Reinforcement Learning over Model Predictive Control
– Why does it work? –

Dirk Reinhardt

Department of Engineering Cybernetics, NTNU, Norway

Predict, Control, Learn: Combining Model Predictive Control and Reinforcement Learning
Tutorial Session at ECC 2026, Reykjavík, July 2026

universität freiburg

NTNU | Norwegian University of Science and Technology

University of Zurich[UZH]

ROBOTICS & PERCEPTION GROUP

Tutorial Roadmap

	Talk	*Presenter*	*Duration*
✓	Markov Decision Processes – a unifying perspective on MPC and RL	Jasper Hoffmann	20min
✓	Synthesis of Model Predictive Control and Reinforcement Learning	Rudolf Reiter	30min
➤	**Reinforcement Learning over MPC: Why does it work?**	**Dirk Reinhardt**	**30min**
	Differentiable NMPC with `acados`	Katrin Baumgärtner	20min
	MPCRL with `leap-c`	Jasper Hoffmann	20min

At the end of each talk there will be time for questions!

Tutorial webpage for slides and more!

Outline

Background: MPC, MDPs, and Model Fitting

MPC as a Model of the MDP Solution

Reinforcement Learning Tunes the MPC

Outline

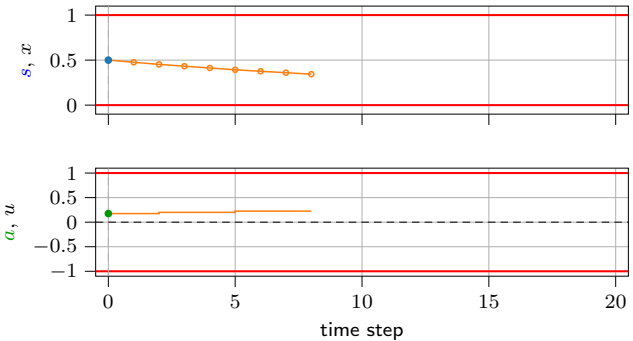
MPC: at current state s

$$\min_{x,u} \quad \bar{V}(x_N) + \sum_{k=0}^{N-1} l(x_k, u_k)$$

$$\text{s.t.} \quad x_{k+1} = f(x_k, u_k)$$

$$h(x_k, u_k) \geq 0, \quad x_0 = s$$

apply action $a = u_0^\star$, then repeat



MPC: at current state s

$$\min_{x,u} \quad \bar{V}(x_N) + \sum_{k=0}^{N-1} l(x_k, u_k)$$

$$\text{s.t.} \quad x_{k+1} = f(x_k, u_k)$$

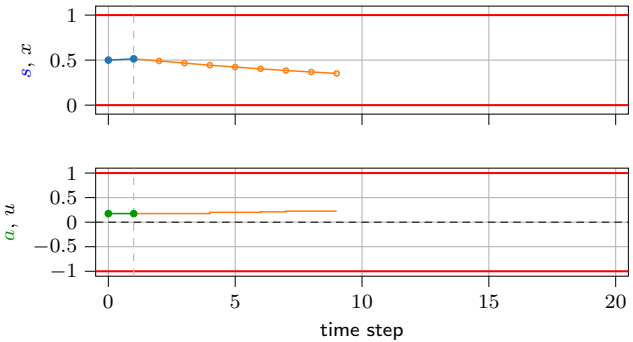
$$h(x_k, u_k) \geq 0, \quad x_0 = s$$

apply action $a = u_0^\star$, then repeat

MPC

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

$$\pi^{\text{MPC}}(s) = u_0^\star$$



MPC: at current state s

$$\min_{x,u} \quad \bar{V}(x_N) + \sum_{k=0}^{N-1} l(x_k, u_k)$$

$$\text{s.t.} \quad x_{k+1} = f(x_k, u_k)$$

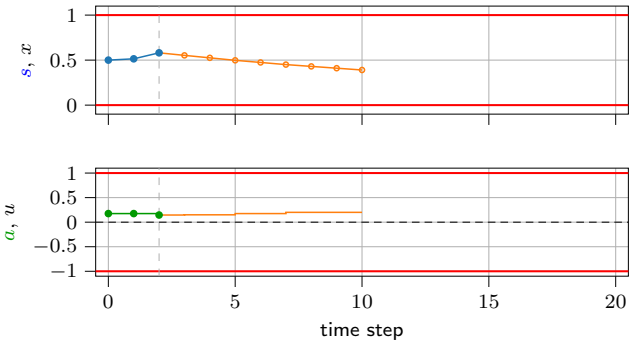
$$h(x_k, u_k) \geq 0, \quad x_0 = s$$

apply action $a = u_0^\star$, then repeat

MPC

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

$$\pi^{\text{MPC}}(s) = u_0^\star$$



MPC: at current state s

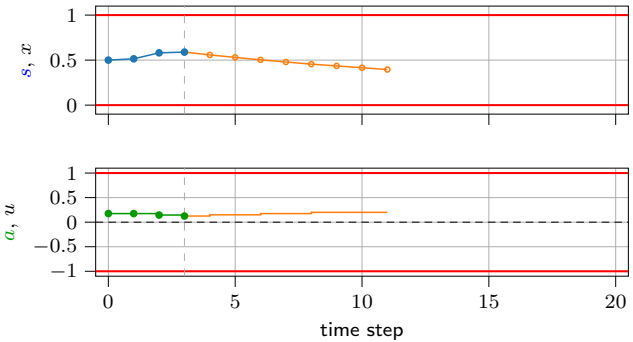
$$\min_{x,u} \quad \bar{V}(x_N) + \sum_{k=0}^{N-1} l(x_k, u_k)$$
$$\text{s.t.} \quad x_{k+1} = f(x_k, u_k)$$
$$h(x_k, u_k) \geq 0, \quad x_0 = s$$

apply action $a = u_0^\star$, then repeat

MPC

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

$$\pi^{\text{MPC}}(s) = u_0^\star$$



MPC: at current state s

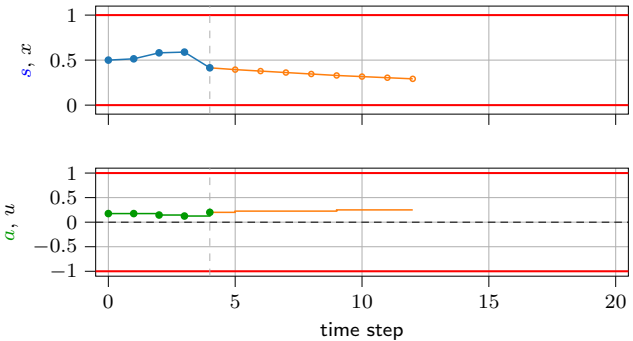
$$\min_{x,u} \quad \bar{V}(x_N) + \sum_{k=0}^{N-1} l(x_k, u_k)$$
$$\text{s.t.} \quad x_{k+1} = f(x_k, u_k)$$
$$h(x_k, u_k) \geq 0, \quad x_0 = s$$

apply action $a = u_0^\star$, then repeat

MPC

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

$$\pi^{\text{MPC}}(s) = u_0^\star$$



MPC: at current state s

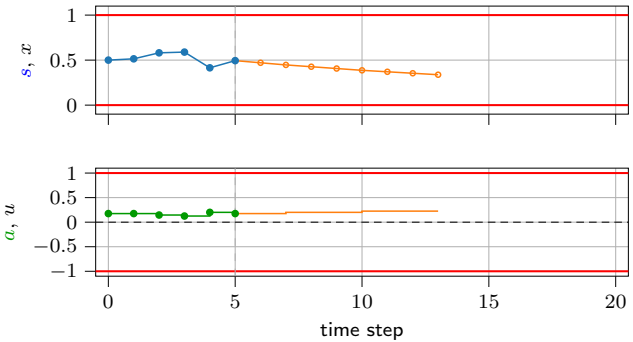
$$\min_{x,u} \quad \bar{V}(x_N) + \sum_{k=0}^{N-1} l(x_k, u_k)$$
$$\text{s.t.} \quad x_{k+1} = f(x_k, u_k)$$
$$h(x_k, u_k) \geq 0, \quad x_0 = s$$

apply action $a = u_0^\star$, then repeat

MPC

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

$$\pi^{\text{MPC}}(s) = u_0^\star$$



MPC: at current state s

$$\min_{x,u} \quad \bar{V}(x_N) + \sum_{k=0}^{N-1} l(x_k, u_k)$$

$$\text{s.t.} \quad x_{k+1} = f(x_k, u_k)$$

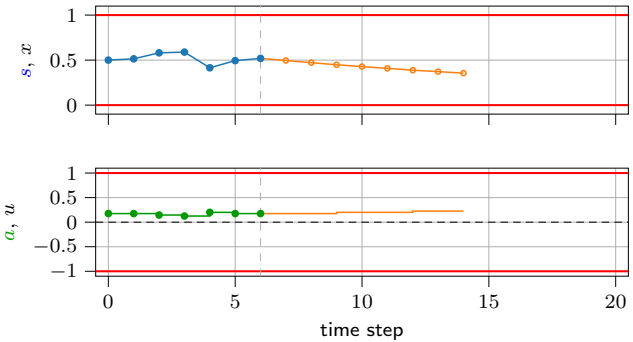
$$h(x_k, u_k) \geq 0, \quad x_0 = s$$

apply action $a = u_0^\star$, then repeat

MPC

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

$$\pi^{\text{MPC}}(s) = u_0^\star$$



MPC: at current state s

$$\min_{x,u} \quad \bar{V}(x_N) + \sum_{k=0}^{N-1} l(x_k, u_k)$$

$$\text{s.t.} \quad x_{k+1} = f(x_k, u_k)$$

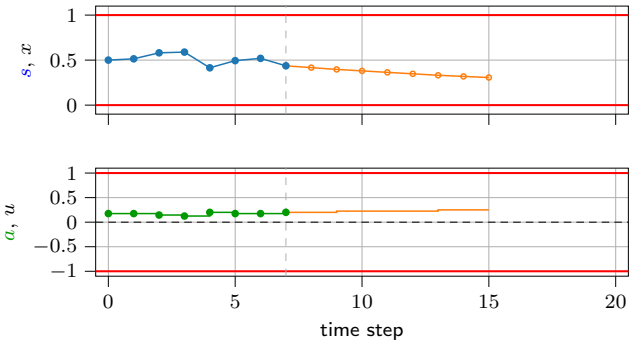
$$h(x_k, u_k) \geq 0, \quad x_0 = s$$

apply action $a = u_0^\star$, then repeat

MPC

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

$$\pi^{\text{MPC}}(s) = u_0^\star$$



MPC: at current state s

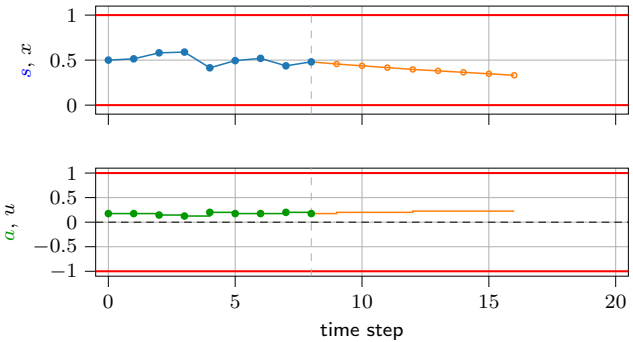
$$\min_{x,u} \quad \bar{V}(x_N) + \sum_{k=0}^{N-1} l(x_k, u_k)$$
$$\text{s.t.} \quad x_{k+1} = f(x_k, u_k)$$
$$h(x_k, u_k) \geq 0, \quad x_0 = s$$

apply action $a = u_0^\star$, then repeat

MPC

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

$$\pi^{\text{MPC}}(s) = u_0^\star$$



MPC: at current state s

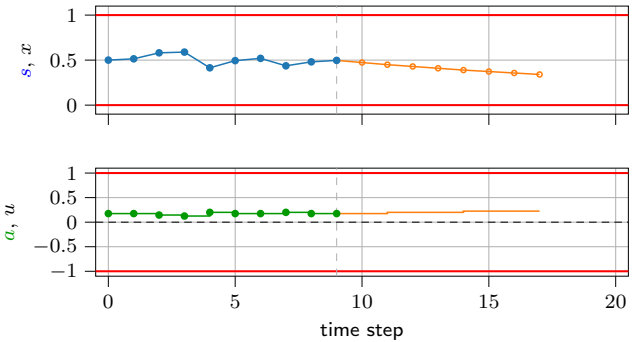
$$\min_{x,u} \quad \bar{V}(x_N) + \sum_{k=0}^{N-1} l(x_k, u_k)$$
$$\text{s.t.} \quad x_{k+1} = f(x_k, u_k)$$
$$h(x_k, u_k) \geq 0, \quad x_0 = s$$

