Torque Ripple Reduction in a.c. Permanent Magnet Servo Motor Drives

MI TANG

Submitted to the University of Nottingham for the degree of Doctor of Philosophy,

June 2017.

Abstract

Servo systems play an important role in industrial automation. A servo system denotes a closed loop controlled system capable of tracking required demands. One way of achieving high performance servo drive systems is to apply the closed loop control of an a.c. permanent magnet synchronous machine (PMSM). PMSM is a type of machine which rotates once three-phase a.c. voltages are supplied. The usage of permanent magnet materials contributes to the high efficiency of PMSM, and makes it a popular type of machine in industrial applications. However, the interaction between the permanent magnets and the machine stator would generate torque ripple and consequently unsmooth speed. Therefore, torque ripple of PMSM need to be considered carefully in the control of such servo systems.

An innovative control scheme combining an enhanced high bandwidth deadbeat current controller and a fractional delay variable frequency angle-based repetitive controller, is developed in this work in order to minimize torque ripple. For the purpose of accurately modelling the cogging torque and flux harmonics in PMSM, a lookup table embedded PMSM model is also proposed. It has been validated by both simulative and experimental tests that the proposed control scheme is able to reduce torque ripple in a PMSM drive system effectively for a wide range of frequencies, and even during transients, which has never been achieved according to the author's knowledge. The proposed method is not only adaptive to variable frequencies, but also adaptive to the variations of electrical and mechanical parameters in normal operating conditions.

Acknowledgements

People behave like a repetitive controller and how we response to any "inputs" in life depend on our memories or experiences. Memories are therefore as important to a repetitive controller as to us. Each part written in this thesis is about a piece of memory of mine on this four year PhD journey. The modelling part is the memory of running finite element analysis day and night for months. The dead beat current control part is the memory of deriving equations again and again, for five times. The repetitive control part is the memory of improving the control structure little by little until different loops can work together... Fortunately, I am not alone in all these memories. Thanks to my dear supervisors, sponsor, colleagues, friends, and family members the bad time and good time have all become precious time to me.

I would like to give my biggest thank you to my supervisors, Prof. Pericle Zanchetta, Prof. Greg Asher, Dr. Alberto Gaeta, Dr. Andrea Formentini for their unreserved support, and my examiners, Prof. Chris Gerada, Dr. Tom Cox, Dr. Mohammad A. Abusara for their suggestions. Thank Prof. Ohyama Kazuhiro for his help in my first journal publication. Thank Dr. Shafiq Odhando for his volunteering help. Thank Dr. Jordi Espina for his important support for EMI issue. Thank technicians John, Ian, Guy, Kevin for the rig construction. Thank Mr. Hui Xu, Mr. Alan Yeung, Mr. Chen Liu, Mr. Jie Liu, Dr. Jiongkang Lin from my sponsor company, Shenzhen Best Motion .

I shall not forget the endless supports from my dear colleagues and friends: Dr. Marco Degano, Dr. David Hind, Dr. Fabio Tinazzi, Dr. Chao Ji, Dr. Tao Yang, Dr. Shuai Shao, Dr. Jing Li, Dr. He Zhang, Dr. Xiaochen Zhang, Dr. Bilal Arif, Dr. PG. Boong, Dr. Christopher Hill, Dr. Manjusha Vijayagopal, Dr. Marija Jankovic, Dr. Savvas Papadopoulos, Dr. Fei Gao, Dr. Yixue Feng, Mr. Giorgio Valente, Mr. Wenping Song, Miss Xiaoyi Bai, Dr. Dewen Meng, Miss Jingru Dai, Miss Laura Sun, Dr. Min Chen, Mr. Yunpeng Tang, Mr. Xiaofeng Ding, Miss Xiang Liu, Mr.Zhenyu Wang, Mr. Jiawei Zhang, Mr. Jiaxing Lei and all the other PEMC group members.

The last but not least, my dear parents Mr. Xiaochun Tang, Mrs. Hedong An, my grandmas Mrs. Jingxiu Huang, Mrs. Yunlan Zhang and all my family members. I can never be me without them.

List of Terms

v_{abc} - Three-phase stator voltages

$v_{\alpha\beta}$ - Stator voltages in $\alpha\beta$ coordinates

v_{dq} - Stator voltages in dq coordinates

i_{abc} - Three-phase stator currents

$i_{\alpha\beta}$ - Stator currents in $\alpha\beta$ coordinates

i_{dq} - Stator currents in dq coordinates

F_{abc} - Three-phase stator flux linkages

F_s - Peak value of the three pahse stator flux linkages

$F_{\alpha\beta}$ - Stator flux linkages in $\alpha\beta$ coordinates

F_{dq} - Stator flux linkages in dq coordinates

$F_d^{ac}(\theta_e, i_d, i_q), F_q^{ac}(\theta_e, i_d, i_q)$ - Lookup tables for a.c. part in dq axis flux linkages

F_m - Magnetic flux (d axis)

F_m^{est} - Estimated magnetic flux for control

$F_{md}(i_d, i_q), F_{mq}(i_d, i_q)$ - Lookup tables for the dq axis components of magnetic flux ($F_{mq}(i_d, i_q)$ is neglected)

P_e - Electromagnetic power

R_s - Stator resistance

R_s^{est} - Estimated stator resistance for control

L_a, L_b, L_c - Three-phase stator self inductances, rotor position and saturation dependent

$L_{ab}, L_{bc}, L_{ac},$ - Mutual inductances between each phase, rotor position and saturation dependent

L_0 - Offset of the three-phase self inductance

L_1 - Amplitude of the three-phase self inductance

L_α, L_β - Equivalent self inductances in $\alpha\beta$ coordinates, rotor position and saturation dependent

$L_{\alpha\beta}$ - Equivalent mutual inductance in $\alpha\beta$ coordinates, rotor position and saturation dependent

L_d, L_q - Inductances in dq coordinates

$L_d(i_d, i_q), L_q(i_d, i_q)$ - Lookup tables for dq axis inductances

L_d^{est}, L_q^{est} - Estimated dq axis inductances for control

T_e - Output electrical torque of machine

T_e^{ref} - Electrical torque reference

T_e^{est} - Estimated electrical torque, output of torque estimator

T_e^{refdc} - d.c. value of the torque reference

$T_e^{err}(\theta_e, i_d, i_q)$ - Lookup table for torque ripple produced by machine

θ_m - Rotor mechanical position

θ_m^{ref} - Rotor mechanical position reference

θ_e - Rotor electrical position

ω_m - Rotor mechanical angular speed

ω_m^{ref} - Rotor mechanical angular speed reference

ω_e - Rotor electrical speed

N_r^{ref} - Rotor mechanical speed reference (rpm)

N_r - Rotor mechanical speed (rpm)

N_r^{rat} - Rated rotor mechanical speed (rpm)

P_p - Number of pole pairs

T_l - Load torque

J - Moment of inertia

J^{est} - Estimated moment of inertia for control

B - Friction factor

B^{est} - Estimated friction factor for control

K_e - Voltage constant of the machine

K_t - Torque constant of the machine

T_e^{err} - Error between the reference torque and estimated torque, input of repetitive controller

i_{qRC}^{ref} - Output of repetitive controller

f_s - Sampling frequency

T_s - Sampling period

T_d - Delay of the current loop

N - The number of samples in the delay chain of time-based repetitive controller, length of mem in the angle-based repetitive controller

k_1 - Rotor position index, within $[0, N-1]$

k_2 - Rotor future position index, within $[0, N-1]$

$mem(k_1)$ - The $(k_1+1)^{th}$ value in mem

$mem^{prev}(k_1)$ - The previous $(k_1+1)^{th}$ value in mem obtained one full cycle rotation ago

M - The number of leading samples for the phase lead compensator z^M

G_{RC} - Gain of repetitive controller

Q_{RC} - Forgetting factor of repetitive controller, within [0,1]

δT_e^{max} - Threshold for transient detector

δt - The inspection duration for transient detector

t^{ste} - Minimum steady state time for being recognized as in steady state

S_{speed} - Transient signal for speed transient (0-transient; 1-steady state)

S_{load} - Transient signal for load transient (0-transient; 1-steady state)

S^{ena} - Enable signal, output of the transient detector (0-transient; 1-steady state)

S_{prev}^{ena} - Enable signal at the previous sampling period (0-transient; 1-steady state)

$G_{open}(z)$ - Open loop transfer function of the proposed control system

$G_{closed}(z)$ - Closed loop transfer function of the proposed control system

$G_{cl}(z)$ - Transfer function of the current loop

$G_{te}(z)$ - Transfer function of the torque estimator

$G_{sp}(z)$ - Transfer function of the speed loop plant

$G_{rc}(z)$ - Transfer function of repetitive controller

$G_{sc}(z)$ - Transfer function of the speed loop controller

Contents

1 Introduction **2**

1.1 Servo Motor Drives 3

1.2 a.c. vs. d.c. Motors 6

1.3 Permanent Magnet Motors vs. Induction Motors 7

1.4 The Need for Torque Ripple Reduction 11

1.5 Methodology and Novelties of the Work 11

1.6 Outline of the Thesis 12

1.7 Summary 13

2 Literature Review **15**

2.1 Sources of Torque Ripple 15

2.1.1 Cogging Effect of PMSM 16

2.1.2 Flux Harmonics in PMSM 17

2.1.3 Current Offset Error in Sampling 19

2.1.4 Current Scaling Error in Sampling 20

2.1.5 Current Harmonics in Converter 20

2.1.6 Mechanical Imperfections 22

2.2 Existing PMSM Models for Simulation 22

2.2.1 Finite Element Model 23

2.2.2 Equivalent Circuit Model: Lumped Parameter Model and Permeance Network Model 23

2.2.3 Simplified Model in *abc* Coordinates 24

2.2.4 Simplified Model in $\alpha\beta$ Coordinates 27

2.2.5 Simplified Model in *dq* Coordinates 27

2.3 Existing Closed Loop Control Methodologies for PMSM Drives 29

2.4 Torque Ripple Reduction by Machine Design 35

2.5 Torque Ripple Reduction by Control 38

2.6 Discussions and Conclusions 43

3 Modelling of PMSM **46**

3.1 Finite Element Model 47

3.1.1 Lookup Tables $F_{dq}^{ac}(\theta_e, i_d, i_q)$ for the a.c. Part of *dq* Axis Flux Linkages 49

3.1.2 Lookup Tables $F_{mdq}(i_d, i_q)$ for the *dq* Axis Magnetic Flux 51

3.1.3 Lookup Tables $L_{dq}(i_d, i_q)$ for the *dq* Axis Inductances 52

3.1.4 Lookup Table $T_e^{err}(\theta_e, i_d, i_q)$ for Error Torque 53

3.2 The Combined Model 54

3.3 Conclusions 58

4 Current Control Loop of PMSM 61

4.1 Deadbeat Current Control Loop 62

4.1.1 Current Predictions 65

4.1.1.1 Two Steps Current Prediction 66

4.1.2 Reference Voltage Calculation 70

4.1.3 Rotor Movement Compensation 70

4.1.3.1 Existing Compensation Methods 71

4.1.3.2 Novel compensation Method 72

4.1.4 Inverter Non-linearity Compensation 74

4.1.5 Steady State Error Compensation 75

4.2 Simulation and Experimental Tests 76

4.2.1 Performance Assessment 80

4.2.1.1 Dynamic Assessment using Bode Diagram 80

4.2.1.2 Steady State Assessment 84

4.2.2 Effectiveness of the proposed DBCC 86

4.2.2.1 Two Steps Prediction and Operating at Physical Limit . . 87

4.2.2.2 Rotor Movement Compensation 88

4.2.2.3 Steady State Error compensation 90

4.2.3 Comparison Between DBCC and PI 90

4.3 Conclusions 92

5 Torque Ripple Reduction Techniques for PMSMs 95

5.1 Existing Techniques 96

5.1.1 Holtz' Compensation 96

5.1.2 Choi's Compensation 99

5.1.3 Repetitive Control 101

5.2 Combination of Repetitive Control and Deadbeat Current Control 104

5.2.1 Possible Topologies 104

5.2.2 Time-based Repetitive Control 107

5.2.3 Angle-based Repetitive Control 108

5.2.3.1 Torque Estimator 109

5.2.3.2 Time to Angle Conversion 110

5.2.3.3 Main Body of Repetitive Control 114

5.2.3.4 Angle to Time Conversion 115

5.2.4 Stability Enhancement 117

5.2.4.1 Influence of Forgetting Factor Q_{RC} on Stability 119

5.2.4.2 Influence of Phase Lead Compensator z^M on Stability . . 120

5.2.4.3 Transient Detector 122

5.3 Conclusions 124

6 The Proposed Torque Ripple Reduction Method in Simulation 127

6.1 Simulation Setup 128

6.2 Performance of the Proposed Control in Ideal Conditions 131

6.2.1 Influence of Forgetting Factor Q_{RC} on Performance 131

6.2.2 Influence of Gain G_{RC} on Performance 132

6.2.3 Performance of Proposed RC with Tuned Q_{RC} and G_{RC} 134

6.3 Performance and Stability of the Proposed Control in Non-ideal Conditions 137

6.3.1 Influence of Uncertainties in Deadbeat Current Control Loop 137

6.3.2 Influence of Uncertainties in Torque Estimator 140

6.3.3 Effectiveness of Interpolations for Solving Fractional Delay Issue . . 144

6.3.4 Effectiveness of Transient Detector 147

6.4 Conclusions 150

7 The Proposed Torque Ripple Reduction Method in Experiment 152

7.1 Experimental Rig Setup 153

7.2 Estimation of Torque Ripple from Position Feedback with FIR Filters . . . 156

7.3 Effectiveness of Torque Ripple Reduction 161

7.3.1 Effectiveness under Integer Delay Condition 162

7.3.2 Effectiveness under Fractional Delay Condition 164

7.3.3 Effectiveness during Speed Transient 167

7.3.4 Effectiveness during Load Transient 169

7.4 Conclusions 171

8 Conclusions 173

8.1 Conclusions 175

8.2 Limitations and Further Works 178

A Fourier Series **190**

B Clarke Transformation and Park Transformation **192**

C DSP Code for the Proposed Repetitive Controller **195**

D Published Papers **208**

List of Figures

1.1 Servo market share of applications 2

1.2 A typical closed loop servo motor control system 4

1.3 Rotor structures for three common types of PMSMs (permanent magnets are represented blue) 10

1.4 The area of this work in the big picture of electrical drives 14

2.1 Cross-section of the surface mounted PMSM used for modelling purposes in this work 17

2.2 Available drive converter topologies 21

2.3 Schematic diagram of a two-level voltage source inverter 21

2.4 Block diagram of field oriented control 29

2.5 Block diagram of direct torque flux control 30

2.6 Block diagram of neural network control 31

2.7 Block diagram of a pre-computed lookup table method for reduction of torque ripple caused by non-sinusoidal flux distribution 39

2.8 Block diagram of the Holtz method for torque ripple reduction 40

2.9 Block diagram of an uncertainty function method for torque ripple reduction 40

2.10 Block diagram of a memory position-based method for torque ripple reduction 41

2.11 Block diagram of an ILC-based method for torque ripple reduction using the measured torque as feedback 42

2.12 Block diagram of an ILC-based method for torque ripple reduction using position as feedback 43

2.13 Block diagram of Choi's method for torque ripple reduction 43

2.14 Key conclusions of Chapter 2 44

3.1 Setting of dq axis currents for finite element analysis 48

3.2 Example of the $F_{dq}^{ac}(\theta_e, i_d, i_q)$ lookup tables 50

3.3 Example of the $F_{md}(i_d, i_q)$ lookup table 52

3.4 Example of the $L_{dq}(i_d, i_q)$ lookup tables 53

3.5 Example of the $T_e^{err}(\theta_e, i_d, i_q)$ lookup table 54

3.6 The proposed lookup table embedded $\alpha\beta$ frame PMSM model with PI current control loops 56

3.7 The agreement between the proposed PMSM model and FE model 57

4.1 Time sequence of the proposed DBCC for the k^{th} sampling period 63

4.2 Block diagram of the proposed deadbeat current controller 65

4.3 dq reference frame for the existing rotor movement compensation 72

4.4 dq reference frame for the proposed rotor movement compensation 72

4.5 The DBCC with plug-in integrator for removing steady state error 76

4.6 Simulation model for the assessment of deadbeat current control 77

4.7 Experimental rig for the assessment of deadbeat current control 79

4.8 Simulation results of bode diagrams of q axis current control loop with parameter detuned (i_d^{ref}=0A, bias of i_q^{ref}=8.34A, T_s=100μs, N_r=3000rpm) 83

4.9 Simulation results of bode diagrams of q axis current control loop with correct parameters (i_d^{ref}=0A, bias of i_q^{ref}=8.34A, T_s=100μs,N_r=10$\leftrightarrow$6000rpm) 84

4.10 Simulation results of steady state current errors of DBCC due to parameter detuning (i_d^{ref}=0A, i_q^{ref}=8.34A, N_r=10$\leftrightarrow$6000rpm) 85

4.11 Simulation results of steady state current errors of DBCC due to dead time (i_d^{ref}=0A, N_r=3000rpm) 86

4.12 Experimental results of rectangular current responses of DBCC 88

4.13 Effectiveness of the proposed rotor movement compensation 89

4.14 Comparison of the proposed rotor movement compensation and existing compensation (Simulation results, i_d^{ref}=-2A, i_q^{ref}=3A, N_r=10000rpm) . . . 89

4.15 Effectiveness of the steady state error compensation with plug-in integrator (Simulation results: All the parameters in controller are 80% of the values in motor,i_d^{ref}=-2A, i_q^{ref} steps from 3A to 7A at 0.1s) 90

4.16 Experimental results of current responses for 10000 rad/s ($N_r^{ref} = 2930 + 30sin(10000t)$rpm , i_d^{ref}=0A) 91

4.17 Experimental results of current responses for 5000 rad/s (i_d^{ref}=0A) 91

5.1 Block diagram for Holtz compensation method 97

5.2 Block diagram for Choi's compensation scheme 100

5.3 Block diagram for Mattavelli's repetitive control 102

5.4 Three available structures for repetitive controller with deadbeat current control loop 106

5.5 The proposed plug-in RC with torque estimator 106

5.6 Block diagram of the repetitive controller in time-domain 107

5.7 Block diagram of the proposed angle-based repetitive control system during the k^{th} sampling period 109

5.8 The relationship between the rotor location and memory location 112

5.9 Principle of the interpolations in the proposed time to angle conversion . . 113

5.10 Block diagram of the main body of RC in angle-domain 114

5.11 Principle of the interpolations in the proposed angle to time conversion . . 116

5.12 Block diagram of the proposed control system 117

5.13 Equivalent diagram of the proposed control system 118

5.14 Pole-zero plot of closed loop system with forgetting factor Q_{RC} varying from 0.01 to 0.99 120

5.15 Pole-zero plot of closed loop system with the delay of current loop T_d increased from $2T_s$ to $3T_s$ 121

5.16 The development of the proposed control scheme in Chapter 5 126

6.1 Benefits and uncertainties of the four innovative strategies 128

6.2 The resultant output torque of PMSM with different forgetting factor Q_{RC} 132

6.3 The resultant output torque of PMSM with different gain G_{RC} 133

6.4 The resultant output torque of PMSM with tuned Q_{RC} and G_{RC} 136

6.5 The resultant output torque of PMSM with parameter mismatches in dead-beat current control 138

6.6 The resultant output torque of PMSM with parameter mismatches in torque estimator 142

6.7 Bode diagram of $G_{te}G_{sp}$ with parameter mismatches 143

6.8 Phase error in samples due to fractional delay issue 145

6.9 The resultant output torque of PMSM with/without proposed interpolations 146

6.10 The resultant output torque of PMSM with/without the proposed transient detector 148

6.11 The resultant output torque of PMSM with/without the proposed transient detector (zoom in) 149

7.1 Experimental validation of the four innovative aspects 153

7.2 Experimental rig 154

7.3 The calculated speed from encoder feedback (ω_m^{ref}=500 rpm, without repetitive controller) 160

7.4 The estimated torque T_e^{est} (ω_m^{ref}=500 rpm, without repetitive controller) . 161

7.5 Effectiveness of the proposed control at 500 rpm (integer delay condition) . 163

7.6 FFT of rotor speed under different speed (integer delay condition) 164

7.7 Effectiveness of the proposed control at 451 rpm (fractional delay condition) 165

7.8 FFT of rotor speed under different speed (fractional delay condition) . . . 166

7.9 Effectiveness of the proposed control during a speed transient at 2s 168

7.10 Effectiveness of the proposed control during a load transient at 1s 170

7.11 Key results and findings in Chapter 7 171

8.1 Map of this work 174

A.1 An example square wave 190

B.1 Three reference frames for a PMSM 193

List of Tables

1.1 Comparison between electrical, hydraulic and pneumatic motors 5

1.2 Comparison between PMSM and IM 8

1.3 The comparison between two types of PMSM 10

2.1 Potential advantages and disadvantages of different control methodologies for PMSM drives 34

2.2 Advantages and disadvantages of torque ripple reduction methodologies by machine design-I 36

2.3 Advantages and disadvantages of torque ripple reduction methodologies by machine design-II 37

3.1 Parameters of the PMSM for modelling according to manufacturer's datasheet 47

4.1 Parameters (tuned) of the PMSM for DB current loop tests 78

4.2 Table of the functionalities implemented in simulation/experimental rig and the corresponding results 93

6.1 Parameters for simulation tests 130

7.1 Parameters (tuned) for experimental validation 155

Chapter 1

Introduction

11.13 billion US dollars, is the expected global servo system market size in 2016 according to [1].

Servo (i.e. closed loop controlled device) plays an irreplaceable role in industrial automation and is widely used in machine tools, textiles, packaging, printing, light industry, robotics, and more as illustrated in Fig.1.1 according to [2].

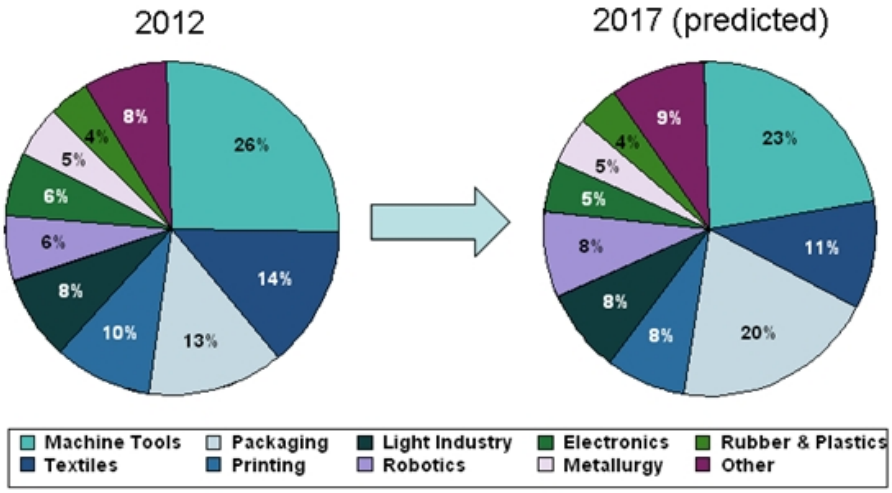


Figure 1.1: Servo market share of applications

The servo system market includes sensors, controller modules, encoders, amplifiers and the largest share is the servo motors. Furthermore, the China servo system market itself was valued at 1.09 billion US dollars in 2015, which shares 11.1% of the world's

total. With policy support from Chinese government, the application of servo systems in industrial fields such as machine tool, wind generation, robotics is steadily growing at a pace of about 10% during 2016-2020 [1].

Despite the fact that the largest servo system market is in China, Chinese local brands plays little role in the market. According to [3], the main players includes Japanese brands: Yaskawa Electric Corporation, Mitsubishi Electric Corporation; German brand: Siemens AG; French brand: Schneider Electric; American brands: Rockwell Automation Inc., Asea Brown Boveri (ABB) ltd., Emerson Electic Co.. This motivated initially the establishment of the Shenzhen Best Motion Technology innovation centre at the University of Nottingham. This work focuses on developing new technologies for servo motor drives applications, and has been carried out within this innovation centre.

This work is under the title of "**Torque Ripple Reduction in a.c. Permanent Magnet Servo Motor Drives**". The principle of **servo motor drives** (in section 1.1), the advantages of using **a.c.** drives instead of d.c. drives (in section 1.2), the benefits of using **permanent magnet** motor instead of induction motors (in section 1.3), and the necessity of **reducing torque ripple** (in section 1.4) will be discussed. The scope of this work as well as its **novelties** will be discussed in 1.5. Finally, the **aims and objectives** of this work will be given in 1.6.

1.1 Servo Motor Drives

In servo motor drive systems, real-time tracking of the demand is required. The demand signal may vary depending on applications. It can be temperature, stress, current and so on. As shown in [4], servo systems can be categorised into three main groups according to the operating principle: hydraulic, pneumatic and electric servos. By definition, the hydraulic and pneumatic servos generate motion by using pressurized oil or air, respectively. Table 1.1 in [4] also shows benefits, drawbacks and some examples of each type of servo motor. Some major points can be summarized in Table 1.1 below.

As it can be seen, the electrical servo may be preferred due to its excellent dynamics and efficiency. It may also become **more and more economical** as the price of the sensors may reduce in the future.

A typical commercial electrical servo driver in the market normally can be characterized by the following three operation modes: 1) position control mode; 2) speed control mode; 3) torque (i.e. current) control mode. The realization of such control is achieved by applying closed loop control techniques. With the development of modern closed loop control techniques, **the electrical servo becomes capable of producing more and more outstanding performance**, since faster performance and higher efficiency become possible.

Fig.1.2 shows the structure of **a typical closed loop servo motor control system**. Where, sensor/transducer/encoder are used for measuring real-time currents, voltages, or rotor position as required. The measured signals are then compared with the demand signals (position, speed or current depending on the control mode). Consequently, the controller takes actions to amend the reference speed, torque or terminal voltage of the inverter, in order to adjust the real position, speed or current of the motor respectively. With the controller tuned properly, the differences between the demand signals and the measured feedback signals could be reduced to values acceptable for the particular application.

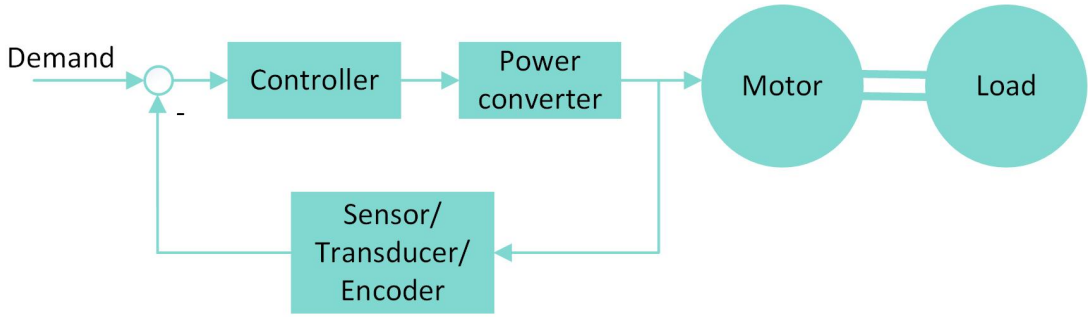


Figure 1.2: A typical closed loop servo motor control system

Table 1.1: Comparison between electrical, hydraulic and pneumatic motors

	Benefits	Drawbacks
Electrical motor	Excellent dynamics; Mature technology; Simple wiring; Wide product variety; Several suppliers available; High speeds possible; Excellent efficiency;	Expensive; Lower torque or force than in hydraulic systems; Requires lots of sensors;
Hydraulic motor	Easy to apply; High torque available; Centralized power source; Easy to use; High force density;	Arrangement of pressurization network; Audible noise; Accurate speed control difficult; Slow positioning; Oil leakages; Poor efficiency; Fire hazard; Regular maintenance required;
Pneumatic motor	Low cost; Easy to use; Requires little maintenance; Centralized power source;	Arrangement of pressurization network; Audible compressor noise; Accurate speed control difficult; Poor efficiency; Low force density; Non-linear due to air compressibility

1.2 a.c. vs. d.c. Motors

According to [5], d.c. motor was preferred until the 1950s for applications requiring variable speed drive for the simplicity of its speed and current control, whereas the a.c. motor drive technology at that time was not capable of smoothly varying speed. However, the development of control techniques and microelectronics made it possible for a.c. drives to be used in high performance applications. The advantages of a.c. motor drives, compared with d.c. motor drives, are discussed in [5] in terms of ten aspects including efficiency, inertia, maintenance, ruggedness, performance, size and weight. It is worth pointing out that many of these advantages benefit from the fact that there are no commutators or brushes in a.c. motors.

The presence of commutator or brush leads to a number of restrictions for the brushed only d.c. motor drives (the permanent magnet brushless d.c. (PMDC) motors are different from the brushed d.c. motor discussed here, the explanation of their differences will be given in section 1.7). As a result:

1) d.c. motor drives have limited current rising rate, and thus limited torque response.

2) a.c. motor can achieve better efficiency when used under variable speed.

3) The inertia of an a.c. motor is normally less than a d.c. motor of the same power.

4) There would be less maintenance cost for the a.c. motor

5) The size and weight of the an a.c. motor is significantly smaller than a d.c. motor of the same torque.

