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, Komatsu
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

\[ J^*_0(x(0)) = \min_{\pi_0, \pi_1, \ldots} \sum_{k=0}^{\infty} h(x_k, u_k) \]

s.t. \( x_{k+1} = f(x_k, u_k) \)
\( u_k = \pi_k(x_k) \)
\( x_k \in \mathcal{X}, u_k \in \mathcal{U}, \)
\( x_0 = x(0) \)

\( \pi_k(\cdot) \) 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

\[
\begin{align*}
\min_{\pi_0, \pi_1, \ldots, \pi_{N-1}} & \quad p(x_{t+N}) + \sum_{k=0}^{N-1} h(x_{t+k}, u_{t+k}) \\
\text{subj. to} & \quad x_{k+1} = f(x_k, u_k) \\
& \quad u_k = \pi_k(x_k) \\
& \quad u_k \in U, \ x_k \in X \\
& \quad x_{t+N} \in X_f \\
& \quad x_t = x(t)
\end{align*}
\]

\[\pi_k(\cdot) \quad \text{Feedback Control Policies:} \quad \pi_k : \ x_k \in X \mapsto u_k \in U\]

Predictive Controller:

\[u(t) = \pi_0^*(x(t))\]
Repeated Solution of
Constrained Finite Time Optimal Control

\[
\begin{align*}
\min_{\pi_0, \pi_1, \ldots, \pi_{N-1}} & \quad p(x_{t+N}) + \sum_{k=0}^{N-1} h(x_{t+k}, u_{t+k}) \\
\text{subj. to} & \quad k = t, \ldots, t + N - 1 \\
\end{align*}
\]

\[
\begin{align*}
\begin{cases}
  x_{k+1} = f(x_k, u_k) \\
  u_k = \pi_k(x_k) \\
  u_k \in \mathcal{U}, x_k \in \mathcal{X} \\
  x_{t+N} \in \mathcal{X}_f \\
  x_t = x(t)
\end{cases}
\end{align*}
\]

\(\pi_k(\cdot)\) 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

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

subject to
\[
\begin{align*}
x_{k+1} &= f(x_k, u_k) \\
u_k &= \pi_k(x_k) \\
u_k &\in \mathcal{U}, x_k \in \mathcal{X} \\
x_{t+N} &\in \mathcal{X}_f \\
x_t &= x(t)
\end{align*}
\]

\(\pi_k(\cdot)\) 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

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

subj. to
\[
\begin{align*}
x_{k+1} &= f(x_k, u_k) \\
u_k &= \pi_k(x_k) \\
u_k &\in \mathcal{U}, x_k \in \mathcal{X} \\
x_{t+N} &\in \mathcal{X}_f \\
x_t &= x(t)
\end{align*}
\]

\(\pi_k(\cdot)\) 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

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

Subject to

\[
\begin{align*}
x_{k+1} &= f(x_k, u_k) \\
u_k &= \pi_k(x_k) \\
u_k &\in \mathcal{U}, x_k \in \mathcal{X} \\
x_{t+N} &\in \mathcal{X}_f \\
x_t &= x(t)
\end{align*}
\]

Terminal cost: Control Lyapunov function

Terminal constraint set: Control Invariant set

Control Invariant Set

\[x_0 \in \mathcal{X}_f \to \exists u_k \in \mathcal{U} : f(x_k, u_k) \in \mathcal{X}_f \quad \forall k > 0\]

Control Lyapunov Function

\[
\min_{u \in \mathcal{U}, f(x, u) \in \mathcal{X}_f} (p(f(x, u)) - p(x) + h(x, u)) \leq 0, \quad \forall x \in \mathcal{X}_f
\]
Repeated Solution of Constrained Finite Time Optimal Control

\[
\min_{\pi_0, \pi_1, \ldots, \pi_{N-1}} \quad p(x_{t+N}) + \sum_{k=0}^{N-1} h(x_{t+k}, u_{t+k}) \\
\text{subj. to} \\
k = t, \ldots, t + N - 1
\]

\[
\begin{align*}
 x_{k+1} &= f(x_k, u_k) \\
u_k &= \pi_k(x_k) \\
u_k &\in \mathcal{U}, x_k \in \mathcal{X} \\
x_{t+N} &\in \mathcal{X}_f \\
x_t &= x(t)
\end{align*}
\]

