Lecture 14: Outlook and Research Insights

Oliver Wallscheid

PADERBORN UNIVERSITY

Table of contents

Recap: optimal control and constraints

Real-world systems are always subject to certain state constraints $\mathcal{X}$ and input limitations $\mathcal{U}$. Violating those can lead to safety issues.

$$
\begin{aligned}
v_k^* &= \max_{\boldsymbol{u}_k} \sum_{i=0}^{N_\mathrm{p}} \gamma^i r_{k+i+1}(\boldsymbol{x}_{k+i}, \boldsymbol{u}_{k+i}), \\
\text{s.t.} \quad \boldsymbol{x}_{k+i+1} &= \boldsymbol{f}(\boldsymbol{x}_{k+i}, \boldsymbol{u}_{k+i}), \quad \boldsymbol{x}_{k+i} \in \mathcal{X}, \quad \boldsymbol{u}_{k+i} \in \mathcal{U}.
\end{aligned}
\tag{14.1}
$$

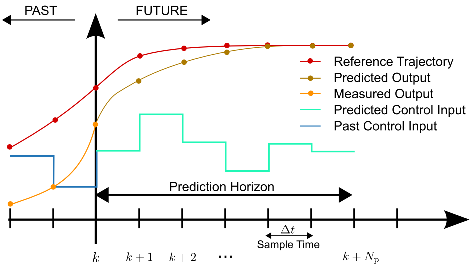
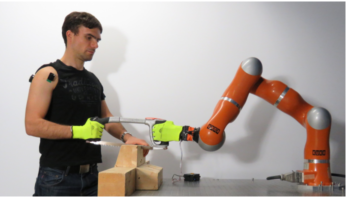


Fig. 14.1: MPC scheme (source: www.wikipedia.org, by Martin Behrendt CC BY-SA 3.0)

Application examples with safety-relevant constraints

Collaborative robot control (source: www.wikipedia.org, CC BY-SA 4.0)



Autonomous car driving (source: www.wikipedia.org, CC BY-SA 4.0)

Energy system control

Medication control (source: www.wikipedia.org, CC BY-SA 4.0)

Safety levels

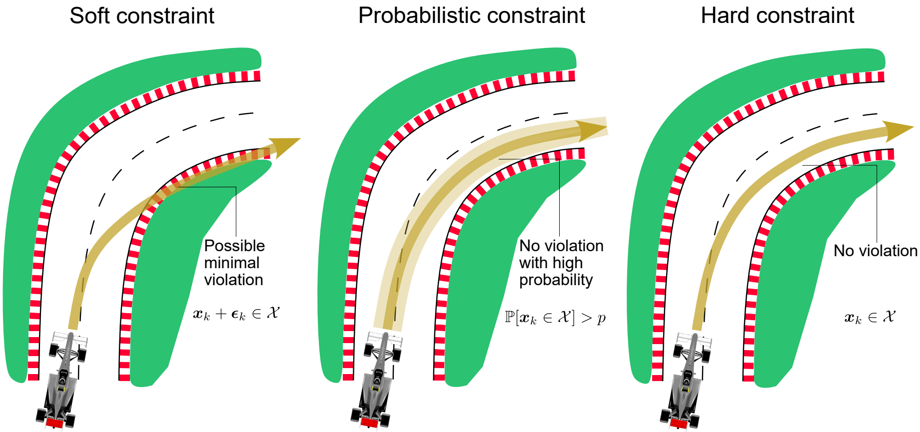
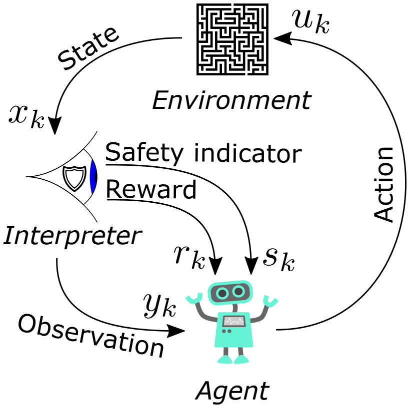
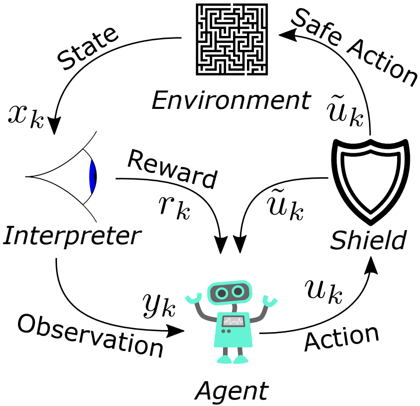


