# Learning Model Predictive Control for Iterative Tasks

## Theory and Application

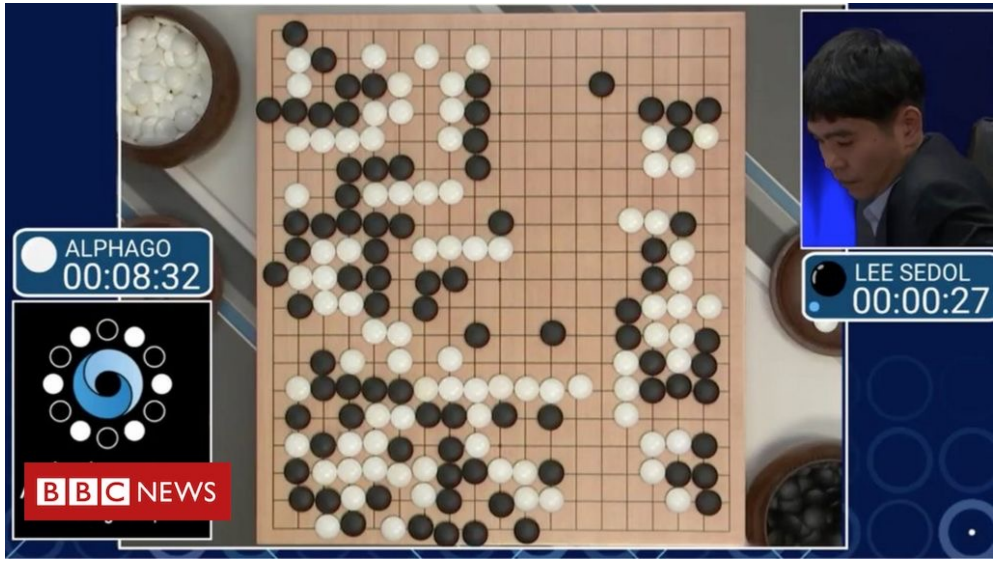
Ugo Rosolia

AMBER Lab
California Institute of Technology

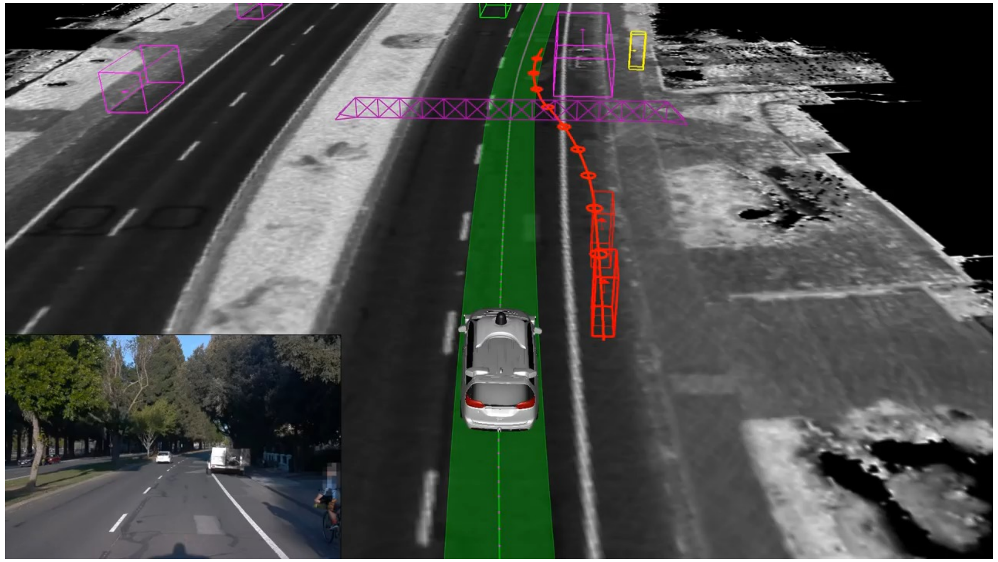
June 10th, 2021

# Success Stories from AI

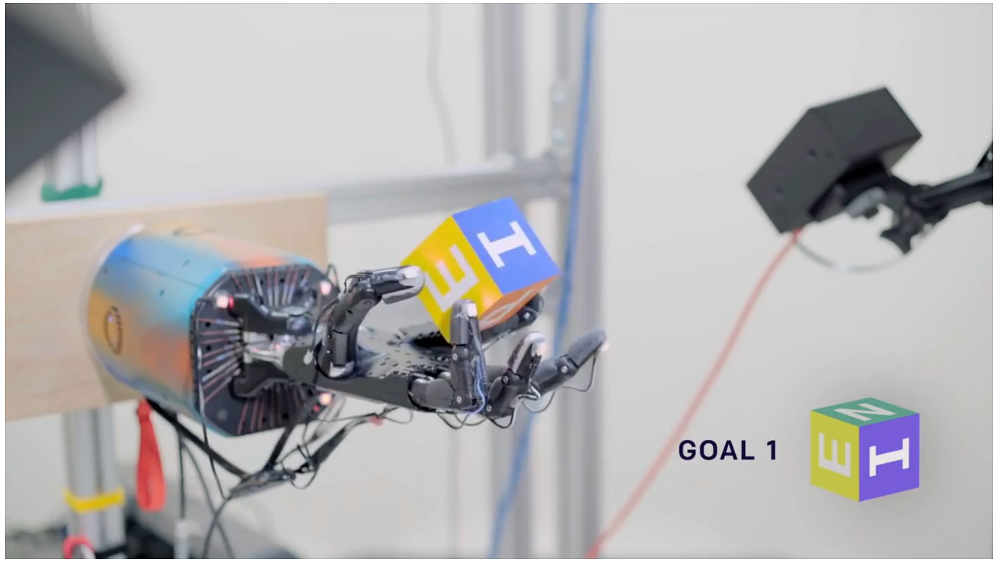
Alpha GO




WayMo's Perception Module



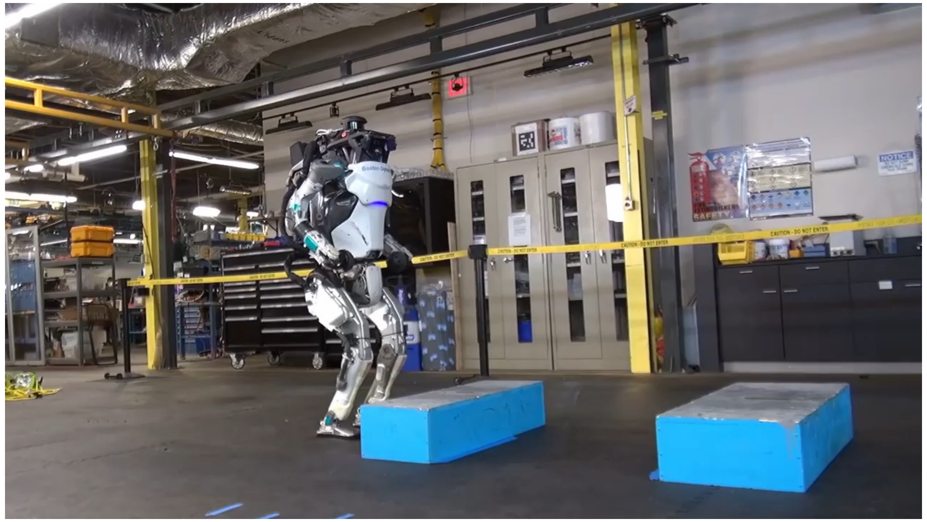
OpenAI




Google

# Success Stories from Control Theory

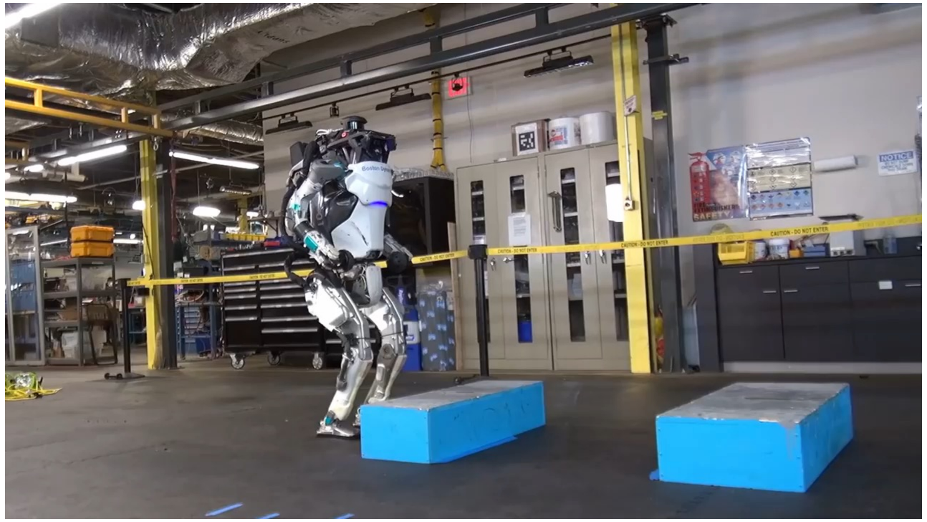
Boston Dynamics



Stanford Dynamic Design Lab

# Success Stories from Control Theory

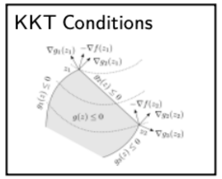
**Boston Dynamics**



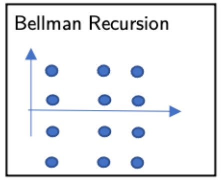
**Stanford Dynamic Design Lab**

## Standard Control Pipeline

### Optimal Trajectory



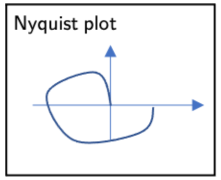

Optimization



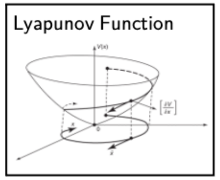

Dynamic Programming

+

### Trajectory Tracking



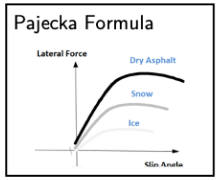

Frequency Domain



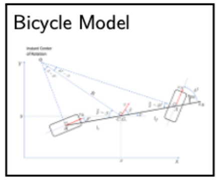

Nonlinear Control

+

### System Identification

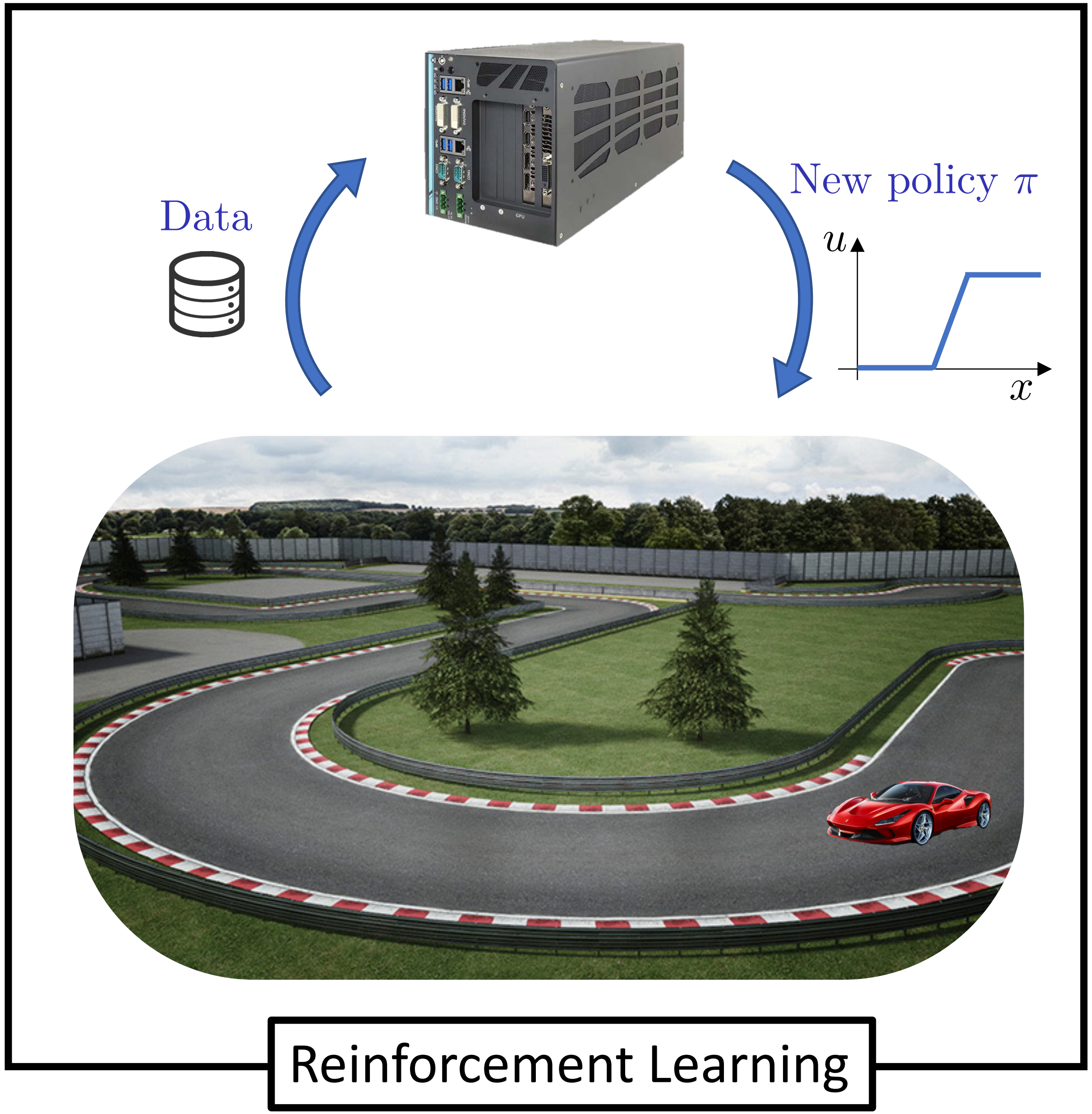


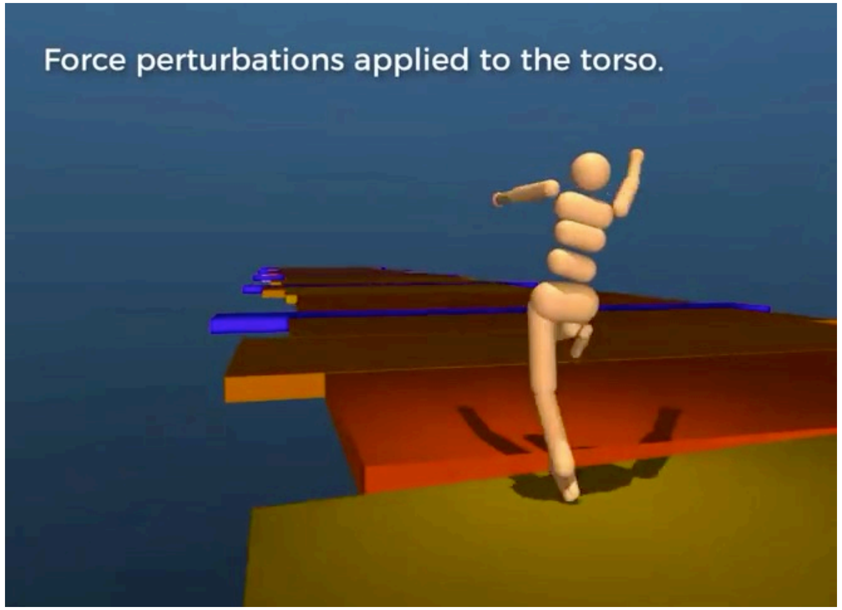

Tire Dynamics



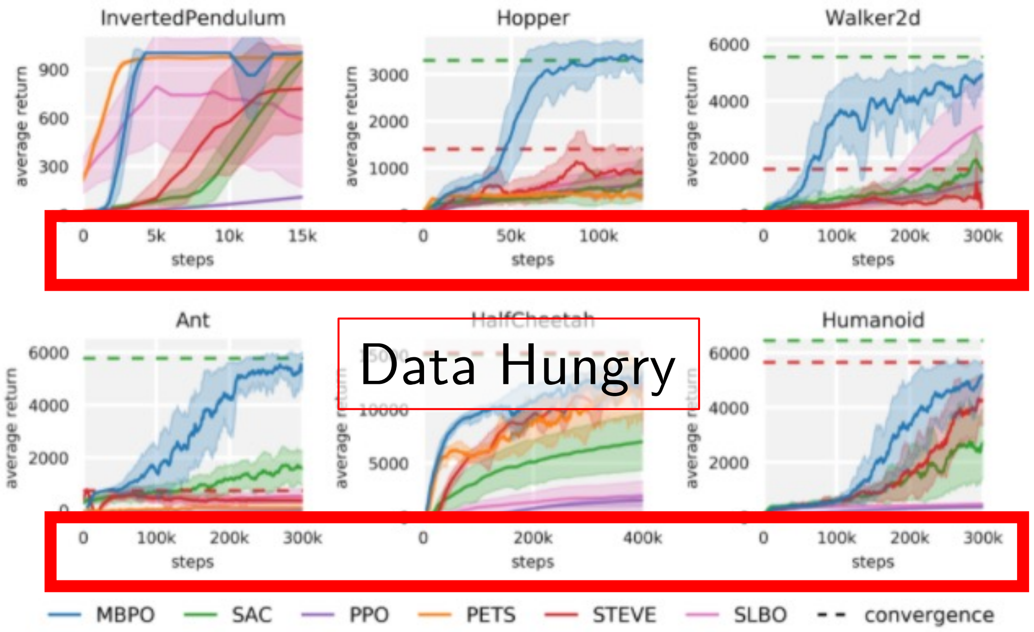

Vehicle Dynamics

# Can we simplify the control design?



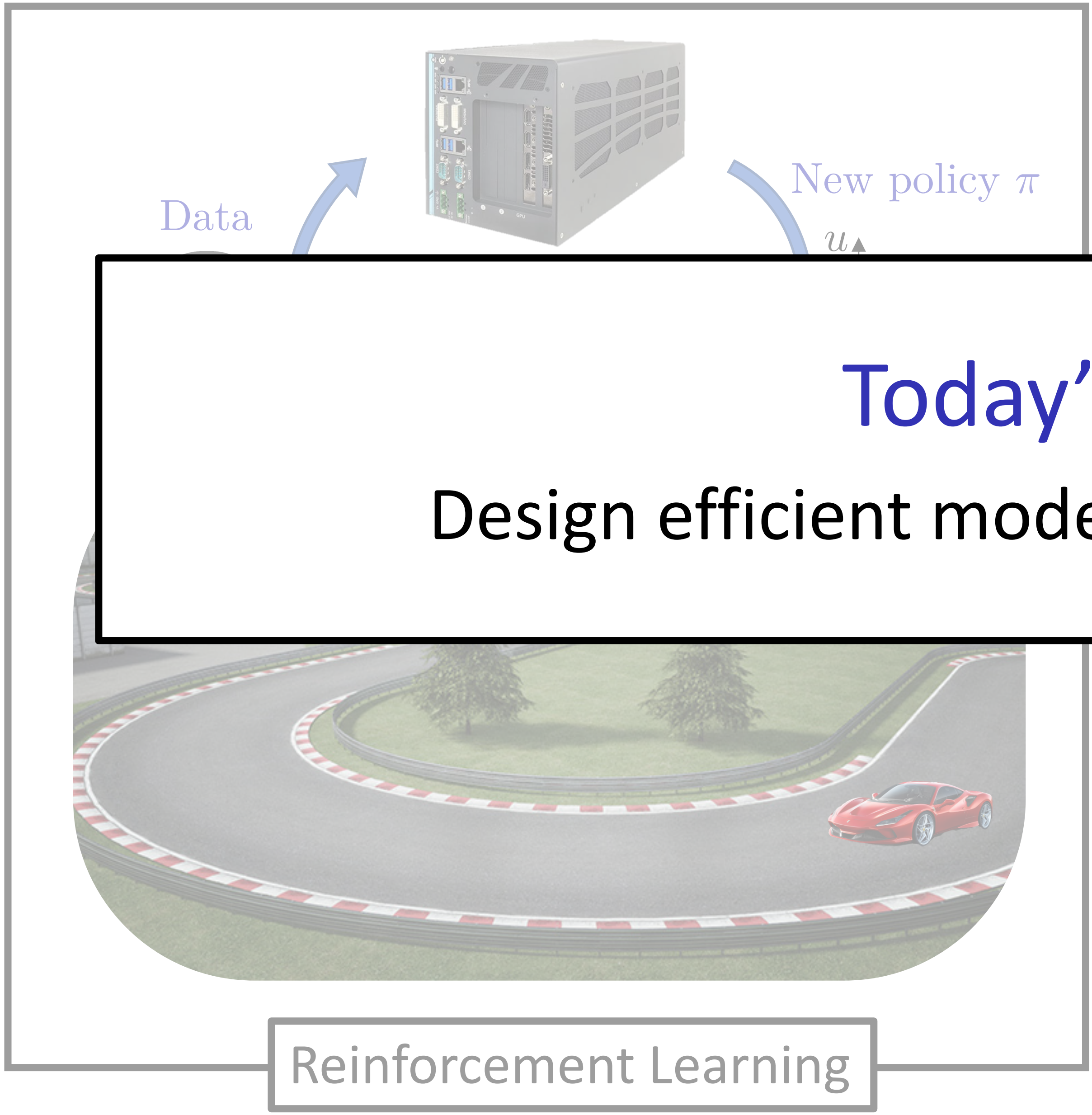



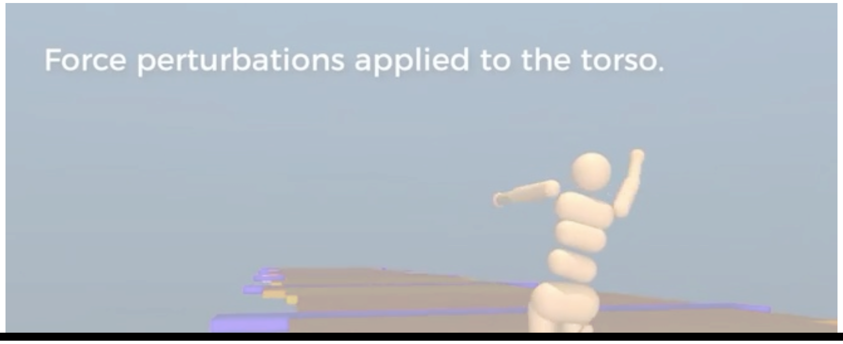

DeepMind




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

# Can we simplify the control design?



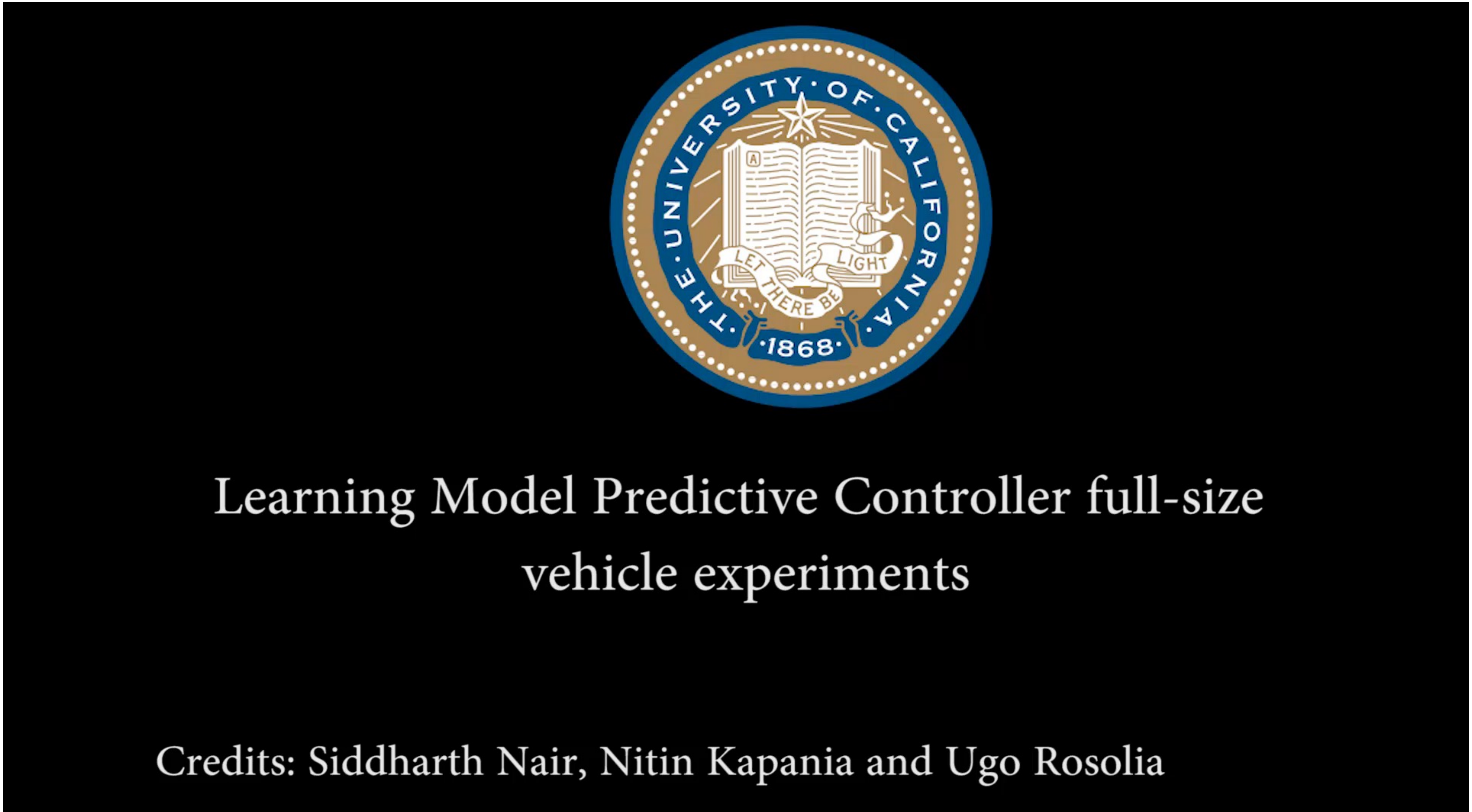

DeepMind

## Today's goals:

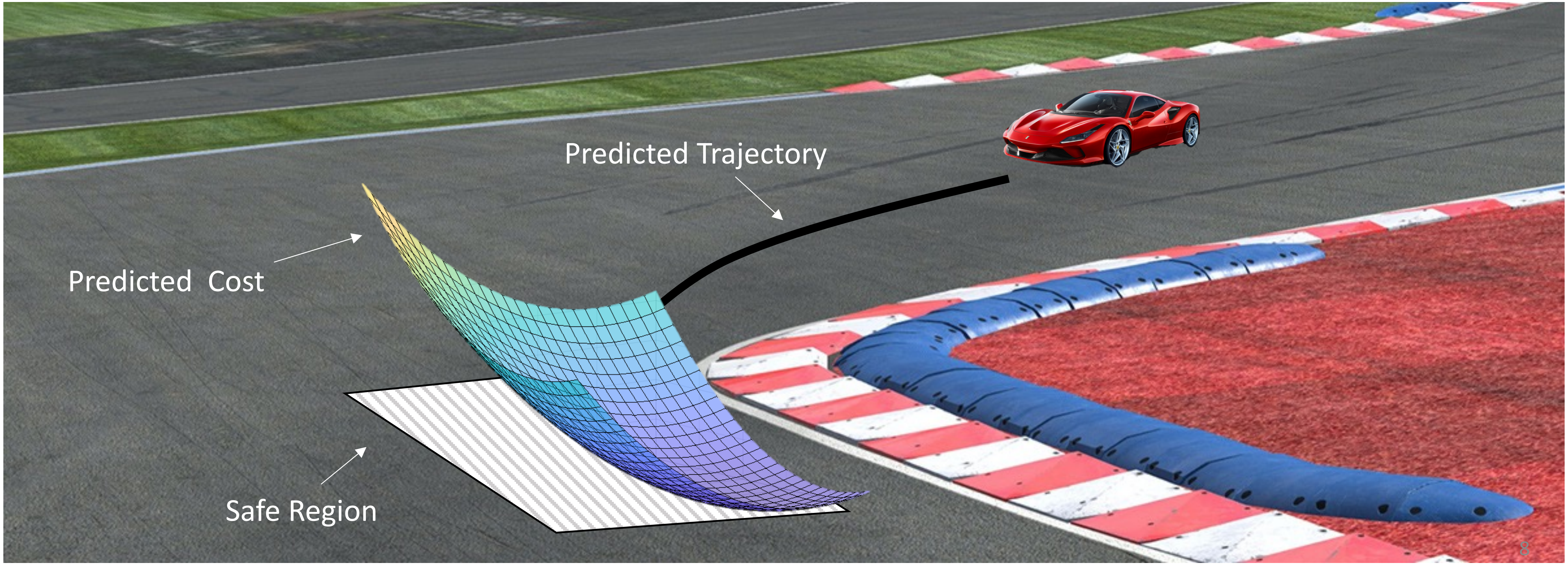
Design efficient model-based RL framework


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

# Today's Example

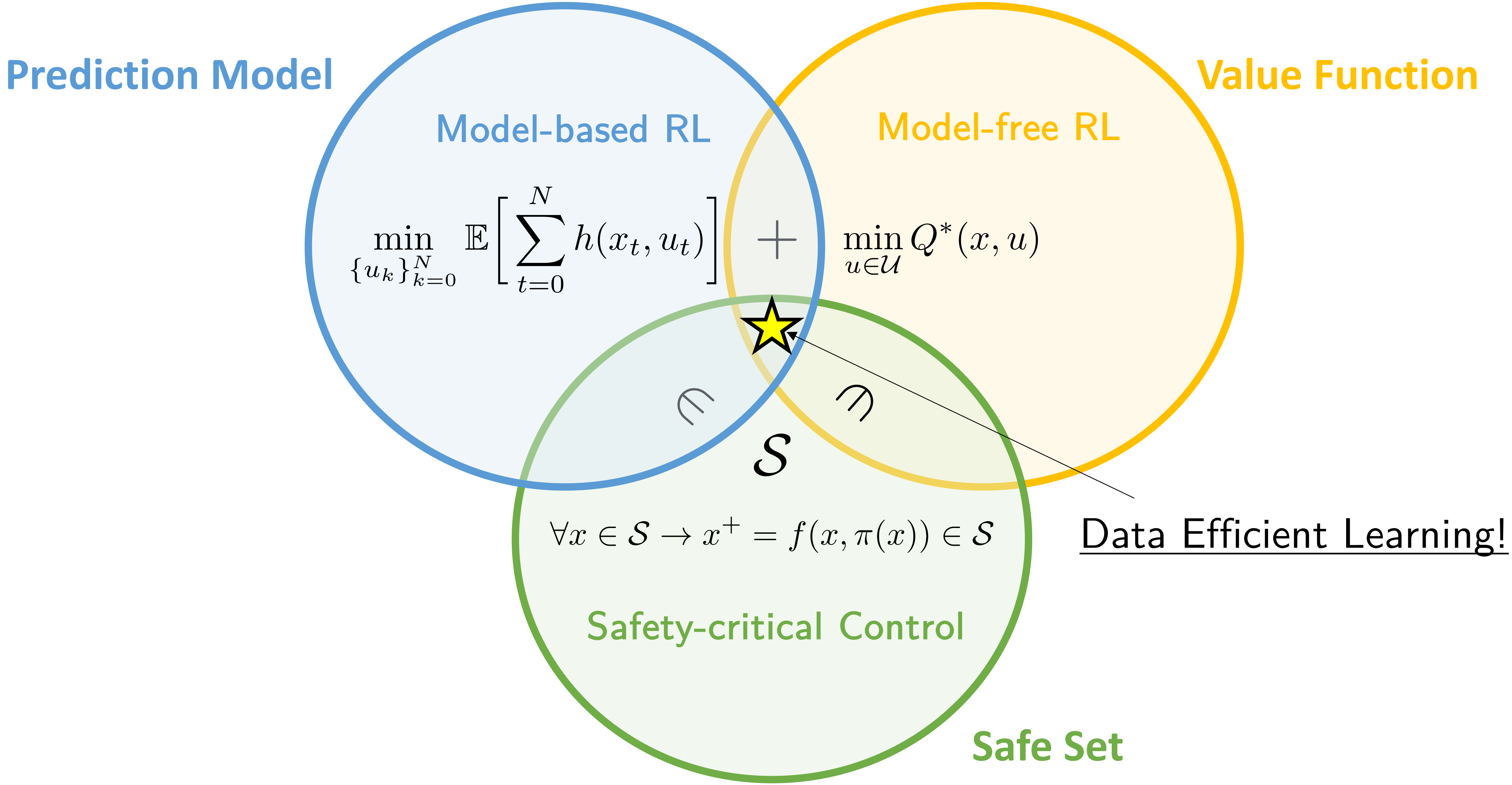


# Lesson from Predictive Control




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

# Three key components to learn



# Iterative Control Synthesis

Controller

# Iterative Control Synthesis

Controller

# Iterative Control Synthesis

Controller

Task Execution

# Iterative Control Synthesis

Controller

Task Execution

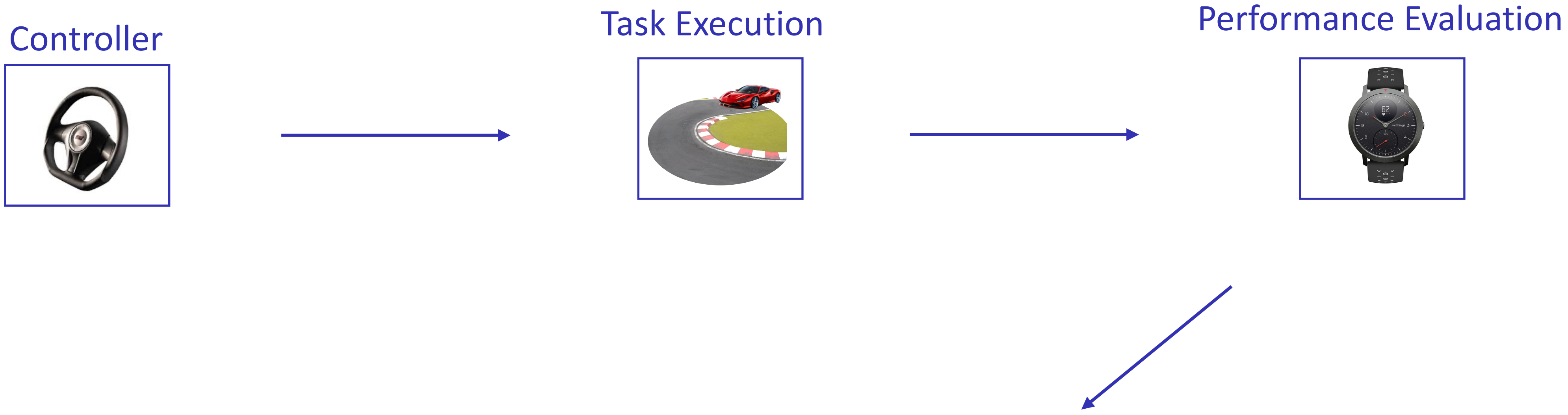
# Iterative Control Synthesis

Controller

Task Execution

Performance Evaluation

# Iterative Control Synthesis



# Iterative Control Synthesis

Controller

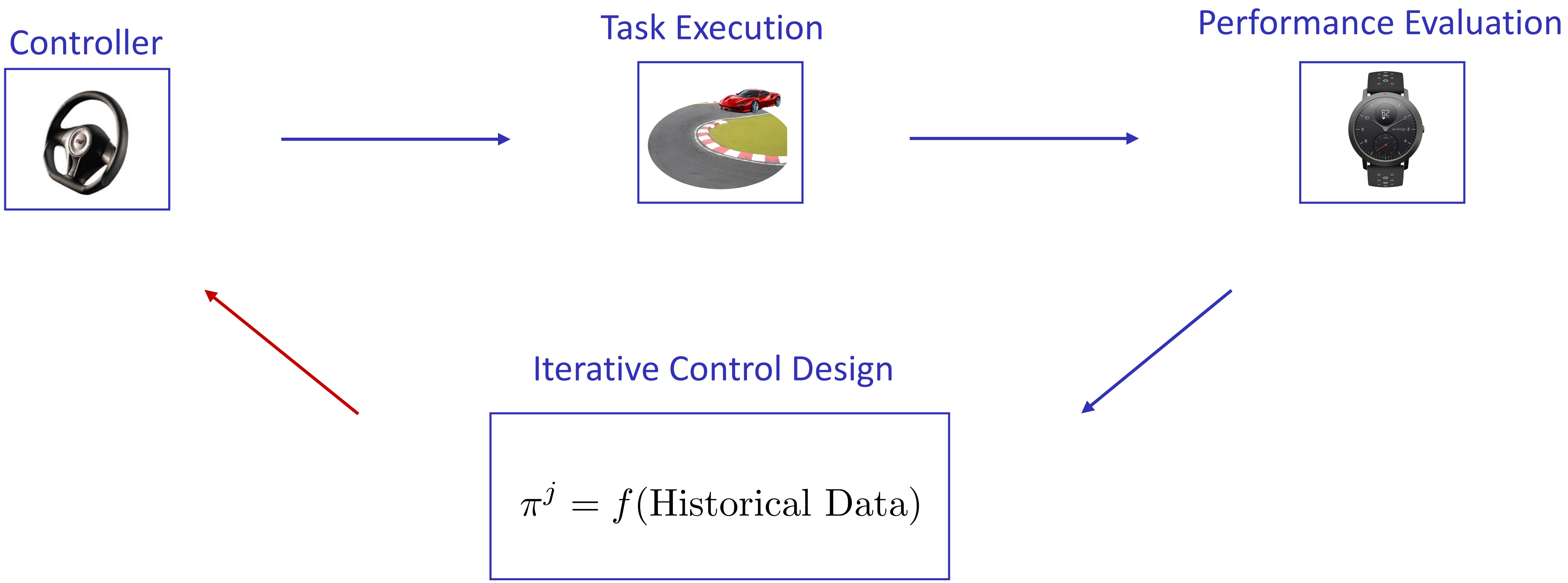
Task Execution

Performance Evaluation

Iterative Control Design

$$\pi^j = f(\text{Historical Data})$$

# Iterative Control Synthesis



# Iterative Control Synthesis

Controller

Task Execution

Performance Evaluation

Safety ?

Iterative Control Design

$$\pi^j = f(\text{Historical Data})$$

# Iterative Control Synthesis

Controller

Task Execution

Performance Evaluation

Safety ?

Performance Improvement?

Iterative Control Design

$$\pi^j = f(\text{Historical Data})$$

# Iterative Control Synthesis

Controller

Task Execution

Performance Evaluation

Iterative Control Design

$$\pi^j = f(\text{Historical Data})$$

Safety ?

Performance Improvement?

(Local) Optimality?

