

So far we have seen...

- The building blocks
 - ▶ MDPs, DP, Bellman, Policy and Value Iteration, Reinforcement Learning, Actor-Critic
 - ▶ LQR, Dynamic Optimization, Nonlinear Programming, MPC, Sensitivities
- MPC & RL
 - ▶ Synthesis - Different flavors of ML / RL and MPC
 - ▶ Software tool leap-c

So far we have seen...

- The building blocks
 - ▶ MDPs, DP, Bellman, Policy and Value Iteration, Reinforcement Learning, Actor-Critic
 - ▶ LQR, Dynamic Optimization, Nonlinear Programming, MPC, Sensitivities
- MPC & RL
 - ▶ Synthesis - Different flavors of ML / RL and MPC
 - ▶ Software tool leap-c

What we will discuss: parametrized / hierarchical RL over MPC

- The theory that supports it and what does it tell us? (*Now*)

So far we have seen...

- The building blocks
 - ▶ MDPs, DP, Bellman, Policy and Value Iteration, Reinforcement Learning, Actor-Critic
 - ▶ LQR, Dynamic Optimization, Nonlinear Programming, MPC, Sensitivities
- MPC & RL
 - ▶ Synthesis - Different flavors of ML / RL and MPC
 - ▶ Software tool leap-c

What we will discuss: parametrized / hierarchical RL over MPC

- The theory that supports it and what does it tell us? (*Now*)
- Safe & stable RL over MPC

So far we have seen...

- The building blocks
 - ▶ MDPs, DP, Bellman, Policy and Value Iteration, Reinforcement Learning, Actor-Critic
 - ▶ LQR, Dynamic Optimization, Nonlinear Programming, MPC, Sensitivities
- MPC & RL
 - ▶ Synthesis - Different flavors of ML / RL and MPC
 - ▶ Software tool leap-c

What we will discuss: parametrized / hierarchical RL over MPC

- The theory that supports it and what does it tell us? (*Now*)
- Safe & stable RL over MPC
- Some current & future prospect

So far we have seen...

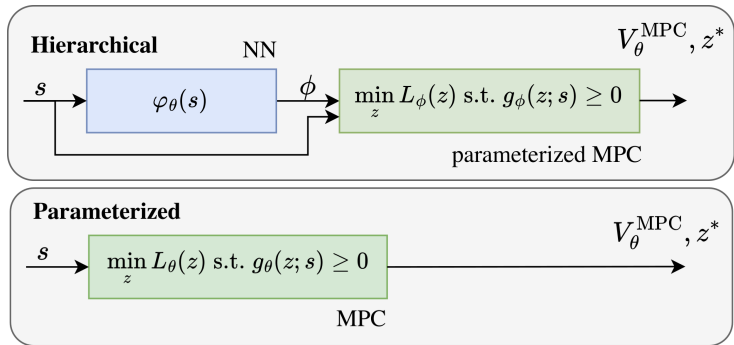
- The building blocks
 - ▶ MDPs, DP, Bellman, Policy and Value Iteration, Reinforcement Learning, Actor-Critic
 - ▶ LQR, Dynamic Optimization, Nonlinear Programming, MPC, Sensitivities
- MPC & RL
 - ▶ Synthesis - Different flavors of ML / RL and MPC
 - ▶ Software tool leap-c

What we will discuss: parametrized / hierarchical RL over MPC

- The theory that supports it and what does it tell us? (*Now*)
- Safe & stable RL over MPC
- Some current & future prospect
- Beyond MPC – Model-based Decisions and AI for decisions

Focus of today's lectures (Leap-c structure)

MPC is parametrized, and RL “discovers” how to improve these parameters for closed-loop performance



Why? MPC model that best fits the real world does not necessarily yield the best MPC policy. This lecture explains how and why. The structure in focus has strong foundations.

Some notation

- Probability (density, measure): $\mathbb{P}[\dots]$, the specifics are context-dependent.
- I will sometimes also use e.g. $\rho(\cdot)$ as self-standing densities.
- Expected values: $\mathbb{E}[\dots]$
- $\mathbb{P}[\mathbf{x}|\mathbf{y}]$ and $\mathbb{E}[\mathbf{x}|\mathbf{y}]$ are probability (density, measure) and expected value of $\mathbf{x}$ knowing $\mathbf{y}$
- Real world state and action (input): $\mathbf{s}, \mathbf{a}$
- Predicted state and action (input): $\mathbf{x}, \mathbf{u}$
- Successor state: $\mathbf{s}_+$ following $\mathbf{s}, \mathbf{a}$

Reinforcement Learning Over MPC

Why Does It Work?

Sebastien Gros

Dept. of Cybernetic, NTNU
Faculty of Information Tech.

Freiburg PhD School

Outline

- 1 Let's rebuild some background – MPC and MDPs
- 2 MPC as a solution to MDPs
- 3 When is RL (most) beneficial for MPC?
- 4 A Deeper Look at the Theory

Model Predictive Control (MPC)

Optimize a plan over finite horizon, apply first move, repeat

MPC: at **current state** $\mathbf{s}$

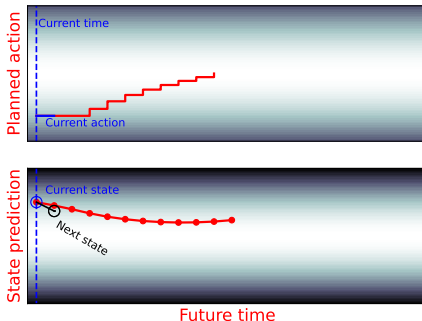
$$\min_{\mathbf{x}, \mathbf{u}} \quad T(\mathbf{x}_N) + \sum_{k=0}^{N-1} L(\mathbf{x}_k, \mathbf{u}_k)$$

$$\text{s.t.} \quad \mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k)$$

$$\mathbf{h}(\mathbf{x}_k, \mathbf{u}_k) \leq 0$$

$$\mathbf{x}_0 = \mathbf{s}$$

apply **action** $\mathbf{a} = \mathbf{u}_0^*$ to the system



Model Predictive Control (MPC)

Optimize a plan over finite horizon, apply first move, repeat

MPC: at **current state** $\mathbf{s}$

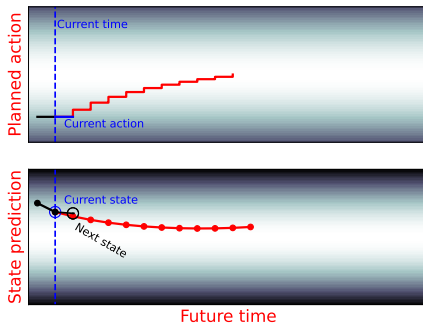
$$\min_{\mathbf{x}, \mathbf{u}} \quad T(\mathbf{x}_N) + \sum_{k=0}^{N-1} L(\mathbf{x}_k, \mathbf{u}_k)$$

$$\text{s.t.} \quad \mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k)$$

$$\mathbf{h}(\mathbf{x}_k, \mathbf{u}_k) \leq 0$$

$$\mathbf{x}_0 = \mathbf{s}$$

apply **action** $\mathbf{a} = \mathbf{u}_0^*$ to the system



Model Predictive Control (MPC)

Optimize a plan over finite horizon, apply first move, repeat

MPC: at **current state** $\mathbf{s}$

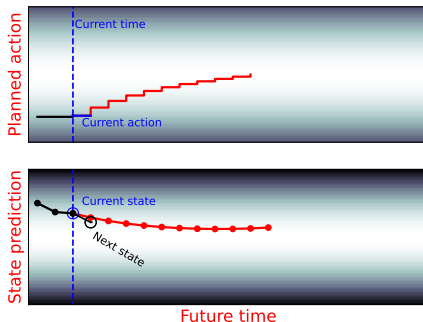
$$\min_{\mathbf{x}, \mathbf{u}} \quad T(\mathbf{x}_N) + \sum_{k=0}^{N-1} L(\mathbf{x}_k, \mathbf{u}_k)$$

$$\text{s.t.} \quad \mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k)$$

$$\mathbf{h}(\mathbf{x}_k, \mathbf{u}_k) \leq 0$$

$$\mathbf{x}_0 = \mathbf{s}$$

apply **action** $\mathbf{a} = \mathbf{u}_0^*$ to the system



Model Predictive Control (MPC)

Optimize a plan over finite horizon, apply first move, repeat

MPC: at **current state** $\mathbf{s}$

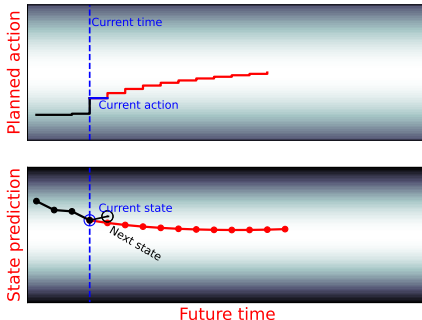
$$\min_{\mathbf{x}, \mathbf{u}} \quad T(\mathbf{x}_N) + \sum_{k=0}^{N-1} L(\mathbf{x}_k, \mathbf{u}_k)$$

$$\text{s.t.} \quad \mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k)$$

$$\mathbf{h}(\mathbf{x}_k, \mathbf{u}_k) \leq 0$$

$$\mathbf{x}_0 = \mathbf{s}$$

apply **action** $\mathbf{a} = \mathbf{u}_0^*$ to the system



Model Predictive Control (MPC)

Optimize a plan over finite horizon, apply first move, repeat

MPC: at **current state** $\mathbf{s}$

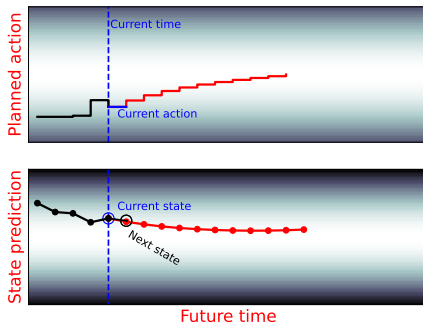
$$\min_{\mathbf{x}, \mathbf{u}} \quad T(\mathbf{x}_N) + \sum_{k=0}^{N-1} L(\mathbf{x}_k, \mathbf{u}_k)$$

$$\text{s.t.} \quad \mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k)$$

$$\mathbf{h}(\mathbf{x}_k, \mathbf{u}_k) \leq 0$$

$$\mathbf{x}_0 = \mathbf{s}$$

apply **action** $\mathbf{a} = \mathbf{u}_0^*$ to the system



Model Predictive Control (MPC)

Optimize a plan over finite horizon, apply first move, repeat

MPC: at current state $\mathbf{s}$

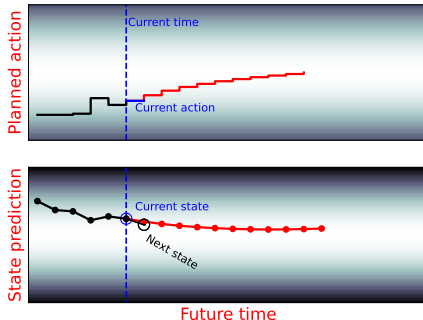
$$\min_{\mathbf{x}, \mathbf{u}} \quad T(\mathbf{x}_N) + \sum_{k=0}^{N-1} L(\mathbf{x}_k, \mathbf{u}_k)$$

$$\text{s.t.} \quad \mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k)$$

$$\mathbf{h}(\mathbf{x}_k, \mathbf{u}_k) \leq 0$$

$$\mathbf{x}_0 = \mathbf{s}$$

apply action $\mathbf{a} = \mathbf{u}_0^*$ to the system



Model Predictive Control (MPC)

Optimize a plan over finite horizon, apply first move, repeat

MPC: at current state $\mathbf{s}$

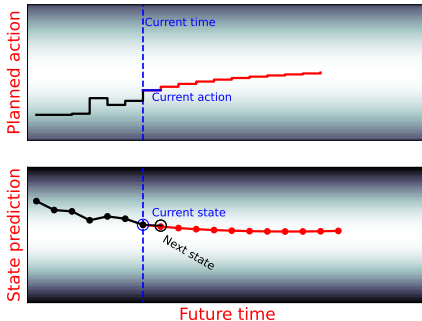
$$\min_{\mathbf{x}, \mathbf{u}} \quad T(\mathbf{x}_N) + \sum_{k=0}^{N-1} L(\mathbf{x}_k, \mathbf{u}_k)$$

$$\text{s.t.} \quad \mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k)$$

$$\mathbf{h}(\mathbf{x}_k, \mathbf{u}_k) \leq 0$$

$$\mathbf{x}_0 = \mathbf{s}$$

apply action $\mathbf{a} = \mathbf{u}_0^*$ to the system



Model Predictive Control (MPC)

Optimize a plan over finite horizon, apply first move, repeat

MPC: at current state $\mathbf{s}$

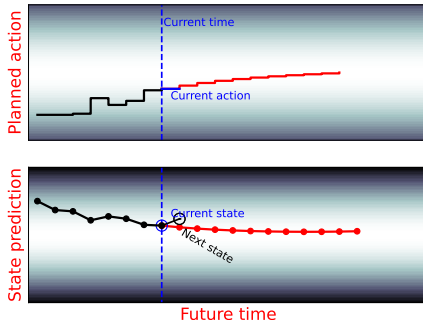
$$\min_{\mathbf{x}, \mathbf{u}} \quad T(\mathbf{x}_N) + \sum_{k=0}^{N-1} L(\mathbf{x}_k, \mathbf{u}_k)$$

$$\text{s.t.} \quad \mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k)$$

$$\mathbf{h}(\mathbf{x}_k, \mathbf{u}_k) \leq 0$$

$$\mathbf{x}_0 = \mathbf{s}$$

apply action $\mathbf{a} = \mathbf{u}_0^*$ to the system



Model Predictive Control (MPC)

Optimize a plan over finite horizon, apply first move, repeat

MPC: at **current state** $\mathbf{s}$

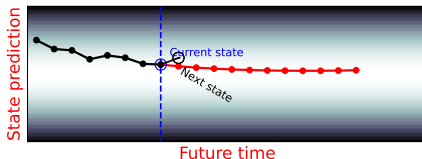
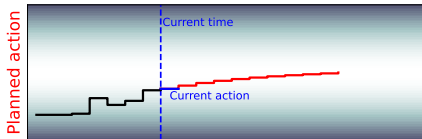
$$\min_{\mathbf{x}, \mathbf{u}} \quad T(\mathbf{x}_N) + \sum_{k=0}^{N-1} L(\mathbf{x}_k, \mathbf{u}_k)$$

$$\text{s.t.} \quad \mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k)$$

$$\mathbf{h}(\mathbf{x}_k, \mathbf{u}_k) \leq 0$$

$$\mathbf{x}_0 = \mathbf{s}$$

apply **action** $\mathbf{a} = \mathbf{u}_0^*$ to the system



MPC

- is **based on planning** the future
- **Policy** from repeated planning

$$\pi^{\text{MPC}}(\mathbf{s}) = \mathbf{u}_0^*$$

Model Predictive Control (MPC)

Optimize a plan over finite horizon, apply first move, repeat

MPC: at **current state** $\mathbf{s}$

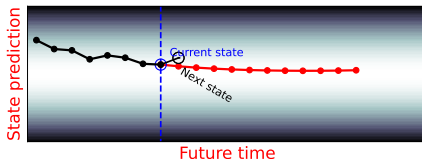
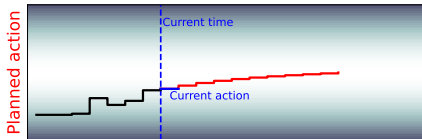
$$\min_{\mathbf{x}, \mathbf{u}} \quad T(\mathbf{x}_N) + \sum_{k=0}^{N-1} L(\mathbf{x}_k, \mathbf{u}_k)$$

$$\text{s.t.} \quad \mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k)$$

$$\mathbf{h}(\mathbf{x}_k, \mathbf{u}_k) \leq 0$$

$$\mathbf{x}_0 = \mathbf{s}$$

apply **action** $\mathbf{a} = \mathbf{u}_0^*$ to the system



MPC

- is **based on planning** the future
- **Policy** from repeated planning

$$\pi^{\text{MPC}}(\mathbf{s}) = \mathbf{u}_0^*$$

What an odd thing to do: we build and throw away plans all the time knowing that they are wrong all along, but kinda use them anyway...

Model Predictive Control (MPC)

Optimize a plan over finite horizon, apply first move, repeat

MPC: at **current state** s

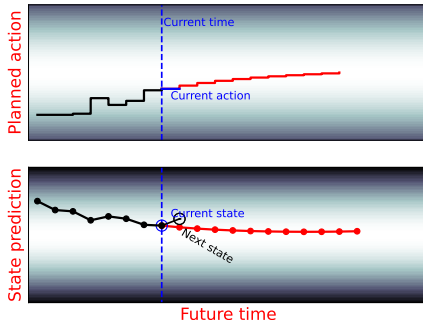
$$\min_{\mathbf{x}, \mathbf{u}} \quad T(\mathbf{x}_N) + \sum_{k=0}^{N-1} L(\mathbf{x}_k, \mathbf{u}_k)$$

$$\text{s.t.} \quad \mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k)$$

$$\mathbf{h}(\mathbf{x}_k, \mathbf{u}_k) \leq 0$$

$$\mathbf{x}_0 = \mathbf{s}$$

apply **action** $\mathbf{a} = \mathbf{u}_0^*$ to the system



MPC

- is **based on planning** the future
- **Policy** from repeated planning

$$\pi^{\text{MPC}}(s) = \mathbf{u}_0^*$$

MPC is a powerful tool to control constrained systems, increasingly used as a practical way of building optimal policies (more on that soon)

Theoretical Framework to connect RL and MPC

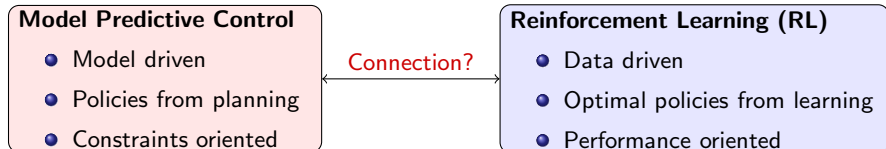
Model Predictive Control

- Model driven
- Policies from planning
- Constraints oriented

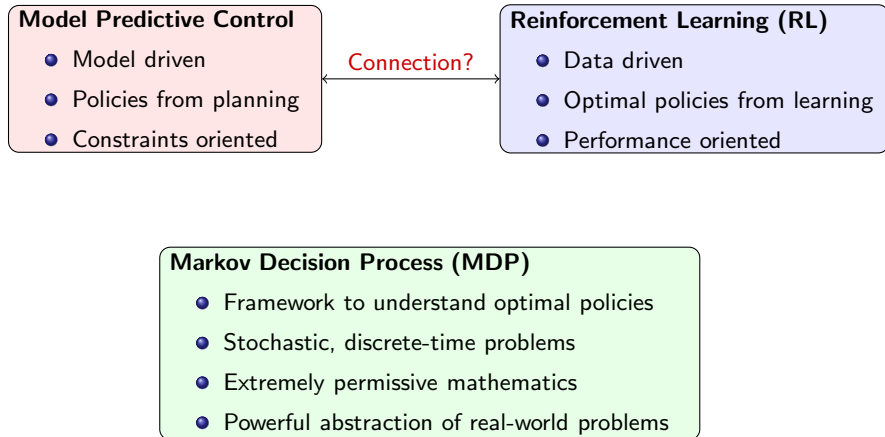
Reinforcement Learning (RL)

- Data driven
- Optimal policies from learning
- Performance oriented

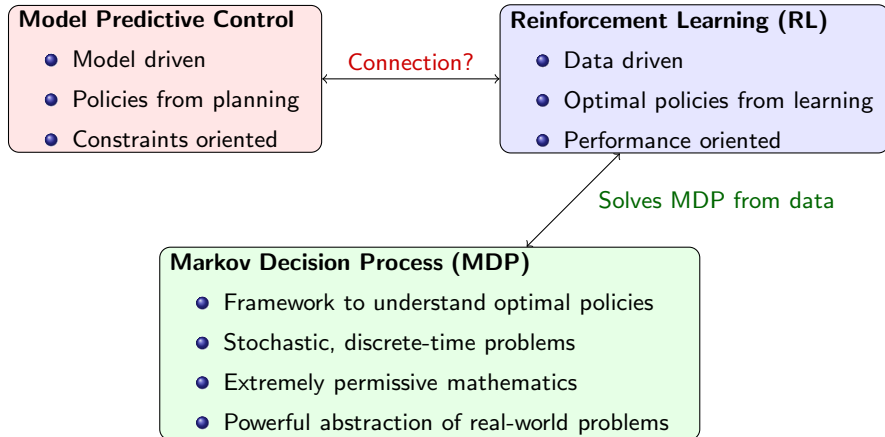
Theoretical Framework to connect RL and MPC



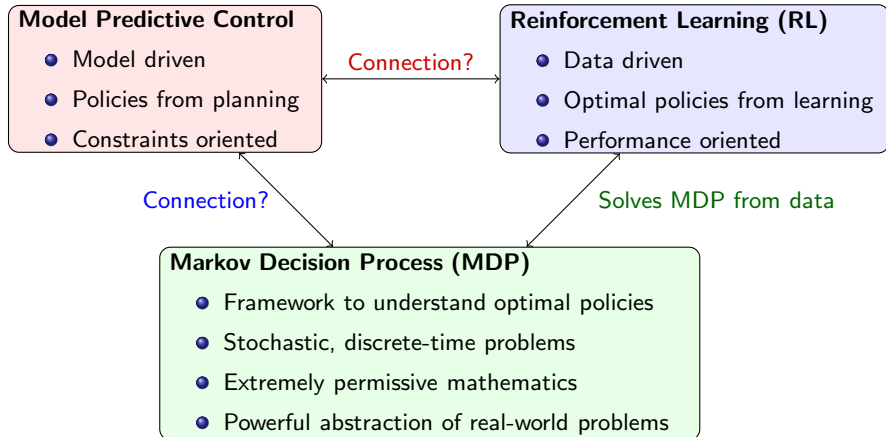
Theoretical Framework to connect RL and MPC



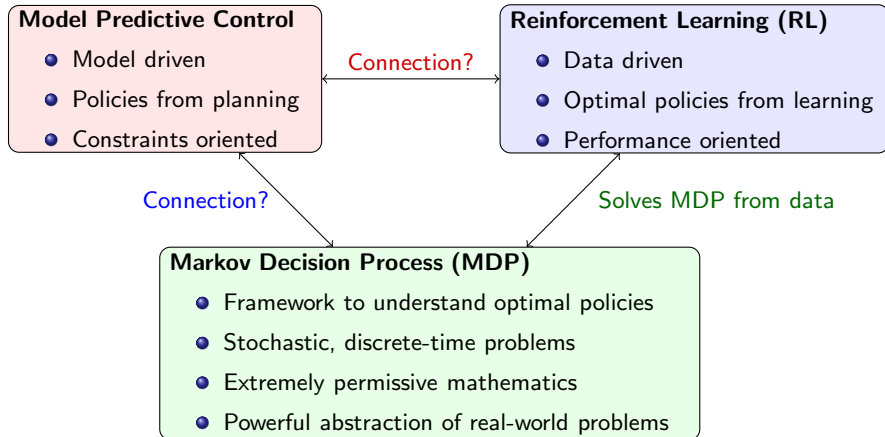
Theoretical Framework to connect RL and MPC



Theoretical Framework to connect RL and MPC

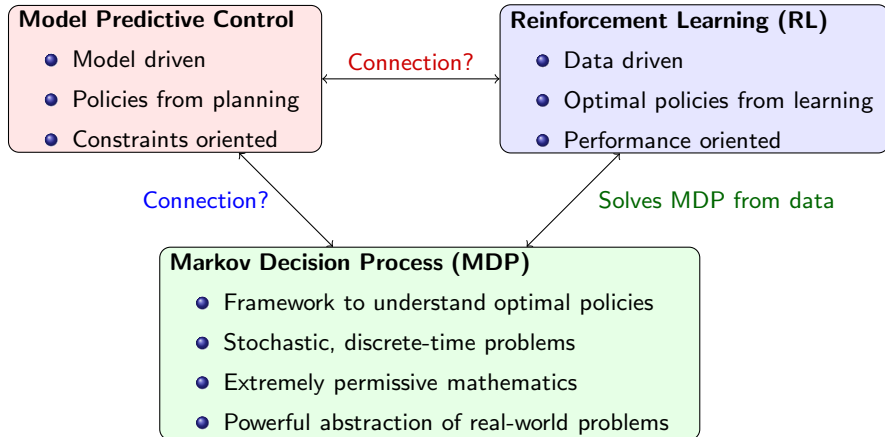


Theoretical Framework to connect RL and MPC



These lectures have connected MPC and RL from an “implementation” point of view

Theoretical Framework to connect RL and MPC



These lectures have connected MPC and RL from an “implementation” point of view

But understanding what we are doing is about connecting MPC to MDPs!

