Application of search algorithms and machine learning on aircraft electrical drive systems: design, optimization and control

by Yuan GAO

Thesis submitted to The University of Nottingham

for the degree of Doctor of Philosophy, Aug 2021



Abstract

The More-Electric Aircraft (MEA) has been introduced to lower the aircraft weight, improve the aircraft operation efficiency and reduce the environmental impact of aviation. Within the MEA concept, there is an evolutionary trend towards more-electric configurations by replacing the conventional mechanical, pneumatic, and hydraulic systems. To this end, there is a requirement of increasing number of on-board electrical drive systems. Likewise, within hybrid/all-electric propulsion concepts, electrical drives play a pivotal role in handling the drivetrain electromechanical power conversions. However, the great challenges concerning the wide applications of aircraft electrical drives are in achieving high power density and high efficiency during their design, optimization and control.

The main target of this PhD thesis is to develop the hybrid application of optimization and machine learning (ML) algorithms on the aircraft electrical drive systems. This hybrid approach considers three subsystems within the aircraft electrical drive system: an input DC filter, a two-level converter and an AC load (three-phase motor or linear loads). The novel contribution of this work is in combining search algorithms and ML to investigate the optimization-based design and control topics of an aircraft electrical drive system, which comprises of that three subsystems.

There are three optimization-based design cases and one control case in this thesis. The first part presents the design and optimization of the input passive filter components. Instead of a stand-alone capacitor, an *LC* filter is applied at the front-end DC interface because it can save the filter volume/weight with smaller required passive values. Sizing models of inductor and capacitor are given to serve as the basis in this optimization-based design topic. More importantly, a novel algorithm category is introduced to solve the filter Multi-Objective Optimization Problem (MOOP). Two categories are search

algorithms and surrogate (ML) algorithms, both of which are tried and finally compared in the case study. Moreover, three filter designs were selected from the optimization results for the experimental validation where, an inverter-based constant power load (CPL) was used as a particular example tested with these three different *LC* filters.

Moreover, an ML-based inverter control topic is also studied, also in the subsystem level. The main motivation is to address the heavy computation burden of model predictive control (MPC) strategy. After collecting large number of samples from the original MPC operation, a ML model can be trained to imitate how the control algorithm performs during the real-time application. The theory basis of this ML imitation approach is that the MPC process is completely deterministic: for the same set of inputs (i.e. circuit measurements) and a given cost function, the control outputs will always be the same. Two different imitation controllers are designed for the converter case, both of which are tested and compared by the simulation and experiment results.

The second optimization-based design case is to investigate the ML aided MOOP of motor, the third component/subsystem in the drive system. The same with the first optimization case, there are two objectives: mass and power losses; however, the utilization of ML is totally different. In this case, neural network, a well-known ML technique, is trained as a surrogate model mapping from design variables to the dedicated correction factors for the motor MOOP. The main motivation is that analytical motor models usually prone to loss the prediction accuracy in a large design space, compared to the finite-element analysis (FEA). With the help of the trained NN, the analytical predictions can be extremely close to the FEA outputs in the whole design space. Therefore, the analytical base models and the well-trained NN(s) can then become an excellent surrogate for the motor global optimization, which would be extremely fast and accurate. Finally, this NN-aided approach is compared with the conventional FEA-based optimization (using search algorithms).

Finally, the third optimization-based design case is to study the optimization problem of an entire motor drive system which comprises a two-level converter and a three-phase motor. Therefore, this study is for the system-level optimization. Though there is only one objective in this optimization problem, the NN application as a correction surrogate model is similar with the second case. In order to get the target data for the NN surrogate training, motor FEA model and circuit model are jointly simulated in a loop (on separated platforms but both called by the codes on MATLAB). After the data collection of small number of designs, the desired NN can be trained extremely fast whose performance is validated by other samples (rather than the training data). At last, based on the large number of samples in the design space, the analytical base models and the well-trained NN can generate the global optimization of the motor drive system, also in the short time. For the training of each NN, only if the training performance is good and/or the predictions are close to a new data set, the NN can be referred as a well-trained NN and then used in following optimization.

In this thesis, the key role is not focusing on the algorithms themselves, but on their applications. All the used search and machine learning algorithms have been developed in the previous studies. However, this thesis first proposes a simple and clear categorization for the commonly used algorithms to discuss their differences and connections. Based on that, this thesis then develops how to use these algorithms to address optimization and control problems for an MEA electrical drive system. Therefore, the main contribution of this thesis should be the application of different types of algorithms on the aircraft drive systems.

Acknowledgement

Have got the chance to work towards this degree at the Power Electronics, Machines and Control (PEMC) Group, the University of Nottingham, is a great gift for me. During this journey, countless people have offered me a help, inspired and challenged me to become a better scholar. I do want to take this opportunity to thank all of them, also to mention a few in particular.

First, my heartfelt gratitude goes to Dr. Tao Yang, for this opportunity to do my Ph.D. at the PEMC Group, and for his great trust over the years, going all the way back since I was a postgraduate at the Beihang University.

I also wish to thank Professor Serhiy Bozhko and Professor Pat Wheeler for their kind supervision over the years, and for offering help whenever I needed it.

I would also like to thank Dr. Tom Cox and Dr. Sergio Vazquez for being my Ph.D. examiners, for their interest in this research and for their time in reviewing this thesis.

I would like to also express my appreciation to Professor Tomislav Dragičević, from Technical University of Denmark, for the opportunity to do a three-months research exchange at his world-leading research group in Aalborg University.

Many thanks to all my PEMC colleagues that I've met along this PhD journey, for their valuable help and support. Especially, I want to thank Dr. Benjamin Cheong, Dr. Cheng Wang, Dr. Zhen Huang, Dr. Chen Li, Dr. Linhui Fan, Dr. Xiaoyu Lang, and Dr. Feng Guo for their friendship and the good times we've shared.

I would like to thank Dr. Seang Shen Yeoh for a nine-month collaboration in a very interesting and meaningful project in Aerospace Technology Centre.

There are many scholars outside United Kingdom that were instrumental and I would like to acknowledge them. I would like to thank my teammate in Aalborg

University, Dr. Changming Zheng, who helped my first experimental testing for filter design. I sincerely thanks to another teammate in Aalborg University, Dr. Songda Wang, for the collaboration in the studies of modular multi-level converter. I also want to thank Dr. Zhan Li in Nanyang Technological University, for the collaboration in open-switch fault diagnosis study.

Being away from home, I am also grateful to my friends at Cornerstone Church, Nottingham, for encouraging me in my life and study, and for all the good memories that I will dearly cherish for the rest of my life.

I sincerely want also to thank my dear family (Dad, Mum, and Jun) for their continuous love, and for everything that I am and have.

Last but not least, a very special thanks to my girlfriend Xin for her love, understanding, and kind support. Thank you so much, dear, for always being there, for cheering me up, and for believing in me.

Nottingham, 26/07/2021

Yuan

Contents

Abstract III

Acknowledgement VI

Contents VIII

List of Publications XII

List of Abbreviations and Acronyms XV

Chapter 1 Introduction 1

1.1 Background and Motivation 1

1.1.1 More-electric aircraft 2

1.1.2 Hybrid-electric and all-electric propulsion 6

1.1.3 Role of electrical drives in aerospace 8

1.2 System Description 9

1.3 Summary of Research Contributions 13

1.4 Thesis Outline 14

Chapter 2 State of the Art: Design, Optimization and Control of Electrical Motor Drive Systems Using Two Types of Algorithms 17

2.1 Design and Optimization of Motor Drive Systems 17

2.1.1 Subsystem design and optimization in a drive system 18

2.1.2 Drive system design and optimization 23

2.2 Drive System Control 27

2.3 Research Algorithms 32

2.3.1 Optimization algorithms 32

2.3.2 Machine learning algorithms 35

2.4 “2SA” Categorization 38

2.4.1 Search algorithms 40

2.4.2 Surrogate algorithms 44

2.4.3 Comparisons of two algorithm categories 48
2.5 Chapter Summary 49
Chapter 3 Optimal Filter Design for Aircraft Electrical Drive Systems Using Search and Surrogate Algorithms 50
3.1 Filter design method choosing 50
3.2 Filter sizing models 54
3.2.1 Inductor Sizing 54
3.2.2 Capacitor Sizing Model 58
3.2.3 Integrated Sizing for Filter Optimization: 60
3.3 Control algorithm in the actuation system 62
3.4 Search algorithm and optimization results 67
3.4.1 Search algorithm – Genetic Algorithm (GA) 67
3.4.2 Optimization results 73
3.5 Surrogate algorithm and optimization results 75
3.5.1 Surrogate algorithm – NN & SVM 76
3.5.2 Optimization results 79
3.6 Experimental validation of optimal filter design 81
3.6.1 Control strategy and delay compensation: 82
3.6.2 Simulation GA optimization results: 84
3.6.3 Experiment test: 87
3.7 Chapter Summary 90
Chapter 4 Machine Learning Based Model Predictive Controller for Power Converters 91
4.1 Previous studies on machine learning based controllers 91
4.2 MMC system principles 93
4.3 FCS-MPC strategy for MMC 95
4.4 Two NN based imitation controllers 97
4.4.1 NN Regression 98
4.4.2 NN Pattern Recognition 99
4.4.3 Data Collection and Training Steps 100

4.4.4 Training performance of two NN imitation controllers 102
4.5 Simulation and experimental validation 106
4.5.1 Steady-state performance 106
4.5.2 Experiment performance 107
4.5.3 Simulation of dynamic performance 109
4.5.4 Comparison summary 111
4.6 Chapter summary 112
Chapter 5 Machine Learning Based Surrogate Correction Model for PMSM Multi-Objective Optimization 113
5.1 Analytical Models for PMSM Mass and Power Loss 113
5.1.1 Mass estimation 114
5.1.2 Power loss estimation 118
5.2 ML-Based Correction Model 121
5.3 ML Aided Multi-Objective Optimization Procedure 125
5.4 PMSM Case for Method Validation 128
5.5 Comparison with FEA-based Optimization 133
5.6 Chapter summary 135
Chapter 6 Optimal Design of Motor Drive Using Machine Learning Algorithms 137
6.1 Motor drive system description and its joint simulation 137
6.1.1 Motor-drive system description 137
6.1.2 Modulation for two-level converter 139
6.1.3 Joint simulation scheme 142
6.2 General research thinking 145
6.3 Math and simulation models of motor drive system 148
6.4 NN-based correction approach 150
6.5 Motor drive case study 152
6.5.1 NN training and validation for power loss correction 153
6.5.2 NN based optimization of drive system 158
6.6 Chapter Summary 160

Chapter 7 Conclusion and Future Study 161

References 166

List of Publications

The findings of this thesis have been partly published in the publications listed as follows:

1. **Y. Gao**, T. Yang, T. Dragičević, S. Bozhko, P. Wheeler and C. Zheng, "Optimal Filter Design for Power Converters Regulated by FCS-MPC in the MEA," in *IEEE Transactions on Power Electronics*, vol. 36, no. 3, pp. 3258-3268, March 2021, doi: 10.1109/TPEL.2020.3017862.
2. **Y. Gao**, T. Yang, S. Bozhko, P. Wheeler and T. Dragičević, "Filter Design and Optimization of Electromechanical Actuation Systems Using Search and Surrogate Algorithms for More-Electric Aircraft Applications," in *IEEE Transactions on Transportation Electrification*, vol. 6, no. 4, pp. 1434-1447, Dec. 2020, doi: 10.1109/TTE.2020.3019729.
3. **Y. Gao**, T. Yang, S. Bozhko, P. Wheeler, T. Dragičević and Chris Gerada, " Neural Network Aided PMSM Multi-Objective Design and Optimization for More-Electric Aircraft Applications," in *Chinese Journal of Aeronautics*, 2021, https://doi.org/10.1016/j.cja.2021.08.006.
4. S. Wang, T. Dragicevic, **Y. Gao** and R. Teodorescu, "Neural Network based Model Predictive Controllers for Modular Multilevel Converters," in *IEEE Transactions on Energy Conversion*, vol. 36, no. 2, Jun. 2021, doi: 10.1109/TEC.2020.3021022.
5. Z. Li, **Y. Gao**, X. Zhang, B. Wang and H. Ma, "A Model-Data-Hybrid-Driven Diagnosis Method for Open-Switch Faults in Power Converters," in *IEEE Transactions on Power Electronics*, vol. 36, no. 5, pp. 4965-4970, May 2021, doi: 10.1109/TPEL.2020.3026176.

6. S. Wang, T. Dragicevic, **Y. Gao**, S. K. Chaudhary and R. Teodorescu, "Machine Learning Based Operating Region Extension of Modular Multilevel Converters Under Unbalanced Grid Faults," in *IEEE Transactions on Industrial Electronics*, vol. 68, no. 5, pp. 4554-4560, May 2021, doi: 10.1109/TIE.2020.2982109.
7. **Y. Gao**, T. Yang, X. Wang, S. Bozhko and P. Wheeler, "Machine Learning Based Correction Model in PMSM Power Loss Estimation for More-Electric Aircraft Applications," *2020 23rd International Conference on Electrical Machines and Systems (ICEMS)*, 2020, pp. 1940-1944, doi: 10.23919/ICEMS50442.2020.9290844.
8. Z. Xu, **Y. Gao**, X. Wang, X. Tao and Q. Xu, "Surrogate Thermal Model for Power Electronic Modules using Artificial Neural Network," *IECON 2019 - 45th Annual Conference of the IEEE Industrial Electronics Society*, 2019, pp. 3160-3165, doi: 10.1109/IECON.2019.8927494.
9. X. Wang, **Y. Gao**, J. Atkin and S. Bozhko, "Neural Network based Weighting Factor Selection of MPC for Optimal Battery and Load Management in MEA," *2020 23rd International Conference on Electrical Machines and Systems (ICEMS)*, 2020, pp. 1763-1768, doi: 10.23919/ICEMS50442.2020.9290968.
10. Z. Li, **Y. Gao**, H. Ma and X. Zhang, "A Simple ANN-Based Diagnosis Method for Open-Switch Faults in Power Converters," *IECON 2020 The 46th Annual Conference of the IEEE Industrial Electronics Society*, 2020, pp. 2835-2839, doi: 10.1109/IECON43393.2020.9254607.
11. H. Hussaini, T. Yang, **Y. Gao**, C. Wang, T. Dragicevic, S. Bozhko. Droop Coefficient Design in Droop Control of Power Converters for

Improved Load Sharing: An Artificial Neural Network Approach. *2021 the 30th International Symposium on Industrial Electronics*, 20-23 Jun 2021, Kyoto, Japan, doi: 10.1109/ISIE45552.2021.9576482.

12. H. Hussaini, T. Yang, Y. Gao, C. Wang, M. A. A. Mohamed and S. Bozhko, "Artificial Neural Network Aided Cable Resistance Estimation in Droop-Controlled Islanded DC Microgrids," *IECON 2021 – 47th Annual Conference of the IEEE Industrial Electronics Society*, 2021, pp. 1-7, doi: 10.1109/IECON48115.2021.9589411.
13. G. Liu, Y. Chen, **Y. Gao**, J. Zhu, B. Wang, T. Yang. Component Based and Machine Learning Aided Optimal Filter Design for Full-Bridge Current Doubler Rectifier. *IECON 2021 - The 47th Annual Conference of the IEEE Industrial Electronics Society*. 13-16 October 2021, Toronto, Canada, doi: 10.1109/IECON48115.2021.9589272.

List of Abbreviations and Acronyms

2SA	Search Algorithm & Surrogate Algorithm
AC	Alternating Current
ACF	Average Correction Factor
AFE	Active Front-End
APU	Auxiliary Power Unit
CF	Cost Function
CPL	Constant Power Load
CPL	Constant Power Loads
DC	Direct Current
DV	Design Variable
ECS	Environmental Control System
EHA	Electro-Hydraulic actuator
EMA	Electro-Mechanical actuator
EMF	Electro-Motive Force
EMI	Electro-Magnetic Interference
EPS	Electrical Power System
ESR	Equivalent Series Resistance
FCS-MPC	Finite Control Set Model Predictive Control
FEA	Finite-Element Analysis
GA	Genetic Algorithm
GDE3	Generalized Differential Evolution 3
IDG	Integrated Drive Generator
LC	Inductor and Capacitor
MEA	More Electric Aircraft
ML	Machine Learning
MLT	Mean Length per Turn

MMC	Modular Multi-level Converter
MOOP	Multi-Objective Optimization Problem
MPC	Model Predictive Control
NN	Neural Network
NSGA	Non-dominated Sorting Genetic Algorithm
PI	Proportional–Integral
POL	Point-Of-Load
PL	Power Loss
PM	Permanent Magnet
PMSM	Permanent Magnet Synchronous Motor
POL	Point-Of-Load
PR	Pattern Recognition
PWM	Pulse-Width Modulation
RL	Reinforcement Learning
RMS	Root Mean Square
RMS	Root Mean Square
RMSE	Root Mean Square Error
SVM	Support Vector Machine
THD	Total Harmonic Distortion
VSC	Voltage Source Converter

e.g.	Example given
i.e.	id est (latin, ‘that is’)

Chapter 1 Introduction

1.1 Background and Motivation

In June 2019, the United Kingdom (UK) parliament passed the legislation of the "Net Zero" target, which requires the government to reduce the UK's net emissions of greenhouse gases by 100% relative to 1990 levels by 2050 [1]. It makes UK become the first major economy in the world to pass the law of net-zero emissions. Prior to this, the UK was committed to reducing net greenhouse gas emissions by at least 80% of their 1990 levels, also by 2050. It is noted that, apart from the increase in greenhouse gas emissions, the world is also confronting other aspects of the global climate change, including wind, sea level, rain, snow, and cloud cover [2]. These environmental impacts are always coupled, which leads a worse situation.

The contribution of aircraft technology (by having the hydrogen and electrified airplanes) to meeting the net-zero target in 2050 can reach to 17% [3]. To cope with the climate change issues, one of the most important measures currently being undertaken in aviation is creating electrically propelled aircraft. However, compared to a conventional liquid fuel propulsion system, the major challenge for electric aircraft is the low energy density of batteries; the power-to-weight ratio is forecast to improve by a factor of 2 to 4 in the next two decades [4]. It is predicted that appropriate batteries for large aircrafts will not be available until at least the late 2030s and, full-integration with an aircraft and potential in-service dates would likely stretch further into the 2040s [5]. Before progressing to a fully electric system with battery, energy storage innovations in aircraft electrification have been applied to the traditional aircraft. Aircraft electrification allows new and exciting opportunities for the aircraft

optimisation, to save its cost and reduce environmental impact. At a high level, aircraft electrification can be categorised into two trends [6]:

- An evolutionary trend towards more-electric configurations, replacing conventional mechanical, pneumatic, and hydraulic systems.
- A disruptive trend towards hybrid-electric and all-electric propulsion.

On traditional aircraft, most of the power is used for propulsion thrust but some power is used for the electrical, mechanical, pneumatic, and hydraulic on-board loads. The first category of aircraft electrification is replacing the non-electrical systems with electrical components and equipment, which is commonly referred to as the More Electric Aircraft (MEA) concept [7-11]. The MEA concept considers the electrification of all on-board systems except for propulsion [6, 12]. For the second trend in aircraft electrification, on the one hand, hybrid-electric propulsion concepts combine the gas turbine with electrical power generation or storage system to drive an aircraft propeller while, all-electric propulsion concept only replies on the electrical power storage system onboard for the propulsion.

1.1.1 More-electric aircraft

The MEA concept offers many potential benefits in the design of more efficient and environment friendly aircraft. Research about the MEA considers the advanced design and manufacture of energy efficient aircraft with versatile electrical onboard systems that were traditionally powered by hydraulic, mechanical or pneumatic power sources [10]. While an increasing amount of MEA-related research is still under-going, aircraft programmes, such as the Airbus A380 and Boeing 787, have already exhibit greater use of electrical power [7, 8, 10-12].

Figure 1.1 shows a summary of the on-going electrification applications for a MEA and the potential requirements (key indices and final targets) as well as the main motivations for developing MEA technology. Generally, there are three top-level targets:

lower cost, improved safety and greener aviation. To achieve these targets, several commonly used indices can be quantified for MEA system/subsystem studies and applications (e.g. motor drive systems), including efficiency, weight, maintenance cost, reliability and emissions. Obviously, weight, maintenance costs and emissions are the minimum objectives while the others are the maximum objectives. Figure 1.1 lists five electrification examples applied to onboard aircraft systems.

The evolutionary changes in the most recent large commercial MEA (the Boeing 787 and Airbus A380) have eliminated the Integrated Drive Generator (IDG), which paves the way for a constant voltage and variable frequency power bus [8]. This design choice makes power electronics critical in the aircraft industry because, unlike the traditional loads/power systems which run at a constant frequency, the onboard power has to be converted for newer airborne loads/requirements. Without the use of power electronics to convert and control electrical energy, none of the MEA benefits would be possible [9]. Therefore, power electronics-based drive system is the main system studied in this thesis.

Another advantage of aircraft electrification is the elimination of the use of bleed air for the Environmental Control System (ECS). The ECS is used to provide air supply and maintain the temperature and pressure for the crew and passengers in the aircraft. Avionics cooling, smoke detection, and fire suppression are also commonly considered part of an aircraft's ECS. Instead of tapping bleed air from the engines, the ECS is powered by the generators on a MEA. Thus, the traditional pneumatic system and air ducts have been eliminated from the engine for the ECS [8]. All subsystems in the ECS (compressors, fans, pumps and heat exchangers) are fed by the generators via buses and electronic devices, which leads to a significant improvement in the turbine efficiency. In addition, the removal of the mechanical and hydraulic systems may also lead to some reductions in the overall weight at a system level once the full potential of MEA technology has been realized [10].

The generators are usually driven by the main aircraft engines, using the efficiency and energy supply of the main turbines [10]. However, the generators of the Auxiliary Power Unit (APU) are also a critical power supply on a MEA, especially when the power from the main generator is not available. It is possible to remove the pneumatic and hydraulic function of the APU with the applications of electric engine start and de-icing [8, 13]. Installed in many commercial airplanes (including the Boeing 787), this type of more electric APU has been showing to have high efficiency, low weight, long lifetime and high reliability [13]. Another electrification example of hydraulic systems is actuation. When replacing hydraulic actuators with electrically powered actuators, the choices are electro-hydraulic actuators (EHAs) and electromechanical actuators (EMAs) [8-10]. Both require a converter and a motor. EHAs have the advantages of weight, volume, and dispatch reliability, as they eliminate the external hydraulic source and piping systems. However, EMAs should be a better option for leak-free operation and reliability [8, 14]. In this thesis, Chapter 3 and Chapter 6 will consider the multi-objective design and optimization problems for an EMA.

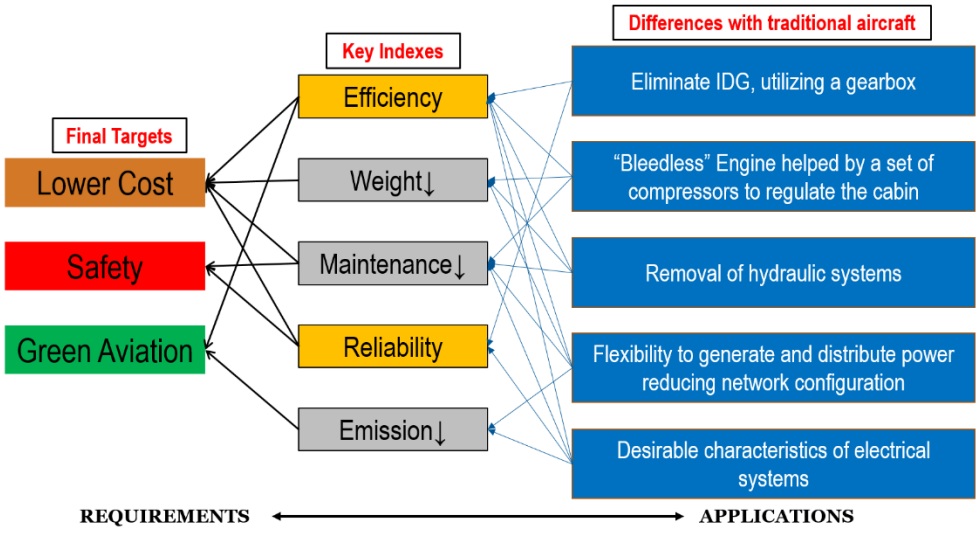
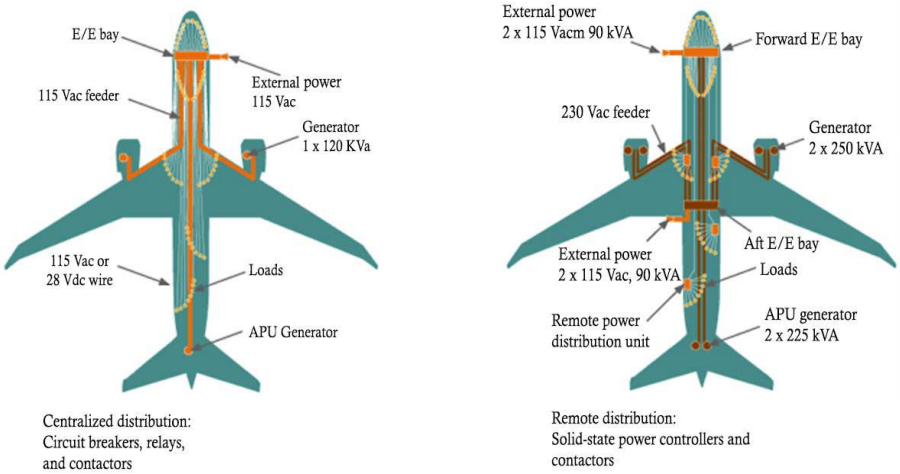


Figure 1.1 Requirements and applications of the More Electric Aircraft

Pneumatic power is traditionally used to start the main aircraft engines, while the more-electric aircraft uses electrical power to achieve the engine start. In the Airbus

A350 and A380, a pneumatic air start turbine system is still used; but more recently on the Boeing 787, the pneumatic system has been eliminated, including the air ducts and air engine starter [8, 15]. In lieu of this, the main engine generators are operated as starter motors.

From a general architecture point of view, an additional advantage that the move to the MEA brings is the flexibility of electrical system to generate and distribute power efficiently to where it is being consumed [7, 8]. Figure 1.2(a) shows the traditional centralized power system distribution architecture. In this architecture, all the electrical power is generated on the wings near the main engines and in the aft near the APU. In the more-electric paradigm of Boeing 787, as shown in Figure 1.2(b), the power distribution units are remotely located. The advanced solid-state power controllers and contactors allow for the possibility to update the traditional centralized configuration. Significant weight and volume savings may be realized, as the power rating of the main conductors may be reduced; furthermore, these savings can contribute to a higher fuel efficiency of the aircraft and a lower total cost of aircraft equipment [8].



(a) Traditional aircraft (d) Boeing 787

Figure 1.2 The 787's electrical system uses a remote distribution system that saves weight and is expected to reduce maintenance costs [15].

1.1.2 Hybrid-electric and all-electric propulsion

Apart from the MEA concept, the other category of aircraft electrification is the hybrid/all-electric propulsion concept. Three architectures for the hybrid/all-electric propulsion concept are shown in Figure 1.3: More Electric-Hybrid, Full Hybrid and All Electric. While other proposals for hybrid systems exist, these three architectures are considered the most feasible at present [4]. Currently, batteries for aircraft propulsion (All-Electric) cannot meet the requirements of energy density and economy for longer flights. Programmes of all-electric aircraft (e.g. E-Fan X [15], Magnus eFusion [16]) are faced with the question of power to weight ratios. By contrast, a hybrid solution is much more likely to be available sooner and could deliver real benefits through reduced emissions.

Based on the existed advances of MEA applications, as the first stage, the 'More Electric-Hybrid' solution can be implemented in a short period. Except for the electric motors to provide thrust, this hybrid system could extend the use of batteries to power all the remaining systems on the aircraft. The gas turbine and the electrical generator are existing and understood solutions; therefore, in this 'More Electric-Hybrid' solution, a new power electronics system is likely to be required to transfer the electrical energy from the generator to the electric motors used for propulsion [4].

For the next stage, the Full Hybrid system relies on the thermal engines (gas turbines or internal combustion engines) for propulsion power and for charging the storage systems. In addition, the storage systems can provide power needed for propulsion and other aircraft systems during some phases of flight. The Full-Hybrid configuration, shown in Figure 1.3, is the series-hybrid concept [4]. In this solution, fuel in the engine is transformed into electrical power via a generator, and it is fed into a grid with energy storage systems. Therefore, series-hybrid systems are compatible with distributed propulsion concepts that use multiple small motors and fans [17, 18]. The main advantage of this solution is that the engine is always operating at an optimised,

single speed during the flight. This allows higher efficiency for the propulsion system [4]. Another full-hybrid configuration is the parallel hybrid-electric solution where both the electrical machine and the engine are mechanically connected to the fan or propeller on the same shaft. An advantage of parallel hybrid configuration is that only one electrical machine is needed as it operates as both a motor and a generator [11, 17].

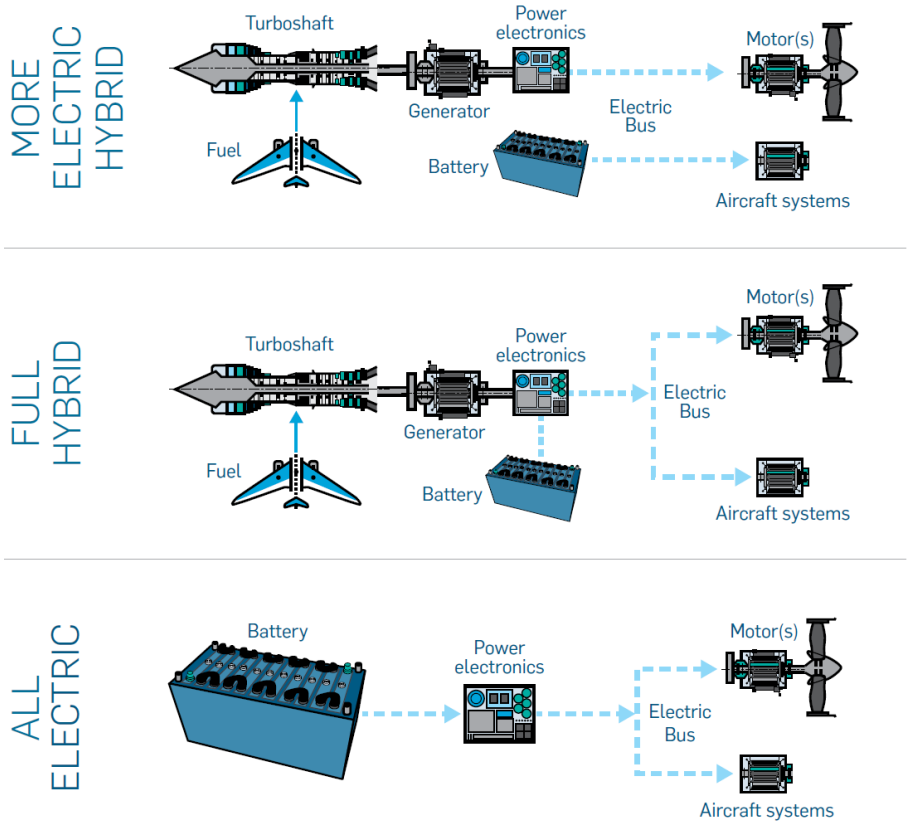


Figure 1.3 Three potential system architectures, from the ‘More Electric – Hybrid’ to ‘Full Hybrid’ to ‘All Electric’ [5].

Finally, in the “All Electric” configuration, the thermal engine system is completely removed. Energy storage elements provide all the power needed onboard for the aircraft propulsion and the various loads. The initial conversion electronics may then need to be adjusted from the Full Hybrid solutions, but the motors and other airborne systems will be unchanged.

1.1.3 Role of electrical drives in aerospace

For these promising MEA and electric propulsion concepts, the electrical drive system, which contains power electronics and aircraft loads (motors and other airborne devices), is always the key part of the aircraft electrical microgrid system for aircraft propulsion, ECSs, braking and actuation systems.

Some practical examples of MEA, hybrid-electric and all-electric motor drives are shown in Figure 1.4. Figure 1.4(a) shows a flap ball-screw actuator for the fly-by-wire flight control system in the Russian SuperJet 100 [16], which is a regional jet for less than 100 passengers. Figure 1.4(b) shows a photo of an electric braking system on-board a Boeing 787 aircraft, which reduced CO_2 emissions and saved fuel consumption by replacing conventional hydraulic pistons with electromechanical actuators [17]. Figure 1.4(c) shows a series hybrid-electric powertrain, developed for a Pipistrel plane under the project HYPSTAIR [18]. In 2016, this was the world's most powerful hybrid electric powertrain for aviation. The complete serial-hybrid-drive system of the HYPSTAIR consists of a 200 kW electric motor, a high-power battery system, four inverters, a 100 kW generator, and an internal combustion engine (ICE) as a range extender (Rotax 914 turbo). Figure 1.4(d) shows the all-electric powertrain of the Extra 330LE two-seater acrobatic plane built by Siemens, which set a new world record for time to climb on 25 November 2016 [19].

In summary, for MEA and electric propulsion concepts, research on power electronics and motors is essential for the development of future onboard electric propulsion system, compressors, pumps, braking and actuation systems. In this thesis, different algorithms are utilized to address the design, optimization and control of the power electronics-based drive systems onboard aircraft.



(a) Ball-screw actuator of Russian SuperJet 100



(b) Boeing 787 electrical braking system



(c) A hybrid-electric Pipistrel plane



(d) All-electric power train in Extra 330LE

Figure 1.4 Motor drive real applications on more-electric aircraft, hybrid-electric and all-electric aircraft

1.2 System Description

Most of electric power today uses the alternating-current (AC) form for generation and distribution [20]. AC is relatively easy to change the voltage of power using a transformer thus the distribution can easily use a low current to achieve small transmission losses. However, a common characteristic of practical power converters is that electricity supply is often in a direct-current (DC) form within the circuit [21]. One benefit of DC systems is that they are significantly simpler to control than AC as there are no synchronization issues, reactive power flows or phase unbalances [22]. For applications in the future MEA or all-electric aircraft, replacing the AC distribution with

DC distribution is a good solution, which introduces many opportunities to optimize the design of Electrical Power System (EPS) since DC microgrids can reduce the total weight of EPS, increase efficiency, and remove reactive power compensation devices [10, 23, 24]. However, it should be noted that, protection is one of the main key issues associated with the application of DC grids, especially for the fault detection, fault location, and the fault interruption devices/methods [25].

An EPS with DC distribution for the MEA has attracted much attention from researchers in this field [7, 10, 23, 26-30]. In [27] a "single-bus" concept was proposed for an on-board microgrid where all energy sources and loads are connected to a single distribution bus (Figure 1.5). The twin generators (G_1 and G_2) take power from one aircraft engine via the high pressure (HP) and low pressure (LP) shafts, and supply power to the main DC bus through separate power electronic converters (active rectifiers) ($AR_{1,2}$). The electrical drive loads (e.g. EMAs as well as other linear and non-linear loads) are connected to the main DC bus through point-of-load (POL) converters for power conditioning and voltage regulation.

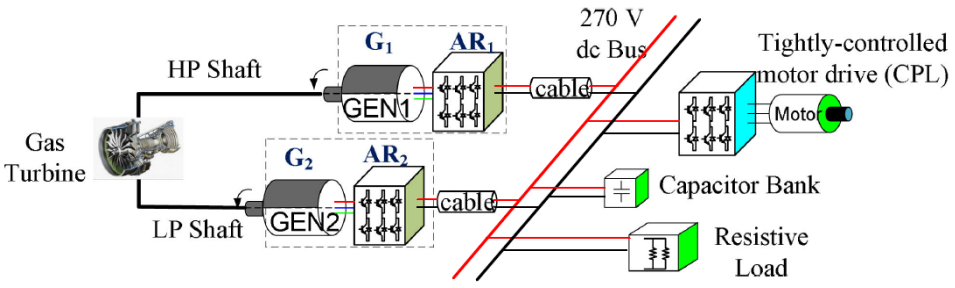


Figure 1.5 Potential DC EPS topology for the MEA [27]

In this thesis, the studied aircraft electrical drive system is a voltage source converter (VSC) (i.e. DC-AC). In order to smooth the DC bus ripple and to meet the electromagnetic interference (EMI) requirement, its input conventionally uses a large DC bus capacitor. However, this DC bus capacitor is usually electrolytic and usually requires a capacitor with big volume/weight, which is not ideal for the aircraft applications. Moreover, the reliability of the inverter is usually limited by the lifetime

of the electrolytic capacitor [31]. In order to improve the power density and prolong the lifetime of the inverter, a large capacitance electrolytic DC bus capacitor can be replaced with a *LC* filter consisting of an inductor and a small film capacitor [28, 29, 31].

The schematic of the electrical drive system considered for the MEA is shown in Figure 1.6. L_1 is the filter inductance, R_1 is the resistance of filter inductor coil, C_1 is the filter capacitance, R_2 is the internal resistance of the capacitor. Apart from the *LC* filter components, the other two subsystems in the studied drive system include a two-level converter (DC-AC) and a three-phase AC load. Different from the airborne motor-drive optimization studies in [12], this thesis will investigate a permanent-magnet synchronous motor (PMSM) for the AC load (Chapters 3 & 5 & 6) but will also study the application of linear loads in the Chapters 3 & 4. In addition, at the end of Chapters 3 & 4, the experiment rigs used for the validation are also described. The finite control set model predictive control (FCS-MPC) strategy is used for both load topologies. However, the AC motor current is controlled in the PMSM topology while the AC filter voltage is controlled in the linear load topology, as described in Chapter 3.

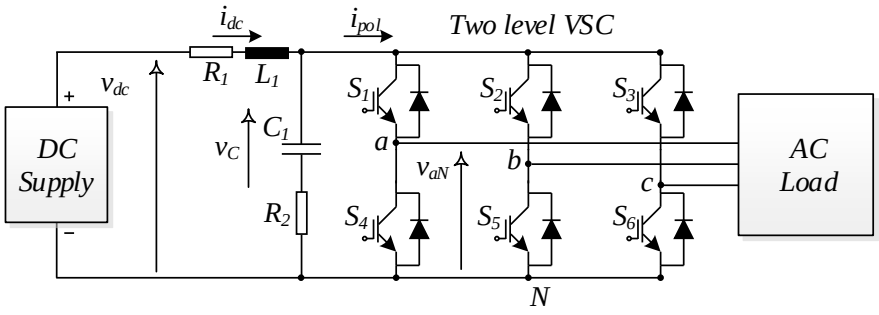


Figure 1.6 The MEA electrical drive system

One of the basic characteristics of FCS-MPC is that it does not involve the pulse-width modulation (PWM). FCS-MPC directly governs the semiconductor switching process of the power converter. However, the variable switching frequency is a specific disadvantage of FCS-MPC, which significantly complicates the filter design compared to standard design procedures for filters of converters operated with linear control (e.g.

PI control) [29]. In Chapter 3, this problem is addressed using a component-simulation-hybrid approach for the filter design and multi-objective optimization problem (MOOP).

It is noted that in the filter MOOP, though the weighting factor (WF) in the FCS-MPC is one of design variables, converter and load side are assumed unchanged (i.e. no other physical or electrical variables in these components are considered in the filter optimization). In contrast, the filter system is fixed for the motor optimization work in Chapter 5 and the "motor + converter" optimization work in Chapter 6. In these two chapters, only machine learning (ML) algorithms are utilized for the optimization works, where some sample data are collected to train the fast and accurate ML models. Therefore, no any search algorithm is used for the motor/motor-drive optimization in these two chapters; instead, exhausted algorithm is used for the global optimization because the trained ML models can yield the desired objective performance very fast.

Apart from the design and optimization works for an electrical drive system onboard MEA, in Chapter 4, ML algorithm will be used for the converter control study. The key point is using a neural network (NN) to imitate the FCS-MPC for the converter control. The related literatures on NN applications for a two-level VSC will be discussed. Furthermore, the NN imitation application for the control of a modular multi-level converter (MMC) will be introduced with technical details and simulation/experiment testing results.

It is noted that the general research sequence of this thesis is from components to the entire drive system; more specifically, the sequence is "filter, converter, motor, and motor-drive". Therefore, the design and optimization of motor-drive system is located at the final technical chapter (Chapter 6) while the ML-aided converter control study (Chapter 4) is put between the optimal filter design (Chapter 3) and the ML-aided motor optimization (Chapter 5). After these technical chapters, this thesis is concluded in Chapter 7.

1.3 Summary of Research Contributions

Novel contributions from this work are summarised in the following points:

- A simple algorithm categorization strategy. In order to quickly classify the basic function of commonly used optimization and machine learning (ML) algorithms, a categorization is proposed with two types: the search algorithm and the surrogate algorithm. Differences and connections of two categories are given. Both categories are explored for a filter multi-objective optimization problem.
- Machine learning aided motor global optimization. In order to speed up the conventional finite-element analysis (FEA) based motor optimization, in this thesis, an ML based surrogate correction model is proposed to bridge the gap between analytical model and FEA model. Based on that, the performance estimation of analytical model can be corrected very close to the FEA model outputs in the whole design space. Finally, a motor multi-objective optimization problem is investigated by using two ML models with two objectives: mass and power losses.
- Machine learning aided joint-simulation based motor-drive power loss (PL) optimization. After establishing the simulation of a motor drive (a two-level converter + a motor), the PL data from both the analysis and detailed simulation model can be collected to train an ML surrogate correction model. Similarly, in a design space, the motor-drive losses generated from the analytical model can be perfectly corrected to the target data of simulation model with the help of the ML surrogate model. Then, a single-objective optimization is done for the PL of entire motor drive system.
- A machine learning based controller for modular multi-level converters. The conventional FCS-MPC suffers the heavy computation burden especially for

the multi-horizon mode. To address this problem, ML based controller is proposed with a short computation which, at the same time, has a similar control performance to the original FCS-MPC. This ML controller is designed with two types: ML regression and ML pattern recognition. Both models are investigated and tested in simulation and with experiments.

1.4 Thesis Outline

This thesis is structured into seven chapters, covering topics related to the design, optimization and control of aircraft electrical drive systems. Chapter 2 illustrates two categories of the studied algorithms (i.e. search and surrogate) and serves as the basis of this thesis. A genetic algorithm (GA) and a neural network are given as the examples of two algorithm categories. Comparisons between two categories as well as the related literature are discussed in Chapter 2.

In Chapter 3, an optimization-based design of the first subsystem, an *LC* filter, is illustrated using both the search and ML algorithms. In addition, the design and optimization methodology are used for two different electrical drive systems: a motor drive system and a linear load drive system. In the ML approach, the inputs of ML models are the design variables and the outputs are the optimization objectives. Finally, three filter prototypes (obtained from the GA optimization results) were manufactured and used in the experimental validation. The experiments prove that, after the GA optimization process, the best filter design not only saves much mass and power losses but also gives a good control performance when used with the FCS-MPC.

Chapter 4 explores the possibility of using the ML algorithm to design controllers for a DC-AC power converter. The training data is collected from the normal FCS-MPC operation of the converter. However, after training, the trained ML models can be used to replace the role of FCS-MPC. Namely, the converter-based drive system can be controlled by the trained models without any help of FCS-MPC. The main motivation

is to reduce the computation time of the real-time control using the conventional FCS-MPC. The theory basis of using ML is that the FCS-MPC process is completely deterministic: for the same set of circuit measurements and a given cost function, the inserted vectors will always be the same. Two different ML controllers are designed. Inputs of two controllers are both the same with the conventional FCS-MPC but their outputs are using different designs. After training and validation, the trained models are tested both in simulations and on an experiment rig, under different conditions. Their advantages and disadvantages are then discussed after the online tests.

Chapter 5 presents the application of the surrogate (i.e. machine learning) algorithm for the optimal PMSM design. Two different neural networks (NNs) act as the dedicated correction models for the two PMSM objectives, mass and power losses. The NN inputs are the studied design variables and the outputs are the correction factor(s) of the objective, mass or power loss. After training the networks, the PMSM optimization-based design can be smoothly executed by only using the analytical model and two networks (without FEA models), because the networks can correct the performance estimation of analytical model to the FEA-based results with very high accuracy. Finally, this ML aided approach is compared with the conventional FEA-based optimization which is based on search algorithms.

Similarly, the ML based correction surrogate is used for the entire motor drive in Chapter 6. In this chapter, simulations of the PMSM and a two-level inverter are established to give the target data to train the surrogate model. In the neural network training, only 27 samples (from analytical model and simulation model) are needed. The NN inputs are the studied design variables and the output is the correction factor of the motor-drive power loss. After training, the excellent surrogate correction performance is validated by 64 samples where only 8 of them are included in the training sample set. Lastly, the surrogate model is utilized to give the best design of motor drive operation according to the minimum power loss.

To conclude the thesis, Chapter 7 summarises the contributions and findings, and suggests potential future works to extend this research, in the area of optimization and control of aircraft electrical derive systems.

Chapter 2 State of the Art: Design, Optimization and Control of Electrical Motor Drive Systems Using Two Types of Algorithms

The previous chapter has presented an introduction and background to the work described in this thesis, initially introducing the electrified aircraft architectures and DC-AC electrical drive systems. This chapter will focus on relevant previous work published by others in this area and the commonly used optimization/control algorithms. The contents in this chapter are divided into four sections: drive system design & optimization, DC-AC inverter control, commonly used algorithms, and algorithm categorization.

2.1 Design and Optimization of Motor Drive Systems

An electrical drive system can be considered to comprise of two subsystems: a converter and a load. In this thesis, a DC-AC inverter and a three-phase load (motor or linear load) are considered. The motor is assumed to be a permanent-magnet synchronous motor (PMSM). In addition, an input *LC* filter is considered at the input of the inverter, this smooths the DC bus ripple to meet the power quality and EMI requirements [28, 29, 31]. The function of the EMI filter is to reduce the EMI from the inverter to the dc bus, i.e. filter out the harmonics of inverter switching activities. As will be discussed in Chapter 3, the EMI requirements specified here are mainly from the military 704 file [32]. In this section, the design and optimization of the motor, power converter, and the entire drive system are reviewed and discussed.

2.1.1 Subsystem design and optimization in a drive system

Many studies have been done for the subsystem-level design and optimization of electrical drive systems. In [28, 29], the DC filter design and optimization problem is studied for an airborne drive system regulated using finite control set model predictive control (FCS-MPC). For aircraft applications, the design and optimization problem usually considers the mass as a critical objective. It is claimed that, for the whole air transport industry, saving 1 *kg* on each flight could save roughly 1700 tons of fuel and 5400 tons of CO_2 per year [11, 33]. In addition, mass reduction could also save the cost of aircraft systems/components. Therefore, the authors in [28, 29] regarded mass and power loss (PL) as the two objectives of the filter optimization. In these two papers, the basic filter design and optimization method is the same though the AC loads are different (one is *LC* filtered linear load, one is a three-phase PMSM). Their method uses circuit simulation and the filter sizing models to find the optimal filter design, where heterogenous constraints are considered during the genetic algorithm (GA) based optimization.

