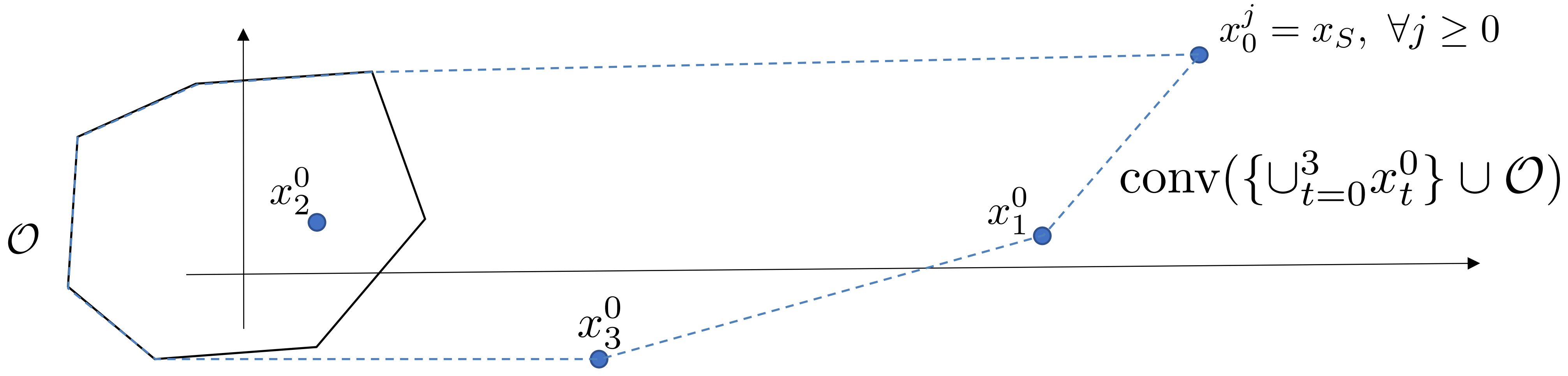
For this reason,

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

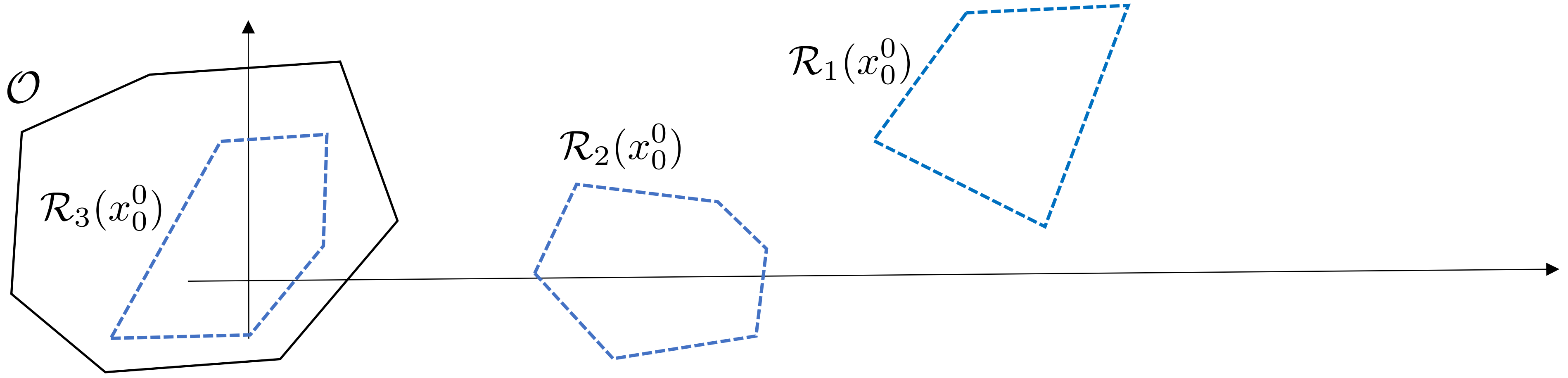
need to be designed differently in Robust LMPC.

A. Robust Safe Set

A. Robust Safe Set

Assume that given control policy $\pi^0 : \mathbb{R}^n \to \mathbb{R}^d$ robustly steers, the closed-loop system $x^0_{k+1} = Ax^0_k + B\pi^0(x^0_k) + w^0_k$ converges to the terminal set $\mathcal{O}$.

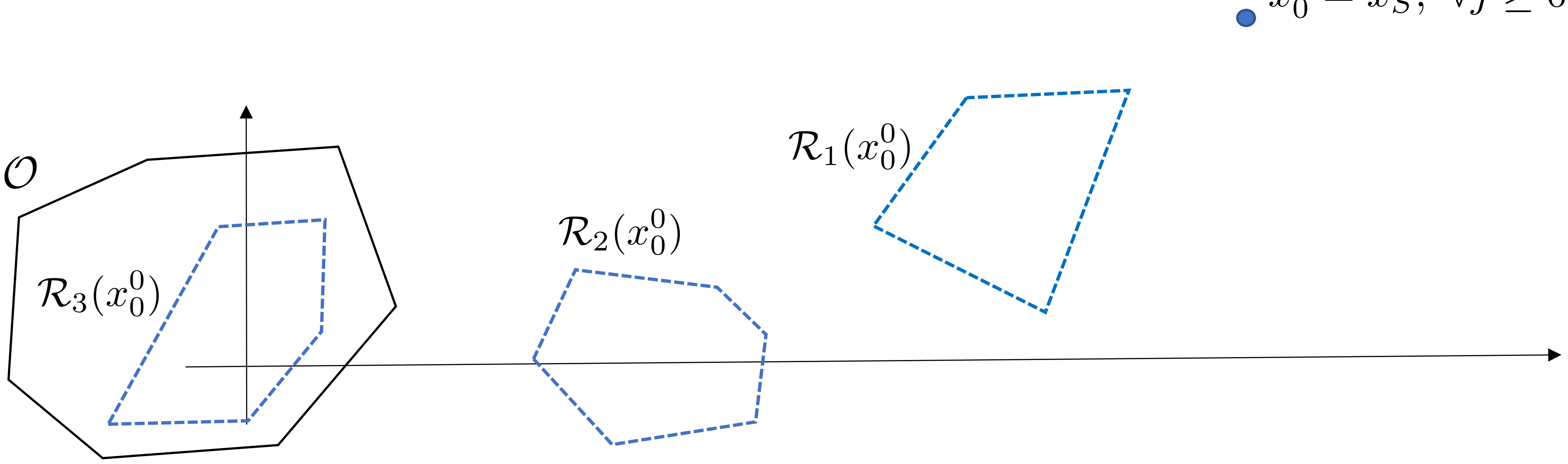
$x^j_0 = x_S, \ \forall j \geq 0$



A. Robust Safe Set

Assume that given control policy $\pi^0 : \mathbb{R}^n \to \mathbb{R}^d$ 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 $\mathcal{O}$.

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

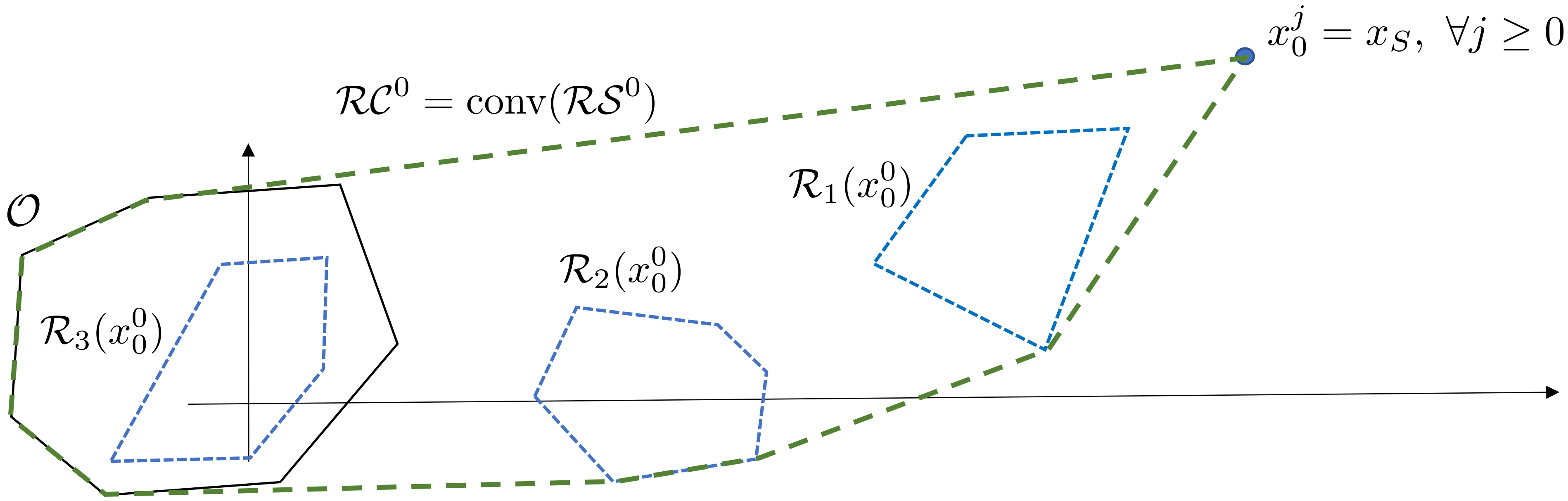


Definition: Robust Safe Set

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

A. Robust Convex Safe Set

Assume that given control policy $\pi^0 : \mathbb{R}^n \to \mathbb{R}^d$ robustly steers, the closed-loop system $x^0_{k+1} = Ax^0_k + B\pi^0(x^0_k) + w^0_k$ converges to the terminal set $\mathcal{O}$.



