Iterative Learning Model Predictive Control

Francesco Borrelli
Email: fborrelli@berkeley.edu
University of California
Berkeley, USA

www.mpc.berkeley.edu

Acknowledgements

Learning MPC: Ugo Rosolia

Adaptive MPC: Monimoy Bujarbaruah, George Xiaojing Zhang
Autonomous Drift: Edo Jelavick, George Xiaojing Zhang
Analog Optimization: Sergey Vicky
Autonomous Drift: Edo Jelavick, Yuri Glauthier
Analog Optimization: Sergey Vicky
Connected Cars: Jacopo Guanetti, Jongsang Suh, Roya Firoozi, Yeojun Kim , Eric Choi
BARC: Jon Gonzales, Tony Zeng, Charlott Vallon

Research Sponsors:

Hyundai Corporation

Ford Research Labs, Siemens, Mobis, Komatzu

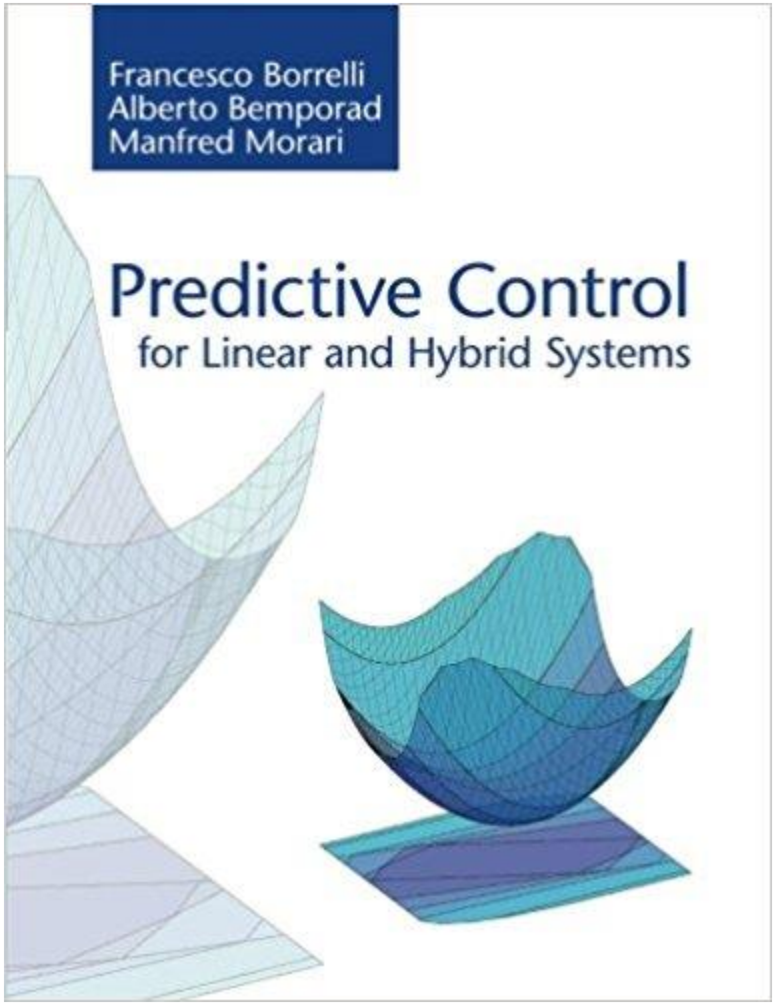
National Science Foundation

Office of Naval Research

Iterative Learning Model Predictive Control

Iterative Learning **Model Predictive Control**

Now Available on Amazon



Constrained Infinite-Time Optimal Control

$$
\begin{aligned}
J_0^*(x(0)) = \min_{\pi_0, \pi_1, \ldots} \; & \sum_{k=0}^{\infty} h(x_k, u_k) \\
\text{s.t. } & x_{k+1} = f(x_k, u_k) \\
& u_k = \pi_k(x_k) \\
& x_k \in \mathcal{X}, u_k \in \mathcal{U}, \\
& x_0 = x(0)
\end{aligned}
$$

$\pi_k(\cdot)$ Feedback Control Policies: $\pi_k : \; x_k \in \mathcal{X} \mapsto u_k \in \mathcal{U}$

"Solved" as..

Repeated Solution of Constrained Finite Time Optimal Control

$$\min_{\pi_0,\pi_1,\ldots,\pi_{N-1}} p(x_{t+N}) + \sum_{k=0}^{N-1} h(x_{t+k}, u_{t+k})$$

$$\begin{array}{c}\text{subj. to}\\ k=t,\ldots,t+N-1\end{array} \left\{ \begin{array}{l} x_{k+1} = f(x_k, u_k) \\ u_k = \pi_k(x_k) \\ u_k \in \mathcal{U}, x_k \in \mathcal{X} \\ x_{t+N} \in \mathcal{X}_f \\ x_t = x(t) \end{array} \right.$$

$\pi_k(\cdot)$ Feedback Control Policies: $\pi_k : \; x_k \in \mathcal{X} \mapsto u_k \in \mathcal{U}$

Predictive Controller:

$$\boxed{u(t) = \pi_0^*(x(t))}$$

Repeated Solution of Constrained Finite Time Optimal Control

$$\min_{\pi_0,\pi_1,\ldots,\pi_{N-1}} p(x_{t+N}) + \sum_{k=0}^{N-1} h(x_{t+k}, u_{t+k})$$

$$\begin{array}{c}\text{subj. to}\\ k = t,\ldots,t+N-1\end{array} \quad \left\{\begin{array}{l} x_{k+1} = f(x_k, u_k) \\ u_k = \pi_k(x_k) \\ u_k \in \mathcal{U}, x_k \in \mathcal{X} \\ x_{t+N} \in \mathcal{X}_f \\ x_t = x(t) \end{array}\right.$$

$\pi_k(\cdot)$ Feedback Control Policies: $\pi_k : \; x_k \in \mathcal{X} \mapsto u_k \in \mathcal{U}$

- $p(\cdot)$ Approximates the `tail' of the cost
- $\mathcal{X}_f$ Approximates the `tail' of the constraints
- *N* constrained by computation and forecast uncertainty
- Robust and stochastic versions subject of current research

Repeated Solution of Constrained Finite Time Optimal Control

$$\min_{\pi_0,\pi_1,\ldots,\pi_{N-1}} p(x_{t+N}) + \sum_{k=0}^{N-1} h(x_{t+k}, u_{t+k})$$

$$\begin{array}{c}\text{subj. to}\\ k=t,\ldots,t+N-1\end{array} \left\{\begin{array}{l} x_{k+1} = f(x_k, u_k)\\ u_k = \pi_k(x_k)\\ u_k \in \mathcal{U}, x_k \in \mathcal{X}\\ x_{t+N} \in \mathcal{X}_f\\ x_t = x(t)\end{array}\right.$$

$\pi_k(\cdot)$ Feedback Control Policies: $\pi_k : \ x_k \in \mathcal{X} \mapsto u_k \in \mathcal{U}$

Predictive Controller: $u(t) = \pi_0^*(x(t))$

Predictive Control: Theory & Computation

Repeated Solution of Constrained Finite Time Optimal Control

$$\min_{\pi_0,\pi_1,\ldots,\pi_{N-1}} p(x_{t+N}) + \sum_{k=0}^{N-1} h(x_{t+k}, u_{t+k})$$

$$\begin{array}{c}\text{subj. to} \\ k = t, \ldots, t+N-1\end{array} \left\{ \begin{array}{l} x_{k+1} = f(x_k, u_k) \\ u_k = \pi_k(x_k) \\ u_k \in \mathcal{U}, x_k \in \mathcal{X} \\ x_{t+N} \in \mathcal{X}_f \\ x_t = x(t) \end{array} \right.$$

$\pi_k(\cdot)$ Feedback Control Policies: $\pi_k : \; x_k \in \mathcal{X} \mapsto u_k \in \mathcal{U}$

Predictive Controller: $u(t) = \pi_0^*(x(t))$

Predictive Control **Classical Theory**

Predictive Control Theory: Sufficient conditions to guarantee

- Convergence to the desired equilibrium point/region
- Constraint satisfaction at all times

$$\min_{\pi_0,\pi_1,\ldots,\pi_{N-1}} \sum_{k=0}^{N-1} h(x_{t+k}, u_{t+k}) + p(x_{t+N})$$

$$\underset{k=t,\ldots,t+N-1}{\text{subj. to}} \quad \begin{cases} x_{k+1} = f(x_k, u_k) \\ u_k = \pi_k(x_k) \\ u_k \in \mathcal{U}, x_k \in \mathcal{X} \\ x_{t+N} \in \mathcal{X}_f \\ x_t = x(t) \end{cases}$$

Terminal cost:
Control Lyapunov function

Terminal constraint set:
Control Invariant set

Control Invariant Set

$$x_0 \in \mathcal{X}_f \rightarrow \exists u_k \in \mathcal{U} : f(x_k, u_k) \in \mathcal{X}_f \ \forall k > 0$$

Control Lyapunov Function

$$\min_{u \in \mathcal{U}, f(x,u) \in \mathcal{X}_f} (p(f(x,u)) - p(x) + h(x,v)) \leq 0, \ \forall x \in \mathcal{X}_f$$

Repeated Solution of Constrained Finite Time Optimal Control

$$\min_{\pi_0,\pi_1,\ldots,\pi_{N-1}} p(x_{t+N}) + \sum_{k=0}^{N-1} h(x_{t+k}, u_{t+k})$$

$$\begin{array}{c}\text{subj. to}\\ k = t, \ldots, t+N-1\end{array} \quad \left\{\begin{array}{l} x_{k+1} = f(x_k, u_k) \\ u_k = \pi_k(x_k) \\ u_k \in \mathcal{U}, x_k \in \mathcal{X} \\ x_{t+N} \in \mathcal{X}_f \\ x_t = x(t) \end{array}\right.$$

$\pi_k(\cdot)$ Feedback Control Policies: $\pi_k : \; x_k \in \mathcal{X} \mapsto u_k \in \mathcal{U}$

Predictive Controller: $u(t) = \pi_0^*(x(t))$

Predictive Control **Computation**

Offline $\pi(\cdot)$ and Online $\pi(x)$ Computation

$$\min_{\pi_0(\cdot),\pi_1(\cdot),\ldots,\pi_{N-1}(\cdot)} J_{0\to N}(x_0,\Pi)$$

$$\begin{array}{c}\text{subj. to}\\ k=0,\ldots,N-1\end{array} \quad \left\{\begin{array}{l} x_{k+1}=f(x_k,u_k,w_k)\\ u_k=\pi_k(x_k)\\ u_k\in\mathcal{U}, x_k\in\mathcal{X}, \quad \forall w_k\in\mathcal{W}\end{array}\right.$$

$\pi_k(\cdot)$ Feedback Control Policies: $\quad \pi_k:\ x_k\in\mathcal{X}\mapsto u_k\in\mathcal{U}$

Option 1 (*Offline Based*): "Complex" Offline, "Simple" Online

- $\pi_0(\cdot)$ often piecewise constant or affine disturbance feedback
- Dynamic Programing is one choice
- Sampling model reduction/aggregation required for n>5

Option 2 (*Online Based*): "Simple" Offline, "Complex" Online

- Compute on-line $\pi_0(x(t))$ with a "sophisticated" algorithm
- Interior point method solver is one choice
- Convexification required for real-time embedded control

Major effort over the past 20 years for enlarging MPC application domain

A very biased story

- Online Based
  - Excellent, (non-) convex open-source solvers
  - Tailored solvers for embedded linear and nonlinear MPC
- Offline Based
  - For linear and piecewise linear systems: explicit MPC
- Mixing pre-computation and online-optimization
- Suboptimal MPC
- Fast Online Implementation on embedded FPGA, GPU
- Analog MPC: microsecond sampling time

Iterative Learning Model Predictive Control

Three Forms of Learning
1 - Skill acquisition

Three Forms of Learning

2 - Performance Improvement

Three Forms of Learning

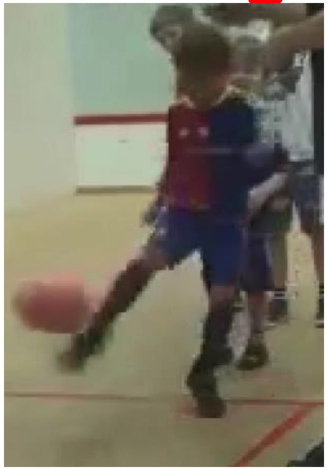
3 - Computation Load Reduction

Three Forms of Learning. Practice in order to:

Acquire a Skill

Improve Performance

Reduce Computational Load



Learning from demonstration
Transfer learning
Learning from simulations

Iterative Learning

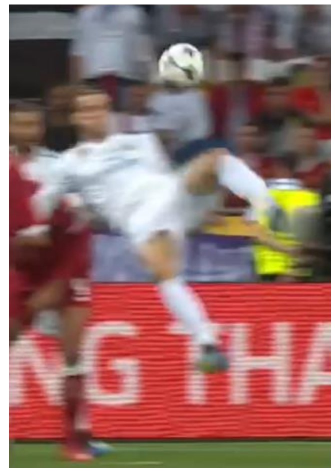
Computational load reduction of control policy

Three Forms of Learning.
Practice in order to:

Acquire a Skill

Learning from demonstration
Transfer learning
Learning from simulations

Improve Performance



Iterative Learning

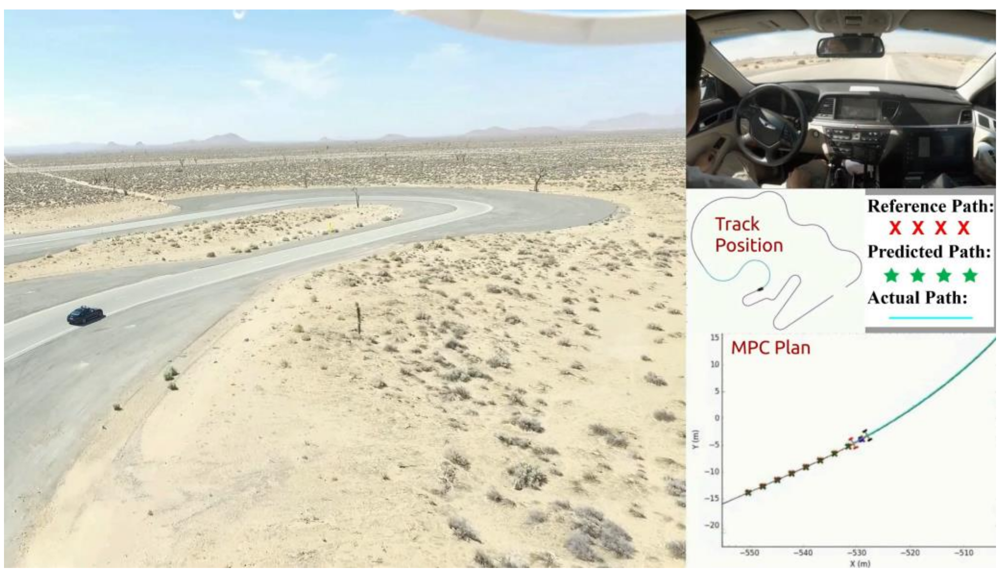
Reduce Computational Load

Computational reduction
of control policy

Learning MPC Applied to Robo-Cars

(instead of robo-soccer players..)

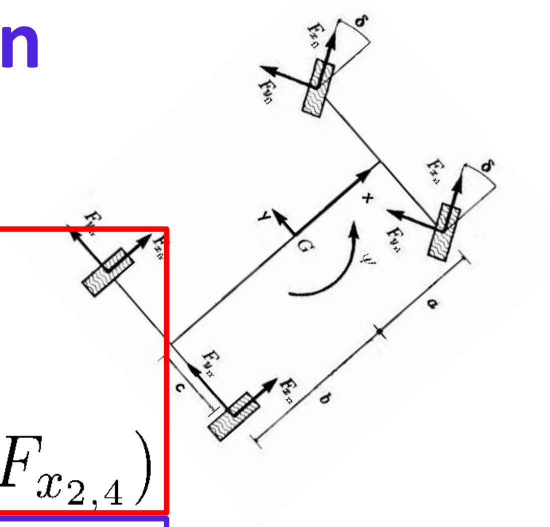
Autonomous Cars @MPC Lab



Autonomous Vehicles- Motion Control Through:

- Acceleration, Braking, Steering
- Also:
  - 4 braking torques
  - Gear Ratio
  - Engine torque + front and rear distribution
  - 4 dampers for active suspensions

Useful Model Abstraction



- Nonlinear Dynamical System

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

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

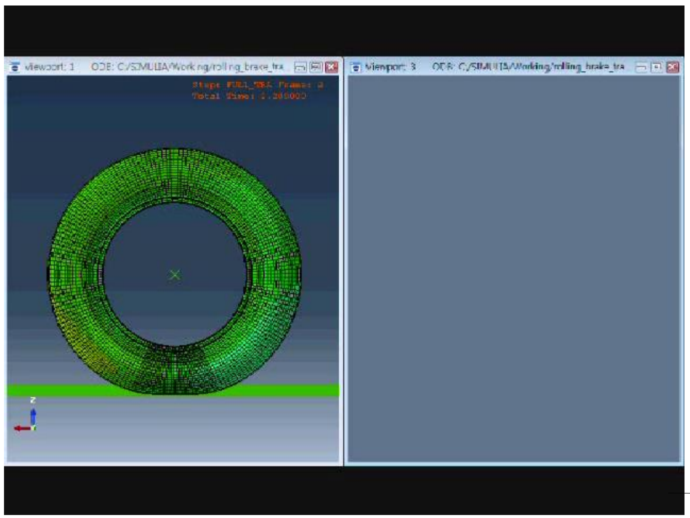
- Static Nonlinearities: Tires

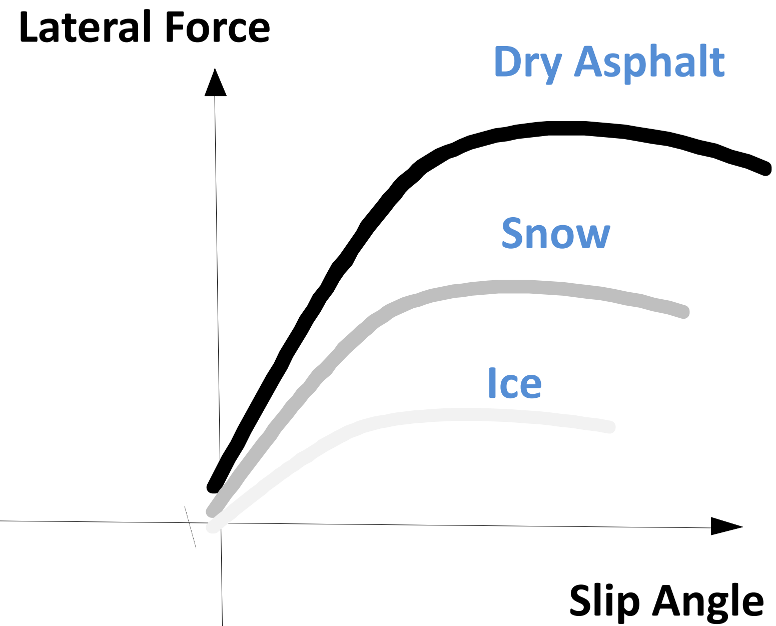
$$F_y = f_y(\alpha, \sigma, \mu, F_z) \qquad F_x = f_x(\alpha, \sigma, \mu, F_z)$$ and $$\sqrt{F_x^2 + F_y^2} <= mg$$