For these reasons and more, **a.c. motor has been gradually replacing the d.c. motor in the recent few decades**. In the current servo motor market, the motors are mainly brushless motors.

1.3 Permanent Magnet Motors vs. Induction Motors

Motors can be classified into different categories when looking from different aspects. For example, by looking at whether the motor current is a.c. or d.c., motors can be classified as a.c. or d.c. motors. Alternatively, if the rotational speed of the magnetic field is different from the rotor speed, the motor can be recognized as an asynchronous motor (i.e. induction motor), if not, a synchronous motor. Synchronous motors can be further classified as non-exited motors (i.e. reluctance motors [6], hysteresis motors [6], permanent magnet motors) and DC-excited motors (i.e. direct current supply would be required for producing magnetic field). Another classification criteria as mentioned in [7], is the rotating motor or linear motor, which is determined by the movement generated by the motor.

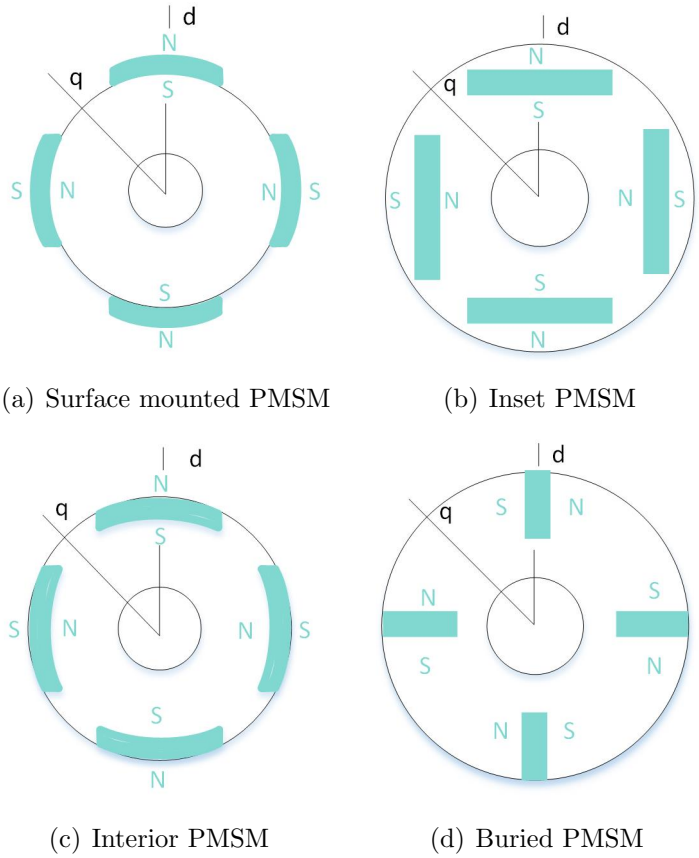
In current servo market, the majority of servo motor products are permanent magnet synchronous motors (PMSM) [4] including a.c. servo motors, brushless d.c. motors, linear motors. A drawback of PMSM can be the increasing price of permanent magnets. Both hysteresis motors and reluctance motors do not have permanent magnets. Hysteresis motors have solid smooth cylindrical rotor, and this can be an advantage in servo applications where constant speed is required. However, the efficiency of hysteresis motors is lower than PMSMs. Besides, reluctance motors would generate more torque ripple than PMSMs, and may not be suitable for servo applications. Comparing PMSMs with induction (asynchronous) motors (IMs), [8] gives the conclusions as in Table 1.2. Due to the synchronous nature of PMSM, its power factor can be 1, this benefits the efficiency and dynamic control of torque and speed. A larger air gap in PMSM than in IM can also help reducing the armature reaction. Moreover, a case study in [8] shows the overall loss of the PMSM drive used in that case is 43% smaller than the IM drive. Generally, the popularity of PMSM is increasing.

Table 1.2: Comparison between PMSM and IM

	PMSM	IM
Speed	Constant, independent of the load	As the load increases, the speed decreases
Power factor	Adjustable, operating at pf=1 is possible	Depends on the air gap, pf≈0.8-0.9 at rated load, pf≈1 at no load
Non-ferromagnetic air gap	From a fraction of mm to a few mms	As small as possible
Torque-voltage characteristic	Torque directly porportional to the input voltage	Torque directly proportional to the input voltage square
Cost	More expensive than IM	Cost effective motor

Depending on the configuration of magnets, PMSMs can be divided into three main categories: the surface mounted PMSM (i.e. SMPM), which has permanent magnets mounted around the surface of the rotor as shown in Fig.1.3(a); the inset PMSM as shown in Fig.1.3(b); and the interior PMSM (i.e. IPMSM) or buried PMSM, where the magnets are buried in the rotor as in Fig.1.3(c) and Fig.1.3(d). According to [8], the interior PM motors are more suitable for high frequency and high speed applications, since the magnets are well protected against centrifugal forces. Whereas, the electromagnetic force produced by the surface mounted PMs is higher than the one produced by the inset PMs. Moreover, the surface mounted PM and the buried PM are compared in details as in Table 1.3.

Another important difference between the different types of PMSMs is their saliency, that is the ratio between the highest and lowest magnetic permeabilities of the rotor in any two directions. For the surface mounted PMs, since the permeability of magnet and air are approximately the same, the d and q axis inductances are almost equal. Whereas, for the inset and interior PMSM, the q axis inductance is greater than the d axis inductance. Higher torque may be produced if the saliency of motor is greater. However, a pulsation torque [9] may also be produced due to the saliency. The torque production principles of motor will be demonstrated in details later in Chapter 2. **In this work, surface mounted PMs are used.**



(a) Surface mounted PMSM (b) Inset PMSM

(c) Interior PMSM (d) Buried PMSM

Figure 1.3: Rotor structures for three common types of PMSMs (permanent magnets are represented blue)

Table 1.3: The comparison between two types of PMSM

	SMPM	Buried PM
Air gap magnetic flux density	$< B_r$*	Can be $> B_r$* (with more than 4 poles)
Motor construction	Simple	More complicated than SMPM
Armature reaction flux	Small	Higher, Consequently more expensive converter
Magnets protection	Not protected against armature fields, centrifugal forces	Protected against armature fields, centrifugal forces
Eddy current losses	Have losses in PMs	No eddy-current losses in PMs
$*$ B_r - Residual flux density, the magnetic induction remaining in a saturated magnetic material after the magnetizing field has been removed.		

1.4 The Need for Torque Ripple Reduction

Force describes the tendency of pushing or dragging (i.e. linear movement) an object. When it comes to rotational movement, "torque" is used to describe the tendency of rotating an object [10]. In other words, the rotation of a motor is yielded by torque.

In servo motor drives, the desired control performance differs from application to application. A general desired drive feature is to achieve fast and accurate control responses. Since servo motors are commonly controlled under speed/position modes, this translates as fast speed/position tracking ability, for which purpose, vibrations or ripple both at steady state and during transients need to be minimized. Consequently, **torque ripple need to be minimized, or it can influence all controlled variables**.

The sources of torque ripple will be demonstrated later in Chapter 2.1; in summary, torque ripple is not only related to the nature of PMSM, but can also be originated from almost every part composing the closed loop system (Fig.1.2). The torque ripple issue is inevitable and therefore, has been studied for decades.

1.5 Methodology and Novelties of the Work

Considering the complexity of the topic, the adoption of a traditional control approach targeted at reducing the torque ripple individually caused by every possible source would raise the control complexity and might not achieve good results.

This work, on the contrary, will study and develop techniques based on automatic heuristic optimization and iterative learning approaches in conjunction with classical control techniques. The aim is to provide control solutions which can effectively reduce torque ripple in the wide scenario of industrial applications.

For this purpose, the following **three novelties** will be proposed:

- For studying, modelling, and simulating the torque ripple of PMSM in a more

accurate way, **an improved PMSM model has been developed using Matlab/Simulink** (Chapter 3). In the model, seven lookup tables are integrated for representing the non-linearties of PMSM in seven aspects. The lookup tables are derived from FE analysis results using Infolytica MagNet.

- **An enhanced deadbeat current controller has been developed** (Chapter 4). One enhancement is to use two steps current prediction, instead of traditional one step. It benefits high accuracy of current sampling. Another enhancement is to apply a rotor movement compensation strategy, which is useful for compensating the voltage offset issue.

- **A novel repetitive controller has been developed** (Chapter 5). First, its location has been carefully chosen in order to co-exist with the deadbeat current controller. Second, the repetitive control has been implemented in the angle-domain, instead of the traditional time-domain, since torque ripple is mainly rotor position dependent and not time dependent. Third, in order to improve the performance and eventually stability of the overall control system, an phase lead compensator has been implemented for compensating the delay of the proposed deadbeat current loop; a transient detector strategy has been developed especially for improving the transient performance.

Thanks to all these novelties, torque ripple reduction of PMSM has been achieved not only **at steady state**, but also **for variable speed/load conditions**, and maintained **even during speed/load transients**.

1.6 Outline of the Thesis

More in details the work will be looking, but not restricted to, the following broad points:

1. Modelling of permanent magnet motor to test torque ripple reduction schemes. (Chapter 3)

2. Implementation of standard vector control, including deadbeat control. (Chapter 4)

3. Investigation of existing methods for torque ripple reduction. (Chapter 5)

4. Background on repetitive control. (Chapter 5)

5. Development of new repetitive control algorithms and to compare with ones in literature. (Chapter 5)

6. Implement fast speed and current loops and assessment of the performance improvement with the torque ripple reduction. (Chapter 6)

7. Experimental rig set up. (Chapter 7)

8. Experimental results. (Chapter 7)

1.7 Summary

This work is in the field of electrical drives, and Fig.1.4 indicates the area of this work in the big picture of electrical drives. **In the following chapter, this area will be reviewed, in order to identify the requirements and focus the research needs for this work.**

It is worth pointing out that the so-called brushless d.c. motor is not "real" d.c.. It refers to a type of permanent magnet motor, of which the back-emf is in trapezoidal shape [11]. Its current is still a.c., and therefore different from the real d.c. motors with brush or commutator. This brushless d.c. motor is normally supplied using a block commutation voltage source inverter (square-wave voltage). On contrary, the brushless a.c. motor refers to the motor with sinusoidal shape back-emf, which is commonly driven by pulse width modulation (PWM) voltage source inverter (sine voltage).

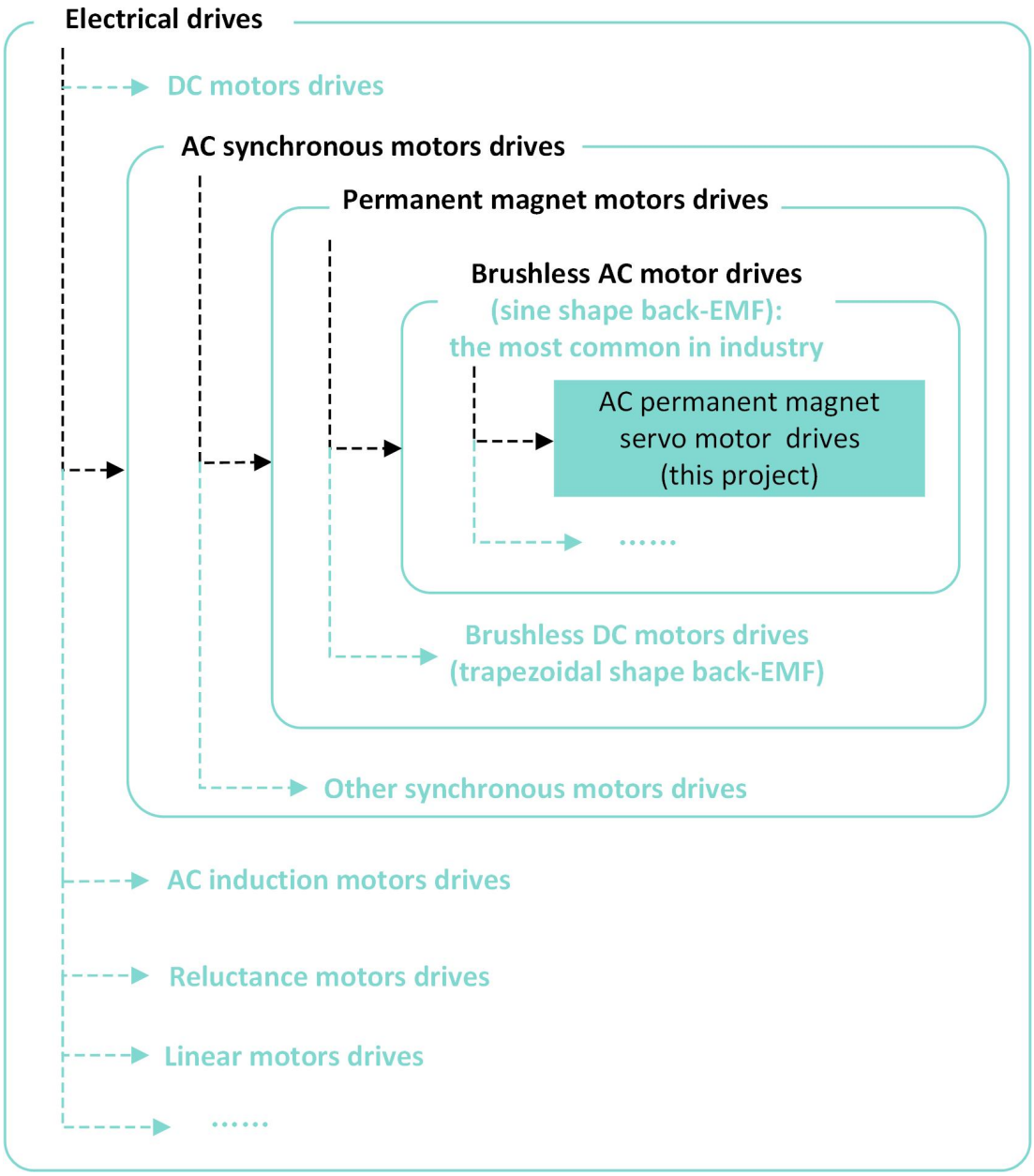


Figure 1.4: The area of this work in the big picture of electrical drives

Chapter 2

Literature Review

This chapter presents a review of existing torque ripple reduction technologies.

The main sources of torque ripple and their characteristics are summarized in 2.1; some existing simulation models for PMSM are introduced in 2.2; commonly used closed loop methodologies for motor drives are reviewed in 2.3; some existing torque ripple reduction technologies are then reviewed in 2.4 regarding machine design, and in 2.5 regarding control. Finally, based on this review, the necessary further research for this work are suggested in 2.6, which also explains the necessity of achieving the objectives as can be seen from 1.6.

2.1 Sources of Torque Ripple

Considering a typical closed loop servo motor control system as shown in Fig.1.2, unfortunately, torque ripple can potentially be generated from everywhere [12] (power converter, motor, load, and sensors/transducer/encoder for measuring feedback signals). In general, the majority of torque ripple is contributed by the PM motor (surface mounted in this work) itself. More details will be given in the following six subsections. For better explaining the characteristics of each type of torque ripple, It may be useful to refer to the general dynamic equations for PMSM which will be given later in Section 2.2.3 - Section

2.2.5.

2.1.1 Cogging Effect of PMSM

The cross-section of an example surface mounted PMSM can be seen in Fig.2.1. Where, **iron** covers the most of the stator and rotor parts, **permanent magnets** are mounted on the surface of rotor, the parts filled with sign of coils are the slots. Since the magnetic permeability of a slot (practically filled with air and coils) is lower than that of a tooth (iron), the teeth can support the formation of magnetic field in a stronger way than the slots. Therefore, the cogging effect can be easily understood as the PM may attract teeth stronger than slots. Such differences in attraction would yield a torque which may act against the rotation at some positions, and act with it at others. This uneven torque generation is called the **cogging torque**.

As explained above, cogging torque is position dependent. Some other **facts about cogging torque** can be summarized as the following:

1) Since the magnets would be always "trying" to align with the stator tooth, the frequency of cogging torque is determined by the number of slots/tooth and poles. For instance, considering a 18 slots/teeth, 3 pole pairs machine, each pair of magnets would "see" 6 slots and teeth per electrical cycle. Consequently, the frequency of cogging torque would be 6 times that of the machine fundamental electrical frequency or electrical speed. In other words, 18 times the mechanical speed.

2) The cogging effect is about the interaction between the rotor magneto-motive force (MMF) and the stator reluctance. It is not necessarily related to the electromotive force (EMF) generated by the stator current. Therefore, the cogging torque would be present even when the stator current is zero. The stator current would only be responsible for an amplitude change in the cogging torque which is usually small.

3) The cogging effect can be stronger on one machine than another, depending on the machine design. However, a typical value for the cogging torque can be 10% of the rated torque [13], which makes it the main source of torque ripple for a PM motor.

4) Since the frequency of cogging torque increases proportionally with the rotor speed, at high speed the high frequency cogging torque would be filtered out by the inertia of the mechanical system (which reacts like a low-pass filter). Consequently the cogging torque effect is more obvious at low speed. Especially, because the bandwidth of the speed loop is normally low when compared to the frequency of the cogging torque, cogging torque would cause more problems if its frequency lies in between the bandwidth of the speed loop and the cut-off frequency of the mechanical plant.

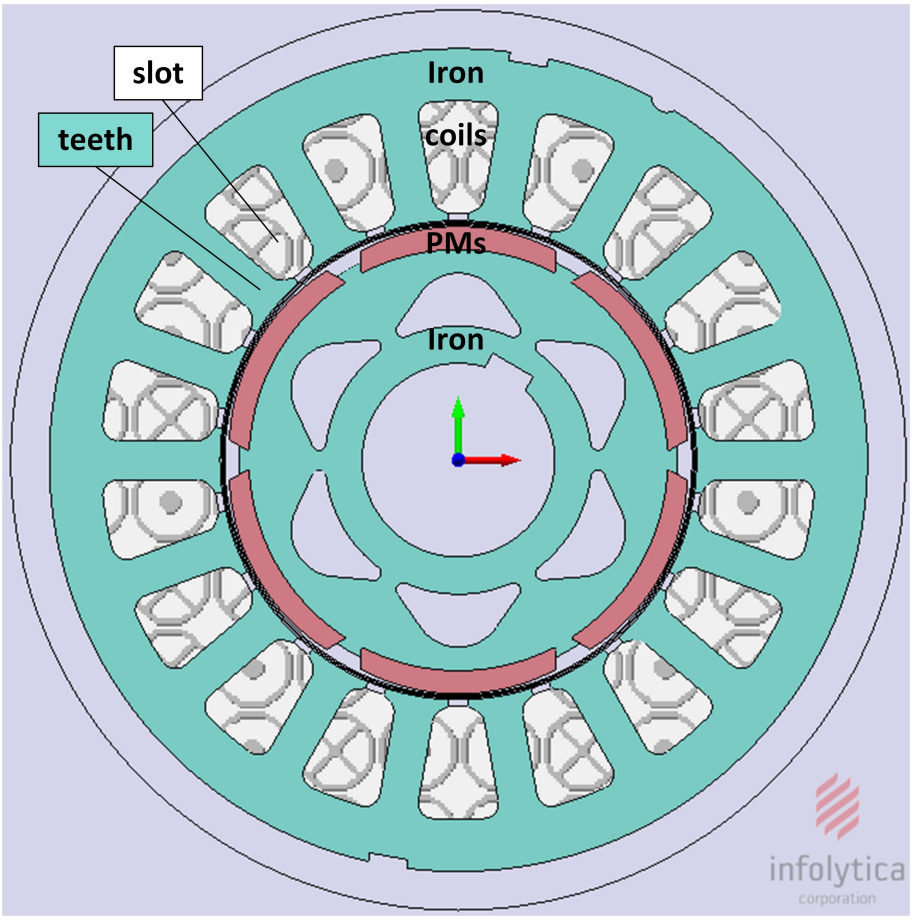


Figure 2.1: Cross-section of the surface mounted PMSM used for modelling purposes in this work

2.1.2 Flux Harmonics in PMSM

The flux harmonics are another source of torque ripple from PMSM. Ideally, the magnetic flux distributed along the air gap should be purely sinusoidal for a brushless a.c. motor. The presence of flux harmonics indicates non-ideal flux distribution. It can be caused

by imperfections in stator windings (necessary effect of imperfect sinusoidal distribution), or parameter variations of the machine due to demagnetization, saturation, temperature and so on.

Considering the magnetic flux of one phase distributed along the rotor, it becomes more like a "rectangular" shape, rather than the ideal "sinusoidal" shape. The resultant three-phase fluxes would contain odd harmonics ($5^{th}, 7^{th}, 11^{th}, ...$), since the Fourier series of a square wave contains all the odd harmonics (refer to Appendix A), whereas the 3^{rd} and its multiple harmonics are not considered in the following since they cannot produce torque ripple in a star connected three-phase system.

After the Park transformation (refer to Appendix B), these odd harmonics under the stator three-phase frame would be responsible for the $(5 \pm 1)^{th}, (7 \pm 1)^{th}, (11 \pm 1)^{th}$, ... harmonics in the rotor dq axis frame. For better explaining this, taking the n^{th} harmonics in the three-phase fluxes for example (assuming the ripple fluxes to be $F_{an}sin(n\theta_e), F_{bn}sin(n\theta_e - \frac{2n\pi}{3}), F_{cn}sin(n\theta_e + \frac{2n\pi}{3})$), the corresponding d axis flux F_{dn} will contain the $(n-1)^{th}$ and $(n+1)^{th}$ harmonics as can be seen in (2.1).

$$
\begin{aligned}
F_{dn} &= \frac{2}{3}[F_{an}sin(n\theta)sin(\theta_e) + F_{bn}sin(n\theta_e - \frac{2n\pi}{3})sin(\theta_e - \frac{2\pi}{3}) + F_{cn}sin(n\theta_e + \frac{2n\pi}{3})sin(\theta_e + \frac{2\pi}{3})] \\
&= \frac{2}{3}F_{an}\frac{cos[(n-1)\theta_e] - cos[(n+1)\theta_e]}{2} \\
&+ \frac{2}{3}F_{bn}\frac{cos[(n-1)(\theta_e - \frac{2\pi}{3})] - cos[(n+1)(\theta_e - \frac{2\pi}{3})]}{2} \\
&+ \frac{2}{3}F_{cn}\frac{cos[(n-1)(\theta_e + \frac{2\pi}{3})] - cos[(n+1)(\theta_e + \frac{2\pi}{3})]}{2}
\end{aligned}
\tag{2.1}
$$

As a result, the torque produced by such dq axis fluxes may also contain the $4^{th}, 6^{th}, 8^{th}, 10^{th}, 12^{th}, ...$ harmonics according to the dynamic equation for machine electrical torque (2.16).

2.1.3 Current Offset Error in Sampling

As well as the PMSM itself, the sensors/transducers errors can also contribute to torque ripple. Based on the field oriented control (FOC)/vector control (VC) (will be demonstrated later in 2.3), the current controller requires the dq axis currents as feedbacks. Practically, the three-phase currents are measured by sensors, transformed into digital signals using A/D converters and then computed into the dq axis values for the controller. Hence, any unbalances in the current sensors or/and any inherent offset in the analog devices can cause a rotor **angle-based** displacement in the dq axis currents.

For instance, considering there are offset errors in the measured i_a and i_b, and assuming the offset to be i_a^{off} and i_b^{off}, according to the Park transformation (Appendix B), the corresponding offset in i_q, i_q^{off}, would be as in (2.2). It is worth pointing out that, in practice, it is common that only 2 current sensors are used for measuring the phase A and B currents, whereas the phase C current can be computed from i_{ab}. In this way, the cost for sensors can be reduced.

$$\begin{aligned} i_q^{off} &= \frac{2}{3}[i_a^{off}cos(\theta_e) + i_b^{off}cos(\theta_e - \frac{2\pi}{3})] \\ &= \frac{2}{3}[i_a^{off}cos(\theta_e) + i_b^{off}cos(\theta_e)cos(\frac{2\pi}{3}) + i_b^{off}sin(\theta_e)sin(\frac{2\pi}{3})] \\ &= \frac{2}{3}[\frac{\sqrt{3}}{2}i_b^{off}sin(\theta_e) + (i_a^{off} - \frac{i_b^{off}}{2})cos(\theta_e)] \\ &= \frac{2}{3}sin(\theta_e + \varphi)\sqrt{(\frac{\sqrt{3}}{2}i_b^{off})^2 + (i_a^{off} - \frac{i_b^{off}}{2})^2} \end{aligned} \tag{2.2}$$

Where, the angle $\varphi = arctan(\frac{2i_a^{off} - i_b^{off}}{\sqrt{3}i_b^{off}})$. As it can be seen, the offset in phase A and B curents would contribute to a 1^{st} order harmonic in the q axis current. Consequently, the torque produced by such i_q current harmonics would also contain a 1^{st} order harmonic. Therefore, even if there is current control offset due to the current sampling offset, it is still possible to achieve smooth torque by tailoring the current reference (e.g. adding the 1^{st} harmonic in i_q reference).

2.1.4 Current Scaling Error in Sampling

The scaling error in the A/D converter can be another cause for torque ripple. Again, in case only two current sensors are used for phase A and B, taking into account the scaling factor K_a, K_b ($i_a = I^{peak}cos(\theta_e)/K_a, i_b = I^{peak}cos(\theta_e - 2\pi/3)/K_b$, where I^{peak} is the peak value of phase current), the corresponding i_q can be obtained as in (2.3).

$$\begin{aligned} i_q^{off} &= \frac{2}{3}[i_a cos(\theta_e) + i_b cos(\theta_e - \frac{2\pi}{3})] \\ &= \frac{2}{3}[\frac{I^{peak}cos(\theta_e)}{K_a}cos(\theta_e) + \frac{I^{peak}cos(\theta_e - 2\pi/3)}{K_b}cos(\theta_e - \frac{2\pi}{3})] \\ &= \frac{2}{3}[\frac{I^{peak}}{K_a}(1 + cos(2\theta_e)) + \frac{I^{peak}}{K_b}(1 + cos(2\theta_e - \frac{4\pi}{3}))] \end{aligned} \tag{2.3}$$

As it can be seen, the resultant q axis current would contain the 2^{nd} harmonic due to the scaling error of A/D converter. Consequently, the torque produced by such q axis current would contain the 2^{nd} harmonic as well. Therefore, even if there is current control error due to the current sampling scaling error, it is still possible to achieve smooth torque by tailoring the current reference (e.g. adding the 2^{nd} harmonic in i_q reference).

2.1.5 Current Harmonics in Converter

According to [14], many available converter topologies for drive applications can be classified into four types: 1) a.c. to d.c. converters; 2) d.c. to d.c. converters; 3) a.c. to a.c. converters with intermediate d.c. link; 4) direct a.c. to a.c. converters. Fig.2.2 below shows a brief summary of converter topologies available for drives. The pulse-width modulated two-level voltage source inverter (as shown in Fig.2.3) is used in this work, since this is widely used in the servo drive market in conjunction with brushless a.c. PMSMs.

