

ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING FOR CONTROL AND
OPERATION OF ELECTRIC VEHICLES AND MACHINE DRIVES

by

WEIZHEN DONG

SHUHUI LI, COMMITTEE CHAIR

BHARAT BALASUBRAMANIAN

FEI HU

TIM A. HASKEW

YU GAN

A DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy
in the Department of Electrical and Computer Engineering
in the Graduate School of
The University of Alabama

TUSCALOOSA, ALABAMA

2021

ABSTRACT

Motor drive and charging system with batteries are two major parts in an electric vehicle (EV) powertrain system. This dissertation investigates the artificial intelligence-based control and operation of EV machine drives and the charging systems.

There are several major challenges related the EV motor drive control such as machine parameter variations, magnetic saturations, accurate torque control, and optimal and efficient operation considering copper loss and iron loss. Regarding the charging and discharging control with DC/DC converters, the stable and robust voltage regulation under disturbances is required. The issues of how to smoothly handle the current/voltage constraints and the power limit still remain.

This dissertation presents a novel machine learning strategy based on a neural network (NN) to achieve MTPA, flux-weakening, and MTPV for the most efficient IPM torque control over its full speed operating range. The NN is trained offline by using the LMBP (Levenberg-Marquardt backpropagation) algorithm, which avoids the disadvantages associated with the online NN training. A special technique is developed to generate NN training data, that is particularly suitable and favorable, to develop a high-performance NN-based IPM torque control system, and the impact of variable motor parameters is embedded into the NN system development and training.

IPM machine modeling and parameter estimation are important for the controller design of high-efficient and high-performance motor drives. The accuracy of the magnetic modeling is a

challenge due to the magnetic saturation, cross saturation, iron loss, and temperature variations. The proposed ANN-based modeling method can capture the nonlinear areas of the model and generate accurate dq-axis flux linkages with saturation and iron loss considered.

For the vehicle to grid (V2G) and vehicle to home (V2H) applications, the battery not only can be charged but also can provide power back to the load and systems through DC/DC converters. The ANN controller presented in this dissertation has a strong ability to track rapidly changing reference commands, maintain stable output voltage for a variable load, and manage maximum duty-ratio and current constraints properly. The presented control algorithm also has the ability of power sharing based on DG capabilities for DC microgrid applications.

LIST OF ABBREVIATIONS AND SYMBOLS

PMSM	Permanent Magnet Synchronous Machine
R_s	Stator phase resistance
$L_{sd,sq}$	Stator d- and q-axis inductance
$\lambda_{sd,sq}$	Stator d- and q-axis flux linkage
$V_{sd,sq}$	Stator d- and q-axis voltage
$i_{sd,sq}$	Stator d- and q-axis current
λ_{fd}	Flux linkage constant
p	Number of pole pairs
ω_m	Mechanical angular velocity
ω_e	Electrical angular velocity ($\omega_r = p * \omega_m$)
T_{em}	Electromagnetic torque
$C(\cdot)$	DP Cost function
J	Jacobian matrix
$\partial C / \partial \vec{w}$	Gradient of weight vector
μ	Parameter that determines the steepest descent algorithm and Gauss Newton
LM	Levenberg-Marquardt
FATT	Forward accumulation through time
FOC	Field-oriented Control
PI	Proportional-Integral

PM	Permanent Magnet
ANN	Artificial Neural Network
NN	Neural Network
DP	Dynamic Programming
PWM	Pulse Width Modulation
SVPWM	Space Vector PWM

ACKNOWLEDGEMENTS

I would like to express the deepest appreciation to my advisor, Dr. Shuhui Li for his invaluable advice, patience, and continuous support during my PhD study. His immense knowledge and innovative ideas helped me in all the time of my academic research work. I am grateful for the guidance, encouragement, and motivation.

I would like to express my sincere gratitude to my committee members: Dr. Bharat Balasubramanian, Dr. Fei Hu, Dr. Tim A. Haskew, and Dr. Yu Gan for their great support and insightful comments. I would like to offer me special thanks to Dr. Bharat Balasubramanian for the research assistantship support. His vision, experience and invaluable advice have deeply inspired me.

Finally, I would like to thank my family and my friends for their understanding and continuous support. Without their love and encouragement, it would be impossible for me to complete my study.

CONTENTS

ABSTRACT	ii
LIST OF ABBREVIATIONS AND SYMBOLS	iv
ACKNOWLEDGEMENTS.....	vi
CONTENTS	vii
LIST OF FIGURES	xii
1. INTRODUCTION.....	1
1.1 Background and Motivation.....	1
1.2 Research Objectives.....	8
1.3 Dissertation Organization.....	10
2. OPTIMAL AND EFFICIENT OPERATION AND MANAGEMENT OF EV IPM MOTORS BASED ON ARTIFICIAL NEURAL NETWORKS	11
2.1 Introduction	11
2.2 Torque Control of IPM motor	15
2.2.1 IPM model in dq reference frame.....	16
2.2.2 Torque control of an IPM motor	17
2.2.3 The nonlinearity of motor parameters	20
2.2.4 Conventional methods to determine MTPA, flux-weakening and MTPV	21

2.3	Novel Neural Network-based MTPA, Flux-weakening and MTPV control	22
2.4	Training Data Generation and NN Training	25
2.4.1	Generation of training data.....	25
2.4.2	NN training.....	28
2.5	Integration of feedback flux-weakening controller in an IPM motor drive.....	31
2.5.1	NN in an overall IPM motor drive and control system.....	31
2.6	Simulation Evaluation.....	33
2.6.1	Operation under base speed (MTPA)	35
2.6.2	Operation at high speed (flux-weakening).....	36
2.6.3	Operation at high speed (flux-weakening) Operation at extremely high speed (flux-weakening on MTPV).....	38
2.6.4	Operation at variable and transient torque conditions	38
2.6.5	Effect of parameter uncertainties.....	39
2.7	Hardware Experiment	41
2.7.1	Hardware setup.....	41
2.7.2	Hardware results.....	42
2.8	Conclusions	48
2.9	References	49
3.	IDENTIFICATION OF IPM MOTOR MAGNETIC MODEL USING NEURAL NETWORK IN A CLOUD COMPUTING FRAMEWORK	53
3.1	Introduction	53

3.2	IPM motor model with/without core loss	56
3.2.1	IPM motor Model without core loss.....	56
3.2.2	IPM motor Model with core loss.....	58
3.3	Identification of IPM motor parameters using neural network in a cloud computing framework	59
3.3.1	Proposed NN-based parameter identification	59
3.3.2	Proposed NN in a cloud computing framework.....	62
3.4	Data Collection and NN training.....	65
3.4.1	Collection of training data.....	65
3.4.2	NN Training	66
3.5	Simulation Evaluation.....	69
3.6	Hardware Evaluation	72
3.7	References	75
4.	CONTROL OF A BUCK DC/DC CONVERTER USING APPROXIMATE DYNAMIC PROGRAMMING AND ARTIFICIAL NEURAL NETWORKS	78
4.1	Introduction	78
4.2	Conventional Control of Buck Converters.....	81
4.2.1	Buck converter model.....	81
4.2.2	VMC based control.....	83
4.2.3	CMC based control	84
4.2.4	Sliding-mode based control.....	86

4.3	ANN Control of Buck converter	87
4.3.1	Buck converter state-space model	87
4.3.2	ANN Control Structure	88
4.3.3	Maximum Duty-ratio and Current Limitations	90
4.4	Training ANN to Implement ADP-Based Control	91
4.4.1	ADP-based Control.....	91
4.4.2	Training ANN to Implement ADP	91
4.5	Simulation Evaluation.....	93
4.5.1	Tuning of Conventional Controller	93
4.5.2	Training of ANN Controller.....	93
4.5.3	Control Evaluation within System Constraints	94
4.5.4	Control Evaluation beyond System Constraints.....	97
4.6	Hardware experiment.....	99
4.6.1	Experiment setup	99
4.6.2	Experiment Results.....	100
4.7	Conclusion.....	101
4.8	References	102
5.	ARTIFICIAL NEURAL NETWORK CONTROL OF A STANDALONE DC MICROGRID	105
5.1	Introduction	105

5.2	Standalone DC microgrid.....	107
5.2.1	Configuration of a standalone DC microgrid.....	107
5.2.2	DC/DC converter model with loads	108
5.3	ANN control of a standalone DC/DC converter.....	109
5.3.1	DC/DC Converter State-Space Model.....	109
5.3.2	ANN Controller Structure.....	110
5.3.3	Training ANN Controller.....	112
5.4	Integrating ANN with Droop Control.....	113
5.4.1	Droop Control Method for DC DG Units	113
5.4.2	Power Sharing Control among Parallel DG Units.....	114
5.4.3	Integrating ANN with Droop Control.....	115
5.5	Simulation Evaluation.....	117
5.6	Conclusions	120
5.7	References	120
6.	CONCLUSIONS AND FUTURE WORKS	123
6.1	Conclusions	123
6.2	Future work	125
	REFERENCES	127

LIST OF FIGURES

Figure 1.1: Global EV sales by key markets [2].....	1
Figure 1.2: General EV configuration.....	2
Figure 1.3: Direct Torque Control	4
Figure 1.4: Field Oriented Control.....	5
Figure 1.5: EV applications	7
Figure 1.6: DC microgrid	8
Figure 2.1: Torque Control of an IPM motor	16
Figure 2.2: Torque-speed curve	18
Figure 2.3: MTPA, MTPV, current circle	19
Figure 2.4: relations of flux linkages with dq currents	21
Figure 2.5: Relations of dq currents with desired torque and flux limit	22
Figure 2.6: NN architecture for MTPA, flux-weakening, MTPA control	23
Figure 2.7: Flowchart for generation of NN training data	27
Figure 2.8: Training NN for MTPA, flux-weakening, and MTPV.....	30
Figure 2.9: Overall IPM motor drive and control system	32
Figure 2.10: Training Error of different configuration	33
Figure 2.11: Rea-time memory sizes of the NN vs. LUT	34

Figure 2.12: MTPA mode: (a) Comparison of actual MTPA line with the generated MTPA lines using LUT and NN methods, (b) d-axis current, (c) q-axis current, (d) current amplitude, (e) torque.....	36
Figure 2.13: Flux-weakening mode: (a) Comparison of desired torque line with the generated torques using LUT and NN methods, (b) d-axis current, (c) q-axis current, (d) current amplitude, (e) torque.....	37
Figure 2.14: MTPV mode: (a) Comparison of desired torque line with the generated torques using LUT and NN methods, (b) d-axis current, (c) q-axis current, (d) current amplitude, (e) torque.....	39
Figure 2.15: Variable and transient torque mode: (a) d-axis current, (b) q-axis current, (c) current amplitude, (d) torque.....	40
Figure 2.16: Impact of motor parameter uncertainty to NN-based MTPA, flux-weakening and MTPV: (a) d-axis current, (b) q-axis current, (c) current amplitude, (d) torque.....	40
Figure 2.17: Experimental testing and control systems: (a) Circuit connection, (b) Experiment setup	42
Figure 2.18: NN over LUT under MTPA mode: (a) actual vs. generated MTPA lines, (b) d-axis current, (c) q-axis current, (d) current amplitude, (e) torque	43
Figure 2.19: NN over LUT in flux weakening mode: (a) actual vs. generated torque lines, (b) d-axis current, (c) q-axis current, (d) current amplitude, (e) torque	45
Figure 2.20: NN-based control under constant flux lines and MTPA mode: (a) actual vs. generated flux and MTPA lines, (b) phase-a current	46
Figure 2.21: NN-based control under constant torque lines and maximum current circle: (a) actual vs. generated torque lines, (b) phase-a current	47
Figure 3.1: Equivalent circuit of an IPM motor without the consideration of the core loss: a) d-axis equivalent circuit, b) q-axis equivalent circuit.....	57
Figure 3.2: Equivalent circuit of an IPM motor with the consideration of the core loss: a) d-axis equivalent circuit, b) q-axis equivalent circuit	58
Figure 3.3: NN architecture used to determine IPM motor control in MTPA, flux-weakening, and MTPV	60
Figure 3.4: NN for parameter identification and the cloud computing framework in an overall IPM motor drive and control system	64
Figure 3.5: Training NN for IPM magnetic modeling	68

Figure 3.6: Estimated λ_d and λ_q vs. i_d using the proposed and convention NN approaches: (a1, b1) – $i_d=10A$, $\omega_e=500rpm$; (a2, b2) – $i_d=200A$ $i_d=10A$, $\omega_e=500rpm$; (a2, b2) – $i_d=200A$, $\omega_e=9000rpm$	70
Figure 3.7: Estimated λ_d and λ_q vs. i_q using the proposed and convention NN approaches: (a1, b1) – $i_d=10A$, $\omega_e=500rpm$; (a2, b2) – $i_d=-200A$, $\omega_e=9000rpm$	71
Figure 3.8: Estimation error of λ_d and λ_q vs. i_q using the proposed and convention NN approaches.....	72
Figure 3.9: Estimation error of λ_d and λ_q vs. i_q using the proposed NN approach.....	74
Figure 4.1:A dc/dcbuck converter with loads.....	82
Figure 4.2: Buck converter graphic model.....	83
Figure 4.3: Buck converter VMC model.....	83
Figure 4.4: Buck converter CMC cascade voltage-current control model.....	85
Figure 4.5: A PWM-based cascade SMC for Buck converter.....	87
Figure 4.6: ANN control for buck dc/dc converter.....	89
Figure 4.7: Simulated results for ANN vs. VMC Type III, CMC cascade PI, and cascade SMC: a) a load change from 7.33Ω to 11Ω , b) a reference voltage change from $18V$ to $24V$, c) an input voltage change from $42V$ to $47V$, d) the same load change as (a) when L decreases by 50%, e) the same load change as (a) when L decreases by 85%.....	96
Figure 4.8: Simulated results for ANN vs. VMC Type III, CMC cascaded PI, and SMC under: a maximum inductor current constraint of $2A$ (a1)-(a2), (b1)-(b2), (c1)-(c2), & (d1)-(d2); b) the maximum duty-ratioconstraint (a3)-(a4), (b3)-(b4), (c3)-(c4), & (d3)-(d4).....	98
Figure 4.9: Hardware laboratory testing and control systems.....	99
Figure 4.10: Hardware results for VMC vs. CMC Cascade PI vs. Cascade SMC vs. ANN: a) Change of v_o^* from $18V$ to $24V$, b) Load change from 7.33Ω to 11Ω	100
Figure 4.11: Hardware results for ANN under a) inductor current constraint; b) duty-ratio constraint.....	101
Figure 5.1: configuration of a DC microgrid.....	108

Figure 5.2: A standalone DC/DC converter with loads	109
Figure 5.3: ANN control for a standalone DC/DC converter with loads.....	111
Figure 5.4: A DC DG unit model in DC microgrid	113
Figure 5.5: Illustration of DC droop technique.	114
Figure 5.6: Two DG units connected to a common bus.....	115
Figure 5.7: Droop, virtual resistance and ANN control integration	116
Figure 5.8: Stand-alone DC microgrid with ANN and droop control	118
Figure 5.9: (a) Output voltage and (b) output currents of the converters with conventional controller	119
Figure 5.10: (a) Output voltage and (b) output currents of the converters with ANN controller	119

1. INTRODUCTION

1.1 Background and Motivation

With the increment of environmental concerns and customer desires, hybrid electric vehicles (HEVs) and electric vehicles (EVs) have drawn lots of attention to the automotive industry and researchers for many years. The number of electric vehicles on the road in the United States is also increasing dramatically since 2011, and it has been more than 1.18 million electrical vehicles on the road in 2020 [1-2]. The global electric vehicle sales by key markets are shown in Figure 1.1 [2]. One of the advantages of the electric vehicles is that it reduces harmful pollutants and greenhouse gases. EVs are also energy efficient compared with the traditional vehicles. Another advantage is their ability of charging and discharging with sustainable energy systems, such as solar panels and wind powered devices.

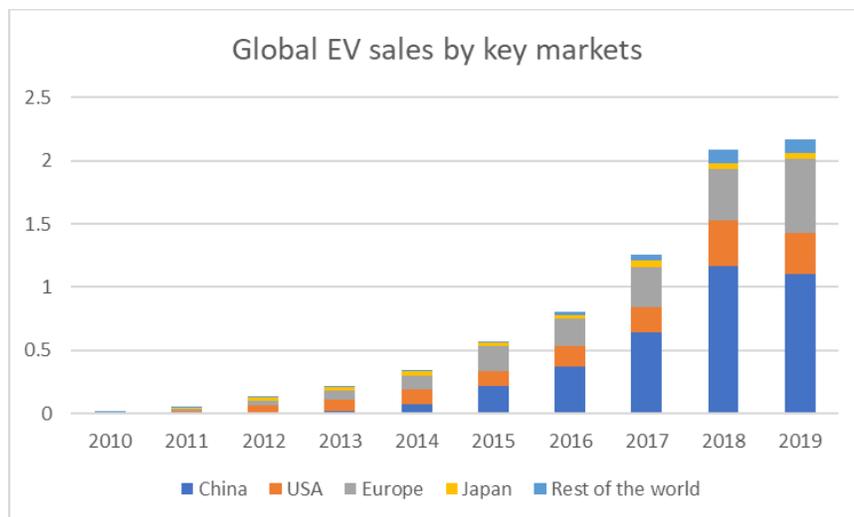


Figure 1.1: Global EV sales by key markets [2]

A general EV configuration is shown in Figure 1.2. There are three major systems in the drive train: electric motor propulsion (vehicle controller, power electronic converter, electric motor, transmission, and driving wheels), energy source (energy source, the energy management unit, and the energy refueling unit), and auxiliary (power steering unit, the auxiliary supply unit, and climate control unit) [3]. In the normal driving condition, the driver is sending the torque command to the vehicle to accelerate or decelerate, and then the torque command signals of the motor drive are received from the powertrain controller. In the cruise control condition, the speed is regulated in the powertrain control system by sending the torque command signal to the electric drive controller.

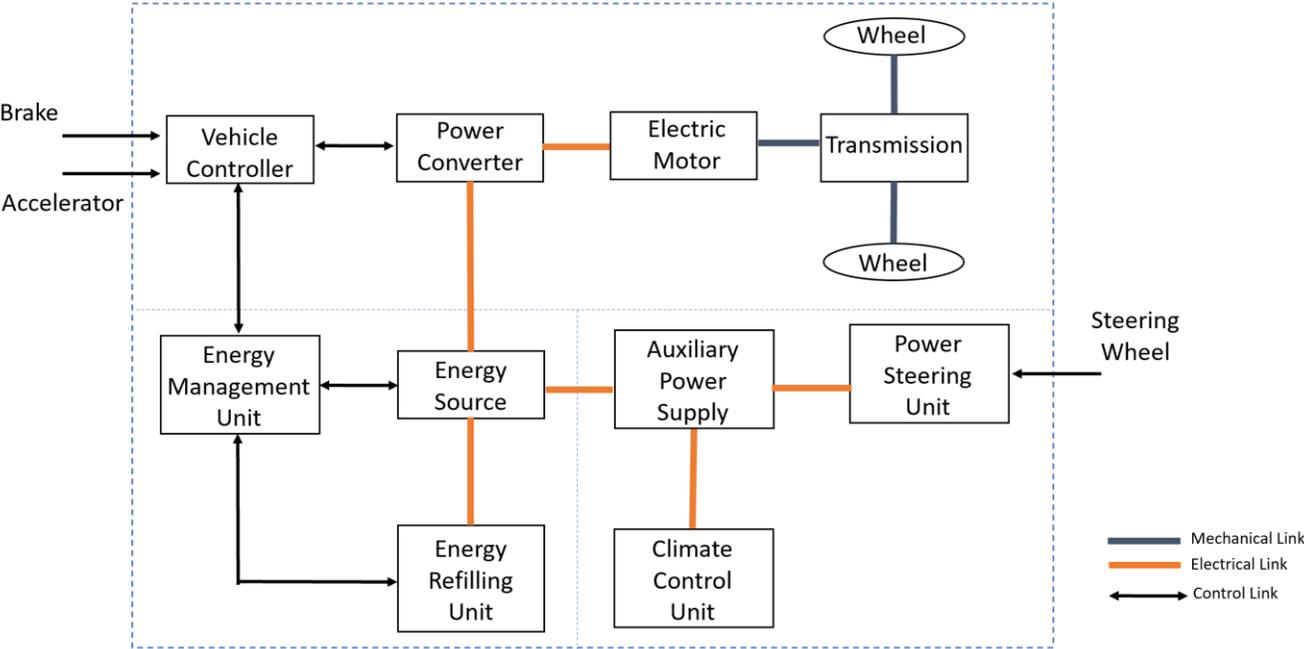


Figure 1.2: General EV configuration

As the main part of the electric propulsion system in EVs and HEVs, high-performance electric motors are required [4], [5]. Because of the advantages such as high efficiency, high power density, high torque density, and wide speed operating range, interior permanent magnet

synchronous motors (IPMSMs) have been widely used for electric vehicle drive systems. The major components are HVDC-link (generally 200V-500V), power inverter, drive controller and interior permanent magnet synchronous machine. The primary goal of this configuration is to generate the desired torque, based on torque command and other limitations such as DC input voltage, AC current, and speed. The generated torque can be positive and negative for two operation modes: motoring and regenerating [3]. A rotor position sensor, stator current sensors, dc-link voltage sensor, and stator temperature sensors are needed for the control of electric motor.

There are several motor control methods for ac motor drives such as volts/hertz control, direct torque control (DTC) and field-oriented control (vector control) [4]. The volts/hertz control is one of the simplest methods for speed control which adjusts the voltage magnitude and frequency to approximately control the speed without a speed sensor. Since it's an open-loop control method, there will be speed errors. It is also not suitable for EV applications since the primary goal is to control torque instead of speed. The direct torque control method is shown in Figure 1.3. Torque and stator flux commands are compared with the estimated values based on the measured voltages and currents to directly control the torque. Then a voltage vector lookup table is applied to send the switching signals to the inverter. The advantages of this control method are fast transient response and simple structure. The disadvantages are high torque ripples, high and variable switching frequency, and low performance at the low-speed region due to flux distortion [5-8].

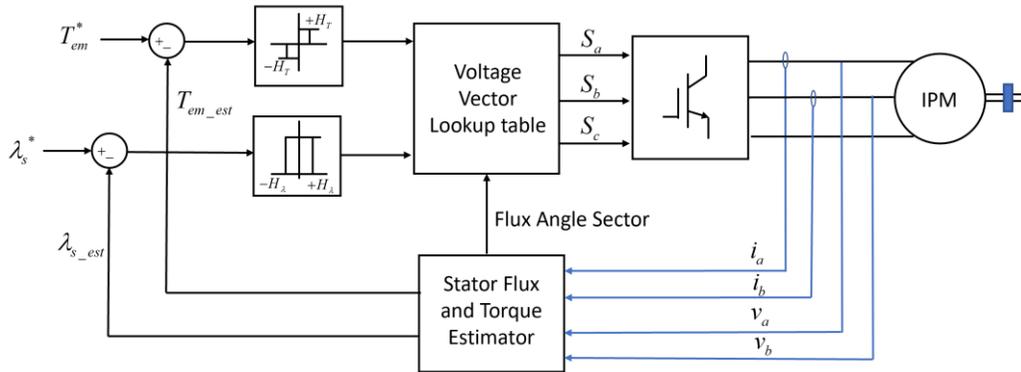


Figure 1.3: Direct Torque Control

Compared to the direct torque control, field-oriented control (FOC) or vector control strategy is more widely used for EV drive application in the industry because of the ability of power optimization, wide speed range operation and the use of PWM. In the FOC control method, the stator currents are controlled directly which allows better MTPA control and maximum efficiency control. The performance at the low-speed range is also better without flux estimation. The field-oriented control strategy for IPM is shown in Figure 1.4. Based on Park transformation, motor stator currents are transferred into two components: d-axis(direct) current and q-axis(quadrature) current. Conventionally, a rotor position sensor is required for the dq transformation and speed calculation. Then the dq currents are sent back to the current controllers. The current references for the closed-loop control are determined from the torque and the flux commands. The PI controller is generally used for the current control.

For EV applications, it is important to design motor drives with high performance such as high efficiency, wide speed operation range and accurate torque control. Therefore, a good selection of dq current commands is needed. For some linear torque controls, the d-axis current is set to zero and the q-axis current is proportional to the torque. However, the feature of IPM with reluctance torque will not be fully used with this control method. To better implement the

reluctance torque and improve the range and efficiency of IPM drives, multiple control strategies have been developed such as flux-weakening control, maximum torque per ampere control (MTPA), maximum torque per volt control (MTPV), and maximum torque per flux control (MTPF). Lots of these strategies have been also widely used in the EV industry [12-13]. At the same time, the physical constraints such as voltage limit, currents limit, torque limit and speed limit need to be considered.

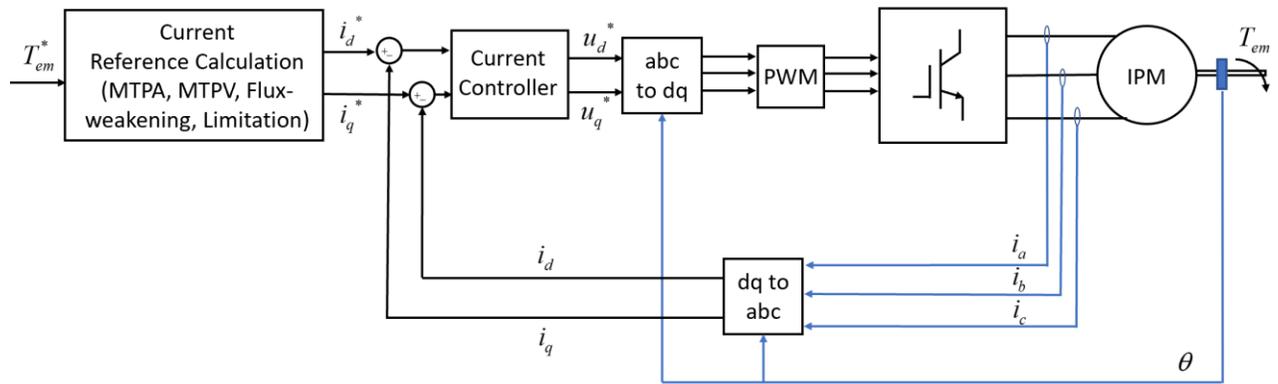


Figure 1.4: Field Oriented Control

It can be seen that it's difficult to meet all the above operating requirements for the controller design. One of the challenges is to find accurate dq-axis currents solution in different operating ranges with lots of limitations such as current, voltage, torque and flux. The solution may cause computational burdens for the standard controller boards. Therefore, a tradeoff between computation issues and control accuracy needs to be considered.

Another major challenge is the varying machine parameters such as L_d , L_q and λ_{fd} . With the iron saturation, the stator flux and permanent magnet flux are nonlinear and dq inductances are not constants [15]. The stator flux should be described as

$$\lambda_{sd} = L_d(i_{sd})i_{sd} + \lambda_{fd}$$

$$\lambda_{sq} = L_q(i_{sq})i_{sq}$$

Especially for IPMs, with sharing the same rotor iron for both d axis and q axis, the cross-saturation effect also exists in practice. Considering the cross-saturation effect, the stator inductance should be described as a function of both i_d and i_q . This feature of IPM makes the solution more complex and will significantly affect the control accuracy when the parameters are extremely nonlinear [15-17].

The third challenge is that the machine math model we generally used for controller design is totally decoupled as dq domain equations. In fact, the d axis and q axis are not independent and the coupling between d and q axis needs to be considered. With the coupled electric machine model, the machine drive controller, especially the current controller can be more complicated.

There are also some challenges for other EV applications. During the charging and discharging, EVs are able to connect with multiple types of energy sources, such as regular charging stations, PV charging stations, wind powered charging stations, smart homes, smart buildings, etc. Figure 1.5 shows the EV charging and discharging through different chargers. Besides the onboard charger, there are offboard chargers installed at the smart homes and buildings to provide higher voltage and power capability. EVs can be connected to homes or grid through the offboard charger for charging and discharging based on the requirements. The battery packs in EVs can also be used as sources to supply power to the loads for energy backups. These applications requires stable and robust voltage regulation under disturbances such as load changes and voltage variations. The Current and voltage limit in the converter and EV battery also need to be considered and handled smoothly. For V2G applications, the power rating difference between PV converter, Battery storage, and EV battery could induce high

currents and unbalancing load sharing problems. The connected DC grid and AC grid also require stable and robust voltage regulation.

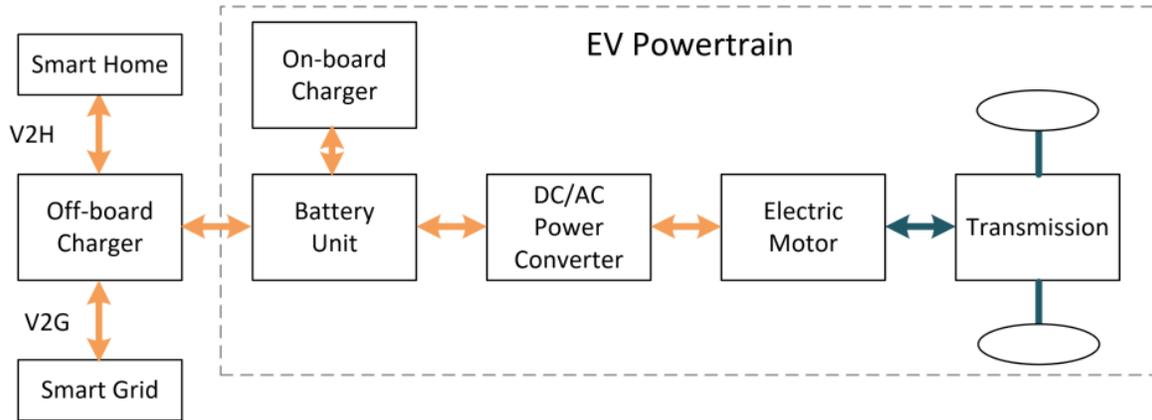


Figure 1.5: EV applications

The concept of microgrid (MG) is becoming more and more important and popular due to the development of DGs (Distributed Generation). It has the potential to integrate different DGs and ESSs (Energy Storage Systems) for many different applications. Figure 1.6 shows a configuration of a DC microgrid. EVs and HEVs are connected to a low voltage DC link bus through power electronics (onboard or offboard chargers). There are other different types of DGs, such as battery storage systems, solar PVs and wind power systems. On the low voltage DC bus, several DC loads and AC loads are connected. The DC microgrid also is able to connected to the main grid through DC/AC converters with bidirectional power flow.

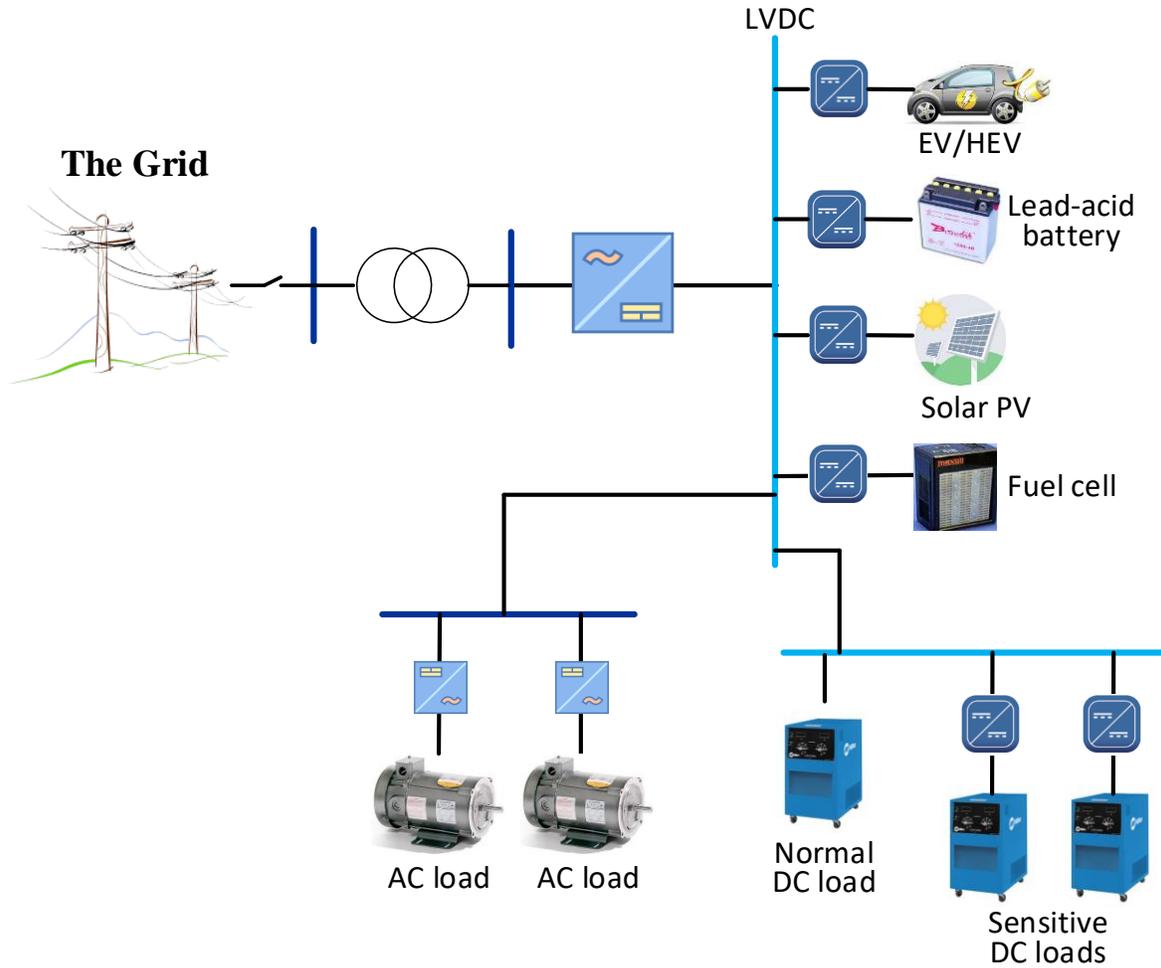


Figure 1.6: DC microgrid

1.2 Research Objectives

The first objective of this research is to develop an optimal and efficient operation and management system for EV IPM motors based on advanced controls and machine learning methods. The whole IPM drive system structure for the EV application needs to be developed and analyzed. The IPM drive needs to be modeled and studied. The system should be able to generate accurate and optimal solutions based on the requirements such as torque command, speed limit, current limit, and voltage limit. Several problems such as the copper loss, accurate torque control, parameter variations of IPM motors and multiple constraints need to be solved. At

the same time, the system response should be fast and stable. The hardware DSP computation burden and storage burden for real-time applications need to be considered.

The second objective is to develop a novel and accurate IPM modeling method and IPM motor parameter estimation to improve the motor operation under uncertainties. As described in Section 1.1, considering the iron saturation and cross-saturation effect, the stator inductances are not constant and should be described as a function of both i_d and i_q . Conventionally, the decoupled electric machine model is also used and the iron loss are eliminated. The developed IPM modeling method need overcome these described challenges and improve the motor control and operation under uncertainties, such as the high efficiency operations and current tracking dynamics.

The third objective is the controller development for DC/DC converters. The DC/DC converter with artificial neural network control is developed and analyzed in order to provide fast and accurate voltage tracking abilities. The output voltage from the EV battery should maintain stable under load disturbances and input DC voltage variations. The developed system should also have the ability of handling current and voltage constraints for the DC microgrid application.

The fourth objective is the controller development for EV integration with DC microgrid and homes. The paralleled DC/DC converters with artificial neural network control is developed and analyzed in order to provide fast and accurate voltage tracking abilities. The output voltages from DC distributed generations such as EV battery, energy storage systems, should maintain stable under load disturbances and grid voltage variations. The developed system should also have the ability of properly power sharing based on EV battery, PV converter, and Battery Storage capabilities.

1.3 Dissertation Organization

The rest of this dissertation is organized as follows.

Chapter 2 presents the Neural-Network based MTPA, Flux-Weakening and MTPV Control for IPM Motors and Drives. Chapter 3 presents the identification of IPM motor magnetic model using artificial neural network in a cloud computing framework. Chapter 4 presents control of a DC/DC converter using approximate dynamic programming and artificial neural networks. Chapter 5 discussed the DC microgrid structure and the integration of artificial neural network control with droop control. Chapter 6 presents the conclusion and the future work.

2. OPTIMAL AND EFFICIENT OPERATION AND MANAGEMENT OF EV IPM MOTORS BASED ON ARTIFICIAL NEURAL NETWORKS

2.1 Introduction

Permanent-magnet synchronous machines (PMSM) are widely used in industry because of their benefits such as high efficiency, high power density, and high torque density. Interior permanent magnet (IPM) synchronous machine is one type of PMSM which is suitable for the operation over a wide speed range due to their magnet location design and relative high saliency [1]–[3]. Especially for vehicular propulsion systems, IPM drives are able to meet the requirements for two operation modes [4], [5], constant torque operation under base speed and field-weakening operation in higher speed range. For the IPM motor drive control, these two operations are restricted by the machine torque limit, speed limit, and inverter voltage and current ratings. To have a good performance in the constant torque operation region, optimal and highly efficient control strategies, such as the maximum-torque-per-ampere (MTPA) approach, are usually adopted. The MTPA technique generates an optimum dq-axis current combination with the minimum stator current to achieve the desired torque, which reduces the motor copper loss. When the speed increases into the high-speed range, the stator voltage will reach the rated value and the flux-weakening (FW) control is adopted [6]–[8]. The IPM motor drive will then work at the voltage limit to generate the maximum available torque. If the center point of the

voltage limit is within the current limit circle, the maximum-torque-per-volt (MTPV) rule is required in the flux-weakening region as the speed continues to increase.

There are several challenges to achieve the optimal and high-efficient control for the motor operation over its full speed range in MTPA, flux-weakening, and MTPV regions. 1) Nonlinear optimization problems are complex to solve considering voltage, flux and current constraints. 2) These constraints are affected by the DC link voltage fluctuation and speed changes. 3) The IPM motor parameters are nonlinear and vary with stator currents due to the saturation and cross-magnetization effects.

To overcome these challenges, the lookup table (LUT) methods [9]–[11] are widely used by the industry, in which the MTPA trajectory, MTPV trajectory, and all the flux-weakening operating points considering all the constraints are calculated offline using different approaches. Usually, the motor inductance and flux data are first obtained through motor characterization experiments; the MTPA, flux-weakening, and MTPV operating points are then calculated using the measured data for different torque commands and flux limits and results are saved into LUTs. In the online control implementation, the d- and q-axis currents are obtained from 2D or 3D LUTs based on an input torque command. With the LUT-based method, the nonlinearity of machine parameters could be embedded in the LUTs for the motor's full-speed operating range and the result is relatively accurate. The issues with this method are the time-consuming experiments and large memory usage in controllers. The interpolation methods of LUTs could also affect the control accuracy.

Several online methods are also proposed for MTPA and flux-weakening control of IPM motors [12]–[16]. The online methods typically need online parameter estimation or premade parameter LUTs for motor inductance and flux data. However, conversion from a input torque

command to the d- and q-axis currents are calculated online using different algorithms instead of using LUTs. In [16], the Ferrari's method and a recursive algorithm are used to obtain the MTPA and flux-weakening operating points. In [15], an improved online parameter estimation technique of motor inductance and flux data is proposed and used with an online algorithm to improve the accuracy of obtaining the optimal MTPA points. Overall, the online methods do not need large memory storage in the microprocessor compared to the LUT-based method, but have more complex structure if the motor operations between MTPA and flux-weakening modes are considered. The dynamic performance is also an issue compared to the off-line methods due to slow convergence and online computation. Sometimes ill-convergence and computation burden also affect the control accuracy and the system stability.

Signal injection techniques [17]–[19] are another method widely used in obtaining MTPA operating points. By injecting a high frequency current or other signal into the motor, the torque or speed variations are detected, and then the MTPA point can be found at the point where the derivative of torque is zero. The advantage of this method is no need to estimate motor parameters. Although the robustness is good, it could generate more torque ripples and harmonics due to the high frequency signal injection. In addition, since only the torque derivative with respect to the current is focused, the methods are not effective in the field-weakening region. Thus, it is challenging to have the smooth transition between the MTPA and flux-weakening operations using a signal injection method, which increases the complexity of the controller design to meet the requirement of the motor's operation over its full-speed range.

