

Synthesis of Model Predictive Control and Reinforcement Learning

Dirk Reinhardt

School on Model Predictive Control and Reinforcement Learning
Freiburg, 7-16 October 2026

universität freiburg



NTNU | Norwegian University of
Science and Technology



**CorPower
Ocean**



Reinforcement Learning

- ▶ Machine-Learning Community: *Use learning to mitigate complexities*
- ▶ **Philosophy:** Trial-and-error learning to build policies directly from experience (implicit)
- ▶ Offline Solution (explicit function)
- ▶ Simulation first
- ▶ Roots in Computer Science



Model Predictive Control

- ▶ Control Community: *Use feedback to mitigate uncertainties*
- ▶ **Philosophy:** Model-based optimization over engineering models for planning (explicit)
- ▶ Online Solution (implicit function)
- ▶ Theory first
- ▶ Roots in Process Control

Motivation:

- ▶ MPC and RL are complementary approaches for solving MDPs
- ▶ Nearly orthogonal advantages: weaknesses of one are strengths of the other
- ▶ Growing research interest in combination methods

Goals of this presentation:

- ▶ Illuminate differences, similarities, and fundamentals enabling combinations
- ▶ Categorize existing work based on actor-critic RL framework
- ▶ Show real world success stories

Synthesis of Model Predictive Control and Reinforcement Learning: Survey and Classification*

Rudolf Reiter^{a,*,*}, Jasper Hoffmann^{b,*,*}, Dirk Reinhardt^d, Florian Messerer^a, Katrin Baumgärtner^a, Shambhura Sawant^d,
Joschka Bödecker^a, Moritz Diehl^{a,c}, Sebastian Gros^d

^aDepartment of Microsystems Engineering, University of Freiburg, Freiburg im Breisgau, 79110, Baden-Württemberg, Germany

^bDepartment of Computer Science, University of Freiburg, Freiburg im Breisgau, 79110, Baden-Württemberg, Germany

^cDepartment of Mathematics, University of Freiburg, Freiburg im Breisgau, 79104, Baden-Württemberg, Germany

^dDepartment of Engineering Cybernetics, Norwegian University of Science and Technology, Trondheim, 7034, Norway



Comparing MPC and RL

MPC+NN: Inference Architectures

- Parameterized

- Parallel

- Hierarchical

- Integrated

- Algorithmic

MPC+RL: Taxonomy of Combination Approaches

- MPC as an Expert Actor

- MPC within the Deployed Policy

 - Aligned Learning

 - Closed-Loop Optimal Learning

- MPC as the Critic

- MPC for Preprocessing/Postprocessing

Current Challenges

Summary

Comparing MPC and RL

MPC+NN: Inference Architectures

- Parameterized

- Parallel

- Hierarchical

- Integrated

- Algorithmic

MPC+RL: Taxonomy of Combination Approaches

- MPC as an Expert Actor

- MPC within the Deployed Policy

 - Aligned Learning

 - Closed-Loop Optimal Learning

- MPC as the Critic

- MPC for Preprocessing/Postprocessing

Current Challenges

Summary

Comparison of Practical Properties



Property	MPC	RL
state-space	model specific ✗	(quite) arbitrary ✓
model requirements	differentiability/ online simulation ✗	offline simulation ✓
uncertainty	guarantees with known uncertainty ✓	probabilistic guarantees ✓
stability	strong theory ✓	minor theory ✗
constraint handling	inherent ✓	hard to achieve ✗
online computation time	high ✓	low ✗
offline computation time	low ✗	high ✓
adaptability	inherent ✓	needs retraining ✗
generalization	inherent ✓	poor when out-of-distribution ✗

Comparison of practical properties between MPC and RL.
The evaluation is simplified and conceptual. Exemptions may exist.

MPC Problem

$$\begin{aligned} \min_z \quad & \sum_{k=0}^{N-1} l^{\text{MPC}}(x_k, u_k) + \bar{V}^{\text{MPC}}(x_N) \\ \text{s.t.} \quad & x_0 = s, \\ & x_{k+1} = f^{\text{MPC}}(x_k, u_k), \\ & 0 \leq h^{\text{MPC}}(x_k, u_k), \\ & 0 \leq h_N^{\text{MPC}}(x_N). \end{aligned}$$

Change in dynamics?

Modify f^{MPC} $\rightarrow$ new optimal policy

RL Policy (DDPG)

$$\begin{aligned} w &\leftarrow w + \frac{\alpha_w}{B} \sum_{i=1}^B \delta_i \nabla_w Q_w(S_i, A_i), \\ \theta &\leftarrow \theta + \frac{\alpha_\theta}{B} \sum_{i=1}^B \nabla_\theta \mu_\theta(S_i) \nabla_a Q_w(S_i, a)|_{a=\mu_\theta(S_i)}, \\ \delta_i &:= l(S_i, A_i) + \gamma Q_{\bar{w}}(S_i^+, A_i^+) - Q_w(S_i, A_i), \\ &\text{with } (S_i, A_i, S_i^+) \sim \mathcal{D}^{\text{buffer}}, \quad A_i^+ = \mu_\theta(S_i^+), \end{aligned}$$

Change in dynamics?

Collect new data $\mathcal{D}^{\text{buffer}}$ $\rightarrow$ retrain policy

MPC Problem

$$\begin{aligned} \min_z \quad & \sum_{k=0}^{N-1} l^{\text{MPC}}(x_k, u_k) + \bar{V}^{\text{MPC}}(x_N) \\ \text{s.t.} \quad & x_0 = s, \\ & x_{k+1} = f^{\text{MPC}}(x_k, u_k), \\ & 0 \leq h^{\text{MPC}}(x_k, u_k), \\ & 0 \leq h_N^{\text{MPC}}(x_N). \end{aligned}$$

Constraints?

Directly handled through h^{MPC} and h_N^{MPC}

RL Policy (DDPG)

$$\begin{aligned} w &\leftarrow w + \frac{\alpha_w}{B} \sum_{i=1}^B \delta_i \nabla_w Q_w(S_i, A_i), \\ \theta &\leftarrow \theta + \frac{\alpha_\theta}{B} \sum_{i=1}^B \nabla_\theta \mu_\theta(S_i) \nabla_a Q_w(S_i, a)|_{a=\mu_\theta(S_i)}, \\ \delta_i &:= l(S_i, A_i) + \gamma Q_{\bar{w}}(S_i^+, A_i^+) - Q_w(S_i, A_i), \\ &\text{with } (S_i, A_i, S_i^+) \sim \mathcal{D}^{\text{buffer}}, \quad A_i^+ = \mu_\theta(S_i^+), \end{aligned}$$

Constraints?

