Technische Universität München
TUM School of Engineering and Design

Deep Reinforcement Learning basierte Leistungsmanagementstrategie in 12 V-Bordnetzen

Ömer Tan

Vollständiger Abdruck der von der TUM School of Engineering and Design der Technischen Universität München zur Erlangung eines

Doktors der Ingenieurwissenschaften (Dr.-Ing.)

genehmigten Dissertation.

Vorsitz: Prof. Dr.-Ing. Hans-Georg Herzog

Prüfende der Dissertation:

1. Prof. Dr.-Ing. Dr. h. c. Ralph Kennel
2. Prof. Dr. rer. nat. Franz Kreupl

Die Dissertation wurde am 10.08.2023 bei der Technischen Universität München eingereicht und durch die TUM School of Engineering and Design am 04.03.2024 angenommen.

Kurzfassung

Die Zahl der elektrischen Verbraucher im 12 V-Bordnetz nimmt stetig zu. Dies lässt sich auf die gestiegene Komfort- und Sicherheitsansprüche in Kraftfahrzeugen zurückführen. Zudem werden wegen Effizienzgründen immer mehr Nebenaggregate elektrifiziert. Dieser Trend wird sich sehr wahrscheinlich aufgrund neuer Mobilitätskonzepte wie dem autonomen Fahren fortsetzen. Die steigende Komplexität der Bordnetze stellt die Fahrzeughersteller vor wachsende Herausforderungen bei der Bordnetzstabilität. Hochleistungsverbraucher wie die elektrische Lenkung oder Bremse können in kritischen Fahrsituationen die Spannung im Bordnetz deutlich senken. Durch die entstehende Unterspannung können Steuergeräte abschalten oder beschädigt werden. Daher werden zur Koordinierung der Stromflüsse innerhalb des Bordnetzes hochentwickelte Energie- und Leistungsmanagementsysteme eingesetzt. Diese basieren auf Regeln oder mathematischer Optimierung und bringen einen hohen zeitlichen Aufwand in der Entwicklung mit sich, welche stark von Erfahrungen und Expertenwissen geprägt ist. Mit Reinforcement Learning (RL) bietet das maschinelle Lernen eine weitere Methode an, die zur optimalen Entscheidungsfindung in komplexen Steuerungsproblemen fähig ist. Dabei ist kein detailliertes Systemverständnis vorausgesetzt. In Zeiten immer komplexerer Bordnetze kann dies ein entscheidender Vorteil bezüglich Know-How, Kosten- und Zeitaufwand im Entwicklungsprozess sein.
In dieser Dissertation wird anhand einer simulationsbasierten Studie analysiert, welcher RL Algorithmus sich für die Entwicklung einer Leistungsmanagementstrategie eignet und wie diese in MATLAB/Simulink umgesetzt sowie optimiert werden kann. Um das Spannungsverhalten eines konventionellen Bordnetzes zu simulieren, werden zunächst Modelle des Generators, der elektrischen Verbrauchern und der Batterie erstellt. Auf Grundlage der Bordnetzsimulation wird eine RL-basierte Methode zur Reduktion von Spannungseinbrüchen in 12 V-Bordnetzen entwickelt.
Die Ergebnisse zeigen, dass mit Double Deep Q-Learning über eine große Anzahl an kritischen Bordnetzszenarien durchschnittlich ein Verbesserungspotenzial der Spannungseinbrüche gegenüber der Strategie eines Mittelklassefahrzeugs besteht. Dabei wird als Deep Q-Network ein rekurrentes neuronales Netz mit Long Short-Term Memory (LSTM) Zellen verwendet. Die Szenarien konnten durch Fahrzeugmessungen simulativ nachgebildet werden. Gleichzeitig ist eine deutliche Verhaltensänderung bei der Degradierung verschiedener Verbraucher erkennbar, sodass die Fahrzeuginsassen insgesamt weniger Komforteinschränkungen erfahren. Insgesamt zeigt die vorliegende Arbeit das Potenzial von RL speziell im Leistungsmanagement auf. Maschinelles Lernen und insbesondere Deep-Learning-basierte Verfahren stellen im Vergleich zu herkömmlichen Entwicklungsmethoden einen besser skalierbaren und anpassbaren Lösungsprozess dar, der eine Ausweitung auf das gesamte Energiemanagement des Fahrzeugs ermöglicht.

Abstract

The number of electrical consumers in the 12 V electrical system is constantly increasing. This can be attributed to the increased demand for comfort and safety in motor vehicles. In addition, more and more auxiliary units are being electrified for efficiency reasons. This trend is very likely to continue due to new mobility concepts such as autonomous driving. The increasing complexity of power nets is presenting vehicle manufacturers with growing challenges in terms of power net stability. High-power consumers such as the electric steering or brakes can significantly reduce the voltage in the vehicle electrical system in critical driving situations. The resulting undervoltage can cause control units to shut down or be damaged. For this reason, sophisticated energy and power management systems are used to coordinate the current flows within the vehicle electrical system. These are based on rules or mathematical optimization and involve a great deal of time in development, which is strongly influenced by experience and expert knowledge. With Reinforcement Learning (RL), machine learning offers another method capable of optimal decision making in complex control problems. It does not require a detailed understanding of the system. In times of increasingly complex vehicle electrical systems, this can be a decisive advantage in terms of know-how, cost and time in the development process.
This dissertation uses a simulation-based study to analyze which RL algorithm is suitable for developing a power management strategy and how it can be implemented and optimized in MATLAB/Simulink. In order to simulate the voltage behavior of a conventional on-board power system, models of the generator, the electrical loads and the battery are first created. Based on the power net simulation, an RL-based method for reducing voltage dips in 12 V electrical systems is developed.
The results show that with Double Deep Q-learning over a large number of critical on-board network scenarios, there is on average a potential for improvement of voltage dips against the strategy of a mid-range vehicle. A recurrent neural network with Long Short-Term Memory (LSTM) cells is used as the Deep Q network. The scenarios could be simulated by vehicle measurements. At the same time, a significant behavioral change in the degradation of various loads is evident, so that vehicle occupants experience less overall comfort degradation. Overall, this work demonstrates the potential of reinforcement learning specifically in power management. Compared to conventional development methods, machine learning and in particular deep learning-based methods represent a more scalable and adaptable solution process that allows for an extension to the entire energy management of the vehicle.

Vorwort und Danksagung

Die vorliegende Dissertation entstand während meiner Tätigkeit als Doktorand bei der IAV GmbH in München in Kooperation mit dem Lehrstuhl für Elektrische Antriebssysteme und Leistungselektronik (EAL) an der Technischen Universität München (TUM).

Mein besonderer Dank gilt Herrn Prof. Dr.-Ing. Ralph Kennel, der mir die Möglichkeiten und Freiheiten zum Anfertigen dieser interessanten Arbeit gab. Darüber hinaus war die fachliche Diskussion mit ihm ein wichtiger Impulsgeber für das Gelingen dieser Arbeit.

Mein weiterer Dank gilt Herrn Prof. Dr. Franz Kreupl für die Übernahme eines Gutachtens sowie für seine Unterstützung und seine wertvollen Hinweise.

Ganz besonders danke ich meinem Mentor Dr. rer. nat. Ahmet Taskiran für die zahlreichen Anregungen, die engagierte Betreuung und die Unterstützung vor und während meiner Promotionstätigkeit sowie für das mir entgegengebrachte Vertrauen.

Des Weiteren danke ich Marco Mecklenburg und Emre Boyacigil, die mir den Einstieg in das interessante Thema Energiemanagementsysteme bei der IAV GmbH ermöglicht haben. Ebenso möchte ich mich bei meinen Kolleginnen und Kollegen für die großartige Zusammenarbeit und ihre fachliche Unterstützung bedanken.

Ein spezieller Dank gilt an die Studenten, welche mich mit ihren Abschlussarbeiten entscheidend unterstützt haben, insbesondere Herrn Maximilian Graf.

Danke sagen möchte ich auch meiner Familie und allen Freunden für ihre Unterstützung und ihr Verständnis. Die motivierenden Gespräche mit meinem Vater Hamit haben wesentlich zum Gelingen dieser Arbeit beigetragen.

An letzter, doch eigentlich an erster Stelle möchte ich mich bei meiner Frau Gözde und meinen Söhnen Mirac und Imran bedanken. Speziell bin ich für ihre Geduld, Rücksichtnahme und für die Motivation in aller Hinsicht dankbar. Dieser Beistand war mir immer sehr wichtig.

Inhaltsverzeichnis

Abkürzungsverzeichnis

A2C	Advantage Actor-Critic
A3C	Asynchronous Advantage Actor-Critic
ABS	Antiblockiersystem
AC	Actor-Critic
AGM	Absorbent Glass Mat
ASR	Antischlupfregelung
BEV	Battery Electric Vehicle
CPU	Central Processing Unit
DDPG	Deep Deterministic Policy Gradient
DDQN	Double Deep Q-Network
DP	Dynamic Programming
DQN	Deep Q-Network
DSC	Dynamic Stability Control
EEMS	Elektrisches Energiemanagementsystem
EPS	Electronic Power Steering
ESB	Ersatzschaltbild
FCEV	Fuel Cell Electric Vehicle
FNN	Feedforward Neural Network
GPU	Graphics Processing Unit
HEV	Hybrid Electric Vehicle
HSR	Hinterachsschräglaufregelung
IB	Intelligent Brake
ICEV	Internal Combustion Engine Vehicle
KL	Kullback-Leibler
KI	Künstliche Intelligenz
LED	Light Emitting Diode
Li-Ionen	Lithium-Ionen
LKW	Lastkraftwagen
LSTM	Long Short-Term Memory
MATLAB	Matrix Laboratory

MDP	Markov-Decision-Process
MP	Markov-Process
MRP	Markov-Reward-Process
NiMH	Nickel-Metallhydrid
PKW	Personenkraftwagen
PG	Policy Gradients
PPO	Proximal Policy Optimization
PWM	Pulsweitenmodulation
RAM	Random-Access Memory
RARL	Reflex-Augmented Reinforcement Learning
ReLU	Rectified Linear Unit
RL	Reinforcement Learning
RNN	Recurrent Neural Network
SAC	Soft Actor-Critic
SARSA	State-Action-Reward-State-Action
SOC	State of Charge
SOF	State of Function
SOH	State of Health
TD	Temporal Difference
TD3	Twin-Delayed Deep Deterministic Policy Gradient
TRPO	Trusted Region Policy Optimization
VSM	Viable System Model
V&V	Verifikation und Validierung

Abbildungsverzeichnis

Tabellenverzeichnis

Algorithmenverzeichnis

1 Einleitung

Mit Neuentwicklungen von Bordnetzkomponenten, einem elektrischen Energiemanagementsystem bis hin zu völlig neuen Bordnetzarchitekturen optimiert die Automobilindustrie den Komfort, die Sicherheit und die Effizienz im Fahrzeug. Durch diese Verbesserungen wird versucht den Kundenwünschen und eigenen Anforderungen bezüglich Effizienz gerecht zu werden. In Folge dessen steigt die Anzahl der elektrischen Verbraucher im Fahrzeugbordnetz stetig an, woraus neue Herausforderungen an die elektrische Energieversorgung entstehen.

1.1 Motivation

Eine der wesentlichen Bestandteile des Kraftfahrzeugs ist das 12-Volt-Bordnetz, welches für das Starten des Motors und die Versorgung einer Reihe anderer wichtiger Funktionen wie Fahrerassistenzsysteme, Steuergeräte, Sensoren und Komfortausstattungen verantwortlich ist. Andererseits sind völlig neue Funktionen wie Airbags, USB-Anschlüsse und Infotainmentsysteme mit Smartphone-Schnittstellen im Auto alltäglich geworden. Ein Wechsel von einem Spannungsniveau von 6 V auf 12 V erfolgte in den 1960er Jahren. Der grundsätzliche Aufbau der Bordnetze hat sich seither nicht wesentlich verändert [1]. Um Verluste zu minimieren, erscheint eine Erhöhung der Spannung von der gesamten Fahrzeugelektronik auf 48 V logisch. Die meisten elektrischen Verbraucher, wie z.B. Druckpumpen für Bremskraftverstärker, werden jedoch von verschiedenen Herstellern für 12 V ausgelegt. Daher wird zumindest ein Teil des Bordnetzes weiterhin auf das 12 V-Spannungsniveau eingestellt [2]. Da die Traktionsbatterie bei Elektrofahrzeugen (BEV, engl. Battery Electric Vehicle) beim Verriegeln des Fahrzeugs vom Hochvoltnetz getrennt wird, hat auch der Trend zur reinen Elektromobilität noch nicht zum Ausstieg aus der 12 V-Technik geführt. Die herkömmliche Batterie verleiht dem Hochvoltspeicher zudem eine für die Sicherheit wichtige Redundanz. Dies impliziert, dass das Bordnetz von Elektrofahrzeugen noch viele Jahre die bewährte 12 V-Technik nutzen wird [3]. Gleiches gilt für Hybrid-Fahrzeuge (HEV, engl. Hybrid Electric Vehicle). Die vorliegende Arbeit beschränkt sich in der Betrachtung auf konventionelle 12 V-Bordnetze, wie sie in Fahrzeugen mit Verbrennungsmotor (ICEV, engl. Internal Combustion Engine Vehicle) eingesetzt werden.

Die Elektrifizierung von Nebenverbrauchern wie Lenkung, Fahrwerk oder Kühlmittelpumpe sorgt für eine weitere Zunahme der Anzahl elektrischer Verbraucher und erhöht die Komplexität moderner Bordnetze. Dadurch steigt der durchschnittliche Leistungsbedarf des Bordnetzes. Seit 1990 hat sich diese teilweise auf über 3 kW mehr als verdoppelt. Darüber hinaus stieg laut Kohler [4] die Spitzenleistung zwischen 1995 und 2006 etwa doppelt so schnell wie die Durchschnittsleistung. Dies stellt die Bordnetze natürlich vor erhebliche Herausforderungen, wie die Pannenstatistik belegt [1]. Dabei wird eine Zunahme von

Batterieausfällen gezeigt.
Die ausschlaggebenden Verbraucher für eine erhöhte Bordnetzauslastung sind die Hochleistungsverbraucher wie die elektrische Lenk- und Bremsunterstützung. Sie sind für einen kurzen Zeitraum aktiv und benötigen neben den anderen Verbrauchern eine sofortige, hohe Energieversorgung aus dem Bordnetz. Diese Leistungsspitzen können je nach Grundlast und Fahrbedingungen zu erheblichen Einbrüchen der Batteriespannung führen und die Bordnetz- bzw. Spannungsstabilität gefährden. Ein Leistungsdefizit tritt im Wesentlichen auf, wenn die Ausgangsleistung des Generators nicht ausreicht, um alle Lasten zu unterstützen. Somit ist die Verwendung der Batterie zur Bereitstellung der verbleibenden Leistung erforderlich. Um Ausfälle von Steuergeräten zu vermeiden, darf die absolute Mindestspannung des Bordnetzes 9 V nicht unterschreiten [4]. Die dafür notwendigen Gegenmaßnahmen werden allgemein unter dem Begriff Leistungsmanagement zusammengefasst. Um Batterieschäden zu vermeiden und sicherheitskritische Funktionen aufrecht zu erhalten, werden bei einer geringeren Energieverfügbarkeit elektrische Verbraucher kurzzeitig degradiert oder abgeschaltet. Dadurch wird die Bordnetzlast reduziert und dem Leistungsmanagement entgegengewirkt. Das Leistungsmanagement kann als Teil des elektrischen Energiemanagements aufgefasst werden, welches neben der Bordnetzstabilität auch eine ausgeglichene Ladebilanz sowie Startfähigkeit des Fahrzeugs gewährleistet [4].

Bisherige Energiemanagement-Strategien basieren auf mathematischer Optimierung oder Regel-basierten Methoden. Diese mathematischen Verfahren sind sehr rechenaufwändig. Dahingegen dauert die Entwicklung regelbasierter Methoden sehr lange, da sie auf Expertenwissen angewiesen sind [5]. Außerdem sind sie durch das Fachwissen der Systemdesigner eingeschränkt. Somit kann auf lange Sicht keine optimale Betriebsstrategie für das elektrische Bordnetz garantiert werden [6]. Je nach Fahrzeugkategorie und Ausstattung sind Anpassungen und ein gewisser Entwicklungsaufwand erforderlich.
Die Verwendung von maschinellem Lernen im Automobilsektor hat in den letzten Jahren zugenommen. Insbesondere im Bereich Fahrerassistenzsysteme und autonomes Fahren kommt die Künstliche Intelligenz vermehrt zum Einsatz. Hierbei wurde Reinforcement Learning (RL, dt. bestärkendes Lernen) als Teilgebiet des maschinellen Lernens im Bereich Energiemanagement für HEVs bereits mehrmals erfolgreich eingesetzt [7]. RL ist zusätzlich zu mathematischen und regelbasierten Methoden ein weiterer Ansatz im Bereich des Energiemanagements, der in den letzten Jahren immer mehr Forschungsaufmerksamkeit erhalten hat. Dies ist auf die Fähigkeit von RL zurückzuführen, optimale Entscheidungen bei komplexen Kontrollproblemen zu treffen. Die Entwicklung aktueller Betriebsstrategien wird insbesondere durch die zunehmende Komplexität Bordnetze negativ beeinflusst. Die Berechnung mathematischer Lösungen wird rechnerisch intensiver. Außerdem müssen Entwickler von Regel-basierten Ansätzen alle Zusammenhänge im elektrischen System des Fahrzeugs im Überblick behalten, um geeignete Lösungen zu finden. Dadurch treten selbstlernende Systeme, die das Lernen komplexer Zusammenhänge ohne Vorwissen über alle Details eines Systems ermöglichen, immer mehr in den Vordergrund.

Nach der Erstoptimierung kann der Entwicklungsaufwand durch die Übertragbarkeit und Anpassungsfähigkeit auf verschiedene Fahrzeugkategorien und -ausstattungen reduziert werden.

1.2 Stand der Forschung

Das 12 V-Bordnetz stößt durch den vermehrten Einsatz von elektrischen Verbrauchern und dem entsprechend erhöhten Energiebedarf immer mehr an seine Belastungsgrenze. Das elektrische System eines Autos ist anfällig für Schwankungen. Bei modernen Fahrzeugen gibt es große Schwankungen der Verbraucherleistung. Dies lässt sich auf Verbraucher wie zum Beispiel die elektrische Lenkunterstützung rückführen, welche zwar nur kurzzeitig aktiv sind, aber hohe Spitzenleistungen fordern. Solche kurzzeitigen Änderungen der Last verursachen Spannungsschwankungen im Bordnetz. Diese müssen möglichst vermieden werden, weil einige Verbraucher anfällig auf solche Unstetigkeiten reagieren und entweder ausgeschaltet oder beschädigt werden können. Die Gewährleistung der Spannungsstabilität ist eine herausfordernde Aufgabe und somit ein Grund für Energiemanagementsysteme. Um auch in Zukunft trotz stetig steigender Energieanforderungen ein stabiles und robustes Bordnetz zu gewährleisten, sind neben Neuerungen in der Bordnetzarchitektur wesentliche Änderungen im Energiemanagementsystem erforderlich. Im Folgenden werden zunächst verwandte Arbeiten zum Energie- und Leistungsmanagement diskutiert. Anschließend werden unterschiedliche RL-Methoden als Lösungsansatz im Bereich Energiemanagement vorgestellt.

Im Bereich ICEV stellen Khayyam et al. [8] einen regelbasierten Ansatz vor, der die Energieeffizienz verbessert und den Kraftstoffverbrauch um etwa $5,6$ % senkt. Um das Pareto-Optimum zu finden, wählt Winter [9] eine multikriterielle Optimierungsstrategie hinsichtlich Energieeffizienz, Batteriebelastung und Spannungsstabilität. Dieses ist erreicht, wenn sich keines der Kriterien zu Ungunsten eines anderen verbessern lässt. Ähnlich wie in [10] wird die hierarchische Struktur mit dem sogenannten Viable System Model (VSM, dt. Modell lebensfähiger Systeme) der Kybernetik abgeleitet. Dabei wird lediglich die Steuerung der Energieerzeugung betrachtet. Ein Management der einzelnen Verbraucher und folglich der elektrischen Last im Bordnetz findet nicht statt.
Dahingegen demonstrieren Lehmann et al. [11], wie die Verbrauchersteuerung die Stabilität des Bordnetzes effektiv unterstützen kann. Obwohl sich ihre Forschung auf eine gemischte Bordnetztopologie aus 48 V- und 12 V-Ebene in Mild-HEVs konzentriert, können ihre Schlussfolgerungen immer noch für reine 12 V-Systeme übernommen werden. Hierbei werden Umweltsensoren verwendet, um den voraussichtlichen Strombedarf der Verbraucher vorherzusagen. Es besteht die Möglichkeit eines Spannungsabfalls, wenn der Generator oder DC-DC-Wandler diesen Strombedarf nicht bewältigen kann. Daraus werden die notwendigen Gegenmaßnahmen errechnet, zu denen das Abschalten von Verbrauchern und die Regulierung der Energieerzeugung zählen. Ein simulierter Spannungsabfall von $0,75$ V

konnte mit dieser Methode auf $0,25\ V$ verringert werden [11]. Der explizite Fokus von [4] liegt auf der Spannungsstabilität, die durch leistungskritische Zustände beeinträchtigt wird. Letzteres definiert Situationen, in denen die Batterie belastet wird, weil der Generator die vom Bordnetz benötigte Leistung nicht liefern kann [9, 11]. Kohler [4] erläutert anhand der Auswertung zahlreicher Messungen, wie der erhöhte Leistungsbedarf im Bordnetz zu Spannungseinbrüchen führt. So werden beispielsweise drei Hochleistungsverbraucher bei einem Brems- und Ausweichmanöver aktiviert. Ohne Schutzmaßnahmen können diese Spitzenströme von über 140 A einen Spannungsabfall von etwa $13,5\ V$ auf $7,2\ V$ verursachen. Laut Kohler [4] kann im Beispielszenario durch eine regelbasierte Strategie der Spannungseinbruch um $2,2\ V$ reduziert werden. Der Autor schlägt den Einsatz von Umgebungssensoren zur Vorhersage von Spannungseinbrüchen vor, um Gegenmaßnahmen frühzeitig vorherzusagen und das Energiemanagement weiter zu optimieren. Im Rahmen der Validierung wurde ein simuliertes Bordnetz mit 14 steuerbaren Verbrauchern mit einem konstanten Strom von 70 A für 2 s belastet. Im Bordnetz des Verursachers sinkt die Spannung ohne Powermanagement bei einer niedrigen Ausgangsspannung von knapp 11 V auf unter 10 V. Laut Kohler [4] lässt das Powermanagement die Spannung nur auf etwa $10,5\ V$ abfallen, vorausgesetzt, dass diese Situation 2 s im Voraus genau vorhergesagt werden kann.

Lange et al. [5] haben zur Verbesserung des Energiemanagements in Brennstoffzellenfahrzeugen (FCEV, engl. Fuel Cell Electric Vehicle) den Lenkwinkelsensor und eine Historiendatenbank verwendet. Somit konnten sie zukünftige Fahrprofile vorhersagen. Die Bewertung der Regel-basierten Betriebsstrategie mit Prognose fand nur nach dem Kriterium der Ladebilanz statt. Die Verwendung der Vorkenntnisse des Fahrprofils steigerte die Ausnutzung der Rekuperationsenergie in der Simulation um mehr als 10 % [35]. Insgesamt zeigt sich, dass Mild-HEVs und ICEVs die gleichen Management-Strategien verwenden (z.B. Verbraucherabschaltung). Die Forschung zeichnet sich durch prädiktive Mechanismen des Fahrprofils als Mittel zur Leistungssteuerung aus.

In den letzten zwei Jahren wurde viel Forschung zur Anwendung von RL auf das HEV-Energiemanagement betrieben. Biswas et al. [12] setzen einen A3C-Agenten (Asynchronous Advantage Actor-Critic) ein, um eine Betriebsstrategie zu erlernen. Dahingegen verwendet die überwiegende Mehrheit der Arbeiten DDPGs (Deep Deterministic Policy Gradients) [13, 14, 15] oder reine DQNs (Deep Q-Networks) [16, 17, 18]. Beide Algorithmen setzen Funktionsapproximatoren ein, die auf neuronale Netze basieren. Über alle Arbeiten hinweg kristallisieren sich zwei Grundformen des Netzes heraus. Beispielsweise werden in [14, 16, 18] neuronale Netze mit zwei bis vier versteckten Schichten (engl. hidden layer) verwendet, wobei jede Schicht eine identische Anzahl von Neuronen aufweist. Dahingegen setzen [12, 13, 15] auf eine pyramidenförmige Struktur mit drei versteckten Schichten und einer abnehmenden Anzahl von Neuronen.

Lee et al. [17] verwenden in ihrer Arbeit ein rekurrentes neuronales Netz (RNN, engl. Recurrent Neural Network), bestehend aus zwei vollvernetzten Schichten, die eine LSTM-

Schicht (Long Short-Term Memory, dt. langes Kurzzeitgedächtnis) einschließen. Ziel dabei ist es, die Fähigkeit von RNNs zu nutzen, um Verbindungen zwischen Zeitreihen zu erkennen. Im Vergleich zum Algorithmus der Gewinner der Vehicular Technology Society Challenge 2018 erreichen die Autoren Verbrauchsverbesserungen zwischen $0,5$ % und $1,5$ % [17]. Die anderen Studien verwenden nur reine feedforward neuronale Netze (FNN, engl. Feedforward Neural Network) mit vollständig vernetzten Schichten.
Im Bordnetz von ICEVs konzentriert sich Heimrath et al. [19] auf die Verbesserung der Energieeffizienz. Er verwendet auch DQNs, allerdings mit viel weniger Neuronen pro Schicht. Darüber hinaus schlägt Heimrath et al. [19] RARL (engl. Reflex-Augmented Reinforcement Learning), eine neuartige Methode des sicheren RL (engl. Safe-RL), als Lösung für die strengen Sicherheitsstandards in der Automobiltechnik vor. Die Grundidee besteht darin, die ausgewählte Aktion manuell durch eine sichere Alternative (Reflex) zu ersetzen wenn der Agent den sicheren Aktionsraum verlassen will. Der sichere Aktionsraum muss im Voraus durch Wissen des Entwicklers definiert werden. In Tests an einem realen Fahrzeug wurde die Verlustleistung im Vergleich zum vorherigen Energiemanagementsystem um 4 % reduziert. Die Arbeiten von Heimrath et al. [19] und anderer Autoren im Bereich der HEVs ähneln sich insofern, als dass die Belohnungsfunktion nach dem gleichen Schema aufgebaut ist. Das Ziel von RL in diesen Arbeiten ist es, den besten Kompromiss zwischen verschiedenen Zielkriterien zu finden.

Zusammenfassend kann festgestellt werden, dass die bisherigen Arbeiten im Bereich Energiemanagement überwiegend auf Deep RL setzen. Zwar liegt der Fokus fast aller Autoren auf HEVs, jedoch können Erkenntnisse beispielsweise über die Hyperparameter oder die Netzarchitektur als Startwerte für die vorliegende Arbeit dienen. Während der Recherche konnte keine verwandte Arbeit zum Thema RL mit dem Schwerpunkt des Leistungsmanagements gefunden werden. Daher muss das Ziel der aktuellen Arbeit klar von anderen Arbeiten abgegrenzt werden.

1.3 Ziele der Arbeit

Der oben aufgeführte Stand der Forschung verdeutlicht eine Lücke im Bereich der Leistungsmanagementsysteme in modernen Energiebordnetzen. Dabei geht hervor, dass die vorhandenen Ansätze zur Spannungsstabilisierung von konventionellen Bordnetzen dem klassischen Leistungsmanagement zuzuordnen sind. Die Anwendung von künstlicher Intelligenz – vor allem Reinforcement Learning – im Leistungsmanagement für 12 -Bordnetze wurde bislang nicht untersucht. Angesichts der immer komplexer werdenden Energiebordnetze stellt sich die Frage, wie KI, am effektivsten in das Leistungsmanagementsystem zur Spannungsstabilisierung eingebunden werden kann.

Ziel der vorliegenden Arbeit ist es, diese Forschungslücke zu füllen und die Eignung von RL im genannten Anwendungsbereich zu analysieren. Dazu soll eine Methodik erarbeitet

werden, welche Spannungseinbrüche im 12 V-Bordnetz unter Verwendung von Stabilisierungsmaßnahmen mit KI reduziert. Mit dem Einsatz von Simulationsmodellen für unterschiedliche Komponenten soll eine Analyse in der frühen Entwicklungsphase durchgeführt werden. Außerdem kann auf diese Art eine Parameterstudie mit verschiedenen Komponenten im Energiebordnetz stattfinden.

Für diese Zielsetzung soll ein Simulationsmodell eines 12 V-Bordnetzes in MATLAB (Matrix Laboratory) Simulink aufgebaut werden. Hierfür müssen Teilmodelle der einzelnen Komponenten im Bordnetz erstellt und zu einem Gesamtsystem zusammengefügt werden. Bei der Modellbildung muss auf eine genaue Abbildung von der Dynamik der Teilmodelle geachtet werden. Basierend auf dem Stand der Technik sollen Stabilisierungsmaßnahmen bei Spannungseinbrüchen herausgefunden werden. Auf Basis dieser Maßnahmen ist die spannungsstabile Auslegung des Bordnetzes als Optimierungsproblem zu definieren. Für die Lösung dieses Problems soll ein RL-Algorithmus gefunden und in das Gesamtbordnetzmodell implementiert werden.

Zudem sollen für das Training des Agenten kritische Belastungsszenarien definiert und mit einem Erprobungsfahrzeug durchgeführt werden. Die dabei aufgenommenen Messungen von Worst-Case-Szenarien sollen dem Agenten beibringen, welche Entscheidungen er in diesen Situationen treffen muss.

Da in der Literatur die Bestimmung von den idealen Hyperparametern des RL-Ansatzes selten zu finden ist, soll in dieser Arbeit eine detaillierte Analyse zur Definition der optimalen Parametern durchgeführt werden. Außerdem soll für die Validierung der erarbeiteten Methode eine vereinfachte Referenzstrategie implementiert werden, welche das Energiemanagementsystem des Erprobungsfahrzeuges nachbilden soll. Diese Strategie ist an die in der Literatur zu findenden Spannungsstabilisierungsmaßnahmen angelehnt.

1.4 Aufbau der Arbeit

Für die Entwicklung eines RL-Ansatzes zur Spannungsstabilisierung in konventionellen 12 V-Bordnetzen wird eine systematische Vorgehensweise benötigt. Um diese Ziele zu erreichen, soll wie folgt vorgegangen werden.

In Kapitel 2 erfolgt die Erarbeitung theoretischer Grundlagen. Dabei wird zunächst auf den aktuellen Stand des 12-V-Bordnetzes eingegangen. Nach einer detaillierten Beschreibung der einzelnen Komponenten werden anschließend die Grundlagen von RL und neuronalen Netzen erläutert.

Kapitel 3 beschäftigt sich mit der Modellierung und Simulation des Bordnetzes. Zunächst

werden verschiedene Modellansätze zur Nachbildung unterschiedlicher Komponenten beschrieben. Anschließend werden die Modelle des Generators und der elektrischen Verbraucher dargestellt. Am Ende des Kapitels wird auf die Modellierung der 12 V-Batterie eingegangen.

Aufbauend auf die Bordnetzsimulation werden in Kapitel 4 unterschiedliche Belastungsszenarien dargestellt, die einen hohen Spannungseinbruch im Bordnetz verursachen. Diese Daten werden anhand von Fahrzeugmessung mit einem Erprobungsfahrzeug erhoben und bilden die Grundlage für das Training des RL-Ansatzes. Im weiteren Verlauf des Abschnitts hilft die Analyse von den einzelnen Fahrzeugmessungen dabei, ein Verständnis für die Problematik der Bordnetzstabilität und den Einfluss von Hochleistungsverbrauchern zu entwickeln. Des Weiteren wird die Gesamtbordnetzsimulation anhand einer Fahrzeugmessung validiert.

In Kapitel 5 werden Überlegungen zu einem geeigneten RL-Algorithmus angestellt. Zunächst wird die Dimension des Zustands- und Aktionsraums definiert. Nach der Formulierung der Belohnungsfunktion erfolgt in mehreren Schritten die Verbesserung der Parametrierung des Agenten. Dabei wird besonders auf die Optimierung der Hyperparameter eingegangen. Diese wird anhand von mehreren Tests und Auswertungskriterien näher erläutert. Nach den ersten Trainingseinheiten wird auf den Einsatz von Recurrent Neural Networks eingegangen.

Die Ergebnisse des resultierenden Agenten werden in Kapitel 6 dargestellt. Dabei wird für die Nachbildung des realen Energiemanagementsystems eine Referenzstrategie vorgestellt. Anschließend wird der optimierte Agent mit der Referenzstrategie verglichen. Zum Schluss wird der RL-Agent gezielt mit unterschiedlichen Bordnetzauslastungen konfrontiert, um die RL-Strategie weiter zu evaluieren.

Abschließend gibt Kapitel 7 eine Zusammenfassung und einen Ausblick auf weiterführende Ideen.

2 Stand der Technik

In diesem Kapitel werden die Grundlagen im Bereich der Kfz-Bordnetze und deren Komponenten erläutert. Anschließend wird auf das Energiemanagementsystem eingegangen. Zum Schluss wird die Methodik des Deep Reinforcement Learning aufgezeigt.

2.1 12 V-Bordnetze

Als Bordnetz wird die Gesamtheit aller elektrischen Komponenten im Fahrzeug bezeichnet. Aufgrund der steigenden Anzahl an elektrischen Verbrauchern in Fahrzeugen wurde die Bordnetzspannung in den fünfziger Jahren von 6 V auf 12 V angehoben [4]. Je höher die Spannung, desto weniger Strom wird für die Erbringung der gleichen Leistung benötigt [2]. Das sogenannte 12 V-Bordnetz ist der Standard heutiger Kraftfahrzeuge. Die Bezeichnung 12 V-Bordnetz ist der gängige Ausdruck, welcher sich von der Sollspannung der Batterie und gleichzeitig vom Spannungsniveau bei ruhendem Motor im Bordnetz ableitet. Auch nach der Anhebung auf ein Spannungsniveau von 12 V sind die Bordnetze in modernen Fahrzeugen weiter gewachsen. Der grundsätzliche Aufbau hat sich dennoch nicht mehr verändert [4]. Dieses Kapitel 2.1 vermittelt die Grundlagen von 12 V-Bordnetzen. Dabei wird in den folgenden Unterkapiteln speziell auf die Hauptkomponenten Generator, Energiespeicher sowie die elektrischen Verbraucher genauer eingegangen.
Das elektrische System im 12 V-Bordnetz ist ein Zusammenspiel des Energiewandlers, des Energiespeichers und der elektrischen Verbraucher [20]. Die Bordnetzspannung wird dabei von der Batterie bestimmt. Bei stehendem Motor werden die elektrischen Verbraucher aus der Batterie versorgt, die sich dabei entleert. Nach dem Starten des Motors ist die Aufgabe des Erzeugers, die Verbraucher mit Energie zu versorgen und gleichzeitig eine ausgeglichene Ladebilanz der Batterie herzustellen. Seine Leistungsabgabe ist jedoch drehzahlabhängig, sodass die Batterie bei niedrigen Drehzahlen die Differenz des Stroms zwischen Verbraucher und Erzeuger decken muss. Erst wenn der vom Erzeuger erzeugte Strom größer als der Verbraucherstrom ist, wird so die Batterie mit dem Batteriestrom geladen. Die grundlegende Struktur eines 12 V-Bordnetzes ist eine Parallelschaltung aus Erzeuger, Verbrauchern und Batterie. Abbildung 2.1 stellt den Aufbau eines Ein-Batterie-Bordnetzes dar [20]. Die Abbildung nimmt eine örtliche Trennung der Verbraucher zwischen Innenraum, Kofferraum und Motorraum vor. Im vorliegenden Fall befindet sich die Batterie im Motorraum, wobei je nach Fahrzeug auch andere Einbaulagen existieren. Der Starter ist vereinfacht als einziger elektrischer Verbraucher im Motorraum aufgeführt und hat mit Abstand den höchsten Strombedarf von ca. $300-500\ A$. Diese hohe Belastung ist jedoch zeitlich auf den Startvorgang begrenzt. Weitere mögliche Verbraucher im Motorraum sind beispielsweise die Zündspulen oder elektronische Einspritzung. Zusätzlich sind in Abbildung 2.1 verschiedene Leitungswiderstände R_{LS}, R_{L1}, R_{L2} und R_{L3} aufgeführt. Der Leitungswiderstand ist proportional zur Leitungslänge und somit über die Distanz vom Generator im Motorraum

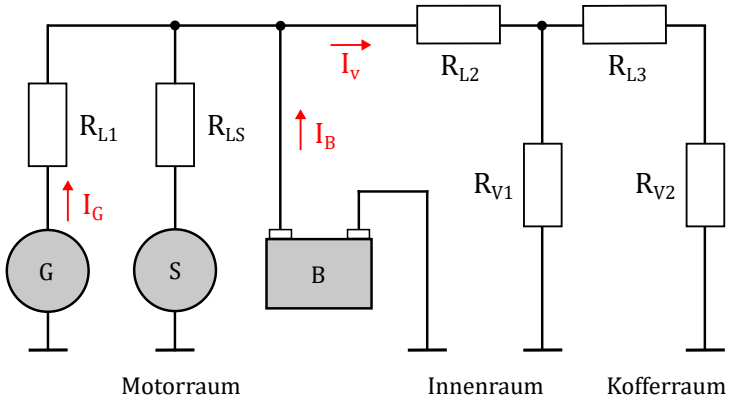


Abbildung 2.1: Schematischer Aufbau eines konventionellen 12 V-Bordnetzes: Es besteht aus einer Parallelschaltung von den Komponenten Generator, Starter, Batterie und den elektrischen Verbrauchern R_{V1} sowie R_{V2}. Zudem werden noch die Leitungswiderstände R_{L1}, R_{LS}, R_{L2} und R_{L3} dargestellt. Adaptiert aus [20].

zu Verbrauchern im Kofferraum am höchsten [20]. Durch die Leitungswiderstände entsteht eine Verlustleistung. Sie sollten daher so gering wie möglich gehalten werden [2]. Die Leistungsaufnahme der Verbraucher wird zusammengefasst durch die Widerstände R_{V1} und R_{V2} repräsentiert. Generell kann zwischen Generatorstrom I_G, Verbraucherstrom I_V und Batteriestrom I_B mit der Knotenregel folgender Zusammenhang aufgestellt werden [20]:

$$I_G = I_V + I_B \tag{2.1}$$

Der Verbraucherstrom ist der Summenstrom der Verbraucher, die hier als Widerstände R_{V1} und R_{V2} dargestellt sind. Aufgrund der kurzen Eingriffszeit findet der Motorstarter hier keine Beachtung. Zusätzlich zu Ein-Batterie-Bordnetzen existieren auch Zwei-Batterien-Bordnetze am Markt, die jedoch in dieser Arbeit nicht betrachtet werden. Im Nachfolgenden soll auf die Anforderungen an das 12 V-Bordnetz eingegangen werden. Anschließend werden die einzelnen Komponenten des 12 V-Bordnetzes näher erläutert.

2.1.1 Anforderungen an das 12V-Bordnetz

Eine der wichtigsten Anforderungen an das Energiebordnetz ist die problemlose und ununterbrochene Erfüllung von Fahrzeug- und Kundenfunktionen. Aus Sicht des Herstellers sollen dazu noch alle Komponenten im Bordnetz möglichst günstig entwickelt und produziert werden. Zusätzlich sind ein geringes Gesamtgewicht und eine gute Gewichtsverteilung gewünscht, um eine bessere Effizienz und Fahrdynamik zu erzielen. Der Bauraum und die Integration von den Komponenten im Fahrzeug sind weitere Aspekte, die betrachtet werden müssen. Manche dieser Anforderungen stehen im Widerspruch zueinander und müssen bei der Systemauslegung gegeneinander abgewägt werden. Abhängig davon, welche Fahrzeugeigenschaften im Fahrzeugkonzept priorisiert sind, müssen entsprechende Kom-

promisse gemacht werden. Die wichtigsten Aspekte sind wie folgt [21]:

- Der Verbrennungsmotor muss im Rahmen von bestimmten Grenzen gestartet werden können (ausreichende Leistungsfähigkeit der Batterie und begrenzter Ruhestrom).
- Positive Ladebilanz muss während der Fahrt durch den Generator gewährleistet sein. Daher ist der Generator sowohl für die Versorgung der elektrischen Verbraucher als auch für das Nachladen der Batterie während der Fahrt verantwortlich.
- Die Batterie muss bei Motorstillstand sicherstellen, dass Standardverbraucher eine bestimmte Zeit versorgt werden können.
- Die Spannung muss immer in den erforderlichen Grenzen bleiben, sodass die Funktion des aktivierten Verbrauchers nicht beeinträchtigt wird.
- Defekte Lasten sind sicher und schnell vom Bordnetz zu trennen, ohne den Betrieb weiterer Systeme zu beeinträchtigen.

2.1.2 Erzeuger

Als Erzeuger werden alle Komponenten bezeichnet, die das Bordnetz mit elektrischer Energie versorgen. In den konventionellen Fahrzeugen stellt der Generator - umgangssprachlich als Lichtmaschine bezeichnet - die Energieversorgung im elektrischen Bordnetz während der Fahrt sicher. Er ist mechanisch mit der Kurbelwelle gekoppelt und erzeugt einen drehzahlabhängigen Strom. Die Gleichstrommaschinen, die früher meist im Einsatz waren, wurden bisweilen durch Drehstrommaschinen ersetzt. Sie sind als Synchronmaschinen ausgeführt und besitzen einen Elektromagneten im Rotor, der das sogenannte Erregermagnetfeld erzeugt [2]. Ein Vorteil gegenüber Permanentmagneten ist, dass die Stärke des Magnetfeldes durch die Höhe des Erregerstroms beeinflusst werden kann. Dadurch ist gleichzeitig die Regelung der induzierten Spannung in den im Stator befindlichen Wicklungen möglich. Der in die Erregerwicklung eingeprägte Strom kann entweder aus einem Energiespeicher wie der Fahrzeugbatterie (Fremderregung) oder vom erzeugten Generatorstrom selbst (Selbsterregung) abgezweigt werden. Während der Motor steht und die Zündung des Fahrzeugs eingeschaltet ist, wird die Erregerwicklung durch die Batterie bestromt und somit fremderregt, da der Generator hier noch keinen Strom erzeugt. Die Selbsterregung erfolgt erst nach Starten des Motors und ausreichend hohem Generatorstrom. Der Einsatz von Drehstrommaschinen erfordert ein Gleichrichten der dreiphasigen Wechselspannung, sodass eine elektrische Kopplung zum 12 V-Bordnetz des Fahrzeugs stattfinden kann [22]. Hierfür wird eine B6-Schaltung (dreiphasiger Brückengleichrichter) genutzt und die resultierende pulsierende Gleichspannung durch die Kondensatoren geglättet [2]. Die gängigen Generatoren erzeugen je nach Fahrzeugklasse Ströme zwischen 90 A und 210 A [23].
Die Generatorspannung ist zusätzlich abhängig von der Drehzahl des Generators, die während der Fahrt je nach Geschwindigkeit und Gangwahl variiert. Um die Bordnetzspannung konstant zu halten, wird der Erregerstrom so geregelt, dass die Generatorspannung

über das gesamte Drehzahlband möglichst konstant einem Sollwert folgt. Dieser Wert ist abhängig davon, ob die Batterie geladen, entladen oder Ladezustand gehalten werden soll. Für einen Ladestrom von Generator zu Batterie muss die Generatorspannung zum Beispiel so viel größer als die Batteriespannung sein, dass der Generatorstrom den Verbraucherstrom übersteigt. Die Differenz ist nach Formel 2.1 der positive Ladestrom, der zum Schutz der Batterie temperaturabhängig begrenzt wird. Zum Entladen muss der Strom dementsprechend ein negatives Vorzeichen haben. Um den Ladestand zu halten wird $I_B = 0A$ gefordert. Der Generator muss demnach den Strombedarf der Verbraucher möglichst genau decken. Die Regelungszyklen laufen im Millisekundenbereich ab [22]. Da der Generator allerdings ein träges Verhalten aufweist, kann er Leistungsspitzen der Verbraucher nicht ausregeln [4].
Der Generator ist über einen Riemenantrieb mit dem Verbrennungsmotor gekoppelt. Deswegen ist der Ausgangsstrom vom Generator von der Motordrehzahl abhängig. Diese Abhängigkeit ist in Abbildung 2.2 dargestellt. Wenn der Generatorstrom I_G größer als der Verbraucherstrom I_V ist, versorgt der Generator die Verbraucher und lädt bei Bedarf auch die Batterie. Umgekehrt, muss die Differenz von der Batterie abgedeckt werden. Somit wird die Batterie entladen.

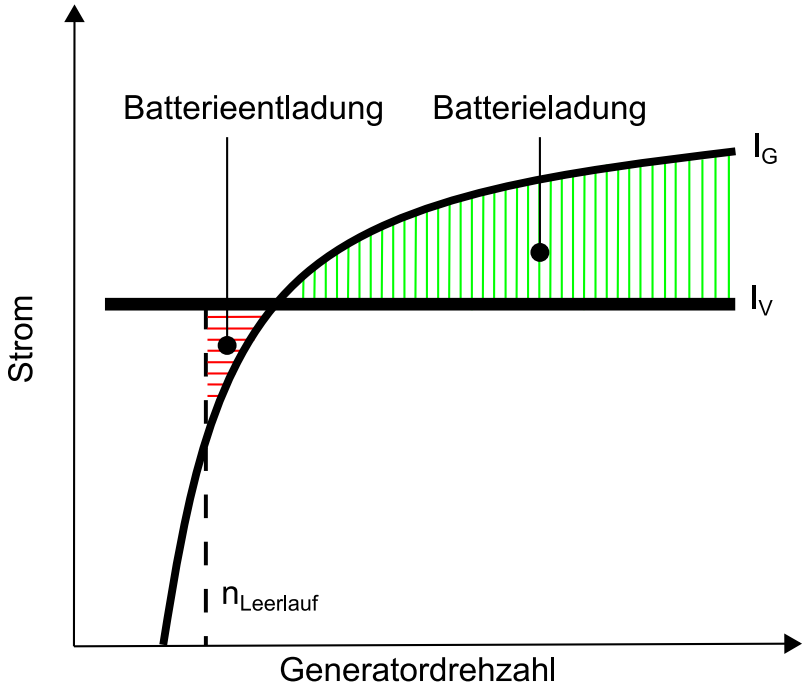


Abbildung 2.2: Darstellung des Ausgangsstroms vom Generator I_G in Abhängigkeit von der Generatordrehzahl. Ist der Generatorstrom I_G größer als der Verbraucherstrom I_V, werden die elektrischen Verbraucher versorgt und die Batterie geladen. Umgekehrt, kommt es zu einer Entladung der Batterie. Adaptiert aus [22].

Aufgrund der großen Erregerkreiszeitkonstante kann der Generator nicht jedem dynamischen Lastwechsel schnell genug folgen [24]. Bei einem plötzlichen Lastanstieg im Bord-

netz entsteht kurzzeitig eine Unterspannung. Analog dazu kommt es bei einem plötzlichen Lastabwurf zu einer Überspannung am Generator. Diese Lastwechsel können bei niedrigen Drehzahlen zu abrupten Momentstößen führen und somit ein Abwürgen des Verbrennungsmotors verursachen [25, 26, 27]. Aus diesem Grund ist die Dynamik des Generators bei niedrigen Drehzahlen zusätzlich künstlich reduziert. Diese Begrenzung wird auch mit dem Begriff "Load-Response Funktion"bezeichnet [28].
Der Generator belastet den Motor über einen Riemenantrieb mit einem zusätzlichen Drehmoment von bis zu 60 Nm. Dadurch steigt der Kraftstoffbedarf und CO_2-Ausstoß mit der abgerufenen Leistung des Generators. Um dem entgegenzuwirken hat beispielsweise die BMW Group eine intelligente Generatorregelung (iGR) eingeführt, die die Ladestrategie der Batterie um eine Rückgewinnung der Schub- und Bremsenergie erweitert. Statt die Batterie jederzeit möglichst voll zu laden, wird nur noch in „energetisch günstigen Fahrphasen“ [23, S. 10] geladen. Diese zeichnen sich dadurch aus, dass kein Kraftstoff verbraucht wird. Die während dieser Schubphasen (z.B. Ausrollen bei eingelegtem Gang) in der Batterie gespeicherte Energie wird in Zugphasen (z.B. Beschleunigen mit Motorkraft) genutzt und muss folglich nicht vom Generator geliefert werden. Diese Art der Energieverteilung sorgt laut Frickenstein et al. [23] je nach äußeren Bedingungen und Fahrzyklus für einen um bis zu 4 % geringeren Kraftstoffverbrauch.

2.1.3 Batterie

Das Bordnetz in konventionellen Fahrzeugen mit Verbrennungsmotor benötigt einen Energiespeicher für die Zeit, in der der Generator die elektrischen Verbraucher nicht mit Energie versorgen oder den Energiebedarf nicht vollumfänglich decken kann. Als Energiespeicher hat sich eine wiederaufladbare Batterie durchgesetzt, die auch als Akkumulator bezeichnet wird. Neben der Energieversorgung der Verbraucher bei stehendem Motor ist das Starten des Motors eine weitere Hauptaufgabe der Batterie. Sie wird daher auch Starterbatterie genannt. Demgegenüber stehen die sogenannten Traktionsbatterien, die beispielsweise in rein batteriebetriebenen Elektrofahrzeugen (BEV, engl. Battery Electric Vehicle) den Strom für den elektrischen Fahrantrieb zur Verfügung stellen. Diese werden nur aus Gründen der Vollständigkeit erwähnt, finden aber in dieser Arbeit keine weitere Betrachtung. Die Batterie unterliegt verschiedensten Anforderungen, die sich wie folgt zusammenfassen lassen:

- hohe Energiedichte (geringes Volumen und Gewicht)
- geringe Selbstentladung
- hoher Wirkungsgrad
- hohe Temperaturbeständigkeit
- geringer Wartungsaufwand
- je nach Einsatzzweck hohe Rüttelfestigkeit (z.B. Geländewagen), Zyklusfestigkeit (z.B. wegen Start-Stopp-Betrieb) [2].

