Modeling and Control of Power Electronics Converters Using Machine Learning

by

Pouria QASHQAI

MANUSCRIPT-BASED THESIS PRESENTED TO ÉCOLE DE TECHNOLOGIE SUPÉRIEURE IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF DOCTOR OF PHILOSOPHY
Ph.D.

MONTREAL, JUNE 18TH, 2025

ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
UNIVERSITÉ DU QUÉBEC

BOARD OF EXAMINERS

THIS THESIS HAS BEEN EVALUATED

BY THE FOLLOWING BOARD OF EXAMINERS

Mr. Kamal Al-Haddad, Thesis Supervisor
Department of Electrical Engineering, École de technologie supérieure

Mr. Gabriel J. Assaf, President of the Board of Examiners
Department of Electrical Engineering, École de technologie supérieure

Mrs. Lyne Woodward, Member of the jury
Department of Electrical Engineering, École de technologie supérieure

Mr. Innocent Kamwa, External Evaluator
Department of Electrical and Computer Engineering, Université Laval

Mr. Rawad Zgheib, External Evaluator
G&W Electric Co.

THIS THESIS WAS PRESENTED AND DEFENDED

IN THE PRESENCE OF A BOARD OF EXAMINERS AND PUBLIC

JUNE 10TH, 2025

AT ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

FOREWORD

The rapid integration of modern technologies such as electric vehicles (EVs), smart grids, and renewable energies has provided significant advantages while posing immense challenges for the modern energy industry. The following thesis is a piece of research conducted to address challenges in modeling and control of power electronic converters, which are the core of these technologies.

When this research was initially proposed, the focus was on modeling chargers of electrical vehicles to better understand their impact on the grid. Conventional grid studies did not face such challenges due to the static nature of the loads connected to them. However, electric vehicles being large portable electrical loads in some scenarios and relatively small, distributed energy generation sources in others (such as vehicle-to-grid mode), present substantial challenges in studying modern electric grids. As the research progressed, it became evident that power electronic converters are at the core of EV chargers. This realization led us to have a necessary shift in focus and broadening our scope of research. Considering the widespread use of power electronic converters, the findings and the proposed methods of this thesis can apply to many other applications as well.

The novel modeling and control techniques proposed and developed in this thesis provide a substantial contribution to the field of research in the energy sector by utilizing artificial intelligence to overcome two key challenges in this domain: accurate black-box modeling and model-free control. At the heart of this thesis, purely data-driven black-box modeling for over-the-counter converters and model-free control using deep reinforcement learning are proposed and their advantages and limitations are investigated.

This research not only advances the state of the art in modeling and control of power electronics but also offers solutions based on artificial intelligence which can be used to facilitate studying and integration of power electronic converters in a broad range of similar applications in the

energy sector. We believe that the findings of this research have the potential to contribute significantly to shaping the future of modern electric grids.

ACKNOWLEDGMENTS

"The deepest principle in human nature is the craving to be appreciated." – William James.

First and foremost, I want to express my deepest gratitude to my wife, Suzanne. Her unwavering love, patience, and understanding throughout this journey have been my greatest source of strength. She's been there through every tough moment, grounding me when things felt overwhelming, and lighting my path when I doubted myself. Suzanne, your belief in me, even when I found it hard to believe in myself, kept me going. I truly owe this achievement to you in so many ways.

I also want to sincerely thank my supervisor, Professor Kamal Al-Haddad, for his exceptional guidance, wisdom, and constant encouragement. Your insights have not only shaped this work but also challenged me to think beyond my limits. And to my co-supervisor, Rawad Zgheib—Rawad, thank you for your invaluable feedback, unwavering support, and for always being open and welcoming whenever I needed advice. Professor Kamal and Rawad, you've both played an essential role in my academic growth, and I'm incredibly grateful for your support.

To my parents, Alamdar and Khadijeh, I am so thankful for everything you've done for me. You've been my foundation, providing constant support through the ups and downs of this journey. I especially want to acknowledge my beloved sisters, Pegah and Pardis. Your love and care have meant everything to me, and your encouragement has been a constant source of comfort and motivation. You're not only my sisters but my closest friends, and I'm so lucky to have you both in my life.

And last but not least, I want to take a moment to thank myself. For every late night, every challenge I faced head-on, and for choosing perseverance over giving up. Through every struggle and triumph, I've grown, and for that, I'm proud of myself.

MODÉLISATION ET CONTRÔLE DES CONVERTISSEURS D'ÉLECTRONIQUE DE PUISSANCE À L'AIDE DE L'APPRENTISSAGE AUTOMATIQUE

Pouria QASHQAI

RÉSUMÉ

Dans cette thèse, une nouvelle technique de modélisation de type « boîte noire » pour les convertisseurs d'électronique de puissance utilisant l'apprentissage automatique est proposée. Les techniques de modélisation conventionnelles peuvent être inadaptées pour les convertisseurs présents sur le marché ne disposant pas de fiches techniques précises. Un modèle purement basé sur les données est développé en utilisant des réseaux à mémoire à long terme (LSTM – Long Short-Term Memory), puis sa performance est comparée aux unités récurrentes à portes (GRU – Gated Recurrent Unit). La comparaison démontre que LSTM et GRU sont des options viables selon l'équilibre souhaité entre performance et précision. De plus, afin d'améliorer l'efficacité computationnelle, l'apprentissage par transfert est utilisé pour accélérer le processus d'entraînement des convertisseurs ayant des topologies similaires.

De plus, Hypersim, un logiciel de simulation en temps réel largement utilisé dans les systèmes électriques et l'électronique de puissance, est principalement conçu pour des modèles mathématiques, ce qui peut être limitant pour les convertisseurs commerciaux dépourvus de fiches techniques détaillées. Puisque Hypersim ne prend pas en charge nativement la méthode de modélisation en « boîte noire » proposée, un algorithme de simulation en temps réel a été développé. Cet algorithme comble l'écart entre les modèles de convertisseurs basés sur l'apprentissage automatique et les environnements de simulation en temps réel, représentant ainsi une avancée significative pour les tests hardware-in-the-loop ainsi que pour l'implémentation pratique.

L'étude présente également une stratégie de contrôle sans modèle pour les convertisseurs d'électronique de puissance utilisant l'apprentissage par renforcement profond (DRL – Deep Reinforcement Learning). À travers de larges simulations et résultats expérimentaux, la méthode de contrôle basée sur le DRL est comparée au contrôle prédictif modélisé (MPC – Model Predictive Control), une méthode de contrôle non linéaire conventionnelle. Il a été démontré que l'approche DRL élimine non seulement la nécessité d'un modèle mathématique précis, mais est également plus résiliente aux écarts de paramètres, aux variations des points de fonctionnement, à l'incertitude et au bruit.

Enfin, la méthode DRL est appliquée à un convertisseur plus complexe, le convertisseur Hybrid Packed U-Cell (HPUC), qui génère 23 niveaux de tension avec une seule source de courant continu. Le contrôle et l'équilibrage des tensions dans ce convertisseur sont complexes, cependant, la méthode DRL a montré des avantages par rapport au MPC traditionnel en termes de robustesse face aux écarts de paramètres, au bruit et aux autres perturbations. Cette recherche ouvre de nouvelles perspectives pour l'utilisation des techniques émergentes d'apprentissage automatique afin de résoudre des problèmes complexes dans le domaine de la modélisation et du contrôle des convertisseurs électroniques de puissance.

Mots-clés: convertisseurs d’électronique de puissance, apprentissage automatique, lstm (long short-term memory), gru (gated recurrent unit), modélisation en boîte noire, apprentissage par renforcement profond (drl – deep reinforcement learning), contrôle sans modèle, simulation en temps réel

MODELING AND CONTROL OF POWER ELECTRONICS CONVERTERS USING MACHINE LEARNING

Pouria QASHQAI

ABSTRACT

In this thesis, a novel black-box modeling technique for power electronics converters using machine learning is proposed. The conventional modeling techniques may be impractical for over-the-counter converters that lack accurate data sheets. A purely data-driven model is developed using Long Short-Term Memory (LSTM) networks, and then its performance is compared against Gated Recurrent Units (GRUs). The comparison demonstrates that both LSTM and GRU are viable options depending on the desired balance between performance and accuracy. Additionally, to improve computational efficiency, transfer learning is employed to speed up the training process for converters with similar topologies.

MATLAB was selected for developing and training the machine learning models due to its robust toolboxes for control systems and deep learning, offering seamless integration with Simulink for dynamic system simulations. Its flexibility and computational efficiency made it a suitable choice over other platforms for implementing LSTM networks and DRL algorithms, while Hypersim facilitated the transition to real-time testing.

Furthermore, Hypersim, a popular real-time simulation software for electrical systems and power electronics, is mainly designed for mathematically based models, which can be limiting for commercial converters that lack detailed data sheets. Since Hypersim does not natively support the proposed black-box modeling method, an algorithm for real-time simulation in Hypersim is developed. This algorithm bridges the gap between the machine-learning-based converter models and real-time simulation environments and it provides a significant step forward for hardware-in-the-loop as well as practical implementation.

The study also presents a model-free control strategy for power electronics converters using Deep Reinforcement Learning (DRL). Through broad simulation and experimental results, the DRL-based control method is compared against conventional Model Predictive Control (MPC), a conventional non-linear control method. The DRL approach is shown to not only eliminate the need for an accurate mathematical model but also to be more resilient to parameter mismatch, variations in operation points, uncertainty, and noise.

Ultimately, the DRL method is applied to a more complex converter, the Hybrid Pakced U-Cell (HPUC) converter, which generates 23 voltage levels using a single DC source. Control and voltage balancing in this converter are challenging, however, the DRL method demonstrated advantages over traditional MPC in terms of robustness under parameter mismatch, noise, and other disturbances. This research presents new opportunities for utilizing the power of emerging machine learning techniques to solve challenging problems in the domain of modeling and control of power electronic converters.

Keywords: power electronics converters, machine learning, lstm, gru, black-box modeling, deep reinforcement learning (drl), model-free control, real-time simulation

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

AC	Alternative Current
AI	Artificial Intelligence
ANN	Artificial Neural Network
ANPC	Active Neutral Point Clamped
CPL	Constant Power Load
DC	Direct Current
DNN	Deep Neural Network
DQN	Deep Q-learning Network
DRL	Deep Reinforcement Learning
ESO	Extended State Observer
FNN	Fuzzy Neural Network
GRU	Gated Recurrent Unit
HIL	Hardware-In-the-Loop
HPUC	Hybrid Packed U-Cell
IGBT	Insulated Gate Bipolar Transistor
LSTM	Long Short-term Memory
ML	Machine Learning
MPC	Model Predictive Control
MLI	Multi-Level Inverter
NPC	Neutral Point Clamped
PEC	Power Electronic Converter
PID	Proportional Integral Derivative

PSO	Particle Swarm Optimization
PUC	Packed U-Cell
PUC5	5-Level Packed U-Cell
PUC7	7-Level Packed U-Cell
PWM	Pulse Width Modulation
RL	Reinforcement Learning
RNN	Recurrent Neural Network
RTS	Real-Time Simulation
SIL	Software-In-the-Loop
SVM	Space Vector Modulation
THD	Total Harmonic Distortion

INTRODUCTION

With the emergence of modern technologies such as Electric Vehicles (EVs), smart grids, renewable energy, and distributed energy generation, studying their impact on the electric grid has become utterly important in the past decades. Power electronic converters (PECs) are at the core of such technologies and they are becoming more complex as the technology advances (Oghenewvogaga, Adetokun, Gamiya, & Nagode, 2022). Thus there is a need for advanced modeling and control techniques for modern power electronic converters (Bacha, Munteanu, & Bratcu, 2014).

The latest advancements in machine learning have enabled us to solve a vast domain of conventional problems in power systems and power electronics (Pouria Qashqai, Vahedi, & Al-Haddad, 2019). Therefore, this piece of research is focused on utilizing the ever-increasing power of artificial intelligence to address the modeling and control of complex power electronic converters (PECs) in modern energy systems. As technologies like electric vehicles, renewable energy generation, and smart grids continue to progress, their interaction with the electric grid imposes new challenges. Power electronic converters, which are integral to these technologies, have grown more complex, and in certain cases, traditional methods may not be adequate and thus require advanced methods for their modeling and control (Krishnamoorthy & Aayer, 2019).

By utilizing artificial intelligence, this research aims to develop new solutions for modeling power electronic converters, particularly black-box models. An advanced AI-based control method is also proposed to overcome the limitations of traditional non-linear control methods, such as model predictive control. These limitations include sensitivity to noise, parameter mismatch, and large disturbances.

Motivation and Challenges

The main motivation behind this thesis, as stated in the proposal report, was to leverage the power of emerging machine-learning technologies in the domain of modeling electrical vehicles. One of the key challenges in studying the impact of EVs is modeling their charging stations (Dubey, Santoso, & Cloud, 2013).

During the early stages of this study, we discovered that the primary challenges in accurately and effectively modeling EV chargers arise from the complex and non-linear nature of the power electronic converters (PECs) used in them. A key issue is the lack of detailed and accurate data sheets for commercially available PECs. As a result, we focused on developing AI-based solutions to treat these converters as black boxes, enabling us to model and control them more effectively and overcome these challenges.

Similarly, in our literature review, we studied numerous non-linear control methods designed for highly complex and non-linear power electronic converters (PECs). However, these methods often relied on having an accurate model of the converter and were highly sensitive to parameter mismatch, noise, disturbances, and other uncertainties.

These limitations, along with the recent advancements in artificial intelligence, motivated us to develop novel machine-learning-based methods for modeling and controlling power electronics, addressing the gaps found in traditional approaches.

Research Objectives

We aimed for two key objectives in this piece of research. First and foremost, developing a purely data-driven approach based on artificial intelligence for black-box modeling of power electronic converters. The idea is to utilize readily available data generated by various sensors in modern power electronics to obtain a model of their small signal and/or large signal behavior. The accuracy of the model is adjustable based on the size of the training data, the

quality of the training data, the size of the recurrent network, and training times. Therefore, arbitrary levels of accuracy and fidelity can be obtained for studying the impact of PECs especially in the integration of modern technologies such as EVs.

The second objective of this thesis is to develop a model-free control method for non-linear complex power electronic converters. Various control methods based on Deep Reinforcement Learning (DRL) are introduced in the literature. Due to the continuous nature of measurement signals and high dimensionality of the states and also considering the discrete nature of actions (switching states), Deep Q-learning Networks (DQN) are used to achieve this objective. In addition to being model-free, DRL can be used to achieve resilience towards uncertainties due to the generalization abilities embedded in the Deep Neural Networks (DNN).

Methodology

Deep Reinforcement Learning (DRL) agents require observations and rewards to select the optimal course of action in each sampling time. In Chapters 2 and 3, we have explained this approach in detail. But as a summary, the simplified methodology of implementation of the DRL agent to control PECs is depicted in Figure 0.1.

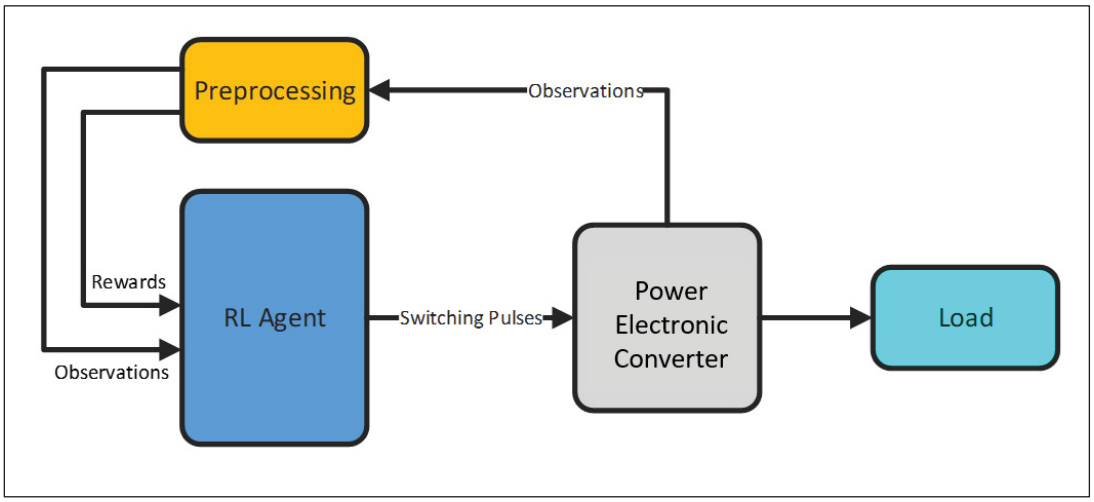

Figure 0.1 Implementation of DRL agent for optimal control of PECs
Taken and adapted from Qashqai et al. (2020, p. 3)

To acquire an accurate black-box model of PECs, we have considered them as two-port networks as shown in Figure 0.2. By measuring the input and output voltages and currents in training operation points, training datasets are obtained to be fed to the recurrent neural networks of the proposed method.

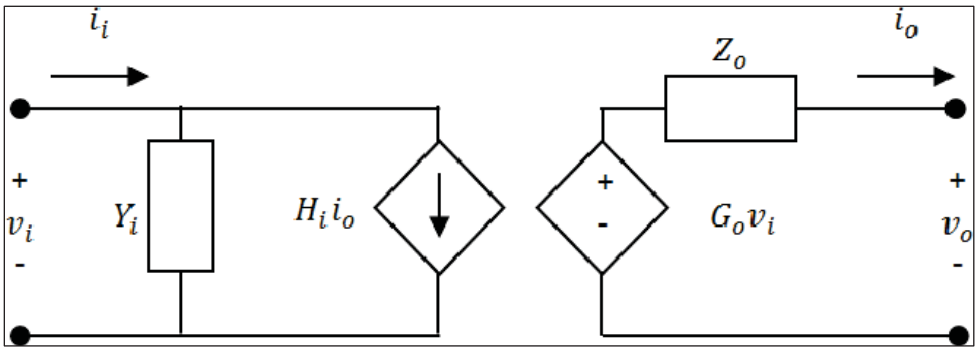

Figure 0.2 Two-port networks approach for modeling PECs

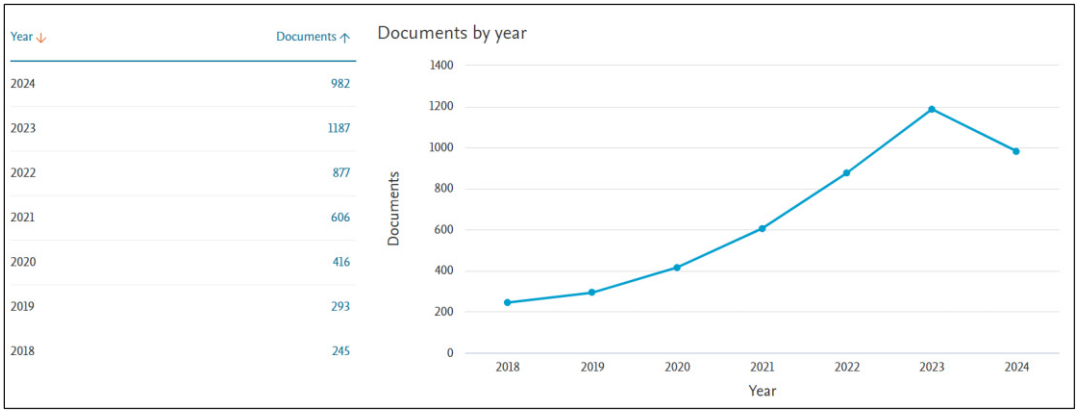

Figure 0.3 Number of publications containing Artificial Intelligence in combination with Either Power electronic Converters or Power Converters

Timeliness of The Problem

There has been a growing number of publications over the past years that demonstrate the increasing popularity of utilizing artificial intelligence on power electronic converters (PECs). As seen in Figure 0.3, publications containing the keywords “Artificial Intelligence” and either “power electronic converters” or “power converters” have grown steadily since 2018 (Scopus

, 2024). Despite this surge in the number of publications, there remain still numerous unresolved challenges that require innovative methods, such as those proposed in this study.

In addition, the second chart demonstrated in Figure 0.4, depicts the top publishers of these studies. These publishers are among the most prestigious in the fields of power electronics and industrial electronics. Their consistent and growing interest highlights the timeliness and relevance of the problem investigated in this thesis.

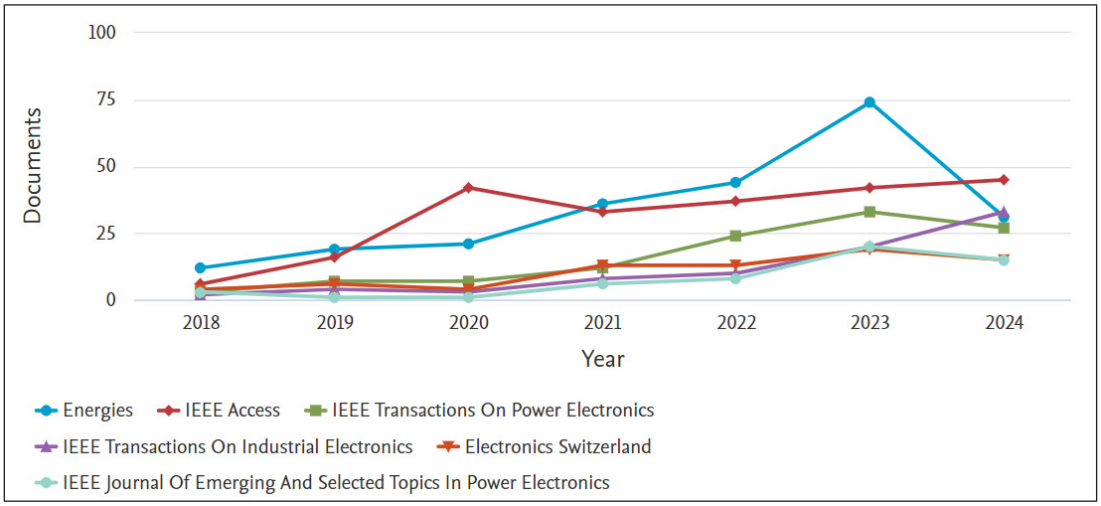

Figure 0.4 Number of journal articles containing Artificial Intelligence and Power Electronic Converters published by the top publishers

Ultimately, the chart depicted in Figure 0.5, illustrates the trends in different combinations of keywords between Artificial Intelligence (AI), Reinforcement Learning (RL), Artificial Neural Networks (ANN), Power Electronic Converters (PEC), and real-time simulation (RTS). The highest volume of publications is associated with keywords like "AI + PEC" and "ML + PEC," indicating that the integration of machine learning and artificial intelligence into PEC modeling is not only timely but essential. This research is positioned at the heart of this intersection, aiming to resolve the critical limitations of conventional approaches by leveraging AI's strengths in adaptability, pattern recognition, and black-box modeling, especially for applications like electric vehicle charging and renewable energy integration.

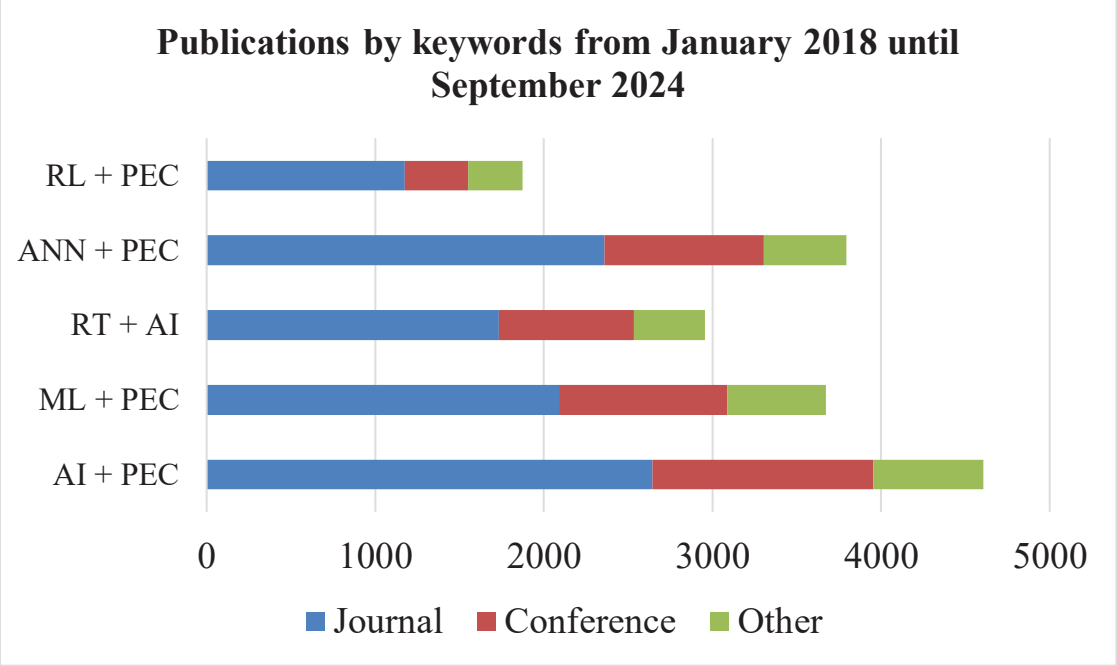

Figure 0.5 Number of publications based on combinations of the key terms used in this thesis since 2018

Thesis Contribution

This thesis makes several key contributions to the field of power electronics modeling and control using machine learning techniques. First, a new black-box modeling method is proposed, based on machine learning algorithms such as Long Short-Term Memory (LSTM) networks and Gated Recurrent Units (GRUs). This approach mitigates the issues associated with modeling over-the-counter and commercial converters which have limited technical data and often lack detailed data sheets. The models developed are purely data-driven and provide accurate predictions for both small and large signal behaviors of power electronic converters (PECs). Moreover, the use of transfer learning is investigated to further increase computational efficiency. Transfer learning is shown to make it possible to accelerate the training process for converters with similar topologies. Additionally, a programmatic algorithm is developed for the integration of the proposed black-box models and other DNN-based models in Hypersim. Hypersim does not natively support such blocks. Thus, using this approach, future researchers can study such models in real time.

Ultimately, this thesis adds to the body of knowledge by proposing a model-free control method based on Deep Reinforcement Learning (DRL). On top of eliminating the need for an accurate mathematical model of complex converters with non-linear control requirements, the proposed method is also capable of generalization. These inherent characteristics of deep neural networks including deep reinforcement learning agents lead to advantages over conventional non-linear methods such as Model Predictive Control (MPC) in terms of resilience to parameter mismatch, large disturbances, noise in measurements, and uncertainties. The proposed DRL method is compared with the traditional MPC method, in simulation, real-time simulation, and experimental tests to assess its advantages. The DRL control method is applied to the three-level Neutral Point Clamped (NPC) as well as the 23-level Hybrid Packed U-Cell (HPUC) converter. The HPUC was selected as a case study because it can generate multiple voltage levels with fewer components, making it highly efficient for modern applications such as electric vehicles and renewable energy. But the main drawback is that controlling and balancing the voltage in the HPUC is challenging. However, the proposed model-free DRL method demonstrates satisfactory control in steady-state mode and superior resilience when facing uncertainties, proving its potential for resilient and intelligent control of modern complex PECs.

Overall, the implementation of the techniques proposed in this thesis into the field of power electronics creates novel opportunities for improving the modeling efficiency and control reliability of complex converters in various modern applications such as renewable energy systems and electric vehicles.

Thesis Outline

This thesis is divided into five main chapters. In the first chapter, a critical review of the existing literature is presented. First, the necessity of studying the impact of EV chargers especially the non-linear core of such chargers which is their power electronic converters is explained then conventional modeling and control methods and their shortcomings are explained. Eventually, modern and particularly AI-based modeling and control techniques are

introduced and the gaps in the literature are explained. This chapter is reproduced and inspired by all the publications of the author along with his colleagues during the course of his Ph.D. (Pouria Qashqai, Al-Haddad, & Zgheib, 2022, 2020b, 2020a; Pouria Qashqai, Babaie, Zgheib, & Al-Haddad, 2023; Pouria Qashqai et al., 2019; Pouria Qashqai, Zgheib, & Al-Haddad, 2021, 2022).

Then in the second chapter, a model-free control method based on DRL is proposed and implemented on a three-level NPC. This chapter is extracted from a paper published by the author of this thesis and his colleagues in IEEE ACCESS.

Moreover, in the third chapter, a similar model-free DRL control method is proposed and applied to the 23-level HPUC converter which requires a significantly more complex control. This chapter is submitted for publication in IEEE ACCESS.

Additionally, in the fourth chapter, a comparative analysis between the DRL method and MPC is performed. The offline simulation and real-time simulations obtained in this chapter are used to validate the performance of the proposed method against MPC. This chapter is submitted for publication in IEEE ACCESS.

Furthermore, CHAPTER 5 is based on four IEEE conference proceedings published by the author of this thesis and his colleagues (Pouria Qashqai, Al-Haddad, et al., 2022; Pouria Qashqai, Al-Haddad, & Zgheib, 2020c; Pouria Qashqai et al., 2021; Pouria Qashqai, Zgheib, et al., 2022). In this chapter, the black-box modeling technique based on machine learning is proposed and its accuracy is evaluated. The performance and computational cost of two architectures called LSTM and GRU for the proposed black-box method are calculated and discussed. Transfer learning is also used to accelerate the training times of the proposed modeling method in PECs with similar topologies. Eventually, since Hypersim a popular real-time simulation tool, does not natively support the implementation of DNNs, an algorithm is introduced to automatically create the equivalent Simulink model of the DNN used in this

chapter. Real-time simulations in Hypersim are presented to demonstrate the satisfactory accuracy of the proposed method.

Ultimately, in the conclusion section, the key results of this research are explained concisely for the readers to understand the main findings of this thesis. Additionally, the limitations of the proposed methods are discussed, and the remaining gaps are explained to guide future researchers in addressing them.

CHAPTER 1

REVIEW OF THE EXISTING LITERATURE

1.1 Introduction

Traditional fossil fuels are known to be associated with numerous serious environmental concerns. The most prominent issue is the rise of the average temperature on planet Earth, which is known as "Global Warming". The effect is believed to be a result of human activities such as burning fossil fuels (Cook et al., 2016; Lendzen, 2010). Moreover, fossil fuel resources are known to be limited, and the industry will eventually run out of them sooner or later. The main fossil fuels are Oil, Gas, and Coal. The oil will run out by 2052, gas is estimated to last until 2060, and finally, coal resources will end by 2090 ("When Fossil Fuels Run Out, What Then?," n.d.). In other words, these energy resources are not renewable.

Furthermore, in recent years, the generation of solar energy, which is a "clean" and renewable source of energy, has become cheaper than some fossil fuels (i.e. coal) and it is estimated that by the end of 2025, coal will be more expensive than other forms of renewable energy such as wind energy ("The Coal Cost Crossover," n.d.).

The drawbacks associated with using fossil fuels as well as the promising technology advancements in renewable energy generation have encouraged industry and academia to develop alternative solutions. One of these solutions which have targeted the transportation area is the development of electrical vehicles (also known as EVs). As the name suggests, electrical vehicles run on electric energy. Whether as their primary source (full EV) or complimentary source (hybrid vehicle). It is estimated that by the end of 2030, there will be between 130-250 million electric vehicles utilized all over the world while China will be the largest player (*IEA (2019) Report, "Global EV Outlook 2019", IEA, Paris. Available at: Www.Iea.Org/Publications/Reports/Globalevoutlook2019/*, n.d.).

In addition to the advantages mentioned above, there are numerous other benefits of using electric vehicles versus conventional internal combustion engine (ICE) vehicles.

In terms of energy consumption, electric motors are, by nature, more efficient than internal combustion engines. Moreover, unlike combustion engines in which braking energy due to friction is wasted as heat, electric motors can harvest the kinetic energy produced by brakes which leads to even more efficiency (Vražić, Vuljaj, Pavasović, & Pauković, 2014).

In terms of fuel cost, in addition to the unsustainability, price drop, and pollution of fossil fuels, there are many benefits of using electric energy that give electric vehicles an immense advantage point. First of all, EVs reduce the dependency of countries on oil and petroleum imports, which is a matter of national security. Secondly, unlike oil, electricity prices do not suffer from sudden fluctuations. Finally, even when electricity is produced utilizing fossil-fuel, transforming it into electricity is still a good idea since controlling gas emissions of central power plants located far from urban areas is immensely easier than millions of fossil fuel vehicles spread all across the country (Wu, Gilchrist, Sealy, Israelsen, & Muhs, 2011).

Finally, in terms of maintenance cost, electric vehicles are cost-effective to maintain since they have a lower number of mechanical parts compared to ICE vehicles. This leads to higher reliability as well as lower maintenance costs and makes EVs more economical in the long run (Niculae, Iordache, Stanculescu, Bobaru, & Deleanu, 2019).

Despite providing numerous benefits over ICE vehicles, EVs suffer from some drawbacks among which two are the most prominent ones: lack of enough charging facilities and long charging times. These two drawbacks have discouraged typical users from purchasing EVs. However, with recent developments in EV battery technologies and the high penetration rate of EVs in developing and developed countries, both of these problems are being solved gradually. The most popular solution to the problem of long charging time is the “fast-charging” technology. Typical regular non-fast-charging EVs are equipped with batteries up

to 50 kilowatts that take a few hours for a full charge. While fast-charging EVs are equipped with batteries with several hundred kilowatts that require a few minutes to charge.

With raising the penetration rate of electric vehicles, especially those equipped with fast-charging technology, their impact on the distribution network becomes a challenge. As shown in Table 1.1, the power capacity of modern commercial EVs is relatively large and they are becoming even larger every year. Thus, not only does utilizing them have a remarkable impact on voltage profile, harmonics, demand response, etc. but also, unlike most consumers in a power network, electric vehicles are mobile and can act as a power storage unit which adds to the complexity of power flow, planning, and utilization of power grids.

Table 1.1 Typical commercial electrical vehicle battery capacities
Taken and adapted from Acharige et al. (2023, p. 5)

EV Model	**Battery Capacity (kWh)**	**Driving Range (km)**
Chevrolet Volt	18.4	85-Battery
Nissan Leaf Plus	64	480
Tesla Model S	100	620
Tesla Model X	100	500
Tesla Model 3	82	580
Audi e-tron	95	400
Jaguar ev400	90	450
Porsche Taycan	93	410

Often these complications and impacts of electric vehicles on the power grid are required to be studied in real time. Real-time simulation tools such as the ones developed by OPAL-RT (i.e. RT-LAB, Hypersim, and ARTEMIS) make it possible to study and monitor electric systems without their physical models. This leads to a reduction in research costs, faster prototyping, and less risk for the pieces of equipment. However, in order to obtain such real-time analysis, accurate models of the various elements should be developed. Power electronic converters are

highly non-linear and complex parts of electric vehicles. These systems should be modeled in a way that can provide the fastest simulation solution. This improvement in simulation speed often comes at the cost of losing some accuracy. Thus, these models should be the result of a trade-off between the two factors. The first part of this project is about developing a black-box model for PECs using machine learning methods.

Since the ultimate goal of these models is to be used in real-time simulation tools, in CHAPTER 5 of this thesis we have developed an algorithm to export the obtained DNN to Hypersim. Hypersim is a popular real-time simulation tool for power systems developed by Hydro-Quebec Research Center (IREQ[1]) and OPAL-RT Technologies.

In addition to modeling, control of modern PECs especially those implemented in commercial EVs are inherently non-linear and are increasingly becoming more complex. Thus, various methods are proposed in the literature to effectively control them.

The shift towards EVs is a vital step in addressing environmental concerns and achieving sustainable energy goals. However, the integration of EVs into existing power systems introduces new challenges that necessitate advanced modeling and control techniques. The following sections of this chapter introduce various modeling and control methods developed for such converters.

1.2 Importance of Modeling and Control of Power Electronics

Power electronics converters (PECs) are fundamental in numerous modern technologies such as renewable energy generation systems, smart grids, and electric vehicles (P. Qashqai, Sheikholeslami, Vahedi, & Al-Haddad, 2015a; G. Zhang, Li, Zhang, & Halang, 2018). The ever-increasing integration of PECs into these technologies necessitates obtaining accurate

[1] Institut de recherche d'Hydro-Québec (Hydro-Quebec Research Institution)

modeling and efficient control strategies to ensure reliable and efficient studying of their behaviors as well as achieving resilient and optimal operation (Bacha et al., 2014).

1.2.1 Challenges in Modeling Power Electronic Converters

PECs especially the modern topologies are comprised of a substantial number of non-linear elements, which leads to inherently non-linear and time-varying behavior. Conventional modeling methods are either incapable of fully capturing the dynamic behaviors of PECs in small-signal and large-signal modes or suffer from high computational burdens due to these complexities. Requiring an accurate model of PECs is not always feasible due to the limited technical data provided with commercial converters which necessitates black-box approaches to be taken. Additionally, the computational burden of conventional detailed switching models makes these methods inappropriate for applications like real-time simulations (Airán Francés, Asensi, García, Prieto, & Uceda, 2018; H. Y. Kanaan, Hayek, & Al-Haddad, 2007; Metri, Vahedi, Kanaan, & Al-Haddad, 2016).

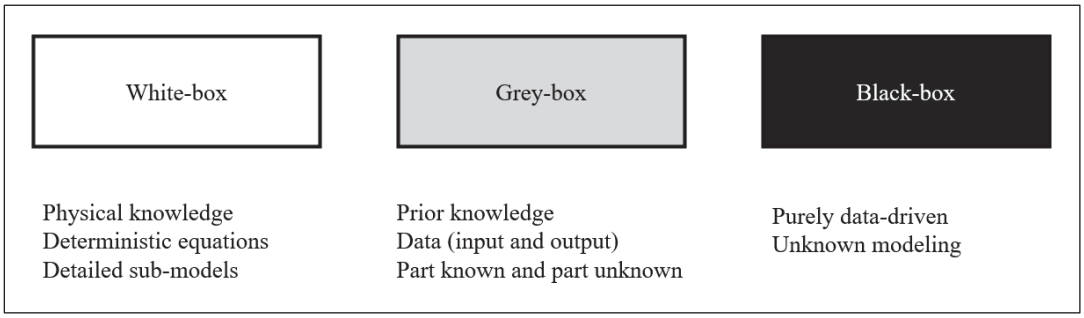

Figure 1.1 Types of modeling based on the depth of knowledge

System identification is an inherently complex problem, especially when dealing with non-linear and time-varying systems. For linear systems, various techniques, including numerical methods, are proposed in the literature for modeling their dynamics (Padhee, Pati, & Mahapatra, 2016). However, for non-linear dynamics of complex systems, conventional system identification methods are either incapable of modeling both smart-signal and large-

signal behaviors or impose heavy computational burdens. Therefore, alternative modeling methods have been introduced in the literature to mitigate these limitations.

Conventional modeling methods for PECs are often based on their mathematical models, which require knowledge about the structure of the converters (Padhee et al., 2016). These methods are known as "white-box" modeling methods as shown in Figure 1.1 (H. Kanaan, Al-Haddad, & Fnaiech, 2005; Ould-Bachir, Blanchette, & Al-Haddad, 2015; Riba, Moreno-Eguilaz, Bogarra, & Garcia, 2018). However, accurate data sheets containing detailed knowledge about the internal structure of most commercial PECs are not readily available. To overcome this issue, as depicted in Figure 1.1, "grey-box" and "black-box" modeling methods have been proposed in the literature (Kicsiny, 2017; Oliver, Prieto, Cobos, Garcia, & Alou, 2009).

In grey-box modeling methods, some components of a PEC are known, and the unknown parts are treated as a black box (Kicsiny, 2017; Oliver et al., 2009). In black-box methods on the other hand, the entire PEC is considered an unknown system, and models are extracted based on measurements in the inputs and outputs (Airán Francés et al., 2018; Guarderas, Francés, Asensi, & Uceda, 2017; Naziris, Asensi, & Uceda, 2018). In the next section, we will explore black-box modeling methods for PECs in greater detail.

1.3 Conventional Modeling Methods for Power Electronic Converters

Conventional modeling methods are primarily based on numerical analysis of Power Electronic Converters (PECs). Since most PECs consist of complex, non-linear components, various techniques have been proposed in the literature to approximate their behavior around specific operational points (small-signal analysis), average their behavior (averaging methods), or analyze them at several operational points to perform an interpolation (polytopic approach).

In the following subsections, the most prominent conventional modeling methods for PECs are introduced, and their advantages and disadvantages are discussed.

1.3.1 Small-Signal Modeling

Small-signal modeling is a conventional method used to analyze the behavior of power electronic converters (PECs) around a predetermined operating point. These models are typically linear approximations and are effective for systems where the operating conditions do not vary significantly (Iyer, Gulur, & Bhattacharya, 2017). However, in smart grids equipped with various types of PECs requiring time-variant complex control methods, PECs often behave as non-linear, time-varying components. In such cases, small-signal models become incompatible and demonstrate substantial limitations and drawbacks (Airán Francés et al., 2018).

1.3.2 Averaging Methods

Averaging methods are mathematical techniques utilized to simplify the modeling of PECs by averaging their behaviors over a switching period. These methods extract continuous-time models that approximate a given converter's operation and thus make it easier to analyze and design control systems. Averaging is often utilized for converters operating at high switching frequencies, where modeling every switching event becomes impractical due to complexity and computational burden (H. Y. Kanaan et al., 2007). This heavy computational burden often arises from the non-linear nature of switches and diodes.

1.3.2.1 State-Space Averaging (SSA)

State-space averaging (SSA) is one of the most popular averaging methods for modeling PECs (Middlebrook & Cuk, 1976). SSA extracts a simplified and approximated representation of the state variables over a switching period. Then this simplified presentation is treated as a continuous-time model that approximates the converter's behavior. This approach allows the converter to be modeled as a continuous-time system. This method is very effective for

converters that operate at high switching frequencies, in which state variables have small ripples and can be averaged with minimal loss of accuracy.

However, SSA has critical limitations when dealing with complex converter topologies such as multilevel converters, where multiple switching states and parasitic components lead to additional modeling challenges. Although capacitors, inductors, and transformers naturally experience voltage and current fluctuations, SSA handles this by averaging over the switching period, which can miss important transient effects and high-frequency interactions. As the number of levels increases, it becomes increasingly challenging to maintain computational efficiency while keeping the model accurate. Moreover, parasitic components often introduce resonances and dynamic interactions that SSA cannot fully capture. These limitations make SSA less suitable for highly complex converter topologies with low switching frequencies or systems where high-frequency effects play a significant role. In such cases, we need a more detailed representation of component dynamics to ensure accurate simulation and control performance. (Jianping Xu & Lee, 1997).

1.3.2.2 Generalized State-Space Averaging (GSSA)

To address the limitations of SSA, a generalized state-space averaging (GSSA) technique was proposed (Jianping Xu & Lee, 1997). GSSA can model large-signal and AC signals. Thus it expands the applicability of averaging methods to a broader range of converters. However, this approach becomes exponentially complex as the number of components in the PEC increases. The increased complexity makes GSSA less practical for converters with a substantial number of elements, as it requires a heavy computational burden.

1.3.2.3 Discrete Averaging Methods

Discrete averaging methods (DA) provide high accuracy in modeling PECs and are generally free from the problems associated with state-space-based averaging methods (X. Li, Ruan, Jin, Sha, & Tse, 2018; Qiu, Liu, & Chen, 2010). DA methods consider the discrete nature of power

converters by analyzing the converter behavior at specific intervals within the switching period. However, DA methods are complex and require significant computational power, which can be a drawback for real-time applications and simulations. The heavy computational requirements make DA less suitable for systems where computational resources are limited.

1.3.3 Limitations of Conventional Modeling Methods

The conventional modeling methods previously discussed are mostly model-based (also known as “white-box”), meaning they require detailed knowledge of the structure and parameters of PECs (Padhee et al., 2016). This limitation becomes utterly important for modeling commercial PECs since detailed information about their internal parameters and structures is often not readily available. Manufacturers often provide general datasheets with limited data, focusing on external characteristics rather than internal configurations.

Moreover, these conventional methods may struggle to accurately model the complex dynamic responses of PECs due to their inherently non-linear and often time-varying behavior (Guarderas et al., 2017). The computational burden associated with methods like GSSA and DA further limits their feasibility in real-time applications or when dealing with converters comprising a substantial number of non-linear elements (Guarderas, Frances, Ramirez, Asensi, & Uceda, 2019).

Therefore, there is a need for alternative modeling methods that can effectively model PECs without requiring detailed internal knowledge and can handle their non-linear, time-varying characteristics. This need has led the literature to the development of black-box modeling methods and advanced techniques based on machine learning, which we will discuss in the following sections.

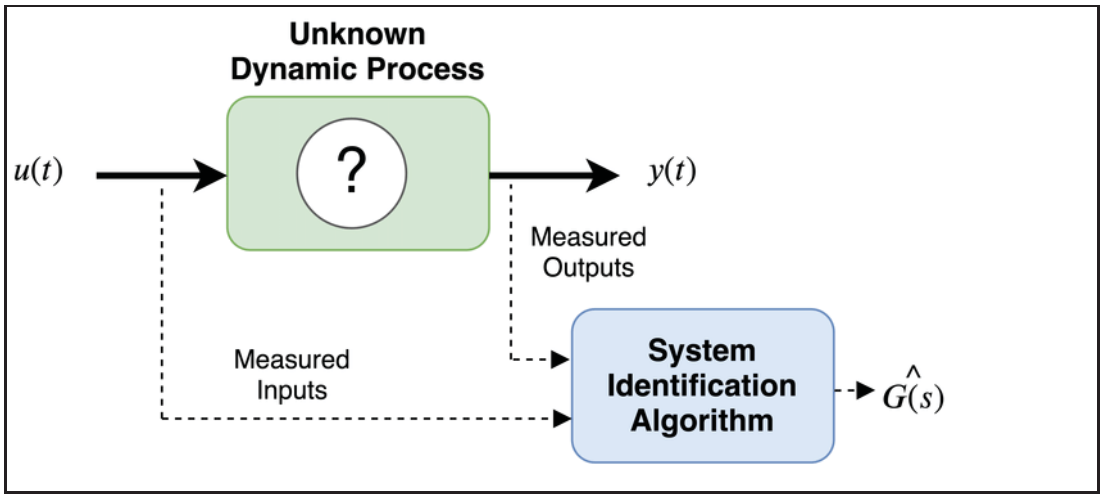

Figure 1.2 Diagram of data-driven system identification algorithms
Taken from Guruwacharya et al. (2020, p. 3)

1.4 Black-Box Modeling Methods

As we discussed earlier, in practice datasheets of commercial off-the-shelf converters are often either not readily available or provide limited information about their dynamics and internal structure. Black-box modeling methods have become popular in the literature for mitigating this very problem. The block diagram of a typical data-driven system identification method is illustrated in Figure 1.2. Black-box modeling treats the entire PEC as an unknown system, focusing exclusively on measurable input-output interactions without requiring detailed knowledge of the internal structure and dynamics (Naziris et al., 2018).

One simple black-box method is the "two-port" method, which generates a Thevenin/Norton equivalent circuit of a PEC based on measurements at the input and output terminals (Chou, Wang, & Blaabjerg, 2020). An example of a typical two-port black-box method is illustrated in Figure 1.3. While this method is fairly straightforward and suitable for simpler linear converters, it cannot model the complex dynamic behavior of higher-order converters (L. Wan et al., 2021).

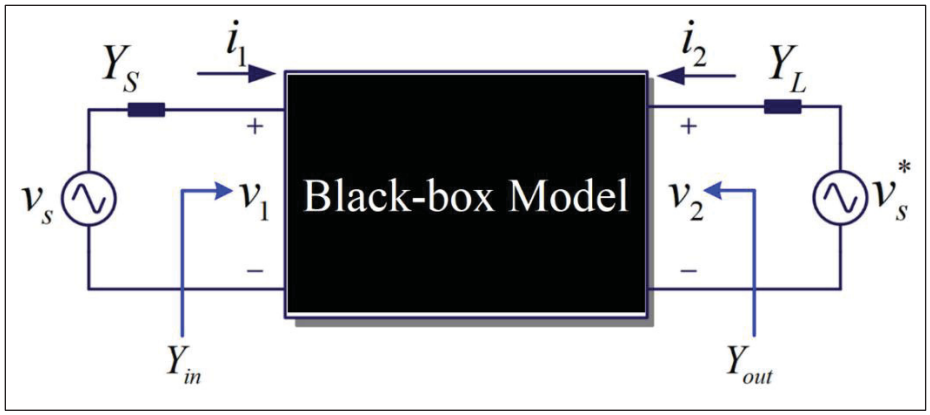

Figure 1.3 A typical two-port black-box modeling method
Adapted from (Chou et al., 2020)Chou et al. (2020, p. 5)

Several improved versions of this method have been proposed in an attempt to utilize the same principles for more complex systems by separating static and dynamic parts of PECs. However, these methods still require knowledge about the structure of the converters.

1.4.1 Black-Box G-Parameters Modeling

The G-parameters modeling method (GP) is an analytical small-signal modeling approach that extracts the Thevenin/Norton equivalent circuit of a PEC based on measurements of its input and output voltages and currents (Guarderas et al., 2019). GP provides a means to model the converter without detailed knowledge of its internal structure by focusing on the external characteristics. Figure 1.4 demonstrates a G-Parameters small-signal equivalent circuit and its equivalent block diagram. While GP is useful for linear systems, it may not accurately extract the non-linear dynamics and time-varying behavior of complex systems, which results in its limited applicability to modern PECs.

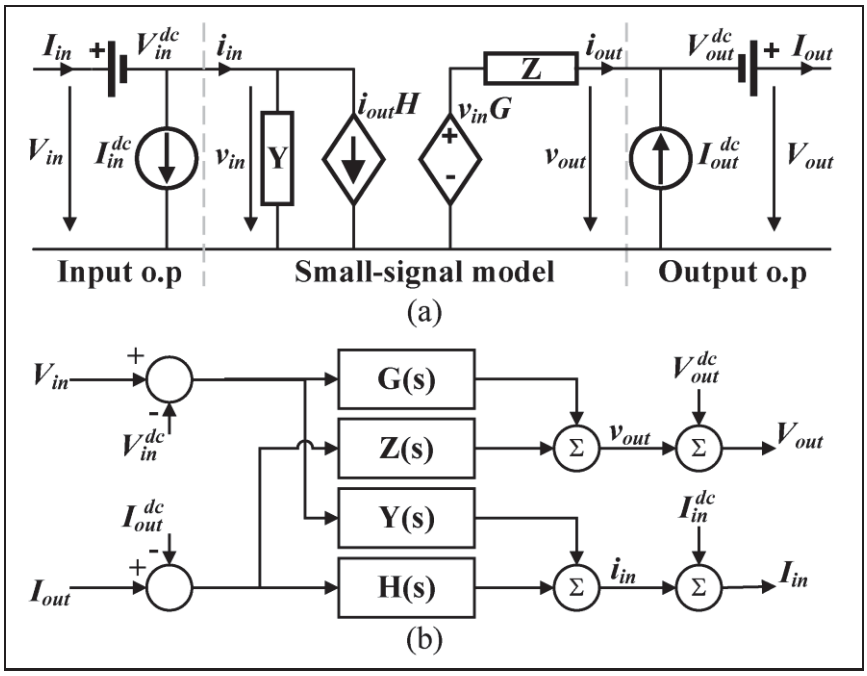

Figure 1.4 G-Parameters small-signal equivalent circuit (a) and equivalent block diagram (b)
Taken from Guarderas et al. (2019, p. 4)

One of the simplest black-box modeling methods is the Black-Box G-Parameters (BB-GP) method (Cvetkovic, Boroyevich, Mattavelli, Lee, & Dong, 2011). This method is a dual of the conventional G-parameters method, with the key difference being that in the BB-GP method, the Thevenin/Norton equivalent of the PEC is obtained via actual measurements of input and output voltages and currents, rather than analytical calculations. Hence the term, “Black-Box” is assigned to it.

The BB-GP method generates an equivalent circuit model based on several measurements performed at the input and output terminals of the converter. While this method is quite effective for simpler converters, it is incapable of modeling the complex dynamic behavior of higher-order converters due to its inability to capture non-linearities and time-varying characteristics inherent in such systems (Chou et al., 2020).

1.4.2 Wiener-Hammerstein Method

The Wiener-Hammerstein method is a popular modeling technique used for PECs which is a blend of white-box and black-box approaches, also known as “grey-box” methods (Oliver et al., 2009). This method is based on the separation of static non-linearities and linear dynamics within the system. By modeling the static non-linear components and linear dynamic components separately and then combining them, the Wiener-Hammerstein method can effectively capture the behavior of non-linear systems. An illustration of the Wiener and Hammerstein methods is depicted in Figure 1.5.

However, this method may require complex identification procedures and extensive data to accurately model the separate components. Additionally, it might not scale well with the increasing complexity of modern PECs, which often exhibit highly non-linear and time-varying behaviors.

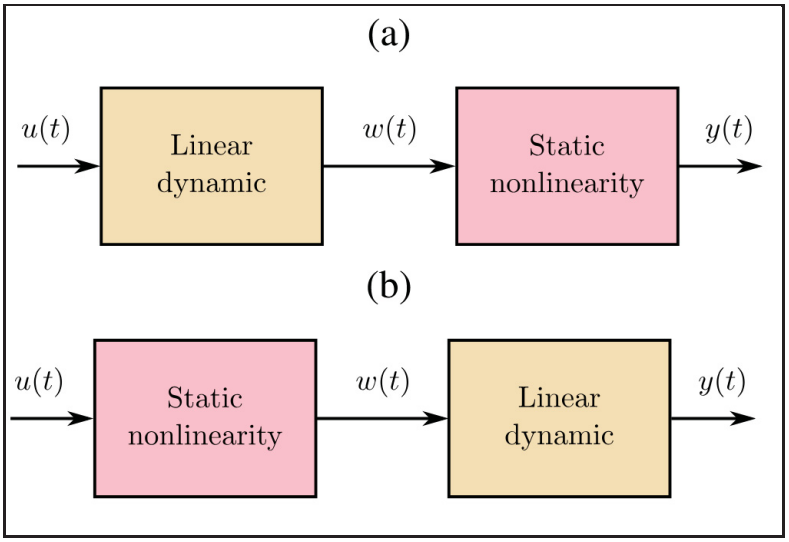

Figure 1.5 Block diagram of a) Wiener model and b) Hammerstein model
Taken from Frances et al. (2017, p. 8)

1.4.3 Polytopic Black-Box Modeling

The polytopic modeling method is a popular large-signal modeling approach for PECs (Arnedo, Boroyevich, Burgos, & Wang, 2008; A. Francés, Asensi, García, Prieto, & Uceda, 2016; Airán Francés, Asensi, & Uceda, 2019). This method is based on obtaining small-signal models of a PEC at different operating points across its operating range. These models are then connected via weighting functions to approximate the converter's non-linear dynamic behavior throughout its entire operating region. As depicted in Figure 1.6, the polytopic method effectively captures the non-linearity and time-varying characteristics of PECs by considering multiple linear models and their transitions.

However, the conventional state-space-based polytopic method is model-based and requires detailed knowledge of the converter's internal structure and parameters at various operating points. As we discussed earlier, such detailed information is often unavailable for commercial off-the-shelf converters, making it challenging to develop accurate models using this method.

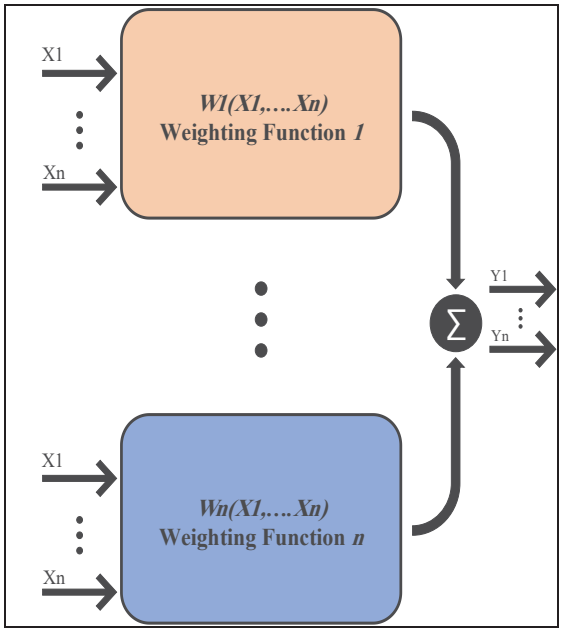

Figure 1.6 Diagram of the polytopic modeling method
Taken from Qashqai et al. (2020, p. 2)

In the black-box polytopic approach, small-signal black-box models of PECs are obtained using measurements at various operational points, and using weighting functions, the large-signal non-linear dynamics of PECs are approximated. Although this model obtains an effective approximation of the non-linear behaviors of PECs, the complexity of this method grows exponentially for complex converters.

1.5 AI-Based Modeling of Power Electronic Converters

Artificial Intelligence (AI) has become substantially popular in various research areas in the past decades due to the remarkable breakthroughs in AI-based technologies and exponential growth of computational capacities. AI refers to various methods that enable machines to mimic human intelligence, allowing them to perform complex tasks even under uncertainty or incomplete data due to their generalization capabilities (Kůrková, Manolopoulos, Hammer, Iliadis, & Maglogiannis, 2018). In power electronics, AI offers promising solutions to the challenges posed by the non-linear and time-varying behavior of power electronic converters (PECs) (W. Hu, Shi, & Borst, 2020). Recent studies have increased the role of AI in power electronics, particularly in optimizing converter design and enhancing system stability. For instance, (Gros, Lü, Oprea, Lu, & Pintilie, 2023) explores AI-based optimization techniques aimed to improve the performance and stability of power electronic converters. These methods offer deeper insights into the dynamic behavior of PECs under various operational conditions.

Traditional modeling methods often rely on detailed knowledge of the converter's internal structure and are based on linear approximations, which may not accurately capture the complex dynamics of modern PECs. As we discussed earlier, data-driven black-box methods are proposed in the literature to address this issue. AI-based modeling approaches are inherently data-driven alternatives that enable the development of accurate models based merely on input-output measurements. For instance, (S. Zhao, Peng, Zhang, & Wang, 2022) propose a physics-informed machine learning (PIML) framework for parameter estimation in power electronic converters, integrating deep neural networks with dynamic Buck converter

models to address challenges in data scarcity and robustness inherent in purely data-driven approaches. Their method demonstrates enhanced accuracy and reduced dependency on large datasets, offering a scalable solution for condition monitoring and adaptive control in power electronics systems.

1.5.1 Feedforward Artificial Neural Networks (FF-ANN)

Artificial Neural Networks (ANNs) are computational models inspired by the human brain's network of neurons, capable of learning patterns from data (Kůrková et al., 2018). In the context of power electronics, Feedforward Artificial Neural Networks (FF-ANNs) have been used for black-box modeling of PECs. FF-ANNs learn the relationships between inputs and outputs by adjusting the weights of the network during training (Liao et al., 2023; Schmidhuber, 2015). The structure of a typical FF-ANN is illustrated in Figure 1.7.