apply action $a = u_0^\star$, then repeat

MPC

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

$$\pi^{\text{MPC}}(s) = u_0^\star$$



MPC: at current state s

$$\min_{x,u} \quad \bar{V}(x_N) + \sum_{k=0}^{N-1} l(x_k, u_k)$$

$$\text{s.t.} \quad x_{k+1} = f(x_k, u_k)$$

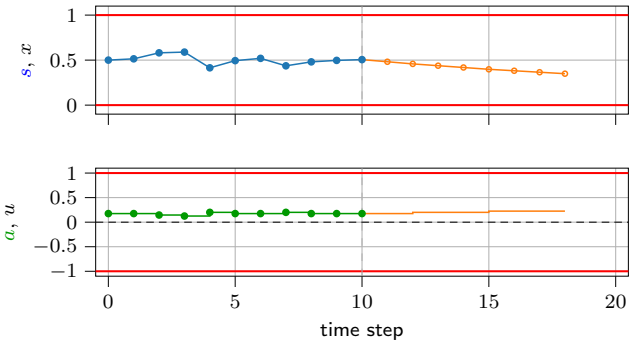
$$h(x_k, u_k) \geq 0, \quad x_0 = s$$

apply action $a = u_0^\star$, then repeat

MPC

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

$$\pi^{\text{MPC}}(s) = u_0^\star$$



Theoretical Framework to connect RL and MPC

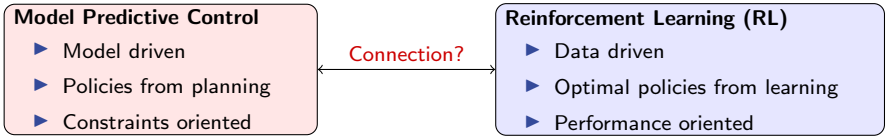
Model Predictive Control

- 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

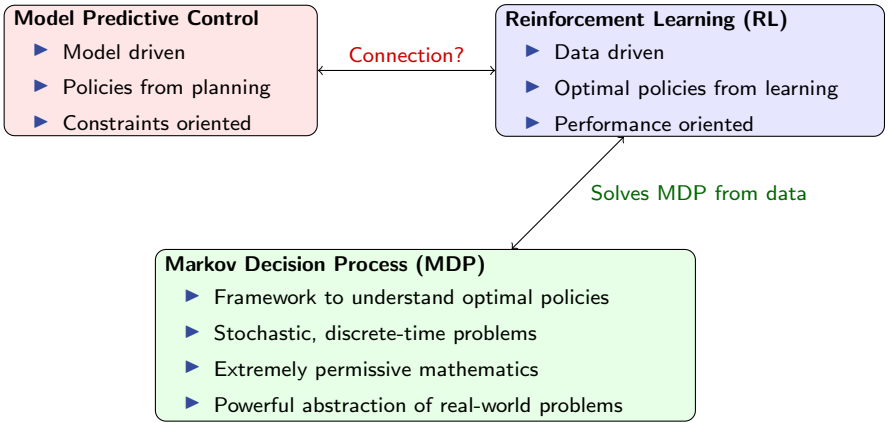
Model Predictive Control
- Model driven
- Policies from planning
- Constraints oriented

Connection?

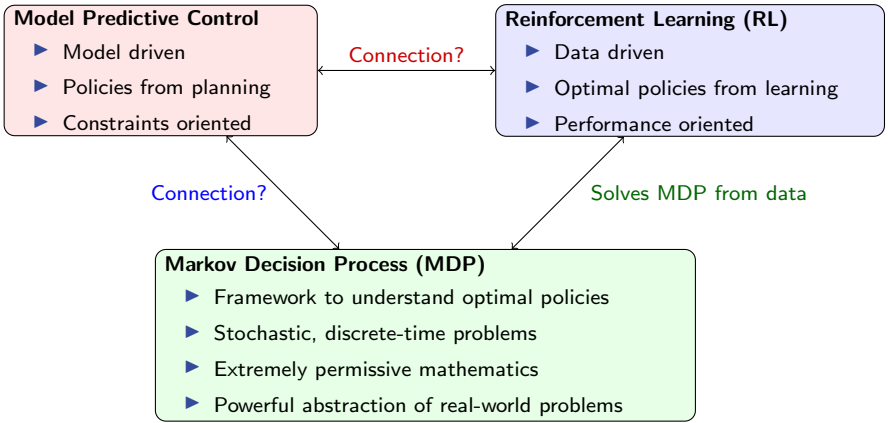
Reinforcement Learning (RL)
- Data driven
- Optimal policies from learning
- Performance oriented

Markov Decision Process (MDP)
- Framework to understand optimal policies
- Stochastic, discrete-time problems
- Extremely permissive mathematics
- Powerful abstraction of real-world problems

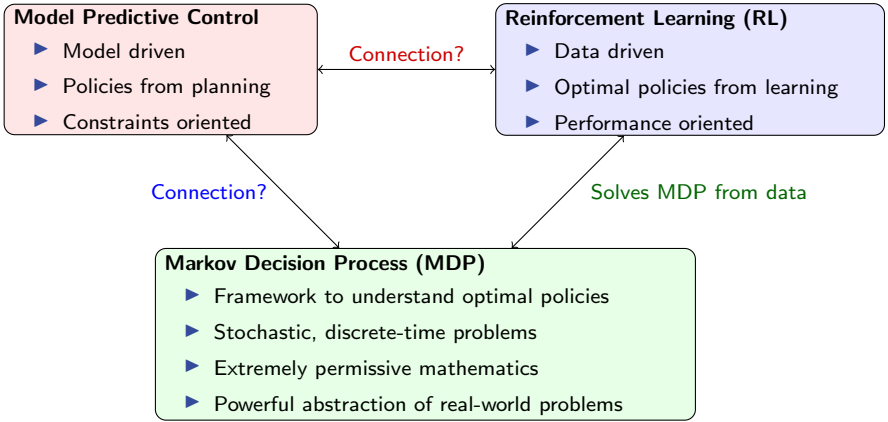
Theoretical Framework to connect RL and MPC



Theoretical Framework to connect RL and MPC

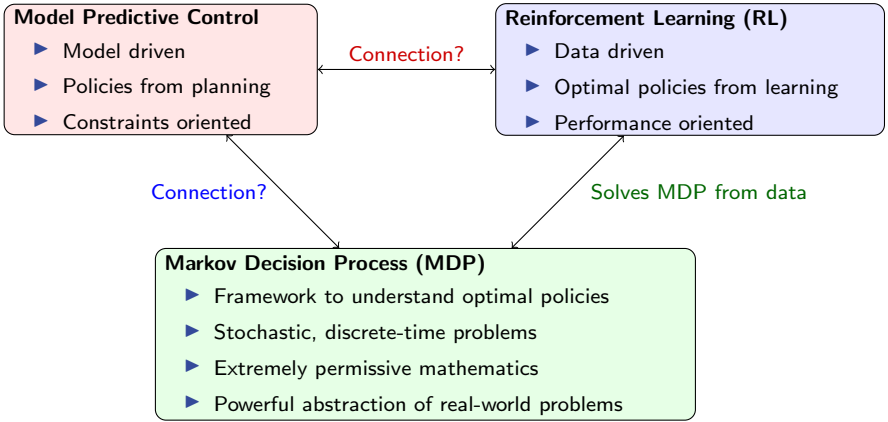


Theoretical Framework to connect RL and MPC



We have connected MPC and RL from an "implementation" point of view

Theoretical Framework to connect RL and MPC



We have connected MPC and RL from an "implementation" point of view
The theory behind is about the connection between MPC and MDPs

Stochastic state transitions (real world)

$$s^+ \sim P(\cdot|s,a)$$

Cost function (instant performance) $L(s,a) \in \mathbb{R}$

Stochastic state transitions (real world)

$$s^+ \sim P(\cdot|s,a)$$

Cost function (instant performance) $L(s,a) \in \mathbb{R}$

- **Policy**

$$a = \pi(s)\,, \quad \text{or} \quad A \sim \pi(\cdot|s)$$

Stochastic state transitions (real world)

$$s^+ \sim P(\cdot|s,a)$$

Cost function (instant performance) $L(s,a) \in \mathbb{R}$

- **Policy**

$$a = \pi(s)\,, \quad \text{or} \quad A \sim \pi(\cdot|s)$$

- **Closed-loop performance**

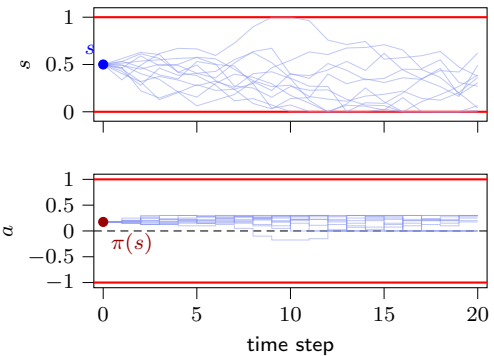
$$J(\pi) = \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k l(s_k, a_k) \,\middle|\, \pi\right]$$

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

Stochastic state transitions (real world)

$$s^+ \sim P(\cdot|s,a)$$

Cost function (instant performance) $L(s,a) \in \mathbb{R}$



- **Policy**

$$a = \pi(s)\,, \quad \text{or} \quad A \sim \pi(\cdot|s)$$

- **Closed-loop performance**

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

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

- **Optimal policy**: $\pi^\star$ from

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

Stochastic state transitions (real world)

$$s^+ \sim P(\cdot|s,a)$$

- **Policy**

$$a = \pi(s)\,, \quad \text{or} \quad A \sim \pi(\cdot|s)$$

- **Closed-loop performance**

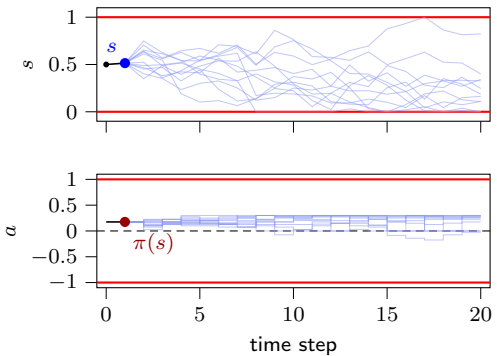
$$J(\pi) = \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k l(s_k, a_k) \,\middle|\, \pi\right]$$

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

- **Optimal policy**: $\pi^\star$ from

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

Cost function (instant performance) $L(s,a) \in \mathbb{R}$



Stochastic state transitions (real world)

$$s^+ \sim P(\cdot|s,a)$$

- **Policy**

$$a = \pi(s)\,, \quad \text{or} \quad A \sim \pi(\cdot|s)$$

- **Closed-loop performance**

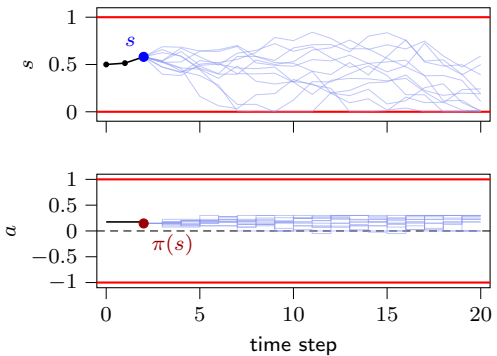
$$J(\pi) = \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k l(s_k, a_k) \,\middle|\, \pi\right]$$

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

- **Optimal policy**: $\pi^\star$ from

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

Cost function (instant performance) $L(s,a) \in \mathbb{R}$



Stochastic state transitions (real world)

$$s^+ \sim P(\cdot|s,a)$$

- **Policy**

$$a = \pi(s)\,, \quad \text{or} \quad A \sim \pi(\cdot|s)$$

- **Closed-loop performance**

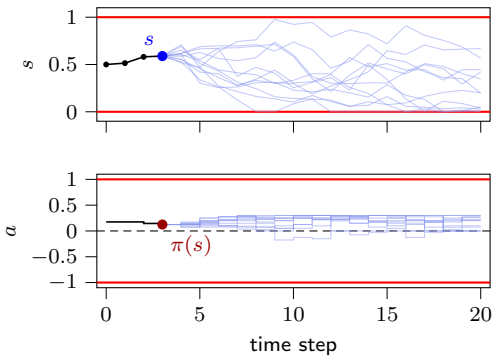
$$J(\pi) = \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k l(s_k, a_k) \,\middle|\, \pi\right]$$