Fig. 14.2: Different levels of safety (derived from L. Brunke et al., *Safe Learning in Robotics: From Learning-Based Control to Safe Reinforcement Learning*, Annual Review of Control, Robotics, and Autonomous Systems, 2022)

Bird's eye view on RL concepts integrating safety



(a) Safety critic: add a critic which indicates to which extent the current data sample fits to a safe situation



(b) Safety shield: use a priori or learned model knowledge of the environment to make predictions identifying actions leading to unsafe situations

Achievable safety levels and model knowledge

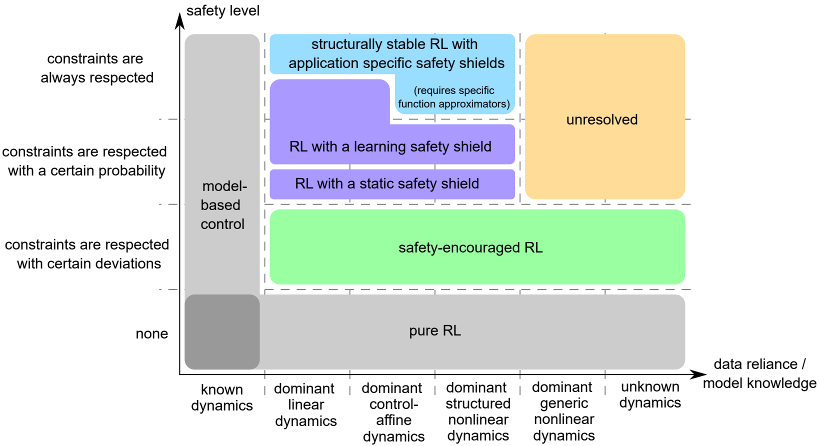
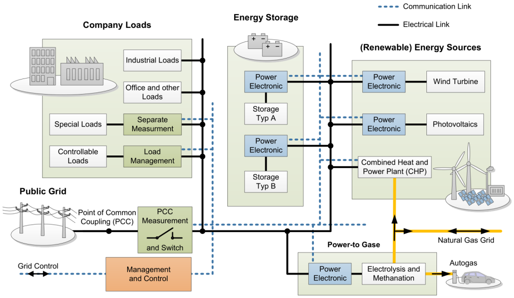
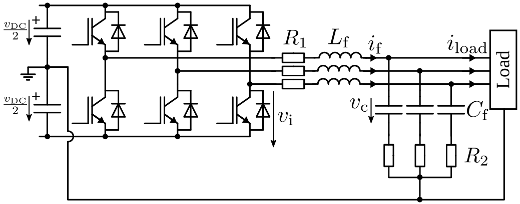


Fig. 14.3: Safety and model knowledge map (derived from L. Brunke et al., *Safe Learning in Robotics: From Learning-Based Control to Safe Reinforcement Learning*, Annual Review of Control, Robotics, and Autonomous Systems, 2022)

Energy system control application



(a) Example microgrid that can be emulated in the LEA Microgrid Laboratory.



