UNIVERSITÄT PADERBORN

LEA

POWER ELECTRONICS AND ELECTRICAL DRIVES

REINFORCEMENT LEARNING

FOR ELECTRIC MOTOR CONTROL

GYM-ELECTRIC-MOTOR TOOLBOX

FINAL PRESENTATION OF PROJECT GROUP
SUMMER SEMESTER 2019

ARNE TRAUE AND GERRIT BOOK

Motivation

Electric Motor Control

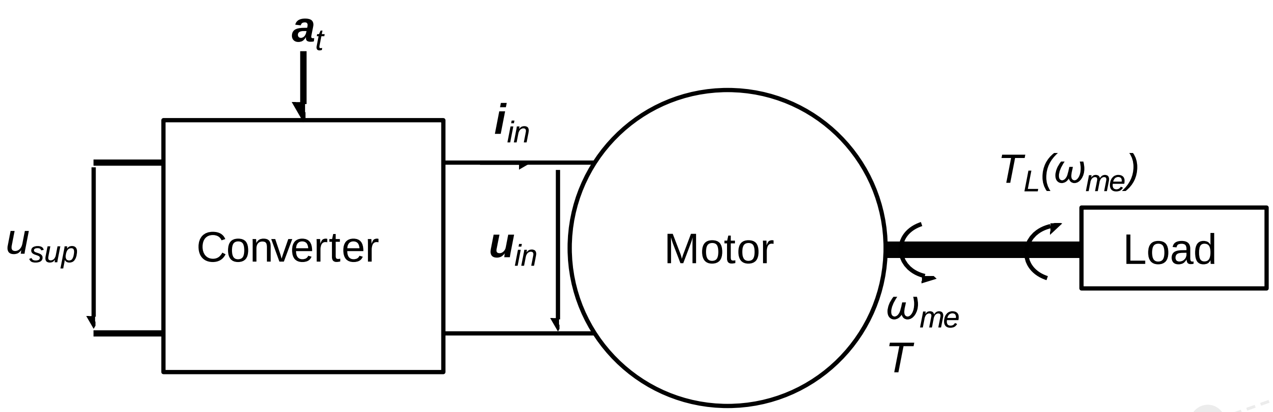
Machine Learning

Reinforcement Learning (RL) Motor Control

- ✓ Proof of concept by Maximilian Schenke
- ✓ Toolboxes with RL algorithms Keras-RL, Tensorforce, OpenAI Baselines