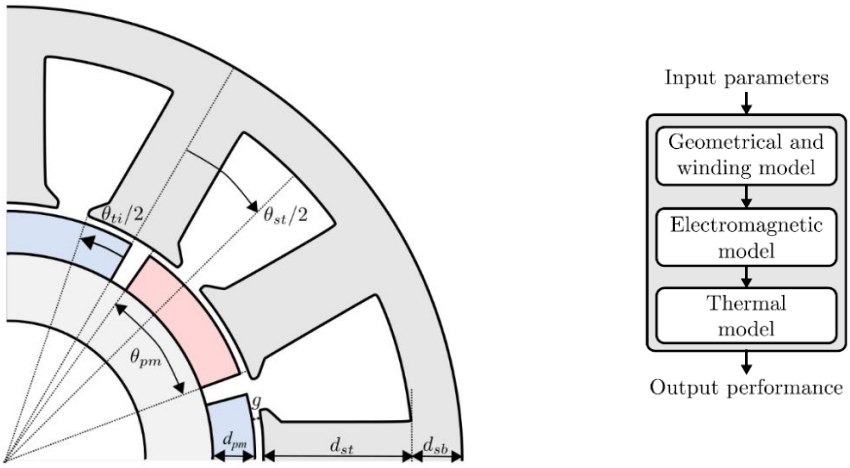
It is noted that there are conventional analytical design methods for filter design [34]. For example, the conventional inductance design is based on the maximum allowable current ripple, converter switching frequency and the voltage drop during switching [34-36]. For the capacitor design, the minimum DC capacitance value is often determined by the maximum allowable voltage ripple, converter switching frequency and the DC-link current [37, 38]. However, for the non-modulated FCS-MPC regulated drive system, the switching frequency is not fixed in the converter control, thus the conventional methods are not applicable. The key role of [28, 29] is to use the components' data and the embedded simulation to optimize the filter in a 3-dimentional design space, which is independent with the drive system switching frequency. See Section 3.1 for more details.

In the aircraft industry, strict requirements are considered for electrical motors in terms of reliability as well as power and torque density [39]. Different types of motors onboard aircraft have been studied, for example PMSMs [12, 30, 40-44], brushless DC machines [45], switched reluctance machines [46]. Among these candidates, PMSMs often offer the highest power density and efficiency, due to low torque ripples, the absence of rotor copper loss, and the replacement of any field windings by high-energy rare-earth magnets to provide excitation [12, 47].

A fast, accurate, and high dimensional multi-physics analytical model for a PMSM was presented in [42], as shown in Figure 2.1. This model was non-iterative and had 20 design variables which includes 12 geometrical parameters. Figure 2.1(a) shows a simplified geometry diagram of the studied surface PMSM with 7 geometrical parameters: PM depth (d_{pm}), airgap height (g), tooth depth (d_{st}), stator back-iron depth (d_{sb}), PM angle (θ_{pm}), tooth tip angle (θ_{ti}), and slot angle (θ_{st}). Other geometrical parameters are mainly in the tooth tip part. Figure 2.1(b) shows the general flowchart of the motor design model. It considered both electromagnetic and thermal behaviour of the surface PMSM. The model evaluated machine electromagnetic performance by calculating the air-gap magnetic field distribution and employing mathematical magnetomotive force winding functions. Furthermore, the machine thermal behaviour was predicted using a simplified 2-dimentional lumped parameter thermal network. Finite-Element Analysis (FEA) and experimental validation were done for some typical design points of PMSM. The resulting analytical model took an average of 0.03 *secs* to run on a standard PC. The efficient model may not guarantee high accuracy, especially in a large design space [48]. However, some typical points have been validated in simulations and with experimental results [42].

Particle swarm algorithms have been used for some machine optimization problems. In [49], a particle swarm based optimization for the permanent magnet (PM) brushless machine design was reported. The machine cross-section topology is shown in Figure

2.2. Four parameters were selected as the design variables to be optimized: motor active length, rotor diameter, teeth width, and magnet height [the last three variables, which are in the motor cross-section, are shown in Figure 2.2(b)].



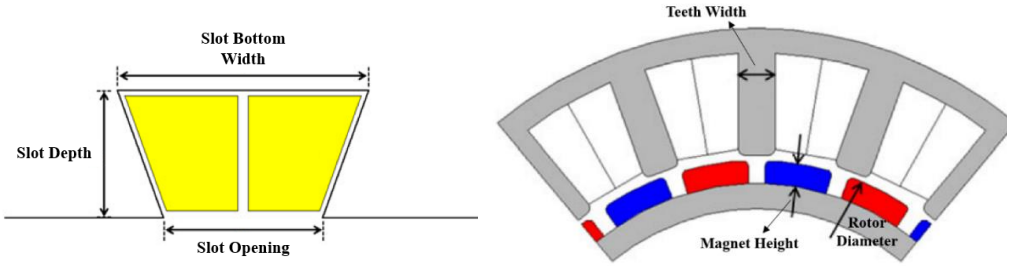
(a) (b)

Figure 2.1 Diagram of the surface PMSM design and optimization problem in [42]. (a) Simplified diagram of a typical surface PMSM. (b) Flowchart of the motor design model.

An analytical closed-form PM machine solution in [49] was used to predict the electromagnetic performances and active weight. The analytical model included both open-circuit magnetic field and armature reaction field distribution with stator slotting effects. Based on the analytical model and FEA, the authors presented an entire design process including both analytical solutions and systematic optimization technique for a fractional-slot PM machine using concentrated windings. Finally, particle swarm optimization technique was applied to the analytical model to search for solutions that were a good compromise between machine conflicting objectives.

Reference [50] utilized explorative particle swarm optimization to find the optimal design of a PMSM with a mesh adaptive direct search. Although the authors declare that the proposed rule of start point selection takes advantage of minimizing the search time, the proposed approach was not quantitatively compared with other search methods regarding the computation time. The shape optimization of a PMSM was considered in

[51] using an analytical Kriging surrogate model (a statistical-based interpolation method). However, this method requires a convergence of the FEA-based iteration, thus it may be time-consuming due to the FEA convergence.



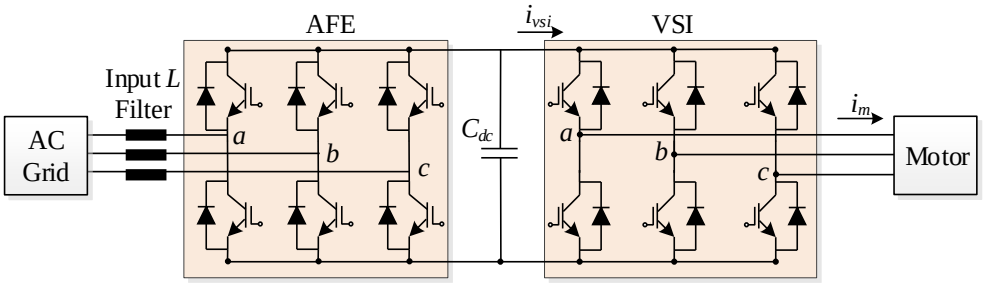
(a) Shape and parameters of stator slot

(b) Geometric configuration of the surface PM machine

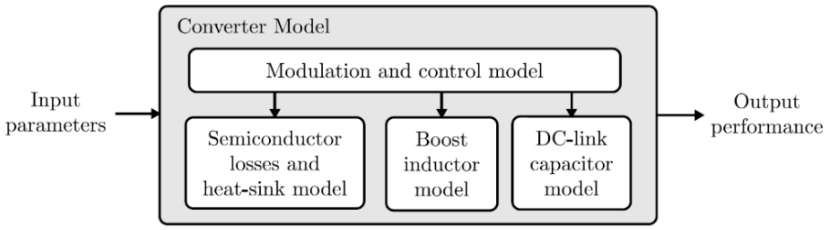
Figure 2.2 Diagram of the Fractional-Slot PM Machine in [49].

Machine learning (ML) aided PMSM design and optimization is still rarely reported. [52] reported a neural network (NN) to build surrogates for machine optimization which have the best accuracy versus training time tradeoff among several surrogate candidates but, most of the NN technical details (for example sample collection, NN training and test) were not given. In Chapter 5 of this thesis, an ML-aided PMSM multi-objective optimization problem (MOOP) will be discussed. The ML aided approach only requires a small number of FEA-based samples but can, in the whole design space, correct the analytical performance estimation with high accuracy. As the time taken for FEA-based data collection can be very short, it is justified by an extremely fast optimization time using the trained NN correction model.

Different from the PMSM, most studies on the power converter design and optimization focus on operation design parameters (such as DC-link voltage, switching frequency) for the performance estimation, rather than the geometry parameters. In order to optimize a two-level voltage source back-to-back converter (VSBBC), [53] proposed a fast and accurate multi-physics model of this AC-to-AC converter. The basic topology and the model flowchart are shown in Figure 2.3.



(a) Basic schematic diagram of VSBBC



(b) Flowchart of converter model

Figure 2.3 Topology and model of a two-level Voltage-Sourced Back-to-Back Converter (VSBBC) [53].

The conventional PI control was used for both active front-end (AFE) rectifier and the voltage source inverter (VSI). Symmetrical space vector modulation strategy was used. Based on that, the voltage drop across the boost input *L*-type filter can be formed and the minimum inductance can be obtained, in a conventional way [12]. The considered parameters included the DC-link voltage, the DC-link capacitance, the input boost inductance, AFE and VSI switching frequencies, the grid and machine related parameters. Finally, the power loss was estimated and experimentally validated with regards to the DC-link voltage, switching frequencies and the phase current.

Another power converter design and optimization study can be found in [54], where filter component models for a VSI system (shown in Figure 2.4) were proposed and embedded in a system design tool for the inverter power density optimization. The considered passive components in the VSI system include a DC-link capacitor at the input and an output *LCL* filter. The capacitor limits the DC-link voltage ripple and the output filter limits the output AC current ripple. Similarly, passive component values

were computed using analytical equations for the inverter optimization. However, advanced Silicon Carbide (SiC) devices were considered and the switching frequency for inverter control was up to 400 kHz. The comprehensive design tool optimized the selection of the devices, heatsink and passive components (including the design of the line, EMI, and dc-link filters) to maximize the inverter power density.

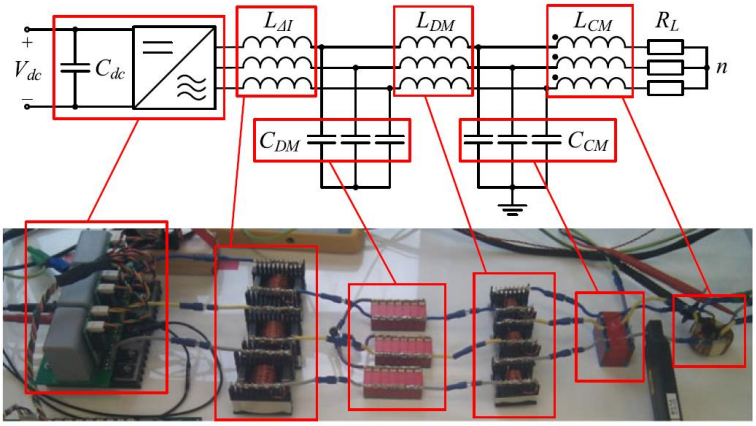


Figure 2.4 Basic schematic diagram and experimental set-up of a 3-phase DC–AC converter [54].

2.1.2 Drive system design and optimization

At the system level, the optimization-based design problem is usually a more complex engineering problem. A system-level optimization of a motor drive design was presented in [55] using an advanced evolutionary multi-objective optimization algorithm, Generalized Differential Evolution 3 (GDE3) [56]. The motor drive topology is shown in Figure 2.5. Based on the proposed analytical motor-drive model, the case study considered 14 design variables, 10 constraints and 2 objectives for the motor drive optimization. Two design points (obtained from the optimized Pareto front using the analytical model) were selected for further verifications using the 2-dimentional finite-element analysis (FEA) and time-domain simulation software. Photos of the surface PM machine prototype, voltage-source converters, and toroidal core inductors used for model verification are shown in Figure 2.6. It was noted by the authors that, in practice, a final designed system will unlikely fall directly on the Pareto front as desired and the

design solutions should be subject to extra experimental verifications. However, the employed analytical models in [55] have already been experimentally verified in [41, 42, 53]. Finally, two analysis examples of the effects of varying the drive system input requirements were further evaluated: (1) Effect of Machine Operating Points; (2) Effect of AC Grid Input Voltage.

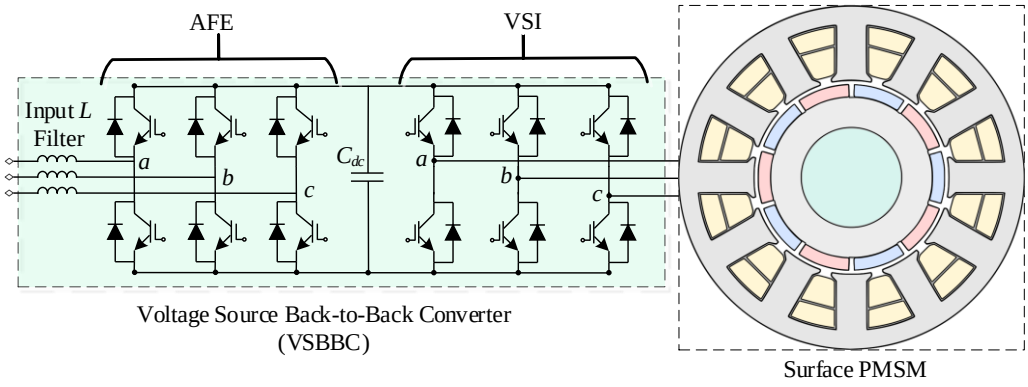


Figure 2.5 Simplified diagram of a VSBBC-based motor drive system.



(a) Toroidal core inductor for the input L-type filter.

(b) Two level voltage-source converter with dc-link capacitors and heatsink

(c) 12-slot 10-pole PM machine

Figure 2.6 Photos of prototype motor drive components in [55].

In [30], a complex multi-level optimal design problem for an inverter based motor drive system using genetic algorithm (GA) was presented (with two objectives: mass and power loss). The schematic drive system architecture is shown in Figure 2.7. The optimal design example refers to the integrated design of an electrical drive system including a high-speed PMSM supplied by a VSI associated with an input filter. In addition, the flight mission profile (for PMSM operating points) and the high-voltage

DC network standards considered for the power supply were also discussed. The input filter subsystem included two cells. The topology of both cells is *LCC*, i.e. one inductor is cascaded to two paralleled capacitors. The reason of applying such a complex topology is that, in [30], the authors considered the "worst case" in terms of voltage variation, which generated an input High Voltage Direct Current (HVDC) voltage considering all maximum harmonics according to the Airbus HVDC standards [57]. In order to obtain an acceptable DC voltage ripple at the input of VSI, a complex filter topology should be used.

Sequential optimization and global optimization were described in [30]. These two approaches have previously been applied [11] to do the subsystem sequential optimization and the global integrated system-level optimization. In sequential optimization, two local optimization problems for a PMSM and a "filter + VSI" combination were formulated. The PMSM optimization problem studied 11 variables and 11 constraints while the "filter + VSI" problem studied 10 design variables and 11 constraints. Both of them considered two local objectives: mass and total losses during the flight mission. In the global optimization, all three subsystems (filter, VSI and PMSM) in the motor drive were considered, thus the complexity was significantly increased, with 21 design variables and 22 constraints in total (all design variables and constraints in sequential optimization problems are aggregated). Noting that these two approaches (sequential and global) were only used for the optimization of one system level.

In order to find compromise solution between two levels (system level and subsystem level), a multi-level optimization problem of the entire motor drive system was discussed in [11] with two different formulation structures: two subsystems (PMSM, "filter + VSI") and three subsystems (PMSM, filter, VSI). As shown in Figure 2.8, the formulation used an analytical target cascading (ATC) method [58] which can link variables between different system levels based on the detailed analytical models.

Generally, the ATC method between the two-level system means the upper system level contains the common (coupling) variables between component (subsystem level) problems as decision variables. These common variables involved in the design of PMSM and the "input filter + VSI" (two-subsystem formulation) were circuit parameters of the motor drive: pole number p, stator winding resistance R_S, motor inductance L_{dq} and flux linkage $flux$. Another ATC formulation scheme is with the three subsystems: PMSM, input filter and VSI. In this formulation, the upper level contains the same decision variables (p, R_S, L_{dq}, and $flux$) with additionally the switching frequency (f_{sw}) to coordinate between the filter and the VSI problems.

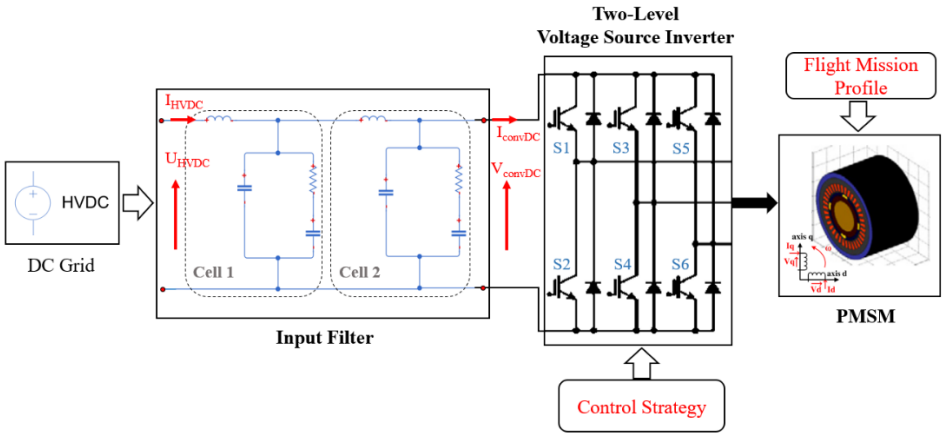


Figure 2.7 A motor drive system architecture for a more-electric aircraft application.

Based on the component models and two formulation structures, a GA based optimization process was introduced with six steps. Finally, all results of "sequential optimization", "global optimization" and "multi-level optimization" approaches were compared. The Pareto fronts of the multi-level ATC formulation problems deal near the "global" solution and clearly outperform the "sequential" approach. The 2-subsystem formulation better converges than the 3-subsystem formulation because the number of decision variables of the system level is greater in the second formulation. However, the calculation of 2-subsystem formulation costs much more time than the 3-subsystem formulation (34 min vs. 9 min).

There are three optimization problems described in this thesis: the input *LC* filter optimization problem will be studied in Chapter 3 and the PMSM optimization will be studied in Chapter 5; Furthermore, a inverter-based motor drive system (DC-AC inverter + PMSM) will be considered in Chapter 6, which focuses on the machine learning (ML) aided power loss optimization. A joint simulation of the motor drive is built in Chapter 6 for the simulation data collection. Common parameters from the PMSM subsystem model are feedforward to the system-level drive simulation. The ML based surrogate model is proposed to be a correction surrogate smoothly linking the analytical model estimation data to the target simulation data. Based on this, the power loss optimization in the design space can be completed efficiently.

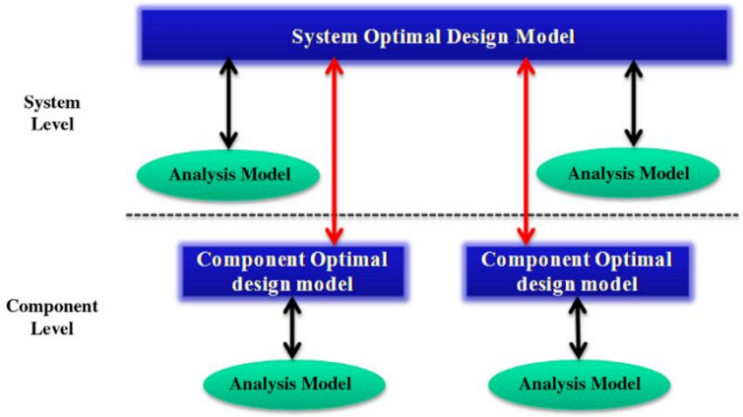


Figure 2.8 Analytical Target Cascading (ATC) formulation (simplified version) [30].

2.2 Drive System Control

The system studied in this thesis is an inverter-based drive system (DC-AC) for a more-electric aircraft application. In the aircraft electrical power system, this drive system is usually connected to a DC bus as an end-user. Such an inverter-based load can extract steady power even under a varying voltage condition at the DC bus. Therefore, it is commonly referred to as a constant power load (CPL). One of CPL's characteristics is that it tends to destabilize the system to which it is connected, because its incremental input impedance is negative.

This challenge was commonly analysed using an impedance-based approach. In this approach, the ratio between the DC source and CPL impedances Z_{out} and Z_{in} (i.e. minor loop gain) determines the system's stability [59]. There are three approaches to properly shape two impedances: passive damping of Z_{out}, active damping of Z_{out}, and active damping of Z_{in} [20]. Passive damping of Z_{out} usually requires large passive components [29, 31] thus this approach is not ideal for aircraft applications. For the active damping approach of Z_{out}, addition of active converters solely for stabilization purposes is usually too costly. In addition, the active-front end interface is not available in many industrial applications where only unidirectional power flow is required [20]. Therefore, this thesis employs the third approach for the DC interface stabilization, i.e. shaping CPL impedance Z_{in}.

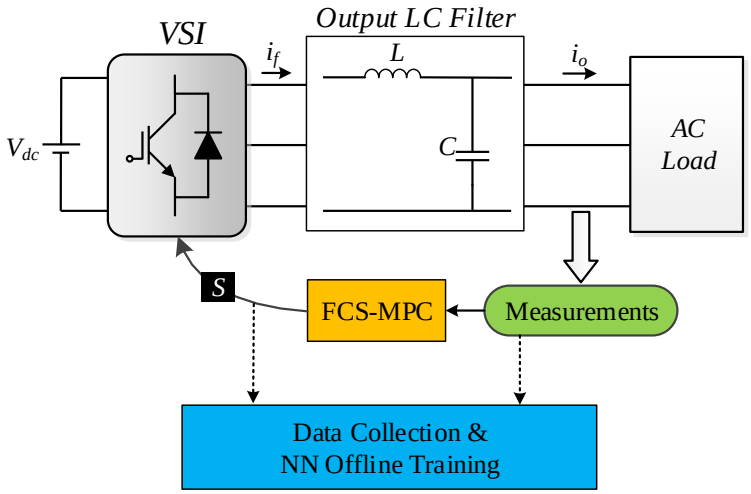
The control strategy used in the inverter-based CPL is the finite control set model predictive control (FCS-MPC). The main reason of not using the cascaded linear control loops is that its adjustment of Z_{in} can only be done by reducing the loop gain of the control system, which inevitably degrades the load regulation performance [60]. During each sampling period, FCS-MPC strategy tries every possible control vector in the finite control set for the converter (8 vectors for the two level and 27 vectors for the three level) and find the optimal one to be applied at the next time instant. FCS-MPC is not restricted to a predetermined switching pattern like Pulse-Width Modulation (PWM) and therefore allows much better transient response compared to linear control methods [20]. In addition, it has a significant advantage over linear control methods, where control parameters need to be carefully tuned to ensure stability [61].

As well as the variable switching frequency, another drawback of FCS-MPC is the computation burden, which can be extremely high for situations with multi-step predictions because all possible control vectors for every single period need to be considered. To reduce the computational burden, [62] proposed to train FCS-MPC for a two-level converter using a neural network (NN). The overview of the proposed NN

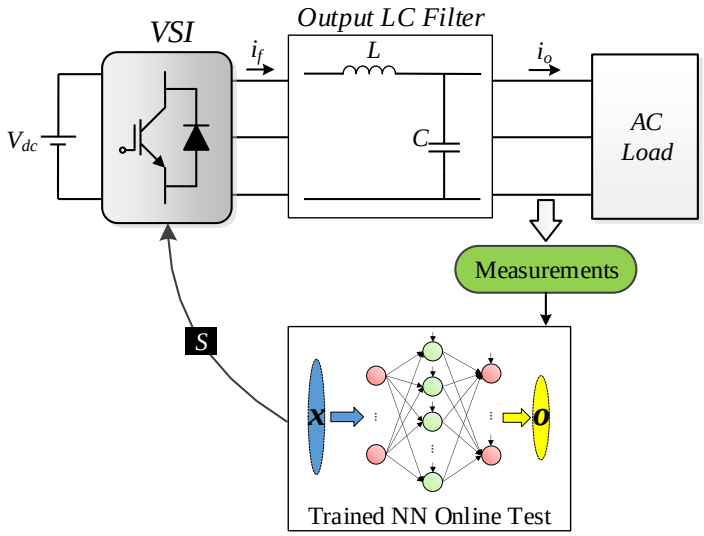
control strategy is shown in Figure 2.9. There are two phases: an offline training phase and an online test phase. The training phase (Phase one) combines between using FCS-MPC for predicting the output voltage of the inverter and collecting data, under full-state observation. This phase is for training the neural network offline. In the test phase (Phase two), the trained neural network is employed online to control the output voltage of the inverter instead of the original FCS-MPC, considering linear and non-linear loads. The basic NN usage method is similar to the machine learning (ML) method used in this thesis: collect data from the original system, train the ML model offline and then test it online. Though the authors in [62] presented the testing of a large number of raw control data points, it is not experimentally validated, with only simulation results provided. It is known that, simulation models are relatively ideal without noise while, in experimental operation, NN prediction performance, practical noise and uncertainty in-circuit all affect the practical control performance.

In [63], a similar approach was proposed to imitate FCS-MPC and implemented the NN approach in an experiment rig. The paper highlighted that the NN based imitator is not intended to improve the performance of the original FCS-MPC algorithm, but to reduce the computational burden. In addition, the authors demonstrated that it is not necessary to run a simulation model for data collection. Instead, only the FCS-MPC algorithm function was used to collect the NN training data. As shown in Figure 2.10, the NN pattern recognition (PR) approach was explicitly applied for training the NN-based controller (based on the raw data collected from the original FCS-MPC algorithm). The inputs to this imitation NN include all the AC measurements in the inverter-based drive system: converter output current ($\bar{\imath}_f$), output filter voltage ($\bar{v}_f$), and output current ($\bar{\imath}_o$), which are also the inputs of the original FCS-MPC algorithm. The NN outputs (total of seven output elements) are all binary values (0 or 1) thus, this study was a classification problem in machine learning. Moreover, the NN outputs, control gate

signals (S_{abc}), are back propagated, as a part of NN inputs. This recurrent design was to improve the training and test performance of the NN imitation controller.



(a) Phase one: data collection and NN offline training



(b) Phase two: test the trained NN online

Figure 2.9 An overview of the NN based control strategy for two-level inverters.

In [64], a general supervised learning framework was proposed to approximate a model predictive controller for nonlinear systems. The NN approximator implementation was evaluated to be 200 timed faster than the original algorithm, showing how this approach can enable computationally efficient implementations of MPC algorithms. However, this approach is only evaluated by a numerical example, and not in the power electronics domain. In [65], a NN imitation controller of long-

horizon FCS-MPC was applied into a two-level inverter based motor drive system. Therefore, different from the studies in [62, 63], this is an AC current controller for a three-phase motor. In addition, the NN controllers for the multi-step prediction horizons up to N = 5 steps were analyzied, implemented and tested in simulations. In addtion, the trained NNs were deployed into a MicroAuto-Box II 1513/1514 for a hardware in loop (HIL) simulation and the execuation time (i.e. turnaround time) was recorded for the NN performance evaluation.

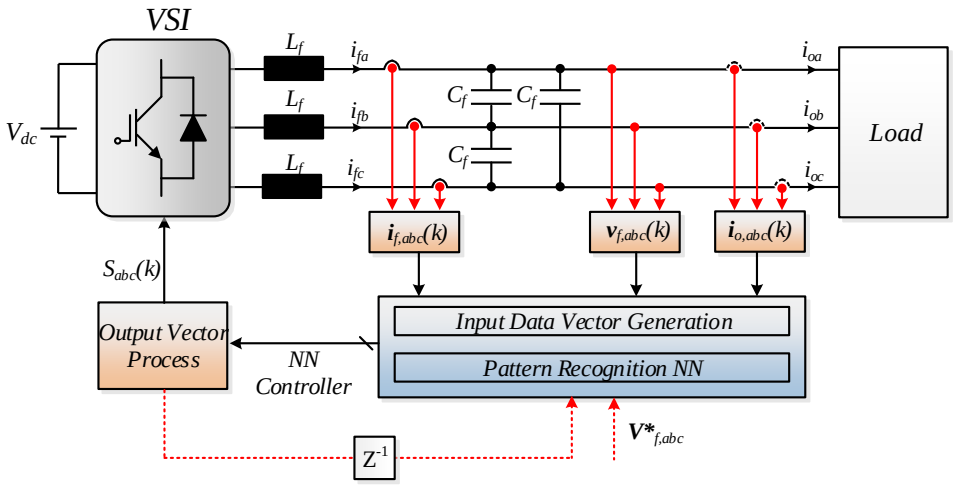


Figure 2.10 Simplified scheme of the NN based imitation control algorithm in [63].

It is noted that there are other strategies to avoid higher computational burden of searching the best vector in using long-horizon FCS-MPC. For example, a K-best sphere decoding algorithm (SDA) was proposed in [66] to solve the computation problem of long prediction horizon. The K-best SDA establishes a different breadth-first search strategy to address some of the main drawbacks of the conventional SDA. However, this thesis will mainly study using the NN imitation model to solve the computation problem.

In the Chapter 4 of this thesis, the NN imitation approach is further explored for a DC-AC modular multilevel converter (MMC) drive system. Similarly, two proposed NN controllers were trained using the data collected from the original FCS-MPC controller with one and two prediction horizons. However, two different NN training

schemes (regression and classification) will be both explored, their testing results in simulations and experiments will be discussed.

In summary, both the optimization algorithms and the ML algorithms have attracted much attention and been widely applied in the design, optimization and control topics of electrical drive systems. However, most of ML applications are mainly for the drive-system control topics [63, 67, 68], the applications of ML on the design and optimization are rarely reported [69, 70]. Some commonly used optimization and ML algorithms will be discussed and then, a simple categorization strategy of these algorithms will be presented.

2.3 Research Algorithms

2.3.1 Optimization algorithms

The design and optimisation of electrical drive systems can generally be split into two parts: modelling and optimisation. Modelling is used to establish the relationship between the system parameters and the system performance (for example power losses, mass, temperature). The modelling techniques can include the use of analytical equations, lumped parameter models, and numerical analysis [12, 71]. Based on the design models, optimisation techniques are generally utilized to search for a trade-off solution for the multi-objective optimization problem (MOOP), i.e. finding the best design point in a variable space subject to the constraints. The constraints usually include system requirements and practical limitations (for example, temperature rise limit, maximum tracking error, maximum total harmonic distortion). This design process can be referred as optimization-based design [72].

As shown in Figure 2.11, the optimization-based design process consists of three parts: the optimization engine, objective function and detailed analysis [72]. Detailed analysis is a modelling process, which provides an understanding of the real-world system and models it to take in a set of input design parameters and, correspondingly,

calculate and output values of interests, such as temperature rise, power loss, and weight. Based on these interests, objective function generates the objective values for optimization which are direct indices for the fitness calculation and iteration. Optimisation engine refers to the function of optimization algorithms, it can provide insight into possible designs that are optimal as defined with a set of objective values. The optimization engine is depicted at the outer level because its computational algorithm is operated on the design variables and the objective function during all the optimization generations. The detailed analysis inside (i.e. modelling work for the optimization) is used to calculate design metrics (system performance) and check the constraints, such as achieving minimum acceptable performance, physical constraints, temperature upper limit, etc. The objective functions are usually designed by the system performance metrics thus, they should be obtained based on the detailed analysis/models.

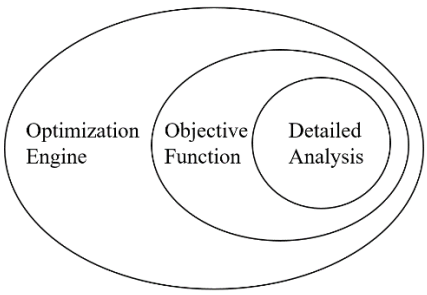


Figure 2.11 Optimization-based design process.

The system formulation of an optimization-based design problem varies from problem to problem. In addition, the architecture of a complex system is usually hierarchical for different engineering and technology problems: system-of-systems level [73], system level [12, 21, 30, 55, 74] and subsystem level [12, 28-30, 34, 35]. Therefore, it is understandable there is no single optimization algorithm that will work for all problems with equal speed and performance. This explains why there are many applicable algorithms, each suitable for a typical problem. In general, optimisation

algorithms can be divided into two categories: deterministic algorithms, and stochastic algorithms, as shown in Figure 2.12 [12, 75].

Deterministic algorithms will always produce “repeatable” outputs when given certain inputs. However, they rely on a comprehensive and systematic enumeration of all the studied design variables and have a strict execution procedure (for example being defined in terms of a state machine). However, deterministic algorithms are generally not computationally efficient for problems with many design variables if direct search approach is employed, due to an inherent “step-size” problem [76]. If the chosen step-size is too small, this leads to excessive computational efforts, whereas if it is too large, optimal points might be missed. In contrast, stochastic algorithms do not have a step-size problem and allow a finer scan in more promising (bigger fitness) regions compared to areas with fewer prospects. They usually start with or consider random parameters thus the optimization results could vary for different runs. If run with large number of samples, however, the final optimal results should be the same or very similar, according to the law of large numbers [77]. Though stochastic algorithms possess inherent randomness, according to the law, the average of the results obtained from a large number of trials will tend to become closer to the expected value as more trials are performed. Further, stochastic algorithms are gradient-free (i.e. do not need to consider gradients in optimization) and have the capability to avoid the local optimum issues based on the continually evolving population [55, 76].

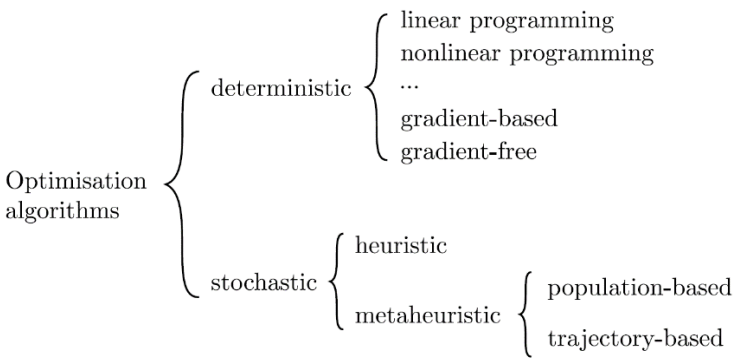


Figure 2.12 Classification of optimisation algorithms [75].

Some examples of the classical deterministic algorithms are dynamic programming [78, 79] and integer programming [78, 80, 81]. For the stochastic algorithms, they can be further divided into two groups: heuristic and meta-heuristic, which include many well-known search algorithms, for example genetic algorithm (GA) [12, 78, 82, 83], particle swarm [12, 49, 84], tabu search [78, 85]. Heuristic algorithms are developed in all kinds of approaches variant from the simple "trial and error" approach to the complicated algorithms such as evolutionary algorithms. Heuristic is a technique designed for finding an approximate solution efficiently, which can address the time-consuming problem of classical algorithms. It also works well when classical algorithms fail to find any exact solution. A metaheuristic refers to a master strategy that guides and modifies other heuristics to produce solutions beyond those that are normally generated in a quest for local optimality [86]. Therefore, compared to some simple heuristics, metaheuristic algorithms generally perform better on computationally complex problems.

2.3.2 Machine learning algorithms

A lot of the published literatures focus on using classical or modern optimization algorithms to search for the optimal solutions for aircraft EPSs [12]. Machine learning (ML) algorithms are rarely found in the optimization studies for aircraft EPSs. ML is the scientific study of algorithms and statistical models that computer systems use in order to perform a specific task effectively without using explicit instructions [87]. ML algorithms can be divided into three groups: supervised learning, unsupervised learning and reinforcement learning [69, 70].

Supervised learning methods in ML have outputs (also called as targets) defined in the training datasets. These targets can either be continuous numbers or integers. When the targets are continuous numbers, the learning task is called as regression. On the other hand, when the targets are integers, the learning task is known as classification. Therefore, both regression and classification are training the sampled input-output data

but their outputs data features are different. Regression has no requirement for sampled data but, in the classification learning, the training outputs should include different categories/classes or even only 0 or 1 for some specific problems. Supervised learning can be used either online or offline.

For the unsupervised learning, different with supervised learning, there is no pre-defined output in the training data. It is learning how to discover patterns and information from a dataset without pre-set outputs/labels/features. Such methods can also be used to reduce a dataset dimensionality without losing important information [69]. Therefore, the training data in unsupervised learning can be very simple only with several elements (for example, in a flower category case, there are 4 elements: sepal length, sepal width, petal length, and petal width). Then unsupervised learning can discover the information was previously undetected (for example categorize different kinds of flowers using some sampled data of 4 elements). Unsupervised learning problems can be further grouped into clustering and association problems. Clustering mainly deals with finding a structure or pattern in an uncategorized/unlabelled dataset. The flower category case is indeed a clustering problem. Association rules can discover associations/relationships between elements in large databases. For example, people that buy a new house most likely to buy new furniture.

The third group of ML is reinforcement learning (RL). RL is learning how intelligent agents can perform a task by interacting with their environment. In the learning process, these agents ought to take actions (i.e. updating the policy) in an environment (getting rewards from each step/trying) in order to maximize the cumulative reward [88]. In other words, at the beginning, how to operate in the environment is unknown, the agents should try different actions in the environment to receive the corresponding rewards. Based on the cumulative rewards, the agents can learn a good policy (which means how to act in a certain position) in the environment.

Obviously, RL is an online learning. It differs from supervised learning because no input/output data will be required or presented in RL.

Supervised learning is the most popular technique among the three groups in ML. According to the statistics in [70], usage of supervised learning is 91% of all ML applications for the power electronics (444 journal papers from 1990 to 2020 were identified). Therefore, this thesis will mainly focus on the supervised learning techniques for the design, optimization and control topics. RL can be a good solution for future studies, as will be discussed in Chapter 7.

In this thesis, ML algorithms will be applied to several optimization-based design problems for the aircraft motor drive systems, including ML aided filter multi-objective optimization in Section 3.5, motor design and optimization in Chapter 5, motor-drive optimization in Chapter 6, and inverter predictive imitation controller in Chapter 4. As shown in Figure 2.13, the optimization engine would be just global exhaustive algorithms because the objective function and the detailed analysis only involve the mathematic ML models, which can provide the desired outputs extremely fast when given an input combination. Obviously, after training, there is no need for detailed derivations in this optimization process. This is due to the fact the original relations between inputs and outputs have been learned by the ML models, based on the collected and processed sample data. Therefore, the main motivation for using ML is due to the ML algorithms representing some advantages over search algorithms [28]. After an ML model is trained, it can be used directly in any future study of this optimization problem, instead of running the detailed system model all over again with search algorithms, which can be very time consuming. Apart from the efficiency, other advantages of ML algorithms include: the trained ML models have no risk of getting stuck in local optimum; in addition, within the original design space, the ML models can be also feasible for the evaluation and optimization in any sub design space. Detailed

advantages of ML compared with conventional optimization methods will be given in Section 5.5, Chapter 5, based on a PMSM optimization study.

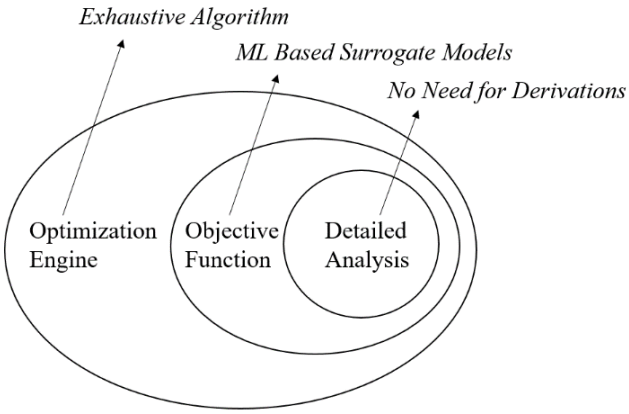


Figure 2.13 Optimization-based design process using ML surrogate models.

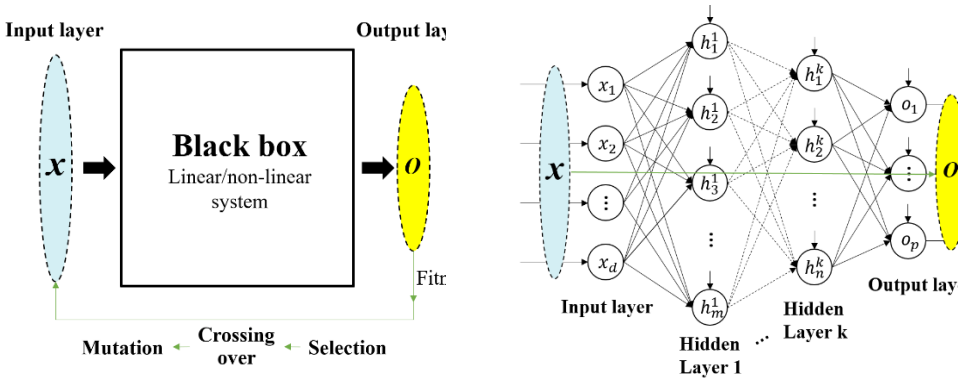
For the control imitation topic in Chapter 4, two different ML based controllers are designed to reduce the computation burden of FCS-MPC for the power converters. Both learning modes (regression and classification) will be used for the imitation supervised learning. Then, two ML controllers are tested in both simulations and experimentally.

2.4 "2SA" Categorization

The main purpose of this thesis is to use search and machine learning algorithms to find optimization design and control topics of aircraft electrical drive systems. It is valuable to investigate the differences between two kinds of algorithms. To this end, a novel category strategy is given for different commonly used algorithms including (GA) [12, 78, 82, 83], particle swarm [84], tabu search [78, 85], integer programming [78, 80] and artificial NN [89-94], Bayesian network [95], and support vector machine (SVM) [96, 97]. Based on this strategy, the relationship (***x***-to-***o***) between inputs (x_1, x_2, …) and outputs (o_1, o_2, …) in algorithm categories can be simply described and compared for the algorithm utilization.

As shown in Figure 2.14, there are two categories: search algorithm and surrogate algorithm(2SA). Both are describing the relationship of ***x***-to-***o*** but their logic directions (green arrows in Figure 2.14) are inverse: the logic direction of search algorithm is from output to input which is "outside" of a studied linear/non-linear system (which can be regarded as a black box); conversely, the logic direction of surrogate algorithm is "inside" the system because the data flow in the algorithm operation is always from input to output (***x***-to-***o***).

Noting that the logic direction shown in Figure 2.14(a) is not a data-flow direction but a "command". This means data in outputs (o_1, o_2, …) can change the inputs (x_1, x_2, …) in certain ways which are specific in different search algorithms (e.g. selection, cross-over, mutation operators in GA). Though, in the operation of surrogate algorithms, there is also a "command": outputs can change the internal specific training parameters in the algorithms (e.g. weights, bias in NN, margin in SVM), the inputs in surrogate algorithms are usually not changed or determined by the outputs. More specifically, in many ML applications, the input-output raw data are evenly sampled in the predefined design space of variables; after which, the desired ML models can be trained offline. Therefore, for the application of surrogate algorithms, the input-output data preparation and the model training are usually in separate steps.



(a) Search algorithm

(b) Surrogate algorithm

Figure 2.14 The proposed "2SA" categorization (giving GA and NN as the examples)

It is noted that the search algorithm considered here mainly refers to the stochastic algorithm shown in Figure 2.12. It is a technique designed for solving a problem more quickly than classic methods, or for finding an approximate solution when classic methods fail to find any exact solution. On the other hand, the surrogate algorithms described here are all using offline machine learning (ML) algorithms which do not consider system uncertainty, variations in time/space domain (spatiotemporal data) or noise models for the ML. Namely, the training is to learn a static ML model for a specific task and, the main motivation is to perform a specific task effectively without using explicit instructions or detailed simulations.

There are three groups of algorithms in ML: supervised learning, unsupervised learning and reinforcement learning. The surrogate algorithms considered here are all in the group of supervised learning. The basic idea to use these surrogate algorithms is first collecting the raw data online (from simulation or experiment), then processing the data for the machine learning and finally testing the trained model(s) online (in a real-time simulation or experiment). Some online ML algorithms (including reinforcement learning algorithms), which store past experience and compute optimal policies that maximize the long-term reward in the training, are out of the scope of this thesis (for example the Deep Deterministic Policy Gradient [98]). This is because, for reinforcement learning, the utility (cumulative reward) of the environment is always changing during the model learning process. That means, given one input (step), the output would be different. This situation is not considered in this thesis.

2.4.1 Search algorithms

The following section will discuss search algorithm and surrogate algorithm separately and then compare them in terms of purpose, operation mechanism, data in operation, model building and interactive functions.

The search algorithm directly focuses on the inputs and outputs of the system and treats it as a black box (no matter linear or non-linear). The search algorithm calculates

the fitness of each candidate (i.e. evaluates the performance of each candidate) based on outputs (o_1, o_2, …) (current generation) and then determine new inputs (x_1, x_2, …) in the next generation according to the algorithm principles. Using a GA as an example: mimicking the natural selection process, the GA searching process relies on the bio-inspired operators such as selection, crossover and mutation [82, 99] to produce a *Child* population based on the fitness of the *Parent* population. The GA requires both a genetic representation of the solution domain and a fitness function to evaluate the solution domain. The idea is to describe every candidate/individual of the population with a specific genotype (or chromosome). Genotype is a set of parameters which define a proposed solution to the problem that the GA is trying to solve. The set of all solutions is known as the population. This approach can solve both constrained and unconstrained optimization problems based on a natural selection process that mimics biological evolution (changing the chromosome to give new candidates). The algorithm repeatedly modifies a population of individual solutions. At each step, the genetic algorithm randomly selects individuals from the current population and uses them as *Parents* to produce the *Children* for the next generation. Over successive generations, the population "evolves" toward an optimal solution.

Figure 2.15 shows a generic procedure Genetic Algorithm with one objective. In this kind of optimization, the objective values of *Parents* (minimum as default) can be directly regarded as the fitness of this generation. Then, *Children* can be generated through selection, crossover and/or mutation operators. The selection is an operator to reduce the number of possible candidates choosing the one that leads to the best results after the fitness evaluation. The crossover operator re-combines the chromosome of the *Parents* to generate *Children* who have the mixed features. The mutation operator is used to avoid uniformity in the population which randomly changes one value of the chromosome. Noted that in Chapter 6, the single objective (power loss) optimization of the entire motor drive will be discussed using an analytical drive model and a trained

ML correction model. As both models are extremely fast, an exhaustive algorithm will be used for the global optimization instead of using a search algorithm. Search algorithms suit the situation that classic methods are too slow or the sampling cost of studied system is very high. Here, the sampling cost means the average time of getting one output when given one input by running the studied system.

In this thesis, a multi-objective GA is used to solve two optimization problems: the filter optimization (Chapter 3) and the motor optimization (Chapter 5). There are two objectives (mass and power loss) in both optimization problems. Being a population-based approach, GA are well suited to solve multi-objective optimization problems. A generic single-objective GA can be modified to find a set of multiple non-dominated solutions in a single run. The ability of GA to simultaneously search different regions of a solution space makes it possible to find a diverse set of solutions for difficult problems with non-convex, discontinuous, and multi-modal solutions spaces. In addition, most multi-objective GA algorithms do not require the user to prioritize, scale, or weigh the optimization objectives.