# Outline

- Iterative Control Design for Deterministic Systems
- Autonomous Racing Experiments

# Outline

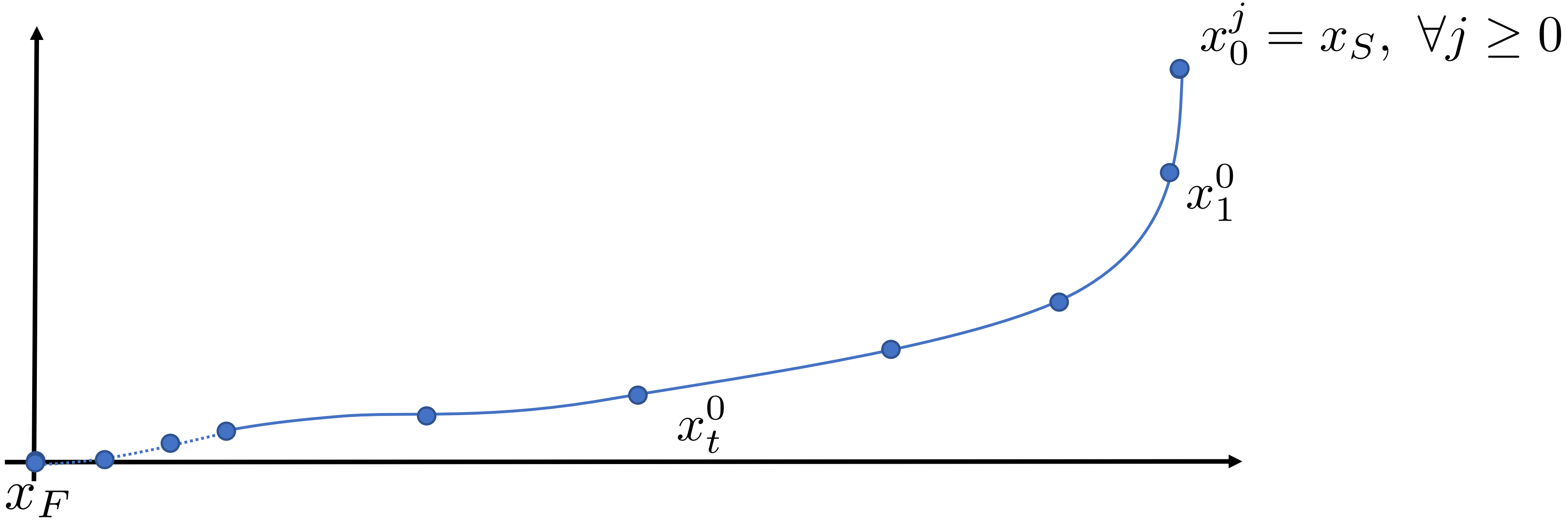
- Iterative Control Design for Deterministic Systems
- Autonomous Racing Experiments

# Episodic Settings

Iterative data collection and policy update

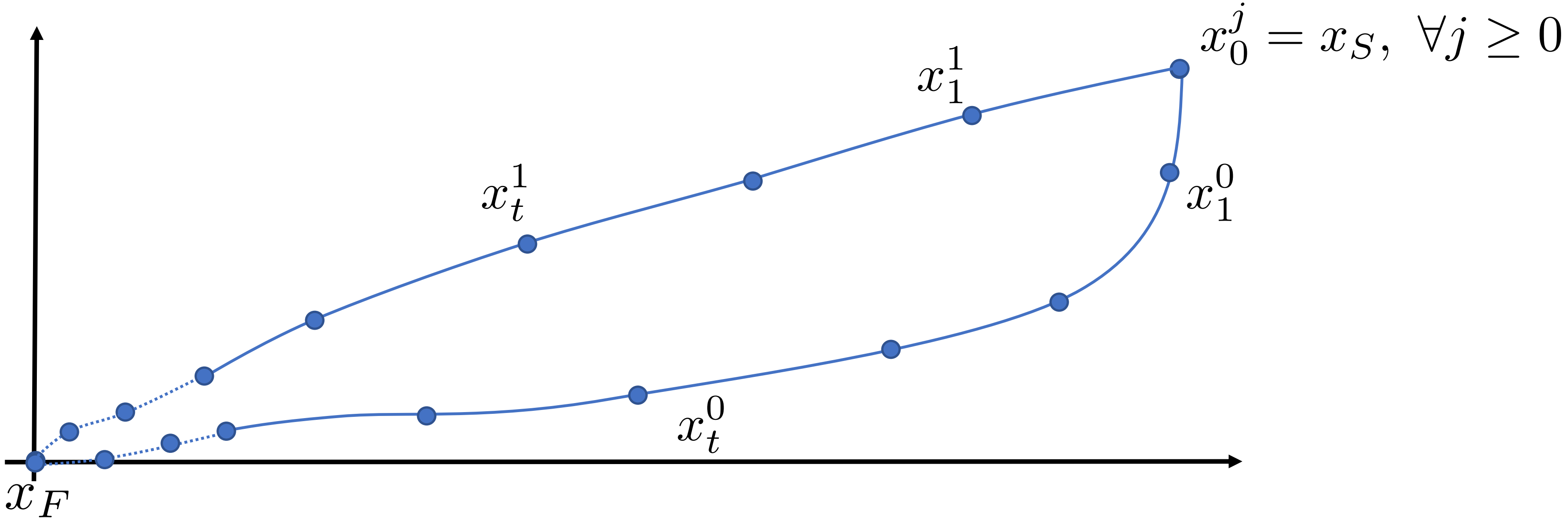
# Iterative Tasks – Problem Setup

- One task execution referred to as "iteration" or "episode"
- Same initial and terminal state at each iteration



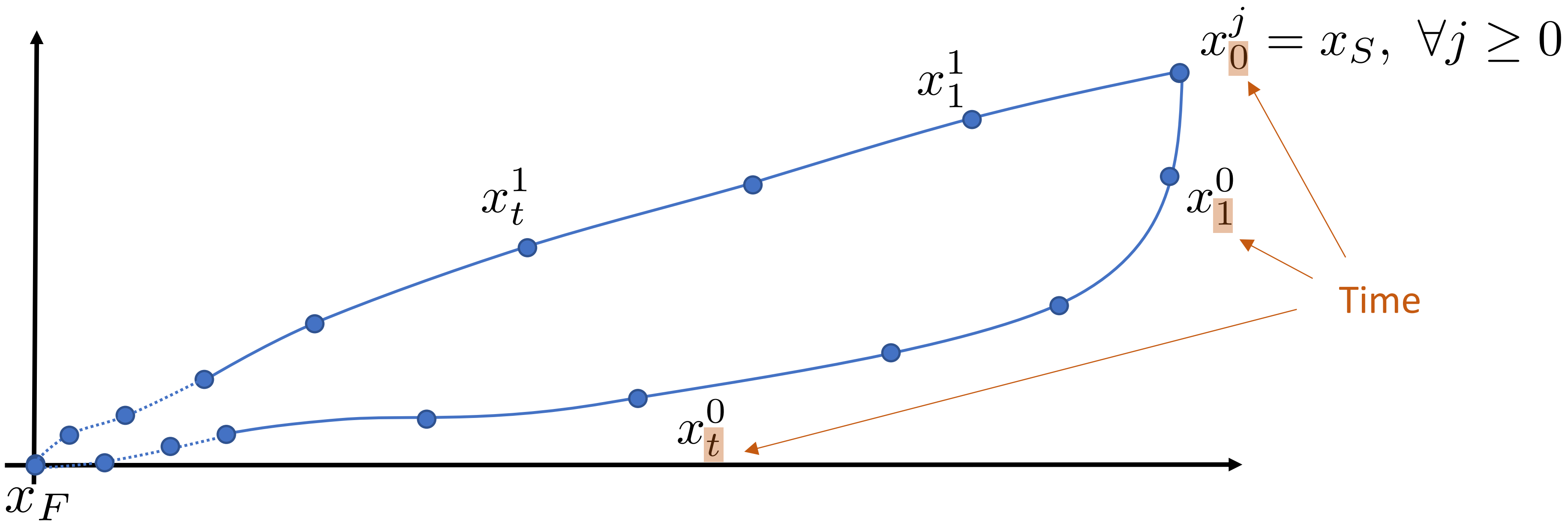
# Iterative Tasks – Problem Setup

- One task execution referred to as "iteration" or "episode"
- Same initial and terminal state at each iteration



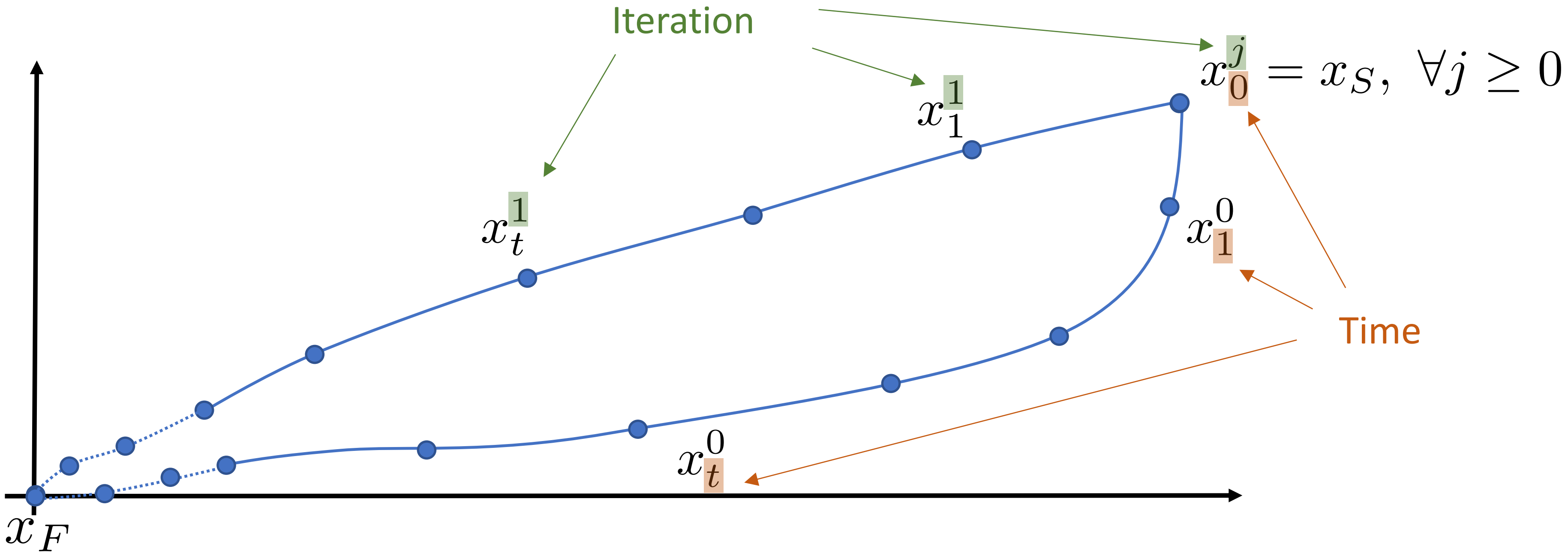
# Iterative Tasks – Problem Setup

- One task execution referred to as "iteration" or "episode"
- Same initial and terminal state at each iteration



# Iterative Tasks – Problem Setup

- One task execution referred to as "iteration" or "episode"
- Same initial and terminal state at each iteration



# Learning Model Predictive Control

Exploit historical data

# Learning Model Predictive Control (LMPC) – Key Idea

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

# Learning Model Predictive Control (LMPC) – Key Idea

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

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

Value Function

# Learning Model Predictive Control (LMPC) – Key Idea

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

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

s.t.

$$x^j_{k+1|t} = f(x^j_{k|t}, u^j_{k|t}), \ \forall k \in [t, \cdots, t+N-1]$$

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

Value Function

Prediction Model

# Learning Model Predictive Control (LMPC) – Key Idea

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

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

s.t.

$$x^j_{k+1|t} = f(x^j_{k|t}, u^j_{k|t}), \ \forall k \in [t, \cdots, t+N-1]$$

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

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

Value Function

Prediction Model

# Learning Model Predictive Control (LMPC) – Key Idea

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

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

s.t.

$$x^j_{k+1|t} = f(x^j_{k|t}, u^j_{k|t}),\ \forall k \in [t, \cdots, t+N-1]$$

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

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

$$x^j_{t+N|t} \in \mathcal{SS}^{j-1},$$

Value Function

Prediction Model

Safe Set

# Learning Model Predictive Control (LMPC) – Key Idea

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

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

s.t.

$$x^j_{k+1|t} = f(x^j_{k|t}, u^j_{k|t}),\ \forall k \in [t, \cdots, t+N-1]$$

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

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

$$x^j_{t+N|t} \in \mathcal{SS}^{j-1},$$

Value Function

Prediction Model

Safe Set

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

# Learning Model Predictive Control (LMPC) – Key Idea

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

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

s.t.

$$x^j_{k+1|t} = f(x^j_{k|t}, u^j_{k|t}),\ \forall k \in [t, \cdots, t+N-1]$$

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

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

$$x^j_{t+N|t} \in \mathcal{SS}^{j-1},$$

Value Function

Safe Set

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

# Learning Model Predictive Control (LMPC) – Key Idea

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

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

s.t.