Definition: Robust Convex Safe Set

The robust convex safe set $\mathcal{RC}^j = \text{conv}\left(\mathcal{RS}^j\right)$ 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

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

B. Robust Q-Function

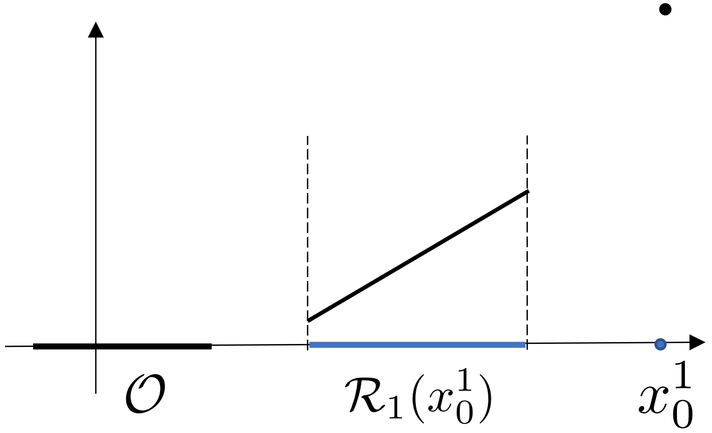
Consider the Bellman recursion for the control policy , π^j

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

Definition: Robust Q-function

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

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



B. Robust Q-Function

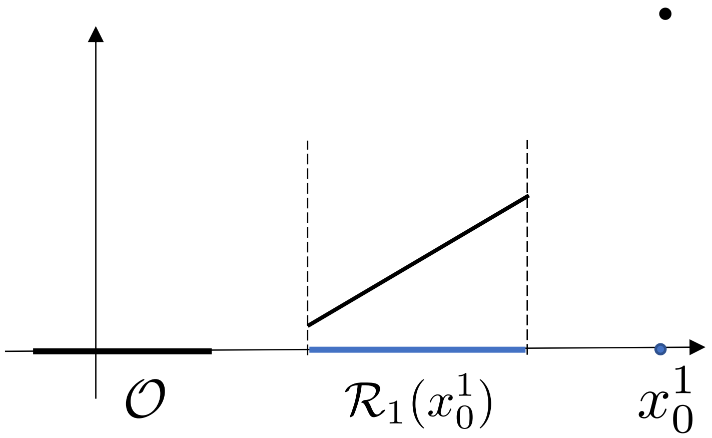
Consider the Bellman recursion for the control policy , π^j

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

Definition: Robust Q-function

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

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

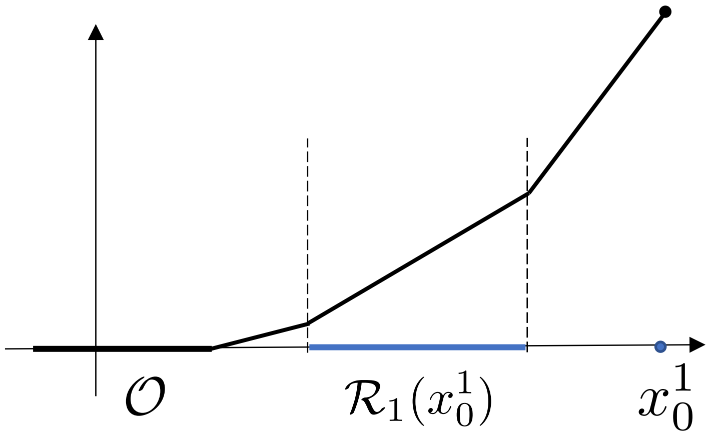


Definition: Robust Convex Q-function

The robust convex Q-function defined as

$$Q^j_{rc}(x) = \min_{\mu} \mu \mid (x,\mu) \in \mathrm{conv}\Big(\bigcup_{k=0}^{j} \mathrm{epi}(Q^j_r(x))\Big),$$

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)

$$
\begin{aligned}
J^{\mathrm{LMPC},j}_{t\rightarrow t+N}(x_t^j) = \min_{\boldsymbol{\pi}_t^j(\cdot)} \max_{\bar{\mathbf{w}}_t^j} \; & \Big[\sum_{k=t}^{t+N-1} h(x^j_{k|t}, u^j_{k|t}) + Q^{j-1}_{rc}(x^j_{t+N|t}) \Big] \\
& x^j_{k+1|t} = A x^j_{k|t} + B u^j_{k|t} + \bar{w}^j_{k|t} \\
& u^j_{k|t} = \pi^j_{k|t}(x^j_{k|t}) \\
& x^j_{k|t} \in \mathcal{X}, u^j_{k|t} \in \mathcal{U} \\
& x^j_{t+N|t} \in \mathcal{RC}^{j-1}, \;\; \forall \bar{w}^j_{k|t} \in \mathcal{W}, k \in \{t, \dots, t+N\}
\end{aligned}
$$

for $\boldsymbol{\pi}_t^j(\cdot) = [\pi^j_{t|t}(\cdot), \dots, \pi^j_{t+N|t}(\cdot)]$ and $\bar{\mathbf{w}}_t^j = [\bar{w}^j_{t|t}, \dots, \bar{w}^j_{t+N|t}]$

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

Robust Learning Model Predictive Control

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

$$J^{\mathrm{LMPC},j}_{t\rightarrow t+N}(x^j_t) = \min_{\boldsymbol{\pi}^j_t(\cdot)} \max_{\bar{\mathbf{w}}^j_t} \Big[\sum_{k=t}^{t+N-1} h(x^j_{k|t}, u^j_{k|t}) + Q^{j-1}_{rc}(x^j_{t+N|t}) \Big]$$

$$x^j_{k+1|t} = Ax^j_{k|t} + Bu^j_{k|t} + \bar{w}^j_{k|t}$$

$$u^j_{k|t} = \pi^j_{k|t}(x^j_{k|t})$$

$$x^j_{k|t} \in \mathcal{X}, u^j_{k|t} \in \mathcal{U}$$

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

Iteratively constructed using the system model

for $\boldsymbol{\pi}^j_t(\cdot) = [\pi^j_{t|t}(\cdot), \dots, \pi^j_{t+N|t}(\cdot)]$ and $\bar{\mathbf{w}}^j_t = [\bar{w}^j_{t|t}, \dots, \bar{w}^j_{t+N|t}]$

Then apply to the system $u^j_t = \pi^{j,*}_{t|t}(x^j_t)$

Robust Learning Model Predictive Control – Remarks

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

- Computed analytically (hard even for linear systems)

Robust Learning Model Predictive Control – Remarks

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

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

Robust Learning Model Predictive Control – Remarks

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

- Computed analytically (hard even for linear systems)
- Computed propagating the vertices of the disturbance $\mathcal{W} \times \ldots \times \mathcal{W}$ (explodes!)
- Approximated for a large number of rollout N_{roll}. Then, we could guarantee

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

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

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

Robust Learning Model Predictive Control – Remarks

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

- Computed analytically (hard even for linear systems)
- Computed propagating the vertices of the disturbance $\mathcal{W} \times \ldots \times \mathcal{W}$ (explodes!)
- Approximated for a large number of rollout N_{roll}. Then, we could guarantee