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

- **Optimal policy**: $\pi^\star$ from

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

Cost function (instant performance) $L(s,a) \in \mathbb{R}$



Stochastic state transitions (real world)

$$s^+ \sim P(\cdot|s,a)$$

- **Policy**

$$a = \pi(s)\,, \quad \text{or} \quad A \sim \pi(\cdot|s)$$

- **Closed-loop performance**

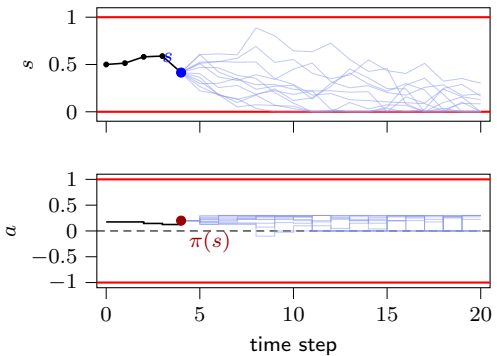
$$J(\pi) = \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k l(s_k, a_k) \,\middle|\, \pi\right]$$

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

- **Optimal policy**: $\pi^\star$ from

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

Cost function (instant performance) $L(s,a) \in \mathbb{R}$



Stochastic state transitions (real world)

$$s^+ \sim P(\cdot|s,a)$$

- **Policy**

$$a = \pi(s)\,, \quad \text{or} \quad A \sim \pi(\cdot|s)$$

- **Closed-loop performance**

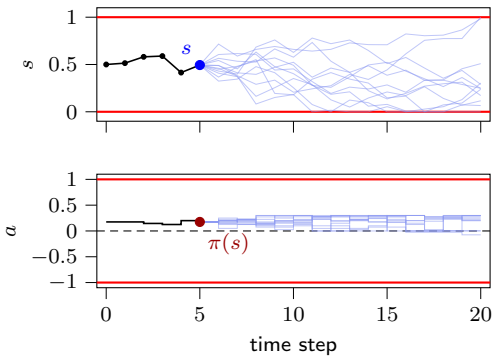
$$J(\pi) = \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k l(s_k, a_k) \,\middle|\, \pi\right]$$

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

- **Optimal policy**: $\pi^\star$ from

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

Cost function (instant performance) $L(s,a) \in \mathbb{R}$



Stochastic state transitions (real world)

$$s^+ \sim P(\cdot|s,a)$$

- **Policy**

$$a = \pi(s)\,, \quad \text{or} \quad A \sim \pi(\cdot|s)$$

- **Closed-loop performance**

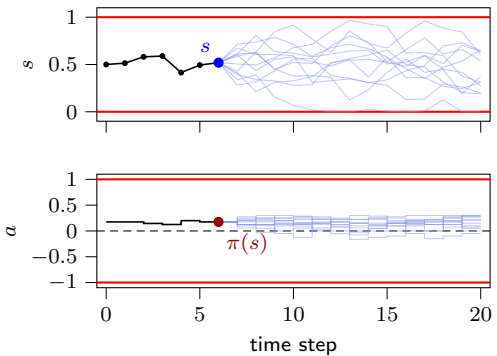
$$J(\pi) = \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k l(s_k, a_k) \,\middle|\, \pi\right]$$

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

- **Optimal policy**: $\pi^\star$ from

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

Cost function (instant performance) $L(s,a) \in \mathbb{R}$



Stochastic state transitions (real world)

$$s^+ \sim P(\cdot|s,a)$$

- **Policy**

$$a = \pi(s)\,, \quad \text{or} \quad A \sim \pi(\cdot|s)$$

- **Closed-loop performance**

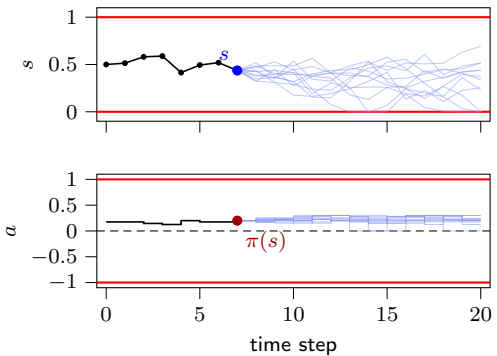
$$J(\pi) = \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k l(s_k, a_k) \,\middle|\, \pi\right]$$

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

- **Optimal policy**: $\pi^\star$ from

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

Cost function (instant performance) $L(s,a) \in \mathbb{R}$



Stochastic state transitions (real world)

$$s^+ \sim P(\cdot|s,a)$$

- **Policy**

$$a = \pi(s)\,, \quad \text{or} \quad A \sim \pi(\cdot|s)$$

- **Closed-loop performance**

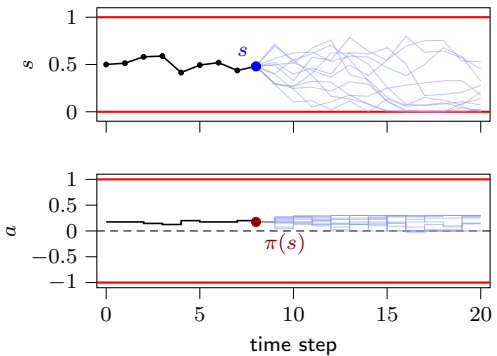
$$J(\pi) = \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k l(s_k, a_k) \,\middle|\, \pi\right]$$

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

- **Optimal policy**: $\pi^\star$ from

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

Cost function (instant performance) $L(s,a) \in \mathbb{R}$



Stochastic state transitions (real world)

$$s^+ \sim P(\cdot|s,a)$$

- **Policy**

$$a = \pi(s)\,, \quad \text{or} \quad A \sim \pi(\cdot|s)$$

- **Closed-loop performance**

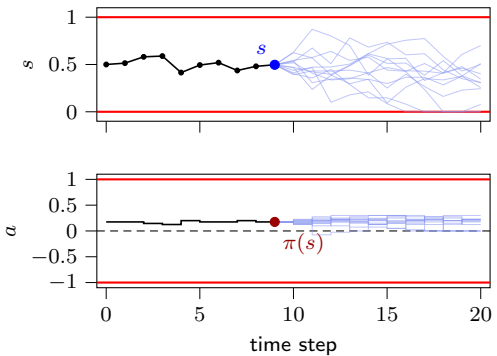
$$J(\pi) = \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k l(s_k, a_k) \,\middle|\, \pi\right]$$

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

- **Optimal policy**: $\pi^\star$ from

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

Cost function (instant performance) $L(s,a) \in \mathbb{R}$



Stochastic state transitions (real world)

$$s^+ \sim P(\cdot|s,a)$$

- **Policy**

$$a = \pi(s)\,, \quad \text{or} \quad A \sim \pi(\cdot|s)$$

- **Closed-loop performance**

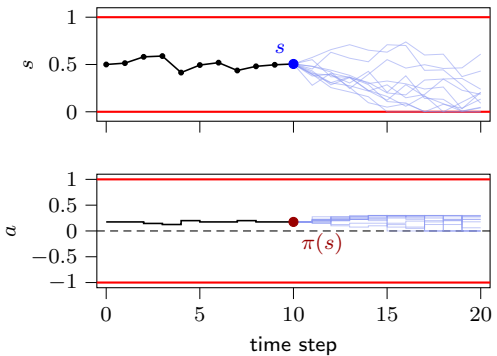
$$J(\pi) = \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k l(s_k, a_k) \,\middle|\, \pi\right]$$

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

- **Optimal policy**: $\pi^\star$ from

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

Cost function (instant performance) $L(s,a) \in \mathbb{R}$



Stochastic state transitions (real world)

$$s^+ \sim P(\cdot|s,a)$$

- **Policy**

$$a = \pi(s)\,, \quad \text{or} \quad A \sim \pi(\cdot|s)$$

- **Closed-loop performance**

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

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

- **Optimal policy**: $\pi^\star$ from

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

Cost function (instant performance) $L(s,a) \in \mathbb{R}$

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

$$l(s,a) = \begin{cases} L(s,a) & \text{if} \quad h(s,a) \geq 0 \\ \infty & \text{if} \quad h(s,a) < 0 \end{cases}$$

Stochastic state transitions (real world)

$$s^+ \sim P(\cdot|s,a)$$

- **Policy**

$$a = \pi(s)\,, \quad \text{or} \quad A \sim \pi(\cdot|s)$$

- **Closed-loop performance**

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

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

- **Optimal policy**: $\pi^\star$ from

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

Cost function (instant performance) $L(s,a) \in \mathbb{R}$

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

$$l(s,a) = \begin{cases} L(s,a) & \text{if} \quad h(s,a) \geq 0 \\ \infty & \text{if} \quad h(s,a) < 0 \end{cases}$$

Solution of an MDP is described by "Bellman" equations (more in a bit), but solving them is very challenging

Stochastic state transitions (real world)

$$s^+ \sim P(\cdot|s,a)$$

- **Policy**

$$a = \pi(s)\,, \quad \text{or} \quad A \sim \pi(\cdot|s)$$

- **Closed-loop performance**

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

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

- **Optimal policy**: $\pi^\star$ from

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

Cost function (instant performance) $L(s,a) \in \mathbb{R}$

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

$$l(s,a) = \begin{cases} L(s,a) & \text{if} \quad h(s,a) \geq 0 \\ \infty & \text{if} \quad h(s,a) < 0 \end{cases}$$

Solution of an MDP is described by "Bellman" equations (more in a bit), but solving them is very challenging

A useful viewpoint: when closed-loop performance (l) matters and real-world is uncertain, MDPs are what we want to solve. MPC is a heuristic to deliver "good enough" solutions, at low computational cost.

MPC as an approximation of the MDP problem

Original MDP problem

$$V^\star(s) = \min_\pi \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k l(S_k, A_k)\right],$$
$$\text{with} \quad S_0 = s,\ S_{k+1} \sim P(\cdot \mid S_k, A_k),\ A_k \sim \pi(\cdot \mid S_k).$$

Steps:

MPC as an approximation of the MDP problem

Approximation 1: finite horizon

$$\min_{\pi} \mathbb{E}\left[\gamma^N \bar{V}^{\pi}(S_N) + \sum_{k=0}^{N-1} \gamma^k l(S_k, A_k)\right],$$
$$\text{with} \quad S_0 = s,\ S_{k+1} \sim P(\cdot \mid S_k, A_k),\ A_k \sim \pi(\cdot \mid S_k).$$

Steps:

1. ∞ horizon $\rightarrow$ horizon N and $\bar{V}$

MPC as an approximation of the MDP problem

Approximation 2: nominal predictions

$$\min_{\pi} \mathbb{E}\left[\gamma^N \bar{V}^{\pi}(S_N) + \sum_{k=0}^{N-1} \gamma^k l(S_k, A_k)\right],$$

$$\text{with} \quad S_0 = s,\ x_0 = s,\ x_{k+1} = f(x_k, u_k),\ A_k \sim \pi(\cdot \mid S_k).$$

Steps:

1. ∞ horizon $\rightarrow$ horizon N and $\bar{V}$
2. stochastic model $\rightarrow$ nominal dynamics

MPC as an approximation of the MDP problem

Approximation 3: open-loop action sequence

$$\min_{u_0,\dots,u_{N-1}} \gamma^N \bar{V}(x_N) + \sum_{k=0}^{N-1} \gamma^k l(x_k, u_k),$$
$$\text{with} \quad x_0 = s,\ x_{k+1} = f(x_k, u_k).$$

Steps:

1. ∞ horizon $\rightarrow$ horizon N and $\bar{V}$
2. stochastic model $\rightarrow$ nominal dynamics
3. policy $\pi \rightarrow$ open-loop sequence

MPC as an approximation of the MDP problem

Unconstrained finite-horizon OCP

$$\min_{u_0,\dots,u_{N-1}} \gamma^N \bar{V}(x_N) + \sum_{k=0}^{N-1} \gamma^k l(x_k, u_k),$$
$$\text{with} \quad x_0 = s,\ x_{k+1} = f(x_k, u_k).$$

Steps:

1. ∞ horizon $\rightarrow$ horizon N and $\bar{V}$
2. stochastic model $\rightarrow$ nominal dynamics
3. policy $\pi \rightarrow$ open-loop sequence

MPC as an approximation of the MDP problem

Deterministic MPC:

$$\begin{aligned} \min_{x,u} \quad & \gamma^N \bar{V}(x_N) + \sum_{k=0}^{N-1} \gamma^k l(x_k, u_k) \\ \text{s.t} \quad & x_{k+1} = f(x_k, u_k) \\ & x_0 = s \end{aligned}$$

Steps:

1. ∞ horizon $\rightarrow$ horizon N and $\bar{V}$
2. stochastic model $\rightarrow$ nominal dynamics
3. policy $\pi \rightarrow$ open-loop sequence

MPC as an approximation of the MDP problem

Deterministic MPC:

$$\begin{aligned} \min_{x,u} \quad & \gamma^N \bar{V}(x_N) + \sum_{k=0}^{N-1} \gamma^k l(x_k, u_k) \\ \text{s.t} \quad & x_{k+1} = f(x_k, u_k) \\ & x_0 = s \end{aligned}$$

Steps:

1. ∞ horizon $\rightarrow$ horizon N and $\bar{V}$
2. stochastic model $\rightarrow$ nominal dynamics
3. policy $\pi \rightarrow$ open-loop sequence

MPC solves this problem online, applies the first action, observes the next state, and repeats.