Die Auslegung der Batterie und des Generators erfolgen in Abhängigkeit der installierten Bordnetzleistung. Typische Batteriekapazitäten liegen zwischen 42 Ah bei Kleinstwagen und 93 Ah in der Oberklasse [4].
Der Prozess des Ladens und Entladens geht auf chemische Reaktionen in den Zellen einer Batterie zurück. Da eine elektrochemische Zelle nicht die Spannung erzeugen kann, die im Fahrzeug benötigt wird, werden mehrere Zellen in Reihe geschaltet. Die Spannung einer aufgeladenen Zelle hängt von der chemischen Reaktion bzw. Zusammensetzung der Elektroden ab und variiert von ca. 1,3 V bei Nickel-Metallhydrid-Batterien (NiMH) und 3,7 V bei Lithium-Ionen-Batterien (Li-Ionen) [2]. Beide Systeme finden meist Anwendung in Hybridfahrzeugen [22]. Für konventionelle 12 V-Bordnetze haben sich Blei-Batterien etabliert, deren Zellen eine Nennspannung von 2 V aufweisen. Die Zellspannung ändert sich abhängig vom Ladezustand und der Temperatur. Um eine Gesamtspannung von 12 V zu erreichen, werden 6 Zellen in Reihe benötigt. Bei zu hoher Ladespannung einer Zelle kann es zum temperaturabhängigen Effekt der Gasung kommen, wobei Wasserstoff freigesetzt wird. Dies sollte durch ein entsprechendes Management der Lade- und Entladeprozesse verhindert werden, da sich durch die Reaktion des Wasserstoffs mit dem Luft-Sauerstoff explosives Knallgas bilden kann [2]. Ein großer Nachteil von Bleiakkumulatoren ist die schnelle Alterung bei häufigen Lade- und Entladezyklen. Als Abhilfe können AGM- (engl. Absorbent Glass Mat) oder Blei-Gel-Batterien dienen, bei denen der Elektrolyt in einem Glasfaservlies oder Gel gebunden wird, um den Alterungseffekt zu mindern. Neben der hohen Zyklusfestigkeit weisen diese Batterietypen einen sehr geringen Wartungsaufwand auf. Da Blei schädlich für die Umwelt ist, müssen die Batterien konsequent recycelt werden [22].

2.1.4 Elektrische Verbraucher

Elektrische Verbraucher sind alle Komponenten im Fahrzeug, welche vom Bordnetz mit Strom versorgt werden müssen. Zu den Anfangszeiten des Automobils waren der Starter und die Beleuchtungen die einzigen Verbraucher im Fahrzeug. Im Laufe der Zeit wurde das Bordnetz stets um weitere Verbraucher erweitert. Die Anzahl der elektrischen Verbraucher in einem 12 V-Bordnetz schwankt je nach Fahrzeugklasse sowie Ausstattung zwischen 60 und 100 Stück. Diese Steigerung hat nicht nur eine steigende Belastung des Bordnetzes zur Folge, sondern auch eine anwachsende Komplexität im Bordnetz.
Eine der Gründe für die steigende Anzahl der elektrischen Verbraucher ist das steigende Spektrum an Komfortausstattungen in den Fahrzeugen [29, 30, 31]. Der Konkurrenzkampf zwischen den Herstellern führt dazu, dass die verfügbaren Komfortoptionen zunehmen. Dazu zählen zum Beispiel elektrisch verstellbare Sitze, Sitzheizungen und -belüftungen und elektrische Schiebedächer.
Der Sicherheitsaspekt ist ein weiterer Grund und spielt für Autofahrer eine immer größere Rolle. Daher versuchen die Hersteller mit verschiedensten Sicherheitsoptionen Kunden für sich zu gewinnen. Diese Sicherheitssysteme müssen ebenso elektrisch versorgt wer-

Tabelle 2.1: Kategorisierung der elektrischer Verbraucher nach Einsatzzeit, Zielsetzung und Leistungsbedarf. Dabei wird die Einsatzzeit in Dauer, Kurz und Lang unterteilt. Bei der Zielsetzung werden die elektrischen Verbraucher in die Kategorien Komfort, Grundlast und Sicherheit aufgeteilt.

Verbraucher	**Einsatzzeit**	**Zielsetzung**	**Leistungsbedarf**
Gebläse/Klima	Dauer	Komfort	100...500 W
Motormanagement	Dauer	Grundlast	175...200 W
Elektro-Kraftstoffpumpe	Dauer	Grundlast	250 W
Elektrische Lenkung	Kurz	Sicherheit	1500 W
Autoradio	Lang	Komfort	15...30 W
Abblendlicht	Lang	Sicherheit	je 55...60 W
Elektrisches Kühlergebläse	Lang	Grundlast	200...800 W
Bremsleuchte	Kurz	Sicherheit	je 18...21 W
Motorstarter	Kurz	Grundlast	Ottomotor: 700...2000 W Dieselmotor: 1400...2600 W
Sitzheizung	Kurz	Komfort	je 100...200 W
Heckscheibenheizung	Kurz	Komfort	120...200 W
Lenkradheizung	Kurz	Komfort	50 W

den. Wichtige Fahrzeugfunktionen, wie Bremsen, Reifen, Pumpen oder Lüfter müssen zusätzlich durch Elektronik überwacht werden. Assistenzsysteme wie das Antiblockiersystem (ABS), Dynamic Stability Control (DSC) oder Electronic Power Steering (EPS) zählen ebenfalls zu den sicherheitsrelevanten Verbrauchern.

Der abgefragte Strombedarf im Bordnetz schwankt zwischen einem Ruhestrom von wenigen Milliampere (z.B. Diebstahlwarnanlage bei verschlossenem Fahrzeug) und einem Spitzenbedarf ca. 300 – 500 A beim Motorstart. Gerade für das Starten des Motors ändert sich der Strombedarf bei kaltem Motor um bis zu Faktor zwei [22]. Der mittlere Strombedarf während der Fahrt beträgt laut Reif [22] zwischen 20 A und 70 A. Gerade in Fahrzeugen der Mittel- und Luxusklasse wird sich diese Zahl zwischenzeitlich deutlich erhöht haben.

Die elektrischen Verbraucher lassen sich in verschiedene Kategorien hinsichtlich ihrer mittleren Einsatzzeit und Zielsetzung klassifizieren. Die Leistungsangaben der einzelnen Verbraucher variieren in der Literatur. Für die oben dargestellte Tabelle 2.1 wurden die Angaben für einen PKW (Personenkraftwagen) aus den Quellen [1, 2, 20, 22, 32] zusammengeführt. Die Tabelle hat keinen Anspruch auf Vollständigkeit und bildet nur einen Teil der Verbraucher beispielhaft ab.

In der Kategorie der Einsatzzeit wird zwischen kurzer und langer Einschaltdauer sowie Dauerverbrauchern unterschieden. Die Zielsetzung des Verbrauchers kann in die Kategorien Komfort, Grundlast und Sicherheit eingeteilt werden. Grundlastverbraucher bezeichnen jene Verbraucher, die elementar wichtig für die Grundfunktion des Fahrzeugs sind [33]. Alle sicherheitsrelevanten Verbraucher können damit auch als Grundlast kategorisiert werden.

Aus der Tabelle wird ersichtlich, dass der Leistungsbedarf der Verbraucher stark unterschiedlich ist. Nicht jedes Fahrzeug ist mit allen aufgezählten Verbrauchern ausgestattet. Außerdem hängt die Nutzung der Verbraucher von der Außentemperatur und der Fahrzeit ab. Im Winter beispielsweise werden Heizungen öfter beansprucht als im Sommer. Je länger die Fahrzeit wird, desto eher wird die Heizleistung stufenweise durch die Insassen reduziert und der Leistungsbedarf sinkt. Im Sommer hingegen muss der Motor im Stau gekühlt werden, was möglicherweise zum Zuschalten von Zusatzlüftern führt, um den Motor zu kühlen. All diese Einflüsse sorgen dafür, dass einige Szenarien rein aus der Sicht des Leistungsbedarfs im Bordnetz kritischer sind als andere [20].

2.2 Elektrische Energiemanagementsysteme (EEMS)

Die steigende Anzahl der elektrischen Verbraucher ist eine wachsende Herausforderung in den immer komplexer werdenden Energiebordnetzen [34]. Bei den modernen Fahrzeugen reicht die Nennleistung der Generatoren nicht mehr aus, um alle Verbraucher gleichzeitig zu versorgen [1]. Das weltweit steigende Umweltbewusstsein und die daraus resultierenden Umweltauflagen zwingen die Automobilindustrie effizientere Fahrzeuge zu entwickeln. Die daraus entstehenden Aufgaben übernehmen sogenannte elektrische Energiemanagementsysteme (EEMS), die in den letzten Jahren immer näher an ihre Grenzen stoßen. Voraussetzung für ein erfolgreiches Energiemanagement ist eine Auslegung der Bordnetzkomponenten wie Batterie und Generator entsprechend der installierten Leistung [20]. Die ursprüngliche Aufgabe von EEMS war, „zu jedem gewünschten Zeitpunkt elektrische Energie in der gewünschten Menge und Güte“ [35] zur Verfügung zu stellen. Die oberste Priorität des Bordnetzes und somit des EEMS ist die Gewährleistung des Motorstarts. Auch der Betrieb einiger Verbraucher bei Motorstillstand muss für längere Zeit gewährleistet werden. Voraussetzung dafür ist, dass die Ladebilanz der Batterie mindestens ausgeglichen oder positiv sein muss [20]. Die Ladebilanz gilt als ausgeglichen, wenn ein Ladezustand gehalten und als positiv, wenn die Batterie nachgeladen wurde. Die Erfassung des Ladezustandes ist für die Verwaltung der Ladebilanz erforderlich. Dieser wird durch den State of Charge (SOC) beschrieben und durch das Verhältnis aus aktueller und maximaler Ladungsmenge definiert [2]:

$$SOC = \frac{Q_{ist}}{Q_{max}} \tag{2.2}$$

Um die maximale Ladungsmenge Q_{max} zu ermittelt, wird eine vollgeladene Batterie mit 20 % des Nenn-Entladestroms bis zu einer Minimalspannung von 10,5 V entladen [20]. Weitere Batterieparameter sind der State of Health (SOH), der die Restlebensdauer bzw. den Alterungsgrad angibt sowie der State of Function (SOF), der die Leistungsfähigkeit unter Berücksichtigung des Ladezustands und Alterungsgrads der Batterie bewertet. Anhand der Generatorleistung kann im Fahrbetrieb die Ladebilanz beeinflusst werden, wobei die temperaturabhängige Ladefähigkeit der Batterie zu berücksichtigen ist. Im Leerlauf können

moderne Generatoren etwa ein Drittel ihrer Leistung abrufen [36]. Daher haben Automobilhersteller die Funktion, die Leerlaufdrehzahl des Generators anzuheben, eingebaut. Durch die erhöhte Drehzahl kann der Generator mehr Strom erzeugen. Dies ist generell nur beim Stillstand des Fahrzeugs im ausgekuppelten Zustand möglich, da während der Fahrt die Motordrehzahl über Getriebe, Differential und Räder fest an die Fahrtgeschwindigkeit gebunden ist. Eine erhöhte Leerlaufdrehzahl führt jedoch zu höheren Emissionen und steht den Forderungen eines immer niedrigeren Schadstoffausstoßes gegenüber [33]. Daher muss ein Kompromiss zwischen Effizienz und Ladebilanz gemacht werden. Die Anhebung der Leerlaufdrehzahl reicht nicht aus, um den Leistungsbedarf der Verbraucher bei ungünstigen Umgebungsbedingungen zu decken. Laut Fabis [32] ist es zum Beispiel im Winter unwahrscheinlich, dass die Batterie in den ersten 10 Minuten eines Fahrzyklus mit viel Leerlaufanteil geladen wird, da viele Heizungen eingeschaltet sind und die Generatorleistung durch die geringen Drehzahlen begrenzt ist. Das EEMS muss dafür sorgen, die Batterie in günstigeren Fahrsituationen wieder zu laden.

Das Fahrzeug so effizient wie möglich zu machen, ist eine weitere Aufgabe des EEMS. Nach Reif [20] kann eine Erhöhung der Leistungsaufnahme der Verbraucher um 100 W einen Mehrverbrauch von etwa $0,17$ l je 100 km zur Folge haben. Dies wiederum führt zu höheren Emissionswerten. Die intelligente Steuerung der Energieflüsse im Bordnetz durch das EEMS hat somit einen großen Einfluss auf den Umweltaspekt. Dies wirkt sich einerseits, wie bereits in Kapitel 3.1 erwähnt, auf die Regelung der Generatorleistung aus, andererseits aber auch auf die bedarfsgerechte Steuerung von Nebenaggregaten. Laut Ennemoser et al. [37] bietet die Betriebsstrategie für Nebenaggregate bei LKWs (Lastkraftwagen) beispielsweise 3–4 % Einsparpotenzial beim Kraftstoffverbrauch. Daraus lässt sich ebenso ein Potenzial für PKWs ableiten, auch wenn dieses möglicherweise durch eine geringere Anzahl an Nebenaggregaten geringer ausfällt. Auch Start-Stopp-Systeme können die Effizienz verbessern, indem sie Leerlaufverluste vermeiden. Beim Abstellen des Motors muss allerdings die erneute Startfähigkeit (z.B. durch eine Bewertung mit dem SOF) gewährleistet sein [20].

Die Sicherstellung der Bordnetzstabilität bildet ein weiteres Ziel von EEMS. Durch Leistungsspitzen kann die Bordnetzspannung in kritischen Situationen einbrechen. Die Tiefe des Spannungseinbruchs ist stark temperaturabhängig [32]. In Abbildung 2.3 werden die Spannungsschwellen des 12 V-Bordnetzes abgebildet. Für den Motorstart werden mindestens 6 V benötigt. Fällt die Spannung beim Starten des Motors drunter, liegt es entweder an einem zu niedrigen Ladezustand oder einem stark erhöhten Innenwiederstand der Batterie. Große Belastungen der Batterie sollten auch während der Fahrt keine Spannung unter 9 V verursachen. Die typische Batteriespannung beim Entladen sollte ungefähr zwischen 11 V und 12 V liegen.

Um zu verhindern, dass Steuergeräte durch eine zu niedrige Spannung ausfallen, ergreift das EEMS Gegenmaßnahmen. Typisch in diesen Situationen ist die Degradierung oder gar Abschaltung von elektrischen Verbrauchern [20]. Sicherheitskritische und für die Grundfunktion relevante Verbraucher (z.B. elektrische Lenksysteme, Blinker oder Motormana-

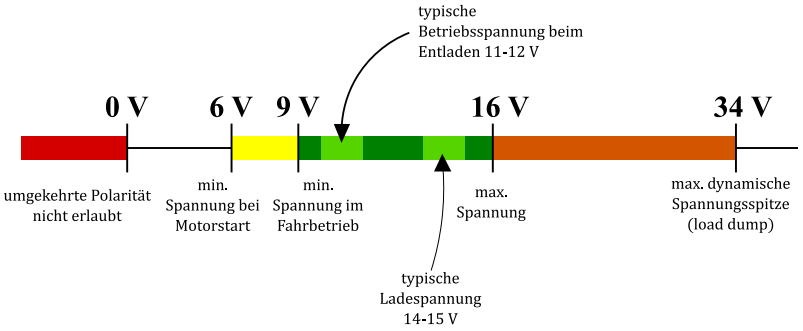


Abbildung 2.3: Spannungsschwellen im Bordnetz: Für den Motorstart werden mindestens 6 V benötigt. Während des Fahrbetriebes wird die Batterie mit einer typischen Ladespannung von $14 - 15$ V geladen. Die Betriebsspannung beim Entladen beträgt $11 - 12$ V. Adaptiert aus [38].

gement) sind von einer solchen Strategie ausgeschlossen. Die Wirkung dieses Instruments ist abhängig von dem in Kapitel 2.1.4 beschriebenen Leistungsbedarf und dem Einschaltmuster der Verbraucher. Es kann nur die Last im Bordnetz abgeworfen werden, die zuvor eingeschaltet war. Folgende Tabelle 2.2 fasst die Instrumente der EEMS in konventionellen Fahrzeugen zusammen [20].
Das Ziel von EEMS ist zusammengefasst, unter Anwendung der genannten Instrumente die Fahrzeugverfügbarkeit und -sicherheit zu optimieren und gleichzeitig den Komfort für die Nutzer so wenig wie möglich und unbemerkt einzuschränken [20]. Heutzutage erfolgt die Umsetzung basierend auf einem komplexen, auf die Fahrzeugausstattung abgestimmten Regelwerk. Dieses bestimmt die Betriebsstrategie im elektrischen Bordnetz [6].

Tabelle 2.2: Übersicht über die Instrumente, Auswirkungen und Zielsetzung eines EEMS.

Instrument	**Auswirkung**	**Zielsetzung**
Start-Stopp-System	Vermeidung von Leerlaufverlusten	Effizienz
Elektrifizierung von Nebenaggregaten	Bedarfsgerechte Steuerung	Effizienz
Degradierung/Abschaltung von Verbrauchern	Lastreduktion	Bordnetzstabilität
Generatorregelung	Situativ angepasste Generatorleistung	Ladebilanz, Effizienz, Bordnetzstabilität
Leerlaufdrehzahlerhöhung	Verbesserter Betriebspunkt des Generators	Ladebilanz

2.3 Reinforcement Learning (RL)

Maschinelles Lernen ist ein bedeutendes Feld der künstlichen Intelligenz, das sich mit der Entwicklung von Algorithmen und Modellen befasst. Diese ermöglichen es Computern aus Erfahrungen zu lernen und Aufgaben zu bewältigen, ohne explizit programmiert zu werden. In den letzten Jahren hat das maschinelle Lernen erhebliche Fortschritte gemacht und zahlreiche Anwendungen in verschiedenen Bereichen wie Bildererkennung, Sprachverarbeitung, Robotik, medizinische Diagnose und Finanzwesen gefunden [39, 40]. Das Feld des maschinellen Lernens umfasst verschiedene Teilbereiche, die im Folgenden näher beschrieben werden.

Reinforcement Learning (RL, dt. bestärkendes Lernen) ist ein maschinelles Lernparadigma, das sich von traditionellen Lernmethoden wie Supervised Learning (dt. überwachtes Lernen) und Unsupervised Learning (dt. unüberwachtes Lernen) unterscheidet. Generell definiert maschinelles Lernen den Prozess, automatisiert Zusammenhänge in Daten zu erkennen und diese zum Lösen einer Aufgabe zu nutzen [41]. Die drei genannten Kategorien unterscheiden sich dabei in der Art und Weise, wie dieses intelligente Verhalten durch ein künstliches System gelernt und optimiert wird. Supervised Learning ist ein zentraler Teilbereich des maschinellen Lernens, der auf externes Wissen – den Labels (Kennzeichnung der gesammelten Daten) – basiert [42]. Es zielt darauf ab, eine Abbildung von Eingabe zu Ausgabe zu erlernen, um in der Lage zu sein, für neue, nicht-gesehene Eingabedaten die entsprechenden Ausgaben vorherzusagen. Dies geschieht durch das Lernen einer generalisierten Funktion, die das zugrundeliegende Muster in den Daten erfasst [39]. Beim Unsupervised Learning hingegen wird auf die Verwendung von gelabelten Ausgabedaten verzichtet. Hier wird stattdessen nach intrinsischen Strukturen und Mustern in den Daten gesucht [43]. Das Ziel besteht darin, die Daten zu gruppieren, Dimensionalitätsreduktion durchzuführen oder andere latente Strukturen zu entdecken [39, 44]. In beiden Kategorien spielen Auswirkungen einer Entscheidung auf das zukünftige Systemverhalten keine Rolle oder werden nicht berücksichtigt. RL dagegen beschreibt das Lernen einer Entscheidungssequenz in einem zeitlichen Kontext und basiert auf dem Belohnungsprinzip (engl. reward hypothesis). Der sogenannte Agent kann als eine Softwarekomponente betrachtet werden, welche Aktionen in einer Umwelt ausführen kann, um die vom Benutzer vordefinierten Ziele zu erreichen. Ähnlich dem Training eines biologischen Wesens kann das Training eines Agenten im Kontext des RL als zielgerichtetes Lernen aus Versuch und Irrtum (engl. trial and error) betrachtet werden [42]. Entsprechend den durchgeführten Aktionen erhält der Agent in jeder Situation entweder Belohnungen oder Strafen von seiner Umwelt. Das Ziel ist eine möglichst optimale Verhaltensstrategie (engl. policy) zu erlernen, mit welcher der Agent die Folgeaktionen situationsbedingt so wählt, dass die Summe der zukünftigen Belohnungen (engl. rewards) aus der Umwelt maximiert wird [42]. Dieses Grundprinzip lässt sich auf jede Art von auf vergangenen Aktionen basierenden konsekutiven Entscheidungen im Kontext einer größeren Problemstellung anwenden [45]. Ein

derartiges Entscheidungsproblem kann durch einen Markov-Decision-Process (MDP, dt. Markov-Entscheidungsprozess) modelliert werden, dieser bildet die Grundlage eines RL-Problems [41].

2.3.1 Markov-Decision-Process (MDP)

Der Markov-Decision-Process ist ein mathematisches Modell, das die Interaktion eines RL-Agenten mit seiner Umgebung formalisiert. Wie bereits erwähnt, können RL-Verfahren verwendet werden, um sequentielle Entscheidungsprobleme zu lösen. Die Basis zur Modellierung eines RL-Problems bildet der zeitdiskrete stochastische Markov-Decision-Process (MDP, dt. Markov-Entscheidungsprozess) bestehend aus dem Tupel $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, T, R, \gamma \rangle$, wobei

- $\mathcal{S}$ den Zustandsraum mit allen möglichen Zuständen $s \in \mathcal{S}$,
- $\mathcal{A}$ den Aktionsraum mit allen möglichen Aktionen $a \in \mathcal{A}$,
- $T\colon \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ die Transitionsfunktion zwischen Zuständen als bedingte Wahrscheinlichkeit,
- $R\colon \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$ die skalare Belohnungsfunktion als Abbildung der Zustände und Aktionen auf eine reelle Zahl und
- $\gamma \in [0, 1]$ den Diskontierungsfaktor beschreibt [46].

Im Folgenden wird immer von einem voll beobachtbaren (engl. fully observable) MDP ausgegangen, sodass die Beobachtungen $w_t \in \mathcal{W}$ des Agenten zu allen diskreten Zeitpunkten t alle Zustände $s_t \in \mathcal{S}$ der Umwelt umfassen [45]. Außerdem wird zunächst von diskreten Zustands- und Aktionsräumen ausgegangen. Für kontinuierliche Zustands- und Aktionsräume ändern sich die betreffenden Summenzeichen in Integrale.
MDPs können als Erweiterung von Markov-Reward-Processes (MRP, dt. Markov-Belohnungsprozess) um eine Entscheidung in Form einer Aktion angesehen werden. MRPs wiederum beschreiben einen einfachen Markov-Process (MP, dt. Markov-Prozess) mit einer zusätzlichen Wertung [44]. Dessen Grundlage ist die Markov-Bedingung. Sie sagt aus, dass die Wahrscheinlichkeit für jeden möglichen Folgezustand s_{t+1} nur vom aktuellen Zustand s_t und der gewählten Aktion a_t abhängt, nicht aber von allen vorangegangenen Zuständen und Aktionen $\{s_{t-1}, a_{t-1}, ..., s_0, a_0\}$ [47]:

$$P(s_{t+1}|s_t, a_t, s_{t-1}, a_{t-1}, ..., s_0, a_0) = P(s_{t+1}|s_t, a_t) = T(s_t, a_t, s_{t+1}) \tag{2.3}$$

Die Transitionsfunktion $T(s_t, a_t, s_{t+1})$ beschreibt die Wahrscheinlichkeit der möglichen Folgezustände s_{t+1} in Abhängigkeit des aktuellen Zustands $s_t \in \mathcal{S}$ und der gewählten Aktion $a_t \in \mathcal{A}$, wobei $\sum_{s_{t+1} \in \mathcal{S}} T(s_t, a_t, s_{t+1}) = 1$ gilt, wenn T eine wohldefinierte Wahrscheinlichkeitsverteilung ist und die Aktion a_t im Zustand s_t ausgeführt werden kann. Trifft dies nicht zu, so gilt für alle Folgezustände s_{t+1} jedoch $T(s_t, a_t, s_{t+1}) = 0$ [47]. Für

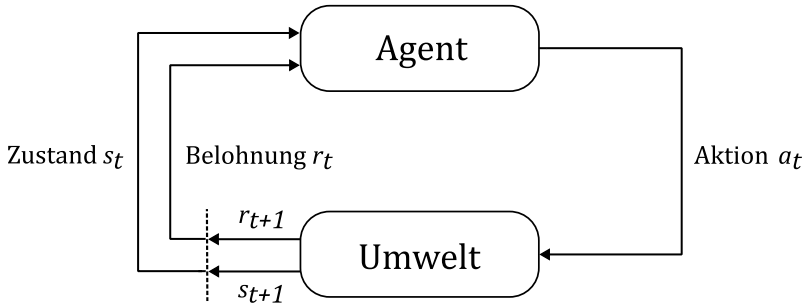


Abbildung 2.4: Schematisches Grundprinzip von RL: Dabei werden die Interaktion eines Agenten mit seiner Umwelt sowie die aus Aktionen resultierenden Zustandsübergänge und Belohnungen dargestellt. Adaptiert aus [42].

die Belohnungfunktion R wird teilweise in der Literatur die mögliche Wertemenge auf eine Teilmenge $\mathcal{R}$ der reellen Zahlen begrenzt (z.B. $\mathcal{R} = [0, R_{max}]$ mit $R_{max} \in \mathbb{R}^+$ [46]). Auch Sutton et al. [42] beschreibt den Belohnungsraum $\mathcal{R}$ als Teilmenge mit $\mathcal{R} \subset \mathbb{R}$. Die Wertemenge der Belohnungsfunktion hängt aber letztlich von ihrer Definition ab. Bei der Betrachtung von RL als Optimierungsproblem, kann die Belohnungsfunktion auch als Zielfunktion angesehen werden [41]. Die Kombination aus Transitionsfunktion T und Belohnungsfunktion R definieren das Modell eines MDP [47]. Der Diskontierungsfaktor γ ist ein Parameter zur Beeinflussung der Weitsichtigkeit eines Agenten, der in Kapitel 2.3.2 ausführlicher besprochen wird.
Das Grundprinzip von RL basierend auf MDPs wird in folgender Abbildung 2.4 schematisch dargestellt. Sie zeigt die Interaktion zwischen Agent und Umwelt. Gestartet wird mit einer Wahrscheinlichkeit $\rho(s_0)$ in einem Startzustand s_0. Anschließend führt der Agent in jedem Zeitschritt t eine Aktion a_t aus, durch die die Umwelt mit der Wahrscheinlichkeit $T(s_t, a_t, s_{t+1})$ in den Zustand s_{t+1} wechselt. Für sein Handeln erhält der Agent die Belohnung $r_{t+1} = R(s_t, a_t, s_{t+1})$. Die Belohnung erfolgt durch die Umwelt [42]. Da es sich um einen zeitdiskreten Prozess handelt, wird die Trennung zweier Zeitschritte in Abbildung 2.4 durch die gestrichelte Linie dargestellt. Aus Sicht des Agenten erfolgt die Belohnung mit r_t damit zeitversetzt für die im vorangegangen Zeitschritt gewählte Aktion.
Mit MDPs lassen sich verschiedene Arten von Aufgaben beschreiben, die in episodisch und kontinuierlich unterteilt werden. Episodische Aufgaben haben einen endlichen Horizont (engl. finite horizon) und schließen mit einem definierten Endzustand (engl. terminal state) ab. Ein Durchlauf vom Start- bis zum Endzustand wird als Episode bezeichnet. Kontinuierliche Aufgaben enden hingegen nicht und erstrecken sich über unendlich viele Zeitschritte [47]. Die kausale Folge von Zuständen, Aktionen und Belohnungen wird als Trajektorie τ bezeichnet [48]:

$$\tau = s_0, a_0, r_1, s_1, a_1, r_2, ..., s_h \tag{2.4}$$

Die Länge h einer Trajektorie hängt von der Art der Aufgabe ab und steht für die Anzahl der Zeitschritte. Im Falle einer episodischen Aufgabe gilt $h \in \mathbb{N}$, bei kontinuierlichen Auf-

gaben wiederum ist $h = \infty$ [49].
Die optimale Verhaltensstrategie zur Auswahl der Aktionen definiert die Lösung eines MDP, die über eine Vielzahl von Ansätzen berechnet werden kann. Beim Lösen kann zwischen Modell-basierten und Modell-freien Verfahren unterschieden werden. Bei Modell-basierten Verfahren wird vorausgesetzt, dass exakte Informationen über die Transition des Zustands und die Generierung der Belohnung a priori bekannt sind. Ansätze, die MDPs auf Basis dieses perfekten Modells der Umwelt lösen, können unter dem Begriff Dynamic Programming (DP, dt. dynamische Programmierung) zusammengefasst werden [42]. DP geht auf Richard Bellman im Jahre 1957 zurück und setzt auf die Zerlegung von komplexen Problemen in einfache Teilprobleme [41]. Beim Schätzen der Lösung wird in Teilen mit bestehenden Schätzungen gerechnet. Dies wird generell als Bootstrapping bezeichnet. In der Praxis sind exakte Modelle in RL-Problemen selten verfügbar und die Berechnung der Lösung mit DP erfordert extrem viel Speicherplatz, da die Anzahl möglicher Zustände auch schon bei einfachen Szenarien sehr hoch ist [50]. Ein alternatives Lösungsverfahren ist die Monte-Carlo-Methode, bei der die Umwelt stichprobenartig erkundet wird. Da hierfür kein Vorwissen benötigt wird, zählt dieser Ansatz grundsätzlich zu den modell-freien Algorithmen [41]. Zusätzlich kann das Lernen durch Sampling von Erfahrungen aus einem Simulationsmodell beschleunigt werden, wobei im Gegensatz zu DP nicht die vollständige Wahrscheinlichkeitsverteilung der Transition bekannt sein muss. Der Nachteil der Monte-Carlo-Methode ist, dass die Berechnung einer Schätzung (z.B. der erwarteten zukünftigen Belohnung) erst am Ende einer Episode durchgeführt werden kann. Daher kann diese Methode nur auf episodische Aufgaben angewendet werden. Eine Kombination aus DP und der Monte-Carlo-Methode – sprich aus Bootstrapping und Sampling – bildet die modell-freie Methode Temporal Difference (TD, dt. zeitliche Differenz) Learning. Hierbei wird die aktuelle Schätzung basierend auf vergangenen Schätzungen und unter Ausnutzung der zeitlichen Differenz verbessert. Das Ende einer Episode muss demnach gegenüber der Monte-Carlo-Methode nicht abgewartet werden. Sowohl Monte-Carlo-Methode als auch TD-Learning nutzen Erfahrungen aus der Vergangenheit um zur Problemlösung zu gelangen [42].

2.3.2 Belohnung

Aktionen eines Agenten führen zu Belohnungen oder Bestrafungen. Das Lernziel des Agenten ist – abgeleitet vom Belohnungsprinzip – die kumulative Belohnung auf lange Sicht zu maximieren. Da die exakten zukünftigen Belohnungen aufgrund des stochastischen Prozesses zum Zeitpunkt t allerdings unbekannt sind, wird generell die Maximierung des Erwartungswerts der berechneten Gesamtbelohnung (engl. return) als Ziel angestrebt [42]. Die zukünftige Gesamtbelohnung ab einem bestimmten Zeitschritt t wird im Folgenden als Return G_t bezeichnet und kann in einer episodischen Aufgabe mit der Trajektorie τ

der Länge h wie folgt berechnet werden [48]:

$$G_t = r_{t+1} + r_{t+2} + r_{t+3} + ... + r_{h-1} = \sum_{k=1}^{h-t-1} r_{t+k} \tag{2.5}$$

Für die Gesamtbelohnung einer ganzen Trajektorie gilt damit $G_0 = R(\tau)$. Bei Verwendung der Formel 2.5 ergibt sich jedoch für kontinuierliche Aufgaben mit $h = \infty$ eine Summe, die abhängig von der Definition der Belohnungsfunktion durchaus unendliche Werte annehmen könnte. Als Abhilfe wird der Diskontierungsfaktor γ mit einem Wertebereich von $0 \leq \gamma \leq 1$ eingeführt [50]:

$$G_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + ... = \sum_{k=1}^{\infty} \gamma^{k-1} r_{t+k} \tag{2.6}$$

G_t wird dann als diskontierter Return bezeichnet, der die Summe der zukünftigen Belohnungen für $\gamma < 1$ begrenzt. Der Diskontierungsfaktor steuert dadurch die Weitsichtigkeit des Agenten. Wenn $\gamma = 0$ ist, dann gilt $G_t = r_{t+1}$. Der Agent wäre sehr kurzsichtig und würde seine Aktionen a_t so wählen, dass die sofortige Belohnung r_{t+1} maximiert wird. Für γ nahe 1 wird der Agent dagegen sehr weitsichtig und gibt zukünftigen Belohnungen mehr Gewicht [50].
Für die Berechnung von RL-Algorithmen spielt der Zusammenhang von Belohnungen in verschiedenen Zeitschritten eine große Rolle. Der diskontierte Return kann rekursiv ausgedrückt werden [42]:

$$\begin{aligned} G_t &= r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \gamma^3 r_{t+4} + ... \\ &= r_{t+1} + \gamma(r_{t+2} + \gamma r_{t+3} + \gamma^2 r_{t+4} + ...) \\ &= r_{t+1} + \gamma G_{t+1} \end{aligned} \tag{2.7}$$

Als Beispiel für die Funktionsweise führen Sutton et al. eine kontinuierliche Belohnung von $+1$ in jedem Zeitschritt t an. Dabei ergibt sich ein konstanter Wert $G_t = \sum_{k=1}^{\infty} \gamma^{k-1} = \frac{1}{1-\gamma}$ als diskontierter Return [42].

2.3.3 Policy

Die Verhaltensstrategie π des Agenten – auch Policy genannt – spiegelt das zu erlernende intelligente Verhalten des Systems wieder. Sie kann als deterministische Policy $\pi \colon \mathcal{S} \to \mathcal{A}$ oder stochastisch mit $\pi \colon \mathcal{S} \times \mathcal{A} \to [0, 1]$ ausgeführt werden [45]. Im stochastischen Fall beschreibt π eine Wahrscheinlichkeitsverteilung über die Aktionen $a \in \mathcal{A}$, die abhängig vom Aktionsraum diskret oder kontinuierlich sein kann [49]. Unter Berücksichtigung der Wahrscheinlichkeit $\rho(s_0)$ in einem Zustand s_0 zu starten, berechnet sich die Wahrscheinlichkeit für das Zustandekommen einer bestimmten Trajektorie τ mit einer gegebenen

stochastischen Policy $\pi(s, a)$ wie folgt [48]:

$$p(\tau) = \rho(s_0) \prod_{t=1}^{h-1} \pi(s_t, a_t) T(s_t, a_t, s_{t+1}) \tag{2.8}$$

Für die stochastische Policy muss dabei gelten, dass sich die Wahrscheinlichkeiten für alle möglichen Aktionen aus dem Aktionsraum $\mathcal{A}$ zu 1 aufsummieren: $\sum_{a \in \mathcal{A}} \pi(s, a) = 1$ [47]. Theoretisch kann eine deterministische Policy auch als stochastisch aufgefasst werden, wobei die Wahrscheinlichkeit $\pi(s_t, a_t)$ für eine bestimmte Aktion in einem Zustand für jeden Zeitschritt t immer 1 ist. Im Folgenden wird, wenn nicht anders gekennzeichnet, von einer stochastischen Policy ausgegangen.
Die optimale Policy π^*, die den erwarteten Return einer Trajektorie maximiert, kann mathematisch wie folgt ausgedrückt werden [46]:

$$\pi^* = \arg\max_{\pi} \mathbb{E}_{\tau \sim p(\tau)}[R(\tau)] \tag{2.9}$$

Trajektorien mit höherer Wahrscheinlichkeit $p(\tau)$ (siehe Formel 2.8) haben dabei ein höheres Gewicht im erwarteten Return [48]. Das Finden der optimalen Policy π^* unter allen möglichen Policies π ist das grundsätzliche Ziel des Agenten in RL. Hierbei kann es vorkommen, dass eine kurzfristig niedrigere Belohnung zu Nutze eines höheren Returns über die gesamte Trajektorie in Kauf genommen wird [42].
Bei der Verfolgung dieses Ziels gibt es ein grundsätzliches Problem. Während dem Lernen per trial and error entwickelt der Agent eine Policy, auf Basis derer er seine Aktionen wählt. Für diese wird der Agent mit einem skalaren Belohnungssignal gefüttert, das jedoch keine Aussage darüber erlaubt, ob eine Aktion richtig oder falsch war. Es besteht also durchaus die Möglichkeit, dass das Wählen einer anderen Aktion eine höhere Belohnung erbracht hätte. Dieses Dilemma aus Erforschung der Umwelt und gierigem Ausnutzen des bisherigen Wissens über ebenjene wird in der Literatur als Problem zwischen Erforschung (engl. exploration) und Ausbeutung (engl. exploitation) bezeichnet. Es erfordert einen Kompromiss, den sogenannten Exploration-Exploitation Trade-off [47]. Theoretisch kann der Agent für jede Policy eine Aktion finden, die die maximale sofortige Belohnung einbringt. Das Verfolgen dieser Strategie wird auch als gierige (engl. greedy) Policy bezeichnet. Einen Kompromiss zwischen dieser Ausbeutung und der fortlaufenden Erforschung der Umwelt stellt die sogenannte ε-greedy Policy dar. Hierbei wird – mit einer geringen Wahrscheinlichkeit $\varepsilon \in [0, 1]$ – von Zeit zu Zeit eine zufällige Aktion aus einer Gleichverteilung gesampelt. Dies stellt sicher, dass während dem Lernprozess hin und wieder Aktionen gewählt werden, die der Agent zuvor in diesem Zustand noch nicht ausprobiert hat [42].

2.3.4 Wertfunktionen und Bellman-Gleichungen

Wie in Kapitel 2.3.2 beschrieben, hat der Agent die Maximierung des erwarteten zukünftigen Returns zum Ziel. Der erwartete zukünftige Return unter einer bestimmten Policy π, aus-

gehend vom aktuellen Zustand s, wird mit der sogenannten Zustands-Wertfunktion (engl. state-value function) $V^\pi \colon \mathcal{S} \to \mathbb{R}$ berechnet [47]:

$$V^\pi(s) = \mathbb{E}_\pi[G_t | s_t = s] = \mathbb{E}_\pi \left[\sum_{k=1}^{\infty} \gamma^{k-1} r_{t+k} \middle| s_t = s \right] \tag{2.10}$$

Die Zustands-Wertfunktion gibt den Wert für den Agent an, sich in einem bestimmten Zustand s zu befinden. Generell wird in der Literatur der aktuelle Zustand s_t als s und der Folgezustand s_{t+1} als s' bezeichnet [49]. Durch die rekursive Eigenschaft des Returns (siehe Formel 2.7) kann $V^\pi(s)$ in die sogenannte Bellman-Gleichung umgeformt werden [42, 45, 47]:

$$\begin{aligned} V^\pi(s) &= \mathbb{E}_\pi[G_t | s_t = s] \\ &= \mathbb{E}_\pi[r_{t+1} + \gamma G_{t+1} | s_t = s] \\ &= \sum_{s' \in \mathcal{S}} T(s, a, s') \big(R(s, a, s') + \gamma \mathbb{E}_\pi[G_{t+1} | s_{t+1} = s'] \big) \\ &= \sum_{s' \in \mathcal{S}} T(s, a, s') \big(R(s, a, s') + \gamma V^\pi(s') \big), \text{ mit } a \sim \pi(s, \cdot) \end{aligned} \tag{2.11}$$

Der Wert eines Zustandes s ist also abhängig von der sofortigen Belohnung $R(s, a, s')$ und dem zukünftig erwarteten diskontierten Wert $V^\pi(s')$, gewichtet mit der Übergangswahrscheinlichkeit T, dass ein bestimmter Folgezustand erreicht wird, summiert über alle möglichen Folgezustände s' [47]. Die Bellman-Gleichung spiegelt damit den Zusammenhang zwischen dem Wert des aktuellen Zustands und dem Wert aller möglichen Folgezustände wieder [42].
Analog zur Zustands-Wertfunktion kann eine Aktions-Wertfunktion (engl. action-value function)
$Q^\pi \colon \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ definiert werden, die zusätzlich zum Zustand die gewählte Aktion berücksichtigt, nach deren Ausführung die Policy π verfolgt wird [47]:

$$Q^\pi(s, a) = \mathbb{E}_\pi[G_t | s_t = s, a_t = a] = \mathbb{E}_\pi \left[\sum_{k=1}^{\infty} \gamma^{k-1} r_{t+k} \middle| s_t = s, a_t = a \right] \tag{2.12}$$

Die Wertfunktionen V^π und Q^π können über die Monte Carlo Methode geschätzt werden. Während der Agent eine Policy π verfolgt, wird der durchschnittliche Return, ausgehend von jedem besuchten Zustand, über die Gesamtzahl der Besuche dieses Zustands gebildet. Je öfter ein Zustand s besucht wird, desto eher konvergiert dieser Durchschnitt zum tatsächlichen Wert. Eine andere Möglichkeit ist das Nutzen von Funktionsapproximatoren (z.B. neuronale Netze), deren Parameter durch den Agent angepasst werden, sodass die parametrisierte Funktion möglichst genau mit der tatsächlichen Wertfunktion übereinstimmt [42].
Zusätzlich zur Zustands- und Aktions-Wertfunktion gibt es die sogenannte Vorteilsfunktion $A(s, a)$, die beschreibt, wie vorteilhaft es für den Agenten ist, sich für eine Aktion a

zu entscheiden anstatt der Policy π zu folgen [45]:

$$A^{\pi}(s,a) = Q^{\pi}(s,a) - V^{\pi}(s) \tag{2.13}$$

Wenn die Aktion a der durch die Policy π gewählten Aktion entspricht, so ergibt sich kein Vorteil.
Für die optimale Policy π^* muss gelten, dass es für alle $s \in \mathcal{S}$ keine andere Policy π gibt, die einen höheren Wert erzielt: $V^{\pi^*} \geq V^{\pi}$ [47]. Dabei kann es mehrere optimale Policies π^* geben, die jedoch derselben Zustands-Wertfunktion zuzuordnen sind. Mathematisch lässt sich diese optimale Zustands-Wertfunktion V^* wie folgt definieren [42]:

$$V^*(s) = \max_{\pi} V^{\pi}(s) \tag{2.14}$$

Analog kann diese Definition für die optimale Aktions-Wertfunktion gemacht werden [45]:

$$Q^*(s,a) = \max_{\pi} Q^{\pi}(s,a) \tag{2.15}$$

Wenn davon ausgegangen wird, dass der Agent rein gierig handelt und die optimale Policy π^* verfolgt, dann stimmen $V^*(s)$ und $Q^*(s,a)$ überein. Die Aktion a entstammt dabei der optimalen Policy und maximiert den Wert im Zustand s. Demnach kann Formel 2.14 angepasst werden zu [50]:

$$V^*(s) = \max_{a} Q^*(s,a) \tag{2.16}$$

Die optimale Zustands-Wertfunktion kann also ohne Wissen über eine spezifische Policy ausgedrückt werden, was zur Bellman-Optimalitätsgleichung führt [42, 47]:

$$\begin{aligned} V^*(s) &= \max_{a} Q^*(s,a) \\ &= \max_{a} \mathbb{E}_{\pi^*}[G_t | s_t = s, a_t = a] \\ &= \max_{a} \mathbb{E}_{\pi^*}[r_{t+1} + \gamma G_{t+1} | s_t = s, a_t = a] \\ &= \max_{a} \mathbb{E}[r_{t+1} + \gamma V^*(s_{t+1}) | s_t = s, a_t = a] \\ &= \max_{a} \sum_{s' \in \mathcal{S}} T(s,a,s') \big(R(s,a,s') + \gamma V^*(s') \big) \end{aligned} \tag{2.17}$$

Das Verfolgen der optimalen Policy ist also gleichzusetzen mit dem Wählen der besten Aktion, durch die der höchste Wert in einem Zustand erreicht wird. Gleichermaßen kann die optimale Aktions-Wertfunktion aufgestellt werden [47]:

$$Q^*(s,a) = \sum_{s' \in \mathcal{S}} T(s,a,s') \big(R(s,a,s') + \gamma \max_{a'} Q^*(s',a') \big) \tag{2.18}$$

Hierbei spielt bei der Angabe des zukünftig erwarteten Returns die Folgeaktion a' eine Rolle. Diese wird so gewählt, dass die Aktions-Wertfunktion maximiert wird. Ausgehend

von $Q^*(s, a)$ kann die optimale Policy wie folgt berechnet werden [45]:

$$\pi^*(s) = \arg\max_a Q^*(s, a) \tag{2.19}$$

Zusammenfassend beschreibt die optimale Policy im Zustand s die Aktion a, die unter der Berücksichtigung aller durch diese Aktion möglichen Folgezustände s' zu einem maximalen Nutzen führt [47].

2.3.5 Kategorisierung von RL-Algorithmen

RL-Algorithmen können nach verschiedenen Eigenschaften kategorisiert werden, die sich nicht zwingend gegenseitig ausschließen. Ein Algorithmus kann also mehrere der im Folgenden vorgestellten Merkmale erfüllen. In Kapitel 2.3.1 wurde bereits die Unterscheidung in Modell-basierte und Modell-freie RL-Algorithmen dargelegt. Ein Modell kann unter dem Begriff Planen (engl. planning) für ein effektiveres und schnelleres Lernen genutzt werden [47]. Der Fokus in RL liegt allerdings mehr auf dem Lernen und nicht dem Planen anhand eines Modells, da dieses in den meisten Szenarien nicht zur Verfügung steht [46].
Zudem kann bei der Art des Lernens zwischen Policy-based (dt. Policy-basiert) und Value-based (dt. Wert-basiert) differenziert werden. Policy-basierte Methoden optimieren direkt eine parametrisierte Policy π_θ, ohne zusätzliche Repräsentation einer Wertfunktion. Sie werden auch als Actor Methoden (dt. Methoden des Handelnden) bezeichnet und basieren auf Policy Gradients (PG, dt. Policy-Gradienten) [42]. Statt einer direkten Approximation der Policy wird die nächste Aktion bei Wert-basierten Methoden aus einer prädizierten Wertfunktion abgeleitet. Diese Methoden werden generell auch Critic Methoden (dt. Kritiker-Methoden) genannt, da die optimale Policy auf Basis der kritischen Beurteilung durch den Wert der Aktionen gesucht wird [50]. Während die Actor Methoden oft langsam lernen und Schätzungen mit hoher Varianz erzeugen, aber mit großer Sicherheit konvergieren und mit kontinuierlichen Aktionsräumen umgehen können, haben Critic Methoden genau gegenteilig oft einen systematischen Fehler (engl. bias), dafür eine niedrige Varianz und konvergieren meist schlechter [42]. Um die Vorteile beider Methoden auszunutzen, können sie zu Actor-Critic Methoden kombiniert werden, wobei sowohl Policy als auch Wertfunktion als approximierte Funktionen vorliegen [50]. Das Update der Policy (Actor) erfolgt dabei mit Hilfe des kritischen Feedbacks aus der geschätzten Wertfunktion (Critic) [47, 51]. Teilweise werden Actor-Critic Verfahren daher in der Literatur auch als Untergruppe der Policy-based Methoden gesehen [46]. Die beschriebenen Zusammenhänge der Kategorien werden in Abbildung 2.5 grafisch zusammengefasst. Darüber hinaus kann eine Differenzierung durch das sogenannte On- bzw. Off-Policy-Lernen erfolgen. On-Policy-Algorithmen zeichnen sich dadurch aus, dass sie dieselbe Policy auswerten und verbessern, die auch zum Treffen der Entscheidungen während des Lernens verwendet wird. Demgegenüber verwalten Off-Policy-Methoden zwei Repräsentationen der Verhaltensstrategie, die Behavior- (dt. Verhaltens-) und Target-Policy (dt. Ziel-Policy). Die Behavior-Policy β wird genutzt, um die Aktionen auszuwählen und repräsentiert damit das Verhalten des

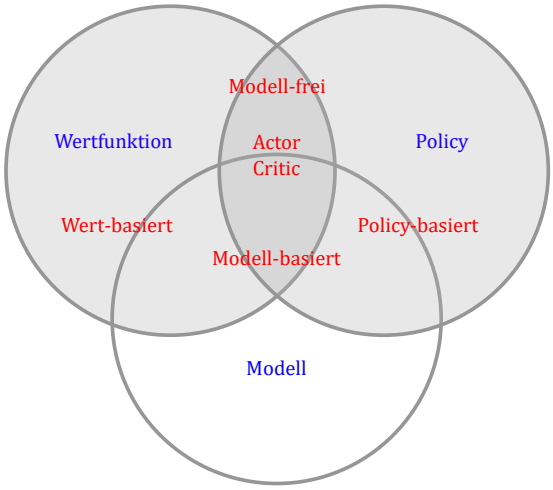


Abbildung 2.5: Kategorien von RL-Agenten: Die Art des Lernens kann grundsätzliche in Modell-frei, Modell-basiert, Wert-basiert oder Policy-basiert unterschieden werden. Adaptiert aus [52].

Agenten. Mit ihrer Hilfe werden Erfahrungen gesammelt, um die Target-Policy π zu optimieren. Es wird also von Trajektorien gelernt, die die zu erlernende Target-Policy nicht selbst verursacht hat. Dies sorgt bei Off-Policy-Algorithmen zu einer höheren Probeneffizienz (engl. sample efficiency), da sie von jeder Erfahrung profitieren können [45]. Allerdings konvergieren Off-Policy-Methoden durch die zwei verschiedenen Repräsentationen der Policy meist schlechter und weisen eine höhere Varianz auf [42]. On-Policy-Verfahren tendieren dagegen zu einem Bias, wenn Erfahrungen für Updates verwendet werden, die nicht unter der aktuellen Policy π entstanden sind [45].
Des Weiteren können RL-Algorithmen in Online- und Offline-Lernen eingeteilt werden. Während beim Offline-Lernen ganze Stapel (engl. batches) an Daten verarbeitet werden, um die Policy zu aktualisieren, findet beim Online-Lernen die Verarbeitung direkt mit dem Vorliegen eines neuen Samples statt. Einige Algorithmen wie beispielsweise Q-Learning gibt es als Online- und Offline-Variante [47].
Die folgenden Kapitel erläutern einige Formulierungen von Algorithmen genauer, kategorisiert nach der Art des Lernens in Policy-basierte, Wert-basierte und Actor-Critic Methoden.