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

Predictive Controller:

\[
u(t) = \pi_0^*(x(t))\]
Offline $\pi(\cdot)$ and Online $\pi(\{\cdot\})$ Computation

\[
\min_{\pi_0(\cdot), \pi_1(\cdot), \ldots, \pi_{N-1}(\cdot)} J_{0\rightarrow N} (x_0, \Pi)
\]

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

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

Option 1 (*Offline Based*): “Complex” Offline, “Simple” Online
- $\pi_0(\cdot)$ often piecewise constant or affine disturbance feedback
- Dynamic Programming is one choice
- Sampling model reduction/aggregation required for $n>5$

Option 2 (*Online Based*): “Simple” Offline, “Complex” Online
- Compute on-line $\pi_0(\{t\})$ 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

- 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

A very biased story
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

Borrelli (UC Berkeley)  Iterative Learning MPC  2018 CDC– Slide 19
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 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**
  \[
  \begin{align*}
  \ddot{x} &= \dot{y}\psi + \frac{1}{m} \sum_i F_{x_i} \\
  \ddot{y} &= -\dot{x}\psi + \frac{1}{m} \sum_i F_{y_i} \\
  \ddot{\psi} &= \frac{1}{I_x} (a(F_{y_{1,2}}) - b(F_{y_{2,3}}) + c(-F_{x_{1,3}} + F_{x_{2,4}})) \\
  \dot{X} &= \dot{x}\cos\psi - \dot{y}\sin\psi, \quad \dot{Y} = \dot{x}\sin\psi + \dot{y}\cos\psi
  \end{align*}
  \]

- **Static Nonlinearities: Tires**
  \[
  F_y = f_y(\alpha, \sigma, \mu, F_z)
  \]
  \[
  F_x = f_x(\alpha, \sigma, \mu, F_z)
  \]
  and \[
  \sqrt{F_x^2 + F_y^2} \leq mg
  \]

- **Inequality Constraints: Safety region**

- **Uncertain Tire Model, Road Friction, Obstacles**
Tires and Road
Simplified Nonlinear Model

\[ \sqrt{F_x^2 + F_y^2} \leq \mu mg \]
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

Initialization
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:

\[ x^j_t = \text{system state at time } t \text{ of the } j\text{-th iteration} \]

\[ x^0 = x_S, \quad \forall \ j \geq 0 \]
Iterative Tasks - Problem Setup

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

Notation:

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

$$x_0^j = x_S, \; \forall j \geq 0$$
Iterative Tasks - Problem Setup

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

\[ x^j_t = \text{system state at time } t \text{ of the } j\text{-th iteration} \]

\[ x^j_0 = x_S, \quad \forall j \geq 0 \]
Iterative Learning MPC
Incorporating data in advanced model based controller

\[ J_{t \rightarrow t+N}^{\text{LMPC},j} (x_t^j) = \min_{u_t|t, \ldots, u_{t+N-1}|t} \sum_{k=t}^{t+N-1} h(x_k|t, u_k|t) + Q_j^{-1}(x_{t+N}|t) \]

s.t.
\[
\begin{align*}
    x_{k+1}|t &= f(x_k|t, u_k|t), \quad \forall k \in [t, \ldots, t+N-1] \\
    x_t|t &= x_t^j, \\
    x_k|t &\in \mathcal{X}, \quad u_k|t \in \mathcal{U}, \quad \forall k \in [t, \ldots, t+N-1] \\
    x_{t+N}|t &\in \mathcal{S}\mathcal{S}_j^{-1},
\end{align*}
\]