Environments for simulation of electric motors in Python

gym-electric-motor (GEM) toolbox

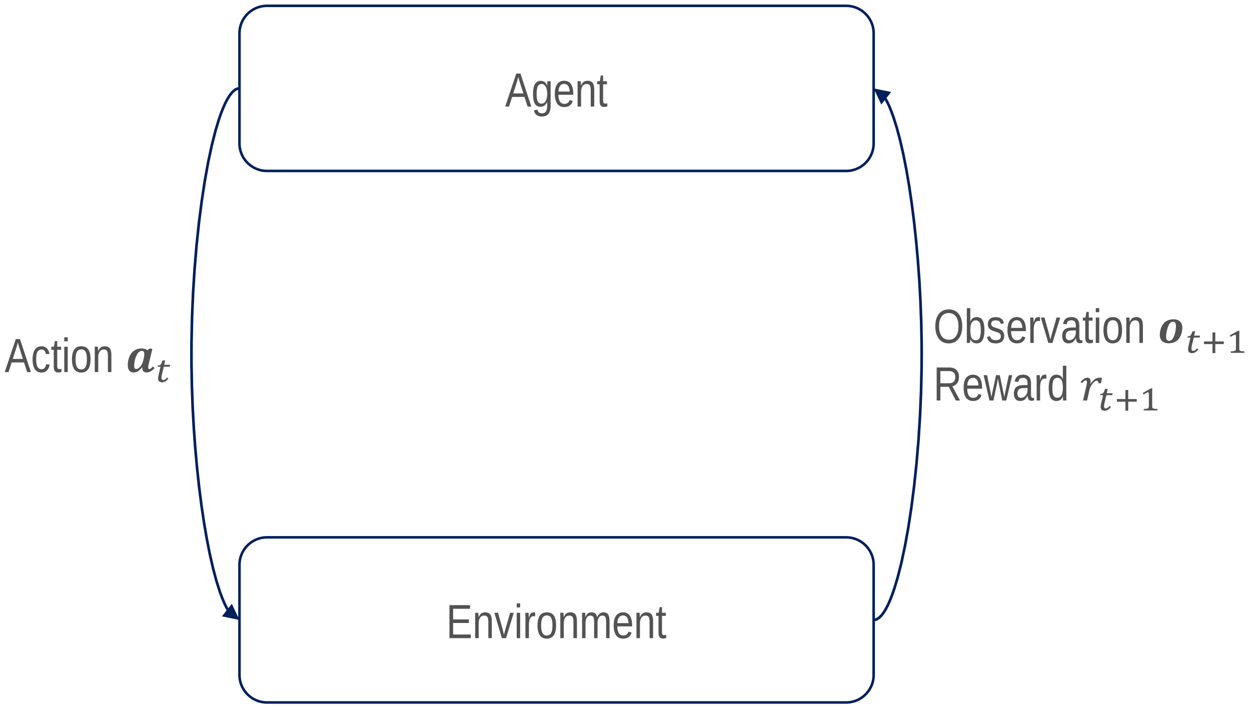


Reinforcement Learning: Cartpole example

https://www.youtube.com/watch?v=46wjA6dqxOM

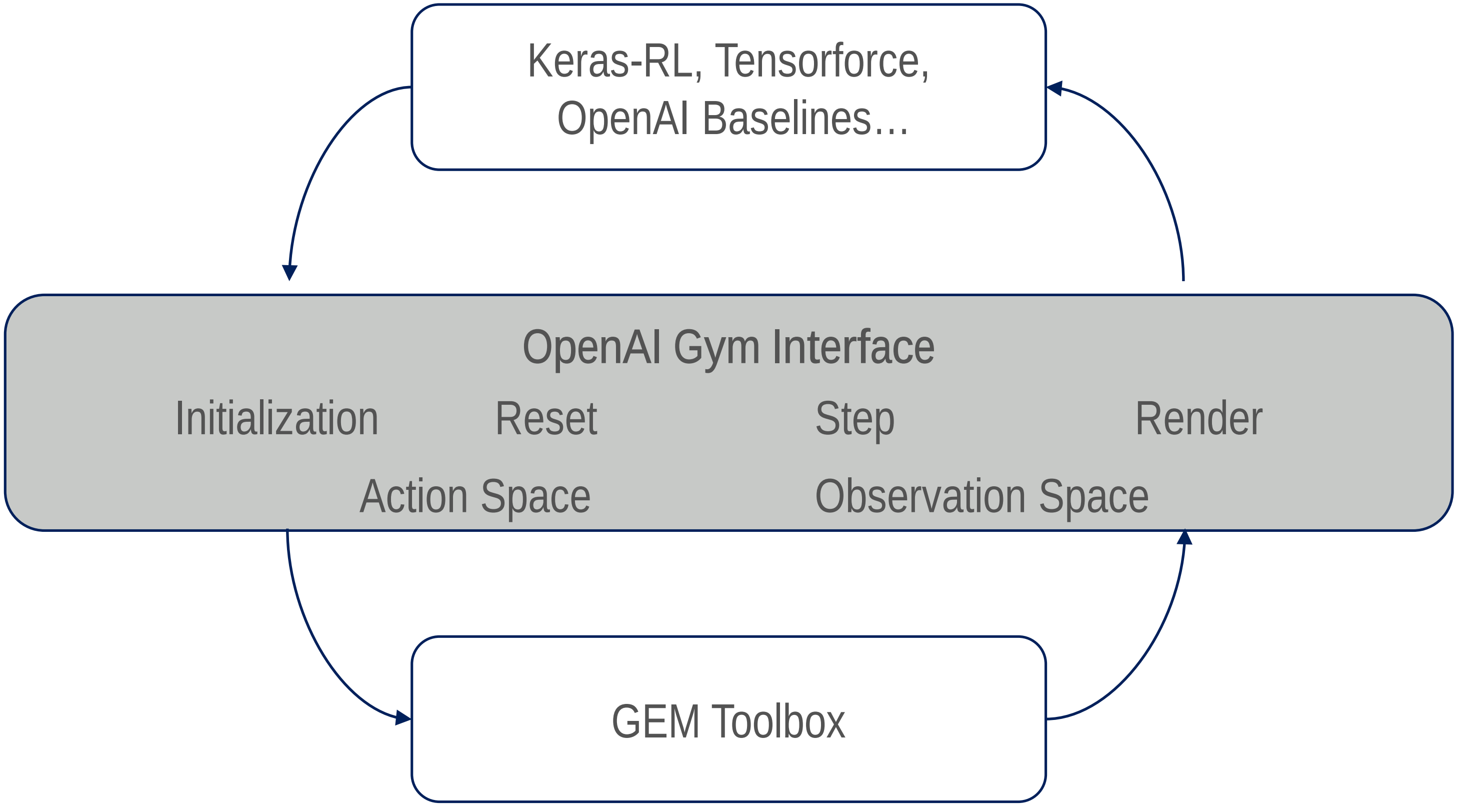
Content

Reinforcement Learning



Aim: Maximize cumulative reward

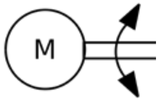
Reinforcement Learning



Content

Requirements to the Toolbox

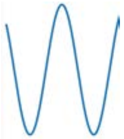
OpenAI Gym Interface
To be applicable to many Python RL-toolboxes



Accurate Technical Models
For close-to-real-world simulation

$

Reward Function
Defines the optimization criteria



Reference Signal Generation
To train a generally applicable policy

Visualization
To show the training process

Content

OpenAI Gym Interface

Action Space

Set of control commands to the converter

- Discrete switching command
- Continuous duty cycle

Observation Space

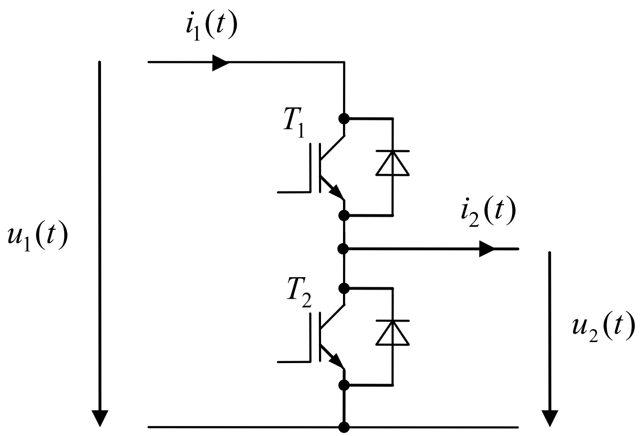
Concatenation of the motor quantities and reference quantities

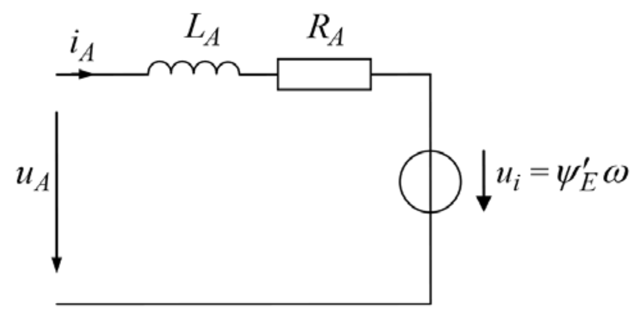
- Continuous values in $[-\mathbf{1}, +\mathbf{1}]^{\boldsymbol{n}}$
- $\boldsymbol{n}$: Dimension of the observation space

