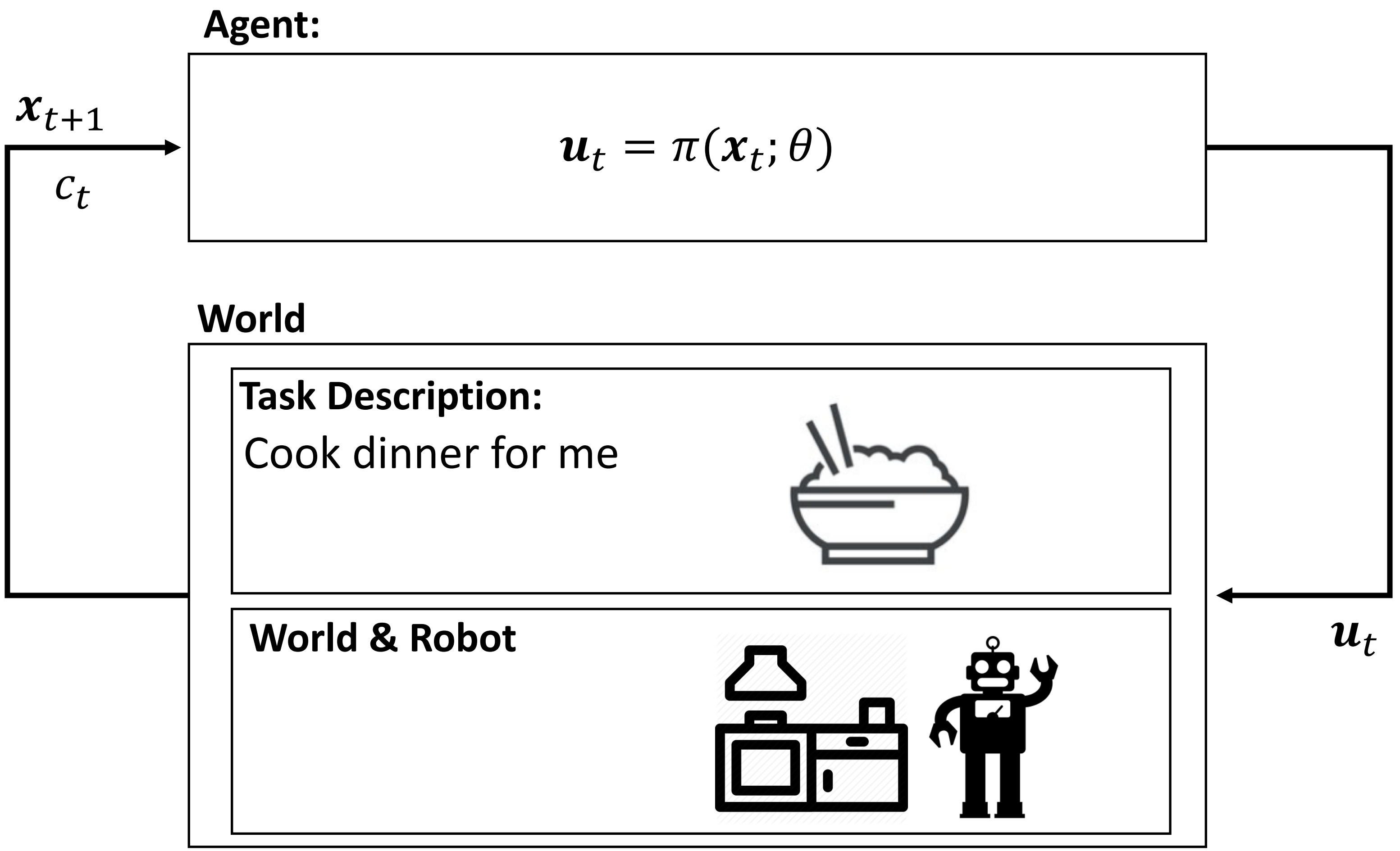
Inductive Biases for Robot Control

Using domain knowledge to enable learning for the physical world

Michael Lutter
michael@robot-learning.de
@_mlutter

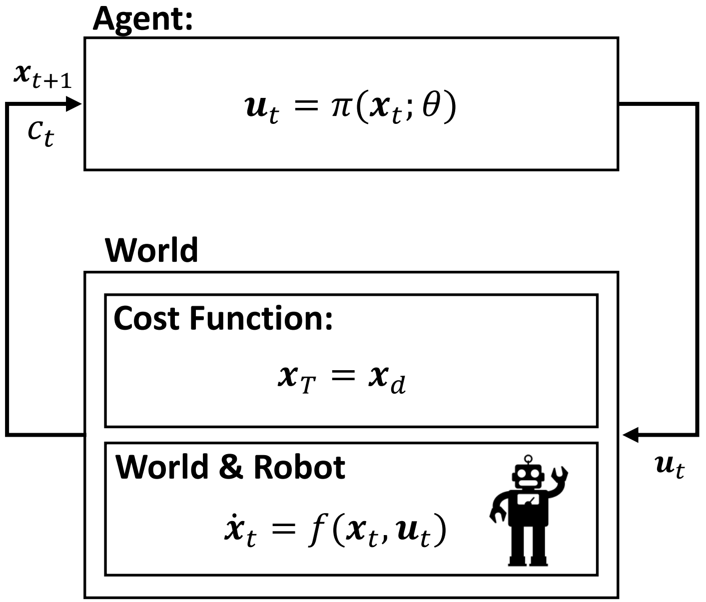
Intelligent Autonomous Systems
Technical University of Darmstadt
London, 20. February 2020