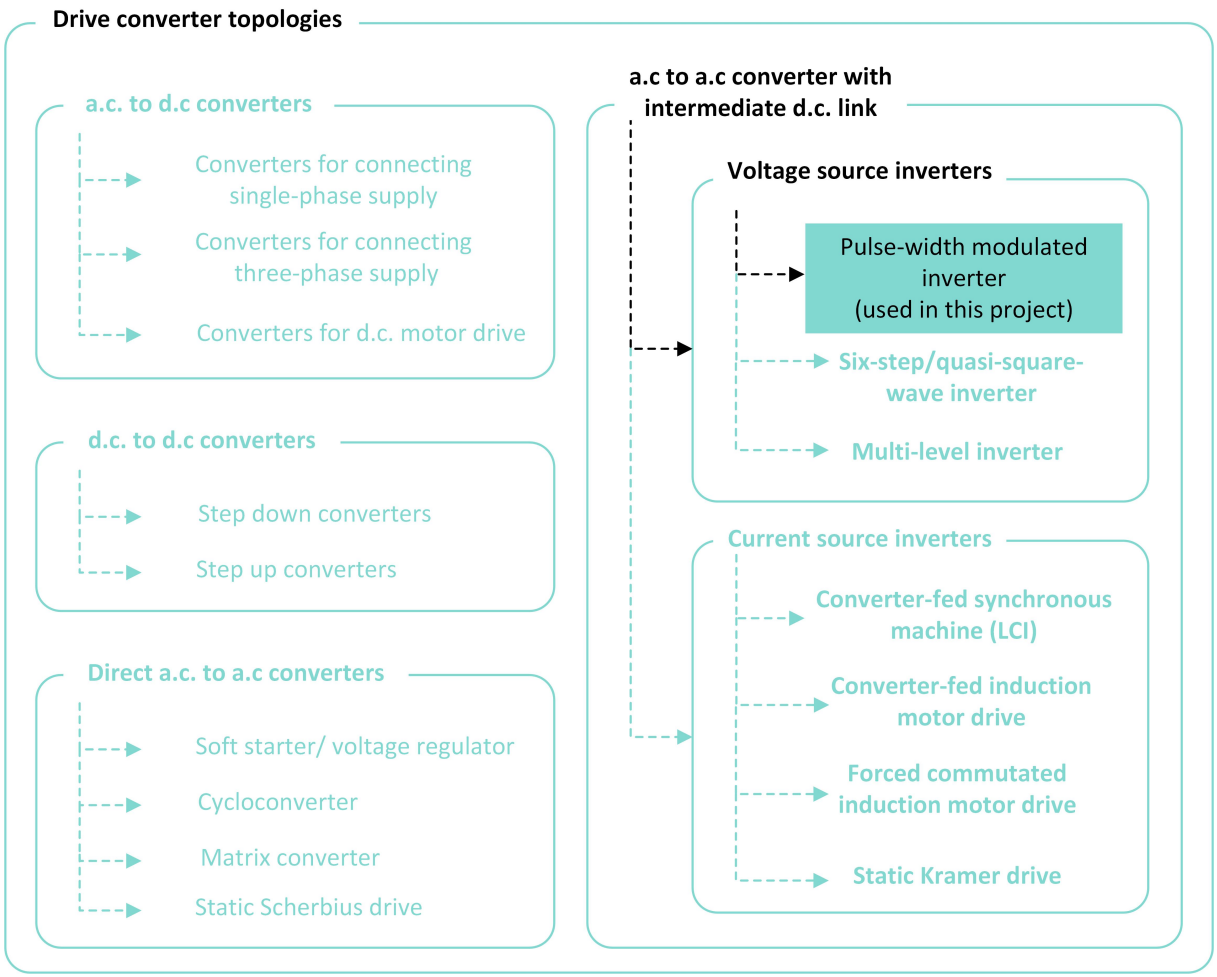


Figure 2.2: Available drive converter topologies

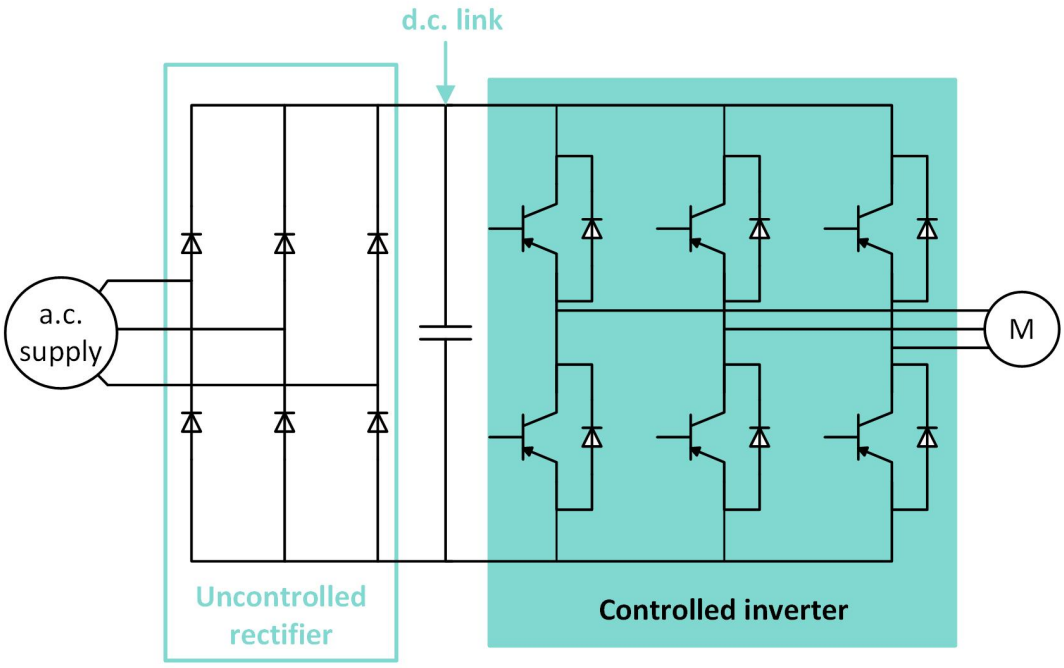


Figure 2.3: Schematic diagram of a two-level voltage source inverter

The pulse-width modulated inverter consists of an uncontrolled diode bridge rectifier, an intermediate d.c. capacitor, and a controlled inverter. It is called the pulse-width modulated inverter since the six gate signals for the six switches of the inverter are pulse-width modulated signals. The operating principle of the three-phase diode bridge rectifier, the three-phase inverter, and the pulse width modulation (PWM) can be found in[15]. They are not the key considerations in this work, hence will not be demonstrated here. Importantly, for considerations about torque ripple, there are inherited $6^{th}, 12^{th}, 18^{th}, 24^{th}, ...$ order harmonics in the d.c. voltage due to the rectifier. According to [16], this 6^{th} harmonic at d.c. side will be responsible for the $5^{th}, 7^{th}, 11^{th}, 13^{th}, 17^{th}, 19^{th}, 23^{th}, ...$ harmonics in the three-phase currents at a.c. side. Again, after applying the Park Transformation, the dq axis current would contain $4^{th}, 6^{th}, 10^{th}, 12^{th}, 14^{th}, 16^{th}, 18^{th}, 20^{th}, ...$ harmonics. Hence, these harmonics can possibly appear in torque as well.

2.1.6 Mechanical Imperfections

Last but not least, any imperfections in the mechanical system, such as misalignment between the motor and its load, or worn bearings, can also lead to serious torque ripple. Such torque ripple is very common in real-life applications and would be difficult to predict, therefore it is also difficult to know in advance its frequency since it is related to the particular features of the imperfection. However, fortunately, it can be seen that the resultant torque ripple is periodic with the rotor position.

2.2 Existing PMSM Models for Simulation

As it has been discussed above, the cogging effect and flux harmonics are two major sources of torque ripple in the PMSM caused by the servo drive itself. It is therefore necessary to review the existing simulation models for PMSM, in order to include cogging effects and flux harmonics.

This topic can be discussed according to three types of modelling approaches in

the existing literature: 1) the finite element model approach; 2) the equivalent circuit approach, such as the lumped parameter model or the permeance network; 3) the simplified model in *abc* coordinates, $\alpha\beta$ coordinates, and *dq* coordinates (The definition of these three coordinates can be seen as in Appendix B). The details of each model will be introduced in the following subsections. In general, the accuracy of the model decreases from 1) to 3), while the computation speed increases.

2.2.1 Finite Element Model

The finite element model can be the most accurate model available. In this approach, the machine is subdivided into very small pieces, or meshes, called finite elements. The more intensive the meshes, the more accurate the model will be. The model is solved by solving the boundary value problems for partial differential equations, for each small piece of the machine.

Not only are the cogging torque and flux harmonics properly simulated, due to the accuracy of such finite element analysis (FEA), but also there are more effects the FE model can simulate. Ref [17] uses FEA to calculate the iron loss of PMSM; Ref [18] focuses on the finite element demagnetization analysis for PM; the rotor loss of PMSM is calculated in [19]; the stator eddy current is computed in [20] and [21]; Ref [22] studies the PMSM under short circuit condition. Ref [23] presents the thermal analysis using FEA.

However, it is relatively inconvenient to use the FE model due to the long computation time (can be days or weeks).

2.2.2 Equivalent Circuit Model: Lumped Parameter Model and Permeance Network Model

One common thing shared by the lumped parameter model [24–30] or permeance network model [31] [32] and the FE model is that they all subdivide the machine into pieces.

However, in the lumped parameter model or permeance network model, the pieces are divided according to the reluctances or permeability of the material. In details:

- The permeability of rotor magnets, air gap, slots and teeth can be represented by many series or parallel connected reluctances.
- The magneto motive force (MMF) produced by rotor magnets, and the electro motive force (EMF) produced by the stator currents can be represented by voltage sources.

One advantage of this kind of model is that, the reluctances can be time and position varying. Therefore, the saturation effect of the magnets, as well as the variation of the air gap length at different positions can be taken into account. The accuracy of this kind of model depends on quantity of reluctances in the network, and calculation accuracy of each reluctance value. It is common to use FE model as a bench mark. As can be seen from [33], the lumped parameter model can possibly achieve comparable accuracy with the FE model.

The main disadvantage of this modelling approach is the difficulty of accurately calculating the reluctances.

2.2.3 Simplified Model in *abc* Coordinates

The dynamic equations of the PMSM can be written based on the standard electromagnetic and mechanical laws. Under normal operations, the following **assumptions** are generally acceptable in order to simplify the model:

Assumption 1: The stator winding is sinusoidal distributed, therefore the electromagnetic force is sinusoidal.

Assumption 2: The iron material is supposed linear, so neglecting hysteresis and saturation. Therefore, the dq axis inductances do not vary with current, and losses caused by magnetic hysteresis are not considered.

Assumption 3: The eddy current losses are ignored and therefore, the permanent magnet field in the air gap is assumed to be sinusoidal. The direction of PM magnetic field is aligned with the d axis, the q axis component of magnetic flux is neglected.

Assumption 4: As a consequence of the first three, the mutual inductances between stator and rotor can be seen as a sinusoidal function of electrical position, therefore the dq axis inductances can be considered as constant values.

Assumption 5: The torque generated by the motor is calculated by the electromagnetic power assuming co-energy equals to the energy stored in the magnetic field, since the system is assumed to be lossless (Assumption 3) and linear (Assumption 2).

As a consequence of the above assumptions, **the cogging effect and magnetic flux harmonics are all ignored**. However, the **computation time is significantly reduced** due to the resulting simplifications. The dynamic equations of PMSM in abc coordinates are as follows:

$$\begin{bmatrix} v_a \\ v_b \\ v_c \end{bmatrix} = R_s \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} + \frac{\mathrm{d}}{\mathrm{d}t} \begin{bmatrix} F_a \\ F_b \\ F_c \end{bmatrix} \tag{2.4}$$

$$\begin{bmatrix} F_a \\ F_b \\ F_c \end{bmatrix} = \begin{bmatrix} L_a & L_{ab} & L_{ac} \\ L_{ab} & L_b & L_{bc} \\ L_{ac} & L_{bc} & L_c \end{bmatrix} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} + F_m \begin{bmatrix} cos(\theta_e) \\ cos(\theta_e - \frac{2\pi}{3}) \\ cos(\theta_e + \frac{2\pi}{3}) \end{bmatrix} \tag{2.5}$$

Where:

- v_{abc} denote the three-phase stator voltages, which are supplied by the three-phase converter in the drive system.
- i_{abc} denote the three-phase stator currents, which is the control target in the drive system.
- F_{abc} denote the three-phase stator flux linkages, which is the sum of the stator fluxes

generated by the three-phase currents and the mutual flux between the PMs and stator fluxes. According to the Assumption 3 above, the mutual flux is represented as the cosine function times the PM flux F_m as can be seen in (2.5).

- R_s is the stator resistance, normally considered to be constant.
- L_{abc} represent the stator self inductances of the three phases, L_{ab}, L_{bc}, L_{bc} are the mutual inductances between each phase. According to Assumption 4 above, these inductances are all considered as sinusoidal functions of the electrical position (2.6).

$$\begin{aligned} L_a &= L_0 + L_1 cos(2\theta_e) \\ L_b &= L_0 + L_1 cos(2\theta_e + \frac{2\pi}{3}) \\ L_c &= L_0 + L_1 cos(2\theta_e - \frac{2\pi}{3}) \\ L_{ab} &= -\frac{L_0}{2} + L_1 cos(2\theta_e - \frac{2\pi}{3}) \\ L_{ac} &= -\frac{L_0}{2} + L_1 cos(2\theta_e + \frac{2\pi}{3}) \\ L_{bc} &= -\frac{L_0}{2} + L_1 cos(2\theta_e) \end{aligned} \tag{2.6}$$

- θ_e is the electrical position, which equals the number of pole pairs times the mechanical position $P_p\theta_m$.

The electrical torque yielded by the machine can be calculated from the output electrical power of the machine as in (2.7). Alternatively, the torque can also be expressed according to the dynamics of the mechanical system as in (2.8).

$$T_e = \frac{P_e P_p}{\omega_m} = \frac{P_p}{2} \begin{bmatrix} i_a & i_b & i_c \end{bmatrix} \frac{\partial}{\partial \theta_e} \begin{bmatrix} F_a \\ F_b \\ F_c \end{bmatrix} \tag{2.7}$$

$$T_e = J\frac{\omega_m}{dt} + B\omega_m + T_l \tag{2.8}$$

Where, P_e denote the electromagnetic power responsible for the electromechanical conversion, J is the moment of inertia, B is the friction factor. Both of these two parameters are determined by the mechanical system. T_l is the load torque. Since (2.7) and (2.8)

should be equal, the dynamic of rotor mechanical speed ω_m can be modelled by solving the two equations.

2.2.4 Simplified Model in $\alpha\beta$ Coordinates

The corresponding dynamic equations of PMSM in $\alpha\beta$ coordinates can be obtained by applying the Clarke transformation (Appendix B).

$$\begin{bmatrix} v_\alpha \\ v_\beta \end{bmatrix} = R_s \begin{bmatrix} i_\alpha \\ i_\beta \end{bmatrix} + \frac{\mathrm{d}}{\mathrm{d}t} \begin{bmatrix} F_\alpha \\ F_\beta \end{bmatrix} \tag{2.9}$$

$$\begin{bmatrix} F_\alpha \\ F_\beta \end{bmatrix} = \begin{bmatrix} L_\alpha & L_{\alpha\beta} \\ L_{\alpha\beta} & L_\beta \end{bmatrix} \begin{bmatrix} i_\alpha \\ i_\beta \end{bmatrix} + F_m \begin{bmatrix} cos\theta_e \\ sin\theta_e \end{bmatrix} \tag{2.10}$$

$$\begin{aligned} L_\alpha &= L_0 + L_1 cos(2\theta_e) \\ L_\beta &= L_0 - L_1 cos(2\theta_e) \end{aligned} \tag{2.11}$$

$$T_e = \frac{3}{2} P_p (F_\alpha i_\beta - F_\beta i_\alpha) \tag{2.12}$$

Where, $v_{\alpha\beta}$ are the supply stator voltages at $\alpha\beta$ axis. $i_{\alpha\beta}$ are the equivalent stator currents at $\alpha\beta$ axis. $F_{\alpha\beta}$ are the $\alpha\beta$ axis flux linkages. $L_\alpha, L_\beta, L_{\alpha\beta}$ are the equivalent self and mutual inductances.

2.2.5 Simplified Model in dq Coordinates

The dynamic equations of the PMSM in dq coordinates can be obtained after applying the Park transformation (Appendix B) to (2.4)-(2.5).

$$\begin{bmatrix} v_d \\ v_q \end{bmatrix} = R_s \begin{bmatrix} i_d \\ i_q \end{bmatrix} + \frac{\mathrm{d}}{\mathrm{d}t} \begin{bmatrix} F_d \\ F_q \end{bmatrix} + \omega_e \begin{bmatrix} -F_q \\ F_d \end{bmatrix} \tag{2.13}$$

$$\begin{bmatrix} F_d \\ F_q \end{bmatrix} = \begin{bmatrix} L_d & 0 \\ 0 & L_q \end{bmatrix} \begin{bmatrix} i_d \\ i_q \end{bmatrix} + \begin{bmatrix} F_m \\ 0 \end{bmatrix} \tag{2.14}$$

$$\begin{aligned} L_d &= L_0 + L_1 \\ L_q &= L_0 - L_1 \end{aligned} \tag{2.15}$$

$$T_e = \frac{3}{2} P_p (F_d i_q - F_q i_d) \tag{2.16}$$

Where, v_{dq} are the supply stator voltages at dq axis. i_{dq} are the equivalent stator currents at dq axis. F_{dq} are the dq axis flux linkages. L_d, L_q are the dq axis inductances. As it can be seen from (2.14), the magnetic flux F_m only has d axis component according to the above mentioned Assumption 3. The equations can be further simplified by substituting (2.14) into (2.13) and (2.16).

$$\begin{bmatrix} v_d \\ v_q \end{bmatrix} = \begin{bmatrix} R_s & -\omega_e L_q \\ \omega_e L_d & R_s \end{bmatrix} \begin{bmatrix} i_d \\ i_q \end{bmatrix} + \begin{bmatrix} L_d & 0 \\ 0 & L_q \end{bmatrix} \frac{\mathrm{d}}{\mathrm{d}t} \begin{bmatrix} i_d \\ i_q \end{bmatrix} + \begin{bmatrix} 0 \\ \omega_e F_m \end{bmatrix} \tag{2.17}$$

$$T_e = \frac{3}{2} P_p [F_m i_q + (L_d - L_q) i_d i_q] \approx \frac{3}{2} P_p F_m i_q \tag{2.18}$$

In the case of a surface mounted PMSM, the dq axis inductance are considered to be equal. **Therefore, it is common to control the d axis current to be zero, and the torque control of a surface mounted PMSM can be achieved through the control of the q axis current.**

Equations (2.13), (2.17) and (2.18) have been used for building the "Permanent Magnet Synchronous Machine" model in the Simulink library, in which (2.8) has been used for modelling the mechanical system.

2.3 Existing Closed Loop Control Methodologies for PMSM Drives

A rich variety of closed loop control methodologies can be found in the existing literature. Some of the most frequently mentioned control methods are listed in the following; their main advantages and disadvantages are summarized in Table 2.1.

- **Field oriented control (FOC)** [5,34–36] belongs to the vector control (VC) family. The vector control has became a "classic" control method for a.c. motors. It was first pioneered around 1970 by K. Hasse from Technische Universität Darmstadt and F. Blaschke from Siemens. Its development history can be found in [37]. FOC is a closed loop control scheme with cascaded loops as shown in Fig.2.4:

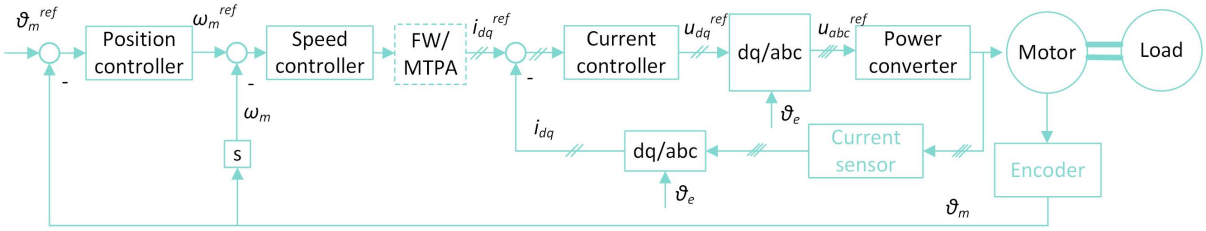


Figure 2.4: Block diagram of field oriented control

Where, the FW (field weakening) and MTPA (maximum torque per ampere) are two control features which can be used if necessary. FW [38,39] is useful if the motor need to operate at a speed much higher than the rated speed. MTPA [39,40] is more useful for salient machine, whereas for surface mounted PMSM, MTPA translates to control $i_d = 0$. The current controller can be **hysteresis controller**, **PI controller** or **deadbeat (DB) controller**. In case of using DB controller, high bandwidth and fast dynamic can be achieved [34]. The abc/dq, dq/abc transformations are demonstrated in Appendix B.

As it can be seen, the demand voltages v_{abc}^{ref} for three phases are generated by the current controller, and such voltage references need to be achieved by the power converter which supplies the motor. For this purpose, extra modulation methods are required for controlling the terminal voltage of the converter. Commonly used

modulation strategies can be pulse width modulation (PWM) [15] or space vector PWM (SVPWM) [41].

It is worth pointing out that the deadbeat current control can also be found as simply predictive current control (PCC) [42], since DB control belongs to the predictive control family. The controller predicts current based on the machine electrical model, and computes a voltage reference which is then implemented by the inverter using pulse width modulation (PWM). Therefore, it is named as **PWM predictive control (PPC)** and classified as a type of model predictive control in [43]. However, it is very different from the other two types of model predictive control mentioned in [43]. More details about the other two types of model predictive control will be reviewed below.

- VC includes not only the FOC above, but also the **Direct torque flux control (DTFC)** [34–36]. The block diagram of DTFC for PMSMs is given in [34] and can be seen from Fig.2.5.

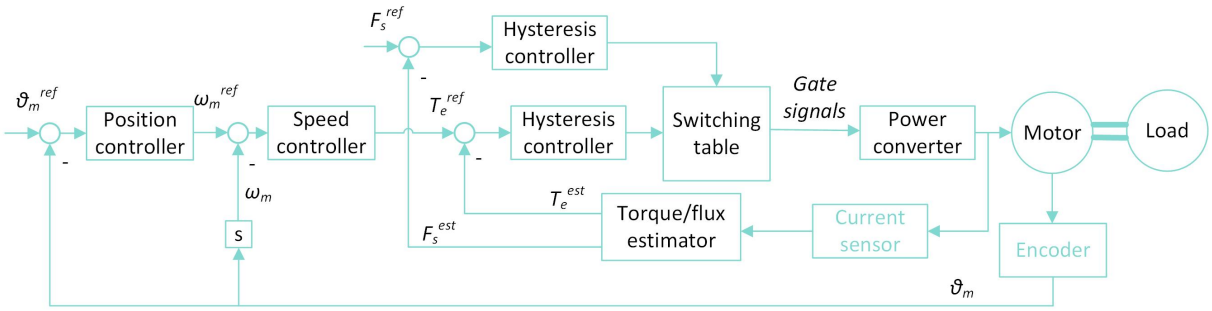


Figure 2.5: Block diagram of direct torque flux control

It can be understood by comparing Fig.2.4 and Fig.2.5 that the speed and torque(current) control in DTFC can be achieved without position information, which allows DTFC to be used for sensorless applications. It has been reported in [36] that DTFC can achieve faster dynamic performance comparable with PI-based FOC at the expense of some flux and torque ripple. Overall, **FOC is chosen for this work instead of DTFC since FOC naturally has smaller torque ripple.**

- **Model predictive control (MPC)** [43–46], MPC includes any control methods that predict the future response of a plant based on its model. Three types of MPC for PMSM current control are compared in [43]. The first type of MPC is called

direct predictive control (DPC). It controls directly the switching signals of the inverter. It computes the optimized switching signals by solving a predetermined cost function in every sampling period. The second type of MPC is called **two-configuration predictive control (2PC)**. It computes not only the switching sequence, but also the number of commutations per modulation period as well as the duration for each commutation. During each modulation period, the switching sequence is applied for the calculated duration, and null voltage is applied for the rest of the switching period. The paper nominates DB control as the third type of MPC. However, it is worth pointing out that both DPC and 2PC are based on the minimization of a cost function, whereas DB does not have cost function at all. All these three types of control requires model information, however **DB is very different from the other two**. The findings in [43] about using DPC, 2PC and DB for PMSM current control can be seen from Table 2.1.

- FOC, DTFC, MPC all requires accurate knowledge of machine parameters. In order to make the control more adaptive to parameter variations, some "smart" control logic has been proposed. **Neural network control (NNC)** [47] is one of these "smart" control methods. [47] presents the basic diagram for NNC (Fig.2.6). Briefly, the controller is able to learn an uncertain function using a way similar to the neural network of human beings. It is all about gradually training the NN controller by comparing the real output of the plant and the output of the NNC model, in order to approximate the real model eventually.

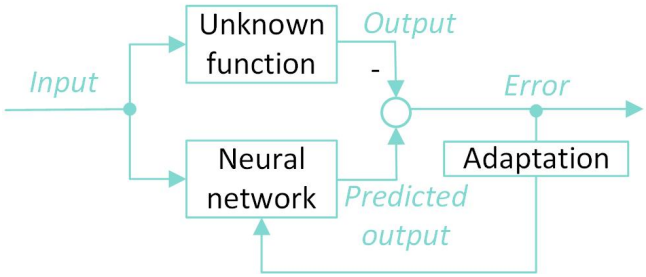


Figure 2.6: Block diagram of neural network control

Examples of using NNC for PMSM drives can be found in [48, 49]. [50, 51] propose a fuzzy NNC for PM drives. The advantages and disadvantages are summarized in Table 2.1 below.

- **Sliding mode control (SMC)** [52] is another adaptive control scheme which can cope with parameter variations, and potentially reject disturbances completely [52]. It copes with a varying control plant by varying the control structure accordingly. Therefore, it is a variable structure control. Examples of applying SMC for PMSM drives can be found in [53–56], where [56] presents a fuzzy SMC. Again, the main advantages and disadvantages are shown in Table 2.1 below.

- **Fuzzy control (FC)** [50, 51, 56–60] is to generate the control output according to a logic or a rule. It is also called fuzzy logic control (FLC). An example of a fuzzy control logic can be [57]:

  1) if error (E) is positive small (PS) and change of error (CE) is positive big (PB), then change of output (CO, i.e. the output of the controller) is negative medium(NM).

  2) if E is zero, CE is zero, then CO is zero.

  Phrases such as "PS", "PB", "NM" are called fuzzy variables, of which the value range need to be defined. Thanks to its fuzzy logic, FC is able to handle uncertain information. In PMSM drives, FC is found to be used in conjunction with NNC [50, 51] or SMC [56], the benefits of such combinations are as shown in Table 2.1 below.

- Besides NNC, two other self-learning control methods are found to be used for PMSM drives are **iterative learning control (ILC)** [61–63] or **repetitive control (RC)** [64, 65]. ILC and RC are compared in [66].

  ILC was initially proposed in 1978 by Uchiyama in Japanese [61]. It was first developed for industrial robotics, where the same control routines need to be repeated again and again. The nature of ILC is " not to make the same mistake again". For this purpose, the control output is always adjusted according to the information of control error during the previous operating routine. Eventually, after a few routines, the control output will be adjusted to the proper value, and the control error will

converge to zero. The general formulation of ILC is described as (2.19) in [66].

$$\begin{aligned} &u(t,k) = Q_{ILC}(u(t,k-1)) + L_{ILC}(e(t,k-1)) \\ &t = 0,1,...,T-1; k = 1,2,... \end{aligned} \tag{2.19}$$

Where, $u(t,k)$ denotes the output of the ILC controller at time t, during the k^{th} routine. T is the length of each routine. $e(t,k-1)$ denotes the control error at t, during the previous (i.e. the $(k-1)^{th}$) routine. Function $Q_{ILC}(\cdot)$ is called the **Q-filter of ILC**. In case $Q_{ILC}(u) = Q \times u$, where Q is a scalar between zero and one. Q is called the **forgetting factor**. Function $L_{ILC}(\cdot)$ is called the **L-filter of ILC**. If $L_{ILC}(e) = K_{ILC} \times e$, the ILC is classified as **P-type**.

RC was originally proposed in 1981 in motion control application, in order to track periodic reference [64]. It can also be used in PMSM drives for rejecting periodic disturbances. Interestingly, RC is designed in a total different way, but ending up sharing the same nature with ILC. The general transfer function of RC is described as (2.20) in [66].

$$U(z^{-1}) = \frac{z^{-T} L_{RC}(z^{-1})}{1 - z^{-T} Q_{RC}(z^{-1})} E(z^{-1}) \tag{2.20}$$

Where, function $Q_{RC}(z^{-1})$ is the **Q-filter of RC**. Function $L_{RC}(z^{-1})$ is the **L-filter of RC**. T is the known period of the periodic reference need to be followed, or the period of the disturbance need to be rejected. These will be better described later in Chapter 5. (2.21) can be acquired by converting (2.20) from frequency-domain (z-domain) into time-domain.

$$u(t) = Q_{RC}(t)u(t-T) + L_{RC}(t)e(t-T) \tag{2.21}$$

Comparing (2.19) and (2.21), we can see that, **RC and ILC can be exactly equivalent to each other**. RC have been applied successfully for torque ripple reductions or periodic reference tracking in PMSM drives. The most convenient thing of using them is that it can be designed without any parameters of the machine. Only the period of the ripple or the reference is needed. **This makes RC a promising solution for this work.** More details of using it for torque ripple reduction will be discussed later in 2.5.

Table 2.1: Potential advantages and disadvantages of different control methodologies for PMSM drives

Methods	Advantages	Disadvantages
FOC	+ Good steady state performance + Classic, proved technology	- Saturation in voltage may degrade current control performance (field-weakening region) - Assuming sine flux distribution, not accurate for the factional slot concentrated winding - System parameters are required
DTFC	+ Good steady state performance + Faster torque transient response than FOC + Eliminated rotor position requirement + Less parameter dependence than FOC	- Bigger torque and flux ripples than FOC (can be improved by other compensation) - Assuming sine flux distribution, not accurate for the factional slot concentrated winding - System parameters are required - Variable switching frequency
MPC	+ DPC can expand the operating region of PMSM + DPC can be faster than DTFC + DPC can reduce the limitations on inverter output + PPC(DB) has fixed switching frequency + PPC(DB) has smaller current ripple than DPC and 2PC	- DPC and 2PC have variable switching frequency, 2PC has fixed modulation period - Accurate knowledge of system parameter is required
NNC	+ Learning capability from processes + Adaptive	- Large amount of representative data need to be collected in advance - Too sensitive to disturbances
SMC	+ Less sensitive to load disturbances than NNC, take variations as disturbances + Adaptive, online parameter identification	- Chattering problems - Large amount of calculation
FC	+ Do not based on precise system model and parameters (FC can deal with uncertain information) + Can minimize the chattering of SMC	-Difficult to derive its transfer function
ILC/RC	+ Not based on system parameters, little information is needed + Learning capability, memory capability + Can be used as a torque ripple minimization approach to reduce ripples from external systems and internal motor	-Long delay, may not suitable for transient response

2.4 Torque Ripple Reduction by Machine Design

Four categories of machine design approaches are reported in [9], which can improve either the flux distribution or cogging effect of PMSM. A brief summary of the advantages and disadvantages of each machine design method can be seen as in Table 2.2 and Table 2.3. The explanation of each method can be found in [9], and will not be demonstrated here.