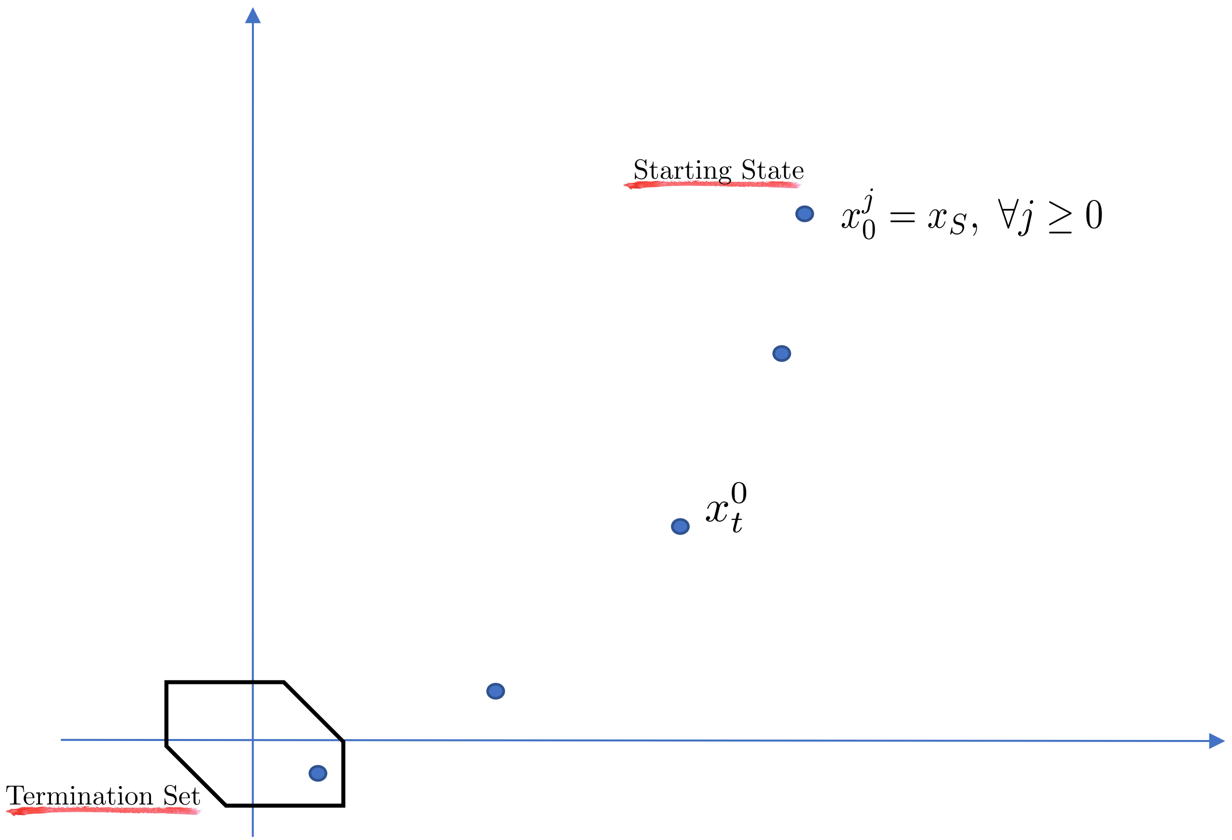
$$\mathbb{P}(Ax + Bu + w \in \mathcal{CS}^j | x \in \mathcal{CS}^j) \geq 1 - \epsilon(N_{roll})$$

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

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

Iterative approximation strategy based on historical data

Example: Constrained LQR Uncertain System



Control Problem

Let

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

be the solution to the Bellman recursion for the control policy $\pi^j(\cdot)$. We want to solve the following finite time robust optimal control problem

$$J^{j,*}_{0 \to T^j}(x_0^j) = \min_{\pi^j(\cdot)} \quad J^j_{\pi^j}(x_0^j)$$

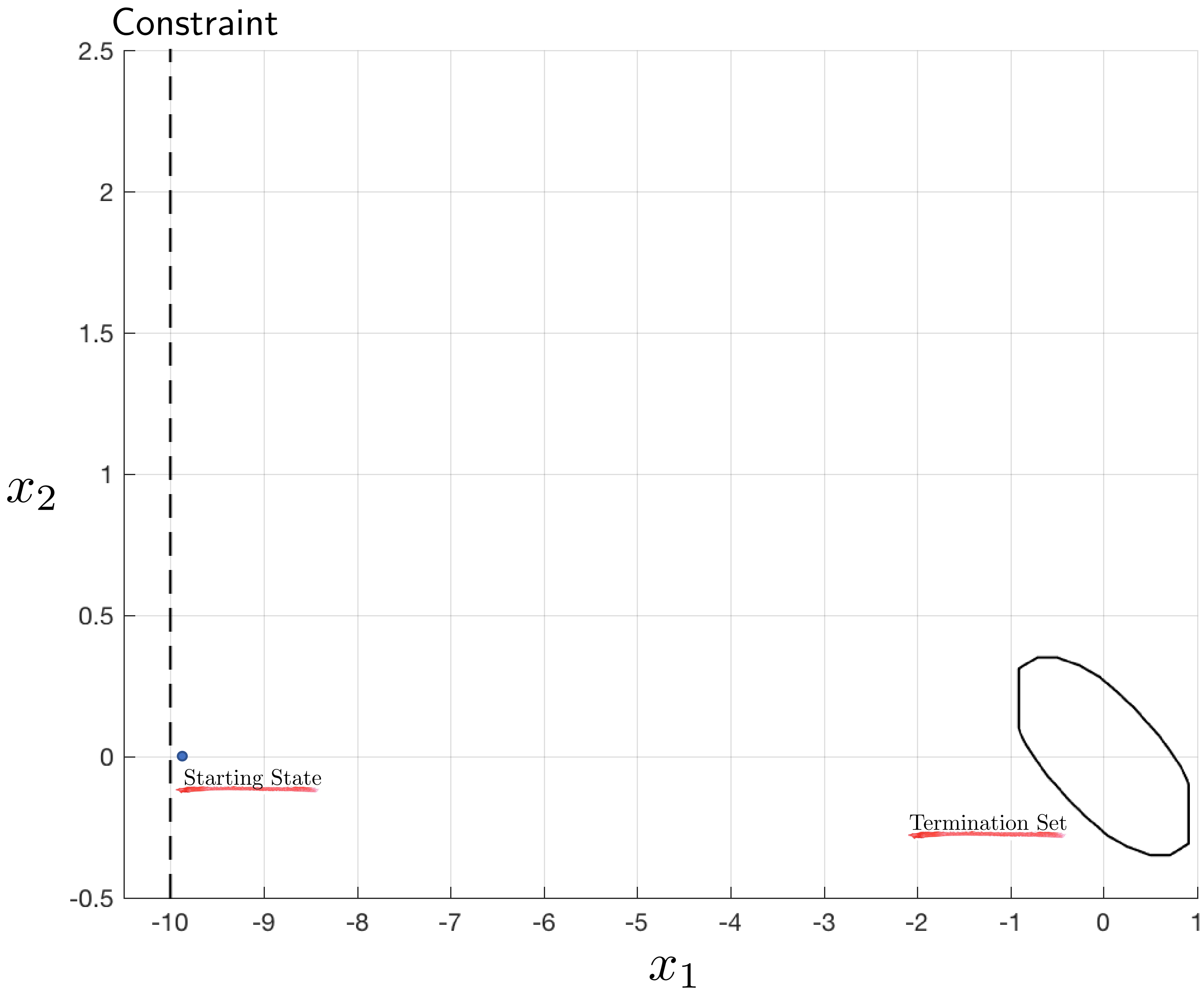
$$x^j_{k+1} = Ax^j_k + B\pi^j(x^j_k) + w^j_k$$