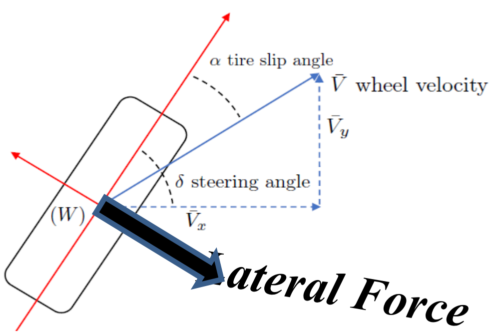
- Inequality Constraints: Safety region

- Uncertain Tire Model, Road Friction, Obstacles

Tires and Road
Simplified Nonlinear Model

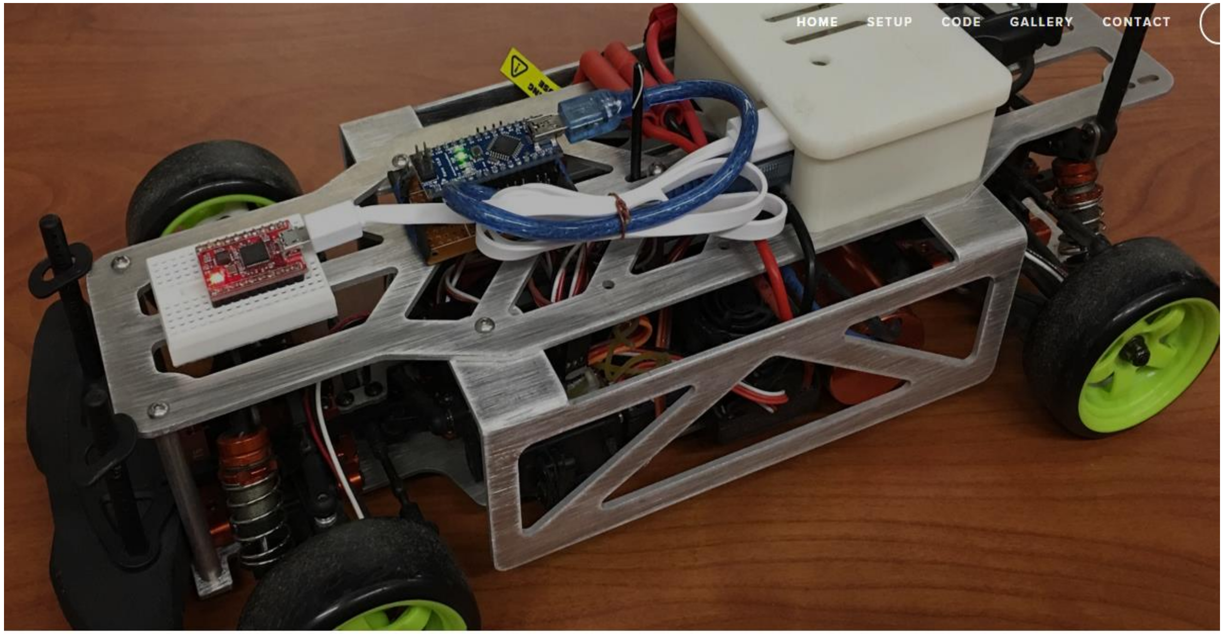






$$\sqrt{F_x^2 + F_y^2} <= \mu m g$$

Berkeley Autonomous 1/10 Race Car Project
www.barc-project.com

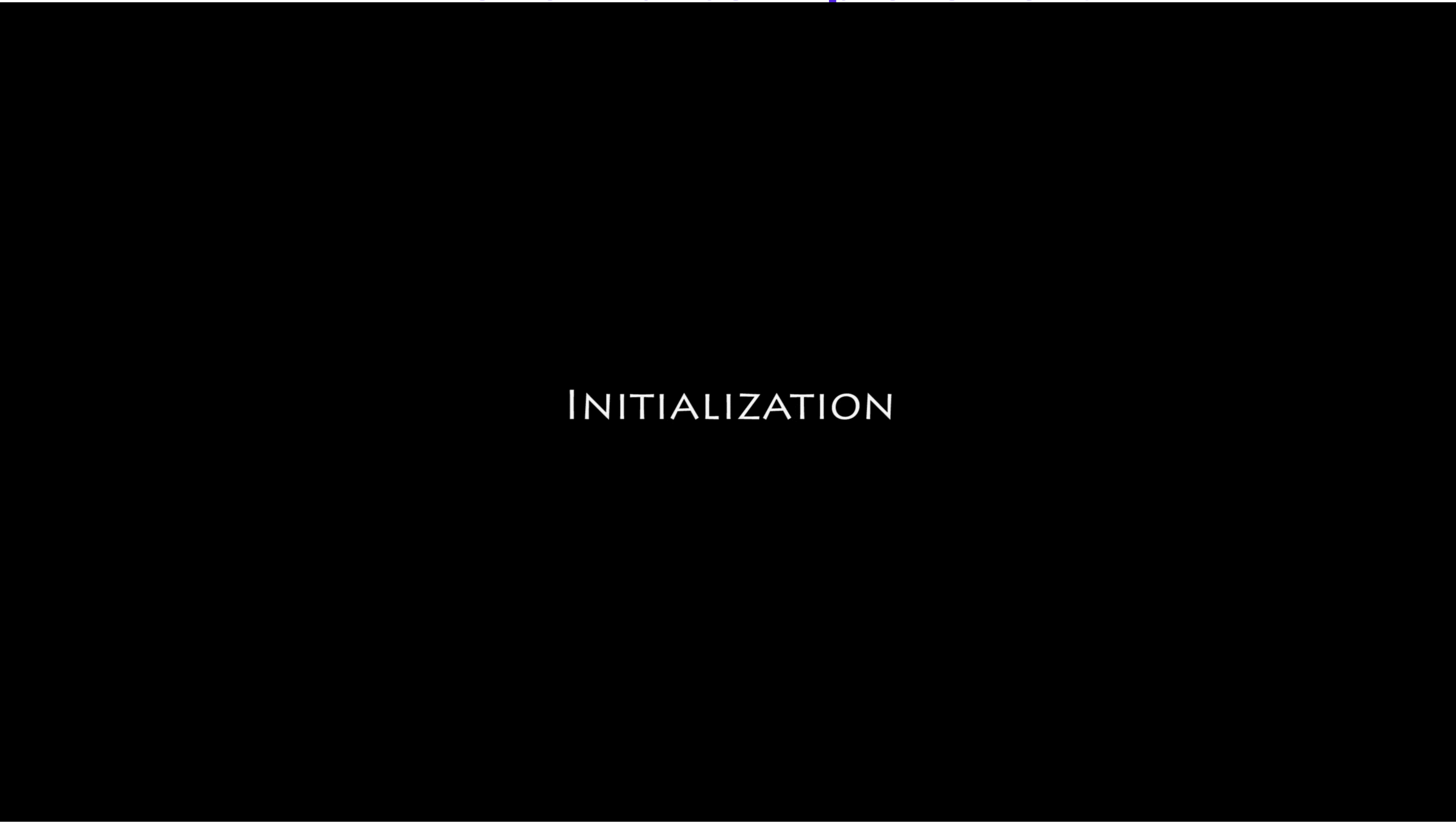


RC Car Racing Meets Cloud Computing

- Complete Open Source
- Ubuntu, RoS, OpenCV, Julia, IPOPT
- Camera, IMU, Ultrasounds, LIDAR
- Cloud-Based

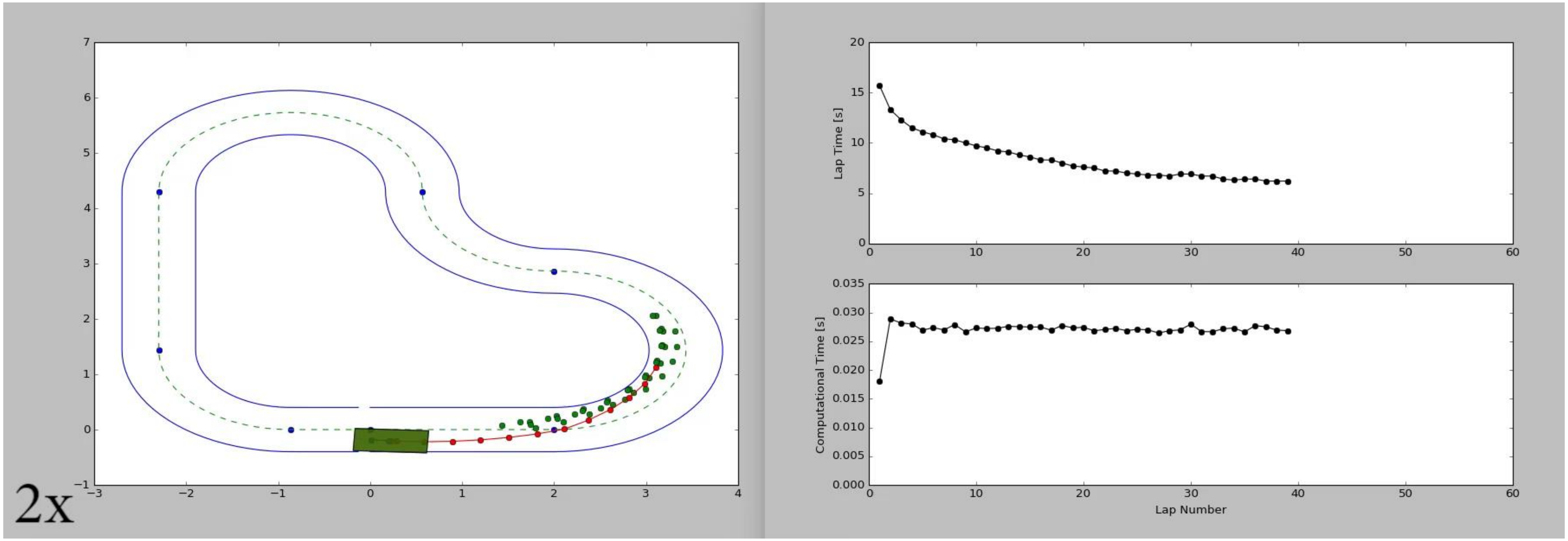
Three Forms of Learning
1 - **Skill acquisition**

Three Forms of Learning
2 - **Performance Improvement**



Three Forms of Learning
3 - **Computation Load Reduction**

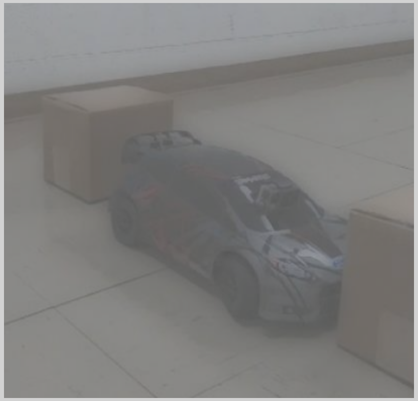
Lap Time at each iteration



Average CPU Load at each iteration

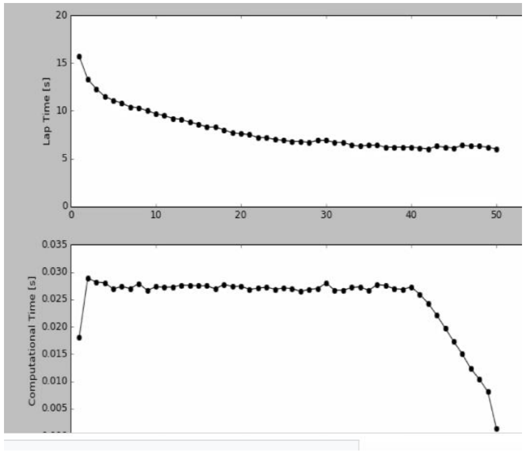
Three Forms of Learning

Acquire a Skill



Improve Performance

Reduce Computational Load



How we do this?

Model Predictive Control

A Simple Idea (which exploits the iterative nature of the tasks)

Important Design Steps

Iterative Learning Model Predictive Control

Iterative Tasks - Problem Setup

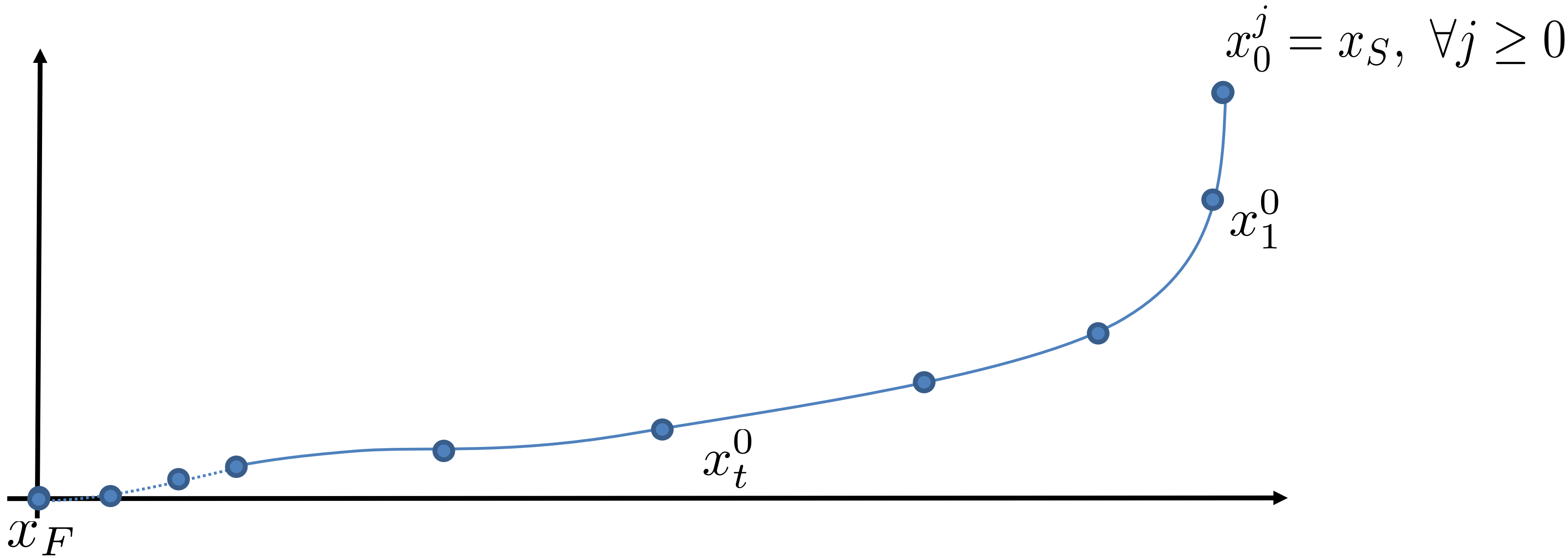
- One task execution referred to as "iteration" or "episode"
- Same initial and terminal state at each iteration
- Notation:

$$x_t^j = \text{system state at time } t \text{ of the } j\text{-th iteration}$$

Iterative Tasks - Problem Setup

- One task execution referred to as “iteration” or “episode”
- Same initial and terminal state at each iteration
- Notation:

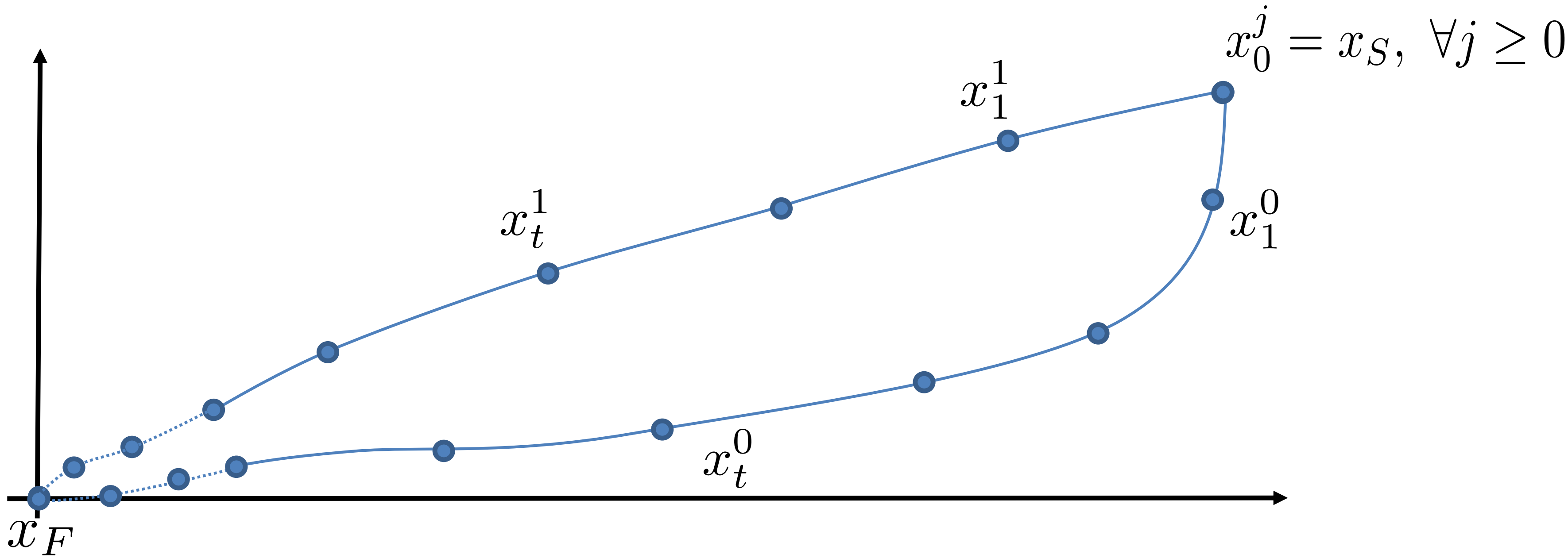
$$x_t^j = \text{system state at time } t \text{ of the } j\text{-th iteration}$$



Iterative Tasks - Problem Setup

- One task execution referred to as “iteration” or “episode”
- Same initial and terminal state at each iteration
- Notation:

$$x_t^j = \text{system state at time } t \text{ of the } j\text{-th iteration}$$



Iterative Learning MPC
Incorporating data in advanced model based controller

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

s.t.