FF-ANNs are relatively simple to implement and require less computational power, making them attractive for modeling non-linear systems (Bose, 2007). FF-ANNs can be used to model the small-signal behavior of PECs, capturing steady-state responses effectively. However, due to the lack of memory elements, FF-ANNs struggle to model large-signal behavior and dynamic responses that depend on past inputs. This is due to the lack of memory blocks which results in these structures without the capability to "remember" previous states. This drawback limits their ability to capture temporal dependencies inherent in PECs.

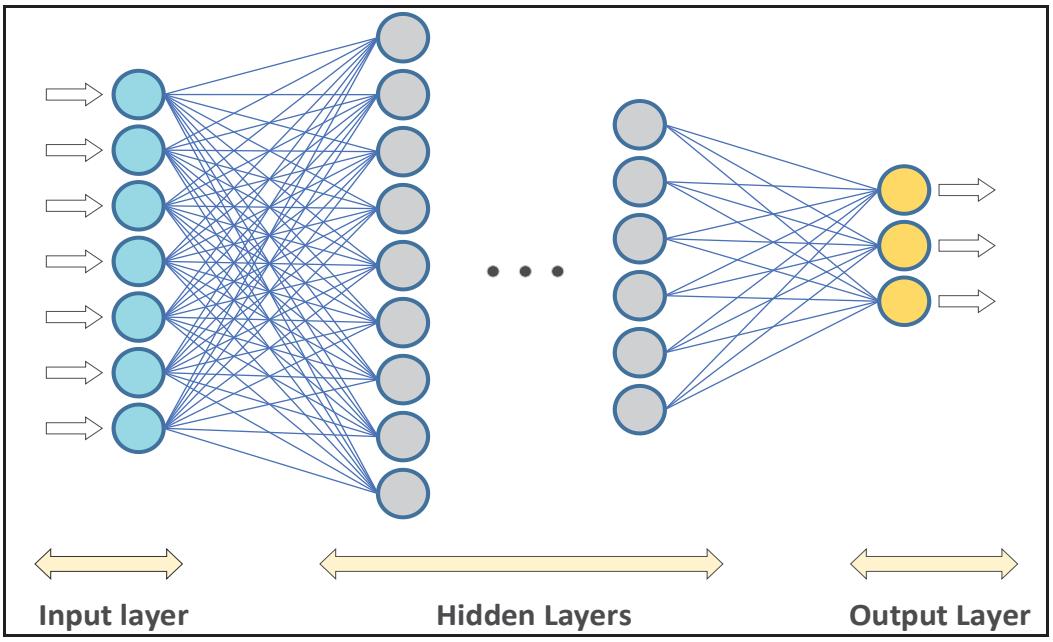

Figure 1.7 Structure of a typical artificial neural network (ANN)
Taken from Qashqai et al. (2020, p. 2)

1.5.2 Recurrent Neural Networks (RNN)

To overcome the limitations of FF-ANNs, Recurrent Neural Networks (RNNs) were introduced. RNNs incorporate feedback loops, allowing information to persist over time and enabling the network to capture temporal dynamics (Schmidhuber, 2015). In RNNs, the output from the previous step is fed back into the network along with the new input, providing a form of memory as illustrated in Figure 1.8.

Conventional RNNs, also known as Vanilla RNNs (VRNNs) in the industry, have been applied to model the large-signal behavior of PECs using a black-box approach (Rojas-Dueñas et al., 2020, 2020). VRNNs can capture short-term dependencies between input and output sequences, improving modeling accuracy over FF-ANNs.

However, VRNNs suffer from the "vanishing gradient" and "exploding gradient" problems during training, which limits their ability to learn long-term dependencies (Bengio, Boulanger-

Lewandowski, & Pascanu, 2013; Pascanu, Mikolov, & Bengio, 2013; Tan & Lim, 2019). As a result, VRNNs may not accurately model the transient responses of PECs, particularly when long-term memory is required.

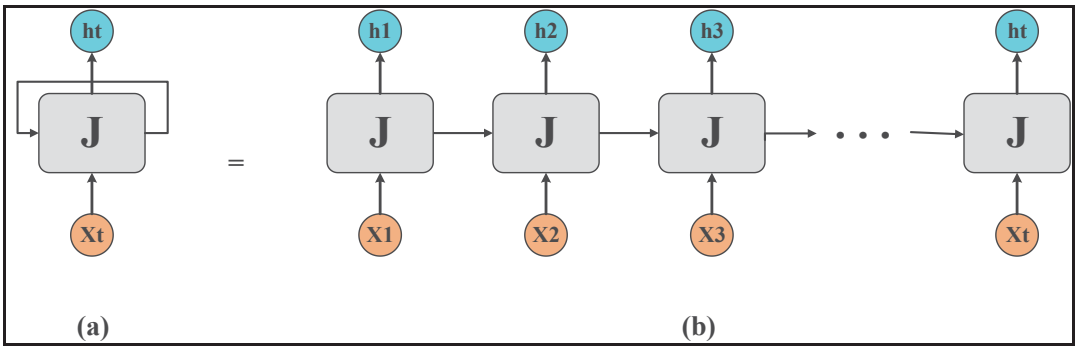

Figure 1.8 A neuron in Recurrent Neural Networks (a); equivalent cascading neurons in FF-ANNs (b)
Taken from Qashqai et al. (2020, p. 2)

1.5.3 Long Short-Term Memory (LSTM) Networks

Long Short-Term Memory (LSTM) networks are advanced recurrent neural networks designed to address the limitations of conventional VRNNs by introducing memory cells and gating mechanisms (Hochreiter & Schmidhuber, 1997a). LSTMs have the ability to learn long-term dependencies by controlling the flow of information through input, output, and forget gates.

The structure of an LSTM unit is illustrated in Figure 1.9. LSTM units are essentially extended variations of VRNNs. These units utilize different gates for different data to disregard the negligible dependencies while retaining the vital ones. Doing this allows LSTM to mostly address the “vanishing gradient” problem of the conventional VRNNs while learning long-term temporal dependencies (Sherstinsky, 2020). These advantages make them ideal for the black-box modeling technique proposed in this thesis. The detailed structure of LSTM units and their inherent differences compared to VRNN and FF-ANN will be discussed in CHAPTER 5.

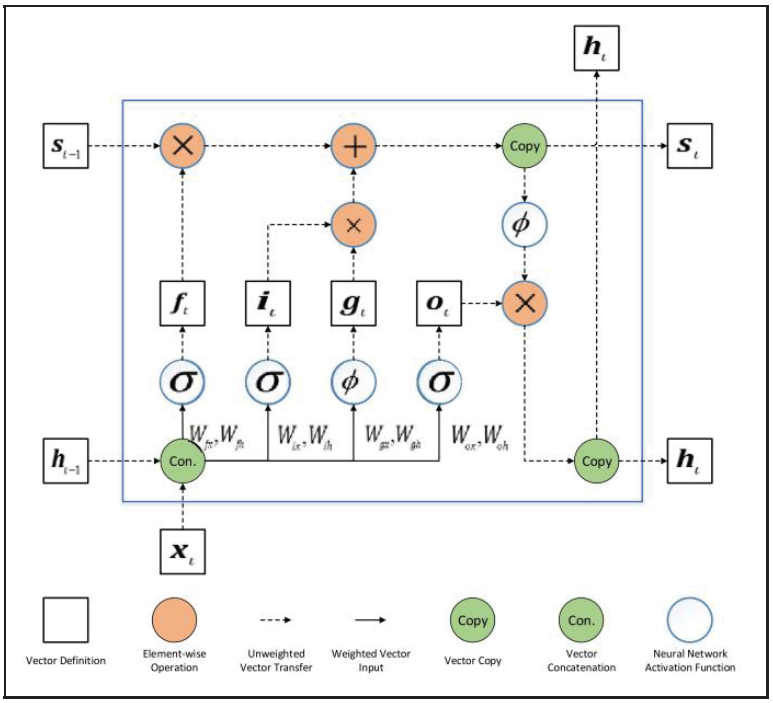

Figure 1.9 Structure of a long short-term memory (LSTM) unit
Taken from Kong et al. (2019, p. 5)

1.5.4 Gated Recurrent Units (GRU)

Gated Recurrent Units (GRUs) were proposed by (Chung, Gulcehre, Cho, & Bengio, 2014a) as a simpler alternative to LSTMs. GRUs are simplified alternatives of LSTMs that combine the input and forget gates into a single update gate and merge the cell state and hidden state. This results in fewer parameters and a less complex structure compared to LSTMs. A simplified representation of the structural differences between GRU and LSTM is depicted in Figure 1.10.

We will discuss and explain these architectural differences in detail in CHAPTER 5.

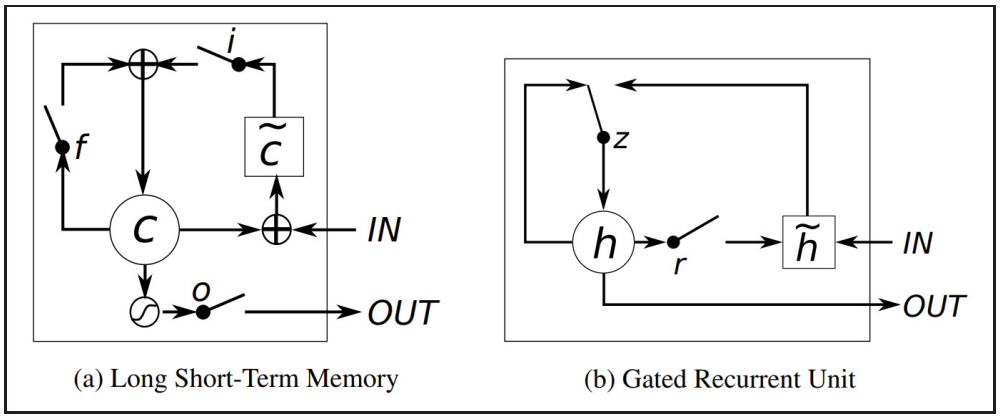

Figure 1.10 Simplified Illustration of an (a) LSTM unit versus (b) a GRU unit
Taken from Chung et al. (2014, p. 3)

1.5.5 Transfer Learning in PEC Modeling

While advanced recurrent networks like LSTM and GRU improve modeling accuracy, they often require long training times, especially for complex PECs. Transfer Learning (TL) is a technique used to utilize the knowledge gained from one task to improve learning in a related task (Weiss, Khoshgoftaar, & Wang, 2016). By transferring learned features from a pre-trained model to a new model, TL can reduce training times and improve performance.

In the context of PEC modeling, by pre-training the neural network on a similar converter or operating condition, they can reduce the training time for the target converter significantly. This approach makes advanced neural network models more practical for real-time applications and complex PECs. We will utilize transfer learning to enhance the training efficiency of LSTM and GRU-based black-box models later in CHAPTER 5.

1.5.6 Conclusion

The integration of AI into the modeling of power electronic converters represents a significant advancement in addressing the complexities of modern PECs. AI-based modeling methods, particularly those utilizing advanced neural networks like LSTM and GRU, overcome the

limitations of conventional methods by effectively capturing non-linear, time-varying, and dynamic behaviors without requiring detailed internal knowledge of the converters.

AI-based modeling methods offer several advantages over traditional modeling approaches. First of all, these methods are data-driven in nature. AI techniques learn directly from input-output data, thus eliminating the need for detailed knowledge of the converter's internal structure. This is especially useful for commercial off-the-shelf converters where internal details are proprietary.

Secondly, some of these methods are inherently compatible with non-linearities and time-variant behaviors. Therefore, advanced neural networks like LSTM and GRU can model non-linear and time-varying behaviors of modern complex PECs effectively and extract both short-term and long-term temporal dependencies.

Additionally, these models can be retrained or fine-tuned for different converters or operating conditions using transfer learning.

However, despite their advantages, AI-based models present a significant drawback which is the lack of formal mathematical frameworks to analyze system stability. Unlike traditional modeling approaches, which allow for well-defined stability analysis using control theory principles, AI models are typically treated as black-box systems, making it difficult to establish analytical stability studies for them. This limitation poses challenges in critical applications where stability assurances are necessary for regulatory compliance and safe operation. Addressing this issue requires further research into explainable AI techniques and hybrid modeling approaches that combine AI with conventional analytical methods.

In conclusion, As the demand for efficient and reliable PECs continues to grow, especially with the rise of electric vehicles and renewable energy systems, AI-based modeling will play a crucial role in progressing the performance and integration of these technologies into modern

power grids. In the next section, we will investigate the integration of AI-based methods for intelligent control of PECs.

1.6 Conventional Control Methods for Power Electronic Converters

In this section of the literature review, we will first introduce the most prominent conventional control methods for PECs, highlighting their advantages and limitations. Next, we will discuss advanced control methods, with a particular focus on AI-based intelligent control techniques. Finally, we will explore deep reinforcement learning and its application in power electronics and power systems, emphasizing its advantages for controlling complex PECs, especially when compared to other nonlinear control methods, such as model predictive control (MPC). Based on this analysis, we will conclude why we have chosen to propose a novel control method for PECs using DRL.

1.6.1 Linear Control Methods

Linear control methods, such as Proportional-Integral-Derivative (PID) controllers, have been the classic control strategies in power electronics due to their simplicity and effectiveness in linear systems (Bacha et al., 2014). PID controllers offer straightforward implementation and robust performance for simpler systems with predictable behavior. These controllers are based on linear approximations of converters, hence they often utilize linearization techniques like averaging and state-space modeling as discussed in the previous sections.

However, the inherent non-linear and time-varying nature of modern PECs imposes significant challenges for linear control methods such as PID. Due to their small-signal nature, linear controllers like PID are inadequate for handling highly non-linear operational points, uncertainties, and large disturbances. As PECs evolve into more complex and highly non-linear systems, the limitations of linear control methods become more evident. Some solutions have been proposed in the literature to partially mitigate these disadvantages by using multiple

controllers to tackle different operating modes (Xu et al., 2020). However, non-linear control methods proposed in the literature are far more effective.

1.6.2 Non-Linear Control Methods

To address the aforementioned limitations of linear control, non-linear control methods have been introduced in the literature. These methods are proposed to handle the inherent non-linearities and dynamic behaviors in PECs which makes them present reliable and robust performance across different operating modes. The most prominent examples are as follows.

1.6.2.1 Feedback Linearization

Feedback linearization transforms a non-linear system into an equivalent linear system and thus, it simplifies the control design by allowing the use of conventional linear control methods (Mahmud, Roy, Saha, Haque, & Pota, 2019). In the context of PECs, feedback linearization tries to linearize the non-linearities of the converters by applying a control input derived from the PECs inverse model.

However, this method requires an accurate model of the converter as well as its inverse, which may be difficult to extract for more complex converters. Modeling unknown internal factors such as parasitic elements is also challenging and may lead to performance degradation. Additionally, feedback linearization is sensitive to sudden and large disturbances, as well as noise in measurement signals, which can impact the stability and effectiveness of this control method.

1.6.2.2 Sliding Mode Control (SMC)

Sliding Mode Control (SMC) is a robust non-linear control method that forces the converter to reach and stay on a predetermined sliding surface thus ensuring the desired system behavior despite the presence of uncertainties and disturbances (Komurcugil, Biricik, Bayhan, & Zhang,

2020; Sebaaly, Vahedi, Kanaan, Moubayed, & Al-Haddad, 2016). The SMC method is resilient to external disturbances and internal parameter variations, making it suitable for controlling PECs under challenging conditions.

However, SMC introduces high switching frequencies to the switches, which can result in increased switching losses, reduced efficiency, and shorter lifespan of the switches. This control method also requires a high-gain controller which makes it sensitive to noise in measurements. These drawbacks limit the applicability of SMC in systems where efficiency and switch longevity are critical considerations or there is substantial noise in the environment (Komurcugil et al., 2020).

That being said, recent advancements in higher-order SMC algorithms, such as the super-twisting algorithm (STA) and dynamic sliding mode techniques, minimize chattering through smoother control actions and reduced high-frequency switching. These methods generate continuous control inputs for PWM signals, improving stability and accuracy while constraining noise and disturbances within defined bounds. Although chattering in these methods is significantly reduced, residual chattering may still cause mechanical wear and energy losses in power converters controlled by these agorithms (Komurcugil, Biricik, & Babaei, 2019).

1.6.2.3 Model Predictive Control (MPC)

Model Predictive Control (MPC) is an advanced control method that utilizes an explicit model of the system to predict future behavior and decide what are the optimal actions in each time step, accordingly (J. Hu et al., 2021; Rodriguez, Garcia, Mora, Davari, et al., 2021; Rodriguez, Garcia, Mora, Flores-Bahamonde, et al., 2021). MPC addresses non-linearities while considering constraints by solving an online optimization problem at each time step. MPC in PECs, selects the switching actions that minimize a predefined cost function over the projected timsteps also known as the “prediction horizon”. An example of MPC utilized to control a grid-connected HPUC is illustrated in Figure 1.11. As you can see, the cost function is

optimized at every timestep considering the constraints of optimization, current observations, and future costs during the prediction horizon.

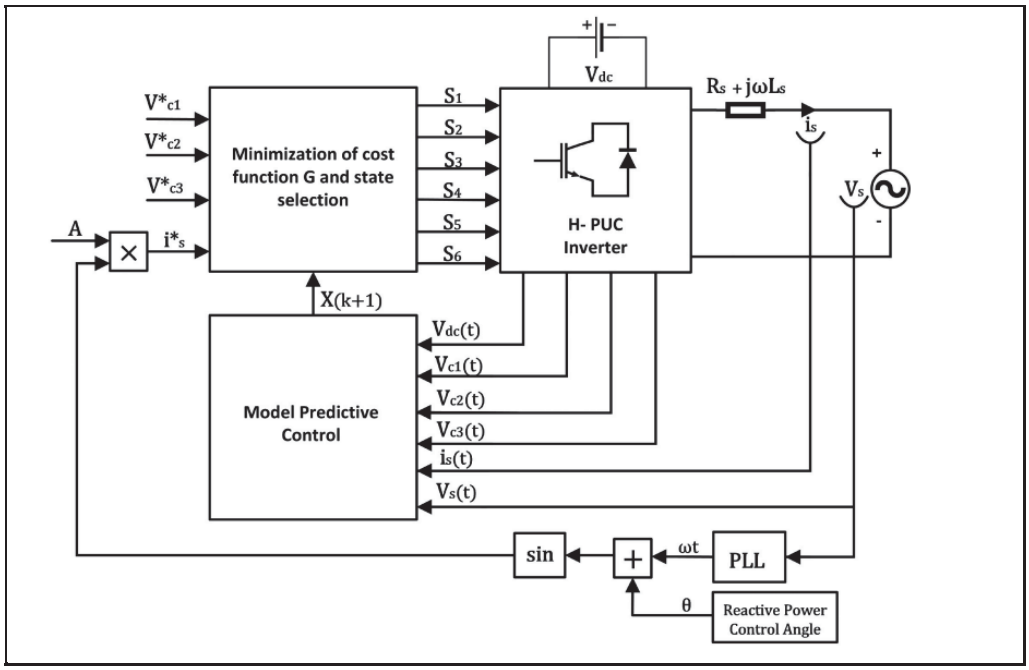

Figure 1.11 Block diagram of model predictive control (MPC) utilized to control a grid-connected HPUC
Adapted from Sorto-Ventura et al. (2020, p. 5)

In the control of PECs, MPC offers several advantages such as the ability to handle multi-variable systems, address constraints, and predict the future cost of actions. However, MPC requires an accurate model of the converter and the load, which may be challenging to obtain in practice, especially for complex or time-varying systems. Moreover, MPC demands high computational power and fast processing to solve the optimization problem at every time step, which can be a significant challenge for complex PECs (Metri et al., 2016). Although several techniques, such as explicit MPC, real-time optimization algorithms, and parallel computing, have been introduced to reduce the computational burden, the processing burden remains significant, especially for systems with high switching frequencies and multi-variable constraints. Compared to AI-based control methods, particularly deep reinforcement learning (DRL) which is the focus of this thesis, MPC's computational burden can still be considerable

during execution. While AI methods require substantial computational resources during the training phase, their execution is typically less demanding compared to the online and ongoing optimization required by MPC. Despite these advancements, the choice between MPC and AI-based control is a trade-off between computational efficiency, accuracy, and adaptability.

Most importantly, MPC is sensitive to parameter mismatch and variations in system dynamics (Mohammad Babaie, Sharifzadeh, Mehrasa, Chouinard, & Al-Haddad, 2021, 2020). Any mismatch between the model and the actual system can degrade performance and potentially lead to instability. Since MPC relies on accurate modeling of the converter, uncertainties and large disturbances can also significantly impact its effectiveness to the point of instability and collapse (Metri et al., 2016).

1.6.3 Limitations of Conventional Control Methods

While conventional control methods—both linear and non-linear—have been successfully applied to various PEC topologies, they have limitations especially when dealing with the complexities and uncertainties of modern converters. Linear control methods are inadequate for highly non-linear systems and cannot effectively handle uncertainties and disturbances. Non-linear control methods like feedback linearization and SMC on the other hand can achieve satisfactory performance in the presence of nonlinearities but they require accurate system models and may introduce undesirable effects such as high switching frequencies and sensitivity to measurement noise.

Moreover, model-based control strategies, including MPC despite being effective for complex converter topologies with demanding control, rely heavily on accurate models and require minimal parameter mismatch and noise for adequate control of complex PECs. The performance of these methods can degrade due to accumulative errors between the model and the actual behavior of the converter.

These challenges highlight the need for alternative control approaches that can handle non-linearities, uncertainties, and time-varying dynamics without relying on precise models or imposing excessive computational demands. The emergence of intelligent control methods, such as those based on artificial intelligence and machine learning seem promising solutions to address these limitations, which will be discussed in the next subsection.

1.7 Adaptive and Intelligent Control Techniques

To overcome the aforementioned limitations of conventional methods, various intelligent control techniques have been introduced. Methods such as particle swarm optimization (PSO) and fuzzy neural networks (FNN) have been employed, either individually or in combination, to enhance control performance in power systems (Agrawal, Panigrahi, & Tiwari, 2008).

Particle Swarm Optimization (PSO) is an optimization algorithm inspired by the social behavior of animals that demonstrate swarm behaviors such as birds or fish. While PSO is not a standalone control method, it is used as an optimization tool to enhance the performance of existing controllers. For instance, it is utilized in power electronic converter (PEC) applications to optimize control parameters, such as tuning PID gains or optimizing the cost functions of model-based controllers. However despite its advantages, PSO suffers from slow convergence and may become fixed in a local optimum. These issues limit its effectiveness in control applications where rapid adaptation is required (Agrawal et al., 2008). These limitations suggest that while PSO can be effective for offline optimization, its applicability in real-time adaptive control is restricted without additional strategies such as hybrid approaches that combine PSO with other adaptive control techniques.

Fuzzy Neural Networks (FNN) combine fuzzy logic and neural networks to handle uncertainties and non-linearities in PECs. FNNs can control complex systems without needing precise mathematical models by using humanly-readable fuzzy rules (Belaidi, Haddouche, & Guendouz, 2012; M. Wang & Wang, 2020). Despite their capabilities, FNNs are similar to ANNs in terms of facing challenges related to computational complexity, especially when

dealing with high-dimensional input spaces or large datasets. While FNNs introduce additional rule-based processing alongside neural computations, their complexity may not necessarily be significantly higher compared to ANNs, as both methods require optimization and tuning for efficient operation (M. Wang & Wang, 2020). Moreover, overfitting is a common issue for both FNNs and ANNs, demanding strategies such as regularization, pruning, and proper data preprocessing to enhance generalization and avoid poor performance in real-world applications..

These intelligent control methods, while offering improvements over traditional approaches, still face challenges such as limited adaptability and learning capacity. Moreover, they may not fully handle the complexities of modern PECs, which require control strategies capable of handling significant non-linearities, uncertainties, and time-varying behaviors. As a result, more advanced control techniques are emerged. Machine learning-based controls and particularly, Deep reinforcement learning has been introduced as a suitable solution to these aforementioned drawbacks.

1.8 Deep Reinforcement Learning in Advanced Non-Linear Control

Deep Reinforcement Learning (DRL) is a subset of machine learning that combines reinforcement learning principles with deep neural networks (Yuxi Li, 2018). In DRL, an agent learns the optimal actions aiming to maximize a cumulative reward over time merely by interacting with an environment. The deep neural networks enable the agent to handle high-dimensional input spaces and learn complex policies that map states to optimal actions (LeCun, Bengio, & Hinton, 2015).

Recent breakthroughs in Artificial Intelligence (AI), especially in Deep Learning, in addition to the increased computational power, have increased the popularity of DRL in various complex applications, including but not limited to robotics (Ibarz et al., 2021; Nguyen & La, 2019), electric vehicles (Du et al., 2020; X. Hu, Liu, Qi, & Barth, 2019; Yanni Wan, Qin, Ma, Fu, & Wang, 2022), renewable energy systems (H. Wang, Lei, Zhang, Zhou, & Peng, 2019;

Z. Zhang, Zhang, & Qiu, 2019), and various aspects of power systems (Cao et al., 2020a; Z. Zhang et al., 2019).

In the context of power electronic converters (PECs), certain DRL methods offer a model-free approach to control, allowing the agent to learn optimal control policies without requiring an explicit mathematical model of the converter. This is especially valuable for the non-linearities, uncertainties, and time-varying dynamics inherent in modern PECs. However, the accuracy of the trained DRL model depends heavily on the quality of the reference data used during the training process. If real-world data is employed as a reference, mismatches and tolerances between different system instances can introduce discrepancies that affect the generalization of the trained model across identical systems.

To address this challenge, several strategies have been proposed in the literature, including transfer learning, domain randomization, hybrid training approaches, and adaptive learning techniques. By leveraging transfer learning, the initial DRL model is trained on generalized datasets and later fine-tuned to specific systems, improving adaptability. Domain randomization techniques are employed to expose the DRL agent to a variety of operating conditions and parameter variations, enhancing its robustness. Additionally, a hybrid training approach is utilized, where the DRL agent is first trained in a high-fidelity simulation environment before being fine-tuned with real-world data to minimize discrepancies.

1.8.1 DRL Applications in Power Electronics

DRL has been applied to various power electronics applications to mitigate the problems that conventional control methods struggle with, problems such as handling non-linearities, parameter variations, and external disturbances.

For example, (Cui, Dong, Dong, Zhang, & Ghias, 2024) propose an adaptive horizon seeking strategy for generalized predictive control using deep reinforcement learning that improves DC/DC converter performance by dynamically changing prediction horizons to speed up

convergence and boost robustness under different operating conditions. Relatedly, (Zeng et al., 2024) develop a deep reinforcement learning–based distributed control framework for DC solid-state transformers in microgrids, allowing multiple agents to work together effectively to keep the grid stable and improve efficiency despite changing renewable energy inputs. Liu et al. Moreover, (X. Liu et al., 2024) propose an online reinforcement learning-based finite control-set model predictive control framework for voltage source inverters that robustly handles system uncertainties and lowers computational demands while ensuring fast response times in real-time settings. In addition, (Jiang, Zeng, Zhu, Pou, & Konstantinou, 2023) present a stability-focused, multiobjective control design for power converters that uses deep reinforcement learning to optimize transient response, steady-state performance, and resistance to load disturbances all at once. Lastly, (Cui, Yang, Dai, Zhang, & Xu, 2023) demonstrate a transfer reinforcement learning approach for DC–DC buck converter control via duty ratio mapping, which bridges the gap between simulation and real-world use by adapting pre-trained neural network policies to actual operating conditions. Overall, these studies show that deep reinforcement learning can greatly improve control systems by making them faster, more robust, and more adaptable to real-world challenges in power electronics.

In an interesting approach, (Tian, Cobaleda, & Martinez, 2022) proposed a DRL-based method as illustrated in Figure 1.12, for parameter optimization in a DC-DC converter. The proposed novel control scheme in this paper was shown to achieve optimal control and voltage ripples by applying a deep Q-network (DQN). This method demonstrates DRL’s ability to adaptively learn optimal control policies without vast prior training data. This advantage makes DRL a promising approach for optimal parameter fine-tuning of power electronics in dynamic environments where conventional approaches may struggle.

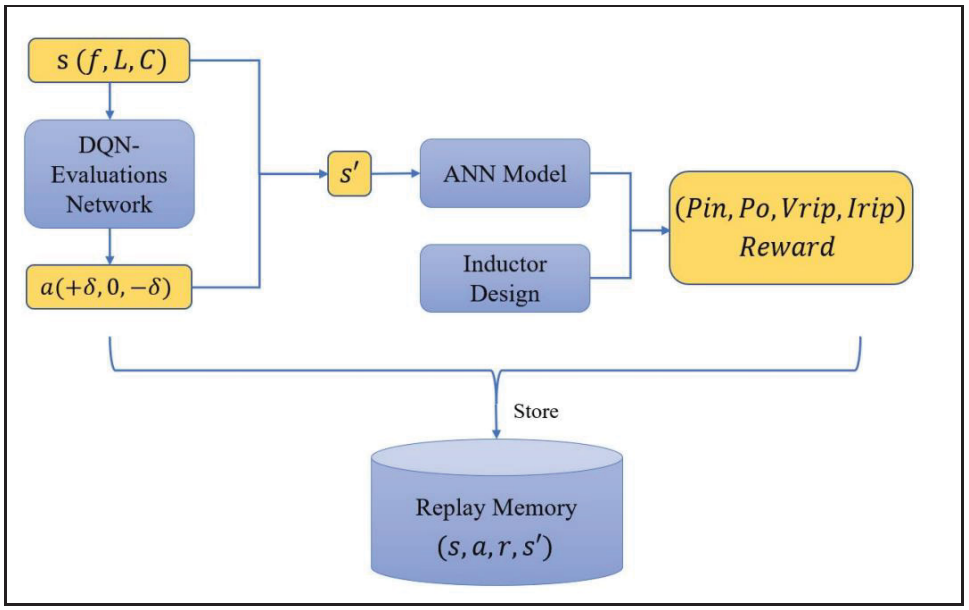

Figure 1.12 Block diagram of parameter optimization approach based on Deep Q-learning

Taken from Tian et al. (2022, p. 5)

In the control of DC-DC converters, especially buck converters feeding constant power loads (CPLs), DRL has demonstrated substantial capability. Conventional control methods often exhibit poor performance in the presence of large changes in CPLs due to the non-linear and destabilizing effect of CPLs on the system (Hajihosseini, Andalibi, Gheisarnejad, Farsizadeh, & Khooban, 2020).

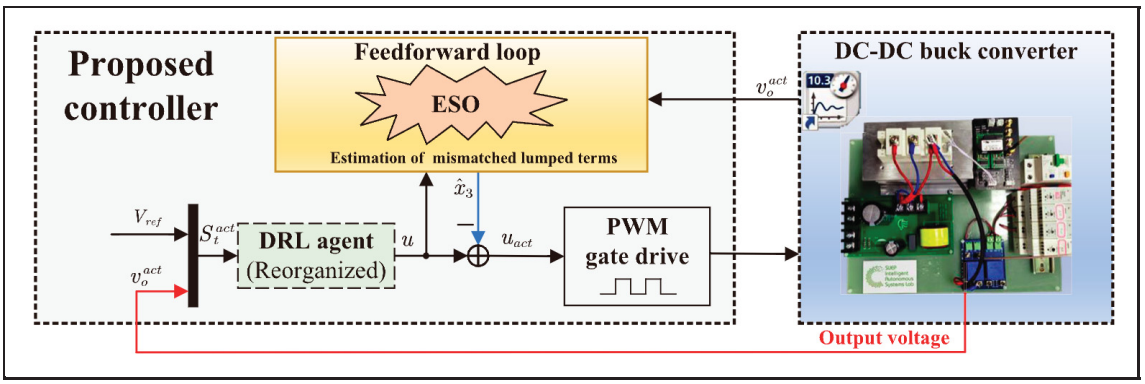

Figure 1.13 Structure of the control strategy for a buck converter feeding a CPL based on a DRL agent

Taken from Yang et al. (2023, p. 3)

In (Hajihosseini et al., 2020), a DRL agent was applied to develop an advanced control method for a DC-DC buck converter feeding a CPL. The DRL agent was used to tune the gains in a feedback loop, improving the converter's performance under varying load conditions. However, this approach did not fully take advantage of the model-free approach provided by DRL, because it still relied on auxiliary control circuits.

(Tianxiao Yang, Cui, & Zhang, 2022) on the other hand, proposed a model-free DRL control method for a DC-DC buck converter feeding CPLs. Their method demonstrated strong dynamic performance despite relatively large changes in the CPL. Unlike the method proposed by (Hajihosseini et al., 2020), the DRL agent in this paper learned the optimal control policy through interactions with the converter, without requiring an explicit model. However, the controller required accurate measurements, making it sensitive to noise and accumulative errors between the trained model and the actual converter.

Consequently, (Chuanlin Zhang, Chengang Cui Tianxiao Yang & Yang, 2023; Cui, Yan, Huangfu, Yang, & Zhang, 2022) introduced a DRL-based control method that aims to enhance resilience against uncertainties and parameter variations in DC-DC buck converters. Their method employed an extended state observer (ESO) that used an offline pre-trained model of the converter to adjust to variations by observing differences between the model and the actual converter. While being very effective, this approach requires an accurate pre-trained model. An illustration of this approach is depicted in Figure 1.13.

Multilevel converters, such as Neutral-Point-Clamped (NPC) converters, present additional challenges due to their complex switching states and the need for voltage balancing across capacitors. Traditional control methods like Model Predictive Control (MPC) while being effective, require an explicit model of the converter and are sensitive to model inaccuracies (Mohammad Babaie et al., 2021; Mohammad Babaie, Sharifzadeh, et al., 2020).

In a study (J. Wang, Yang, & Yao, 2022) explored the application of DRL in active neutral point clamped converters, focusing on efficiency optimization of a three-level ANPC inverter.

The proposed method adapts to changing design requirements and rapidly generates optimal parameters to minimize power losses.

In this thesis, DRL is used to achieve robust model-free control for a three-level grid-connected neutral point-clamped converter. This proposed method is discussed in detail in CHAPTER 2.

The Hybrid Packed U-Cell (H-PUC) converter is an advanced multilevel converter topology known for its reduced component count and ability to generate multiple voltage levels from a single DC source (Sorto-Ventura, Abarzadeh, Al-Haddad, & Dessaint, 2020). Controlling H-PUC converters is challenging due to the numerous switching states and the dependency of DC-link capacitor voltages on each other, which increases control complexity. In CHAPTER 3, we have proposed a DRL-based controller that effectively handles the non-linearities and time-varying dynamics of the H-PUC converter, offering robustness against uncertainties, noise, and parameters mismatch while being entirely model-free. This study highlighted the potential of DRL in controlling complex PEC topologies that are difficult to manage using conventional approaches. In CHAPTER 4, we performed a comparative analysis to show the superiority of the DRL method compared to MPC.

1.8.2 Conclusion

Deep Reinforcement Learning represents a significant advancement in the control of power electronic converters. By leveraging model-free, adaptive, and robust control strategies, DRL addresses many of the challenges associated with conventional control methods in handling the complexities of modern PECs. The successful application of DRL in various power system applications, especially controlling DC-DC converters and complex multi-level converters, shows a promising potential to enhance performance, efficiency, robustness, and reliability in power electronics converters.

As this literature review has demonstrated, conventional approaches face significant limitations on two fronts: modeling and control. On the control side, methods like MPC are

powerful but rely on precise mathematical models that are often unavailable and are sensitive to parameter drift and noise. On the modeling side, obtaining accurate white-box models for complex commercial converters is a challenge in itself, which limits both simulation studies and the applicability of such model-based controllers.

This thesis addresses these interconnected challenges by proposing advanced, data-driven solutions for both problems. The research is structured to tackle each aspect directly:
For the control problem, this thesis proposes and validates a purely model-free DRL approach, demonstrating its superiority over conventional model-based methods. This is explored through:

- The control of a three-level grid-connected NPC converter (CHAPTER 2).
- The extension of the DRL method to a more complex H-PUC in a grid-connected implementation (CHAPTER 3).
- A comparative analysis between DRL and MPC for the H-PUC to quantify the benefits in terms of robustness and resilience (CHAPTER 4).

For the modeling problem, this thesis develops and validates an advanced black-box modeling technique using AI to create accurate, data-driven models of PECs when a mathematical model is unknown. This contribution is detailed in:

- The creation of a black-box modeling framework using Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) networks to accurately capture converter dynamics from measurement data alone (CHAPTER 5).

The proposed DRL-based controllers effectively handle the non-linearities and time-varying dynamics of these converters without relying on a precise model. Simultaneously, the work in CHAPTER 5 on black-box modeling provides the foundational techniques for creating highly accurate converter models when one is required—for instance, for high-fidelity simulation or for designing the next generation of data-driven, model-based controllers. While pursued as

distinct research paths in this thesis, these contributions lay the groundwork for a future hybrid approach, directly addressing the critical question of whether the AI-based models developed in CHAPTER 5 could be used to build a more robust, data-driven MPC, thus bridging the gap between model-free and model-based control paradigms.

CHAPTER 2

A MODEL-FREE SWITCHING AND CONTROL METHOD FOR THREE-LEVEL NEUTRAL POINT CLAMPED CONVERTER USING DEEP REINFORCEMENT LEARNING

Pouria Qashqai [a], Mohammad Babaie [a], Rawad Zgheib [b], and Kamal Al-Haddad [a],

[a] Department of Electrical Engineering, École de Technologie Supérieure,
1100 Notre-Dame West, Montreal, Quebec, Canada H3C 1K3
[b] Institut de recherche d'Hydro-Québec (IREQ), Varennes, QC J3X 1S1, Canada

Paper published in *IEEE Access*, September 2023

Abstract

This chapter presents a novel model-free switching and control method for a three-level neutral point clamped (NPC) converter using deep reinforcement learning (DRL). Our approach targets two primary control objectives: voltage balancing and current control. In this method, voltage balancing, current control, and selection of optimal switches are achieved using a reward function which is calculated based on various signals measured as observations of the DRL agent. Since the action space is discrete, a deep Q-network (DQN) agent is utilized. DQN is used due to its capability of handling high-dimensional state spaces. In order to highlight its pros and cons, the proposed method is compared with model predictive control (MPC), which is another popular non-linear control method for power electronic converters. The proposed method is evaluated and compared with the MPC method in grid-connected mode using simulations in Matlab/Simulink. To evaluate the practical performance of the DRL method, various experimental results are obtained. The simulation and experimental results demonstrate that the proposed method effectively achieves accurate voltage balancing and ensures steady operation even in the presence of various dynamic changes, including variations in the reference currents and grid voltage. Additionally, the method successfully handles uncertainties, such as sensor measurement noise, and accommodates parameter variations, such as changes in the capacity of the DC-link capacitors and line impedance. The results

demonstrate that this method exhibits superior adaptability to real-time changes and uncertainties, delivering more robust performance compared to similar conventional methods like MPC. Thus, this method can be considered a promising approach for intelligent control of power electronic converters, especially when conventional methods such as MPC face challenges in performance and accuracy under severe parameter variations and uncertainties. The findings from this stand-alone system provide a solid foundation for future extensions to more complex operational scenarios, such as grid-tied and EV rectifying modes.

2.1 Introduction

Power electronics converters are essential components of numerous modern applications such as power supplies, renewable energies, electrical vehicles, and energy storage units (Siwakoti, Forouzesh, & Pham, 2018; G. Zhang et al., 2018). Control of power electronic converters is crucial to ensure their reliable, consistent, and efficient performance (Bacha et al., 2014).

Linear control methods for power electronics converters, such as PID, are based on linear approximations of the converters using linearization techniques such as averaging and state space modeling (Frances, Asensi, García, Prieto, & Uceda, 2017). These methods are easy to study and implement while providing sufficient performance for simpler converters with linear or weakly non-linear operating modes. However, due to the small-signal nature of these methods, they are not proper for highly non-linear operating modes, uncertainties, and disturbances (Xu et al., 2020). Some solutions are proposed in the literature to mitigate these drawbacks using several controllers to tackle different operating modes (Naziris, Guarderas, Francés, Asensi, & Uceda, 2019).

On the other hand, non-linear control methods use non-linear feedback or optimization techniques to address the non-linearity in power electronic converters and consequently provide robust performance across different operating modes. There are various non-linear control methods proposed in the literature but some of the most popular methods are feedback linearization, sliding mode control (SMC), and model predictive control (MPC).

Feedback linearization (Mahmud et al., 2019) transforms the non-linearity of converters into a linear representation so that the outcome could be controlled using conventional linear methods. This method requires an accurate model of the converter as well as its inverse model which may be difficult to extract in some complex converters. It is difficult to model unknown internal factors such as parasitic elements using this method which may lead to degradation in its performance. This method may be sensitive to sudden and large disturbances as well as noise in measurement signals.

Sliding mode control (SMC) (Komurcugil et al., 2020; Sebaaly et al., 2016) forces the converter to reach and stay on a predetermined sliding surface. This method is resilient to uncertainties, external disturbances, and internal parameter variations. However, it introduces high switching frequencies to the switches which may result in switching loss, shorter lifespan of the switches, and efficiency degradation. Additionally, it requires a high-gain controller which makes this method sensitive to noise in measurements.

Model predictive control (MPC) (J. Hu et al., 2021; Rodriguez, Garcia, Mora, Davari, et al., 2021; Rodriguez, Garcia, Mora, Flores-Bahamonde, et al., 2021) is inherently an online optimization problem that utilizes an accurate model of the converter to predict its future behavior. Then, it will be able to choose the best sets of actions from all the possible ones to aim for the highest possible optimization. Although this method can address non-linearities and constraints, it requires an accurate model of the converter and load, which may be difficult in practice. Moreover, this method demands high computational power and fast processing for solving an optimization problem at every single time stamp. Finally, since it relies on accurate modeling of the converter, this method is sensitive to parameter variations and other changes in system dynamics (Mohammad Babaie & Al-Haddad, 2021; Mohammad Babaie, Mehrasa, Sharifzadeh, & Al-Haddad, 2022). Some studies have taken advantage of machine learning for tuning MPC parameters. For instance, in (Mohammad Babaie, Sharifzadeh, et al., 2020) a supervised learning model predictive control (SLMPC) method is utilized and optimized using the artificial bee colony (ABC) algorithm, to train the weighting factors of the cost function

for controlling a three-phase NPC. This method replaces the conventional time-consuming and imprecise methods.

On top of all the challenges the aforementioned that conventional non-linear control methods are facing, they also suffer from performance degradation due to the accumulative error between a model and the actual behavior of a given power electronic converter (Darbali-Zamora & Ortiz-Rivera, 2019; Krommydas & Alexandridis, 2020). Various intelligent methods such as particle swarm optimization (PSO) and fuzzy neural network (FNN) are used alone or in conjunction with each other to mitigate these problems (Krishnamoorthy & Narayanan Aayer, 2019; Liao et al., 2023). However, these methods often suffer from weak adaptability and limited learning capacity. For instance, PSO suffers from slow convergence and local optimal (Huang, 2008), whereas FNN has high computational complexity and overfitting problems (M. Wang & Wang, 2020). Thus, more advanced non-linear control methods have gained popularity in recent years to mitigate these problems.

DRL is a subset of machine learning (ML) that utilizes deep neural networks (DNN) in reinforcement learning (RL) so that an agent learns from interactions with an environment to achieve the maximum long-term reward. In recent years, due to the breakthroughs in deep learning and the availability of high computational power, Deep reinforcement learning (DRL) has gained popularity to solve various complex problems including but not limited to robotics (Ibarz et al., 2021; Nguyen & La, 2019), electrical vehicles and hybrid vehicles (Du et al., 2020; X. Hu et al., 2019), renewable energies (H. Wang et al., 2019; Z. Zhang et al., 2019) and power systems (Cao et al., 2020a; Z. Zhang et al., 2019).

The advantages and disadvantages of the aforementioned methods are listed in Table 1. As seen, despite requiring high computational power, long training times, and large training datasets, DRL is utilized as an advanced control method for power electronic converters due to its resilience towards uncertainties, noise in measurements, and parameter variations.

Table 2.1 Overview of Non-Linear Control Methods for Power Electronic Converters

Control Method	**Advantage(s)**	**Disadvantage(s)**
Feedback Linearization	1. Handles non-linearity 2. Uses linear control techniques.	1. Needs precise models 2. Sensitive to disturbances
Sliding Mode Control (SMC)	1. Resilient to disturbances 2. Robust to model inaccuracies	1. High switching frequencies 2. Noise sensitive
Model Predictive Control (MPC)	1. Handles non-linearities and constraints 2. Predictive of future system behavior	1. Requires high computational power 2. Sensitive to parameter variations
Deep Reinforcement Learning (DRL)	1. Adaptable to complex tasks 2. Model-free	1. High computational needs 2. Needs large, diverse training data

In (Hajihosseini et al., 2020), DRL is applied as a solution to the shortcoming of conventional methods for control of DC/DC buck converter when feeding constant power load (CPL). The conventional methods often demonstrate poor performance in the presence of large changes in the CPL. Although the DRL method proposed in this paper can deliver satisfactory performance, the agent is used for tuning gains in the feedback loop. Thus, not only auxiliary control is required, but also a significant advantage of DRL which is being model-free is not exploited.

In (Tianxiao Yang et al., 2022), a model-free DRL controlling method for DC/DC buck converter feeding CPL is proposed. This method demonstrates good dynamic performance when large changes are applied to the CPL. However, the controller proposed in this paper requires accurate measurements which results in sensitivity to noise and accumulative error

between a trained model and a real-world converter. Thus, this method may not be able to provide reliable performance in practice.

In (Cui et al., 2022), a novel technique is used to enable a DRL-controlled DC/DC buck converter to demonstrate resilient performance facing uncertainties and parameter variations. However, this method requires an off-line pre-trained model of the converter so that an extended state observer (ESO) observes the error between this model and the real-world converter to adapt to parameter variations and uncertainties.

In recent years, due to the significant breakthroughs in artificial intelligence algorithms especially machine learning (ML), and thanks to the remarkable surge in computational capabilities, research on the use of artificial intelligence in power electronics has gained substantial popularity (Pouria Qashqai et al., 2019). Some studies have focused on machine learning-based modeling approaches (Pouria Qashqai, Al-Haddad, et al., 2022; Pouria Qashqai et al., 2020b, 2021; Pouria Qashqai, Zgheib, et al., 2022), while others have utilized machine learning to enhance the control of power electronics.

Although a considerable amount of research is focused on applications of DRL in power electronics, the status of research on the advantages of this method for the control of power electronic converters is still in its infancy (Pouria Qashqai et al., 2019). For instance, In (Alfred, Czarkowski, & Teng, 2021), DRL is used for control of a simple buck converter but no experimental tests are performed for evaluation. Similarly in (Gheisarnejad, Farsizadeh, & Khooban, 2021), a buck converter is controlled by DRL and real-time simulations are performed for evaluation but no real-life practical results are obtained. In (Yihao Wan, Dragicevic, Mijatovic, Li, & Rodriguez, 2021), a hybrid method is implemented for the control of a two-level converter. A DRL agent is utilized for obtaining the weighting factor design of an MPC controller. Few papers have studied applications of DRL in more complex converters such as Neutral Point Clamped (NPC) converters. For instance, in (J. Wang et al., 2022), a DRL agent is used for the efficiency optimization design of a three-level NPC. In (Pouria Qashqai et al., 2020a), a model-free DRL method for controlling the three-level NPC is

proposed. This method utilizes an actor-critic method to apply all the possible switching states under different conditions to learn the optimal switching algorithm that satisfies the control objectives. Although this method demonstrates a satisfactory steady-state performance, it is not studied under dynamic changes, uncertainties, internal and external parameter variations, and noise in measurements. Additionally, experimental results are not obtained to prove the practical feasibility of the proposed control method. Most importantly, the proposed method is not compared against a conventional non-linear control method like MPC.

As seen, despite the recent gaining interest in this subject, there is a gap in research on the application of DRL as a control method for multi-level power electronic converters such as NPC, emphasizing its advantages regarding uncertainties which conventional non-linear methods face challenges. This chapter aims to improve the literature on this subject by proposing a new control method using DRL.

In this chapter, a model-free DRL control method for three-level NPC is proposed. Through simulation results as well as experimental results, the method is proven to be resilient facing dynamic changes, uncertainties, internal and external parameter variations, and noise in measurements without requiring any auxiliary controllers or pre-trained models. Thus, this method may be a promising solution to the common problems associated with conventional non-linear control methods for power electronic converters.

The rest of this chapter is structured as follows: In section 2.2, the fundamentals of DRL, various popular DRL methods as well as their advantages and disadvantages are explained in detail. In section 2.3, the proposed method is introduced. Simulation results are obtained and discussed in section section 2.4 where the proposed method is compared with another conventional non-linear control method. Section 2.5 is dedicated to the experimental results that prove the real-world feasibility of the method. Finally, the limitations of the proposed method as well as the potential future studies are explained in section 2.6, and the whole chapter is concluded.

2.2 Fundamentals of Deep Reinforcement Learning

In this section, reinforcement learning will be introduced. Then different methods for finding optimal policy are explained and their limitations are discussed. Finally, a modern type of RL that utilizes deep learning (DL), also known as deep reinforcement learning (DRL) is introduced.

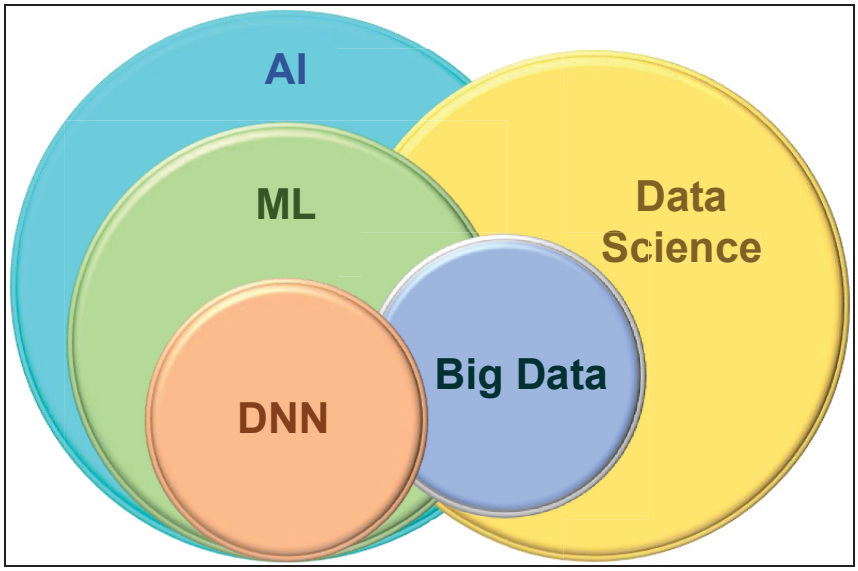

Figure 2.1 Diagram of overlap between data science and its subset big data with artificial intelligence (AI) and its subsets machine learning (ML) and deep neural networks (DNN)

2.2.1 What is Deep Reinforcement Learning

As shown in Figure 2.1, machine learning (ML) is a subset of artificial intelligence (AI). Machine learning is a term used to describe various methods that enable computer programs to make decisions or predictions solely by learning from data. The major difference between these methods and numerical methods is that these methods are capable of generalization. There are three major categories of machine learning algorithms: supervised learning, unsupervised learning, and reinforcement learning as depicted in Figure 2.2.

Supervised learning is inherently an advanced curve-fitting method that learns the relationship between data and maps the input data and their corresponding target data. These sets of data can be labels (classification) or numbers (regression).

Unsupervised learning, on the other hand, finds the relationship between data sets without using predetermined labels. This method is popular for finding anomalies in data and fault detection.

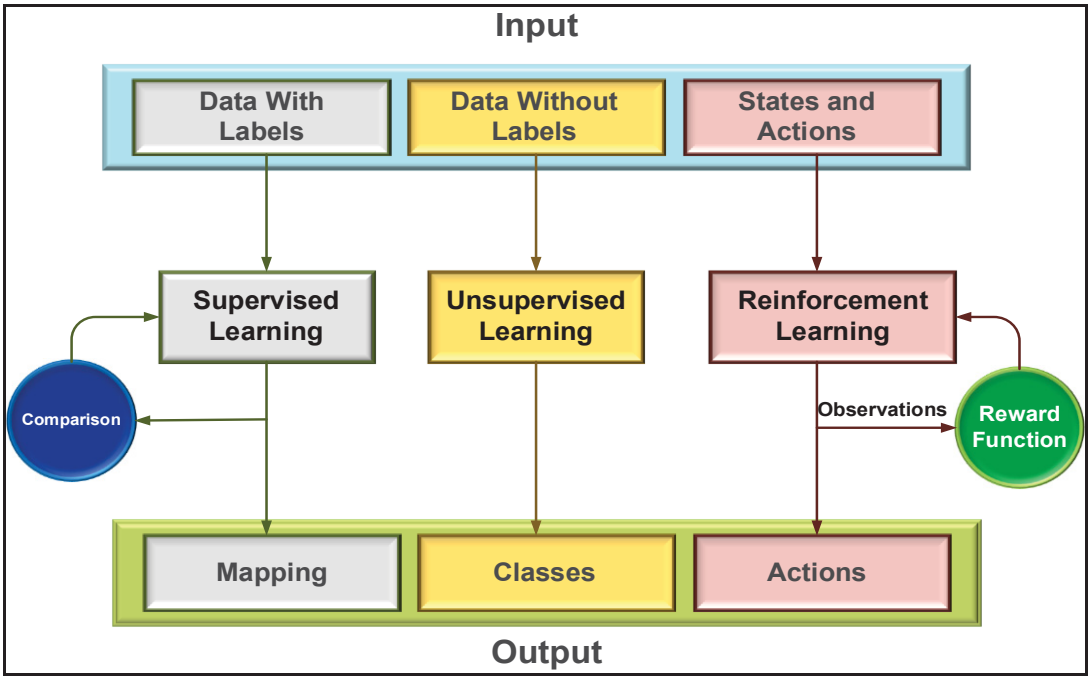

Figure 2.2 Diagram of different machine learning techniques

Reinforcement learning, as shown in Figure 2.3, finds an optimal policy to achieve maximum cumulative reward through interactions with the environment without having access to the model of the environment. Learning solely by trial and error and learning from past experiences, enables RL to be used in various applications where extracting an accurate model is difficult or impossible.

As shown in Figure 2.3, reinforcement learning is comprised of an agent, the environment, action, state, and reward. This can be expressed as a four-tuple Markov Decision Process

(MDP) of $\{s, a, p_a, r_a\}$ where s is the current state, a is the action, p_a is the probability of going to state s' by taking action a in state s and finally $r_{a_}$is the immediate reward. The objective of the RL agent is to find an optimal policy that leads to the maximum accumulative reward as shown in (2.1):

$$\pi^* = \text{argmax}_\pi \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r_t | s_0 \ \pi\right] \tag{2.1}$$

π^* is an optimal policy continuously updated through (2.1), r_t is the reward at the time step t where the initial state is s_0, and γ is a discount factor that indicates how much the previous actions with their corresponding rewards can affect the future reward.

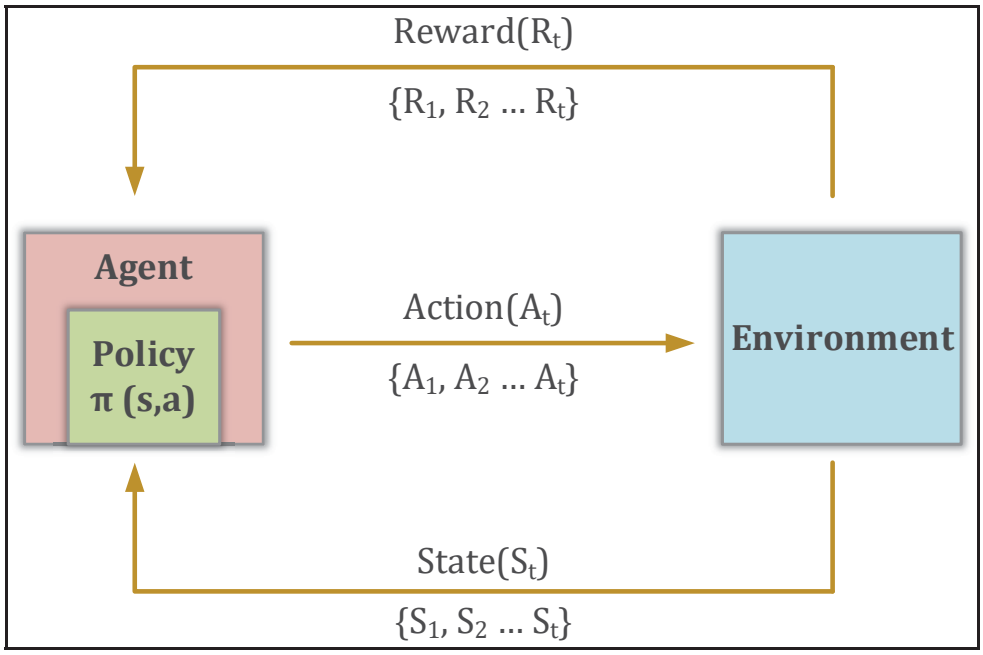

Figure 2.3 Diagram of a reinforcement learning agent exploring an environment, updating a policy of π (s,a) at time step of t

2.2.2 Finding the optimal policy

There are three main methods for solving RL problems and obtaining the optimal policy for maximum accumulative long-term reward.

2.2.2.1 Value-based methods

These methods try to find the optimal value function without explicitly dealing with the policy. Using iterative algorithms such as Q-learning (Jang, Kim, Harerimana, & Kim, 2019) and SARSA (D. Zhao, Wang, Shao, & Zhu, 2016), these methods update a value function. This value function can predict the reward of taking action based on current states and previous rewards. To update the value function, the Bellman equation is utilized:

$$V(s,a) = R(s,a) + \gamma\, max\left(V(s^{'},a^{'})\right) \tag{2.2}$$

Where $V(s,a)$ is the value function in state s, $R(s,a)$ is the immediate reward for taking action a in state s, γ is the discount factor that determines how much the future rewards are taken into account, and finally $maxV(s^{'})$ is the maximum value function in the state s' for taking all actions of a'. Where a' represents the action that maximizes the value function in the state s'.

There are several disadvantages to using value-based methods, such as slow convergence rates, especially for high-dimensional state spaces, the inherent difficulty in handling continuous action spaces, and suboptimal performance in capturing time information in time series. To mitigate these problems, DQN-D and DRQN are proposed (Y. Zhang & Wang, 2022). Despite these efforts, value-based methods are still recommended to be implemented in discrete action spaces.