$$u^j_k = \pi^j(x^j_k)$$

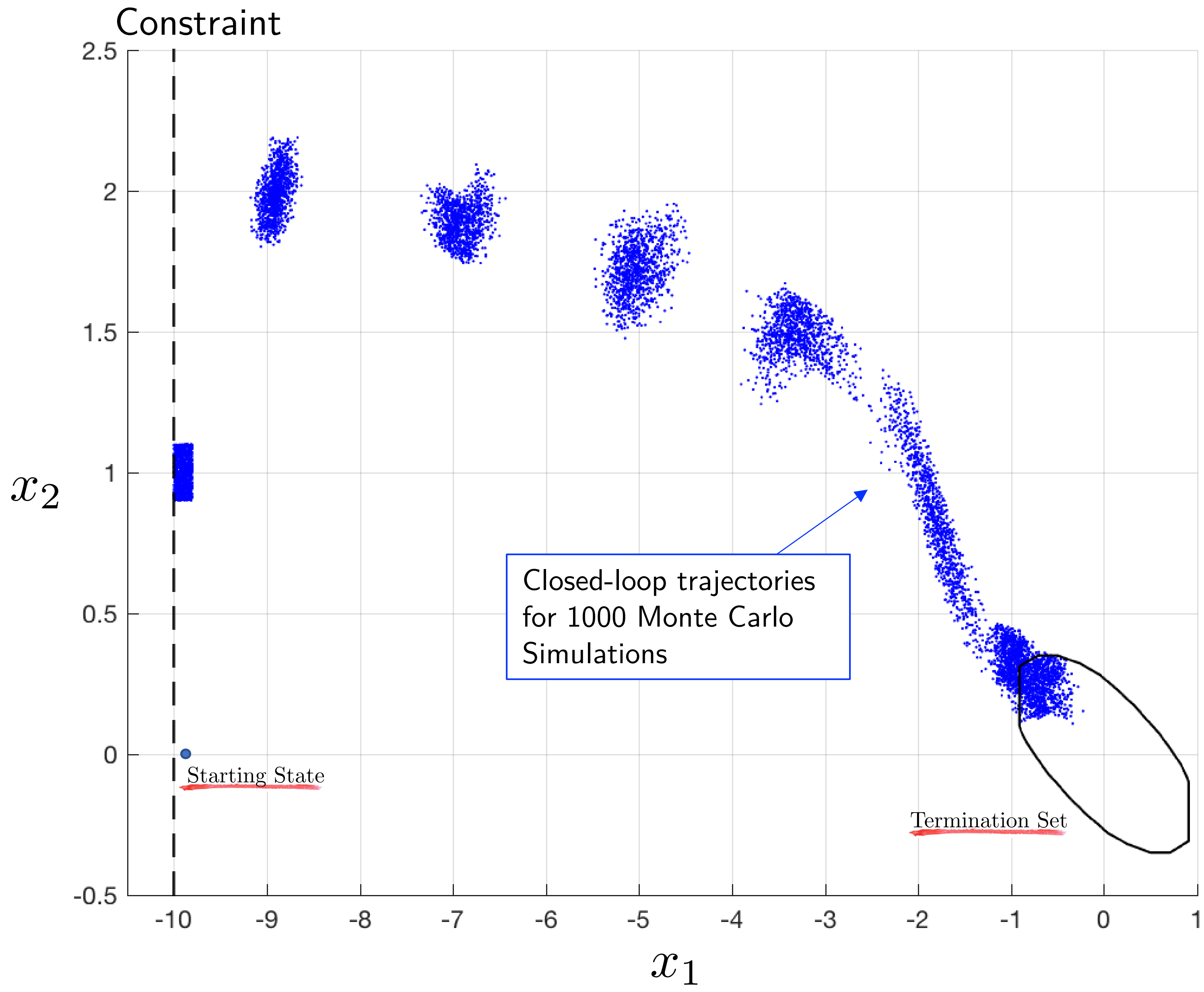
$$x^j_k \in \mathcal{X}, u^j_k \in \mathcal{U}, x^j_{T^j} \in \mathcal{O}$$

$$\forall w^j_k \in \mathcal{W}, k \in \{0, \ldots, T^j\}$$

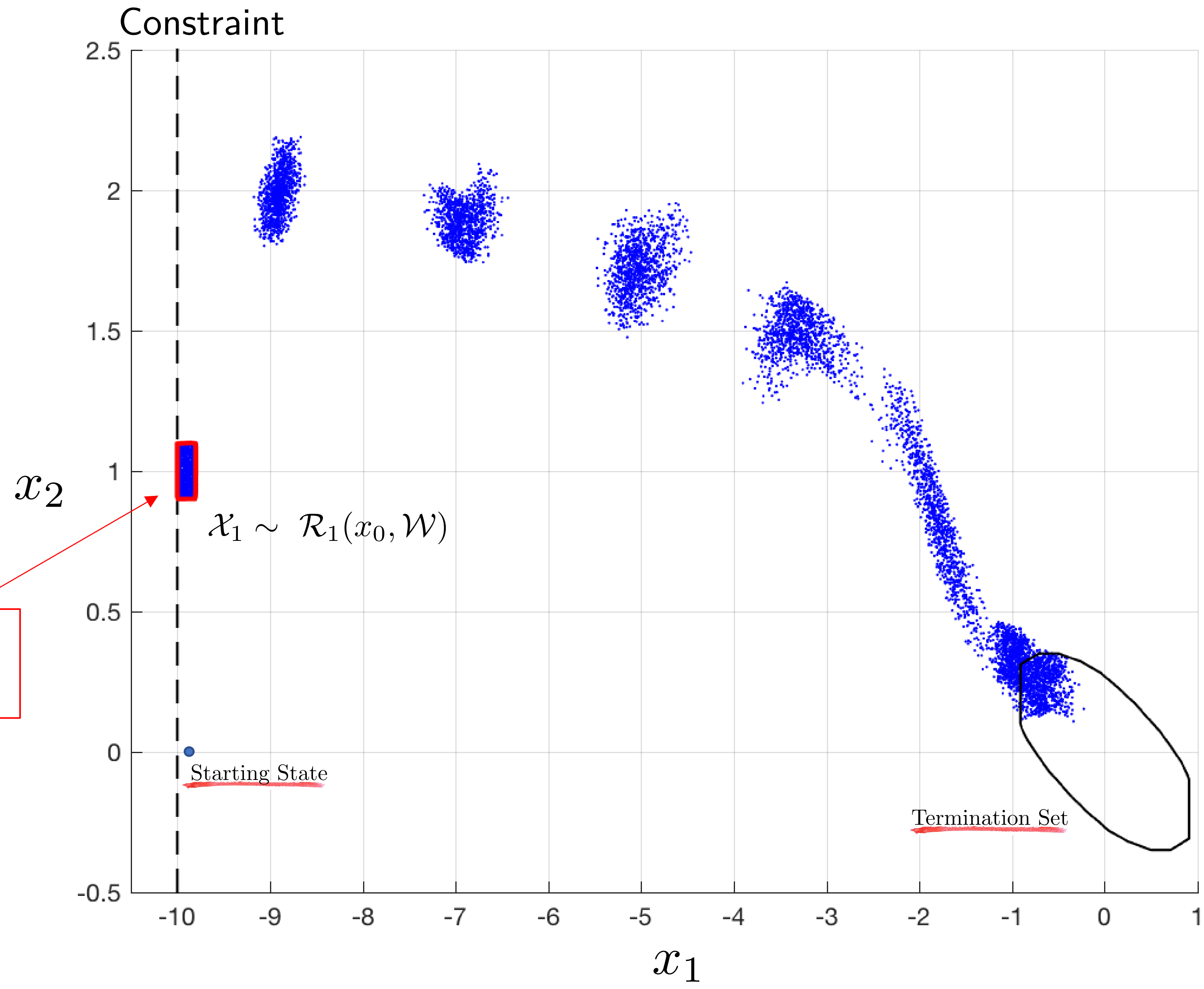
Example: Constrained LQR Uncertain System