Markov Decision Processes (MDP)

Stochastic state transitions (real world)

$$\mathbf{s}, \mathbf{a} \rightarrow \mathbf{s}_+$$

(state-action $\rightarrow$ next state)

Cost function (instant performance)

$$L(\mathbf{s}, \mathbf{a}) \in \mathbb{R}$$

A (somewhat) general way of describing optimal control

Markov Decision Processes (MDP)

Stochastic state transitions (real world)

$$\mathbf{s}, \mathbf{a} \rightarrow \mathbf{s}_+$$

(state-action $\rightarrow$ next state)

Cost function (instant performance)

$$L(\mathbf{s}, \mathbf{a}) \in \mathbb{R}$$

- **Policy**

$$\mathbf{a} = \pi(\mathbf{s})$$

is how we act on the system

A (somewhat) general way of describing optimal control

Markov Decision Processes (MDP)

Stochastic state transitions (real world)

$$\mathbf{s}, \mathbf{a} \rightarrow \mathbf{s}_+$$

(state-action $\rightarrow$ next state)

Cost function (instant performance)

$$L(\mathbf{s}, \mathbf{a}) \in \mathbb{R}$$

- **Policy**

$$\mathbf{a} = \pi(\mathbf{s})$$

is how we act on the system

- **Closed-loop performance**

$$J(\pi) = \mathbb{E} \left[\sum_{k=0}^{\infty} \gamma^k L(\mathbf{s}_k, \mathbf{a}_k) \mid \pi \right]$$

with discount $\gamma \in [0, 1]$

A (somewhat) general way of describing optimal control

Markov Decision Processes (MDP)

Stochastic state transitions (real world)

$$s, a \rightarrow s_+$$

(state-action $\rightarrow$ next state)

- Policy

$$a = \pi(s)$$

is how we act on the system

- Closed-loop performance

$$J(\pi) = \mathbb{E} \left[\sum_{k=0}^{\infty} \gamma^k L(s_k, a_k) \mid \pi \right]$$

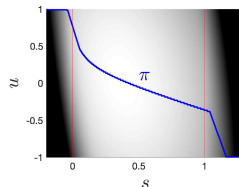
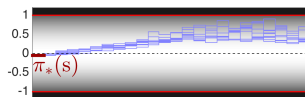
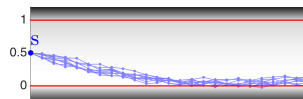
with discount $\gamma \in [0, 1]$

- Optimal policy: π^* from

$$\min_{\pi} J(\pi)$$

Cost function (instant performance)

$$L(s, a) \in \mathbb{R}$$



A (somewhat) general way of describing optimal control

Markov Decision Processes (MDP)

Stochastic state transitions (real world)

$$s, a \rightarrow s_+$$

(state-action $\rightarrow$ next state)

- **Policy**

$$a = \pi(s)$$

is how we act on the system

- **Closed-loop performance**

$$J(\pi) = \mathbb{E} \left[\sum_{k=0}^{\infty} \gamma^k L(s_k, a_k) \mid \pi \right]$$

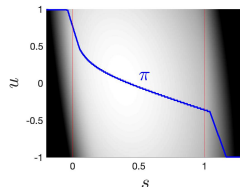
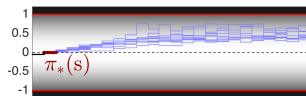
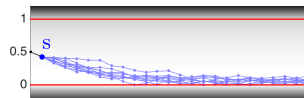
with discount $\gamma \in [0, 1]$

- **Optimal policy:** π^* from

$$\min_{\pi} J(\pi)$$

Cost function (instant performance)

$$L(s, a) \in \mathbb{R}$$



A (somewhat) general way of describing optimal control

Markov Decision Processes (MDP)

Stochastic state transitions (real world)

$$s, a \rightarrow s_+$$

(state-action $\rightarrow$ next state)

- **Policy**

$$a = \pi(s)$$

is how we act on the system

- **Closed-loop performance**

$$J(\pi) = \mathbb{E} \left[\sum_{k=0}^{\infty} \gamma^k L(s_k, a_k) \mid \pi \right]$$

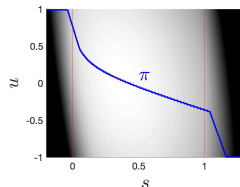
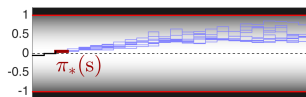
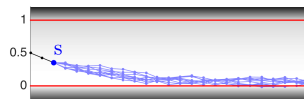
with discount $\gamma \in [0, 1]$

- **Optimal policy:** π^* from

$$\min_{\pi} J(\pi)$$

Cost function (instant performance)

$$L(s, a) \in \mathbb{R}$$



A (somewhat) general way of describing optimal control

Markov Decision Processes (MDP)

Stochastic state transitions (real world)

$$s, a \rightarrow s_+$$

(state-action $\rightarrow$ next state)

- **Policy**

$$a = \pi(s)$$

is how we act on the system

- **Closed-loop performance**

$$J(\pi) = \mathbb{E} \left[\sum_{k=0}^{\infty} \gamma^k L(s_k, a_k) \mid \pi \right]$$

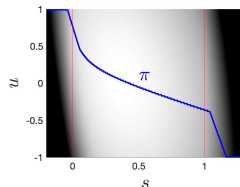
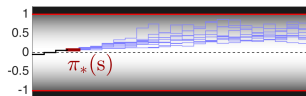
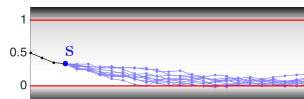
with discount $\gamma \in [0, 1]$

- **Optimal policy:** π^* from

$$\min_{\pi} J(\pi)$$

Cost function (instant performance)

$$L(s, a) \in \mathbb{R}$$



A (somewhat) general way of describing optimal control

Markov Decision Processes (MDP)

Stochastic state transitions (real world)

$$s, a \rightarrow s_+$$

(state-action $\rightarrow$ next state)

- **Policy**

$$a = \pi(s)$$

is how we act on the system

- **Closed-loop performance**

$$J(\pi) = \mathbb{E} \left[\sum_{k=0}^{\infty} \gamma^k L(s_k, a_k) \mid \pi \right]$$

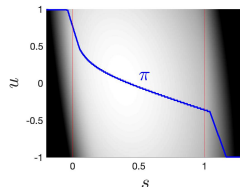
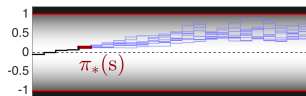
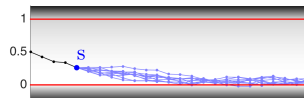
with discount $\gamma \in [0, 1]$

- **Optimal policy:** π^* from

$$\min_{\pi} J(\pi)$$

Cost function (instant performance)

$$L(s, a) \in \mathbb{R}$$



A (somewhat) general way of describing optimal control

Markov Decision Processes (MDP)

Stochastic state transitions (real world)

$$s, a \rightarrow s_+$$

(state-action $\rightarrow$ next state)

- **Policy**

$$a = \pi(s)$$

is how we act on the system

- **Closed-loop performance**

$$J(\pi) = \mathbb{E} \left[\sum_{k=0}^{\infty} \gamma^k L(s_k, a_k) \mid \pi \right]$$

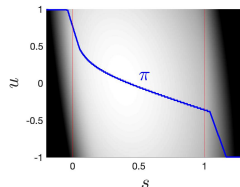
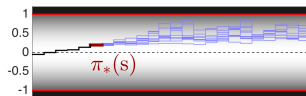
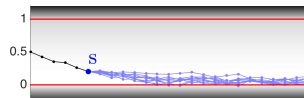
with discount $\gamma \in [0, 1]$

- **Optimal policy:** π^* from

$$\min_{\pi} J(\pi)$$

Cost function (instant performance)

$$L(s, a) \in \mathbb{R}$$



A (somewhat) general way of describing optimal control

Markov Decision Processes (MDP)

Stochastic state transitions (real world)

$$s, a \rightarrow s_+$$

(state-action $\rightarrow$ next state)

- **Policy**

$$a = \pi(s)$$

is how we act on the system

- **Closed-loop performance**

$$J(\pi) = \mathbb{E} \left[\sum_{k=0}^{\infty} \gamma^k L(s_k, a_k) \mid \pi \right]$$

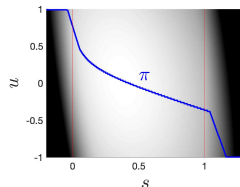
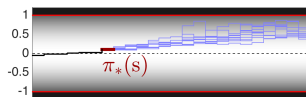
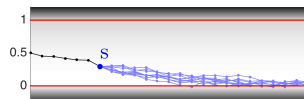
with discount $\gamma \in [0, 1]$

- **Optimal policy:** π^* from

$$\min_{\pi} J(\pi)$$

Cost function (instant performance)

$$L(s, a) \in \mathbb{R}$$



A (somewhat) general way of describing optimal control

Markov Decision Processes (MDP)

Stochastic state transitions (real world)

$$s, a \rightarrow s_+$$

(state-action $\rightarrow$ next state)

- **Policy**

$$a = \pi(s)$$

is how we act on the system

- **Closed-loop performance**

$$J(\pi) = \mathbb{E} \left[\sum_{k=0}^{\infty} \gamma^k L(s_k, a_k) \mid \pi \right]$$

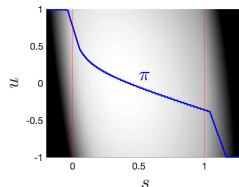
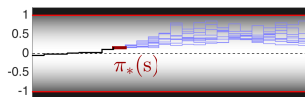
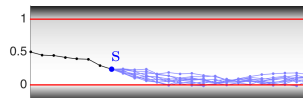
with discount $\gamma \in [0, 1]$

- **Optimal policy:** π^* from

$$\min_{\pi} J(\pi)$$

Cost function (instant performance)

$$L(s, a) \in \mathbb{R}$$



A (somewhat) general way of describing optimal control

Markov Decision Processes (MDP)

Stochastic state transitions (real world)

$$s, a \rightarrow s_+$$

(state-action $\rightarrow$ next state)

- **Policy**

$$a = \pi(s)$$

is how we act on the system

- **Closed-loop performance**

$$J(\pi) = \mathbb{E} \left[\sum_{k=0}^{\infty} \gamma^k L(s_k, a_k) \mid \pi \right]$$

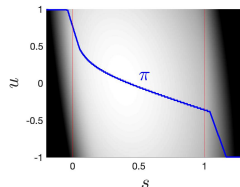
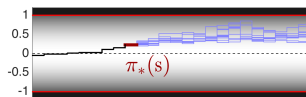
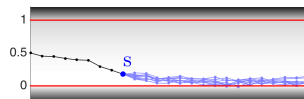
with discount $\gamma \in [0, 1]$

- **Optimal policy:** π^* from

$$\min_{\pi} J(\pi)$$

Cost function (instant performance)

$$L(s, a) \in \mathbb{R}$$



A (somewhat) general way of describing optimal control

Markov Decision Processes (MDP)

Stochastic state transitions (real world)

$$s, a \rightarrow s_+$$

(state-action $\rightarrow$ next state)

- **Policy**

$$a = \pi(s)$$

is how we act on the system

- **Closed-loop performance**

$$J(\pi) = \mathbb{E} \left[\sum_{k=0}^{\infty} \gamma^k L(s_k, a_k) \mid \pi \right]$$

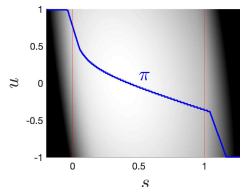
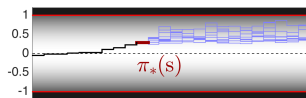
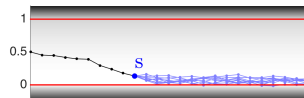
with discount $\gamma \in [0, 1]$

- **Optimal policy:** π^* from

$$\min_{\pi} J(\pi)$$

Cost function (instant performance)

$$L(s, a) \in \mathbb{R}$$



A (somewhat) general way of describing optimal control

Markov Decision Processes (MDP)

Stochastic state transitions (real world)

$$s, a \rightarrow s_+$$

(state-action $\rightarrow$ next state)

- **Policy**

$$a = \pi(s)$$

is how we act on the system

- **Closed-loop performance**

$$J(\pi) = \mathbb{E} \left[\sum_{k=0}^{\infty} \gamma^k L(s_k, a_k) \mid \pi \right]$$

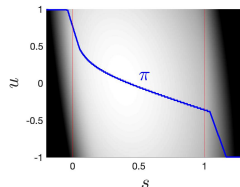
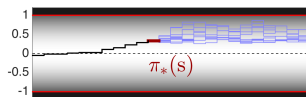
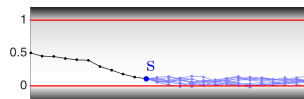
with discount $\gamma \in [0, 1]$

- **Optimal policy:** π^* from

$$\min_{\pi} J(\pi)$$

Cost function (instant performance)

$$L(s, a) \in \mathbb{R}$$



A (somewhat) general way of describing optimal control

Markov Decision Processes (MDP)

Stochastic state transitions (real world)

$$s, a \rightarrow s_+$$

(state-action $\rightarrow$ next state)

- **Policy**

$$a = \pi(s)$$

is how we act on the system

- **Closed-loop performance**

$$J(\pi) = \mathbb{E} \left[\sum_{k=0}^{\infty} \gamma^k L(s_k, a_k) \mid \pi \right]$$

with discount $\gamma \in [0, 1]$

- **Optimal policy:** π^* from

$$\min_{\pi} J(\pi)$$

Cost function (instant performance)

$$L(s, a) \in \mathbb{R}$$

Impose hard constraints $h(s, a) \leq 0$?

$$L(s, a) = \begin{cases} \ell(s, a) & \text{if } h(s, a) \leq 0 \\ \infty & \text{if } h(s, a) > 0 \end{cases}$$

I will use the same view in MPC for a bit

A (somewhat) general way of describing optimal control

Markov Decision Processes (MDP)

Stochastic state transitions (real world)

$$\mathbf{s}, \mathbf{a} \rightarrow \mathbf{s}_+$$

(state-action $\rightarrow$ next state)

- **Policy**

$$\mathbf{a} = \pi(\mathbf{s})$$

is how we act on the system

- **Closed-loop performance**

$$J(\pi) = \mathbb{E} \left[\sum_{k=0}^{\infty} \gamma^k L(\mathbf{s}_k, \mathbf{a}_k) \mid \pi \right]$$

with discount $\gamma \in [0, 1]$

- **Optimal policy:** π^* from

$$\min_{\pi} J(\pi)$$

Cost function (instant performance)

$$L(\mathbf{s}, \mathbf{a}) \in \mathbb{R}$$

Impose hard constraints $\mathbf{h}(\mathbf{s}, \mathbf{a}) \leq 0$?

$$L(\mathbf{s}, \mathbf{a}) = \begin{cases} \ell(\mathbf{s}, \mathbf{a}) & \text{if } \mathbf{h}(\mathbf{s}, \mathbf{a}) \leq 0 \\ \infty & \text{if } \mathbf{h}(\mathbf{s}, \mathbf{a}) > 0 \end{cases}$$

I will use the same view in MPC for a bit

MDP is a go-to framework when considering expected-value-based optimal control problems, useful for applications with stochastic dynamics.

A (somewhat) general way of describing optimal control

Markov Decision Processes (MDP)

Stochastic state transitions (real world)

$$s, a \rightarrow s_+$$

(state-action $\rightarrow$ next state)

- **Policy**

$$a = \pi(s)$$

is how we act on the system

- **Closed-loop performance**

$$J(\pi) = \mathbb{E} \left[\sum_{k=0}^{\infty} \gamma^k L(s_k, a_k) \mid \pi \right]$$

with discount $\gamma \in [0, 1]$

- **Optimal policy:** π^* from

$$\min_{\pi} J(\pi)$$

Cost function (instant performance)

$$L(s, a) \in \mathbb{R}$$

Impose hard constraints $h(s, a) \leq 0$?

$$L(s, a) = \begin{cases} \ell(s, a) & \text{if } h(s, a) \leq 0 \\ \infty & \text{if } h(s, a) > 0 \end{cases}$$

I will use the same view in MPC for a bit

MDP is a go-to framework when considering expected-value-based optimal control problems, useful for applications with stochastic dynamics.

Solution of an MDP is described by “simple” equations, but solving them is very challenging

A (somewhat) general way of describing optimal control

From Policy to Repeated Planning (MPC)

Infinite horizon & discounted

$$\pi_{\infty}^* = \arg \min_{\pi} \mathbb{E} \left[\sum_{k=0}^{\infty} \gamma^k L(\mathbf{x}_k, \mathbf{a}_k) \right]$$

Policy π : state $\rightarrow$ action
belongs to a function space

*Let's transform an MDP into an MPC and
understand the approximations we make*

From Policy to Repeated Planning (MPC)

Infinite horizon & discounted

$$\pi_{\infty}^* = \arg \min_{\pi} \mathbb{E} \left[\sum_{k=0}^{\infty} \gamma^k L(\mathbf{x}_k, \mathbf{a}_k) \right]$$

Policy π : state $\rightarrow$ action
belongs to a function space

*Let's transform an MDP into an MPC and
understand the approximations we make*

Finite-horizon equivalent:

$$\pi_{0,\dots,N-1}^* = \arg \min_{\pi_{0,\dots,N-1}} \mathbb{E} \left[T(s_N) + \sum_{k=0}^{N-1} \gamma^k L(s_k, \mathbf{a}_k) \right]$$

If $T = \gamma^N V^*$, then $\pi_{0,\dots,N-1}^* = \pi_{\infty}^*$

From Policy to Repeated Planning (MPC)

Infinite horizon & discounted

$$\pi_{\infty}^* = \arg \min_{\pi} \mathbb{E} \left[\sum_{k=0}^{\infty} \gamma^k L(\mathbf{x}_k, \mathbf{a}_k) \right]$$

Policy π : state $\rightarrow$ action
belongs to a function space

*Let's transform an MDP into an MPC and
understand the approximations we make*

Finite-horizon equivalent:

$$\pi_{0,\dots,N-1}^* = \arg \min_{\pi_{0,\dots,N-1}} \mathbb{E} \left[T(s_N) + \sum_{k=0}^{N-1} \gamma^k L(s_k, \mathbf{a}_k) \right]$$

If $T = \gamma^N V^*$, then $\pi_{0,\dots,N-1}^* = \pi_{\infty}^*$

Planning instead of a policy:

$$\min_{\mathbf{a}_{0,\dots,N-1}} \mathbb{E} \left[T(s_N) + \sum_{k=0}^{N-1} \gamma^k L(s_k, \mathbf{a}_k) \mid \mathbf{s}_0 = \mathbf{s} \right]$$

i.e. **restrict policies** to fixed $\mathbf{a}_{0,\dots,N-1}$

From Policy to Repeated Planning (MPC)

Infinite horizon & discounted

$$\pi_{\infty}^* = \arg \min_{\pi} \mathbb{E} \left[\sum_{k=0}^{\infty} \gamma^k L(\mathbf{x}_k, \mathbf{a}_k) \right]$$

Policy π : state $\rightarrow$ action
belongs to a function space

Let's transform an MDP into an MPC and understand the approximations we make

Finite-horizon equivalent:

$$\pi_{0,\dots,N-1}^* = \arg \min_{\pi_{0,\dots,N-1}} \mathbb{E} \left[T(s_N) + \sum_{k=0}^{N-1} \gamma^k L(s_k, \mathbf{a}_k) \right]$$

If $T = \gamma^N V^*$, then $\pi_{0,\dots,N-1}^* = \pi_{\infty}^*$

Planning instead of a policy:

$$\min_{\mathbf{a}_{0,\dots,N-1}} \mathbb{E} \left[T(s_N) + \sum_{k=0}^{N-1} \gamma^k L(s_k, \mathbf{a}_k) \mid \mathbf{s}_0 = \mathbf{s} \right]$$

i.e. **restrict policies** to fixed $\mathbf{a}_{0,\dots,N-1}$

Deterministic model, policy

$$\min_{\pi_{0,\dots,N-1}} T(s_N) + \sum_{k=0}^{N-1} \gamma^k L(s_k, \mathbf{a}_k)$$

$$\text{s.t. } s_{k+1} = \mathbf{f}(s_k, \mathbf{a}_k)$$

$$s_0 = \mathbf{s}$$

i.e. adopt **deterministic model**

From Policy to Repeated Planning (MPC)

Infinite horizon & discounted

$$\pi_{\infty}^* = \arg \min_{\pi} \mathbb{E} \left[\sum_{k=0}^{\infty} \gamma^k L(\mathbf{x}_k, \mathbf{a}_k) \right]$$

Policy π : state $\rightarrow$ action
belongs to a function space

Let's transform an MDP into an MPC and understand the approximations we make

Finite-horizon equivalent:

$$\pi_{0,\dots,N-1}^* = \arg \min_{\pi_{0,\dots,N-1}} \mathbb{E} \left[T(s_N) + \sum_{k=0}^{N-1} \gamma^k L(s_k, \mathbf{a}_k) \right]$$

If $T = \gamma^N V^*$, then $\pi_{0,\dots,N-1}^* = \pi_{\infty}^*$

Why attacking the problem in these ways?