2.3.6 Policy-basierte Methoden

In Policy-basierten Verfahren wird direkt eine parametrisierte Policy π_θ optimiert. Die hierzu benötigte Zielfunktion wird auch als Performancefunktion $J(\theta)$ bezeichnet und bemisst die Qualität der Policy unter den aktuellen Parametern θ [48]. Die Art der Funktionsapproximation spielt zunächst keine Rolle, solange $\nabla\pi_\theta$ für alle $s \in \mathcal{S}$ und $a \in \mathcal{A}$ existiert und endliche Werte annimmt. Generell wird hierbei von Policy Gradients (PG) gesprochen. Für den episodischen Fall kann die Performance als $J(\theta) = V^{\pi_\theta}(s_0)$ definiert werden, wobei V^{π_θ} den erwarteten Return beim Verfolgen der Policy π_θ angibt [42]. Eine

Verbesserung der Policy π_θ beim Update der Parameter θ kann durch das Policy Gradient Theorem garantiert werden [42]:

$$\begin{aligned}\nabla J(\theta) &\propto \sum_{s\in\mathcal{S}} \mu(s) \sum_{a\in\mathcal{A}} Q^\pi(s,a)\nabla\pi(s,a,\theta) \\ &= \mathbb{E}\left[\sum_{a\in\mathcal{A}} Q^\pi(s_t,a)\nabla\pi(s_t,a,\theta)\right]\end{aligned} \tag{2.20}$$

$\mu(s)$ ist hierbei Zustandsverteilung, $Q^\pi(s,a)$ die Aktions-Wertfunktion und $\nabla\pi(s,a,\theta)$ der Gradient der Policy in Bezug auf deren Parameter θ. Durch $\mu(s)$ findet eine Gewichtung nach der Häufigkeit des Auftretens der Zustände statt, die abhängig von der verfolgten Policy π ist. Die Änderung der Performancefunktion in Abhängigkeit der Parameter θ ist nach Formel 2.20 proportional zur Änderung der Policy. Anders ausgedrückt führt eine Erhöhung der Auswahlwahrscheinlichkeit bestimmter Aktionen unter der Policy π zu einer Erhöhung der Performance, jeweils in Abhängigkeit der Parameter θ. Um umgekehrt betrachtet also eine möglichst optimale Policy zu erreichen, muss die Performancefunktion maximiert werden. Dies kann mit Hilfe des stochastischen Gradientenanstiegs umgesetzt werden. Dabei wird das Update der Parameter θ basierend auf der geschätzten Performancefunktion wie folgt durchgeführt [42]:

$$\theta_{t+1} \leftarrow \theta_t + \alpha\widehat{\nabla J(\theta_t)} \tag{2.21}$$

Mit $\alpha > 0$ lässt sich die Schrittweite bzw. Lerngeschwindigkeit des Gradientenanstiegverfahrens beeinflussen [50]. Eine mögliche Implementierung der dargestellten Theorie ist der REINFORCE Algorithmus. Dabei wird der Erwartungswert aus Formel 2.20 umgeformt zu $\mathbb{E}\big[G_t\nabla\ln\big(\pi(s_t,a_t,\theta)\big)\big]$ und kann mit Hilfe der Monte-Carlo-Methode aus den Samples einer Trajektorie berechnet werden. Der REINFORCE Algorithmus wird wie folgt formuliert [50]: Hierbei definiert d die Dimension der Parameter θ. Für ausreichend klei-

Algorithmus 2.1 REINFORCE: Monte Carlo Policy Gradient [50]

Eingabe: Differenzierbare, parametrisierte Policy $\pi_\theta(a,s,\theta)$
Eingabe: Lernrate $\alpha > 0$
  Initialisiere Policy-Parameter $\theta \in \mathbb{R}^d$
  for jede Episode **do**
    Erzeuge Episode $\tau = s_0, a_0, r_1, ..., s_h$ und folge dabei $\pi(\cdot,\cdot,\theta)$
    for t = 0, 1, ..., h-1 **do**
      $G \leftarrow \sum_{k=t+1}^{h-1} \gamma^{k-t-1} r_k$
      $\theta \leftarrow \theta + \alpha G \nabla \ln\big(\pi(s_t,a_t,\theta)\big)$
    end for
  end for

ne Werte α ist die Konvergenz zu einem lokalen Maximum garantiert. Dennoch hat die REINFORCE-Methode eine hohe Varianz, der mit einer sogenannten Baseline (dt. Grundlinie) entgegen gewirkt werden kann. Diese Baseline wird durch eine gelernte Wertfunktion

ausgedrückt, mit der gute und schlechte Aktionen beim Parameter-Update entsprechend gewichtet werden. Dadurch nimmt die Varianz ab und die Lerngeschwindigkeit wird erhöht [42]. Es existieren zahlreiche Weiterentwicklungen der Policy Gradients, wie beispielsweise Trusted Region Policy Optimization (TRPO) und Proximal Policy Optimization (PPO), die hier nicht näher erklärt werden.

2.3.7 Wert-basierte Methoden

Wert-basierte Methoden sind indirekte Policy-Schätzer. Dabei wird eine Wertfunktion durch Funktionsapproximation gelernt (engl. value function estimation), aus der die Policy abgeleitet wird. Hierzu kann sowohl die Zustands-Wertfunktion V als auch die Aktions-Wertfunktion Q herangezogen werden. Die Prädiktion der Wertfunktion erfolgt – ähnlich wie bei Policy-basierten Methoden aus Kapitel 2.3.6 – durch Trajektorien-Sampling mit der Monte-Carlo-Methode [48]:

$$Q^{\pi}(s,a) = \mathbb{E}_{\tau \sim p(\tau)}[R(\tau)] \tag{2.22}$$

Bei entsprechend großer Anzahl an Trajektorien N kann dabei ein adäquater Erwartungswert und somit eine Schätzung für die Wertfunktion berechnet werden [48]:

$$\widehat{Q}^{\pi}(s,a) = \frac{1}{N} \sum_{n=1}^{N} [R(\tau_n)] \tag{2.23}$$

Anschließend kann das Optimum der Wertfunktion mit dem Gradientenanstieg berechnet werden. Allerdings erfolgt die Auswertung aufgrund der Monte-Carlo-Methode erst zum Ende einer Trajektorie. Dies führt zu einer schlechten Effizienz dieses Verfahrens und setzt ein episodisches Szenario voraus [48, 50].
Eine weitere Lösungsmethode ist das Temporal Difference (TD) Learning, wobei der rekursive Charakter der in Kapitel 2.3.4 beschriebenen Bellman-Gleichungen genutzt wird, um iterativ in jedem Zeitschritt eine aktualisierte Wertfunktion zu berechnen. Die Berechnungsmethode erfolgt dabei auf der zeitlichen Differenz zwischen zwei oder mehreren λ Zeitschritten. Bei der Berechnung auf Basis von zwei aufeinanderfolgenden Zeitschritten wird von TD(0) oder auch Ein-Schritt-TD gesprochen. Es stellt eine besondere Form von TD(λ)-Learning dar. Die Update-Regel für TD(0)-Learning nutzt die Zustands-Wertfunktion und lautet wie folgt [47]:

$$V(s_t) \leftarrow V(s_t) + \alpha\big(r_{t+1} + \gamma V(s_{t+1}) - V(s_t)\big) \tag{2.24}$$

Anhand von Formel 2.24 ist sichtbar, dass TD(0)-Learning die neue Wertfunktion in Teilen mit der aktuellen Schätzung berechnet, also Bootstrapping anwendet. Der Fehler über die zeitliche Differenz – auch TD-error genannt – geht mit $\delta_t = r_{t+1} + \gamma V(s_{t+1}) - V(s_t)$ in Formel 2.24 ein und bemisst die Differenz zwischen dem Wert von Zustand s_t und der Schätzung für s_{t+1} [42]. Aus TD-Learning kann das sogenannte Q-Learning abgeleitet wer-

den. Diese Methode verwendet die Aktions-Wertfunktion statt der Zustands-Wertfunktion. Die Update-Regel basiert auf der Bellman-Optimalitätsgleichung aus Formel 2.18 [49]:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha\big(r_{t+1} + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t)\big) \tag{2.25}$$

Beim Q-Learning wird der max-Operator über alle möglichen Aktionen ausgehend vom Folgezustand s_{t+1} angewendet. Es wird dann die Aktion gewählt, die den höchsten Wert in Zustand s_{t+1} bringt, statt der aktuellen Policy zu folgen. Damit ist Q-Learning ein Off-Policy-Verfahren. Für Q-Learning kann folgender Algorithmus formuliert werden [47]: Demgegenüber steht SARSA (State-Action-Reward-State-Action). Bei diesem Verfahren

Algorithmus 2.2 Q-Learning [47]

Eingabe: Diskontierungsfaktor γ
Eingabe: Lernrate $\alpha > 0$
  Initialisiere Q zufällig
  for jede Episode **do**
    Initialisiere Startzustand s
    repeat
      Wähle Aktion $a \in \mathcal{A}$ ausgehend von Q-Funktion und Exploration-Strategie
      Führe a aus
      Beobachte neuen Zustand s' und Belohnung r'
      $Q(s, a) \leftarrow Q(s, a) + \alpha\big(r' + \gamma \max_{a'} Q(s', a') - Q(s, a)\big)$
      $s \leftarrow s'$
    until s' ist ein Endzustand
  end for

wird die Folgeaktion a_{t+1} durch die aktuelle Policy gewählt [47]:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha\big(r_{t+1} + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)\big) \tag{2.26}$$

SARSA ist eine On-Policy-Methode, da s_t, a_t, r_{t+1}, s_{t+1} und a_{t+1} aus der gleichen Policy π stammen [48]. Für eine geringe Anzahl von Zuständen und Aktionen kann das Lernen der Wertfunktion in einer Tabelle erfolgen. Jede Zeile repräsentiert einen Zustand, jede Spalte eine Aktion. In der Tabelle wird der Wert für jede Kombination aus Zustand und Aktion $Q(s, a)$ gespeichert und aktualisiert. Diese als Tabular-Q-Learning (dt. tabellarisches Q-Learning) bezeichnete Methode hat den Nachteil, dass sie für hochdimensionale Zustands- und Aktionsräume sehr viel Speicher benötigt [44]. Mit steigender Anzahl an diskreten Zuständen und Aktionen steigen auch die möglichen Kombinationen exponentiell. Demnach ist die tabellarische Methode für kontinuierliche Zustandsräume erst recht nicht einsetzbar. Diese Problematik bei hoher Dimensionalität wird in der Literatur auch als Curse of Dimensionality (dt. Fluch der Dimensionalität) bezeichnet. In solchen Fällen, die in realen Szenarien (z.B. Robotik) die überwiegende Mehrheit bilden, werden meist bei approximierten Funktionen eingesetzt [47]. Auf weitere Algorithmen, die neuronale Netze als Funktionsapproximatoren einsetzen, wird in Kapitel 2.3.9 eingegangen.

2.3.8 Actor-Critic Methoden

In Kapitel 2.3.5 wurde bereits beschrieben, dass die Actor-Critic Methoden je eine Repräsentation der Policy und der Wertfunktion verwalten, wobei die Policy als Actor und die Wertfunktion als Critic bezeichnet wird. Der Critic berechnet den TD-error δ_t für eine bestimmte Aktion, also die Differenz zwischen Wert des alten Zustands und dem geschätzten Wert des neuen Zustands. Mit Hilfe des TD-errors kann der Vorteil der Aktionen bewertet und als Gewichtung für den geschätzten Policy Gradienten genutzt werden [47]. Das Parameter-Update des Policy-basierten REINFORCE Algorithmus ohne Diskontierungsfaktor kann damit wie folgt verändert werden [42]:

$$\theta_{t+1} \leftarrow \theta_t + \alpha\delta_t \nabla \ln\big(\pi(s_t, a_t, \theta_t)\big) \tag{2.27}$$

Um den TD-error δ_t zu berechnen wird hierbei der Ein-Schritt Return genutzt. Mit dem TD-error als Baseline wird die Varianz gegenüber herkömmlichen Policy-basierten Methoden reduziert, ohne einen zusätzlichen Bias – wie bei Wert-basierten Methoden üblich – zu erzeugen. Gegenüber dem REINFORCE Algorithmus muss zusätzlich zu θ für den Actor ein zweiter Parametersatz w für den Critic verwaltet werden. Als Ein-Schritt Actor-Critic lässt sich folgender Algorithmus formulieren [42]:

Algorithm 2.3 Ein-Schritt Actor-Critic [42]

Eingabe: Differenzierbare, parametrisierte Policy $\pi_\theta(a, s, \theta)$
Eingabe: Differenzierbare, parametrisierte Zustands-Wertfunktion $\widehat{V}(s, \mathrm{w})$
Eingabe: Lernraten $\alpha^\theta > 0$ und $\alpha^\mathrm{w} > 0$
Initialisiere Policy-Parameter $\theta \in \mathbb{R}^{d_1}$ und Parameter der Zustands-Wertfunktion $\mathrm{w} \in \mathbb{R}^{d_2}$
for jede Episode **do**
  Initialisiere Startzustand s
  $I \leftarrow 1$
  repeat
    $a \in \pi(s, \cdot, \theta)$
    Führe a aus
    Beobachte neuen Zustand s' und Belohnung r'
    $\delta \leftarrow r' + \gamma\widehat{V}(s', \mathrm{w}) - \widehat{V}(s', \mathrm{w})$
    $\mathrm{w} \leftarrow \mathrm{w} + \alpha^\mathrm{w}\delta\nabla\widehat{V}(s', \mathrm{w})$
    $\theta \leftarrow \theta + \alpha^\theta I\delta\nabla \ln\pi(s, a, \theta)$
    $I \leftarrow \gamma I$
    $s \leftarrow s'$
  until s' ist ein Endzustand
end for

Die Actor-Critic Verfahren bringen gegenüber Wert-basierten Methoden den Hauptvorteil mit, dass sie mit kontinuierlichen Aktionsräumen umgehen können. Dies ist deswegen

möglich, da die Policy als wohldefinierte, parametrisierte Funktion vorliegt und die Aktionen nicht aus einer Wertfunktion abgeleitet werden [45]. Welches Verfahren letztlich verwendet wird, hängt von der Aufgabenstellung sowie der Beschaffenheit des Zustands- und Aktionsraums ab.

2.3.9 Deep Reeinforcement Learning

Deep Learning beschreibt eine Form des maschinellen Lernens, bei dem Informationen schichtweise durch verkettete Funktionen verarbeitet werden. Diese Schichten, die wiederum aus mehreren sogenannten Neuronen bestehen, extrahieren jeweils Merkmale für die Folgeschicht. Merkmale werden gelernt, statt sie wie beim klassischen maschinellen Lernen (z.B. Support Vector Machine) vorzugeben. Diese Art der Lernmaschine wird als neuronales Netz bezeichnet und besteht aus einer Eingangs- und Ausgangsschicht sowie mindestens einer versteckten Schicht. Da neuronale Netze mit einer versteckten Schicht noch einer Support Vector Machine gleichkommen, wird erst ab zwei versteckten Schichten von Deep Learning gesprochen [53]. Mit Hilfe von Deep Learning kann RL zum Lösen komplexer Entscheidungsprobleme mit hochdimensionalen Zustands- und Aktionsräumen ertüchtigt werden. Alle Algorithmen in Deep RL basieren auf dem gleichen Ansatz, die Funktionen der optimalen Policy, die Wertfunktion oder die Vorteilsfunktion durch neuronale Netze zu approximieren [46]. Im Folgenden wird zunächst auf Wert-basierte Methoden im Bereich des Deep RL eingegangen, da diese für die vorliegende Arbeit am relevantesten sind. Im Anschluss werden einige andere bekannte Algorithmen zusammengefasst.
Die Geschichte von Deep RL begann 2013 mit der Veröffentlichung des Deep Q-Learning Algorithmus. Hierbei konnte der Agent in drei Atari Spielen gegen menschliche professionelle Spieler gewinnen [54]. Deep Q-Learning setzt auf die Approximation der Aktions-Wertfunktion durch ein neuronales Netz, mit den Gewichten θ [54]:

$$Q(s, a; \theta) \approx Q^*(s, a) \tag{2.28}$$

Das neuronale Netz bezeichnen Mnih et al. [54] als Deep Q-Network. Um ein stabiles Training der neuronalen Netze zu erreichen, wurde ein sogenannter Erfahrungspuffer $\mathcal{D}$ (engl. experience replay) eingeführt. Dieser bezeichnet einen Speicher vergangener Kombinationen aus Zustand s, Aktion a, Belohnung r und Folgezustand s'. Während des Lernprozesses werden zufällige Erfahrungen als sogenannter Minibatch $\mathcal{M}$ aus dem Erfahrungspuffer gesampelt. Dies erhöht die Probeneffizienz und verhindert große Schwankungen des Lernfortschritts. Beim Training werden die Verlustfunktionen (engl. loss functions) L_i für alle Samples aus $\mathcal{M}$ durch den stochastischen Gradientenabstieg minimiert. L_i berechnet sich durch den quadratischen Fehler zwischen Ziel (engl. target) y_i und Approximation $Q(s, a; \theta_i)$ [54]:

$$L_i(\theta_i) = \mathbb{E}_{s,a,r,s' \sim \mathcal{M}(\mathcal{D})}\left[(y_i - Q(s, a; \theta_i))^2\right] \tag{2.29}$$

Die Ziele y_i werden dabei auf Basis der Gewichte θ_{i-1} aus der vorigen Iteration bestimmt [54]:

$$y_i = \mathbb{E}_{s,a,r,s' \sim \mathcal{M}(\mathcal{D})} \big[r + \gamma \max_{a'} Q(s', a'; \theta_{i-1} | s, a \big] \tag{2.30}$$

Zwei Jahre später brachten Mnih et al. [55] einen Verbesserungsvorschlag ein, der Korrelationen zwischen den Zielen und der Approximation reduzieren soll. Ein zweites neuronales Netz – das sogenannte Target Network – bildet die Aktions-Wertfunktion ab, auf deren Basis das Ziel y_i berechnet wird. Die Gewichte θ_i^- dieses Netzes werden nicht automatisch in jeder Iteration i mit den Werten von θ_i überschrieben, sondern erst nach einer einstellbaren Anzahl an Schritten C. Für alle Schritte dazwischen werden die Gewichte θ_i^- konstant gehalten. Die Berechnung der Verlustfunktion ändert sich nicht, nur θ_{i-1} in Formel 2.30 wird durch θ_i^- ersetzt [55].
Hasselt et al. [56] schlagen eine Erweiterung der DQNs vor, die sie Double Deep Q-Learning nennen. Diese soll dem Problem der Überschätzung entgegen wirken und eine Verbesserung der Performance in den Atari Spielen bewirken. Die Wahrscheinlichkeit für eine Überschätzung ist höher, wenn dieselbe Wertfunktion zum Auswählen und Bewerten einer Aktion genutzt wird. Daher nehmen sie eine Trennung mit Hilfe zweier Gewichtssätze vor. Der bisherige Algorithmus hält nach der Neuerung von Mnih et al. [55] schon zwei Netze in Form von Behavior und Target Network bereit. Es bieten sich die Gewichte θ_i und θ_i^- zur Verwendung an, sodass kein zusätzlicher Rechenaufwand durch weitere Parameter entsteht. Demnach lässt sich die Formel 2.30 umschreiben in [56]:

$$y_i = \mathbb{E}_{s,a,r,s' \sim \mathcal{M}(\mathcal{D})} \big[r + \gamma Q(s', \arg\max_{a} Q(s, a; \theta_i); \theta_i^- | s, a \big] \tag{2.31}$$

Die Autoren [56] zeigen in ihren Tests, dass sich Double DQNs (DDQN) robuster verhalten und besser generalisieren als die Vorgängeralgorithmen. Aufgrund des Erfahrungspuffers und des Target Networks gelten alle genannten Formen des Deep Q-Learnings als Off-Policy-Algorithmen. Es wird nicht direkt auf Basis der ausgeführten Policy gelernt.
Neben Deep Q-Learning als reine Wert-basierte Methode existieren einige andere Actor-Critic Algorithmen wie beispielsweise DDPGs, die zusätzlich zur Wertfunktion die Policy durch ein neuronales Netz abbilden. Dadurch wird der Umgang mit kontinuierlichen Aktionsräumen möglich, doch steigt gleichzeitig der Rechenaufwand durch zusätzliche neuronale Netze sowie der Speicherbedarf für die Erfahrungspuffer. Mit Graphics Processing Units (GPUs) kann der Rechenaufwand für neuronale Netze durch die Optimierung auf Matrix- und Vektoroperationen schneller bewältigt werden. Demgegenüber steht beispielsweise der Algorithmus A3C, der die Rechenlast durch eine Verteilung von Kopien des Agenten auf verschiedene Kerne einer Central Processing Unit (CPU) verteilt. Da die verteilten Agenten jeweils in ihrer eigenen Umwelt trainieren, kommen die Updates der neuronalen Netze aus unabhängigen Erfahrungen zustande. Die parallele Berechnung führt zum Entfall des Erfahrungspuffers und macht A3C zu einer On-Policy-Methode [12]. Andere Algorithmen wie TRPO und PPO haben sich aus den PGs entwickelt und adressieren das Problem

schlechter Parameter-Updates, ähnlich wie die Baseline für den REINFORCE Algorithmus (siehe Kapitel 2.3.6). TRPO optimiert direkt eine als neuronales Netz abgebildete Policy, deren Updates auf eine glaubhafte Region, in der die Approximation noch gültig ist, beschränkt sind. Dabei wird die Distanz zwischen aktueller und vorgeschlagener Policy berechnet und anhand einer Baseline begrenzt. Als Entfernungsmaß zwischen den zwei Wahrscheinlichkeitsverteilungen dient die sogenannte KL-Divergenz (Kullback-Leibler). Gleichzeitig wird eine Schätzung des Vorteils der Aktionen – abgeleitet vom Critic – zur Bewertung der Policy in die Optimierung miteinbezogen [46]. Aufgrund der hohen Komplexität des Algorithmus stellten Schulman et al. [57] den von TRPO abgeleiteten Ansatz PPO vor, der einfacher zu implementieren ist und gleichzeitig die Performance von TRPO übertrifft. Insgesamt haben sich die Algorithmen in den letzten Jahren beständig weiterentwickelt. Es existiert eine Vielzahl an Ansätzen, deren Eignung in Bezug auf Rechenkapazität, Komplexität und Performance abhängig vom Anwendungsbereich bewertet werden muss.

Da Deep RL auf neuronalen Netzen aufbaut, ist die Anzahl der einstellbaren Hyperparameter gegenüber klassischen RL-Algorithmen nochmals deutlich erhöht. Eine besondere Herausforderung stellt dabei sowohl die Komplexität der Algorithmen als auch der zeitaufwendige Trainingsprozess dar, der meist auf Basis von Simulationen stattfindet. Über die Optimierung der Hyperparameter wird in Bezug auf RL in der Literatur wenig diskutiert [58]. Dennoch gibt es neben dem händischen Ausprobieren verschiedene Ansätze, systematisch die beste Parameterkombination zu finden, um einen schnellen Lernerfolg und eine hohe Performance zu erreichen. Die einfachste und meistverbreitete Methode ist die sogenannte Grid Search (dt. Rastersuche). Dabei werden bestimmte Hyperparameter ausgewählt, die einen sogenannten Suchraum bilden. Dieser wird systematisch abgearbeitet, indem ausgewählte Werte der Hyperparameter in Kombination miteinander getestet werden. Der Nachteil dieser Methode ist, dass der Zeitaufwand mit der Anzahl der Hyperparameter exponentiell zunimmt [59]. Abbildung 2.6 veranschaulicht den Suchraum einer Grid Search mit zwei verschiedenen Hyperparametern a und b. Im Beispiel aus Abbildung 2.6 werden je Hyperparameter drei verschiedene Werte für die Optimierung betrachtet. Es ergeben sich demnach $3 \cdot 3 = 9$ Kombinationen, die durch die orangenen Punkte repräsentiert werden. Die Performance in Abhängigkeit der Werte der Hyperparameter ist vereinfacht in rot und blau dargestellt. Der Hyperparameter b kann dem Beispiel nach als irrelevant betrachtet werden, da die Performance über seinen Wertebereich durchgängig ähnlich gering bleibt. Hyperparameter a hingegen hat einen größeren Einfluss auf die Performance und erreicht das Maximum im mittleren Wertebereich. Dieser Wert gilt dann als optimal. Das Gitter der Grid Search muss eng genug sein, um keine Performance-Maxima zu übersehen. Ähnlich wie die Grid Search basiert auch die Random Search (dt. Zufallssuche) auf dem Prinzip, verschiedene Kombinationen aus Hyperparametern systematisch zu trainieren und auszuwerten. Im Gegensatz zur Grid Search folgen die gewählten Werte der Hyperparameter dabei jedoch keinem Gitter, sondern werden gleichverteilt aus einem bestimmten Wertebereich gezogen. Der Genetische Algorithmus und die Bayes'sche Optimie-

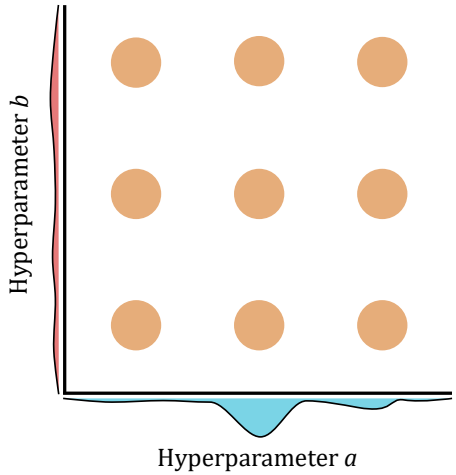


Abbildung 2.6: Grafische Veranschaulichung der Grid Search: Ausgewählte Hyperparameter bilden einen Suchraum. Dabei werden systematisch alle Kombinationen miteinander getestet. Adaptiert aus [60].

rung erzeugen gegenüber Grid und Random Search die Kombination der Hyperparameter im Laufe der Optimierung. Beim Genetischen Algorithmus wird das Optimierungsproblem in Anlehnung an die biologische Evolution in Chromosomen kodiert, aus denen eine Ausgangspopulation gebildet wird. Die Performance der Individuen wird bewertet und eine natürliche Auslese vorgenommen. Ist das Abbruchkriterium noch nicht erreicht, so werden aus den zwei gewählten Elternteilen die Nachfahren erzeugt und zur Population hinzugefügt [58, 61]. Die Bayes'sche Optimierung nutzt für die Auswahl einer neuen Parameterkombination ein probabilistisches Modell statt einer Fitnessfunktion, das versucht, die Performance ungetesteter Parameterkombinationen vorherzusagen. Zusammenfassend lässt sich festhalten, dass der Genetische Algorithmus und die Bayes'sche Optimierung gegenüber der Grid und Random Search zwar effektiver arbeiten, jedoch komplexer und aufwendiger zu implementieren sind [59].

Aktivierungsfunktionen sind ein sehr wichtiger Bestandteil im Bereich Deep-Learning und bestimmen den Aktivierungszustand des Neurons. Sie sind entscheidend für die Leistungsfähigkeit eines neuronalen Netzwerks und ermöglichen die Einführung von Nichtlinearität in das Modell. Der Unterschied zwischen ihnen liegt im Definitionsbereich. Sie helfen dabei, die Ausgabe jedes Eingangs in ihrem Definitionsbereich zu normalisieren. Dies reduziert die Komplexität im Modell und verringert die Rechenleistung gegenüber Modellen ohne eine Aktivierungsfunktion. Im Folgenden wird auf die drei gängigen Aktivierungsfunktionen eingegangen.

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{2.32}$$

$$\tanh(x) = \frac{2}{1 + e^{-2x}} - 1 \tag{2.33}$$

$$ReLU(x) = \max\{0, x\} \tag{2.34}$$

Die Sigmoidfunktion ist eine glatte nichtlineare Aktivierungsfunktion und kann zur Entscheidungsfindung bzw. zur Vorhersage eingesetzt werden, weil sich der Wertebereich von x zwischen $[0, 1]$ befindet. Der Nachteil bei der Verwendung der Sigmoidfunktion ist, dass das „Vanishing Gradient“ -Problem (dt: verschwindender Gradient) auftritt. Dies hat den Grund, dass die Funktion große Eingabewerte zwischen ihrem Definitionsbereich umwandelt und dadurch ihre Ableitungen viel kleiner werden. Je mehr Schichten das Neuronale Netz hat, desto mehr Informationen werden komprimiert und gehen in jeder Schicht verloren.

Die hyperbolische Tangenten-Aktivierungsfunktion (tanh) ist eine skalierte Sigmoidfunktion und hat ihre Werte x im Intervall $[-1, 1]$. Die tanh-Funktion wird gegenüber der Sigmoidfunktion dahingehend bevorzugt, weil der verschwindende Gradient (zum Teil) behoben wird. Der Vorteil bei der Verwendung der tanh-Funktion ist, dass die Werte auf nahe 0 gebracht werden und die negativen Werte als stark negativ bewertet werden. Dies ist bei der Sigmoidfunktion nicht der Fall, weil der Bereich für die Sigmoidfunktion zwischen 0 und 1 liegt. Da die tanh-Funktion Null zentriert ist, wird sie zwar im Gegensatz zur Sigmoidfunktion bevorzugt, löst jedoch das Problem des verschwindenden Gradienten nicht vollständig.

Die am häufigsten verwendete Aktivierungsfunktion bei neuronalen Netzen ist die sogenannte Rectified Linear Unit. Das Prinzip der ReLU-Funktion wird dadurch beschrieben, indem negative Werte durch 0 ersetzt und positive Werte gleichbleibend weitergegeben werden. Somit entspricht die Ausgabe des Neurons die Höhe der Aktivierung x, wenn x im positiven Wertebereich liegt. Daher weist ReLU eine begrenzte Nichtlinearität auf. Das Problem bei der Verwendung von ReLU ist jedoch, dass die negativen Neuronen vollständig abgeschaltet werden, dadurch negative Werte nicht vorhergesagt werden können. Dennoch wird sie gegenüber anderen Aktivierungsfunktionen bevorzugt, weil sie eine geringere Wahrscheinlichkeit des verschwindenden Gradienten aufweist.

Im Bereich des Deep Learnings gibt es verschiedene Architekturen neuronaler Netzwerke, die für unterschiedliche Anwendungsfälle und Datenstrukturen entwickelt wurden. Zwei solcher Architekturen sind das Feedforward Neural Network und das Long Short-Term Memory Netzwerk [39].

Das FNN, auch als vorwärtsgerichtetes neuronales Netzwerk bekannt, ist die grundlegendste Form eines künstlichen Neuronalen Netzwerks. Es besteht aus einer oder mehreren Schichten von Neuronen, die miteinander verbunden sind, aber keine Rückkopplungsschleifen aufweisen. Informationen fließen nur in eine Richtung, vom Eingang bis zum Ausgang, ohne dass Zwischenzustände gespeichert werden. Die Struktur eines FNN besteht typi-

scherweise aus einer Eingabeschicht, einer oder mehreren versteckten Schichten und einer Ausgabeschicht. Jedes Neuron in den versteckten Schichten und der Ausgabeschicht berechnet eine gewichtete Summe seiner Eingaben und wendet eine Aktivierungsfunktion auf diese Summe an. Während des Trainings wird das FNN durch Backpropagation und Gradientenabstieg angepasst, um den Fehler zwischen den Vorhersagen des Netzwerks und den tatsächlichen Zielwerten zu minimieren.

LSTM ist eine spezielle Variante eines Recurrent Neural Network und wurde entwickelt, um das Problem des Verschwindens oder Explodierens des Gradienten in herkömmlichen RNN zu überwinden. RNN haben Rückkopplungsschleifen, die es ihnen ermöglichen, Informationen über vergangene Zeitschritte zu speichern, was sie für die Verarbeitung sequenzieller Daten wie Zeitreihen ideal macht. Im Gegensatz zu herkömmlichen RNNs verwenden LSTM-Netzwerke spezielle Schaltungen, die als Zellen bezeichnet werden. Diese Zellen verfügen über Eingangstore, Vergessenstore und Ausgangstore, die es ihnen ermöglichen, Informationen über lange Zeiträume zu speichern und relevante Informationen auszuwählen, die an zukünftige Zeitschritte weitergegeben werden [39]. Dadurch können LSTMs komplexe Abhängigkeiten in den Daten erfassen. Sie eignen sich gut für die Verarbeitung von Zeitreihen, natürlicher Sprache und anderen sequenziellen Datenstrukturen.

3 Modellierung und Simulation

Eines der Ziele der Wissenschaft ist es, komplexe Systeme zu verstehen und ihre Reaktionen auf bestimmte Einflüsse vorhersagen zu können. Um mehr darüber zu erfahren, ist es notwendig, das System in den Experimenten verschiedenen Bedingungen auszusetzen. Allerdings ist es nicht immer möglich, diese Experimente direkt an realen Objekten durchzuführen. Daher ist die Anwendung digitaler Simulationsmethoden in vielen Bereichen der Technologieentwicklung unverzichtbar geworden. Somit kann das Verhalten komplexer Prozesse oder Systeme bereits früh im Entwicklungsprozess vorhergesagt und analysiert werden. Computersimulationen sind heute entscheidend, insbesondere bei der Entwicklung innovativer und komplexer Energiemanagementkonzepte für Fahrzeuge.

Die Aufgabe bei der Erstellung eines Modells besteht darin, die Eigenschaften des realen Systems nachzubilden. Um diese Aufgabe zu erfüllen, muss Zugang zu Informationen über das zu modellierende reale System bereitgestellt werden. Diese Informationen sind in drei Typen unterteilt [62]. In erster Linie muss Wissen über das reale System vorhanden sein. Die zweite Art von Informationen sind Daten, die aus gezielten Experimenten stammen. Der dritte Typ sind Annahmen, die über die Beziehung zwischen Modell und System getroffen werden. Da bei allen Modelltypen die selben Annahmen getroffen werden, wird zwischen den beiden anderen Arten von Informationen, Wissen und Daten, unterschieden. Ein Systemmodell kann in mehrere Teilmodelle aufgeteilt werden. Dadurch kann bei der Modellierung gezielt auf die Anforderungen der Teilmodelle eingegangen werden. Nachdem alle Teilmodelle erstellt sind, können sie zu einem Gesamtmodell verknüpft werden [63]. Bei der Modellierung einzelner Komponenten kann nun zwischen Modellen unterschieden werden, die nur mit Hilfe von Wissen (White–Box–Modelle) erstellt werden oder die mit Hilfe von experimentellen Daten (Black–Box–Modelle) geschätzt werden. Es ist aber auch eine Kombination beider Modellierungsarten möglich. Solche Modelle werden als gemischte Modelle (Grey–Box–Modelle) bezeichnet.
Physikalische Modelle, auch White–Box–Modelle genannt, simulieren Systemverhalten mit Hilfe mathematischer Gleichungen, die die physikalischen Vorgänge im System beschreiben. White–Box–Modelle werden aus präziser theoretischer Analyse realer Prozesse abgeleitet. Das Modell zeichnet sich dadurch aus, dass die Modellstruktur genau bekannt ist und die Modellparameter den physikalischen Parametern entsprechen. White–Box–Modelle haben eine hohe Genauigkeit, erfordern jedoch eine sehr sorgfältige Analyse des Systemverhaltens. Je nach gewünschter Genauigkeit kann das Modell jedoch beliebig komplex werden oder der Modellierungsaufwand sehr hoch sein.
Die zweite Art an Modellierung sind Black–Box–Modelle. Diese werden verwendet, wenn die inneren Prozesse eines Systems sehr komplex oder nicht zugänglich sind. Systemeigenschaften können z.B. aufgrund von Konstruktionen nicht zugänglich sein oder nicht vollständig verstanden werden. Der Vorteil dieser Abbildung ist der geringe Aufwand

zur Darstellung von Systemeigenschaften. Komplexe Systeme können durch Black–Box–Modelle mit nur wenigen Funktionen oder Kennfelder beschrieben werden. Jedoch besteht bei dieser Modellierungsart die Gefahr, dass wichtige Einflussfaktoren und Abhängigkeiten nicht erkannt oder übersehen werden.
Es besteht noch die Möglichkeit, diese beiden Modellierungsarten zu kombinieren. Solche Modelle werden Grey–Box–Modelle genannt. Bei der Grey–Box–Modellierung ist es möglich beide Arten von Informationen – qualitatives Wissen und quantitatives (Daten–)Wissen – zu integrieren. Ein Grey–Box–Modell besteht somit aus einer vereinfachten physikalischen Beschreibung des Systemverhaltens, dessen Parameter anhand gemessener Eingangs- und Ausgangsdaten geschätzt werden. Sie beinhalten in der Regel Informationen aus physikalischen Gleichungen und Messdaten sowie qualitative Informationen in Form von Regeln.

In dieser Arbeit wird eine DRL–basierte Energiemanagementstrategie für ein 12 V-Bordnetz simuliert und anschließend optimiert. Um die Spannungsstabilität in einem Bordnetz zu simulieren, muss das dynamische Verhalten aller relevanten Komponenten möglichst genau abgebildet werden. Neben der eigentlichen Umsetzung von Energiemanagementstrategien werden möglichst genaue Simulationsmodelle von Generator, Verbrauchern und Batterie benötigt. Die Modelle wurden unter Verwendung eigener Messungen, Literaturangaben und Fittingverfahren parametrisiert. Für die Simulationsumgebung wurden MATLAB (Matrix Laboratory) und Simulink gewählt. MATLAB ist ein leistungsfähiges Programm, welches vornehmlich für numerische Berechnungen und Simulation von technischen Systemen genutzt wird [64]. Für Simulationen steht dabei Simulink, eine grafische Oberfläche zur Modellierung von physikalischen Systemen durch Signalflussbilder, zur Verfügung.

3.1 Generator

Ein Generator ist eine elektrische Maschine, welche mechanische Energie in elektrische Energie umwandelt. Mit dieser elektrischen Energie werden alle aktiven Verbraucher im Bordnetz versorgt. bei einem Energieüberschuss wird zusätzlich noch die Batterie geladen. Moderne Generatoren haben eine maximale Ausgangsleistung von etwa 3 kW [65]. Diese Leistung wird generiert, indem der Generator über den Keilriemen an der Kurbelwelle angetrieben wird. In Abbildung 3.1 ist das Schaltbild eines Generators dargestellt.
In konventionellen Fahrzeugen werden üblicherweise Klauenpolgeneratoren verwendet [22, 65], welche einer elektrischen Synchronmaschine entsprechen. Da allerdings die elektrischen Verbraucher und die Batterie mit Gleichstrom versorgt werden, wird der Wechselstrom des Generators gleichgerichtet. Dies erfolgt üblicherweise anhand einer B6-Brückenschaltung [22]. Da die Drehzahl des Generators proportional zur Motordrehzahl ist, muss verhindert werden, dass durch die sich ständig ändernde Drehzahl eine schwankende Ausgangsspannung entsteht. Daher ist der Generator mit einem Spannungsregler ausgestattet. Durch Variation des Erregerstroms wird die induzierte Spannung gesteuert. Dabei wird durch

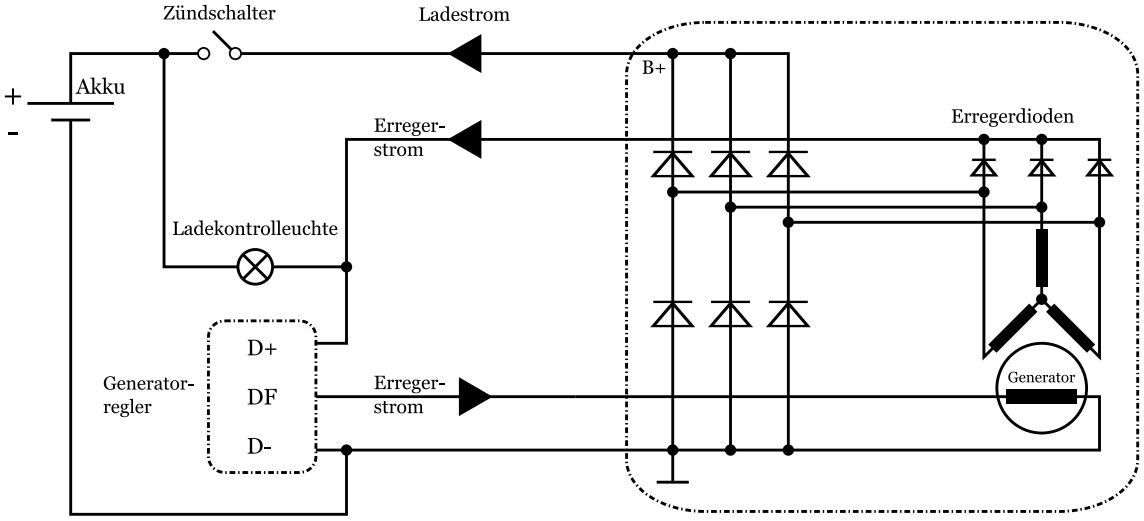


Abbildung 3.1: Schaltplan des Generators: Der Generatorregler regelt über den Erregerstrom die Ausgangsspannung des Generators. Über die B6-Brückenschaltung wird der Wechselstrom des Generators gleichgerichtet. Adaptiert aus [22]

ständiges Zu- oder Abschalten des Erregerstroms bei Unterschreiten des Sollspannungswertes die Erregung angeregt bzw. beim Überschreiten der Sollspannung die Erregung geschwächt. Somit werden die elektrischen Verbraucher und die Batterie mit einer konstanten Spannung versorgt. Bei plötzlichen Lastwechsel im Bordnetz besteht das Risiko hoher Spannungseinbrüche und Drehmomentsprünge im Antriebsstrang. Um dies zu vermeiden, wird der Erregerstrom des Generators langsam nachgeregelt. Dadurch kommt es bei der Leistungsbereitstellung vom Generator zu einer zeitlichen Verzögerung, die Load–Response–Zeit [22, 66, 67]. Die fehlende Leistung wird in dieser Zeit von der Batterie bereitgestellt [68].

Das Generatormodell muss für die Untersuchung der Spannungsstabilität im 12 V-Bordnetz das dynamische Verhalten des Generators möglichst genau nachbilden. Für den Generator wird in dieser Arbeit ein Kennlinien-basiertes Modell aufgebaut. Das Modell ist in Abbildung 3.2 dargestellt. Das Generatormodell besteht aus mehreren Teilmodellen. Für den Regler sind die Load-Response-Zeit t_{LR}, die Generatorsollspannung $U_{Gen,soll}$, die aktuelle Generatorspannung $U_{Gen,ist}$ und die Motordrehzahl n_{Mot} als Input–Parameter definiert. In diesem Block wird über die Ist-Spannung und der Soll-Spannung des Generators die Generatorauslastung $Gen_{Auslastung}$ und daraus der Erregerstrom $I_{Erreger}$ bestimmt. Des Weiteren wird die Dynamik des Generators anhand der Load-Response-Zeit begrenzt. Dadurch wird die Anstiegsgeschwindigkeit des Stromes bestimmt. Die Auslastung $Gen_{Auslastung}$, der Erregerstrom $I_{Erreger}$ und die Generatordrehzahl n_{Gen} werden in einen weiteren Teilblock übergeben. Hierbei wird der Generatorstrom I_{Gen} bestimmt. Diese Bestimmung beruht auf hinterlegte Kennfelder und Korrekturfunkionen. Als letztes wird der Generatorstrom I_{Gen} an eine gesteuerte Stromquelle übergeben. Der maximal mögliche Ausgangsstrom des

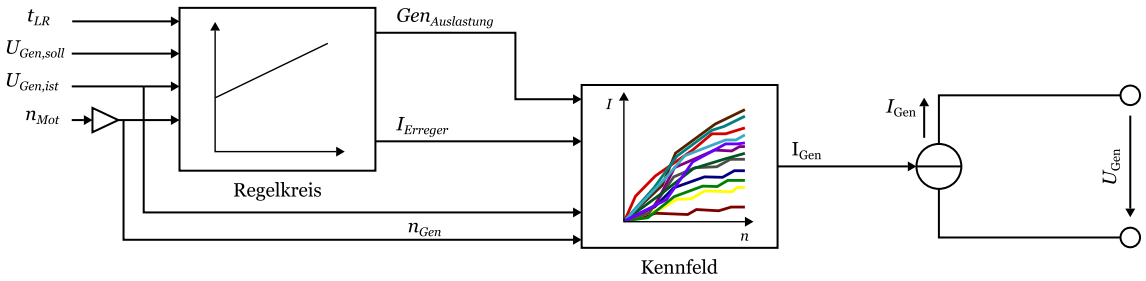


Abbildung 3.2: Modellierung des Generators: Als Input dienen die Load-Response-Zeit t_{LR}, die Generatorsollspannung $U_{Gen,soll}$, die aktuelle Generatorspannung $U_{Gen,ist}$ und die Motordrehzahl n_{Mot}. Aus dem Regler-Block kommen die Generatorauslastung $Gen_{Auslastung}$ und der Erregerstrom $I_{Erreger}$. Gemeinsam mit der Generatordrehzahl n_{Gen} zählen sie als Input-Parameter für das hinterlegte Generatorkennfeld.

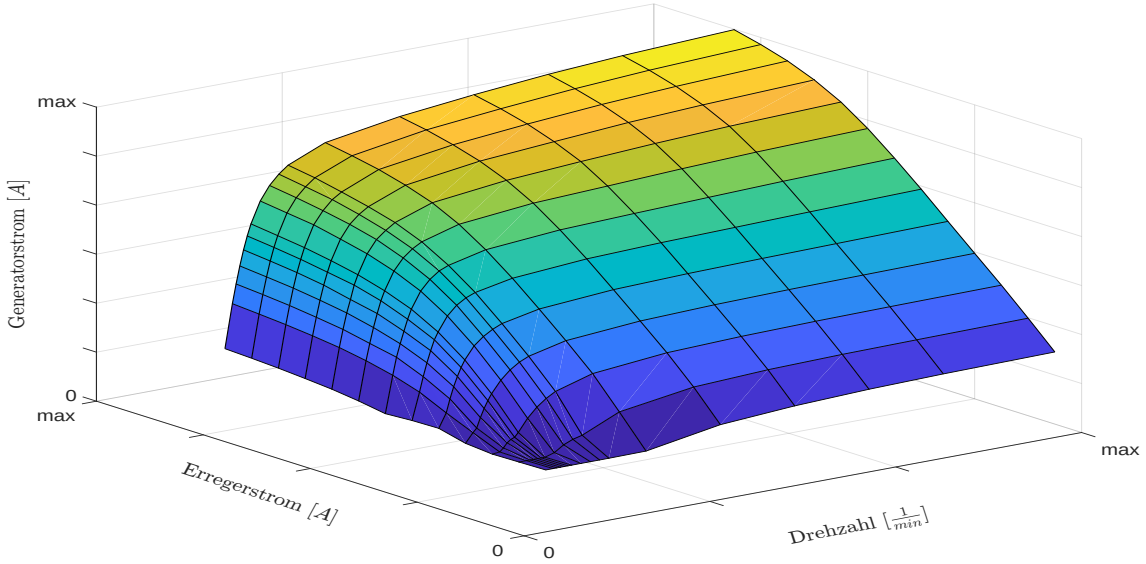


Abbildung 3.3: Generatorkennfeld: Dargestellt wird die Abhängigkeit des maximalen Generatorstroms von dem Erregerstrom und der Generatordrehzahl gemessen bei einer konstanten Generatorspannung.

Generators ist durch die hinterlegten Kennlinien festgelegt.

In Abbildung 3.3 ist das durch Messungen aufgenommene Kennfeld des Generators dargestellt. Darin wird die Abhängigkeit des maximalen Generatorstroms von der Generatordrehzahl und dem Erregerstrom deutlich.

3.2 Elektrische Verbraucher

Die Simulation von elektrischen Verbrauchern im 12 V-Bordnetz kann auf unterschiedliche Weise erfolgen. Viele Verbraucher sind komplexe Systeme, deren Stromverbrauch von vielen Parametern anhängt. Dies macht das physikalische Modell sehr komplex. Um den Modellierungsaufwand gering zu halten, können Simulationen auf Basis der jeweiligen

Tabelle 3.1: Liste der im 12 V–Bordnetz vorkommenden elektrischen Verbraucher und deren mögliche Modellierungsarten.

Verbraucher	**Modellierung**
Steuergeräte	1,3
Lichtelemente	1,2
Heizelemente	1,2
Nebenaggregate	1,3
Lüfter	1,3

Eigenschaften des Verbrauchers und entsprechend hinterlegten Kennlinien durchgeführt werden [69]. Profile für solche Verbrauchermodelle können beispielsweise aus Referenzmessungen in Fahrzeugen ermittelt werden.
In diesem Abschnitt werden die in dieser Arbeit verwendeten Modelle für die unterschiedlichen elektrischen Verbraucher im 12 V-Bordnetz erläutert. Da sich die Arbeit auf, für die Spannungsstabilität kritische Situationen, fokussiert, werden elektrische Verbraucher mit hohem Energiebedarf und Einfluss auf die Bordnetzspannung näher betrachtet. Die Verbrauchermodelle stehen in Interaktion mit anderen Bordnetzkomponenten, dem Energiemanagementsystem sowie der Umwelt. Neben dem rein physikalischen Verhalten wird auch die für das 12 V-Bordnetz relevante Logik modelliert. Somit reagieren die Modelle sowohl auf Umwelteinflüsse und die Interaktion der Insassen mit dem Fahrzeug, als auch auf Vorgaben des Energiemanagementsystems. Die Parametrierung der Verbrauchermodelle erfolgt dabei durch eigene Messungen an Referenzfahrzeugen und Literaturangaben. Abhängig von dem Verbraucherverhalten können die elektrischen Verbraucher in drei Kategorien unterteilt werden [9, 69, 70]:

1. Strom-basiert
2. Spannung- oder Widerstand-basiert
3. Leistung-basiert

Tabelle 3.1 stellt die möglichen Modelle für die unterschiedlichen Verbraucherarten dar. Dabei wird ersichtlich, dass durch einige wenige Grundmodelle ein großer Teil der relevanten elektrischen Verbraucher ausreichend genau abgebildet werden.

3.2.1 Strom-basierte Verbraucher

Der einfachste und schnellste Weg, einen elektrischen Verbraucher zu modellieren, ist eine stromgesteuerte Senke, wie in Abbildung 3.4 dargestellt. Elektrische Verbraucher, die unabhängig von der Spannung einen konstanten Strom verbrauchen oder über lange Zeiträume kleine Schwankungen aufweisen, können auf diese Weise modelliert werden. Aufgrund der geringen Stromaufnahme beeinflussen diese Art an Verbraucher das Spannungsniveau im 12 V-Bordnetz sehr gering. Das Stromprofil kann zeitabhängig, durch bei-

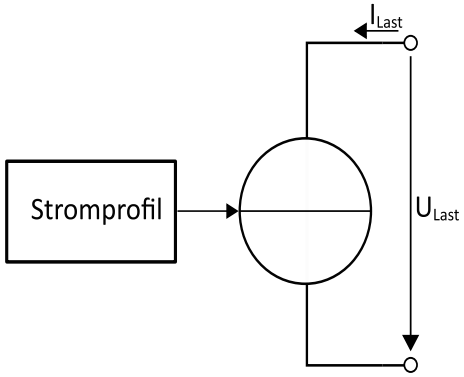


Abbildung 3.4: Struktur eines stromgesteuerten Verbrauchermodells: Das hinterlegte Stromprofil wird anhand einer Stromsenke eingespeist.

spielsweise im Modell hinterlegte Stromprofile oder konstant eingestellt werden. Unter Verwendung der Gleichung 3.1 kann das Modell in Simulink durch die in Abbildung 3.4 dargestellte Form repräsentiert werden:

$$I_{Aus} = I_{Stromprofil}\,(t) \tag{3.1}$$

Weitere Beispiele für Verbraucher, die auf diese Art modelliert werden können, sind LED-Beleuchtungen (Light Emitting Diode). Diese haben einen konstanten Stromverbrauch, welcher unabhängig von der Temperatur und Spannungslage ist [71].