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

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

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

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

Learned from data

Goal

- **Safety guarantees:**

  Constraint satisfaction at iteration j → satisfaction at iteration j+1
- **Performance improvement guarantees:**

  Closed loop cost at iteration j+1 ≤cost at iteration j

Learning MPC
Incorporating data in advance model based controller

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

s.t.

$$x_{k+1|t} = f(x_{k|t},u_{k|t}),\ \forall k \in [t,\cdots,t+N-1]$$
$$x_{t|t} = x^j_t,$$
$$x_{k|t} \in \mathcal{X},\ u_{k|t} \in \mathcal{U},\ \forall k \in [t,\cdots,t+N-1]$$
$$x_{t+N|t} \in \boxed{\mathcal{SS}^{j-1}},$$

Learned from data

Simplification (general case later)

- Known/nominal model
- Infinite Horizon Task
- Uncertainty and model adaptation later (and at this conference)

Learning Model Predictive Control (LMPC)

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

s.t.

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

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

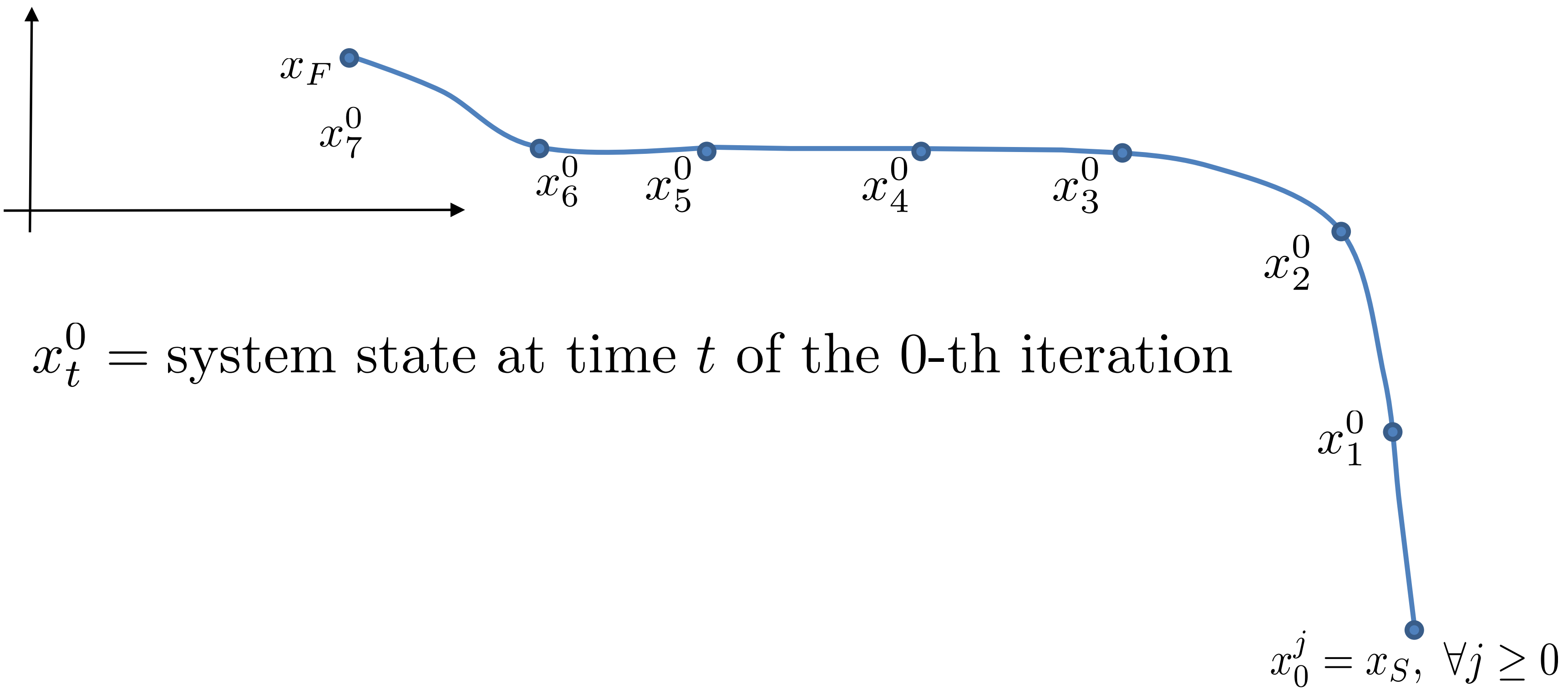
$$x_{k|t} \in \mathcal{X},\ u_{k|t} \in \mathcal{U},\ \forall k \in [t, \cdots, t+N-1]$$

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

- **Recursive feasibility**
- **Iterative feasibility**

Iteration 0

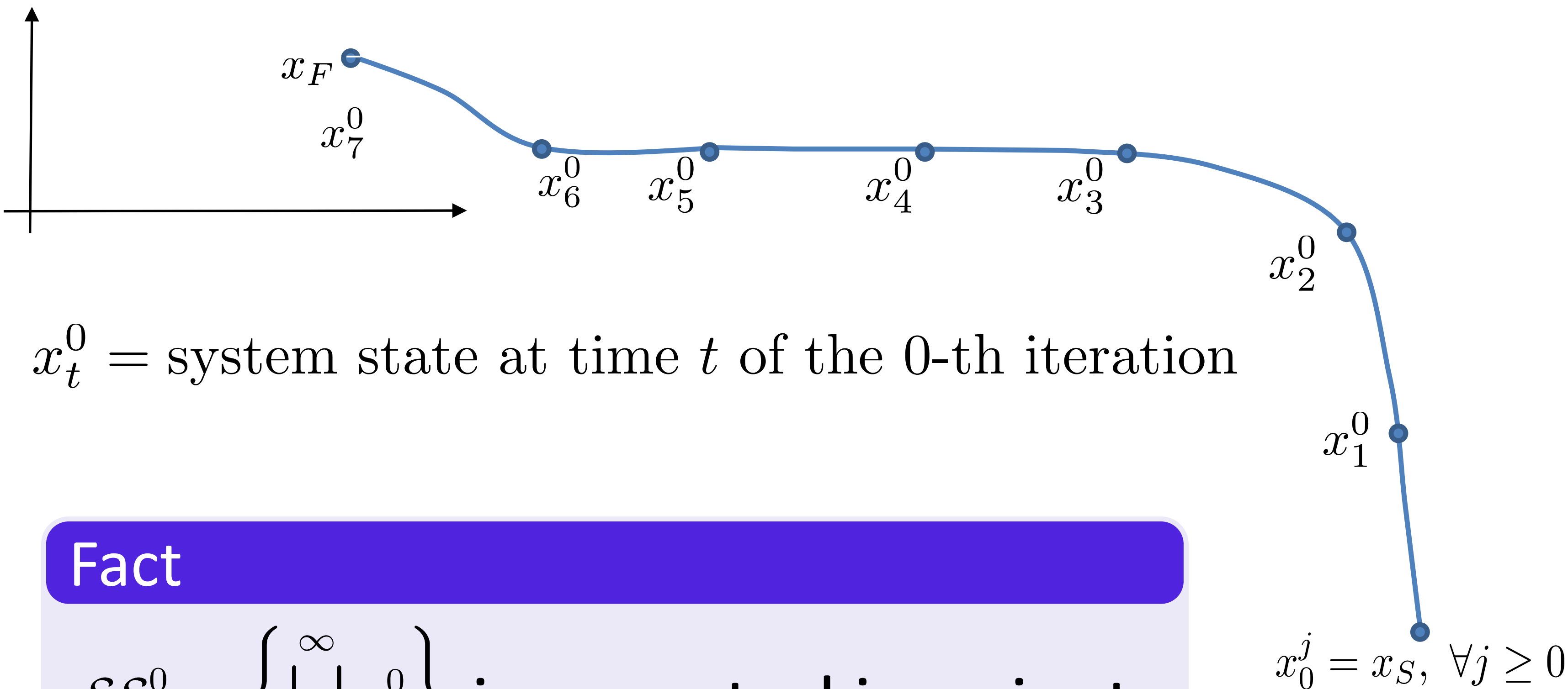
Assume at iteration 0 the closed-loop trajectory is feasible



$x_t^0 =$ system state at time t of the 0-th iteration

Iteration 0

Assume at iteration 0 the closed-loop trajectory is feasible



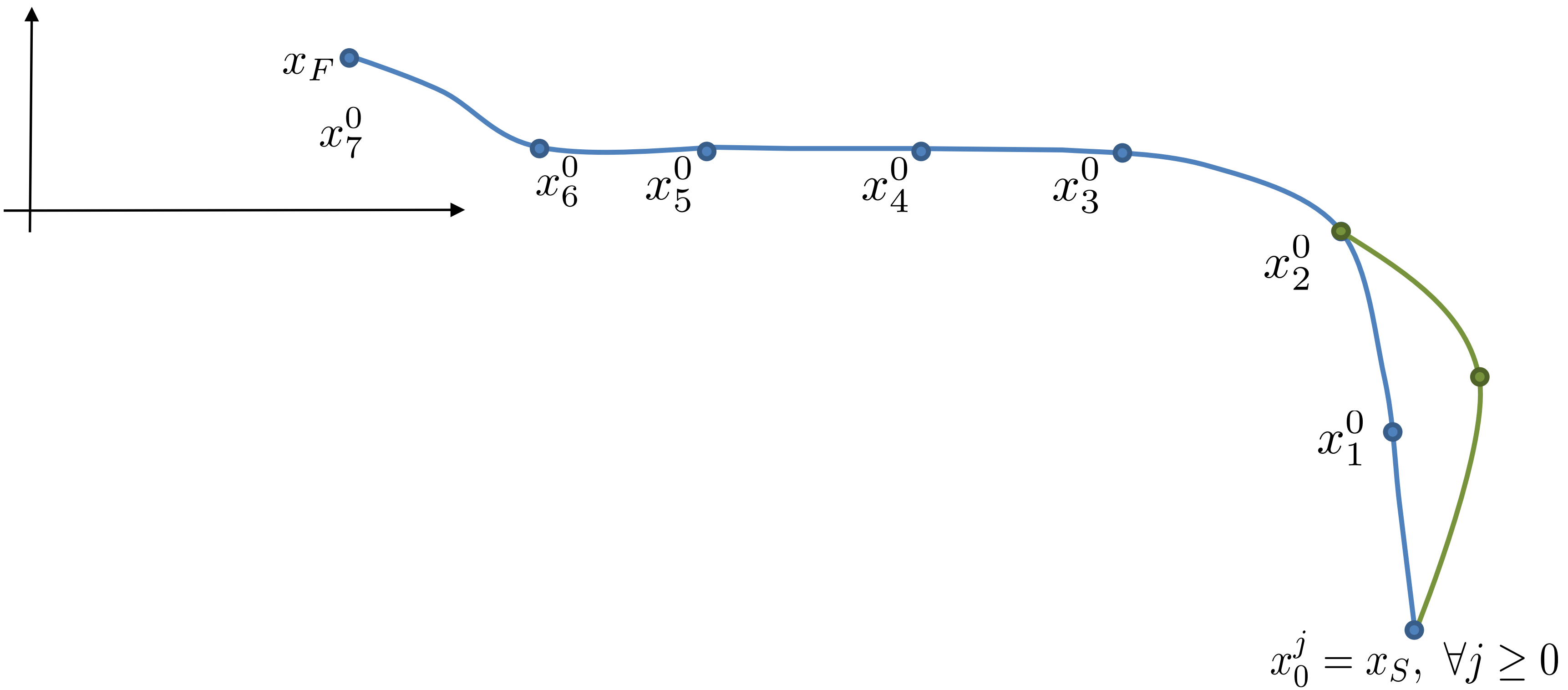
$$x_t^0 = \text{system state at time } t \text{ of the 0-th iteration}$$

Fact

$\mathcal{SS}^0 = \left\{ \bigcup_{t=0}^{\infty} x_t^0 \right\}$ is a control invariant