Deterministic MPC:

$$\begin{aligned} \min_{x,u} \quad & \gamma^N \bar{V}(x_N) + \sum_{k=0}^{N-1} \gamma^k l(x_k, u_k) \\ \text{s.t} \quad & x_{k+1} = f(x_k, u_k) \\ & x_0 = s \end{aligned}$$

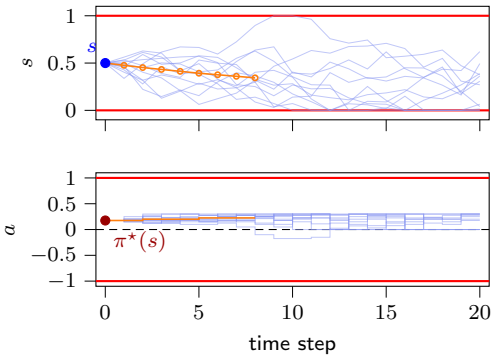
MPC defines a policy:

$$\pi^{\text{MPC}}(s) = u_0^\star$$

How does π^{MPC} relate to $\pi^\star$?

MDP

$$\pi^\star = \arg\min_\pi \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k l(S_k, A_k)\right]$$



Deterministic MPC:

$$\begin{aligned} \min_{x,u} \quad & \gamma^N \bar{V}(x_N) + \sum_{k=0}^{N-1} \gamma^k l(x_k, u_k) \\ \text{s.t} \quad & x_{k+1} = f(x_k, u_k) \\ & x_0 = s \end{aligned}$$

MPC defines a policy:

$$\pi^{\text{MPC}}(s) = u_0^\star$$

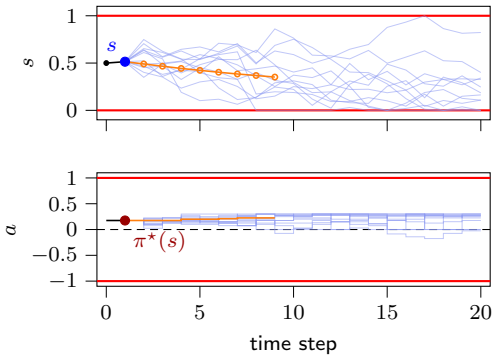
How does π^{MPC} relate to $\pi^\star$?

No reason to match:

- MPC model approximates stochasticity, often deterministic
- Planning $\neq$ policy for stochastic problems

MDP

$$\pi^\star = \arg\min_\pi \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k l(S_k, A_k)\right]$$



MPC as a policy

Deterministic MPC:

$$\begin{aligned} \min_{x,u} \quad & \gamma^N \bar{V}(x_N) + \sum_{k=0}^{N-1} \gamma^k l(x_k, u_k) \\ \text{s.t} \quad & x_{k+1} = f(x_k, u_k) \\ & x_0 = s \end{aligned}$$

MPC defines a policy:

$$\pi^{\text{MPC}}(s) = u_0^\star$$

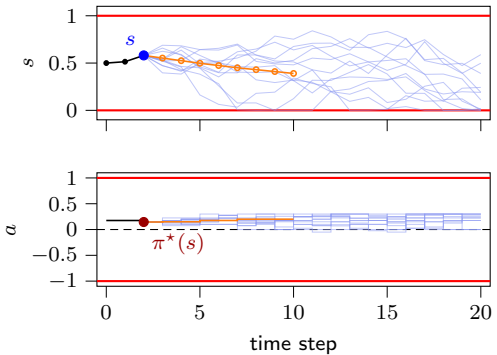
How does π^{MPC} relate to $\pi^\star$?

No reason to match:

- MPC model approximates stochasticity, often deterministic
- Planning $\neq$ policy for stochastic problems

MDP

$$\pi^\star = \arg\min_\pi \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k l(S_k, A_k)\right]$$



MPC as a policy

Deterministic MPC:

$$\begin{aligned} \min_{x,u} \quad & \gamma^N \bar{V}(x_N) + \sum_{k=0}^{N-1} \gamma^k l(x_k, u_k) \\ \text{s.t} \quad & x_{k+1} = f(x_k, u_k) \\ & x_0 = s \end{aligned}$$

MPC defines a policy:

$$\pi^{\text{MPC}}(s) = u_0^\star$$

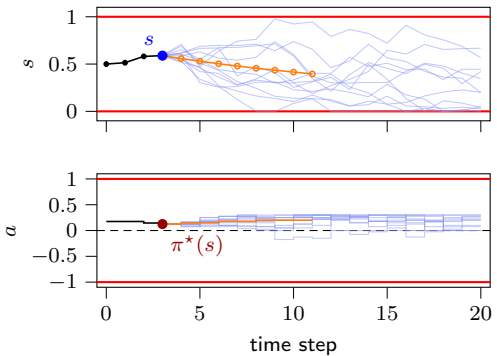
How does π^{MPC} relate to $\pi^\star$?

No reason to match:

- MPC model approximates stochasticity, often deterministic
- Planning $\neq$ policy for stochastic problems

MDP

$$\pi^\star = \arg\min_\pi \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k l(S_k, A_k)\right]$$



MPC as a policy

Deterministic MPC:

$$\begin{aligned} \min_{x,u} \quad & \gamma^N \bar{V}(x_N) + \sum_{k=0}^{N-1} \gamma^k l(x_k, u_k) \\ \text{s.t} \quad & x_{k+1} = f(x_k, u_k) \\ & x_0 = s \end{aligned}$$

MPC defines a policy:

$$\pi^{\text{MPC}}(s) = u_0^\star$$

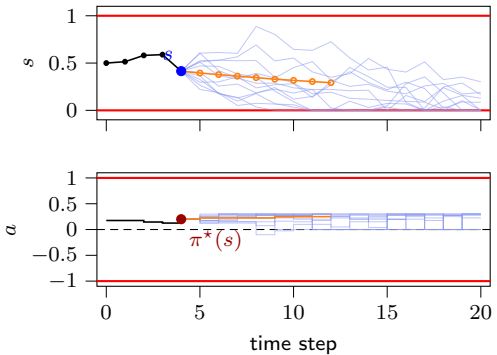
How does π^{MPC} relate to $\pi^\star$?

No reason to match:

- MPC model approximates stochasticity, often deterministic
- Planning $\neq$ policy for stochastic problems

MDP

$$\pi^\star = \arg\min_\pi \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k l(S_k, A_k)\right]$$



Deterministic MPC:

$$\begin{aligned} \min_{x,u} \quad & \gamma^N \bar{V}(x_N) + \sum_{k=0}^{N-1} \gamma^k l(x_k, u_k) \\ \text{s.t} \quad & x_{k+1} = f(x_k, u_k) \\ & x_0 = s \end{aligned}$$

MPC defines a policy:

$$\pi^{\text{MPC}}(s) = u_0^\star$$

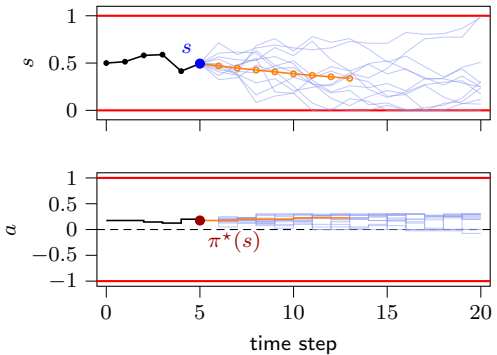
How does π^{MPC} relate to $\pi^\star$?

No reason to match:

- MPC model approximates stochasticity, often deterministic
- Planning $\neq$ policy for stochastic problems

MDP

$$\pi^\star = \arg\min_\pi \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k l(S_k, A_k)\right]$$



Deterministic MPC:

$$\begin{aligned} \min_{x,u} \quad & \gamma^N \bar{V}(x_N) + \sum_{k=0}^{N-1} \gamma^k l(x_k, u_k) \\ \text{s.t} \quad & x_{k+1} = f(x_k, u_k) \\ & x_0 = s \end{aligned}$$

MPC defines a policy:

$$\pi^{\text{MPC}}(s) = u_0^\star$$

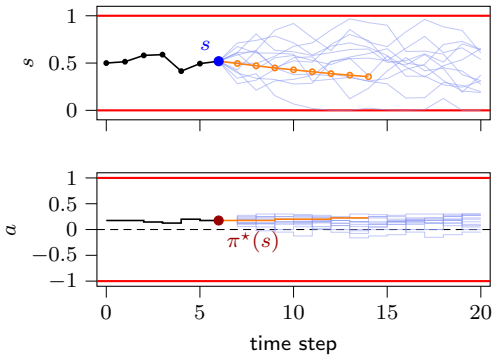
How does π^{MPC} relate to $\pi^\star$?

No reason to match:

- MPC model approximates stochasticity, often deterministic
- Planning $\neq$ policy for stochastic problems

MDP

$$\pi^\star = \arg\min_\pi \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k l(S_k, A_k)\right]$$



MPC as a policy

Deterministic MPC:

$$\begin{aligned} \min_{x,u} \quad & \gamma^N \bar{V}(x_N) + \sum_{k=0}^{N-1} \gamma^k l(x_k, u_k) \\ \text{s.t} \quad & x_{k+1} = f(x_k, u_k) \\ & x_0 = s \end{aligned}$$

MPC defines a policy:

$$\pi^{\text{MPC}}(s) = u_0^\star$$

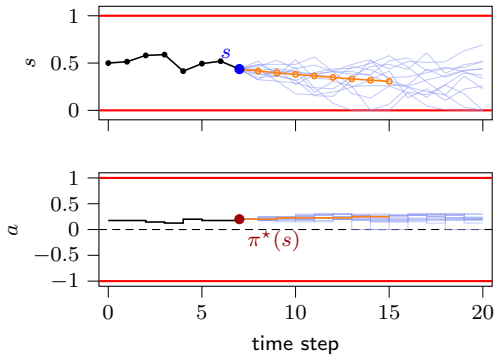
How does π^{MPC} relate to $\pi^\star$?

No reason to match:

- MPC model approximates stochasticity, often deterministic
- Planning $\neq$ policy for stochastic problems

MDP

$$\pi^\star = \arg\min_\pi \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k l(S_k, A_k)\right]$$



Deterministic MPC:

$$\begin{aligned} \min_{x,u} \quad & \gamma^N \bar{V}(x_N) + \sum_{k=0}^{N-1} \gamma^k l(x_k, u_k) \\ \text{s.t} \quad & x_{k+1} = f(x_k, u_k) \\ & x_0 = s \end{aligned}$$

MPC defines a policy:

$$\pi^{\text{MPC}}(s) = u_0^\star$$

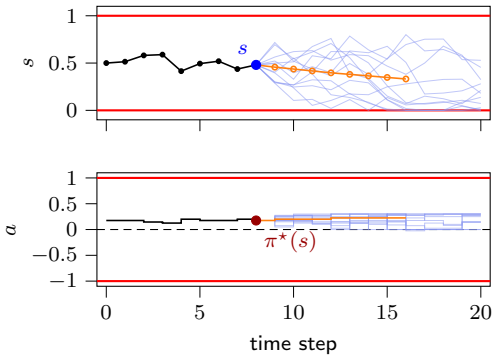
How does π^{MPC} relate to $\pi^\star$?

No reason to match:

- MPC model approximates stochasticity, often deterministic
- Planning $\neq$ policy for stochastic problems

MDP

$$\pi^\star = \arg\min_\pi \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k l(S_k, A_k)\right]$$



Deterministic MPC:

$$\begin{aligned} \min_{x,u} \quad & \gamma^N \bar{V}(x_N) + \sum_{k=0}^{N-1} \gamma^k l(x_k, u_k) \\ \text{s.t} \quad & x_{k+1} = f(x_k, u_k) \\ & x_0 = s \end{aligned}$$

MPC defines a policy:

$$\pi^{\text{MPC}}(s) = u_0^\star$$

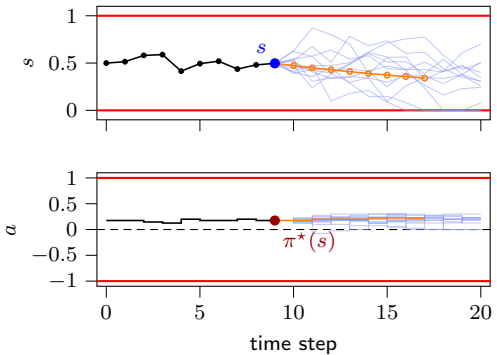
How does π^{MPC} relate to $\pi^\star$?