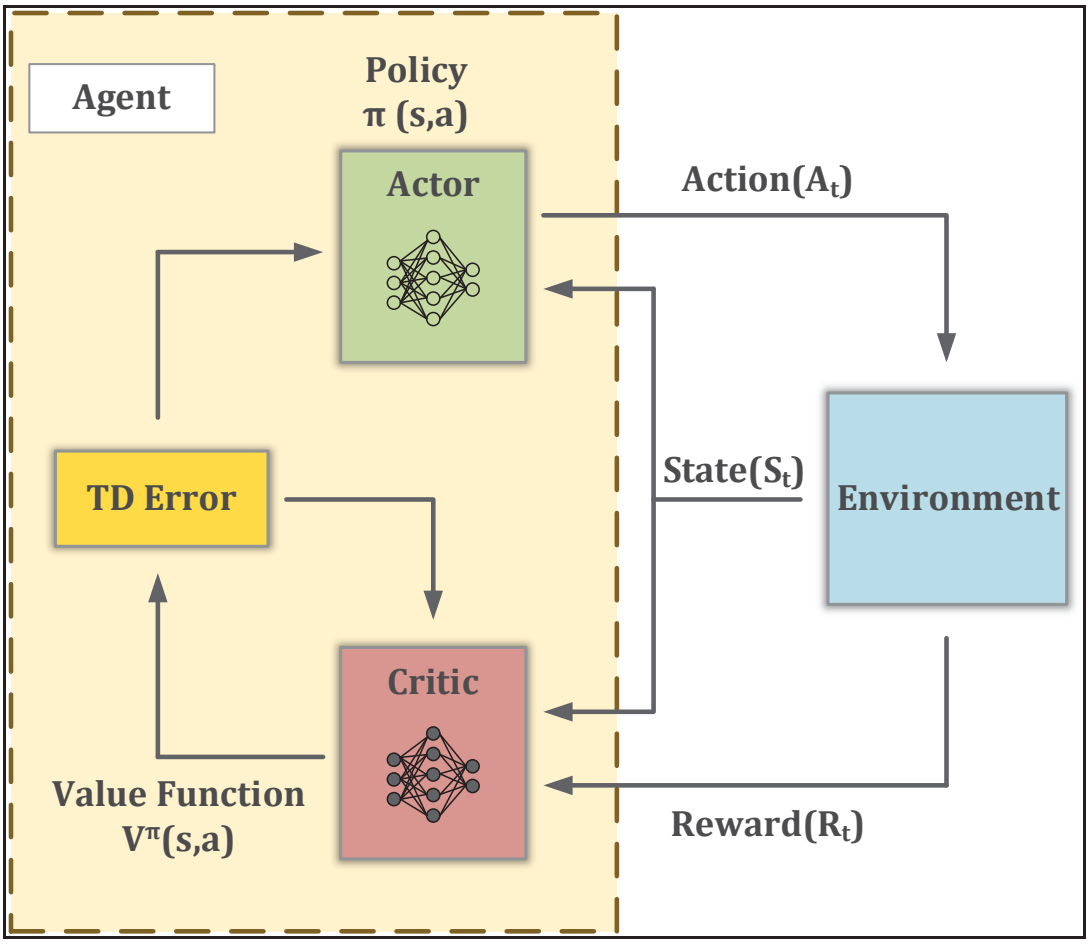

Figure 2.4 Block diagram of an actor-critic reinforcement learning method

2.2.2.2 Policy-based methods

These methods try to find the optimal policy function without calculating the value function first. Using gradient-based algorithms such as Proximal Policy Optimization (PPO) (Schulman, Wolski, Dhariwal, Radford, & Klimov, 2017) and Advantage Actor-Critic (A2C) (Mukhopadhyay, Bandyopadhyay, Sutradhar, & Chattopadhyay, 2019), these methods update the policy parameters judging by their performance. In (2.3) the equation for updating the policy parameters in gradient ascent, one of the most popular methods is depicted.

$$\theta \leftarrow \theta + \propto \nabla_{\theta} J(\theta) \tag{2.3}$$

Where θ is the policy parameter, $\propto$ is the learning rate, $J(\theta)$ is the objective function, and finally $\nabla_\theta J(\theta)$ is the gradient of the objective function $J(\theta)$.

These methods suffer from several disadvantages such as high fluctuations in training episodes, becoming stuck in local optimal, and difficulty in handling discrete actions (Y. Liu, Yang, Chen, Guo, & Jiang, 2020).

2.2.2.3 Actor-critic methods

These methods combine the two previous methods in the form of two neural networks. The neural network which is responsible for policy is called "actor", whereas the neural network responsible for value function is called "critic". A diagram of an actor-critic RL is depicted in Figure 2.4. As it can be seen, the RL agent uses the actor to generate actions and the critic evaluates how well the actions were chosen based on the value function. The actor judges by its temporal difference (TD) error. This enables the actor-critic methods to be more data efficient compared to other methods.

However, these methods suffer from noisy gradients which leads to unstable learning and sensitivity to hyperparameters which leads to unreliable performance in disturbances (Dutta & Upreti, 2022).

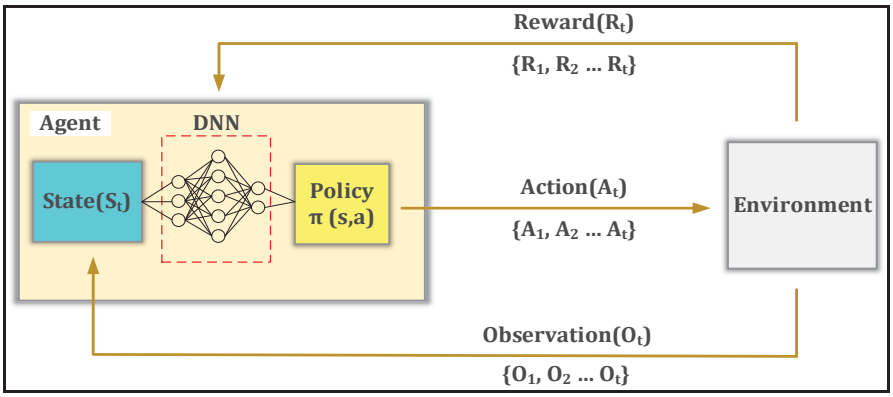

Figure 2.5 Block diagram of a reinforcement learning agent exploring an environment, updating a policy of π (s, a) at the time step of t

2.2.3 Deep Reinforcement Learning

Conventional RL methods use linear function approximations. Therefore, their applications are limited to less complex problems with low levels of non-linearity. These methods may not be able to find an optimal policy when large state spaces or action spaces are present. By implementing neural networks into RL, as shown in Figure 2.5, a more powerful generation of RL methods titled Deep Reinforcement Learning (DRL) emerged. Using neural networks provides several advantages in comparison to conventional RL methods: first, neural networks can be generalized, which makes them effective in dealing with uncertainties and unseen scenarios. Secondly, neural networks are capable of effectively handling high dimensions of input and output which enables them to deal with large state and action spaces. Moreover, neural networks can approximate non-linear functions, making them suitable for highly complex applications. And finally, neural networks can learn from experience, which enables them to learn a policy that is not known beforehand or may change in the future (X. Wang et al., 2022).

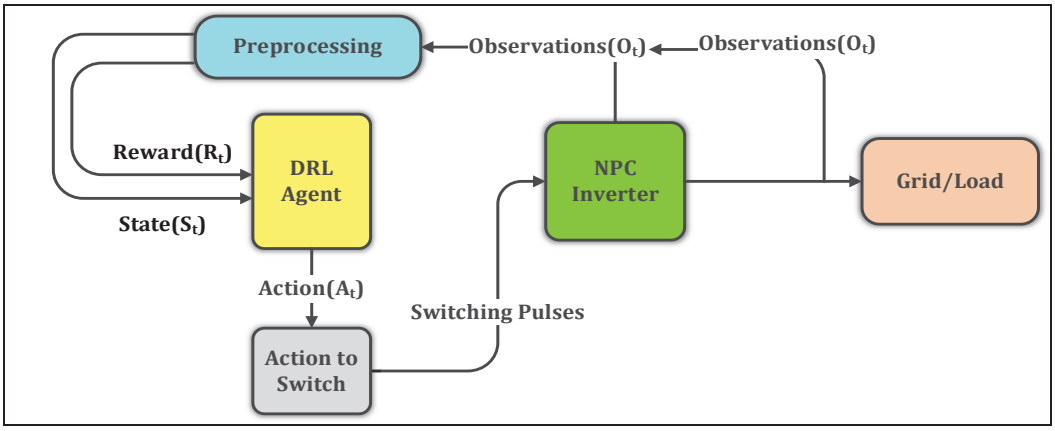

Figure 2.6 Block diagram of the proposed method using a DRL agent at time t to generate action A_t which is converted to switching pulses using the "Action to Switch" algorithm

Table 2.2 Popular agent types and other action spaces

Agent	Type	Action Space	On/Off Policy
Q-Learning Agents (Q)	Value-Based	Discrete	Off
SARSA Agents	Value-Based	Discrete	On
Deep Q-Network (DQN) Agents	Value-Based	Discrete	Off
Policy Gradient Agents (PG)	Policy-Based	Discrete or continuous	On
Actor-Critic Agents (AC)	Actor-Critic	Discrete or continuous	On
Deep Deterministic Policy Gradient (DDPG) Agents	Actor-Critic	Continuous	Off
Twin-Delayed Deep Deterministic Policy Gradient Agents (TD3)	Actor-Critic	Continuous	Off
Soft Actor-Critic Agents (SAC)	Actor-Critic	Continuous	Off
Proximal Policy Optimization Agents (PPO)	Actor-Critic	Discrete or continuous	On
Trust Region Policy Optimization Agents (TRPO)	Actor-Critic	Discrete or continuous	On
Model-Based Policy Optimization Agents (MBPO)	Actor-Critic	Discrete or continuous	Off

2.3 The Proposed Method

This section discusses the different aspects of the proposed method. First, the agent type is chosen. Secondly, observations that are signals measured by sensors, are selected. Then reward function and action space are proposed. Finally, the implementation of the DRL agent on a

three-level NPC, as a switching and control method is discussed. The overall diagram of the proposed method is shown in Figure 2.6.

2.3.1 Selecting Agent Type

As mentioned in the previous section, there are different methods for finding the optimal policy for RL agents. Each of these methods has its advantages and disadvantages. Policy-based methods are more efficient when action space is continuous, however, value-based methods are better at handling large continuous state spaces while tackling discrete action spaces (Y. Liu et al., 2020). Actor-critic methods are effective in both continuous and discrete state spaces and action spaces.

However, these methods require more computational power and memory due to utilizing two separate networks. They also suffer from instability, delayed reward, and extended training times due to the correlation problems between actors and critics (Mukhopadhyay et al., 2019). The most popular DRL agent types supported by MATLAB ("Reinforcement Learning Agents - MATLAB & Simulink," n.d.) are listed in Table 2.2.

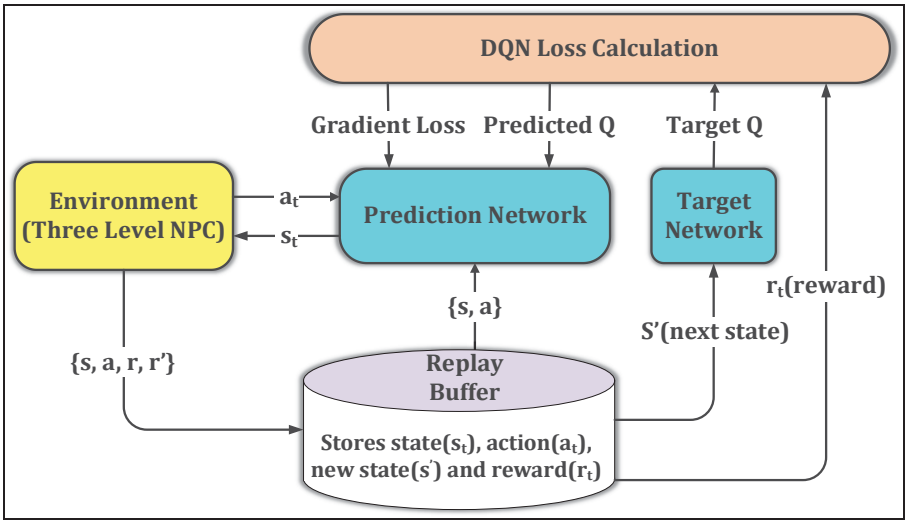

Figure 2.7 Topology of the deep Q-learning network (DQN) agent utilizing a replay buffer to find the optimal value function for control of the environment

Since control of a three-level NPC is inherently comprised of a large continuous state space and a small action space, a value-based method like Q-learning would be a proper choice. Moreover, since deep learning provides many advantages over conventional numerical methods of RL, in this chapter, the deep network form of Q-learning which is Deep Q-network (DQN) is selected. A diagram of the DQN network is shown in Figure 2.7. Unlike Q-learning, which uses a Q-table as a lookup table to store pairs of state-action and their values, DQN uses neural networks and a replay buffer to achieve the same goal.

2.3.2 Observations and preprocessing

To generate a proper state space as depicted in (2.4), various observations are required.

$$\mathrm{s_t} = \mathrm{f}(o_t) \tag{2.4}$$

In (2.4), s_t is the state space at time t, whereas o_t is the matrix of observations at time t, and f is a function that manipulates the observations through a preprocessing algorithm to map them to the state space. In (2.5), the observation matrix is depicted:

$$o_t = \left[V_{abc}, id_{ref}, iq_{ref}, I_{abc}, V_{C1}, V_{DC}\right] \tag{2.5}$$

Where V_{abc} and I_{abc} are the three-phase voltage and current of the grid (or load, depending on the operational mode) respectively, id_{ref} and iq_{ref} are the d and q reference currents in the park's transformation respectively, V_{C1} is the voltage across C_1 one of the capacitors in the DC link (since the voltage of C_2 is dependent on this voltage and balancing one capacitor is enough to balance both of them), and finally, V_{DC} is the voltage of the DC link.

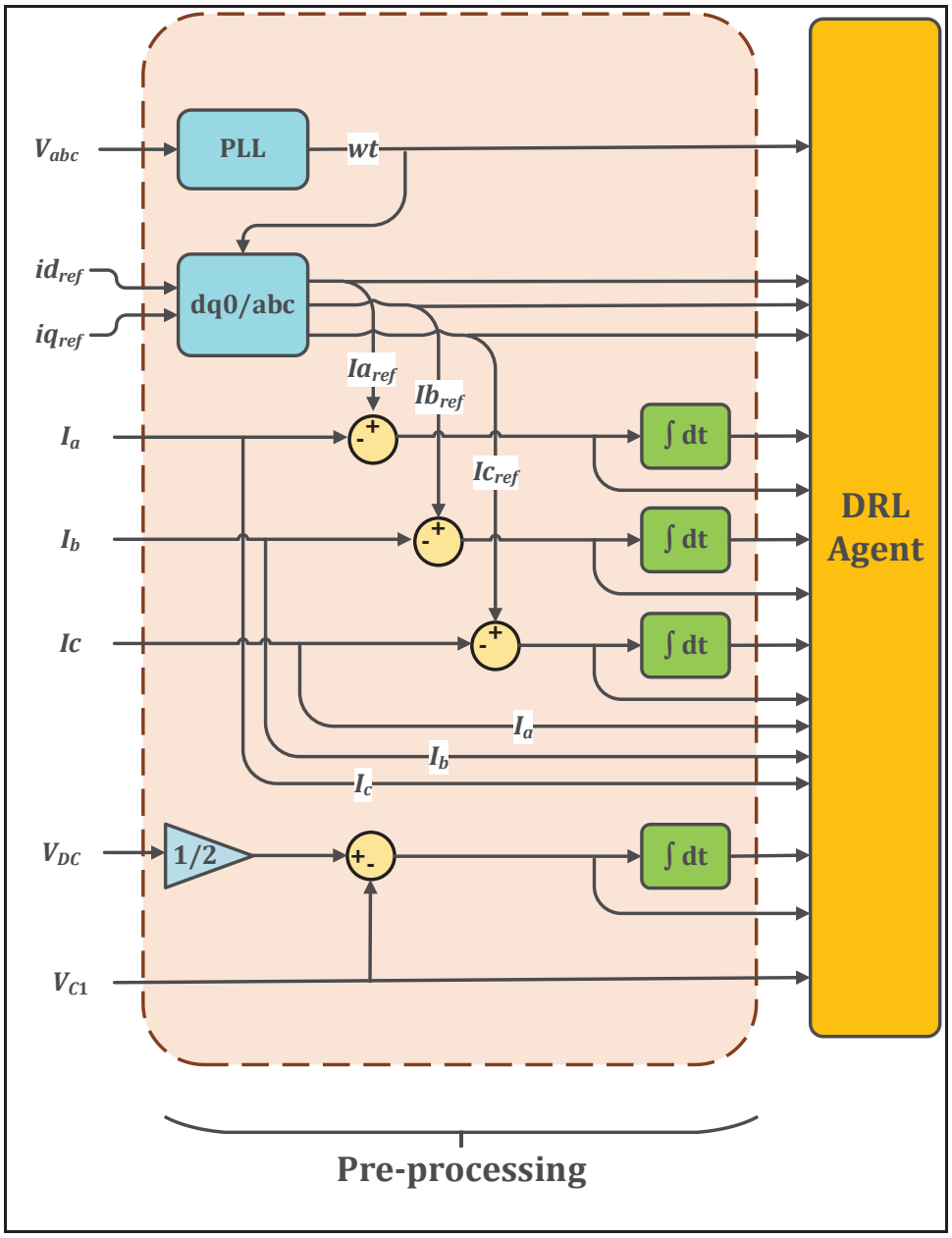

Figure 2.8 Diagram of the pre-processing unit that manipulates observations to become states that are comprehensible by the agent

Using the measurements in o_t may not converge or may lead to significantly high training times. Thus, a pre-processing unit is used to manipulate the signals so that the agent can determine the performance of its policy more effectively. The pre-processing block is shown

in Figure 2.8. As seen, the state space is continuous and comprised of various signals and measurements mapped to represent the behavior of the converter.

2.3.3 Reward Function

Reward function creation is probably the most important and most challenging part of DRL. Since the agent has no agency, it can easily exploit undesirable outcomes to achieve an optimal reward. Therefore, it is vital to create a reward function that eliminates any potentially undesirable or nonsensical scenarios that may lead to high rewards. For instance, let us assume that a DRL agent is utilized for driving a vehicle and a reward function delivers a medium reward for maintaining speed and a high reward for not colliding with the environment. By constructing such a flawed reward function, the agent may decide not to move the vehicle at all to achieve maximum reward, which would be undesirable. Having this notion in mind, the following reward function is constructed for this control algorithm:

$$R_T^t = \alpha(\varphi^{\mathrm{t}}(\mathrm{i_d}) + \varphi^{\mathrm{t}}(\mathrm{i_q}) + \varphi^{\mathrm{t}}(\mathrm{V_{C1}})) \tag{2.6}$$

Where t is the time at any given time step. $\varphi(\mathrm{i_d})$ and $\varphi(i_q)$ are the reward functions of the d and q currents in park's transform respectively, $\varphi(V_{C1})$ is the reward function of the voltage across C_1, α is the gain, and finally, R_T is the total reward applied to the DQN agent. The reward functions of i_d, i_q and V_{C1} can be obtained using (2.7), (2.8) and (2.9), respectively.

$$\varphi^{\mathrm{t}}(\mathrm{i_d}) = \begin{cases} -2|\nabla(i_d)|^2 & |\nabla(i_d)| > 0.2\ \mathrm{A} \\ -2|\nabla(i_d)|^2 + 2e^{-2} & |\nabla(i_d)| \leq 0.2\ \mathrm{A} \end{cases} \tag{2.7}$$

$$\varphi^{\mathrm{t}}(\mathrm{i_q}) = \begin{cases} -2|\nabla(i_q)|^2 & |\nabla(i_q)| > 0{,}2\ \mathrm{A} \\ -2|\nabla(q)|^2 + 2e^{-2} & |\nabla(i_q)| \leq 0{,}2\ \mathrm{A} \end{cases} \tag{2.8}$$

$$\varphi^{\mathrm{t}}(V_{C1}) = \begin{cases} -|\nabla(V_{C1})|^2 & |\nabla(V_{C1})| > 5\ \mathrm{V} \\ -|\nabla(V_{C1})|^2 + 5e^{-3} & |\nabla(V_{C1})| \leq 5\ \mathrm{V} \end{cases} \tag{2.9}$$

Where $\nabla(i_d)$, $\nabla(i_q)$, and $\nabla(V_{C1})$ are the errors between each of these signals and their reference values, as shown in (2.10), (2.11) and (2.12), respectively.

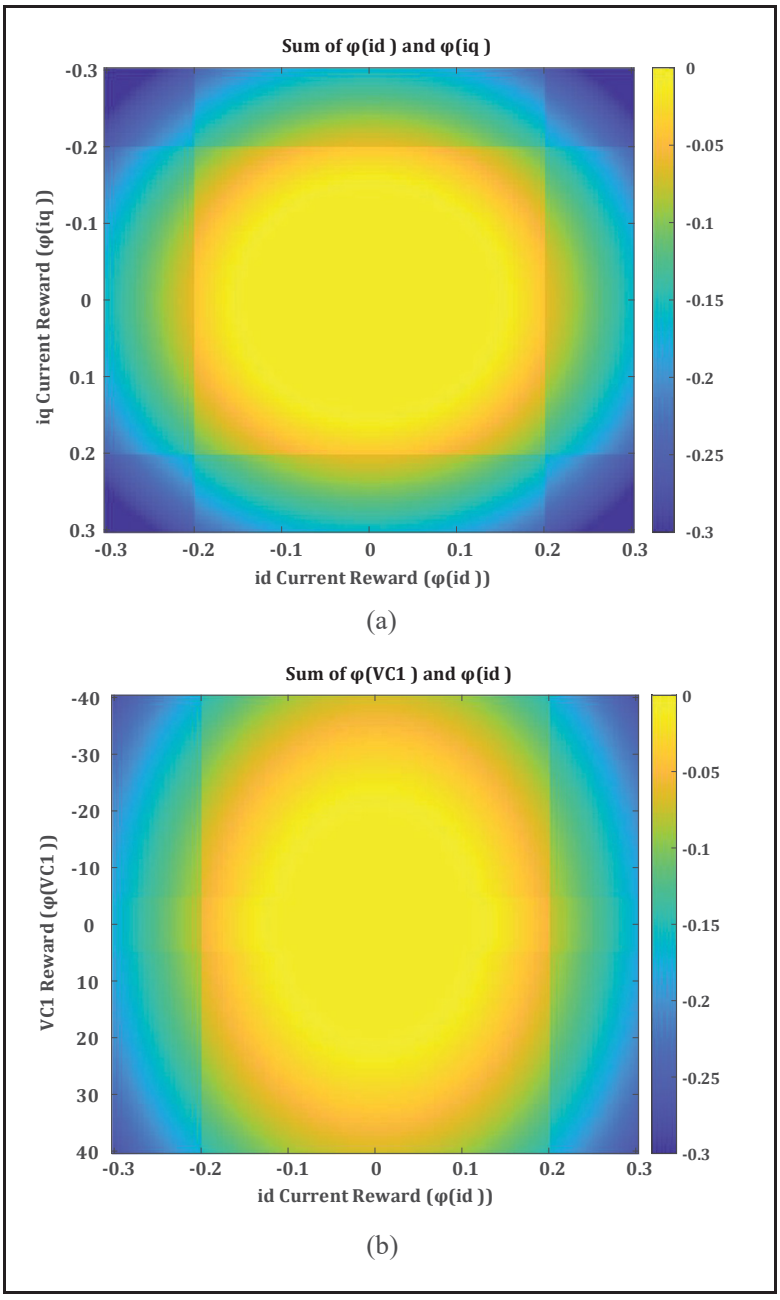

Figure 2.9 Heatmap of the combined reward for errors between id and iq with their reference values (a); and errors between id and VC1 and their reference values (b)

$$\nabla(i_d) = i_d - i_{d_{ref}} \tag{2.10}$$

$$\nabla(i_q) = i_q - i_{q_{ref}} \tag{2.11}$$

$$\nabla(i_d) = V_{C1} - \frac{V_{DC}}{2} \tag{2.12}$$

To better understand the reward function, a heatmap of the combined rewards of $\varphi^t(i_d)$ and $\varphi^t(i_q)$, as well as, $\varphi^t(i_d)$ and $\varphi^t(V_{C1})$ are depicted in Figure 2.9-a. and Figure 2.9-b, respectively. In order to demonstrate the overall reward, a 3D graph of the total reward under three scenarios is depicted in Figure 2.10. Scenario-1 is when only $\varphi^t(i_d)$ is considered. Scenario-2 combines the rewards of $\varphi^t(i_q)$ and $\varphi^t(i_q)$. Eventually, scenario-3, adds $\varphi^t(i_q)$ to the equation which makes it equal to the total reward. The *x*-axis represents the percentage of deviation from nominal errors of i_d, i_q, and V_{C1}. As can be seen, negative rewards are used to punish undesirable behaviors of the agent. Thus, the optimal reward for this agent is nearly zero.

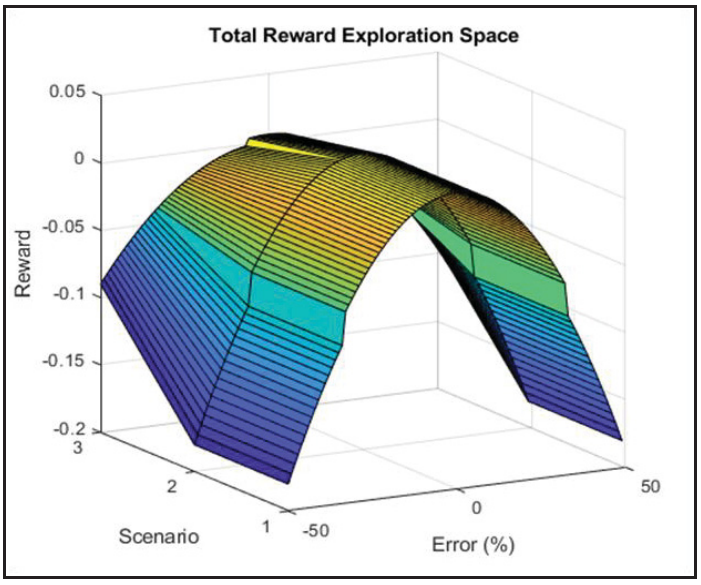

Figure 2.10 Exploration space for the DRL agent to achieve maximum reward

2.3.4 Action space

Before discussing the action space, consider the topology of a three-level NPC illustrated in Figure 2.11. As seen, the converter is comprised of 12 switches. Since each switch has two states of On and Off. Therefore, there are $2^4 = 16$ combinations at each leg of the converter. However, only three combinations are permitted for each leg as is listed in Table 2.3 for leg X (i.e., A, B, or C). The rest of the combinations lead to fault or short-circuit. Since the converter has three permitted states and it has three legs, the total number of permitted switching states is $3^3 = 27$. Since these labels are unintelligible for the DRL agent, an integer number is used to represent each combination. Thus, the discrete action space in this method would be:

$$a_t = [0\ 1\ \ldots\ 26] \tag{2.13}$$

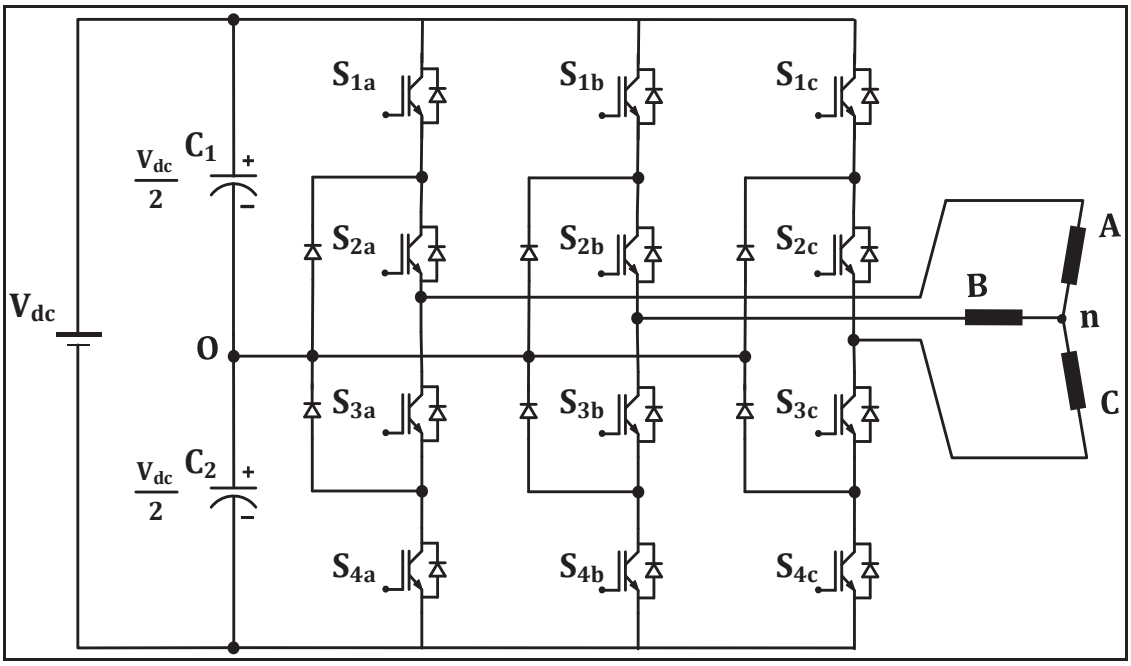

Figure 2.11 Topology of a three-level neutral point clamped (NPC) converter

Using a lookup table connected to the output of the DRL agent, each action which is an integer, is converted to its corresponding switching signals. As shown in Figure 2.6, the "Action to Switch" block maps actions to switching signals.

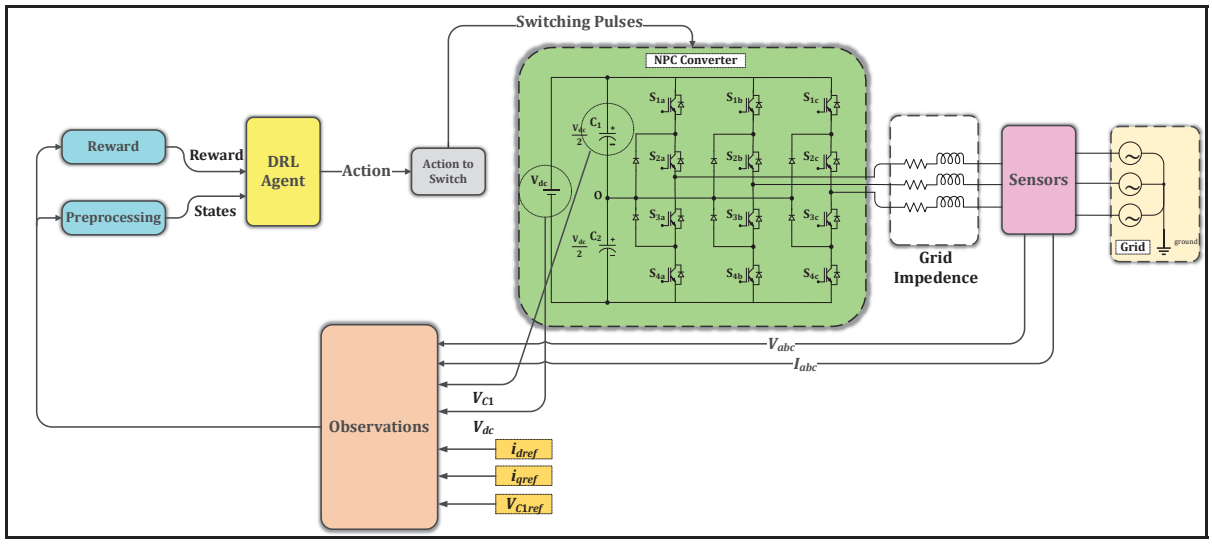

Figure 2.12 Block diagram of the proposed method for control and switching of a three-level NPC

Table 2.3 Switching states of leg X of a three-level NPC

Switching State	S_{1X}	S_{2X}	S_{3X}	S_{4X}	Output Voltage
P(+)	On	On	Off	Off	$+\frac{V_{dc}}{2}$
O	Off	On	On	Off	0
N(-)	Off	Off	On	On	$-\frac{V_{dc}}{2}$

2.3.5 Connecting the agent to the converter

The diagram of a three-level NPC controlled by the proposed method is shown in Figure 2.12. The actions taken by the DRL agent do not change until the next sampling period of the agent

which is not necessarily equal to the sampling time of the simulation. Therefore, the sampling period of the DRL agent is equal to the switching frequency (f_{sw}).

2.4 Simulation Results

To evaluate the performance of the proposed method, we implemented it in Matlab/Simulink simulation environment usi. Simulation parameters, as well as training parameters, are listed in Table 2.4 and Table 2.5, respectively.

Table 2.4 Simulation Parameters

Parameter	**Value**	**Unit**
Grid Voltage (per phase)	170	*V*
Grid Frequency	60	*Hz*
Grid Resistance (per phase)	0.1	Ω
Grid Inductance (per phase)	5	*mH*
DC-link Capacitor(s)	1000	μF
DC-link Voltage	400	*V*

Table 2.5 Training Parameters

Parameter	**Value**
Layer Size of the State Path	140
Layer Size of the State Path	48
Learning Rate	0.001
Normalization	None
Bias Learn Rate Factor	0
Double DQN	No
Mini Batch Size	320
Discount Factor	0.01
Score Averaging Window Length	5

After training the DRL agent with a sampling time of 50µs, which is equal to a switching frequency of 20 kHz, the saved agent is initiated with a sampling time of 100 µs which is equal to a switching frequency of 10 kHz.

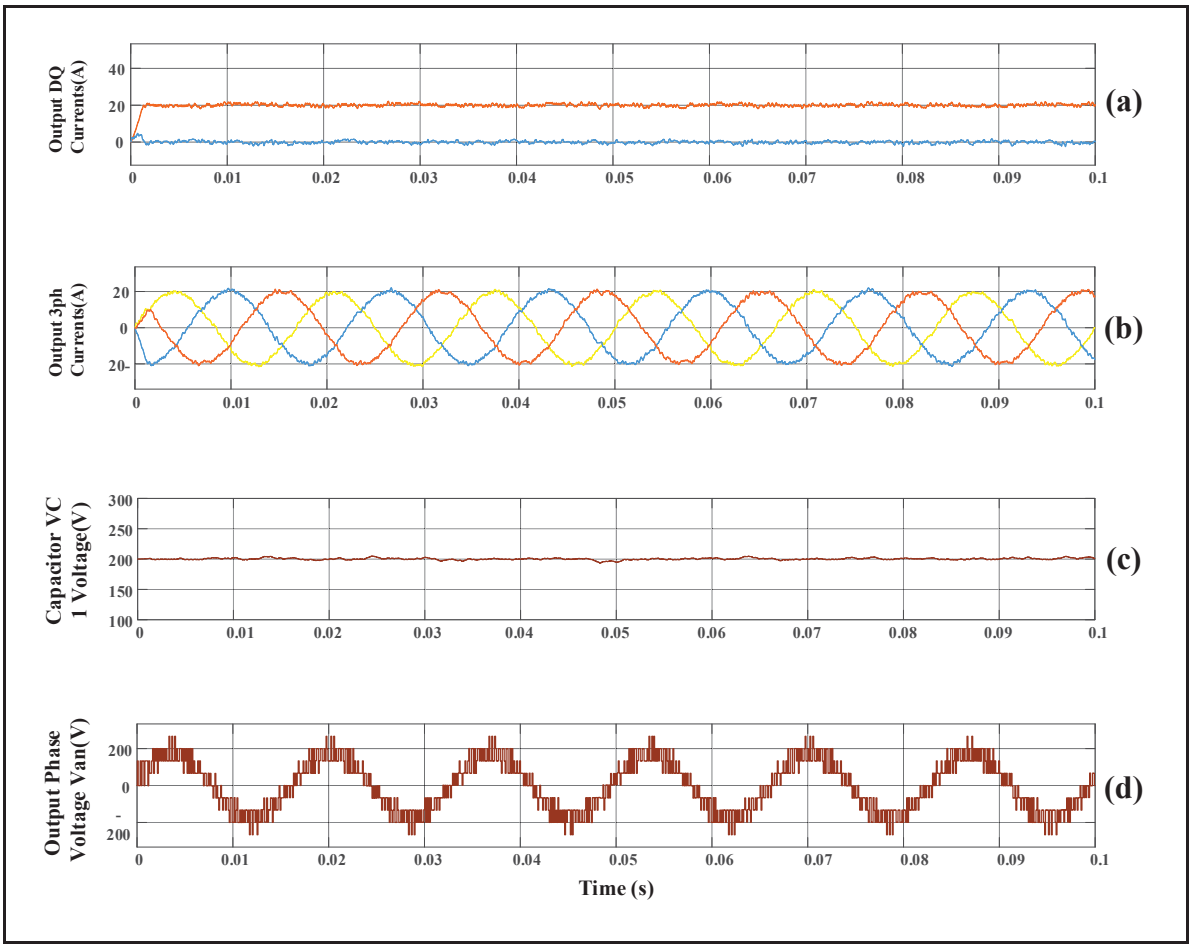

Figure 2.13 Waveforms of the output i_d and i_q currents (a); output three-level currents (b); the voltage across the C_1 capacitor (c); and V_{an} the output phase voltage (d), in steady-state operation, when $i_{dref} = 20$ and $i_q = 0$

2.4.1 Steady-state operation

By setting the reference currents of i_{dref} and i_{qref} to 20 A and 0A, respectively, the steady-state results are obtained. The steady-state waveforms of the output i_d and i_q currents, output

three-level currents, the voltage across the C_1 capacitor, and V_{an} the output phase voltage is illustrated in Figure 2.13-a, Figure 2.13-b, Figure 2.13-c, Figure 2.13-d, respectively. As can be seen, the DRL agent is capable of effectively following the reference currents, while balancing the dc-link capacitors. It is worth mentioning that the THD of the three-phase output currents is 3.62%, which is within the standard limits.

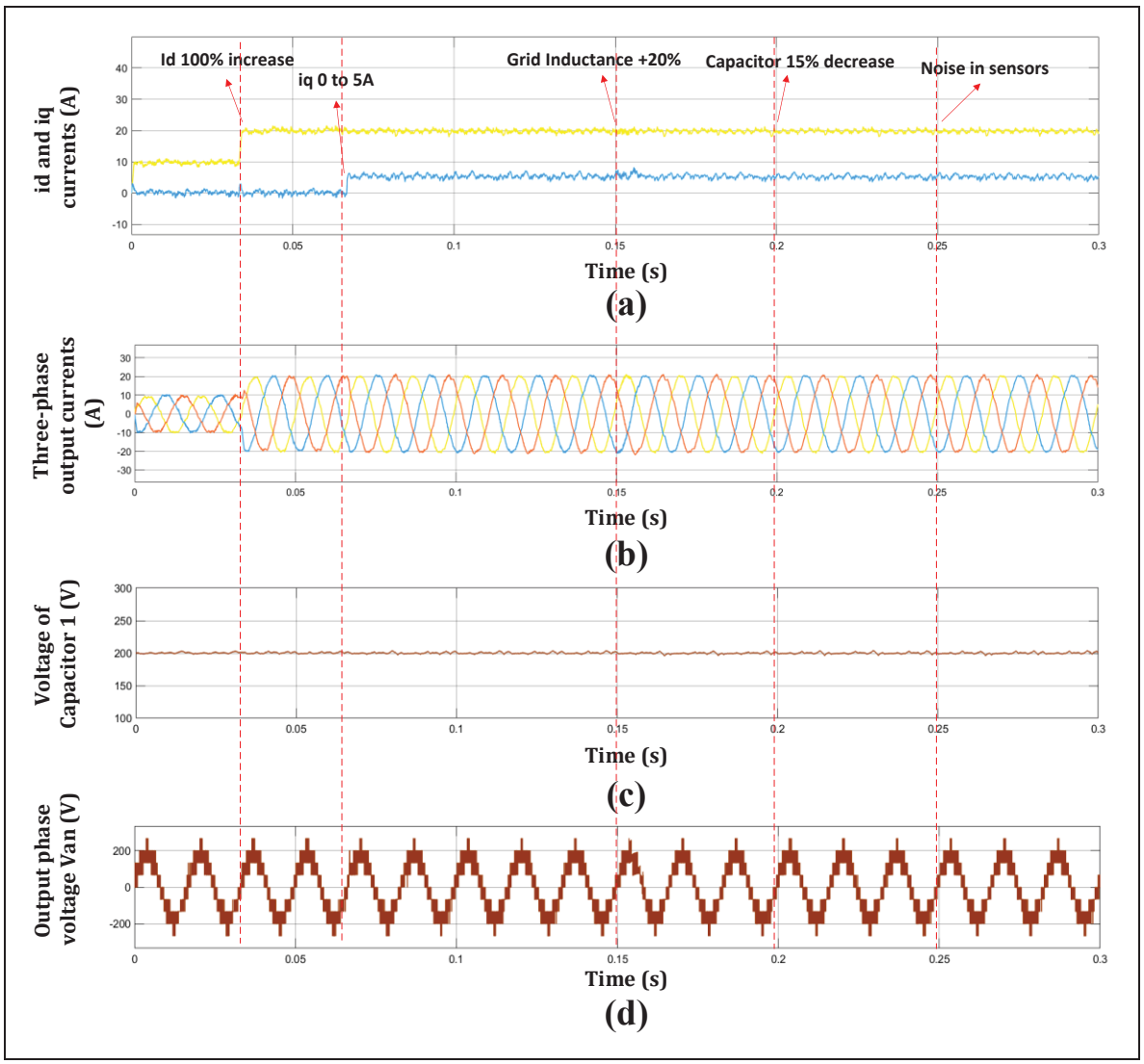

Figure 2.14 Waveforms of the output i_d and i_q currents (a); output three-phase currents (b); the voltage across the C_1 capacitor (c); and V_{an} the output phase voltage (d), when facing active power changes, reactive power changes, grid inductance increase, capacitor degradation, and noise in measurements

2.4.2 Uncertainties and Parameter Variation

As mentioned earlier, the proposed method in this chapter can take advantage of the characteristics of DRL to generalize its knowledge of the converter when facing uncertainties. To evaluate the performance of the proposed method, the simulation is run for 300 milliseconds with different operation modes. Five scenarios are examined, and their simulation results are depicted in Figure 2.14, where Figure 2.14-a shows the output i_d and i_q currents, Figure 2.14-b shows the output three-level currents, Figure 2.14-c illustrates the voltage across the C_1 capacitor, and Figure 2.14-d depicts V_{an} the output line voltage. As seen at time $t = 33\,ms$, i_{dref} is changed from 10A to 20A to evaluate the dynamic response of the DRL agent to sudden active power changes. Similarly, at time $t = 66\,ms$, i_{qref} is changed from zero to 20A to evaluate the dynamic response of the DRL agent to sudden reactive power changes. To assess the resilience of the proposed method in the presence of external parameter variations, the grid inductance is increased by 20% at $t = 150\,ms$. There is a slight distortion in the waveforms but after a few samples, the agent adapts to the new condition. To evaluate the resilience of the proposed method in the presence of internal parameter variations, the capacity of each one of the dc-link capacitors is reduced by 15% at $t = 200ms$.

2.4.3 Noise in measurement

Ultimately, to evaluate the robustness of the proposed method when there is noise in measurements, a white Gaussian noise (WGN) is added to each measurement signal at $t = 250\,ms$. The power of each added WGN is adjusted so that the signal-to-noise ratio (SNR) is 25 dB. As shown in (2.14) and (2.15), the root-mean-square (RMS) of the noise signal should be around 5% of the RMS of the measured signal so that SNR is approximately 25 dB. In (2.14) and (2.15), P and A are the power and RMS of signals or noise, respectively.

$$SNR = \frac{P_{signal}}{P_{noise}} = \left(\frac{A_{signal}}{A_{noise}}\right)^2 \tag{2.14}$$

$$SNR_{dB} = 10\log_{10}(\frac{P_{signal}}{P_{noise}}) \quad (2.15)$$

By doing so it can be seen that the agent continues to perform satisfactorily despite the presence of noise in measurement.

2.4.4 Comparison with the Model Predictive Control (MPC) Method

To further evaluate the proposed method and assess its advantages, disadvantages, and limitations, the same topology and parameters are controlled by both the DRL method, and another conventional non-linear control method known as the Model Predictive Control (MPC). The cost function of the MPC controller is multi-objective as shown in (2.16).

$$\mathrm{G} = \lambda_1 g_1 + \lambda_2 g_2 + \lambda_3 g_3 \quad (2.16)$$

It aims at controlling current and reducing common mode voltage (CMV) and capacitor voltages. Where $\mathrm{g_1}$-$\mathrm{g_3}$ represent the reference current, capacitor voltages, and CMV, respectively. Similarly, λ_1-λ_3 are the corresponding weighting factors.

To perform a fair comparison, the DRL agent is re-trained using the same training options mentioned in Table 2.5. Only this time, the sampling time of both the DRL agent and the MPC controller is equally set to 50µs.

2.4.4.1 Steady-state operation and step response

To assess the performance of each method in terms of steady-state operation and step response, the reference current i_d is increased from 10 A to 20 A at $t = 100\ ms$ as shown in Figure 2.15-a for the DRL agent and Figure 2.15-c for the MPC controller, respectively. As seen, the initialization phase (warm-up) of the DRL method is superior to that of the MPC method.

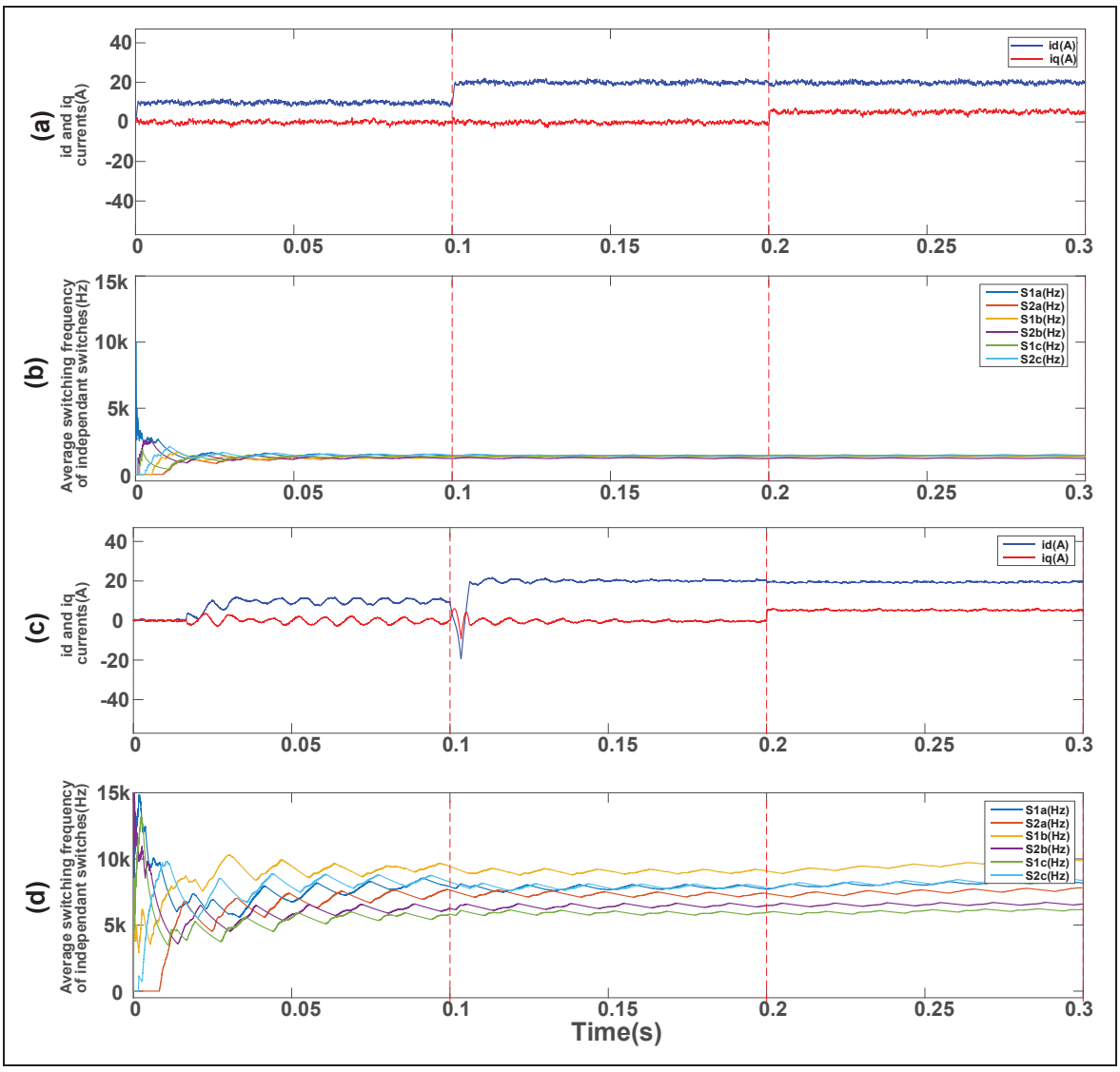

Figure 2.15 Waveforms of the output i_d and i_q currents using the DRL method (a); average frequency of the switches using the DRL method (b); the output i_d and i_q currents using the MPC method (c); and average frequency of the switches using the MPC method (d)

Additionally, the step response of the MPC controller results in a larger undershoot, more prolonged oscillations, and longer settling times. Similarly, as shown in Figure 2.15-a and Figure 2.15-c, the reference current of i_q is changed from 0 A to 5 A at $t = 200ms$. As depicted, both methods resulted in satisfactory transient results as well as accurate and smooth

steady-state performances. It can be concluded that depending on the magnitude of step changes, the DRL agent demonstrates superior warm-up and transient responses compared to the MPC method.

2.4.4.2 Switching losses

In order to evaluate the effect of each method on switching losses, the switching frequencies can be studied. By counting the rising edges of each switch and dividing it by time, the average switching frequency corresponding to each switch can be calculated. Since in the NPC, there are supplementary switch couples where the state of each switch is the inverse of its dual switch (e.g., S_{1a} and S_{3a}), we have selected one switch from each couple. Thus, the switches S_{1a}, S_{2a}, S_{1b}, S_{2b}, S_{1c}, and S_{2c} are considered for an apparent presentation.

As you can see in Figure 2.15-c and Figure 2.15-d, despite having the same sampling time (T_S), the switching frequencies of switches in the DRL method are approximately around 2.5 kHz with a low degree of variance, whereas in the MPC method, not only the switching frequencies are higher and consequently the switching losses are higher but also the switching frequencies have a large degree of variance which makes the design of a filter more complex and more challenging.

In addition, considering the dynamic changes of the reference currents at $t = 100\ ms$ and $t = 200\ ms$ a depicted in Figure 2.15-a and Figure 2.15-c, we can conclude that both methods demonstrate near-constant switching frequencies when facing dynamic changes. However, in the MPC method, the average switching frequency change slightly when the reference currents change but this change is negligible.

Ultimately, it is worth mentioning that despite having higher switching frequencies the MPC method results in slightly smoother current waveforms with a THD of 2.44% compared to the THD of the DRL method which equals 3.62%. Considering the less satisfactory performances of the NPC method in terms of switching frequency and switching losses, and despite resulting

in slightly smoother waveforms and better THD, we can conclude that the DRL method is superior to the MPC method in this domain.

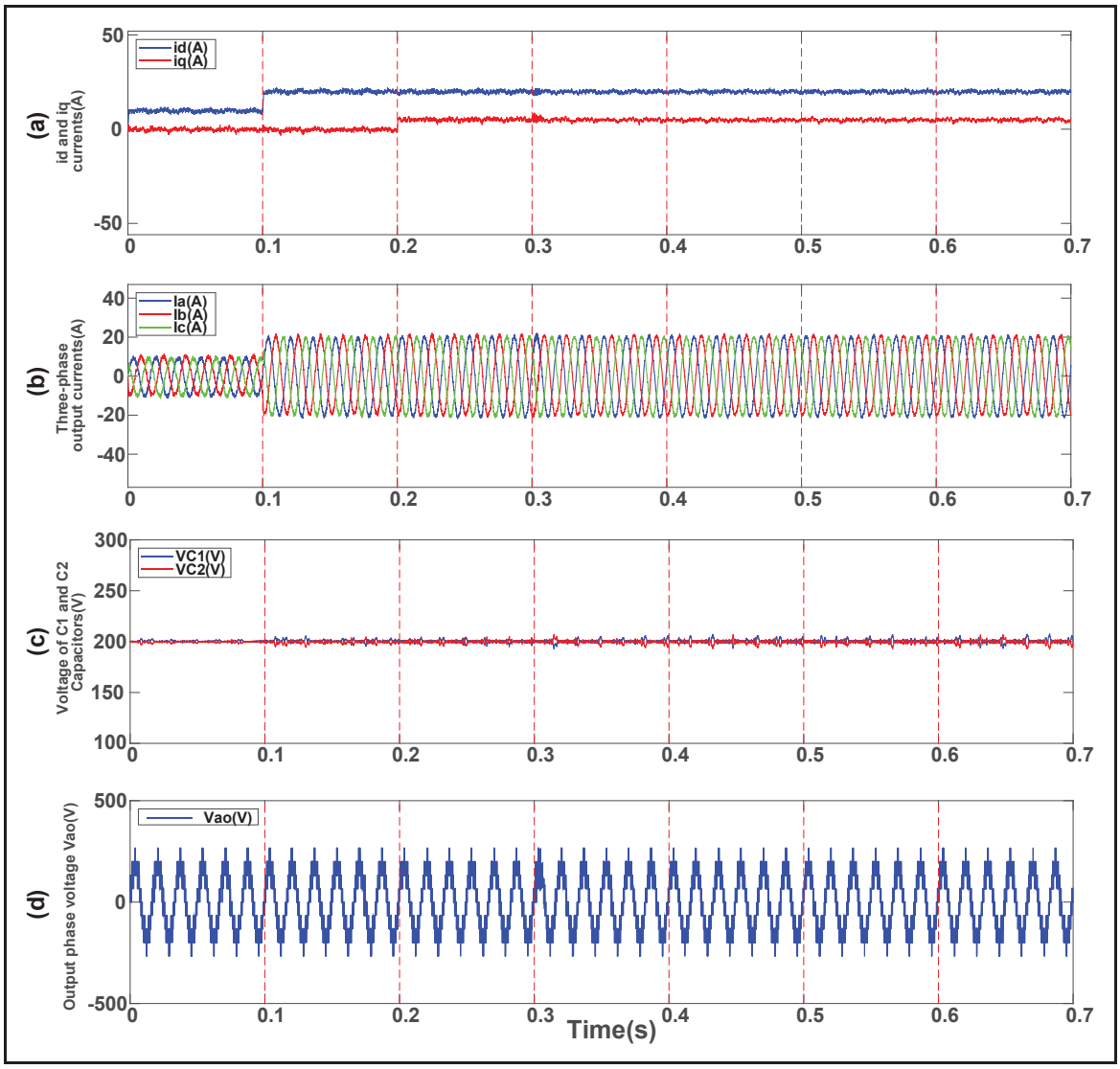

Figure 2.16 Waveforms of the output i_d and i_q currents (a); output three-phase currents (b); the voltage across the C_1 and C_2 capacitors (c); and V_{ao} the output phase voltage (d), when facing active power changes, reactive power changes, grid inductance increase, capacitor degradation, grid voltage increase and noise in measurements controlled by the DRL method

2.4.4.3 Uncertainty and parameter variations

As stated before, control methods based on ML including DRL have the ability to generalize policies to never-seen scenarios. Furthermore, the DRL method proposed in this chapter is inherently model-free and consequently, not sensitive to parameter changes. In contrast, the MPC method which has an acceptable performance for nonlinear control of power electronic converters, not only requires the precise model of the converter but also cannot generalize its policy to unexplored areas and unknown scenarios. For this reason, we have performed a series of simulation tests in this section of the chapter to evaluate and compare the resilience of each method when facing parameter changes and uncertainties. Six scenarios are considered for comparison as explained below.

A step increase in the reference current i_d from 10 A to 20 A at $t = 100\ ms$ is applied to the DRL agent as shown in Figure 2.16 and to the MPC controller as shown in Figure 2.17, respectively. As can be seen and stated earlier, the transient response to this change is superior in the DRL method in terms of settling time, undershoot, and oscillations. In the MPC method the output phase voltage v_{ao} is slightly distorted. Although the MPC controller mitigates this distortion after a short time, such deformation is not present in the DRL method. In addition, the waveform of v_{ao} is more symmetrical in the DRL method. As stated before, THD is slightly lower in the MPC method resulting in smoother output current waveforms. Nonetheless, this smoothness in the waveform comes at the cost of higher switching frequencies and consequently higher switching losses. The voltages of the DC-link capacitors are successfully balanced in both methods, but their ripples are slightly increased in both methods upon increasing i_d.

A step increase in the reference current i_q from 0 A to 5 A at $t = 200\ ms$ is applied to the DRL method as shown in Figure 2.16 and to the MPC method as shown in Figure 2.17. The transient response of the DRL method is similar to the previous scenario while the transient response of the MPC method to a step change in i_q is better than the previous scenario.

Contrasting the previous scenario, the waveform of v_{ao} has remained without distortion in both methods. The voltages of the DC-link capacitors remain unchanged.