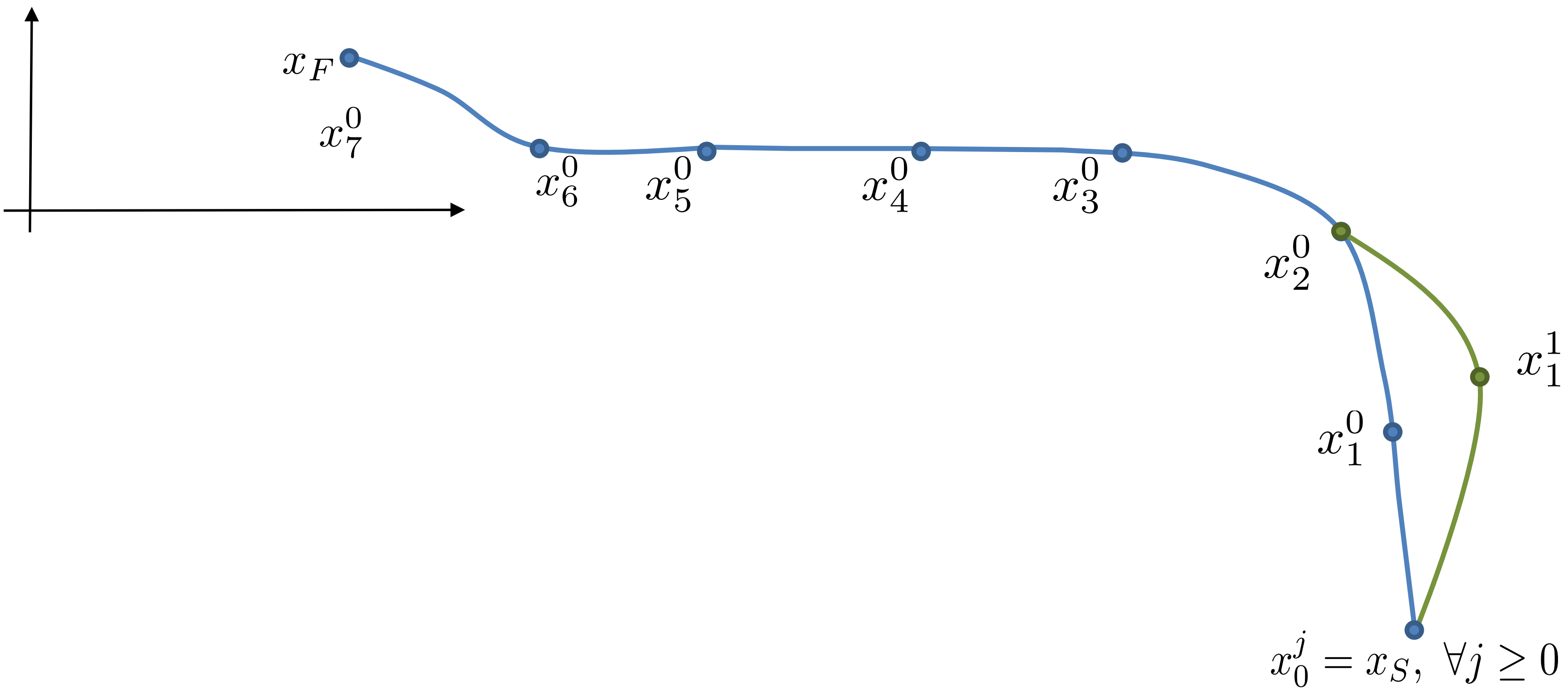
Iteration 1, Step 0

Use SS^0 as terminal set at Iteration 1



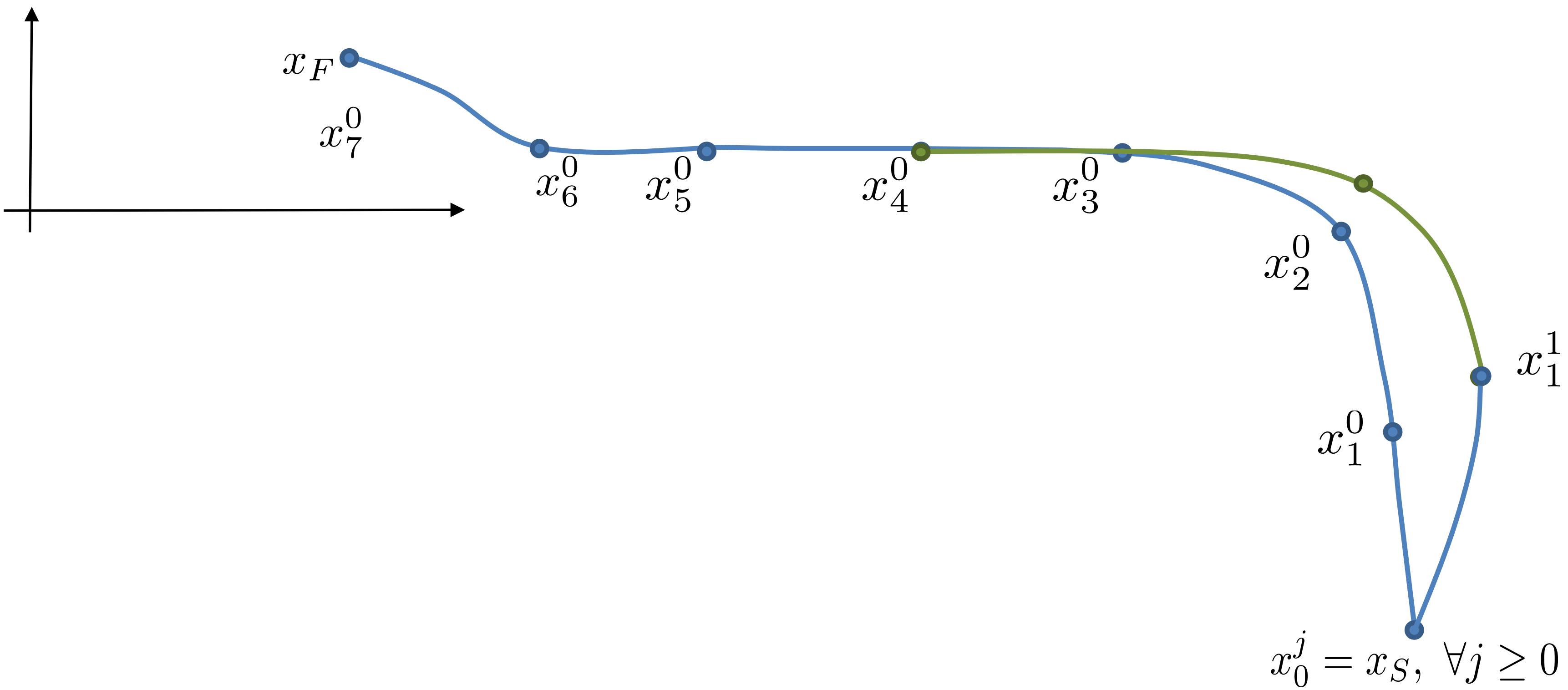
Iteration 1, Step 0

Use SS^0 as terminal set at Iteration 1



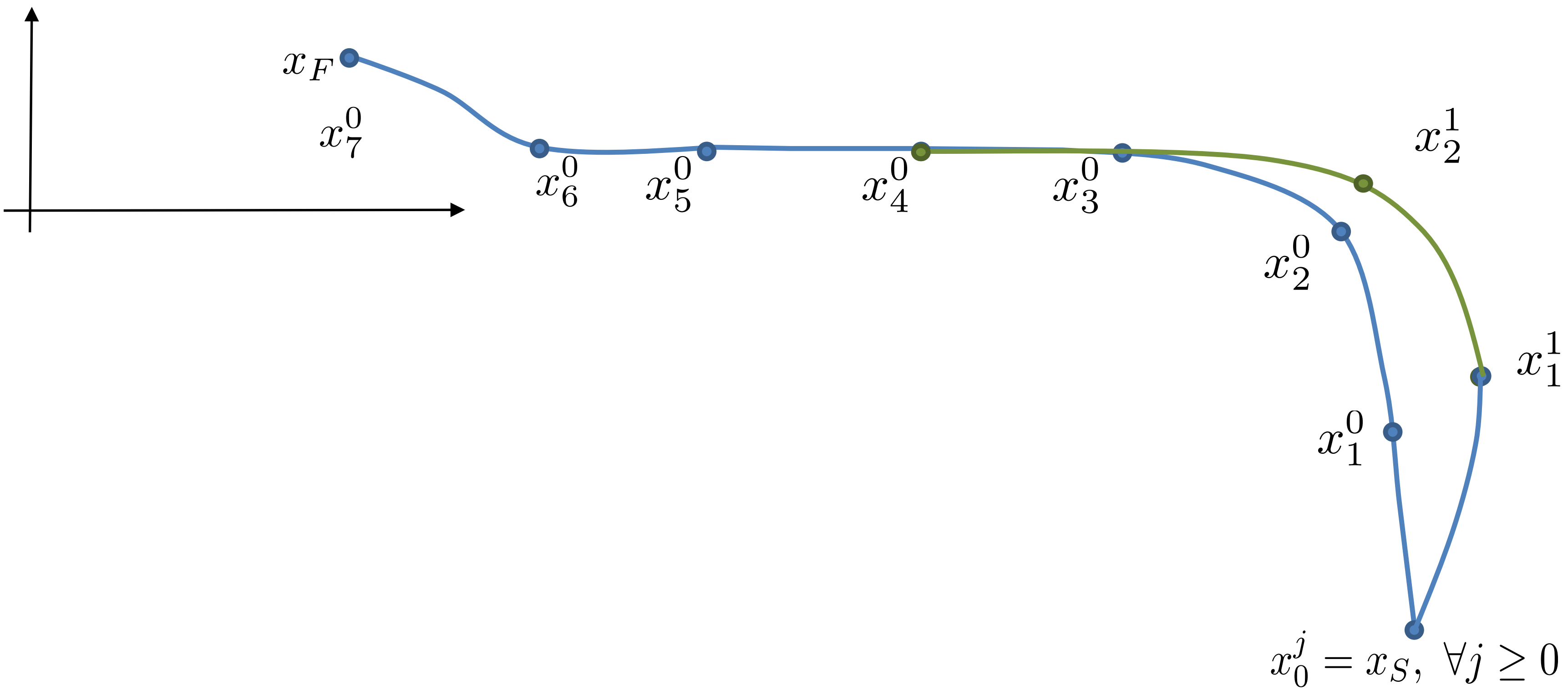
Iteration 1, Step 1

Use SS^0 as terminal set at Iteration 1



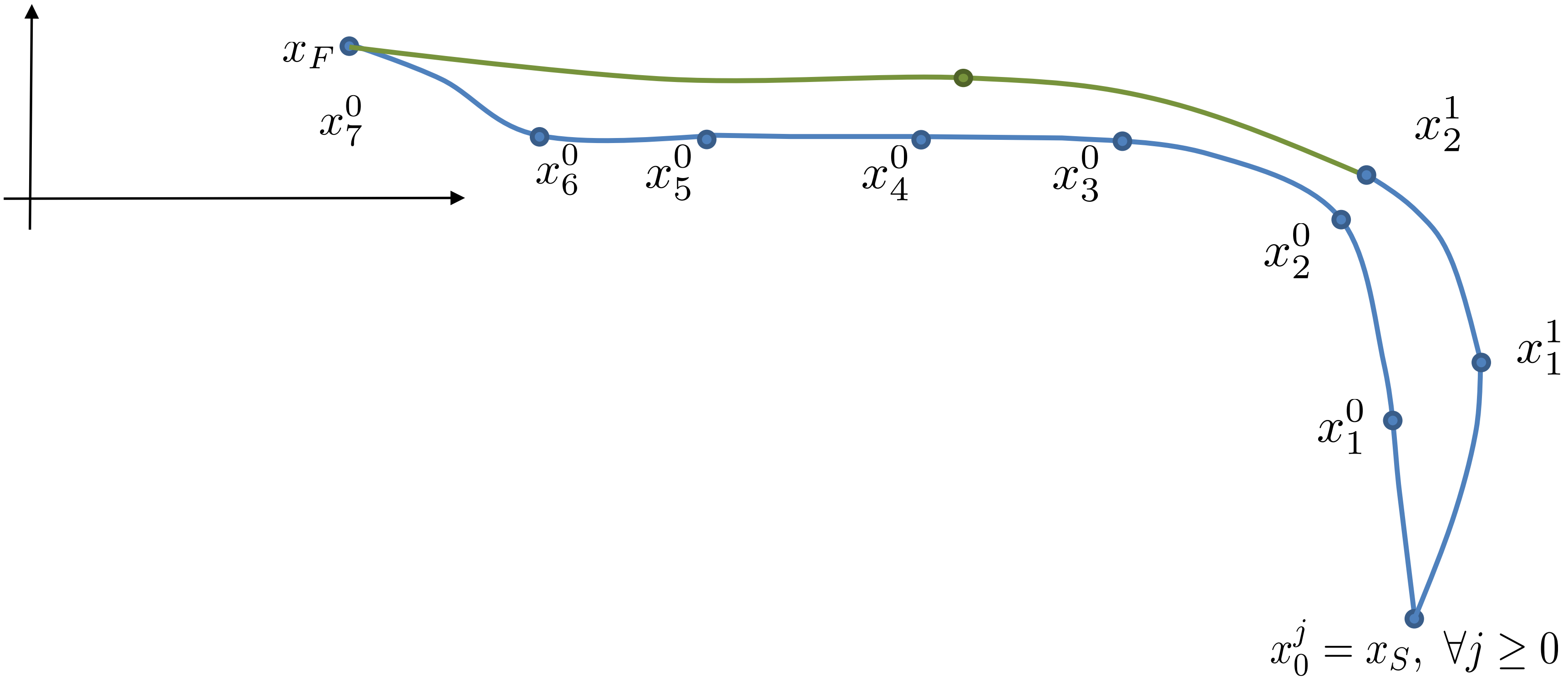
Iteration 1, Step 1

Use SS^0 as terminal set at Iteration 1



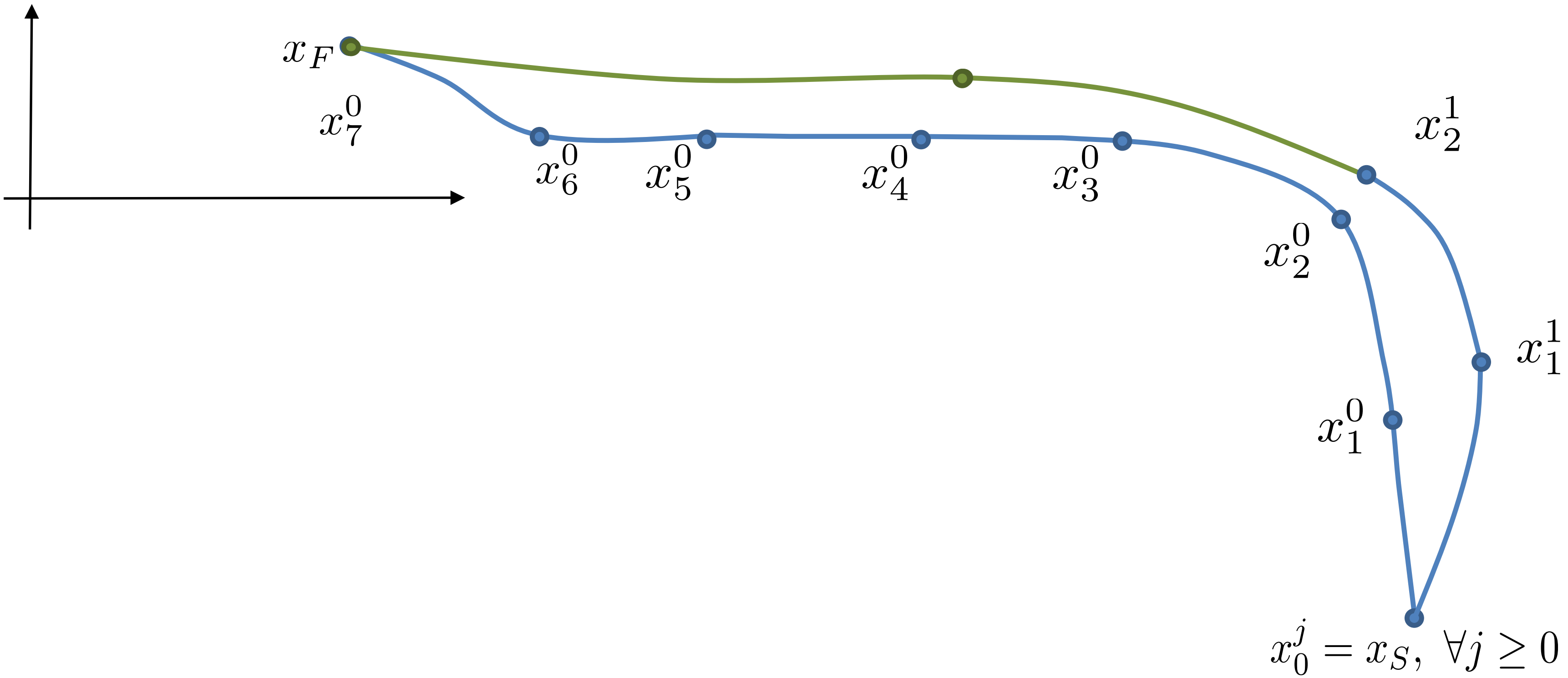
Iteration 1, Step 2

Use SS^0 as terminal set at Iteration 1



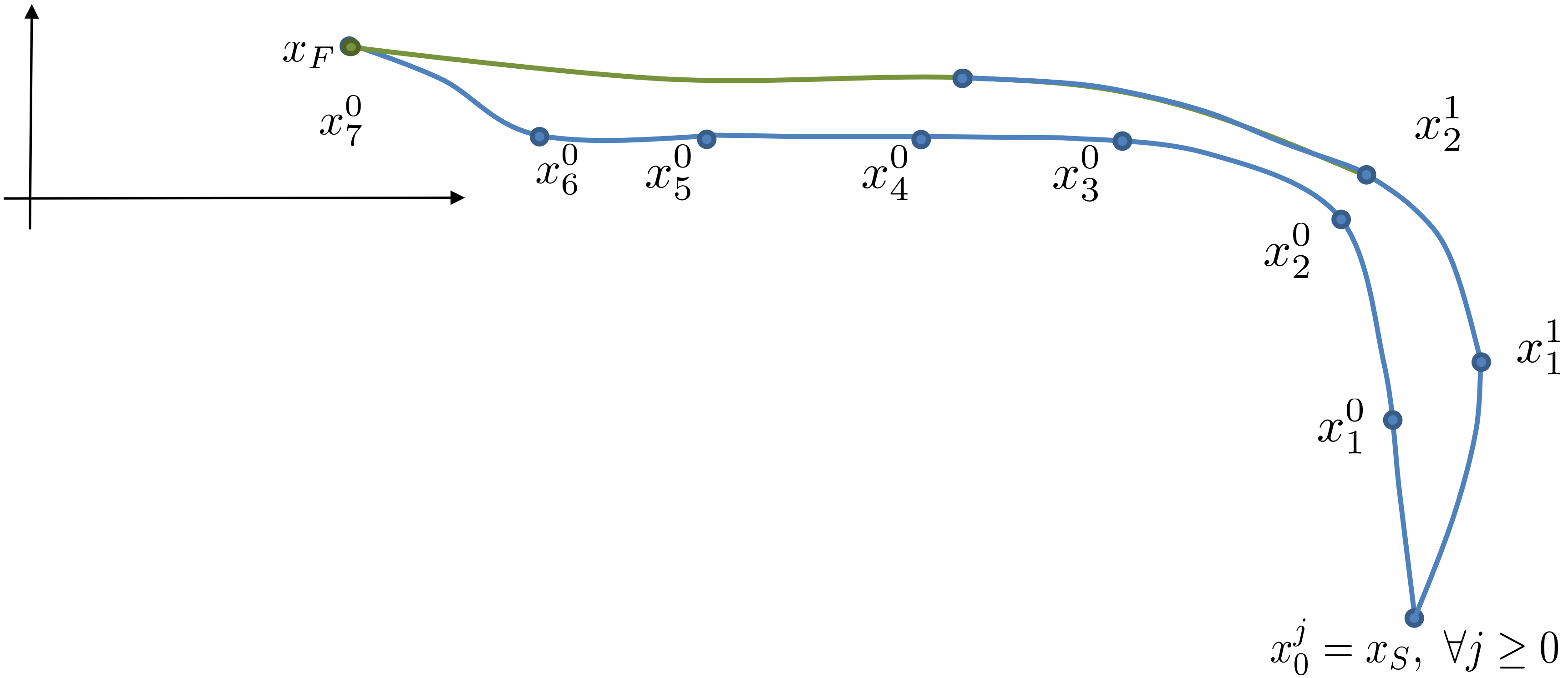
Iteration 1, Step 2

Use SS^0 as terminal set at Iteration 1



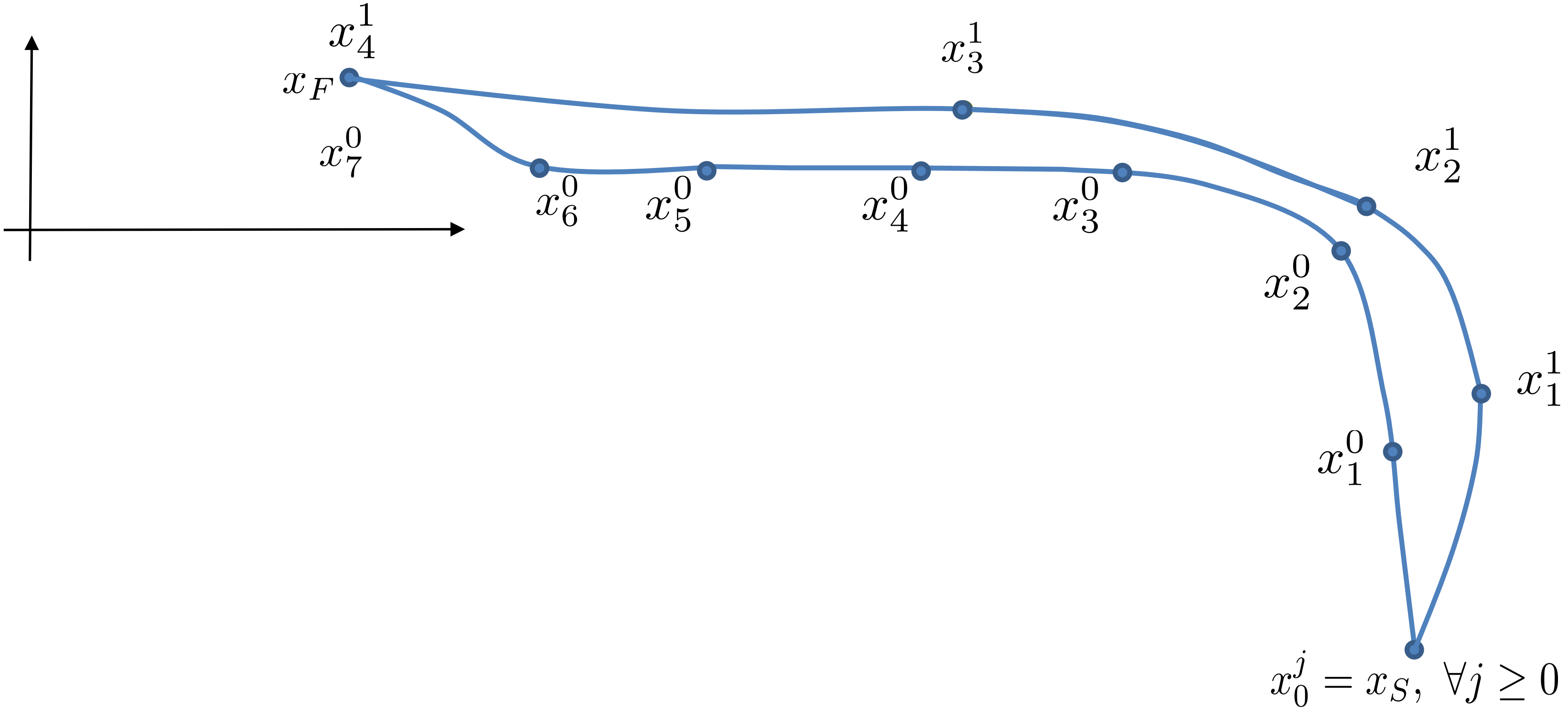
Iteration 1, Step 3

Use SS^0 as terminal set at Iteration 1

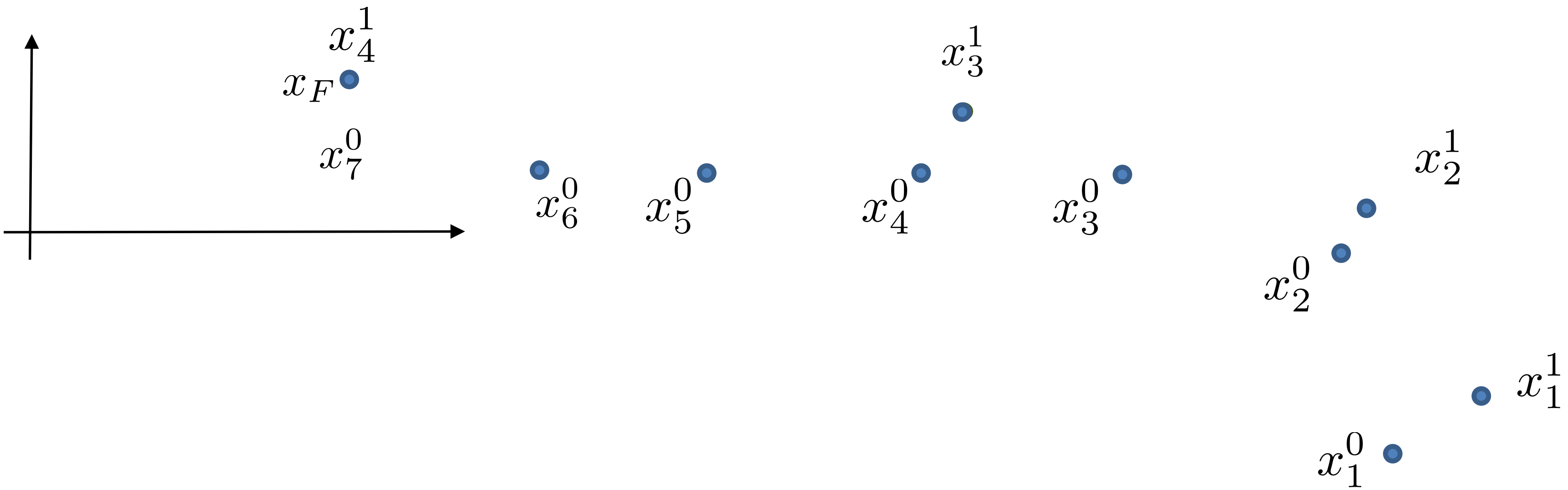


Iteration 1, Step 4

Use SS^0 as terminal set at Iteration 1



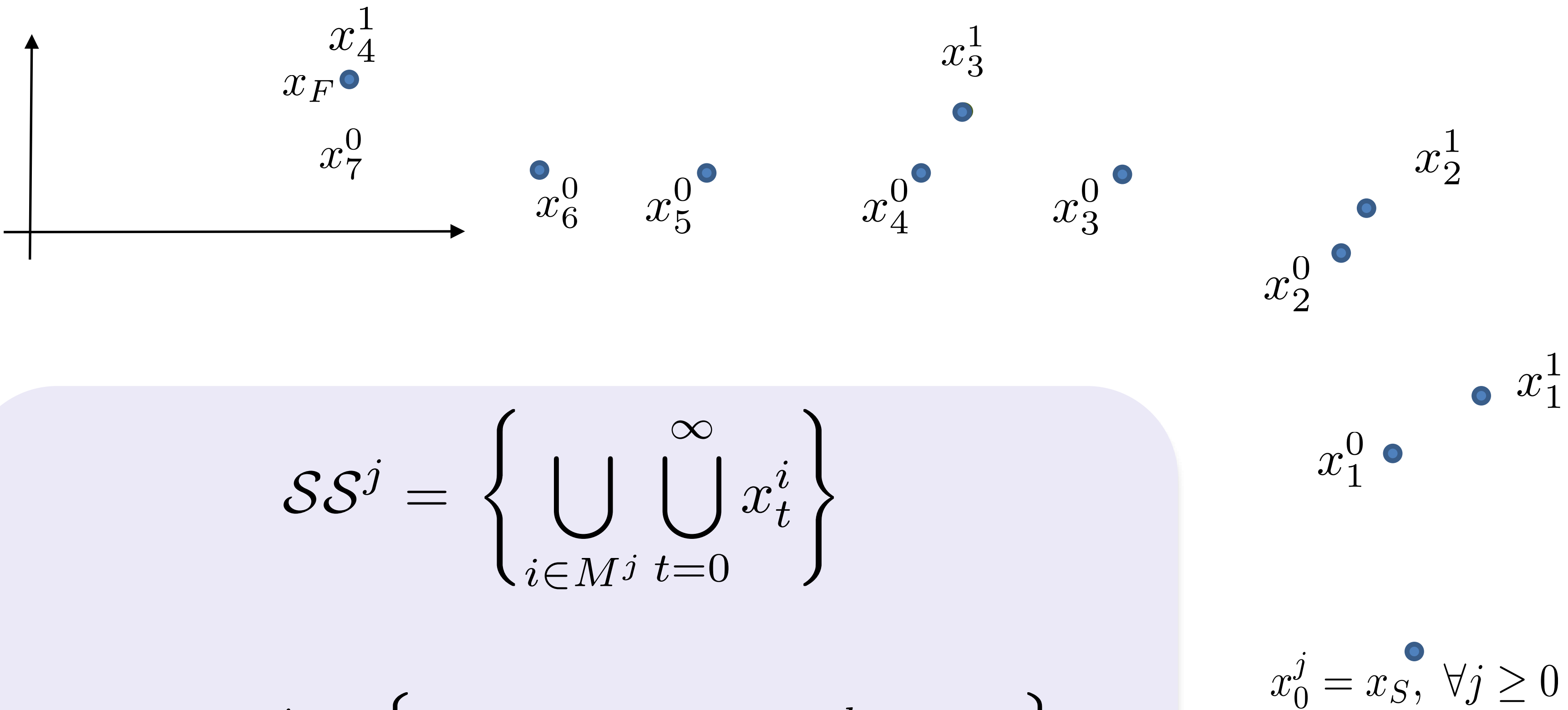
Iteration 2 Safe Set



$$\mathcal{SS}^0 = \left\{ \bigcup_{t=0}^{\infty} x_t^0 \right\} \quad \mathcal{SS}^1 = \left\{ \bigcup_{j=0}^{1} \bigcup_{t=0}^{\infty} x_t^j \right\} \quad \mathcal{SS}^1 \supset \mathcal{SS}^0$$