(b) Application under investigation: Three-phase grid-forming inverter disturbed by stochastic load

Reference tracking with disturbance rejection

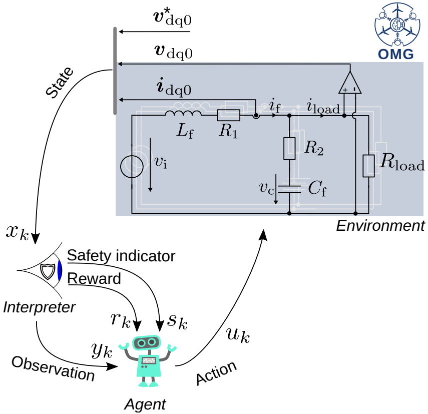


Fig. 14.4: Simulation setting with environment modeled using OpenModelica Microgrid Gym

- Cont. state- and actionspace
- Deep deterministic policy gradient agent
- Gird-forming inverter
- Stochastic load acts as disturbance
- State per phase: $\boldsymbol{x_k} = [i_\mathrm{f}, v_\mathrm{C}]$, $v_\mathrm{i} = v_\mathrm{DC} \cdot u_k$
- $r_k = f(v_\mathrm{C}, v^*, i_\mathrm{f}) \in [1, -0.75]$
- $s_k = -1$, if limit $(i_\mathrm{f}$ or $v_\mathrm{C})$ is exceeded

Reward design for grid-forming inverter

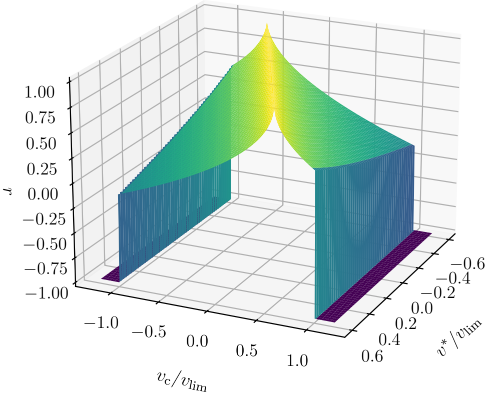


Fig. 14.5: Reward function 14.2 for different reference and measured voltages and currents below nominal current

- Three cases, depending on operation point

$$r = \begin{cases} \mathrm{MRE}(v_\mathrm{C}, v^*), & \text{A} \\ \mathrm{MRE}(v_\mathrm{C}, v^*) + f(i_\mathrm{f}), & \text{B} \\ -1, & \text{C} \end{cases} \tag{14.2}$$

- A $v_\mathrm{C} \leq v_\mathrm{lim} \wedge i_\mathrm{f} \leq i_\mathrm{nom}$
- B $v_\mathrm{C} \leq v_\mathrm{lim} \wedge i_\mathrm{nom} \leq i_\mathrm{f} \leq i_\mathrm{lim}$
- C otherwise
- Linear punishment term $f(i_\mathrm{f})$

Reference tracking with disturbance rejection using saftey shield

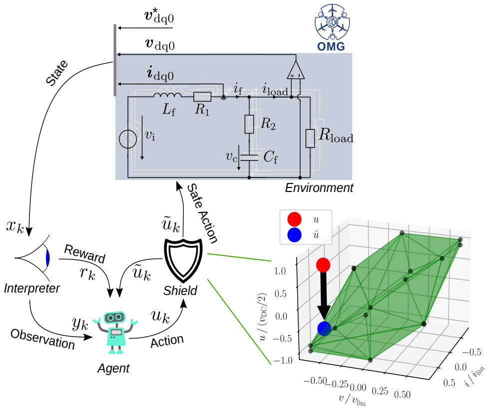


Fig. 14.6: Safety shield based on feasible set

- Safety shield: Ensure that action does not cause state limit violation in future system trajectories
- Such a state action pair is called feasible
- Calculation of feasible set requires a model
- Training data can be utilized to identify model
- Here, recursive least squares (RLS) is applied

Saftey shield based on feasible set - proof of concept (1)