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

Learned from data
Learning MPC
Incorporating data in advance model based controller

\[ J_{t\rightarrow t+N}^{LMPC,j}(x_t^j) = \min_{u_t|t, \ldots, u_{t+N-1}|t} \sum_{k=t}^{t+N-1} h(x_k|t, u_k|t) + Q_{j-1}^j(x_{t+N}|t) \]

s.t.
\[
\begin{align*}
x_{k+1|t} &= f(x_k|t, u_k|t), \quad \forall k \in [t, \ldots, t+N-1] \\
x_{t|t} &= x_t^j, \\
x_k|t &\in \mathcal{X}, \ u_k|t \in \mathcal{U}, \quad \forall k \in [t, \ldots, t+N-1] \\
x_{t+N|t} &\in \mathcal{S}\mathcal{S}^{j-1},
\end{align*}
\]

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)

\[ J_{t \rightarrow t+N}^{\text{LMPC},j}(x_t^j) = \min_{u_t|t, \ldots, u_{t+N-1}|t} \sum_{k=t}^{t+N-1} h(x_k|t, u_k|t) + Q^{j-1}(x_{t+N}|t) \]

s.t.

\[ x_{k+1}|t = f(x_k|t, u_k|t), \ \forall k \in [t, \ldots, t + N - 1] \]

\[ x_{t|t} = x_t^j, \]

\[ x_k|t \in \mathcal{X}, \ u_k|t \in \mathcal{U}, \ \forall k \in [t, \ldots, t + N - 1] \]

\[ x_{t+N}|t \in \mathcal{S} \mathcal{S}^{j-1}, \]

- Recursive feasibility
- Iterative feasibility
Iteration 0

Assume at iteration 0 the closed-loop trajectory is feasible

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

\[ x_0^j = x_S, \quad \forall j \geq 0 \]
Iteration 0

Assume at iteration 0 the closed-loop trajectory is feasible

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

Fact

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

$x^0_0 = x_S, \ \forall j \geq 0$
Iteration 1, Step 0

Use $S^0$ as terminal set at Iteration 1

$x_F$, $x_7^0$, $x_6^0$, $x_5^0$, $x_4^0$, $x_3^0$, $x_2^0$, $x_1^0$, $x_0^j = x_S$, $\forall j \geq 0$
Iteration 1, Step 0

Use $S^0$ as terminal set at Iteration 1

$x_F$  
$x_7^0$  
$x_6^0$  
$x_5^0$  
$x_4^0$  
$x_3^0$  
$x_2^0$  
$x_1^1$  
$x_1^0$  
$x_0^j = x_S$, $\forall j \geq 0$
Iteration 1, Step 1

Use $S^0$ as terminal set at Iteration 1

$x^F$
$x^0_7$
$x^0_6$
$x^0_5$
$x^0_4$
$x^0_3$
$x^0_2$
$x^0_1$
$x^1_1$
$x^j_0 = x_S$, $\forall j \geq 0$
Iteration 1, Step 1

Use $S^0$ as terminal set at Iteration 1
Iteration 1, Step 2

Use \( S^0 \) as terminal set at Iteration 1

\[ x_{0}^{j} = x_{S}, \ \forall j \geq 0 \]
Iteration 1, Step 2

Use $S^0$ as terminal set at Iteration 1

$x_F$ $x_7^0$ $x_6^0$ $x_5^0$ $x_4^0$ $x_3^0$ $x_2^0$ $x_1^0$ $x_1^1$ $x_2^1$ $x_3^0$ $x_4^0$ $x_5^0$ $x_6^0$ $x_7^0$

$x_0^j = x_S$, $\forall j \geq 0$
Iteration 1, Step 3

Use $SS^0$ as terminal set at Iteration 1

$x_F \quad x_7^0 \quad x_6^0 \quad x_5^0 \quad x_4^0 \quad x_3^0 \quad x_2^0 \quad x_1^0 \quad x_1^1 \quad x_2^1 \quad x_3^0 \quad x_3^1 \quad x_4^0 \quad x_5^0 \quad x_6^0 \quad x_7^0$