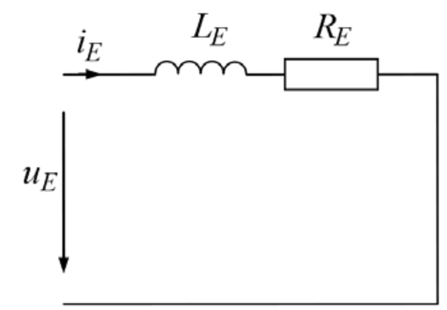
Example observation for a speed controlled series DC motor:

$$(\omega, T, i, u_{in}, u_{sup}, \omega^*)$$

Technical DC Models







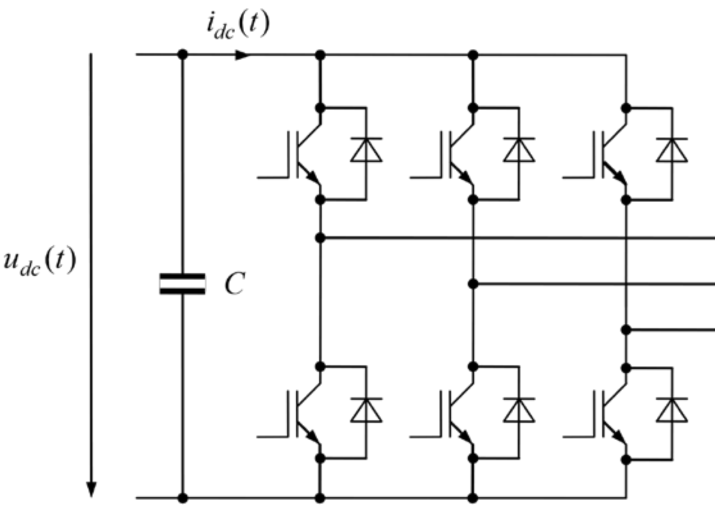
	i	u
1QC	≥ 0	≥ 0
2QC	$\geq \cup \leq 0$	≥ 0
4QC	$\geq \cup \leq 0$	$\geq \cup \leq 0$

Externally Excited Motor
Permanently Excited Motor
Series Motor
Shunt Motor

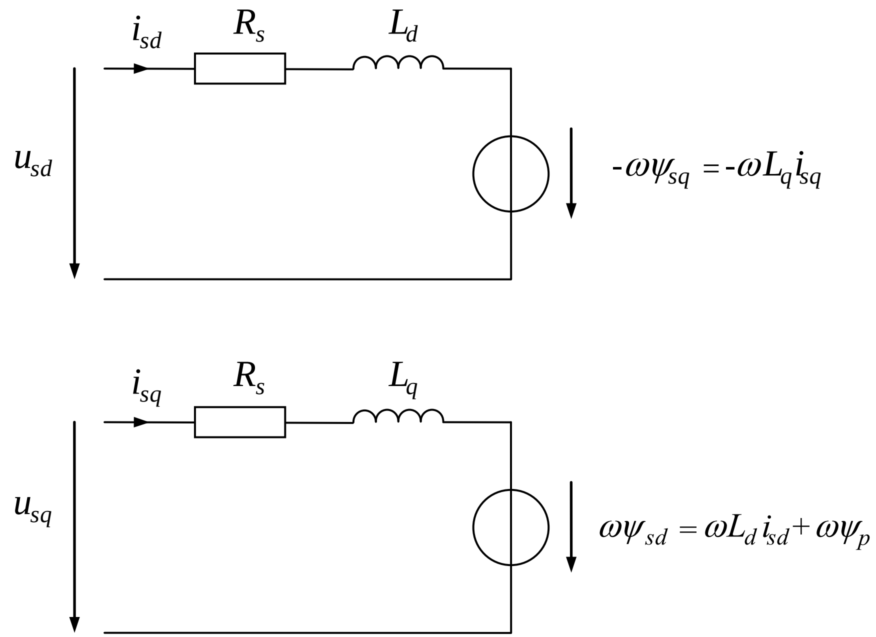
$$T_L(\omega_{me}) = \text{sign}(\omega)(c\omega^2 + b|\omega| + a)$$

Technical Three Phase Models

	i	u
B6-bridge	≥ ∪ ≤ 0	≥ ∪ ≤ 0



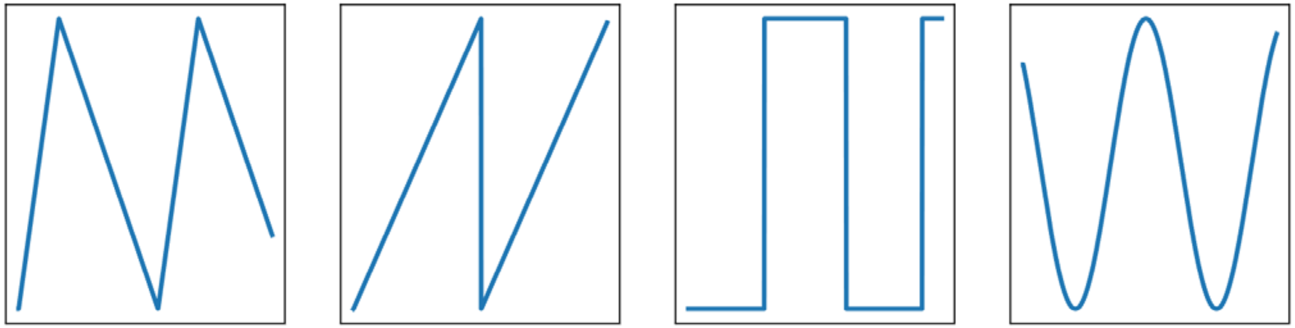
Permanent Magnet Synchronous Motor



$$T_L(\omega_{me}) = \mathrm{sign}(\omega)(c\omega^2 + b|\omega| + a)$$

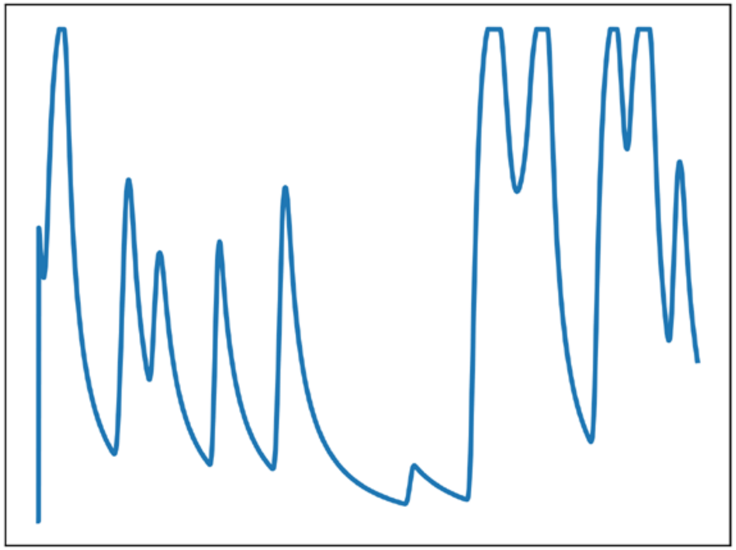
Reference Generation

Standard shapes



Random shape

- Random Fourier spectrum for input voltage
- Limit bandwidth
- Transform to time domain
- Simulate motor
- Clip references to nominal values