3.2.2 Spannung-basierte Verbraucher

Spannungsgesteuerte Verbraucher haben eine Strom- beziehungsweise Leistungsaufnahme, die von der anliegenden Spannung abhängig ist. Hierzu gehören ohmsche Verbraucher sowie Verstellmotoren. Als ohmsche Verbraucher zählen im 12 V-Bordnetz Heizungs- und Lichtsysteme. Folgende Gleichung dient zur Beschreibung des Verbraucherstroms eines spannungsgesteuerten Verbrauchermodells:

$$I_{Aus} = \frac{U_{Ein} \cdot i'\,(t)}{u'\,(t)} = \frac{U_{Ein}}{R\,(t, \vartheta)} \tag{3.2}$$

Mit der angelegten Eingangsspannung U_{Ein} und den Parametern u' und i', die den ohmschen Widerstand R bestimmen, lässt sich der Ausgangsstrom I_{Aus} berechnen. Die Gleichung 3.2 zeigt, dass die Leistung des Verbrauchers proportional zur angelegten Spannung ist. Je höher die Spannung, desto höher der Stromverbrauch und somit die Verbraucherleistung. Daher können niedrige Spannungspegel zu einer Funktionseinschränkung führen. Bei der Auslegung solcher Komponenten müssen deshalb diese Zusammenhänge berücksichtigt werden. Verbraucher müssen so ausgelegt werden, dass sie ihre Funktion bei niedrigen Spannungslagen erfüllen können, ohne dass der Kunde diese Zusammenhänge wahrnimmt. Andererseits kann es bei einer zu hohen Spannungslage zu einem erhöhten Stromverbrauch und dadurch einer Überlastung der Komponenten kommen. Deshalb werden spannungs-

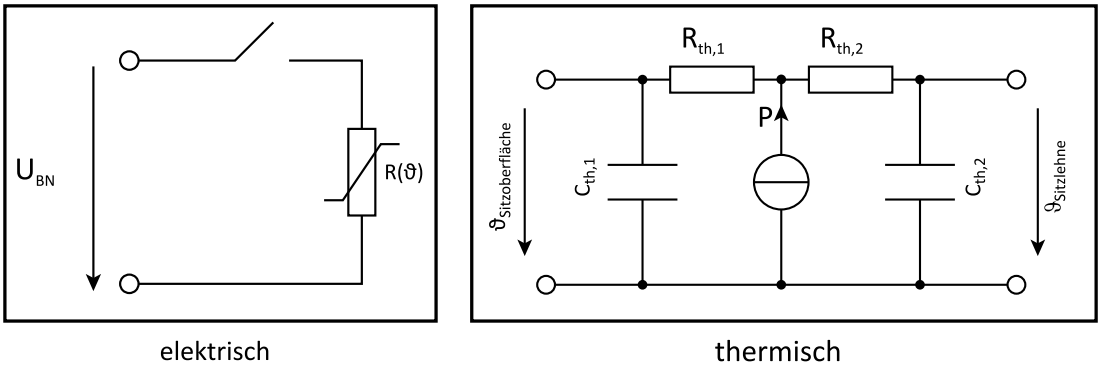


Abbildung 3.5: Blockschaltbild einer Sitzheizung: Auf der linken Seite ist das elektrische Modell dargestellt. Hier befindet sich ein temperaturabhängiger Heizwiderstand $R(\vartheta)$, der über einen Leistungsschalter mit dem Bordnetz verbunden ist. Auf der rechten Seite wird das thermische Modell abgebildet. Dieses besteht aus zwei Massen mit je einer spezifischen Wärmekapazität C_{th} und dem entsprechenden Wärmewiderstand R_{th}.

gesteuerte Verbraucher meist über ein PWM-Signal (Pulsweitenmodulation) angesteuert.

Ein weiterer Aspekt ist die Temperaturabhängigkeit des ohmschen Verbrauchers. Glüh- und Halogenlampen erreichen das thermische Gleichgewicht in wenigen Millisekunden [72]. Da diese kleine thermische Zeitkonstante keinen großen Einfluss auf den Gesamtenergieverbrauch hat, kann ihr Widerstand als ein konstanter Wert angenommen werden. Andererseits spielt die Temperaturabhängigkeit des Wiederstandes von Heizelementen, wie die Sitzheizung oder Heckscheibenheizung eine wichtige Rolle. Hier liegen die thermischen Zeitkonstanten bei mehreren Sekunden [73] und sind daher nicht zu vernachlässigen. Da der Widerstand temperaturabhängig ist, können diese Lasten bei gleichem Spannungsniveau unterschiedliche Leistungsaufnahmen haben. Um die Heizleistung konstant zu halten, wird der Verbraucher über einen Leistungsschalter periodisch ein- und ausgeschaltet. Aufgrund der großen thermischen Zeitkonstanten dieser Systeme stellt sich hierbei eine niedrige PWM-Frequenz ein. Auch Elektromotoren und LEDs werden auf diese Weise angesteuert, jedoch mit deutlich höheren Frequenzen.

Das in dieser Arbeit verwendete Modell zur Darstellung der physikalischen Eigenschaften der Sitzheizung ist in Abbildung 3.5 dargestellt. Auf der elektrischen Seite befindet sich ein temperaturabhängiger Heizwiderstand $R(\vartheta)$, der über einen Leistungsschalter mit dem Bordnetz des Fahrzeugs verbunden ist. Über diesen Schalter wird die Temperatur der Sitzheizung so geregelt, dass eine vordefinierte Soll-Temperatur erreicht und gehalten wird. Das thermische Modell besteht aus zwei Massen mit je einer spezifischen Wärmekapazität C_{th} und dem entsprechenden Wärmewiderstand R_{th}. Die in Widerstand umgesetzte elektrische Leistung wird auf der thermischen Seite des Modells als Wärmestrom P dargestellt.

Die Verlustleistung des Widerstandes wird mit folgender Gleichung berechnet:

$$P_N = I_N \cdot U_N = \frac{U_N^2}{R_N} \tag{3.3}$$

Dies wird bei der thermischen Modellierung verwendet, um die Temperatur des Leiters unter Verwendung des ersten Hauptsatzes der Thermodynamik zu berechnen. Die Temperatur wird im Modell unter Verwendung der folgenden Gleichung bestimmt:

$$T_n = T_{n-1} + \frac{1}{C_{th}} \left(\frac{U^2}{R\left(T_{n-1}\right)} - \frac{T_{n-1} - T_{Umgebung}}{R_{th}} \right) \tag{3.4}$$

Zur Berechnung der Temperatur wird die Summe aller Wärmeströme durch die Wärmekapazität dividiert. Der erste Term in dieser Gleichung beschreibt die in Wärme umgewandelte elektrische Energie. Der zweite Term stellt die Wärme, die vom Sitz an die Umgebung abgegeben wird, dar. Der Wärmewiderstand R_{th} steht für den Temperaturübergang zwischen Leiter und Umgebung. Bei der Sitzheizung entspricht die Umgebungstemperatur der der Insassen. Im Falle der Heckscheibenheizung ist sie identisch mit der Außentemperatur. Ausgehend von dieser Temperatur wird der Widerstandswert mit folgender Gleichung berechnet [74]:

$$R\left(\vartheta\right) = R_{20}\left(1 + \alpha\vartheta\right) \tag{3.5}$$

Die Stabilität des 12 V-Bordnetzes ist stark von den thermischen Komfortsystemen im Fahrzeug abhängig, da diese sehr leistungsstark und für eine lange Zeit aktiv sind. Aufgrund der hohen thermischen Zeitkonstanten werden Temperaturänderungen von den Insassen nicht sofort wahrgenommen. Daher spielen thermische Komfortsysteme eine wichtige Rolle beim Energie- und Leistungsmanagement. Wie in Abschnitt 2.2 beschrieben, können diese Verbraucher in kritischen Situationen degradiert oder abgeschaltet werden.

3.2.3 Leistung-basierte Verbraucher

Aufgrund der technischen Weiterentwicklung der Komponenten und der steigenden Anforderungen an die Systeme werden viele Aktoren mit einer Leistungsregelung ausgestattet. Dadurch wird sichergestellt, dass das System unabhängig von der Spannungslage mit der erforderlichen Leistung versorgt wird. In einem 12 V-Bordnetz macht dies Sinn, da hier eine variable Bordnetzspannung vorhanden ist. Somit sorgt die Leistungsregelung dafür, dass für die Verbraucher nur die tatsächlich benötigte Energie entnommen wird. Eine Vielzahl von Steuergeräten, Sensoren und Aktoren gehören zu dieser Kategorie. Sie erfüllen die meisten Komfortfunktionen, insbesondere im Bereich Infotainment, sowie Assistenz- und Sicherheitsfunktionen, wie etwa die Fahrdynamikregelung. Der Ausgangsstrom der Leistung-basierten Verbraucher wird durch folgende Gleichung berechnet:

$$I_{Aus} = \frac{U_N \cdot I_N}{U_{Ein}} = \frac{P_N}{U_{Ein}} \tag{3.6}$$

Das Verbrauchermodell hat eine konstante Leistungsaufnahme P_N, ermittelt aus Strom- und Spannungsmessungen. Die Spannung sollte dabei im Spannungsbereich $U_{max} > U_{Ein} >$

U_{min} liegen. Außerhalb dieses Spannungsbereichs weisen elektrische Verbraucher ein spannungsgesteuertes Verhalten auf.

Einen wichtigen Teil der Leistungsverbrauchern bilden die sogenannten Hochleistungsverbraucher. Wie in Kapitel 2.1.4 beschrieben, spielen sie bei der Betrachtung der kritischen Spannungslage im Bordnetz eine wichtige Rolle. Diese sind vor allem Aktoren aus dem Fahrwerkbereich, wie z.B. die elektrische Lenkkraftunterstützung, die dynamische Stabilitätskontrolle (DSC) oder die Hinterachsschräglaufregelung (HSR). Da bei der Aktivierung dieser Hochleistungsverbraucher mehrere Aktoren gleichzeitig arbeiten, kommt es zu einer Überlagerung der Leistungsaufnahmen und somit zu hohen Belastungen des 12 V-Bordnetzes. Diese Systeme sind so ausgelegt, dass der Aktuator nur innerhalb eines vorgegebenen Spannungsbereichs zuverlässig angesteuert werden kann. Sinkt die Spannung unter einen kritischen Wert, begrenzt der Regler die Leistung des Aktuators. Dadurch kann es jedoch zu sicherheitskritischen Zuständen kommen. Eine Begrenzung der Servolenkung würde eine Verhärtung der Lenkung verursachen. Solch ein Ereignis kann zu einem Unfall führen. Daher sollte bei der Auslegung des Bordnetzes darauf geachtet werden, dass auch bei kurzen Spitzenlasten eine ausreichende Spannung für den sicheren Betrieb aller Systeme gewährleistet wird.
Das Verhalten der Hochleistungsverbraucher kann auf verschiedene Weise modelliert werden. Eine genaue physikalische Abbildung jedes Verbrauchers erfordert aufgrund der vielen komplexen Zusammenhänge viel Aufwand. Um den Modellierungsaufwand zu reduzieren und Rechenzeit zu sparen, kann das Modell vereinfacht werden, indem die Hochleistungsverbraucher im Modell als Strom- oder Leistungsprofile hinterlegt werden.

In dieser Arbeit werden Hochleistungsverbraucher als Leistung-gesteuerte Stromsenken modelliert. Die Profile werden aus verschiedenen Fahrzeugmessungen extrahiert und in das Modell eingebunden. Über ein Aktivierungssignal werden diese bei Simulationen aktiviert. Für die Zwecke dieser Arbeit ist die beschriebene Modellierungsart, dargestellt in Abbildung 3.6, ausreichend.
Abbildung 3.7 zeigt die einzelnen Leistungen der verwendeten Hochleistungsverbraucher und die Summenleistung. Die Leistungsprofile werden aus Messungen ermittelt, bei deinen ein Ausweichmanöver durchgeführt wurde. Hierbei werden die elektrische Lenkkraftunterstützung (EPS), die Hinterrachsschräglaufregelung (HSR) und die dynamische Stabilitätskontrolle (DSC) als Hochleistungsverbraucher bezeichnet. Wie die schwarze Linie zeigt, überlagern sich die einzelnen Leistungen der Verbraucher und führen zu einer hohen Belastung des Bordnetzes. Das dargestellte Profil ist 4 s lang. Der Zeitraum, in dem der größte Leistungsbedarf auftritt, dauert weniger als 1 s. Die maximale Summenleistung der drei oben genannten Verbraucher beträgt in diesem Bereich $2,47\ kW$.

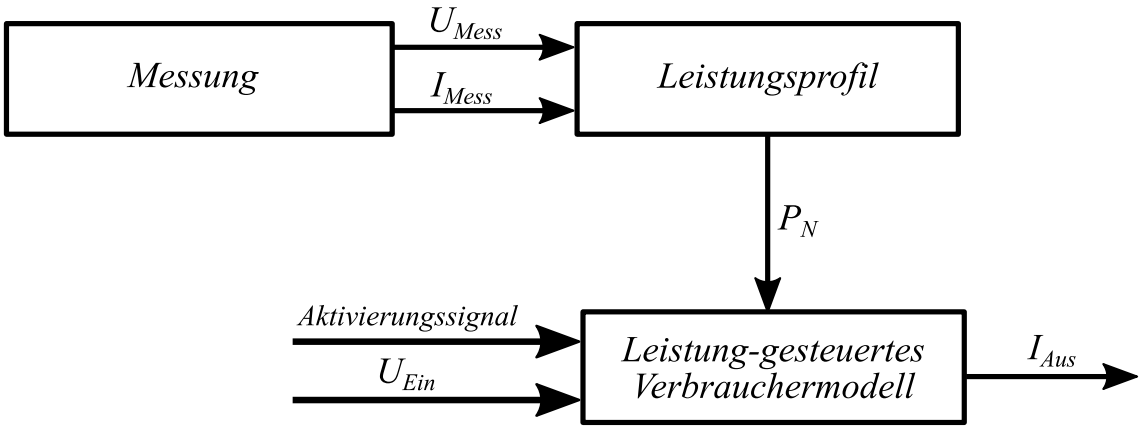


Abbildung 3.6: Blockschaltbild Hochleistungsverbraucher: Aus Fahrzeugmessungen werden durch die gemessene Spannung und den Strom das Leistungsprofil der Hochleistungsverbraucher berechnet. Diese Profile werden in das Modell hinterlegt. Mit einem Aktivierungssignal und der aktuell anliegenden Spannung wird der Ausgangsstrom der Verbraucher bestimmt.

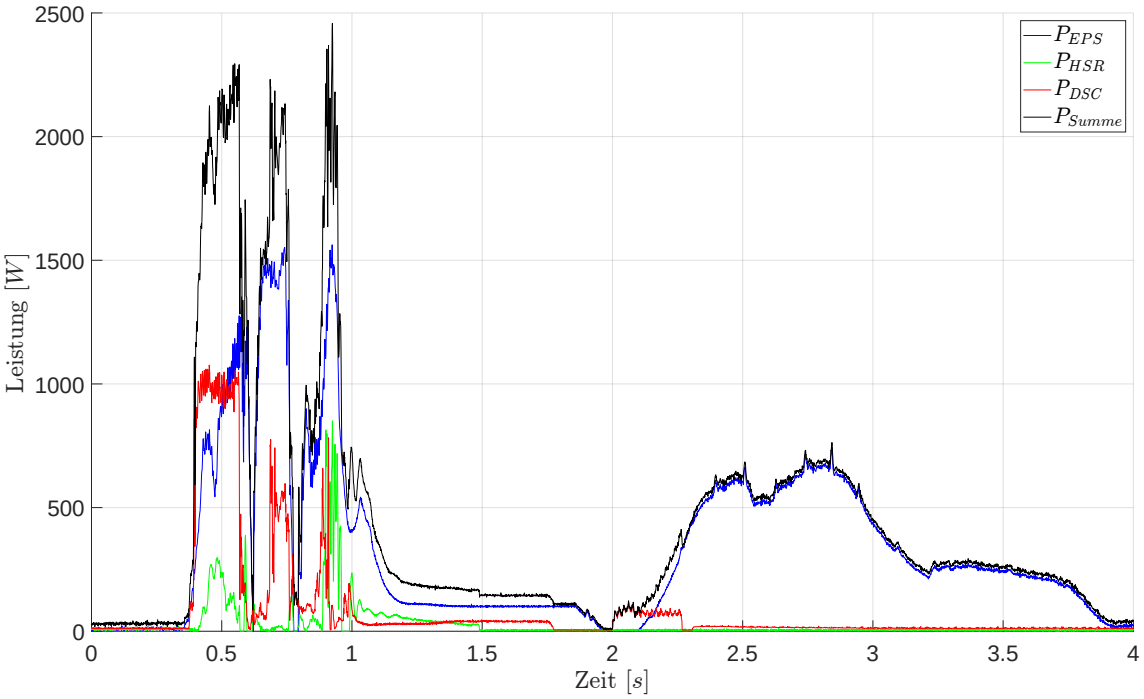


Abbildung 3.7: Darstellung der Leistung von den Hochleistungsverbrauchern. Die Addition der Leistungen vom EPS (blau), HSR (grün) und DSC (rot) ergibt die Gesamtleistung (schwarz) der Hochleistungsverbraucher.

3.3 12 V–Batterie

In diesem Abschnitt werden zunächst die Ansätze zur Modellierung von Batterien dargestellt. Im Anschluss daran erfolgt die Beschreibung des Simulationsmodells für die verwendete Batterie sowie die Erläuterung der Parametrisierung des Modells.

3.3.1 Ansätze zur Modellierung von Batterien

Mit der Modellierung des elektrischen Batterieverhaltens soll eine möglichst effiziente und genaue Nachbildung der realen Batterie erzielt werden. Um dieses Ziel zu erreichen ist eine genaue Wiedergabe aller relevanten Polarisationsabläufe erforderlich. Für die Darstellung dieser Effekte bieten sich unterschiedliche Modellansätze, die sich im Wesentlichen in folgende drei Kategorien einteilen lassen [75, 76, 77, 78]:

- physikalisch–chemische Modelle (White–Box–Modelle)
- mathematische bzw. empirische Modelle (Black–Box–Modelle)
- semi–empirische Modelle (Grey–Box–Modelle)

Physikalische Modelle versuchen das innere Zellverhalten, durch die Berechnung umfangreicher und voneinander abhängigen Gleichungen möglichst realitätsnah nachzubilden. Bei diesen Modellen wird ein hohes Maß an Vorwissen sowie viel Rechenleistung zur Berechnung der einzelnen Effekte benötigt. Hierfür ist die Ermittlung zahlreicher Parameter, unter anderem die Diffusionskoeffizienten der einzelnen Materialien sowie deren Abhängigkeiten von Temperatur und Alterung, äußerst komplex [75, 79].
Bei empirischen Ansätzen, wie etwa dem Peukertansatz [80], wird das Batterieverhalten durch lediglich mathematische Gleichungen ohne physikalische Hintergründe beschrieben. Diese Modelle können unter anderem zu Modellen mit neuronalen Netzen [81, 82, 83, 84, 85, 86] oder einer Fuzzy–Logik [87] aufgebaut werden.
Einen Mittelweg zwischen den beiden beschriebenen Ansätzen stellen semi–physikalische Modelle dar. Damit können elektrochemische Prozesse insbesondere durch das elektrische Ersatzschaltbild (ESB) auf ihre Komplexität reduziert werden, womit eine Beschreibung der Polarisation und der Ruhespannung möglich ist. Diese Modelle bieten somit einen guten Kompromiss zwischen dem Modellierungs– und Rechenaufwand, der Genauigkeit bei der Wiedergabe des Spannungsverhaltens sowie der physikalischen Aussagekraft der Batterieparameter [88]. In der Literatur werden ESB–Modelle in impedanzbasierte ESB–Modelle und äquivalente elektrische Ersatzschaltbildmodelle aufgeteilt. Beim erstgenannten Ansatz werden die Parameter mit Hilfe von Messergebnissen aus der elektrochemischen Impedanzspektroskopie [89, 90, 91, 92, 93, 94, 95] bestimmt. Die Bestimmung der Parameter von äquivalenten elektrischen Ersatzschaltbildern erfolgt dahingegen z.B. anhand von Strompulsen [96, 97, 98, 99, 100, 101].

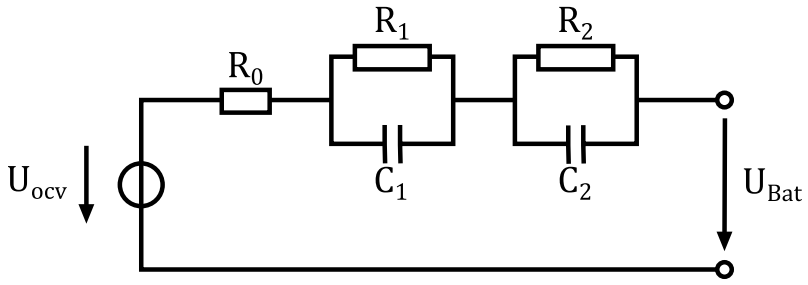


Abbildung 3.8: Elektrisches Ersatzschaltbild einer 12 V–Batterie: Das Klemmenverhalten wird durch eine Gleichspannungsquelle U_{OCV}, einem Innenwiderstand R_0 und zwei RC–Gliedern modelliert.

Da für die Betrachtung der Spannungsstabilität im Bordnetz die Vorgänge im Inneren der Batterie vernachlässigt werden können, ist für diese Arbeit lediglich der Verlauf der Klemmenspannung relevant. Daher wurde hier ein ESB–Modell eingesetzt. Für die Modellierung von AGM–Batterien haben sich in der Literatur verschiedene Varianten an elektrischen Ersatzschaltbildern bewährt.

Das Rint–Modell beschreibt das Verhalten der Batteriespannung mithilfe der Ruhespannung und eines Widerstands [88, 102]. Dabei wird der Spannungsabfall nur durch ein Widerstand beschrieben. Eine Abbildung des dynamischen Verhaltens ist mit solch einem vereinfachten Modell nicht möglich. Um alle ablaufenden Prozesse zu berücksichtigen, muss der Spannungseinbruch unter Last über einige Sekunden gemessen werden. Dadurch kann schließlich der Widerstand berechnet werden. In der Automobilbranche wird für die Bestimmung des Innenwiderstandes meist ein Zeitbereich von 5 s oder 10 s gewählt. Der Widerstand lässt sich damit über die Spannungsänderung an der Batterie und dem Stromsprung innerhalb des Zeitbereiches berechnen.

Beim Thevenin–Modell handelt es sich um ein Relaxationsmodell erster Ordnung. Hier wird zusätzlich zum Widerstand vom Rint–Modell ein RC–Glied [103] seriell verschaltet [88, 102, 104, 105, 106]. Dadurch kann das dynamische Verhalten beschrieben werden, welches allerdings mehrere Prozesse zusammenfasst.

Das Thevenin–Modell kann um ein zusätzliches RC–Glied erweitert werden, welches auch als dual–polarization-Modell (DP–Modell) bezeichnet wird [88, 107, 108, 109]. Dadurch kann ein weiterer Prozess dargestellt werden. Somit können detailliertere Aussagen über das physikalische Verhalten der Batterie getroffen werden.

Abbildung 3.8 zeigt das in dieser Arbeit verwendete Batteriemodell einer AGM–Batterie. Das Modell besteht aus einer Spannungsquelle U_{OCV}, einem Innenwiderstand R_0 und zwei in Reihe geschalteten RC–Gliedern. Dabei beschreibt U_{OCV} die Leerlaufspannung und das statische Verhalten der Batterie. Es symbolisiert, dass zwischen der negativen und der positiven Elektrode im Leerlauf- und Gleichgewichtszustand eine messbare Spannung besteht. Die Zeit, die die Batterie benötigt, um nach der Anregung wieder eine konstante Ruhespannung zu erreichen, wird als Relaxationszeit bezeichnet. Das Verhalten der Span-

nungsquelle wird durch die Temperatur und den Ladezustand beeinflusst.

$$U_{OCV} = f(T, SoC) \tag{3.7}$$

Der rein ohmsche Widerstand R_0 stellt den eigentlichen Innenwiderstand der Batterie dar und ist nur vom Schließen bis zum Öffnen des Stromkreises relevant. Je nach Richtung des Stromflusses kann dies beim Entladen der Batterie einen plötzlichen Abfall der Batterieklemmenspannung oder beim Laden der Batterie einen Anstieg der Batterieklemmenspannung verursachen. Die folgenden RC–Glieder bestehen jeweils aus einer Parallelschaltung mit einem ohmschen Widerstand R_1 bzw. R_2 und einem Kondensator C_1 bzw. C_2. Die beiden RC–Glieder sind nur beim Schließen oder Öffnen des Stromkreises aktiv. Sie reagieren auf ein plötzliches Ansteigen oder Abschalten der Last mit einer zeitlich verzögerten Spannungszunahme oder Spannungsabnahme. Dieses Verhalten bildet den Einschwingvorgang der Batteriespannung ab. Die Stärke der Zeitverzögerung und der Spannungsänderung hängt von der Wahl der Widerstands– und Kapazitätswerte ab.
Das Thevenin–Modell lässt sich außerdem durch eine Vielzahl zusätzlicher RC–Glieder erweitern. Dadurch wird eine bessere Annäherung an das reale Spannungsverhalten erreicht. Jedoch steigt mit zunehmenden RC–Gliedern auch die Anforderung an Rechenleistung. So kann zwar beispielsweise ein Modell mit vier RC–Gliedern detailreiche Prozesse nachbilden, allerdings nur bei vergleichsweise sehr langen Simulationszeiten. Deshalb muss hier ein Kompromiss zwischen Genauigkeit des Simulationsmodells und des Berechnungsaufwandes gefunden werden [75, 110].

3.3.2 Parameteridentifikation

In diesem Abschnitt wird die Herangehensweise zur Bestimmung der Modellparameter vorgestellt. Im ersten Messschritt wird die tatsächlich verfügbare Kapazität der Batterie bestimmt. Für die Ermittlung der Kapazität erfolgt eine Konstantstrommessung an der Batterie. Bei diesem Vorgang wird die Batterie zunächst auf 100 % SoC und einer festgelegten Temperatur konditioniert. Anschließend wird die Batterie nach einer Ruhephase mit einem konstanten Strom entladen. Während der Entladung wird der Spannungs– und Stromverlauf aufgezeichnet. Sobald die Entladeschlussspannung erreicht ist, gilt die Messung als beendet. Die ermittelte Kapazität bildet die Grundlage für die Berechnung und Einstellung des Ladezustandes aller weiteren Untersuchungen. Mithilfe dieser Messung wird zudem die verfügbare Ladungsmenge der Batterie bei unterschiedlichen Temperaturen untersucht. Die Tests wurden mit einer 50 Ah, 60 Ah, 80 Ah und 105 Ah Batterie für die Temperaturen von 25 °C, 0 °C, −10 °C und −20 °C durchgeführt. Nach der Kapazitätsmessung werden im nächsten Schritt die Rückrelaxationsmessungen zur Parameteridentifikation des ESB´s durchgeführt. Wie in Kapitel 3.3.1 beschrieben, gibt es mehrere Methoden die elektrochemischen EBS–Parameter zu bestimmen. Eine häufig genutzte Methode ist hierbei die Konstantstrommessung. Bei diesem Verfahren erfolgt bei einer festgelegten Temperatur innerhalb einer Messung für den SoC–Bereich 95 % bis 5 % je-

weils eine Relaxation. Dies ist in der Abbildung 3.9 dargestellt. Hierbei lässt sich die Relaxationsmessung für jeden SoC–Bereich in zwei Phasen unterteilen:

1. Entladung der Batterie
2. Relaxation der Spannung zu einem neuen Gleichgewichtszustand

Während der Relaxationsmessung wurde die Batterie schrittweise um 5 % SoC entladen. Um irreversible Schäden zu vermeiden und somit die sicherheitskritische Ober– bzw. Unterspannung nicht zu über– bzw. unterschreiten, wurden die SoC–Grenzwerte 100 % und 0 % bewusst ausgelassen. Somit kann mit der Aufzeichnung des Spannungs– und Stromverlaufs nach jedem Stromsprung die Spannungsantwort der Batterie näher betrachtet werden. Für eine dynamische Abbildung der Batteriespannung bei hohen Stromflanken wurde an der Messtechnik für die Aufzeichnung der schnellen Polarisationsabläufe eine Abtastrate von 1 kHz gewählt.

Mithilfe der Spannungsantwort können die Modellparameter bestimmt werden. Die Parametrierung kann hierbei in der Phase während des Stromflusses oder nach Ende des Stromimpulses im Bereich der Relaxation erfolgen. Über die Relaxationsmessung kann als erstes die Leerlaufspannung U_{OCV} und der Innenwiderstand R_0 für jeden SoC–Bereich bestimmt werden. Dazu wird der Spannungsverlauf der gesamten Messung zunächst abschnittweise nach SoC getrennt. Während der Entladephase ändert sich der SoC um 5 %. Die Spannung sinkt dabei unter konstantem Strom immer weiter. Darauf folgt der Relaxationsvorgang der Batterie, bis die Ruhespannung erreicht ist. Als erster Parameter kann die ladezustandsbezogene Leerlaufspannung ermittelt werden, die am Ende jeder Relaxationsphase erfasst wird. Um einen Gleichgewichtszustand zu erreichen, in dem alle Prozesse abklingen, muss der Relaxationsprozess länger anhalten, um die Ruhespannung möglichst genau zu bestimmen. Endet die Relaxationsphase vorzeitig, bevor sich die Batterie im Gleichgewicht befindet, wird die Leerlaufspannung gemessen, die in etwa dem tatsächlichen Leerlaufspannungswert entspricht [79, 111]. Da die Relaxationsmessung mit 19 Stützstellen und einer Ruhephase im Zeitbereich von 2 h bis 6 h mehrere Tage andauert, wurde hierfür eine kürzere Zeit gewählt. Um alle Messungen innerhalb des Zeitplans durchführen zu können, wurde eine einstündige Pause festgelegt.
Nach der Leerlaufspannung U_{OCV} wird anhand der Relaxationsmessung der Innenwiderstand R_0 bestimmt. Wie dem Spannungsverlauf in Abbildung 3.9 zu entnehmen ist, steigt die Spannung in den ersten Sekunden nach Beendigung des Entladevorgangs stark an, danach fällt die Kurve langsam mit der Zeit ab. Die Spannungsänderung nach der Entladung kann durch den Innenwiderstand R_0 beschrieben werden. Der Innenwiderstand lässt sich nach dem Ohmschen Gesetz wie folgt berechnen:

$$R_0 = \frac{U_{R_0}}{I_{Impuls}} = \frac{U_{t_{R_0}} - U_{min}}{I_{Impuls}} \tag{3.8}$$

Die Spannung U_{R_0} ergibt sich aus der Differenz zwischen einem zeitlich festgelegtem Span-

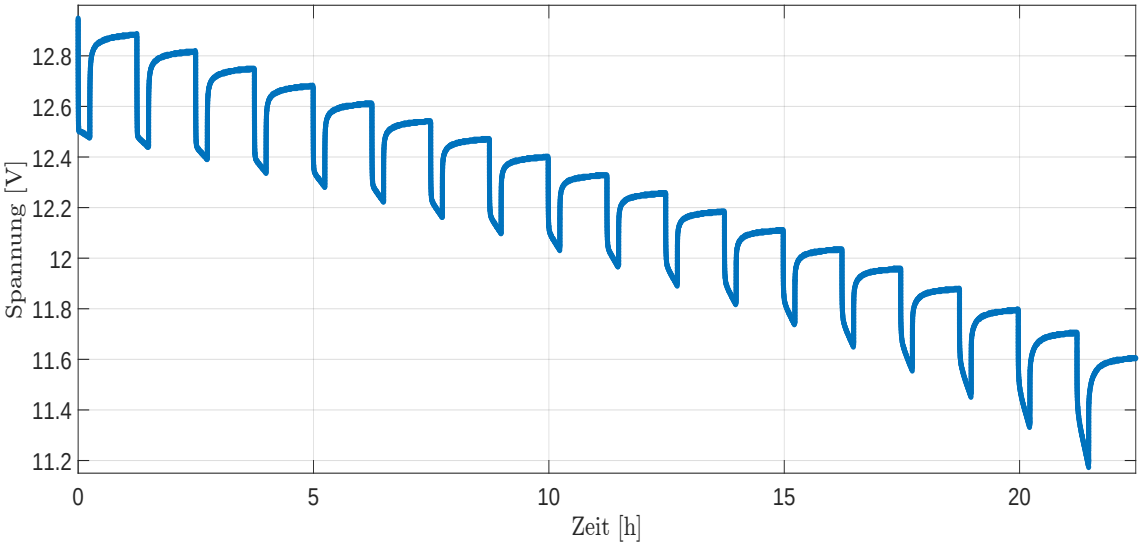


Abbildung 3.9: Spannungsverlauf einer 80 Ah Batterie während der Entladung. Die Batterie wird mit einem konstanten Strom entladen, wobei der Ladezustand schrittweise von 95 % auf 5 % reduziert wird. Der Spannungsverlauf zeigt, wie die Batteriespannung im Laufe der Entladung abnimmt.

nungsmesspunkt $U_{t_{R_0}}$ nach dem Stromsprung und der minimalen Spannung U_{min} kurz vor Ende der Entladung. Der Strom I_{Impuls} entspricht dem konstantem Stromwert.

Nachdem bereits die Leerlaufspannung und der Innenwiderstand bestimmt wurden, sind die Variablen R_1, τ_1, R_2 und τ_2 der Überspannungstherme $\mathrm{U}_{\mathrm{RC}_1}(\mathrm{t})$ und $\mathrm{U}_{\mathrm{RC}_2}(\mathrm{t})$ noch zu ermitteln. Um den dynamischen Spannungsverlauf der Überspannungen simulieren zu können, wurde zur Parametrierung der RC–Glieder das Pulse–Fitting Verfahren angewandt. Hierzu werden für jeden einzelnen SoC–Schritt aus der gemessenen Relaxationskurve die Widerstands–, Kapazitätswerte und die Zeitkonstanten der RC–Glieder bestimmt. Um die passenden Zeitkonstanten zu finden, werden unterschiedliche Zeitbereiche gefittet. Für die Beschreibung des gesamten Batteriespannungsverhaltens werden für das DP–Modell folgende Gleichungen verwendet:

$$U_{Bat}(t) = U_{OCV} + U_{R_0} + U_{RC_1}(t) + U_{RC_2}(t) \tag{3.9}$$

$$U_{Bat}(t) = U_{OCV} + R_0 \cdot I_{Bat} + R_1 \cdot I_{Bat} \cdot (1 - e^{-\frac{t}{\tau_1}}) + R_2 \cdot I_{Bat} \cdot (1 - e^{-\frac{t}{\tau_2}}) \tag{3.10}$$

Beim Curve–Fitting handelt es sich um eine mathematische Optimierungsmethode zur Bestimmung von unbekannten Parametern einer vorgegebenen Funktion mit dem Ziel der Beschreibung des Kurvenverlaufs. Während dem ablaufenden Prozess werden über mehrere Approximationsschritte die Parameter sowie die Abweichung zwischen der Funktion und der Messung berechnet und verglichen. Dadurch kann eine möglichst gute Annäherung zur Messung erzielt werden.

4 Fahrzeugmessungen und Datenvorverarbeitung

In diesem Abschnitt werden zunächst kritische Fahrmanöver vorgestellt. Anschließend wird der Prozess der Datenaufbereitung sowie die Analyse von den realen Fahrzeugmessungen beschrieben. Dadurch wird die Problematik der Spannungsstabilität im 12 V-Bordnetz anhand von realen Messungen verdeutlicht. Daraus resultierende Daten werden in das Simulationsmodell eingespeist, sodass der RL-Agent während des Trainings mit kritischen Spannungslevel konfrontiert wird und aus diesen lernen kann. Die Rohdaten beinhalten Messungen aus einem Erprobungsfahrzeug. Dabei handelt es sich um ein konventionelles Fahrzeug mit einem 12 V-Bordnetz. Die generierten Daten repräsentieren unterschiedliche Fahrmanöver.

4.1 Fahrmanöver

Ein Energieungleichgewicht zwischen Angebot und Nachfrage kann zu Spannungsinstabilitäten im 12 V-Bordnetz führen. Wenn das Ungleichgewicht zu einer Beeinträchtigung der Funktionen oder einem teilweisen Versagen führt, wird der Zustand als kritisch bezeichnet. Um die Spannungsstabilität näher zu analysieren, werden in diesem Abschnitt verschiedene kritische Belastungsmanöver näher beschrieben. Vor der Simulation der Bordnetzstabilität müssen daher unterschiedliche Fahrzeugmessungen durchgeführt und analysiert werden. In den Versuchsfahrten wurden vordefinierte Manöver gefahren, die einen hohen Spannungseinbruch im Bordnetz verursachen. Hierbei handelt es sich um dynamische Fahrmanöver, die vor allem durch den Einsatz der Komponenten im Fahrdynamik- und Fahrwerksbereich dazu führen, dass die Verbraucherdynamik die Generatordynamik übersteigt. Die Strom- und Spannungsverläufe werden während der Fahrten direkt an den Komponenten gemessen. Im Folgenden werden Manöver, die zu Instabilitäten im 12 V-Bordnetz führen können, näher erläutert.

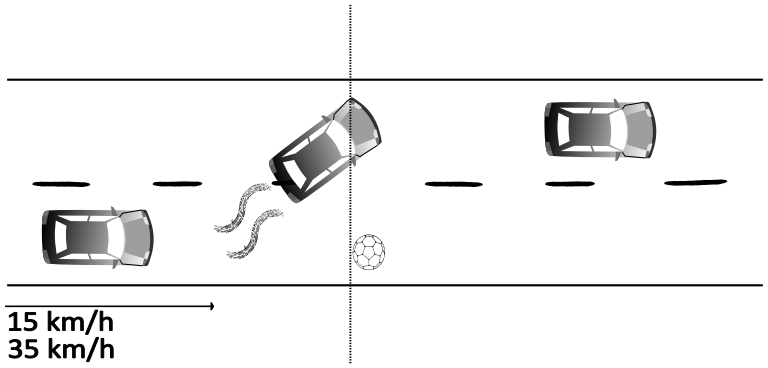
Ausweichmanöver mit 15 $\frac{\mathbf{km}}{\mathbf{h}}$ bzw. 35 $\frac{\mathbf{km}}{\mathbf{h}}$
Bei einem Ausweichmanöver wird während einer Geradeausfahrt versucht, plötzlich einem Objekt auszuweichen. In Abbildung 4.1(a) ist dieses Manöver dargestellt. Dabei werden aufgrund der gleichzeitigen Lenk- und Bremsbewegung sowohl das Lenkunterstützungssystem als auch die Fahrdynamikregelung aktiviert. Durch die Überlagerung dieser Systeme und einer hohen Grundlast kommt es zu einem hohen Spannungsabfall im Bordnetz. Das Manöver wird mit 15 $\frac{km}{h}$ und 35 $\frac{km}{h}$ gefahren. Durch die niedrige Generatordrehzahl bei geringen Geschwindigkeiten kann der Generator nur eine geringe Leistung zur Verfügung stellen. Die Erweiterung des Manövers um eine niedrige Geschwindigkeit führt dadurch zu einer Verschärfung des kritischen Zustandes.

Slalomfahrt
Bei der Slalomfahrt geht es darum, Eingriffe der Fahrwerksregelsysteme zu erzwingen. Hierbei werden Pylonen in 18 Meter Abständen aufgestellt [69, 112]. Der Fahrer versucht diese wechselseitig zu umfahren. Das Manöver ist in Abbildung 4.1(b) dargestellt. Durch das ständige Lenken nach links und rechts wird das elektrische Lenksystem EPS aktiviert. Zudem versucht das dynamische Stabilitätssystem das Fahrzeug stabil zu halten. Die hohen Stromspitzen der einzelnen Systeme führen auch hier zu einer Überlagerung und somit zu Spannungsschwankungen im 12 V-Bordnetz.

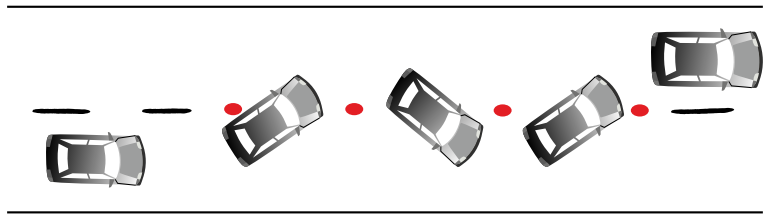
Wenden in 3 Zügen
Bei einem Wendemanöver treten sehr hohe Leistungsspitzen auf. Dabei wird versucht in drei Zügen in die entgegengesetzte Richtung zu fahren. Das Manöver, dargestellt in Abbildung 4.1(c), wird in geringen Geschwindigkeiten ausgeführt. Hier muss mehrmals gelenkt werden. Aufgrund der niedrigen Generatorleistung und der erhöhten Lenkmomente treten sehr hohe Stromspitzen auf, die zu einer Spannungsschwankung im Bordnetz führen.

Vollbremsung
Bei diesem Manöver wird bei einer Geradeausfahrt mit konstanter Geschwindigkeit eine plötzliche Vollbremsung durchgeführt. Das Manöver ist in Abbildung 4.1(d) dargestellt. Bei dem Bremsvorgang werden unterschiedliche Sicherheitssysteme aktiviert. Dazu zählen das Antiblockiersystem (ABS), die Antischlupfregelung (ASR) und Dynamic Stability Control (DSC). Als Folge davon steigt der Leistungsbedarf im Bordnetz sehr stark.

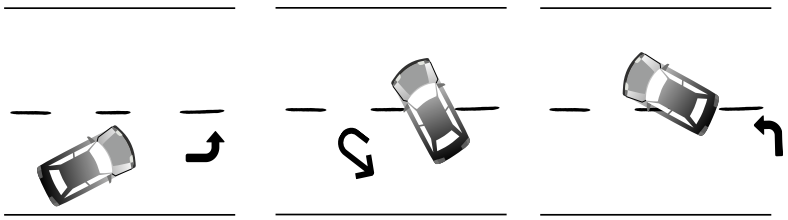
Während der Fahrt wurden alle Komfortverbraucher auf die maximale Stufe eingestellt. Somit entstand zusätzlich zu den Manöver-spezifisch aktivierten Hochleistungsverbrauchern eine hohe Grundlast.



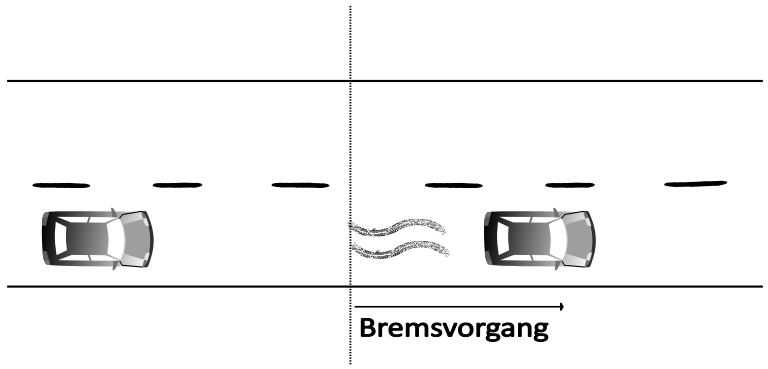
(a) Ausweichmanöver 15 $\frac{km}{h}$ bzw. 35 $\frac{km}{h}$



(b) Slalomfahrt



(c) Wenden in 3 Zügen



(d) Vollbremsung

Abbildung 4.1: Unterschiedliche Fahrmanöver: Dargestellt werden fünf unterschiedliche Fahrmanöver, die zu einem starken Spannungseinbruch im 12 V-Bordnetz führen. Das Ausweichmanöver wird bei zwei unterschiedlichen Geschwindigkeiten durchgeführt.

4.2 Fahrzeugmessungen und Datenvorbereitung

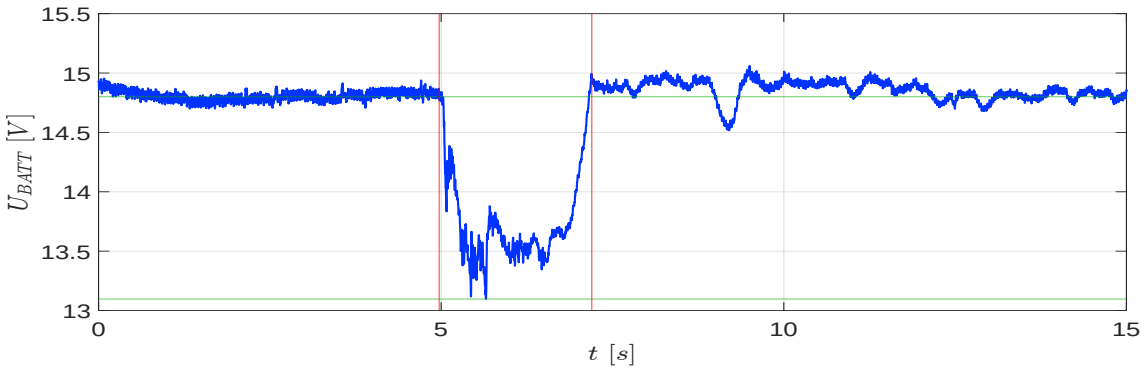
Diese Messungen wurden in einem Mittelklassefahrzeug mit einem 12 V-Bordnetz aufgezeichnet. Das Fahrzeug ist mit einem EPS und einem Bremsassistenten (IB, Intelligent Brake) ausgestattet. Diese gelten als Hochleistungsverbraucher. Im Folgenden werden die im vorherigen Abschnitt dargestellten fünf Typen als Manöver und die jeweilige Fahrt eines Manövers als Szenario bezeichnet. Um die Grundlast im 12 V-Bordnetz zu erhöhen, wurden während der Fahrten Komfortverbraucher wie die Sitzheizungen, Klimagebläse und Heckscheibenheizung auf die maximale Stufe geschaltet. Durch die Überlagerung der Ströme entsteht somit ein Worst-Case-Szenario. Die bei den Messungen verwendete Ausstattung des Testfahrzeugs ist in Tabelle 4.1 dargestellt.

Tabelle 4.1: Auflistung der Ausstattungsparameter und deren Werte vom Testfahrzeug.

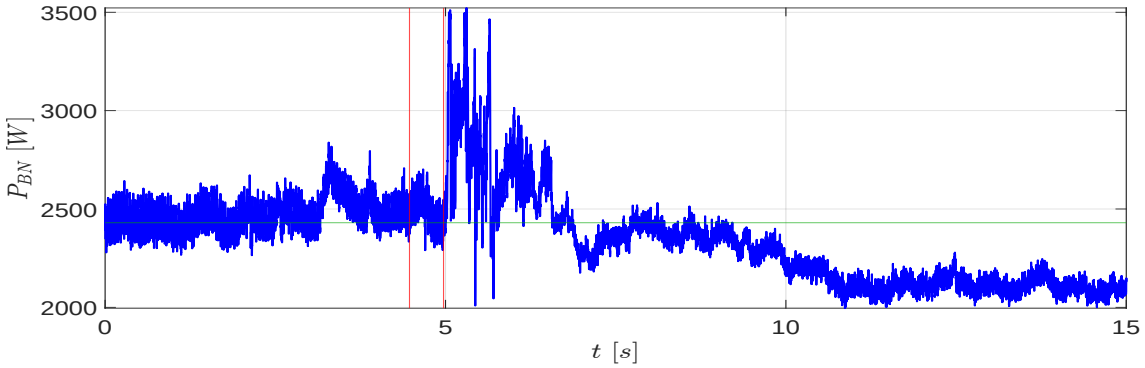
Parameter	**Wert**
Batterietyp	AGM
Nennspannung Batterie	12 V
Nennkapazität Batterie	50 Ah
Maximalstrom Batterie	570 A
Nennstrom Generator	180 A
Maximalstrom Generator	210 A

Im Folgenden wird der Prozess der Datenvorverarbeitung und somit die Messdatenanalyse beschrieben. Die Rohdaten werden für das Training und die Validierung der Neuronalen Netze vorbereitet. Zunächst werden diese Rohdaten mit einer Software visualisiert. Mit Hilfe dieser Software werden aufgezeichnete Daten von Steuergeräten und gemessene Analogwerte, wie Strom und Spannung, mit einer Abtastrate von 1 kHz exportiert. Da innerhalb einer Messung das Fahrmanöver mehrmals durchgeführt wurde, werden daraus mehrere Dateien derselben Länge erzeugt. Die Auswertung der Ausschnitte erfolgt dann in Matrix Laboratory. Dies macht die Szenarien untereinander vergleichbar, da für jeden Ausschnitt die selben Auswertungsregeln angewendet werden. Bei der Auswertung liegt der Fokus auf dem verursachten Spannungseinbruch im Bordnetz. Hierbei werden für alle Szenarien die Anfangs- und Minimalwerte der Batteriespannung, der SOC sowie die Tiefe und Dauer des Spannungseinbruchs gemeinsam mit der Bordnetzleistung ermittelt. In Abbildung 4.2 wird anhand eines Ausweichmanövers mit 35 $\frac{km}{h}$ die Analyse der Messung dargestellt.

In der Abbildung 4.2 (a) wird die Batteriespannung U_{BATT} während des Ausweichmanövers dargestellt. Für die Analyse der Spannungsstabilität im 12 V-Bordnetz ist die Tiefe ΔU_{BATT} und die Dauer Δt des Spannungseinbruchs relevant. Anhand der roten Linien wird die ermittelte Dauer verdeutlicht. Die grünen Linien dagegen zeigen die Tiefe des Spannungsein-



(a) Analyse der Batteriespannung U_{Batt}



(b) Analyse der Bordnetzleistung P_{BN}

Abbildung 4.2: Automatisierte Analyse der Messung am Beispiel eines Ausweichmanövers mit 35 $\frac{km}{h}$. In der oberen Abbildung wird der starke Einbruch der Batteriespannung dargestellt. Die roten Linien kennzeichnen die Dauer des Einbruchs, während die grünen Linien die Einbruchstiefe darstellen. Unten ist die Leistung abgebildet. Die roten Linien berechnen einen Durchschnitt aus den 500 vorangegangenen Messwerten der Bordnetzleistung. Die grüne Linie stellt diesen Durchschnitt dar.

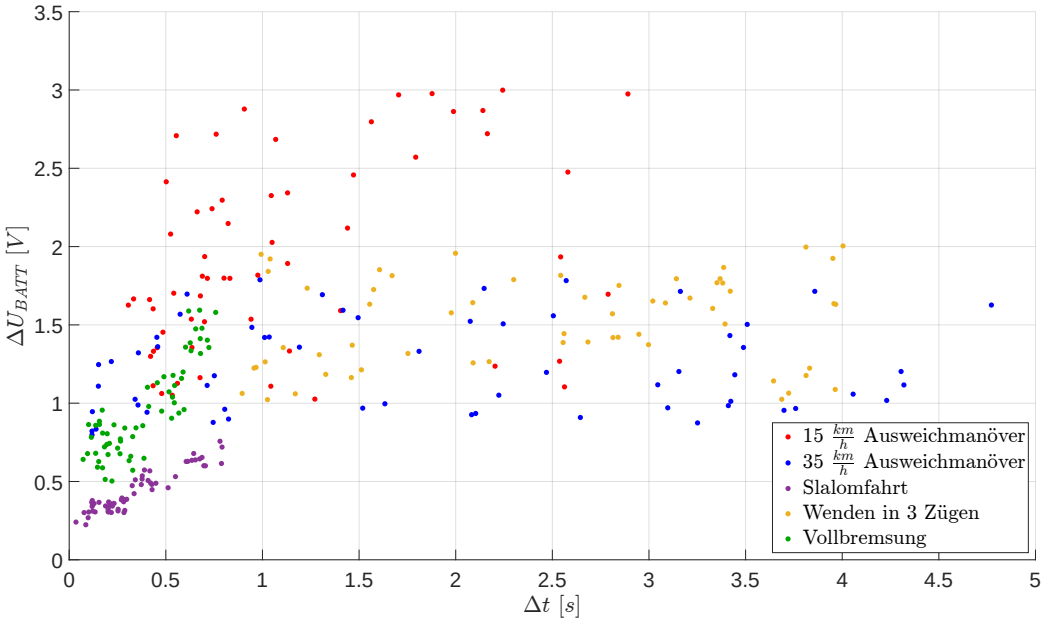


Abbildung 4.3: Streudiagramm aller zur Verfügung stehenden Messreihen nach Dauer und Tiefe des Spannungseinbruchs, eingefärbt nach dem Typ des Manövers.

bruchs während des Manövers. Durch den niedrigsten Spannungswert wird die maximale Tiefe des Einbruchs bestimmt. Das untere Diagramm stellt die berechnete Bordnetzleistung P_{BN} dar. Anhand des Strombedarfs der Hochleistungsverbraucher wird der Beginn des Manövers festgelegt. Ein Durchschnitt aus den 500 vorangegangen Messwerten der Bordnetzleistung berechnet die Grundleistung. Diese wird durch die grüne Linie verdeutlicht. Nach Auswertung der Daten aller Szenarien wird die Abhängigkeit zwischen der Tiefe und Dauer des Spannungseinbruchs vom Typ des Manövers analysiert. Ein Streudiagramm, mit dem die Tiefe ΔU_{BATT} und Dauer Δt des Spannungseinbruchs für alle Szenarien dargestellt wird, ist in Abbildung 4.3 zu sehen.