Regarding the effectiveness of cogging torque elimination, [67] confirms that it is impractical to completely eliminate the cogging torque due to manufacturing tolerance. However, cogging torque can be reduced to a satisfactory level required by the application. Moreover, since machine design cannot solve torque ripple induced from other parts of the drive system, the machine design methods for torque ripple reduction will not be considered for this work. This work will be focusing on investigating the control methods for torque ripple suppression.

Table 2.2: Advantages and disadvantages of torque ripple reduction methodologies by machine design-I

	Examples	Advantages	Disadvantages
Skewing	Skew the stator lamination stack or rotor magnetization (one slot pitch skewing)	+Simple +Effective +Widely used +Reduces the variation of reluctance seen by rotor magnets and hence cogging torque +Improves the stator windings distribution and substantially reduces higher order back-EMF harmonics	−Loss in average torque −More complex stator construction −Increases in leakage inductance and stray losses
Stator electrical winding type	Winding distribution (short-pitch windings, fractional-slot pitch windings)	+Well-known	−Reduces the high order with some loss of average torque
	Increasing the number of phases	+Increase the frequency of ripple	−Increase complexity of drive electronics and wiring
	Stepped skewing of magnet blocks	+Cancels all cogging torque harmonics except for multiples of the number of blocks	−Loss in average torque −Reduces back-emf
Stator magnetic design	Dummy slots and dummy teeth/ skewing the rotor by one dummy slot pitch	+Increases the frequency of cogging torque	−Does not eliminate, but makes harmonics less objectionable
	Others (decreasing the stator slot openings, using magnetic slot wedges .....)		

Table 2.3: Advantages and disadvantages of torque ripple reduction methodologies by machine design-II

	Examples	Advantages	Disadvantages
Rotor magnetic design	Air gap flux distribution (arc-shaped magnets with parallel magnetization, 2-pole)	+Produces a sinusoidal air gap flux distribution +Most Common +Most easily obtained in isotropic magnets	−Loss in average torque −Reduces back-emf
	Air gap flux distribution (thin magnetic retaining can on the rotor)	+More sinusoidal flux distribution than the one above	−Increases the magnet leakage at the pole tips −Decreases the efficiency − loss in torque
	Air gap flux distribution (tapering the magnet poles and/or air gap using bread loaf style magnets)	+More sinusoidal flux distribution than the two above	−Loss in average torque
	Air gap flux distribution (constructing the magnet poles out of blocks with varying magnetization directions)	+More sinusoidal flux distribution than the three above	−Loss in average torque
	Magnet pole arc width and positioning (choosing magnet arc to be 150°)	+Minimizes 6^{th} harmonic ripple torque	−Loss in average torque
	Magneti pole arc width and positioning (choosing magnet arc as (integer+0.14)slot pitch)	+Cancels the 1^{st} harmonic ripple torque	−Increases the magnet leakage at the pole tips
	Magnet pole arc width and positioning (choosing one set of magnets by 1/4 slot pitch)	+Cancels the 2^{nd} harmonic ripple torque	−Loss in average torque
	Stepped skewing of magnet blocks	+Cancels all cogging torque harmonics except for multiples of the number of blocks	−Loss in average torque −Reduces back-emf

2.5 Torque Ripple Reduction by Control

Alternatively, torque ripple can be compensated by means of control. A variety of control schemes have been presented in literature. These schemes are all about compensation techniques, they differ from each other in the sense of how they compute the compensation quantities, and where the compensation quantities are added in the system.

How the compensation quantity can be computed depends on what kind of feedback information is available. Commercial servo drive systems normally include three-phase current sensors for measuring in real-time the three-phase stator currents, an encoder or resolver for detecting the rotor position, and a voltage sensor only for measuring the d.c. bus voltage of the inverter. Some other additional sensors used in literature can be, for example, a torque sensor for achieving a direct measurement of torque, three-phase voltage sensors for measuring in real-time the three-phase stator voltages and so on. The key for reducing torque ripple by control is to extract the ripple torque information from these measurements. It would be, of course, desired to achieve maximum ripple reduction with minimum amount of information. Before such a desired compensation scheme can be developed, some existing schemes are compared in the following:

1) In the case that **only the torque ripple caused by non-sinusoidal flux distribution needs to be compensated**, the knowledge of stator back-emf would be enough for extracting torque ripple information. A compensation current can be generated according to the back-emf and other machine parameters (mainly inductances), and consequently can produce the reverse torque ripple to cancel the original ripple. The key point of this type of torque ripple reduction method is the accuracy of compensating current generation. Obviously, the most straight forward way is to take some measurements on the machine in advance as in the example below:

   - The back-emf and inductances of the motor can be pre-measured **offline**. For example, [65, 68] presents a procedure for acquiring a lookup table for flux linkage as a function of rotor position. The control diagram in [65] is shown in

Fig.2.7:

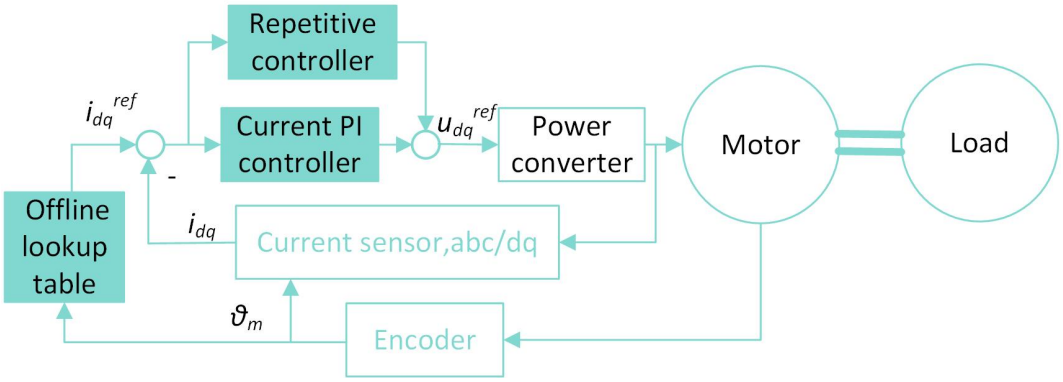


Figure 2.7: Block diagram of a pre-computed lookup table method for reduction of torque ripple caused by non-sinusoidal flux distribution

As shown in Fig.2.7, the current reference is generated by a pre-determined lookup table. The repetitive controller is used here to help the tracking of such current reference since the bandwidth of the conventional PI current control loop may not be high enough. **A current loop with higher bandwidth is required.** This method will be compared in details later in 5.1. However, the drawback of this offline approach is that the distribution of flux linkage may also vary due to saturation. **In this approach the effects of demagnetization or magnetic saturation are not considered.**

2) **More adaptive** torque ripple reduction can be achieved, if the real-time **online** estimation of back-emf or flux linkage is available. This requires the controller to have learning or observing capability. In such way, the torque ripple reduction can be more adaptive not only to machine parameter uncertainties, but also to inverter induced distortions.

- Self learning procedures for offline determination the stator resistance, stator inductances, and **online commissioning of flux harmonics** are proposed in [69]. The block diagram of the proposed control method in [69] is shown in Fig.2.8. It is worth pointing out that **a DB current control loop is used for achieving high bandwidth current control.** This method is known as "**Holtz' compensation** " in this work, and will be discussed in details later

in 5.1. However, the proposed algorithm assumes no voltage distortion in the inverter.

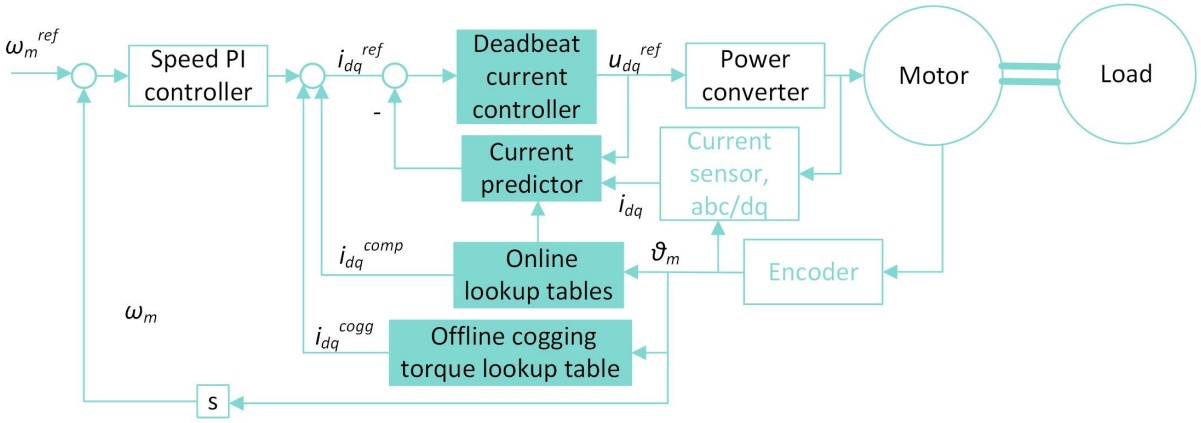


Figure 2.8: Block diagram of the Holtz method for torque ripple reduction

- Ref [70] claims to achieve good performance of torque ripple reduction adaptive to magnet saturation, inductance variation, as well as the **voltage distortion of inverter**. In the paper, a function for estimating uncertainties is presented as shown in Fig.2.9.

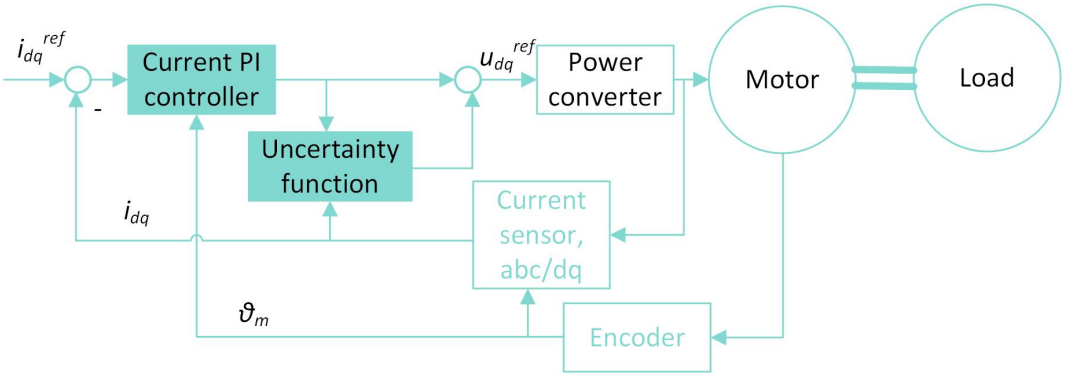


Figure 2.9: Block diagram of an uncertainty function method for torque ripple reduction

- Since the torque ripple produced by machine parameter uncertainties and inverter distortions is periodic with respect to rotor position, a position based memory space is applied in [71] for **online learning of the voltage distortion** as shown in Fig.2.10.

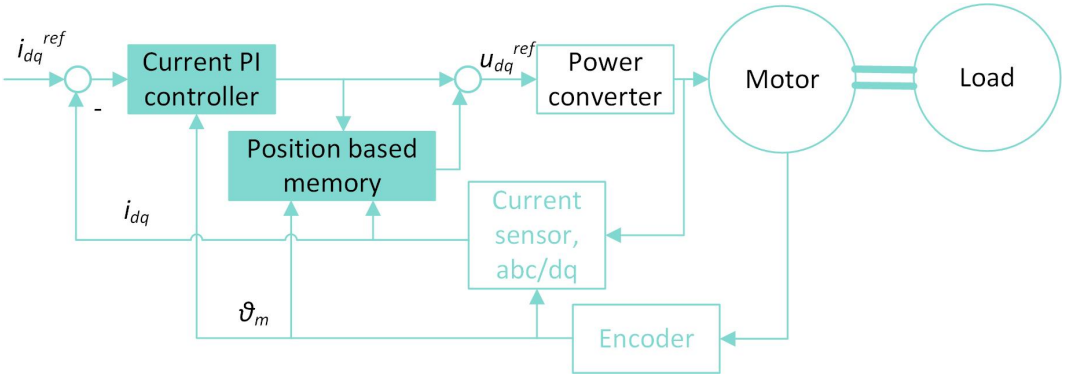


Figure 2.10: Block diagram of a memory position-based method for torque ripple reduction

3) Still, the cogging torque is not considered in the control schemes above. In case that **the cogging torque needs to be compensated**, its knowledge is required.

   - The amplitude of cogging torque is not affected much by the saturation, it mainly varies with rotor position. A procedure of accurately measuring cogging torque has been presented in [69, 72]. Again, a lookup table is built in [69] according to pre-measurements, for representing the cogging torque as a function of rotor position.

4) The compensation approaches above, all aim to reduce some specific sources of ripple. However, ripple caused by the mechanical imperfections (misalignment), or load side ripple is difficult to predict. In fact, torque ripple reduction can be achieved without necessarily knowing its sources. Once the knowledge of the machine output torque is available, the a.c. part in the torque can be extracted for generating proper compensation actions. **The overall torque ripple from all sources can be compensated as a whole.** In this case, many existing compensation schemes use ILC or RC which requires only the knowledge of the frequency or the order of the torque ripple. In other words, **torque ripple can be reduced without necessarily knowing the machine electrical parameters** (resistance, inductance, magnetic flux). Some examples of this approach is given as below:

   - One direct way of knowing torque ripple is to use a **torque sensor** [62] as in Fig.2.11. But this is not practical in commercial markets.

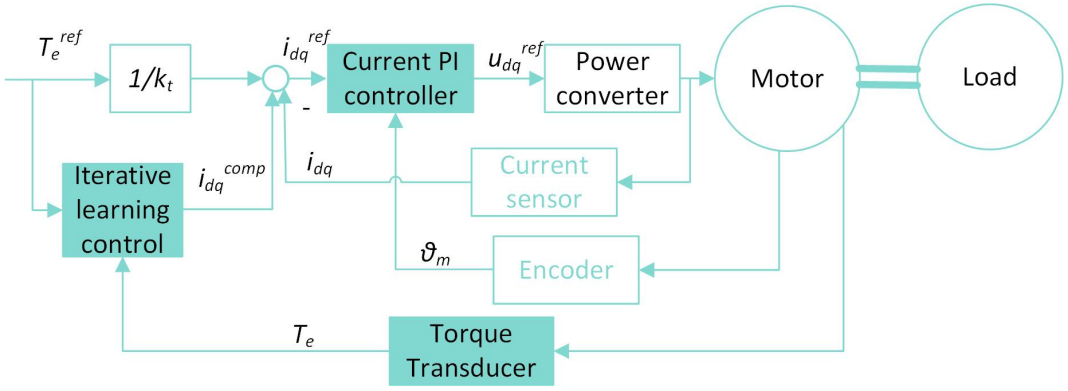


Figure 2.11: Block diagram of an ILC-based method for torque ripple reduction using the measured torque as feedback

- Alternatively, since torque ripple will be transformed into speed ripple, and the speed ripple is the key feature which matters in many applications, the knowledge of speed ripple would also be sufficient for observing the part of torque ripple which is large enough and low frequency enough (below the cut-off frequency of the mechanical plant) to cause obvious speed ripple. In this case, the use of a **torque sensor can be avoided**, and a high revolution encoder is required [63, 73]. The block diagram of the proposed control scheme in [63] is shown in Fig.2.12.

  Importantly, since torque ripple is mainly "position-varying" (i.e. varies with rotor position) as demonstrated before in 2.1, [73] presents an **angle-based ILC for achieving torque ripple reduction at variable frequency**. This approach can be still described by the control diagram in Fig.2.12; however, in this case, the iterative learning controller learns the ripple as a function of rotor position. Unfortunately, the proposed angle-based ILC in [73] has been tested only at low speed (50 rpm to 70 rpm). Whereas, in servo drive applications, the frequency is likely to be varying at a much wider range. **This paper proposes new methods for achieving torque ripple reduction under variable frequency by means of RC control in the full speed operating range. However, further developments are required.**

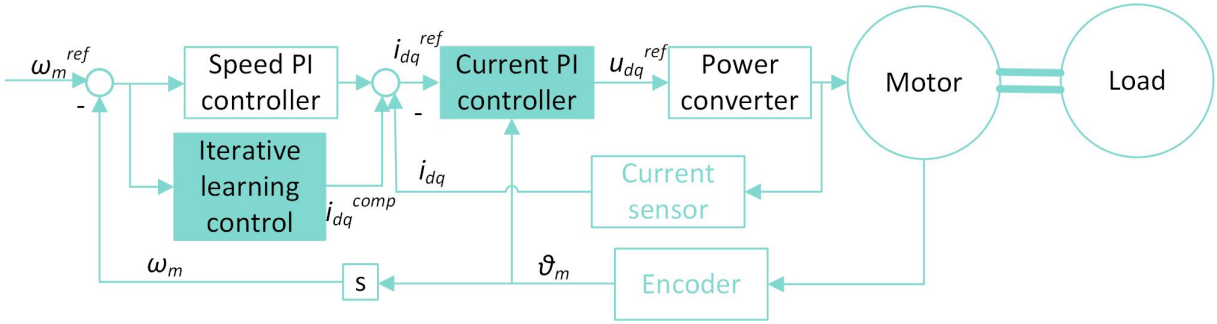


Figure 2.12: Block diagram of an ILC-based method for torque ripple reduction using position as feedback

- It is possible that the ripple torque components are mainly at **a few specific frequencies**. When the order of these frequencies is known, [74] presents a self learning compensation scheme which also only requires the knowledge of the order of the main ripple components as shown in Fig.2.13. Again, an encoder is required for detecting the position, and thus the speed ripple. This method will be named as "**Choi's compensation scheme**", and will be further discussed later in 5.1.

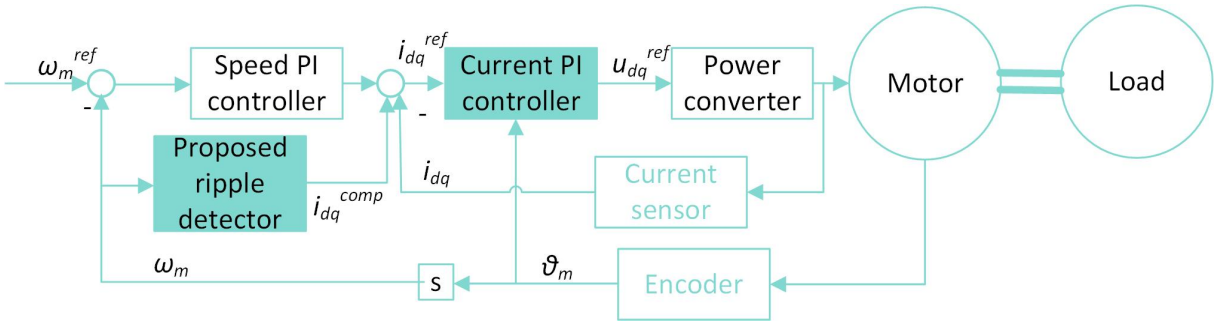


Figure 2.13: Block diagram of Choi's method for torque ripple reduction

2.6 Discussions and Conclusions

This chapter focuses on reviewing the state of the art technologies for general closed loop control and torque ripple reduction. From the literature review, the research tasks for this work can be suggested. The most important findings and suggestions can be summarized by the diagram Fig.2.14 below:

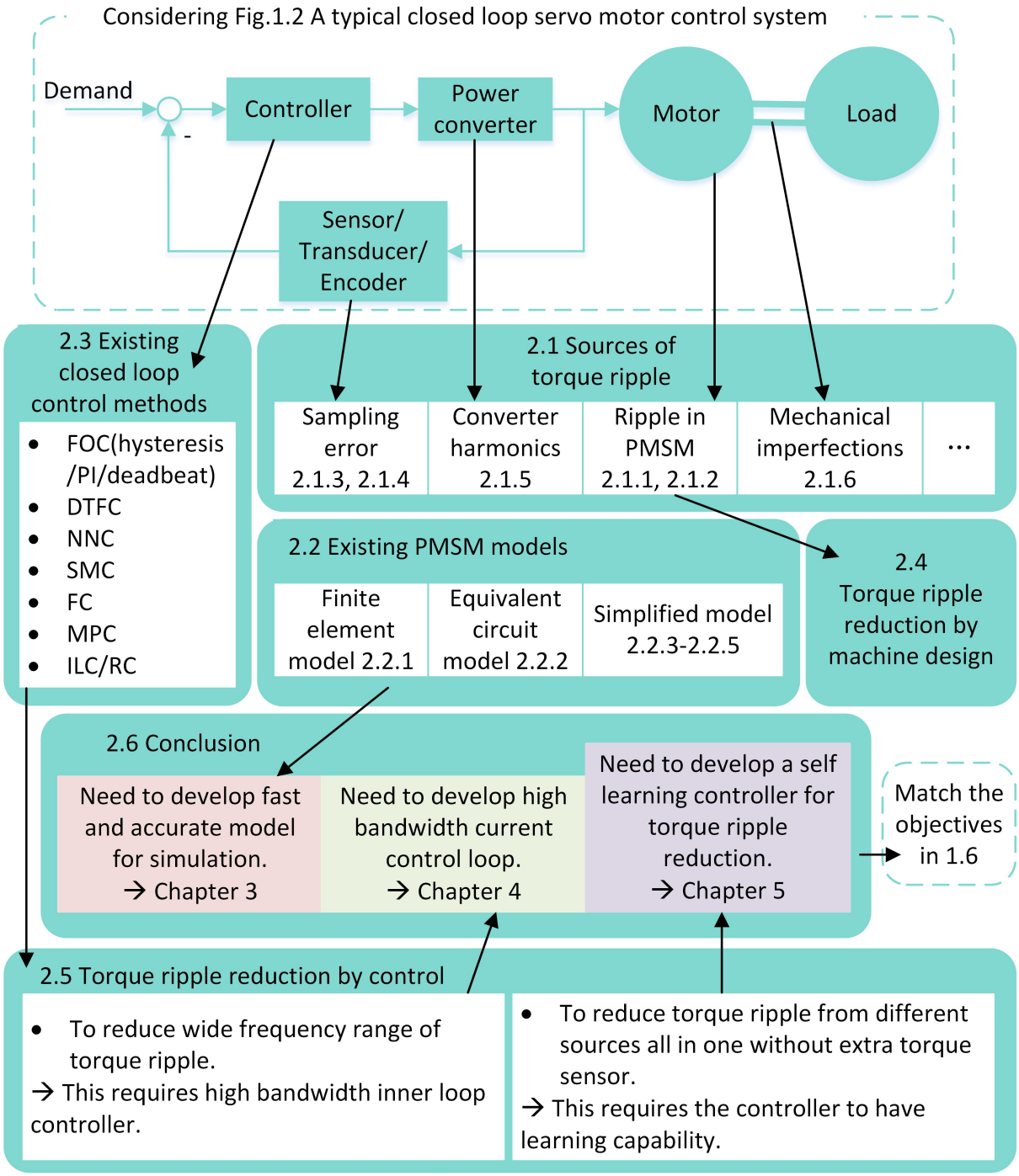


Figure 2.14: Key conclusions of Chapter 2

- From Section 2.1, we know that torque ripple can be generated from everywhere in the drive system as shown in Fig.1.2. One common nature of the torque ripple from all theses sources is that the ripple is periodic with the rotor position.

- From Section 2.2, we know that the existing simulation models for PMSM is either too slow, or too simplified (cannot shown the torque ripple due to cogging effect and flux harmonics in the machine). Therefore, a fast and accurate simulation model for PMSM will be developed in Chapter 3.

- From Sections 2.3 and 2.5, we know that torque ripple reduction can be achieved by generating proper compensation currents, which potentially contains high frequency components, therefore high bandwidth current controller (such as deadbeat (DB) current controller) is required. Consequently, an enhanced DB current controller will be developed in Chapter 4.

- From Sections 2.3 and 2.5, we also know that torque ripple can be learned through speed ripple, without a torque sensor. Moreover, the ripple can be learned using self-learning control techniques such as repetitive control (RC), without necessarily knowing the machine parameters. For achieving torque ripple reduction at variable frequency, the angle-based RC [73] can be a promising solution. The desired torque ripple reduction method for this work will be developed in Chapter 5.

Chapter 3

Modelling of PMSM

As mentioned in Chapter 2, a fast and accurate PMSM model needs to be developed for simulation. As described in 2.2, the finite element (FE) model can meet the required accuracy for this work. Although full FE analysis is time consuming, it is possible to save a significant amount of calculation time if the results for the same conditions have been calculated and stored in advance. For this reason, this chapter is divided into three parts:

First, a FE model has been built using Infolytica MagNet to analyse the cogging effect, non-sinusoidal flux distribution, and inductances variation of the PMSM. Seven lookup tables have been derived for these imperfections.

Second, an ideal PMSM model in alpha-beta coordinates as demonstrated in 2.2.4 has been built using Matlab/Simulink.

Third, the lookup tables have been embedded in the ideal Simulink model to include non-ideal features.

In such a way, the combined model is able to achieve fast computation and high accuracy at the same time. All the models mentioned above are for the surface mounted PMSM with parameters as shown in Table 3.1.

Table 3.1: Parameters of the PMSM for modelling according to manufacturer's datasheet

Parameter	Value [Unit]	Parameter	Value [Unit]
Rated power	2.54 [kW]	Phase to phase inductance	0.0114 [H]
Rated speed	3000 [rpm]	Phase to phase resistance	2.02 [Ω]
Rated torque	9.4 [Nm]	Voltage constant k_e	98.0 (V_{rms}/krpm)
Rated line voltage (RMS)	400 [V]	Torque constant k_t	1.6 (Nm/A_{rms})
Rated current (RMS)	5.9 [A]	Inertia J	0.00078 [kg/m^2]
Number of pole pairs	3	Number of slots and teeth	18

3.1 Finite Element Model

The finite element (FE) model has been built using Infolytica MagNet 7 as shown in Fig.2.1.

The FE analysis of this model is performed by setting the dq axis currents i_{dq} of the motor to be constant, and driving the motor under constant speed (1 rpm) for a whole electrical cycle (i.e. a third of the mechanical cycle). During such period, the three-phase stator flux linkages F_{abc}, voltages v_{abc}, and electrical output torque T_e are solved for 1000 different rotor positions. The machine structure is assumed symmetric. Therefore, the voltages, flux linkages and torque profiles in any of the three electrical cycles over a mechanical revolution are identical.

The test has been repeated for different i_{dq} settings as illustrated in Fig.3.1. Bearing in mind the i_d is usually controlled to be zero for a surface mounted PMSM and that field weakening is not considered in this work, the settings represent the following two conditions:

1) Pink zone in Fig.3.1 indicates the typical operating condition of surface mounted PMSM when $i_d = 0$ and $i_q = 0 \leftrightarrow 1$ p.u.. It is worth pointing out that cogging torque can be represented by the torque when $i_d = i_q = 0$.

2) Green zone in Fig.3.1 indicates conditions when $i_q = 0.5$ p.u. and $i_d = -1 \leftrightarrow 0.2$ p.u.. As mentioned, field weakening is not considered during the control design phase and i_d would be kept zero. However, it would be interesting to verify from the FE analysis how i_d would affect the machine voltages, fluxes and torque. Therefore,

extra points are collected for conditions when i_d varies from -1 to 0.2 p.u..

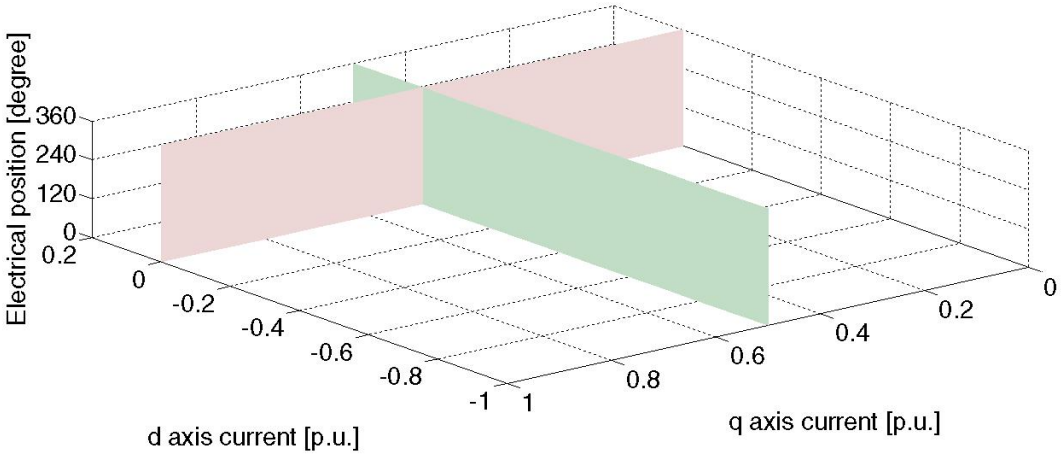


Figure 3.1: Setting of dq axis currents for finite element analysis

In the test, for each of the 1000 different rotor positions, the constant dq axis currents i_{dq} are first converted into the sinusoidal three-phase currents, and the current in the coil of each phase is set accordingly. During the FE computation, the three-phase voltages v_{abc}, flux linkages F_{abc} are solved. After the FE computation, the results are converted into dq voltages v_{dq} and flux linkages F_{dq}, for further calculating the electrical parameters as demonstrated below.