$$x^j_{k+1|t} = f(x^j_{k|t}, u^j_{k|t}),\ \forall k \in [t, \cdots, t+N-1]$$

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

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

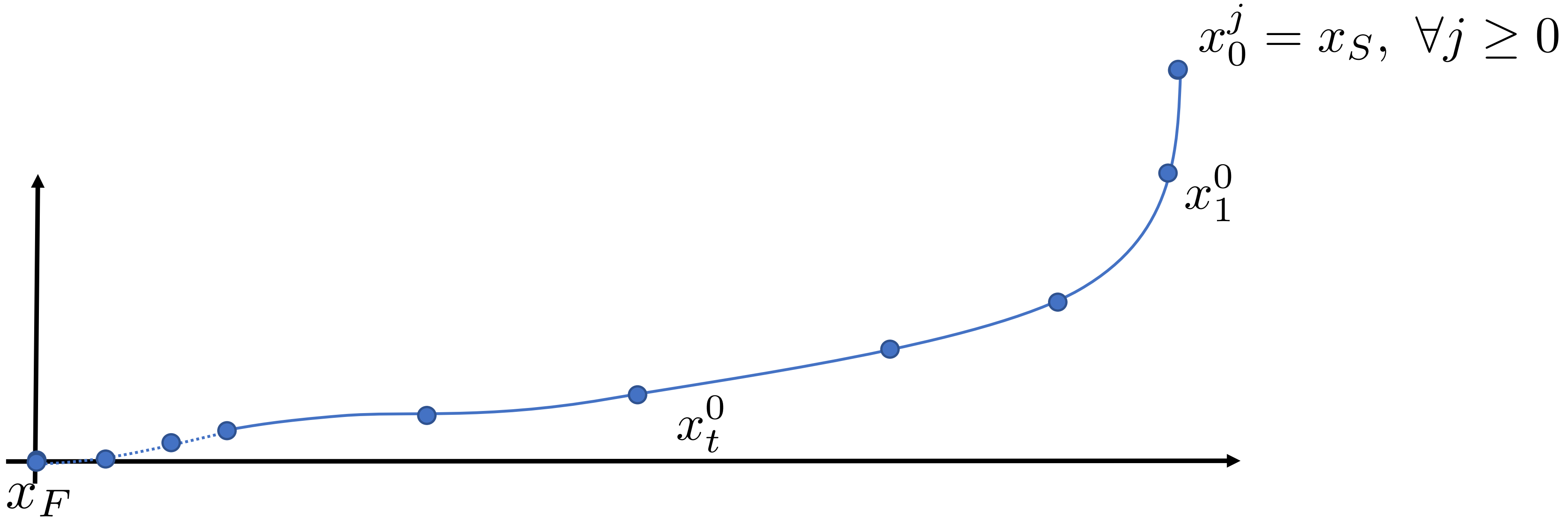
$$x^j_{t+N|t} \in \mathcal{SS}^{j-1},$$

**Safe Set**

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

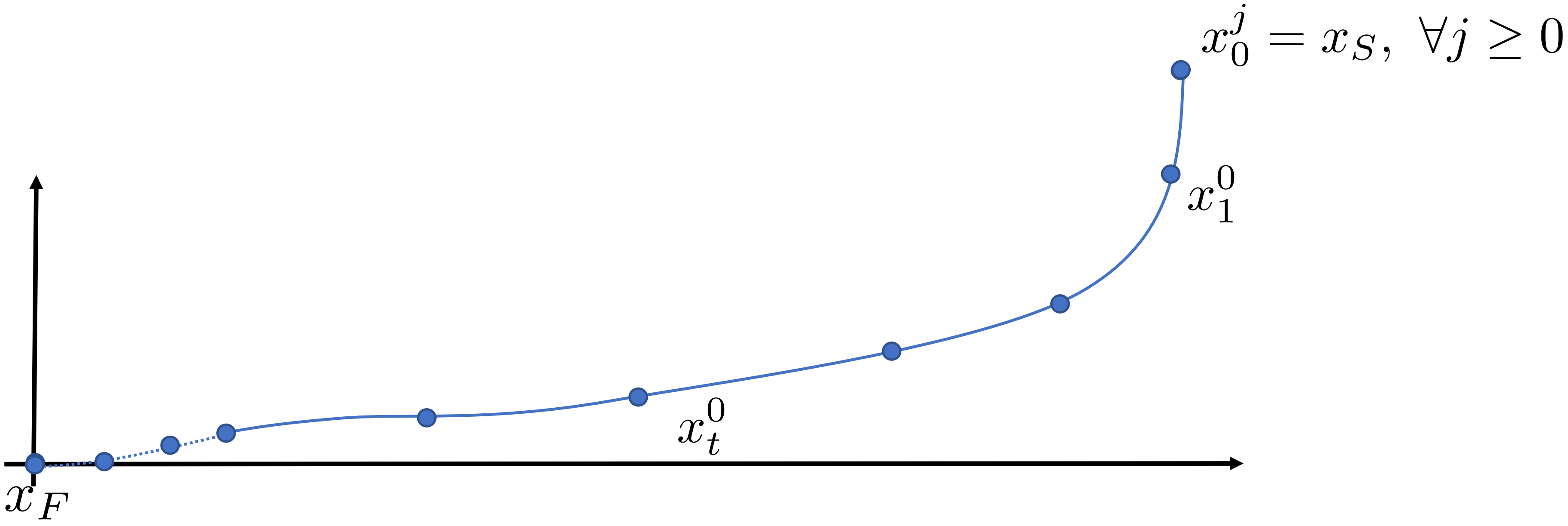
# Iteration 0

Assume that at iteration 0 a feasible trajectory is known



# Iteration 0

Assume that at iteration 0 a feasible trajectory is known

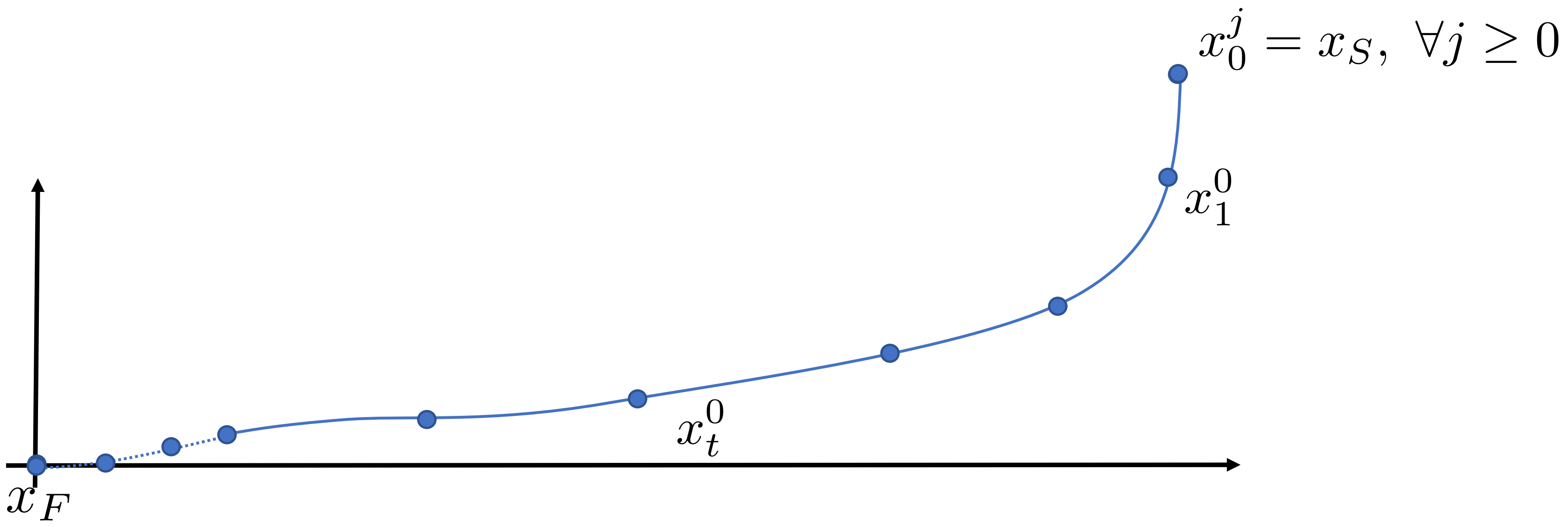



**Definition: Sampled Safe Set**

$$\mathcal{SS}^0 = \left\{ \bigcup_{t=0}^{\infty} x_t^0 \right\}$$

# Iteration 0

Assume that at iteration 0 a feasible trajectory is known




**Definition: Sampled Safe Set**

$$\mathcal{SS}^0 = \left\{ \bigcup_{t=0}^{\infty} x_t^0 \right\}$$

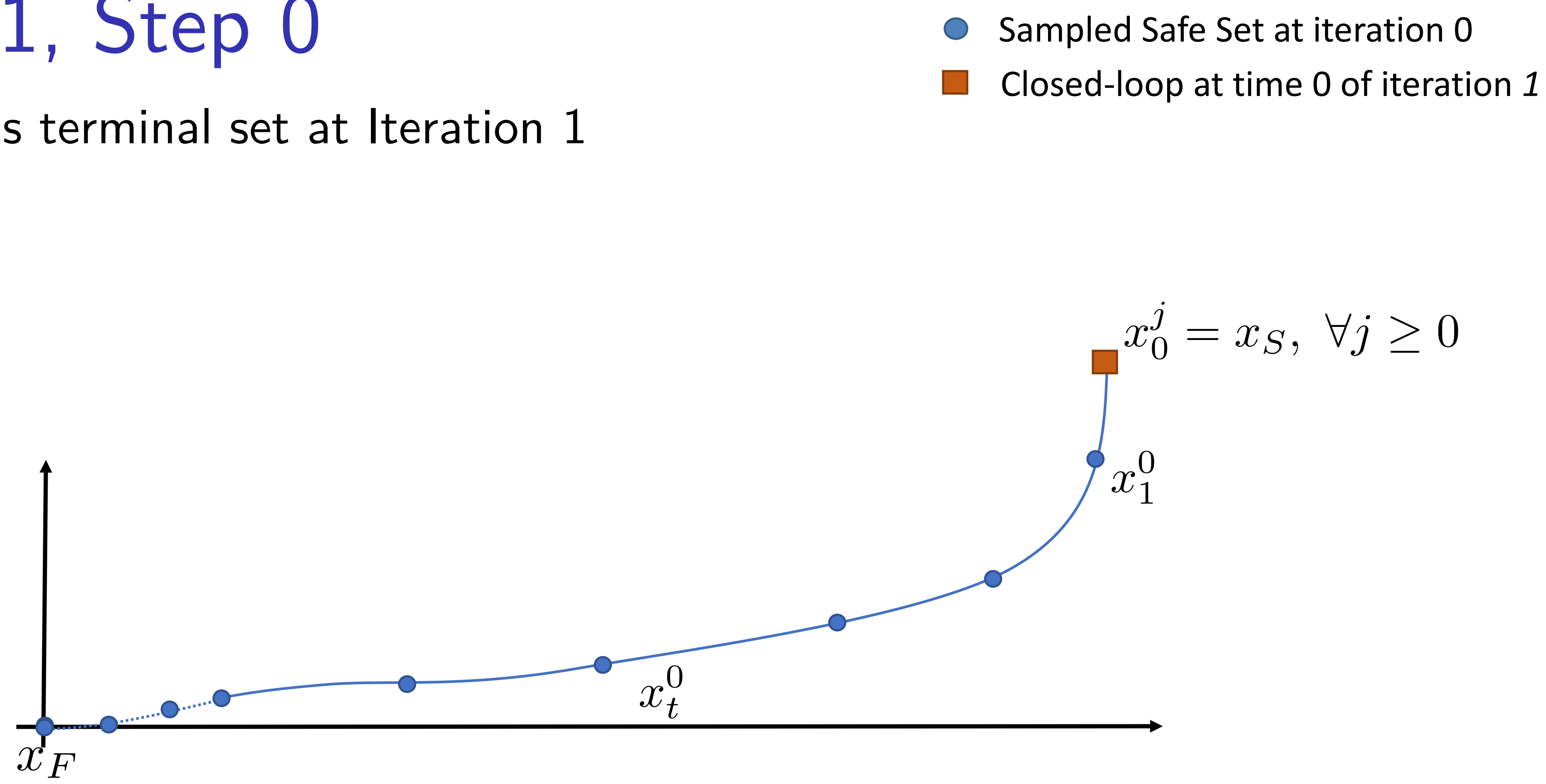
At iteration *0*

A Control Invariant Set

for Constrained Nonlinear Dynamical Systems

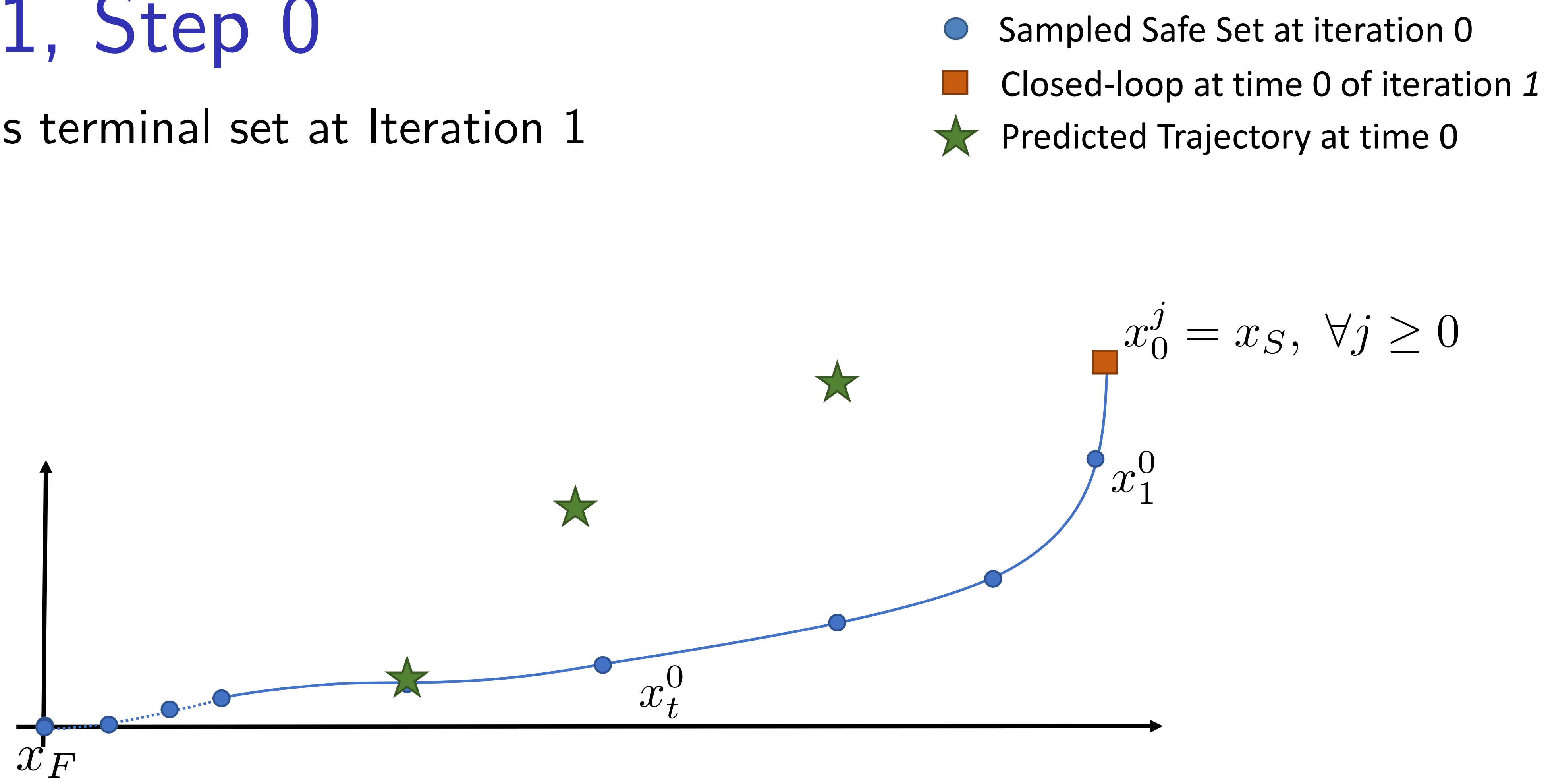
# Iteration 1, Step 0

Use $\mathcal{SS}^0$ as terminal set at Iteration 1



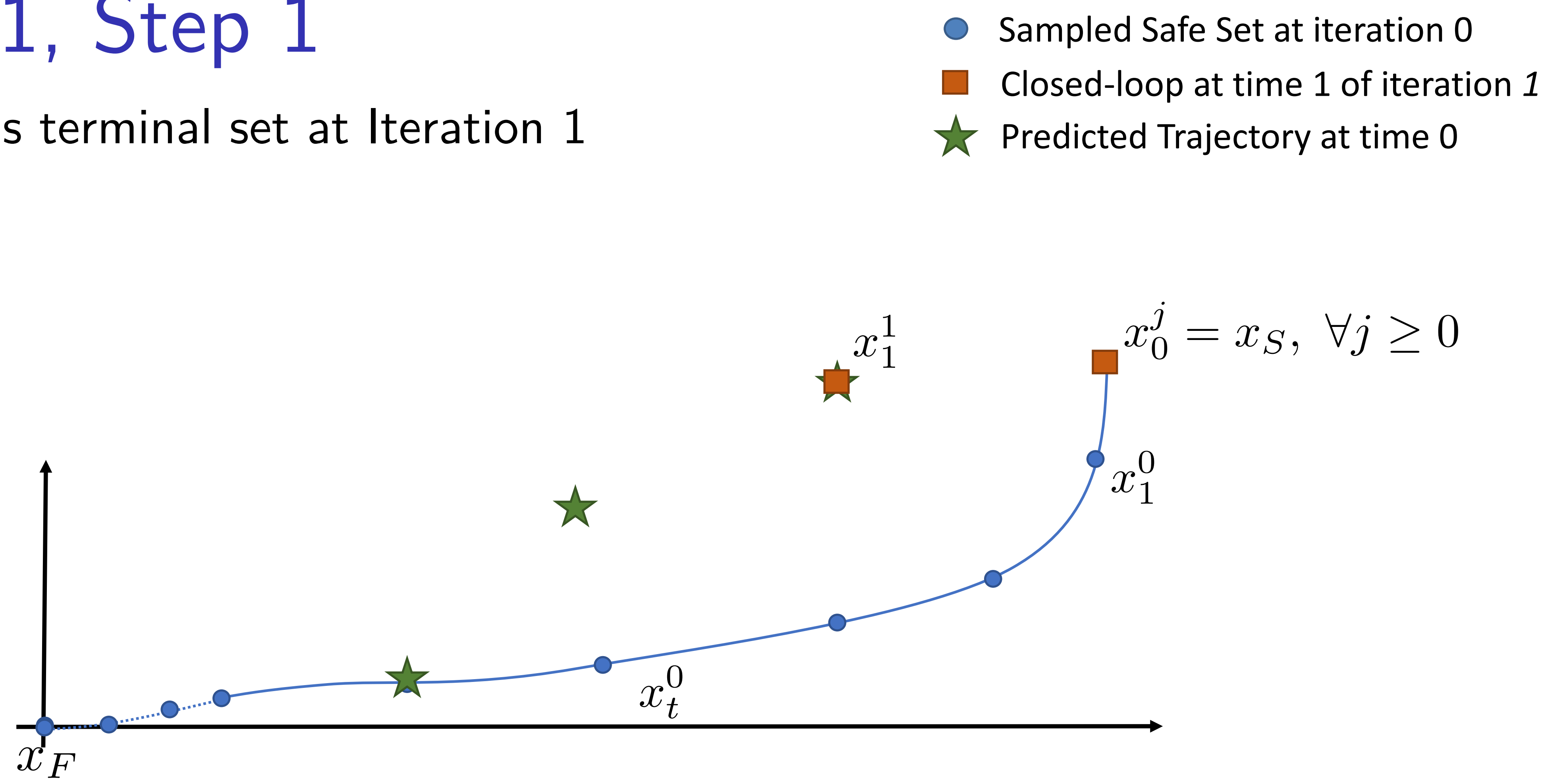
# Iteration 1, Step 0

Use $\mathcal{SS}^0$ as terminal set at Iteration 1



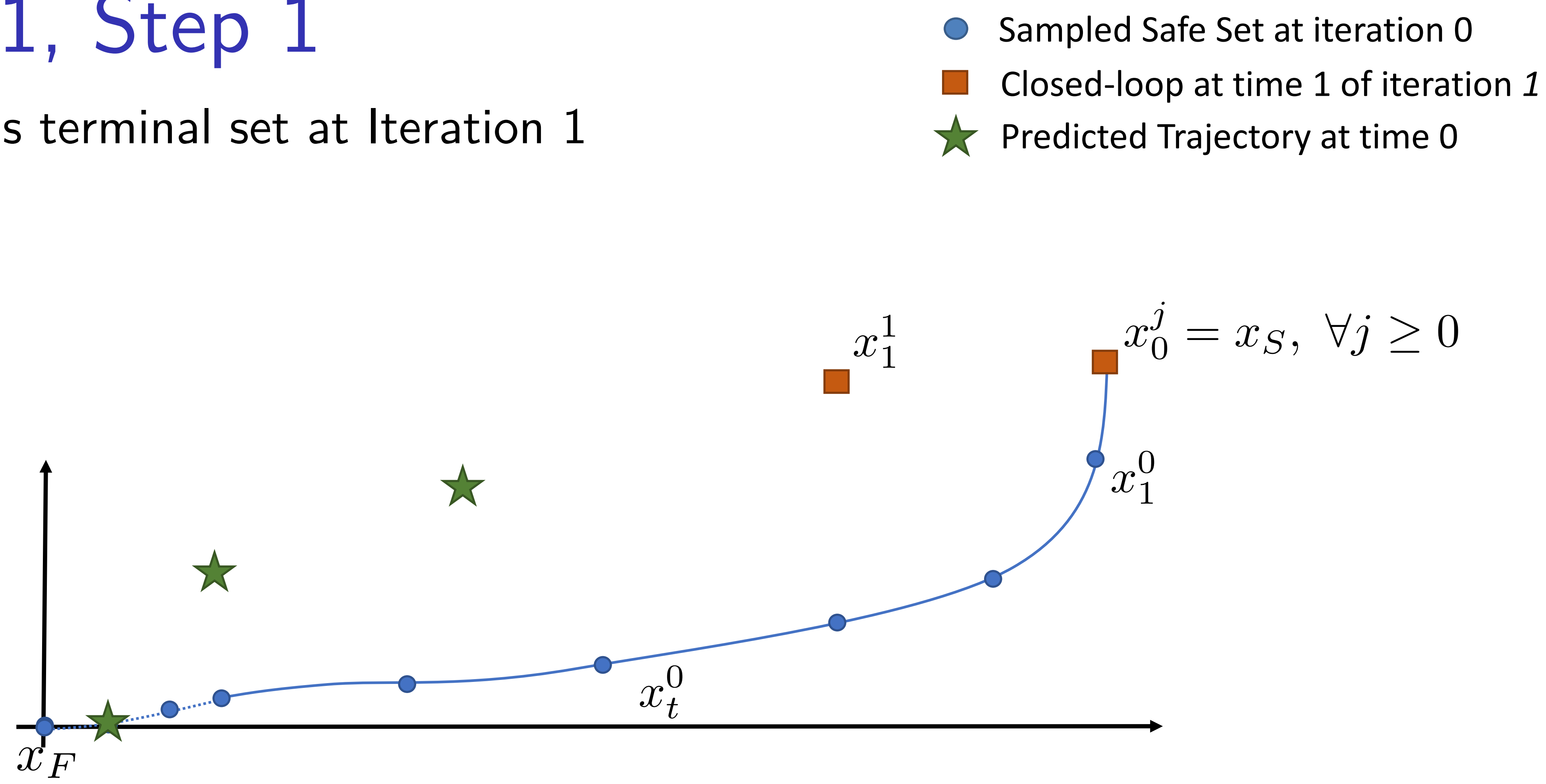
# Iteration 1, Step 1

Use $\mathcal{SS}^0$ as terminal set at Iteration 1



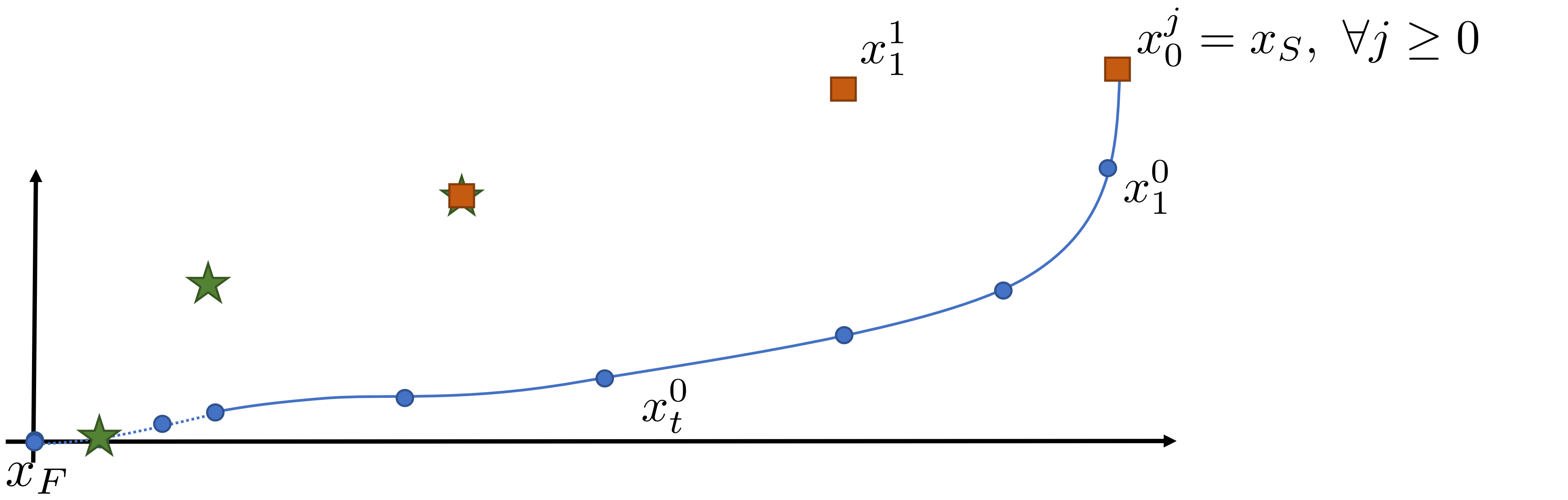
# Iteration 1, Step 1

Use $\mathcal{SS}^0$ as terminal set at Iteration 1



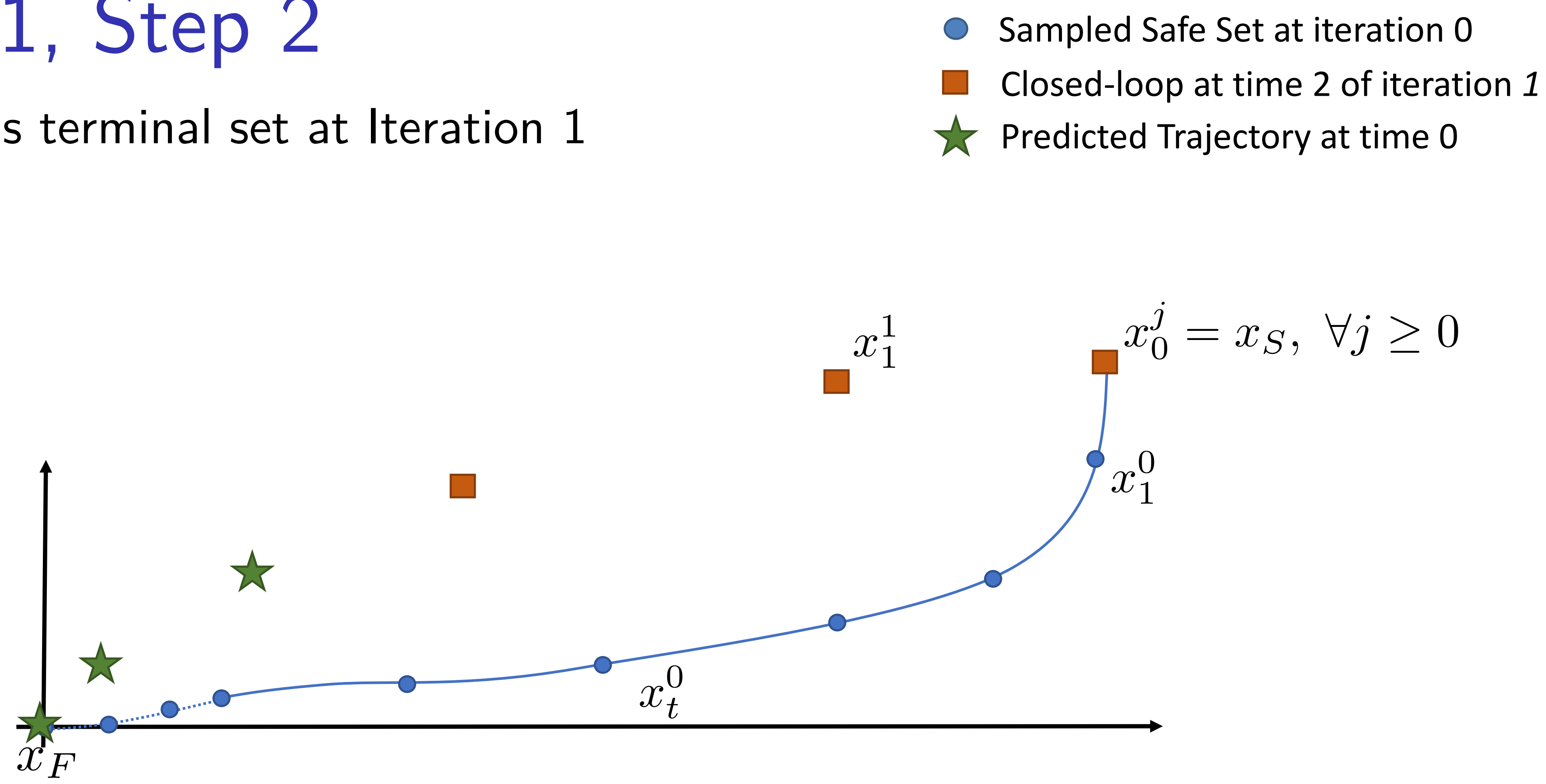
# Iteration 1, Step 2

Use $\mathcal{SS}^0$ as terminal set at Iteration 1



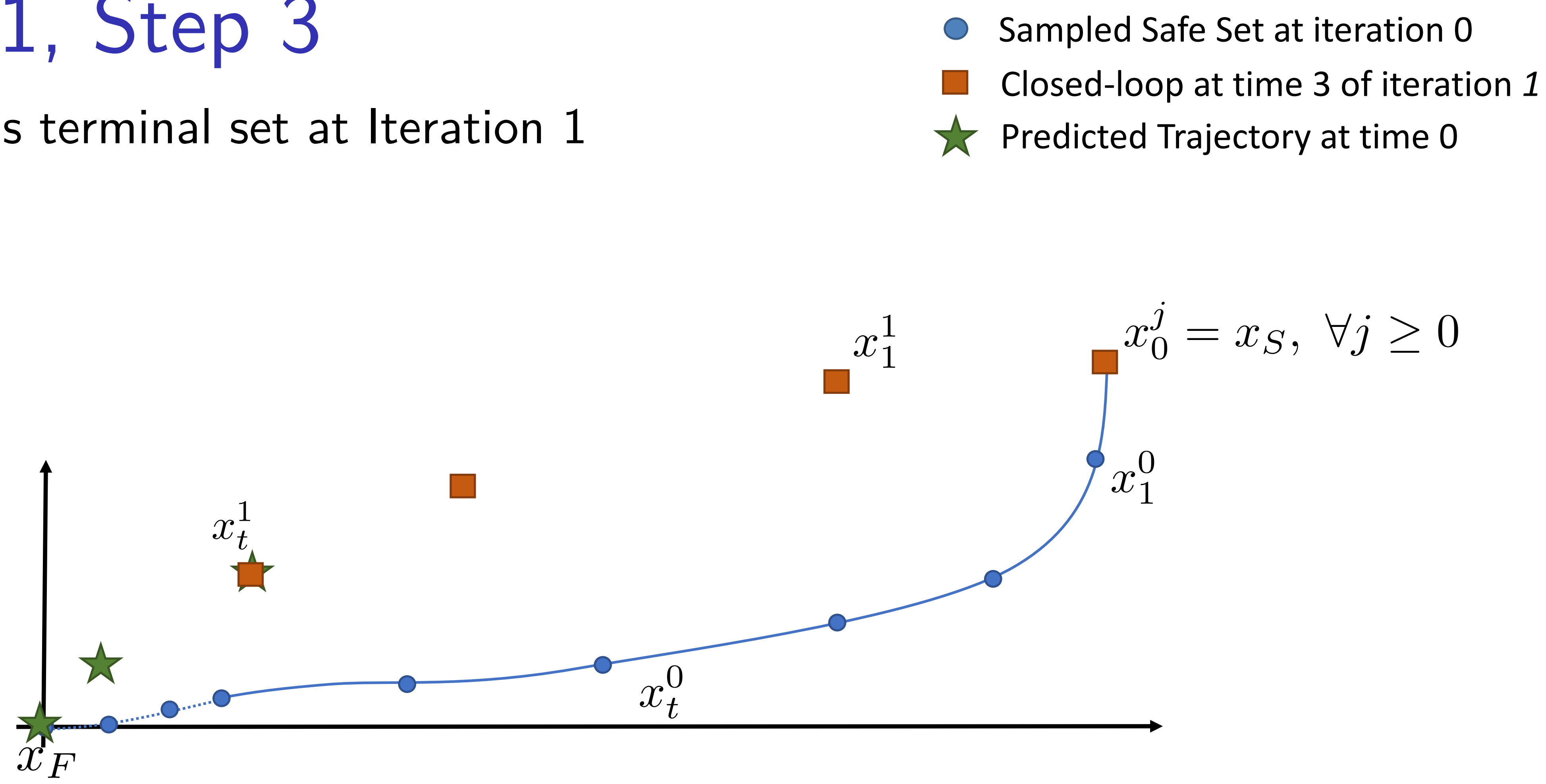
# Iteration 1, Step 2

Use $\mathcal{SS}^0$ as terminal set at Iteration 1



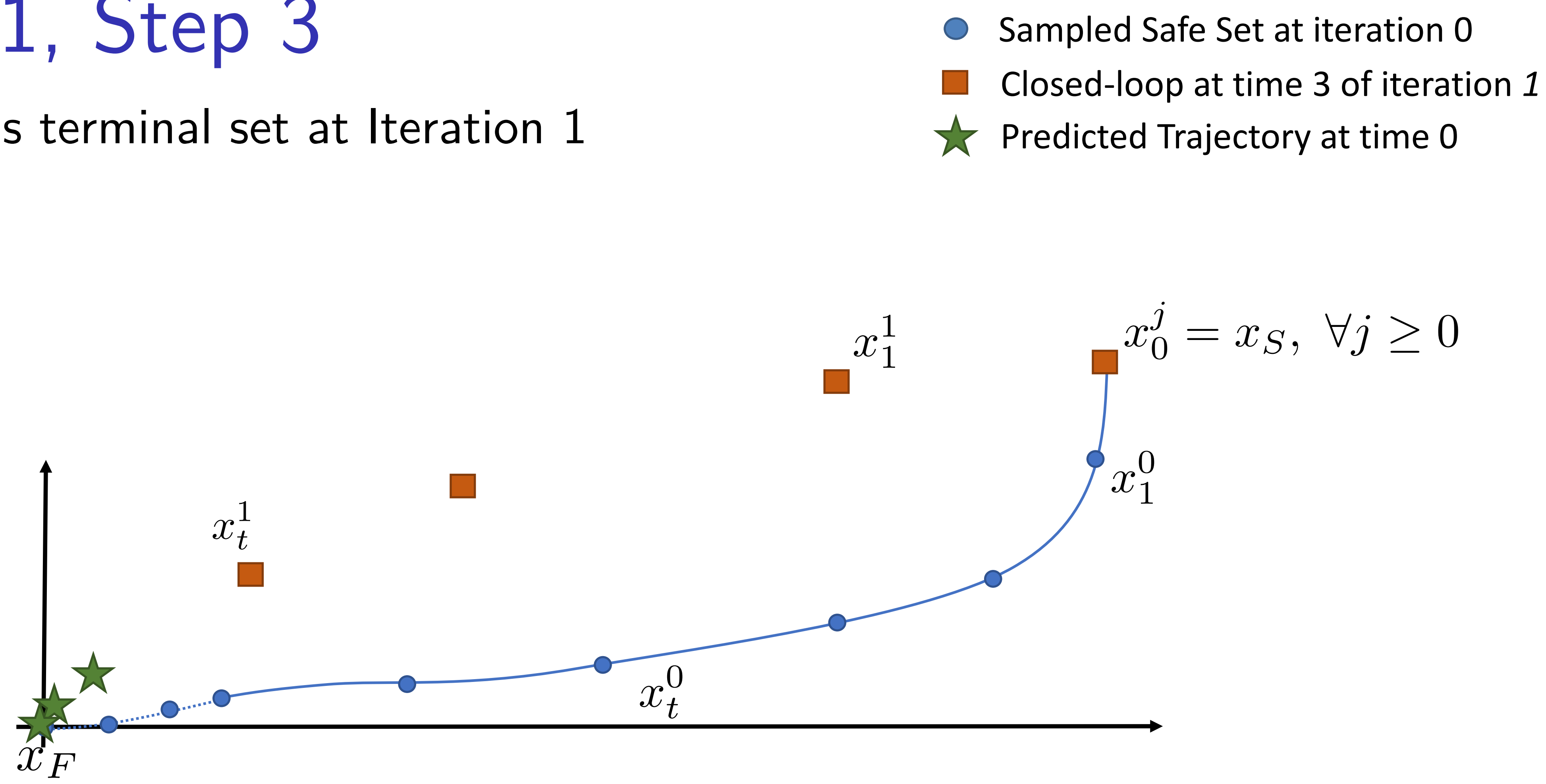
# Iteration 1, Step 3

Use $\mathcal{SS}^0$ as terminal set at Iteration 1



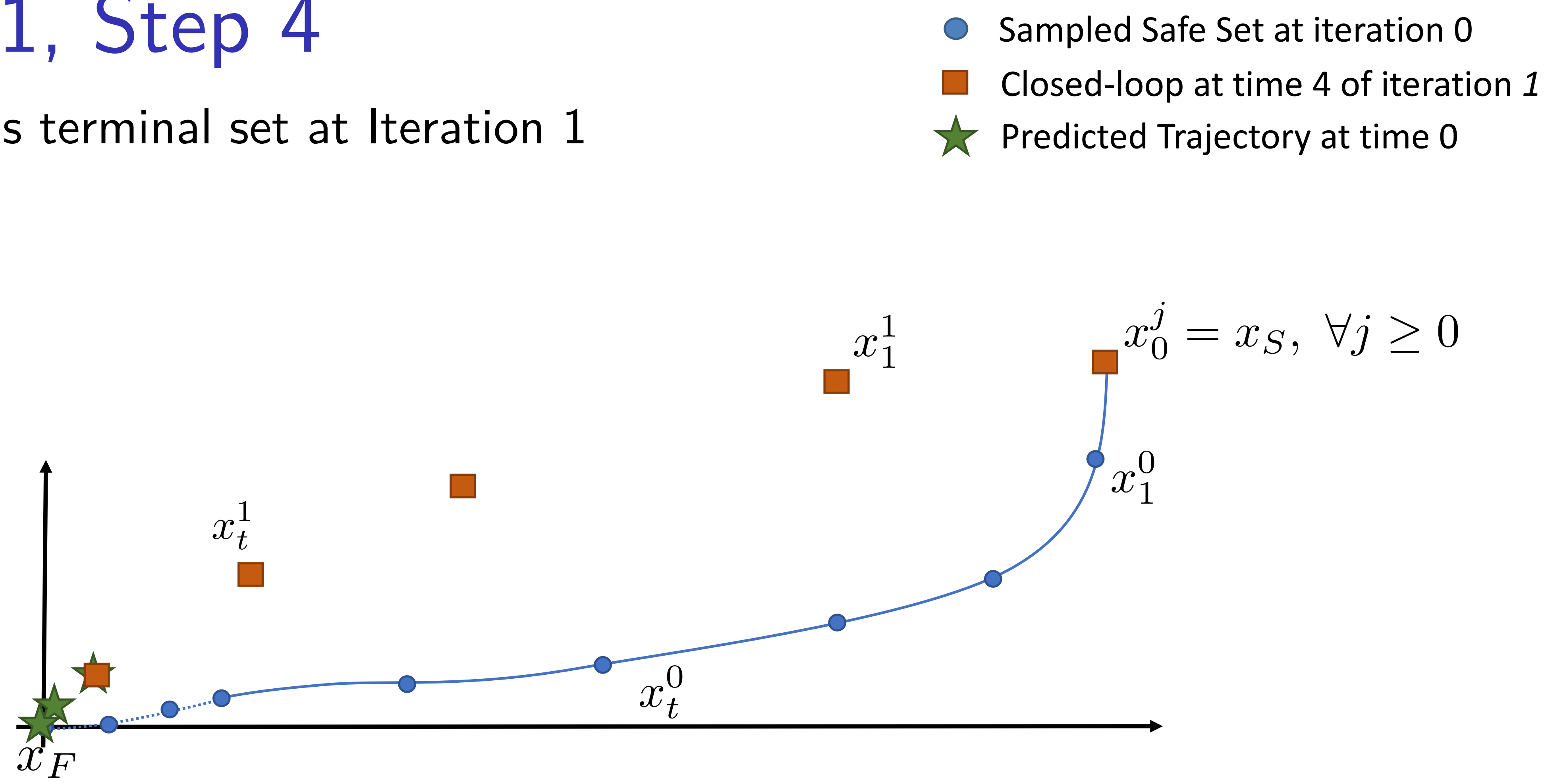
# Iteration 1, Step 3

Use $\mathcal{SS}^0$ as terminal set at Iteration 1



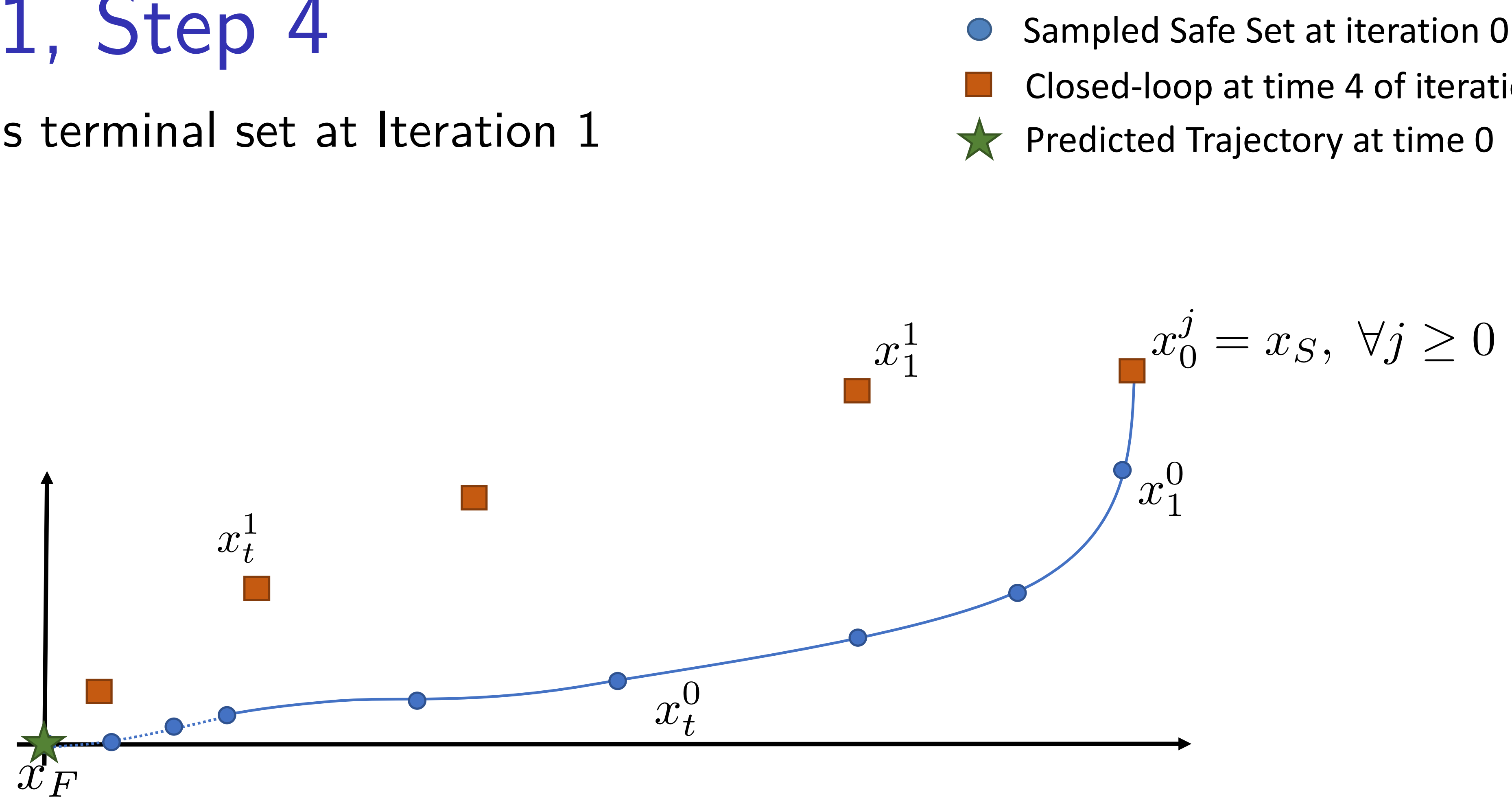
# Iteration 1, Step 4

Use $\mathcal{SS}^0$ as terminal set at Iteration 1



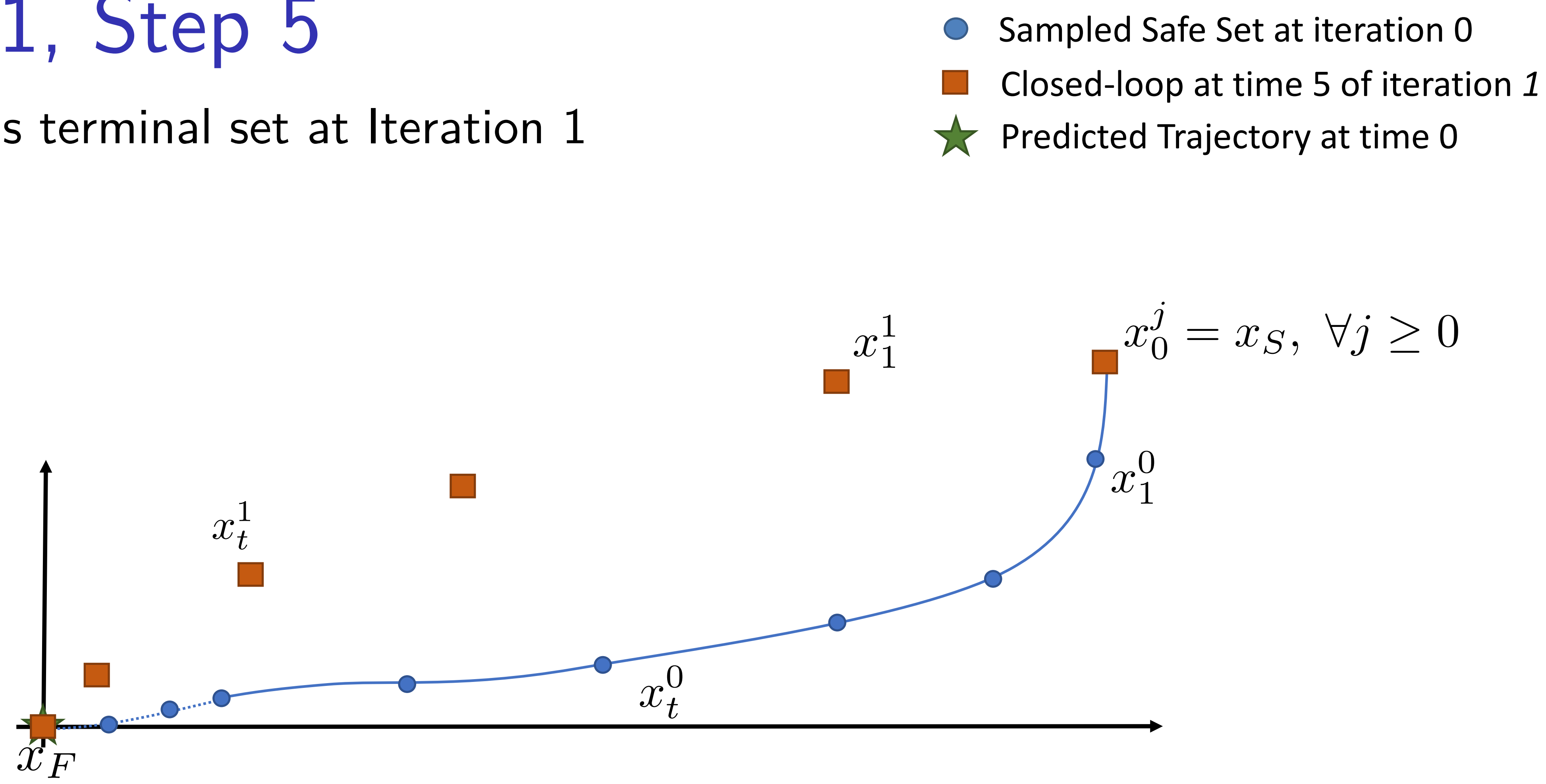
# Iteration 1, Step 4

Use $\mathcal{SS}^0$ as terminal set at Iteration 1



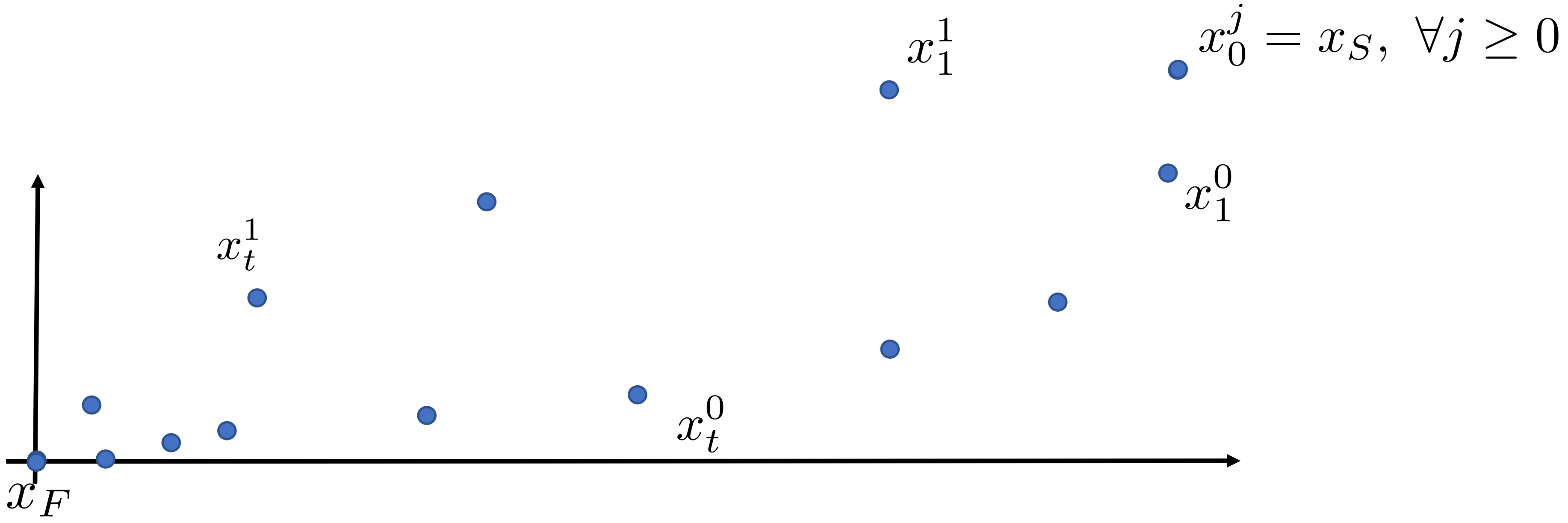
# Iteration 1, Step 5

Use $\mathcal{SS}^0$ as terminal set at Iteration 1



# Iteration 1 Safe Set

Update the safe set with the new closed-loop trajectory

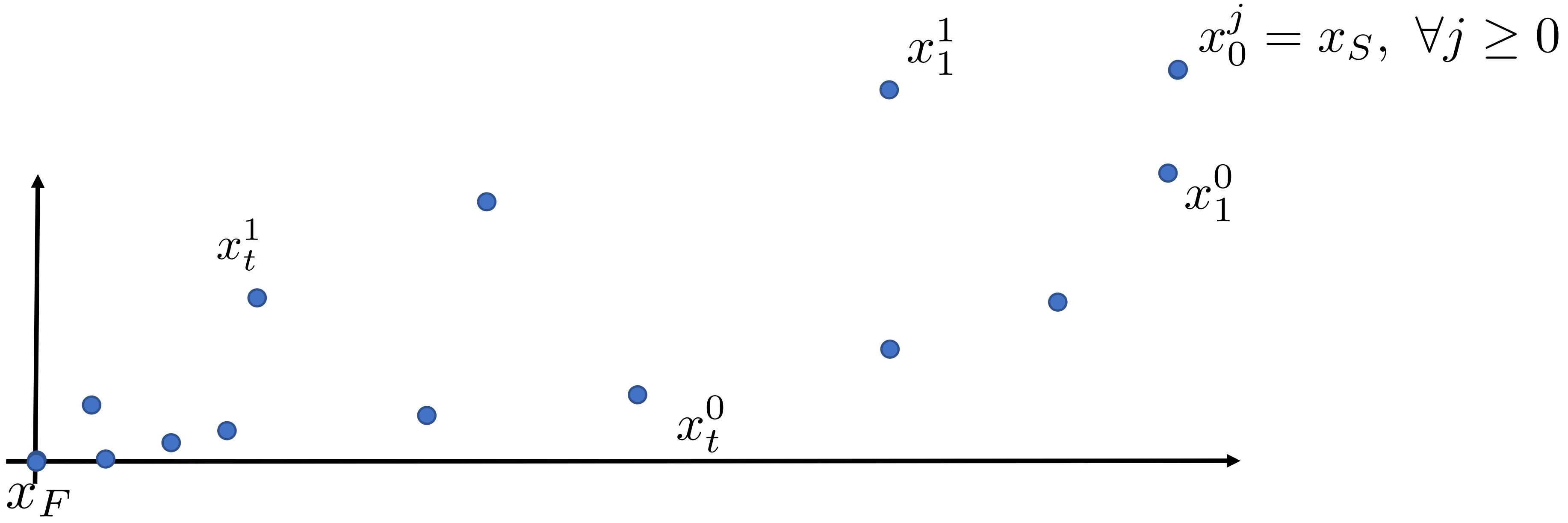



**Definition: Sampled Safe Set**

$$\mathcal{SS}^1 = \left\{ \bigcup_{i=0}^{1} \bigcup_{t=0}^{\infty} x_t^i \right\} \supseteq \mathcal{SS}^0$$

# Iteration $j$ Safe Set

Update the safe set with the new closed-loop trajectory

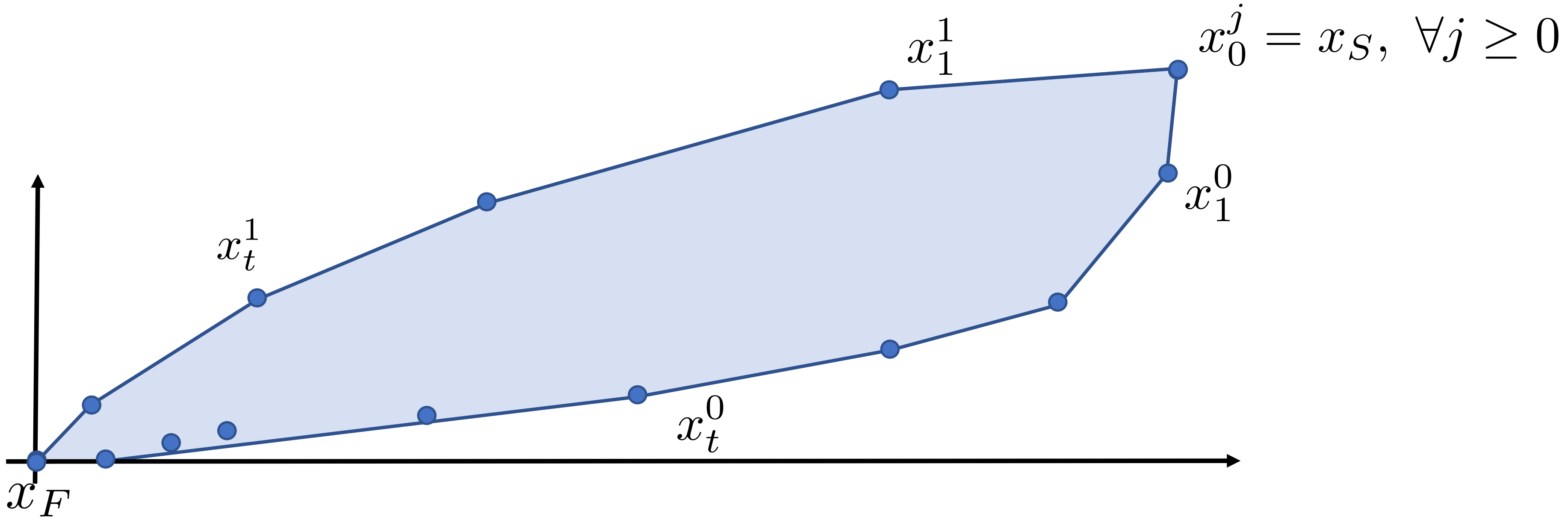



## Definition: Sampled Safe Set

$$\mathcal{SS}^j = \left\{ \bigcup_{i=0}^{j} \bigcup_{t=0}^{\infty} x_t^i \right\} \supseteq \mathcal{SS}^{j-1}$$

# Terminal Set: Convex hull of Sample Safe Set

Update the safe set with the new closed-loop trajectory




**Definition: Convex Safe Set**

$$\mathcal{CS}^j = \text{Conv}(\mathcal{SS}^j) = \text{Conv}\left(\left\{\bigcup_{k=0}^{j}\bigcup_{t=0}^{\infty} x_t^i\right\}\right)$$

At every iteration *j*
A Control Invariant Set
for Constrained Linear Dynamical Systems

# Learning Model Predictive Control (LMPC) – Key Idea

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

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

s.t.

$$x^j_{k+1|t} = f(x^j_{k|t}, u^j_{k|t}),\ \forall k \in [t, \cdots, t+N-1]$$

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

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

$$x^j_{t+N|t} \in \mathcal{SS}^{j-1},$$

**Safe Set**

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

# Learning Model Predictive Control (LMPC) – Key Idea

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

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

s.t.

$$x_{k+1|t}^j = f(x_{k|t}^j, u_{k|t}^j),\ \forall k \in [t,\cdots,t+N-1]$$

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

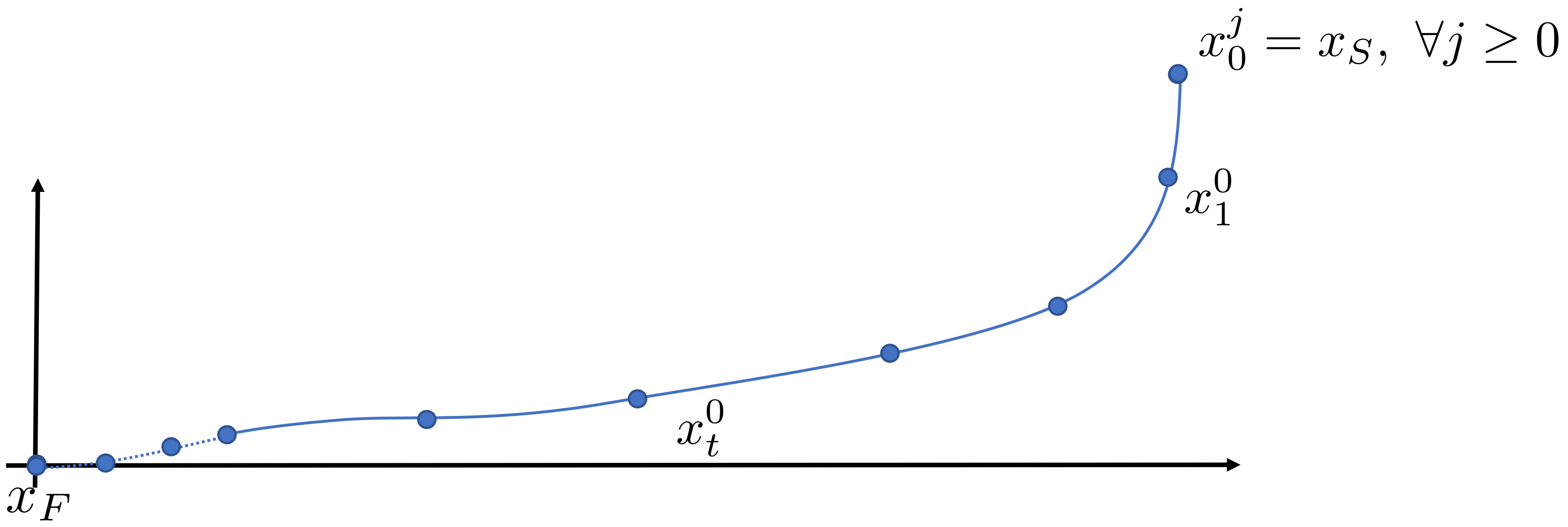
$$x_{k|t}^j \in \mathcal{X},\ u_{k|t}^j \in \mathcal{U},\ \forall k \in [t,\cdots,t+N-1]$$