Planning instead of a policy:

$$\min_{\mathbf{a}_0,\dots,\mathbf{a}_{N-1}} \mathbb{E} \left[T(s_N) + \sum_{k=0}^{N-1} \gamma^k L(s_k, \mathbf{a}_k) \mid \mathbf{s}_0 = \mathbf{s} \right]$$

i.e. **restrict policies** to fixed $\mathbf{a}_0,\dots,\mathbf{a}_{N-1}$

Deterministic model, planning

$$\min_{\mathbf{a}_0,\dots,\mathbf{a}_{N-1}} T(s_N) + \sum_{k=0}^{N-1} \gamma^k L(s_k, \mathbf{a}_k)$$

$$\text{s.t. } \mathbf{s}_{k+1} = \mathbf{f}(\mathbf{s}_k, \mathbf{a}_k)$$

$$\mathbf{s}_0 = \mathbf{s}$$

i.e. adopt **deterministic model**

MPC as a policy

Deterministic MPC:

$$\begin{aligned} \min_{\mathbf{u}_0, \dots, \mathbf{u}_{N-1}} \quad & T(\mathbf{x}_N) + \sum_{k=0}^{N-1} \gamma^k L(\mathbf{x}_k, \mathbf{u}_k) \\ \text{s.t.} \quad & \mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k) \\ & \mathbf{x}_0 = \mathbf{s} \end{aligned}$$

Defines policy:

$$\pi^{\text{MPC}}(\mathbf{s}) = \mathbf{u}_0^*$$

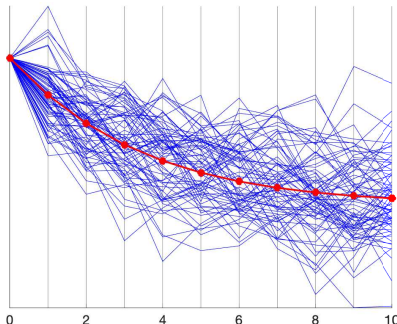
How does π^{MPC} relate to π^* ?

No reason to match:

- Planning rather than policing
- Model approximates stochasticity, often deterministic

Infinite horizon & discounted

$$\pi_\infty^* = \arg \min_{\pi} \mathbb{E} \left[\sum_{k=0}^{\infty} \gamma^k L(\mathbf{s}_k, \mathbf{a}_k) \right]$$



MPC as a policy

Deterministic MPC:

$$\begin{aligned} \min_{\mathbf{u}_0, \dots, \mathbf{u}_{N-1}} \quad & T(\mathbf{x}_N) + \sum_{k=0}^{N-1} \gamma^k L(\mathbf{x}_k, \mathbf{u}_k) \\ \text{s.t.} \quad & \mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k) \\ & \mathbf{x}_0 = \mathbf{s} \end{aligned}$$

Defines policy:

$$\pi^{\text{MPC}}(\mathbf{s}) = \mathbf{u}_0^*$$

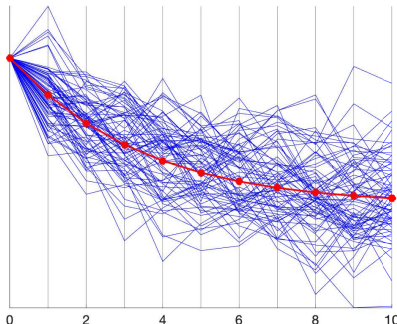
How does π^{MPC} relate to π^* ?

No reason to match:

- Planning rather than policing
- Model approximates stochasticity, often deterministic

Infinite horizon & discounted

$$\pi_{\infty}^* = \arg \min_{\pi} \mathbb{E} \left[\sum_{k=0}^{\infty} \gamma^k L(\mathbf{s}_k, \mathbf{a}_k) \right]$$



MPC as a policy

Deterministic MPC:

$$\begin{aligned} \min_{\mathbf{u}_0, \dots, \mathbf{u}_{N-1}} \quad & T(\mathbf{x}_N) + \sum_{k=0}^{N-1} \gamma^k L(\mathbf{x}_k, \mathbf{u}_k) \\ \text{s.t.} \quad & \mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k) \\ & \mathbf{x}_0 = \mathbf{s} \end{aligned}$$

Defines policy:

$$\pi^{\text{MPC}}(\mathbf{s}) = \mathbf{u}_0^*$$

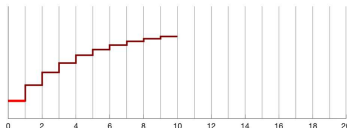
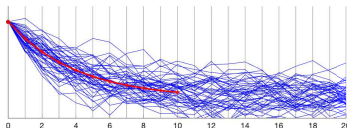
How does π^{MPC} relate to π^* ?

No reason to match:

- Planning rather than policing
- Model approximates stochasticity, often deterministic

Infinite horizon & discounted

$$\pi_\infty^* = \arg \min_{\pi} \mathbb{E} \left[\sum_{k=0}^{\infty} \gamma^k L(\mathbf{s}_k, \mathbf{a}_k) \right]$$



MPC as a policy

Deterministic MPC:

$$\begin{aligned} \min_{\mathbf{u}_0, \dots, \mathbf{u}_{N-1}} \quad & T(\mathbf{x}_N) + \sum_{k=0}^{N-1} \gamma^k L(\mathbf{x}_k, \mathbf{u}_k) \\ \text{s.t.} \quad & \mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k) \\ & \mathbf{x}_0 = \mathbf{s} \end{aligned}$$

Defines policy:

$$\pi^{\text{MPC}}(\mathbf{s}) = \mathbf{u}_0^*$$

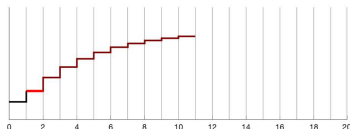
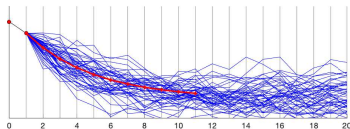
How does π^{MPC} relate to π^* ?

No reason to match:

- Planning rather than policing
- Model approximates stochasticity, often deterministic

Infinite horizon & discounted

$$\pi_\infty^* = \arg \min_{\pi} \mathbb{E} \left[\sum_{k=0}^{\infty} \gamma^k L(\mathbf{s}_k, \mathbf{a}_k) \right]$$



MPC as a policy

Deterministic MPC:

$$\begin{aligned} \min_{\mathbf{u}_0, \dots, \mathbf{u}_{N-1}} \quad & T(\mathbf{x}_N) + \sum_{k=0}^{N-1} \gamma^k L(\mathbf{x}_k, \mathbf{u}_k) \\ \text{s.t.} \quad & \mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k) \\ & \mathbf{x}_0 = \mathbf{s} \end{aligned}$$

Defines policy:

$$\pi^{\text{MPC}}(\mathbf{s}) = \mathbf{u}_0^*$$

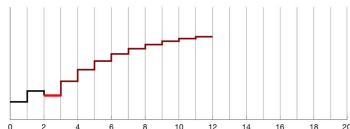
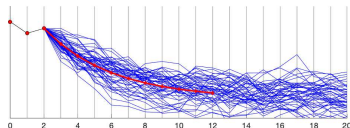
How does π^{MPC} relate to π^* ?

No reason to match:

- Planning rather than policing
- Model approximates stochasticity, often deterministic

Infinite horizon & discounted

$$\pi_\infty^* = \arg \min_{\pi} \mathbb{E} \left[\sum_{k=0}^{\infty} \gamma^k L(\mathbf{s}_k, \mathbf{a}_k) \right]$$



MPC as a policy

Deterministic MPC:

$$\begin{aligned} \min_{\mathbf{u}_0, \dots, \mathbf{u}_{N-1}} \quad & T(\mathbf{x}_N) + \sum_{k=0}^{N-1} \gamma^k L(\mathbf{x}_k, \mathbf{u}_k) \\ \text{s.t.} \quad & \mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k) \\ & \mathbf{x}_0 = \mathbf{s} \end{aligned}$$

Defines policy:

$$\pi^{\text{MPC}}(\mathbf{s}) = \mathbf{u}_0^*$$

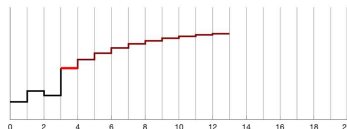
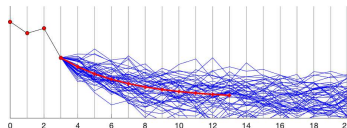
How does π^{MPC} relate to π^* ?

No reason to match:

- Planning rather than policing
- Model approximates stochasticity, often deterministic

Infinite horizon & discounted

$$\pi_\infty^* = \arg \min_{\pi} \mathbb{E} \left[\sum_{k=0}^{\infty} \gamma^k L(\mathbf{s}_k, \mathbf{a}_k) \right]$$



MPC as a policy

Deterministic MPC:

$$\begin{aligned} \min_{\mathbf{u}_0, \dots, \mathbf{u}_{N-1}} \quad & T(\mathbf{x}_N) + \sum_{k=0}^{N-1} \gamma^k L(\mathbf{x}_k, \mathbf{u}_k) \\ \text{s.t.} \quad & \mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k) \\ & \mathbf{x}_0 = \mathbf{s} \end{aligned}$$

Defines policy:

$$\pi^{\text{MPC}}(\mathbf{s}) = \mathbf{u}_0^*$$

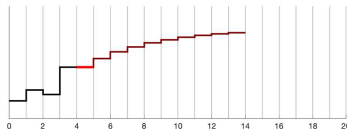
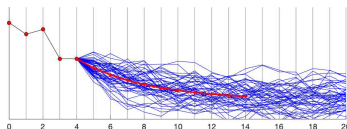
How does π^{MPC} relate to π^* ?

No reason to match:

- Planning rather than policing
- Model approximates stochasticity, often deterministic

Infinite horizon & discounted

$$\pi_\infty^* = \arg \min_{\pi} \mathbb{E} \left[\sum_{k=0}^{\infty} \gamma^k L(\mathbf{s}_k, \mathbf{a}_k) \right]$$



MPC as a policy

Deterministic MPC:

$$\begin{aligned} \min_{\mathbf{u}_0, \dots, \mathbf{u}_{N-1}} \quad & T(\mathbf{x}_N) + \sum_{k=0}^{N-1} \gamma^k L(\mathbf{x}_k, \mathbf{u}_k) \\ \text{s.t.} \quad & \mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k) \\ & \mathbf{x}_0 = \mathbf{s} \end{aligned}$$

Defines policy:

$$\pi^{\text{MPC}}(\mathbf{s}) = \mathbf{u}_0^*$$

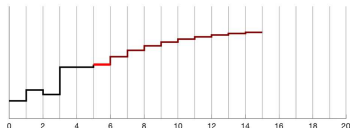
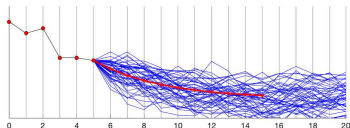
How does π^{MPC} relate to π^* ?

No reason to match:

- Planning rather than policing
- Model approximates stochasticity, often deterministic

Infinite horizon & discounted

$$\pi_\infty^* = \arg \min_{\pi} \mathbb{E} \left[\sum_{k=0}^{\infty} \gamma^k L(\mathbf{s}_k, \mathbf{a}_k) \right]$$



MPC as a policy

Deterministic MPC:

$$\begin{aligned} \min_{\mathbf{u}_0, \dots, \mathbf{u}_{N-1}} \quad & T(\mathbf{x}_N) + \sum_{k=0}^{N-1} \gamma^k L(\mathbf{x}_k, \mathbf{u}_k) \\ \text{s.t.} \quad & \mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k) \\ & \mathbf{x}_0 = \mathbf{s} \end{aligned}$$

Defines policy:

$$\pi^{\text{MPC}}(\mathbf{s}) = \mathbf{u}_0^*$$

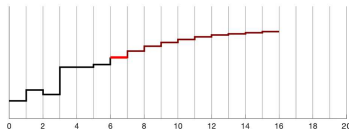
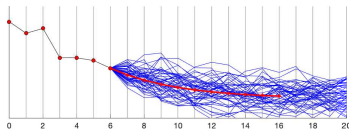
How does π^{MPC} relate to π^* ?

No reason to match:

- Planning rather than policing
- Model approximates stochasticity, often deterministic

Infinite horizon & discounted

$$\pi_\infty^* = \arg \min_{\pi} \mathbb{E} \left[\sum_{k=0}^{\infty} \gamma^k L(\mathbf{s}_k, \mathbf{a}_k) \right]$$



MPC as a policy

Deterministic MPC:

$$\begin{aligned} \min_{\mathbf{u}_0, \dots, \mathbf{u}_{N-1}} \quad & T(\mathbf{x}_N) + \sum_{k=0}^{N-1} \gamma^k L(\mathbf{x}_k, \mathbf{u}_k) \\ \text{s.t.} \quad & \mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k) \\ & \mathbf{x}_0 = \mathbf{s} \end{aligned}$$

Defines policy:

$$\pi^{\text{MPC}}(\mathbf{s}) = \mathbf{u}_0^*$$

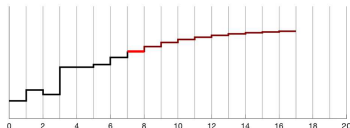
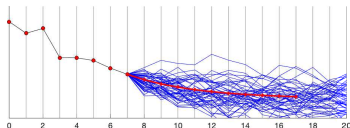
How does π^{MPC} relate to π^* ?

No reason to match:

- Planning rather than policing
- Model approximates stochasticity, often deterministic

Infinite horizon & discounted

$$\pi_{\infty}^* = \arg \min_{\pi} \mathbb{E} \left[\sum_{k=0}^{\infty} \gamma^k L(\mathbf{s}_k, \mathbf{a}_k) \right]$$



MPC as a policy

Deterministic MPC:

$$\begin{aligned} \min_{\mathbf{u}_0, \dots, \mathbf{u}_{N-1}} \quad & T(\mathbf{x}_N) + \sum_{k=0}^{N-1} \gamma^k L(\mathbf{x}_k, \mathbf{u}_k) \\ \text{s.t.} \quad & \mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k) \\ & \mathbf{x}_0 = \mathbf{s} \end{aligned}$$

Defines policy:

$$\pi^{\text{MPC}}(\mathbf{s}) = \mathbf{u}_0^*$$

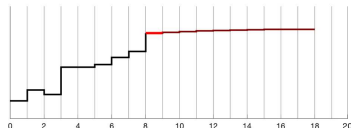
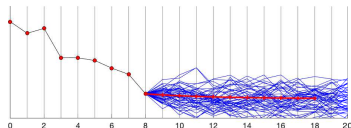
How does π^{MPC} relate to π^* ?

No reason to match:

- Planning rather than policing
- Model approximates stochasticity, often deterministic

Infinite horizon & discounted

$$\pi_\infty^* = \arg \min_{\pi} \mathbb{E} \left[\sum_{k=0}^{\infty} \gamma^k L(\mathbf{s}_k, \mathbf{a}_k) \right]$$



MPC as a policy

Deterministic MPC:

$$\begin{aligned} \min_{\mathbf{u}_0, \dots, \mathbf{u}_{N-1}} \quad & T(\mathbf{x}_N) + \sum_{k=0}^{N-1} \gamma^k L(\mathbf{x}_k, \mathbf{u}_k) \\ \text{s.t.} \quad & \mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k) \\ & \mathbf{x}_0 = \mathbf{s} \end{aligned}$$

Defines policy:

$$\pi^{\text{MPC}}(\mathbf{s}) = \mathbf{u}_0^*$$

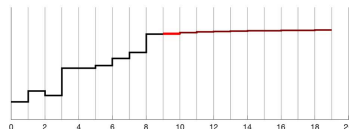
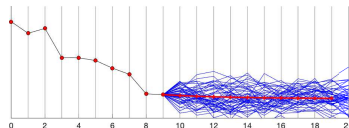
How does π^{MPC} relate to π^* ?

No reason to match:

- Planning rather than policing
- Model approximates stochasticity, often deterministic

Infinite horizon & discounted

$$\pi_\infty^* = \arg \min_{\pi} \mathbb{E} \left[\sum_{k=0}^{\infty} \gamma^k L(\mathbf{s}_k, \mathbf{a}_k) \right]$$



MPC as a policy

Deterministic MPC:

$$\begin{aligned} \min_{\mathbf{u}_0, \dots, \mathbf{u}_{N-1}} \quad & T(\mathbf{x}_N) + \sum_{k=0}^{N-1} \gamma^k L(\mathbf{x}_k, \mathbf{u}_k) \\ \text{s.t.} \quad & \mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k) \\ & \mathbf{x}_0 = \mathbf{s} \end{aligned}$$

Defines policy:

$$\pi^{\text{MPC}}(\mathbf{s}) = \mathbf{u}_0^*$$

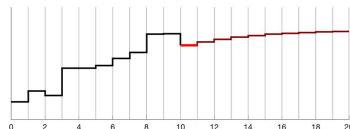
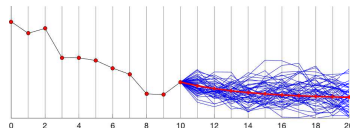
How does π^{MPC} relate to π^* ?

No reason to match:

- Planning rather than policing
- Model approximates stochasticity, often deterministic

Infinite horizon & discounted

$$\pi_\infty^* = \arg \min_{\pi} \mathbb{E} \left[\sum_{k=0}^{\infty} \gamma^k L(\mathbf{s}_k, \mathbf{a}_k) \right]$$



MPC as a policy

Deterministic MPC:

$$\begin{aligned} \min_{\mathbf{u}_0, \dots, \mathbf{u}_{N-1}} \quad & T(\mathbf{x}_N) + \sum_{k=0}^{N-1} \gamma^k L(\mathbf{x}_k, \mathbf{u}_k) \\ \text{s.t.} \quad & \mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k) \\ & \mathbf{x}_0 = \mathbf{s} \end{aligned}$$

Defines policy:

$$\pi^{\text{MPC}}(\mathbf{s}) = \mathbf{u}_0^*$$

How does π^{MPC} relate to π^* ?

No reason to match:

- Planning rather than policing
- Model approximates stochasticity, often deterministic

Infinite horizon & discounted

$$\pi_\infty^* = \arg \min_{\pi} \mathbb{E} \left[\sum_{k=0}^{\infty} \gamma^k L(\mathbf{s}_k, \mathbf{a}_k) \right]$$



MPC as a policy

Deterministic MPC:

$$\begin{aligned} \min_{\mathbf{u}_0, \dots, \mathbf{u}_{N-1}} \quad & T(\mathbf{x}_N) + \sum_{k=0}^{N-1} \gamma^k L(\mathbf{x}_k, \mathbf{u}_k) \\ \text{s.t.} \quad & \mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k) \\ & \mathbf{x}_0 = \mathbf{s} \end{aligned}$$

Defines policy:

$$\pi^{\text{MPC}}(\mathbf{s}) = \mathbf{u}_0^*$$

How does π^{MPC} relate to π^* ?

No reason to match:

- Planning rather than policing
- Model approximates stochasticity, often deterministic

Infinite horizon & discounted

$$\pi_{\infty}^* = \arg \min_{\pi} \mathbb{E} \left[\sum_{k=0}^{\infty} \gamma^k L(\mathbf{s}_k, \mathbf{a}_k) \right]$$



Can we clarify the relationship?

Some more context on MPC for performance

Historically MPC focuses on **constraints satisfaction & stability**. Cost is for reference tracking, not representative of the system performance.

Some more context on MPC for performance

Historically MPC focuses on **constraints satisfaction & stability**. Cost is for reference tracking, not representative of the system performance.

- “Tracking MPC”
- Classic stability theory
- Uncertainty via
 - ▶ Robust MPC
 - ▶ Stochastic MPC

Some more context on MPC for performance

Historically MPC focuses on **constraints satisfaction & stability**. Cost is for reference tracking, not representative of the system performance.

More recently, focus shifted to **closed-loop performance**, e.g. energy, time, money. Cost is generic, representative of the system performance.

- “Tracking MPC”
- Classic stability theory
- Uncertainty via
 - ▶ Robust MPC
 - ▶ Stochastic MPC

Some more context on MPC for performance

Historically MPC focuses on **constraints satisfaction & stability**. Cost is for reference tracking, not representative of the system performance.

More recently, focus shifted to **closed-loop performance**, e.g. energy, time, money. Cost is generic, representative of the system performance.

- “Tracking MPC”
- Classic stability theory
- Uncertainty via
 - ▶ Robust MPC
 - ▶ Stochastic MPC

- “Economic MPC”
- Dissipativity theory
- Uncertainty via
 - ▶ Robust MPC
 - ▶ Stochastic MPC
- MPC can optimize the system performance...

Some more context on MPC for performance

Historically MPC focuses on **constraints satisfaction & stability**. Cost is for reference tracking, not representative of the system performance.

More recently, focus shifted to **closed-loop performance**, e.g. energy, time, money. Cost is generic, representative of the system performance.

- “Tracking MPC”
- Classic stability theory
- Uncertainty via
 - ▶ Robust MPC
 - ▶ Stochastic MPC

- “Economic MPC”
- Dissipativity theory
- Uncertainty via
 - ▶ Robust MPC
 - ▶ Stochastic MPC
- MPC can optimize the system performance...