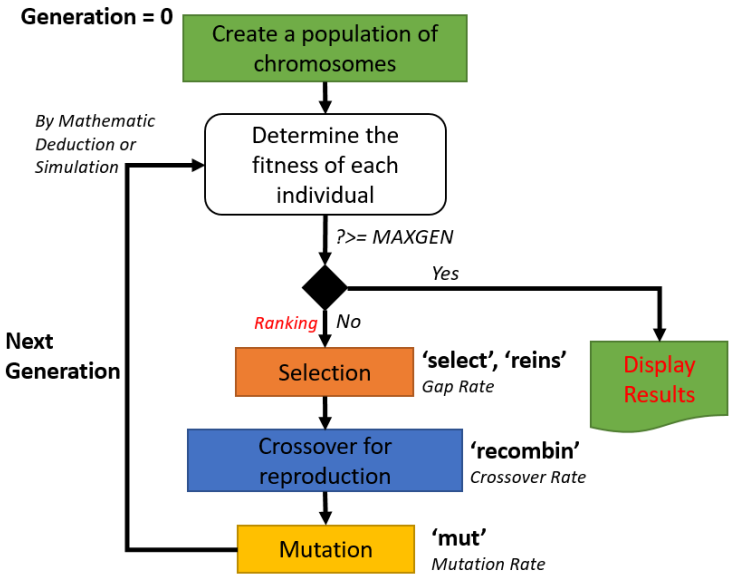


Figure 2.15 A generic procedure of using Genetic Algorithm with one objective.

There are several methods of multi-objective GA that can be chosen for the target optimization problems. The weighted-based Genetic Algorithm [100] is a classical approach which assigns weights to each sub-objective (the weights represent their importance) and linear weighted sum is regarded as the final fitness function, thus the problem is converted to a single objective problem with a scalar objective function. The Non-dominated Sorting Genetic Algorithm (NSGA) [56, 99] classifies the population into non-dominated fronts using an algorithm, in which the population is ranked according to a dominance rule, and then each solution is assigned a fitness value based on its rank in the population, not its actual objective function value. As implementing elitism and diversity preservation strategies usually require substantial computational effort and computer memory, Parallel Distributed Genetic Algorithm [101] attracts much attention to reduce execution time and resource requirements of multi-objective GA using advanced data structures.

Figure 2.16 shows the procedure for a GA using parallel selection: the crossover and mutation steps are the same with that for one objective GA. Every chromosome generation should be divided into two groups during selection process then, each group independently has its selecting deduction with regards to one objective (two objectives in total). After that, these two group are combined for the other two genetic operators which can be the same as that shown in Figure 2.15.

Other algorithms or GA variants such as particle swarm [84], tabu search [78, 85] also fit this searching characteristic and can be categorized as search algorithms. Particle swarm utilizes fitness values in current generation and math formulae to update the particle's velocity and uses a new position in design space for the next generation; Tabu search calculates the fitness of both current generation and former generations to form memory structures then moves current solution to an improved solution.

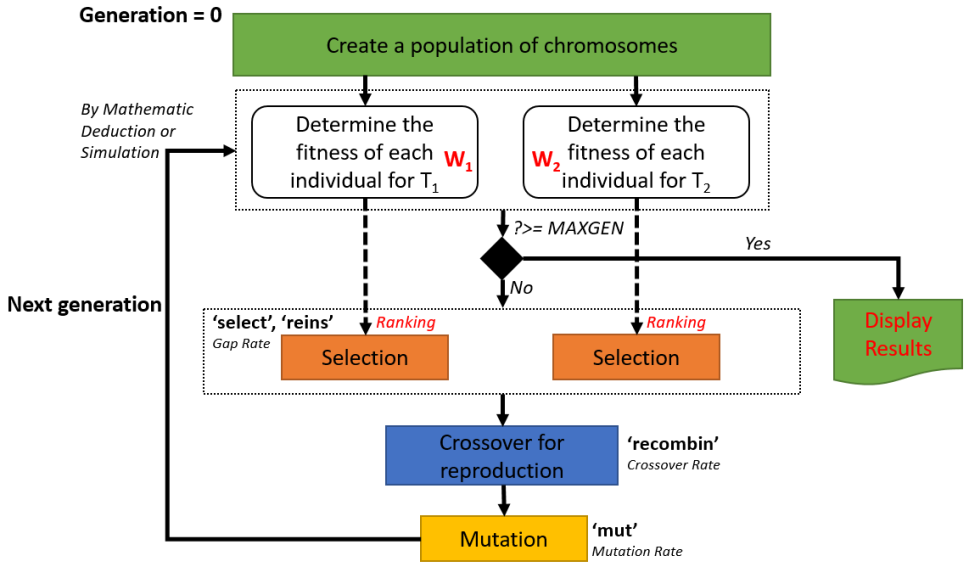


Figure 2.16 Genetic Algorithm procedure with two objectives (parallel selection).

2.4.2 Surrogate algorithms

The other category in 2SA, surrogate algorithm, builds a mathematical mapping model based on a set of sampled data (known as the training data) which should be collected and processed for specific problems. The well-trained surrogate model can effectively predict the desired outputs without running the original system(s) and being explicitly programmed. For the training of surrogate/ML models, only if the training performance is good and/or the predictions are close to a new data set, the model can be referred as a well-trained model and then used in following optimization.

During the training, also called a learning process, internal algorithm parameters are continually updated by using the errors between samples (raw data) and current predictions [89, 90]. Therefore, a mapping/surrogate model is established by a supervised learning process until when it can be used to replace the original system(s) to perform a task. That is the reason why it has been named “surrogate”, reflecting the replacement function [28]. As mentioned, the surrogate algorithm category here mainly refers to the supervised learning algorithms, does not include the unsupervised or reinforcement learning. The reason is that, in most cases of ML applications in the

electrical domain, the corresponding outputs (rather than policies) should be obtained/collected against the studied inputs in a certain design space.

Figure 2.17 shows the schematic diagrams of a feedforward NN and the mathematic operation in a certain neuron. Within a NN, a neuron is a mathematical function that model the functioning of a biological neuron [69]. As shown in Figure 2.17(a), a basic feedforward NN comprises an input layer (*x*, marked in red), one or more hidden layers (*h*, marked in green), and an output layer (*o*, marked in red). Weights and biases are omitted for simplicity. Each layer has one or several artificial neurons which receive and process signal(s) from predecessor neurons, after that neurons deliver the processed data to the next layer.

In a layer l (hidden layer or output layer), to calculate the output of a certain neuron n_i^l, the outputs of all the neurons p_j^{l-1} in Layer $l-1$ ($j = [1..N_{l-1}]$, N_{l-1} denotes the neuron number of Layer $l-1$) are multiplied with given weights ω_{ij}^l and then the bias b_i^l is added. The result is further processed through an activation (propagation) function f_σ that usually takes the form of a sigmoid function, $f_\sigma(A) = 1/(1+e^{-A})$, to generate the output p_i^l, as shown in Figure 2.17(b). This output then becomes one of the inputs for the next layer, $l+1$, and the same procedure is repeated to calculate the output of other neurons in layer l.

In Layer 1 (input layer), p_i^1 takes the form of inputs through the neuron n_i^1. On the other side, Layer L (output layer) typically uses the linear activation function to integrate signal(s) of Layer $L-1$ for the desired output data p_i^L. In summary, the complete data flow of NN can be described as follows:

- Layer 1 (input layer):

$$p_i^1 = x_i, i = 1,..,N_1 \tag{2.1}$$

  where x_i denotes the inputs of NN.

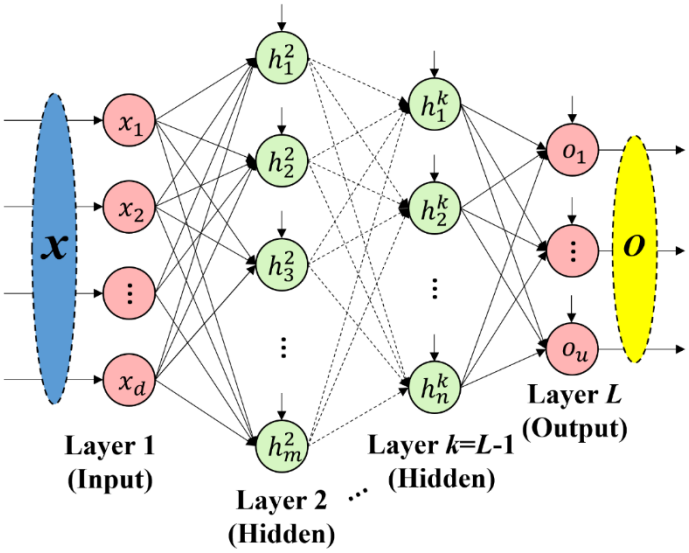
- Layers $l = 2,..L-1$ [hidden layer(s)]:

$$p_i^l = f_\sigma\left(\sum_{j=1}^{N_{l-1}} \omega_{ij}^l p_j^{l-1} + b_i^l\right), i = 1,..,N_l \tag{2.2}$$

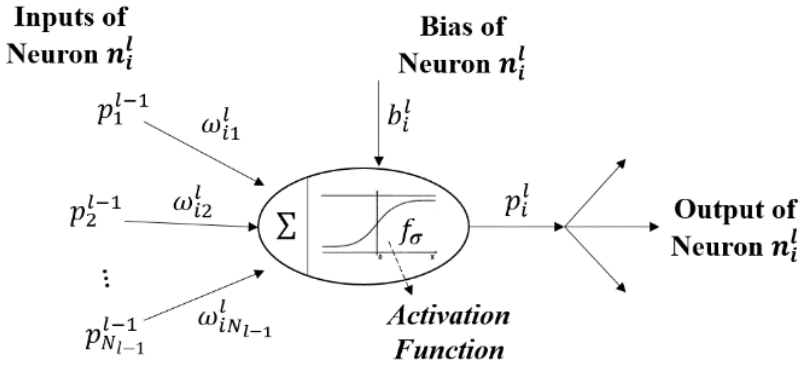
- Layer L (output layer):

$$o_i = \omega_i^L p_i^L, i = 1,..,N_L \tag{2.3}$$

where o_i represents the outputs of NN.



(a) A feedforward NN



(b) A certain neuron

Figure 2.17 Diagram of a feedforward NN and a neuron in feedforward NN.

The number of neurons in the input and output layers are determined by the samples and they are specific to the studied problem. The neuron number in the hidden layer(s) can be set an NN algorithm developer. Therefore, neuron number selection for different NN structures is flexible and this has been a persistent and hot topic in multi-discipline

studies [93, 102, 103]. Every neuron is a node which determines the input-output relationship of the signal. A NN can be trained to a nonlinear model using a proper data set. Such general nonlinear model can approximate any given input-output function with arbitrary precision [90]. More details of NN can be found in two case studies in this thesis: filter design and optimization (Section 3.5, Chapter 3); motor performance correction model (Section 5.2, Chapter 5).

Bayesian network, SVM and other supervised ML methods can also be trained and used in similar ways thus they should be categorized as surrogate algorithms. Besides, the surrogate models can be further divided into two modes as surrogate regression and surrogate classification whose output types are different. For example, SVM is a well-known classification learning technique which can give a discriminative classifier in ML formally defined by training a separating hyperplane with the maximum distance from the nearest training data [97]. NN is usually utilized for the ML regression problems due to its excellent non-linear generalization capability; however, it can also be applied into the classification learning through the NN pattern recognition, which limits all predictions into (0, 1) by the Softmax function [28, 63, 104]. The cases of two ML modes will be used in Chapters 3 & 4.

It is worthy to mention that all the ML applications in this thesis are following this general thinking: first collect data from the studied system, then train the dedicate surrogate ML models; after the training and the model validation (offline), the trained models can be applied in the studied system (online). However, on the other hand, ML algorithms (e.g. Neural Networks) can be directly applied online as the adaptive controllers, which means they should be trained in every sampling period and the internal parameters (e.g. weights and bias) can be totally different in every period. An example of control of single-phase grid-connected inverters using recurrent NN can be found in [105]. This kind of ML application will be a good direction of the future studies (will be discussed in Chapter 7).

2.4.3 Comparisons of two algorithm categories

Differences and connections of 2SA are concluded in Table 2.1. Search algorithm is to do the optimization while surrogate algorithm is used to train a data-based 'agent' mapping $\boldsymbol{x}$-to-$\boldsymbol{o}$ relations but cannot do optimization itself. But, one of the most important connections is that surrogate algorithm can be trained as an agent to effectively serve the optimization due to its computation efficiency [89, 90, 106]. Besides, the operation of search algorithm is a data changing process (both inputs and outputs) but for a surrogate model discussed here, these data usually stay unchanged during the training process. Last but not the least, in order to optimize the studied system using surrogate algorithms, an exhaustive method (or search algorithm) could be applied based on the trained surrogate(s) which can perform the specific task (giving dedicate predictions) very quickly [90], a PMSM optimization case will be given in Chapter 5.

Table 2.1 Differences and connections between search algorithm and surrogate algorithm

	Search algorithm	Surrogate algorithm
Purpose	Used for the optimization in a certain design space	Used to train a data-based 'agent' which maps specific relations but cannot do optimization by itself
Operation mechanism	Calculate the 'fitness' of outputs and determine new inputs according to the algorithm theory	Based on raw data, build a mathematical 'agent' model by updating algorithm parameters
Data in operation	Both input and output data are changing during optimization process	Raw or sample data usually stays unchanged and the model is trained offline
Model building	Can use the trained 'agent' to help find the search boundaries for search algorithm operation	Can use search algorithm to find the optimum parameters for a desired surrogate model
Interactive functions	Can utilize the trained 'agent' model to generate outputs for the corresponding inputs to serve the optimization	After training, search algorithms can help do the following optimization works in a certain design space

2.5 Chapter Summary

This chapter first reviewed the previous studies of different algorithms. These studies are mainly on two research domains of electrical drive systems: (1) design and optimization; (2) system control. In addition, the algorithms considered in the literatures were discussed. Based on that, a simple algorithm categorization, 2SA, was proposed to quickly distinguish all the commonly used algorithms. Two categories in 2SA (and their algorithm examples) were discussed separately. Finally, the connections and differences between two categories were given.

Chapter 3 Optimal Filter Design for Aircraft Electrical Drive Systems Using Search and Surrogate Algorithms

In this Chapter, the first optimization-based design case, an *LC* filter (input passive components in the aircraft electrical drive system), will be discussed using the proposed 2SA method. Other two subsystems (a DC-AC converter and an AC load) assume unchanged and the control strategy applied in the drive system is the finite control set model predictive control (FCS-MPC). It is noted that, for the AC side, both a PMSM and an *LC*-filtered linear load will be investigated where the AC side control strategy is slightly different. The optimal filter design for an electro-mechanical actuator (EMA) onboard aircraft will be firstly investigated and the corresponding optimization results from 2SA will be compared. Furthermore, with the help of the same filter design methodology and genetic algorithm (GA) tool, the optimal filter design for a linear-load drive system will be identified via the optimization Pareto set. Lastly, three filter prototypes (two in the Pareto front) are finally tested for the experimental validation.

3.1 Filter design method choosing

As shown in Figure 1.5, an electrical drive (a power converter connected to an electrical machine) extract steady power from the DC network. With a constant mechanical power output in steady states, these drives act as constant power loads (CPLs) when viewed from the DC bus. It is well-known that the incremental input impedance of these CPL is negative [20, 31] and the CPLs tend to destabilize the entire system due to their negative impedance. This chapter applies a DC *LC* filter in front of the DC-AC voltage source converter (VSC), as shown in Figure 3.1. L_1 is the filter inductance, R_1 is the

resistance of filter inductor coil, C_1 is the filter capacitance, R_2 is the resistance of the capacitor. After adding this DC filter, the DC-link capacitor voltage can be stabilized by the used FCS-MPC algorithm (meanwhile, the AC machines can be controlled by the same control strategy, see Section 3.3). Furthermore, the filter multi-objective design and optimization problem will be investigated in this chapter based on the filter design models. Even for non-constant power loads (for example actuators onboard MEA), the proposed filter design method is also feasible since this method is independent with the circuit topology, it only depends on the filter sizing model/data.

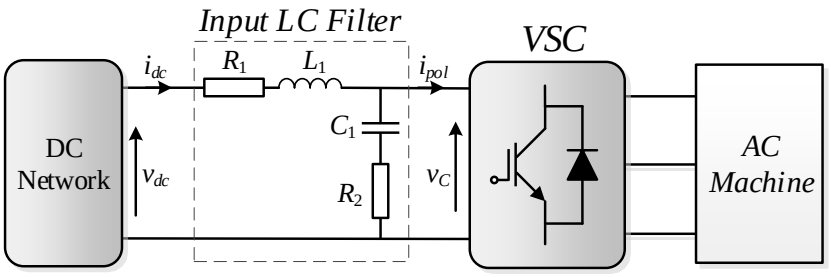


Figure 3.1 Input *LC* filter system for a DC-AC drive system.

As shown in Figure 3.2, a filter design can be generally divided into two approaches: inductance/capacitance (*L*/*C*) analysis and component-based design.

- For the analysis design approach, the filter principle equations can be analyzed to get a minimum allowable component value [34, 35, 107]. The design process is generally with a few steps: (1) Derive the voltage drop across the inductor (to-be-designed) to give the min inductance value ($L_{\min}$); (2) Derive the current flowing into the capacitor (to-be-designed) to give the min capacitance ($C_{\min}$); (3) Deduce the filter cut-off frequency according to the required harmonic attenuation, which can define a relation between *L* and *C* (subject to the attenuation capability). This kind of design approach is based on the circuit simplification, thus several assumptions have to be taken. For example, the grid voltage is assumed to be balanced and the system is with a unity power factor. In practice, this kind of assumption may not be feasible for the dc-bus-based

power system onboard MEA, because this bus usually needs to feed various loads under different flight stages.

- For the component-based design, the inductor/capacitor commercial data are utilized to give the relation between performance and design variables. Different commercial inductor cores can be given as a selection pool. The number of turns for inductor windings should be designed and the optimal core can be given by a multi-objective optimization. For capacitors, the commercial data can be directly referred in each capacitor series to establish the relations between the potential capacitor performance and capacitance. Therefore, this component-based design approach can link passive values to the filter performance based on the capability parameters of physical products.

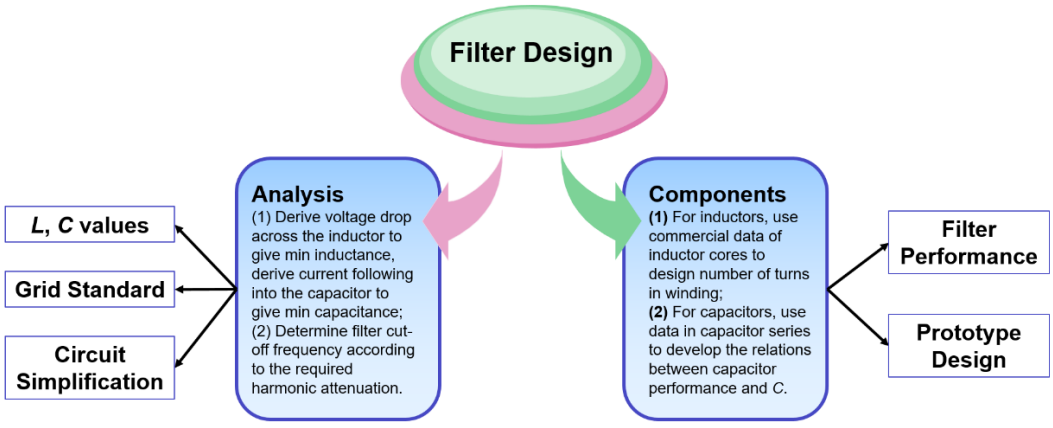


Figure 3.2 Two aspects of filter design.

The *LC* values of a filter can be easily and directly given by the filter analysis approach. Give two examples: For a PI controlled converter, the switching frequency is usually given as a parameter. Then, when designing an inductor, the voltage drop across this inductor, at where the maximum current ripple (ΔI_{pp}) occurs, can be derived. Based on that, the minimum L value can be obtained according to a predefined ΔI_{pp} and the switching frequency [34, 36]. On the other hand, if given a Total Harmonic Distortion (THD) upper limit and simply assume that $THD = \Delta I_{pp}/I_{pp}^{1}$ (where I_{pp}^{1} denotes the

fundamental peak-to-peak value of the mains current), the minimum product value of L and C in an LC filter can be derived by the filter cutoff frequency [34].

However, these equations in the analysis approach can only give extreme values of LC which do not guarantee the minimum mass and losses of the entire filter. To solve this issue, a component-based filter optimization design can be used [29, 34, 107]. More importantly, conventional filter design approaches can hardly be applied for the non-modulated FCS-MPC due to its variable switching frequency. This time-varying switching frequency significantly complicates the filter design process, compared to standard design procedures for filters of converters operated with linear control systems. Therefore, to identify a proper way to design a LC filter for a power converter with a time-varying switching frequency is the main motivation for the study performed in this chapter.

Component-based filter design methods commonly incorporate sizing models for inductors/capacitors with regards to physical parameters (e.g. weight, geometry), inner resistance and thermal performance. Different from conventional design and optimization methods in [12, 34-36, 107], the component-based design method directly treats LC as design variables (DVs) in an optimization and, the control performance (e.g. current ripple, THD) is carefully considered as the constraints of optimization. Based on that, desired LC values can be obtained directly from the searching process. Another advantage of this method is that the internal filter resistance and its damping effect [35] are smoothly reflected in the simulation system which is embedded in the integrated sizing process.

In summary, this chapter will present an automated and integrated optimal filter design method with embedded circuit simulations. The following section will introduce LC filter sizing models and the FCS-MPC strategy which serve as the basis of the filter design and optimization problem in EMA.

3.2 Filter sizing models

Based on the developed algorithm categorization in Section 2.4, this thesis will use a GA tool and establish a generic surrogate structure to optimize the *LC* filter for an EMA. However, before discussing the operation of 2SA models, it is essential to introduce the *LC* filter sizing models and its control strategy. For an inductor design, the basic architecture should be first confirmed, i.e. inductor core type (e.g. toroidal, E-type, etc.) and winding area on the core. For inductors within the LC filter, it is assumed that distributed-gap cores are used because they are small in size and thus suitable for aerospace applications [12, 28]. For DC-link capacitors, film capacitors are selected due to their excellent temperature characteristics and current ripple tolerance [28, 108]. The sizing models of the inductor and capacitor will be discussed in the following sections.

3.2.1 Inductor Sizing

Toroidal powder cores made by magnetic material MPP60 from Magnetics Inc [109] are employed here to form an initial selection pool because they have good AC and DC magnetization properties and provide good achievable inductance per volume. One example core is shown below whose part number is *55180A2*. Its outer diameter is 4.65 *mm*, inner diameter is 2.36 *mm*, and height is 2.54 *mm*. Other parameters (for example window area, path length, inductance factor) can also be obtained from the commercial manual of powder cores [109].

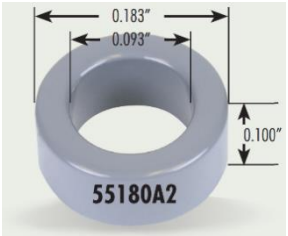


Figure 3.3 One example of toroidal powder cores from Magnetics Inc [109].

The inductor sizing model essentially utilizes the Maxwell-Ampere's law Eq. (3.1) to calculate conductor cross-sectional area with current RMS value [40]:

$$\oint H \cdot dl = \int_A J \cdot dA = \sum i(t) = \Theta \tag{3.1}$$

where H (A/m) denotes the magnetic field strength along an enclosed path with length l (m), J (A/m^2) represents the current density of current i (A) flowing through a cross-sectional area of A (m^2) and Θ (A) is the current linkage.

As shown in Figure 3.4, given a inductance value L_1, there are generally five constraints in this sizing model: maximum number of turns $N_{t,max}$, field intensity H_{max} (A/m), maximum permeability drop γ_{max} (H/m), minimum cross-section area $A_{c,min}$ (m^2) and temperature rise T_{lim} (℃). In general, only toroidal powder cores that meet all these requirements will be reserved in the selection pool. At the end, the smallest core is chosen from the remaining available cores. Based on that, the final comprehensive inductor design for L_1 as well as its mass and loss can be derived as output of the process. The final inductor design output includes number of turns, wire diameter, and the selected toroidal core for the use of future manufacture. If there is no core left in the selection pool, the inductor sizing for L_1 fails.

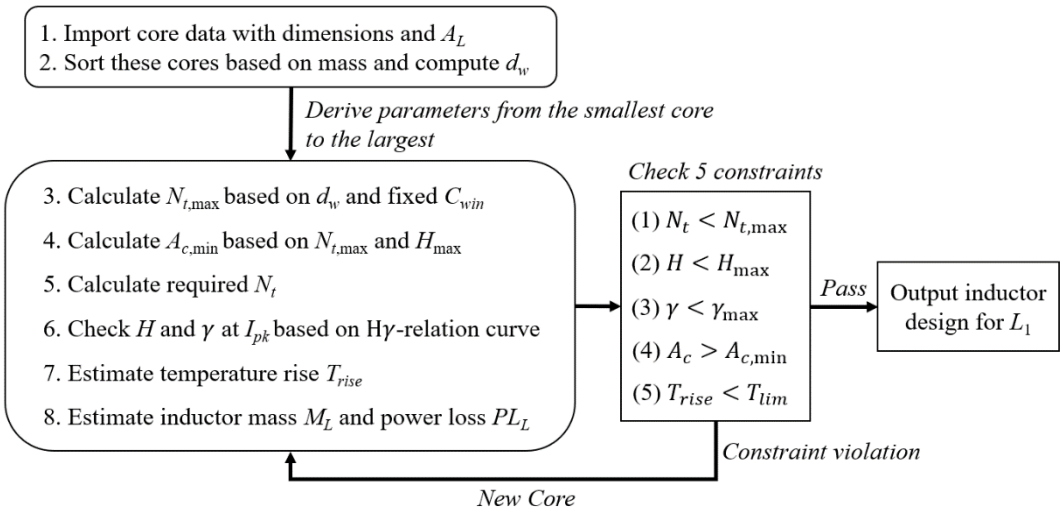


Figure 3.4 Inductor Sizing Model

As shown in Figure 3.4, at the beginning of the sizing inductors procedure, discrete core sizes and data from Magnetics Inc. are imported to a selection pool. The inductance is designed according to the maximum energy storage requirements. The Root Mean Square (RMS) current I_L and peak current I_{pk} flowing through the inductor should also be considered. Based on a choice of maximum RMS current density J_{max} and insulation thickness h_{ins}, the required coil diameter can be computed as [12, 109]:

$$d_w = 2\sqrt{\frac{I_L}{\pi J_{\max}}} + 2h_{ins} \cdot \tag{3.2}$$

With this coil diameter, the maximum number of turns $N_{t,max}$ can be calculated as a geometrical constraint for a specific core selected, using information of the core's internal diameter, minimum spacing between conductors and the circumference (C_{win}) of its inner cycle of each core:

$$N_{t,\max} = \frac{C_{win}}{\gamma_{sp} d_w} = \frac{\pi\left(d_{i,core} - d_w\right)}{\gamma_{sp} d_w} \tag{3.3}$$

where $d_{i,core}$ is the inner core diameter and γ_{sp} is the wire spacing ratio (if γ_{sp} is 1, no spacing is allocated between wires). A simplified geometrical model is depicted in Figure 3.5, where $d_{o,core}$ denotes the outer core diameter.

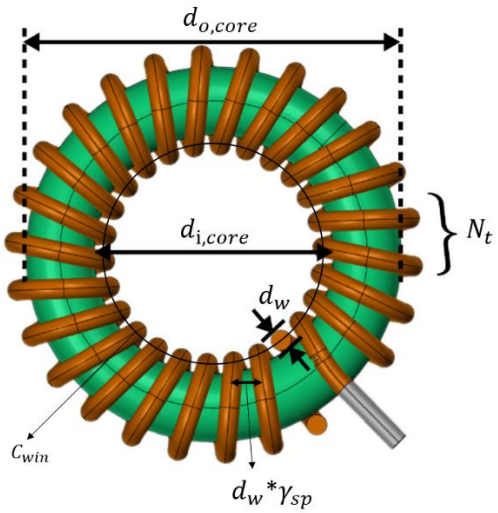


Figure 3.5 Simplified geometrical model of a distributed-gap toroidal core inductor [12].

Then, the minimum core cross-section area $A_{c,min}$ can be given by $N_{t,max}$, magnetic permeability μ (with drop γ), predefined H_{max}, and I_L at a desired inductance L_1, as:

$$A_{c,\min} = \frac{\Theta}{B_{\max}} = \frac{I_L L_1}{\gamma \mu H_{\max} N_{t,\max}}. \tag{3.4}$$

After that, the required number of turns N_t with the core's nominal inductance factor A_L for L_1 can be calculated as:

$$N_t = \sqrt{\frac{L_1}{A_L}}. \tag{3.5}$$

where A_L denotes the inductance factor for each core, which is obtained from the core data sheet [109]. As shown in Figure 3.1, L_1 denotes the inductance of the studied *LC* input filter. In this chapter, this inductance will be regarded as a design variable in the filter optimization thus, a design range will be assumed for L_1 and the L_1 values of design candidates will be determined by their corresponding objective function values and the search algorithm used in the filter optimization, see Section 3.4 for more details.

In addition, for magnetic constraints of the toroidal inductor, as magnetic force *H* within the toroidal core increases, permeability μ drops due to saturation of the material. This relationship between permeability drop γ and *H* is specific to the core material, and it can be found from the material datasheet. The resulting *H* in the core is obtained from

$$H = N_t * I_{pk}/l_e \tag{3.6}$$

where l_e is the magnetic path length, based on that, the permeability drop γ of the potential inductor design can be interpolated from the material's $\gamma(H)$ relationship, it should be smaller than γ_{max} (0.8 at this work). Regarding the core temperature rise, it can be estimated using Eq. (3.7) [109] and the max limitation of rise is set as 55 ℃:

$$T_{rise} = \left(\frac{PL_L}{A_{core,s}} \right)^{0.833} \tag{3.7}$$

where PL_L is the total inductor DC power loss, $A_{core,s}$ is the core surface area. Noting that, 55 ℃ is just a parameter for temperature rise which can be changed for specific applications, which may yield different feasible design points; However, the whole filter design method would be totally the same.

After checking the constraints one by one, the smallest core is chosen from the remaining available cores in the selection pool, the mass and loss of this inductor design can hence be given for each design candidate during the optimization.

3.2.2 Capacitor Sizing Model

The capacitors are selected from the EPCOS B2562* MKP DC film capacitors series [108] which have excellent temperature characteristics and current ripples tolerance. This series have a rated voltage of 700 V and the dissipation factor is specified as (at 100 *Hz*):

$$\tan\delta = \begin{cases} 1.2\times10^{-3}, C_1 < 450\mu\text{F} \\ 1.5\times10^{-3}, 450\mu\text{F} \le C_1 \le 800\mu\text{F} \\ 2.0\times10^{-3}, C_1 > 800\mu\text{F} \end{cases}, \tag{3.8}$$

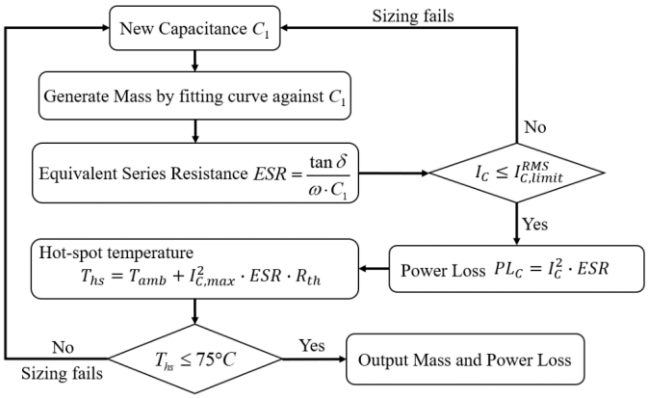
which is the ratio between the Equivalent Series Resistance (*ESR*) and the capacitive reactance of a capacitor at a specific sinusoidal alternating voltage, frequency and temperature. δ is the capacitor loss angle. Based on the dissipation factor, the ESR can be derived as

$$ESR = \frac{tan\delta}{\omega \cdot C_1} \tag{3.9}$$

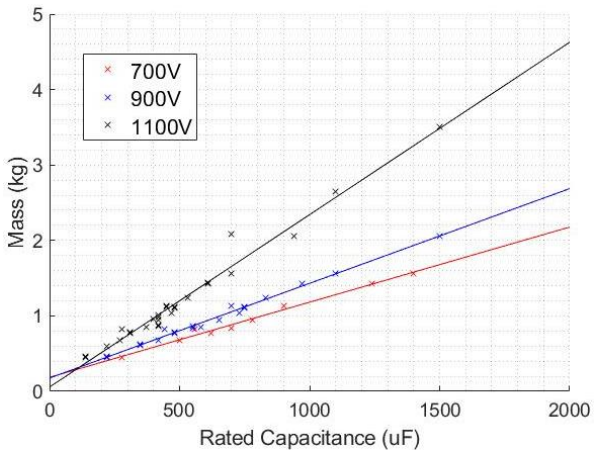
The ESR is used to represent the active power consumed by its internal resistor in a real capacitor.

Figure 3.6(a) shows the loss estimation process with two constraints of current and temperature. First, the RMS current I_C of capacitor should be smaller than the current limitation I_{max}^{RMS} given by manufacture. Then, capacitor DC loss PL_C can be computed using *ESR* and I_C with

$$PL_C = I_C^2 ESR \tag{3.10}$$



(a) Power loss calculation



(b) Mass fitting for B2562* capacitors

Figure 3.6 Capacitor sizing function

In addition, the thermal model in it consists of only one thermal resistance (R_{th}, from manufacture) [34, 108] and the ambient temperature T_{amb} is assumed to be 20 ℃. Temperature rise is simply computed by max capacitor current $I_{C,max}$, *ESR* and R_{th}. Then, the final hot-spot temperature T_{hs} can be given as

$$T_{hs} = T_{amb} + T_{C,rise} = T_{amb} + I_{C,max}^2 \cdot ESR \cdot R_{th} \quad (3.11)$$

If $T_{C,rise}$ is smaller than 55 ℃, the capacitor sizing succeeds, its mass and loss can be finally output. Noting that, both 55 and 20 ℃ are parameters for capacitor thermal performance which can be changed for specific applications (for example aerospace); However, the filter design method will not be changed for different settings.

Using a linearly fitting the capacitance value and mass of the according component category book, the capacitance density of this series to deduce the capacitors' weight can be approximated. The mass fitting of capacitance for three sets of capacitors in B2562* series are shown in Figure 3.6(b). The fitting performance of capacitor with three different rated voltages (700V, 900V and 1100V) are with good linear relations between mass and capacitance.

3.2.3 Integrated Sizing for Filter Optimization:

As discussed in two sizing functions above, currents flowing into the inductors and capacitors should be given before deriving their inner resistance or giving judgement for the constraint violation. However, the paradox is that, only after knowing the internal resistance values of capacitors and inductors, the current can be generated by the circuit simulation. While the values of capacitors and inductors are actually to be determined through the design process. To this end, an integrated filter design approach is proposed to determine the inner resistance by using above two sizing models and the circuit simulation, as shown in Figure 3.7.

At the beginning of our proposed design process, inner resistances of inductor and capacitor (R_1 and R_2 respectively) are both assumed with an initial value of 1 $m\Omega$, such that the circuit system can be simulated for each set of design variables (including L and C). Then, the capacitor current signal (from the offline simulation in Simulink) is processed and feedforward to the capacitor sizing model for the *ESR* calculation. After that, a simulation iteration is exercised to size the inductor for the given L value. As

mentioned, commercial inductor cores (data sets) are given as a selection pool. For every step in the simulation loop, the inductor sizing model (Figure 3.4) should be run and then generate the smallest feasible inductor design (if there is at least one core feasible). The internal resistance (R_1) of this feasible design can then be given with the selected inductor using its corresponding data from manufacturers. Obviously, the inductor current signal from each-time simulation should be processed and feedforward to the inductor sizing model for checking constraints. The simulation loop will terminate when R_1 is converged (the error threshold is set as 5%). If no convergence after N times of simulation, this data point is not feasible.

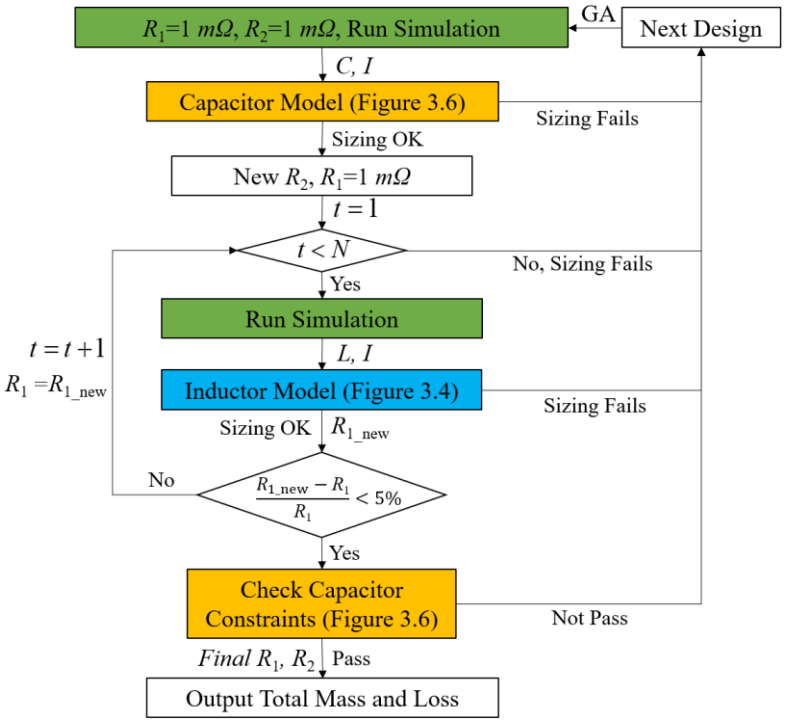


Figure 3.7 Integrated sizing for filter optimization using relatively independent inductor model and capacitor model.

The reason of adding a simulation loop for inductor sizing is that, with a new R_1 input, the simulation will output a new I_L. As discussed in Section 3.2.1, I_L is a critical input for the inductor sizing and, with different I_L values, different toroidal cores could be finally chosen by the sizing function. Therefore, a convergence of R_1 is needed to

make sure the final inductor design is suitable and accurate for the studied drive system. It is noted that, during the simulation loop, *ESR* of capacitor design stays unchanged. Moreover, if the inductor sizing fails after any one simulation in the loop, this data point is not feasible.

Finally, the capacitor model is utilized again to check the capacitor constraint compliance and this is followed by the output of two objective values (total mass and PL of the desired *LC* filter). It can be seen that, the capacitor sizing model (orange block) runs before and after a simulation loop. The inductor sizing model (blue block) is embedded into the simulation loop. The simulation step is marked in green.

3.3 Control algorithm in the actuation system

Figure 3.8 shows the schematic of the studied EMA onboard aircraft. v_s is an ideal DC source, L_0 and R_0 denote the equivalent impedance of transition cables in DC microgrid. On the AC side, machine winding resistance R_m (with a long cable), motor inductance L_m and back electromotive force (EMF) e_0 are utilized to represent the PMSM. The control strategy FCS-MPC is applied to the two-level converter. This thesis will only focus on the conventional non-modulated FCS-MPC based approach [20], as follows.

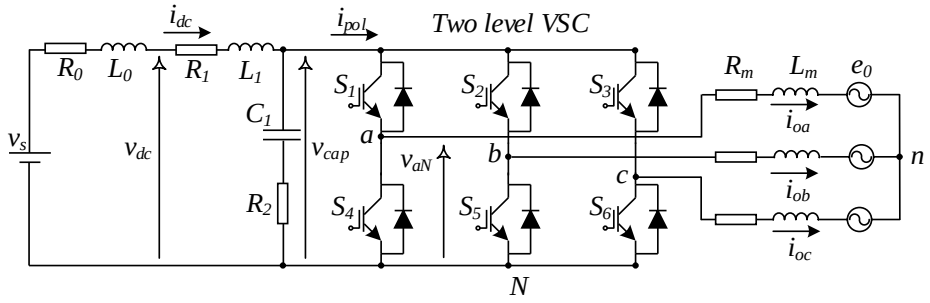


Figure 3.8 EMA case study: Two-level VSC fed from a DC grid via *LC* filter and supplying an AC motor

In the studied actuator system, FCS-MPC is used to regulate both the DC-link voltage v_C and the AC motor current i_0. Regarding the derivation of FCS-MPC, a

stationary α-β reference frame is applied for VSC modelling. Thus, all the three-phase variables x_a, x_b and x_c, are transformed into a corresponding α-β frame by applying an amplitude-invariant Clarke transformation [110].

In two-level converter switching functions, S_a, S_b, S_c are representing the gate signals (0-1). There are totally 8 voltage vectors with 6 active vectors and 2 zero vectors. A common mode voltage drop that results in reduction of voltage across the filter given as:

$$v_{nN} = \frac{v_{aN} + v_{bN} + v_{cN}}{3} \tag{3.12}$$

The voltages across the AC motor can thus be given as

$$\begin{cases} v_{an} = v_{aN} - v_{nN} = S_a \cdot v_C - v_{nN} \\ v_{bn} = v_{bN} - v_{nN} = S_b \cdot v_C - v_{nN} \\ v_{cn} = v_{cN} - v_{nN} = S_c \cdot v_C - v_{nN} \end{cases} \tag{3.13}$$

With v_{an}, v_{bn} and v_{cn}, the 8 possible voltage combinations of the inductor voltage on load side can be given by using the Clarke transformation. These voltage vectors, represented as $\bar{v}_i = v_{i\alpha} + jv_{i\beta}$, can be found in Table 3.1. Different from the operation of FCS-MPC in [84, 89], this study deals with an AC motor rather than linear loads thus FCS-MPC regulates the AC current instead of the AC capacitor voltage.

The 8 possible voltage vectors can be applied to a motor with resistance R_m, motor inductance L_m and back-EMF e_0. The equations that describe the dynamics of the motor current based on the voltage vector v_i are given as follows:

$$L_m \frac{d\bar{i}_0}{dt} = \bar{v}_i - R_m \bar{i}_0 - \bar{e}_0 \tag{3.14}$$

Apart from relations that describe the motor side, dynamics on the DC side should also be considered as shown in the Figure 3.9. The inductance L_0 in series with the resistance R_0 represents the impedance of DC-network transition cables. The *LC* filter is with the

inductance L_1 and the capacitance C_1 with inner resistances R_1 and R_2 respectively. The DC capacitor dynamics is expressed as

$$C_1 \frac{dv_{cap}}{dt} = i_{dc} - i_{pol} \tag{3.15}$$

where the current i_{dc} is the current flowing through L_1, i_{pol} is the current flowing into the VSC. It can be synthesized from the load currents and the gating signals, as:

$$i_{pol} = S_a i_{oa} + S_b i_{ob} + S_c i_{oc} \tag{3.16}$$

The equations above define the state-space model of a two-level three-phase VSC with dynamics on both AC and DC sides. The zero-order hold discretization method is used for the predictive control model which can ensure the consistency of discrete time and continuous time at the sampling instants.

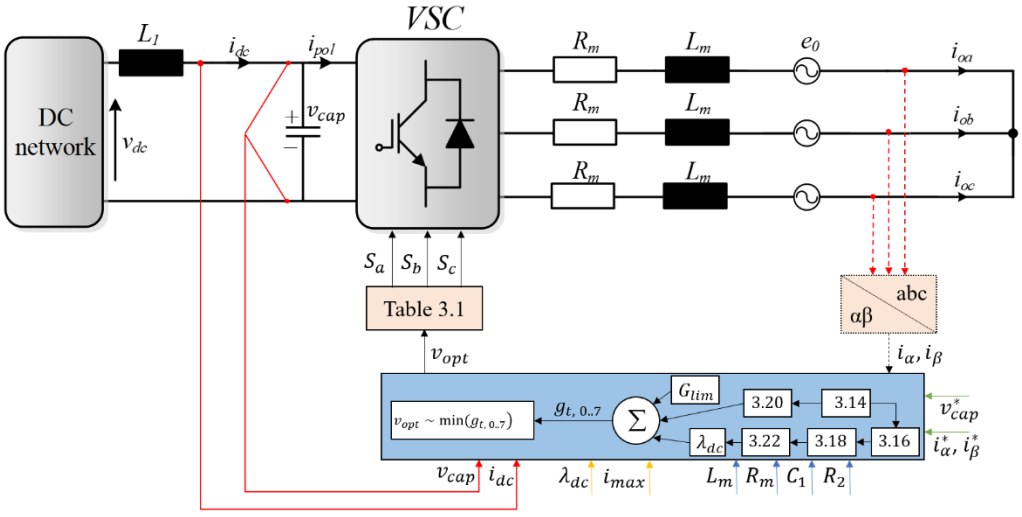


Figure 3.9 Complete implementation of stabilization approach for motor application

The simulation system was built in MATLAB/Simulink where a predictive controller makes contribution to giving the gating signals to be applied at the next sampling instant. The back-EMF v_e estimation of the load and the prediction of AC current i_0 and capacitor voltage v_{cap} should be done in a sampling period. The back-EMF estimation (at k instant) is using the differentiation of Eq. (3.14) while this equation also

predicts the $k+1$ current using all 8 voltage vectors v_i to give candidate value for cost function (CF). Then, for the v_{cap} derivative in Eq. (3.15), the discretization can be made by approximating the derivative as the difference over one sampling interval:

$$\frac{dv_{cap}}{dt} = \frac{v_{cap}(k+1) - v_{cap}(k)}{T_s} \tag{3.17}$$

Based on that, the capacitor voltage can be estimated by the state of change of this capacitor during one sampling period using an averaging operation [20]:

$$v_{cap}(k+1) = v_{cap}(k) + (i_{dc} - \frac{i_{pol,i} + i_{pol,f}}{2}) \cdot (\frac{T_s}{C_1} + R_2) \tag{3.18}$$

where $i_{pol,i}$ and $i_{pol,f}$ are the initial and final currents flowing into the VSC during the following time step, respectively. Compared to (9) from [111], where only $i_{pol,f}$ was taken into account, by doing the averaging operation, Eq. (3.18) allows a better estimation of the energy taken out of the dc link during each sampling time and consequently a more accurate prediction of the capacitor voltage [20]. The same as currents at $k+1$ instant, v_{cap} at $k+1$ instant should be estimated under all potential v_i for CF of which the following contents will give details.

Noting that, $\mathrm{v_{cap}}$ can be shaped by actively controlling the active-front end interface or power conditioning modules, in the DC network. Among the strategy candidates, droop control provides additional damping effect and it also allows power sharing between multiple converters at the same time [23]. However, the active-front end interface is not available in numerous industrial applications of DC microgrids, in which only unidirectional power flow is required [20].