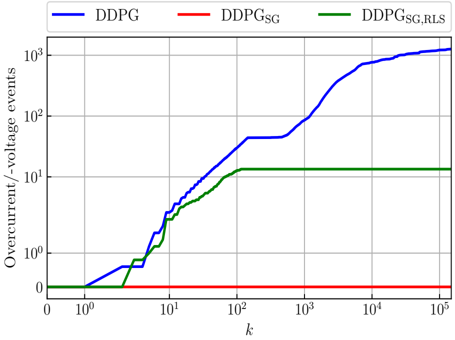


Fig. 14.7: Accumulated unsafe events (overcurrent/-voltage) per trainingstep k

- Three different approaches
- DDPG: Agent without safety shield
- DDPG$_{\mathrm{SG}}$: Agent with safety shield using perfect a priori knowledge
- DDPG$_{\mathrm{SG,RLS}}$: Agent with safety shield without a priori knowledge, identifying model using RLS
- Five agents trained per approach
- Results in D. Weber et al., *Safe Reinforcement Learning-Based Control in Power Electronic Systems*, 2023

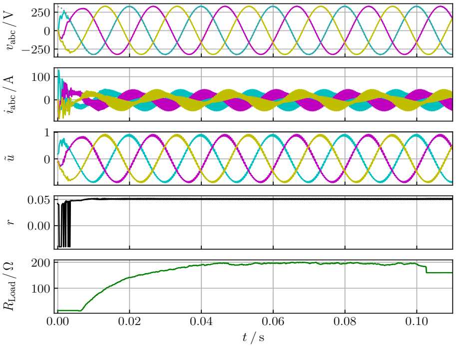


Fig. 14.8: Blackstart after training using $\mathrm{DDPG}_{\mathrm{SG,RLS}}$

- $\mathrm{DDPG}_{\mathrm{SG,RLS}}$ agent trained for 150000 steps
- R_{Load} changes every step based on random process
- Additional events – load steps and drifts – trigged randomly

Table of contents

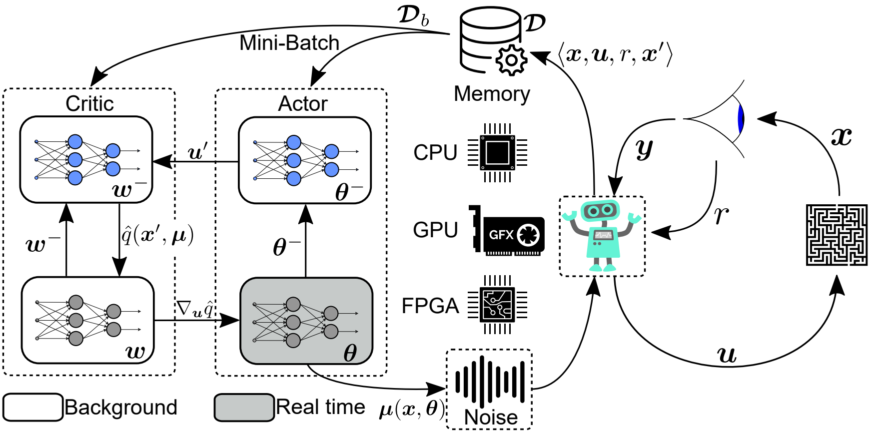
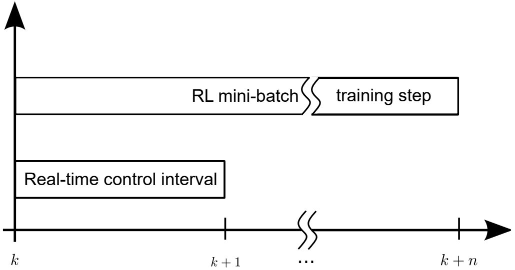
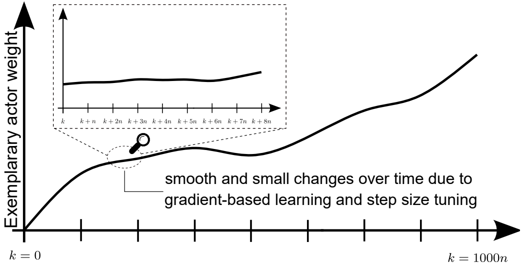