MPC for closed-loop performance

- is not a very old topic
- “historical robustness” of MPC does not hold in the presence of stochasticity

Some more context on MPC for performance

Historically MPC focuses on **constraints satisfaction & stability**. Cost is for reference tracking, not representative of the system performance.

More recently, focus shifted to **closed-loop performance**, e.g. energy, time, money. Cost is generic, representative of the system performance.

- “Tracking MPC”
- Classic stability theory
- Uncertainty via
 - ▶ Robust MPC
 - ▶ Stochastic MPC

- “Economic MPC”
- Dissipativity theory
- Uncertainty via
 - ▶ Robust MPC
 - ▶ Stochastic MPC
- MPC can optimize the system performance...

MPC for closed-loop performance

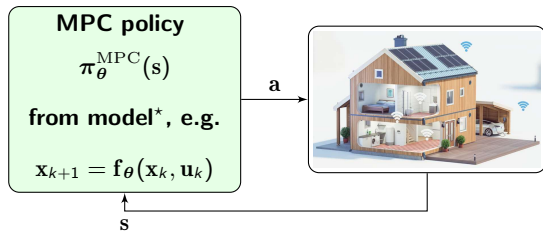
- is not a very old topic
- “historical robustness” of MPC does not hold in the presence of stochasticity

Soft claim: MPC can be used as a practical toolbox to model the solution of MDPs. This view is the “best way” for understanding what we are doing with economic MPC

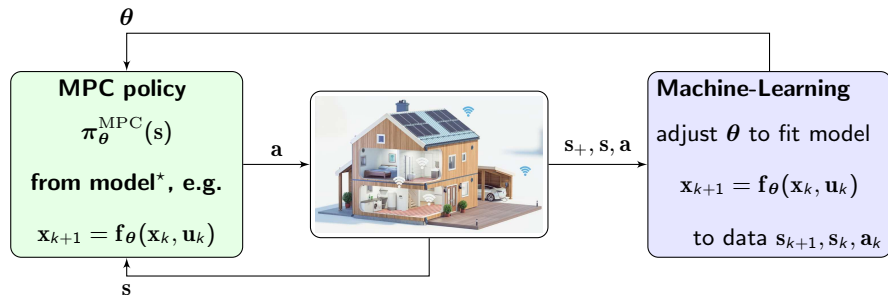
Outline

- 1 Let's rebuild some background – MPC and MDPs
- 2 MPC as a solution to MDPs
- 3 When is RL (most) beneficial for MPC?
- 4 A Deeper Look at the Theory

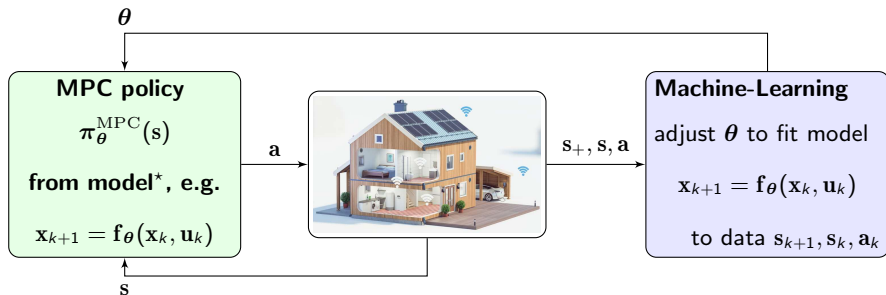
Learning for MPC - Machine Learning in-the-loop



Learning for MPC - Machine Learning in-the-loop



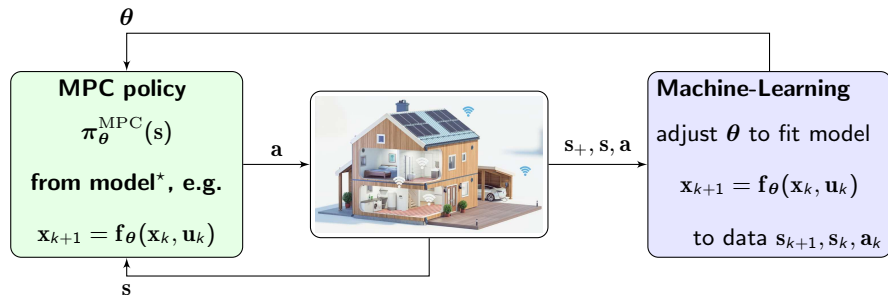
Learning for MPC - Machine Learning in-the-loop



“Machine-Learning” in-the-loop f_{θ} from

- **Physics-based:** first principles + SYSID
- **Neural Network:** DNN, LSTM, TFT, ...
- **Statistical:** GP, GPC, Probabilistic AI ...

Learning for MPC - Machine Learning in-the-loop

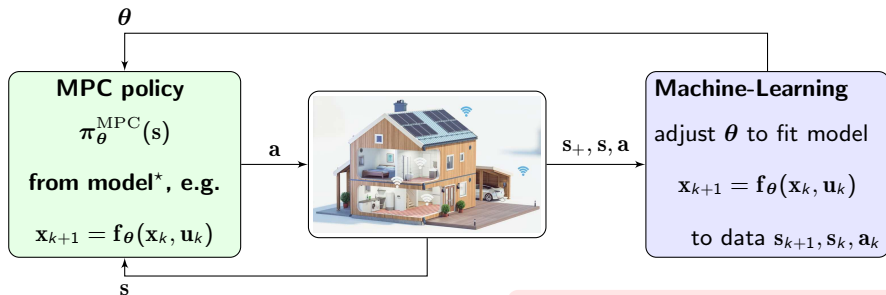


“Machine-Learning” in-the-loop f_{θ} from

- **Physics-based:** first principles + SYSID
- **Neural Network:** DNN, LSTM, TFT, ...
- **Statistical:** GP, GPC, Probabilistic AI ...

**can replace “model” by any prediction strategies:
input-output predictors, multi-step predictors, etc...*

Learning for MPC - Machine Learning in-the-loop



“Machine-Learning” in-the-loop f_{θ} from

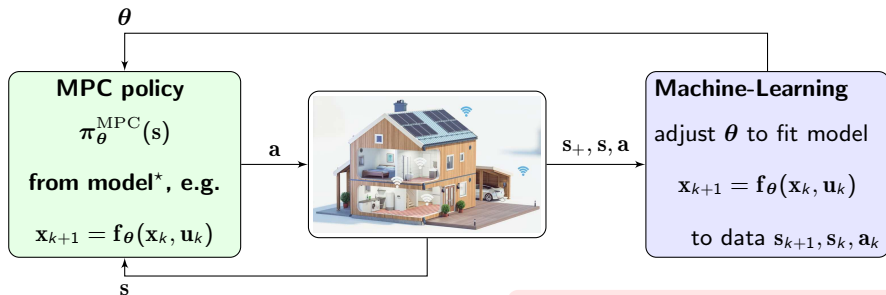
- **Physics-based:** first principles + SYSID
- **Neural Network:** DNN, LSTM, TFT, ...
- **Statistical:** GP, GPC, Probabilistic AI ...

Paradigm

- Performance tied to prediction accuracy
- Target accuracy via ML
- Ignore that MPC is a policy

**can replace “model” by any prediction strategies:
input-output predictors, multi-step predictors, etc...*

Learning for MPC - Machine Learning in-the-loop



“Machine-Learning” in-the-loop f_{θ} from

- **Physics-based:** first principles + SYSID
- **Neural Network:** DNN, LSTM, TFT, ...
- **Statistical:** GP, GPC, Probabilistic AI ...

**can replace “model” by any prediction strategies: input-output predictors, multi-step predictors, etc...*

Paradigm

- Performance tied to prediction accuracy
- Target accuracy via ML
- Ignore that MPC is a policy

We focus on “breaking” this paradigm
Learning / RL plays a key role

How can MPC model an MDP?

MDP optimal policy

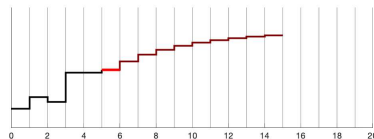
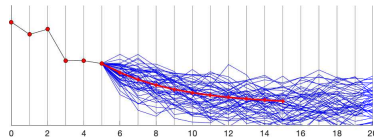
$$\pi^* = \arg \min_{\pi} \mathbb{E} \left[\sum_{k=0}^{\infty} \gamma^k L(\mathbf{x}_k, \mathbf{u}_k) \right]$$

is entirely described by $Q^*(s, \mathbf{a})$

MPC policy $\pi^{\text{MPC}}(s) = \mathbf{u}_0^*$ from

$$\min_{\mathbf{x}, \mathbf{u}} \quad T(\mathbf{x}_N) + \sum_{k=0}^{N-1} \gamma^k L(\mathbf{x}_k, \mathbf{u}_k)$$

$$\text{s.t.} \quad \mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k), \quad \mathbf{x}_0 = \mathbf{s} \\ \mathbf{h}(\mathbf{x}_k, \mathbf{u}_k) \leq 0$$



How can MPC model an MDP?

MDP optimal policy

$$\pi^* = \arg \min_{\pi} \mathbb{E} \left[\sum_{k=0}^{\infty} \gamma^k L(\mathbf{x}_k, \mathbf{u}_k) \right]$$

is entirely described by $Q^*(s, a)$

MPC as a model of the MDP

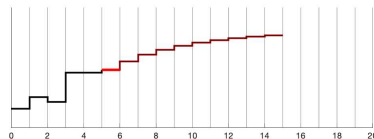
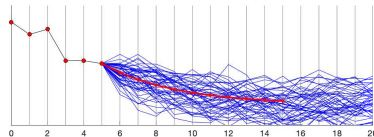
$$V^{\text{MPC}}(\mathbf{s}) := \min_{\mathbf{x}, \mathbf{u}} T(\mathbf{x}_N) + \sum_{k=0}^{N-1} \gamma^k L(\mathbf{x}_k, \mathbf{u}_k)$$
$$\text{s.t. } \mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k), \quad \mathbf{x}_0 = \mathbf{s}$$
$$\mathbf{h}(\mathbf{x}_k, \mathbf{u}_k) \leq 0$$

MPC policy $\pi^{\text{MPC}}(s) = \mathbf{u}_0^*$ from

$$\min_{\mathbf{x}, \mathbf{u}} T(\mathbf{x}_N) + \sum_{k=0}^{N-1} \gamma^k L(\mathbf{x}_k, \mathbf{u}_k)$$

$$\text{s.t. } \mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k), \quad \mathbf{x}_0 = \mathbf{s}$$

$$\mathbf{h}(\mathbf{x}_k, \mathbf{u}_k) \leq 0$$



How can MPC model an MDP?

MDP optimal policy

$$\pi^* = \arg \min_{\pi} \mathbb{E} \left[\sum_{k=0}^{\infty} \gamma^k L(\mathbf{x}_k, \mathbf{u}_k) \right]$$

is entirely described by $Q^*(s, a)$

MPC as a model of the MDP

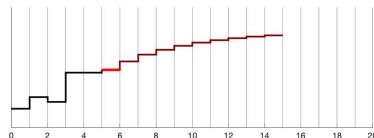
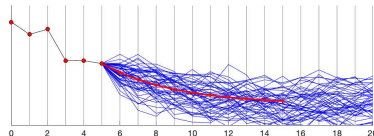
$$V^{\text{MPC}}(\mathbf{s}) := \min_{\mathbf{x}, \mathbf{u}} T(\mathbf{x}_N) + \sum_{k=0}^{N-1} \gamma^k L(\mathbf{x}_k, \mathbf{u}_k)$$
$$\text{s.t. } \mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k), \quad \mathbf{x}_0 = \mathbf{s}$$
$$\mathbf{h}(\mathbf{x}_k, \mathbf{u}_k) \leq 0$$

MPC policy $\pi^{\text{MPC}}(\mathbf{s}) = \mathbf{u}_0^*$ from

$$\min_{\mathbf{x}, \mathbf{u}} \quad T(\mathbf{x}_N) + \sum_{k=0}^{N-1} \gamma^k L(\mathbf{x}_k, \mathbf{u}_k)$$

$$\text{s.t. } \mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k), \quad \mathbf{x}_0 = \mathbf{s}$$
$$\mathbf{h}(\mathbf{x}_k, \mathbf{u}_k) \leq 0$$

- Solving MDP $\leftarrow$ building model of Q^*
- E.g. Q-learning does that from data
- MPC can model a value function...
- Can it model an action-value function?



How can MPC model an MDP?

MDP optimal policy

$$\pi^* = \arg \min_{\pi} \mathbb{E} \left[\sum_{k=0}^{\infty} \gamma^k L(\mathbf{x}_k, \mathbf{u}_k) \right]$$

is entirely described by $Q^*(s, a)$

MPC as a model of the MDP

$$V^{\text{MPC}}(\mathbf{s}) := \min_{\mathbf{x}, \mathbf{u}} T(\mathbf{x}_N) + \sum_{k=0}^{N-1} \gamma^k L(\mathbf{x}_k, \mathbf{u}_k)$$

s.t. $\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k), \quad \mathbf{x}_0 = \mathbf{s}$

$$\mathbf{h}(\mathbf{x}_k, \mathbf{u}_k) \leq 0$$

$$Q^{\text{MPC}}(\mathbf{s}, \mathbf{a}) := \min_{\mathbf{x}, \mathbf{u}} T(\mathbf{x}_N) + \sum_{k=0}^{N-1} \gamma^k L(\mathbf{x}_k, \mathbf{u}_k)$$

s.t. $\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k)$

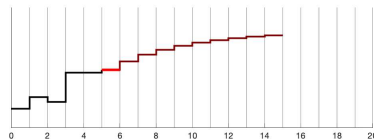
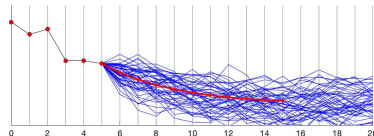
$$\mathbf{h}(\mathbf{x}_k, \mathbf{u}_k) \leq 0$$
$$\mathbf{x}_0 = \mathbf{s}, \quad \mathbf{u}_0 = \mathbf{a}$$

MPC policy $\pi^{\text{MPC}}(\mathbf{s}) = \mathbf{u}_0^*$ from

$$\min_{\mathbf{x}, \mathbf{u}} T(\mathbf{x}_N) + \sum_{k=0}^{N-1} \gamma^k L(\mathbf{x}_k, \mathbf{u}_k)$$

s.t. $\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k), \quad \mathbf{x}_0 = \mathbf{s}$

$$\mathbf{h}(\mathbf{x}_k, \mathbf{u}_k) \leq 0$$



How can MPC model an MDP?

MDP optimal policy

$$\pi^* = \arg \min_{\pi} \mathbb{E} \left[\sum_{k=0}^{\infty} \gamma^k L(\mathbf{x}_k, \mathbf{u}_k) \right]$$

is entirely described by $Q^*(s, a)$

MPC as a model of the MDP

$$V^{\text{MPC}}(\mathbf{s}) := \min_{\mathbf{x}, \mathbf{u}} T(\mathbf{x}_N) + \sum_{k=0}^{N-1} \gamma^k L(\mathbf{x}_k, \mathbf{u}_k)$$

s.t. $\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k), \quad \mathbf{x}_0 = \mathbf{s}$
 $\mathbf{h}(\mathbf{x}_k, \mathbf{u}_k) \leq 0$

$$Q^{\text{MPC}}(\mathbf{s}, \mathbf{a}) := \min_{\mathbf{x}, \mathbf{u}} T(\mathbf{x}_N) + \sum_{k=0}^{N-1} \gamma^k L(\mathbf{x}_k, \mathbf{u}_k)$$

s.t. $\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k)$
 $\mathbf{h}(\mathbf{x}_k, \mathbf{u}_k) \leq 0$
 $\mathbf{x}_0 = \mathbf{s}, \quad \mathbf{u}_0 = \mathbf{a}$

MPC policy $\pi^{\text{MPC}}(\mathbf{s}) = \mathbf{u}_0^*$ from

$$\min_{\mathbf{x}, \mathbf{u}} \quad T(\mathbf{x}_N) + \sum_{k=0}^{N-1} \gamma^k L(\mathbf{x}_k, \mathbf{u}_k)$$

s.t. $\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k), \quad \mathbf{x}_0 = \mathbf{s}$
 $\mathbf{h}(\mathbf{x}_k, \mathbf{u}_k) \leq 0$

MPC is consistent (for correct T):

$$V^{\text{MPC}}(\mathbf{s}) = \min_{\mathbf{a}} Q^{\text{MPC}}(\mathbf{s}, \mathbf{a})$$
$$\pi^{\text{MPC}}(\mathbf{s}) = \arg \min_{\mathbf{a}} Q^{\text{MPC}}(\mathbf{s}, \mathbf{a})$$

How can MPC model an MDP?

MDP optimal policy

$$\pi^* = \arg \min_{\pi} \mathbb{E} \left[\sum_{k=0}^{\infty} \gamma^k L(\mathbf{x}_k, \mathbf{u}_k) \right]$$

is entirely described by $Q^*(s, a)$

MPC as a model of the MDP

$$\begin{aligned} V^{\text{MPC}}(s) &:= \min_{\mathbf{x}, \mathbf{u}} T(\mathbf{x}_N) + \sum_{k=0}^{N-1} \gamma^k L(\mathbf{x}_k, \mathbf{u}_k) \\ \text{s.t. } \mathbf{x}_{k+1} &= \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k), \quad \mathbf{x}_0 = \mathbf{s} \\ \mathbf{h}(\mathbf{x}_k, \mathbf{u}_k) &\leq 0 \end{aligned}$$

$$\begin{aligned} Q^{\text{MPC}}(s, \mathbf{a}) &:= \min_{\mathbf{x}, \mathbf{u}} T(\mathbf{x}_N) + \sum_{k=0}^{N-1} \gamma^k L(\mathbf{x}_k, \mathbf{u}_k) \\ \text{s.t. } \mathbf{x}_{k+1} &= \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k) \\ \mathbf{h}(\mathbf{x}_k, \mathbf{u}_k) &\leq 0 \\ \mathbf{x}_0 &= s, \quad \mathbf{u}_0 = \mathbf{a} \end{aligned}$$

MPC policy $\pi^{\text{MPC}}(s) = \mathbf{u}_0^*$ from

$$\begin{aligned} \min_{\mathbf{x}, \mathbf{u}} \quad & T(\mathbf{x}_N) + \sum_{k=0}^{N-1} \gamma^k L(\mathbf{x}_k, \mathbf{u}_k) \\ \text{s.t. } \quad & \mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k), \quad \mathbf{x}_0 = s \\ & \mathbf{h}(\mathbf{x}_k, \mathbf{u}_k) \leq 0 \end{aligned}$$

MPC is consistent (for correct T):

$$\begin{aligned} V^{\text{MPC}}(s) &= \min_{\mathbf{a}} Q^{\text{MPC}}(s, \mathbf{a}) \\ \pi^{\text{MPC}}(s) &= \arg \min_{\mathbf{a}} Q^{\text{MPC}}(s, \mathbf{a}) \end{aligned}$$

MPC is a complete model of MDP if:

$$Q^{\text{MPC}}(s, \mathbf{a}) = Q^*(s, \mathbf{a})$$

for all $s, \mathbf{a}$. Then **optimality** holds:

$$\pi^{\text{MPC}}(s) = \pi^*(s)$$

Modelling Q^* or only π^* ?

MPC solves MDP if for all s, a

$$\pi^{\text{MPC}}(s) = \pi^*(s)$$

Modelling Q^* or only π^* ?

MPC solves MDP if for all s, a

$$\pi^{\text{MPC}}(s) = \pi^*(s)$$

which requires

$$\underbrace{\arg \min_a Q^{\text{MPC}}(s, a)}_{=\pi^{\text{MPC}}(s)} = \underbrace{\arg \min_a Q^*(s, a)}_{=\pi^*(s)}$$

Modelling Q^* or only π^* ?

MPC solves MDP if for all s, a

$$\pi^{\text{MPC}}(s) = \pi^*(s)$$

which requires

$$\underbrace{\arg \min_a Q^{\text{MPC}}(s, a)}_{=\pi^{\text{MPC}}(s)} = \underbrace{\arg \min_a Q^*(s, a)}_{=\pi^*(s)}$$

Hence condition

$$Q^{\text{MPC}}(s, a) = Q^*(s, a)$$

is sufficient but *not necessary* for MPC to solve the MDP. [Why do we want it then?](#)

Modelling Q^* or only π^* ?

MPC solves MDP if for all s, a

$$\pi^{\text{MPC}}(s) = \pi^*(s)$$

which requires

$$\underbrace{\arg \min_a Q^{\text{MPC}}(s, a)}_{=\pi^{\text{MPC}}(s)} = \underbrace{\arg \min_a Q^*(s, a)}_{=\pi^*(s)}$$

Because of the **arg mins**, the conditions for which

$$\arg \min_a Q^{\text{MPC}}(s, a) = \arg \min_a Q^*(s, a)$$

holds are intricate, difficult to analyze and use.

Hence condition

$$Q^{\text{MPC}}(s, a) = Q^*(s, a)$$

is sufficient but *not necessary* for MPC to solve the MDP. **Why do we want it then?**

Modelling Q^* or only π^* ?

MPC solves MDP if for all s, a

$$\pi^{\text{MPC}}(s) = \pi^*(s)$$

which requires

$$\underbrace{\arg \min_a Q^{\text{MPC}}(s, a)}_{=\pi^{\text{MPC}}(s)} = \underbrace{\arg \min_a Q^*(s, a)}_{=\pi^*(s)}$$

Hence condition

$$Q^{\text{MPC}}(s, a) = Q^*(s, a)$$

is sufficient but *not necessary* for MPC to solve the MDP. Why do we want it then?

Because of the **arg mins**, the conditions for which

$$\arg \min_a Q^{\text{MPC}}(s, a) = \arg \min_a Q^*(s, a)$$

holds are intricate, difficult to analyze and use. In contrast

$$Q^{\text{MPC}}(s, a) = Q^*(s, a)$$

is a “simple” condition, yielding simple requirements on the MPC scheme

Modelling Q^* or only π^* ?