On one hand, electrical parameters such as the dq axis inductances L_{dq}, magnetic flux F_m are position and current dependent in the FE model. On the other hand, the normal dq or $\alpha\beta$ frame machine models as introduced in 2.2.4 simplifies the computation by assuming those parameters to be constant. One solution for improving the accuracy of these normal dq or $\alpha\beta$ frame machine models, while maintaining their fast computation, is to use lookup tables for those electrical parameters. For this purpose, seven lookup tables are derived based on the modified dynamic equations (3.1-3.3) for better reflecting the non-linearities in the machine.

$$\begin{aligned} v_d &= R_s i_d + \frac{dF_d}{dt} - \omega_e F_q \\ v_q &= R_s i_q + \frac{dF_q}{dt} + \omega_e F_d \end{aligned} \tag{3.1}$$

$$\begin{aligned} F_d &= L_d(i_d, i_q)i_d + F_{md}(i_d, i_q) + F_d^{ac}(\theta_e, i_d, i_q) \\ F_q &= L_q(i_d, i_q)i_q + F_{mq}(i_d, i_q) + F_q^{ac}(\theta_e, i_d, i_q) \end{aligned} \tag{3.2}$$

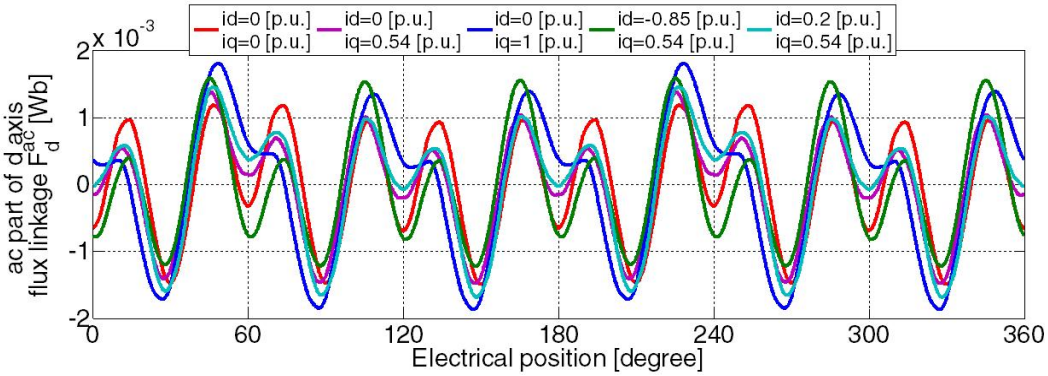
$$T_e = 1.5P_p(F_d i_q - F_q i_d) + T_e^{err}(\theta_e, i_d, i_q) \tag{3.3}$$

In these equations, $L_{dq}(i_d, i_q)$ are the two lookup tables for the dq axis inductances. $F_{mdq}(i_d, i_q)$ are the two lookup tables for the dq axis components of the magnetic flux. $F_{dq}^{ac}(\theta_e, i_d, i_q)$ are the two lookup tables for the a.c. part of the dq axis flux linkages, which are position dependant. $T_e^{err}(\theta_e, i_d, i_q)$ represents a missing component of torque which cannot be represented by the torque equation (2.16) mentioned in Chapter 2. When $i_d = i_q = 0$, T_e^{err} equals to the cogging torque.

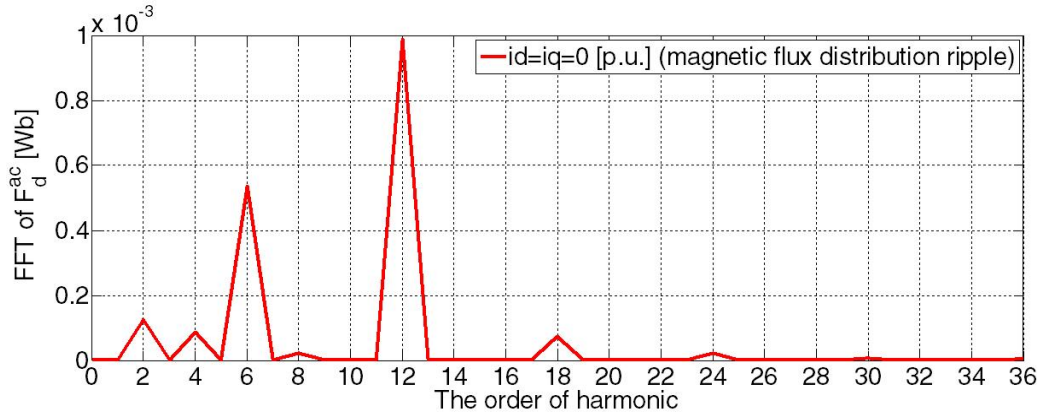
Since v_{dq} in (3.1), F_{dq} in (3.2), and T_e in (3.3) have been collected from the FE analysis, all the seven lookup tables can be derived. The following three subsections will demonstrate in details how the seven lookup tables are computed.

3.1.1 Lookup Tables $F_{dq}^{ac}(\theta_e, i_d, i_q)$ for the a.c. Part of dq Axis Flux Linkages

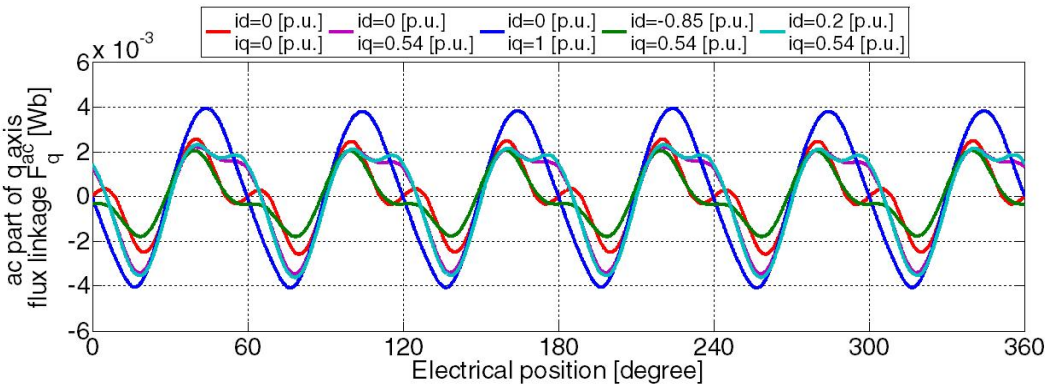
The recorded dq axis flux linkages F_{dq} can actually be seen as two parts: a.c. part and d.c. part. The a.c. part is the ripple part whose frequency content varies according to the rotor speed and so it is position dependent, whereas the amplitude of the harmonic content varies according to currents i_{dq}. The d.c. part value varies with currents i_{dq}. An example of the a.c. part of dq axis flux linkages $F_{dq}^{ac}(\theta_e, i_d, i_q)$ extracted are shown in Fig.3.2(a) and Fig.3.2(c):



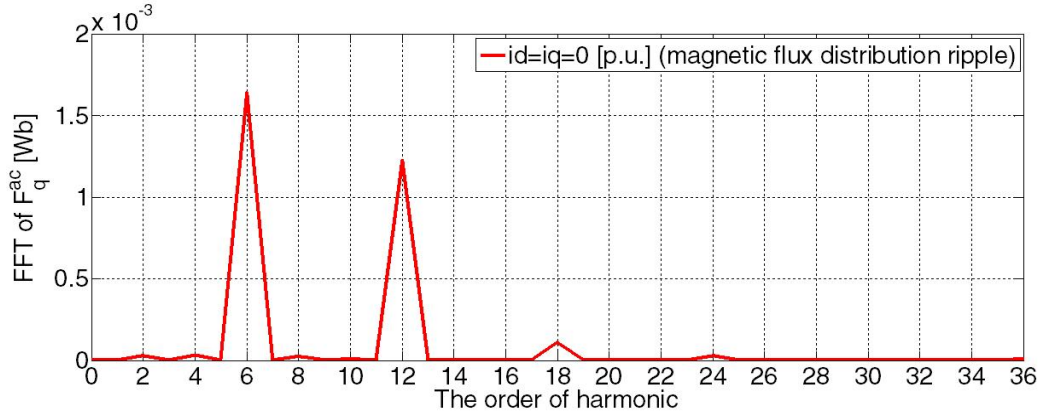
(a) $F_d^{ac}(\theta_e, i_d, i_q)$



(b) Ripple components in magnectic flux distribution



(c) $F_q^{ac}(\theta_e, i_d, i_q)$



(d) Ripple components of L_q

Figure 3.2: Example of the $F_{dq}^{ac}(\theta_e, i_d, i_q)$ lookup tables

Fig.3.2 shows that both the d axis and q axis flux linkages contain rotor position dependent ripple. Its shape and value are determined by the magnetic flux distribution. However, such ripple may not make significant difference on the dynamic performance of the machine, since the ripple is between -0.004 wb and +0.004 Wb, which is within ±1.6% of the rated magnetic flux ($= \frac{k_t}{1.5\times\sqrt{2}P_p} = \frac{1.6}{1.5\times\sqrt{2}\times 3} = 0.251$ Wb) calculated from Table 3.1. However, it is worth noting that this flux variation value depends on the modelling accuracy (the effect of mesh size, step size etc.). Therefore, for the parameter sensitivity tests of the proposed control in Chapter 4 and Chapter 6, the variation of magnetic flux from 50% to 150% of its rated value is considered.

The Fast Fourier Transformation (FFT) results for F_d^{ac} when $i_d = i_q = 0$ [p.u.] is shown in Fig.3.2(b). Under such condition and from (3.2), it follows that the a.c. part of F_d only depends on the distribution of the machine magnetic flux F_m (no other magnetic field can be generated by the zero currents). As can be seen, the magnetic flux is not distributed sinusoidally around the rotor, and it contains $2^{nd}, 4^{th}, 6^{th}, 8^{th}, 12^{th}, 18^{th}, 24^{th}$ harmonics, which would yield torque error. The FFT results of F_q^{ac} when $i_d = i_q = 0$ [p.u.] is shown in Fig.3.2(d). As can be seen, non-sinusoidal distribution of magnetic flux is also responsible for ripple in q axis flux linkage, its frequency components match the ripple in d axis flux linkage. Again, the ripple components shown in Fig.3.2(b) and Fig.3.2(d) are all small (10^{-3} Wb), whereas, the proposed control methods are verified considering the variation of magnetic flux far greater than 10^{-3} Wb.

3.1.2 Lookup Tables $F_{mdq}(i_d, i_q)$ for the dq Axis Magnetic Flux

The magnetic flux can be decoupled into d axis magnetic flux F_{md} and q axis magnetic flux F_{mq}. According to (3.2), F_{md} equals to F_d when $i_d = 0$, and F_{mq} equals to F_q when $i_q = 0$. Since the a.c. parts (i.e. the rotor position dependent parts) of F_{dq} have already been extracted in the previous subsection, the F_{mdq} lookup tables only vary with stator currents, not the rotor position. It is worth pointing out that $F_{mq}(i_d, i_q)$ **is neglected** for the rest of this work, since it is found to be very small (10^{-6} Wb). Although its value may vary according to the mesh size, step size of the FE model, the fact that q axis

magnetic flux is negligible is still true because magnetic flux mainly aligns with d axis. Some calculation results of $F_{md}(i_d, i_q)$ are shown in Fig.3.3.

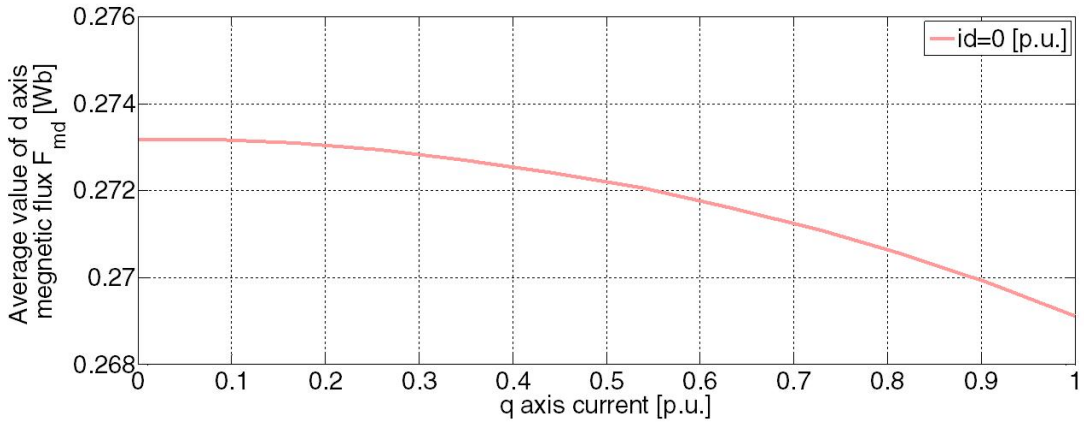
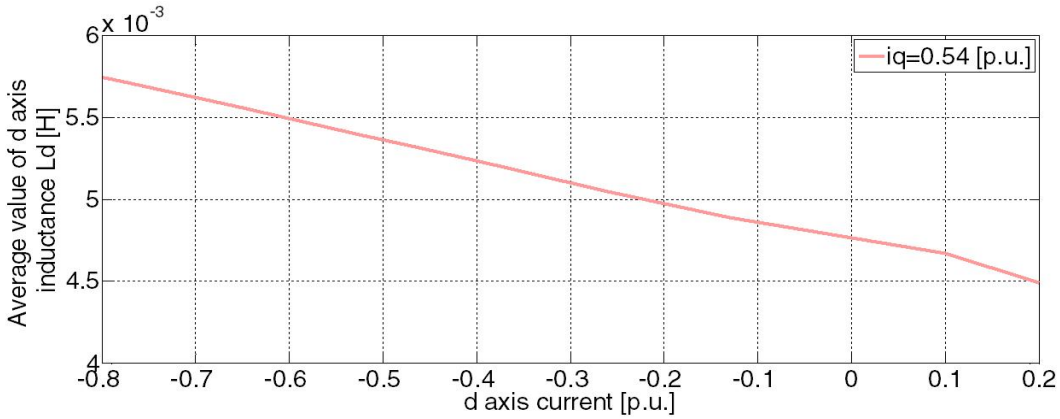


Figure 3.3: Example of the $F_{md}(i_d, i_q)$ lookup table

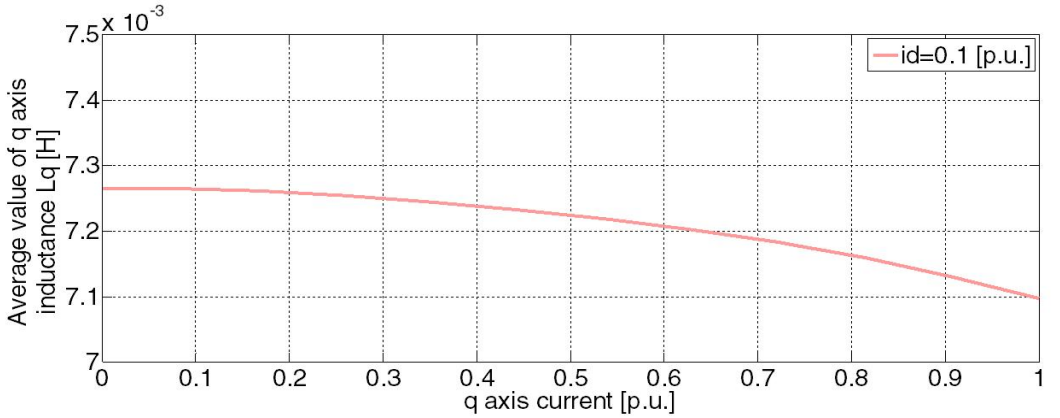
As can be seen, the average value of d axis magnetic flux decreased by 1.5% as i_q increases from zero to the rated value. It has been observed from these results that the magnetic flux would barely vary with current. The effect of magnetic flux variation on the control performance will be assessed in 4.2.

3.1.3 Lookup Tables $L_{dq}(i_d, i_q)$ for the dq Axis Inductances

After deriving $F_{dq}^{ac}(\theta_e, i_d, i_q)$ and $F_{mdq}(i_d, i_q)$ lookup tables, the only unknown terms in (3.2) are the dq axis inductances $L_{dq}(i_d, i_q)$. L_d can be subtracted from (3.2) when $i_d \neq 0$, L_q can be subtracted when $i_q \neq 0$. The L_{dq} values when $i_d = 0$ or $i_q = 0$ can be estimated by interpolation. Again, since the rotor position dependent part has already been extracted by $F_{dq}^{ac}(\theta_e, i_d, i_q)$, these inductances are the average values over one electrical revolution. Their values would only be affected by the dq axis currents. Some calculation results of $L_{dq}(i_d, i_q)$ are shown in Fig.3.4.



(a) $L_d(i_d, i_q)$



(b) $L_q(i_d, i_q)$

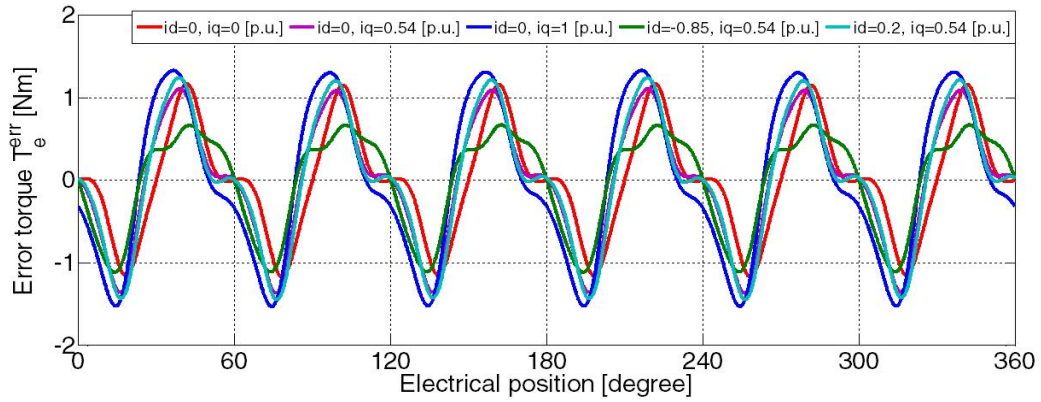
Figure 3.4: Example of the $L_{dq}(i_d, i_q)$ lookup tables

As can be seen, the average value of L_d drops 22.5% as i_d increases from -80% to $+20\%$ of the rated value, whereas, the average value of L_q drops 2.3% as i_q increases from zero to the rated value. The effect of inductances variation on the control performance will be assessed in 4.2.

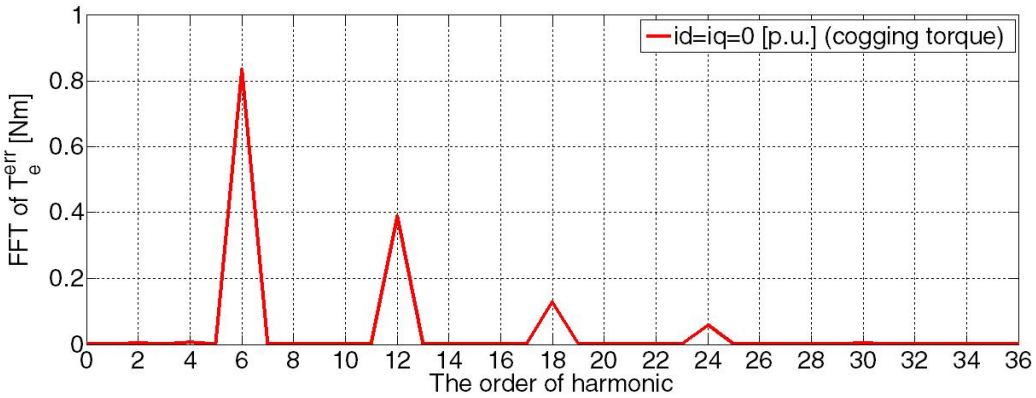
3.1.4 Lookup Table $T_e^{err}(\theta_e, i_d, i_q)$ for Error Torque

In (3.3), the dq axis flux linkages F_{dq}, the electrical output torque T_e are acquired from the FE analysis; the dq axis currents i_{dq} are pre-determined constant values; the number of pole pairs P_p is known. Therefore, the error torque $T_e^{err}(\theta_e, i_d, i_q)$ between the real electrical output torque T_e and the one calculated from flux linkage and current can be easily extracted (as refer to the torque equation in (3.3)). Some examples of the resultant error torque waveforms are shown in Fig.3.5. It is worth noting that in case $i_d = i_q = 0$

(red line) , T_e^{err} represents the cogging torque.



(a) $T_e^{err}(\theta_e, i_d, i_q)$



(b) Ripple components in cogging torque

Figure 3.5: Example of the $T_e^{err}(\theta_e, i_d, i_q)$ lookup table

The ripple torque in Fig.3.5(a) varies from around -1.53 Nm to 1.33 Nm, which is -16.3% to +14.1% of the rated torque (9.4 Nm). Fig.3.5(b) shows the FFT results of the cogging torque. It can be seen that the cogging torque contains the 6^{th} harmonics and its integer multiples, which is determined by the number of slots (equals to 18 as can be seen from Fig.2.1 and Table 3.1) divided by the number of pole pairs (equals to 3 as in Fig.2.1 and Table 3.1).

3.2 The Combined Model

A model of the same surface mounted PMSM has been built under $\alpha\beta$ frame using Matlab/Simulink. The model has been built based on (3.4)-(3.6). Distinct from the conven-

tional $\alpha\beta$ frame dynamic equations for PMSM given in 2.2.4, these new equations include the pre-mentioned six lookup tables (lookup table $F_{mq}(i_d, i_q)$ is neglected as explained in 3.1.2, therefore six lookup tables remain).

$$\begin{aligned} v_\alpha &= R_s i_\alpha + \frac{dF_\alpha}{dt} \\ v_\beta &= R_s i_\beta + \frac{dF_\beta}{dt} \end{aligned} \tag{3.4}$$

$$\begin{aligned} F_\alpha &= L_\alpha i_\alpha + L_{\alpha\beta} i_\beta + [F_{md}(i_d, i_q) + F_d^{ac}(\theta_e, i_d, i_q)]cos\theta_e - F_q^{ac}(\theta_e, i_d, i_q)sin\theta_e \\ F_\beta &= L_{\alpha\beta} i_\alpha + L_\beta i_\beta + [F_{md}(i_d, i_q) + F_d^{ac}(\theta_e, i_d, i_q)]sin\theta_e + F_q^{ac}(\theta_e, i_d, i_q)cos\theta_e \\ &\text{where:} \\ L_\alpha &= L_0 + L_1 cos(2\theta_e), L_{\alpha\beta} = L_1 sin(2\theta_e), L_\beta = L_0 - L_1 cos(2\theta_e) \\ L_0 &= \frac{L_d(i_d, i_q) + L_q(i_d, i_q)}{2}, L_1 = \frac{L_d(i_d, i_q) - L_q(i_d, i_q)}{2} \end{aligned} \tag{3.5}$$

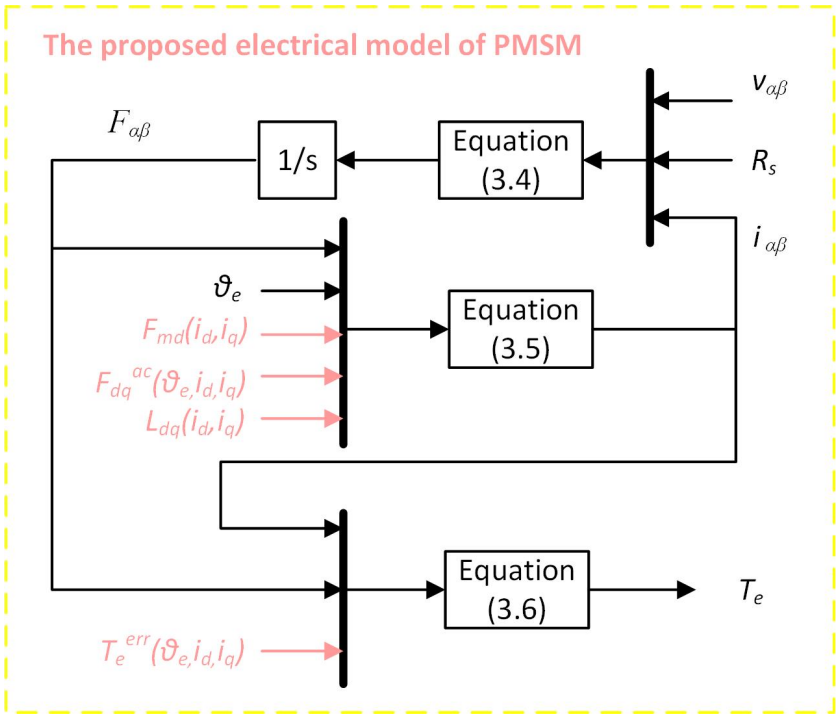
$$T_e = 1.5 P_p (F_\alpha i_\beta - F_\beta i_\alpha) + T_e^{err}(\theta_e, i_d, i_q) \tag{3.6}$$

A simulation test has been carried out in order to check whether this Simulink model agrees with the FE model in 3.1. As shown in Fig.3.6, the PMSM lookup table model is built according to the equations above. Two PI regulators are designed for controlling the dq axis currents i_{dq}. The speed of the motor is set to be 1 rpm. The dq axis reference currents i_d^{ref} and i_q^{ref} are set to be 0 A and 5.75 A (i.e. 0.7 p.u.) respectively. The test has been repeated for different reference current settings ($i_d^{ref} = i_q^{ref} = 0$ and $i_d^{ref} = -0.27 p.u., i_q^{ref} = 0.5 p.u.$).

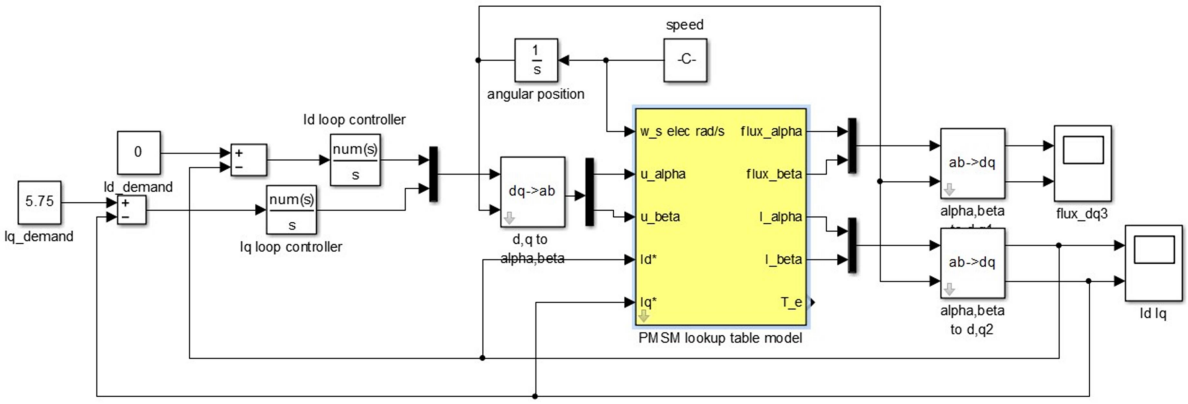
The resultant dq axis currents i_{dq}, dq axis voltages v_{dq}, dq axis flux F_{dq}, and output torque T_e waveforms are plotted from both the proposed Simulink model and the FE model in 3.1 are compared in Fig.3.7.

As shown in Fig.3.7, under the same dq axis currents and rotor speed, both models share the same dq axis flux F_{dq}, the same dq axis voltage v_{dq}, and the same output electrical

torque T_e. (Some waveforms seem invisible in the plots since the waveforms from the two model are almost identical.)