No reason to match:

- MPC model approximates stochasticity, often deterministic
- Planning $\neq$ policy for stochastic problems

MDP

$$\pi^\star = \arg\min_\pi \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k l(S_k, A_k)\right]$$



Deterministic MPC:

$$\begin{aligned} \min_{x,u} \quad & \gamma^N \bar{V}(x_N) + \sum_{k=0}^{N-1} \gamma^k l(x_k, u_k) \\ \text{s.t} \quad & x_{k+1} = f(x_k, u_k) \\ & x_0 = s \end{aligned}$$

MPC defines a policy:

$$\pi^{\text{MPC}}(s) = u_0^\star$$

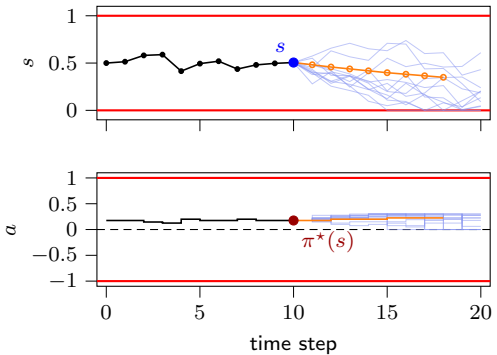
How does π^{MPC} relate to $\pi^\star$?

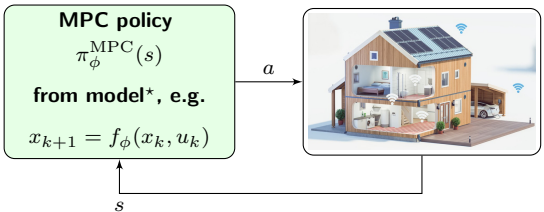
No reason to match:

- MPC model approximates stochasticity, often deterministic
- Planning $\neq$ policy for stochastic problems

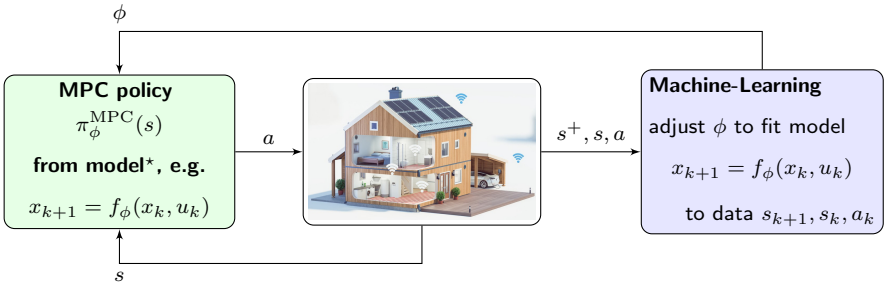
MDP

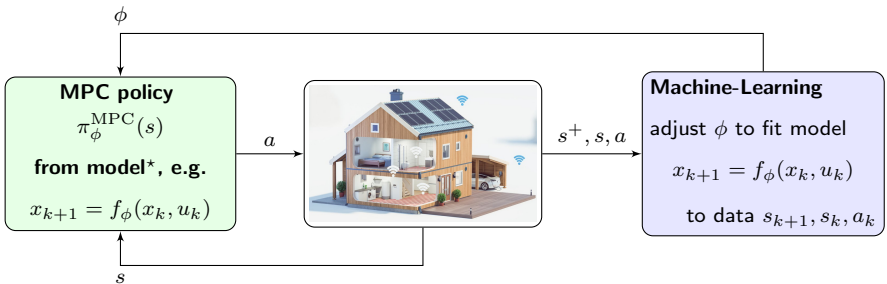
$$\pi^\star = \arg\min_\pi \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k l(S_k, A_k)\right]$$




MPC policy
$\pi_\phi^{\mathrm{MPC}}(s)$
from model*, e.g.
$x_{k+1} = f_\phi(x_k, u_k)$
a
s

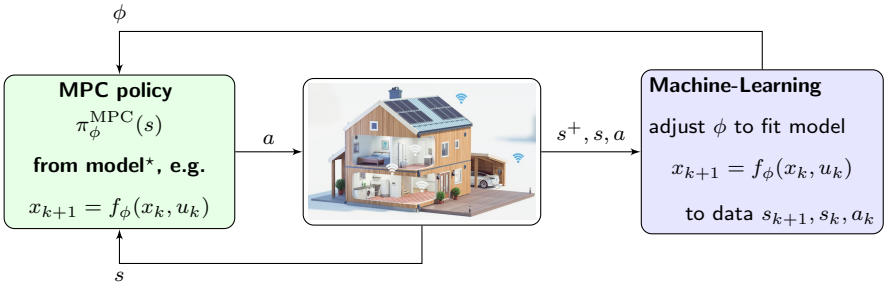
Learning for MPC - Machine Learning in-the-loop





"Machine-Learning" in-the-loop f_ϕ from

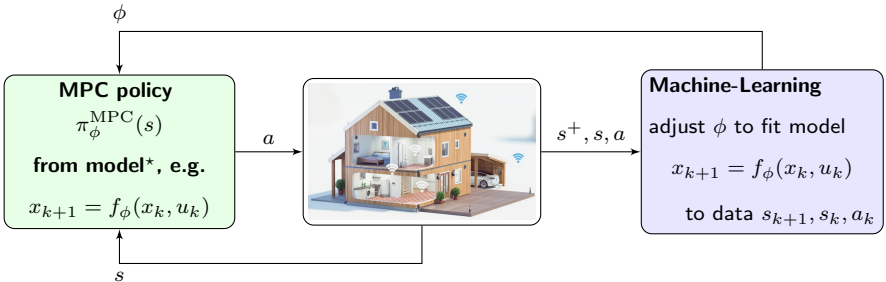
- **Physics-based**: first principles + SYSID
- **Neural Network**: DNN, LSTM, TFT, ...
- **Statistical**: GP, Kernel ...



"Machine-Learning" in-the-loop f_ϕ from

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

** any consistent predictor: extends to input-output, multi-step, Koopman, etc...*



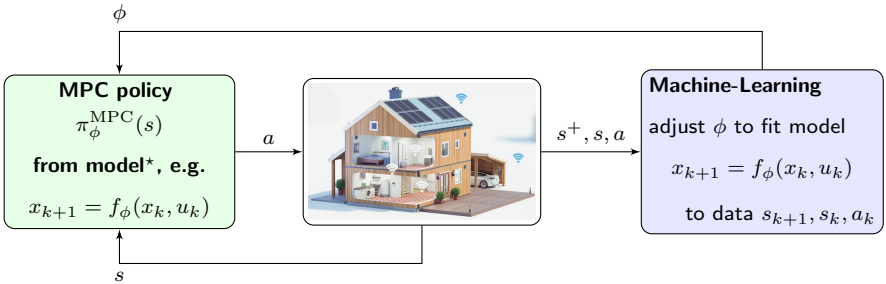
"Machine-Learning" in-the-loop f_ϕ from

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

** any consistent predictor: extends to input-output, multi-step, Koopman, etc...*

Limitations of this paradigm

- Open-loop prediction accuracy $\xrightarrow{\textit{hopefully}}$ Closed-loop performance
- Ignores that MPC is a policy



"Machine-Learning" in-the-loop f_ϕ from

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

RLMPC focuses on "breaking" these limitations
RL plays a key role

** any consistent predictor: extends to input-output, multi-step, Koopman, etc...*

Limitations of this paradigm

- Open-loop prediction accuracy $\xrightarrow{\textit{hopefully}}$ Closed-loop performance
- Ignores that MPC is a policy

Outline

Background: MPC, MDPs, and Model Fitting

MPC as a Model of the MDP Solution

Reinforcement Learning Tunes the MPC

Solving MDPs?

MDP

$$\pi^\star = \arg\min_{\pi} \ \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k l(S_k, A_k)\right]$$

Bellman equations

$$V^\star(s) = \min_a l(s,a) + \gamma\, \mathbb{E}\left[V^\star\left(S^+\right) \mid s, a\right]$$

minimizer a for s gives the optimal policy

MDP

$$\pi^\star = \arg\min_{\pi} \ \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k l(S_k, A_k)\right]$$

Bellman equations

$$V^\star(s) = \min_a l(s,a) + \gamma\, \mathbb{E}\left[V^\star\left(S^+\right) \mid s, a\right]$$

minimizer a for s gives the optimal policy

A more expressive form

$$Q^\star(s,a) = \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k l(S_k, A_k) \,\middle|\, \begin{array}{c} S_0 = s \\ A_0 = a \\ A_{k>0} = \pi^\star(S_k) \end{array}\right]$$

Remarks: the action-value function $Q^\star$ is

- the optimal "cost-to-go" from state s if we force the first action to be a, and follow the optimal policy after that

MDP

$$\pi^\star = \arg\min_\pi \ \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k l(S_k, A_k)\right]$$

Bellman equations

$$V^\star(s) = \min_a l(s,a) + \gamma\, \mathbb{E}\left[V^\star\left(S^+\right) \mid s, a\right]$$

minimizer a for s gives the optimal policy

A more expressive form

$$Q^\star(s,a) = \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k l(S_k, A_k) \,\middle|\, \begin{array}{c} S_0 = s \\ A_0 = a \\ A_{k>0} = \pi^\star(S_k) \end{array}\right]$$

$$Q^\star(s,a) = l(s,a) + \gamma\, \mathbb{E}\left[V^\star\left(S^+\right) | s, a\right]$$

$$V^\star(s) = \min_a \ Q^\star(s,a)$$

$$\pi^\star(s) = \arg\min_a \ Q^\star(s,a)$$

Remarks: the action-value function $Q^\star$ is

- the optimal "cost-to-go" from state s if we force the first action to be a, and follow the optimal policy after that

MDP

$$\pi^\star = \arg\min_\pi \ \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k l(S_k, A_k)\right]$$

Bellman equations

$$V^\star(s) = \min_a l(s,a) + \gamma\, \mathbb{E}\left[V^\star\left(S^+\right) \mid s, a\right]$$

minimizer a for s gives the optimal policy

A more expressive form

$$Q^\star(s,a) = \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k l(S_k, A_k) \,\middle|\, \begin{array}{c} S_0 = s \\ A_0 = a \\ A_{k>0} = \pi^\star(S_k) \end{array}\right]$$

$$Q^\star(s,a) = l(s,a) + \gamma\, \mathbb{E}\left[V^\star\left(S^+\right) | s, a\right]$$

$$V^\star(s) = \min_a \ Q^\star(s,a)$$

$$\pi^\star(s) = \arg\min_a \ Q^\star(s,a)$$

Remarks: the action-value function $Q^\star$ is

- the optimal "cost-to-go" from state s if we force the first action to be a, and follow the optimal policy after that
- the "whole" MDP solution

MDP

$$\pi^\star = \arg\min_\pi \ \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k l(S_k, A_k)\right]$$

Bellman equations

$$V^\star(s) = \min_a l(s,a) + \gamma\, \mathbb{E}\left[V^\star\left(S^+\right) \mid s, a\right]$$

minimizer a for s gives the optimal policy

A more expressive form

$$Q^\star(s,a) = \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k l(S_k, A_k) \,\middle|\, \begin{array}{c} S_0 = s \\ A_0 = a \\ A_{k>0} = \pi^\star(S_k) \end{array}\right]$$

$$Q^\star(s,a) = l(s,a) + \gamma\, \mathbb{E}\left[V^\star\left(S^+\right) | s, a\right]$$

$$V^\star(s) = \min_a \ Q^\star(s,a)$$

$$\pi^\star(s) = \arg\min_a \ Q^\star(s,a)$$

Remarks: the action-value function $Q^\star$ is

- the optimal "cost-to-go" from state s if we force the first action to be a, and follow the optimal policy after that
- the "whole" MDP solution
- very expensive to compute if s, a are not "trivially small-dimensional"

MDP

$$\pi^\star = \arg\min_\pi \ \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k l(S_k, A_k)\right]$$

Bellman equations

$$V^\star(s) = \min_a l(s,a) + \gamma\, \mathbb{E}\left[V^\star\left(S^+\right) \mid s, a\right]$$

minimizer a for s gives the optimal policy

A more expressive form

$$Q^\star(s,a) = \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k l(S_k, A_k) \,\middle|\, \begin{array}{c} S_0 = s \\ A_0 = a \\ A_{k>0} = \pi^\star(S_k) \end{array}\right]$$

$$\begin{aligned} Q^\star(s,a) &= l(s,a) + \gamma\, \mathbb{E}\left[V^\star\left(S^+\right) | s, a\right] \\ V^\star(s) &= \min_a \ Q^\star(s,a) \\ \pi^\star(s) &= \arg\min_a \ Q^\star(s,a) \end{aligned}$$

Remarks: the action-value function $Q^\star$ is

- the optimal "cost-to-go" from state s if we force the first action to be a, and follow the optimal policy after that
- the "whole" MDP solution
- very expensive to compute if s, a are not "trivially small-dimensional"
- a "cornerstone" in Reinforcement Learning

MPC as a model of the MDP solution?