MPC solves MDP if for all s, a

$$\pi^{\text{MPC}}(s) = \pi^*(s)$$

which requires

$$\underbrace{\arg \min_a Q^{\text{MPC}}(s, a)}_{=\pi^{\text{MPC}}(s)} = \underbrace{\arg \min_a Q^*(s, a)}_{=\pi^*(s)}$$

Hence condition

$$Q^{\text{MPC}}(s, a) = Q^*(s, a)$$

is sufficient but *not necessary* for MPC to solve the MDP. Why do we want it then?

Because of the **arg mins**, the conditions for which

$$\arg \min_a Q^{\text{MPC}}(s, a) = \arg \min_a Q^*(s, a)$$

holds are intricate, difficult to analyze and use. In contrast

$$Q^{\text{MPC}}(s, a) = Q^*(s, a)$$

is a “simple” condition, yielding simple requirements on the MPC scheme

Bottom line:

- in practice, we often target $\pi^{\text{MPC}}(s) = \pi^*(s)$
- for theoretical analysis, working with $Q^{\text{MPC}}(s, a) = Q^*(s, a)$ is much easier

Modelling the MDP solution with MPC?

How to achieve

$$Q^{\text{MPC}}(s, a) = Q^*(s, a)?$$

Modelling the MDP solution with MPC?

How to achieve

$$Q^{\text{MPC}}(s, a) = Q^*(s, a)?$$

Classic view on MPC

MPC: at current state s solve

$$\min_{\mathbf{x}, \mathbf{u}} \quad T(\mathbf{x}_N) + \sum_{k=0}^{N-1} L(\mathbf{x}_k, \mathbf{u}_k)$$

$$\text{s.t.} \quad \mathbf{x}_{k+1} = \mathbf{f}_{\theta}(\mathbf{x}_k, \mathbf{u}_k)$$

$$\mathbf{h}(\mathbf{x}_k, \mathbf{u}_k) \leq 0$$

$$\mathbf{x}_0 = s$$

gives policy $\pi_{\theta}^{\text{MPC}}(s) = \mathbf{u}_0^*$

Find θ such that MPC predictions fit the data

Modelling the MDP solution with MPC? Paradigm shifts...

How to achieve

$$Q^{\text{MPC}}(s, a) = Q^*(s, a)?$$

Shift 1: focus on performance instead of fitting

- **from:** f_θ is a model for the system dynamics
- **to:** MPC is a model of the MDP solution Q^*

Classic view on MPC

MPC: at current state s solve

$$\min_{\mathbf{x}, \mathbf{u}} \quad T(\mathbf{x}_N) + \sum_{k=0}^{N-1} L(\mathbf{x}_k, \mathbf{u}_k)$$

$$\text{s.t.} \quad \mathbf{x}_{k+1} = \mathbf{f}_\theta(\mathbf{x}_k, \mathbf{u}_k)$$

$$\mathbf{h}(\mathbf{x}_k, \mathbf{u}_k) \leq 0$$

$$\mathbf{x}_0 = s$$

gives policy $\pi_\theta^{\text{MPC}}(s) = \mathbf{u}_0^*$

Find θ such that MPC predictions fit the data

Modelling the MDP solution with MPC? Paradigm shifts...

How to achieve

$$Q^{\text{MPC}}(s, a) = Q^*(s, a)?$$

Classic view on MPC

MPC: at current state s solve

$$\min_{\mathbf{x}, \mathbf{u}} \quad T(\mathbf{x}_N) + \sum_{k=0}^{N-1} L(\mathbf{x}_k, \mathbf{u}_k)$$

$$\text{s.t.} \quad \mathbf{x}_{k+1} = \mathbf{f}_{\theta}(\mathbf{x}_k, \mathbf{u}_k)$$

$$\mathbf{h}(\mathbf{x}_k, \mathbf{u}_k) \leq 0$$

$$\mathbf{x}_0 = s$$

gives policy $\pi_{\theta}^{\text{MPC}}(s) = \mathbf{u}_0^*$

Find θ such that MPC predictions fit the data

Shift 1: focus on performance instead of fitting

- **from:** $\mathbf{f}_{\theta}$ is a model for the system dynamics
- **to:** MPC is a model of the MDP solution Q^*

Shift to...

Find θ that “fits MPC to optimality” according to the data

$$(Q^{\text{MPC}} \rightarrow Q^* \text{ or at least } \pi^{\text{MPC}} \rightarrow \pi^*)$$

Modelling the MDP solution with MPC? Paradigm shifts...

How to achieve

$$Q^{\text{MPC}}(s, a) = Q^*(s, a)?$$

Classic view on MPC

MPC: at current state s solve

$$\min_{\mathbf{x}, \mathbf{u}} \quad T(\mathbf{x}_N) + \sum_{k=0}^{N-1} L(\mathbf{x}_k, \mathbf{u}_k)$$

$$\text{s.t.} \quad \mathbf{x}_{k+1} = \mathbf{f}_{\theta}(\mathbf{x}_k, \mathbf{u}_k)$$

$$\mathbf{h}(\mathbf{x}_k, \mathbf{u}_k) \leq 0$$

$$\mathbf{x}_0 = s$$

gives policy $\pi_{\theta}^{\text{MPC}}(s) = \mathbf{u}_0^*$

Find θ such that MPC predictions fit the data

Shift 1: focus on performance instead of fitting

- **from:** $\mathbf{f}_{\theta}$ is a model for the system dynamics
- **to:** MPC is a model of the MDP solution Q^*

Shift to...

Find θ that “fits MPC to optimality” according to the data

$$(Q^{\text{MPC}} \rightarrow Q^* \text{ or at least } \pi^{\text{MPC}} \rightarrow \pi^*)$$

- $\rightarrow$ Best model for closed-loop performance
- $\neq$ Best model to fit the data!

Modelling the MDP solution with MPC? Paradigm shifts...

How to achieve

$$Q^{\text{MPC}}(s, a) = Q^*(s, a)?$$

Classic view on MPC

MPC: at current state s solve

$$\min_{\mathbf{x}, \mathbf{u}} \quad T(\mathbf{x}_N) + \sum_{k=0}^{N-1} L(\mathbf{x}_k, \mathbf{u}_k)$$

$$\text{s.t.} \quad \mathbf{x}_{k+1} = \mathbf{f}_{\theta}(\mathbf{x}_k, \mathbf{u}_k)$$

$$\mathbf{h}(\mathbf{x}_k, \mathbf{u}_k) \leq 0$$

$$\mathbf{x}_0 = s$$

gives policy $\pi_{\theta}^{\text{MPC}}(s) = \mathbf{u}_0^*$

Find θ such that MPC predictions fit the data

Shift 1: focus on performance instead of fitting

- **from:** $\mathbf{f}_{\theta}$ is a model for the system dynamics
- **to:** MPC is a model of the MDP solution Q^*

Shift to...

Find θ that “fits MPC to optimality” according to the data

$$(Q^{\text{MPC}} \rightarrow Q^* \text{ or at least } \pi^{\text{MPC}} \rightarrow \pi^*)$$

- $\rightarrow$ Best model for closed-loop performance
- $\neq$ Best model to fit the data!

But $\pi^{\text{MPC}} = \pi^*$ places “high demands” on $\mathbf{f}_{\theta}$

Can we do better?

Modelling the MDP solution with MPC? Paradigm shifts...

How to achieve

$$Q^{\text{MPC}}(s, a) = Q^*(s, a)?$$

Classic view on MPC

MPC: at current state s solve

$$\min_{\mathbf{x}, \mathbf{u}} \quad T(\mathbf{x}_N) + \sum_{k=0}^{N-1} L(\mathbf{x}_k, \mathbf{u}_k)$$

$$\text{s.t.} \quad \mathbf{x}_{k+1} = \mathbf{f}_{\theta}(\mathbf{x}_k, \mathbf{u}_k)$$

$$\mathbf{h}(\mathbf{x}_k, \mathbf{u}_k) \leq 0$$

$$\mathbf{x}_0 = s$$

gives policy $\pi_{\theta}^{\text{MPC}}(s) = \mathbf{u}_0^*$

Find θ such that MPC predictions fit the data

Shift 1: focus on performance instead of fitting

- **from:** $\mathbf{f}_{\theta}$ is a model for the system dynamics
- **to:** MPC is a model of the MDP solution Q^*

Shift to...

Find θ that “fits MPC to optimality” according to the data

$$(Q^{\text{MPC}} \rightarrow Q^* \text{ or at least } \pi^{\text{MPC}} \rightarrow \pi^*)$$

- $\rightarrow$ Best model for closed-loop performance
- $\neq$ Best model to fit the data!

But $\pi^{\text{MPC}} = \pi^*$ places “high demands” on $\mathbf{f}_{\theta}$

Can we do better? **Yes!**

Modelling the MDP solution with MPC? Paradigm shifts...

How to achieve

$$Q^{\text{MPC}}(s, a) = Q^*(s, a)?$$

Classic view on MPC

MPC: at current state s solve

$$\min_{\mathbf{x}, \mathbf{u}} \quad T(\mathbf{x}_N) + \sum_{k=0}^{N-1} L(\mathbf{x}_k, \mathbf{u}_k)$$

$$\text{s.t.} \quad \mathbf{x}_{k+1} = \mathbf{f}_{\theta}(\mathbf{x}_k, \mathbf{u}_k)$$

$$\mathbf{h}(\mathbf{x}_k, \mathbf{u}_k) \leq 0$$

$$\mathbf{x}_0 = s$$

gives policy $\pi_{\theta}^{\text{MPC}}(s) = \mathbf{u}_0^*$

Find θ such that MPC predictions fit the data

Shift 1: focus on performance instead of fitting

- **from:** $\mathbf{f}_{\theta}$ is a model for the system dynamics
- **to:** MPC is a model of the MDP solution Q^*

Shift to...

Find θ that “fits MPC to optimality” according to the data

$$(Q^{\text{MPC}} \rightarrow Q^* \text{ or at least } \pi^{\text{MPC}} \rightarrow \pi^*)$$

Shift 2: “holistic” parametrization

$$\min_{\mathbf{x}, \mathbf{u}} \quad T_{\theta}(\mathbf{x}_N) + \sum_{k=0}^{N-1} L_{\theta}(\mathbf{x}_k, \mathbf{u}_k)$$

$$\text{s.t.} \quad \mathbf{x}_{k+1} = \mathbf{f}_{\theta}(\mathbf{x}_k, \mathbf{u}_k), \quad \mathbf{x}_0 = s$$

$$\mathbf{h}_{\theta}(\mathbf{x}_k, \mathbf{u}_k) \leq 0$$

i.e. cost and constraints are part of the Q^* model

Modelling the MDP solution with MPC? Paradigm shifts...

How to achieve

$$Q^{\text{MPC}}(s, a) = Q^*(s, a)?$$

Classic view on MPC

MPC: at current state s solve

$$\min_{\mathbf{x}, \mathbf{u}} \quad T(\mathbf{x}_N) + \sum_{k=0}^{N-1} L(\mathbf{x}_k, \mathbf{u}_k)$$

$$\text{s.t.} \quad \mathbf{x}_{k+1} = \mathbf{f}_{\theta}(\mathbf{x}_k, \mathbf{u}_k)$$

$$\mathbf{h}(\mathbf{x}_k, \mathbf{u}_k) \leq 0$$

$$\mathbf{x}_0 = s$$

gives policy $\pi_{\theta}^{\text{MPC}}(s) = \mathbf{u}_0^*$

Find θ such that MPC predictions fit the data

Shift 1: focus on performance instead of fitting

- **from:** $\mathbf{f}_{\theta}$ is a model for the system dynamics
- **to:** MPC is a model of the MDP solution Q^*

Shift to...

Find θ that “fits MPC to optimality” according to the data

$$(Q^{\text{MPC}} \rightarrow Q^* \text{ or at least } \pi^{\text{MPC}} \rightarrow \pi^*)$$

Shift 2: “holistic” parametrization

$$\min_{\mathbf{x}, \mathbf{u}} \quad T_{\theta}(\mathbf{x}_N) + \sum_{k=0}^{N-1} L_{\theta}(\mathbf{x}_k, \mathbf{u}_k)$$

$$\text{s.t.} \quad \mathbf{x}_{k+1} = \mathbf{f}_{\theta}(\mathbf{x}_k, \mathbf{u}_k), \quad \mathbf{x}_0 = s$$

$$\mathbf{h}_{\theta}(\mathbf{x}_k, \mathbf{u}_k) \leq 0$$

i.e. cost and constraints are part of the Q^* model

Intuitively ok... formal justification?

“Holistic” parametrization - Is that formally justified? Yes...

Full MPC parametrization:

$$\min_{\mathbf{x}, \mathbf{u}} \quad T_{\theta}(\mathbf{x}_N) + \sum_{k=0}^{N-1} L_{\theta}(\mathbf{x}_k, \mathbf{u}_k)$$

$$\text{s.t.} \quad \mathbf{x}_{k+1} = \mathbf{f}_{\theta}(\mathbf{x}_k, \mathbf{u}_k)$$

$$\mathbf{h}_{\theta}(\mathbf{x}_k, \mathbf{u}_k) \leq 0$$

$$\mathbf{x}_0 = \mathbf{s}$$

gives policy $\pi_{\theta}^{\text{MPC}}(\mathbf{s}) = \mathbf{u}_0^*$

“Holistic” parametrization - Is that formally justified? Yes...

Full MPC parametrization:

$$\min_{\mathbf{x}, \mathbf{u}} \quad T_{\theta}(\mathbf{x}_N) + \sum_{k=0}^{N-1} L_{\theta}(\mathbf{x}_k, \mathbf{u}_k)$$

$$\text{s.t.} \quad \mathbf{x}_{k+1} = \mathbf{f}_{\theta}(\mathbf{x}_k, \mathbf{u}_k)$$

$$\mathbf{h}_{\theta}(\mathbf{x}_k, \mathbf{u}_k) \leq 0$$

$$\mathbf{x}_0 = \mathbf{s}$$

gives policy $\pi_{\theta}^{\text{MPC}}(\mathbf{s}) = \mathbf{u}_0^*$

Theorem: under some technical conditions and for a “rich” parametrization of the MPC, there is a θ such that

$$Q_{\theta}^{\text{MPC}}(\mathbf{s}, \mathbf{a}) = Q^*(\mathbf{s}, \mathbf{a})$$

$$\pi_{\theta}^{\text{MPC}}(\mathbf{s}) = \pi^*(\mathbf{s})$$

even if the model $\mathbf{f}_{\theta}$ *cannot* describe the real system accurately

“Holistic” parametrization - Is that formally justified? Yes...

Full MPC parametrization:

$$\min_{\mathbf{x}, \mathbf{u}} \quad T_{\theta}(\mathbf{x}_N) + \sum_{k=0}^{N-1} L_{\theta}(\mathbf{x}_k, \mathbf{u}_k)$$

$$\text{s.t.} \quad \mathbf{x}_{k+1} = \mathbf{f}_{\theta}(\mathbf{x}_k, \mathbf{u}_k)$$

$$\mathbf{h}_{\theta}(\mathbf{x}_k, \mathbf{u}_k) \leq 0$$

$$\mathbf{x}_0 = \mathbf{s}$$

gives policy $\pi_{\theta}^{\text{MPC}}(\mathbf{s}) = \mathbf{u}_0^*$

Theorem: under some technical conditions and for a “rich” parametrization of the MPC, there is a θ such that

$$Q_{\theta}^{\text{MPC}}(\mathbf{s}, \mathbf{a}) = Q^*(\mathbf{s}, \mathbf{a})$$

$$\pi_{\theta}^{\text{MPC}}(\mathbf{s}) = \pi^*(\mathbf{s})$$

even if the model $\mathbf{f}_{\theta}$ *cannot* describe the real system accurately

Remarks

- **Compensate** MPC model deficiencies in the cost + constraints

“Holistic” parametrization - Is that formally justified? Yes...

Full MPC parametrization:

$$\min_{\mathbf{x}, \mathbf{u}} \quad T_{\theta}(\mathbf{x}_N) + \sum_{k=0}^{N-1} L_{\theta}(\mathbf{x}_k, \mathbf{u}_k)$$

$$\text{s.t.} \quad \mathbf{x}_{k+1} = \mathbf{f}_{\theta}(\mathbf{x}_k, \mathbf{u}_k)$$

$$\mathbf{h}_{\theta}(\mathbf{x}_k, \mathbf{u}_k) \leq 0$$

$$\mathbf{x}_0 = \mathbf{s}$$

gives policy $\pi_{\theta}^{\text{MPC}}(\mathbf{s}) = \mathbf{u}_0^*$

Theorem: under some technical conditions and for a “rich” parametrization of the MPC, there is a θ such that

$$Q_{\theta}^{\text{MPC}}(\mathbf{s}, \mathbf{a}) = Q^*(\mathbf{s}, \mathbf{a})$$

$$\pi_{\theta}^{\text{MPC}}(\mathbf{s}) = \pi^*(\mathbf{s})$$

even if the model $\mathbf{f}_{\theta}$ *cannot* describe the real system accurately

Remarks

- **Compensate** MPC model deficiencies in the cost + constraints
- **Generic:** works for robust MPC, stochastic MPC, economic MPC, Multi-Step PC, etc.

“Holistic” parametrization - Is that formally justified? Yes...

Full MPC parametrization:

$$\min_{\mathbf{x}, \mathbf{u}} \quad T_{\theta}(\mathbf{x}_N) + \sum_{k=0}^{N-1} L_{\theta}(\mathbf{x}_k, \mathbf{u}_k)$$

$$\text{s.t.} \quad \mathbf{x}_{k+1} = \mathbf{f}_{\theta}(\mathbf{x}_k, \mathbf{u}_k)$$

$$\mathbf{h}_{\theta}(\mathbf{x}_k, \mathbf{u}_k) \leq 0$$

$$\mathbf{x}_0 = \mathbf{s}$$

gives policy $\pi_{\theta}^{\text{MPC}}(\mathbf{s}) = \mathbf{u}_0^*$

Theorem: under some technical conditions and for a “rich” parametrization of the MPC, there is a θ such that

$$Q_{\theta}^{\text{MPC}}(\mathbf{s}, \mathbf{a}) = Q^*(\mathbf{s}, \mathbf{a})$$

$$\pi_{\theta}^{\text{MPC}}(\mathbf{s}) = \pi^*(\mathbf{s})$$

even if the model $\mathbf{f}_{\theta}$ *cannot* describe the real system accurately

Remarks

- **Compensate** MPC model deficiencies in the cost + constraints
- **Generic:** works for robust MPC, stochastic MPC, economic MPC, Multi-Step PC, etc.
- **Sanity check:** technical conditions are mild but forbid MPC model to be “very absurd”

“Holistic” parametrization - Is that formally justified? Yes...

Full MPC parametrization:

$$\min_{\mathbf{x}, \mathbf{u}} \quad T_{\theta}(\mathbf{x}_N) + \sum_{k=0}^{N-1} L_{\theta}(\mathbf{x}_k, \mathbf{u}_k)$$

$$\text{s.t.} \quad \mathbf{x}_{k+1} = \mathbf{f}_{\theta}(\mathbf{x}_k, \mathbf{u}_k)$$

$$\mathbf{h}_{\theta}(\mathbf{x}_k, \mathbf{u}_k) \leq 0$$

$$\mathbf{x}_0 = \mathbf{s}$$

gives policy $\pi_{\theta}^{\text{MPC}}(\mathbf{s}) = \mathbf{u}_0^*$

MPC is a model of the optimal policy

not a policy approximation using open-loop (model-based) predictions

Theorem: under some technical conditions and for a “rich” parametrization of the MPC, there is a θ such that

$$Q_{\theta}^{\text{MPC}}(\mathbf{s}, \mathbf{a}) = Q^*(\mathbf{s}, \mathbf{a})$$

$$\pi_{\theta}^{\text{MPC}}(\mathbf{s}) = \pi^*(\mathbf{s})$$

even if the model $\mathbf{f}_{\theta}$ *cannot* describe the real system accurately

Remarks

- **Compensate** MPC model deficiencies in the cost + constraints
- **Generic:** works for robust MPC, stochastic MPC, economic MPC, Multi-Step PC, etc.
- **Sanity check:** technical conditions are mild but forbid MPC model to be “very absurd”

How to use this? Reinforcement Learning

Policy $\pi_{\theta}^{\text{MPC}}(\mathbf{s}) = \mathbf{u}_0^*$ from

$$\min_{\mathbf{x}, \mathbf{u}} \quad T_{\theta}(\mathbf{x}_N) + \sum_{k=0}^{N-1} L_{\theta}(\mathbf{x}_k, \mathbf{u}_k)$$

$$\text{s.t.} \quad \mathbf{x}_{k+1} = \mathbf{f}_{\theta}(\mathbf{x}_k, \mathbf{u}_k)$$

$$\mathbf{h}_{\theta}(\mathbf{x}_k, \mathbf{u}_k) \leq 0, \quad \mathbf{x}_0 = \mathbf{s}$$

- $\min_{\theta} J(\pi_{\theta}^{\text{MPC}})$ using data
- $\theta \rightarrow J(\pi_{\theta}^{\text{MPC}})$ very implicit
- $J(\cdot)$ is the real-system!

How to use this? Reinforcement Learning

Reinforcement Learning

Tools to approximate π^* from data
This is not (necessarily) about DNNs

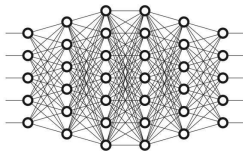
Policy $\pi_{\theta}^{\text{MPC}}(\mathbf{s}) = \mathbf{u}_0^*$ from

$$\min_{\mathbf{x}, \mathbf{u}} \quad T_{\theta}(\mathbf{x}_N) + \sum_{k=0}^{N-1} L_{\theta}(\mathbf{x}_k, \mathbf{u}_k)$$

$$\text{s.t.} \quad \mathbf{x}_{k+1} = \mathbf{f}_{\theta}(\mathbf{x}_k, \mathbf{u}_k)$$

$$\mathbf{h}_{\theta}(\mathbf{x}_k, \mathbf{u}_k) \leq 0, \quad \mathbf{x}_0 = \mathbf{s}$$

- $\min_{\theta} J(\pi_{\theta}^{\text{MPC}})$ using data
- $\theta \rightarrow J(\pi_{\theta}^{\text{MPC}})$ very implicit
- $J(\cdot)$ is the real-system!