As shown in the blue block of Figure 3.9, the FCS-MPC algorithm is ended with the minimization of CF which determines the optimal voltage vector v_{opt} to be applied at the next instant. The CF consists of the errors between the predicted voltage/current values and their references; further, the current limiting term G_{lim} is also included:

$$G_{lim} = \begin{cases} 0, if \left|\overline{i_o}\right| < i_{max} \\ \infty, if \left|\overline{i_o}\right| \geq i_{max} \end{cases} \tag{3.19}$$

The control term for AC current regulation is given in a conventional way, as:

$$g_{con} = (i_{o\alpha}^{*} - i_{o\alpha})^2 + (i_{o\beta}^{*} - i_{o\beta})^2 \tag{3.20}$$

Lastly, both two control terms are integrated into a total cost function:

$$g_t = g_{con} + \lambda_{dc} g_C + G_{lim} \tag{3.21}$$

where

$$g_C = (v_{cap}^{*} - v_{cap})^2 \tag{3.22}$$

Here, λ_{dc} is the weighting factor of DC voltage control term while the weight of AC current in FCS-MPC fitness function stays 1 in this study. As this work mainly focuses on the control preference between v_{cap} and i_0, other control terms (e.g. derivative of the voltage reference, switching penalization) are not considered here thus there is only one WF in the cost function g_t. However, the two terms of derivative and switching penalization should be considered if the issues of stability and switching loss need to be further analyzed [20]. In contrast to [20] and [89], λ_{dc} in this study is set as a design variable in the optimization rather than using a pre-defined adaptive function. It is supervised by GA objectives but, more importantly, the feasible design points must provide good control performance. Only the potential design that meets various performance constraints in GA optimization can be used to give a feasible filter design.

The following two sections will introduce the application of 2SA into the filter optimization work after the description of the multi-objective optimization problem (MOOP). One is to use GA as the search algorithm while the other is to use two different ML algorithms as the surrogate algorithm in 2SA.

Table 3.1 Voltage vectors used in 2-level converter

$[S_a, S_b, S_c]$	Voltage vector $\overline{v}_i$	$[x_a, x_b, x_c]$
[0, 0, 0]	$\overline{v}_0 = 0$	(0, 0, 0)
[1, 0, 0]	$\overline{v}_1 = \frac{2}{3}v_{cap}$	$(\frac{2}{3}, -\frac{1}{3}, -\frac{1}{3})v_{cap}$
[1, 1, 0]	$\overline{v}_2 = \frac{1}{3}v_{cap} + j\frac{\sqrt{3}}{3}v_{cap}$	$(\frac{1}{3}, -\frac{1}{3}, -\frac{2}{3})v_{cap}$
[0, 1, 0]	$\overline{v}_3 = -\frac{1}{3}v_{cap} + j\frac{\sqrt{3}}{3}v_{cap}$	$(-\frac{1}{3}, \frac{2}{3}, -\frac{1}{3})v_{cap}$
[0, 1, 1]	$\overline{v}_4 = -\frac{2}{3}v_{cap}$	$(-\frac{2}{3}, \frac{1}{3}, \frac{1}{3})v_{cap}$
[0, 0, 1]	$\overline{v}_5 = -\frac{1}{3}v_{cap} - j\frac{\sqrt{3}}{3}v_{cap}$	$(-\frac{1}{3}, -\frac{1}{3}, \frac{2}{3})v_{cap}$
[1, 0, 1]	$\overline{v}_6 = \frac{1}{3}v_{cap} - j\frac{\sqrt{3}}{3}v_{cap}$	$(\frac{1}{3}, -\frac{2}{3}, \frac{1}{3})v_{cap}$
[1, 1, 1]	$\overline{v}_7 = 0$	(0, 0, 0)

3.4 Search algorithm and optimization results

3.4.1 Search algorithm – Genetic Algorithm (GA)

To do this filter multi-objective optimization work, genetic algorithm (GA), one of the search algorithms, is employed. As discussed in Section 2.3.1, GA is a population-based evolutionary algorithm. It is characterized by a population of solutions candidates that evolve towards the optimal trade-offs.

It is well-known that, for an optimization work, the design variables, objectives, and constraints should be predefined before searching. There are three design variables: L_1, C_1 and λ_{dc}. The ranges of three design variables are [1, 100] μH, [1, 2000] μF, [0.1, 10], respectively. Two objectives of optimization are Mass *M* and Power loss *PL*. Apart from two objectives, the sizing models can estimate the filter temperature and volume as well, but this thesis only focuses on the mass *M* and the power loss *PL*.

Since this study considers two objectives, not one, the optimal results will be the trade-off solution between different objectives, which is known as the Pareto optimal solution. Pareto optimal solution is defined as a set of 'non-inferior' solutions in the

objective space defining a boundary beyond which none of the objectives can be improved without sacrificing at least one of the other objectives [112]. In general, most of feasible designs generated during the optimization would be inferior. Pareto optimal solution is a set of feasible designs balancing all the objectives while every design of them should be better in at least one objective. Obviously, for the two-objective study, the designs in Pareto optimal solution will form a curve in the 2-dimensional objective space, which is usually marked as Pareto front. Pareto-front results of the filter MOOP for EMA will be given in Section 3.4.2. In addition, after generating the Pareto front results, an integrated index of objectives (marked as *r*) should be used to determine the best design point of the studied MOOP by integrating both objectives. In this filter optimization, the minimal distance from ideal objectives criterion will be used for that determination.

Regarding the considered optimization constraints, apart from the sizing constraints in afore-mentioned filter sizing models (Section 3.2, Chapter 3), there are four constraints for the DC link voltage (v_{cap} and two constraints for the AC currents. DC voltage constraints include the voltage transient envelop, the voltage spectrum upper limit, the maximum distortion factor (0.015), and the maximum voltage ripple (6 *V*). They are all from a military standard [32], the reason of selecting it is that a system using this stricter standard would also meet commercial rigs. However, the limits from other standards are also feasible for setting the optimization constraints. Differently, the AC current constraints include total harmonic distortion (THD) and tracking error of i_0, they should be manually set by trial-and-error to give limits for the control performance of FCS-MPC. The main reason is that the real feasible design points are changeable and not always guaranteed for different sizing/simulation models. Therefore, for specific applications, the important limits usually need be manually set by trial-and-error method based on the common standards. These six constraints are summarized in Table 3.2.

Table 3.2 Filter optimization constraints and parameters for EMA application

Constraints		Parameters	
Voltage transient	Figure 3.12(a)	Supply voltage	270 V
Distortion spectrum	Figure 3.12(b)	MG* transmission cable impedance	L_0: 1 μH, R_0: 1 $m\Omega$
Max. distortion factor	0.015	AC load	L_m: 2.5 μH, R_m: 10 Ω
Max voltage ripple	6 V	Peak e_0	100 V
Max tracking error	0.35 A	Sampling time	T_s: 25 μs
Max THD of AC current	6%	Reference	I_{ref}: 10 A, f_{ref} : 50 Hz

* MG: microgrid

Finally, multi-objective Genetic Algorithm is utilized as the search algorithm to find the best solution of this MOOP based on the filter sizing models. Figure 3.10 generally summarizes the MOOP of filter design of an aircraft actuator with three design variables and two objectives.

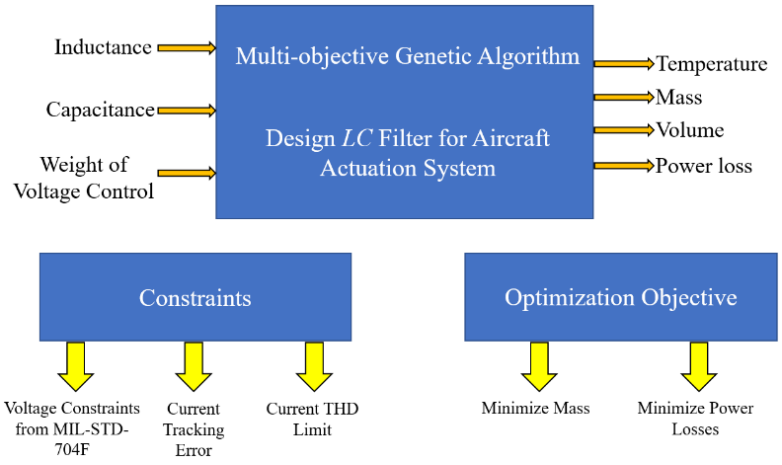


Figure 3.10 MOOP of *LC* filter in aircraft actuation system.

To start the *LC* filter design optimization process, after setting the ranges of design variables and the GA operator parameters (selection rate, crossover rate and mutation rate), the first generation of chromosomes needs be created randomly in the design space [78, 82, 99, 113]. A chromosome (individual) is characterized by a set of design variables via genes. Genes are joined into a string to form a chromosome. In a genetic algorithm, the set of genes of an individual is represented using a string, in terms of an

alphabet. Usually, binary values are used (string of 1s and 0s). This process is called encoding the genes in a chromosome.

In the commercial GA MATLAB tool [114], users can set different values to these parameters. The default values of these parameters within this toolbox are used for the sake of convenience. More specifically, the selection rate is 0.05, crossover fraction is 0.8, and mutation rate is 0.01. They can give suitable results in this study. However, if the feasible designs turn out to be few or the searched feasible range is too small, different operators can be tried for a new round of optimization. The selection operator is also called reproduction operator. As discussed in Section 2.4.1, this operator is used to identify how "good" a candidate is (according to the fitness value) and reduce the number of possible candidates. Therefore, this operator is choosing the candidates (5% in this study) that leads to the best results after the fitness evaluation. Crossover is a process where candidates of the last population are mated at random in the mating pool. A pair of *Children* will be generated by two candidates in this population. In this study, 80% of candidates will be operated by the crossover. Mutation is the occasional alteration of the genes in chromosome with a small probability (change from 0 to 1, or from 1 to 0). Therefore, mutation is a process of random walk through the coded parameter space. Its purpose is to ensure that important information contained within gene strings may not be lost prematurely. 1% of candidates will be operated by the mutation in each generation.

Regarding multiple objective optimization, other GA parameters (e.g. Objective weight [30, 82]) should also be confirmed before running the optimization process. There are two objectives and their weights are both set 1. That means two objectives, mass and power loss, have the same priority in the Pareto front generation. As will be discussed in the machine optimization study (Section 5.3), in the same minimizing optimization problem, increasing the weight of one objective will lead to the reduction of its objective value in the final best design point. The reason is that the increased

weight means more impact on determining the best design point, while in the same candidate pool, the designs with small values of this objective will more likely become the best design point. Furthermore, the population size and maximum generations of GA algorithm should also be predefined to confirm how long the optimization process would be. This study defined 300 individuals (population size) and 100 as the maximum generation number.

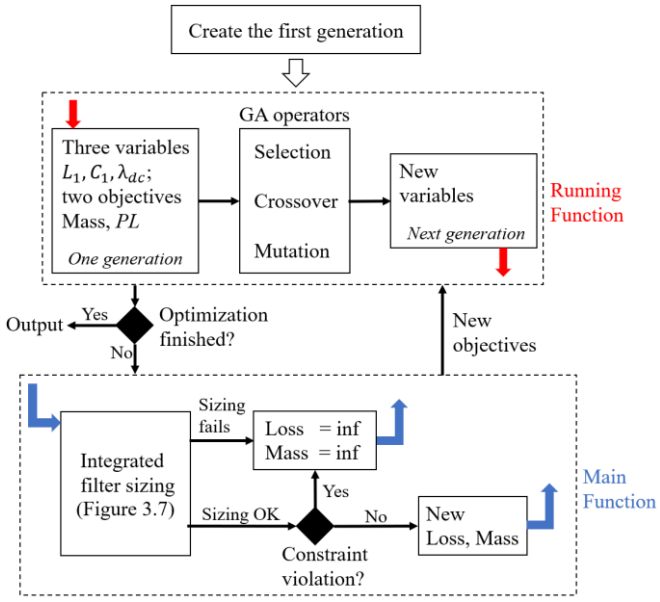
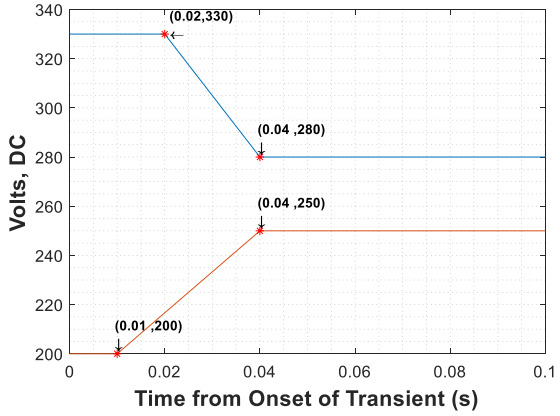


Figure 3.11 Filter optimization process based on the multi-objective GA

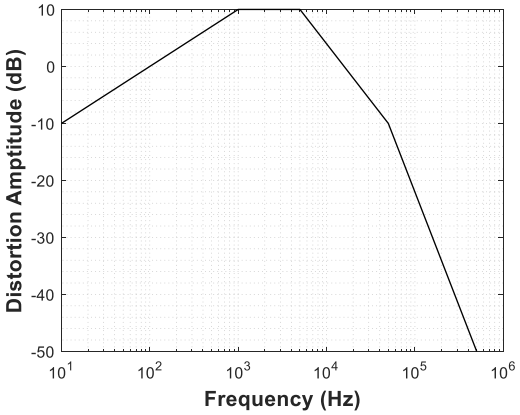
Fitness of individuals (*parents* and *children*) here consist of the values of two objectives whose weighting factors are set equally for GA operation. Figure 3.11 summarizes the overall optimization procedure. The upper rectangle part of Figure 3.11 shows the running function (marked in red) of "*gamultiobj*" in MATLAB [114]; the below rectangle depicts the main function (marked in red) of this study which is case-specific with the target MOOP. The main function refers to the basic models in the GA optimization which provides an understanding of the studied system and models it to take in a set of input design parameters and, correspondingly, calculate and output values of interests, i.e. filter power loss and mass. On the other hand, the running function stands for the GA optimization engine, which can provide insight into possible

designs that are optimal as defined with a set of objective values during a gene-based evolving process.

In the running function, a new generation of chromosomes (*children*) which maps the variables can be created by the three GA operators after ranking the fitness of the current generation (*parents*). If the iteration times does not exceed the pre-set maximum generation number, individuals (only know values of three variables) of *children* should go to the main function to get their objective values until reaching the maximum generation number (GA terminates).



(a) Envelope of normal voltage transient for 270 volts DC system



(b) Maximum distortion spectrum for 270 volts DC system

Figure 3.12 Two DC voltage constraints for the inverted based drive system [32].

In the main function, objectives and their fitness of the new generation (*children*) should be obtained by the integrated filter sizing model (Figure 3.7) for the next selection of chromosome population (*children of children*). However, every individual should be checked by constraints including sizing constraints (Figure 3.7) and signal

constraints (Table 3.2). If constraints violate a design, their objectives will be assigned the infinite values and this design will stay at the bottom of the fitness ranking in the new generation. If not, mass and power loss of this filter design can be given as its two objectives.

Two DC constraints are drawn in Figure 3.12, they are both from the military standard [32]. During the optimization, every DC voltage signal v_{cap} should be in the envelop of the voltage transient limits. In addition, the signal is analyzed by the fast Fourier transform and then compared with the spectrum upper limit.

3.4.2 Optimization results

This section reports the optimization results obtained from running the multi-objective GA with a population size of 300 and number of generations of 100, where 21734 designs are violated by constraints and 8566 designs are feasible. Pareto-optimal solutions of filter mass and loss set are generated after which the design distribution of the passive components (L_1, C_1) during the optimization process is shown with an integrated index of two objectives (r). Then, the design distribution of failure sizing points is extracted to show the distribution regularities and the violation situations of different constraints. Finally, the study on weighting factor (λ_{dc}) is exercised to demonstrate the trade-off function of this factor.

The optimal design results using GA are represented in Figure 3.13. Figure 3.13(a) shows the distributions of feasible design points which is marked by multi-colored circles. The color corresponds to the value of the integrated index (r) of two objectives, which is utilized to select one particular point as the optimal for each Pareto front. In contrast to the analytic hierarchy process (AHP) employed in [115], the criterion for this decision-making solution is the minimal distance from ideal objectives [30]:

$$Solution \equiv \min(r_i) \tag{3.23}$$

where

$$r_i = \sqrt{\left(\frac{M_i}{M_{\max}}\right)^2 + \left(\frac{PL_i}{PL_{\max}}\right)^2} \tag{3.24}$$

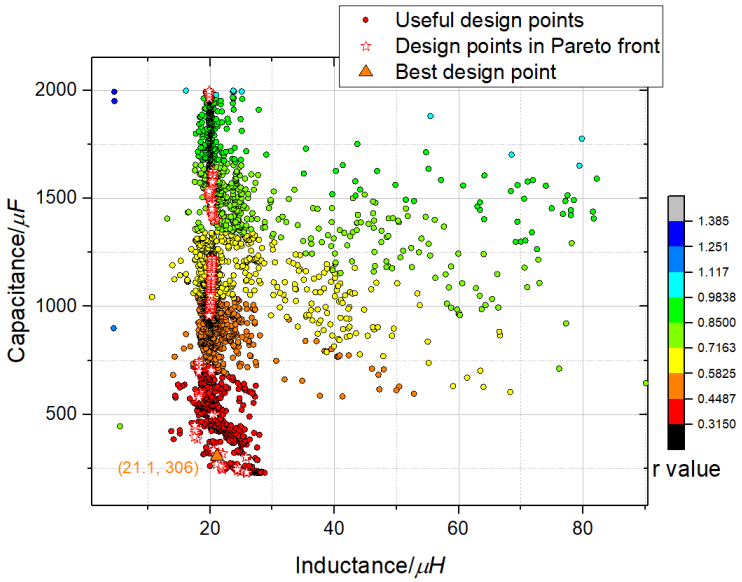
M_i: Mass of the i-th solution; M_{max}: Maximal mass; PL_i: Power losses of the i-th solution; PL_{max}: Maximal power losses. Obviously, in this equation, the weights of two objectives are the same. The reason of using this equation is that, it considers the max values of feasible designs thus the general data distribution of feasible range can be obtained. In Sections 5.3 & 5.4, Chapter 5, different weights of the Mass objective will be investigated for a motor optimization problem. Seen from the color distribution in Figure 3.13(a), there is a decreased trend of r value as the inductance and capacitance reduce. The design points in Pareto front are also marked by stars in the two subfigures of Figure 3.13. However, Pareto-front points are not all good solutions according to the r-value criterion. The best design point chosen by Eq. (3.23) is represented in Table 3.3 together with the inner resistance and the final r value.

Table 3.3 Best Filter Design Point in GA Optimization

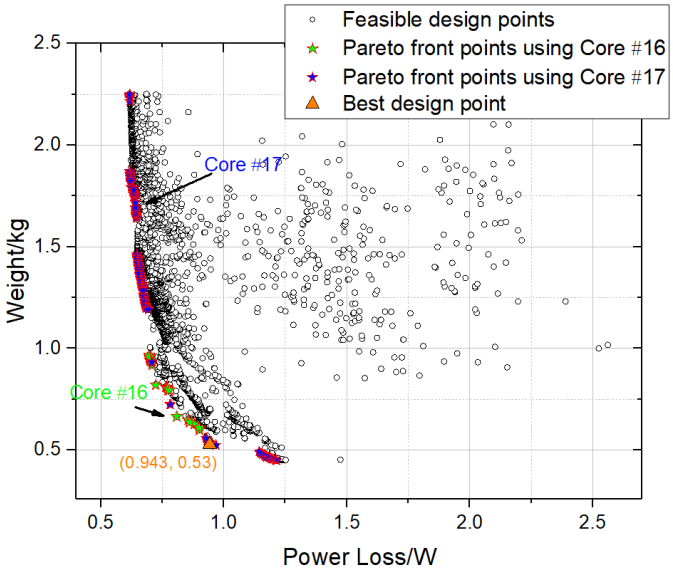
Inductor Inductance L_1	21.1 μH
Inductor resistance R_1	13.86 $m\Omega$
Capacitor capacitance C_1	306 μF
Capacitor resistance R_2	12.48 $m\Omega$
Weighting factor λ_{dc}	3.3586
Power loss PL	0.9431 W (max. 4.388 W)
Mass M	0.52753 kg (max. 2.891 kg)
r value	0.3168

Figure 3.13(b) represents the two objectives of useful design points where the Pareto-optimal design configuration of the studied filter is generated. For the motor optimization problems in [30], equality constraints are added for the solution distribution of Pareto front to ensure that there is no vacancy between the solutions in two different system levels. Conversely, in this thesis, the final Pareto solution is not processed by equality constraints; therefore, there are data gaps among the Pareto-front points. It is noted that the Pareto-front solutions use two different cores (No. 16 and No. 17 in [109]) for inductor sizing which also contributes to their big weight differences.

Another reason of data gaps is from the various GA constraints which limit a lot of potential design points for the Pareto front.



(a) Design distribution of L_1 and C_1



(b) Feasible design points and the Pareto front

Figure 3.13 Optimal design results of EMA based on multi-objective GA

3.5 Surrogate algorithm and optimization results

After the GA optimization, this study utilizes two ML methods to explore an alternative for the same MOOP. The motivation is mainly due to the ML/surrogate methods representing some advantages over search algorithms. Once trained a comprehensive

and well-verified surrogate model, it can be used efficiently in any future study of this MOOP (for example search the optimal design in a sub design space). There is no need to run the detailed simulation system model with search algorithm all over again, which can be very time-consuming.

Firstly, a generic ML structure is proposed which should be available for all optimization problems (no matter what the optimization algorithm is). Then, SVM is employed to address constraints as the constraint judgement model. Finally, NN is applied for the mapping surrogate from three design variables to two objectives in this filter MOOP.

3.5.1 Surrogate algorithm – NN & SVM

As shown in Figure 3.14, a generic ML optimization structure is utilized with two different ML models for the studied filter MOOP. The first model is for constraint violation, namely it is to find which data in design space would be feasible according to the afore-mentioned various constraints in Section 3.2. After that, feasible designs are regarded as the input of the second ML model to bridge the gap between three design variables and two objectives. NN is suggested for this mapping function. Both of two models are obtained from the supervised learning which means input-output data are required. In this thesis, 30300 sample data set (first generation + 100 generations, all sampling designs in the above GA optimization) is given for training the constraint violation model while only 8566 feasible designs are used to train the NN models.

SVM is a discriminative classifier formally defined by training a separating hyperplane [97] with the maximum distance from the nearest training data. Given the training data (the output is 1 for feasible designs and 0 for failure designs), the algorithm outputs an optimal hyperplane which can categorize new samples. The supervised classifier learning is implemented by “fitcsvm” function in MATLAB where “KernelFunction” is set as Gaussian radial basis function (“rbf”) which is computationally efficient [116]. “fitcsvm” function can train or cross-validate a support

vector machine model for one-class and two-class (binary) classification on a low-dimensional or moderate-dimensional predictor data set.

In this study, for the SVM training data, there are three input elements: L_1, C_1 and λ_{dc}, and one element in output: 1 for feasible designs and 0 for failure designs. Therefore, there are 8566 data points whose output is 1 and 21734 data points whose output is 0. After training, the trained function can predict a continuous number when given an input combination, if this number is larger than or equals 0.5, the predicted category should be feasible design; if smaller than 0.5, the predicted category should be failure design. Based on this processing, the predictions of all training data design points can be given and compared with the original data to get the prediction loss (or accuracy).

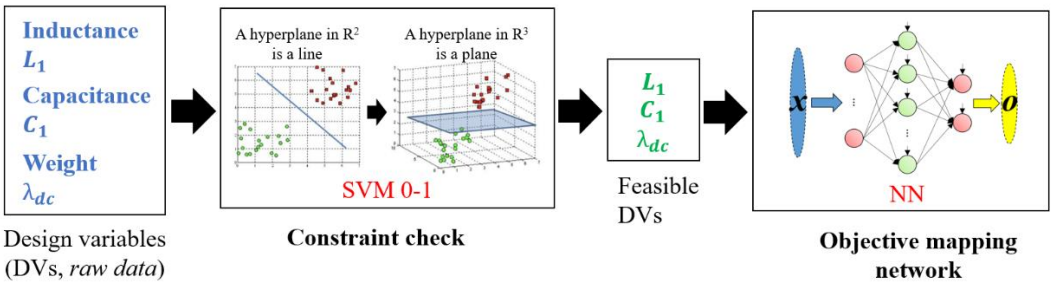


Figure 3.14 Structure of generic ML optimization

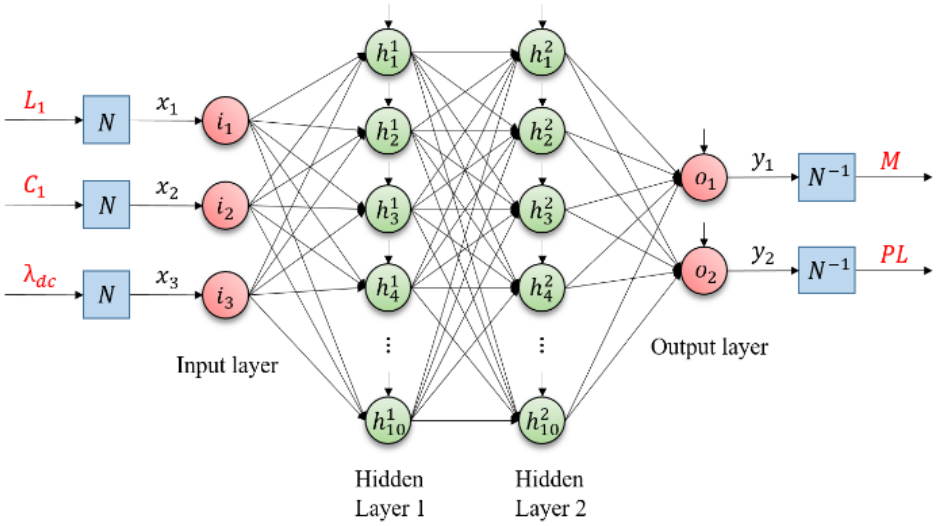
In this ML aided filter optimization study, it takes around 10 *mins* to train this SVM model in MATLAB on a standard computer which includes the cross-validation of training [93] and prediction-loss calculation of SVM. As a result, the class loss of cross-validation is 7.53%. Different ML algorithms for this constraint violation model were tried but no better results obtained. For example, ANN Fitting was first utilized for this classification training with the outputs processed to 0-1; However, the accuracy of ANNs with various structures is always below 89%. We also trained this classification problem using ANN Patter Recognition (PR) but the accuracy results cannot exceed 88%. Therefore, the trained SVM model is utilized as the final constraint violation model for the studied ML optimization problem.

After the SVM training, two different NNs are designed for the objective mapping, termed as NN_1 and NN_2, as shown in Figure 3.15. Both are using the 4-layer NN structure; however, for NN_1 the output layer is mapping two optimization objectives (*M* and *PL*) while NN_2 has only one element in the output layer, which is *r* in Eq. (3.23). The hidden-layer neural numbers of two networks are simply selected as 10. Other hidden-layer numbers are also feasible because, in this filter optimization study, the precision of the trained NNs turned out to be highly robust to the hidden-layer neural numbers. Due to the excellent NN regression capability, the empirical investigations have shown the negligible differences in NN precision when the neural number set was from 5 to 15. In order to compare the prediction performance of two different networks, except setting the same neuron numbers (10 neurons) in the hidden layers, both NN_1 and NN_2 are using the Bayesian Regularization training function and the deep learning toolbox in MATLAB. Furthermore, the maximum epochs for training is both set 500, learning rate is set as 0.1.

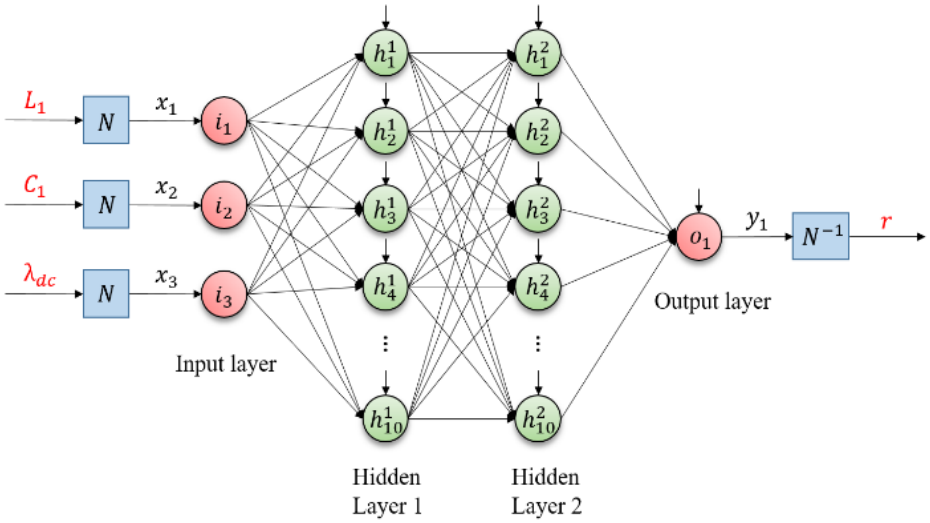
In the training process, the raw data set was randomly divided into three data sets: training data (70%), validation data (15%) and test data (15%). Training data is used to directly modify/update the weight and bias values in NN; validation data is used for validating the stopping condition and the ranking of net candidates; testing data is used to obtain unbiased estimates of non-training data [28, 117, 118]. In the following contents, the Root Mean Square Error (RMSE) of a trained NN is computed as the difference between all the raw data and the corresponding predicted data, which stands for the general NN training/mapping performance. After training, in this filter optimization topic, the predict performance of NN_1 for Mass (RMSE) is 0.003 *kg* and for power loss prediction, the error is 0.0197 *Watt*; on the other hand, the predict performance of NN_2 for *r* value (RMSE) is only 0.0019.

It is worth noting that, it is not necessarily using GA optimizaiton results to train ML models for the filter design, evenly sampling data are also feasible [119]. The reason

of using optimization results here is mainly to compare the performance of ML with the optimization algorithms.



(a) Mapping two objectives, NN_1



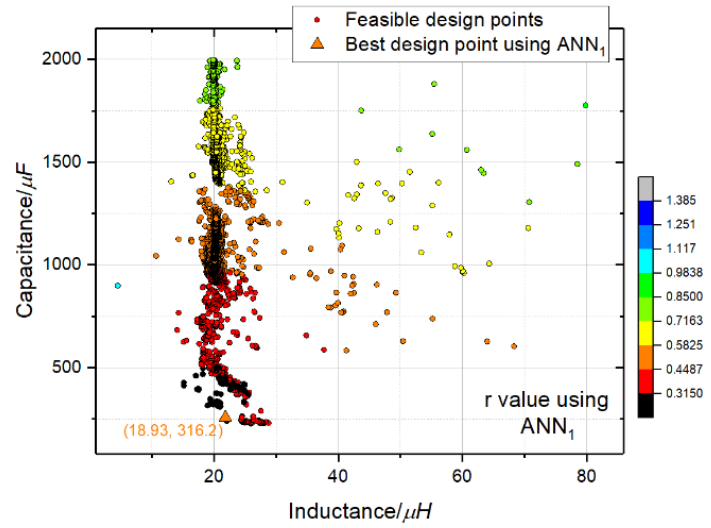
(b) Mapping one objective, NN_2

Figure 3.15 Deployment of the NN surrogate models for filter MOOP with normalization functions at the inputs and de-normalization functions at the output. Weights and biases are omitted for simplicity.

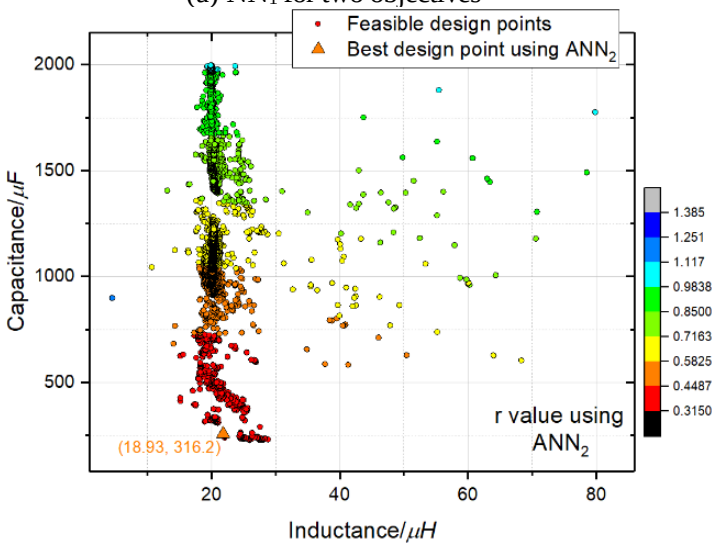
3.5.2 Optimization results

Based on the trained SVM and NN models, the filter optimization can be quickly completed since the internal derivations in above ML models are all mathematical and not too complex. Therefore, this surrogate-based approach will only rely on the math models, with no need for detailed simulation, analysis and derivations. Figure 3.16

shows the distributions of *r* value using above trained models. *r* value is an index integrating the two objectives: mass and power loss, based on Eqs. (3.23) & (3.24). Noting that SVM model is the same in the proposed generic surrogate structure while neural networks are NN_1 and NN_2 respectively. Therefore, the feasibile design points after using the SVM model are the same but the distributions of *r* value using two networks varies. Two subfigures in Figure 3.16 are using the same color map scale with Figure 3.13(a) (raw data). Thus the difference of *r* value distributions is obvious and Figure 3.16 (b) approximates the GA optimization results much closer than Figure 3.16 (a).



(a) NN_1 for two objectives



(b) NN_2 for one objective

Figure 3.16 Optimal filter design results using two NNs.

However, as shown in Table 3.4, it is important that the inputs of best design points obtained from two networks are the same and closed to that from the GA optimization process. After training, both NN_1 and NN_2 can provide good prediction performance with regards to their specific output(s). But NN_1 perform worse than NN_2 on r value prediction (since the RMSE of NN_1 is much larger). Even though NN_1 can predict close values for the solo objective: Mass or Power Loss, after using Eq. (3.24), r value becomes quite different because NN_1 gives slightly larger PL but smaller M. Therefore, the second way of building neural network (NN_2) should be better for potential MOOPs which have an index (r value) integrating one or more objectives. Even though the final design using NN_2 has a small error compared to the best design from GA algorithm, it is still acceptable because this design is very close to the GA best design and the filter system performance (M, PL, and control) should be similar and feasible.

Table 3.4. Best Design Point Using ML Models

	NN_1	NN_2	GA results
Inductance L_1	18.93 μH	18.93 μH	21.1 μH
Capacitance C_1	316.23 μF	316.23 μF	306 μF
Weighting factor λ_{dc}	3.4162	3.4162	3.3586
Power loss PL	0.9337 W	-	0.9431 W
Mass M	0.5349 kg	-	0.52753 kg
r value	0.282	0.3177	0.3168
RMSE - r value	**0.14142**	**0.0018842**	-

3.6 Experimental validation of optimal filter design

In order to validate the proposed filter optimization-based design methodology, three filter design points were selected in this section for the prototype preparation and experimental operation. The stand-alone AC load in the drive system comprises an LC filter and a linear load to match the rig in lab, where the AC capacitor voltage (v_f) is regulated by the FCS-MPC control strategy. Sizing models as well as the FCS-MPC fundamentals are the same; Moreover, there is no difference on the two-level inverter

and the DC passive front-end interface. The schematic of the studied experiment system is shown in Figure 3.17.

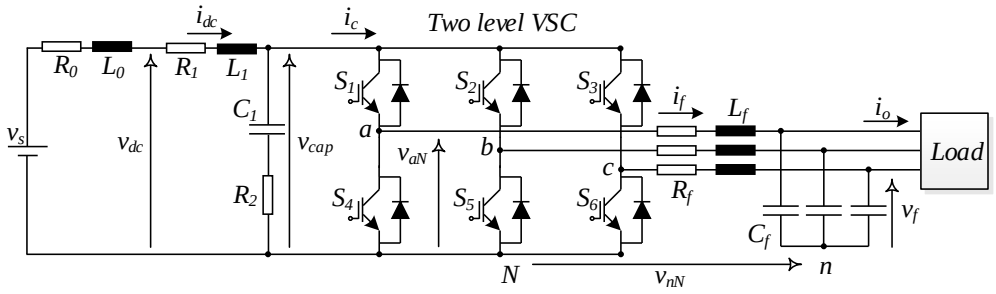


Figure 3.17 A CPL case study for experimental validation: Two-level VSC fed from a DC grid via *LC* filter and supplying an AC load.

3.6.1 Control strategy and delay compensation:

As mentioned, the FCS-MPC strategy should be adjusted from the above equations since the AC voltage is controlled here instead of the AC currents in the above EMA case. In addition, for the experiment concern, delay compensation of the controlled voltage vectors must be utilized, which are discussed below.

In the operating principle, Clarke transformation and 8 voltage vectors (Table 3.1) for the AC load are the same with the afore-mentioned FCS-MPC controlled drive system; however, the equations that describe the dynamics of AC side are for the AC *LC* filter and the linear loads in this case are:

$$L_f \frac{d\bar{i}_f}{dt} = \bar{v}_i - \bar{v}_f - R_f \bar{i}_f \tag{3.25}$$

$$C_f \frac{d\bar{v}_f}{dt} = \bar{i}_f - \bar{i}_L \tag{3.26}$$

where L_f, R_f, C_f are AC filter inductance, inductor inner resistance, filter capacitance, respectively. i_f is the AC filter current while i_L is the AC load currents. v_f is the AC capacitor voltage. Based on these two differential equations, the zero-order hold discretization method is used to precisely estimate v_f and i_f since that this method can

ensure the discrete-time model coincides with the continuous model at sampling instants. The discretization method is given as [20]:

$$\bar{i}_f(k+1) = \bar{i}_f(k) + \frac{T_s}{L_f}[\bar{v}_i(k) - \bar{v}_f(k) - R_f \bar{i}_f(k)] \tag{3.27}$$

$$\bar{v}_f(k+1) = \bar{v}_f(k) + \frac{T_s}{C_f}[\bar{i}_f(k) - \bar{i}_L(k)] \tag{3.28}$$

Different from using load current i_o to give i_{pol} in Eq. (3.16), here the current following into the VSC is given by the AC inductor current i_f and switching states of each inverter leg, as:

$$i_{pol} = S_a i_{fa} + S_b i_{fb} + S_c i_{fc} \tag{3.29}$$

The dynamics describing the DC capacitor charging and DC-side topology are the same with the former FCS-MPC thus, the estimation of $v_{cap}(k+1)$ is also using Eq. (3.18). However, in digital implementations of experiment, to compensate the inherent one-step computational delay, a two-step forward prediction approach is employed, which is implemented by predicting the *k+2* instant $\bar{i}_f^p(k+1)$ and $\bar{v}_f^p(k+1)$ using the same form of Eqs. (3.27) & (3.28) which stays at the same loop of v_i rolling optimization. Then, v_{cap} at k+2 instant can be predicted as:

$$v_{cap}(k+2) = v_{cap}(k+1) + (i_{dc} - \frac{i_{pol,i}^p + i_{pol,f}^p}{2}) \cdot (\frac{T_s}{C_1} + R_2) \tag{3.30}$$

where $\bar{i}_{pol,i}^p$ and $\bar{i}_{pol,f}^p$ are given by $\bar{i}_f^p(k+1)$ and $\bar{i}_f^p(k+2)$ using Eq. (3.29). This procedure effectively compensates the computational delay in the experiment operation. On the other hand, one of the basic characteristics of the conventional FCS-MPC is that it does not comprise the PWM. In stark contrast, FCS-MPC directly governs the semiconductor switching process. Therefore, there is no delay introduced by the PWM.

In the terminated step, CF is defined to determine the applied v_i at k+1. Generally, the total CF g_t is the same as Eq. (3.21) comprises three control terms, but as mentioned, the AC side voltage v_f is the control term g_{con} in this experiment, thus:

$$g_{con} = (v_{f\alpha}^* - v_{f\alpha})^2 + (v_{f\beta}^* - v_{f\beta})^2 \tag{3.31}$$

Finally, in the current limiting term G_{lim} Eq. (3.19), i_o should be replaced by the filter current i_f. The completed implementation of stabilization approach in experiment is depicted in Figure 3.18.

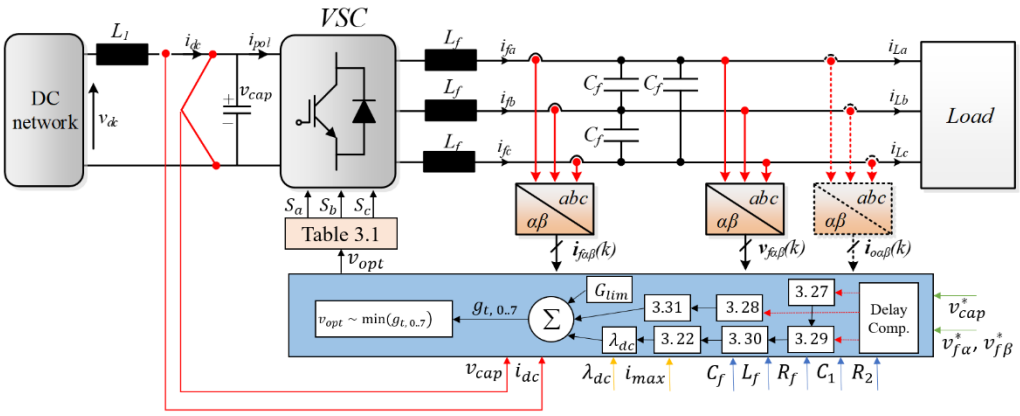


Figure 3.18 Complete implementation of FCS-MPC in experiment

3.6.2 Simulation GA optimization results:

Based on the same filter sizing models in Section 3.2, GA multi-objective optimization was exercised with a population size of 150 and 50 generations. Design variables (and their ranges), GA search method are all the same with above EMA case. The specific constraints and parameters in this optimization are summarized in Table 3.5. The signal constraints of DC-link voltage are the same with above EMA case. It is noted that two special constraints in this experimental case for AC voltage v_f are: the THD limit 3.5% and the maximum tracking error 2.5 V . Both are significantly sensitive to the optimization work because when they become smaller there would be lots of design points allowed to be feasible. Therefore, setting AC voltage constraints directly determines the final optimal results and they are usually set by specific applications.

The parameters of DC grid, loads and AC filter are predefined in both simulation and experiment. The turnaround time in experiment approximates 17 μs.

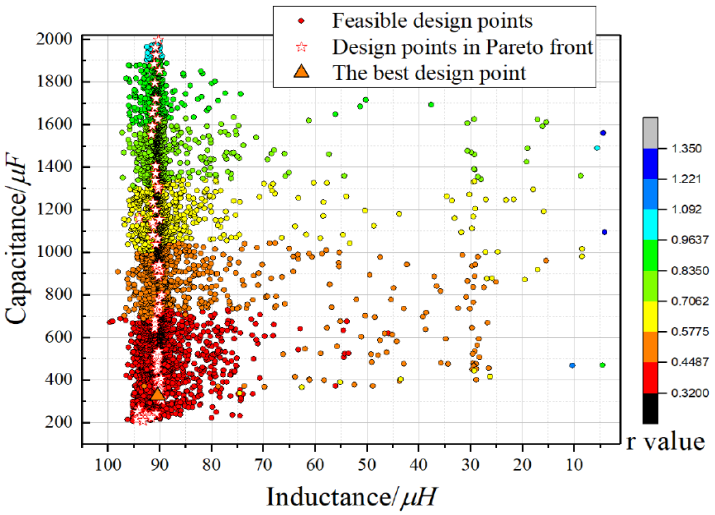
Table 3.5 System constraints and parameters in experiment

Two AC Constraints		**Parameters**	
Max tracking error of v_f	2.5 V	Supply voltage	270 V
Max THD of v_f	3.5%	MG impedance	L_0: 1 μH, R_0: 1 $m\Omega$
Parameters		Filter on AC side	L_f: 2.4 mH, C_f: 15 μF
AC reference voltage v_f^*	100 V	Sampling time	T_s: 25 μs
AC linear loads	60 Ω	Turnaround time	17 μs

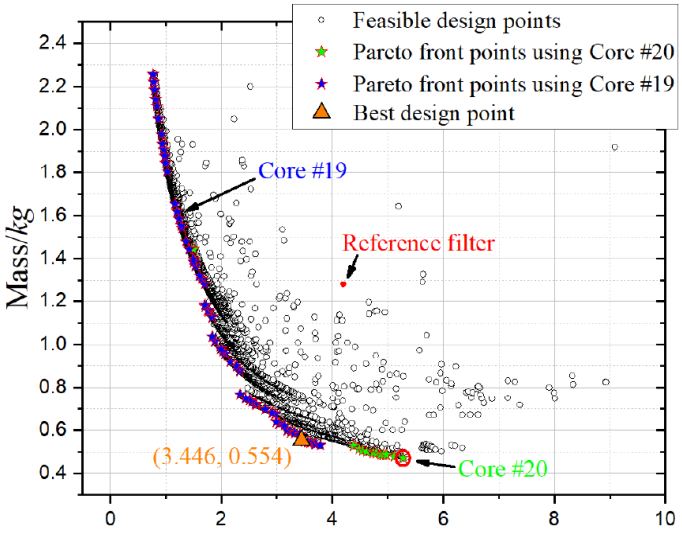
The GA optimization results using the detailed circuit simulation are depicted in Figure 3.19. The same with above EMA case, the criterion for choosing the best design from the Pareto front (marked by stars) is using Eqs. (3.23) & (3.24). Obviously, there are also data gaps among the Pareto-front points mainly due to the discrete inductor cores and the AC voltage constraints. Similar with the EMA optimization results in Figure 3.19, the inductance values of Pareto front points stay in a small range, [89, 95] μH, while their capacitance values cover most of the original design range, [1, 2000] μF.

As seen from Figure 3.19(a), there is a decreased trend of r value as L_1, C_1 reduce. The design points in Pareto front are also shown; however, they are not all preferred solutions according to the r-value criterion. The best design point is given in Table 3.6 together with the inner resistance. Figure 3.19(b) depicts two objectives of feasible designs. In [30], equality constraints are added to the solution distribution of Pareto front to ensure no vacancy or design gap appears in this front. In contrast, the Pareto front here is not processed by equality constraints. Similar with the EMA case, the Pareto-front solutions use two different cores (No. 19 and 20 here) for the inductor sizing. Apart from the best design, one Core-20 design (with the smallest mass in Pareto front) and a reference design (marked in Figure 3.19) are selected for the prototype making and experimental validation. The reference design is using a big capacitor thus regulation performance should be good. Therefore, it can be a benchmark for the two

Pareto-front designs. The design terms of Core-20 design and reference design are also given in Table 3.6.