Solving Tasks with Robots



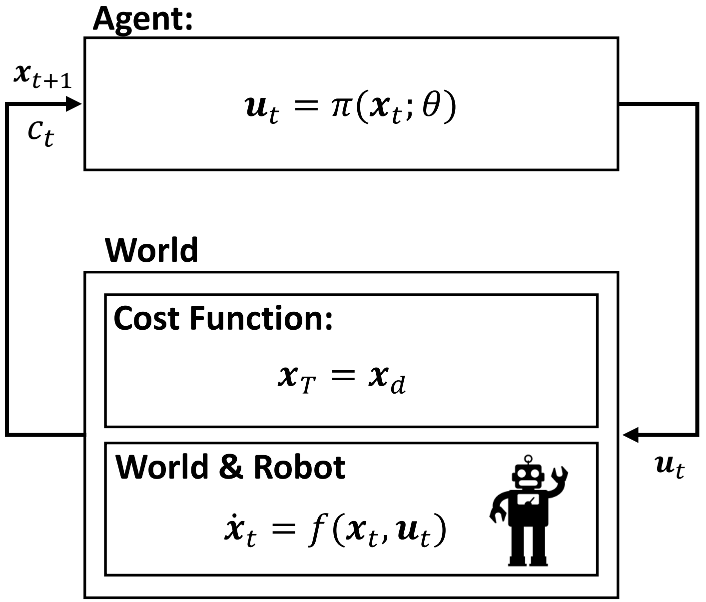
Solving Tasks with Robots

Engineering Approach

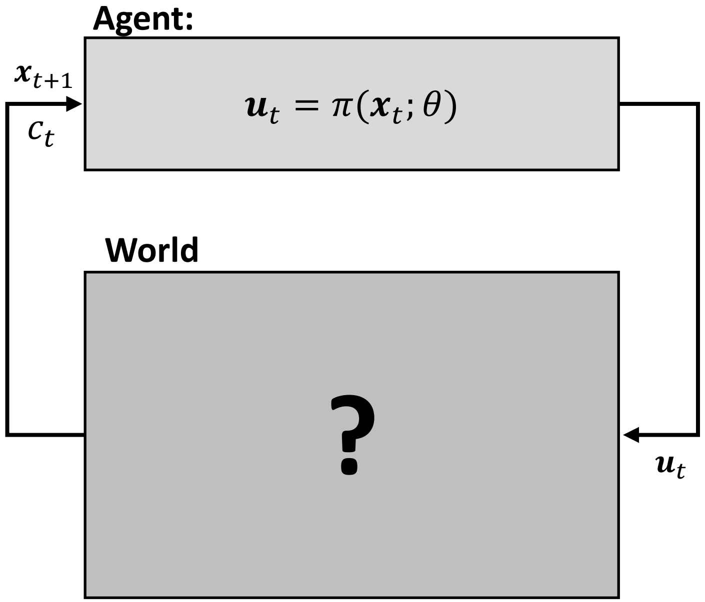


Solving Tasks with Robots

Engineering Approach

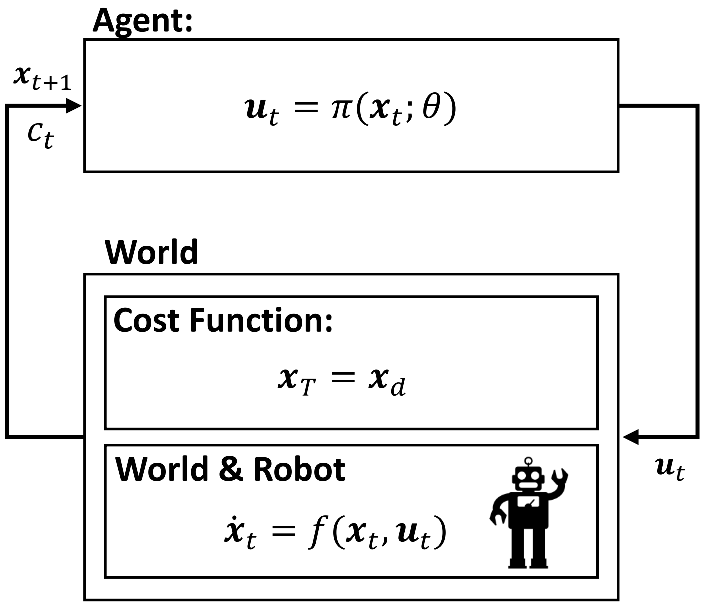


Reinforcement Learning

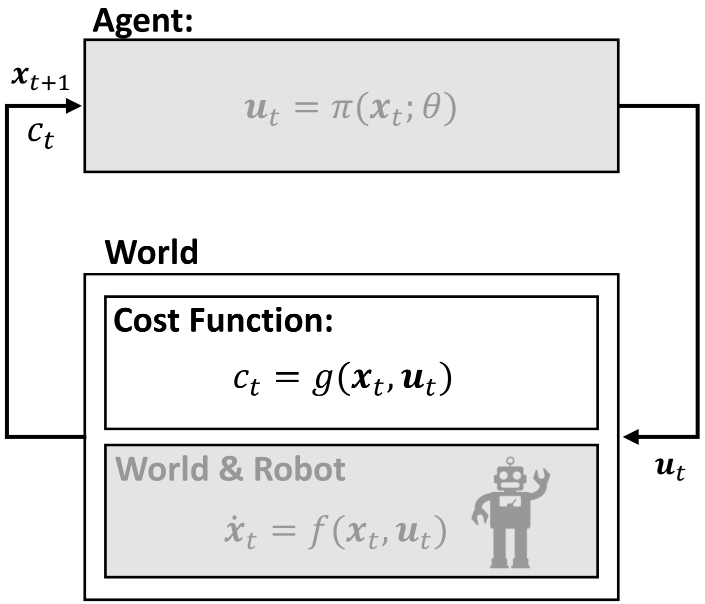


Solving Tasks with Robots

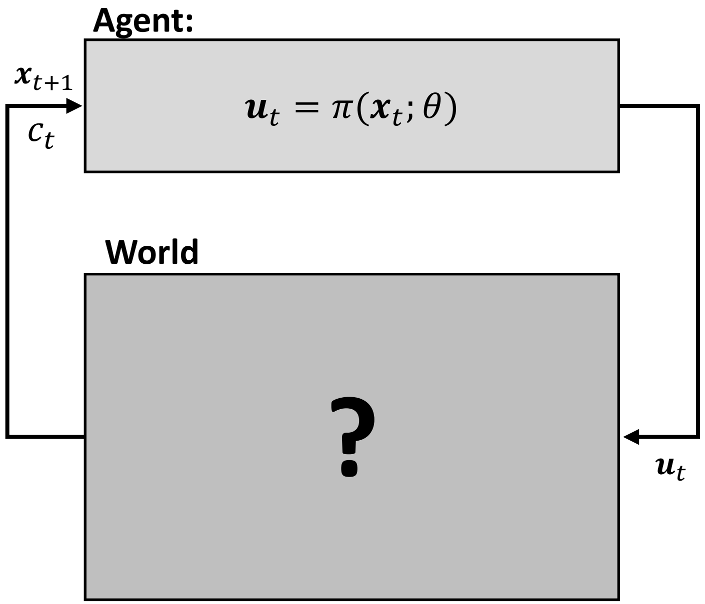
Engineering Approach



Reinforcement Learning + Domain Knowledge

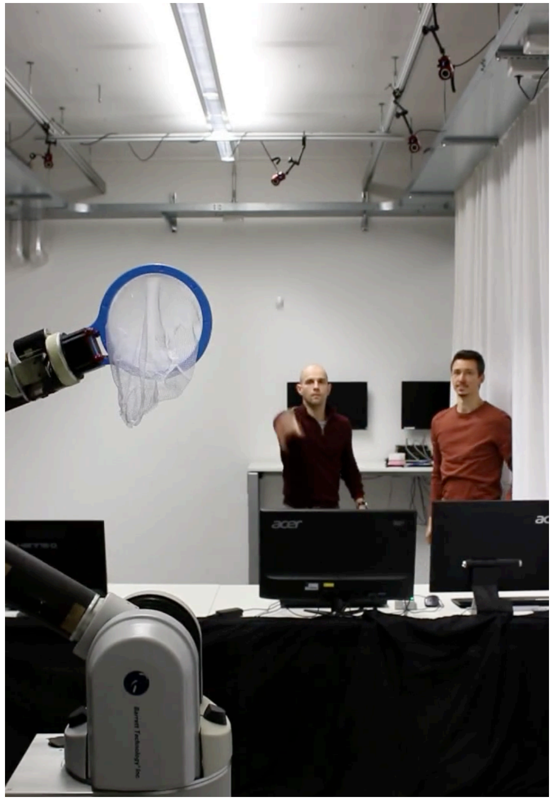


Reinforcement Learning

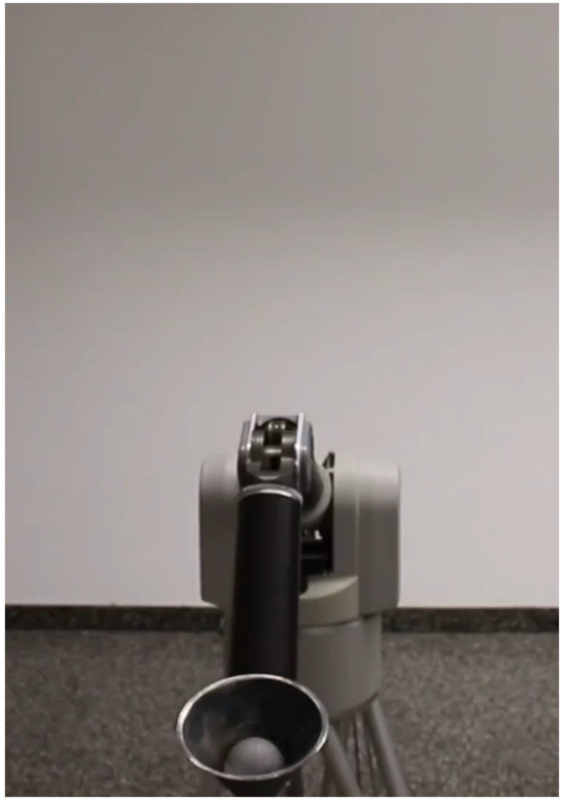


…. and many ‘classical’ robot learning has been doing this years

Tabletennis
(Mülling; Kober; Krömer; Lampert; Schölkopf; Peters; 2013)



Ball Catching
(Patzwahl*; Lutter; Peters; 2020)



Ball Catching
(Plöger*; Lutter*; Peters; 2020)

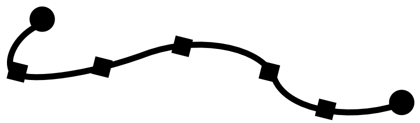
From single movements to complex behaviors

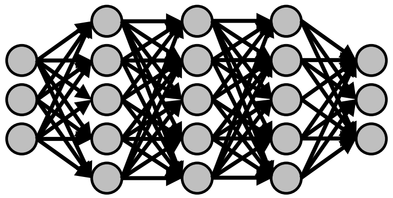
Task Complexity

Low **High**

Use task specific knowledge to structure representations & optimization

?