Rewards

Reward weights

w_k for each environment state

Reward functions

Negative rewards

Weighted sum of absolute error (WSAE)

$$r_t = -\sum_{k=0}^{N} w_{\{k\}} |s_{\{k\}t} - s^*_{\{k\}t}|$$

Weighted sum of squared error (WSSE)

$$r_t = -\sum_{k=0}^{N} w_{\{k\}} \left(s_{\{k\}t} - s^*_{\{k\}t}\right)^2$$

Positive rewards

Shifted weighted sum of absolute error (SWSAE)

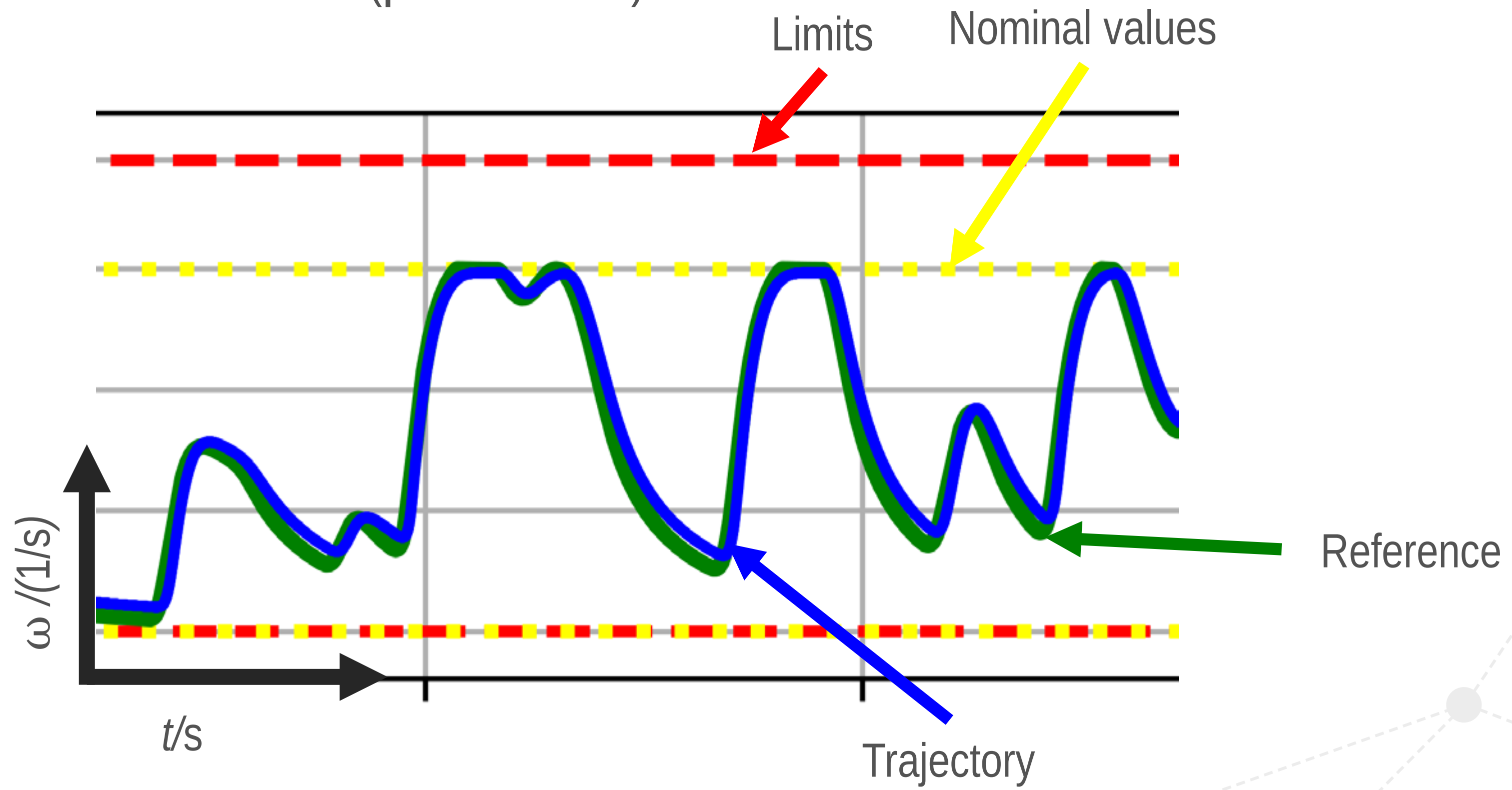
$$r_t = 1 - \sum_{k=0}^{N} w_{\{k\}} |s_{\{k\}t} - s^*_{\{k\}t}|$$

Shifted weighted sum of squared error (SWSSE)

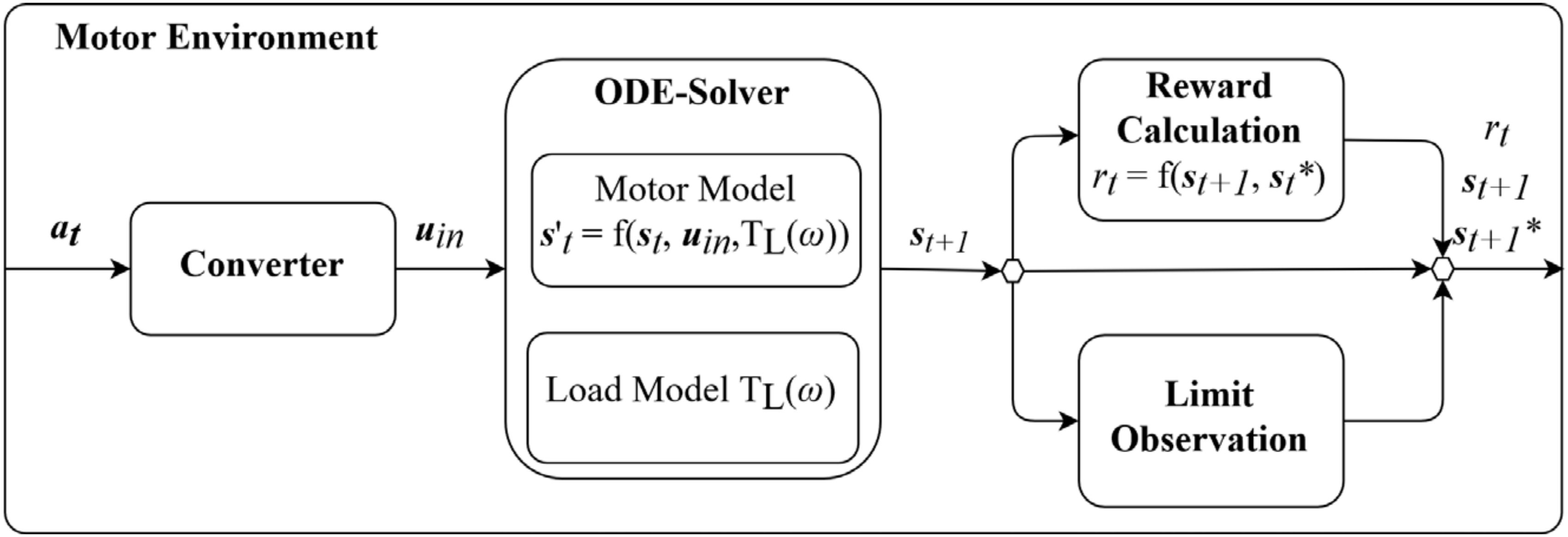
$$r_t = 1 - \sum_{k=0}^{N} w_{\{k\}} \left(s_{\{k\}t} - s^*_{\{k\}t}\right)^2$$