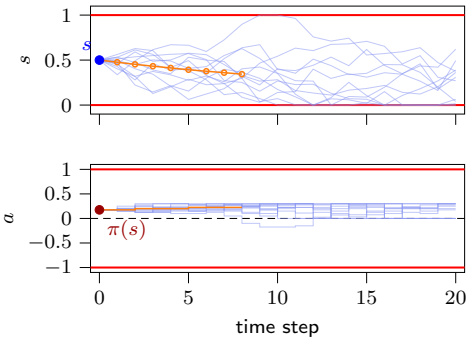
MDP optimal policy

$$\pi^\star = \arg\min_\pi \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k l(S_k, A_k)\right]$$

is "solved" by $Q^\star(s,a)$

MPC policy $\pi^{\mathrm{MPC}}(s) = u_0^\star$ from

$$\begin{aligned} \min_{x,u} \quad & \bar{V}(x_N) + \sum_{k=0}^{N-1} \gamma^k l(x_k, u_k) \\ \text{s.t} \quad & x_{k+1} = f(x_k, u_k), \quad x_0 = s \\ & h(x_k, u_k) \geq 0 \end{aligned}$$



MPC as a model of the MDP solution?

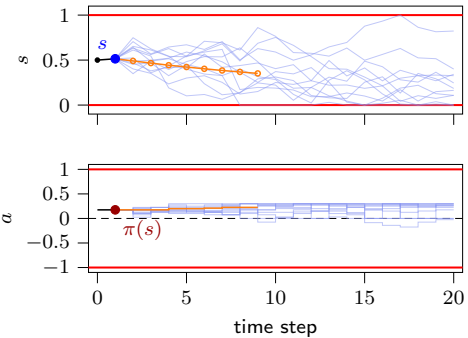
MDP optimal policy

$$\pi^\star = \arg\min_\pi \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k l(S_k, A_k)\right]$$

is "solved" by $Q^\star(s,a)$

MPC policy $\pi^{\mathrm{MPC}}(s) = u_0^\star$ from

$$\begin{aligned} \min_{x,u} \quad & \bar{V}(x_N) + \sum_{k=0}^{N-1} \gamma^k l(x_k, u_k) \\ \text{s.t} \quad & x_{k+1} = f(x_k, u_k), \quad x_0 = s \\ & h(x_k, u_k) \geq 0 \end{aligned}$$



- Solving MDP $\equiv$ build $Q^\star$ model
- MPC can model a value function...
- Can it model an action-value function?

MPC as a model of the MDP solution?

MDP optimal policy

$$\pi^\star = \arg\min_\pi \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k l(S_k, A_k)\right]$$

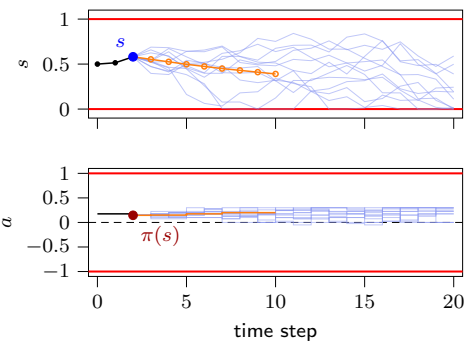
is "solved" by $Q^\star(s,a)$

$$V^{\mathrm{MPC}}(s) := \min_{x,u} \bar{V}(x_N) + \sum_{k=0}^{N-1} \gamma^k l(x_k, u_k)$$
$$\text{s.t} \quad x_{k+1} = f(x_k, u_k), \quad x_0 = s$$
$$h(x_k, u_k) \geq 0$$

- Solving MDP $\equiv$ build $Q^\star$ model
- MPC can model a value function...
- Can it model an action-value function?

MPC policy $\pi^{\mathrm{MPC}}(s) = u_0^\star$ from

$$\min_{x,u} \quad \bar{V}(x_N) + \sum_{k=0}^{N-1} \gamma^k l(x_k, u_k)$$
$$\text{s.t} \quad x_{k+1} = f(x_k, u_k), \quad x_0 = s$$
$$h(x_k, u_k) \geq 0$$



MPC as a model of the MDP solution?

MDP optimal policy

$$\pi^\star = \arg\min_{\pi} \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k l(S_k, A_k)\right]$$

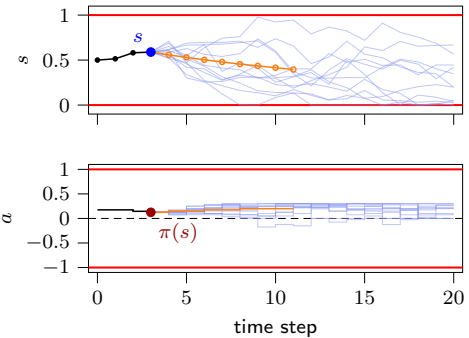
is "solved" by $Q^\star(s,a)$

$$V^{\mathrm{MPC}}(s) := \min_{x,u} \bar{V}(x_N) + \sum_{k=0}^{N-1} \gamma^k l(x_k, u_k)$$
$$\text{s.t} \quad x_{k+1} = f(x_k, u_k), \quad x_0 = s$$
$$h(x_k, u_k) \geq 0$$

$$Q^{\mathrm{MPC}}(s,a) := \min_{x,u} \bar{V}(x_N) + \sum_{k=0}^{N-1} \gamma^k l(x_k, u_k)$$
$$\text{s.t} \quad x_{k+1} = f(x_k, u_k)$$
$$h(x_k, u_k) \geq 0$$
$$x_0 = s, \quad u_0 = a$$

MPC policy $\pi^{\mathrm{MPC}}(s) = u_0^\star$ from

$$\min_{x,u} \quad \bar{V}(x_N) + \sum_{k=0}^{N-1} \gamma^k l(x_k, u_k)$$
$$\text{s.t} \quad x_{k+1} = f(x_k, u_k), \quad x_0 = s$$
$$h(x_k, u_k) \geq 0$$



MPC as a model of the MDP solution?

MDP optimal policy

$$\pi^\star = \arg\min_\pi \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k l(S_k, A_k)\right]$$

is "solved" by $Q^\star(s,a)$

$$V^{\mathrm{MPC}}(s) := \min_{x,u} \bar{V}(x_N) + \sum_{k=0}^{N-1} \gamma^k l(x_k, u_k)$$
$$\text{s.t} \quad x_{k+1} = f(x_k, u_k), \quad x_0 = s$$
$$h(x_k, u_k) \geq 0$$

$$Q^{\mathrm{MPC}}(s,a) := \min_{x,u} \bar{V}(x_N) + \sum_{k=0}^{N-1} \gamma^k l(x_k, u_k)$$
$$\text{s.t} \quad x_{k+1} = f(x_k, u_k)$$
$$h(x_k, u_k) \geq 0$$
$$x_0 = s, \quad u_0 = a$$

MPC policy $\pi^{\mathrm{MPC}}(s) = u_0^\star$ from

$$\min_{x,u} \quad \bar{V}(x_N) + \sum_{k=0}^{N-1} \gamma^k l(x_k, u_k)$$
$$\text{s.t} \quad x_{k+1} = f(x_k, u_k), \quad x_0 = s$$
$$h(x_k, u_k) \geq 0$$

MPC is consistent:

$$V^{\mathrm{MPC}}(s) = \min_a Q^{\mathrm{MPC}}(s,a)$$
$$\pi^{\mathrm{MPC}}(s) = \arg\min_a Q^{\mathrm{MPC}}(s,a)$$

MPC as a model of the MDP solution?

MDP optimal policy

$$\pi^\star = \arg\min_\pi \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k l(S_k, A_k)\right]$$

is "solved" by $Q^\star(s,a)$

$$V^{\mathrm{MPC}}(s) := \min_{x,u} \bar{V}(x_N) + \sum_{k=0}^{N-1} \gamma^k l(x_k, u_k)$$
$$\text{s.t} \quad x_{k+1} = f(x_k, u_k), \quad x_0 = s$$
$$h(x_k, u_k) \geq 0$$

$$Q^{\mathrm{MPC}}(s,a) := \min_{x,u} \bar{V}(x_N) + \sum_{k=0}^{N-1} \gamma^k l(x_k, u_k)$$
$$\text{s.t} \quad x_{k+1} = f(x_k, u_k)$$
$$h(x_k, u_k) \geq 0$$
$$x_0 = s, \quad u_0 = a$$

MPC policy $\pi^{\mathrm{MPC}}(s) = u_0^\star$ from

$$\min_{x,u} \quad \bar{V}(x_N) + \sum_{k=0}^{N-1} \gamma^k l(x_k, u_k)$$
$$\text{s.t} \quad x_{k+1} = f(x_k, u_k), \quad x_0 = s$$
$$h(x_k, u_k) \geq 0$$

MPC is consistent:

$$V^{\mathrm{MPC}}(s) = \min_a Q^{\mathrm{MPC}}(s,a)$$
$$\pi^{\mathrm{MPC}}(s) = \arg\min_a Q^{\mathrm{MPC}}(s,a)$$

MPC is a complete MDP model if:

$$Q^{\mathrm{MPC}}(s,a) = Q^\star(s,a)$$

for all s, a. Then **optimality** holds:

$$\pi^{\mathrm{MPC}}(s) = \pi^\star(s)$$

MPC as a model of the MDP solution?

MDP optimal policy

$$\pi^\star = \arg\min_\pi \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k l(S_k, A_k)\right]$$

is "solved" by $Q^\star(s,a)$

$$V^{\mathrm{MPC}}(s) := \min_{x,u} \bar{V}(x_N) + \sum_{k=0}^{N-1} \gamma^k l(x_k, u_k)$$
$$\text{s.t} \quad x_{k+1} = f(x_k, u_k), \quad x_0 = s$$
$$h(x_k, u_k) \geq 0$$

$$Q^{\mathrm{MPC}}(s,a) := \min_{x,u} \bar{V}(x_N) + \sum_{k=0}^{N-1} \gamma^k l(x_k, u_k)$$
$$\text{s.t} \quad x_{k+1} = f(x_k, u_k)$$
$$h(x_k, u_k) \geq 0$$
$$x_0 = s, \quad u_0 = a$$

MPC policy $\pi^{\mathrm{MPC}}(s) = u_0^\star$ from

$$\min_{x,u} \quad \bar{V}(x_N) + \sum_{k=0}^{N-1} \gamma^k l(x_k, u_k)$$
$$\text{s.t} \quad x_{k+1} = f(x_k, u_k), \quad x_0 = s$$
$$h(x_k, u_k) \geq 0$$

Punchline: see MPC as a $Q^\star$ model i.e. what we want to achieve is

$$Q^{\mathrm{MPC}}(s,a) \approx Q^\star(s,a)$$

or at least

$$\pi^{\mathrm{MPC}}(s) \approx \pi^\star(s)$$

$\neq$ fitting f to the real world

How to model $Q^\star$ with MPC?

How to model $Q^\star$ with MPC?

Classical view...

MPC: at current state s solve

$$\begin{aligned} \min_{x,u} \quad & \bar{V}(x_N) + \sum_{k=0}^{N-1} l(x_k, u_k) \\ \text{s.t.} \quad & x_{k+1} = f_\phi(x_k, u_k) \\ & h(x_k, u_k) \geq 0 \\ & x_0 = s \end{aligned}$$

gives policy $\pi_\phi^{\text{MPC}}(s) = u_0^\star$

Find ϕ such that prediction "fits" the data

Shift 1: focus on performance instead of fitting

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

Classical view...

MPC: at current state s solve

$$\begin{aligned} \min_{x,u} \quad & \bar{V}(x_N) + \sum_{k=0}^{N-1} l(x_k, u_k) \\ \text{s.t.} \quad & x_{k+1} = f_\phi(x_k, u_k) \\ & h(x_k, u_k) \geq 0 \\ & x_0 = s \end{aligned}$$

gives policy $\pi_\phi^{\text{MPC}}(s) = u_0^\star$

Find ϕ such that prediction "fits" the data

How to model $Q^\star$ with MPC?

Shift 1: focus on performance instead of fitting

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

Classical view...

MPC: at current state s solve

$$\begin{aligned} \min_{x,u} \quad & \bar{V}(x_N) + \sum_{k=0}^{N-1} l(x_k, u_k) \\ \text{s.t.} \quad & x_{k+1} = f_\phi(x_k, u_k) \\ & h(x_k, u_k) \geq 0 \\ & x_0 = s \end{aligned}$$

gives policy $\pi_\phi^{\text{MPC}}(s) = u_0^\star$

Find ϕ such that prediction "fits" the data

Shift to...