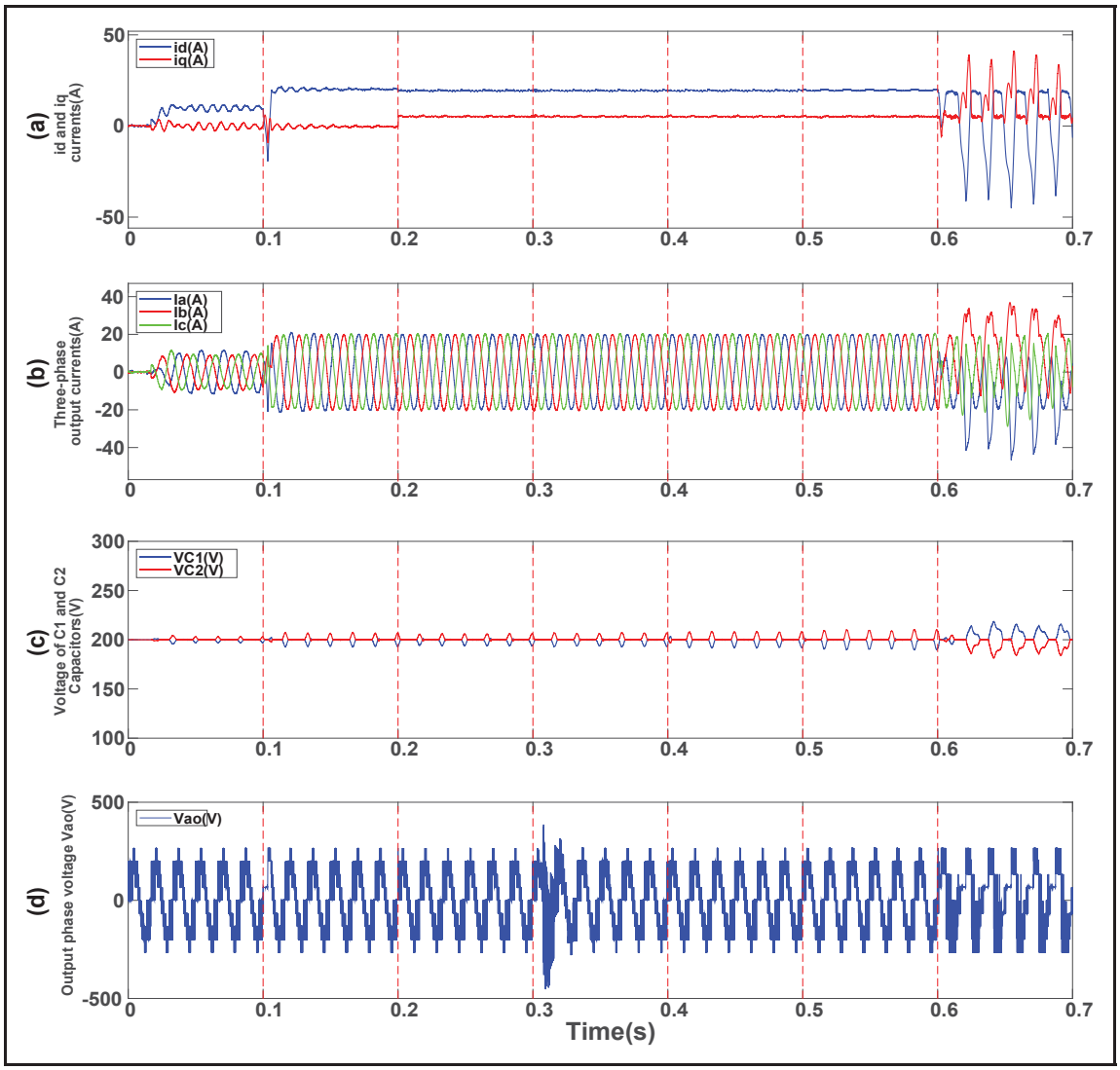

Figure 2.17 Waveforms of the output i_d and i_q currents (a); output three-phase currents (b); the voltage across the C_1 and C_2 capacitors (c); and V_{ao} the output phase voltage (d), when facing active power changes, reactive power changes, grid inductance increase, capacitor degradation, grid voltage increase and noise in measurements controlled by the MPC method

The inductance and resistance of the grid line are increased by 20% at $t = 300\,ms$. This scenario is designed to evaluate the performance of both methods under grid line parameter

variations. As seen in Figure 2.16 and Figure 2.17, this change has resulted in waveform distortion of the output current waveforms in both methods. This distortion of the waveforms is slightly worse in the DRL method. The waveform of the output phase voltage has distortion in both methods, but this distortion is mild in the DRL method but considerable in the MPC method. It is worth mentioning that these distortions are temporary and are eliminated after several sampling times. The waveforms of the voltages of the DC-link capacitors remain unchanged.

To simulate the performance under degraded capacitors, the capacitance of the DC-link capacitors is decreased by 15% at $t = 400\ ms$. As seen in Figure 2.16 and Figure 2.17, this change did not result in a considerable change in the output currents and the output phase voltage. However, the voltage ripples of the DC-link capacitors are increased, which is not unexpected. Thus, it can be concluded that both methods demonstrate resilience toward moderate degradation in the DC-link capacitors.

The grid voltage is increased by 10% at $t = 500\ ms$. As seen in Figure 2.16 and Figure 2.17, the output currents of the converter and the output phase voltage v_{ao} remain unchanged. The voltages of the DC-link capacitors remain unchanged in the DRL method, but their ripples are increased slightly in the MPC method. It can be concluded that both methods demonstrate resilience toward variations of the voltage grid.

Ultimately, the most important scenario is considered in this section to evaluate the resistance of both methods when facing a common uncertainty which is noise in measurements. A noise of 25 dB is added to the measurements of the output currents I_{abc} at $t = 600\ ms$. As seen in Figure 2.16 and Figure 2.17 as soon as the noise is introduced, the MPC controller becomes unstable. Not only do the output currents become severely distorted but also the output phase voltage v_{ao} is also heavily distorted and the voltage ripples of the DC-link capacitors are drastically increased. In the meantime, the DRL method is not affected by this uncertainty. This is due to its characteristic of being model-free and being able to generalize which are the inherent characteristics of machine learning algorithms.

In conclusion, based on the simulation results obtained in this section, it can be stated that the MPC method despite demonstrating satisfactory performance in many scenarios is not a reliable solution when moderate uncertainties are present in the control environment. This is due to the fact that the MPC method requires a precise model of the converter, and it assumes the measurements are ideally obtained with minimal noise or distortion. On the other hand, the DRL method demonstrates resilience toward various kinds of parameter changes and uncertainties, especially noise in measurements. Thus, the DRL method despite being more computationally intensive is the better choice for applications where a moderate degree of uncertainty is present.

Table 2.6 Experimental Test Parameters

Parameter	**Value**
DC-link voltage	400 V
Grid frequency	60 Hz
Grid filter inductance and resistance	5 mH, 0.1 Ω
DC-link capacitors	650 μF
Hardware sampling time	20 μs
Linear load inductance	50 mH
Linear load resistance	40 Ω
Nonlinear load	80 Ω, 2200 μF

2.5 Experimental Results

In this section, the proposed DRL method has been experimentally evaluated using an advanced testbed constructed based on dSPACE 1202, OP8662 (voltage and current measurements), a power board of NPC, an autotransformer, and several loads detailed in Table 2.6. The proposed experimental setup is illustrated in Figure 2.18, in which the test equipment

is highlighted. The intelligent controller has been trained and implemented using the "Reinforcement Learning Toolbox" provided by MATLAB.

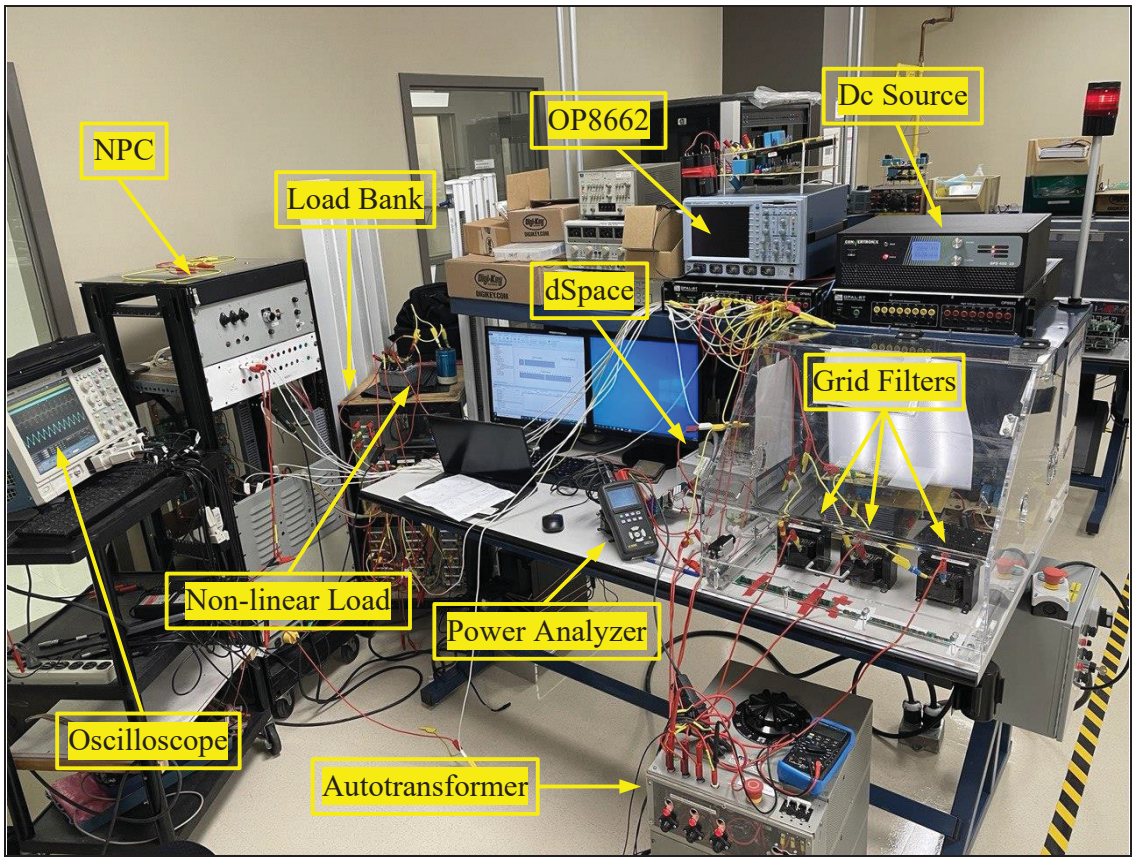

Figure 2.18 Details of the experimental setup built in the GREPCI laboratory to implement and evaluate the DRL control algorithm

The steady-state performance of the NPC converter in the grid-connected mode is obtained when i_{dref} is set to 12 A and i_{qref} is 0 A, as depicted in Figure 2.19. As seen, the agent effectively controls the NPC despite not having access to the mathematical model.

The dynamic performance of the proposed DRL controller under a step change of the *d*-reference current from 4 A to 12 A is shown in Figure 2.20. The captured results from this dynamic test demonstrate the fast dynamic performance of the intelligent controller.

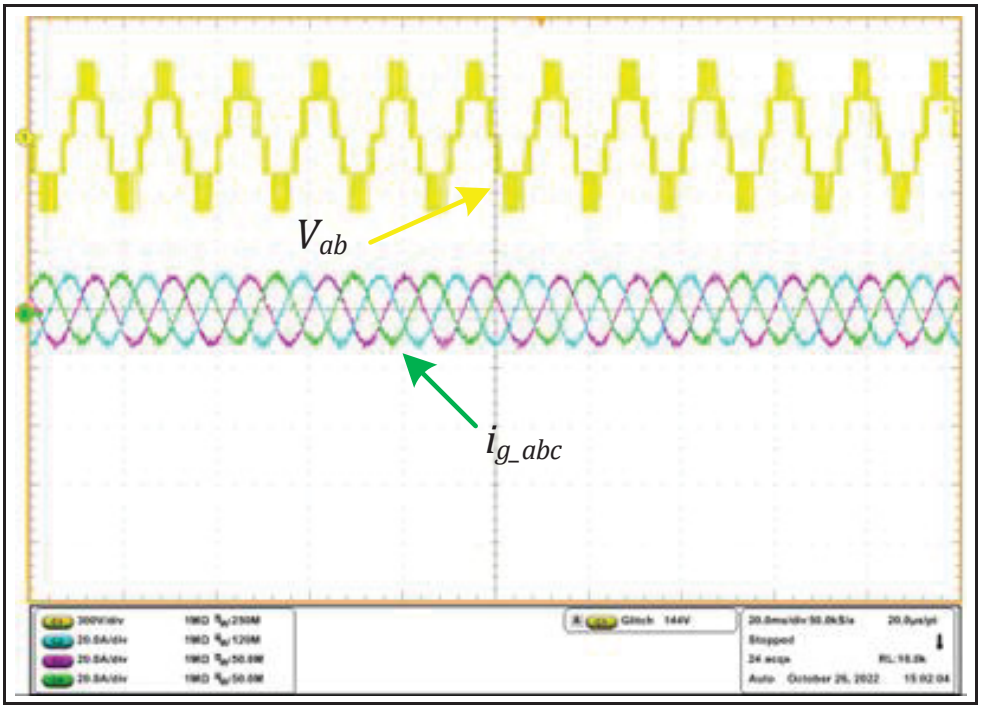

Figure 2.19 Experimental results of the line voltage and the three-phase current in steady-state mode when i_{dref} is set to 12A and i_{qref} is set to 0A

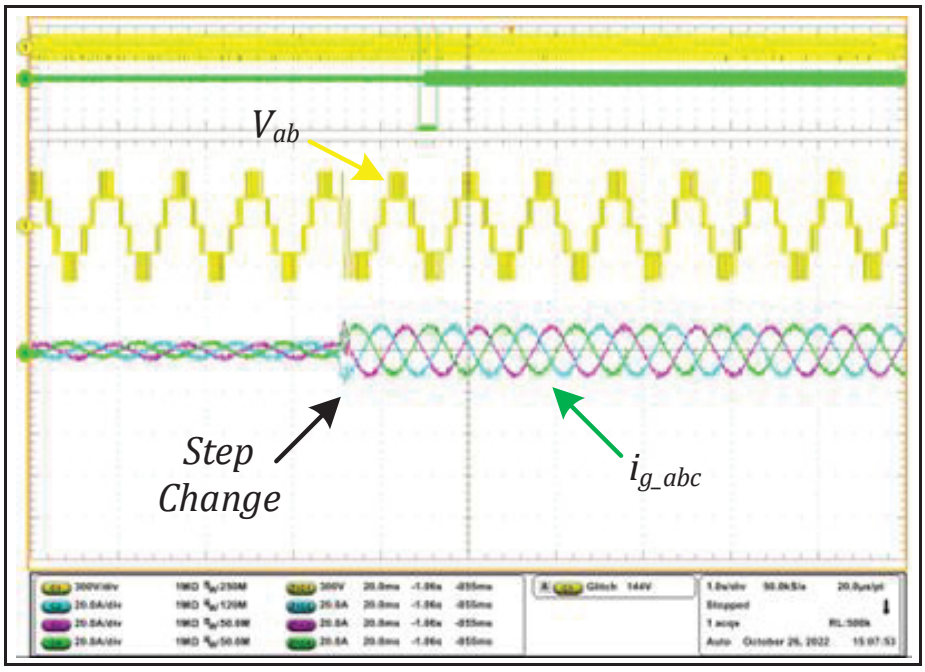

Figure 2.20 Experimental results of the line voltage and the three-phase currents under a step change of idref from 4A to 12A

Ultimately, to evaluate the performance of the proposed method in the presence of non-linearity, it is connected to a full-bridge non-linear load with a dc impedance of 80 Ω and 2200 μF. The dynamic response of the proposed method in this scenario is obtained when a step change of i_{dref} from 3 A to 12 A is applied, as depicted in Figure 2.21. Regarding the results of this test scenario, the proposed method demonstrates the robust performance of the DRL controller in the presence of perturbations caused by the non-linear load. In addition, Figure 2.21 proves that the controller can perfectly synchronize the converter's current with the grid voltage, regardless of perturbations caused by loads or the grid environment.

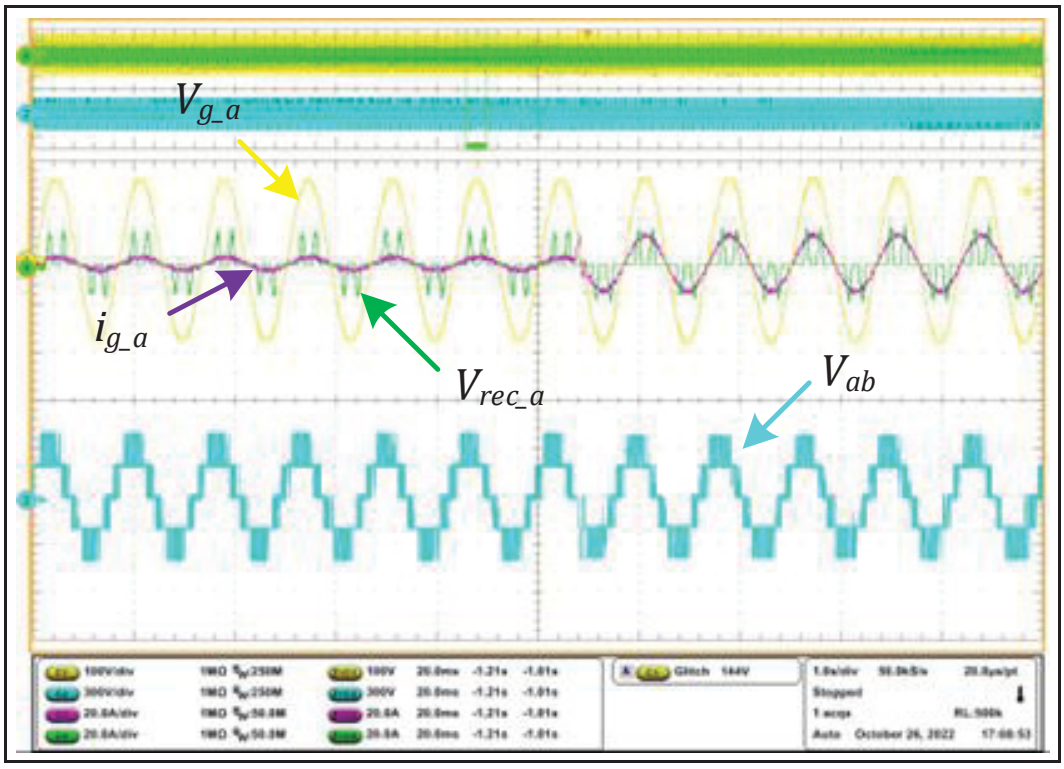

Figure 2.21 Dynamic performance of the proposed DRL controller under perturbations caused by a nonlinear load

The experimental results of another test scenario in Figure 2.22 indicate that the proposed DRL algorithm can successfully preserve the unity power factor of the grid by injecting reactive power demanded by loads. The THD analysis of the grid voltage and current in Figure 2.23

also validates the optimal switching control performance of the proposed DRL control algorithm in practice.

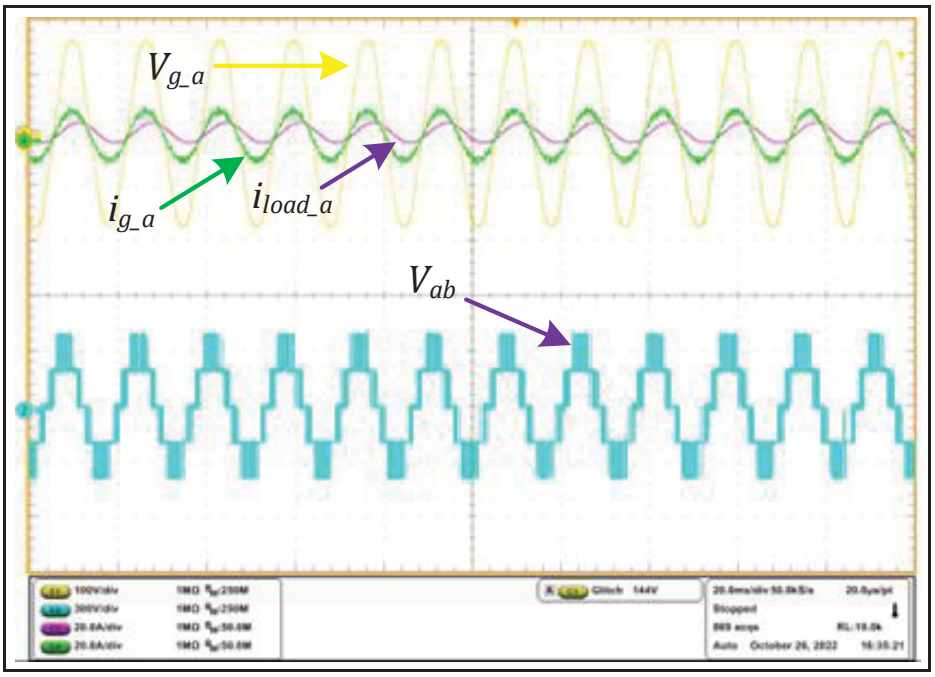

Figure 2.22 Experimental results of reactive power compensation under the operation of a reactive load

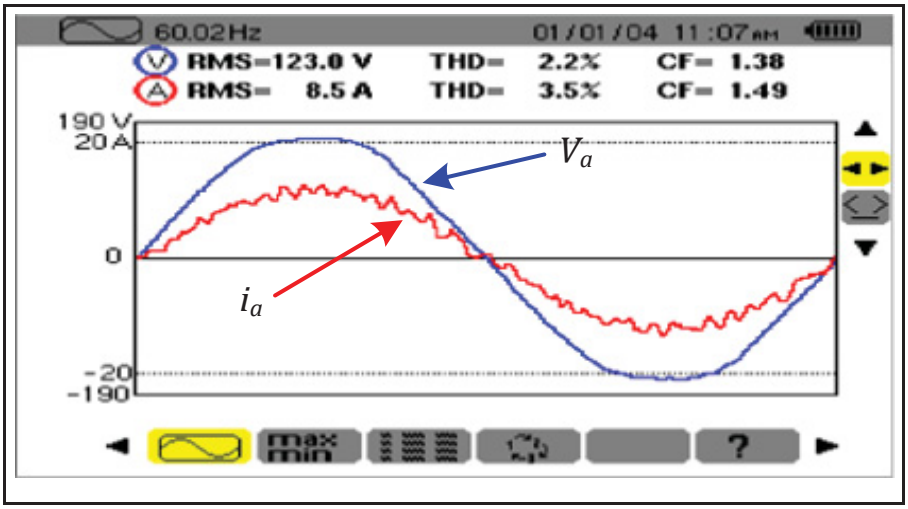

Figure 2.23 Experimental results of THD analysis of the grid voltage and current (phase A) under steady-state performance

2.6 Conclusion

In conclusion, this chapter presents a novel model-free switching and control method for a three-level NPC converter using deep reinforcement learning. The proposed method provides a robust performance under both simulation and experimental tests and achieves accurate voltage balancing and reference current tracking not only in steady-state mode but also under various dynamic changes, parameter variations, harmonic perturbations, and uncertainties. By applying various scenarios, the proposed method is compared against MPC, another conventional nonlinear control method for power electronic converters. The comparison results demonstrate the resilience of the proposed method against various types of parameter changes and uncertainties. The proposed DRL method is proven to be resilient against the presence of noise in sensor measurements whereas the MPC method becomes unstable in such an environment. However, it should be noted that this method requires long training times, and creating a reward function is very challenging. Nevertheless, future studies can be conducted to extend the application of this proposed method to other power electronic converters, especially those with complex control. Overall, this method can be considered a promising approach for model-free, non-linear control of power electronic converters, especially under parameter variations and uncertainties where conventional methods face performance and accuracy challenges.

Although the proposed method is implemented in a stand-alone system, this choice was intentional to provide a controlled environment for validating the effectiveness of the DRL-based approach without external influences. The stand-alone system allowed a detailed analysis of dynamic behavior, response to uncertainties, and operational challenges, ensuring a thorough evaluation of the proposed method's capabilities. The insights gained from this case study serve as a foundation for future extensions to grid-tied applications and electric vehicle rectifying modes. These extensions will further validate the robustness and practical applicability of the proposed DRL-based control approach in real-world scenarios where grid interactions introduce additional complexities and uncertainties.

CHAPTER 3

IMPLEMENTATION OF DEEP REINFORCEMENT LEARNING FOR MODEL-FREE SWITCHING AND CONTROL OF A 23-LEVEL SINGLE DC SOURCE HYBRID PACKED U-CELL (HPUC)

Pouria Qashqai [a], Mohammad Babaie [a], Rawad Zgheib [b], and Kamal Al-Haddad [a],

[a] Department of Electrical Engineering, École de Technologie Supérieure,
1100 Notre-Dame West, Montreal, Quebec, Canada H3C 1K3
[b] Institut de recherche d'Hydro-Québec (IREQ), Varennes, QC J3X 1S1, Canada

Paper submitted for publication in *IEEE Access*, March 2025

Abstract

In this chapter, a novel Deep Reinforcement Learning (DRL) method for controlling a 23-level Single DC Source Hybrid Packed U-Cell (HPUC) converter is proposed. The HPUC topology generates a high number of voltage levels with minimal components but presents control challenges due to its numerous switching states and dynamic charging behavior. Unlike conventional control methods, which require accurate models and are sensitive to noise and parameter mismatches, DRL offers a model-free and resilient approach to the non-linear control of such complex systems. A Deep Q-Network (DQN) agent which is inherently model-free and suited for high-dimentional state spaces and discrete action spaces, is employed to address these issues. To validate the proposed method, simulations were conducted in the MATLAB/Simulink environment. The obtained results demonstrated the satisfactory performance of the proposed DRL method, particularly its resilience in handling dynamic changes, parameter variations, and uncertainties.

3.1 Introduction

Power electronics converters play a crucial role in modern life due to their numerous applications such as renewable energies, smart grids, and electric vehicles (Impram, Nese, & Oral, 2020; P. Qashqai, Sheikholeslami, Vahedi, & Al-Haddad, 2015b; G. Zhang et al., 2018).

The high rate of integration of power electronics necessitates the implementation of advanced converter topologies as well as the integration of more efficient and reliable control methods (Krishna & Suresh, 2016; Mittal, Singh, Singh, Dixit, & Kumar, 2012; Singh, Gairola, Singh, Chandra, & Al-Haddad, 2008).

One of the prominent examples of such topologies is the 23-level Single DC Source Hybrid Packed U-cell (H-PUC) Converter (Sorto-Ventura et al., 2020). This converter is known for its reduced component count and the ability to generate a 23-level output voltage from a single DC source. The topology used in this converter combines two Packed U-cell (Vahedi, Kanaan, & Al-Haddad, 2015; Vahedi, Labbé, & Al-Haddad, 2015) converters as high-voltage low-frequency and low-voltage high-frequency sub-modules, in order to offer improved efficiency and reduced power losses while utilizing only twelve switches and three DC-link capacitors. Despite demonstrating promising performance using a relatively reduced number of components, this topology requires complex control due to its numerous switching states that demonstrate time-variant effects on the charging and discharging of the capacitors. Still more, the voltages of DC-link capacitors are dependent on the voltages of their upper capacitor(s) which increases the complexity of the control.

Linear control methods such as PID, have been the classic control methods in power electronics due to their simplicity and effectiveness in linear systems. They offer simple implementation and robust performance for systems with predictable behavior. However, their inability to handle complex dynamics and significant non-linearity inherent in modern power electronics limits their application to simple linear systems (Bacha et al., 2014).

To mitigate these limitations of linear control, nonlinear control methods are introduced in the literature. These methods are proposed to handle the inherent non-linearity and dynamic changes in power electronics, resulting in reliable and robust performance.

Nonlinear control methods, such as Feedback Linearization and Sliding Mode Control (SMC), are introduced in the literature but they come with other limitations. For instance, Feedback

Linearization is introduced as a non-linear control method that converts non-linear systems to their linear equivalent which makes the control design much less complex. However, this method requires an accurate model of power electronics, and it is sensitive to disturbances and uncertainties (Mahmud et al., 2019).

Unlike Feedback Linearization, Sliding Mode Control (SMC) is a non-linear control method that is resilient against parameter variations. However, SMC requires high switching frequencies which results in reduced efficiency (Komurcugil et al., 2020; Sebaaly et al., 2016).

As power electronics evolved into more complex and highly non-linear systems, the need for even more advanced non-linear control methods became more evident. Advanced control methods were introduced to handle these highly non-linear systems that, in many cases, exhibit time-variant behavior. Fuzzy Logic Control (Belaidi et al., 2012; M. Wang & Wang, 2020) is capable of handling uncertainties and performs using humanly understandable control rules. However, its performance and reliability highly depend on the quality of the control rules created.

Genetic Algorithm-Based Control (Maswood & Wei, 2005; V., 2017) methods take an approach based on natural evolution in genetics which provides powerful optimization but is computationally intensive and may not be suitable in power electronics systems where real-time response is utterly important.

Other advanced control methods such as H-Infinity Control (Mandal & Mishra, 2018; Mishra & Mandal, 2020; Patra, Garg, Panda, & Shukla, 2021), Predictive Functional Control (PFC) (H. Liu & Li, 2011), and Kalman Filter-Based Control (Çelik, Ahmed, & Meral, 2022) are introduced in the literature.

Artificial Neural Networks (ANN) are vastly used in power electronics for modeling (P. Qashqai, Al-Haddad, & Zgheib, 2020; Pouria Qashqai, Al-Haddad, et al., 2022; Pouria Qashqai et al., 2021; Pouria Qashqai, Zgheib, et al., 2022) as well as control (M. Babaie,

Sharifzadeh, & Al-Haddad, 2019; Mohammad Babaie, Saeidi, et al., 2020; Kůrková et al., 2018) purposes. Control methods based on ANN, consider power electronics as block-boxes. Thus, they can learn and generate control policies purely driven by measurement data, which makes them suitable for complex systems. However, these methods require extensive training data, which is a huge drawback in situations where such data is not readily available. Moreover, ANN-based control methods cannot adapt to uncertainties once they are trained which makes them susceptible to instability in "unseen" scenarios.

In this chapter, a popular non-linear control method known as Model Predictive Control (MPC) (Mohammad Babaie, Sharifzadeh, et al., 2020; Rodriguez, Garcia, Mora, Davari, et al., 2021; Rodriguez, Garcia, Mora, Flores-Bahamonde, et al., 2021) is used as a reference for performance comparison of the newly proposed DRL method. The main advantage of MPC is to predict the future behavior of a system and choose optimal control actions, which makes it a suitable method for handling non-linearities and constraints. Despite its advantages, MPC demands high computational power and an accurate model of the converter, which can be difficult or impossible. Moreover, this method is sensitive to parameter variations, noise in measurements, and uncertainties in the system.

Deep reinforcement Learning (DRL) (Yuxi Li, 2018), a subset of machine learning, is a modern control method that combines the generalization power of deep neural networks (LeCun et al., 2015) with reinforcement learning. It shows excellent potential in complex non-linear systems such as power systems where there is a notable degree of uncertainty in the environment as well as parameter variations (Pouria Qashqai et al., 2019). DRL works by learning the optimal policies of control through trial and error and interactions with the system. Despite requiring large training data, high computational power, and challenging reward function procedure, DRL is used in many areas that require advanced non-linear control such as robotics (Nguyen & La, 2019; W. Zhao, Queralta, & Westerlund, 2020), video games (Shao, Tang, Zhu, Li, & Zhao, 2019), electric vehicles (Yanni Wan et al., 2022), and large language models (LLMs) such as ChatGPT (Chang et al., 2023). Its applications in power systems and power electronics include but are not limited to the optimization of power distribution (Gao &

Yu, 2021), energy management of distributed energy sources (Lee & Choi, 2020), dynamic load management (Sheikhi, Rayati, & Ranjbar, 2016), Integrated Energy Systems (IES) management (Ting Yang, Zhao, Li, & Zomaya, 2021), and smart-grid operations (Yuanzheng Li et al., 2023).

Recent advancements in artificial intelligence, particularly machine learning (ML), along with increased commercially available computational power, have increased the popularity of AI in power electronics research (Pouria Qashqai et al., 2019). Some studies have investigated machine learning for modeling (Pouria Qashqai, Al-Haddad, et al., 2022; Pouria Qashqai et al., 2020b, 2021; Pouria Qashqai, Zgheib, et al., 2022), while others have applied it to improve power electronics control.

Only recently, DRL has been used as a method for control of power electronics converters (Cao et al., 2020b). In (Hajihosseini et al., 2020), DRL is applied as a solution to the shortcomings of conventional methods for control of DC/DC buck converter when feeding constant power load (CPL). For large dynamic changes in CPLs, conventional control algorithms often do not demonstrate satisfactory performance. Even though the DRL method proposed in this study has mitigated this issue and can in fact deliver satisfactory performance, the DRL agent is utilized for gain tuning in the feedback loop. Therefore, not only auxiliary control is required, but also it does not take advantage of a significant benefit of using DRL which is its model-free nature.

In (Tianxiao Yang et al., 2022), a model-free DRL control is proposed for a DC/DC buck converter with CPL which showed strong dynamic performance despite significant CPL changes. However, its reliance on precise measurements leads to sensitivity to noise and error accumulation between the trained model and the actual converter. In (Cui et al., 2022), DRL is used to control a DC/DC buck converter to be resilient against uncertainties and parameter changes. However, it relies on an offline pre-trained converter model for an extended state observer (ESO) to adjust to variations by observing differences between the model and the actual converter.

Research into applying DRL in power electronics is growing and gaining popularity. However, its advantages in control of power electronics converters are still in the early stages (Pouria Qashqai et al., 2019). Some studies, for instance (Alfred et al., 2021), focus on DRL control for simple converters without experimental results. Others, such as (Gheisarnejad et al., 2021), perform real-time simulations as a safe substitute for experimental results under harsh operational points. Some studies explore more complex converters like the Neutral Point Clamped (NPC) converter, as seen in (J. Wang et al., 2022) and (Pouria Qashqai et al., 2020a). While (Pouria Qashqai et al., 2020a) shows promising steady-state performance, it lacks testing under dynamic changes, noise, and uncertainties, and lacks comparison to conventional control methods such as MPC.

Traditional control strategies, such as Model Predictive Control (MPC), despite being effective, come with certain limitations and drawbacks such as the need for an accurate model of the converter, sensitivity to noise and parameter changes, and inability to adapt to uncertainties. These are limiting factors for practical real-world applications. DRL, on the other hand, takes a model-free approach that mitigates these limitations. Based on the control and switching method proposed by (Pouria Qashqai et al., 2020a), a DRL-based control method for NPC is developed by (Pouria Qashqai et al., 2023). The proposed method in this paper is shown to demonstrate satisfactory performance in steady state as well as dynamic changes. Moreover, the proposed DRL method is compared against MPC through simulation and experimental results, and it is proven that the DRL method is more resilient against uncertainties, noise, and parameter variations. However, the proposed method does not explore implementation on a fairly complex topology with demanding control considerations such as HPUC.

To explore this gap in the literature, this chapter proposes a similar implementation of DRL for the 23-level H-PUC converter, aiming to take advantage of DRL's adaptability and robustness to control an efficient converter that demands a challenging control. The main

objective of this research is to demonstrate the feasibility and effectiveness of DRL in switching and controlling the complex 23-level H-PUC converter.

This chapter is structured as follows. Section 3.2 introduces the modern 23-level HPUC topology, highlighting its strengths and weaknesses, especially emphasizing its control challenges. In section 3.3, the fundamentals of deep reinforcement learning are discussed and the proposed control method is presented in section 3.4. Moreover, Section 3.5 is dedicated to results. In this section, simulation results obtained from the MATLAB/Simulink environment are presented and analyzed. Lastly, the findings of this chapter are discussed in 3.6.

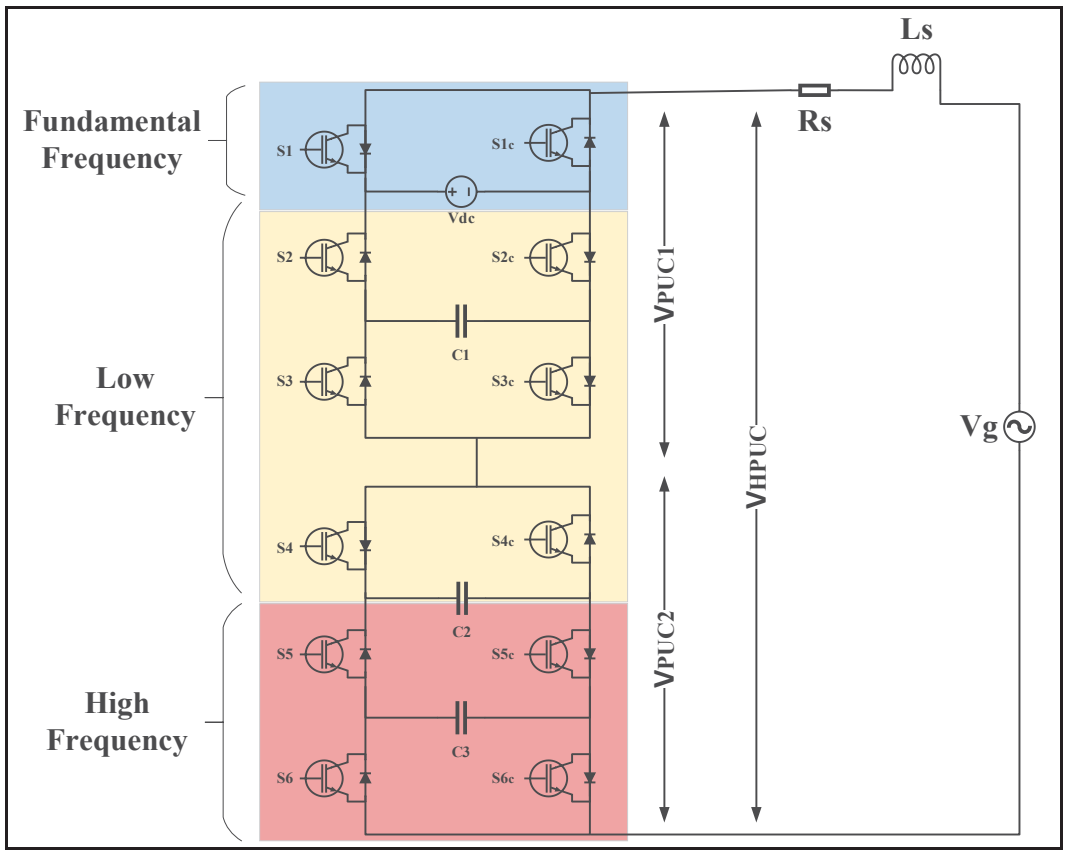

Figure 3.1 Topology of a 23-level Hybrid Packed U-cell (HPUC) comprised of two cascaded PUC5 converters

Table 3.1 Switching States and Capacitor Charging Directions of The Upper Leg (PUC1) and The Lower Leg (PUC2)

Upper Leg (PUC1)						
State	**S_1**	**S_2**	**S_3**	**C_1**		**V_{PUC1}**
1	0	0	0	-		0
2	0	0	1	↓		$-V_{C1}$
3	0	1	0	↑		$V_{C1} - V_{dc}$
4	0	1	1	-		$-V_{dc}$
5	1	0	0	-		V_{dc}
6	1	0	1	↑		$V_{dc} - V_{c1}$
7	1	1	0	↓		V_{C1}
8	1	1	1	-		0
Lower Leg (PUC2)						
State	**S_4**	**S_5**	**S_6**	**C_2**	**C_3**	**V_{PUC2}**
1	0	0	0	-	-	0
2	0	0	1	-	↓	$-V_{C3}$
3	0	1	0	↓	↑	$V_{C3} - V_{c2}$
4	0	1	1	↑	-	$-V_{c2}$
5	1	0	0	↑	-	V_{c2}
6	1	0	1	↓	↑	$V_{c2} - V_{c3}$
7	1	1	0	-	↓	V_{C3}
8	1	1	1	-	-	0

3.2 Single DC Source Hybrid Packed U-Cell (HPUC)

The Hybrid Packed U-Cell also known as HPUC (Sorto-Ventura et al., 2020) as shown in Figure 3.1, is a modern topology based on two cascaded Packed U-Cell (PUC) (Vahedi, Kanaan, et al., 2015) converters in order to generate multiple voltage levels in the output. Instead of a second DC source for the lower leg PUC converter, a capacitor is used. The PUC

in the upper leg is responsible for generating low-frequency voltage generation whereas the lower leg is mostly responsible for generating high-frequency voltages. This modification enables HPUC to generate multiple voltage levels using a single DC source despite making the control of the HPUC significantly more complex compared to PUC.

Despite providing these advantages over other combinations of PUC topology, voltage balancing in HPUC is rather challenging. As seen in Table 3.1 Switching States and Capacitor Charging Directions of The Upper Leg (PUC1) and The Lower Leg (PUC2), each sub-module consists of six switches and each switch has two states of "on" and "off". Thus the total number of combinations among the switches is $2^3 = 8$ per sub-module. Therefore, the whole topology has a combination of $8 \times 8 = 64$ switching states as shown in:

$$S_i = \bar{S}_i = \begin{cases} 1 & if\ S_i\ is\ on \\ 0 & if\ S_i\ is\ off \end{cases} \quad i = 1, 2, \dots, 6 \tag{3.1}$$

Where S_i and $\overline{S_i}$ are complementary switches that are never on and off simultaneously to prevent short circuits and other complications.

As shown in Table 3.1, depending on the instantaneous voltages across each capacitor and corresponding to different switching combinations, various currents will be drawn from them. Thus, the voltage of each capacitor is dependent on the capacitor in its upper position which makes voltage balancing relatively challenging. For this very reason, conventional linear control methods will struggle to deliver satisfactory performance when handling such a complex system let alone being resistant to uncertainties. To mitigate this issue, a model predictive control (MPC) capable of effectively regulating the capacitor voltages and efficiently generating the output voltages is proposed in (Sorto-Ventura et al., 2020). Using DRL the mathematical model of the HPUC converter is not required. Similarly, voltage balancing equations are not required to be obtained because the DRL agent spontaneously discovers the optimal policy for voltage balancing based on observations and reward functions. Therefore, the inherent model-free characteristics of the DRL method can provide advantages

in scenarios where a model of the HPUC converter is either not readily accessible or obtaining one proves to be challenging. In the next section, the DRL method is described in detail.

3.3 Fundamentals of Deep Reinforcement Learning (DRL)

In this section initially the fundamentals of deep enforcement learning will be introduced. Then the proposed control method based on DRL for intelligent control of HPUC will be discussed. Lastly, the advantages and limitations of the proposed DRL method will be discussed.

3.3.1 Reinforcement Learning

Deep Reinforcement Learning (DRL) combines deep neural networks (DNN) with reinforcement learning (RL) to control complex and non-linear systems. In DRL, as shown in Figure 3.2, an agent interacts with the environment and selects actions based on the observed states and received rewards. The objective is to update the policy of $\pi(s, a)$ in each iteration to achieve the optimal policy based on actions A_t applied to the environment and inspecting the observations S_t and calculating the instantaneous and accumulative rewards of R_t where t is the time step of the experience episode.

Reinforcement learning can be expressed as a four-tuple Markov Decision Process (MDP) of $\{s, a, pa, ra\}$ to find an optimal policy of π^*, to maximize long-term rewards as shown in (3.2):

$$\pi^* = argmax_\pi \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r_t | s_0 , \pi\right] \tag{3.2}$$

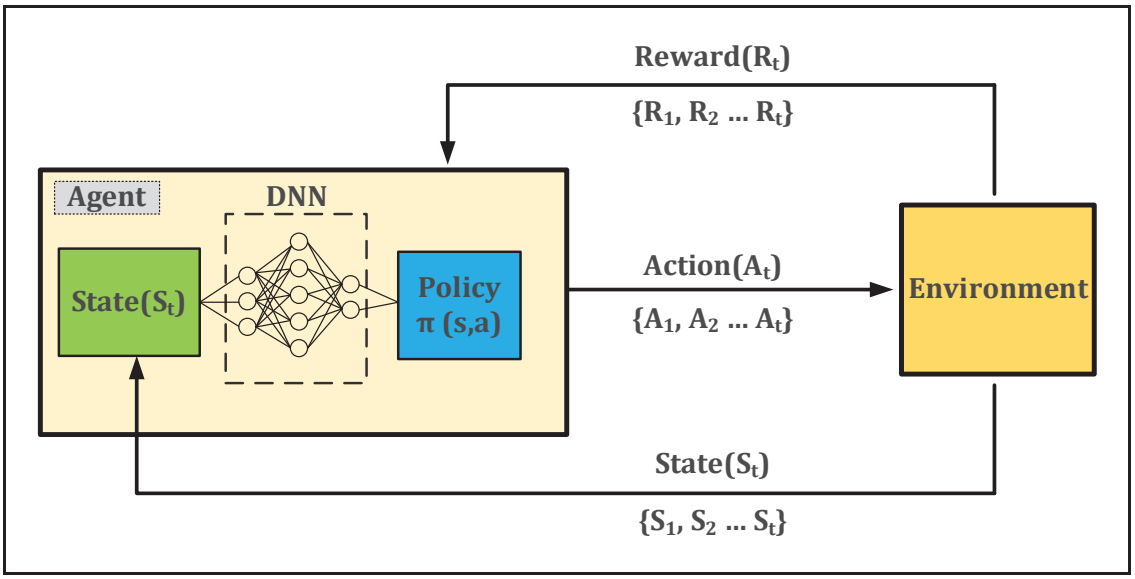

Figure 3.2 Diagram of a deep reinforcement learning agent aiming to optimize policy of $\pi(s, a)$ at the time step

Where π^* is the optimal policy, t is the time step. γ is the discount factor which is a number between 0 and 1 indicating how actions affect the accumulative reward in long-term. Ultimately, s_0 is the initial state and r_t is the instantaneous reward at the time step of t.

There are various algorithms to achieve the aforementioned optimal policy for a given RL agent. The most prominent ones in the literature are as follows:

- **Value-based** which solves the Bellman optimality equation to find the best value function for selecting actions in each state.

- **Policy-based** which alternates between estimating and improving a policy based on its value function.

- **Q-Learning** which is a model-free and off-policy method that estimates the value of state-action pairs and updates them based on its calculated rewards.

- **Actor-Critic Methods** which combines policy-based (actor) and value-based (critic) methods, where the actor chooses actions and the critic assesses them.

3.3.2 Q-learning

Both value iteration and policy iteration algorithms are model-based. Unlike such algorithms, Q-learning (Jang et al., 2019) is an off-policy model-free algorithm that learns the action-value function directly from experience episodes without requiring an accurate model of the environment. It approximates the values i.e. "Q-values" corresponding to taking each action in a given state. Q-learning uses (3.3) to update Q-values in each iteration of interactions with the environment.

$$Q(s,a) \leftarrow Q(s,a) + \alpha[R(s,a) + \gamma \max_{a'} Q(s',a') - Q(s,a)] \tag{3.3}$$

Where, $Q(s,a)$ is the Q-value for taking the action of a in the state of s, while α is the learning rate, $R(s,a)$ is the instantaneous reward after taking action of a in the state of s, γ is the discount factor and ultimately, a' and s' are the next action and state after taking action a in the state of s. Using this equation, the new value of $Q(s,a)$ is updated after each iteration.

3.3.3 Deep Reinforcement Learning

As stated earlier, conventional reinforcement learning algorithms use numerical methods to find the optimal policies or values. Despite being easier to implement and less demanding in terms of computational burden, such algorithms are not suited for fairly complex and high-dimensional applications. To mitigate these limitations, deep neural networks (DNNs) are integrated into reinforcement learning, giving rise to a modern approach known as "Deep Reinforcement Learning". Taking advantage of neural networks not only enables DRL to handle high-dimensional and complex problems but also leverages the data-driven nature of neural networks. Being data-driven empowers DRL algorithms to learn optimal control of various applications without acquiring an accurate model of their corresponding systems.

Additionally, neural networks are excellent for generalization. Thus, these types of control methods are well suited for non-linear control schemes where unseen data is likely to degrade the performance of conventional non-linear control methods.

In this chapter, a DNN-based Q-learning method called Deep Q-Network (DQN), is used due to its suitability for model-free environments and discrete action spaces.

3.4 The Proposed Control Method Based on Reinforcement Learning

Using an approach similar to (Pouria Qashqai et al., 2023), a control method based on DRL is developed and proposed in this section.

3.4.1 Introduction

The overall diagram of the proposed method is shown in Figure 3.3. As seen, the proposed method consists of a pre-processing block that feeds the observation signals obtained via measurements to the DRL agent. Using these signals that act as the state space, the agent can estimate the model of the converter and learn its dynamic behavior. This block is equal to the input layer of the DNN block as shown in Figure 3.2. The DRL agent finds the optimal control policy based on previous experience, current observations, and the accumulated reward.

The part of the DRL agent block responsible for the generation of actions is equal to the output layer of the DNN. To improve efficiency and to reduce training times of the agent. Two actions are generated. One for the upper leg and one for the lower leg of the HPUC. In the literature, this approach is known as multi-dimensional action space. This enables the agent to observe and learn the effect of switching states on each corresponding leg, resulting in a faster convergence to the optimal switching. Finally, the “actions to switch” block takes the actions from the DRL agent which are discrete values, usually integers. This block then maps these values to their corresponding switching states for the upper leg and lower leg of the HPUC using a look-up table.

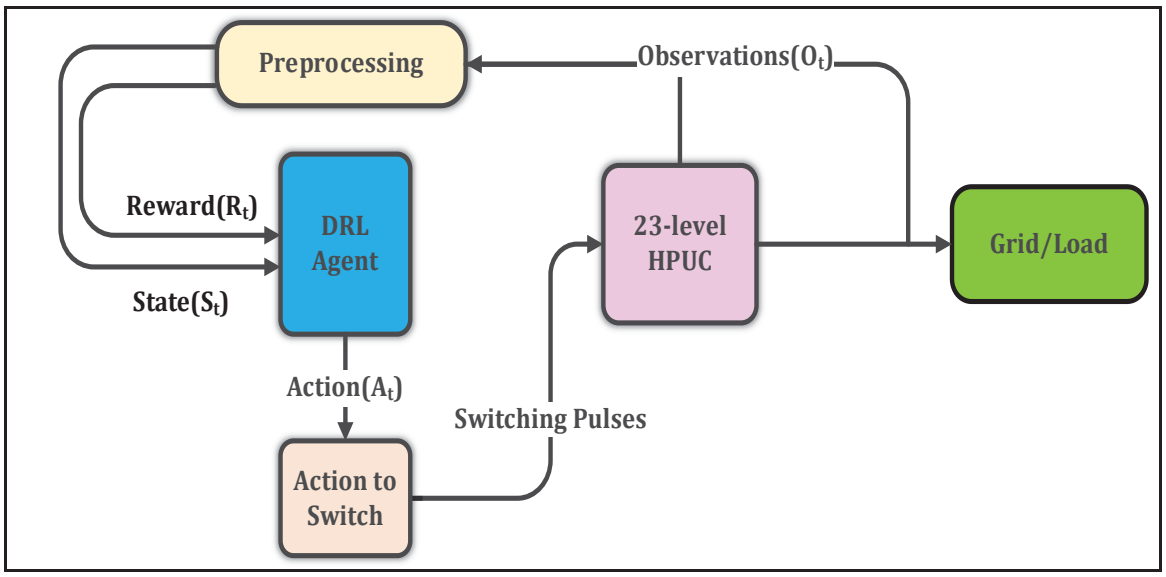

Figure 3.3 Block diagram of the proposed DRL method for model-free control of a grid-connected 23-level single DC source HPUC

In the following subsections, initially, the agent type is selected and justified. Moreover, observations (signals measured) are selected to generate the state space, and the idea behind choosing each one is discussed. Furthermore, to effectively achieve the optimal control policy the reward functions specific to this application are designed. Lastly, the action space and the method of mapping the numerical actions generated by the proposed DRL agent to the switching gates of the HPUC converter are introduced.

3.4.2 Choosing the agent type

The most popular agent types supported by MATLAB ("Reinforcement Learning Agents - MATLAB & Simulink," n.d.) with their characteristics are described in Table 3.2. An off-policy agent is well-suited for the model-free approach proposed in this chapter. Q-learning-based agents are preferred in comparison to actor-critic-based agents due to the sensitivity of the actor-critic approach towards hyperparameters. In this chapter, Deep Q-Network (DQN), a form of Q-learning algorithm is used for training the DRL agent. The DQN agent type is not only model-free but also is excellent for handling high-dimensional state spaces in which,

action space is discrete. These characteristics make it meet the requirements for model-free control of a complex converter with discrete action space (i.e. switching states) such as HPUC.

Table 3.2 Popular DRL Agent Types Supported by MATLAB

Agent	**Type**	**Action Space**	**On/Off Policy**
Q-Learning (Q) Agents	Value-Based	Discrete	Off
SARSA Agents	Value-Based	Discrete	On
Deep Q-Network (DQN) Agents	Value-Based	Discrete	Off
Policy Gradient Agents (PG)	Policy-Based	Discrete or continuous	On
Actor-Critic Agents (AC)	Actor-Critic	Discrete or continuous	On
Deep Deterministic Policy Gradient (DDPG) Agents	Actor-Critic	Continuous	Off
Twin-Delayed Deep Deterministic Policy Gradient Agents (TD3)	Actor-Critic	Continuous	Off
Soft Actor-Critic Agents (SAC)	Actor-Critic	Continuous	Off
Proximal Policy Optimization Agents (PPO)	Actor-Critic	Discrete or continuous	On
Trust Region Policy Optimization Agents (TRPO)	Actor-Critic	Discrete or continuous	On
Model-Based Policy Optimization Agents (MBPO)	Actor-Critic	Discrete or continuous	Off

3.4.3 Creating the state space (observations)

To effectively capture the dynamic behavior of the converter, different measurements are required to be applied to the DRL agent. These measurements are called observations in the literature. The following equation demonstrates the observation array of the DRL agent:

$$O_t = [i_d, v_g, v_o, V_{DC}, i_{dref}, v_{c1}, v_{c2}, v_{c3}] \tag{3.4}$$

Where O_t is the observations array, i_d is the output current, v_g is the grid voltage, v_o is the output voltage of the converter, V_{DC} is the input DC voltage, i_{dref} is the reference current, and finally v_{c1}, v_{c2}, and v_{c3} are the voltages across capacitors c_1, c_2 and c_3, respectively.
As illustrated in Figure 3.4, the pre-processing unit manipulates the measurements provided by the observations unit to form the ultimate state space of the controller. The generated state space of the controller, marked as S_t is shown in (3.5):

$$\begin{aligned} S_t = [&i_d, v_g, v_o, v_{c1}, v_{c2}, v_{c3}, i_{derr}, \int i_{derr}, i_{dref}, \\ &\omega t, v_{c1err}, v_{c2err}, v_{c3err}, \int v_{c1err}, \int v_{c2err}, \int v_{c3err}] \end{aligned} \tag{3.5}$$

Where i_{derr} is the instantaneous error between the reference current i_d and actual output current i_g while i_{derr} is its accumulative error. Similarly v_{c1err}, v_{c2err}, and v_{c3err} are the errors between actual and reference voltages of capacitors, c_1, c_2, and c_3 respectively while $\int v_{c1err}$, $\int v_{c2err}$, and $\int v_{c3err}$ are their accumulative counterparts. Ultimately ωt is the angular speed of the grid voltage equal to $2\pi f$ where f is the line frequency of v_g.

The diagram of the observations and pre-processing block is depicted in Figure 3.4. As seen, in order to capture the accumulative effect of the regulated parameters (i.e. i_d, v_{c1}, v_{c2}, and v_{c3}) their accumulative errors are also calculated and added as state parameters. Principally, only six signal measurements and two variables are needed for this control scheme. However,

the state array connected to the DRL agent has 16 elements as shown in Figure 3.4. Although raw measurement signals can be likewise utilized as state space, the pre-processing unit provides more insight into the dynamics of the converter for the DRL agent and improves its efficiency, and reduces the training times significantly.

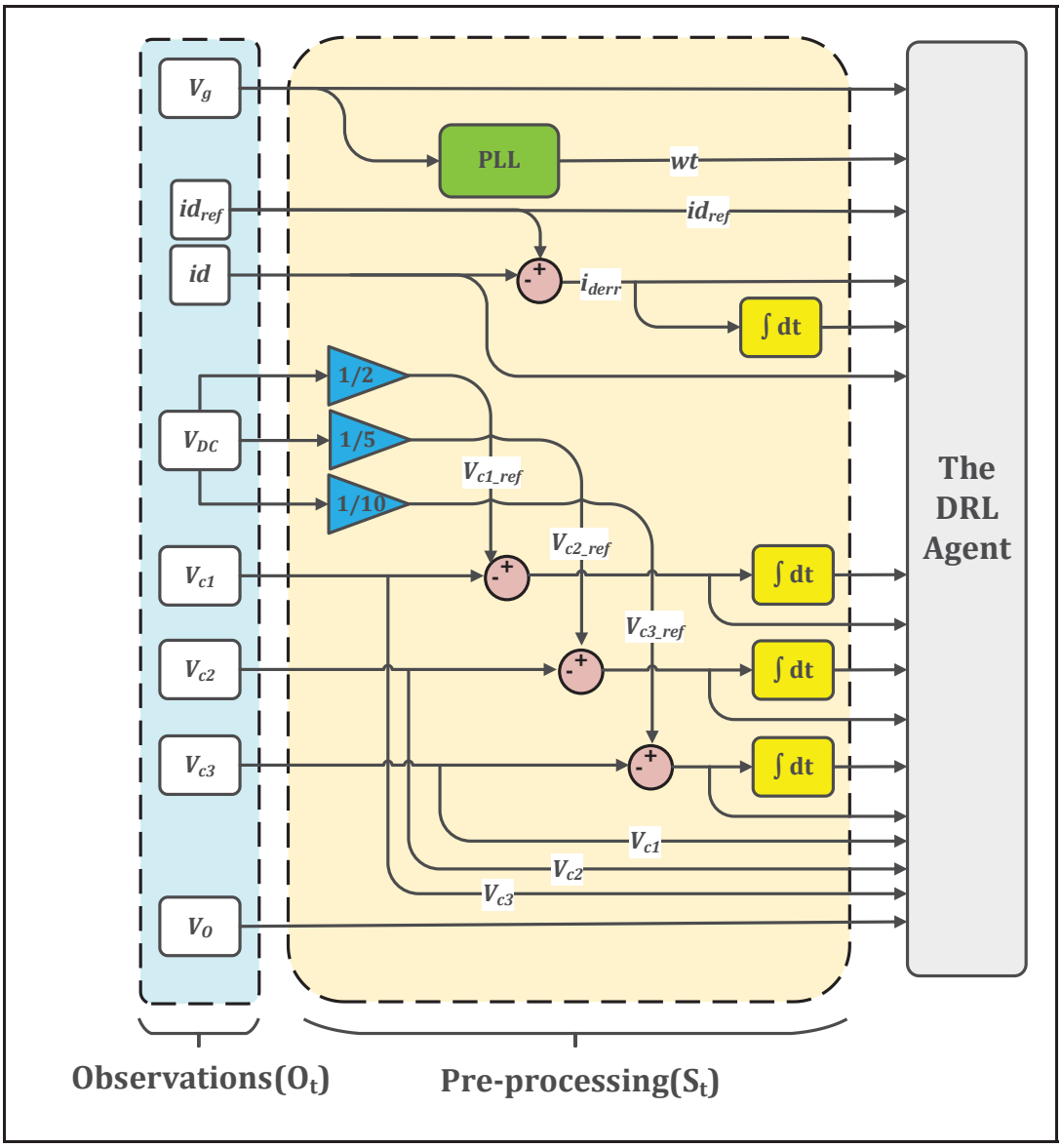

Figure 3.4 Block diagram of the state space generation unit comprised of observations (measurements) and pre-processing

3.4.4 Reward function

The most challenging part of a deep reinforcement learning application is to design a reward function. The challenge arises from the fact that the optimal operation may seem intuitive to humans with greater insight but the DRL agent may struggle to find an optimal policy through mathematical equations of its reward function.

On the other hand, a poorly designed reward function can result in a policy that leads to the optimal reward by exploiting the system in ways that are not desirable or sometimes even hazardous. For instance, let's consider a DRL agent is used to control a Buck converter, and its reward function is designed to aim for minimizing loss and puts significantly high emphasis on achieving this goal. Therefore, there is a chance that the DRL agent may find out that there is a hefty penalty for power loss thus it may opt to avoid turning on its switch(es) to achieve the highest reward possible. This example demonstrates the importance of designing carefully curated reward functions as the most important step in implementing DRL for the control of various applications including power electronics.

In this chapter, a reward function is designed that empowers the DRL agent to achieve four objectives efficiently while prioritizing each objective based on its importance.