(a) Design distribution of L_1 and C_1



(b) Feasible design points and the Pareto front

Figure 3.19 Optimal design results of drive system feeding a stand-alone linear load

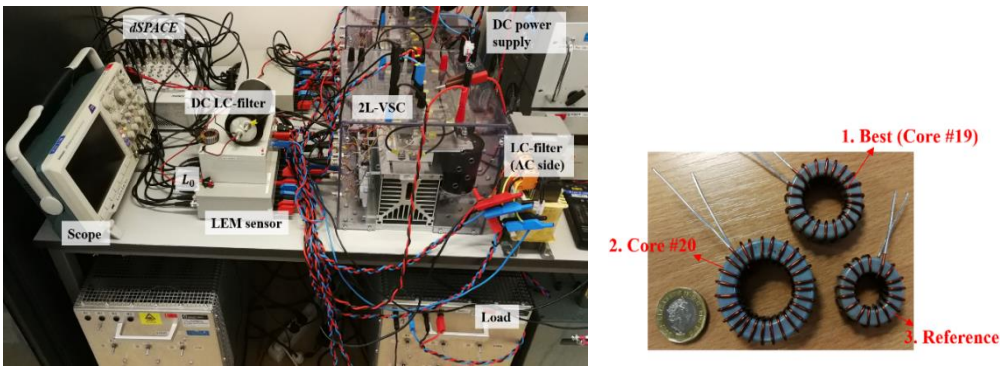
Table 3.6 Three Design Points for Experiment

	Best Design	Core-20	Reference
WF λ_{dc}	9.32268	7.7298	0.1665
L_1 (μH)	90.4	93.22	24.76
R_1 ($m\Omega$)	88.94	97.80	45.84
C_1 (μF)	326.7	208.87	1001.4
R_2 ($m\Omega$)	11.69	18.29	6.36
PL (W, Max:16.675)	3.446	5.275	4.199

	Best Design	Core-20	Reference
WF λ_{dc}	9.32268	7.7298	0.1665
M (kg, Max: 2.262)	0.5535	0.4714	1.2793
r value	0.32034	0.3788	0.6192

3.6.3 Experiment test:

Figure 3.20 shows the experimental setup and three inductors. The DC part consists of power supply, inductor L_0 and the studied LC filter. The AC side is comprised of an LC-filtered VSC and linear loads. FCS-MPC is implemented using a dSPACE DS1202 board. Figure 3.20(b) shows three selected inductors which are designed and manufactured following the proposed inductor sizing function (Section 3.2). Three capacitors utilized in experiment are found to match the corresponding C_1 thus the experiment can validate the accuracy of simulation results.



(a) Photo of experiment rig (b) Three inductors

Figure 3.20 Experimental setup for method validation

V_{cap} results of three designs in both simulation (grey background) and experiment (with signals zoomed in) are depicted in Figure 3.21. Results from simulations and experiments can be matched very well. Though ripple performance in experiment are not as good as simulation, all of them are acceptable. There is a clear trend that v_{cap} ripple declines as C_1 increases. However, the mass of capacitor will go up dramatically with the climbing C_1. Thus, the reference design should not be selected as a perfect LC

design though its DC stabilization is excellent. L_1value can affect voltage bias but not too obvious.

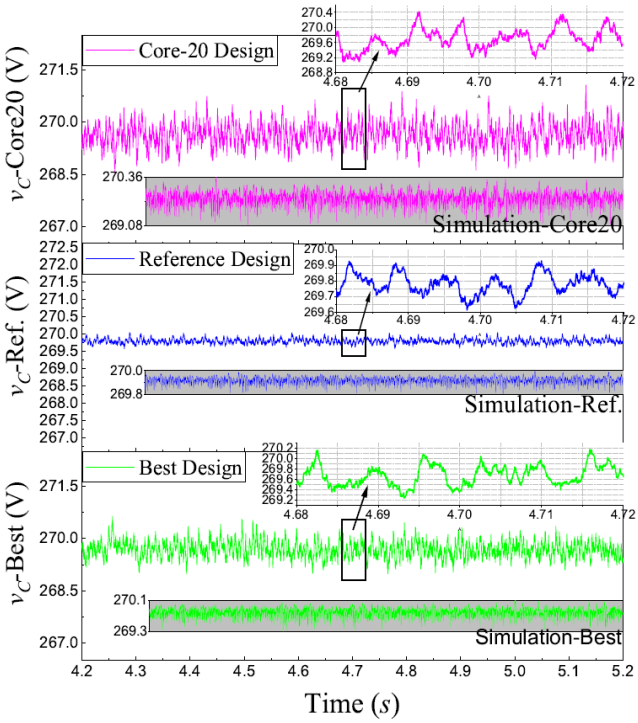


Figure 3.21 DC voltage results from simulation and experiment.

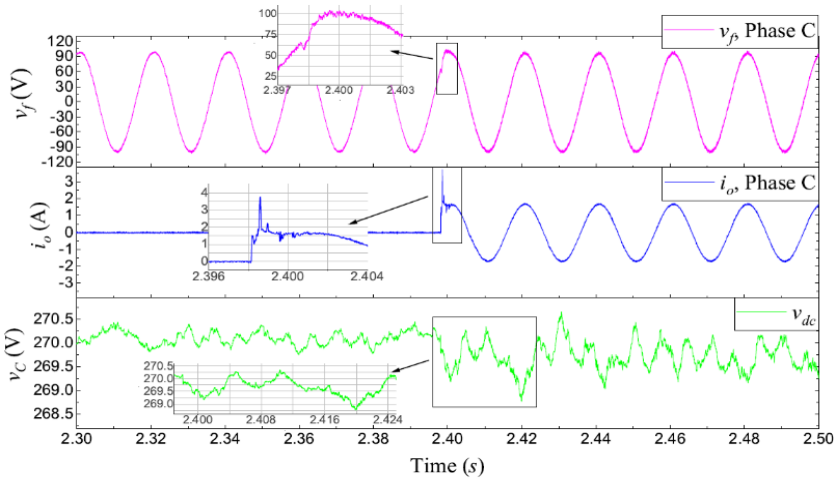


Figure 3.22 Transient performance of the best design point in experiment.

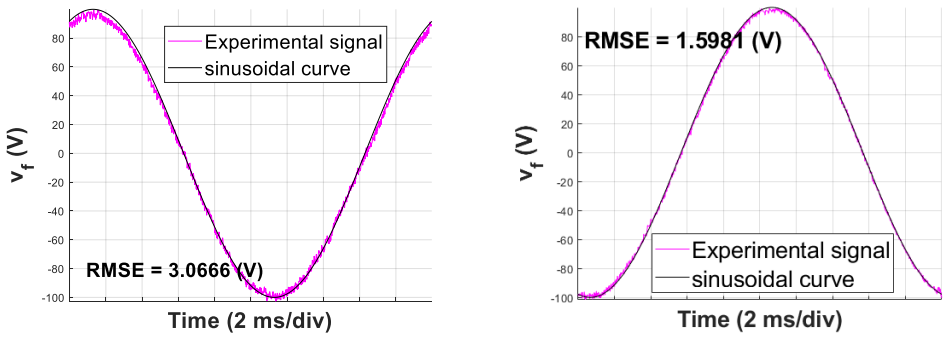
Figure 3.22 shows the dynamic performance of best design in experiment. As shown in the zoom-in signals, load switching has negligible effects on both AC and DC sides also i_L can be limited by G_{lim} (5 A in the experiment). Table 3.7 summarizes the v_f

THD, tracking errors and v_{dc} distortion factor performance under best design, Core-20 and reference design.

Table 3.7 Experimental performance of three design points

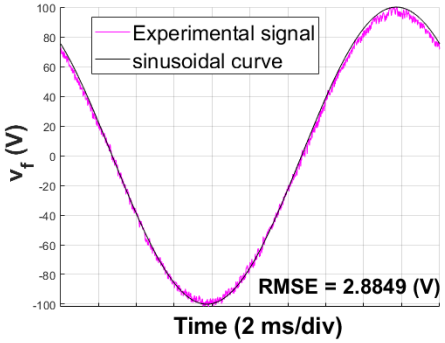
	Best Design	Core-20	Reference
THD	2.6%	3.23%	2.68%
Tracking error (V)	2.885	3.066	1.598
Distortion Factor	8.7×10^{-4}	12×10^{-4}	2.6×10^{-4}

The tracking errors are obtained by the extracted one-period data in Figure 3.23. THD results are all obtained by fast Fourier transform with two-cycle signals. The best design has the best THD performance (2.6%) though its tracking error is a little bigger than the Reference design. Therefore, the best design provides not only the best r value (given in Table 3.6) but excellent control performance: a similar steady-state performance with the reference design but without big C_1. That means the best design saves much mass also gives a good control performance based on the used FCS-MPC and GA searching ability.



(a) Core-20 design

(b) Reference design



(c) Best design

Figure 3.23 AC experimental signals under three filter prototypes

In addition, three distortion factors are all much smaller than 0.015, the maximum limit for 270 V DC systems [32]. The factors reflect a better DC regulation performance of the best design than the Core-20 design, which are both in the Pareto-front data set of the GA optimization.

3.7 Chapter Summary

This Chapter uses the proposed "2SA" to address a filter design and optimization problem onboard aircraft. The search algorithm used is multi-objective genetic algorithm (GA) and the other approach (surrogate algorithm) considers two machine learning algorithms: neural network (NN) and support vector machine (SVM). Commercial data-based filter sizing models as well as the drive control strategy (FCS-MPC) are discussed to serve as the basis of this optimization study. Apart from the simulation results, three filter design points were selected and tested on an experimental rig. The online tests demonstrate that the best design point not only saves much in weight but also guarantees a very good control performance using the FCS-MPC.

Chapter 4 Machine Learning Based Model Predictive Controller for Power Converters

In this Chapter, the machine learning (ML) algorithm is applied to address the power converters control topic. The main motivation is that ML algorithm has the strong ability to approximate the control laws (input-output relationship) [120] and the trained ML model requires very little computation burden for the control operation [63, 104]. Therefore, this ML approach can reduce the heavy computation burden of multi-step FCS-MPC while providing a similar control performance. In the following contents, related literatures are first reviewed, i.e. ML based controllers for power converters (including two-level and multi-level converters). Then, the system model and used FCS-MPC strategy are briefly introduced. Furthermore, the data collection and neural network (NN) training process are discussed. Finally, two trained ML controllers are tested in both simulation and experiment.

4.1 Previous studies on machine learning based controllers

In recent years, an interest of using machine learning techniques (e.g. neural network, NN) is growing in the control field of power converters [62-65]. In particular, neural networks are utilized to imitate the dynamic response of model predictive control (MPC) based on the MPC data collection and the NN offline training. The theory basis of this imitation approach is that the predictive control process is completely deterministic, i.e. for the same set of input variables (i.e. circuit measurements) and a given cost function, the outputs (inserted vectors/submodules) will always be the same. In this context, while the conventional MPC uses exhaustive rolling optimization in every time instant to

identify the optimal actuation, this is not necessary. It should be possible to represent the deterministic input-output relationship with more computationally efficient structure. Therefore, the same control effect as the MPC can be achieved by a NN, but with a lower online computational burden.

A deep neural network based predictive control strategy is generally presented in [68] for the application to high-frequency multi-level converters. However, none of technical details of machine learning work was provided, for example data collection, NN training & validation, etc. Therefore, the generalization of the five-hidden-layer NN strategy is very limited. In contrast, [62] clearly presents a NN learning approach for the control of two-level voltage source inverter feeding linear and non-linear loads. System descriptions, FCS-MPC principles, proposed NN architecture, train & test & validate are all presented in detail. Moreover, the authors collected sample data from 60 circuit conditions (covering the linear/non-linear load) and tested the trained NN for all the considered circuit conditions. Even though the proposed method in [62] was not validated in experiment, it is presented with all necessary technical details thus easy to be implemented and generalized. Following the same approach, [63] trained different NN controllers for up to N=3 prediction horizons and tested them in the experiment rig.

Apart from the NN imitation controllers, as mentioned, NN can also be directly applied online for the current/voltage control by training the NN in every sampling period. Namely, no need for the sample data collection and NN offline training, the internal parameters of NN (weights and bias) can be smoothly updated by the signal tracking errors at real-time [105]. This kind of approach could have the capability to perform a better dynamic response than the conventional cascaded linear controllers (e.g. PI or PR controller). However, this thesis will mainly focus on the imitation approach of NN for the control topic based on the offline training. Following the previous NN studies on two-level converters, this thesis will present two NN imitated predictive controllers for the modular multilevel converters (MMC).

4.2 MMC system principles

The system topology of the studied half-bridge submodule MMC (DC-AC inverter) is shown in Figure 4.1. In Chapter 3, for the input-filter design and optimization, a motor-drive system (EMA) was studied in Subsections 3.3-3.5 and the *LC* filtered linear-load system was studied in Subsection 3.6 for the experimental validation. In this chapter, the surrogate algorithm (neural network) is used to address the control issues of MMC, instead of the component design and optimization; Therefore, the components/topology in this MMC-based drive system are assumed and the control algorithm will be explored in detail.

The MMC considered consists of three phases, where each phase has two arms: upper and lower arm. Each arm is comprised of *N* series-connected half bridge submodules, and an arm inductance L_{arm} [121]. The submodule capacitor voltage is kept close to the rated DC voltage by MMC controller. In this way, the single submodule can be controlled as a voltage source by inserting or bypassing the submodule. The MMC output AC voltage can be controlled by changing the number of inserted submodules [122].

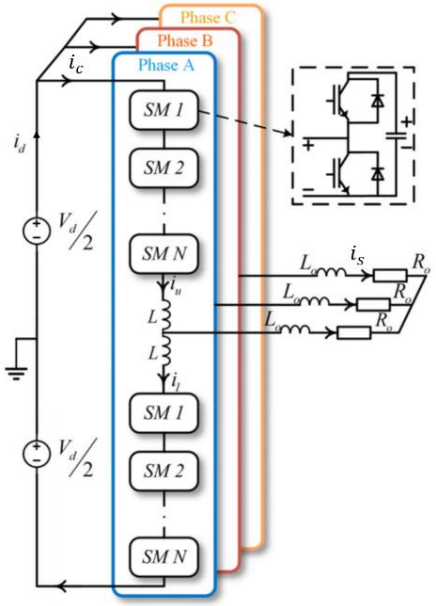


Figure 4.1 MMC diagram

The direction of upper arm current and lower arm current is shown in Figure 4.1. By applying the Kirchhoff's voltage law to the MMC circuit, the MMC dynamic equations can be derived as follows:

$$\frac{V_d(t)}{2} - v_{cuk}(t) - L\frac{di_{uk}(t)}{dt} = R_o i_{sk}(t) + L_o \frac{di_{sk}(t)}{dt} \tag{4.1}$$

$$\frac{V_d(t)}{2} - v_{clk}(t) - L\frac{di_{lk}(t)}{dt} = -R_o i_{sk}(t) - L_o \frac{di_{sk}(t)}{dt} \tag{4.2}$$

where V_d is DC-link voltage, v_{cuk} and $v_{clk}$are the upper and lower arm voltage in phase *k* respectively, i_{uk} and i_{lk} are the upper and lower arm current in phase *k* respectively. *k* means the phase number, *k*=0, 1, 2 (0 for phase a, 1 and 2 for b and c). R_o and L_o are the output resistance and inductance respectively. L denotes the upper/lower arm inductance. i_{sk} is output current.

High capacitor voltage ripples in MMC deteriorate the converter lifetime and may even cause tripping of MMC system. To restrict them, injecting double fundamental frequency circulating currents i_{ck} is a well-known approach; and Reference [94] proposed a NN-based surrogate model for providing suitable i_{ck} references for MMC under unbalanced condition. Define the output current i_{sk} and circulating current i_{ck} by using:

$$i_{sk}(t) = i_{uk}(t) - i_{lk}(t) \tag{4.3}$$

$$i_{ck}(t) = \frac{1}{2}\left[i_{uk}(t) + i_{uk}(t)\right] - \frac{1}{3} i_d(t) \tag{4.4}$$

Then, the dynamic equations of the output AC current and circulating current can be derived from above four equations:

$$\frac{di_{sk}(t)}{dt} = \frac{1}{L + 2L_0}\left[v_{clk}(t) - v_{cuk}(t) - 2R_o i_{sk}(t)\right] \tag{4.5}$$

$$\frac{di_{ck}(t)}{dt} = \frac{1}{2L}[v_d(t) - v_{clk}(t) - v_{cuk}(t)] \tag{4.6}$$

when the submodule is at 'on' state, the dynamic of submodule capacitor voltage can be expressed by the relation of capacitance and the arm current:

$$\frac{du_{cuki}(t)}{dt} = \frac{i_{uki}}{C_{SM}} \qquad \frac{du_{clki}(t)}{dt} = \frac{i_{lk}}{C_{SM}} \tag{4.7}$$

where u_{cuki} and u_{clki} are the *i*th upper and lower submodule capacitor voltages respectively (i=1…N, N is the number of SMs per arm); C_{SM} is submodule capacitance; i_{uk} and i_{uk} are the upper arm current and lower arm current in phase *k* respectively.

4.3 FCS-MPC strategy for MMC

In this section, the working principle of FCS-MPC for MMCs is introduced by using the dynamic equations in Section 4.2. This FCS-MPC model will be used to collect the input/output sample data for training the NN-based controllers in the next Section (Section 4.4).

Taking one-phase MMC as an example, the conventional FCS-MPC for MMCs can be introduced step by step as follows, where the detailed information of this MPC method is described in [123]. The control scheme of MPC MMC is shown in Figure 4.2.

The MMC discrete mathematical model can be established using the Euler forward equation:

$$\frac{dx(t)}{dt} \approx \frac{x(k+1) - x(k)}{T_s} \tag{4.8}$$

where $x(k+1)$ and $x(k)$ are the variable of at time instant *k+1* and *k* respectively; T_s is the sampling interval. Based on that, the dynamic Eqs. (4.5)-(4.7) can be transferred to the following discrete model:

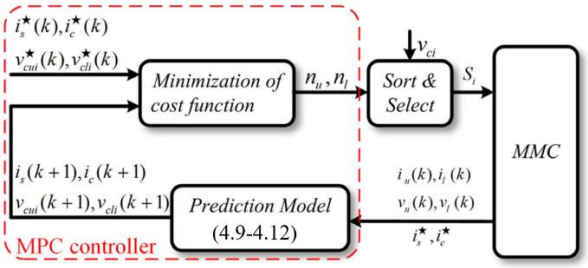
$$\begin{cases} i_s(k+1) = \mathbf{A}[(n_l(k)\cdot v_{cl}(k+1) - n_u(k)\cdot v_{cu}(k+1))/N] + \mathbf{B}i_s(k) \\ \mathbf{A} = 2T_s/(L+2L_o), \mathbf{B} = 1-2T_sR_o/(L+2L_o) \end{cases} \tag{4.9}$$

$$\begin{cases} i_c(k+1) = \mathbf{C}[V_d - (n_l(k)\cdot v_{cl}(k+1) + n_u(k)\cdot v_{cu}(k+1))/N] + i_c(k) \\ \mathbf{C}=T_s/(2L_o) \end{cases} \tag{4.10}$$

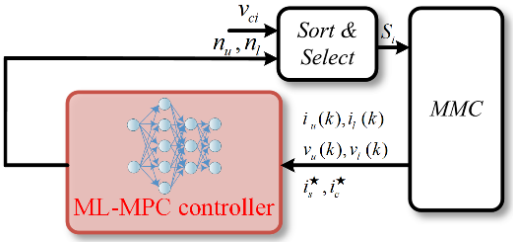
$$v_{cu}(k+1) = \frac{n_u(k)\cdot T_s}{C_{SM}} i_u(k) + v_{cu}(k) \tag{4.11}$$

$$v_{cl}(k+1) = \frac{n_l(k)\cdot T_s}{C_{SM}} i_l(k) + v_{cl}(k) \tag{4.12}$$

n_u, n_l are the insert number of upper arm and lower arm, respectively. Noting that they need to be derived in phase a, b, and c separately during the control.



(a) Traditional MMC-MPC controller



(b) Proposed ML-MPC controller

Figure 4.2 The comparison of ML MPC and traditional MPC.

In order to reduce the computation burden of FCS-MPC, the sorting and selecting of submodule capacitor voltages is achieved by independent sorting and balancing block, which is outside of the MPC algorithm, as depicted in Figure 4.2. The sorting and balancing method of MMC capacitor voltages is introduced in [122]. Finally, the cost function (CF) is the terminated step in this case is [104, 124]:

$$g = w_1 \cdot |i_s^\star(k) - i_s(k+1)| + w_2 \cdot |i_c^\star(k) - i_c(k+1)| \tag{4.13}$$

where w_1, w_2 are weighting factors for i_s and i_c respectively and, i_s^*, i_c^* are the references of output current and circulating current. This CF is based on the absolute error operation, squared operation of tracking error can also be applicable. In [125], two operations are analysed and the benefits of squared operation are given. However, it is not the key role of this study, no more details here. Actually, no matter which CF is used, the following NN learning process should be the same.

Compared to Eq. (3.31) in Section 3.3, obviously FCS-MPC for MMC (DC-AC) here only regulates the currents rather than the DC-link voltage. In this thesis, the control of output current and circulating current assumed to have the same priority thus, w_1 and w_2 both assumed as 1. Similar with the Section 3.6.1, Chapter 3, in order to compensate the delay of FCS-MPC in experimental setup, the delay compensation method should be applied. Here, a two-step forward prediction approach is employed. Generally, the i_s and i_c should be predicted at k+2 instant, not just at k+1, in the same loop of v_c rolling optimization of FCS-MPC ("Sort & Select" block in Figure 4.2). The compensation details can be found in [126].

4.4 Two NN based imitation controllers

In this section, two offline NN models are introduced to represent the deterministic relationship between inputs and outputs in the MMC FCS-MPC algorithm. As seen from Figure 4.2, the main difference between the traditional MPC MMC and the NN controlled MMC is that the NN-based controller simply replaces the MPC controller block. The NN inputs include six elements: upper arm voltage v_{cu}, lower arm voltage v_{cl}, output current reference i_s^*, upper arm current i_u, lower arm current i_l and circulating current reference i_c^*. The NN outputs include the upper arm insert number n_u and the lower arm insert number n_l.

The following content will present the design and training details of two different neural networks: NN regression and NN Pattern Recognition (PR). Though the training data collected from the FCS-MPC algorithm are the same for both networks, their data processing varies due to the different requirements of NN outputs. NN regression has no limits for the output element values but NN PR requires the elements must be either 0 or 1.

4.4.1 NN Regression

For the NN regression controller, it is designed as a 3-layer NN whose inputs/outputs are directly using the same design with the training data [shown in Figure 4.3(a)]. The reason of not using a 4+ layer NN is again to save the computation burden. This controller represents the following relation:

$$(n_{ui}, n_{li}) = F(v_{cu}, v_{cl}, i_s^*, i_u, i_l, i_c^*) \tag{4.14}$$

Here *i* means the phase number, i=0, 1, 2 (0 for phase a, 1 and 2 for b and c);

It is noted that the two outputs (insert number of lower/upper arm in each phase) should be integer and their values could only be 0, 1, 2, 3 or 4 in this studied system. However, for NN regression training, the predicted outputs are all continuous numbers, not integers. Therefore, two predicted outputs of this controller should both be rounded up to these 5 features/integers. Moreover, for all predicted negative values, the corresponding output should be 0 while it would be 4 for all the NN predictions larger than 4. After processing, the predictions of the trained NN controller can be compared with the outputs of raw data (collected from the original FCS-MPC algorithm) and generate the prediction accuracy of this trained NN (i.e. training performance). The accuracy results will be given in Section 4.4.4.

This NN regression controller is designed just following the original data structure of FCS-MPC in the studied MMC system: 6 inputs and 2 outputs, by using one NN. The

next subsection will introduce a novel design of NN controller based on the NN PR training method. This design can achieve a better training performance.

4.4.2 NN Pattern Recognition

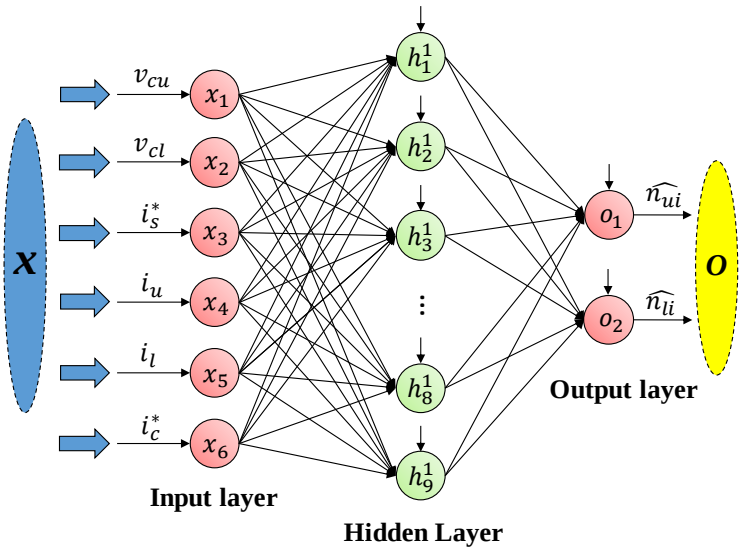
The NN PR controller design is shown in Figure 4.3(b). There are 5 elements in the NN output layer which corresponds to the 5 features of n_{ui}, n_{li} in Figure 4.3(a) (noting that the input elements are the same). The purpose of this design is to comprehensively explore the value space [0, 4] of n_{ui}, n_{li} and to give a more accurate prediction for MMC control using NN PR. Based on that, according to the value of n_{ui}, n_{li}, each element in output can be set 0 or 1 for the NN PR training. Therefore, the NN PR controller represents this relation:

$$(n_0, n_1, n_2, n_3, n_4) = F(v_{cu}, v_{cl}, i_s^*, i_u, i_l, i_c^*) \tag{4.15}$$

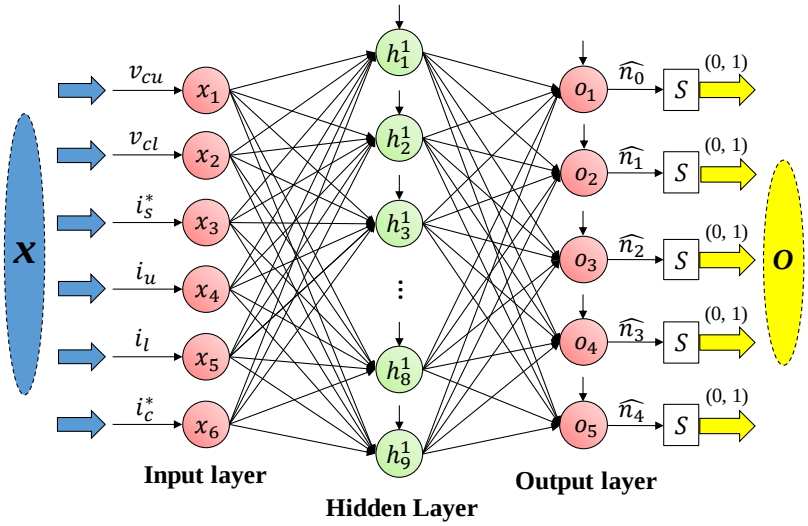
where n_i ($i = 0,1,2,3,4$) denotes the PR value that could quantify the probability of the situation "insert number equals i", because NN PR training can limit all predictions into (0, 1) by the Softmax function [87]. Softmax function, also known as normalized exponential function, is a generalization of the logistic function to multiple dimensions. It takes as input a vector [the output vector $(n_0, n_1, n_2, n_3, n_4)$ of NN PR in this case] and then normalizes it into a (0, 1) probability distribution of each component in this vector. It is used in the final layer of the NN PR controller as a 0-1 classifier. It's noted that the outputs n_i after the Softmax operation cannot be precisely 0 or 1, because there are totally five components in a vector and the sum probability of them should be 1.

The NN PR generates the probabilities for 5 features, which correspond to the 5 possible values of an insert number. Since there are two insert numbers considered, this NN PR controller for the studied MMC should use 2 networks, one for n_{ui} and the other for n_{li}. Besides, after getting the output vector $(n_0, n_1, n_2, n_3, n_4)$ in each network from a given input combination, 5 feature possibilities should be compared and the largest one n_j determines that the final resulting insert number is j for this input combination.

In this way, the trained two NNs can be used to replace the original FCS-MPC controller block in Figure 4.2(a).



(a) NN regression controller



(b) NN PR controller

Figure 4.3 Deployment of the proposed two NN-based controllers. Both have 3 layers with 9 neurons in the hidden layer. Weights and biases are omitted for simplicity.

4.4.3 Data Collection and Training Steps

To clarify the whole procedure of the NN imitation approach for MMC, all the steps are shown in Figure 4.4. Each step is further discussed as follows.

1) Generation of data samples from the original FCS-MPC operation. The first step should be collecting the raw data. The data can be extracted solely from the FCS-MPC algorithm block (only the math FCS-MPC model is used to generate the raw data). For the data collection works, the design ranges and sampling steps of inputs should be pre-confirmed. The sweep values of input data are v_{cu}, v_{cl}: [0:10:350], which means the range of upper and lower arm voltages is from 0 to 350V with the gap of 10V, therefore each arm voltage has 36 data points; i_s^*: [-6:1:6], 13 data points; i_u, i_l: [-6:1:6], 13 data points each; i_c^*:[0:0.2:2], 11 data points. Therefore, the training data includes around 31.32 million samples but, their collection time can be only few minutes on a standard computer because the FCS-MPC algorithm model used is a mathematic model.
2) NN models offline training. The extracted data in the last step was used to train the desired NN controllers (regression and PR), which represent the afore-mentioned relationships between input variables and output variables [Eq. (4.14) & (4.15)]. A workstation PC with Dual Intel Xeon Silver 4110 CPU is used to train the NN controllers, i.e. a feedforward NN and two pattern NNs. All networks have the same structure in the hidden layer and are trained for 800 iterations. Their training tool is the MATLAB Deeping Learning Toolbox [127]. More specifically, the NN regression controller is trained by using the "*feedforwardnet*" function in MATLAB while the two PR nets are trained by using the "*trainbr*" function in MATLAB. It is noted that, after offline training and validation, the quantization and sensitivity of the trained ANN can be further analyzed based on new samples and simulations, more details can be found in [128]. In this study, the trained model is directly used for the following online tests.
3) NN models online test. After getting the two trained NNs, they can be used to calculate n_{ui}, n_{li} of MMC in a real-time simulation or an experiment rig, the results will be provided in Section 4.5. As discussed in the last subsection, the

outputs of two networks should be typically processed to efficiently give the final predict n_{ui}, n_{li}, with which the output current i_s and the circulating current i_c can be controlled to track their references in the MMC operation.

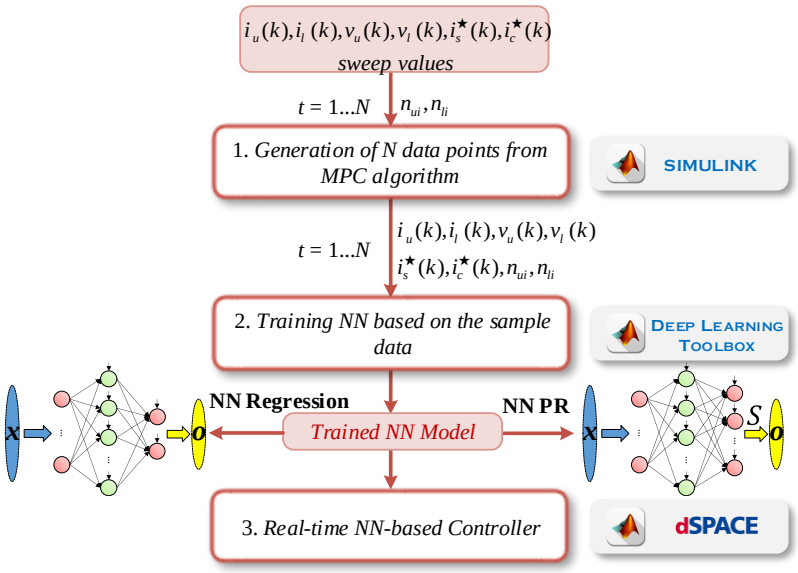
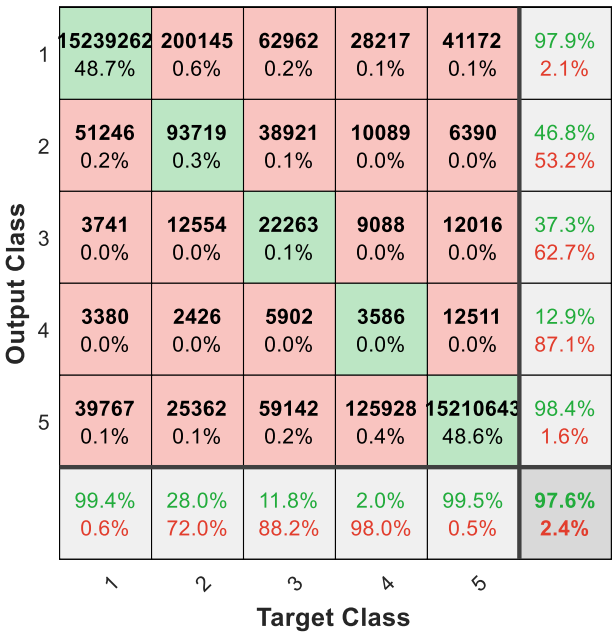


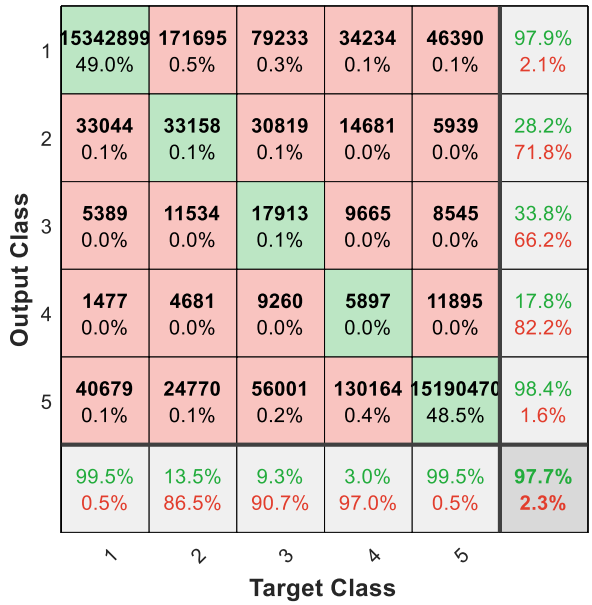
Figure 4.4 The proposed NN-based controller.

4.4.4 Training performance of two NN imitation controllers

To clearly demonstrate the training performance of the proposed NN controller, Figure 4.5 gives two confusion matrixes obtained from the NN PR training in MATLAB (one for n_{ui}, the other for n_{li}). As discussed, there are 5 features/classes in each NN PR controller and for every feature it is a classification problem. Each row in Figure 4.5 corresponds to an output class (i.e. n_0, n_1, n_2, n_3, n_4), and the columns are the target classes which are from the sample data. The green cells in the diagonal of the matrix show the number and percentage of correctly classified data points (after training, at the final training iteration), while all other red cells show the incorrect number of classifications and proportions. Therefore, nearly 98% of the predictions are matched with the raw data collected from FCS-MPC.



(a) NN PR training for n_{ui}.

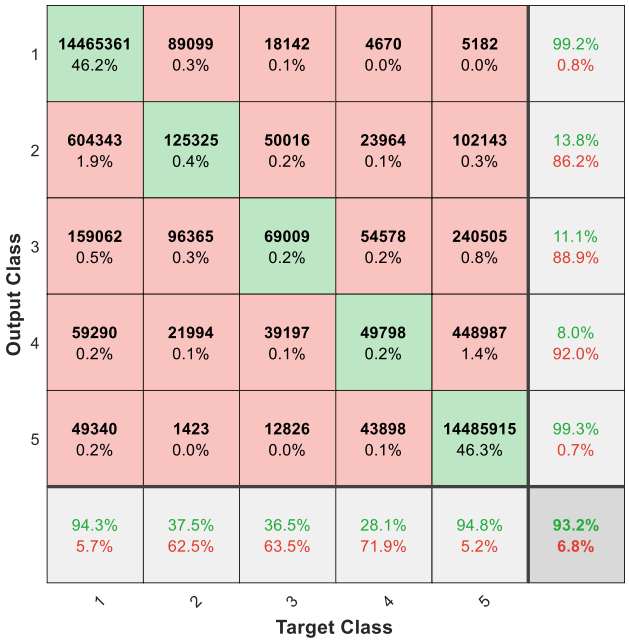


(b) NN PR training for n_{li}.

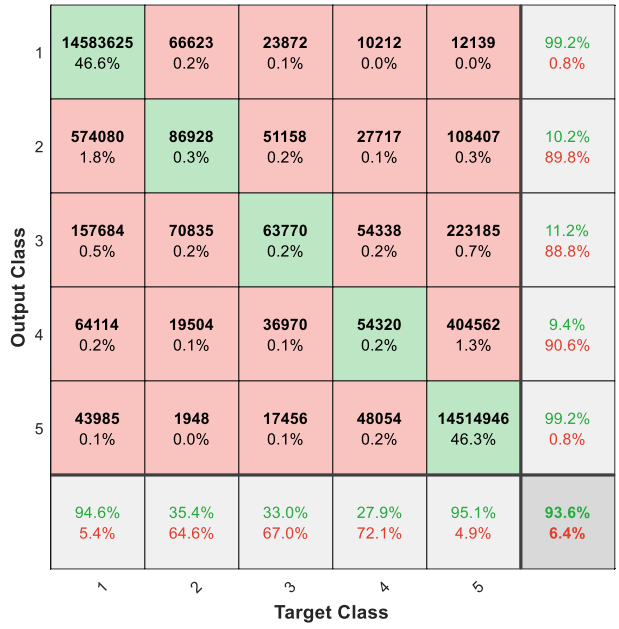
Figure 4.5 Confusion matrix of NN PR training.

Regarding the trained NN regression controller, its performance needs to be additionally calculated based on the sample data since there are no classification results during the training process (no Softmax operators). Two outputs in Figure 4.3(a) should be rounded up into [0, 4] and then compared with the target classes of sample. Then, the confusion matrixes for n_{ui} and n_{li} can also be obtained, as shown in Figure 4.6.

Obviously, the predict accuracy of the trained regression NN is only around 93.4%, lower than that of NN PR.



(a) NN regression training for n_{ui}.



(b) NN regression training for n_{li}.

Figure 4.6 Confusion matrix of NN regression training.

It is easy to find that, compared to NN PR, the main difference of NN regression is the better prediction performance of n_1, n_2, and n_3, but the worse performance of n_0 and n_4. Even though the general performance of NN regression is worse than NN PR,

the predict performance of n_1, n_2 and n_3 can be very important for the online controller test. Test results on the experiment rig will be discussed in Section 4.5.

Regarding the last column of Figures 4.5 & 4.6, they show the percentage of corrected prediction numbers for each row. For example, in 1st row, the green value is the percentage presenting how many predictions of 1st class are matched with the target data. Similarly, the bottom row of Figures 4.5 & 4.6 show that, in the target data of each class, how many of them can be matched by the corresponding predictions.

In Figure 4.7, the number of calculations and the horizon length results are shown. The number of calculations in conventional FCS-MPC control will significantly increases with horizon length. When the length is 1, the number of calculations is 16, however, the number of calculations increases to 256 when the length is 2. The NN based controllers have a small number of calculations even if the length is high. Due to the different design of two NN controllers, the corresponding numbers of calculations are different. The NN regression is 17, and NN PR is 20*2. The control results as well as the turn-around time of two NN controllers in the simulation/experiment will be given in the next section.

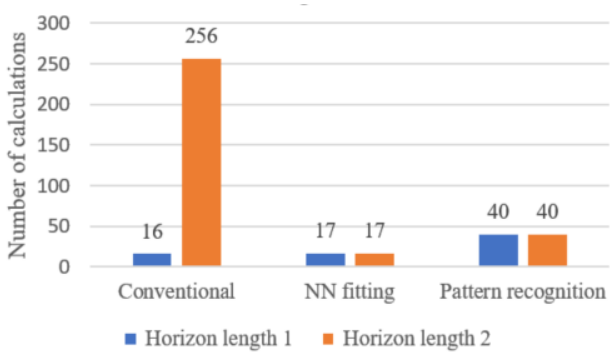


Figure 4.7 Comparison of the number of calculations and the horizon length.

4.5 Simulation and experimental validation

Both simulation and experiment results to validate the proposed controllers will be presented in this section. The parameters of the MMC case study are summarized in Table 4.1. Furthermore, in the simulation validation, both steady and dynamic performance of both controllers (regression and PR) are compared, as follows.

Table 4.1 MMC parameters

Number of SMs per arm	4	Arm inductance (L)	10 mH
Rated DC voltage (v_d)	200 V	Sample frequency	10 kHz
Capacitance C_{SM}	2000 μF	Load inductance (L_o)	1.8 mH
Capacitor Voltage (v_c)	50 V	Load resistance (R_o)	10.8 Ω
Rated frequency (f)	50 Hz		

4.5.1 Steady-state performance

The simulation results of steady-state operation are given in Figure 4.8 for the FCS-MPC controller, NN regression controller and NN pattern recognition controller. Seen from Figure 4.8(a1) to (a3), the three-phase output currents are controlled to track their references: AC currents with 5.5 A amplitudes and 50 Hz frequency. The THD performance is the target of designing the NN-based controllers, and THD results were obtained from both simulation and experiment for the method validation. The MPC controller has the best performance of THD which is 0.29%. And the NN regression has 3.9% THD, and the NN PR controller has 1.5% THD. Regarding the MMC circulating current i_c, the conventional FCS-MPC and NN PR controller both have good circulating current reduction effects, but the NN regression controller has the worst results which are shown in Figure 4.8(b2). The RMS of circulating currents of FCS-MPC and NN PR are 0.71 A and 0.73 A respectively, the RMS of NN regression is 0.81 A. Finally, the capacitor voltages are well balanced due to the common sorting and selection block, but the average voltage in the NN PR controller is higher which is depicted in Figure 4.8(c3).

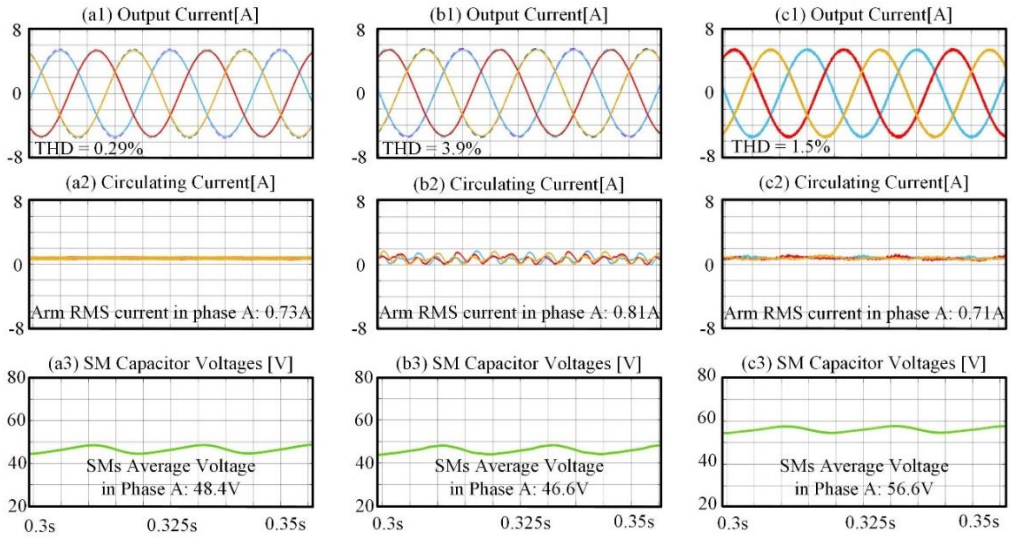


Figure 4.8 Steady state performance of three controllers in simulation: (a) conventional FCS-MPC controller, (b) NN regression controller, (c) NN pattern recognition controller.

Even though the load current ripple performance of NN PR controller is not as good as the conventional FCS-MPC and NN regression controllers, its THD performance in simulation [Figure 4.8(c1)] is better than the other NN controller [Figure 4.8(b1)] due to the higher overall NN training performance (see Figure 4.5). In addition, another priority of the NN PR controller locates at the dynamic response of load current which will be discussed in Subsection 4.5.3.

4.5.2 Experiment performance

Two proposed NN controllers are verified in the experiments which are carried out in a three-phase MMC, with 4 half-bridge SMs per arm. The proposed controller is implemented in DS1006 from dSPACE. The experimental setup is shown in Figure 4.9. The parameters of the simulation model and experimental setup are shown in Table 4.1.

Figure 4.10 shows the experimental results of the proposed methods. This figure clearly shows that, even though NN PR can achieve good control performance in simulation, two NN PR nets do not work well in the experiment (THD becomes larger than 8%). The main reason is that, as shown in Figure 4.5, the training performance of

the 2nd class (n_1), 3rd class (n_2) and 4th class (n_3) are very poor (their loss rates are up to 84.3%, 98.9% and 99.8% respectively). In contrast, the training performance of these three components for NN regression controller is much better, as displayed in Figure 4.6. The simulation results using NN PR are still acceptable because the simulated circuit and the environment in Simulink are relatively ideal without noise. But in experimental operation, the low-accuracy predictions of n_1, n_2 and n_3, practical noise and uncertainty in-circuit all affect the practical current control performance.

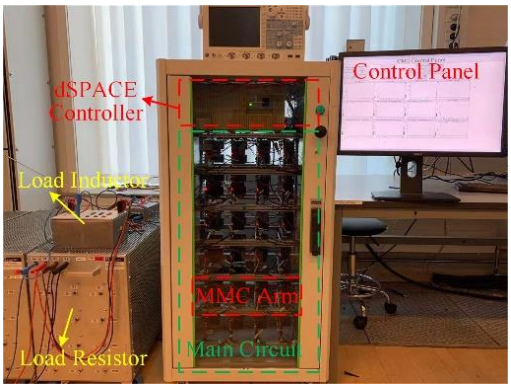


Figure 4.9 The MMC setup.