Limit Observation

Technical limits must be hold
Limit violation: additional reward (**punishment**)



Interaction of the Components



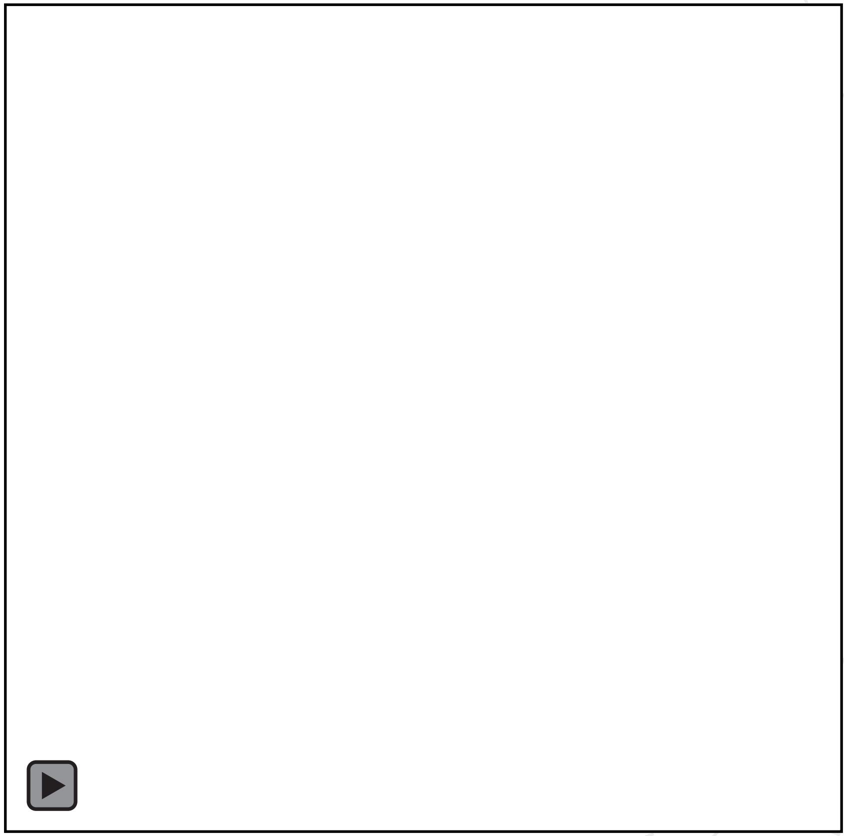
Content

Example

Series DC Motor
Speed control (reward weight for ω is 1)
1QC $(i \geq 0, u \geq 0)$

DDPG-Agent (continuous actions)
Actor and Critic network
$7\,500\,000$ training steps (12.5 min)
Shifted weighted sum of absolute error
Random white gaussian noise (decreasing power)