Hard to handle formally $\rightarrow$ use penalties in cost l or restrict action space of policy μ_θ .

MPC Problem

$$\begin{aligned} \min_z \quad & \sum_{k=0}^{N-1} l^{\text{MPC}}(x_k, u_k) + \bar{V}^{\text{MPC}}(x_N) \\ \text{s.t.} \quad & x_0 = s, \\ & x_{k+1} = f^{\text{MPC}}(x_k, u_k), \\ & 0 \leq h^{\text{MPC}}(x_k, u_k), \\ & 0 \leq h_N^{\text{MPC}}(x_N). \end{aligned}$$

Inference time?

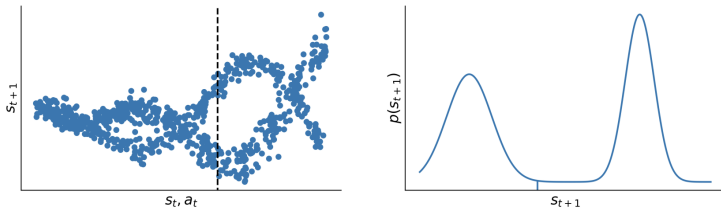
Often high due to **online optimization** → may be prohibitive for fast systems

RL Policy (DDPG)

$$\begin{aligned} w &\leftarrow w + \frac{\alpha_w}{B} \sum_{i=1}^B \delta_i \nabla_w Q_w(S_i, A_i), \\ \theta &\leftarrow \theta + \frac{\alpha_\theta}{B} \sum_{i=1}^B \nabla_\theta \mu_\theta(S_i) \nabla_a Q_w(S_i, a)|_{a=\mu_\theta(S_i)}, \\ \delta_i &:= l(S_i, A_i) + \gamma Q_{\bar{w}}(S_i^+, A_i^+) - Q_w(S_i, A_i), \\ &\text{with } (S_i, A_i, S_i^+) \sim \mathcal{D}^{\text{buffer}}, \quad A_i^+ = \mu_\theta(S_i^+), \end{aligned}$$

Inference time?

Low, only need forward pass of μ_θ → suitable for fast systems



[Image: Moerland et al. 2022]

- ▶ **Definition:** Environment is inherently stochastic - deterministic future state prediction impossible even with perfect model knowledge
- ▶ **RL approaches:**
 - ▶ **Natural fit:** Stochasticity inherent in MDP framework
 - ▶ **Direct training:** Policies trained on real-world environment to account for actual uncertainty
 - ▶ **Challenge:** Rare events (large but infrequent deviations) make value estimation difficult

Challenge: Training environment often differs from deployment environment

RL approaches:

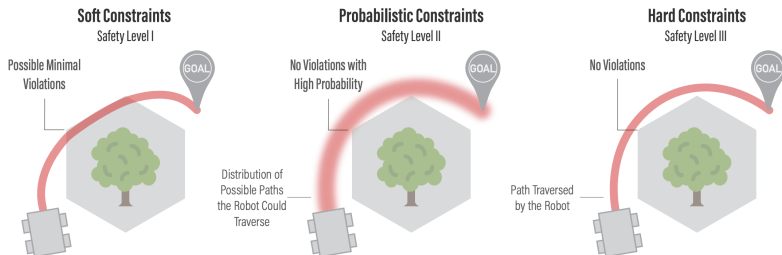
- ▶ Domain randomization
- ▶ Robust RL
- ▶ Meta RL - training on adaptive distribution of models

RL assumption: Real-world environment lies within training model distribution

MPC approaches:

- ▶ Stochastic MPC - explicitly accounts for model mismatch
- ▶ Robust MPC - handles correlated predictive errors across time
- ▶ Distributionally robust MPC - robustifies against distribution mismatches

MPC advantage: Inherent robustness - performs well even without explicit model mismatch consideration



[Image: Brunke et al. 2023]

- ▶ **MPC:** Constraint satisfaction and stability are key strengths:
 - ▶ **Challenge:** Hard to handle probabilistic constraints with (unknown) uncertainty
 - ▶ **Common MPC approaches:** Robust MPC, tube-based MPC
- ▶ **RL:** Increasingly focused on safety guarantees using constrained MDPs:
 - ▶ **Challenge:** Safe RL policies often become too conservative or may still violate constraints.
 - ▶ **Common RL approaches:** Safe action sets, Lagrange formulation, trust-region optimization, control-barrier functions, MPC based safety filter

Comparing MPC and RL

MPC+NN: Inference Architectures

- Parameterized

- Parallel

- Hierarchical

- Integrated

- Algorithmic

MPC+RL: Taxonomy of Combination Approaches

- MPC as an Expert Actor

- MPC within the Deployed Policy

 - Aligned Learning

 - Closed-Loop Optimal Learning

- MPC as the Critic

- MPC for Preprocessing/Postprocessing

Current Challenges

Summary

Parameterized NLP with parameters ϕ :

OCP:

$$\begin{aligned} \min_z \quad & L_\phi(z) \\ \text{s.t.} \quad & x_0 = s, \\ & x_{k+1} = f_\phi^{\text{MPC}}(x_k, u_k), \quad 0 \leq k < N, \\ & 0 \leq h_\phi^{\text{MPC}}(x_k, u_k), \quad 0 \leq k < N, \\ & 0 \leq \tilde{h}_\phi^{\text{MPC}}(x_N), \end{aligned}$$

Objective:

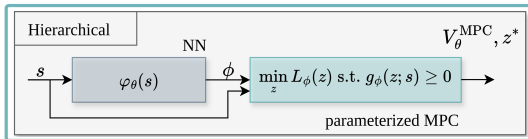
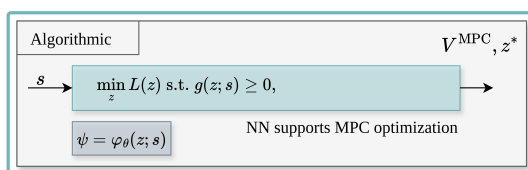
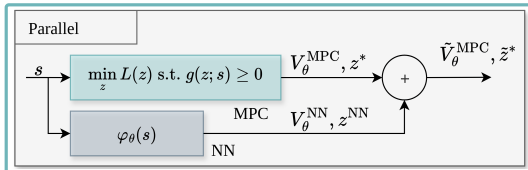
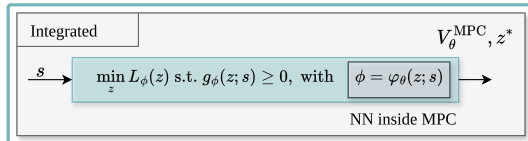
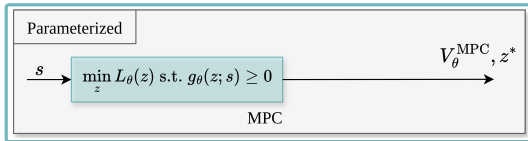
$$L_\phi(z) := \bar{V}_\phi^{\text{MPC}}(x_N) + \sum_{k=0}^{N-1} l_\phi^{\text{MPC}}(x_k, u_k).$$

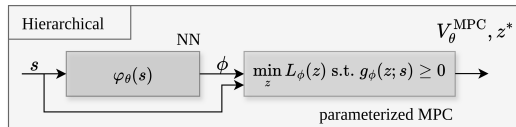
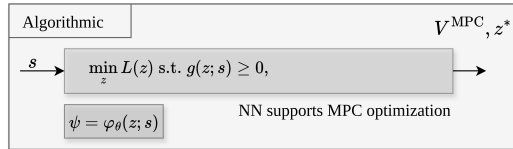
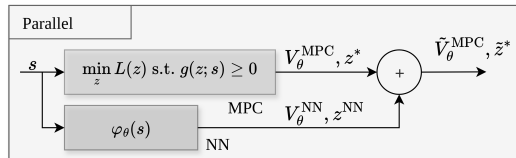
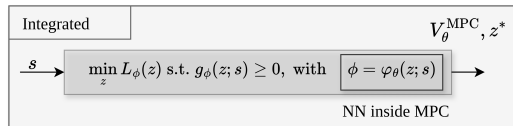
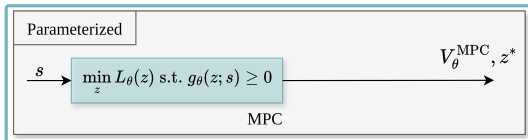
Short notation:

$$\min_z L_\phi(z) \quad \text{s.t.} \quad g_\phi(z; s) \geq 0.$$

Parameters ϕ can affect: objective L_ϕ , dynamics f_ϕ^{MPC} , and constraints h_ϕ^{MPC} .

Note: g_ϕ encodes both dynamics and constraints.





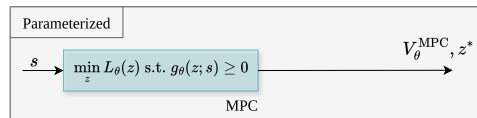
Architecture: Parameters θ are **constant** and independent of decision variables z or state s

Key characteristics:

- ▶ Parameters learned offline, fixed during deployment
- ▶ Optimization-friendly MPC formulation
- ▶ No evaluation of highly nonlinear functions

Advantages:

- ✓ Preserves MPC problem structure
- ✓ No NN evaluation during inference
- ✓ Suitable for standard MPC solvers
- ✓ Lower computational complexity
- ✗ Less flexibility



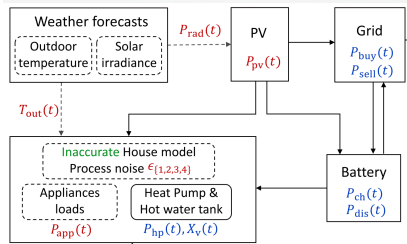
Examples:

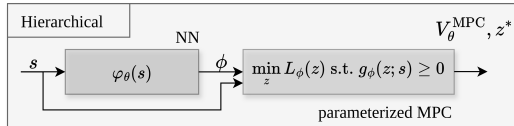
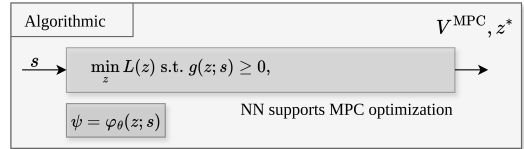
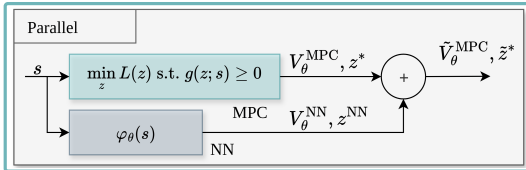
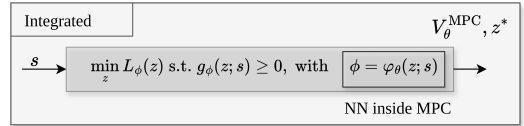
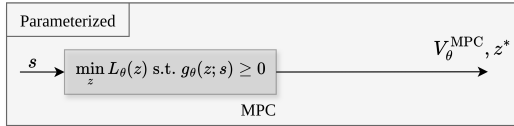
- ▶ Simplified models tuned in RL loop for building energy control [Cai et al. 2023]

A Learning-Based Model Predictive Control Strategy for Home Energy Management Systems

WENQI CAI¹, SHAMBHURAJ SAWANT¹, DIRK REINHARDT¹,
SOROUSH RASTEGARPOUR², AND SÉBASTIEN GROS¹

- ▶ Build a simplified MPC scheme and parameterize model, cost constraints
- ▶ Learn parameters via RL
- ▶ Maintains MPC structure, lower inference time compared to full model





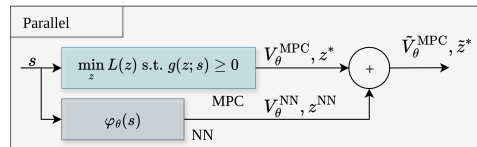
Architecture: MPC problem usually not parameterized; NN evaluated in parallel to correct MPC output

Key characteristics:

- ▶ MPC problem typically not parameterized
- ▶ NN evaluated in parallel to MPC optimization
- ▶ NN output corrects optimal solution or value function
- ▶ MPC and NN operate independently

Trade-offs:

- ▶ No differentiation through optimization ✓
- ▶ Potentially unsafe actions due to perturbation ✗



Examples:

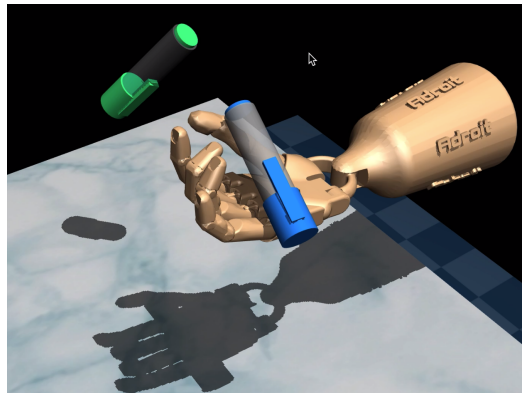
- ▶ Parallel architecture for value function approximation [Bhardwaj et al. 2021]

BLENDING MPC & VALUE FUNCTION APPROXIMATION FOR EFFICIENT REINFORCEMENT LEARNING

Mohak Bhardwaj¹ Sanjiban Choudhury² Byron Boots¹

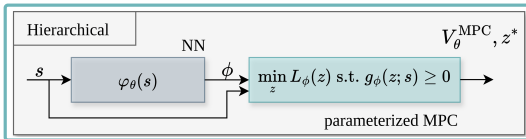
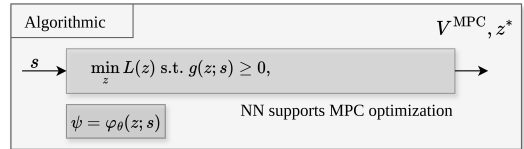
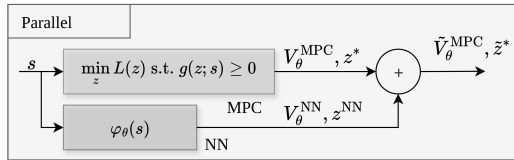
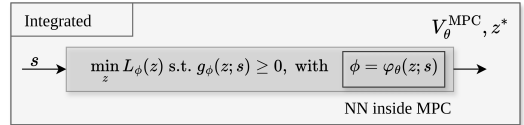
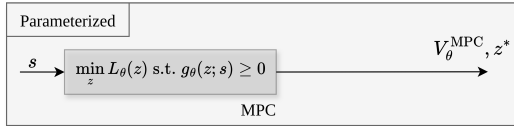
¹ University of Washington ² Aurora Innovation Inc.

- ▶ Use MPC for local value estimates
- ▶ NN provides corrections
- ▶ Blend both estimates via weighting factor λ
- ▶ Improves data efficiency and adds structure
- ▶ Evaluated on 24DOF dexterous hand (fit pose) among others



[Image: Bhardwaj et al. 2020]

Hierarchical Architecture



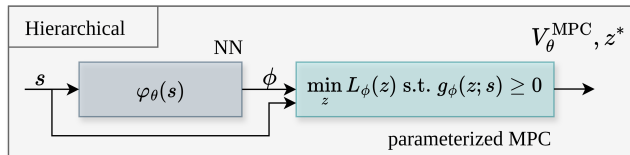
Architecture: NN provides input to MPC, parameter ϕ depends on state via $\phi = \varphi_{\theta}(s)$

Key characteristics:

- ▶ NN evaluated **before** solving optimization problem
- ▶ Function $\varphi_{\theta}(s)$ evaluated **independently** of MPC
- ▶ Nonlinearity does **not affect** optimization problem

Tradeoffs:

- ✓ Optimization not harder to solve
- ✓ Gradient-Based MPC solvers work
- ✗ MPC theory may not hold

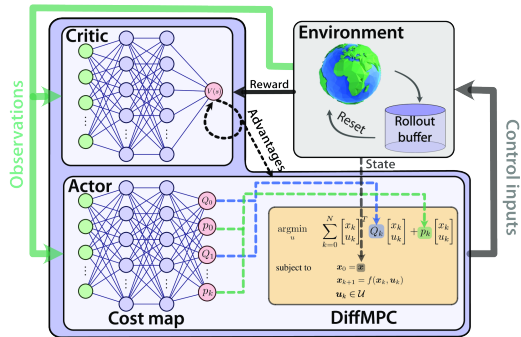
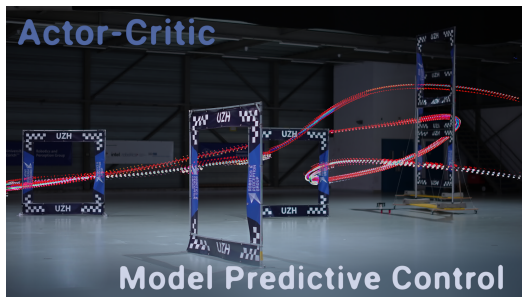


Examples: Modification of:

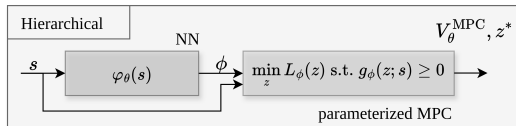
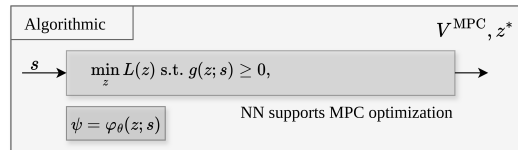
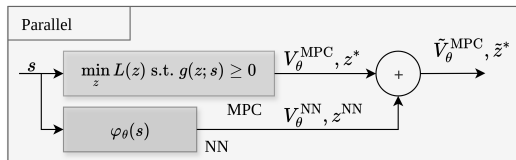
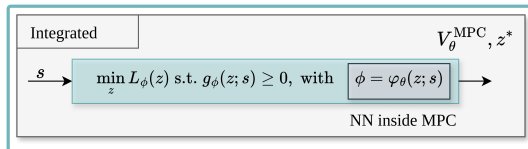
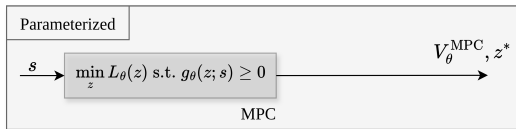
- ▶ Reference trajectory [Brito et al. 2021]
- ▶ Cost [Reiter et al. 2023] [Romero et al. 2024]
- ▶ Constraints [Jacquet et al. 2024]

Actor-Critic Model Predictive Control

Angel Romero, Yunlong Song, Davide Scaramuzza



[Image: Romero et al. 2024]



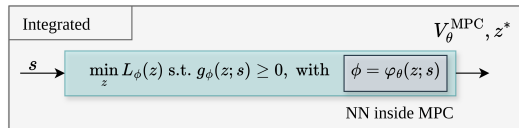
Architecture: Parameters depend on both state s and optimization variables

Key characteristics:

- ▶ NN is **embedded** in the optimization problem
- ▶ Cannot be evaluated separately
- ▶ May output value V_{θ}^{MPC} or optimal decision variables z^*

Tradeoffs:

- ✓ Highly expressive
- ✗ Highly nonlinear, possibly nonconvex
- ✗ Computationally expensive
- ✗ Requires $\nabla_z \varphi_{\theta}(z; s)$



Examples:

- ▶ Latent space MPC [Lowrey et al. 2019] [Hansen et al. 2022]
- ▶ Learned nonlinear residual dynamics [Salzmann et al. 2023]
- ▶ Cost function [Seel et al. 2022]

Real-Time Neural MPC:

- ▶ Learn NN residual from data
- ▶ Use model for MPC planning
- ▶ Real-time feasible
- ▶ Tracking error reduced by 82%

Anyone wants to try? Wrappers for
acados and CasADi available:



[l4casadi]



[l4acados]

IEEE ROBOTICS AND AUTOMATION LETTERS, VOL. 8, NO. 4, APRIL 2023

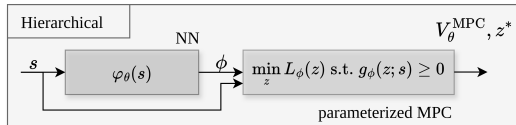
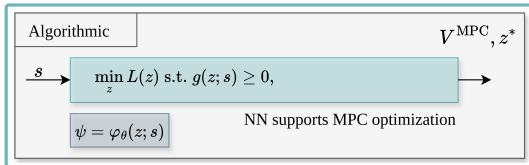
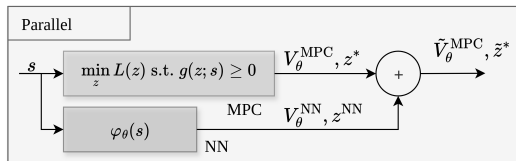
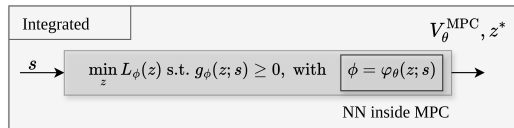
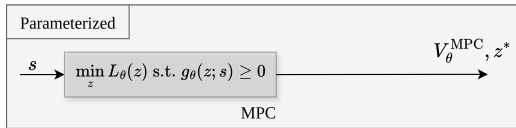
2397

Real-Time Neural MPC: Deep Learning Model Predictive Control for Quadrotors and Agile Robotic Platforms

Tim Salzmann[✉], Elia Kaufmann[✉], *Member, IEEE*, Jon Arrizabalaga[✉], *Graduate Student Member, IEEE*,
Marco Pavone[✉], *Member, IEEE*, Davide Scaramuzza[✉], *Senior Member, IEEE*, and Markus Ryll[✉], *Member, IEEE*



[Image: Salzmann et al. 2023]



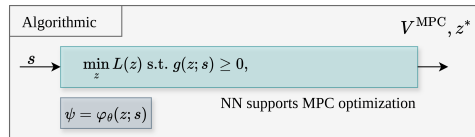
Architecture: RL policy guides the MPC optimization algorithm

Key characteristics:

- ▶ RL policy alters solver hyperparameters ψ
- ▶ Optimization problem remains unchanged
- ▶ Examples of hyperparameters:
 - ▶ Initial guess (MPPI/SQP)
 - ▶ Active-set predictions
 - ▶ Horizon length

Tradeoffs:

- ✓ Improves online computation time
- ✓ Can help escape local minima
- ✗ Increased complexity



Examples:

- ▶ MPPI Warm-starting [Hansen et al. 2024]
- ▶ SQP-RTI Warm-starting [Reiter et al. 2025]

Idea:

- ▶ MPC in latent space (MPPI)
- ▶ Learned: Encoder, Decoder, Model, Policy, Latent Value Function
- ▶ Model learned only via reward decoding - no reconstruction loss
- ▶ Learns policy in latent space used as initializer in deployment

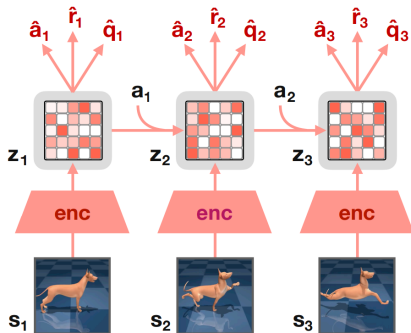
Architectures:

- ▶ Initializing policy: **algorithmic**
- ▶ Learned MPC components: **integrated**

TD-MPC²:
Scalable, Robust World Models for Continuous Control

Nicklas Hansen*, Hao Su[†], Xiaolong Wang[†]

^{*}University of California San Diego, [†]Equal advising
{nihansen, haosu, xiw012}@ucsd.edu



[Image: Hansen et al. 2023]

Comparing MPC and RL

MPC+NN: Inference Architectures

Parameterized

Parallel

Hierarchical

Integrated

Algorithmic

MPC+RL: Taxonomy of Combination Approaches

MPC as an Expert Actor

MPC within the Deployed Policy

Aligned Learning

Closed-Loop Optimal Learning

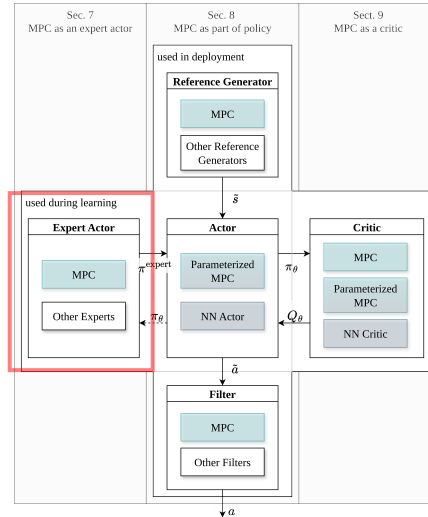
MPC as the Critic

MPC for Preprocessing/Postprocessing

Current Challenges

Summary

MPC+RL: Taxonomy of Combination Approaches



MPC used to generate training data for RL policy

Characteristics:

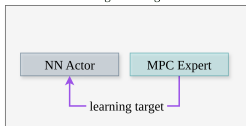
- ▶ MPC generates state-action pairs
- ▶ RL policy learns to mimic MPC behavior
- ▶ Can use behavioral cloning or DAgger
- ▶ See also [Karg and Lucia 2020]