Find ϕ that "fits MPC to optimality" according to the data

($Q^{\text{MPC}} \to Q^\star$ or at least $\pi^{\text{MPC}} \to \pi^\star$)

How to model $Q^\star$ with MPC?

Shift 1: focus on performance instead of fitting

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

Classical view...

MPC: at current state s solve

$$\begin{aligned} \min_{x,u} \quad & \bar{V}(x_N) + \sum_{k=0}^{N-1} l(x_k, u_k) \\ \text{s.t.} \quad & x_{k+1} = f_\phi(x_k, u_k) \\ & h(x_k, u_k) \geq 0 \\ & x_0 = s \end{aligned}$$

gives policy $\pi_\phi^{\text{MPC}}(s) = u_0^\star$

Find ϕ such that prediction "fits" the data

Shift to...

Find ϕ that "fits MPC to optimality" according to the data

($Q^{\text{MPC}} \to Q^\star$ or at least $\pi^{\text{MPC}} \to \pi^\star$)

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

* Theorem 1 in Anand et al., "Optimality conditions for model predictive control: Rethinking predictive model design." Automatica (2026).

Shift 1: focus on performance instead of fitting

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

Classical view...

MPC: at current state s solve

$$\begin{aligned} \min_{x,u} \quad & \bar{V}(x_N) + \sum_{k=0}^{N-1} l(x_k, u_k) \\ \text{s.t.} \quad & x_{k+1} = f_\phi(x_k, u_k) \\ & h(x_k, u_k) \geq 0 \\ & x_0 = s \end{aligned}$$

gives policy $\pi_\phi^{\text{MPC}}(s) = u_0^\star$

Find ϕ such that prediction "fits" the data

Shift to...

Find ϕ that "fits MPC to optimality" according to the data

($Q^{\text{MPC}} \to Q^\star$ or at least $\pi^{\text{MPC}} \to \pi^\star$)

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

But this places "high demands" on f_ϕ

Can we do better?

* Theorem 1 in Anand et al., "Optimality conditions for model predictive control: Rethinking predictive model design." Automatica (2026).

Shift 1: focus on performance instead of fitting

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

Classical view...

MPC: at current state s solve

$$\begin{aligned} \min_{x,u} \quad & \bar{V}(x_N) + \sum_{k=0}^{N-1} l(x_k, u_k) \\ \text{s.t.} \quad & x_{k+1} = f_\phi(x_k, u_k) \\ & h(x_k, u_k) \geq 0 \\ & x_0 = s \end{aligned}$$

gives policy $\pi_\phi^{\text{MPC}}(s) = u_0^\star$

Find ϕ such that prediction "fits" the data

Shift to...

Find ϕ that "fits MPC to optimality" according to the data

($Q^{\text{MPC}} \to Q^\star$ or at least $\pi^{\text{MPC}} \to \pi^\star$)

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

But this places "high demands" on f_ϕ

Can we do better? **Yes**!

* Theorem 1 in Anand et al., "Optimality conditions for model predictive control: Rethinking predictive model design." Automatica (2026).

Shift 1: focus on performance instead of fitting

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

Classical view...

MPC: at current state s solve

$$\begin{aligned} \min_{x,u} \quad & \bar{V}\left(x_N\right) + \sum_{k=0}^{N-1} l\left(x_k, u_k\right) \\ \text{s.t.} \quad & x_{k+1} = f_\phi\left(x_k, u_k\right) \\ & h\left(x_k, u_k\right) \geq 0 \\ & x_0 = s \end{aligned}$$

gives policy $\pi_\phi^{\mathrm{MPC}}\left(s\right) = u_0^\star$

Find ϕ such that prediction "fits" the data

Shift to...

Find ϕ that "fits MPC to optimality" according to the data

($Q^{\mathrm{MPC}} \to Q^\star$ or at least $\pi^{\mathrm{MPC}} \to \pi^\star$)

Shift 2: "holistic" parametrization

$$\begin{aligned} \min_{x,u} \quad & \bar{V}_\phi\left(x_N\right) + \sum_{k=0}^{N-1} l_\phi\left(x_k, u_k\right) \\ \text{s.t.} \quad & x_{k+1} = f_\phi\left(x_k, u_k\right), \quad x_0 = s \\ & h_\phi\left(x_k, u_k\right) \geq 0 \end{aligned}$$

real-world performance still assessed via l and h

Shift 1: focus on performance instead of fitting

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

Classical view...

MPC: at current state s solve

$$\begin{aligned} \min_{x,u} \quad & \bar{V}(x_N) + \sum_{k=0}^{N-1} l(x_k, u_k) \\ \text{s.t.} \quad & x_{k+1} = f_\phi(x_k, u_k) \\ & h(x_k, u_k) \geq 0 \\ & x_0 = s \end{aligned}$$

gives policy $\pi_\phi^{\text{MPC}}(s) = u_0^\star$

Find ϕ such that prediction "fits" the data

Shift to...

Find ϕ that "fits MPC to optimality" according to the data

($Q^{\text{MPC}} \to Q^\star$ or at least $\pi^{\text{MPC}} \to \pi^\star$)

Shift 2: "holistic" parametrization

$$\begin{aligned} \min_{x,u} \quad & \bar{V}_\phi(x_N) + \sum_{k=0}^{N-1} l_\phi(x_k, u_k) \\ \text{s.t.} \quad & x_{k+1} = f_\phi(x_k, u_k), \quad x_0 = s \\ & h_\phi(x_k, u_k) \geq 0 \end{aligned}$$

real-world performance still assessed via l and h

Intuitively ok... but does that make sense?

Full MPC parametrization:

$$\begin{aligned} \min_{x,u} \quad & \bar{V}_\phi(x_N) + \sum_{k=0}^{N-1} l_\phi(x_k, u_k) \\ \text{s.t.} \quad & x_{k+1} = f_\phi(x_k, u_k) \\ & h_\phi(x_k, u_k) \geq 0 \\ & x_0 = s \end{aligned}$$

gives policy $\pi_\phi^{\text{MPC}}(s) = u_0^\star$

Full MPC parametrization:

$$\begin{aligned} \min_{x,u} \quad & \bar{V}_\phi\left(x_N\right) + \sum_{k=0}^{N-1} l_\phi\left(x_k, u_k\right) \\ \text{s.t.} \quad & x_{k+1} = f_\phi\left(x_k, u_k\right) \\ & h_\phi\left(x_k, u_k\right) \geq 0 \\ & x_0 = s \end{aligned}$$

gives policy $\pi_\phi^{\mathrm{MPC}}\left(s\right) = u_0^\star$

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

$$Q_\phi^{\mathrm{MPC}}\left(s,a\right) = Q^\star\left(s,a\right)$$
$$\pi_\phi^{\mathrm{MPC}}\left(s\right) = \pi^\star\left(s\right)$$

even if the model f_ϕ *cannot* describe the real system accurately

* Theorem 1 from Gros et al., "Data-driven economic NMPC using reinforcement learning," IEEE Transactions on Automatic Control 65.2 (2019): 636–648.

Full MPC parametrization:

$$\begin{aligned} \min_{x,u} \quad & \bar{V}_\phi\left(x_N\right) + \sum_{k=0}^{N-1} l_\phi\left(x_k, u_k\right) \\ \text{s.t.} \quad & x_{k+1} = f_\phi\left(x_k, u_k\right) \\ & h_\phi\left(x_k, u_k\right) \geq 0 \\ & x_0 = s \end{aligned}$$

gives policy $\pi_\phi^{\mathrm{MPC}}\left(s\right) = u_0^\star$

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

$$Q_\phi^{\mathrm{MPC}}\left(s,a\right) = Q^\star\left(s,a\right)$$
$$\pi_\phi^{\mathrm{MPC}}\left(s\right) = \pi^\star\left(s\right)$$

even if the model f_ϕ *cannot* describe the real system accurately

Remarks

- Generic: works for robust MPC, stochastic MPC, economic MPC, Multi-Step PC, etc.

* Theorem 1 from Gros et al., "Data-driven economic NMPC using reinforcement learning," IEEE Transactions on Automatic Control 65.2 (2019): 636–648.

Full MPC parametrization:

$$\begin{aligned} \min_{x,u} \quad & \bar{V}_\phi\left(x_N\right) + \sum_{k=0}^{N-1} l_\phi\left(x_k, u_k\right) \\ \text{s.t.} \quad & x_{k+1} = f_\phi\left(x_k, u_k\right) \\ & h_\phi\left(x_k, u_k\right) \geq 0 \\ & x_0 = s \end{aligned}$$

gives policy $\pi_\phi^{\mathrm{MPC}}\left(s\right) = u_0^\star$

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

$$Q_\phi^{\mathrm{MPC}}\left(s,a\right) = Q^\star\left(s,a\right)$$
$$\pi_\phi^{\mathrm{MPC}}\left(s\right) = \pi^\star\left(s\right)$$

even if the model f_ϕ *cannot* describe the real system accurately

Remarks

- Generic: works for robust MPC, stochastic MPC, economic MPC, Multi-Step PC, etc.
- Sanity check: technical conditions are mild but forbid f_ϕ to lead to non-finite $V^\star$ under the resulting closed-loop policy

* Theorem 1 from Gros et al., "Data-driven economic NMPC using reinforcement learning," IEEE Transactions on Automatic Control 65.2 (2019): 636–648.

Full MPC parametrization:

$$\begin{aligned} \min_{x,u} \quad & \bar{V}_\phi\left(x_N\right) + \sum_{k=0}^{N-1} l_\phi\left(x_k, u_k\right) \\ \text{s.t.} \quad & x_{k+1} = f_\phi\left(x_k, u_k\right) \\ & h_\phi\left(x_k, u_k\right) \geq 0 \\ & x_0 = s \end{aligned}$$

gives policy $\pi_\phi^{\mathrm{MPC}}\left(s\right) = u_0^\star$

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

$$Q_\phi^{\mathrm{MPC}}\left(s,a\right) = Q^\star\left(s,a\right)$$
$$\pi_\phi^{\mathrm{MPC}}\left(s\right) = \pi^\star\left(s\right)$$

even if the model f_ϕ *cannot* describe the real system accurately

Remarks

- Generic: works for robust MPC, stochastic MPC, economic MPC, Multi-Step PC, etc.
- Sanity check: technical conditions are mild but forbid f_ϕ to lead to non-finite $V^\star$ under the resulting closed-loop policy

* Theorem 1 from Gros et al., "Data-driven economic NMPC using reinforcement learning," IEEE Transactions on Automatic Control 65.2 (2019): 636–648.

Full MPC parametrization:

$$\begin{aligned} \min_{x,u} \quad & \bar{V}_\phi\left(x_N\right) + \sum_{k=0}^{N-1} l_\phi\left(x_k, u_k\right) \\ \text{s.t.} \quad & x_{k+1} = f_\phi\left(x_k, u_k\right) \\ & h_\phi\left(x_k, u_k\right) \geq 0 \\ & x_0 = s \end{aligned}$$

gives policy $\pi_\phi^{\mathrm{MPC}}\left(s\right) = u_0^\star$

Full MPC parametrization enables modeling $Q^\star$, $\pi^\star$ without "torturing" f_ϕ

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

$$Q_\phi^{\mathrm{MPC}}\left(s,a\right) = Q^\star\left(s,a\right)$$
$$\pi_\phi^{\mathrm{MPC}}\left(s\right) = \pi^\star\left(s\right)$$

even if the model f_ϕ *cannot* describe the real system accurately

Remarks

- Generic: works for robust MPC, stochastic MPC, economic MPC, Multi-Step PC, etc.
- Sanity check: technical conditions are mild but forbid f_ϕ to lead to non-finite $V^\star$ under the resulting closed-loop policy

* Theorem 1 from Gros et al., "Data-driven economic NMPC using reinforcement learning," IEEE Transactions on Automatic Control 65.2 (2019): 636–648.

Outline

Policy $\pi_\phi^{\mathrm{MPC}}(s) = u_0^\star$ from

$$\begin{aligned} \min_{x,u} \quad & \bar{V}_\phi(x_N) + \sum_{k=0}^{N-1} l_\phi(x_k, u_k) \\ \text{s.t.} \quad & x_{k+1} = f_\phi(x_k, u_k) \\ & h_\phi(x_k, u_k) \geq 0, \quad x_0 = s \end{aligned}$$

We want

$$\min_\phi J\left(\pi_\phi^{\mathrm{MPC}}\right)$$

where $J(.)$ is the closed-loop cost in the real-world

What to do?