The artificial intelligence (AI) based methods have been implemented in different motor drive control areas such as machine modeling [19], current control [20], speed control [21], and optimal efficiency control [22], [23]. Recently, some intelligent-based methods for MTPA and

flux-weakening control have been proposed, including fuzzy-logic, Levenberg- Marquardt (LM), particle swarm optimization (PSO), and neural network (NN) methods. In [24], a fuzzy logic rule is developed to operate as an artificial decision-maker to generate optimal d-q currents references. However, since the output current values are based on the linguistic values with defuzzification, the results are not accurate and even less precise than the traditional MTPA methods [24]. In [25], [26], fuzzy logic methods are applied in an online searching structure to find the optimal point by only adjusting the d-axis current. In [27], [28], a PSO technique is used to solve the optimization problem online to generate the optimal current references or the optimal angle. In [12], a LM method is employed to calculate the optimal d-q current references. Overall, these approaches, are applied in an online searching structure, which can cause isolation problems and convergence issues during the searching process and has the limitations to address all the aspects and needs of MTPA, flux-weakening, and MTPV.

The conventional NN-based MTPA and flux-weakening approaches developed by others generally focus on MTPA only, require training of the NN online, or perform the NN training offline. In [22], a NN is designed and trained online to achieve an online MTPA tracking control. One limitation of this NN approach is that it is trained to support the MTPA control only. Other disadvantages are related to its online training nature, including (i) difficulty for real-time implementation, (ii) potential destabilization of the NN weights at runtime, (iii) instabilities of online trained NN under disturbances/noises, all of which could cause disastrous effects to IPM drives and control. In [23], an offline trained NN is proposed to achieve the MTPA control only. Although the offline trained NN avoids the disadvantages of the online training, this NN design cannot assure the most efficient MTPA as the training is based on random torque and flux signals. Also, the NN cannot be applied in the flux-weakening region as it is trained to achieve

the MTPA control only, which requires other complex designs to be added for the motor's operation in the high-speed range. Thus, the transition between the MTPA and flux-weakening would also be a challenge.

The novel contributions of this research include: 1) an offline- trained NN to cover all the MTPA, flux-weakening, and MTPV needs into the NN design for the most efficient IPM torque control over its full speed operating range of an IPM motor, 2) a special technique to generate training data that is suitable to the development of a high-performance NN-based IPM torque control system based on motor operation data collected and measured either offline or online, 3) an embedment of variable motor parameters into the training data and NN system development, and 4) a cloud computing concept to improve the lifetime NN adaptivity and learning capabilities. The paper provides a fast and accurate current reference generation technique with a simple NN structure, for optimal torque control under different conditions and an efficient NN training mechanism based on the LM backpropagation (LMBP) algorithm. The performance of the NN-based MTPA, flux-weakening, and MTPV is evaluated via both electromagnetic transient (EMT) simulation and hardware experiments in various aspects.

2.2 Torque Control of IPM motor

Torque control of an IPM motor is usually achieved through the typical configuration shown in Figure 2.1. Basically, a reference torque command T_{em}^* is first converted into motor d- and q-axis current references, i_d^* and i_q^* . Then, the current controller regulates the motor stator currents, i_d and i_q , to follow the reference currents. The current controller outputs d- and q-axis control voltages, v_d^* and v_q^* , which control the operation of the IPM motor normally via the space vector pulse-width modulation (SVPWM) scheme.

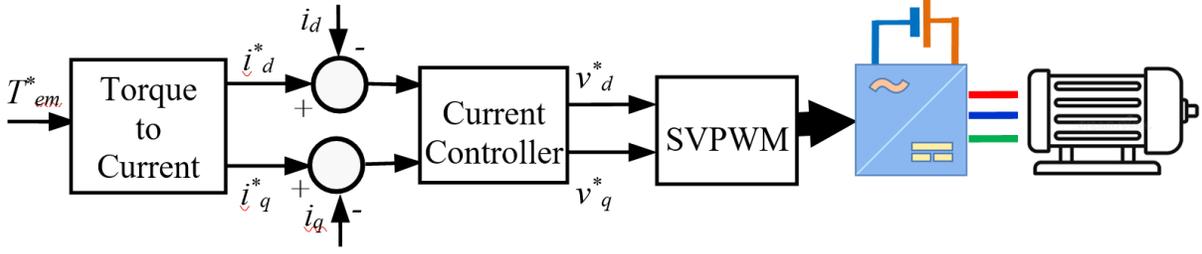


Figure 2.1: Torque Control of an IPM motor

2.2.1 IPM model in dq reference frame

Based on the Park-Clarke transformation, the stator dq flux linkages of an IPM motor are described by:

$$\begin{bmatrix} \lambda_d \\ \lambda_q \end{bmatrix} = \begin{bmatrix} L_d & 0 \\ 0 & L_q \end{bmatrix} \begin{bmatrix} i_d \\ i_q \end{bmatrix} + \begin{bmatrix} \lambda_{pm} \\ 0 \end{bmatrix} \quad (2-1)$$

where λ_d , λ_q , L_d , L_q , i_d and i_q are stator d- and q-axis flux linkages, inductances, and currents and λ_{pm} is the flux linkage of the rotor permanent-magnet. The stator dq voltage equation is:

$$\begin{bmatrix} v_d \\ v_q \end{bmatrix} = R \begin{bmatrix} i_d \\ i_q \end{bmatrix} + \frac{d}{dt} \begin{bmatrix} \lambda_d \\ \lambda_q \end{bmatrix} + \omega_e \begin{bmatrix} -\lambda_q \\ \lambda_d \end{bmatrix} \quad (2-2)$$

where v_d and v_q are d- and q-axis stator voltages, and ω_e is the motor electrical speed. The steady-state voltage equation is

$$v_d = Ri_d - \omega_e \lambda_q = Ri_d - \omega_e L_q i_q \quad (2-3)$$

$$v_q = Ri_q + \omega_e \lambda_d = Ri_q + \omega_e (L_d i_d + \lambda_{pm}) \quad (2-4)$$

The electromagnetic torque of the motor can be written as:

$$T_{em} = \frac{P}{2} (\lambda_d i_q - \lambda_q i_d) = \frac{P}{2} [\lambda_{pm} - (L_d - L_q) i_d] i_q \quad (2-5)$$

2.2.2 Torque control of an IPM motor

According to (2-5), for a given torque command T_{em}^* , there are multiple current vectors that can generate the specified torque. Hence, in order to get the most efficient current references and, at the same time, assure the motor operation within its physical constraints, the conversion from a torque reference to the current reference should be obtained by considering the following equations:

$$\text{Minimize } \sqrt{i_d^2 + i_q^2} \text{ as much as possible} \quad (2-6)$$

Subject to:

$$\sqrt{i_d^2 + i_q^2} \leq i_{\max} \quad (2-7)$$

$$\begin{aligned} \sqrt{v_d^2 + v_q^2} &= \sqrt{\left[\omega_e (L_d i_d + \lambda_{pm})\right]^2 + (\omega_e L_q i_q)^2} \leq v_{\max} \\ \text{or } \sqrt{(L_d i_d + \lambda_{pm})^2 + (L_q i_q)^2} &\leq \frac{\sqrt{6}V_{dc}}{\pi\omega_e} = \lambda_{\text{lim},\omega_e} \end{aligned} \quad (2-8)$$

where (2-6) means that the amplitude of the current reference should be as small as possible; (2-7) means that the amplitude of the current reference should not exceed the rated current limit of the motor and inverter; (2-8) means that the current reference should not require a voltage higher than the maximum voltage $v_{\max} = \sqrt{6}V_{dc}/\pi$ that the inverter can provide under the SVPWM, or that the current reference should not require a flux linkage higher than the maximum flux linkage $\lambda_{\text{lim},\omega_e} = \sqrt{6}V_{dc}/(\pi\omega_e)$ that can be provided at speed ω_e . Note: $\sqrt{6}V_{dc}/\pi$ represents the maximum dq voltage that can be achieved under the six-step operation mode, under which more torque oscillations would appear depending on the design of the current-loop controller [29].

With these two limits, the operating ranges can be divided into three parts: the base speed range, the lower field weakening range and the upper field weakening range. The speed-torque

curve is shown in Figure 2.2. In the based speed range, our objective is to generate the desired torque while having the maximum efficiency, and the maximum torque is usually limited by the maximum current. In order to have the optimal efficiency, the MTPA control method is used to reduce the copper loss in the machine. The stator current i_s should be minimized while generating the desired torque. In the lower field weakening range, both the stator voltage and current are at their limits. The maximum torque in the base speed range cannot be achieved in this operating range. The operating points are the intersections of current and voltage limits. In the upper field weakening range, the speed is increasing and needs lower torque which does not need to consider the current limit. Only the voltage is constant as v_{max} in this range. With the speed increasing, the voltage limit circle is shrinking to the center point.

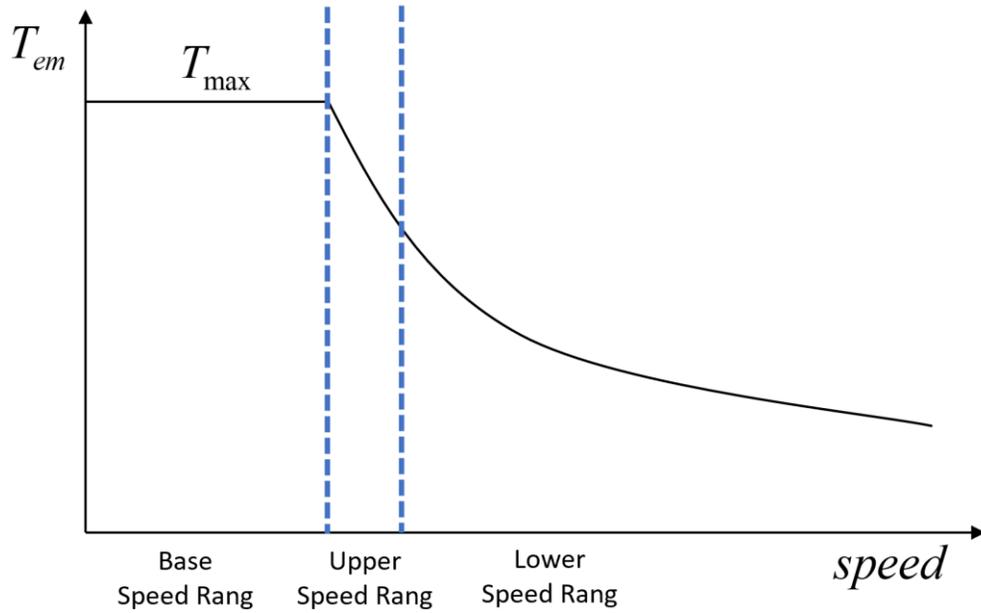


Figure 2.2: Torque-speed curve

Within the stator d- and q-axis current plane, i.e. i_d and i_q i_d plane (Figure 2.3), the current limit is shown by one circle and the flux linkage limit is shown by multiple ellipses that shrink to the center point $X = (-\lambda_{pm}/L_d, 0)$ as the motor speed increases from 0 rad/sec to the infinity.

Depending on the motor operating speed, the solution of the current reference from (2-7) and (2-8) is illustrated by the OA (MTPA), AB (flux weakening I), BC (flux weakening II), and CX (MTPV) segments [9].

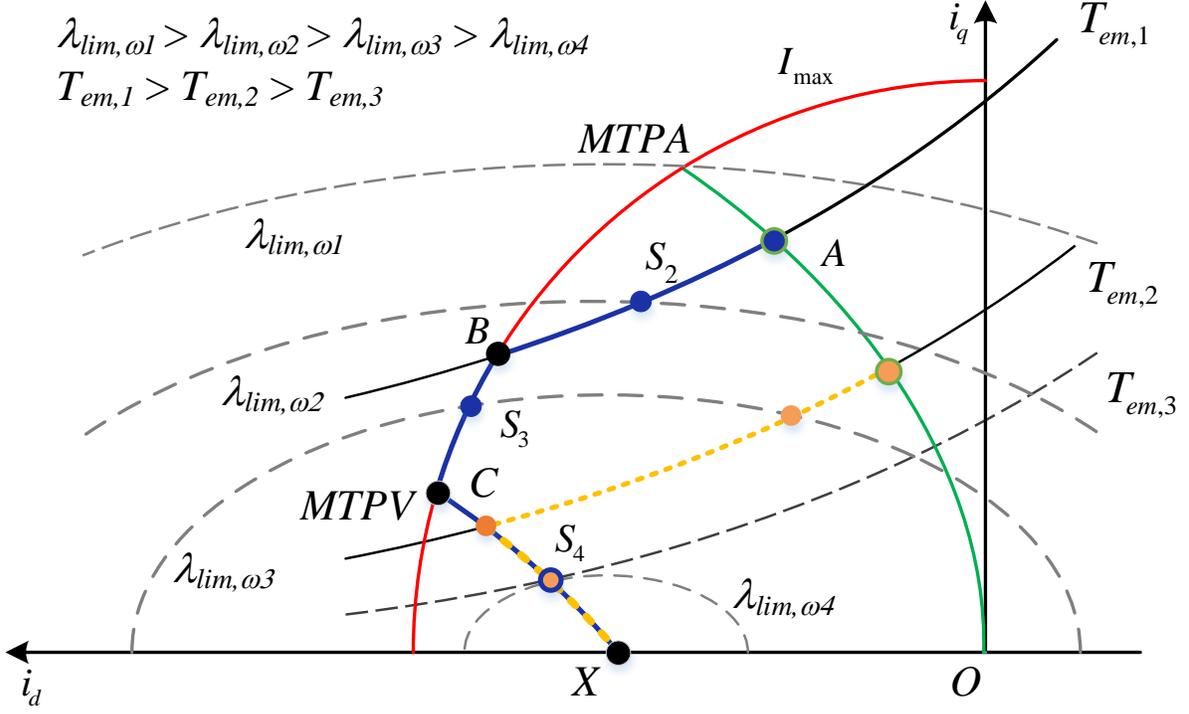


Figure 2.3: MTPA, MTPV, current circle

(1) **MTPA**. In the OA segment, the motor operates at point A for a given torque $T_{em,1}^*$, which is within both the current and flux linkage limits if the motor speed is below ω_1 . Thus, the goal is to minimize the reference current for a desired torque. This is mathematically described as

$$\begin{aligned} &\text{minimize } |i_s| = \sqrt{i_d^2 + i_q^2} \\ &\text{subject to } T_{em}^* = \frac{P}{2} \left(\lambda_{pm} i_q + (L_d - L_q) i_d i_q \right) \end{aligned}$$

(2) **Flux-weakening I**. In the AB segment, with the increase of the motor speed beyond ω_1 , the flux linkage limit is reached but the current limit is not. Thus, the goal is to minimize the

current reference for a desired torque considering the flux linkage limit. This is mathematically described as

$$\begin{aligned} & \text{minimize } |i_s| = \sqrt{i_d^2 + i_q^2} \\ & \text{subject to } T_{em}^* = \frac{P}{2} (\lambda_{df} i_q + (L_d - L_q) i_d i_q) \\ & \quad \sqrt{(L_d i_d + \lambda_{pm})^2 + (L_q i_q)^2} \leq \lambda_{lim, \omega_e} \end{aligned}$$

(3) **Flux-weakening II.** In the BC segment, with the further increase of the motor speed, both the flux linkage and current limits are reached. Thus, the desired torque cannot be achieved and the goal is to minimize the difference between the desired and actual torque while the motor operates at the rated current. This is mathematically described as

$$\begin{aligned} & \text{minimize } \left| T_{em}^* - \frac{P}{2} (\lambda_{pm} i_q + (L_d - L_q) i_d i_q) \right| \\ & \text{subject to } \sqrt{i_d^2 + i_q^2} = i_{max} \\ & \quad \sqrt{(L_d i_d + \lambda_{pm})^2 + (L_q i_q)^2} \leq \lambda_{lim, \omega_e} \end{aligned}$$

(4) **MTPV.** In the CX segment, with the further increase of the motor speed, the goal is to minimize the difference between the desired and actual torque while considering flux linkage and current limits. This is mathematically described as

$$\begin{aligned} & \text{minimize } \left| T_{em}^* - \frac{P}{2} (\lambda_{pm} i_q + (L_d - L_q) i_d i_q) \right| \\ & \text{subject to } \sqrt{i_d^2 + i_q^2} < i_{max} \\ & \quad \sqrt{(L_d i_d + \lambda_{pm})^2 + (L_q i_q)^2} \leq \lambda_{lim, \omega_e} \end{aligned}$$

2.2.3 The nonlinearity of motor parameters

One of the major challenges for the IPM motor control is the machine parameter variations as shown in . With the iron saturation and cross-saturation, the dq-axis flux-linkages

are nonlinear and can be described as functions of stator currents: $\lambda_d(i_d, i_q)$ and $\lambda_q(i_d, i_q)$. To improve the accuracy of converting the torque reference to the current references, the parameter variation properties of the motor need to be considered.

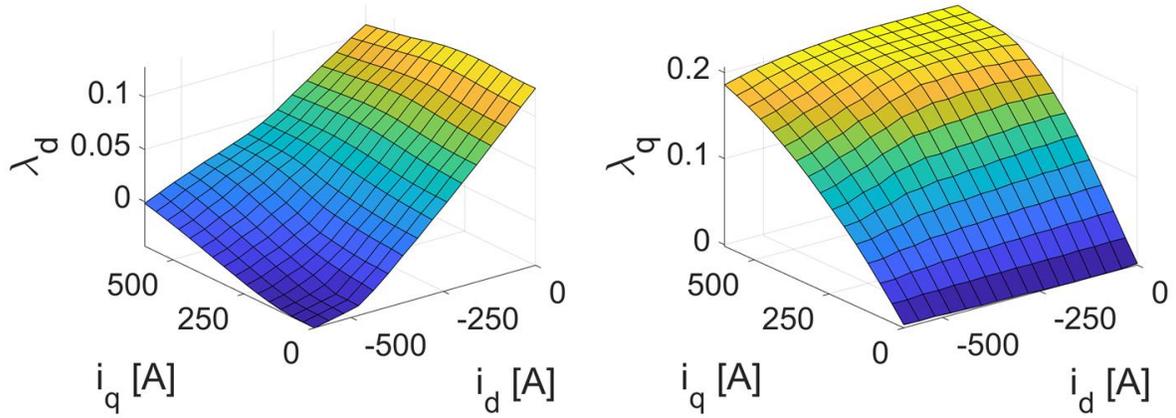


Figure 2.4: relations of flux linkages with dq currents

2.2.4 Conventional methods to determine MTPA, flux-weakening and MTPV

As shown in Section 2.1, there have been several conventional methods to determine the MTPA, flux-weakening, and MTPV operating points for an IPM motor, among which the LUT approach is a widely used technique in motor drives and EV (electric vehicle) industry [8-10].

The basic idea of the approach is to generate two lookup tables: 1) $i_d^* = f_d(T_{em}^*, \lambda_{lim, \omega_e})$ and 2) $i_q^* = f_q(T_{em}^*, \lambda_{lim, \omega_e})$ as illustrated in Figure 2.5. The generated LUTs will give the optimal d- and q-axis current references for the desired torque command under a given motor operating speed and inverter DC voltage.

One advantage is that the motor d- and q-axis flux linkages, $\lambda_d(i_d, i_q)$ and $\lambda_q(i_d, i_q)$, can be embedded in the LUTs. Since the LUTs are generated offline, this method also has good stability and dynamic performance and does not have a high computational burden compared to

the online-based methods. However, the disadvantage is the requirement for a large memory size to store the lookup tables. Also, the LUT size and interpolation methods will affect the accuracy in determining the d- and q-axis reference currents.

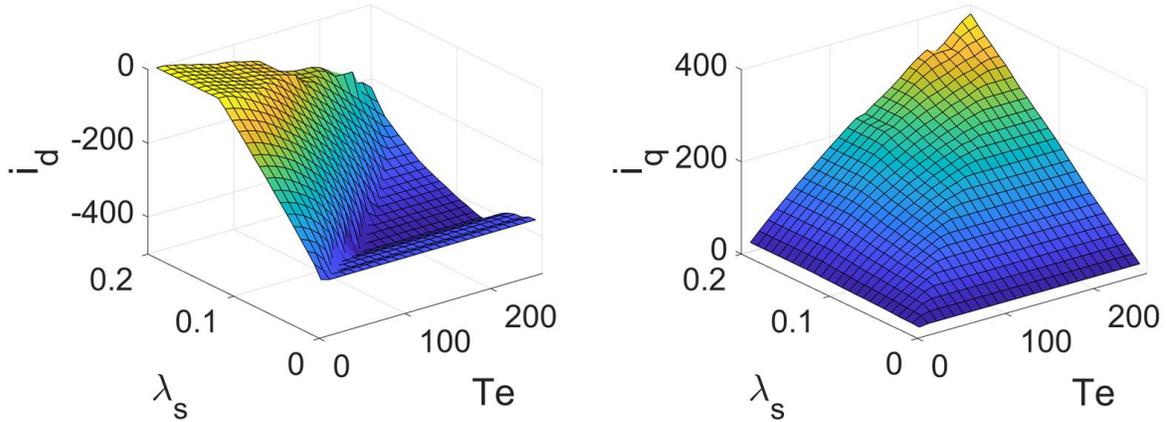


Figure 2.5: Relations of dq currents with desired torque and flux limit

2.3 Novel Neural Network-based MTPA, Flux-weakening and MTPV control

The designed NN structure is a feedforward NN as shown in Figure 2.6. The NN consists of an input layer, hidden layers, and an output layer. There are two inputs at the input layer, one is the desired torque command, the other is the flux linkage limit which is calculated based on the inverter DC voltage and the motor speed. At the output layer, the two outputs are the dq reference currents. In Section 2.5, how the selection of the number of hidden layers and the number of neurons in each hidden layer would affect the NN performance is further evaluated to get a specific NN structure for the proposed work.

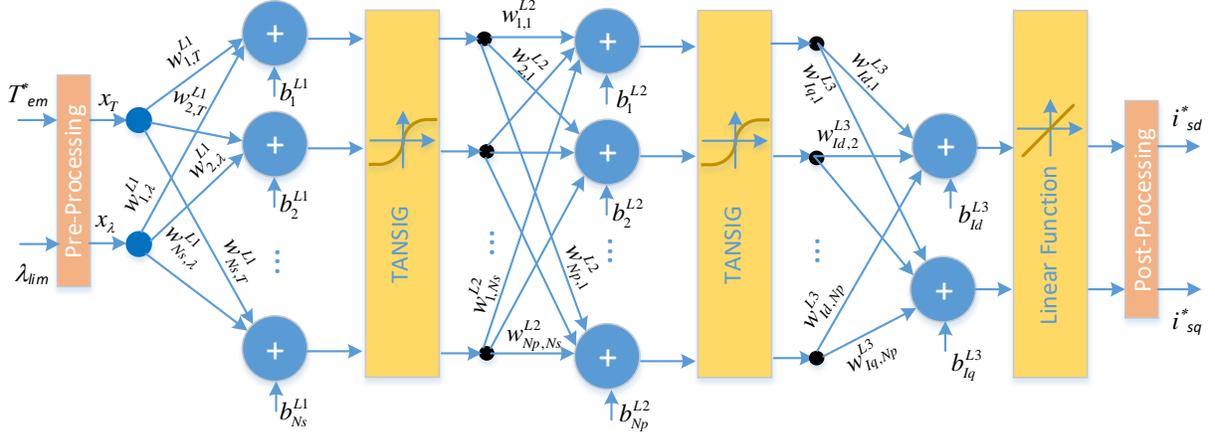


Figure 2.6: NN architecture for MTPA, flux-weakening, MTPA control

In the first hidden layer, the output of the s^{th} neuron can be expressed as

$$n_s^{L1} = w_{s,T}^{L1} \cdot x_T + w_{s,\lambda}^{L1} \cdot x_\lambda + b_s^{L1} \quad \forall s \quad (2-9)$$

where x_T and x_λ are the desired torque command and the flux linkage limit after preprocessing,

$w_{s,T}^{L1}$ and $w_{s,\lambda}^{L1}$ are the input weights corresponding to x_T and x_λ , and b_s^{L1} is the bias for the s^{th}

neuron in the first hidden layer. Then, the hyperbolic tangent sigmoid transfer functions are

adopted for the neuron outputs n_s^{L1}

$$a_s^{L1} = \frac{2}{1 + \exp(-2 \cdot n_s^{L1})} - 1 \quad \forall s \quad (2-10)$$

In the second hidden layer, the p^{th} neuron output is

$$m_p^{L2} = \sum_{s=1}^{N_s} w_{p,s}^{L2} a_s^{L1} + b_p^{L2} \quad \forall s, \forall p \quad (2-11)$$

where N_s is the number of neurons of the first hidden layer, $w_{p,s}^{L2}$ and b_p^{L2} are the weights and bias for the p^{th} neuron of the second hidden layer. Similarly, the sigmoid functions are adopted for the neuron outputs m_p^{L2} of the second hidden layer

$$a_p^{L2} = \frac{2}{1 + \exp(-2 \cdot m_p^{L2})} - 1 \quad \forall s, \forall p \quad (2-12)$$

Then, the normalized reference currents y_{Id} and y_{Iq} are generated at the output layer

$$y_{Id} = \sum_{p=1}^{N_p} w_{Id,p}^{L3} a_p^{L2} + b_{Id}^{L3} \quad (2-13)$$

$$y_{Iq} = \sum_{p=1}^{N_p} w_{Iq,p}^{L3} a_p^{L2} + b_{Iq}^{L3} \quad (2-14)$$

where N_p is the number of neurons of the second hidden layer, $w_{Iq,p}^{L3}$, b_{Id}^{L3} and b_{Iq}^{L3} are the weights the bias of the output layer.

The NN inputs and outputs are related to the physical variables of torque, flux linkage, and stator d- and q-axis currents via pre- and post-processing by

$$x_{T,i} = 2 \cdot \frac{T_{em,i}^* - \min(\mathbf{T}_{em}^*)}{\max(\mathbf{T}_{em}^*) - \min(\mathbf{T}_{em}^*)} - 1 \quad (2-15)$$

$$x_{\lambda,i} = 2 \cdot \frac{\lambda_{lim,i} - \min(\lambda_{lim})}{\max(\lambda_{lim}) - \min(\lambda_{lim})} - 1 \quad (2-16)$$

$$i_{sd,i}^* = \frac{(y_{Id,i} + 1)(\max(\mathbf{i}_{sd}^*) - \min(\mathbf{i}_{sd}^*))}{2} + \min(\mathbf{i}_{sd}^*) \quad (2-17)$$

$$i_{sq,i}^* = \frac{(y_{Iq,i} + 1)(\max(\mathbf{i}_{sq}^*) - \min(\mathbf{i}_{sq}^*))}{2} + \min(\mathbf{i}_{sq}^*) \quad (2-18)$$

where $T_{em,i}^*$, $\lambda_{lim,i}$, $i_{d,i}^*$ and $i_{q,i}^*$ are the i th sample of the desired torque, flux linkage limit, d-axis current reference, and q-axis current reference, respectively; \mathbf{T}_{em}^* , λ_{lim} , \mathbf{i}_{d}^* , and \mathbf{i}_{q}^* represent the training data set of all the desired torques, flux linkage limits, d-axis current references, and q-axis current references, respectively.

2.4 Training Data Generation and NN Training

2.4.1 Generation of training data

The training data is generated based on actual measured motor operation data that can be obtained either through offline measurement or online measurement via the cloud computing platform. The data set for the NN training consists of a large number of input-output pairs. Each pair contains two input data of desired torque command T_{em}^* and flux linkage limit λ_{lim} , and two output data of optimal d- and q-axis reference currents i_{d}^* and i_{q}^* . The data set should be large and detailed enough to assure accurate, optimal, and efficient IPM operation under MTPA, flux-weakening, and MTPV modes. Section 2.5 evaluates the size of the training data set for the IPM motor suitable for practical EV applications and used in the simulation study.

Figure 2.7 shows the flowchart to generate the training data. First, measured data of d- and q-axis flux leakages vs. d- and q-axis currents are loaded into the memory. Then, the training data generation follows these steps: 1) getting all the input-output pairs $\left[\left(T_{em}^*, \lambda_{lim} \right) \rightarrow \left(i_d^*, i_q^* \right) \right]$ along the MTPA path, current constraint path, and MTPV path and storing all the input-output pairs $\left[\left(T_{em}^*, \lambda_{lim} \right) \rightarrow \left(i_d^*, i_q^* \right) \right]$ into the data storage, 2) calculating all the input-output pairs $\left[\left(T_{em}^*, \lambda_{lim} \right) \rightarrow \left(i_d^*, i_q^* \right) \right]$ inside the MTPA path, current constraint path, and MTPV path and storing all the input-output pairs $\left[\left(T_{em}^*, \lambda_{lim} \right) \rightarrow \left(i_d^*, i_q^* \right) \right]$ into the data storage, 3) extending the data generation along the MTPA path to the full flux range, and 4) extending the data generation along the MTPV path to the full torque reference range. The need for Step 3 is due to the fact that the MTPA solution, i_d^* and i_q^* , for a desired torque T_{em}^* under $\lambda_{lim,\omega 1}$ condition, as shown in Figure 2.3, is also the MTPA solution of the same desired torque T_{em}^* under any other $\lambda_{lim,\omega}$ which has a value larger than that of $\lambda_{lim,\omega 1}$. Thus, those input-output pairs $\left[\left(T_{em}^*, \lambda_{lim,\omega} \right) \rightarrow \left(i_d^*, i_q^* \right) \right]$ over the MTPA path should be included in the training data set. Similarly, the need for Step 4 is due to the fact that the MTPV solution, i_d^* and i_q^* , under $\lambda_{lim,\omega 3}$ condition for a desired torque T_{em}^* , as shown in Figure 2.3, is also the MTPV solution of any other desired torque $T_{em,i}^*$ that has a value larger than that of T_{em}^* . Therefore, those input-output pairs over the MTPV path should also be included in the training data set.

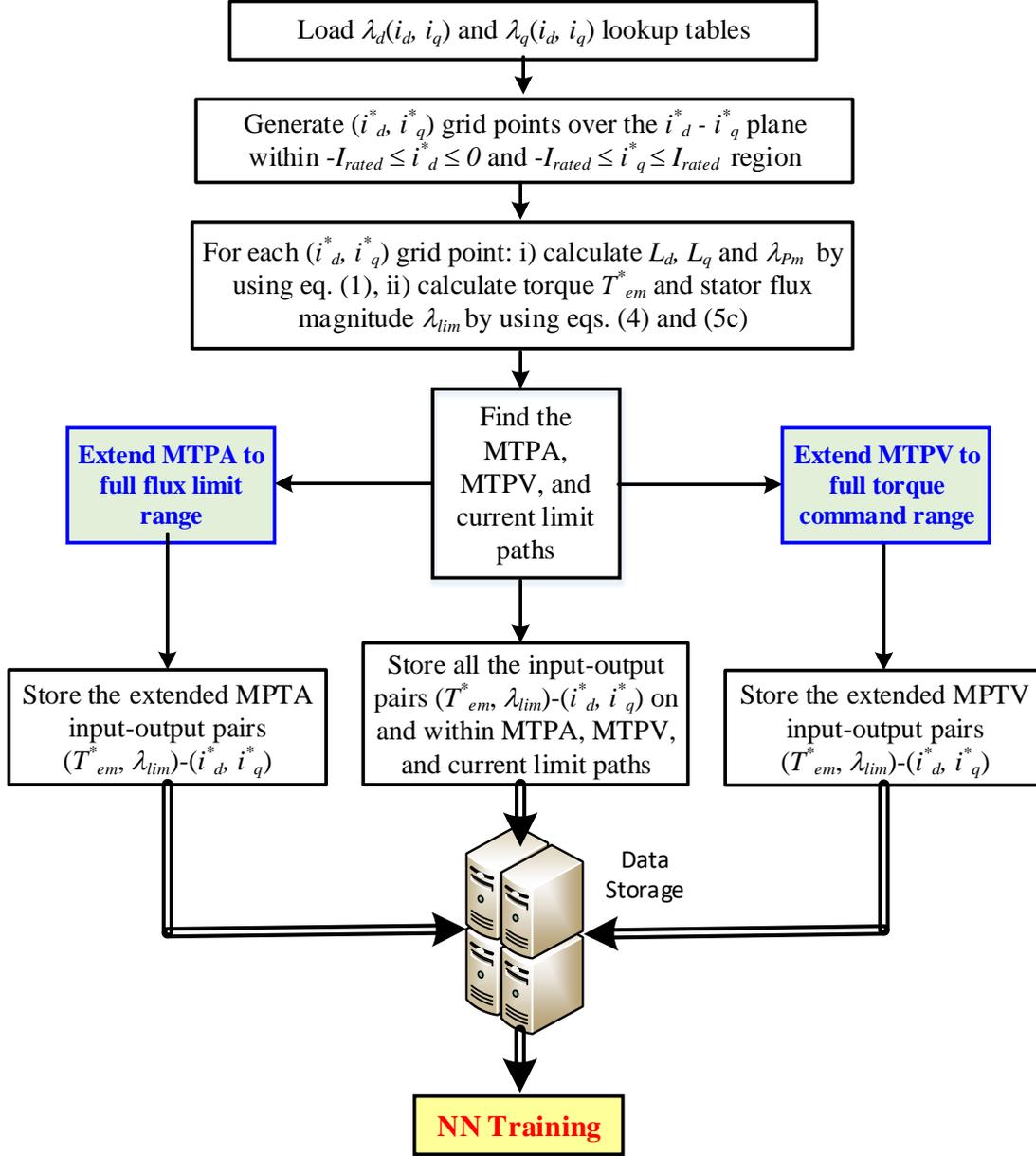


Figure 2.7: Flowchart for generation of NN training data

Some other important characteristics about the training data generated in this paper include: i) Unlike the traditional LUT approaches, the training data can be stored in the data storage in a random manner instead of a well-structured style that is typically needed in order to search an i_d^* and i_q^* solution from the LUTs efficiently; ii) The motor parameter variation is embedded into the training data set because all the i_d^* and i_q^* solutions are obtained and

calculated through actual measured data of d- and q-axis flux leakages over d- and q-axis currents that contain the motor current impact on the flux linkages; iii) A large amount of training data can be added into the data storage conveniently to enable information equity and sufficiency to benefit the training and the trained NN to handle the extremely nonlinear portions between the input data $(T_{em}^*, \lambda_{lim,\omega})$ and output data (i_d^*, i_q^*) than other methods; iv) Just an adequate amount of data that can cover all possible operating conditions of an IPM motor under MTPA, flux-weakening, and MTPV is needed to be kept as training data. In Section 2.5, the size of the training data is about one order of magnitude more than that for the LUT approach.

2.4.2 NN training

The NN is trained offline to obtain the optimum weights and biases. The performance function used for training the NN is

$$C = \frac{1}{N} \sum_{k=1}^N \vec{e}_{dq}(k) \quad (2-19)$$

$$\vec{e}_{dq}(k) = \sqrt{\left[i_{d_NN}^*(k) - i_d^*(k) \right]^2 + \left[i_{q_NN}^*(k) - i_q^*(k) \right]^2} \quad (2-20)$$

where N is the total number of samples of the training data set, $i_d^*(k)$ and $i_q^*(k)$ are the target d- and q-axis current references of the kth training sample, and $i_{d_NN}^*(k)$ and $i_{q_NN}^*(k)$ are the corresponding NN outputs.

The NN is trained to minimize the performance function (2-19) and (2-20). We used the LMBP algorithm [25] to update the weights and biases at each training iteration. In order to use the LMBP for training the NN efficiently, the performance function is rewritten as

$$C = \frac{1}{N} \sum_{k=1}^N e_{dq}(k) \xrightarrow{\text{define } V(k)=\sqrt{e_{dq}(k)}} C = \frac{1}{N} \sum_{k=1}^N (V(k))^2 \quad (2-21)$$

Then, the gradient $\partial C / \partial \vec{w}$ can be written in a matrix as

$$\frac{\partial C}{\partial \vec{w}} = \frac{\partial \sum_{k=1}^N [V(k)]^2}{\partial \vec{w}} = \sum_{k=1}^N 2V(k) \frac{\partial V(k)}{\partial \vec{w}} = 2J(\vec{w})^T \vec{V} \quad (2-22)$$

where the Jacobian matrix $J(\vec{w})$ is

$$J(\vec{w}) = \begin{bmatrix} \frac{\partial V(1)}{\partial w_1} & \dots & \frac{\partial V(1)}{\partial w_M} \\ \vdots & \ddots & \vdots \\ \frac{\partial V(N)}{\partial w_1} & \dots & \frac{\partial V(N)}{\partial w_M} \end{bmatrix}, \quad \vec{V} = \begin{bmatrix} V(1) \\ \vdots \\ V(N) \end{bmatrix} \quad (2-23)$$

Therefore, the weight update [30] is expressed by

$$\Delta \vec{w} = - \left[J(\vec{w})^T J(\vec{w}) + \mu \mathbf{I} \right]^{-1} J(\vec{w})^T \vec{V} \quad (2-24)$$

$$\vec{w}_{\text{update}} = \vec{w} + \Delta \vec{w} \quad (2-25)$$

where μ is the damping parameter.

The training procedure for each epoch is illustrated in Figure 2.8. The batch training method is adopted because of its fast convergence performance with smaller errors than other methods. For each iteration, all the stored data are used to generate the errors and Jacobian matrix. Then, the gradients are computed to update the weights and biases. The training stops when the performance function meets the requirement.

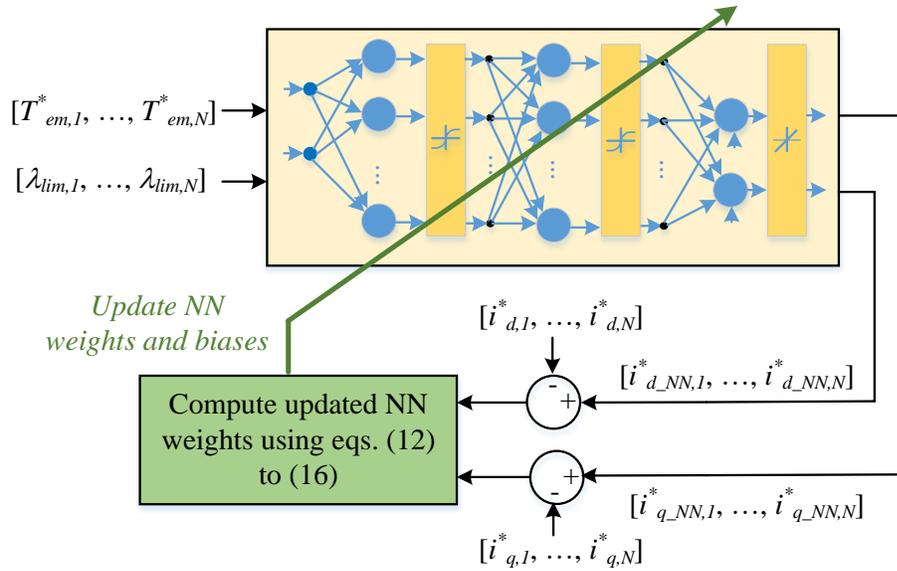


Figure 2.8: Training NN for MTPA, flux-weakening, and MTPV

For the routine offline training of the NN over the cloud computing platform, it will follow the same strategy except that data needs to be collected and transmitted to a remote cloud device for the NN training. Firstly, real-time motor operation data is collected smartly, meaning that only meaningful and valuable data is collected and stored, and data collected should cover the full operating range of the motor at the normal and critical conditions. When sufficient data is obtained, data is sent to a remote cloud computing device, where data is analyzed and, if needed, a cloud-based training of the NN will be conducted thoroughly. After the training, the new NN weights are transmitted back to the EV computer to replace the previous NN weights. Such an offline learning strategy, based upon a cloud computing platform, will guaranty the safe and reliable NN system development that cannot be achieved for an online NN learning system, which will assure the high-performance operation of individual EV motor over its lifetime.

2.5 Integration of feedback flux-weakening controller in an IPM motor drive

2.5.1 NN in an overall IPM motor drive and control system

The NN in the overall IPM motor control system is shown in Figure 2.9. The desired torque first passes through a Torque Limiter block to ensure that the torque reference presented to the NN is below the maximum allowable torque command for a given flux-linkage limit. The Torque Limiter block is implemented through a small NN that is trained based on the maximum allowable torque data at the maximum current circle and the MTPV trajectory with respect to different flux limit lines (Figure 2.3). Then, the NN converts the torque reference to d- and q-axis current references that are presented to the motor current controller. The current controller outputs d- and q-axis control voltages to control the IPM motor based on the SVPWM. To ensure the amplitude of the control voltage generated by the current controller does not exceed the six-step voltage, a feedback flux weakening mechanism, as shown in [6], is adopted in Figure 2.9. The NN needs to be trained. The objective is that after the training, the NN can give the optimal d- and q-axis current references, in MTPA, flux-weakening I and II, and MTPV modes, for a desired torque demand and flux linkage limit under a specified motor speed and inverter DC voltage. Besides, to assure lifetime adaptivity and learning capabilities of both NNs, a cloud computing framework is proposed for future work to support efficient and reliable lifetime updates of both NNs, based on routine offline training over the cloud computing platform. Details about the NN training are shown in the following section.