There are four objectives. For $\forall i \in \{1,2,3,4\}$ let R_i be an objective and $\Delta(i)$ be its corresponding error, aiming to minimize the errors between the reference values and the actual values of i_d, v_{c1}, v_{c2}, and v_{c3}, respectively. The reward function of each objective is presented as a conditional equation as depicted in (3.6).

$$R_i = \begin{cases} 1 - \log_{10}|10\Delta(i)| & if\ 1^{-4} < |\Delta(i)| < 1 \\ 1 - \sqrt{|\Delta(i)|} & if\ \ |\Delta(i)| \geq 1 \\ 1 - \log_{10} 1^{-3} & otherwise \end{cases} \tag{3.6}$$

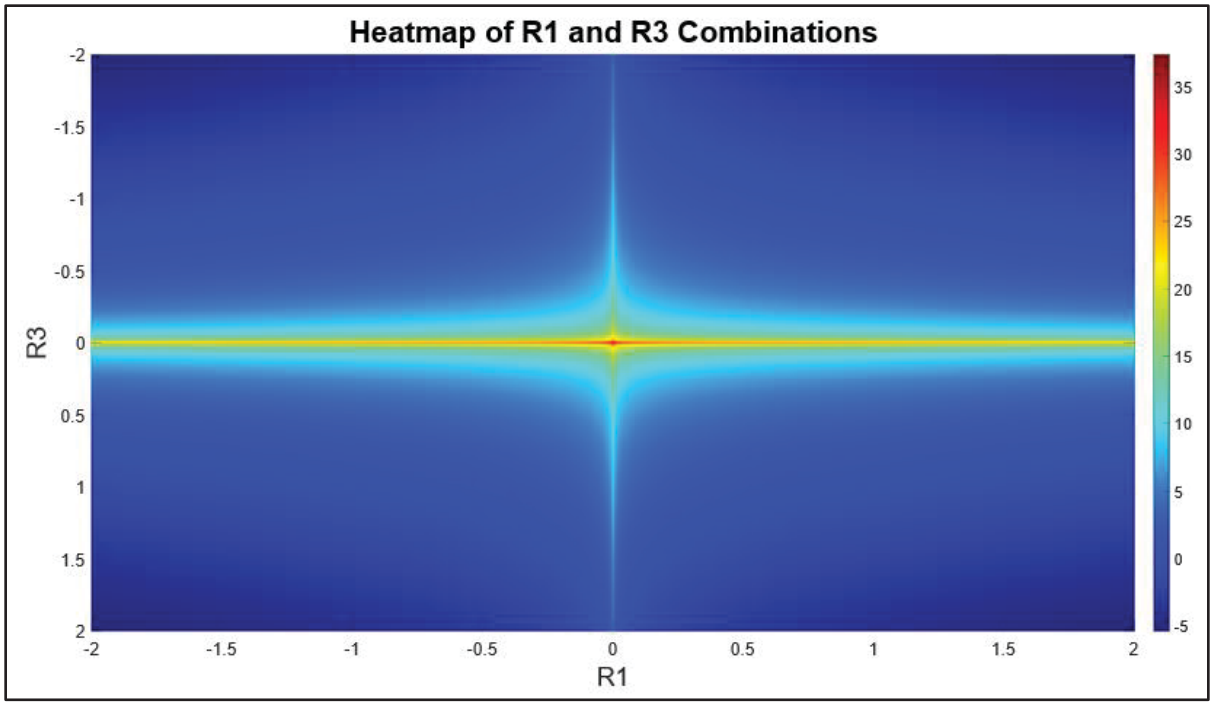

Figure 3.5 Heatmap of the reward functions R_1 and R_3 representing $\Delta(i_d)$ and $\Delta(v_{c3})$

As seen, the reward for errors greater than one is negative, which means the DRL agent receives "punishment" for deviating too far from the reference values. Moreover, for errors smaller than one and greater than 1^{-4}, the reward function rises exponentially enabling fine-tuning in these areas. A constant value of $1 - \log_{10} 1^{-3}$ is applied to errors smaller than 1^{-4} to avoid astronomically large numbers for near-zero errors. Omitting this constant value will result in unstable training.

As mentioned earlier, R_i is merely the fundamental reward function and each objective has a unique reward function which is a coefficient of R_i. The coefficients ∂_f for $\forall f \in \{i_d, v_{c1}, v_{c2}, v_{c3}\}$ are selected based on the priority of each objective. In this chapter, the following coefficients are selected for each objective $\partial_{i_d} = 10$, $\partial_{v_{c1}} = 5$, $\partial_{v_{c2}} = 3$, and $\partial_{v_{c3}} = 2$. Considering these coefficients, the ultimate reward function is generated as shown in (3.7):

$$R = \partial_{i_d} R_1 + \partial_{v_{c1}} R2 + \partial_{v_{c2}} R3 + \partial_{v_{c3}} R4 \tag{3.7}$$

To demonstrate the exploration environment of the agent, a heatmap for a combination of R_1 and R_3 is illustrated in Figure 3.5. As seen, the agent gives higher priority to achieving the minimum error for i_d which is represented by R_1 and v_{c1} which is represented by R_3. As you can see, the DRL agent puts a higher priority on regulating the output current i_d in comparison to balancing the voltage of v_{c1}.

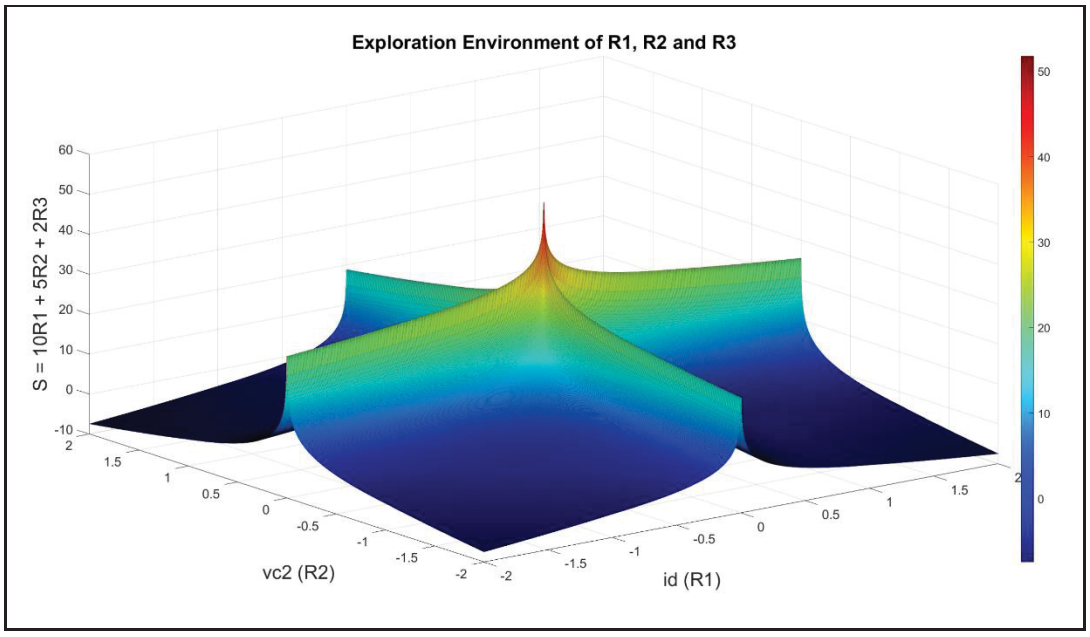

Figure 3.6 Exploration environment of a combination of the reward functions R_1, R_2, and R_3 representing $\Delta(i_d)$, $\Delta(v_{c2})$, and $\Delta(v_{c3})$

Ultimately, since depicting a four-dimensional diagram is not feasible, to visualize the reward exploration environment that the DRL agent can explore, it is illustrated as a three-dimensional map of exploration between the reward functions R_1, R_2, and R_3 as shown in Figure 3.6. As seen, the agent explores the reward function with a low reward variation for medium and large errors resulting in a plateau of rewards when all objectives deviate too far from their reference values. Once the agent reaches areas that correspond to minimal errors, it receives higher rewards. The overall reward of "R" peaks when all objectives have near-zero errors. This configuration not only encourages the agent to stay in near-zero error areas but also allows it

to fine-tune the policy to achieve the smallest error possible while respecting the priority of accomplishing each objective.

3.4.5 Action space

In order to generate the action space, consider the topology of the HPUC converter as shown in Figure 3.1. As stated earlier, the HPUC converter is comprised of 12 switches, 6 on each submodule. Since each switch and its complementary pair have two states of On and Off. Therefore, there are $2^3 = 8$ combinations at each submodules (PUC1 and PUC2) of the converter. However, only 8 distinct voltage levels are generated by different combinations of the switching states as is listed in Table 3.1. Thus, the total number of permitted switching states (i.e. $8 \times 8 = 64$) is considered for creating the action space A_t. Since these switching states are incomprehensible for the DRL agent, an integer number is used to represent each combination. Thus, the discrete two-dimensional action space in this method would be a matrix as shown in (3.8).

$$A_t = \begin{bmatrix} 0 & 1 & \dots & 6 & 7 \\ 0 & 1 & \dots & 6 & 7 \end{bmatrix} \tag{3.8}$$

A lookup table linked to the DRL agent's output is utilized to map each action, represented as an integer as shown in (3.8), into its corresponding switching state. This process is illustrated in Figure 3.3. As seen, the "Action to Switch" block enables the mapping of actions to switching signals.

3.4.6 Finalizing the DRL controller

Figure 3.7, depicts the schematic of the proposed DRL controller connected to the 23-level single DC source HPUC. The actions generated by the DRL agent remain unchanged until the following sampling period of the agent, which may be different from the simulation's sampling

time. Therefore, the DRL agent's sampling period can be considered as the switching frequency (f_{sw}) of the converter.

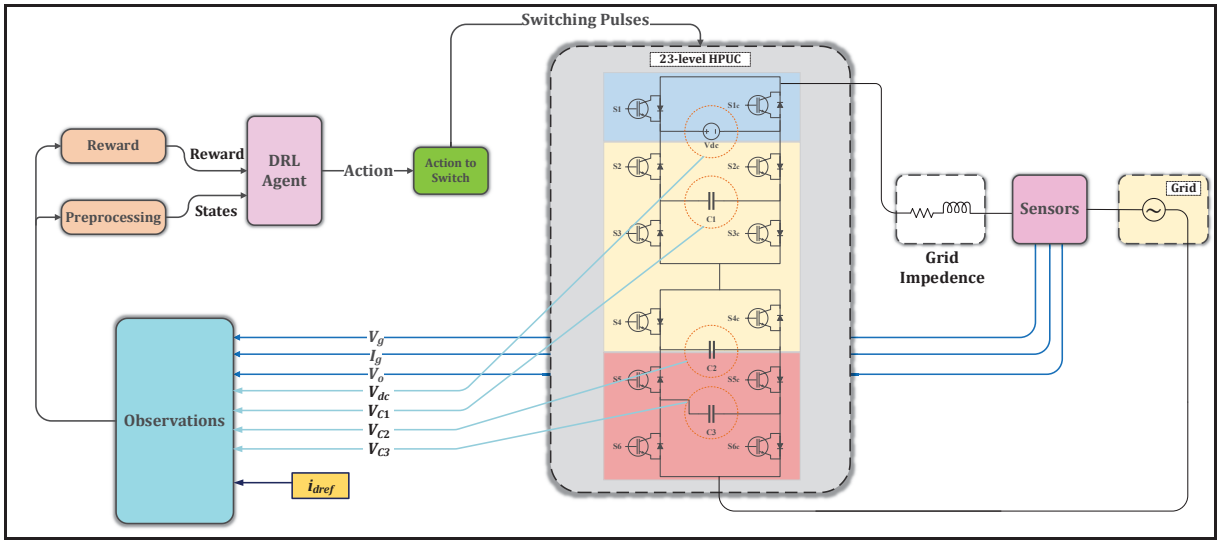

Figure 3.7 Diagram of a 23-level single DC source HPUC, controlled by the proposed DRL method

Table 3.3 Simulation Parameters

Parameter	Value	Unit
Peak Grid Voltage	170	V
Grid Frequency	60	Hz
Grid Resistance (per phase)	0.1	Ω
Grid Inductance (per phase)	5	mH
DC-link Capacitors	2000	µF
Simulation Time Step (Ts)	50	µs
DC-link Voltage	160	V

3.5 Results

In order to assess the performance of the proposed approach against a conventional method like MPC in controlling a 23-level HPUC converter, both methods are implemented in the Matlab/Simulink simulation environment. Different operation modes, dynamic changes, noise in measurements, parameter changes, and uncertainties are studied.

Table 3.3 and Table 3.4 illustrate the simulation parameters and the training parameters, respectively. By applying these values along with the previously obtained reward function, the following results are obtained in various operation conditions.

Table 3.4 Training Parameters

Parameter	**Value**
Layer Size of the State Path	[224,224]
Layer Size of the Action Path	124
Learning Rate	0.01
Gradient Threshold	1
Normalization	None
Bias Learn Rate Factor	0
Double DQN	No
Target Smooth Factor	0.001
Experience Buffer Length	10^5
Mini Batch Size	640
Discount Factor	0.9
Score Averaging Window Length	5
Epsilon Greedy Exploration Epsilon	1
Epsilon Decay	0.05
Epsilon Min	0.01
Agent Sampling Time	100μs

3.5.1 Steady-state operation

To evaluate the steady-state operation, the performance of the proposed DRL method is examined. Figure 3.8 illustrates the steady-state operation of the HPUC converter using DRL. In addition to demonstrating satisfactory steady-state performance, the DRL method shows a relatively small rising time. The Total Harmonic Distortion (THD) of the output current is measured as 2.71% ensuring that THD remains within the acceptable range of standards.

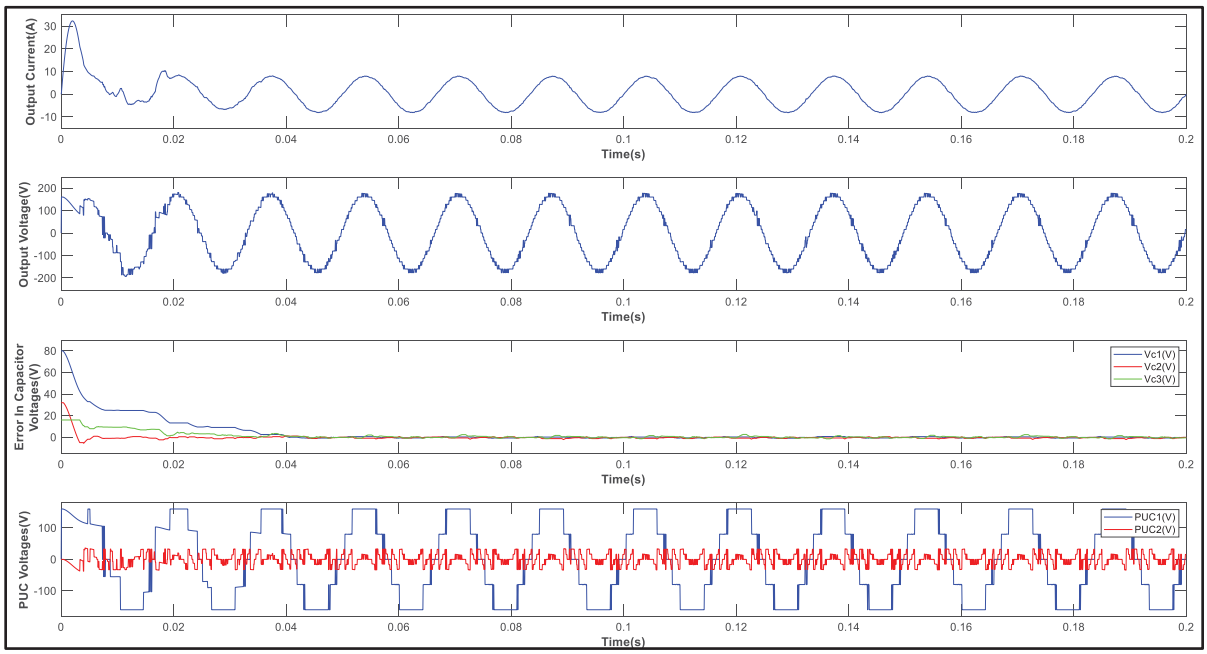

Figure 3.8 Waveforms of the output current(A), output voltage(V), errors of the DC-link capacitors(V), and voltages across submodules PUC1 and PUC2 in steady-state operation controlled by the DRL agent

3.5.2 Dynamic response

Additionally, in order to assess the dynamic response of the DRL method, step changes in the active and reactive power delivered to the grid using the HPUC converter are studied. As shown in Figure 3.9, a step change from $i_d = 10A$, $i_q = 0A$ to $i_d = 8A$, $i_q = 0A$ at t=200ms

and another one to $i_d = 10A$, $i_q = 0.4A$ at t=300ms are applied to the HPUC converter controlled by the DRL agent. The results demonstrated good performance when facing these dynamic changes.

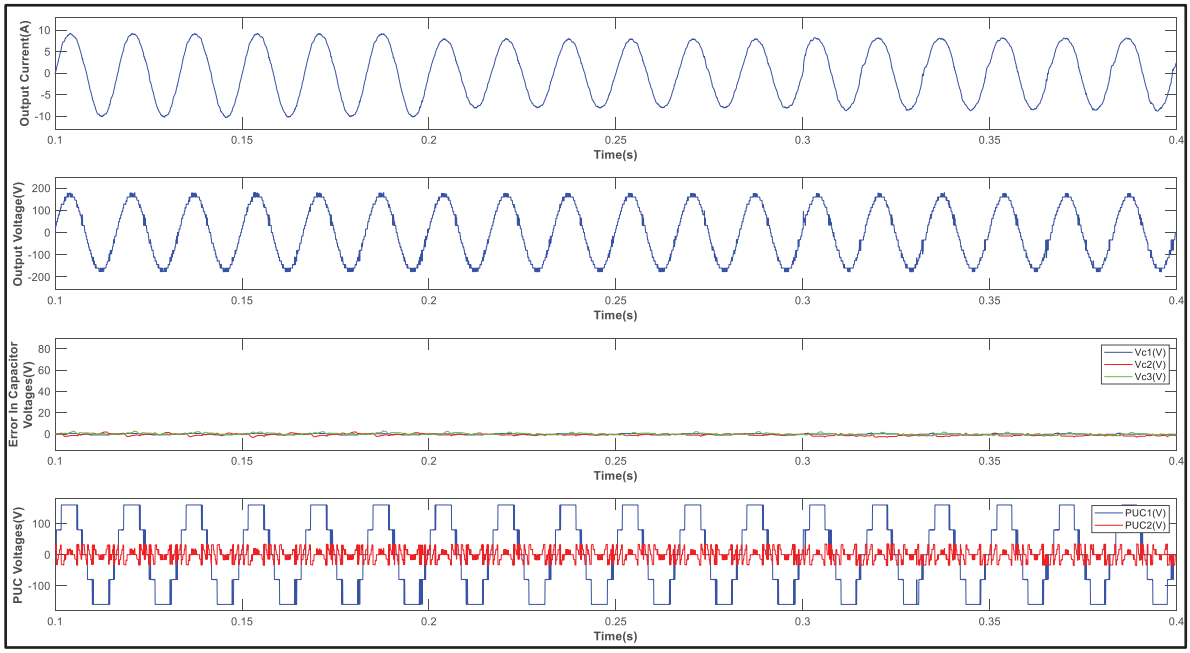

Figure 3.9 Waveforms of the output current(A), output voltage(V), errors of the DC-link capacitors(V), and voltages across submodules PUC1 and PUC2 controlled by the DRL agent facing step changes from $i_d = 10A$; $i_q = 0A$ to $i_d = 8A$; $i_q = 0A$ at t=200ms and $i_d = 10A$; $i_q = 0.4A$ at t=300ms

3.5.3 Noise, parameter variations, and uncertainty

The main objectives of utilizing DRL for the control of power electronic converters, such as HPUC, are its model-free approach and the generalization characteristics of this control method. In the previous sub-sections, the satisfactory performance of the DRL agent without prior knowledge of the topology under steady-state as well as the dynamic changes of the aforementioned converter are demonstrated. In this section, the converter is utilized under various non-regular operational points to assess the resilience of the DRL method under uncertain conditions.

The HPUC converter is set to deliver a current of $i_d = 6A$ to the grid using the DRL agent as shown in Figure 3.10. At t=100ms, an active power increase of $i_d = 10A$ is administrated. As seen, the DRL agent followed this dynamic change closely. Moreover, a reactive power increase from $i_q = 0A$ to $i_q = 0.2A$ is added at t=200ms, and the agent displayed a favorable response to this step change.

Moreover, a white Gaussian noise (WGN) is added to the output current (i_g) signal. The magnitude of the added WGN is calculated using (3.9) and (3.10) to achieve a signal-to-noise (SNR) ratio of 25 dB. Where P is the power and A is the amplitude of the signal or the added noise.

$$SNR = \frac{P_{signal}}{P_{noise}} = \left(\frac{A_{signal}}{A_{noise}}\right)^2 \tag{3.9}$$

$$SNR_{dB} = 10\log_{10}\left(\frac{P_{signal}}{P_{noise}}\right) \tag{3.10}$$

The aforementioned noise is added at t=300ms to the HPUC converter when it is controlled by the DRL agent as shown in Figure 3.10. As depicted, the DRL method exhibited a distortion in the output voltage waveform, albeit the distortion was not significant. This verifies the robust performance and superior capability of the DRL method in resisting performance degradation in noisy environments.

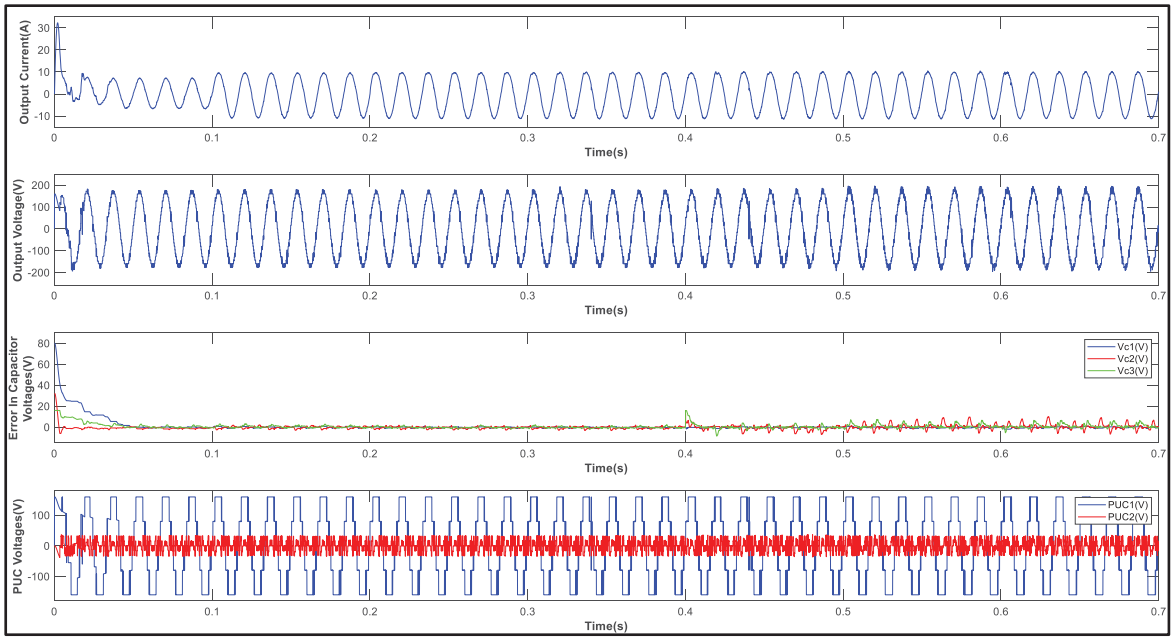

Figure 3.10 Waveforms of the output current(A), output voltage(V), errors of the DC-link capacitors(V) and voltages across submodules PUC1 and PUC2 controlled by the DRL agent under accumulated effect of active power changes at t=100ms, reactive power changes at t=200ms, noise in measurements at t=300ms, capacitor degradation at t=400ms, grid voltage changes at t=500ms and line impedance changes at t=600ms

Additionally, to evaluate the performance of the DRL method when facing parameter variations, a degradation manifested as a reduction of 15% in capacity of all three DC link capacitors of the HPUC converter is administrated at t=400ms. As shown in Figure 3.10, the DRL method experiences a slight increase in voltage ripples of the DC link capacitors. These ripples become less pronounced in a short period. The output voltage becomes slightly distorted as well.

At t=500ms, a 5% increase in the grid voltage is applied to the HPUC. As seen in Figure 3.10, this change results in slightly higher voltage ripples in the capacitors.

On top of all the previously mentioned changes, an increase of 28% is applied to the line impedance at t=600ms. The cumulative effect of all these changes imposes severe challenges

to traditional control methods. However, the DRL method only exhibits a moderate distortion when all the previous conditions impact its performance at t=600ms.

Considering the results obtained in this section, it can be concluded that the DRL agent, despite requiring prior training with relatively long training times, not only can obtain the optimal switching and control policy of the HPUC converter in a model-free approach but also demonstrates relatively high resilience towards uncertainties, parameter variations, noise, and dynamic responses.

3.6 Conclusion

Based on the results obtained in this chapter, it is shown that the implementation of Deep Reinforcement Learning (DRL) for controlling the 23-level Single DC source Hybrid Packed U-Cell (HPUC) converter offers significant advantages over traditional methods that face challenges such as requiring an accurate model of the converter as well as sensitivity to parameter variations, noise, and other uncertainties, leading to performance degradation or even instability. In contrast, DRL as a subset of machine learning, provides a modern approach to nonlinear control of power electronics. Without a prior model of the converter and only by training its neural network layers and acquiring optimal control policy through interactions with the converter, DRL demonstrates higher resilience towards uncertainties, parameter variations, noise, and dynamic changes. The model-free nature of DRL, powered by its interactive training procedure and the implementation of deep neural networks, improves its adaptability and effectiveness in addressing the challenges in the control of complex power electronics converters such as HPUC. Simulations carried out in MATLAB/Simulink confirmed the satisfactory performance of the proposed method, particularly in the presence of noise, parameter variations, and uncertainties. Therefore, a promising approach is introduced that empowers further studies on advanced and intelligent control of complex power electronics converters. It should be noted that the validation in this chapter is based on high-fidelity simulations. Due to the complexity of the 23-level HPUC topology and the risks associated with hardware implementation at this stage, experimental validation was not

performed. The subsequent chapter will further validate these findings using Software-in-the-Loop (SiL), which serve as a critical step towards real-time simulation (RTS) and eventually hardware implementation.

CHAPTER 4

COMPARATIVE ANALYSIS OF DEEP REINFORCEMENT LEARNING AND MODEL PREDICTIVE CONTROL FOR ENHANCED RESILIENCE TOWARDS MISMATCH AND NOISE IN HYBRID PACKED U-CELL CONVERTERS: A HYPERSIM-BASED STUDY

Pouria Qashqai [a], Mohammad Babaie [a], Rawad Zgheib [b], and Kamal Al-Haddad [a],

[a] Department of Electrical Engineering, École de Technologie Supérieure,
1100 Notre-Dame West, Montreal, Quebec, Canada H3C 1K3
[b] Institut de recherche d'Hydro-Québec (IREQ), Varennes, QC J3X 1S1, Canada

Paper submitted for publication in *IEEE Access*, March 2025

Abstract

This chapter presents a comparative analysis of Deep Reinforcement Learning (DRL) against Model Predictive Control (MPC) for the control of a grid-tied 23-level Single DC source Hybrid Packed U-Cell (HPUC) converter, emphasizing DRL's implementation in Hypersim for real-time simulation purposes. Advanced non-linear control methods, like MPC, while mitigating the limitations of linear control methods in power electronics, struggle with requiring accurate models, as well as performance degradation and instability under parameter mismatch and noise. DRL, as an emerging intelligent control technique, however, offers a robust, model-free approach, taking advantage of neural networks to learn optimal control strategies through interaction with the converter. The focus of this chapter is on demonstrating DRL's superior performance and resilient control under uncertainties, such as mismatches and noise. This is firstly evaluated via simulations in MATLAB/Simulink environment. Furthermore, the feasibility of DRL as a robust control method for real-time simulation purposes is confirmed by the results obtained from the Software-in-the-Loop (SiL) implementation of both methods in Hypersim.

4.1 Introduction

Power electronic converters are vital components of various modern applications, such as renewable energies, smart grids, and electric vehicles (P. Qashqai et al., 2015b; G. Zhang et al., 2018).In recent years, this rapidly evolving demand has been addressed by the integration of advanced multilevel converter topologies, alongside more efficient and reliable control methods (Xu et al., 2020). Among these advanced topologies, the 23-level Single DC Source Hybrid Packed U-Cell (H-PUC) converter stands out for its reduced component count and capability to generate a high number of voltage levels from a single DC source (Sorto-Ventura et al., 2020). This topology, which combines two five-level PUC converters (Vahedi, Labbé, et al., 2015) as high-voltage low-frequency and low-voltage high-frequency sub-modules, provides improved efficiency and reduced power losses while using a low number of switches and capacitors.

Despite these advantages, the control of the grid-tied H-PUC converter is challenging, mainly due to its complex switching states, the dynamic behavior of the grid, and the dependency of capacitor voltages on each other. Conventional linear control methods, such as PID controllers, are not suitable for modern power electronics, due to their simplicity which makes them efficient only in linear systems with predictable behavior (Xu et al., 2020). Therefore, various advanced control strategies are proposed in the literature to control non-linear and complex power electronics converters (Mohammad Babaie et al., 2021; Mahmud et al., 2019; Xu et al., 2020).

Model Predictive Control (MPC) is a famous example of such advanced non-linear control methods. MPC overcomes the limitations of linear control methods by predicting the future behavior of a system and choosing optimal control actions. Despite its advantages, MPC requires an accurate model of the converter and is known to be sensitive to parameter variations, noise in measurements, and system uncertainties, which can lead to performance degradation and instability under certain conditions. In addition, it suffers from tedious tuning

of the weighting factors (Pouria Qashqai et al., 2023; Rodriguez, Garcia, Mora, Davari, et al., 2021; Rodriguez, Garcia, Mora, Flores-Bahamonde, et al., 2021).

In response to these limitations, this study proposes the implementation of Deep Reinforcement Learning (DRL) as a modern, model-free approach to the non-linear control of the H-PUC converter. DRL, a subset of machine learning, utilizes deep neural networks to learn optimal control strategies via direct interaction with the environment (i.e. the converter), without the need for an accurate model of the system (Cao et al., 2020b). This approach not only does not require an accurate model of the converter but also demonstrates resilience to uncertainties, parameter variations, and noise, as a result of the generalized and adaptive nature of neural networks. While Chapter 3 introduced the DRL control strategy for the HPUC, this chapter significantly expands on that work by conducting a rigorous comparative analysis against a conventional MPC controller and, crucially, validating the approach through Software-in-the-Loop (SiL) simulations. The primary contribution here is to show not just the feasibility of DRL control for the HPUC, but its quantifiable superiority in terms of resilience and robustness under a wide range of non-ideal conditions, an aspect not explored in the previous chapter.

This chapter presents a comparative analysis of two advanced nonlinear control methods: DRL and the traditional MPC. It highlights the superior performance and resilience of DRL in the face of parameter mismatches and noise. Section 4.2 introduces the H-PUC converter and explains how DRL can address its control challenges. Then, Section 4.3 evaluates the advantages of a model-free, robust control approach in improving the reliability and efficiency of power electronics converters, with a particular focus on the H-PUC converter. This evaluation is supported by simulations in MATLAB Simulink and SiL implementation in Hypersim in Section 4.4. The chapter concludes in Section 4.5 by discussing how this comparative analysis contributes to ongoing research on advanced control strategies in power electronics, particularly the adoption of intelligent control methods, such as DRL in complex systems.

4.2 HPUC Control Using Deep Reinforcement Learning

This section introduces the HPUC converter, its control challenges are discussed, and then the DRL control method is developed for the grid-connected configuration of HPUC known as a complex and nonlinear power electronic system.

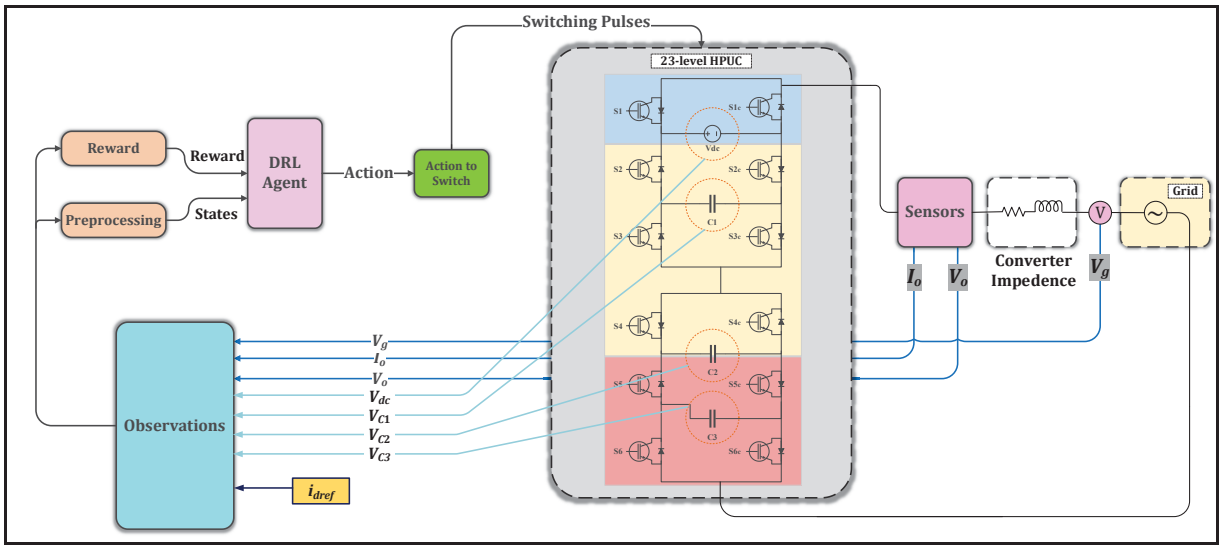

Figure 4.1 Block diagram of a 23-level single DC source HPUC, controlled by the proposed DRL method

4.2.1 Topology and Control Challenges of HPUC

The 23-level HPUC shown in Figure 4.1 is a cutting-edge topology in power electronics and is famous for its improved efficacy and reduced power loss as well as its ability to generate multiple voltage levels from a single DC source while using a reduced number of switches. Despite its advantages, controlling the grid-tied HPUC is quite challenging due to its complex switching behavior, the dynamic behavior of the grid, and the interdependence of capacitor voltages, which result in a complex optimization problem that linear control methods fail to solve (Sorto-Ventura et al., 2020).

Advanced non-linear control methods like Model MPC(Rodriguez, Garcia, Mora, Davari, et al., 2021; Rodriguez, Garcia, Mora, Flores-Bahamonde, et al., 2021) have been developed to

mitigate these challenges by predicting the future behavior of the converter and optimizing control accordingly. However, MPC-based methods often require an accurate mathematical model of the converter and are sensitive to parameter mismatch, noise, and uncertainties, leading to potential performance degradation in practical operating conditions, especially in a grid environment.

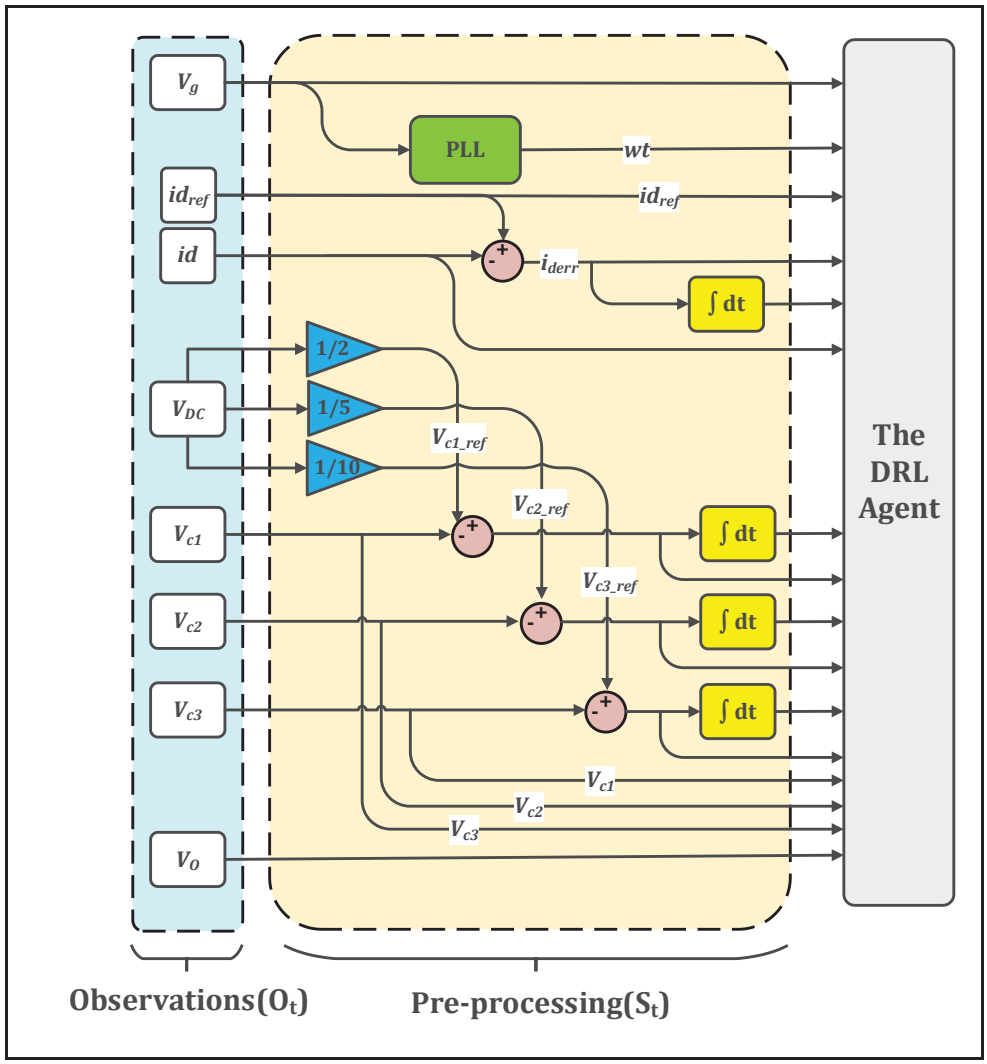

Figure 4.2 Block Diagram of the state space generation unit comprised of observations (measurements) and pre-processing

4.2.2 Deep Reinforcement Learning Approach

In this study, we adopt a model-free deep reinforcement learning (DRL) approach based on the deep Q-network (DQN) algorithm to control a 23-level hybrid packed U-cell (HPUC) converter. The converter's dynamics are simulated in a Simulink environment, where the observation space is defined as an 18-dimensional vector that includes integrated error, instantaneous error, and measured height. The discrete action space consists of 64 possible control actions that adjust the converter's flow. Deep Reinforcement Learning is proposed as an innovative solution for the control challenges of power electronics converters by taking a model-free approach (Pouria Qashqai et al., 2023). Unlike conventional methods, the proposed DRL method does not require an exact model of the system. Instead, as depicted in Figure 4.1, it utilizes a Deep Q-Network (DQN) due to the continuous observation and discrete action spaces. The DQN agent learns optimal control strategies through direct interaction with the converter.

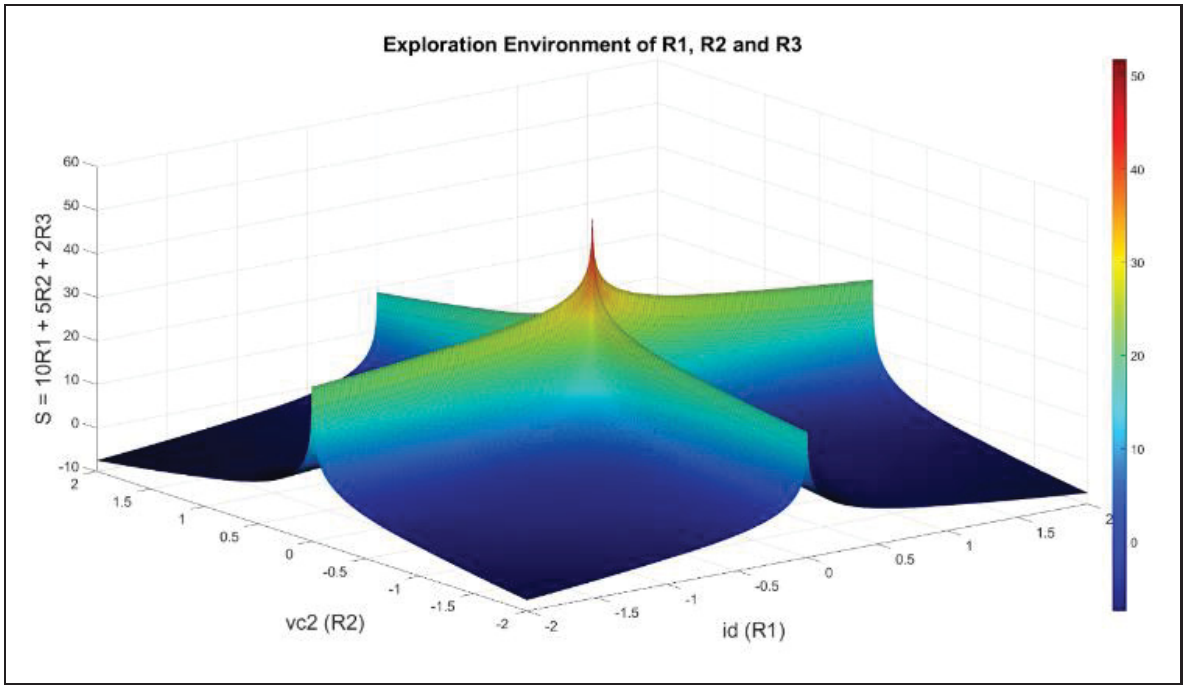

Figure 4.3 Exploration environment of a combination of the reward functions R_1, R_2, and R_3 representing $\Delta(i_d)$, $\Delta(v_{c2})$, and $\Delta(v_{c3})$

The Q-value function $Q(s, a)$ is approximated by a deep neural network with a bifurcated architecture: one branch processes the state input (via an input layer followed by multiple fully connected layers with ReLU activations) and another branch processes the action input. The outputs from these two branches are combined through an "addition layer", followed by further ReLU activation and a final fully connected layer that generates the Q-value:

$$\mathrm{Q(s, a; \theta)} \approx \mathrm{f(s, a; \theta)} \tag{4.1}$$

where θ represents the network parameters.

The network is trained by minimizing the loss function:

$$\mathrm{L(\theta)} = \mathrm{E}_{\{s,a,r,s'\}}[\left(\mathrm{r} + \Upsilon \underset{\{a'\}}{\mathrm{max}} \mathrm{Q(s', a';\ \theta^{-})} - \mathrm{Q(s, a;\ \theta)} \right)^{2}] \tag{4.2}$$

where $\Upsilon = 0.01$ is the discount factor and θ^{-} represents the parameters of the target network.

The corresponding Q-learning update rule is:

$$\mathrm{Q(s_t, a_t)} \leftarrow \mathrm{Q(s_t, a_t)} + \alpha(\mathrm{r_t} + \gamma \underset{\{a'\}}{\mathrm{max}} \mathrm{Q}\left(\mathrm{s_{\{t+1\}}}, \mathrm{a'}\right) - \mathrm{Q(s_t\ , a_t)}\ ) \tag{4.3}$$

with α as the learning rate. An epsilon-greedy exploration strategy, with an initial epsilon of 1 that decays by a factor of 0.05, is employed. The agent is trained in over 10,000 episodes, where each episode lasts $\left\lceil \frac{T_f}{T_{sa}} \right\rceil$ steps, with a simulation sample time $T_s = 50\mu s$, an agent sample time $T_{sa} = \frac{1}{10000}\, s$, and a total simulation duration $T_f = \frac{500}{60}$ seconds per iteration.

4.2.3 Observations and Pre-processing

In order to capture the essential dynamic behavior of the HPUC converter, we measure several signals as part of the DRL agent's observation array as shown in (4.4):

$$O_t = [i_d, v_g, v_o, V_{DC}, i_{dref}, v_{c1}, v_{c2}, v_{c3}] \tag{4.4}$$

Where O_t is the observations array, i_d is the output current, v_g is the grid voltage, v_o is the output voltage of the converter, V_{DC} is the input DC voltage, i_{dref} is the reference current, and finally v_{c1}, v_{c2}, and v_{c3} are the voltages across capacitors c_1, c_2 and c_3, respectively. Figure 4.2 shows that the pre-processing unit transforms measurements from the observation unit into the controller's state space S_t, as given in (4.5):

$$\begin{aligned} S_t = [i_d, v_g, v_o, v_{c1}, v_{c2}, v_{c3}, i_{derr}, \int i_{derr}\,, i_{dref}, \\ \omega t, v_{c1err}, v_{c2err}, v_{c3err}, \int v_{c1err}\,, \int v_{c2err}\,, \int v_{c3err}] \end{aligned} \tag{4.5}$$

Here i_{derr} is the instantaneous error between the reference current i_d and actual output current i_g and i_{derr} is its accumulative error. Similarly, v_{c1err}, v_{c2err}, and v_{c3err} represent the differences between the actual and reference voltages of capacitors c_1, c_2, and c_3 respectively with their integrals providing the accumulated errors. Lastly, ωt represents the angular speed of the grid voltage, calculated as $2\pi f$ where f is the line frequency of v_g.

4.2.4 Reward Function and Exploration Environment

The reward function is as (3) where $\Delta(i)$ is the error corresponding to the output current and voltages of capacitors:

$$R_i = \begin{cases} 1 - \log_{10}|10\Delta(i)| & if\ 1^{-4} < |\Delta(i)| < 1 \\ 1 - \sqrt{|\Delta(i)|} & if\ |\Delta(i)| \geq 1 \\ 1 - \log_{10} 1^{-3} & otherwise \end{cases} \quad (4.6)$$

The reward function was designed to guide the DRL agent toward maintaining minimal errors in both the output current and the capacitor voltages. Specifically, the function is defined using a logarithmic penalty approach to provide a steep corrective signal for larger deviations while still offering fine-grained rewards when the system operates close to its desired state. This structure ensures that even small errors are penalized in a way that promotes precise control, whereas larger errors raise a more severe penalty to quickly discourage undesirable behavior. Such a tailored reward strategy is critical in managing the complex dynamics of the 23-level HPUC converter.

The model-free nature of DRL is particularly advantageous for managing the complexity of HPUC. It allows for a flexible control strategy that can adapt to changes in the system's dynamics without the need for constant re-tuning or model updates. Moreover, DRL's inherent resilience to uncertainties and its capability to learn from experience make it well-suited to address the issues of parameter mismatch and noise, which are prevalent challenges in the control of advanced power electronics systems.

To visualize how the DRL agent explores different states, Figure 4.3 depicts the exploration environment of a combination of the reward functions R_1, R_2, and R_3 representing $\Delta(i_d)$, $\Delta(v_{c2})$, and $\Delta(v_{c3})$. Using this approach, the DRL agent thus “learns” to remain in or return quickly to states corresponding to low errors.

4.2.5 Action Space

The HPUC converter has 12 switches embedded into its two submodules (each with 6 switches). Each submodule can achieve $2^3 = 8$ unique switching combinations representing different voltage levels. Therefore, the total number of permitted switching states is 64 which defines the action space A_t as shown below:

$$A_t = \begin{bmatrix} 0 & 1 & \dots & 6 & 7 \\ 0 & 1 & \dots & 6 & 7 \end{bmatrix} \tag{4.7}$$

For the DRL agent, each switching state is represented by an integer, and a lookup table maps these integers to the corresponding switching states (as depicted in Figure 4.1)

4.2.6 Parameter Selection and Its Impact on Deep Reinforcement Learning (DRL) Performance

The performance of DRL models heavily depends on the careful tuning of several hyperparameters. Key parameters include the learning rate, discount factor, batch size, neural network architecture, exploration strategy, and replay buffer size. Each parameter influences how efficiently and effectively a DRL model learns from its environment:

4.2.6.1 Learning Rate (α)

The learning rate determines the magnitude of updates to the model's parameters when new data is encountered. An excessively high learning rate can cause the model to overshoot optimal solutions, potentially leading to premature convergence on suboptimal policies. Conversely, a very low learning rate can slow down the training process, requiring more iterations and computational resources.

4.2.6.2 Discount Factor (γ)

The discount factor weights the importance of future rewards relative to immediate rewards. A higher discount factor encourages the agent to value long-term benefits, which is important in applications where long-term planning is crucial. A lower discount factor, on the other hand, makes the agent focus more on immediate rewards, which can be advantageous in environments that change rapidly.

4.2.6.3 Batch Size

Batch size affects the stability and efficiency of the learning process. Using larger batches generally results in more stable gradient estimates and smoother convergence, though it demands greater memory and processing power. Smaller batches allow for more frequent updates but can introduce higher variance (noise) in the gradient estimates. The batch size in the simulation results of this chapter is obtained by an initial guess and improved via trial and error.

4.2.6.4 Neural Network Architecture

The depth (number of layers) and width (number of neurons per layer) of the network determine its capacity to capture complex patterns in the data. Deeper and wider networks can model more intricate functions but may require additional techniques such as regularization to prevent overfitting.

4.2.6.5 Exploration vs. Exploitation Balance:

Balancing exploration (trying new actions) and exploitation (using known rewarding actions) is critical for effective policy learning. Strategies like epsilon-greedy dynamically adjust the probability of random actions to ensure sufficient exploration while gradually favoring exploitation as the model learns more about the environment.

4.2.6.6 Replay Buffer Size:

In value-based methods like Deep Q-Networks (DQN) used in this chapter, a replay buffer stores past experiences. A larger buffer allows the agent to learn from a more diverse set of experiences, which can help stabilize training by breaking correlations between sequential data. However, a very large buffer might introduce stale experiences that are less relevant to the current policy.

The choice of these parameters can significantly influence system performance. Poorly tuned parameters may lead to unstable training, suboptimal policy learning, or inefficient resource utilization. Consequently, parameter tuning in DRL is often performed through empirical testing—using techniques such as grid search, random searches, or adaptive methods—along with validation against performance metrics tailored to the specific application.

In our study, we selected parameter values based on both established guidelines from literature and extensive empirical testing. We then validated these choices through simulations that tested the model's robustness under various operational scenarios, including changes in parameter settings and external disturbances. This comprehensive approach ensured that our DRL models not only performed well under controlled conditions but also maintained resilience and adaptability in real-world applications.

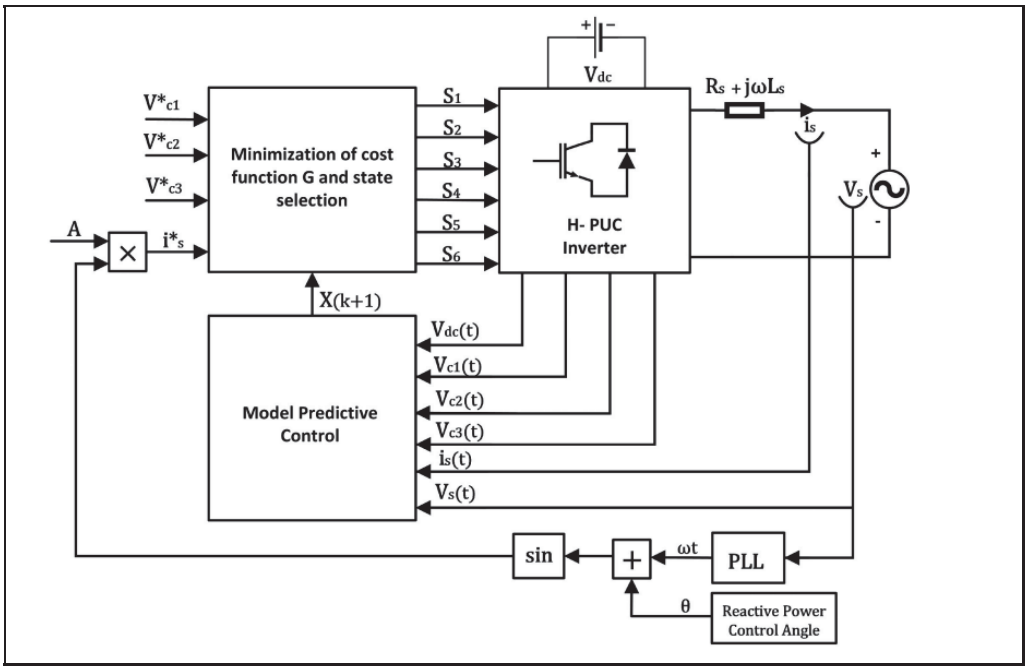

Figure 4.4 Block diagram of model predictive control (MPC) utilized to control a grid-connected HPUC
Adapted from Sorto-Ventura et al. (2020, p. 5)

4.2.7 MPC Cost Function

The MPC method implemented in (Sorto-Ventura et al., 2020) is used as a point of reference to be compared with the DRL method. The MPC method implemented in this study is a finite control set model predictive control (FCS-MPC) strategy that utilizes a simplified, discretized model of the 23-level HPUC converter based on the Forward Euler approximation. In this approach, the inverter output voltage is expressed as a linear combination of the switching variables and capacitor voltages, with the converter's behavior predicted for all 64 possible switching states at each time step. The controller computes a cost function that penalizes deviations of the predicted capacitor voltages and grid current from their respective reference values, and it selects the switching state that minimizes this cost, thereby generating the appropriate gate signals for the inverter. This method enables robust control of the HPUC converter by effectively handling its nonlinearities and ensuring low total harmonic distortion without the need for additional modulation techniques. The block diagram of this control method is depicted in Figure 4.4. For a fair comparison, the MPC controller's parameters were also carefully tuned. The cost function C_f, for the MPC is given by:

$$C_f = \lambda_1 g_{id} + \lambda_2 g_{vc1} + \lambda_3 \mathrm{g_{vc2}} + \lambda_4 \mathrm{g_{vc3}} \tag{4.8}$$

Where g_{id}, g_{vc1}, g_{vc2} and g_{vc3} respectively represent tracking errors for the grid current and capacitor voltages. The weighting factors λ_i are critical to the controller's performance and were tuned empirically through an iterative process. Starting from an equal weight distribution, the factors were adjusted to prioritize current tracking while ensuring capacitor voltage stability, mirroring the objectives set in the DRL reward function. The final tuned values aimed to achieve the fastest stable response with minimal overshoot. It is crucial to note that, unlike the DRL agent's offline training, MPC tuning is a manual process and its performance is highly sensitive to these values, representing a significant challenge in its practical application. (Rodriguez, Garcia, Mora, Davari, et al., 2021).

4.3 Comparative Analysis: DRL vs. MPC

This section presents the comparative analysis between Deep Reinforcement Learning (DRL) and Model Predictive Control (MPC) in the control of the hybrid packed U-cell converter. Using simulation and Software-in-the-Loop (SiL) results based on parameters shown in Table 4.1 and Table 4.2, this analysis demonstrates the superior adaptability and resilience of DRL in contrast to MPC method across various operational conditions, including steady-state performance, dynamic response, and robustness against noise and parameter mismatch.

Table 4.1 Simulation Parameters

Parameter	Value	Unit
Peak Grid Voltage	170	V
Grid Frequency	60	Hz
Grid Resistance (per phase)	0.1	Ω
Grid Inductance (per phase)	5	mH
DC-link Capacitors	2000	μF
Simulation Time Step (Ts)	50	μs
DC-link Voltage	160	V

Table 4.2 Training Parameters

Parameter	Value
Layer Size of the State Path	[224,224]
Layer Size of the Action Path	124
Learning Rate	0.01
Gradient Threshold	1
Normalization	None
Bias Learn Rate Factor	0
Double DQN	No

Target Smooth Factor	0.001
Experience Buffer Length	10^5
Mini Batch Size	640
Discount Factor	0.9
Score Averaging Window Length	5
Epsilon Greedy Exploration Epsilon	1
Epsilon Decay / Epsilon Min	0.05 / 0.01
Agent Sampling Time	100µs

4.3.1 Steady-State Performance

As illustrated in Figure 4.5 and Figure 4.6, both DRL and MPC effectively control the HPUC converter in steady-state conditions. However, compared to MPC, DRL shows a slightly faster response to setpoint and a shorter settling time despite a slight overshoot, which is due to instant control instead of doing a switching optimization like MPC. The total harmonic distortion (THD) for the output current is 2.71% and 2.14% for the DRL and MPC, respectively. Thus, both methods remained within acceptable limits of power quality standards.

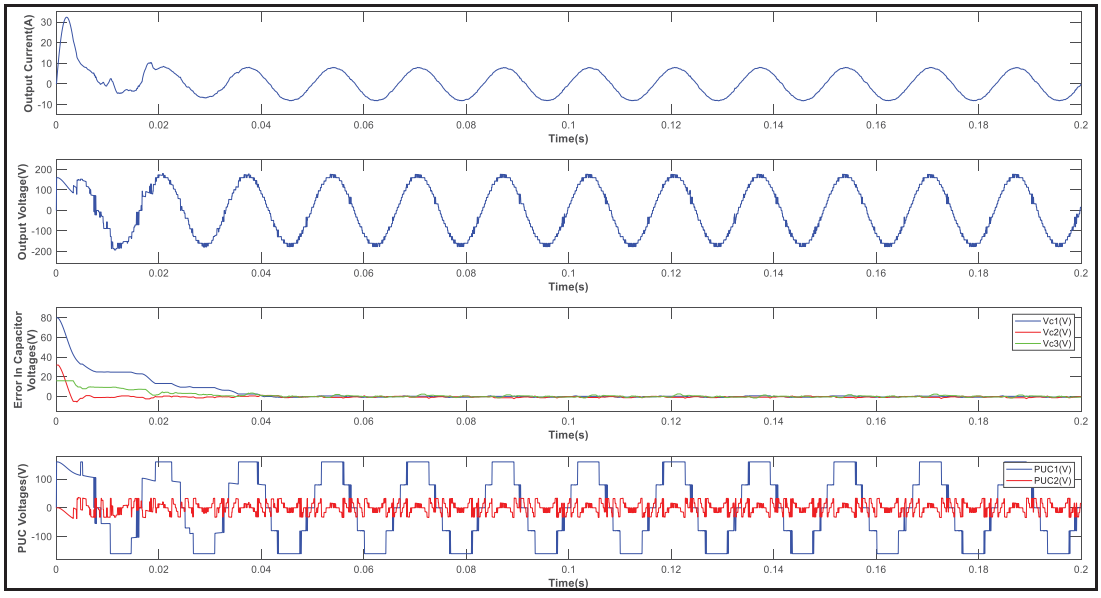

Figure 4.5 Waveforms of the output current (A), output voltage (V), errors of the DC-link capacitors (V) and voltages across submodules PUC1 and PUC2 in steady state operation controlled by the DRL agent

4.3.2 Dynamic Response

As presented in Figure 4.8, the active and reactive power is changed at $t = 100ms \; and \; t = 200ms$ for both methods. The results depict DRL's superior flexibility, efficiently managing abrupt load changes with minimal impact on system stability and performance. Conversely, MPC displayed a notable delay in adapting to these sudden changes, a limitation attributed to its reliance on predictive modeling.

