PADERBORN UNIVERSITY

FACULTY OF
COMPUTER SCIENCE,
ELECTRICAL ENGINEERING
AND MATHEMATICS

Direct Torque Control of Permanent Magnet Synchronous Motor Drives via Safe Reinforcement Learning

From the Faculty of Electrical Engineering, Computer Science and Mathematics of Paderborn University

to obtain the academic degree
Doctor of Engineering (Dr.-Ing.)

approved dissertation
by

Maximilian Schenke

First Reviewer:	Prof. Dr.-Ing. Joachim Böcker
Second Reviewer:	Prof. Dr.-Ing. Oliver Wallscheid
Date of the dissertation defense:	18 May 2026

Paderborn 2026
Diss. EIM-E / 401

Acknowledgments

This work has been realized during my scientific employment at the Department of Power Electronics and Electrical Drives (LEA) at Paderborn University. Particularly the beginning of my academic service has been overshadowed by the COVID-19 pandemic, which rendered research and teaching quite cumbersome. First and foremost, I hence want to thank my former and current colleagues at LEA. Without you, these strange times working from home, and later in mostly empty university buildings, would have been so much harder to endure. Particularly, I would herein like to thank Anian Brosch, Sören Hanke, Marius Stender and Daniel Weber for their friendship and company beyond the office desk.

Further, I want to state particular thanks to each colleague who contributed to the making of this thesis, specifically to Barnabas Haucke-Korber, who was a critical supporter during setup of our test bench, and to my proofreaders Darius Jakobeit, Mario Peña, Dominik Schmies and (again) Daniel Weber. Altogether, we formed the subgroup that was trusted with the research project Learning Algorithms for the Automatization of Intelligent Power Electronic Systems (LEA-AI). It has been funded by the Federal Ministry of Research, Technology and Space (grant: 03VP11210), to whom I express my gratitude for the opportunities and innovations that it has allowed me. Moreover, my research has been funded via the project Very Efficient Deep Learning in the Internet of Things (VEDLIoT), that has been subject to the European Union's Horizon 2020 research and innovation program (grant: 957197), to which I extend my esteem.

My sincere appreciation goes to Prof. Dr.-Ing. Joachim Böcker, whom I thank for the academic freedom and the atmosphere of inherent scientific ambition that he established at LEA, which I always enjoyed and hope to carry forward. Special thanks go out to Prof. Dr.-Ing. Oliver Wallscheid, who mentored and guided my studies since I started my bachelor's thesis, and who has given me the opportunity to work on this unexplored and original topic. I would like to thank both of them for examining this dissertation. Further special thanks go out to Wilhelm Kirchgässner, whom I consider a key figure for establishing artificial intelligence in electric power systems, and to Elke Münsterteicher, whom I could always trust for her administrative support and organizational foresight.

Finally, I would like to thank my friends and my family for their constant encouragement and – not to forget – their often welcome distractions from my work. I apologize that I cannot include each of your names explicitly, but I want to be clear about the gratitude I have for your support, which I consider a perhaps indirect but surely irreplaceable contribution to the making of this thesis.

Abstract

The prevalent utilization of electric motors in automation and manufacturing, as well as the ongoing electrification of individual and public transport come with growing requirements for the control performance and efficiency of corresponding drives. While the operational quality benefits from the availability of powerful computational hardware, the complexity of corresponding algorithms increases the demand for highly qualified developers, development time and, in conclusion, implementation cost. Facing this evolution, the consideration of data-driven, highly-automated controller design procedures has potential to simplify the development process while conserving full performance potential.

In this work, the discipline of reinforcement learning, a subdomain of machine learning, is considered for approaching the task of data-driven optimal drive control. Herein, the focus is on a proof of concept for controlling the torque of a permanent magnet synchronous motor. Algorithms from the domain of reinforcement learning provide adaption of an approximate optimal controller in direct interaction with the plant system and without the necessity of mathematical models that describe the drive dynamics. The trial-and-error training phase incorporates the entire parasitic behavior into the optimization, but exposes the plant to damage if the controller operates without respecting the plant's limitations, which must be expected during the early stages of training.

Over the course of the training phase, the control performance is optimized on the basis of a reward function that is designed to assess torque tracking behavior and efficiency, and that is constructed in a model-free fashion. Herein, the targeted operational behavior of the drive is inspired by the well-established paradigms of the maximum-torque-per-current and maximum-torque-per-voltage characteristic. To address safety concerns, a safeguarding procedure is proposed that overwrites improper control actions in order to avoid hazardous system states. This algorithm is rolled out to the finite control set, (wherein the drive is controlled by means of distinct inverter switching states), and to the continuous control set (considering a modulator and averaged voltage references), and is subjected to the same condition of operability without available plant parameters.

Finally, an experimental validation is presented, wherein the safeguarding procedure, the training convergence and the torque tracking behavior is investigated. In this context, particular focus is on the real-time capable implementation of the control algorithm, allowing for a fully automated torque controller training process that completes within ten minutes and without human intervention. To the author's best knowledge, this is the worldwide first successful proof of concept for a torque controller synthesis based on reinforcement learning in the field of electric drives.

Zusammenfassung

Die Verbreitung von Elektromotoren in Automatisierung und Produktion, sowie die fortschreitende Elektrifizierung des öffentlichen und Individualverkehrs gehen mit wachsenden Anforderungen an die Regelperformanz und Effizienz entsprechender Antriebe einher. Während das Betriebsverhalten von der Verfügbarkeit leistungsfähiger Rechner profitiert, erhöht die Komplexität der entsprechenden Algorithmen den Bedarf an qualifizierten Entwicklern, die Entwicklungsdauer und letztlich die Implementierungskosten. Angesichts dieser Umstände bieten datengetriebene, automatisierte Reglerentwurfsverfahren eine Perspektive um den Entwicklungsprozess zu vereinfachen und weiterhin das volle Leistungspotential auszuschöpfen.

In dieser Arbeit wird Reinforcement Learning, ein Teilbereich des maschinellen Lernens, für die Aufgabe der datenbasierten optimalen Antriebsregelung betrachtet. Der Fokus liegt dabei auf einem Machbarkeitsnachweis bezüglich der Drehmomentregelung eines Permanentmagnet-Synchronmotors. Algorithmen des Reinforcement Learning ermöglichen die Adaption einer approximativ optimalen Regelung in direkter Interaktion mit der Regelstrecke und ohne die Notwendigkeit von mathematischen Modellen der Antriebsdynamik. Der Trainingsvorgang nach Versuch-und-Irrtum bezieht das gesamte parasitäre Verhalten in die Optimierung ein, riskiert aber die Schädigung der Anlage falls der untrainierte Regler die Einhaltung der Systemgrenzen nicht gewährleisten kann.

Im Verlauf der Trainingsphase wird die Regelungsperformanz auf Grundlage einer Reward-Funktion optimiert, welche Drehmomenttreue und Wirkungsgrad modellunabhängig bewertet. Das angestrebte Betriebsverhalten des Antriebs orientiert sich dabei an den Strategien der Maximum-Torque-per-Current- und Maximum-Torque-per-Voltage-Charakteristik. Um Sicherheitsbedenken zu begegnen wird ein Safeguarding-Algorithmus vorgestellt, der Steuerbefehle überschreibt um gefährliche Systemzustände zu vermeiden. Dieser Algorithmus wird auf den Betrieb mit diskretem (Schaltzustände des Umrichters) sowie kontinuierlichem Stellsignal (Spannung an den Motorklemmen) ausgerollt und muss gleichermaßen ohne Kenntnis der Systemparameter lauffähig sein.

In der abschließenden experimentellen Umsetzung wird der Safeguarding-Algorithmus, die Konvergenz des Trainings und die erreichte Drehmomenttreue betrachtet. Besonderes Augenmerk liegt dabei auf der echtzeitfähigen Implementierung, welche einen vollautomatische Trainingsablauf ohne menschliches Zutun innerhalb von zehn Minuten ermöglicht. Nach Kenntnis des Autors handelt es sich hierbei um den weltweit ersten erfolgreichen Machbarkeitsnachweis für eine auf Reinforcement Learning basierende Synthese einer Drehmomentregelung im Bereich der elektrischen Antriebe.

Own Publications

Articles with Direct Significance for this Thesis

The content of this work is strongly based upon the publications [A1, A2, A3], whose main innovations and contents have been developed, investigated and documented by the author of this thesis. Parts of the content of this thesis have been adopted from these publications in either modified or unmodified form, relating to text, depictions, data and mathematical formulary. Corresponding self-citations are not explicitly listed in this work for reasons of readability.

[A1] M. Schenke and O. Wallscheid. "A Deep Q-Learning Direct Torque Controller for Permanent Magnet Synchronous Motors". In: *IEEE Open Journal of the Industrial Electronics Society* 2 (2021), pp. 388–400. DOI: 10.1109/OJIES.2021.3075521.

[A2] M. Schenke, B. Haucke-Korber, and O. Wallscheid. "Finite-Set Direct Torque Control via Edge-Computing-Assisted Safe Reinforcement Learning for a Permanent-Magnet Synchronous Motor". In: *IEEE Transactions on Power Electronics* 38.11 (2023), pp. 13741–13756. DOI: 10.1109/TPEL.2023.3303651.

[A3] M. Schenke, B. Haucke-Korber, and O. Wallscheid. "Safe Reinforcement Learning Direct Torque Control for Continuous Control Set Permanent Magnet Synchronous Motor Drives". In: *IEEE Access* (2026). DOI: 10.1109/ACCESS.2026.3696042.

Further Articles in the Broader Context of this Thesis

As Main Author

[B4] M. Schenke, W. Kirchgässner, and O. Wallscheid. "Controller Design for Electrical Drives by Deep Reinforcement Learning: a Proof of Concept". In: *IEEE Transactions on Industrial Informatics* 16.7 (2020), pp. 4650–4658. DOI: 10.1109/TII.2019.2948387.

[B5] M. Schenke and O. Wallscheid. "Improved Exploring Starts by Kernel Density Estimation-based State-Space Coverage Acceleration in Reinforcement Learning". In: (2021). URL: https://arxiv.org/abs/2105.08990.

As Co-Author

[C6] O. Wallscheid, M. Schenke, and J. Böcker. "A Combined Approach to Identify Induction Machine Parameters and to Design an Extended Kalman Filter for Speed and Torque Estimation". In: *IEEE International Power Electronics and Motion Control Conference (PEMC)*. 2018, pp. 793–799. DOI: 10.1109/EPEPEMC.2018.8522008.

[C7] O. Wallscheid, M. Schenke, and J. Böcker. "Improving Torque and Speed Estimation Accuracy by Conjoint Parameter Identification and Unscented Kalman Filter Design for Induction Machines". In: *IEEE International Conference on Electrical Machines and Systems (ICEMS)*. 2018, pp. 1181–1186. DOI: 10.23919/ICEMS.2018.8549514.

[C8] G. Book, A. Traue, P. Balakrishna, A. Brosch, M. Schenke, S. Hanke, W. Kirchgässner, and O. Wallscheid. "Transferring Online Reinforcement Learning for Electric Motor Control from Simulation to Real-World Experiments". In: *IEEE Open Journal of Power Electronics* 2 (2021), pp. 187–201. DOI: 10.1109/OJPEL.2021.3065877.

[C9] P. Balakrishna, G. Book, W. Kirchgässner, M. Schenke, A. Traue, and O. Wallscheid. "gym-electric-motor (GEM): a Python Toolbox for the Simulation of Electric Drive Systems". In: *Journal of Open Source Software* 6.58 (2021), p. 2498. DOI: 10.21105/joss.02498.

[C10] B. Haucke-Korber, M. Schenke, and O. Wallscheid. "Reinforcement Learning-based Deep Q Direct Torque Control with Adaptable Switching Frequency Towards Six-Step Operation of Permanent Magnet Synchronous Motors". In: *VDE Innovative Kleinantriebs- und Kleinmotorentechnik (IKMT)*. 2022.

[C11] D. Weber, M. Schenke, and O. Wallscheid. "Safe Reinforcement Learning-based Control in Power Electronic Systems". In: *IEEE International Conference on Future Energy Solutions (FES)*. 2023. DOI: 10.1109/FES57669.2023.10182718.

[C12] D. Weber, M. Schenke, and O. Wallscheid. "Steady-State Error Compensation for Reinforcement Learning-based Control of Power Electronic Systems". In: *IEEE Access* 11 (2023), pp. 76524–76536. DOI: 10.1109/ACCESS.2023.3297274.

[C13] D. Jakobeit, M. Schenke, and O. Wallscheid. "Meta-Reinforcement-Learning-based Current Control of Permanent Magnet Synchronous Motor Drives for a Wide Range of Power Classes". In: *IEEE Transactions on Power Electronics* 38.7 (2023), pp. 8062–8074. DOI: 10.1109/TPEL.2023.3256424.

[C14] B. Haucke-Korber, M. Schenke, and O. Wallscheid. "Deep Q Direct Torque Control with a Reduced Control Set towards Six-Step Operation of Permanent Magnet Synchronous Motors". In: *IEEE International Electric Machines & Drives Conference (IEMDC)*. 2023. DOI: 10.1109/IEMDC55163.2023.10239018.

[C15] F. Book, A. Traue, M. Schenke, B. Haucke-Korber, and O. Wallscheid. "Gym-Electric-Motor (GEM) Control: an Automated Open-Source Controller Design Suite for Drives". In: *IEEE International Electric Machines & Drives Conference (IEMDC)*. 2023. DOI: 10.1109/IEMDC55163.2023.10239044.

[C16] B. Haucke-Korber, N. N. Aung, M. Schenke, M. Peña, D. Jakobeit, and O. Wallscheid. "Reinforcement Learning-based Direct Torque Control of Externally Excited Synchronous Motors: a Proof of Concept". In: *IEEE International Electric Machines & Drives Conference (IEMDC)*. 2025, pp. 916–921. DOI: 10.1109/IEMDC60492.2025.11061093.

[C17] M. Peña, M. Schenke, D. Jakobeit, B. Haucke-Korber, and O. Wallscheid. "Reinforcement Learning Control of Three-Level Converter Permanent Magnet Synchronous Machine Drives". In: *IEEE International Electric Machines & Drives Conference (IEMDC)*. 2025, pp. 57–64. DOI: 10.1109/IEMDC60492.2025.11061032.

[C18] D. Jakobeit, M. Peña, M. Schenke, B. Haucke-Korber, and O. Wallscheid. "Structural Optimization of Meta-Reinforcement Learning-based Finite-Control-Set Direct Torque Control of Permanent Magnet Synchronous Motors". In: *IEEE International Electric Machines & Drives Conference (IEMDC)*. 2025, pp. 673–678. DOI: 10.1109/IEMDC60492.2025.11061179.

[C19] D. Weber, D. Schmies, J. H. Lange, M. Schenke, and O. Wallscheid. "Optimal Control of Voltage-Forming Grid Inverters by Model Predictive Control and Reinforcement Learning". In: *IEEE Access* 14 (2026), pp. 38517–38535. DOI: 10.1109/ACCESS.2026.3670948.

[C20] D. Jakobeit, M. Schenke, and O. Wallscheid. "Universal Direct Torque Controller for Permanent Magnet Synchronous Motors via Meta-Reinforcement Learning". In: *IEEE Transactions on Power Electronics* 41.5 (2026), pp. 7394–7406. DOI: 10.1109/TPEL.2025.3635741.

[C21] N. Förster, D. Urbaneck, M. Schenke, A. Ebers, N. Schönlau, O. Wallscheid, and F. Schafmeister. "Improving the Usability of Calorimetric Measuring Chambers for Reliable Thermal Measurements". In: *International Exhibition and Conference for Power Electronics, Intelligent Motion, Renewable Energy and Energy Management (PCIM)*. 2025, pp. 2881–2890. DOI: 10.30420/566541386.

Nomenclature

Abbreviations

AC	Alternating current
ANN	Artificial neural network
CCS	Continuous control set
CPU	Central processing unit
DC	Direct current
DDPG	Deep deterministic policy gradient
DTC	Direct torque control
DQN	Deep q network
DUT	Device under test
FCS	Finite control set
FOC	Field-oriented control
FPGA	Field-programmable gate array
HPO	Hyperparameter optimization
MLP	Multilayer perceptron
MPC	Model predictive control
MTPC	Maximum torque per current
MTPV	Maximum torque per voltage
ODE	Ordinary differential equation
PMSM	Permanent magnet synchronous motor
PWM	Pulse-width modulation
RCPH	Rapid-control-prototyping hardware
RL	Reinforcement learning
RLS	Recursive least squares
SVM	Space vector modulation
SynRM	Synchronous reluctance motor
VSI	Voltage source inverter

Symbols

$\mathbf{0}$	Vector of zeros
$\mathbf{1}$	Vector of ones
β	Learning rate
γ	Discount factor
ϵ	Probability of a random action
$\varepsilon, \varepsilon_{\mathrm{el}}$	Angle / electric angle
$\boldsymbol{\zeta}$	Parameters of a policy function approximator
η	Update gain vector of the RLS
$\boldsymbol{\theta}$	Parameters of a state-action value function approximator
κ	Low-pass filter constant of the target network update
λ	Forgetting factor of the RLS
μ	Ensemble average
$\boldsymbol{\xi}$	Regressor vector of the RLS
$\boldsymbol{\pi}$	Policy function
ρ	Slack variable
σ	Ensemble standard deviation
τ	Termination flag
$\boldsymbol{\phi}$	Feature function
$\chi, \boldsymbol{\chi}$	Parameter / parameter vector of the RLS
$\psi, \boldsymbol{\psi}$	Magnetic flux linkage / magnetic flux linkage vector
ψ_{p}	Permanent magnet flux linkage
$\omega_{\mathrm{el}}, \omega_{\mathrm{me}}$	Electric / mechanic angular velocity
$\emptyset$	Empty set
$\mathbb{A}, \mathbb{B}, \mathbb{C}, \mathbb{D}, \mathbb{E}$	Subsets of the state space
$\mathbb{R}$	Set of real numbers
$\mathbb{N}$	Set of natural numbers
$\mathcal{A}$	Control set / action space (terms used synonymously)
$\mathcal{A}_{\mathrm{CCS}}$, $\mathcal{A}_{\mathrm{FCS}}$	Continuous control set / finite control set
$\mathcal{C}_i, \mathcal{C}_u$	Set of actions that fulfill the current / voltage feasibility condition
$\mathcal{E}$	Experience tuple

$\boldsymbol{A}$	Dynamic matrix of an affine linear system
a	Switching state index
$\boldsymbol{B}$	Input matrix of an affine linear system
$b, \boldsymbol{b}$	Bias / bias vector
c_ρ	Weighting factor of the slack variable
$d, \boldsymbol{d}$	Duty cycle / duty cycle vector
$\boldsymbol{e}$	Displacement vector of an affine linear system
$\boldsymbol{f}$, $\tilde{\boldsymbol{f}}$	Dynamic function
f_{act}	Activation function
$\boldsymbol{g}^\top, \boldsymbol{G}$	Linear parameter of a (vectorial) linear inequality
g	Return function
$h, \boldsymbol{h}$	Constant parameter of a (vectorial) linear inequality
$\boldsymbol{I}$	Identity matrix
$i, \boldsymbol{i}$	Current / current vector
$i_{\text{d+}}$	Upper threshold for the d-current
i_{lim}	Current limit
i_{n}	Nominal current
i_{s}	Stator current
$\boldsymbol{J}$	$\frac{\pi}{2}$ Rotation matrix
$\boldsymbol{k}^\top, \boldsymbol{K}$	Neuron weight vector, layer weight matrix
k	Discrete time index
$\boldsymbol{L}_{\text{dq}}$	Inductance matrix
$L_{\text{d}}, L_{\text{q}}$	Main inductance of the d-axis and q-axis
$L_{\text{dq}}, L_{\text{qd}}$	Cross inductance
m	Mean reversion rate of an Ornstein-Uhlenbeck process
$\boldsymbol{n}$	Noise vector
n, n_{me}	Speed / mechanic speed
$\boldsymbol{o}$	Observation vector
$\boldsymbol{P}$	Covariance matrix of the RLS estimator
P_{loss}	Dissipated power
p	Number of pole pairs
$\boldsymbol{Q}_{x,y}$	Transformation matrix from y to x with $\{x, y\} \in \{\text{abc}, \text{dq}, \alpha\beta\}$
q	State-action value function
R_{s}	Stator resistance

r	Reward function
$s, \boldsymbol{s}$	Switching state / switching state vector
T	Measured torque
T^*	Reference torque
$\hat{T}_{\mathrm{EM}}$	Electromagnetic torque
T_{TA}	Turnaround time
T_{s}	Sampling time
t	Continuous time
u_{DC}	DC-link voltage
$u, \boldsymbol{u}$	Voltage / voltage vector / general control input
$\boldsymbol{u}_{\mathrm{dq,e}}$	Equilibrium voltage
$\boldsymbol{u}_{\mathrm{dq,f}}$	Fundamental voltage
$\boldsymbol{W}$	Linear parameter of an affine linear relation
$\boldsymbol{w}$	Constant parameter of an affine linear relation
$\boldsymbol{x}$	System state
$\boldsymbol{y}$	Layer / network output

Notation

$x(t)$	Continuous-time signal
$x[k]$	Discrete-time signal
$\boldsymbol{x}$	Column vector
$\boldsymbol{x}^\top$	Row vector
$\boldsymbol{X}$	Matrix
$\boldsymbol{X}^\top$	Transpose of $\boldsymbol{X}$
$\boldsymbol{X}^{-1}$	Inverse of $\boldsymbol{X}$
$\|\boldsymbol{x}\|_2$	Euclidean norm of vector $\boldsymbol{x}$
$\|\boldsymbol{x}\|_\infty$	Maximum norm of vector $\boldsymbol{x}$
$\hat{x}$	Estimation / prediction of x
x^*	Reference / optimal value for x
$\overline{x}$	Dynamic average of x
$f(\boldsymbol{x})$	Element-wise application of the scalar function f
$[a, b]$	Closed interval, i.e., $[a, b] = \{x \in \mathbb{R} \mid a \leq x \leq b\}$
$]a, b]$	Left-open interval, i.e., $]a, b] = \{x \in \mathbb{R} \mid a < x \leq b\}$
$[a, b[$	Right-open interval, i.e., $[a, b[= \{x \in \mathbb{R} \mid a \leq x < b\}$
$]a, b[$	Open interval, i.e., $]a, b[= \{x \in \mathbb{R} \mid a < x < b\}$
$\nabla_{\boldsymbol{w}} x_{\boldsymbol{w}}$	Gradient of $x_{\boldsymbol{w}}$ with respect to $\boldsymbol{w}$
$\boldsymbol{x}_{\mathrm{dq}}$	Representation of the quantity $\boldsymbol{x}$ in rotor-fixed dq-coordinates
$\boldsymbol{x}_{\alpha\beta}$	Representation of the quantity $\boldsymbol{x}$ in stator-fixed $\alpha\beta$-coordinates
$\boldsymbol{x}_{\mathrm{abc}}$	Representation of the quantity $\boldsymbol{x}$ in stator-fixed abc-coordinates
$\boldsymbol{x} \in \mathbb{X}$	$\boldsymbol{x}$ is an element of the set $\mathbb{X}$
$\boldsymbol{x} \notin \mathbb{X}$	$\boldsymbol{x}$ is not an element of the set $\mathbb{X}$
$\in_{\mathrm{R}} \mathbb{X}$	Arbitrarily distributed random sample from the set $\mathbb{X}$
$\mathcal{N}_{(\boldsymbol{\mu}, \boldsymbol{\Sigma})}$	Random sample from a multivariate normal distribution with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$
$\mathcal{U}_{\mathbb{X}}$	Random sample from the uniform distribution on the set $\mathbb{X}$
$\partial\mathbb{X}$	Edge of the set $\mathbb{X}$

Contents

1 Introduction

1.1 Motivation

Control of three-phase AC drives is a well-studied but comprehensive discipline with over sixty years of history. Established methods such as proportional-integral control on field-oriented coordinates (field-oriented control[1], FOC) [1] or model predictive control (MPC) [2] rely on accurate modeling of the motor and the supplying inverter. While corresponding drive models do not strongly differ on a conceptual level from one application to another, they may convey structural assumptions or omissions with concern to occurring systematic and parasitic behavior. Hence, usual modeling approaches are often targeted at simplifying controller design, which comes at the potential cost of performance and efficiency.

Moreover, utilized models would ideally need to be parameterized for each individual drive setup. Herein, effort is often spared through manual controller adaption at run-time (tuning), reliance on nameplate and datasheet specifications, or by assuming identical parameters for components from the same manufacturing series. If such heuristics are not permissible in the face of performance requirements, the parameterization needs to be identified, e.g., in the form of flux linkage and saturation maps [3]. The selection, construction and implementation of suitable control algorithms and identification experiments comes with skill requirements and manual effort and, consequently, resulting performance may vary depending on the control designer's experience level. Unfortunately, the increasing demand for engineering competence stands in opposition to a declining number of university entrants in corresponding fields [4].

Especially the performance potential of MPC is tightly coupled to model quality. Herein, the actuator signals are determined by optimizing a performance metric (e.g., minimize the tracking error) in consideration of the predicted future behavior of the drive, rendering it an optimal control approach. Naturally, theoretical optimality does hardly transfer to reality in full extent, as the necessary prediction accuracy is not easily achieved in the light of the previously discussed modeling challenges. Further, the considered prediction horizon is limited by the available computing capacity, i.e., potent MPC necessitates powerful

[1] Although many drive control schemes use the concept of field orientation, only applications with proportional-integral regulators are generally referred to as 'field-oriented controllers'.

computational hardware. Due to these economic counterarguments, utilization of MPC in electric drive applications is rather untypical in industrial contexts, and corresponding products (e.g., [5]) are targeted at highly specialized tasks only.

These considerations motivate the investigation of reinforcement learning (RL) algorithms as data-driven approaches for electric drive control, which could benefit the overall system performance operating largely independent of idealizing assumptions by building their control policy partly or merely on factual cause-and-effect relations. RL is a machine-learning discipline that strongly corresponds to optimal control. Related algorithms allow controller design on the basis of real-world measurement data and with reduced human effort, but have been a rather new candidate for power system control at the time of the making of this work.

Hence, this work is to be understood as a proof of concept for the application of RL-based drive control for the specific task of torque regulation. It is dedicated to determining potentials, complications and shortcomings of the data-driven design approach in general, and experimental RL in particular, with respect to the requirements of power electronic systems.

1.2 Related Work

Before the role of RL in electric power systems is discussed, some light is to be shed on RL in general.

1.2.1 Reinforcement Learning on the Rise

Contemporary deep RL for sequential problem solving had its first significant success and growth in visibility when Google DeepMind presented their AlphaGo program to beat world-leading professionals in the board game of Go in 2016 [6]. Prior to this event, Go had been considered too complex to be reliably winnable by means of computational methods, and its professional base was certain not to be outperformed by corresponding algorithms. Later on, in 2017, the successor program named AlphaZero was critically acclaimed as the best-available board game engine [7]. Herein, it was also capable of outperforming established chess-playing algorithms, which came as a surprise because chess was considered a mostly computationally resolved game since the early 1990s, and the possible existence of even better programs was largely doubted.

Despite being toy examples with no direct societal impact, the domination of RL in board games has lead to a vast increase of applications that rely on RL for decision making. The underlying problems usually have in common that they are analytically comprehensive to model, and that numerical optimization for the optimal decision comes with infeasible requirements in terms of calculation effort and memory space. This category also includes the successful attempt on positional stabilization of the plasma inside a nuclear fusion reactor, which was again driven by Google DeepMind in 2022 [8]. Although not being

the concluding building block for enabling fusion power yet, this application has been regarded a possible outlook of RL affecting society.

Lastly, the expected disruption of everyday life was caused by the immense capacity growth in large language models such as OpenAI's ChatGPT [9] or DeepSeek-AI's DeepSeek [10]. From 2023 onward, corresponding tools have changed many processes that deal with synthesis and revision of text-based content, e.g., journalism [11] or education [12], and come with the potential of replacing classical web-search engines [13]. Despite not being build upon RL exclusively, it is an essential component of the training procedure for large language models.

While each of the named applications in RL have contributed to its popularity in the academic sector and – specifically in the case of large language models – to the acceptance of machine-learning tools in society in general, their dissemination and establishment in power system control is still in progress. Many corresponding developments have taken place simultaneously to the making of this thesis, or stand in direct relation to its (partial) results. Critical features and milestones of RL in power system control are to be discussed in the following.

1.2.2 Reinforcement Learning Control in Electric Power Systems

1.2.2.1 Prerequisites

The application of RL in the context of control systems comes with a set of requirements that extend beyond the previously-discussed decision making tasks in board games or language models. Herein, the two most fundamental differences consist of

1) the requirement of real-time capability (with particularly fast dynamcis in electric power systems),

   and

2) the direct safety threat that originates from inappropriate or insufficient controller behavior, which must be expected particularly during early training.

Both of these circumstances complicate the experimental validation of RL controllers in power systems and are, hence, of major interest when targeting their investigation. The real-time condition 1) is mostly a problem of affordable hardware resources and implementation effort, i.e, corresponding tasks have initially been restricted to larger institutions[2]. Although increasing the entry hurdle to RL control applications for the time being, this issue is not fundamental and may prospectively vanish completely with ongoing technical progress in chip manufacturing [14].

[2]E.g., the earlier-mentioned plasma stabilization problem from [8], despite not being a power electronic system.

In contrast, the safety condition 2) is a task specific issue that must be addressed when practical investigations are targeted. In order to avoid risk to personnel, facilities and equipment, many proposals of RL control in power systems have yet been tested only in simulation, or after comprehensive simulative validation. Meanwhile, learning the control policy in direct interaction with the physical plant system has yet been largely unaddressed.

Corresponding conclusions are supported by the literature overview that is presented in Tab. 1.1. Herein, it can be seen that available publications focus simulations with considerable majority. In this context, the designation of RL as a data-driven approach to optimal control must often be doubted when pure simulation or transfer from simulation to reality is investigated, because the simulative training phase is commonly enabled by system models that necessarily employ distorting simplifications and assumptions to synthesize the training data.

On the other hand, the possibility of training an RL controller through interaction with a real-world plant is notably underrepresented. Although such a scenario poses a higher safety risk for implementation, it is evident that (artificial) simulation data cannot describe the behavior of the plant system with concluding accuracy. Consequently, a performance increase from data-driven control methods can only be expected when said data is acquired from interaction with the physical plant, which may only be enabled when addressing the associated safety concerns.

1.2.2.2 Topics

Considering the topics that have yet been investigated in terms of RL-based drive control, several research gaps can be identified. With respect to Tab. 1.1, it can be seen that outside of Paderborn University, current and speed control drew most attention while torque control was mostly ignored. A possible explanation for this could be the structure of corresponding control loops: speed and current are available for measurement in commercial drive applications and, hence, closed-loop control is the obvious approach[3]. Contrary, torque control applications must be designed with the knowledge that commercial implementations do not incorporate a torque sensor and, hence, end-to-end closed-loop control structures are disqualified for final deployment. Furthermore, measuring torque comes with significant parasitic effects such as measurement delay and drive train oscillation, which need special attention when real-world training is targeted, but which are oftentimes overlooked in simulation.

For grid-forming inverters, the restrictions of mechanical subsystems do not apply and, hence, corresponding applications can be setup with less effort and may be crafted in a closed-loop fashion more easily[4]. Furthermore, available publications indicate that perfor-

[3]Specialized applications may spare the speed or position measurement. These would usually incorporate an observer-like structure for a corresponding regulation task and have not yet been discussed in RL-related literature.

[4]Also in the domain of electric grids, specific applications may prefer the omission of sensory equipment, increasing the complexity of the control task.

Tab. 1.1: Overview of the coverage of RL control tasks in electric power systems; highlighted entries correspond to the content of this work

Application	Control Variable	Set	Training: Validation:	Simulation Simulation	Simulation Experiment	Experiment Experiment
Electric drives	Current	FCS		[15, 16]	[17]	
		CCS		[B4, B5, C12, C13], [16, 21–24]	[18–20]	[C8]
	Torque	FCS		[A1], [C10, C14, C17]	[C20]	[A2]
		CCS		[C16] [25, 26]		[A3]
	Speed	FCS		[27, 28]		
		CCS		[29–37]	[38, 39]	
	Position	CCS		[40]		
Electric grids	Voltage / Current (Component level)	FCS		[41, 42]	[43, 44]	
		CCS		[C11, C12], [48, 49]	[44–47]	[C19]
	Power (Distribution level)	FCS		[50–53]		
		CCS		[53–62]		
	Operating cost / Power loss (Economic level)	FCS		[63–66]		
		CCS		[67, 68]		

mance levels that can compete with established methods from optimal control are already in reach for RL-driven control algorithms (cf. [C19]). In terms of control tasks, Tab. 1.1 suggests that investigations that focus power distribution and efficient grid operation are quite popular, but are yet to be applied in real-world experiments. It can be assumed that this is at least partly based on the fact that corresponding experimental rigs can be highly expensive [69]. Further, the most complicated dynamics and safety risks are encountered and handled on the layer of electric components and, therefore, the superimposed control problems of power distribution and efficiency maximization might not benefit significantly from preferring a real-world investigation over an idealized simulation environment.

Despite the fact that real-world training would be generally preferable for crafting RL controllers that can handle the plant's behavior under consideration of all parasitic effects, it must be noted that Tab. 1.1 also incorporates some publications that propose meta-RL approaches. In meta-RL, the difference in between plant systems are to be addressed within the RL controller, i.e, corresponding controllers can be adaptively applied to systems they have not been trained on, which also allows for a smoother transfer from

simulation to reality. Therefore, corresponding publications do commonly feature purely simulative training without loss of generality (cf. [C20, 17]).

Another classification of publications can be made on the basis of the operated action space. Herein, finite-control-set (FCS) implementations can be distinguished from continuous-control-set (CCS) implementations. While an FCS interface is characterized by a finite number of distinguishable control actions (e.g., different inverter switching states), a CCS interface refers to setups that are controlled via real-valued quantities (e.g., voltages that are applied to a device). While both of these control sets can also be found in conventional control solutions, CCS interfacing is much more common in industrial drive applications, with FOC being its most prominent representative. In terms of RL, FCS algorithms usually come with less complexity and computational requirement, but CCS-RL seems yet to be the more popular research subject in drive control as suggested by Tab. 1.1. Beyond the distinction into FCS and CCS lies the possibility to merge both options into a hybrid control set, which is then characterized by a combined consideration of discrete and continuous control variables (e.g., the finite number of inverter switching states and the real-valued duration for which they apply). Although corresponding implementations can be found in the domain of optimal drive control [70] and fitting RL algorithms are available [71], RL-based applications that interface a power systems via a hybrid control set are yet to be presented.

Finally, it must be questioned whether a hypothetic future utilization of RL in drive control applications is beneficial when utilized on a purely electrical level. Since electric drives are electro-mechanic systems, it is rarely sufficient to implement a current controller without superimposed structures that are concerned with regulating the mechanical quantities. Despite being more prominently covered in literature, corresponding investigations on RL-based current control should, hence, be understood as proofs of concept or intermediate results without a definitive target application. These considerations motivate an increased engagement with the torque tracking task by means of RL-based drive control, which is the focus of this work.

1.3 Objectives and Content Structure

This work targets the proof of concept of an RL-based torque controller setup for permanent magnet synchronous motors (PMSMs). Herein, the main challenge lies within the data-driven nature of the considered control structure, which requires handling of the following aspects:

- The quality of torque control performance must be evaluated without the availability of a drive model and / or drive parameters to render the training of the RL-based controller structure truly data-driven.
- The resulting controller must be functional without a torque sensor, which is almost always spared for economic reasons. Only the training phase is herein permitted to be equipped with a corresponding measurement device.

- The training phase is to be conducted in an online fashion, without human intervention and in direct interaction with the plant drive, which requires a real-time capable inference of the RL controller and a distributed computing tool chain that enables completion of the training in a timely manner.
- Violations of safety limitations are to be avoided via a safeguarding routine that disallows application of harmful voltage (i.e., the RL controller is hindered from steering the plant motor into hazardous states), which would be a significant risk during the training procedure if left unchecked.
- The RL-based torque controller is to be validated for both, FCS and CCS interfacing of the supplying inverter.

Due to the early stage of development that RL-based control still is in, and because of the unusual presumption of model unavailability the controller design process is subjected to, this investigation does not target competitive performance to established control methods such as FOC or MPC. Instead, the proof of feasibility in consideration of the named challenges is the main contribution of this work. Herein, the autonomous nature of the controller training process and the independence of model knowledge provide advantages that are outside the quantification of torque tracking performance and efficiency.

This work is structured into four parts. After a review of the theoretical fundamentals in the first part, the second part focuses the formulation of the torque control task as an optimization problem in terms of an RL setting. Herein, also the software and hardware setup for the practical application are outlined. Thereafter, the third and fourth part of this work revolve around implementation of the RL-based torque controller in the FCS and the CCS scenario, respectively. For each, a safeguarding procedure is derived on the basis of system-theoretical considerations before experimental results are presented and discussed. The last chapter concludes on the yielded insights and looks out to promising research questions for future investigations. The appendix supplies background information to round off this work. It mainly covers the RL algorithms that have been utilized, whereas their specific selection is to be understood as a degree of freedom within the proposed torque control scheme.

2 Fundamentals

2.1 Permanent Magnet Synchronous Motor

Electric three-phase drives can be efficiently described and modeled using the field-oriented dq-coordinate system (denoting the direct and the quadrature axis), wherein the d axis has the same orientation as the rotor's magnetic field, resulting in a rotating reference frame. The orientation of the rotor field is equivalent to the rotor's electric angular position $\varepsilon_{\mathrm{el}}$ within synchronous motor applications. As sensorless control is not within the scope of this work, it is assumed that a measurement of $\varepsilon_{\mathrm{el}}$ is available at all times. For an arbitrary vector quantity $\boldsymbol{x}$, which could be the voltages $\boldsymbol{u}$, the currents $\boldsymbol{i}$ or the magnetic flux linkages $\boldsymbol{\psi}$, the corresponding coordinate transformation is then defined by

$$\underbrace{\begin{bmatrix} x_{\mathrm{d}}(t) \\ x_{\mathrm{q}}(t) \end{bmatrix}}_{\boldsymbol{x}_{\mathrm{dq}}(t)} = \overbrace{\underbrace{\begin{bmatrix} \cos(\varepsilon_{\mathrm{el}}(t)) & \sin(\varepsilon_{\mathrm{el}}(t)) \\ -\sin(\varepsilon_{\mathrm{el}}(t)) & \cos(\varepsilon_{\mathrm{el}}(t)) \end{bmatrix}}_{\boldsymbol{Q}_{\mathrm{dq},\alpha\beta}(\varepsilon_{\mathrm{el}}(t))} \underbrace{\begin{bmatrix} \frac{2}{3} & -\frac{1}{3} & -\frac{1}{3} \\ 0 & \frac{1}{\sqrt{3}} & -\frac{1}{\sqrt{3}} \end{bmatrix}}_{\boldsymbol{x}_{\alpha\beta}(t)} \begin{bmatrix} x_{\mathrm{a}}(t) \\ x_{\mathrm{b}}(t) \\ x_{\mathrm{c}}(t) \end{bmatrix}}^{\boldsymbol{Q}_{\mathrm{dq,abc}}(\varepsilon_{\mathrm{el}}(t))} \overbrace{\phantom{\begin{bmatrix} x_{\mathrm{a}}(t) \\ x_{\mathrm{b}}(t) \\ x_{\mathrm{c}}(t) \end{bmatrix}}}^{\boldsymbol{x}_{\mathrm{abc}}(t)}. \tag{2.1}$$

Herein, the stator-fixed, two-phase αβ-reference frame appears as an intermediate result, and the stator-fixed, three-phase abc-reference frame corresponds to the physical ports of the drive system. The relation between abc-, αβ-, and dq-reference frame is illustrated in Fig. 2.1.