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

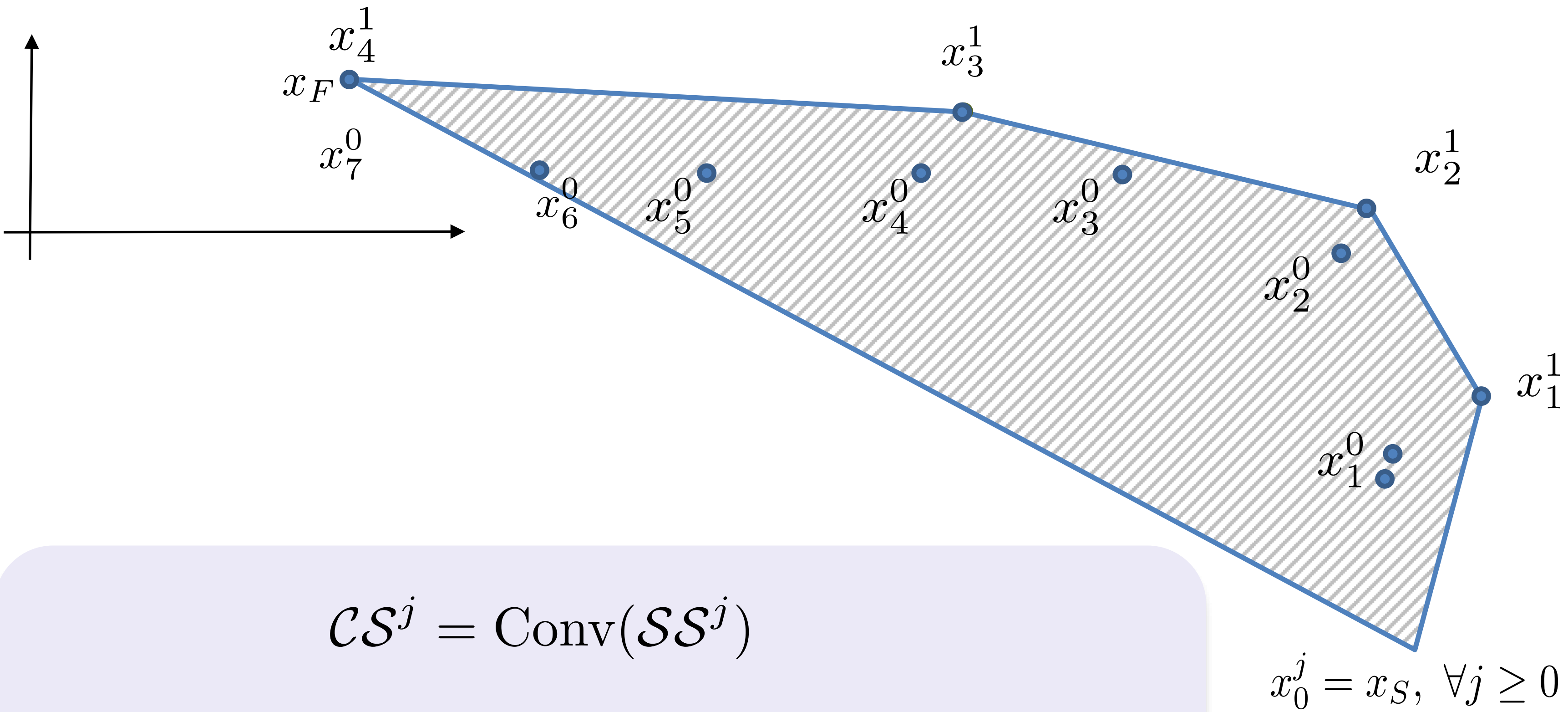
Constructing the terminal set



$$\mathcal{SS}^j = \left\{ \bigcup_{i \in M^j} \bigcup_{t=0}^{\infty} x_t^i \right\}$$

$$M^j = \left\{ k \in [0, j] : \lim_{t \to \infty} x_t^k = x_F \right\}$$

Terminal Set : Convex all of Sample Safe Set



$$\mathcal{CS}^j = \mathrm{Conv}(\mathcal{SS}^j)$$

for Constrained Linear Dynamical Systems
is a Control Invariant Set

Learning Model Predictive Control (LMPC)

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

s.t.

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

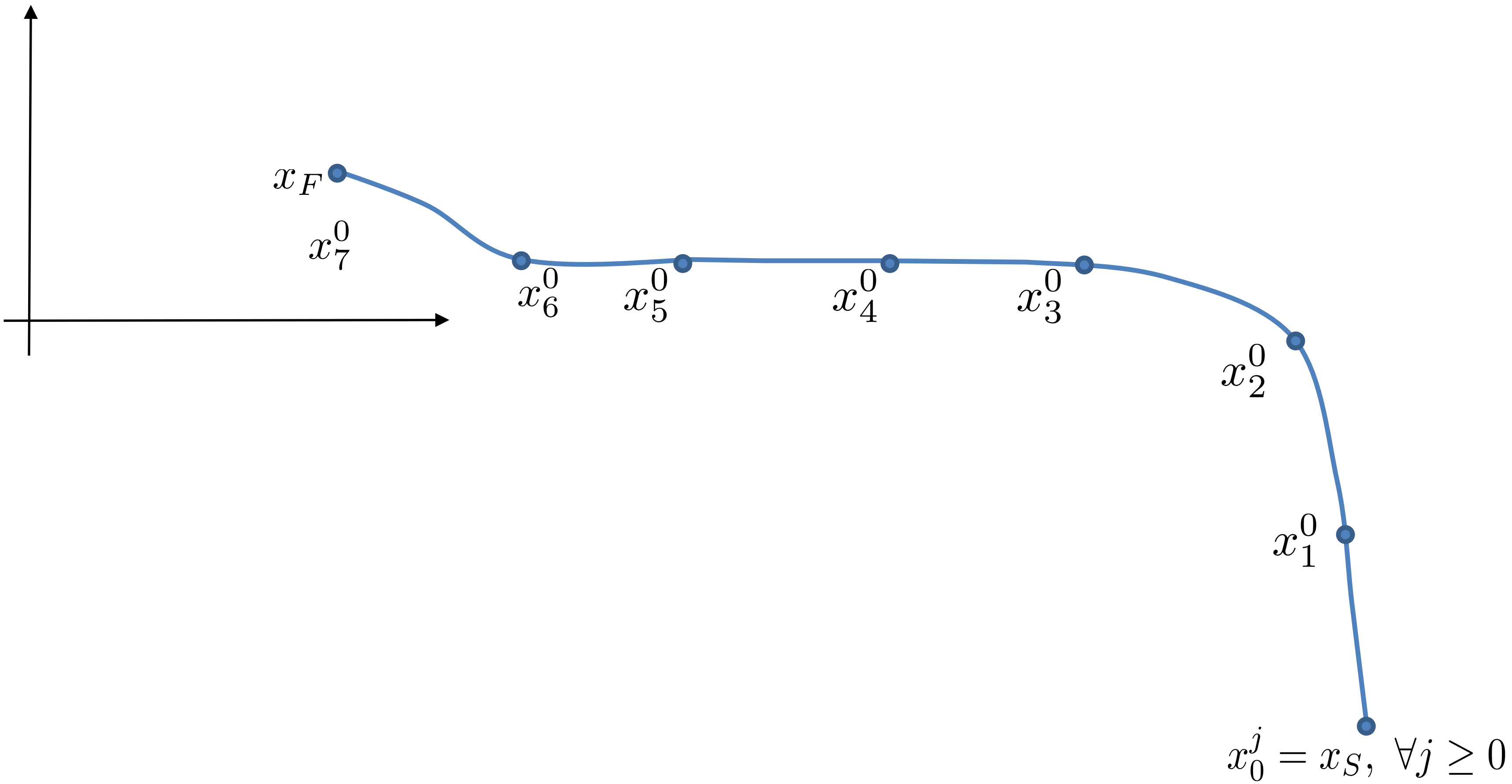
$$x_{t|t} = x^j_t,$$

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

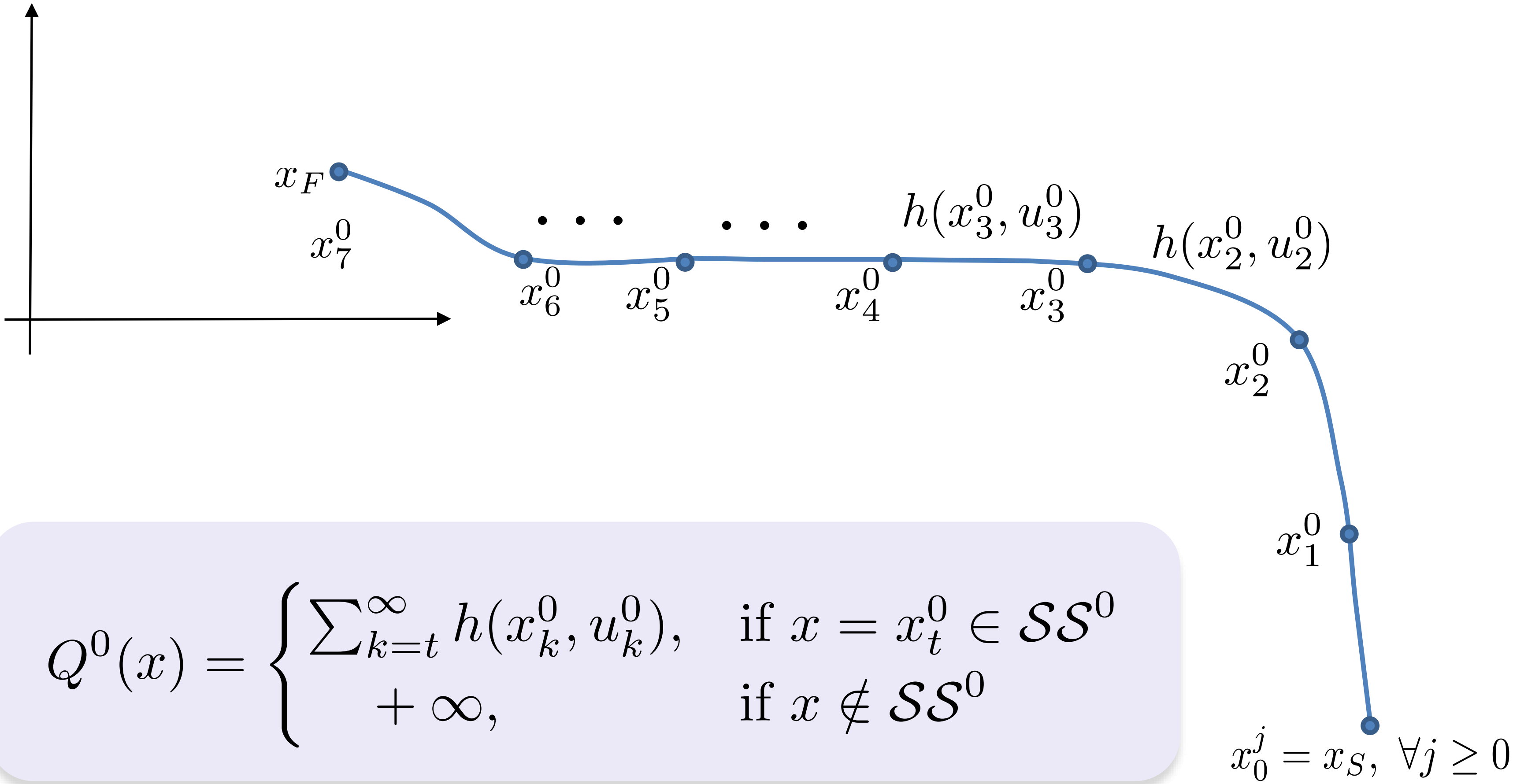
$$x_{t+N|t} \in \ \mathcal{SS}^{j-1},$$

- **Convergence**
- **Performance improvement**
- **Local optimality**

Terminal Cost at Iteration 0



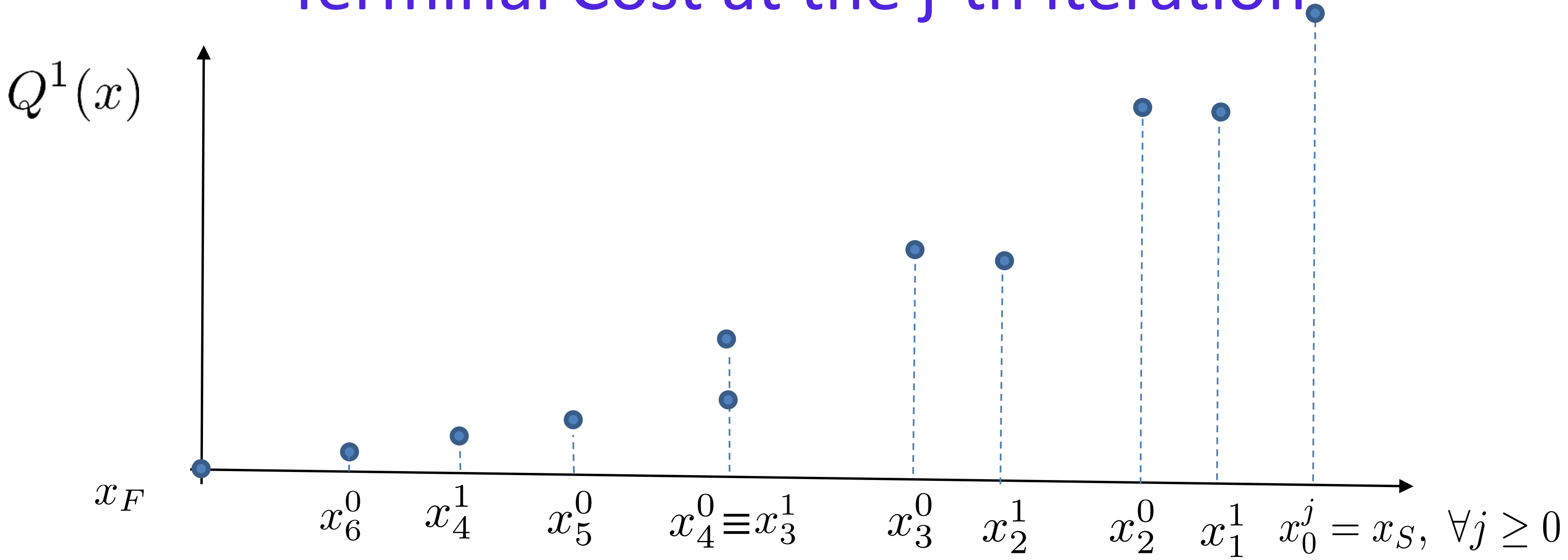
Terminal Cost at Iteration 0



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

A control Lyapunov "function"