How to use this? Reinforcement Learning

Policy $\pi_{\theta}^{\text{MPC}}(\mathbf{s}) = \mathbf{u}_0^*$ from

$$\min_{\mathbf{x}, \mathbf{u}} \quad T_{\theta}(\mathbf{x}_N) + \sum_{k=0}^{N-1} L_{\theta}(\mathbf{x}_k, \mathbf{u}_k)$$

$$\text{s.t.} \quad \mathbf{x}_{k+1} = \mathbf{f}_{\theta}(\mathbf{x}_k, \mathbf{u}_k)$$

$$\mathbf{h}_{\theta}(\mathbf{x}_k, \mathbf{u}_k) \leq 0, \quad \mathbf{x}_0 = \mathbf{s}$$

- $\min_{\theta} J(\pi_{\theta}^{\text{MPC}})$ using data
- $\theta \rightarrow J(\pi_{\theta}^{\text{MPC}})$ very implicit
- $J(\cdot)$ is the real-system!

Reinforcement Learning

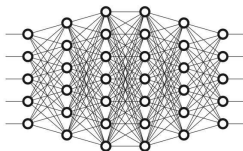
Tools to approximate π^* from data
This is not (necessarily) about DNNs

For MPC: tools to find best θ , e.g.

- **Policy Gradient:** estimations of

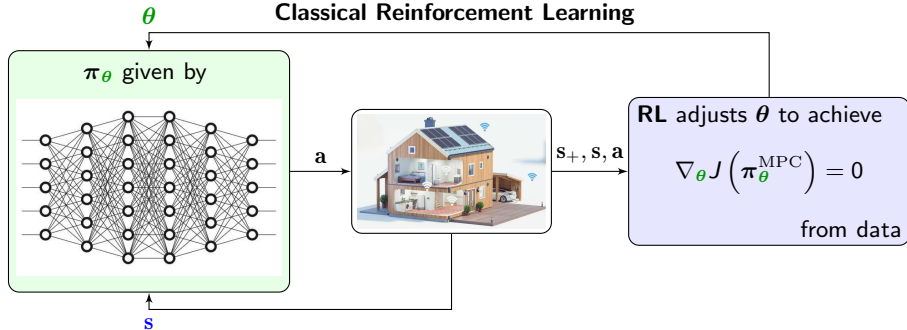
$$\nabla_{\theta} J(\pi_{\theta}^{\text{MPC}}), \quad \text{possibly} \quad \nabla_{\theta}^2 J(\pi_{\theta}^{\text{MPC}})$$

- **Q-learning:** direct “shaping” of MPC



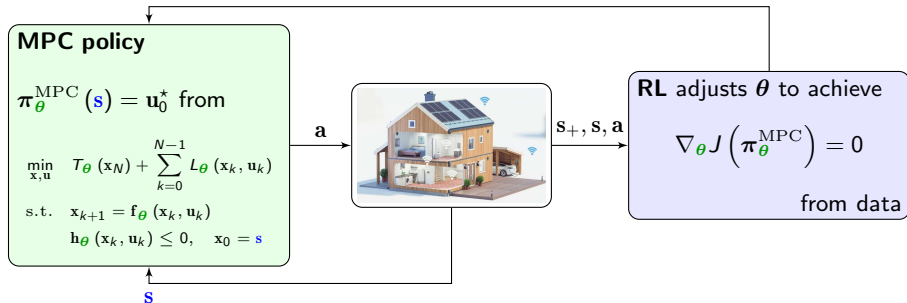
Reinforcement Learning & MPC

Classical Reinforcement Learning



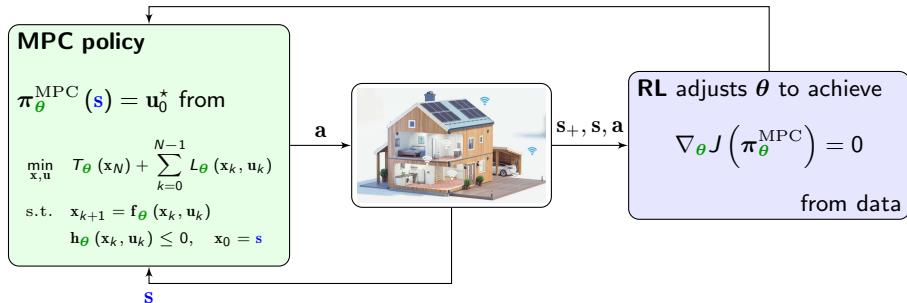
Reinforcement Learning & MPC

Reinforcement Learning over MPC



Reinforcement Learning & MPC

Reinforcement Learning over MPC

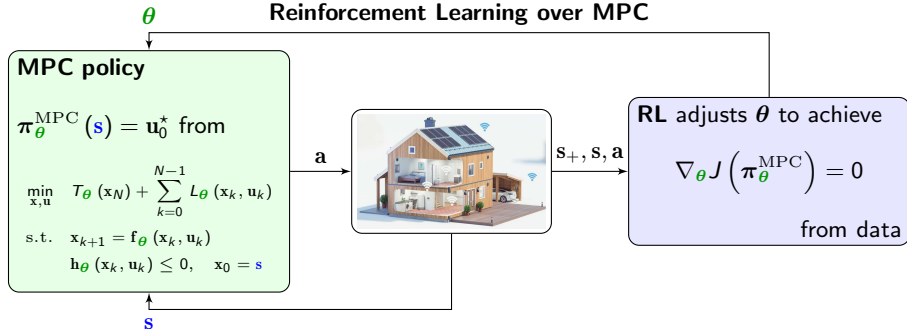


Why is it useful?

- ✓ Tunes your optimization model for real-world performance
- ✓ MPC: 25 years of results on formal guarantees & methods
- ✓ Easy to inject knowledge
- ✓ Learning does not start from scratch
- ✓ Planning provides explainability
- ✓ Software now available

Reinforcement Learning & MPC

Reinforcement Learning over MPC



Why is it useful?

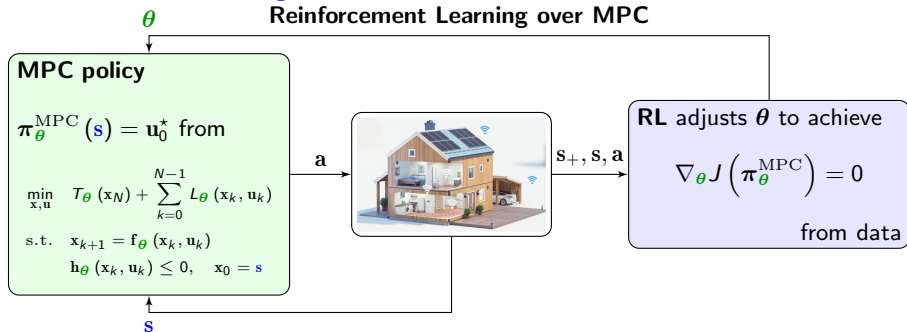
- ✓ Tunes your optimization model for real-world performance
- ✓ MPC: 25 years of results on formal guarantees & methods
- ✓ Easy to inject knowledge
- ✓ Learning does not start from scratch
- ✓ Planning provides explainability
- ✓ Software now available

Why can it be difficult?

- ✗ Optimization is computationally more expensive than a DNN
- ✗ Deployment on GPUs is lagging
- ✗ Non-convexity can get in the way...

Reinforcement Learning & MPC

Reinforcement Learning over MPC



Why is it useful?

- ✓ Tunes your optimization model for real-world performance
- ✓ MPC: 25 years of results on formal guarantees & methods
- ✓ Easy to inject knowledge
- ✓ Learning does not start from scratch
- ✓ Planning provides explainability
- ✓ Software now available

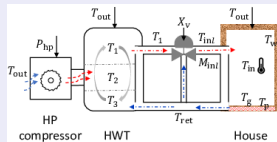
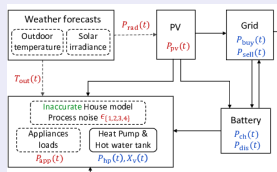
Why can it be difficult?

- ✗ Optimization is computationally more expensive than a DNN
- ✗ Deployment on GPUs is lagging
- ✗ Non-convexity can get in the way...

RL over MPC can tune your MPC for performance *beyond* what classical methods can do!

Example - Home Energy Management (simulated)

Setup



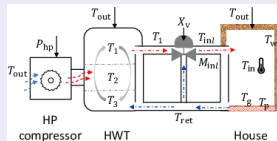
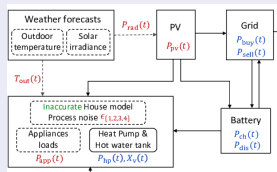
Example - Home Energy Management (simulated)

Setup



Learning process: aligned followed with closed-loop

Real world
data



Example - Home Energy Management (simulated)

Setup

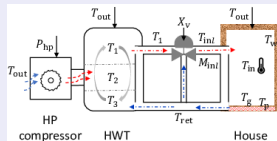
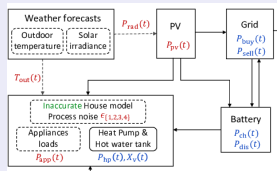


Learning process: aligned followed with closed-loop

Real world
data

Data

Classical
model fitting
to build
MPC model

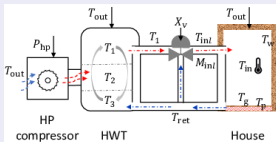
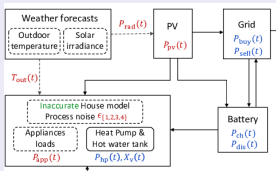
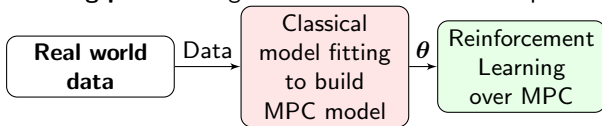


Example - Home Energy Management (simulated)

Setup

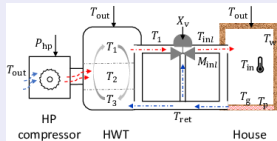
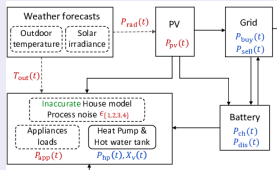


Learning process: aligned followed with closed-loop

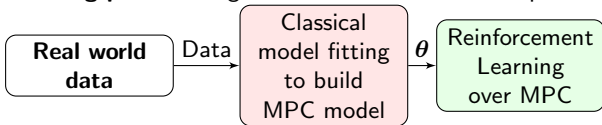


Example - Home Energy Management (simulated)

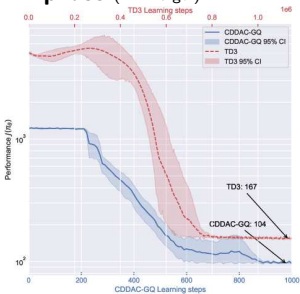
Setup



Learning process: aligned followed with closed-loop

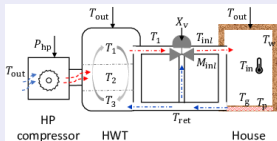
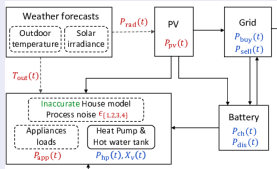


RL phase (2 RL algo.)

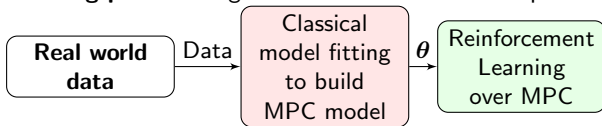


Example - Home Energy Management (simulated)

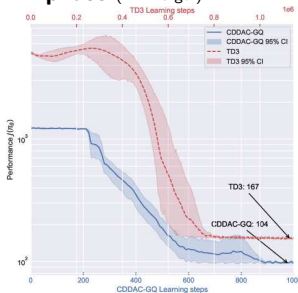
Setup



Learning process: aligned followed with closed-loop



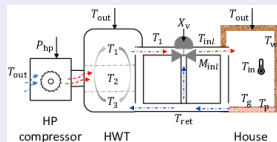
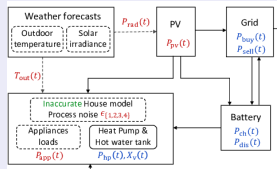
RL phase (2 RL algo.)



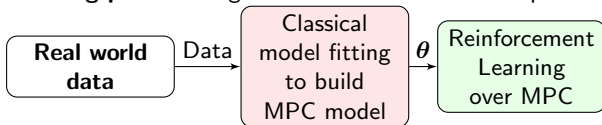
The RL step improves the MPC performance significantly from only model fitting

Example - Home Energy Management (simulated)

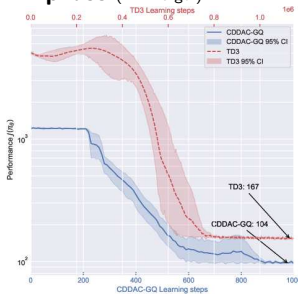
Setup



Learning process: aligned followed with closed-loop



RL phase (2 RL algo.)



The RL step improves the MPC performance significantly from only model fitting

RL over MPC can tune your MPC for performance **beyond** what classical methods can do!

Outline

- 1 Let's rebuild some background – MPC and MDPs
- 2 MPC as a solution to MDPs
- 3 When is RL (most) beneficial for MPC?
- 4 A Deeper Look at the Theory

Model fitting?

Proposed paradigm

Policy $\pi_{\theta}^{\text{MPC}}(\mathbf{s}) = \mathbf{u}_0^*$ from

$$\min_{\mathbf{x}, \mathbf{u}} \quad T_{\theta}(\mathbf{x}_N) + \sum_{k=0}^{N-1} L_{\theta}(\mathbf{x}_k, \mathbf{u}_k)$$

$$\text{s.t.} \quad \mathbf{x}_{k+1} = \mathbf{f}_{\theta}(\mathbf{x}_k, \mathbf{u}_k)$$

$$\mathbf{h}_{\theta}(\mathbf{x}_k, \mathbf{u}_k) \leq 0, \quad \mathbf{x}_0 = \mathbf{s}$$

Model fitting?

A step back: **model adjustment?**

Policy $\pi_{\theta}^{\text{MPC}}(\mathbf{s}) = \mathbf{u}_0^*$ from

$$\begin{aligned} \min_{\mathbf{x}, \mathbf{u}} \quad & T(\mathbf{x}_N) + \sum_{k=0}^{N-1} L(\mathbf{x}_k, \mathbf{u}_k) \\ \text{s.t.} \quad & \mathbf{x}_{k+1} = \mathbf{f}_{\theta}(\mathbf{x}_k, \mathbf{u}_k) \\ & \mathbf{h}(\mathbf{x}_k, \mathbf{u}_k) \leq 0, \quad \mathbf{x}_0 = \mathbf{s} \end{aligned}$$

Adjust θ according to (e.g.)

1. $\min_{\theta} \mathbb{E} \left[\|\mathbf{f}_{\theta}(\mathbf{s}, \mathbf{a}) - \mathbf{s}_+\|^2 \right]$ vs.
2. $\min_{\theta} J(\pi_{\theta}^{\text{MPC}})$

... is that different?

Model fitting?

A step back: **model adjustment**?

Policy $\pi_{\theta}^{\text{MPC}}(\mathbf{s}) = \mathbf{u}_0^*$ from

$$\min_{\mathbf{x}, \mathbf{u}} \quad T(\mathbf{x}_N) + \sum_{k=0}^{N-1} L(\mathbf{x}_k, \mathbf{u}_k)$$

$$\text{s.t.} \quad \mathbf{x}_{k+1} = \mathbf{f}_{\theta}(\mathbf{x}_k, \mathbf{u}_k) \\ \mathbf{h}(\mathbf{x}_k, \mathbf{u}_k) \leq 0, \quad \mathbf{x}_0 = \mathbf{s}$$

Adjust θ according to (e.g.)

1. $\min_{\theta} \mathbb{E} \left[\|\mathbf{f}_{\theta}(\mathbf{s}, \mathbf{a}) - \mathbf{s}_+\|^2 \right]$ vs.
2. $\min_{\theta} J(\pi_{\theta}^{\text{MPC}})$

... is that different?

Empirical answers:

- In general it is different...
- Good to do 1. first, then 2.
- Is step 2 necessary?
 - ▶ Improves closed-loop performance
 - ▶ But gain is very case dependent (also with full parametrization)

What causes that?

Sufficient Condition for MPC Optimality (for MDPs)

MPC policy $\pi^{\text{MPC}}(\mathbf{s}) = \mathbf{u}_0^*$ from

$$\begin{aligned} \min_{\mathbf{u}, \mathbf{x}} \quad & T(\mathbf{x}_N) + \sum_{k=0}^{N-1} L(\mathbf{x}_k, \mathbf{u}_k) \\ \text{s.t.} \quad & \mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k) \\ & \mathbf{h}(\mathbf{x}_k, \mathbf{u}_k) \leq 0, \quad \mathbf{x}_0 = \mathbf{s} \end{aligned}$$

Sufficient Condition for MPC Optimality (for MDPs)

MPC policy $\pi^{\text{MPC}}(\mathbf{s}) = \mathbf{u}_0^*$ from

$$\begin{aligned} \min_{\mathbf{u}, \mathbf{x}} \quad & T(\mathbf{x}_N) + \sum_{k=0}^{N-1} L(\mathbf{x}_k, \mathbf{u}_k) \\ \text{s.t.} \quad & \mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k) \\ & \mathbf{h}(\mathbf{x}_k, \mathbf{u}_k) \leq 0, \quad \mathbf{x}_0 = \mathbf{s} \end{aligned}$$

Sufficient condition of optimality of MPC
relates model $\mathbf{f}$ to optimal value function and
conditional distribution $\mathbf{s}, \mathbf{a} \rightarrow \mathbf{s}_+$ of the MDP
(+ condition on T)

Sufficient Condition for MPC Optimality (for MDPs)

MPC policy $\pi^{\text{MPC}}(s) = \mathbf{u}_0^*$ from

$$\min_{\mathbf{u}, \mathbf{x}} \quad T(\mathbf{x}_N) + \sum_{k=0}^{N-1} L(\mathbf{x}_k, \mathbf{u}_k)$$
$$\text{s.t.} \quad \mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k)$$
$$\mathbf{h}(\mathbf{x}_k, \mathbf{u}_k) \leq 0, \quad \mathbf{x}_0 = s$$

Sufficient condition of optimality of MPC
relates model $\mathbf{f}$ to optimal value function and
conditional distribution $s, \mathbf{a} \rightarrow s_+$ of the MDP
(+ condition on T)

In general

- ridge regression

$$\mathbf{f}(s, \mathbf{a}) = \mathbb{E}[s_+ | s, \mathbf{a}]$$

- max likelihood

$$\mathbf{f}(s, \mathbf{a}) = \max \rho[s_+ | s, \mathbf{a}]$$

do not satisfy the optimality
conditions

Sufficient Condition for MPC Optimality (for MDPs)

MPC policy $\pi^{\text{MPC}}(s) = u_0^*$ from

$$\min_{\mathbf{u}, \mathbf{x}} \quad T(\mathbf{x}_N) + \sum_{k=0}^{N-1} L(\mathbf{x}_k, \mathbf{u}_k)$$
$$\text{s.t.} \quad \mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k)$$
$$\mathbf{h}(\mathbf{x}_k, \mathbf{u}_k) \leq 0, \quad \mathbf{x}_0 = s$$

Sufficient condition of optimality of MPC
relates model $\mathbf{f}$ to optimal value function and
conditional distribution $s, \mathbf{a} \rightarrow s_+$ of the MDP
(+ condition on T)

In general

- ridge regression

$$\mathbf{f}(s, \mathbf{a}) = \mathbb{E}[s_+ | s, \mathbf{a}]$$

- max likelihood

$$\mathbf{f}(s, \mathbf{a}) = \max \rho[s_+ | s, \mathbf{a}]$$

do not satisfy the optimality
conditions

Model from “classic SYSID” does not necessarily
yield the best MPC policy

Sufficient Condition for MPC Optimality (for MDPs)

MPC policy $\pi^{\text{MPC}}(s) = \mathbf{u}_0^*$ from

$$\min_{\mathbf{u}, \mathbf{x}} \quad T(\mathbf{x}_N) + \sum_{k=0}^{N-1} L(\mathbf{x}_k, \mathbf{u}_k)$$
$$\text{s.t.} \quad \mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k)$$
$$\mathbf{h}(\mathbf{x}_k, \mathbf{u}_k) \leq 0, \quad \mathbf{x}_0 = s$$

Sufficient condition of optimality of MPC relates model $\mathbf{f}$ to optimal value function and conditional distribution $s, \mathbf{a} \rightarrow s_+$ of the MDP (+ condition on T)

In general

- ridge regression

$$\mathbf{f}(s, \mathbf{a}) = \mathbb{E}[s_+ | s, \mathbf{a}]$$

- max likelihood

$$\mathbf{f}(s, \mathbf{a}) = \max \rho[s_+ | s, \mathbf{a}]$$

do not satisfy the optimality conditions

Model from “classic SYSID” does not necessarily yield the best MPC policy

Notable exceptions: model gives $\mathbb{E}[s_+ | s, \mathbf{a}]$ and

- LQR with process noise
- By extension, locally optimal policies for
 - ▶ dissipative MDP and
 - ▶ $V^* \approx$ quadratic in positive invariant set

Smooth tracking MPC, fixed reference away from constraints, “reasonable” stochasticity

Sufficient Condition for MPC Optimality (for MDPs)

MPC policy $\pi^{\text{MPC}}(\mathbf{s}) = \mathbf{u}_0^*$ from

$$\begin{aligned} \min_{\mathbf{u}, \mathbf{x}} \quad & T(\mathbf{x}_N) + \sum_{k=0}^{N-1} L(\mathbf{x}_k, \mathbf{u}_k) \\ \text{s.t.} \quad & \mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k) \\ & \mathbf{h}(\mathbf{x}_k, \mathbf{u}_k) \leq 0, \quad \mathbf{x}_0 = \mathbf{s} \end{aligned}$$

Sufficient condition of optimality of MPC
relates model $\mathbf{f}$ to optimal value function and
conditional distribution $\mathbf{s}, \mathbf{a} \rightarrow \mathbf{s}_+$ of the MDP
(+ condition on T)

When is RL (most) useful for MPC? *“practical” view*

- Not much if smooth tracking problem, spend most of the time close to fixed steady state, rare transients away from the constraints $\rightarrow$ small performance gain

Sufficient Condition for MPC Optimality (for MDPs)

MPC policy $\pi^{\text{MPC}}(\mathbf{s}) = \mathbf{u}_0^*$ from

$$\min_{\mathbf{u}, \mathbf{x}} \quad T(\mathbf{x}_N) + \sum_{k=0}^{N-1} L(\mathbf{x}_k, \mathbf{u}_k)$$
$$\text{s.t.} \quad \mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k)$$
$$\mathbf{h}(\mathbf{x}_k, \mathbf{u}_k) \leq 0, \quad \mathbf{x}_0 = \mathbf{s}$$

Sufficient condition of optimality of MPC relates model $\mathbf{f}$ to optimal value function and conditional distribution $\mathbf{s}, \mathbf{a} \rightarrow \mathbf{s}_+$ of the MDP (+ condition on T)

When is RL (most) useful for MPC? “practical” view

- Not much if smooth tracking problem, spend most of the time close to fixed steady state, rare transients away from the constraints $\rightarrow$ small performance gain
- Model is not rich enough to predict the (relevant) expected state transition
- Economic problem with “low dissipativity” (trajectories spread over the state space)
- Value function curvature changes a lot, problem is non-smooth
- Optimal steady-state near / at constraints
- Varying exogeneous inputs (e.g. varying prices in energy-related problems, changing reference)
- Task-based problems: racing, minimum time, termination conditions, etc.