Aus dem Streudiagramm wird ersichtlich, dass die Bordnetzstabilität stärker beeinträchtigt ist, je weiter rechts und je weiter oben ein Punkt im Diagramm angeordnet ist. In diese Richtung nehmen die Dauer Δt und Tiefe ΔU_{BATT} des Spannungseinbruchs zu. Die Manöver Slalomfahrt und Vollbremsung weisen einen kurzzeitigen Spannungseinbruch delta $\Delta t < 0,8\ s$ auf. Zudem ist die Tiefe der Einbrüche $\Delta U_{BATT} < 1,6\ V$ im Vergleich zu den anderen Manövern sehr gering. Der geringste Spannungseinbruch wird bei der Slalomfahrt verursacht. Beim Ausweichmanöver mit einer Geschwindigkeit von 35 $\frac{km}{h}$ sind Instabilitäten mit sehr unterschiedlichen zeitlichen Abständen festzuhalten. Die Tiefe des Spannungseinbruchs reichen von $0,75\ V$ bis $1,8\ V$. Das Szenario Wenden in 3 Zügen verhält sich ähnlich. Die Dauer der Einbrüche ist auch hier sehr unterschiedlich. Die Tiefe der Einbrüche befindet sich zwischen 1 V und 2 V. Die größte Belastung des Bordnetzes werden durch die Ausweichmanöver mit 15 $\frac{km}{h}$ verursacht. Aufgrund der Überlagerung der Ströme und der geringen Fahrzeuggeschwindigkeit treten hier Spannungseinbrüche bis zu 3 V auf. Die Dauer ist dabei mehrheitlich $\Delta t < 3\ s$. Aus der vorangegangenen Analyse

stellt sich heraus, dass für das Training und anschließender Validierung des RL-Agenten vor allem die Ausweichmanöver und das Wenden in 3 Zügen relevant sind.

4.3 Abgleich der Simulation mit realen Messungen

Die in Kapitel 3 beschriebenen Teilkomponenten wurden zu einem Gesamtsystem zusammengefasst. Das Modell stellt ein 12 V-Bordnetz mit einem Generator, einer Batterie sowie mehreren elektrischen Verbrauchern dar. Neben diesen Teilmodellen werden Leitungswiderstände zwischen Generator und Batterie sowie zwischen Batterie und Verbraucher eingefügt.
Das gesamte Modell wird früh in der Entwicklung validiert, um sicherzustellen, dass die Gesamtfunktionalitäten der Simulation das elektrische System des Fahrzeugs so genau wie möglich abbilden. Ziel der Verifikation und Validierung (V&V) ist laut Rabe [113], Fehler zu eliminieren und die Zuverlässigkeit eines Modells zu erhöhen. Die Kombination verschiedener V&V-Techniken steigert deren jeweilige Effizienz. Ein mögliches Verfahren ist der Vergleich mit realen Daten. Daher werden im Folgenden die Simulationsergebnisse mit den in Kapitel 4.2 dargestellten Fahrzeugmessungen verglichen. Hierfür wird aufgrund der Tiefe und Dauer des Spannungseinbruchs ein Ausweichmanöver als Vergleich verwendet. Zunächst wird das Modell auf die Parameter aus Tabelle 4.1 abgestimmt. Da bei diesem Manöver die Fahrzeuggeschwindigkeit und damit die Motordrehzahl niedrig ist, kann der Generator unter diesen Umständen nicht seine Nennleistung erbringen. Darüber hinaus wird das Bordnetz durch die manöverbedingte Aktivierung der Hochleistungsverbraucher stark belastet. Durch die Überlagerung aller Verbraucher und dem ungünstigen Betriebspunkt des Generators muss die Batterie hohe Ströme liefern. Dies verursacht einen starken Spannungseinbruch. In Abbildung 4.4 sind die Leistungskurven der Komfortverbraucher P_{KV}, Hochleistungsverbraucher P_{HLV} und die sich daraus ergebende Bordnetzleistung P_{BN} dargestellt. Um die Bordnetz-Grundlast hochzuhalten, werden zu Beginn der Fahrten viele elektrische Verbraucher eingeschaltet. Hierzu zählen unter andrem alle zur Verfügung stehenden Komfortverbraucher, wie z.B. die Sitzheizungen, Lenkradheizung und Thermo-Cupholder. Somit treten im 12 V-Bordnetz Spitzenleistungen von über 3 kW auf. Dies entspricht einem Worst-Case-Szenario und eignet sich optimal für die Untersuchung der Bordnetzstabilität. Die Grundleistung aus der betrachteten Messung beträgt ca. $2,5$ kW. Ab $t = 5$ s wird das Manöver ausgeführt und somit die Hochleistungsverbraucher aktiviert. Aus der Grafik ist zu sehen, dass während dem Manöver die Bordnetzleistung für 10 s abnimmt. Diese Degradierung oder gar Abschaltung der elektrischen Verbraucher stellt die Strategie des betrachteten Fahrzeugs zur Stabilisierung der Bordnetzspannung dar. Wie die Strategie im Detail aussieht, lässt sich aus den Messdaten nicht ermitteln. Dennoch kann ein starker Abfall der Batteriespannung gemessen werden, der mit den Simulationsergebnissen in Abbildung 4.5 verglichen wird. Dabei wurden die Leistungsprofile und die Motordrehzahl aus der Messung in das Modell eingespeist. Zu Beginn der Messung beträgt die Spannung $14,8$ V. Zum Zeitpunkt $t = 5$ s wird das Manöver ausgeführt und

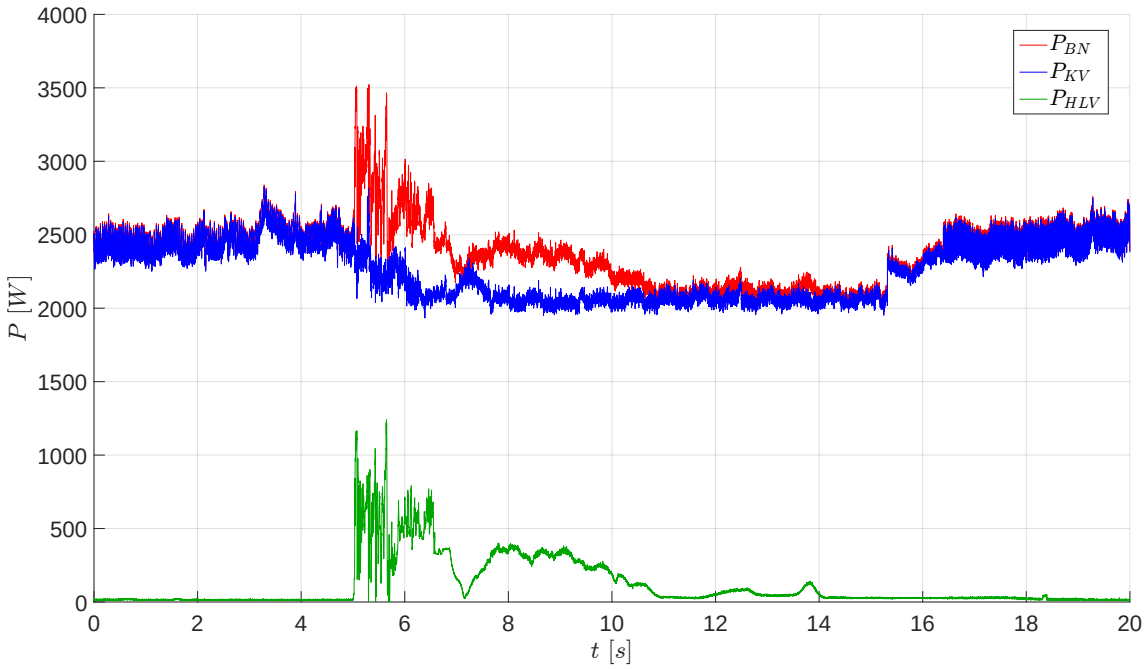


Abbildung 4.4: Messung der abgefragten Leistung während des Ausweichmanövers. Die Summe aus den Leistungen der Hochleistungsverbraucher P_{HLV} (grün) und der Komfortverbraucher P_{KV} (blau) ergibt die gesamte Bordnetzleistung P_{BN} (rot).

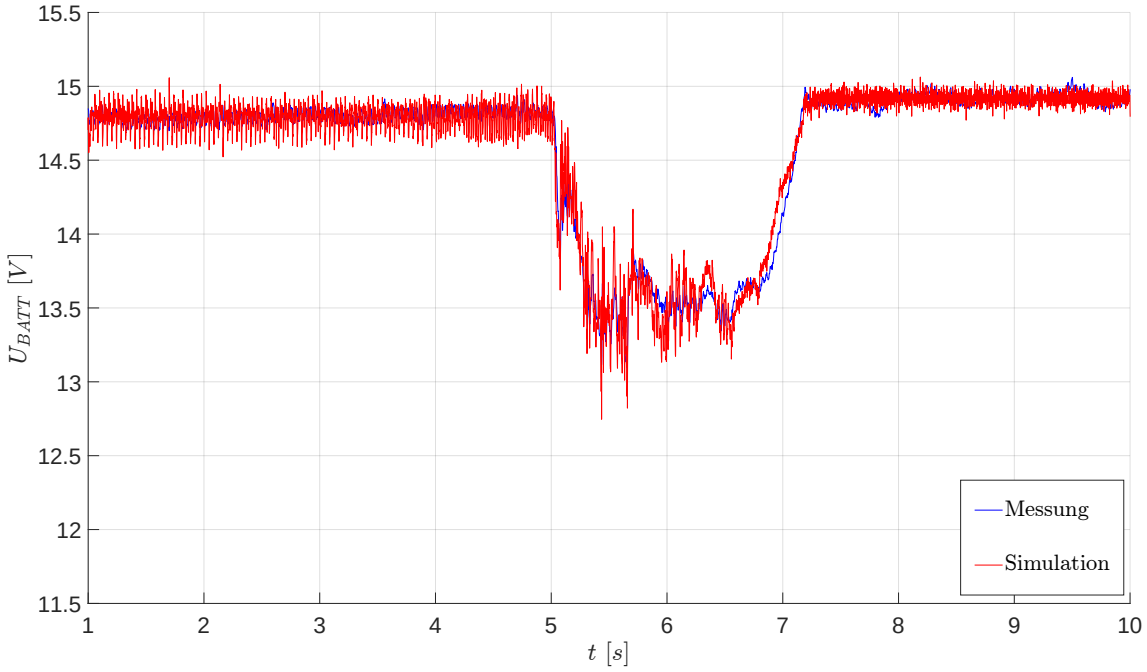
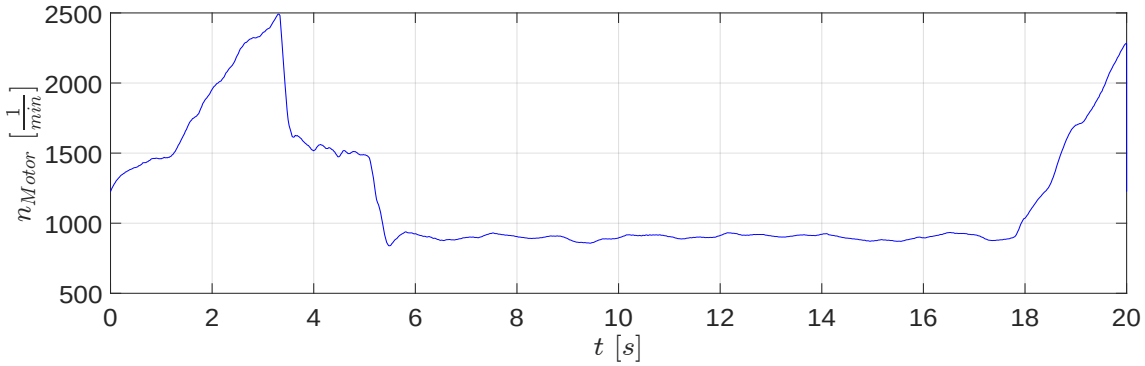
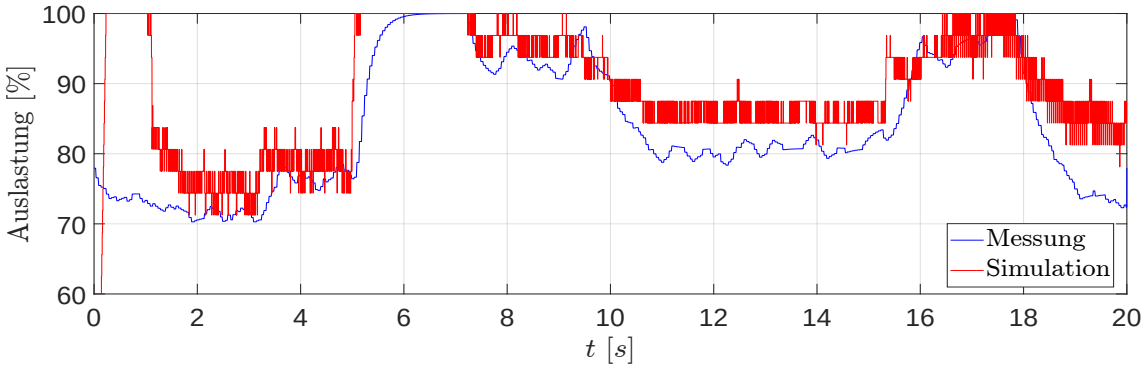


Abbildung 4.5: Vergleich der Batteriespannung aus der Messung und Simulation während eines Ausweichmanövers.



(a) Motordrehzahl n_{Motor}



(b) Generatorauslastung

Abbildung 4.6: Vergleich der Generatorauslastung von Messung und Modell während eines Ausweichmanövers. Die dabei gefahrene Motordrehzahl ist in der oberen Abbildung dargestellt.

die Spannung bricht ein. Der Spannungseinbruch erfolgt sowohl in der Messung als auch in der Simulation. Hierbei ist zu erkennen, dass die Kurven der Batteriespannung für die Messung und der Simulation sehr gut übereinstimmen. Sowohl die Dauer des Einbruchs als auch der Spannungsanstieg am Ende werden gut abgebildet. Somit ist das Modell hinreichend genau, um die Bordnetzstabilität in dieser Arbeit zu untersuchen.

In Abbildung 4.6 ist die gesamte Motordrehzahl und ein Vergleich der Generatorauslastung zwischen der Messung und Simulation dargestellt. Der Abbildung ist zu entnehmen, dass die Auslastung des Generators in der Simulation unmittelbar auf 100 % steigt. Der Grund hierfür ist, dass der Anfangswert der Bordnetzspannung in der Simulation von der Batterie bestimmt wird. Dieser Wert wird vom Generator wieder auf den Sollwert von $14,8\ V$ geregelt. Anschließend sinkt die Auslastung unter 80 %. Zum Zeitpunkt des Manövers $t = 5\ s$ steigt die Auslastung wieder auf 100 %. Es ist zu erkennen, dass die Reaktion des Generators während des gesamten Simulationszeitraums mit einer hohen Genauigkeit der Messung entspricht.

In den Simulationen bilden alle steuerbaren Verbraucher sowie eine konstante Grundlast die Grundleistung im Bordnetz. Durch die Degradierung oder Abschaltung der Leistung von den Komfortverbrauchern besteht die Möglichkeit einer möglichen Strategie zum Umgang mit Spannungsschwankungen im 12 V-Bordnetz. Die steuerbaren Komfortver-

braucher stellen, wie in Kapitel 2.2 erläutert, eine wichtige Möglichkeit zur Sicherung der Bordnetzstabilität dar. Nach Tabelle 2.2 ist die Generatorregelung eine weitere Eingriffsmöglichkeit. Jedoch wird diese Regelung in dieser Arbeit nicht betrachtet, da sie gleichzeitig die Effizienz und Ladebilanz stark beeinflusst.

5 Deep Reinforcement Learning Agent

In den vorangegangenen Kapiteln wurden die Komponenten im 12 V-Bordnetz, deren Modellierung und die relevanten Fahrzeugmessungen vorgestellt. Somit wurden das Simulationsmodell und dessen Eingangsparameter näher erläutert. In diesem Kapitel wird die Implementierung eines RL-Agenten näher beschrieben. Zunächst wird die Auswahl des Algorithmus begründet. Anschließend wird in mehreren Schritten die Umsetzung erläutert. Zum Schluss erfolgt mit Hilfe der Anpassung der Hyperparameter eine Optimierung des Agenten in der Simulation.

Das RL-System besteht aus einem Agenten und einer Umwelt. Das vorgestellte 12 V-Bordnetzmodell stellt im Bezug auf RL die Umwelt des Agenten dar. Hier kann der Agent mit seinen Aktionen Änderungen vornehmen und wird anschließend mit einer definierten Belohnungsfunktion belohnt oder bestraft. Die Höhe der Belohnung gibt an, wie gut die durchgeführte Aktion bei der Erfüllung der Aufgabe ist. Der neue Beobachtungsparameter wird dann an den Agenten weitergegeben und der Agent entscheidet über die nächstbestmögliche Aktion. Unter Verwendung von Informationen aus Aktionen, Beobachtungen und Belohnungen kann der Lernalgorithmus die Strategie kontinuierlich verbessern, um die erwartete kumulative Belohnung zu maximieren.

In dieser Arbeit wird ein RL-Agent in MATLAB Simulink implementiert. Zunächst werden in MATLAB alle notwendigen Parameter des Agenten definiert. Dabei variiert die Struktur dieser Parameter je nach Typ des Agenten. Für die Interaktion mit der Umwelt besitzt der RL-Agent die Eingänge `observation`, `reward` und `isDone` sowie den Ausgang `action`. Zusätzlich kann die kumulative Belohnung über den Ausgang `cumulativeReward` ausgegeben werden. Mit Hilfe der Eingänge `observation` und `reward` wird der Zustand der Umwelt und die dort generierte Belohnung repräsentiert. Der Ausgang `action` gibt dahingegen die Handlungen des Agenten vor. Dadurch ist das Prinzip von RL in Abbildung 2.4 von Kapitel 2.3.1 erläutert. Während des Trainings kann durch den Eingang `isDone` ein vorzeitiger Abbruch der aktuellen Episode definiert werden, falls z.B. durch Erfüllen der Aufgabe vor Simulationsende ein Endzustand erreicht wird. Dies ist in dieser Arbeit nicht vorgesehen, da die Simulation mit zeitlich begrenzten Messdaten in Episoden aufgeteilt wird.

Diese Arbeit betrachtet ein MDP, der rein theoretisch eine kontinuierliche – nicht episodische – Aufgabe darstellt. Ziel ist es, die Stabilität des Bordnetzes zu verbessern und Spannungseinbrüche in kritischen Situationen zu minimieren oder sogar zu vermeiden.

Zum Training der RL-Agenten stehen zwei verschiedene Rechner mit den in Tabelle 5.1 beschriebenen Eigenschaften zur Verfügung.

Tabelle 5.1: Auflistung von den Eigenschaften der genutzten Hardware.

Name	**GPU(s)**	**CUDA Version**	**CPU**	**RAM**
KI-Rechner 1	NVIDIA RTX 2080 Ti	V11.7	Intel® Xeon® W-2133	32 GB
KI-Rechner 2	2x NVIDIA RTX 2080 Ti	V11.2	Intel® Xeon® W-2133	32 GB

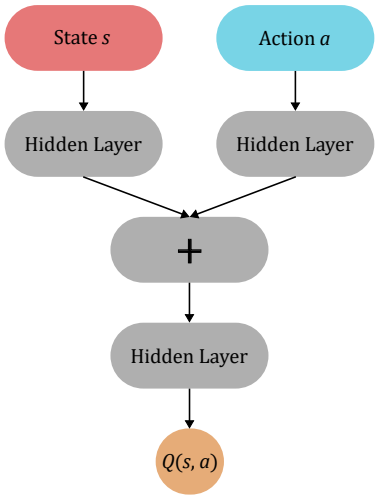
Die Verteilung der Trainings auf zwei verschiedene Rechner bringt einen zeitlichen Vorteil. Interessant wird zu beobachten sein, ob KI-Rechner 2 durch eine zweite GPU einen Leistungsvorteil gegenüber KI-Rechner 1 hat. Da nur die Berechnung der neuronalen Netze auf der GPU stattfindet, die Berechnungsdauer der Simulation jedoch von der CPU abhängig ist, wird der Vorteil in RL schätzungsweise geringer sein als beim klassischen Deep Learning.

5.1 Auswahl des Algorithmus

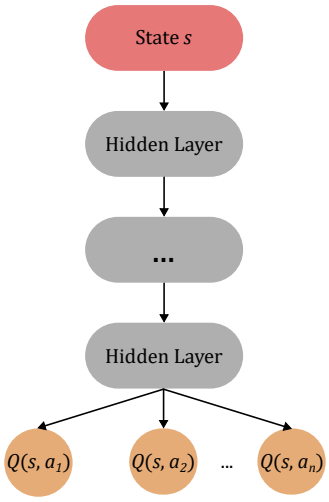
MATLAB ermöglicht die Verwendung verschiedener Algorithmen, deren Hyperparameter über diverse Strukturen eingestellt werden können. Eine Übersicht über die Agenten ist in Tabelle 5.2 dargestellt. Darin ist sowohl der Typ als auch die Eigenschaft des Aktionsraumes beschrieben. Bei der Auswahl des Agenten ist die Übereinstimmung des Aktionsraumes und der Beobachtungen zwischen Modell und Agent ein wichtiges Kriterium. Dadurch kann eine gute policy bei der Erfüllung der Aufgabe erzielt werden. In einem Modell mit einem kontinuierlichen Aktionsraum wird ein Agent mit einem diskreten Aktionsraum schlechter arbeiten als ein Agent mit kontinuierlichem Aktionsraum.
Für das Deep Q-Network stehen zwei Möglichkeiten der Netzarchitektur zur Verfügung. Diese sind in Abbildung 5.1 dargestellt. Dabei kann die Anzahl der Hidden Layer je nach Anwendung für beide Architekturen variiert werden. In Abbildung 5.1(a) wird ein neuronales Netz mit zwei Eingangsparametern, Zustand s und Aktion a, vorgestellt. Der Ausgang besteht aus einem einzelnen Neuron, welches den Q-Wert angibt. Dahingegen zeigt Abbildung 5.1(b) ein neuronales Netz mit nur einem Eingans-Neuron, der Zustand s. Jedoch besteht hier die Ausgangsschicht aus mehreren Neuronen. Dabei gibt die Anzahl der diskreten Aktionen im Aktionsraum die Größe der Ausgangsschicht vor. Jedes Neuron steht für den Q-Wert einer der diskreten Aktionen. Mit der Verwendung dieser Netzarchitektur können rekurrente Schichten, wie z.B. LSTM-Layer eingebaut werden. Die Verwendung dieser Netzarchitektur wird von MathWorks empfohlen [115].
Im Rahmen dieser Arbeit soll ein Algorithmus zur Steuerung von unterschiedlichen elektrischen Verbrauchern, die stufenweise eingestellt werden können, untersucht werden. Die Generatorregelung könnte ebenfalls als Freiheitsgrad genutzt werden. Jedoch wird eine kontinuierliche Regelung des Soll-Spannungswerts durch den Agenten nicht als sinnvoll erachtet, da damit gegen den im Generatormodell bereits integrierten Regler gearbeitet wurde.

Tabelle 5.2: Auflistung der in MATLAB implementierten Agenten nach Typ und Aktionsraum. Die Unterscheidung des Typs erfolgt nach Wert-basiert, Policy-basiert oder Actor-Critic. Der Aktionsraum wird in diskret oder kontinuierlich aufgeteilt [114].

Agent	**Typ**	**Aktionsraum**
Q-Learning	Wert-basiert	diskret
Deep Q-Network	Wert-basiert	diskret
State-Action-Reward-State-Action (SARSA)	Wert-basiert	diskret oder kontinuierlich
Policy Gradients (PG) bzw. REINFORCE	Policy-basiert	diskret oder kontinuierlich
Advantage Actor-Critic (A2C) bzw. Asynchronous Advantage Actor-Critic (A3C)	Actor-Critic	diskret oder kontinuierlich
Proximal Policy Optimization (PPO)	Actor-Critic	diskret oder kontinuierlich
Trusted Region Policy Optimization (TRPO)	Actor-Critic	diskret oder kontinuierlich
Deep Deterministic Policy Gradient (DDPG)	Actor-Critic	kontinuierlich
Twin-Delayed Deep Deterministic Policy Gradient (TD3)	Actor-Critic	kontinuierlich
Soft Actor-Critic (SAC)	Actor-Critic	kontinuierlich



(a) Neuronales Netz mit einzelnem Ausgangsneuron



(b) Neuronales Netz mit je einem Ausgangsneuron für jede diskrete Aktion

Abbildung 5.1: Mögliche Definitionen eines Deep Q-Networks für einen diskreten Aktionsraum in MATLAB. In der linken Grafik besteht die Eingangsschicht des neuronalen Netzes aus dem Zustand s und der Aktion a. Als Ausgang wird ein einzelnes Neuron als Q-Wert definiert. In der rechten Abbildung besteht die Eingangsschicht nur aus dem Zustand s. Die Ausgangsschicht besteht aus mehreren Neuronen. Adaptiert aus [116].

Der RL-Agent muss entsprechend für diskrete Aktionsräume geeignet sein. Wie in Tabelle 5.2 zu sehen, können aber nicht alle Algorithmen mit diskreten und kontinuierlichen Aktionsräumen umgehen. Deshalb sind vor der Wahl des Agenten bestimmte Vorüberlegungen zu den Zustands- und Aktionsräumen notwendig. Da der Aktionsraum aus nur diskreten Werten besteht, werden die Algorithmen DDPG, TD3 und SAC nicht für die Implementierung berücksichtigt. Eine Diskretisierung der kontinuierlichen Aktionen wird in dieser Arbeit nicht berücksichtigt. Aufgrund der hohen Dimension des Aktionsraumes werden das tabellarische Q-Learning oder SARSA bei der Anwendung nicht verwendet. Das in Kapitel 2.3.7 angesprochene Curse of Dimensionality spricht gegen deren Verwendung. Diese hohe Dimensionalität entsteht dadurch, dass in der Simulation ausschließlich kontinuierliche Signale beobachtet werden. Neben Zustands- und Aktionsraum spielt auch die algorithmische Komplexität eine wichtige Rolle. Obwohl einige wichtige Agentenmethoden wie A2C/A3C, PPO und TRPO mit diskreten Aktionsräumen umgehen können, sind sie für die gewünschte Anwendung nicht geeignet. Diese Algorithmen verwenden mehrere Agenten, die gelernte Gewichtungen untereinander austauschen [117]. Die Folge ist ein hoher Rechenaufwand beim Training im Gegensatz zu z.B. DQNs, die nur auf einem Critic-Netz basieren. Desweiteren steigt die Komplexität der Implementierung und die Optimierung bei zwei neuronalen Netzen.
In einem Experiment wurden die Agenten AC, DQN und PG näher untersucht [118]. Dabei wurden die Agenten mit aktivierten Komfortverbrauchern und zusätzlichen Lasten trainiert. Anschließend wurde der Spannungsabfall und die Reaktion der Agenten in den kritischen Situationen näher analysiert. Die starken Spannungseinbrüche sollten den jeweiligen Agenten dazu veranlassen, elektrische Verbraucher herunterzustufen. Um die Agenten nach dem Training zu testen, wurden zu verschiedenen Zeitpunkten Hochleistungsverbraucher aktiviert. Abbildung 5.2 zeigt den Spannungsverlauf der Batterie für die Agenten, die auf die kurzfristige Aktivierung der Hochleistungsverbraucher angewandt wurden.

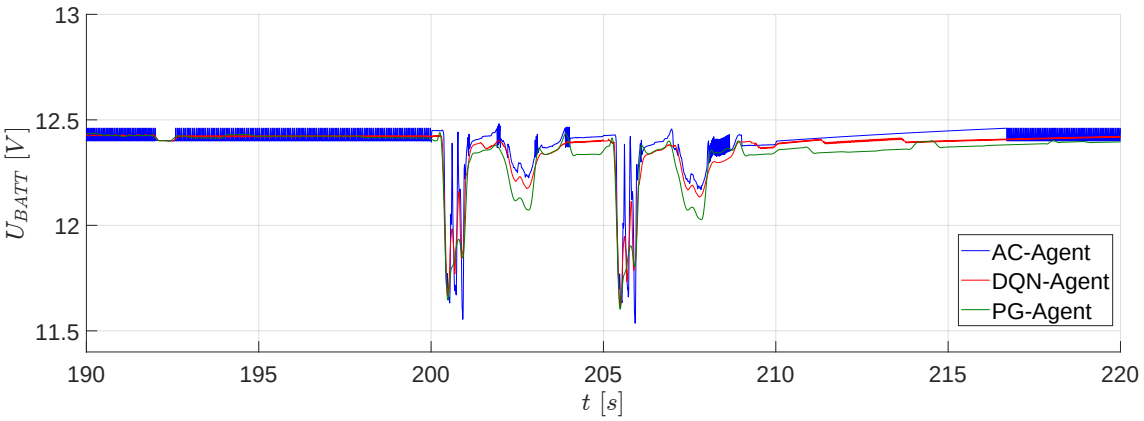


Abbildung 5.2: Spannungsverlauf der Batterie für den Vergleich der Agenten AC, DQN und PG.

Obwohl der starke Abfall der Batteriespannung reduziert werden konnte, sind die vielen

Schwankungen der Spannungswerte innerhalb kurzer Zeit auf häufige Zustandsschwankungen bei den Komfortverbrauchern zurückzuführen. Dieser Effekt war bei den Agenten AC und PG zu beobachten. Der DQN-Agent hat sich in diesem Experiment als am effektivsten erwiesen.

Als Schlussfolgerung fällt die Wahl des Algorithmus auf Deep Q-Learning. Hier gibt es noch zusätzlich die Option des Double DQNs [119]. Dadurch können den bekannten Nachteilen von DQNs, wie z.B. Überschätzung oder Instabilitäten im Lernprozess entgegengewirkt werden. Die Auswahl lässt sich mit verwandten Arbeiten aus dem Bereich Energiemanagement unterstützen [16, 17, 18, 19, 120, 121]. Um die Verwendung eines RNN nicht von vornherein auszuschließen, wird die Variante der Abbildung 5.1(b) als Q-Vektor zur Definition von DQN verwendet.

5.2 Wahl des Zustandsraums

Je nach Lernziel wird der Zustandsraum definiert. Die Zustände, die der Agent von der Umwelt zurückbekommt, sollten jede Situation der Umwelt so gut wie möglich darstellen. Zu viele Signale führen zu einer komplexen Wertfunktion, die schwer zu erlernen ist. Andererseits führen zu wenige Signale zum erfolglosem Lernen aufgrund fehlender Informationen. Es müssen Signale definiert werden, die als Indikator für kritische Situationen der Bordnetzstabilität eines Fahrzeugs dienen können. Diese Spannungseinbrüche entstehen dann, wenn das Zusammenspiel zwischen Erzeuger und Verbraucher im System aus dem Gleichgewicht gerät. Da der Generatorstrom I_{GEN}, Batteriestrom I_{BATT} und Bordnetzstrom I_{BN} in den Messungen aufgenommen wurden, können sie als Zustände für die Anwendung dienen. Der Agent kennt jedoch nicht den maximalen Generatorstrom. Dadurch ist die Leistungsgrenze nicht direkt offensichtlich. Hier eine Konstante als Zustand einzuführen ist nicht Erfolg bringend. Stattdessen wird der Spannungsabfall als Anhaltspunkt verwendet. Damit der Agent unabhängig vom Spannungsniveau arbeitet, wird anstelle des absoluten Spannungsniveaus der Batterie U_{BATT} der relative Einbruch ΔU_{BATT} betrachtet, der durch einen Zielwert berechnet wird. Der relative Einbruch ΔU_{BATT} vervollständigt mit den Strömen I_{GEN}, I_{BATT} und I_{BN} den Zustandsraum im ersten Schritt.

Nach den ersten Tests wurden kleine Änderungen vorgenommen. Verrauschte Signale werden nach Möglichkeit durch Alternativen ersetzt. Der Generatorstrom I_{GEN} wird durch die Generatorauslastung ersetzt. Diese stehen im direkten Zusammenhang zueinander. Aufgrund der Berechnung im Generatorblock ist die Generatorauslastung mit weniger Schwingungen belastet. Zusätzlich zum Generatorstrom wird der gesamte Verbraucherstrom durch den Strom der Hochleistungsverbraucher I_{HLV} ersetzt. Der Grund dafür ist, dass die Daten oder Signale mit maximaler Varianz für das Lernen wichtig sind. Für das Lernziel dieser Arbeit kann auf den Offset, der sich durch die restlichen Verbraucher ergibt,

verzichtet werden. Somit ergibt sich folgender Zustandsraum:

$$\mathcal{S} = \{I_{BATT}, I_{HLV}, Generatorauslastung, \Delta U_{BATT}\} \tag{5.1}$$

Um eine vergleichbare Gewichtung unterschiedlicher Signale mit gleichem Wertebereich zu gewährleisten, werden diese im Intervall $[0, 1]$ skaliert. Dadurch werden die Signale direkt in den Arbeitsbereich der neuronalen Aktivierungsfunktion gelegt. Bei ReLUs als Aktivierungsfunktion ist der lineare Arbeitsbereich bei positiven Werten immer gegeben. Dahingegen werden negative Werte des gewichteten Inputs komplett eliminiert. Da die Generatorauslastung in Prozent vorliegt und dem ausgewählten Intervall entspricht, ist keine weitere Skalierung erforderlich. Der Batteriestrom I_{BATT} und der Strom der Hochleistungsverbraucher I_{HLV} sind auf den maximalen Generatorstrom aus Tabelle 4.1 skaliert. Der Spannungsabfall ΔU_{BATT} ist auf $2,5\ V$ skaliert, da die meisten Messungen aus der Abbildung 4.3 diesen Wert nicht überschreiten.

5.3 Dimensionalität des Aktionsraums

Das Bordnetzmodell des Fahrzeugs lässt im Wesentlichen zwei Eingriffsmöglichkeiten für die Stabilisierung des Bordnetzes zu. Einerseits ist es möglich, die Generatorspannung zu regeln und somit indirekt die vom Generator bereitgestellte Leistung zu liefern. Andererseits bietet die Simulation die Möglichkeit, die Status der elektrischen Verbraucher zu ändern und somit den Leistungsbedarf im 12 V-Bordnetz zu verringern. Die Sollwerte für die elektrischen Verbraucher werden über Zeitreihen vorgegeben und repräsentieren die Wünsche der Fahrzeuginsassen. Jede durch den Agenten verursachte Abweichung vom Sollwert bedeutet Komfortverlust. Dennoch ist das Degradieren bzw. Abschalten der elektrischen Verbraucher gegenüber der Regelung der Generatorspannung das deutlich dynamischere Mittel.

Der Aktionsraum beschreibt, welche Aktionen der Agent ausführen kann, um mit der Umwelt zu interagieren. Um diesen zu bestimmen, wurden mehrere Experimente durchgeführt. Ursprünglich war geplant, dass der Agent alle simulierten Verbraucher steuern kann. Dies musste aber aufgrund des hohen Ressourcenverbrauchs abgebrochen werden. Da Verbraucher auf unterschiedliche Stufen gesteuert werden können, ergibt sich ein diskreter Aktionsraum. Dieser wird in MATLAB definiert und erfordert alle möglichen Kombinationen der einstellbaren Verbraucherstufen. Die Simulation bietet sechs vierstufige, einen dreistufigen, zwei zweistufige und einen siebenstufigen Verbraucher. Unter Berücksichtigung aller möglichen Verbraucherstufen ergibt sich ein Aktionsraum von $4^6 \cdot 3^1 \cdot 2^2 \cdot 7^1 = 344.064$ Aktionen. Versuche, mit einem solchen Aktionsraum ein Training zu starten, sind immer wieder gescheitert. Die Reduzierung der Anzahl an steuerbaren Verbrauchern wirkt dem zwar entgegen, ist aber aus Sicht der Problemdarstellung dieser Anwendung nicht optimal. Eine Reduktion der steuerbaren elektrischen Verbraucher bedeutet gleichermaßen eine Reduktion der Eingriffsmöglichkeiten in die abgefragte Bordnetzleistung. Um die Anzahl der

Aktionen des Agenten zu reduzieren, wird der Aktionsraum in Abschaltstufen unterteilt. Dabei wird auf eine gleichmäßige Reduzierung der Leistung zwischen den Stufen geachtet. Somit wird der Trainingsprozess des Agenten vereinfacht. Die Anzahl der Abschaltstufen ergibt sich aus der maximalen Leistungsdifferenz zwischen zwei Stufen unter allen elektrischen Verbrauchern. In der aktuellen Anwendung sind für die Degradierung elf Stufen vorhanden. Einschließlich der Stufe 0, in der kein Eingriff erfolgt, gibt es zwölf Degradationsstufen.

Hierbei wird sicher gestellt, dass Verbraucher zuerst heruntergestuft oder abgeschaltet werden, wenn Eingriffe für die Fahrzeuginsassen am wenigsten wahrnehmbar sind. Aufgrund der hohen Zeitkonstante gilt dies grundsätzlich für alle Heizelemente (z.B. Sitzheizungen). Gleichzeitig wird dem Agenten nicht erlaubt, zwischen Insassen zu priorisieren. So werden beispielsweise die Sitzheizungen für Fahrer und Beifahrer in einem Schritt ausgeschaltet. Damit die genannten Überlegungen vom Agenten gelernt werden können, müssen sie Teil der Belohnungsfunktion werden. Zusammenfassend lässt sich durch die Verwendung von Abschaltstufen sowohl der Aktionsraum des Agenten reduzieren als auch systembedingtes Wissen und Verhaltensregeln integrieren.

5.4 Wahl des Aktionsraums zur Vermeidung hochfrequenter Aktionswechsel

Im vorherigen Kapitel wurde die Entstehung der zwölf Degradierungslevel näher erläutert. Die Größe und die Anzahl der zwölf diskreten Aktionen stellt im Bezug auf Rechenaufwand kein Problem dar. Allerdings wird aus mehreren Gründen gegen diese Wahl entschieden. Zunächst einmal ist somit möglich, dass von einem Zeitschritt zum nächsten über mehrere Stufen gesprungen werden kann. Dies würde ein sehr unruhiges Verhalten verursachen. Gerade zu Beginn des Trainings, wenn ε hoch und die Entdeckung sehr ausgeprägt ist, können zufällige Aktionen zu Leistungssprüngen führen. Zudem wurden in verschiedenen Tests – auch mit weniger Exploration – hochfrequente Aktionsänderungen festgestellt. Diese entstehen über die Zustände, mit denen die von der Wertfunktion abgeleiteten Aktionen in direktem Zusammenhang stehen. Alle potenziellen Signale sind jedoch mit hochfrequenten Schwingungen belastet (siehe Abbildung 4.2 (b)). Es besteht eine Rückkopplung dieser Schwingungen über die Wertfunktion bis hin zu den Aktionen.

Durch die Wahl von nur drei Aktionen mit $\mathcal{A} = \{-1, 0, +1\}$ können die beschriebenen Probleme gelöst werden. Dadurch sind die Abschaltstufen nicht mehr absolut wählbar. Stattdessen kann die Stufe der elektrischen Verbraucher mit $a_1 = -1$ verringert, mit $a_2 = 0$ gehalten oder mit $a_3 = +1$ erhöht werden. Dadurch ist ein Sprung über mehrere Stufen nicht mehr möglich und die Wahrscheinlichkeit, die Stufe zu halten, steigt. Somit reduziert sich auch die Rückkopplung von hochfrequenten Schwingungen in den Zustandssignalen auf die Aktionen. Jedoch werden sie nicht vollständig eliminiert. Dazu wird eine optionale Gegenmaßnahme implementiert, die die vom Agenten neu eingestellte Stufe der

Verbraucher für eine bestimmte Zeit t_{hold} hält. Diese Zeit wird mit $t_{hold} = 10\ ms$ definiert und umfasst somit zehn diskrete Zeitschritte in der Simulation. Dieser Mechanismus wird ins Simulink-Modell implementiert und wird erst wirksam, wenn der Agent vollständig trainiert ist. Die Policy des Agenten bleibt daher dieselbe. Anders als bei der Verwendung von Zustandsfiltern wird die Reaktionsgeschwindigkeit des Agenten bei dem erwähnten Mechanismus nicht begrenzt. Solange eine Stufe gehalten wird, besteht die Möglichkeit, in jedem diskreten Zeitschritt einzugreifen, während die Zustände ohne Verzögerung beobachtet werden können. Bei einer Filterung der Zustände, tritt eine Verzögerung unabhängig von der Aktion auf, sodass der Agent den Spannungsabfall nur mit Verzögerung erkennt. In Kapitel 5.6 werden neben der Optimierung von Hyperparametern auch die Auswirkungen des beschriebenen Mechanismus dargestellt.

5.5 Formulierung der Belohnungsfunktion

Die Belohnung ist ein Wert, der die Güte einer gewählten Aktion in einem gegebenen Zustand definiert. Mithilfe dieser Belohnungsfunktion versucht der Agent seine Policy zu verbessern, um zukünftig gewinnbringendere Aktionen zu wählen. Die Funktion ist im RL primär für den Austausch zwischen dem Agenten und der Umwelt zuständig und beschreibt das Lernziel. In der vorliegenden Arbeit wird das Lernziel als Kompromiss zwischen Einhalten der Verbraucheranforderungen durch die Insassen und der Vermeidung von Spannungseinbrüchen im Sinne der Bordnetzstabilität definiert. Belohnungsfunktionen können als Belohnung (positiv) oder Bestrafung (negativ) formuliert werden, wobei der Agent immer versucht, diese zu maximieren. Die Art der Formulierung hängt von der Aufgabenstellung und den Beobachtungen ab. Kompromisse zwischen den verschiedenen Teilzielen können mathematisch als gewichtete Summe ausgedrückt werden. Dabei wird die Priorisierung der Teilziele durch Gewichtungsfaktoren beeinflusst. Viele Arbeiten im Bereich Energiemanagement in HEVs basieren auf eine gewichtete Summe zwischen Kraftstoffverbrauch und Batterieladeziel. Beispielsweise muss in [12, 16, 122, 123, 124] der Ziel-SOC während des Fahrzyklus erreicht werden. Dazu wird der quadratische Fehler des SOC-Zielwerts als Bestrafung in die Belohnungsfunktion aufgenommen. Die quadratische Funktion stellt sicher, dass das Überschreiten und Unterschreiten des Ziel-Ladezustandes berücksichtigt werden. Außerdem erhöht sich die Strafe, wenn der aktuelle SOC weiter vom Ziel entfernt ist. Darauf basierend wird das Teilziel der Vermeidung von Spannungseinbrüchen in dieser Arbeit als quadratischer Fehler zwischen der aktuellen Batteriespannung und der angestrebten Batteriespannung definiert:

$$R_1 = -(U_{BATT,\ Soll} - U_{BATT,\ Ist})^2 = -\Delta {U_{BATT}}^2 \tag{5.2}$$

Als zweites Teilziel wird das Einhalten der Verbraucheranforderungen von den Insassen definiert. Um dies zu erreichen, wird der Agent für seine Eingriffe in den Komfort der Fahrzeuginsassen bestraft. Diese Bestrafung nimmt mit der Höhe der Abweichung δ zwi-

schen dem Soll-Status und Ist-Status der Verbraucher zu. Dieses Teilziel lässt sich wie folgt beschreiben:

$$R_2 = -\delta, \text{ mit } \delta \in \mathbb{Z} \wedge \delta \in [0, 11] \tag{5.3}$$

Der Agent darf sich mit seinen Aktionen nur im Rahmen der elf definierten Degradierungslevel aus Kapitel 5.3 bewegen. Daher wird er wie folgt bestraft, wenn er die Grenzen der möglichen Degradierungslevel überschreiten will:

$$R_3 = \begin{cases} -1 & , \quad \delta = 11 \wedge a = +1 \\ -1 & , \quad \delta = 0 \wedge a = -1 \\ 0 & , \quad \text{sonst} \end{cases} \tag{5.4}$$

$$\text{mit } \delta, a \in \mathbb{Z} \wedge \delta \in [0, 11] \wedge a \in [-1, +1]$$

Die gewichtete Summe ergibt sich damit insgesamt zu:

$$R_{ges} = \sum_i a_i \cdot R_i = \Big(a_1 \cdot R_1 + a_2 \cdot R_2 + a_3 \cdot R_3 \Big) \tag{5.5}$$

Beim Skalieren der Gewichtungsfaktoren a_i ist zu beachten, dass jeder Teil der Summe im Worst-Case mit -1 bestraft wird und sich somit im Intervall $[0, -1]$ bewegt. Diese Art der Skalierung (engl. reward scaling) und Begrenzung (engl. reward clipping) von Belohnungen oder Strafen wird zur Vermeidung von Problemen bei den Ableitungen für den Gradientenabstieg im Deep Q-Learning angewendet [54]. Nach einigen Tests ergeben sich die Gewichtungsfaktoren zu $a_1 = \frac{5}{2}$, $a_2 = \frac{1}{11}$ und $a_3 = 1$. Somit legt der Agent Wert auf die Einhaltung der Sollwerte und versucht gleichzeitig hohe Spannungseinbrüche zu vermeiden.

5.6 Optimierung von Hyperparametern

Die Suche nach den optimalen Hyperparametern spielt eine wichtige Rolle für die Gesamtleistung von Modellen im Bereich maschinelles Lernen. Aufgrund der wachsenden Größe von Datensätzen und der steigenden Komplexität von Modellen, erhöht sich die Trainingsdauer dieser Modelle. Zudem werden mehr Ressourcen erfordert, was die Hyperparameteroptimierung noch komplexer macht. In der Wissenschaft haben sich mehrere Lösungen für die Optimierung von Hyperparametern entwickelt. Diese unterscheiden sich in Bezug auf die Rechenkomplexität und Skalierbarkeit [125]. Neben der manuellen Suche, Gittersuche (grid search) und Zufallssuche [60] (random search) haben sich noch Verfahren wie Bayes´sche Optimierungstechniken [126, 127] durchgesetzt. Das Bestimmen der besten Hyperparameterkonfiguration ist ein sequentieller Entscheidungsprozess [128], bei dem zunächst Anfangsparameter definiert und dann durch eine Mischung aus Intuition und Trial-and-Error angepasst werden, um die Genauigkeit zu maximieren oder den Verlust zu minimieren. Daher ist es wichtig, eine effiziente und skalierbare Strategie für das

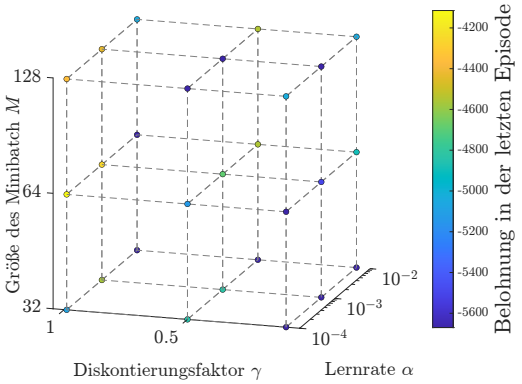
Hyperparameter-Tuning zu finden.

Das in dieser Arbeit verwendete Deep Q-Learning hat eine große Anzahl von Hyperparametern. Ein grundlegendes Problem in der Literatur besteht darin, dass in vielen Fällen nur Teile der angewendeten Hyperparameter angegeben werden. Ob und wie die Hyperparameteroptimierung durchgeführt wird, welche Parameter und Wertebereiche von den Autoren untersucht werden, ist in der Regel nicht dokumentiert. Dies wird auch von Henderson et al. [129] in ihrer Arbeit zur Reproduzierbarkeit von Deep RL Experimenten festgestellt. Zunächst stellt sich die Frage, welche Hyperparameter den größten Einfluss auf den Trainingsverlauf und die Leistung haben. Die in der Literatur angegebene Auswahl von Parametern bietet dabei einen Anhaltspunkt. Bei Wert-basierten Algorithmen spielen vor allem die Lernrate α, der Diskontierungsfaktor γ und die Größe des Minibatch M eine Rolle. Ein Blick auf die Aktualisierung der Wert-Funktion im Q-Learning-Algorithmus aus der Gleichung 2.25 zeigt den Effekt der Lernrate. Eine gut gewählte Lernrate kann, laut Kiran et al. [58], zu schnellen Lernfolgen in weniger Episoden führen. Die Autoren nennen auch den Diskontierungsfaktor γ, der die Priorisierung zwischen sofortigen und zukünftigen Belohnungen beeinflusst, sowie die Wahrscheinlichkeit ε zur Steuerung des Exploration-Exploitation Trade-off. Da ε in jeder Episode um einen bestimmten Prozentsatz reduziert wird, wird dieser Hyperparameter nicht in die Optimierung mit aufgenommen. Liessner et al. [130] betrachten die Minibatch M, den Diskontierungsfaktor γ, die Lernrate α und den Schichtaufbau des neuronalen Netzes als die wichtigsten Parameter in Deep RL. In dieser Arbeit wurden auf Basis der Recherche zunächst M, γ und α in die Optimierung einbezogen.

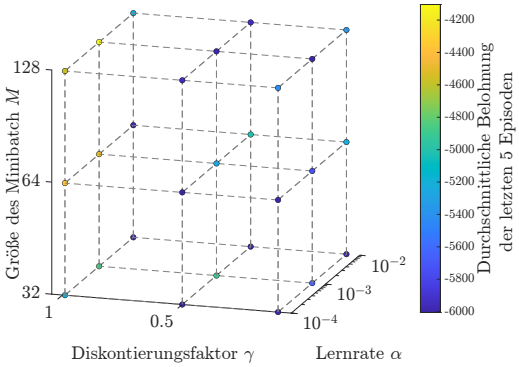
Da MATLAB im Vergleich zu bekannten Frameworks wie Scikit-Learn keine eingebauten Optimierungsmethoden bereitstellt, fällt die Wahl aufgrund des geringen zeitlichen Aufwands und der Zuverlässigkeit in niedrigdimensionalen Suchräumen auf die weit verbreitete Grid Search [58]. Einen groben Anhaltspunkt dafür können die im Bereich Energiemanagement genannten Hyperparameter bieten. Aus der Literatur [12, 13, 16, 17, 124] wird deutlich, dass die meisten Autoren eine Minibatch-Größe $M = 64$, eine Lernrate $0,0001 \leq \alpha \leq 0,001$ sowie einen Diskontierungsfaktor $\gamma \geq 0,95$ verwenden. Für jeden Hyperparameter werden drei Werte ausgewählt, wodurch sich ein Parameterraster mit Minibatch-Größe $M = [32, 64, 128]$, Lernrate $\alpha = [0,0001; 0,001; 0,01]$ und Diskontierungsfaktor $\gamma = [0,1; 0,5; 0,99]$ ergibt. Die Zahl der möglichen Kombinationen und durchzuführenden Traininseinheiten beträgt $3^3 = 27$. Hier wird der Curse of Dimensionality der Grid Search deutlich. Je mehr Hyperparameter, desto mehr Dimensionen. Dadurch wird die Darstellung beeinflusst und die benötigte Zeit steigt exponentiell an.