4.3.3 Resilience to Noise and Parameter Variations

Moreover, the HPUC converter's response to various disturbances under control by both DRL and MPC was obtained as seen in Figure 4.8. At $t = 300ms$, a 25dB noise in measurements introduced visible distortion in the output voltage waveforms, with the DRL method showing significantly lower distortion than the MPC, indicating its superior noise resistance.

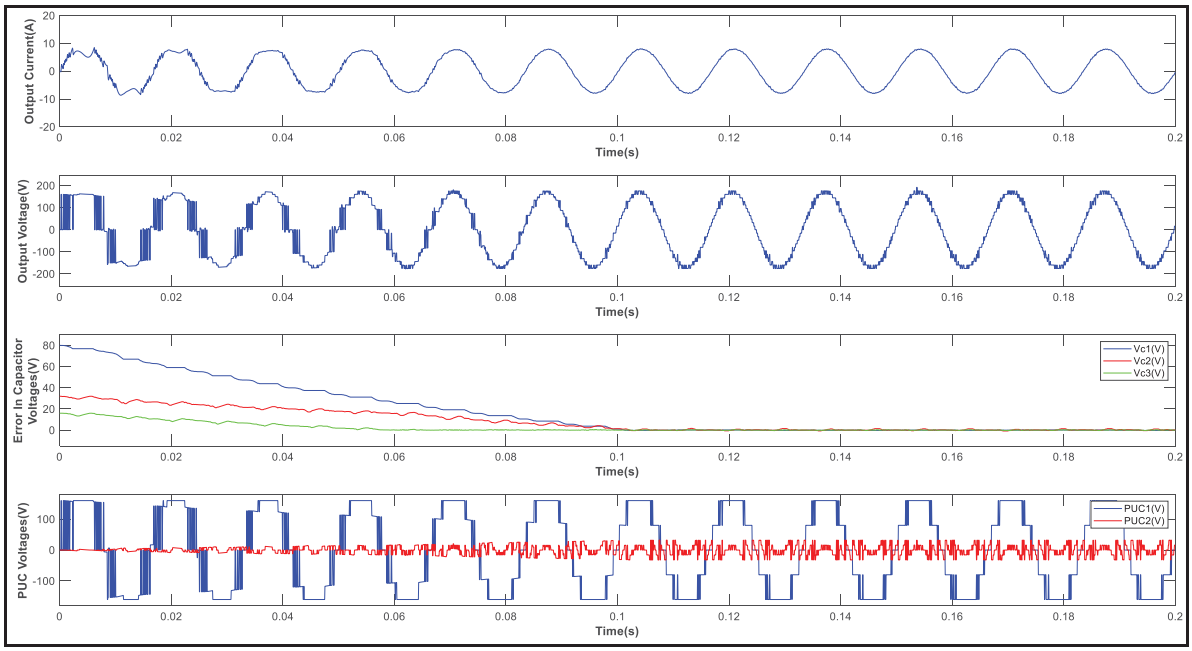

Figure 4.6 Waveforms of the output current(A), output voltage(V), errors of the DC-link capacitors(V) and voltages across submodules PUC1 and PUC2 in steady state operation controlled by the MPC controller

Moreover, by reducing capacitance of C1 to C3 by 15% at $t = 400ms$, the DRL method showed slight voltage ripple which settled rapidly, but the MPC method displayed noticeable capacitor voltage deviations and output voltage distortion. Under a 5% grid voltage increase at $t = 500ms$ and a 28% converter output impedance increase at $t = 600ms$, MPC showed heavy voltage distortion and instability, while the DRL method had moderate distortion and maintained stability. These results highlight the DRL method's resilience to uncertainties, parameter variations, and dynamic changes, outperforming traditional methods like MPC in handling complex disturbances.

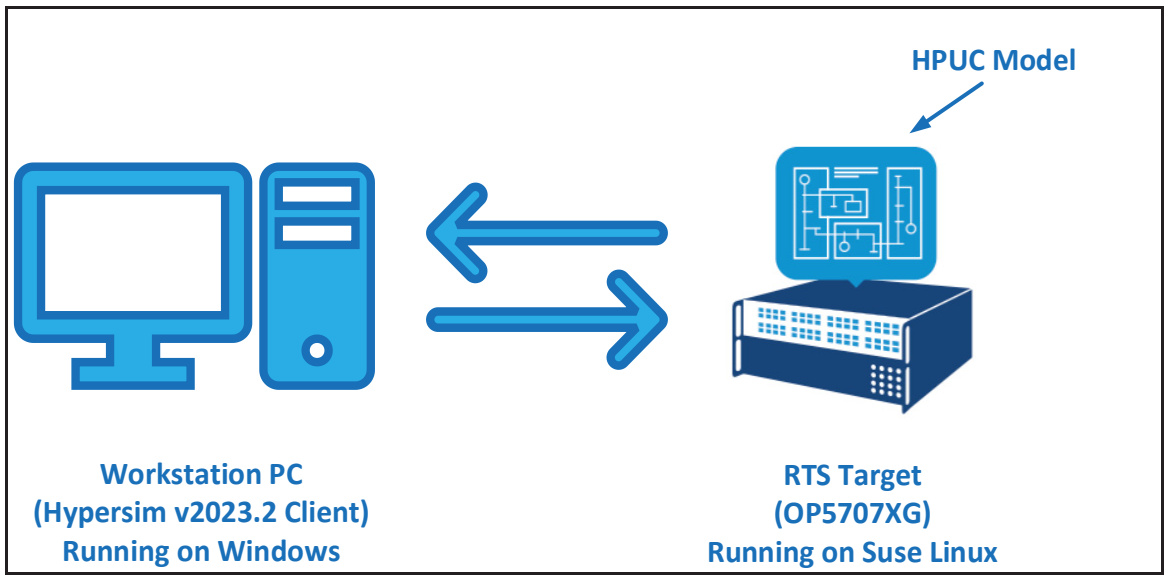

Figure 4.7 Diagram of the real-time simulation based on software-in-the-loop (SIL) using OP5707XG target

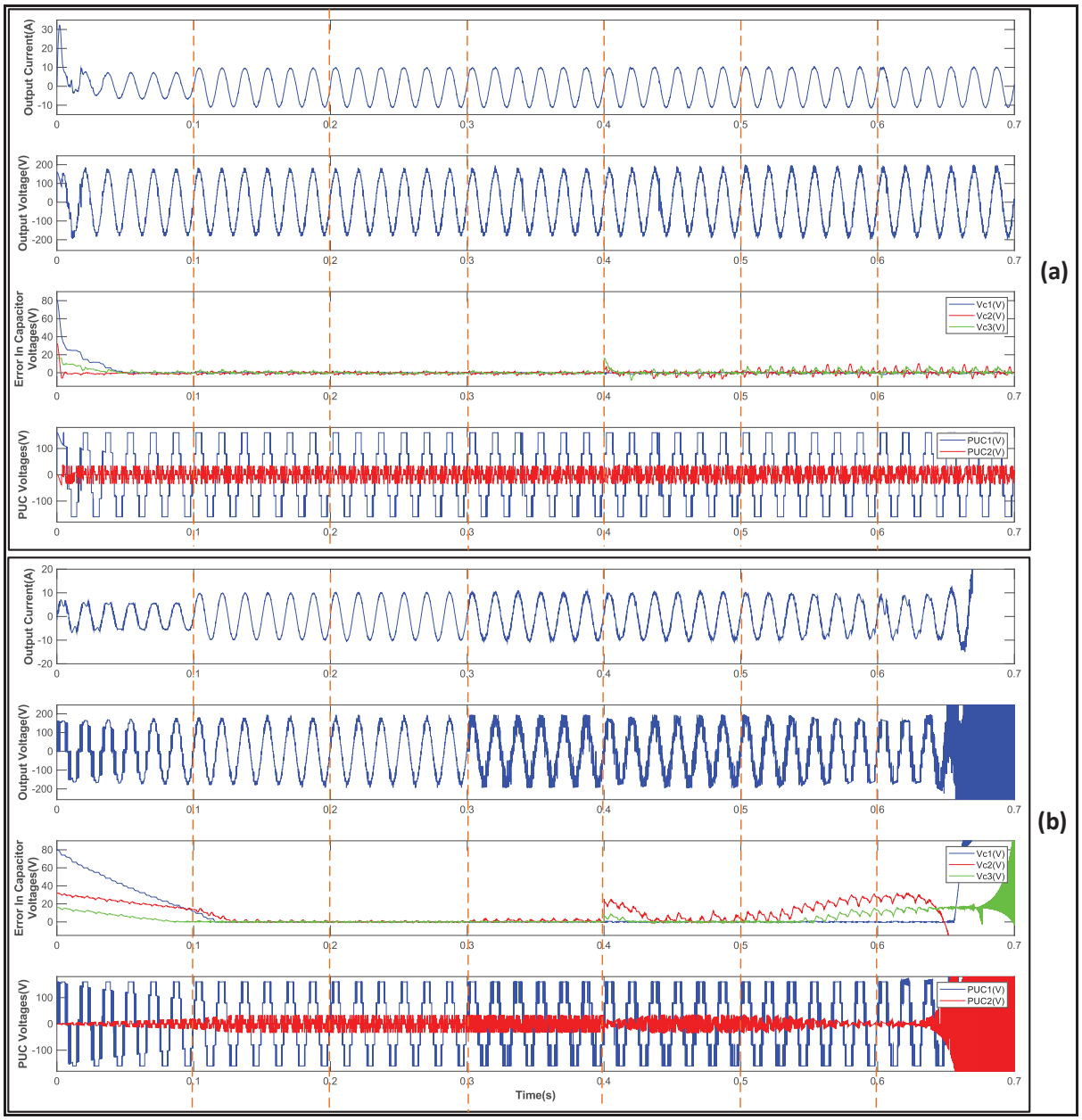

Figure 4.8 Waveforms of the output current(A), output voltage(V), errors of the DC-link capacitors(V) and voltages across submodules PUC1 and PUC2 controlled by the DRL agent (a) and MPC controller (b), under accumulate effect of active power changes at t=100ms, reactive power changes at t=200ms, noise in measurements at t=300ms, capacitor degradation at t=400ms, grid voltage changes at t=500ms and converter impedance changes at t=600ms

4.4 Real-Time (Software-in-the-Loop) Simulation Results

In this chapter, Software-in-the-Loop (SIL) which is a subset of Real-time Simulation (RTS) is utilized in Hypersim, in place of experimental results to assess the converter under extreme conditions. SIL provides academic researchers and engineers with a virtual precise model of components for fast development and testing of complex systems such as HPUC.

This approach offers the advantage of continuously testing and developing control algorithms using a target, eliminating the need for costly hardware prototypes or test benches. SiL also enables early detection of flaws or bugs, reducing troubleshooting costs in later stages of development.

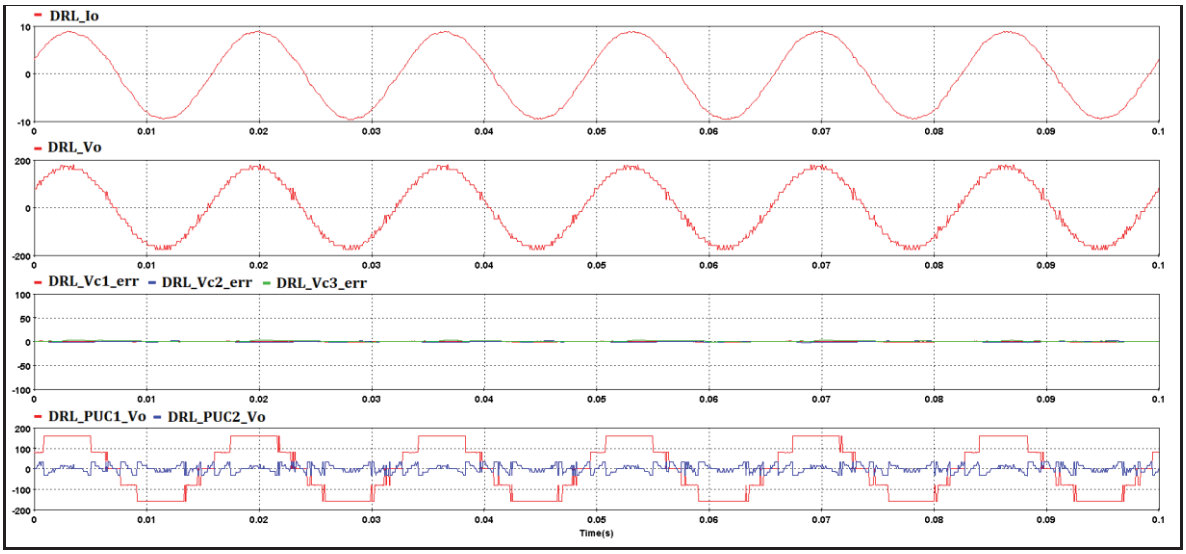

Figure 4.9 Waveforms of the output current(A), output voltage(V), errors of the DC-link capacitors(V) and voltages across submodules PUC1 and PUC2 controlled by the DRL in Hypersim

Furthermore, SiL can be combined with the traditional Hardware-in-the-Loop (HiL) approach, accelerating the development cycle. This method aligns with the aim of this study to comprehensively evaluate the performance of both DRL and MPC control methods under extreme conditions while ensuring the safety of the researchers and pieces of equipment as well as optimizing development time and resources.

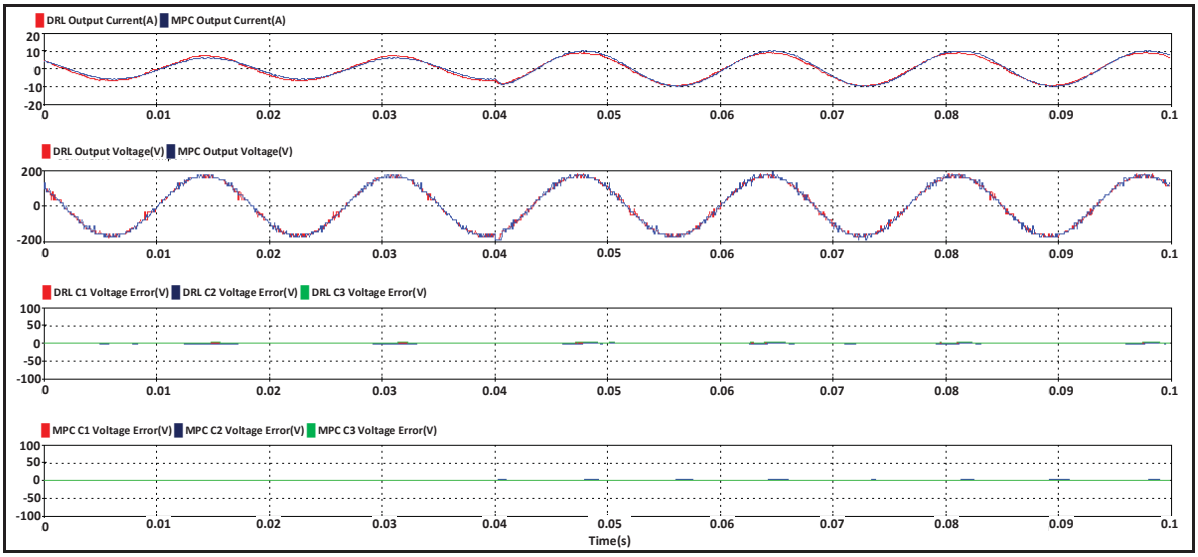

Figure 4.10 Waveforms of the output current(A), output voltage(V), errors of the DC-link capacitors(V) controlled by the DRL agent and MPC controller in Hypersim facing a step change from $i_d = 8A$ to $i_d = 10A$ at t=40ms

The tests in this section are carried out using the Hypersim platform, a tool developed by OPAL-RT, specializing in real-time simulation of power systems. Since the DRL agent is not supported by Hypersim, an equivalent of the DRL agent was written in C programming language and exported as a UCM block. The parameters of the equivalent DRL agent were obtained using the trained agent object in MATLAB.

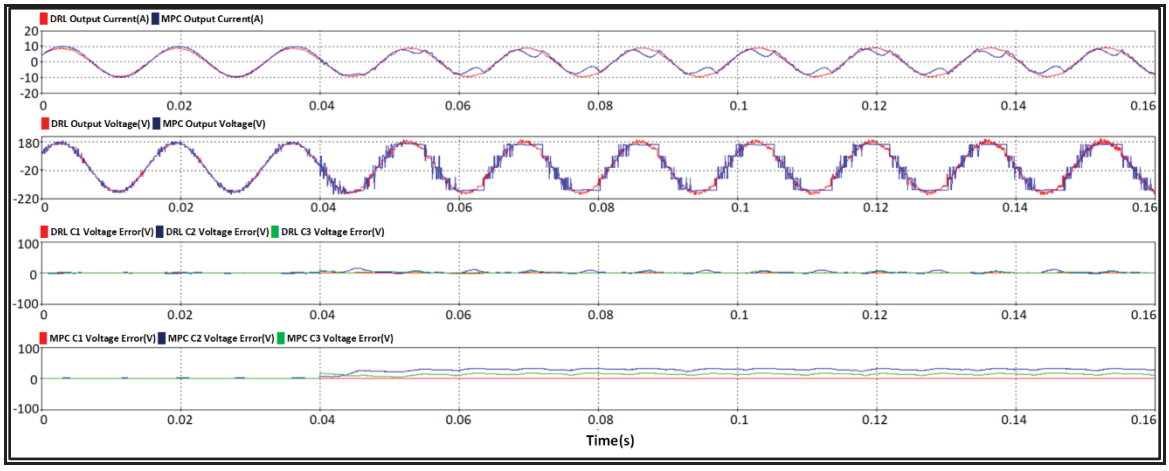

Figure 4.11 Waveforms of the output current(A), output voltage(V), errors of the DC-link capacitors(V) controlled by the DRL agent and MPC controller in Hypersim facing noise in measurements, capacitor degradation, and grid voltage increase at t=40ms

Figure 4.7 shows a simplified diagram of the overall RTS approach, highlighting that the DRL agent and the converter model run in real time at the desired sampling frequency. This ensures the same timing constraints found in an actual hardware setup but without risking physical equipment. The real time simulation is performed by OP5707XG, one of the flagship real-time simulation targets developed by OPAL-RT. This target is powered by a 16-core Intel® Xeon® CPU clocked at 3.5GHz per core and a Xilinx® Virtex®7 485T FPGA. It has native compatibility with Hypersim and RT-LAB and it supports various tools such as ARTEMiS, eHS, ePHASORSIM, Orchestra, and RT-XSG. In-depth specifications of OP5707XG are accessible in ("Flagship Real-Time Digital Simulator │ Simulation Tools │ OP5707XG," n.d.).

The steady-state operation of the HPUC converter controlled by the DRL method is depicted in Figure 4.9. As seen, the SiL results closely match the simulation results and prove the satisfactory steady-state performance of the DRL method.

Additionally, the dynamic responses of the HPUC converter under the control of the DRL agent and the MPC controller are obtained in Hypersim. As depicted in Figure 4.10, the step change of the output current from $i_d = 8A$ to $i_d = 10A$ is applied to both control methods at t=40ms. The waveforms of the output current and output voltage for DRL (red) and MPC (blue) verify similar dynamic responses of both methods.

Ultimately, to compare the performance of both methods under uncertainty, 25dB noise in measurements, 15% capacitor degradation, and 5% grid voltage increase are applied to the HPUC converter at t=40ms simultaneously. As shown in Figure 4.11, the DRL method exhibited a slight distortion in output current and voltage (depicted as red waveforms) and a moderate ripple increase of the DC link capacitors. On the other hand, the MPC method showed a tremendous distortion in the output current and voltage (depicted as blue waveforms) while suffering from a significant deviation and ripple of the DC link capacitors.

Equally from the SiL results it can be concluded that the DRL method shows an advantage in uncertainties over the conventional MPC method.

4.5 Conclusion

This chapter presents a comparative analysis of DRL and MPC for controlling a grid-tied HPUC converter, focusing on resilience to mismatches and noise. DRL's superior performance is shown through simulations and SiL simulation implementations, highlighting its robust, model-free approach that utilizes neural networks for optimal control. While MPC is predicated on an accurate system model and requires continuous online optimization, DRL shifts this burden to a one-time, offline training process. The outcome is a controller that excels precisely where model-based methods struggle which is under significant parameter mismatch, measurement noise, and dynamic uncertainties. This trade-off (initial training effort for superior real-world resilience) is the central advantage demonstrated in this chapter.

A key aspect of this comparison is the tuning effort. While DRL requires a significant, automated, offline training phase to learn its policy, the MPC controller requires careful manual tuning of its cost function's weighting factors. The results demonstrated that even with careful tuning, the MPC method's performance degrades significantly under parameter mismatch and noise, whereas the DRL agent, once trained, maintains robust performance without further intervention. This highlights a fundamental advantage of DRL which is trading manual, model-sensitive tuning for an automated, data-driven learning process that yields a more resilient controller.

While the DRL-based control method demonstrates superior resilience and adaptability in simulations and real-time tests, its practical implementation presents certain challenges. One of the primary concerns is the computational complexity associated with training the DRL agent. The training process requires a significant amount of data and extended time to achieve optimal performance, particularly when applied to complex topologies like the 23-level HPUC converter. The discrete action space involves managing 64 possible switching states, further

increasing the computational load. Additionally, deploying these models in real-time systems introduces hardware constraints. The Hypersim platform, utilizing a powerhouse such as OP5707XG target with a 16-core Intel® Xeon® CPU and a Xilinx® Virtex®7 FPGA, facilitates real time testing. However, practical deployment on less powerful embedded systems may face limitations in processing speed and memory capacity, especially in high frequency switching scenarios. Balancing these computational demands with system responsiveness remains a critical challenge for real-world applications.

This study could be used for future research in intelligent control, and a step towards enhancing the reliability and efficiency of power electronics converters.

CHAPTER 5

AN ADVANCED BLACK-BOX MODELING TECHNIQUE FOR POWER ELECTRONIC CONVERTERS USING MACHINE LEARNING

This chapter is largely based on work published by the author in IEEE conference proceedings. The content presented here is a compilation of four papers: (P. Qashqai et al., 2020; Pouria Qashqai, Al-Haddad, et al., 2022; Pouria Qashqai et al., 2021; Pouria Qashqai, Zgheib, et al., 2022). Minor revisions and modifications have been made to align the content with the context of this thesis and to present the research as a cohesive narrative.

5.1 Introduction

As we discussed in CHAPTER 1, Black-box modeling techniques are particularly advantageous for modeling commercial over-the-counter converters for which, there is a lack of detailed knowledge of internal components or a lack of accurate data sheets. Among various black-box modeling approaches introduced earlier, those based on Artificial Neural Networks (ANNs) can identify system dynamics only from input and output data. Thus, these methods are often called “data-driven” in the literature.

In the following sections, conventional ANNs also known as Feed-Forwards ANNs (FF-ANNs), and Conventional Recurrent Neural Networks also known as Vanilla Recurrent Neural Networks (VRNNs) are used for black-box modeling. Generally speaking, Recurrent Neural Network (RNN) based methods, outperform traditional FF-ANNs due to their ability to learn the temporal dependencies through recurrent connections. However, since VRNNs utilize simple memory loops, they lack the ability to fully extract and learn long-term temporal dependencies.

To solve this problem, a long-short term memory (LSTM) network is used in this chapter as the recurrent layer in a deep neural network (DNN) approach to extract the black-box model

of PECs. Through comparative analysis, the superiority of the proposed LSTM method is investigated.

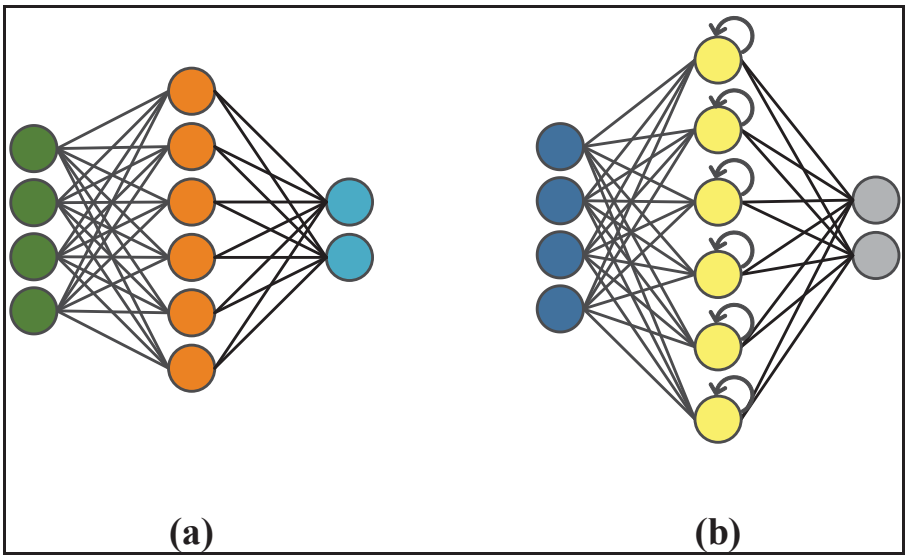

Figure 5.1 Structure of an FF-ANN (a) and a VRNN (b)

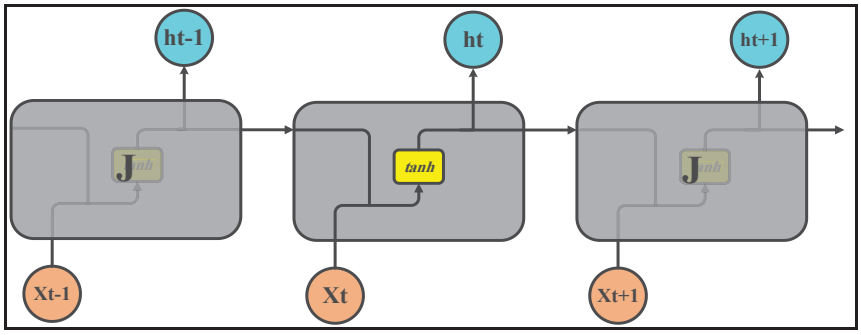

Figure 5.2 A vanilla recurrent neural network (VRNN) structure

Despite being more effective than conventional RNN and ANN networks, LSTM is relatively computationally expensive. To solve this problem, a similar but more computationally efficient architecture known as the Gated Recurrent Unit (GRU) is proposed in the literature. Through comparative analysis, the advantages and disadvantages of using GRU in place of LSTM as the recurrent layer of the proposed black-box method are investigated in the following sections.

Ultimately, an algorithm is proposed to export the trained DNN network containing the black-box model of PECs obtained through the method proposed in this chapter into Hypersim for real-time simulation purposes. The performance of the proposed algorithm is evaluated in Hypersim against Simulink.

5.1.1 Feed-Forward Artificial Neural Network (FF-ANN)

Artificial neural networks are used widely for solving various problems. From simple curve-fitting problems to higher degree time-series forecasting. As shown in Figure 5.1, a feedforward ANN is comprised of an arbitrary number of neurons that form its hidden layer(s). if multiple layers are used, the term “deep” neural networks is assigned to them while for a smaller number of layers, “shallow” neural networks are often used in the literature. Feedforward ANNs can virtually fit any outputs to their corresponding inputs. However, there are two main problems associated with them.

First, these networks may effectively fit the given data but they may fail to generalize them to the actual input-output relationships of the system. This problem is addressed as the “over-fitting” problem which is an innate problem of neural networks. To overcome this problem, various solutions are proposed in the literature (Schmidhuber, 2015).

Another problem is that modeling a dynamic system like a power electronic converter using such networks requires data from numerous operating points before it can effectively predict the dynamic response of the converter in unseen operating points. This problem is due to the fact that feedforward ANNs lack any memories. Without any memory units, these networks regard each time step independently which significantly reduces their performance for large signal modeling of systems. Thus it is suggested to use such networks for small-signal modeling of power converters.

One workaround for mitigating the problems associated with FF-ANN is using delay blocks. Another solution is to use artificial recurrent neural networks (RNNs) which inherently provide memory blocks for learning the short term dependencies between data sequences.

5.1.2 Recurrent Neural Networks (RNN)

As stated earlier, conventional feedforward ANNs perform inadequately when it comes to sequential datasets where there are dependencies between previous inputs and outputs. Recurrent neural networks (RNNs) are an extension of feedforward ANNs by memory blocks to deal with short-term and long-term dependencies. As shown in Figure 5.3-a, a neuron in an RNN is the extension of a conventional neuron with a closed loop for preserving the memory of the previous time steps. This architecture is dual to cascading t number of conventional neurons as shown in Figure 5.3-b.

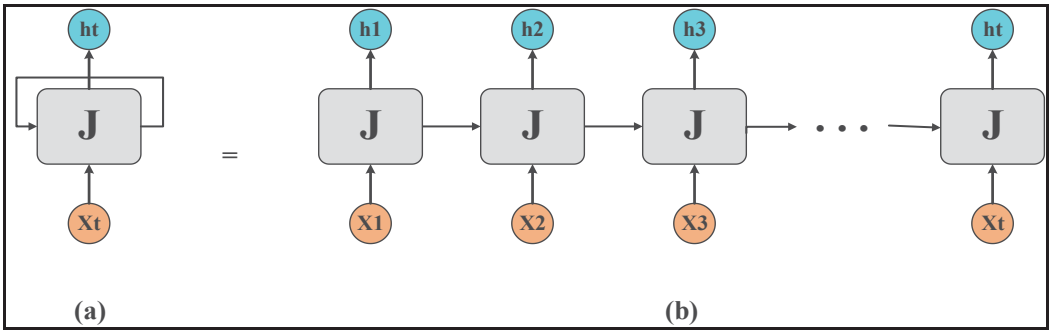

Figure 5.3 A neuron in Recurrent Neural Networks (a); equivalent cascading neurons in FF-ANNs (b)

An RNN comprised of a single neuron for training data of three time steps ($t-1$, t, and $t+1$) is depicted in Figure 5.4. Where X_t is the input at the time step t with the corresponding output of h_t. First, the weighted sum of X_t and the output value h_{t-1} of the previous time-set input X_{t-1} is calculated. Then, a nonlinear activation function such as tangent hyperbolic ($tanh$) is used to produce h_t the corresponding output of X_t. Finally, h_t is fed for training the next time-step input X_{t+1} and this procedure repeats until the last data in sequence. Thus, h_t which is the update function of RNN states, can e formulated as (Chung, Gulcehre, Cho, & Bengio, 2014b):

$$h_t = \begin{cases} 0 & t = 0 \\ \varphi(h_{t-1}\ X_t) & otherwise \end{cases} \quad (5.1)$$

Where φ is an activation function such as *sigmoid*. Consequently, the update function of RNN can be expressed as (5.2), where g is an activation function such as sigmoid or tanh.

$$h_t = g(W_{X_t} + U_{h_{t-1}}) \quad (5.2)$$

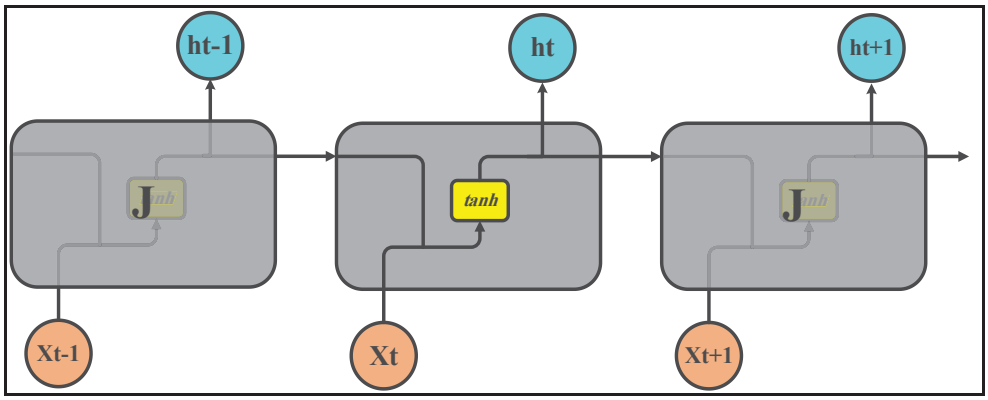

Figure 5.4 A vanilla RNN unit with a single memory gate

Conventional RNNs also called vanilla RNNs (VRNN) utilize hidden states as their memory blocks, thus they are able to preserve the dependencies among sequential data and they are widely used in various machine learning problems (Bengio et al., 2013). However, these types of neural networks suffer from a problem known as the "vanishing gradient" problem (Schmidhuber, 2015). In summary, the neural network problems that are based on gradient learning methods and backpropagation may face a situation in which, the gradient decreases exponentially for each layer, thus leading to slow updating of the weighting functions. Another major problem associated with RNNs is the "exploding gradient" problem (Schmidhuber, 2015) which is the opposite of the vanishing gradient problem. Depending on the nature of data used to train RNNs, long-term errors may accumulate or long-term data may vanish which will result in gradient explosion and gradient vanishing respectively. Although several workarounds are proposed in the literature to mitigate these problems (Tan & Lim, 2019),

RNNs are highly prone to these problems. Thus their application remains limited to the problems with short-term dependencies.

An extended recurrent network architecture known as long-short term memory (LSTM) is first proposed in (Hochreiter & Schmidhuber, 1997b) and later improved in many studies such as (Graves, Mohamed, & Hinton, 2013), which is capable of learning both short-term as well as long-term dependencies among sequential data. Thus mitigating the problems associated with vanilla RNNs.

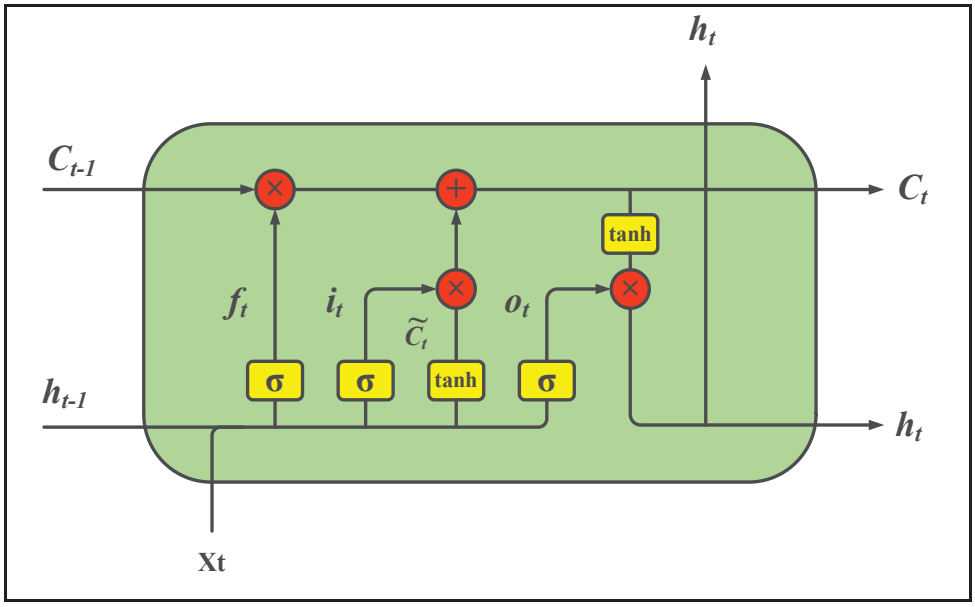

Figure 5.5 Structure of an LSTM network

Vanilla RNNs do not have any cell states. They only have hidden states and those hidden states serve as the memory for VRNNs. Meanwhile, LSTM networks have both cell states and hidden states. On top of that, LSTM networks are equipped with a "forget gate" which allows them to preserve only the important dependencies from past values. LSTM is resistant to the "vanishing gradient" problem. Thus, unlike VRNNs, LSTM networks are capable of learning the long-term impacts of data sequences which makes them suitable for modeling complex nonlinear dynamics of PECs.

5.1.3 Long short-term Memory (LSTM)

An LSTM unit is an extended version of an RNN. While a VRNN unit is comprised of a single gate, an LSTM unit is equipped with four gates as depicted in Figure 5.4 and Figure 5.5, respectively. Consider the unit for the input sequence data X at the time step t (X_t). Along with X_t, the output of the previous time step (h_{t-1}) and the corresponding cell state (c_{t-1}) are considered as the inputs of the LSTM unit at time step t. Correspondingly, the outputs of the LSTM unit are comprised of h_t, the output sequence and c_t the corresponding cell state at time step t.

For the j[th] LSTM unit, the output h_t is obtained by (5.3):

$$h_t^j = o_t^j \tanh\,(c_t^j) \tag{5.3}$$

Where c_t is the data saved in the memory and o_t is the output gate that decides what part of the memory data should be passed to the output h_t. The output gate is calculated using (5.4):

$$o_t^j = \sigma(\mathrm{W_o}[X_t\ h_{t-1}] + B_o)^j \tag{5.4}$$

Where σ is a sigmoid function. W_o is the weighting function and B_o is the bias value. The memory state c_t is obtained using (5.5):

$$c_t^j = f_t^j c_{t-1}^j + i_t^j \tilde{c}_t^j \tag{5.5}$$

Where f_t is the updating function of the previous cell value c_{t-1} and i_t is the updating function of the new memory data $\tilde{c}_t$ which is obtained by (5.6).

$$\tilde{c}_t^j = tanh(W_c X_t + U_c h_{t-1})^j \tag{5.6}$$

Where W_c and U_c are the updating functions of the current input X_t and the previous output h_{t-1}. The f_t function decides how much of the previous data should be forgotten, while the i_t function decides how much of the new data should be added to the memory. These forget and input functions are described in (5.7) and (5.8).

$$f_t^j = \sigma\left(W_f X_t + U_f h_{t-1} + V_f c_{t-1}\right)^j \tag{5.7}$$

$$i_t^j = \sigma(W_i X_t + U_i h_{t-1} + V_i c_{t-1})^j \tag{5.8}$$

Where W, U, and V are the updating functions for the current input data, the previous output data, and the previous memory data, respectively. This enables LSTM to learn short-term as well as long-term dependencies by memorizing the important data and forgetting the junk data.

5.1.4 Gated Recurrent Unit (GRU)

The Gated Recurrent Unit is an RNN similar to LSTM but with fewer gates and without an output gate. The structure of GRU is depicted in Figure 5.6. The output h_t is calculated using (5.9) to (5.12).

$$z_t = \sigma(\mathrm{W_z} x_t + U_z h_{t-1} + B_z) \tag{5.9}$$

$$r_t = \sigma(\mathrm{W_r} x_t + U_r h_{t-1} + B_r) \tag{5.10}$$

$$\tilde{h}_t = \sigma(\mathrm{W_h} x_t + U_h(r_t \odot h_{t-1}) + B_z) \tag{5.11}$$

$$h_t = (1 - \mathrm{z_t}) \odot h_{t-1} + z_t \odot \tilde{h}_t \tag{5.12}$$

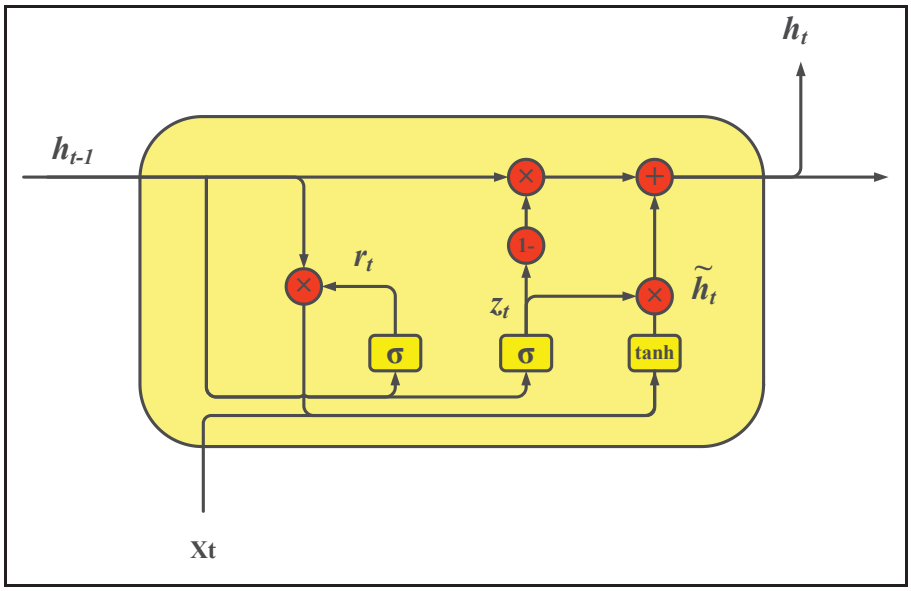

Figure 5.6 Structure of a GRU network

Where x_t is the input vector, h_t is the output vector, $\tilde{h}_t$ is the candidate activation vector, z_t is the update gate vector, r_t is the reset gate vector, W, $U$and B are the weighting functions and bias and ultimately σ is the sigmoid activation function.

It has been claimed that GRU demonstrates a performance similar to LSTM while outperforming LSTM in certain applications with smaller datasets and shorter sequences (Chung et al., 2014b).

In the next sections, a black-box-based Deep Neural Network (DNN) modeling technique is used to perform a comparison between LSTM and GRU as its RNN layer.

5.2 Black-box Modeling Based on LSTM

Inspired by the polytopic black box modeling technique proposed by (Airán Francés et al., 2019), an LSTM network is used in this chapter for learning the short-term and long-term dependencies between input and output signals of power electronic converters.

5.2.1 Data acquisition

The black-box modeling techniques for various PECs utilize different sets of signals as raw training data. For instance, for the buck converter used as a case study later in this chapter, input voltage and output current signals create the input dataset while output voltage and input current signals are used to construct the target dataset. The choice is highly case-dependent. However, the fundamentals of the proposed black-box method remain the same.

5.2.2 Data Preprocessing

Since both LSTM and GRU work with time series and sequences, the input and target datasets ought to be converted into a format that is intelligible for the DNN.

To convert the input signals into a sequential format, each n simulation time-step data can be used to construct a single sequence. The number n is arbitrary and case-dependent. The input dataset X is then created using (5.13) and (5.14).

$$X_t^j = \begin{cases} X_o^j & t < nTs \\ \begin{bmatrix} x_{t-n}^j & \dots & \dots & x_{t-1}^j & x_t^j \end{bmatrix} & t \geq nTs \end{cases} \tag{5.13}$$

$$X_t = [X_t^1 \quad X_t^2 \quad \dots \quad X_t^n]^T \tag{5.14}$$

Where X_t is the input dataset matrix, x_t^j is the value associated with the jth signal in t time step. Furthermore, Ts is the simulation or sampling time step and X_o is the initialization matrix which can consist of zeroes.

The target dataset is generated using (5.15) and (5.16). The principle is to calculate the mean value of each output signal for a given input sequence. The sequence length is again n which is the same as the one chosen in the previous step.

$$Y_t^j = \begin{cases} Y_o^j & t < nTs \\ \frac{1}{n}\sum_{i=0}^{n} y_{t-i}^j & t \geq nTs \end{cases} \tag{5.15}$$

$$Y_t = [Y_t^1 \quad Y_t^2 \quad \dots \quad Y_t^n]^T \tag{5.16}$$

In these sets of equations, j represents the j[th] output signal. Y_o is the initialization output matrix. T_s is the simulation or sampling time step and finally, y_t^j is the value of the j[th] output in time step t.

The structure of the proposed DNN is illustrated in Figure 5.7. To avoid over-fitting, a dropout layer is added to the proposed method as depicted in Figure 5.8.

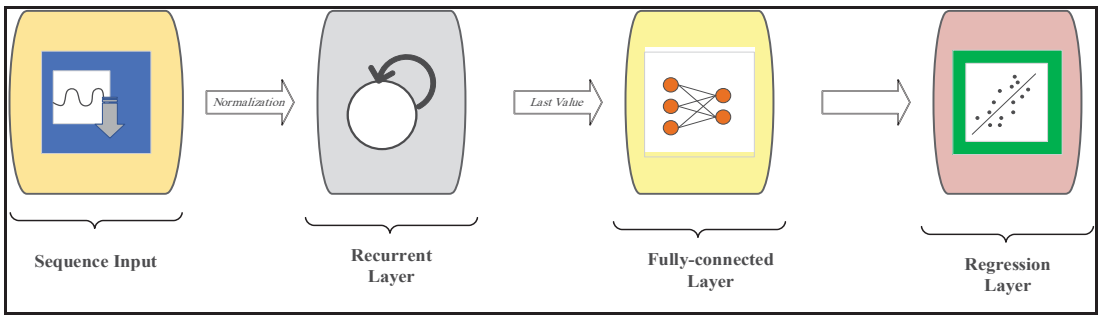

Figure 5.7 The proposed DNN-based black-box modeling method

A normalization method known as Zscore is used to minimize bias. The normalization is done using (5.17). Where for the jth signal, the mean value of a sequence is μ and the standard deviation is σ. Later in the results section of this chapter for LSTM comparison against FF-

ANN and VRNN, the Zscore normalization is applied to all the data in a sequence. However, the normalization is applied element-wise later for comparison against GRU.

$$Z_t^j = \frac{x_t^j - \mu_t^j}{\sigma_t^j} \tag{5.17}$$

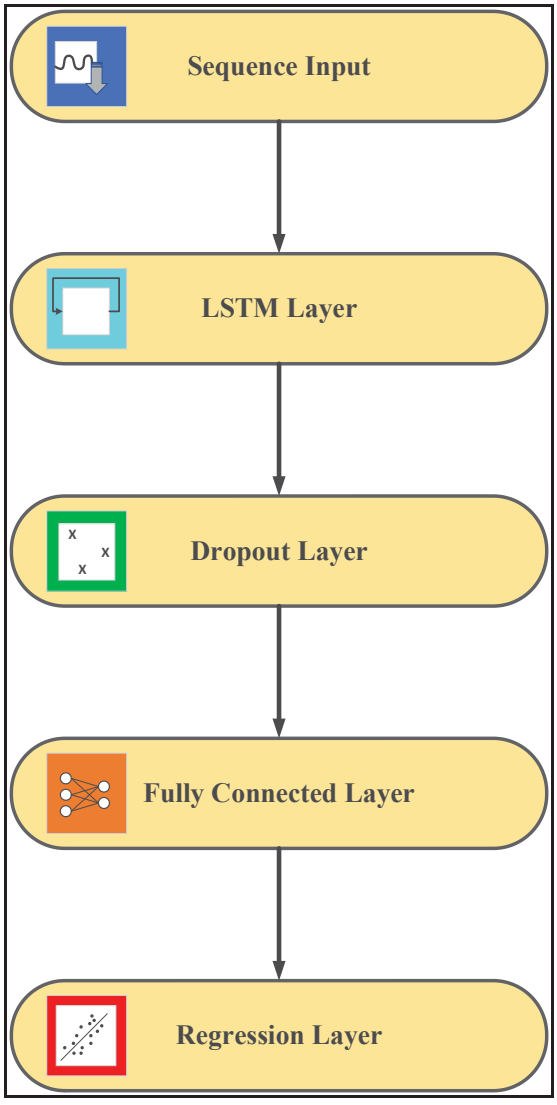

Figure 5.8 Diagram of the LSTM-based deep network utilizing the dropout layer

5.2.3 Network Architecture

As shown in Figure 5.9, the DNN is comprised of several layers. The first layer is the sequence input layer which is responsible for the normalization of the input data. The output mode of this layer is "sequence". The second layer and the most important layer is the recurrent layer. This layer can be either an LSTM or a GRU. The output mode of this layer is "last" and the activation function is $tanh$. The next layer is the fully connected layer which is used to convert the sequential output of the recurrent layer to non-sequential data so that it conforms to the target data format. The final layer is the regression layer that produces the corresponding target data. Using a dropout layer is arbitrary and case dependent.

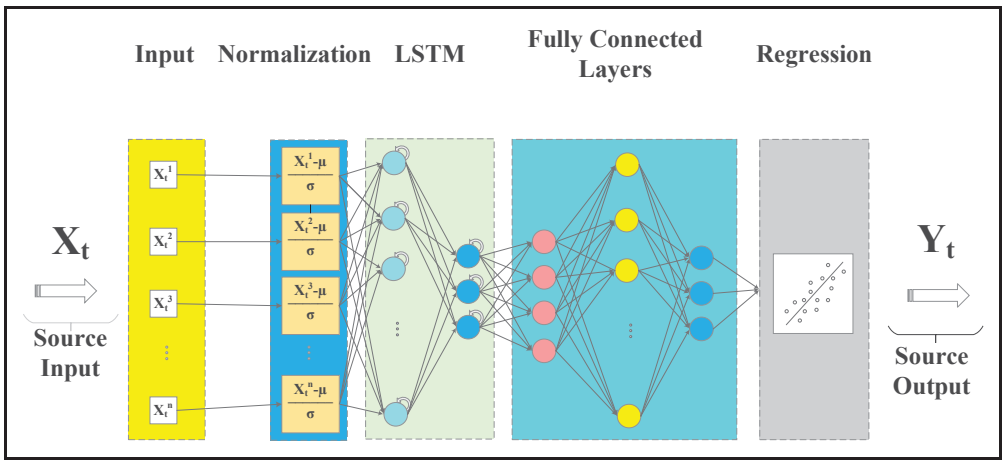

Figure 5.9 Detailed structure of the proposed black-box method

5.2.4 Validation

As shown in Figure 5.10, in order to provide enough training, validation, and test datasets, the obtained data is divided into three sections. The first 70% of the data is used for training the DNN, the next 30% is used for validation, and finally, completely new and unfamiliar operational point datasets are applied as test data to assess the performance of each method under new data.

Aside from the visual comparison, numerical tools are used for the comparison of the modeling methods. To evaluate the overall performance of each method, the Root Mean Square Error (RMSE) as shown in (5.18) is calculated.

$$RMSE = \sqrt{\frac{1}{N}\sum_{j=0}^{n}(y_j - \hat{y}_j)}^{2} \tag{5.18}$$

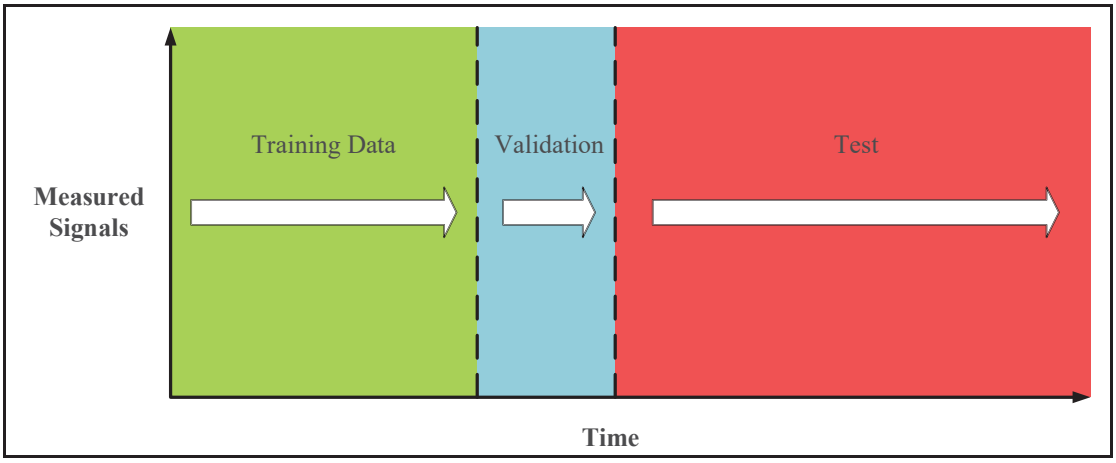

Figure 5.10 Time-domain of the training data, validation data, and test data

Where N is the total number of data points, j is the j^{th} data point, y_j is the actual target of the j^{th} data point and finally $\hat{y}_j$ is the estimated target for the j^{th} data point.

To show the speed of convergence, a fixed number of epochs is determined. The idea is to give each method the same time to prove itself. Finally, the CPU time needed to generate 1 second of output data based on 1 second of input data is calculated to demonstrate the execution speed of each DNN modeling method.

5.3 Transfer Learning (TL)

In the previous sections, we discussed the methodology of the proposed black-box modeling method. However, LSTM and GRU methods demand long training times that can grow exponentially for more complex PECs. Transfer learning (TL) is a method used in various machine-learning applications to transfer the knowledge obtained from one case to similar cases. This section discussed the implementation of transfer learning to improve the training times for LSTM/GRU-based black-box modeling techniques in similar topologies.

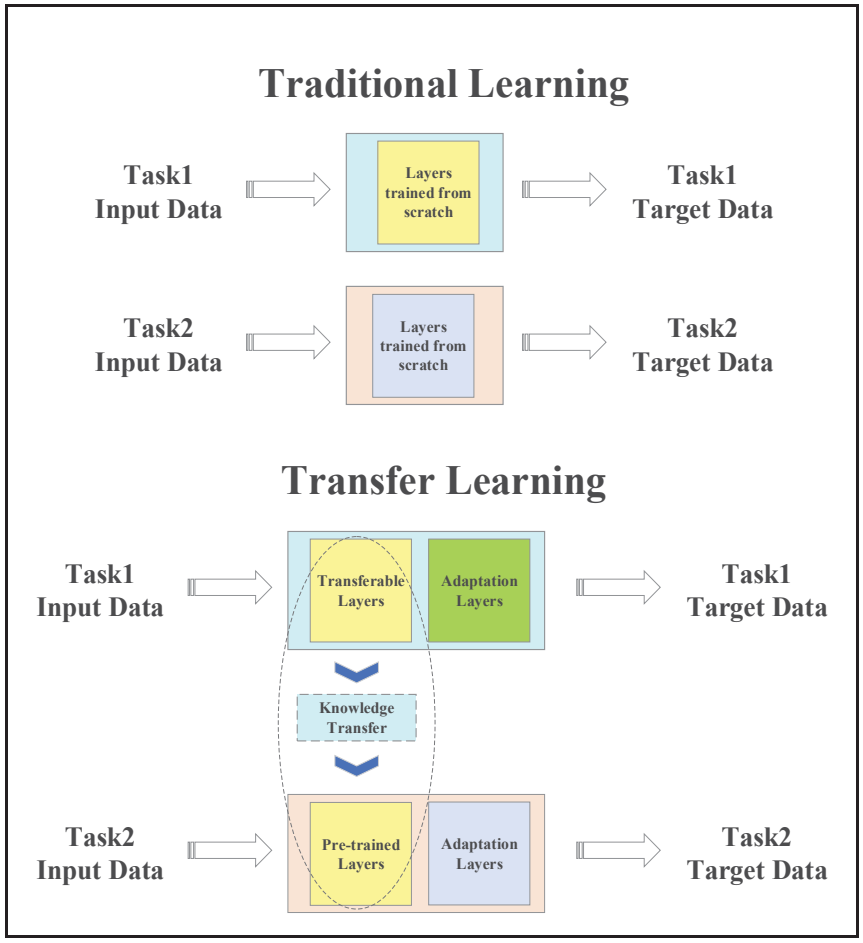

Figure 5.11 Diagrams of Transfer learning and traditional learning

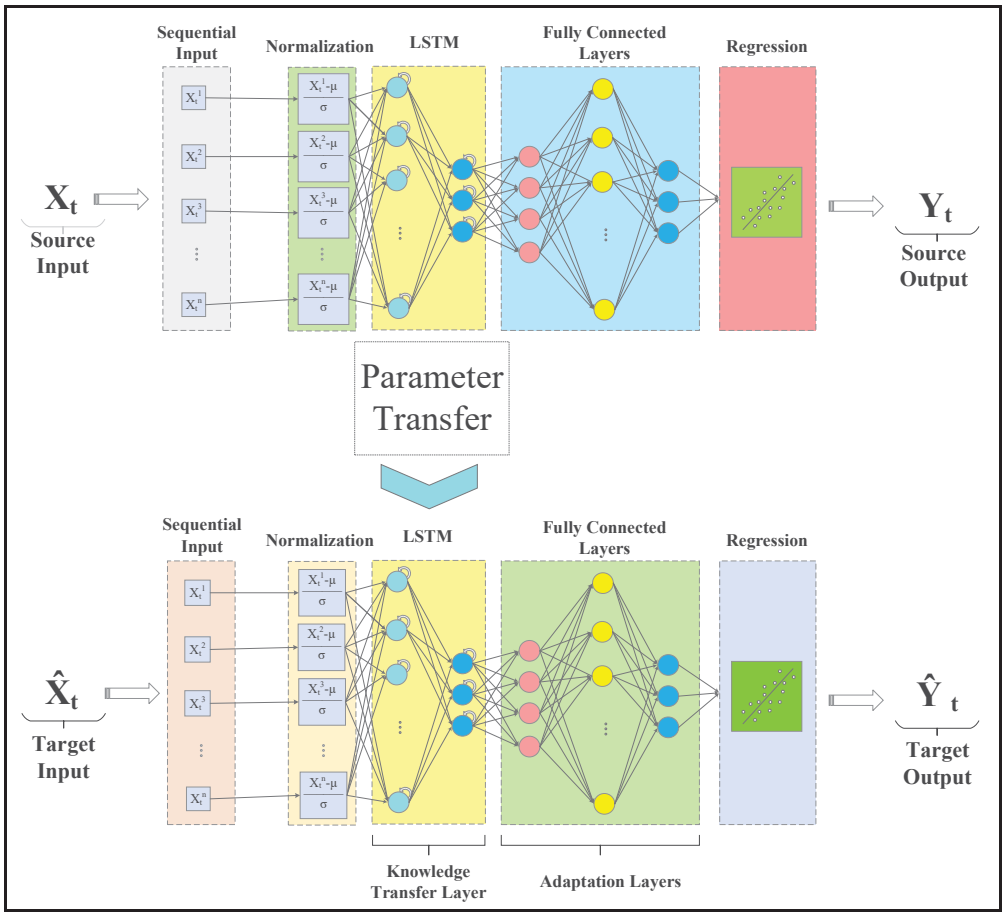

Figure 5.12 Transfer-learning process in LSTM-based Black-box modeling

5.3.1 What is transfer learning?

According to (Pan & Yang, 2010), "Transfer-Learning" is knowledge transfer between different tasks $\tau_T \neq \tau_S$ with different domains $\mathcal{D}_T \neq \mathcal{D}_S$. As shown in Figure 5.11, in traditional learning, each task is trained independently. However, in transfer learning (TL), particular layers among all the trained layers of a task can be reused in another similar task. The unique properties of the target task can be "learned" via the adaptation layer(s). The structure of the black-box deep neural network (DNN) used in this chapter for modeling PECs is illustrated in Figure 5.12. As seen, the weights and biases of the recurrent layers of the source PEC are transferred to the DNN of the target PEC for faster training. The adaptation layers will learn the unique properties of the target PEC and the training times for the target PEC will be

remarkably reduced. While the gating formulas of the source LSTM layer are shown by (5.3) and (5.4), the transferred equivalents of these layers are represented by (5.19) and (5.20). Equally, Root mean square error (RMSE) is calculated using (5.18) as the criteria for the accomplishment of each training scenario.

$$\widetilde{h_t} = \widetilde{o_t} \tanh(\widetilde{c_t}\ ) \tag{5.19}$$

$$\widetilde{o_t} = \sigma(\mathrm{W_o}[X_t\ \ h_{t-1}] + B_o) \tag{5.20}$$

In the results section, this structure is used to train two different power electronic converters. A relatively simple converter which is a synchronous DC/DC buck converter and a relatively complex Hybrid PUC (HPUC) converter.