Terminal Cost at the j-th iteration



Define

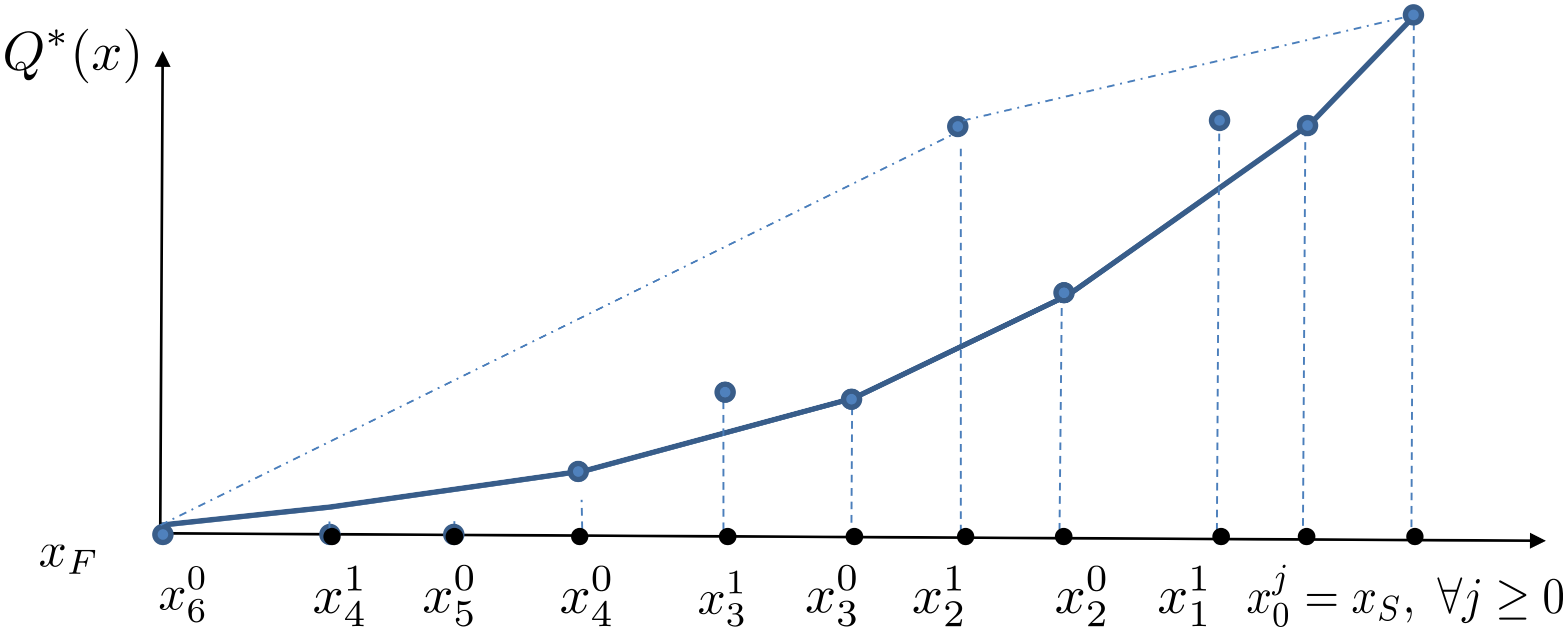
$$J_{t\to\infty}^j(x_t^j) = \sum_{k=t}^{\infty} h(x_k^j, u_k^j),$$

Compute terminal cost as

$$Q^j(x) = \begin{cases} \min\limits_{(i,t)\in F^j(x)} J_{t\to\infty}^i(x), & \text{if } x \in \mathcal{SS}^j \\ +\infty, & \text{if } x \notin \mathcal{SS}^j \end{cases}$$

$$F^j(x) = \left\{(i,t) : i \in [0,j], \ t \geq 0 \text{ with } x = x_t^i; \ \text{for } x_t^i \in \mathcal{SS}^j\right\}$$

Terminal Cost: Barycentric Approximation of Q()



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

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

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

Control Lyapunov Function

(for Constrained Linear Dynamical Systems)

ILMPC Summary

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

s.t.

$$x_{k+1|t} = A_{k|t}x_{k|t} + B_{k|t}u_{k|t} + C_{k|t}, \ \forall k \in [t, \cdots, t+N-1]$$

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

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

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

MPC strategy: $u^j_t = u^{*,j}_{t|t}(x^j_t)$

- Optimize over inputs and lambdas
- For constrained linear systems
  - **Safety guarantees:**
    - Constraint satisfaction at iteration j=> satisfaction at iteration j+1
  - **Performance improvement guarantees:**
    - Closed loop cost at iteration j >= cost at iteration j+1
  - Convergence to global optimal solution
  - Constraint qualification conditions required for cost decrease

Performance Improvement Proof

- Conjecture

$$J^{j-1}_{0\rightarrow\infty}(x^{j-1}_0) \geq J^{j}_{0\rightarrow\infty}(x^{j}_0)$$

- Notation

$$\mathbf{x}^j \;=\; [x^j_0,\; x^j_1,\; ...,\; x^j_t,\; ...] \qquad \mathbf{u}^j \;=\; [u^j_0,\; u^j_1,\; ...,\; u^j_t,\; ...]$$

Closed-loop state and input trajectory at iteration *j*

Performance Improvement Proof

Step 1: $J_{0\rightarrow\infty}^{j-1}(x_0^{j-1}) \geq J_{0\rightarrow N}^{LMPC,j}(x_0^j)$

$$J_{0\rightarrow\infty}^{j-1}(x_0^{j-1}) = \sum_{k=0}^{\infty} h(x_k^{j-1}, u_k^{j-1}) =$$

Performance Improvement Proof

Step 1: $J^{j-1}_{0\rightarrow\infty}(x^{j-1}_0) \geq J^{LMPC,j}_{0\rightarrow N}(x^j_0)$

$$J^{j-1}_{0\rightarrow\infty}(x^{j-1}_0) = \sum_{k=0}^{\infty} h(x^{j-1}_k, u^{j-1}_k) = \sum_{k=0}^{N-1} h(x^{j-1}_k, u^{j-1}_k) + \sum_{k=N}^{\infty} h(x^{j-1}_k, u^{j-1}_k)$$

Performance Improvement Proof

Step 1: $J_{0\to\infty}^{j-1}(x_0^{j-1}) \geq J_{0\to N}^{LMPC,j}(x_0^j)$

$$J_{0\to\infty}^{j-1}(x_0^{j-1}) = \sum_{k=0}^{\infty} h(x_k^{j-1}, u_k^{j-1}) = \sum_{k=0}^{N-1} h(x_k^{j-1}, u_k^{j-1}) + \underbrace{\sum_{k=N}^{\infty} h(x_k^{j-1}, u_k^{j-1})}_{Q^{j-1}(x_N^{j-1})}$$

Performance Improvement Proof

Step 1: $J_{0\to\infty}^{j-1}(x_0^{j-1}) \geq J_{0\to N}^{LMPC,j}(x_0^j)$

$$J_{0\to\infty}^{j-1}(x_0^{j-1}) = \sum_{k=0}^{\infty} h(x_k^{j-1}, u_k^{j-1}) = \sum_{k=0}^{N-1} h(x_k^{j-1}, u_k^{j-1}) + \underbrace{\sum_{k=N}^{\infty} h(x_k^{j-1}, u_k^{j-1})}_{Q^{j-1}(x_N^{j-1})}$$

$$\rightarrow \quad J_{0\to\infty}^{j-1}(x_0^{j-1}) = \sum_{k=0}^{N-1} h(x_k^{j-1}, u_k^{j-1}) + Q^{j-1}(x_N^{j-1}) \geq J_{0\to N}^{LMPC,j}(x_0^j)$$

Performance Improvement Proof

Step 1: $J^{j-1}_{0\rightarrow\infty}(x^{j-1}_0) \geq J^{LMPC,j}_{0\rightarrow N}(x^j_0)$

Step 2: $J^{LMPC,j}_{0\rightarrow N}(x^j_0) \geq J^{j}_{0\rightarrow\infty}(x^j_0)$

$$J^{LMPC,j}_{1\rightarrow 1+N}(x^j_1) - J^{LMPC,j}_{0\rightarrow N}(x^j_0) \leq -h(x^j_0, u^j_0)$$

Performance Improvement Proof

Step 1: $J^{j-1}_{0\rightarrow\infty}(x^{j-1}_0) \geq J^{LMPC,j}_{0\rightarrow N}(x^j_0)$

Step 2: $J^{LMPC,j}_{0\rightarrow N}(x^j_0) \geq J^{j}_{0\rightarrow\infty}(x^j_0)$

$$J^{LMPC,j}_{1\rightarrow 1+N}(x^j_1) - J^{LMPC,j}_{0\rightarrow N}(x^j_0) \leq -h(x^j_0, u^j_0)$$

$$\rightarrow\ J^{LMPC,j}_{0\rightarrow N}(x^j_0) \geq J^{LMPC,j}_{1\rightarrow 1+N}(x^j_1) + h(x^j_0, u^j_0) \geq J^{LMPC,j}_{2\rightarrow 2+N}(x^j_2) + h(x^j_0, u^j_0) + h(x^j_1, u^j_1)$$

Performance Improvement Proof

Step 1: $J^{j-1}_{0\to\infty}(x^{j-1}_0) \geq J^{LMPC,j}_{0\to N}(x^j_0)$

Step 2: $J^{LMPC,j}_{0\to N}(x^j_0) \geq J^{j}_{0\to\infty}(x^j_0)$

$$J^{LMPC,j}_{1\to 1+N}(x^j_1) - J^{LMPC,j}_{0\to N}(x^j_0) \leq -h(x^j_0, u^j_0)$$

$$\to \ J^{LMPC,j}_{0\to N}(x^j_0) \geq J^{LMPC,j}_{1\to 1+N}(x^j_1) + h(x^j_0, u^j_0) \geq J^{LMPC,j}_{2\to 2+N}(x^j_2) + h(x^j_0, u^j_0) + h(x^j_1, u^j_1)$$

$$\to \ J^{LMPC,j}_{0\to N}(x^j_0) \geq \underbrace{\cancel{\lim_{t\to\infty} J^{LMPC,j}_{t\to t+N}(x^j_t)}}_{0} + \sum_{k=0}^{\infty} h(x^j_k, u^j_k)$$

Performance Improvement Proof

Conclusion: $J_{0\rightarrow\infty}^{j-1}(x_0^{j-1}) \geq J_{0\rightarrow N}^{LMPC,j}(x_0^j) \geq J_{0\rightarrow\infty}^{j}(x_0^j)$

The iteration cost $J_{0\rightarrow\infty}^{j}$ is non-increasing at each iteration

Iterative Learning MPC

- Optimize over inputs and lambdas
- Simple proofs
- For constrained linear systems
  - Safety and Performance improvement guarantees
  - Convergence to global optimal solution (for linear
  - Constraint qualification conditions required for cost decrease

$$x_N = A^N x_0 + [A^{N-1}B \dots B] \begin{bmatrix} u_0 \\ \vdots \\ u_{N-1} \end{bmatrix}$$

If full column rank, improvement cannot be obtained

Constrained LQR Example

$$\min_{u_0,u_1,\ldots} \boxed{\sum_{k=0}^{\infty} \left[||x_k||_2^2 + ||u_k||_2^2 \right]}$$

Control objective

$$\text{s.t.} \quad x_{k+1} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} x_k + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u_k, \ \forall k \geq 0$$

System dynamics
System constraints

$$x_k \in \text{box}[-4,4] \quad \forall k \geq 0$$

$$u_k \in [-1,1] \quad \forall k \geq 0$$

Starting Position

$$\boxed{x_0 = [-2.120,\ 0.066]^T,}$$

Iterative LMPC with horizon N=2

$$\min_{u_{0|t},u_{1|t}} \boxed{\sum_{k=0}^{2} \left[||x_{k|t}||_2^2 + ||u_{k|t}|_2^2\right] + Q^{j-1}(x_{2|t})}$$

Control objective

$$\text{s.t.} \quad \boxed{\begin{aligned} & x_{k+1|t} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} x_{k|t} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u_{k|t}, \ \forall k = [0,1] \\ & x_{k|t} \in \text{box}[-4,4] \quad \forall k = [0,1] \\ & u_{k|t} \in [-1,1] \qquad \forall k = [0,1] \end{aligned}}$$

System dynamics
System constraints

Terminal Constraint $\boxed{x_{2|t} \in \mathcal{CS}^{j-1}}$

Initial Condition $\boxed{x_{0|t} = x(t),}$

Will not work!

Will work if one sets N=3

Comparison with R.L.??

- RL term too broad
- Two good references:
  - Bertsekas paper connecting MPC and ADP*
  - Lewis and Vrabile survey on CSM**
  - Recht survey (section 6): https://arxiv.org/abs/1806.09460

- ILMPC highlights
  - Continuous state formulation
  - Constraints satisfaction and Sampled Safe Sets
  - Q-function constructed (learned) locally based on cost/model driven exploration and past trails
  - Q-function at stored state is “exact” and lowerbounds property at intermediate points (for convex problems)

*Dynamic Programming and Suboptimal Control: A Survey from ADP to MPC
**Reinforcement Learning and Adaptive Dynamic Programming for Feedback Control

About Model Learning in Racing

Autonomous Racing 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

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

Obstacle avoidance

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

Learning Model Predictive Control (LMPC)

$$\min_{u_{t|t},\ldots,u_{t+N-1|t}} \sum_{k=t}^{t+N-1} \left(\mathbb{1}_{x_{k|t} \in \mathcal{X}_F} \right) + \boxed{Q^{j-1}(x_{t+N|t})}$$

s.t.

$$x_{k+1|t} = A_{k|t} x_{k|t} + B_{k|t} u_{k|t} + C_{k|t}, \;\; \forall k \in [t, \cdots, t+N-1]$$

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

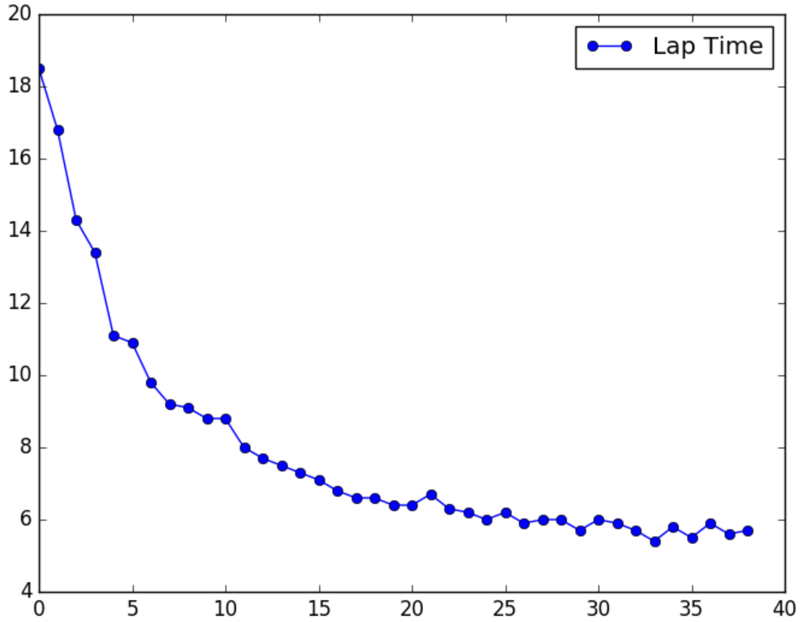
$$x_{k|t} \in \mathcal{X}, \;\; u_{k|t} \in \mathcal{U}, \;\; \forall k \in [t, \cdots, t+N-1]$$

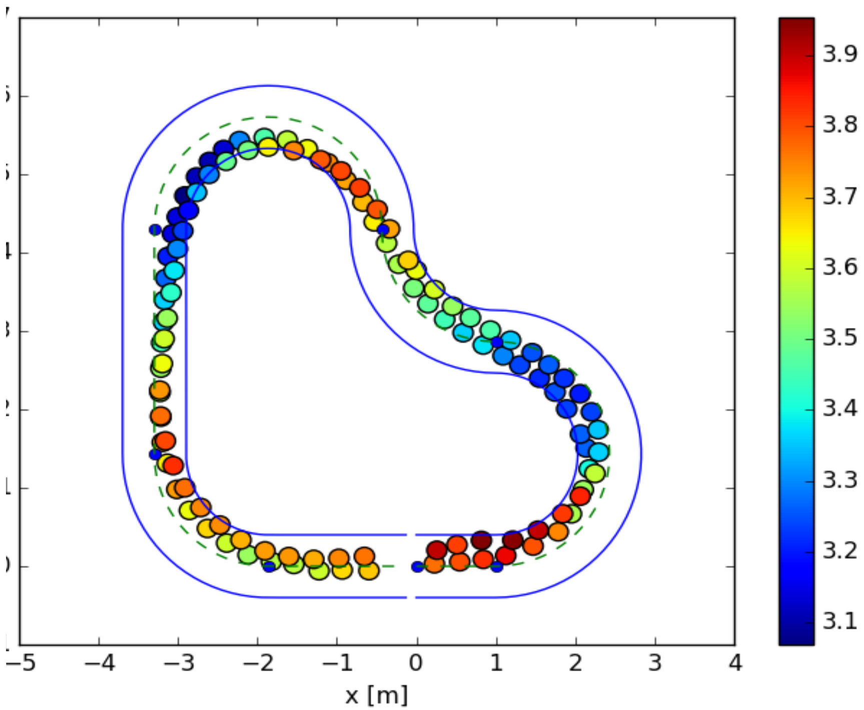
$$x_{t+N|t} \in \boxed{\mathcal{CS}^{j-1}},$$