The IPM drive and control system in Figure 2.9 has been considered in two IPM motor cases: one for simulation and one for a hardware experiment. The simulation case uses parameters of an IPM motor that are typical for an electric vehicle (EV) application [39]. The hardware experiment is based on a laboratory IPM motor [40], which has a smaller power rating

and is mainly used for the purpose of experimental validation. Table I shows the IPM motor parameters used in each case.

Table 1
IPM Data Used in Simulation/Experimental Study

Parameter	Simulation	Hardware	Units
Rated Power	100	1.6	kW
DC voltage	500	220	V
Maximum Speed	15000	3600	RPM
Permanent magnet flux	0.1266	0.20	Wb
Inductance in q-axis, L_q	0.59	27.11	mH
Inductance in d-axis, L_d	0.35	11.24	mH
Stator copper resistance, R_s	0.2	1.24	Ω
Inertia	0.095	0.00114	kg-m ²
Pole pairs	4	2	

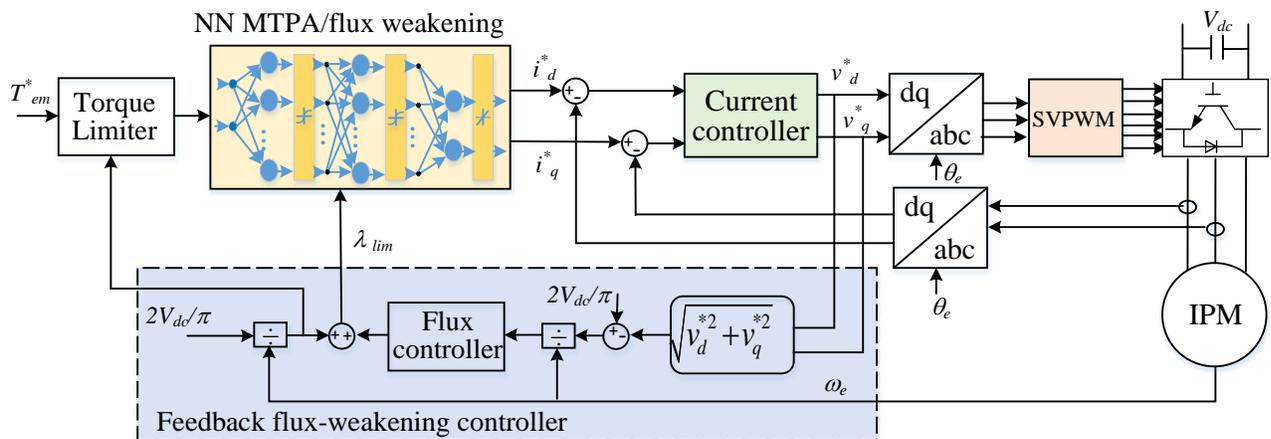


Figure 2.9: Overall IPM motor drive and control system

2.6 Simulation Evaluation

The simulation evaluation is based on a 3-phase 8-pole, 100 kW IPM motor from [31]. The simulation model was developed according to the IPM drive and control system shown in Figure 2.9 by using MATLAB Simscape Electrical [32]. The maximum current and DC link voltage of the motor are 450A and 500V, respectively. The motor parameter variations are built into the model [31]. Before the simulation evaluation, we first conducted a series of simulation experiments to obtain or “measure” the motor d- and q-axis flux-linkage LUTs, as shown in Figure 2.4, based on which we generated LUTs of motor d- and q-axis currents vs. torque and flux linkage limits, as shown in Figure 2.5, to be used as the conventional LUT approach. The size of the two LUTs is $25 \times 25 \times 2 = 1250$.

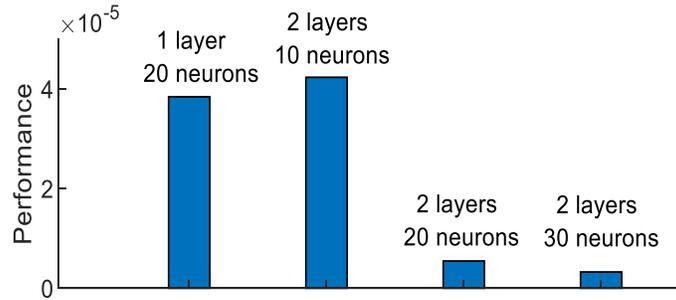


Figure 2.10: Training Error of different configuration

Then, to train the NN, it involves: 1) generating training data according to Section 2.4.A based on the motor d- and q-axis flux-linkage LUTs obtained above and 2) training the NN according to Section 2.4. In this paper, the torque command resolution is set to 1Nm for the full torque control range and the flux limit resolution is set to 0.001Wb. Thus, the size of the generated data set is around 5×10^4 . We conducted training experiments considering several NN configurations, in terms of the number of hidden layers and the number of nodes in the hidden layers. We trained each NN through multiple training experiments and selected the best-trained

one as the final NN. Each training experiment usually took about 10 to 20 minutes to complete on a PC with a 2.3GHz CPU and 16GB RAM. Figure 2.10 shows the errors of the trained NNs, from which the NN with 2 hidden layers and 10 nodes in each hidden layer is selected. The size of the NN is $3 \times 10 + 11 \times 10 + 11 \times 2 = 162$. After the NN is determined and obtained, the NN method is compared with the LUT-based method to determine MTPA, flux-weakening, and MTPV operating points of the simulation IPM motor.

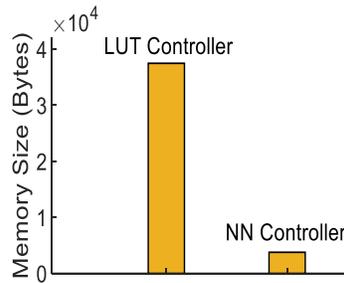


Figure 2.11: Real-time memory sizes of the NN vs. LUT

Now, some evident comparison results between the LUT and NN methods are as follows.

- 1) For the LUT method, its size and interpolation methods will affect the accuracy in determining the d- and q-axis reference currents, especially in the nonlinear areas (areas with high torque command and near the voltage boundary). Although a large LUT size can improve the output accuracy, it will increase more storage burden in the controller.
- 2) The proposed NN can be trained with a large number of data but only a much smaller memory is needed to store its weights and biases. Figure 2.11 shows a comparison of the DSP memory sizes in bytes for real-time implementation in storing the LUTs and NN weights by using dSPACE MicroLabBox [33].
- 3) The NN method is not only able to reduce the storage burden but also capture the nonlinear impact and increase the calculation accuracy. The real-time execution time for the NN-based

controller is about 4 μ s. More detailed comparisons between the LUT and NN methods are shown below.

2.6.1 Operation under base speed (MTPA)

Figure 2.12(a) shows a set of generated current results in the dq-current plane using the NN and LUT methods for the MTPA operation of the IPM under the based speed. The figure also shows the actual MTPA line to provide a baseline for the comparison. When the torque command is low, the generated d-q current lines follow the actual MTPA line closely using both the NN and LUT methods. However, when the torque command is higher, the generated d-q current lines reach the current limit of the motor, and the actual MTPA line shows nonlinearity due to the optimization solution and variable nature of motor parameters. As shown in Figure 2.12, the generated d-q current line is not able to follow the actual MTPA line when it is near to the current limit circle using the LUT method but can follow the actual MTPA line correctly using the proposed NN method.

Figure 2.12(b) to Figure 2.12(e) compare the LUT and NN methods in the closed-loop control environment as shown in Figure 2.9 via the EMT simulation, in which the speed of the IPM motor is constant at 2000rpm. The commanded torque is 216Nm, 221 Nm, 226 Nm, 231 Nm, and 236 Nm at 0ms, 25ms, 50ms, 75ms, and 100ms, respectively. The figure shows that at the same commanded torque, the reference current amplitude using the NN method is clearly smaller than that using the LUT method, further demonstrating the advantage of the proposed technique for the IPM motor operation in the MTPA mode in the EMT simulation environment.

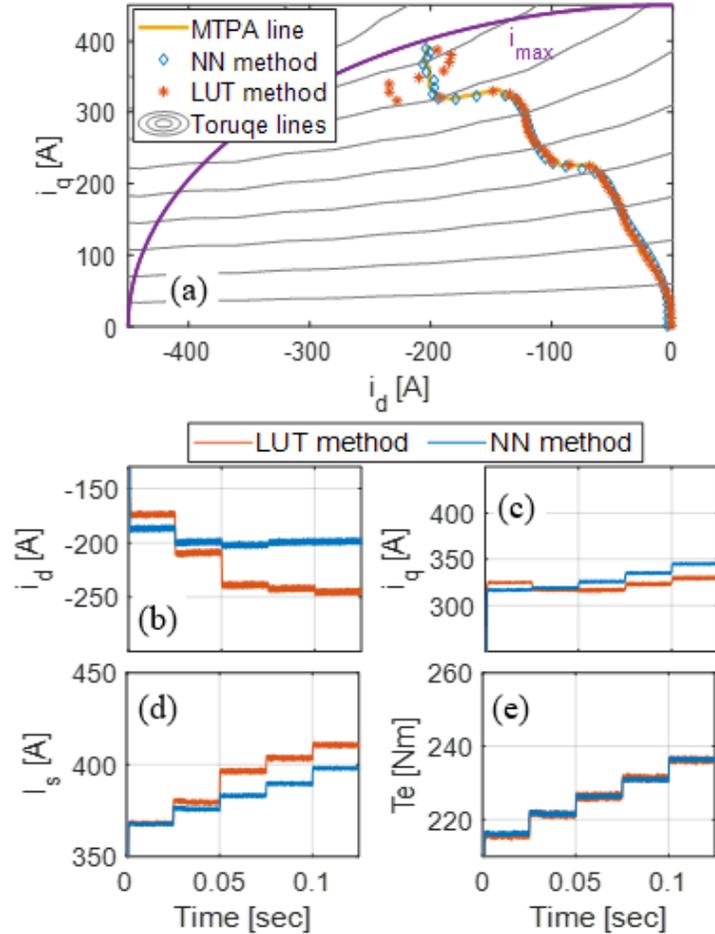


Figure 2.12: MTPA mode: (a) Comparison of actual MTPA line with the generated MTPA lines using LUT and NN methods, (b) d-axis current, (c) q-axis current, (d) current amplitude, (e) torque

2.6.2 Operation at high speed (flux-weakening)

Figure 2.13(a) shows the generated current results in the dq-current plane using the NN and LUT methods for the flux-weakening operation of the IPM at a high speed, in which the commanded torque is 101 Nm. As shown in the figure, with the increase of the motor speed, the operation enters the flux-weakening region, and the operating points are not on the MTPA line anymore but on the intersection of the flux-limit line and the equal torque line. From the figure, it can be seen that the generated dq-currents using the LUT method are farther away from the origin than those using the NN method, meaning that under the same demanded

torque, the motor current using the LUT method is larger than that using the proposed NN method.

Figure 2.13 (b) to Figure 2.13(e) show the EMT simulation using the LUT and NN methods, in which the commanded torque is 101 Nm and the speed initially is 8777rpm and changes to 8406rpm at 25ms and 8066rpm at 60ms. The torque produced by using the NN method is a little bit higher than that of the LUT method while the motor current amplitude using the NN method is smaller than that using the LUT method.

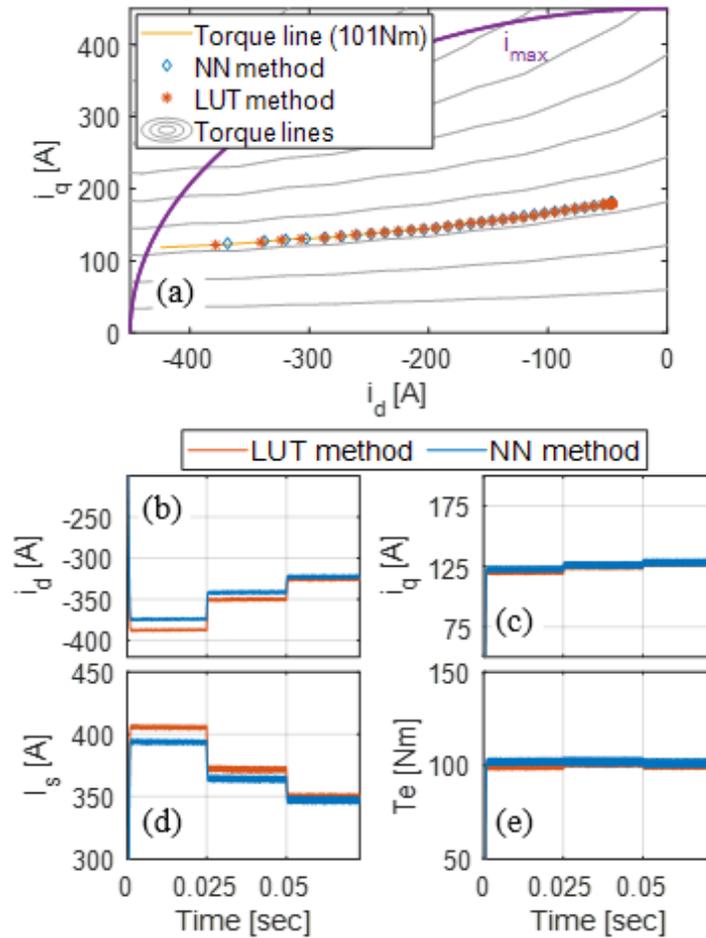


Figure 2.13: Flux-weakening mode: (a) Comparison of desired torque line with the generated torques using LUT and NN methods, (b) d-axis current, (c) q-axis current, (d) current amplitude, (e) torque

2.6.3 Operation at high speed (flux-weakening) Operation at extremely high speed (flux-weakening on MTPV)

Figure 2.14(a) demonstrates the generated current results in the dq-current plane using the NN and LUT methods as the IPM motor operates toward the MTPV mode at extremely high speed, in which the commanded torque is 78 Nm. As shown in the figure, when the speed keeps increasing, the commanded torque cannot be reached because of the voltage limit. It can be seen that the generated dq-currents using the LUT method cannot follow the MTPV line accurately in the nonlinear area.

Figure 2.14 (b) to Figure 2.14(e) show the EMT simulation using the LUT and NN methods, in which the commanded torque is 76 Nm and the speed initially is 11261rpm and changes to 11937rpm at 25ms, 12699rpm at 50ms, and 13564rpm at 75ms. The figures show that although the desired torque command cannot be reached as the speed keeps increasing, the average output torque using the NN method is larger than that using the LUT method, demonstrating that the NN method can better meet the torque demand in the MTPV mode.

2.6.4 Operation at variable and transient torque conditions

Figure 2.15 demonstrates the performance comparison using the NN and LUT methods under a variable and transient torque demand as shown in Figure 2.15(d). It was found that, in general, the difference between the LUT and NN methods depends strongly on how linear the dq-current over the torque and flux linkage characteristics is for a given torque demand region. For a torque demand in a nonlinear region, the NN is clearly better and more efficient than the LUT approach, which is consistent with the results shown in Figure 2.12, Figure 2.13, Figure 2.14

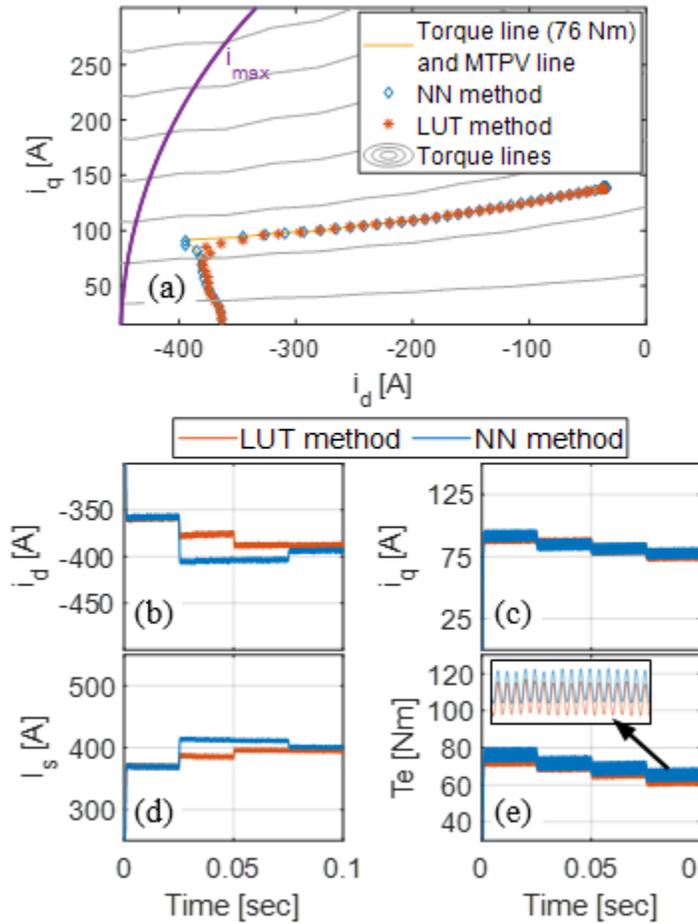


Figure 2.14: MTPV mode: (a) Comparison of desired torque line with the generated torques using LUT and NN methods, (b) d-axis current, (c) q-axis current, (d) current amplitude, (e) torque

2.6.5 Effect of parameter uncertainties

Figure 2.16 provides an evaluation of the parameter uncertainty impact on the NN-based MTPA, flux-weakening, and MTPV, in which a 10% deviation of the motor inductance is added to the actual motor inductance. From the figure, it can be seen that for the same torque demand, the parameter deviation causes a higher motor current amplitude for the same torque command. As a result, the torque control efficiency would be reduced a little but the stability of the IPM

motor drive is not affected. Certainly, this efficiency degradation can be improved after the NN is updated and retrained over the cloud computing platform as shown in Figure 2.9.

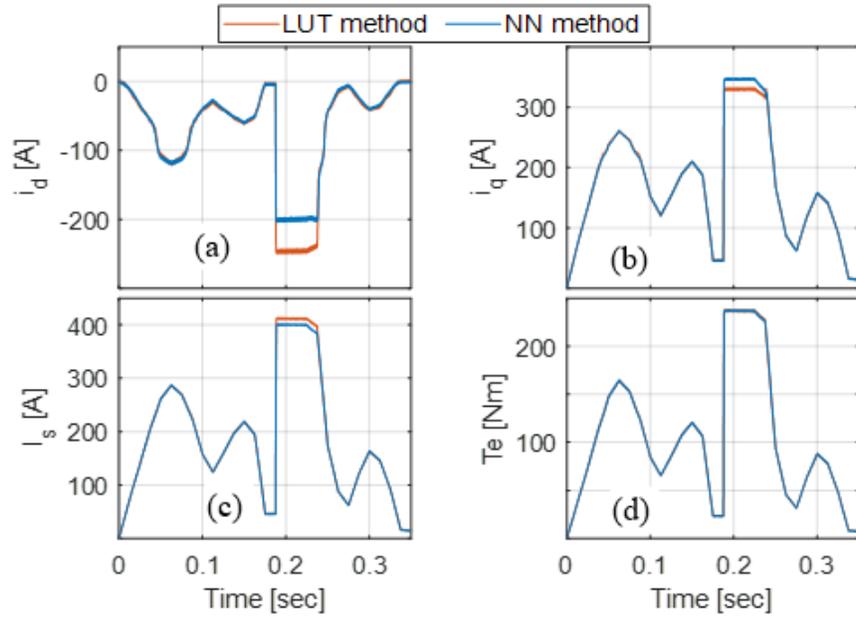


Figure 2.15: Variable and transient torque mode: (a) d-axis current, (b) q-axis current, (c) current amplitude, (d) torque

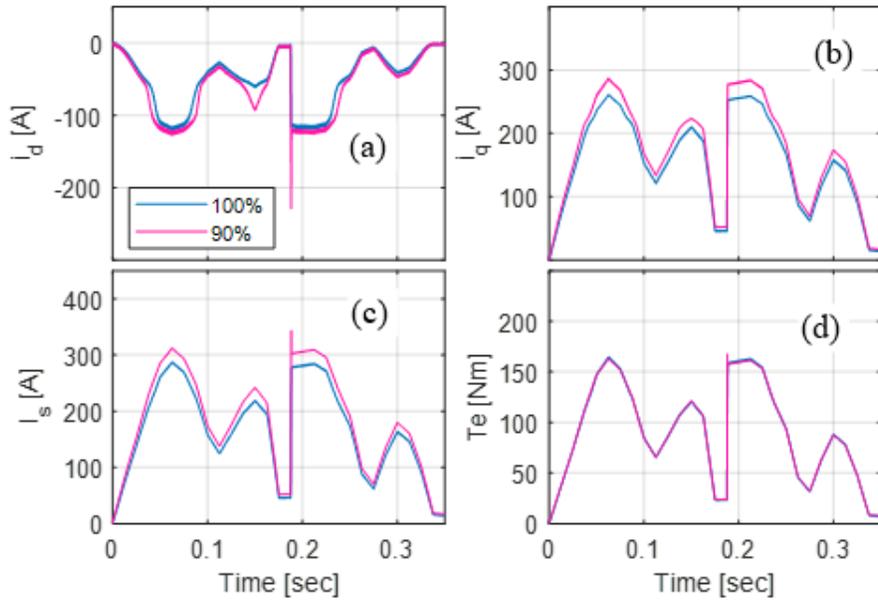


Figure 2.16: Impact of motor parameter uncertainty to NN-based MTPA, flux-weakening and MTPV: (a) d-axis current, (b) q-axis current, (c) current amplitude, (d) torque

2.7 Hardware Experiment

2.7.1 Hardware setup

To further validate the NN-based MTPA and flux-weakening controller, a DSP-based laboratory dyno system was built (Figure 2.17). The experimental setup (Figure 2.17b) comprises: (i) a dyno system containing a Kollmorgen IPM motor coupled to a FESTO LabVolt dynamometer; (ii) a 3-phase SEMIKRON IGBT inverter as the motor drive; (iii) a dSPACE MicroLabBox real-time controller [34] used to generate PWM control signals for controlling the IPM motor; (iv) an OP8660 sensor to collect measured voltage and current signals for the dSPACE real-time controller; (v) a MAGNA-POWER DC source as the power supply to the motor. An RC low-pass filter with a cutoff frequency of 1000rad/s was used after the voltage measurement from the OP8660 to remove the switching noise impact. The phase delay and amplitude attenuation caused by the RC filter was compensated and corrected in the dSPACE experiment setup system.

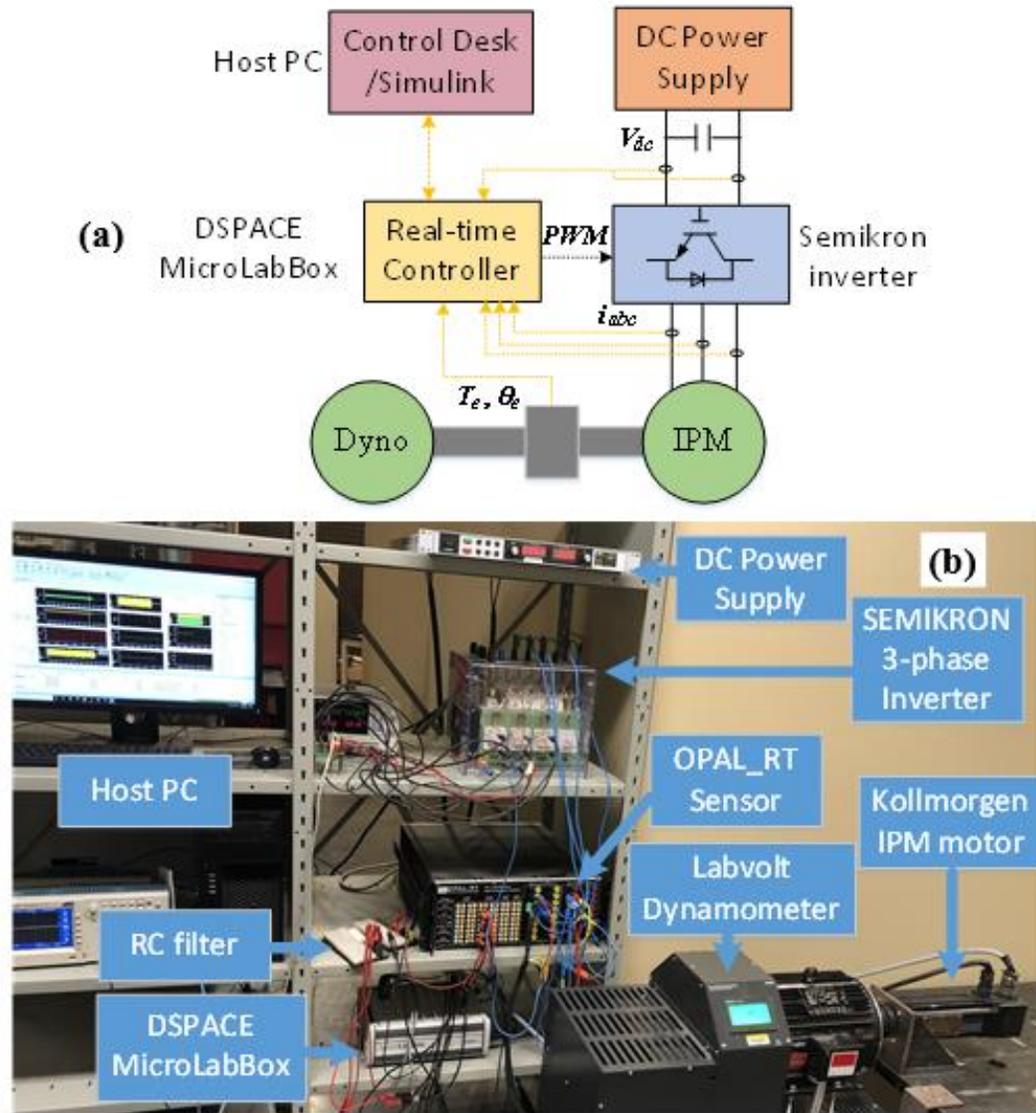


Figure 2.17: Experimental testing and control systems: (a) Circuit connection, (b) Experiment setup

2.7.2 Hardware results

The IPM motor parameters, including the LUTs of dq-axis flux linkages as illustrated in Section 2.2, were first characterized through a set of experiments [35]. Then, the conventional LUTs of motor d- and q-axis currents vs. torque and flux linkage limits were obtained, training data was generated, and the NN was trained according to Section 2.4. The current constraint of

the motor drive is 5A and the DC voltage constraint is 220V. After a well-trained NN was obtained, the NN was integrated with the motor current controller, as shown in *Figure 2.9*, in the experiment system to evaluate the performance of the NN- and LUT-based MTPA, flux-weakening, and MTPV control.

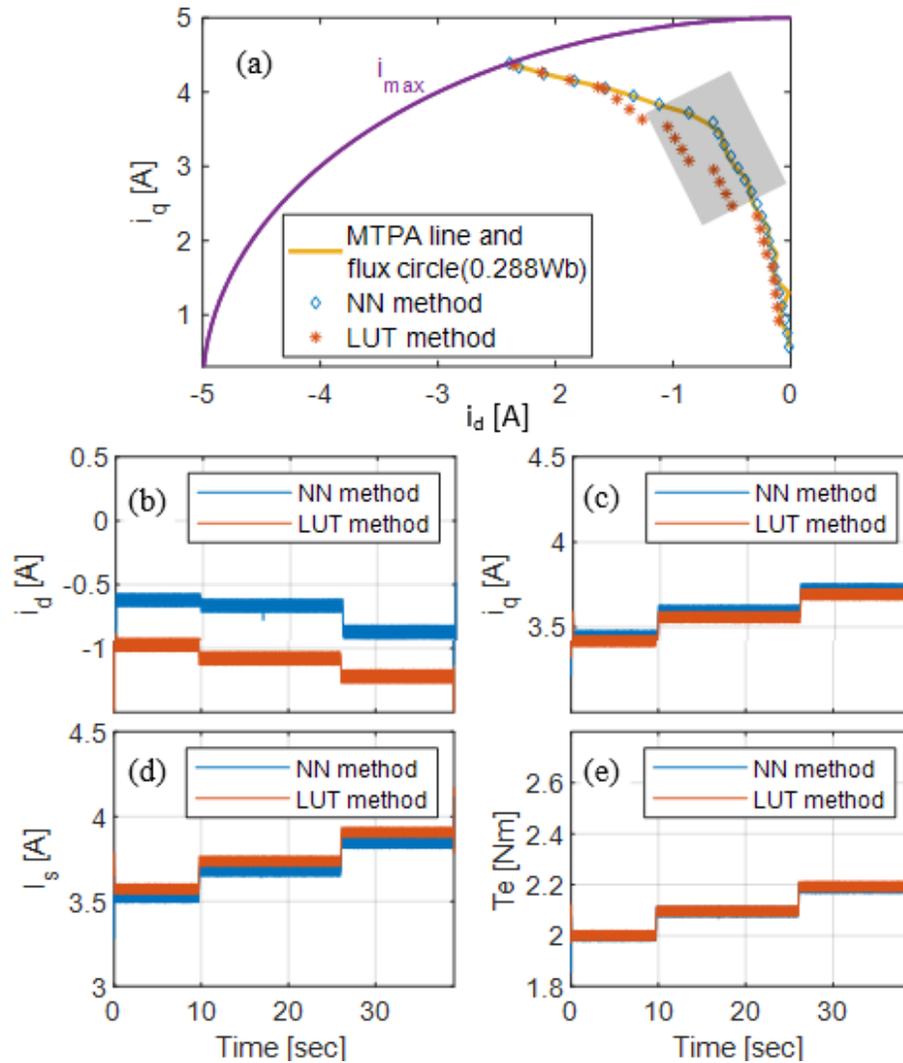


Figure 2.18: NN over LUT under MTPA mode: (a) actual vs. generated MTPA lines, (b) d-axis current, (c) q-axis current, (d) current amplitude, (e) torque

Figure 2.18 shows the measured current and torque results when the dyno speed was set as 300rpm. Figure 2.18a shows the contour plot comparison of the proposed NN and LUT

methods along the MTPA line under the flux limit of 0.288Wb corresponding to the motor speed of 300rpm and DC voltage of 200V. Figure 2.18b to 16e show the transient current and torque using the NN and LUT methods, in which the torque command was initially 2Nm and changed to 2.1Nm and 2.2Nm. In the experiment environment, there were more noises than those in the simulation environment (Section 2.6). As a result, the measured currents were processed through a low-pass filter so that the impact of noises can be eliminated for a better evaluation and comparison. The transient response shown in Figure 2.18b to Figure 2.18e corresponds to the shadow part in Figure 2.18a. From both the contour plot and transient response results, it can be seen that the NN-based controller generated more accurate optimal dq-current points (Figure 2.18a) and had a very good transient response. Although both the NN- and LUT-based controllers offered the same torque (Figure 2.18e), the motor current magnitude of the NN-based controller is clearly smaller than that using the LUT-based controller (Figure 2.18d).

Figure 2.19 shows the results for the IPM motor operating at higher speeds. The DC-link voltage was 150V and the torque command was 2.3Nm. The motor speed was initially 1680rpm and increased to 1700rpm. Figure 2.19a shows the contour plot comparison between the proposed NN and LUT methods along the equal torque line of 2.3Nm when the flux limit decreases (speed increases). Figure 2.19b to Figure 2.19e shows the transient current and torque using the NN and LUT methods corresponding to the shadow part of the contour plot in Figure 2.19a. From the results, it can be seen that the operation of the motor entered the deep flux-weakening range, and the output torque could not reach the torque command anymore as it gets to the rated current circle due to the high-speed operation of the motor under the low voltage limit. However, the NN-based controller had a very good transient response to the flux-weakening operation and was able to generate more accurate optimal current commands, which

means a higher torque in flux-weakening conditions than that of the LUT-based controller (Figure 2.19e).

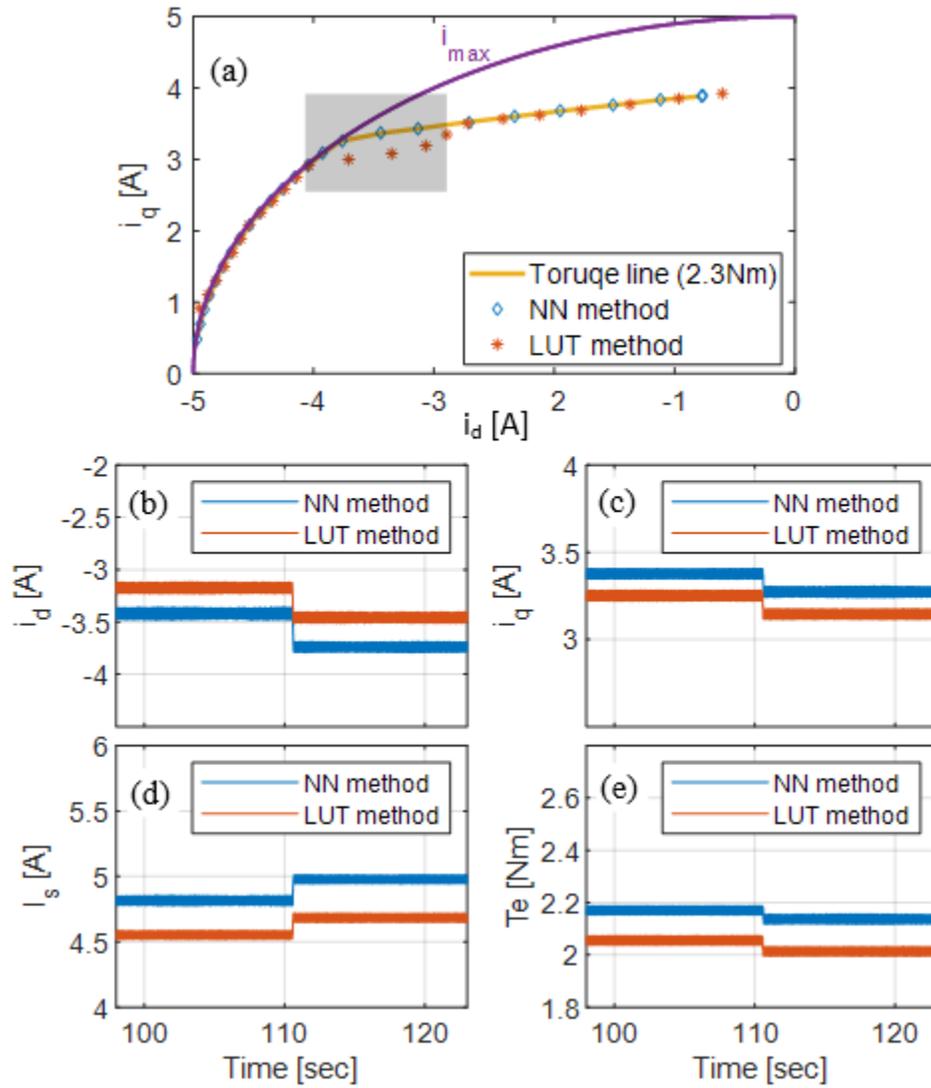


Figure 2.19: NN over LUT in flux weakening mode: (a) actual vs. generated torque lines, (b) d-axis current, (c) q-axis current, (d) current amplitude, (e) torque

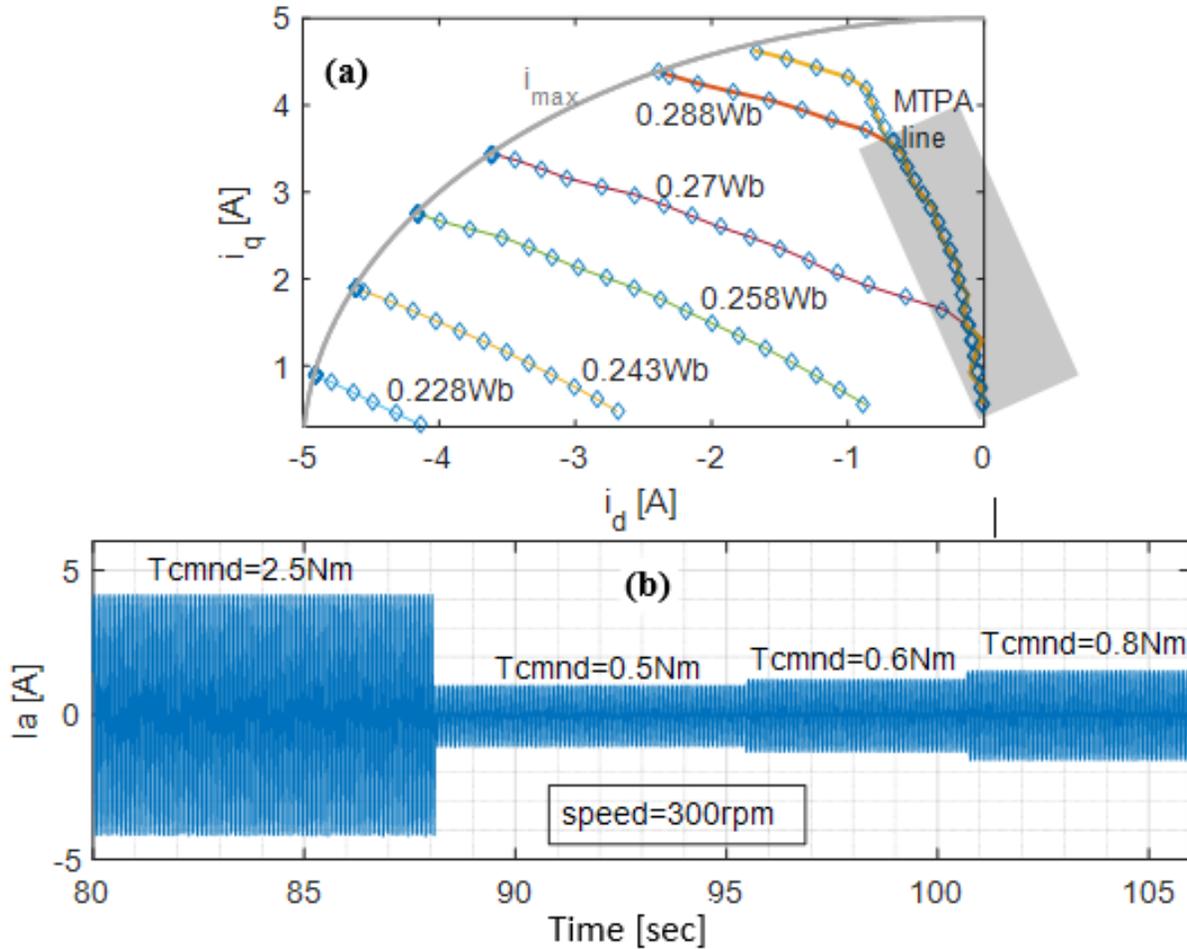


Figure 2.20: NN-based control under constant flux lines and MTPA mode: (a) actual vs. generated flux and MTPA lines, b) phase-a current

Figure 2.20a shows the contour plot of the dq-currents outputs generated by the NN method for each flux limit line (or circle) when the torque command increases. It can be seen from the figure that the current solutions shown in the contour plot, generated by using the NN method, follow the flux limit circles very well to generate the corresponding torque based on the torque command. Figure 2.20b shows the phase-a current response, corresponding to the shadow part of the contour plot in Fig. 18a under the MTPA operation, when torque command changes as [2.5 0.5 0.6 0.8] Nm under 300rpm. It can be seen from the figure that the NN-based MTPA control presents a very good dynamic response.