Die Implementierung der Hyperparameteroptimierung erfolgt in Matlab. Das Training findet zunächst mit einem Ausweichmanöver bei 35 $\frac{km}{h}$ für 100 Episoden statt. Es wird davon ausgegangen, dass die ermittelten Hyperparameter auch für andere Manöver geeignet sind. Das Modell wird mit einer Grundleistung von 2500 W konfiguriert, wobei alle steuerbaren elektrischen Verbraucher zu Beginn eingeschaltet sind. Dies entspricht einem

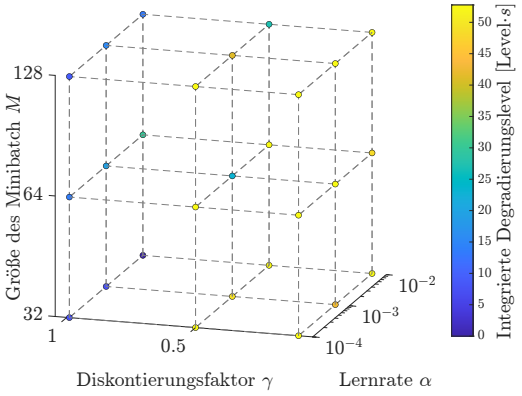
Worst-Case-Szeanrio. Da die Last hoch ist und zusätzlich durch die Hochleistungsverbraucher Instabilitäten im Bordnetz hervorgerufen werden, kann der Agent hierbei lernen, mit solchen Situationen umzugehen. Drei vollvernetzte versteckte Schichten wurden verwendet, um das neuronale Netzwerk zu konfigurieren. Die Anzahl der Neuronen wurde dabei auf 50 pro Schicht festgelegt. Um Probleme beim Gradientenabstieg während der Backpropagation (dt. Fehlerrückführung) zu vermeiden, wurde als Aktivierungsfunktion ReLU (Rectified Linear Unit) verwendet.
Die Ergebnisse der ersten Optimierungstests sind zum einen in Abbildung 5.3 ohne und in Abbildung 5.4 mit dem Haltemechanismus dargestellt. Abbildung 5.3 besteht aus fünf Teilfiguren (a) bis (e), die das Paramaterraster mit unterschiedlichen Bewertungskriterien auf der Farbskala darstellen. Jede Koordinatenachse repräsentiert einen Hyperparameter. Jeder farbige Punkt steht für eine Trainingseinheit mit einer anderen Kombination von Hyperparametern. Bei der Auswertung wurde deutlich, dass der Diskontierungsfaktor γ aus dem Parameterraster herausgenommen werden kann, da die endgültige Belohnung (siehe Abbildung 5.3(a)) nur für $\gamma = 0,99$ gute Ergebnisse zeigt. Für $\gamma \leq 0,99$ ist die Belohnung mehrheitlich geringer. Die Belohnung für die letzten fünf Episoden, dargestellt in Abbildung 5.3(b), funktioniert ähnlich. Die ergänzenden Abbildungen c bis e veranschaulichen den Kompromiss zwischen dem Eingriff des Agenten und damit dem Verlust des Insassenkomforts sowie einer Verringerung des Spannungsabfalls. Für alle Trainingseinheiten mit $\gamma = 0,5$ und $\gamma = 0,1$ ist in den Abbildungen 5.3(c) und (d) ersichtlich, dass der Agent sehr stark degradiert und daher den Spannungsabfall nahezu vollständig eliminiert. Dies vernachlässigt jedoch den Trade-off, der sich bei $\gamma = 0,99$ bemerkbar macht. Hier stehen Degradation und Spannungsabfall punktuell in einem besseren Verhältnis. Dies lässt den Schluss zu, dass Agenten, die mehr Wert auf kurzfristigere Belohnungen legen, für das Lernziel dieser Arbeit ungeeignet sind.
Daher werden anstelle des Diskontierungsfaktors zwei unterschiedliche Netzaufbauten des Critis in das Parameter-Grid aufgenommen. Diese basieren an die Formen aus der Literatur. Die unterschiedlichen Strukturen des neuronalen Netzes sind in Abbildung 5.5 zu sehen. Neben dem beschriebenen Netz mit 50 Neuronen in jeweils drei Hidden Layers wird auch eine Pyramidenform mit drei Hidden Layers implementiert. Die Neuronenanzahl (75, 50 und 25) nimmt in Richtung Ausgangsschicht ab. Die Gesamtzahl der Neuronen ist jedoch in beiden Strukturen gleich und damit auch die Anzahl der trainierbaren Gewichte im neuronalen Netz. Aus der Abbildung kann entnommen werden, dass alle Schichten vollvernetzt (eng. fully connected) ausgeführt sind. Für die Aktivierungsfunktion wird weiterhin ReLU verwendet. Die Anzahl der Neuronen in der Eingans- und Ausgangsschicht sind durch die Zustands- und Aktionsräume aus Kapitel 5.2 und 5.4 definiert.



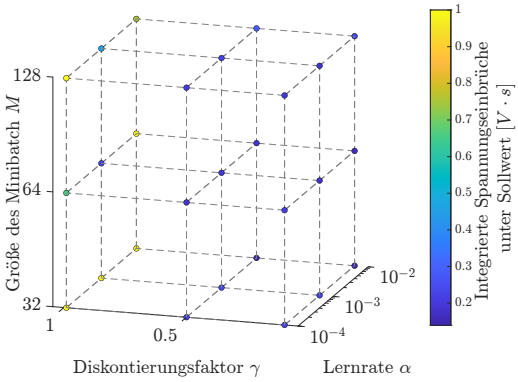
(a) Belohnung in der letzten Episode



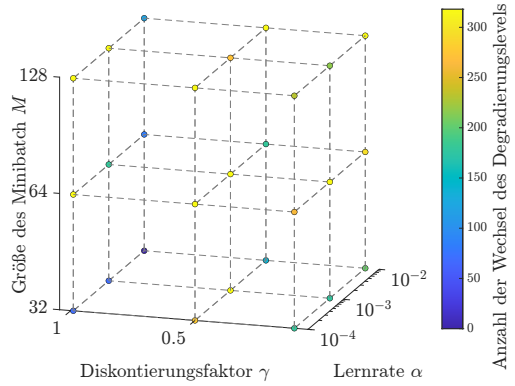
(b) Durchschnittliche Belohnung der letzten 5 Episoden



(c) Integrierte Degradierungslevel

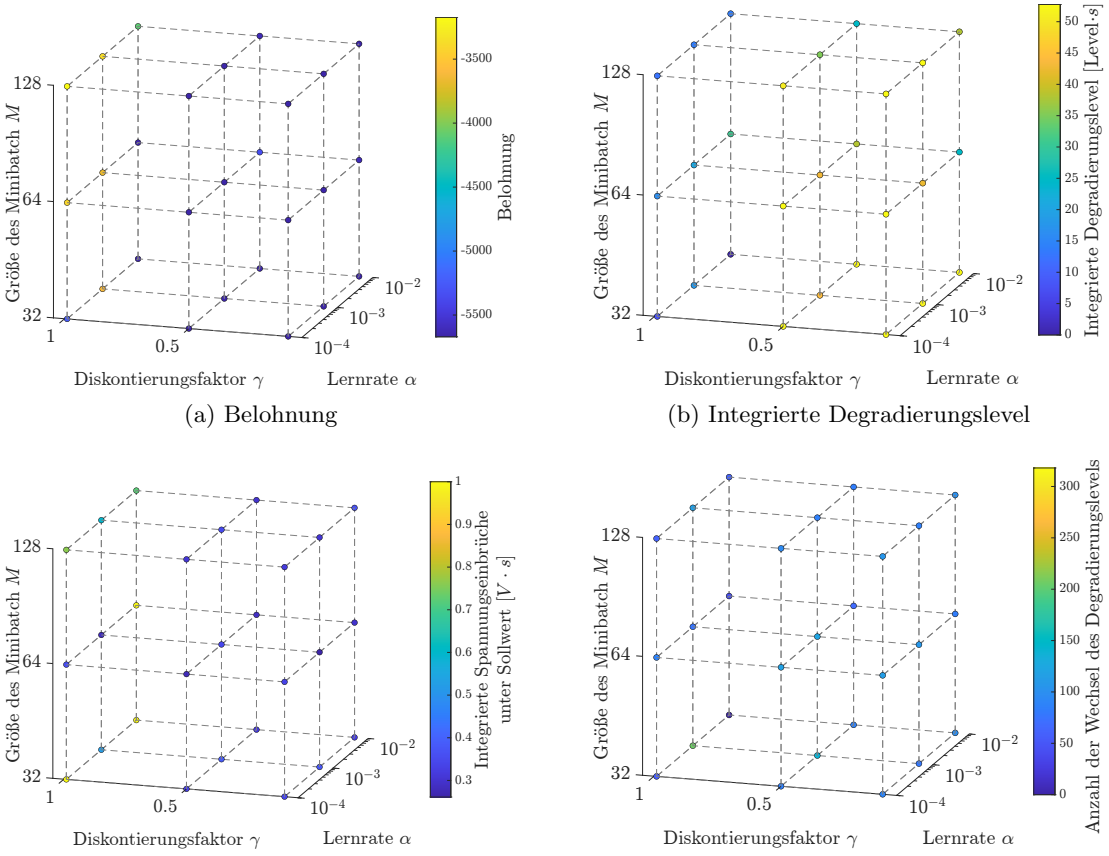


(d) Integrierte Spannungseinbrüche



(e) Anzahl der Wechsel des Degradierungslevels

Abbildung 5.3: Grid Search der Hyperparameter Lernrate $\alpha = [10^{-2}; 10^{-3}; 10^{-4}]$, Diskontierungsfaktor $\gamma = [0,1; 0,5; 0,99]$ und der Minibatch-Größe $M = [32; 64; 128]$. Aktionen des Agenten werden im Raster von 1 ms ausgeführt.



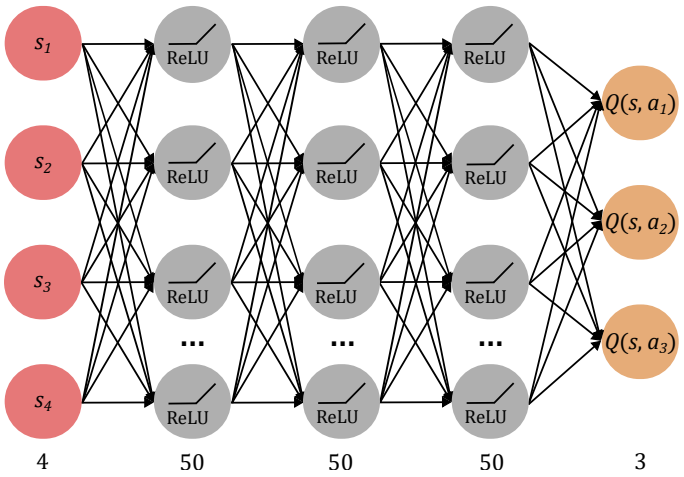
(a) Belohnung

(b) Integrierte Degradierungslevel

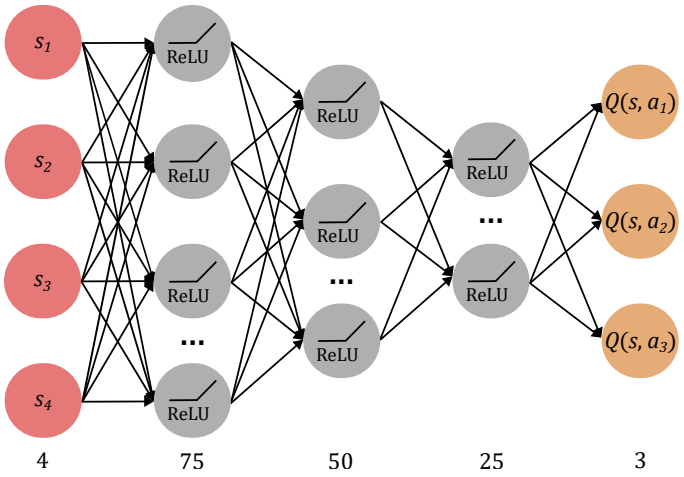
(c) Integrierte Spannungseinbrüche

(d) Anzahl der Wechsel des Degradierungslevels

Abbildung 5.4: Grid Search der Hyperparameter Lernrate $\alpha = [10^{-2}; 10^{-3}; 10^{-4}]$, Diskontierungsfaktor $\gamma = [0,1; 0,5; 0,99]$ und der Minibatch-Größe $M = [32; 64; 128]$. Degradierungslevel werden nach Wechsel für 10 ms gehalten.



(a) Neuronales Netz mit gleichbleibender Neuronenanzahl in den versteckten Schichten



(b) Neuronales Netz mit pyramidenartiger Form

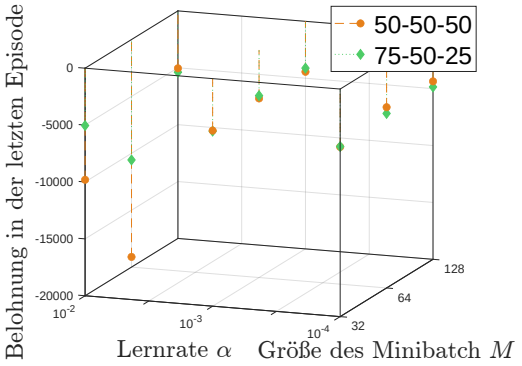
Abbildung 5.5: Grafische Darstellung der genutzten Topologien der neuronalen Netze. In der oberen Grafik ist ein neuronales Netz mit gleichbleibender Neuronenanzahl in den versteckten Schichten dargestellt. Untere Abbildung zeigt eine Pyramiden-artige Form.

Abbildung 5.6 zeigt die Ergebnisse der Grid Search mit geänderter Parameterwahl. Wird die Belohnung der letzten Episode aus der Abbildung 5.6(a) als Anhaltspunkt für die Performance des Agenten genommen, stechen die folgenden drei Kombinationen der Hyperparameter hervor:

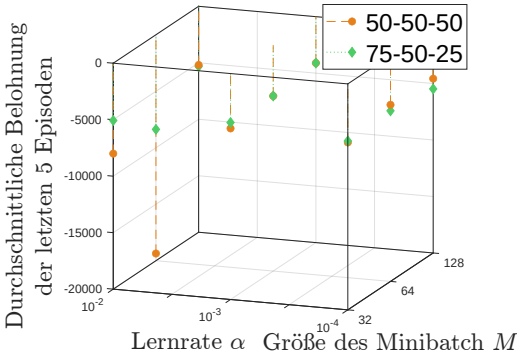
Tabelle 5.3: Übersicht über die Hyperparameter-Kombinationen der getesteten besten drei Agenten.

Platz	**Lernrate** α	**Größe des Minibatch** M	**Netzform**
1	$0,001$	64	75-50-25
2	$0,001$	128	75-50-25
3	$0,0001$	64	50-50-50

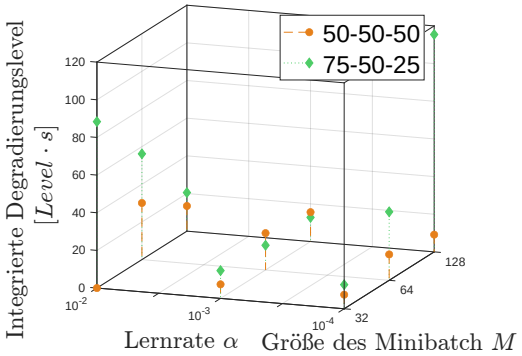
Gleichzeitig stellen sie den besten Kompromiss zwischen dem Eingreifen des Agenten und Reduzierung des Spannungsabfalls dar. Dies wird in den Abbildungen 5.6(c) und (d) deutlich gezeigt. Diese veranschaulichen die Flächenintegrale der Degradierungslevel und Spannungseinbrüche. Dabei gilt: je kleiner der Wert, desto besser. Bei den Hyperparameter-Kombinationen in Tabelle 5.3 sind die Flächenintegrale aus den Teilabbildungen 5.6(c) und (d) nicht so hoch, wie die restlichen Kombinationen. Dies deutet auf ein Gleichgewicht zwischen Eingreifen des Agenten und Reduktion der Spannungseinbrüche. Dahingegen ist der Kompromiss bei einer Minitbach-Größe von 32 für eine gleichverteilte Netzstruktur nicht gut abgebildet. Das nicht-eingreifen des Agenten führt zu hohen Spannungseinbrüchen.
Teilabbildung 5.6(e) stellt ein Defizit bei der Anzahl der Wechsel des Degradierungslevels dar. Diese Nachteile sind auch in Abbildung 5.7 zu erkennen, die das simulierte Verhalten der drei Agenten in einem Ausweichmanöver gegenüberstellt. Hierbei werden die Batteriespannung U_{BATT} und das Degradierungslevel dargestellt.
Aus der Abbildung 5.7 kann entnommen werden, dass der Spannungseinbruch, der ab ca. $t = 5\ s$ eintritt, durch die Eingriffe des Agenten in weiten Teilen gedämpft wird. Hierbei schaffen alle ausgewählten Agenten, die Spannung über 14 V zu halten. Die Problematik mit hochfrequenten Wechseln zwischen zwei benachbarten Degradierungsleveln ist im Intervall $7\ s < t < 10\ s$ deutlich zu sehen. Dies wirkt sich zum einen nachteilig auf den Komfort der Fahrzeuginsassen, zum anderen entstehen dabei Schaltungsverluste.



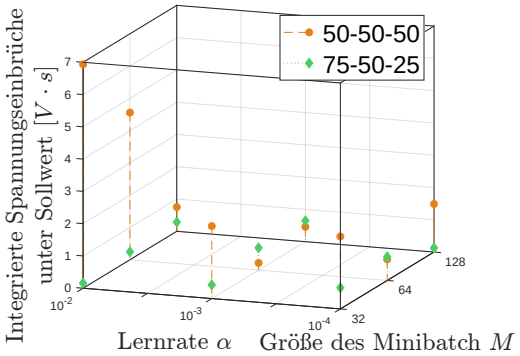
(a) Belohnung in der letzten Episode



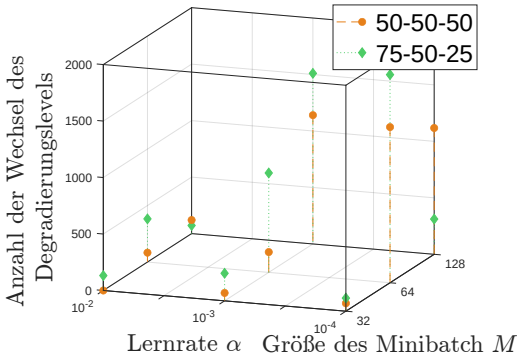
(b) Durchschnittliche Belohnung der letzten 5 Episoden



(c) Integrierte Degradierungslevel



(d) Integrierte Spannungseinbrüche



(e) Anzahl der Wechsel des Degradierungslevels

Abbildung 5.6: Grid Search der Hyperparameter Lernrate $\alpha = [10^{-2}; 10^{-3}; 10^{-4}]$, Minibatch-Größe $M = [32; 64; 128]$ und dem Aufbau des dreischichtigen neuronalen Netzes mit [50-50-50;75-50-25] Neuronen. Diskontierungsfaktor festgesetzt auf $\gamma = 0,99$. Aktionen des Agenten werden im Raster von 1 ms ausgeführt.

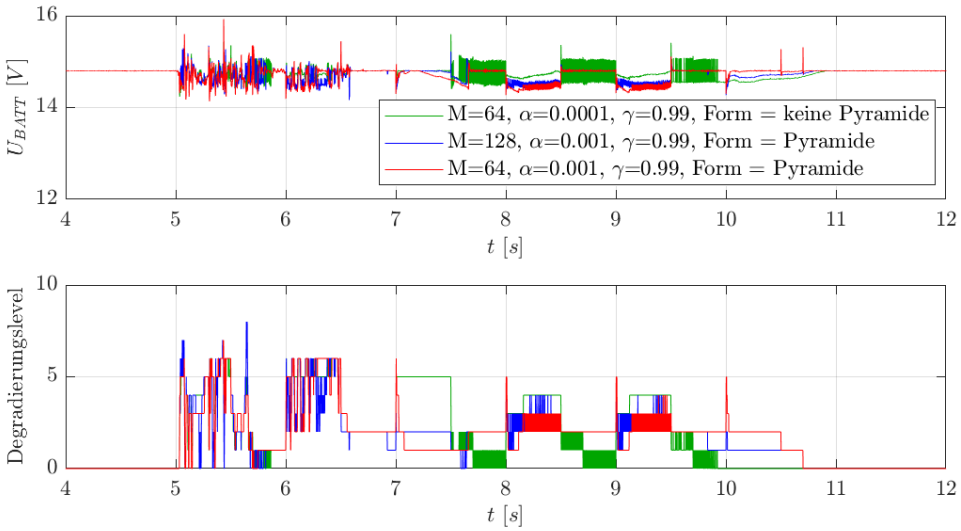
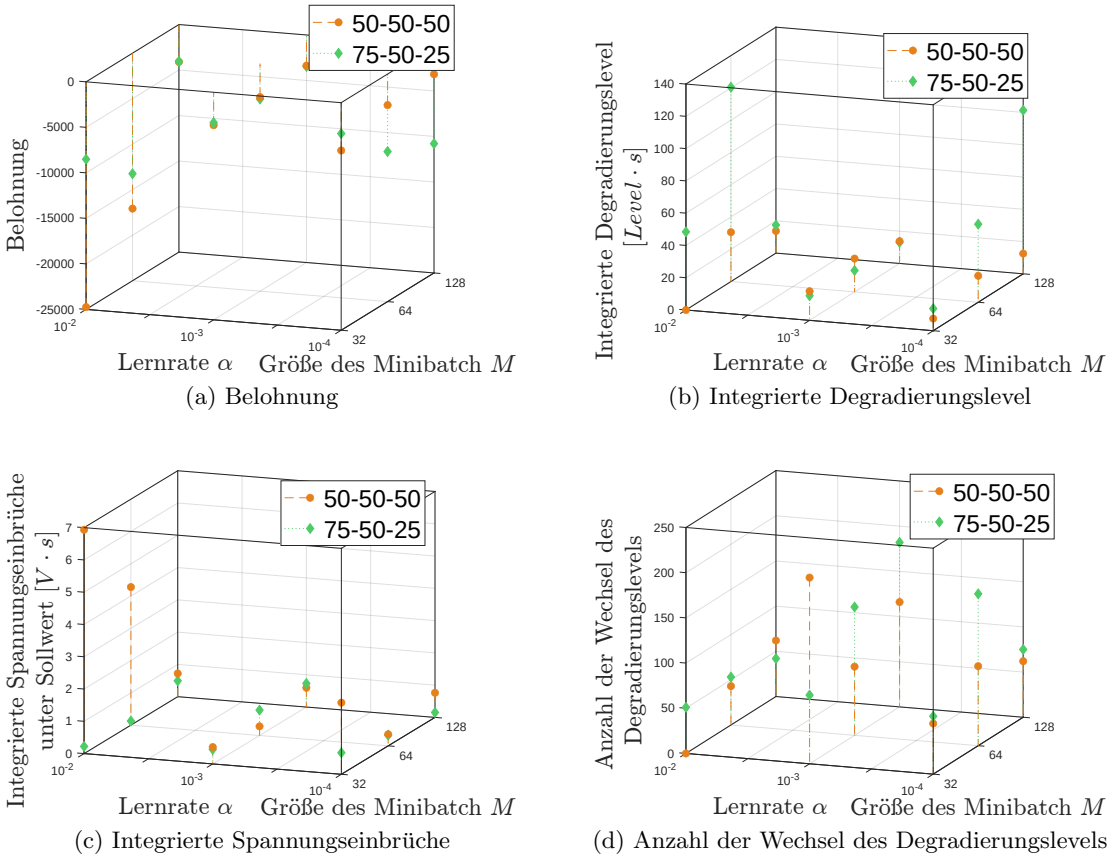


Abbildung 5.7: Vergleich der besten drei Agenten im Ausweichmanöver. Obere Abbildung stellt die Einbrüche der Batteriespannung dar. Unten sind die Wechsel der Degradierungslevel für die jeweiligen Agenten abgebildet.

Das Problem kann durch das Halten eines Degradierungslevels nach einem Wechsel für 10 ms, wie bereits in Kapitel 5.4 beschrieben, behoben werden. Das Ergebnis der Tests mit dem Mechanismus des Haltens ist in Abbildung 5.8 dargestellt. Dabei fand kein zusätzliches Training stand. Daher kann die Episoden-abhängige Belohnung der letzten 5 Episoden in der Abbildung nicht dargestellt werden.

In Abbildung 5.8(d) ist zu erkennen, dass die Anzahl der Wechsel des Degradierungslevels gegenüber Abbildung 5.6(e) deutlich abgenommen haben. Dabei ist die Belohnung in Abbildung 5.8(a) vergleichbar mit der Belohnung aus 5.6(a). Auf die Aufnahme von der Anzahl der Wechsel des Degradierungslevels in die Belohnungsfunktion wird verzichtet. Die Lernkurve leidet aufgrund der komplexen Belohnungsfunktionen. Zudem müsste die Trainingsdauer erhöht sowie die Optimierung wiederholt werden.

Abbildung 5.9 zeigt die besten drei Agenten analog zu Abbildung 5.7 mit dem Mechanismus, dass die Degradierungslevel nach einem Wechsel für 10 ms gehalten werden. Im Vergleich zu Abbildung 5.7 ist zu erkennen, dass Degradierungslevel überwiegend länger als 10 ms gehalten werden. Der Agent aus Abbildung 5.9 kann die Auswirkungen des für 10 ms festgesetzten Wechsel im Degradierungslevel über die Zustände deutlicher beobachten. Anschließend bleibt er in dem Degradierungslevel, da er den Wert besser abschätzen kann. Der Mechnismus des Halten hat somit einen filterartigen Effekt auf die Aktionen des Agenten. Jedoch ist im Intervall $8s < t < 10s$ zu sehen, dass die hochfrequenten Wechsel nicht vollständig vermieden werden können. Um die Reaktionsgeschwindigkeit des Agenten nicht zu verringern, wurde die Überlegung den Agenten statt im 1 ms-Takt nur alle 10 ms zu berechnen, verworfen. Außerdem würde das bedeuten, dass der Agent mehr Episoden benötigt, um den gleichen Lernerfolg zu erreichen.



(a) Belohnung

(b) Integrierte Degradierungslevel

(c) Integrierte Spannungseinbrüche

(d) Anzahl der Wechsel des Degradierungslevels

Abbildung 5.8: Grid Search der Hyperparameter Lernrate $\alpha = [10^{-2}; 10^{-3}; 10^{-4}]$, Minibatch-Größe $M = [32; 64; 128]$ und dem Aufbau des dreischichtigen neuronalen Netzes mit [50-50-50;75-50-25] Neuronen. Diskontierungsfaktor festgesetzt auf $\gamma = 0,99$. Degradierungslevel werden nach Wechsel für 10 ms gehalten.

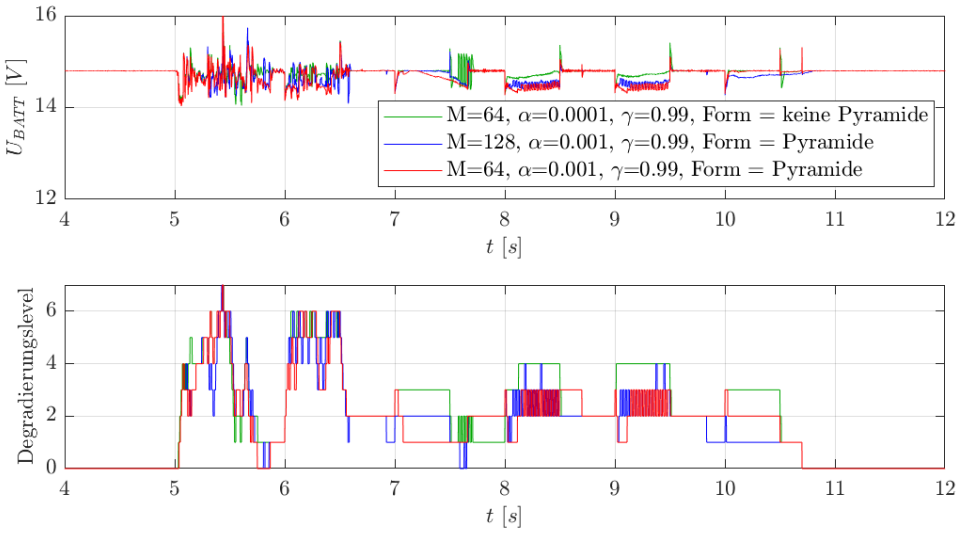
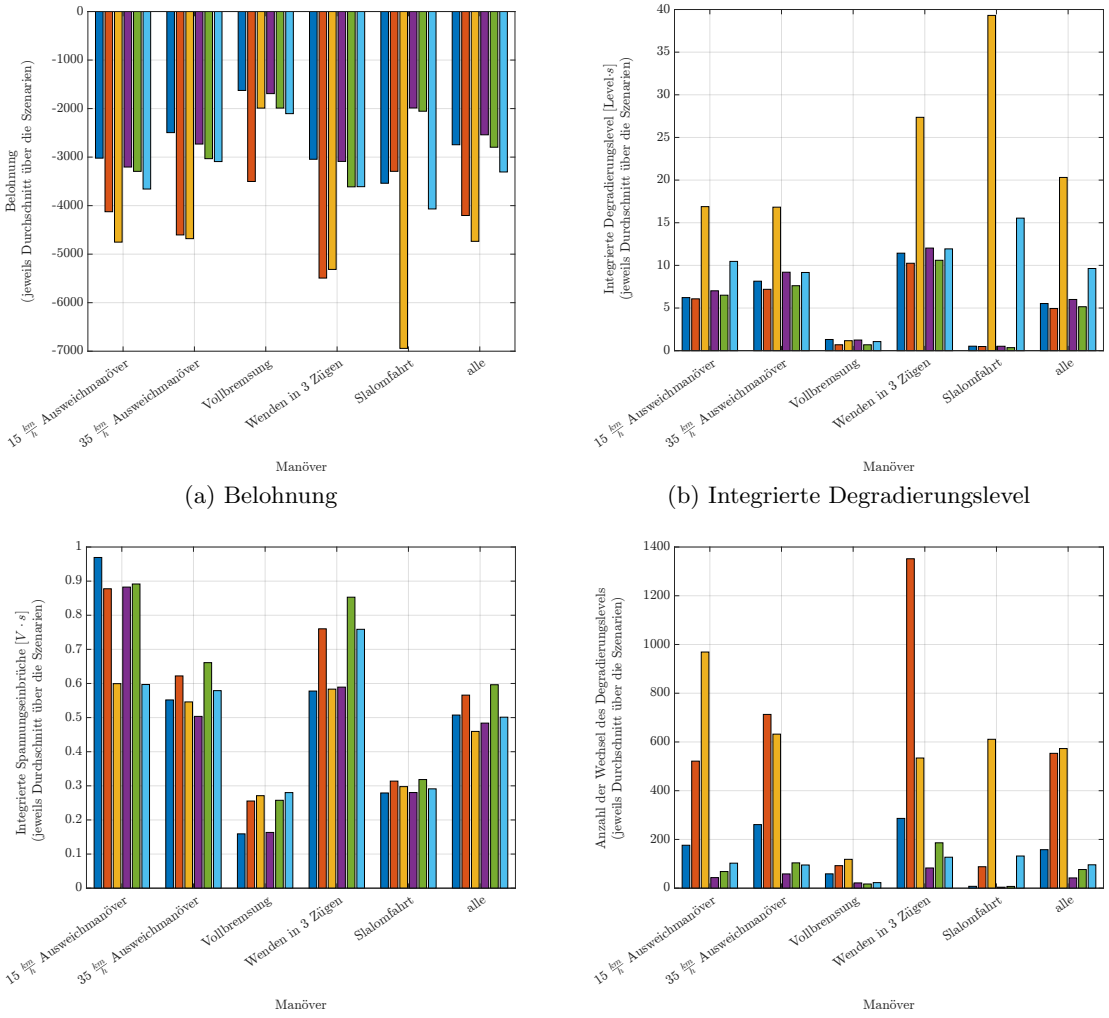


Abbildung 5.9: Vergleich der besten drei Agenten im Ausweichmanöver, wobei das Degradierungslevel nach einem Wechsel für 10 ms gehalten wird.

Da die Optimierung auf einem einzigen Szenario basiert, werden im Anschluss alle vorhandenen Szenarien mit den drei besten Agenten simuliert und ausgewertet. Bei Abbildung 5.10 handelt es sich um ein Säulendiagramm. Es beruht auf dieselben Bewertungskriterien wie aus der Abbildung 5.8.

Zur Simulation von allen Szenarien wurde die gemessene Leistung von Hochleistungsverbrauchern in das Modell aufgenommen. Alle Szenarien werden mit der gleichen Grundlast simuliert. Dadurch ist sichergestellt, dass die Ergebnisse in jeder Situation vergleichbar sind, da immer von der gleichen kritischen Bordnetzsituation ausgegangen wird.

Das Säulendiagramm vergleicht die drei ausgewählten Agenten mit sowie ohne dem Halten des Degradierungslevels. In allen Teilabbildungen stellt die x-Achse die fünf unterschiedlichen Manöver, die in Kapitel 4.1 beschrieben wurden, und ein Durchschnitt über alle Manöver dar. Die einzelnen Balken stellen auch den Durchschnitt aller verfügbaren Szenarien für ein bestimmtes Manöver. Jede Gruppe von sechs Balken besteht aus den besten drei Agenten. Diese Agenten wurden jeweils mit und ohne den Mechanismus des Haltens für 10 ms simuliert. Dabei stellen die drei linken Balken (dunkelblau, orange, gelb) den Agenten ohne Mechanismus dar, die drei rechten Balken (lila, grün, hellblau) den Agenten mit dem Mechanismus. Als erstes wird der Fokus auf die ersten drei Balken gelegt. In Abbildung 5.10(a) wird deutlich, dass der dunkelblaue Agent bei vier von fünf Manövern stets die maximale Belohnung erhält. Bei der Slalomfahrt ist der dunkelblaue Agent dem orangen Agenten nur geringfügig unterlegen. Insgesamt ist zu erkennen, dass unter den nicht-pyramidenartigen Netzstrukturen der dunkelblaue Agent über alle Manöver hinweg die höchste Belohnung erzielt. Bei der Untersuchung des Kompromisses zwischen reduziertem Spannungsabfall und Verlust des Insassenkomforts, dargestellt in den Abbildungen 5.10(b) bis (d), zeigt, dass der dunkelblaue Agent hier ebenfalls die besten Ergebnisse



(a) Belohnung

(b) Integrierte Degradierungslevel

(c) Integrierte Spannungseinbrüche

(d) Anzahl der Wechsel des Degradierungslevels

Abbildung 5.10: Vergleich der drei besten Agenten ohne sowie mit dem Halten des Degradierungslevels nach einem Wechsel für 10 ms in verschiedenen Manövern. Die Werte der Balken werden jeweils über den Durchschnitt aus allen verfügbaren Szenarien eines Manövers berechnet.

liefert. Mit Ausnahme des Ausweichmanövers mit 15 $\frac{km}{h}$ ist seine Leistung bei den integrierten Spannungseinbrüchen in Abbildung 5.10(c) gleich oder besser als die beiden Konkurrenten. Außerdem ist das Integral der Degradierungslevel nur geringfügig höher als der in orange dargestellte Agent. Der gelb markierte Agent hat zwar das niedrigste Flächenintegral der Spannungseinbrüche über alle Manöver, jedoch degradiert er mehr als doppelt so stark. Die stärkste Runterstufung der elektrischen Verbraucher findet in der Slalomfahrt statt. Basierend auf die Analyse der Fahrzeugmessungen aus der Abbildung 4.3, ist bei diesem Manöver der Spannungseinbruch gegenüber den anderen Manövern gering. Dadurch lässt es sich vermuten, dass der gelbe Agent das Lernziel in Bezug auf geringe Komforteinbußen verfehlt hat. Die Analyse aus der Abbildung 5.10(d) stellt die Defizite des orange markierten Agenten dar. Zwar bringt dieser Agent in Bezug auf die integrierten Degradierungslevel und Spannungseinbrüche ähnlich gute Ergebnisse, jedoch wechselt er die Degradierungslevels öfter als der dunkelblau markierte Agent, der in jeder Schicht des Critic die gleiche Neuronenanzahl besitzt. Dadurch entsteht ein unruhiges Verhalten des Agenten.

Die Tatsache, dass die nicht-pyramidenartige Netzstruktur in Kombination mit anderen Hyperparametern bessere Leistung bringt, wird auch deutlich, wenn die drei Agenten mit Anwendung des Mechanismus zum Halten der Degradierungslevel verglichen werden. Der hellblau markierte Agent repräsentiert den gelben Agenten mit dem Haltemechanismus. Aus den Teilabbildungen 5.10(a) und (b) wird deutlich, dass der hellblaue Agent bessere Leistungen bringt, als der gelb markierte Agent. Die Ergebnisse des orangen und hellgrünen Agenten sind nahezu gleich geblieben. Eine signifikante Verbesserung ist nur im Bereich der Änderung des Degradierungslevels in Teilabbildung 5.10(d) zu sehen. Jedoch ist in der Teilabbildung 5.10(a) zu sehen, dass der lila markierte Agent unter Anwendung des Haltemechanismus im Vergleich zu dem grünen und hellblauen Agenten eine höhere Belohnung erzielt. Auch im Vergleich der integrierten Spannungseinbrüche und der Anzahl der Wechsel des Degradierungslevels schneidet der lila markierte Agent besser ab.

Der dunkelblaue Agent und der lila markierte Agent sind im Wesentlichen gleich und unterscheiden sich nur durch die Anwendung des Haltemechanismus. Nach vorangegangener Analyse hat es sich herausgestellt, dass diese beiden Agenten die besten Ergebnisse liefern. Deshalb wird für die weiteren Experimente in den folgenden Kapiteln die Hyperparameter-Kombination mit $M = 64$, $\alpha = 0,0001$, $\gamma = 0,99$ verwendet. Somit wird als Netzstruktur des neuronalen Netzes im Critic eine konstant bleibende Neuronenanzahl gewählt. Aus der Analyse ist festzustellen, dass das Haltemechanismus die Spannungseinbrüche nicht mehrheitlich verbessert. Auch im Vergleich der Belohnungen bringt dieses Mechanismus keine großen Vorteile. Da aber die Eingriffe der Agenten reduziert werden, wird deren Verhalten stark beruhigt.

5.7 Training mit zufälliger Szenarienwahl

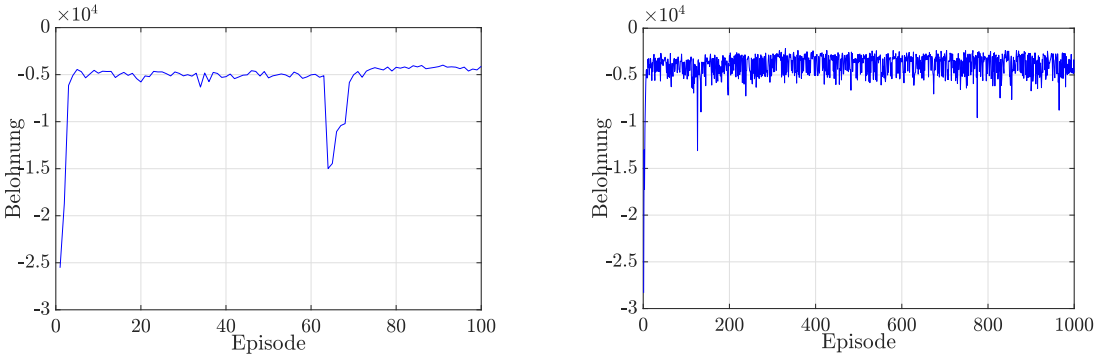
Das Training der Hyperparameter-Optimierung aus dem vorherigen Kapitel basierte auf einem einzelnen Ausweichmanöver bei 35 $\frac{km}{h}$. Mit dem Ziel eine bessere Generalisierung zu erreichen, wird in diesem Kapitel ein längeres Training mit zufällig ausgewähltem Szenario gestartet. In jeder Episode wird ein Szenario zufällig aus einer Gleichverteilung der verfügbaren Messreihen gezogen. Die definierte Anzahl der Episoden ist größer als die Gesamtzahl der verfügbaren Szenarien. Somit treten alle Szenarien mit einer gleichen Häufigkeit auf. Unter den verfügbaren Manövern verursachen die Ausweichmanöver sowie das Wenden in 3 Zügen die größten Spannungseinbrüche. Abbildung 5.11 zeigt die errechnete Belohnung von zwei Trainingsprozessen. In der linken Abbildung ist die Belohnung von dem besten Agenten aus Kapitel 5.6 dargestellt. Hier wurde der Agent im selben Szenario für 100 Episoden trainiert. Die rechte Grafik zeigt die Belohnung eines Trainingsprozess, indem der Agent mit zufälliger Szenarienwahl für 1000 Episoden trainiert wurde. Die Hyperparameter aus den beiden Trainingsprozessen sind identisch.

Die Belohnungskurven der beiden Abbildungen zeigen, dass der primäre Lernerfolg der beiden Agenten während den ersten zehn Episoden stattfindet. Weiterhin wird deutlich, dass die Varianz der Belohnung pro Episode durch die zufällige Situationsauswahl deutlich höher ist. Der mittlere Belohnungswert aus der Abbildung 5.11(b) ist größer als der Wert in der Teilabbildung (a). Dies ist ein Zeichen dafür, dass durch das längere Training ein besserer Kompromiss für alle Manöverarten gefunden wurde. Um diese Aussage zu bekräftigen, muss der Agent aus der Abbildung 5.11(a) aus allen Szenarien getestet werden. Eine Analyse dazu folgt in diesem Kapitel.

In beiden Abbildungen treten immer wieder größere negative Belohnungen auf. In der linken Grafik ist dies beispielhaft in der Episode 64 zu sehen. In der rechten Abbildung kann Episode 126 als Beisiel aufgezeigt werden. Diese zeigen Situationen an, in denen der Agent aufgrund kleiner Änderungen in der Wertfunktion eine signifikante Verschlechterung der Policy erfährt. Dieser Nachteil ist auf das Prinzip Wert-basierter Algorithmen zurückzuführen. Bei diesen wird die Policy von der Wertfunktion abgeleitet und nicht direkt optimiert.

Um die Generalisierbarkeit der beiden trainierten Agenten besser vergleichen zu können, wurden diese in allen Szenarien simuliert. Der Vollständigkeit halber wurde für jeden Agent der Haltemechanismus aus dem vorherigen Kapitel angewendet. Damit ergeben sich vier verschiedene Verhaltensstrategien, die in folgender Abbildung 5.12 nach verschiedenen Kriterien als Säulendiagramme ausgewertet werden. Ähnlich wie in Abbildung 5.10 repräsentieren in der Gruppe der vier Verhaltensstrategien die beiden linken Balken den Agenten ohne Haltemechanismus, während die beiden Balken rechts das Ergebnis des Agenten mit dem Mechanismus darstellen.

In Teilabbildung 5.12(a) zeigen die beiden linken Agenten den Vergleich der Belohnungen unter den Agenten ohne dem Haltemechanismus. Mit Ausnahme des 35 $\frac{km}{h}$ Ausweichmanövers zeigen diese beiden Balken, dass der über 1000 Episoden auf zufälligen Szenari-



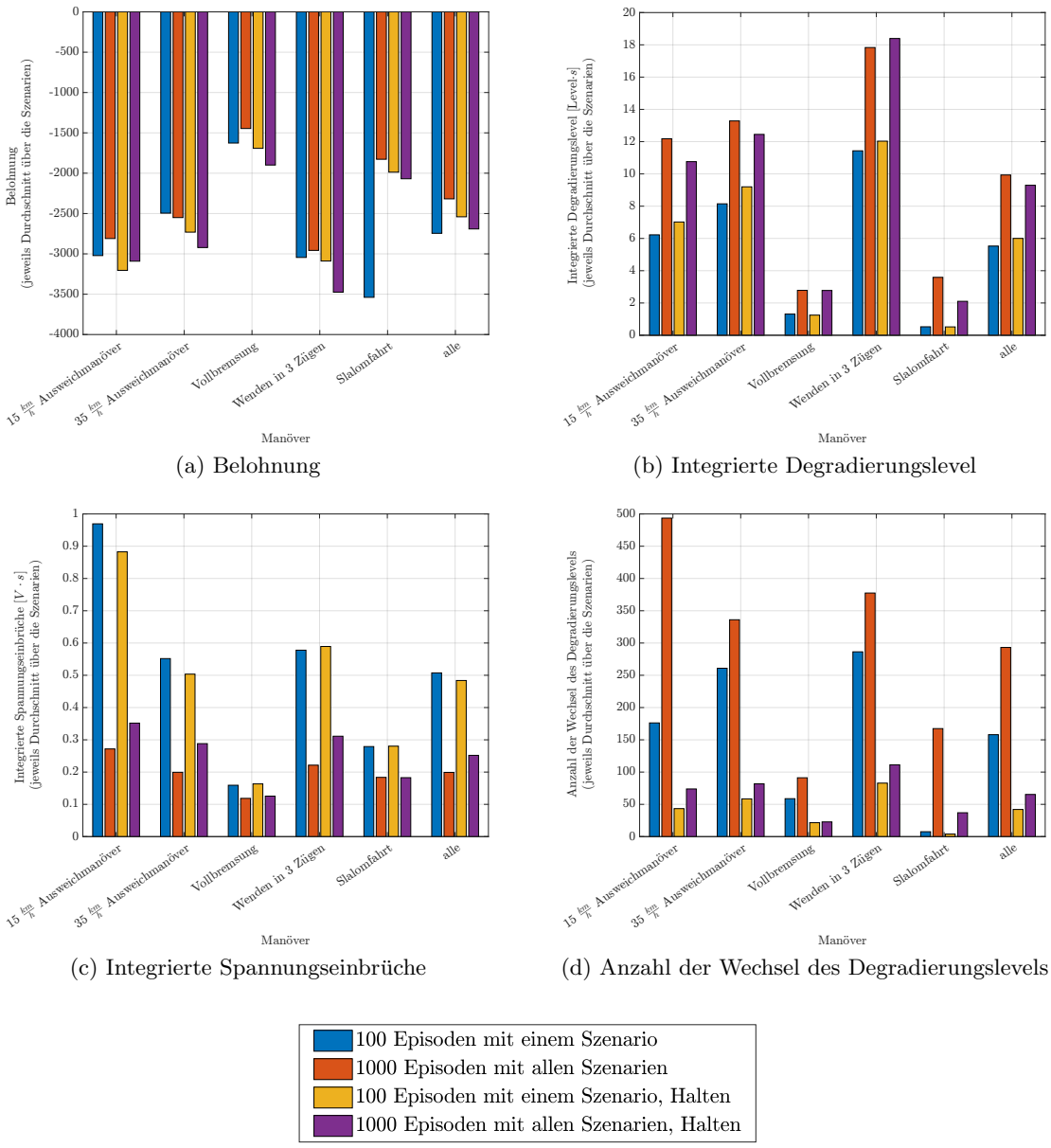
(a) Training mit 100 Episoden auf einem Szenario

(b) Training mit 1000 Episoden und zufälliger Szenarienwahl

Abbildung 5.11: Vergleich der Belohnung zwischen Trainings mit einem und allen Szenarien. Bei einem Training mit mehreren Episoden und allen Szenarien ist der mittlere Belohnungswert größer als der Wert aus der linken Abbildung.

en trainierte Agent in oranger Markierung die höchste Belohnung unter all den Agenten erzielt. Somit übertrifft er auch den besten Agenten aus Kapitel 5.6, der hier in blau markiert ist. Dies lässt sich dadurch erklären, dass der blau markierte Agent nur auf das 35 $\frac{km}{h}$ Ausweichmanöver trainiert wurde und sich darauf spezialisiert hat. Der orange markierte Agent erzielt die größte Verbesserung bei der Slalomfahrt. Der Unterschied ist jedoch viel kleiner als die bei den unterschiedlichen Kombinationen von Hyperparametern aus Abbildung 5.10. Interessanterweise bewegt sich das Verhalten des orange markierten Agenten deutlich in Richtung stärkerer Degradierung. Diese Verschiebung ist in Abbildung 5.12(b) gut zu erkennen. Darin wird die integrierte Degradierungslevel dargestellt. Infolgedessen kann der in der Abbildung 5.12(c) gezeigte Spannungsabfall erheblich reduziert werden. Die integrierten Spannungseinbrüche können um mehr als die Hälfte reduziert werden. Die größte Einschränkung des orangefarbenen Agenten wird in 5.12(d) ersichtlich. Hierbei ist zu sehen, dass die Anzahl der Wechsel des Degradierungslevels deutlich zunimmt. Die maximalen Unterschiede ergeben sich dabei im Ausweichmanöver mit 15 $\frac{km}{h}$ und bei der Slalomfahrt. Insgesamt lässt sich festhalten, dass sich durch das Zusammenspiel des Agenten mit verschiedenen Manövern der Kompromiss zwischen der Reduktion von Spannungseinbrüchen und Degradierung mehr zugunsten einer höheren Wechsel der Degradierungslevels verschiebt. Dadurch ist die Belohnung des über 1000 Episoden trainierten Agenten höher als die des mit einem Szenario trainierten Agenten (blau).

Der Haltemechanismus bewirkt unterschiedliche Auswirkungen. Obwohl die Belohnung in Abbildung 5.12(a) in allen Situationen von dem blauen Agenten zum gelben Agenten zunimmt, nimmt die Belohnung des orangen Agenten zum lila markierten Agenten ab. Die Abbildungen 5.12(b) und (c) zeigen ähnliche Änderungen. Die Degradierung vom blauen zum gelben Agenten nimmt zu, wobei die Spannungseinbrüche annähernd gleich bleiben. Der Mechanismus vom orangen zum violetten Agenten hat dahingegen größtenteils den gegenteiligen Effekt. Lediglich in Abbildung 5.12(d) ist zu erkennen, dass die Anzahl der



(a) Belohnung

(b) Integrierte Degradierungslevel

(c) Integrierte Spannungseinbrüche

(d) Anzahl der Wechsel des Degradierungslevels

Abbildung 5.12: Vergleich des besten Agenten aus Kapitel 5.6 sowie des Agenten, der 1000 Episoden mit zufälliger Szenarienwahl auf Basis identischer Hyperparameter trainiert wurde. Die Werte der Balken werden jeweils über den Durchschnitt aus allen verfügbaren Szenarien eines Manövers berechnet.

Wechsel des Degradierungslevels mit dem Haltemechanismus deutlich sinkt.

Abschließend ist anzumerken, dass sich das Training mit einer höheren Episodenzahl und einer zufälligen Auswahl der Szenarien positiv auf die Leistung des Agenten auswirkt. Die Belohnung steigt über alle Manöver hinweg. Auch werden die Spannungseinbrüche reduziert. Als Ergebnis wird eine bessere Verallgemeinerung des Modells erhalten. Durch den Haltemechanismus können hohe Anzahl an Wechsel des Degradierungslevels verkleinert werden.