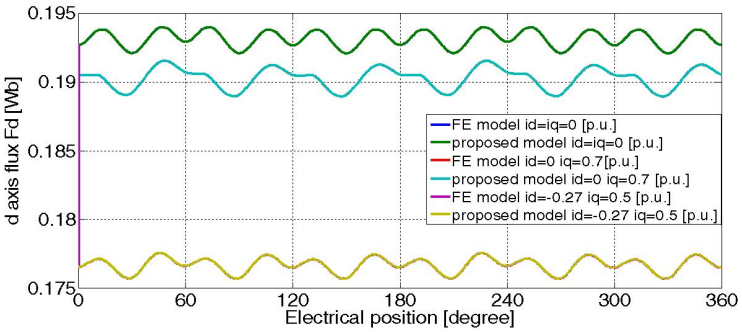


(a) The proposed electrical model of PMSM

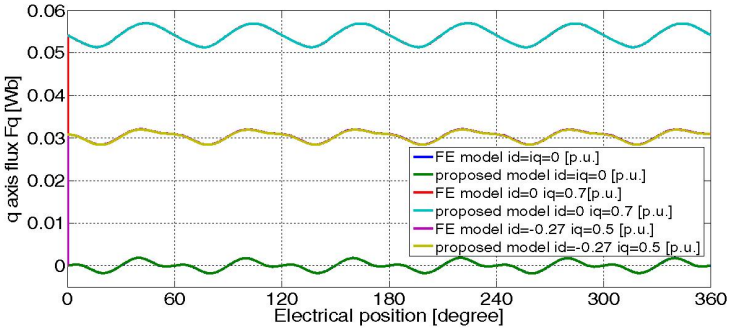


(b) The proposed model with current loops

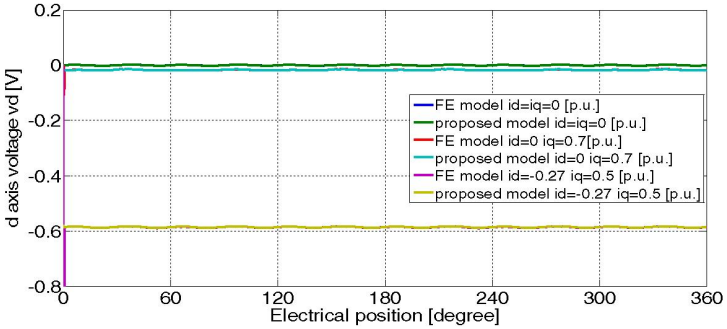
Figure 3.6: The proposed lookup table embedded $\alpha\beta$ frame PMSM model with PI current control loops



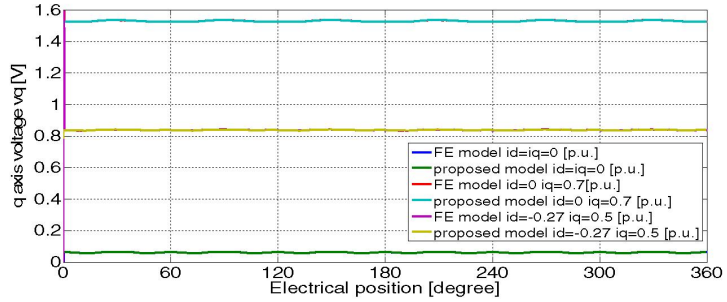
(a) d axis flux linkage of PMSM F_d



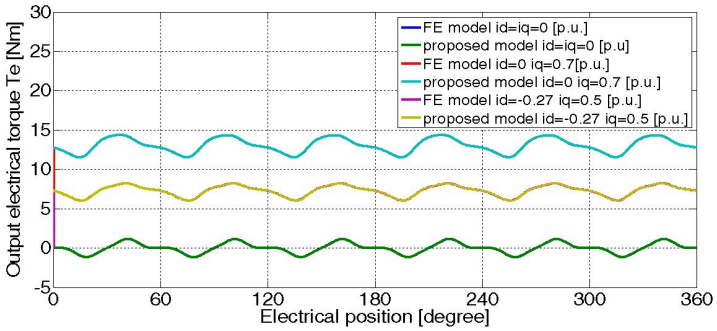
(b) q axix flux linkage of PMSM F_q



(c) d axix voltage of PMSM v_d



(d) q axix voltage of PMSM v_q



(e) Ouput electrical torque of PMSM T_e

Figure 3.7: The agreement between the proposed PMSM model and FE model

3.3 Conclusions

This Chapter has proposed a lookup table embedded $\alpha\beta$ frame PMSM model using Matlab/Simulink. Seven lookup tables are embedded for representing the a.c. part of dq axis flux linkages $F_{dq}^{ac}(\theta_e, i_d, i_q)$, the dq axis inductances $L_{dq}(i_d, i_q)$, the dq axis components of magnetic flux $F_{mdq}(i_d, i_q)$, and the error torque $T_e^{err}(\theta_e, i_d, i_q)$ which is not produced by the interaction between the magnetic flux and stator currents. Some of these lookup tables are rotor electrical position θ_e dependent, while some are only dq axis currents i_{dq} dependent.

For calculating these lookup tables, a FE model has been built first using the Infolytica MagNet 7, the details of the FE model and each lookup table can be seen in 3.1. After all the lookup tables are acquired, the proposed lookup table embedded model has been built as can be seen in 3.2. It has also been verified in 3.2 that the proposed model matches the FE model. Before applying this model for the rest of the work, some important **facts about the proposed model** need to be highlighted as follow:

1) **Operating region of the proposed model**:

   Fast computation of the proposed model is achieved by using pre-calculated lookup tables, therefore, the accuracy and operating conditions of the proposed model depend on the accuracy and derivation conditions of the lookup tables. In case of this work, the lookup tables are calculated according to the FE analysis results only under selected dq axis currents as shown in Fig.3.1. Interpolations are used for current conditions different from the selective ones. Such selective current conditions in Fig.3.1 include **the normal operating condition for a surface mounted PMSM without field weakening** ($i_d = 0$ A, i_q varies from zero to the rated value). It should be noted that the model will be used only for simulation purposes and not for control since the proposed torque ripple compensation technique does not require the knowledge of any machine parameter. Consequently the accuracy of the lookup tables is important only with regard to the their frequency content which must be related to the rotor position. Any inaccuracy in the amplitude of the

harmonic content will, of course, introduce a difference between real and simulated motors but the resulting different torque ripple would be compensated in any case by the proposed RC control as long as position dependent.

2) **Parameter variations in the operating region**:

Under the pre-mentioned normal surface mounted PMSM operating condition, and according to the lookup table plots in 3.1, the variation range of machine electrical parameters can be outlined:

From Fig.3.2, it can be known that variation of dq axis flux linkages with respect to rotor position is small. Since dq axis flux linkages are the function of dq axis inductances L_{dq} and the magnetic flux F_m, it is observed that L_{dq} and F_m vary little with respect to the rotor position.

Fig.3.3 shows that the d axis magnetic flux decreases only 1.5% as i_q increases from zero to the rated current (i_d=0 A). Under the same current condition, Fig.3.4 shows L_q only decreases by 2.3%.

For a better design of the current controller, it is necessary to verify the parameter variation range. As it can be seen later in Chapter 4, the magnetic flux and q axis inductance are the key parameters that may degrade the steady state and dynamic performance of the proposed current controller respectively. Hence, **it is important now to confirm from the FE model that the variations of these two parameters (against rotor position and against machine current) are all small enough for maintaining the high performance of current control (see Chapter 4).**

3) **Possible torque ripple components in the operating region**:

Equation (3.3) indicates that torque ripple may be produced by F_{dq}, i_{dq}, and/or T_e^{err}. The purpose of this Chapter is to model accurately the possible torque ripple produced by the machine itself using lookup tables. Two important facts can be confirmed from the lookup tables:

First, as can be seen from Fig.3.5, the ripple in T_e^{err} can be as high as 16% of the rated torque (9.4 Nm). This confirms the fact that cogging torque is the largest

ripple contributor. Second, Fig.3.2 and Fig.3.5 confirm the fact that the ripple is periodic with respect to rotor position.

This finding would give a good reason for the further development of the proposed angle-based repetitive control for torque ripple reduction in Chapter 5.

Chapter 4

Current Control Loop of PMSM

A well designed inner control loop can lay a good foundation for torque ripple reduction in PMSM drives. Based on the conventional field orientated control (FOC), this chapter presents and evaluates the design of the inner control loop (i.e. the current control loop). According to the characteristics of the current loop analysed in this chapter, the outer loops including torque ripple reduction function are further developed later in Chapter 5.

As discussed in 2.3, under the FOC scheme (Fig.2.4), the existing torque ripple reduction methods for PMSM drives are generally implemented by current profiling. Therefore, the performance of the current control loop can influence directly, the effectiveness of the torque ripple reduction.

Regarding the current control loop, since the reference current for torque ripple reduction is likely to contain components of different frequencies, the current loop is required to be able to follow the reference of all these frequencies. Therefore, a high bandwidth current control loop is a desired feature from the viewpoint of torque ripple reduction. Among the existing control methods for the current loop, the deadbeat current control (DBCC) is chosen for this work.

This chapter is outlined as follows. Equations of the deadbeat current control are arranged according to three functionalities in 4.1. The proposed DBCC is evaluated in 4.2. This chapter is concluded finally in 4.3 with simulations and experimental results. It

is to be noticed that the tests of the control loops in this chapter are implemented using the same surface mounted PMSM as for chapter 3 with the parameters shown in Table 3.1.

4.1 Deadbeat Current Control Loop

It is a widely held view that the deadbeat current control (DBCC) is a high bandwidth control method. Its high bandwidth characteristic is achieved by predicting the response of the model according to the pre-known parameters. As a consequence, the deadbeat control is always claimed to be sensitive to the parameter variations of the model. The deadbeat current control loop has been designed without considering the parameter variations in the following section, and the influence of detuned parameters on the characteristics of the deadbeat control loop is assessed later in 4.2.

A few assumptions are made in the design of deadbeat control. The reasons for using these assumptions are explained in the following paragraphs.

Assumption 1: In order to simplify the dynamic equations for the machine as shown in (3.1) and (3.2), the machine is assumed to be always operating in the non-saturation region; therefore the dq axis inductances L_{dq} are assumed to be constant. Meanwhile, the magnetic flux F_m is assumed to be ideally sinusoidally distributed; therefore F_m is a constant and perfectly aligned with the d axis (thus, F_m=F_{md}, F_{mq}=0). Hence, equations (3.1) and (3.2) can be simplified as in (4.1) (which is exactly (2.17)):

$$\begin{aligned} v_d &= R_s i_d + L_d \frac{di_d}{dt} - \omega_e L_q i_q \\ v_q &= R_s i_q + L_q \frac{di_q}{dt} + \omega_e L_d i_d + \omega_e F_m \end{aligned} \tag{4.1}$$

Before introducing assumptions 2 to 4, it is now necessary to introduce the principle of the proposed DBCC. For clarity, the time sequence of DBCC for only the k^{th} sampling period is drawn as in Fig.4.1.

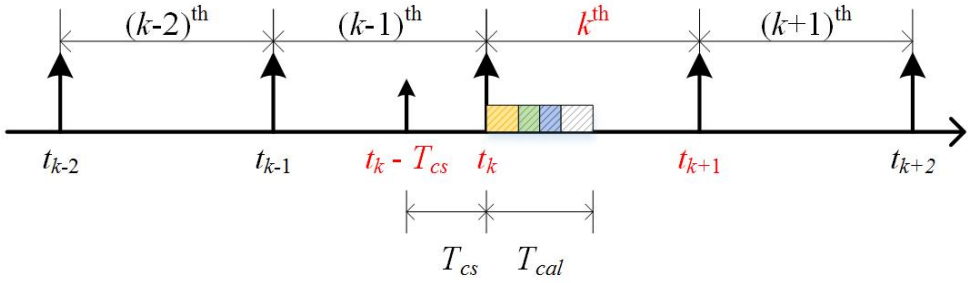


Figure 4.1: Time sequence of the proposed DBCC for the k^{th} sampling period

The principle of the DBCC in this work is to calculate the $(k+1)^{th}$ dq voltage references considering the sampling advance time T_{cs} due to current detection (depending on when exactly the currents are sampled) and the sampling period T_s of the digital control. To ensure the availability of samples at the beginning the k^{th} interval, T_{cs} should be chosen long enough to cover A/D conversion and transmission times. As a consequence of including T_{cs} in the time sequence, the proposed DBCC has two steps current predictions instead of traditionally one. Traditionally, the computation phase is executed a small period ($> T_{cs}$) after the beginning of each sampling period to ensure the availability of samples. Even though not compulsory, including T_{cs} can save more time for the computation phase. More advantages of having T_{cs} and two steps predictions will be demonstrated later in 4.1.1. Full calculations for the DBCC are executed in each sampling period T_s. Therefore the calculation time T_{cal} needs to be smaller than T_s.

Since the deadbeat control is discrete, (4.1) needs to be converted into a discrete form. The equation (4.1) can be discretized using the forward rectangular method. Considering in the digital control implementation, the controller is able to update the measurement inputs (such as measurements of the three phases currents i_{abc} and rotor position θ) and the control outputs (the dq axis reference voltages v_{dq}^{ref}) only once per sampling period T_s. Therefore, the assumptions 2 to 4 hold to simplify the discretization.

Assumption 2: The rotor electrical angular speed ω_e is assumed to be constant during each sampling period. In fact, $\omega_e(k)$ can also be assumed constant even in several sampling periods, since the dynamics of the mechanical system is normally much slower than the electrical system. Therefore, the ω_e sampled for the k^{th} sampling period can be represented as $\omega_e(k)$ in the digital controller.

Assumption 3: The dq axis reference voltages v_{dq}^{ref} can only be updated at the beginning of each sampling interval. Hence, the real dq axis voltages in the motor v_{dq} are assumed to be constant during each sampling period. Therefore, the v_{dq} applied to the motor for the k^{th} sampling period are represented as $v_{dq}(k)$. This simplification will be removed later when considering the rotor movement compensation.

Assumption 4: The dq axis currents i_{dq} are assumed to vary linearly during each sampling period T_s. Therefore, the current differential term in (4.1) can be represented as the difference between the instantaneous current at the beginning of the sampling period and the value at the end of the sampling period over T_s. The instantaneous dq axis currents at the beginning of the k^{th} sampling period are represented by $i_{dq}(t_k)$.

Following principle of the DBCC and assumptions 2-4 above, the discrete dynamic equations of PMSM can be written as :

$$\begin{aligned} v_d(k) &= R_s i_d(t_k) + L_d \frac{i_d(t_{k+1}) - i_d(t_k)}{T_s} - \omega_e(k) L_q i_q(t_k) \\ v_q(k) &= R_s i_q(t_k) + L_q \frac{i_q(t_{k+1}) - i_q(t_k)}{T_s} + \omega_e(k) L_d i_d(t_k) + \omega_e(k) F_m \end{aligned} \tag{4.2}$$

Furthermore, according to the time sequence shown in Fig.4.1, the principle of DBCC is to apply a reference voltage at t_{k+1}, which is right enough to bring the current to the reference value at t_{k+2}; and such calculations of the reference voltage is done in advance in the k^{th} sampling period. As a result, the assumption 5 below holds for DBCC.

Assumption 5: The real currents at t_{k+2}, $i_{dq}(t_{k+2})$, are supposed to match the reference currents $i_{dq}^{ref}(t_k)$ which are used for calculating the $(k+1)^{th}$ reference voltages $v_{dq}^{ref}(k+1)$ during the k^{th} sampling period (thus, $i_{dq}(t_{k+2}) = i_{dq}^{ref}(t_k)$). Theoretically, a minimum delay of $2T_s$ is required for the current to reach the reference. Of course, it may take longer for the current to track the reference when the voltage required is beyond the physical limitation of the inverter. An example of the performance of DBCC at physical limit is given in 4.2.2.1.

In fact, an enhanced deadbeat current control is developed for this work. In de-

tails, the two steps current prediction presented in 4.1.1.1 would improve the accuracy of the current predictions especially during transients; the rotor movement compensation presented in 4.1.3.2 would reduce the offset errors in the dq axis current responses, which is especially effective at high speed. The block diagram of the proposed deadbeat current controller is shown in Fig.4.2. Essentially, the required calculations are performed under **three functionalities**:

1) Current predictions (the yellow part in Fig.4.1 and Fig.4.2)

2) Reference voltage calculation (the green part in Fig.4.1 and Fig.4.2)

3) Rotor movement compensation (the blue part in Fig.4.1 and Fig.4.2)

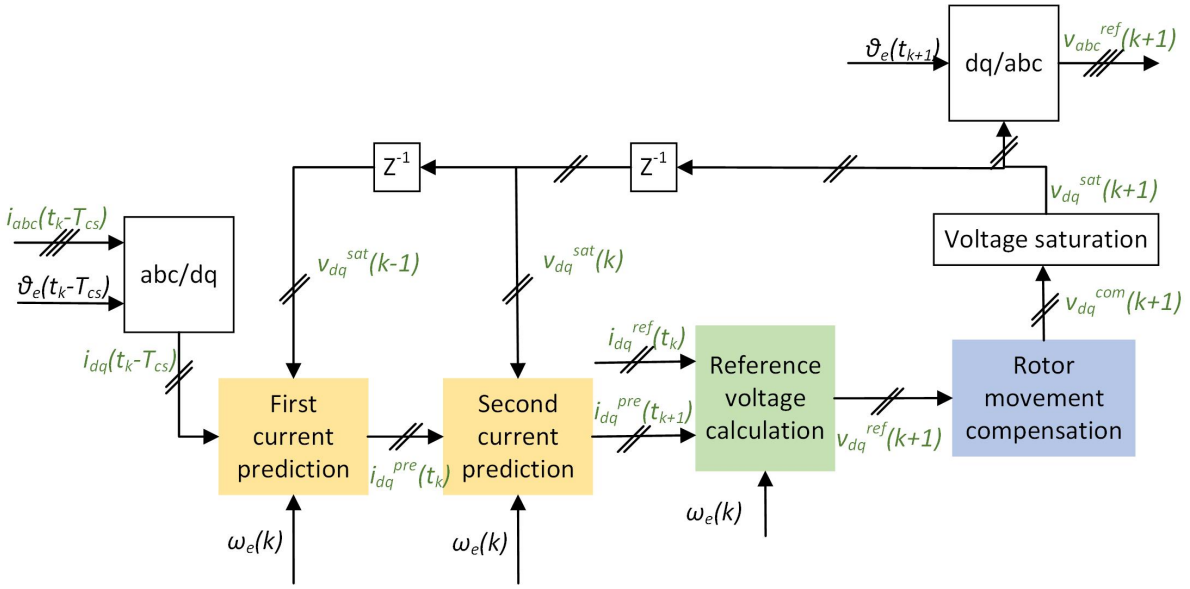


Figure 4.2: Block diagram of the proposed deadbeat current controller

4.1.1 Current Predictions

According to the principle of DBCC as mentioned before, the $(k+1)^{th}$ dq voltage references need to be calculated in the k^{th} interval and will be effectively applied at t_{k+1}, so the initial dq currents used for calculation should be those at t_{k+1} and not the ones measured at $t = t_k - T_{cs}$. Since the $(k+1)^{th}$ dq currents are in the future with respect to the k^{th} sampling period (during which calculations are being performed), they must be predicted using the

motor model. Considering that two different dq reference voltages (the $(k-1)^{th}$ and k^{th}) are applied in the time interval $[t_k - T_{cs},\, t_{k+1}]$, the predictions must be performed in two steps. However, the sampling advance time T_{cs} is normally neglected and therefore the current prediction in the traditional DBCC is normally one step. A comparison between DBCC with the proposed two steps current prediction and the traditional DBCC is given later in 4.2.2.1. Generally, **having two steps current prediction instead of one step can improve the accuracy of the predictions both at steady state and during transients**. Some reasons are as follow:

1) As a consequence of the rotor rotation, the current profile within each sampling period at steady state can be very irregular (as shown later in Fig.4.13) and the instantaneous current at the middle of the sampling interval can be very different from the value at the end of the sampling interval. Therefore, the sampling advance time T_{cs} can not be neglected especially at high speed. In such cases, it is also possible to sample the average value of the current by choosing proper T_{cs} in order to bring the average value of current to the reference current. Some examples of the wavy current profile are given in 4.2.2.2.

2) During transients, the accuracy of current predictions can directly affect whether the sudden change in the reference current can be tracked in the shortest possible period of time ($=2T_s$ according to assumption 5). Specially, when the **mean value** of the current during one sampling period instead of the instantaneous value is used as the feedback, an overshoot may occur during transients when using the traditional DBCC; and such overshoots can be reduced by using the two steps current prediction. An example is given in Fig.4.12.

4.1.1.1 Two Steps Current Prediction

An important fact that needs to be considered for the current predictions in DBCC is: The reference voltages $v_{dq}^{ref}(k-1)$ and $v_{dq}^{ref}(k)$ could be different from the voltages effectively applied to the motor when they are above the physical limit of the inverter. As shown

in Fig.4.2, the voltage saturation block calculates the saturated reference voltages v_{dq}^{sat} considering the saturation effect of the inverter. These saturated reference voltages can be used for the current predictions instead of the real dq axis voltage of the motor since voltage sensors are not usually present in a drive for cost reasons.

The first current prediction is to predict the dq currents at t_k using the $(k-1)^{th}$ dq voltages effectively applied (the saturated reference voltages) and the measured dq currents at $t_k - T_{cs}$. The second current prediction is to predict the dq currents at t_{k+1} using the k^{th} dq voltage applied and the previously predicted dq currents at t_k.

For the **first current prediction**, the saturated reference voltages $v_{dq}^{sat}(k-1)$ are used. Please note that $v_{dq}^{sat}(k-1)$ are available during the k^{th} sampling period since calculated during the $(k-2)^{th}$ sampling period. Furthermore, under assumption 2 and 3, (4.1) can be rewritten as:

$$\begin{aligned} v_d^{sat}(k-1) &= R_s i_d + L_d \frac{di_d}{dt} - \omega_e(k) L_q i_q \\ v_q^{sat}(k-1) &= R_s i_q + L_q \frac{di_q}{dt} + \omega_e(k) L_d i_d + \omega_e(k) F_m \end{aligned} \tag{4.3}$$

Both sides of (4.3) are integrated from $t_k - T_{cs}$ to t_k.

$$\begin{aligned} \int_{t_k - T_{cs}}^{t_k} v_d^{sat}(k-1)dt &= \int_{t_k - T_{cs}}^{t_k} \left\{ R_s i_d + L_d \frac{di_d}{dt} - \omega_e(k) L_q i_q \right\} dt \\ \int_{t_k - T_{cs}}^{t_k} v_q^{sat}(k-1)dt &= \int_{t_k - T_{cs}}^{t_k} \left\{ R_s i_q + L_q \frac{di_q}{dt} + \omega_e(k) L_d i_d + \omega_e(k) F_m \right\} dt \end{aligned} \tag{4.4}$$

(4.5) is then obtained by calculating the integration terms in (4.4) according to the assumption 4 which is necessary for avoiding a closed-form solution of the differential equations (4.3). Although the assumption will not be true once the motor starts rotating and will even cause steady state errors in current responses, this steady state error can be cancelled by the rotor movement compensation method proposed in 4.1.3.2. The assumption 4 may also be wrong if T_s is not sufficiently small (10 times) compared with the electrical time constant of the motor. In the case study of this work, T_s (=0.0001s) is 52 times smaller than the electrical time constant (=L_q/R_s=0.00528).

$$
\begin{aligned}
T_{cs}v_d^{sat}(k-1) &= \left(\frac{R_sT_{cs}}{2}+L_d\right)i_d(t_k) \\
&+\left(\frac{R_sT_{cs}}{2}-L_d\right)i_d(t_k-T_{cs})-\frac{\omega_e(k)L_qT_{cs}}{2}\{i_q(t_k)+i_q(t_k-T_{cs})\} \\
T_{cs}v_q^{sat}(k-1) &= \left(\frac{R_sT_{cs}}{2}+L_q\right)i_q(t_k) \\
&+\left(\frac{R_sT_{cs}}{2}-L_q\right)i_q(t_k-T_{cs})+\frac{\omega_e(k)L_dT_{cs}}{2}\{i_d(t_k)+i_d(t_k-T_{cs})\}+\omega_e(k)F_mT_{cs}
\end{aligned}
\tag{4.5}
$$

The estimated dq currents $i_{dq}^{pre}(k)$ of the first current prediction can therefore be derived by solving (4.5).

$$
\begin{aligned}
i_d^{pre}(k) &= \frac{1}{a_1\omega_e(k)^2+a_2a_3}\Bigg[a_3T_{cs}v_d^{sat}(k-1)+\frac{L_qT_{cs}^2}{2}\omega_e(k)v_q^{sat}(k-1) \\
&-\{\omega_e(k)^2a_1+a_3a_4\}i_d(t_k-T_{cs})+L_q^2T_{cs}\omega_e(k)i_q(t_k-T_{cs})-\frac{L_qT_{cs}^2}{2}F_m\omega_e(k)^2\Bigg] \\
i_q^{pre}(k) &= \frac{1}{-a_1\omega_e(k)^2-a_2a_3}\Bigg[\frac{L_dT_{cs}^2}{2}\omega_e(k)v_d^{sat}(k-1)-a_2T_{cs}v_q^{sat}(k-1) \\
&+L_d^2T_{cs}\omega_e(k)i_d(t_k-T_{cs})+\{\omega_e(k)^2a_1+a_2a_5\}i_q(t_k-T_{cs})+a_2T_{cs}F_m\omega_e(k)\Bigg]
\end{aligned}
$$

where,

$$
a_1=\frac{L_dL_qT_{cs}^2}{4}, a_2=\frac{R_sT_{cs}}{2}+L_d, a_3=\frac{R_sT_{cs}}{2}+L_q, a_4=\frac{R_sT_{cs}}{2}-L_d, a_5=\frac{R_sT_{cs}}{2}-L_q
\tag{4.6}
$$

The **second current prediction** predicts the dq currents at t_{k+1} using the k^{th} saturated reference voltages $v_{dq}^{sat}(k)$ and the dq currents at t_k obtained from the first current prediction (4.3). Again, the $v_{dq}^{sat}(k)$ are available during the k^{th} sampling period since calculated during the $(k-1)^{th}$ sampling period; and under assumption 2 and 3, (4.1) can be rewritten as:

$$
\begin{aligned}
v_d^{sat}(k) &= R_si_d+L_d\frac{di_d}{dt}-\omega_e(k)L_qi_q \\
v_q^{sat}(k) &= R_si_q+L_q\frac{di_q}{dt}+\omega_e(k)L_di_d+\omega_e(k)F_m
\end{aligned}
\tag{4.7}
$$

Both sides of (4.7) are integrated from t_k to t_{k+1}.

$$\begin{aligned}\int_{t_k}^{t_{k+1}} v_d^{sat}(k)dt &= \int_{t_k}^{t_{k+1}} \Big\{R_s i_d + L_d\frac{di_d}{dt} - \omega_e(k)L_q i_q\Big\}dt \\ \int_{t_k}^{t_{k+1}} v_q^{sat}(k)dt &= \int_{t_k}^{t_{k+1}} \Big\{R_s i_q + L_q\frac{di_q}{dt} + \omega_e(k)L_d i_d + \omega_e(k)F_m\Big\}dt\end{aligned} \tag{4.8}$$

Again, based on the assumption 4, is obtained from 4.8.

$$\begin{aligned}T_s v_d^{sat}(k) &= \left(\frac{R_s T_s}{2} + L_d\right) i_d(t_{k+1}) \\ &+ \left(\frac{R_s T_s}{2} - L_d\right) i_d(t_k) - \frac{\omega_e(k)L_q T_s}{2}\big\{i_q(t_{k+1}) + i_q(t_k)\big\} \\ T_s v_q^{sat}(k) &= \left(\frac{R_s T_s}{2} + L_q\right) i_q(t_{k+1}) \\ &+ \left(\frac{R_s T_s}{2} - L_q\right) i_q(t_k) + \frac{\omega_e(k)L_d T_s}{2}\big\{i_d(t_{k+1}) + i_d(t_k)\big\} + \omega_e(k)F_m T_s\end{aligned} \tag{4.9}$$

The estimated dq currents $i_{dq}^{pre}(k+1)$ of the second current prediction can therefore be derived by solving (4.9).

$$\begin{aligned}i_d^{pre}(k+1) &= \frac{1}{b_1\omega_e(k)^2 + b_2 b_3}\Bigg[b_3 T_s v_d^{sat}(k) + \frac{L_q T_s^2}{2}\omega_e(k)v_q^{sat}(k) \\ &- \big\{\omega_e(k)^2 b_1 + b_3 b_4\big\} i_d^{pre}(t_k) + L_q^2 T_s \omega_e(k) i_q^{pre}(t_k) - \frac{L_q T_s^2}{2}F_m\omega_e(k)^2\Bigg] \\ i_q^{pre}(k+1) &= \frac{1}{-b_1\omega_e(k)^2 - b_2 b_3}\Bigg[\frac{L_d T_s^2}{2}\omega_e(k)v_d^{sat}(k) - b_2 T_s v_q^{sat}(k) \\ &+ L_d^2 T_s \omega_e(k) i_d^{pre}(t_k) + \big\{\omega_e(k)^2 b_1 + b_2 b_5\big\} i_q^{pre}(t_k) + b_2 T_s F_m \omega_e(k)\Bigg]\end{aligned} \tag{4.10}$$

where,

$$b_1 = \frac{L_d L_q T_s^2}{4}, b_2 = \frac{R_s T_s}{2} + L_d, b_3 = \frac{R_s T_s}{2} + L_q, b_4 = \frac{R_s T_s}{2} - L_d, b_5 = \frac{R_s T_s}{2} - L_q$$

4.1.2 Reference Voltage Calculation

After the current at t_{k+1} has been predicted, according the assumptions 1-5, (4.2) can be rewritten as below to calculate the $(k+1)^{th}$ reference voltages $v_{dq}^{ref}(k+1)$:

$$\begin{aligned} v_d^{ref}(k+1) &= R_s i_d^{pre}(t_{k+1}) + L_d \frac{i_d^{ref}(t_k) - i_d^{pre}(t_{k+1})}{T_s} - \omega_e(k) L_q i_q^{pre}(t_{k+1}) \\ v_q^{ref}(k+1) &= R_s i_q^{pre}(t_{k+1}) + L_q \frac{i_q^{ref}(t_k) - i_q^{pre}(t_{k+1})}{T_s} + \omega_e(k) L_d i_d^{pre}(t_{k+1}) + \omega_e(k) F_m \end{aligned} \tag{4.11}$$

As mentioned before in assumption 5, it is the nature of the DBCC to have a minimum delay of two sampling periods for the current to reach the reference, half of which is introduced by the calculation delay of the digital control system. The only way to reduce the delay of $2T_s$ is to predict the reference currents $i_{dq}^{ref}(t_{k+2})$ during the k^{th} sampling period. Consequently, (4.11) can be rewritten as (4.12).

$$\begin{aligned} v_d^{ref}(k+1) &= R_s i_d^{pre}(t_{k+1}) + L_d \frac{i_d^{ref}(t_{k+2}) - i_d^{pre}(t_{k+1})}{T_s} - \omega_e(k) L_q i_q^{pre}(t_{k+1}) \\ v_q^{ref}(k+1) &= R_s i_q^{pre}(t_{k+1}) + L_q \frac{i_q^{ref}(t_{k+2}) - i_q^{pre}(t_{k+1})}{T_s} + \omega_e(k) L_d i_d^{pre}(t_{k+1}) + \omega_e(k) F_m \end{aligned} \tag{4.12}$$

However, the reference currents can only be predicted if they are periodic. The prediction of reference currents are therefore **not** implemented in this work.