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

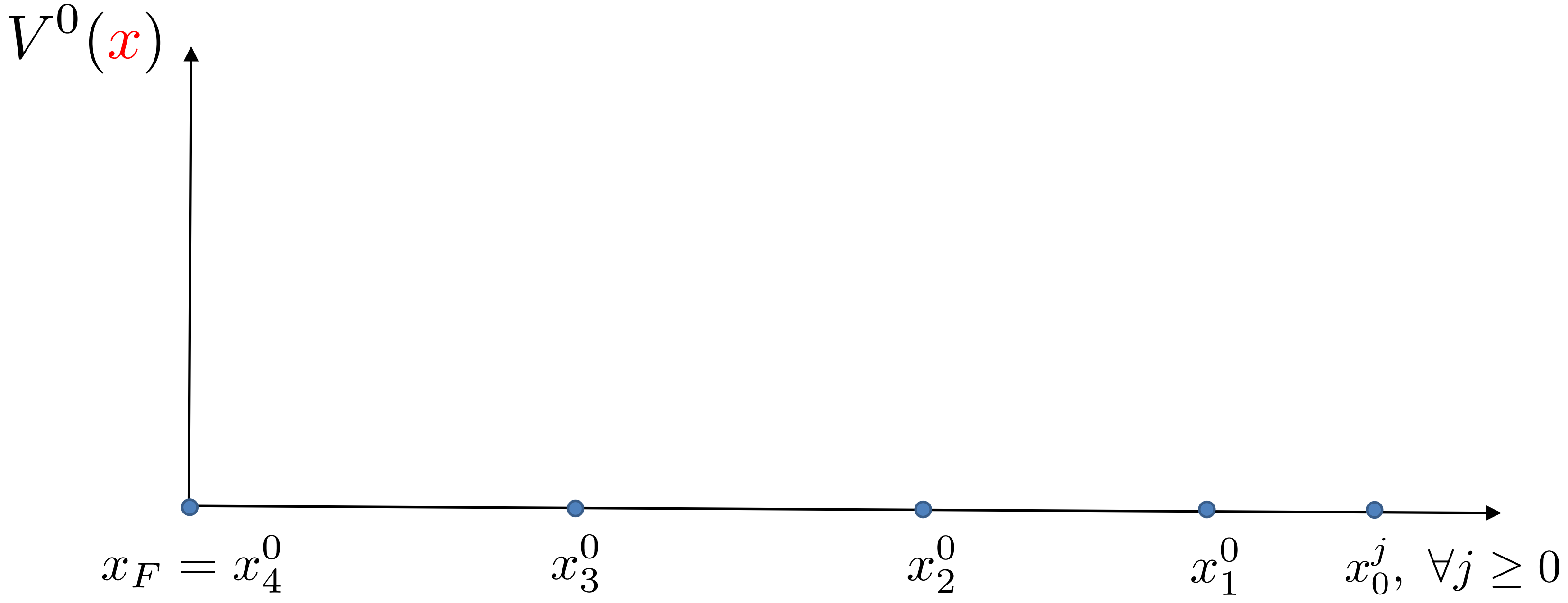
Value Function

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

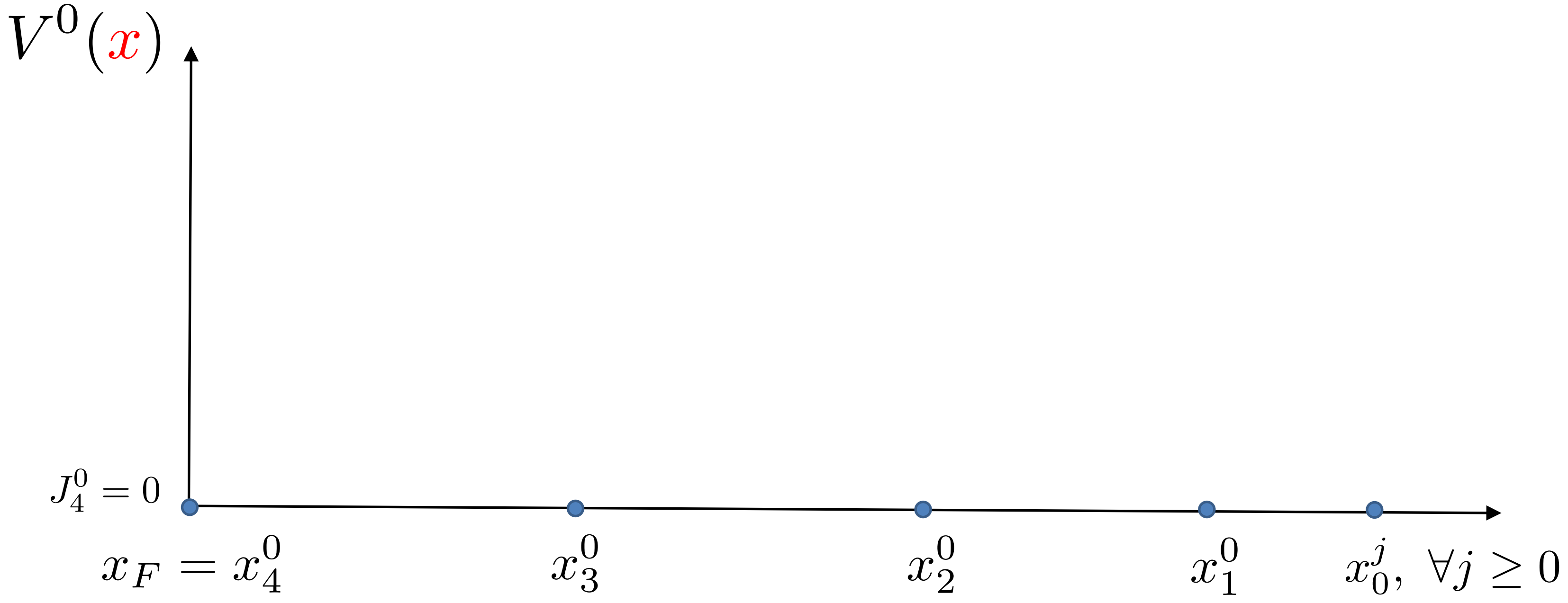
# Terminal Cost at Iteration 0



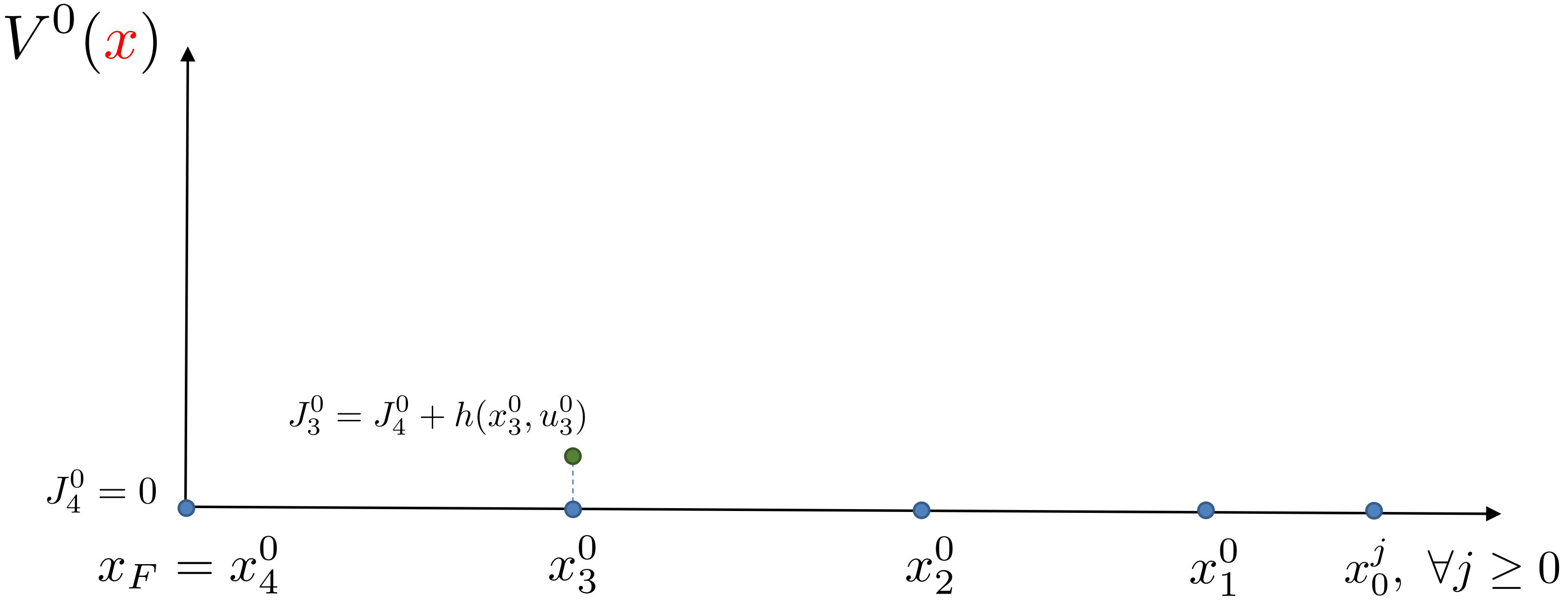
# Terminal Cost at Iteration 0



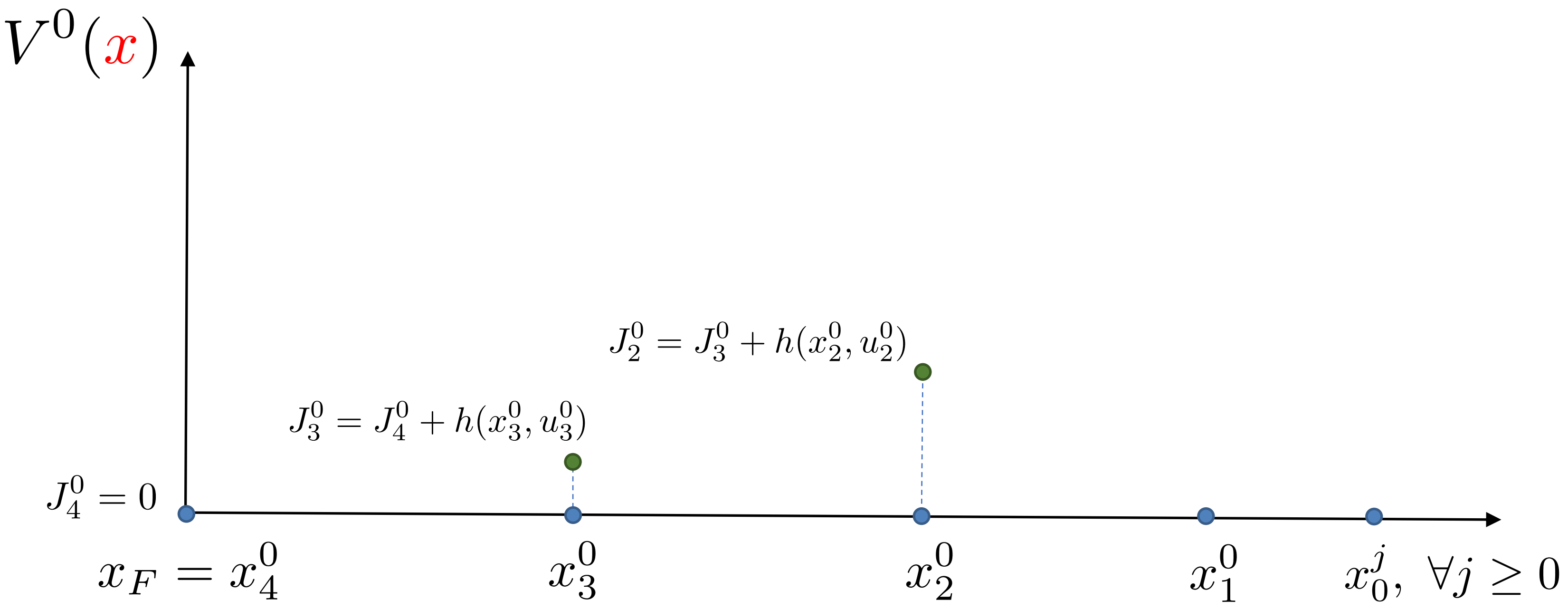
# Terminal Cost at Iteration 0



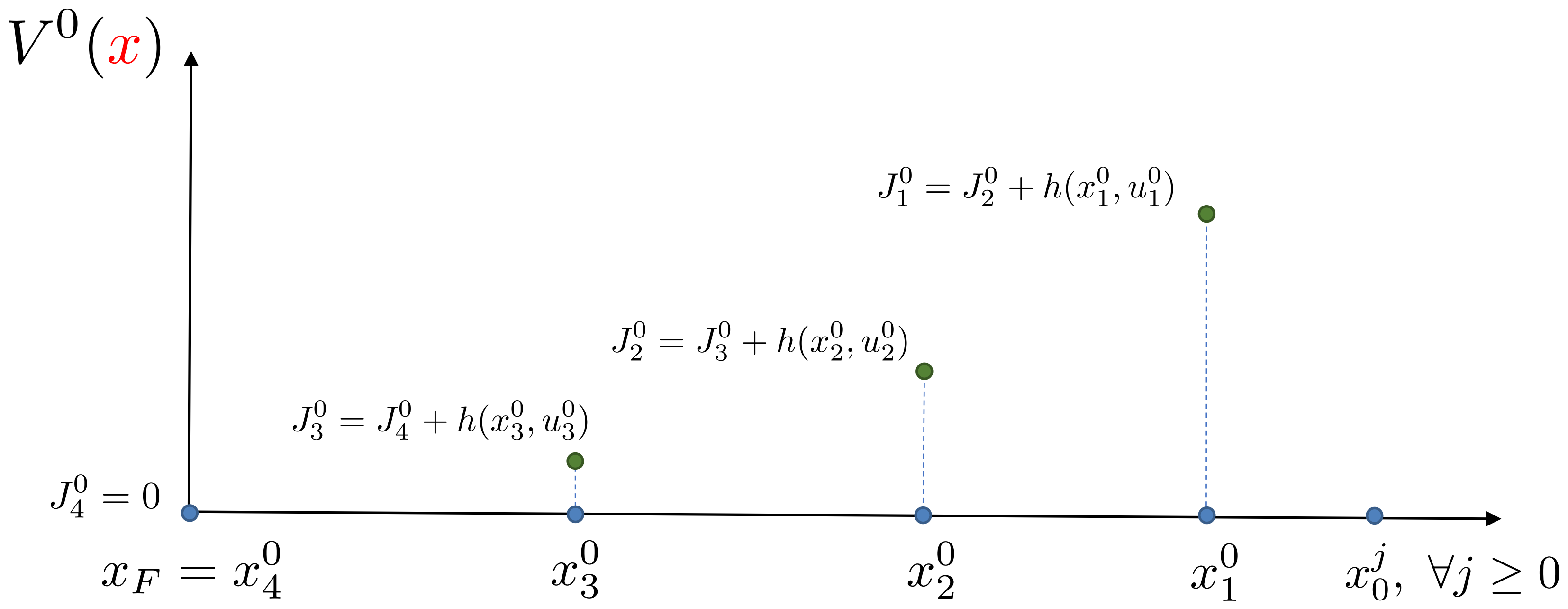
# Terminal Cost at Iteration 0



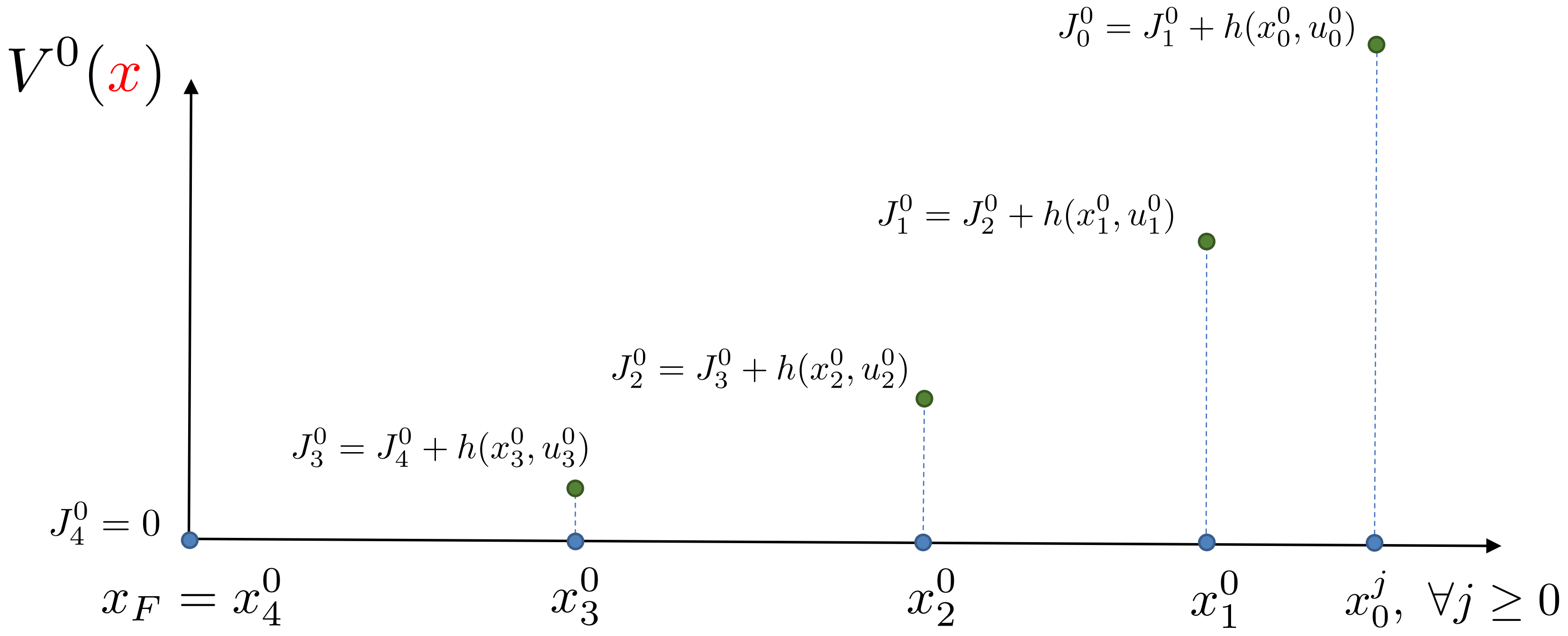
# Terminal Cost at Iteration 0



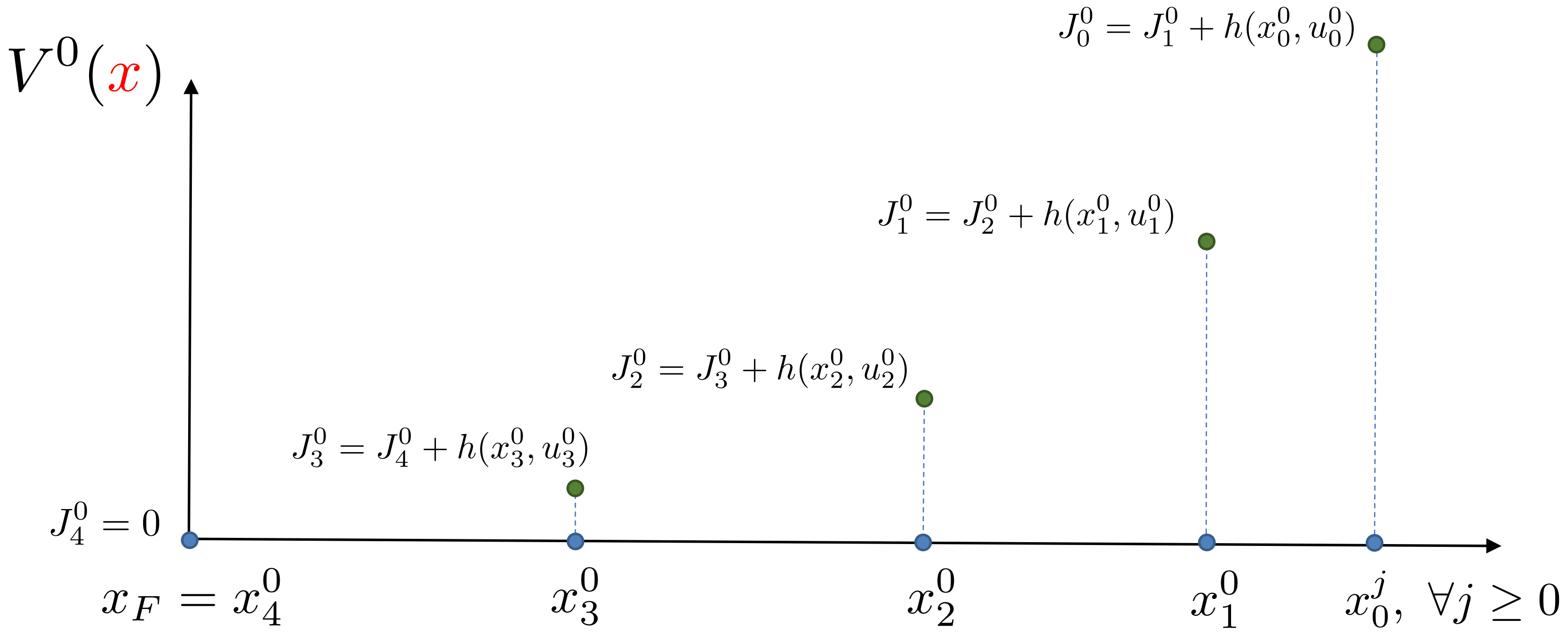
# Terminal Cost at Iteration 0



# Terminal Cost at Iteration 0



# Terminal Cost at Iteration 0

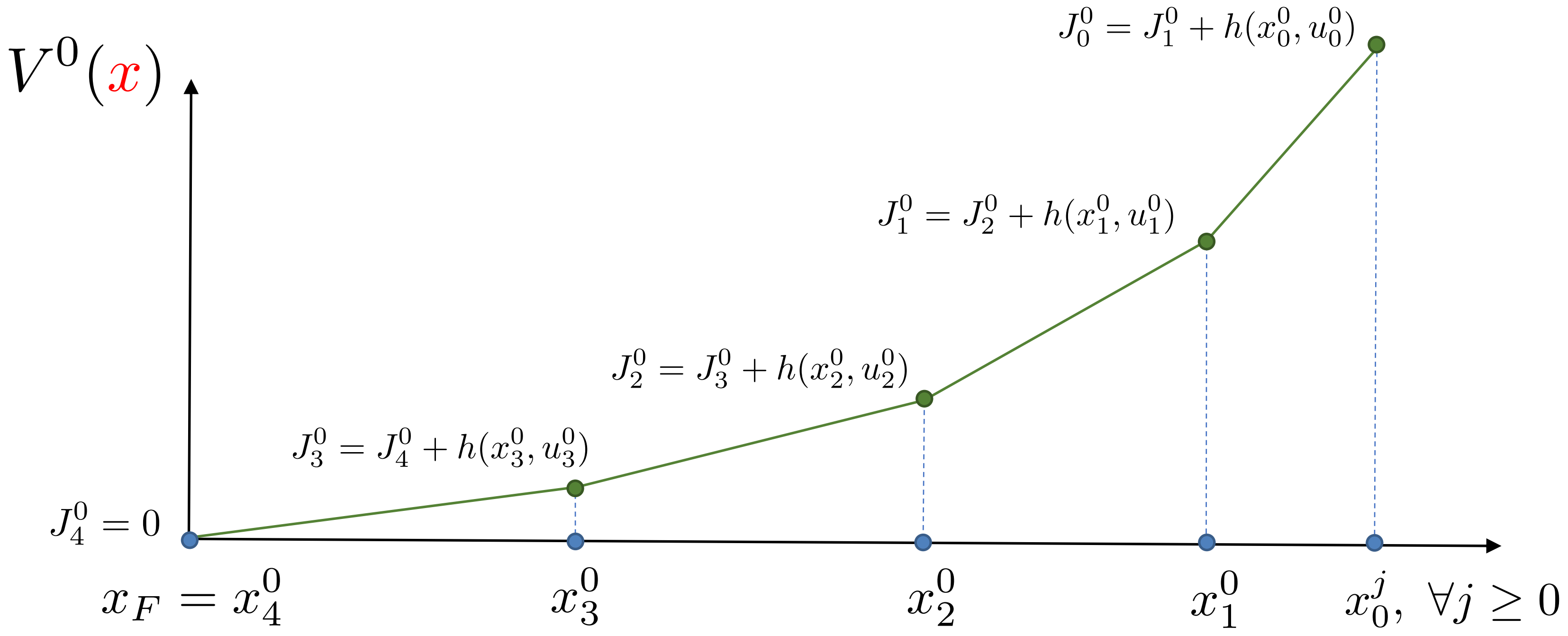



**Definition: V-function**

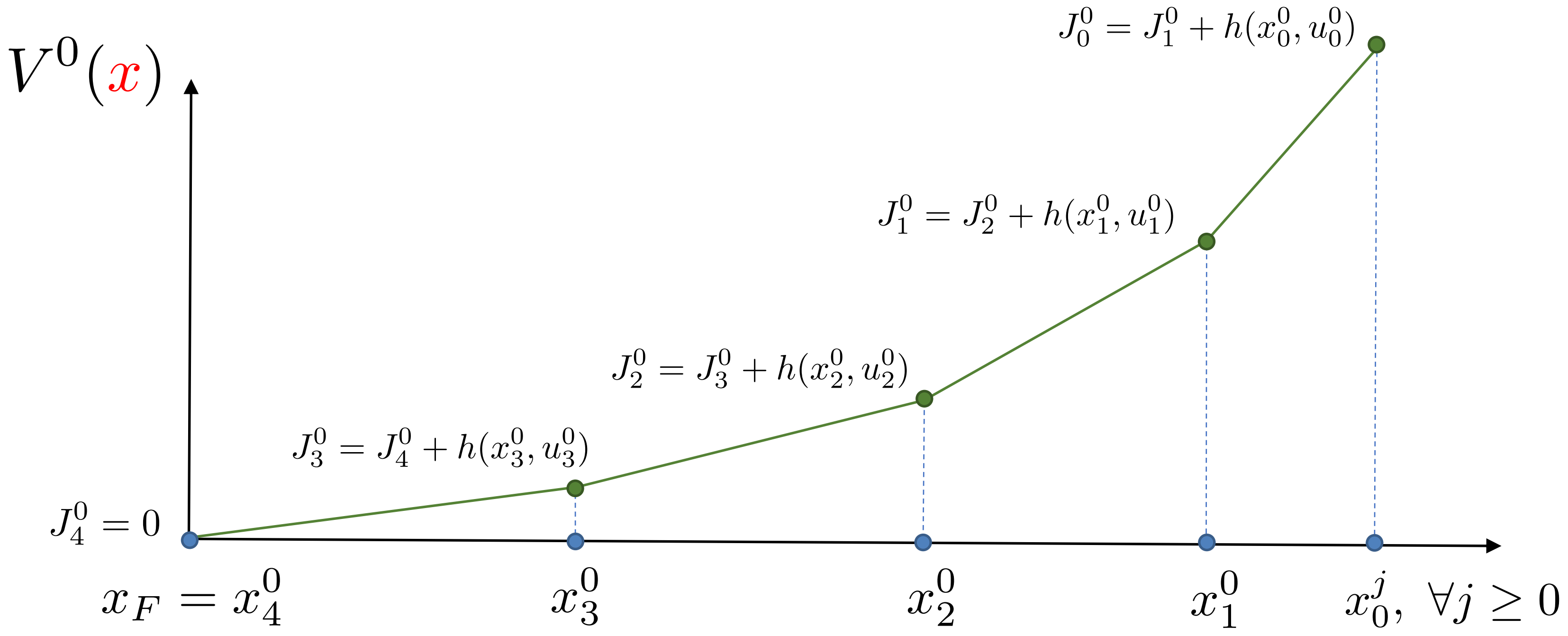
$$V^0(x) = \begin{cases} \sum_{k=t}^{\infty} h(x_k^0, u_k^0), & \text{if } x = x_t^0 \in \mathcal{SS}^0 \\ +\infty, & \text{if } x \notin \mathcal{SS}^0 \end{cases}$$

At iteration *0*
A Control Lyapunov Function
for Constrained Nonlinear Dynamical Systems

# Convex Terminal Cost at Iteration 0



# Convex Terminal Cost at Iteration 0

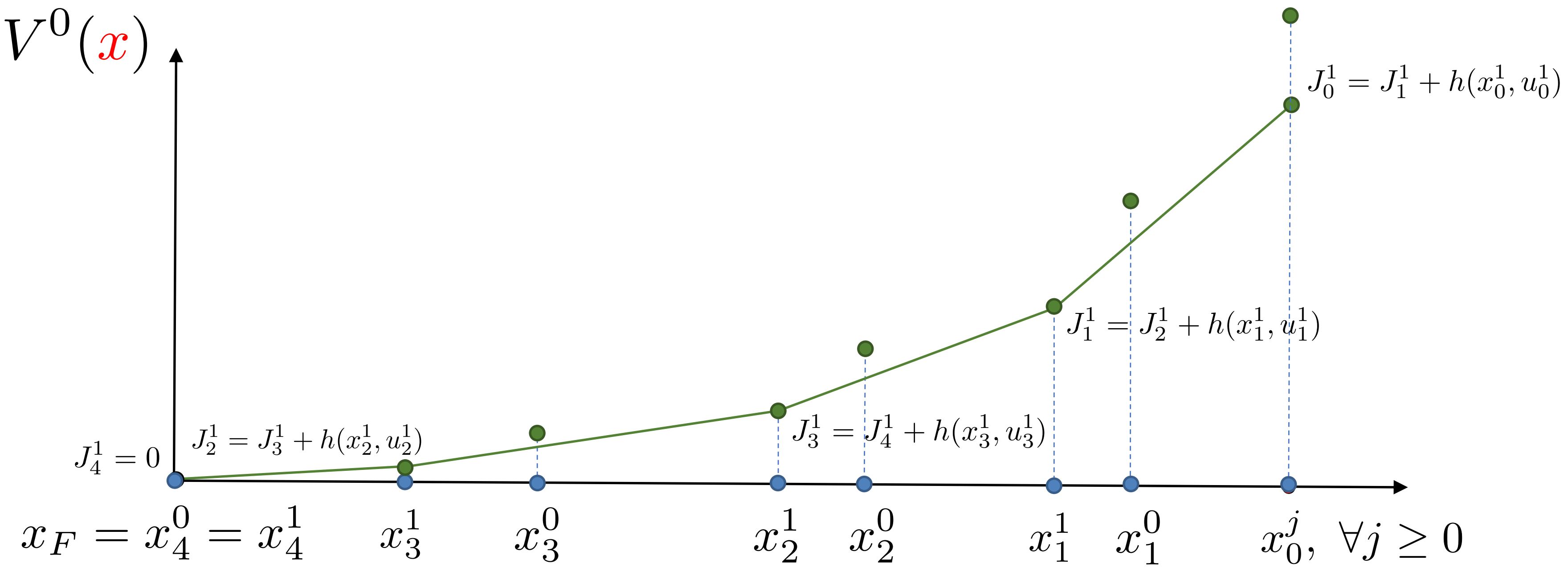



**Value Function Approximation**

$$V_c^0(x) = \min_{\lambda_i^0 \in [0,1]} \quad \sum_i J_i^0 \lambda_i^0$$
$$\text{s.t} \quad \sum_i x_i^0 \lambda_i^0 = x$$
$$\sum_i \lambda_i^0 = 1$$

At iteration *0*
A Control Lyapunov Function
for Constrained Linear Dynamical Systems

# Convex Terminal Cost at Iteration 1




**Value Function Approximation**

$$V_c^j(x) = \min_{\lambda_i^j \in [0,1]} \sum_i \sum_j J_i^j \lambda_i^j$$

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

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

At every iteration *j*
A Control Lyapunov Function
for Constrained Linear Dynamical Systems

# LMPC Summary

At each time $t$ of iteration $j$, solve

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

s.t.

$$x^j_{k+1|t} = f(x^j_{k|t}, u^j_{k|t}),\ \forall k \in [t, \cdots, t+N-1]$$

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

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

$$x^j_{t+N|t} \in \ \mathcal{SS}^{j-1},$$

# LMPC Summary

At each time $t$ of iteration $j$, solve

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

s.t.

$$x^j_{k+1|t} = f(x^j_{k|t}, u^j_{k|t}),\ \forall k \in [t,\cdots,t+N-1]$$

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

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

$$x^j_{t+N|t} \in \mathcal{SS}^{j-1},$$

Constructed using historical data

# LMPC Summary

At each time $t$ of iteration $j$, solve

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

$$\begin{aligned}
\text{s.t.}\quad & \\
& x^j_{k+1|t} = f(x^j_{k|t}, u^j_{k|t}),\ \forall k \in [t, \cdots, t+N-1] \\
& x^j_{t|t} = x^j_t, \\
& x^j_{k|t} \in \mathcal{X},\ u^j_{k|t} \in \mathcal{U},\ \forall k \in [t, \cdots, t+N-1] \\
& x^j_{t+N|t} \in \mathcal{SS}^{j-1},
\end{aligned}$$

Constructed using historical data

**Guarantees for constrained (linear) systems [1,2]**

The properties of the (convex) safe set and (convex) Q-function allows us to guarantee:

- Safety: constraint satisfaction at iteration *j* → satisfaction at iteration *j+1*
- Non-decreasing Performance: closed-loop cost at iteration *j* >= closed-loop cost at iteration *j+1*
- Performance Improvement: closed-loop cost strictly deceasing at each iteration (LICQ required)
- (Global) optimality: steady state trajectory is optimal for the original problem (LICQ required)


[1] U. Rosolia, F. Borrelli. "Learning model predictive control for iterative tasks. a data-driven control framework." *IEEE Transactions on Automatic Control* (2018).
[2] U. Rosolia, F. Borrelli. "Learning model predictive control for iterative tasks: A computationally efficient approach for linear system." *IFAC-PapersOnLine* (2017)

# Practical Implementation

LMPC convex formulation and the constrained LQR example

# Linear(ized) LMPC

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

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

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)

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

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)

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

$$\sum_{i=0}^{j-1}\sum_k J^i_k(x^i_k)\lambda^i_k \Longleftrightarrow V^{j-1}_c(x_{t+N|t})$$

s.t.

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

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

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

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

# Linear(ized) LMPC

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

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

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

s.t.

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

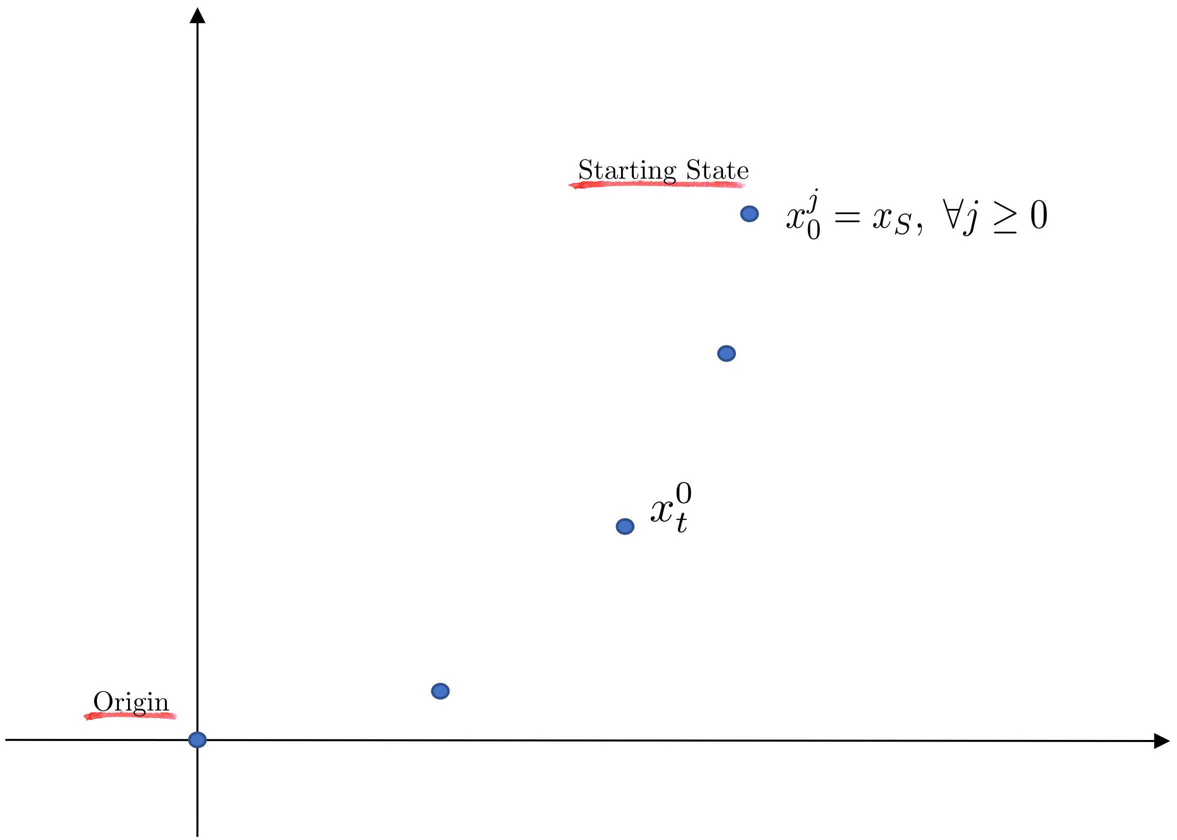
$$x_{t|t} = x^j_t,$$

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

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

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

# Example I: Constrained LQR




**Infinite Time Optimal Control Problem**

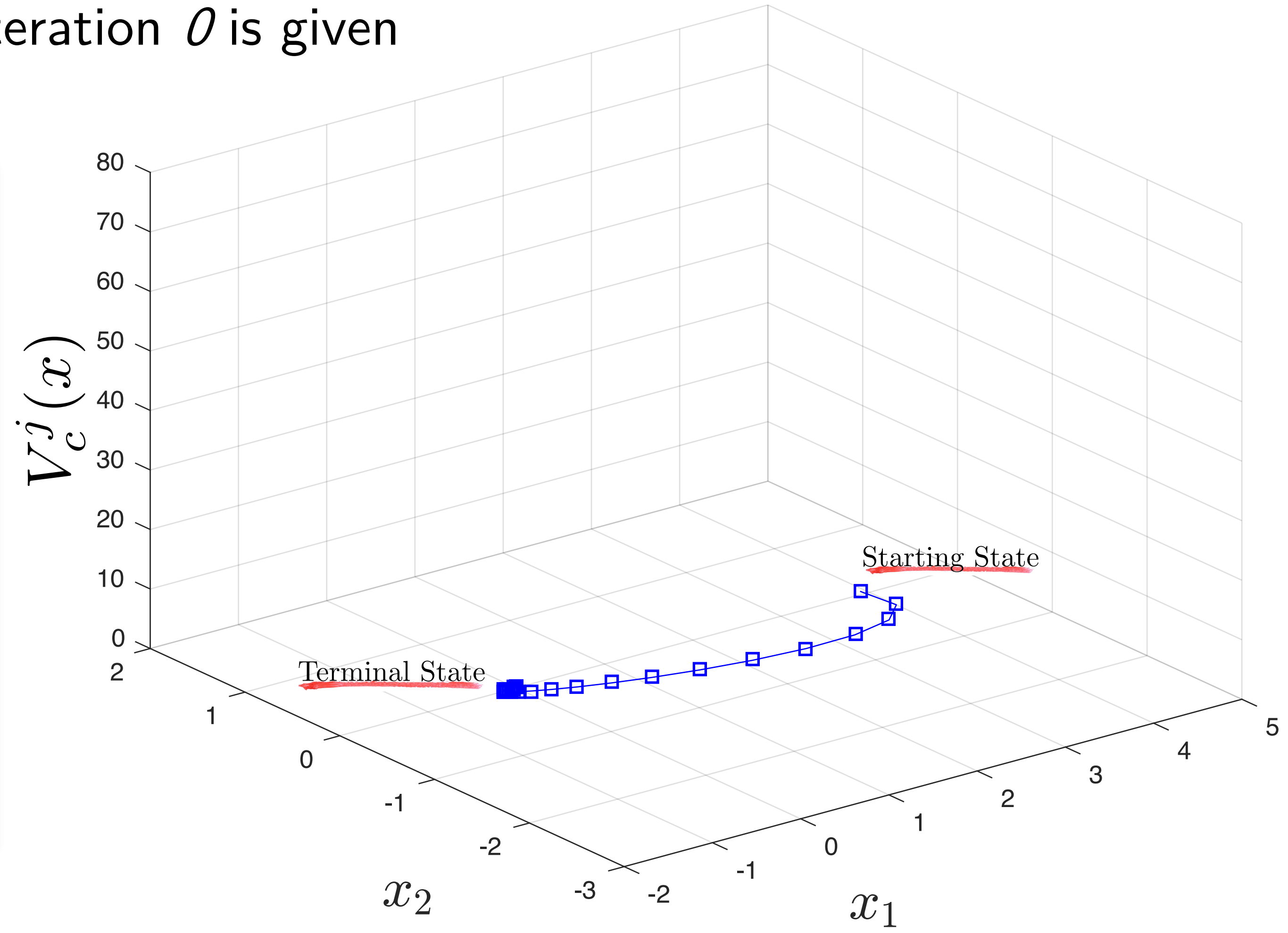
The goal of the control design is to solve the following constrained LQR problem for the double integrator system,

$$
\begin{aligned}
\min_{u_0, u_1, \ldots} \quad & \sum_{k=0}^{\infty} x_k^\top Q x_k + u_k^\top R u_k \\
\text{s.t.} \quad & x_0 = x_S, \\
& x_{k+1} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} x_k + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u_k, \\
& x_k \in \mathcal{X},\ u_k \in \mathcal{U},\ \forall k \geq 0
\end{aligned}
$$

# Example I: Constrained LQR

Assumption: A first feasible trajectory at iteration *0* is given

## Iterative LMPC



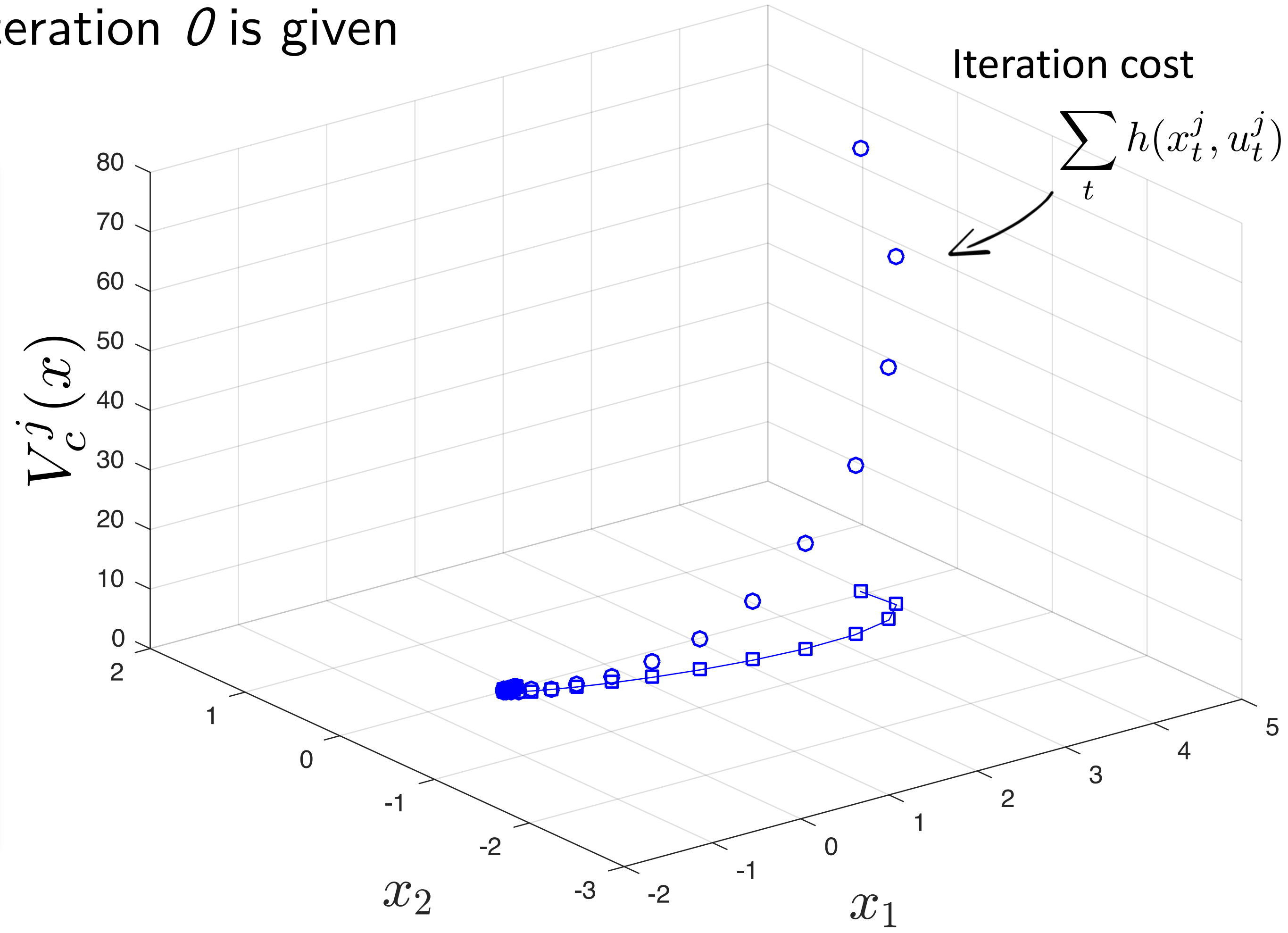
# Example I: Constrained LQR

Assumption: A first feasible trajectory at iteration *0* is given

**Iterative LMPC**

Step 0: Set iteration counter $j=0$

→ Step 1: Compute the roll-out cost for the recorded data up to iteration $j$



# Example I: Constrained LQR

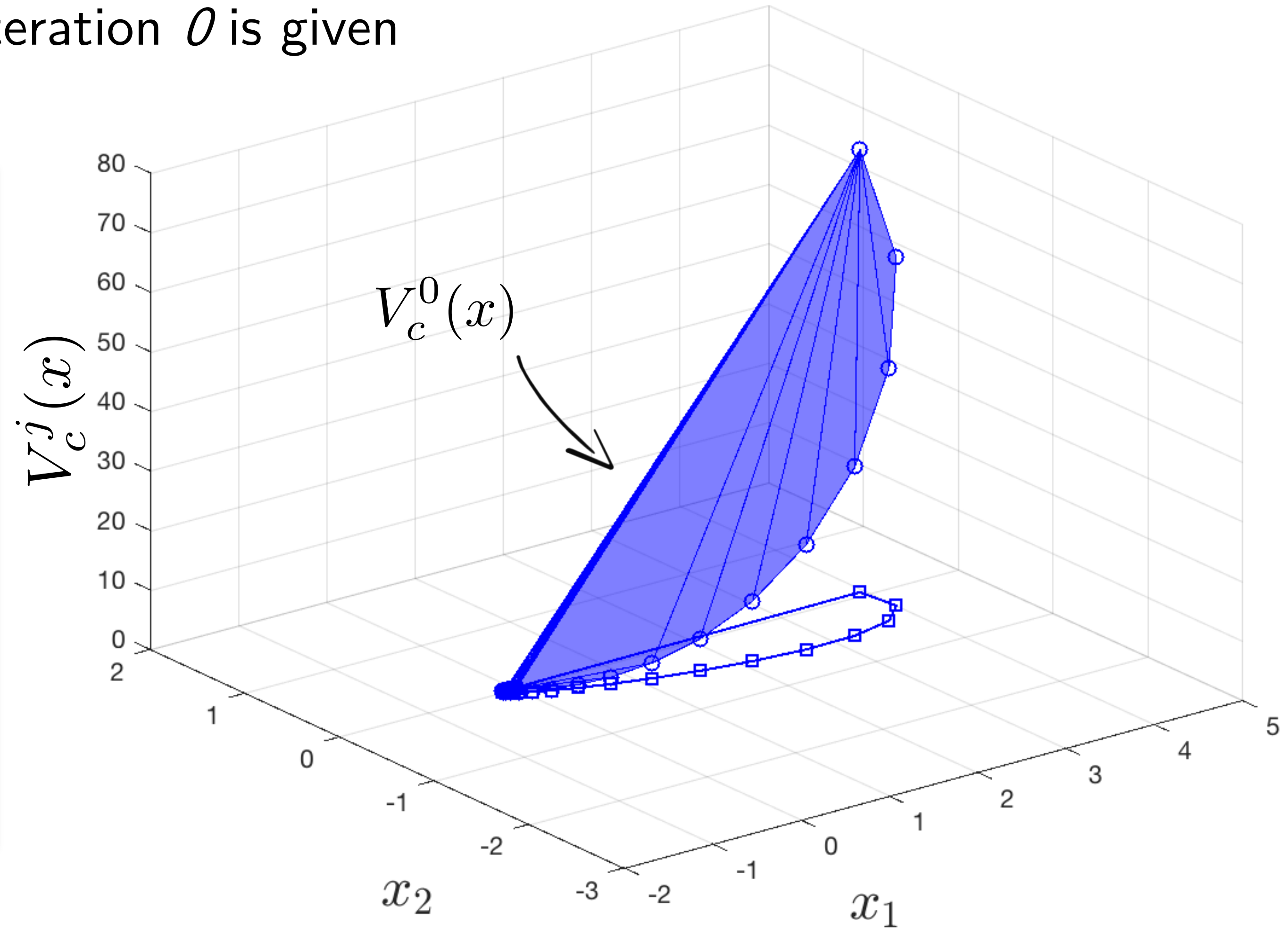
Assumption: A first feasible trajectory at iteration *0* is given

**Iterative LMPC**

Step 0: Set iteration counter $j=0$

Step 1: Compute the roll-out cost for the recorded data up to iteration $j$

→ Step 2: Define $V^j$ which interpolates linearly the roll-out cost



# Example I: Constrained LQR

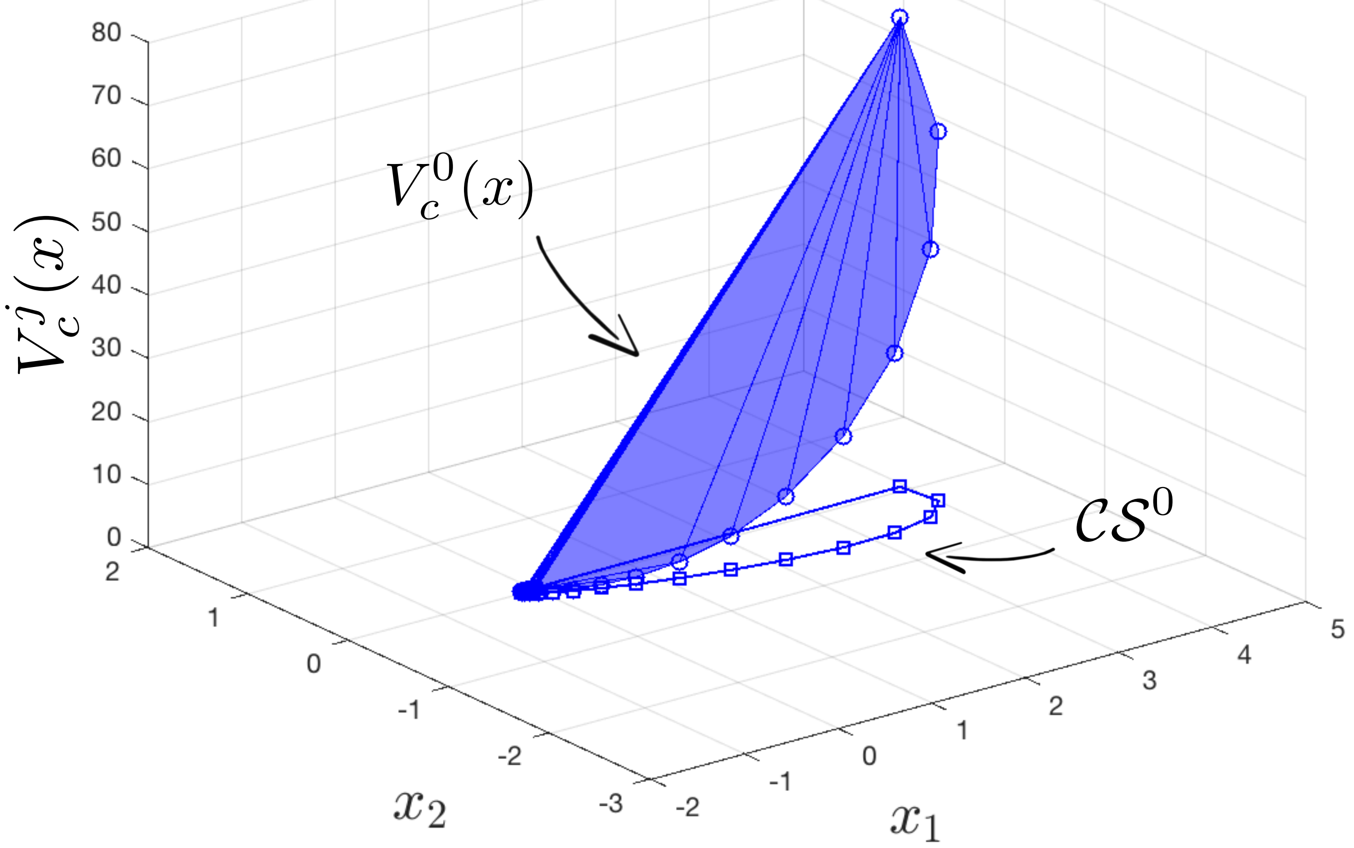
Assumption: A first feasible trajectory at iteration *0* is given