$x_0^j = x_S, \; \forall j \geq 0$
Iteration 1, Step 4

Use $S_0^0$ as terminal set at Iteration 1

$x_0^j = x_S$, $\forall j \geq 0$
Iteration 2 Safe Set

\[ SS^0 = \left\{ \bigcup_{t=0}^{\infty} x^0_t \right\} \quad SS^1 = \left\{ \bigcup_{j=0}^{1} \bigcup_{t=0}^{\infty} x^j_t \right\} \quad SS^1 \supset SS^0 \]

\[ x^j_0 = x_s, \quad \forall j \geq 0 \]
Constructing the terminal set

\[ SS^j = \left\{ \bigcup_{i \in M^j} \bigcup_{t=0}^{\infty} x^i_t \right\} \]

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

\[ x_0^j = x_S, \ \forall j \geq 0 \]
Terminal Set: Convex all of Sample Safe Set

\[ CS^j = \text{Conv}(SS^j) \]

for Constrained Linear Dynamical Systems is a Control Invariant Set
Learning Model Predictive Control (LMPC)

\[
J_{t \rightarrow t+N}^{\text{LMPC},j}(x_t^j) = \min_{u_t|t, \ldots, u_{t+N-1}|t} \sum_{k=t}^{t+N-1} h(x_k|t, u_k|t) + Q^{-1}(x_{t+N|t})
\]

s.t.

\[
\begin{align*}
x_{k+1|t} &= f(x_k|t, u_k|t), \quad \forall k \in [t, \ldots, t+N-1] \\
x_{t|t} &= x_t^j, \\
x_k|t &\in \mathcal{X}, \quad u_k|t \in \mathcal{U}, \quad \forall k \in [t, \ldots, t+N-1] \\
x_{t+N|t} &\in \mathcal{S}S^j^{-1},
\end{align*}
\]

• Convergence
• Performance improvement
• Local optimality
Terminal Cost at Iteration 0

\[ x_F \]

\[ x_7^0 \]

\[ x_6^0 \quad x_5^0 \quad x_4^0 \quad x_3^0 \]

\[ x_2^0 \]

\[ x_1^0 \]

\[ x_0^i = x_S, \quad \forall j \geq 0 \]
Terminal Cost at Iteration 0

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

A control Lyapunov “function”
Define terminal cost as:

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

\[ Q^j(x) = \begin{cases} 
\min_{(i,t) \in F^j(x)} J_{i \to \infty}^i(x), & \text{if } x \in SS^j \\
+\infty, & \text{if } x \notin SS^j 
\end{cases} \]

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

\[ Q^*(x) = \min_{\lambda^j \in [0,1]} \sum_i \sum_j Q^j_i \lambda^j_i \]
\[
\text{s.t. } \sum_i \sum_j x^j_i \lambda^j_i = x, \\
\sum_i \sum_j \lambda^j_i = 1 \]

Control Lyapunov Function (for Constrained Linear Dynamical Systems)
ILMPC Summary

MPC strategy:

\[ J_{t \rightarrow t+N}^{\text{ILMPC}, j}(x^j_t) = \min_{u_t^j, \ldots, u_{t+N-1}^j} \sum_{k=t}^{t+N-1} h(x_k|t, u_k|t) + Q^{j-1}(x_{t+N}|t) \]

s.t.

\[
\begin{align*}
    x_{k+1|t} &= A_{k|t}x_k|t + B_{k|t}u_k|t + C_{k|t}, \quad \forall k \in [t, \ldots, t+N-1] \\
    x_t|t &= x_t^j, \\
    x_k|t &\in \mathcal{X}, \quad u_k|t \in \mathcal{U}, \quad \forall k \in [t, \ldots, t+N-1] \\
    x_{t+N}|t &\in \mathcal{CS}^{j-1}
\end{align*}
\]

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

\[ J_{0 \rightarrow \infty}^{j-1}(x_{0}^{j-1}) \geq J_{0 \rightarrow \infty}^{j}(x_{0}^{j}) \]