Example: Constrained LQR Uncertain System

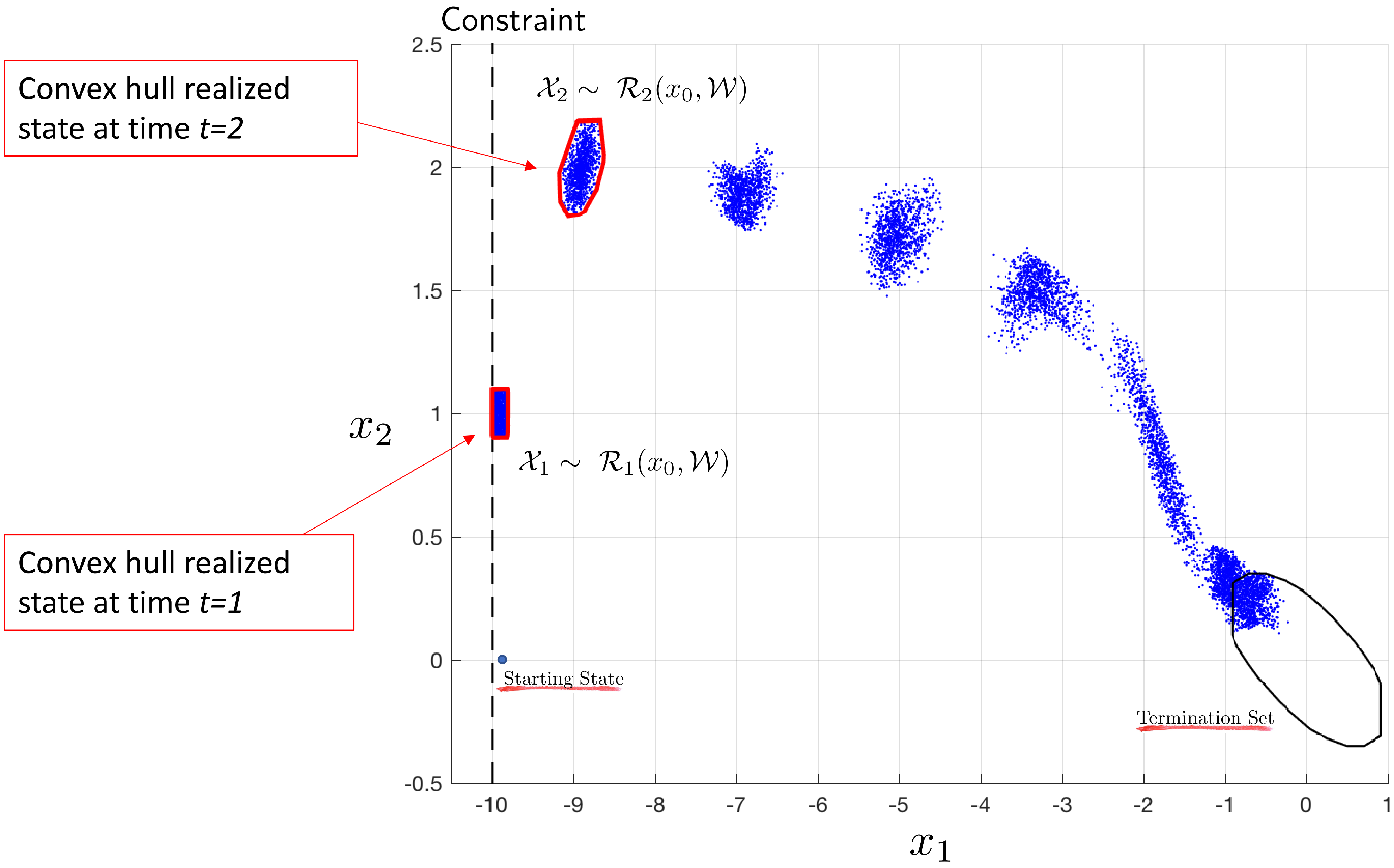


Example: Constrained LQR Uncertain System



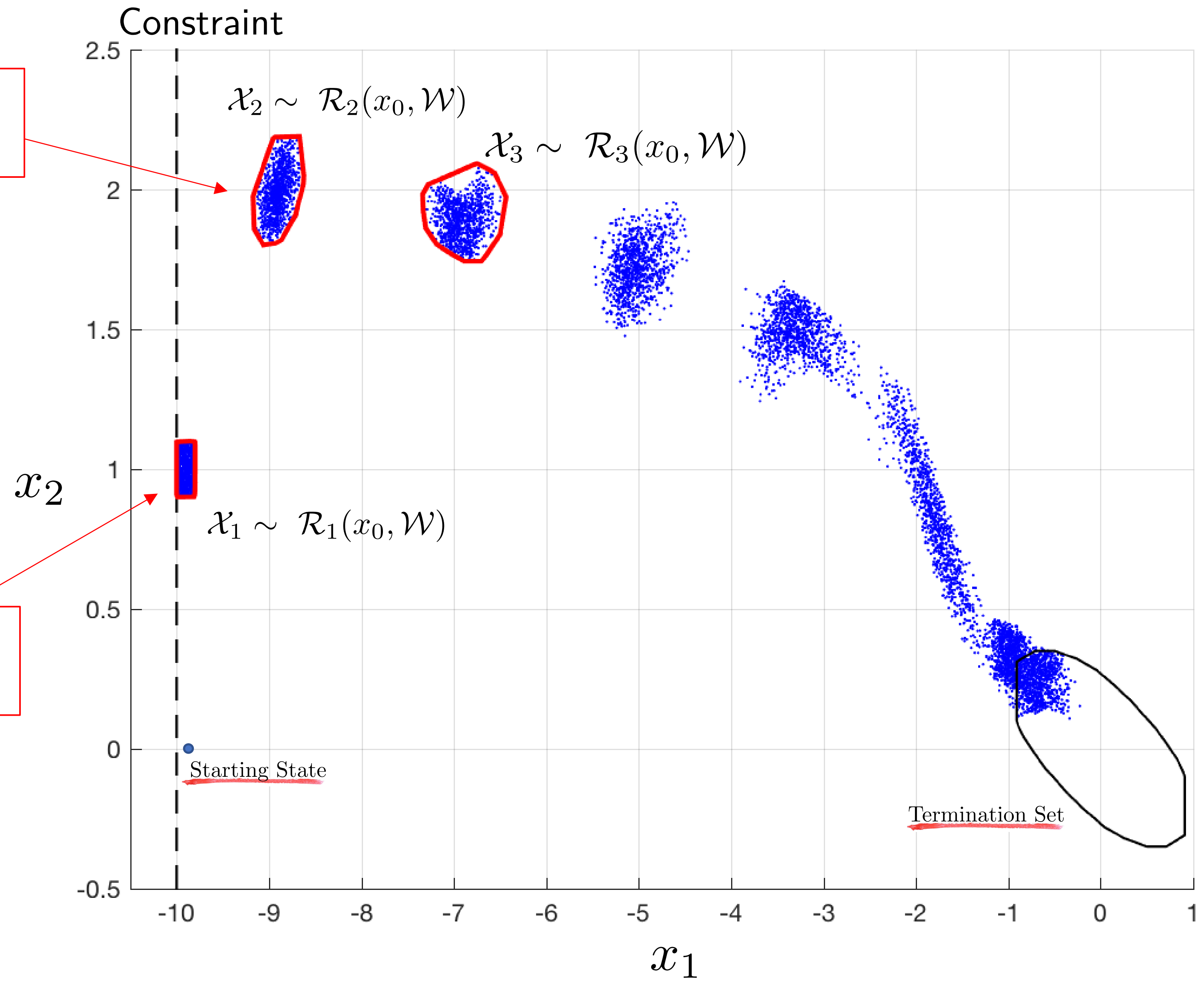
Convex hull realized state at time *t=1*