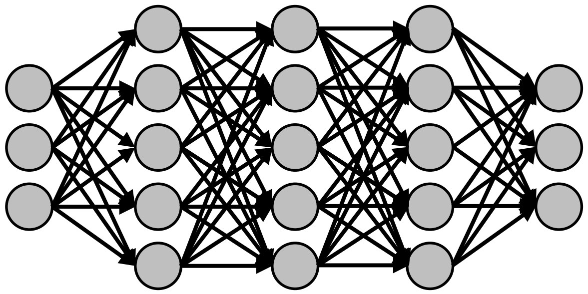
Paddle Juggling (Aboaf et. al., 1988,)
Ball in a Cup (Kober & Peters, 2009)
Pancake Flipping (Kormushev et. al., 2010)
Tabletennis (Muelling et. al., 2013)
Toss Juggling (Ploeger et. al., 2020)

Overview

- Introduction
- Differential Equations as Prior for Deep Networks
- Robotics Control Applications:
  - Deep Lagrangian Networks – Learning physically plausible Models
  - HJB Feedback Control – Exploiting the HJB Equation for Optimal Feedback Control
- Conclusion

First-Order Principles as Model Prior for Deep Networks

Deep Neural Network



First-Order Differential Equation

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{\boldsymbol{q}}} - \frac{\partial L}{\partial \boldsymbol{q}} = \boldsymbol{\tau} \qquad \boldsymbol{F} = m\,\ddot{\boldsymbol{x}}$$

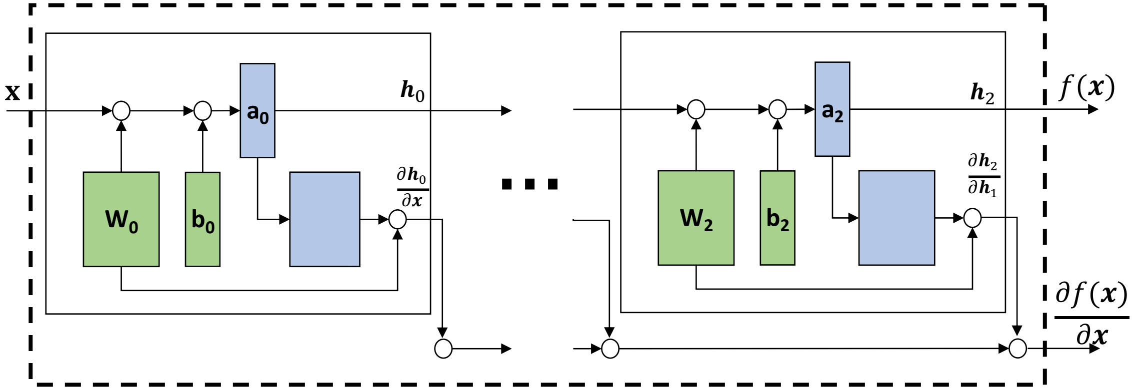
$$\frac{d\boldsymbol{p}}{dt} = -\frac{\partial H}{\partial \boldsymbol{q}} \qquad \frac{d\boldsymbol{q}}{dt} = +\frac{\partial H}{\partial \boldsymbol{p}}$$

$$\rho V^* = q + \boldsymbol{a}^T V_{\boldsymbol{x}}^* - \tilde{g}(-\boldsymbol{B}^T V_{\boldsymbol{x}}^*)$$

- **Meaningful representations** suitable for the physical system
- **Interpretability** enabling connecting theory to learning
- **Sample Efficiency** enabling learning on single instances

First-Order Non-Linear Differential Equations with Deep Networks

Deep Differential Network



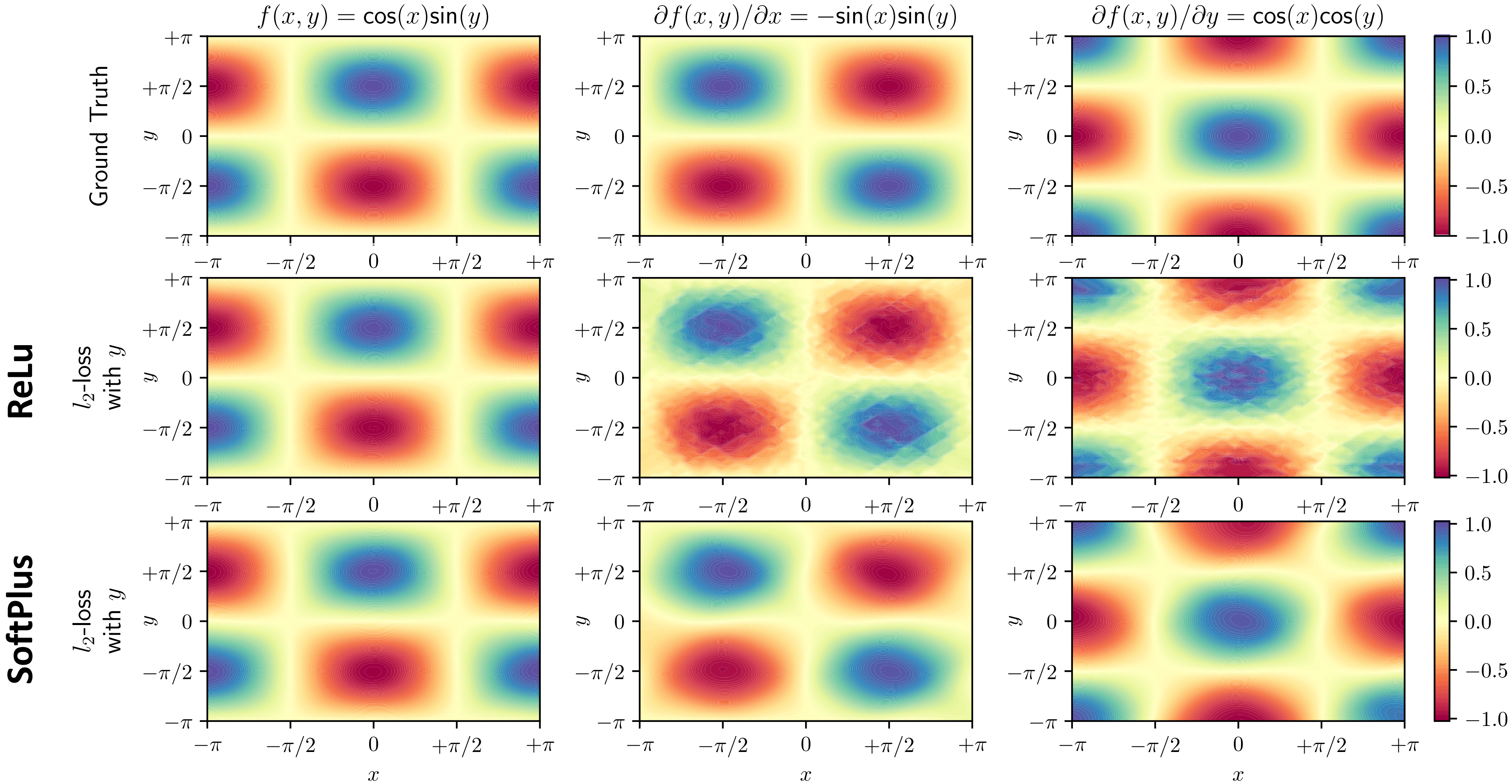
First-Order Differential Equation

$$F\left(\boldsymbol{x}, f(\boldsymbol{x}), \frac{\partial f(\boldsymbol{x})}{\partial \boldsymbol{x}}\right) = 0$$

$$\theta^* = \underset{\theta}{\operatorname{argmin}} \sum_{i=0}^{N} \left| F\left(\boldsymbol{x}_i, f(\boldsymbol{x}_i; \theta), \frac{\partial f(\boldsymbol{x}_i; \theta)}{\partial \boldsymbol{x}}\right) \right|^{\mathbf{2}}$$

with $\quad \boldsymbol{x}_i \in \Omega \subseteq \mathbb{R}^n$

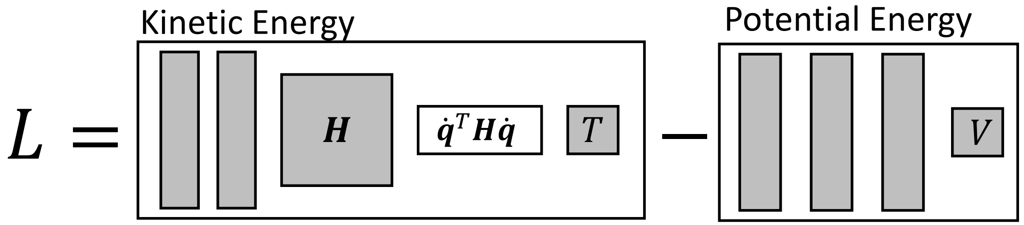
First-Order Non-Linear Differential Equations with Deep Networks