Notation

\[ x^{j} = [x_{0}^{j}, x_{1}^{j}, ..., x_{t}^{j}, ...] \quad u^{j} = [u_{0}^{j}, u_{1}^{j}, ..., u_{t}^{j}, ...] \]

Closed-loop state and input trajectory at iteration \( j \)
Performance Improvement Proof

Step 1: \[ J^{j-1}_{0 \rightarrow \infty} (x_{0}^{j-1}) \geq J^{LMPC}_{0 \rightarrow N} (x_{0}^{j}) \]

\[ J^{j-1}_{0 \rightarrow \infty} (x_{0}^{j-1}) = \sum_{k=0}^{\infty} h(x_{k}^{j-1}, u_{k}^{j-1}) = \]
Performance Improvement Proof

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

\[ J_{0 \rightarrow j}^{j-1} (x_0^{j-1}) = \sum_{k=0}^{\infty} h(x_k^{j-1}, u_k^{j-1}) = \sum_{k=0}^{N-1} h(x_k^{j-1}, u_k^{j-1}) + \sum_{k=N}^{\infty} h(x_k^{j-1}, u_k^{j-1}) \]
Performance Improvement Proof

Step 1: 
\[
J_0^{j-1}(x_0^{j-1}) \geq J_{0 \rightarrow N}^{LMP C, j}(x_0^j)
\]

\[
J_0^{j-1}(x_0^{j-1}) = \sum_{k=0}^{\infty} h(x_k^{j-1}, u_k^{j-1}) = \sum_{k=0}^{N-1} h(x_k^{j-1}, u_k^{j-1}) + \sum_{k=N}^{\infty} h(x_k^{j-1}, u_k^{j-1})
\]

\[
Q^{j-1}(x_N^{j-1})
\]
Performance Improvement Proof

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

\[ J_{0\to\infty}^{j-1}(x_0^{j-1}) = \sum_{k=0}^{\infty} h(x_k^{j-1}, u_k^{j-1}) = \sum_{k=0}^{N-1} h(x_k^{j-1}, u_k^{j-1}) + \sum_{k=N}^{\infty} h(x_k^{j-1}, u_k^{j-1}) \]

\[ Q_{N}^{j-1}(x_N^{j-1}) \]

\[ \rightarrow J_{0\to\infty}^{j-1}(x_0^{j-1}) = \sum_{k=0}^{N-1} h(x_k^{j-1}, u_k^{j-1}) + Q_{N}^{j-1}(x_N^{j-1}) \geq J_{0\to N}^{LMPC,j}(x_0^j) \]
Performance Improvement Proof

Step 1: \[ J_{0 \to \infty}^{\lambda^{-1}}(x_{0}^{\lambda^{-1}}) \geq J_{0 \to N}^{LMPC,j}(x_{0}^{j}) \]

Step 2: \[ J_{0 \to N}^{LMPC,j}(x_{0}^{j}) \geq J_{0 \to \infty}^{j}(x_{0}^{j}) \]

\[ J_{1 \to 1+N}^{LMPC,j}(x_{1}^{j}) - J_{0 \to N}^{LMPC,j}(x_{0}^{j}) \leq -h(x_{0}^{j}, u_{0}^{j}) \]
Performance Improvement Proof

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

Step 2: \( J_{0\to N}^{LMPC,j} (x_0^j) \geq J_{0\to \infty}^j (x_0^j) \)

\[
J_{1\to 1+N}^{LMPC,j} (x_1^j) - J_{0\to N}^{LMPC,j} (x_0^j) \leq -h(x_0^j, u_0^j)
\]

\[
\rightarrow J_{0\to N}^{LMPC,j} (x_0^j) \geq J_{1\to 1+N}^{LMPC,j} (x_1^j) + h(x_0^j, u_0^j) \geq J_{2\to 2+N}^{LMPC,j} (x_2^j) + h(x_0^j, u_0^j) + h(x_1^j, u_1^j)
\]
Performance Improvement Proof