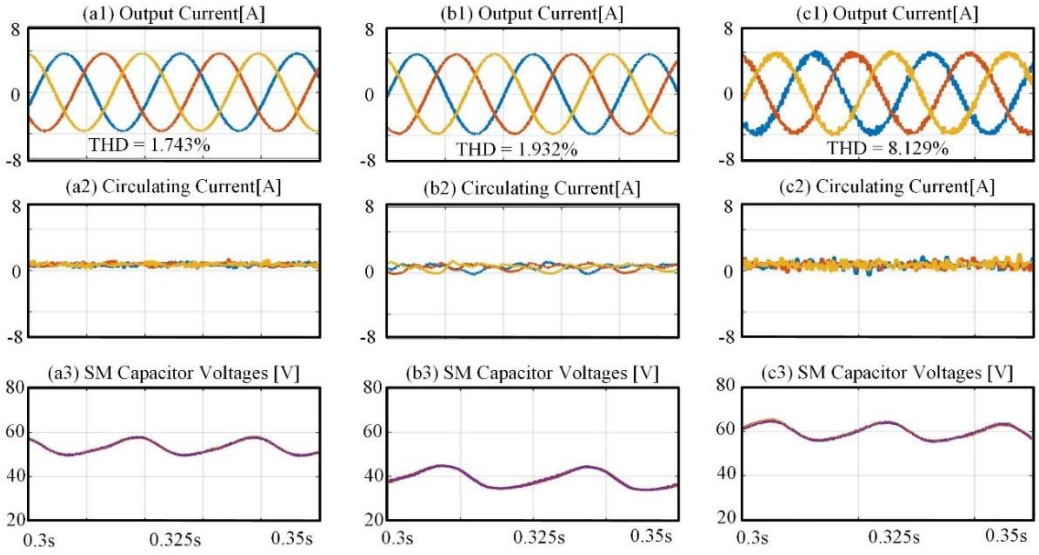


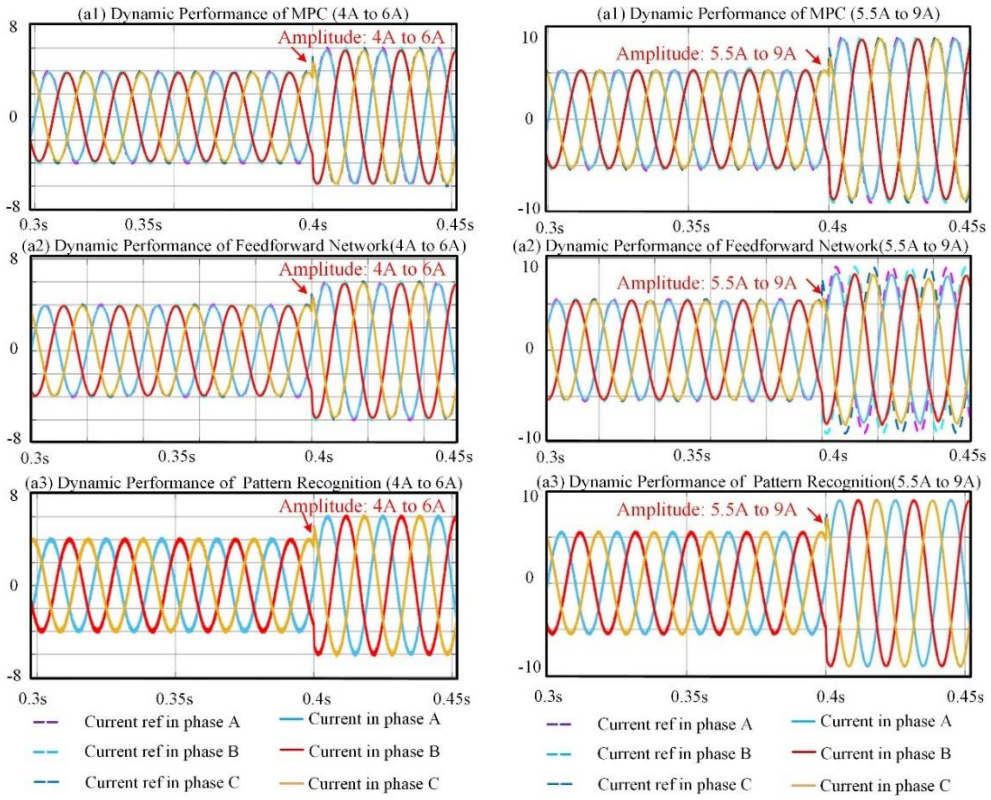
Figure 4.10 Experimental results of the (a) Conventional FCS-MPC, (b) NN Regression controller, and (c) NN PR controller.

It is noted that features of training data and the decoupled training of n_{ui} and n_{li} using two independent NNs determine the NN PR prediction performance (see Figure 4.5). The data amounts of n_2 and n_3 are both very small (smaller than 0.7% in the whole raw dataset); Therefore, to pursue a higher general NN training accuracy, the data points of n_0, n_1 and n_4 are given higher priorities in training which sacrifices the training performance of n_2 and n_3. On the other hand, two NNs are trained separately thus only one insert number (n_{ui} or n_{li}) is set as the training output in a NN. That means the mappings of n_{ui} and n_{li} are trained independently, no impact from each other. Even though the NN PR training can achieve high prediction accuracy, the trained two NN PR nets cannot reflect the general FCS-MPC characteristics of n_{ui} and n_{li} properly. In order to further verify this conclusion, two nets using NN regression for n_{ui} and n_{li} were trained separately and it has been found that, even though this two-net-regression method can achieve good general training accuracy (95.61% for n_{ui}, 95.12% for n_{li}) and acceptable simulation control, the experimental results indicate a poor control of the studied currents (THD is larger than 5%).

In contrast, the NN from one-net regression training [Eq. (4.14)] has a similar control performance with the conventional FCS-MPC in experiment, which demonstrates the fact that it has learned the control characteristics of FCS-MPC very well, also with lower computation burden. Therefore, this regression NN is finally chosen as the best NN approach for the MMC imitation control.

4.5.3 Simulation of dynamic performance

It is important to note that, although NN PR controller does not perform excellent in the experiment, it has the capability of covering an extra current range, which is out of the collected sample range (the range of i_s^* is from -6 A to 6 A). Figure 4.11(a) shows the dynamic performance of two NN controllers when the output current references are suddenly changed step-wise from 4A to 6A. Both NN controllers can process the stepped references easily in this range with a fast dynamic response.



(a) Dynamic performance within training range

(b) Dynamic performance out of training range

Figure 4.11 Dynamic performance in simulation (1). FCS-MPC, (2). NN Regression Controller, and (3). NN Pattern Recognition Controller.

In this thesis, the data collection range of i_s^* is from -6A to 6A. The dynamic performance of two proposed controllers were further tested when the range of i_s^* is larger than the NN training range (9 *A*). In Figure 4.11(b), the results show that, the NN pattern recognition controller can track the references even out of the NN training range. However, NN regression controller, cannot properly track the reference. Therefore, the robustness of the NN PR controller is better than the NN regression controller. It is noted that this subsection is mainly introducing one advantage of NN PR controller: it can work well when the output current reference i_s^* is beyond the NN training-data range. Therefore, the specific dynamic response speed of NN PR controller is not studied here, which could be reflected by the rotation form of the output current vector.

It is noted that, the design range of all the inputs are selected according to the specific testing rig in the lab. They should be changeable for other tests or specific

applications, e.g. aerospace; However, the designed NNs should be the same also feasible for other design ranges of variables.

4.5.4 Comparison summary

The networks are both trained by the collected data offline, during which the suitable bias and weights can be obtained. The computational burden of the NNs is very low due to its simple mathematical operation with bias and weights, thus NNs are very suitable to be implemented in a DSP or dSPACE controller for the sake of memory saving. The computational performance was verified using the dSPACE Profiler software. The turnaround time (i.e. code execution time) obtained using this software is given in Table 4.2.

Table 4.2 Turnaround time in experiment

	NN Regression	NN PR	FCS-MPC
Time (N=1)	1.123 μs	6.073 μs	1.615 μs
Time (N=2)	1.104 μs	6.088 μs	16.102 μs

As it can be seen from the experimental results, the NN regression has the lowest computational burden no matter what horizon length is. When the horizon length is high, the computational burden of FCS-MPC will increase dramatically, but the NN based controllers keep the same computational burden which should be a significant advantage.

The advantages and disadvantages of the proposed methods are finally summarized in Table 4.3. As concluded above, the NN regression method has the lowest computation no matter what horizon length is. Furthermore, the computation burden of NN controllers keep the same when the horizon length increased. This is the key advantage of the NN based method compared to the conventional FCS-MPC method. Regarding the control performance, the THD of output current is almost the same between FCS-MPC and NN regression controller. The THD of the NN PR is the worst in experiment. However, NN PR has a better ability to handle the input variables which are beyond the training data ranges.

Table 4.3 Comparison of three controllers

	NN Regression	NN PR	FCS-MPC
Computation burden (N=1)	Very Low	Medium	Low
Computation burden (N=2)	Very Low	Medium	High
Output Current THD	Low	Medium	Low
The ability to handle the input variables out of training range	Medium	Good	Good

4.6 Chapter summary

This chapter presented two different NN based predictive controllers for the control of MMC. The principles and the conventional FCS-MPC was introduced and two NN controllers (NN regression and NN PR) were illustrated in detail. The NN training data was collected from the controller operation of conventional FCS-MPC. The training performance of NN controllers was then compared and their differences were analysed. Finally, the three controllers were tested in both simulation and an experimental rig. In general, three controllers can both perform well in the simulation but, the NN PR controller is not working well in the experiment. The testing results demonstrate that the NN regression controller has learned the control characteristics of FCS-MPC very well, also with lower computation burden. In contrast, NN PR controller is not working as good as the regression controller but it can perform better than the regression controller when the output current is out of the range of training data. The reasons of these situations are analysed based on the confusion matrix results of trained NNs.

Chapter 5 Machine Learning Based Surrogate Correction Model for PMSM Multi-Objective Optimization

In this Chapter, the second optimization-based design case, a permanent magnet synchronous motor (PMSM) drive, will be discussed using surrogate algorithms in 2SA. This study aims at establishing the machine learning (ML) surrogate models to enhance the performance estimation of PMSM math models. The surrogate models are trained using a small amount of finite element analysis (FEA) data; therefore, this approach can speed up the conventional FEA-based PMSM optimization [52, 129] and no need to use a search algorithm. The main motivation is that, for a large design space, the analytical models of PMSM tend to lose the estimation accuracy (compared to FEA) due to their assumptions and simplifications [48]. The hybrid method (analytical models and ML models) will combine the accuracy of 3D finite element models with extremely low computational cost of ML models. Therefore, different from the generic structure of ML based optimization in Section 3.5, Chapter 3, ML algorithm is utilized here to train a dedicated correction model smoothly linking the math models to FEA in the entire design space. Noting that, although ML training requires a certain additional time for collecting a small amount of sample data, this is justified by a much faster evaluation of PMSM designs in the ML execution stages [129].

5.1 Analytical Models for PMSM Mass and Power Loss

This section will discuss the analytical mass and loss estimation methods as well as their simplifications and limitations. Generally, these analytical (math) methods pursue the design efficiency thus usually depend on one or more factors/coefficients, which is

prone to giving a poor estimation performance for large design spaces. It is important to notice that the purpose of analytical models is to give the base data for the following ML training (while the FEA model provides target data) thus, the models provided here can be changed to any other feasible or desired motor models with no impacts on the method application. The following subsections will introduce the mass and loss analytical estimations separately.

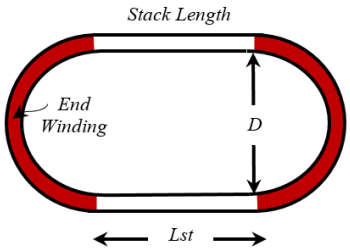
5.1.1 Mass estimation

In general, accurate estimation of winding mass is the most challenging part in PMSM because winding mass are dependent on quite a few parameters including material, end winding, number of turns, etc. Therefore, this section mainly is to discuss the limitations of conventional analytical estimation approaches for winding mass.

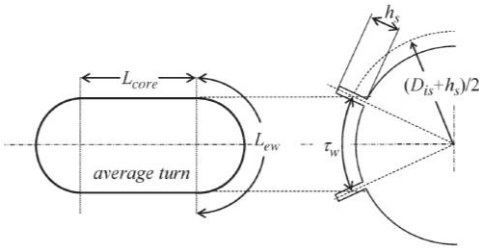
Mean Length per Turn (MLT) is usually utilized for analytically design of the motor winding based on the given number of turns, wire diameter and winding material [40, 42, 130, 131]. Based on the geometrical parameters, MLT is traditionally given by two parts: stack length and end-winding length. As shown in Figure 5.1(a), in [130], one MLT estimation method is using twice the sum of the stack length and the semi-circle arc around the corner (using slot pitch). This method is termed as "MLT Method 1". The equation is:

$$MLT_1 = 2L_{st} + L_{ew} = 2L_{st} + K_{ew1}D \tag{5.1}$$

where L_{st} is the stack length of motor and D here represents the slot pitch to estimate the end-winding length (L_{ew}) by using a coefficient K_{ew1}. K_{ew1} usually equals π which assumes that the one-side end-winding is a semi-circle, but this is not always suitable for different winding configurations, In practice, D can be corrected with K_{ew1} depending on the actual L_{ew} length.



(a) MLT Method 1, Sing Conductor with End Winding Length [130].



(b) MLT Method 2, Winding Average-Turn-Length Determination [131].

Figure 5.1 Two conventional computational methods of MLT

Another MLT determination method can be referred to [131], as shown in Figure 5.1(b), which also considers two contributions: the turn part embedded in the slot (L_{core}, which equals L_{st}) and the end-winding (L_{ew}), named as "MLT Method 2". However, the computational equations are slightly different:

$$MLT_2 = 2(L_{st} + L_{ew}) \tag{5.2}$$

where

$$L_{ew} = K_{ew2}\tau_w \tag{5.3}$$

K_{ew2} is the end-winding shape coefficient whose value is usually close to π/2 for wire windings, assuming a semi-circumference end-winding shape with a diameter equal to τ_w as

$$\tau_w = (1 - \frac{n_r N_{pole}}{N_{ss}}) \frac{\pi}{N_{pole}} (D_{is} + h_s) \tag{5.4}$$

where n_r is the pitch shortening ($\frac{5}{6}$ in this study), N_{ss} refers to the number of slots, N_{pole} is the pole number (defined by the air-gap fundamental spatial harmonic), D_{is} denotes the inner diameter of stator and h_s represents the slot height. Obviously, before applying Method 2 into the motor design and optimization, K_{ew} must be given as a parameter.

This study proposes another MLT computational method to derive the L_{ew} in Eq. (5.2), which utilizes the tooth width (Wid_{to}) and the bottom lengths of slot rather than the pole/slot pitch in above two methods. The main motivation is that, in traditional MLT

methods, the pole-pitch or slot-pitch based estimation only considers the circle at the middle position of slot in the motor cross-section but, the slot shape is not fully considered. In contrast, in the proposed method, two bottoms of slot are both considered. The proposed computational equation of L_{ew} is:

$$L_{ew} = \left(\frac{\pi}{2}\right)[Wid_{to} + K_{p,ew}(b_1 + b_2)] \tag{5.5}$$

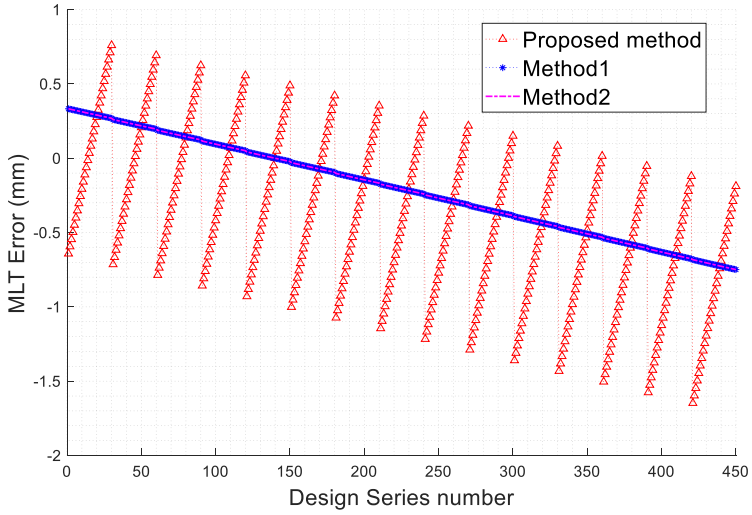
where $K_{p,ew}$ is the proposed end-winding coefficient, b_1 and b_2 are the lengths of slot upper/lower bottoms, as shown in Figure 5.3.

In order to compare the estimation performance of these three MLT methods, two design variables (DVs), tooth base fraction and PM height, are chosen to give the comparative sensitivity analysis based on the PMSM case of [48]. First, a base design point in the 3D model of a software, MotorCAD, is given to derive the values of three coefficients: K_{ew1}, K_{ew2} and $K_{p,ew}$. Namely, assume a motor design in MotorCAD and get the MLT value to inversely derive three coefficients by using above equations. The base design and the variable ranges can be found in Table 5.1. Two variables are tooth base fraction and PM height, three coefficients are obtained by the base design of (0.6, 4.4) in MotorCAD: $K_{ew1} = 0.5452$, $K_{ew2} = 1.4869$, $K_{p,ew} = 0.7383$. Then, evenly sample two DVs in a range (sampling range and step of two variables can be found in Table 5.1) and further collect the MLT results of both the 3D model and three afore-mentioned methods. Lastly, calculate the errors between analytical methods and the 3D model. The error variation and comparison results are shown in Figure 5.2.

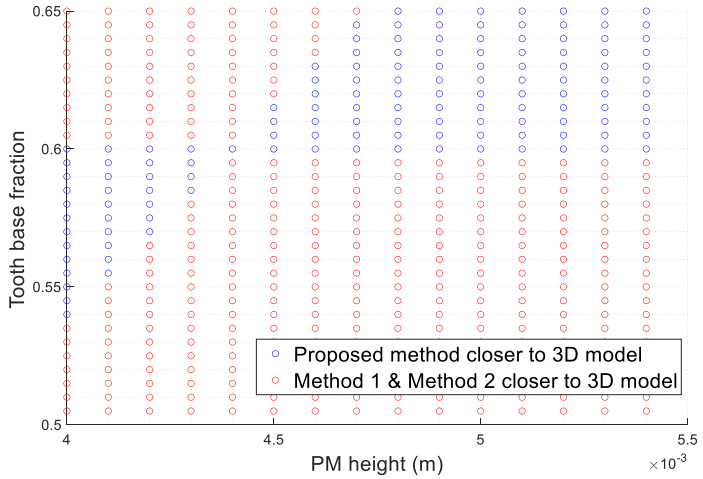
Table 5.1 Variables in MLT Estimation Studies

Variables or Coefficients	Tooth base fraction	PM height (*mm*)
Base Value	0.6	4.4
Sampling range	[0.5, 0.65]	[4, 5.5]
Sampling step	0.005	0.1

The horizontal axis of Figure 5.2(a) refers to the series number of the design points (circle marked) in Figure 5.2(b). Obviously, Method 1 has the same estimation performance with Method 2 though they are using slightly different equations. In contrast, the proposed MLT method appears a totally different variation direction from the other two methods. However, as shown in Figure 5.2(b), in these three MLT methods, there is no best analytical method for MLT estimation in the given ranges of two DVs. The proposed method can give closer estimation than Method 1&2 when using the blue circle design points but loses the priority for the red circle designs.



(a) MLT error variation



(b) Comparison of three MLT methods

Figure 5.2 MLT Estimation Results

Therefore, the analysis and investigations demonstrate that, in these three MLT methods, there is no one analytical method of MLT estimation which can cover all the situations in a PMSM design and optimization problem. The key reason comes from the fact that three coefficients (K_{ew1}, , K_{ew2} and $K_{p,ew}$) all stay unchanged in the design space and are thus not adaptive against the variable changing. To this end, this thesis proposes the ML aided approach to correct the motor total mass estimation. This is a data-driven approach: based on the data collection from analytical and FEA models, the dedicated ML correction model can be trained which will provide an accuracy and adaptive correction factor for the analytical mass estimation (against DV changes). This approach can get rid of choosing coefficient values and guarantee the high accuracy in the whole DV space.

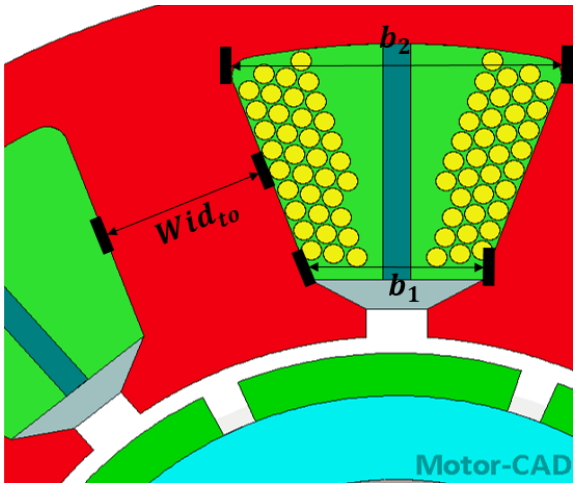


Figure 5.3 Geometrical parameters for the estimation of end-winding length.

5.1.2 Power loss estimation

When a PMSM operates under the no-load condition, iron losses contribute the largest part of the motor loss because the current in winding is nearly zero and Permanent Magnet (PM) loss is much smaller. To calculate iron loss in the soft ferromagnetic material, modified Steinmetz equation is used with two different terms, hysteresis loss and eddy current loss. The specific iron loss parameter W_{iron} (W/kg) for a certain material is given as [12, 42, 48]:

$$W_{iron}(f_e, \hat{B}) = K_h f_e^{\alpha} \hat{B}^{\beta} + K_e (K_{sf} f_e \hat{B})^2 \quad (5.6)$$

where $\hat{B}$ is the peak amplitude of flux density and f_e is the electrical frequency, K_{sf} is the stacking factor of the lamination sheets (assumed 1 in this study). K_h is the hysteresis coefficient and K_e is the eddy coefficient. K_h, K_e, α and β are all Steinmetz coefficients which can be determined by fitting the loss data from manufactures for specific materials. In this thesis, M235-35A steel is utilized for iron modelling whose Steinmetz coefficients can be found in Table 5.2. From mechanical perspective, M235-35A steel has an excellent and consistent processability in the manufacture of highly efficient electrical components.

Table 5.2 Parameters of M235-35A steel

Density (kg/m^3)	Lamination thickness (mm)	K_h	α	K_e	β
7600	0.35	0.0081294	1.208357	3.442366	1.78619

The first assumption of using Eq. (5.6) is that the stator flux density must vary sinusoidally (under f_e). However, in practice, the flux densities in stator tooth tip and in the back-iron parts linking stator base are far from a sine wave. Furthermore, as discussed, analytical loss models traditionally use the $\hat{B}$ of typical points to estimate the whole stator loss for the sake of simplicity. That also contributes to errors of analytical method compared with FEA models. Namely, the limit is that there is a non-uniform flux distribution on PMSM stator all the time. Lastly, flux saturation is not accounted for in this Steinmetz equation. Due to these limitations, this study proposes a ML-aided correction model to correct the predictions from Steinmetz equation, according to the samples collected from the FEA model. After the ML training, the ML model can provide dedicated correction factors for the iron loss estimation. And the corrected results can match the FEA values very well in the entire design space.

Three significant parts of motor losses are usually considered in the PM synchronous machines: iron losses, winding losses, and eddy current losses in PM [132]. However, in some optimization-oriented motor design models, power losses of PM are usually not accounted for because they are relatively much smaller than the iron loss [40-42], especially under the low-speed condition. Since in this study ML model is expected to correct the loss estimation with very high accuracy, PM loss should be considered as a part of PMSM total losses. The base PM loss model used here is assuming each PM as a plate of a sector of the circular ring (as shown in Figure 5.4) and Eq. (5.7) from [133] is utilized to predict the PM losses:

$$P_{PM} = \frac{1}{12}(2\pi f_e)^2 \hat{B}_{PM}^2 b\sigma R_2^4 (1-k_c^2)(1-k_c)^2 \beta (1-\frac{tanh(\beta\, c)}{\beta c}) \tag{5.7}$$

where $\hat{B}_{PM}$ is the estimated peak flux density in PM, σ is the conductivity of PM material, β is the half of the circular angle of PM (rad). R_1 and R_2 are the internal and external radius of the PM circular ring, c and k_c are defined as

$$c = \sqrt{\frac{2(1-k_c^2)(1-k_c)^2}{(1-k_c^2)(1-8k_c+k_c^2)-12k_c^2\, ln(k_c)}} \tag{5.8}$$

$$k_c = \frac{R_1}{R_2} \tag{5.9}$$

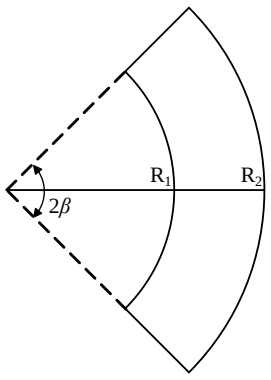


Figure 5.4 A plate of the shape of a sector of the circular ring [133].

Similar with the analytical methods in the above subsections, Eq. (5.7) only uses the $\hat{B}_{PM}$ of one specific point in PM to calculate the whole PM losses, which can

generate significant errors compared with the FEA. More importantly, for the analytical method itself, it does not consider the PM eddy current reaction effect. The reaction effect makes the flux more difficult to pass through the PM and more flux line passes through the leakage path [134]. Therefore, it directly affects the generation and calculation of PM losses. Authors in [135] found that the reaction field should be taken into account when the machine is excited by a pulse-width modulated current, especially for high-speed machines. [134] found that, if the reaction effect is neglected, the analytical results can only match the FEA for the PM with large numbers of segmentations. In addition, this analytical method is very sensitive to the PM height increase. Compared with FEA, the estimation error of Eq. (5.7) will quickly go up when the PM height increases.

To solve the above-mentioned PM loss estimation problems, this study utilizes an adaptive factor k_{PM} (as one of NN outputs) to correct the loss estimation in the whole design space of PMSM without PM segmentation. As mentioned, this PM loss model can be replaced with any other desired models which will not affect the proposed NN-aided approach. This is due to the excellent NN regression capability, the well-trained NN can smoothly correct the math estimation to the corresponding FEA results based on a very small amount of samples. A PMSM case study using ML models to correct the analytical estimation will be given in Section 5.4. Before that, the proposed ML-based correction model and the entire ML-aided multi-objective optimization procedure will be introduced in the next two subsections.

5.2 ML-Based Correction Model

This section presents the ML based correction approach for mass and loss estimation in PMSM design. There are five correction factors, four for power loss correction and one for mass correction: PM factor (k_{PM}), yoke factor (k_{YO}), tooth factor (k_{TO}), copper factor (k_{CO}) and total mass factor (k_{MA}). The training data of these factors are all

computed by dividing the FEA value by the corresponding analytical value. Average Correction Factor (ACF) method was proposed and utilized to compare with the neural network (NN) approach in [48]. Though it demonstrates the excellent correction performance of the NN approach, the estimation correction of PMSM mass and copper/winding loss are not considered in that study. This thesis will employ the NN approach for the correction of both PMSM total mass and total loss (iron loss + winding loss + PM loss). After NN training, the trained NNs can be served as the dedicated surrogate model providing the correction factors for the analytical model based PMSM multi-objective optimization. This approach can make sure that all the predictions from analytical models will be corrected to be extremely close to FEA results, in the entire design space.

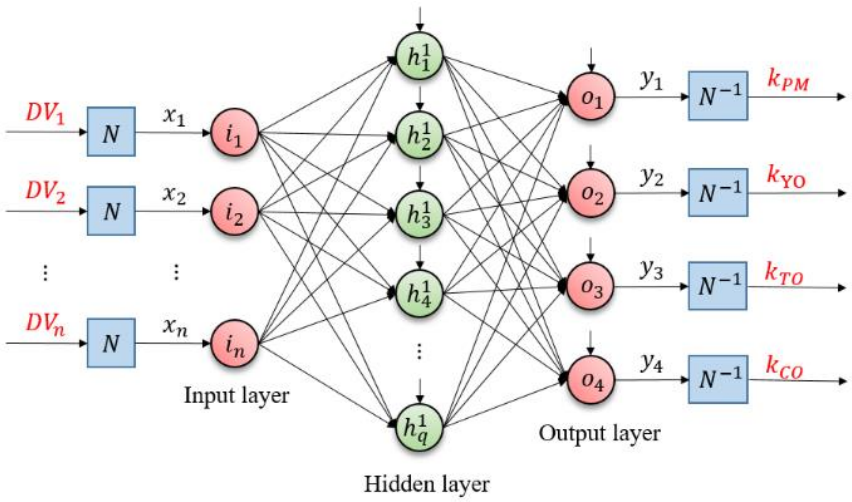
The same with the filter case in Section 3.5, Chapter 3, this thesis employs a feedforward NN to implement the mapping from DVs to PMSM correction factors. Feedforward NN is a well-known supervised learning technique in the machine learning and artificial intelligence domain. Though feedforward NN is the simplest type of NN techniques, it is a universal function approximator and has a powerful regression ability thus, it has already been applied into many studies of electrical and electronic research area [70], for example reliability design of power electronic system [90], inductor design and optimization [69], predictive control of modular multi-level converter [104].

Feedforward NN is based on a non-parametric regression model. Users do not need to specify the relations between the predictors (input) and responses (output) with a pre-determined function since NN will learn them automatically by updating the internal training parameters (i.e. weights and bias) [90, 93, 104]. This is particularly useful in multi-dimensional input-output spaces where parametric nonlinear regression approaches are not suitable since it is difficult to assume the structure of the function that best models input-output data relationships [90].

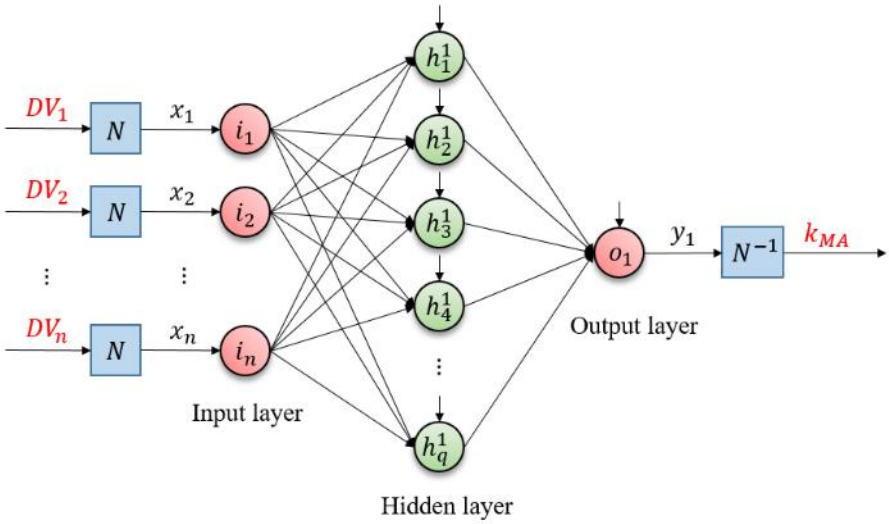
As discussed in Section 2.4, Chapter 2, a basic feedforward NN comprises an input layer, one or more hidden layers and an output layer, each hidden layer has one or several neurons. There are two different types of NN learning: regression [28, 90, 93] and pattern recognition / classification [28]. The PMSM performance correction study here is not a classification problem thus this paper focuses on the regression learning. NN regression can be trained to a nonlinear model for specific tasks by using a simple NN structure. Such a general nonlinear model can approximate any given input-output function with arbitrary precision [90, 136]. Therefore, its generalization capability is utilized here to correct the analytical PMSM sizing mass and losses in a whole design space. The correction performance will be in a very high level: the general relative error is usually smaller than 1%, see Section 5.4 for more details.

Two different NNs are used to correct power loss and mass separately, named as **NN_{loss}** and **NN_{mass}**. The input elements of two NNs are the studied DVs and they have the same hidden-layer structure, but their output layers are different. The output elements of NN_{loss} are four correction factors for power losses: PM factor k_{PM}, yoke factor k_{YO}, tooth factor k_{TO}, and copper factor k_{CO} , while there is one output element k_{MA} in NN_{mass} for the PMSM total mass. A three-layer NN is used; Regarding the neuron number (q) in the hidden layer, it can be set by a NN algorithm developer. In the following PMSM studies, it is simply set as 6 by trial-and-error. The reason of using this NN structure is that there are only 64 sample points in NN training thus good regression performance can be easily obtained with one hidden layer and 6 neurons, which benefits from the good global generalization capability of NN. More neuron numbers as well as additional hidden layers can be tried but the training performance would not change much. The diagrams of these two NNs used for the mass/loss correction are shown in Figure 5.5, where input and output layers are marked in red and the hidden layer is marked in green.

The same with the NN training in Section 3.5, Chapter 3, the raw data set was randomly divided into three data sets: training data (70%), validation data (15%) and test data (15%). Therefore, only 70% of raw data are directly used for training NN, other 30% of raw data are new for the trained networks. It should be noted that the precision of NN turned out to be highly robust to data division ratios, similar with the NN training work in [90]. Due to the small sample set and the great NN regression capability, empirical investigation has shown negligible differences in NN precision when the training set was from 50% to 90%. In the following contents, the RMSE of a trained NN is used to estimate as the error between the raw data and the corresponding predicted data, which stands for the NN training/mapping performance.



(a) Mapping loss correction factors, **NN_{loss}**



(b) Mapping total mass correction factor, **NN_{mass}**

Figure 5.5 NN deployments for loss/mass correction with normalization functions at the inputs (*i*) and de-normalization function at the output (*o*). Both NNs have one input layer, one hidden layer and one output layer. For simplicity, weights and bias terms are omitted from the figures.

5.3 ML Aided Multi-Objective Optimization Procedure

This section presents the proposed NN aided multi-objective optimization methodology. As shown in Figure 5.6, it is a three-stage-seven-step procedure. Three stages in this methodology are: PMSM modelling, NN training and multi-objective optimization. In the first two stages, there are three steps in each stage while only one step is in the third stage. Generally, the first stage is to establish PMSM analytical and FEA models, which serve as the basis of this optimization study; The second stage is collecting data from the established PMSM models and training the proposed NN-based correction models; Lastly, the third stage is using the trained correction models and the PMSM analytical models to efficiently do the global optimization. Three stages with seven steps will be introduced as follows.

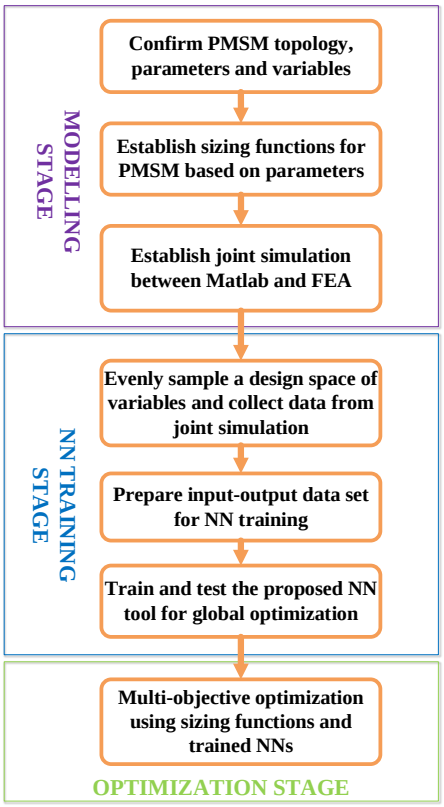


Figure 5.6 Procedure of ML aided multi-Objective PMSM optimization

At the beginning of modelling stage, the PMSM pole-slot topology, input parameters, design variables (and their ranges) and output performance indices (mass and power losses in this study) should all be confirmed as the optimization basis. Then, establish the analytical model which can derive the output performance by the given lumped parameters and the detailed math equations. After that, the joint simulation between the analytical model and the target FEA model should be built for sample collection (i.e. raw data for training). This study is using MATLAB to do the analytical modelling and collect FEA results from MotorCAD. Other software platforms should also be feasible and compatible with the methodology; as long as they can generate the acceptable results of motor performance with regards to the studied DVs. The next stage would be the data collection and NNs training stage to bridge the gap between the analytical models and FEA.

The ranges of DVs form the whole design space of the studied optimization problem. In the first step of NN training stage, samples are collected by the joint simulation built in the last stage. This study suggests evenly sampling four values for each DV (upper/lower boundaries and two values located in the middle). The scheme of sampling three values will be used in Section 6.5, Chapter 6, a motor-drive case. After data collection, the raw data of samples should be processed as the designed input-output training data set following the designs in Figure 5.5. Namely, the input elements are both DVs for two NNs; however, the output elements are four loss correction factors for $\mathbf{NN_{loss}}$ while only the total mass correction factor is in the output layer of $\mathbf{NN_{mass}}$. After the training data preparation, the last step of the second stage is to train two feedforward NNs. Many commercial NN training tools can be utilized in this step, e.g. Deep Learning Toolbox in MATLAB. It is noted that, this training step is usually a loop of trial-and-error, rather than one-time training. For a certain data set, different parameters (e.g. hidden-layer neuron number, training function in NN, Kernel function in SVM) in ML model can be tried to get a good performance. This step will terminate until the errors between ML model prediction and raw data are acceptable.

After the modeling stage and the NN training stage are accomplished, two desired NNs are prepared. They can become fast and dedicated surrogate correction models for the performance indices which are objectives of the studied PMSM optimization problem. Thus, in the final step of the methodology, the time intensive FEA model can be deleted and, the optimization in the original design space can be finished by only using the analytical models and the trained NNs (hybrid approach). Noting that the optimization in any subspace can be exercised in the totally same way. Since both analytical model and NNs are mathematic thus very efficient, exhaustive algorithm is used in this step for the global optimization. Namely, sample large number of design points in the whole design space (with very small steps between samples) and then generate all corresponding values of objectives. If there is one objective, the best design

can be directly found with the minimum objective value. Otherwise, the obtained objective values should be further processed, as follows.

In order to generate the best design point, the same with two filter optimization cases in Chapter 3, an integrated index r of two objectives is utilized to select one particular point for the Pareto front in objective space. The criterion for this decision-making solution is the minimal distance from ideal objectives [28-30]. However, in this Chapter, a weight of mass is added:

$$r_i = \sqrt{\lambda_{MA}\left(\frac{M_i}{M_{\max}}\right)^2 + \left(\frac{PL_i}{PL_{\max}}\right)^2} \tag{5.10}$$

M_i: Mass of the i^{th} solution; M_{max}: Maximal mass of all designs; PL_i: Power losses of the i^{th} solution; PL_{max}: Maximal power losses of all designs; λ_{MA} is the weight of mass objective.

λ_{MA} is usually set as 1 which means two objectives have the same priority. However, in practice, different objectives can be prioritized by changing the weight value in Eq. (5.10). Obviously, the best design obtained from Eq. (5.10) will differ with regards to λ_{MA}. In particular, when λ_{MA} increases, power loss of the best design will climb up while the mass will decrease, because the mass objective becomes more important. Conversely, if λ_{MA} decreases, for the same multi-objective optimization results, the best design point will have a higher mass and lower power losses. For the aerospace application, λ_{MA} is usually set larger than 1 since the mass objective has a higher priority. A PMSM optimization case and its optimization results will be given in the Section 5.4 for the methodology validation. After that, comparisons with the conventional FEA-based optimization will be discussed.

5.4 PMSM Case for Method Validation

In order to demonstrate the excellent correction performance of the NN aided approach, a 12-slot-10-pole PMSM design case is illustrated to generate the mass/loss correction

results using NN. The analytical PMSM sizing model in [42] is used as the basis of this study which is a non-iterative and high dimensional sizing model. It can generate a comprehensive design of multi-physics fractional-slot PMSM in a very short time. There are more than 20 parameters to be designed which includes 12 geometries. Three of them are selected as DVs in this case: stator tooth base fraction (α_t), airgap height (h_{ag}) and wire diameter (D_{wire}) whose ranges are given in Table 5.3.

Table 5.3. Variables and Parameters of PMSM Case and NN Training Performance

Design Variable	DV1: Tooth base faction	DV2: Airgap height (*mm*)	DV3: Wire diameter (*mm*)
Range	[0.5, 0.7]	[1, 2]	[0.6, 0.7]
Sampling Step	0.01	0.1	0.002

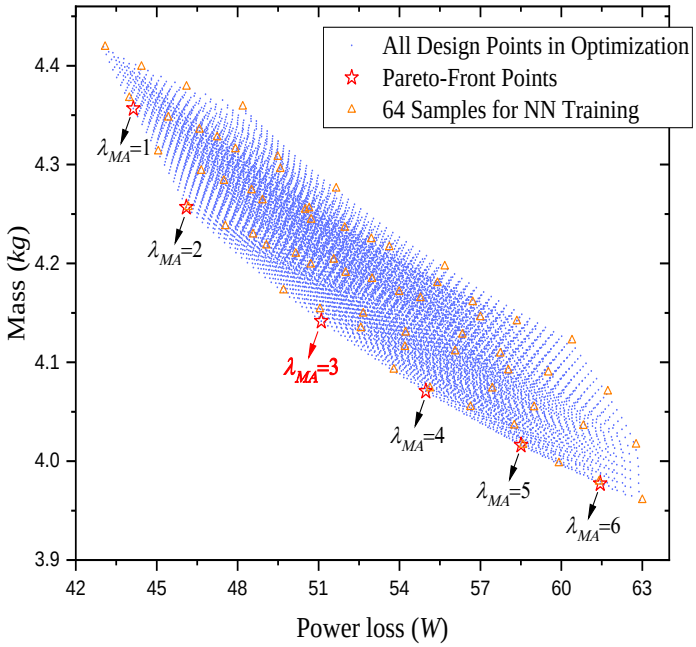
Parameters		Rated speed (rpm)	1500	Electrical frequency (Hz)	125
Input RMS current (*A*)	10.162	No. of turns per phase	80	Depth of tooth tip (*mm*)	1.26
Axial length (*mm*)	54	Depth of tooth base (*mm*)	19.67	Tooth tang angle (Deg)	38.38
Coil density (kg/m^3)	8960	Coil conductivity (S)	5.7×10^7	PM height (*mm*)	4.4
Rotor radius (*mm*)	28.6	Yoke Depth (*mm*)	6.88	Tooth tip fraction	78.83%
PM fraction	88.19%	PM density (kg/m^3)	7500	PM relative permeability	1.033
Shaft radius (*mm*)	17.5	Shaft density (kg/m^3)	7820		

Error in NN	$E_{PM}\ (W)$	$E_{YO}\ (W)$	$E_{TO}\ (W)$	$E_{CO}\ (W)$	E_{MA} (kg)
RMSE Value	0.001015	0.005925	0.0010199	0.0001177	5.8629e-05

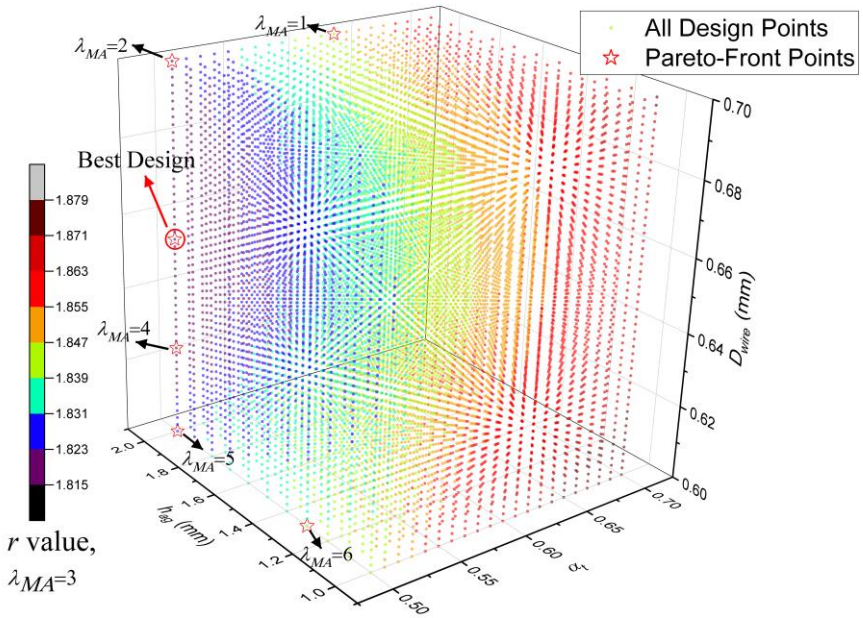
As discussed, the first stage of the proposed approach is to collect training data from both math and FEA models. Therefore, the sample scheme of the raw data should be clarified. In this case study, four values were evenly sampled for each DV, which include two boundary values and the other two are evenly distributed in the middle. Therefore, there are $4^3 = 64$ sample points collected for the NN training in this PMSM optimization.

In the NN training stage of optimization, 64 samples were first collected from the joint simulations (MATLAB + MotorCAD) and they were processed as the input-output data set for two NNs (see Figure 5.5). Then, two NNs were trained using the *train* command in MATLAB [137], both in 1 *sec*. The RMSE training performance of two NNs are given at the bottom of Table 5.3. Finally, sample large number of design points in the 3D design space of α_t, h_{ag} and D_{wire}, the objective values of all these designs can be quickly obtained by using the analytical model and two trained NNs.

In the last stage, the small steps of DVs for the global optimization are given in the third row of Table 5.3. Thus, 21 values of α_t, 11 values of h_{ag}, and 51 values of D_{wire} were sampled. The total sample number is 11781 and the collapsed time for generating all the objective values of samples is 365.44 *secs* on a standard computer. Then, the optimal design was then simply obtained by finding the lowest value of *r* in Eq. (5.10), which can be done in MATLAB almost instantaneously using the embedded *min* function [138]. The PMSM optimization results are depicted in Figure 5.7.



(a) Objective-value distributions and Pareto-front points



(b) *r*-value distribution and Pareto-front points in DV space

Figure 5.7 PMSM optimization results with different λ_{MA} value

All the design points in optimization as well as six Pareto-front points are presented. The objective value distributions of design points are shown in Figure 5.7(a) where there is a clear Pareto front at the bottom-left side. Two objectives of all designs were integrated into one (*r* value) by using Eq. (5.10) with a specific λ_{MA}. This study swept λ_{MA} value from 1 to 6 and used Eq. (5.10) to generate the six Pareto-front points in

Figure 5.7(a) (M_{max} and PL_{max} should be found beforehand). Obviously, with the increase of λ_{MA}, power loss of the best design goes up while the mass declines. Since this study is for MEA applications, mass should have a priority over the other objective thus, λ_{MA} is set as 3 here for the final best design confirmation. Another reason of choosing 3 is that, the data distribution in the objective space determines this "$\lambda_{MA} = 3$" best design point can get a good balance between two objectives, not like "$\lambda_{MA} = 1$" or "$\lambda_{MA} = 6$". In the same subfigure, the 64 samples collected from joint simulations are also shown, which demonstrates that the sample amount is very small compared with the optimization sampling.

On the other hand, Figure 5.7(b) shows the design points in the 3D DV space (α_t, h_{ag} and D_{wire}). Noting that the color of design points only shows the r value distribution when λ_{MA} equals 3. The color distribution will be changed with different λ_{MA} values. In this subfigure, there is a clear trend that the distance from the "Best Design" in this 3D space has a positive relation with r value, i.e. the far the larger. Corresponding to Figure 5.7(a), the three DV values of six Pareto-front points are marked in Figure 5.7(b) among which the min-r design of "$\lambda_{MA} = 3$" is the final best design point in this PMSM optimization case.

Finally, after finding these six Pareto-front points, their 3D values were input into the FEA platform, MotorCAD, to verify the accuracy of the NN aided optimization methodology. As summarized in Table 5.4. Performance verification of six Pareto-front points, all performance results are extremely close to the target FEA values no matter for mass or power losses. The absolute relative errors of M are all below **0.01%** and for PL, the relative errors are around **-0.15%**. Therefore, the mass mapping performance (**NN_{mass}**) is better than that of power losses (**NN_{loss}**). It should be noted that most of the six Pareto-front points are not included in the 64 samples, which again validates the excellent performance of NN aided optimization approach.