Fig. 14.9: DDPG implementation example (derivative work of Fig. 1.1 and wikipedia.org, CC0 1.0)

Real-time implementation aspects (2)



(a) Real-time control requirement vs. learning time



(b) Typical evolution of RL parameter weights during learning

Application example: deep Q direct torque control

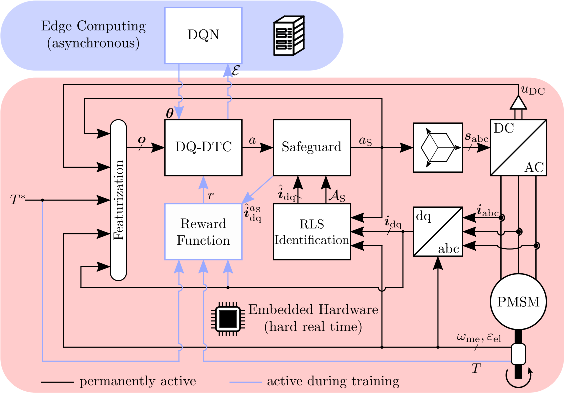


Fig. 14.10: Deep Q direct torque control schematic

- The DQ-DTC is basically a DQN
- Sampling time of the plant system is $T_s = 50\,\mu s$
- DQN inference, safeguarding and system identification must fit into T_s
- Source: M. Schenke et al., *Finite-Set Direct Torque Control via Edge Computing-Assisted Safe Reinforcement Learning for a Permanent Magnet Synchronous Motor*, 2023

Fast neural network inference

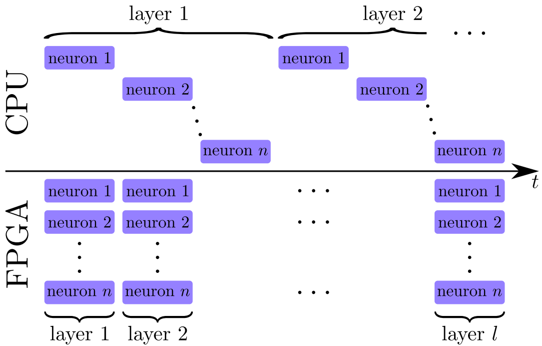


Fig. 14.11: Conceptual comparison of CPU and FPGA evaluation of a neural network

- Each neuron has the same job $y_{n,l+1} = f(\boldsymbol{y}_l^\top \boldsymbol{w}_{n,l} + b_{n,l})$
- CPU must evaluate each neuron sequentially
- FPGA can evaluate each neuron at the same time
- Maximum number of parallel computations is limited

Edge reinforcement learning

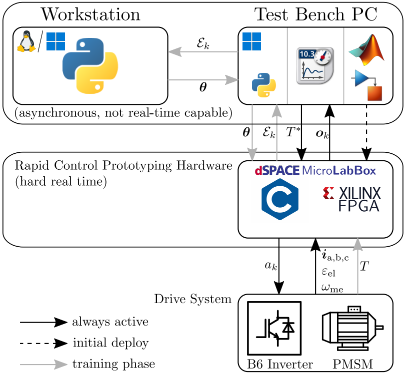


Fig. 14.12: Our edge reinforcement learning pipeline

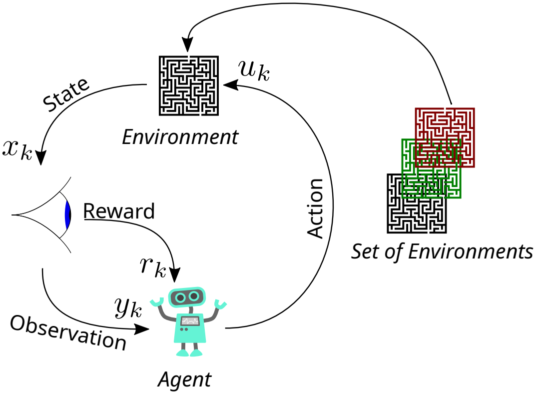
- Backward pass / learning steps are outsource to workstation
- Communication between test bench and workstation is based on TCP/IP
- Backward pass is generic and has no time constraints $\rightarrow$ low application effort