Cascaded PI-Controller
Outer speed control
Inner current control



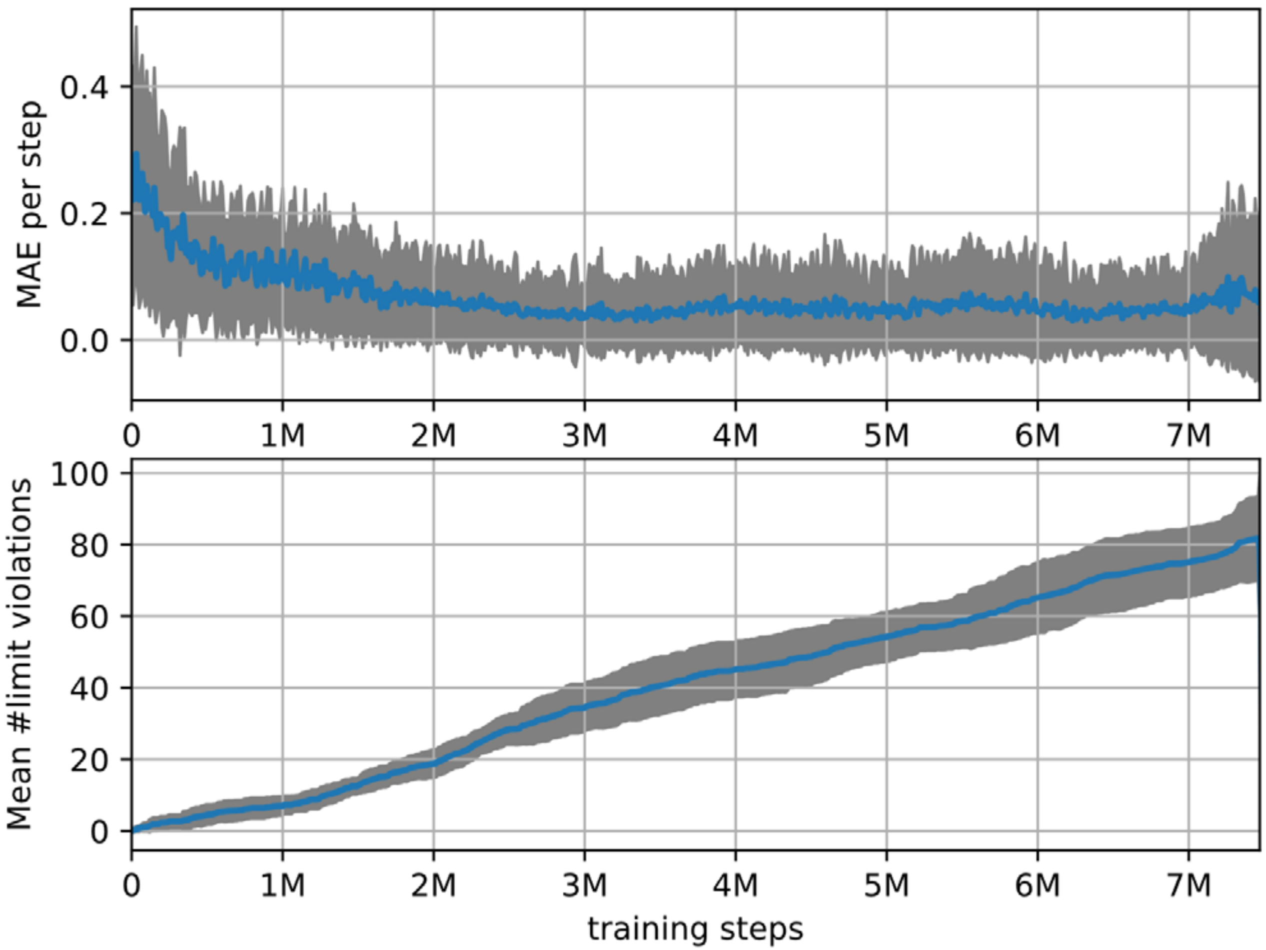
Learning Curve

Mean of 10 DDPG-agents with standard deviation

Mean absolute error per step decreases

Large standard deviation
- Reference types

Mean number of limit violations increases linear
- Gaussian noise
- Action repetition