Overview

- **Introduction**
- **Deep Learning and Differential Equations**
- **Robotics Control Applications:**
  - **Deep Lagrangian Networks – Learning physically plausible Models**
  - **HJB Feedback Control – Exploiting the HJB Equation for Optimal Feedback Control**
- **Conclusion**

Deep Lagrangian Networks (DeLaN)



$$\frac{d}{dt}\frac{\partial L}{\partial \dot{\boldsymbol{q}}} - \frac{\partial L}{\partial \boldsymbol{q}} = \boldsymbol{\tau}$$

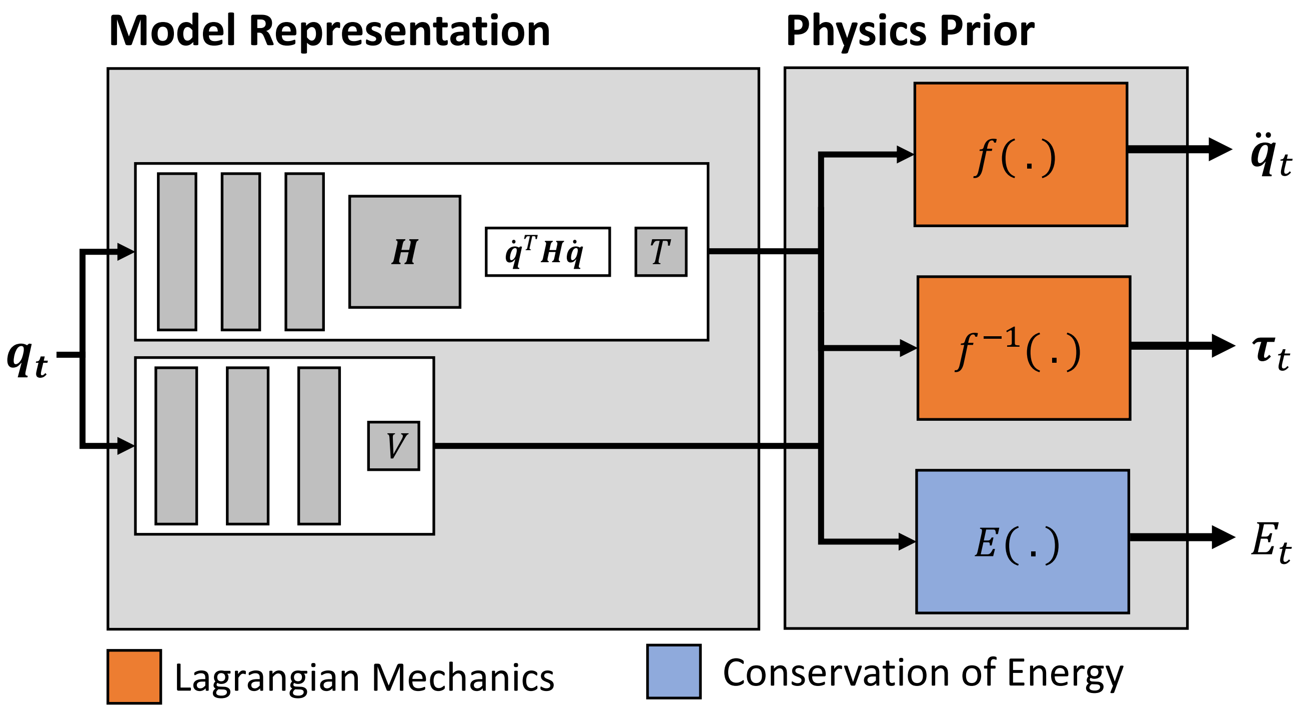
Physical plausibility, i.e., every possible parameter configuration is a mechanical system.

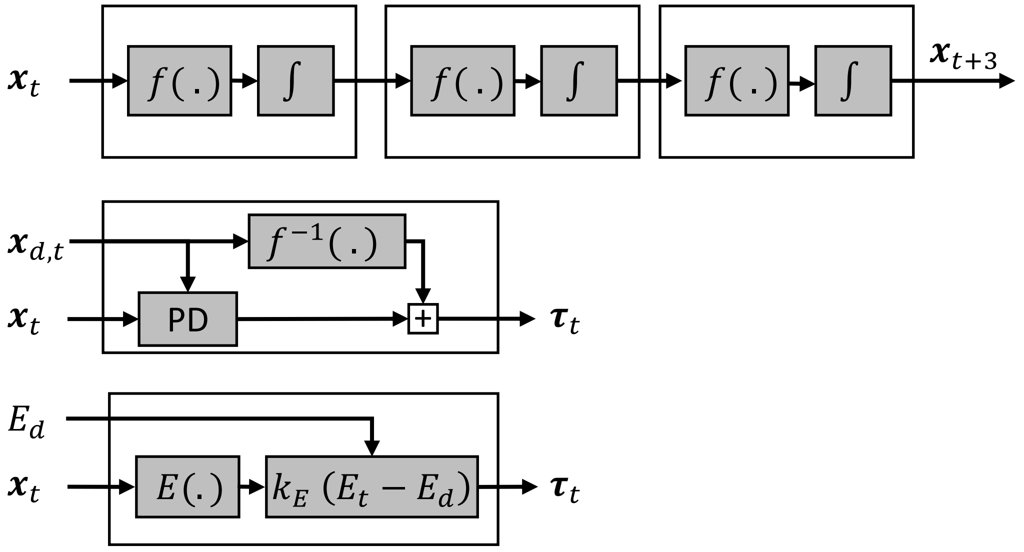
Interpretability, i.e., the learned force decomposition resembles inertial, centrifugal, Coriolis & gravitational forces

Unsupervised Learning, i.e., the learned system energies are not directly observable and can only be learned unsupervised

Multiple Models, i.e., the learned representation can be used as forward, inverse & energy model

DeLaN: Model Representation



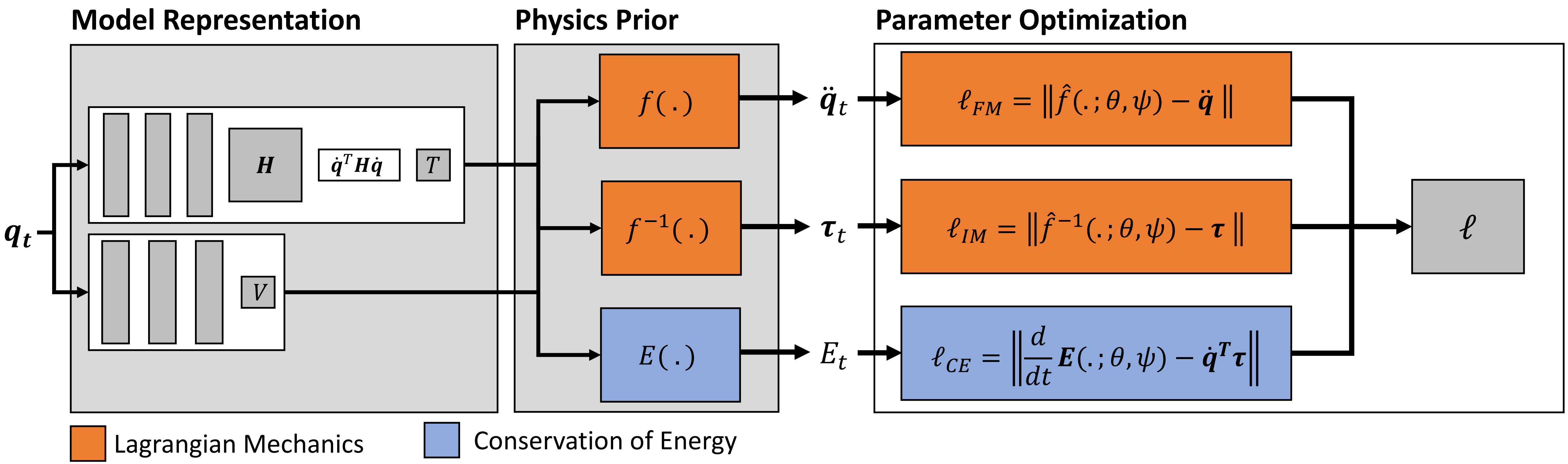


$$f(.) = \boldsymbol{H}^{-1}\left(\boldsymbol{\tau} - \dot{\boldsymbol{H}}\dot{\boldsymbol{q}} + \frac{1}{2}\left(\frac{\partial}{\partial \boldsymbol{q}}\dot{\boldsymbol{q}}^T \boldsymbol{H}\dot{\boldsymbol{q}}\right) - \frac{\partial V}{\partial \boldsymbol{q}}\right)$$ with $\boldsymbol{H}$ being p.d.