Properties:

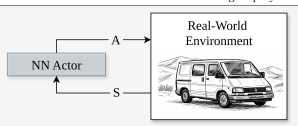
- ✓ Faster inference than MPC
- ✓ Leverages domain knowledge
- ✓ Sample-efficient learning
- ✗ Limited to MPC performance

MPC within the Expert Actor (Sect. 6)

Architecture During Learning



Architecture During Deployment



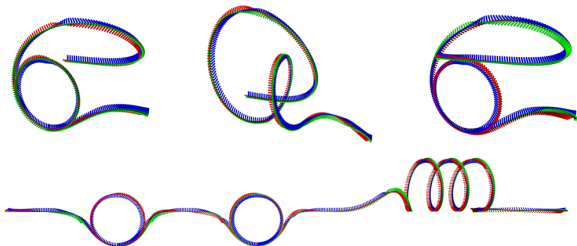
Examples:

- ▶ Drone acrobatics [Kaufmann et al. 2020]
- ▶ Autonomous driving [Hoffmann et al. 2024]
- ▶ Robot manipulation [Mamedov et al. 2024]

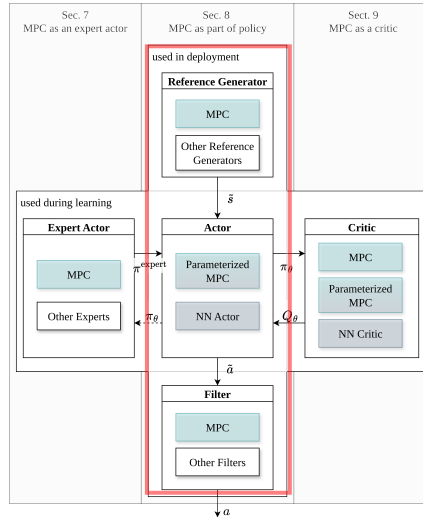
Deep Drone Acrobatics

Elia Kaufmann^{*†}, Antonio Loquercio^{*†}, René Ranftl[†], Matthias Müller[†], Vladlen Koltun[†], Davide Scaramuzza[†]

- ▶ Use a privileged MPC expert actor
- ▶ At deployment visual abstractions only (no state)
- ▶ Zero-shot transfer



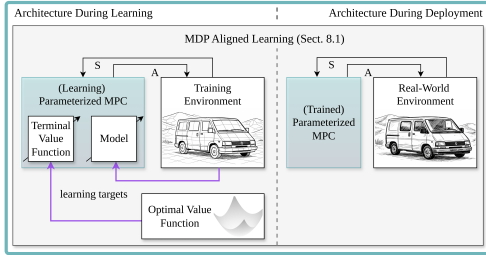
[Image: Kaufmann et al., RSS 2020,]



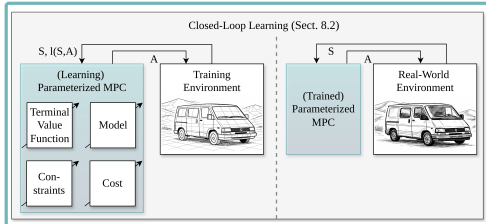
MPC within the Deployed Policy



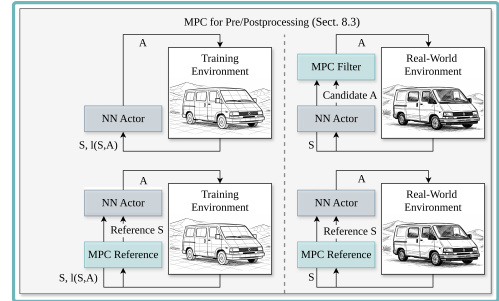
MDP Aligned

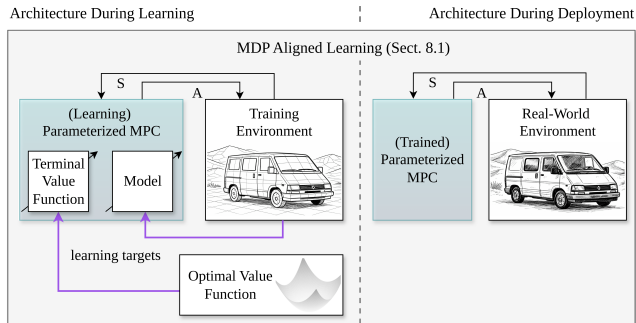


Closed-Loop



Pre/Postprocessing





Paradigm: Learn individual components of the MPC controller to **approximate the true environment (MDP)**. This aligns the controller's internal understanding with reality.

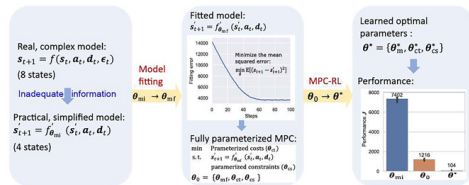
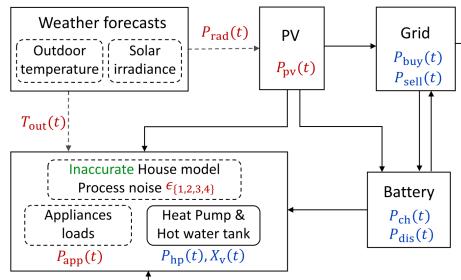
Components aligned with MDP:

- ⚙️ Approximate the system's dynamics by learning: $f_{\theta}^{\text{MPC}} \approx P$
- 🧭 Approximate the infinite-horizon cost with RL: $\bar{V}_{\theta}^{\text{MPC}} \approx V^*$

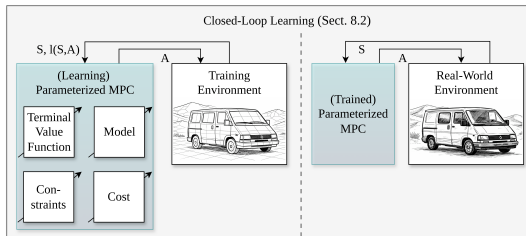
A Learning-Based Model Predictive Control Strategy for Home Energy Management Systems

WENQI CAI¹, SHAMBHURAJ SAWANT¹, DIRK REINHARDT¹,
SOROUSH RASTEGARPOUR², AND SÉBASTIEN GROS¹

- Build a simplified MPC scheme and parameterize model, cost constraints
- Learn parameters via RL
- Maintains MPC structure, lower inference time compared to full model



[Image: Cai et al. 2023]



Paradigm:

- 🏆 The focus is on what **works best**, not what is most **accurate**. The internal model or costs may become misaligned.