5.8 Training mit rekurrentem Q-Netz

Wie bereits in Kapitel 5.1 erwähnt, kann die Netzarchitektur des Q-Netzes mit MATLAB um rekurrente Schichten erweitert werden. Aufgrund der besonderen Fähigkeit von LSTM-Zellen, sequenzielle Daten zu verarbeiten und Beziehungen in Zeitreihen zu identifizieren, wird in diesem Kapitel untersucht, ob die Leistung der Agenten verbessert werden kann. Zudem wird erwartet, dass sich die Art der Eingriffe ändert. Das Gedächtnis der Zellen soll die Häufigkeit von Degradierungswechseln und die Reaktionszeit bei Spannungseinbrüchen reduzieren. LSTM-Schichten werden bereits erfolgreich im Bereich von HEVs eingesetzt [17]. Hierbei wird eine Architektur verwendet, in der zwei vollständig vernetzte Schichten die LSTM-Schicht umhüllen. Die Struktur für die vorliegende Arbeit ist in gleicher Weise gewählt. Basierend auf die Erkenntnisse der früheren Kapitel wurde jedoch die Anzahl der Neuronen in den Schichten konstant gewählt. Das rekurrente Netzwerk wird mit einer Sequenzlänge von 50 diskreten Zeitschritten bedatet. Bei einem Zeitschritt von einer Millisekunde entspricht dies einem Speicher von 50 ms. In Abbildung 5.13 ist das verwendete neuronale Netz dargestellt. Der in Kapitel 5.1 beschriebene Q-Vektor wird durch die drei Ausgangsneuronen repräsentiert.
Aufgrund der speziellen Struktur von LSTMs gibt es viermal so viele Gewichte wie in herkömmlichen rekurrenten Zellen. Dadurch wird das Trainieren erschwert und die Trainingszeit erhöht sich deutlich. Dadurch war es nicht möglich, die gleiche Anzahl von Episoden auszuwählen wie im vorangegangenen Kapitel 5.7. Dennoch wird durch die Adressierung der zeitlichen Zusammenhänge im Zustandsvektor eine Verhaltensänderung gegenüber dem bisherigen Agenten erwartet. Abbildung 5.14 vergleicht den Agenten mit FNN als Q-Netz und den Agenten mit LSTM-Zellen. Um die Auswirkungen des Haltemechanismus weiter zu analysieren, wird jeder Agent zusätzlich mit diesem Mechanismus ausgewertet. Aus dem vorherigen Kapitel 5.7 wurden alle Hyperparameter und Trainingsbedingungen beibehalten. Vier Säulendiagramme werden verwendet, um die berechnete Belohnung (a), die Flächenintegrale der Degradierungslevel (b) und Spannungseinbrüche (c) sowie die Anzahl der Wechsel des Degradierungslevels (d) in der Auswertung darzustellen.
Wird die Belohnung aus der Abbildung 5.14(a) betrachtet, so wird deutlich, dass die Verwendung eines rekurrenten Netzes mit LSTM-Zellen (orange) dem bisher besten Agenten

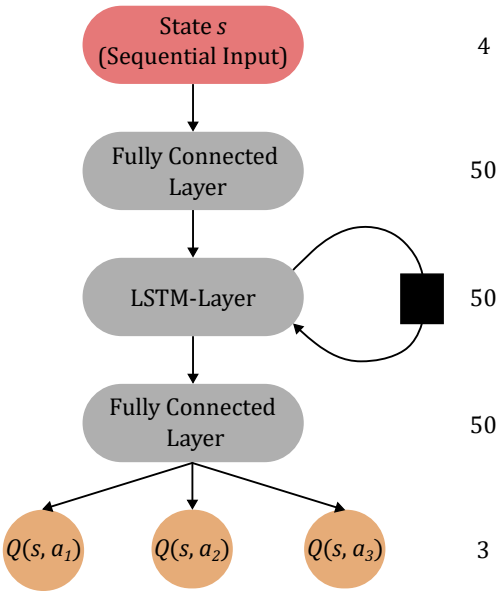
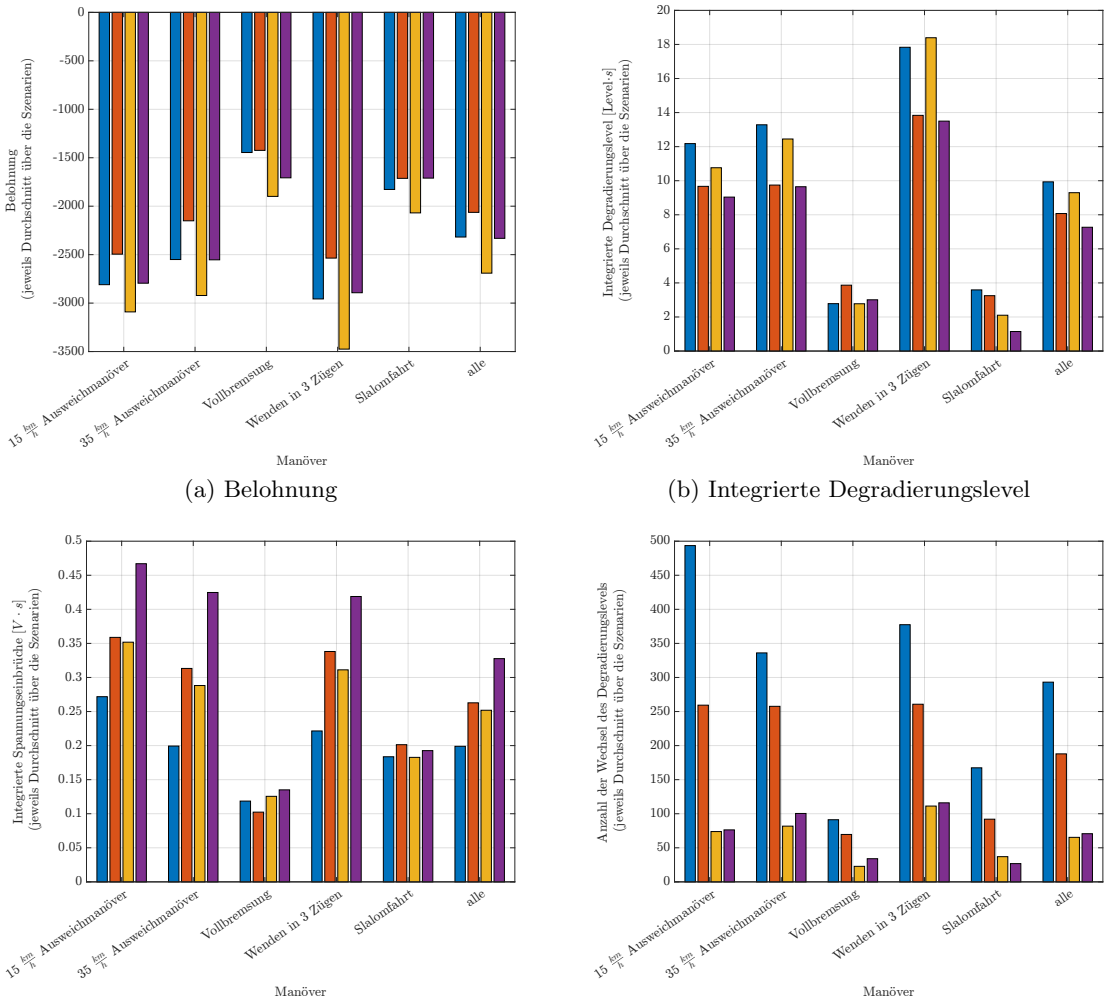


Abbildung 5.13: Darstellung der Netztopologie mit LSTM-Zellen und Q-Vektor als Ausgangsschicht.

(blau) in allen Manövern überlegen ist. Der kleinste Unterschied ist bei der Vollbremsung und Slalomfahrt zu beobachten. Jedoch müssen nach der Analyse aus Kapitel 4.2 bei diesen Manövern die geringsten Maßnahmen zur Stabilisierung der Bordnetzspannung ergriffen werden, da die Spannungseinbrüche nicht so stark ausgeprägt und von geringer Dauer sind. Letztendlich übertrifft dieses rekurrente Netz die bisherigen Netzstrukturen darin, den Kompromiss zu finden, den die Belohnungsfunktion sucht. Dies deutet einerseits darauf hin, dass der zeitliche Zusammenhang zwischen Zuständen bei der Bewältigung der Lernaufgabe eine Rolle spielt. Andererseits lernt der Agent sie und erzielt höhere Lernerfolge. Der Kompromiss zwischen der Reduzierung der Spannungseibrüche und dem Eingreifen des Agenten in den Abbildungen (b) und (c) hat sich zwischen den orangen und blau markierten Agenten verschoben. Da die Degradierung beim orange markierten Agenten abnimmt, kommt es zu stärkeren Spannungseinbrüchen. Hier ist ebenfalls zu bemerken, dass die Änderungen bei der Vollbremsung und Slalomfahrt minimal sind. Der orange markierte Agent kann jedoch den blauen Agenten bei diesen beiden Manövern in der Reduktion der Spannungseinbrüche übertreffen. Abbildung 5.14(d) zeigt den größten Nutzen des RNNs in Bezug auf die Lernaufgabe. Insgesamt verringert sich die Anzahl der Wechsel des Degradierungslevels um etwa 33 %. Die stärkste Veränderung vom blauen Agenten zum orangenen Agenten ist im Ausweichmanöver mit 15 $\frac{km}{h}$ zu erkennen. Dabei Wechselt der Agent nur noch halb so viel. Abbildung 5.15 zeigt das Beispiel eines Ausweichmanövers mit 15 $\frac{km}{h}$. Hierbei ist der Unterschied der Wechsel mit 2620 Wechsel des Degradierungslevels zwischen den beiden Agenten maximal.

In Abbildung 5.15 sind die Verhaltensstrategien des Agenten mit FNN (rot) und RNN (blau) dargestellt. Hierbei ist zu erkennen, dass der rot markierte Agent tendenziell mehr



(a) Belohnung

(b) Integrierte Degradierungslevel

(c) Integrierte Spannungseinbrüche

(d) Anzahl der Wechsel des Degradierungslevels

Abbildung 5.14: Vergleich der Agenten mit FNN und RNN. Die Werte der Balken werden jeweils über den Durchschnitt aus allen verfügbaren Szenarien eines Manövers berechnet.

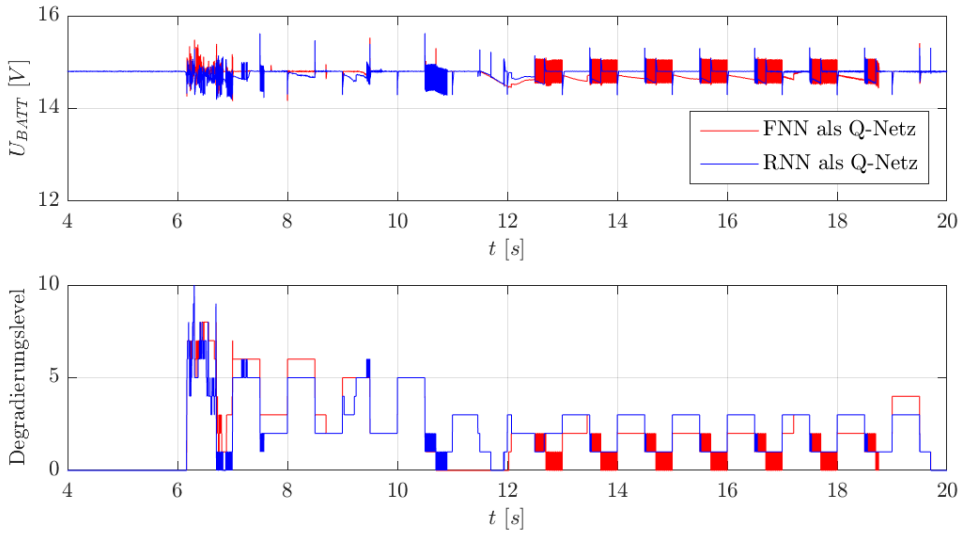


Abbildung 5.15: Vergleich der Verhaltensstrategie der Agenten mit FNN und RNN in einem Ausweichmanöver mit 15 $\frac{km}{h}$, in dem der höchste Unterschied in der Anzahl der Degradierungswechsel zwischen den beiden Agenten zu verzeichnen ist.

zu Beginn des Spannungseinbruchs ($6\ s < t < 12\ s$) degradiert. Ab dem Zeitpunkt $t > 12\ s$ sind die Degradierungslevel des blauen Agenten größer. Dieser Zeitrahmen zeigt auch den signifikanten Verhaltensunterschied. Der Agent mit FNN weist ein deutliches Rauschen auf, das sich jede Sekunde wiederholt, während der Agent mit RNN gelegentlich das Degradierungslevel für mehrere hundert Millisekunden ändert. Als Ergebnis wird deutlich, dass die Art der Verhaltensstrategie des Agenten durch die Verwendung von RNNs stabilisiert wird. Dabei wird das Flächenintegral des Spannungsabfalls nur um etwa $0,06\ Vs$ verringert.

Wie in den Kapiteln 5.6 und 5.7 bereits festgestellt wurde, beruhigt der Haltemechanismus die Eingriffe des Agenten. Jedoch wird dabei ein Leistungsabfall betrachtet. Die Schlussfolgerungen können in dieses Kapitel übernommen werden, wobei in den Abbildungen 5.14(a) bis (c) geringe Leistungsverluste erkennbar sind. Teilabbildung 5.14(d) zeigt entgegen den Erwartungen, dass die Anzahl der Degradierungsänderungen bei der Verwendung von RNNs im Vergleich zu dem Agenten mit FNN in einem geringeren Verhältnis abnimmt. Die Anzahl der Änderungen ist für den lila markierten Agenten mit RNN und Haltemechanismus höher als für den gelb markierten Agenten mit FNN. Die einzige Ausnahme ist die Slalomfahrt. Dies deutet auf einen Konflikt zwischen der Verwendung von RNNs und dem Haltemechanismus. Agenten mit LSTM-Zellen können zeitliche Bezüge in den Daten erkennen, dies wird jedoch durch den Mechanismus manipuliert. Aus diesem Grund wird festgestellt, dass der Haltemechanismus keine Vorteile in Bezug auf diese Aufgabenstellung liefert.

Der Agent mit LSTM-Zellen wird trotz der Einbußen bei der Reduzierung der Spannungseinbrüche insgesamt besser bewertet als der Agent mit Feedforward-Architektur aus

Kapitel 5.7. Bei allen Manövern erhält er eine größere Belohnung, erfüllt den gewünschten Kompromiss der Lernaufgabe besser und verändert die Degradierungslevel in geringerer Anzahl.

6 Ergebnisse

Im folgenden Kapitel werden die Ergebnisse des optimierten RL-Agenten vorgestellt. Anschließend wird der Ansatz mit einer Referenzstrategie verglichen. Die Erklärung der Referenzstrategie findet in Kapitel 6.1 statt. Außerdem wird der RL-Agent in Kapitel 6.3 gezielt mit unterschiedlichen Bordnetzauslastungen konfrontiert, um die mit RL umgesetzte Verhaltensstrategie weiter zu evaluieren.

6.1 Nachbildung der Referenzstrategie aus den Messungen

Da die interne Strategie des Referenzfahrzeugs nicht bekannt ist, wird die in den Messungen ersichtliche Strategie in eine simulationsbasierte Referenzstrategie transformiert. Basierend auf Messungen eines Mittelklassefahrzeugs wurde die Referenzstrategie nachgebildet. Anschließend wird die Leistung des RL-Ansatzes beurteilt. In Abschnitt 6.2 konkurriert der Agent in allen Situationen gegen diese Referenzstrategie. Dazu muss zunächst sichergestellt werden, dass die Umgebungsbedingungen der Simulation mit den Bedingungen der Messung übereinstimmen. Dies betrifft hauptsächlich die Grundleistung, die kurz vor Auftreten eines kritischen Bordnetzzustandes vorhanden ist. Wie in Kapitel 5.3 beschrieben, berechnet sich die Gesamtleistung der Simulation aus den Leistungen der steuerbaren elektrischen Verbrauchern und einer konstanten Grundlast. Die Leistungen der Komfortverbraucher stehen für die Reduktion in kritischen Situationen zur Verfügung. Die Differenz zwischen der Grundleistung und dem Reduktionspotenzial ergibt die Grundlast. Da in seltenen Fällen einige Grundleistungswerte das Reduktionspotenzial unterschreiten, kann die Differenz auch negativ sein. Dadurch wird gewährleistet, dass die zu Beginn eines Manövers gemessene Leistung durchgängig erreicht wird und die Strategien die gleichen Anfangsbedingungen haben. Der Agent wird somit auch mit geringeren Bordnetzlasten konfrontiert, als die mit denen er trainiert wurde. Trainiert wurde kontinuierlich mit einer Grundleistung von ca. 2500 W.
Von der Gesamtbordnetzleistung wird die Leistung der Hochleistungsverbraucher abgezogen. Somit kann die Referenzstrategie abgebildet werden. Während der Fahrzeugmessungen werden signifikante Leistungsänderungen beobachtet. Diese können als Strategie zur Bordnetzstabilisierung interpretiert werden. Dies verdeutlicht die Messung (blau) im oberen Teil von der Abbildung 6.1 anhand eines Ausweichmanövers. Die verbleibende Leistung wird auf die Leistungen der steuerbaren Verbraucher sowie eine konstante Grundlast aufgeteilt. Wird die gemessene Leistung im Bordnetz reduziert, führt dies zu der entsprechenden Degradierung gemäß Kapitel 5.3. Diese übersetzte Degradierungslevel werden im unteren Teil der Abbildung 6.1 dargestellt. Der obere Teil von Abbildung 6.1 zeigt die Leistung als Ergebnis der Degradierung in roter Farbe. Für alle Szenarien wird automatisch mit Matlab die Übersetzung der Bordnetzleistung ohne Hochleistungsverbraucher (blau) durchgeführt. Das Verhalten des realen Fahrzeugs kann simuliert werden, auch wenn die

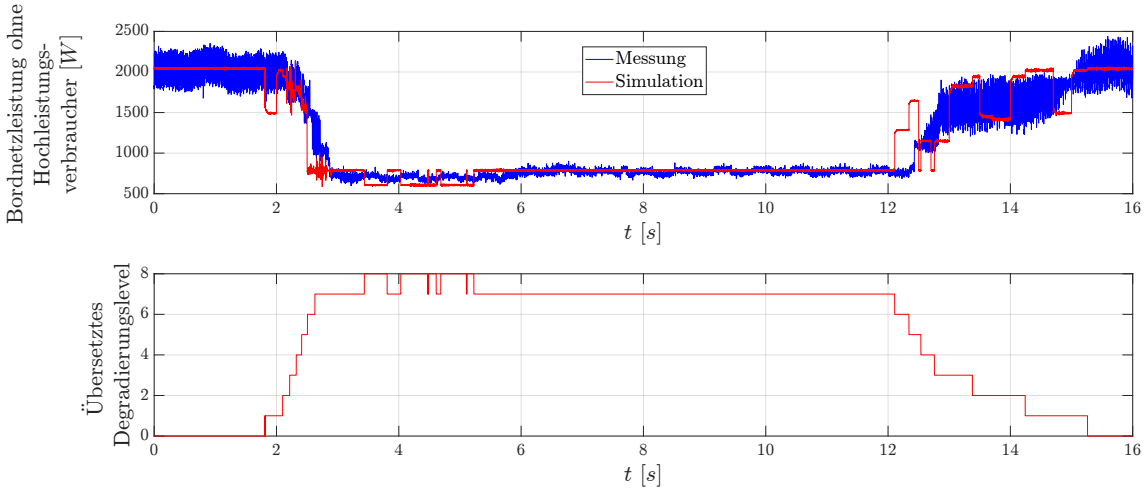


Abbildung 6.1: Übersetzen der gemessenen Bordnetzleistung ohne Hochleistungsverbraucher in eine Referenzstrategie, die ein Degradierungslevel vorgibt. Beispielhafter Ausschnitt aus einem Ausweichmanöver.

genauen Hintergründe der gemessenen Strategie und die Anzahl der Stufen unbekannt sind. Nach diesem Vorgang spielt die Reihenfolge der Degradierung im Testfahrzeug keine Rolle.

Gemäß der Messung in Abbildung 6.1 beträgt die durchschnittliche Grundleistung ungefähr 2000 W. Im Bereich von $0\ s < t < 1\ s$ ohne Degradation lässt sich dies durch die rot markierte Simulation demonstrieren. Die Degradation beginnt mit dem Auftreten des Spannungsabfalls bei $t \approx 2\ s$. Die im oberen Teil von Abbildung 6.1 gemessene Leistung nimmt deutlich auf etwa 750 W ab. Dies ist zweifelsfrei keine Handlung der Fahrzeuginsassen. Während der Testfahrt befand sich nur der Fahrer im Fahrzeug der, während der Manöver nur mit der Steuerung des Fahrzeugs beschäftigt war. Die Abbildung zeigt, dass die Strategie erfolgreich repliziert werden kann. Insbesondere im Intervall von $3\ s < t < 12,5\ s$ wird dies ersichtlich. Insgesamt stellt dies eine Vergleichbarkeit mit dem RL-Ansatz für das folgende Kapitel 6.2.

6.2 Vergleich der Referenzstrategie mit dem RL-Ansatz

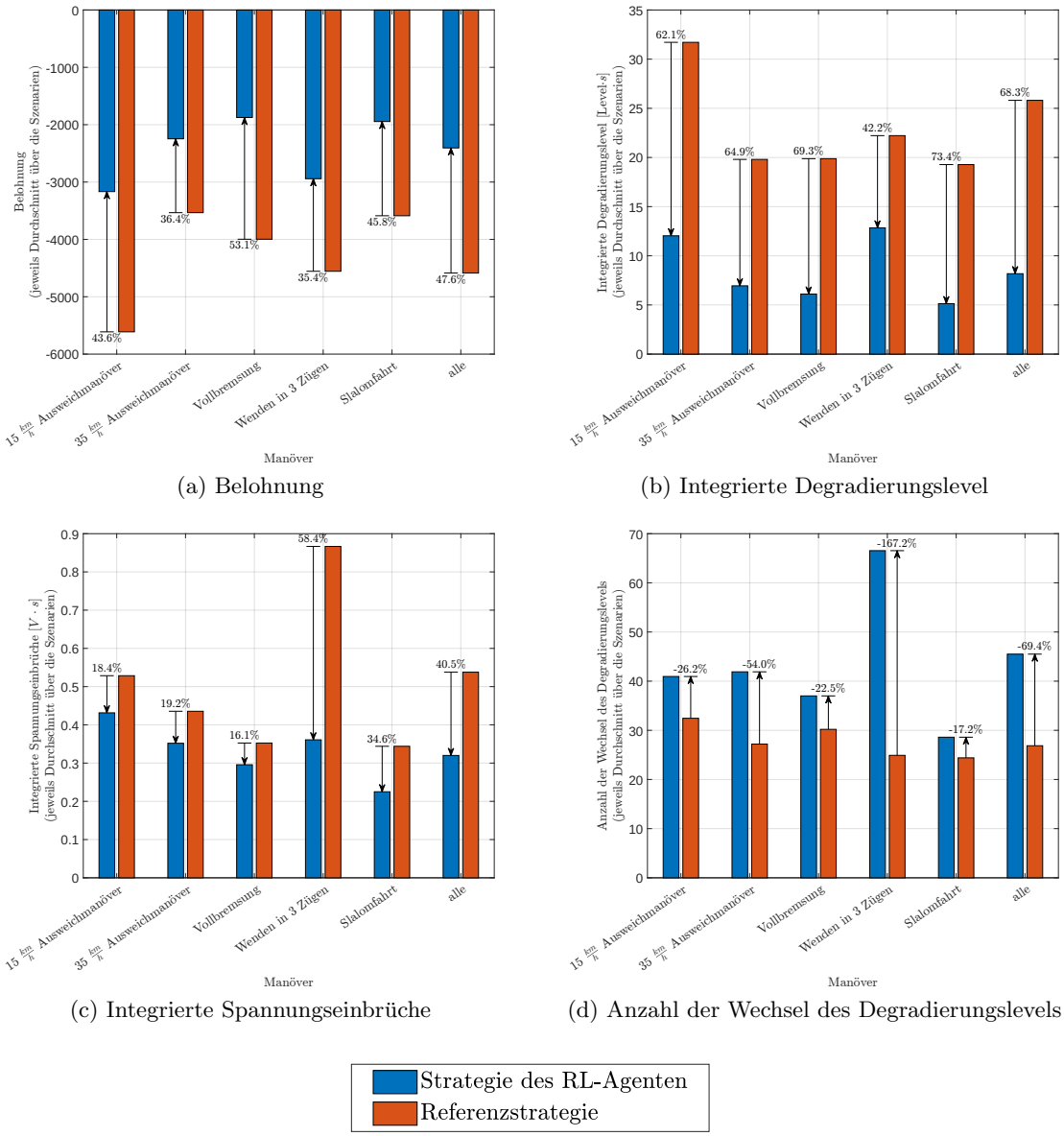
In diesem Abschnitt wird die vorgestellte Referenzstrategie mit dem leistungsstärksten Agenten verglichen. Deshalb ist es wichtig, vor dem Vergleich klar zu bestimmen, welcher RL-Ansatz für die Aufgabenstellung in dieser Arbeit geeignet ist. Nach dem Vergleich verschiedener Kombinationen von Hyperparametern und Trainingsmethoden in den Kapiteln 5.6, 5.7 und 5.8 ist der RL-Agent mit DDQN am besten für die Aufgabe geeignet. Dieser Agent setzt ein rekurrentes Q-Netz mit LSTM-Zellen ein. Tabelle 6.1 zeigt die Trainingsdetails des Agenten.

Tabelle 6.1: Übersicht der Trainingsdetails des für am besten befundenen RL-Ansatzes aus Kapitel 5.8.

Parameter	**Beschreibung**
Lernrate α	10^{-4}
Größe des Minibatch M	64
Diskontierungsfaktor γ	$0,99$
ε-greedy Strategie	$\varepsilon = 1$; Verfall von 10^{-4}; Grenze $\varepsilon = 0,1$
Größe des Erfahrungspuffers D	10^6
Aufbau Q-Netz	rekurrentes neuronales Netz; 3 versteckte Schichten mit je 50 Neuronen/Zellen; mittlere Schicht mit LSTMs; Sequenzlänge von 50 diskreten Zeitschritten
Anzahl trainierter Episoden	200
Trainingsmethode	zufälliges Szenario aus allen Verfügbaren in zeitdiskreter Simulation mit $\Delta t = 0,001\ s$ und $T = 20\ s$

Zunächst wurde ein Haltemechanismus für Aktionen des Agenten eingeführt. Dabei war das Ziel eine Verbesserung der Aufgabenstellung zu erreichen. Jedoch wurde aus den Auswertungen deutlich, dass die Leistung des Agenten unter diesem Mechanismus leidet. Außerdem ist der Haltemechanismus nicht Teil der Belohnungsfunktion und manipuliert den Agenten von außen. Daher wird dieser Mechanismus in diesem Kapitel nicht mehr berücksichtigt.

Alle Szenarien werden automatisch jeweils mit dem RL-Agenten und der Referenzstrategie simuliert, um die Leistung des Agenten mit der Referenzstrategie zu vergleichen. Die Grundleistung, die der jeweiligen Messung entspricht, wird vor der Simulation in das Modell geladen. Nach der Simulation werden alle signifikanten Werte als Ausgaben in einer Datei mit dem entsprechenden Szenarionamen gespeichert. Auch wenn kein Training stattfindet, wird die errechnete Belohnung mit erfasst. Die Belohnung kann dennoch einen Anhaltspunkt bieten, da sie bereits entscheidende Bewertungskriterien enthält und unabhängig vom RL als Leistungsindikator verwendet werden kann. Zudem werden die Daten in Abhängigkeit des Szenarios ausgewertet. Abbildung 6.2 zeigt diese Auswertung. Die Belohnung, die Flächenintegrale der Degradierungsstufen und Spannungseinbrüche sowie die Anzahl der Stufenwechsel werden wie in den vorangegangenen Kapiteln berechnet. Die Ergebnisse werden in vier Säulendiagrammen (a) bis (d) dargestellt.
Die Abbildung 6.2(a) zeigt deutlich, dass die RL-Methode der Referenzstrategie in Bezug



(a) Belohnung

(b) Integrierte Degradierungslevel

(c) Integrierte Spannungseinbrüche

(d) Anzahl der Wechsel des Degradierungslevels

Abbildung 6.2: Vergleich des besten RL-Ansatzes aus Kapitel 5.8 mit der Referenzstrategie aus Kapitel 6.1. Als Kriterien werden die Belohnung, Integrierte Degradierungslevel, Integrierte Spannungseinbrüche und die Anzahl der Wechsel des Degradierungslevels betrachtet.

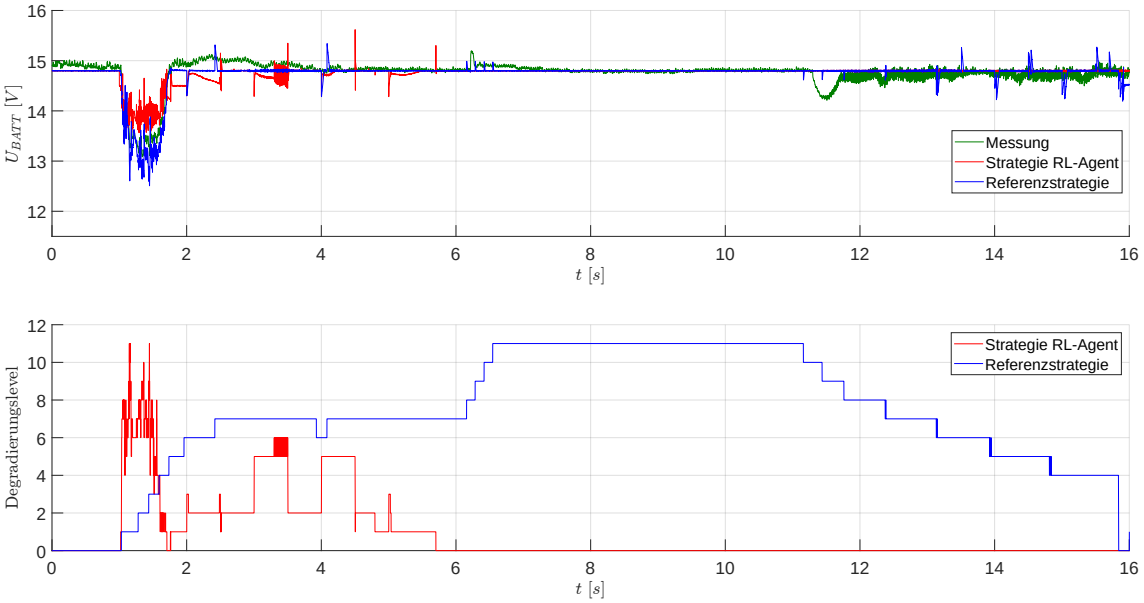


Abbildung 6.3: Vergleich der simulierten Batteriespannung und Degradierung im Ausweichmanöver mit 15 $\frac{km}{h}$ zwischen RL-Ansatz und Referenzstrategie. Zusätzlich ist die gemessene Batteriespannung im oberen Diagramm aufgetragen.

auf die Belohnung über alle Manöver hinweg deutlich überlegen ist. Das muss natürlich so eingeordnet werden, dass nicht klar ist, welche Bewertungskriterien mit welcher Gewichtung bei der Strategieentwicklung des Referenzfahrzeugs eine Rolle gespielt haben. Im Gegensatz dazu wird hier die RL-Methode speziell für die Maximierung dieser Belohnung trainiert. Daher kann für die Abbildung 6.2(a) nur gesagt werden, dass die RL-Methode die Referenzstrategie bei der Erreichung der Ziele dieser Arbeit deutlich übertrifft. Bei der Analyse der Abbildungen 6.2(b) und (c) fällt auf, dass der Spannungsabfall erheblich reduziert werden kann, obwohl die Degradierung in allen Manövertypen geringer ist. Hierbei wird die maximale Reduktion der Spannungseinbrüche um $58,4\,\%$ beim Wenden in 3 Zügen erreicht. Dies ist in Abbildung 6.2(c) zu sehen. Das größte Potenzial zur Verbesserung des Komforts durch Verringerung der Degradierung findet sich in Abbildung 6.2(b) mit $73,4\,\%$ bei der Slalomfahrt. Im Vergleich zur Referenzstrategie wurde die Gesamtzahl der Eingriffe des Agenten über alle Manöver um $68,3\,\%$ reduziert. Die Verbesserung der Spannungseinbrüche und des Degradierungsgrads geht zu Lasten von merklich mehr Änderungen des Degradierungslevels, wie aus Abbildung 6.2(d) abgeleitet werden kann. Beim Vergleich des RL-Ansatzes mit der Referenzstrategie, gibt es insgesamt $69,4\,\%$ mehr Wechsel der Degradierungsstufen. Das Wenden in 3 Zügen bringt mit $167,2\,\%$ die mit Abstand größte Steigerung der Wechsel. Ein Ausweichmanöver mit 15 $\frac{km}{h}$ dient zur Veranschaulichung, wie sich die beiden Strategien voneinander unterscheiden. Diese ist in Abbildung 6.3 zu sehen. Hierbei handelt es sich um ein Szenario, in dem der größte relative Unterschied der Batteriespannung $\Delta U_{BATT} = 1,77\ V$ bei $t = 1,44\ s$ zwischen dem RL-Ansatz (rot) und der Referenzstrategie (blau) zu verzeichnen ist.

Auf den ersten Blick ist deutlich zu erkennen, wie sich die Degradierungsstrategien voneinander unterschieden. Die Referenzstrategie (blau) im unteren Teil von der Abbildung 6.3 degradiert im Intervall $1\ s < t < 15,5\ s$ über einen längeren Zeitraum die elektrischen Verbraucher. Dahingegen reagiert der RL-Agent (rot) sofort zu Beginn des Einbruchs bei $t = 1\ s$ stärker. Dies führt zur vollen Ausschöpfung des Reduktionspotenzials zum Degradierungslevel $\delta = 11$. Die Referenzstrategie setzt die maximale Degradierungsstufe erst bei etwa $t = 6,5\ s$. Um zu demonstrieren, dass über den nächsten Zeitraum bis etwa $t = 11\ s$ kein Spannungseinbruch mehr in der Messung auftritt, ist die gemessene Batteriespannung in der oberen Grafik grün eingezeichnet. Die Messung erfasst ähnlich wie die Simulation nur einen Spannungseinbruch durch die Hochleistungsverbraucher zum Zeitpunkt $t = 1\ s$. Aufgrund des Einschaltens der Verbraucher gibt es bei der Messung einen kleineren Einbruch bei etwas $t = 11,3\ s$. Dies zeigt sich in der nachgebildeten Referenzstrategie. Im Vergleich zum RL-Ansatz ist die Batteriespannung nach dem Einbruch im Intervall $2\ s < t < 6\ s$ stabiler, da die Eingriffe der Referenzstrategie umfangreicher sind. Die Messung zeigt sogar einen Anstieg der Batteriespannung in diesem Zeitraum. Dies lässt sich auf die Anhebung der Generatorsollspannung als Managementstrategie zurückführen. Der RL-Agent degradiert im beschriebenen Bereich deutlich weniger und sorgt so für kleine Einbrüche von einer Tiefe von z.B. $\Delta U_{BATT} = 0,51\ V$ bei $t = 3\ s$. Interessanterweise hat der RL-Agent gelernt, dass unterschiedliche Verbraucher in bestimmten Degradierungslevel getaktet werden. Über einen Zeitraum von $3\ s < t < 5\ s$ zeigt der untere Teil der Abbildung 6.3, dass der RL-Agent die Abstufung während des Einschaltimpulses der Taktung um drei Stufen erhöht und für die restliche Dauer wieder reduziert.

Insgesamt lässt sich feststellen, dass sich die Ansätze grundlegend unterscheiden. Im Vergleich zur Referenzstrategie zeigt der RL-Ansatz über alle Manöver hinweg ein Verbesserungspotenzial in der Stabilität des Bordnetzes. Gleichzeitig werden weniger Komforteinbußen für die Insassen verursacht.

6.3 Verhalten des RL-Agenten bei unterschiedlichen Grundleistungen

Der RL-Agent wurde durchgehend mit der Grundleistung von etwa 2500 W trainiert. In diesem Kapitel wird das Verhalten des Agenten bei unterschiedlichen Bordnetzleistungen evaluiert. Hierzu wird der Agent mit unterschiedlichen Grundleistung $P_{Grund} = [1500\ W; 2000\ W; 2500\ W]$ konfrontiert. Die verschiedenen Bordnetzleistungen P_{BN} sind in Abbildung 6.4 dargestellt. Darin wird ein Ausweichmanöver mit 35 $\frac{km}{h}$ dargestellt, wobei die Hochleistungsverbraucher aktiv sind.

Einige Komfortverbraucher werden nicht mit voller Leistung betrieben, um die Grundleistung zu erreichen. Die folgende Aufzählung gibt einen Überblick, welche Verbraucherstufen (in Klammern) gewählt werden, um die entsprechende Grundleistung zu erreichen:

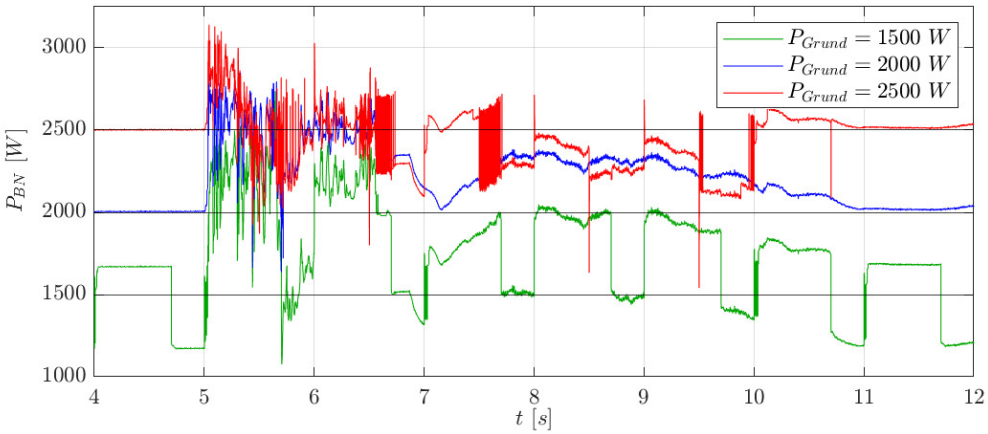


Abbildung 6.4: Bordnetzleistung im Ausweichmanöver mit 35 $\frac{km}{h}$ bei verschiedenen Grundleistungen P_{Grund}.

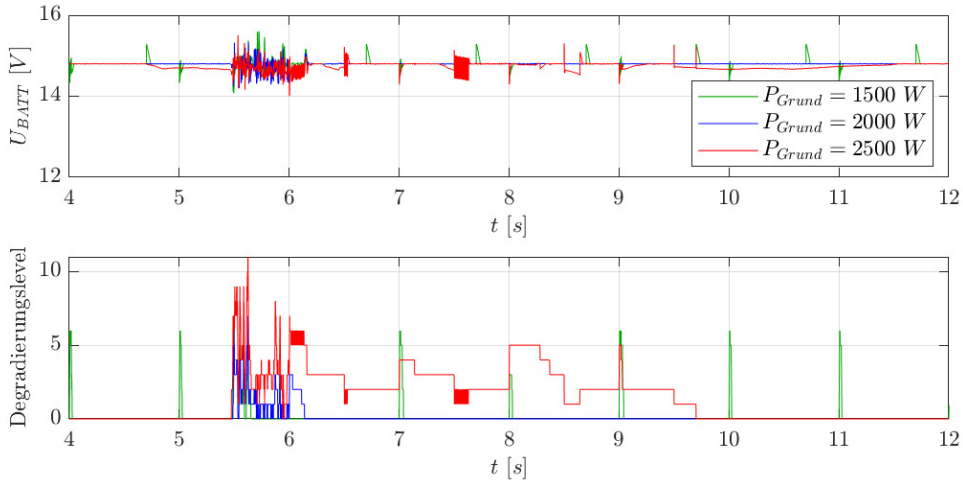


Abbildung 6.5: Vergleich des Verhaltens des RL-Ansatzes im Ausweichmanöver mit 15 $\frac{km}{h}$ bei verschiedenen Grundleistungen P_{Grund}.

- $P_{Grund} = 1500\ W$: Gebläse (3), Sitzheizungen (3), Heizwischblatt (1), Waschdüsenheizung (1), Lenkradheizung (1), Außenspiegelheizung (1) und USB (1), restliche Verbraucher mit voller Leistung,
- $P_{Grund} = 2000\ W$: Gebläse (4), restliche Verbraucher mit voller Leistung,
- $P_{Grund} = 2500\ W$: Alle Verbraucher mit voller Leistung.

Das Verhalten des RL-Ansatzes wird im folgenden Abschnitt anhand eines Szenarios gegenübergestellt, das drei Grundleistungen für jeden Manövertyp umfasst. Ein weiteres Ausweichmanöver bei 15 $\frac{km}{h}$ ist in Abbildung 6.5 dargestellt. Im oberen Diagramm wird die Batteriespannung U_{BATT}, im unteren Diagramm das Degradierungslevel gezeigt.
Der Agent wurde bei einer Grundleistung von etwa 2500 W mit vollständigem Degradierungspotenzial trainiert. Der untere Teil von Abbildung 6.5 zeigt den Effekt der Re-

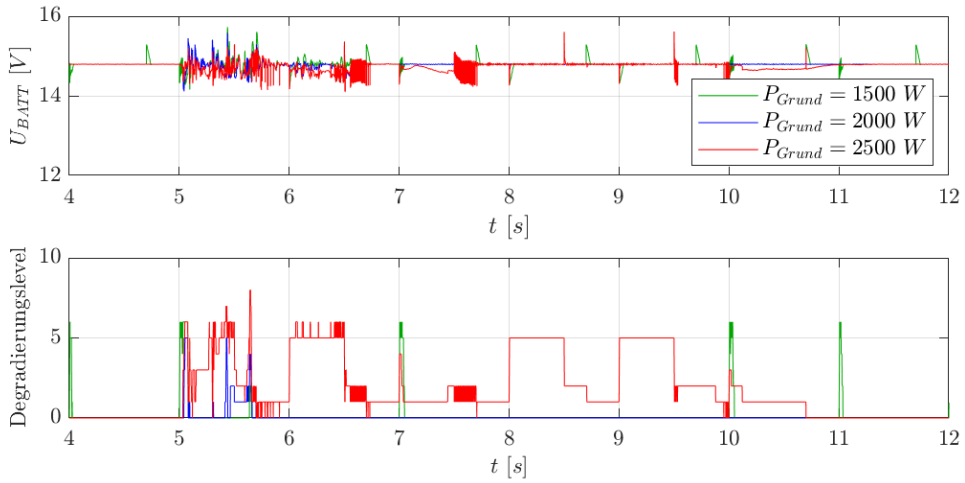


Abbildung 6.6: Vergleich des Verhaltens des RL-Ansatzes in einem Ausweichmanöver mit 35 $\frac{km}{h}$ bei verschiedenen Grundleistungen P_{Grund}.

duzierung des Spannungseinbruchs für diesen Agenten (rot) durch die Degradierung der elektrischen Verbraucher. Bei einer Grundleistung von 2000 W (blau) wird die Belastung des Bordnetzes reduziert, da weniger Verbraucher aktiviert sind. Es ist zu erwarten, dass der Spannungseinbruch ohne Eingriffe weniger stark ausgeprägt wäre, sodass der Agent entsprechend weniger degradieren muss. Diese Vermutung wird durch den unteren Teil der Abbildung 6.5 bestätigt. Bei der niedrigsten getesteten Grundleistung von 1500 W (grün) weist das Modell ein Defizit auf. Die sekündlich wiederkehrenden Spitzen bis zu Degradierungslevel $\delta = 6$ im unteren Diagramm sind eine Reaktion auf die Spannungseinbrüche, die beim Einschalten durch die Taktung einiger Verbraucher entstehen. Während der Messung konnten keine taktungsbedingten Spannungseinbrüche festgestellt werden. Es wird daher davon ausgegangen, dass die genannten Peaks pro Sekunde im realen Fahrzeug mit dem RL-Ansatz nicht auftreten.

Das in Abbildung 6.6 dargestellte Ausweichmanöver bei 35 $\frac{km}{h}$ zeigt ein vergleichbares Verhalten über alle Grundleistungen hinweg. Das Ausmaß der Eingriffe wird in Abbildung 6.6 analog zum 15 $\frac{km}{h}$ Ausweichmanöver mit der Grundleistung reduziert. Dagegen stehen die Manöver Slalomfahrt, Wenden in drei Zügen und Vollbremsung. Hier sind die Ergebnisse ähnlich wie bei den Ausweichmanövern mit Grundleistungen von 2000 W und 2500 W (blau und rot), jedoch ändert sich das Verhalten des Agenten bei einer Grundleistung von nur 1500 W (grün) deutlich. Dies ist in Abbildung 6.7 grafisch dargestellt. Abbildung 6.7 zeigt, dass der RL-Agent bei einer Grundleistung von 1500 W (grün) fast über die gesamte Dauer des Szenarios die höchste Degradationsstufe auswählt. Eine der Ursachen dafür ist, dass dem Agenten nicht bewusst ist, welche elektrischen Verbraucher eingeschaltet sind. Durch die dauerhaft abgeschalteten Verbraucher haben einige Stufen keine Funktion mehr. Es zeigt sich jedoch, dass sich der Agent im Intervall $10\ s < t < 11\ s$ nicht mehr intelligent verhält, indem er alle Degradationen zum Zeitpunkt der Bordnetzinstabilität zurücknimmt. Bei 1500 W Grundleistung (grün) zeigen die beiden Abbildungen

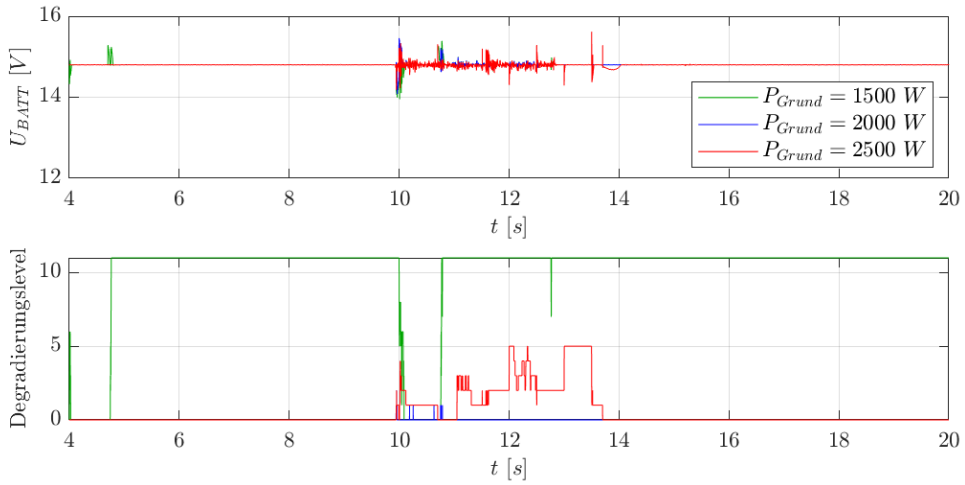


Abbildung 6.7: Vergleich des Verhaltens des RL-Ansatzes in dem Manöver Vollbremsung bei verschiedenen Grundleistungen P_{Grund}.

6.8 und 6.9 ein ähnliches Verhalten. Die Maßnahmen des Agenten erscheinen nicht mehr sinnvoll, da das Degradieren zu Zeiten erfolgt, in denen keine Instabilität im Bordnetz des Fahrzeugs vorliegt.
Bei einer Grundleistung von 1500 W treten jedoch keine Bordnetzinstabilitäten aufgrund der geringen Bordnetzauslastung mehr auf. Dies wird in den Abbildungen 6.5 bis 6.9 ersichtlich. Die Spannungseinbrüche bei der grün markierten Grundleistung gehen vollständig auf das Zuschalten von Verbrauchern zurück. In diesem niedrigen Leistungsbereich ist ein Eingriff nach dem RL-Ansatz nicht mehr erforderlich. Wenn die Auslastung des Bordnetzes gering ist, kann der Agent entweder deaktiviert oder auf eben jene Situationen trainiert werden.

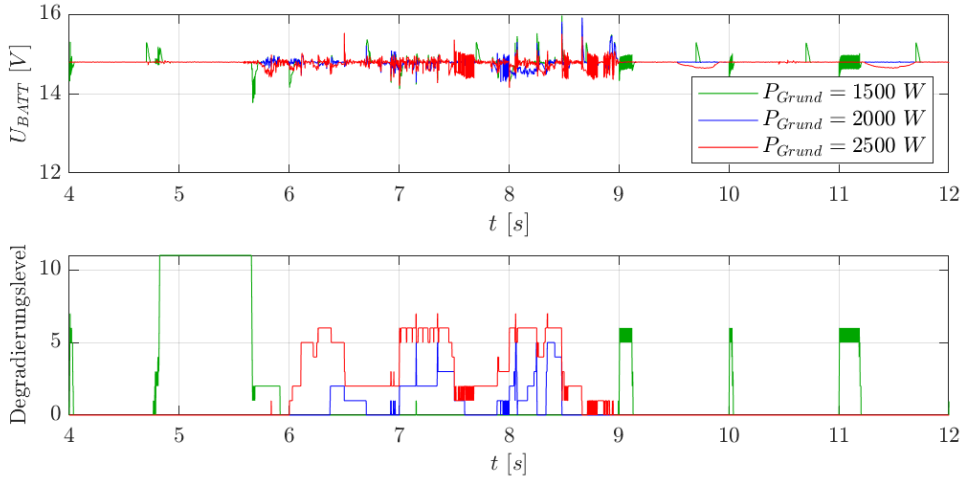


Abbildung 6.8: Vergleich des Verhaltens des RL-Ansatzes im Manöver Wenden in 3 Zügen bei verschiedenen Grundleistungen P_{Grund}.

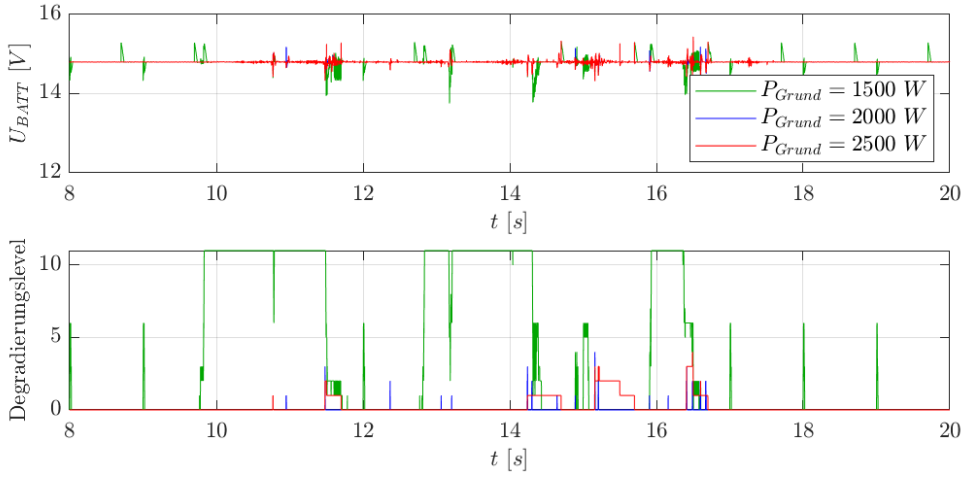


Abbildung 6.9: Vergleich des Verhaltens des RL-Ansatzes im Manöver Slalomfahrt bei verschiedenen Grundleistungen P_{Grund}.