$$f^{-1}(.) = \boldsymbol{H}\ddot{\boldsymbol{q}} + \dot{\boldsymbol{H}}\dot{\boldsymbol{q}} - \frac{1}{2}\left(\frac{\partial}{\partial \boldsymbol{q}}\dot{\boldsymbol{q}}^T \boldsymbol{H}\dot{\boldsymbol{q}}\right) + \frac{\partial V}{\partial \boldsymbol{q}}$$

$$E(.) = T + V$$

DeLaN: Model Representation

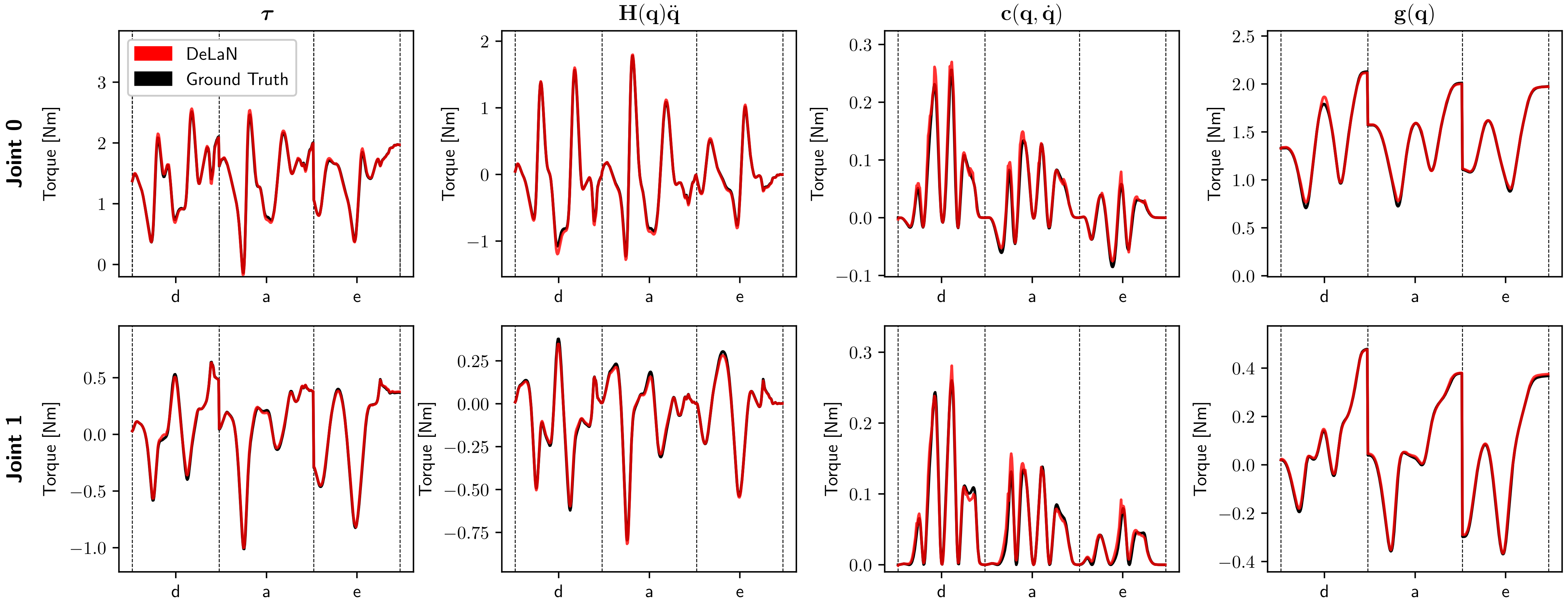


$$f(.) = \boldsymbol{H}^{-1}\left(\boldsymbol{\tau} - \dot{\boldsymbol{H}}\dot{\boldsymbol{q}} + \frac{1}{2}\left(\frac{\partial}{\partial \boldsymbol{q}}\dot{\boldsymbol{q}}^T\boldsymbol{H}\dot{\boldsymbol{q}}\right) - \frac{\partial V}{\partial \boldsymbol{q}}\right)$$ with $\boldsymbol{H}$ being p.d.

$$f^{-1}(.) = \boldsymbol{H}\ddot{\boldsymbol{q}} + \dot{\boldsymbol{H}}\dot{\boldsymbol{q}} - \frac{1}{2}\left(\frac{\partial}{\partial \boldsymbol{q}}\dot{\boldsymbol{q}}^T\boldsymbol{H}\dot{\boldsymbol{q}}\right) + \frac{\partial V}{\partial \boldsymbol{q}}$$