Outline

- 1 Let's rebuild some background – MPC and MDPs
- 2 MPC as a solution to MDPs
- 3 When is RL (most) beneficial for MPC?
- 4 A Deeper Look at the Theory

Let's discuss the assumptions

Theorem: under some **technical conditions** and for a “**rich**” **parametrization** of the MPC, there is a θ such that

$$Q_{\theta}^{\text{MPC}}(s, a) = Q^*(s, a)$$
$$\pi_{\theta}^{\text{MPC}}(s) = \pi^*(s)$$

Full MPC parametrization:

$$\begin{aligned} \min_{\mathbf{x}, \mathbf{u}} \quad & T_{\theta}(\mathbf{x}_N) + \sum_{k=0}^{N-1} L_{\theta}(\mathbf{x}_k, \mathbf{u}_k) \\ \text{s.t.} \quad & \mathbf{x}_{k+1} = \mathbf{f}_{\theta}(\mathbf{x}_k, \mathbf{u}_k) \\ & \mathbf{h}_{\theta}(\mathbf{x}_k, \mathbf{u}_k) \leq 0 \\ & \mathbf{x}_0 = \mathbf{s} \end{aligned}$$

Let's discuss the assumptions

Theorem: under some **technical conditions** and for a “**rich**” **parametrization** of the MPC, there is a θ such that

$$Q_{\theta}^{\text{MPC}}(s, a) = Q^*(s, a)$$
$$\pi_{\theta}^{\text{MPC}}(s) = \pi^*(s)$$

Full MPC parametrization:

$$\begin{aligned} \min_{\mathbf{x}, \mathbf{u}} \quad & T_{\theta}(\mathbf{x}_N) + \sum_{k=0}^{N-1} L_{\theta}(\mathbf{x}_k, \mathbf{u}_k) \\ \text{s.t.} \quad & \mathbf{x}_{k+1} = \mathbf{f}_{\theta}(\mathbf{x}_k, \mathbf{u}_k) \\ & \mathbf{h}_{\theta}(\mathbf{x}_k, \mathbf{u}_k) \leq 0 \\ & \mathbf{x}_0 = \mathbf{s} \end{aligned}$$

Rich MPC parametrization:

- this is classical in Machine Learning
- in practice limited parametrizations often do a good job
- theorem applies “asymptotically” for richer and richer parametrizations
- quantifying the performance loss for limited parametrization is result “for grab”

Let's discuss the assumptions

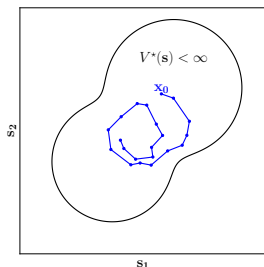
Theorem: under some **technical conditions** and for a “**rich**” **parametrization** of the MPC, there is a θ such that

$$Q_{\theta}^{\text{MPC}}(s, a) = Q^*(s, a)$$
$$\pi_{\theta}^{\text{MPC}}(s) = \pi^*(s)$$

Full MPC parametrization:

$$\min_{\mathbf{x}, \mathbf{u}} \quad T_{\theta}(\mathbf{x}_N) + \sum_{k=0}^{N-1} L_{\theta}(\mathbf{x}_k, \mathbf{u}_k)$$
$$\text{s.t.} \quad \mathbf{x}_{k+1} = \mathbf{f}_{\theta}(\mathbf{x}_k, \mathbf{u}_k)$$
$$\mathbf{h}_{\theta}(\mathbf{x}_k, \mathbf{u}_k) \leq 0$$
$$\mathbf{x}_0 = \mathbf{s}$$

Technical assumptions (*simplified*) MPC model $\mathbf{x}_{k+1} = \mathbf{f}_{\theta}(\mathbf{x}_k, \mathbf{u}_k)$ ought to be such that for $\mathbf{u}_k = \pi^*(\mathbf{x}_k)$ then $V^*(\mathbf{x}_k)$ is bounded for all k and $\mathbf{x}_0$ such that $V^*(\mathbf{x}_0) < \infty$



Let's discuss the assumptions

Theorem: under some **technical conditions** and for a “**rich**” **parametrization** of the MPC, there is a θ such that

$$Q_{\theta}^{\text{MPC}}(s, a) = Q^*(s, a) \\ \pi_{\theta}^{\text{MPC}}(s) = \pi^*(s)$$

Full MPC parametrization:

$$\begin{aligned} \min_{\mathbf{x}, \mathbf{u}} \quad & T_{\theta}(\mathbf{x}_N) + \sum_{k=0}^{N-1} L_{\theta}(\mathbf{x}_k, \mathbf{u}_k) \\ \text{s.t.} \quad & \mathbf{x}_{k+1} = \mathbf{f}_{\theta}(\mathbf{x}_k, \mathbf{u}_k) \\ & \mathbf{h}_{\theta}(\mathbf{x}_k, \mathbf{u}_k) \leq 0 \\ & \mathbf{x}_0 = \mathbf{s} \end{aligned}$$

Technical assumptions (simplified) MPC model $\mathbf{x}_{k+1} = \mathbf{f}_{\theta}(\mathbf{x}_k, \mathbf{u}_k)$ ought to be such that for $\mathbf{u}_k = \pi^*(\mathbf{x}_k)$ then $V^*(\mathbf{x}_k)$ is bounded for all k and $\mathbf{x}_0$ such that $V^*(\mathbf{x}_0) < \infty$

- MPC model remains feasible & “does not escape” under the optimal policy and admissible initial conditions
- If this works only for a subset of the set of admissible initial conditions, then MPC solves the MDP on that subset

Let's discuss the assumptions

Theorem: under some **technical conditions** and for a “**rich**” **parametrization** of the MPC, there is a θ such that

$$Q_{\theta}^{\text{MPC}}(s, a) = Q^*(s, a) \\ \pi_{\theta}^{\text{MPC}}(s) = \pi^*(s)$$

Full MPC parametrization:

$$\begin{aligned} \min_{\mathbf{x}, \mathbf{u}} \quad & T_{\theta}(\mathbf{x}_N) + \sum_{k=0}^{N-1} L_{\theta}(\mathbf{x}_k, \mathbf{u}_k) \\ \text{s.t.} \quad & \mathbf{x}_{k+1} = \mathbf{f}_{\theta}(\mathbf{x}_k, \mathbf{u}_k) \\ & \mathbf{h}_{\theta}(\mathbf{x}_k, \mathbf{u}_k) \leq 0 \\ & \mathbf{x}_0 = \mathbf{s} \end{aligned}$$

Technical assumptions (simplified) MPC model $\mathbf{x}_{k+1} = \mathbf{f}_{\theta}(\mathbf{x}_k, \mathbf{u}_k)$ ought to be such that for $\mathbf{u}_k = \pi^*(\mathbf{x}_k)$ then $V^*(\mathbf{x}_k)$ is bounded for all k and $\mathbf{x}_0$ such that $V^*(\mathbf{x}_0) < \infty$

- MPC model remains feasible & “does not escape” under the optimal policy and admissible initial conditions
- If this works only for a subset of the set of admissible initial conditions, then MPC solves the MDP on that subset
- Condition is unverifiable because it requires V^* and π^*

Let's discuss the assumptions

Theorem: under some **technical conditions** and for a “**rich**” **parametrization** of the MPC, there is a θ such that

$$Q_{\theta}^{\text{MPC}}(s, a) = Q^*(s, a)$$
$$\pi_{\theta}^{\text{MPC}}(s) = \pi^*(s)$$

Full MPC parametrization:

$$\begin{aligned} \min_{\mathbf{x}, \mathbf{u}} \quad & T_{\theta}(\mathbf{x}_N) + \sum_{k=0}^{N-1} L_{\theta}(\mathbf{x}_k, \mathbf{u}_k) \\ \text{s.t.} \quad & \mathbf{x}_{k+1} = \mathbf{f}_{\theta}(\mathbf{x}_k, \mathbf{u}_k) \\ & \mathbf{h}_{\theta}(\mathbf{x}_k, \mathbf{u}_k) \leq 0 \\ & \mathbf{x}_0 = \mathbf{s} \end{aligned}$$

Technical assumptions (simplified) MPC model $\mathbf{x}_{k+1} = \mathbf{f}_{\theta}(\mathbf{x}_k, \mathbf{u}_k)$ ought to be such that for $\mathbf{u}_k = \pi^*(\mathbf{x}_k)$ then $V^*(\mathbf{x}_k)$ is bounded for all k and $\mathbf{x}_0$ such that $V^*(\mathbf{x}_0) < \infty$

- MPC model remains feasible & “does not escape” under the optimal policy and admissible initial conditions
- If this works only for a subset of the set of admissible initial conditions, then MPC solves the MDP on that subset
- Condition is unverifiable because it requires V^* and π^*
- In practice this assumption is “self-fulfilling” in the learning context, because RL “tweaks” the parameters to extend the set where the assumption holds

Let's discuss the assumptions

Theorem: under some **technical conditions** and for a “**rich**” **parametrization** of the MPC, there is a θ such that

$$Q_{\theta}^{\text{MPC}}(s, a) = Q^*(s, a)$$
$$\pi_{\theta}^{\text{MPC}}(s) = \pi^*(s)$$

Full MPC parametrization:

$$\begin{aligned} \min_{\mathbf{x}, \mathbf{u}} \quad & T_{\theta}(\mathbf{x}_N) + \sum_{k=0}^{N-1} L_{\theta}(\mathbf{x}_k, \mathbf{u}_k) \\ \text{s.t.} \quad & \mathbf{x}_{k+1} = \mathbf{f}_{\theta}(\mathbf{x}_k, \mathbf{u}_k) \\ & \mathbf{h}_{\theta}(\mathbf{x}_k, \mathbf{u}_k) \leq 0 \\ & \mathbf{x}_0 = \mathbf{s} \end{aligned}$$

Is that all? Unfortunately no: we are making a strong assumption in plain sight.

Implicit Assumption on State

Theorem: under some technical conditions and for a “rich” parametrization of the MPC, there is a θ such that

$$Q_{\theta}^{\text{MPC}}(s, a) = Q^*(s, a)$$

$$\pi_{\theta}^{\text{MPC}}(s) = \pi^*(s)$$

Full MPC parametrization:

$$\min_{\mathbf{x}, \mathbf{u}} \quad T_{\theta}(\mathbf{x}_N) + \sum_{k=0}^{N-1} L_{\theta}(\mathbf{x}_k, \mathbf{u}_k)$$

$$\text{s.t.} \quad \mathbf{x}_{k+1} = \mathbf{f}_{\theta}(\mathbf{x}_k, \mathbf{u}_k)$$

$$\mathbf{h}_{\theta}(\mathbf{x}_k, \mathbf{u}_k) \leq 0$$

$$\mathbf{x}_0 = \mathbf{s}$$

Notes

- MDP has Markov state s , i.e. s is “all we need” to predict the future (statistically)

$$\mathbb{P}[\mathbf{s}_{k+1} \mid \mathbf{s}_k, \dots, 0] = \mathbb{P}[\mathbf{s}_{k+1} \mid \mathbf{s}_k]$$

Beware: MDP is the real world in our story!

Implicit Assumption on State

Theorem: under some technical conditions and for a “rich” parametrization of the MPC, there is a θ such that

$$Q_{\theta}^{\text{MPC}}(s, a) = Q^*(s, a)$$

$$\pi_{\theta}^{\text{MPC}}(s) = \pi^*(s)$$

Full MPC parametrization:

$$\min_{\mathbf{x}, \mathbf{u}} \quad T_{\theta}(\mathbf{x}_N) + \sum_{k=0}^{N-1} L_{\theta}(\mathbf{x}_k, \mathbf{u}_k)$$

$$\text{s.t.} \quad \mathbf{x}_{k+1} = \mathbf{f}_{\theta}(\mathbf{x}_k, \mathbf{u}_k)$$

$$\mathbf{h}_{\theta}(\mathbf{x}_k, \mathbf{u}_k) \leq 0$$

$$\mathbf{x}_0 = \mathbf{s}$$

Notes

- MDP has Markov state s , i.e. s is “all we need” to predict the future (statistically)

$$\mathbb{P}[\mathbf{s}_{k+1} | \mathbf{s}_k, \dots, 0] = \mathbb{P}[\mathbf{s}_{k+1} | \mathbf{s}_k]$$

Beware: MDP is the real world in our story!

- We assume implicitly that MPC is built on that state too

Implicit Assumption on State

Theorem: under some technical conditions and for a “rich” parametrization of the MPC, there is a θ such that

$$Q_{\theta}^{\text{MPC}}(s, a) = Q^*(s, a) \\ \pi_{\theta}^{\text{MPC}}(s) = \pi^*(s)$$

Full MPC parametrization:

$$\begin{aligned} \min_{\mathbf{x}, \mathbf{u}} \quad & T_{\theta}(\mathbf{x}_N) + \sum_{k=0}^{N-1} L_{\theta}(\mathbf{x}_k, \mathbf{u}_k) \\ \text{s.t.} \quad & \mathbf{x}_{k+1} = \mathbf{f}_{\theta}(\mathbf{x}_k, \mathbf{u}_k) \\ & \mathbf{h}_{\theta}(\mathbf{x}_k, \mathbf{u}_k) \leq 0 \\ & \mathbf{x}_0 = \mathbf{s} \end{aligned}$$

Notes

- MDP has Markov state s , i.e. s is “all we need” to predict the future (statistically)

$$\mathbb{P}[s_{k+1} | s_k, \dots, 0] = \mathbb{P}[s_{k+1} | s_k]$$

Beware: MDP is the real world in our story!

- We assume implicitly that MPC is built on that state too
- In practice, that translates into the assumption that we know the Markov state and can access or estimate it

Implicit Assumption on State

Theorem: under some technical conditions and for a “rich” parametrization of the MPC, there is a θ such that

$$Q_{\theta}^{\text{MPC}}(s, a) = Q^*(s, a)$$

$$\pi_{\theta}^{\text{MPC}}(s) = \pi^*(s)$$

Full MPC parametrization:

$$\min_{\mathbf{x}, \mathbf{u}} \quad T_{\theta}(\mathbf{x}_N) + \sum_{k=0}^{N-1} L_{\theta}(\mathbf{x}_k, \mathbf{u}_k)$$

$$\text{s.t.} \quad \mathbf{x}_{k+1} = \mathbf{f}_{\theta}(\mathbf{x}_k, \mathbf{u}_k)$$

$$\mathbf{h}_{\theta}(\mathbf{x}_k, \mathbf{u}_k) \leq 0$$

$$\mathbf{x}_0 = \mathbf{s}$$

Notes

- MDP has Markov state s , i.e. s is “all we need” to predict the future (statistically)

$$\mathbb{P}[s_{k+1} | s_k, \dots, 0] = \mathbb{P}[s_{k+1} | s_k]$$

Beware: MDP is the real world in our story!

- We assume implicitly that MPC is built on that state too
- In practice, that translates into the assumption that we know the Markov state and can access or estimate it
 - state estimation is a classical observation problem

Implicit Assumption on State

Theorem: under some technical conditions and for a “rich” parametrization of the MPC, there is a θ such that

$$Q_{\theta}^{\text{MPC}}(s, a) = Q^*(s, a)$$

$$\pi_{\theta}^{\text{MPC}}(s) = \pi^*(s)$$

Full MPC parametrization:

$$\min_{\mathbf{x}, \mathbf{u}} \quad T_{\theta}(\mathbf{x}_N) + \sum_{k=0}^{N-1} L_{\theta}(\mathbf{x}_k, \mathbf{u}_k)$$

$$\text{s.t.} \quad \mathbf{x}_{k+1} = \mathbf{f}_{\theta}(\mathbf{x}_k, \mathbf{u}_k)$$

$$\mathbf{h}_{\theta}(\mathbf{x}_k, \mathbf{u}_k) \leq 0$$

$$\mathbf{x}_0 = \mathbf{s}$$

Notes

- MDP has Markov state s , i.e. s is “all we need” to predict the future (statistically)

$$\mathbb{P}[s_{k+1} | s_k, \dots, 0] = \mathbb{P}[s_{k+1} | s_k]$$

Beware: MDP is the real world in our story!

- We assume implicitly that MPC is built on that state too
- In practice, that translates into the assumption that we know the Markov state and can access or estimate it
 - state estimation is a classical observation problem
 - but knowing the Markov state (e.g. its dimension) is a strong assumption

Implicit Assumption on State

Theorem: under some technical conditions and for a “rich” parametrization of the MPC, there is a θ such that

$$Q_{\theta}^{\text{MPC}}(s, a) = Q^*(s, a) \\ \pi_{\theta}^{\text{MPC}}(s) = \pi^*(s)$$

Full MPC parametrization:

$$\begin{aligned} \min_{\mathbf{x}, \mathbf{u}} \quad & T_{\theta}(\mathbf{x}_N) + \sum_{k=0}^{N-1} L_{\theta}(\mathbf{x}_k, \mathbf{u}_k) \\ \text{s.t.} \quad & \mathbf{x}_{k+1} = \mathbf{f}_{\theta}(\mathbf{x}_k, \mathbf{u}_k) \\ & \mathbf{h}_{\theta}(\mathbf{x}_k, \mathbf{u}_k) \leq 0 \\ & \mathbf{x}_0 = \mathbf{s} \end{aligned}$$

Notes

- MDP has Markov state s , i.e. s is “all we need” to predict the future (statistically)

$$\mathbb{P}[s_{k+1} | s_k, \dots, 0] = \mathbb{P}[s_{k+1} | s_k]$$

Beware: MDP is the real world in our story!

- We assume implicitly that MPC is built on that state too
- In practice, that translates into the assumption that we know the Markov state and can access or estimate it
 - 1 state estimation is a classical observation problem
 - 2 but knowing the Markov state (e.g. its dimension) is a strong assumption
 - 3 no “robustness card” here because we work on performance

Implicit Assumption on State

Theorem: under some technical conditions and for a “rich” parametrization of the MPC, there is a θ such that

$$Q_{\theta}^{\text{MPC}}(s, a) = Q^*(s, a) \\ \pi_{\theta}^{\text{MPC}}(s) = \pi^*(s)$$

Full MPC parametrization:

$$\begin{aligned} \min_{\mathbf{x}, \mathbf{u}} \quad & T_{\theta}(\mathbf{x}_N) + \sum_{k=0}^{N-1} L_{\theta}(\mathbf{x}_k, \mathbf{u}_k) \\ \text{s.t.} \quad & \mathbf{x}_{k+1} = \mathbf{f}_{\theta}(\mathbf{x}_k, \mathbf{u}_k) \\ & \mathbf{h}_{\theta}(\mathbf{x}_k, \mathbf{u}_k) \leq 0 \\ & \mathbf{x}_0 = \mathbf{s} \end{aligned}$$

Notes

- MDP has Markov state s , i.e. s is “all we need” to predict the future (statistically)

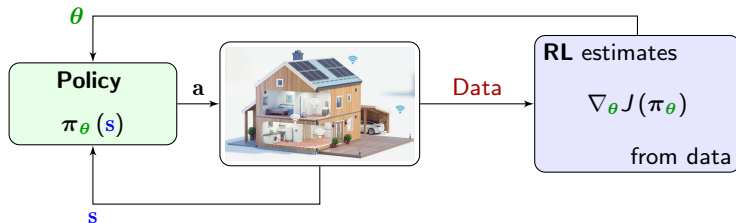
$$\mathbb{P}[s_{k+1} | s_k, \dots, 0] = \mathbb{P}[s_{k+1} | s_k]$$

Beware: MDP is the real world in our story!