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

Step 2: \[ J_{0\to N}^{LMPC,j}(x_0^j) \geq J_{0\to\infty}(x_0^j) \]

\[ J_{1\to1+N}^{LMPC,j}(x_1^j) - J_{0\to N}^{LMPC,j}(x_0^j) \leq -h(x_0^j, u_0^j) \]

\[ \rightarrow J_{0\to N}^{LMPC,j}(x_0^j) \geq J_{1\to1+N}^{LMPC,j}(x_1^j) + h(x_0^j, u_0^j) \geq J_{2\to2+N}^{LMPC,j}(x_2^j) + h(x_0^j, u_0^j) + h(x_1^j, u_1^j) \]

\[ \rightarrow J_{0\to N}^{LMPC,j}(x_0^j) \geq \lim_{t\to\infty} J_{t\to t+N}^{LMPC,j}(x_t^j) + \sum_{k=0}^{\infty} h(x_k^j, u_k^j) \]

0
Performance Improvement Proof

Conclusion: \[ J_{0\to\infty}^{j-1}(x_{0}^{j-1}) \geq J_{0\to N}^{LMPC,j}(x_{0}^{j}) \geq J_{0\to\infty}^{j}(x_{0}^{j}) \]

The iteration cost \( J_{0\to\infty}^{j} \) 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

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

If full column rank, improvement cannot be obtained
Constrained LQR Example

\[
\begin{align*}
\text{min} & \quad \sum_{k=0}^{\infty} \left[ \|x_k\|_2^2 + \|u_k\|_2^2 \right] \\
\text{s.t.} & \quad x_{k+1} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} x_k + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u_k, \quad \forall k \geq 0 \\
& \quad x_k \in \text{box}[-4, 4], \quad \forall k \geq 0 \\
& \quad u_k \in [-1, 1], \quad \forall k \geq 0 \\
& \quad x_0 = [-2.120, 0.066]^T,
\end{align*}
\]
Iterative LMPC with horizon N=2

\[
\min_{u_0|t, u_1|t} \sum_{k=0}^{2} \left[ \|x_k|t\|_2^2 + \|u_k|t\|_2^2 \right] + Q^{i-1}(x_2|t)
\]

Control objective

\[
x_{k+1}|t = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} x_k|t + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u_k|t, \ \forall k = [0, 1]
\]

System dynamics

\[
x_k|t \in \text{box}[-4, 4] \ \forall k = [0, 1]
\]

System constraints

\[
u_k|t \in [-1, 1] \ \forall k = [0, 1]
\]

Terminal Constraint

\[
x_{2|t} \in CS^{i-1}
\]

Initial Condition

\[
x_{0|t} = x(t),
\]

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**

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

\[ \min_{T, u} T \quad \text{Control objective} \]

\[ x_0 = x_s, \ x_T = x_F \quad \text{Start & end position} \]

\[ x_{k+1} = f(x_k, u_k), \ \forall k \in \{0, \ldots, T - 1\} \quad \text{System dynamics} \]

\[ x_k \in \mathcal{X}, \ u_k \in \mathcal{U}, \ \forall k \in \{0, \ldots, T - 1\} \quad \text{System constraints} \]

\[ \text{Obstacle avoidance} \]
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.
\[
\begin{align*}
x_{k+1|t} &= A_{k|t} x_k|t + B_{k|t} u_k|t + C_{k|t}, \quad \forall k \in [t, \ldots, t + N - 1] \\
x_{t|t} &= x_t^j, \\
x_k|t &\in \mathcal{X}, \quad u_k|t \in \mathcal{U}, \quad \forall k \in [t, \ldots, t + N - 1] \\
x_{t+N|t} &\in CS^{j-1}
\end{align*}
\]

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.
\[
x_{k+1|t} = A_{k|t}x_k|t + B_{k|t}u_{k|t} + C_{k|t}, \quad \forall k \in [t, \ldots, t + N - 1]
\]
\[
x_t|t = x_t^j
\]
\[
x_{k|t} \in \mathcal{X}, \quad u_{k|t} \in \mathcal{U}, \quad \forall k \in [t, \ldots, t + N - 1]
\]
\[
x_{t+N|t} \in \mathcal{S}^{j-1}
\]