Table 5.4. Performance verification of six Pareto-front points

λ_{MA}	α_t	h_{ag} (mm)	D_{wire} (mm)	*PL*, NN aided approach	*PL*, FEA	*M*, NN aided approach	*M*, FEA
1	0.62	2	0.7	44.1356	44.2044	4.3566	4.3568
2	0.5	2	0.7	46.1027	46.1758	4.2568	4.2570
3	0.5	2	0.654	51.1041	51.1791	4.1416	4.1418
4	0.5	2	0.624	54.9784	55.0549	4.0706	4.0709
5	0.5	2	0.6	58.5054	58.5831	4.0162	4.0166
6	0.5	1.3	0.6	61.4448	61.5482	3.9771	3.9774

5.5 Comparison with FEA-based Optimization

Compared with the conventional FEA-based optimization, the first advantage of the proposed methodology comes from the efficiency of doing optimization. As discussed, the sample amount of NN training (obtained from the joint simulation) can be very small (64 points in the case study), which means this approach only relies on the time-consuming FEA model in a very short time. Then the multi-objective optimization can be finished by only using the extremely fast trained NNs and analytical models (both of them are mathematic models). In the case study, the proposed NN approach can generate a PMSM design in 40 milli-seconds while it costs more than 1 minute in the FEA model on MotorCAD.

Another advantage of the NN approach is that it has no risk of getting stuck on local optimum. Namely, it is a global optimization approach based on the exhaustive algorithm, without using any classical or modern search algorithm. Since the FEA consumes much time, it is usually not realistic to use the exhausted method to generate the multi-objective optimization [139]. Popular solutions are using search/heuristic algorithms, e.g. genetic algorithm [28-30, 55], particle swarm [84]. However, in that cases, sample hundreds even thousands of design points are still needed to get the Pareto

front for a multi-objective optimization problem [30, 115]. More importantly, search algorithms would bring the risk of getting stuck on local optimum. As discussed in Section 2.4, Chapter 2, if the crossover and mutation rates are both small values, GA will probably get stuck on the local optimum because the probability of finding a better local optimum becomes very little in a finite evolving process. A global optimum is a point whose objective function value is better than all other feasible points. A local optimum is a point where the objective function value is better than that of nearby points, but possibly not good as at a distant point (this distant point maybe the global optimum or it can be another local optimum). Based on the trained correction models and the PMSM analytical models, the proposed NN aided optimization method is obviously finding the global optimum of studied problem, because the entire design space is fully swept for the evaluation of objective function(s) with large numbers of design points, which is based on an exhaustive approach.

Last but not the least, the hybrid model of trained NNs and math models can provide the best design point smoothly for any potential DV subspace. Moreover, it would also be finished in a very short time due to the high efficiency of the hybrid model. In fact, after getting the global optimization results with very small sampling steps, the sub design space optimization can be just implemented based on a specific part of the obtained data. That means, if the former sampling steps are acceptable in a subspace optimization, the subspace data set, obtained from the hybrid-model-based global optimization, can be directly used to give the best design point, no need to run the hybrid model anymore. However, for the FEA-based optimization using search algorithms, the FEA model may need to be run all over again. The reason is that, in a subspace, the previous searched samples can be very small thus cannot be utilized to give the optimal design in a new optimization problem.

However, there are two specific limitations of the proposed NN aided optimization method:

1) The trained NN may not work well for the variable ranges out of the original design space. Namely, if enlarge the design space, the prediction performance of the trained NN may be affected. The reason is that the raw data collected for NN training do not cover the whole design space. For the design points out of range, the FEA model is still needed for the accurate performance estimation though it is not efficient. Therefore, for the method generalization, it is suggested the motor variables and their design ranges should be well selected before the data collection and the final NN training.
2) After doing the NN-aided optimization, the final best design point obtained may not be precisely close to the corresponding FEA model result due to the training error of NN correction model. Even though the best design generated by the hybrid analytical model can be very accurate (due to the NN's powerful regression capability), this point should be further validated by the FEA model. Therefore, the high prediction precision could not be guaranteed for specific applications. If the difference with FEA model is too big, the main reason is that the sample data is few (especially for the design points near this best design) thus, it is suggested that more data points should be collected for a new round of NN training.

5.6 Chapter summary

Chapter 3 discussed a filter multi-objective optimization problem (MOOP) in a drive system, which is using both search algorithm and surrogate algorithm. Differently, in this chapter, a permanent magnet synchronous motor (PMSM, another component/subsystem in the drive system) MOOP is presented by only using a surrogate algorithm, neural network (NN). NN is trained here as the surrogate model to correct the performance estimation from the PMSM analytical model. In this way, the corrected performance values can be highly close to the corresponding finite element analysis (FEA) results. It is noted that, the presented analytical approach (with NN

correction) is very computationally light thus should be much faster than the FEA approach. A PMSM case is given, which demonstrates that the NN correction models can be easily trained by a small data set. Based on the PMSM analytical model and established NN correction models, the best design of PMSM was quickly found in the whole design space.

Chapter 6 Optimal Design of Motor Drive Using Machine Learning Algorithms

In this Chapter, the third optimization-based design case, a permanent magnet synchronous motor (PMSM) drive system, will be presented. Different from the Chapters 3 & 5, which studied individual components in the motor-drive systems, this chapter will study the design and optimization problem of the entire motor-drive system. The *LC* input filter system assumes to be fixed and the control strategy applied in the drive system is the conventional proportional–integral (PI) control. Similar to Chapter 5, neural network will be used to train a dedicated surrogate model for the performance correction. However, unlike the PMSM optimization case, only one objective (i.e. power loss) will be considered in the drive system optimization. Two objectives, mass and power loss, will be considered in the future study. Math models as well as joint-simulation models for the drive system will be first introduced. A small amount of samples are collected from joint-simulation models and used to train a neural network (NN). The trained NN will be validated by comparing the NN predictions with the target loss values collected from the joint simulation. Finally, the validated NN is further utilized to do the global optimization in the 3-dimentional design case.

6.1 Motor drive system description and its joint simulation

6.1.1 Motor-drive system description

In the studied motor drive system, there are two main parts: a two-level VSI (DC-AC) and a PMSM, as shown in Figure 6.1. The DC filter at the DC supply side has been comprehensively studied in Chapter 3; thus, in this Chapter, it is assumed the *LC* filter

has been predefined and remains unchanged. Thus, this *LC* filter will not be considered during the power-loss oriented motor drive design. In addition, since the vector PI control strategy and space vector modulation are applied, the conventional filter design approach is feasible based on the fixed switching frequency [12, 34].

The PI controller of a two-level inverter provides the converter modulators with appropriate output voltage vector references, to regulate the motor output torque and speed [12, 53]. For field-oriented vector control, the sinusoidal *abc* current and voltage components are transformed into a rotating *dq* frame. This enables the zero steady state error tracking of the current references with simple PI compensators [12, 47, 53, 140]. With the conventional cascaded control scheme employed, motor currents are controlled in the inner control loop, and the speed control is in the outer loop. In addition, to ensure that converter voltage limits are not exceeded, a field-weakening strategy is required [140].

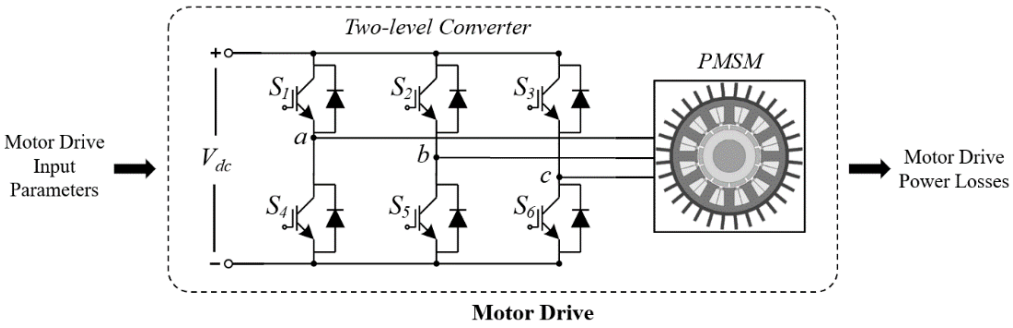


Figure 6.1 The diagram of the motor-drive study.

In the studied optimization problem of motor-drive system, for each design point, the PMSM is fed by a converter operating at a certain condition where both DC supply and motor output reference should be assumed (for example, motor speed, motor output torque). Then, power losses of two subsystems (converter and PMSM) are computed separately and the input parameters of their loss estimation models are different. However, in the analytical approach, some key parameters of the PMSM (such as motor torque, *dq* inductance, etc.) should be provided to the converter analytical model for the

further evaluation of the converter's performance [12, 41]. Therefore, in the detailed simulation, PMSM finite element analysis (FEA) model should be first exercised then the drive system can be run with all necessary parameters. In order to pursue high accuracy of the detailed simulation, a joint simulation scheme of motor drive systems is proposed here. This scheme can combine the PMSM FEA model (using MotorCAD) with the drive simulation system (using PLECs) under the operation of MATLAB codes.

In the following subsections, the modulation schemes for the two-level converters will be first illustrated. After that, the proposed joint simulation scheme for the motor-drive optimization will be discussed in detail.

6.1.2 Modulation for two-level converter

Converter modulation is a process of switching semiconductor devices within a power converter from one operation state to another, which determines the converter operation conditions, power quality and semiconductor losses [141]. As discussed in Section 3.3, three-phase two-level converters with three phases have 8 switching states. At time instant, an arbitrary target output voltage vector of a converter can be synthesized by at least two active vectors applied for a fraction of one switching period. After calculating the relative turn-on times for the switching state vectors, the sequence, in which they are realized within each switching period, is determined by the modulation scheme. Standard converter modulation schemes include the symmetrical Space Vector Modulation, and 60°-clamping discontinuous PWM (DPWM1). The fundamentals of these two modulation strategies will be discussed below. Before that, some definitions of the converter switching should be given.

The two-level converter topology is shown in Figure 6.2(a), where S represents the Insulated Gate Bipolar Transistors (IGBT) device, D represents the Diode device. A switching function j_M can be defined for each phase leg. When $j_{M,i}$ is 'p', the high-side switch of phase leg i is turned on, while its complementary low-side switch is turned off. When $j_{M,i}$ is 'n', the opposite is true. A combination of switching functions for all

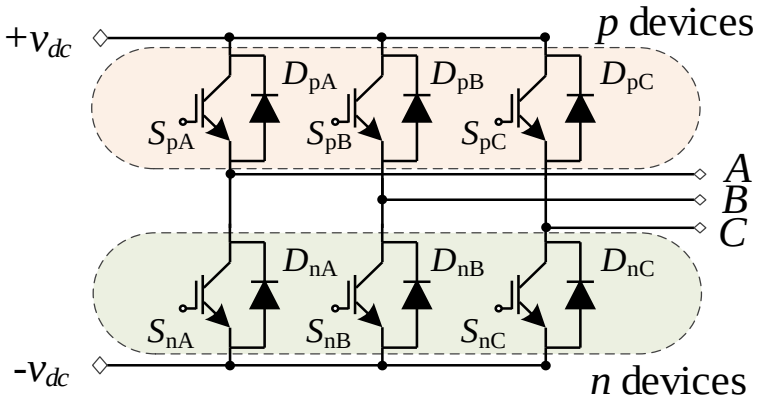
three phases is known as the converter switching state K_M ($j_{M,1}$, $j_{M,2}$, $j_{M,3}$). For example, when switches S_{pA}, S_{pB}, and S_{nC} are turned on, K_M is (ppn).

The 8 switching states ($\bar{v}_{0..7}$) for two-level converter can be plotted as $\alpha\beta$ stationary vectors on a complex plane which is also known as the *space-vector* diagram, as shown in Figure 6.2(b). The K_M which corresponds to 8 switching states are also marked. The switching states $\bar{v}_0$ and $\bar{v}_7$ have zero magnitude thus they are known as zero vectors, while the other six states have a magnitude of $\frac{2}{3}V_{dc}$ (V_{dc} is the DC-link voltage) and they are known as active vectors. In time domain, an arbitrary target output voltage vector $\bar{V}_{ref}$ can be synthesized by at least two actors.

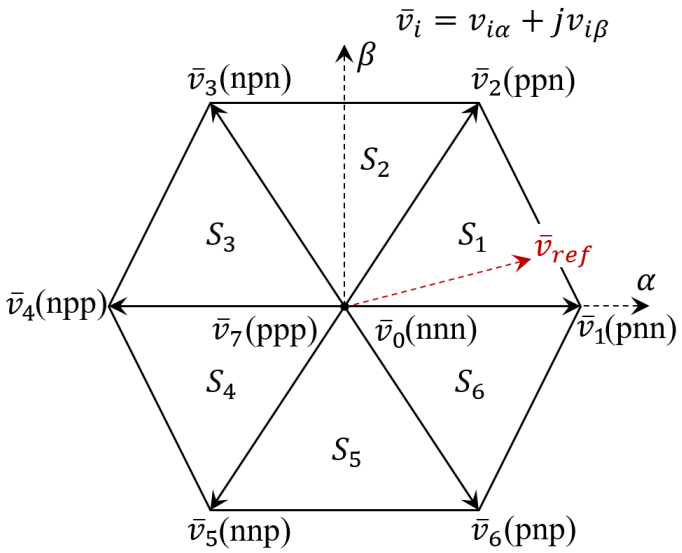
For the symmetrical SVM modulation strategy, six basic sectors are defined ($S_{1..6}$, on the complex plane) are defined based on two active vectors nearest to $\bar{V}_{ref}$, as it rotates around. With this modulation method, in each sector, the two active vectors and both zero vectors are utilized to output $\bar{V}_{ref}$ in the switching period. For example, in Sector 1 (S_1), the two nearest active vectors are (pnn) and (ppn). Since for each switching, only the state of one device can be changed; therefore, the determined switching states sequence is (nnn)-(pnn)-(ppn)-(ppp)-(ppn)-(pnn)-(nnn), involving six commutations per switching period.

For DPWM1, on the other hand, twelve sectors are defined in the space vector diagram, by the two active vectors nearest to the $\bar{V}_{ref}$, and zero vectors, which alternate every $\pi/3$ interval. The twelve sectors space vector diagram can be seen in Figure 6.2(c). The main difference between symmetrical SVM and DPWM1 is that, in each sector, the latter utilizes the two active vectors and only one zero vector to realize $\bar{V}_{ref}$ in the switching period. For the same Sector 1 example, the DPWM1 switching states sequence is (pnn)-(ppn)-(ppp)-(ppn)-(pnn), where switch S_{pA} is effectively clamped.

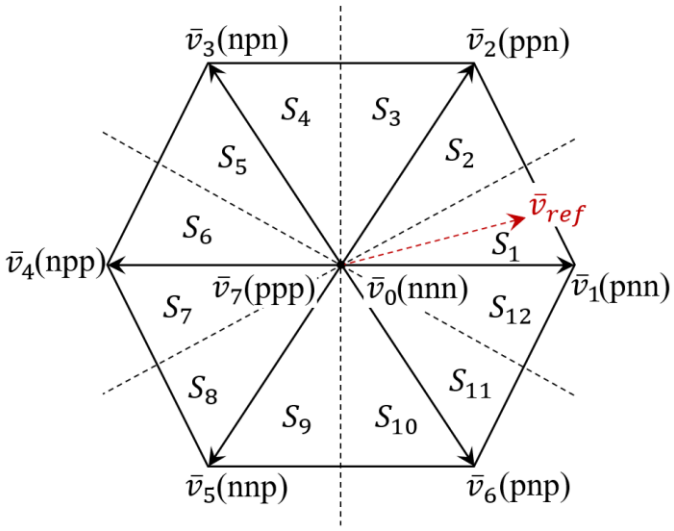
This study will utilize the symmetrical Space Vector Modulation due to its simple implementation [12, 141]. The rest of this section will deliver a joint-simulation scheme which helps to implement the NN-aided motor drive power loss optimization.



(a) Circuit diagram of a two-level voltage source converter



(b) Standard space vector diagram with 6 sectors and 8 switching states



(c) Modified space vector diagram with 12 sectors for the DPWM1 modulation.

Figure 6.2 Circuit diagram and Modulation strategies for a two-level voltage source converter.

6.1.3 Joint simulation scheme

Conventionally, the motor-drive system is mainly simulated on a system-level platform (for example, MATLAB/Simulink, PLECs), which does not consider the geometrical parameters of components. In contrast, the proposed joint simulation takes PMSM FEA model into account thus it can regard the detailed PMSM geometries as the design parameters for a motor-drive optimization problem. Furthermore, the PMSM FEA model is utilized here to generate the accurate motor key parameters (which are feedforward to PLECs for the system-level simulation) and performance (for example motor mass, power losses). Lastly, since the key role of this study is to establish a NN-based surrogate model of power loss correction linking the analytical model to the detailed simulation model, the data from both analytical model and simulation model are required for the NN training. To this end, the proposed joint simulation scheme integrates and exercises all the required models using MATLAB codes, which can collect the desired data efficiently and automatically.

The diagram of the proposed joint simulation scheme is depicted in Figure 6.3. First, the motor-drive analytical model is built on the MATLAB [41, 42, 55]. For each design point, the analytical model can be run based on the inputs of parameters (for example, DC voltage, motor current, geometry, materials) and the performance of both converter and PMSM can be estimated [41]. Then, PMSM FEA model is exercised based on the same motor input parameters, and the motor power loss (and other performance indices) can be obtained by FEA. At the same time, some key parameters of PMSM, which are required in the motor-drive system-level simulation, can be extracted from the FEA model and then feedforward to the motor-drive system on PLECs. These parameters include Torque reference (T), dq-axis inductance (L_d, L_q), flux linkage established by magnets ($flux$), DC cable resistance (R) and the motor inertia. Moreover, for the converter subsystem, the basic operation input parameters (i.e. input power, motor electrical frequency, DC voltage, and switching frequency) are also the same with the MATLAB analytical model. Finally, run the entire drive system and the converter power loss data can be obtained. It is noted, in order to run the PLECs model (together with other models) using MATLAB codes and collect the training data easily, this PLECs model is embedded into Simulink platform in this study.

As discussed Section 5.1, analytical design methods pursue computational efficiency based on simplifications and assumptions, thus they are prone to lose the estimation accuracy compared with detailed simulation models. Most of factors that could generate power loss are not considered in analytical methods (for example, non-uniformed flux distributions in stator tooth or yoke). Moreover, analytical models often use linearization techniques to simplify the loss calculation, however practically, the power losses within a PMSM drive also include losses due to the non-linear characteristics.

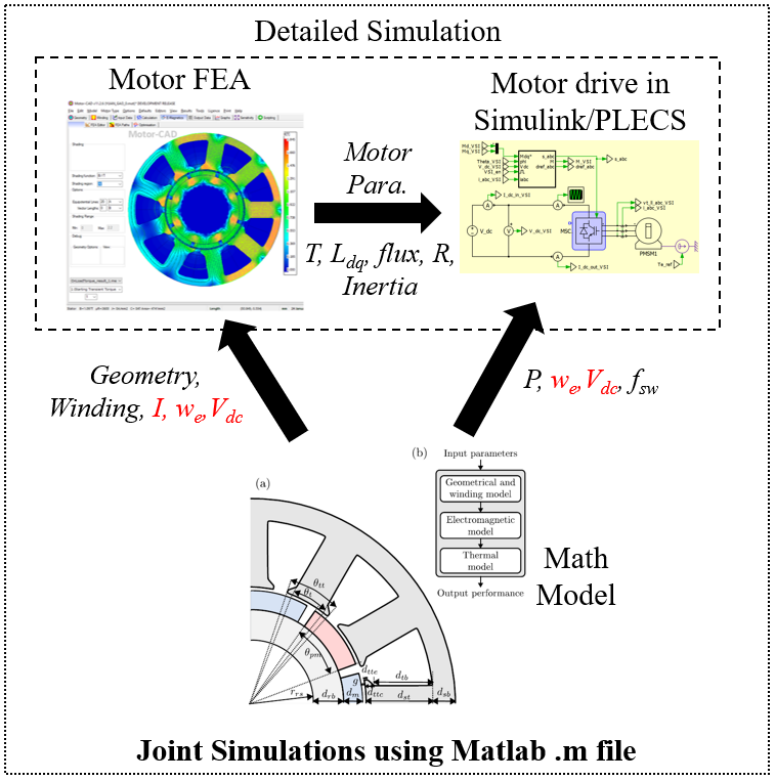


Figure 6.3 Joint simulation scheme of motor drive system.

To this end, this study proposes to use a NN to bridge the gap between analytical loss models and the detailed simulations (FEA + drive circuit system). The NN-aided method will not only correct the analytical results with high accuracy but also leave the complicate simulations of motor drive in the power loss optimization. Though the analytical basic models are just simple equations, the dedicated NN-based correction model can effectively and efficiently link them to the joint simulation in a design space. After the NN training, the hybrid model of the trained NN (for correction) and simple analytical models can quickly predict the motor-drive losses as accurate as the detailed simulations in the entire design space. Based on that, a global optimization process of drive systems can be quickly completed which generates the best design point with the minimum power loss of this drive system.

The following sections will first introduce the general research thinking of data collection and NN training (Section 6.2); then, the analytical loss estimation methods in the motor drive is briefly discussed (Section 6.3); after that, the proposed NN correction

approach is given in Section 6.4 with the technical details; finally, a case study is illustrated in Section 6.5.

6.2 General research thinking

Generally, this work is to train a dedicated correction surrogate model by a neural network whose target is linking analytical models to the detailed simulation models. In order to implement that, a small number of sample data should be collected from the joint simulation. after that, the surrogate NN can be efficiently trained by these samples. Furthermore, the trained NN will be tested by the new data collected from another round of joint simulation for the NN model validation. The new round of joint simulation will generate new data for NN validation. If the NN predictions are very close to the simulation data at these new samples, the validation is successful. Otherwise, the validation fails.

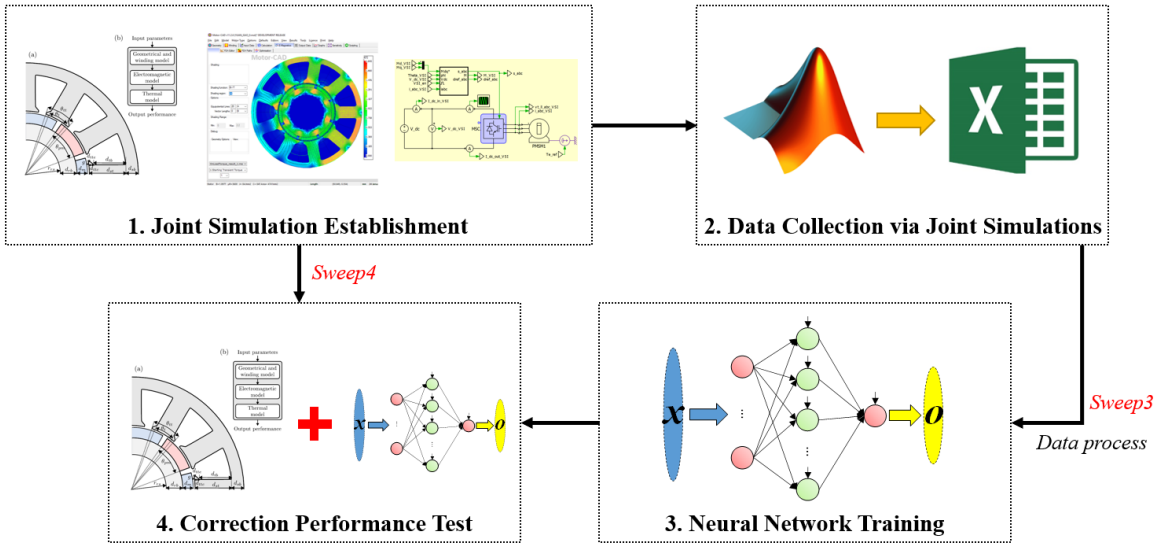


Figure 6.4 Research diagram of motor-drive power loss correction based on the neural network and the joint simulation scheme.

The diagram of the motor-drive power loss correction is depicted in Figure 6.4. Details of the four steps in this approach are discussed as follows:

1) The first step is the joint simulation establishment. As discussed in the last subsection, MATLAB analytical model is used to feedforward parameters to two different simulation platforms (MotorCAD for PMSM loss analysis, PLECs for converter loss analysis). Noting that most of the parameters delivered to MotorCAD (geometries, windings, input current etc.) do not need be delivered to PLECs (e.g. PM height, slot fraction, motor axial length), because the MotorCAD model is the detailed finite-element analysis (FEA) model of PMSM while the PLECs is a high-level simulation model of the motor drive. Moreover, simulate MotorCAD first before running the PLECs because the key parameters of PMSM in MotorCAD should be feedforwarded to the PLECs, including Torque reference (T), dq-axis inductance (L_d, L_q), flux linkage established by magnets ($flux$), DC resistance (R) and the motor inertia.

2) The second step is the data collection via the established joint simulation. At the beginning, a design space with one or several variables should be assumed for a motor-drive power-loss optimization problem. Then, sample at least two values (i.e. upper/lower boundary) for each design variable. For example, in the case study (Section 6.5), there are 3 design variables and 3 values are sampled for each variable during the first round of joint simulation. Therefore, there are totally $3^3 = 27$ design points. Since 3 different values are sampled for each variable, this sampling scheme is named by "sweep3". For each design point, the joint simulation should be run for one time and the corresponding power loss results of both analytical and simulation models can be obtained. Therefore, the second step of the motor-drive power loss correction is mainly running the joint simulation based on a sampling scheme in the design space and collected the detailed power loss data. The details of "sweep3" sampling scheme will be introduced in Section 6.5.

3) The third step is NN training part after getting the training data from the joint-simulation models. The power losses of different components are considered in the joint simulation: the PM loss, stator (York and Tooth) iron loss, and copper loss in PMSM, as well as the conduction loss, switching loss in the converter devices (i.e. six IGBTs and six Diodes). That means all the analytical models are needed and all the loss values should be collected from the joint simulation (with regards to design variables). However, in the NN training stage for loss correction, all these power losses were summated up to generate the whole power loss of the motor drive, because the entire drive loss is the unique optimization objective in this chapter. It is noted that, the same with Chapter 5, the NN here is not designed mapping from design variables directly to power loss, but to the power loss correction factor. Therefore, after collecting the raw data from joint simulation, the correction factor values should be calculated for all the sampled design points. These values are designed as the output of the NN-based surrogated model, see Section 6.4 for more details.

4) The final step is to validate/test the trained NN using another round of joint simulation. In this step, it is suggested using new data as much as possible thus the sampling scheme should be different from that in Step 2. In the case study (Section 6.5), 4 different values are sampled for each variable (named as "sweep4") and the second round of joint simulation is then exercised. Under this sampling scheme, for each design point, the loss estimation from the analytical model need to be corrected by the corresponding correction factor (given by NN prediction); after that, the corrected loss value should be compared with the collected simulation date to see whether the error is acceptable. If the error at every design point is very small (for example, relative error smaller than 3%), the validation is successful. Finally, this validated NN

can be used to do the motor-drive power loss optimization using the exhausted algorithm, the same approach with the PMSM optimization study in Chapter 5.

In summary, this study first utilizes the joint simulation models to generate the analytical loss data and the target simulation data (using the joint simulation scheme). Based on the sample date collection, the input-output training data set can be processed and provided for the NN training. After that, a dedicated NN can be trained in a very short time because the sample number can be very small (usually less than 100 for 3 or less variables). Using this trained NN, the analytical predicted data can be corrected effectively in the entire design space. In addition, the prediction process will be extremely fast due to the fact both NN model and the analytical model are based on simple mathematic equations.

6.3 Math and simulation models of motor drive system

The PMSM math models used in this chapter are the same with that in Chapter 5. In addition, for the target data, MotorCAD is also used as the FEA platform which generates the PMSM power loss target data in this Chapter. Differently, this chapter only considers one objective in the studied motor-drive optimization, drive total power losses. Other motor-drive performance (for example, mass, temperature, reliability) can be considered in the future studies but, the general idea of the NN-aided approach for the performance correction should be the same.

For converter power losses, semiconductor switching devices produce the largest portion which is typically divided into conduction and switching losses. Conduction losses occur during device on-state and are a function of its on-state resistance, while switching losses occur during device on-off switching transitions and are due to its parasitic capacitance and inductance. The non-punch through concept with field-stop layer and trench-gate structure of Infineon IGBT4 devices enables higher current densities, lower conduction and switching losses, and higher breakdown voltages to be

achieved, as compared to planar gate structures. For the mentioned reasons, in this thesis, Infineon silicon IGBT-T4 chips and diodes are considered as the reference device technology for the studied two-level inverter (six IGBTs + six diodes) [142].

Generally, conduction losses are calculated by voltage drop, on-state resistance, device average and RMS currents over each fundamental period. To model the device on-state characteristics, a linearized fitting is performed on the voltage-current data from manufacture to give an approximate forward voltage drop $V_{f,VSI}$ and on-state resistance R_{on} parameter. Then, conduction loss can be given as:

$$P_{cond} = V_{f,VSI} I_{avg,VSI} + R_{on} I_{rms,VSI}^2 \tag{6.1}$$

where $I_{avg,VSI}$ and $I_{rms,VSI}$ are average and RMS squared currents over each fundamental period. Noting that, conduction loss is a function of on-state resistance for MOSFET, not for IGBT; Therefore, here Eq. (6.1) is only used to compute the conduction loss of diodes, not IGBT.

Switching loss is the energy dissipated when the device voltage and current overlap during switching transitions. For IGBTs, both turn-on and turn-off energies, are significant, but for diodes, turn-on energy is negligible and only turn-off energy needs to be considered [142]. For both IGBTs and diodes, dissipated energies are given as the product of current dependency energy (by lookup table) and voltage dependency energy (by empirical coefficients). Assuming a constant junction temperature, the following equations are used to model the blocking voltage dependency of the device switching energies:

$$E_{sw,S}(i,v) = [E_{on,S}(i) + E_{off,S}(i)] \cdot (\frac{v}{V_{ref}})^{K_{v,S}} \tag{6.2}$$

$$E_{sw,D}(i,v) = E_{on,D}(i) \cdot (\frac{v}{V_{ref}})^{K_{v,D}} \tag{6.3}$$

where $E_{sw,S}$ represents the IGBT switching energy and $E_{sw,D}$ represents diode switching energy; $E_{on,S}$, $E_{off,S}$ and $E_{on,D}$ are all the current dependency energy, they can be directly interpolated from manufacturer provided data, based on the current going through (i); V_{ref} is the rated blocking voltage; K_v is the blocking voltage dependency coefficients. $K_{v,S}$ and $K_{v,D}$ are set as 1.2 and 0.6, respectively, according to the Semikron manual [142]. Based on these two equations, the final averaged device switching losses are computed by the switching frequency f_{sw} and the integration of dissipated energies ($E_{sw,S}$, $E_{sw,D}$) in a fundamental period [12, 53]:

$$P_{sw} = f_{sw} \int_0^{2\pi} E_{sw}(i(\theta), V_{dc}) d\theta \tag{6.4}$$

For the target data generation, in the PLECs of joint simulation, IGBT and diode characteristics are modelled and integrated in the Infineon FS50R12KT4B15 power module (*.xml* file) [142] for the two-level converter. This module can generate both conduction and switching losses at real-time based on the simulated voltage/current values. In this thesis, the average power losses of converter (six IGBTs and six diodes) during one time period are collected for each design point of the motor-drive optimization problem.

6.4 NN-based correction approach

This study also utilizes the widely spread machine learning technique, neural network (NN), to link analytical models to joint simulations for the motor drive loss estimation. The trained NN will become a surrogate model which maps loss correction factor from design variables (DVs). As discussed in Section 2.4.2, NN is a globally generalized regression model based on the simple mathematic operations between neurons (for example, multiplication, summation, sigmoid function). Unlike the fitting models (e.g. polynomial, exponential), the functional relationships between the NN model inputs-outputs do not need to be predefined, because the powerful NN will learn it

automatically by using only several internal parameters (which include weights between neurons and biases at each neuron).

As shown in Figure 6.5, the NN in this chapter is straightforward mapping DVs to the correction factor of the total motor-drive loss (k_{PL_t}). k_{PL_t} can be simply obtained: dividing the target data by the corresponding analytical data. This network comprises of one input layer, one hidden layer and one output layer. The number of neurons in the input layer equals the number of DVs while only one output neuron in the final layer. Regarding the neuron number (q) in the hidden layer, it can be set by a NN algorithm developer. However, it is simply set as 6 in this motor-drive study. The reason is that there are only 27 sample points in NN training thus good regression performance can be easily obtained with 6 hidden neurons, which benefits from the good global generalization capability of NN.

Before NN training, similar with the procedure in Section 5.3 (a PMSM multi-objective optimization problem), DVs need to be confirmed from all the design parameters, also the sample steps of these DVs should be determined to collect the sample data, i.e. DVs and power loss values from analytical models and simulation models. In addition, as discussed above, the joint simulation should be well established for the following data generation and NN training. After collecting the desired power-loss data from both analytical models and the joint-simulation models, the correction factors (k_{PL_t}) for these samples can be easily computed with regards to DVs using division. The NN can then be trained in MATLAB by the input-output data matrix: DV-factor. After training the NN model, another round of joint simulation should be exercised and, some simulation data (not included in the training data set) will be utilized for the NN model verification (see Figure 6.4). Lastly, for the power loss optimization, the trained NN and analytical models (i.e. the hybrid model) will be used to effectively give the loss value in the entire design space, which is using the exhaustive algorithm to do the global optimization.

Therefore, the general NN-aided optimization method is similar to that of Chapter 5; the main differences locate at the target data collection and the NN correction model design. In Chapter 5, only a subsystem of motor drive system, PMSM, is considered thus, only PMSM FEA model is utilized to generate the target data for the NN-based correction. However, in this chapter, the motor-drive system comprises of a two-level converter and a PMSM; therefore, for the target data collection, a joint simulation scheme is proposed to combine the PMSM FEA model with the system-level simulation. On the other hand, two different NNs are designed for the PMSM performance correction in Chapter 5 since there are two optimization objectives. In this chapter, only one objective (total motor-drive power loss) is considered thus one NN with one output is designed. Besides, for the NN training in the case studies, 64 samples were collected in Chapter 5 while only 27 samples were collected in this chapter, though both chapters consider 3 design variables for the case studies.

The next section will give a case study to validate the proposed NN aided approach.

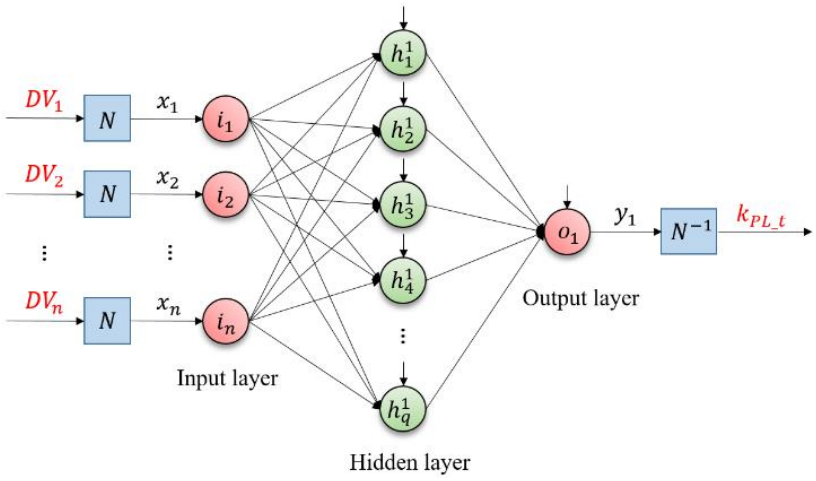


Figure 6.5 Deployment of the NN surrogate model for the motor-drive power loss study.

6.5 Motor drive case study

This section will mainly discuss two aspects in the NN-based motor-drive optimization case study: (1). NN training and validation; (2). NN-based power loss optimization.

Obviously, after the sample data collection, the first aspect would choose a NN structure and train a NN; if the trained NN could satisfy the validation condition (for example, the general relative error is smaller than 1%), it can be used for the following optimization work. Then, in the NN-based optimization work, the process would be the same with the Section 5.3, Chapter 5: sample large number of points in the design space and use the math models to generate all the corresponding outputs, the optimal design can thereby be obtained by a *min* function.

6.5.1 NN training and validation for power loss correction

In order to demonstrate the excellent correction performance of the NN aided approach and further do the global optimization, the 12-slot-10-pole PMSM (from Chapter 5) fed by a two-level VSI case is illustrated to generate the loss correction results using NN. Three DVs are studied here: motor speed (DV1, *rpm*, [600, 2100]), DC-link voltage (DV2, V_{dc}, [200, 310] *V*) and PMSM Q-axis current (DV3, I_q, [10, 18] *A*). Other parameters in this case can be found in Table 5.3. The design ranges and parameters are given according to the Reference [12], which is for the aerospace application. However, as mentioned, the proposed NN aided approach should be independent with the selection and settings of parameter/variable.

The analytical PMSM sizing model in [42] is used as the basis of this study, which is a non-iterative and high dimensional sizing model which can efficient provide a comprehensive design of PMSM. As discussed in Section 6.2, this thesis will use the afore-mentioned NNs to mainly link the estimated losses of this sizing model to the joint-simulation target values (from MotorCAD & PLECs software) in a desired design space.

It is hard to depict the variations of loss data or correction performance against three variables. Therefore, before showing the correction relative errors of NN, the series numbers of design points for 3-DV combinations are defined in Figure 6.6. Different

from Chapter 5, in this Chapter, $3^3 = 27$ samples are utilized to train the NN and $4^3 = 64$ samples are then for NN validation.

Figure 6.6 shows the overall scheme of sweep3 and sweep4. The first and the last values of each DV are the boundary values of their ranges while the other sampled values are evenly distributed in the middle. For example, in sweep4, a1 and a4 are the lower and upper boundaries of DV1, and the distance of adjacent values equals $(a4 - a1)/3$. One the other hand, Figure 6.6 also depicts the series number of DV combinations for each sample value of DV3. Obviously, in sweep4, when a value of DV3 is confirmed, there will be 4*4 designs whose series numbers are marked as 1,2 ... , 16. And in sweep3, the series numbers are given as 1,2 ... , 9 for every value of DV3.

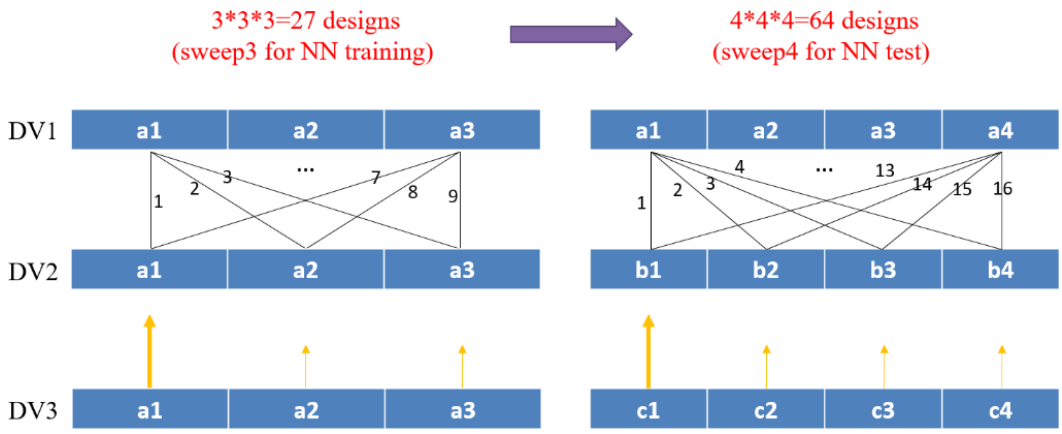


Figure 6.6 Design variable combinations for sweep3 and sweep4.

This study utilized the sweep3 samples (27 data points) to train the NN and then the sweep4 samples to test the NN correction performance. Obviously, in sweep4, half of sample points (32 points) are not included in sweep3 sampling which provides all the training data (27 samples) for NN. Relative errors of sweep4 samples (64 points), after the NN training and correction, are depicted in Figure 6.7. All 64 absolute errors are smaller than 0.3% and the root mean square error (RMSE) is only 0.108%, extremely close to the target data. Though the overall correction error is a little larger than that of

sweep3 (RMSE is 0.027%), all the correction error results are very small and assume as acceptable.

Figure 6.7 depicts and summarizes the relative correction performance of all the samples in sweep4, but not all DV values demonstrated. The NN correction performance can be demonstrated more clearly by the total power loss variation against 3 DVs. For example, first confirm one value of DV3, then depict the loss variation (generated by the hybrid model) against DV1 and DV2, as shown in the four subfigures of Figure 6.8. Furthermore, four 3D figures fixing DV1 (*rpm*) can be found in Figure 6.9 and Figure 6.10 depicts the 3-Dimensional figures fixing DV2 (V_{dc}). It is important to notice that all the corrected performance results (coloured surface in the figure) are correct by the same analytical models and the correction surrogate NN. This NN was only trained by 27 samples but, obviously, the correction performance turned out to be very good. This effectively reflects the excellent generalization capability of NN in this motor-drive optimization case.

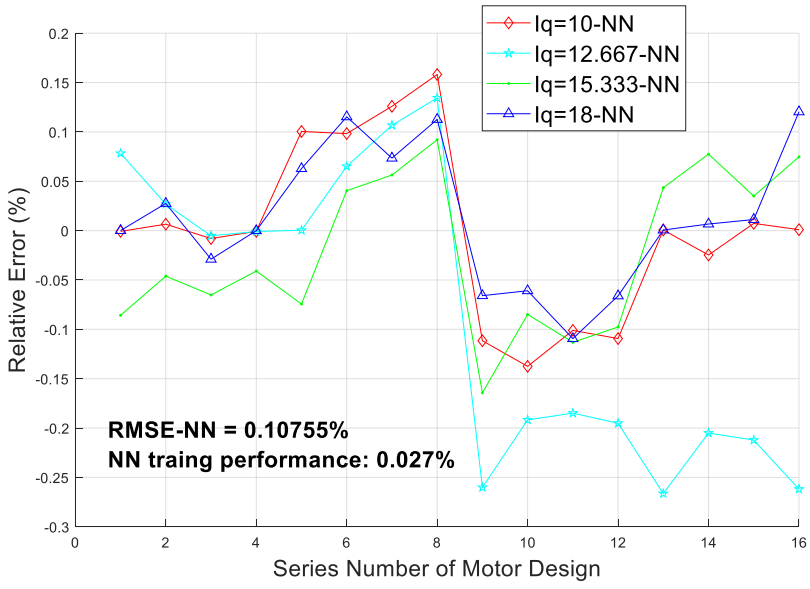
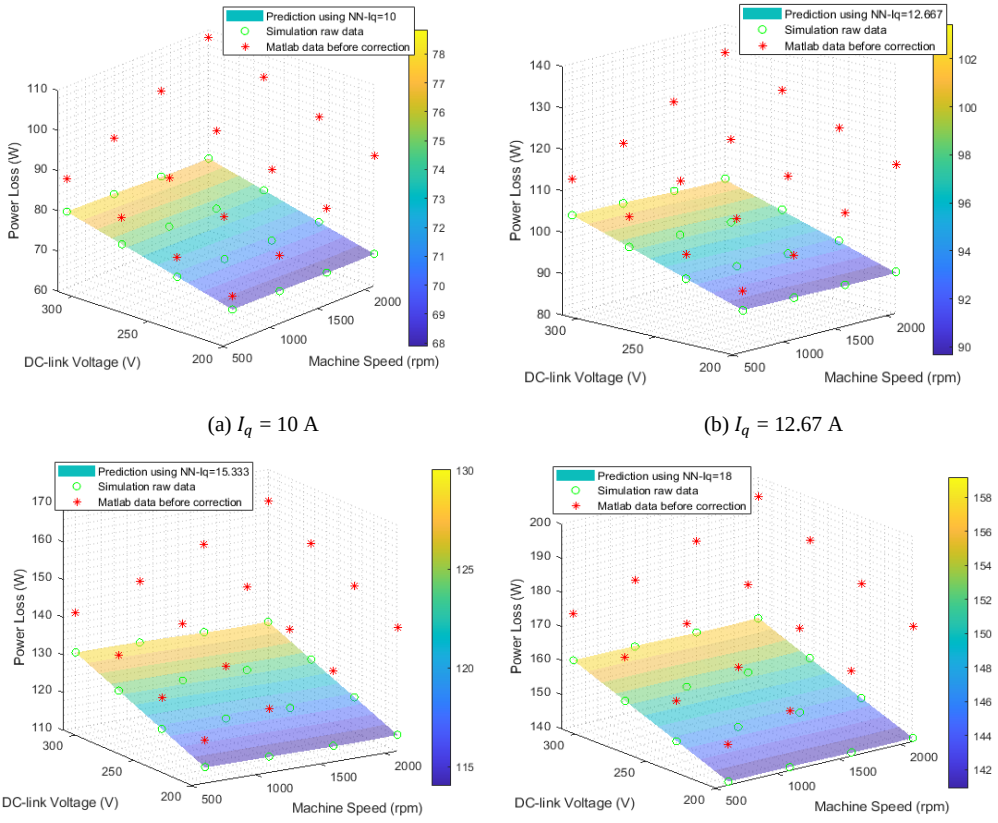


Figure 6.7 NN correction performance based on the sweep4 samples.



(a) I_q = 10 A

(b) I_q = 12.67 A

(c) I_q = 15.33 A

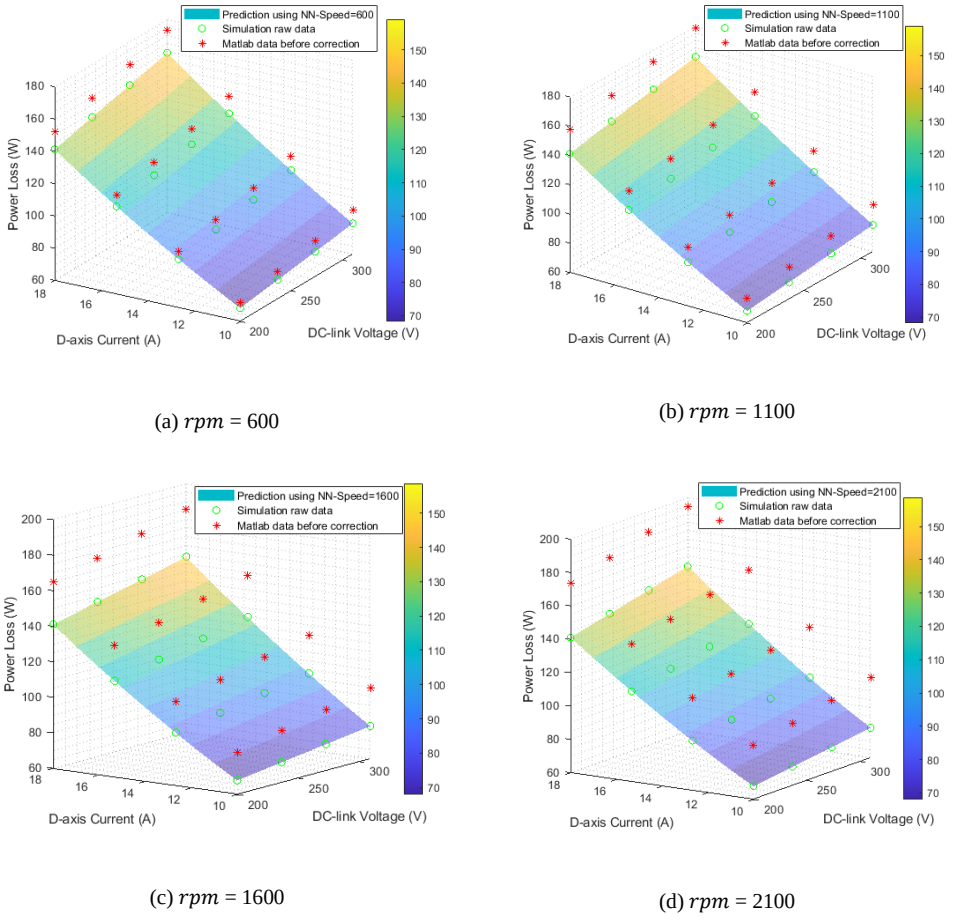
(d) I_q = 18 A

Figure 6.8 NN correction results and the sweep4 samples with four I_q values.