Demonstration video

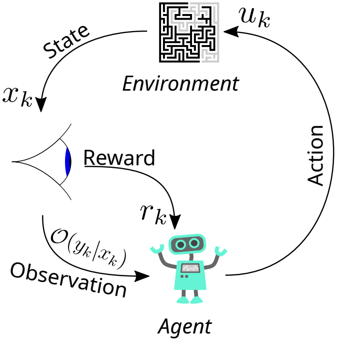
Youtube link: Coffee machine vs. deep Q direct torque control

Table of contents

Meta reinforcement learning - the setting (1)



(a) General problem class is similar, environments only differ in some characteristics, the agent could transfer learned behavior



(b) Solution approach: treat the environment as partially observable, distinguishing details are not directly available

Meta reinforcement learning - the setting (2)

- The agent must have some mechanism that allows adaptation to the specific environment
- This means, the distinguishing details must be extracted in some way
- Usually, they can be retrieved from a larger set of observations

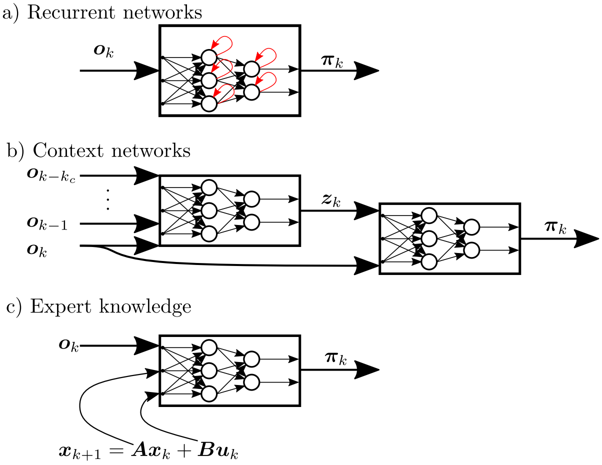


Fig. 14.13: Different concepts of meta learning

Usage in electric drive control: classical agent

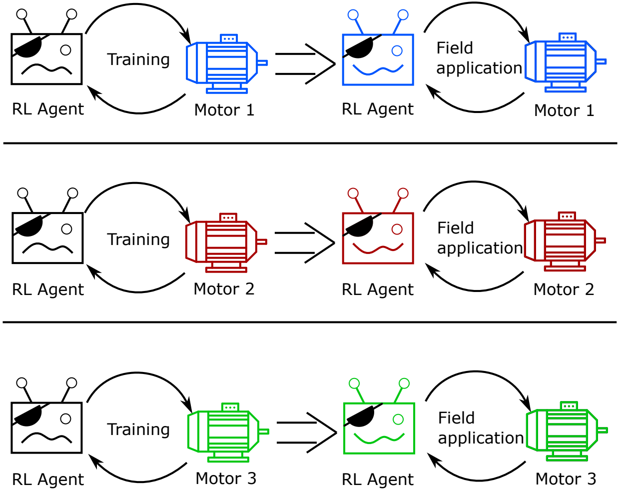


Fig. 14.14: Each agent must be trained individually $\rightarrow$ huge effort

Usage in electric drive control: meta agent

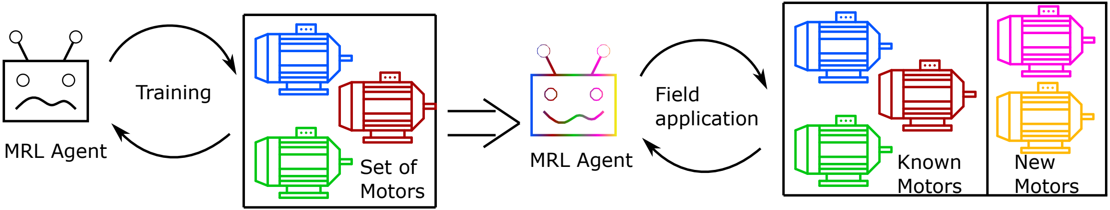


Fig. 14.15: One agent to control them all → effort is limited and independent of the number of controlled environments

Our setup