Receding Horizon Strategy:

$$u_t^j = u_0^*(x_t^j)$$

Learning Process





The lap time decreases until the LMPC converges to a set of trajectories

Learning Model Predictive Control (LMPC)

$$\min_{u_{t|t},\ldots,u_{t+N-1|t}} \sum_{k=t}^{t+N-1} \left(\mathbb{1}_{x_{k|t} \in \mathcal{X}_F} \right) + Q^{j-1}(x_{t+N|t})$$

s.t.

$$\begin{aligned}
& x_{k+1|t} = \boxed{A_{k|t} x_{k|t} + B_{k|t} u_{k|t} + C_{k|t},} \ \forall k \in [t, \cdots, t+N-1] \\
& x_{t|t} = x_t^j, \\
& x_{k|t} \in \mathcal{X}, \ u_{k|t} \in \mathcal{U}, \ \forall k \in [t, \cdots, t+N-1] \\
& x_{t+N|t} \in \ \mathcal{CS}^{j-1},
\end{aligned}$$

Receding Horizon Strategy:

$$u_t^j = u_0^*(x_t^j)$$

Useful Vehicle Model Abstraction

- **Nonlinear Dynamical System**

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

Useful Vehicle Model Abstraction

- 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

Useful Vehicle Model Abstraction

- **Nonlinear Dynamical System**

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

Kinematic Equations

- **Identifying the Dynamical System**

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

Linearization around predicted trajectory

Useful Vehicle Model Abstraction

- **Nonlinear Dynamical System**

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

Dynamic Equations

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

Kinematic Equations

- **Identifying the Dynamical System**

Local Linear Regression

$$z_{k+1|t} = \begin{bmatrix} \dot{x}_{k+1|t} \\ \dot{y}_{k+1|t} \\ \ddot{\psi} \\ \psi_{k+1|t} \\ X_{k+1|t} \\ Y_{k+1|t} \end{bmatrix} = \left[\begin{array}{c|c} \multicolumn{2}{c}{\arg\min \sum_i K(z_{k|t} - z_i)\|\Lambda_y \begin{bmatrix} z_{k|t} \\ u_{k|t} \\ 1 \end{bmatrix} - y_{i+1}\|, \forall y \in \{\dot{x}, \dot{y}, \ddot{\psi}\}} \\ \begin{matrix} \text{Linearized Kinematics} \\ \text{Linearized Kinematics} \\ \text{Linearized Kinematics} \end{matrix} & \begin{matrix} \text{Linearized Kinematics} \\ \text{Linearized Kinematics} \\ \text{Linearized Kinematics} \end{matrix} \end{array} \right] \begin{matrix} z_{k|t} + \\ \begin{bmatrix} u_{k|t} \\ 1 \end{bmatrix} \end{matrix}$$

Linearization around predicted trajectory

Useful Vehicle Model Abstraction

- **Identifying the Dynamical System**

Local Linear Regression

$$z_{k+1|t} = \begin{bmatrix} \dot{x}_{k+1|t} \\ \dot{y}_{k+1|t} \\ \ddot{\psi} \\ \psi_{k+1|t} \\ X_{k+1|t} \\ Y_{k+1|t} \end{bmatrix} = \begin{bmatrix} \arg\min \sum_i K(z_{k|t} - z_i) \left\| \Lambda_y \begin{bmatrix} z_{k|t} \\ u_{k|t} \\ 1 \end{bmatrix} - y_{i+1} \right\|, \forall y \in \{\dot{x}, \dot{y}, \ddot{\psi}\} & \\ \begin{matrix} \text{Linearized Kinematics} \\ \text{Linearized Kinematics} \\ \text{Linearized Kinematics} \end{matrix} \; z_{k|t} + & \begin{matrix} \text{Linearized Kinematics} \\ \text{Linearized Kinematics} \\ \text{Linearized Kinematics} \end{matrix} \end{bmatrix} \begin{bmatrix} u_{k|t} \\ 1 \end{bmatrix}$$

Linearization around predicted trajectory

- **Important Design Steps**
  1. Compute **trajectory to linearize around** uses previous optimal inputs and inputs in the safe set
  2. Enforce model-based **sparsity** in local linear regression

Useful Vehicle Model Abstraction

- Nonlinear **Dynamical System**

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

The velocity update is not affected by **Position** and **Acceleration** command

$$\Lambda_{\dot{x}} = \begin{bmatrix}\lambda_1 & \lambda_2 & \lambda_3 & \boxed{0 \quad 0 \quad 0} & \lambda_4 & \boxed{0} & \lambda_5\end{bmatrix}$$

Useful Vehicle Model Abstraction

- **Identifying the Dynamical System**

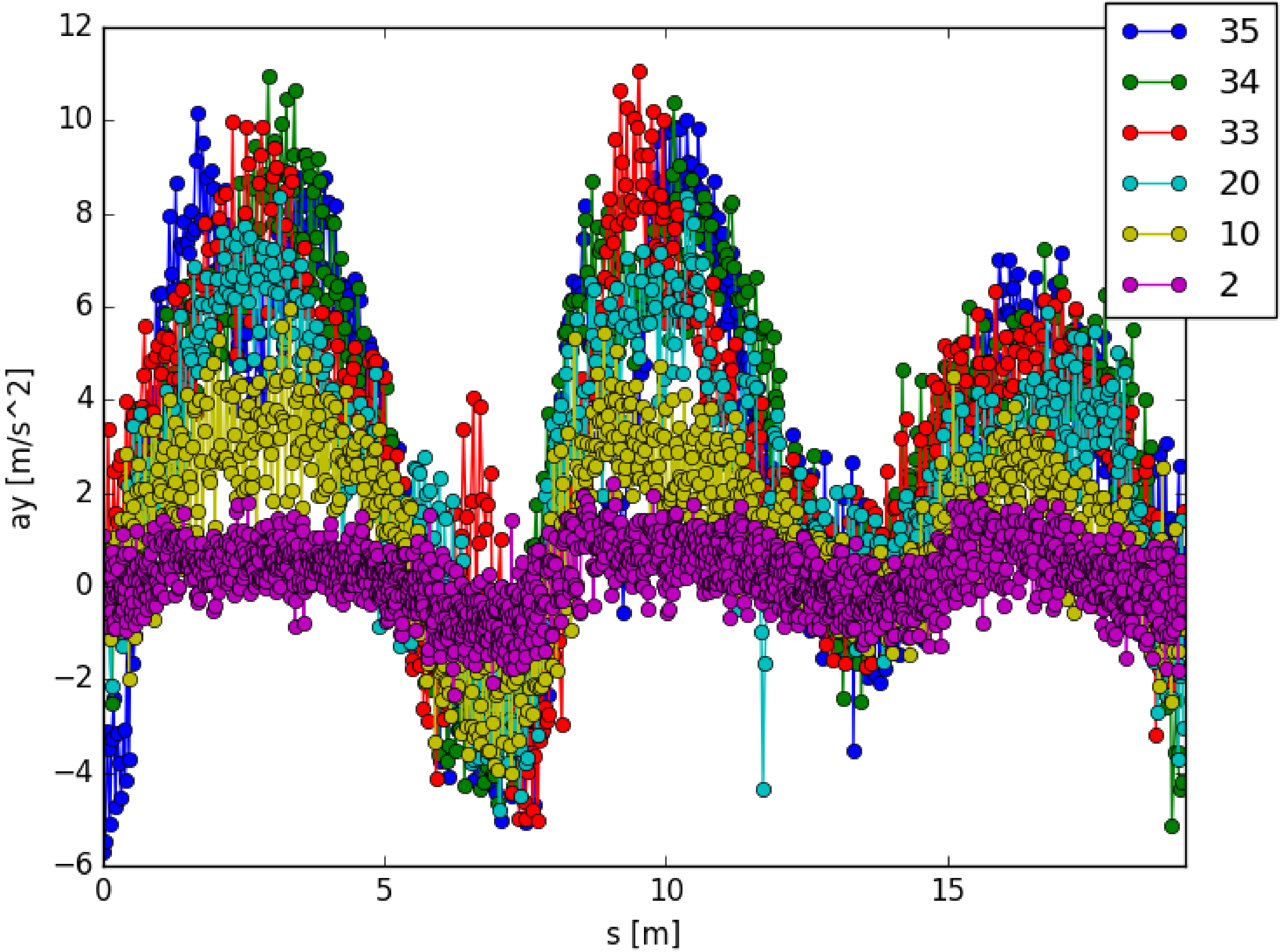
Local Linear Regression

$$z_{k+1|t} = \begin{bmatrix} \dot{x}_{k+1|t} \\ \dot{y}_{k+1|t} \\ \ddot{\psi} \\ \psi_{k+1|t} \\ X_{k+1|t} \\ Y_{k+1|t} \end{bmatrix} = \left[\begin{array}{ccc} \multicolumn{3}{c}{\boxed{\arg\min \sum_i K(z_{k|t} - z_i) ||\Lambda_y \begin{bmatrix} z_{k|t} \\ u_{k|t} \\ 1 \end{bmatrix} - y_{i+1}||, \forall y \in \{\dot{x}, \dot{y}, \ddot{\psi}\}}} \\ \boxed{\begin{array}{c} \text{Linearized Kinematics} \\ \text{Linearized Kinematics} \\ \text{Linearized Kinematics} \end{array}} & z_{k|t} + & \boxed{\begin{array}{c} \text{Linearized Kinematics} \\ \text{Linearized Kinematics} \\ \text{Linearized Kinematics} \end{array}} \end{array} \right] \begin{bmatrix} u_{k|t} \\ 1 \end{bmatrix}$$

Linearization around predicted trajectory

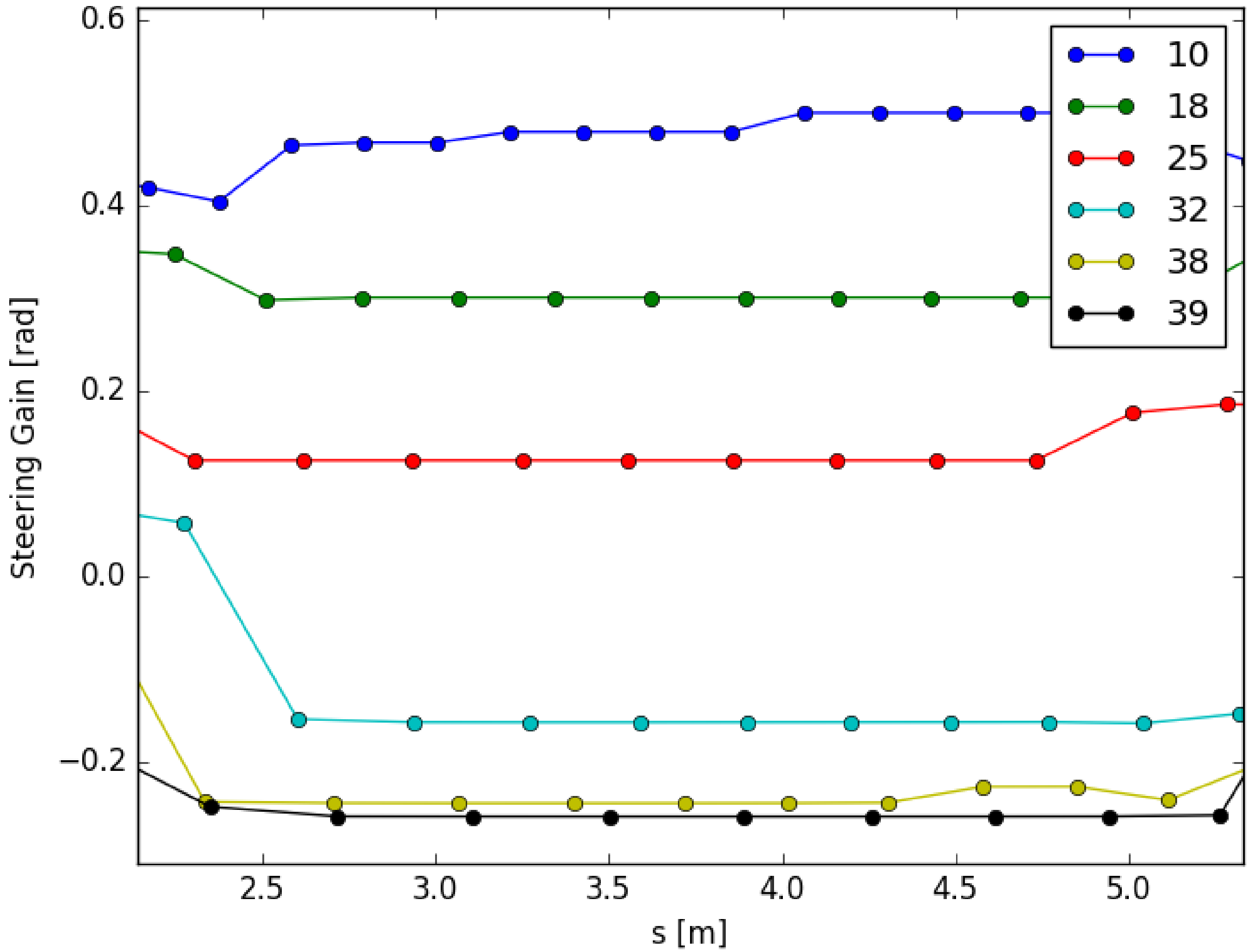
- **Important Design Steps**
  1. Compute **trajectory to linearize around**pusing previous optimal inputs and inputs in the safe set
  2. Enforce model-based **sparsity** in local linear regression
  3. Use **data close** to current state trajectory for parameter ID
  4. Use **kernel** K() to weight differently data as a function of distance to linearized trajectory

Accelerations



Results

- **Gain from steering to lateral velocity**



About Model Learning Ball in Cup

Ball in a Cup System with MuJoCo

Ball in a Cup Control Problem

$$\min_{T,\mathbf{u}} \quad \boxed{T + y^2}$$ Control objective

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

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

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

Learning Model Predictive Control (LMPC)

$$\min_{u_{t|t},\ldots,u_{t+N-1|t}} \sum_{k=t}^{t+N-1} \left(\mathbb{1}_{x_{k|t}\in\mathcal{X}_F} + y_{k|t}^2 \right) + \boxed{Q^{j-1}(x_{t+N|t})}$$

s.t.

$$x_{k+1|t} = A_{k|t}x_{k|t} + B_{k|t}u_{k|t} + C_{k|t}, \ \forall k \in [t, \cdots, t+N-1]$$

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

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

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

Receding Horizon Strategy:

$$u_t^j = u_0^*(x_t^j)$$

Useful Mujoco Model Abstraction

- **Identifying the Dynamical System**

Local Linear Regression

$$z_{k+1|t} = \begin{bmatrix} \dot{x}^{\text{ball}}_{k+1|t} \\ \dot{y}^{\text{ball}}_{k+1|t} \\ \dot{x}^{\text{cup}}_{k+1|t} \\ \dot{y}^{\text{cup}}_{k+1|t} \\ x^{\text{ball}}_{k+1|t} \\ y^{\text{ball}}_{k+1|t} \\ x^{\text{cup}}_{k+1|t} \\ y^{\text{cup}}_{k+1|t} \end{bmatrix} = \left[\begin{array}{c} \boxed{\arg\min \sum_i K(z_{k|t} - z_i) || \Lambda_y \begin{bmatrix} z_{k|t} \\ u_{k|t} \\ 1 \end{bmatrix} - y_{i+1} ||, \forall y \in \{\dot{x}, \dot{y}, \dot{e}^x, \dot{e}^y\}} \\ \boxed{\begin{bmatrix} \text{Linearized Kinematics} \\ \text{Linearized Kinematics} \\ \text{Linearized Kinematics} \\ \text{Linearized Kinematics} \end{bmatrix}} z_{k|t} + \boxed{\begin{bmatrix} \text{Linearized Kinematics} \\ \text{Linearized Kinematics} \\ \text{Linearized Kinematics} \\ \text{Linearized Kinematics} \end{bmatrix}} \end{array} \right] \begin{bmatrix} u_{k|t} \\ 1 \end{bmatrix}$$

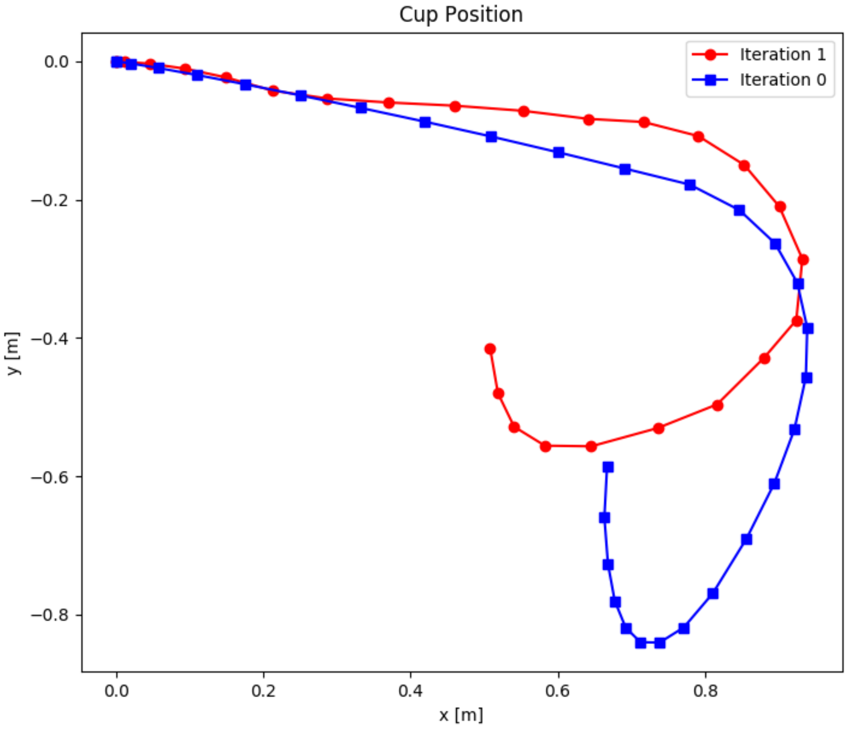
Linearization around predicted trajectory

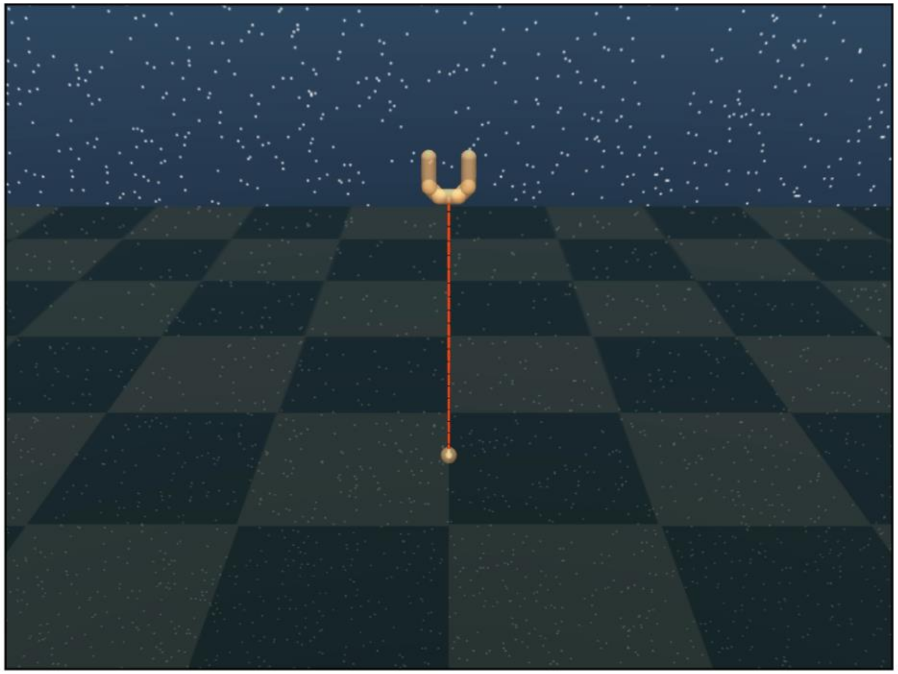
- **Important Design Steps**

1. Compute **trajectory to linearize around**pusing previous optimal inputs and inputs in the safe set
2. Enforce model-based **sparsity** in local linear regression
3. Use **data close** to current state trajectory for parameter ID
4. Use **kernel** K() to weight differently data as a function of distance to linearized trajectory

Ball in a Cup System

- At iteration 0 find a sequence by sampling parametrized inputs profiles (takes 5mins)
- Use ILMPC: At iteration 1, time reduced of 10%, cup height movement reduced of 35%





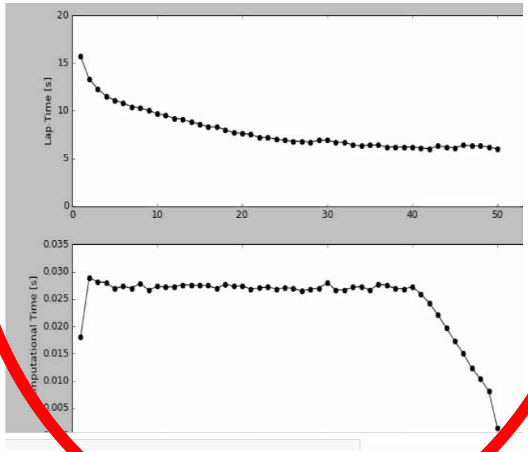
Back to our main chart..