Sweep4 data from both analytical and joint-simulation models are also depicted in Figure 6.8. As marked by the red stars, under the same input current condition, the analytical model of motor-drive gives the dramatically increasing power losses when DV1 or DV2 climbs up while the estimation from the joint simulations does not show a clear increasement. Therefore, their gaps can become very large under certain conditions. As shown by the depicted coloured surfaces, the prediction results after NN correction can follow the joint simulation (green circles) very well under all the conditions. The surface results were generated extremely fast because analytical and NN models are all math models thereby saving much time compared with the joint simulations.

Generally, no matter for the join-simulation data or for the analytical model data in Figure 6.8, the power losses seem to have a linear relation with DV1 (*rpm*) and DV2 (V_{dc}). However, as shown in Figure 6.9, and Figure 6.10, the power losses have a clear

polynomial relation with the DV3 (I_q). The linear relations are due to the simplifications and assumptions in this study, for example the negligible harmonic hysteresis and harmonic ohmic losses, PM segmentation and flux saturation in PMSM are not accounted for. The non-linear factors can be further considered in the future. The key role of this study is to propose the NN-aided surrogate model for the correction of motor-drive analytical estimations, which is building a mapping based on the sample data and should be independent with the specific detailed analytical models or FEA models. Thus, these model simplifications and assumptions are considered acceptable.



(a) rpm = 600

(b) rpm = 1100

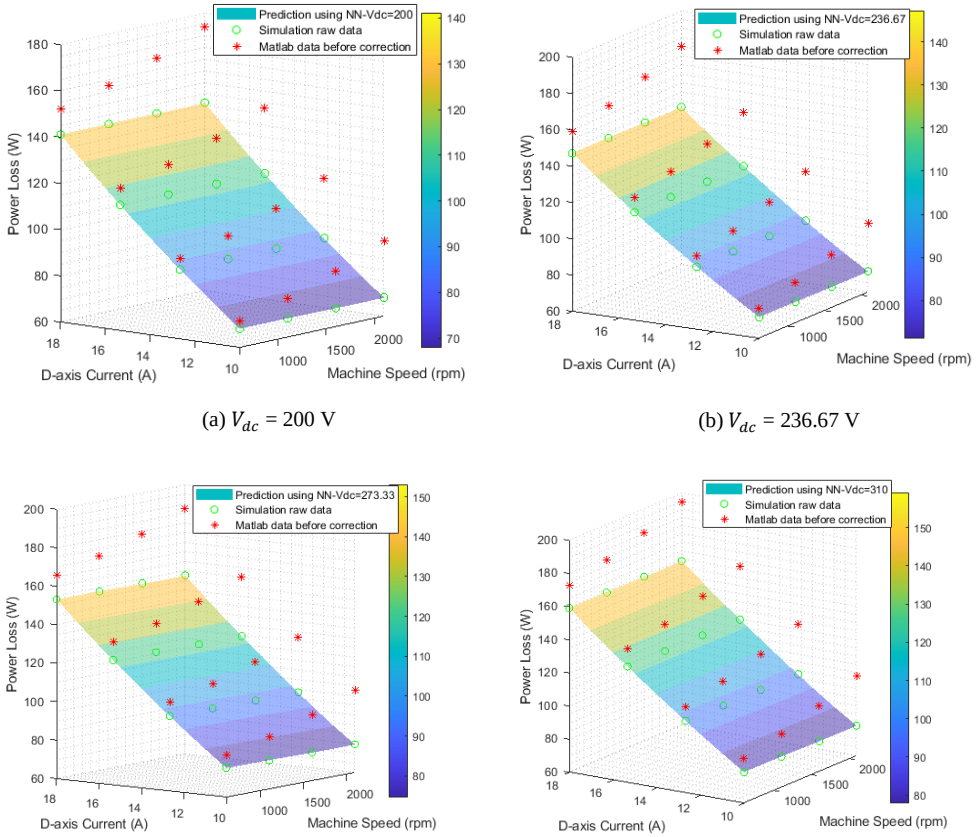
(c) rpm = 1600

(d) rpm = 2100

Figure 6.9 NN correction results and the sweep4 samples with four rpm values.

Noting that all the coloured surface results in Figure 6.8, Figure 6.9, and Figure 6.10 are generated by the same NN which is trained only using 27 data points (sweep3 scheme). Therefore, the proposed NN aided approach is a simple, efficient and effective method linking the analytical motor-drive estimation to the detailed joint simulations.

Lastly, the proposed method should not be limited for the detailed simulation data, it is also feasible for any data from practical conditions/rigs as the target samples. The basic thinking would be all the same. For the practical application, if the number of design variable is little, sweep3 even sweep4 sampling scheme can be utilized because it is possible to do the experiment for 27 times or even 64 times. However, if the variable number is large, the design of experiments (DOEs) method can be a more realistic approach for the parameter tuning and/or data collection [143].



(a) V_{dc} = 200 V

(b) V_{dc} = 236.67 V

(c) V_{dc} = 273.33 V

(d) V_{dc} = 310 V

Figure 6.10 NN correction results and the sweep4 samples with four V_{dc} values.

6.5.2 NN based optimization of drive system

Same with the terminated optimization step of Figure 5.6 in Chapter 5, for the power loss optimization of drive system, a large amount of data was sampled for the global optimization work. However, the only difference is that, the optimization here is a one-objective searching process thus, there is no extra index/weight for the final

sorting. As summarized in Table 6.1, 31 values of rpm, 12 values of V_{dc}, and 9 values of I_q were sampled. The total sample number is 3348 and the collapsed time for generating all the corresponding objective values is 113.018 *secs* on a standard computer. Then, the optimal design was then simply obtained by finding the lowest PL value of drive system, which can be done in MATLAB almost instantaneously using the embedded *min* function.

After finding the best design point, for the NN-approach validation, it was used as the inputs to simulate the detailed joint-simulation (one design costs more than 1.5 *mins*). As shown in Table 6.1, the joint-simulation power loss is very close to what NN and math model predicted: relative error is only +0.037%, which is in the same level of NN training performance (see Figure 6.7). Therefore, this motor-drive case once again demonstrates the efficiency, accuracy and practicability of the NN surrogate model approach.

Noting that, the minimum FEA *PL* value in the data collection (27 samples) is 67.982 *W* (Row 2, Column 6 of Table 6.1). After the NN-based optimization, the FEA validation value of the best design is smaller, 67.8527 *W* (Row 4, Column 6 of Table 6.1). Even though the difference is not much in this case study, it can be very big for the cases in a large motor-drive design space. In summary, using a very small amount of FEA-based sample collection, the proposed NN can be easily trained and analytical-NN-hybrid model can smoothly give a better design point in the design space which is not be one of the collected samples.

Table 6.1. Design space and optimal design of motor drive system

	rpm	V_{dc} (*V*)	I_q (*A*)	*PL*, NN aided approach (*W*)	*PL*, FEA (*W*)
Design Range	[600, 2100]	[200, 310]	[10, 18]	**[67.8776, 159.1784]**	**[67.982, 159.1702]**
Sampling Step	50	10	1	-	-

Best Design	1850	200	10	**67.8776 (+0.037%)**	**67.8527**

6.6 Chapter Summary

In Chapter 3 and Chapter 5, design and optimization problems of an input dc filter and a PMSM were studied separately, which are two subsystems in a motor-drive system onboard aircraft. In this chapter, the continued optimization study focuses on the entire motor drive system which comprises of an inverter and a PMSM. The same with Chapter 5, only the neural network (NN) algorithm is used for the motor-drive optimization work. The NN was trained as a dedicated surrogate model for the performance correction of analytical motor-drive models. The correction target data were from a joint simulation which includes a PMSM finite-element model and the drive system simulation. With the help of the NN surrogate, the outputs of analytical drive system models can be corrected to be extremely close to the corresponding target data. Based on that, the best design of a motor-drive optimization problem can be quickly found in the entire design space.

Chapter 7 Conclusion and Future Study

This chapter briefly reviews the most important findings of this thesis and then suggests potential future topics to be studied in this area of research.

With an expected growth in air travel over the next decades and increasingly stringent regulations for carbon emissions, the aerospace industry is facing the challenges of flight electrification. Flight electrification is a significant way to reduce the aircraft environmental impact, improve fuel efficiency, and lower operating costs. One of challenges concerning the realization of many flight electrification concepts is to achieve light-weight and efficient electrical drives. On the one hand, this thesis investigated the design, optimization and control problems of the electrical drives in both system level and subsystem level. On the other hand, machine learning (ML) algorithms, which are increasingly attracting attentions, are used to study both optimization and control problems of the aircraft electrical drives; they are also compared with the conventional search algorithms. Therefore, the key role of this thesis is the hybrid application of search and ML algorithms on the multi-objective optimisation and control problems for aircraft electrical drive systems, with a focus on minimising their mass and power losses.

This thesis first introduced the background and motivations of this study. Practical applications of drive systems in aerospace were also presented. The relevant works, already published by others in this target area, are given in Chapter 2: design, optimization and control of aircraft electrical drive systems. Based on that, a simple category strategy, search algorithm and surrogate algorithm (2SA), was proposed to classify and compare the commonly used algorithms, with an emphasize on the differences and interactions between two categories.

Then, in Chapters 3, 5, and 6 of this thesis, three optimization-based design cases were presented one by one, introducing the optimization works of a filter subsystem, a motor subsystem and an inverter-based motor drive system. Both algorithm categories in the "2SA" categorization strategy were investigated in the optimization works. However, in the motor and motor-drive optimization, surrogate algorithms were used to train the surrogate performance correction models rather than providing the direct and simple mapping from parameters to system performance. These surrogate correction models can yield the fast and accurate optimization instead of the conventional time-consuming FEA-based optimization. This surrogate-aided approach was proved to be extremely efficient and have no risk of stacking on local optimum, as discussed in Section 5.5, Chapter 5. Although surrogate training requires a certain additional time for collecting the sample data, this is justified by a much faster global optimization in the execution stages of surrogate models.

Moreover, in Chapter 4, the surrogate algorithm was applied to the inverter control topic, aimed at building the surrogate imitation controllers to replace the original FCS-MPC controller but with a lower computation burden. It should be highlight that the imitation controller is not intended to improve the control performance of the original FCS-MPC algorithm but to reduce the computational burden. Two different ML controllers were designed, their strength and weakness were discussed after testing in the simulation system and experiment rig.

In summary, for the key conclusions, the thesis first presented a novel classification strategy for the commonly used algorithms. Then, both search algorithms and ML algorithms are used for the studied aircraft drive system. In addition, apart from the design & optimization problems of filter, motor, and "motor + filter" system, this thesis also studied how to use ML to address the control problem of converter.

During conduction of this research work, several aspects have been identified as follow-on studies to enhance the topic and to broaden the applications of search and ML algorithms, namely:

1) Surrogate-model-aided searching. As discussed in Section 2.4, Chapter 2, surrogate algorithm can be trained as an agent to effectively serve a search problem. The optimization works performed in this thesis are all off-line optimization which means uncertainty, noise, randomness and other factors were not considered. However, in an online optimization, some relations in the detailed systems can be expressed by a trained surrogate model thereby speeding up the searching process. Therefore, the static relations in the studied system can be represented by surrogate models which can leave substantial derivations and complex calculations. Moreover, these relations may not necessarily be the whole part of the studied problem (i.e. can be a small part), which provides the flexibility of real application. In general, for the optimization engine, the exhaustive algorithm used in the motor optimization (see Section 5.3, Chapter 5) may not be efficient enough. Therefore, search algorithms may need to be considered for the online and global optimization of the whole system.
2) Online neural-network controller. The ML-based imitation controller in Chapter 4 follows a traditional procedure: collect sample data, train ML (surrogate) models offline, test ML models online. However, NN can also be directly applied online for the current/voltage regulation with no need for the data collection and offline training. As discussed in Section 4.1, Chapter 4, this kind of controller have the potential capability to perform a better dynamic response than the conventional cascaded linear controllers. The conventional controllers usually require the parameter tuning for a specific application. However, after tuning, the selected parameters stay

unchanged thus may not be optimal for a new circuit condition. The NN-based adaptive controller has the capability to address the changes of circuit condition due to its internal mechanism of online training.

3) Reinforcement learning on aircraft electrical power systems. As mentioned, reinforcement learning (RL) is one of three basic machine learning groups concerned with how intelligent agents ought to take actions (i.e. policy of moving agents) in an environment in order to maximize the notion of cumulative reward (called utility in RL). Rewards should be predefined in a RL problem but, the policy as well as the utility (i.e. critic of environment) should be updated/learned by the samples/episodes of agent moving in the environment. On the aircraft power systems, RL algorithms could be used for the microgrid energy management and the voltage/current adaptive control.

4) In Chapter 6, for the optimization of the entire drive system, only one objective was considered. In the future, both mass and power loss can be studied as the objectives for this optimization. However, it is hard to give the mass model of the converter system. A feasible way is to assume a mass value for the converter while the motor mass will change with regards to the design variables. Based on that, the NN-aided approach should be applicable for the new optimization problems.

With the optimization-based design, control imitation topics presented in this thesis, a step of investigating the hybrid application of search and ML algorithms on the aircraft drive systems has been taken. However, up to now, online controllers using ML algorithms for the aircraft electrical power system are still rarely reported. Therefore, the main challenge that researchers still need to tackle is the ML-based control and/or the microgrid energy management onboard aircraft. This gives room for new ideas and concepts as well as further research, development and commercial design projects.

References

[1] U. government. (2019). *The Climate Change Act 2008 (2050 Target Amendment) Order 2019* [Online]. Available: https://www.legislation.gov.uk/uksi/2019/1056/contents/made.

[2] G. Gratton, A. Padhra, S. Rapsomanikis, and P. D. Williams, "The impacts of climate change on Greek airports," *Climatic Change*, pp. 1-13, 2020.

[3] Sustainable Aviation, "Decarbonisation road-Map: A path to net zero," *Sustainable Aviation*, p. 0011, 2020.

[4] P. Wheeler, T. S. Sirimanna, S. Bozhko, and K. S. Haran, "Electric/Hybrid-Electric Aircraft Propulsion Systems," *Proceedings of the IEEE*, vol. 109, no. 6, pp. 1115-1127, 2021.

[5] J. Domone. (2018). *The challenges and benefits of the electrification of aircraft* [Online]. Available: https://m.atkinsglobal.com/~/media/Files/A/Atkins-Corporate/Electrification%20White%20Paper%20-%20digital.pdf.

[6] Aerospace Technology Institute. (2018). *Insight_07 electrical power systems* [Online]. Available: https://www.ati.org.uk/media/ntlocbb4/insight_07-electrical-power-systems.pdf.

[7] J. Rosero, J. Ortega, E. Aldabas, and L. Romeral, "Moving towards a more electric aircraft," *IEEE Aerospace and Electronic Systems Magazine*, vol. 22, no. 3, pp. 3-9, 2007.

[8] B. Sarlioglu and C. T. Morris, "More electric aircraft: Review, challenges, and opportunities for commercial transport aircraft," *IEEE transactions on Transportation Electrification*, vol. 1, no. 1, pp. 54-64, 2015.

[9] P. Wheeler, "Technology for the more and all electric aircraft of the future," in *2016 IEEE International Conference on Automatica (ICA-ACCA)*, 2016: IEEE, pp. 1-5.

[10] P. Wheeler and S. Bozhko, "The more electric aircraft: Technology and challenges," *IEEE Electrification Magazine*, vol. 2, no. 4, pp. 6-12, 2014.

[11] X. Roboam, B. Sareni, and A. De Andrade, "More electricity in the air: Toward optimized electrical networks embedded in more-electrical aircraft," *IEEE industrial electronics magazine*, vol. 6, no. 4, pp. 6-17, 2012.

[12] B. Cheong, "Advanced Modelling and Evolutionary Multiobjective Optimization of Motor Drive for Aerospace Applications," Doctor of Philosophy, Department of Electrical and Electronic Engineering, University of Nottingham, Nottingham, United Kingdom, 2019.

[13] C. Anghel, "A novel start system for an aircraft auxiliary power unit," in *Collection of Technical Papers. 35th Intersociety Energy Conversion Engineering Conference and Exhibit (IECEC)(Cat. No. 00CH37022)*, 2000, vol. 1: IEEE, pp. 7-11.

[14] M. Liu and Y. Zhou, "The reliability prediction of an electro-mechanical actuator of aircraft with the hybrid redundant structure," in *2008 IEEE Vehicle Power and Propulsion Conference*, 2008: IEEE, pp. 1-5.

[15] M. Sinnett, "787 no-bleed systems: saving fuel and enhancing operational efficiencies," *Aero Quarterly*, vol. 18, pp. 6-11, 2007.

[16] Liebherr-Aerospace Lindenberg GmbH. (2019). *Flight control and actuation systems* [Online]. Available: https://www.liebherr.com/en/usa/products/aerospace-and-transportation-systems/aerospace/products-and-solutions/flight-control-and-actuation-systems/flight-control-and-actuation-systems.html

[17] Safran landing systems. (2020). *Boeing 787 Dreamliner brake* [Online]. Available: https://www.safran-landing-systems.com/wheels-and-brakes/products/boeing-787-dreamliner-brake.

[18] Pipstrel d.o.o. Ajdovščina. (2016). *Most powerful hybrid electric powertrain powers up* [Online]. Available: http://www.hypstair.eu/most-powerful-hybrid-electric-powertrain-powers-up/.

[19] Siemens. (2016). *Electric motor from Siemens sets new world climb record* [Online]. Available: https://press.siemens.com/global/en/pressrelease/electric-motor-siemens-sets-new-world-climb-record.

[20] T. Dragičević, "Dynamic stabilization of DC microgrids with predictive control of point-of-load converters," *IEEE Transactions on Power Electronics*, vol. 33, no. 12, pp. 10872-10884, 2018.

[21] J. M. Carrasco *et al.*, "Power-electronic systems for the grid integration of renewable energy sources: A survey," *IEEE Transactions on industrial electronics*, vol. 53, no. 4, pp. 1002-1016, 2006.

[22] T. Dragičević, X. Lu, J. C. Vasquez, and J. M. Guerrero, "DC microgrids—Part I: A review of control strategies and stabilization techniques," *IEEE Transactions on power electronics*, vol. 31, no. 7, pp. 4876-4891, 2015.

[23] F. Gao, S. Bozhko, A. Costabeber, G. Asher, and P. Wheeler, "Control design and voltage stability analysis of a droop-controlled electrical power system for more electric aircraft," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 12, pp. 9271-9281, 2017.

[24] D. Salomonsson and A. Sannino, "Low-voltage DC distribution system for commercial power systems with sensitive electronic loads," *IEEE Transactions on Power Delivery*, vol. 22, no. 3, pp. 1620-1627, 2007.

[25] M. Monadi, M. A. Zamani, J. I. Candela, A. Luna, and P. Rodriguez, "Protection of AC and DC distribution systems Embedding distributed energy resources: A comparative review and analysis," *Renewable and sustainable energy reviews*, vol. 51, pp. 1578-1593, 2015.

[26] F. Gao and S. Bozhko, "Modeling and impedance analysis of a single DC bus-based multiple-source multiple-load electrical power system," *IEEE Transactions on Transportation Electrification*, vol. 2, no. 3, pp. 335-346, 2016.

[27] F. Gao, X. Zheng, S. Bozhko, C. I. Hill, and G. Asher, "Modal analysis of a PMSG-based DC electrical power system in the more electric aircraft using eigenvalues sensitivity," *IEEE Transactions on Transportation Electrification*, vol. 1, no. 1, pp. 65-76, 2015.

[28] Y. Gao, T. Yang, S. Bozhko, P. Wheeler, and T. Dragičević, "Filter Design and Optimization of Electromechanical Actuation Systems Using Search and Surrogate Algorithms for More-Electric Aircraft Applications," *IEEE Transactions on Transportation Electrification*, vol. 6, no. 4, pp. 1434-1447, 2020.

[29] Y. Gao, T. Yang, T. Dragičević, S. Bozhko, P. Wheeler, and C. Zheng, "Optimal filter design for power converters regulated by FCS-MPC in the MEA," *IEEE Transactions on Power Electronics*, vol. 36, no. 3, pp. 3258-3268, 2020.

[30] H. Ounis, B. Sareni, X. Roboam, and A. De Andrade, "Multi-level integrated optimal design for power systems of more electric aircraft," *Mathematics and Computers in Simulation*, vol. 130, pp. 223-235, 2016.

[31] Z. Ma, X. Zhang, J. Huang, and B. Zhao, "Stability-constraining-dichotomy-solution-based model predictive control to improve the stability of power conversion system in the MEA," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 7, pp. 5696-5706, 2018.

[32] Military Standard, "Aircraft electric power characteristics," *Department of Defense Interface Standard (MIL-STD-704F)*, 2004.

[33] X. Roboam, "New trends and challenges of electrical networks embedded in “more electrical aircraft”," in *2011 IEEE International Symposium on Industrial Electronics*, 2011: IEEE, pp. 26-31.
[34] J. Muhlethaler, M. Schweizer, R. Blattmann, J. W. Kolar, and A. Ecklebe, "Optimal design of LCL harmonic filters for three-phase PFC rectifiers," *IEEE Transactions on Power Electronics*, vol. 28, no. 7, pp. 3114-3125, 2012.
[35] Y. Jiao and F. C. Lee, "LCL filter design and inductor current ripple analysis for a three-level NPC grid interface converter," *IEEE Transactions on Power Electronics*, vol. 30, no. 9, pp. 4659-4668, 2014.
[36] A. Kouchaki and M. Nymand, "Analytical design of passive LCL filter for three-phase two-level power factor correction rectifiers," *IEEE Transactions on power electronics*, vol. 33, no. 4, pp. 3012-3022, 2017.
[37] A. Berzoy, A. Mohamed, and O. Mohammed, "Design considerations and predictive direct current control of active regenerative rectifiers for harmonic and current ripple reduction," in *2016 IEEE Applied Power Electronics Conference and Exposition (APEC)*, 2016: IEEE, pp. 928-935.
[38] T. A. dos Santos Barros, P. J. dos Santos Neto, P. S. Nascimento Filho, A. B. Moreira, and E. Ruppert Filho, "An approach for switched reluctance generator in a wind generation system with a wide range of operation speed," *IEEE Transactions on Power Electronics*, vol. 32, no. 11, pp. 8277-8292, 2017.
[39] C. Gerada, M. Galea, and A. Kladas, "Electrical machines for aerospace applications," in *2015 IEEE Workshop on Electrical Machines Design, Control and Diagnosis (WEMDCD)*, 2015: IEEE, pp. 79-84.
[40] B. Cheong, P. Giangrande, M. Galea, P. Zanchetta, and P. Wheeler, "Integrated motor drive design for weight optimization," in *2017 IEEE Energy Conversion Congress and Exposition (ECCE)*, 2017: IEEE, pp. 816-823.
[41] B. Cheong, P. Giangrande, X. Zhang, M. Galea, P. Zanchetta, and P. Wheeler, "System-Level Motor Drive Modelling for Optimization-based Designs," in *2019 21st European Conference on Power Electronics and Applications (EPE'19 ECCE Europe)*, 2019: IEEE, pp. P. 1-P. 9.
[42] B. Cheong, P. Giangrande, X. Zhang, M. Galea, P. Zanchetta, and P. Wheeler, "Fast and Accurate Model for Optimization-based Design of Fractional-Slot Surface PM Machines," in *2019 22nd International Conference on Electrical Machines and Systems (ICEMS)*, 2019: IEEE, pp. 1-6.
[43] P. Giangrande *et al.*, "Considerations on the development of an electric drive for a secondary flight control electromechanical actuator," *IEEE Transactions on Industry Applications*, vol. 55, no. 4, pp. 3544-3554, 2019.
[44] P. Giangrande, V. Madonna, G. Sala, A. Kladas, C. Gerada, and M. Galea, "Design and testing of PMSM for aerospace EMA applications," in *IECON 2018-44th Annual Conference of the IEEE Industrial Electronics Society*, 2018: IEEE, pp. 2038-2043.
[45] M. M. AG. (2019). *Taking to the high skies with maxon motors* [Online]. Available: https://www.maxongroup.com/maxon/view/application/Taking-to-the-high-skies-with-maxon-motors.
[46] R. Krishnan, D. Blanding, A. Bhanot, A. Staley, and N. Lobo, "High reliability SRM drive system for aerospace applications," in *IECON'03. 29th Annual Conference of the IEEE Industrial Electronics Society (IEEE Cat. No. 03CH37468)*, 2003, vol. 2: IEEE, pp. 1110-1115.
[47] T. A. Lipo, *Introduction to AC machine design*. John Wiley & Sons, 2017.
[48] Y. Gao, T. Yang, X. Wang, S. Bozhko, and P. Wheeler, "Machine Learning Based Correction Model in PMSM Power Loss Estimation for More-Electric Aircraft Applications," in *2020 23rd International Conference on Electrical Machines and Systems (ICEMS)*, 2020: IEEE, pp. 1940-1944.
[49] S. G. Min, G. Bramerdorfer, and B. Sarlioglu, "Analytical modeling and optimization for electromagnetic performances of fractional-slot PM brushless

machines," *IEEE Transactions on Industrial Electronics,* vol. 65, no. 5, pp. 4017-4027, 2017.

[50] J. H. Lee, J.-W. Kim, J.-Y. Song, Y.-J. Kim, and S.-Y. Jung, "A novel memetic algorithm using modified particle swarm optimization and mesh adaptive direct search for PMSM design," *IEEE Transactions on Magnetics,* vol. 52, no. 3, pp. 1-4, 2015.

[51] J. Bao, J. Xing, Y. Luo, and P. Zheng, "Multi-objective shape optimization of Permanent Magnet Synchronous Motor based on Kriging surrogate model and design domain reduction," in *2019 22nd International Conference on Electrical Machines and Systems (ICEMS)*, 2019: IEEE, pp. 1-4.

[52] G. Bramerdorfer and A.-C. Zăvoianu, "Surrogate-based multi-objective optimization of electrical machine designs facilitating tolerance analysis," *IEEE Transactions on Magnetics,* vol. 53, no. 8, pp. 1-11, 2017.

[53] B. Cheong, P. Giangrande, X. Zhang, M. Galea, P. Zanchetta, and P. Wheeler, "Fast and accurate multi-physics model for optimization-based design of VSBBC," in *IECON 2019-45th Annual Conference of the IEEE Industrial Electronics Society*, 2019, vol. 1: IEEE, pp. 1762-1767.

[54] I. Laird, X. Yuan, J. Scoltock, and A. J. Forsyth, "A design optimization tool for maximizing the power density of 3-phase DC–AC converters using silicon carbide (SiC) devices," *IEEE Transactions on Power Electronics,* vol. 33, no. 4, pp. 2913-2932, 2017.

[55] B. Cheong, P. Giangrande, X. Zhang, M. Galea, P. Zanchetta, and P. Wheeler, "Evolutionary Multiobjective Optimization of a System-Level Motor Drive Design," *IEEE Transactions on Industry Applications,* vol. 56, no. 6, pp. 6904-6913, 2020.

[56] S. Kukkonen and J. Lampinen, "GDE3: The third evolution step of generalized differential evolution," in *2005 IEEE congress on evolutionary computation*, 2005, vol. 1: IEEE, pp. 443-450.

[57] Airbus Directives (ABD) and Procedures, "ABD0100-Equipment-Design, General Requirements For Suppliers. Issue C," ed: Airbus, 1998.

[58] H. M. Kim, N. F. Michelena, P. Y. Papalambros, and T. Jiang, "Target cascading in optimal system design," *J. Mech. Des.,* vol. 125, no. 3, pp. 474-480, 2003.

[59] C. M. Wildrick, F. C. Lee, B. H. Cho, and B. Choi, "A method of defining the load impedance specification for a stable distributed power system," *IEEE Transactions on power Electronics,* vol. 10, no. 3, pp. 280-285, 1995.

[60] P. Liutanakul, A.-B. Awan, S. Pierfederici, B. Nahid-Mobarakeh, and F. Meibody-Tabar, "Linear stabilization of a DC bus supplying a constant power load: A general design approach," *IEEE Transactions on Power Electronics,* vol. 25, no. 2, pp. 475-488, 2009.

[61] F. De Bosio, L. A. de Souza Ribeiro, F. D. Freijedo, M. Pastorelli, and J. M. Guerrero, "Effect of state feedback coupling and system delays on the transient performance of stand-alone VSI with LC output filter," *IEEE Transactions on Industrial Electronics,* vol. 63, no. 8, pp. 4909-4918, 2016.

[62] I. S. Mohamed, S. Rovetta, T. D. Do, T. Dragicević, and A. A. Z. Diab, "A neural-network-based model predictive control of three-phase inverter with an output $ LC $ filter," *IEEE Access,* vol. 7, pp. 124737-124749, 2019.

[63] M. Novak and T. Dragicevic, "Supervised Imitation Learning of Finite-Set Model Predictive Control Systems for Power Electronics," *IEEE Transactions on Industrial Electronics,* vol. 68, no. 2, pp. 1717-1723, 2020.

[64] M. Hertneck, J. Köhler, S. Trimpe, and F. Allgöwer, "Learning an approximate model predictive controller with guarantees," *IEEE Control Systems Letters,* vol. 2, no. 3, pp. 543-548, 2018.

[65] I. Hammoud, S. Hentzelt, T. Oehlschlaegel, and R. Kennel, "Long-Horizon Direct Model Predictive Control Based on Neural Networks for Electrical

Drives," in *IECON 2020 The 46th Annual Conference of the IEEE Industrial Electronics Society*, 2020: IEEE, pp. 3057-3064.
[66] E. Zafra, S. Vazquez, A. M. Alcaide, L. G. Franquelo, J. I. Leon, and E. P. Martin, "K-best sphere decoding algorithm for long prediction horizon FCS-MPC," *IEEE Transactions on Industrial Electronics*, 2021.
[67] Q. Wang, G. Zhang, G. Wang, C. Li, and D. Xu, "Offline parameter self-learning method for general-purpose PMSM drives with estimation error compensation," *IEEE Transactions on Power Electronics*, vol. 34, no. 11, pp. 11103-11115, 2019.
[68] D. Wang *et al.*, "A deep neural network based predictive control strategy for high frequency multilevel converters," in *2018 IEEE Energy Conversion Congress and Exposition (ECCE)*, 2018: IEEE, pp. 2988-2992.
[69] T. Guillod, P. Papamanolis, and J. W. Kolar, "Artificial neural network (ANN) based fast and accurate inductor modeling and design," *IEEE Open Journal of Power Electronics*, vol. 1, pp. 284-299, 2020.
[70] S. Zhao, F. Blaabjerg, and H. Wang, "An overview of artificial intelligence applications for power electronics," *IEEE Transactions on Power Electronics*, 2020.
[71] M. Mirjafari and R. S. Balog, "Survey of modelling techniques used in optimisation of power electronic components," *IET Power Electronics*, vol. 7, no. 5, pp. 1192-1203, 2014.
[72] S. D. Sudhoff, *Power magnetic devices: a multi-objective design approach*. John Wiley & Sons, 2014.
[73] H. Liu, Y. Tian, Y. Gao, J. Bai, and J. Zheng, "System of systems oriented flight vehicle conceptual design: Perspectives and progresses," *Chinese Journal of Aeronautics*, vol. 28, no. 3, pp. 617-635, 2015.
[74] G. Bachelor, E. Brusa, D. Ferretto, and A. Mitschke, "Model-Based Design of Complex Aeronautical Systems Through Digital Twin and Thread Concepts," *IEEE Systems Journal*, 2019.
[75] X.-S. Yang, *A Brief History of Optimization*. John Wiley & Sons, Inc., 2010.
[76] C. A. C. Coello, G. B. Lamont, and D. A. Van Veldhuizen, *Evolutionary algorithms for solving multi-objective problems*. Springer, 2007.
[77] G. Grimmett and D. Stirzaker, *Probability and random processes*. Oxford university press, 2020.
[78] E. K. Burke and G. Kendall, *Search methodologies*. Springer, 2005.
[79] Q. Wei, Z. Liao, R. Song, P. Zhang, Z. Wang, and J. Xiao, "Self-Learning Optimal Control for Ice-Storage Air Conditioning Systems Via Data-Based Adaptive Dynamic Programming," *IEEE Transactions on Industrial Electronics*, 2020.
[80] X. Wang, J. Atkin, S. Bozhko, and C. Hill, "Application of a MILP-based Algorithm for Power Flow Optimisation within More-Electric Aircraft Electrical Power Systems," in *2019 21st European Conference on Power Electronics and Applications (EPE '19 ECCE Europe)*, 2019.
[81] X. Wang, Y. Gao, J. Atkin, and S. Bozhko, "Neural Network based Weighting Factor Selection of MPC for Optimal Battery and Load Management in MEA," in *2020 23rd International Conference on Electrical Machines and Systems (ICEMS)*, 2020: IEEE, pp. 1763-1768.
[82] B. Sareni, J. Regnier, and X. Roboam, "Recombination and self-adaptation in multi-objective genetic algorithms," in *International Conference on Artificial Evolution (Evolution Artificielle)*, 2003: Springer, pp. 115-126.
[83] K. Wang and B. Zhang, "Multiobjective trajectory optimization for a suborbital spaceplane using Directed Search Domain approach," *Aerospace Science and Technology*, vol. 77, pp. 713-724, 2018.
[84] A. G. Sarigiannidis, M. E. Beniakar, P. E. Kakosimos, A. G. Kladas, L. Papini, and C. Gerada, "Fault tolerant design of fractional slot winding permanent

magnet aerospace actuator," *IEEE Transactions on Transportation Electrification*, vol. 2, no. 3, pp. 380-390, 2016.
[85] A. Sarkheyli, A. Bagheri, B. Ghorbani-Vaghei, and R. Askari-Moghadam, "Using an effective tabu search in interactive resources scheduling problem for LEO satellites missions," *Aerospace Science and Technology*, vol. 29, no. 1, pp. 287-295, 2013.
[86] F. Glover and M. Laguna, "Tabu search," in *Handbook of combinatorial optimization*: Springer, 1998, pp. 2093-2229.
[87] C. M. Bishop, *Pattern recognition and machine learning*. springer, 2006.
[88] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.
[89] T. Dragičević and M. Novak, "Weighting factor design in model predictive control of power electronic converters: An artificial neural network approach," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 11, pp. 8870-8880, 2018.
[90] T. Dragičević, P. Wheeler, and F. Blaabjerg, "Artificial intelligence aided automated design for reliability of power electronic systems," *IEEE Transactions on Power Electronics*, vol. 34, no. 8, pp. 7161-7171, 2018.
[91] J. Shen and A. Khaligh, "A supervisory energy management control strategy in a battery/ultracapacitor hybrid energy storage system," *IEEE Transactions on Transportation Electrification*, vol. 1, no. 3, pp. 223-231, 2015.
[92] L. Zhang, Z. Zhou, Q. Chen, R. Long, and S. Quan, "Model Predictive Control for Electrochemical Impedance Spectroscopy Measurement of Fuel Cells Based on Neural Network Optimization," *IEEE Transactions on Transportation Electrification*, vol. 5, no. 2, pp. 524-534, 2019.
[93] Z. Xu, Y. Gao, X. Wang, X. Tao, and Q. Xu, "Surrogate Thermal Model for Power Electronic Modules using Artificial Neural Network," in *IECON 2019-45th Annual Conference of the IEEE Industrial Electronics Society*, 2019, vol. 1: IEEE, pp. 3160-3165.
[94] S. Wang, T. Dragicevic, Y. Gao, S. Chaudhary, and R. Teodorescu, "Machine Learning based Operating Region Extension of Modular Multilevel Converters under Unbalanced Grid Faults," *IEEE Transactions on Industrial Electronics*, vol. 68, no. 5, pp. 4554-4560, 2020.
[95] M. Jafari, L. E. Brown, and L. Gauchia, "Hierarchical Bayesian Model for Probabilistic Analysis of Electric Vehicle Battery Degradation," *IEEE Transactions on Transportation Electrification*, 2019.
[96] Z. Li *et al.*, "Online implementation of SVM based fault diagnosis strategy for PEMFC systems," *Applied energy*, vol. 164, pp. 284-293, 2016.
[97] C.-C. Chang and C.-J. Lin, "LIBSVM: a library for support vector machines," *ACM transactions on intelligent systems and technology (TIST)*, vol. 2, no. 3, pp. 1-27, 2011.
[98] T. P. Lillicrap *et al.*, "Continuous control with deep reinforcement learning," *arXiv preprint arXiv:1509.02971*, 2015.
[99] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE transactions on evolutionary computation*, vol. 6, no. 2, pp. 182-197, 2002.
[100] T. Murata and H. Ishibuchi, "MOGA: Multi-objective genetic algorithms," in *IEEE international conference on evolutionary computation*, 1995, vol. 1, pp. 289-294.
[101] F. De Toro, J. Ortega, J. Fernández, and A. Díaz, "PSFGA: a parallel genetic algorithm for multiobjective optimization," in *Proceedings 10th Euromicro Workshop on Parallel, Distributed and Network-based Processing*, 2002: IEEE, pp. 384-391.

[102] R. Norouzi, A. Kosari, and M. H. Sabour, "Real time estimation of impaired aircraft flight envelope using feedforward neural networks," *Aerospace Science and Technology*, vol. 90, pp. 434-451, 2019.
[103] R. Thirumalainambi and J. Bardina, "Training data requirement for a neural network to predict aerodynamic coefficients," in *Independent Component Analyses, Wavelets, and Neural Networks*, 2003, vol. 5102: International Society for Optics and Photonics, pp. 92-103.
[104] S. Wang, T. Dragicevic, Y. Gao, and R. Teodorescu, "Neural Network based Model Predictive Controllers for Modular Multilevel Converters," *IEEE Transactions on Energy Conversion*, 2020.
[105] X. Fu and S. Li, "Control of single-phase grid-connected converters with LCL filters using recurrent neural network and conventional control methods," *IEEE Transactions on Power Electronics*, vol. 31, no. 7, pp. 5354-5364, 2015.
[106] A. Boutemedjet, M. Samardžić, L. Rebhi, Z. Rajić, and T. Mouada, "UAV aerodynamic design involving genetic algorithm and artificial neural network for wing preliminary computation," *Aerospace Science and Technology*, vol. 84, pp. 464-483, 2019.
[107] F. Wang, G. Chen, D. Boroyevich, S. Ragon, M. Arpilliere, and V. R. Stefanovic, "Analysis and design optimization of diode front-end rectifier passive components for voltage source inverters," *IEEE transactions on power electronics*, vol. 23, no. 5, pp. 2278-2289, 2008.
[108] TDK. (2018). *Film Capacitor MKP DC B2562*, ser. MKP DC B2562** [Online]. Available: https://en.tdk-electronics.tdk.com/inf/20/50/ds/B2562.pdf.
[109] Magnetics. (2017). *2017 Magnetics Powder Core Catalog, ser. Toroid Data* [Online]. Available: https://www.mag-inc.com/Company/News/2017%E5%B9%B411%E6%9C%88/2017-Magnetics-Powder-Core-Catalog-in-Chinese.
[110] J. Rodriguez *et al.*, "Predictive current control of a voltage source inverter," *IEEE transactions on industrial electronics*, vol. 54, no. 1, pp. 495-503, 2007.
[111] R. Vargas, P. Cortés, U. Ammann, J. Rodríguez, and J. Pontt, "Predictive control of a three-phase neutral-point-clamped inverter," *IEEE Transactions on Industrial Electronics*, vol. 54, no. 5, pp. 2697-2705, 2007.
[112] H. Y. Alhammadi and J. A. Romagnoli, "Incorporating environmental, profitability, heat integration and controllability considerations," *The integration of process design and control*, vol. 17, p. 264, 2004.
[113] K. Deb and H. Jain, "An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: solving problems with box constraints," *IEEE transactions on evolutionary computation*, vol. 18, no. 4, pp. 577-601, 2013.
[114] MathWorks. (2019). *Support Documentation, gamultiobj (R2019b)* [Online]. Available: https://uk.mathworks.com/help/gads/gamultiobj.html.
[115] Z. Jiao *et al.*, "An intelligent design method for actuation system architecture optimization for more electrical aircraft," *Aerospace Science and Technology*, vol. 93, p. 105079, 2019.
[116] MathWorks. (2019). *Support Documentation, fitcsvm (R2019b)* [Online]. Available: https://uk.mathworks.com/help/stats/fitcsvm.html.
[117] S. Ganesh. (2020). *What's The Role Of Weights And Bias In a Neural Network?* [Online]. Available: https://towardsdatascience.com/whats-the-role-of-weights-and-bias-in-a-neural-network-4cf7e9888a0f.
[118] WIKIPEDIA. (2021). *Training, validation, and test sets* [Online]. Available: https://en.wikipedia.org/wiki/Training,_validation,_and_test_sets.
[119] G.-H. Liu, Y.-B. Chen, Y. Gao, J.-N. Zhu, B.-X. Wang, and T. Yang, "Component Based and Machine Learning Aided Optimal Filter Design for Full-Bridge Current Doubler Rectifier," in *IECON 2021–47th Annual Conference of the IEEE Industrial Electronics Society*, 2021: IEEE, pp. 1-6.

[120] S. Chen *et al.*, "Approximating explicit model predictive control using constrained neural networks," in *2018 Annual American control conference (ACC)*, 2018: IEEE, pp. 1520-1527.
[121] K. Ilves, S. Norrga, L. Harnefors, and H.-P. Nee, "On energy storage requirements in modular multilevel converters," *IEEE transactions on power electronics*, vol. 29, no. 1, pp. 77-88, 2013.
[122] K. Sharifabadi, L. Harnefors, H.-P. Nee, S. Norrga, and R. Teodorescu, *Design, control, and application of modular multilevel converters for HVDC transmission systems*. John Wiley & Sons, 2016.
[123] Z. Gong, P. Dai, X. Yuan, X. Wu, and G. Guo, "Design and experimental evaluation of fast model predictive control for modular multilevel converters," *IEEE Transactions on Industrial Electronics*, vol. 63, no. 6, pp. 3845-3856, 2015.
[124] S. Wang, T. Dragicevic, G. F. Gontijo, S. K. Chaudhary, and R. Teodorescu, "Machine Learning Emulation of Model Predictive Control for Modular Multilevel Converters," *IEEE Transactions on Industrial Electronics*, 2020.
[125] P. Karamanakos, T. Geyer, and R. Kennel, "On the choice of norm in finite control set model predictive control," *IEEE Transactions on Power Electronics*, vol. 33, no. 8, pp. 7105-7117, 2017.
[126] P. Cortes, J. Rodriguez, C. Silva, and A. Flores, "Delay compensation in model predictive current control of a three-phase inverter," *IEEE Transactions on Industrial Electronics*, vol. 59, no. 2, pp. 1323-1325, 2011.
[127] MathWorks. (2020). *Support Documentation, trainbr (R2020a)* [Online]. Available: https://uk.mathworks.com/help/deeplearning/ref/trainbr.html.
[128] S. Vazquez *et al.*, "An Artificial Intelligence Approach for Real-Time Tuning of Weighting Factors in FCS-MPC for Power Converters," *IEEE Transactions on Industrial Electronics*, 2021.
[129] G. Bramerdorfer, A.-C. Zăvoianu, S. Silber, E. Lughofer, and W. Amrhein, "Possibilities for speeding up the FE-based optimization of electrical machines—A case study," *IEEE Transactions on Industry Applications*, vol. 52, no. 6, pp. 4668-4677, 2016.
[130] M. R. Khowja, C. Gerada, G. Vakil, C. Patel, and P. Wheeler, "Design optimization of integrated rotational inductor for high-speed AC drive applications," in *2017 IEEE International Electric Machines and Drives Conference (IEMDC)*, 2017: IEEE, pp. 1-8.
[131] A. Boglietti, A. Cavagnino, and M. Lazzari, "Computational algorithms for induction-motor equivalent circuit parameter determination—Part I: Resistances and leakage reactances," *IEEE Transactions on Industrial Electronics*, vol. 58, no. 9, pp. 3723-3733, 2010.
[132] E. Schmidt, M. Kaltenbacher, and A. Wolfschluckner, "Eddy current losses in permanent magnets of surface mounted permanent magnet synchronous machines—Analytical calculation and high order finite element analyses," *e & i Elektrotechnik und Informationstechnik*, vol. 134, no. 2, pp. 148-155, 2017.
[133] R. Sikora, J. Purczynski, W. Lipinski, and M. Gramz, "Use of variational methods to the eddy currents calculation in thin conducting plates," *IEEE Transactions on Magnetics*, vol. 14, no. 5, pp. 383-385, 1978.
[134] M. Cheng and S. Zhu, "Calculation of PM eddy current loss in IPM machine under PWM VSI supply with combined 2-D FE and analytical method," *IEEE Transactions on Magnetics*, vol. 53, no. 1, pp. 1-12, 2016.
[135] A. Hemeida, P. Sergeant, and H. Vansompel, "Comparison of methods for permanent magnet eddy-current loss computations with and without reaction field considerations in axial flux PMSM," *IEEE Transactions on Magnetics*, vol. 51, no. 9, pp. 1-11, 2015.
[136] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural networks*, vol. 2, no. 5, pp. 359-366, 1989.

[137] MathWorks. (2021). *train, Train shallow neural network* [Online]. Available: https://uk.mathworks.com/help/deeplearning/ref/network.train.html?searchHighlight=train&s_tid=srchtitle.
[138] MathWorks. (2021). *min, Minimum elements of an array* [Online]. Available: https://uk.mathworks.com/help/matlab/ref/min.html?searchHighlight=min&s_tid=srchtitle.
[139] A.-C. Zăvoianu, G. Bramerdorfer, E. Lughofer, S. Silber, W. Amrhein, and E. P. Klement, "Hybridization of multi-objective evolutionary algorithms and artificial neural networks for optimizing the performance of electrical drives," *Engineering Applications of Artificial Intelligence*, vol. 26, no. 8, pp. 1781-1794, 2013.
[140] P.-Y. Lin and Y.-S. Lai, "Voltage control technique for the extension of DC-link voltage utilization of finite-speed SPMSM drives," *IEEE Transactions on Industrial Electronics*, vol. 59, no. 9, pp. 3392-3402, 2011.
[141] D. G. Holmes and T. A. Lipo, *Pulse width modulation for power converters: principles and practice*. John Wiley & Sons, 2003.
[142] A. Wintrich, U. Nicolai, W. Tursky, and T. Reimann. (2015). *Application Manual Power Semiconductors* [Online]. Available: https://www.semikron.com/service-support/application-manual.html.
[143] J. Miret, P. P. Balestrassi, A. Camacho, R. Guzmán, and M. Castilla, "Optimal tuning of the control parameters of an inverter-based microgrid using the methodology of design of experiments," *IET Power Electronics*, vol. 13, no. 16, pp. 3651-3660, 2020.