Example: Constrained LQR Uncertain System



Example: Constrained LQR Uncertain System

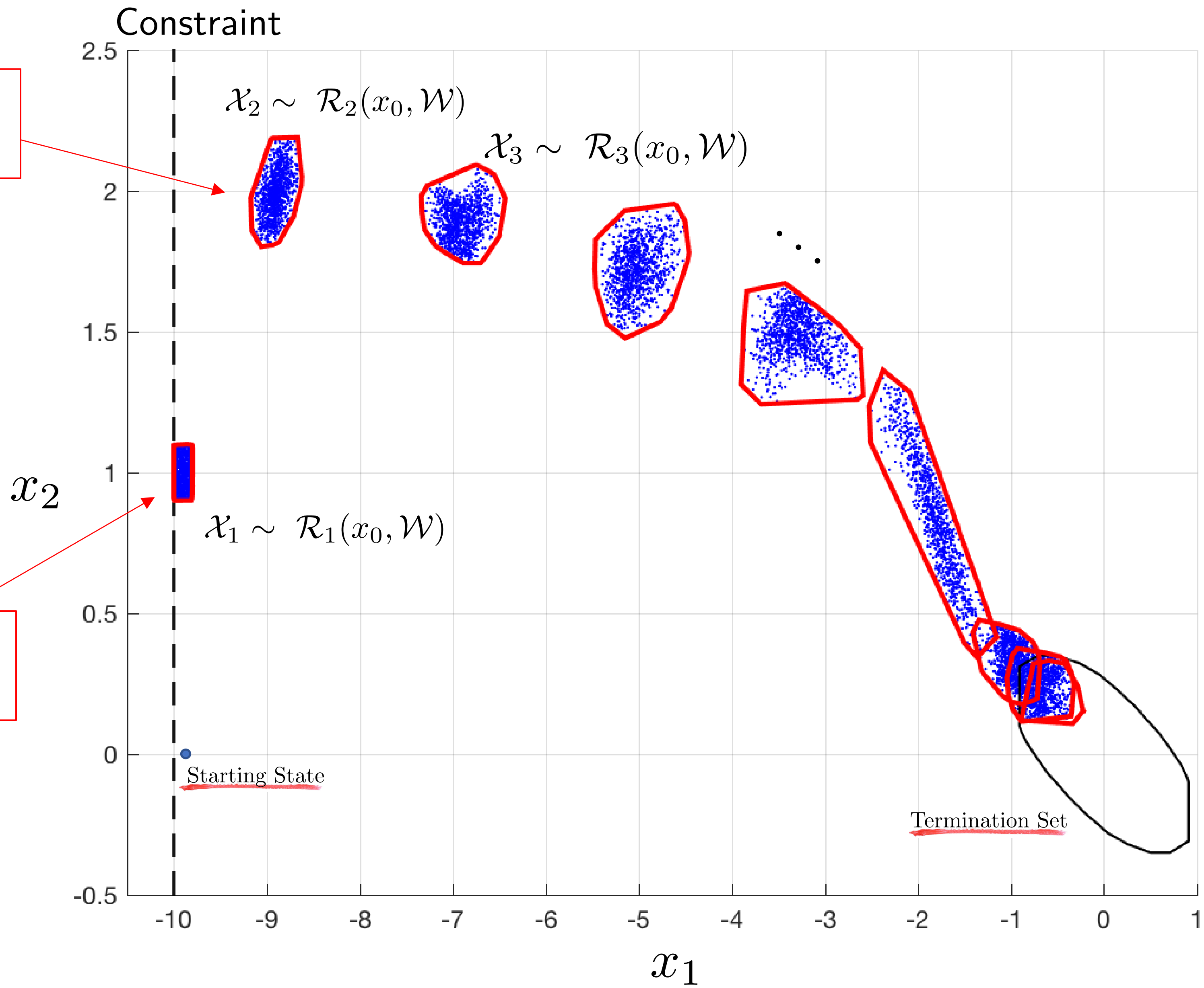
Convex hull realized state at time *t=2*



Convex hull realized state at time *t=1*

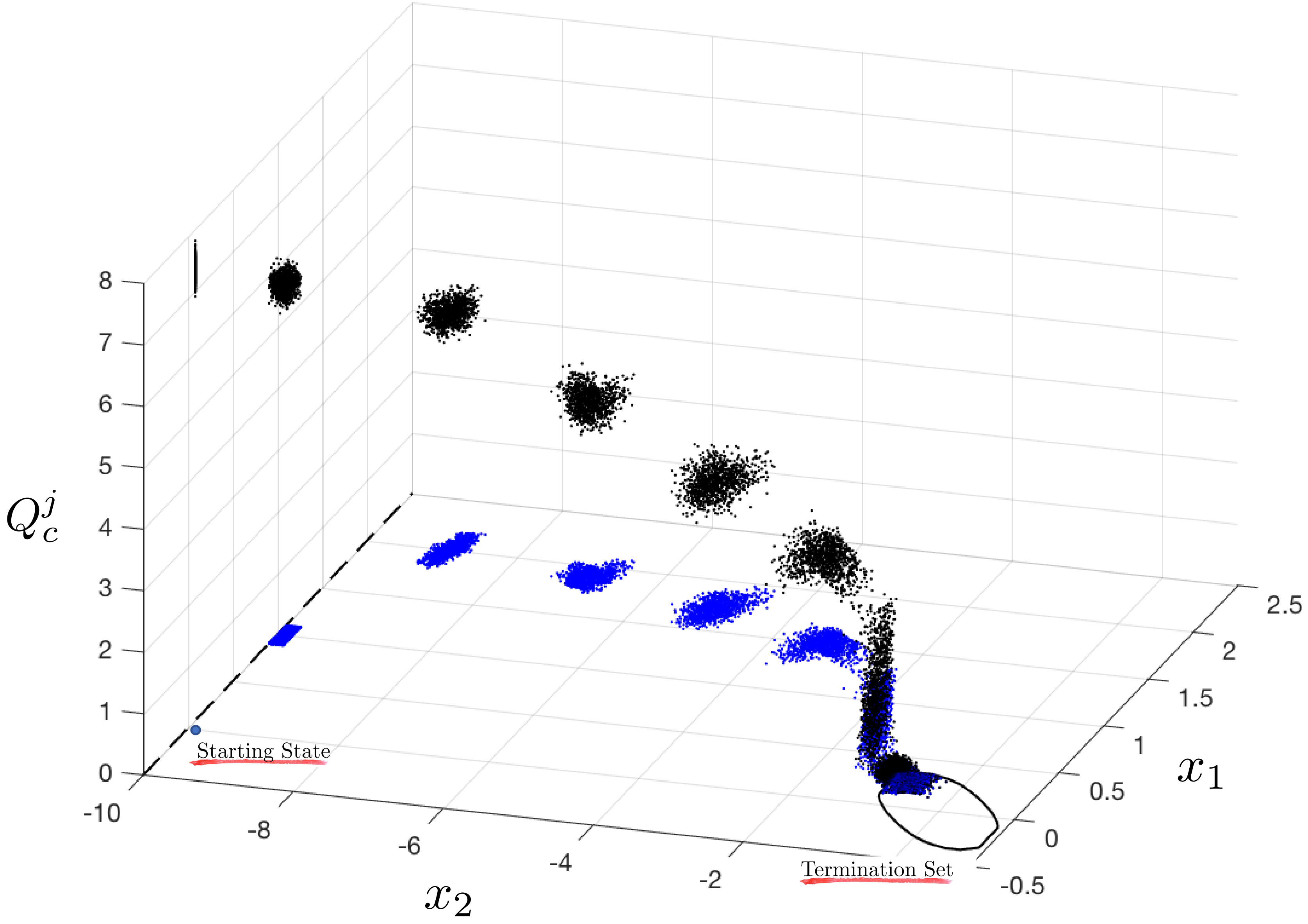
Example: Constrained LQR Uncertain System