The electric rotor angle $\varepsilon_{\mathrm{el}}$ corresponds to the mechanic angular velocity ω_{me} via

$$\frac{\mathrm{d}}{\mathrm{d}t}\varepsilon_{\mathrm{el}}(t) = \omega_{\mathrm{el}}(t) = p\omega_{\mathrm{me}}(t), \tag{2.2}$$

with p denoting the number of pole pairs and ω_{el} denoting the electric angular velocity. Motor speed n_{me} and mechanical angular velocity ω_{me} correspond via

$$\omega_{\mathrm{me}} = 2\pi n_{\mathrm{me}}. \tag{2.3}$$

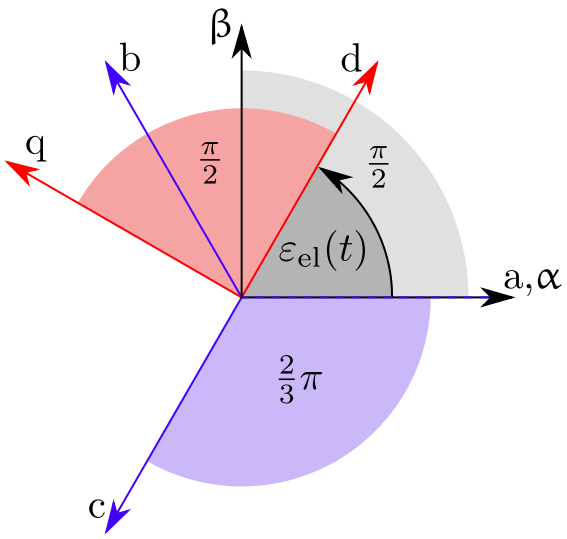


Fig. 2.1: Graphical depiction of the different coordinate systems; blue: stator-fixed, three-phase abc-reference frame, black: stator-fixed, two-phase αβ-reference frame, red: rotor-fixed, two-phase dq-reference frame

Within the dq frame, the electric behavior of a PMSM can be efficiently described by a system of ordinary differential equations (ODEs) [72]:

$$\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{\psi}_{\mathrm{dq}}(t) = \boldsymbol{u}_{\mathrm{dq}}(t) - R_{\mathrm{s}}\boldsymbol{i}_{\mathrm{dq}}(t) - p\omega_{\mathrm{me}}(t)\boldsymbol{J}\boldsymbol{\psi}_{\mathrm{dq}}(t), \tag{2.4}$$

with

$$\boldsymbol{\psi}_{\mathrm{dq}}(t) = \begin{bmatrix}\psi_{\mathrm{d}}(t)\\ \psi_{\mathrm{q}}(t)\end{bmatrix}, \ \boldsymbol{i}_{\mathrm{dq}}(t) = \begin{bmatrix}i_{\mathrm{d}}(t)\\ i_{\mathrm{q}}(t)\end{bmatrix}, \ \boldsymbol{u}_{\mathrm{dq}}(t) = \begin{bmatrix}u_{\mathrm{d}}(t)\\ u_{\mathrm{q}}(t)\end{bmatrix}, \text{ and } \boldsymbol{J} = \begin{bmatrix}0 & -1\\ 1 & 0\end{bmatrix}.$$

Herein, $\boldsymbol{u}_{\mathrm{dq}}$ describes the voltage vector, $\boldsymbol{i}_{\mathrm{dq}}$ the current vector, $\boldsymbol{\psi}_{\mathrm{dq}}$ denotes the magnetic flux linkage vector and R_{s} the electric stator resistance. By first principle, the relation between flux linkage and current is algebraic:

$$\boldsymbol{\psi}_{\mathrm{dq}}(t) = \boldsymbol{\psi}_{\mathrm{dq}}(\boldsymbol{i}_{\mathrm{dq}}(t), \psi_{\mathrm{p}}), \tag{2.5}$$

with ψ_{p} being the permanent magnet flux linkage. The differential relation between $\boldsymbol{\psi}_{\mathrm{dq}}$ and $\boldsymbol{i}_{\mathrm{dq}}$ can further be interpreted via a differential inductance matrix $\boldsymbol{L}_{\mathrm{dq}}$:

$$\begin{aligned}\boldsymbol{L}_{\mathrm{dq}}(\boldsymbol{i}_{\mathrm{dq}}) &= \begin{bmatrix}L_{\mathrm{d}}(\boldsymbol{i}_{\mathrm{dq}}) & L_{\mathrm{dq}}(\boldsymbol{i}_{\mathrm{dq}})\\ L_{\mathrm{qd}}(\boldsymbol{i}_{\mathrm{dq}}) & L_{\mathrm{q}}(\boldsymbol{i}_{\mathrm{dq}})\end{bmatrix} = \frac{\partial\boldsymbol{\psi}_{\mathrm{dq}}}{\partial\boldsymbol{i}_{\mathrm{dq}}} = \begin{bmatrix}\frac{\partial\psi_{\mathrm{d}}}{\partial i_{\mathrm{d}}} & \frac{\partial\psi_{\mathrm{d}}}{\partial i_{\mathrm{q}}}\\ \frac{\partial\psi_{\mathrm{q}}}{\partial i_{\mathrm{d}}} & \frac{\partial\psi_{\mathrm{q}}}{\partial i_{\mathrm{q}}}\end{bmatrix}\\ \Rightarrow \quad \frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{\psi}_{\mathrm{dq}}(t) &= \boldsymbol{L}_{\mathrm{dq}}(\boldsymbol{i}_{\mathrm{dq}}(t))\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{i}_{\mathrm{dq}}(t).\end{aligned} \tag{2.6}$$

Inserting (2.6) into (2.4) yields the ODE system

$$\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{i}_{\mathrm{dq}}(t) = \underbrace{-\boldsymbol{L}_{\mathrm{dq}}^{-1}(\boldsymbol{i}_{\mathrm{dq}}(t))R_{\mathrm{s}}}_{\boldsymbol{A}(t)}\boldsymbol{i}_{\mathrm{dq}}(t) + \underbrace{\boldsymbol{L}_{\mathrm{dq}}^{-1}(\boldsymbol{i}_{\mathrm{dq}}(t))}_{\boldsymbol{B}(t)}\boldsymbol{u}_{\mathrm{dq}}(t) \underbrace{-p\omega_{\mathrm{me}}(t)\boldsymbol{L}_{\mathrm{dq}}^{-1}(\boldsymbol{i}_{\mathrm{dq}}(t))\boldsymbol{J}\boldsymbol{\psi}_{\mathrm{dq}}(\boldsymbol{i}_{\mathrm{dq}}(t),\psi_{\mathrm{p}})}_{\boldsymbol{e}(t)}, \tag{2.7}$$

which is a parameter-variant, affine-linear state-space model that can be abbreviated by

$$\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{i}_{\mathrm{dq}}(t) = \boldsymbol{A}(t)\boldsymbol{i}_{\mathrm{dq}}(t) + \boldsymbol{B}(t)\boldsymbol{u}_{\mathrm{dq}}(t) + \boldsymbol{e}(t). \tag{2.8}$$

In consideration of the usually high sampling frequencies in electric power systems [73, 74], ω_{me} and $\boldsymbol{L}_{\mathrm{dq}}$ are assumed constant during one sampling period T_{s}, leading to the discrete-time system representation

$$\boldsymbol{i}_{\mathrm{dq}}[k+1] = \boldsymbol{A}[k]\boldsymbol{i}_{\mathrm{dq}}[k] + \boldsymbol{B}[k]\boldsymbol{u}_{\mathrm{dq}}[k] + \boldsymbol{e}[k], \tag{2.9}$$

wherein k is the discrete time index[1]. In the following, the discrete-time representation of signals and systems (denoted by square brackets $x[k]$) is preferred over the continuous-time representation (denoted by round brackets $x(t)$), which is pragmatic for the targeted digital implementation of corresponding algorithms.

In the context of safe drive utilization, it is sensible to define the stator current i_{s} and the equilibrium voltage $\boldsymbol{u}_{\mathrm{dq,e}}$

$$i_{\mathrm{s}}[k] = ||\boldsymbol{i}_{\mathrm{dq}}[k]||_2 = \sqrt{i_{\mathrm{d}}^2[k] + i_{\mathrm{q}}^2[k]}, \tag{2.10}$$

$$\boldsymbol{u}_{\mathrm{dq,e}}[k] = \boldsymbol{B}^{-1}[k]\left((\boldsymbol{I} - \boldsymbol{A}[k])\,\boldsymbol{i}_{\mathrm{dq}}[k] - \boldsymbol{e}[k]\right), \tag{2.11}$$

which will be critical for the later discussion of feasible operating conditions. Herein, $\boldsymbol{u}_{\mathrm{dq,e}}$ describes the voltage vector that would need to be applied in order to maintain the momentary operating point, i.e.,

$$\boldsymbol{i}_{\mathrm{dq}}[k+1] = \boldsymbol{i}_{\mathrm{dq}}[k] \qquad \text{if } \boldsymbol{u}_{\mathrm{dq}}[k] = \boldsymbol{u}_{\mathrm{dq,e}}[k]. \tag{2.12}$$

Lastly, the mechanical behavior of the PMSM is characterized by torque T and angular speed ω_{me}. Whereas the mechanical speed is in this work assumed to be dictated by the connected mechanical process (i.e., the speed is assumed independent of the drive's dynamics), the torque can be directly inferred from electric quantities:

$$T[k] = \frac{3}{2}p\left(\psi_{\mathrm{d}}[k]i_{\mathrm{q}}[k] - \psi_{\mathrm{q}}[k]i_{\mathrm{d}}[k]\right). \tag{2.13}$$

While conventional torque control methods usually require availability of the electrical and mechanical model, the targeted RL-based control algorithm needs to learn to regulate the motor torque without such expert knowledge. Hence, parameter values for ψ_{p}, R_{s} and $\boldsymbol{L}_{\mathrm{dq}}$ are in the following assumed unavailable and have been introduced for the readers convenience only.

[1] Please note that the discrete-time system matrices $\boldsymbol{A}[k], \boldsymbol{B}[k], \boldsymbol{e}[k]$ are not identical to the continuous-time system matrices $\boldsymbol{A}(t), \boldsymbol{B}(t), \boldsymbol{e}(t)$. Their symbolic distinction is herein limited to the different indication of time dependency.

2.2 Voltage Source Inverter

A voltage source inverter (VSI) is the usual power electronic component that comes to use when supplying an electric drive, with a three-phase, two-level architecture being the standard case (also referred to as B6 inverter topology) [72, 75]. A circuit diagram is provided in Fig. 2.2 with u_{DC} denoting the DC-link voltage, i_{DC} the DC-link current and $s_{\mathrm{a,b,c}}$, $\neg s_{\mathrm{a,b,c}}$ the non-inverted and inverted switching signals for each phase, respectively. For operating the drive, it plays a major role whether the VSI is interfaced on an FCS or a CCS. Corresponding characteristics are discussed in the following.

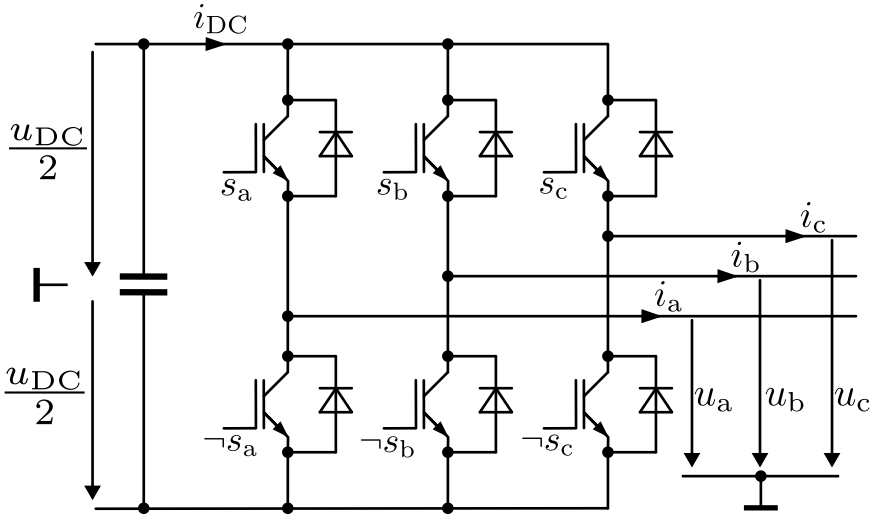


Fig. 2.2: Circuit diagram of the three-phase, two-level voltage source inverter, realized with insulated-gate bipolar transistors

2.2.1 Finite Control Set

In FCS operation, the number of applicable voltage vectors $\boldsymbol{u}_{\mathrm{abc}}$ is limited by the number of distinguishable switching states, as it is assumed that one single switching state persists for the duration of one full control cycle time T_{s}. For the given topology with two levels and three phases, there results a total of $|\mathcal{A}_{\mathrm{FCS}}| = 2^3 = 8$ different switching states, which is depicted in Fig. 2.3. These switching states are detailed in Tab. 2.1, the resulting voltage vectors relate to the switching states via

$$\boldsymbol{u}_{\mathrm{dq}}[k] = u_{\mathrm{DC}}[k]\boldsymbol{Q}_{\mathrm{dq,abc}}(\varepsilon_{\mathrm{el}}[k])\boldsymbol{s}_{\mathrm{abc}}[k], \tag{2.14}$$

with $\boldsymbol{s}_{\mathrm{abc}} = [s_{\mathrm{a}} \quad s_{\mathrm{b}} \quad s_{\mathrm{c}}]^{\top} \in \{0,1\}^3$ denoting the switching vector. Accordingly, the action space $\mathcal{A}_{\mathrm{FCS}}$ evaluates to

$$\mathcal{A}_{\mathrm{FCS}}[k] = \{\boldsymbol{u}_{\mathrm{dq}} \in \mathbb{R}^2 | \boldsymbol{u}_{\mathrm{dq}} = u_{\mathrm{DC}}[k]\boldsymbol{Q}_{\mathrm{dq,abc}}(\varepsilon_{\mathrm{el}}[k])\boldsymbol{s}_{\mathrm{abc}}\}. \tag{2.15}$$

For ease of reading, a switching state index $a \in \{0,\ldots,7\}$ is introduced to refer to the voltage vectors $\boldsymbol{u}_{\mathrm{dq}}(a) \in \mathcal{A}_{\mathrm{FCS}}$ as of Tab. 2.1. In this context, the notation

$$a[k] \in \mathcal{A}_{\mathrm{FCS}}[k], \tag{2.16}$$

is to be understood as an abbreviation for

$$a[k] \in \{0, \cdots, 7\} | \boldsymbol{u}_{\mathrm{dq}}(a[k]) \in \mathcal{A}_{\mathrm{FCS}}[k]. \tag{2.17}$$

Tab. 2.1: Correspondence between the switching index a, the switching state $s_{\mathrm{a,b,c}}$ and applied voltages $u_{\mathrm{a,b,c}}$ and $u_{\alpha,\beta}$ for the given three-phase, two-level VSI in FCS operation (cf. Fig. 2.2)

a	s_{a}	s_{b}	s_{c}	u_{a}	u_{b}	u_{c}	u_{α}	u_{β}
0	0	0	0	$-\frac{u_{\mathrm{DC}}}{2}$	$-\frac{u_{\mathrm{DC}}}{2}$	$-\frac{u_{\mathrm{DC}}}{2}$	0	0
1	1	0	0	$\frac{u_{\mathrm{DC}}}{2}$	$-\frac{u_{\mathrm{DC}}}{2}$	$-\frac{u_{\mathrm{DC}}}{2}$	$\frac{2u_{\mathrm{DC}}}{3}$	0
2	1	1	0	$\frac{u_{\mathrm{DC}}}{2}$	$\frac{u_{\mathrm{DC}}}{2}$	$-\frac{u_{\mathrm{DC}}}{2}$	$\frac{u_{\mathrm{DC}}}{3}$	$\frac{u_{\mathrm{DC}}}{\sqrt{3}}$
3	0	1	0	$-\frac{u_{\mathrm{DC}}}{2}$	$\frac{u_{\mathrm{DC}}}{2}$	$-\frac{u_{\mathrm{DC}}}{2}$	$-\frac{u_{\mathrm{DC}}}{3}$	$\frac{u_{\mathrm{DC}}}{\sqrt{3}}$
4	0	1	1	$-\frac{u_{\mathrm{DC}}}{2}$	$\frac{u_{\mathrm{DC}}}{2}$	$\frac{u_{\mathrm{DC}}}{2}$	$-\frac{2u_{\mathrm{DC}}}{3}$	0
5	0	0	1	$-\frac{u_{\mathrm{DC}}}{2}$	$-\frac{u_{\mathrm{DC}}}{2}$	$\frac{u_{\mathrm{DC}}}{2}$	$-\frac{u_{\mathrm{DC}}}{3}$	$-\frac{u_{\mathrm{DC}}}{\sqrt{3}}$
6	1	0	1	$\frac{u_{\mathrm{DC}}}{2}$	$-\frac{u_{\mathrm{DC}}}{2}$	$\frac{u_{\mathrm{DC}}}{2}$	$\frac{u_{\mathrm{DC}}}{3}$	$-\frac{u_{\mathrm{DC}}}{\sqrt{3}}$
7	1	1	1	$\frac{u_{\mathrm{DC}}}{2}$	$\frac{u_{\mathrm{DC}}}{2}$	$\frac{u_{\mathrm{DC}}}{2}$	0	0

2.2.2 Continuous Control Set

Although switching behavior is inherent to modern power electronic devices such as the utilized VSI, the consideration of each switching procedure can oftentimes complicate the control task and is usually not of major importance to the plant system's performance. Therefore, power electronic components are commonly selected such that their switching frequency is considerably higher than the electric time constants of the supplied system[2]. This allows to neglect the momentary switching behavior in favor of a dynamically averaged view upon electric signals [77], i.e., controlling the average current is prioritized. Utilizing e.g., pulse-width modulation (PWM) or space vector modulation (SVM) [78–80], the switching states of each half bridge can be changed several times during one sampling interval and, hence, the dynamic average of the applied voltage can be commanded arbitrarily in between the elementary vectors (cf. Tab. 2.1, Fig. 2.3), whereas the momentary current ripple can remain unaddressed. The average phase voltage during one sampling period can then be expressed via

$$\boldsymbol{u}_{\mathrm{dq}}[k] = u_{\mathrm{DC}}[k] \boldsymbol{Q}_{\mathrm{dq,abc}}(\varepsilon_{\mathrm{el}}[k]) \boldsymbol{d}_{\mathrm{abc}}[k], \tag{2.18}$$

[2]While already decisive in the context of purely electric systems, the difference between switching frequency and mechanical time constants is even more significant [76].

wherein $\boldsymbol{d}_{\mathrm{abc}} = [d_{\mathrm{a}} \quad d_{\mathrm{b}} \quad d_{\mathrm{c}}]^{\top} \in [0,1]^3$ denotes the duty cycle vector. The corresponding CCS action space evaluates to

$$\mathcal{A}_{\mathrm{CCS}}[k] = \{\boldsymbol{u}_{\mathrm{dq}} \in \mathbb{R}^2 | \boldsymbol{u}_{\mathrm{dq}} = u_{\mathrm{DC}}[k] \boldsymbol{Q}_{\mathrm{dq,abc}}(\varepsilon_{\mathrm{el}}[k]) \boldsymbol{d}_{\mathrm{abc}}\}, \tag{2.19}$$

which can be depicted as a continuous, hexagonal set, cf. Fig. 2.3.

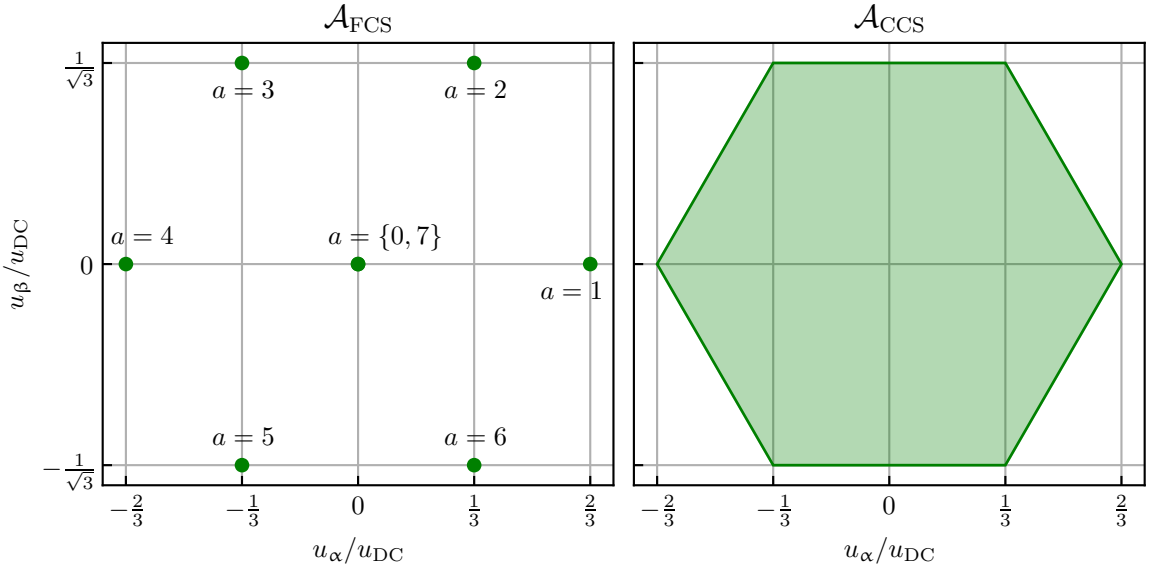


Fig. 2.3: Illustration of $\mathcal{A}_{\mathrm{FCS}}$ and $\mathcal{A}_{\mathrm{CCS}}$ projected to the stator-fixed αβ-reference frame

2.3 Optimal Torque Control

The considered problem of optimal torque control necessitates a controller that chooses a voltage in consideration of the available action space, such that the primary condition of tracking the reference torque T^* has priority over the secondary condition of maximum efficiency. Herein, the latter demand can be generally condensed into the minimization of dissipated power P_{loss}:

$$\begin{aligned} &\min_{\boldsymbol{u}_{\mathrm{dq}}[k]} P_{\mathrm{loss}}[k] \quad \forall k \\ \text{s.t.} \quad &T[k] = T^*[k]. \end{aligned} \tag{2.20}$$

Realistically, however, the torque tracking condition $T = T^*$ cannot be satisfied at all times. The reference T^* may change at an arbitrarily fast rate (e.g., in a step-like manner), whereas the provided torque T is subjected to the drive system's dynamic behavior, resulting in a tracking transient. Hence, drive behavior that is optimal according to (2.20) can usually not be encountered outside of a stationary state. Further, $T = T^*$ may correspond to an operating point of infeasible current or voltage, which will be discussed in Sec. 2.4.1.

Note that the incorporation of loss minimization as a secondary interest is possible because the mechanism of torque generation as a function of the current $T(\boldsymbol{i}_{\mathrm{dq}})$ is not uniquely invertible (cf. (2.13)), and only the installation of a secondary control goal defines a unique mapping $T^* \rightarrow \boldsymbol{i}^*_{\mathrm{dq}}$. Here, the torque control task is oftentimes split into the loss-optimal selection of the desired current operating point $\boldsymbol{i}^*_{\mathrm{dq}}$ (open-loop control) and a subsequent current regulation task (closed-loop control). The utilization of RL, however, suggests a more holistic approach that does not separate operational strategy and current control, which is then known as direct torque control (DTC). Since the formal definition of the control problem (2.20) is not readily suitable for RL, a reward function design that incorporates these control objectives is to be discussed later (cf. Sec. 3.2).

Although the electric and mechanic system behavior are highly characteristic of each individual drive system, the RL-DTC agent is crafted without access to a plant model and must, therefore, learn a policy to optimize (2.20) purely in a data-driven fashion. Since plant-specific knowledge will not be integrated into the design process, this approach is applicable to further special cases of the PMSM[3] described by (2.7), e.g., the surface-mounted PMSM (where $L_{\mathrm{d}} = L_{\mathrm{q}}$), the synchronous reluctance motor (SynRM, where $\psi_{\mathrm{p}} = 0$) and the highly utilized PMSM (where $\boldsymbol{L}_{\mathrm{dq}} = \boldsymbol{L}_{\mathrm{dq}}(\boldsymbol{i}_{\mathrm{dq}})$ is strongly current-dependent). The externally excited synchronous motor follows a different architecture than the PMSM. It is not a special case of the motor model described in Sec. 2.1, but a first adaption of the following concepts has already been transferred to an externally excited synchronous motor application [C16].

2.4 Drive Limitations and Performance Optimization

While the optimal torque control problem in (2.20) defines the basic task of operating the PMSM drive efficiently according to an externally provided reference T^*, several characteristics of safe and feasible PMSM operation can be formulated for the general case to consider the presence of state and action limitations.

2.4.1 Safety-Critical Limitations

Primarily, the limit current i_{lim} defines the current magnitude that is assumed to result in immediate harm to the motor and / or the VSI. Secondarily, it is to be considered that also operation below i_{lim} can lead to significant heating of the drive components, wearing the drive not instantaneously but foreseeable[4]. This circumstance motivates definition of the nominal current i_{n} which is the maximum current magnitude that can be upheld permanently without risking damage. Any operating point $i_{\mathrm{s}} \in]i_{\mathrm{n}}, i_{\mathrm{lim}}[$ overloads the drive

[3] The general case PMSM as of (2.7) with $L_{\mathrm{d}} \neq L_{\mathrm{q}}$ and $\psi_{\mathrm{p}} \neq 0$ is also referred to as interior PMSM, because these characteristics are achieved by burying magnets inside the rotor metal (cf. Fig. 3.1, [81]).

[4] Thermal time constants of electric drives are commonly several orders of magnitude larger than their electric time constants [82], rendering time-limited overloading usually permissible [83].

thermally and must therefore succeed under precise temperature monitoring to prevent overheating. Such an application scenario is outside the scope of this work and, hence, $i_\mathrm{s} \leq i_\mathrm{n}$ is targeted as operating range for the given PMSM.

Apart from the current limitations, several voltage-related conditions are of interest. Firstly, the VSI's voltage limit (2.15), (2.19) is physically impossible to violate (cf. Fig. 2.3), which must be considered within any closed- or open-loop control application when selecting an operating point. Further, the available voltage corresponds directly to the drive's controllability. In the case of insufficient voltage, the magnitude of the equilibrium voltage $||\boldsymbol{u}_\mathrm{dq,e}||_2$ exceeds the voltage magnitude that the VSI can provide. Consequently, the momentary operating point may become unsustainable and the stator current i_s could increase unintentionally. Therefore, it is a secondary safety concern to keep $\boldsymbol{i}_\mathrm{dq}$ within the controllable subspace.

The mathematical formulation of the aforementioned conditions results in

$$i_\mathrm{s}[k] \leq i_\mathrm{n} \qquad \forall\, k, \tag{2.21a}$$

$$\boldsymbol{u}_\mathrm{dq,e}[k] \in \mathcal{A}[k] \qquad \forall\, k. \tag{2.21b}$$

The condition (2.21b) is commonly softened in the sense of a dynamically averaged perspective. Herein, it is sufficient for the VSI to provide the equilibrium voltage on average over the course of one (electrical) rotation, but not necessarily at each individual sampling instant. Specifically in an FCS context, it would be rarely possible to match $\boldsymbol{u}_\mathrm{dq,e}$ precisely by means of the VSI's switching states and, hence, a criterion of the form (2.21b) is not directly applicable to that scenario. Generally, however, these safety-related limitations can be guaranteed to sufficient extent when a superimposed safeguarding mechanism is utilized, as originally proposed in [A2] and [A3]. The setup of such a safeguard on the FCS and the CCS will be presented in Sec. 4.1 and Sec. 5.1, respectively.

2.4.2 Efficiency-Boosting Characteristics: Maximum Torque per Current and Maximum Torque per Voltage

In addition to the safety-ensuring conditions, a set of efficiency-related operating characteristics can be defined without necessitating parameter knowledge. For PMSMs, efficient operation is restricted to operating points with $i_\mathrm{d} \leq 0$. A positive d current is not principally dangerous but leads to a larger current drain than necessary (and therefore to higher losses), whereas the available torque range remains the same[5], and should, hence, be avoided if possible.

Moreover, the consideration of the well established maximum-torque-per-current (MTPC)[6] and maximum-torque-per-voltage (MTPV) characteristic [84] is desired.

[5]For SynRMs, operation at positive and negative i_d is equally feasible. The former is not considered in the scope of this work in favor of general applicability to PMSMs.

[6]The widely-spread mislabel 'maximum-torque-per-Ampere (MTPA)' is avoided due to its inconsistency of relating quantities to units.

Whenever voltage reserve is sufficient, operation with MTPC characteristic targets minimization of ohmic losses when adhering to the torque reference $T = T^*$. At higher speed, MTPC operation may not be possible due to the diminished voltage reserve. In this case, MPTV operation is targeted to maintain the commanded torque. In conventional control approaches, the MTPC and MTPV characteristic are computed by means of the drive model parameters, whereas the proposed RL-based DTC is designed to learn the optimal operation characteristics in a data-driven fashion.

Apart from ohmic copper losses, further loss mechanisms such as iron and switching losses are not considered within the MTPC approach (i.e., it is assumed that $\hat{P}_{\mathrm{loss}} = i_{\mathrm{s}}^2 R_{\mathrm{s}}$). Hence, it can merely be expected to be an approximation of maximum efficiency operation. Model-free determination of the overall system efficiency would demand a comprehensive electric and mechanic power measurement setup, which is not targeted within this contribution. Application of this simplification yields

$$\begin{aligned} &\min_{\boldsymbol{u}_{\mathrm{dq}}[k]} i_{\mathrm{s}}[k] \quad \forall k \\ \text{s.t.} \quad &T[k] = T^*[k], \\ &i_{\mathrm{d}}[k] \leq 0, \end{aligned} \tag{2.22}$$

whereas the safety-related conditions (2.21) still apply.

Since performance-related conditions do not pose a safety risk when violated, they are only to be considered within the configuration of the controller training goal (cf. Sec. 3.2). Note that the stated conditions ($i_{\mathrm{d}} \leq 0$, MTPC, MTPV) only apply to the stationary state. During transients, deviating behavior can still be favorable, e.g., in terms of reaction time [85].

2.5 Reinforcement Learning: a Brief Overview

This section aims at informing the reader about the very fundamentals of reinforcement learning (RL) [86, 87], as its application to power system control was a development of the younger past at the time of the making of this thesis. As the following considerations reflect the workings of RL in general, the direct connection to power systems is avoided in this section and a more general notation is utilized.

In the given context, it is assumed that the controlled system and the targeted optimal control policy are deterministic. While the available theory fully covers the general, stochastic case, it is pragmatic to assume random and unpredictable events to be negligible in technical plants. In the following, the notation is streamlined accordingly by omitting random variables and expected value operators.

RL is a machine learning discipline that is suitable for optimal decision making and feedback control in dynamic environments. At its core, it is about maximizing the return g, which is defined by the cumulative future reward r:

$$g[k] = r[k+1] + \gamma r[k+2] + \gamma^2 r[k+3] + \gamma^3 r[k+4] + \gamma^4 r[k+5] + \cdots . \tag{2.23}$$

Virtually, this series continues indefinitely, meaning that the optimization of g considers an infinite time horizon. Practically, the discount factor $\gamma \in [0, 1[$ ensures numerical convergence of the series and represents a degree of freedom that can be tuned to optimize the control behavior in terms of a short sighted or a far sighted view upon the task.

The reward function r describes the momentary performance of the plant system in terms of the control task. It is, therefore, only a snapshot that does not consider development of the system state over time[7]. Whereas the usual perspective on RL problems in computer science views r as an inherent and unchangeable component of the control task, it is seen as an important degree of freedom from an engineering point of view. As the form of r defines the actual control task and describes what is to be understood as 'good' or 'bad' performance, proper design of the reward function is critical for achieving the desired plant behavior and allows to incorporate expert knowledge, if applicable. While usual comparable real-time capable, optimal CCS-MPC approaches are limited to quadratic cost functions in conjunction with linear systems [89, 90], reward functions in RL are not subjected to such conditions and may contain case distinctions, which is exploited in the later presented reward design (cf. Sec. 3.2). FCS-MPC applications are in the same way unrestricted in terms of plant behavior and objective function, but are limited due to an exponential dependency between prediction horizon and computational complexity.

For systems that satisfy the Markov property

$$\boldsymbol{x}[k+1] = \boldsymbol{f}(\boldsymbol{x}[k], \boldsymbol{x}[k-1], \cdots, \boldsymbol{u}[k], \boldsymbol{u}[k-1], \cdots) = \boldsymbol{f}(\boldsymbol{x}[k], \boldsymbol{u}[k]), \tag{2.24}$$

the momentary system state $\boldsymbol{x}[k]$ and input $\boldsymbol{u}[k]$ suffice to uniquely define the successor state $\boldsymbol{x}[k+1]$. This allows to define the action-value function q such that

$$q(\boldsymbol{x}[k], \boldsymbol{u}[k]) = g[k], \tag{2.25}$$

which means that an algebraic relation exists between measured state $\boldsymbol{x}$, commanded action $\boldsymbol{u}$ and achievable return g. Determination, or rather estimation of q is the main challenge within most RL-based control algorithms[8] for which, apart from the Markov property (2.24), no further assumption or specific knowledge on the dynamic function $\boldsymbol{f}$ is needed. Once a sufficiently accurate estimate $\hat{q}$ is available, the optimal action $\boldsymbol{u}^*$ can be extracted via

$$\boldsymbol{u}^*[k] = \arg\max_{\boldsymbol{u} \in \mathcal{A}} \hat{q}(\boldsymbol{x}[k], \boldsymbol{u}). \tag{2.26}$$

If $\mathcal{A} = \mathcal{A}_{\mathrm{FCS}}$ is finite and, hence, an FCS task is to be resolved, the arg max operation can be evaluated in an exhaustive-search manner by iterating over all elements in $\mathcal{A}_{\mathrm{FCS}}$

[7] Comparing conventional optimal control and RL, the return function g takes the role of the cost function, while the reward function r takes the role of the stage cost [88].

[8] Apart from finding a control policy, the domain of RL also offers algorithms that target the pure evaluation of preexisting controllers.

and determining the resulting action value estimate $\hat{q}$. If, however, the action space[9] is continuous $\mathcal{A} = \mathcal{A}_{\mathrm{CCS}}$, the arg max operation cannot generally be evaluated trivially, and utilization of corresponding optimization algorithms is rarely feasible at run-time. Instead, a policy function $\boldsymbol{\pi}(\boldsymbol{x})$ is to be determined such that

$$\boldsymbol{u}^*[k] = \boldsymbol{\pi}(\boldsymbol{x}[k]) = \underset{\boldsymbol{u}\in\mathcal{A}_{\mathrm{CCS}}}{\arg\max}\, \hat{q}(\boldsymbol{x}[k], \boldsymbol{u}) \quad \forall \boldsymbol{x}, \tag{2.27}$$

which defines the optimal policy $\boldsymbol{\pi}$ as a (generally nonlinear) feedback control law.

The adaption of action-value estimator $\hat{q}$ and assumed optimal policy $\boldsymbol{\pi}$ happens on the basis of measured state transition experiences $\mathcal{E}$

$$\mathcal{E} = \{\boldsymbol{x}[k], \boldsymbol{u}[k], r[k+1], \tau[k+1], \boldsymbol{x}[k+1]\}, \tag{2.28}$$

rendering the learning process a data-driven algorithm. Herein, τ denotes a termination flag that labels discontinuation of the control task, which plays a role for handling of the training data within the specifically employed algorithms (cf. Sec. A.2, Sec. A.3).

A plethora of RL algorithms has emerged to tackle the challenge of approximating the action-value function $\hat{q}$ and – if necessary – the policy function $\boldsymbol{\pi}$. In this work, two such algorithms will be utilized to solve the RL control problem: the deep q network (DQN) for the FCS, and the deep deterministic policy gradient (DDPG) for the CCS scenario. Please note that the specific algorithm that determines $\hat{q}$ and $\boldsymbol{\pi}$ is not critical for the approaches that are presented in the remainder of this thesis. The main concerns of the proposed RL torque control scheme are the definition of the reward function r (which is needed within every RL algorithm) and the safeguarding measures (which are entirely independent of the method used to determine $\boldsymbol{u}^*$). Therefore, a detailed explanation of the specifically utilized algorithms is spared at this point and the interested reader is referred to Sec. A.2 and Sec. A.3, where a more elaborate introduction to the DQN and the DDPG is provided, and hyperparameter selection is discussed. At the time of publishing of this work, it is likely that new algorithms with further advantages have emerged, and the reader is hereby encouraged to combine the proposed reward design and safeguarding measures with any RL algorithm of their choice.

2.5.1 Exploration and Exploitation

Independently of the specifically employed algorithm, it is necessary to ensure sufficient coverage of the state and action space during training to enable actual learnability of the optimal policy. Instructively, the optimal action for a given state cannot be expected to be found in application when it has never been tried and evaluated during training. Analytically, the reason for this is that action-values $\hat{q}$ are assigned to and learned from the seen state-action tuples $(\boldsymbol{x}, \boldsymbol{u})$, which should hence be as diverse as possible. To

[9] The domain of computer science utilizes the term 'action space' for the control set, whose naming originates from engineering sciences. Both terms are used synonymously within this work.

a limited extent, it is therefore of interest to allow randomness in the action selection process during the training phase:

$$\boldsymbol{u}^*[k] = \begin{cases} \arg\max_{\boldsymbol{u}\in\mathcal{A}} \hat{q}(\boldsymbol{x}[k], \boldsymbol{u}) & \text{during application,} \\ \in_{\mathrm{R}} \mathcal{A} & \text{during training,} \end{cases} \tag{2.29}$$

whereas $\in_{\mathrm{R}}$ denotes random sampling from the specified set. The specific randomization strategy during the training phase depends on the form of $\mathcal{A}$ (FCS or CCS). The specifically employed exploration strategies are discussed in Sec. A.2 and Sec. A.3.

Overall, the random initialization of the employed function approximators and the superimposed exploration noise may result in significant plant excitation. While usually beneficial for gathering training data, the compliance to plant limitations cannot be guaranteed with this setup. On the other hand, insufficient excitation oftentimes results in a low-performance controller, because better suited control commands have not been explored during the training. Hence, safety is a widely discussed issue for the application of RL training in real-world experiments [91].

2.5.2 Function Approximation

For the following discussion, and particularly for efficient computation (cf. Sec. 3.4.2.1), it is of importance in which form $\hat{q}$ and / or $\boldsymbol{\pi}$ are provided to enable real-time capable inference. As can be taken from the names of the utilized algorithms – DQN and DDPG – deep artificial neural networks (ANNs) are the established method of approximating the state-action value function q and the policy function $\boldsymbol{\pi}$. In this thesis, only the feedforward multilayer perceptron (MLP) architecture [92] is investigated for this purpose, which means that the function approximator itself is considered stateless[10].

As the name implies, an MLP consists of several layers with identical structure:

$$\begin{aligned} \boldsymbol{y}_1 &= f_{\mathrm{act},1}(\boldsymbol{K}_1\boldsymbol{x} + \boldsymbol{b}_1), \\ \boldsymbol{y}_2 &= f_{\mathrm{act},2}(\boldsymbol{K}_2\boldsymbol{y}_1 + \boldsymbol{b}_2), \\ &\vdots \\ \boldsymbol{y} = \boldsymbol{y}_l &= f_{\mathrm{act},l}(\boldsymbol{K}_l\boldsymbol{y}_{l-1} + \boldsymbol{b}_l). \end{aligned} \tag{2.30}$$

Herein, $\boldsymbol{x}$ denotes the input and $\boldsymbol{y}$ denotes the output of the network with $\boldsymbol{y}_{1,2,\dots}$ denoting the output of each individual layer and l being the number of layers. Each layer consists of an affine linear transformation with weight matrix $\boldsymbol{K}$ and bias vector $\boldsymbol{b}$, which contain the trainable network weights. The scalar activation function f_{act} is a generally nonlinear but stateless function which is applied in element-wise fashion [96].

In this work, MLPs are employed to approximate the action value q and the policy function $\boldsymbol{\pi}$, leading to the notation $\hat{q}_{\boldsymbol{\theta}}$ for the action-value network and $\boldsymbol{\pi}_{\boldsymbol{\zeta}}$ for the policy network.

[10] However, a multitude of stateful ANN network architectures is available, which are usually based on explicit [93] or implicit [94, 95] recursion of intermediate signals.

Herein, $\boldsymbol{\theta}$ and $\boldsymbol{\zeta}$ refer to the entirety of corresponding network weights, i.e., the complete set of weight matrices $\boldsymbol{K}_{1,\dots,l}$ and bias vectors $\boldsymbol{b}_{1,\dots,l}$ that are needed to define either $\hat{q}_{\boldsymbol{\theta}}$ or $\boldsymbol{\pi}_{\boldsymbol{\zeta}}$.

2.5.3 Feature Engineering

To simplify learning tasks, the input values to the employed function approximators are often preprocessed to enrich the informational content and include expert knowledge, which is known as feature engineering [97]. Formally, a feature function $\boldsymbol{\phi}$ is defined, which is then used to compute an observation vector $\boldsymbol{o}$ to replace the state input $\boldsymbol{x}$ within the function approximators:

$$\begin{aligned} &\hat{g}[k] = \hat{q}_{\boldsymbol{\theta}}(\boldsymbol{x}[k], \boldsymbol{u}[k]) \\ &\boldsymbol{u}[k] = \boldsymbol{\pi}_{\boldsymbol{\zeta}}(\boldsymbol{x}[k]) \end{aligned} \quad \rightarrow \quad \begin{aligned} &\hat{g}[k] = \hat{q}_{\boldsymbol{\theta}}(\boldsymbol{o}[k], \boldsymbol{u}[k]), \\ &\boldsymbol{u}[k] = \boldsymbol{\pi}_{\boldsymbol{\zeta}}(\boldsymbol{o}[k]), \end{aligned} \tag{2.31}$$

with

$$\boldsymbol{o}[k] = \boldsymbol{\phi}(\boldsymbol{x}[k], \boldsymbol{x}[k-1], \dots, \boldsymbol{u}[k-1], \boldsymbol{u}[k-2], \dots). \tag{2.32}$$

As can be seen, $\boldsymbol{\phi}$ can be designed to incorporate information from past state transitions into the momentary observation vector. This trait can be exploited to craft $\boldsymbol{o}$ such that the Markov property (2.24) is satisfied whereas the momentary system-inherent measurement $\boldsymbol{x}$ would not suffice

$$\boldsymbol{x}[k+1] = \boldsymbol{f}(\boldsymbol{x}[k], \boldsymbol{x}[k-1], \cdots, \boldsymbol{u}[k], \boldsymbol{u}[k-1], \cdots) \rightarrow \boldsymbol{o}[k+1] = \tilde{\boldsymbol{f}}(\boldsymbol{o}[k], \boldsymbol{u}[k]). \tag{2.33}$$

In such cases, proper feature engineering is absolutely necessary to create the conditions under which RL is possible. Again, please note that neither $\boldsymbol{f}$ nor $\tilde{\boldsymbol{f}}$ need to be known in order to proceed, however, the design process of $\boldsymbol{\phi}$, i.e., the sensible composition of $\boldsymbol{o}$ usually benefits from available expert knowledge, which can be utilized for the selection, normalization and transformation of available process data. The training data is then altered accordingly:

$$\begin{aligned} \mathcal{E} = \{&\boldsymbol{x}[k], \boldsymbol{u}[k], r[k+1], \tau[k+1], \boldsymbol{x}[k+1]\} \\ \rightarrow \quad &\mathcal{E} = \{\boldsymbol{o}[k], \boldsymbol{u}[k], r[k+1], \tau[k+1], \boldsymbol{o}[k+1]\}. \end{aligned} \tag{2.34}$$

3 Reinforcement Learning Direct Torque Control

After reviewing the principles of PMSM drive setups and the fundamentals of RL, this chapter deals with the preliminary considerations for bringing these two topics together. Firstly, this includes observation design, which is necessary to satisfy the Markov property and enable state-value estimation in the first place. Secondly, reward design, that needs to encode the control task without parameter knowledge, i.e., independently of the typically utilized model parameters of stator resistance R_s, permanent magnet flux linkage ψ_p or inductance matrix $\boldsymbol{L}_\mathrm{dq}$. Thirdly, data-driven system identification, that is later utilized to distinguish safe from unsafe actions to avoid the latter. The succeeding considerations are based on the original publications [A1] and [A2].