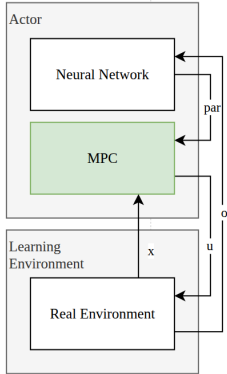
How is it implemented?

- 📈 Learn MPC parameters ϕ to directly optimize the final closed-loop performance $J(\phi)$.
- 🔗 **MPC as part of the Actor:** Differentiable MPC as a layer in a policy network.
- 📦 **MPC as part of the Environment:** An outer-loop RL agent learns to choose the best MPC parameters ϕ as its "action", treating the MPC-controlled system as a black box.

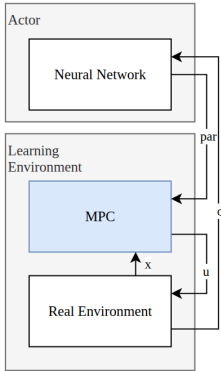
Differentiate Through MPC?



MPC as Part of Actor

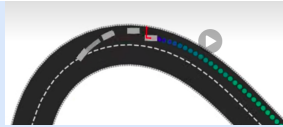


MPC as Part of Environment



A Hierarchical Approach for Strategic Motion Planning in Autonomous Racing

Rudolf Reiter¹, Jasper Hoffmann², Joschka Boedecker² and Moritz Diehl^{1,3}



Actor-Critic Model Predictive Control: Differentiable Optimization meets Reinforcement Learning for Agile Flight



Angel Romero, Elie Aljalbout, Yunlong Song, Davide Scaramuzza



Aligned Learning

 Approximate the true Markov Decision Process

Model Learning: Supervised learning on transition data, e.g.,

$$\min_{\theta} \mathcal{L}(S^+ - f_{\theta}^{\text{MPC}}(S, A))$$

Terminal Value Learning: RL-style updates towards

$$V_{\theta}^{\text{MPC}} \approx V^*$$

Tradeoffs:

- ✓ Interpretable model, cost, constraints
- ✓ Division of design into manageable subtasks
- ✓ Safe, generalizes better
- ✗ Sub-optimal closed-loop performance
- ✗ May require complex models

Closed-Loop Learning

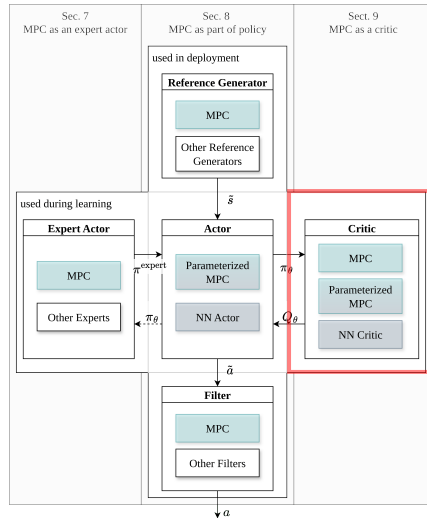
 Approximate the optimal policy

Policy Learning: Directly optimize closed-loop performance, towards

$$\mu_{\theta}^{\text{MPC}} \approx \mu^*$$

Tradeoffs:

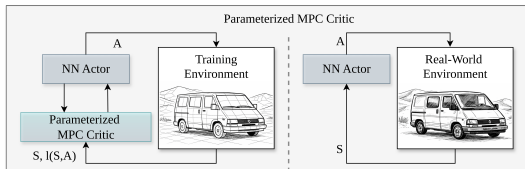
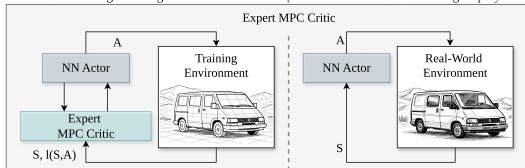
- ✓ Direct optimization of actual objective
- ✓ Can correct for model errors through learning
- ✓ Flexibility in model choice
- ✗ Limited generalization to new scenarios
- ✗ Not representing true system
- ✗ Reduced adaptability when requirements change



MPC within the Critic (Sect. 9)

Architecture During Learning

Architecture During Deployment



$$Q^{\text{MPC}}(s, a) = \min_z \sum_{k=0}^{N-1} l^{\text{MPC}}(x_k, u_k) + \bar{V}^{\text{MPC}}(x_N)$$

s.t. $x_0 = s,$

$$u_0 = a,$$

$$x_{k+1} = f^{\text{MPC}}(x_k, u_k), 0 \leq k < N,$$

$$0 \leq h^{\text{MPC}}(x_k, u_k), 1 \leq k < N,$$

$$0 \leq h_N^{\text{MPC}}(x_N).$$

- ▶ We can use MPC to derive a Q-function Q^{MPC} , which can be used as a critic!
- ▶ We constrain the initial control u_0 to be a .

Instead of using a fixed MPC critic, we can further learn to improve the critic:

- ▶ In [Bhardwaj et al. 2020], the author combine sampling-based MPC with a learned value function using a parallel architecture solving multiple robot manipulation tasks. A simplified version of the critic is given by:

$$Q(s, a) = (1 - \lambda) \underbrace{Q_w(s, a)}_{\text{learned critic}} + \lambda \underbrace{Q^{\text{MPC}}(s, a)}_{\text{model-based}},$$

where Q_w is a learned critic and Q^{MPC} is the MPC-based Q-function.

- ▶ The parameter λ balances the two Q-functions and is decayed over time.
- ▶ (Remember: This was the parallel architecture example)

Example: Learnable Critic from parameterized MPC

- ▶ [Anand et al. 2023] proposed a simple actor-critic scheme, where a single MPC scheme is used for both actor and critic.
- ▶ **Idea:** Close-to-optimal MPC parameters ϕ allow approximation of the optimal Q-function Q^* .