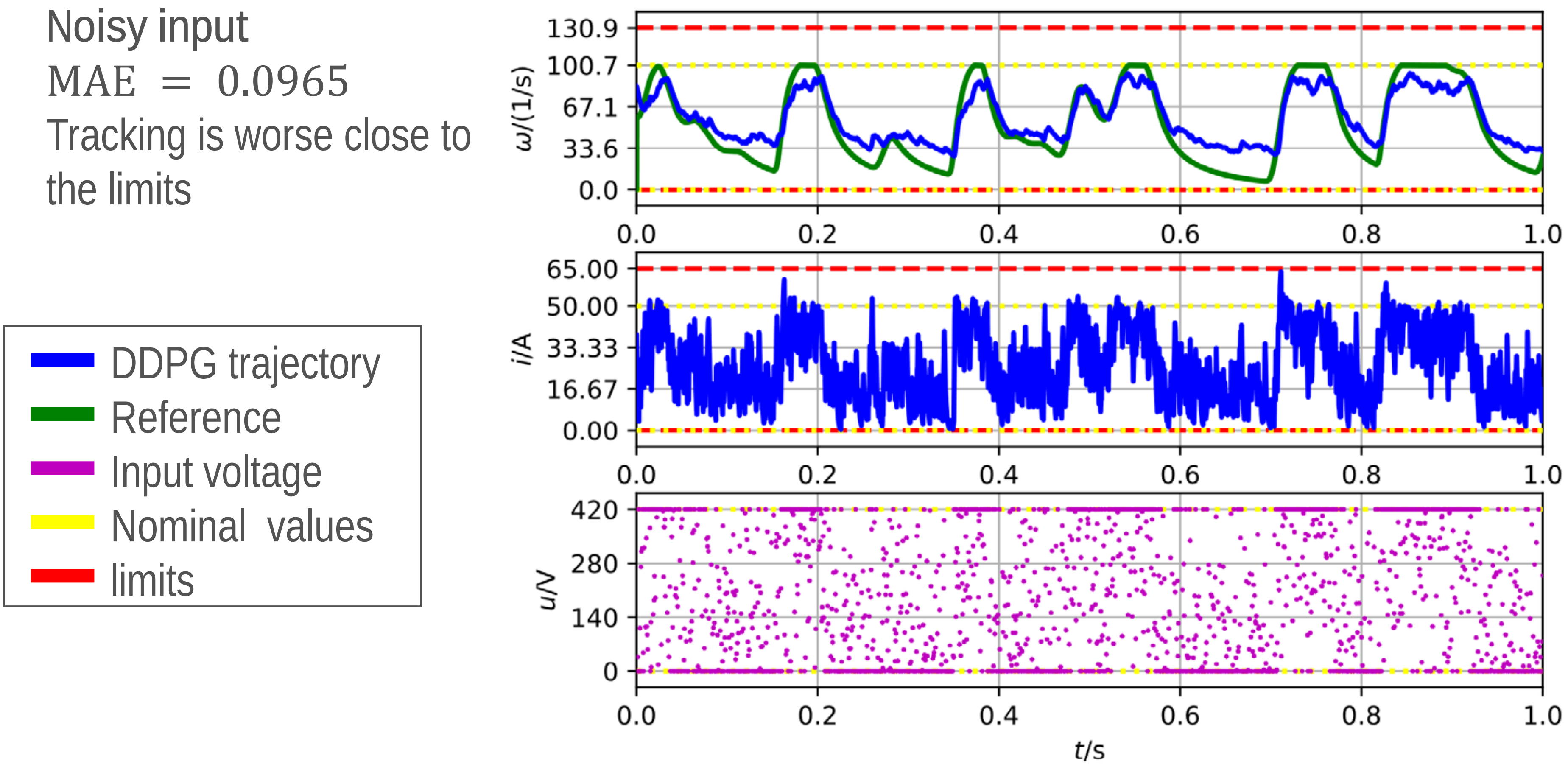


After about 1 000 000 training steps

Noisy input

$\mathrm{MAE} = 0.0965$

Tracking is worse close to the limits



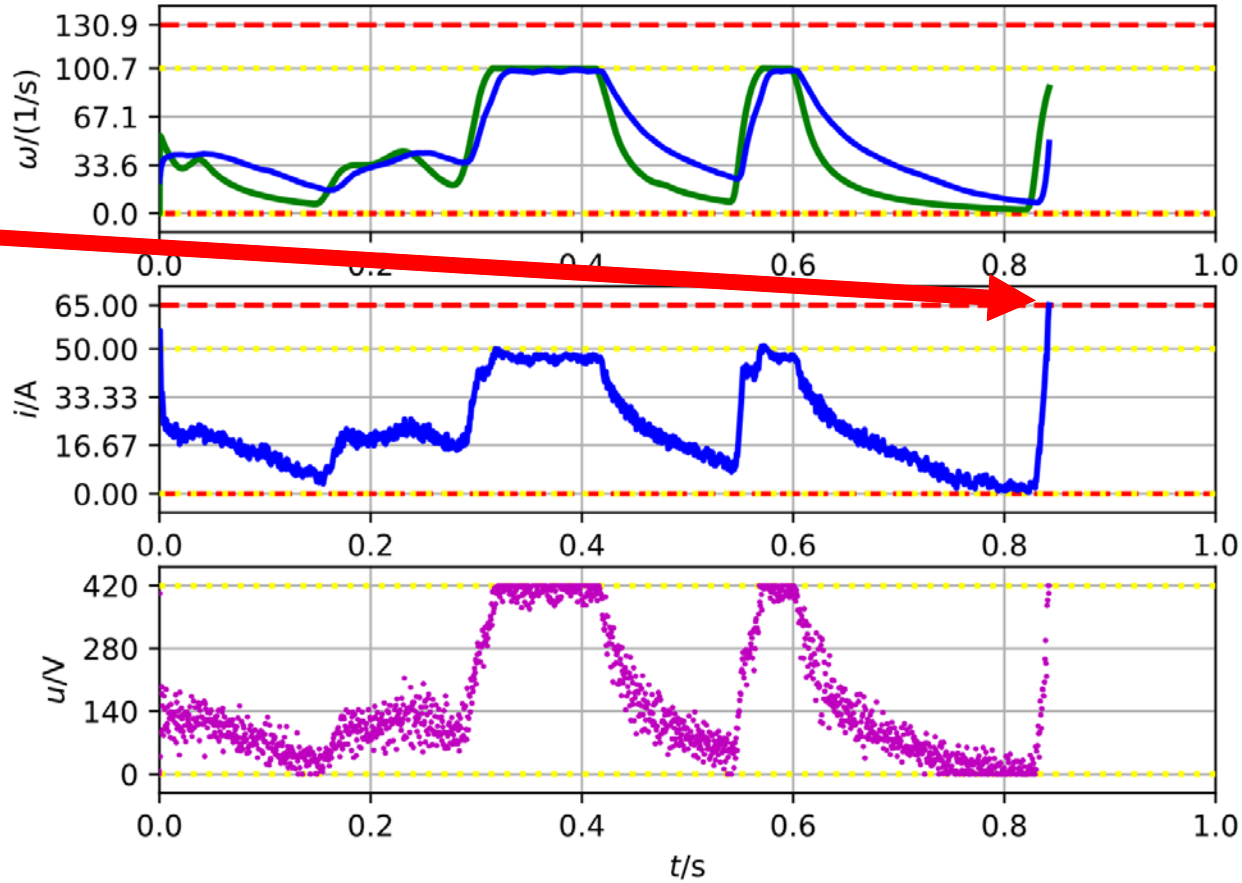
After 5 900 000 training steps

Less noisy input

Larger $\mathrm{MAE} = 0.1122$

Trajectory is delayed

Current limit violation



DDPG trajectory

Reference

Input voltage

Nominal values

limits

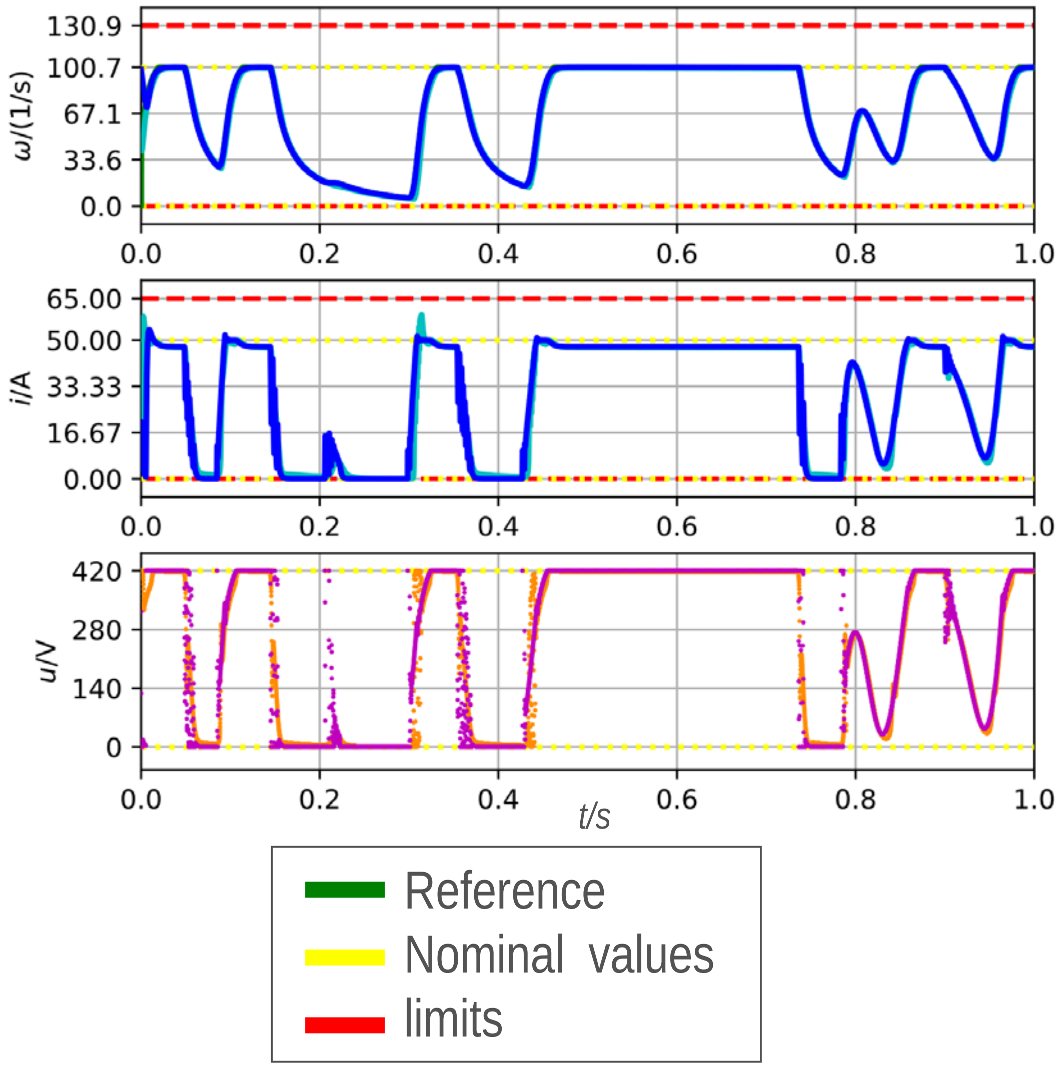
7.5M training steps

DDPG-agent (blue, pink)
Good tracking $\mathrm{MAE} = 0.0133$
no noise in the input

PI-Controller (cyan, orange)
$MAE = 0.0024$

Larger steady state error with DDPG

MAE of 100 traj.	DDPG	PI
Min	0.0009	0.001
Mean	0.0631	0.0323
Max	0.7037	0.6381



Content

Summary

Gym Electric Motor (GEM): An OpenAI Gym Environment for Electric Motors

reinforcement-learning · openai-gym-environments · machinelearning · openai-gym · benchmark · electrical-engineering · electric-drive

10 commits · 1 branch · 0 releases

Branch: master · New pull request

Wilhelm Kirchgaessner move banner file into docs/plots

dist	add distributions for pypi
docs	move banner file into docs/plots
examples	initial commit
gym_electric_motor	initial commit
.gitignore	add first sphinx html output
LICENSE	Initial commit
README.md	update readme, add project bann
requirements.txt	initial commit
setup.py	initial commit

Towards a Reinforcement Learning Environment Toolbox for Intelligent Electric Motor Control

Arne Traue, Gerrit Book, Wilhelm Kirchgässner, *Member, IEEE* and Oliver Wallscheid, *Member, IEEE*

***Abstract*—Electric motors are used in many applications and their efficiency is strongly dependent on their control. Among others, PI approaches or model predictive control methods are well-known in the scientific literature and industrial practice. A novel approach is to use reinforcement learning (RL) to have an agent learn electric drive control from scratch merely by interacting with a suitable control environment. RL achieved remarkable results with super-human performance in many games (e.g. Atari classics or Go) and also becomes more popular in control tasks like cartpole or swinging pendulum benchmarks. In this work, the open-source Python package gym-electric-motor (GEM) is developed for ease of training of RL-agents for electric motor control. Furthermore, this package can be used to compare the trained agents with other state-of-the-art control approaches. It is based on the OpenAI Gym framework that provides a widely used interface for the evaluation of RL-agents. The initial package version covers different DC motor variants and the prevalent permanent magnet synchronous motor as well as different power electronic converters and a mechanical load model. Due to the modular setup of the proposed toolbox, additional motor, load, and power electronic devices can be easily extended in the future. Furthermore, different secondary effects like controller interlocking time or noise are considered. An intelligent controller example based on the deep deterministic policy gradient algorithm which controls a series DC motor is presented and compared to a cascaded PI-controller as a baseline for future research. Fellow researchers are encouraged to use the framework in their RL investigations or to contribute to the functional scope (e.g. further motor types) of the package.**

***Index Terms*—electrical motors, power electronics, control, electric drive control, reinforcement learning, OpenAI Gym.**

I. Introduction