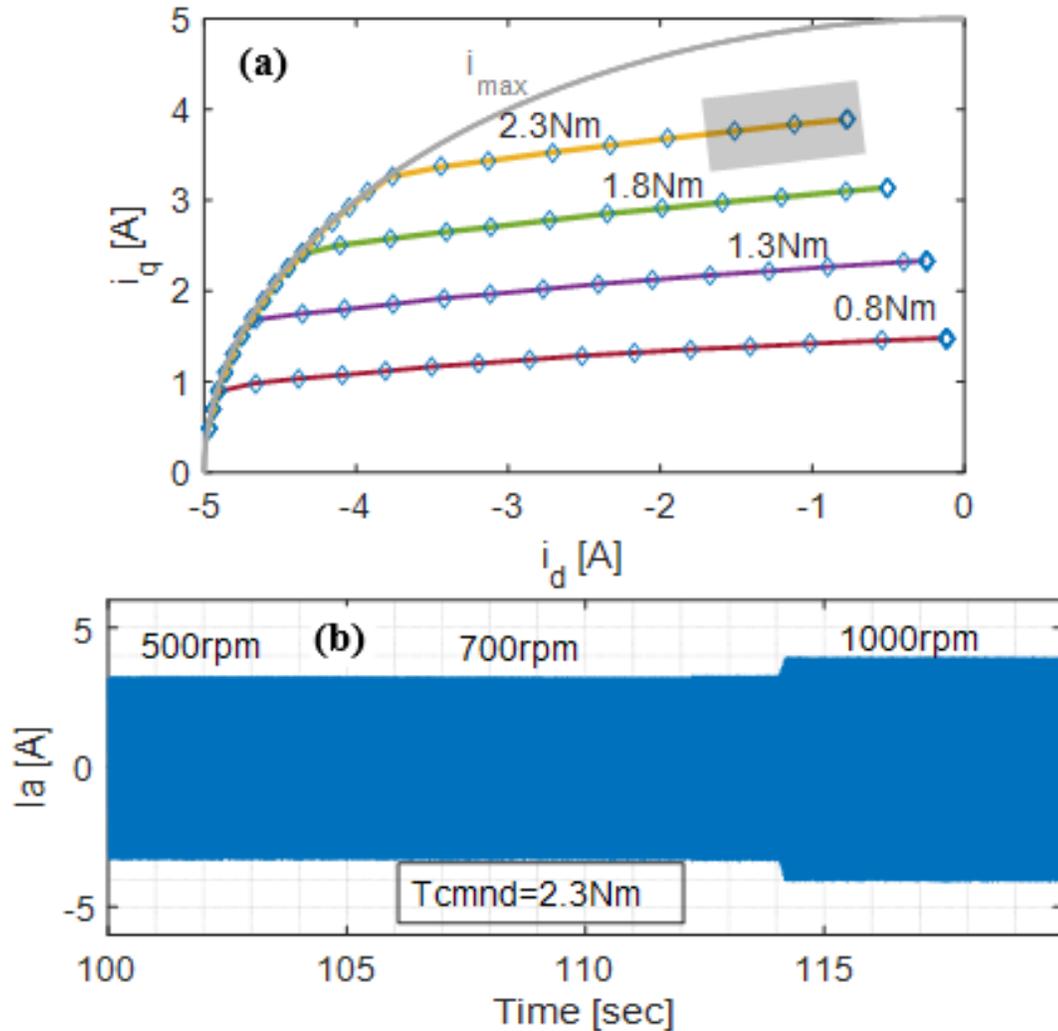


Figure 2.21: NN-based control under constant torque lines and maximum current circle: (a) actual vs. generated torque lines, b) phase-a current

Figure 2.21a shows the contour plot of the dq-currents outputs generated by the NN method for each equal torque line when the flux limit decreases (speed increases). It can be seen from the figure that the current solutions shown in the contour plot, generated by using the NN method, follow the equal torque lines very well to generate the maximum available torque and maintain the minimum current at the same time under the flux-weakening and MTPV operation modes. Figure 2.21b shows the phase-a current response, corresponding to the shadow part of the contour plot in Fig. 19a under the flux-weakening operation when speed changes from 500 rpm

to 1000rpm with a 2.3Nm torque command. It can be seen from the figure again that the NN-based control presents a very good dynamic response.

2.8 Conclusions

The efficient operation of an IPM motor is important and also challenging, particularly when considering nonlinear parameter variations of the motor. This paper presents a novel machine-learning strategy based on a neural network (NN) for the MTPA, flux-weakening, and MTPV control over the full speed range. Since the proposed NN controller is trained and generated offline, it has all the advantages that the LUT-based methods have, such as fast current reference generation, good stability, and dynamic performance. The proposed control structure is also very simple and does not have any convergence issues and high computational burden compared to the online-based methods.

At the same time, the proposed method overcomes the disadvantages of the LUT-based methods, such as 1) requirement for a large memory size to store the lookup tables and 2) limited control accuracy due to the LUT size and interpolation. The study shows that compared to the traditional methods, the proposed NN-based method requires a very small memory size, is much more accurate to get the optimal operating points than the LUT approaches, is able to capture the nonlinear impact caused by the optimization solution and variable motor parameters, and requires a lower computational cost. The hardware experiments further demonstrate the effectiveness of the NN method applicable to practical IPM motors.

To ensure the lifetime adaptivity and learning capabilities of the NN, a cloud-based offline training concept is proposed for future work, which will guaranty the safe and reliable

NN system development and assure high-performance operation of individual IPM motor over its lifetime.

2.9 References

- [1] X. Liu, H. Chen, J. Zhao, and A. Belahcen, "Research on the Performances and Parameters of Interior PMSM Used for Electric Vehicles," *IEEE Trans. Ind. Electron.*, vol. 63, no. 6, pp. 3533–3545, Jun. 2016, doi: 10.1109/TIE.2016.2524415.
- [2] S. Morimoto, M. Sanada, and Y. Takeda, "Wide-speed operation of interior permanent magnet synchronous motors with high-performance current regulator," *IEEE Trans. on Ind. Applicat.*, vol. 30, no. 4, pp. 920–926, Aug. 1994, doi: 10.1109/28.297908.
- [3] J. Lara, J. Xu, and A. Chandra, "Effects of Rotor Position Error in the Performance of Field Oriented Controlled PMSM Drives for Electric Vehicle Traction Applications," *IEEE Trans. Ind. Electron.*, pp. 1–1, 2016, doi: 10.1109/TIE.2016.2549983.
- [4] T. Windisch and W. Hofmann, "A Novel Approach to MTPA Tracking Control of AC Drives in Vehicle Propulsion Systems," *IEEE Trans. Veh. Technol.*, vol. 67, no. 10, pp. 9294–9302, Oct. 2018, doi: 10.1109/TVT.2018.2861083.
- [5] T. M. Jahns, "Flux-Weakening Regime Operation of an Interior Permanent-Magnet Synchronous Motor Drive," *IEEE Trans. on Ind. Applicat.*, vol. IA-23, no. 4, pp. 681–689, Jul. 1987, doi: 10.1109/TIA.1987.4504966.
- [6] T.-S. Kwon, G.-Y. Choi, M.-S. Kwak, and S.-K. Sul, "Novel Flux-Weakening Control of an IPMSM for Quasi-Six-Step Operation," *IEEE Trans. on Ind. Applicat.*, vol. 44, no. 6, pp. 1722–1731, 2008, doi: 10.1109/TIA.2008.2006305.
- [7] L. Sepulchre, M. Fadel, M. Pietrzak-David, and G. Porte, "MTPV Flux-Weakening Strategy for PMSM High Speed Drive," *IEEE Transactions on Industry Applications*, vol. 54, no. 6, pp. 6081–6089, Nov. 2018, doi: 10.1109/TIA.2018.2856841.
- [8] H. Ge, Y. Miao, B. Bilgin, B. Nahid-Mobarakeh, and A. Emadi, "Speed Range Extended Maximum Torque Per Ampere Control for PM Drives Considering Inverter and Motor Nonlinearities," *IEEE Trans. Power Electron.*, vol. 32, no. 9, pp. 7151–7159, Sep. 2017, doi: 10.1109/TPEL.2016.2630051.
- [9] H. W. de Kock, A. J. Rix, and M. J. Kamper, "Optimal Torque Control of Synchronous Machines Based on Finite-Element Analysis," *IEEE Transactions on Industrial Electronics*, vol. 57, no. 1, pp. 413–419, Jan. 2010, doi: 10.1109/TIE.2009.2030209.

- [10] Bing Cheng and T. R. Tesch, "Torque Feedforward Control Technique for Permanent-Magnet Synchronous Motors," *IEEE Trans. Ind. Electron.*, vol. 57, no. 3, pp. 969–974, Mar. 2010, doi: 10.1109/TIE.2009.2038951.
- [11] L. Ortombina, F. Tinazzi, and M. Zigliotto, "Adaptive Maximum Torque per Ampere Control of Synchronous Reluctance Motors by Radial Basis Function Networks," *IEEE J. Emerg. Sel. Topics Power Electron.*, vol. 7, no. 4, pp. 2531–2539, Dec. 2019, doi: 10.1109/JESTPE.2018.2858842.
- [12] H.-S. Kim, Y. Lee, S.-K. Sul, J. Yu, and J. Oh, "Online MTPA Control of IPMSM Based on Robust Numerical Optimization Technique," *IEEE Trans. on Ind. Applicat.*, vol. 55, no. 4, pp. 3736–3746, Jul. 2019, doi: 10.1109/TIA.2019.2904567.
- [13] Q. Liu and K. Hameyer, "High-Performance Adaptive Torque Control for an IPMSM With Real-Time MTPA Operation," *IEEE Trans. Energy Convers.*, vol. 32, no. 2, pp. 571–581, Jun. 2017, doi: 10.1109/TEC.2016.2633302.
- [14] K. D. Hoang and H. K. A. Aorith, "Online Control of IPMSM Drives for Traction Applications Considering Machine Parameter and Inverter Nonlinearities," *IEEE Trans. Transp. Electrific.*, vol. 1, no. 4, pp. 312–325, Dec. 2015, doi: 10.1109/TTE.2015.2477469.
- [15] S.-Y. Jung and K. Nam, "Current Minimizing Torque Control of the IPMSM Using Ferrari's Method," *IEEE TRANSACTIONS ON POWER ELECTRONICS*, vol. 28, no. 12, p. 15, 2013.
- [16] Z. Han, J. Liu, W. Yang, D. B. Pinhal, N. Reiland, and D. Gerling, "Improved Online Maximum-Torque-Per-Ampere Algorithm for Speed Controlled Interior Permanent Magnet Synchronous Machine," *IEEE Trans. Ind. Electron.*, vol. 67, no. 5, pp. 3398–3408, May 2020, doi: 10.1109/TIE.2019.2918471.
- [17] F.-J. Lin, Y.-T. Liu, and W.-A. Yu, "Power Perturbation Based MTPA With an Online Tuning Speed Controller for an IPMSM Drive System," *IEEE Trans. Ind. Electron.*, vol. 65, no. 5, pp. 3677–3687, May 2018, doi: 10.1109/TIE.2017.2762634.
- [18] Z. Li, G. Feng, C. Lai, W. Li, and N. C. Kar, "Machine Parameter-Independent Maximum Torque Per Ampere Control for Dual Three-Phase PMSMs," *IEEE Trans. Transp. Electrific.*, vol. 5, no. 4, pp. 1430–1440, Dec. 2019, doi: 10.1109/TTE.2019.2953656.
- [19] L. Ortombina, F. Tinazzi, and M. Zigliotto, "Magnetic Modeling of Synchronous Reluctance and Internal Permanent Magnet Motors Using Radial Basis Function Networks," *IEEE Trans. Ind. Electron.*, vol. 65, no. 2, pp. 1140–1148, Feb. 2018, doi: 10.1109/TIE.2017.2733502.
- [20] S. Li, H. Won, X. Fu, M. Fairbank, D. C. Wunsch, and E. Alonso, "Neural-Network Vector Controller for Permanent-Magnet Synchronous Motor Drives: Simulated and Hardware-

Validated Results,” IEEE Trans. Cybern., vol. 50, no. 7, pp. 3218–3230, Jul. 2020, doi: 10.1109/TCYB.2019.2897653.

- [21] L. Guo and L. Parsa, “Model Reference Adaptive Control of Five-Phase IPM Motors Based on Neural Network,” IEEE Transactions on Industrial Electronics, vol. 59, no. 3, pp. 1500–1508, Mar. 2012, doi: 10.1109/TIE.2011.2163371.
- [22] F.-J. Lin, M.-S. Huang, S.-G. Chen, and C.-W. Hsu, “Intelligent Maximum Torque per Ampere Tracking Control of Synchronous Reluctance Motor Using Recurrent Legendre Fuzzy Neural Network,” IEEE Trans. Power Electron., vol. 34, no. 12, pp. 12080–12094, Dec. 2019, doi: 10.1109/TPEL.2019.2906664.
- [23] J. Chen, J. Li, and R. Qu, “Maximum-Torque-per-Ampere and Magnetization-State Control of a Variable-Flux Permanent Magnet Machine,” IEEE Trans. Ind. Electron., vol. 65, no. 2, pp. 1158–1169, Feb. 2018, doi: 10.1109/TIE.2017.2733494.
- [24] M.-S. Wang, M.-F. Hsieh, Y.-S. Kung, and G. T. Lin, “Maximum torque per ampere control of IPMSM drive by fuzzy logic,” Microsyst Technol, vol. 24, no. 1, pp. 19–26, Jan. 2018, doi: 10.1007/s00542-016-3119-5.
- [25] M. Nasir Uddin and J. Khastoo, “Fuzzy Logic-Based Efficiency Optimization and High Dynamic Performance of IPMSM Drive System in Both Transient and Steady-State Conditions,” IEEE Trans. on Ind. Applicat., vol. 50, no. 6, pp. 4251–4259, Nov. 2014, doi: 10.1109/TIA.2014.2317845.
- [26] M. N. Uddin and R. S. Rebeiro, “Online Efficiency Optimization of a Fuzzy-Logic-Controller-Based IPMSM Drive,” IEEE Trans. on Ind. Applicat., vol. 47, no. 2, pp. 1043–1050, Mar. 2011, doi: 10.1109/TIA.2010.2103293.
- [27] L. Griffin, F. Fleming, and C. S. Edrington, “A particle swarm optimization based maximum torque per ampere control for a switched reluctance motor,” in IECON 2014 - 40th Annual Conference of the IEEE Industrial Electronics Society, Dallas, TX, USA, Oct. 2014, pp. 343–348, doi: 10.1109/IECON.2014.7048522.
- [28] A. M. A. Amin, M. I. El Korfally, A. A. Sayed, and O. T. M. Hegazy, “Efficiency Optimization of Two-Asymmetrical-Winding Induction Motor Based on Swarm Intelligence,” IEEE Trans. Energy Convers., vol. 24, no. 1, pp. 12–20, Mar. 2009, doi: 10.1109/TEC.2008.2011831.
- [29] Y. Sun, S. Li, X. Fu, W. Dong, M. Ramezani, and B. Balasubramanian, “Approximate Dynamic Programming Vector Controllers for Operation of IPM Motors in Linear and Over-modulation Regions,” IEEE Transactions on Transportation Electrification, pp. 1–1, 2020, doi: 10.1109/TTE.2020.3034044.

- [30] M. T. Hagan and M. B. Menhaj, "Training feedforward networks with the Marquardt algorithm," *IEEE Trans. Neural Netw.*, vol. 5, no. 6, pp. 989–993, Nov. 1994, doi: 10.1109/72.329697.
- [31] "JMAG-RT Model Library," *Simulation Technology for Electromechanical Design : JMAG*. <https://www.jmag-international.com/modellibrary/> (accessed Oct. 24, 2020).
- [32] "Simscape." <https://www.mathworks.com/products/simscape.html> (accessed Feb. 22, 2021).
- [33] "FAQ 269: Determining the Size of a Real-Time Application - dSPACE." <https://www.dspace.com/en/inc/home/support/kb/faqs/faq269.cfm> (accessed Feb. 22, 2021).
- [34] "MicroLabBox Hardware." <https://www.dspace.com/en/inc/home/products/hw/microlabbox.cfm> (accessed Feb. 22, 2021).
- [35] E. Armando, R. I. Bojoi, P. Guglielmi, G. Pellegrino, and M. Pastorelli, "Experimental Identification of the Magnetic Model of Synchronous Machines," *IEEE Trans. on Ind. Applicat.*, vol. 49, no. 5, pp. 2116–2125, Sep. 2013, doi: 10.1109/TIA.2013.2258876.

3. IDENTIFICATION OF IPM MOTOR MAGNETIC MODEL USING NEURAL NETWORK IN A CLOUD COMPUTING FRAMEWORK

3.1 Introduction

Electric Vehicles (EVs) have drawn attention of researchers for years due to their advantages such as high energy efficiency, reduction of harmful pollutants and greenhouse gases. Electric machines and power electronics are important elements in EV powertrains. Permanent-magnet synchronous machines (PMSM) are good candidates for the EV application because of their high-power density, high efficiency, and high torque density. Interior permanent magnet (IPM) synchronous machines are widely adopted because of the safety design and saliency features over a wide speed operation range. IPM machine modeling and parameter estimation are important for the controller design of high-efficient and high-performance motor drives. The IPM model can be used in several control algorithms, such as MTPA (maximum torque per ampere) and MTPV (maximum torque per volt) [1]–[3], current feedforward control, torque control, and torque estimation. The accurate machine model is also important for motor performance evaluation [4]–[7].

The IPM magnetic models are usually expressed in the dq- axis reference frame, and the dq-axis flux linkages are usually difficult to determine because of the magnetic saturation and cross-saturation [8]–[10]. Traditionally, the dq-axis flux linkages are models as 2D functions of the dq-axis magnetizing currents [2], [9]–[11]. However, in order to generate accurate magnetic model, the iron loss and copper loss are necessary to be considered. The copper loss is related to

the stator resistance that is affected by the temperature, and the iron loss is difficult to estimate due to its nonlinearity with varying machine speed, currents, and temperature [4], [9], [10], [12]–[14].

There are many methods proposed previously for IPM magnetic model estimation [9-17], and they can be divided into online-method and offline method. The offline method [15] is an analytical method based on finite-element analysis and the equivalent magnetic circuits. These methods are dependent on the machine design data which may not be accurate for the real electric machine after manufacturing. The method is based on the experiment measurement and test, in which the motor is typically operated at a constant speed and the dq-axis currents are controlled over a full current operating range [9], [10]. During the experiment, the dq-axis currents, dq-axis voltages, and the motor speed are measured at steady-state, and the motor flux linkages are calculated based on the steady-state dq-axis voltage equations. The voltages need to be processed using analog filters or FFT method to eliminate the high frequency harmonics. The machine flux linkages finally are obtained as 2D functions of the dq currents, usually via two look-up tables (LUTs) for the motor magnetic modeling. However, the offline generated LUTs are fixed and cannot deal with the manufacturing mismatch, parameter changes under different online conditions, aging issues, and temperature variations. The accuracy of the model and the parameters are also affected by the size of the LUTs and their interpolation methods. Another issue with this procedure is that the iron loss is not actually considered. The experiment test is usually conducted at a relatively low speed to avoid the effect of the iron loss. But, during the high-speed operation, the measured currents are affected by the iron loss and the real magnetizing currents remain unknown. This will lead to the inaccuracy results using the 2D LUTs.

Several online methods [5], [16]–[20] are proposed to overcome some of the issues. In [16], a recursive least-square (RLS) algorithm is used for online estimation of double-star IPM machine parameters, in which the flux linkages generated by using the motor current model is compared with the flux linkages generated by using the motor voltage model to improve the estimation accuracy of the motor inductance and permanent flux linkage used in the current model and thus the current model is adapted during the online operating. In [17], two affine projection algorithms are proposed for the motor parameter estimation and the two algorithms are designed to have different convergence speeds to enhance the estimation of different motor parameters. In [5], a Gauss-Newton iteration algorithm is proposed to estimate the flux linkages and the inductances of a SPMSM. The flux linkages are tuned and estimated based on the dq-currents, electrical speed, and the estimated back-EMF. Several AI-based methods such as artificial neural networks (NNs) are also used for the machine modeling [18]–[20]. In [19], the estimation model is considered as a two-dimensional radial based function (RBF) NN. Similar as [19], an NN-based magnetic field modeling of a switched reluctance motor is proposed in [18]. Both the two NN models are trained offline based on measured data that is obtained under a constant and low motor operating speed and they cannot handle parameter changes during the operation and other uncertainties. In [20], an online training algorithm for an NN-based model is proposed. However, like other online algorithms, the stability, ill-convergence, and computational burden can be a big issue during the operation. In addition, most of these methods use dq voltage generated from the controllers as a replacement of actual voltage measurement, which can cause accuracy issues for the motor modeling. At the same time, in [9-20], the iron loss is not considered, and the flux-linkage estimation can be inaccurate especially during the high-speed operation of the motor in the EV application.

To overcome the challenges of the existing methods, this paper proposes a novel strategy for identification of IPM motor magnetic model. The novel contributions include: 1) an offline-trained NN for IPM motor magnetic model identification that is robust and reliable for its implementation in real-life conditions, 2) a cloud computing based NN training and updating mechanism to assure the life-time adaptivity and learning capacity of the NN, 3) an NN structure design that enables the iron loss impact to be learned by the NN through training, and 4) a specific data collection strategy suitable for training of a laboratory test motor and for training of an actual motor over the cloud system. The performance of the NN-based model identification technique is validated via simulation and hardware experiments in various aspects.

3.2 IPM motor model with/without core loss

The model of an IPM motor is classically developed based on well-known Park-Clarke transformation. Traditionally, the core loss is not considered in the model development. However, as the operating speed of the motor increases, the core loss becomes a significant factor that needs to be considered.

3.2.1 IPM motor Model without core loss

Without the consideration of the core loss, the stator dq flux linkages of an IPM motor are described by:

$$\begin{bmatrix} \lambda_d \\ \lambda_q \end{bmatrix} = \begin{bmatrix} L_d & 0 \\ 0 & L_q \end{bmatrix} \begin{bmatrix} i_d \\ i_q \end{bmatrix} + \begin{bmatrix} \lambda_{pm} \\ 0 \end{bmatrix} \quad (3-1)$$

where λ_d , λ_q , L_d , L_q , i_d and i_q are stator d- and q-axis flux linkages, inductances, and currents and λ_{pm} is the flux linkage of the rotor permanent-magnet. The stator dq voltage equation is:

$$\begin{bmatrix} v_d \\ v_q \end{bmatrix} = R \begin{bmatrix} i_d \\ i_q \end{bmatrix} + \frac{d}{dt} \begin{bmatrix} L_d & 0 \\ 0 & L_q \end{bmatrix} \begin{bmatrix} i_d \\ i_q \end{bmatrix} + \omega_e \begin{bmatrix} -\lambda_q \\ \lambda_d \end{bmatrix} \quad (3-2)$$

where v_d and v_q are d- and q-axis stator voltages, and ω_e is the motor electrical speed. The steady-state voltage equation is

$$v_d = Ri_d - \omega_e \lambda_q = Ri_d - \omega_e L_q i_q \quad (3-3)$$

$$v_q = Ri_q + \omega_e \lambda_d = Ri_q + \omega_e (L_d i_d + \lambda_{pm}) \quad (3-4)$$

The electromagnetic torque of the motor can be written as:

$$T_{em} = \frac{p}{2} (\lambda_d i_q - \lambda_q i_d) = \frac{p}{2} [\lambda_{pm} - (L_d - L_q) i_d] i_q \quad (3-5)$$

where p is the number of the poles. From (3-1) to (3-4), an equivalent circuit of an IPM motor in d-q reference frame without the iron loss is obtained as shown in Figure 3.1, the flux linkages λ_d and λ_q can be obtained from the measurements of the stator dq voltages and currents.

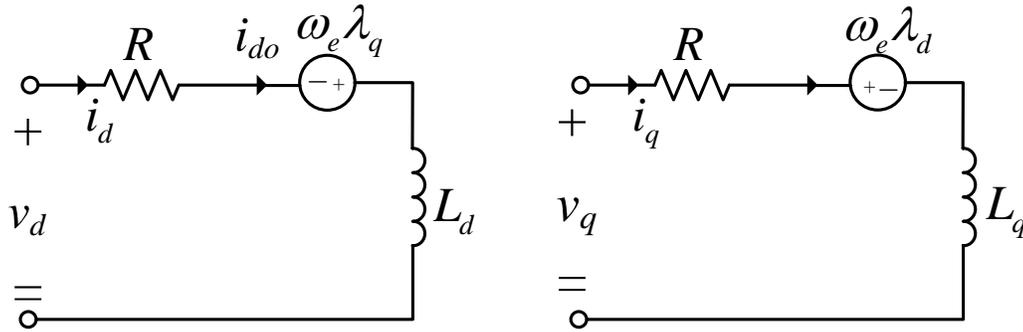


Figure 3.1: Equivalent circuit of an IPM motor without the consideration of the core loss: a) d-axis equivalent circuit, b) q-axis equivalent circuit

3.2.2 IPM motor Model with core loss

As the motor speed increases, the impact of the induced voltages, $\omega_e \lambda_q$ and $\omega_e \lambda_d$, in the d- and q-axis loops, respectively, becomes evident and cannot be neglected. This impact is complicated and typically involves magnetic saturation and cross coupling effects on the d-q axis flux-linkage. Usually, the impact is modeled as the core loss of the motor by adding a variable resistance R_C into Figure 3.1 as shown in Figure 3.2.

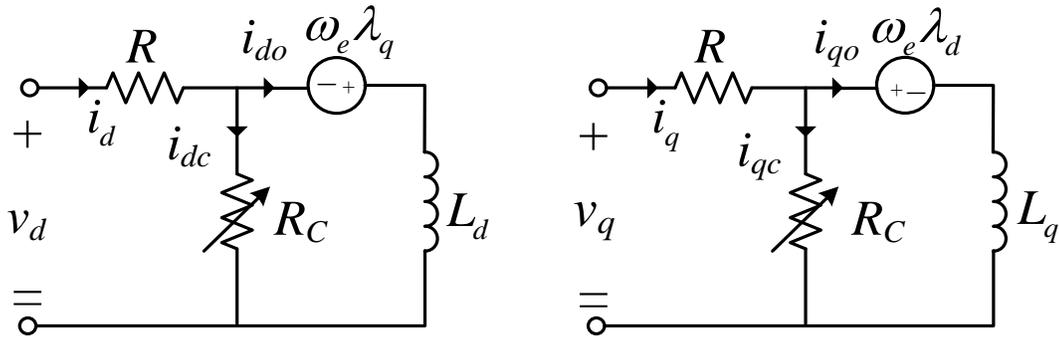


Figure 3.2: Equivalent circuit of an IPM motor with the consideration of the core loss: a) d-axis equivalent circuit, b) q-axis equivalent circuit

In Figure 3.2, i_{dc} and i_{qc} are core loss components and i_{do} and i_{qo} are torque producing components of i_d and i_q . Therefore, the d- and q-axis flux linkages are described by:

$$\begin{bmatrix} \lambda_d \\ \lambda_q \end{bmatrix} = \begin{bmatrix} L_d & 0 \\ 0 & L_q \end{bmatrix} \begin{bmatrix} i_{do} \\ i_{qo} \end{bmatrix} + \begin{bmatrix} \lambda_{pm} \\ 0 \end{bmatrix} \quad (3-6)$$

The steady-state voltage equation is

$$v_d = R i_d - \omega_e \lambda_q = R i_d - \omega_e L_q i_{qo} \quad (3-7)$$

$$v_q = R i_q + \omega_e \lambda_d = R i_q + \omega_e (L_d i_{do} + \lambda_{pm}) \quad (3-8)$$

The transient stator dq voltage equation is:

$$\begin{bmatrix} v_d \\ v_q \end{bmatrix} = R \begin{bmatrix} i_d \\ i_q \end{bmatrix} + \frac{d}{dt} \begin{bmatrix} L_d & 0 \\ 0 & L_q \end{bmatrix} \begin{bmatrix} i_{do} \\ i_{qo} \end{bmatrix} + \omega_e \begin{bmatrix} -\lambda_q \\ \lambda_d \end{bmatrix} \quad (3-9)$$

The electromagnetic torque of the motor can be written as:

$$T_{em} = \frac{P}{2} (\lambda_d i_{qo} - \lambda_q i_{do}) = \frac{P}{2} [\lambda_{pm} - (L_d - L_q) i_{do}] i_{qo} \quad (3-10)$$

In the equivalent circuit model and (5) to (8), the d-q flux linkages are usually described as the function of both d- and q- magnetizing currents, $\lambda_d(i_{do}, i_{qo})$ and $\lambda_q(i_{do}, i_{qo})$. From the above analysis, it can be seen that the IPM motor model can be very complex due to the saturation, cross-magnetization, varying core loss, copper loss, time variation, etc.

3.3 Identification of IPM motor parameters using neural network in a cloud computing framework

3.3.1 Proposed NN-based parameter identification

An NN, with multiple neurons and layers, has an ability to solve complex problems through learning, which offers a potential for improved parameter identification for an IPM motor. According to Section II, the stator flux linkages of an IPM motor depend on the torque producing currents i_{do} and i_{qo} that are not only related to the stator d- and q-axis currents but also affected by the core currents (Figure 3.2). However, the determination of the core currents is complicated. An important factor to influence the core currents is the motor operating speed. Based on these evaluations, the motor stator dq currents, i_d and i_q , and electrical speed ω_e are adopted in this paper as inputs of the NN. The NN has two hidden layers and an output layer as

shown in Figure 3.3. The two outputs at the output layer are the estimated dq flux-linkages, λ_d and λ_q .

In the first hidden layer, the output of the s^{th} neuron can be expressed as

$$n_s^{L1} = w_{s,id}^{L1} \cdot x_{id} + w_{s,iq}^{L1} \cdot x_{iq} + w_{s,\omega}^{L1} \cdot x_\omega + b_s^{L1} \quad \forall s \quad (3-11)$$

where x_{id} , x_{iq} and x_ω are the measured dq-axis currents and the electrical speed after

preprocessing, $w_{s,id}^{L1}$, $w_{s,iq}^{L1}$ and $w_{s,\omega}^{L1}$ are the input weights corresponding to x_{id} , x_{iq} and x_ω . b_s^{L1} is the

bias for the s^{th} neuron in the first hidden layer. Then, the hyperbolic tangent sigmoid transfer

functions are adopted for the neuron outputs n_s^{L1}

$$a_s^{L1} = \frac{2}{1 + \exp(-2 \cdot n_s^{L1})} - 1 \quad \forall s \quad (3-12)$$

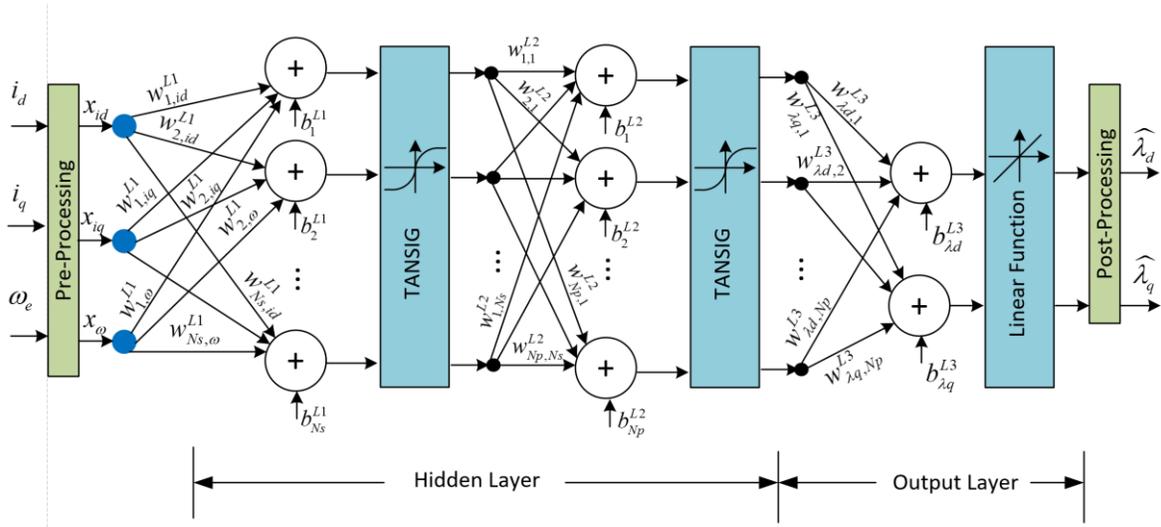


Figure 3.3: NN architecture used to determine IPM motor control in MTPA, flux-weakening, and MTPV

In the second hidden layer, the p^{th} neuron output is

$$m_p^{L2} = \sum_{s=1}^{N_s} w_{p,s}^{L2} a_s^{L1} + b_p^{L2} \quad \forall s, \forall p \quad (3-13)$$

where N_s is the number of neurons of the first hidden layer, $w_{p,s}^{L2}$ and b_p^{L2} are the weights and bias for the p^{th} neuron of the second hidden layer. Similarly, the sigmoid functions are adopted for the neuron outputs m_p^{L2} of the second hidden layer

$$a_p^{L2} = \frac{2}{1 + \exp(-2 \cdot m_p^{L2})} - 1 \quad \forall s, \forall p \quad (3-14)$$

Finally, at the output layer, the normalized reference currents $y_{\lambda d}$ and $y_{\lambda q}$ are generated as follows

$$y_{\lambda d} = \sum_{p=1}^{N_p} w_{\lambda d,p}^{L3} a_p^{L2} + b_{\lambda d}^{L3} \quad (3-15)$$

$$y_{\lambda q} = \sum_{p=1}^{N_p} w_{\lambda q,p}^{L3} a_p^{L2} + b_{\lambda q}^{L3} \quad (3-16)$$

where N_p is the number of neurons of the second hidden layer, $w_{\lambda d,p}^{L3}$, $w_{\lambda q,p}^{L3}$, $b_{\lambda d}^{L3}$ and $b_{\lambda q}^{L3}$ are the weights the bias of the output layer.

In order to have an efficient training and better performance, the dq-axis currents, speed, and the estimated flux-linkages are normalized via pre- and post-processing by

$$x_{id} = 2 \cdot \frac{i_{sd} - \min(\mathbf{i}_{sd})}{\max(\mathbf{i}_{sd}) - \min(\mathbf{i}_{sd})} - 1 \quad (3-17)$$

$$x_{iq} = 2 \cdot \frac{i_{sq} - \min(\mathbf{i}_{sq})}{\max(\mathbf{i}_{sq}) - \min(\mathbf{i}_{sq})} - 1 \quad (3-18)$$

$$x_{\omega} = 2 \cdot \frac{\omega_e - \min(\mathbf{\omega}_e)}{\max(\mathbf{\omega}_e) - \min(\mathbf{\omega}_e)} - 1 \quad (3-19)$$

$$\lambda_d = \frac{(y_{\lambda_d} + 1)(\max(\lambda_d) - \min(\lambda_d))}{2} + \min(\lambda_d) \quad (3-20)$$

$$\lambda_q = \frac{(y_{\lambda_q} + 1)(\max(\lambda_q) - \min(\lambda_q))}{2} + \min(\lambda_q) \quad (3-21)$$

where i_d , i_q , ω_e , λ_d and λ_q are the d-axis current input, q-axis current input, motor electrical speed, estimated d-axis flux-linkage, and estimated q-axis flux-linkage, respectively; λ_d and λ_q , and \mathbf{i}_{sq} , \mathbf{i}_{sd} , and $\mathbf{\omega}_e$, represent the training data set of all the desired d- and q-axis flux-linkages and corresponding d- and q-axis currents, and electrical speed, respectively. These input-output data samples are collected and generated as detailed in Section IV-A shown below and the NN is trained offline (section IV-B) to overcome the challenges of the online NN training (Section 3.1).

3.3.2 Proposed NN in a cloud computing framework

An issue associated with an NN trained offline is how to assure lifetime adaptivity and learning capabilities of the NN for the parameter identification as an IPM motor changes from new to old over time. On the other side, even for the same motor type having the same power rating, the parameters of two individual IPM motors could deviate notably due to the mass

production impact. To overcome the challenges, we propose a training system based on a cloud computing framework that can support a highly efficient and reliable lifetime update of the NN based on offline training over the cloud computing platform as shown in Figure 3.4. Basically, for a new IPM motor, the NN for the parameter identification is developed based on the measured data of a sample test motor of the same motor type and will be used by all the motors with the same motor type for the motor parameter identification. However, after an individual motor is put into real-life operation, actual motor operation data will be collected and transmitted to a remote cloud device. Based on the newly obtained motor data, the NN will be retrained and updated if necessary and the trained NN weights will be transmitted back to the EV computer to replace the previous NN weights. Such an offline learning strategy based upon a cloud computing platform will guaranty the safe and reliable NN system development that cannot be achieved for an online NN learning system, which will assure the high-performance and adaptive operation of each individual EV motor over its lifetime. Details about the cloud-based NN training are shown in the following section.

The IPM drive and control system in Figure 3.4 has been considered in two IPM motor cases: one for simulation and one for a hardware experiment. The simulation case uses parameters of an IPM motor that are typical for an electric vehicle (EV) application. The hardware experiment is based on a laboratory IPM motor, which has a smaller power rating and is mainly used for the purpose of experimental validation. Table I shows the IPM motor parameters used in each case.

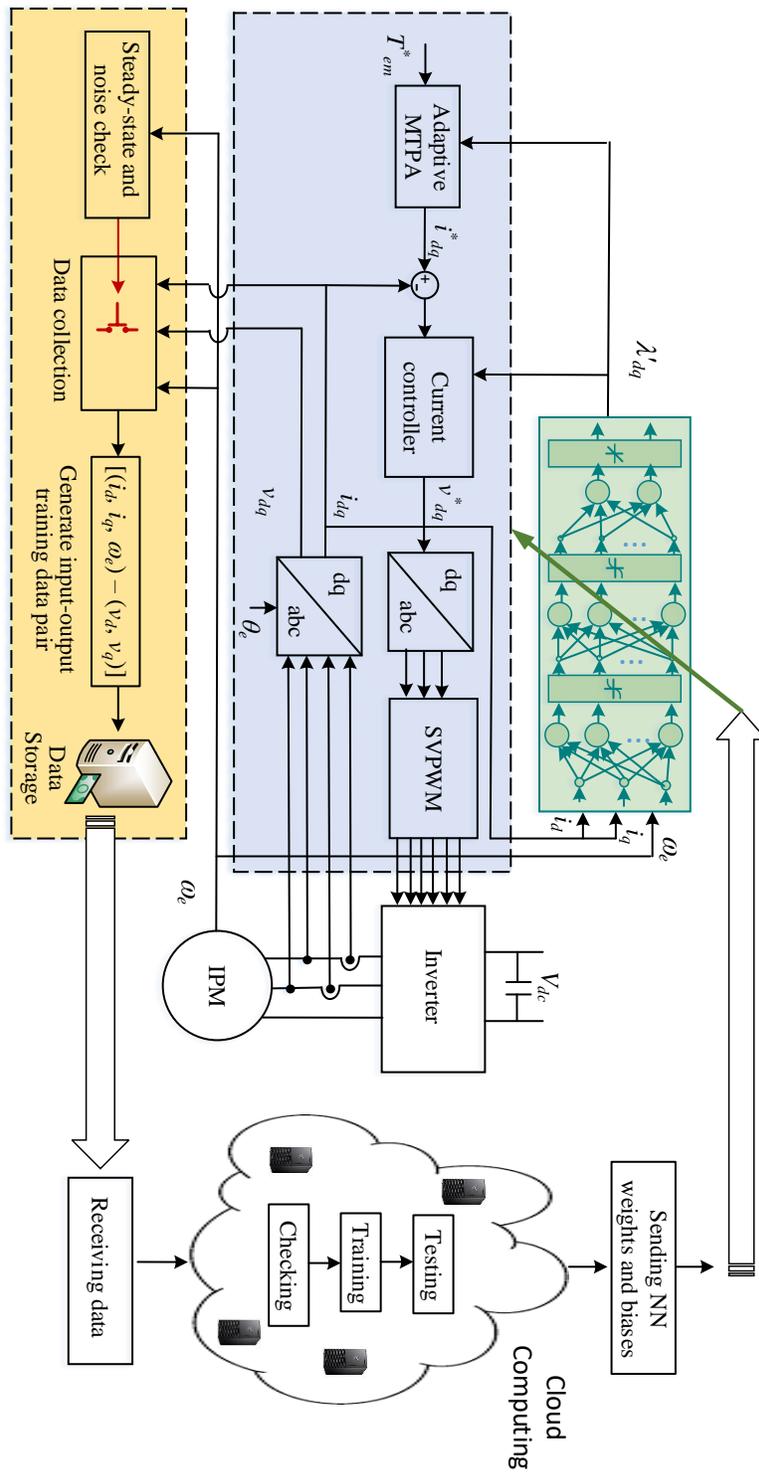


Figure 3.4: NN for parameter identification and the cloud computing framework in an overall IPM motor drive and control system

3.4 Data Collection and NN training

3.4.1 Collection of training data

The collection of the training data is different for the NN development at the production and operation stages.

At the production stage, the training data is collected from a sample test motor of the same motor type base on a set of experiments. During the experiments, the test motor is put on a dyno system, where the motor operating speed and driving torque can be adjusted. For each operating speed and driving torque, the stator dq voltage and current are collected and formulated as one training input-output data pair $[(i_d, i_q, \omega_e) - (v_d, v_q)]$. Then, for the same operating speed, the measurement and data collection process are repeated to cover the full range of the rated current of the motor evenly by adjusting the motor driving torque. After that, the above process and experiment are repeated for the next operating speed until the data collection process covers the full speed range of the test motor. All the data obtained are saved in a storage device that will be used to train the test motor as shown in Section IV-B. Figure 3.4 shows the flowchart of the offline data collection process at the production stage of a sample test IPM motor.