In each case, an LSTM layer is chosen as a transfer-learning layer, and the other layer or layers are selected as "adaptation layer(s)". Adaptation layers are unique to each training dataset. Using the transfer-learning layer(s), common properties of similar converters can be transferred. This way, redundant calculations are omitted, resulting in a reduction in training time. This claim is investigated in the results section.

5.4 Implementation in Hypersim

Real-time (Software-in-the-Loop (SiL)) simulation of PECs is essential for studying and investigating their impact on the power grid (C. Liu et al., 2020; Ould-Bachir et al., 2015). Hypersim ("Power System Simulation │ Power System Analysis │ HYPERSIM," n.d.), developed by OPAL-RT, is a powerful tool for real-time simulation of power electronics and power systems. One of the main advantages of Hypersim is the ability to import Simulink models into its environment directly. Introduced in Matlab 2021a, the stateful predict block ("Predict Responses Using a Trained Recurrent Neural Network - Simulink," n.d.) allows the import of pre-trained recurrent networks into Simulink. Before the introduction of this block, Simulink did not natively support DNNs. Thus, (Shin, 2018/2023) proposed a method to

rebuild the equivalent of LSTM networks using the basic blocks of Simulink. This method only focused on classification problems and had to be manually tuned and edited for different networks. Unfortunately, even though Simulink natively supports the stateful predict block, it is not compatible with, nor exportable to Hypersim. To mitigate this issue and considering the importance of real-time study of LSTM models in the power grid, a programmatic approach is introduced in this chapter. The proposed method is fully compatible with Hypersim due to utilizing the basic blocks of Simulink and requires zero configuration as a result of it being fully automatic.

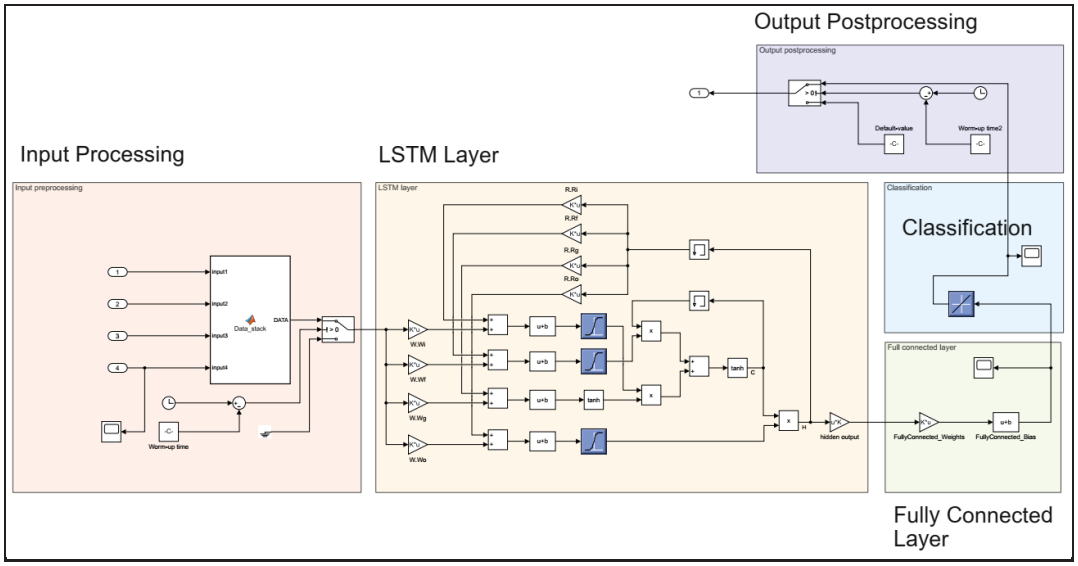

Figure 5.13 The LSTM equivalent model
Adapted from Shin (2018)

5.4.1 LSTM Equivalent subsystem generation

As stated earlier, despite the introduction of the Stateful Predict block since Matlab 2021a, this block is not compatible with Hypersim. Moreover, it imposes a high computational burden due to its unspecialized code.

Before implementing this block in Matlab and natively supporting DNNs in Simulink, (Shin, 2018/2022) proposed a method for generating the equivalent model of LSTM networks using

the basic blocks of Simulink. The schematic of the proposed method in the Simulink environment is illustrated in Figure 5.13.

As seen, this system has a buffer with a warm-up equal to the length of the most extended input sequence in the LSTM network. When the warm-up interval is over, and the buffer is filled using the function implemented inside $Data\_stack$, at each given time, an array of $N_s \times N_i$ is applied to the blocks in the LSTM unit. Where, N_s is the sequence length and N_i is the number of inputs. The output of this layer is given to the fully connected layer so that after applying weights and biases, is exported to the final stages of classification and postprocessing.

Despite being efficient, this method is not automated. Thus, it requires time and effort to manually generate the schematic, implement the buffer function, and apply the corresponding values to the blocks. Therefore, a fully automated and programmatic method is proposed in this chapter to mitigate the issues mentioned above.

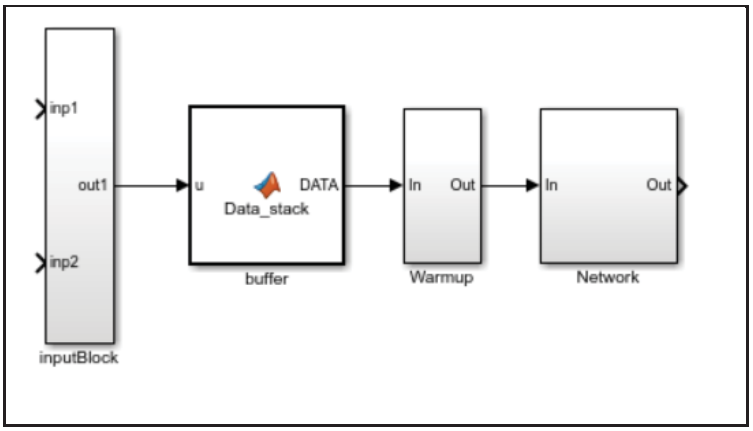

Figure 5.14 Block diagram of the equivalent model generated by MATLAB commands

5.4.2 Programmatical method of LSTM equivalent generation

This method is comprised of two main stages: (1) generating the equivalent schematic of the LSTM model using the basic blocks of Simulink and (2) creating the buffer function and

extracting the values corresponding to each block from the trained LSTM network, then applying these values to the blocks generated in the first stage.

5.4.2.1 Generating the equivalent schematic

First, as an initialization step, the sampling time T_s, the longest sequence length S_L, and the warm-up interval W_t, which is obtained using (5.21), are defined.

$$W_t = S_L \times T_s \tag{5.21}$$

Then, to generate a schematic similar to Figure 5.13, *the add_block* command ("Add Block to Model - MATLAB Add_block," n.d.) is used. Similarly, in order to connect the added blocks using the automated routing algorithm of Matlab, the *add_line* command ("Add Line to Simulink Model - MATLAB Add_line," n.d.) is used. Implementing this approach results in generation of the diagrams depicted in Figure 5.14, Figure 5.15 and Figure 5.16.

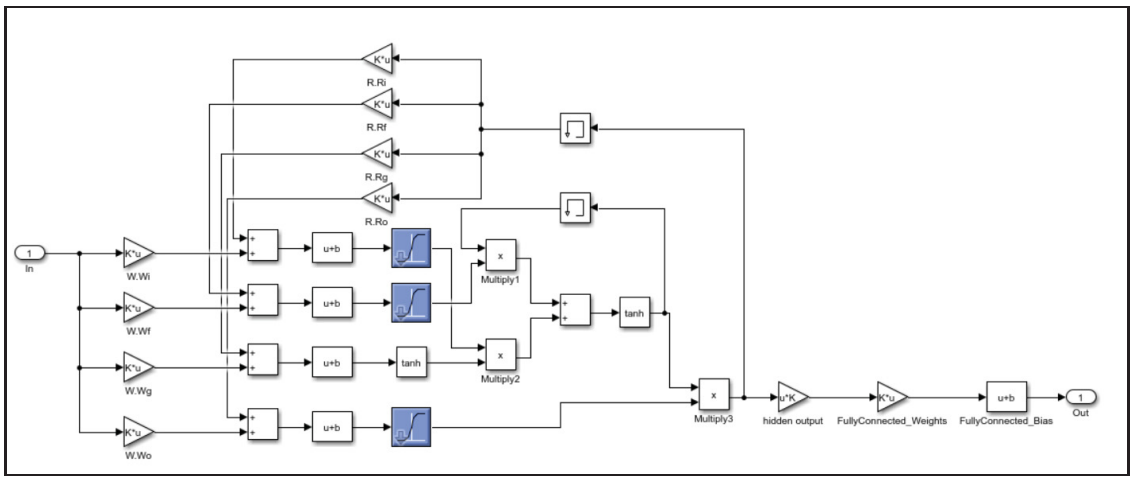

Figure 5.15 Block diagram of the LSTM layer (Network subsystem)

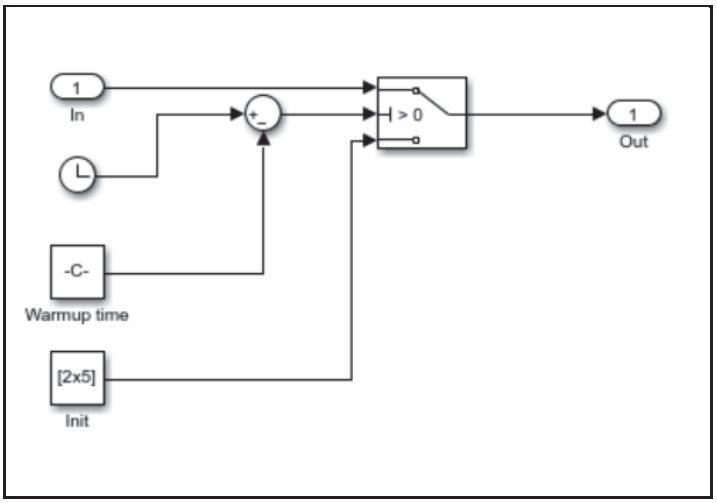

Figure 5.16 Block diagram of the Warm-up subsystem

```
function DATA=Data_stack(u)

persistent save_input1;
persistent save_input2;

if isempty(save_input1)
save_input1 = zeros(1,5);
save_input2 = zeros(1,5);
end

for i=2:1:5
save_input1(i-1) = save_input1(i);
save_input2(i-1) = save_input2(i);
end

save_input1(5) = u(1);
save_input2(5) = u(2);

DATA = [save_input1;save_input2];
```

Figure 5.17 Code lines of the Data_stack function (buffer)

5.4.2.2 Creating functions and applying values

After adding and routing the blocks mentioned above, the next step is applying values to them. For the $Data\_stack$ function that acts as a buffer, the codes illustrated in Figure 5.17, are used. It is worth noting that these lines of code are generated by the proposed method for an LSTM

with only two inputs. Logically, for an LSTM with n inputs, the variables $save\_input1$ to $save\_inputn$, and their corresponding lines will be generated using "for loops."

The values of the other blocks consisting of weights and biases are extracted from the pre-trained LSTM imported to the workspace and addressed as the variable called "net". The code lines used to extract such variables from the pre-trained LSTM are illustrated in Figure 5.18 as proposed by (Shin, 2018/2023).

```
% Weight Generation process
NumHiddenUnits=net.Layers(3, 1).InputSize;
LSTM_W=net.Layers(2,1).InputWeights;
LSTM_b=net.Layers(2,1).Bias;
LSTM_R=net.Layers(2,1).RecurrentWeights;

FullyConnected_Weights=net.Layers(3,1).Weights;
FullyConnected_Bias=net.Layers(3,1).Bias;

W.Wi=LSTM_W(1:NumHiddenUnits,:);
W.Wf=LSTM_W(NumHiddenUnits+1:2*NumHiddenUnits,:);
W.Wg=LSTM_W(2*NumHiddenUnits+1:3*NumHiddenUnits,:);
W.Wo=LSTM_W(3*NumHiddenUnits+1:4*NumHiddenUnits,:);

R.Ri=LSTM_R(1:NumHiddenUnits,:);
R.Rf=LSTM_R(NumHiddenUnits+1:2*NumHiddenUnits,:);
R.Rg=LSTM_R(2*NumHiddenUnits+1:3*NumHiddenUnits,:);
R.Ro=LSTM_R(3*NumHiddenUnits+1:4*NumHiddenUnits,:);

b.bi=LSTM_b(1:NumHiddenUnits,:);
b.bf=LSTM_b(NumHiddenUnits+1:2*NumHiddenUnits,:);
b.bg=LSTM_b(2*NumHiddenUnits+1:3*NumHiddenUnits,:);
b.bo=LSTM_b(3*NumHiddenUnits+1:4*NumHiddenUnits,:);

Last_result = zeros(SequenceLength,1);
Last_result(end,:) = 1;
```

Figure 5.18 Code lines used to extract values from the pre-trained LSTM

Where, $InputSize$ is the number of inputs, $NumHiddenUnits$ is the number of LSTM units in the recurrent layer, $LSTM\_W$, $LSTM\_b$ and $LSTM\_R$ are the input weights, biases and recurrent weights respectively. In addition, $FullyConnected\_Weights$ and $FullyConnected\_Bias$ are the fully connected layer weights and biases, respectively. Lastly, "i", "f", "g" and "o" are the suffixes corresponding to the input, forget, gate, and output values of the LSTM unit, respectively.

Since, in this chapter, the output is generated through regression, the classification block is not generated. However, this block can be generated as needed, using the steps explained above.

5.5 Results

In this section, the results of the study are presented and discussed. First, for evaluating the proposed black-box modeling method using LSTM, is compared against FF-ANN and VRNN, and the results are presented in this section. Moreover, using the same architecture and implementing LSTM and GRU as the recurrent layer of the proposed method, is used to compare the performance of these two structures. Ultimately, the Hypersim-compatible model is exported and evaluated against its counterpart in Simulink while running on a local host target.

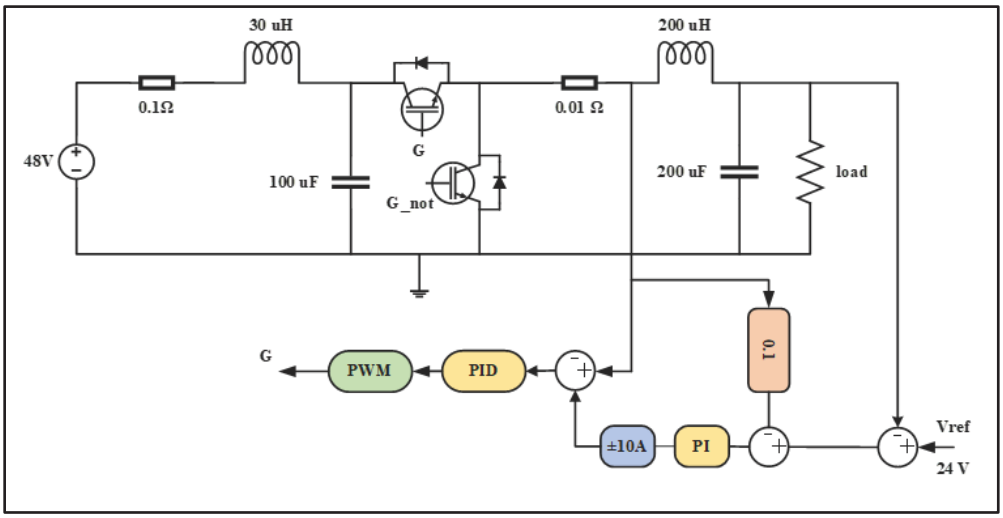

Figure 5.19 Diagram of the DC/DC buck converter with droop control

5.5.1 Comparison of LSTM-based Black-box modeling against FF-ANN and VRNN

For validating the proposed LSTM-based black-box method, a regulated synchronous DC/DC buck converter as depicted in Figure 5.19 is simulated in a MATLAB/Simulink environment. The buck converter is equipped with a PID controller for adjusting the output voltage, droop

control, and output current limiter, and it is operated using the SPWM switching method. The parameters of the selected converter and the training parameters are listed in Table 5.1 and Table 5.2, respectively.

Table 5.1 Simulation Parameters for LSTM-based DC/DC converter modeling

Parameter	**Value**
Switching Frequency	10 kHz
Input Voltage	48 V
Sampling Timestep	50 µs
Simulation Mode	Discrete
Reference Voltage	24 V
Droop Control Gain	0.1
Output Current Limits	±10 A

Input voltage V_i and output current I_o are considered as the input signals, while the output voltage V_o and input current I_i are selected as the output signals.

Six different operation points are applied to the converter from $T = 0s$ until $T = 600ms$ each for a period of $100ms$. The aforementioned signals are logged into the MATLAB workplace. Using the preprocessing methods described in 5.2.2, the training datasets are obtained.

In order to demonstrate the superior performance of the proposed LSTM method over FF-ANN and VRNN methods, a constant number of hidden layer neurons are applied to each of them. The buck converter is modeled using the same training data. The performance of each method in generating the output voltage while facing known operation points is demonstrated in Figure 5.20. As seen, all of the three methods effectively model the steady-state signals while in transient modes, VRNN demonstrates a poor performance.

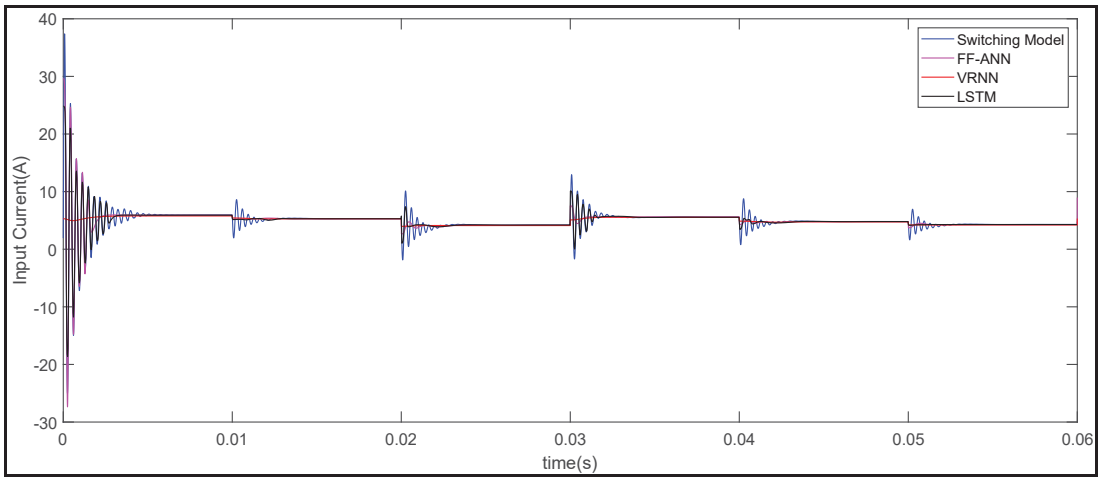

Figure 5.20 Input Current(A) signals for known operation points

Table 5.2 Training Options and Hyper-parameters

Parameter	**Value**
Solver	adam
Gradient Threshold	10
Initial Learning Rate	0.0001
Learn Rate Schedule	piecewise
Learn Rate Drop Factor	0.99
Learn Rate Drop Period	10
MiniBatch Size	256
Shuffle	once
Validation Frequency	30

The performance of each method in generating the output voltage and the input current, facing new operation points is demonstrated in Figure 5.21 and Figure 5.22, respectively. As shown, the FF-ANN method performs poorly against new operation points while VRNN is still successfully following the steady-state signals. By mitigating the problems of FF-ANN and VRNN, the LSTM method can effectively model both transient and steady-state responses of

the buck converter. For a better comparison, the response signals are zoomed in and depicted in Figure 5.23.

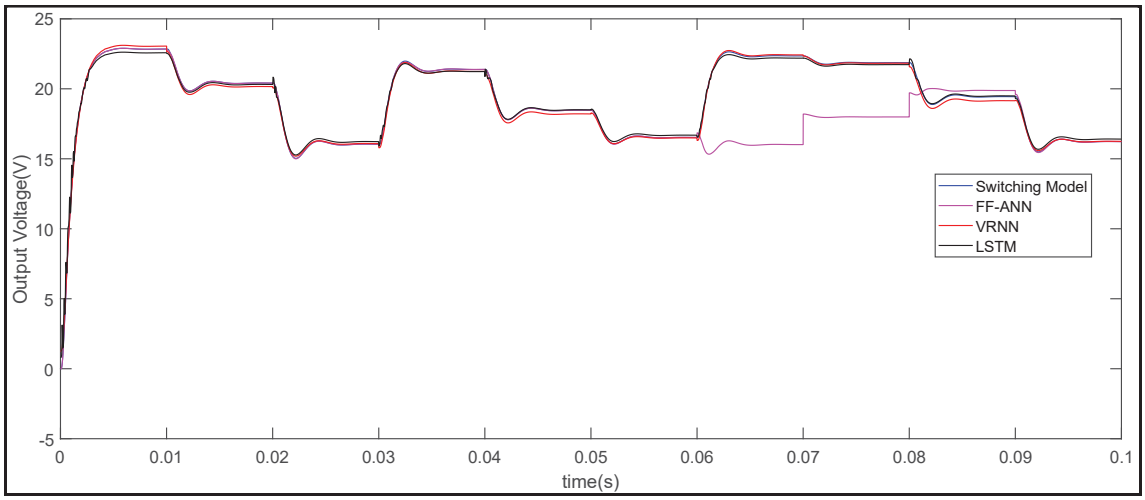

Figure 5.21 Output Voltage(V) signals for new operation points

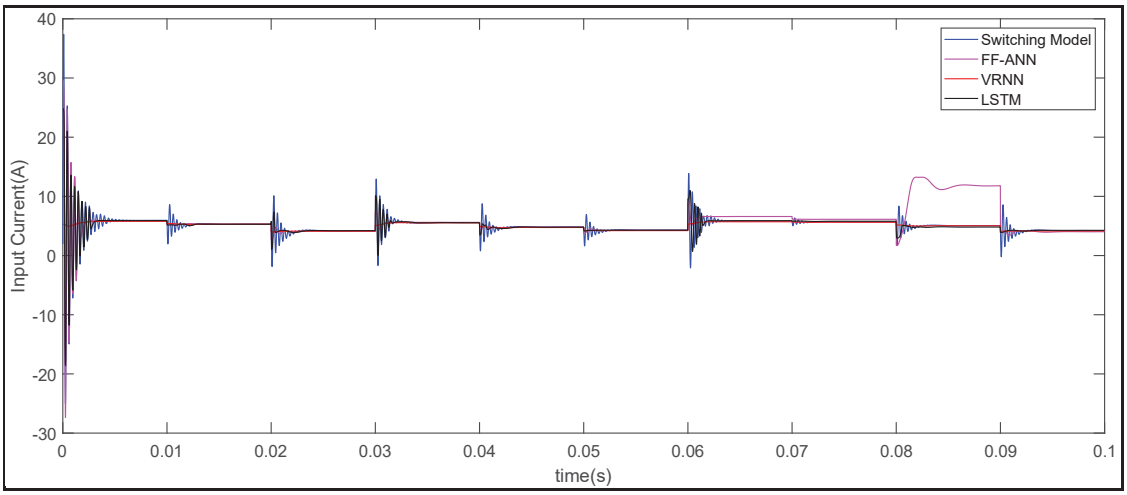

Figure 5.22 Input Current(A) signals for new operation points

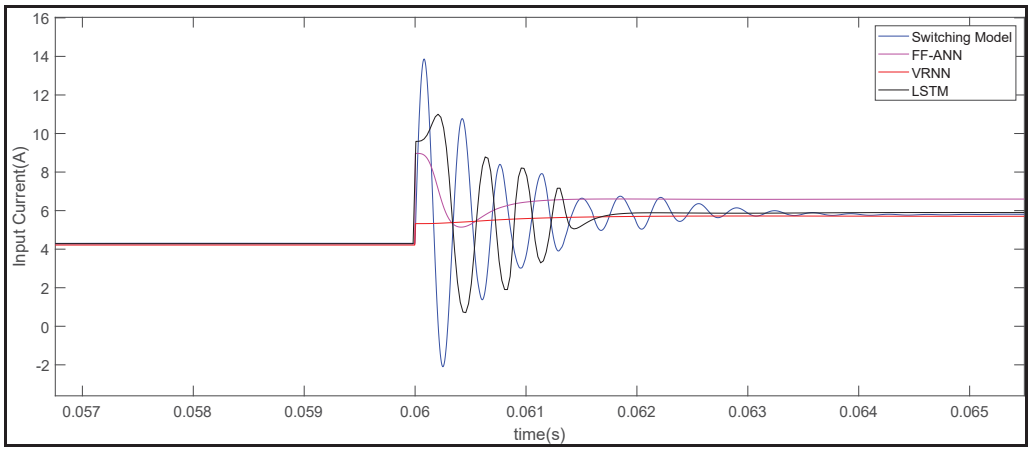

Figure 5.23 Comparison of the modeling methods as a new operating point is selected at $t = 600ms$

The results demonstrated that, unlike FF-ANN and VRNN methods, the proposed LSTM-based method is effectively capable of generating a large-signal model that performs well under steady state as well as transient modes.

The accuracy of the proposed model is a function of its network size. Adjusting the network size of the LSTM layer gives enough degree of freedom for a trade-off between accuracy and computational burden.

5.5.2 Comparison of LSTM Modeling Method Against GRU

For a fair comparison between LSTM and GRU, the same regulated synchronous buck converter shown in Figure 5.19 is modeled using the proposed method with both GRU and LSTM as its recurrent layer. The parameters of the simulation are the same as in Table 5.1. The sequence length n is set to 5 and both GRU and LSTM layers are comprised of 100 units. The training options and hyper-parameters match the ones presented in Table 5.2.

The training process is performed in 4 different manners: 1- full-data GRU, 2- full-data LSTM, 3- limited-data GRU, and 4- limited-data LSTM. Full-data indicates that the original training and validation datasets are used while limited-data means that only a fraction of the original

datasets are used. The idea is to investigate the superiority of GRU when fewer samples are available.

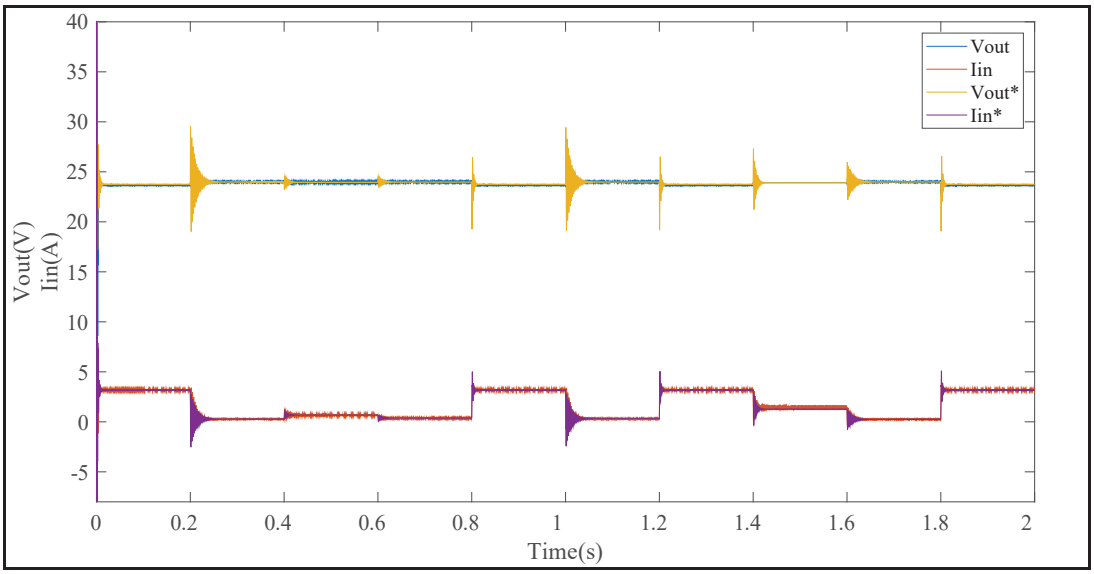

Figure 5.24 Input current(A) and output voltage(V) using GRU-Limited

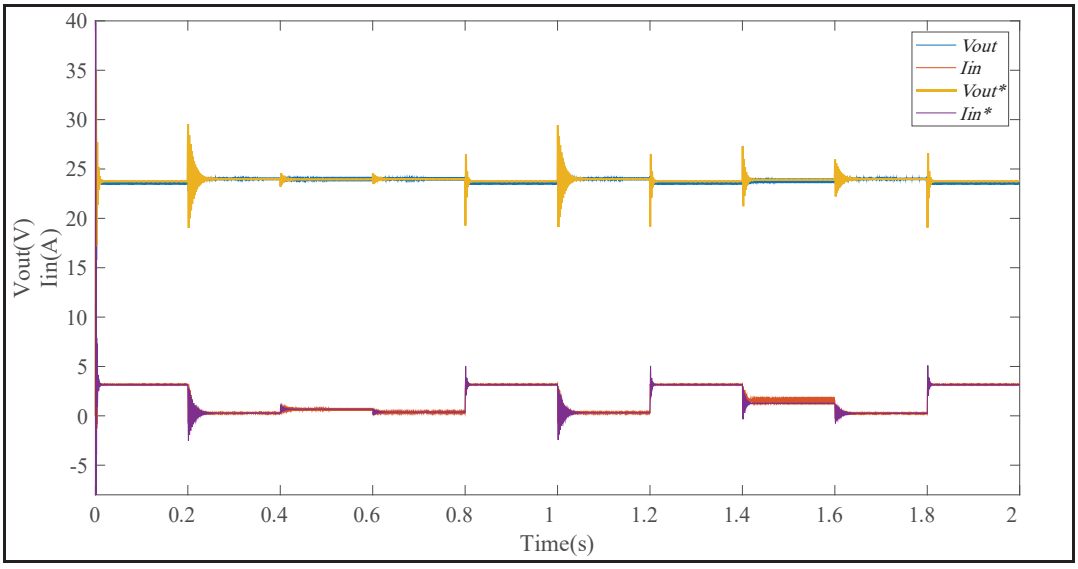

Figure 5.25 Input current(A) and output voltage(V) using LSTM-Limited

In order to generate the limited version of the datasets, they have been down-sized to 1/1000th of the original data size.

The input signals V_i and I_o are considered as the input and the signals V_o and I_i are considered as the target. The converter is simulated for $t = 2s$ and the load is changed each $100ms$. From $t = 0$ to $t = 1s$ is used for training and validation and from $t = 1s$ to $t = 2s$ is used for testing. In order to investigate the execution speed of each method, the "tic-tac" method in Matlab is used. The processor used for performing these tests is a 4-core Intel(R) Xeon(R) CPU E3-1225 v6 @ 3.30GHz using Windows 10 – 64bit, equipped with 16GB of RAM.

As shown in Table 5.2, the initial learning rate drops by 0.99 every 10 epochs. To have a fair comparison, a fixed number of maximum epochs is considered. This fixed number is 100 for the full data size and 10000 for the limited data size. The reason for this difference is that the down-sized data are processed significantly faster but convergence is unlikely to happen within 100 epochs.

The black-box deep neural network is trained under four different scenarios and the corresponding results are depicted in Figure 5.24-Figure 5.27. Where V_{out} and I_{in} are the target output signals, while V_{out}^* and I_{in}^* are the output signals predicted by each method. As can be seen, LSTM and GRU demonstrate a close performance under the original dataset. The down-sized dataset is slightly better modeled using GRU.

The numerical results of each method are listed in Table 5.3. As shown, the GRU method under the original dataset takes less training time, leads to less RMSE, and executes faster. The limited-data versions of both LSTM and GRU when compared to their full-data counterparts, take significantly less time while resulting in notably smaller RMSE and slightly faster execution speed. The limited version of GRU takes slightly more time for training than LSTM.

From the results obtained and demonstrated in this section, it can be interpreted that the GRU method is superior compared to LSTM in several ways. First of all, GRU takes less time to train when the sample size is enormous. Moreover, GRU provides a smaller RMSE compared to LSTM and this is independent of the sample size. And ultimately, GRU executes slightly faster.

The reason for the better performance of the limited versions compared to the full-data versions is that the limited versions are minuscule which allows 100 times iterations.

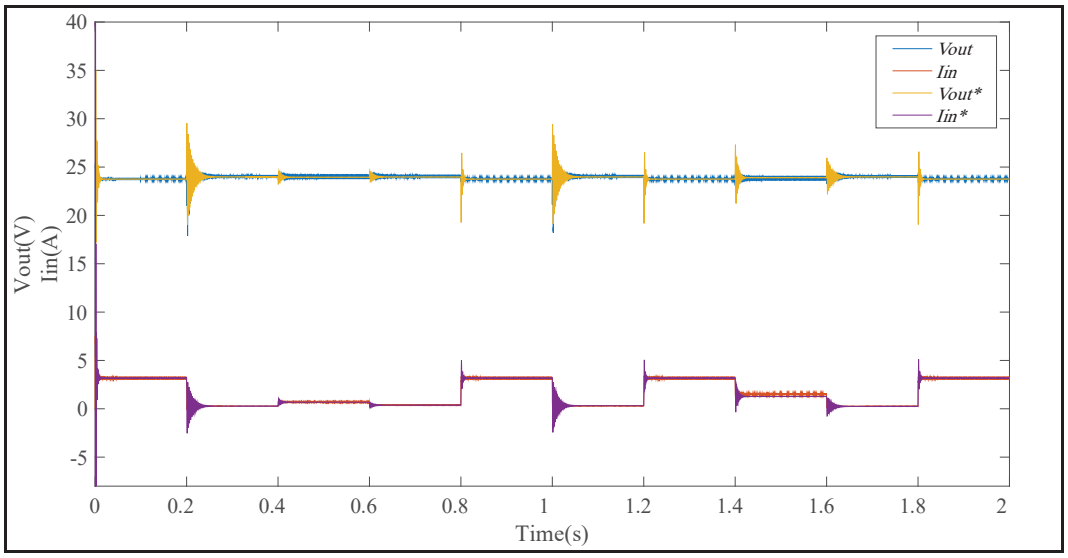

Figure 5.26 Input current(A) and output voltage(V) using GRU-Full

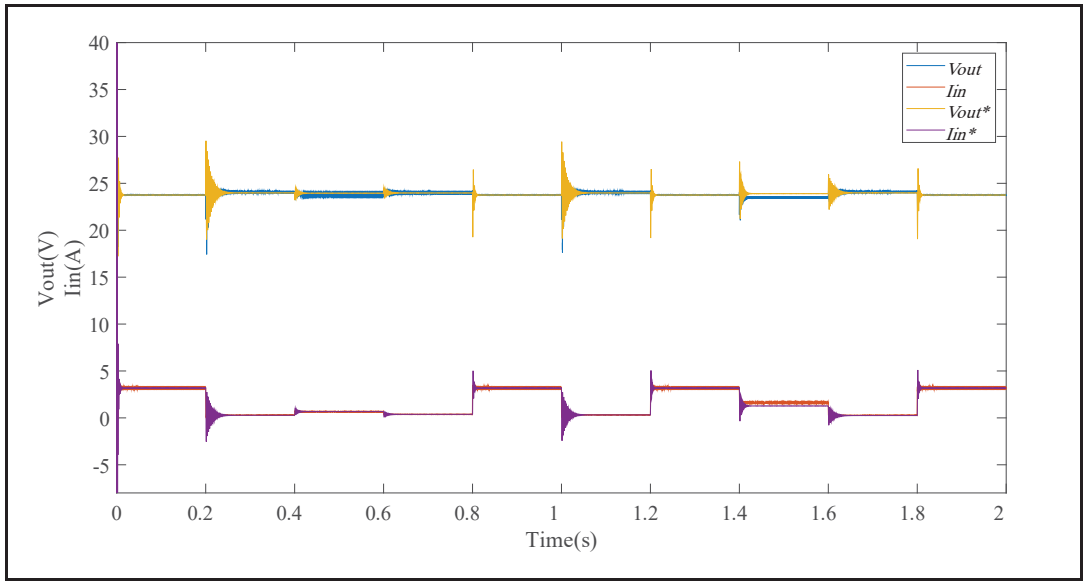

Figure 5.27 Input current(A) and output voltage(V) using LSTM-Full

Table 5.3 Training Results Based on the Type of The RNN Layer and Sample Size

Method	LSTM-Full	GRU-Full	LSTM-Limited	GRU-Limited
Training Time(s)	476	348	108	110
RMSE	0.4322	0.3282	0.1976	0.1106
Sample Size	28000	28000	28	28
Execution Time(s)	2.4051	1.90762	2.3774	1.9317
Max Epoch	100	100	10000	10000

The results obtained in this section demonstrate the superiority of the GRU method for model extraction of PECs when enormous data samples are available. The GRU method is remarkably faster to train, with notably less error (RMSE) while providing faster execution. The LSTM method is faster to train when fewer data samples are available. However, the GRU method remains more accurate (less RMSE) and slightly faster to execute.

Although both methods are effectively capable of modeling PECs, the GRU method is preferred when the training data are limited and/or when execution speed is vital.

Table 5.4 Common simulation parameters for Buck1 and Buck2

Parameter	**Value**
Simulation Time-step	50 µs
Input Voltage	100 V
Reference Output Voltage	48 V
Switching Frequency	10 kHz

5.5.3 Using Transfer Learning to Accelerate Training Times

In the first part of this section, the same synchronous DC/DC buck converter depicted in Figure 5.19 and discussed in the previous sections is used to assess transfer learning in this section. However, the parameters are different for each case called Buck1 and Buck2. The common simulation parameters as well as different topology parameters for each case are shown in Table 5.4 and Table 5.5, respectively. Ultimately, training parameters are shown in Table 5.6.

To investigate the impact of data sampling accuracy in transfer learning, a smaller version of each original dataset (full-data) is generated using a sampling ratio of 1:1000. In this chapter, this new dataset is called “limited-data”. It is worth mentioning that certain training parameters in Table 5.6 are different for full-data and limited-data datasets.

Table 5.5 Buck1 and Buck2 parameters

Parameter	**Buck1**	**Buck2**
R1	0.1 Ω	0.2 Ω
L1	30 µH	40 µH
R2	0.01 Ω	0.03 Ω
L2	200 µH	400 µH
C1	100 µF	50 µF
C2	200 µF	300 µF

The training progress diagram for Buck2, before and after TL, is illustrated in Figure 5.28, where the x-axis is the training epochs and the y-axis is the root-mean-square error (RMSE). Training times for each scenario are shown in Table 5.7. It is worth mentioning that a PC equipped with Intel® Xeon® E3-1225 3.30 GHz CPU, 16GB DDR4 RAM, and 1TB SSD is used for training.

As we see, training times for Buck2 are significantly reduced after using TL in both full-data and limited-data scenarios. The actual and predicted values of the target signals of Buck2 i.e. output voltage(V) and input current(A), after using TL are shown in Figure 5.29.

Table 5.6 Training Parameters for the Buck Converters

Parameter	**Value**
LSTM Layer1 Size	10
Fully Connected Layer1 Size	10
Solver	adam
Gradient Threshold	1
Initial Learning Rate	0.01
Learn Rate Schedule	piecewise
Learn Rate Drop Factor	0.9
Learn Rate Drop Period (Full data)	10
Learn Rate Drop Period (Limited Data)	100
MiniBatch Size (Full Data)	128
MiniBatch Size (Limited Data)	8
Shuffle	Every-epoch
Validation Frequency	50
Validation Patience	10

Different operation points are applied to demonstrate the appropriate dynamic response of the model. The model is accurate before and after TL. However, the training times are remarkably different. This is important for modeling very complex PECs where training times are extensive.

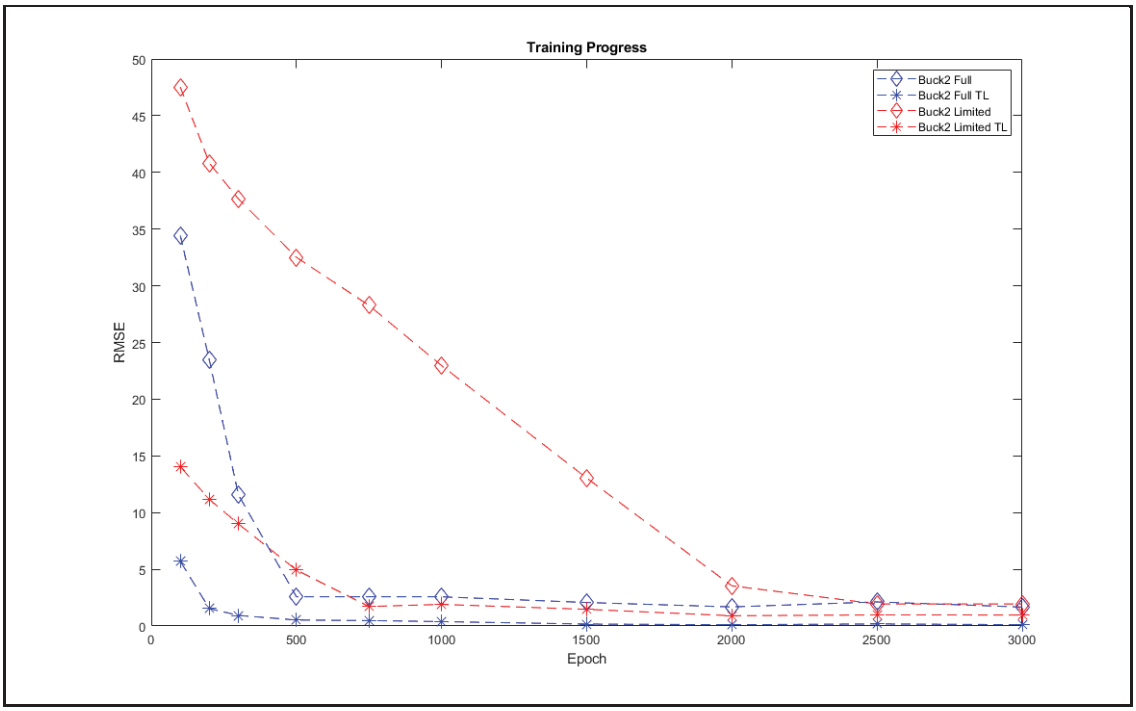

Figure 5.28 Training progress diagram of Buck2 before and after transfer learning

Table 5.7 Training Times of Different Training Modes for Buck1 and Buck2

Training Mode	**Training Time(s)**
Buck1 Full-data	48.21
Buck1 Limited-data	246.78
Buck2 Full-data	121.92
Buck2 Limited-data	601.68
Buck2 Full-data TL	84.24
Buck2 Limited-data TL	169.03

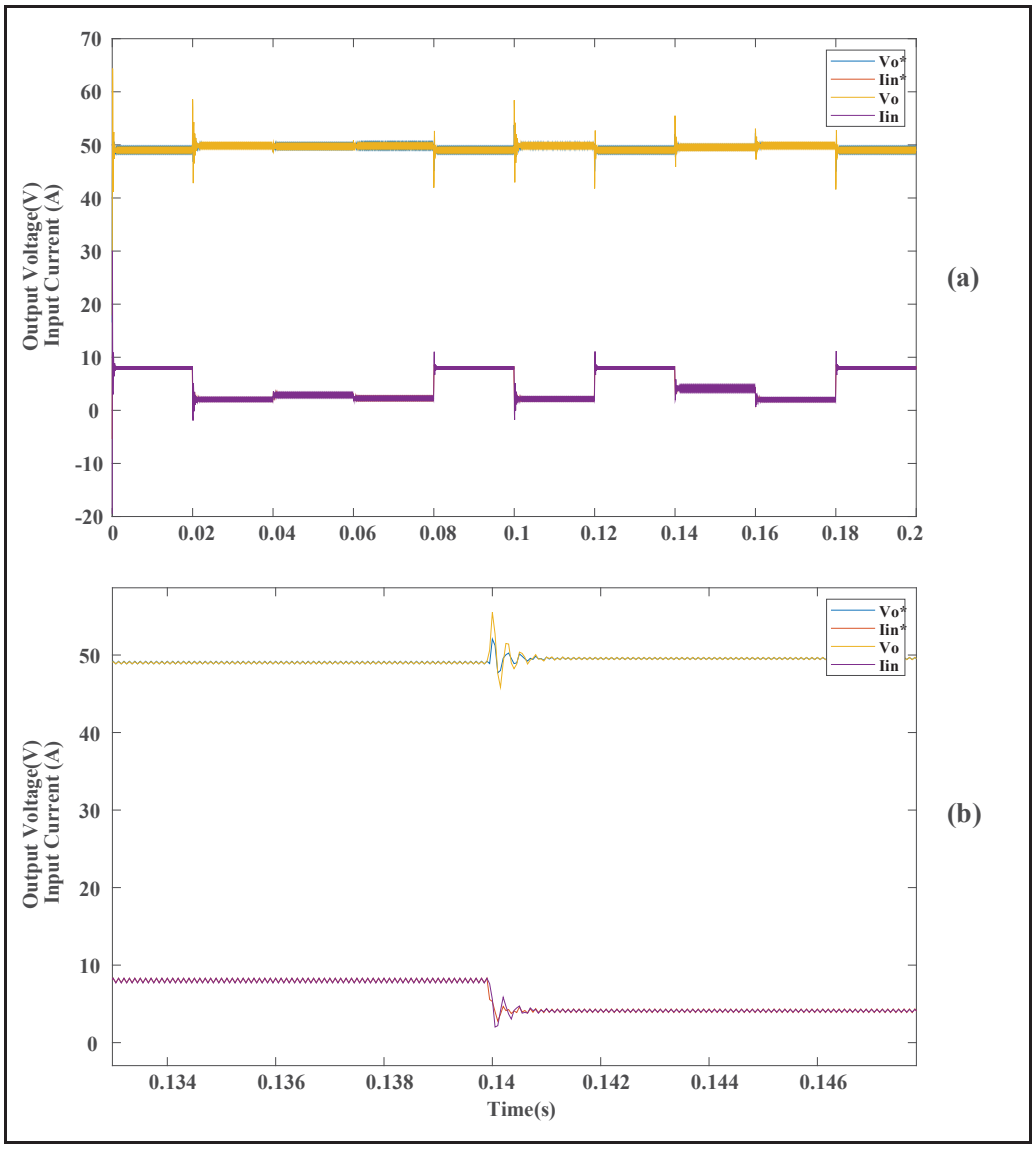

Figure 5.29 Actual and predicted output voltage(V) and input current (A) of Buck2 after TL (a); Operation point change of the identical waveforms of Buck2 at t=0.14 (b)

Using the same PC, a more complex and more demanding topology which is the HPUC illustrated in Figure 5.30, is simulated and trained based on the parameters shown in Table 5.8 and Table 5.9. Additionally, the training Parameters are listed in Table 5.10.

The results are shown in Table 5.11. As we see, considering the same training times, transfer learning results in lower RMSE and, thus, superior accuracy. However, the execution times remain unchanged.

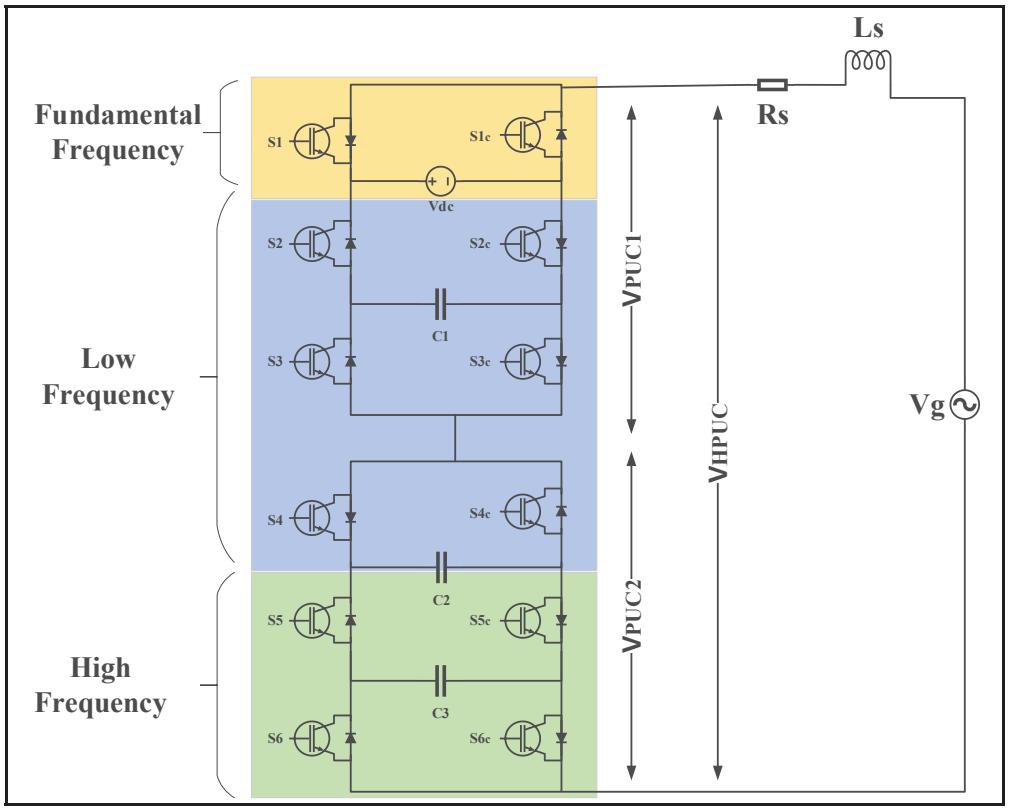

Figure 5.30 Topology of the test case Hybrid PUC (HPUC)

Table 5.8 Simulation Parameters for HPUC Converters

Parameter	Value
Simulation Time-step	20 µs
Grid Voltage (Peak)	170 V
Line Frequency	60 Hz

Table 5.9 HPUC1 and HPUC2 Parameters

Parameter	**HPUC1**	**HPUC2**
Grid Resistance	0.2 Ω	0.1 Ω
Grid Inductance	1000 μH	500 μH
Capacitor 1	2500 μF	2100 μF
Capacitor 2	2500 μF	2100 μF
Capacitor 3	2500 μF	2100 μF
Input DC Voltage	200 V	160V

Table 5.10 HPUC training parameters

Parameter	**Value**
LSTM Layer1 Size	20
LSTM Layer2 Size	18
Fully Connected Layer1 Size	17
Solver	adam
Gradient Threshold	1
Initial Learning Rate	0.01
Learn Rate Schedule	piecewise
Learn Rate Drop Factor	0.9
Learn Rate Drop Period	5
MiniBatch Size	512
Shuffle	Every-epoch
Validation Frequency	10
Validation Patience	70

Table 5.11 Training times of different training modes for HPUC1 and HPUC2

Parameter	HPUC1	HPUC2 Pre-TL	HPUC2 Post-TL
Training Time	361s	360s	363s
Execution Time	4.65s	4.50s	4.50s
RMSE	6.045	6.6	3.38

The output voltage (Vo) of HPUC2 and its predicted equivalent (marked by the "*" character in the legend) before and after transfer learning is shown in Figure 5.31-a and Figure 5.31-b, respectively. As seen, the output voltage predicted by the LSTM-based model is superior using transfer learning.

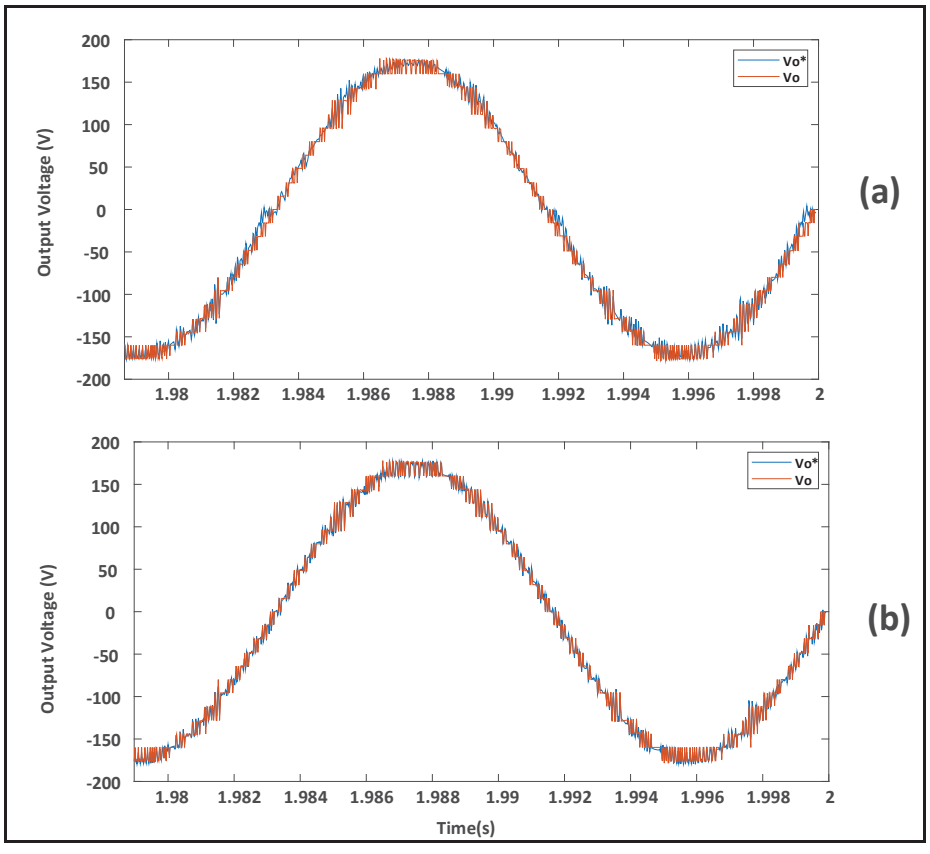

Figure 5.31 Output voltage (Vo) of HPUC2 before (a) and after (b) transfer learning

Training times are relatively extensive for DNN-based black-box modeling of PECs. In this section, a technique called transfer learning is used to mitigate this problem. Through simulations in Matlab/Simulink, where different operation points are applied to a buck converter, transfer learning is proven to reduce the training times and prediction accuracy of an LSTM-based black-box PEC modeling method. This section of the chapter also evaluates the impact of transfer learning on a relatively new and more complex power electronic converter called HPUC. The transfer learning was applied to this converter in two different configurations, and the simulation results proved that for the same training time, the accuracy of an LSTM-based DNN model was considerably improved.

5.5.4 Implementation of the proposed method on Hypersim

In this section of the chapter, To investigate the performance of the proposed method, the same synchronous DC/DC buck converter shown in Figure 5.19 is simulated, and the obtained model is imported to Hypersim for assessment. A variable resistive load is connected to the converter to act as a time-variant load. The simulation parameters are listed in Table 5.12.

Table 5.12 Simulation parameters for the buck converter

Parameter	Value
Simulation Time-step	50 µs
Input Voltage	100 V
Reference Output Voltage	48 V
Switching Frequency	10 kHz

The waveforms of the simulation results, as well as the predicted values using the predict command in Matlab, are illustrated in Figure 5.32. Where Vout is the output voltage, Vout* is the predicted output voltage using LSTM, Iin is the input current and Iin* is the input current predicted by the LSTM model. As seen, the LSTM model is almost identical to the detailed switching model used in the Simulink simulation.

In order to verify the performance of the equivalent LSTM generated by the method proposed in this chapter, an equivalent standalone circuit is created. The corresponding waveforms of the original trained LSTM and its equivalent circuit are illustrated in Figure 5.32 and Figure 5.33, respectively. As seen, the equivalent circuit effectively models the behavior of the buck converter.

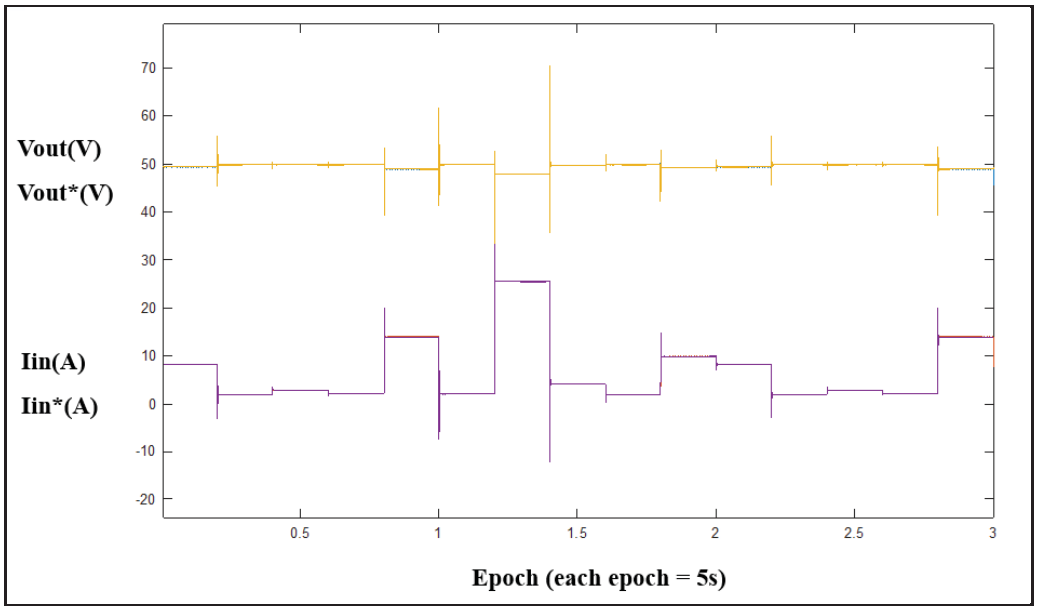

Figure 5.32 The actual and predicted waveforms of the buck converter using the trained LSTM network

Using “Hyperlink.tlc”, the general hyperlink target library, and “hyperlink.grt.tlc”, which is the C code creation library, the C code and the binary file of the model are compiled. The libraries mentioned above generate a *.def file in the same folder that the C code and the binary file are exported. The *.def file can be imported to Hypersim using the Simulink toolbox import tool known as UCM, into the environment of Hypersim.

After importing the equivalent LSTM model into Hypersim, Software-in-the-Loop (SiL) simulation is done using a Windows 10 PC equipped with an SSD, an Intel Xeon E3-1225 V6 3.3GHz CPU, and a 16GB DDR5 2400MHz RAM in a localhost mode. The output waveforms

are illustrated in Figure 5.34 where the time stamp is 10ms, and in Figure 5.35 where the time stamp is 1s.

As seen, the proposed method can effectively extract and generate the equivalent of LSTM-based black-box models of PECs into Hypersim for SiL simulation purposes.