3.1 Observation Design

As introduced in the previous chapter, proper design of the observation vector $\boldsymbol{o}$ is often necessary to ensure that the Markov property holds (cf. (2.33)), which is essential for a successful RL control setup [86]. The corresponding considerations for the PMSM drive are discussed in the following.

On the basis of the previously discussed system description with consideration of the dq-coordinate transformation (2.1), the motor dynamics (2.7) and the inverter behavior (cf. Tab. 2.1, Fig. 2.3), the measurable state quantities[1] summarized in $\boldsymbol{x}$ need to be available to allow proper selection of a sensible controlling voltage $\boldsymbol{u}_\mathrm{dq}$:

$$\boldsymbol{x}[k] = \begin{bmatrix} i_\mathrm{d}[k] & i_\mathrm{q}[k] & \omega_\mathrm{me}[k] & \varepsilon_\mathrm{el}[k] & u_\mathrm{DC}[k] \end{bmatrix}^\top . \tag{3.1}$$

Herein, the potentially different value range of features might complicate the subsequent training of ANNs [99]. Assuming that corresponding value ranges are known (e.g., from

[1] Commonly, the angular speed ω_me is not directly measured by a distinct speed sensor but must be inferred by taking past angle measurements ε_el into account. Since this is a standard procedure that can be easily resolved by employing a phase-locked loop [98], it is not further discussed within this contribution.

technical documentation about the drive's operational limitations), this issue can be tackled by applying min-max normalization. The rotor angle $\varepsilon_{\mathrm{el}}$ is not bounded by operational constraints, rendering a min-max transformation infeasible for normalization. Hence, the angular signal is bounded via representation in two-dimensional cartesian coordinates $\begin{bmatrix}\cos(\varepsilon_{\mathrm{el}}) & \sin(\varepsilon_{\mathrm{el}})\end{bmatrix}$ without loss of relevant information[2].

Naturally, the torque reference signal T^* needs to be appended, because it is critical for the definition of the control task (2.20) while not being an inherent physical quantity of the drive system. With $\boldsymbol{i}_{\mathrm{dq}}$ already being part of the state vector $\boldsymbol{x}$, appending the stator current i_{s} is arguably redundant information, but since its correspondence to the drive limitations is linear (whereas the relation of $\boldsymbol{i}_{\mathrm{dq}}$ to i_{n} or i_{lim} is not), it can be expected to improve the learning behavior.

Lastly, the real-world drive application underlies several systematic and parasitic time delays that stem from the digital controller realization and the (slower) dynamics of the torque sensor, with a more in-depth discussion being delivered in Sec. 3.4. From a controller point of view, these delays pose as unmeasureable states, whose behavior is unknown but deterministic. To compensate this lack of information, the past commanded voltages $\boldsymbol{u}_{\mathrm{dq}}[k-1, k-2, k-3]$ are appended to $\boldsymbol{o}$ [100]. This approach can be compared to the employment of classical state observers in model-based control (with most prominent representants being [101, 102]), wherein it is likewise targeted to estimate the inaccessible momentary states by means of past measurements and inputs and – other than in RL – known system dynamics, whereas here, the dynamics are assumed unknown while the state estimation is not required to be explicit or physically interpretable.

The observation vector design, hence, results to

$$\boldsymbol{o}[k] = \begin{bmatrix} \frac{\omega_{\mathrm{me}}[k]}{\omega_{\mathrm{me,lim}}} & \frac{\boldsymbol{i}_{\mathrm{dq}}^{\top}[k]}{i_{\mathrm{lim}}} & \frac{3\boldsymbol{u}_{\mathrm{dq}}^{\top}[k-1]}{2u_{\mathrm{DC}}[k]} & \frac{3\boldsymbol{u}_{\mathrm{dq}}^{\top}[k-2]}{2u_{\mathrm{DC}}[k]} & \frac{3\boldsymbol{u}_{\mathrm{dq}}^{\top}[k-3]}{2u_{\mathrm{DC}}[k]} \\ \cos(\varepsilon_{\mathrm{el}}[k]) & \sin(\varepsilon_{\mathrm{el}}[k]) & 2\frac{i_{\mathrm{s}}[k]}{i_{\mathrm{lim}}} - 1 & 2\frac{u_{\mathrm{DC}}[k] - u_{\mathrm{DC,min}}}{u_{\mathrm{DC,max}} - u_{\mathrm{DC,min}}} - 1 & \frac{T^*[k]}{T_{\mathrm{lim}}} \end{bmatrix}^{\top}, \tag{3.2}$$

which limits all entries to the value range $[-1, 1]$ and includes sufficient information to satisfy the Markov property in real-world application.

While it is necessary to have a torque sensor available for the training phase (cf. Sec. 3.2), omitting T within $\boldsymbol{o}$ allows to later apply the resulting control scheme without corresponding measurement equipment. This is motivated by the usual absence of torque sensors in commercial applications, as they add to the cost, space and weight demand, as well as introducing potential for failure [103, 104], and is possible because the correspondence

[2] In practice, output signals of angular sensors are oftentimes discontinuous and map $\varepsilon_{\mathrm{el}}$ to the interval $[0, 2\pi]$, i.e., the sensor signal may step periodically during operation. This discontinuity does not carry relevant information to the torque control task and might aggravate the ANN training if left included. By utilizing the cartesian representation, this characteristic is fully addressed as it additionally continualizes the angle information in such cases.

between current $\boldsymbol{i}_{\mathrm{dq}}$ and torque T is algebraic (cf. (2.13)), i.e., $\boldsymbol{i}_{\mathrm{dq}}$ uniquely defines the torque T.

3.2 Reward Design

The reward function r is to be designed such that the optimization problem denoted in (2.22) is encoded without the need of parameter knowledge. This means that the highest achievable reward should correspond to $T = T^*$ at minimum required i_{s}. As a secondary condition, the range of estimated action values is to be fixed to $q \in [-1, 1]$, resulting in the utilized critic network $\hat{q}_{\boldsymbol{\theta}}$ to have normalized outputs, which is beneficial for the training [92, 99]. In an assumed case of immediate system termination, no look-ahead action value will be considered (cf. (2.23)), and the terminal action value is equivalent to the terminal reward:

$$q_{\mathrm{term}} = r_{\mathrm{term}} \overset{!}{=} -1. \tag{3.3}$$

This can be interpreted as a penalty that is applied whenever the safety-critical system limitations are violated, i.e., the RL agent is encouraged to stay within the safe subset of the state space. Note that this definition is only feasible if termination of the control task is considered undesired, which is usually the case for closed-loop control of technical plants that are assumed to be permanently active unless an emergency shutdown is necessary[3].

For ongoing operation, the normalization can be achieved when considering the geometric series for best and worst case operation (cf. (2.23)), which are to be rewarded by $r_{\max}$ and $r_{\min}$, respectively:

$$\begin{aligned} & q_{\max} = \sum_{i=k}^{\infty} \gamma^{i-k} r_{\max} = \frac{r_{\max}}{1-\gamma} \overset{!}{=} 1 \\ \Rightarrow \quad & r_{\max} = 1 - \gamma, \\ & q_{\min} = \sum_{i=k}^{\infty} \gamma^{i-k} r_{\min} = \frac{r_{\min}}{1-\gamma} \overset{!}{=} -1 \\ \Rightarrow \quad & r_{\min} = -(1-\gamma). \end{aligned} \tag{3.4}$$

Hence, during ongoing operation (i.e., in the absence of a termination event), the reward should be defined on the interval $r \in [-(1-\gamma), 1-\gamma]$. As an auxiliary signal, the terminal flag $\tau \in \{0, 1\}$ is introduced to distinguish terminal states from ongoing operation in the upcoming definitions. With the computed reward range it is now possible to design the reward function in dependency of the feasible operation region as discussed in Sec. 2.4. The resulting reward gradients that will ultimately determine the targeted operation point are depicted in Fig. 3.1.

[3] In settings from the realm of optimal trajectory planning, termination events could also correspond to successful completion of a task. Naturally, this would advise for maximum rather than minimum reward.

$\mathbb{E}$: **Excess Current Region**
Entering this region will trigger an emergency system shutdown, which is a termination event. The learning is temporarily discontinued and the drive system must be re-initialized. Minimum reward as of (3.3) to discourage system termination is defined:

$$\begin{aligned} &\text{if } i_{\lim} \leq i_{\mathrm{s}}[k]: \\ &r_{\mathbb{E}}[k+1] = -1, \quad \tau[k+1] = 1. \end{aligned} \tag{3.5a}$$

$\mathbb{D}$: **Region of Short-Time Overcurrent**
Reward rises with decreasing stator current i_{s} for safety reasons, i.e., the agent is encouraged to reduce i_{s} in order to not overload the system, while an emergency shutdown is not being triggered:

$$\begin{aligned} &\text{if } i_{\mathrm{s}}[k] \in]i_{\mathrm{n}}, i_{\lim}[: \\ &r[k+1] = \left(1 - \frac{i_{\mathrm{s}}[k] - i_{\mathrm{n}}}{i_{\lim} - i_{\mathrm{n}}}\right) \frac{1-\gamma}{2} - (1-\gamma), \\ &\Rightarrow r_{\mathbb{D}}[k+1] \in \left[-(1-\gamma), -\frac{1-\gamma}{2}\right], \qquad \tau[k+1] = 0. \end{aligned} \tag{3.5b}$$

The specification of i_{n} and $i_{\lim}$ could herein be extracted from nameplate data, but could also be configured more conservatively if desired. Please note that this approach does not take the time under overload into consideration, i.e., long-term overload operation might still occur if favorable controller behavior is learned too slowly. Since overloading is not to be covered within this work (cf. Sec. 2.4.1), entire avoidance of this operation region is targeted and will be further addressed by the superimposed safeguard (cf. Sec. 3.3).

$\mathbb{C}$: **Region of Unfavorable Efficiency**
Although safe operation could be achieved for positive i_{d} current, the resulting efficiency for synchronous drives is inferior to operating the drive in the left $\boldsymbol{i}_{\mathrm{dq}}$-half plane (cf. Sec. 2.4.2). Hence, reward is defined anti-proportional to i_{d} as long as i_{d} exceeds the tolerated d current boundary $i_{\mathrm{d+}}$:

$$\begin{aligned} &\text{if } i_{\mathrm{s}}[k] \leq i_{\mathrm{n}} \text{ and } 0 \leq i_{\mathrm{d+}} < i_{\mathrm{d}}[k]: \\ &r[k+1] = \left(1 - \frac{i_{\mathrm{d}}[k] - i_{\mathrm{d+}}}{i_{\mathrm{n}} - i_{\mathrm{d+}}}\right) \frac{1-\gamma}{2} - \frac{1-\gamma}{2}, \\ &\Rightarrow r_{\mathbb{C}}[k+1] \in \left[-\frac{1-\gamma}{2}, 0\right], \qquad \tau[k+1] = 0. \end{aligned} \tag{3.5c}$$

Herein, $i_{\mathrm{d+}} \geq 0$ is a design parameter. Particularly in FCS applications, allowing low positive d-axis current is recommended to make zero-average i_{d} achievable.

$\mathbb{B}$: **Torque Tracking Region**
If the absolute torque tracking error $|T^* - T|$ is intolerably large, reward rises with decreasing tracking error to encourage accurate reference tracking. For that, a tracking error tolerance T_{tol} is introduced as a design parameter:

$$\begin{aligned}
&\text{if } i_\mathrm{s}[k] \leq i_\mathrm{n} \text{ and } i_\mathrm{d}[k] \leq i_\mathrm{d+} \text{ and } T_\mathrm{tol} < |T^*[k] - T[k]| : \\
&r[k+1] = \left(1 - \left|\frac{T^*[k] - T[k]}{2T_\mathrm{lim}}\right|\right) \frac{1-\gamma}{2}, \\
&\Rightarrow r_\mathbb{B}[k+1] \in \left[0, \frac{1-\gamma}{2}\right], \qquad \tau[k+1] = 0.
\end{aligned} \tag{3.5d}$$

$\mathbb{A}$: **Reference Torque Isoline**
While the torque tracking error is tolerably small, reward rises with decreasing stator current to increase the efficiency (cf. Sec. 2.4.2):

$$\begin{aligned}
&\text{if } i_\mathrm{s}[k] < i_\mathrm{n} \text{ and } i_\mathrm{d}[k] < i_\mathrm{d+} \text{ and } |T^*[k] - T[k]| \leq T_\mathrm{tol} : \\
&r[k+1] = \left(1 - \frac{i_\mathrm{s}[k]}{i_\mathrm{lim}}\right) \frac{1-\gamma}{2} + \frac{1-\gamma}{2}, \\
&\Rightarrow r_\mathbb{A}[k+1] \in \left[\frac{1-\gamma}{2}, 1-\gamma\right], \qquad \tau[k+1] = 0.
\end{aligned} \tag{3.5e}$$

The reward intervals $\mathbb{E}, \ldots, \mathbb{A}$ as of (3.5) cover the entire motor operation region including both, constant torque and flux weakening operation. They represent the control objective from (2.22) without requiring knowledge of the specific drive behavior, i.e., motor parameters or look-up tables are not utilized to compute r. Only the nominal and limit data of the drive should be (approximately) known in order to be able to perform the presented normalization of the reward functions. Note that the torque tracking tolerance T_tol is herein usually dictated by the mechanical process or system the drive is connected to. While it should generally be selected as small as possible, mechanical oscillation of the drive train (and corresponding oscillation of the torque measurement signal T) might conflict the optimization according to $\mathbb{A}$, if the oscillation magnitude relates infeasibly to the magnitude of T_tol. Apart from the mechanical oscillation capability, the switching behavior in both, FCS and CCS applications, will result in current ripple[4] which will in turn lead to torque ripple, posing as consistent excitation for the drive train. The magnitude of the current ripple is herein a sensible orientation for parameterizing the d current leeway $i_\mathrm{d+}$.

Depending on the specific drive architecture, the resulting reward gradient $\nabla_{\boldsymbol{i}_\mathrm{dq}} r$ has a different appearance within the current plane. This is schematically depicted within Fig. 3.1

3.2.1 Interdependency of Reward and Safeguard

The previously presented reward design does not yet take into account that the plant might be protected by means of a safeguard, which intervenes when limitations of the drive system are conflicted by the commanded voltages. During such events, the safeguard

[4]The employment of regular sampling in CCS applications will usually mask most of the current ripple in the measurement signal [105].

algorithm may overwrite the control command that was selected by the RL agent in order to sustain operation of the drive system in a safe state. If left unaddressed, the corresponding reward would reflect a safe and uncritical state transition, despite the fact that the original control command was potentially harmful.

In order to discourage problematic action selection, it is therefore sensible to penalize safeguard interventions by means of a reduced reward r. The corresponding extension to the reward distribution looks as follows:

$\mathbb{E}_\mathrm{S}$: **Safeguard Intervention to Prevent Excess Current**
The RL agent commanded an input voltage that would have resulted in a system-terminating overcurrent. Minimum reward is assigned to discourage such behavior:

$$\begin{aligned} &\text{if } i_\mathrm{lim} \leq \hat{i}_\mathrm{s}[k] : \\ &r_{\mathbb{E}_\mathrm{S}}[k+1] = -1, \qquad \tau[k+1] = 0. \end{aligned} \tag{3.6a}$$

Note that, as the system termination is prevented, the termination flag τ is not set. Accordingly, in terms of accumulatable reward, this scenario is analytically favorable over an actual system shutdown as of $\mathbb{E}$, because further reward may still be gathered after this event. Such intervention is critical for timely completion of the training phase, as it avoids securing the system by means of an emergency shutdown, which would otherwise add to the time demand of the procedure.

$\mathbb{D}_\mathrm{S}$: **Safeguard Intervention to Prevent Short-Time Overcurrent**
The RL agent commanded an input voltage that would have resulted in i_s exceeding i_n without provoking a system shutdown. While such behavior should be discouraged, it should not be penalized as severe as actually entering $\mathbb{D}$ (with $\hat{i}_\mathrm{s}$ taking the place of i_s), because activating the safeguard is always preferable in terms of system safety. Naturally, it should also not be rewarded any better than region $\mathbb{D}$, because this could encourage to trigger safeguard intervention more often when close to i_n. Taking both of these conditions into account, such an event must be rewared with $\max_\mathbb{D} r$:

$$\begin{aligned} &\text{if } \hat{i}_\mathrm{s}[k] \in]i_\mathrm{n}, i_\mathrm{lim}[: \\ &r_{\mathbb{D}_\mathrm{S}}[k+1] = -\frac{1-\gamma}{2}, \qquad \tau[k+1] = 0. \end{aligned} \tag{3.6b}$$

$\mathbb{C}_\mathrm{S}$: **Safeguard Intervention to Prevent Voltage Deficiency**
In this case, the commanded input voltage would have lead to an operating point that is not sustainable by means of the available action space. Such state transitions are to be prevented to avoid unwanted and uncontrollable increase of the stator current i_s. Corresponding events are of concern in regions $\mathbb{C}, \mathbb{B}$ and $\mathbb{A}$, as expected operating points in $\mathbb{E}$ and $\mathbb{D}$ should trigger the previously discussed reward assignments instead.

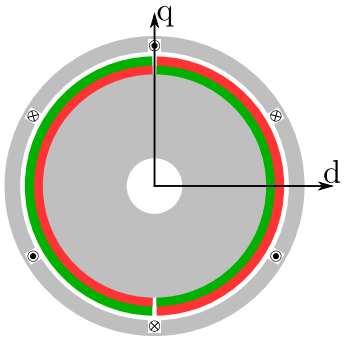
$$\begin{aligned} &\text{if } i_\mathrm{s}[k] \leq i_\mathrm{n} \text{ and } i_\mathrm{d}[k] \leq i_{\mathrm{d}+} \text{ and } \hat{\boldsymbol{u}}_{\mathrm{dq,e}}[k+1] \notin \mathcal{A}_{\mathrm{FCS/CCS}} : \\ &r_{\mathbb{C}_\mathrm{S}}[k+1] = 0, \qquad \tau[k+1] = 0. \end{aligned} \tag{3.6c}$$

Note that, in order to allow the RL-DTC to learn from mistakes in the presence of a safeguard, the non-safeguarded action $\boldsymbol{u}_{\mathrm{dq}}$ must be memorized within the experience tuple $\mathcal{E}$ (2.28) even though it may have been overruled by a safeguard action $\boldsymbol{u}_{\mathrm{dq,S}}$:

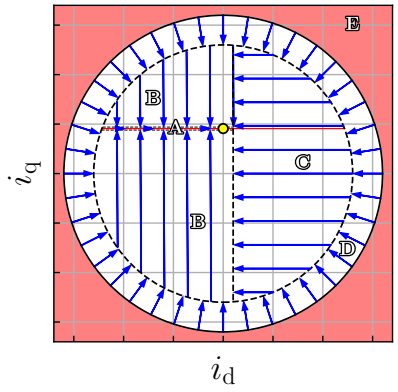
$$\mathcal{E} = \{\boldsymbol{o}[k], \boldsymbol{u}_{\mathrm{dq}}[k], r[k+1], \tau[k+1], \boldsymbol{o}[k+1]\} \quad \text{even if} \quad \boldsymbol{o}[k+1] = \tilde{\boldsymbol{f}}(\boldsymbol{o}[k], \boldsymbol{u}_{\mathrm{dq,S}}[k]). \tag{3.7}$$

If $\boldsymbol{u}_{\mathrm{dq,S}}$ would be included within $\mathcal{E}$ instead, the agent would relate it – although unobjectionable – to the previously discussed suboptimal reward. This would subsequently discourage to select $\boldsymbol{u}_{\mathrm{dq,S}}$, but the future choice of $\boldsymbol{u}_{\mathrm{dq}}$ would likely remain largely unaffected.

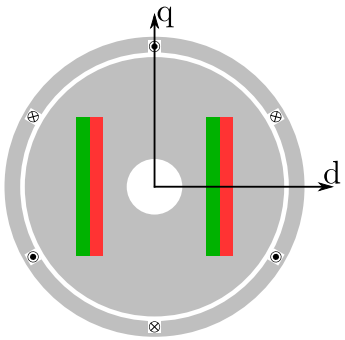
A concise summary of the reward design is provided in Tab. 3.1.



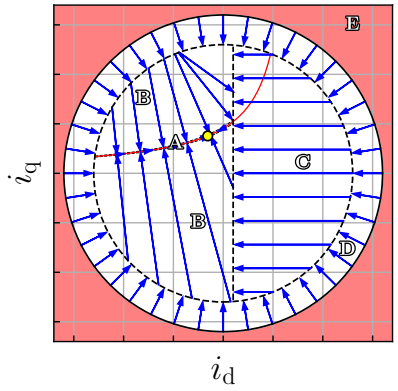
(a) Cross section of a surface PMSM



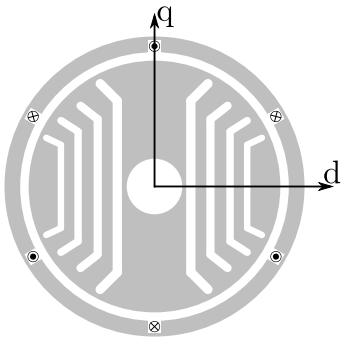
(b) $\nabla_{\boldsymbol{i}_\mathrm{dq}} r$ for a surface PMSM



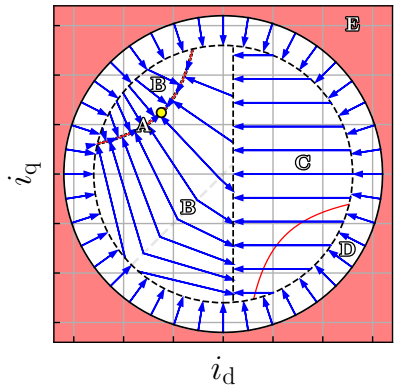
(c) Cross section of an interior PMSM



(d) $\nabla_{\boldsymbol{i}_\mathrm{dq}} r$ for an interior PMSM



(e) Cross section of a SynRM



(f) $\nabla_{\boldsymbol{i}_\mathrm{dq}} r$ for a SynRM

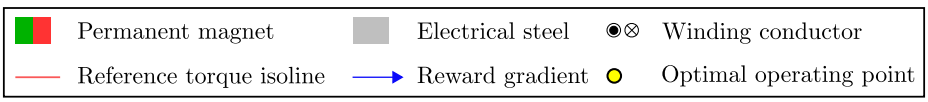


Fig. 3.1: Cross section of different synchronous motor architectures with $p = 1$ and their resulting reward function gradients for an exemplary $T^* > 0$, assuming ideally linear motor behavior

Tab. 3.1: Reward definition for the RL-DTC

	Definition	**Condition**	**Reward** $r[k+1]$
$\mathbb{E}$	Excess current	$i_{\mathrm{lim}} < i_{\mathrm{s}}[k] \rightarrow \tau[k+1] = 1$	-1
$\mathbb{E}_{\mathrm{S}}$	Prevented excess current	$i_{\mathrm{lim}} < \hat{i}_{\mathrm{s}}[k]$	$-(1-\gamma)$
$\mathbb{D}$	Short-time overcurrent	$i_{\mathrm{s}}[k] \in]i_{\mathrm{n}}, i_{\mathrm{lim}}[$	$\left(1 - \frac{i_{\mathrm{s}}[k]-i_{\mathrm{n}}}{i_{\mathrm{lim}}-i_{\mathrm{n}}}\right)\frac{1-\gamma}{2} - (1-\gamma)$
$\mathbb{D}_{\mathrm{S}}$	Prev. short-time overcurrent	$\hat{i}_{\mathrm{s}}[k] \in]i_{\mathrm{n}}, i_{\mathrm{lim}}[$	$-\frac{1-\gamma}{2}$
$\mathbb{C}$	Unfavorable efficiency	$i_{\mathrm{s}}[k] < i_{\mathrm{n}} \wedge i_{\mathrm{d}+} < i_{\mathrm{d}}[k]$	$\left(1 - \frac{i_{\mathrm{d}}[k]-i_{\mathrm{d}+}}{i_{\mathrm{n}}-i_{\mathrm{d}+}}\right)\frac{1-\gamma}{2} - \frac{1-\gamma}{2}$
$\mathbb{C}_{\mathrm{S}}$	Prevented voltage deficiency	$i_{\mathrm{s}}[k] < i_{\mathrm{n}} \wedge i_{\mathrm{d}}[k] < i_{\mathrm{d}+} \wedge \hat{\boldsymbol{u}}_{\mathrm{dq,e}}[k+1] \notin \mathcal{A}$	0
$\mathbb{B}$	Torque tracking	$i_{\mathrm{s}}[k] < i_{\mathrm{n}} \wedge i_{\mathrm{d}}[k] < i_{\mathrm{d}+} \wedge \hat{\boldsymbol{u}}_{\mathrm{dq,e}}[k+1] \in \mathcal{A} \wedge T_{\mathrm{tol}} < \lvert T^*[k] - T[k]\rvert$	$\left(1 - \left\lvert\frac{T^*[k]-T[k]}{2T_{\mathrm{lim}}}\right\rvert\right)\frac{1-\gamma}{2}$
$\mathbb{A}$	Reference torque isoline	$i_{\mathrm{s}}[k] < i_{\mathrm{n}} \wedge i_{\mathrm{d}}[k] < i_{\mathrm{d}+} \wedge \hat{\boldsymbol{u}}_{\mathrm{dq,e}}[k+1] \in \mathcal{A} \wedge \lvert T^*[k] - T[k]\rvert < T_{\mathrm{tol}}$	$\left(1 - \frac{i_{\mathrm{s}}[k]}{i_{\mathrm{lim}}}\right)\frac{1-\gamma}{2} + \frac{1-\gamma}{2}$

3.3 Data-Driven Safeguarding

Although a system model is not necessary to enable control with RL methods, it is initially unsafe and therefore unwanted to train an RL-based controller on real-world systems without employing any safety measures. For its utilization within a safeguarding routine, a data-driven system model is to be identified from the available measurement data, allowing to employ a safeguarding procedure independently of a priori knowledge about the corresponding drive system. The identification is conducted under utilization of the recursive least squares (RLS) algorithm, which is revisited in the following. After that, an overview about the safeguarding procedure is provided on a conceptual level.

3.3.1 Recursive Least Squares

The RLS algorithm is employed to estimate the linear parameters of the PMSM model as of the general difference equation (2.9) that resulted from the physical modeling approach:

$$\boldsymbol{i}_{\mathrm{dq}}[k+1] = \boldsymbol{A}[k]\boldsymbol{i}_{\mathrm{dq}}[k] + \boldsymbol{B}[k]\boldsymbol{u}_{\mathrm{dq}}[k] + \boldsymbol{e}[k]. \tag{3.8}$$

This model is linear with concern to its parameters $\boldsymbol{A}$, $\boldsymbol{B}$ and $\boldsymbol{e}$, i.e.,

$$\boldsymbol{i}_{\mathrm{dq}}[k+1] = \begin{bmatrix}\boldsymbol{A}[k] & \boldsymbol{B}[k] & \boldsymbol{e}[k]\end{bmatrix} \begin{bmatrix}\boldsymbol{i}_{\mathrm{dq}}[k] \\ \boldsymbol{u}_{\mathrm{dq}}[k] \\ 1\end{bmatrix}. \tag{3.9}$$

Exploiting this linear form, the RLS approach is applied to estimate these parameters in an online fashion [106]. In this context, the term 'data-driven' refers to both, the determination of estimations $\hat{\boldsymbol{A}}, \hat{\boldsymbol{B}}, \hat{\boldsymbol{e}}$ from acquired measurements, as well as the characteristic that only numerical dependencies are identified with this method. Determination of physical parameters, such as inductance matrix $\boldsymbol{L}_{\mathrm{dq}}$, stator resistance R_{s} or permanent magnet flux linkage ψ_{p}, would require more elaborate estimation effort, but could succeed on the basis of the acquired estimations $\hat{\boldsymbol{A}}, \hat{\boldsymbol{B}}, \hat{\boldsymbol{e}}$. Since predictability of the motor current is sufficient for the targeted safeguarding procedure, the lack of interpretability of these numerical identification results is not critical and further investigation with respect to a physical system description is not pragmatic in the scope of this work.

In the considered default form, no further auxiliary conditions are imposed upon the parameters, meaning that no interdependence of the entries $\boldsymbol{A}, \boldsymbol{B}$ and $\boldsymbol{e}$ is assumed[5]. The RLS algorithm can be concisely defined via

[5]Note that such interdependence of $\boldsymbol{A}, \boldsymbol{B}$ and $\boldsymbol{e}$ has initially been derived by first principle as of Sec. 2.1. I.e., the identified model has more independent parameters than the derived ODE system.

$$\begin{aligned}
\boldsymbol{\eta}[k-1] &= \frac{\boldsymbol{P}[k-1]\boldsymbol{\xi}[k-1]}{\lambda + \boldsymbol{\xi}^\top[k-1]\boldsymbol{P}[k-1]\boldsymbol{\xi}[k-1]},\\
\hat{\boldsymbol{\chi}}_\mathrm{d}[k] &= \hat{\boldsymbol{\chi}}_\mathrm{d}[k-1] + \boldsymbol{\eta}[k-1]\left(i_\mathrm{d}[k] - \boldsymbol{\xi}^\top[k-1]\hat{\boldsymbol{\chi}}_\mathrm{d}[k-1]\right),\\
\hat{\boldsymbol{\chi}}_\mathrm{q}[k] &= \hat{\boldsymbol{\chi}}_\mathrm{q}[k-1] + \boldsymbol{\eta}[k-1]\left(i_\mathrm{q}[k] - \boldsymbol{\xi}^\top[k-1]\hat{\boldsymbol{\chi}}_\mathrm{q}[k-1]\right),\\
\boldsymbol{P}[k] &= \frac{1}{\lambda}\left(\boldsymbol{I} - \boldsymbol{\eta}[k-1]\boldsymbol{\xi}^\top[k-1]\right)\boldsymbol{P}[k-1],
\end{aligned} \tag{3.10}$$

with regressor vector

$$\boldsymbol{\xi}[k-1] = \begin{bmatrix}\boldsymbol{i}_\mathrm{dq}^\top[k-1] & \boldsymbol{u}_\mathrm{dq}^\top[k-1] & 1\end{bmatrix}^\top, \tag{3.11}$$

and parameter vectors

$$\begin{aligned}
\hat{\boldsymbol{\chi}}_\mathrm{d}[k] &= \begin{bmatrix}\hat{\chi}_\mathrm{d1}[k] & \hat{\chi}_\mathrm{d2}[k] & \hat{\chi}_\mathrm{d3}[k] & \hat{\chi}_\mathrm{d4}[k] & \hat{\chi}_\mathrm{d5}[k]\end{bmatrix}^\top,\\
\hat{\boldsymbol{\chi}}_\mathrm{q}[k] &= \begin{bmatrix}\hat{\chi}_\mathrm{q1}[k] & \hat{\chi}_\mathrm{q2}[k] & \hat{\chi}_\mathrm{q3}[k] & \hat{\chi}_\mathrm{q4}[k] & \hat{\chi}_\mathrm{q5}[k]\end{bmatrix}^\top,
\end{aligned} \tag{3.12}$$

such that

$$\hat{\boldsymbol{A}}[k] = \begin{bmatrix}\hat{\chi}_\mathrm{d1}[k] & \hat{\chi}_\mathrm{d2}[k]\\ \hat{\chi}_\mathrm{q1}[k] & \hat{\chi}_\mathrm{q2}[k]\end{bmatrix}, \qquad \hat{\boldsymbol{B}}[k] = \begin{bmatrix}\hat{\chi}_\mathrm{d3}[k] & \hat{\chi}_\mathrm{d4}[k]\\ \hat{\chi}_\mathrm{q3}[k] & \hat{\chi}_\mathrm{q4}[k]\end{bmatrix}, \qquad \hat{\boldsymbol{e}}[k] = \begin{bmatrix}\hat{\chi}_\mathrm{d5}[k]\\ \hat{\chi}_\mathrm{q5}[k]\end{bmatrix}. \tag{3.13}$$

Herein, λ takes the role of a forgetting factor $\lambda \in]0, 1]$, wherein $\lambda = 1$ denotes no forgetting. The usual choice for λ lies within the range $\lambda \in [0.9, 1[$. In this work, forgetting factors were heuristically selected to $\lambda = 0.9999$ for the FCS case ($T_\mathrm{s} = 50\,\mu\mathrm{s}$) and $\lambda = 0.999$ for the CCS case ($T_\mathrm{s} = 100\,\mu\mathrm{s}$). These values were chosen to compromise between adequate noise suppression and adaptivity of the parameter estimates to changing motor speed.

The matrix $\boldsymbol{P}$ is proportional to the covariance of the parameter estimates $\boldsymbol{P} \sim \mathrm{Cov}(\hat{\boldsymbol{\chi}}, \hat{\boldsymbol{\chi}})$. An initial guess $\boldsymbol{P}[0]$, $\hat{\boldsymbol{\chi}}_\mathrm{d}[0]$ and $\hat{\boldsymbol{\chi}}_\mathrm{q}[0]$ must be set in order to start this algorithm. While this allows to incorporate potentially available knowledge about the system to be identified, the algorithm is usually rather robust concerning uninformed initialization and, therefore, absence of such a priori knowledge is rarely a problem.

3.3.2 Safeguarding Concept

Assuming an identified linear model $\hat{\boldsymbol{A}}, \hat{\boldsymbol{B}}, \hat{\boldsymbol{e}}$ of sufficient precision, it is possible to predict both, the upcoming motor current $\boldsymbol{i}_\mathrm{dq}[k+1]$ as well as the corresponding equilibrium voltage $\boldsymbol{u}_\mathrm{dq,e}[k+1]$. Herein, the latter prediction addresses the concept of recursive feasibility: whenever $\boldsymbol{u}_\mathrm{dq,e}$ cannot be provided by means of the applicable voltage $\boldsymbol{u}_\mathrm{dq}$, the temporal development of $\boldsymbol{i}_\mathrm{dq}$ would be subjected to the internal dynamics of the PMSM, and cannot be dictated by the controller. Hence, RL-DTC-selected actions that violate either of these safety conditions are to be overruled by the safeguard and under consideration of safe alternative actions $\boldsymbol{u}_\mathrm{dq,S}$. As implementation of these measures differs

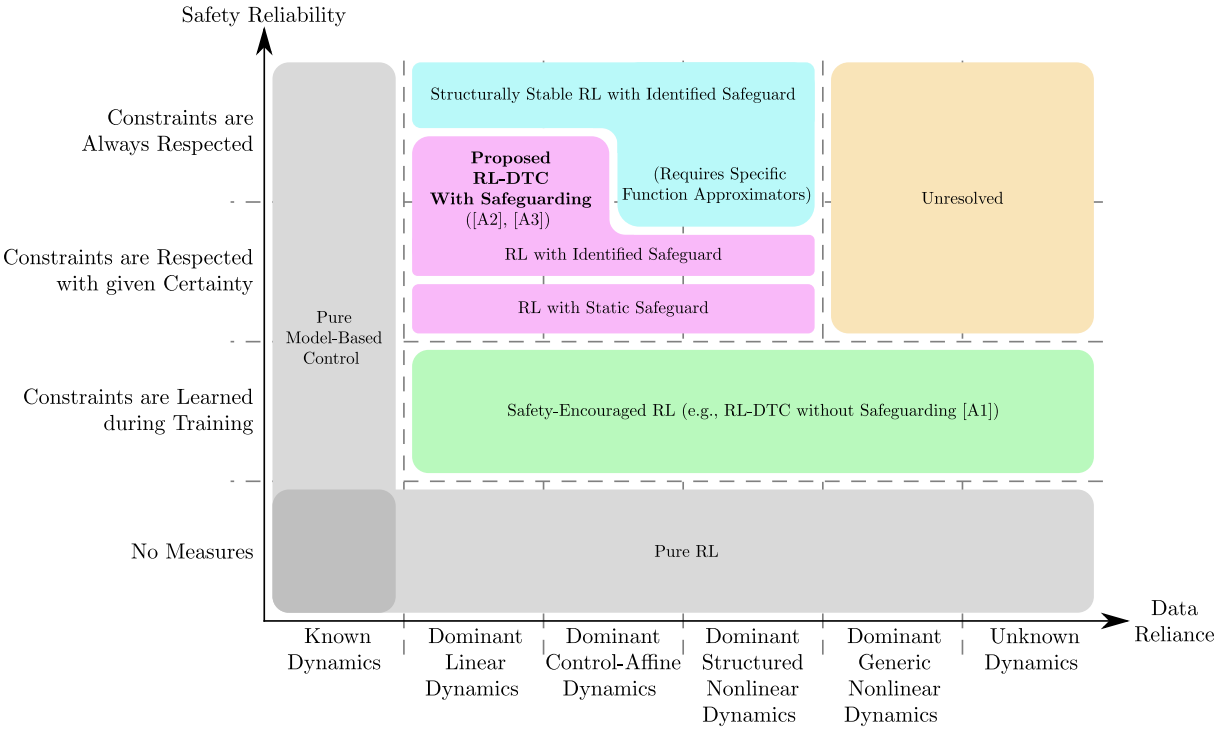


Fig. 3.2: Overview of available safeguarding mechanisms and classification of the proposed RL-DTC (derived from [91])

with concern to their complexity depending on whether an FCS or a CCS is employed, implementation details will be discussed in sections Ch. 4 and Ch. 5, respectively.

In terms of safeguarding research, the proposed approach is closely related to post-posed shielding [107], whose main characteristic is the correcting intervention after the RL agent's decision was made. However, instead of attempting to learn about the necessity of intervention from seen constraint violations (which would require such violations to take place in sufficient quantity), the concept of recursive feasibility [108] as mentioned above is utilized for predictive safety certification [109]. Assuming that the identified affine-linear system dynamics (3.9) are sufficiently accurate, it ultimately becomes obsolete to provoke safety violations for the purpose of collecting data.

According to the safeguard classes that have been introduced in [91], the targeted RL-DTC can be classified as an application with an identified safeguard. As depicted in Fig. 3.2, corresponding scenarios in the context of dominantly linear systems promise for a high level of safety, depending particularly on the identification quality and its underlying system-linearity assumption.

Notably, the identified model is only utilized to avoid harmful operation. It is not available to the RL controller, and the safeguarding measures are to be designed such that interference does only take place if a violation of limitations is foreseeable.

3.4 Experimental Setup

The hardware and software setup are to be described in more detail in the following. The experimental setup is largely independent of the employed control set (FCS or CCS) and only their common parts are explained in this section.

3.4.1 Hardware Architecture

The employed test bench system is depicted in Fig. 3.3. Its main components are the PMSM drive which is controlled by means of the proposed RL-based DTC, serving as device under test (DUT), and a load PMSM drive in speed control mode, allowing training and testing of the control approach in the whole operating range of interest. A schematic of the test bench setup is depicted in Fig. 3.4 with the components being specified in Tab. 3.2. As can be seen, the DUT and the load drive are supplied by distinct DC links at different voltage levels. The load drive is operated via a commercial inverter system that integrates rectifier, inverter, controller and chopper resistance[6] in one casing. It is interfaced with a speed reference signal ω_{me}^{*} and employs standard FOC.

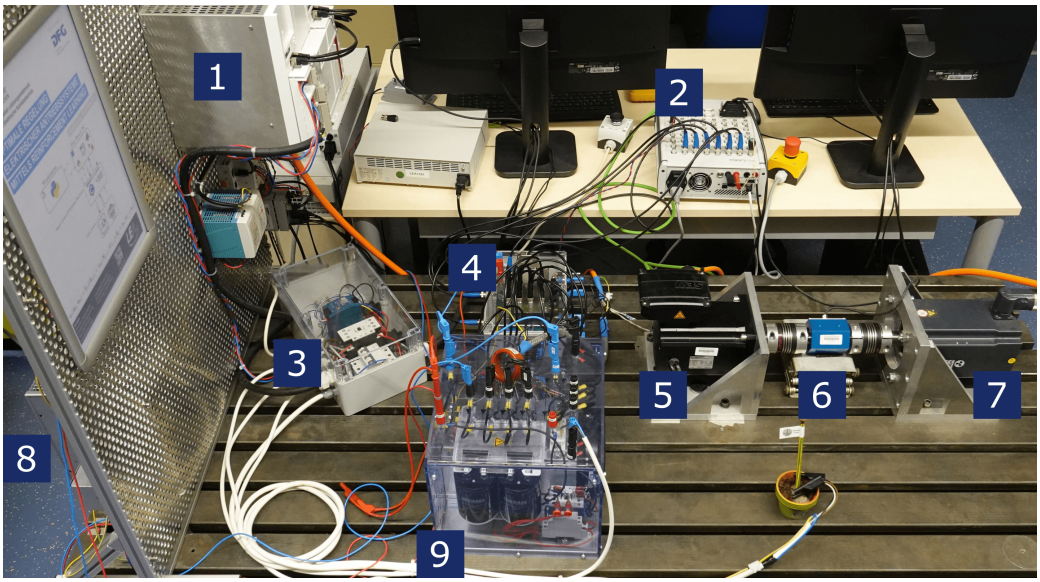


Fig. 3.3: Experimental test bench setup; 1) integrated load inverter, 2) RCPH, 3) protective switch and auxiliary power supply, 4) electric sensors, 5) DUT, 6) drive train and torque sensor (with temporarily removed safety cover), 7) load motor, 8) DC-link chopper resistor, 9) DUT inverter

The entire test bench system is operated via a single rapid-control-prototyping hardware (RCPH) that directly interfaces the driving stage of the DUT inverter, and sends the speed

[6]Standard rectifiers for drive applications are unidirectional, i.e., incapable of supplying back to the grid. Instead, chopper resistors are utilized to dissipate excess energy on the DC link in order to limit the DC-link voltage u_{DC}.