At the operation stage of each individual IPM motor, the training data is collected based on the real-time measurement of the motor during the real-life operation of the EV. The first thing that is needed before the online data collection is to divide the motor operation ranges into multiple lattices in terms of the motor speed from 0 to the maximum speed ω_{e_max} and motor d- and q-axis currents from $-I_{rated}$ to I_{rated} . Each lattice will have a flag indicator to indicate whether there are sufficient data collected corresponding to that lattice, such as 10 data samples for a lattice. This is to assure the appropriate data equity for training the NN. In general, the data

collection process follows the following strategies. (i) Real-time measured data will be collected when the motor is in a steady-state condition and with no or minimum noise impact. (ii) Corresponding to each lattice discussed above, the measured data will be saved in the storage only when there is not enough data collected for that lattice. For example, when there are 10 data samples that have been collected and saved in the storage corresponding to a lattice, any new measured data will not be collected and saved as the training data. (iii) After enough data are collected and saved into the storage corresponding to all the lattices, the online data collection process will be stopped and the data is ready to be transmitted to the cloud for the NN training over the cloud platform. (iv) After the data is sent to the cloud, the storage corresponding to all the lattices will be cleared and the system is ready for the next-round data collection process. Figure 3.4 shows the flowchart of the online data collection process at the operation stage of each individual IPM motor.

3.4.2 NN Training

According to Figure 3.3, for a specified input of (i_d, i_q, ω_e) to the NN, the NN gives an estimate of d- and q-axis flux linkages, (λ'_d, λ'_q) . Then, the output of the NN, together with the NN input data, (i_d, i_q, ω_e) , can be used in (6) to estimate the d- and q-axis voltages, v'_d and v'_q , applied to the IPM motor. This estimated dq voltage is compared with the actual stator dq voltage applied to the motor to examine the NN performance. In other words, the NN can be trained to minimize the mean difference between the estimated and actual stator dq voltages as shown by

$$C = \frac{1}{N} \sum_{k=1}^N \vec{e}_{dq}(k) \quad (3-22)$$

$$\vec{e}_{dq}(k) = \sqrt{\left[v'_d(k) - v_d(k) \right]^2 + \left[v'_q(k) - v_q(k) \right]^2} \quad (3-23)$$

where N is the total number of samples of the training data set, $v_d(k)$ and $v_q(k)$ are the target d- and q-axis voltages of the k th training sample, and $v'_d(k)$ and $v'_q(k)$ are the corresponding estimated d- and q-axis voltages obtained from the NN and (6).

The NN is trained to minimize the performance function (12). We used the LMBP algorithm [25] to update the weights and biases at each training iteration. In order to use the LMBP for training the NN efficiently, the performance function is rewritten as

$$C = \frac{1}{N} \sum_{k=1}^N e_{dq}(k) \xrightarrow{\text{define } V(k) = \sqrt{(e_{dq}(k))}} C = \frac{1}{N} \sum_{k=1}^N (V(k))^2 \quad (3-24)$$

Then, the gradient $\partial C / \partial \vec{w}$ can be written in a matrix as

$$\frac{\partial C}{\partial \vec{w}} = \frac{\partial \sum_{k=1}^N [V(k)]^2}{\partial \vec{w}} = \sum_{k=1}^N 2V(k) \frac{\partial V(k)}{\partial \vec{w}} = 2J(\vec{w})^T \vec{V} \quad (3-25)$$

where the Jacobian matrix $J(\vec{w})$ is

$$J(\vec{w}) = \begin{bmatrix} \frac{\partial V(1)}{\partial w_1} & \dots & \frac{\partial V(1)}{\partial w_M} \\ \vdots & \ddots & \vdots \\ \frac{\partial V(N)}{\partial w_1} & \dots & \frac{\partial V(N)}{\partial w_M} \end{bmatrix}, \quad \vec{V} = \begin{bmatrix} V(1) \\ \vdots \\ V(N) \end{bmatrix} \quad (3-26)$$

Therefore, the weight update [30] is expressed by

$$\Delta \vec{w} = - \left[J(\vec{w})^T J(\vec{w}) + \mu \mathbf{I} \right]^{-1} J(\vec{w})^T \vec{V} \quad (3-27)$$

$$\vec{w}_{\text{update}} = \vec{w} + \Delta \vec{w} \quad (3-28)$$

where μ is the damping parameter.

The training procedure for each epoch is illustrated in Figure 3.5. The batch training method is adopted because of its fast convergence performance with smaller errors than other methods. For each iteration, all the stored data are used to generate the errors and Jacobian matrix. Then, the gradients are computed to update the weights and biases. The training stops when the performance function meets the requirement.

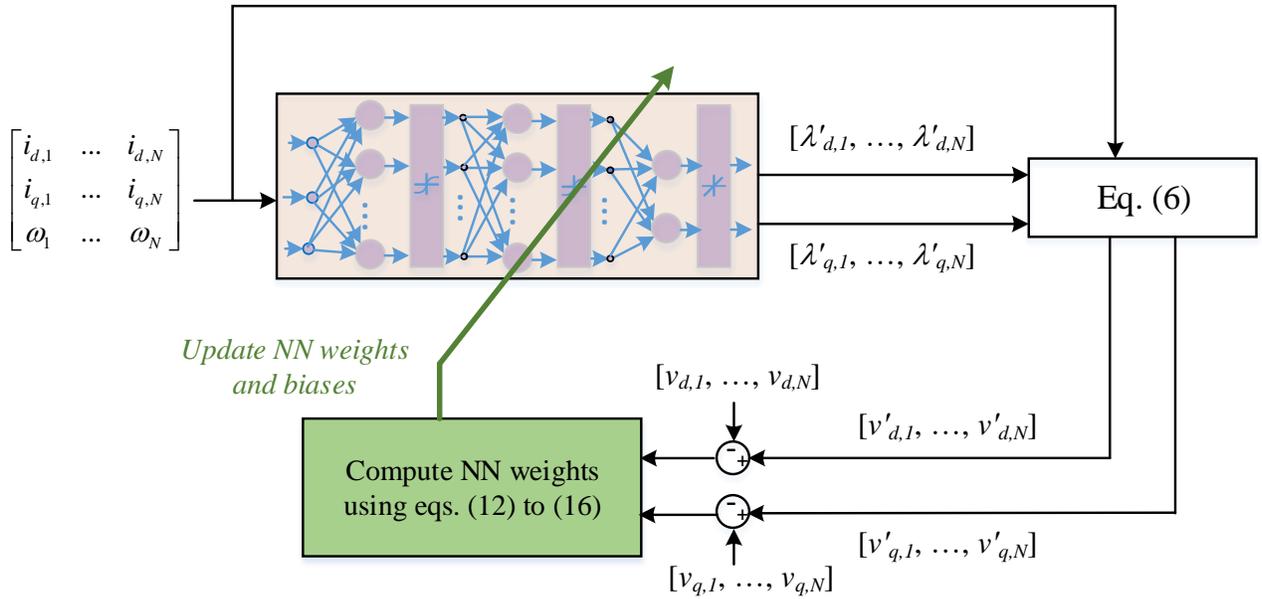


Figure 3.5: Training NN for IPM magnetic modeling

For the routine offline training of the NN over the cloud computing platform, it will follow the same strategy except that data needs to be collected and transmitted to a remote cloud device for the NN training. Firstly, real-time motor operation data is collected smartly as shown

in Section IV-A, meaning that only meaningful and valuable data is collected and stored, and data collected should cover the full operating range of the motor at the normal and critical conditions. When sufficient data is obtained, data is sent to a remote cloud computing device, where data is analyzed and, if needed, a cloud-based training of the NN will be conducted thoroughly. After the training, the new NN weights are transmitted back to the EV computer to replace the previous NN weights. Such an offline learning strategy, based upon a cloud computing platform, will guaranty the safe and reliable NN system development that cannot be achieved for an online NN learning system, which will assure the high-performance operation of individual EV motor over its lifetime.

3.5 Simulation Evaluation

The simulation evaluation is based on a 3-phase 8-pole, 100 kW IPM The simulation evaluation is based on a 3-phase 8-pole, 100 kW IPM motor from [25]. The simulation experiments for the IPM magnetic model identification was conducted based on an EMT simulation model of the IPM motor developed by using MATLAB Simscape Electrical [26]. The maximum current and dc-link voltage of the motor are 450A and 500V, respectively. The motor parameter variations are built into the model [25]. We conducted a series of simulation experiments to obtain or “measure” the motor speed and d- and q-axis currents and voltages based on the flowchart shown in Figure 3.3. This generates training dataset of motor d- and q-axis currents and speed vs. d- and q-axis voltages $[(i_d, i_q) - (v_d, v_q)]$ that are used to obtain the motor magnetic model. Based on the data, the proposed approach is evaluated and compared with measured results and results generated using other approaches, such as the LUT and NN approaches shown in Section II-C. Note: as the LUT and NN approaches shown in Section II-C

are equivalent, only the results of the conventional NN method is presented in the following comparison studies.

Figure 3.6 shows the comparison results with the d-axis current changing from -200A to 0A. In Figs. 9a1 and 9b1, the q-axis current and motor speed are 10A and 500rpm, respectively. In Figs. 9a2 and 9b2, the q-axis current and motor speed are 200A and 9000rpm, respectively. It can be seen the proposed NN method can generate more accurate results for the dq flux linkages of the IPM motor.

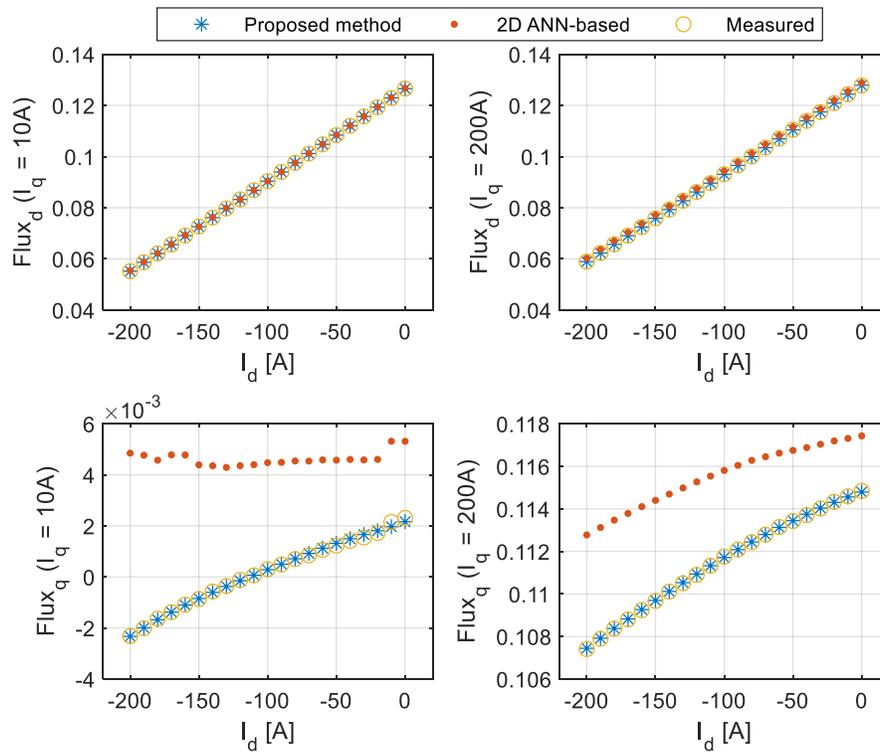


Figure 3.6: Estimated λ_d and λ_q vs. i_d using the proposed and convention NN approaches: (a1, b1) – $i_d=10A$, $\omega_e=500rpm$; (a2, b2) – $i_d=200A$ $i_d=10A$, $\omega_e=500rpm$; (a2, b2) – $i_d=200A$, $\omega_e=9000rpm$

Figure 3.7 shows the comparison results with the q-axis current changing from 0A to 200A. In Figure 3.7a1 and Figure 3.7b1, the d-axis current and motor speed are -10A and 200rpm, respectively. In Figure 3.7a2 and Figure 3.7b2, the d-axis current and motor speed are -

200A and 9000rpm, respectively. The comparison shows again that the proposed NN method can generate more accurate results for the motor dq flux linkage estimations.

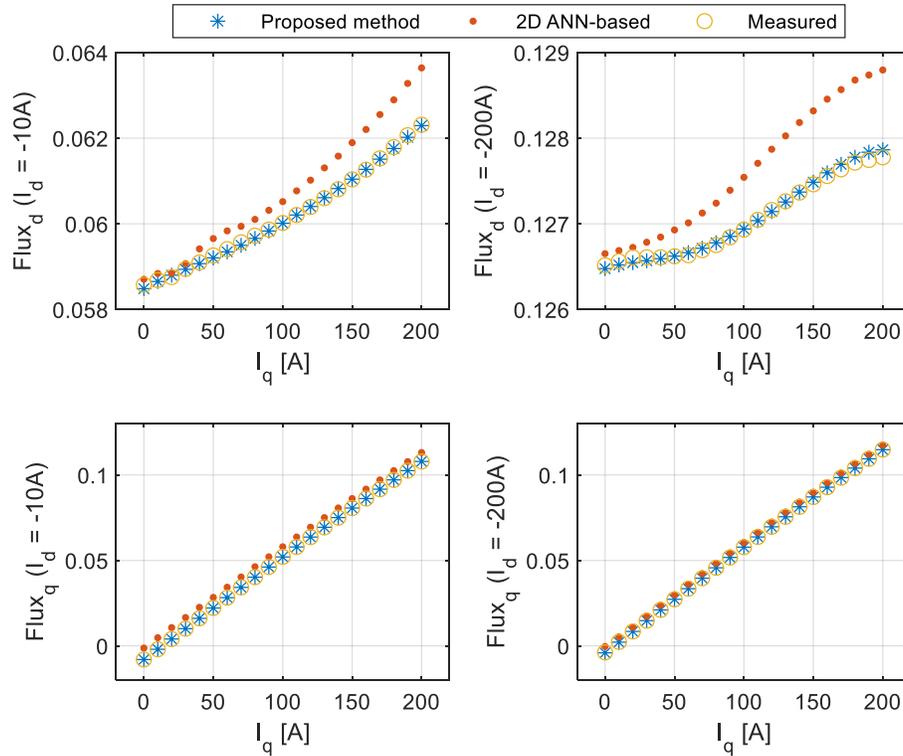


Figure 3.7: Estimated λ_d and λ_q vs. i_q using the proposed and convention NN approaches: (a1, b1) – $i_d=10A$, $\omega_e=500rpm$; (a2, b2) – $i_d=-200A$, $\omega_e=9000rpm$

Figure 3.8a and Figure 3.8b show the estimation error compared with the real measured value when the motor speed is 2000rpm. It can be seen the estimated flux linkages are very accurate with very small errors. *Figure 3.8c* and *d* show the estimation error of the proposed method and the NN method shown in Section II-C at the same motor speed. From the results, it can be seen the proposed method has smaller estimation error than the conventional one.

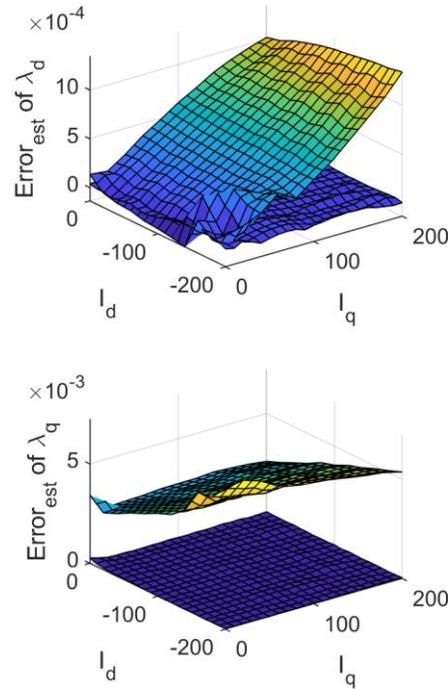


Figure 3.8: Estimation error of λ_d and λ_q vs. i_q using the proposed and convention NN approaches

3.6 Hardware Evaluation

To further validate the proposed method, a DSP-based laboratory system was built for the data collection and NN-based IPM motor magnetic model identification. The experimental setup comprises: (i) a dyno system containing a Kollmorgen IPM motor coupled to a FESTO LabVolt dynamometer; (ii) a 3-phase SEMIKRON IGBT inverter as the motor drive; (iii) a dSPACE MicroLabBox real-time controller [27] used to generate PWM control signals for controlling the IPM motor; (iv) an OP8660 sensor to collect measured voltage and current signals for the dSPACE real-time controller; (v) a MAGNA-POWER DC source as the power supply to the motor. An RC low-pass filter with a cutoff frequency of 1000rad/s was used after the voltage measurement from the OP8660 to remove the switching noise impact. The phase delay and

amplitude attenuation caused by the RC filter was compensated and corrected in the dSPACE experiment setup system.

The testing data was collected based on the procedure shown in Section IV. The measured data include stator d- and q-axis currents and voltages as well as motor operating speed. 1500 data points were collected to formulate the training dataset, which was used to train the NN model for the motor magnetic model identification. Figure 3.9a and Figure 3.9b shows the estimated dq-axis flux linkages vs dq-axis currents under a constant speed (300rpm) using proposed NN method. The errors between the estimated and measured d- and q-axis flux linkages vs. the motor d- and q-axis currents at the two different constant motor operating speeds are shown in Figs. 13a and 13b, respectively. It can be seen from the figure that the estimation errors at both speeds are very small, indicating that the estimated flux linkages using the proposed NN method are very accurate in comparing to the actual motor d- and q-axis flux linkages.

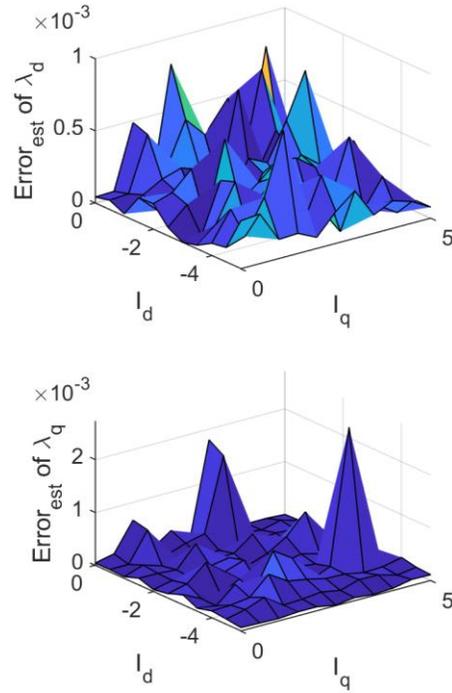


Figure 3.9: Estimation error of λ_d and λ_q vs. i_q using the proposed NN approach

4.7 Conclusion

The efficient operation of an IPM motor is important and also challenging, particularly when considering nonlinear parameter variations of the motor. Accurate identification of the motor parameters or magnetic model is critical for control and management of the IPM motor. This paper presents a novel machine learning strategy based on a neural network (NN) for the motor magnetic model identification over its full operating speed range. Since the proposed NN identifier is trained and generated offline, it has all the advantages that the traditional LUT-based methods have, such as fast current reference generation and good stability, and is able to capture multidimensional impacts into the NN-based magnetic model that the other methods are difficult to achieve.

The proposed method is also very simple to implement and does not have high computational burden compared to the online-based identification methods. At the same time, the proposed method overcomes the disadvantages of LUT-based methods, such as 1) requirement for a large memory size to store the LUTs and 2) limited accuracy due to the LUT size and interpolation. The study shows that compared to the traditional methods, the proposed NN-based method requires a very small memory size and is much more accurate to get accurate motor parameters than the LUT approaches. The hardware experiments further demonstrate the effectiveness of the NN method applicable to practical IPM motors.

To ensure the lifetime adaptivity and learning capabilities of the NN, a cloud-based offline training system is proposed, which will guaranty the safe and reliable NN system development considering motor part-to-part variation and assure high-performance operation of individual IPM motor over its lifetime.

3.7 References

- [1] M. Preindl and S. Bolognani, "Optimal State Reference Computation With Constrained MTPA Criterion for PM Motor Drives," *IEEE Trans. Power Electron.*, vol. 30, no. 8, pp. 4524–4535, Aug. 2015, doi: 10.1109/TPEL.2014.2354299.
- [2] K. D. Hoang and H. K. A. Aorith, "Online Control of IPMSM Drives for Traction Applications Considering Machine Parameter and Inverter Nonlinearities," *IEEE Trans. Transp. Electrific.*, vol. 1, no. 4, pp. 312–325, Dec. 2015, doi: 10.1109/TTE.2015.2477469.
- [3] Z. Han, J. Liu, W. Yang, D. B. Pinhal, N. Reiland, and D. Gerling, "Improved Online Maximum-Torque-Per-Ampere Algorithm for Speed Controlled Interior Permanent Magnet Synchronous Machine," *IEEE Trans. Ind. Electron.*, vol. 67, no. 5, pp. 3398–3408, May 2020, doi: 10.1109/TIE.2019.2918471.
- [4] A. Accetta, M. Cirrincione, M. Pucci, and A. Sferlazza, "State-Space Vector Model of Linear Induction Motors Including End-Effects and Iron Losses—Part II: Model Identification and Results," *IEEE Trans. on Ind. Applicat.*, vol. 56, no. 1, pp. 245–255, Jan. 2020, doi: 10.1109/TIA.2019.2952034.

- [5] Y. Wang, Y. Xu, and J. Zou, "Online Multiparameter Identification Method for Sensorless Control of SPMSM," *IEEE Trans. Power Electron.*, vol. 35, no. 10, pp. 10601–10613, Oct. 2020, doi: 10.1109/TPEL.2020.2974870.
- [6] Y. Inoue, Y. Kawaguchi, S. Morimoto, and M. Sanada, "Performance Improvement of Sensorless IPMSM Drives in a Low-Speed Region Using Online Parameter Identification," *IEEE Trans. on Ind. Applicat.*, vol. 47, no. 2, pp. 798–804, Mar. 2011, doi: 10.1109/TIA.2010.2101994.
- [7] R. Ni, D. Xu, G. Wang, L. Ding, G. Zhang, and L. Qu, "Maximum Efficiency Per Ampere Control of Permanent-Magnet Synchronous Machines," *IEEE Trans. Ind. Electron.*, vol. 62, no. 4, pp. 2135–2143, Apr. 2015, doi: 10.1109/TIE.2014.2354238.
- [8] B. Stumberger, G. Stumberger, D. Dolinar, A. Hamler, and M. Trlep, "Evaluation of saturation and cross-magnetization effects in interior permanent-magnet synchronous motor," *IEEE Trans. on Ind. Applicat.*, vol. 39, no. 5, pp. 1264–1271, Sep. 2003, doi: 10.1109/TIA.2003.816538.
- [9] E. Armando, R. I. Bojoi, P. Guglielmi, G. Pellegrino, and M. Pastorelli, "Experimental Identification of the Magnetic Model of Synchronous Machines," *IEEE Trans. on Ind. Applicat.*, vol. 49, no. 5, pp. 2116–2125, Sep. 2013, doi: 10.1109/TIA.2013.2258876.
- [10] K. M. Rahman and S. Hiti, "Identification of machine parameters of a synchronous motor," *IEEE Transactions on Industry Applications*, vol. 41, no. 2, pp. 557–565, Mar. 2005, doi: 10.1109/TIA.2005.844379.
- [11] H.-S. Kim, Y. Lee, S.-K. Sul, J. Yu, and J. Oh, "Online MTPA Control of IPMSM Based on Robust Numerical Optimization Technique," *IEEE Trans. on Ind. Applicat.*, vol. 55, no. 4, pp. 3736–3746, Jul. 2019, doi: 10.1109/TIA.2019.2904567.
- [12] M. Basic, D. Vukadinovic, I. Grgic, and M. Bubalo, "Speed-Sensorless Vector Control of an Induction Generator Including Stray Load and Iron Losses and Online Parameter Tuning," *IEEE Trans. Energy Convers.*, vol. 35, no. 2, pp. 724–732, Jun. 2020, doi: 10.1109/TEC.2019.2952666.
- [13] V. Ruuskanen, J. Nerg, M. Rilla, and J. Pyrhonen, "Iron Loss Analysis of the Permanent-Magnet Synchronous Machine Based on Finite-Element Analysis Over the Electrical Vehicle Drive Cycle," *IEEE Trans. Ind. Electron.*, vol. 63, no. 7, pp. 4129–4136, Jul. 2016, doi: 10.1109/TIE.2016.2549005.
- [14] B.-S. Jun, J. Park, J.-H. Choi, K.-D. Lee, and C.-Y. Won, "Temperature Estimation of Stator Winding in Permanent Magnet Synchronous Motors Using d-Axis Current Injection," *Energies*, vol. 11, no. 8, p. 2033, Aug. 2018, doi: 10.3390/en11082033.
- [15] Y. Wang et al., "Initial Rotor Position and Magnetic Polarity Identification of PM Synchronous Machine Based on Nonlinear Machine Model and Finite Element Analysis,"

IEEE Transactions on Magnetics, vol. 46, no. 6, pp. 2016–2019, Jun. 2010, doi: 10.1109/TMAG.2010.2042690.

- [16] S. Kallio, J. Karttunen, P. Peltoniemi, P. Silventoinen, and O. Pyrhönen, “Online Estimation of Double-Star IPM Machine Parameters Using RLS Algorithm,” IEEE Transactions on Industrial Electronics, vol. 61, no. 9, pp. 4519–4530, Sep. 2014, doi: 10.1109/TIE.2013.2290761.
- [17] D. Q. Dang, M. S. Rifaq, H. H. Choi, and J.-W. Jung, “Online Parameter Estimation Technique for Adaptive Control Applications of Interior PM Synchronous Motor Drives,” IEEE Trans. Ind. Electron., vol. 63, no. 3, pp. 1438–1449, Mar. 2016, doi: 10.1109/TIE.2015.2494534.
- [18] S. Ozden, G. Manav, and M. Dursun, “ANN based magnetic field and inductance modeling of double sided linear switched reluctance motor,” in 2018 5th International Conference on Electrical and Electronic Engineering (ICEEE), Istanbul, May 2018, pp. 133–137. doi: 10.1109/ICEEE2.2018.8391316.
- [19] L. Ortombina, F. Tinazzi, and M. Zigliotto, “Magnetic Modeling of Synchronous Reluctance and Internal Permanent Magnet Motors Using Radial Basis Function Networks,” IEEE Trans. Ind. Electron., vol. 65, no. 2, pp. 1140–1148, Feb. 2018, doi: 10.1109/TIE.2017.2733502.
- [20] L. Ortombina, F. Tinazzi, and M. Zigliotto, “Adaptive Maximum Torque per Ampere Control of Synchronous Reluctance Motors by Radial Basis Function Networks,” IEEE J. Emerg. Sel. Topics Power Electron., vol. 7, no. 4, pp. 2531–2539, Dec. 2019, doi: 10.1109/JESTPE.2018.2858842.

4. CONTROL OF A BUCK DC/DC CONVERTER USING APPROXIMATE DYNAMIC PROGRAMMING AND ARTIFICIAL NEURAL NETWORKS

4.1 Introduction

With the fast developments of microgrids, electric vehicles and renewable generations, dc/dc converters have been widely used to regulate output dc voltage and power from the distributed energy sources [1-3]. In these applications, the controller design of dc/dc converters is still facing the challenge to accurately and rapidly maintain desired output voltages due to the low switching frequency normally required in high-power converters, load variations, dc input voltage disturbances, parameter deviation, and current and PWM saturation constraints of the converters [2-4].

Two types of conventional control methods, voltage mode control (VMC) and current mode control (CMC), are typically used for the control of a dc/dc converter. The traditional VMC uses PI, Type II, or Type III compensators, and has a single control loop with voltage feedback [5, 6]. The implementation is simple, but the load-disturbance rejection ability is poor. The CMC improves the performance through a cascade structure, by introducing an inner inductor current-control loop. This structure has the ability to limit the inductor current due to the introduction of the inner current-loop controller. However, the response speed of the output voltage control could be affected due to the two-nested-loop configuration.

In recent years, various advanced control techniques for dc/dc converters have been developed [7-13]. Sliding-mode control (SMC) is a popular method developed in recent years for

dc/dc converter control. The technology has been shifted from early first-order SMC [7] to recent second-order SMC [8-10]. Second-order SMC improves performance measures such as transient response, in comparison to first-order SMC, but an extra capacitor current sensor is usually needed to achieve this. Conventionally, hysteresis-modulation (HM) based SMC is used for control of a dc/dc converter, but one of the major problems is that the switching frequency is not constant [7-9]. Recently, PWM-based SMC was developed to overcome the variable switching frequency issue [10, 11]. In [10], a nested SMC strategy is adopted in both voltage and current control loops for DC/DC converters. With this design, the robustness of the paralleled converter system is improved. In [11], a disturbance observer is integrated with a PWM-based sliding mode approach to improve the voltage tracking performance. But the PWM-based SMC typically requires high switching frequency and high sampling rate in order to assure a good dynamic response, which can cause excessive losses and complicated filter designs, and is not suitable for high-power converters. A few research articles show the use of model predictive control (MPC) for control of dc/dc converters, because of its fast dynamic response [12, 13]. However, a weakness of the MPC is it would become unstable when the model parameters differ from the actual values.

Artificial neural networks (ANNs) have been applied to dc/dc converter control in recent years. Nevertheless, ANNs have not been developed to implement predictive and optimal control of the dc/dc converter based on approximate dynamic programming (ADP). In [14], a feedforward ANN is proposed to assist the sliding-mode based control of a dc/dc Cuk converter, which is fundamentally still a sliding-mode based controller. In [15], the authors introduced a neural network to improve the performance of a fuzzy-controlled dc/dc converter. In [16], an adaptive fuzzy-neural-network control scheme is designed based upon the SMC for the voltage

tracking of a dc/dc boost converter. Similar to [14] though, the overall control structure is a sliding-mode based controller, while the purpose of the fuzzy-neural-network scheme is to help improve the SMC performance.

Although significant research has been conducted in optimal control of nonlinear systems based on ADP [17-21], none focuses on dc/dc converter control. In [21], an ANN control strategy was developed for control of dc/ac inverters based on ADP [17] while how to implement ADP-based ANN control for dc/dc converters remains unknown. The authors in [20] proposed an ADP-based optimal switching strategy for dc/dc converter control without using ANNs. However, the ADP-based power-converter switching mechanism is similar to a hysteresis switching strategy used in SMC [7, 8]. As a result, the switching frequency varies depending on the optimal action generated by the ADP strategy proposed in [19], which is difficult to implement in practical applications.

This paper develops ADP-based ANN control in the PWM switching framework for dc/dc buck converters. Some special features of the proposed method include: 1) The control objective of a dc/dc buck converter is defined based on ADP and implemented via an ANN; 2) The complete system dynamic equation for the dc/dc converter is integrated into the ANN development to achieve the ADP-based optimal control; 3) A recurrent network structure is formulated by integrating the dc/dc converter feedback and the ANN as an integrated system; 4) Error signals and integrals of error signals are used as network inputs to let the ANN gain PI control ability; 5) The ANN is trained offline to avoid the instability of the ANN at runtime that could be caused by network weight adjustments of a real-time training algorithm. On the other hand, compared to the conventional control methods, there are two main limitations associated with the proposed control method. One is that training of the ANN controller is needed in the

design stage of the controller. The other one is that more computing time is needed in the implementation stage of the controller. However, it is appropriate to point out that since the proposed ANN controller is trained offline, the proposed ANN controller can be easily implemented using a low-cost DSP as demonstrated by the hardware experiment shown in Section 4.4.

The rest of the chapter is structured as follows: Section 4.2 reviews conventional control methods of a buck converter, Section 4.3 presents the proposed ANN-based control of the buck converter. Section 4.4 shows how to train the ANN to implement the ADP-based optimal control for the dc/dc buck converter. Section 4.5 presents simulation evaluation, and the hardware experiment evaluation is presented in Section 4.6. Finally, Section 4.8 summarizes with conclusions.

4.2 Conventional Control of Buck Converters

4.2.1 Buck converter model

A basic buck converter is shown in Figure 4.1, where V_{dc} represents the input *dc* voltage. Using the converter average model and the generator sign convention, the voltage and current-balance equations across the smoothing inductor and capacitor of the *dc/dc* converter are

$$v_A = R_L i_L + L \cdot di_L/dt + v_o \quad (4-1)$$

$$C \cdot dv_c/dt = i_L - v_o/R \quad (4-2)$$

$$v_o = R_C (i_L - v_o/R) + v_c \quad (4-3)$$

where R_L and L are the resistance and inductance of the inductor, R_C and C are the resistance and capacitance of the capacitor, v_A represents the average voltage at the diode, v_C is the capacitor voltage, and v_o is the output voltage to the load R , and i_L is the current flowing through the inductor.

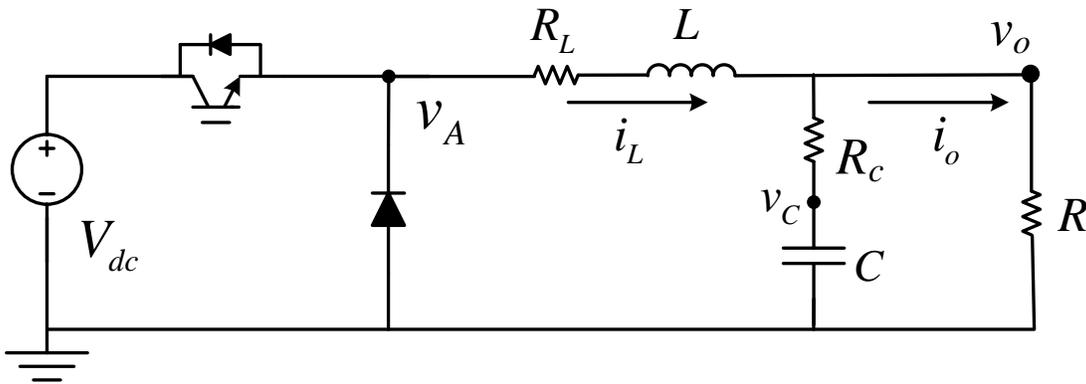


Figure 4.1: A dc/dc buck converter with loads

In typical controller design of a buck converter, the impact of the capacitor resistance is generally neglected, making the model of the buck converter as follows:

$$v_A = R_L i_L + L \cdot di_L/dt + v_o \quad (4-4)$$

$$C \cdot dv_o/dt = i_L - v_o/R \quad (4-5)$$

Also, a graphic representation of (4-4) is usually used for the design of a conventional controller [8], as shown in Figure 4.2.

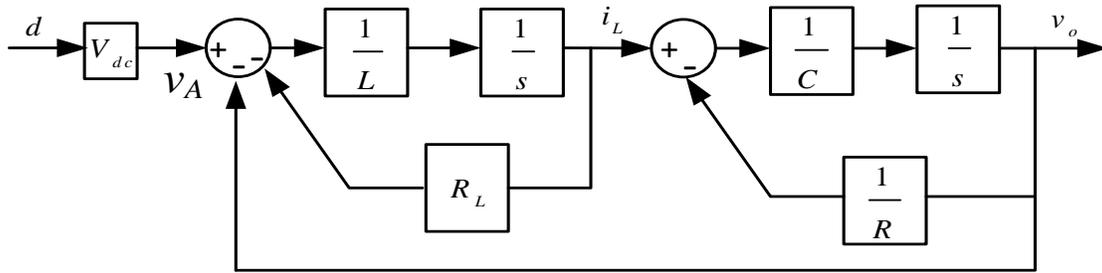


Figure 4.2: Buck converter graphic model

4.2.2 VMC based control

VMC-based control typically has one voltage control loop. To design a VMC controller, a transfer function is needed between the buck-converter output voltage v_o and the control voltage v_A generated by the VMC. This is obtained from (4-4) (4-5) or Figure 4.2 as follows:

$$\frac{V_o(s)}{V_A(s)} = \frac{V_o(s)}{d \cdot V_{dc}} = \frac{1}{s^2 LC + s(R_L C + L/R) + R_L/R + 1} \quad (4-6)$$

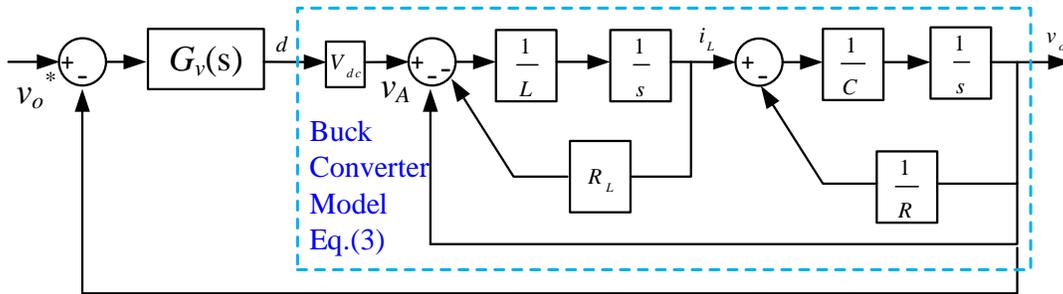


Figure 4.3: Buck converter VMC model

In terms of the buck-converter graphic model shown in Figure 4.3, the block diagram of the closed-loop control system can be obtained as shown in Figure 4.3, in which $G_v(s)$ represents the transfer function of the VMC controller, and v_o^* is the reference output voltage of the dc/dc

converter. A Type-III compensator is usually employed [5, 6]. To design the VMC controller $G_v(s)$ using the Bode plot design approach, the cutoff frequency of the controller is generally selected as one to two orders smaller than the converter switching frequency.

4.2.3 CMC based control

CMC typically has a cascade control structure [5]. The overall block diagram of the cascade control is shown in Figure 4.4, which consists of an inner current-loop controller, plus an outer voltage-loop controller. Typically, decoupling between the voltage v_o and the current i_L is needed [5, 22, 23]. Thus, the transfer function between the buck-converter output current i_L and the control voltage v_A generated by the current-loop controller is:

$$\frac{I_L(s)}{V_A(s)} = \frac{I_L(s)}{d \cdot V_{dc}} = \frac{1}{s \cdot L + R_L} \quad (4-7)$$

In terms of the buck-converter graphic model shown in Figure 4.2, the block diagram of the current-loop control system is represented by the inner-block enclosed by the dashed green line shown in Figure 4.4, in which $G_i(s)$ represents the transfer function of the inner current-loop controller, and i_L^* is the reference current generated by the voltage-loop controller.

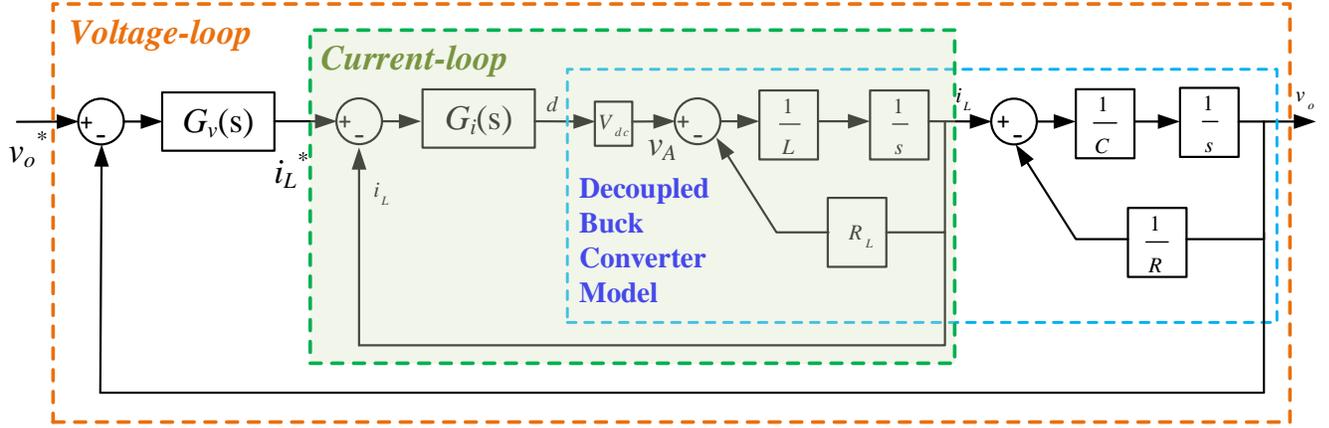


Figure 4.4: Buck converter CMC cascade voltage-current control model

Typically, the current-loop controller is much faster than the voltage-loop controller, and is generally assumed to be ideal. Hence, the transfer function of the current-loop is assumed to be 1 during the design of the voltage-loop controller [5, 22, 23]. To design the voltage-loop controller, the transfer function between the buck-converter output voltage v_o , and the control action i_L generated by the voltage-loop controller, is obtained as follows:

$$\frac{V_o(s)}{I_L(s)} = \frac{1}{s \cdot C + 1/R} \quad (4-8)$$

In terms of the graphic illustration shown in Figure 4.4, $G_v(s)$ represents the transfer function of the outer voltage-loop controller, which is designed according to (4-8), and i_L^* is the reference current generated by the voltage-loop controller.