4.1.3 Rotor Movement Compensation

As mentioned above, assumptions 3 and 4 are applied to simplify the dynamic equations of PMSM. It is true in assumption 3 that the dq axis reference voltages v_{dq}^{ref} are updated only at the beginning of each sampling period and hold constant until the next sampling period begins. However, the real dq axis shifts from the dq axis used in the controller due

to the rotor movement. Therefore the instantaneous dq voltages $v_{dq}^{real}(t)$ seen by the motor may be different from v_{dq}^{ref}. **Especially at high speed, the difference between these two can be significant**. The non-constant dq axis voltages in the motor is responsible for the non-linear current profile. Hence, assumption 4 may not hold true for the same reason of rotor movement.

Overall, due to the rotor movement, assumptions 3 and 4 may be wrong. These mistakes are caused by the mismatches between reference voltages and voltages effectively applied to the motor, and lead to offsets in current predictions as well as in current responses of DBCC. In order to avoid this effect, the rotor movement needs to be compensated. An existing compensation method as in 4.1.3.1 and the proposed compensation method as in 4.1.3.2 are compared in Fig.4.14(a).

4.1.3.1 Existing Compensation Methods

An existing way of compensating the error caused by rotor movement as presented in [75] is a simple, but inaccurate method. For example, in Fig.4.3, during the $(k+1)^{th}$ sampling period, rotor moves from rotor dq axis $dq(t_{k+1})$ to $dq(t_{k+2})$. The rotor speed during the $(k+1)^{th}$ sampling period is assumed to be the same as the speed during the k^{th} sampling period. Therefore, the relationship of the electrical position $\theta_e(t_{k+2}) - \theta_e(t_{k+1}) = \omega_e(k)T_s$ holds; and the middle position of the rotor dq axis during the $(k+1)^{th}$ sampling period can be estimated as $\theta_e(t_{k+1}) + 0.5\omega_e(k)T_s$. The key concept of this compensation method is to align the dq axis in the controller to this estimated middle position, which can be implemented simply by using $\theta_e(t_{k+1}) + 0.5\omega_e(k)T_s$ instead of $\theta_e(t_{k+1})$ in the dq to abc transformation as shown in Fig.4.2. The rotor movement compensation block in Fig.4.2 is not needed in this case.

However, the instantaneous dq axis voltages when the rotor electrical position is $\theta_e(t_{k+1}) + 0.5\omega_e(k)T_s$ are not necessarily equal to the average value of the real dq axis voltages over the whole sampling period. Therefore, the effectiveness of this compensation method degrades as the rotor speed increases.

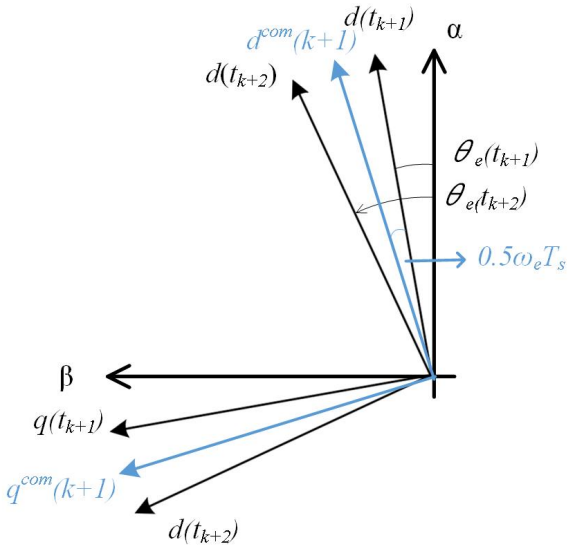


Figure 4.3: dq reference frame for the existing rotor movement compensation

4.1.3.2 Novel compensation Method

The novel rotor movement compensation technique proposed for this work is to apply the compensated dq voltage references $v_{dq}^{com}(k+1)$ at t_{k+1} so that the average dq voltages $v_{dq}^{avg}(k+1)$ effectively applied to the motor during the $(k+1)^{th}$ sampling period equal to $v_{dq}^{ref}(k+1)$. The relationships between the instantaneous $v_{dq}^{real}(t)$ and the $v_{dq}^{com}(k+1)$ are described by (4.13) with reference to the rotor dq reference frame as shown in Fig.4.4.

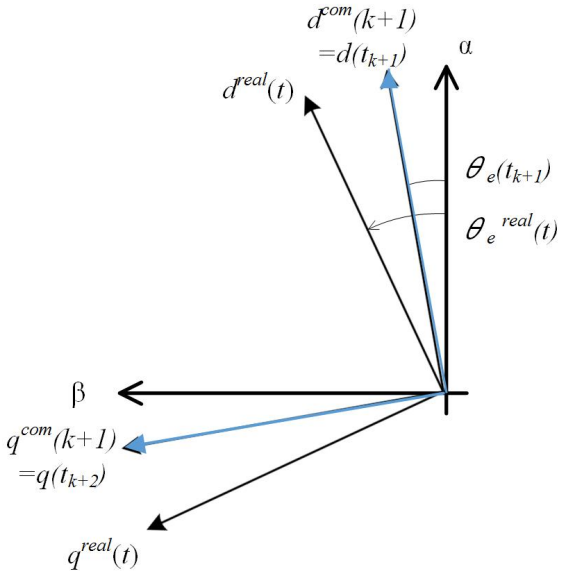


Figure 4.4: dq reference frame for the proposed rotor movement compensation

$$
\begin{aligned}
v_d^{real}(t) &= cos\Big(\theta_e^{real}(t) - \theta_e(t_{k+1})\Big)v_d^{com}(k+1) + sin\Big(\theta_e^{real}(t) - \theta_e(t_{k+1})\Big)v_q^{com}(k+1) \\
v_q^{real}(t) &= -sin\Big(\theta_e^{real}(t) - \theta_e(t_{k+1})\Big)v_d^{com}(k+1) + cos\Big(\theta_e^{real}(t) - \theta_e(t_{k+1})\Big)v_q^{com}(k+1)
\end{aligned}
\tag{4.13}
$$

The term $\big(\theta_e^{real}(t) - \theta_e^{real}(t_{k+1})\big)$ represents the difference between the instantaneous electrical position and the initial electrical position at the beginning of each sampling period. Therefore, $\big(\theta_e^{real}(t) - \theta_e^{real}(t_{k+1})\big)$ can be replaced by $\omega_e(k)t$, $t \in [0, T_s]$. Both sides of (4.13) are integrated from 0 to T_s to obtain:

$$
\begin{aligned}
T_s v_d^{avg}(k+1) &= \int_0^{T_s} v_d^{real}(t)dt \\
&= \int_0^{T_s} cos\Big(\omega_e(k)t\Big)v_d^{com}(k+1)dt + \int_0^{T_s} sin\Big(\omega_e(k)t\Big)v_q^{com}(k+1)dt \\
T_s v_q^{avg}(k+1) &= \int_0^{T_s} v_q^{real}(t)dt \\
&= -\int_0^{T_s} sin\Big(\omega_e(k)t\Big)v_d^{com}(k+1)dt + \int_0^{T_s} cos\Big(\omega_e(k)t\Big)v_q^{com}(k+1)dt
\end{aligned}
\tag{4.14}
$$

The compensated voltages $v_{dq}^{com}(k+1)$ and the rotor angular speed $\omega_e(k)$ are assumed constant for the integration interval as the assumption 3. By imposing that the applied $v_{dq}^{avg}(k+1)$ equals to $v_{dq}^{ref}(k+1)$, the following equations are obtained from (4.14).

$$
\begin{aligned}
T_s v_d^{ref}(k+1) &= \frac{sin\Big(\omega_e(k)T_s\Big)}{\omega_e(k)}v_d^{com}(k+1) - \frac{cos\Big(\omega_e(k)T_s\Big) - 1}{\omega_e(k)}v_q^{com}(k+1) \\
T_s v_q^{ref}(k+1) &= \frac{cos\Big(\omega_e(k)T_s\Big) - 1}{\omega_e(k)}v_d^{com}(k+1) + \frac{sin\Big(\omega_e(k)T_s\Big)}{\omega_e(k)}v_q^{com}(k+1)
\end{aligned}
\tag{4.15}
$$

Therefore the $v_{dq}^{com}(k+1)$ are derived from (4.15).

$$v_d^{com}(k+1) = \frac{T_s sin\Big(\omega_e(k)T_s\Big)\omega_e(k)v_d^{ref}(k+1) + T_s\Big\{cos\Big(\omega_e(k)T_s\Big) - 1\Big\}\omega_e(k)v_q^{ref}(k+1)}{sin^2\Big(\omega_e(k)T_s\Big) + \Big\{cos\Big(\omega_e(k)T_s\Big) - 1\Big\}^2}$$
$$v_q^{com}(k+1) = \frac{-T_s\Big\{cos\Big(\omega_e(k)T_s\Big) - 1\Big\}\omega_e(k)v_d^{ref}(k+1) + T_s sin\Big(\omega_e(k)T_s\Big)\omega_e(k)v_q^{ref}(k+1)}{sin^2\Big(\omega_e(k)T_s\Big) + \Big\{cos\Big(\omega_e(k)T_s\Big) - 1\Big\}^2} \tag{4.16}$$

4.1.4 Inverter Non-linearity Compensation

Apart from the rotor movement, another possible cause for an error between the reference voltage calculated in the deadbeat current controller and the voltage effectively applied to the motor is the non-linearities of the inverter, including the dead time effect and voltage drop on components. As a consequence of such voltage error, steady state error occurs in the current response of DBCC. The influence of dead time on the steady state error is shown in Fig.4.11.

The compensation method for inverter non-linearities chosen for this work is the one proposed in [76]. A stationary dq axis PI current control loop is used for regulating the three-phase currents in the inverter to be constant d.c. values when $i_q = 0$. Then, the steady state dq axis reference voltages are recorded for different levels of the three-phase currents. According to [76], the overall effect of inverter non-linearities on each inverter leg can be seen as combination of a time offset due to the time distortions caused by dead-time and delay times, and two voltage offsets caused by the voltage drops on power devices. Offset is mainly dependent on dead-time and on the sign of each phase current. The time offset and two voltage offsets can be derived from the recorded voltage waveforms. Based on the equations given in [76], the time offset and two voltage offsets can be calculated, and a lookup table is therefore built for compensating the errors at different phase currents. An example of the compensation results can be seen from Fig.4.12(a).

4.1.5 Steady State Error Compensation

The steady state error of the current response in DBCC can be yielded by several causes. Some main causes can be listed as follow:

1) The mismatch of parameters as shown in Fig.4.10

2) The rotor movement effect as discussed in 4.1.3.

3) The inverter non-linearities as discussed in 4.1.4.

Although both of the latter two causes can be compensated individually, considering the imperfections in such compensations and the parameter variations during operation, it is likely to have remaining steady state errors in the current response. In order to compensate the remaining steady state errors, a common method is to combine the DBCC with an integrator. However, it is tricky to tune the gain of integrator due to the following reasons:

1) If the gain is too high, the strong integration effect during transients may produce overshoots and degrade the dynamics of DBCC.

2) If the gain is too low, it may take a long time for the offsets to be reduced to zero and make the compensation less valued.

Overall, a possible solution is to connect the integrator as a plug-in structure as shown in Fig.4.5. The input of the integrator is limited to small values to disable the integration at transients. The gain of the integrator can then be chosen to be a high value to achieve fast convergence.

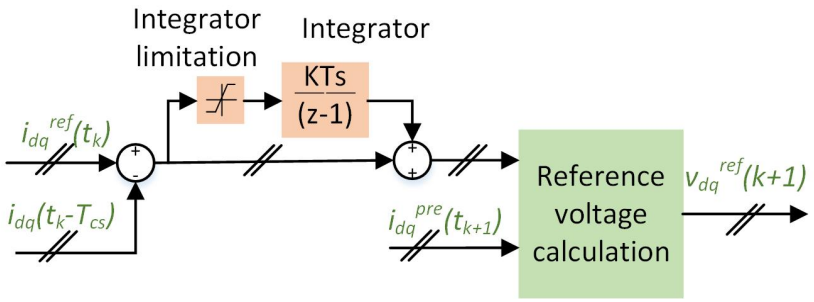


Figure 4.5: The DBCC with plug-in integrator for removing steady state error

The performance of this compensation is shown in Fig.4.15. However, the repetitive controller proposed later in Chapter 5 already has the ability to compensate the steady state errors in the current responses. For the purpose of torque ripple reduction, it is **not** necessary to include this integrator in the current loop.

4.2 Simulation and Experimental Tests

An ideal simulation model is built using Matlab/Simulink as shown in Fig.4.6. The simulation model includes an ideal PMSM with perfect sine distributed magnetic flux and fixed dq axis inductances, the deadbeat current controller as shown in Fig.4.2, the integrator (only activated in the test in 4.1.4) for steady state error compensation as shown in Fig.4.5, and a three-phase inverter which is modelled considering a dead time related voltage error of which the polarity varies according to the directions of phase currents. The rotor electrical position at $t_k - T_{cs}$ (i.e. $\theta_e(t_k - T_{cs})$) is estimated in order to synchronize the position feedback with the current feedback.

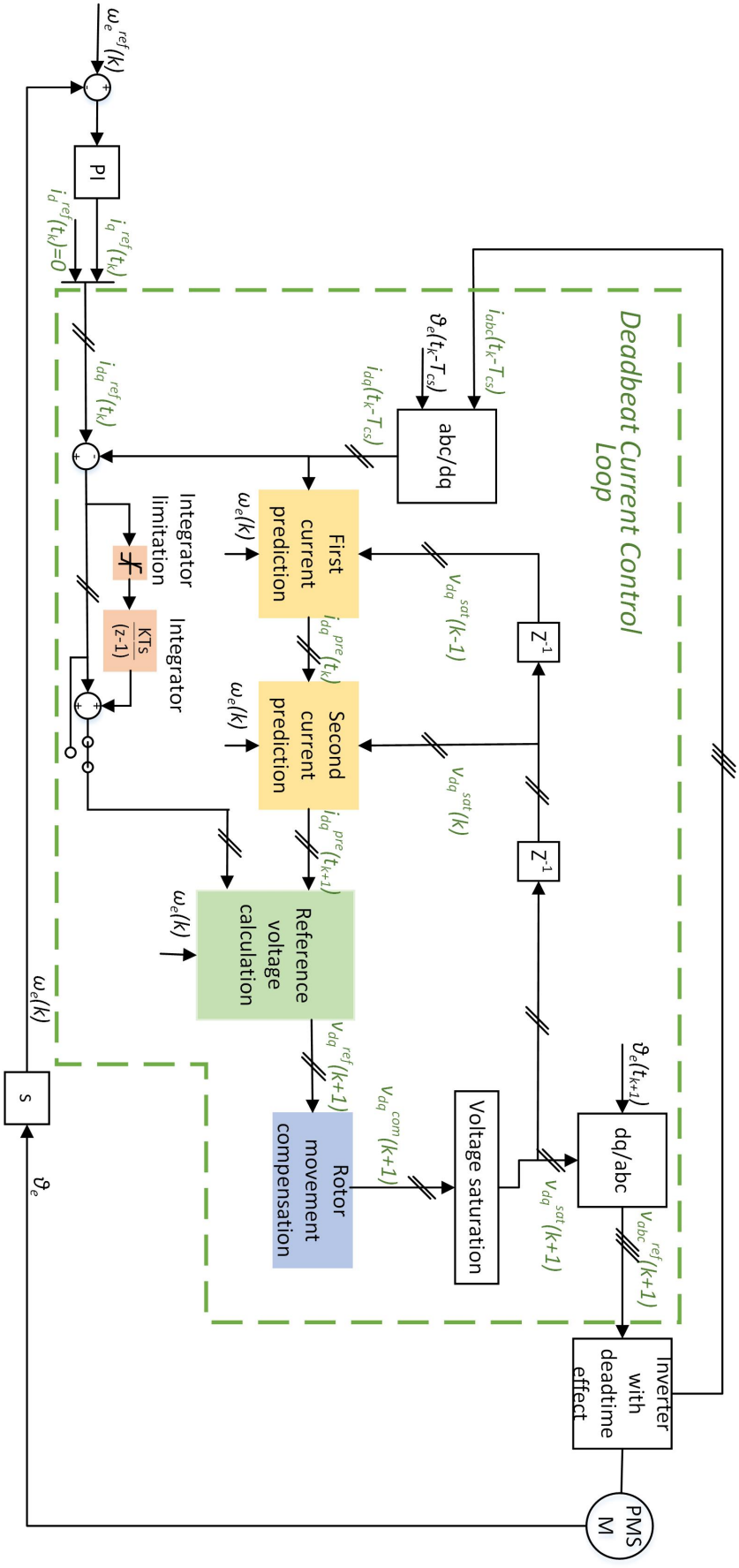
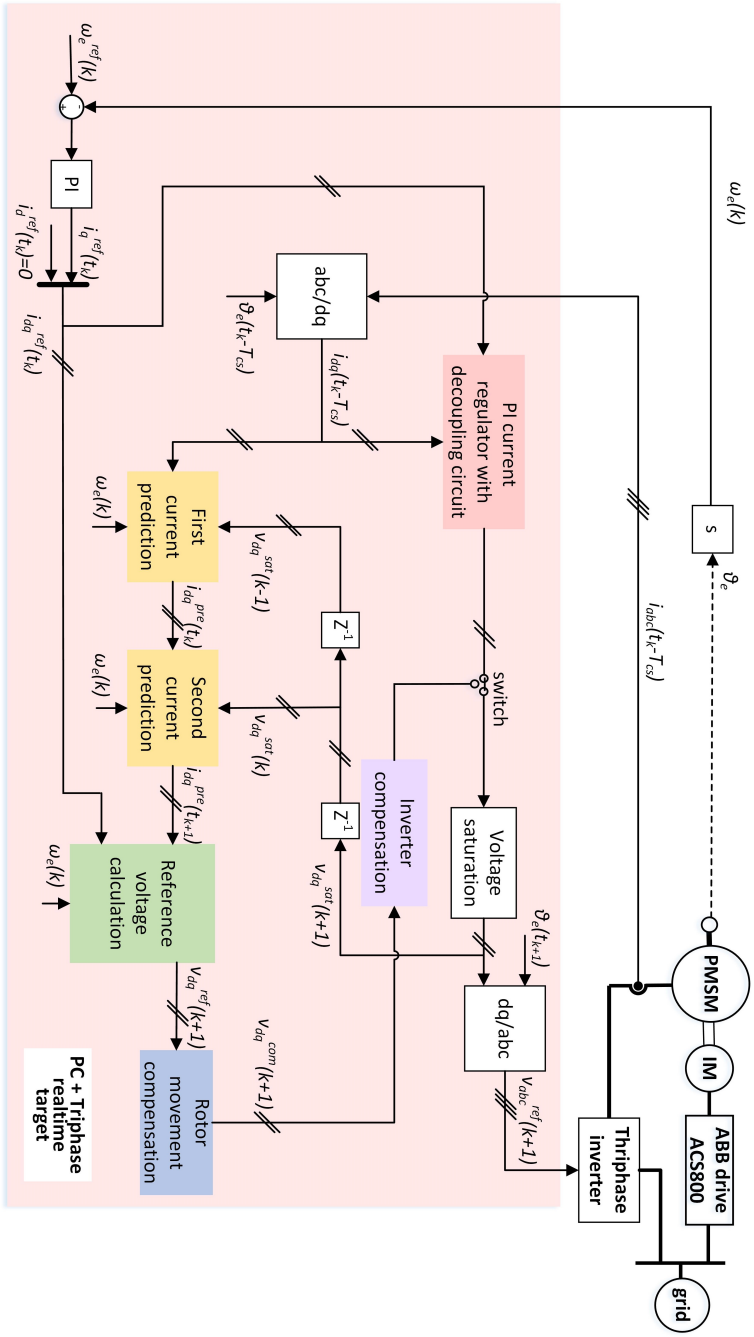


Figure 4.6: Simulation model for the assessment of deadbeat current control

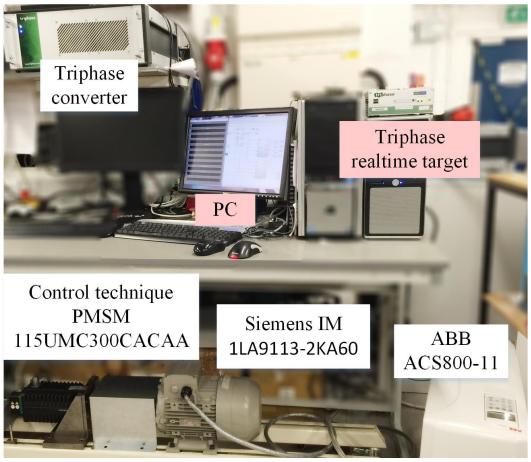
Meanwhile, an experimental test rig is set up as in Fig.4.7 below. As can be seen, a DBCC and a PI controller are both implemented for comparison and can be switched in the software easily. A lookup table is built for compensating the inverter non-linearities as in 4.1.4. A PI regulator based speed control loop is used to keep the rotor speed constant or generate a high frequency sinusoidal signal in the i_q^{ref} for test reasons; the speed loop bandwidth is chosen to be 65Hz. A "Triphase" evaluation system, composed of an inverter and a real-time control platform is used in the experimental tests. The program in the FPGA inside the Triphase Realtime Target can be compiled directly using Matlab Simulink, and the Triphase converter is controlled by the Realtime Target to drive a Control Technique PMSM (115UMC300) with parameters as shown in Table 4.1. In such a "Triphase" evaluation system, current is over-sampled and the **mean value** is used as feedback. A Siemens induction motor (IM) (1LA9113) is used as load.

Table 4.1: Parameters (tuned) of the PMSM for DB current loop tests

Parameter	**Value [Unit]**	**Parameter**	**Value [Unit]**
Rated power	2.54 [kW]	d axis inductance L_d	0.0045 [H]
Rated speed	3000 [rpm]	q axis inductance L_q	0.0074 [H]
Rated torque	9.4 [Nm]	Stator resistance R_s	1.4 [Ω]
Rated line voltage (RMS)	400 [V]	Magnetic flux linkage F_m	0.237 [Wb]
Rated current (RMS)	5.9 [A]	Inertia J	0.0007 [kg/m^2]
Number of pole pairs	3	Number of slots and teeth	18



(a) the diagram



(b) the photo

Figure 4.7: Experimental rig for the assessment of deadbeat current control

The evaluation of DBCC is carried out with respect to **three main criteria** as below. This section is therefore divided into three parts, each one presenting the results relative to these three issues.

1) the influence of detuned parameters and inverter dead time on the performance of DBCC

2) the effectiveness of the proposed DBCC with two steps ahead predictions and the rotor movement compensation

3) the advantage of DBCC over classical PI regulator

4.2.1 Performance Assessment

Regarding the desired performance for reducing torque ripple in PMSMs, it would be ideal if the DBCC could track a wide frequency range of sinusoidal references with no attenuation, smallest possible phase shift and no offset in the average value. In this section, the Bode diagram is mapped **in simulation** to verify bandwidth, amplitude and phase response of DBCC, while the steady state error in step response is mapped to verify the offset. In summary, the following results in 4.2.1.1 and 4.2.1.2 show that the dynamic performance (amplitude and delay) is affected mainly by detuned inductances while the steady state performance is affected by the detuned magnetic flux.

4.2.1.1 Dynamic Assessment using Bode Diagram

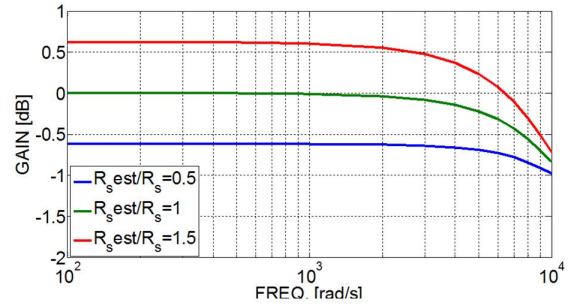
For mapping Bode diagrams as shown in Fig.4.8 and Fig.4.9, the reference current i_q^{ref} is set to be a sinusoidal signal with bias of 8.34A (the peak value of rated current), the motor mechanical speed N_r is set to be constant in simulation, and the amplification or attenuation in the magnitude of the response as well as the phase shift are calculated. This procedure has been repeated iteratively for current references of different frequencies and with different motor parameter detuned (Fig.4.8) and different motor speed (Fig.4.9). For example, $L_d^{est}/L_d = 0.5$ in Fig.4.8(b) means that L_d^{est} (estimated in the control) is

50% of L_d (real value in the motor). It is also to be noticed that the results in Fig.4.9 for 10000rad/s and 5000rad/s are confirmed by the experimental results shown in Fig.4.16 and Fig.4.17 in section 4.2.3.

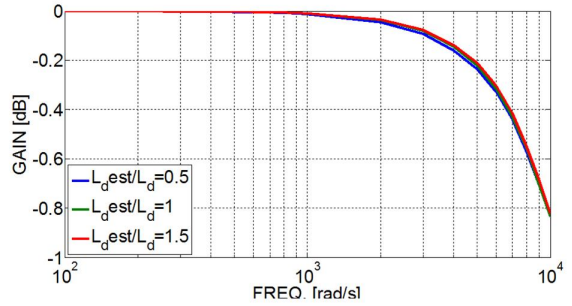
Regarding **bandwidth and amplitude response**, the high bandwidth characteristic of DBCC is reliably maintained for varying mechanical speed N_r (Fig.4.9) and for parameter mismatch of the magnetic flux F_m (Fig.4.8(d)). Also, since i_d^{ref}=0 A, the control performance is reliable with detuned L_d (Fig.4.8(b)). A mismatch of R_s (Fig.4.8(a)) will not necessarily reduce the bandwidth, but may result in an increase (when $R_s^{est} > R_s$) or a decrease (when $R_s^{est} < R_s$) in the amplitude of i_q. In case of detuned L_q (Fig.4.8(c)), the bandwidth of DBCC can be significantly reduced when $L_q^{est} < L_q$; when $L_q^{est} > L_q$ the amplification introduced at high frequencies may bring challenges for the stability of the i_q control loop. **It can thus be suggested that the design of DBCC needs to consider a smaller dq axis inductance for stability reasons, but not too small to avoid sacrificing the bandwidth.** Fig.4.9 shows that the dynamic performance of DBCC barely varies with rotor speed, and **the bandwidth of DBCC, when the sampling frequency is 10kHz, can reach as high as about 19000rad/s(=3kHz)**.

Before discussing the **phase response**, it would be necessary to define the smallest possible phase shift for DBCC. Theoretically, the reference current can be achieved only $2T_s$ after the reference change has been detected by the control or even longer (an example is given in Fig.4.12 in 4.2.2.1) depending on the demanded current change and the available DC bus voltage. As a result, when operating below the voltage limitation, the phase shift of a signal is proportional to its frequency and equals to the frequency multiplied by $2T_s$. This calculation is supported by almost all the phase plots in Fig.4.8 and Fig.4.9 apart from the case of the detuned L_q (Fig.4.8(c)). Considering the worst case when $L_q^{est} = 0.5L_q$ in Fig.4.8(c), the phase lagging for 5000 rad/s is 85.7°(=1.496rad) which is 28.4° more than the theoretical value (57.3°). However, if we convert the phase delay into a time delay (i.e. $1.496/5000 = 299\mu s$), it can be seen that the delay is less than 150 % of the theoretical value ($2T_s$). **It is interesting to notice that the delay introduced by DBCC is affected mainly by detuned inductances, however, no significant influence is noticed when the detuned inductance is within a reasonable range**

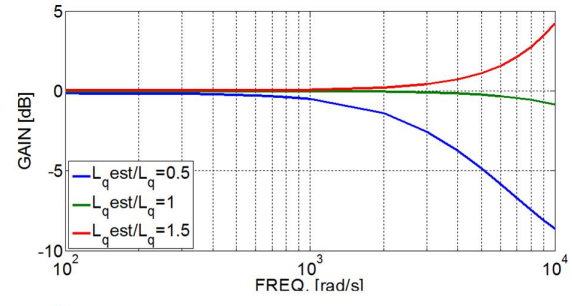
(50%–150% of the real value).