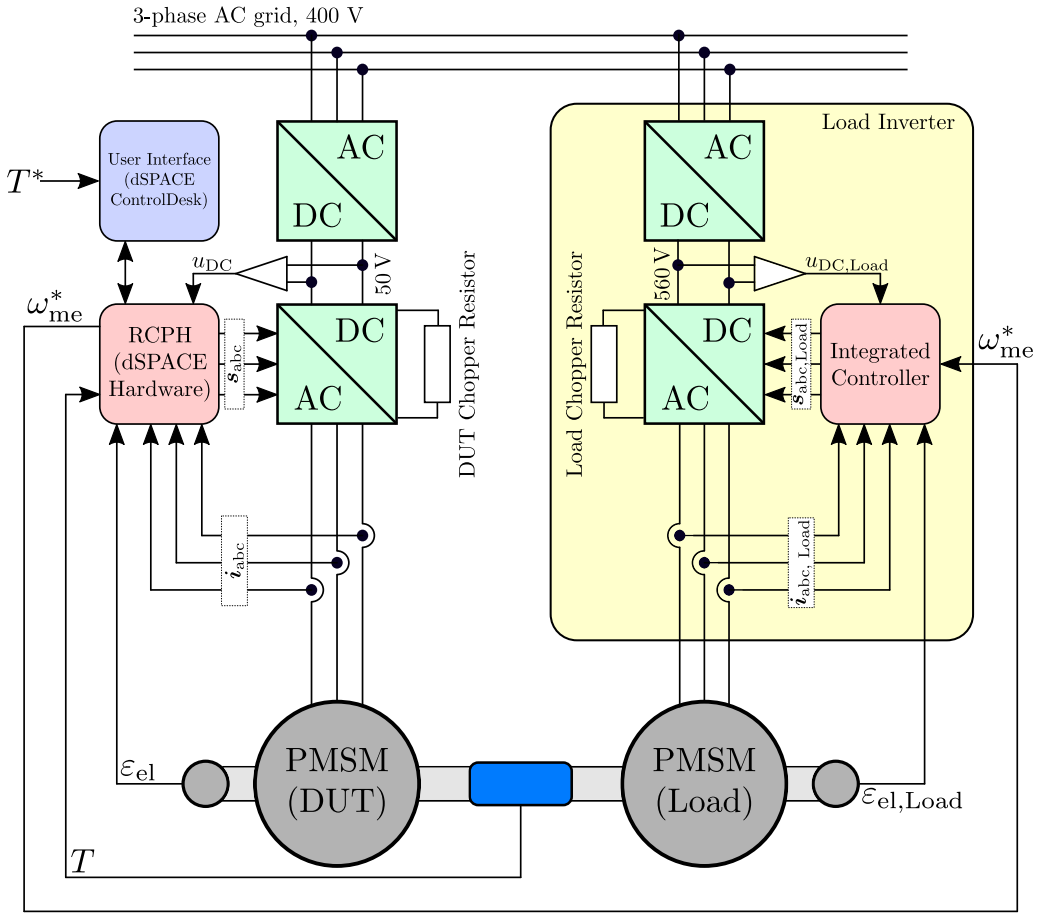


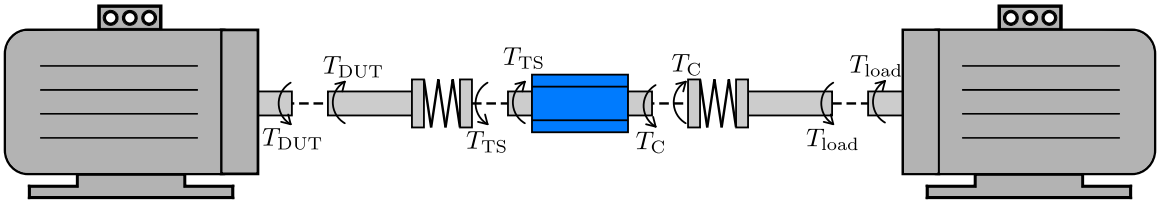
Fig. 3.4: Schematic depiction of the test bench setup

reference ω^*_{me} to the load inverter. Beyond being operated on the same computational hardware, the speed reference generation and the RL-DTC are distinct software systems that are independent of each other, i.e., ω^*_{me} is unknown to the RL-DTC.

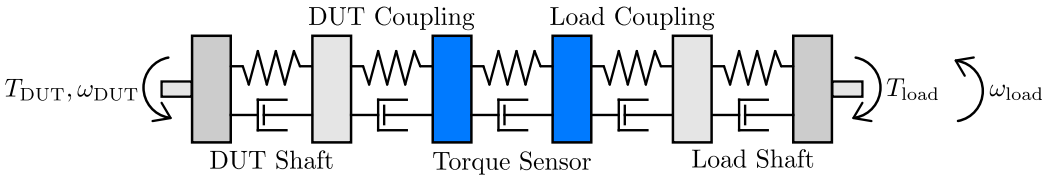
The drive train is schematically depicted in more detail in Fig. 3.5. Herein, it can be seen that the mechanical part of the drive setup is highly oscillatable (which is also observable in the later-presented measurements, cf. Fig. 4.8a). As the mechanical system is expected to introduce a delay in between electromagnetic torque generation and mechanical torque measurement, the observation vector $\boldsymbol{o}$ has been assembled with the use of past command voltages to be capable of compensating for this measurement delay (cf. (2.24), (2.33)). In fact, feasible torque tracking behavior has not been trainable with less than the employed past three voltage features $\boldsymbol{u}_{dq}[k-1, k-2, k-3]$. From these, a delay of two steps is accounted to the mechanical behavior, whereas another one-step delay can be accounted to the digital control execution, which is further discussed in Sec. 3.4.2. The test bench is equipped with a torque sensor (cf. Fig. 3.3, Fig. 3.5), which is necessary for evaluating the reward function (3.5) during the training phase of the RL-DTC. Further, the electromagnetic torque has been experimentally identified for the DUT according to (2.13) [3].

Tab. 3.2: Components of the test bench system

Device	Type	Manufacturer
RCPH	MicroLabBox (built-in FPGA)	dSPACE
FPGA	Kintex-7 XC7K325T FPGA	Xilinx
DUT	CM3C80S	SEW Eurodrive
DUT inverter	SEMITEACH B6U+E1CI	Semikron
Load motor	AM8062-0RHA-0000	Beckhoff Automation
Load inverter	AX5125-0000-0202	Beckhoff Automation
Torque sensor	T8-20NM	interfaceforce



(a) Free body diagram of the drive train



(b) Equivalent scheme of the drive train

Fig. 3.5: Schematic depiction of the mechanical drive train

To distinguish it from the torque measurement T, the electromagnetic torque estimation is referred by the symbol $\hat{T}_{\mathrm{EM}}$ in the following. It is entirely unavailable to the RL-DTC but poses as useful validation quantity as it is largely unaffected by mechanical dynamics of the drive train or signal delay that originates from limited sensor bandwidth.

3.4.2 Software Architecture

As a special characteristic, the RL-DTC training routine is separated from the controller execution algorithmically and by hardware. As the training routine is realized with the use of Python and without any necessity of real-time capability, it is outsourced to a standard workstation computer. The control routine, however, is bound to the defined sampling time T_{s}, which therefore defines an upper boundary for the allowed time demand

Tab. 3.3: Nominal parameterization of the considered drive system SEW CM3C80S (derived from nameplate data) and reward configuration of the RL-DTC with definition of the permitted operation regions

Symbol	Description	Value
p	Pole-pair number	4
R_s	Stator resistance	203 mΩ
L_d	d inductance	1.44 mH
L_q	q inductance	1.44 mH
ψ_p	Permanent magnet flux linkage	112 mVs
U_DC	Configured DC-link voltage	50 V
T_s	Sampling time (FCS/CCS)	50 / 100 µs
i_n	Nominal current	13 A
i_lim	Maximum current	16 A
$i_\mathrm{d+}$	Tolerable positive d current	4 A
$u_\mathrm{DC,min}$	Minimum DC-link voltage	25 V
$u_\mathrm{DC,max}$	Maximum DC-link voltage	75 V
$n_\mathrm{me,lim}$	Maximum speed	750 min^{-1}
T_lim	Maximum torque	10.5 N · m
T_tol	Torque control tolerance	0.1 N · m

of controller execution within each sampling period. Since the turnaround time demand of the control routine varies depending on ANN size, and the turnaround time demand of the training routine varies with the employed RL algorithm, control set and also ANN size, corresponding evaluation is not an architectural trait of the herein proposed toolchain, but rather of its parameterization. Hence, computational time demand will be assessed in the upcoming chapters that are concerned with application to the FCS and CCS scenario (cf. Sec. 4.2 and Sec. 5.2).

Note that, within this contribution, different sampling times are defined in dependency of the control set, with $T_\mathrm{s} = 50\,\mu\mathrm{s}$ for the FCS, and $T_\mathrm{s} = 100\,\mu\mathrm{s}$ for the CCS case. While shorter sampling time is preferable for the minimization of current and torque ripple [73], the computational demand of the later-proposed CCS safeguarding procedure renders it necessary to provide more time for calculations (cf. Ch. 5). On the other hand, the utilization of regular sampling in conjunction with a modulator delimit the impact of the current ripple upon the controller (cf. Sec. 2.2.2).

A schematic overview of the distributed RL pipeline is provided by Fig. 3.6. Herein, it can be seen that the control routine running on the RCPH subjects to real-time conditions, while the training of the RL-DTC is operated asynchronously to the real-time process

on the workstation computer. The communication between test bench PC and workstation is realized via TCP/IP, because this protocol ensures causal integrity of transmitted packages and, hence, no further effort has to be put into package sorting[7]. Otherwise, confusion about the origin and destination of state transitions could arise when the workstation receives their data out of order. This would disrupt the controller training process because the drive dynamics would be represented incorrectly.

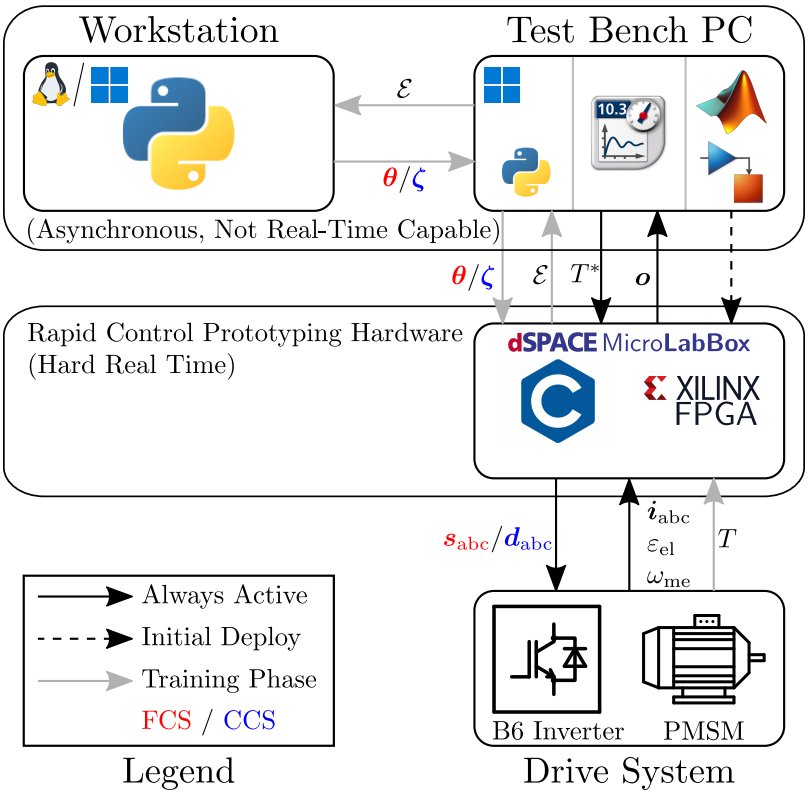


Fig. 3.6: Schematic of the edge RL pipeline

3.4.2.1 FPGA Utilization for Fast ANN Inference

Due to the high sampling frequency, the real-time evaluation of the employed neural networks of $\hat{q}_{\boldsymbol{\theta}}$ in the FCS case and $\boldsymbol{\pi}_{\boldsymbol{\zeta}}$ in the CCS case pose a challenge for purely CPU based execution on the RCPH. As MLPs are structurally well-suited for parallelization, implementation of the employed MLP on the integrated FPGA offers sufficient computational acceleration to render real-time execution feasible [110, 111]. Herein, the calculation of each matrix-vector product is partitioned into the parallel computation of vector-vector scalar products:

[7]The UDP protocol does not inherently ensure causal integrity, i.e., data packages may be received out of order. In highly local networks, however, this risk is negligible and the simpler protocol may reduce latency, rendering UDP a preferable alternative under corresponding circumstances [C19].

$$\boldsymbol{y}_j = f_{\mathrm{act},j}(\boldsymbol{K}_j \boldsymbol{y}_{j-1} + \boldsymbol{b}_j) = f_{\mathrm{act},j}\left(\underbrace{\begin{bmatrix} \boldsymbol{k}_{j,1}^\top & b_{j,1} \\ \boldsymbol{k}_{j,2}^\top & b_{j,2} \\ \vdots & \vdots \end{bmatrix}}_{\tilde{\boldsymbol{K}}_j} \tilde{\boldsymbol{y}}_{j-1} \right) = \begin{bmatrix} f_{\mathrm{act},j}(\tilde{\boldsymbol{k}}_{j,1}^\top \tilde{\boldsymbol{y}}_{j-1}) \\ f_{\mathrm{act},j}(\tilde{\boldsymbol{k}}_{j,2}^\top \tilde{\boldsymbol{y}}_{j-1}) \\ \vdots \end{bmatrix}, \tag{3.14}$$

wherein

$$\tilde{\boldsymbol{K}} = \begin{bmatrix} \boldsymbol{K} & \boldsymbol{b} \end{bmatrix}, \qquad \tilde{\boldsymbol{k}}^\top = \begin{bmatrix} \boldsymbol{k}^\top & b \end{bmatrix}, \qquad \tilde{\boldsymbol{y}} = \begin{bmatrix} \boldsymbol{y}^\top & 1 \end{bmatrix}^\top,$$

with j being the index of an arbitrary layer. Whereas a CPU would need to compute the components of the layer output $\boldsymbol{y}_j$ in a serial fashion, parallel implementation as of (3.14) provides that all entries of $\boldsymbol{y}_j$ are available at the same time. The structure of this architecture is depicted in Fig. 3.7.

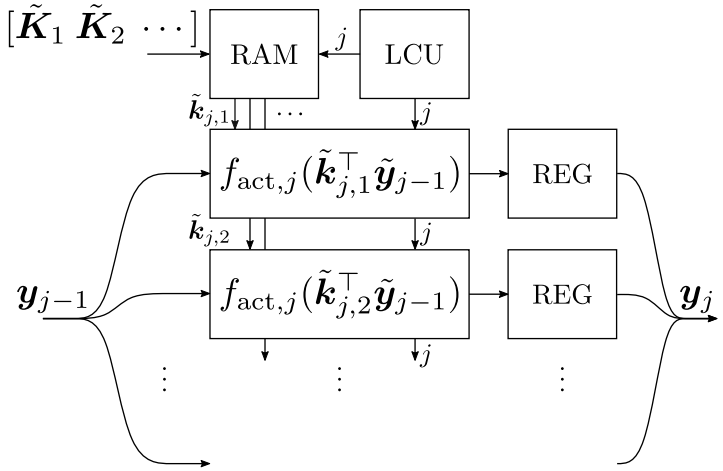


Fig. 3.7: Structural depiction of the parallelization of ANN execution within the FPGA; RAM: random-access memory, LCU: layer control unit, REG: register, $\tilde{\boldsymbol{K}}_j$: ANN parameters of layer j (FCS: $\tilde{\boldsymbol{K}} \in \boldsymbol{\theta}$, CCS: $\tilde{\boldsymbol{K}} \in \boldsymbol{\zeta}$); $\boldsymbol{y}_j$: output of layer j; inspired from [110]

3.4.2.2 Digital Control Delay

Because of the digital controller implementation, the input voltage $\boldsymbol{u}_{\mathrm{dq}}[k]$ that reacts upon a measurement $\boldsymbol{i}_{\mathrm{dq}}[k]$ will not be available at the same time instant as the measurement is available. Due to the demand in computation time, $\boldsymbol{u}_{\mathrm{dq}}[k]$ will be applied one sampling period after $\boldsymbol{i}_{\mathrm{dq}}[k]$ has been measured, leading to an inherent delay of one control cycle [112], which is conceptually depicted in Fig. 3.8.

This behavior cannot be avoided, but it must be handled as it plays a critical role for the later-presented safeguarding routines: the safeguard must consider the plant state at the instant when the commanded control signal becomes active. Hence, the selection of $\boldsymbol{u}_{\mathrm{dq}}[k]$ would need to happen under consideration of $\boldsymbol{i}_{\mathrm{dq}}[k+1]$ (cf. Fig. 3.8), which is yet

unavailable at time step k. According to the identified system behavior (cf. Sec. 3.3), however, it is straightforward to predict

$$\hat{\boldsymbol{i}}_{\mathrm{dq}}[k+1] = \begin{bmatrix} \hat{\boldsymbol{A}}[k] & \hat{\boldsymbol{B}}[k] & \hat{\boldsymbol{e}}[k] \end{bmatrix} \begin{bmatrix} \boldsymbol{i}_{\mathrm{dq}}[k] \\ \boldsymbol{u}_{\mathrm{dq}}[k-1] \\ 1 \end{bmatrix}, \tag{3.15}$$

wherein $\hat{\boldsymbol{i}}_{\mathrm{dq}}[k+1]$ is the estimated upcoming motor current. Hence, the digital control delay is accounted for, and the motor state that will be active at the instant when the safeguard is capable of intervention is approximated by means of the identified RLS predictor. Control delay compensation by means of this scheme is assumed for all following investigations. Therefore, the control delay is omitted in subsequent notation for ease of reading.

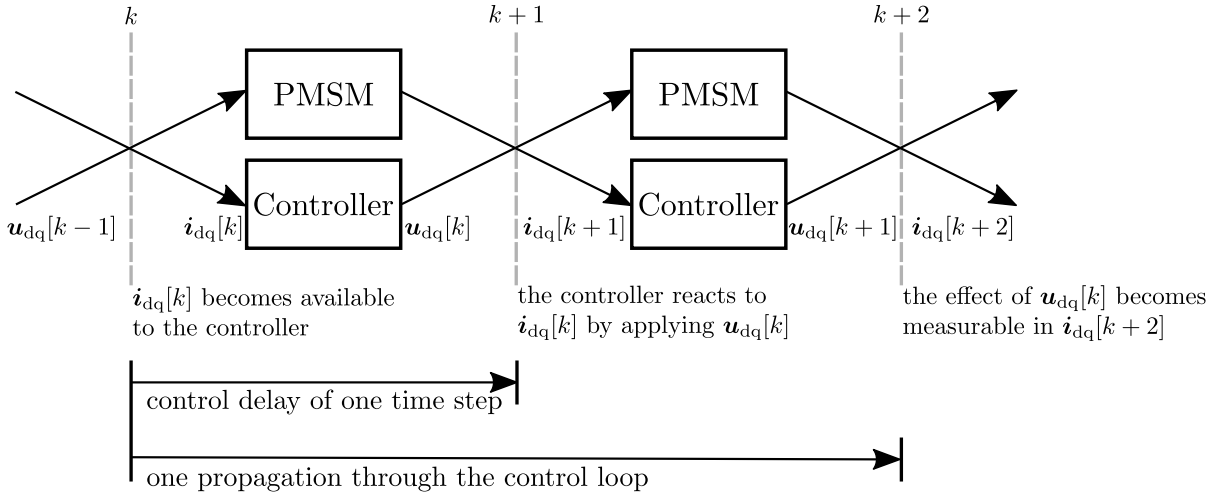


Fig. 3.8: Chronology of the digital control loop

4 Reinforcement Learning-Based Direct Torque Control on the Finite Control Set

This chapter covers the application of the aforementioned considerations to the PMSM drive setup with FCS interface, which has originally been published in [A1] and [A2]. Therefore, the sampling frequency is set to 20 kHz ($T_\mathrm{s} = 50\,\mu\mathrm{s}$), and the DQN algorithm (cf. Sec. A.2) is applied to solve the FCS-DTC problem, which is encoded by means of the previously discussed reward function (cf. Sec. 3.2). A general scheme of the controller setup is depicted in Fig. 4.1. Before experimental results are reviewed, it is discussed how the safeguarding considerations are applied in FCS mode.

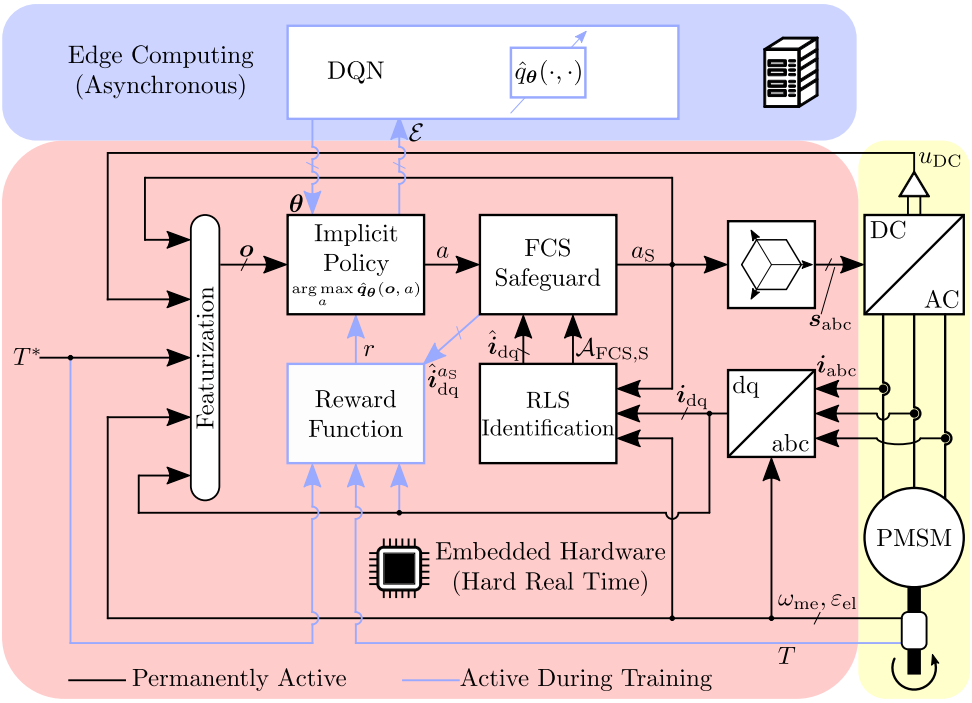


Fig. 4.1: Schematic of the FCS-RL-DTC with safeguarding

4.1 Finite-Control-Set Safeguarding

As already mentioned in Sec. 2.4.1, the critical operation limitations for the motor are specified by means of the nominal current boundary $i_\mathrm{s} \leq i_\mathrm{n}$, which is the most important safety condition, and the voltage boundary $\boldsymbol{u}_\mathrm{dq,e} \in \mathcal{A}_\mathrm{FCS}$, which prevents unintended changes of $\boldsymbol{i}_\mathrm{dq}$ and, hence, ensures safety indirectly. Utilizing the data-driven plant model identified by the RLS, the linear predictor (3.9) can be used to check adherence to both of these boundaries for the upcoming state.

Concerning the current boundary, this is performed by verifying that the predicted current $\hat{\boldsymbol{i}}_\mathrm{dq}[k+1]$ adheres to the specified nominal current:

$$\begin{aligned} & \|\hat{\boldsymbol{i}}_\mathrm{dq}[k+1]\|_2 \leq i_\mathrm{n} \quad \forall \boldsymbol{u}_\mathrm{dq,S}[k] \in \mathcal{A}_\mathrm{FCS,S} \\ \Leftrightarrow \quad & \|\hat{\boldsymbol{A}}[k]\boldsymbol{i}_\mathrm{dq}[k] + \hat{\boldsymbol{B}}[k]\boldsymbol{u}_\mathrm{dq,S}[k] + \hat{\boldsymbol{e}}[k]\|_2 \leq i_\mathrm{n} \quad \forall \boldsymbol{u}_\mathrm{dq,S}[k] \in \mathcal{A}_\mathrm{FCS,S}, \end{aligned} \tag{4.1}$$

wherein the safe action $\boldsymbol{u}_\mathrm{dq,S}$ denotes an element from the safe action space $\mathcal{A}_\mathrm{FCS,S}$. To also ensure compliance with the voltage boundary (2.21b), it must be examined whether the predicted operating point $\hat{\boldsymbol{i}}_\mathrm{dq}[k+1]$ could be sustained – i.e., whether $\hat{\boldsymbol{i}}_\mathrm{dq}[k+2] = \hat{\boldsymbol{i}}_\mathrm{dq}[k+1]$ is possible – with the available DC-link voltage u_DC. However, it is not necessary and, due to the nature of the FCS, also highly unlikely that the corresponding equilibrium voltage $\boldsymbol{u}_\mathrm{dq,e}$ can be matched exactly by means of one of the voltage vectors within $\mathcal{A}_\mathrm{FCS}$. Instead, it is to be verified whether the available fundamental voltage magnitude[1] of the VSI at $\frac{2}{\pi}u_\mathrm{DC}$ suffices to provide the magnitude of the equilibrium voltage $\|\boldsymbol{u}_\mathrm{dq,e}\|_2$:

$$\begin{aligned} & \|\hat{\boldsymbol{u}}_\mathrm{dq,e}[k+1]\|_2 \leq \frac{2}{\pi}\hat{u}_\mathrm{DC}[k+1] \\ \Leftrightarrow \quad & \|\hat{\boldsymbol{B}}^{-1}[k]\left((\boldsymbol{I} - \hat{\boldsymbol{A}}[k])\hat{\boldsymbol{i}}_\mathrm{dq}[k+1] - \hat{\boldsymbol{e}}[k]\right)\|_2 \leq \frac{2}{\pi}u_\mathrm{DC}[k] \\ \Leftrightarrow \quad & \|\hat{\boldsymbol{B}}^{-1}[k]\left((\boldsymbol{I} - \hat{\boldsymbol{A}}[k])\left(\hat{\boldsymbol{A}}[k]\boldsymbol{i}_\mathrm{dq}[k] + \hat{\boldsymbol{B}}[k]\boldsymbol{u}_\mathrm{dq,S}[k]\right) - \hat{\boldsymbol{A}}[k]\hat{\boldsymbol{e}}[k]\right)\|_2 \leq \frac{2}{\pi}u_\mathrm{DC}[k], \end{aligned} \tag{4.2}$$

for all

$$\boldsymbol{u}_\mathrm{dq,S}[k] \in \mathcal{A}_\mathrm{FCS,S}. \tag{4.3}$$

Herein, it is assumed that u_DC will not change drastically from one sampling instant to the next [113], i.e., $\hat{u}_\mathrm{DC}[k+1] = u_\mathrm{DC}[k]$. Naturally, inspection of these conditions must succeed for all switching states a, i.e., all $\boldsymbol{u}_\mathrm{dq} \in \mathcal{A}_\mathrm{FCS}$ that can lead to different follow-up states $\hat{\boldsymbol{i}}_\mathrm{dq}[k+1]$. Each switching state that violates any of the constraints may not be selected in the momentary time step, neither in case of an exploiting nor in case of an exploring action. Under these circumstances, the safeguard overwrites the agent's policy entirely to ensure safe plant operation.

Abbreviating the set of applicable voltages that satisfy (4.1) with $\mathcal{C}_i$ such that

[1] Mathematical and graphical derivation of this value is provided in Sec. A.1.

$$\mathcal{C}_i[k] = \left\{ \boldsymbol{u}_{\mathrm{dq}} \in \mathbb{R}^2 \middle| \, \|\hat{\boldsymbol{i}}_{\mathrm{dq}}[k+1]\|_2 \leq i_{\mathrm{n}} \right\}, \tag{4.4}$$

and the set that satisfies (4.2) with $\mathcal{C}_u$ such that

$$\mathcal{C}_u[k] = \left\{ \boldsymbol{u}_{\mathrm{dq}} \in \mathbb{R}^2 \middle| \, \|\hat{\boldsymbol{u}}_{\mathrm{dq,e}}[k+1]\|_2 \leq \frac{2}{\pi} u_{\mathrm{DC}}[k] \right\}, \tag{4.5}$$

the approximate set of safe actions $\mathcal{A}_{\mathrm{FCS,S}} = \mathcal{A}_{\mathrm{FCS}} \cap \mathcal{C}_i \cap \mathcal{C}_u$ can be graphically constructed as depicted in Fig. 4.2.

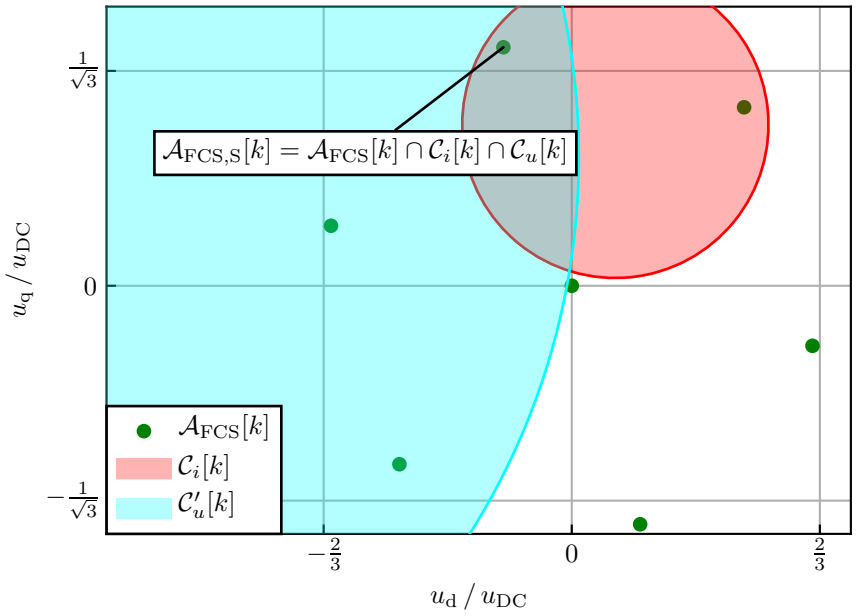


Fig. 4.2: Exemplary graphical construction of the safe finite action set $\mathcal{A}_{\mathrm{FCS,S}}$ in the voltage plane

Taking deployment conditions into account, the presence of measurement noise or sub-optimal RLS identification might lead to an empty set of safe actions $\mathcal{A}_{\mathrm{FCS,S}} = \emptyset$. As a fallback solution, such cases are handled by selecting

$$\boldsymbol{u}_{\mathrm{dq,S}}[k] = \underset{\boldsymbol{u}_{\mathrm{dq}} \in \mathcal{A}_{\mathrm{FCS}}[k]}{\arg\min} \|\hat{\boldsymbol{i}}_{\mathrm{dq}}[k+1]\|_2 \qquad \text{if } \mathcal{A}_{\mathrm{FCS,S}}[k] = \emptyset, \tag{4.6}$$

to define an applicable switching state for reasons of computational integrity. Moreover, it might often happen that the safeguard has to intervene in cases where there is no unique safe action, i.e., $|\mathcal{A}_{\mathrm{FCS,S}}| > 1$. In such situations, the action selection process may incorporate the momentary targeted compromise between exploration and exploitation (cf. Sec. 2.5.1, (2.29), Sec. A.2):

$$\boldsymbol{u}_{\mathrm{dq,S}}[k] = \begin{cases} \underset{\boldsymbol{u}_{\mathrm{dq}} \in \mathcal{A}_{\mathrm{FCS,S}}[k]}{\arg\max} \; \hat{q}_{\boldsymbol{\theta}}(\boldsymbol{o}[k], \boldsymbol{u}_{\mathrm{dq}}) & \text{for exploitation,} \\ \in_{\mathrm{R}} \mathcal{A}_{\mathrm{FCS,S}}[k] & \text{for exploration.} \end{cases} \tag{4.7}$$

Finally, the FCS-RL-DTC agent is able to learn optimized operation behavior by means of measured real-world data, while the consequences of safety-critical actions are predicted using the data-driven model. Actual terminations of plant operation are therefore obsolete for the training success, and the training can be continued without time loss or harm to the system.

4.2 Experimental Results

In order to validate the proposed FCS-RL-DTC approach, experimental verification is performed on the test system specified in Sec. 3.4. The validation of the FCS-RL-DTC is conducted in three steps:

- Firstly, the safeguard is tested concerning its functionality.
- Secondly, the training phase for the RL agent is analyzed in terms of physical and numerical stability.
- Lastly, the performance of the FCS-RL-DTC is presented in a series of exemplary test scenarios while exploration actions are deactivated.

Within all experiments, the control cycle time is synchronized to the sampling time $T_\mathrm{s} = 50\,\mu\mathrm{s}$ (cf. Tab. 3.3), i.e., all required calculations have to be completed within $50\,\mu\mathrm{s}$, whose feasibility is confirmed as per the measurements of the control turnaround time $T_\mathrm{C,TA}$ stated in Tab. 4.1. Further, the turnaround time demand of the training routine $T_\mathrm{T,TA}$ is listed, which is not subjected to real-time conditions. For overall configuration, a hyperparameter optimization (HPO) has been conducted as described in Sec. A.2.1, from which most settings are derived[2].

Tab. 4.1: Statistical evaluation of the FCS-RL-DTC control turnaround time $T_\mathrm{C,TA}$ for the speed ramp experiment depicted in Fig. 4.8b and of the training turnaround time $T_\mathrm{T,TA}$ for an exemplary training procedure.

	μ	σ	$\lVert T_\mathrm{TA} \rVert_\infty$
Control $T_\mathrm{C,TA}$	32.888 µs	0.699 µs	38.760 µs
Training $T_\mathrm{T,TA}$	9.926 ms	7.528 ms	1.285 s

4.2.1 Safeguard Functionality

To ensure safety of components and personnel and, secondarily, to avoid training downtimes it is of priority to validate the functionality of the safeguarding procedure that has been presented in Sec. 4.1. This investigation is performed for two scenarios: at low

[2]The HPO is not included in the main part of this work for its dependency on the specifically employed algorithm (here: DQN), whereas the RL-DTC can generally be paired with any RL algorithm.

speed, where the induced voltage plays no significant role, and secondly at higher speed where the voltage boundary becomes significant. The tests are conducted by presenting arbitrary, random switching actions to the safeguard algorithm, which then should be able to intervene in the case of unsafe actions to subject the motor current trajectory $\boldsymbol{i}_{\mathrm{dq}}$ to the current and voltage limitations.

The resulting long-term current trajectories are depicted in Fig. 4.3 and Fig. 4.4. Please note that the test bench includes a load machine as well as open-loop controlled DC-link choppers, whose dynamics are not included in the identified model Sec. 3.3 and, therefore, cannot be considered by the safeguard. Still, the current boundary has visibly been avoided with no striking violation. Further, the identified voltage boundary is displayed with the identified voltage limit being adhered to rather consistently as of Fig. 4.4.

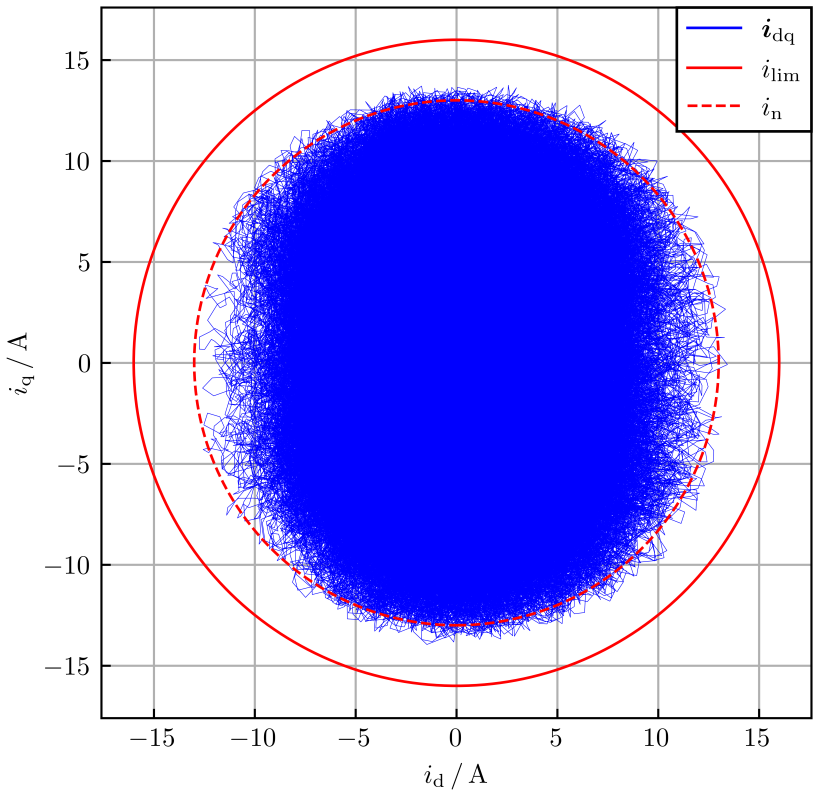


Fig. 4.3: Operation at low speed $|n_{\mathrm{me}}| \leq 50\,\mathrm{min}^{-1}$, only the current boundary is active

A statistical evaluation of the safeguard's prediction uncertainty is delivered in Tab. 4.2, which indicates that the minor violations, which are mainly visible in Fig. 4.4, can be attributed to the prediction uncertainty that remains during usage of the RLS [106]. This result validates the proposed safeguard architecture, which allows certain prediction error in correspondence to the leeway between i_{n} and i_{lim}.

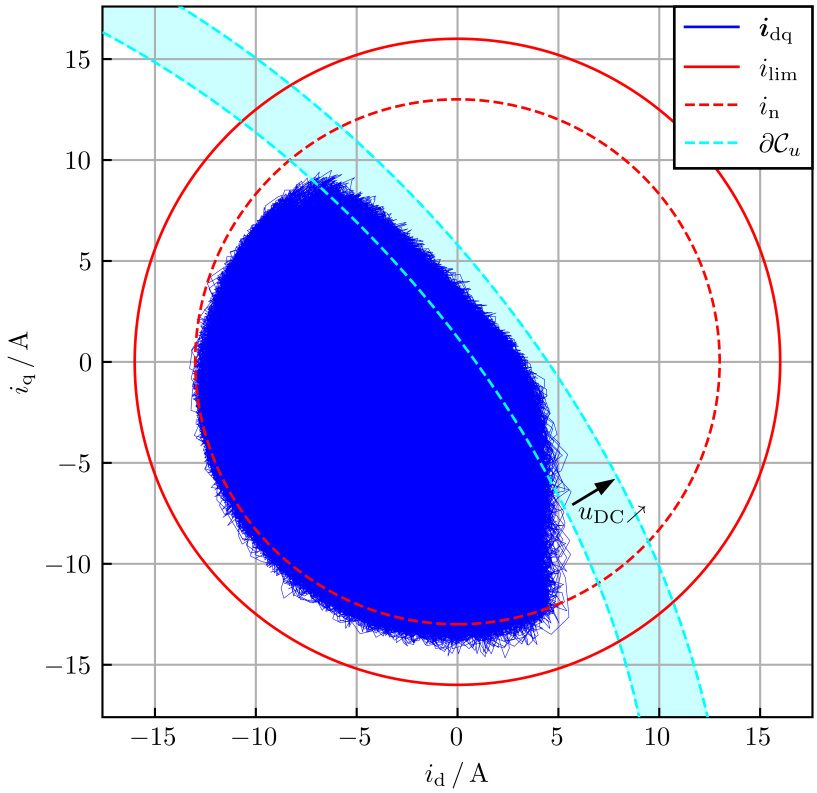


Fig. 4.4: Operation at high speed $n_{\mathrm{me}} = 700\,\mathrm{min}^{-1}$, current and voltage boundaries are active

Overall, the functionality of the safeguard can be confirmed. In fact, no violation of the current limit occurred and, hence, no emergency shutdown was necessary for the entirety of investigations in this work.

4.2.2 Training Phase

After asserting the safeguard functionality, the training phase is investigated in detail. To analyze the numerical convergence characteristic of the training phase, ten separate FCS-RL-DTC agents have been trained independently for ten minutes each, i.e., with re-initialization of a new set of random network weights for each individual agent. Apart from the random network initialization, the FCS-RL-DTC is configured according to the HPO result Sec. A.2.1. During the training, the torque references and the operated motor speed that is enforced by the load machine are resampled uniformly at random sampling instants as specified in Tab. 4.3. Herein, both signals, T^* and n^*_{me} have generally step-like form, with the load motor acceleration capability being limited as of Tab. 4.3. The learning rate β_q and the ϵ-greedy exploration parameter are linearly decreased from the beginning until completion of the training phase, which is referred to as scheduling. The

Tab. 4.2: Statistical evaluation of the one-step absolute current prediction error $|e_{\mathrm{d,q}}| = |\hat{i}_{\mathrm{d,q}} - i_{\mathrm{d,q}}|$ with mean μ, standard deviation σ and worst-case prediction error $\|e\|_\infty$ for the recordings depicted in Fig. 4.3 and Fig. 4.4

n_{me}	μ	σ	$\|e\|_\infty$
$\leq 50\,\mathrm{min}^{-1}$	$\mu_{\lvert e_{\mathrm{d}}\rvert} = 3.788 \cdot 10^{-1}\,\mathrm{A}$	$\sigma_{\lvert e_{\mathrm{d}}\rvert} = 2.862 \cdot 10^{-1}\,\mathrm{A}$	$\|e_{\mathrm{d}}\|_\infty = 2.501\,\mathrm{A}$
	$\mu_{\lvert e_{\mathrm{q}}\rvert} = 4.263 \cdot 10^{-1}\,\mathrm{A}$	$\sigma_{\lvert e_{\mathrm{q}}\rvert} = 3.221 \cdot 10^{-1}\,\mathrm{A}$	$\|e_{\mathrm{q}}\|_\infty = 2.141\,\mathrm{A}$
$700\,\mathrm{min}^{-1}$	$\mu_{\lvert e_{\mathrm{d}}\rvert} = 3.004 \cdot 10^{-1}\,\mathrm{A}$	$\sigma_{\lvert e_{\mathrm{d}}\rvert} = 2.586 \cdot 10^{-1}\,\mathrm{A}$	$\|e_{\mathrm{d}}\|_\infty = 1.454\,\mathrm{A}$
	$\mu_{\lvert e_{\mathrm{q}}\rvert} = 2.732 \cdot 10^{-1}\,\mathrm{A}$	$\sigma_{\lvert e_{\mathrm{q}}\rvert} = 3.016 \cdot 10^{-1}\,\mathrm{A}$	$\|e_{\mathrm{q}}\|_\infty = 1.573\,\mathrm{A}$

initial and final parameters of the corresponding schedules are strongly related to the specific type of algorithm, and are, hence included within the HPO Sec. A.2.