Using the Bode-plot approach, the cutoff frequency of the voltage-loop controller is normally one order smaller than that of the current-loop controller. As a result, under the same switching frequency, the response speed of the output voltage for the cascade control strategy is generally slower than that of the VMC approach.

4.2.4 Sliding-mode based control

Figure 4.5 shows a PWM-based second-order SMC approach [24], which can overcome the variable switching frequency issue associated with traditional SMC, and also can use low switching frequency and sampling rate for control of the *dc/dc* buck converter. Similar to the CMC, a cascade SMC control structure is employed, which includes a second-order current-loop SMC and a second-order voltage-loop SMC. The sliding surfaces for the current- and voltage-loop SMCs are defined as (4-9) and (4-10), respectively, as shown by

$$S_i = e_i + \lambda_i \cdot \int e_i dt \quad (4-9)$$

$$S_v = e_v + \lambda_v \cdot \int e_v dt \quad (4-10)$$

where $e_i = i_L^* - i_L$ and $e_v = v_o^* - v_o$. Here v_o^* and i_L^* represent output voltage and inductor current references, respectively. According to (4-4) and (4-5), the control actions generated by the current- and voltage-loop SMCs are designed as (4-11) and (4-12), respectively, as shown by

$$\dot{i}_L^* = C \cdot \frac{dv_o^*}{dt} + \frac{v_o}{R} + C \cdot (\lambda_v e_v + a_v \cdot \text{sgn}(S_v) + b_v \cdot S_v) \quad (4-11)$$

$$u = \frac{L}{V_{dc}} \frac{di_L^*}{dt} + \frac{i_L R_L}{V_{dc}} + \frac{v_o}{V_{dc}} + \frac{L}{V_{dc}} (\lambda_i e_i + a_i \cdot \text{sgn}(S_i) + b_i \cdot S_i) \quad (4-12)$$

where a_v , b_v and λ_v are the parameters of the voltage-loop SMC, a_i , b_i and λ_i are the parameters of the current-loop SMC, and u is the duty ratio. Details about the PWM-based cascade SMC control design and analysis are provided in [24].

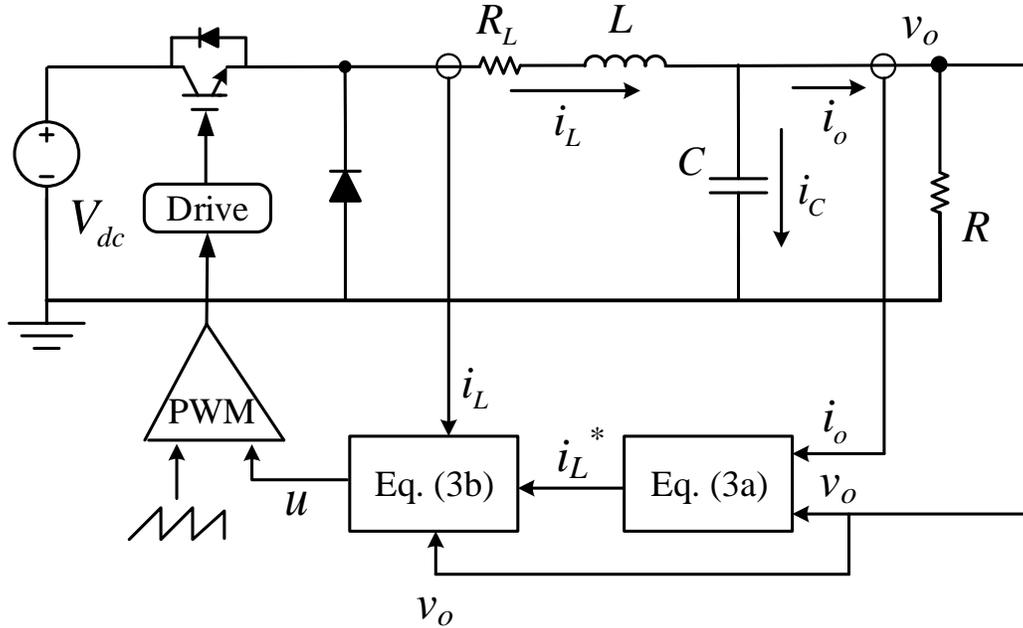


Figure 4.5: A PWM-based cascade SMC for Buck converter

4.3 ANN Control of Buck converter

4.3.1 Buck converter state-space model

The ANN controller is developed based on the complete state-space model of the buck converter, which is obtained from (4-1) by first rearranging (4-3) to get $v_c = v_o - R_c(i_L - v_o/R)$.

Substituting this into (4-2) and combining with (4-1) gives the state-space model in terms of i_L and v_o , as follows:

$$\frac{d}{dt} \begin{bmatrix} i_L \\ v_o \end{bmatrix} = \begin{bmatrix} -R_L/L & -1/L \\ \frac{(L - CR_L R_C)R}{(R + R_C)CL} & -\frac{L + CRR_C}{(R + R_C)CL} \end{bmatrix} \begin{bmatrix} i_L \\ v_o \end{bmatrix} + \begin{bmatrix} 1/L \\ \frac{RR_C}{(R + R_C)L} \end{bmatrix} v_A \quad (4-13)$$

where the system states are i_L and v_o , and v_A is proportional to the output of the ANN controller [25].

As the ANN controller is a digital controller, a discrete model of (4-13) is needed. This is obtained via a zero- or first-order hold discrete equivalent mechanism as [26]:

$$\begin{bmatrix} i_L(kT_s + T_s) \\ v_o(kT_s + T_s) \end{bmatrix} = \mathbf{A} \begin{bmatrix} i_L(kT_s) \\ v_o(kT_s) \end{bmatrix} + \mathbf{B} \begin{bmatrix} v_A(kT_s) \\ 0 \end{bmatrix} \quad (4-14)$$

in which T_s represents the sampling time, \mathbf{A} is the system matrix, and \mathbf{B} is the input matrix. Note: matrices \mathbf{A} and \mathbf{B} are obtained from (4-13) based on a chosen discrete equivalent mechanism [26]. Since T_s is present on both sides, (4-14) can be simplified as (4-15) where k is an integer time step.

$$\begin{bmatrix} i_L(k+1) \\ v_o(k+1) \end{bmatrix} = \mathbf{A} \begin{bmatrix} i_L(k) \\ v_o(k) \end{bmatrix} + \mathbf{B} \begin{bmatrix} v_A(k) \\ 0 \end{bmatrix}, \quad (4-15)$$

4.3.2 ANN Control Structure

The overall ANN control structure is shown in the lower part of Figure 4.6, in which the ANN is a feedforward network. The ANN consists of four different layers: an input layer, two hidden layers, and an output layer. The input layer contains two inputs: the error term and the integral of the error term as defined by:

$$e_{v_o}(k) = v_o(k) - v_o^*(k), \quad s_{v_o}(k) = \int_0^{kT_s} e_{v_o}(t) dt, \quad (4-16)$$

where $v_o^*(k)$ is the reference output voltage of the dc/dc converter. The two inputs are divided by their appropriate gains, and then processed through a hyperbolic tangent activation function. Each gain is selected as 4 for the simulation and experimental Buck converter setup shown in Sections 4.5 and 4.6. The input layer then feeds into the hidden layers. Each of the two hidden layers contains six nodes. Each node at the hidden layers uses a hyperbolic-tangent activation

function. Finally, the output layer gives $v_A^*(k)$, the output of the ANN. This output is multiplied by a gain, k_{PWM} , which represents the PWM gain, to obtain the final control action, v_A , of the *dc/dc* converter given by:

$$v_A(k) = k_{PWM} \cdot A(e_{v_o}(k), s_{v_o}(k), \vec{w}), \quad (4-17)$$

where \vec{w} represents the network's overall weight vector, and $A(\bullet)$ denotes the whole ANN. The error signal and integral of the error signal as the network inputs would enable the ANN to gain important PI control characteristics. Besides, there are hundreds of 'PI' gains for the ANN controller instead of two gains for a conventional PI controller, and the training of the network should enable its performance to match, and potentially exceed, that of an optimal "PI" controller.

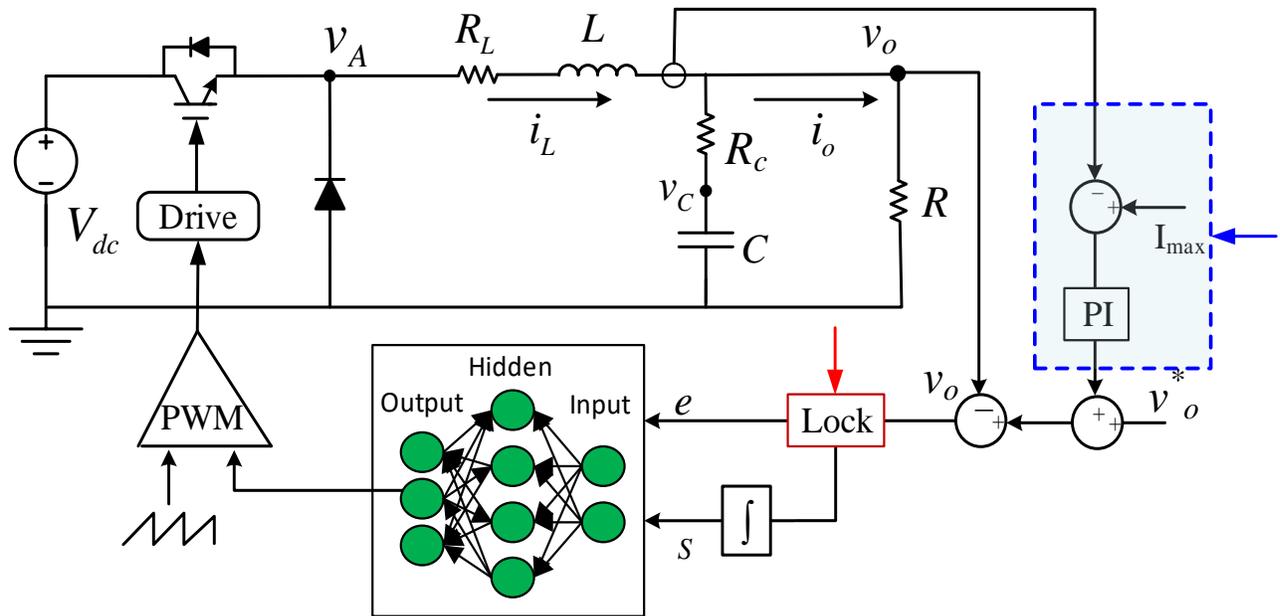


Figure 4.6: ANN control for buck *dc/dc* converter

It is also important to point out that although the ANN is a feedforward network, the feedback signal of the dc/dc converter applied as the input to the ANN makes the combined ANN and dc/dc converter equivalent to a recurrent neural network. This property is considered properly and accurately in training the ANN as shown in Section 4.4, which would allow the ANN to gain important predictive control ability.

4.3.3 Maximum Duty-ratio and Current Limitations

During the real-time control stage, it is possible that the controller output voltage may be beyond the maximum duty cycle constraint, or the inductor current may be beyond the maximum inductor current limitation.

To handle the maximum duty cycle constraint, a locking mechanism (Figure 4.6) is developed with the ANN controller. The mechanism first detects whether the controller output voltage is beyond the PWM saturation limit. If so, the error signal passed to the ANN controller will be blocked and the controller just maintains the output voltage at the maximum duty cycle until there is a potential to draw the ANN controller out of the PWM saturation limit.

To handle the maximum inductor current constraint, a PI regulation block (Figure 4.6) is added to adjust the reference output voltage of the dc/dc converter. However, this PI regulation block is only initiated when the actual inductor current is over the maximum current constraint and stops when the actual inductor current is about 2% below the maximum current constraint. Here, 2% is a dead-band margin to assure that the PI controller for the maximum current limitation does not turn on and off constantly at the maximum current constraint. Later, it will be demonstrated by the simulation and hardware experiments shown in Sections 4.5 and 4.6 that the proposed ANN controller correctly handles the maximum current constraint, even using such a small dead-band margin.

4.4 Training ANN to Implement ADP-Based Control

4.4.1 ADP-based Control

ADP employs the principle of Bellman's optimality [17] and is a very useful tool for solving optimization and optimal control problems. The typical structure of discrete-time ADP includes a discrete-time state-space system model and a performance index or cost associated with the system [17, 18]. For the ADP-based control of the *dc/dc* converter, the discrete-time state-space model is (10) and the performance index or cost is

$$C(v_o) = \sum_{k=1}^N \alpha \sqrt{[v_o(k) - v_o^*(k)]^2} \quad (4-18)$$

where N is the trajectory length and α is a fractional number. The objective of the ADP-based control for the *dc/dc* converter is to determine a sequence of control actions $v_A(k)$, $k=1, 2, \dots, N$, such that the ADP cost (4-18) is minimized. Compared to the cost function normally used for the conventional MPC methods, the ADP cost function emphasizes minimizing the difference between the actual and reference voltages over a much longer time horizon, instead of the one-step ahead that is normally used in the conventional MPC [13]. Thus, ADP-based control would provide a much stronger predictive control ability than conventional MPC.

4.4.2 Training ANN to Implement ADP

The ADP-based control is achieved through the ANN that is trained to minimize the ADP cost function (4-18). We used the Levenberg-Marquardt (LM) algorithm [27] to train the ANN, and the Jacobian matrix needed by the LM algorithm is calculated via a Forward Accumulation Through Time (FATT) algorithm [28]. Similar to many other neural network training algorithms, the most important part of the training algorithm is the calculation of the gradient of (4-18)

regarding the weight vector. Define $U(k) = \sqrt{\alpha [v_o(k) - v_o^*(k)]^2}$ and $V(k) = \sqrt{U(k)}$, then, the gradient $\partial C / \partial \vec{w}$ can be written in matrix form as

$$\frac{\partial C}{\partial \vec{w}} = \frac{\partial \sum_{k=1}^N [V(k)]^2}{\partial \vec{w}} = \sum_{k=1}^N 2V(k) \frac{\partial V(k)}{\partial \vec{w}} = 2J(\vec{w})^T \vec{V} \quad (4-19)$$

where the Jacobian matrix $J(\vec{w})$ is

$$J(\vec{w}) = \begin{bmatrix} \frac{\partial V(1)}{\partial w_1} & \dots & \frac{\partial V(1)}{\partial w_M} \\ \vdots & \ddots & \vdots \\ \frac{\partial V(N)}{\partial w_1} & \dots & \frac{\partial V(N)}{\partial w_M} \end{bmatrix}, \quad \vec{V} = \begin{bmatrix} V(1) \\ \vdots \\ V(N) \end{bmatrix} \quad (4-20)$$

Therefore, the weight update can be expressed by

$$\Delta \vec{w} = - \left[J(\vec{w})^T J(\vec{w}) + \mu \mathbf{I} \right]^{-1} J(\vec{w})^T \vec{V} \quad (4-21)$$

Here $\mu > 0$ is an adaptable parameter set by the LM algorithm [28]. The network weights are updated by

$$\vec{w}_{\text{update}} = \vec{w} + \Delta \vec{w} \quad (4-22)$$

Note: the combination of the ANN and the dc/dc converter is equivalent to a recurrent network as explained in Section 4.3. Also, the ANN is trained offline, meaning that there is no further training involved at the real-time control stage. A more detailed description about training a recurrent network using LM and FATT algorithms is provided in [28]. In general, in each experiment, training continued until one of the following three stopping criteria were met [28]:

1) when the training epoch reaches the maximum number of training epochs, 2) when μ is larger than μ_{\max} , a predefined maximum μ value, or 3) when the smallest gradient of (4-19) is less than a predefined minimum gradient.

4.5 Simulation Evaluation

The parameters of the *dc/dc* buck converter used in both the simulation and experiment evaluation are as follows: $R_L=0.3\Omega$, $L=5.63\text{mH}$, $R_C=0.02\Omega$, and $C=5\mu\text{F}$. The nominal input voltage is 42V.

4.5.1 Tuning of Conventional Controller

The conventional VMC and CMC controllers were tuned based on the description shown in Sections 4.2.2 and 4.2.3 using the phase-margin of 60° . In the simulation, the converter switching frequency was 20kHz. To reject the switching noises and disturbances, the crossover frequency of the VMC Type-III compensator is selected lower than the switching frequency, usually from $0.1f_{sw}$ to $0.05f_{sw}$. Therefore, the bandwidth of the VMC compensator was selected as 1000 Hz. For the CMC, a cascade PI configuration was used. To limit the switching noise in the current loop, the bandwidth of the current PI controller was selected as 1000 Hz. Then the bandwidth of the outer voltage loop was selected as $1000/10 = 100$ Hz. The SMC controller was tuned according to Section 4.2.4 and [24].

4.5.2 Training of ANN Controller

The ANN was trained to implement ADP-based control through multiple training experiments. In each experiment, the ANN was trained repeatedly to track a variety of randomly generated reference voltage trajectories. The procedure of each training experiment is as follows:

1) randomly generate a sample reference output voltage trajectory; 2) randomly generate a

sample initial state $v_o(1)$ where the value 1 indicates a start time; 3) unroll the converter output voltage trajectory from the initial state; 4) train the ANN as detailed in Section 4.4; and 5) repeat the process for other randomly generated reference voltage trajectories and sample initial states. In each training experiment, a dozen randomized reference voltage trajectories were created to train the network. Each reference trajectory duration was 1 second, with a sampling time of $T_s=0.1\text{ms}$, and was changed randomly every 0.1 seconds. The training of each experiment for all randomly generated reference output voltage trajectories continued until reaching a stop criterion (Section 4.4.2). Each training experiment started with randomly generated network weights, which were initially randomized using a uniform distribution within ± 0.1 . The impact of load and input voltage variations are considered as noises in each training experiment. Each training experiment took about 10 to 30 minutes to complete on a PC with a 2.3GHz CPU and 16GB RAM. Since each experiment starts with randomly generated weights, each may converge to a different ADP cost. The final network weights were selected from the training experiment having the lowest ADP cost. Compared to the conventional control methods, the trained ANN controller has a very strong adaptive ability to withstand circuit parameter changes that may appear in real-life conditions, such as the increase or decrease of L and C values beyond the nominal values, as shown in the subsection below.

4.5.3 Control Evaluation within System Constraints

The tuned conventional controllers shown in Section 4.5.1 and the trained ANN controller shown in Section 4.5.2 are first evaluated and compared via simulation. The simulation models of traditional and ANN-controlled dc/dc buck converter were built by using SimPowerSystems. Again, the switching frequency was 20kHz. The evaluation focuses on the

output voltage and inductor current under different conditions using conventional and ANN control techniques as presented in Sections 4.2 to 4.4.

Figure 4.7 a) compares the control of the buck converter under a load change from 7.33Ω to 11Ω using VMC-Type III, CMC-Cascade PI, Cascade SMC, and ANN. The comparison shows both the conventional VMC and the proposed ANN controller responding faster to maintain the output voltage at a constant level than the cascade PI and SMC control. This is due to the fact that there are two control loops for the CMC and SMC approaches. Thus, the response speed of the external voltage loop would be slower than that of a single voltage control loop according to the design rules presented in Section II. Overall, the ANN controller shows the best performance and fastest response speed, demonstrating the higher bandwidth of the ANN controller than that of others.

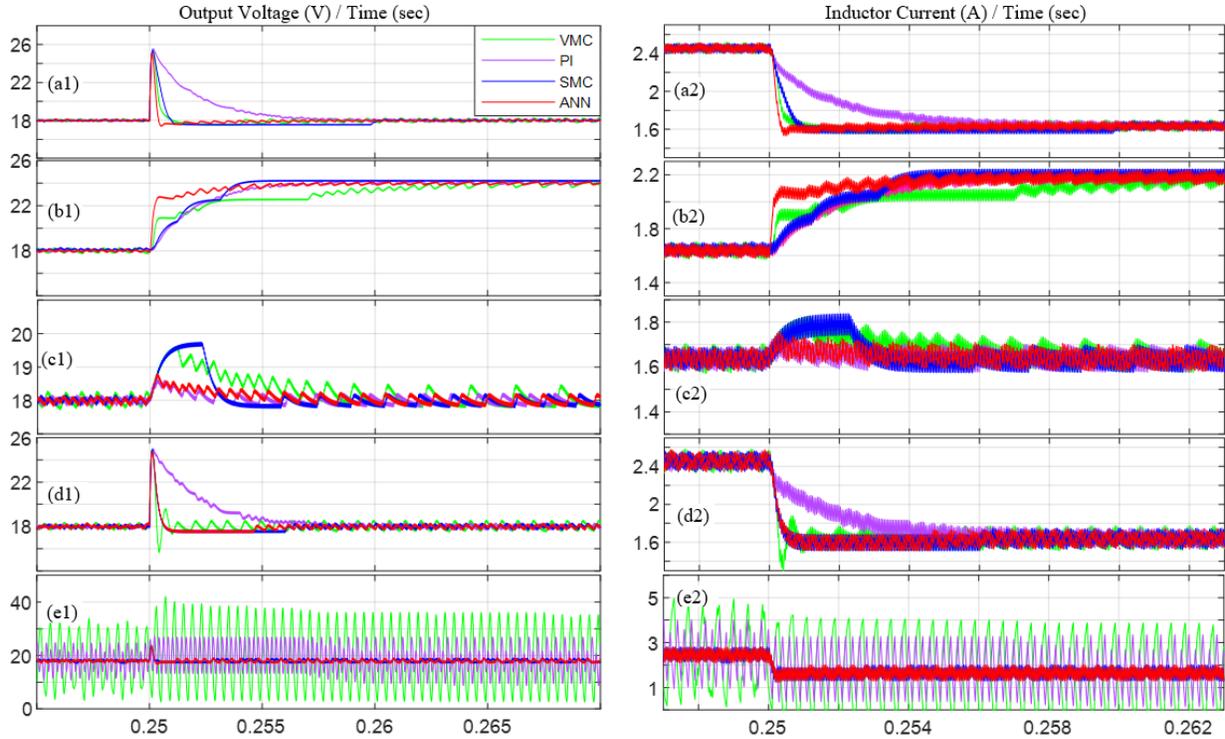


Figure 4.7: Simulated results for ANN vs. VMC Type III, CMC cascade PI, and cascade SMC: a) a load change from 7.33Ω to 11Ω , b) a reference voltage change from 18V to 24V, c) an input voltage change from 42V to 47V, d) the same load change as (a) when L decreases by 50%, e) the same load change as (a) when L decreases by 85%

Figure 4.7 b) compares the control of the buck converter under a reference-voltage change from 18V to 24V using VMC, CMC, SMC, and ANN. The comparison shows that the ANN controller has the fastest response speed to track a reference voltage change than other control methods.

Figure 4.7 c) compares the control of the buck converter as the input voltage changes from 42V to 47V, to examine how the four different control methods behave in maintaining output-voltage stability when there is a disturbance in the supply voltage. Since there is no load or reference output voltage change, the CMC-Cascade PI has better performance than the VMC-Type III. However, the ANN still has the best performance, demonstrating its strong adaptive ability to manage a system condition variation.

Figure 4.7d) and Figure 4.7e) compare the control of the buck converter when the inductance value is different from the inductance value used in tuning conventional controllers and training the ANN controller. Normally, the performance of the controllers would be worse as the inductance value reduces, because this would make the dc/dc converter more likely to get out of continuous-conduction mode. Although more oscillations are shown with all the four control methods, the ANN has the smallest degradation in performance, especially for a large parameter variation away from its nominal value (Figure 4.7e), demonstrating its strong robust ability under system parameter variations.

4.5.4 Control Evaluation beyond System Constraints

Physical system constraints of the dc/dc converter are an important issue that needs to be addressed. Typically, there are two constraints: maximum duty-ratio constraint and inductor current constraint. Figure 4.8 evaluates the performance of the dc/dc converter using the ANN and conventional control strategies under the two physical constraint conditions. It is assumed in the simulation that the maximum inductor current is 2A. Also, the dc supply voltage is 30V.

For a fair comparison, the mechanism used to handle the PWM saturation limit shown in Figure 4.6 is applied to all the conventional methods. However, the mechanism used to handle the current limitation shown in Figure 4.6 cannot be applied to VMC, as this would result in high oscillations of the output voltage. For both the CMC and SMC, the current limit control is handled by the inner-loop current controller. For the ANN controller (Figure 4.8d), when the inductor current is over the maximum current limit, the controller is able to react immediately to maintain the output voltage at a lower value, while preventing the inductor current from exceeding the maximum limit; when a high reference voltage command is presented to the

controller, the ANN can maintain the output voltage at the highest voltage that can be outputted by the converter, while stability and controllability of the dc/dc converter are not affected before and after the maximum duty-ratio operation period.

The CMC cascade-PI and cascade SMC control structures can properly prevent the inductor current from exceeding the maximum inductor current limit too. But, the VMC Type III controller is unable to limit the inductor current, because the current limit control used for the ANN cannot be applied to the VMC as explained above. When using the locking mechanism presented in Figure 4.6 and Section III-C, all the three conventional methods can manage the PWM saturation constraint properly.

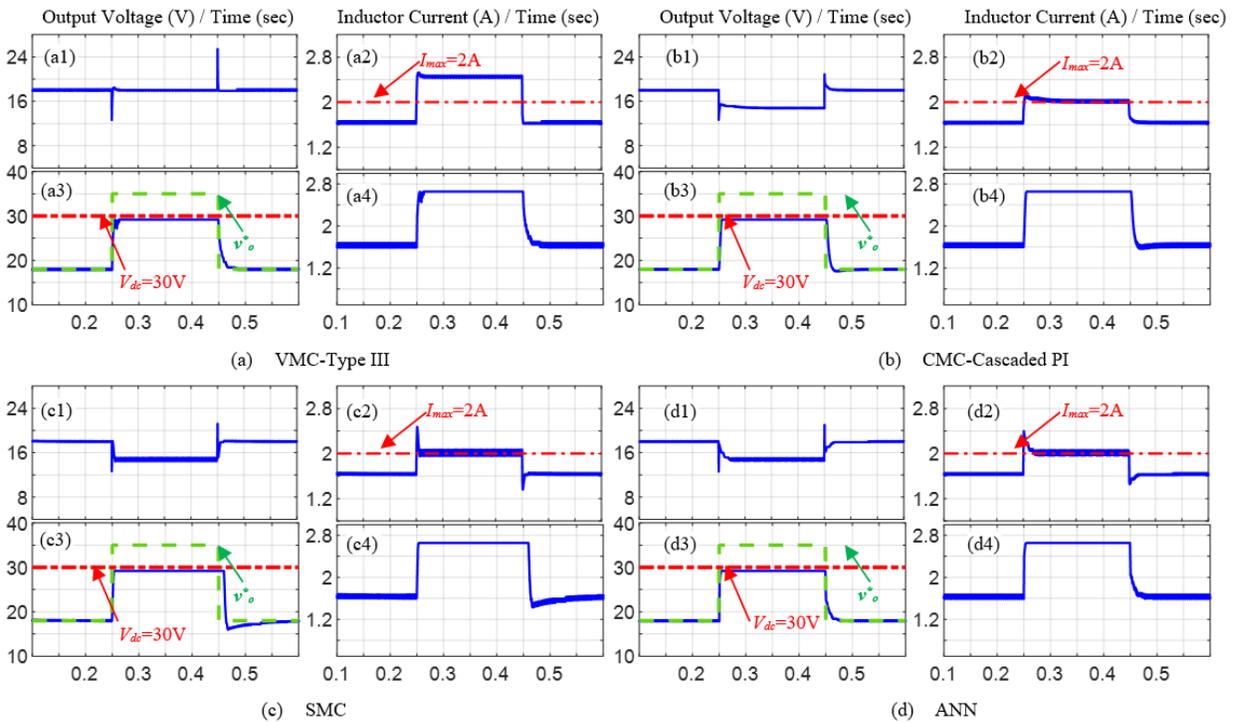


Figure 4.8: Simulated results for ANN vs. VMC Type III, CMC cascaded PI, and SMC under: a) maximum inductor current constraint of 2A (a1)-(a2), (b1)-(b2), (c1)-(c2), & (d1)-(d2); b) the maximum duty-ratio constraint (a3)-(a4), (b3)-(b4), (c3)-(c4), & (d3)-(d4)

4.6 Hardware experiment

4.6.1 Experiment setup

To further validate the proposed ANN controller, a DSP-based digital control system was implemented. The experimental setup (Figure 4.9) consists of four main parts: (i) *adc/dc* Buck converter built by using a three-phase converter board from Vishay HiRel Systems which has the maximum allowable switching frequency of 20kHz, (ii) a LabVolt LC circuit module representing the inductor and capacitor of the Buck converter, (iii) a dSPACE DS1103 controller board to collect inductor current and output voltage/current of the *dc/dc* converter, and (iv) a sensor board to convert measured voltage and current to dSPACE compatible format. The converter switching frequency is 10kHz and the controller sampling time is 0.1ms.

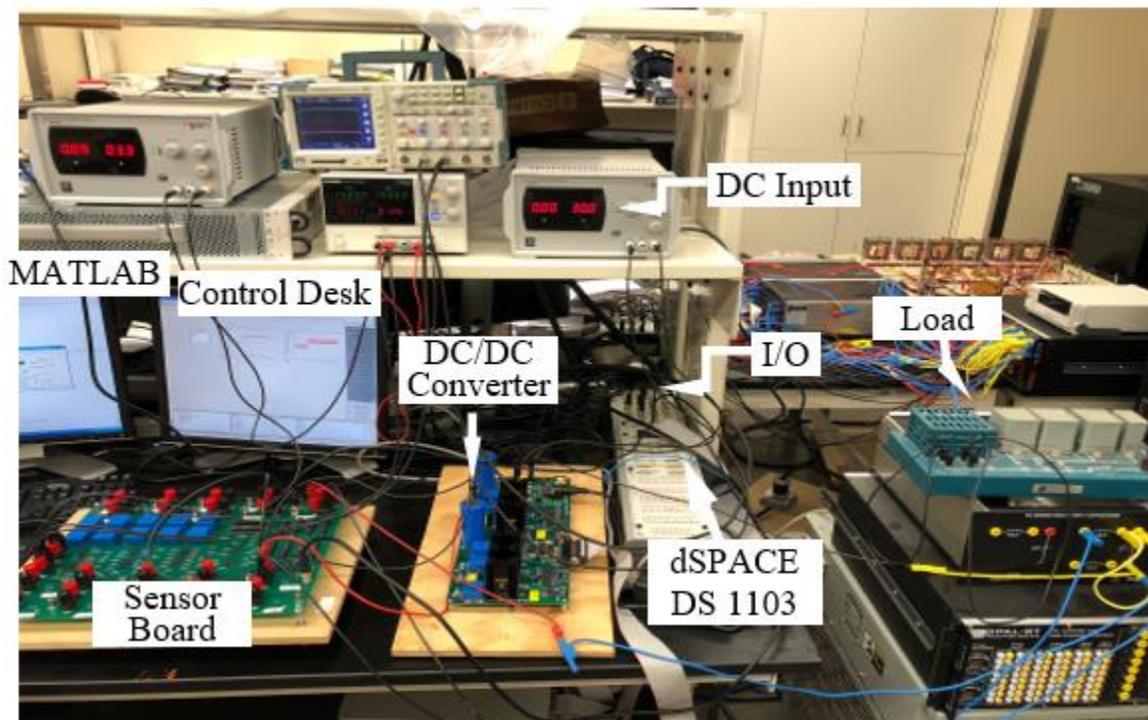


Figure 4.9: Hardware laboratory testing and control systems

4.6.2 Experiment Results

Figure 4.10 shows the comparison of VMC-Type III, CMC-Cascade PI, cascade SMC and ANN for control of the dc/dc Buck converter. The left side of Figure 4.10 shows the Buck converter's ability to follow a reference voltage change from 18V to 24V. Again, in the experiment condition, the ANN controller shows less overshoot and faster response speed than the conventional control methods in tracking the reference output voltage change. The right side of Figure 4.10 shows the Buck converter's ability to maintain a constant output voltage under a load change. As shown in the figure, the ANN controller presents a much stronger ability in maintaining output voltage stability under the variable load condition.

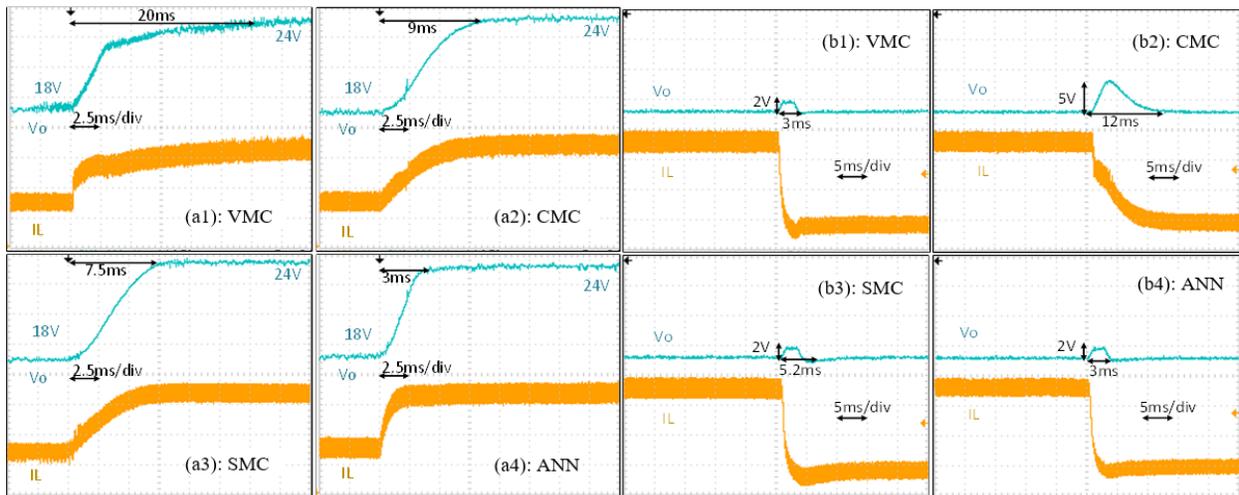


Figure 4.10: Hardware results for VMC vs. CMC Cascade PI vs. Cascade SMC vs. ANN: a) Change of v_o^* from 18V to 24V, b) Load change from 7.33Ω to 11Ω

Figure 4.11 shows the experiment results of the ANN controller when considering the maximum inductor current (2A) and duty-ratio constraints. In Figure 4.11a, when a load change causes the inductor current to be over the maximum current limit, the output voltage of the dc/dc converter is dropped automatically to regulate the inductor current within the maximum current limitation. In Figure 4.11, when the reference voltage increases and makes the duty-ratio over the

maximum duty-ratio constraint, the output voltage is automatically limited; and when the reference voltage reduces, the ANN controller is able to return to its normal condition.

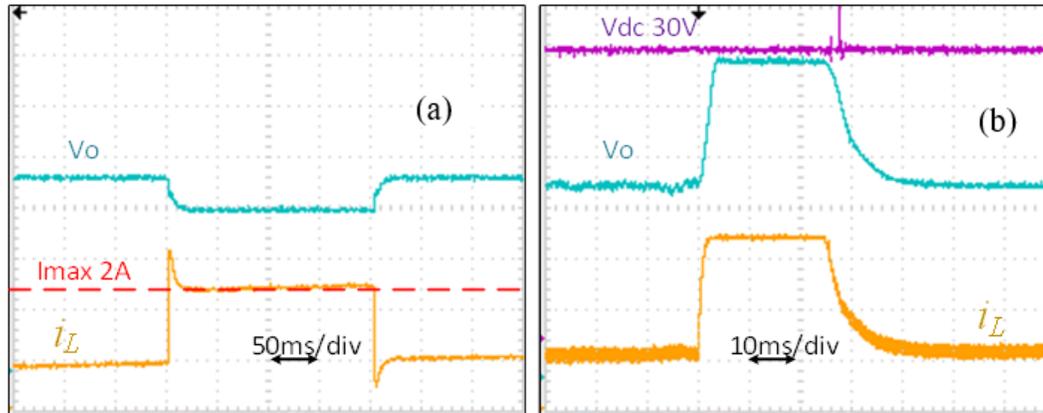


Figure 4.11: Hardware results for ANN under a) inductor current constraint; b) duty-ratio constraint

4.7 Conclusion

This chapter presents an ANN-based optimal and predictive control based on ADP for dc/dc Buck converters. Compared to conventional control methods, the ANN controller shows better performance in various aspects. In addition, the ANN controller can handle the control of the dc/dc converter properly under both the maximum inductor current and duty ratio constraints, while a conventional controller needs to have an inner-current loop through a cascade control structure to handle the current limit control. Compared to the conventional control methods, the ANN controller responds faster and maintains a more stable output voltage. The hardware experiment confirmed that the ANN controller is able to track reference commands, maintain output voltage stability under variable load and input voltage conditions, and manage the control of the dc/dc converter correctly under the maximum duty-ratio and inductor current constraints. The study shows that it is feasible to implement the ANN-based control for practical dc/dc Buck converters.

The proposed ANN control method can be extended to other dc/dc converters, such as Boost and Buck-Boost converters. However, since the state-space models of the Boost and Buck-Boost converters are different from that of the Buck converter, the training algorithms for each of the other dc/dc converters need to be redesigned and revalidated. We plan to extend the proposed ANN-ADP control method to other dc/dc converters through future research.

4.8 References

- [1] P. J. d. S. Neto, T. A. d. S. Barros, J. P. C. Silveira, E. R. Filho, J. C. Vasquez and J. M. Guerrero, "Power Management Strategy Based on Virtual Inertia for DC Microgrids," in *IEEE Transactions on Power Electronics*, vol. 35, no. 11, pp. 12472-12485, Nov. 2020, doi: 10.1109/TPEL.2020.2986283.
- [2] W. W. A. G. Silva, T. R. Oliveira and P. F. Donoso-Garcia, "An Improved Voltage-Shifting Strategy to Attain Concomitant Accurate Power Sharing and Voltage Restoration in Droop-Controlled DC Microgrids," in *IEEE Transactions on Power Electronics*, vol. 36, no. 2, pp. 2396-2406, Feb. 2021, doi: 10.1109/TPEL.2020.3009619.
- [3] Z. Xia and J. Abu Qahouq, "State-of-charge Balancing of Lithium-ion Batteries with State-of-health Awareness Capability," in *IEEE Transactions on Industry Applications*, doi: 10.1109/TIA.2020.3029755.
- [4] M. S. Sadabadi, "A Distributed Control Strategy for Parallel DC-DC Converters," in *IEEE Control Systems Letters*, vol. 5, no. 4, pp. 1231-1236, Oct. 2021, doi: 10.1109/LCSYS.2020.3025411.
- [5] N. Mohan, *Power Electronics: A First Course*, Wiley, 2011
- [6] R.A. Mammano, *Fundamentals of Power Supply Design*, Texas Instruments, 2017.
- [7] S. Tan, Y. M. Lai and C. K. Tse, "General Design Issues of Sliding-Mode Controllers in DC-DC Converters," *IEEE Trans. Ind. Electron.*, vol. 55, no. 3, pp. 1160-1174, March 2008.
- [8] M. Ordonez, M. T. Iqbal and J. E. Quicoe, "Selection of a curved switching surface for buck converters," *IEEE Trans. on Power Electron.*, vol. 21, no. 4, pp. 1148-1153, July 2006.