$$E(.) = T + V$$

DeLaN: Learned Decomposition



DeLaN: Learned Energy Control

DeLaN | **Analytic Model** | **System Identification**

DeLaN: Learned Energy Control

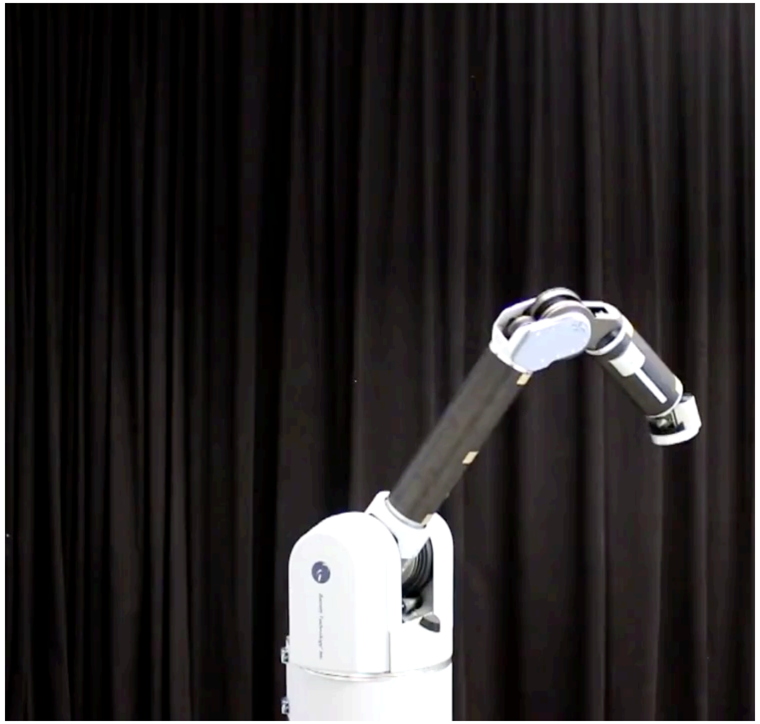
DeLaN

Analytic Model

System Identification

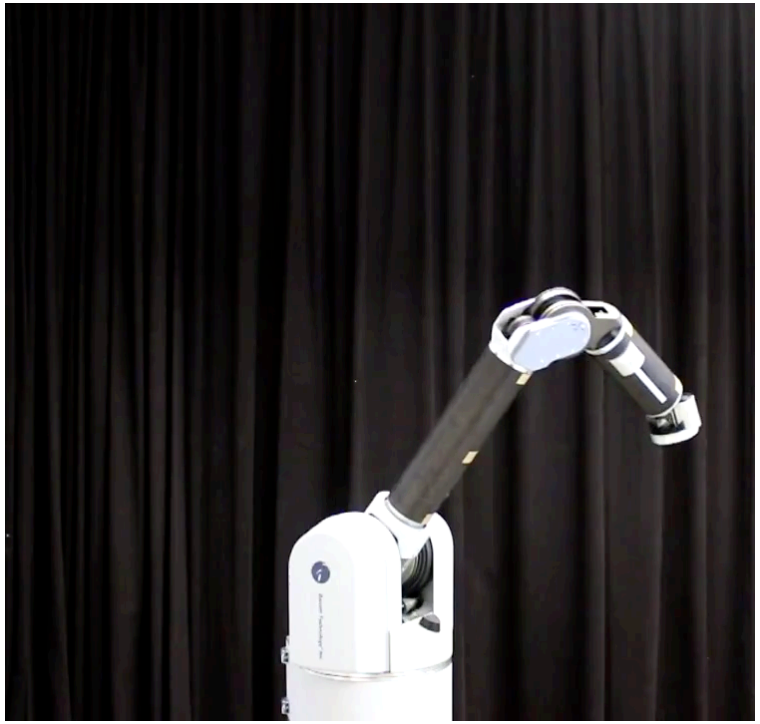
DeLaN: Online Learned Tracking Control

Training

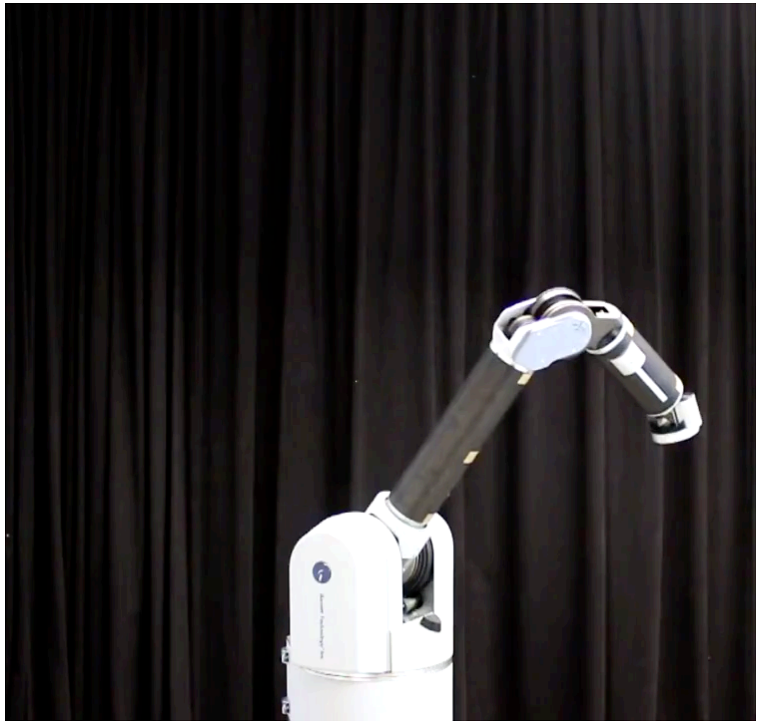


1x

Testing

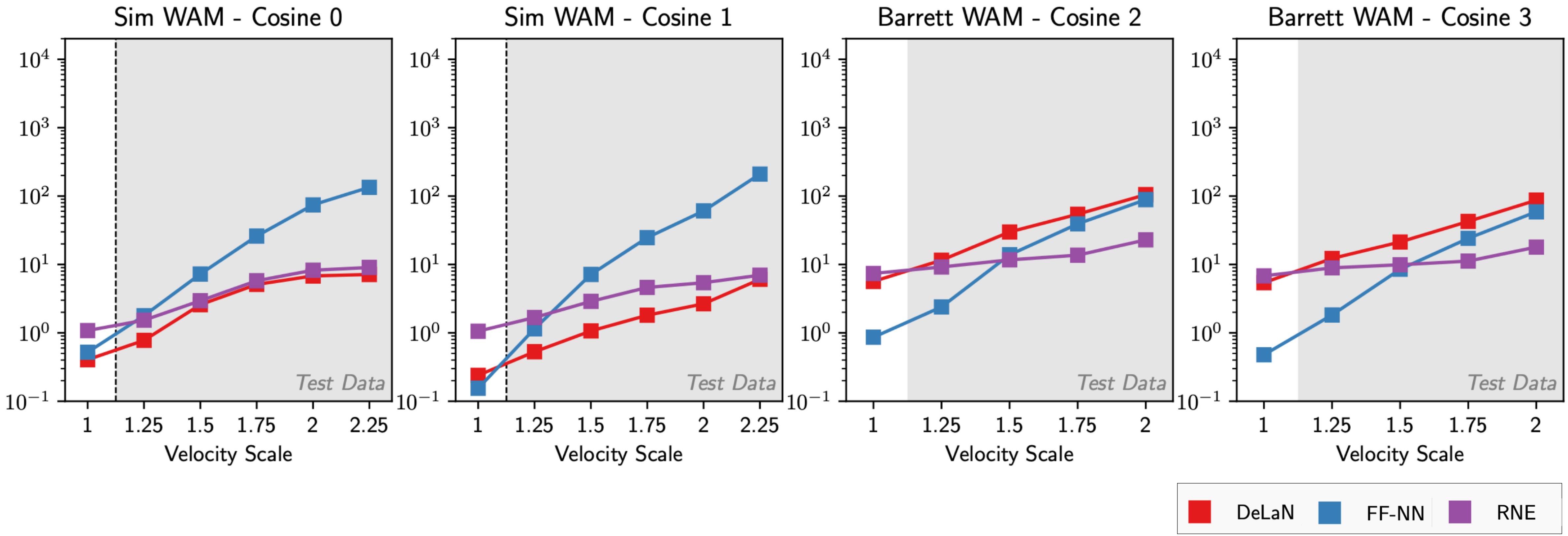


1.5x



2x

DeLaN: Online Learned Tracking Control



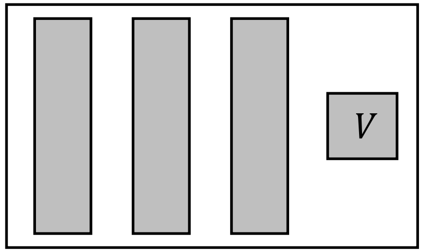
Overview

- **Introduction**
- **Deep Learning and Differential Equations**
- **Robotics Control Applications:**
  - **Deep Lagrangian Networks – Learning physically plausible Models**
  - **HJB Feedback Control – Exploiting the HJB Equation for Optimal Feedback Control**
- **Conclusion**

HJB-Feedback Control: Problem Statement

Value Function

$$V^* =$$



$$\rho V^* = q + \boldsymbol{a}^T V_{\boldsymbol{x}}^* - \tilde{g}(-\boldsymbol{B}^T V_{\boldsymbol{x}}^*)$$

$$\pi^* = \nabla \tilde{g}(-\boldsymbol{B}^T V_{\boldsymbol{x}}^*)$$

Optimal Feedback Control, i.e., optimal policy that does not require replanning or sampling of actions.

Action Constraints, i.e., principled cost functions that shape the optimal policy to be action limited

Interpretability, i.e., capable of linking the concepts of stability and robustness to the learned solution

Assumptions:

- mechanical System with holonomic constraints
- separable cost function, i.e., $c = q(\boldsymbol{x}) + g(\boldsymbol{u})$
- strictly convex action cost $g(\boldsymbol{u})$

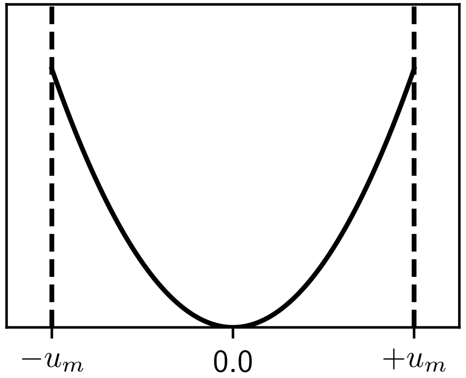
$\tilde{g}$ is the convex conjugate of $g(\boldsymbol{u})$

HJB-Feedback Control: Principled Cost for Action Constraints

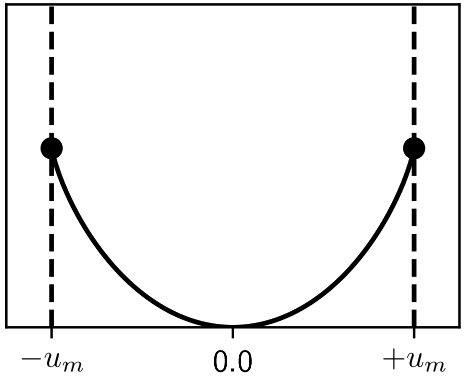
Action Cost:

$g(\boldsymbol{x})$

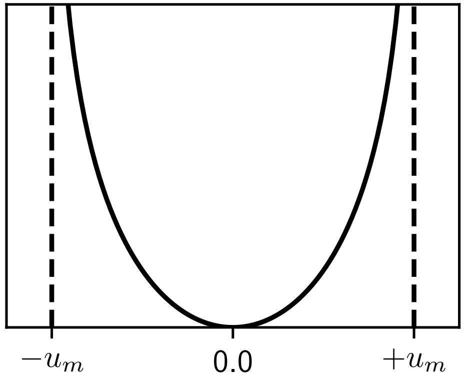
$\boldsymbol{u}^T \boldsymbol{R} \boldsymbol{u}$



$\frac{\boldsymbol{u}+\mathbf{1}}{2}^T \log\frac{\boldsymbol{u}+\mathbf{1}}{2} + \frac{\mathbf{1}-\boldsymbol{u}}{2}^T \log\frac{\mathbf{1}-\boldsymbol{u}}{2} - \log\frac{\mathbf{1}}{\mathbf{2}}$