Tab. 4.3: Training configuration of the FCS-RL-DTC

Configuration Parameter	**Specification**
Torque reference range	$[-6.5\,\mathrm{N \cdot m}, 6.5\,\mathrm{N \cdot m}]$
Torque reference change probability	10^{-4}
Speed range	$[-675\,\mathrm{min}^{-1}, 675\,\mathrm{min}^{-1}]$
Speed change probability	$5 \cdot 10^{-6}$
Maximum acceleration	$80\,\frac{\mathrm{min}^{-1}}{\mathrm{s}}$
Training duration	$10\,\mathrm{min}$

Several snapshots from one exemplary training phase are depicted in Fig. 4.5. As can be seen, the reference torque T^* and the speed n_{me} are varying over the training time. While the early performance looks quite insufficient due to the untrained control agent, the performance at the end of the training phase is a lot more satisfying. Please note that the agent follows an explorative policy (cf. Sec. A.2) for the whole training time and, therefore, even the latest depicted interval features suboptimal exploration behavior. Moreover, the rather high motor speed would necessitate operation beyond the voltage boundary, which cannot be realized and always results in tracking errors. Further, it is visible that the safeguard is activated quite often within the early stages of the training whereas the number of interventions strongly decreases close to the training finish. Both, the visibly increasing control accuracy as well as the decreasingly frequent safeguard interventions confirm the feasibility of the FCS-RL-DTC training.

Over the course of each training, the measured reward is recorded and a statistical evaluation of the learning behavior over all ten agents is depicted in Fig. 4.6. As visible, the mean reward of the agent ensemble is converging reliably, and also the corresponding standard deviation $\sigma_{\overline{r}}$ is decreasing. Strikingly, no negative outliers are observed during the procedure and, despite the randomness of ANN initialization and training routine,

the final performance in the training has been measured to be quite comparable to each other. Notable control performance is consistently reached within less than ten minutes of training [114].

4.2.3 Torque-Tracking Behavior

To validate the FCS-RL-DTC in practice, the performance of an exemplary trained agent is assessed in several test bench experiments:

- torque reference step at constant speed
- speed ramp from negative to positive velocity at constant torque reference
- torque reference ramp from negative to positive torque at constant speed
- small-signal investigation with several torque reference steps at constant speed.

For these tests, the best-performing agent from the previous training investigation was selected concerning its cumulative reward during training. Moving-average quantities are determined with a window size of 50 ms and are denoted by an overline ($\overline{T}, \overline{i}$).

4.2.3.1 Torque Reference Step

In the first experiment, a torque reference step is investigated to evaluate the control loop's reaction to transients. The obtained measurement is depicted in Fig. 4.8a. Beside the measured torque T, also the calculated electromagnetic torque estimation $\hat{T}_{\mathrm{EM}}$ is presented, which does not feature the low-pass behavior and the oscillations of the drive train and is, hence, a more feasible basis for investigating the control behavior precisely.

Unfortunately, the given drive train features a dominant oscillatory behavior with limited bandwidth (as can be inferred from the time lag between $\hat{T}_{\mathrm{EM}}$ and T). Since the torque measurement T is utilized within the reward formulation, the parasitic drive train behavior is assumed to limit the reachable torque tracking precision of the control loop. Moreover, the current and torque ripple that is inherent to FCS approaches leads to consistent excitation of the oscillation, such that a true steady state is never reached.

Despite these complications, which are beyond the controller's regime, fast torque tracking of roughly 5 ms can be observed in Fig. 4.8a, and even the time series of applied actions a features the familiar overlapping staircase form (cf. [C10, C14, 115]). Interestingly, a rather large $|i_{\mathrm{d}}|$ was observed during the experiment, whose origin is discussed later in more detail.

4.2.3.2 Speed Ramp

A speed ramp experiment with constant torque reference is depicted in Fig. 4.8b. Over the course of the acceleration, the torque ripple can be seen to be quite significant, which is usual for FCS control schemes (and partly also attributed to the drive train oscillation).

The moving average of the torque measurement $\overline{T}$, however, features clearly that the torque reference is approximately met for the whole considered speed range.

Further, the i_d current can be observed to contain striking harmonics, whose frequency seems related to the ramping speed. A frequency analysis as of the short-time Fourier transform $\mathcal{S}$ of i_d depicted in Fig. 4.7 reveals, that the harmonic components of i_d in fact oscillate at different orders of the fundamental electric frequency $\omega_\mathrm{el} \sim pn_\mathrm{me}$. Despite being unwanted for efficiency maximization, the torque of the given surface-mounted PMSM can be assumed independent of i_d and, hence, corresponding oscillations are irrelevant in the context of torque tracking.

4.2.3.3 Torque Reference Ramp

Fig. 4.8c presents an experiment wherein the FCS-RL-DTC is tracking a ramping reference torque. This scenario reveals no significant shortcomings. Only a small offset error can be measured when speed hits its upper steady state.

4.2.3.4 Small-Signal Behavior

The small-signal behavior of the FCS-RL-DTC is featured in Fig. 4.8d. The momentary torque measurement is omitted for clarity in this case and only the moving average $\overline{T}$ is shown. In this plot, the control agent can be observed to react with no visible delay to the changing torque reference T^*. Again, some of the commanded operating points exhibit a visible torque offset, which can presumably be attributed to their proximity to the current boundary.

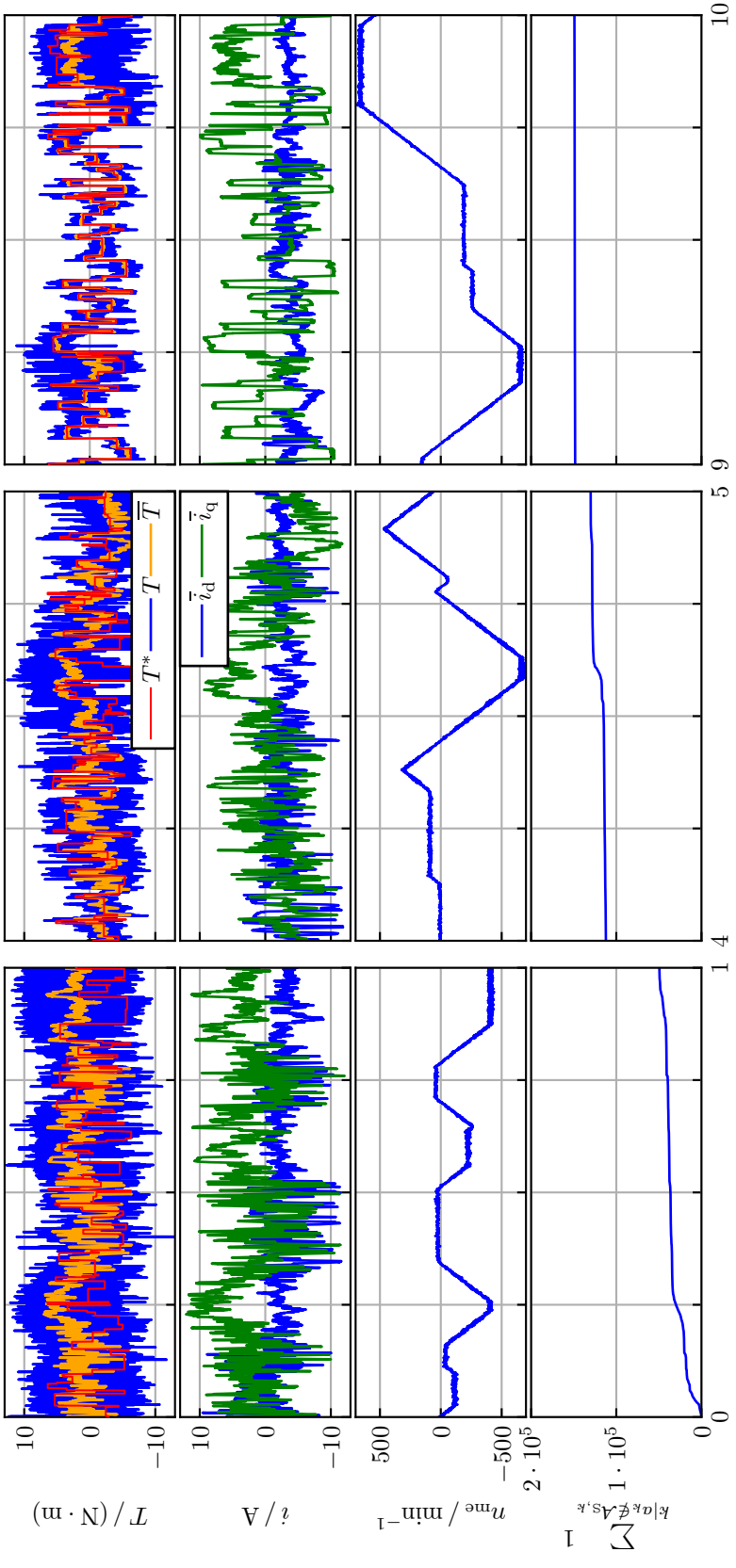


Fig. 4.5: Control performance in an early (left), intermediate (center) and late (right) phase of an exemplary FCS-RL-DTC training

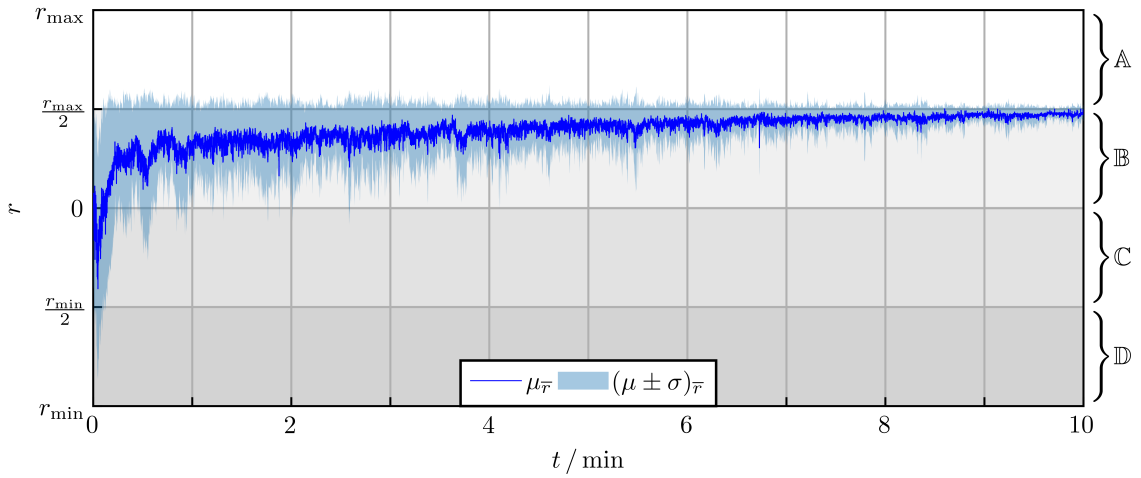


Fig. 4.6: Ensemble mean learning behavior $\mu_{\bar{r}}$ over ten separate FCS-RL-DTC trainings with highlighted variational range of one standard deviation $\sigma_{\bar{r}}$, moving average filter applied, the bottom plot depicts the cumulative amount of safeguard activations

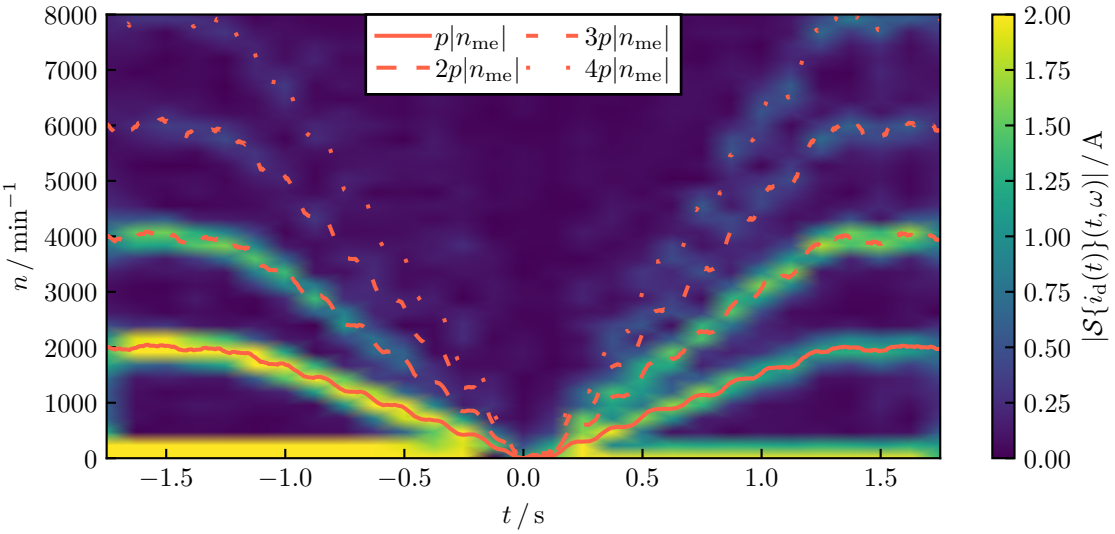
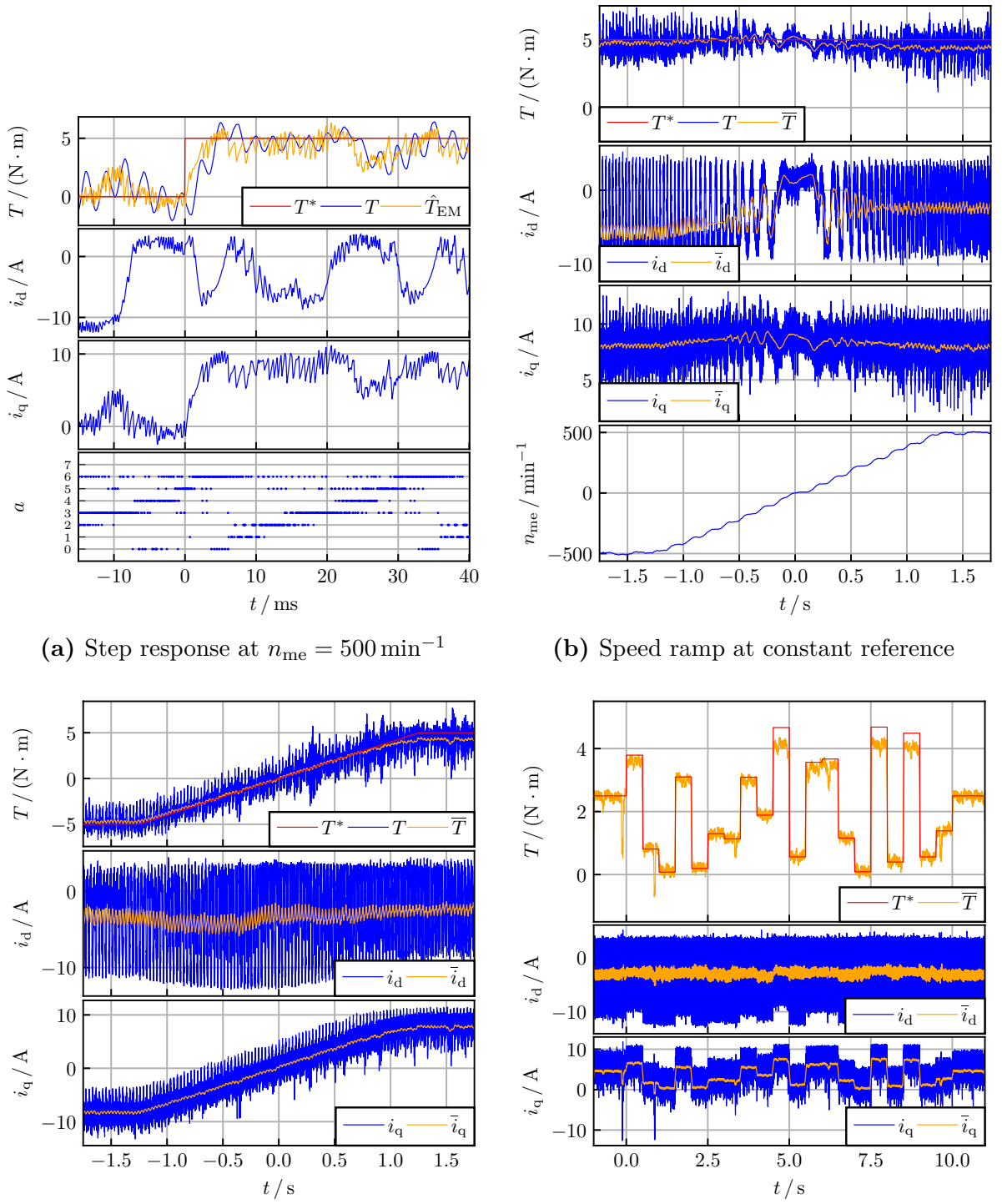


Fig. 4.7: Absolute of the short-time Fourier transform $\mathcal{S}(\cdot)$ of i_{d} for the speed ramp experiment from Fig. 4.8b;
The magnitudal scale is limited to the range of $[0\,\mathrm{A}, 2\,\mathrm{A}]$ for reasons of visualization, despite higher magnitudes have been observed for the DC component of i_{d}.

4.3 Discussion

Concerning all of the presented experiments, it is observed that $\bar{i}_\mathrm{d} \neq 0$, although such behavior seems counterintuitive when dealing with a surface PMSM. It is suboptimal in the sense of the MTPC premise that takes effect at lower speed and rather fits the MTPV demand that is only valid for higher speed. Therefore, it also seems suboptimal in the sense of the reward design that is derived with consideration of this circumstance (cf. Sec. 3.2). This observation may correspond to the observed (average) training behavior as of Fig. 4.6, wherein it is indicated that the reward region $\mathbb{A}$ (that incorporates the minimization of i_s and, hence, resembles the MTPC premise) is not reached reliably. A more thorough interpretation of the stated observations is spared at this point and will be presented at the end of the upcoming chapter, allowing to incorporate relevant outcome from the CCS experiment.

Apart from these open points concerning the efficiency of the proposed FCS-RL-DTC, it can be concluded that torque tracking control has been demonstrated. Training and application of the FCS-RL-DTC have been safely conducted in the presence of an empirically validated safeguarding program, the training phase converged reproducibly to the featured performance level, and the torque tracking quality itself has been verified in a multitude of scenarios. Herein, it was apparent that the tracking error was larger in motor operation while being almost negligible in generator operation. The proof of concept of the RL-based DTC with concern to the FCS implementation is therefore considered successful.



(a) Step response at $n_{\mathrm{me}} = 500\,\mathrm{min}^{-1}$

(b) Speed ramp at constant reference

(c) Torque reference ramp at $n_{\mathrm{me}} = 500\,\mathrm{min}^{-1}$

(d) Small-signal behavior at $n_{\mathrm{me}} = 500\,\mathrm{min}^{-1}$

Fig. 4.8: Test scenarios with the trained FCS-RL-DTC, the electromagnetic torque estimate $\hat{T}_{\mathrm{EM}}$ is only depicted for reference and was not available to the controller

5 Reinforcement Learning-Based Direct Torque Control on the Continuous Control Set

This chapter covers the application of the RL-DTC to the PMSM drive setup with CCS interface and relates to the publication [A3]. In the following, the sampling frequency is set to 10 kHz ($T_\mathrm{s} = 100\,\mu\mathrm{s}$), and the DDPG algorithm is applied to solve the CCS-RL problem (cf. Sec. A.3). A general scheme of the controller setup is depicted in Fig. 5.1. Again, the chapter firstly covers implementation of safeguarding procedures within the CCS before investigating experimental results.

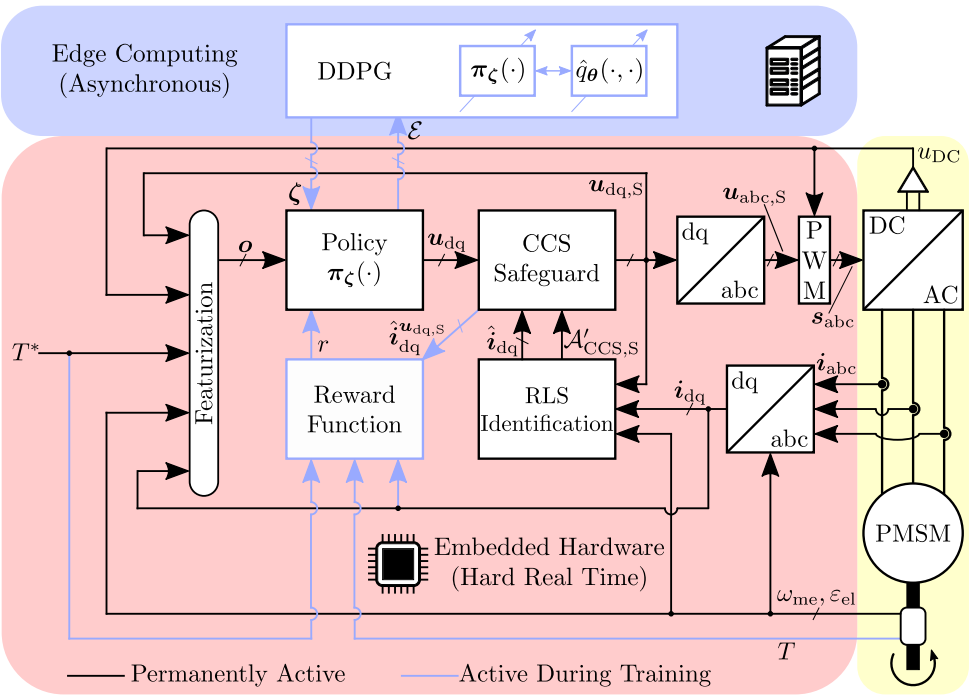


Fig. 5.1: Schematic of the CCS-RL-DTC with safeguarding

5.1 Continuous-Control-Set Safeguarding

In the following, the FCS safeguarding setup is extended to be applicable in the targeted modulator-driven CCS drive control loop. As before, its task is to prevent damage and avoid situations that would necessitate an emergency shutdown of the drive system by ensuring that any applied voltage $\boldsymbol{u}_{\mathrm{dq}}$ complies with the safety-critical operating conditions that have been discussed in Sec. 2.4.1. Identically to the FCS case as of Ch. 4, the controller has no access to the identified linear system approximation provided by the RLS (cf. Sec. 3.3), which is only utilized to safely restrict the system operation.

Whereas the FCS case in Ch. 4 (cf. [A2]) allowed exclusion of unsafe actions by simple evaluation of the conditions for all eight available switching actions, the safe set in the CCS case cannot be determined in the same manner of trial-and-error prediction. Instead, real-time-capable optimization methods will be utilized to ensure that the applied action satisfies all safety constraints. Such programs are commonly restricted to quadratic objective functions and linear constraints, which now poses as a boundary condition for the CCS safeguard. This procedure has its origin in MPC, wherein the optimization is simultaneously concerned with safety and control performance [2]. Contrary, the proposed safeguard only monitors adherence to the safety limitations without considering any performance-related metric, because the latter are within the scope of the RL policy and are herein, unlike MPC, principally evaluated on an infinite time horizon. This way, the one-step prediction that is applied by means of the safeguard is not restrictive with consideration to the achievable performance.

For mathematical formulation, the given representation of the available action set $\mathcal{A}_{\mathrm{CCS}}$ in (2.19) is quite illustrative concerning the physical architecture of the VSI. However, a description of $\mathcal{A}_{\mathrm{CCS}}$ that directly considers $\boldsymbol{u}_{\mathrm{dq}}$ is more compatible for the later utilization of numeric optimization solvers:

$$\mathcal{A}_{\mathrm{CCS}}[k] = \{\boldsymbol{u}_{\mathrm{dq}} \in \mathbb{R}^2 | \boldsymbol{G}_{\mathrm{h}} \boldsymbol{Q}_{\alpha\beta,\mathrm{dq}}(\varepsilon_{\mathrm{el}}[k]) \boldsymbol{u}_{\mathrm{dq}} \leq \boldsymbol{h}_{\mathrm{h}}\}, \tag{5.1a}$$

with

$$\boldsymbol{h}_{\mathrm{h}} = \frac{u_{\mathrm{DC}}[k]}{\sqrt{3}} \mathbf{1}, \qquad \boldsymbol{G}_{\mathrm{h}} = \begin{bmatrix} -\frac{\sqrt{3}}{2} & 0 & \frac{\sqrt{3}}{2} & \frac{\sqrt{3}}{2} & 0 & -\frac{\sqrt{3}}{2} \\ \frac{1}{2} & 1 & \frac{1}{2} & -\frac{1}{2} & -1 & -\frac{1}{2} \end{bmatrix}^{\top}, \tag{5.1b}$$

which represents the set of voltages that the VSI is physically capable of applying to the PMSM (cf. Fig. 2.3). $\mathbf{1}$ denotes a column vector of ones.

Again, the prediction model (3.9) is necessary for evaluation of compliance to the current limitation:

$$\hat{\boldsymbol{i}}_{\mathrm{dq}}[k+1] = \hat{\boldsymbol{A}}[k] \boldsymbol{i}_{\mathrm{dq}}[k] + \hat{\boldsymbol{B}}[k] \boldsymbol{u}_{\mathrm{dq}}[k] + \hat{\boldsymbol{e}}[k] = \underbrace{(\hat{\boldsymbol{A}}[k] \boldsymbol{i}_{\mathrm{dq}}[k] + \hat{\boldsymbol{e}}[k])}_{\boldsymbol{w}_i[k]} + \underbrace{\hat{\boldsymbol{B}}[k]}_{\boldsymbol{W}_i[k]} \boldsymbol{u}_{\mathrm{dq}}[k]. \tag{5.2}$$

Similarly, the equilibrium voltage $\hat{\boldsymbol{u}}_{\mathrm{dq,e}}[k+1]$ that must be matched by the input voltage $\boldsymbol{u}_{\mathrm{dq}}[k+1]$ to sustain operation at $\hat{\boldsymbol{i}}_{\mathrm{dq}}[k+1]$ is estimated via

$$\begin{aligned}\hat{\boldsymbol{u}}_{\mathrm{dq,e}}[k+1] &= \hat{\boldsymbol{B}}^{-1}[k]\left((\boldsymbol{I}-\hat{\boldsymbol{A}}[k])\hat{\boldsymbol{i}}_{\mathrm{dq}}[k+1]-\hat{\boldsymbol{e}}[k]\right)\\ &= \underbrace{\hat{\boldsymbol{B}}^{-1}[k]\hat{\boldsymbol{A}}[k]\left((\boldsymbol{I}-\hat{\boldsymbol{A}}[k])\boldsymbol{i}_{\mathrm{dq}}[k]-\hat{\boldsymbol{e}}[k]\right)}_{\boldsymbol{w}_u[k]}+\underbrace{\left(\boldsymbol{I}-\hat{\boldsymbol{B}}^{-1}[k]\hat{\boldsymbol{A}}[k]\hat{\boldsymbol{B}}[k]\right)}_{\boldsymbol{W}_u[k]}\boldsymbol{u}_{\mathrm{dq}}[k],\end{aligned} \tag{5.3}$$

with respective abbreviations $\boldsymbol{w}_{i,u}$ and $\boldsymbol{W}_{i,u}$ for the constant and linear parameter with concern to $\boldsymbol{u}_{\mathrm{dq}}$. To guarantee safety according to Sec. 2.4.1, the predicted current and voltage demand must satisfy

$$\|\hat{\boldsymbol{i}}_{\mathrm{dq}}[k+1]\|_2 \leq i_{\mathrm{n}}, \tag{5.4a}$$

$$\hat{\boldsymbol{u}}_{\mathrm{dq,e}}[k+1] \in \mathcal{A}_{\mathrm{CCS}}[k+1], \tag{5.4b}$$

for all

$$\boldsymbol{u}_{\mathrm{dq,S}}[k] \in \mathcal{A}_{\mathrm{CCS,S}}[k]. \tag{5.5}$$

Herein, $\boldsymbol{u}_{\mathrm{dq,S}}$ denotes an element of the safe action space $\mathcal{A}_{\mathrm{CCS,S}}$. (5.4b) corresponds to the recursive feasibility concept [108], invalidating state transitions that lead to violation of (5.4a) in the long run. Identically to (4.1), (5.4a) is represented by the elliptic subset:

$$\mathcal{C}_i[k] = \{\boldsymbol{u}_{\mathrm{dq}} \in \mathbb{R}^2 |\, \|\boldsymbol{w}_i[k] + \boldsymbol{W}_i[k]\boldsymbol{u}_{\mathrm{dq,S}}\|_2 \leq i_n\}, \tag{5.6}$$

which is a quadratic relation that cannot be handled by usual real-time-capable optimization solvers, as these require strictly linear inequality conditions. Hence, a linear approximation $\mathcal{C}_i'$ of the elliptic set (5.6) is utilized to yield a convex and polygonal set of linear inequalities defining the current range:

$$\mathcal{C}_i'[k] = \{\boldsymbol{u}_{\mathrm{dq}} \in \mathbb{R}^2 | \boldsymbol{G}_i[k]\boldsymbol{u}_{\mathrm{dq,S}} \leq \boldsymbol{h}_i[k]\}. \tag{5.7}$$

Herein, the computation of $\boldsymbol{G}_i$ and $\boldsymbol{h}_i$ from $\boldsymbol{W}_i$ and $\boldsymbol{w}_i$ (cf. (5.2)) follows purely geometric considerations, whose detailed description is omitted in this section in favor of an algorithmic approach stated in Sec. A.4. Notably, a resolution parameter $R \in \mathbb{N}$ must be specified for the linear approximation of an ellipsis. While a larger value of R is preferable in terms of numerical accuracy, it also increases the computational complexity. A conceptual impression of the procedure can be gained from Fig. 5.2 and Fig. A.8.

The recursive feasibility condition (5.4b) is projected to the voltage plane by analyzing compliance of the voltage demand (5.3) at the upcoming time step $k+1$ with the available action space (5.1a). As (5.1a) is a linear relation already, no further approximation must be employed and the condition evaluates to

$$\begin{aligned} & \boldsymbol{G}_{\mathrm{h}}\boldsymbol{Q}_{\alpha\beta,\mathrm{dq}}(\hat{\varepsilon}_{\mathrm{el}}[k+1])\hat{\boldsymbol{u}}_{\mathrm{dq,e}}[k+1] \leq \boldsymbol{h}_{\mathrm{h}} \\ \Leftrightarrow \quad & \boldsymbol{G}_{\mathrm{h}}\boldsymbol{Q}_{\alpha\beta,\mathrm{dq}}(\varepsilon_{\mathrm{el}}[k]+\omega_{\mathrm{el}}[k]T_{\mathrm{s}})(\boldsymbol{w}_u[k]+\boldsymbol{W}_u[k]\boldsymbol{u}_{\mathrm{dq,S}}[k]) \leq \boldsymbol{h}_{\mathrm{h}} \\ \Leftrightarrow \quad & \boldsymbol{G}_u[k]\boldsymbol{u}_{\mathrm{dq,S}}[k] \leq \boldsymbol{h}_u[k], \end{aligned} \tag{5.8}$$

with

$$\begin{aligned}\boldsymbol{G}_u[k] &= \boldsymbol{G}_\mathrm{h}\boldsymbol{Q}_{\alpha\beta,\mathrm{dq}}(\varepsilon_\mathrm{el}[k] + \omega_\mathrm{el}[k]T_\mathrm{s})\boldsymbol{W}_u[k],\\ \boldsymbol{h}_u[k] &= \boldsymbol{h}_\mathrm{h} - \boldsymbol{G}_\mathrm{h}\boldsymbol{Q}_{\alpha\beta,\mathrm{dq}}(\varepsilon_\mathrm{el}[k] + \omega_\mathrm{el}[k]T_\mathrm{s})\boldsymbol{w}_u[k],\\ \boldsymbol{u}_{\mathrm{dq,S}}[k] &\in \mathcal{A}_{\mathrm{CCS,S}}[k].\end{aligned} \tag{5.9}$$

Hence, the feasibility condition (5.8) leads to the set expression

$$\mathcal{C}_u[k] = \{\boldsymbol{u}_\mathrm{dq} \in \mathbb{R}^2 | \boldsymbol{G}_u[k]\boldsymbol{u}_\mathrm{dq} \leq \boldsymbol{h}_u[k]\}. \tag{5.10}$$

Because the current limitation is approximated by means of a linear proxy $\mathcal{C}_i'$, the stated considerations with respect to (5.7) and (5.10) lead to a linear approximation of the safe action space $\mathcal{A}_{\mathrm{CCS,S}} \rightarrow \mathcal{A}'_{\mathrm{CCS,S}}$:

$$\mathcal{A}'_{\mathrm{CCS,S}}[k] = \mathcal{A}_{\mathrm{CCS}}[k] \cap \mathcal{C}_i'[k] \cap \mathcal{C}_u[k] = \left\{\boldsymbol{u}_\mathrm{dq} \in \mathbb{R}^2 | \boldsymbol{G}[k]\boldsymbol{u}_\mathrm{dq} \leq \boldsymbol{h}[k]\right\}, \tag{5.11}$$

with

$$\begin{aligned}\boldsymbol{G}[k] &= \left[\left(\boldsymbol{G}_\mathrm{h}\boldsymbol{Q}_{\alpha\beta,\mathrm{dq}}\left(\varepsilon_\mathrm{el}[k]\right)\right)^\top \quad \boldsymbol{G}_i^\top[k] \quad \boldsymbol{G}_u^\top[k]\right]^\top,\\ \boldsymbol{h}[k] &= \left[\boldsymbol{h}_\mathrm{h}^\top \quad \boldsymbol{h}_i^\top[k] \quad \boldsymbol{h}_u^\top[k]\right]^\top,\end{aligned} \tag{5.12}$$

wherein all safety-critical operating conditions are incorporated. Finally, the safeguarding procedure of the CCS-RL-DTC can be embedded into a constrained optimization problem, whose solution yields a safely applicable voltage $\boldsymbol{u}_{\mathrm{dq,S}}$

$$\begin{aligned}\boldsymbol{u}_{\mathrm{dq,S}}[k] = \arg\min_{\boldsymbol{u}} \; &||\boldsymbol{u} - \boldsymbol{\pi}_{\boldsymbol{\zeta}}(\boldsymbol{o}[k])||_2^2 + c_\rho\rho\\ \text{s.t.} \; &\begin{bmatrix}\boldsymbol{G}_\mathrm{h}\boldsymbol{Q}_{\alpha\beta,\mathrm{dq}}(\varepsilon_\mathrm{el}[k]) & \boldsymbol{0}\\ \boldsymbol{G}_i[k] & -\boldsymbol{1}\\ \boldsymbol{G}_u[k] & -\boldsymbol{1}\\ \boldsymbol{0} & -1\end{bmatrix}\begin{bmatrix}\boldsymbol{u}\\ \rho\end{bmatrix} \leq \begin{bmatrix}\boldsymbol{h}_\mathrm{h}\\ \boldsymbol{h}_i[k]\\ \boldsymbol{h}_u[k]\\ 0\end{bmatrix}.\end{aligned} \tag{5.13}$$

Herein, the distance between $\boldsymbol{u}_{\mathrm{dq,S}}$ and $\boldsymbol{u}_\mathrm{dq} = \boldsymbol{\pi}_{\boldsymbol{\zeta}}(\boldsymbol{o})$ is minimized under consideration of all given constraints (cf. Fig. 5.2). ρ denotes a slack variable with weighting factor c_ρ that allows numerical satisfaction of these constraints even if the operational constraints exclude each other, which would result in $\mathcal{A}'_{\mathrm{CCS,S}} = \emptyset$. The same approach is often used in the MPC domain to numerically cope with disturbance and noise [116]. Introduction of this slack variable ensures that a numerical solution of (5.13) exists, even if no commandable voltage could be considered safe. In that case, the violation of the constraints cannot be avoided, but is kept minimal[1]. Naturally, this measure is only considered a fall-back solution: if the control agent is not already initialized in an unsafe operating point, (5.13) is supposed to limit the applied voltages to $\mathcal{A}'_{\mathrm{CCS,S}}$. However, also the presence of measurement noise might cause $\mathcal{A}'_{\mathrm{CCS,S}} = \emptyset$. While $\mathcal{A}'_{\mathrm{CCS,S}} \neq \emptyset$ it must result

[1]The VSI's voltage range $\mathcal{A}_{\mathrm{CCS}}$ (cf. (5.1a)) is physically impossible to violate due to the inverter setup and bounded DC-link voltage and, hence, a slack variable cannot be feasibly considered in the first line of the auxiliary conditions in (5.13).

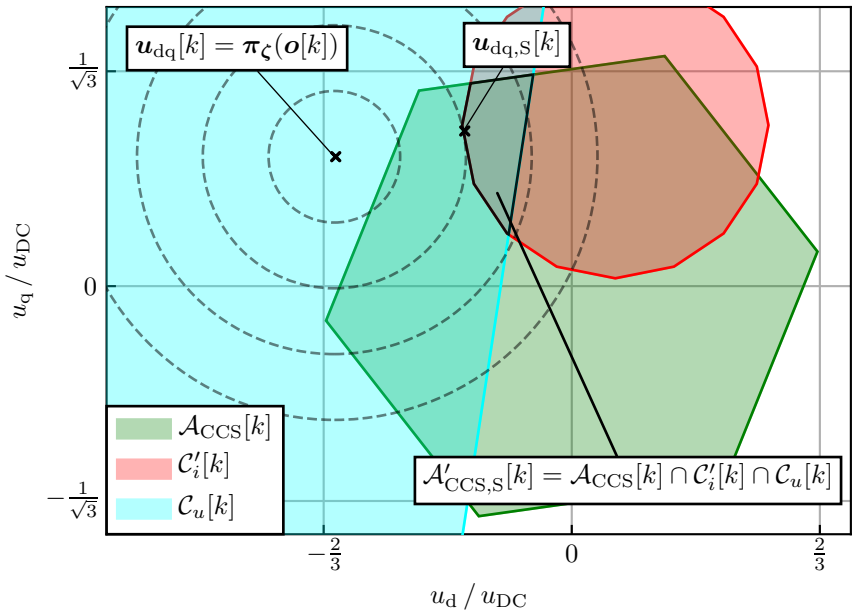


Fig. 5.2: Exemplary graphical construction of the approximate safe action set $\mathcal{A}'_{\mathrm{CCS,S}}$ in the voltage plane. The current set $\mathcal{C}_i$ is herein approximated by $R = 16$ linear inequations $\mathcal{C}'_i$. Dashed isolines denote points of equal distance to $\boldsymbol{u}_{\mathrm{dq}}[k]$.

that $\rho = 0$ and, hence, no performance loss or disturbance has to be accepted with this augmentation. Usually, it is preferable to introduce different slack variables that apply to different types of limitations, which could herein be realized by distinguished terms for current- and voltage-related limitations[2]. However, this approach would also come with added computational demand, because the optimization problem would be extended in its dimension. Therefore, only a single slack variable is considered here. Within the scope of this contribution, the active set solver from MATLAB is used for evaluation of this (soft-)constrained optimization problem [117].

5.2 Experimental Results

The proposed CCS-RL-DTC is empirically validated on the test bench drive system depicted in Fig. 3.3 and Fig. 3.4, whose components are specified in Tab. 3.2. The evaluation is conducted in three steps, following the same procedure as for the FCS case:

- Firstly, the safeguarding procedure from Sec. 5.1 is tested for its functionality.
- Secondly, the training phase is investigated concerning its safety and convergence.

[2]In the given setup, a necessary softening of one constraint does also entail an unnecessary softening of other constraints, resulting in the latter being potentially violated without the need to do so.

- Finally, the trained control agent's performance is evaluated in a multitude of scenarios, wherein the exploration noise is deactivated.

Within all experiments, the control cycle time is synchronized to the sampling time $T_\mathrm{s} = 100\,\mu\mathrm{s}$ (cf. Tab. 3.3), i.e., all required calculations have to be completed within $100\,\mu\mathrm{s}$, which is verified via measurements of the control turnaround time $T_\mathrm{C,TA}$ that are listed in Tab. 5.1. The further stated training turnaround time $T_\mathrm{T,TA}$ is not subjected to real-time conditions and corresponds to the time demand of one parameter update during the training process. The utilized CCS-RL algorithm DDPG and its configuration are comprehensively discussed in Sec. A.3, and the parameters of the safeguarding problem are configured to $R = 12$ (resolution of linear ellipsis approximation) and $c_\rho = 1 \cdot 10^4$ (slack variable to soften feasibility constraints).

Tab. 5.1: Statistical evaluation of the CCS-RL-DTC control turnaround time $T_\mathrm{C,TA}$ for the speed ramp experiment depicted in Fig. 5.8b and of the training turnaround time $T_\mathrm{T,TA}$ for an exemplary training procedure.

	μ	σ	$\|T_\mathrm{TA}\|_\infty$
Control $T_\mathrm{C,TA}$	78.145 µs	0.964 µs	89.920 µs
Training $T_\mathrm{T,TA}$	11.660 ms	3.715 ms	0.795 s

5.2.1 Safeguard Functionality

Before actual learning experiments are conducted, the safeguard's functionality is tested. For this, the safeguard is being confronted with random actions in a practical experiment, in order to provoke intervention whenever necessary.

A long-term current trajectory is recorded for each of the two scenarios: negligible speed and significant speed. The corresponding trajectories are depicted in Fig. 5.3 and Fig. 5.4, respectively. Herein, the set of visited currents can be interpreted as the subset of the state space that the safeguarding procedure (5.13) evaluates to be safe. As can be seen, the current boundary is closely adhered to, with violations being of insignificant magnitude. The voltage boundary cannot be uniquely mapped to the current plane for the given experiment, because it depends on the DC-link voltage u_DC (cf. (5.1), (5.9)), which, in the employed setup, changes rapidly during current transients. The set of possible boundary trajectories can be seen in Fig. 5.4. For this, no definitive statement can be made about the safeguard's functionality yet, as each operating point needs to be evaluated in consideration of the momentary u_DC, which cannot succeed from a state-space plot. The change rate of u_DC is mainly dependent on the DC-link capacitance and its supply but, in contrast to the presented experiment, u_DC can be assumed constant in most commercial applications.