Convex hull realized state at time *t=2*

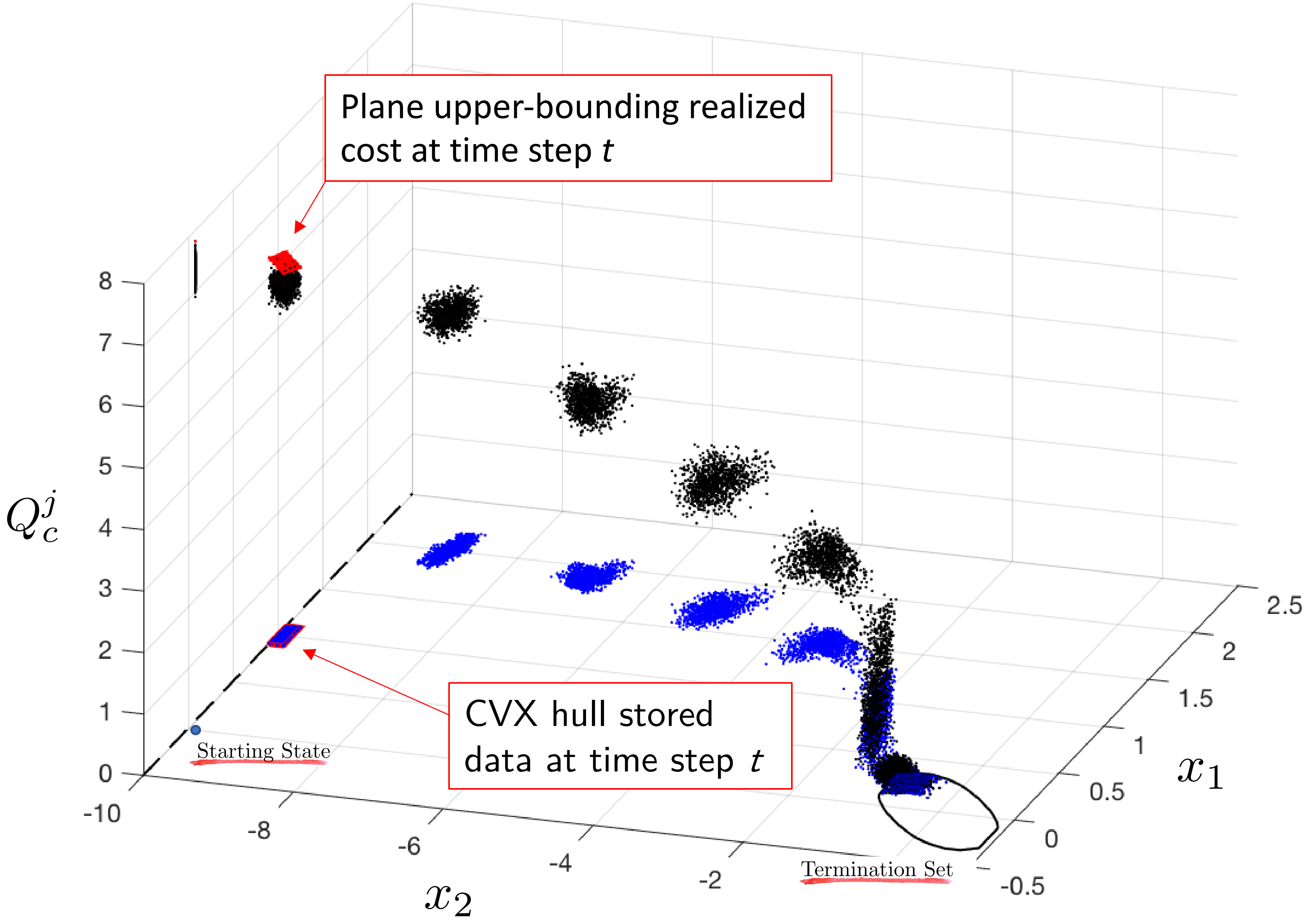


Convex hull realized state at time *t=1*

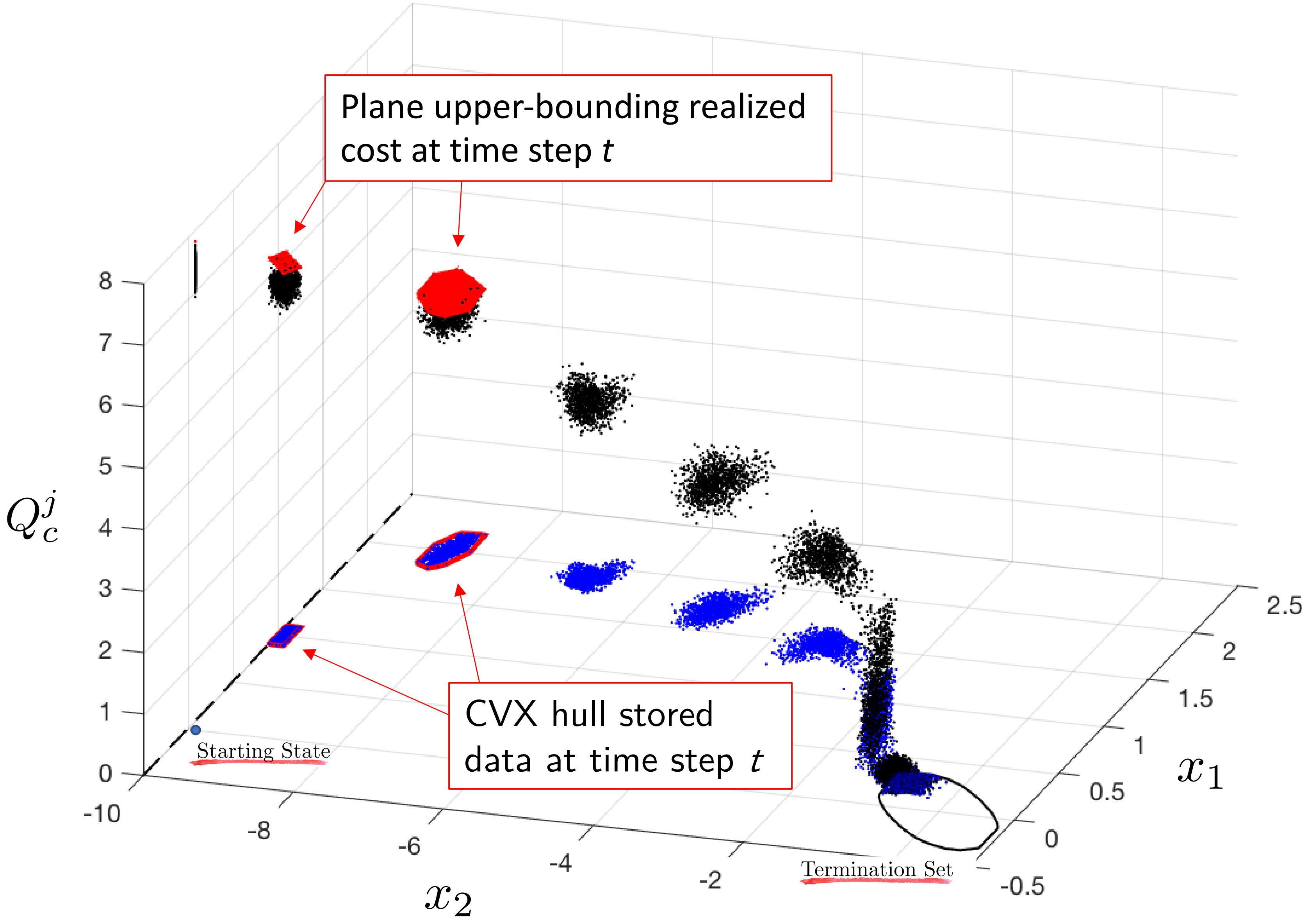
Example: Constrained LQR Uncertain System



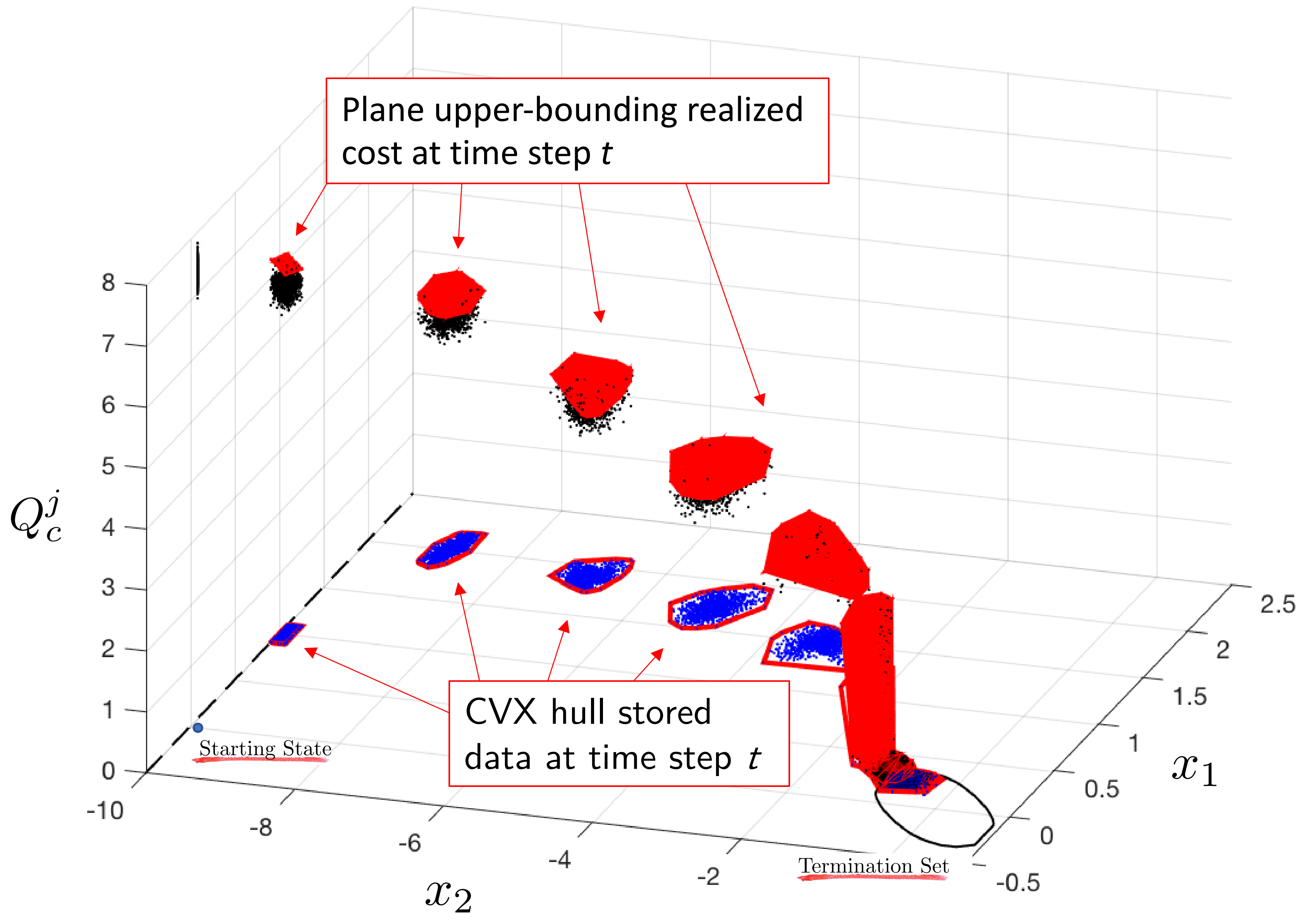
Example: Constrained LQR Uncertain System