- Make use of context network
- Generate context $\boldsymbol{z}$ with a fix set of observations $\rightarrow \boldsymbol{z} = \text{const.}$
- Source: D. Jakobeit et al., *Meta-Reinforcement Learning-Based Current Control of Permanent Magnet Synchronous Motor Drives for a Wide Range of Power Classes*, IEEE TPEL, 2023

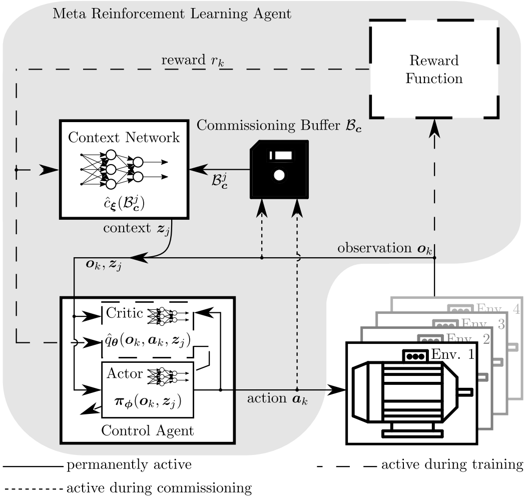
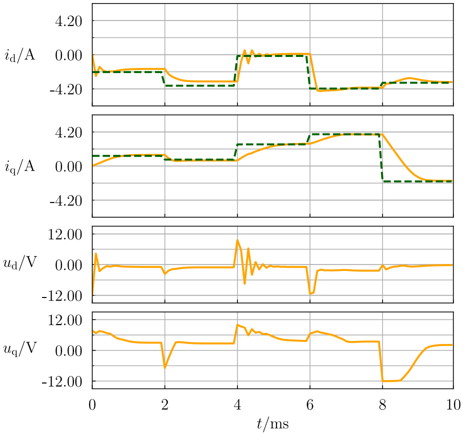
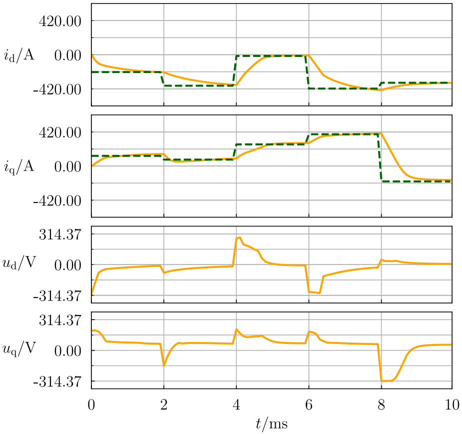


Fig. 14.16: A meta learning concept that we implemented successfully

Evaluation on (very) different motors



(a) Current control on a PMSM with low rated power



(b) Current control on a PMSM with high rated power

Summary

- Application of RL on technical systems comes with many challenges, e.g.,
  - Safety limits,
  - Real-time / computational constraints,
  - Varying and/or partially unknown environments.
- Real-world implementations often require more than bare RL algorithms, e.g.,
  - Integration of available a priori expert knowledge,
  - Combination with model-based control engineering tools.
- Ideal integration of data-driven RL solutions together with expert-based control engineering parts is subject to many open research question.