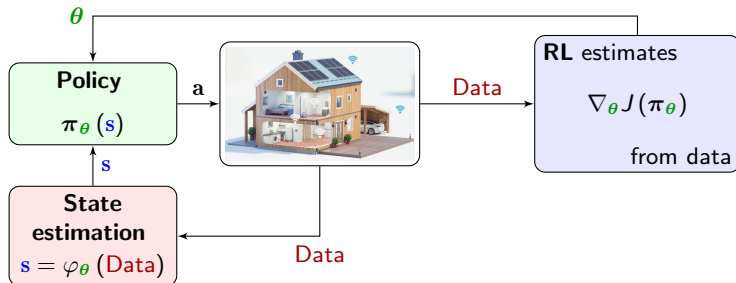
- We assume implicitly that MPC is built on that state too
- In practice, that translates into the assumption that we know the Markov state and can access or estimate it
 - 1 state estimation is a classical observation problem
 - 2 but knowing the Markov state (e.g. its dimension) is a strong assumption
 - 3 no “robustness card” here because we work on performance

Can we address that? Yes, but with caution

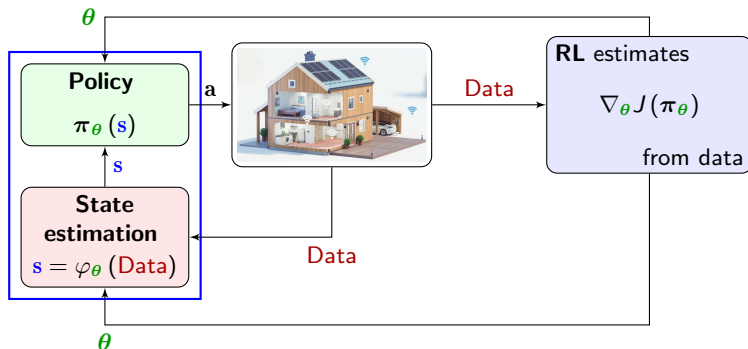
State Estimation in the Learning Loop



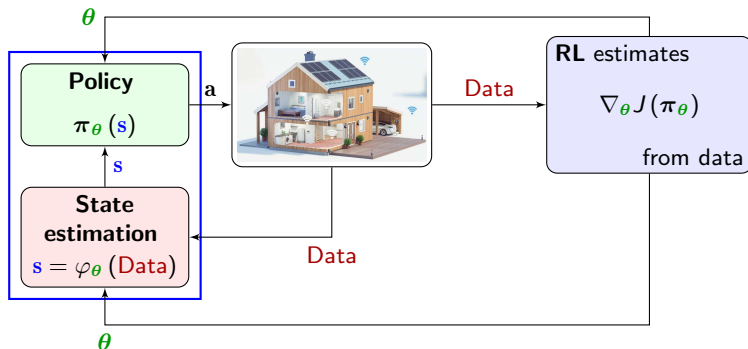
State Estimation in the Learning Loop



State Estimation in the Learning Loop



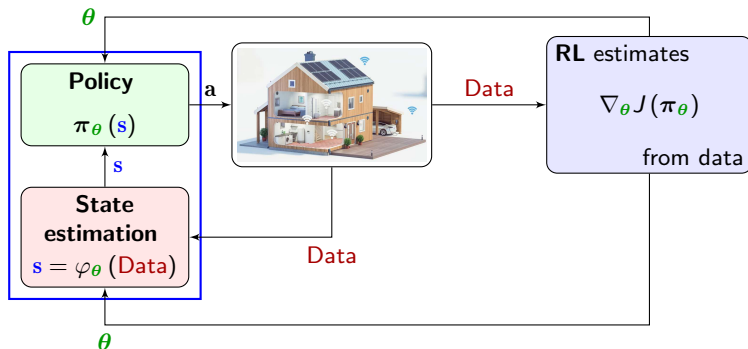
State Estimation in the Learning Loop



Remarks

- Policy is $\pi_{\theta}^E(\text{Data}) = \pi_{\theta}(\varphi_{\theta}(\text{Data}))$

State Estimation in the Learning Loop

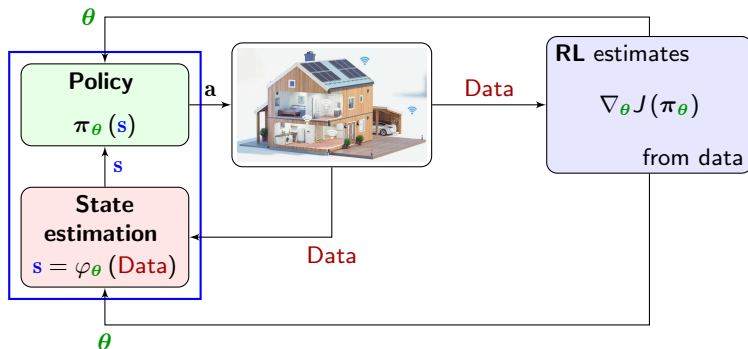


Remarks

- Policy is $\pi_{\theta}^E(\text{Data}) = \pi_{\theta}(\varphi_{\theta}(\text{Data}))$
- Gradient of the policy now is:

$$\nabla_{\theta} \pi_{\theta}^E(\text{Data}) = \nabla_{\theta} \pi_{\theta}(s) + \nabla_{\theta} \varphi_{\theta}(\text{Data}) \nabla_s \pi_{\theta}(s)$$

State Estimation in the Learning Loop



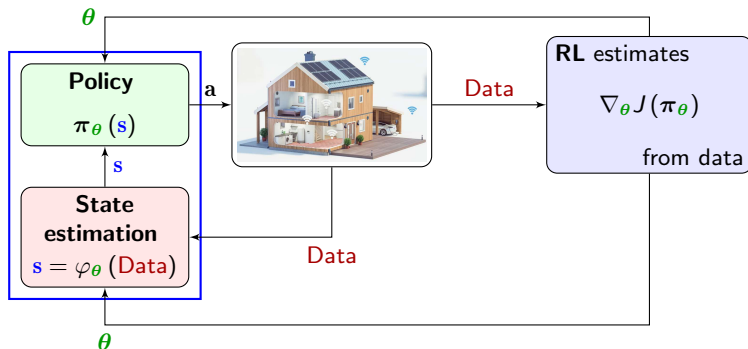
Remarks

- Policy is $\pi_{\theta}^E(\text{Data}) = \pi_{\theta}(\varphi_{\theta}(\text{Data}))$
- Gradient of the policy now is:

$$\nabla_{\theta} \pi_{\theta}^E(\text{Data}) = \nabla_{\theta} \pi_{\theta}(s) + \nabla_{\theta} \varphi_{\theta}(\text{Data}) \nabla_s \pi_{\theta}(s)$$

- Critic is in the form $Q^{\pi_{\theta}^E}(\text{Data}, a)$

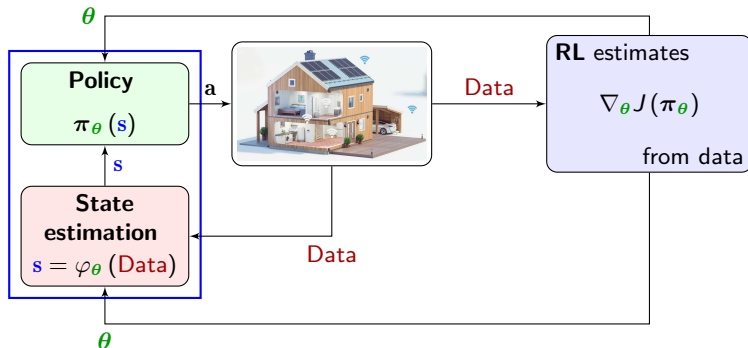
State Estimation in the Learning Loop



If $\varphi_{\theta}(\text{Data})$ is

- MHE, gradient using NLP sensitivities (same as MPC)
- DNN, then efficient sensitivity computations exist
- Bayesian inference? Belief state? Open research topic...

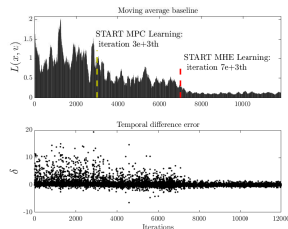
State Estimation in the Learning Loop



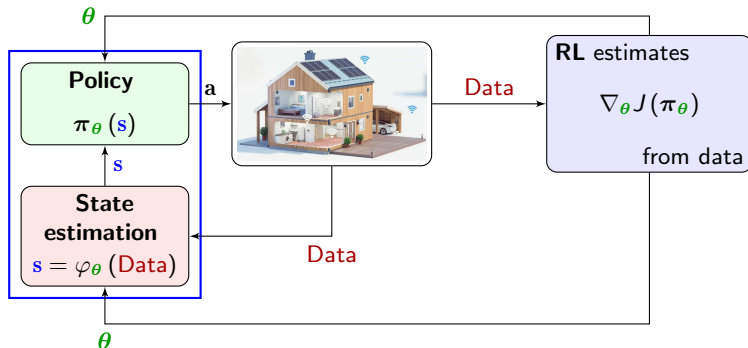
If $\varphi_{\theta}(\text{Data})$ is

- MHE, gradient using NLP sensitivities (same as MPC)
- DNN, then efficient sensitivity computations exist
- Bayesian inference? Belief state? Open research topic...

These ideas are strongly connected to the notion of “latent space” in AI, i.e. a state is eventually an abstraction of the real-world observations into a “low”-dimensional object which carries relevant predictive information



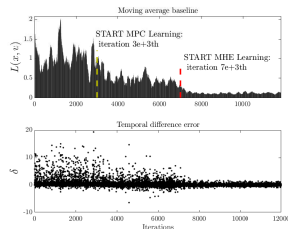
State Estimation in the Learning Loop



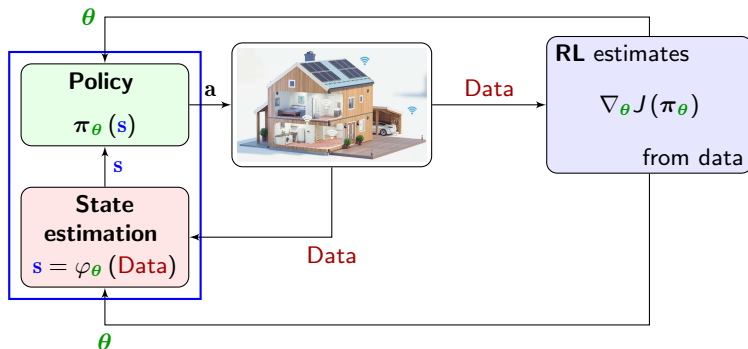
If $\varphi_\theta(\text{Data})$ is

- MHE, gradient using NLP sensitivities (same as MPC)
- DNN, then efficient sensitivity computations exist
- Bayesian inference? Belief state? Open research topic...

Through RL, s can in principle “drift” away from its original physical meaning. This is a current observation in latent space construction. Regularize towards physical meaning? Open research question.



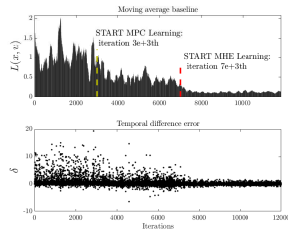
State Estimation in the Learning Loop



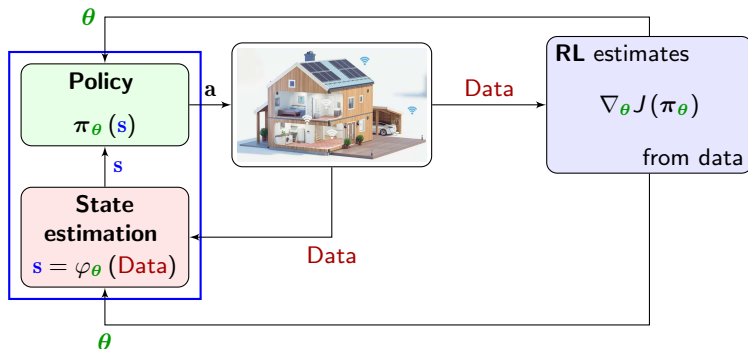
If $\varphi_{\theta}(\text{Data})$ is

- MHE, gradient using NLP sensitivities (same as MPC)
- DNN, then efficient sensitivity computations exist
- Bayesian inference? Belief state? Open research topic...

Difficulty in this context, $\text{Data} \equiv \text{state}$



State Estimation in the Learning Loop

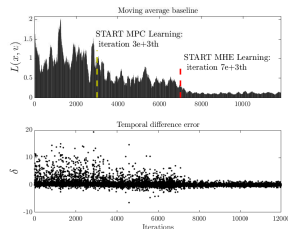


If $\varphi_{\theta}(\text{Data})$ is

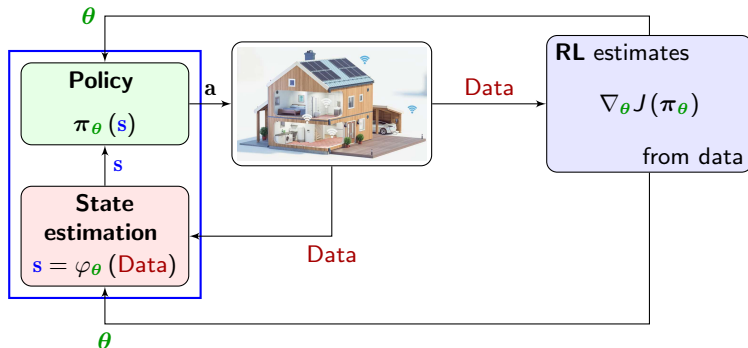
- MHE, gradient using NLP sensitivities (same as MPC)
- DNN, then efficient sensitivity computations exist
- Bayesian inference? Belief state? Open research topic...

Difficulty in this context, $\text{Data} \equiv \text{state}$, hence

$$\nabla_{\theta} J(\pi_{\theta}) = \mathbb{E} \left[\nabla_{\theta} \pi_{\theta}^E(\text{Data}) \nabla_a Q^{\pi_{\theta}}(\text{Data}, a) \Big|_{a=\pi_{\theta}(s)} \right]$$



State Estimation in the Learning Loop



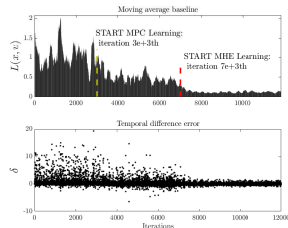
If $\varphi_\theta(\text{Data})$ is

- MHE, gradient using NLP sensitivities (same as MPC)
- DNN, then efficient sensitivity computations exist
- Bayesian inference? Belief state? Open research topic...

Difficulty in this context, $\text{Data} \equiv \text{state}$, hence

$$\nabla_\theta J(\pi_\theta) = \mathbb{E} \left[\nabla_\theta \pi_\theta^E(\text{Data}) \nabla_a Q^\pi(\text{Data}, a) \Big|_{a=\pi_\theta(s)} \right]$$

requires high-dimensional “critic”



RL+MPC “Without” State - Input-Output predictions

Assumption: a long-enough window of past observations-action is a Markov state.

$$\mathbf{s}_t = \{\mathbf{o}_{t,\dots,t-n}, \mathbf{a}_{t,\dots,t-m}\} \rightarrow \mathbb{P}[\mathbf{o}_{t+1} | \mathbf{s}_t] = \mathbb{P}[\mathbf{o}_{t+1} | \mathbf{o}_{t,\dots,0}, \mathbf{a}_{t,\dots,0}]$$

RL+MPC “Without” State - Input-Output predictions

Assumption: a long-enough window of past observations-action is a Markov state.

$$\mathbf{s}_t = \{\mathbf{o}_{t,\dots,t-n}, \mathbf{a}_{t,\dots,t-m}\} \rightarrow \mathbb{P}[\mathbf{o}_{t+1} | \mathbf{s}_t] = \mathbb{P}[\mathbf{o}_{t+1} | \mathbf{o}_{t,\dots,0}, \mathbf{a}_{t,\dots,0}]$$

Input-output MPC, e.g. AR model

$$\min_{\mathbf{y}, \mathbf{u}} \sum_{k=0}^N L(\mathbf{y}_k, \mathbf{u}_k)$$

$$\text{s.t. } \mathbf{y}_{k+1} = \mathbf{f}(\mathbf{y}_k, \dots, \mathbf{y}_{k-n}, \mathbf{u}_k, \dots, \mathbf{u}_{k-m})$$

$$\mathbf{h}(\mathbf{y}_k, \mathbf{u}_k) \leq 0$$

$$\mathbf{y}_{k-i} = \mathbf{o}_{t-i}, \quad i = 0, \dots, n$$

$$\mathbf{u}_{k-i} = \mathbf{a}_{t-i}, \quad i = 0, \dots, m$$

RL+MPC “Without” State - Input-Output predictions

Assumption: a long-enough window of past observations-action is a Markov state.

$$\mathbf{s}_t = \{\mathbf{o}_{t,\dots,t-n}, \mathbf{a}_{t,\dots,t-m}\} \rightarrow \mathbb{P}[\mathbf{o}_{t+1} | \mathbf{s}_t] = \mathbb{P}[\mathbf{o}_{t+1} | \mathbf{o}_{t,\dots,0}, \mathbf{a}_{t,\dots,0}]$$

Input-output MPC, e.g. AR model

$$\begin{aligned} \min_{\mathbf{y}, \mathbf{u}} \quad & \sum_{k=0}^N L(\mathbf{y}_k, \mathbf{u}_k) \\ \text{s.t.} \quad & \mathbf{y}_{k+1} = \mathbf{f}(\mathbf{y}_k, \dots, \mathbf{y}_{k-n}, \mathbf{u}_k, \dots, \mathbf{u}_{k-m}) \\ & \mathbf{h}(\mathbf{y}_k, \mathbf{u}_k) \leq 0 \\ & \mathbf{y}_{k-i} = \mathbf{o}_{t-i}, \quad i = 0, \dots, n \\ & \mathbf{u}_{k-i} = \mathbf{a}_{t-i}, \quad i = 0, \dots, m \end{aligned}$$

Notes

- Build model $\mathbf{f}$ from data

$$\mathcal{D} = \{\mathbf{o}_{0,\dots,M}, \mathbf{a}_{0,\dots,M}\}$$

- With regularization, one can overestimate n and $m \rightarrow$ no need for an accurate order

RL+MPC “Without” State - Input-Output predictions

Assumption: a long-enough window of past observations-action is a Markov state.

$$\mathbf{s}_t = \{\mathbf{o}_{t,\dots,t-n}, \mathbf{a}_{t,\dots,t-m}\} \rightarrow \mathbb{P}[\mathbf{o}_{t+1} | \mathbf{s}_t] = \mathbb{P}[\mathbf{o}_{t+1} | \mathbf{o}_{t,\dots,0}, \mathbf{a}_{t,\dots,0}]$$

Input-output MPC, e.g. AR model

$$\min_{\mathbf{y}, \mathbf{u}} \Phi_{\theta}(\mathbf{y}, \mathbf{u})$$

$$\text{s.t. } \mathbf{y}_{k+1} = \mathbf{f}_{\theta}(\mathbf{y}_k, \dots, \mathbf{y}_{k-n}, \mathbf{u}_k, \dots, \mathbf{u}_{k-m})$$

$$\mathbf{H}_{\theta}(\mathbf{y}, \mathbf{u}) \leq 0$$

$$\mathbf{y}_{k-i} = \mathbf{o}_{t-i}, \quad i = 0, \dots, n$$

$$\mathbf{u}_{k-i} = \mathbf{a}_{t-i}, \quad i = 0, \dots, m$$

Notes

- Build model $\mathbf{f}$ from data

$$\mathcal{D} = \{\mathbf{o}_{0,\dots,M}, \mathbf{a}_{0,\dots,M}\}$$

- With regularization, one can overestimate n and $m \rightarrow$ no need for an accurate order

RL-MPC theory applies but...

- Modifications is no longer localized in time

RL+MPC “Without” State - Input-Output predictions

Assumption: a long-enough window of past observations-action is a Markov state.

$$\mathbf{s}_t = \{\mathbf{o}_{t,\dots,t-n}, \mathbf{a}_{t,\dots,t-m}\} \rightarrow \mathbb{P}[\mathbf{o}_{t+1} | \mathbf{s}_t] = \mathbb{P}[\mathbf{o}_{t+1} | \mathbf{o}_{t,\dots,0}, \mathbf{a}_{t,\dots,0}]$$

Input-output MPC, e.g. AR model

$$\min_{\mathbf{y}, \mathbf{u}} \Phi_{\theta}(\mathbf{y}, \mathbf{u})$$

$$\text{s.t. } \mathbf{y}_{k+1} = \mathbf{f}_{\theta}(\mathbf{y}_k, \dots, \mathbf{y}_{k-n}, \mathbf{u}_k, \dots, \mathbf{u}_{k-m})$$

$$\mathbf{H}_{\theta}(\mathbf{y}, \mathbf{u}) \leq 0$$

$$\mathbf{y}_{k-i} = \mathbf{o}_{t-i}, \quad i = 0, \dots, n$$

$$\mathbf{u}_{k-i} = \mathbf{a}_{t-i}, \quad i = 0, \dots, m$$

Notes

- Build model $\mathbf{f}$ from data

$$\mathcal{D} = \{\mathbf{o}_{0,\dots,M}, \mathbf{a}_{0,\dots,M}\}$$

- With regularization, one can overestimate n and $m \rightarrow$ no need for an accurate order

RL-MPC theory applies but...

- Modifications is no longer localized in time
- High-dimensional parameter space for RL

RL+MPC “Without” State - Input-Output predictions

Assumption: a long-enough window of past observations-action is a Markov state.

$$\mathbf{s}_t = \{\mathbf{o}_{t,\dots,t-n}, \mathbf{a}_{t,\dots,t-m}\} \rightarrow \mathbb{P}[\mathbf{o}_{t+1} | \mathbf{s}_t] = \mathbb{P}[\mathbf{o}_{t+1} | \mathbf{o}_{t,\dots,0}, \mathbf{a}_{t,\dots,0}]$$

Input-output MPC, e.g. AR model

$$\begin{aligned} \min_{\mathbf{y}, \mathbf{u}} \quad & \Phi_{\theta}(\mathbf{y}, \mathbf{u}) \\ \text{s.t.} \quad & \mathbf{y} = \mathbf{F}_{\theta}(\mathbf{y}_0, \dots, \mathbf{y}_{-n}, \mathbf{u}_N, \dots, \mathbf{u}_{-m}) \\ & \mathbf{H}_{\theta}(\mathbf{y}, \mathbf{u}) \leq 0 \\ & \mathbf{y}_{k-i} = \mathbf{o}_{t-i}, \quad i = 0, \dots, n \\ & \mathbf{u}_{k-i} = \mathbf{a}_{t-i}, \quad i = 0, \dots, m \end{aligned}$$

Notes

- Build model $\mathbf{f}$ from data

$$\mathcal{D} = \{\mathbf{o}_0, \dots, \mathbf{o}_M, \mathbf{a}_0, \dots, \mathbf{a}_M\}$$

- With regularization, one can overestimate n and $m \rightarrow$ no need for an accurate order
- Extension: multi-step predictors, can have powerful statistical properties

RL-MPC theory applies but...

- Modifications is no longer localized in time
- High-dimensional parameter space for RL

What we have seen:

- MPC can be understood as a model of the optimal action-value function Q^* of real-world MDPs and/or of the optimal policy π^*
- MPC cost (and constraints) become part of that model
- Model that best fits the real-world does not (necessarily) yield the best policy
- RL is a toolbox to tune the MPC as a model of the MDP solution
- MPC state space should match the real world, strong assumption that can be alleviated

What we have seen:

- MPC can be understood as a model of the optimal action-value function Q^* of real-world MDPs and/or of the optimal policy π^*
- MPC cost (and constraints) become part of that model
- Model that best fits the real-world does not (necessarily) yield the best policy
- RL is a toolbox to tune the MPC as a model of the MDP solution
- MPC state space should match the real world, strong assumption that can be alleviated

What we will do next: RL over MPC

- Safe & Stable RL over MPC (*In the afternoon*)
- RL over MPC with belief states – a future prospect (*In the afternoon*)
- Beyond MPC – Model-based Decisions and AI for decisions (*Tomorrow*)