- [9] R. Ling, D. Maksimovic and R. Leyva, "Second-Order Sliding-Mode Controlled Synchronous Buck DC–DC Converter," *IEEE Trans. on Power Electron.*, vol. 31, no. 3, pp. 2539-2549, March 2016.
- [10] Cheng, Zhiping, et al. "A novel cascaded control to improve stability and inertia of parallel buck-boost converters in DC microgrid." *International Journal of Electrical Power & Energy Systems* 119 (2020): 105950.
- [11] Z. Wang, S. Li and Q. Li, "Continuous Nonsingular Terminal Sliding Mode Control of DC–DC Boost Converters Subject to Time-Varying Disturbances," in *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 67, no. 11, pp. 2552-2556, Nov. 2020
- [12] B. Wang, V. R. K. Kanamarlapudi, L. Xian, X. Peng, K. T. Tan and P. L. So, "Model Predictive Voltage Control for Single-Inductor Multiple-Output DC–DC Converter With Reduced Cross Regulation," in *IEEE Transactions on Industrial Electronics*, vol. 63, no. 7, pp. 4187-4197, July 2016.
- [13] Z. Liu, L. Xie, A. Bemporad and S. Lu, "Fast Linear Parameter Varying Model Predictive Control of Buck DC-DC Converters Based on FPGA," in *IEEE Access*, vol. 6, pp. 52434-52446, 2018, doi: 10.1109/ACCESS.2018.2869043.
- [14] J. Mahdavi, M.R. Nasiri, A. Agah, and A. Emadi, "Application of neural networks and state-space averaging to DC/DC PWM converters in sliding-mode operation," *IEEE/ASME Trans. Mechatronics*, 10(1), 2005 pp. 60-67.
- [15] A. Rubaai, A.R. Ofoli, L. Burge, and M. Garuba, "Hardware implementation of an adaptive network-based fuzzy controller for DC-DC converters," *IEEE Trans. Ind. Appl.*, 41(6), 2005, pp.1557-1565.
- [16] R.J. Wai and L.C. Shih, "Adaptive fuzzy-neural-network design for voltage tracking control of a DC–DC boost converter," *IEEE Trans. Power Electron.*, 27(4), 2012, pp.2104-2115.
- [17] Dimitri P. Bertsekas, "Dynamic Programming and Optimal Control: Approximate Dynamic Programming," 4th Ed., Athena Scientific, 2012.
- [18] F.L. Lewis and D. Liu (eds.), *Reinforcement Learning and Approximate Dynamic Programming for Feedback Control*, IEEE Press / Wiley, 2012, pp. 474-493.
- [19] H. Zhang, C. Li, X. Zhang, and Y. Luo, "Data-Driven Robust Approximate Optimal Tracking Control for Unknown General Nonlinear Systems Using Adaptive Dynamic Programming Method," *IEEE Trans. Neural Netw.*, 22(12), 2011, pp. 2226-2236.
- [20] A. Heydari, "Optimal Switching of DC–DC Power Converters Using Approximate Dynamic Programming," in *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 3, pp. 586-596, March 2018.

- [21] S. Li, M. Fairbank, C. Johnson, D.C. Wunsch, E. Alonso, and J.L. Proao, "Artificial neural networks for control of a grid-connected rectifier/inverter under disturbance, dynamic and power converter switching conditions," *IEEE Trans. Neural Netw. Learn. Syst.*, 25(4), 2014, pp. 738–750.
- [22] E. d. C. Gomes, L. A. de S. Ribeiro, J. V. M. Caracas, S. Y. C. Catunda and R. D. Lorenz, "State space decoupling control design methodology for switching converters," 2010 IEEE Energy Conversion Congress and Exposition, Atlanta, GA, 2010, pp. 4151-4158, doi: 10.1109/ECCE.2010.5617741.
- [23] K.M. Tsang, and W.L. Chan. "Cascade controller for DC/DC buck convertor." *IEE Proc. Electr, Power Appl.*, 152(4), (2005), pp. 827-831.
- [24] Z. Li, C. Zang, P. Zeng, H. Yu, S. Li and J. Bian, "Control of a Grid-Forming Inverter Based on Sliding-Mode and Mixed H_2/H_∞ Control," *IEEE Trans. Ind. Electron.*, vol. 64, no. 5, pp. 3862-3872, May 2017.
- [25] N. Mohan, T.M. Undeland, and W.P. Robbins, *Power Electronics: Converters, Applications, and Design*, 3rd Ed., John Wiley & Sons Inc., Oct. 2002.
- [26] G.F. Franklin, J.D. Powell, M.L. Workman, *Digital control of dynamic systems*, 3rd edition, Addison-Wesley, 1998.
- [27] M. Hagan and M. Menhaj, "Training feedforward networks with the Marquardt algorithm," *IEEE Trans. Neural Netw.*, vol. 5(6), pp. 989-993, Nov. 1994.
- [28] X. Fu, S. Li, M. Fairbank, D. C. Wunsch, and E. Alonso, "Training recurrent neural networks with the Levenberg-Marquardt algorithm for optimal control of a grid connected converter," *IEEE Trans. Neural Netw. Learn. Syst.*, Oct. 2014.

5. ARTIFICIAL NEURAL NETWORK CONTROL OF A STANDALONE DC MICROGRID

5.1 Introduction

The concept of microgrid (MG) is becoming more and more important and popular due to the development of DGs (Distributed Generation) [1]. It has the potential to integrate different DGs and ESSs (Energy Storage Systems) for many different applications [2].

There are generally two types of microgrids: AC microgrids and DC microgrids [3]-[4]. AC microgrids have been studied and implemented by many researchers. Recently, DC microgrids also start to draw a wide attention for many reasons, such less energy conversion, lower losses, and their controllability [5]. In particular, some generations and loads are inherently DC, such as solar photovoltaic (PV), battery, electrical vehicle (EV), etc [6]-[9], where a DC microgrid would integrate them more naturally and improve system efficiency and reliability.

There are two operation modes for a DC microgrid: grid-tied and islanded modes [10]. In the grid-tied mode, a DC microgrid is connected to the main AC grid through DC/AC inverters and the DC microgrid can absorb power or provide power to the main grid. At the same time, the voltage is maintained by the main grid. In the islanded mode, it is necessary to maintain the bus voltages and meanwhile provide the active powers of loads. One of the most popular control techniques for DC microgrid operation in islanded mode and grid connected mode is the droop control technique, which maintains MG voltage stability and uses the droop gains to share power among distributed generations (DGs) [10]-[12].

Recently, an artificial neural network (ANN) control strategy has been developed for control of voltage source DC/AC converters in different applications [13], including control of inverters in AC microgrids. The ANN control has shown a very fast response time, strong stability, and reliability. It has a potential to improve frequency and voltage stability of an AC microgrid, enhance power quality, and assure fast response from one mode to another. However, no research has been done to develop ANN-based control for DC/DC converters and apply ANNs to control of DC microgrids.

This research proposes a novel artificial neural network (ANN) based control method, integrated with droop control, for control of an islanded DC microgrid. The ANN controller is trained based on ADP (approximate dynamic programming) using LM (Levenberg-Marquardt) algorithm. A FATT (Forward Accumulation Through Time) algorithm is applied to calculate Jacobian matrix. The ANN performance is evaluated by using switching models of power DC converters. Performance of ANN in DC microgrid shows that the proposed controller has the ability to maintain voltage stability of standalone DC microgrid and manage the power sharing among the parallel-connected distributed generation units. For different transient scenarios, the ANN controller in DC microgrids also performs very well to tolerate load disturbances and track voltage references rapidly.

This chapter presents a preliminary study on developing an ANN-based control for DC/DC converters and integrate it with droop mechanism for control of a standalone DC microgrid. In the following sections, the paper first presents a typical DC microgrid structure in Section 5.2. Section III presents an ANN-based control method as the primary control layer for microgrid DC/DC converters as well as how to train the ANN to realize optimal control via ADP (approximate dynamic programming). Section 5.4 presents how to integrate droop and ANN

control algorithm to handle the DC microgrid operation in different scenarios. Section 5.5 presents simulation evaluation of the proposed approach for control of a DC microgrid in islanded mode. Finally, the results are concluded, and the future work is presented.

5.2 Standalone DC microgrid

5.2.1 Configuration of a standalone DC microgrid

The configuration of a typical DC MG is shown by Figure 5.1. The DC microgrid is tied to the AC main grid through a bidirectional DC/AC converter and a switch. DG sources within the MG may include PV arrays, fuel cells, EVs, and battery storage, each of which is connected to a DC bus through a DC/DC converter. The PV array and fuel cell powers are always generating and unidirectional from the PV array or fuel cell to the DC bus while the EV and battery power can be generating or absorbing. Normally, the PV array operates in MPPT (maximum power point tracking) mode, and thus its power is not controllable. However, the power of EV and battery can be regulated to maintain voltage and power sharing of the DC microgrid, especially in standalone mode. Loads in the DC microgrid may include: 1) normal DC loads directly connected to DC bus; 2) sensitive DC loads, each of which is connected to the microgrid through a DC/DC converter; 3) AC loads connected to the DC grid through inverters. The DC microgrid operates in grid-tied mode when it's connected to the main AC or DC grid, the DC microgrid operates in islanded mode when it's disconnected to the main AC or DC grid, and it has to provide the stable voltage and manage the power sharing among the DGs, energy storage system, and loads.

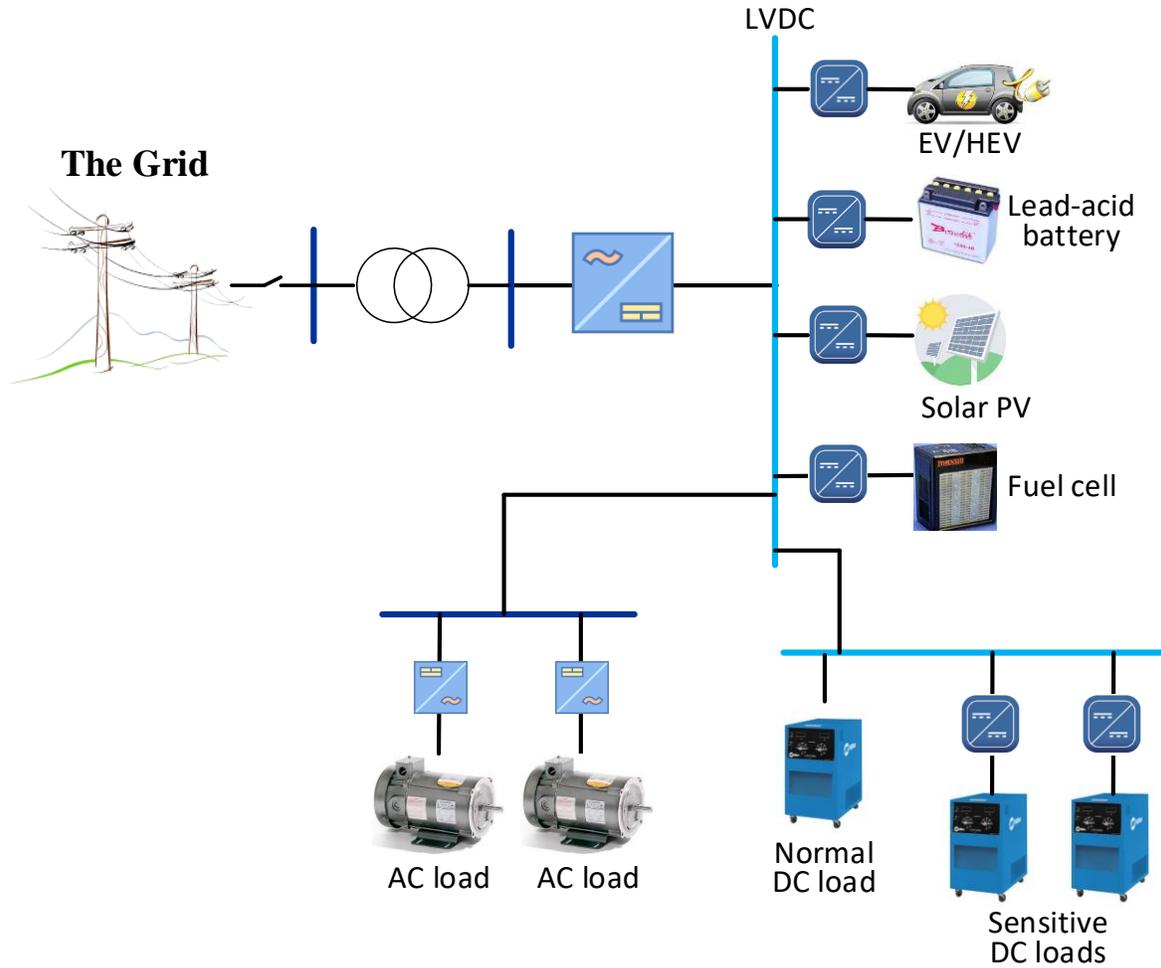


Figure 5.1: configuration of a DC microgrid

5.2.2 DC/DC converter model with loads

According to Section 4.2.1 and Figure 5.1, the structure of a DC microgrid in the standalone condition is a DC source and load which are connected through a DC/DC converter. The structure is shown in Figure.2. The purpose of the controller of the DC/DC converter is to maintain voltage stability of the loads and provide proper power sharing DG units. To develop an ANN controller, the topology of the system shown by Figure 5.2 is needed. Using power converter average model, Kirchoff's current/voltage law, the voltage at point A and the inductor current of the DC/DC converter are

$$v_A = R_L i_L + L \frac{di_L}{dt} + v_c \quad (5-1)$$

$$C \frac{dv_c}{dt} = i_L + \frac{v_c}{R} \quad (5-2)$$

where R_L and L are the internal resistance and inductance of the inductor, C is the capacitance of the smoothing capacitor, v_A is output voltage of the switch of DC/DC converter, v_c is the capacitor or the voltage at DC PCC (Point of common coupling), i_L is the inductor current, and R represents load resistance.

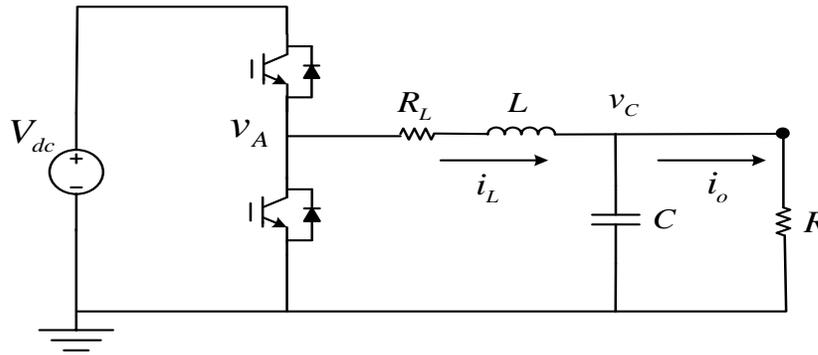


Figure 5.2: A standalone DC/DC converter with loads

5.3 ANN control of a standalone DC/DC converter

In the standalone case, the purpose is maintain the DC/DC output voltage and, meanwhile, manage power sharing through integrating voltage control with droop control as shown in Section 4.4.

5.3.1 DC/DC Converter State-Space Model

The ANN controller is developed based on the state-space model of the DC/DC converter. The state-space model is derived from (5-1) (5-2). The model is shown below:

$$\frac{d}{dt} \begin{pmatrix} i_L \\ v_c \end{pmatrix} = - \begin{pmatrix} R_L/L & -1/L \\ 1/C & -1/RC \end{pmatrix} \begin{pmatrix} i_L \\ v_c \end{pmatrix} + \begin{pmatrix} v_A/L \\ 0 \end{pmatrix} \quad (5-3)$$

where the system states are i_L and v_c . The converter output voltages v_A is controlled by the switches with signal voltage generated by the ANN controller [14].

The ANN controller is developed as a digital controller. Therefore, the digital model of the DC/DC converter is needed. Using a zero-order hold transformation, the discrete state-space model can be represented as:

$$\begin{pmatrix} i_L(kT_s + T_s) \\ v_c(kT_s + T_s) \end{pmatrix} = \mathbf{A} \begin{pmatrix} i_L(kT_s) \\ v_c(kT_s) \end{pmatrix} + \mathbf{B} \begin{pmatrix} v_A(kT_s) \\ 0 \end{pmatrix} \quad (5-4)$$

in which T_s is the system sampling time. T_s can be further eliminated, and the modified model is:

$$\begin{pmatrix} i_L(k+1) \\ v_c(k+1) \end{pmatrix} = \mathbf{A} \begin{pmatrix} i_L(k) \\ v_c(k) \end{pmatrix} + \mathbf{B} \begin{pmatrix} v_A(k) \\ 0 \end{pmatrix}, \quad (5-5)$$

where k is an integer time step.

5.3.2 ANN Controller Structure

The proposed configuration of the ANN controller is shown in the lower part of Figure 5.3. There are four different layers in the ANN controller: an input layer, two hidden layers, and an output layer. There are two inputs in the configuration: the error term and the integral of the error term:

$$e_{v_c}(k) = v_c(k) - v_c^*(k), \quad s_{v_c}(k) = \int_0^k e_{v_c}(t) dt, \quad (5-6)$$

where $v_c^*(k)$ is the reference DC PCC voltage. As shown by (5-6), we can see that similar as the traditional PI controllers, the input signals for the ANN control are the same: one is the error terms, and another is integrals of error. Hence, the ANN controller has the simple structure as traditional PI but with better response. The proper gains are selected and divide the inputs, and then processed through a hyperbolic tangent function. Each node at the hidden layers uses a hyperbolic-tangent activation function. After the calculation, $v_A^*(k)$ is generated from the output layer as the output of the controller. $v_A^*(k)$ is the normalized signal. Therefore, for the switching power converter, the PWM gain is needed [14, 15], to have the DC/DC converter control signal, v_A , given by:

$$v_A(k) = k_{PWM} \cdot A(e_{v_c}(k), s_{v_c}(k), \vec{w}), \quad (5-7)$$

where \vec{w} is the ANN weight vector, and A is the function for overall network.

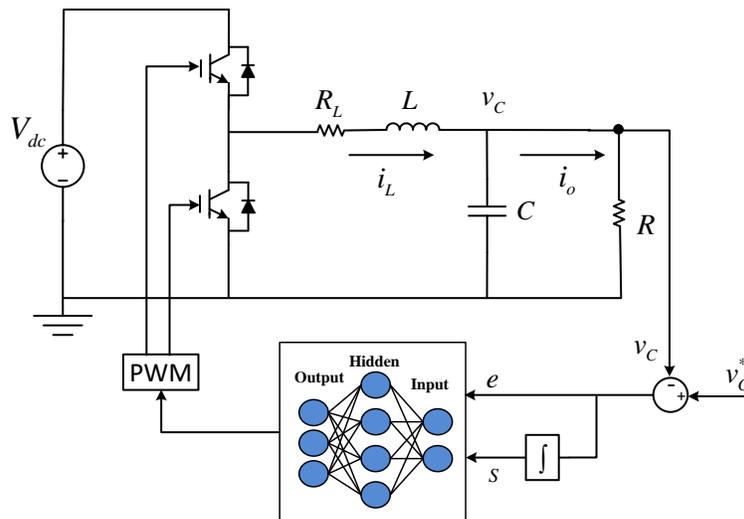


Figure 5.3: ANN control for a standalone DC/DC converter with loads

5.3.3 Training ANN Controller

The ANN controller needs to be trained using approximate dynamic programming (ADP) to have the optimal performance [16]. The approximate dynamic programming cost-to-go function used for the DC microgrid application is:

$$C(v_c) = \sum_{k=1}^N [v_c(k) - v_c^*(k)]^2 \quad (5-8)$$

The cost-to-go function (5-8) needs to be minimized by training the ANN to achieve the optimal control using ADP, the. In the closed-loop control environment, the ANN has the recurrent network nature as the output of the DC/DC converter serves as a feedback to the input of the network at the next time step. For training the recurrent network, we need to calculate according to the gradient of (5-8) the weight vector $\partial C / \partial \vec{w}$ and the chain rule. When the inputs are obtained, the calculation begins at cost-to-go function (5-8) and the calculated by (5-4), (5-6) and (5-7), and then based on the chain-rule, the calculated results are sent back to the ANN input as the next step. The Levenberg-Marquardt (LM) algorithm is used to train the network, and then a Forward Accumulation Through Time (FATT) algorithm is employed to calculated Jacobian matrix. A more thorough explanation about training is available in [17]. After that, the adjustment of ANN weights $\Delta \vec{w}$ for a training epoch can be acquired, and the weights of the network are updated by

$$\vec{w}_{\text{update}} = \vec{w} + \Delta \vec{w} \quad (5-9)$$

When the process meets a criterion, it can be stopped. The network weights are adjusted repeatedly to minimize the function of ADP cost-to-go (5-8). After successful training, PCC

voltage is optimized by ANN based on ADP, and the ANN would be able to track the voltage reference.

5.4 Integrating ANN with Droop Control

5.4.1 Droop Control Method for DC DG Units

In practical situation, a DC converter simplified concept is shown in Figure 5.1. The unit can be connected to a DC bus through a transmission line with resistance R_{dc} as shown by Figure.4. The power P_{dc} provided by the DG equals:

$$P_{dc} = E_{dc} \cdot I_{dc} \quad (5-10)$$

where E_{dc} is the converter output voltage or voltage at the capacitor of the unit (DC/DC converter) and I_{dc} is the transmission line current:

$$I_{dc} = (E_{dc} - V_{dc}) / R_{dc} \quad (5-11)$$

where V_{dc} is the DC bus voltage and R_{dc} is the line resistance.

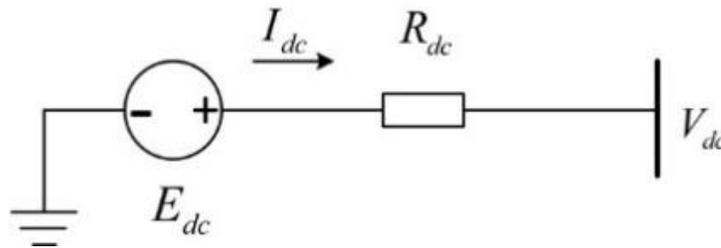


Figure 5.4: A DC DG unit model in DC microgrid

From (5-10) and (5-11), it could be concluded that the power provided by a DG is dependent on the transmission line and voltage difference [10]. The equation to represent DC droop technique is expressed as:

$$E_{dc} = E_{dc}^* - R_d P_{dc} \quad (5-12)$$

where E_{dc}^* is the reference DC converter output voltage, and R_d is called droop gain. Eq. (5-12) is depicted graphically in Figure 5.5. In general, as the power output of the DG increases, its output voltage drops [18].

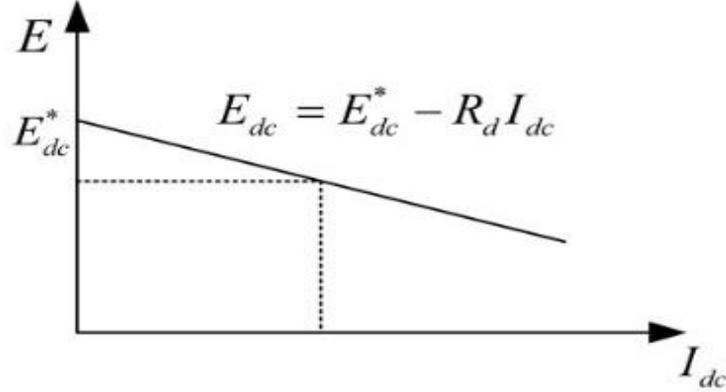


Figure 5.5: Illustration of DC droop technique.

5.4.2 Power Sharing Control among Parallel DG Units

If the two or more units, having same capacity, The power sharing between them should have the same amount [18]. However, with the different transmission line, the units with same capability still cannot share the equal power (Figure 5.6), the following according to the droop characteristics can be derived

$$V_{load} = \begin{cases} E_{dc}^* - R_{d1} I_{dc1} - R_{line1} I_{dc1} \\ E_{dc}^* - R_{d2} I_{dc2} - R_{line2} I_{dc2} \end{cases} \quad (5-13)$$

Then, it can be obtained

$$\frac{I_{dc1}}{I_{dc2}} = \frac{R_{d1}}{R_{d2}} + \frac{R_{line2} - R_{line1} R_{d2}/R_{d1}}{R_{d1} + R_{line1}} \quad (5-14)$$

which shows that the current sharing error between the two DC converters cannot be eliminated unless (14) is satisfied in designing the droop gain of each converters. This is the power sharing issue and needs to be considered in the integrated ANN and droop control design.

$$R_{d1}/R_{d2} = R_{line1}/R_{line2} \quad (5-15)$$

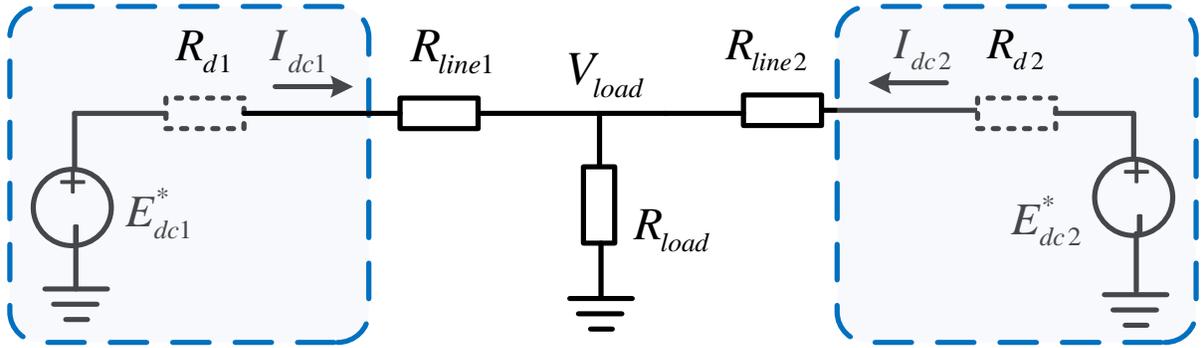


Figure 5.6: Two DG units connected to a common bus

5.4.3 Integrating ANN with Droop Control

The proposed controller structure with ANN controllers and droop is based on the DC microgrid system shown by Figure 5.1. Basically, for each DG unit, the droop controller handles the power sharing among multiple parallel DG units based on droop gains while the ANN controller maintains the voltage at each DC unit based the reference voltages signals generated by the droop controller (Figure 5.7).

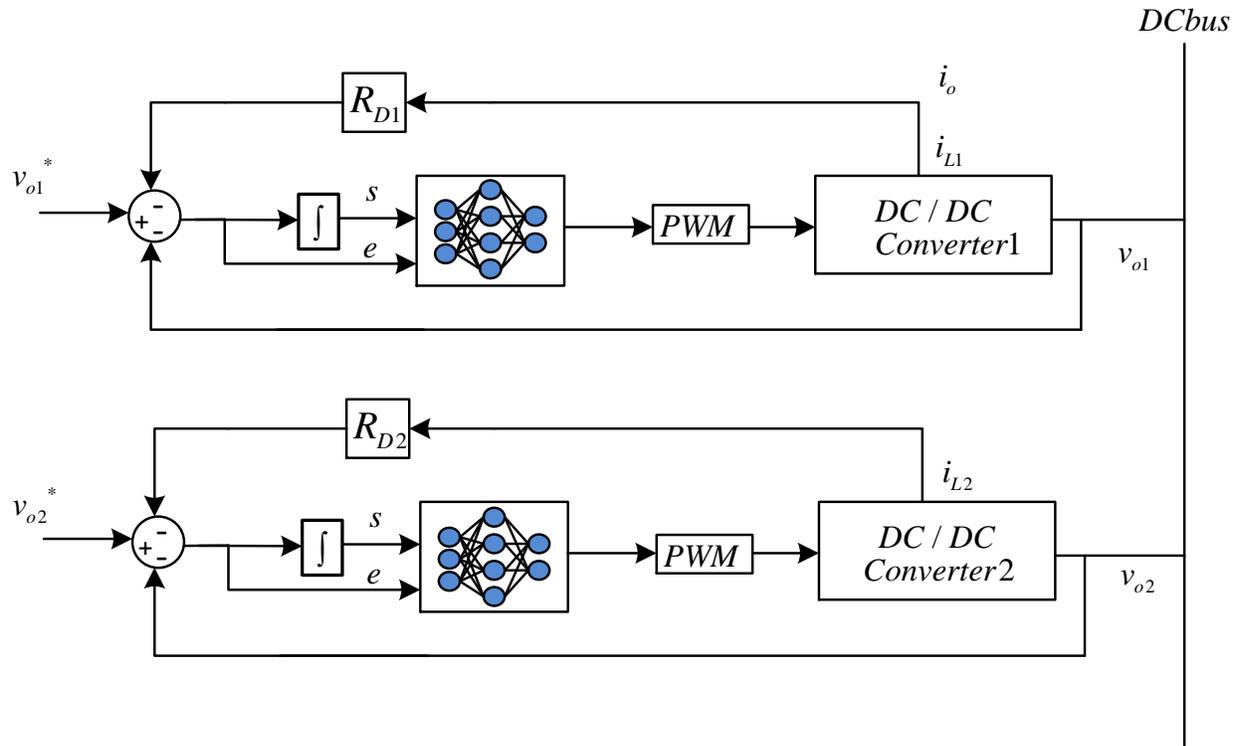


Figure 5.7: Droop, virtual resistance and ANN control integration

The design of the droop controller of each DG unit considered the power sharing requirement based on the capacity of each unit, and then the virtual resistance was included to assure that the condition of (5-15) is satisfied. The droop controller together with the virtual impedance generates a voltage reference which is sent to the ANN controller.

An ANN controller can be operated in the DC microgrid system after being trained. The ANN training was based on the model and topology of the DC/DC converter shown by Figure 5.2 instead of Figure 5.1. The ANN was trained over and over again and the different reference voltage trajectories are tracked to obtain the best tracking performance. It is possible to have different cost because the network weight is randomly picked at the beginning of the ANN training. The weights can be chosen when the lowest ADP costs are found. After the optimal

training is finished, the ANN controller is used to regulate the DC/DC converter output voltage based on the error signal between the measured and reference PCC voltage. However, since the ANNs were trained independently according to Figure 5.2, detailed validation of the controllers integration is important and needed. Simulation evaluation of the proposed strategy is discussed in Section 5.5.

5.5 Simulation Evaluation

The proposed ANN control of stand-alone DC microgrid is verified using Simulink. The model is shown in Figure 5.8. The nominal output voltage is 400V, and the nominal DC input voltage is 600V. In the simulation, two DC/DC converters are connected in parallel with a changing load through transmission lines with the equal resistance of 0.5Ω . *Table 2* shows the parameters of the DC microgrid system. The virtual resistances for the two converters are 2Ω and 1.5Ω , which are determined by considering transmission line resistance and power sharing between the two converter units based on the power rating of the two units. Simulation results are shown in Figure 9.

Table 2 Parameters of DC MG

Parameters	Value
Inductance L	600 μ H
Capacitance C	42 μ F
Load resistance R	12 Ω
Switching frequency f	50kHz

R_{D1} / R_{D2}	$2\Omega / 1.5\Omega$
-------------------	-----------------------

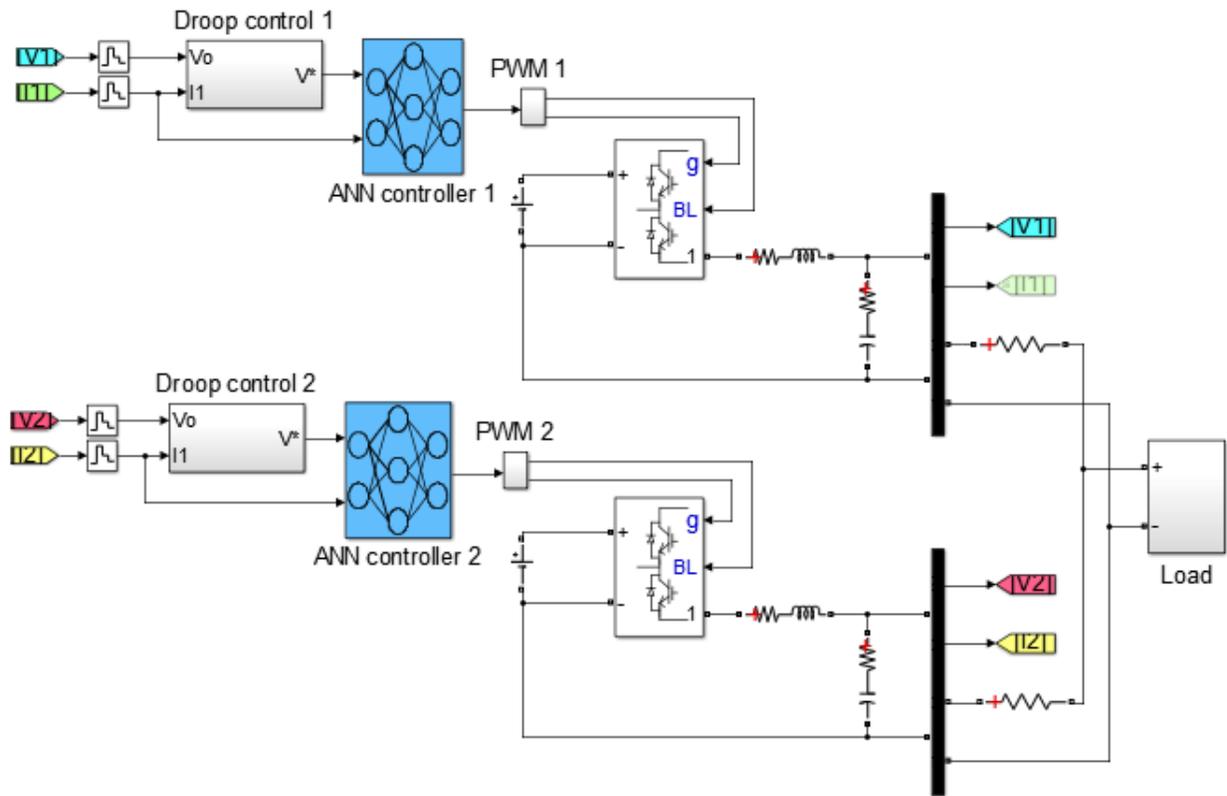


Figure 5.8: Stand-alone DC microgrid with ANN and droop control

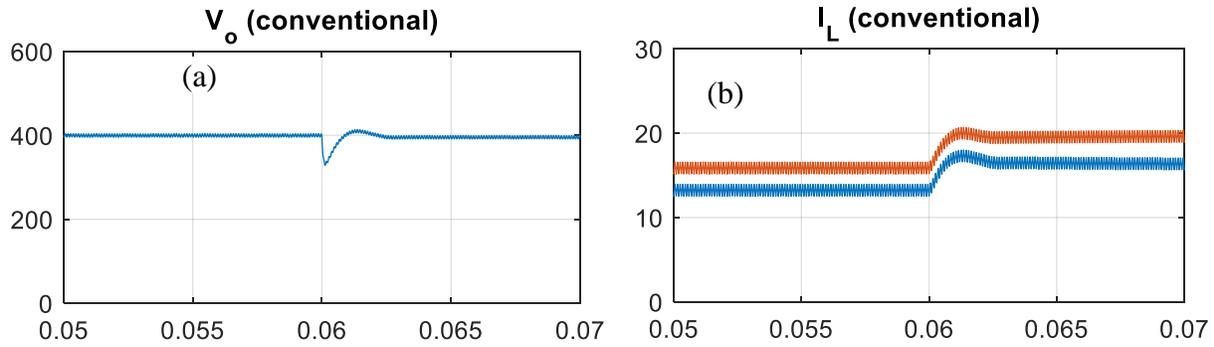


Figure 5.9: (a) Output voltage and (b) output currents of the converters with conventional controller

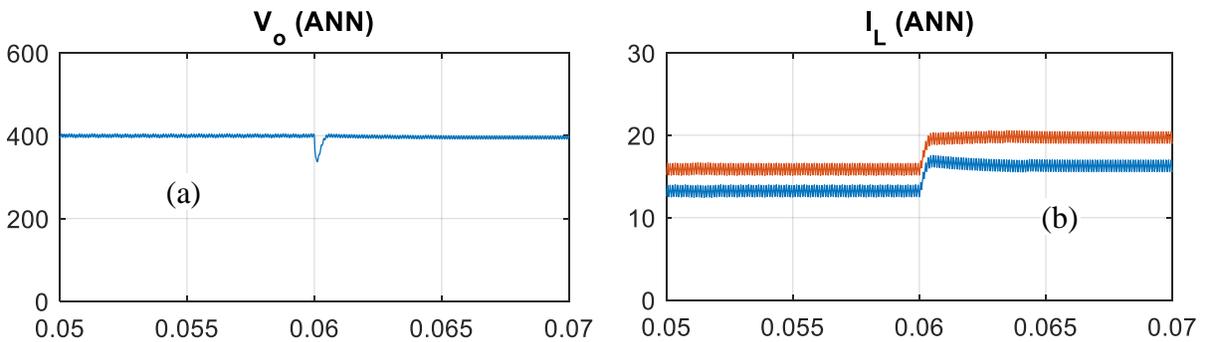


Figure 5.10: (a) Output voltage and (b) output currents of the converters with ANN controller

Figure 5.9 and Figure 5.10 show the performance of stand-alone DC microgrid with ANN and droop control. Figure 5.9(a) shows the output voltage, and Figure 5.9(b) shows the output currents of the two DC/DC converters. It can be seen from the figure that the output voltage of the converters at the load is maintained around the desired value during the entire simulation, and the ANN controller has a better performance. Before $t = 0.06$ sec, the load current is properly shared by the two converters according to the virtual resistance ratio of $R_{D2}:R_{D1}$ defined in the droop controllers. At $t = 0.06$ sec, the load at DC bus decreases, and it causes output currents of the two DC/DC converters to increase to meet the new power demand of the loads. The figure shows that the output currents of the converters change rapidly to reach the new demanded load current using the ANN control. Again, after $t = 0.06$ sec, the

load current is properly shared according to the ratio of their droop gains. There is a slight voltage drop after 0.06 sec due to the droop control and virtual resistances. At the same time, the output currents are affected a little by the transmission line.

5.6 Conclusions

The operation of a standalone DC microgrid has many challenges since the voltage stability and power sharing must be handled by the DC microgrid itself. This paper presents a preliminary study on developing an ANN-based control for DC/DC converters and integrating it with droop mechanism for control of a standalone DC microgrid. This paper describes how the ANN-based control is developed, and how it works with droop control to manage microgrid operation in standalone conditions.

The system performance evaluation shows that the bus voltage can be regulated at the desired value and the output power of the converters can be regulated rapidly when there are reference or load changes. The paper shows that the proposed strategy can be implemented and well-performed in a DC microgrid. It has the ability to maintain voltage stability of standalone DC microgrid and manage the power sharing among the parallel-connected distributed generation units. For different transient scenarios, the ANN controller in DC microgrids also performs well to track voltage references rapidly and tolerate load disturbances.

5.7 References

- [1] *Magazine*, vol. 8, no. 1, pp. 18-28, January-February 2010.
- [2] K. Kurohane, T. Senjyu, A. Yona, N. Urasaki, T. Goya and T. Funabashi, "A Hybrid Smart AC/DC Power System," in *IEEE Transactions on Smart Grid*, vol. 1, no. 2, pp. 199-204, Sept. 2010.

- [3] H. Han, X. Hou, J. Yang, J. Wu, M. Su and J. M. Guerrero, "Review of Power Sharing Control Strategies for Islanding Operation of AC Microgrids," in *IEEE Transactions on Smart Grid*, vol. 7, no. 1, pp. 200-215, Jan. 2016.
- [4] A. Kahrobaeian and Y. A. R. I. Mohamed, "Interactive Distributed Generation Interface for Flexible Micro-Grid Operation in Smart Distribution Systems," *IEEE Trans. Sust. Energy*, vol. 3, no. 2, pp. 295- 305, April 2012.
- [5] A. Mehrizi-Sani and R. Iravani, "Potential-Function Based Control of a Microgrid in Islanded and Grid-Connected Modes," *IEEE Trans. Power Syst.*, vol. 25, no. 4, pp. 1883-1891, Nov. 2010.
- [6] T. Dragicevic, J. C. Vasquez, J. M. Guerrero and D. Skrlec, "Advanced LVDC Electrical Power Architectures and Microgrids: A step toward a new generation of power distribution networks.," *IEEE Electrification Magazine*, vol. 2, no. 1, pp. 54-65, March 2014.
- [7] Justo, J.J., Mwasilu, F., Lee, J. and Jung, J.W., " AC-microgrids versus DC-microgrids with distributed energy resources: A review," *Renewable and Sustainable Energy Reviews*, 24, pp.387-405, 2013
- [8] U. Manandhar, A. Ukil and T. K. K. Jonathan, "Efficiency comparison of DC and AC microgrid," *Smart Grid Technologies - Asia (ISGT ASIA), 2015 IEEE Innovative*, Bangkok, 2015, pp. 1-6.
- [9] H. Lotfi; A. Khodaei, "AC Versus DC Microgrid Planning," in *IEEE Transactions on Smart Grid* , vol.PP, no.99, pp.1-1
- [10] J. M. Guerrero, J. C. Vasquez, J. Matas, L. G. de Vicuna and M. Castilla, "Hierarchical Control of Droop-Controlled AC and DC Microgrids—A General Approach Toward Standardization," in *IEEE Transactions on Industrial Electronics*, vol. 58, no. 1, pp. 158-172, Jan. 2011.
- [11] H. Kakigano, Y. Miura, T. Ise and R. Uchida, "DC micro-grid for super high quality distribution — System configuration and control of distributed generations and energy storage devices," 2006 37th IEEE Power Electronics Specialists Conference, Jeju, 2006, pp. 1-7.
- [12] J. Schonbergerschonberger, R. Duke and S. D. Round, "DC-Bus Signaling: A Distributed Control Strategy for a Hybrid Renewable Nanogrid," in *IEEE Transactions on Industrial Electronics*, vol. 53, no. 5, pp. 1453-1460, Oct. 2006.
- [13] S. Li, M. Fairbank, C. Johnson, D.C. Wunsch and E. Alonso, "Artificial Neural Networks for Control of a Grid-Connected Rectifier/Inverter under Disturbance, Dynamic and Power Converter Switching Conditions," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 4, pp. 738–750, Apr. 2014.