Three Forms of Learning

Skill acquisition

Performance improvement

Reduce load for Routine Execution



How we do this?

Model Predictive Control +
A Simple Idea +
Good Practices

Offline $\pi(\cdot)$ and Online $\pi(x)$ Computation

$$\min_{\pi_0(\cdot),\pi_1(\cdot),\ldots,\pi_{N-1}(\cdot)} J_{0\to N}(x_0,\Pi)$$

$$\begin{array}{c}\text{subj. to}\\ k=0,\ldots,N-1\end{array} \quad \begin{cases} x_{k+1} = f(x_k,u_k,w_k) \\ u_k = \pi_k(x_k) \\ u_k \in \mathcal{U}, x_k \in \mathcal{X}, \ \forall w_k \in \mathcal{W} \end{cases}$$

$\pi_k(\cdot)$ Feedback Control Policies: $\pi_k: \ x_k \in \mathcal{X} \mapsto u_k \in \mathcal{U}$

In iterative tasks you can use both

Option 1 (*Offline Based*): "Complex" Offline, "Simple" Online

- $\pi(\cdot)$ often Piecewise Constant (except special classes)
- Dynamic Programming is one choice
- Basic rule: n>5 impossible

Option 2 (*Online Based*): "Simple" Offline, "Complex" Online

- Compute on-line $\pi(x)$ with a "sophisticated" algorithm
- Interior point method solver is one choice
- Basic Rule: avoid use `home-made' solvers

One Simple Way: Data-Based Policy for $\pi(\cdot)$

At time t, given the state $x(t)$ solve the following LP

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

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

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

Given the optimizer compute the input at time t

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

One Simple Way: Data-Based Policy for $\pi(\cdot)$

At time t, given the state $x(t)$ solve the following LP

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

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

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

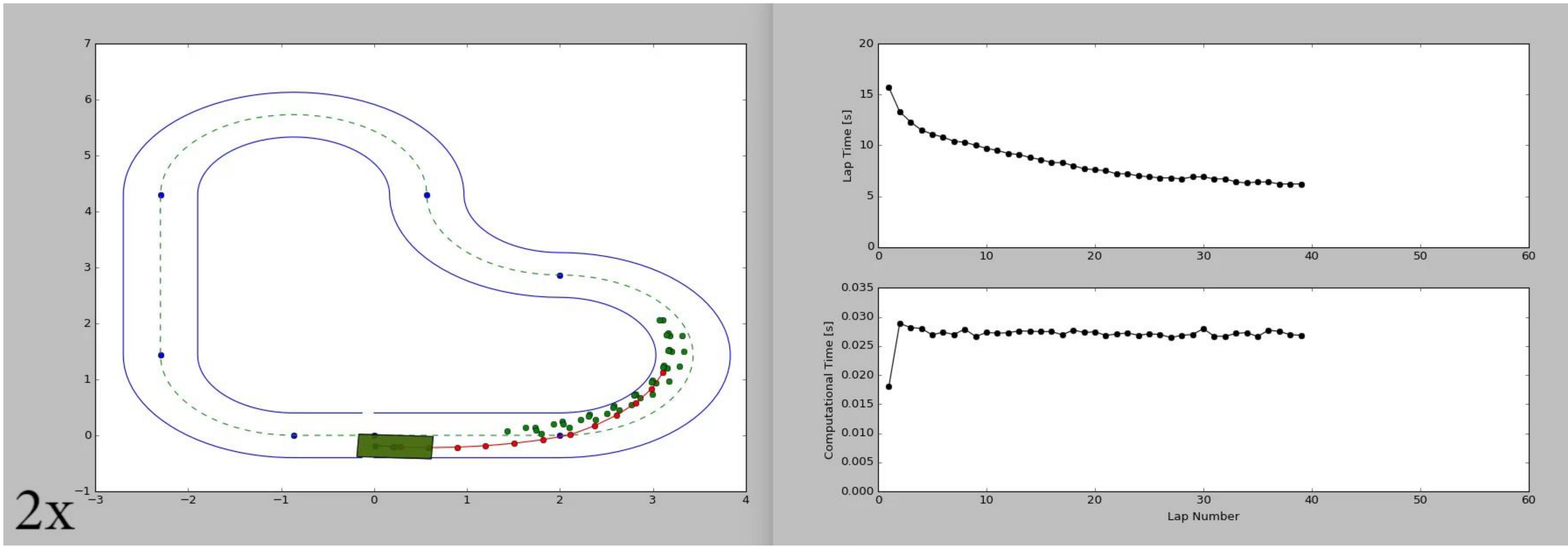
Historical data
of converged iterations

Given the optimizer compute the input

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

Three Forms of Learning
3 - **Computation Load Reduction**

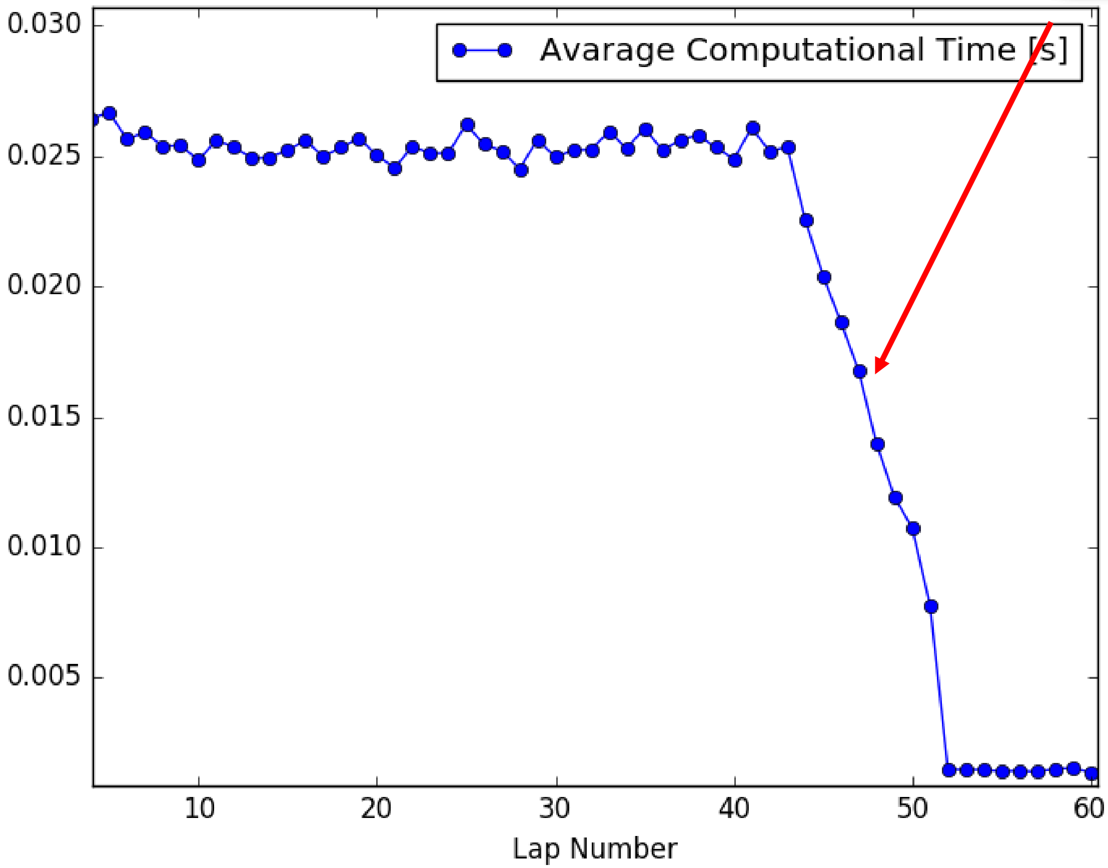
Lap Time at each iteration



Average CPU Load at each iteration

Experimental Results

Factor of 10



Data Based Policy: Alternatives

- Nearest Neighbor
- Train ReLU Neural Network
- Local Explicit MPC

All Continuous Piecewise Affine Policies

Learning MPC
Incorporating data in advance model based controller

$$J^{\mathrm{LMPC},j}_{t\to t+N}(x^j_t) = \min \sum^{t+N-1} h(x_{k|t}, u_{k|t}) + \boxed{Q^{j-1}(x_{t+N|t})}$$

What about noise and model uncertainty?

data

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

In Practice

- Noise and model uncertainty: Robust case

ILMPC – Robust and Adaptive design

At Iteration 0

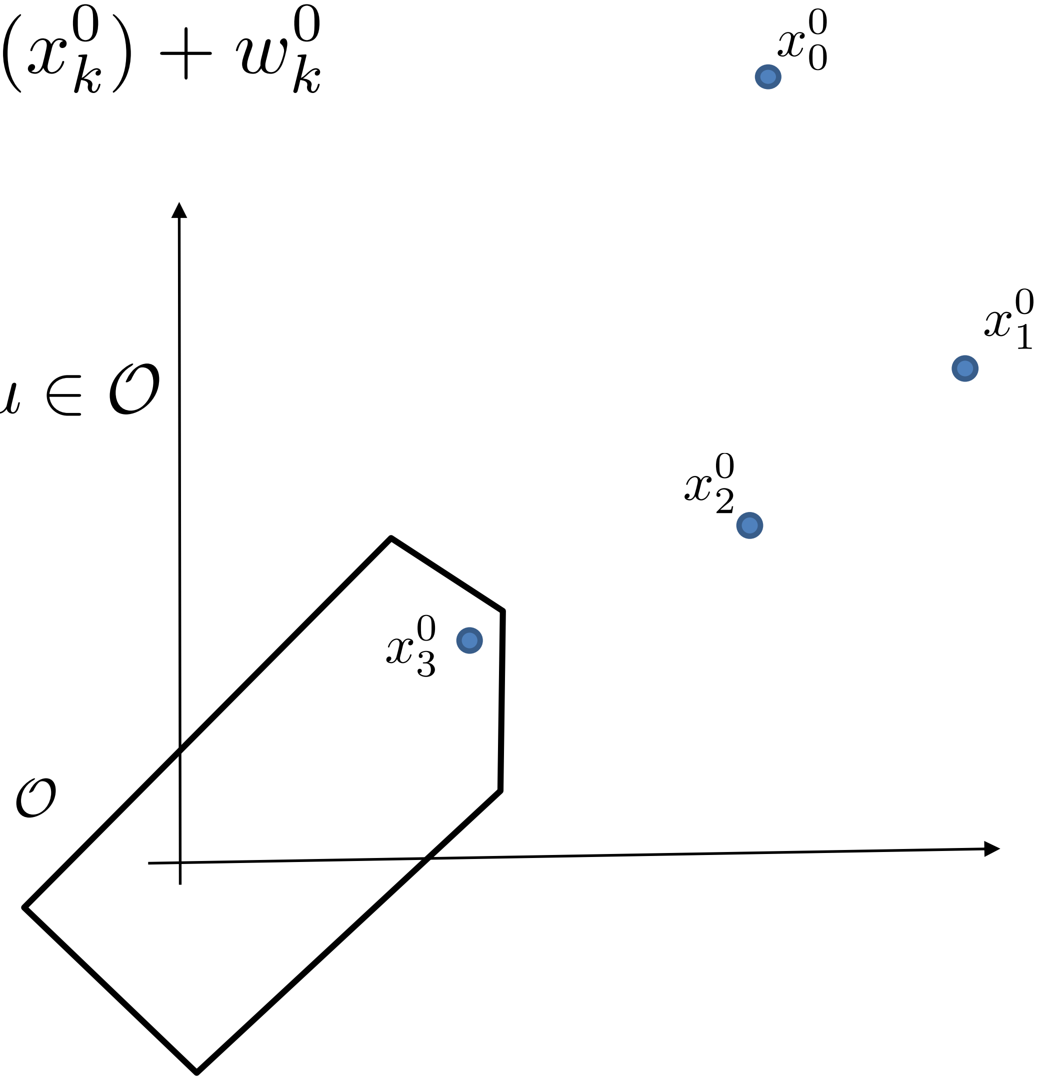
- Linear System

$$x_{k+1}^0 = Ax_k^0 + B\pi^1(x_k^0) + w_k^0$$

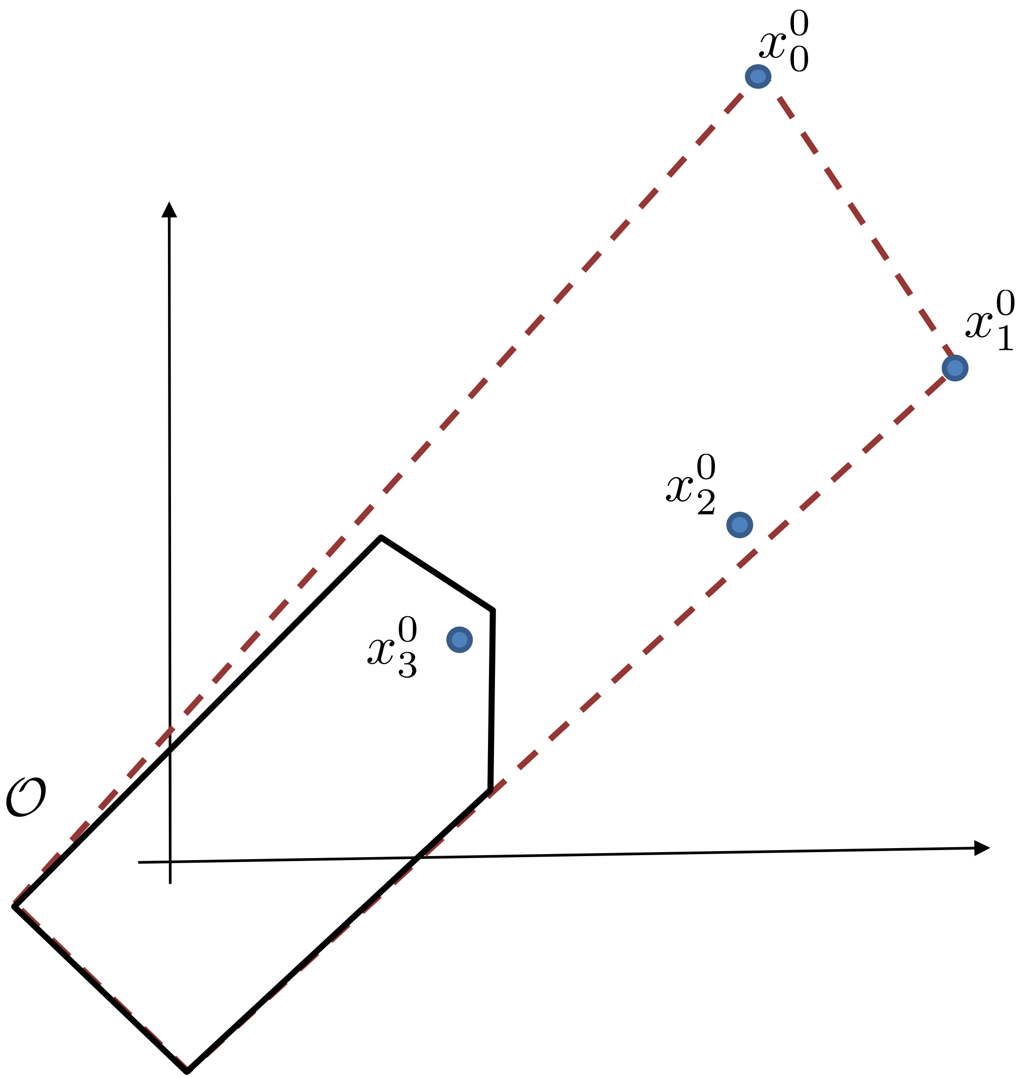
- Terminal Goal Set

$$\forall x \in \mathcal{O} \rightarrow Ax + BKu \in \mathcal{O}$$

- Successful Iteration



At Iteration 1



CVX hull is not a robust invariant!

ILMPC – Robust and Adaptive design

- Robust invariants
- “Robustify” Q-function (and dualize for computational efficiency)
- Chance constraints

See my group papers at this conference if interested..

For Iterative tasks I discussed

How to obtain performance improvement and reduced computational load while satisfying constraints

By using Iterative learning MPC, i.e.

- Model Predictive Control
- A Simple Idea
  (which exploits the iterative nature of the tasks)
- A Few Important Design Steps

The End