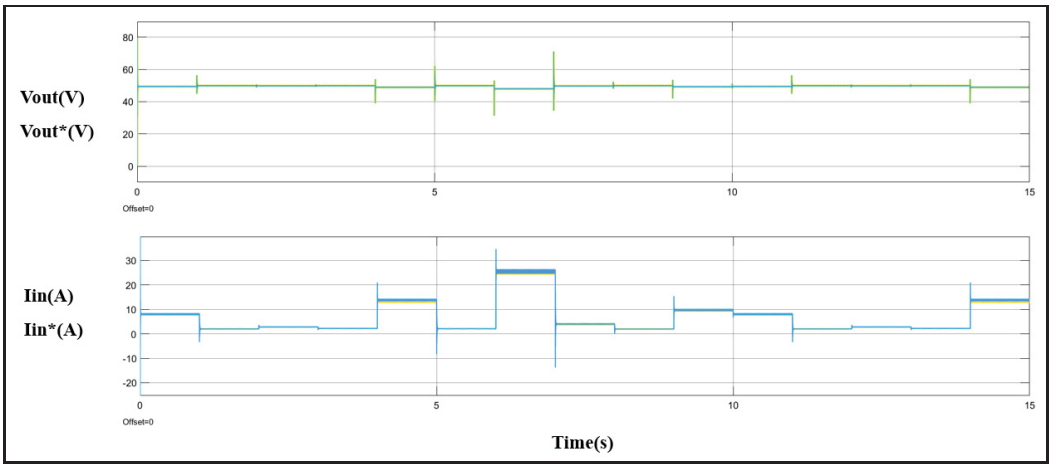

Figure 5.33 The actual and predicted waveforms of the buck converter using the equivalent LSTM model generated by the proposed method

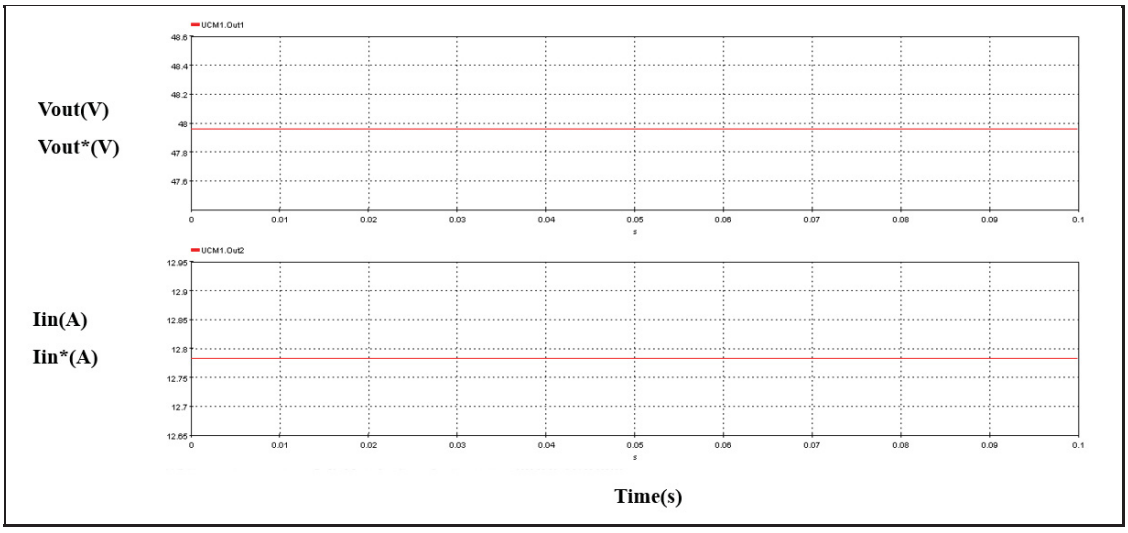

Figure 5.34 SiL simulation results with the time stamp of 10ms using the equivalent LSTM model

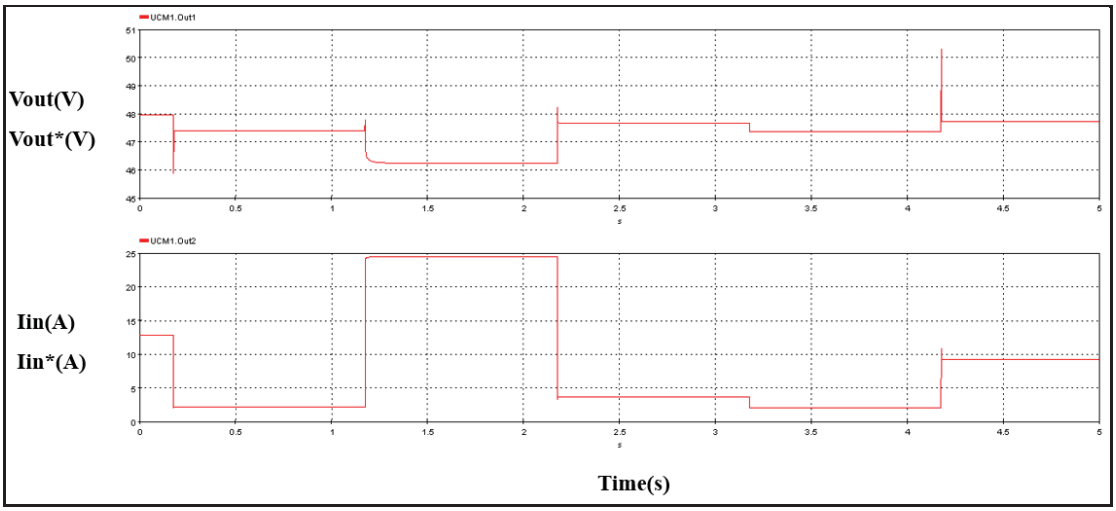

Figure 5.35 SiL simulation results with the time stamp of 1s using the equivalent LSTM model

Based on the obtained results in this section using simulations in Simulink and Hypersim, we can conclude that the proposed equivalent circuit is capable of accurately reproducing the behavior of the original LSTM-based black-box model in Hypersim. This results in a model feasible for real-time simulation studies.

5.6 Conclusion

In this chapter, we presented an advanced black-box modeling technique for power electronic converters (PECs) using machine learning. The proposed method takes advantage of modern deep learning architectures of Long Short-Term Memory (LSTM) networks and Gated Recurrent Units (GRUs). The motivation and inspiration for developing such a black-box modeling method were to effectively extract the non-linear behavior of practical PECs that often ack accurate data sheets.

We began the study by evaluating the limitations of traditional Feed-Forward Artificial Neural Networks (FF-ANNs) and Vanilla Recurrent Neural Networks (VRNNs) in modeling dynamic systems. Through simulation results, we demonstrated the superiority of the LSTM-based black-box modeling approach over FF-ANN and VRNN methods. We used these methods for

modeling the same synchronous DC/DC buck converter. The LSTM network effectively captured both steady-state and transient responses, even under unseen operating points. This is due to the memory and generalization capability of this architecture.

Through comparing the proposed black-box method when using LSTM and a simpler dual of it called GRU, we found that GRUs often outperform LSTMs in scenarios where limited data is available or in cases where execution time is more important.

To mitigate the problem of long training times associated with deep learning models, we used transfer learning. Through tests on a simpler buck converter and a more complex HPUC converter, we demonstrated that transfer learning can be a powerful tool for considerably reducing training times.

Eventually, due to the need for real-time simulation in power systems analysis, we developed a programmatic method to export the trained deep neural networks based on the proposed black-box method into Hypersim. This involved generating equivalent models using basic Simulink blocks to make a compatible model for integration into this real-time simulation environment. The exported model accurately replicated the behavior of the original trained LSTM-based model and validated the practicality of this approach.

CONCLUSION

The rapid growth in the integration of electric vehicles (EVs), smart grids, and renewable energy generation has made power electronic converters (PECs) key components in modern power grids. However, conventional modeling and control techniques have limitations to meet the demands of complex, nonlinear converters, especially for modern applications such as EVs that are essentially large mobile electricity loads. This dissertation investigated advanced machine learning (ML) and deep reinforcement learning (DRL) techniques for black-box modeling and model-free control of PECs. These methods offer practical solutions to the challenges in modeling and control of PECs in modern power grids.

The key contributions of this work include two major advancements. First, and representing a significant contribution to the field, this thesis develops a novel black-box modeling framework using machine learning, specifically Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRUs). These methods accurately capture both small-signal as well as large-signal behaviors of PECs, allowing effective representation of converters that often lack detailed technical data in practice. These ML methods require extensive training time. Thus, transfer learning is utilized to reduce training times and make them more efficient and applicable to real-time scenarios. Second, in the control domain, a DRL-based approach is implemented to achieve model-free, adaptive control for converters, particularly the three-level Neutral Point Clamped (NPC) and the Hybrid Packed U-Cell (HPUC) topologies. Through simulation, real-time simulations, and experimental results we demonstrated that the DRL-based control outperforms traditional non-linear methods, particularly Model Predictive Control (MPC), in terms of resilience to uncertainties, large disturbances, and parameter variations.

The detailed conclusions for each chapter are as follows:

- CHAPTER 1, presents a comprehensive literature review that highlights the limitations of conventional modeling and control methods for PECs, highlighting the need for

precise mathematical models, which are often unavailable for commercial over-the-counter converters. This chapter also discussed various non-linear controls proposed in the literature to overcome the challenges associated with PECs, highlighting the shortcomings of conventional non-linear control methods. Ultimately, the advantages and previous implementations of DRL are presented to set the foundation for the model-free control proposed in CHAPTER 2 and CHAPTER 3.

- CHAPTER 2, introduces a model-free DRL-based control method for a three-level NPC converter. The DRL method showed significant improvements in stability and adaptability over MPC in various operational scenarios. The results demonstrate this method's robustness against parameter mismatches, noise, and other disturbances. Simulation results and experimental results validated the satisfactory performance of the proposed DRL method.

- CHAPTER 3, applied a similar DRL approach to the 23-level HPUC converter, and achieved satisfactory model-free control performance and voltage balancing even for such complex multi-level topology. The results obtained in this section proved the superiority of the proposed DRL method in terms of resilience to parameter mismatch, noise, and uncertainties.

- CHAPTER 4, performed a comparative analysis. The results obtained in this section proved the superiority of the proposed DRL method in terms of resilience to parameter mismatch, noise, and uncertainties compared to a model-based non-linear method like MPC.

- Eventually, CHAPTER 5 proposed a black-box modeling method for PECs using LSTM and GRU networks. By treating PECs as two-port networks, this chapter introduced a purely data-driven modeling method that can achieve high accuracy without access to the internal structure of PECs. This chapter also highlights transfer learning's role in reducing training times for converters with similar topologies.

Ultimately, to facilitate the integration of the modeling method proposed in this chapter, an algorithm was developed that converts LSTM-based methods trained in Simulink to basic blocks compatible with simulation tools like Hypersim.

In conclusion, this dissertation proposes innovative solutions based on DRL and ML to mitigate the complex challenges in PEC modeling and control, facilitating studying their impact on the grid especially when used as part of applications such as electric vehicle chargers and renewable energy sources.

Future works

While this dissertation has successfully demonstrated the potential of deep reinforcement learning (DRL) and machine learning (ML) techniques in integrating advanced modeling and control of power electronic converters (PECs), several areas remain uncharted for further exploration. The following suggestions outline key areas that we recommend for future work:

Extending to three-phase Systems and more complex topologies

Since this dissertation focused primarily on single-phase and specific multi-level converters, applying DRL-based control to three-phase systems and more complex PEC topologies, could be valuable. Investigating the DRL method's performance and adaptability in handling the additional complexity of three-phase systems could broaden its applicability in industrial and high-power applications. Likewise, implementing the black-box LSTM-based modeling method for a three-phase converter could be another valuable area to explore the applicability of this method in more challenging topologies.

Developing hybrid control methods

Combining the proposed DRL method with conventional control methods, such as Model Predictive Control (MPC) or Sliding Mode Control (SMC), may provide hybrid approaches

that leverage the strengths of both model-free adaptability and model-based precision. Further research on such hybrid control methods could optimize PEC performance, especially in scenarios where high accuracy and robust adaptability are critical.

Real-time simulation with Hardware-In-the-Loop (HIL) setup

To further validate the proposed models and control strategies, future studies should use Hardware-in-the-Loop (HIL) setups. Implementing the proposed model compatible with Hypersim could be a valuable tool for the integration of the proposed LSTM-based method in HIL and Co-simulation studies.

Impact study of EV chargers and charging stations

The proposed ML-based modeling method proposed in this dissertation provides an arbitrary level of precision. Therefore, future studies can implement such models for studying the impact of numerous non-linear PECs used in EVs and EV chargers on the power grid.

Investigation robustness and safety measures for the DRL control method

Ensuring robust control in uncertain environments is crucial for PEC applications. Future research could focus on developing safety constraints and robustness measures within DRL frameworks, enabling PEC controllers to handle sudden changes or extreme conditions without compromising system stability or reliability.

By addressing these areas, future work can build on the findings of this dissertation, refining the proposed ML-based approaches for PEC modeling and control to accelerate and facilitate the integration of modern applications in modern power grids.

Enhancing fault detection and diagnosis in AI-Based black-box models

In addition to the contributions discussed in CHAPTER 5, the proposed AI-based black-box modeling approach has the potential to serve as a digital twin for real-time monitoring and health assessment of power electronic converters. However, one of the challenges associated with this approach is accurately representing faulty components such as switches and passive elements. To address this, future work can explore integrating fault detection mechanisms within the AI model by incorporating fault data into the training process and utilizing anomaly detection techniques to identify deviations from normal operation. Additionally, hybrid methods that combine AI-based models with physics-based analytical approaches could enhance the fault representation capability, ensuring improved reliability and predictive maintenance of power electronic systems

Future work on EV charger modeling

While this dissertation focused on developing AI-based modeling techniques for power electronic converters (PECs) in general, applying these methods to EV chargers remains an important future research direction. The complex and non-linear nature of PECs in EV chargers poses significant challenges, which our black-box modeling approach is well suited to address. Future studies could focus on:

- Implementing the proposed LSTM-based black-box modeling techniques to simulate the behavior of EV chargers and assess their impact on the power grid.
- Extending the model to consider various types of EV chargers, including fast-charging technologies, to evaluate their effects on grid stability, voltage profiles, and harmonic distortion.
- Utilizing Hardware-in-the-Loop (HIL) setups to validate the models in real-time environments, enhancing the reliability and applicability of these models for practical use in EV infrastructure planning and management.

By pursuing these avenues, future research can bridge the current gap in EV charger-specific modeling and contribute to a deeper understanding of their integration into modern power systems

Hybrid implementation of DRL for control and passive component optimization

Future work could extend the application of DRL beyond control optimization to include passive component optimization in PECs. By integrating multi-objective optimization techniques, DRL agents could be trained to simultaneously optimize both control performance and hardware design, such as selecting the optimal grid inductance and DC-link capacitor size to balance efficiency, cost, and stability. This approach would allow the controller to dynamically adapt to different system configurations, reducing reliance on fixed design parameters, self-regulating design when facing component degradation and improving overall system performance in various operating conditions.

THD optimization in DRL-based control

Future research can focus on further optimizing the current THD of the DRL-based controller while maintaining its model-free advantages. This can be achieved by modifying the reward function to directly penalize high harmonic content, making certain that the DRL agent prioritizes lower THD without increasing switching losses. Additionally, integrating adaptive filter design or hybrid control strategies (e.g., combining DRL with conventional predictive techniques) could enhance waveform quality while preserving the adaptability of DRL. Another promising direction is using multi-objective optimization to jointly minimize THD and optimize switching frequency for improved overall performance in grid-connected applications.

Integrating black-box models with model-based control

A compelling path for future research is to bridge the gap between the model-free control of CHAPTER 2, CHAPTER 3, and CHAPTER 4 and the black-box modeling of CHAPTER 5. This suggests a powerful hybrid approach where the LSTM or GRU models developed in CHAPTER 5 could serve as the predictive engine within an MPC framework. This would create a "data-driven MPC" that replaces the need for a precise analytical model with a learned one. Such a controller could potentially combine the structured optimization of MPC with the robustness and adaptability of an AI-based model, offering a promising solution for power electronic converters where explicit models are unavailable or inaccurate.

Expanding the black-box modeling framework and perturbation analysis

This dissertation described the PEC model using a two-port network approach. A more precise characterization could be achieved using a quadripole model, which would require identifying multiple transfer functions to fully capture the system's dynamics and degrees of freedom. This represents a clear path for enhancing the theoretical consistency of the modeling framework.

Furthermore, the validation in CHAPTER 5 primarily focused on the model's response to changes in input reference signals. Future work should include a comprehensive perturbation analysis where disturbances are applied directly to the output voltage (V_o) and output current (I_o). This would thoroughly test the model's ability to capture the bilateral dynamics of the converter and validate its robustness under a wider range of realistic operating conditions, directly addressing the limitations of the current study.

Rigorous model validation and application in large-scale system studies

The validation in CHAPTER 5 focused on comparing the performance of LSTM and GRU architectures. For broader adoption, a more rigorous validation of the final data-driven model is essential. Future research should prioritize this by:

- Testing with Unseen Excitation Signals: The model should be validated against complex excitation signals (e.g., pseudo-random binary sequences, frequency sweeps) that were not part of the training dataset to ensure it has truly learned the system's underlying dynamics rather than just memorizing training patterns.
- Introducing Fault Conditions: The model's predictive accuracy under simulated fault conditions (e.g., switch failures, capacitor degradation beyond training limits) should be assessed to evaluate its utility for diagnostics and fault-tolerant control.

LIST OF REFERENCES

Add block to model—MATLAB add_block. (n.d.). Retrieved October 16, 2022, from https://www.mathworks.com/help/simulink/slref/add_block.html

Add line to Simulink model—MATLAB add_line. (n.d.). Retrieved October 16, 2022, from https://www.mathworks.com/help/simulink/slref/add_line.html

Agrawal, S., Panigrahi, B. K., & Tiwari, M. K. (2008). Multiobjective particle swarm algorithm with fuzzy clustering for electrical power dispatch. *IEEE Transactions on Evolutionary Computation*, *12*(5), 529–541.

Alfred, D., Czarkowski, D., & Teng, J. (2021). Model-Free Reinforcement-Learning-Based Control Methodology for Power Electronic Converters. *2021 IEEE Green Technologies Conference (GreenTech)*, 81–88. https://doi.org/10.1109/GreenTech48523.2021.00024

Arnedo, L., Boroyevich, D., Burgos, R., & Wang, F. (2008). Polytopic black-box modeling of dc-dc converters. *2008 IEEE Power Electronics Specialists Conference*, 1015–1021. https://doi.org/10.1109/PESC.2008.4592063

Babaie, M., Sharifzadeh, M., & Al-Haddad, K. (2019). Adaptive ANN based Single PI Controller for Nine-Level PUC Inverter. *2019 IEEE Electrical Power and Energy Conference (EPEC)*. https://doi.org/10.1109/EPEC47565.2019.9074823

Babaie, Mohammad, & Al-Haddad, K. (2021). ANN Based Model-Free Sliding Mode Control for Grid-Connected Compact Multilevel Converters: An Experimental Validation. *2021 IEEE 30th International Symposium on Industrial Electronics (ISIE)*, 1–6. https://doi.org/10.1109/ISIE45552.2021.9576194

Babaie, Mohammad, Mehrasa, M., Sharifzadeh, M., & Al-Haddad, K. (2022). Floating Weighting Factors ANN-MPC Based on Lyapunov Stability for Seven-Level Modified PUC Active Rectifier. *IEEE Transactions on Industrial Electronics*, *69*(1), 387–398. https://doi.org/10.1109/TIE.2021.3050375

Babaie, Mohammad, Saeidi, M., Sharifzadeh, M., Hamadi, A., Al-Haddad, K., & Chandra, A. (2020). Hybrid ANN-Linear controller for maximum PV energy harvesting in grid-tied packed e-cell inverter. *2020 International Symposium on Power Electronics, Electrical Drives, Automation and Motion (SPEEDAM)*, 871–875. IEEE.

Babaie, Mohammad, Sharifzadeh, M., Mehrasa, M., Chouinard, G., & Al-Haddad, K. (2020). Supervised learning model predictive control trained by ABC algorithm for common-mode voltage suppression in NPC inverter. *IEEE Journal of Emerging and Selected Topics in Power Electronics*, *9*(3), 3446–3456.

Babaie, Mohammad, Sharifzadeh, M., Mehrasa, M., Chouinard, G., & Al-Haddad, K. (2021). Supervised Learning Model Predictive Control Trained by ABC Algorithm for Common-Mode Voltage Suppression in NPC Inverter. *IEEE Journal of Emerging and Selected Topics in Power Electronics*, *9*(3), 3446–3456. https://doi.org/10.1109/JESTPE.2020.2984674

Bacha, S., Munteanu, I., & Bratcu, A. I. (2014). Power electronic converters modeling and control. *Advanced Textbooks in Control and Signal Processing*, *454*, 454.

Belaidi, R., Haddouche, A., & Guendouz, H. (2012). Fuzzy logic controller based three-phase shunt active power filter for compensating harmonics and reactive power under unbalanced mains voltages. *Energy Procedia*, *18*, 560–570.

Bengio, Y., Boulanger-Lewandowski, N., & Pascanu, R. (2013). Advances in optimizing recurrent networks. *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, 8624–8628. IEEE.

Bose, B. K. (2007). Neural Network Applications in Power Electronics and Motor Drives amp;mdash;An Introduction and Perspective. *IEEE Transactions on Industrial Electronics*, *54*(1), 14–33. https://doi.org/10.1109/TIE.2006.888683

Cao, D., Hu, W., Zhao, J., Zhang, G., Zhang, B., Liu, Z., … Blaabjerg, F. (2020a). Reinforcement Learning and Its Applications in Modern Power and Energy Systems: A Review. *Journal of Modern Power Systems and Clean Energy*, *8*(6), 1029–1042. https://doi.org/10.35833/MPCE.2020.000552

Cao, D., Hu, W., Zhao, J., Zhang, G., Zhang, B., Liu, Z., … Blaabjerg, F. (2020b). Reinforcement learning and its applications in modern power and energy systems: A review. *Journal of Modern Power Systems and Clean Energy*, *8*(6), 1029–1042.

Çelik, D., Ahmed, H., & Meral, M. E. (2022). Kalman filter-based super-twisting sliding mode control of shunt active power filter for electric vehicle charging station applications. *IEEE Transactions on Power Delivery*, *38*(2), 1097–1107.

Chang, Y., Wang, X., Wang, J., Wu, Y., Yang, L., Zhu, K., … Xie, X. (2023, December 28). *A Survey on Evaluation of Large Language Models*. arXiv. Retrieved from http://arxiv.org/abs/2307.03109

Chou, S.-F., Wang, X., & Blaabjerg, F. (2020). Two-Port Network Modeling and Stability Analysis of Grid-Connected Current-Controlled VSCs. *IEEE Transactions on Power Electronics*, *35*(4), 3519–3529. https://doi.org/10.1109/TPEL.2019.2934513

Chuanlin Zhang, Chengang Cui Tianxiao Yang, & Yang, J. (2023). Robustness enhancement of DRL controller for DC–DC buck convertersfusing ESO. *Journal of Control and Decision*, *0*(0), 1–10. https://doi.org/10.1080/23307706.2023.2201587

Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014a). Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. *arXiv:1412.3555 [Cs]*. Retrieved from http://arxiv.org/abs/1412.3555

Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014b). Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv Preprint arXiv:1412.3555*.

Cook, J., Oreskes, N., Doran, P. T., Anderegg, W. R. L., Verheggen, B., Maibach, E. W., … Rice, K. (2016). Consensus on consensus: A synthesis of consensus estimates on human-caused global warming. *Environmental Research Letters*, *11*(4), 048002. https://doi.org/10.1088/1748-9326/11/4/048002

Cui, C., Dong, Y., Dong, X., Zhang, C., & Ghias, A. M. Y. M. (2024). Adaptive Horizon Seeking for Generalized Predictive Control via Deep Reinforcement Learning With Application to DC/DC Converters. *IEEE Transactions on Circuits and Systems I: Regular Papers*, *71*(5), 2217–2228. https://doi.org/10.1109/TCSI.2023.3325590

Cui, C., Yan, N., Huangfu, B., Yang, T., & Zhang, C. (2022). Voltage Regulation of DC-DC Buck Converters Feeding CPLs via Deep Reinforcement Learning. *IEEE Transactions on Circuits and Systems II: Express Briefs*, *69*(3), 1777–1781. https://doi.org/10.1109/TCSII.2021.3107535

Cui, C., Yang, T., Dai, Y., Zhang, C., & Xu, Q. (2023). Implementation of Transferring Reinforcement Learning for DC–DC Buck Converter Control via Duty Ratio Mapping. *IEEE Transactions on Industrial Electronics*, *70*(6), 6141–6150. https://doi.org/10.1109/TIE.2022.3192676

Cvetkovic, I., Boroyevich, D., Mattavelli, P., Lee, F. C., & Dong, D. (2011). Un-terminated, low-frequency terminal behavioral model of dc-dc converters. *2011 Twenty-Sixth Annual IEEE Applied Power Electronics Conference and Exposition (APEC)*, 1873–1880. https://doi.org/10.1109/APEC.2011.5744851

Darbali-Zamora, R., & Ortiz-Rivera, E. I. (2019). An Overview into the Effects of Nonlinear Phenomena in Power Electronic Converters for Photovoltaic Applications. *2019 IEEE 46th Photovoltaic Specialists Conference (PVSC)*, 2908–2915. https://doi.org/10.1109/PVSC40753.2019.8980933

Du, G., Zou, Y., Zhang, X., Liu, T., Wu, J., & He, D. (2020). Deep reinforcement learning based energy management for a hybrid electric vehicle. *Energy*, *201*, 117591.

Dubey, A., Santoso, S., & Cloud, M. P. (2013). Average-Value Model of Electric Vehicle Chargers. *IEEE Transactions on Smart Grid*, *4*(3), 1549–1557. https://doi.org/10.1109/TSG.2013.2258692

Dutta, D., & Upreti, S. R. (2022). A survey and comparative evaluation of actor-critic methods in process control. *The Canadian Journal of Chemical Engineering*, *100*(9), 2028–2056. https://doi.org/10.1002/cjce.24508

Flagship Real-Time Digital Simulator │ Simulation tools │ OP5707XG. (n.d.). Retrieved February 5, 2024, from OPAL-RT website: https://www.opal-rt.com/simulator-platform-op5707/

Francés, A., Asensi, R., García, O., Prieto, R., & Uceda, J. (2016). The performance of polytopic models in smart DC microgrids. *2016 IEEE Energy Conversion Congress and Exposition (ECCE)*, 1–8. https://doi.org/10.1109/ECCE.2016.7855507

Frances, A., Asensi, R., García, Ó., Prieto, R., & Uceda, J. (2017). Modeling electronic power converters in smart DC microgrids—An overview. *IEEE Transactions on Smart Grid*, *9*(6), 6274–6287.

Francés, Airán, Asensi, R., García, Ó., Prieto, R., & Uceda, J. (2018). Modeling Electronic Power Converters in Smart DC Microgrids—An Overview. *IEEE Transactions on Smart Grid*, *9*(6), 6274–6287. https://doi.org/10.1109/TSG.2017.2707345

Francés, Airán, Asensi, R., & Uceda, J. (2019). Blackbox Polytopic Model With Dynamic Weighting Functions for DC-DC Converters. *IEEE Access*, *7*, 160263–160273. https://doi.org/10.1109/ACCESS.2019.2950983

Gao, Y., & Yu, N. (2021). Deep reinforcement learning in power distribution systems: Overview, challenges, and opportunities. *2021 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT)*, 1–5. IEEE. Retrieved from https://ieeexplore.ieee.org/abstract/document/9372283/?casa_token=ItTTo2lYmXMAAAAA:tp7ehYUUzjRTkoZ8EamI3F4hav_Ml4mBdubPSNSqGghkpG-zSBr5pPwscH7rcOPIJwwRqUIZiQ0

Gheisarnejad, M., Farsizadeh, H., & Khooban, M. H. (2021). A Novel Nonlinear Deep Reinforcement Learning Controller for DC–DC Power Buck Converters. *IEEE Transactions on Industrial Electronics*, *68*(8), 6849–6858. https://doi.org/10.1109/TIE.2020.3005071

Graves, A., Mohamed, A., & Hinton, G. (2013). Speech recognition with deep recurrent neural networks. *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, 6645–6649. IEEE.

Gros, I.-C., Lü, X., Oprea, C., Lu, T., & Pintilie, L. (2023). Artificial intelligence (AI)-based optimization of power electronic converters for improved power system stability and performance. *2023 IEEE 14th International Symposium on Diagnostics for Electrical Machines, Power Electronics and Drives (SDEMPED)*, 204–210. https://doi.org/10.1109/SDEMPED54949.2023.10271490

Guarderas, G., Francés, A., Asensi, R., & Uceda, J. (2017). Large-signal black-box behavioral modeling of grid-supporting power converters in AC microgrids. *2017 IEEE 6th International Conference on Renewable Energy Research and Applications (ICRERA)*, 153–158. IEEE.

Guarderas, G., Frances, A., Ramirez, D., Asensi, R., & Uceda, J. (2019). Blackbox Large-Signal Modeling of Grid-Connected DC-AC Electronic Power Converters. *Energies*, *12*(6), 989. https://doi.org/10.3390/en12060989

Hajihosseini, M., Andalibi, M., Gheisarnejad, M., Farsizadeh, H., & Khooban, M.-H. (2020). DC/DC Power Converter Control-Based Deep Machine Learning Techniques: Real-Time Implementation. *IEEE Transactions on Power Electronics*, *35*(10), 9971–9977. https://doi.org/10.1109/TPEL.2020.2977765

Hochreiter, S., & Schmidhuber, J. (1997a). Long Short-Term Memory. *Neural Computation*, *9*(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

Hochreiter, S., & Schmidhuber, J. (1997b). Long short-term memory. *Neural Computation*, *9*(8), 1735–1780.

Hu, J., Shan, Y., Guerrero, J. M., Ioinovici, A., Chan, K. W., & Rodriguez, J. (2021). Model predictive control of microgrids–An overview. *Renewable and Sustainable Energy Reviews*, *136*, 110422.

Hu, W., Shi, D., & Borst, T. (2020). Guest Editorial: Applications of Artificial Intelligence in Modern Power Systems: Challenges and Solutions. *Journal of Modern Power Systems and Clean Energy*, *8*(6), 1–2.

Hu, X., Liu, T., Qi, X., & Barth, M. (2019). Reinforcement learning for hybrid and plug-in hybrid electric vehicle energy management: Recent advances and prospects. *IEEE Industrial Electronics Magazine*, *13*(3), 16–25.

Huang, F. (2008). A Particle Swarm Optimized Fuzzy Neural Network for Credit Risk Evaluation. *2008 Second International Conference on Genetic and Evolutionary Computing*, 153–157. https://doi.org/10.1109/WGEC.2008.25

Ibarz, J., Tan, J., Finn, C., Kalakrishnan, M., Pastor, P., & Levine, S. (2021). How to train your robot with deep reinforcement learning: Lessons we have learned. *The International Journal of Robotics Research*, *40*(4–5), 698–721.

IEA (2019) Report, "Global EV Outlook 2019", IEA, Paris. Available at: Www.iea.org/publications/reports/globalevoutlook2019/. (n.d.).

Impram, S., Nese, S. V., & Oral, B. (2020). Challenges of renewable energy penetration on power system flexibility: A survey. *Energy Strategy Reviews*, *31*, 100539.

Iyer, V. M., Gulur, S., & Bhattacharya, S. (2017). Small-signal modeling and stability analysis of a bidirectional electric vehicle charger. *2017 IEEE 6th International Conference on Renewable Energy Research and Applications (ICRERA)*, 1030–1035. https://doi.org/10.1109/ICRERA.2017.8191214

Jang, B., Kim, M., Harerimana, G., & Kim, J. W. (2019). Q-learning algorithms: A comprehensive classification and applications. *IEEE Access*, *7*, 133653–133667.

Jiang, S., Zeng, Y., Zhu, Y., Pou, J., & Konstantinou, G. (2023). Stability-Oriented Multiobjective Control Design for Power Converters Assisted by Deep Reinforcement Learning. *IEEE Transactions on Power Electronics*, *38*(10), 12394–12400. https://doi.org/10.1109/TPEL.2023.3299979

Jianping Xu, & Lee, C. Q. (1997). Generalized state-space averaging approach for a class of periodically switched networks. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, *44*(11), 1078–1081. https://doi.org/10.1109/81.641772

Kanaan, H., Al-Haddad, K., & Fnaiech, F. (2005). Modelling and control of three-phase/switch/level fixed-frequency PWM rectifier: State-space averaged model. *IEE Proceedings - Electric Power Applications*, *152*(3), 551–557. https://doi.org/10.1049/ip-epa:20041042

Kanaan, H. Y., Hayek, A., & Al-Haddad, K. (2007). Small-Signal Average Modeling, Simulation and Carrier-Based Linear Control of a Three-Phase Four-Leg Shunt Active Power Filter. *2007 IEEE International Electric Machines Drives Conference*, *1*, 601–607. https://doi.org/10.1109/IEMDC.2007.382735

Kicsiny, R. (2017). Grey-box model for pipe temperature based on linear regression. *International Journal of Heat and Mass Transfer*, *107*, 13–20. https://doi.org/10.1016/j.ijheatmasstransfer.2016.11.033

Komurcugil, H., Biricik, S., & Babaei, E. (2019). Super Twisting Algorithm Based Sliding Mode Control Method for Single-Phase Dynamic Voltage Restorers. *2019 2nd International Conference on Smart Grid and Renewable Energy (SGRE)*, 1–6. https://doi.org/10.1109/SGRE46976.2019.9020687

Komurcugil, H., Biricik, S., Bayhan, S., & Zhang, Z. (2020). Sliding mode control: Overview of its applications in power converters. *IEEE Industrial Electronics Magazine*, *15*(1), 40–49.

Krishna, R. A., & Suresh, L. P. (2016). A brief review on multi level inverter topologies. *2016 International Conference on Circuit, Power and Computing Technologies (ICCPCT)*, 1–6. IEEE.

Krishnamoorthy, H. S., & Aayer, T. N. (2019). Machine learning based modeling of power electronic converters. *2019 IEEE Energy Conversion Congress and Exposition (ECCE)*, 666–672. IEEE.

Krishnamoorthy, H. S., & Narayanan Aayer, T. (2019). Machine Learning based Modeling of Power Electronic Converters. *2019 IEEE Energy Conversion Congress and Exposition (ECCE)*, 666–672. https://doi.org/10.1109/ECCE.2019.8912608

Krommydas, K. F., & Alexandridis, A. T. (2020). Nonlinear Analysis Methods Applied on Grid-Connected Photovoltaic Systems Driven by Power Electronic Converters. *IEEE Journal of Emerging and Selected Topics in Power Electronics*, *8*(4), 3293–3306. https://doi.org/10.1109/JESTPE.2020.2992969

Kůrková, V., Manolopoulos, Y., Hammer, B., Iliadis, L., & Maglogiannis, I. (Eds.). (2018). *Artificial Neural Networks and Machine Learning – ICANN 2018: 27th International Conference on Artificial Neural Networks, Rhodes, Greece, October 4-7, 2018, Proceedings, Part III*. Cham: Springer International Publishing. https://doi.org/10.1007/978-3-030-01424-7

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436–444.

Lee, S., & Choi, D.-H. (2020). Federated reinforcement learning for energy management of multiple smart homes with distributed energy resources. *IEEE Transactions on Industrial Informatics*, *18*(1), 488–497.

Lendzen, R. (2010). Global warming: The origin and nature of the alleged scientific consensus. *Problems of Sustainable Development*, *5*(2), 13–28.

Li, X., Ruan, X., Jin, Q., Sha, M., & Tse, C. K. (2018). Approximate Discrete-Time Modeling of DC–DC Converters With Consideration of the Effects of Pulse Width Modulation. *IEEE Transactions on Power Electronics*, *33*(8), 7071–7082. https://doi.org/10.1109/TPEL.2017.2752419

Li, Yuanzheng, Yu, C., Shahidehpour, M., Yang, T., Zeng, Z., & Chai, T. (2023). Deep Reinforcement Learning for Smart Grid Operations: Algorithms, Applications, and Prospects. *Proceedings of the IEEE*. Retrieved from https://ieeexplore.ieee.org/abstract/document/10241311/?casa_token=WJLqFR4-ODsAAAAA:mE_UENnEHsnIOUag0m1lInmvfDNievLyjNAMk9nW8gW15GwWhrYIB7vVhbrRDBfhV0kXNjSBFRQ

Li, Yuxi. (2018, November 25). *Deep Reinforcement Learning: An Overview*. arXiv. Retrieved from http://arxiv.org/abs/1701.07274

Liao, Y., Li, Y., Chen, M., Nordström, L., Wang, X., Mittal, P., & Poor, H. V. (2023). Neural Network Design for Impedance Modeling of Power Electronic Systems Based on

Latent Features. *IEEE Transactions on Neural Networks and Learning Systems*, 1–13. https://doi.org/10.1109/TNNLS.2023.3235806

Liu, C., Bai, H., Zhuo, S., Zhang, X., Ma, R., & Gao, F. (2020). Real-Time Simulation of Power Electronic Systems Based on Predictive Behavior. *IEEE Transactions on Industrial Electronics*, *67*(9), 8044–8053. https://doi.org/10.1109/TIE.2019.2941135

Liu, H., & Li, S. (2011). Speed control for PMSM servo system using predictive functional control and extended state observer. *IEEE Transactions on Industrial Electronics*, *59*(2), 1171–1183.

Liu, X., Qiu, L., Fang, Y., Wang, K., Li, Y., & Rodríguez, J. (2024). Predictive Control of Voltage Source Inverter: An Online Reinforcement Learning Solution. *IEEE Transactions on Industrial Electronics*, *71*(7), 6591–6600. https://doi.org/10.1109/TIE.2023.3303626

Liu, Y., Yang, J., Chen, L., Guo, T., & Jiang, Y. (2020). Overview of Reinforcement Learning Based on Value and Policy. *2020 Chinese Control And Decision Conference (CCDC)*, 598–603. https://doi.org/10.1109/CCDC49329.2020.9164615

Mahmud, M. A., Roy, T. K., Saha, S., Haque, M. E., & Pota, H. R. (2019). Robust Nonlinear Adaptive Feedback Linearizing Decentralized Controller Design for Islanded DC Microgrids. *IEEE Transactions on Industry Applications*, *55*(5), 5343–5352. https://doi.org/10.1109/TIA.2019.2921028

Mandal, S., & Mishra, D. (2018). Robust control of buck converter using H-infinity control algorithm. *2018 IEEE Applied Signal Processing Conference (ASPCON)*, 163–167. IEEE. Retrieved from https://ieeexplore.ieee.org/abstract/document/8748623/?casa_token=LauBG6PyNz8AAAAA:a7P-VM_TQ93M1DeIQj6h3ybmCHYwV9YdeETqNCr0dVISz3s5cHkOfL31dBfXzX_1WCaL2MnA_dM

Maswood, A. I., & Wei, S. (2005). Genetic-algorithm-based solution in PWM converter switching. *IEE Proceedings - Electric Power Applications*, *152*(3), 473–478. https://doi.org/10.1049/ip-epa:20040803

Metri, J. I., Vahedi, H., Kanaan, H. Y., & Al-Haddad, K. (2016). Real-Time Implementation of Model-Predictive Control on Seven-Level Packed U-Cell Inverter. *IEEE Transactions on Industrial Electronics*, *63*(7), 4180–4186. https://doi.org/10.1109/TIE.2016.2542133

Middlebrook, R. D., & Cuk, S. (1976). A general unified approach to modelling switching-converter power stages. *1976 IEEE Power Electronics Specialists Conference*, 18–34. https://doi.org/10.1109/PESC.1976.7072895

Mishra, D., & Mandal, S. (2020). Voltage regulation of DC-DC boost converter using H-infinity controller. *2020 International Symposium on Devices, Circuits and Systems (ISDCS)*, 1–5. IEEE. Retrieved from https://ieeexplore.ieee.org/abstract/document/9263019/?casa_token=ZzqOe2EITScAAAAA:IU57emwM_DUs_0YnK2fVDpqOItE2a3ZqEFftq0CyHqtUHjQKRYC1XYZh8lapynHQqmzXSU-EN5o

Mittal, N., Singh, B., Singh, S. P., Dixit, R., & Kumar, D. (2012). Multilevel inverters: A literature survey on topologies and control strategies. *2012 2nd International Conference on Power, Control and Embedded Systems*, 1–11. IEEE.

Mukhopadhyay, R., Bandyopadhyay, S., Sutradhar, A., & Chattopadhyay, P. (2019). Performance Analysis of Deep Q Networks and Advantage Actor Critic Algorithms in Designing Reinforcement Learning-based Self-tuning PID Controllers. *2019 IEEE Bombay Section Signature Conference (IBSSC)*, 1–6. https://doi.org/10.1109/IBSSC47189.2019.8973068

Naziris, A., Asensi, R., & Uceda, J. (2018). Black Box Modelling of a Bidirectional Battery Charger for Electric Vehicles. *2018 7th International Conference on Renewable Energy Research and Applications (ICRERA)*, 469–473. https://doi.org/10.1109/ICRERA.2018.8566794

Naziris, A., Guarderas, G., Francés, A., Asensi, R., & Uceda, J. (2019). Large-Signal Black-Box Modelling of Bidirectional Battery Charger for Electric Vehicles. *2019 IEEE Applied Power Electronics Conference and Exposition (APEC)*, 3195–3198. https://doi.org/10.1109/APEC.2019.8721930

Nguyen, H., & La, H. (2019). Review of deep reinforcement learning for robot manipulation. *2019 Third IEEE International Conference on Robotic Computing (IRC)*, 590–595. IEEE.

Niculae, D., Iordache, M., Stanculescu, M., Bobaru, M. L., & Deleanu, S. (2019). A Review of Electric Vehicles Charging Technologies Stationary and Dynamic. *2019 11th International Symposium on Advanced Topics in Electrical Engineering (ATEE)*, 1–4. https://doi.org/10.1109/ATEE.2019.8724943

Oghenewvogaga, O., Adetokun, B. B., Gamiya, B. G., & Nagode, A. B. (2022). The Role of Power Electronics in Renewable Energy System. *2022 IEEE Nigeria 4th International Conference on Disruptive Technologies for Sustainable Development (NIGERCON)*, 1–5. https://doi.org/10.1109/NIGERCON54645.2022.9803111

Oliver, J. A., Prieto, R., Cobos, J. A., Garcia, O., & Alou, P. (2009). Hybrid wiener-hammerstein structure for grey-box modeling of dc-dc converters. *2009 Twenty-Fourth Annual IEEE Applied Power Electronics Conference and Exposition*, 280–285. IEEE.

Ould-Bachir, T., Blanchette, H. F., & Al-Haddad, K. (2015). A Network Tearing Technique for FPGA-Based Real-Time Simulation of Power Converters. *IEEE Transactions on Industrial Electronics*, *62*(6), 3409–3418. https://doi.org/10.1109/TIE.2014.2365752

Padhee, S., Pati, U. C., & Mahapatra, K. (2016). Modelling switched mode DC-DC converter using system identification techniques: A review. *2016 IEEE Students' Conference on Electrical, Electronics and Computer Science (SCEECS)*, 1–6. https://doi.org/10.1109/SCEECS.2016.7509303

Pan, S. J., & Yang, Q. (2010). A survey on transfer learning. IEEE Transactions on knowledge and data engineering. *22 (10): 1345*, *1359*.

Pascanu, R., Mikolov, T., & Bengio, Y. (2013). On the difficulty of training Recurrent Neural Networks. *arXiv:1211.5063 [Cs]*. Retrieved from http://arxiv.org/abs/1211.5063

Patra, N. K., Garg, M. M., Panda, A. K., & Shukla, R. R. (2021). H-infinity Robust Control of DC-DC Converter. In S. S. Dash, B. K. Panigrahi, & S. Das (Eds.), *Sixth International Conference on Intelligent Computing and Applications* (pp. 257–267). Singapore: Springer Singapore. https://doi.org/10.1007/978-981-16-1335-7_23

Power system simulation │ Power system Analysis │ HYPERSIM. (n.d.). Retrieved from OPAL-RT website: https://www.opal-rt.com/systems-hypersim/

Predict responses using a trained recurrent neural network—Simulink. (n.d.). Retrieved October 16, 2022, from https://www.mathworks.com/help/deeplearning/ref/statefulpredict.html

Qashqai, P., Al-Haddad, K., & Zgheib, R. (2020). Modeling Power Electronic Converters Using A Method Based on Long-Short Term Memory (LSTM) Networks. *IECON 2020 The 46th Annual Conference of the IEEE Industrial Electronics Society*, 4697–4702. https://doi.org/10.1109/IECON43393.2020.9255375

Qashqai, P., Sheikholeslami, A., Vahedi, H., & Al-Haddad, K. (2015a). A Review on Multilevel Converter Topologies for Electric Transportation Applications. *2015 IEEE Vehicle Power and Propulsion Conference (VPPC)*, 1–6. https://doi.org/10.1109/VPPC.2015.7352882

Qashqai, P., Sheikholeslami, A., Vahedi, H., & Al-Haddad, K. (2015b). A Review on Multilevel Converter Topologies for Electric Transportation Applications. *2015 IEEE Vehicle Power and Propulsion Conference (VPPC)*, 1–6. https://doi.org/10.1109/VPPC.2015.7352882

Qashqai, Pouria, Al-Haddad, K., & Zgheib, R. (2020a). A new model-free space vector modulation technique for multilevel inverters based on deep reinforcement learning. *IECON 2020 The 46th Annual Conference of the IEEE Industrial Electronics Society*, 2407–2411. IEEE.

Qashqai, Pouria, Al-Haddad, K., & Zgheib, R. (2020b). Modeling power electronic converters using a method based on long-short term memory (LSTM) networks. *IECON 2020 The 46th Annual Conference of the IEEE Industrial Electronics Society*, 4697–4702. IEEE.

Qashqai, Pouria, Al-Haddad, K., & Zgheib, R. (2020c). Modeling power electronic converters using a method based on long-short term memory (LSTM) networks. *IECON 2020 The 46th Annual Conference of the IEEE Industrial Electronics Society*, 4697–4702. IEEE.

Qashqai, Pouria, Al-Haddad, K., & Zgheib, R. (2022). Deep neural network-based black-box modeling of power electronic converters using transfer learning. *2022 IEEE Energy Conversion Congress and Exposition (ECCE)*, 1–6. IEEE.

Qashqai, Pouria, Babaie, M., Zgheib, R., & Al-Haddad, K. (2023). A Model-Free Switching and Control Method for Three-Level Neutral Point Clamped Converter Using Deep Reinforcement Learning. *IEEE Access*, *11*, 105394–105409. https://doi.org/10.1109/ACCESS.2023.3318264

Qashqai, Pouria, Vahedi, H., & Al-Haddad, K. (2019). Applications of artifical intelligence in power electronics. *2019 IEEE 28th International Symposium on Industrial Electronics (ISIE)*, 764–769. IEEE.

Qashqai, Pouria, Zgheib, R., & Al-Haddad, K. (2021). GRU and LSTM comparison for black-box modeling of power electronic converters. *IECON 2021–47th Annual Conference of the IEEE Industrial Electronics Society*, 1–5. IEEE.

Qashqai, Pouria, Zgheib, R., & Al-Haddad, K. (2022). A Programmatical Method for Real-time Simulation of Black-box LSTM-based Models of Power Electronic Converters in Hypersim. *2022 IEEE 1st Industrial Electronics Society Annual On-Line Conference (ONCON)*, 1–5. IEEE.

Qiu, Y., Liu, H., & Chen, X. (2010). Digital Average Current-Mode Control of PWM DC–DC Converters Without Current Sensors. *IEEE Transactions on Industrial Electronics*, *57*(5), 1670–1677. https://doi.org/10.1109/TIE.2009.2032130

Reinforcement Learning Agents—MATLAB & Simulink. (n.d.). Retrieved March 15, 2023, from https://www.mathworks.com/help/reinforcement-learning/ug/create-agents-for-reinforcement-learning.html

Riba, J.-R., Moreno-Eguilaz, M., Bogarra, S., & Garcia, A. (2018). Parameter Identification of DC-DC Converters under Steady-State and Transient Conditions Based on White-Box Models. *Electronics*, *7*(12), 393. https://doi.org/10.3390/electronics7120393

Rodriguez, J., Garcia, C., Mora, A., Davari, S. A., Rodas, J., Valencia, D. F., … Tarisciotti, L. (2021). Latest advances of model predictive control in electrical drives—Part II: Applications and benchmarking with classical control methods. *IEEE Transactions on Power Electronics*, *37*(5), 5047–5061.

Rodriguez, J., Garcia, C., Mora, A., Flores-Bahamonde, F., Acuna, P., Novak, M., … Zhang, Z. (2021). Latest advances of model predictive control in electrical drives—Part I: Basic concepts and advanced strategies. *IEEE Transactions on Power Electronics*, *37*(4), 3927–3942.

Rojas-Dueñas, G., Riba, J.-R., Kahalerras, K., Moreno-Eguilaz, M., Kadechkar, A., & Gomez-Pau, A. (2020). Black-Box Modelling of a DC-DC Buck Converter Based on a Recurrent Neural Network. *2020 IEEE International Conference on Industrial Technology (ICIT)*, 456–461. https://doi.org/10.1109/ICIT45562.2020.9067098

Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural Networks*, *61*, 85–117. https://doi.org/10.1016/j.neunet.2014.09.003

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (2017, August 28). *Proximal Policy Optimization Algorithms*. arXiv. https://doi.org/10.48550/arXiv.1707.06347

Scopus—Analyze search results. (2024, September 27). Retrieved September 27, 2024, from https://www.scopus.com/term/analyzer.uri?sort=plf-f&src=s&sid=a41b66ca08de14bc2452ac5ea57ffc7f&sot=a&sdt=a&sl=125&s=ALL%28%22Artificial+Intelligence%22+AND++%22Power+Converters%22+OR+%22Power+electronic+converters%22%29+AND+PUBYEAR+%3e+2017+AND+PUBYEAR+%3c+2025&origin=resultslist&count=10&analyzeResults=Analyze+results

Sebaaly, F., Vahedi, H., Kanaan, H. Y., Moubayed, N., & Al-Haddad, K. (2016). Design and Implementation of Space Vector Modulation-Based Sliding Mode Control for Grid-Connected 3L-NPC Inverter. *IEEE Transactions on Industrial Electronics*, *63*(12), 7854–7863. https://doi.org/10.1109/TIE.2016.2563381

Shao, K., Tang, Z., Zhu, Y., Li, N., & Zhao, D. (2019, December 26). *A Survey of Deep Reinforcement Learning in Video Games*. arXiv. Retrieved from http://arxiv.org/abs/1912.10944

Sheikhi, A., Rayati, M., & Ranjbar, A. M. (2016). Dynamic load management for a residential customer; reinforcement learning approach. *Sustainable Cities and Society*, *24*, 42–51.

Sherstinsky, A. (2020). Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) network. *Physica D: Nonlinear Phenomena*, *404*, 132306. https://doi.org/10.1016/j.physd.2019.132306

Shin, H. (2022). *README* [Matlab]. Retrieved from https://github.com/HyunkiShin/LSTM2Simulink (Original work published 2018)

Shin, H. (2023). *HyunkiShin/LSTM2Simulink* [Matlab]. Retrieved from https://github.com/HyunkiShin/LSTM2Simulink (Original work published 2018)

Singh, B., Gairola, S., Singh, B. N., Chandra, A., & Al-Haddad, K. (2008). Multipulse AC–DC converters for improving power quality: A review. *IEEE Transactions on Power Electronics*, *23*(1), 260–281.

Siwakoti, Y. P., Forouzesh, M., & Pham, N. H. (2018). Power electronics converters—An overview. *Control of Power Electronic Converters and Systems*, 3–29.

Sorto-Ventura, K.-R., Abarzadeh, M., Al-Haddad, K., & Dessaint, L. A. (2020). 23-level single DC source hybrid PUC (H-PUC) converter topology with reduced number of components: Real-time implementation with model predictive control. *IEEE Open Journal of the Industrial Electronics Society*, *1*, 127–137.

Tan, H. H., & Lim, K. H. (2019). Vanishing Gradient Mitigation with Deep Learning Neural Network Optimization. *2019 7th International Conference on Smart Computing & Communications (ICSCC)*, 1–4. IEEE.

The Coal Cost Crossover: Economic Viability Of Existing Coal Compared To New Local Wind And Solar Resources. (n.d.). Retrieved from Energy Innovation: Policy and Technology website: https://energyinnovation.org/publication/the-coal-cost-crossover/

Tian, F., Cobaleda, D. B., & Martinez, W. (2022). Deep Reinforcement Learning for DC-DC Converter Parameters Optimization. *2022 IEEE 31st International Symposium on Industrial Electronics (ISIE)*, 325–330. https://doi.org/10.1109/ISIE51582.2022.9831660

V., J. (2017). GENETIC ALGORITHM BASED SOLUTION IN PWM CONVERTER SWITCHING FOR VOLTAGE SOURCE INVERTER FEEDING AN INDUCTION MOTOR DRIVE. *ASEAN Journal on Science and Technology for Development*, *27*(2), 45–60. https://doi.org/10.29037/ajstd.251

Vahedi, H., Kanaan, H. Y., & Al-Haddad, K. (2015). PUC converter review: Topology, control and applications. *IECON 2015-41st Annual Conference of the IEEE Industrial Electronics Society*, 004334–004339. IEEE.

Vahedi, H., Labbé, P.-A., & Al-Haddad, K. (2015). Sensor-less five-level packed U-cell (PUC5) inverter operating in stand-alone and grid-connected modes. *IEEE Transactions on Industrial Informatics*, *12*(1), 361–370.

Vražić, M., Vuljaj, D., Pavasović, A., & Pauković, H. (2014). Study of a vehicle conversion from internal combustion engine to electric drive. *2014 IEEE International Energy Conference (ENERGYCON)*, 1544–1548. https://doi.org/10.1109/ENERGYCON.2014.6850628

Wan, L., Beshir, A. H., Wu, X., Liu, X., Grassi, F., Spadacini, G., … Chiumeo, R. (2021). Black-Box Modelling of Low-Switching-Frequency Power Inverters for EMC

Analyses in Renewable Power Systems. *Energies*, *14*(12), 3413. https://doi.org/10.3390/en14123413

Wan, Yanni, Qin, J., Ma, Q., Fu, W., & Wang, S. (2022). Multi-agent DRL-based data-driven approach for PEVs charging/discharging scheduling in smart grid. *Journal of the Franklin Institute*, *359*(4), 1747–1767.

Wan, Yihao, Dragicevic, T., Mijatovic, N., Li, C., & Rodriguez, J. (2021). Reinforcement Learning Based Weighting Factor Design of Model Predictive Control for Power Electronic Converters. *2021 IEEE International Conference on Predictive Control of Electrical Drives and Power Electronics (PRECEDE)*, 738–743. https://doi.org/10.1109/PRECEDE51386.2021.9680964

Wang, H., Lei, Z., Zhang, X., Zhou, B., & Peng, J. (2019). A review of deep learning for renewable energy forecasting. *Energy Conversion and Management*, *198*, 111799.

Wang, J., Yang, R., & Yao, Z. (2022). Efficiency Optimization Design of Three-Level Active Neutral Point Clamped Inverter Based on Deep Reinforcement Learning. *2022 IEEE 6th Conference on Energy Internet and Energy System Integration (EI2)*, 605–610. https://doi.org/10.1109/EI256261.2022.10117037

Wang, M., & Wang, Y. (2020). Fuzzy Neural-Network-based Output Tracking Control for Nonlinear Systems with Unknown Dynamics. *2020 Chinese Automation Congress (CAC)*, 5124–5129. https://doi.org/10.1109/CAC51589.2020.9327892

Wang, X., Wang, S., Liang, X., Zhao, D., Huang, J., Xu, X., … Miao, Q. (2022). Deep Reinforcement Learning: A Survey. *IEEE Transactions on Neural Networks and Learning Systems*, 1–15. https://doi.org/10.1109/TNNLS.2022.3207346

Weiss, K., Khoshgoftaar, T. M., & Wang, D. (2016). A survey of transfer learning. *Journal of Big Data*, *3*(1), 9. https://doi.org/10.1186/s40537-016-0043-6

When Fossil Fuels Run Out, What Then? (n.d.). Retrieved from MAHB website: https://mahb.stanford.edu/library-item/fossil-fuels-run/

Wu, H. H., Gilchrist, A., Sealy, K., Israelsen, P., & Muhs, J. (2011). A review on inductive charging for electric vehicles. *2011 IEEE International Electric Machines Drives Conference (IEMDC)*, 143–147. https://doi.org/10.1109/IEMDC.2011.5994820

Xu, Q., Vafamand, N., Chen, L., Dragičević, T., Xie, L., & Blaabjerg, F. (2020). Review on advanced control technologies for bidirectional DC/DC converters in DC microgrids. *IEEE Journal of Emerging and Selected Topics in Power Electronics*, *9*(2), 1205–1221.

Yang, Tianxiao, Cui, C., & Zhang, C. (2022). On the Robustness Enhancement of DRL Controller for DC-DC Converters in Practical Applications. *2022 IEEE 17th International Conference on Control & Automation (ICCA)*, 225–230. https://doi.org/10.1109/ICCA54724.2022.9831887

Yang, Ting, Zhao, L., Li, W., & Zomaya, A. Y. (2021). Dynamic energy dispatch strategy for integrated energy system based on improved deep reinforcement learning. *Energy*, *235*, 121377.

Zeng, Y., Pou, J., Sun, C., Li, X., Liang, G., Xia, Y., … Gupta, A. K. (2024). Deep Reinforcement Learning-Enabled Distributed Uniform Control for a DC Solid State Transformer in DC Microgrid. *IEEE Transactions on Industrial Electronics*, *71*(6), 5818–5829. https://doi.org/10.1109/TIE.2023.3294584

Zhang, G., Li, Z., Zhang, B., & Halang, W. A. (2018). Power electronics converters: Past, present and future. *Renewable and Sustainable Energy Reviews*, *81*, 2028–2044.

Zhang, Y., & Wang, T. (2022). Applying Value-Based Deep Reinforcement Learning on KPI Time Series Anomaly Detection. *2022 IEEE 15th International Conference on Cloud Computing (CLOUD)*, 197–202. https://doi.org/10.1109/CLOUD55607.2022.00039

Zhang, Z., Zhang, D., & Qiu, R. C. (2019). Deep reinforcement learning for power system applications: An overview. *CSEE Journal of Power and Energy Systems*, *6*(1), 213–225.

Zhao, D., Wang, H., Shao, K., & Zhu, Y. (2016). Deep reinforcement learning with experience replay based on SARSA. *2016 IEEE Symposium Series on Computational Intelligence (SSCI)*, 1–6. https://doi.org/10.1109/SSCI.2016.7849837

Zhao, S., Peng, Y., Zhang, Y., & Wang, H. (2022). Parameter Estimation of Power Electronic Converters With Physics-Informed Machine Learning. *IEEE Transactions on Power Electronics*, *37*(10), 11567–11578. https://doi.org/10.1109/TPEL.2022.3176468

Zhao, W., Queralta, J. P., & Westerlund, T. (2020). Sim-to-real transfer in deep reinforcement learning for robotics: A survey. *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*, 737–744. IEEE. Retrieved from https://ieeexplore.ieee.org/abstract/document/9308468/