**Iterative LMPC**

Step 0: Set iteration counter *j=0*

Step 1: Compute the roll-out cost for the recorded data up to iteration *j*

→ Step 2: Define $V^j$ which interpolates linearly the roll-out cost



# Example I: Constrained LQR

Assumption: A first feasible trajectory at iteration *0* is given

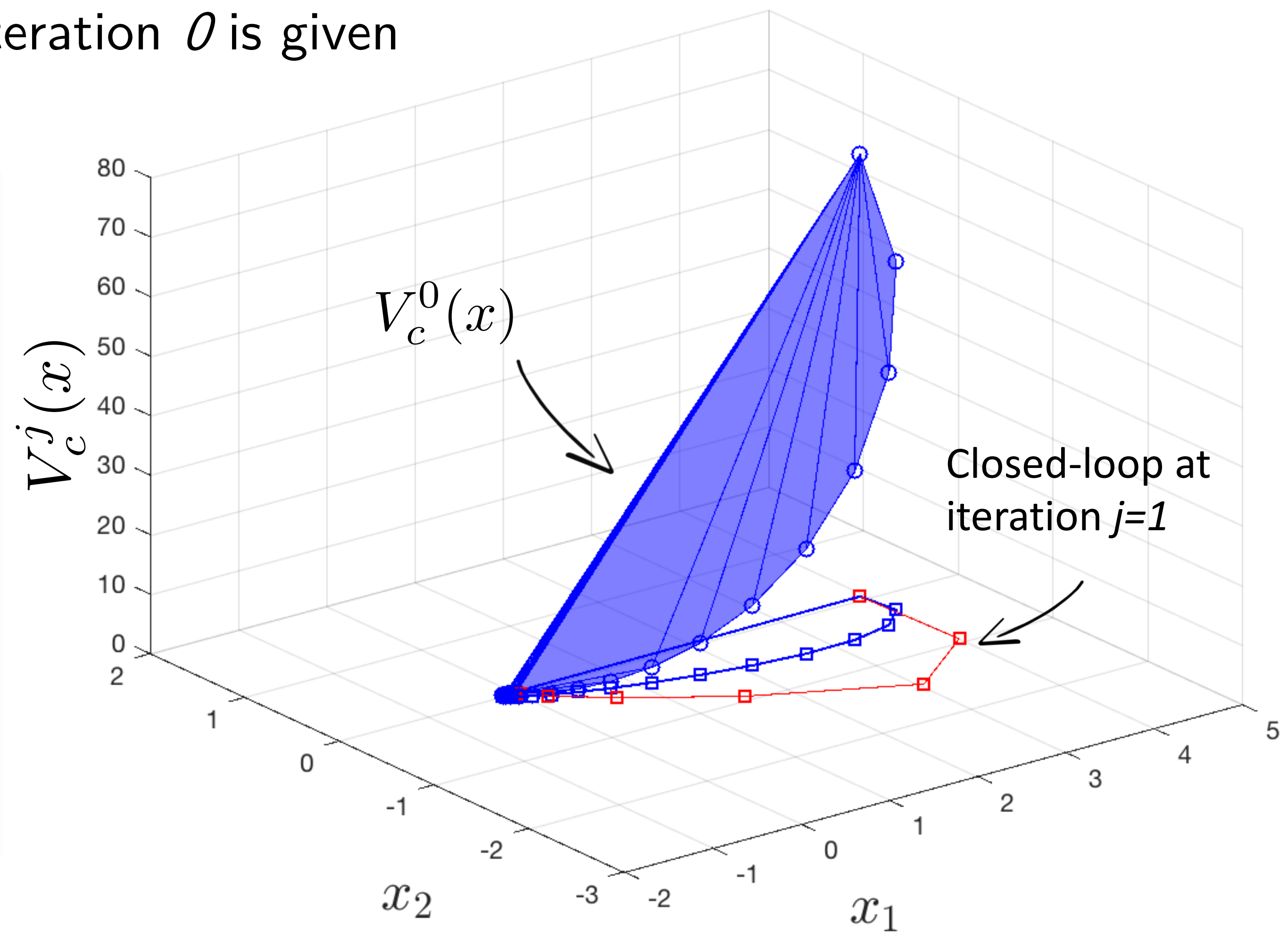
**Iterative LMPC**

Step 0: Set iteration counter *j=0*

Step 1: Compute the roll-out cost for the recorded data up to iteration *j*

Step 2: Define $V^j$ which interpolates linearly the roll-out cost

→ Step 3: Run a closed-loop simulation at iteration *j+1*



# Example I: Constrained LQR

Assumption: A first feasible trajectory at iteration *0* is given

**Iterative LMPC**

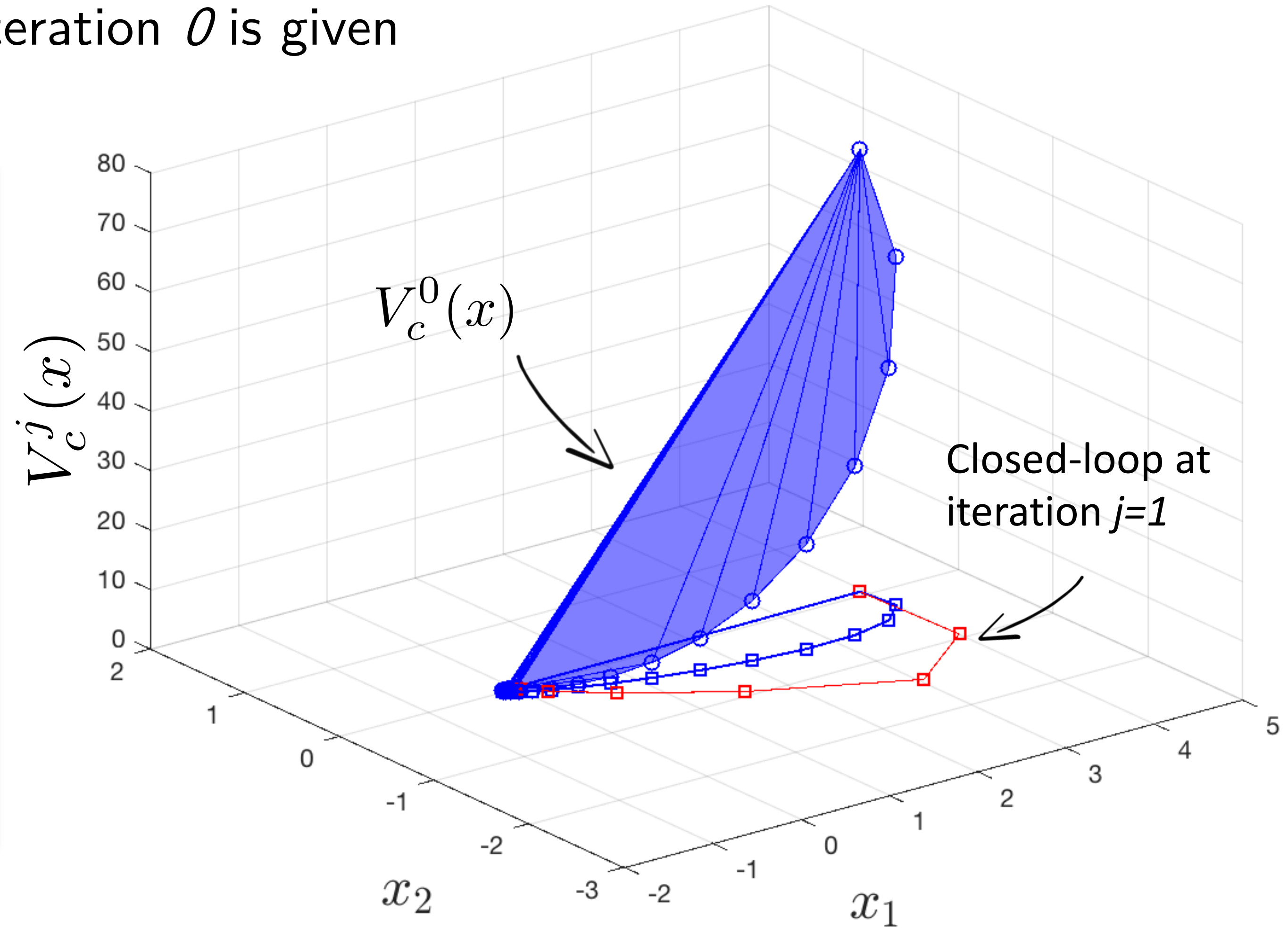
Step 0: Set iteration counter *j=0*

Step 1: Compute the roll-out cost for the recorded data up to iteration *j*

Step 2: Define $V^j$ which interpolates linearly the roll-out cost

Step 3: Run a closed-loop simulation at iteration *j+1*

→ Step 5: Set iteration counter *j = j+1.* Go to Step 1



# Example I: Constrained LQR

Assumption: A first feasible trajectory at iteration *0* is given

**Iterative LMPC**

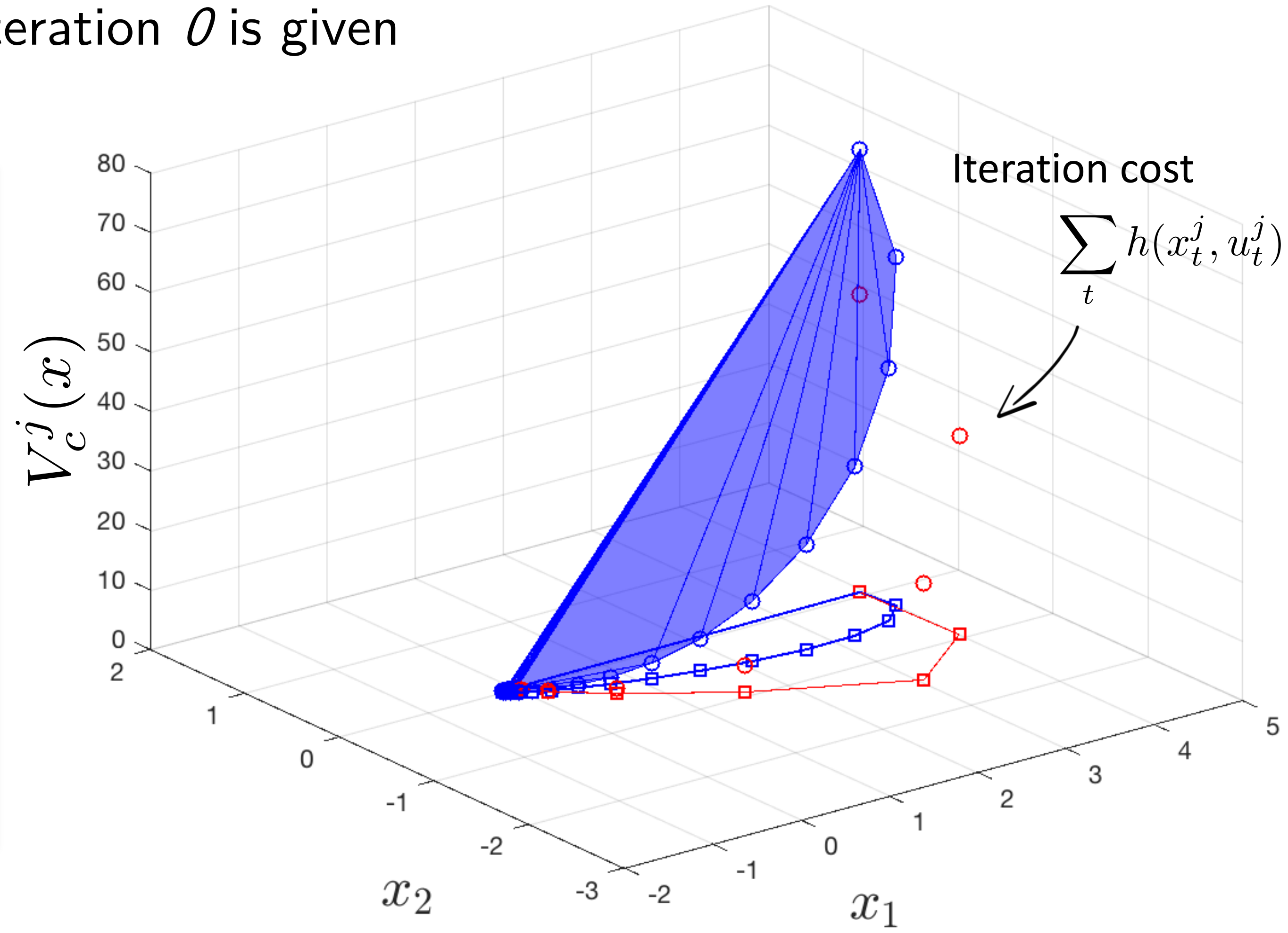
Step 0: Set iteration counter $j=0$

→ Step 1: Compute the roll-out cost for the recorded data up to iteration $j$

Step 2: Define $V^j$ which interpolates linearly the roll-out cost

Step 3: Run a closed-loop simulation at iteration $j+1$

Step 5: Set iteration counter $j = j+1$. Go to Step 1



# Example I: Constrained LQR

Assumption: A first feasible trajectory at iteration *0* is given

**Iterative LMPC**

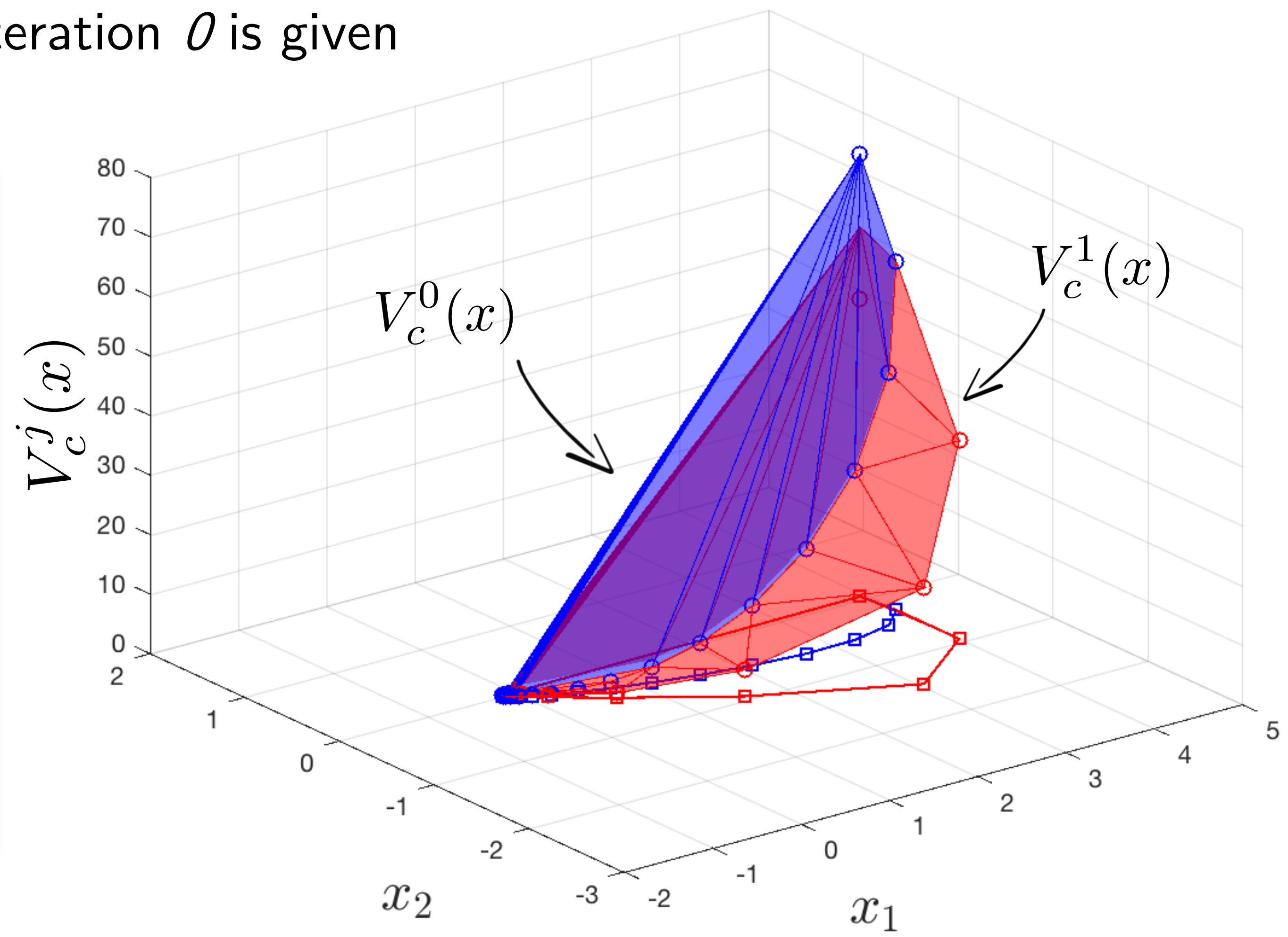
Step 0: Set iteration counter *j=0*

Step 1: Compute the roll-out cost for the recorded data up to iteration *j*

→ Step 2: Define $V^j$ which interpolates linearly the roll-out cost

Step 3: Run a closed-loop simulation at iteration *j+1*

Step 5: Set iteration counter *j = j+1.* Go to Step 1



# Example I: Constrained LQR

Assumption: A first feasible trajectory at iteration *0* is given

**Iterative LMPC**

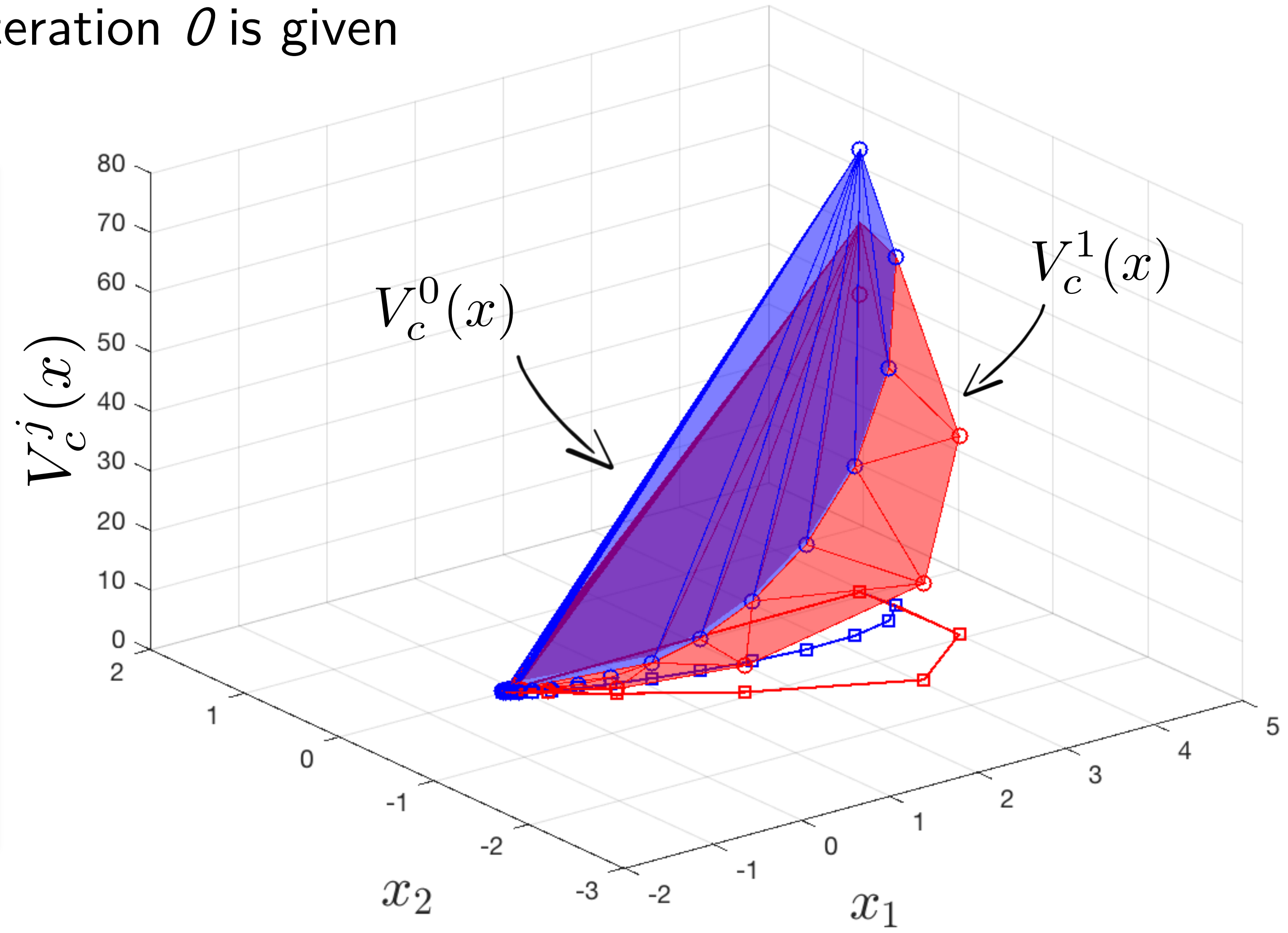
Step 0: Set iteration counter *j=0*

Step 1: Compute the roll-out cost for the recorded data up to iteration *j*

Step 2: Define $V^j$ which interpolates linearly the roll-out cost

Step 3: Run a closed-loop simulation at iteration *j+1*

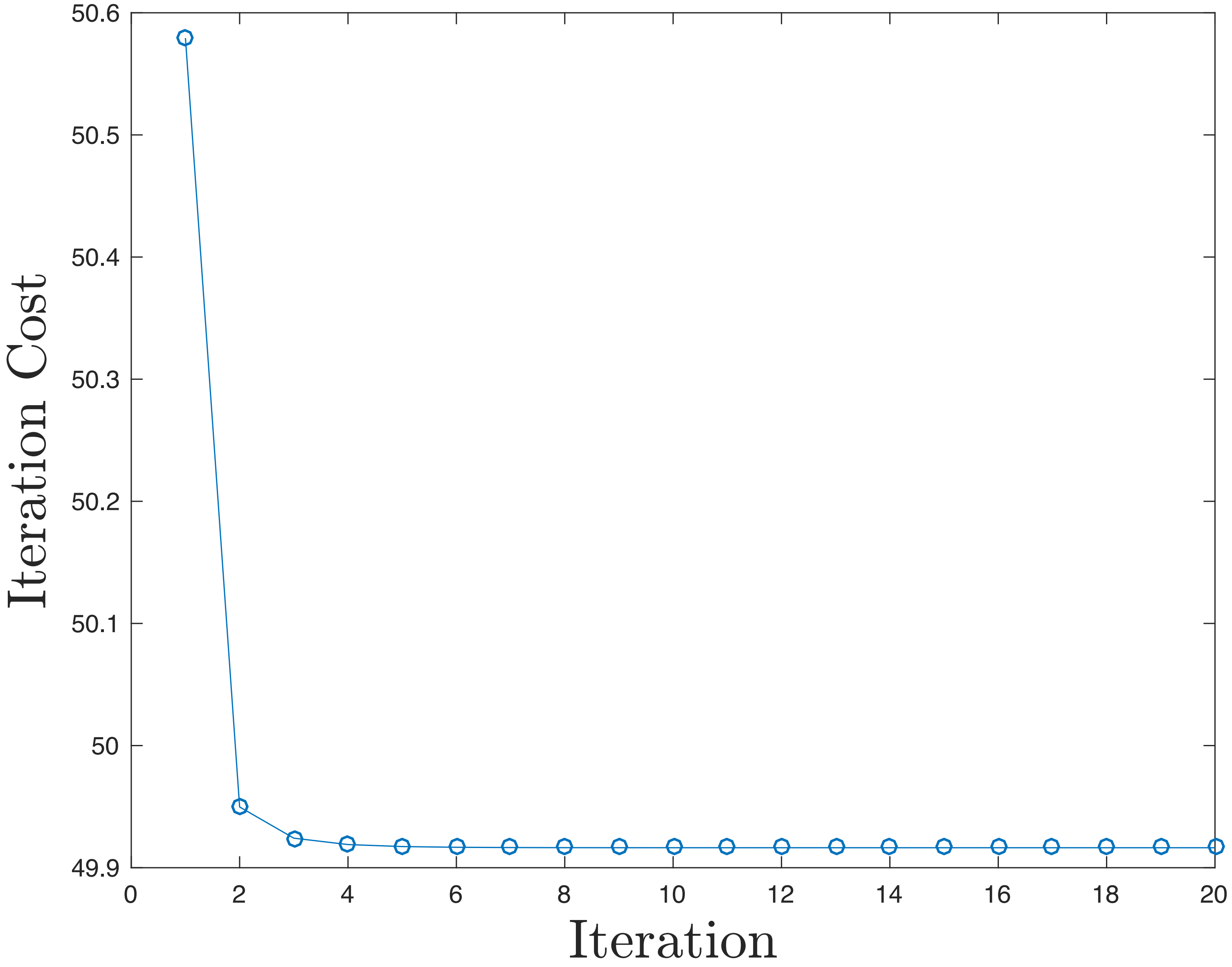
Step 5: Set iteration counter *j = j+1*. Go to Step 1



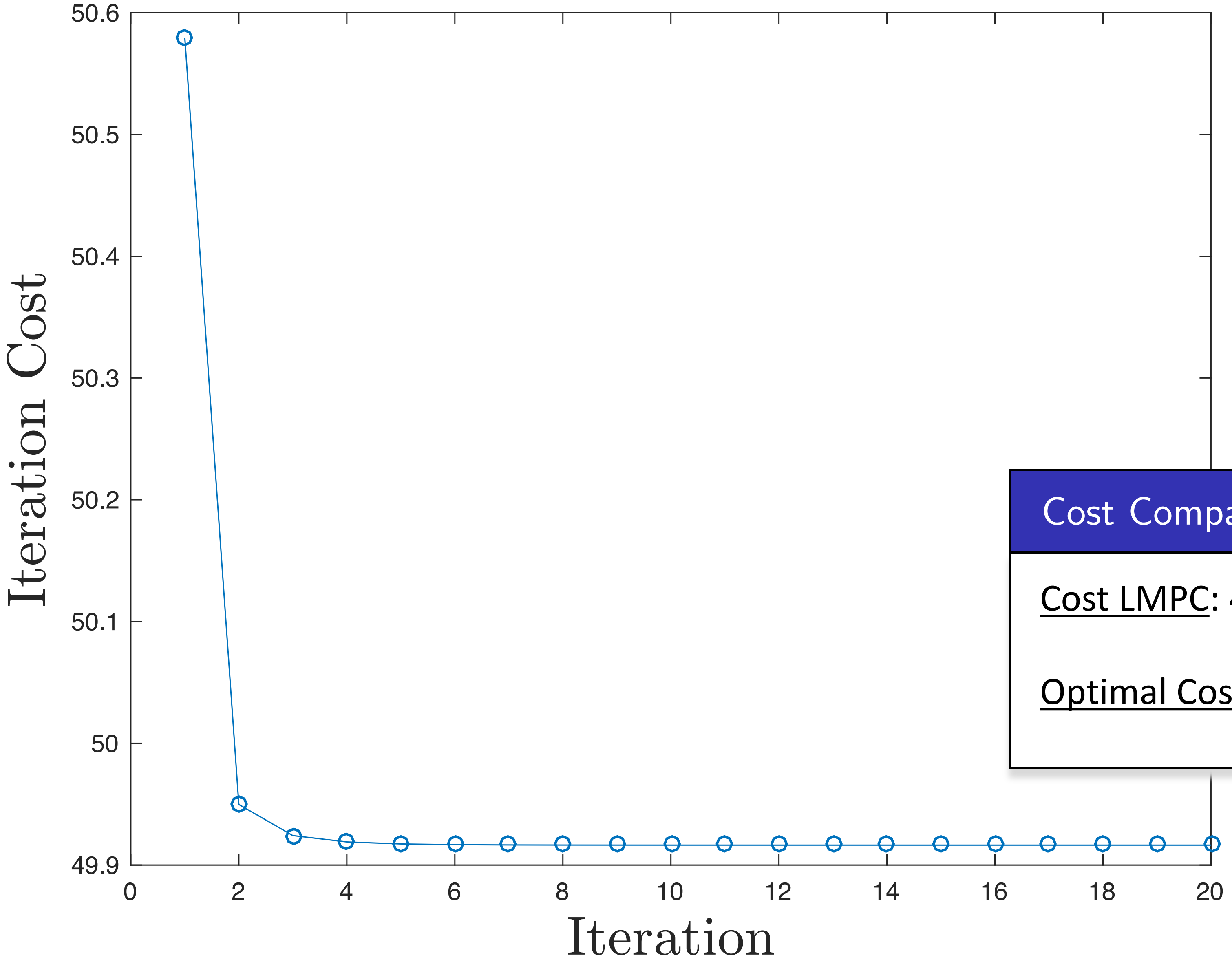

## Key Messages:

- The cost function is defined on a **subset** of the state space.
- The LMPC **explores** the state space in order to enlarge the terminal cost domain.

# Iteration Cost



# Iteration Cost



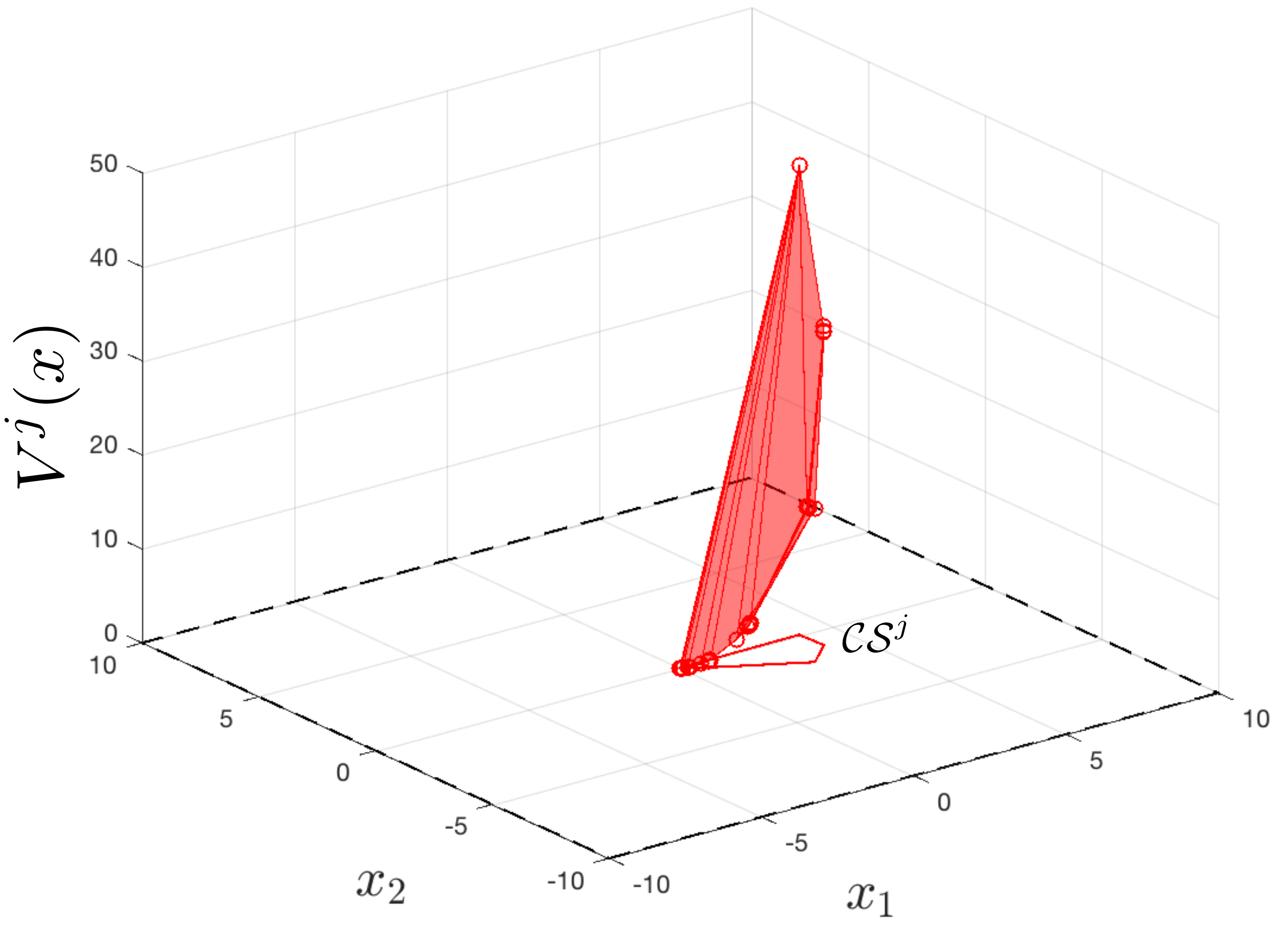

**Cost Comparison**

Cost LMPC: 49.9164

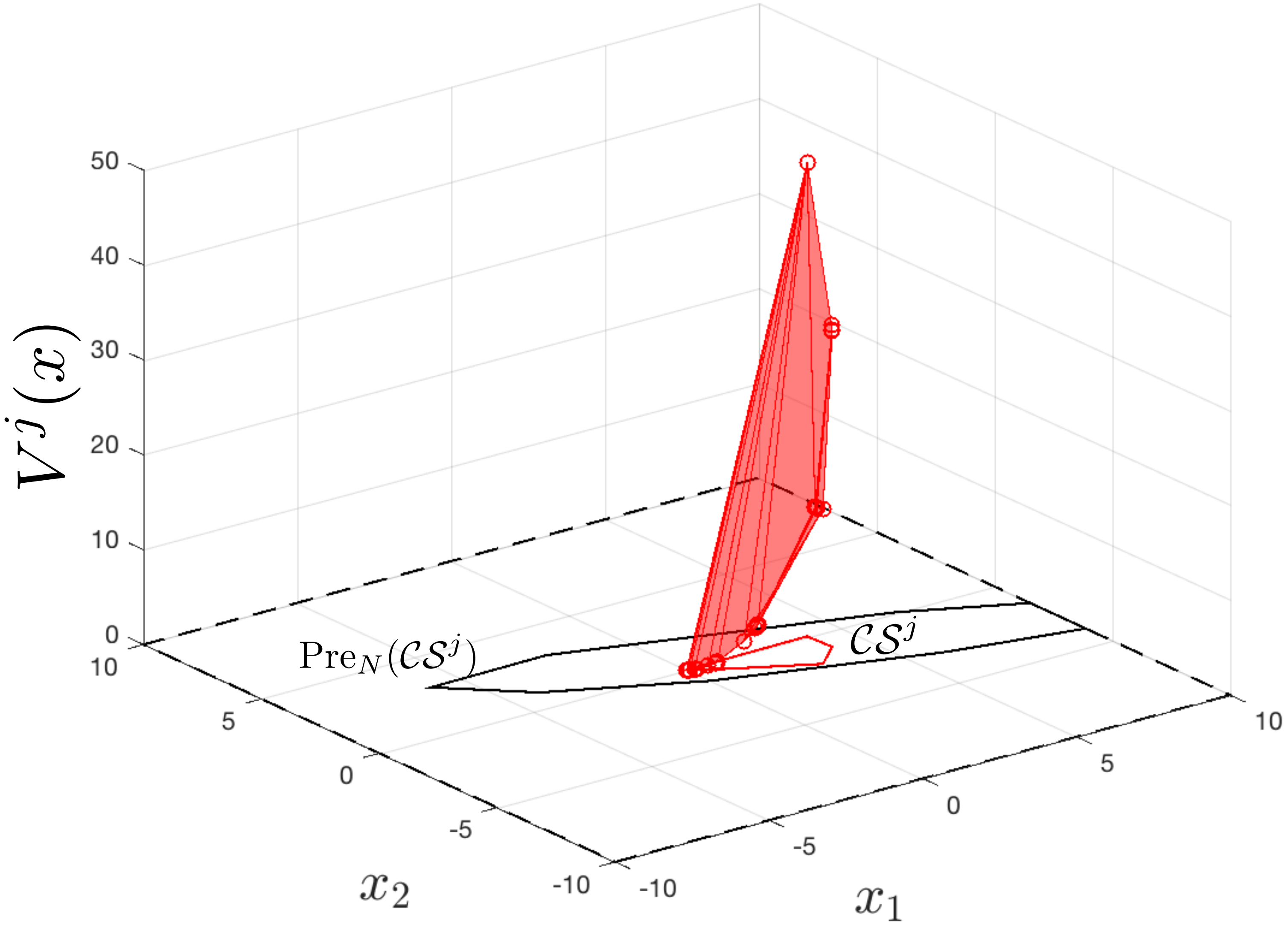
Optimal Cost: 49.9164

# Constrained LQR: LMPC region of attraction

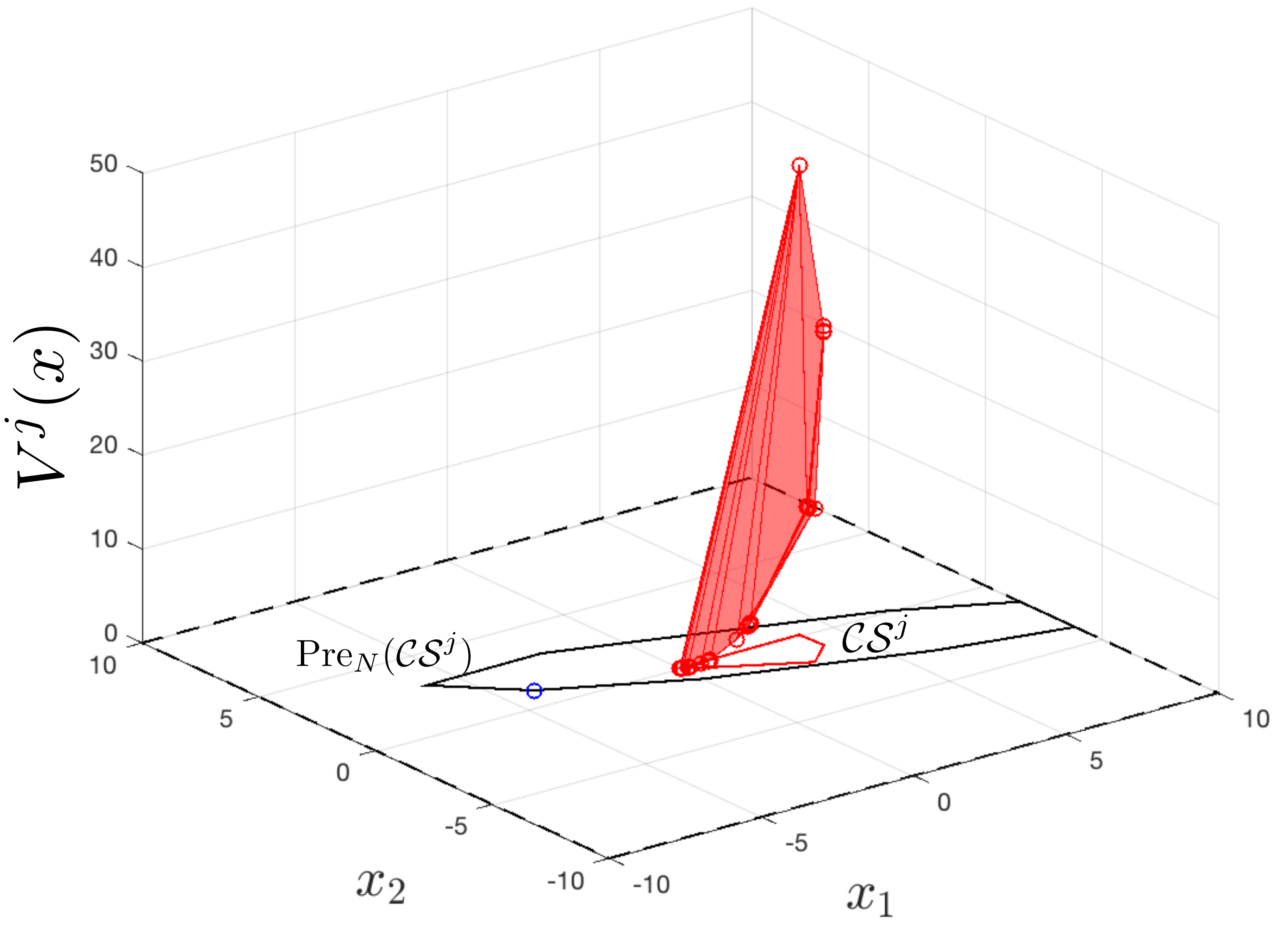
# Constrained LQR: LMPC region of attraction



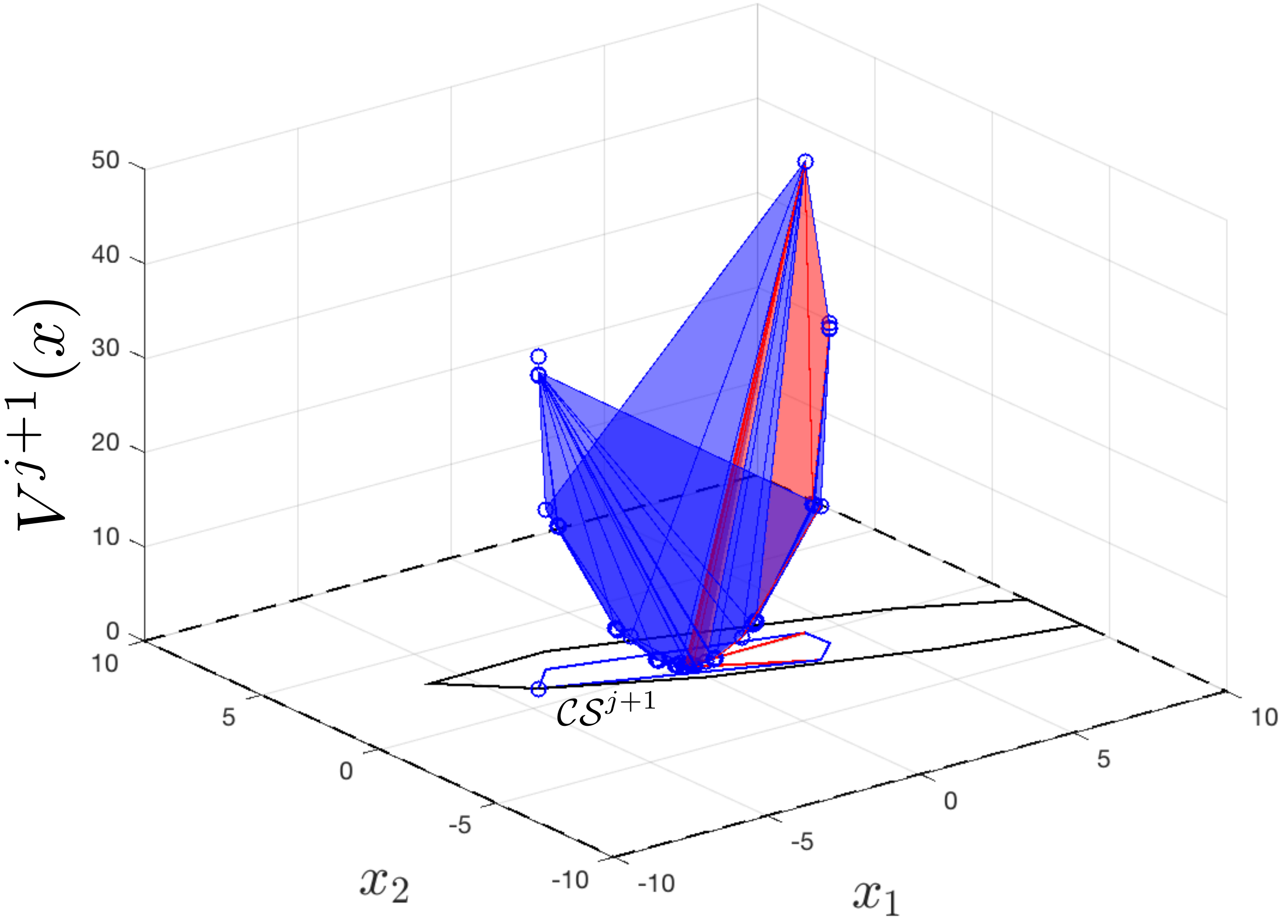
# Constrained LQR: LMPC region of attraction