- [14] N. Mohan, T.M. Undeland, and W.P. Robbins, *Power Electronics: Converters, Applications, and Design*, 3rd Ed., John Wiley & Sons Inc., Oct. 2002.
- [15] N. Mohan, *Advanced Electric Drives – Analysis, Modeling and Control using Simulink*, MN: Minnesota Power Electronics Research & Education, ISBN 0-9715292-0-5, 2001.
- [16] F.Y. Wang, H. Zhang, and D. Liu, "Adaptive dynamic programming: An introduction," *IEEE Comput. Intell. Mag.*, pp. 39–47, 2009.
- [17] X. Fu, S. Li, M. Fairbank, D. C. Wunsch, and E. Alonso, "Training recurrent neural networks with the Levenberg-Marquardt algorithm for optimal control of a grid connected converter," *IEEE Trans. Neural Netw. Learn. Syst.*, Oct. 2014.
- [18] S. Anand, B. G. Fernandes and J. Guerrero, "Distributed Control to Ensure Proportional Load Sharing and Improve Voltage Regulation in Low-Voltage DC Microgrids," in *IEEE Transactions on Power Electronics*, vol. 28, no. 4, pp. 1900-1913, April 2013.

6. CONCLUSIONS AND FUTURE WORKS

6.1 Conclusions

In this dissertation, the control and operation of EV machine drives and V2G applications are investigated. The artificial intelligence-based control algorithms especially the ANN control are designed and implemented for the high-efficiency motor control and machine modeling. The DC/DC converter and DC microgrid control using ADP, ANN and droop are presented. Multiple controllers sliding mode control, LUT-based control, and other AI-based control are analyzed and compared with proposed method in simulation and hardware experiments. The major contributions of this research are as followings:

- The efficient operation of an IPM motor is important and also challenging, particularly when considering nonlinear parameter variations of the motor. A novel machine-learning strategy based on a neural network (NN) is developed for the MTPA, flux-weakening, and MTPV control over the full speed range. The proposed method overcomes the disadvantages of the traditional methods, such as 1) requirement for a large memory size to store the lookup tables and 2) limited control accuracy due to the LUT size and interpolation. The study shows that compared to the traditional methods, the proposed NN-based method requires a very small memory size, is much more accurate to get the optimal operating points than the LUT approaches, is able to capture the nonlinear impact caused by the optimization solutions and variable motor parameters, and requires a lower computational cost. The hardware experiments further demonstrate the effectiveness of the NN method applicable to practical IPM motors.

- The accuracy of IPM magnetic modeling is important for the control of high-performance motor drives. The developed ANN-based modeling method overcomes the inaccuracy issues due to the high-speed operation, saturation, and other uncertainties. Compared with traditional methods, the proposed modeling method has a simple structure and can capture the unknown nonlinear areas of the model accurately without any computation and convergence issues. It also can capture the iron loss effects with speed variations and has a better performance in high-speed range.
- Compared to conventional control methods for DC/DC converters, the ANN-based optimal and predictive controller shows better performance in various aspects. In addition, the ANN controller can handle the control of the *dc/dc* converter properly under both the maximum inductor current and duty ratio constraints, while a conventional controller needs to have an inner-current loop through a cascade control structure to handle the current limit control. Compared to the conventional control methods, the ANN controller responds faster and maintains a more stable output voltage. The hardware experiment confirmed that the ANN controller is able to track reference commands, maintain output voltage stability under variable load and input voltage conditions, and manage the control of the *dc/dc* converter correctly under the maximum duty-ratio and inductor current constraints. The study shows that it is feasible to implement the ANN-based control for practical *dc/dc* Buck converters.
- This research also presents a preliminary study on developing an ANN-based control for DC/DC converters and integrating it with droop mechanism for control of a standalone DC microgrid. The system performance evaluation shows that the bus voltage can be regulated at the desired value and the output power of the converters can be regulated

rapidly when there are reference or load changes. It also shows that the proposed strategy can be implemented and well-performed in a DC microgrid. The proposed method has the ability to maintain voltage stability of standalone DC microgrid and manage the power sharing among the parallel-connected distributed generation units. For different transient scenarios, the ANN controller in DC microgrids also performs well to track voltage references rapidly and tolerate load disturbances.

6.2 Future work

The NN-based controller for MTPA control and magnetic modeling can be extended to a cloud computing platform. For the routine offline training of the NN over the cloud computing platform, it will follow the same strategy except that data needs to be collected and transmitted to a remote cloud device for the NN training. Firstly, real-time motor operation data is collected smartly, meaning that only meaningful and valuable data is collected and stored, and data collected should cover the full operating range of the motor at the normal and critical conditions. When sufficient data is obtained, data is sent to a remote cloud computing device, where data is analyzed and, if needed, a cloud-based training of the NN will be conducted thoroughly. After the training, the new NN weights are transmitted back to the EV computer to replace the previous NN weights. Such an offline learning strategy, based upon a cloud computing platform, will guaranty the safe and reliable NN system development that cannot be achieved for an online NN learning system, which will assure the high-performance operation of individual EV motor over its lifetime.

The proposed ANN control method can be extended to other dc/dc converters, such as Boost and Buck-Boost converters. However, since the state-space models of the Boost and Buck-

Boost converters are different from that of the Buck converter, the training algorithms for each of the other dc/dc converters need to be redesigned and revalidated.

REFERENCES

- [1] “Global electric car sales by key markets, 2010-2020 – Charts – Data & Statistics,” IEA. <https://www.iea.org/data-and-statistics/charts/global-electric-car-sales-by-key-markets-2015-2020> (accessed Mar. 03, 2021).
- [2] “EV-Volumes - The Electric Vehicle World Sales Database.” <https://www.ev-volumes.com/> (accessed Jun. 20, 2021).
- [3] M. Ehsani, Ed., *Modern electric, hybrid electric, and fuel cell vehicles: fundamentals, theory, and design*. Boca Raton: CRC Press, 2005.
- [4] P. C. Krause, O. Wasynczuk, S. D. Sudhoff, and S. Pekarek, *Analysis of electric machinery and drive systems*, Third edition. Hoboken, New Jersey: Wiley, 2013.
- [5] M. N. Uddin and Md. M. Rahman, “Online Torque-Flux Estimation-Based Nonlinear Torque and Flux Control Scheme of IPMSM Drive for Reduced Torque Ripples,” *IEEE Trans. Power Electron.*, vol. 34, no. 1, pp. 636–645, Jan. 2019, doi: 10.1109/TPEL.2018.2827332.
- [6] Y. Sangsefidi, S. Ziaeinejad, A. Mehrizi-Sani, H. Pairodin-Nabi, and A. Shoulaie, “Estimation of Stator Resistance in Direct Torque Control Synchronous Motor Drives,” *IEEE Trans. Energy Convers.*, vol. 30, no. 2, pp. 626–634, Jun. 2015, doi: 10.1109/TEC.2014.2364191.
- [7] “A comparative overview of indirect field oriented control (IFOC) and deadbeat-direct torque and flux control (DB-DTFC) for AC Motor Drives,” *Chin. J. Electr. Eng.*, vol. 1, no. 1, pp. 9–20, Dec. 2015, doi: 10.23919/CJEE.2015.7933134.
- [8] D. Casadei, F. Profumo, G. Serra, and A. Tani, “FOC and DTC: two viable schemes for induction motors torque control,” *IEEE Trans. Power Electron.*, vol. 17, no. 5, pp. 779–787, Sep. 2002, doi: 10.1109/TPEL.2002.802183.
- [9] J. J. Justo, F. Mwasilu, E.-K. Kim, J. Kim, H. H. Choi, and J.-W. Jung, “Fuzzy Model Predictive Direct Torque Control of IPMSMs for Electric Vehicle Applications,” *IEEE/ASME Trans. Mechatron.*, vol. 22, no. 4, pp. 1542–1553, Aug. 2017, doi: 10.1109/TMECH.2017.2665670.
- [10] T. D. Do, H. H. Choi, and J.-W. Jung, “Nonlinear Optimal DTC Design and Stability Analysis for Interior Permanent Magnet Synchronous Motor Drives,” *IEEE/ASME*

Trans. Mechatron., vol. 20, no. 6, pp. 2716–2725, Dec. 2015, doi: 10.1109/TMECH.2015.2426725.

- [11] N. Mohan, “Advanced Electric Drives,” p. 199.
- [12] C.-T. Pan and S.-M. Sue, “A Linear Maximum Torque Per Ampere Control for IPMSM Drives Over Full-Speed Range,” *IEEE Trans. On Energy Conversion*, vol. 20, no. 2, pp. 359–366, Jun. 2005, doi: 10.1109/TEC.2004.841517.
- [13] S. Morimoto, M. Sanada, and Y. Takeda, “Wide-speed operation of interior permanent magnet synchronous motors with high-performance current regulator,” *IEEE Trans. on Ind. Applicat.*, vol. 30, no. 4, pp. 920–926, Aug. 1994, doi: 10.1109/28.297908.
- [14] S. D. Umans and A. E. Fitzgerald, *Fitzgerald & Kingsley’s electric machinery*, 7th ed. New York: McGraw-Hill Companies, 2014.
- [15] Y. Miao, H. Ge, M. Preindl, J. Ye, B. Cheng, and A. Emadi, “MTPA Fitting and Torque Estimation Technique Based on a New Flux-Linkage Model for Interior-Permanent-Magnet Synchronous Machines,” *IEEE Trans. on Ind. Applicat.*, vol. 53, no. 6, pp. 5451–5460, Nov. 2017, doi: 10.1109/TIA.2017.2726980.
- [16] H.-S. Kim, Y. Lee, S.-K. Sul, J. Yu, and J. Oh, “Online MTPA Control of IPMSM Based on Robust Numerical Optimization Technique,” *IEEE Trans. on Ind. Applicat.*, vol. 55, no. 4, pp. 3736–3746, Jul. 2019, doi: 10.1109/TIA.2019.2904567.
- [17] M. Preindl and S. Bolognani, “Optimal State Reference Computation With Constrained MTPA Criterion for PM Motor Drives,” *IEEE Trans. Power Electron.*, vol. 30, no. 8, pp. 4524–4535, Aug. 2015, doi: 10.1109/TPEL.2014.2354299.
- [18] X. Liu, H. Chen, J. Zhao, and A. Belahcen, “Research on the Performances and Parameters of Interior PMSM Used for Electric Vehicles,” *IEEE Trans. Ind. Electron.*, vol. 63, no. 6, pp. 3533–3545, Jun. 2016, doi: 10.1109/TIE.2016.2524415.
- [19] S. Morimoto, M. Sanada, and Y. Takeda, “Wide-speed operation of interior permanent magnet synchronous motors with high-performance current regulator,” *IEEE Trans. on Ind. Applicat.*, vol. 30, no. 4, pp. 920–926, Aug. 1994, doi: 10.1109/28.297908.
- [20] J. Lara, J. Xu, and A. Chandra, “Effects of Rotor Position Error in the Performance of Field Oriented Controlled PMSM Drives for Electric Vehicle Traction Applications,” *IEEE Trans. Ind. Electron.*, pp. 1–1, 2016, doi: 10.1109/TIE.2016.2549983.
- [21] T. Windisch and W. Hofmann, “A Novel Approach to MTPA Tracking Control of AC Drives in Vehicle Propulsion Systems,” *IEEE Trans. Veh. Technol.*, vol. 67, no. 10, pp. 9294–9302, Oct. 2018, doi: 10.1109/TVT.2018.2861083.

- [22] T. M. Jahns, "Flux-Weakening Regime Operation of an Interior Permanent-Magnet Synchronous Motor Drive," *IEEE Trans. on Ind. Applicat.*, vol. IA-23, no. 4, pp. 681–689, Jul. 1987, doi: 10.1109/TIA.1987.4504966.
- [23] T.-S. Kwon, G.-Y. Choi, M.-S. Kwak, and S.-K. Sul, "Novel Flux-Weakening Control of an IPMSM for Quasi-Six-Step Operation," *IEEE Trans. on Ind. Applicat.*, vol. 44, no. 6, pp. 1722–1731, 2008, doi: 10.1109/TIA.2008.2006305.
- [24] L. Sepulchre, M. Fadel, M. Pietrzak-David, and G. Porte, "MTPV Flux-Weakening Strategy for PMSM High Speed Drive," *IEEE Transactions on Industry Applications*, vol. 54, no. 6, pp. 6081–6089, Nov. 2018, doi: 10.1109/TIA.2018.2856841.
- [25] H. Ge, Y. Miao, B. Bilgin, B. Nahid-Mobarakeh, and A. Emadi, "Speed Range Extended Maximum Torque Per Ampere Control for PM Drives Considering Inverter and Motor Nonlinearities," *IEEE Trans. Power Electron.*, vol. 32, no. 9, pp. 7151–7159, Sep. 2017, doi: 10.1109/TPEL.2016.2630051.
- [26] H. W. de Kock, A. J. Rix, and M. J. Kamper, "Optimal Torque Control of Synchronous Machines Based on Finite-Element Analysis," *IEEE Transactions on Industrial Electronics*, vol. 57, no. 1, pp. 413–419, Jan. 2010, doi: 10.1109/TIE.2009.2030209.
- [27] Bing Cheng and T. R. Tesch, "Torque Feedforward Control Technique for Permanent-Magnet Synchronous Motors," *IEEE Trans. Ind. Electron.*, vol. 57, no. 3, pp. 969–974, Mar. 2010, doi: 10.1109/TIE.2009.2038951.
- [28] L. Ortombina, F. Tinazzi, and M. Zigliotto, "Adaptive Maximum Torque per Ampere Control of Synchronous Reluctance Motors by Radial Basis Function Networks," *IEEE J. Emerg. Sel. Topics Power Electron.*, vol. 7, no. 4, pp. 2531–2539, Dec. 2019, doi: 10.1109/JESTPE.2018.2858842.
- [29] H.-S. Kim, Y. Lee, S.-K. Sul, J. Yu, and J. Oh, "Online MTPA Control of IPMSM Based on Robust Numerical Optimization Technique," *IEEE Trans. on Ind. Applicat.*, vol. 55, no. 4, pp. 3736–3746, Jul. 2019, doi: 10.1109/TIA.2019.2904567.
- [30] Q. Liu and K. Hameyer, "High-Performance Adaptive Torque Control for an IPMSM With Real-Time MTPA Operation," *IEEE Trans. Energy Convers.*, vol. 32, no. 2, pp. 571–581, Jun. 2017, doi: 10.1109/TEC.2016.2633302.
- [31] K. D. Hoang and H. K. A. Aorith, "Online Control of IPMSM Drives for Traction Applications Considering Machine Parameter and Inverter Nonlinearities," *IEEE Trans. Transp. Electrification*, vol. 1, no. 4, pp. 312–325, Dec. 2015, doi: 10.1109/TTE.2015.2477469.
- [32] S.-Y. Jung and K. Nam, "Current Minimizing Torque Control of the IPMSM Using Ferrari's Method," *IEEE TRANSACTIONS ON POWER ELECTRONICS*, vol. 28, no. 12, p. 15, 2013.

- [33] Z. Han, J. Liu, W. Yang, D. B. Pinhal, N. Reiland, and D. Gerling, "Improved Online Maximum-Torque-Per-Ampere Algorithm for Speed Controlled Interior Permanent Magnet Synchronous Machine," *IEEE Trans. Ind. Electron.*, vol. 67, no. 5, pp. 3398–3408, May 2020, doi: 10.1109/TIE.2019.2918471.
- [34] F.-J. Lin, Y.-T. Liu, and W.-A. Yu, "Power Perturbation Based MTPA With an Online Tuning Speed Controller for an IPMSM Drive System," *IEEE Trans. Ind. Electron.*, vol. 65, no. 5, pp. 3677–3687, May 2018, doi: 10.1109/TIE.2017.2762634.
- [35] Z. Li, G. Feng, C. Lai, W. Li, and N. C. Kar, "Machine Parameter-Independent Maximum Torque Per Ampere Control for Dual Three-Phase PMSMs," *IEEE Trans. Transp. Electrific.*, vol. 5, no. 4, pp. 1430–1440, Dec. 2019, doi: 10.1109/TTE.2019.2953656.
- [36] L. Ortombina, F. Tinazzi, and M. Zigliotto, "Magnetic Modeling of Synchronous Reluctance and Internal Permanent Magnet Motors Using Radial Basis Function Networks," *IEEE Trans. Ind. Electron.*, vol. 65, no. 2, pp. 1140–1148, Feb. 2018, doi: 10.1109/TIE.2017.2733502.
- [37] S. Li, H. Won, X. Fu, M. Fairbank, D. C. Wunsch, and E. Alonso, "Neural-Network Vector Controller for Permanent-Magnet Synchronous Motor Drives: Simulated and Hardware-Validated Results," *IEEE Trans. Cybern.*, vol. 50, no. 7, pp. 3218–3230, Jul. 2020, doi: 10.1109/TCYB.2019.2897653.
- [38] L. Guo and L. Parsa, "Model Reference Adaptive Control of Five-Phase IPM Motors Based on Neural Network," *IEEE Transactions on Industrial Electronics*, vol. 59, no. 3, pp. 1500–1508, Mar. 2012, doi: 10.1109/TIE.2011.2163371.
- [39] F.-J. Lin, M.-S. Huang, S.-G. Chen, and C.-W. Hsu, "Intelligent Maximum Torque per Ampere Tracking Control of Synchronous Reluctance Motor Using Recurrent Legendre Fuzzy Neural Network," *IEEE Trans. Power Electron.*, vol. 34, no. 12, pp. 12080–12094, Dec. 2019, doi: 10.1109/TPEL.2019.2906664.
- [40] J. Chen, J. Li, and R. Qu, "Maximum-Torque-per-Ampere and Magnetization-State Control of a Variable-Flux Permanent Magnet Machine," *IEEE Trans. Ind. Electron.*, vol. 65, no. 2, pp. 1158–1169, Feb. 2018, doi: 10.1109/TIE.2017.2733494.
- [41] M.-S. Wang, M.-F. Hsieh, Y.-S. Kung, and G. T. Lin, "Maximum torque per ampere control of IPMSM drive by fuzzy logic," *Microsyst Technol*, vol. 24, no. 1, pp. 19–26, Jan. 2018, doi: 10.1007/s00542-016-3119-5.
- [42] M. Nasir Uddin and J. Khastoo, "Fuzzy Logic-Based Efficiency Optimization and High Dynamic Performance of IPMSM Drive System in Both Transient and Steady-State Conditions," *IEEE Trans. on Ind. Applicat.*, vol. 50, no. 6, pp. 4251–4259, Nov. 2014, doi: 10.1109/TIA.2014.2317845.

- [43] M. N. Uddin and R. S. Rebeiro, "Online Efficiency Optimization of a Fuzzy-Logic-Controller-Based IPMSM Drive," *IEEE Trans. on Ind. Applicat.*, vol. 47, no. 2, pp. 1043–1050, Mar. 2011, doi: 10.1109/TIA.2010.2103293.
- [44] L. Griffin, F. Fleming, and C. S. Edrington, "A particle swarm optimization based maximum torque per ampere control for a switched reluctance motor," in *IECON 2014 - 40th Annual Conference of the IEEE Industrial Electronics Society*, Dallas, TX, USA, Oct. 2014, pp. 343–348, doi: 10.1109/IECON.2014.7048522.
- [45] A. M. A. Amin, M. I. El Korfally, A. A. Sayed, and O. T. M. Hegazy, "Efficiency Optimization of Two-Asymmetrical-Winding Induction Motor Based on Swarm Intelligence," *IEEE Trans. Energy Convers.*, vol. 24, no. 1, pp. 12–20, Mar. 2009, doi: 10.1109/TEC.2008.2011831.
- [46] Y. Sun, S. Li, X. Fu, W. Dong, M. Ramezani, and B. Balasubramanian, "Approximate Dynamic Programming Vector Controllers for Operation of IPM Motors in Linear and Over-modulation Regions," *IEEE Transactions on Transportation Electrification*, pp. 1–1, 2020, doi: 10.1109/TTE.2020.3034044.
- [47] M. T. Hagan and M. B. Menhaj, "Training feedforward networks with the Marquardt algorithm," *IEEE Trans. Neural Netw.*, vol. 5, no. 6, pp. 989–993, Nov. 1994, doi: 10.1109/72.329697.
- [48] "JMAG-RT Model Library," *Simulation Technology for Electromechanical Design : JMAG*. <https://www.jmag-international.com/modellibrary/> (accessed Oct. 24, 2020).
- [49] "Simscape." <https://www.mathworks.com/products/simscape.html> (accessed Feb. 22, 2021).
- [50] "FAQ 269: Determining the Size of a Real-Time Application - dSPACE." <https://www.dspace.com/en/inc/home/support/kb/faqs/faq269.cfm> (accessed Feb. 22, 2021).
- [51] "MicroLabBox Hardware." <https://www.dspace.com/en/inc/home/products/hw/microlabbox.cfm> (accessed Feb. 22, 2021).
- [52] E. Armando, R. I. Bojoi, P. Guglielmi, G. Pellegrino, and M. Pastorelli, "Experimental Identification of the Magnetic Model of Synchronous Machines," *IEEE Trans. on Ind. Applicat.*, vol. 49, no. 5, pp. 2116–2125, Sep. 2013, doi: 10.1109/TIA.2013.2258876.
- [53] M. Preindl and S. Bolognani, "Optimal State Reference Computation With Constrained MTPA Criterion for PM Motor Drives," *IEEE Trans. Power Electron.*, vol. 30, no. 8, pp. 4524–4535, Aug. 2015, doi: 10.1109/TPEL.2014.2354299.

- [54] K. D. Hoang and H. K. A. Aorith, "Online Control of IPMSM Drives for Traction Applications Considering Machine Parameter and Inverter Nonlinearities," *IEEE Trans. Transp. Electric.*, vol. 1, no. 4, pp. 312–325, Dec. 2015, doi: 10.1109/TTE.2015.2477469.
- [55] Z. Han, J. Liu, W. Yang, D. B. Pinhal, N. Reiland, and D. Gerling, "Improved Online Maximum-Torque-Per-Ampere Algorithm for Speed Controlled Interior Permanent Magnet Synchronous Machine," *IEEE Trans. Ind. Electron.*, vol. 67, no. 5, pp. 3398–3408, May 2020, doi: 10.1109/TIE.2019.2918471.
- [56] A. Accetta, M. Cirrincione, M. Pucci, and A. Sferlazza, "State-Space Vector Model of Linear Induction Motors Including End-Effects and Iron Losses—Part II: Model Identification and Results," *IEEE Trans. on Ind. Applicat.*, vol. 56, no. 1, pp. 245–255, Jan. 2020, doi: 10.1109/TIA.2019.2952034.
- [57] Y. Wang, Y. Xu, and J. Zou, "Online Multiparameter Identification Method for Sensorless Control of SPMSM," *IEEE Trans. Power Electron.*, vol. 35, no. 10, pp. 10601–10613, Oct. 2020, doi: 10.1109/TPEL.2020.2974870.
- [58] Y. Inoue, Y. Kawaguchi, S. Morimoto, and M. Sanada, "Performance Improvement of Sensorless IPMSM Drives in a Low-Speed Region Using Online Parameter Identification," *IEEE Trans. on Ind. Applicat.*, vol. 47, no. 2, pp. 798–804, Mar. 2011, doi: 10.1109/TIA.2010.2101994.
- [59] R. Ni, D. Xu, G. Wang, L. Ding, G. Zhang, and L. Qu, "Maximum Efficiency Per Ampere Control of Permanent-Magnet Synchronous Machines," *IEEE Trans. Ind. Electron.*, vol. 62, no. 4, pp. 2135–2143, Apr. 2015, doi: 10.1109/TIE.2014.2354238.
- [60] B. Stumberger, G. Stumberger, D. Dolinar, A. Hamler, and M. Trlep, "Evaluation of saturation and cross-magnetization effects in interior permanent-magnet synchronous motor," *IEEE Trans. on Ind. Applicat.*, vol. 39, no. 5, pp. 1264–1271, Sep. 2003, doi: 10.1109/TIA.2003.816538.
- [61] E. Armando, R. I. Bojoi, P. Guglielmi, G. Pellegrino, and M. Pastorelli, "Experimental Identification of the Magnetic Model of Synchronous Machines," *IEEE Trans. on Ind. Applicat.*, vol. 49, no. 5, pp. 2116–2125, Sep. 2013, doi: 10.1109/TIA.2013.2258876.
- [62] K. M. Rahman and S. Hiti, "Identification of machine parameters of a synchronous motor," *IEEE Transactions on Industry Applications*, vol. 41, no. 2, pp. 557–565, Mar. 2005, doi: 10.1109/TIA.2005.844379.
- [63] H.-S. Kim, Y. Lee, S.-K. Sul, J. Yu, and J. Oh, "Online MTPA Control of IPMSM Based on Robust Numerical Optimization Technique," *IEEE Trans. on Ind. Applicat.*, vol. 55, no. 4, pp. 3736–3746, Jul. 2019, doi: 10.1109/TIA.2019.2904567.

- [64] M. Basic, D. Vukadinovic, I. Grgic, and M. Bubalo, "Speed-Sensorless Vector Control of an Induction Generator Including Stray Load and Iron Losses and Online Parameter Tuning," *IEEE Trans. Energy Convers.*, vol. 35, no. 2, pp. 724–732, Jun. 2020, doi: 10.1109/TEC.2019.2952666.
- [65] V. Ruuskanen, J. Nerg, M. Rilla, and J. Pyrhonen, "Iron Loss Analysis of the Permanent-Magnet Synchronous Machine Based on Finite-Element Analysis Over the Electrical Vehicle Drive Cycle," *IEEE Trans. Ind. Electron.*, vol. 63, no. 7, pp. 4129–4136, Jul. 2016, doi: 10.1109/TIE.2016.2549005.
- [66] B.-S. Jun, J. Park, J.-H. Choi, K.-D. Lee, and C.-Y. Won, "Temperature Estimation of Stator Winding in Permanent Magnet Synchronous Motors Using d-Axis Current Injection," *Energies*, vol. 11, no. 8, p. 2033, Aug. 2018, doi: 10.3390/en11082033.
- [67] Y. Wang et al., "Initial Rotor Position and Magnetic Polarity Identification of PM Synchronous Machine Based on Nonlinear Machine Model and Finite Element Analysis," *IEEE Transactions on Magnetics*, vol. 46, no. 6, pp. 2016–2019, Jun. 2010, doi: 10.1109/TMAG.2010.2042690.
- [68] S. Kallio, J. Karttunen, P. Peltoniemi, P. Silventoinen, and O. Pyrhönen, "Online Estimation of Double-Star IPM Machine Parameters Using RLS Algorithm," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 9, pp. 4519–4530, Sep. 2014, doi: 10.1109/TIE.2013.2290761.
- [69] D. Q. Dang, M. S. Rifaq, H. H. Choi, and J.-W. Jung, "Online Parameter Estimation Technique for Adaptive Control Applications of Interior PM Synchronous Motor Drives," *IEEE Trans. Ind. Electron.*, vol. 63, no. 3, pp. 1438–1449, Mar. 2016, doi: 10.1109/TIE.2015.2494534.
- [70] S. Ozden, G. Manav, and M. Dursun, "ANN based magnetic field and inductance modeling of double sided linear switched reluctance motor," in *2018 5th International Conference on Electrical and Electronic Engineering (ICEEE)*, Istanbul, May 2018, pp. 133–137. doi: 10.1109/ICEEE2.2018.8391316.
- [71] L. Ortombina, F. Tinazzi, and M. Zigliotto, "Magnetic Modeling of Synchronous Reluctance and Internal Permanent Magnet Motors Using Radial Basis Function Networks," *IEEE Trans. Ind. Electron.*, vol. 65, no. 2, pp. 1140–1148, Feb. 2018, doi: 10.1109/TIE.2017.2733502.
- [72] L. Ortombina, F. Tinazzi, and M. Zigliotto, "Adaptive Maximum Torque per Ampere Control of Synchronous Reluctance Motors by Radial Basis Function Networks," *IEEE J. Emerg. Sel. Topics Power Electron.*, vol. 7, no. 4, pp. 2531–2539, Dec. 2019, doi: 10.1109/JESTPE.2018.2858842.
- [73] P. J. d. S. Neto, T. A. d. S. Barros, J. P. C. Silveira, E. R. Filho, J. C. Vasquez and J. M. Guerrero, "Power Management Strategy Based on Virtual Inertia for DC Microgrids,"

- in *IEEE Transactions on Power Electronics*, vol. 35, no. 11, pp. 12472-12485, Nov. 2020, doi: 10.1109/TPEL.2020.2986283.
- [74] W. W. A. G. Silva, T. R. Oliveira and P. F. Donoso-Garcia, "An Improved Voltage-Shifting Strategy to Attain Concomitant Accurate Power Sharing and Voltage Restoration in Droop-Controlled DC Microgrids," in *IEEE Transactions on Power Electronics*, vol. 36, no. 2, pp. 2396-2406, Feb. 2021, doi: 10.1109/TPEL.2020.3009619.
- [75] Z. Xia and J. Abu Qahouq, "State-of-charge Balancing of Lithium-ion Batteries with State-of-health Awareness Capability," in *IEEE Transactions on Industry Applications*, doi: 10.1109/TIA.2020.3029755.
- [76] M. S. Sadabadi, "A Distributed Control Strategy for Parallel DC-DC Converters," in *IEEE Control Systems Letters*, vol. 5, no. 4, pp. 1231-1236, Oct. 2021, doi: 10.1109/LCSYS.2020.3025411.
- [77] N. Mohan, *Power Electronics: A First Course*, Wiley, 2011
- [78] R.A. Mammano, *Fundamentals of Power Supply Design*, Texas Instruments, 2017.
- [79] S. Tan, Y. M. Lai and C. K. Tse, "General Design Issues of Sliding-Mode Controllers in DC-DC Converters," *IEEE Trans. Ind. Electron.*, vol. 55, no. 3, pp. 1160-1174, March 2008.
- [80] M. Ordonez, M. T. Iqbal and J. E. Quicoe, "Selection of a curved switching surface for buck converters," *IEEE Trans. on Power Electron.*, vol. 21, no. 4, pp. 1148-1153, July 2006.
- [81] R. Ling, D. Maksimovic and R. Leyva, "Second-Order Sliding-Mode Controlled Synchronous Buck DC-DC Converter," *IEEE Trans. on Power Electron.*, vol. 31, no. 3, pp. 2539-2549, March 2016.
- [82] Cheng, Zhiping, et al. "A novel cascaded control to improve stability and inertia of parallel buck-boost converters in DC microgrid." *International Journal of Electrical Power & Energy Systems* 119 (2020): 105950.
- [83] Z. Wang, S. Li and Q. Li, "Continuous Nonsingular Terminal Sliding Mode Control of DC-DC Boost Converters Subject to Time-Varying Disturbances," in *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 67, no. 11, pp. 2552-2556, Nov. 2020
- [84] B. Wang, V. R. K. Kanamarlapudi, L. Xian, X. Peng, K. T. Tan and P. L. So, "Model Predictive Voltage Control for Single-Inductor Multiple-Output DC-DC Converter With Reduced Cross Regulation," in *IEEE Transactions on Industrial Electronics*, vol. 63, no. 7, pp. 4187-4197, July 2016.

- [85] Z. Liu, L. Xie, A. Bemporad and S. Lu, "Fast Linear Parameter Varying Model Predictive Control of Buck DC-DC Converters Based on FPGA," in *IEEE Access*, vol. 6, pp. 52434-52446, 2018, doi: 10.1109/ACCESS.2018.2869043.
- [86] J. Mahdavi, M.R. Nasiri, A. Agah, and A. Emadi, "Application of neural networks and state-space averaging to DC/DC PWM converters in sliding-mode operation," *IEEE/ASME Trans. Mechatronics*, 10(1), 2005 pp. 60-67.
- [87] A. Rubaai, A.R. Ofoli, L. Burge, and M. Garuba, "Hardware implementation of an adaptive network-based fuzzy controller for DC-DC converters," *IEEE Trans. Ind. Appl.*, 41(6), 2005, pp.1557-1565.
- [88] R.J. Wai and L.C. Shih, "Adaptive fuzzy-neural-network design for voltage tracking control of a DC–DC boost converter," *IEEE Trans. Power Electron.*, 27(4), 2012, pp.2104-2115.
- [89] Dimitri P. Bertsekas, "Dynamic Programming and Optimal Control: Approximate Dynamic Programming," 4th Ed., Athena Scientific, 2012.
- [90] F.L. Lewis and D. Liu (eds.), *Reinforcement Learning and Approximate Dynamic Programming for Feedback Control*, IEEE Press / Wiley, 2012, pp. 474-493.
- [91] H. Zhang, C. Li, X. Zhang, and Y. Luo, "Data-Driven Robust Approximate Optimal Tracking Control for Unknown General Nonlinear Systems Using Adaptive Dynamic Programming Method," *IEEE Trans. Neural Netw.*, 22(12), 2011, pp. 2226-2236.
- [92] A. Heydari, "Optimal Switching of DC–DC Power Converters Using Approximate Dynamic Programming," in *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 3, pp. 586-596, March 2018.
- [93] S. Li, M. Fairbank, C. Johnson, D.C. Wunsch, E. Alonso, and J.L. Proao, "Artificial neural networks for control of a grid-connected rectifier/inverter under disturbance, dynamic and power converter switching conditions," *IEEE Trans. Neural Netw. Learn. Syst.*, 25(4), 2014, pp. 738–750.
- [94] E. d. C. Gomes, L. A. de S. Ribeiro, J. V. M. Caracas, S. Y. C. Catunda and R. D. Lorenz, "State space decoupling control design methodology for switching converters," 2010 IEEE Energy Conversion Congress and Exposition, Atlanta, GA, 2010, pp. 4151-4158, doi: 10.1109/ECCE.2010.5617741.
- [95] K.M. Tsang, and W.L. Chan. "Cascade controller for DC/DC buck convertor." *IEE Proc. Electr. Power Appl.*, 152(4), (2005), pp. 827-831.
- [96] Z. Li, C. Zang, P. Zeng, H. Yu, S. Li and J. Bian, "Control of a Grid-Forming Inverter Based on Sliding-Mode and Mixed H_2/H_∞ Control," *IEEE Trans. Ind. Electron.*, vol. 64, no. 5, pp. 3862-3872, May 2017.

- [97] N. Mohan, T.M. Undeland, and W.P. Robbins, *Power Electronics: Converters, Applications, and Design*, 3rd Ed., John Wiley & Sons Inc., Oct. 2002.
- [98] G.F. Franklin, J.D. Powell, M.L. Workman, *Digital control of dynamic systems*, 3rd edition, Addison-Wesley, 1998.
- [99] M. Hagan and M. Menhaj, "Training feedforward networks with the Marquardt algorithm," *IEEE Trans. Neural Netw.*, vol. 5(6), pp. 989-993, Nov. 1994.
- [100] X. Fu, S. Li, M. Fairbank, D. C. Wunsch, and E. Alonso, "Training recurrent neural networks with the Levenberg-Marquardt algorithm for optimal control of a grid connected converter," *IEEE Trans. Neural Netw. Learn. Syst.*, Oct. 2014.
- [101] *Magazine*, vol. 8, no. 1, pp. 18-28, January-February 2010.
- [102] K. Kurohane, T. Senjyu, A. Yona, N. Urasaki, T. Goya and T. Funabashi, "A Hybrid Smart AC/DC Power System," in *IEEE Transactions on Smart Grid*, vol. 1, no. 2, pp. 199-204, Sept. 2010.
- [103] H. Han, X. Hou, J. Yang, J. Wu, M. Su and J. M. Guerrero, "Review of Power Sharing Control Strategies for Islanding Operation of AC Microgrids," in *IEEE Transactions on Smart Grid*, vol. 7, no. 1, pp. 200-215, Jan. 2016.
- [104] A. Kahrobaeian and Y. A. R. I. Mohamed, "Interactive Distributed Generation Interface for Flexible Micro-Grid Operation in Smart Distribution Systems," *IEEE Trans. Sust. Energy*, vol. 3, no. 2, pp. 295- 305, April 2012.
- [105] A. Mehrizi-Sani and R. Iravani, "Potential-Function Based Control of a Microgrid in Islanded and Grid-Connected Modes," *IEEE Trans. Power Syst.*, vol. 25, no. 4, pp. 1883-1891, Nov. 2010.
- [106] T. Dragicevic, J. C. Vasquez, J. M. Guerrero and D. Skrlec, "Advanced LVDC Electrical Power Architectures and Microgrids: A step toward a new generation of power distribution networks.," *IEEE Electrification Magazine*, vol. 2, no. 1, pp. 54-65, March 2014.
- [107] Justo, J.J., Mwasilu, F., Lee, J. and Jung, J.W., " AC-microgrids versus DC-microgrids with distributed energy resources: A review," *Renewable and Sustainable Energy Reviews*, 24, pp.387-405, 2013
- [108] U. Manandhar, A. Ukil and T. K. K. Jonathan, "Efficiency comparison of DC and AC microgrid," *Smart Grid Technologies - Asia (ISGT ASIA), 2015 IEEE Innovative*, Bangkok, 2015, pp. 1-6.
- [109] H. Lotfi; A. Khodaei, "AC Versus DC Microgrid Planning," in *IEEE Transactions on*

- [110] J. M. Guerrero, J. C. Vasquez, J. Matas, L. G. de Vicuna and M. Castilla, "Hierarchical Control of Droop-Controlled AC and DC Microgrids—A General Approach Toward Standardization," in *IEEE Transactions on Industrial Electronics*, vol. 58, no. 1, pp. 158-172, Jan. 2011.
- [111] H. Kakigano, Y. Miura, T. Ise and R. Uchida, "DC micro-grid for super high quality distribution — System configuration and control of distributed generations and energy storage devices," 2006 37th IEEE Power Electronics Specialists Conference, Jeju, 2006, pp. 1-7.
- [112] J. Schonbergerschonberger, R. Duke and S. D. Round, "DC-Bus Signaling: A Distributed Control Strategy for a Hybrid Renewable Nanogrid," in *IEEE Transactions on Industrial Electronics*, vol. 53, no. 5, pp. 1453-1460, Oct. 2006.
- [113] S. Li, M. Fairbank, C. Johnson, D.C. Wunsch and E. Alonso, "Artificial Neural Networks for Control of a Grid-Connected Rectifier/Inverter under Disturbance, Dynamic and Power Converter Switching Conditions," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 4, pp. 738–750, Apr. 2014.
- [114] N. Mohan, T.M. Undeland, and W.P. Robbins, *Power Electronics: Converters, Applications, and Design*, 3rd Ed., John Wiley & Sons Inc., Oct. 2002.
- [115] N. Mohan, *Advanced Electric Drives – Analysis, Modeling and Control using Simulink*, MN: Minnesota Power Electronics Research & Education, ISBN 0-9715292-0-5, 2001.
- [116] F.Y. Wang, H. Zhang, and D. Liu, "Adaptive dynamic programming: An introduction," *IEEE Comput. Intell. Mag.*, pp. 39–47, 2009.
- [117] X. Fu, S. Li, M. Fairbank, D. C. Wunsch, and E. Alonso, "Training recurrent neural networks with the Levenberg-Marquardt algorithm for optimal control of a grid connected converter," *IEEE Trans. Neural Netw. Learn. Syst.*, Oct. 2014.
- [118] S. Anand, B. G. Fernandes and J. Guerrero, "Distributed Control to Ensure Proportional Load Sharing and Improve Voltage Regulation in Low-Voltage DC Microgrids," in *IEEE Transactions on Power Electronics*, vol. 28, no. 4, pp. 1900-1913, April 2013.