Therefore, the momentary voltage reserve is determined with the corresponding histogram being depicted in Fig. 5.5. In this form, it is easily visible that a voltage reserve is available

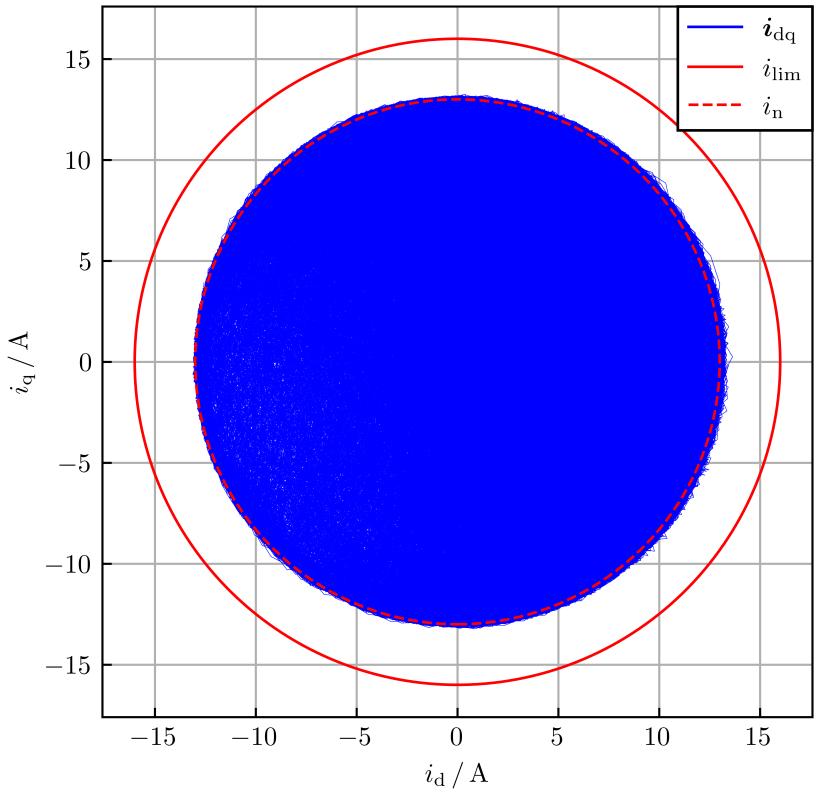


Fig. 5.3: Operation at low speed $|n_{\mathrm{me}}| \leq 50\,\mathrm{min}^{-1}$, only the current boundary is active

almost the entire time, which validates the safeguard's functionality. The fraction of samples in which voltage deficiency (i.e., a negative voltage reserve) is apparent, can hardly be seen from the plot. Statistical investigation gives away that a voltage reserve was available for 99.994 % of the time, according to the system matrices that were identified by the RLS. Consequently, the safeguarding routine is considered as valid as long as the assumption of sufficiently accurate system identification is fulfilled.

To check whether the RLS identification is of sufficient precision, the experiment's data from Fig. 5.3 and Fig. 5.4 are furthermore analyzed with concern to achieved prediction accuracy. The corresponding results can be viewed in Tab. 5.2. As can be seen, the mean and standard deviation of the prediction error are in the range of $10^{-2}\,\mathrm{A}$, which is insignificant given the nominal current of $i_{\mathrm{n}} = 13\,\mathrm{A}$. The worst-case prediction error $\max|e|$ for the high-speed case is significant, however, the preceding investigations Fig. 5.4 and Fig. 5.5 suggest that such error occurs at lower current magnitudes, where neither the current nor the voltage boundary is critical.

In total, it can therefore be concluded that both parts of the safeguarding procedure fulfill their task satisfactory. While the RLS provides a prediction model $\hat{\boldsymbol{A}}, \hat{\boldsymbol{B}}, \hat{\boldsymbol{e}}$ of sufficient precision, the operation in unsafe states can be avoided reliably by means of the optimization problem (5.13).

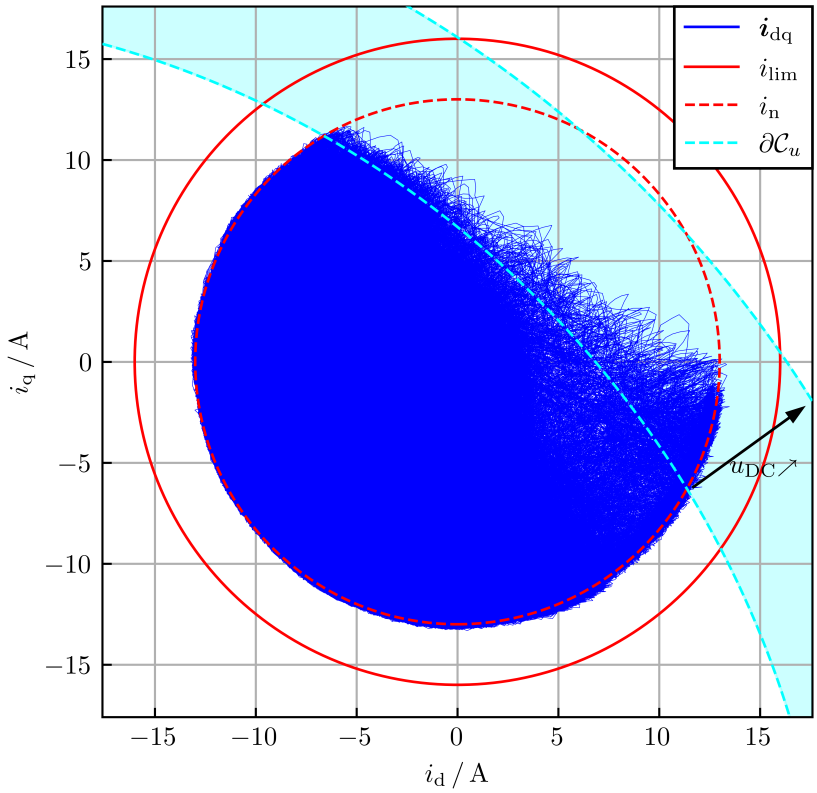


Fig. 5.4: Operation at high speed $n_{\mathrm{me}} = 700\,\mathrm{min}^{-1}$, current and voltage boundaries are active

5.2.2 Training Phase

The second experimental investigation is concerned with the composition, safety and convergence behavior of the training phase. For this, a total of ten training runs are conducted with ten minutes duration respectively. The hyperparameters are identical in each training run, and are directly selected as of the best-performing set from Tab. A.3. During the training, the reference torque T^* is randomly changed in a step-wise manner. The speed n_{me} is controlled via the load motor and is randomly changed with ramp-like behavior. The numeric training configuration is listed in Tab. 5.3.

The convergence behavior of all ten conducted trainings is analyzed via the controller's mean and variance of the reward r over the course of the ten minutes. This investigation is depicted in Fig. 5.6, and it can be seen that the training process generally converges to reward values between regions $\mathbb{B}$ and $\mathbb{A}$ (cf. Sec. 3.2), which means that T^* can be provided almost the entire time. Optimization of the operating point concerning MTPC behavior, which corresponds to the upper end of $\mathbb{A}$, cannot be inferred from this investigation. This observation will be further discussed in Sec. 5.3. Interestingly, the reward variance is not as convergent as in the corresponding FCS investigation Fig. 4.6, despite the exploration parameter being slowly set to zero over the course of the training (cf. Sec. A.3). In

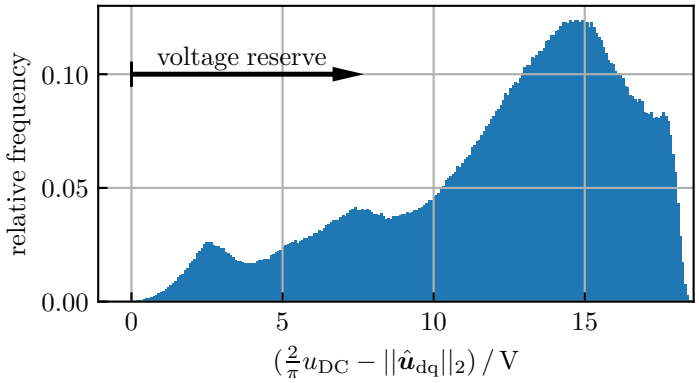


Fig. 5.5: Histogram of the available voltage reserve for the experiment depicted in Fig. 5.4, the voltage demand $||\hat{\boldsymbol{u}}_{\mathrm{dq}}||_2$ that is needed to maintain the momentary operating point is computed according to (5.3).

Tab. 5.2: Statistical evaluation of the one-step absolute current prediction error $|e_{\mathrm{d,q}}| = |\hat{i}_{\mathrm{d,q}} - i_{\mathrm{d,q}}|$ with mean μ, standard deviation σ and worst-case prediction error $\|e\|_\infty$ to validate the accuracy of the data-driven RLS identification

n_{me}	μ	σ	$\|e\|_\infty$
$\leq 50\,\mathrm{min}^{-1}$	$\mu_{\lvert e_{\mathrm{d}}\rvert} = 4.081 \cdot 10^{-2}\,\mathrm{A}$	$\sigma_{\lvert e_{\mathrm{d}}\rvert} = 3.535 \cdot 10^{-2}\,\mathrm{A}$	$\|e_{\mathrm{d}}\|_\infty = 5.286 \cdot 10^{-1}\,\mathrm{A}$
	$\mu_{\lvert e_{\mathrm{q}}\rvert} = 4.152 \cdot 10^{-2}\,\mathrm{A}$	$\sigma_{\lvert e_{\mathrm{q}}\rvert} = 3.241 \cdot 10^{-2}\,\mathrm{A}$	$\|e_{\mathrm{q}}\|_\infty = 4.100 \cdot 10^{-1}\,\mathrm{A}$
$700\,\mathrm{min}^{-1}$	$\mu_{\lvert e_{\mathrm{d}}\rvert} = 4.034 \cdot 10^{-2}\,\mathrm{A}$	$\sigma_{\lvert e_{\mathrm{d}}\rvert} = 3.740 \cdot 10^{-2}\,\mathrm{A}$	$\|e_{\mathrm{d}}\|_\infty = 1.849\,\mathrm{A}$
	$\mu_{\lvert e_{\mathrm{q}}\rvert} = 4.521 \cdot 10^{-2}\,\mathrm{A}$	$\sigma_{\lvert e_{\mathrm{q}}\rvert} = 3.941 \cdot 10^{-2}\,\mathrm{A}$	$\|e_{\mathrm{q}}\|_\infty = 1.425\,\mathrm{A}$

general, however, the training safety and convergence can be confirmed and the controller is capable of tracking the reference torque after ten minutes of training time.

An exemplary training time series is depicted in Fig. 5.7, where changes to both, the drive speed and the reference torque can be seen. Moreover, the bottom plot in Fig. 5.7 shows the cumulative number of safeguard activations, indicating that the safeguard is most challenged upon startup. Over the course of all trainings, no safety violation has been registered and no emergency shutdown was necessary.

5.2.3 Torque-Tracking Behavior

Analogously to the FCS case, the tracking performance is analyzed in four different experiments:

- torque reference step at constant speed
- speed ramp from negative to positive speed at constant torque reference
- torque reference ramp at constant speed

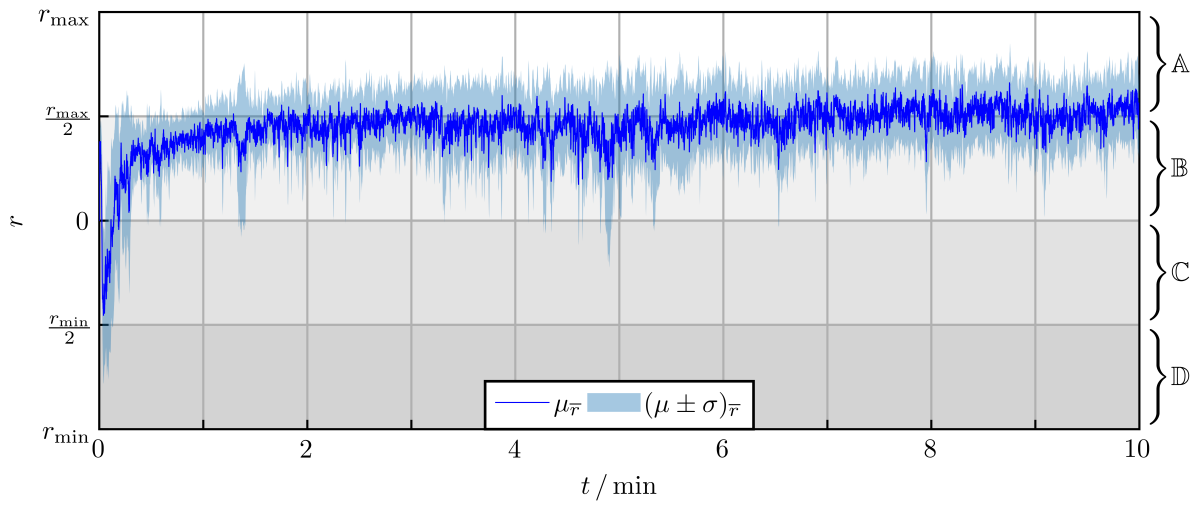


Fig. 5.6: Ensemble mean learning behavior $\mu_{\bar{r}}$ over ten separate CCS-RL-DTC trainings with marked variational range of one standard deviation $\sigma_{\bar{r}}$, moving average filter applied

Tab. 5.3: Training configuration of the CCS-RL-DTC

Configuration Parameter	**Specification**
Torque reference range	$[-8\,\mathrm{N \cdot m}, 8\,\mathrm{N \cdot m}]$
Torque reference change probability	$5 \cdot 10^{-4}$
Speed range	$[-700\,\mathrm{min}^{-1}, 700\,\mathrm{min}^{-1}]$
Speed change probability	10^{-4}
Maximum acceleration	$100\,\frac{\mathrm{min}^{-1}}{\mathrm{s}}$
Training duration	10 min

- small-signal investigation, i.e., several torque reference steps at constant speed.

For these investigations, the best-scoring control agent from the previous section (ten trainings with ten minutes duration each) is applied. Herein, the decision is made on the basis of highest cumulative reward during training. The four given scenarios are documented in Figs. 5.8a-5.8d, and are discussed in more detail in the following.

5.2.3.1 Torque Reference Step

The time series of a torque reference step is depicted in Fig. 5.8a. While the measured torque T oscillates visibly, the severeness of that oscillation has been reduced significantly when compared to the FCS case in Ch. 4. Yet, the agent has not learned to actively dampen the oscillation, which might be an interesting scope for future research. Further, it can be seen that both, the measured and the electromagnetic torque underpass the

reference slightly. This effect presumably results from the given torque tracking tolerance T_{tol} the controller was trained for (cf. Tab. 3.3).

The electromagnetic torque $\hat{T}_{\mathrm{EM}}$ tracks the reference with a very fast response time of 2.2 ms, which also is a substantial improvement concerning the FCS case Fig. 4.8a, wherein a rise time of 5 ms was observed.

5.2.3.2 Speed Ramp

The measurement plot in Fig. 5.8b showcases the torque-tracking fidelity at changing speed. Herein, it can be seen that T^* is within reach of the torque measurement oscillation for almost the whole time. However, especially after entering motor operation ($n_{\mathrm{me}} > 0$), the reference value is underpassed slightly.

In the FCS application from Ch. 4, a striking harmonic oscillation of i_{d} was visible, whose frequency corresponded to the motor speed (cf. Fig. 4.8b and Fig. 4.7). This behavior is not observed anymore within the CCS application. A similarity to the FCS case, however, is the presence of an offset $i_{\mathrm{d}} < 0$ that persists within the full speed range. Its origin is discussed more comprehensively at the end of this chapter.

5.2.3.3 Torque Reference Ramp

A time-series recording of a torque reference ramp is presented in Fig. 5.8c. Again, the tracking precision lacks slightly in motor operation ($T > 0$), but seems accurate in generator operation. Further, the presence of higher order harmonics looks negligible in both, i_{q} and T, but is quite visible in i_{d}, where it should have no effect concerning torque generation.

5.2.3.4 Torque Reference Profile

Fig. 5.8d depicts the small-signal behavior of the torque controller by means of several reference steps with lower magnitude at constant speed. In accordance with the preceding experiments, a steady-state offset between T^* and T can also be observed here, as only motor operation is covered. Again, a negative offset and some harmonic content is striking in i_{d}, but the improvement concerning superimposed harmonics is obvious in comparison to Ch. 4.

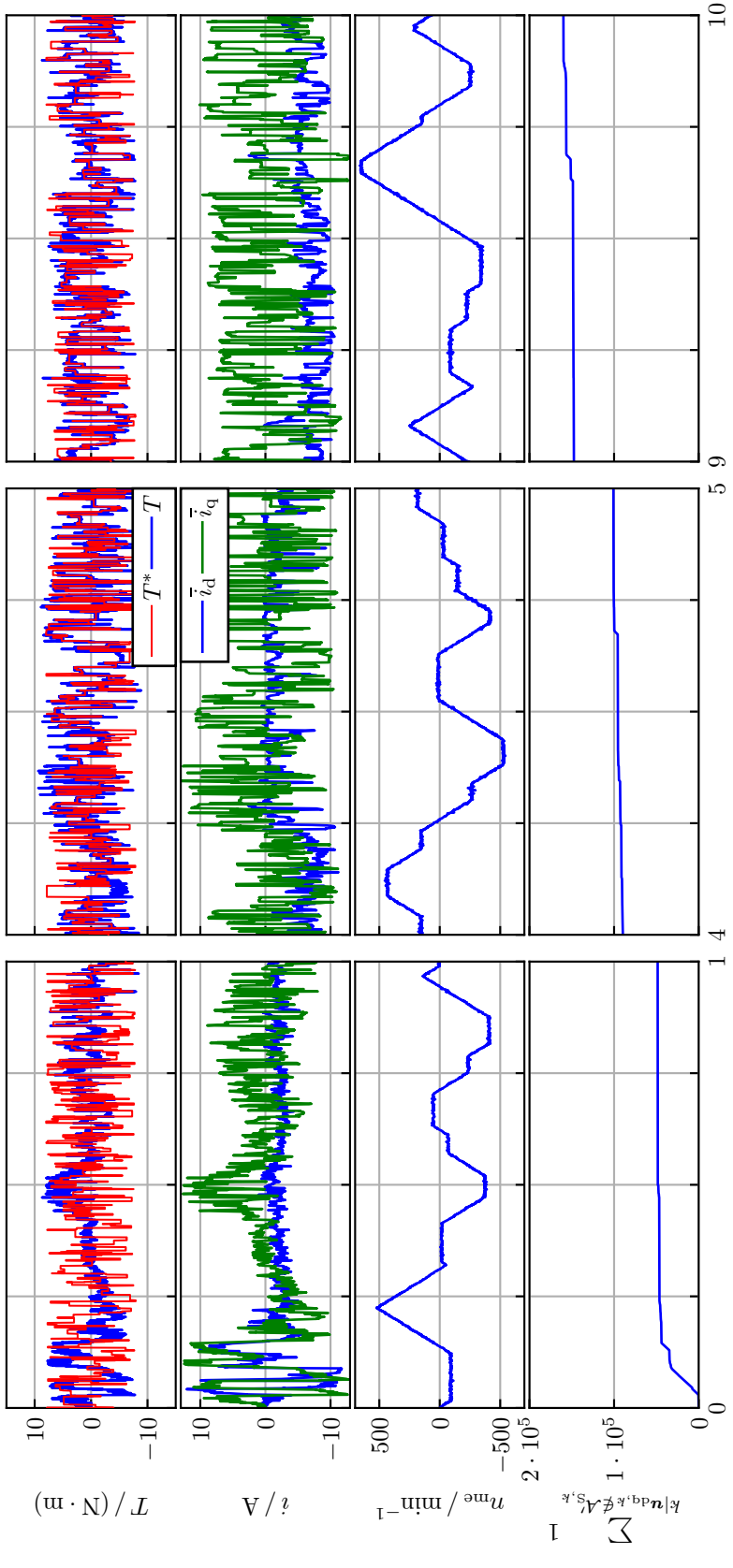
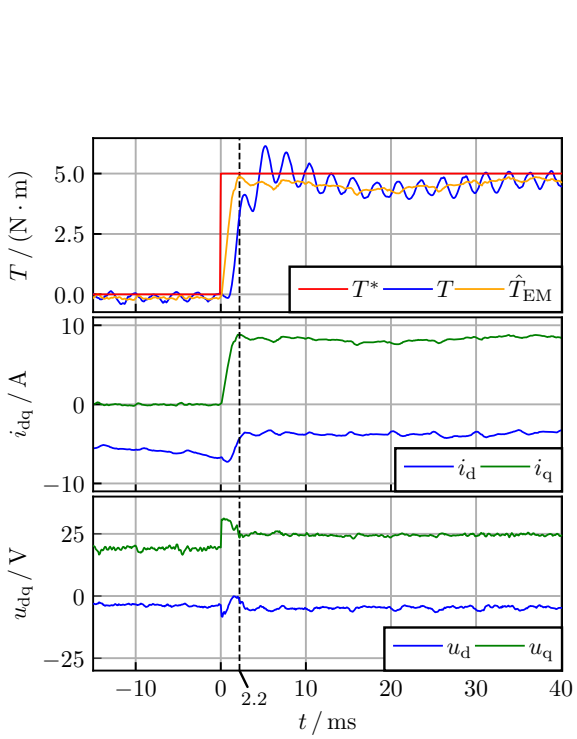
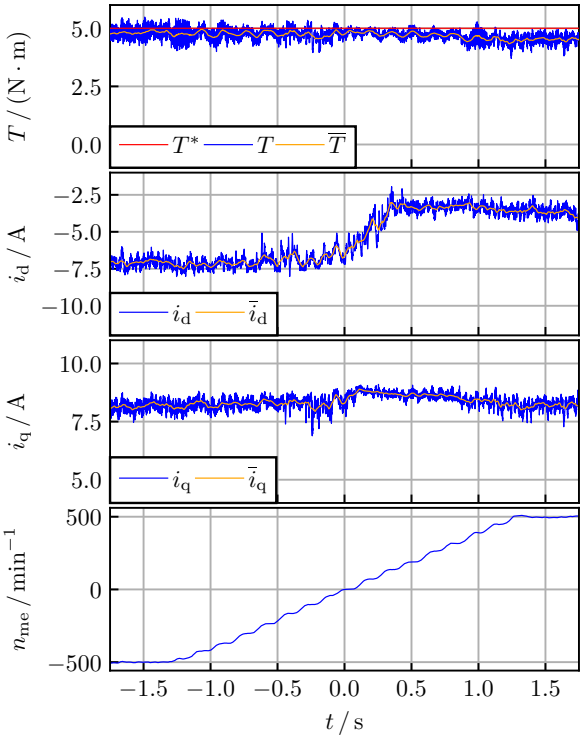


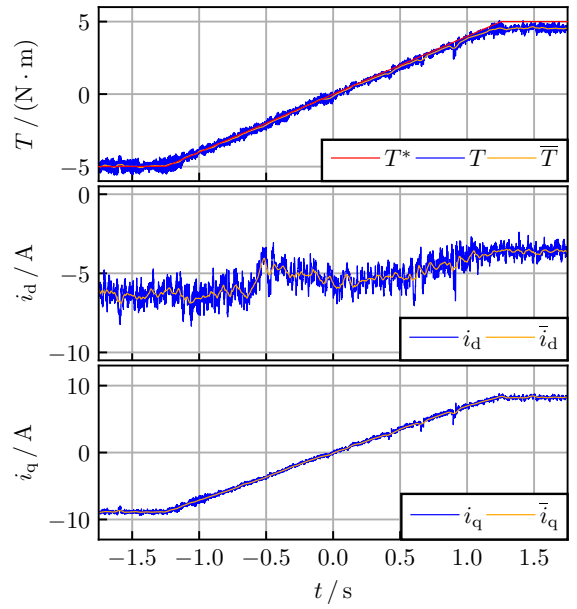
Fig. 5.7: Control performance in an early (left), intermediate (center) and late (right) phase of an exemplary CCS-RL-DTC training, the bottom plot depicts the cumulative amount of safeguard activations



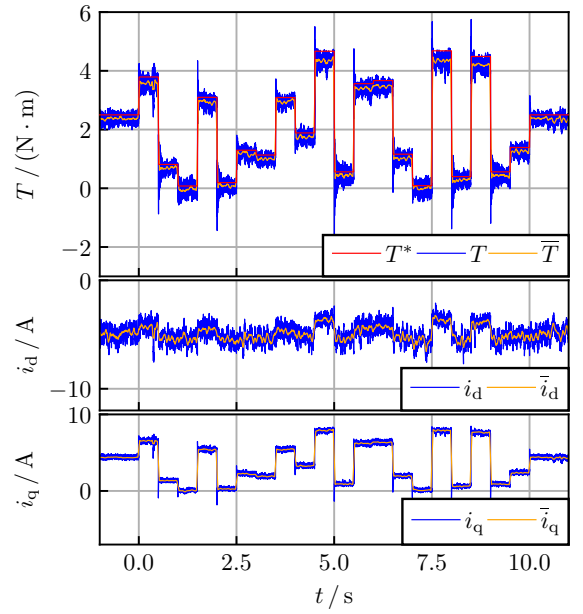
(a) Step response at $n_{\mathrm{me}} = 500\,\mathrm{min}^{-1}$



(b) Speed ramp with constant reference



(c) Torque reference ramp at $n_{\mathrm{me}} = 500\,\mathrm{min}^{-1}$



(d) Small-signal behavior at $n_{\mathrm{me}} = 500\,\mathrm{min}^{-1}$

Fig. 5.8: Test scenarios with the trained CCS-RL-DTC, the torque estimate $\hat{T}_{\mathrm{EM}}$ is only depicted for reference and was not available to the controller

5.3 Discussion

As already observed for the FCS implementation (cf. Sec. 4.3) of the RL-DTC, also the CCS version features a significant $i_\mathrm{d} < 0$, which must be considered suboptimal when operating a surface-mounted PMSM under MTPC premise. As such a scenario would prefer $i_\mathrm{d} = 0$ in stationary state, possible causes for this outcome are discussed in the following:

1) The information that the given motor is a surface PMSM has not been used for setting up the RL controller and, hence, it is not initially clear that $i_\mathrm{d} = 0$ should be targeted for maximizing efficiency, but it must be learned during the training phase. That corresponding behavior was not included in the final control agent might indicate that more training is necessary. Such an issue could be resolved by increasing training time or by using more potent workstation hardware and software.

2) Operation at $i_\mathrm{d} < 0$ would be expected at higher speed, where MTPC operation loses priority in favor of MTPV operation. Due to the diverse training phase incorporating all types of mechanical loading, however, it seems rather unlikely that the training could be biased towards MTPV behavior.

3) The selection of $i_\mathrm{d} = 0$ is only optimal in terms of steady-state efficiency, which is given a lower priority than torque tracking error reduction by means of Sec. 3.2. For reward accumulation during the training phase, it could have been more beneficial to maintain $i_\mathrm{d} < 0$, as this could increase reaction speed, allowing faster transients. For the given training scenario with consecutive random torque reference changes of arbitrary magnitude (cf. Fig. 4.5, Fig. 5.7), this mechanism could plausibly be the dominant reason for the observed behavior. If so, operation with $i_\mathrm{d} < 0$ would be a consequence of the training profile's performance requirements and could, hence, be handled by changing the training procedure to prioritize the stationary state more prominently (e.g., by decreasing the probability of reference changes).

4) As of the observed training convergence Fig. 4.6 and Fig. 5.6, reward region $\mathbb{A}$ does not seem primarily relevant during the first ten minutes of training. While this could be a hint at incomplete training as of argument 1), it could also be an effect of the drive train's parasitic oscillations that have been observed throughout all experiments (cf. Fig. 4.8, Fig. 5.8). While an increase of the permitted torque tracking error tolerance T_tol would lead to region $\mathbb{A}$ being entered more frequently (encouraging a more visible decrease of the stator current), it would at the same time decrease the tracking performance. To avoid such a compromise, the training should be conducted on a drive train with sufficient stiffness, which would be less prone for oscillation. However, an algorithmic solution that does not require physical modifications to the system would be more preferable from a control engineering perspective, with [118] being a promising approach to eliminate the T_tol parameter entirely, unifying regions $\mathbb{B}$ and $\mathbb{A}$ (torque tracking and minimization of ohmic losses, respectively) without the necessity to manually tune the compromise between both.

The question concerning the significant i_d current is yet to be resolved when targeting to draw even with the established class of model-based torque control methods from the MPC and FOC domain, as is the origin of the differing torque tracking precision between motor and generator operation. Still, it is to be summarized that the proof of concept for this approach was successful.

6 Conclusion and Outlook

6.1 Conclusion

In this work, a reinforcement learning direct torque controller (RL-DTC) was conceptualized for permanent magnet synchronous motors (PMSMs) and demonstrated in real-world experiments on a surface-mounted PMSM test bench. The successful implementation is to be understood as a proof of concept, which – to the author's best knowledge – was the first of its kind, and has been rolled out to finite-control-set (FCS) and continuous-control-set (CCS) operation of the voltage source inverter.

Herein, the data-driven training phase of ten minutes duration has been enabled by designing a parameter-independent reward function. It quantifies the success of safe, precise and efficient operation, while only requiring the electric and mechanic measurement quantities of current, angle, speed and torque, as well as their safety boundaries. Beyond that, no knowledge about the electric or mechanic drive parameterization has been assumed for training the RL torque controller. The reward function was then utilized for adapting the parameters of artificial neural networks (ANNs), which are concerned with solving the RL-DTC problem at runtime.

In order to deal with the increased computational demand of ANN inference within sampling periods of 50 µs (FCS) to 100 µs (CCS), the corresponding evaluation has been implemented on field-programmable gate array (FPGA) hardware. Beside real-time-capable inference, this implementation allowed reparameterization of the ANN at run-time, which was critical for the online learning phase. The corresponding RL algorithm has been outsourced to more powerful workstation hardware without real-time capability, rendering the training setup an edge computing application.

To safely proceed through the learning phase, a safeguarding routine has been developed for both, the FCS and the CCS scenario. This algorithm evaluates potential state transitions in consideration of their momentary and foreseeable adherence to the PMSM's current limitation with respect to the input voltage constraints. It is enabled by recursive least squares online system identification, which provided an abstract prediction model that was used to decide to overrule the RL-DTC whenever its native voltage selection was at risk of provoking an emergency shutdown.

Finally, the full setup consisting of edge-learning pipeline, FPGA inference, parameter-independent controller optimization and safeguarding procedure has been applied in both, the FCS and the CCS scenario. While a performance gap to model-based controllers must yet be accepted, it has been verified that safe training and feasible torque tracking is possible with the proposed RL-DTC.

Remaining limitations of the approach correspond to the complex initial setup and hardware-intensive training process, which resulted in a controller with precise and responsive tracking capability, but with improvable efficiency. The short training time of just ten minutes indicates that the RL-DTC can be easily transferred to additional PMSM drives with minimal human intervention, which is a unique characteristic of this work. Such ease of applicability represents a significant practical benefit that extends beyond conventional control performance metrics.

Given the relatively recent introduction of RL in electrical power systems, it is reasonable to expect that the lacking efficiency achieved by the proposed approach can be overcome in the future. Moreover, with the continued integration of artificial intelligence in both, technical and everyday applications, hardware costs are likely to decline, making the required components increasingly affordable. Nonetheless, from today's perspective, the adoption of RL-based drive controllers in industrial manufacturing or transportation remains uncertain. Most prominently, the lack of conventional stability guarantees must presently be weighed against the benefit of minimum-interaction controller synthesis.

6.2 Outlook

The practical feasibility of RL-DTC as demonstrated encourages a multitude of further research topics to increase the control performance and widen the application range in order to increase the overall acceptance for corresponding approaches. Upon the provided basis, several research questions can be investigated in continuation of this effort:

- As discussed in Sec. 5.3, measures are to be found that allow a more reliable reduction of the stator current i_s as dictated by the targeted MTPC premise. After this, a performance and efficiency comparison with classical torque control methods, such as FOC and MPC, can be pursued. In this context, it is also of interest to optimize for general efficiency instead of reducing copper losses only. Herein, accurate efficiency measurement at runtime may pose as challenging [119].

- Notable research questions in the domain of synchronous drives include the operation of sensorless architectures, which are mainly challenging in terms of satisfying the Markov property, the optimization of pulse patterns, which would necessitate RL algorithms that are equipped for hybrid action sets, and the extension of the proposed RL-DTC to externally excited synchronous motors, for which initial investigation has been presented in [C16], without yet addressing the topic of safeguarding.

- Moving the proposed approach to induction motor drives, it would be necessary to replace the usually applied flux observer with a model-free alternative that allows to define an observation vector in the sense of the Markov property. Correspondingly, the safeguarding procedure must be formulated with regard to measureable quantities instead of utilizing flux or rotor current as state variables, as these are unavailable when not accessing parameter knowledge.
- Overmodulation and, particularly, six-step operation, which is of interest when approaching the drive's power limit, has not been investigated in this thesis. First attempts to incorporate this operation mode, e.g., [C14], did not yet indicate an obvious method how to transition to block commutation reliably. Moreover, it needs investigation whether this mode is compatible with the proposed safeguarding technique.
- Highly utilized PMSMs might pose as a safety risk when applying the RLS to parameterize the safeguarding algorithm, because the effects of magnetic saturation might be changing the system behavior at a faster pace than the RLS is capable of identifying. To avoid limit violations during transients, it may be pragmatic to utilize long-term memory RLS [120] to deal with such dynamics.
- The concept of meta-RL allows a smooth transfer of the learned control behavior from one PMSM to another without the necessity to undergo a further training phase. First approaches have demonstrated feasibility with respect to the current control problem [C13] and the FCS-RL-DTC [C20] already.

Appendix

A.1 Maximum Fundamental Voltage of the VSI

The voltage supplied by the VSI is limited in dependency of the DC-link voltage u_{DC} that is momentarily available. For safe utilization of the motor, it is of major interest to have the voltage demand not exceed the supply capability of the inverter. Herein, the voltage demand of an operating point is mainly characterized by the magnitude of the fundamental voltage $\boldsymbol{u}_{\alpha\beta,\mathrm{f}}$, which corresponds to the harmonic component of the applied voltage vector $\boldsymbol{u}_{\alpha\beta}$ that rotates with electric angular velocity ω_{el}. Instead of the momentary fundamental voltage magnitude $\|\boldsymbol{u}_{\alpha\beta,\mathrm{f}}\|_2$, feasible steady-state operating points of the drive are characterized by the average voltage demand over the course of one rotation $\overline{\|\boldsymbol{u}_{\alpha\beta,\mathrm{f}}\|}_2$. By principle, $\overline{\|\boldsymbol{u}_{\alpha\beta,\mathrm{f}}\|}_2$ is maximized when only the voltage vectors with maximum magnitude are applied, referring to the corners of the voltage hexagon as also depicted in Fig. A.1. The specific value of $\max \overline{\|\boldsymbol{u}_{\alpha\beta,\mathrm{f}}\|}_2$ can then be evaluated to

$$\begin{aligned}
\max \overline{\|\boldsymbol{u}_{\alpha\beta,\mathrm{f}}\|}_2 &= \frac{1}{2\pi} \int_0^{2\pi} \left(\arg\max_{\boldsymbol{u}_{\alpha\beta}} \|\boldsymbol{u}_{\alpha\beta,\mathrm{f}}(\varepsilon)\|_2 \right)^{\top} \begin{bmatrix} \cos(\varepsilon) \\ \sin(\varepsilon) \end{bmatrix} \mathrm{d}\varepsilon \\
&= \frac{12}{2\pi} \int_0^{\frac{\pi}{6}} \begin{bmatrix} \frac{2}{3} u_{\mathrm{DC}} & 0 \end{bmatrix} \begin{bmatrix} \cos(\varepsilon) \\ \sin(\varepsilon) \end{bmatrix} \mathrm{d}\varepsilon \\
&= \frac{4}{\pi} u_{\mathrm{DC}} \underbrace{\int_0^{\frac{\pi}{6}} \cos(\varepsilon) \mathrm{d}\varepsilon}_{=\frac{1}{2}} \\
&= \frac{2}{\pi} u_{\mathrm{DC}}.
\end{aligned} \tag{A.1}$$

Wherein it has been exploited that the voltage hexagon can be disassembled into twelve identical triangular sections that each cover an angle of $\frac{\pi}{6}$.

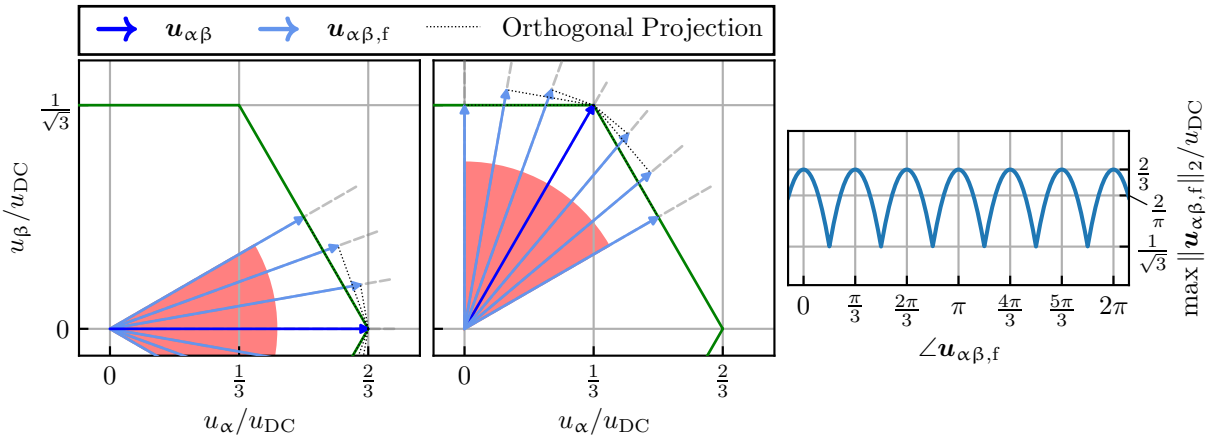


Fig. A.1: Depiction of the applied voltage $\boldsymbol{u}_{\alpha\beta}$ in six-step operation and its fundamental component $\boldsymbol{u}_{\alpha\beta,\mathrm{f}}$ for $-\frac{\pi}{6} \leq \angle\boldsymbol{u}_{\alpha\beta,\mathrm{f}} \leq \frac{\pi}{6}$ (left) and $\frac{\pi}{6} \leq \angle\boldsymbol{u}_{\alpha\beta,\mathrm{f}} \leq \frac{\pi}{2}$ (center), on the right: fundamental voltage magnitude $\|\boldsymbol{u}_{\alpha\beta,\mathrm{f}}\|_2$ in dependency of fundamental voltage angle $\angle\boldsymbol{u}_{\alpha\beta,\mathrm{f}}$

A.2 Reinforcement Learning in the FCS: Deep q Network

This section targets to introduce the deep q network (DQN) algorithm, which is the most established algorithm to learn the state-action value function q with usage of an ANN in systems with continuous state and finite action space [121, 122]. Aligning with (2.32), it is assumed that an observation vector $\boldsymbol{o}$ is available that satisfies the Markov property (2.24) for the given plant system.

To approximate q by means of an ANN $\hat{q}_{\boldsymbol{\theta}}$ with network weights $\boldsymbol{\theta}$, a cost function must be formulated to allow training / optimization of $\boldsymbol{\theta}$. Firstly, please note that according to (2.25) and (2.23), the state-action value approximation[1] must satisfy the Bellman equation [86]:

$$\hat{q}_{\boldsymbol{\theta}}(\boldsymbol{o}[k], a[k]) = r[k+1] + \gamma \hat{q}_{\boldsymbol{\theta}}(\boldsymbol{o}[k+1], a[k+1]). \tag{A.2}$$

This property is exploited in many pertinent RL algorithms because it allows the formulation of a cost function J_q for optimization: the left-hand side and the right-hand side of (A.2) should be equivalent, which means their (quadratic) difference is to be minimized, yielding

[1] The hat notation $\hat{x}$ denotes an estimation / approximator for x.

$$J_q(\boldsymbol{\theta}) = \frac{1}{|\mathcal{B}|} \sum_{\mathcal{E} \in \mathcal{B}} \left(\hat{q}_{\boldsymbol{\theta}}(\boldsymbol{o}[k], a[k]) - \underbrace{\left(r[k+1] + \gamma(1 - \tau[k+1]) \max_{a' \in \mathcal{A}_{\mathrm{FCS}}} \hat{q}_{\tilde{\boldsymbol{\theta}}}(\boldsymbol{o}[k+1], a') \right)}_{\text{estimation target}} \right)^2 . \tag{A.3}$$

Herein, $\mathcal{B}$ denotes a minibatch of experiences $\mathcal{E}$:

$$\mathcal{E}[k] = \{\boldsymbol{o}[k], a[k], r[k+1], \tau[k+1], \boldsymbol{o}[k+1]\}, \tag{A.4}$$

which contains the relevant information about the state transition that is learned from. The Boolean flag τ marks the termination of the control task, which nullifies the future value when, e.g., the control task is halted or violation of safety constraints trigger an emergency shutdown. As realized by means of the max$(\cdot)$ operator in the estimation target, this cost function focuses the momentary action value for the assumption of subsequent optimal control, i.e., the controller learns on the basis of the expected best achievable trajectory instead of the actually observed one. This implementation detail enables off-policy training, meaning that transition experiences $\mathcal{E}$ can be considered in any order and from any control policy to optimize for $\boldsymbol{\theta}$. The parameter update is then performed via stochastic gradient descent (or variations thereof, commonly [123])

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \beta_q \nabla_{\boldsymbol{\theta}} J_q(\boldsymbol{\theta}), \tag{A.5}$$

with learning rate β_q.