Local Q-function approximation:

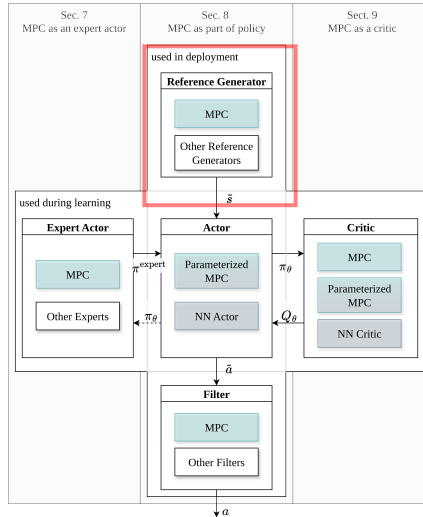
$$Q_w(s, a) = Q_\phi^{\text{MPC}}(s, a) + \nabla_\phi Q_\phi^{\text{MPC}}(s, a)^\top w$$

MPC Q-Function

$$Q_\phi^{\text{MPC}}(s, a) = \min_z \sum_{k=0}^{N-1} l_\phi^{\text{MPC}}(x_k, u_k) + \bar{V}_\phi^{\text{MPC}}(x_N)$$

s.t.

$$\begin{aligned} x_0 &= s, \\ u_0 &= a, \\ x_{k+1} &= f_\phi^{\text{MPC}}(x_k, u_k), \\ 0 &\leq h_\phi^{\text{MPC}}(x_k, u_k), \\ 0 &\leq h_{N_\phi}^{\text{MPC}}(x_N). \end{aligned}$$



MPC used in deployed policy, but not during training

Preprocessing: Reference Generator

Purpose: Provide reference trajectory for RL policy

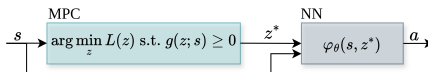
Characteristics:

- ▶ MPC outputs planned trajectory
- ▶ RL policy uses reference as input

Benefit:

- ✓ Interpretable reference trajectory
- ✓ Bandwidth and learning at lower level

MPC as reference generator

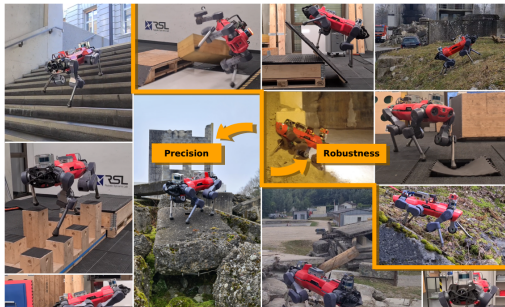


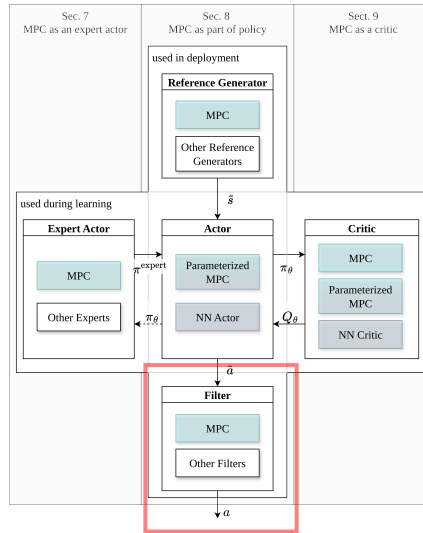
DTC: Deep Tracking Control

FABIAN JENELTEN,^{1*} JUNZHE HE,¹ FARBOD FARSHIDIAN,² AND MARCO HUTTER¹

¹Robotic Systems Lab, ETH Zurich, 8092 Zurich, Switzerland.

²Currently at Boston Dynamics AI Institute, 145 Broadway, Cambridge MA, USA





MPC used in deployed policy, but not during training

Postprocessing: Safety Filter

Purpose: Ensure safety w.r.t. model and constraints

Characteristics:

- ▶ MPC filters RL policy output
- ▶ Projects actions to safe set

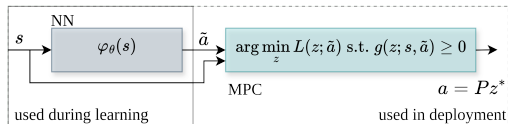
Benefit:

- ✓ Provides safety guarantees w.r.t. model and constraints

Issue:

- ✗ Policy unaware of filter during training
- ✗ May lead to suboptimal actions

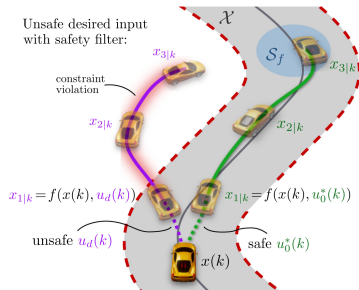
MPC for postprocessing



IEEE ROBOTICS AND AUTOMATION LETTERS, VOL. 6, NO. 4, OCTOBER 2021

A Predictive Safety Filter for Learning-Based Racing Control

Ben Tearle¹, Kim P. Wabersich², Andrea Caron³, and Melanie N. Zeilinger⁴, Member, IEEE



Comparing MPC and RL

MPC+NN: Inference Architectures

- Parameterized

- Parallel

- Hierarchical

- Integrated

- Algorithmic

MPC+RL: Taxonomy of Combination Approaches

- MPC as an Expert Actor

- MPC within the Deployed Policy

 - Aligned Learning

 - Closed-Loop Optimal Learning

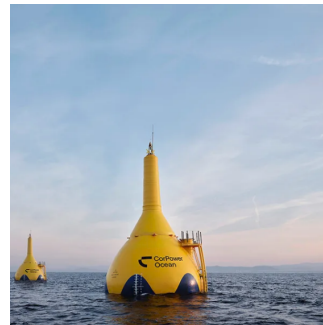
- MPC as the Critic

- MPC for Preprocessing/Postprocessing

Current Challenges

Summary

-  **Computational efficiency:** Scaling to massive parallelization
-  **Algorithms:** *Slow* solving of *complex* problems (MPPI) vs. *fast* solving of *simple* optimization problems (SQP)
-  **Expanding frameworks:** Generalization from MPC to broader planning (combinatorial optimization, mixed-integer, stochastic multi-stage programming)
-  **Multi-agent systems:** Coordination, communication, and distributed decision-making largely unresolved
-  **Applications:** Practical demonstrations beyond energy and robotics (healthcare, logistics)



Having heard this lecture, you should have a better understanding of...



Identify strengths and weaknesses of MPC and RL



Recognize potential benefits of combining both approaches



MPC+NN: Distinguish between parameterized MPC architectures:

▶ Integrated, hierarchical, parallel, parameterized



MPC+RL Design MPC-RL algorithms based on:



MPC as expert actor



MPC within deployed policy



MPC as critic



MPC as reference generator



MPC as a safety filter

Thank you for your attention!