Receding Horizon Strategy:

\[
u_t^j = u_0^*(x_t^j)
\]
Useful Vehicle Model Abstraction

- **Nonlinear Dynamical System**

\[
\begin{align*}
\ddot{x} &= \dot{y} \psi + \frac{1}{m} \sum_i F_{xi} \\
\ddot{y} &= -\dot{x} \psi + \frac{1}{m} \sum_i F_{yi} \\
\ddot{\psi} &= \frac{1}{I_z} \left( a(F_{y1,2}) - b(F_{y2,3}) + c(-F_{x1,3} + F_{x2,4}) \right) \\
\dot{X} &= \dot{x} \cos \psi - \dot{y} \sin \psi, \quad \dot{Y} = \dot{x} \sin \psi + \dot{y} \cos \psi
\end{align*}
\]
Useful Vehicle Model Abstraction

- **Nonlinear Dynamical System**

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

Kinematic Equations
Useful Vehicle Model Abstraction

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

- **Identifying the Dynamical System**

\[
\begin{bmatrix}
  \dot{x}_{k+1|t} \\
  \dot{y}_{k+1|t} \\
  \dot{\psi}_{k+1|t} \\
  \psi_{k+1|t} \\
  X_{k+1|t} \\
  Y_{k+1|t}
\end{bmatrix} = \begin{bmatrix}
  \text{Linearized Kinematics} \\
  \text{Linearized Kinematics} \\
  \text{Linearized Kinematics} \\
  \text{Linearized Kinematics} \\
  \text{Linearized Kinematics} \\
  \text{Linearized Kinematics}
\end{bmatrix} \begin{bmatrix}
  z_{k|t} \\
  \text{Linearization around predicted trajectory}
\end{bmatrix} + \begin{bmatrix}
  u_{k|t} \\
  1
\end{bmatrix}
\]
Useful Vehicle Model Abstraction

- **Nonlinear Dynamical System**

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

- **Kinematic Equations**

- **Dynamic Equations**

- **Identifying the Dynamical System**

- **Local Linear Regression**

\[
z_{k+1|t} = \begin{bmatrix}
\dot{x}_{k+1|t} \\
\dot{y}_{k+1|t} \\
\dot{\psi}_{k+1|t} \\
\psi_{k+1|t} \\
X_{k+1|t} \\
Y_{k+1|t}
\end{bmatrix} = \text{arg min} \sum_i K(z_{k|t} - z_i) \| \Lambda_y \begin{bmatrix}
z_{k|t} \\
u_{k|t}
\end{bmatrix} - y_{i+1} \|, \forall y \in \{\dot{x}, \dot{y}, \dot{\psi}\}
\]

- **Linearization around predicted trajectory**
Useful Vehicle Model Abstraction

Identifying the **Dynamical System**

Local Linear Regression

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

\[
\begin{align*}
\text{arg min} \sum_i K(z_{k|t} - z_i) || \Lambda_y \begin{bmatrix} z_{k|t} \\ u_{k|t} \\ 1 \end{bmatrix} - y_{i+1} ||, \forall y \in \{\dot{x}, \dot{y}, \dot{\psi}\}
\end{align*}
\]

**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

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

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

\[ \Lambda \dot{x} = \begin{bmatrix} \lambda_1 & \lambda_2 & \lambda_3 & 0 & 0 & 0 & \lambda_4 & 0 & \lambda_5 \end{bmatrix} \]
Useful Vehicle Model Abstraction

Identifying the Dynamical System

Local Linear Regression

\[ z_{k+1|t} = \begin{bmatrix} \dot{x}_{k+1|t} \\ \dot{y}_{k+1|t} \\ \dot{\psi}_{k+1|t} \\ X_{k+1|t} \\ Y_{k+1|t} \end{bmatrix} = \arg \min \sum_i K(z_{k|t} - z_i) || \Lambda_y \begin{bmatrix} z_{k|t} \\ u_{k|t} \end{bmatrix} - y_{i+1} ||, \forall y \in \{\dot{x}, \dot{y}, \dot{\psi}\} \]