The utilization of $\hat{q}_{\boldsymbol{\theta}}$ to estimate its own estimation target is labeled a bootstrapping method [86]. In order to reduce the variance of parameter updates, estimator $\hat{q}_{\boldsymbol{\theta}}$ and target $r + \gamma \hat{q}_{\tilde{\boldsymbol{\theta}}}$ are usually not updated at the same rate. Instead, a set of less-frequently or slower-updated target parameters $\tilde{\boldsymbol{\theta}}$ is used to determine the estimation target, commonly by means of a low-pass filter constant $\kappa \in]0, 1[$:

$$\tilde{\boldsymbol{\theta}} \leftarrow (1 - \kappa)\tilde{\boldsymbol{\theta}} + \kappa \boldsymbol{\theta}. \tag{A.6}$$

Lastly, the training benefits from randomness within the selected actions because the optimal action for a given observation can be identified much more easily if it has been tried out before. Since DQN is an off-policy RL algorithm, an arbitrary control policy may be used during training. However, it is established practice to make use of the so-called ϵ-greedy policy, defined by

$$a[k] = \begin{cases} \arg\max\limits_{a' \in \mathcal{A}_{\mathrm{FCS}}} \hat{q}_{\boldsymbol{\theta}}(\boldsymbol{o}[k], a') & \text{during application,} \\ \begin{cases} \arg\max\limits_{a' \in \mathcal{A}_{\mathrm{FCS}}} \hat{q}_{\boldsymbol{\theta}}(\boldsymbol{o}[k], a') & \text{with probability } 1 - \epsilon \\ \mathcal{U}_{\mathcal{A}_{\mathrm{FCS}}} & \text{with probability } \epsilon \end{cases} & \text{during training.} \end{cases} \tag{A.7}$$

Accordingly, the exploration rate ϵ denotes the probability that decides whether a random action is selected, and $\mathcal{U}_{\mathcal{A}_{\mathrm{FCS}}}$ denotes a uniformly distributed random sample from the finite action space $\mathcal{A}_{\mathrm{FCS}}$.

During a training process, it is usually helpful to schedule learning rate β_q and exploration rate ϵ to decrease over time. That way, the weights $\boldsymbol{\theta}$ can be fine-tuned with higher granularity, and the closed-loop behavior converges slowly to (assumed) optimal performance. The latter characteristic will result in more transition experiences $\mathcal{E}$ being collect in regions of the state and action space that are relevant for performance improvement.

Despite the DQN being the most popular FCS-RL algorithm at the time of writing this work, several further methods and extensions have been developed on its basis. Some of these candidate algorithms for FCS-RL are listed in the following for the reader's convenience but without claim of completeness:

- categorical 51-atom deep q learning (C51) [124]
- quantile regression deep q learning (QR-DQN) [125]
- hindsight experience replay (HER) [126]
- several marginal improvements to DQN are cumulated in the so-called 'rainbow' algorithm [127].

A.2.1 Hyperparameter Optimization for the FCS-RL-DTC

In consideration of the multitude of hyperparameters that are introduced by the DQN, a hyperparameter optimization (HPO) is carried out to determine a sensible configuration for the control agent. The corresponding results have originally been published in [A1]. Herein, a multitude of 500 differently parameterized DQN agents is simulatively trained and evaluated by making use of the Paderborn Center for Parallel Computing [128] in conjunction with the HPO software library Hyperopt [129]. As mentioned in Sec. 2.5.2, only stateless feedforward networks are considered.

For evaluation, the reward as of Sec. 3.2 is considered in a normalized fashion to account for the impact of the tunable discount factor γ on the reward range $r \in [-1, (1-\gamma)]$, with the utilized evaluation profile depicted in Fig. A.2. The considered hyperparameters and their search intervals are listed in Tab. A.1, as well as the three best-performing hyperparameter configurations. As these results are rather concentrated, they can be assumed to be quite reliable in terms of achievable performance. The upper performance boundary is also confirmed by the convergence diagram for this HPO, which is presented in Fig. A.3, wherein it can be seen that a variety of HPO samples yielded similar results.

The HPO suggestions concerning the number of neurons cannot be considered for being incompatible with the real-time constraint on the available RCPH. For the same reason, despite touching the upper limit of the search space with respect to the number of network layers, a further increase of the search space is not pragmatic.

The importance value measures in how far the performance outcome depends on proper selection of each hyperparameter. It is herein computed making use of the fANOVA definition of importance [130], utilizing the Optuna framework [131]. Particularly, this importance metric targets to measure dependency instead of (multi)correlation, and is

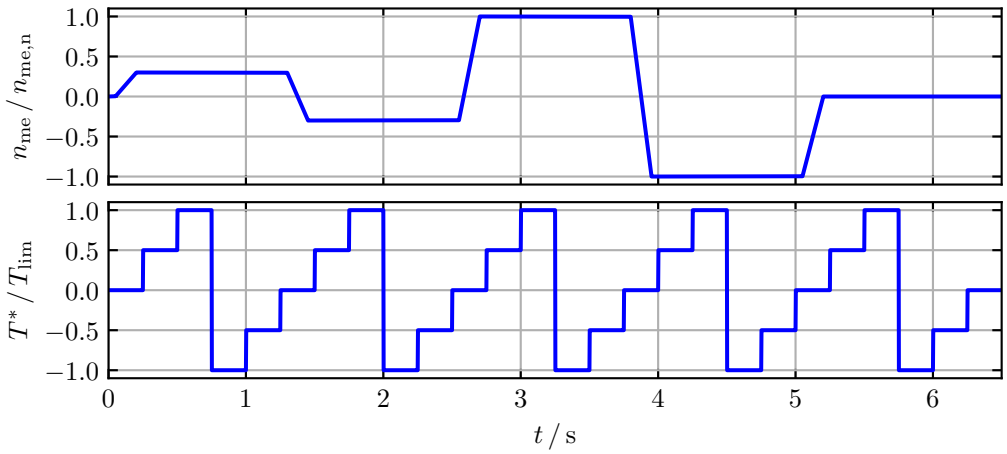


Fig. A.2: Evaluation profile on which the performance of differently parameterized agents is quantized during the HPO. Note that the HPO results were acquired assuming a differently parameterized drive than the one utilized in this work (cf. [A1]). For generality, the evaluation profile is herein specified in a per-unit fashion.

therefore better suited to identify critical tuning quantities. The results of this importance analysis are depicted in Fig. A.5 for the ten most critical hyperparameters. Please note that the relative importance adds up to one when summing over all hyperparameters. This evaluation clearly highlights that successful controller performance is mainly dependent on proper selection of the discount factor γ, whereas all other parameters are significantly less critical for the outcome.

Tab. A.1: HPO search space and results for the FCS-RL-DTC

	Description	Search Space	Best Config.	2nd	3rd
General	Discount factor γ	$[0, 1[$	0.868	0.881	0.849
	Target update parameter κ	$[1, 999] \cdot 10^{-3}$ and $\{1, ..., 500\} \cdot 10^{2}$	$2.096 \cdot 10^{-1}$	$1.767 \cdot 10^{-1}$	$1.425 \cdot 10^{-1}$
	Memory buffer size $\lvert\mathcal{D}\rvert$	$\{5 \cdot 10^{3}, ..., 5 \cdot 10^{5}\}$	$3.650 \cdot 10^{5}$	$3.950 \cdot 10^{5}$	$4.000 \cdot 10^{5}$
	Batch size $\lvert\mathcal{B}\rvert$	$2^{\{3,...,6\}}$	32	64	64
	Episode length	$\{100, ..., 5 \cdot 10^{4}\}$	$1.490 \cdot 10^{4}$	$0.610 \cdot 10^{4}$	$1.920 \cdot 10^{4}$
	Total training steps	$3 \cdot 10^{6}$ (fix)			
DQN $\hat{q}_{\boldsymbol{\theta}}$	Layers	$\{1, ..., 10\}$	10	10	10
	Neurons per layer	$\{20, ..., 1000\}$	560	540	440
	Activation function f_{act}	cf. Tab. A.2	LeakyReLU	LeakyReLU	LeakyReLU
	Activation function parameter α_q	cf. Tab. A.2	$2.908 \cdot 10^{-1}$	$3.425 \cdot 10^{-1}$	$1.785 \cdot 10^{-1}$
	Initial learning rate β_q^{start}	$[10^{-8}, 10^{-4}]$	$2.887 \cdot 10^{-5}$	$3.170 \cdot 10^{-5}$	$3.477 \cdot 10^{-5}$
	Final learning rate β_q^{end}	$[10^{-8}, 10^{-4}]$	$1.736 \cdot 10^{-5}$	$8.756 \cdot 10^{-5}$	$8.602 \cdot 10^{-5}$
	Learning rate reduction start $k_{\beta_q}^{\text{start}}$	$\{0, ..., 10^{6}\}$	$4.600 \cdot 10^{5}$	$3.600 \cdot 10^{5}$	$9.950 \cdot 10^{5}$
	Learning rate reduction interval $k_{\beta_q}^{\text{int}}$	$\{5 \cdot 10^{4}, ..., 3 \cdot 10^{6}\}$	$2.710 \cdot 10^{6}$	$1.995 \cdot 10^{6}$	$2.100 \cdot 10^{6}$
	Initial exploration rate ϵ^{start}	$[0, 0.5]$	$2.119 \cdot 10^{-1}$	$1.797 \cdot 10^{-1}$	$2.553 \cdot 10^{-1}$
	Final exploration rate ϵ^{end}	$[0, 0.2]$	$1.774 \cdot 10^{-1}$	$1.111 \cdot 10^{-1}$	$1.859 \cdot 10^{-1}$
	Exploration rate reduction start $k_{\epsilon}^{\text{start}}$	0 (fix)			
	Exploration rate reduction interval $k_{\epsilon}^{\text{int}}$	$\{5 \cdot 10^{4}, ..., 3 \cdot 10^{6}\}$	$2.210 \cdot 10^{6}$	$2.160 \cdot 10^{6}$	$2.445 \cdot 10^{6}$
	Achieved reward	$\frac{1}{K} \sum_k \frac{r_k}{1-\gamma}$	67.51 %	67.41 %	67.40 %
	Torque control MSE	$\frac{1}{K} \sum_k \left(\frac{T_k^* - T_k}{2T_{\text{lim}}^*} \right)^2$	0.47 %	0.36 %	0.43 %
	Stator current RMS	$\sqrt{\frac{1}{K} \sum_k \left(\frac{i_{\text{s},k}}{i_{\text{lim}}} \right)^2}$	61.62 %	62.67 %	61.98 %

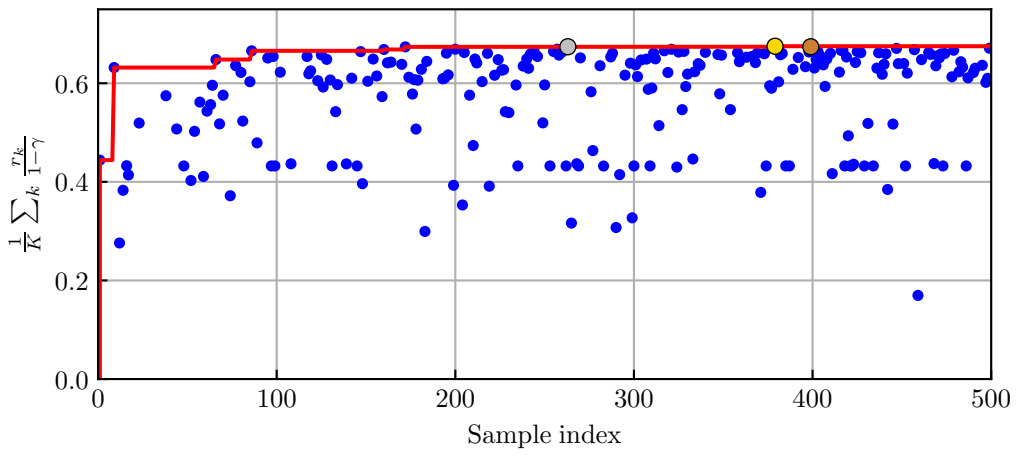
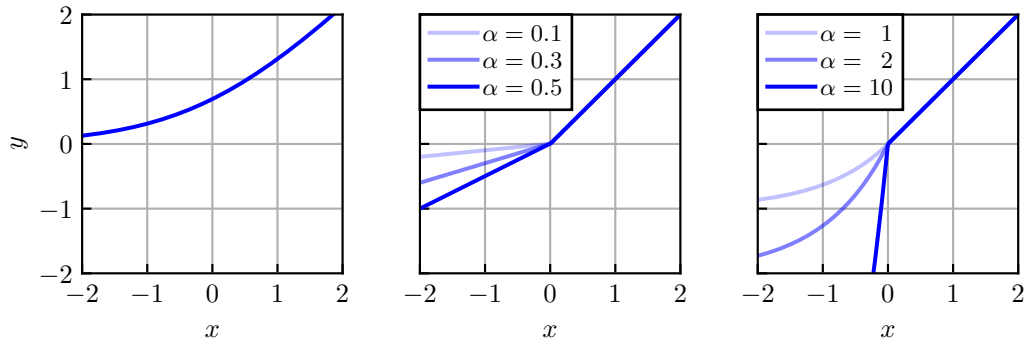


Fig. A.3: Convergence diagram of the HPO for the FCS-RL-DTC
From originally 500 differently parameterized control agents, only the 267 agents that respected the plant's constraints during the evaluation episode are depicted.

Tab. A.2: Search space of the DQN activation functions [96]

Name	Definition	Parameters
SoftPlus	$y = \ln(1 + e^x)$	
LeakyReLU	$y = \max(\alpha x, x)$	$\alpha \in [0, 0.5]$
ELU	$y = \begin{cases} \alpha(e^x - 1) & \text{if } x \leq 0 \\ x & \text{if } x > 0 \end{cases}$	$\alpha \in [0, 10]$
SELU	$y = \lambda \begin{cases} \alpha(e^x - 1) & \text{if } x \leq 0 \\ x & \text{if } x > 0 \end{cases}$	$\lambda = 1.051$ $\alpha = 1.673$
Sigmoid	$y = (1 + e^{-x})^{-1}$	
Tanh	$y = 1 - 2(e^{2x} - 1)^{-1}$	



(a) SoftPlus **(b)** LeakyReLU **(c)** ELU

(d) SELU **(e)** Sigmoid **(f)** Tanh

Fig. A.4: Considered activation functions as of Tab. A.2

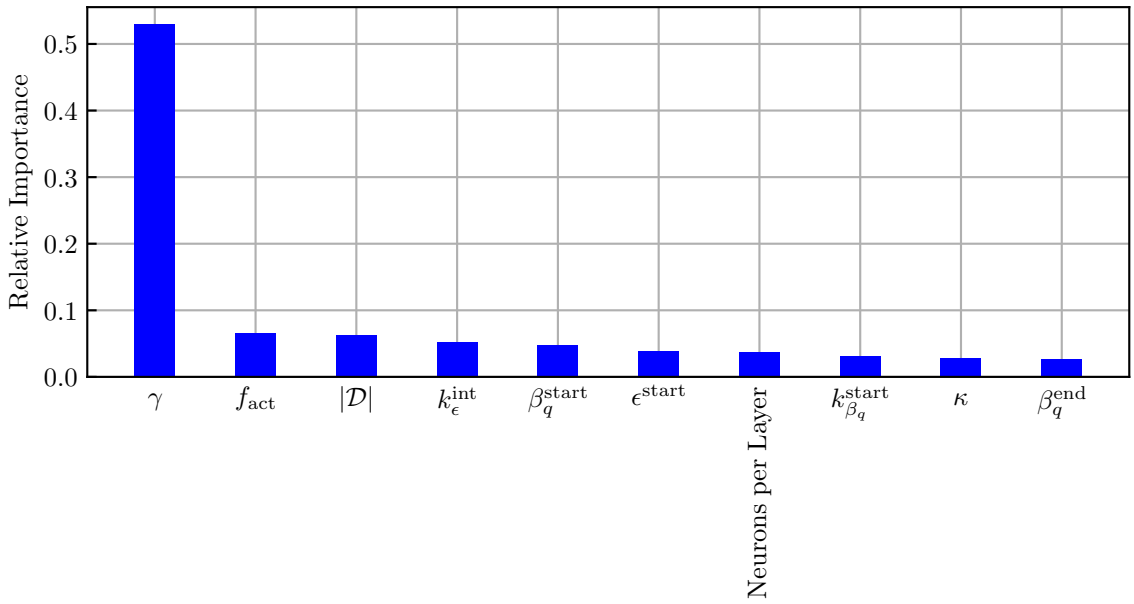


Fig. A.5: Relative importance of the ten most important parameters that were considered in the FCS HPO (cf. Tab. A.1); these cover an accumulated relative importance of 91.62 %

A.3 Reinforcement Learning in the CCS: Deep Deterministic Policy Gradient

Whereas the DQN provides a solid framework for learning the action-value function $\hat{q}_{\boldsymbol{\theta}}$, it is yet insufficient for CCS control because an arg max operation cannot be explicitly resolved on a CCS as it is the case for the FCS. Implicit optimization with the use of (necessarily nonlinear) solvers is possible, but can hardly be realized in a real-time capable fashion for sampling frequencies of 5 kHz and beyond, that are common in power electronic systems [73, 74]. Hence, the deep deterministic policy gradient (DDPG) algorithm is utilized to deal with corresponding CCS environments [132, 133].

Basically, the DDPG builds upon the action-value estimation $\hat{q}_{\boldsymbol{\theta}}$ established by means of the DQN and extends it by a policy approximator $\boldsymbol{\pi}_{\boldsymbol{\zeta}}$ that is characterized by network weights $\boldsymbol{\zeta}$. The arg max operation is condensed into this policy approximator:

$$\boldsymbol{\pi}_{\boldsymbol{\zeta}}(\boldsymbol{o}[k]) = \underset{\boldsymbol{u}' \in \mathcal{A}_{\mathrm{CCS}}}{\arg\max}\, \hat{q}_{\boldsymbol{\theta}}(\boldsymbol{o}[k], \boldsymbol{u}'). \tag{A.8}$$

The separation of tasks into action selection through $\boldsymbol{\pi}_{\boldsymbol{\zeta}}$ and action evaluation through $\hat{q}_{\boldsymbol{\theta}}$ has acquired corresponding approaches the name actor critic algorithms. Exploiting the differentiability of $\hat{q}_{\boldsymbol{\theta}}$ with concern to its action input, the cost function for the training of $\boldsymbol{\pi}_{\boldsymbol{\zeta}}$ can be formulated as

$$J_{\boldsymbol{\pi}}(\boldsymbol{\zeta}) = -\frac{1}{|\mathcal{B}|} \sum_{\mathcal{E} \in \mathcal{B}} \hat{q}_{\boldsymbol{\theta}}(\boldsymbol{o}[k], \boldsymbol{\pi}_{\boldsymbol{\zeta}}(\boldsymbol{o}[k])), \tag{A.9}$$

such that minimization of $J_{\boldsymbol{\pi}}$ via gradient descent leads to the update rule

$$\boldsymbol{\zeta} \leftarrow \boldsymbol{\zeta} - \beta_{\boldsymbol{\pi}} \nabla_{\boldsymbol{\zeta}} J_{\boldsymbol{\pi}}(\boldsymbol{\zeta}), \tag{A.10}$$

which corresponds to maximization of $\hat{q}_{\boldsymbol{\theta}}$, as demanded in (A.8). Note that the policy approximator $\boldsymbol{\pi}_{\boldsymbol{\zeta}}$ must also be utilized to translate the q learning cost function (A.3) to the CCS:

$$\begin{aligned} J_q(\boldsymbol{\theta}) = \frac{1}{|\mathcal{B}|} \sum_{\mathcal{E} \in \mathcal{B}} \Big(& \hat{q}_{\boldsymbol{\theta}}(\boldsymbol{o}[k], \boldsymbol{u}[k]) - \\ & \Big(r[k+1] + \gamma (1 - \tau[k+1]) \hat{q}_{\tilde{\boldsymbol{\theta}}}(\boldsymbol{o}[k+1], \boldsymbol{\pi}_{\tilde{\boldsymbol{\zeta}}}(\boldsymbol{o}[k+1])) \Big) \Big)^2. \end{aligned} \tag{A.11}$$

Particularly, the estimation target is herein computed using target parameters $\tilde{\boldsymbol{\theta}}, \tilde{\boldsymbol{\zeta}}$ for both, $\hat{q}_{\tilde{\boldsymbol{\theta}}}$ and $\boldsymbol{\pi}_{\tilde{\boldsymbol{\zeta}}}$, which are updated in the same way as in the FCS case:

$$\tilde{\boldsymbol{\theta}} \leftarrow (1 - \kappa) \tilde{\boldsymbol{\theta}} + \kappa \boldsymbol{\theta}, \qquad \tilde{\boldsymbol{\zeta}} \leftarrow (1 - \kappa) \tilde{\boldsymbol{\zeta}} + \kappa \boldsymbol{\zeta}. \tag{A.12}$$

Finally, also the exploration noise needs adjustment to match the CCS. Herein, the policy action $\boldsymbol{\pi}_{\boldsymbol{\zeta}}$ is superimposed with a noise signal $\boldsymbol{n}$:

$$\boldsymbol{u}[k] = \begin{cases} \boldsymbol{\pi_\zeta}(\boldsymbol{o}[k]) & \text{during application,} \\ \boldsymbol{\pi_\zeta}(\boldsymbol{o}[k]) + \boldsymbol{n}[k] & \text{during training.} \end{cases} \tag{A.13}$$

While $\boldsymbol{n}$ could be generated arbitrarily, it is common practice to make use of an Ornstein-Uhlenbeck process [134]:

$$\boldsymbol{n}[k+1] = (1 - m\frac{T_\mathrm{s}}{1\,\mathrm{s}})\boldsymbol{n}[k] + \sigma\sqrt{\frac{T_\mathrm{s}}{1\,\mathrm{s}}}\mathcal{N}_{(\boldsymbol{0},\boldsymbol{I})}, \tag{A.14}$$

with $\mathcal{N}_{(\boldsymbol{0},\boldsymbol{I})}$ denoting a random sample from a multivariate Gaussian distribution with mean $\boldsymbol{0}$ and covariance $\boldsymbol{I}$. Ornstein-Uhlenbeck noise is much more likely to excite the plant system at frequencies below T_s^{-1} than stateless noise generators such as Gaussian noise or uniform noise. It has therefore been found to be a sensible design choice in the context of systems with inertia [133].

While the DDPG could be labeled the most established CCS-RL algorithm at the time of writing this work, several extensions to it have produced a variety of algorithms that come with different advantages and disadvantages each. As a discussion of their characteristics is not inside the scope of this thesis, some CCS-RL candidate algorithms are to be listed in the following for the reader's convenience but without claiming completeness:

- trust region policy optimization (TRPO) [135]
- proximal policy optimization (PPO) [136]
- twin delayed deep deterministic policy gradient (TD3) [137]
- soft actor critic (SAC) [138].

A.3.1 Hyperparameter Optimization for the CCS-RL-DTC

The HPO results for the CCS case have originally been published in [A3], for which they have been acquired through the efforts of Barnabas Haucke-Korber.

For the CCS HPO, some conclusions from the FCS HPO are transferred directly to pragmatically limit the search space. Primarily, the LeakyReLU activation function is now fixed for both, actor network $\boldsymbol{\pi_\zeta}$ and critic network $\hat{q}_{\boldsymbol{\theta}}$, and the actor architecture search space is limited to real-time capable ANN complexity. Again, the Paderborn Center for Parallel Computing [128] is utilized for the simulative HPO, which in the CCS case spans a total of 7553 agents evaluated via the Optuna framework [131]. Again, the agents are subjected to the evaluation profile depicted in Fig. A.2 to quantize their performance. The search spaces and best three parameter configurations are listed in Tab. A.3, and the corresponding convergence diagram is depicted in Fig. A.6.

Strikingly, the results exhibit that the actor-critic structure favors critic networks $\hat{q}_{\boldsymbol{\theta}}$ with higher complexity while the actor networks $\boldsymbol{\pi_\zeta}$ remain rather simple. This characteristic is preferable for the transfer to real-world systems, as only the actor is subjected to real-time constraints. In comparison to the FCS case, however, the DDPG critic is

Tab. A.3: HPO search space and results for the CCS-RL-DTC

	Description	Search Space	Best Config.	2nd	3rd
General	Discount factor γ	$[0, 1[$	$8.423 \cdot 10^{-1}$	$8.927 \cdot 10^{-1}$	$8.567 \cdot 10^{-1}$
	Target update parameter κ	$[1, 999] \cdot 10^{-3}$	$3.471 \cdot 10^{-1}$	$5.948 \cdot 10^{-1}$	$1.946 \cdot 10^{-1}$
	Memory buffer size $\lvert\mathcal{D}\rvert$	$\{5 \cdot 10^3, \dots, 5 \cdot 10^5\}$	$4.081 \cdot 10^5$	$4.517 \cdot 10^5$	$3.929 \cdot 10^5$
	Batch size $\lvert\mathcal{B}\rvert$	$2^{\{0,\dots,8\}}$	128	128	128
	Episode length	$\{10^2, \dots, 5 \cdot 10^4\}$	$3.668 \cdot 10^3$	$2.242 \cdot 10^4$	$4.907 \cdot 10^3$
	Total training steps	$3 \cdot 10^6$ (fix)			
Critic $\hat{q}_{\boldsymbol{\theta}}$	Layers	$\{1, \dots, 10\}$	6	3	3
	Neurons per layer	$\{1, \dots, 10^3\}$	186	200	515
	Hidden layer activation function	LeakyReLU (fix)			
	Activation function parameter α_q	$[0, 0.5]$	$4.067 \cdot 10^{-1}$	$1.438 \cdot 10^{-1}$	$2.767 \cdot 10^{-1}$
	Initial learning rate β_q^{start}	$[10^{-8}, 10^{-2}]$	$1.731 \cdot 10^{-4}$	$1.039 \cdot 10^{-4}$	$2.120 \cdot 10^{-4}$
	Final learning rate β_q^{end}	$[0, 1] \cdot \beta_q^{\text{start}}$	$1.021 \cdot 10^{-4}$	$8.729 \cdot 10^{-5}$	$2.458 \cdot 10^{-5}$
	Learning rate reduction start $k_{\beta_q}^{\text{start}}$	$\{0, \dots, 10^6\}$	$4.181 \cdot 10^3$	$4.089 \cdot 10^5$	$6.090 \cdot 10^5$
	Learning rate reduction interval $k_{\beta_q}^{\text{int}}$	$\{5 \cdot 10^3, \dots, 3 \cdot 10^6\}$	$5.305 \cdot 10^5$	$2.094 \cdot 10^5$	$2.994 \cdot 10^6$
Actor $\boldsymbol{\pi}_{\boldsymbol{\zeta}}$	Layers	$\{1, \dots, 10\}$	6	9	9
	Neurons per Layer	$\{1, \dots, 90\}$	20	42	39
	Hidden layer activation function	LeakyReLU (fix)			
	Activation function parameter $\alpha_{\boldsymbol{\pi}}$	$[0, 0.5]$	$2.794 \cdot 10^{-1}$	$4.258 \cdot 10^{-1}$	$4.322 \cdot 10^{-1}$
	Initial learning rate $\beta_{\boldsymbol{\pi}}^{\text{start}}$	$[10^{-8}, 10^{-2}]$	$3.358 \cdot 10^{-3}$	$1.808 \cdot 10^{-3}$	$2.466 \cdot 10^{-3}$
	Final learning rate $\beta_{\boldsymbol{\pi}}^{\text{end}}$	$[0, 1] \cdot \beta_{\boldsymbol{\pi}}^{\text{start}}$	$3.982 \cdot 10^{-4}$	$3.541 \cdot 10^{-4}$	$8.544 \cdot 10^{-5}$
	Learning rate reduction start $k_{\beta_{\boldsymbol{\pi}}}^{\text{start}}$	$\{0, \dots, 10^6\}$	$3.723 \cdot 10^5$	$3.925 \cdot 10^5$	$7.241 \cdot 10^5$
	Learning rate reduction interval $k_{\beta_{\boldsymbol{\pi}}}^{\text{int}}$	$\{5 \cdot 10^3, \dots, 3 \cdot 10^6\}$	$6.148 \cdot 10^5$	$1.597 \cdot 10^6$	$1.114 \cdot 10^6$
	OU mean reversion rate m	$[10^{-10}, 2 \cdot 10^2]$	$1.579 \cdot 10^2$	$8.995 \cdot 10^1$	$4.842 \cdot 10^1$
	Initial OU diffusion coefficient σ^{start}	$[10^{-4}, 1]$	$4.941 \cdot 10^{-1}$	$9.654 \cdot 10^{-1}$	$6.879 \cdot 10^{-1}$
	Final OU diffusion coefficient σ^{end}	0 (fix)			
	OU diffusion coefficient reduction interval k_{σ}^{int}	$\{10^3, \dots, 3 \cdot 10^6\}$	$1.074 \cdot 10^6$	$4.613 \cdot 10^5$	$2.522 \cdot 10^6$
	Achieved reward	$\frac{1}{K} \sum_k \frac{r_k}{1-\gamma}$	68.926 %	68.839 %	68.788 %
	Torque control MSE	$\frac{1}{K} \sum_k \left(\frac{T_k^* - T_k}{2T_{\text{lim}}^*} \right)^2$	0.433 %	0.479 %	0.448 %
	Stator current RMS	$\sqrt{\frac{1}{K} \sum_k \left(\frac{i_{\text{s},k}}{i_{\text{lim}}} \right)^2}$	64.025 %	62.728 %	62.962 %

notably smaller than the DQN. This could be explained by the difference in coordinate interpretation between the respective action spaces: the DDPG agent is fully interfaced by quantities represented in the dq reference frame, wherein no further relation between angle ε_{el} and dynamic system behavior should be present. Contrary, in the FCS case, this transformation is not applicable to the action selection and, hence, the DQN needs to incorporate the mapping between rotor-fixed interpretation of electrical measurements and stator-fixed interpretation of switching states. Accordingly, the DQN agent must incorporate a dependency between ε_{el} and plant behavior, requiring a more comprehensive estimation architecture.

In terms of hyperparameter importance, Fig. A.7 indicates that the complexity of critic architecture is crucial for application, wherein again the fANOVA importance concept [130] was in use. Further, the learning rates β are of considerable priority. Interestingly, the discount factor γ plays much less of a role in the CCS case, which could be explained

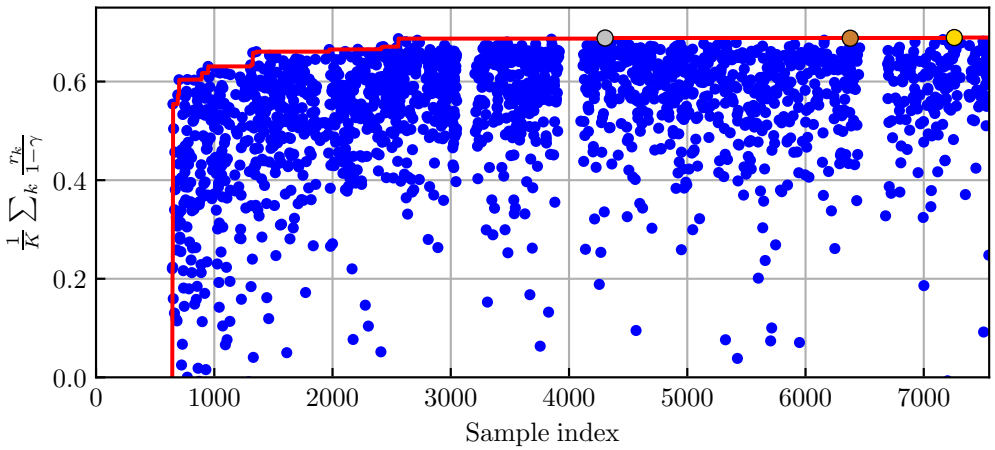


Fig. A.6: Convergence diagram of the HPO for the CCS-RL-DTC
From originally 7553 differently parameterized control agents, only the 5710 agents that respected the plant's constraints during the evaluation episode are depicted.

by the different impact of action selection. Contrary to the FCS case, the CCS case also allows to alter the voltage only slightly. The difference in outcome is in such cases much less striking than it is between switching states, allowing to correct suboptimal voltage commands also on a short-sighted horizon.

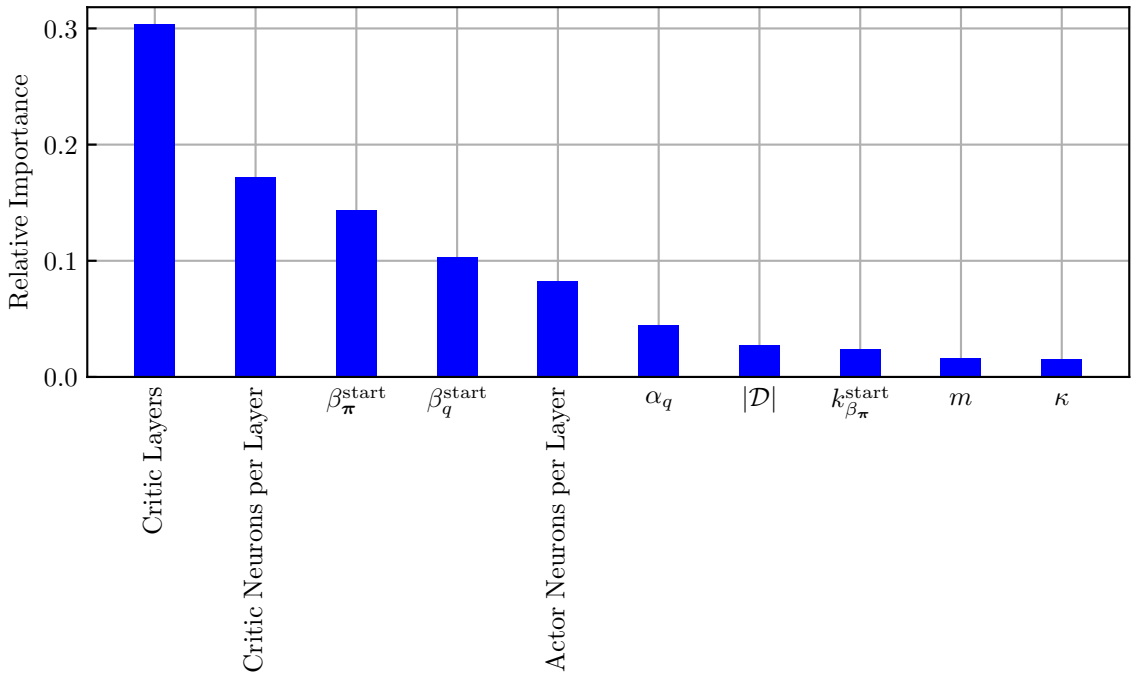


Fig. A.7: Relative importance of the ten most important parameters that were considered in the CCS HPO (cf. Tab. A.3); these cover an accumulated relative importance of 92.90 %

A.4 Linear Approximation of Elliptic Sets in $\mathbb{R}^2$

As real-time capable optimization solvers are usually designed to handle linear constraints only, the consideration of elliptic constraints is not directly possible. Instead, linear approximation of elliptic constraints is targeted in this contribution. While the approximation principle follows purely geometric considerations and can be inferred quite simply from corresponding depictions (cf. Fig. A.8), the underlying algorithm is to be documented for completeness. Please note that only the two-dimensional case is addressed for the given application, i.e., only the case $\boldsymbol{x} = \begin{bmatrix} x_\mathrm{d} & x_\mathrm{q} \end{bmatrix}^\top$ is to be covered in the following. The full procedure is concisely documented in Alg. A.1.

The general elliptic constraint is defined by

$$\begin{aligned} & ||\boldsymbol{w} + \boldsymbol{W}\boldsymbol{x}||_2 \leq x_\mathrm{lim}, \\ \Leftrightarrow \quad & \sqrt{(\boldsymbol{w} + \boldsymbol{W}\boldsymbol{x})^\top (\boldsymbol{w} + \boldsymbol{W}\boldsymbol{x})} \leq x_\mathrm{lim}, \\ \Leftrightarrow \quad & \boldsymbol{w}^\top \boldsymbol{w} + 2\boldsymbol{w}^\top \boldsymbol{W}\boldsymbol{x} + \boldsymbol{x}^\top \boldsymbol{W}^\top \boldsymbol{W}\boldsymbol{x} \leq x_\mathrm{lim}^2. \end{aligned} \tag{A.15}$$

The introduction of several abbreviations allow to streamline the further notation:

$$\begin{aligned}
\mathbb{V} &= \{\boldsymbol{x} \in \mathbb{R}^2 | \, ||\boldsymbol{w} + \boldsymbol{W}\boldsymbol{x}||_2 \leq x_{\text{lim}}\}, \\
\partial\mathbb{V} &= \{\boldsymbol{x} \in \mathbb{R}^2 | \, ||\boldsymbol{w} + \boldsymbol{W}\boldsymbol{x}||_2 = x_{\text{lim}}\}, \\
l &= \boldsymbol{w}^\top \boldsymbol{w} - x_{\text{lim}}^2, \\
\boldsymbol{m}^\top &= \begin{bmatrix} m_1 & m_2 \end{bmatrix} = 2\boldsymbol{w}^\top \boldsymbol{W}, \\
\boldsymbol{N} &= \begin{bmatrix} n_{11} & n_{12} \\ n_{21} & n_{22} \end{bmatrix} = \boldsymbol{W}^\top \boldsymbol{W}, \quad \text{with } n_{12} = n_{21}.
\end{aligned} \tag{A.16}$$

To formulate a constraint, mainly the edge $\partial\mathbb{V}$ of the ellipsis $\mathbb{V}$ is of interest. The geometric characteristics of $\partial\mathbb{V}$ can be extracted via

$$\begin{aligned}
a, b &= \frac{-\sqrt{2}}{4(n_{12}^2 - n_{11}n_{22})} \sqrt{n_{11}m_2^2 + n_{22}m_1^2 - 2n_{12}m_1m_2 + 4l(n_{12}^2 - n_{11}n_{22})} \\
&\quad \cdot \sqrt{n_{11} + n_{22} \pm \sqrt{(n_{11} - n_{22})^2 + 4n_{12}^2}}, \\
\boldsymbol{c} &= \frac{1}{2(n_{12}^2 - n_{11}n_{22})} \begin{bmatrix} n_{22}m_1 - n_{12}m_2 \\ n_{11}m_2 - n_{12}m_1 \end{bmatrix}, \\
\varepsilon_0 &= \begin{cases} \arctan\left(\dfrac{-2n_{12}}{n_{22} - n_{11} + \sqrt{(n_{22} - n_{11})^2 + 4n_{12}^2}}\right) & \text{if } n_{12} \neq 0, \\ 0 & \text{if } n_{12} = 0 \text{ and } n_{11} \leq n_{22}, \\ \dfrac{\pi}{2} & \text{if } n_{12} = 0 \text{ and } n_{11} > n_{22}, \end{cases}
\end{aligned} \tag{A.17}$$

with a being the major semi-axis, b being the minor semi-axis and $\boldsymbol{c}$ being the ellipsis' center point [139]. The angle ε_0 denotes the rotation of the major semi-axis with respect to the coordinate system [140]. The topological interpretation of these quantities is further motivated in Fig. A.8.

For the linearization procedure, it is of interest to determine a set of points on $\partial\mathbb{V}$ with uniform angular separation. This can be realized with $a, b, \boldsymbol{c}$ and ε_0 being already known:

$$\partial\mathbb{V} = \left\{ \boldsymbol{v} \in \mathbb{R}^2 \,\middle|\, \boldsymbol{v} = \boldsymbol{c} + \boldsymbol{Q}_{\alpha\beta,\text{dq}}(\varepsilon_0) \begin{bmatrix} a\cos(\varepsilon) \\ b\sin(\varepsilon) \end{bmatrix} \wedge \varepsilon \in [0, 2\pi] \right\}. \tag{A.18}$$

A set of angularly equidistant vertices can be identified on the basis of (A.18). As a last step, neighboring pairs of these vertices have to be reinterpreted to formulate a new set of linear constraints $\hat{\mathbb{V}}$ that approximates the original $\mathbb{V}$. As each pair of vertices will result in one linear constraint, the quality of approximation of the original ellipsis can be increased by increasing the number of vertices R. In the context of online optimization, this has to be configured cautiously, because each added line of constraints will increase the calculation effort at runtime.