7 Zusammenfassung und Ausblick

Die Zahl der elektrischen Verbraucher im 12 V-Bordnetz nimmt seit Jahrzehnten stetig zu. Dies resultiert sowohl aus der Elektrifizierung von Verbrauchern als auch aus den deutlichen technologischen Weiterentwicklungen der letzten Jahre in den Bereichen Komfort, Entertainment und Assistenzsysteme. Die steigende Komplexität der Bordnetze stellt die Fahrzeughersteller vor wachsende Herausforderungen bei der Bordnetzstabilität. Leistungsstarke Verbraucher wie die elektrische Lenkung oder Bremsen können die Spannung im Bordnetz in Extremsituationen deutlich senken. Ohne ein geregeltes Leistungsmanagement besteht die Gefahr, dass sich Steuergeräte wegen Unterspannung abschalten. Um dies zu verhindern werden Strategien eingesetzt, die auf Regeln oder mathematischer Optimierung basieren. Die Entwicklung solcher Ansätze bringt eine hohen zeitlichen Aufwand mit sich und ist zudem stark von Erfahrungen und Expertenwissen abhängig.

7.1 Ergebnisse der Arbeit

In der vorliegenden Arbeit wurde ein RL-Ansatz zur Stabilisierung von Spannungseinbrüchen in 12 V-Bordnetzen entwickelt. Nach einer Einleitung und der Beschreibung der Zielsetzung für die vorliegende Arbeit erfolgt im anschließenden Kapitel eine allgemeine Einführung in die Thematik der automobilen Bordnetze. Zudem wurden die Grundlagen von RL und Deep-RL erläutert.

Basierend auf dieser Grundlagen wurden im dritten Kapitel Simulationsmodelle von Generator, Verbrauchern und Batterie erstellt, um die Eigenschaften des realen Systems nachzubilden und das Spannungsverhalten eines Energiebordnetzes in kritischen Belastungsszenarien analysieren zu können. Das Generatormodell besteht aus mehreren Teilmodellen und wird als Kennlinien-basiertes Modell aufgebaut. Bei der simulativen Nachbildung der elektrischen Verbraucher wurden die einzelnen Verbraucher in Strom-basierte, Spannung-basierte oder Leistung-basierte Systeme aufgeteilt. Somit können die Spannungsabhängigkeiten berücksichtigt werden. Komplexere Verbraucher, wie z.B. die Hochleistungsverbraucher wurden mit hinterlegten Profilen modelliert, die aus Messungen an realen Komponenten stammen. Für das Batteriemodell wurde ein Thevenin-Modell zweiten Grades verwendet. Dieses besteht aus einer Spannungsquelle, einem Innenwiderstand und zwei in Reihe geschalteten RC-Gliedern. Mit mehreren Tests am Prüfstand wurden diese einzelnen Parameter bei unterschiedlichen Temperaturen bestimmt.

Für die Untersuchung der Spannungsstabilität wurden kritische Fahrmanöver definiert. Dabei handelt es sich um Worst-Case-Szenarien, die einen hohen Spannungseinbruch im 12 V-Bordnetz verursachen. Die Messungen wurden in einem Erprobungsfahrzeug mit ei-

nem 12 V-Bordnetz aufgezeichnet. Das Fahrzeug ist mit Hochleistungsverbrauchern ausgestattet, die in den kritischen Manövern für einen starken Spannungseinbruch sorgen. Die Rohdaten wurden anschließend für das Training des RL-Agenten vorbereitet. Da innerhalb einer Messung das Manöver mehrmals durchgeführt wurde, werden aus einer Messung mehrere Daten derselben Länge erzeugt. Zudem wurde das Gesamtmodell mit diesen Fahrzeugmessungen validiert.

Nach der Datenvorbereitung wurde die Implementierung des RL-Agenten näher erläutert. Zunächst wurde die Wahl des RL-Algorithmus begründet. Da im Rahmen dieser Arbeit ein Algorithmus zur Steuerung von unterschiedlichen elektrischen Verbrauchern untersucht wurde, musste der Agent für diskrete Aktionsräume geeignet sein. In einem Experiment wurden drei unterschiedliche Agenten analysiert. Als Schlussfolgerung fiel die Wahl des Algorithmus auf DQN. Nachdem die Dimension des Zustands- und Aktionsraums festgelegt wurde, folgte die Formulierung der Belohnungsfunktion. Die Komplexität dieser Funktion wurde in der vorliegenden Arbeit bewusst gering gehalten, um eine kürzere Trainingszeit und schnellere Lernerfolge zu erzielen.
Die Optimierung der Hyperparameter ist entscheidend für die Gesamtleistung von Modellen. Das in dieser Arbeit verwendete DQN hat eine große Anzahl an Hyperparametern. Da in der Literatur in vielen Fällen die Optimierung dieser Parameter nicht dokumentiert wird, wurde in dieser Arbeit ein besonderer Fokus auf das Hyperparametertuning gelegt. Mit mehreren Tests und verschiedenen Auswertungskriterien wurde die Parametrierung begründet. Zudem wurde auf das Thema der RNN eingegangen. Aufgrund der besonderen Fähigkeit von LSTM-Zellen, sequenzielle Daten zu verarbeiten und Beziehungen in Zeitreihen zu identifizieren, wurde untersucht, ob sich die Art der Eingriffe des Agenten ändern. Dieser erhielt bei allen Manövern eine größere Belohnung, erfüllte den gewünschten Kompromiss der Lernaufgabe besser und veränderte die Degradierungslevel in geringerer Anzahl.
Aus der Analyse wurde festgestellt, dass die RL-basierte Strategie sehr viel einfacher skaliert werden kann als ein Regel-basiertes System. Hauptgrund ist hierbei, dass kein detailliertes Systemverständnis für die Implementierung eines RL-Ansatzes verlangt wird. Je komplexer die Bordnetze werden, desto schwieriger wird es für die Entwickler, dieses Systemverständnis aufrecht zu erhalten, um eine Regel-basierte Strategie zu implementieren.

Anschließend wurden die Ergebnisse des optimierten Agenten dargestellt. Dafür wurde zunächst eine Referenzstrategie entwickelt, welche für die Nachbildung des realen Energiemanagementsystems steht. Nachdem der Agent mit der Referenzstrategie verglichen wurde, wurde er gezielt mit unterschiedlichen Bordnetzauslastungen konfrontiert. Dabei wurde ein deutliches Verbesserungspotenzial der Spannungseinbrüche des RL-Agenten gegenüber der Regel-basierten Referenzstrategie über alle kritischen Manöver aufgezeigt. Der signifikante Unterschied, der sich konkret in der Analyse des Verhaltens zwischen RL-Ansatz und Referenzstrategie zeigt, beweist in der frühen Entwicklungsphase, dass der

Agent erfolgreich eine Strategie erlernt hat, die dem bisherigen Vorgehen auch in der realen Welt überlegen sein kann.

Nicht zuletzt ist anzumerken, dass die aktuelle Arbeit das Potenzial von RL anhand einer exklusiven Problemstellung erfolgreich demonstriert hat. Im Rahmen dieser Arbeit wurden alle Teilziele aus Kapitel 1.3 erreicht. Dabei wurde gezeigt, dass KI den Vorteil hat, mit weitaus komplexeren Problemen umgehen zu können, die rein mit logischem Denken oder mathematischen Lösungsansätzen nur mit viel Know-How und unter großen Zeit- und Kostenaufwänden lösbar sind. Um die zunehmende Komplexität des Bordnetzes zukünftig bewältigen zu können, stellt RL einen Lösungsprozess zur Verfügung, der im Vergleich zu traditionellen Entwicklungsmethoden einfach skalierbar und anpassbar ist.

7.2 Ausblick

Für zukünftige Arbeiten könnte die Anzahl von Wechseln im Degradierungslevel in die Belohnungsfunktion aufgenommen werden. Insbesondere kann jede Änderung des Degradationsgrads mit einem kleinen konstanten Offset bestraft werden. Vor allem in Kombination mit einem rekurrenten Q-Netz wird durch die Erfassung des zeitlichen Zusammenhangs der Zustände ein erfolgreiches Lernen erwartet. Alternativ bietet Imitation Learning die Möglichkeit, das Halten eines Degradierungslevels in den Lernprozess einzubauen. Anders als der in Kapitel 5.4 erwähnte Mechanismus, würde der Agent dann auf Basis der vorgegebenen Aktionen lernen. Das Integrieren der Generatorlast in die Belohnungsfunktion ist eine weitere Strategie zum Verbessern der Verhaltensstrategie des RL-Agenten. Die Wahrscheinlichkeit, dass tiefe Spannungseinbrüche vermieden werden können, steigt, wenn die Last niedriger gehalten wird. Gleichzeitig soll aber der Komfort, den die Insassen des Fahrzeugs wünschen, möglichst uneingeschränkt sein. Dies kann zur Bildung einer vorbeugenden Maßnahme führen, die in der Lage ist, Spannungseinbrüchen schon weit vor der Entstehung vorzubeugen.

Die hier beschriebene Arbeit trägt zur Weiterentwicklung des Leistungsmanagements in 12 V-Bordnetzen bei. Jedoch berücksichtigt ein ganzheitliches Energiemanagementsystem auch andere Faktoren wie Ladebilanz und Effizienz. Um auch diese Punkte abzudecken und die Vorteile von RL zu nutzen, wird für die weitere Forschung der Einsatz mehrerer Agenten im Zusammenspiel, sogenanntes Multi Agent Reinforcement Learning, empfohlen.

Im Bereich Deep RL könnte zudem das sogenannte Transfer Learning untersucht werden. Hierbei werden die Gewichte eines vortrainierten Netzes eingangsseitig festgesetzt. Dahingegen werden die letzten Schichten bis zum Ausgang neu trainiert. Der Zweck besteht darin, den Vorteil der hierarchischen Merkmalsextraktion neuronaler Netze auszunutzen. Auf die vorliegende Lernaufgabe bezogen, bleiben die Beobachtungen im Bordnetz für den Agenten unabhängig vom Fahrzeugtyp gleich. Es sind jedoch verschiedene Fahrzeug- oder

Ausrüstungsvariationen vorstellbar, die den Bewegungsbereich des Agenten verändern könnten. Die Eignung des Transfer Learnings für diese besonderen Anpassungen muss geprüft werden. Es wäre lediglich ein Nachtrainieren der letzten Schichten des neuronalen Netzes und eine Anpassung der Ausgabeschicht erforderlich. Dies könnte zu einer noch größeren Zeit- und Kostenersparnis im Vergleich zu bisherigen regelbasierten Strategien führen.

Literatur

[1] Kurt Kruppok, Benedict Jäger und Reiner Kriesten. „Auswirkungen der Elektrifizierung von Nebenverbrauchern auf das Energiemangement im Kraftfahrzeug“. In: *Forschung aktuell* 2016 (2016), S. 47–49.

[2] Kai Borgeest. *Elektronik in der Fahrzeugtechnik*. Wiesbaden: Springer Fachmedien Wiesbaden, 2021. ISBN: 978-3-658-23663-2. DOI: 10.1007/978-3-658-23664-9.

[3] Gerd Stegmaier. *Warum E-Autos eine 12-Volt-Batterie brauchen*. 2019. URL: https://www.auto-motor-und-sport.de/tech-zukunft/elektroauto-12-volt-batterie/ (besucht am 15. Aug. 2022).

[4] Tom P. Kohler. *Prädiktives Leistungsmanagement in Fahrzeugbordnetzen*. Wiesbaden: Springer Fachmedien Wiesbaden, 2014. ISBN: 978-3-658-05011-5. DOI: 10.1007/978-3-658-05012-2.

[5] Stephan Lange und Michell Schimanski. „Energiemanagement in Fahrzeugen mit alternativen Antrieben“. Doctoral Thesis. Braunschweig: Technische Universität Carolo-Wilhelmina zu Braunschweig, 1.01.2007.

[6] Andreas Heimrath, Joachim Froeschl, Razieh Rezaei, Martin Lamprecht und Uwe Baumgarten. „Reflex-Augmented Reinforcement Learning for Operating Strategies in Automotive Electrical Energy Management“. In: *2019 International Conference on Computing, Electronics & Communications Engineering (iCCECE)*. 2019, S. 62–67. DOI: 10.1109/iCCECE46942.2019.8941819.

[7] Qin Feiyan und Li Weimin. „A Review of Machine Learning on Energy Management Strategy for Hybrid Electric Vehicles“. In: *2021 6th Asia Conference on Power and Electrical Engineering (ACPEE)*. 2021, S. 315–319. DOI: 10.1109/ACPEE51499.2021.9437082.

[8] Hamid Khayyam, Abbas Z. Kouzani, Khashayar Khoshmanesh und Eric J. Hu. „A Rule Based Intelligent Energy Management System for an Internal Combustion Engine Vehicle“. In: *TENCON* (2008), S. 1–5. DOI: 10.1109/TENCON.2008.4766637.

[9] Michael Harald Winter. „Zur Optimierung und Validierung von Managementsystemen für elektrische Energiebordnetze im Kraftfahrzeug“. Dissertation. München: Technische Universität München, 1.01.2019.

[10] Joachim Fröschl. *Kybernetisches Energiemanagement elektrischer Energiewandlung in Kraftfahrzeugen*. 1. Auflage. Elektrotechnik. München: Verlag Dr. Hut, 2020. ISBN: 978-3-8439-4526-4.

[11] Janis Lehmann, Benjamin Löwer, Björn Mohrmann und Rainer Knorr. „Prädiktives Leistungsmanagement für künftige Fahrzeugbordnetze“. In: *ATZ Extra* 24 (2019), S. 48–52. DOI: 10.1007/s35778-019-0019-1.

[12] Atriya Biswas, Pier Giuseppe Anselma und Ali Emadi. „Real-Time Optimal Energy Management of Multimode Hybrid Electric Powertrain With Online Trainable Asynchronous Advantage Actor–Critic Algorithm“. In: *IEEE Transactions on Transportation Electrification* 8.2 (2022), S. 2676–2694. DOI: 10.1109/TTE.2021.3138330.

[13] Ruoyan Han, Renzong Lian, Hongwen He und Xuefeng Han. „Continuous Reinforcement Learning Based Energy Management Strategy for Hybrid Electric Tracked Vehicles“. In: *IEEE Journal of Emerging and Selected Topics in Power Electronics* (2021), S. 1. ISSN: 2168-6777. DOI: 10.1109/JESTPE.2021.3135059.

[14] Bo Hu und Jiaxi Li. „A Deployment-Efficient Energy Management Strategy for Connected Hybrid Electric Vehicle Based on Offline Reinforcement Learning“. In: *IEEE Transactions on Industrial Electronics* 69.9 (2022), S. 9644–9654. ISSN: 0278-0046. DOI: 10.1109/TIE.2021.3116581.

[15] Renzong Lian, Huachun Tan, Jiankun Peng, Qin Li und Yuankai Wu. „Cross-Type Transfer for Deep Reinforcement Learning Based Hybrid Electric Vehicle Energy Management“. In: *IEEE Transactions on Vehicular Technology* 69.8 (2020), S. 8367–8380. ISSN: 0018-9545. DOI: 10.1109/TVT.2020.2999263.

[16] Guodong Du, Yuan Zou, Xudong Zhang, Lingxiong Guo und Ningyuan Guo. „Heuristic Energy Management Strategy of Hybrid Electric Vehicle Based on Deep Reinforcement Learning With Accelerated Gradient Optimization“. In: *IEEE Transactions on Transportation Electrification* 7.4 (2021), S. 2194–2208. ISSN: 2332-7782. DOI: 10.1109/TTE.2021.3088853.

[17] Woong Lee, Haeseong Jeoung, Dohyun Park, Tacksu Kim, Heeyun Lee und Namwook Kim. „A Real-Time Intelligent Energy Management Strategy for Hybrid Electric Vehicles Using Reinforcement Learning“. In: *IEEE Access* 9 (2021), S. 72759–72768. DOI: 10.1109/ACCESS.2021.3079903.

[18] Zhaoxuan Zhu, Yuxing Liu und Marcello Canova. „Energy Management of Hybrid Electric Vehicles via Deep Q-Networks“. In: *2020 American Control Conference (ACC)*. 2020, S. 3077–3082. DOI: 10.23919/ACC45564.2020.9147479.

[19] Andreas Anton Heimrath und Uwe Baumgarten. *An approach to a machine learning-based operating strategy in automotive electrical energy management*. 1. Auflage. München: Verlag Dr. Hut, 2021. ISBN: 978-3-8439-4942-2.

[20] Konrad Reif. *Grundlagen Fahrzeug- und Motorentechnik*. Wiesbaden: Springer Fachmedien Wiesbaden, 2017. ISBN: 978-3-658-12635-3. DOI: 10.1007/978-3-658-12636-0.

[21] Gerhard Henneberger. *Elektrische Motorausrüstung: Starter, Generator, Batterie und ihr Zusammenwirken im Kfz-Bordnetz*. Wiesbaden: Vieweg+Teubner Verlag, 1990. ISBN: 3-528-04764-X. DOI: 10.1007/978-3-322-84365-4. URL: http://dx.doi.org/10.1007/978-3-322-84365-4.

[22] Konrad Reif. *Bosch Autoelektrik und Autoelektronik*. Wiesbaden: Vieweg+Teubner, 2011. ISBN: 978-3-8348-1274-2. DOI: 10.1007/978-3-8348-9902-6.

[23] Elmar Frickenstein, Manfred Wier, Marcus Hafkemeyer, Fathi El-Dwaik und Elmar Hockgeier. „Intelligente Generatorregelung“. In: *ATZelektronik* 1.4 (2006), S. 6–15. ISSN: 1862-1791. DOI: 10.1007/BF03223834.

[24] D. J. Perreault und V. Caliskan. „Automotive Power Generation and Control“. In: *IEEE Transactions on Power Electronics* 19.3 (2004), S. 618–630. ISSN: 0885-8993. DOI: 10.1109/TPEL.2004.826432.

[25] Willibald Dörfler und Werner Peschek. *Einführung in die Mathematik für Informatiker*. Völlig neubearb. Ausg. des zweibändigen Werkes "Dörfler, Mathematik für Informatiker". Hanser-Studienbücher. München: Hanser, 1988. ISBN: 3446151125.

[26] Mehrdad Ehsani, Yimin Gao und John M. Miller. „Hybrid Electric Vehicles: Architecture and Motor Drives“. In: *Proceedings of the IEEE* 95.4 (2007), S. 719–728. ISSN: 0018-9219. DOI: 10.1109/JPROC.2007.892492.

[27] C. D. Rakopoulos, E. G. Giakoumis, D. T. Hountalas und D. C. Rakopoulos. „The Effect of Various Dynamic, Thermodynamic and Design Parameters on the Performance of a Turbocharged Diesel Engine Operating under Transient Load Conditions“. In: *SAE 2004 World Congress & Exhibition*. SAE Technical Paper Series. SAE International400 Commonwealth Drive, Warrendale, PA, United States, 2004. DOI: 10.4271/2004-01-0926.

[28] Klaus Rechberger. „Spannunsregler mit Load-Response Funktion für Kurbelwellenstartergeneratoren“. DE 103 13 215 B4 2014.02.27. 25.03.2003.

[29] Stefan Büchner. „Energiemanagement Strategien für elektrische Energiebordnetze in Kraftfahrzeugen“. Dissertation. Dresden: Technische Universität Dresden, 8.12.2008. URL: https://nbn-resolving.org/urn:nbn:de:bsz:14-ds-1228736572957-56492.

[30] Klaus Heuck, Klaus-Dieter Dettmann und Detlef Schulz. *Elektrische Energieversorgung*. Wiesbaden: Vieweg+Teubner, 2010. ISBN: 978-3-8348-0736-6. DOI: 10.1007/978-3-8348-9761-9.

[31] Xi Zhang und Chris Mi. *Vehicle Power Management: Modeling, Control and Optimization*. Power Systems. London: Springer-Verlag London Limited, 2011. ISBN: 085729735X. DOI: 10.1007/978-0-85729-736-5. URL: http://site.ebrary.com/lib/alltitles/docDetail.action?docID=10494185.

[32] Radomir Fabis. „Beitrag zum Energiemanagement in Kfz-Bordnetzen“. Dissertation. Berlin: Technische Universität Berlin, 11.10.2006. DOI: 10.14279/depositonce-1443. URL: http://dx.doi.org/10.14279/depositonce-1443.

[33] Fred Schäfer und Johannes Liebl. „Energiemanagement in Motor und Fahrzeug“. In: *Handbuch Verbrennungsmotor: Grundlagen, Komponenten, Systeme, Perspektiven*. Hrsg. von Richard van Basshuysen und Fred Schäfer. Wiesbaden: Springer Fachmedien Wiesbaden, 2015, S. 1179–1186. ISBN: 978-3-658-04678-1. DOI: 10.1007/978-3-658-04678-1{\textunderscore}31. URL: https://doi.org/10.1007/978-3-658-04678-1_31.

[34] M. Kabisch, M. Heuer, G. Heideck und Z. A. Styczynski. „Energy management of vehicle electrical system with auxiliary power unit“. In: *2009 IEEE Vehicle Power and Propulsion Conference*. 2009, S. 358–363. DOI: 10.1109/VPPC.2009.5289826.

[35] Wolfgang Ruttor. „Leistungsverteilung und Energiemanagement im Blick“. In: *AUTOMOBIL ELEKTRONIK* (2007), S. 21–23.

[36] Benjamin Hesse. „Wechselwirkung von Fahrzeugdynamik und Kfz-Bordnetz unter Berücksichtigung der Fahrzeugbeherrschbarkeit“. Dissertation. Duisburg-Essen: Universität Duisburg-Essen, 2011.

[37] Andreas Ennemoser, Heimo Schreier und Heinz Petutschnig. „Optimierte Betriebsstrategie für Nebenaggregate im LKW“. In: *MTZ-Motortechnische Zeitschrift* 73.3 (2012), S. 220–225.

[38] B. Bäker und S. Büchner. „Kraftfahrzeugelektrik und -elektronik“. Vorlesungsskriptum. Dresden: Technische Universität Dresden, 2005.

[39] Ian Goodfellow, Yoshua Bengio und Aaron Courville. *Deep learning*. Cambridge, Massachusetts und London, England: MIT Press, 2016. ISBN: 978-0262035613. URL: http://www.deeplearningbook.org/.

[40] Martin L. Puterman. *Markov decision processes: Discrete stochastic dynamic programming*. Wiley series in probability and statistics. Hoboken, NJ: Wiley-Interscience, 2005. ISBN: 978-0471619772. DOI: 10.1002/9780470316887. URL: http://site.ebrary.com/lib/alltitles/docDetail.action?docID=10344088.

[41] Alexander Zai und Brandon Brown. *Einstieg in Deep Reinforcement Learning: KI-Agenten mit Python und PyTorch programmieren*. Carl Hanser Verlag, 2020. ISBN: 978-3-446-46608-1.

[42] Richard S. Sutton und Andrew G. Barto. *Reinforcement Learning: An Introduction*. Bd. 2. Adaptive Computation and Machine Learning series. MIT Press, 2018. ISBN: 978-0-262-19398-6.

[43] L. P. Kaelbling, M. L. Littman und A. W. Moore. „Reinforcement Learning: A Survey“. In: *Journal of Artificial Intelligence Research* 4 (1996), S. 237–285. ISSN: 1076-9757. DOI: 10.1613/jair.301.

[44] Arnaldo Pérez Castaño. *Practical Artificial Intelligence: Machine Learning, Bots, and Agent Solutions Using C#*. Bd. 1. Apress Berkeley, CA, 2018. ISBN: 978-1-4842-3357-3. DOI: 10.1007/978-1-4842-3357-3.

[45] Vincent François-Lavet, Peter Henderson, Riashat Islam, Marc G. Bellemare und Joelle Pineau. „An Introduction to Deep Reinforcement Learning“. In: *Foundations and Trends® in Machine Learning* 2011.3-4 (2018), S. 219–354. ISSN: 1935-8237. DOI: 10.1561/2200000071.

[46] Kai Arulkumaran, Marc Peter Deisenroth, Miles Brundage und Anil Anthony Bharath. „A Brief Survey of Deep Reinforcement Learning“. In: *CoRR* abs/1708.05866 (2017).

[47] Marco Wiering und Martijn Otterlo. *Reinforcement Learning: State-of-the-Art*. Berlin, Heidelberg: Springer Berlin, Heidelberg, 2012. ISBN: 978-3-642-27645-3. DOI: 10.1007/978-3-642-27645-3.

[48] Liangqu Long und Xiangming Zeng. *Beginning Deep Learning with TensorFlow: Work with Keras, MNIST Data Sets, and Advanced Neural Networks*. Bd. 1. Apress Berkeley, CA, 2022. ISBN: 978-1-4842-7915-1. DOI: 10.1007/978-1-4842-7915-1.

[49] Pierre Marquis, Odile Papini und Henri Prade. *A Guided Tour of Artificial Intelligence Research: Volume III: Interfaces and Applications of Artificial Intelligence*. Bd. 3. Springer Cham, 2020. ISBN: 978-3-030-06170-8. DOI: 10.1007/978-3-030-06170-8.

[50] Uwe Lorenz. *Reinforcement Learning: Aktuelle Ansätze verstehen - mit Beispielen in Java und Greenfoot*. Bd. 1. Springer Vieweg Berlin, Heidelbergg, 2020. ISBN: 978-3-662-61651-2. DOI: 10.1007/978-3-662-61651-2.

[51] Muddasar Naeem, Syed Tahir Hussain Rizvi und Antonio Coronato. „A Gentle Introduction to Reinforcement Learning and its Application in Different Fields“. In: *IEEE Access* 8 (2020), S. 209320–209344. DOI: 10.1109/ACCESS.2020.3038605.

[52] David Silver. *Lecture 1: Introduction to Reinforcement Learning*. 2015. URL: https://www.davidsilver.uk/wp-content/uploads/2020/03/intro_RL.pdf (besucht am 30. Mai 2022).

[53] Matthias Franz und Oliver Dürr. „Neuronale Netze“. Vorlesungsskript 1. Fakultät für Informatik. Konstanz: Hochschule für Technik, Wirtschaft und Gestaltung, 2021.

[54] Volodymyr Mnih und Koray Kavukcuoglu. „Playing Atari with Deep Reinforcement Learning“. In: *CoRR* 2013 (2013). URL: http://arxiv.org/abs/1312.5602.

[55] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg und Demis Hassabis. „Human-level control through deep reinforcement learning“. In: *Nature* 518.7540 (2015), S. 529–533. DOI: 10.1038/nature14236.

[56] Hado Hasselt, Arthur Arthur Guez und David Silver. „Deep Reinforcement Learnin with Double Q-Learning“. In: *CoRR* (2015). URL: http://arxiv.org/abs/1509.06461 (besucht am 30. Mai 2022).

[57] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford und Oleg Klimov. „Proximal Policy Optimization Algorithms“. In: *CoRR* (2017). URL: http://arxiv.org/abs/1707.06347.

[58] Mariam Kiran und Buse Melis Ozyildirim. „Hyperparameter Tuning for Deep Reinforcement Learning Applications“. In: *CoRR* abs/2201.11182 (2022). URL: https://arxiv.org/abs/2201.11182.

[59] Hussain Alibrahim und Simone A. Ludwig. „Hyperparameter Optimization: Comparing Genetic Algorithm against Grid Search and Bayesian Optimization“. In: *2021 IEEE Congress on Evolutionary Computation (CEC)*. 2021, S. 1551–1559. DOI: 10.1109/CEC45853.2021.9504761.

[60] James Bergstra und Yoshua Bengio. „Random search for hyper-parameter optimization“. In: *Journal of Machine Learning Research* 13 (2012), S. 281–305.

[61] Petro Liashchynskyi und Pavlo Liashchynskyi. „Grid Search, Random Search, Genetic Algorithm: A Big Comparison for NAS“. In: *CoRR* 2019 (). URL: http://arxiv.org/abs/1912.06059.

[62] Jose M. Bernardo. „Bayesian Statistics“. In: *Encyclopedia of Life Support Systems (EOLSS)* Probability and Statistics. UNESCO, Oxford, UK, 2003 (2003).

[63] Jan Lunze. *Automatisierungstechnik: Methoden für die Überwachung und Steuerung kontinuierlicher und ereignisdiskreter Systeme*. 2., überarb. Aufl. München: Oldenbourg, 2008. ISBN: 978-3486580617. URL: http://dx.doi.org/10.1524/9783486595116.

[64] Angelika Bosl. *Einführung in MATLAB/Simulink: Berechnung, Programmierung, Simulation*. 3., vollständig überarbeitete Auflage. München: Hanser, Carl, 2020. ISBN: 978-3-446-46403-2.

[65] Klaus Heuck, Klaus-Dieter Dettmann und Detlef Schulz. *Elektrische Energieversorgung: Erzeugung, Übertragung und Verteilung elektrischer Energie für Studium und Praxis*. 9., aktualisierte u. korr. Aufl. 2013. Wiesbaden: Springer Fachmedien Wiesbaden und Imprint: Springer Vieweg, 2013. ISBN: 978-3-8348-2174-4.

[66] Horst Bauer, Anton Beer, Karl-Heinz Dietsche, Jürgen Crepin und Folkhart Dinkler. *Autoelektrik Autoelektronik*. Wiesbaden: Vieweg+Teubner Verlag, 1998. ISBN: 978-3-322-91537-5. DOI: 10.1007/978-3-322-91536-8.

[67] Henning Wallentowitz und Konrad Reif. *Handbuch Kraftfahrzeugelektronik*. Wiesbaden: Vieweg, 2006. ISBN: 978-3-528-03971-4. DOI: 10.1007/978-3-8348-9121-1.

[68] Matthias Schöllmann, Hrsg. *Innovative Ansätze für modernes Energiemanagement und zuverlässige Bordnetzarchitekturen ; [Beiträge der Tagung "Energiemanagement & Bordnetze" vom 7. und 8. Mai 2007 im Haus der Technik e.V. in Essen] ; mit 13 Tabellen*. Bd. 71. Fachbuch / Haus der Technik. Renningen: expert-Verl., 2007. ISBN: 978-3816926498. URL: http://deposit.d-nb.de/cgi-bin/dokserv?id=2858035&prov=M&dok_var=1&dok_ext=htm.

[69] Rainer Gehring. *Beitrag zur Untersuchung und Erhöhung der Spannungsstabilität des elektrischen Energiebordnetzes im Kraftfahrzeug: Zugl.: München, Techn. Univ., Diss., 2012*. 1. Aufl. Elektrotechnik. München: Dr. Hut, 2013. ISBN: 978-3-8439-0874-0.

[70] Tom P., Rainer Gehring, Joachim Froeschl, Dominik Buecherl und Hans-Georg Herzog. „Voltage Stability Analysis of Automotive Power Nets based on Modeling and Experimental Results". In: *New Trends and Developments in Automotive System Engineering*. Hrsg. von Marcello Chiaberge. InTech, 2011. ISBN: 978-953-307-517-4. DOI: 10.5772/13127.

[71] Hongtao Mu, Li Geng und Jun Liu. „A High Precision Constant Current Source Applied in LED Driver". In: *2011 Symposium on Photonics and Optoelectronics (SOPO)* (2011), S. 1–4.

[72] Torsten Hauck und Anton Kolbeck. „Bond wire design for eXtreme Switch devices". In: *2010 11th International Thermal, Mechanical & Multi-Physics Simulation, and Experiments in Microelectronics and Microsystems (EuroSimE)* (2010), S. 1–4.

[73] Dan S. Mihai. „Fuzzy control for temperature of the driver seat in a car". In: *2012 International Conference on Applied and Theoretical Electricity (ICATE)* (2012), S. 1–8.

[74] Wolfgang Mathis und Albrecht Reibiger. *Küpfmüller Theoretische Elektrotechnik: Elektromagnetische Felder, Schaltungen und elektronische Bauelemente*. 20., aktualisierte Auflage. Berlin und Heidelberg: Springer Vieweg, 2017. ISBN: 978-3-662-54836-3. DOI: 10.1007/978-3-662-54837-0. URL: http://dx.doi.org/10.1007/978-3-662-54837-0.

[75] Peter Keil und Andreas Jossen. „Aufbau und Parametrierung von Batteriemodellen". Diss. München: Technische Universität München. URL: https://mediatum.ub.tum.de/doc/1162416/.

[76] Lennart Ljung. *System identification: Theory for the user*. 2. ed., 14. printing. Prentice Hall information and system sciences series. Upper Saddle River, NJ: Prentice Hall PTR, 2012. ISBN: 0-13-656695-2.

[77] Dominik Schledde. „Modellbasierte Identifikation von physikalischen Parametern zur Bestimmung der Veränderung charakteristischer Eigenschaften einer C/NMC Lithium-Ionen-Zelle durch Alterungsmechanismen zur Anwendung in Batteriema-

nagementsystemen“. Dissertation. Kassel: Universität und Kassel University Press GmbH, 2017. URL: http://nbn-resolving.de/urn:nbn:de:0002-405213.

[78] Dierk Schröder und Martin Buss. *Intelligente Verfahren.* Berlin, Heidelberg: Springer Berlin Heidelberg, 2017. ISBN: 978-3-662-55326-8. DOI: 10.1007/978-3-662-55327-5.

[79] Jan Philipp Schmidt. „Verfahren zur Charakterisierung und Modellierung von Lithium Ionen Zellen“. Dissertation. 2013. URL: http://digbib.ubka.uni-karlsruhe.de/volltexte/1000036622.

[80] Wilhelm Peukert. „Über die Abhängigkeit der Kapazität von der Entladestromstärke bei Bleiakkumulatoren“. In: *Elektrotechnische Zeitschrift* 18 (1897), S. 287–288.

[81] Antonio Affanni, Alberto Bellini, Carlo Concari, Giovanni Franceschini, Emilio Lorenzani und Carla Tassoni. *EV battery state of charge: neural network based estimation.* Madison, WI, USA, 1-04 June 2003.

[82] Daniel Jerouschek, Ömer Tan, Ralph Kennel und Ahmet Taskiran. „Data Preparation and Training Methodology for Modeling Lithium-Ion Batteries Using a Long Short-Term Memory Neural Network for Mild-Hybrid Vehicle Applications“. In: *Applied Sciences* 10.21 (2020), S. 7880. ISSN: 2076-3417. DOI: 10.3390/app10217880.

[83] Daniel Jerouschek, Ömer Tan, Ralph Kennel und Ahmet Taskiran. „Modeling Lithium-Ion Batteries Using Machine Learning Algorithms for Mild-Hybrid Vehicle Applications“. In: *2021 International Conference on Smart Energy Systems and Technologies (SEST).* 2021, S. 1–6. DOI: 10.1109/SEST50973.2021.9543225.

[84] Jinchun Peng, Yaobin Chen und Russell Eberhart. *Battery pack state of charge estimator design using computational intelligence approaches.* Long Beach, CA, USA, 11-14 January 2000.

[85] Matthew Ragsdale, Job Brunet und Babak Fahimi. *A novel battery identification method based on pattern recognition.* Harbin, China, 3-05 September 2008.

[86] Yanqing Shen. „Adaptive online state-of-charge determination based on neuro-controller and neural network“. In: *Energy Conversion and Management* 51.5 (2010), S. 1093–1098. ISSN: 01968904. DOI: 10.1016/j.enconman.2009.12.015.

[87] Pritpal Singh, Ramana Vinjamuri, Xiquan Wang und David Reisner. „Design and implementation of a fuzzy logic-based state-of-charge meter for Li-ion batteries used in portable defibrillators“. In: *Journal of Power Sources* 162.2 (2006), S. 829–836. ISSN: 03787753. DOI: 10.1016/j.jpowsour.2005.04.039.

[88] Alexandros Nikolian, Joris de Hoog, Karel Fleuerbay, Jean-Marc Timmermans, Omar Noshin, Peter van de Bossche und Joeri van Mielo. „Classification of Electric modelling and Characterization methods of Lithium-ion Batteries for Vehicle Applications“. In: *European Electric Vehicle Congress* (2015).

[89] *2019 International Conference on Computing, Electronics & Communications Engineering (iCCECE)*. 2019.

[90] Uwe Tröltzsch, Olfa Kanoun und Hans-Rolf Tränkler. „Characterizing aging effects of lithium ion batteries by impedance spectroscopy“. In: *Electrochimica Acta* 51.8-9 (2006), S. 1664–1672. ISSN: 00134686. DOI: 10.1016/j.electacta.2005.02.148.

[91] P. L. Moss, G. Au, E. J. Plichta und J. P. Zheng. „An Electrical Circuit for Modeling the Dynamic Response of Li-Ion Polymer Batteries“. In: *Journal of The Electrochemical Society* 155.12 (2008), A986. ISSN: 0013-4651. DOI: 10.1149/1.2999375. URL: http://dx.doi.org/10.1149/1.2999375.

[92] Shalini Rodrigues, N. Munichandraiah und A. K. Shukla. „AC impedance and state-of-charge analysis of a sealed lithium-ion rechargeable battery“. In: *Journal of Solid State Electrochemistry* 3.7-8 (1999), S. 397–405. ISSN: 1432-8488. DOI: 10.1007/s100080050173.

[93] D. Andre, M. Meiler, K. Steiner, H. Walz, T. Soczka-Guth und D. U. Sauer. „Characterization of high-power lithium-ion batteries by electrochemical impedance spectroscopy. II: Modelling“. In: *Selected papers presented at the 12th Ulm ElectroChemical Talks (UECT):2015 Technologies on Batteries and Fuel Cells* 196.12 (2011), S. 5349–5356. ISSN: 0378-7753. DOI: 10.1016/j.jpowsour.2010.07.071.

[94] K. Takeno. „Quick testing of batteries in lithium-ion battery packs with impedance-measuring technology“. In: *Journal of Power Sources* 128.1 (2004), S. 67–75. ISSN: 03787753. DOI: 10.1016/j.jpowsour.2003.09.045.

[95] A. Cuadras und O. Kanoun, Hrsg. *SoC Li-ion battery monitoring with impedance spectroscopy: 2009 6th International Multi-Conference on Systems, Signals and Devices*. 2009. DOI: 10.1109/SSD.2009.4956761.

[96] Xiaosong Hu, Shengbo Li und Huei Peng. „A comparative study of equivalent circuit models for Li-ion batteries“. In: *Journal of Power Sources* 198 (2012), S. 359–367. ISSN: 03787753. DOI: 10.1016/j.jpowsour.2011.10.013. URL: https://www.sciencedirect.com/science/article/pii/S0378775311019628.

[97] Y. Hu, S. Yurkovich, Y. Guezennec und B. J. Yurkovich. „A technique for dynamic battery model identification in automotive applications using linear parameter varying structures“. In: *Control Engineering Practice* 17.10 (2009), S. 1190–1201. ISSN: 0967-0661. DOI: 10.1016/j.conengprac.2009.05.002. URL: https://www.sciencedirect.com/science/article/pii/S0967066109001075.

[98] Il-Song Kim. „The novel state of charge estimation method for lithium battery using sliding mode observer“. In: *Special issue including selected papers presented at the Second International Conference on Polymer Batteries and Fuel Cells together with regular papers* 163.1 (2006), S. 584–590. ISSN: 0378-7753. DOI: 10.1016/j.jpowsour.2006.09.006. URL: https://www.sciencedirect.com/science/article/pii/S0378775306018349.

[99] Seongjun Lee, Jonghoon Kim, Jaemoon Lee und B. H. Cho. „State-of-charge and capacity estimation of lithium-ion battery using a new open-circuit voltage versus state-of-charge“. In: *Journal of Power Sources* 185.2 (2008), S. 1367–1373. ISSN: 03787753. DOI: 10.1016/j.jpowsour.2008.08.103. URL: https://www.sciencedirect.com/science/article/pii/S0378775308017965.

[100] M. McIntyre, T. Burg, D. Dawson und B. Xian, Hrsg. *Adaptive state of charge (SOC) estimator for a battery: 2006 American Control Conference*. 2006. ISBN: 2378-5861. DOI: 10.1109/ACC.2006.1657640.

[101] Margot Ruschitzka, Harry Ott und Rene Degen. „Batteriemodellierung“. In: *Mechatronische Produktentwicklung im Kontext der Mikromobilität*. Hrsg. von Margot Ruschitzka, Harry Ott und Rene Degen. Berlin, Heidelberg: Springer Berlin Heidelberg, 2021, S. 205–249. ISBN: 978-3-662-64622-9. DOI: 10.1007/978-3-662-64623-6{\textunderscore}6.

[102] Hongwen He, Rui Xiong und Jinxin Fan. „Evaluation of Lithium-Ion Battery Equivalent Circuit Models for State of Charge Estimation by an Experimental Approach“. In: *Energies* 4.4 (2011), S. 582–598. ISSN: 1996-1073. DOI: 10.3390/en4040582.

[103] J. G. Thevenin und R. H. Muller. „Impedance of Lithium Electrodes in a Propylene Carbonate Electrolyte“. In: *Journal of The Electrochemical Society* 134.2 (1987), S. 273–280. ISSN: 0013-4651. DOI: 10.1149/1.2100445.

[104] Hannes Hopp. „Thermomanagement von Hochleistungsfahrzeug-Traktionsbatterien anhand gekoppelter Simulationsmodelle“. Dissertation. 2015.

[105] Min Chen und G. A. Rincon-Mora. „Accurate electrical battery model capable of predicting runtime and I-V performance“. In: *IEEE Transactions on Energy Conversion* 21.2 (2006), S. 504–511. ISSN: 0885-8969. DOI: 10.1109/TEC.2006.874229.

[106] S. Badwal und N. Nardella. „Polarization studies on solid electrolyte cells with a full automated galvanostatic current interruption technique“. In: *Solid State Ionics* 40-41 (1990), S. 878–881. ISSN: 01672738. DOI: 10.1016/0167--2738(90)90142--E.

[107] B. Schweighofer, K. M. Raab und G. Brasseur. „Modeling of high power automotive batteries by the use of an automated test system“. In: *IEEE Transactions on Instrumentation and Measurement* 52.4 (2003), S. 1087–1091. ISSN: 0018-9456. DOI: 10.1109/TIM.2003.814827.

[108] Ralf Benger, Heinz Wenzl, Hans-Peter Beck, Meina Jiang, Detlef Ohms und Gunter Schaedlich. „Electrochemical and thermal modeling of lithium-ion cells for use in HEV or EV application“. In: *World Electric Vehicle Journal* 3.2 (2009), S. 342–351. ISSN: 2032-6653. DOI: 10.3390/wevj3020342.

[109] V. Ganesh Kumar. „Electrode impedance parameters and internal resistance of a sealed LiC/Li1-xCoO2 lithium-ion rechargeable battery“. In: *Journal of Applied Electrochemistry* 27.1 (1997), S. 43–49. ISSN: 0021891X. DOI: 10.1023/A:1026462815110.

[110] Ryan Ahmed, Javier Gazzarri, Simona Onori, Saeid Habibi, Robyn Jackey, Kevin Rzemien, Jimi Tjong und Jonathan LeSage. „Model-Based Parameter Identification of Healthy and Aged Li-ion Batteries for Electric Vehicle Applications“. In: *SAE International Journal of Alternative Powertrains* 4.2 (2015), S. 233–247. ISSN: 2167-4205. DOI: 10.4271/2015-01-0252.

[111] Ana-Irina Stroe, Jinhao Meng, Daniel-Ioan Stroe, Maciej Świerczyński, Remus Teodorescu und Søren Knudsen Kær. „Influence of Battery Parametric Uncertainties on the State-of-Charge Estimation of Lithium Titanate Oxide-Based Batteries“. In: *Energies* 11.4 (2018). ISSN: 1996-1073. DOI: 10.3390/en11040795. URL: https://www.mdpi.com/1996-1073/11/4/795.

[112] Bodo Schönemann und Roman Henze. *Auswirkungen alternativer Antriebskonzepte auf die Fahrdynamik von Pkw: [Bericht zum Forschungsprojekt FE 82.0525/2011].* Bd. 96. Berichte der Bundesanstalt für Straßenwesen Fahrzeugtechnik. Bremen: Fachverl. NW, 2014. ISBN: 978-3-95606-106-6. URL: http://bast.opus.hbz-nrw.de/volltexte/2015/837/pdf/F96b.pdf.

[113] Markus Rabe, Sven Spieckermann und Sigrid Wenzel. *Verifikation und Validierung für die Simulation in Produktion und Logistik: Vorgehensmodelle und Techniken.* Bd. 1. Berlin, Heidelberg: Springer Berlin, Heidelberg, 2008. ISBN: 978-3-540-35282-2. DOI: 10.1007/978-3-540-35282-2.

[114] MathWorks. *Reinforcement Learning Toolbox: User´s Guide 2022b.* Hrsg. von The MathWorks. 2022. URL: https://de.mathworks.com/help/pdf_doc/reinforcement-learning/rl_ug.pdf (besucht am 20. Aug. 2022).

[115] MathWorks. *Create Policies and Value Functions.* URL: https://de.mathworks.com/help/reinforcement-learning/ug/create-policy-and-value-function-representations.html (besucht am 30. Aug. 2022).

[116] Maximilian Graf. „Entwicklung und Analyse einer Deep-Reinforcement-Learning-basierten Leistungsmanagementstrategie für 12 V-Bordnetze“. Masterarbeit. Konstanz: Hochschule für Technik, Wirtschaft und Gestaltung, 2022.

[117] Volodymyr Mnih, Adrià Puigdomènech Badia, Mehdi Mirza, Alex Graves, Tim Harley, Timothy P. Lillicrap, David Silver und Koray Kavukcuoglu. „Asynchronous Methods for Deep Reinforcement Learning“. In: *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48.* ICML'16. JMLR.org, 2016, S. 1928–1937.

[118] Ömer Tan, Daniel Jerouschek, Ralph Kennel und Ahmet Taskiran. „Energy Management Strategy in 12-Volt Electrical System Based on Deep Reinforcement Learning“. In: *Vehicles* 4.2 (2022), S. 621–638. ISSN: 2624-8921. DOI: 10.3390/vehicles4020036.

[119] Hado van Hasselt, Arthur Guez und David Silver. „Deep Reinforcement Learning with Double Q-Learning“. In: *Proceedings of the AAAI Conference on Artificial Intelligence* 30.1 (2016). ISSN: 2374-3468. DOI: 10.1609/aaai.v30i1.10295.

[120] Dingbo He, Yuan Zou, Jinlong Wu, Xudong Zhang, Zhigang Zhang und Ruizhi Wang. „Deep Q-Learning Based Energy Management Strategy for a Series Hybrid Electric Tracked Vehicle and Its Adaptability Validation“. In: *2019 IEEE Transportation Electrification Conference and Expo (ITEC)*. 2019, S. 1–6. DOI: 10.1109/ITEC.2019.8790630.

[121] Xuewei Qi, Yadan Luo, Guoyuan Wu, Kanok Boriboonsomsin und Matthew J. Barth. „Deep reinforcement learning-based vehicle energy efficiency autonomous learning system“. In: *2017 IEEE Intelligent Vehicles Symposium (IV)*. 2017, S. 1228–1233. DOI: 10.1109/IVS.2017.7995880.

[122] Xiaowei Guo, Teng Liu, Bangbei Tang, Xiaolin Tang, Jinwei Zhang, Wenhao Tan und Shufeng Jin. „Transfer Deep Reinforcement Learning-Enabled Energy Management Strategy for Hybrid Tracked Vehicle“. In: *IEEE Access* 8 (2020), S. 165837–165848. DOI: 10.1109/ACCESS.2020.3022944.

[123] Xinyou Lin, Binhao Zhou und Yutian Xia. „Online Recursive Power Management Strategy Based on the Reinforcement Learning Algorithm With Cosine Similarity and a Forgetting Factor“. In: *IEEE Transactions on Industrial Electronics* 68.6 (2021), S. 5013–5023. ISSN: 0278-0046. DOI: 10.1109/TIE.2020.2988189.

[124] Francisco Sanchez Gorostiza und Francisco M. Gonzalez-Longatt. „Deep Reinforcement Learning-Based Controller for SOC Management of Multi-Electrical Energy Storage System“. In: *IEEE Transactions on Smart Grid* 11.6 (2020), S. 5039–5050. ISSN: 1949-3053. DOI: 10.1109/TSG.2020.2996274.

[125] Hadi S. Jomaa, Josif Grabocka und Lars Schmidt-Thieme. „Hyp-RL : Hyperparameter Optimization by Reinforcement Learning“. In: *arXiv* (2019). DOI: 10.48550/ARXIV.1906.11527.

[126] Frank Hutter, Holger H. Hoos und Kevin Leyton-Brown. „Sequential Model-Based Optimization for General Algorithm Configuration“. In: *Learning and Intelligent Optimization*. Hrsg. von Carlos A. Coello Coello. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, S. 507–523. ISBN: 978-3-642-25566-3.

[127] Bobak Shahriari, Kevin Swersky, Ziyu Wang, Ryan P. Adams und Nando de Freitas. „Taking the Human Out of the Loop: A Review of Bayesian Optimization“. In: *Proceedings of the IEEE* 104.1 (2016), S. 148–175. DOI: 10.1109/JPROC.2015.2494218.

[128] Chang Xu, Tao Qin, Gang Wang und Tie-Yan Liu. „Reinforcement Learning for Learning Rate Control“. Diss. Tianjin, China: Nankai University. DOI: 10.48550/arXiv.1705.11159.

[129] Peter Henderson, Riashat Islam, Philip Bachman, Joelle Pineau, Doina Precup und David Meger. „Deep Reinforcement Learning That Matters“. In: *Proceedings of the AAAI Conference on Artificial Intelligence* 32.1 (2018). ISSN: 2374-3468. DOI: 10.1609/aaai.v32i1.11694.

[130] Roman Liessner, Jakob Schmitt, Ansgar Dietermann und Bernard Bäker. „Hyperparameter Optimization for Deep Reinforcement Learning in Vehicle Energy Management“. In: *11th International Conference on Agents and Artificial Intelligence*, S. 134–144. DOI: 10.5220/0007364701340144.

Anhang

A.1 Notation

Tabelle A.1: Übersicht der Hinweise zur mathematischen Notation.

Notation	**Beschreibung**
$\mathbb{E}_\pi$	Erwartungswert unter Verfolgen einer spezifischen Policy π
V^π	Zustands-Wertfunktion unter Verfolgen einer spezifischen Policy π (gilt ebenso für Q^π und A^π)
π^*	Optimale Policy
V^*	Optimale Zustands-Wertfunktion (gleichbedeutend mit Zustands-Wertfunktion unter Verfolgen der optimalen Policy π^*; gilt ebenso für Q^*)
$a \sim \pi(s, \cdot)$	Sampling der Aktion a aus Wahrscheinlichkeitsverteilung der Policy π in Abhängigkeit vom Zustand s
$\tau \sim\ p(\tau)$	Sampling einer Trajektorie τ aus der Wahrscheinlichkeitsverteilung $p(\tau)$
$\propto$	Proportionalität
$\widehat{Q}$	Schätzwert für Wertfunktion