Important Design Steps

1. Compute trajectory to linearize around using 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

Linearization around predicted 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

\[ \min_{T,u} \quad T + y^2 \]

Control objective

\[ x_0 = x_s, \quad x_T = \chi_F \]

Start & end position

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

Obstacle avoidance
\[ x_k \in \chi, \quad u_k \in \mathcal{U}, \quad \forall k \in \{0, \ldots, T - 1\} \]
Learning Model Predictive Control (LMPC)

\[
\min_{u_t|t, \ldots, u_{t+N-1}|t} \sum_{k=t}^{t+N-1} \left( \mathbb{1}_{x_k|t \in X_F} + y_k^2|t \right) + Q^{-1}(x_{t+N}|t)
\]

s.t.
\[
\begin{align*}
    x_{k+1|t} &= A_{k|t}x_k|t + B_{k|t}u_k|t + C_{k|t}, \quad \forall k \in [t, \ldots, t + N - 1] \\
    x_t|t &= x_t^j, \\
    x_k|t &\in \mathcal{X}, \quad u_k|t \in \mathcal{U}, \quad \forall k \in [t, \ldots, t + N - 1] \\
    x_{t+N|t} &\in \mathcal{C}S^{-1}. 
\end{align*}
\]

Receding Horizon Strategy:
\[
u_t^j = u_0^*(x_t^j)
\]
Useful Mujoco Model Abstraction

Identifying the Dynamical System

Local Linear Regression

\[
\begin{bmatrix}
\dot{\text{ball}}_{k+1|t} \\
\dot{\text{cup}}_{k+1|t} \\
\dot{\text{ball}}_{k+1|t} \\
\dot{\text{ball}}_{k+1|t} \\
\dot{\text{cup}}_{k+1|t} \\
\dot{\text{cup}}_{k+1|t}
\end{bmatrix}
= \arg\min_i K(z_k|t - z_i)||\Lambda_y \begin{bmatrix}
z_k|t \\
u_k|t \\
1
\end{bmatrix} - y_{i+1}||, \forall y \in \{\dot{x}, \dot{y}, e^x, e^y\}
\]

Important Design Steps

1. Compute trajectory to linearize around using 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

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

subj. to
$$k = 0, \ldots, N - 1$$
$$x_{k+1} = f(x_k, u_k, w_k)$$
$$u_k = \pi_k(x_k)$$
$$u_k \in \mathcal{U}, x_k \in \mathcal{X}, w_k \in \mathcal{W}$$

$\pi_k(\cdot)$ Feedback Control Policies: $\pi_k: \mathcal{X} \rightarrow \mathcal{U}$

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

In iterative tasks you can use both
One Simple Way: Data-Based Policy for $\pi(\cdot)$

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

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

Given the optimizer compute the input at time $t$

$$\pi(x(t)) = \sum_i \sum_j w_i^j \lambda_i^j,^*$$
One Simple Way: Data-Based Policy for $\pi(\cdot)$

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

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

Given the optimizer compute the input of converged iterations

$$\pi(x(t)) = \sum_i \sum_j w_i^j \lambda_i^j, \ast$$

Historical data of converged iterations
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

\[
J_t^{\text{LMPC},j}(x_t^j) = \min_{u_t, \ldots, u_{t+N-1}} \sum_{k=t}^{t+N-1} h(x_{k|t}, u_{k|t}) + Q^{j-1}(x_{t+N|t})
\]

What about noise and model uncertainty?

In Practice

Noise and model uncertainty: Robust case
ILMPC – Robust and Adaptive design
At Iteration 0

- Linear System
  \[ x_{k+1}^0 = Ax_k^0 + B\pi_1^1(x_k^0) + w_k^0 \]

- 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