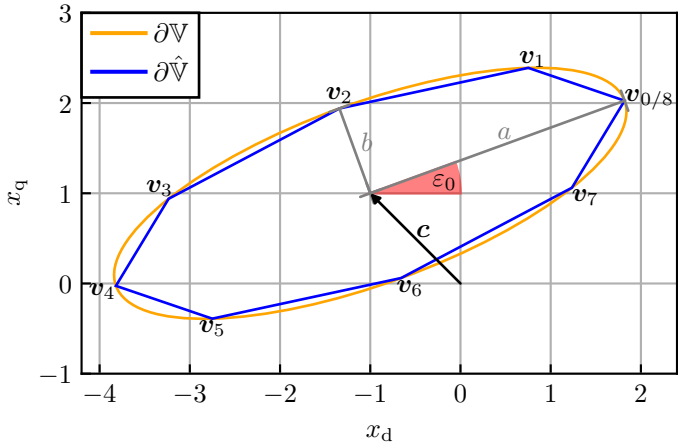


Fig. A.8: Exemplary ellipsis $\mathbb{V}$ with parameters $\boldsymbol{c} = \begin{bmatrix}-1 & 1\end{bmatrix}^\top$, $a = 3$, $b = 1$ and $\varepsilon_0 = \frac{\pi}{6}$, and corresponding linear approximation $\hat{\mathbb{V}}$ with vertices $\boldsymbol{v}_{0,\dots,8}$ and approximation resolution $R = 8$

Assuming two neighboring vertices $\boldsymbol{v}_r$ and $\boldsymbol{v}_{r+1}$ that are arranged anti-clockwise around $\boldsymbol{c}$ (cf. Fig. A.8), the corresponding constraint can be computed as follows:

$$\begin{aligned} &\underbrace{\left(v_{\mathrm{q},r+1} - v_{\mathrm{q},r}\right)}_{g_{\mathrm{d},r}} x_\mathrm{d} + \underbrace{\left(v_{\mathrm{d},r} - v_{\mathrm{d},r+1}\right)}_{g_{\mathrm{q},r}} x_\mathrm{q} \leq \underbrace{v_{\mathrm{q},r+1} v_{\mathrm{d},r} - v_{\mathrm{d},r+1} v_{\mathrm{q},r}}_{h_r} \\ \Leftrightarrow \quad &\boldsymbol{g}_r^\top \boldsymbol{x} \leq h_r. \end{aligned} \tag{A.19}$$

Finally, the constraint for $\hat{\mathbb{V}}$ is compounded from a set of R linear constraints:

$$\hat{\mathbb{V}} = \left\{ \boldsymbol{x} \in \mathbb{R}^2 \;\middle|\; \underbrace{\begin{bmatrix} \boldsymbol{g}_0^\top \\ \boldsymbol{g}_1^\top \\ \vdots \\ \boldsymbol{g}_{R-1}^\top \end{bmatrix}}_{\boldsymbol{G}} \boldsymbol{x} \leq \underbrace{\begin{bmatrix} h_0 \\ h_1 \\ \vdots \\ h_{R-1} \end{bmatrix}}_{\boldsymbol{h}} \right\}. \tag{A.20}$$

Alg. A.1 Linearization of Elliptic Constraints in $\mathbb{R}^2$

input: Parameters $\boldsymbol{w}$, $\boldsymbol{W}$ and x_{lim} of an elliptic constraint $||\boldsymbol{w} + \boldsymbol{W}\boldsymbol{x}||_2 \leq x_{\text{lim}}$
input: Approximation resolution R
output: Parameters $\boldsymbol{G}$ and $\boldsymbol{h}$ of a linear constraint $\boldsymbol{G}\boldsymbol{x} \leq \boldsymbol{h}$
Compute l, $\boldsymbol{m}^\top$ and $\boldsymbol{N}$ from $\boldsymbol{w}$, $\boldsymbol{W}$ and x_{lim} according to (A.16)
Compute a, b, $\boldsymbol{c}$ and ε_0 from l, $\boldsymbol{m}^\top$ and $\boldsymbol{N}$ according to (A.17)
$r \leftarrow 0$
while $r \leq R$ **do** ▷ Compute vertices $\boldsymbol{v}_r$ of the approximate polygon $\partial\hat{\mathbb{V}}$
  $\varepsilon \leftarrow 2\pi\frac{r}{R}$
  $\boldsymbol{v}_r \leftarrow \boldsymbol{c} + \boldsymbol{Q}_{\alpha\beta,\text{dq}}(\varepsilon_0)\begin{bmatrix} a\cos(\varepsilon) & b\sin(\varepsilon)\end{bmatrix}^\top$
  $r \leftarrow r + 1$
end while
$\boldsymbol{G} \leftarrow [\,]$
$\boldsymbol{h} \leftarrow [\,]$
$r \leftarrow 0$
while $r < R$ **do** ▷ Formulate a constraint $\boldsymbol{g}_r^\top\boldsymbol{x} \leq h_r$ for each pair of vertices $\boldsymbol{v}_r$, $\boldsymbol{v}_{r+1}$
  Compute $\boldsymbol{g}_r^\top$ and h_r from $\boldsymbol{v}_r$ and $\boldsymbol{v}_{r+1}$ according to (A.19)
  $\boldsymbol{G} \leftarrow \begin{bmatrix}\boldsymbol{G}^\top & \boldsymbol{g}_r\end{bmatrix}^\top$
  $\boldsymbol{h} \leftarrow \begin{bmatrix}\boldsymbol{h}^\top & h_r\end{bmatrix}^\top$
  $r \leftarrow r + 1$
end while

List of Figures

List of Tables

References

[1] R. Gabriel, W. Leonhard, and C. J. Nordby. "Field-Oriented Control of a Standard AC Motor Using Microprocessors". In: *IEEE Transactions on Industry Applications* IA-16.2 (1980), pp. 186–192. DOI: 10.1109/TIA.1980.4503770.

[2] P. Karamanakos, E. Liegmann, T. Geyer, and R. Kennel. "Model Predictive Control of Power Electronic Systems: Methods, Results, and Challenges". In: *IEEE Open Journal of Ind. Appl.* 1 (2020), pp. 95–114.

[3] A. Brosch, O. Wallscheid, and J. Böcker. "Torque and Inductances Estimation for Finite Model Predictive Control of Highly Utilized Permanent Magnet Synchronous Motors". In: *IEEE Transactions on Industrial Informatics* 17.12 (2021), pp. 8080–8091. DOI: 10.1109/TII.2021.3060469.

[4] VDE Association for Electrical, Electronic & Information Technologies. *Fewer and Fewer Students are Studying Electrical Engineering: A Series of Image Studies Identifies Reasons and Possible Solutions*. 2023. URL: https://www.vde.com/en/press/press-releases/2023-03-16-image-elektrotechnik.

[5] ABB. *Industrial drive, ACS6080*. Accessed: 2025-07-10. URL: https://new.abb.com/drives/medium-voltage-ac-drives/acs6080.

[6] D. Silver, A. Huang, C. J. Maddison, et al. "Mastering the Game of Go with Deep Neural Networks and Tree Search". In: *Nature* 529 (2016), pp. 484–503. URL: http://www.nature.com/nature/journal/v529/n7587/full/nature16961.html.

[7] D. Silver, T. Hubert, J. Schrittwieser, et al. "A General Reinforcement Learning Algorithm that Masters Chess, Shogi, and Go through Self-Play". In: *Science* 362.6419 (2018), pp. 1140–1144. DOI: 10.1126/science.aar6404. eprint: https://www.science.org/doi/pdf/10.1126/science.aar6404.

[8] J. Degrave, F. Felici, J. Buchli, et al. "Magnetic Control of Tokamak Plasmas through Deep Reinforcement Learning". In: *Nature* 602.7897 (2022), pp. 414–419.

[9] OpenAI, J. Achiam, S. Adler, et al. *GPT-4 Technical Report*. 2024. arXiv: 2303.08774 [cs.CL]. URL: https://arxiv.org/abs/2303.08774.

[10] DeepSeek-AI, A. Liu, B. Feng, et al. *DeepSeek-V2: A Strong, Economical, and Efficient Mixture-of-Experts Language Model*. 2024. arXiv: 2405.04434 [cs.CL]. URL: https://arxiv.org/abs/2405.04434.

[11] N. G. Brigham, C. Gao, T. Kohno, F. Roesner, and N. Mireshghallah. *Developing Story: Case Studies of Generative AI's Use in Journalism*. 2024. arXiv: 2406.13706 [cs.CL]. URL: https://arxiv.org/abs/2406.13706.

[12] E. Kasneci, K. Sessler, S. Küchemann, et al. "ChatGPT for good? On opportunities and challenges of large language models for education". In: *Learning and*

Individual Differences 103 (2023), p. 102274. DOI: https://doi.org/10.1016/j.lindif.2023.102274.

[13] Y. Zhu, H. Yuan, S. Wang, et al. *Large Language Models for Information Retrieval: a Survey*. 2024. arXiv: 2308.07107 [cs.CL]. URL: https://arxiv.org/abs/2308.07107.

[14] M. S. Lundstrom and M. A. Alam. "Moore's Law: The Journey Ahead". In: *Science* 378.6621 (2022), pp. 722–723. DOI: 10.1126/science.ade2191. eprint: https://www.science.org/doi/pdf/10.1126/science.ade2191.

[15] Z. Tang, C. Ma, J. Rodriguez, C. Garcia, and W. Song. "Data-Driven Finite-Set Predictive Current Control via Deep Q-Learning for Permanent Magnet Synchronous Motor Drives". In: *2023 IEEE International Conference on Predictive Control of Electrical Drives and Power Electronics (PRECEDE)*. 2023. DOI: 10.1109/PRECEDE57319.2023.10174508.

[16] Vikas, P. Yadav, B. Singh, and R. Kumar. "Model Free Reinforcement Learning based Control of Permanent Magnet Synchronous Motor Drive". In: *2023 International Conference on Computer, Electronics & Electrical Engineering & their Applications (IC2E3)*. 2023. DOI: 10.1109/IC2E357697.2023.10262459.

[17] N. Farah, G. Lei, J. Zhu, and Y. Guo. "Robust Model-Free Reinforcement Learning based Current Control of PMSM Drives". In: *IEEE Transactions on Transportation Electrification* 11.1 (2025), pp. 1061–1076. DOI: 10.1109/TTE.2024.3400534.

[18] T. Schindler, L. Broghammer, P. Karamanakos, A. Dietz, and R. Kennel. "Deep Reinforcement Learning Current Control of Permanent Magnet Synchronous Machines". In: *IEEE International Electric Machines & Drives Conference (IEMDC)*. 2023. DOI: 10.1109/IEMDC55163.2023.10238988.

[19] L. Broghammer, D. Hufnagel, T. Schindler, et al. "Reinforcement Learning Control of Six-Phase Permanent Magnet Synchronous Machines". In: *International Electric Drives Production Conference (EDPC)*. 2023. DOI: 10.1109/EDPC60603.2023.10372153.

[20] T. Schindler, L. Broghammer, D. Hufnagel, et al. "Steady-State Error Reduction of Reinforcement Learning based Indirect Current Control of Permanent Magnet Synchronous Machines". In: *PCIM Europe 2024; International Exhibition and Conference for Power Electronics, Intelligent Motion, Renewable Energy and Energy Management*. 2024, pp. 140–149. DOI: 10.30420/566262017.

[21] S. Bhattacharjee, S. Halder, A. Balamurali, M. Towhidi, L. V. Iyer, and N. C. Kar. "An Advanced Policy Gradient based Vector Control of PMSM for EV Application". In: *2020 10th International Electric Drives Production Conference (EDPC)*. 2020. DOI: 10.1109/EDPC51184.2020.9388187.

[22] F. Yin, X. Yuan, Z. Ma, and X. Xu. "Vector Control of PMSM using TD3 Reinforcement Learning Algorithm". In: *Algorithms* 16.9 (2023). DOI: 10.3390/a16090404.

[23] J. Jegan and I. Karuppasamy. "Simulation and Validation of Permanent Magnet Synchronous Motor Drives using Reinforcement Learning". In: *2023 IEEE 8th International Conference for Convergence in Technology (I2CT)*. 2023. DOI: 10.1109/I2CT57861.2023.10126378.

[24] Z. Hu, Y. Zhang, M. Li, and Y. Liao. "Speed Optimization Control of a Permanent Magnet Synchronous Motor based on TD3". In: *Energies* 18.4 (2025). DOI: 10.3390/en18040901.

[25] H. M. Kaboolio, S. Schueller, A. v. Hoegen, R. W. De Doncker, and N. Fuengwarodsakul. "Design and Analysis of a Torque Controller for an IPMSM using Reinforcement Learning". In: *2023 IEEE Transportation Electrification Conference and Expo, Asia-Pacific (ITEC Asia-Pacific)*. 2023. DOI: 10.1109/ITECAsia-Pacific59272.2023.10372318.

[26] D. Gao, S. Wang, Y. Yang, et al. "An Intelligent Control Method for Servo Motor based on Reinforcement Learning". In: *Algorithms* 17.1 (2024). DOI: 10.3390/a17010014.

[27] A. Kushwaha and M. Gopal. "Reinforcement Learning-based Controller for Field-Oriented Control of Induction Machine". In: *Soft Computing for Problem Solving*. Ed. by J. C. Bansal, K. N. Das, A. Nagar, K. Deep, and A. K. Ojha. Singapore: Springer Singapore, 2019, pp. 737–749.

[28] X. Li, Y. Qi, T. Zhao, et al. "Data-Driven Deep Reinforcement Learning Control: Application to New Energy Aircraft PMSM". In: *2021 China Automation Congress (CAC)*. 2021, pp. 7127–7132. DOI: 10.1109/CAC53003.2021.9728191.

[29] Z. Song, J. Yang, X. Mei, T. Tao, and M. Xu. "Deep Reinforcement Learning for Permanent Magnet Synchronous Motor Speed Control Systems". In: *Neural Computing and Applications* 33.10 (May 2021), pp. 5409–5418. DOI: 10.1007/s00521-020-05352-1.

[30] T. Pajchrowski, P. Siwek, and A. Wójcik. "Application of the Reinforcement Learning method for Adaptive Electric Drive Control with Variable Parameters". In: *2021 IEEE 19th International Power Electronics and Motion Control Conference (PEMC)*. 2021, pp. 687–694. DOI: 10.1109/PEMC48073.2021.9432592.

[31] A. Traue, G. Book, W. Kirchgässner, and O. Wallscheid. "Toward a Reinforcement Learning Environment Toolbox for Intelligent Electric Motor Control". In: *IEEE Transactions on Neural Networks and Learning Systems* 33.3 (2022), pp. 919–928. DOI: 10.1109/TNNLS.2020.3029573.

[32] M. Nicola, C.-I. Nicola, and D. Selișteanu. "Improvement of PMSM Sensorless Control based on Synergetic and Sliding Mode Controllers using a Reinforcement Learning Deep Deterministic Policy Gradient Agent". In: *Energies* 15.6 (2022). DOI: 10.3390/en15062208.

[33] Ayesha and A. Y. Memon. "Reinforcement Learning-based Field Oriented Control of an Induction Motor". In: *2022 Third International Conference on Latest trends in Electrical Engineering and Computing Technologies (INTELLECT)*. 2022. DOI: 10.1109/INTELLECT55495.2022.9969403.

[34] D. Zholtayev, M. Rubagotti, and T. D. Do. "Deep Reinforcement Learning for PMSG Wind Turbine Control via Twin Delayed Deep Deterministic Policy Gradient (TD3)". In: *Optimal Control Applications and Methods* 45.4 (2024), pp. 1889–1906. DOI: https://doi.org/10.1002/oca.3129. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/oca.3129.

[35] E. Kiliç. “Deep Reinforcement Learning-based Controller for Field-Oriented Control of SynRM”. In: *IEEE Access* 13 (2025), pp. 2855–2861. DOI: 10.1109/ACCESS.2024.3524156.

[36] A. M. Hassan, J. Ababneh, H. Attar, T. Shamseldin, A. Abdelbaset, and M. E. Metwally. “Reinforcement Learning Algorithm for improving Speed Response of a Five-Phase Permanent Magnet Synchronous Motor based Model Predictive Control”. In: *PLOS ONE* 20 (Jan. 2025). DOI: 10.1371/journal.pone.0316326.

[37] S. Bhattacharjee, S. Halder, Y. Yan, A. Balamurali, L. V. Iyer, and N. C. Kar. “Real-Time SIL Validation of a Novel PMSM Control based on Deep Deterministic Policy Gradient Scheme for Electrified Vehicles”. In: *IEEE Transactions on Power Electronics* 37.8 (2022), pp. 9000–9011. DOI: 10.1109/TPEL.2022.3153845.

[38] L. N. Tan and T. C. Pham. “Optimal Tracking Control for PMSM with Partially Unknown Dynamics, Saturation Voltages, Torque, and Voltage Disturbances”. In: *IEEE Transactions on Industrial Electronics* 69.4 (2022), pp. 3481–3491. DOI: 10.1109/TIE.2021.3075892.

[39] J. Zhao, C. Yang, W. Gao, and L. Zhou. “Reinforcement Learning and Optimal Control of PMSM Speed Servo System”. In: *IEEE Transactions on Industrial Electronics* 70.8 (2023), pp. 8305–8313. DOI: 10.1109/TIE.2022.3220886.

[40] W.-L. Peng, Y.-W. Lan, S.-G. Chen, F.-J. Lin, R.-I. Chang, and J.-M. Ho. “Reinforcement Learning Control for Six-Phase Permanent Magnet Synchronous Motor Position Servo Drive”. In: *2020 3rd IEEE International Conference on Knowledge Innovation and Invention (ICKII)*. 2020, pp. 332–335. DOI: 10.1109/ICKII50300.2020.9318882.

[41] O. Menendez, F. Ruiz, D. Pesantez, J. Vasconez, and J. Rodriguez. “Model-Free Neural Network-based Current Control for Voltage Source Inverter”. In: *2024 IEEE International Conference on Automation/XXVI Congress of the Chilean Association of Automatic Control (ICA-ACCA)*. 2024. DOI: 10.1109/ICA-ACCA62622.2024.10766747.

[42] A. N. Alquennah, A. Krama, H. Abu-Rub, A. Ghrayeb, and S. Bayhan. “Model Free Reinforcement Learning based Controller for Grid-tied 9-Level Packed-E-Cell Multi-level Inverter”. In: *2024 IEEE Energy Conversion Congress and Exposition (ECCE)*. 2024, pp. 4437–4443. DOI: 10.1109/ECCE55643.2024.10861696.

[43] Y. Wan, Q. Xu, and T. Dragičević. “Reinforcement Learning-based Predictive Control for Power Electronic Converters”. In: *IEEE Transactions on Industrial Electronics* (2024). DOI: 10.1109/TIE.2024.3472299.

[44] O. Zandi and J. Poshtan. “Voltage Control of DC–DC Converters through Direct Control of Power Switches using Reinforcement Learning”. In: *Engineering Applications of Artificial Intelligence* 120 (2023), p. 105833. DOI: https://doi.org/10.1016/j.engappai.2023.105833.

[45] O. Menéndez, D. López-Caiza, L. Tarisciotti, F. Ruiz, F. Auat-Cheein, and J. Rodríguez. “Assessment of Deep Reinforcement Learning Algorithms for Three-Phase Inverter Control”. In: *2023 IEEE 8th Southern Power Electronics Conference and 17th Brazilian Power Electronics Conference (SPEC/COBEP)*. 2023. DOI: 10.1109/SPEC56436.2023.10407331.

[46] N. Mazaheri, D. Santamargarita, E. Bueno, D. Pizarro, and S. Cobreces. "A Deep Reinforcement Learning Approach to DC-DC Power Electronic Converter Control with Practical Considerations". In: *Energies* 17.14 (2024). DOI: 10.3390/en17143578.

[47] J. Piela, F. Ecker, N. Szabó, B. H. Zacher, and C. Schumann. "Comparison of Supervised and Reinforcement Learning based Current Control in Power Electronic Circuits". In: *2024 IEEE Design Methodologies Conference (DMC)*. 2024. DOI: 10.1109/DMC62632.2024.10812121.

[48] S. Boshoff, J. Stenner, D. Weber, et al. "Hybrid Control of Interconnected Power Converters using both Expert-Driven Droop and Data-Driven Reinforcement Learning Approaches". In: *PESS 2023; Power and Energy Student Summit*. 2023, pp. 124–129.

[49] A. Rajamallaiah, S. P. K. Karri, M. L. Alghaythi, and M. S. Alshammari. "Deep Reinforcement Learning based Control of a Grid Connected Inverter with LCL-Filter for Renewable Solar Applications". In: *IEEE Access* 12 (2024), pp. 22278–22295. DOI: 10.1109/ACCESS.2024.3364058.

[50] E. C. Kara, M. Berges, B. Krogh, and S. Kar. "Using Smart Devices for System-Level Management and Control in the Smart Grid: a Reinforcement Learning Framework". In: *2012 IEEE Third International Conference on Smart Grid Communications (SmartGridComm)*. 2012, pp. 85–90. DOI: 10.1109/SmartGridComm.2012.6485964.

[51] Q. Huang, R. Huang, W. Hao, J. Tan, R. Fan, and Z. Huang. "Adaptive Power System Emergency Control using Deep Reinforcement Learning". In: *IEEE Transactions on Smart Grid* 11.2 (2020), pp. 1171–1182. DOI: 10.1109/TSG.2019.2933191.

[52] Q. Yang, G. Wang, A. Sadeghi, G. B. Giannakis, and J. Sun. "Two-Timescale Voltage Control in Distribution Grids using Deep Reinforcement Learning". In: *IEEE Transactions on Smart Grid* 11.3 (2020), pp. 2313–2323. DOI: 10.1109/TSG.2019.2951769.

[53] J. Duan, D. Shi, R. Diao, et al. "Deep-Reinforcement-Learning-based Autonomous Voltage Control for Power Grid Operations". In: *IEEE Transactions on Power Systems* 35.1 (2020), pp. 814–817. DOI: 10.1109/TPWRS.2019.2941134.

[54] S. Wang, J. Duan, D. Shi, et al. "A Data-Driven Multi-Agent Autonomous Voltage Control Framework using Deep Reinforcement Learning". In: *IEEE Transactions on Power Systems* 35.6 (2020), pp. 4644–4654. DOI: 10.1109/TPWRS.2020.2990179.

[55] S. Totaro, I. Boukas, A. Jonsson, and B. Cornélusse. "Lifelong Control of Off-Grid Microgrid with Model-based Reinforcement Learning". In: *Energy* 232 (2021), p. 121035. DOI: https://doi.org/10.1016/j.energy.2021.121035.

[56] Y. Gao, Y. Matsunami, S. Miyata, and Y. Akashi. "Operational Optimization for Off-Grid Renewable Building Energy System using Deep Reinforcement Learning". In: *Applied Energy* 325 (2022), p. 119783. DOI: https://doi.org/10.1016/j.apenergy.2022.119783.

[57] W. Cui, J. Li, and B. Zhang. “Decentralized Safe Reinforcement Learning for Inverter-based Voltage Control”. In: *Electric Power Systems Research* 211 (2022), p. 108609. DOI: https://doi.org/10.1016/j.epsr.2022.108609.

[58] R. Huang, Y. Chen, T. Yin, et al. “Learning and Fast Adaptation for Grid Emergency Control via Deep Meta Reinforcement Learning”. In: *IEEE Transactions on Power Systems* 37.6 (2022), pp. 4168–4178. DOI: 10.1109/TPWRS.2022.3155117.

[59] R. Huang, Y. Chen, T. Yin, et al. “Accelerated Derivative-Free Deep Reinforcement Learning for Large-Scale Grid Emergency Voltage Control”. In: *IEEE Transactions on Power Systems* 37.1 (2022), pp. 14–25. DOI: 10.1109/TPWRS.2021.3095179.

[60] D. Chen, K. Chen, Z. Li, et al. “PowerNet: Multi-Agent Deep Reinforcement Learning for Scalable Powergrid Control”. In: *IEEE Transactions on Power Systems* 37.2 (2022), pp. 1007–1017. DOI: 10.1109/TPWRS.2021.3100898.

[61] F. Alfaverh, M. Denaï, and Y. Sun. “Optimal Vehicle-to-Grid Control for Supplementary Frequency Regulation using Deep Reinforcement Learning”. In: *Electric Power Systems Research* 214 (2023), p. 108949. DOI: https://doi.org/10.1016/j.epsr.2022.108949.

[62] R. R. Hossain, T. Yin, Y. Du, et al. “Efficient Learning of Power Grid Voltage Control Strategies via Model-based Deep Reinforcement Learning”. In: *Machine Learning* 113.5 (May 2024), pp. 2675–2700. DOI: 10.1007/s10994-023-06422-w.

[63] F.-D. Li, M. Wu, Y. He, and X. Chen. “Optimal Control in Microgrid using Multi-Agent Reinforcement Learning”. In: *ISA Transactions* 51.6 (2012), pp. 743–751. DOI: https://doi.org/10.1016/j.isatra.2012.06.010.

[64] R. Rocchetta, L. Bellani, M. Compare, E. Zio, and E. Patelli. “A Reinforcement Learning Framework for Optimal Operation and Maintenance of Power Grids”. In: *Applied Energy* 241 (2019), pp. 291–301. DOI: https://doi.org/10.1016/j.apenergy.2019.03.027.

[65] T. Remani, E. Jasmin, and T. I. Ahamed. “Residential Load Scheduling with Renewable Generation in the Smart Grid: a Reinforcement Learning Approach”. In: *IEEE Systems Journal* 13.3 (2019), pp. 3283–3294. DOI: 10.1109/JSYST.2018.2855689.

[66] M. Roesch, C. Linder, R. Zimmermann, A. Rudolf, A. Hohmann, and G. Reinhart. “Smart Grid for Industry Using Multi-Agent Reinforcement Learning”. In: *Applied Sciences* 10.19 (2020). DOI: 10.3390/app10196900.

[67] X. Hu, Y. Zhang, H. Xia, W. Wei, Q. Dai, and J. Li. “Toward Fair Power Grid Control: A Hierarchical Multiobjective Reinforcement Learning Approach”. In: *IEEE Internet of Things Journal* 11.4 (2024), pp. 6582–6595. DOI: 10.1109/JIOT.2023.3314522.

[68] W.-G. Lee and H.-M. Kim. “Deep Reinforcement Learning-based Dynamic Droop Control Strategy for Real-Time Optimal Operation and Frequency Regulation”. In: *IEEE Transactions on Sustainable Energy* 16.1 (2025), pp. 284–294. DOI: 10.1109/TSTE.2024.3454298.

[69] K. S. Stille, D. Weber, J. Lange, T. Vogt, O. Wallscheid, and J. Böcker. “Emulation of Microgrids for Research and Validation of Control and Operation Strategies”. In:

2020 International Symposium on Power Electronics, Electrical Drives, Automation and Motion (SPEEDAM). 2020, pp. 324–329. DOI: 10.1109/SPEEDAM48782.2020.9161971.

[70] X. Lin-Shi, F. Morel, A. M. Llor, B. Allard, and J.-M. Retif. "Implementation of Hybrid Control for Motor Drives". In: *IEEE Transactions on Industrial Electronics* 54.4 (2007), pp. 1946–1952. DOI: 10.1109/TIE.2007.898303.

[71] J. Xiong, Q. Wang, Z. Yang, et al. "Parametrized Deep Q-Networks Learning: Reinforcement Learning with Discrete-Continuous Hybrid Action Space". In: *CoRR* abs/1810.06394 (2018). arXiv: 1810.06394. URL: http://arxiv.org/abs/1810.06394.

[72] R. De Doncker, D. Pulle, and A. Veltman. *Advanced Electrical Drives: Analysis, Modeling, Control*. Power Systems. Springer Netherlands, 2010.

[73] M. Peña, M. Meyer, O. Wallscheid, and J. Böcker. "Model Predictive Direct Self-Control for Six-Step Operation of Permanent-Magnet Synchronous Machines". In: *IEEE Transactions on Power Electronics* 38.10 (2023). DOI: 10.1109/TPEL.2023.3286713.

[74] P. Rehlaender, F. Schafmeister, and J. Böcker. "Interleaved Single-Stage LLC Converter Design utilizing Half- and Full-Bridge Configurations for Wide Voltage Transfer Ratio Applications". In: *IEEE Transactions on Power Electronics* 36.9 (2021). DOI: 10.1109/TPEL.2021.3067843.

[75] E. Torán, M. Liberos, I. Patrao, R. González-Medina, G. Garcerá, and E. Figueres. "Comparative Analysis of B4 and B6 Inverter Topologies for Grid-Connected Operation". In: *2022 IEEE 1st Industrial Electronics Society Annual On-Line Conference (ONCON)*. 2022, pp. 1–6. DOI: 10.1109/ONCON56984.2022.10126925.

[76] W. Leonhard. *Control of Electrical Drives*. Springer Science & Business Media, 2001.

[77] R. W. Erickson and D. Maksimović. *Fundamentals of Power Electronics*. Springer Nature, 2020.

[78] J. Holtz. "Pulsewidth Modulation for Electronic Power Conversion". In: *Proceedings of the IEEE* 82.8 (1994), pp. 1194–1214. DOI: 10.1109/5.301684.

[79] J. Holtz, M. Höltgen, and J. O. Krah. "A Space Vector Modulator for the High-Switching Frequency Control of Three-Level SiC Inverters". In: *IEEE Transactions on Power Electronics* 29.5 (2014), pp. 2618–2626. DOI: 10.1109/TPEL.2013.2280768.

[80] K. Zhou and D. Wang. "Relationship Between Space-Vector Modulation and Three-Phase Carrier-based PWM: a Comprehensive Analysis". In: *IEEE Transactions on Industrial Electronics* 49.1 (2002), pp. 186–196. DOI: 10.1109/41.982262.

[81] Y. Yang, S. M. Castano, R. Yang, et al. "Design and Comparison of Interior Permanent Magnet Motor Topologies for Traction Applications". In: *IEEE Transactions on Transportation Electrification* 3.1 (2017), pp. 86–97. DOI: 10.1109/TTE.2016.2614972.

[82] A. Boglietti, E. Carpaneto, M. Cossale, A. Lucco Borlera, D. Staton, and M. Popescu. "Electrical Machine First Order Short-Time Thermal Transients Model: Measurements and Parameters Evaluation". In: *Annual Conference of the IEEE Industrial Electronics Society (IECON)*. 2014. DOI: 10.1109/IECON.2014.7048555.

[83] O. Wallscheid. “Thermal Monitoring of Electric Motors: State-of-the-Art Review and Future Challenges”. In: *IEEE Open Journal of Industry Applications* 2 (2021), pp. 204–223. DOI: 10.1109/OJIA.2021.3091870.

[84] C. Hackl, J. Kullick, and N. Monzen. “Generic Loss Minimization for Nonlinear Synchronous Machines by Analytical Computation of Optimal Reference Currents considering Copper and Iron Losses”. In: *IEEE International Conference on Industrial Technology (ICIT)*. 2021. DOI: 10.1109/ICIT46573.2021.9453497.

[85] A. Brosch, O. Wallscheid, and J. Böcker. “Time-Optimal Model Predictive Control of Permanent Magnet Synchronous Motors Considering Current and Torque Constraints”. In: *IEEE Transactions on Power Electronics* 38.7 (2023), pp. 7945–7957. DOI: 10.1109/TPEL.2023.3265705.

[86] R. S. Sutton and A. G. Barto. *Reinforcement Learning: an Introduction*. Second. The MIT Press, 2018. URL: http://incompleteideas.net/book/the-book-2nd.html.

[87] D. Silver. *Lectures on Reinforcement Learning*. URL: https://www.davidsilver.uk/teaching/. 2015.

[88] D. Görges. “Relations between Model Predictive Control and Reinforcement Learning”. In: *IFAC-PapersOnLine* 50.1 (2017). 20th IFAC World Congress, pp. 4920–4928. DOI: https://doi.org/10.1016/j.ifacol.2017.08.747.

[89] G. C. Goodwin, S. F. Graebe, and M. F. Salgado. *Control System Design*. Addison-Wesley, 2000.

[90] E. Camacho and C. Alba. *Model Predictive Control*. Advanced Textbooks in Control and Signal Processing. Springer London, 2013. URL: https://books.google.de/books?id=tXZDAAAAQBAJ.

[91] L. Brunke, M. Greeff, A. W. Hall, et al. “Safe Learning in Robotics: From Learning-based Control to Safe Reinforcement Learning”. In: *Annual Review of Control, Robotics, and Autonomous Systems* 5.1 (2022), pp. 411–444. DOI: 10.1146/annurev-control-042920-020211. eprint: https://doi.org/10.1146/annurev-control-042920-020211.

[92] C. M. Bishop. *Neural Networks for Pattern Recognition*. Clarendon Press - Oxford, 1995.

[93] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. “Learning Internal Representations by Error Propagation”. In: *Neurocomputing, Volume 1: Foundations of Research*. The MIT Press, Apr. 1988. DOI: 10.7551/mitpress/4943.003.0128. eprint: https://direct.mit.edu/book/chapter-pdf/2299556/c018389_9780262267137.pdf.

[94] S. Hochreiter and J. Schmidhuber. “Long Short-Term Memory”. In: *Neural Computation* 9.8 (Nov. 1997), pp. 1735–1780. DOI: 10.1162/neco.1997.9.8.1735. eprint: https://direct.mit.edu/neco/article-pdf/9/8/1735/813796/neco.1997.9.8.1735.pdf.

[95] K. Cho, B. van Merrienboer, C. Gulcehre, et al. *Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation*. 2014. arXiv: 1406.1078 [cs.CL]. URL: https://arxiv.org/abs/1406.1078.

[96] A. Apicella, F. Donnarumma, F. Isgrò, and R. Prevete. "A Survey on Modern Trainable Activation Functions". In: *Neural Networks* 138 (2021), pp. 14–32. DOI: https://doi.org/10.1016/j.neunet.2021.01.026.

[97] T. Verdonck, B. Baesens, M. Óskarsdóttir, and S. vanden Broucke. "Special Issue on Feature Engineering Editorial". In: *Machine Learning* 113 (Aug. 2021). DOI: 10.1007/s10994-021-06042-2.

[98] R. E. Best. *Phase-Locked Loops: Design, Simulation, and Applications*. en. 6th Edition. New York: McGraw-Hill Education, 2007. URL: https://www.accessengineeringlibrary.com/content/book/9780071493758.

[99] S. Ioffe and C. Szegedy. "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift". In: *CoRR* abs/1502.03167 (2015). arXiv: 1502.03167. URL: http://arxiv.org/abs/1502.03167.

[100] M. Agarwal and V. Aggarwal. "Blind Decision Making: Reinforcement Learning with Delayed Observations". In: *Pattern Recognition Letters* 150 (2021), pp. 176–182. DOI: https://doi.org/10.1016/j.patrec.2021.06.022.

[101] D. Luenberger. "Observers for Multivariable Systems". In: *IEEE Transactions on Automatic Control* 11.2 (1966), pp. 190–197. DOI: 10.1109/TAC.1966.1098323.

[102] R. E. Kalman. "A New Approach to Linear Filtering and Prediction Problems". In: *Journal of Basic Engineering* 82.1 (Mar. 1960), pp. 35–45. DOI: 10.1115/1.3662552. eprint: https://asmedigitalcollection.asme.org/fluidsengineering/article-pdf/82/1/35/5518977/35_1.pdf.

[103] O. Wallscheid, M. Meyer, and J. Böcker. "An Open-Loop Operation Strategy for Induction Motors considering Iron Losses and Saturation Effects in Automotive Applications". In: *2015 IEEE 11th International Conference on Power Electronics and Drive Systems*. 2015, pp. 981–985. DOI: 10.1109/PEDS.2015.7203404.

[104] S.-K. Sul. *Control of Electric Machine Drive Systems*. John Wiley & Sons, 2011.

[105] S. Buso and P. Mattavelli. *Digital Control in Power Electronics*. Springer Nature, 2022.

[106] A. Brosch, S. Hanke, O. Wallscheid, and J. Böcker. "Data-Driven Recursive Least Squares Estimation for Model Predictive Current Control of Permanent Magnet Synchronous Motors". In: *IEEE Transactions on Power Electronics* 36.2 (2021), pp. 2179–2190. DOI: 10.1109/TPEL.2020.3006779.

[107] M. Alshiekh, R. Bloem, R. Ehlers, B. Könighofer, S. Niekum, and U. Topcu. "Safe Reinforcement Learning via Shielding". In: *CoRR* abs/1708.08611 (2017). arXiv: 1708.08611. URL: http://arxiv.org/abs/1708.08611.

[108] J. Löfberg. "Oops! I cannot do it again: Testing for Recursive Feasibility in MPC". In: *Automatica* 48.3 (2012), pp. 550–555. DOI: https://doi.org/10.1016/j.automatica.2011.12.003.

[109] K. P. Wabersich and M. N. Zeilinger. "Linear Model Predictive Safety Certification for Learning-based Control". In: *CoRR* abs/1803.08552 (2018). arXiv: 1803.08552. URL: http://arxiv.org/abs/1803.08552.

[110] T. Schindler and A. Dietz. "Real-Time Inference of Neural Networks on FPGAs for Motor Control Applications". In: *International Electric Drives Production Conference (EDPC)*. 2020. DOI: 10.1109/EDPC51184.2020.9388185.

[111] M. Rothmann and M. Porrmann. "A Survey of Domain-Specific Architectures for Reinforcement Learning". In: *IEEE Access* 10 (2022), pp. 13753–13767. DOI: 10.1109/ACCESS.2022.3146518.

[112] I. D. Landau and G. Zito. *Digital Control Systems: Design, Identification and Implementation*. Vol. 130. Springer, 2006.

[113] J. Holtz, W. Lotzkat, and A. Khambadkone. "On Continuous Control of PWM Inverters in the Overmodulation Range including the Six-Step Mode". In: *IEEE Transactions on Power Electronics* 8.4 (1993), pp. 546–553. DOI: 10.1109/63.261026.

[114] M. Schenke, B. Haucke-Korber, and O. Wallscheid. *Coffee Machine vs. Machine Learning: Who is Quicker?* 2023. URL: https://www.youtube.com/watch?v=hQ49Mc6LV78.

[115] A. Brosch, O. Wallscheid, and J. Böcker. "Model Predictive Torque Control for Permanent-Magnet Synchronous Motors using a Stator-Fixed Harmonic Flux Reference Generator in the Entire Modulation Range". In: *IEEE Transactions on Power Electronics* 38.4 (2023), pp. 4391–4404. DOI: 10.1109/TPEL.2022.3229619.

[116] E. C. Kerrigan and J. M. Maciejowski. *Soft Constraints and Exact Penalty Functions in Model Predictive Control*. 2000. URL: https://api.semanticscholar.org/CorpusID:18511401.

[117] C. Schmid and L. Biegler. "Quadratic Programming Methods for Reduced Hessian SQP". In: *Computers & Chemical Engineering* 18.9 (1994). An International Journal of Computer Applications in Chemical Engineering. DOI: https://doi.org/10.1016/0098-1354(94)E0001-4.

[118] I. D. De Martin, A. Brosch, F. Tinazzi, and M. Zigliotto. "Continuous Control Set Model Predictive Torque Control with Minimum Current Magnitude Criterion for Synchronous Motor Drives". In: *IEEE Transactions on Industrial Electronics* 71.7 (2024), pp. 6787–6796. DOI: 10.1109/TIE.2023.3308132.

[119] L. Hölsch and O. Wallscheid. "Evaluation of the Efficiency Measurement Uncertainty of Electric Drive Test Benches for Direct Data-Driven Control Optimization". In: *(Under review)* (Jan. 2025). DOI: 10.36227/techrxiv.173603411.17559954/v1.

[120] A. Brosch, O. Wallscheid, and J. Böcker. "Long-Term Memory Recursive Least Squares Online Identification of Highly Utilized Permanent Magnet Synchronous Motors for Finite-Control-Set Model Predictive Control". In: *IEEE Transactions on Power Electronics* 38.2 (2023), pp. 1451–1467. DOI: 10.1109/TPEL.2022.3206598.

[121] V. Mnih, K. Kavukcuoglu, D. Silver, et al. *Playing Atari with Deep Reinforcement Learning*. 2013. arXiv: 1312.5602 [cs.LG].

[122] V. Mnih, K. Kavukcuoglu, D. Silver, et al. "Human-Level Control through Deep Reinforcement Learning". In: *Nature* 518 (Feb. 2015), pp. 529–33. DOI: 10.1038/nature14236.

[123] D. P. Kingma and J. Ba. *Adam: A Method for Stochastic Optimization*. 2014. DOI: 10.48550/ARXIV.1412.6980.

[124] M. G. Bellemare, W. Dabney, and R. Munos. *A Distributional Perspective on Reinforcement Learning*. 2017. arXiv: 1707.06887 [cs.LG]. URL: https://arxiv.org/abs/1707.06887.

[125] W. Dabney, M. Rowland, M. G. Bellemare, and R. Munos. *Distributional Reinforcement Learning with Quantile Regression*. 2017. arXiv: 1710.10044 [cs.AI]. URL: https://arxiv.org/abs/1710.10044.

[126] M. Andrychowicz, F. Wolski, A. Ray, et al. *Hindsight Experience Replay*. 2018. arXiv: 1707.01495 [cs.LG]. URL: https://arxiv.org/abs/1707.01495.

[127] M. Hessel, J. Modayil, H. van Hasselt, et al. *Rainbow: Combining Improvements in Deep Reinforcement Learning*. 2017. arXiv: 1710.02298 [cs.AI]. URL: https://arxiv.org/abs/1710.02298.

[128] C. Bauer, T. Kenter, M. Lass, et al. "Noctua 2 Supercomputer". In: *Journal of large-scale research facilities JLSRF* 9 (2024). DOI: https://doi.org/10.17815/jlsrf-8-187.

[129] J. Bergstra, D. Yamins, and D. Cox. "Making a Science of Model Search: Hyperparameter Optimization in Hundreds of Dimensions for Vision Architectures". In: ed. by S. Dasgupta and D. McAllester. Vol. 28. Proceedings of Machine Learning Research 1. Atlanta, Georgia, USA: PMLR, June 2013, pp. 115–123. URL: http://proceedings.mlr.press/v28/bergstra13.html.

[130] F. Hutter, H. Hoos, and K. Leyton-Brown. "An Efficient Approach for Assessing Hyperparameter Importance". In: *Proceedings of the 31st International Conference on Machine Learning*. Ed. by E. P. Xing and T. Jebara. Vol. 32. Proceedings of Machine Learning Research 1. Bejing, China: PMLR, June 2014, pp. 754–762. URL: https://proceedings.mlr.press/v32/hutter14.html.

[131] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama. *Optuna: A Next-generation Hyperparameter Optimization Framework*. 2019. arXiv: 1907.10902 [cs.LG]. URL: https://arxiv.org/abs/1907.10902.

[132] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller. "Deterministic Policy Gradient Algorithms". In: *Proceedings of the 31st International Conference on Machine Learning*. Ed. by E. P. Xing and T. Jebara. Vol. 32. Proceedings of Machine Learning Research 1. Bejing, China: PMLR, June 2014, pp. 387–395. URL: https://proceedings.mlr.press/v32/silver14.html.

[133] T. P. Lillicrap, J. J. Hunt, A. Pritzel, et al. *Continuous Control with Deep Reinforcement Learning*. 2019. arXiv: 1509.02971 [cs.LG]. URL: https://arxiv.org/abs/1509.02971.

[134] G. E. Uhlenbeck and L. S. Ornstein. "On the Theory of the Brownian Motion". In: *Phys. Rev.* 36 (5 Sept. 1930), pp. 823–841. DOI: 10.1103/PhysRev.36.823.

[135] J. Schulman, S. Levine, P. Moritz, M. I. Jordan, and P. Abbeel. "Trust Region Policy Optimization". In: *CoRR* abs/1502.05477 (2015). arXiv: 1502.05477. URL: http://arxiv.org/abs/1502.05477.

[136] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. "Proximal Policy Optimization Algorithms". In: *CoRR* abs/1707.06347 (2017). arXiv: 1707.06347. URL: http://arxiv.org/abs/1707.06347.

[137] S. Fujimoto, H. van Hoof, and D. Meger. “Addressing Function Approximation Error in Actor-Critic Methods”. In: *CoRR* abs/1802.09477 (2018). arXiv: 1802.09477. URL: http://arxiv.org/abs/1802.09477.

[138] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine. “Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor”. In: *CoRR* abs/1801.01290 (2018). arXiv: 1801.01290. URL: http://arxiv.org/abs/1801.01290.

[139] D. Zwillinger. *CRC Standard Mathematical Tables and Formulae*. Chapman and Hall/CRC, 2002.

[140] *Ellipse*. Accessed: 2025-02-19. URL: https://en.wikipedia.org/wiki/Talk:Ellipse.