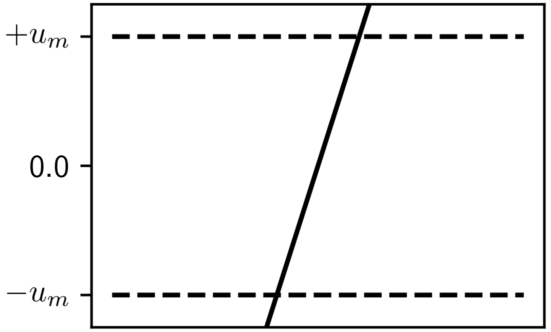
$-\mathbf{1}^T \log(\cos(\boldsymbol{u}))$



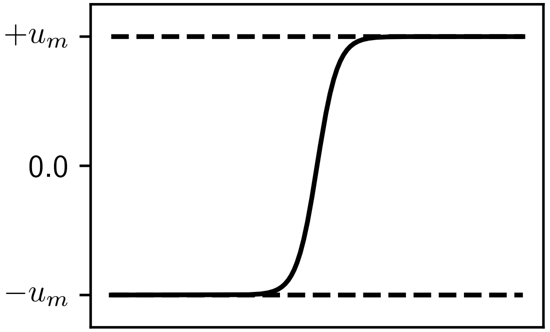
Optimal Policy:

$\nabla \tilde{g}(-\boldsymbol{B}^T V_{\boldsymbol{x}}^*)$

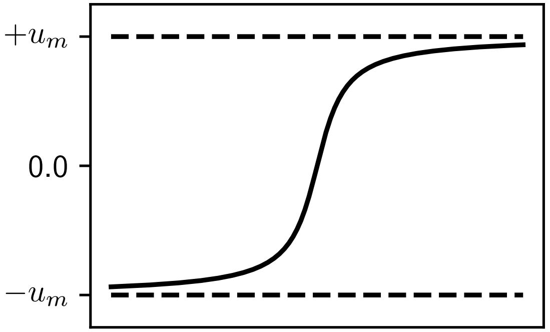
$-\boldsymbol{R}^{-1} \boldsymbol{B}^T V_{\boldsymbol{x}}^*$



$\tanh(-\boldsymbol{B}^T V_{\boldsymbol{x}}^*)$



$\mathrm{atan}(-\boldsymbol{B}^T V_{\boldsymbol{x}}^*)$



HJB-Feedback Control: Curriculum Learning

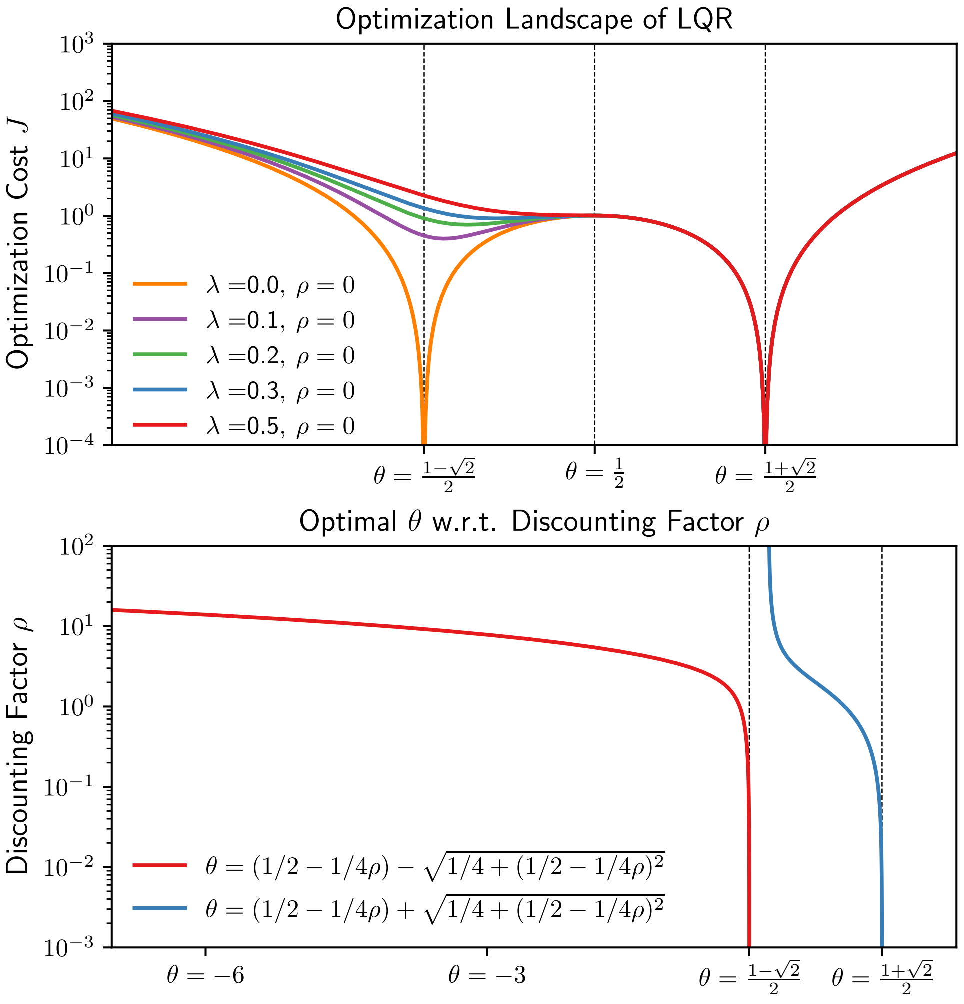
The solution to the optimization problem is only unique and stable if the boundary condition

$$(\boldsymbol{a} + \boldsymbol{B}\boldsymbol{u}^*)^T \, \boldsymbol{\eta}(\overline{\boldsymbol{x}}) \leq 0 \quad \text{for} \quad \overline{\boldsymbol{x}} \in \partial\Omega$$

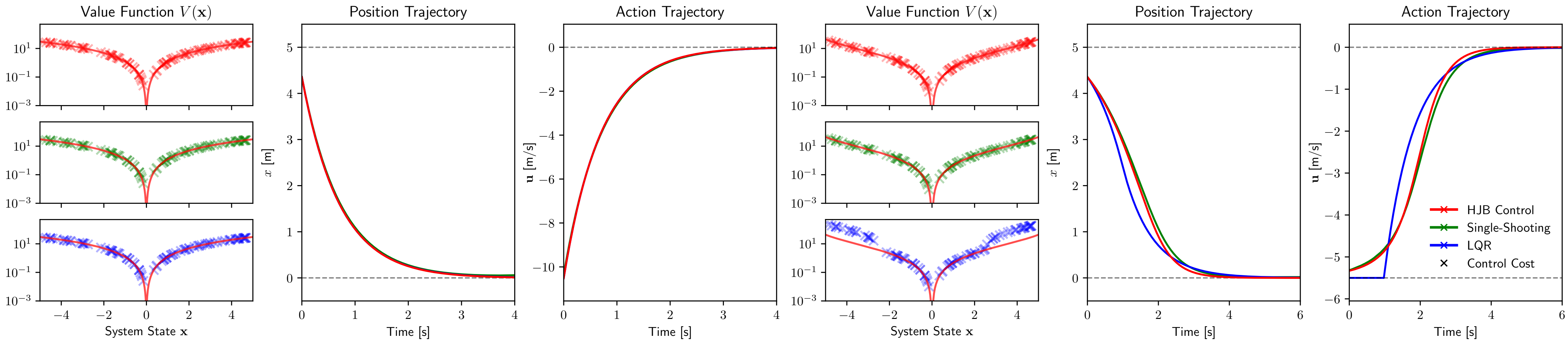
$$\left(\boldsymbol{a} + \boldsymbol{B}\nabla\tilde{g}(-\boldsymbol{B}^T V_{\boldsymbol{x}}^*)\right)^T \boldsymbol{\eta}(\overline{\boldsymbol{x}}) \leq 0$$

is fulfilled. Therefore, the naïve optimization does not necessarily learn the optimal and stable V^*.