Electric motor control has been an important topic in research and industry for decades, and a lot of different strategies have been invented, e.g. PI-controller and model predictive control (MPC) [1]. The latter methods require an accurate model of the system. Based on this, the next control action is calculated through an online optimization over the next time steps [2]. Typical challenges when implementing MPC algorithms in drive systems are the computational burden due to the real-time optimization requirement and plant model deviations leading to inferior control performance during tran-

fication challenge in 2012, DNN have dominated research in many high level image processing tasks. Even reinforcement learning (RL) was influenced by DNN. New algorithms like deep-Q-learning (DQN) [4] and deep deterministic policy gradient (DDPG) [5] have been established. A famous example is the RL-agent AlphaGo [6] which has beaten the currently best human player in the game of Go recently, and sparked new interest in the field of self-learned decision-making. In the past years, RL has been applied to many control tasks like the inverse pendulum [7], the double pendulum [8] or the cartpole problem [9], and the application in electric power systems is also investigated [10].

Applying RL to electric motor control is an emerging approach [11]. In contrast to MPC, RL control methods do not need an online optimization in each step, which is often computational costly. Instead, RL-agents try to find an optimal control policy during an offline training phase before they are implemented in real-world application [2]. However, many modern RL algorithms are model-free and do not require model knowledge. Therefore, RL control methods can not only be trained in simulations but also in the field applications and optimize their control with respect to all the physical and parasitic effects as well as nonlinearities. Additionally, the same RL model architecture could be trained to control many different motors without expert's modification, similar to the RL-agent that learns to play different Atari games [12].

The authors' contribution to this research field is the development of a toolbox for training and validation of RL motor controllers called gym-electric-motor (GEM)[1]. It is based on OpenAI Gym environments [13]. Furthermore, different open-source RL toolboxes like Keras-rl [14], Tensorforce [15] or OpenAI Baselines [16] build upon the OpenAI Gym interface, which adds to its prevalence. For easy and fast development, RL-agents can be designed with those toolboxes and afterwards trained and tested with GEM before applying them to real-world motor control.

Currently, the GEM toolbox contains four different DC motors, namely the series motor, shunt motor, permanently excited motor and the externally excited motor as well as the

Outlook

Extend the toolbox
Induction Machine, Synchronous Reluctance Motor,..
Modularization (Exchangeable components), Unit tests

Optimize the RL-agents
Higher control performance and training speed

Test on real-world motors
Simulation trained agents performance on real motors
Train on real-world data
Combined online and offline training

Embedded online learning
Train the agents directly on real-world motors

Thanks for your Attention!

Reinforcement Learning
for Electric Motor Control
Gym-Electric-Motor Toolbox

Arne Traue and Gerrit Book

Questions??