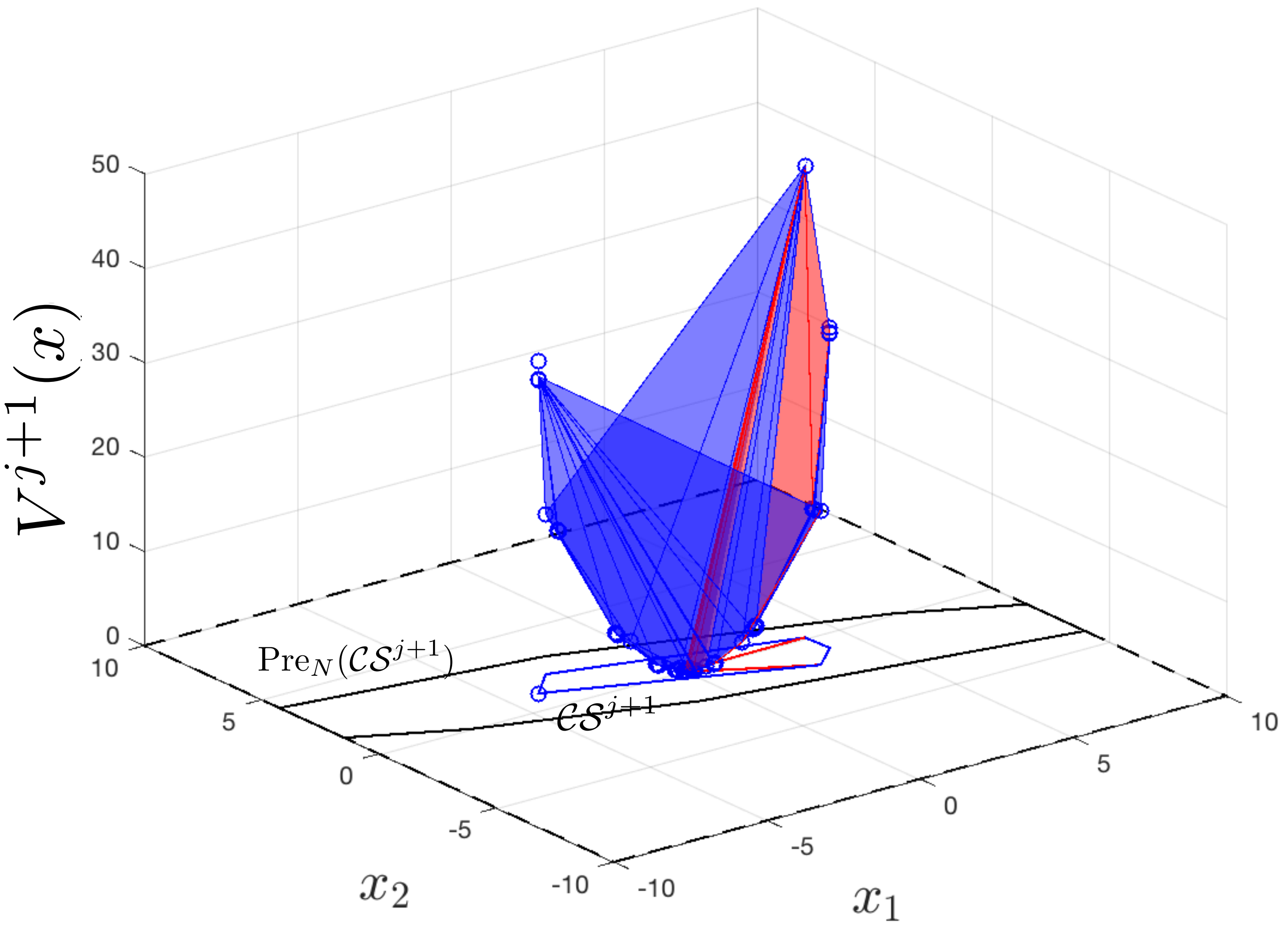
# Constrained LQR: LMPC region of attraction



# Constrained LQR: LMPC region of attraction



# Constrained LQR: LMPC region of attraction



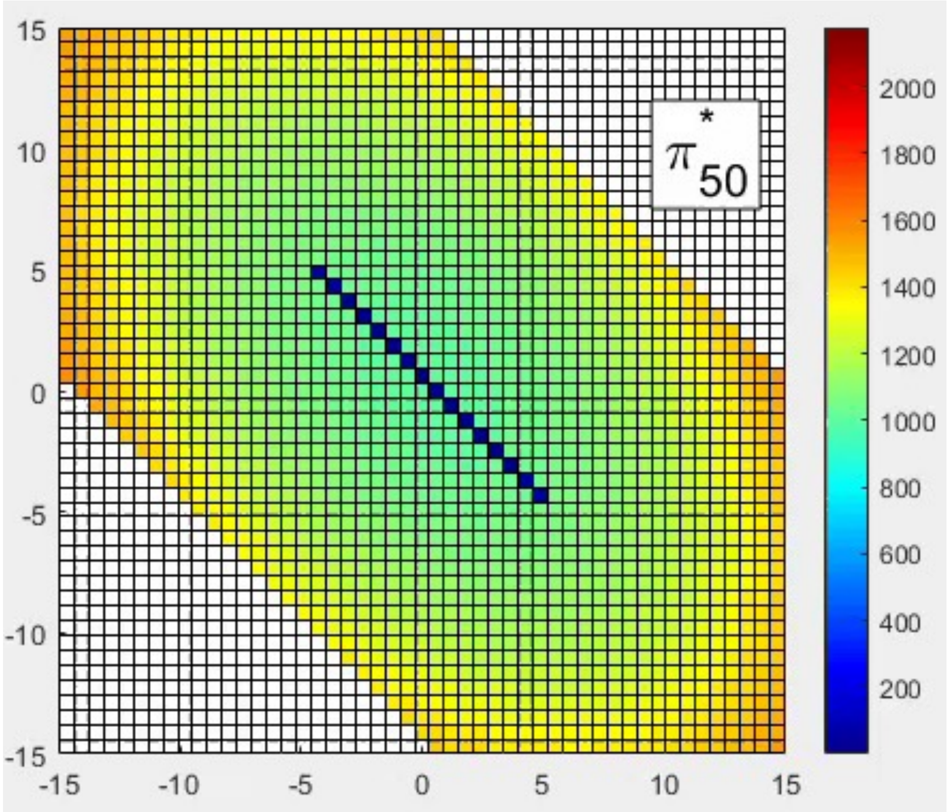
# Comparison with Approximate DP (aka RL)

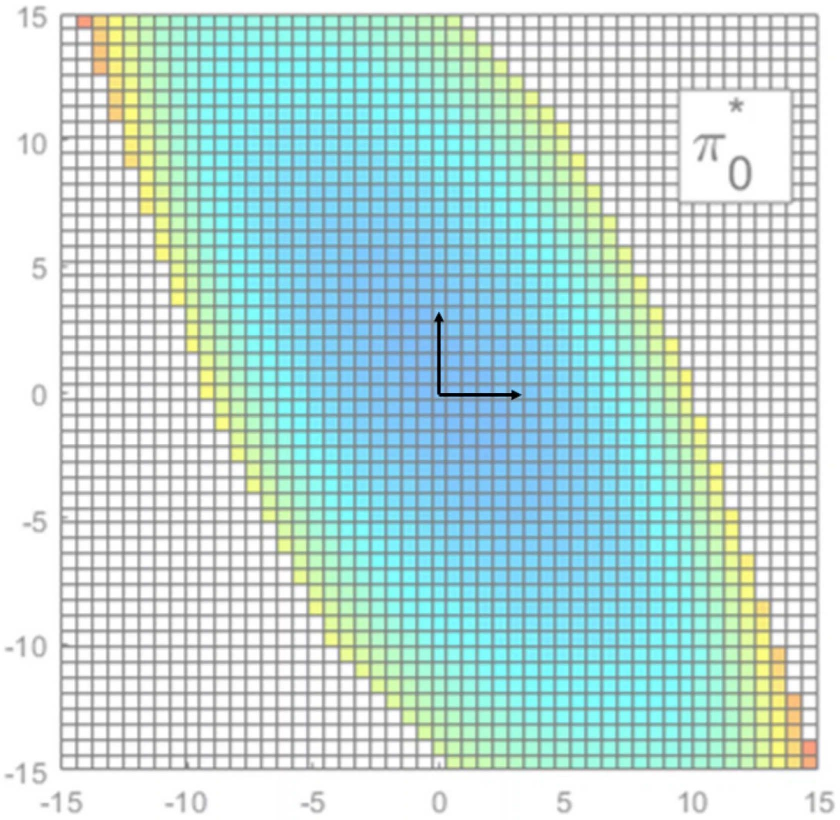
- Some references:
  - Bertsekas paper connecting MPC and ADP [1], books on RL and OC [2,3]
  - Lewis and Vrabie survey [4]
  - Recht survey [5]

- LMPC highlights
  - Continuous state and action formulation
  - Constraints satisfaction and Sampled Safe Sets
  - V-function constructed locally based on cost/model driven exploration
  - V-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

# Outline

- Iterative Control Design for Deterministic Systems
- Autonomous Racing Experiments

# Learning MPC for Autonomous Racing

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

# Problem Formulation

## Minimum Time Control Problem

$$\min_{T,\mathbf{u}} \quad T$$

Control objective

$$x_0 = x_s, \;\; x_T = \mathcal{X}_F$$

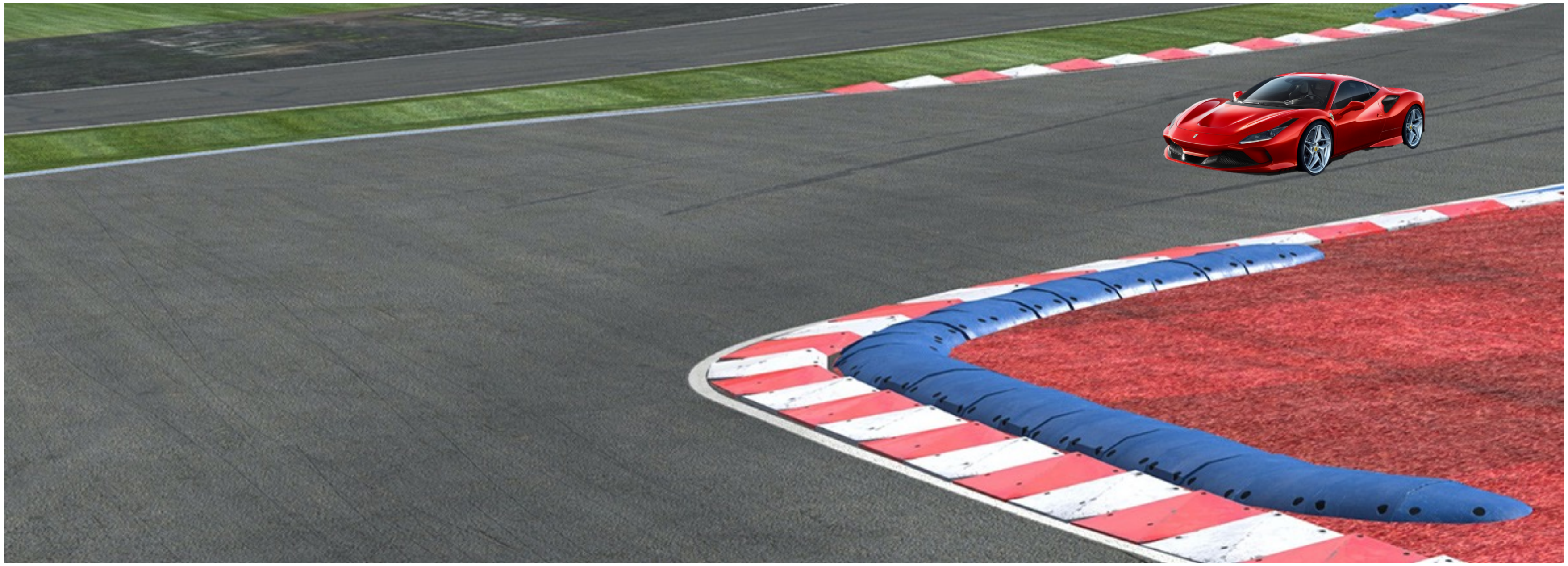
Start & end position

System dynamics
System constraints

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

Safety constraints

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



# Problem Formulation

## Minimum Time Control Problem

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

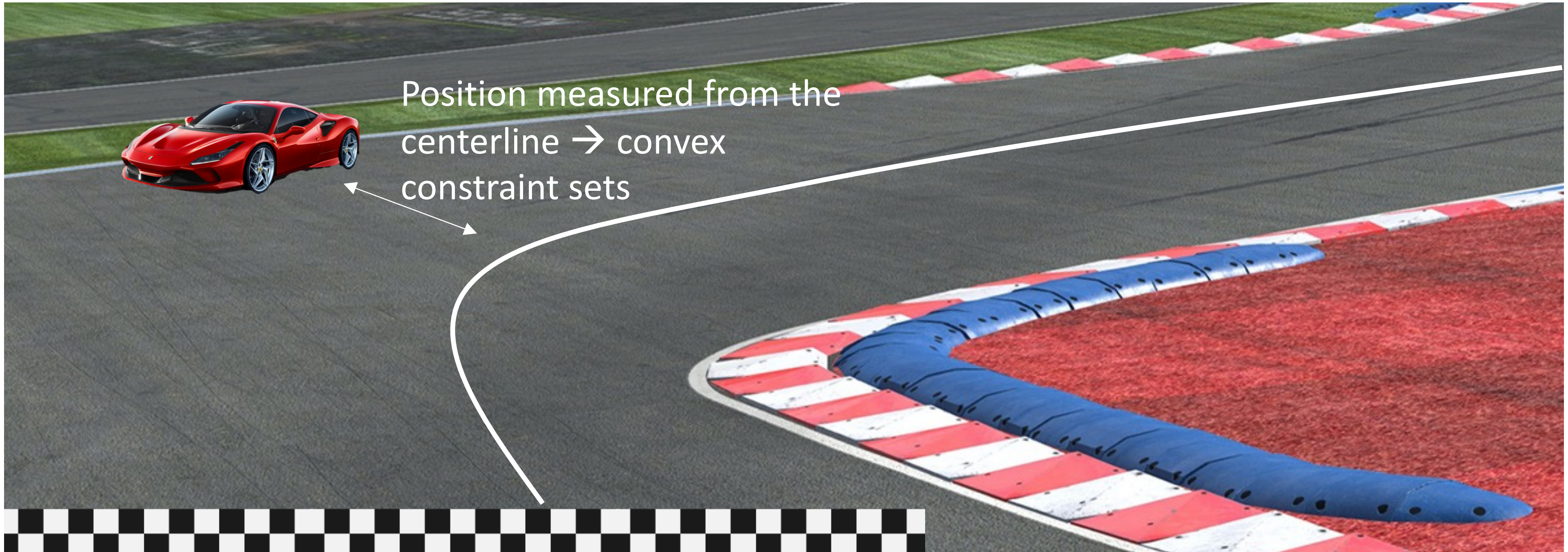
$$\boxed{x_0 = x_s,\ x_T = \mathcal{X}_F}$$ Start & end position

System dynamics
System constraints

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

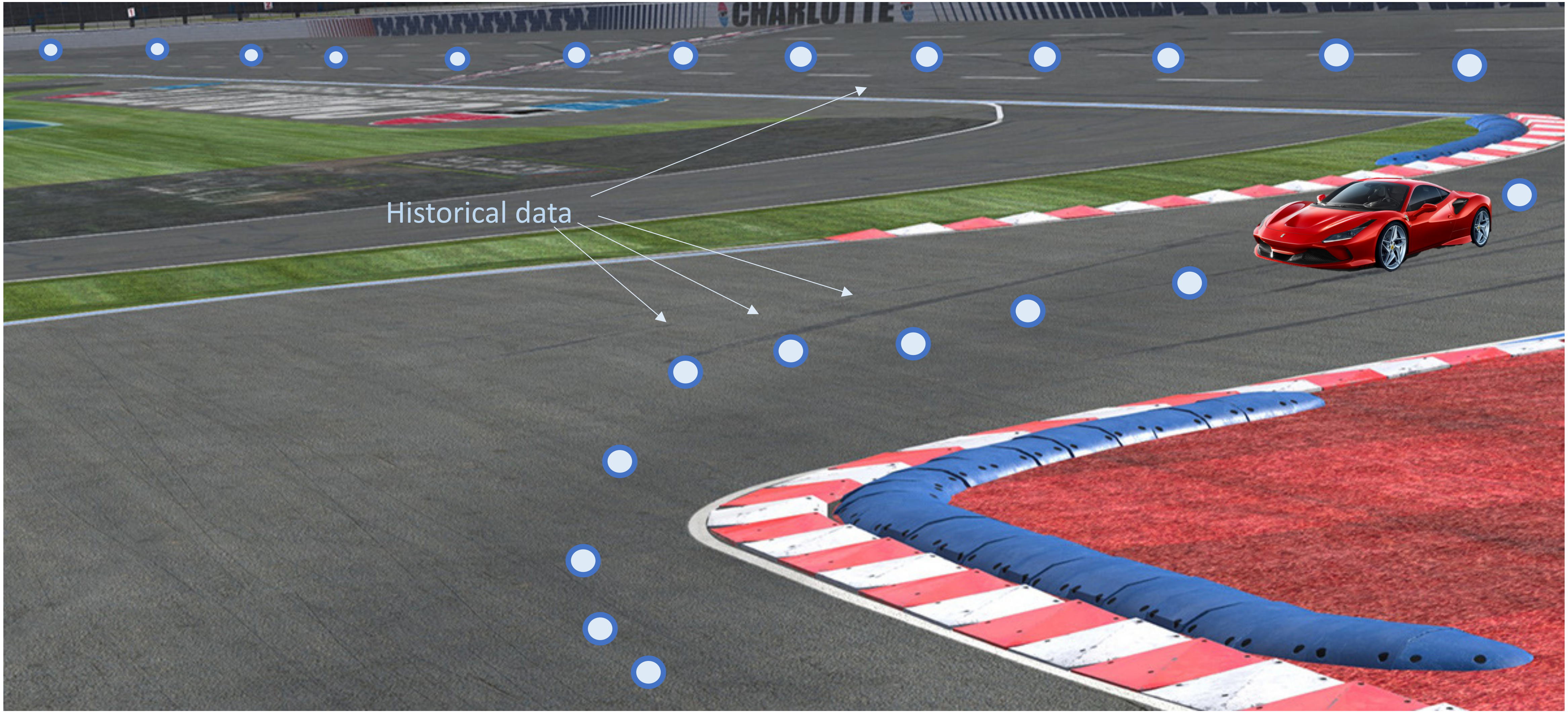
Safety constraints

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



# Key Assumption

We are given a first feasible trajectory and/or controller



# Learning Model Predictive Controller

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

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

s.t.

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

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

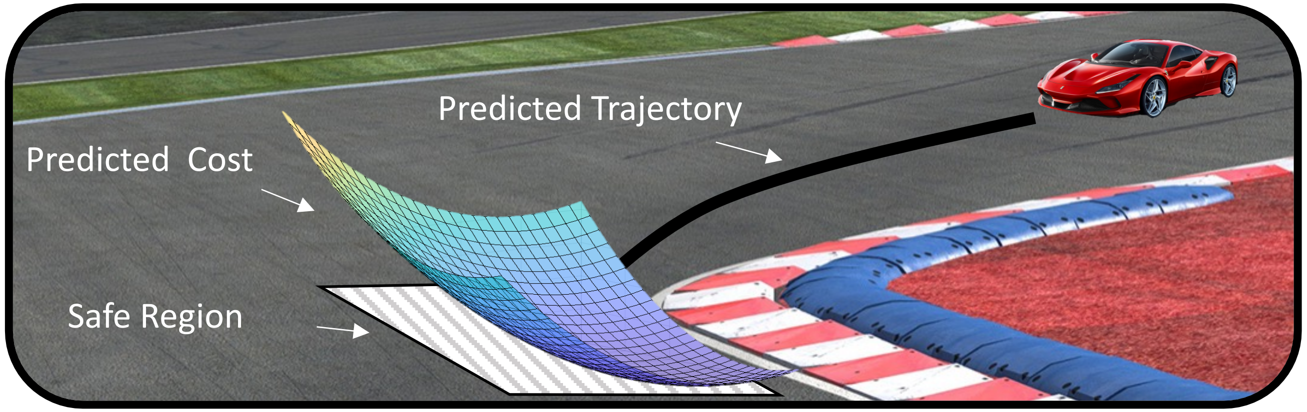
$$x^j_{k|t} \in \mathcal{X},\; u^j_{k|t} \in \mathcal{U},\; \forall k \in [t, \cdots, t+N-1]$$

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

Value Function

Prediction Model

Safe Set



# Learning Model Predictive Controller

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

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

s.t.

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

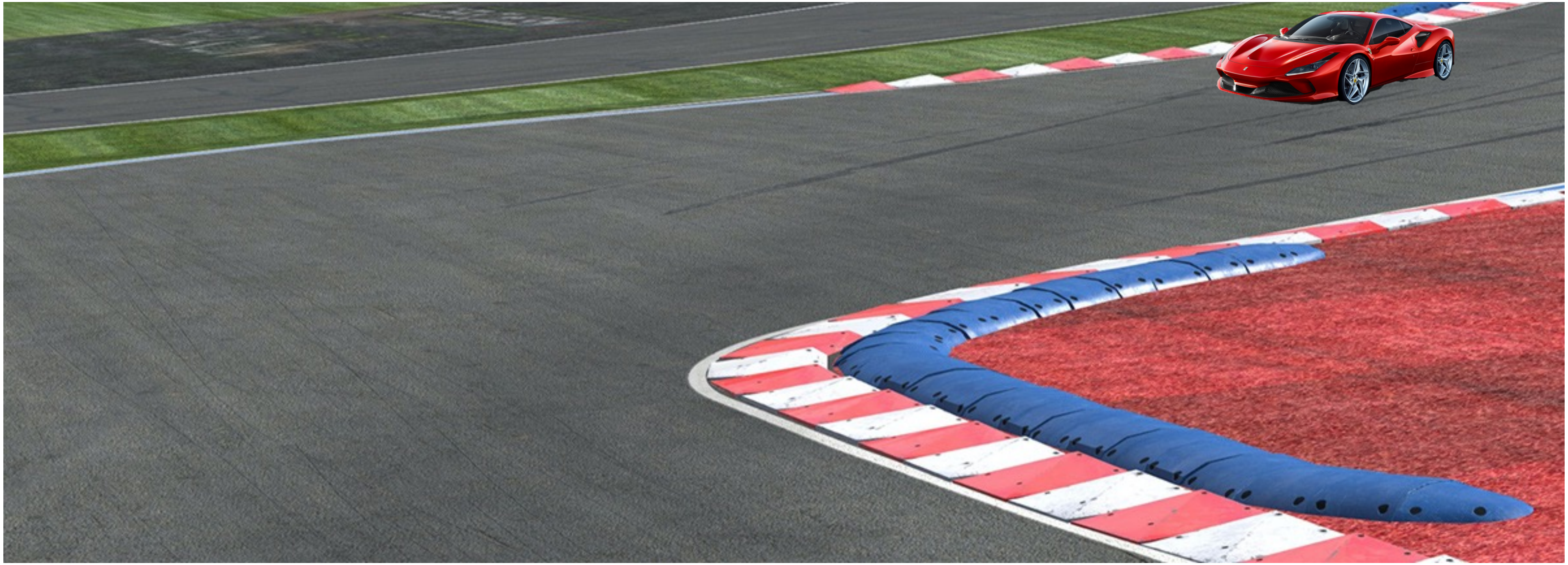
$$x^j_{t|t} = x^j_t,$$

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

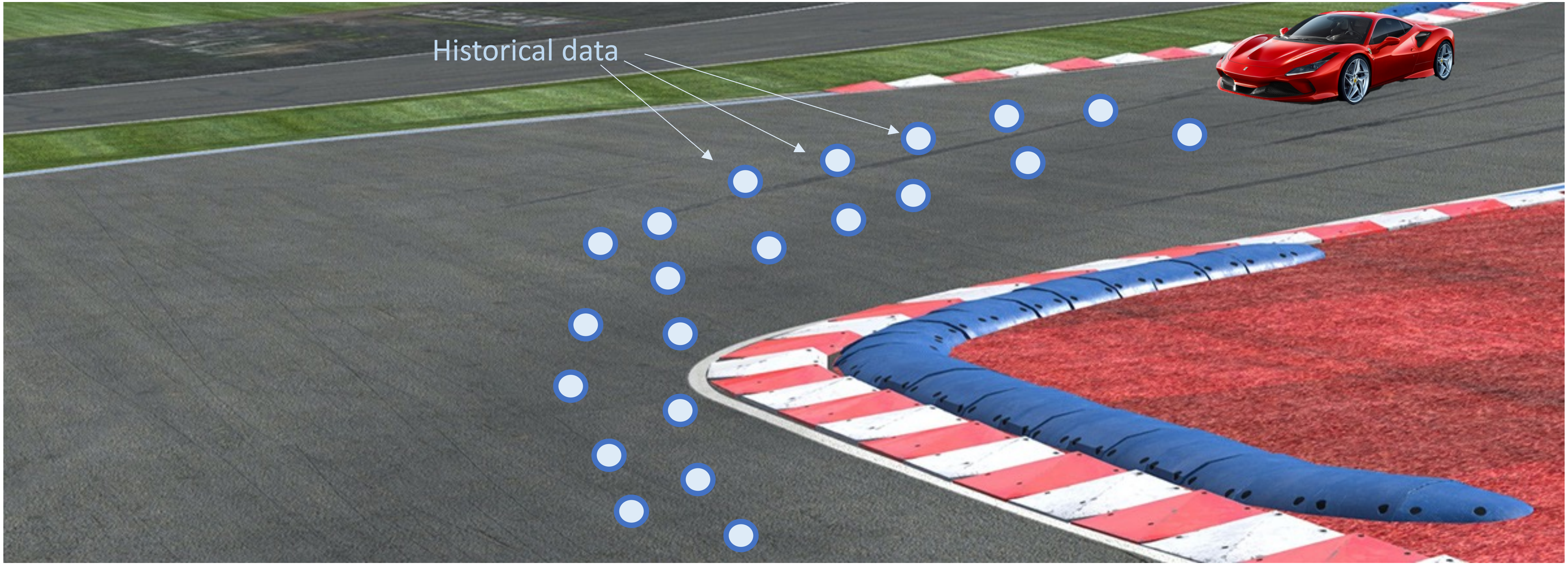
$$x^j_{t+N|t} \in \mathcal{CS}^{j-1}({\color{red}x}),$$

**Safe Set**

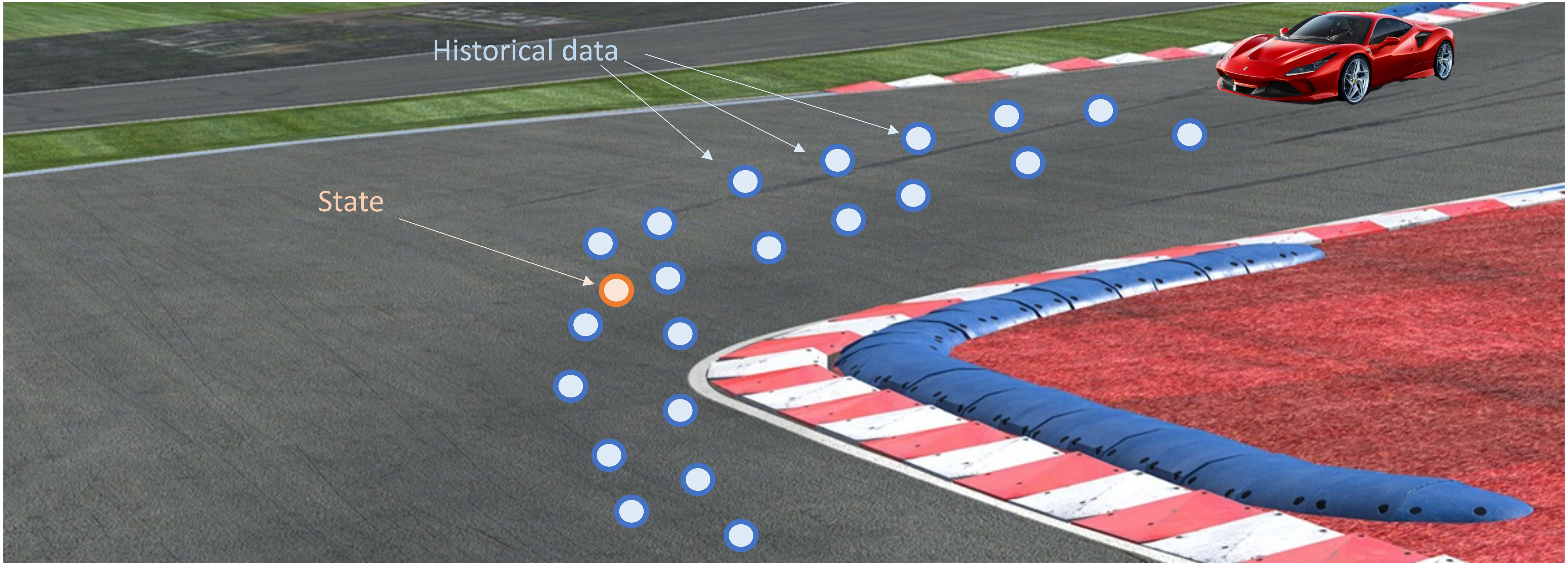
# Safe Set Local Approximations



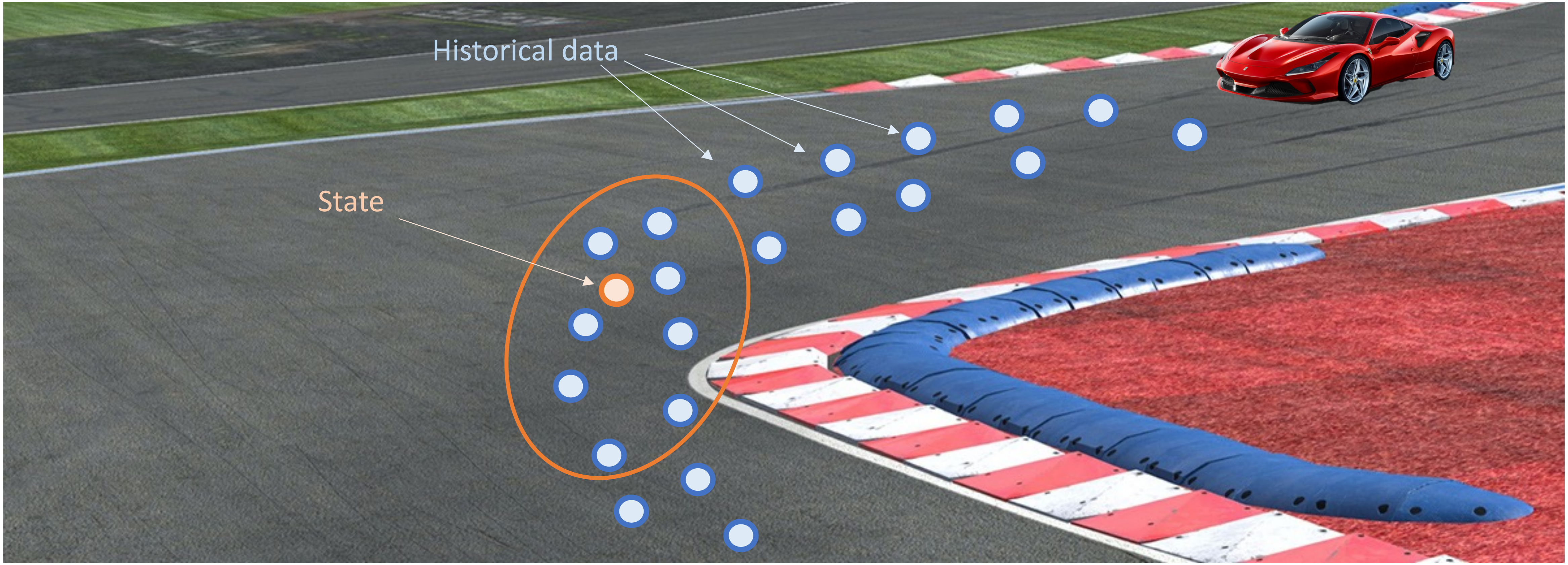
# Safe Set Local Approximations



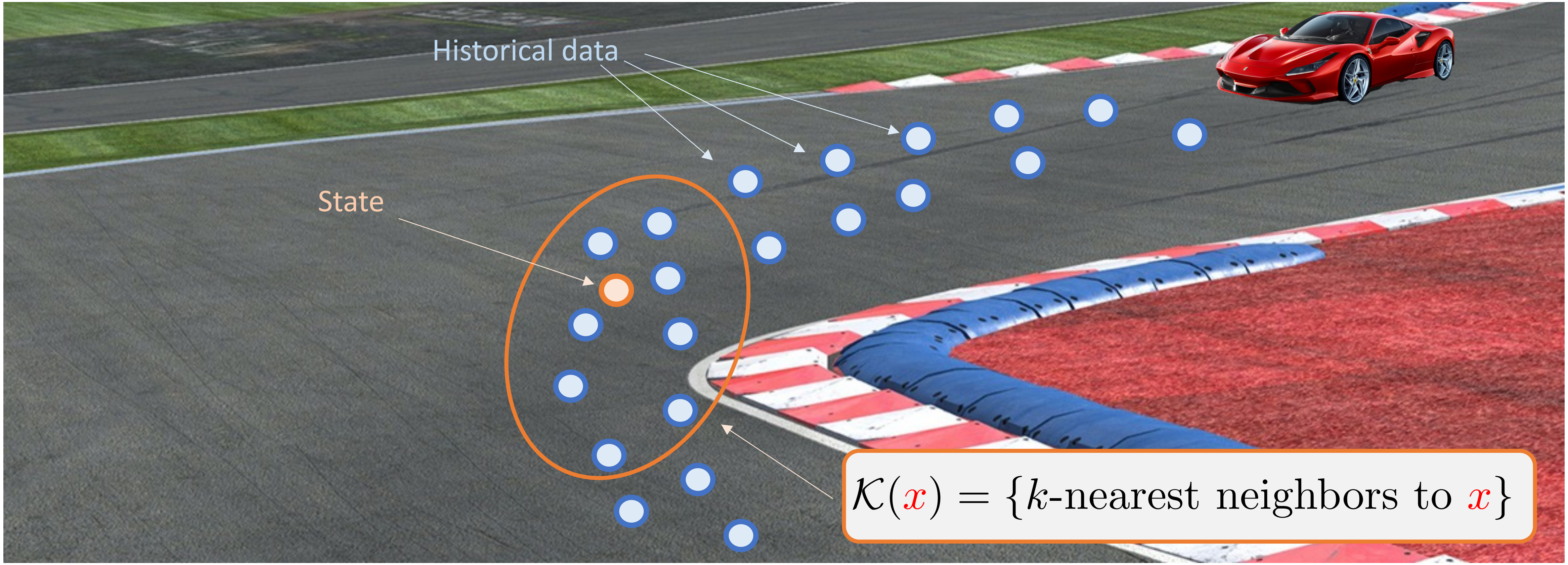
# Safe Set Local Approximations



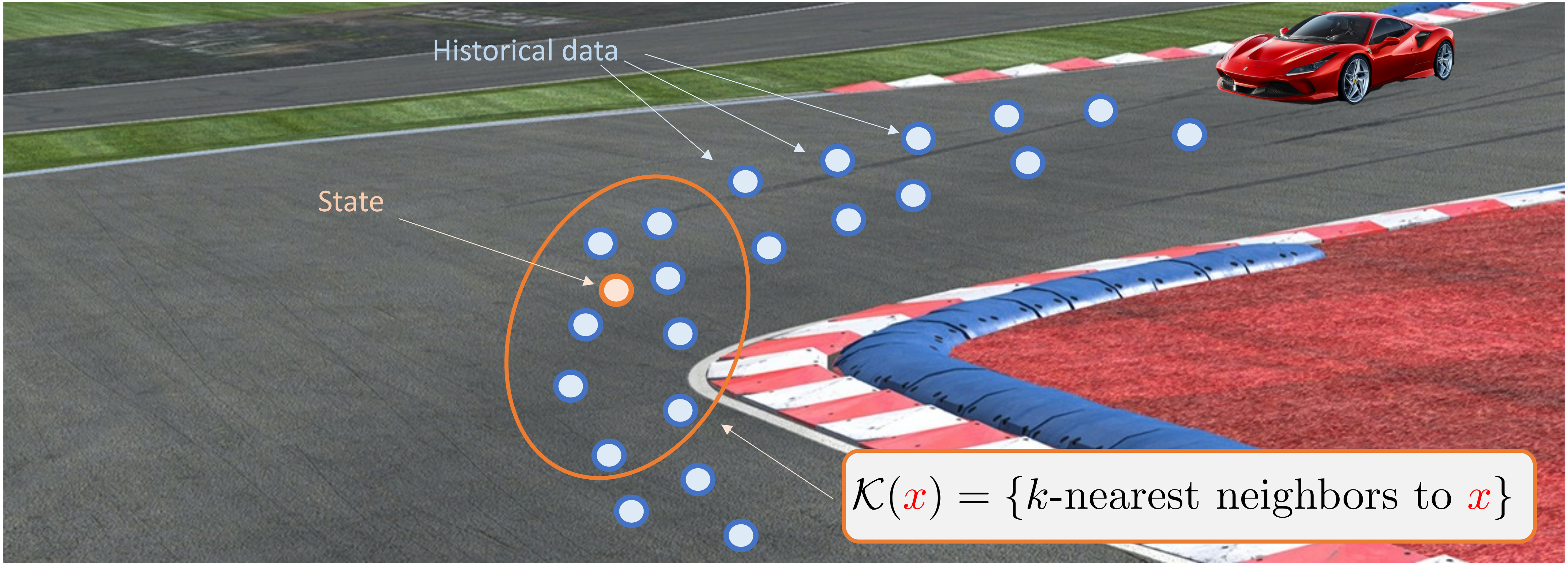
# Safe Set Local Approximations



# Safe Set Local Approximations



# Safe Set Local Approximations




Local convex safe set approximation:

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

# Learning Model Predictive Controller

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

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

s.t.

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

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

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

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

**Safe Set**

where $x = g$(Previous Optimal Trajectory)

# Learning Model Predictive Controller

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

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

s.t.