- $\min_\phi J\left(\pi_\phi^{\mathrm{MPC}}\right)$ using data
- Complex "chain": $\phi \rightarrow \pi_\phi^{\mathrm{MPC}}(s) \rightarrow$ real world $\rightarrow J\left(\pi_\phi^{\mathrm{MPC}}\right)$

Policy $\pi_\phi^{\text{MPC}}(s) = u_0^\star$ from

$$\begin{aligned} \min_{x,u} \quad & \bar{V}_\phi(x_N) + \sum_{k=0}^{N-1} l_\phi(x_k, u_k) \\ \text{s.t.} \quad & x_{k+1} = f_\phi(x_k, u_k) \\ & h_\phi(x_k, u_k) \geq 0, \quad x_0 = s \end{aligned}$$

We want

$$\min_\phi J\left(\pi_\phi^{\text{MPC}}\right)$$

where $J(.)$ is the closed-loop cost in the real-world

What to do?

- $\min_\phi J\left(\pi_\phi^{\text{MPC}}\right)$ using data
- Complex "chain": $\phi \to \pi_\phi^{\text{MPC}}(s) \to$ real world $\to J\left(\pi_\phi^{\text{MPC}}\right)$

Reinforcement Learning

Toolbox to solve MDPs from data
This is not (necessarily) about DNNs

Policy $\pi_\phi^{\mathrm{MPC}}(s) = u_0^\star$ from

$$\begin{aligned} \min_{x,u} \quad & \bar{V}_\phi(x_N) + \sum_{k=0}^{N-1} l_\phi(x_k, u_k) \\ \text{s.t.} \quad & x_{k+1} = f_\phi(x_k, u_k) \\ & h_\phi(x_k, u_k) \geq 0, \quad x_0 = s \end{aligned}$$

We want

$$\min_\phi J\left(\pi_\phi^{\mathrm{MPC}}\right)$$

where $J(.)$ is the closed-loop cost in the real-world

What to do?

- $\min_\phi J\left(\pi_\phi^{\mathrm{MPC}}\right)$ using data
- Complex "chain": $\phi \to \pi_\phi^{\mathrm{MPC}}(s) \to$ real world $\to J\left(\pi_\phi^{\mathrm{MPC}}\right)$

Reinforcement Learning

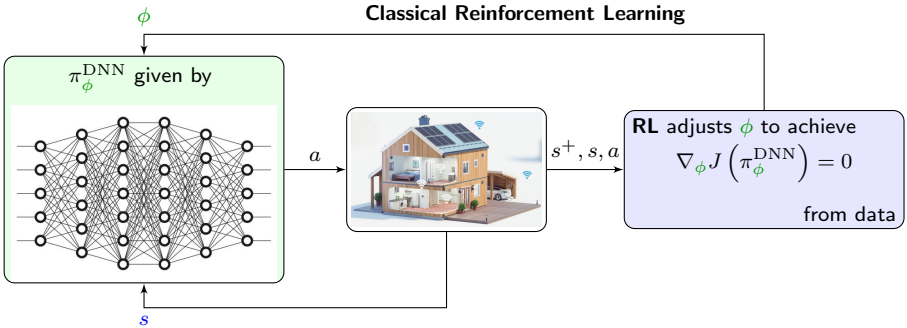
Toolbox to solve MDPs from data
This is not (necessarily) about DNNs

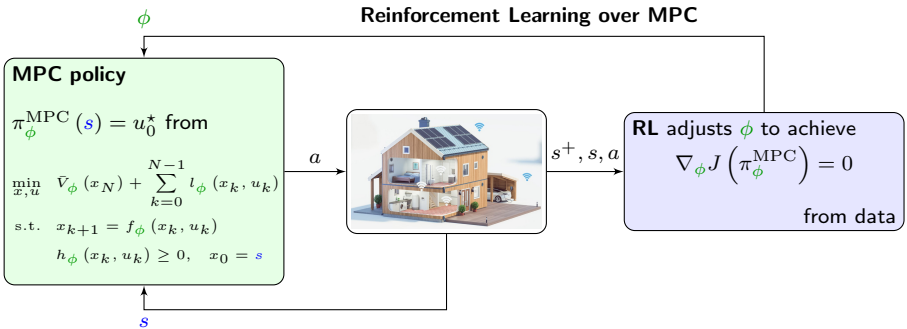
For MPC: tools to find best ϕ, e.g.

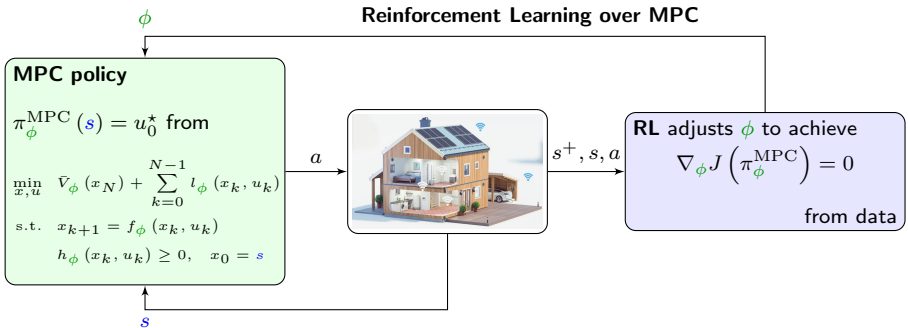
- **Policy Gradient**: estimations of

$$\nabla_\phi J\left(\pi_\phi^{\mathrm{MPC}}\right), \quad \text{possibly} \quad \nabla_\phi^2 J\left(\pi_\phi^{\mathrm{MPC}}\right)$$

- **Q-learning**: directly seek $Q^{\mathrm{MPC}} \to Q^\star$

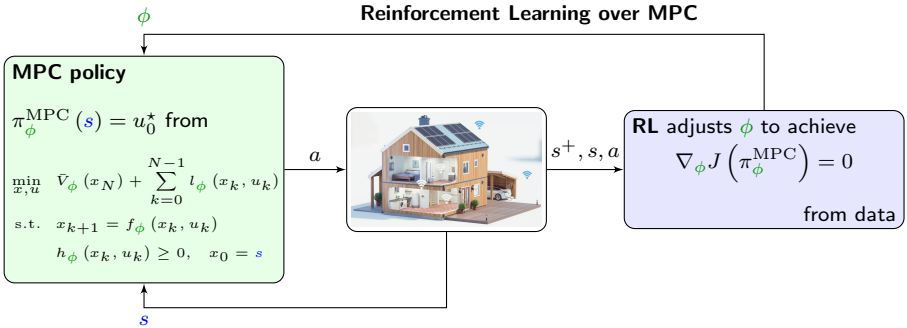

Classical Reinforcement Learning
ϕ
π_ϕ^{DNN} given by
a
s^+, s, a
RL adjusts ϕ to achieve
$\nabla_\phi J\left(\pi_\phi^{\mathrm{DNN}}\right) = 0$
from data
s


ϕ
Reinforcement Learning over MPC
MPC policy
$\pi_\phi^{\mathrm{MPC}}(s) = u_0^\star$ from
$\min_{x,u} \; \bar{V}_\phi(x_N) + \sum_{k=0}^{N-1} l_\phi(x_k, u_k)$
s.t. $x_{k+1} = f_\phi(x_k, u_k)$
$h_\phi(x_k, u_k) \geq 0, \quad x_0 = s$
a
s^+, s, a
RL adjusts ϕ to achieve
$\nabla_\phi J\left(\pi_\phi^{\mathrm{MPC}}\right) = 0$
from data
s



Why is it useful?

- ✓ Tune MPC for real-world performance
- ✓ 25 years of guarantees & methods
- ✓ Learning does not start from scratch
- ✓ Well accepted in the industry
- ✓ Repeated Planning provides explainability
- ✓ Great at processing forecast information



Why is it useful?

- ✓ Tune MPC for real-world performance
- ✓ 25 years of guarantees & methods
- ✓ Learning does not start from scratch
- ✓ Well accepted in the industry
- ✓ Repeated Planning provides explainability
- ✓ Great at processing forecast information

Why can it be difficult?

- ✘ MPC is computationally more expensive than a DNN
- ✘ Deployment on GPUs is lagging
- ✘ Non-convexity can get in the way...
- ✘ RL methods are tailored to DNN, not MPC. There is still a gap here.

Proposed paradigm

Policy $\pi_{\phi}^{\mathrm{MPC}}(s) = u_0^{\star}$ from

$$\begin{aligned} \min_{x,u} \quad & \bar{V}_{\phi}(x_N) + \sum_{k=0}^{N-1} l_{\phi}(x_k, u_k) \\ \text{s.t.} \quad & x_{k+1} = f_{\phi}(x_k, u_k) \\ & h_{\phi}(x_k, u_k) \geq 0, \quad x_0 = s \end{aligned}$$

A step back: **model adjustment?**

Policy $\pi_{\phi}^{\mathrm{MPC}}(s) = u_0^{\star}$ from

$$\begin{aligned} \min_{x,u} \quad & \bar{V}(x_N) + \sum_{k=0}^{N-1} l(x_k, u_k) \\ \text{s.t.} \quad & x_{k+1} = f_{\phi}(x_k, u_k) \\ & h(x_k, u_k) \geq 0, \quad x_0 = s \end{aligned}$$

Adjust ϕ according to (e.g.)

1. $\min_{\phi} \mathbb{E}\left[\left\| f_{\phi}(S, A) - S^{+} \right\|^2\right]$ vs.
2. $\min_{\phi} J\left(\pi_{\phi}^{\mathrm{MPC}}\right)$

... **is that different?**

A step back: **model adjustment?**

Policy $\pi_{\phi}^{\mathrm{MPC}}(s) = u_0^{\star}$ from

$$\begin{aligned} \min_{x,u} \quad & \bar{V}(x_N) + \sum_{k=0}^{N-1} l(x_k, u_k) \\ \text{s.t.} \quad & x_{k+1} = f_{\phi}(x_k, u_k) \\ & h(x_k, u_k) \geq 0, \quad x_0 = s \end{aligned}$$

Adjust ϕ according to (e.g.)

1. $\min_{\phi} \mathbb{E}\left[\left\| f_{\phi}(S, A) - S^{+} \right\|^2\right]$ vs.
2. $\min_{\phi} J\left(\pi_{\phi}^{\mathrm{MPC}}\right)$

... **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

RL improves over model fitting?

A step back: **model adjustment?**

Policy $\pi_\phi^{\mathrm{MPC}}(s) = u_0^\star$ from

$$\begin{aligned} \min_{x,u} \quad & \bar{V}(x_N) + \sum_{k=0}^{N-1} l(x_k, u_k) \\ \text{s.t.} \quad & x_{k+1} = f_\phi(x_k, u_k) \\ & h(x_k, u_k) \geq 0, \quad x_0 = s \end{aligned}$$

Adjust ϕ according to (e.g.)

1. $\min_\phi \mathbb{E}\left[\left\| f_\phi(S, A) - S^+ \right\|^2\right]$ vs.
2. $\min_\phi J\left(\pi_\phi^{\mathrm{MPC}}\right)$

... **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

Formal answers:* 1. and 2. are

- the same for LQR
- *locally* the same for smooth MDPs with a "narrow" optimal steady state distribution away from the constraints

RL improves over model fitting?

A step back: **model adjustment?**

Policy $\pi_\phi^{\mathrm{MPC}}(s) = u_0^\star$ from

$$\begin{aligned} \min_{x,u} \quad & \bar{V}(x_N) + \sum_{k=0}^{N-1} l(x_k, u_k) \\ \text{s.t.} \quad & x_{k+1} = f_\phi(x_k, u_k) \\ & h(x_k, u_k) \geq 0, \quad x_0 = s \end{aligned}$$

Adjust ϕ according to (e.g.)

1. $\min_\phi \mathbb{E}\left[\left\| f_\phi(S, A) - S^+ \right\|^2\right]$ vs.
2. $\min_\phi J\left(\pi_\phi^{\mathrm{MPC}}\right)$

... **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

Formal answers:* 1. and 2. are

- the same for LQR
- *locally* the same for smooth MDPs with a "narrow" optimal steady state distribution away from the constraints

Practical answers:

- RL for tracking(-like) MPC to a fixed reference is disappointing
- Otherwise the gain of performance from RL is often good

Wrap-up – Why does it work?

Take-home messages

- The **MDP** is the common language linking **MPC** and **RL**
- Read MPC as a **structured model of** $Q^\star, \pi^\star$ – via a holistic parametrization ($\bar{V}_\phi$, l_ϕ, f_ϕ, h_ϕ), not as fitting f to reality – exact even with an inexact model
- **RL** tunes ϕ to $\min_\phi J(\pi_\phi^{\mathrm{MPC}})$ from data (policy gradient, Q-learning)
- Keep MPC's structure, constraints & guarantees; gain data-driven real-world performance

Next in this tutorial:

- Differentiable NMPC with `acados` – Katrin Baumgärtner
- MPCRL with `leap-c` – Jasper Hoffmann

Thank you! Questions?

Slides & more