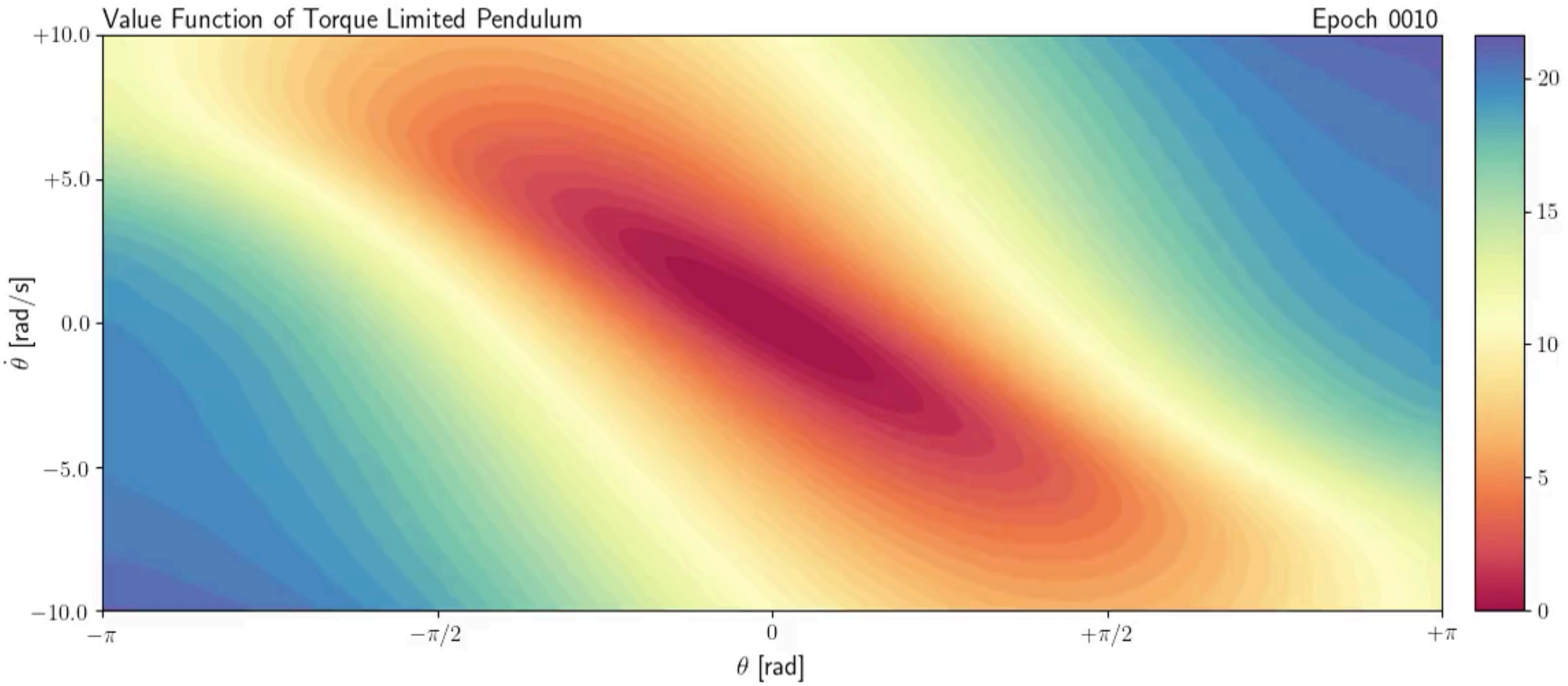
=> How can one ensure coherent solutions given local boundary constraints and local function approximators?



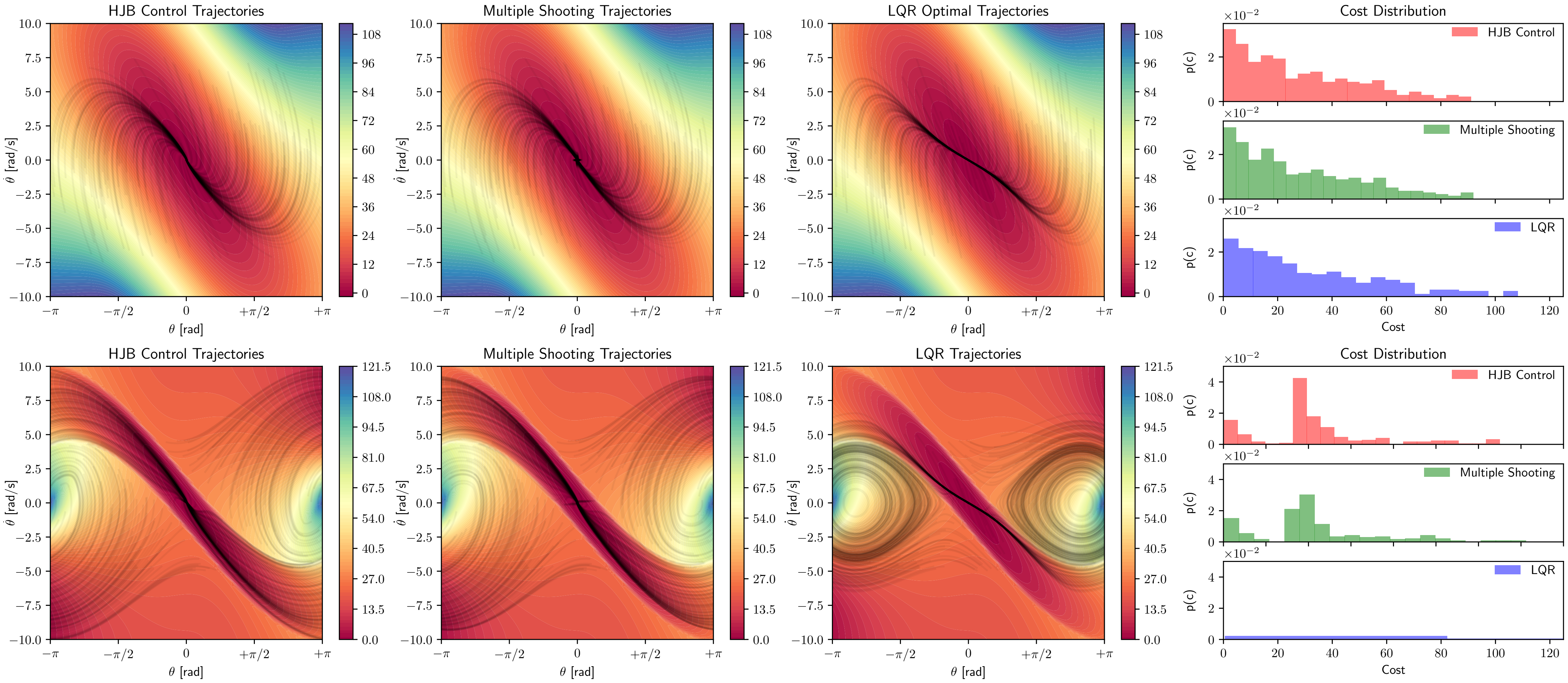
HJB-Feedback Control: 1d Integrator



HJB-Feedback Control: Torque Limited Pendulum



HJB-Feedback Control: Torque Limited Pendulum



Summary

- **Differential Equations as Model Prior for Deep Networks:**
  (1) Using differential equations as model prior enables learning of meaningful & interpretable representations
  (2) Combining differential equations and deep networks is natural as these networks are fully differentiable

- **Deep Lagrangian Networks:**
  (1) Introduced physics prior to deep learning to enable unsupervised learning of the underlying structure.
  (2) Demonstrated that these deep networks models can be used for real-time energy control with up to 500Hz.

- **HJB-Feedback Control:**
  (1) Derived principled action cost from the Hamilton-Jacobi-Bellman differential equation.
  (2) Demonstrated that solving the HJB with a differential network obtains an optimal feedback policy

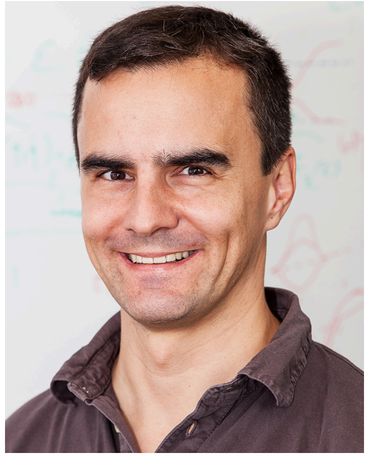
Inductive Biases for Robot Control

Michael Lutter
IAS TU Darmstadt

Boris Beluosov
IAS TU Darmstadt

Debora Clever
ABB AG

Kim Listmann
ABB AG



Jan Peters
IAS TU Darmstadt

Students:
Johannes Silberbauer, Markus Semmler, Jonas Jäger, Kai Plöger, Christian Ritter, Jasper Süß, Janosch Moos, Kay Hansel

Funding:
ABB AG, NVIDIA, European Research Council