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

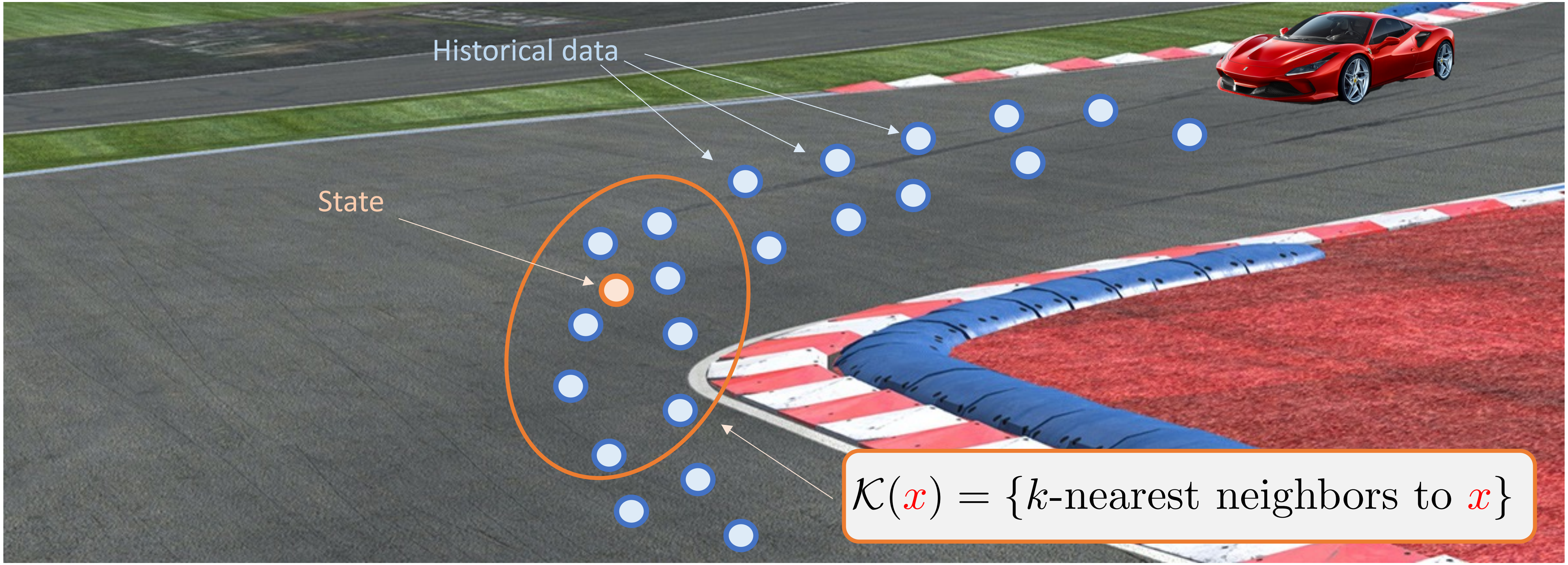
$$x^j_{t|t} = x^j_t,$$

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

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

Value Function

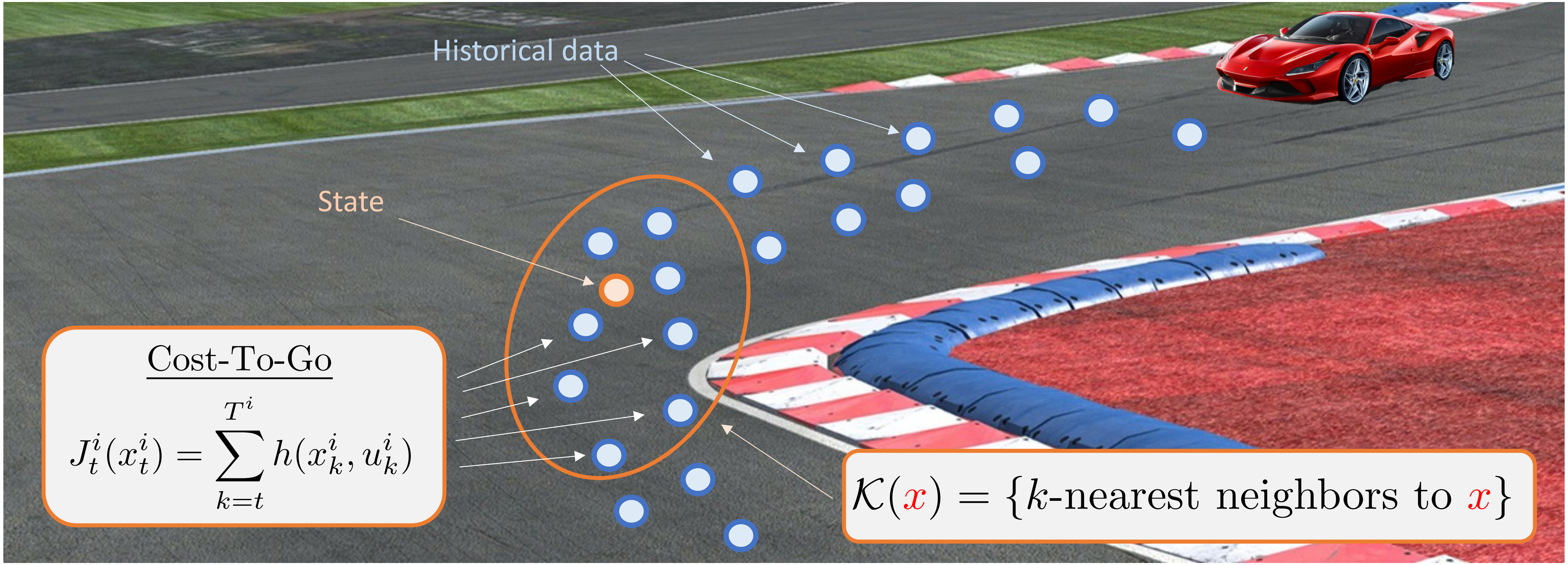
# Value Function Local Approximations




Local convex safe set approximation:

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

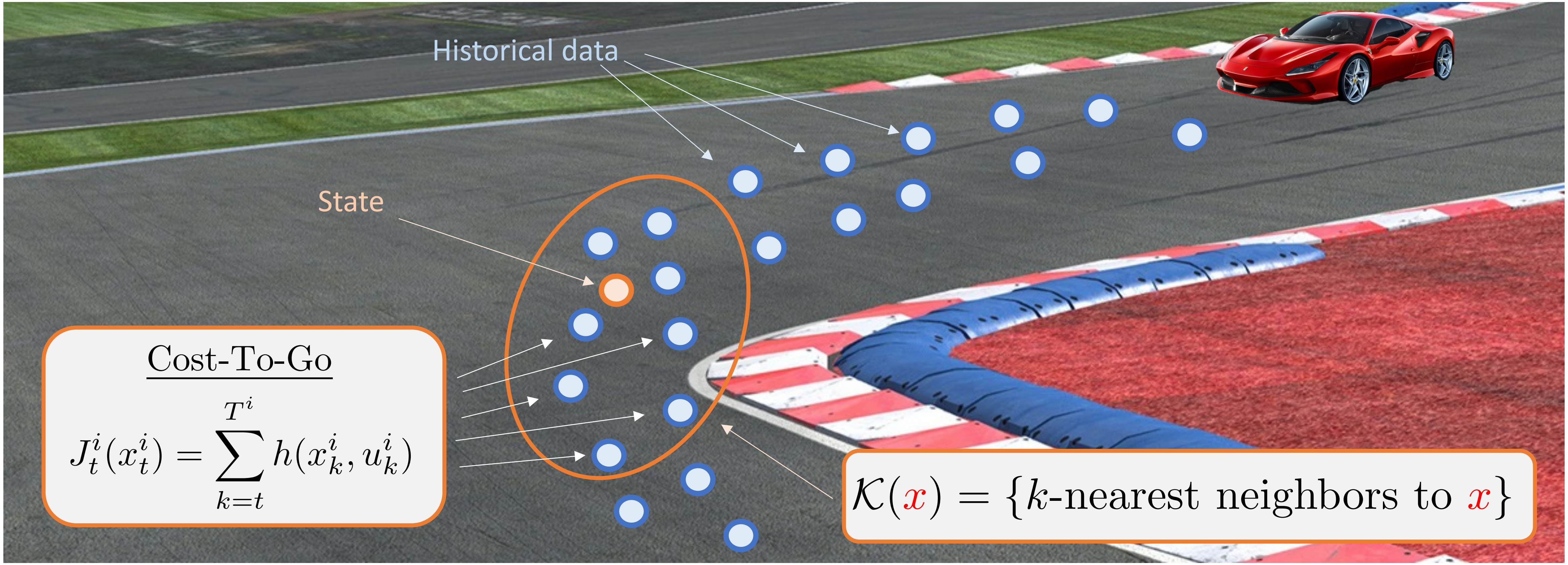
# Value Function Local Approximations




Local convex safe set approximation:

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

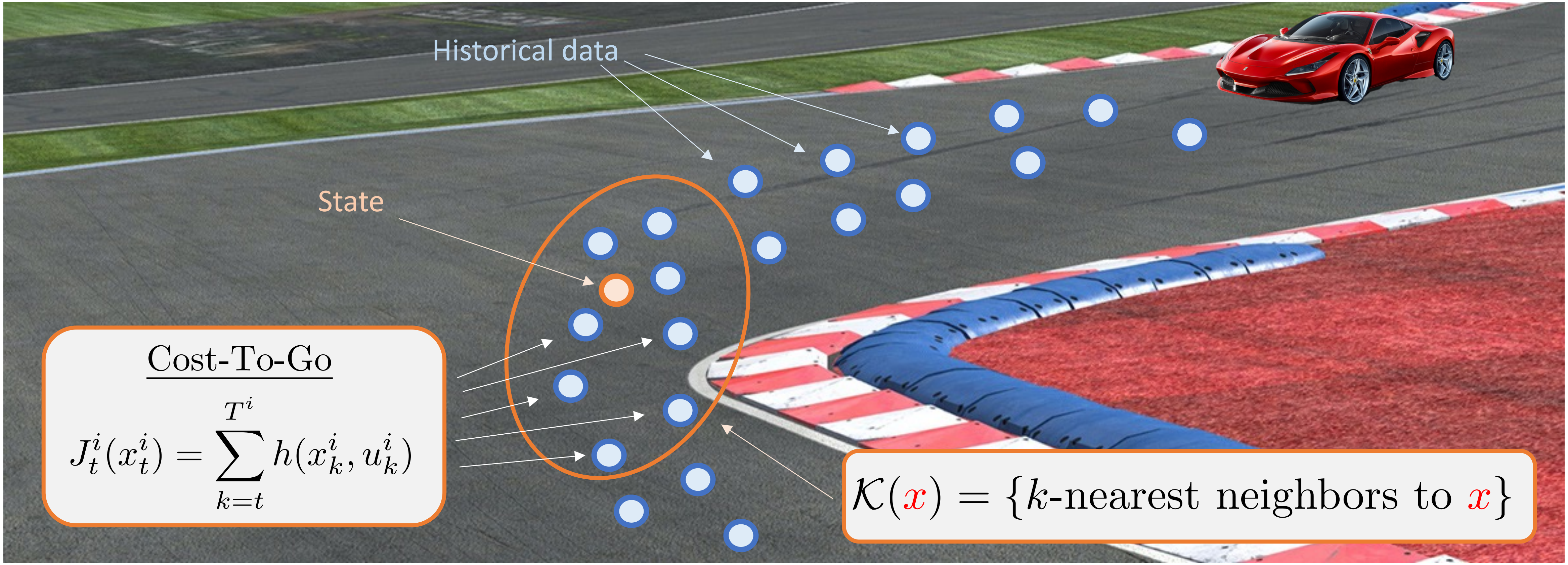
# Value Function Local Approximations




Local value function approximation:

$$V^j(x, x) = \text{ Interpolation of the cost-to-go } J_t^i(x_t^i) = \sum_{k=t}^{T^i} h(x_k^i, u_k^i)$$

# Value Function Local Approximations




Local value function approximation:

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

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

# Learning Model Predictive Controller

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

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

s.t.

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

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

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

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

Prediction Model

# System ID in Autonomous Racing

- Nonlinear Dynamical System,

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

# System ID in Autonomous Racing

- Nonlinear Dynamical System,

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

Kinematic Equations

# System ID in Autonomous Racing

- Nonlinear Dynamical System,

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

Kinematic Equations

- Identifying the Dynamical System

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

Linearization around predicted trajectory

# System ID in Autonomous Racing

- Nonlinear Dynamical System,

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

Dynamic Equations

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

Kinematic Equations

- Identifying the Dynamical System

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

Linearization around predicted trajectory

# System ID in Autonomous Racing

- Nonlinear Dynamical System,

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

Dynamic Equations

Kinematic Equations

- Identifying the Dynamical System

Local Linear Regression

$$
x^j_{k+1|t} = \begin{bmatrix} \dot{x}_{k+1|t} \\ \dot{y}_{k+1|t} \\ \ddot{\psi} \\ \psi_{k+1|t} \\ X_{k+1|t} \\ Y_{k+1|t} \end{bmatrix} = \left[ \begin{array}{c|c} \multicolumn{2}{c}{\operatorname*{argmin}_{\Lambda_y} \sum_{i,s} K(x^j_{k|t} - x^i_s)||\Lambda_y \begin{bmatrix} x^i_s \\ u^i_s \\ 1 \end{bmatrix} - y^i_{s+1}||, \forall y \in \{\dot{x}, \dot{y}, \ddot{\psi}\}} \\ \begin{matrix} \text{Linearized Kinematics} \\ \text{Linearized Kinematics} \\ \text{Linearized Kinematics} \end{matrix} \; x^j_{k|t} + & \begin{matrix} \text{Linearized Kinematics} \\ \text{Linearized Kinematics} \\ \text{Linearized Kinematics} \end{matrix} \end{array} \right] \begin{bmatrix} u^j_{k|t} \\ 1 \end{bmatrix}
$$

Linearization around predicted trajectory

# System Identification – Design Steps

## Identifying the Dynamical System

Local Linear Regression

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

Linearization around predicted trajectory

### Implementation Details

# System Identification – Design Steps

## Identifying the Dynamical System

Local Linear Regression

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

Linearization around predicted trajectory

### 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

$$x^j_{k+1|t} = \begin{bmatrix} \dot{x}_{k+1|t} \\ \dot{y}_{k+1|t} \\ \ddot{\psi} \\ \psi_{k+1|t} \\ X_{k+1|t} \\ Y_{k+1|t} \end{bmatrix} = \begin{bmatrix} \multicolumn{2}{c}{\underset{\Lambda_y}{\operatorname{argmin}} \sum_{i,s} K(x^j_{k|t} - x^i_s)||\Lambda_y \begin{bmatrix} x^i_s \\ u^i_s \\ 1 \end{bmatrix} - y^i_{s+1}||, \forall y \in \{\dot{x}, \dot{y}, \ddot{\psi}\}} \\ \begin{bmatrix} \text{Linearized Kinematics} \\ \text{Linearized Kinematics} \\ \text{Linearized Kinematics} \end{bmatrix} x^j_{k|t} + & \begin{bmatrix} \text{Linearized Kinematics} \\ \text{Linearized Kinematics} \\ \text{Linearized Kinematics} \end{bmatrix} \end{bmatrix} \begin{bmatrix} u^j_{k|t} \\ 1 \end{bmatrix}$$

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

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

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

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

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

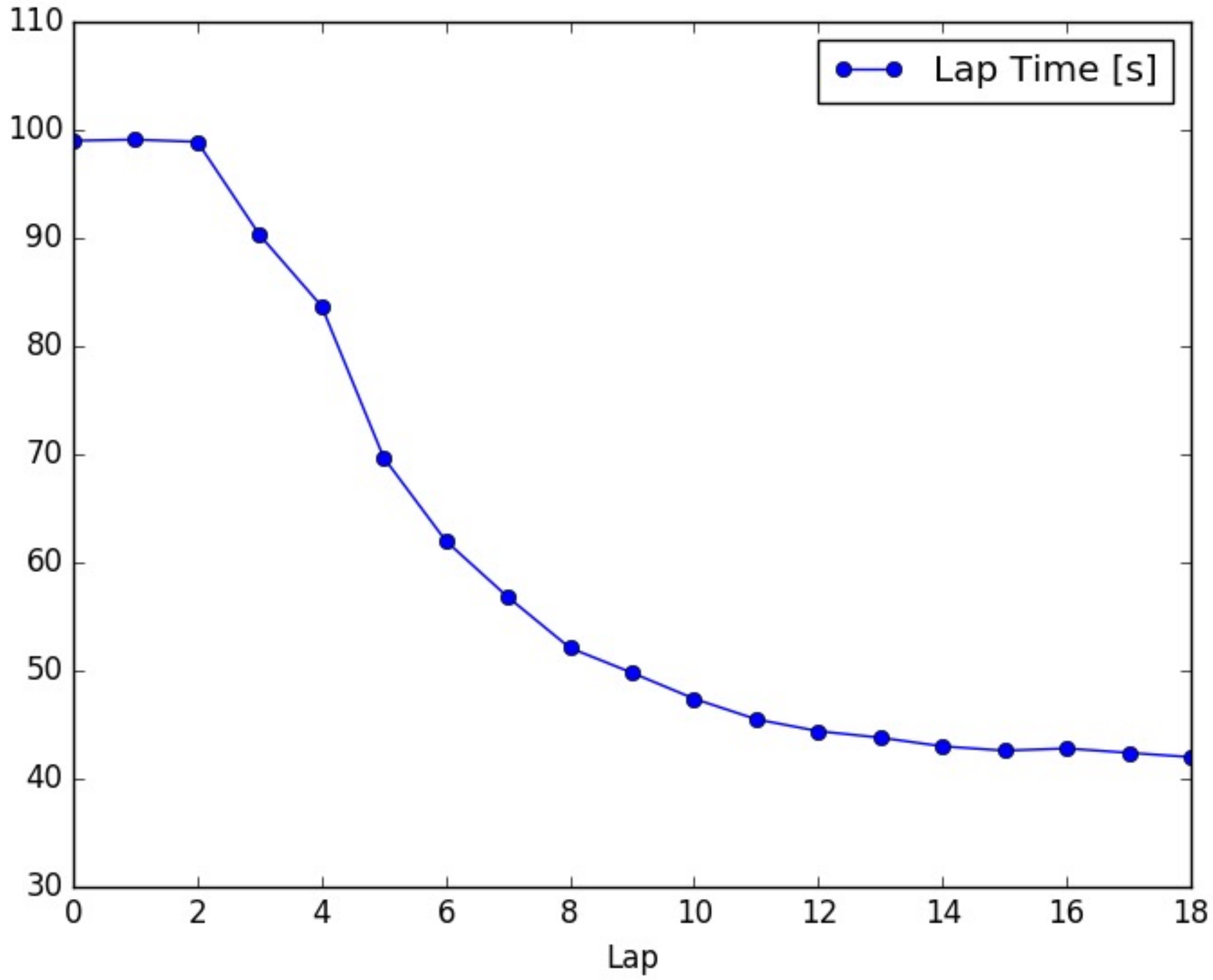
# Hyundai California Proving Ground

# Hyundai California Proving Ground

# Learning Model Predictive Controller full-size vehicle experiments

Credits: Siddharth Nair, Nitin Kapania and Ugo Rosolia

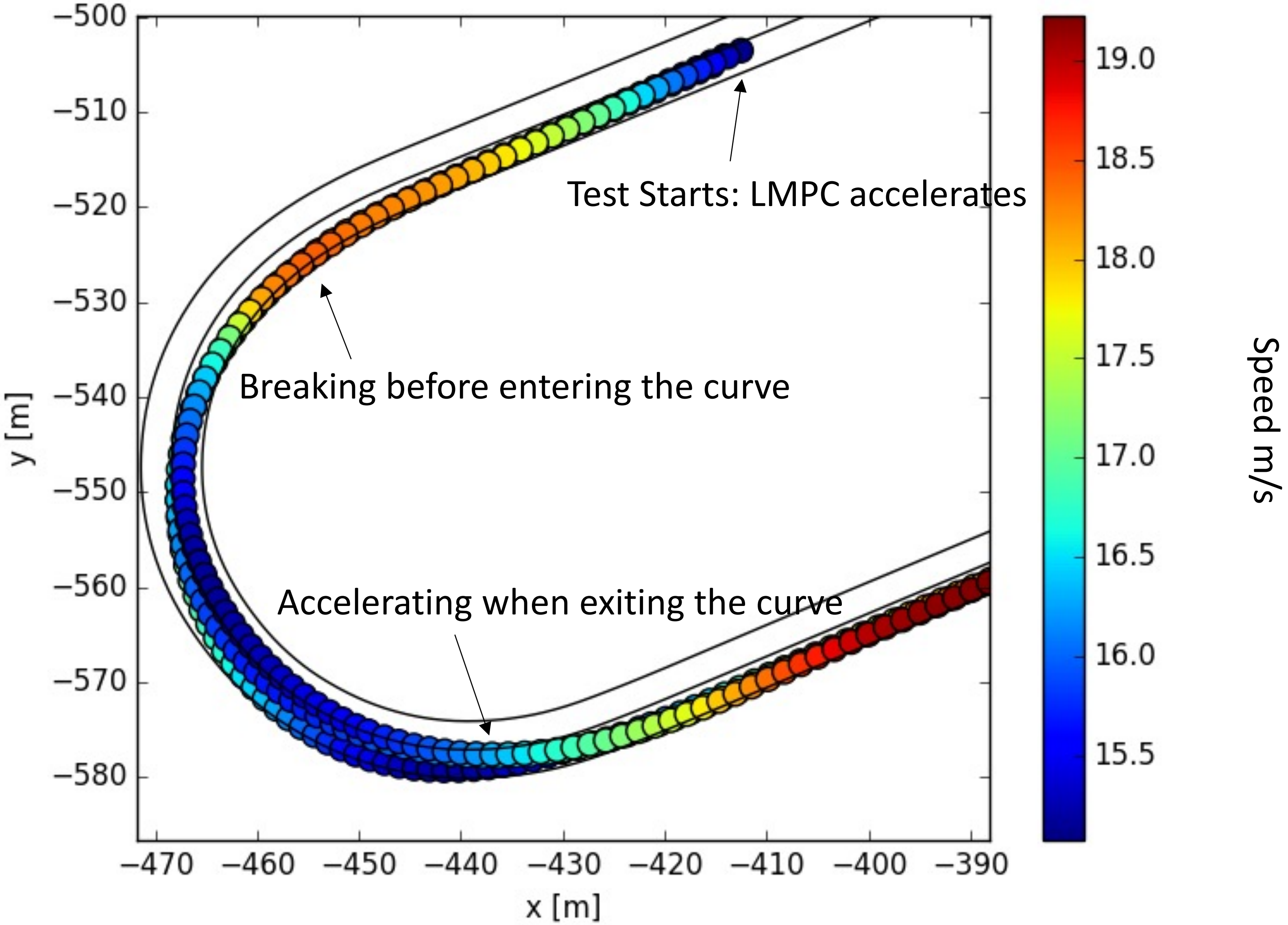
# Lap Time



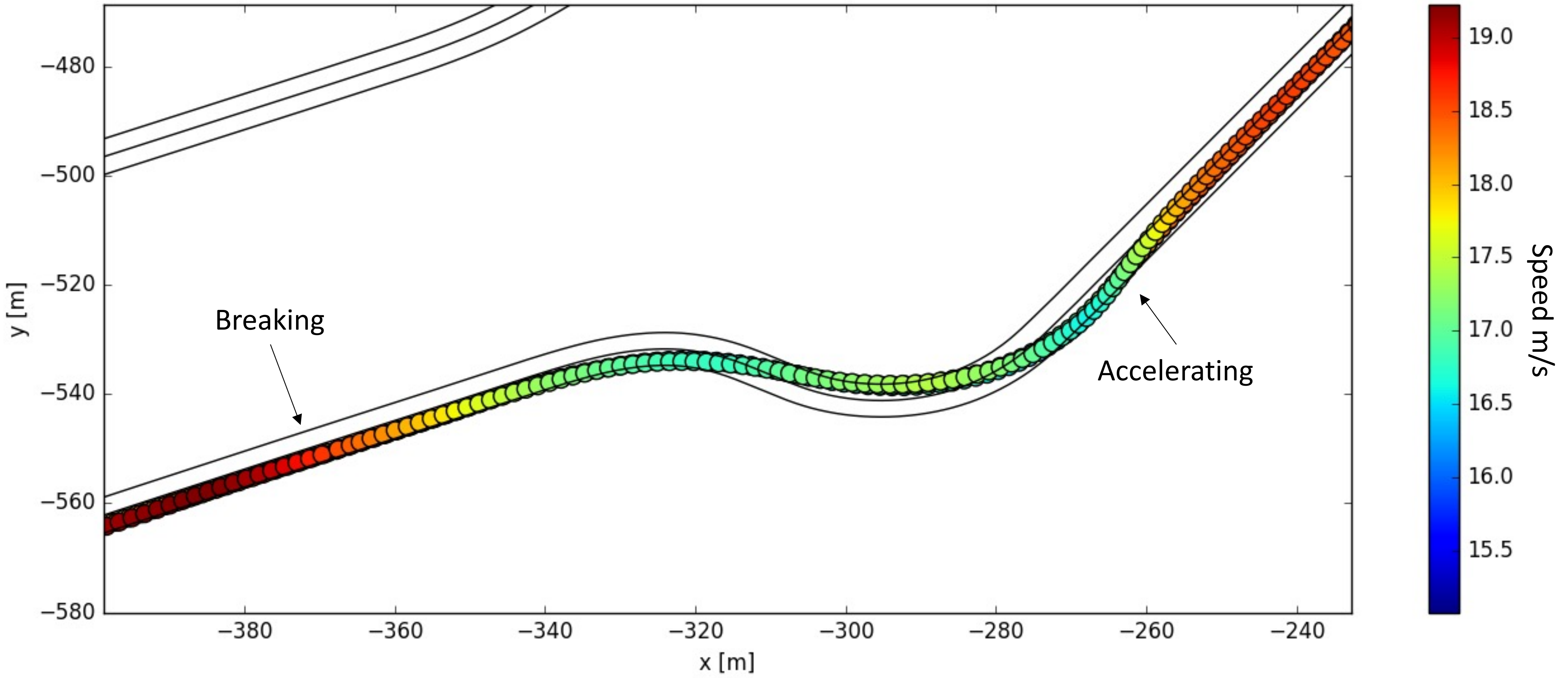
# Learning Model Predictive Controller full-size vehicle experiments

Credits: Siddharth Nair, Nitin Kapania and Ugo Rosolia

# Velocity Profile at Convergence (Curve 1)

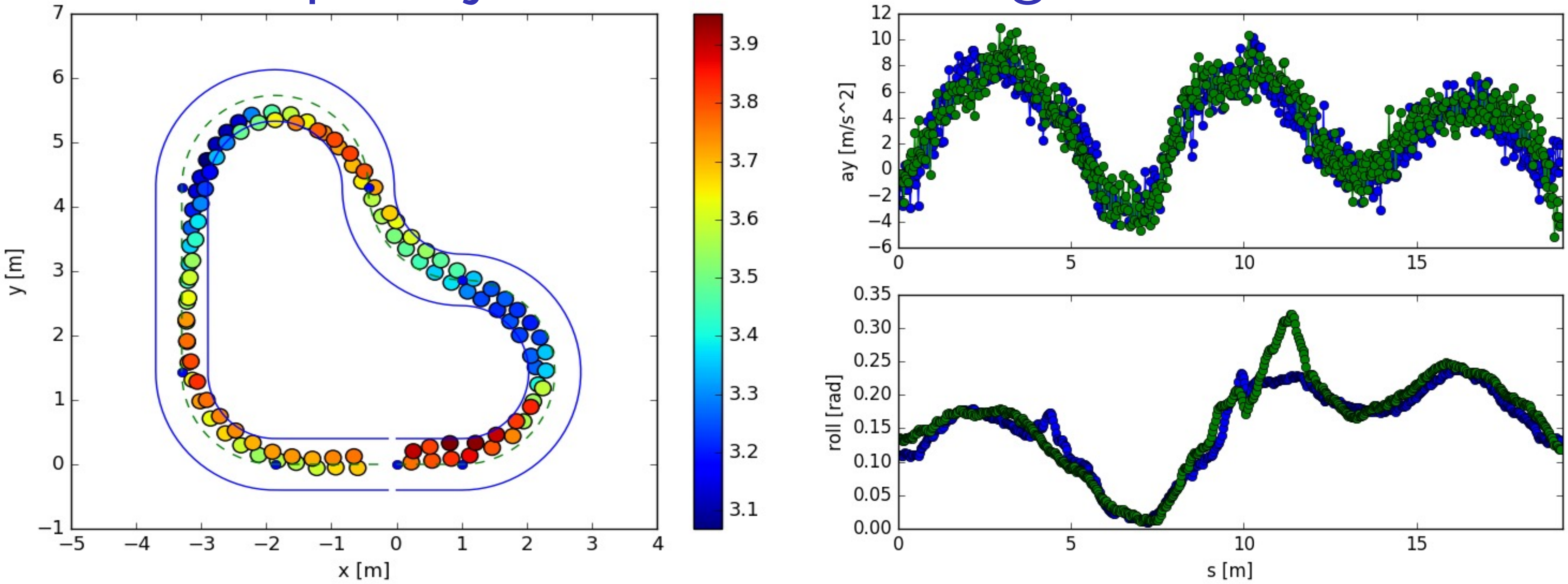


# Velocity Profile at Convergence (Chicane)



# Learning Model Predictictive Control for Autonomous Racing

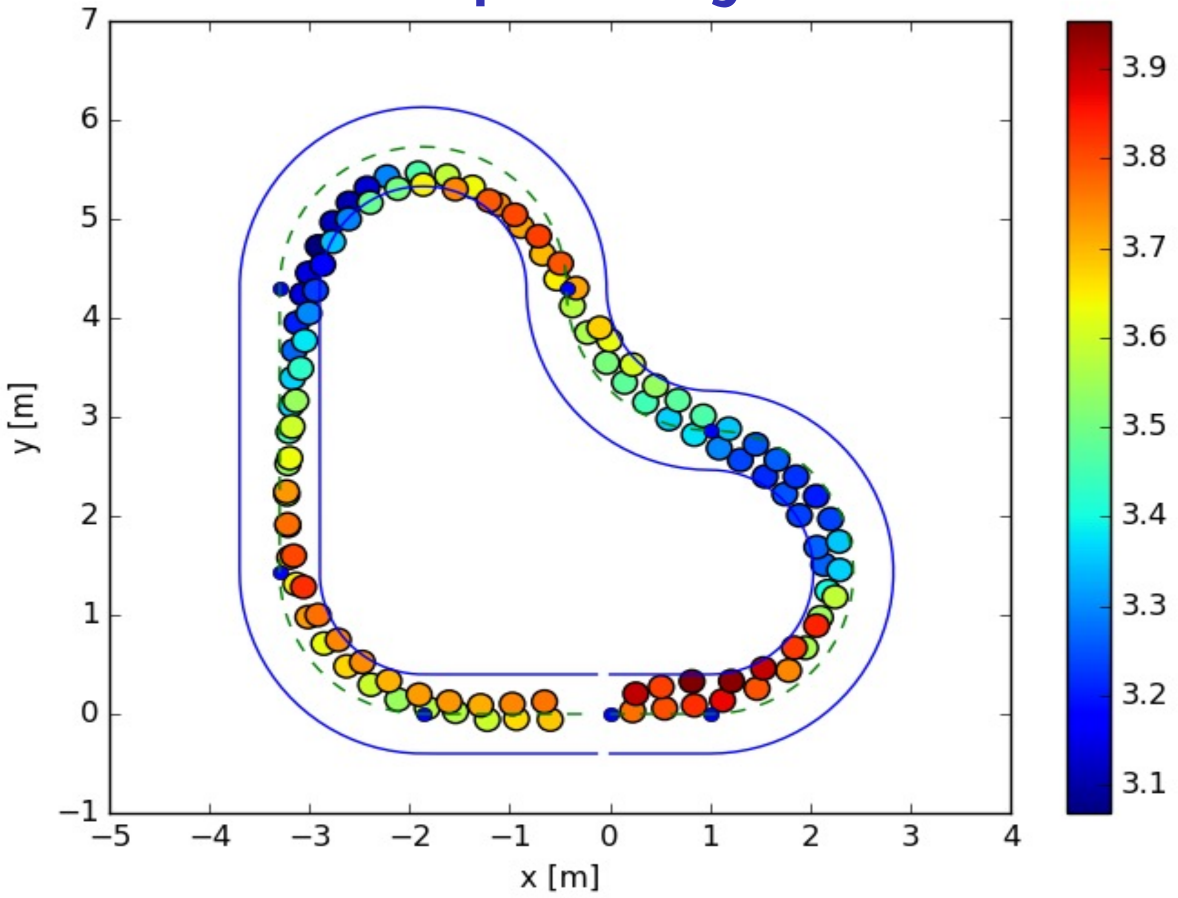
# Closed-loop Trajectories at Convergence