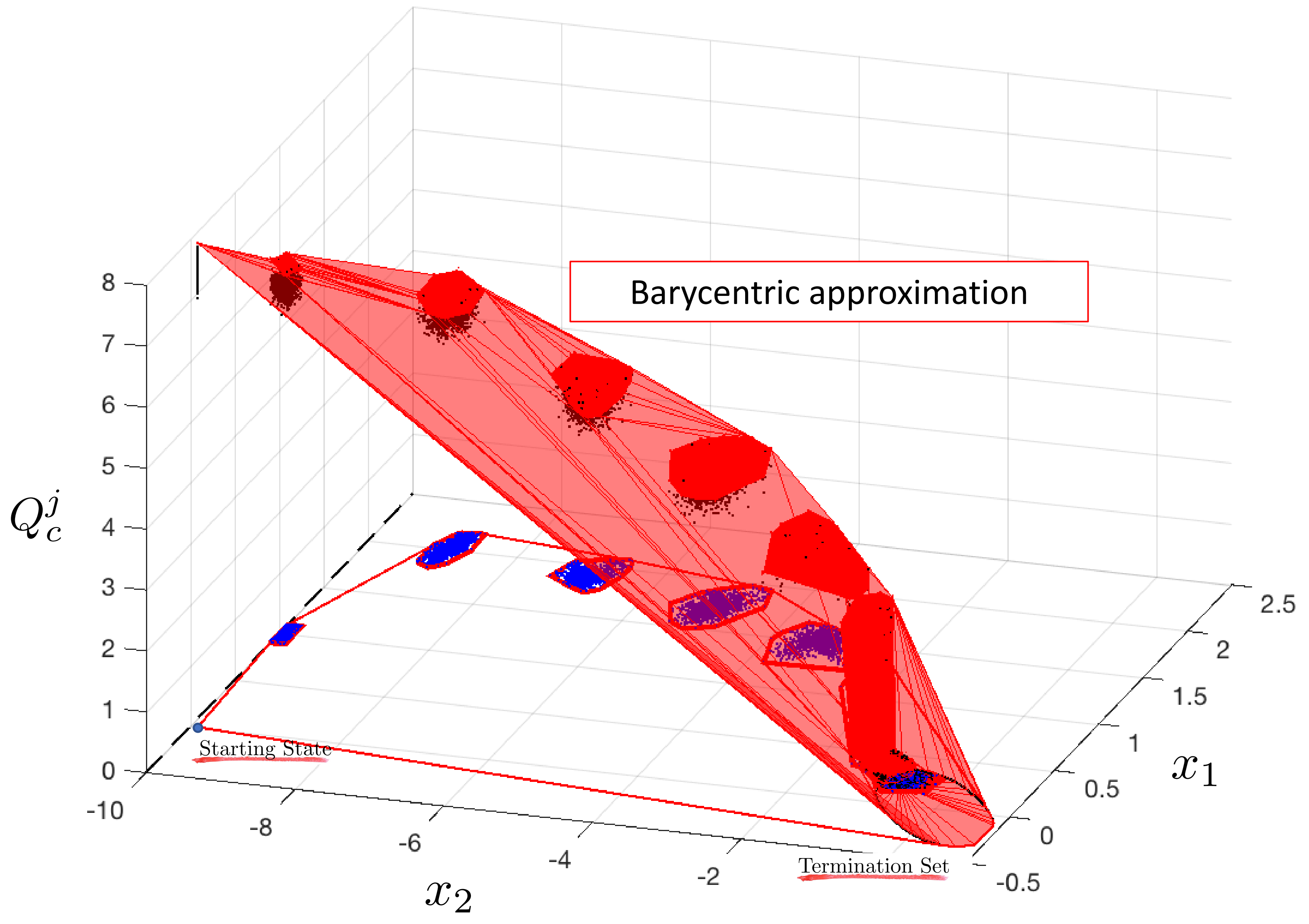
Example: Constrained LQR Uncertain System



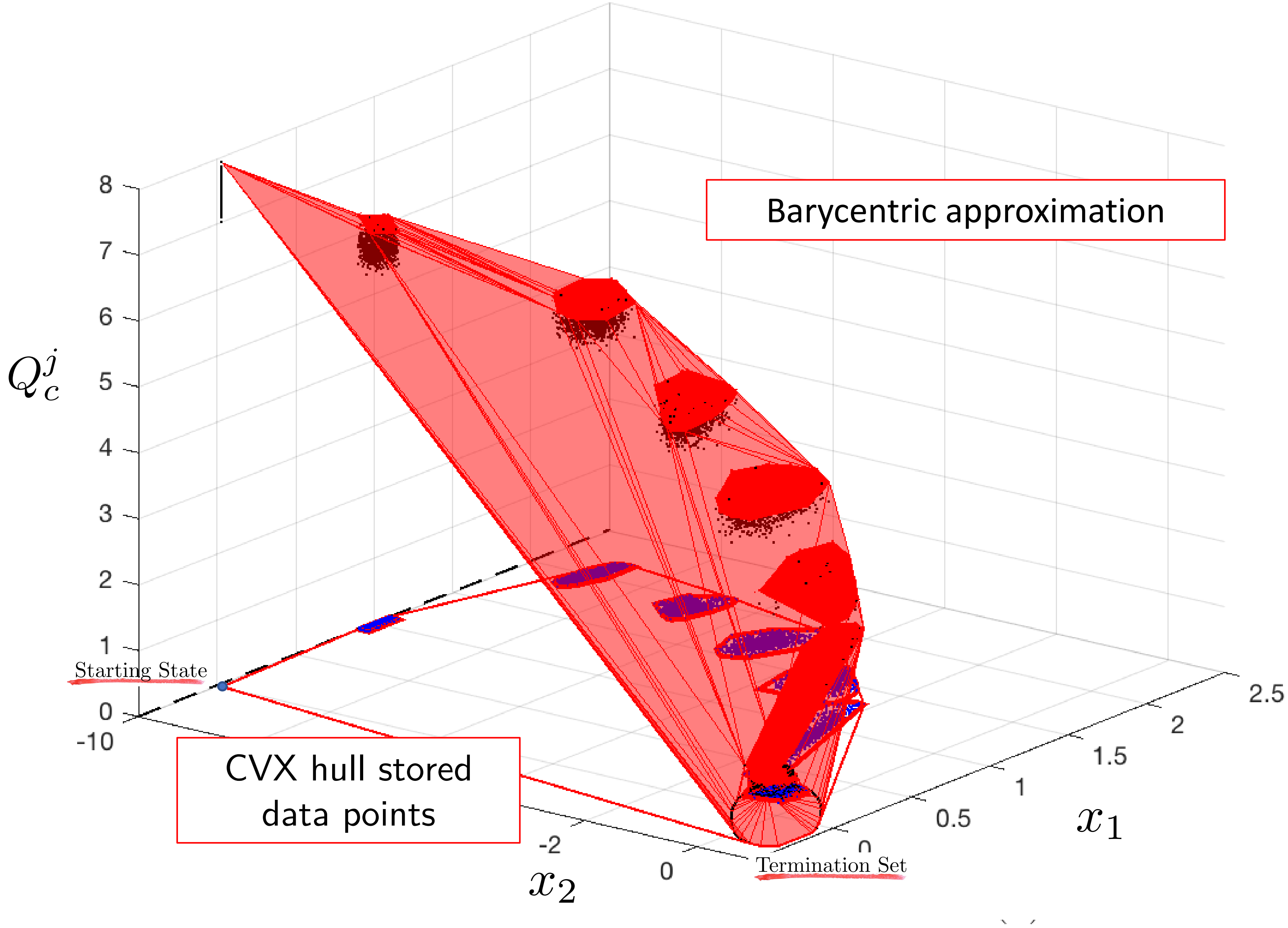
Example: Constrained LQR Uncertain System



Example: Constrained LQR Uncertain System



Example: Constrained LQR Uncertain System



Example: Constrained LQR Uncertain System