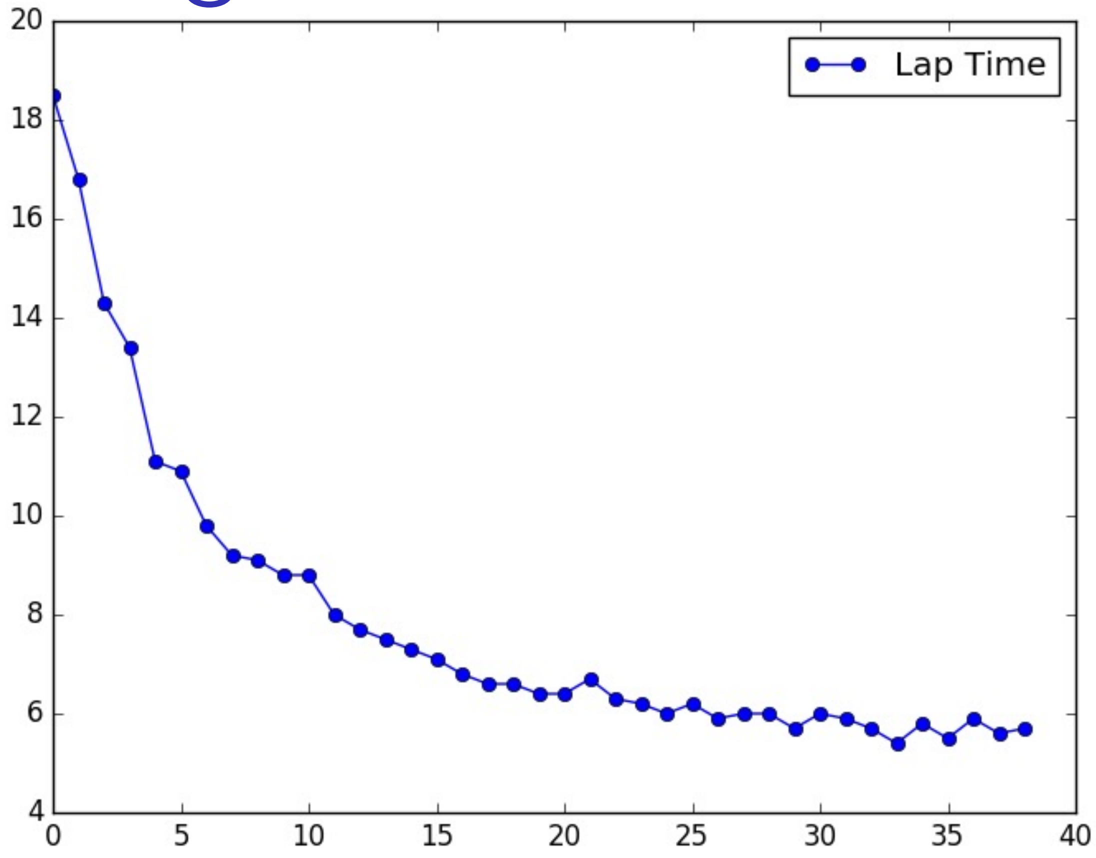



## Remarks

- 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

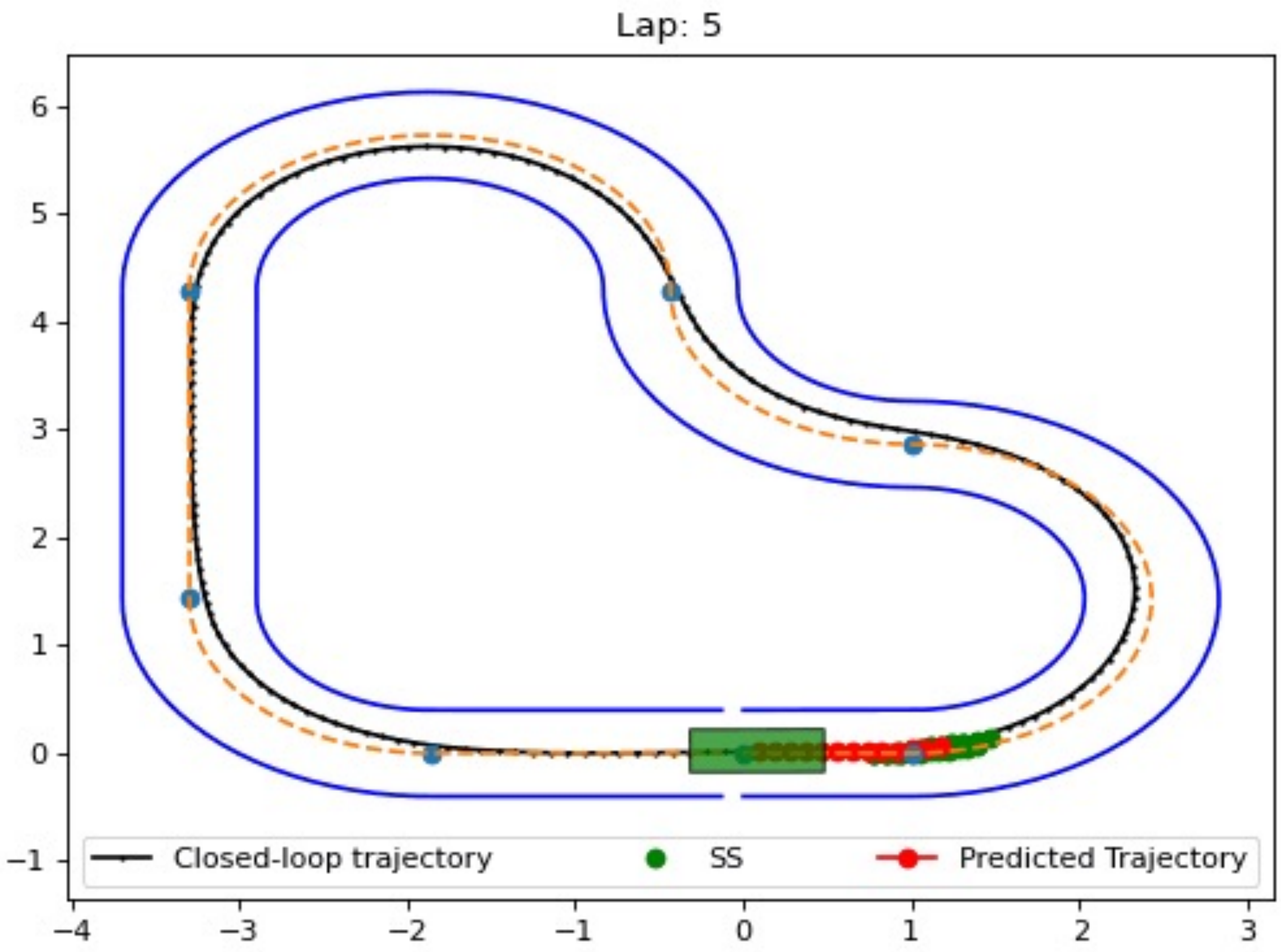


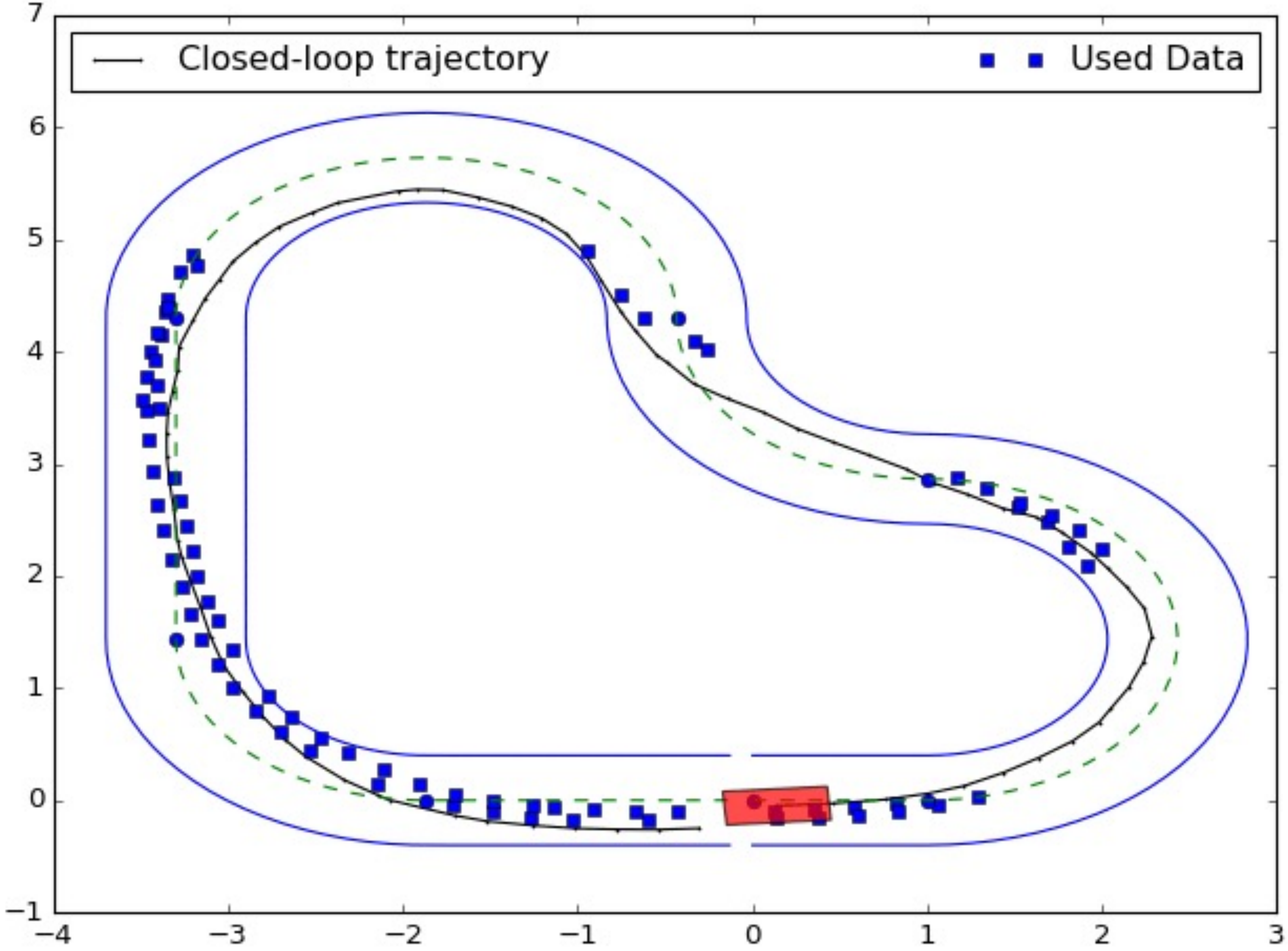

## Remarks

- 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

# Learning Safe Sets and Value Functions

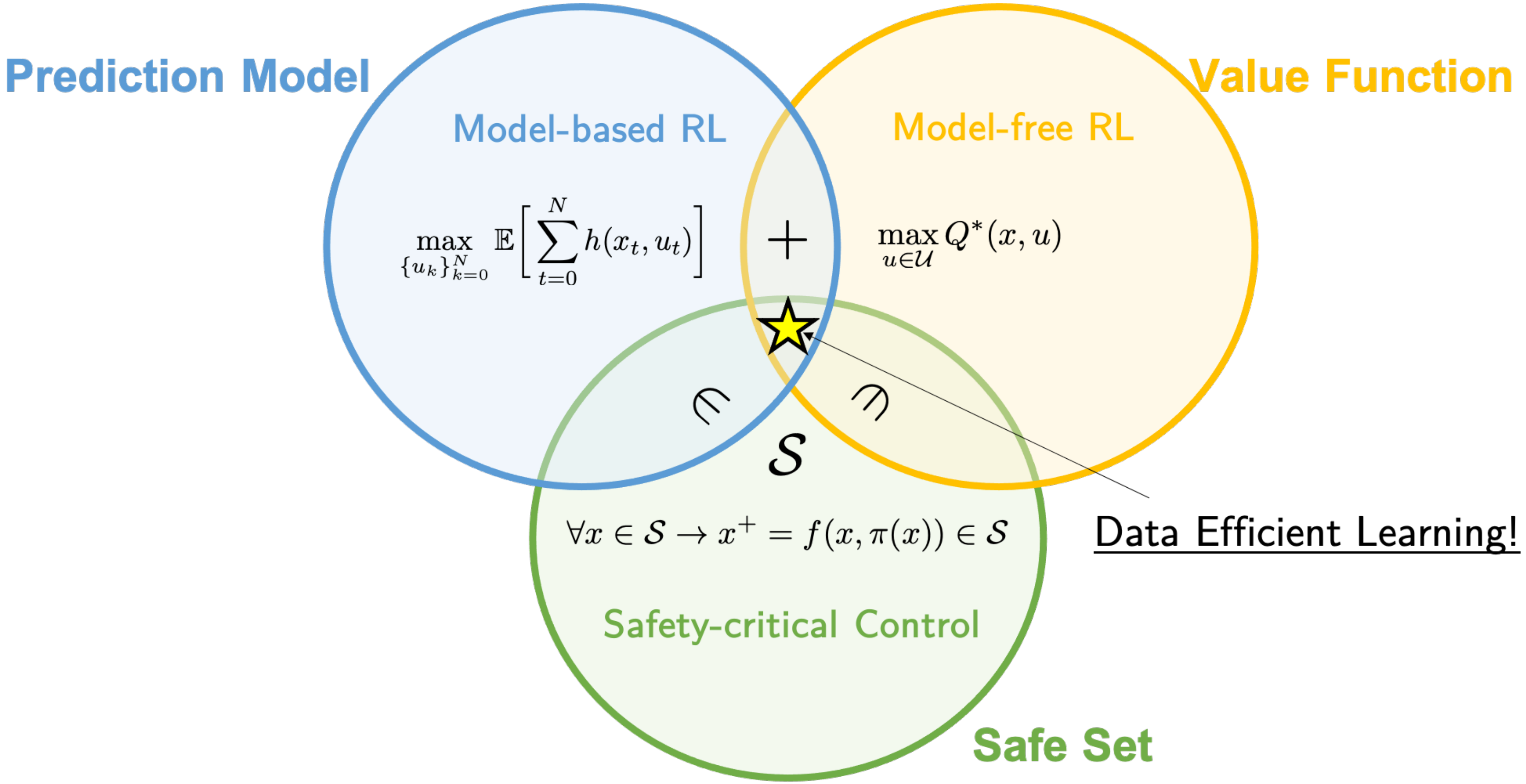


# Learning the Vehicle Model



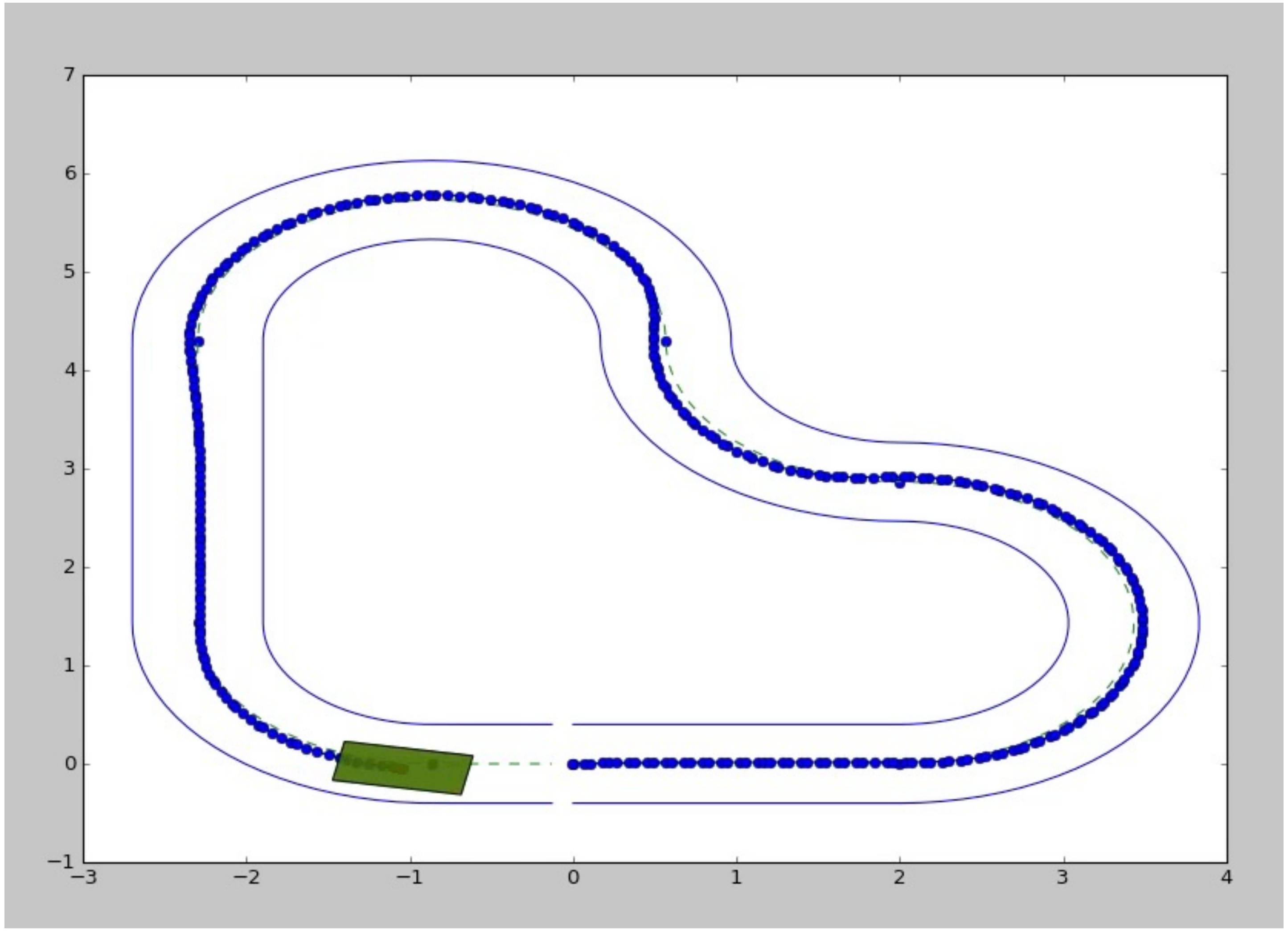
# The key components

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



# **Do you need the safe set?** – Yes
## LMPC without Invariant Set

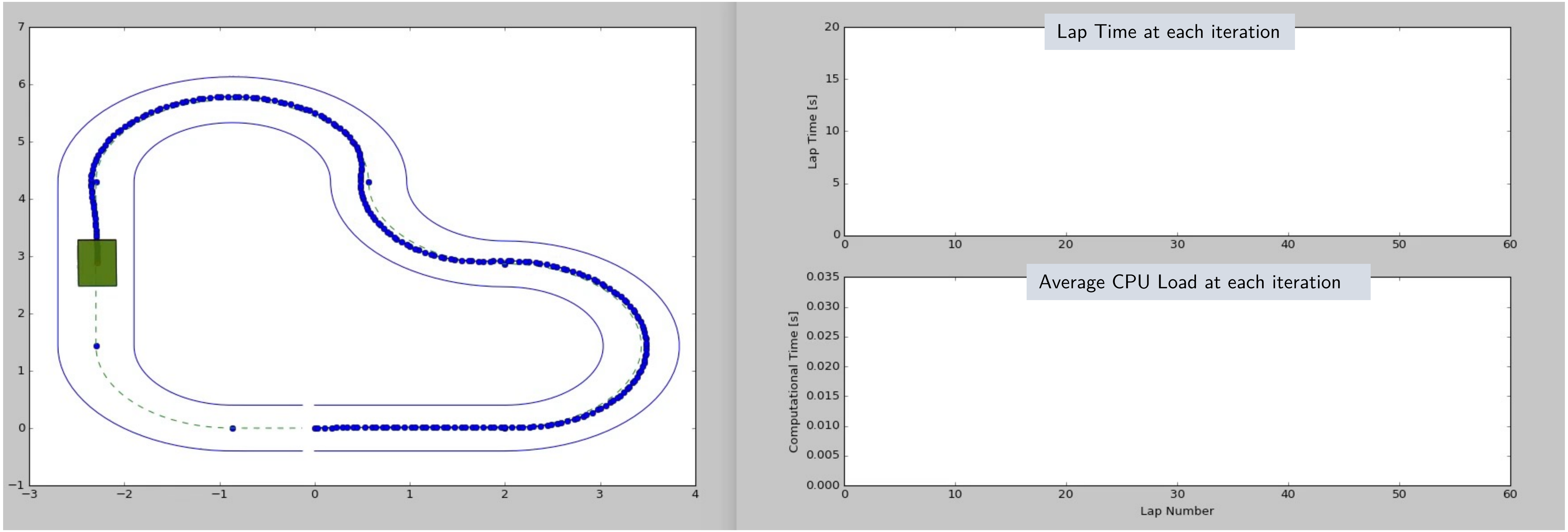
The controller extrapolates the Q-function on the Vx dimension



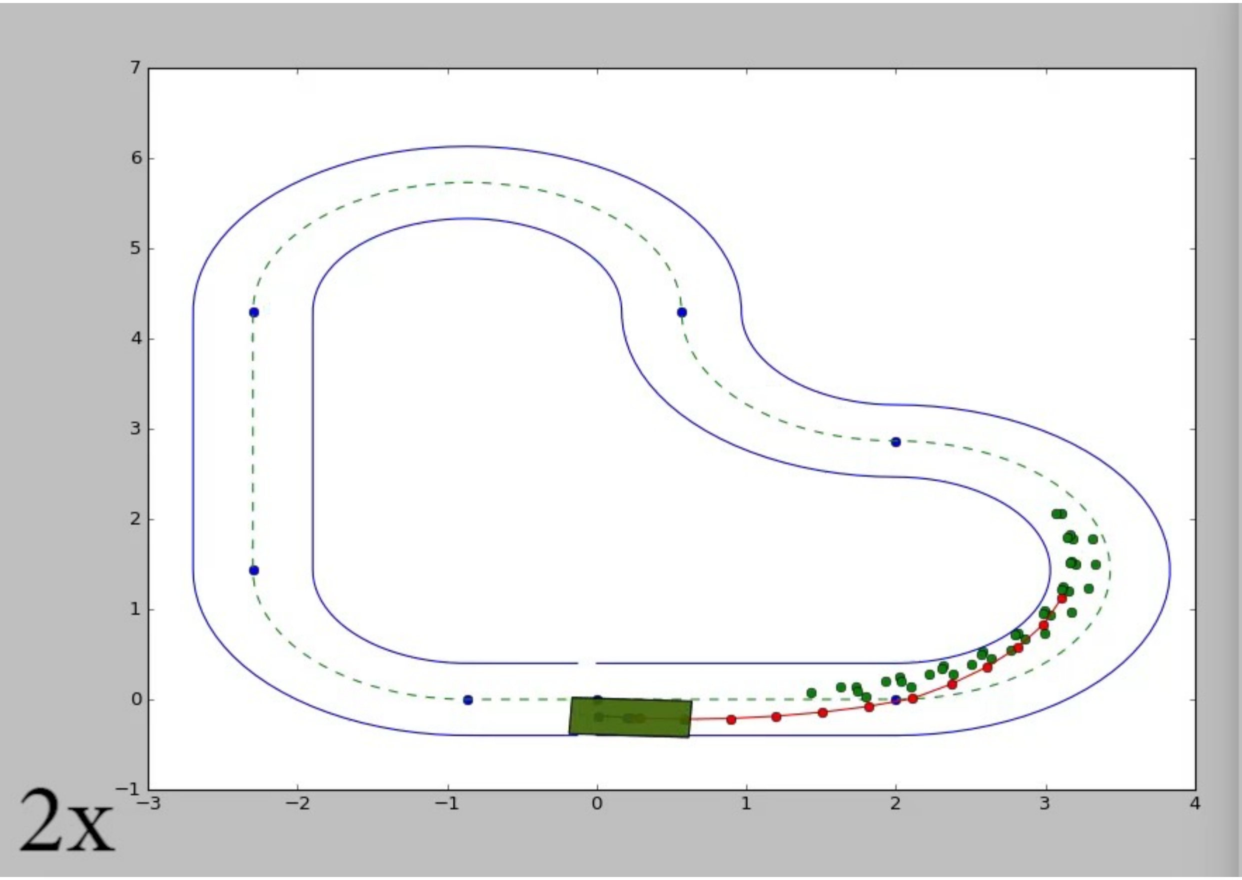
# Do you need to Predict to Learn? Yes

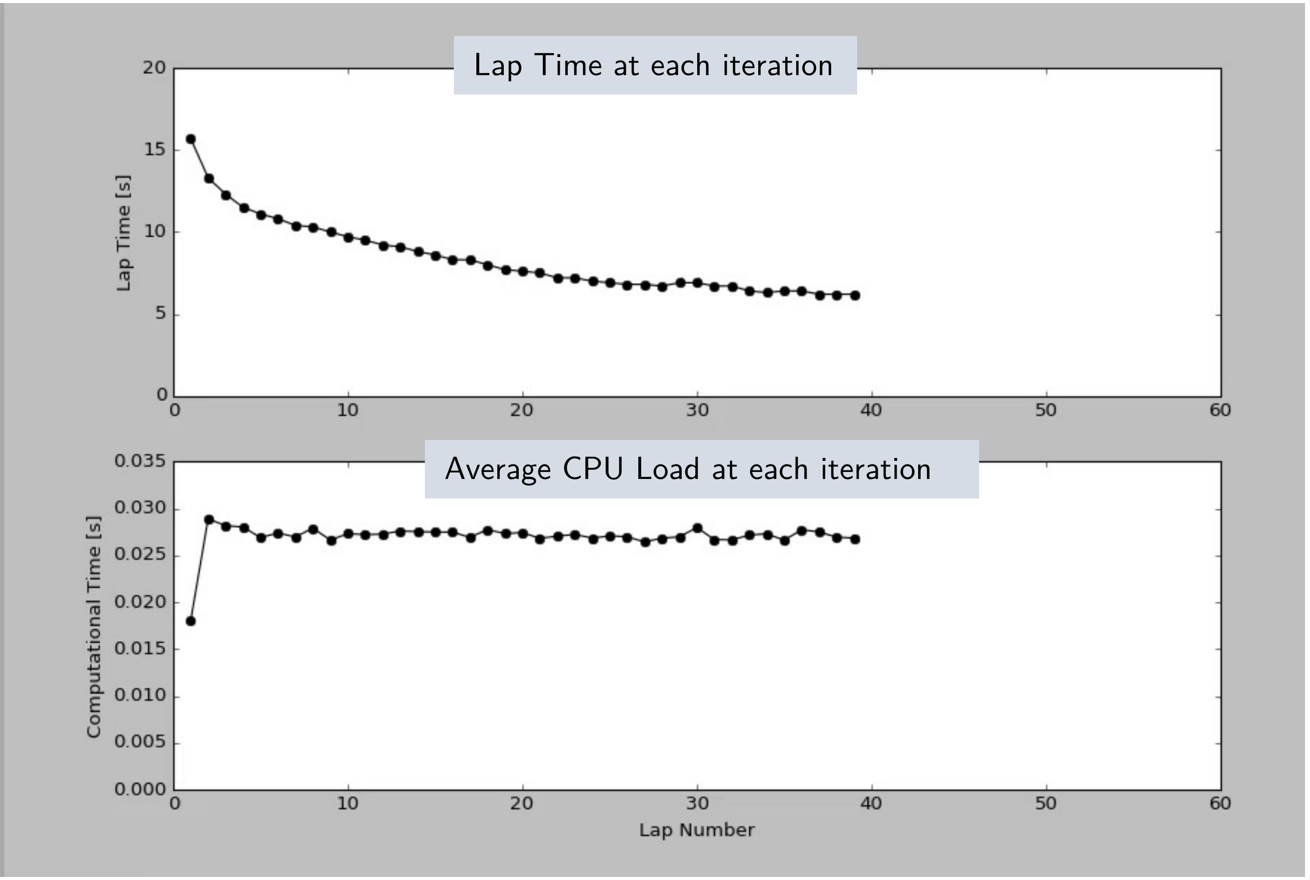
When the LMPC horizon is $N = 1$ the controller

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

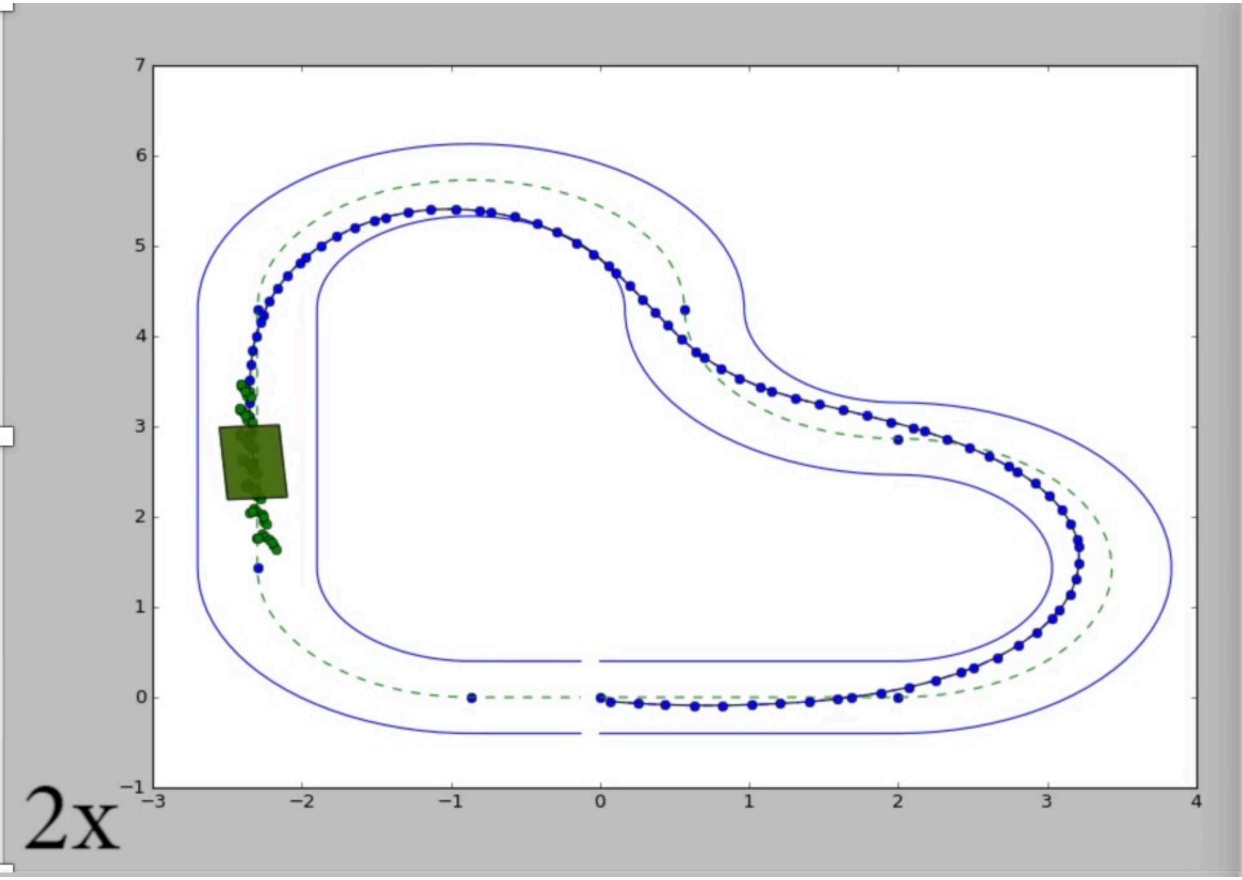


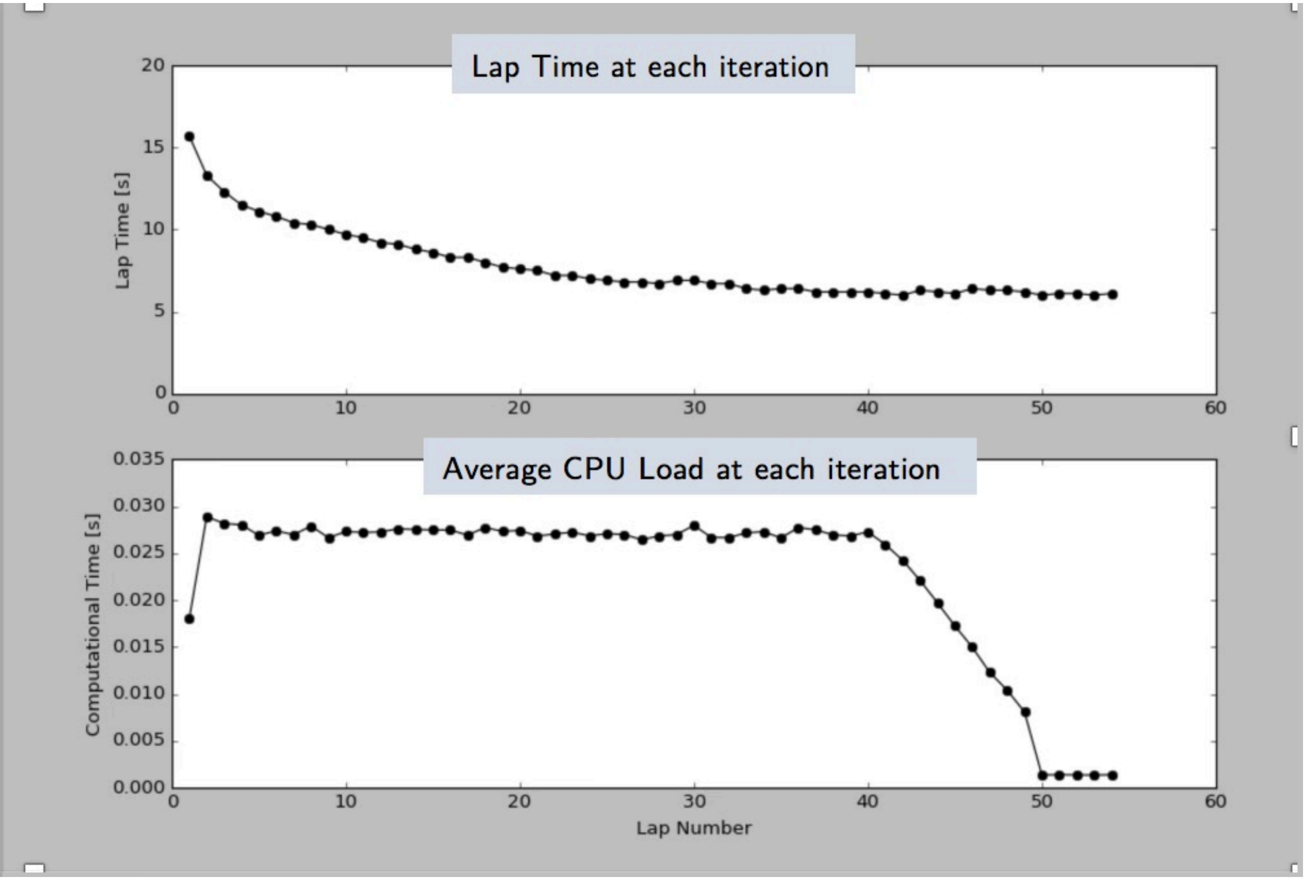
# Do you need to Predict at Convergence? No



# Do you need to Predict at Convergence? No

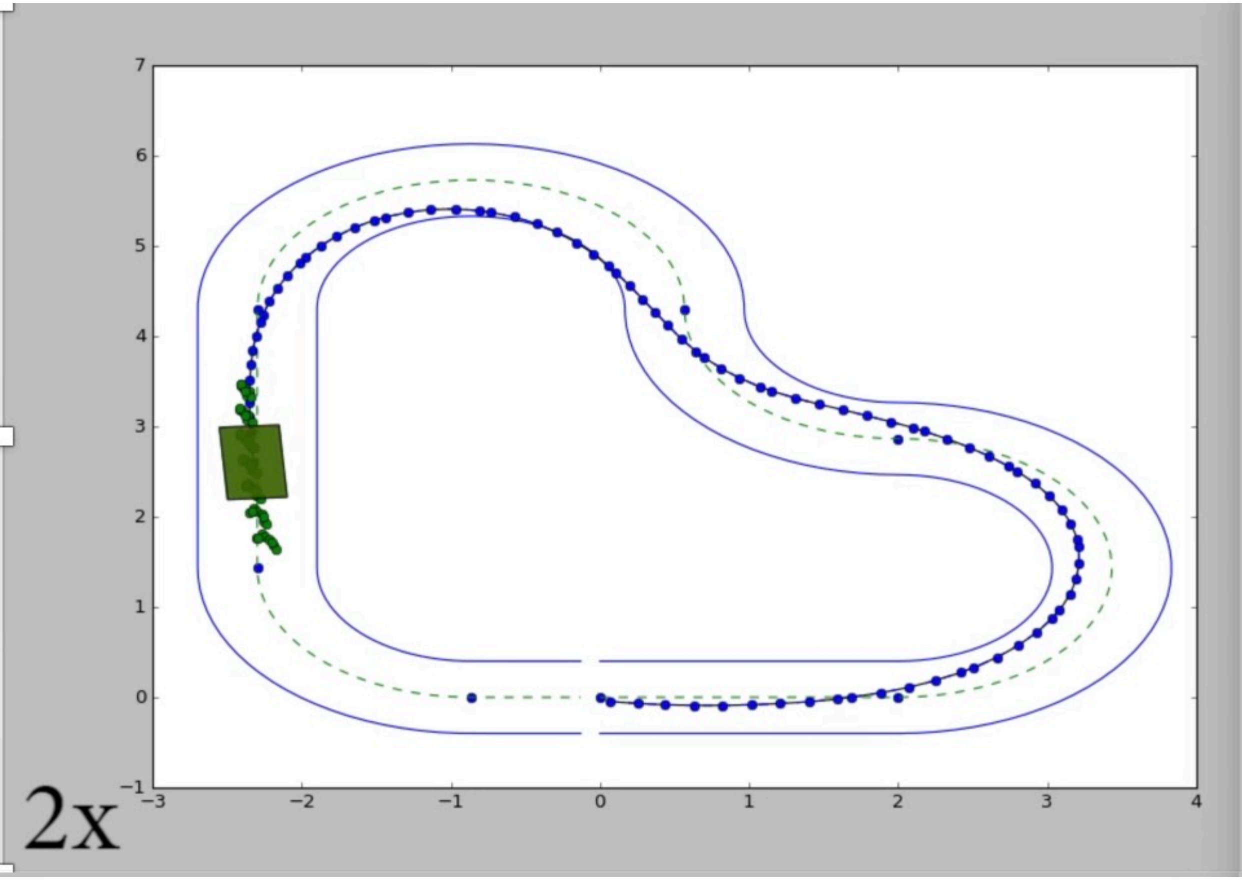


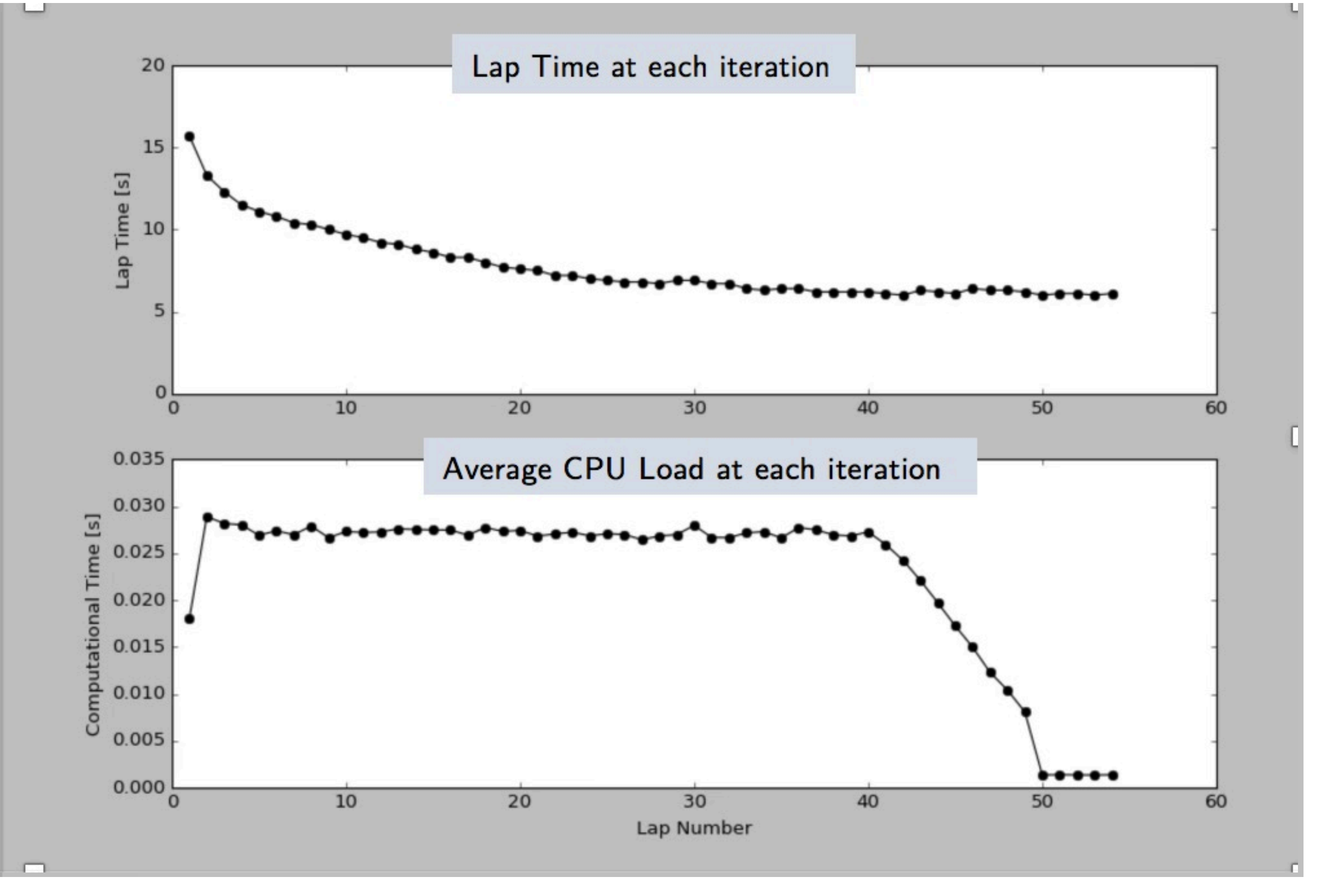




## Value Function Approximation

$$
\begin{aligned}
[\lambda_0^{0,*}, \ldots, \lambda_i^{j,*}] = \arg\min_{\lambda_i^j \in [0,1]} \quad & \sum_i \sum_j J_i^j \lambda_i^j \\
\text{s.t} \quad & \sum_i \sum_j x_i^j \lambda_i^j = x(t), \\
& \sum_i \sum_j \lambda_i^j = 1
\end{aligned}
$$

# Do you need to Predict at Convergence? No




## Value Function Approximation

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

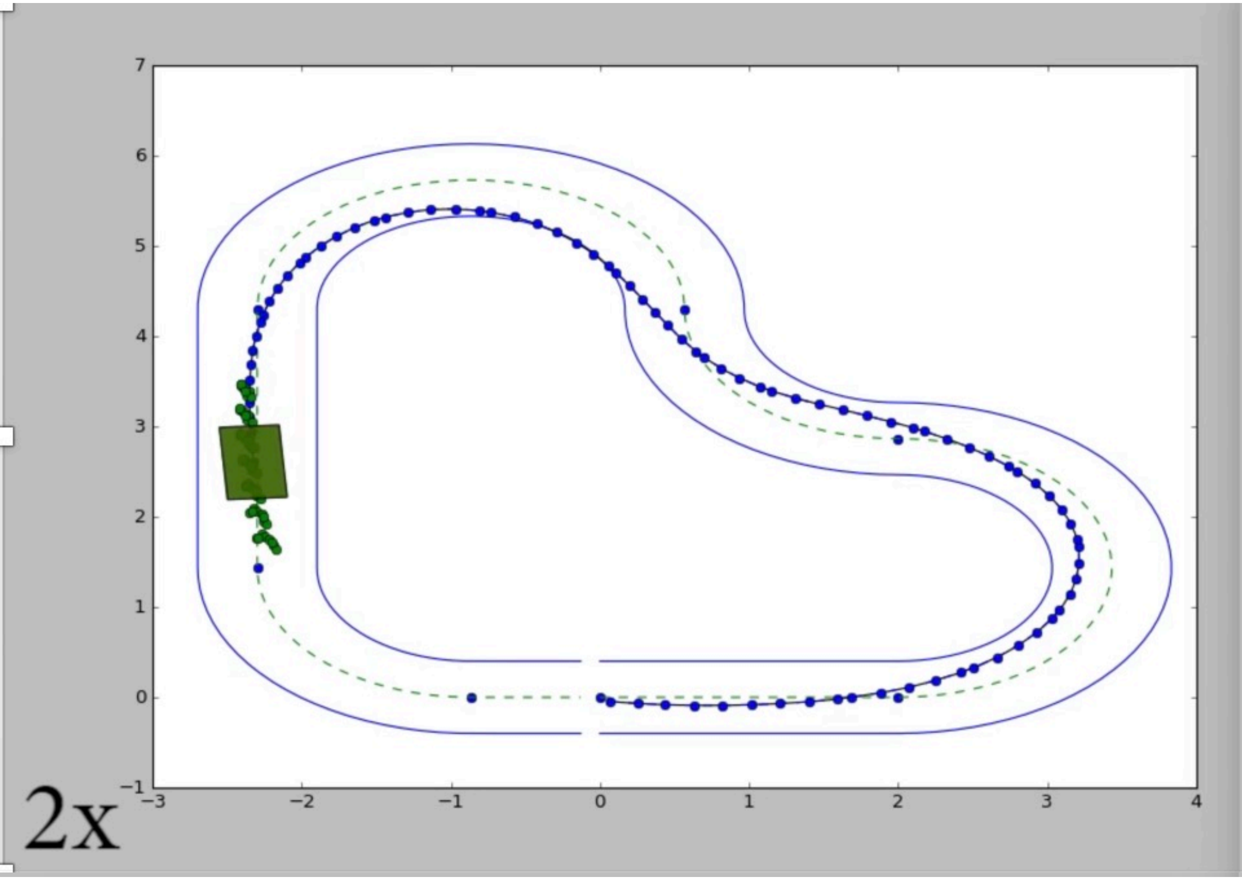
$$\text{s.t} \quad \sum_i \sum_j x_i^j \lambda_i^j = x(t),$$

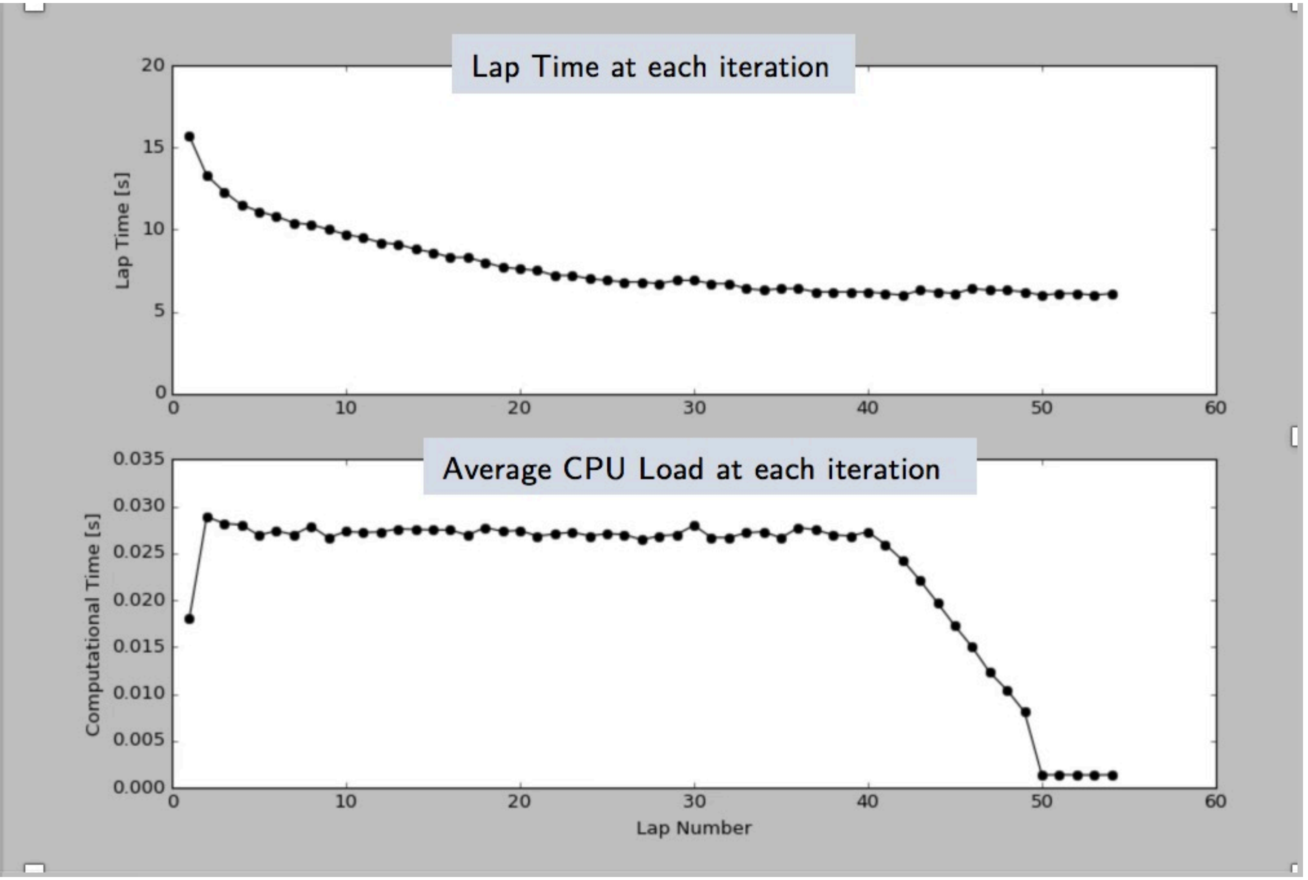
$$\sum_i \sum_j \lambda_i^j = 1$$

## Control Policy

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

# Do you need to Predict at Convergence? No




## Value Function Approximation

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

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

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

## Control Policy

Stored Data

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

# LMPC Summary

At each time *t* of iteration *j*, solve

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

s.t.

$$x^j_{k+1|t} = f(x^j_{k|t}, u^j_{k|t}),\ \forall k \in [t,\cdots,t+N-1]$$

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

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

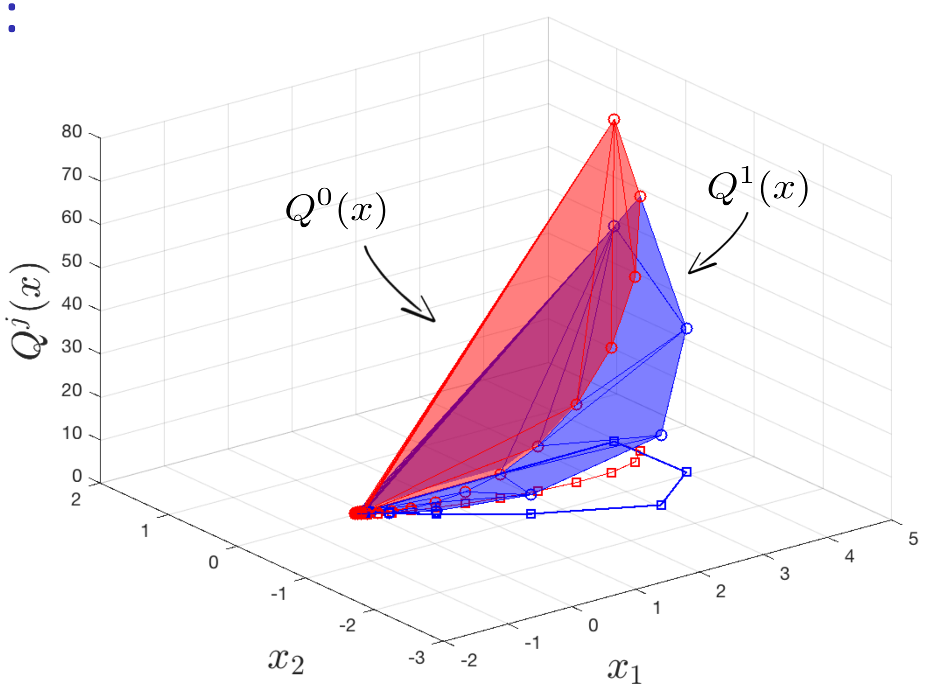
$$x^j_{t+N|t} \in \mathcal{SS}^{j-1},$$

Identified online

Constructed using LMPC strategy

# Conclusions

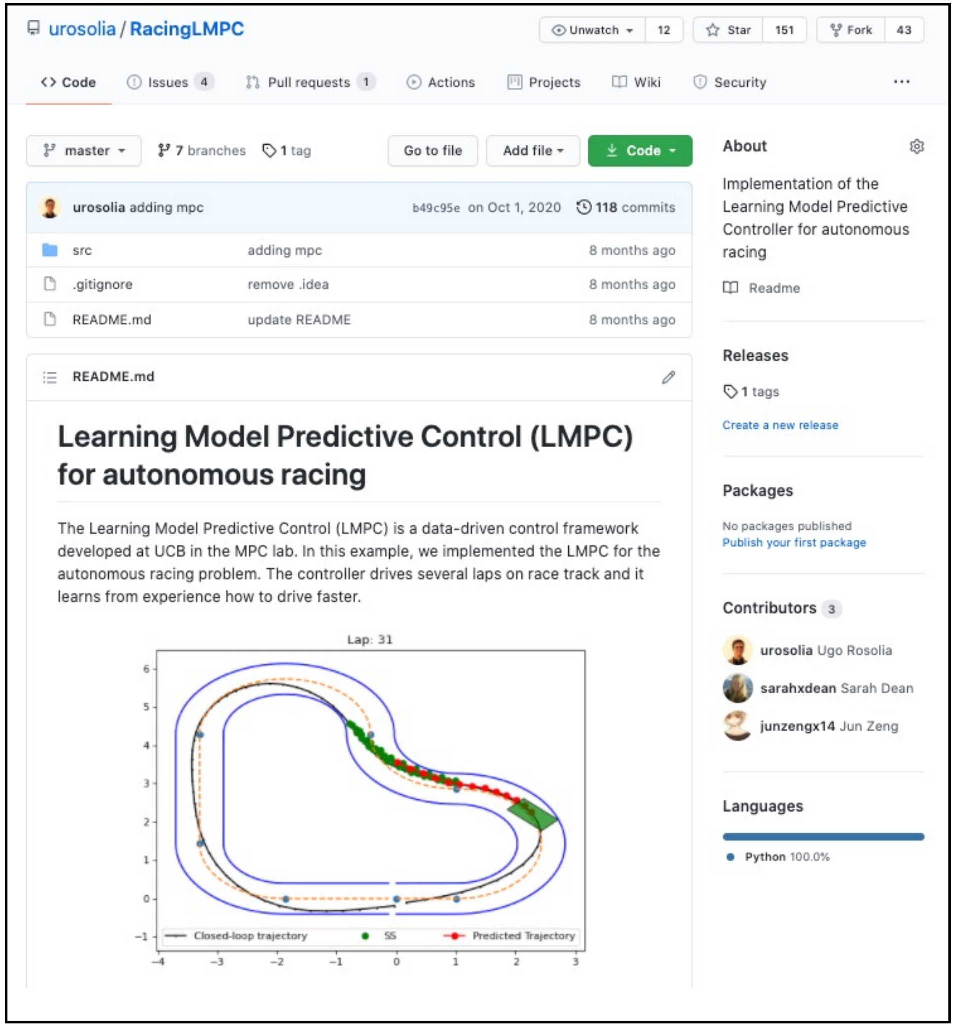
- Terminal Cost function (Value function approximation):
  - Defined on a subset of the state space
  - Constructed using a subset of the stored data




- Control Policy defined using MPC guarantees:
  - Safety (w/ some probability for sample-based LMPC)
  - Convergence (or ISS for expected cost)
  - Exploration and performance improvement (also for deterministic systems)

# Thanks! Questions?

Code available online




Course material online