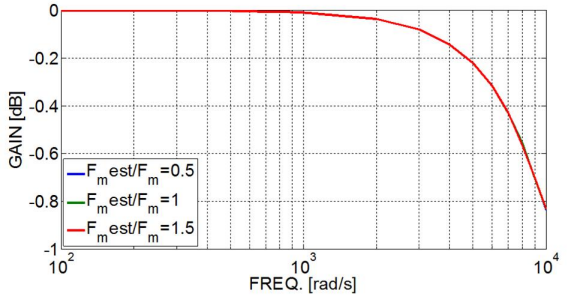
(a) R_s^{est}=(0.5↔1.5)R_s



(b) L_d^{est}=(0.5↔1.5)L_d



(c) L_q^{est}=(0.5↔1.5)L_q



(d) F_m^{est}=(0.5↔1.5)F_m

Figure 4.8: Simulation results of bode diagrams of q axis current control loop with parameter detuned (i_d^{ref}=0A, bias of i_q^{ref}=8.34A, T_s=100μs, N_r=3000rpm)

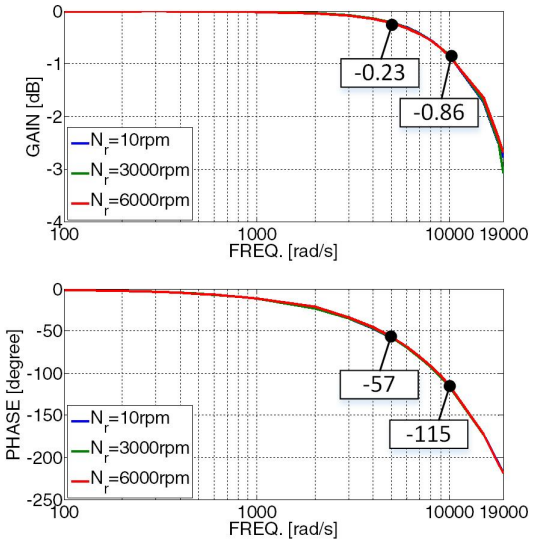
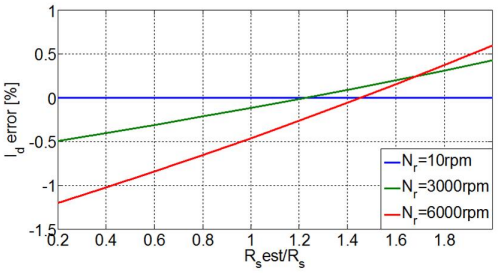


Figure 4.9: Simulation results of bode diagrams of q axis current control loop with correct parameters (i_d^{ref}=0A, bias of i_q^{ref}=8.34A, T_s=100μs,N_r=10$\leftrightarrow$6000rpm)

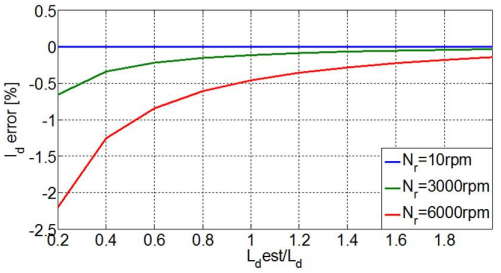
4.2.1.2 Steady State Assessment

For the steady state error plots shown in Fig.4.10 and Fig.4.11, both the reference current i_{dq}^{ref} and motor mechanical speed N_r are set to be constant, and the errors at steady state in the response current i_{dq} are calculated using (4.17). Again, the procedure is repeated for different settings of motor speed, detuned motor parameter, reference current value, and inverter dead time. Fig.4.11 shows the influence of switching devices dead time on the steady state errors in current responses using DBCC. The sampling frequency of DBCC is fixed at 10kHz while the dead time varies from $0\mu s$ to $10\mu s$.

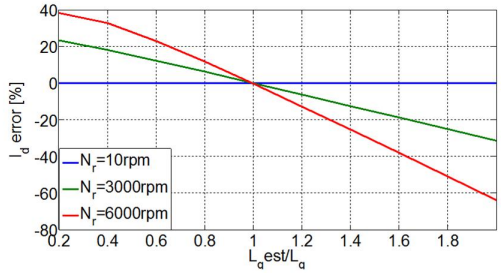
$$\begin{aligned} \text{Steady-state error for} \quad i_d &= \frac{i_d - i_d^{ref}}{\sqrt{\left(i_d^{ref}\right)^2 + \left(i_q^{ref}\right)^2}} \times 100 \\ \text{Steady-state error for} \quad i_q &= \frac{i_q - i_q^{ref}}{\sqrt{\left(i_d^{ref}\right)^2 + \left(i_q^{ref}\right)^2}} \times 100 \end{aligned} \tag{4.17}$$



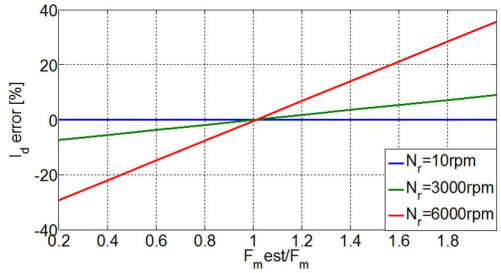
(a) R_s^{est}=(0.2↔2)R_s



(b) L_d^{est}=(0.2↔2)L_d

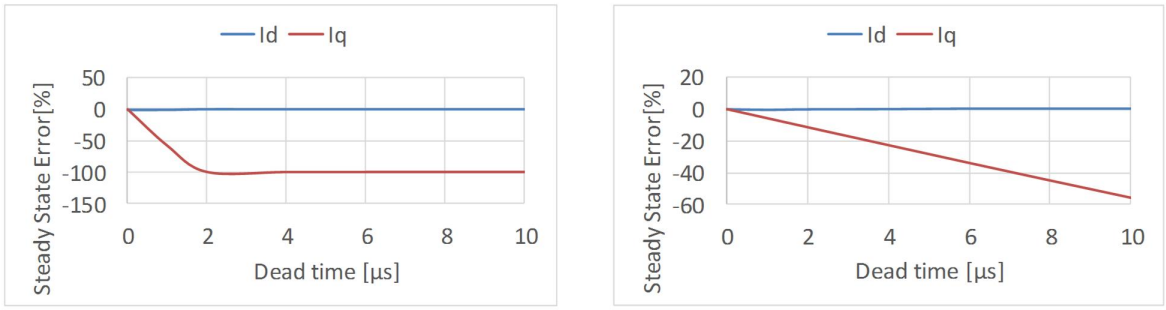


(c) L_q^{est}=(0.2↔2)L_q



(d) F_m^{est}=(0.2↔2)F_m

Figure 4.10: Simulation results of steady state current errors of DBCC due to parameter detuning (i_d^{ref}=0A, i_q^{ref}=8.34A, N_r=10↔6000rpm)



(a) i_q^{ref}=0.8A (b) i_q^{ref}=8.34A

Figure 4.11: Simulation results of steady state current errors of DBCC due to dead time (i_d^{ref}=0A, N_r=3000rpm)

Considering that practically the machine parameters are tuned before actual operation, the mismatches in parameters are likely to be within ±20%; also for surface mounted PMSM, the d axis reference current i_d^{ref} is normally controlled to be zero without field weakening, one important finding from Fig.4.10 and Fig.4.11 is that the steady state offset of DBCC is the most sensitive to the magnetic flux F_m at high speed and the dead time of inverter. **A concern is that even if no steady state errors are observed, there is still a chance that both dead time compensation and the magnetic flux F_m^{est} are wrong. A straight forward way to check is that speed dependent steady state error will be observed if F_m^{est} is wrong.**

Overall, the results in Fig.4.10 and Fig.4.11 give **a clue for offline tuning of the machine parameters through steady state errors**. For example, if the d axis current i_d is observed to be too small, it may indicate that F_m^{est}, R_s^{est} are too small and/or L_q^{est} is too large; similarly, if the q axis current i_q is observed to be too small, it may indicate that F_m^{est}, L_q^{est}, R_s^{est} are too small and/or L_d^{est} is too large.

4.2.2 Effectiveness of the proposed DBCC

Turning now to the **experimental** tests of the proposed DBCC with two steps current prediction and rotor movement compensation, the machine parameters are tuned based on the findings in 4.2.1.2 and the results are as shown in Table 4.1.

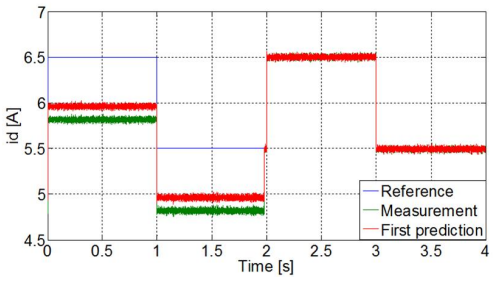
4.2.2.1 Two Steps Prediction and Operating at Physical Limit

The current responses of DBCC to rectangular current references within and beyond voltage limitation are shown in Fig.4.12(c) and Fig.4.12(d). In both cases, the rectangular current references are generated by the i_d current loop. Moreover, the current responses of the same rectangular reference using the proposed DBCC (with two steps current prediction) and the traditional DBCC (with only one step prediction) are compared in Fig.4.12(b) to show the necessity of having two steps current prediction instead of one.

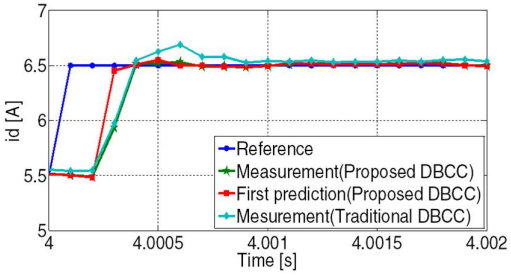
It can be seen from Fig.4.12(a) that with properly tuned parameters and properly compensated inverter nonlinearities, **the proposed DBCC can achieve zero steady state error in the response**.

By comparing the measured current (green) and the first prediction current (red) in Fig.4.12(b), it can be seen that **the first prediction current reaches its demand after** $2T_s$ and the measurement current, since it is a **mean current**, reaches the demand after $3T_s$. By comparing the measured current (green) for the proposed DBCC and the measured current (light blue) for the traditional DBCC (the first prediction is removed), it can be seen that **an overshoot of 18.6% occurs without the first prediction**. This is because the first prediction works to predict the real instantaneous current at 4.0003s so that the controller can have a fairly accurate judgement of whether the demand has been achieved or not, and the overshoot is avoided consequently. These results confirm the necessity of the first prediction as discussed at the beginning of 4.1.1.

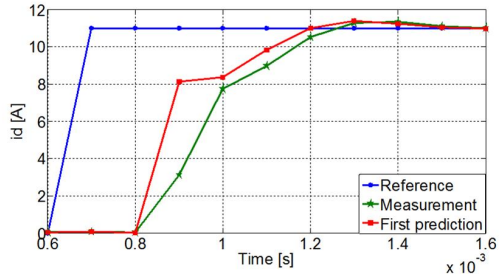
Fig.4.12(c) shows that **a longer settling time** (as discussed in 4.2.1.1 regarding the phase response) of 5Ts is required for DBCC to achieve a current step of 11A with the peak value of the three-phase voltage limited at 450V.



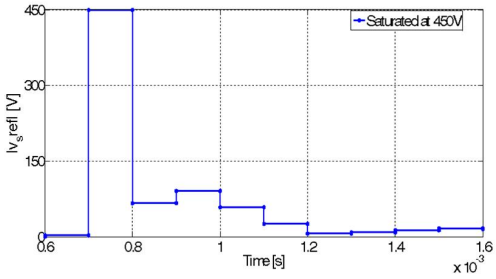
(a) the inverter is compensated at 2s



(b) the traditional DBCC vs the proposed DBCC



(c) current response at physical limit

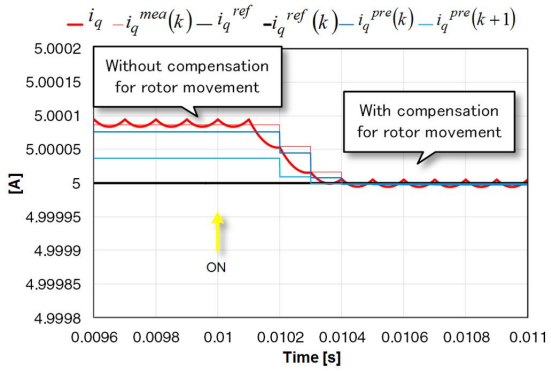


(d) voltage module ($\sqrt{v_d^2 + v_q^2}$) at physical limit

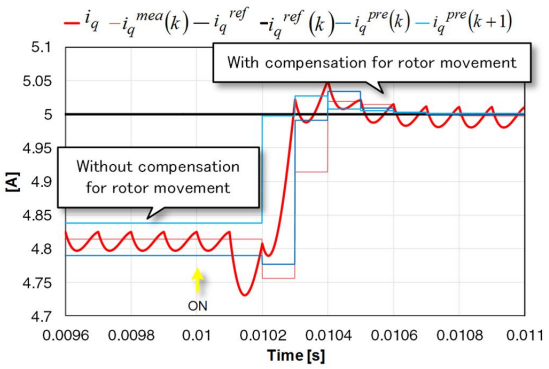
Figure 4.12: Experimental results of rectangular current responses of DBCC

4.2.2.2 Rotor Movement Compensation

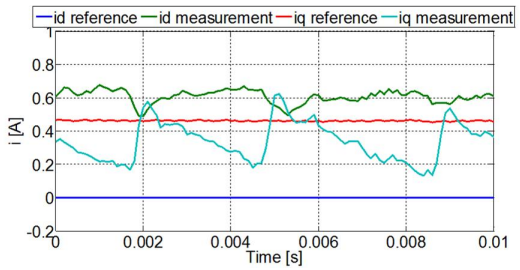
Figs.4.13(a) and 4.13(b) show the effects of the rotor movement compensation in simulation, which starts at 0.01s. The voltage references maintain two sampling time delays including a calculation step. Therefore it is shown that the offsets start to decrease after 2 sampling periods. Similarly, the experimental results in Figs.4.13(c) and 4.13(d) confirm the effectiveness of the proposed rotor movement compensation. The steady errors can be calculated by (4.17), where mean values are used for i_d and i_q in the equations. By adding the proposed rotor position compensation, at 1000 rpm, the steady state error in i_d reduces from 132% to 122%, while the error in i_q reduces slightly from -19% to -17%. Meanwhile, at 2930 rpm, the steady state error in i_d reduces significantly from 81% to 27%, while the error in i_q reduces by half from -37% to -18%. It is obvious from both the simulation and the experiment that **the rotor position compensation works for all speeds, but is much more effective at high speed** as discussed in 4.1.3.2. Unfortunately, due to the parameter variations in motor and imperfect inverter compensation, the rotor position compensation method cannot remove all steady state errors in the experiments.



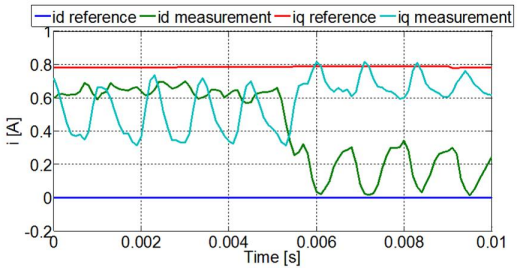
(a) Simulation results, N_r=100rpm, i_d^{ref}=0A, i_q^{ref}=5A



(b) Simulation results, N_r=5000rpm, i_d^{ref}=0A, i_q^{ref}=5A



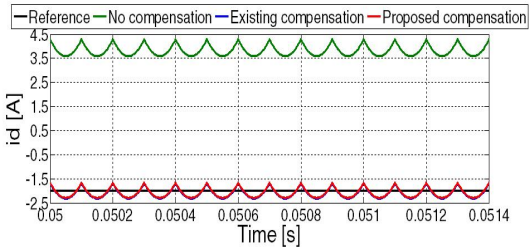
(c) Experimental results, speed control N_r=1000 rpm, compensation activated at 0.005s



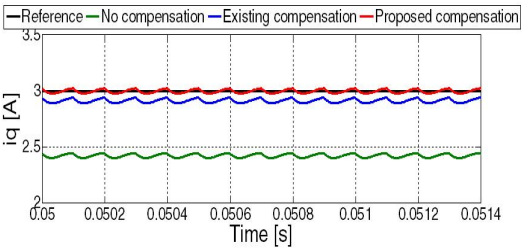
(d) Experimental results, speed control N_r=2930 rpm, compensation activated at 0.005s

Figure 4.13: Effectiveness of the proposed rotor movement compensation

A comparison between the existing rotor movement compensation in 4.1.3.1 and the proposed compensation in 4.1.3.2 is shown in Fig.4.14(a). It can be seen that, both of the compensation methods work, however, **the proposed method performs better offset reduction at high speed.**



(a) Simulation results of i_d



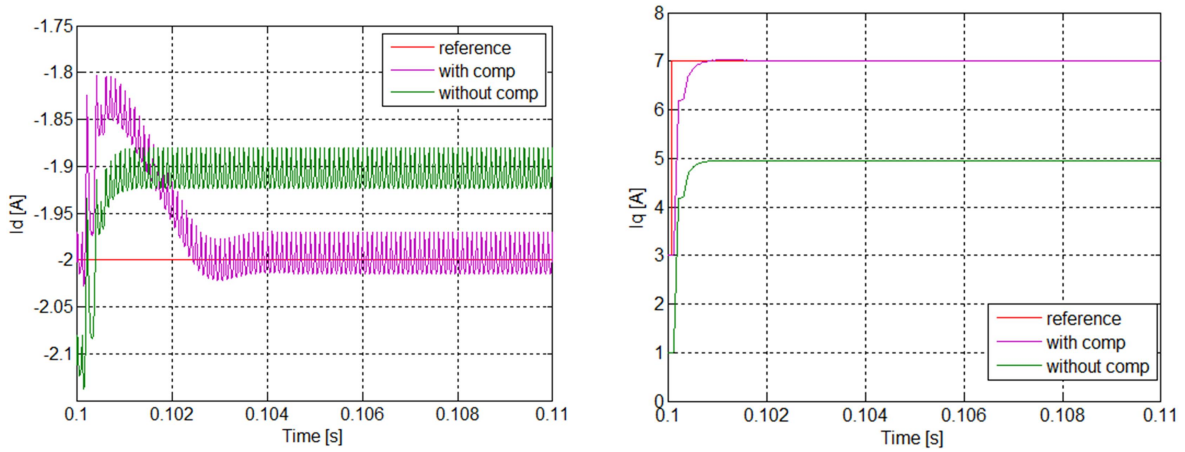
(b) Simulation results of i_q

Figure 4.14: Comparison of the proposed rotor movement compensation and existing compensation (Simulation results, i_d^{ref}=-2A, i_q^{ref}=3A, N_r=10000rpm)

4.2.2.3 Steady State Error compensation

As discussed in 4.1.5, the integrator limitation and the gain of integrator in Fig.4.5 need to be carefully tuned. The following compensation results are acquired by setting the integrator limitation to be ±0.05A and gain of the integrator equals to 2000. All the parameters in the control are set to be 80% of the real values set in the motor.

As it can be seen, this compensation can achieve zero steady state error in 0.003s. It may be useful in case only current control loop is activated (no outer loops activated, like when under the torque control mode of servomotor drives) and when the steady state magnitude of the current response is critical. In case outer loops are present, there may be no need to include this integrator since the outer loop may already be able to cancel the offsets.



(a) Simulation results of i_d

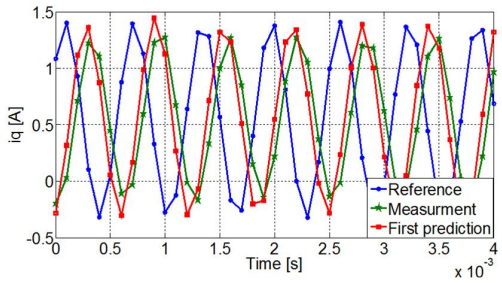
(b) Simulation results of i_q

Figure 4.15: Effectiveness of the steady state error compensation with plug-in integrator (Simulation results: All the parameters in controller are 80% of the values in motor,i_d^{ref}=-2A, i_q^{ref} steps from 3A to 7A at 0.1s)

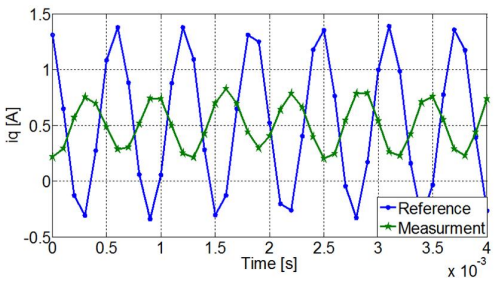
4.2.3 Comparison Between DBCC and PI

Furthermore, the performance of DBCC is compared with that of a conventional *dq* PI current regulator with a decoupling circuit. The bandwidth of the *dq* PI control loop used for comparison is designed to be around 900Hz (explanations will be given later) with classical SISO control design methods. The no load test results for the current responses

to a very high frequency current reference (10000rad/s) are shown in Fig.4.16. The no load and full load test results for the current responses to a high frequency current reference (5000rad/s) are shown in Fig.4.17. The full load operation is tested using a commercial drive (ABB ACS800-11). The ABB drive is operated under a speed control mode. The rotor speed is kept at 2500rpm, and the i_q^{ref} is set at the rated value of 8.34A. No tests are performed for frequencies higher than 10000rad/s since it is unlikely to have speed ripple at such high frequencies due to the low-pass filtering effect of the mechanical system.

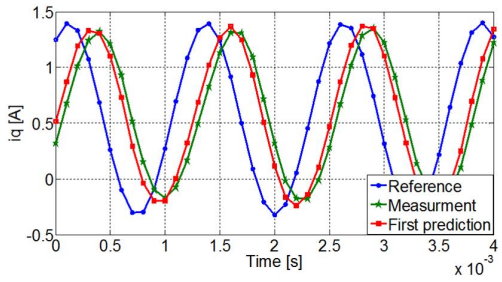


(a) No load, DBCC

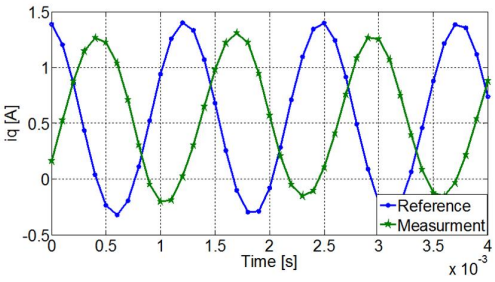


(b) No load, PI

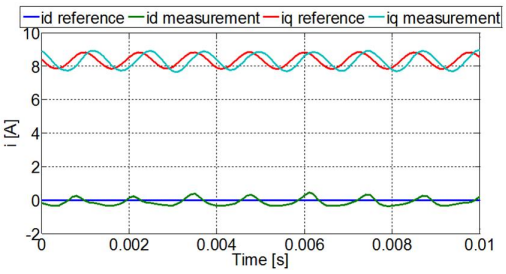
Figure 4.16: Experimental results of current responses for 10000 rad/s ($N_r^{ref} = 2930 + 30sin(10000t)$rpm , i_d^{ref}=0A)



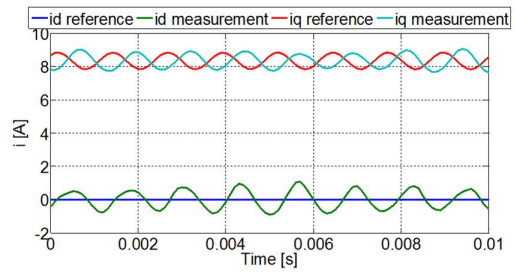
(a) No load, DBCC, N_r^{ref}=2930+30$sin(5000t)$rpm



(b) No load, PI, N_r^{ref}=2930+30$sin(5000t)$rpm



(c) Full load, DBCC, N_r^{ref}=2500rpm, $i_q^{ref} = 8.34 + 0.5sin(5000t)$A



(d) Full load, PI, N_r^{ref}=2500rpm, $i_q^{ref} = 8.34 + 0.5sin(5000t)$A

Figure 4.17: Experimental results of current responses for 5000 rad/s (i_d^{ref}=0A)

For DBCC, it can be seen from Fig.4.16(a) that the first current prediction is able to track the reference after two samples which is the smallest possible delay as discussed

in 4.2.1.1. The phase shift is 115° (=2rad) which is exactly 10000rad/s multiplied by $2T_s$ (=0.0002s). This last result confirms the phase plot in Fig.4.9. Additionally, the attenuation is about -0.55dB. This is reasonably close to the simulated result for 10000rad/s (-0.8dB) as shown in Fig.4.9. Furthermore, the attenuation (-0.25dB) and phase shift (57°) for 5000rad/s in Fig.4.9 is also supported by the experimental results as shown in Fig.4.17(a) and Fig.4.17(c), where the phase lag of the first prediction current equals to 5000rad/s multiplied by $2T_s(= 0.0002s)$, thus 57°, and the attenuation is about -0.61dB.

When using a traditional PI regulator, as can be seen from Fig.4.16(b), Fig.4.17(b) and Fig.4.17(d), the phase lag for 10000rad/s is about 200° (=3.5rad), which can be also represented by a delay time of $350\mu s$ (=3.5rad divided by 10000rad/s); the phase lag for 5000rad/s is about 133° (=2.3rad), which is equivalent to a delay time of $460\mu s$. Moreover, the signal is attenuated by -9.7dB (out of bandwidth) at 10000rad/s and by -1.4dB at 5000rad/s. Although the bandwidth and response can be improved by a better designed PI, the fact that the **delay** of the PI control loop **varies** with frequency cannot be changed.

Analysing the results for DBCC and PI, considering the current reference is likely to be a signal containing more than one frequencies and that the operating point is likely to be within physical limits, in order to reduce torque ripple, **DBCC is preferred for its capability of producing the same delay time for all frequencies.**

4.3 Conclusions

Following the modelling of PMSM in Chapter 3, another precondition for torque ripple reduction is to have a high performance current control loop. This chapter has introduced and evaluated an enhanced deadbeat current control (DBCC) loop. Table 4.2 summarizes the algorithms or equipment that have been included in each simulation/experimental test. Where, the algorithms for “Current Predictions”, “Reference Voltage Calculation”, “Rotor Movement Compensation”, “Inverter Non-linearity Compensation”, and “Steady State Error Compensation” can be found in section 4.1.

Table 4.2: Table of the functionalities implemented in simulation/experimental rig and the corresponding results

Algorithms and Equipment			4.2.1.1 Fig.4.8,4.9 Simulation	4.2.1.2 Fig.4.10 Simulation	4.2.1.2 Fig.4.11 Simulation	4.2.2.1 Fig.4.12 Experiment	4.2.2.2 Fig.4.13ab Simulation
DBCC	Current Predictions	Two steps	√	√	√	√	√
		Traditional				√	
	Reference Voltage Calculation		√	√	√	√	√
	Rotor Movement Compensation	Existing					
		Proposed	√	√	√	√	√
Inverter Non-linearity Compensation						√	
Steady State Error Compensation							
PI Current Controller							
PI Speed Controller							
Inverter	Ideal		√	√			√
	With Non-linearites				√	√	
Surface Mounted PMSM			√	√	√	√	√
Load Motor							
Algorithms and Equipment			4.2.2.2 Fig.4.13cd Experiment	4.2.2.2 Fig.4.14 Simulation	4.2.2.3 Fig.4.15 Simulation	4.2.3 Fig.4.16 Experiment	4.2.3 Fig.4.17 Experiment
DBCC	Current Predictions	Two steps	√	√	√	√	√
		Traditional					
	Reference Voltage Calculation		√	√	√	√	√
	Rotor Movement Compensation	Existing		√			
		Proposed	√	√	√	√	√
Inverter Non-linearity Compensation			√			√	√
Steady State Error Compensation					√		
PI Current Controller						√	√
PI Speed Controller			√			√	√
Inverter	Ideal			√	√		
	With Non-linearities		√			√	√
Surface Mounted PMSM			√	√	√	√	√
Load Motor							√

The main findings acquired from the tests listed in Table 4.2 can be concluded as follow:

1) Comparing with the traditional DBCC, **the proposed method is enhanced with two new functionalities**: the two steps current prediction in stead of the traditional one step to reduce overshoots in current responses during transient (as shown in Fig.4.12); the rotor movement compensation which is more accurate than the existing method in [75] at high speed (as shown in Fig.4.14).

2) **The main pros of DBCC can be concluded as**:

   First, the bandwidth of DBCC when the sampling frequency is 10kHz can reach up to around 3kHz as can be seen from Fig.4.9;

   Second, the bandwidth is mainly affected by parameter L_q (Fig.4.8), however the influence may not be significant in the operating conditions without field weakening since parameter L_q is relatively stable.;

   Third, the delay time of DBCC for different frequencies are the same (i.e. $2T_s$). It barely varies in operating conditions without field weakening as supported by the results in Fig.4.8, Fig.4.16 and Fig.4.17. For this reason, DBCC is also preferred rather than PI current controller from the viewpoint of torque ripple reduction, since delay time or phase shift of current response can have direct impact on the effectiveness of torque ripple suppression.

3) **The main cons of DBCC can be concluded as**:

   The steady sate error of DBCC is strongly affected by parameter F_m (magnetic flux) (Fig.4.10) and the inverter non-linearities (Fig.4.11). The inverter non-linearities can be compensated according to the method in [76] and the results are shown in Fig.4.12. In case of operating under torque (current) control mode in typical servo motor drives, the magnitude of current is important, and it may be necessary to compensate the remaining steady state error by adding a plug-in integrator as shown in Fig.4.5. The results are shown in Fig.4.15. However, under speed or position control mode, the speed or position loop controller may already be enough for compensating the steady state error.

Chapter 5

Torque Ripple Reduction Techniques for PMSMs

We now move to torque ripple reduction techniques for PMSMs. Following the discussions in Chapter 2, ripple can be reduced either by eliminating it from each source (improve machine design, for example), or by adding compensations in the control. The main benefit of the latter option is the possibility of reducing the overall torque ripple from different sources together. For compensating torque ripple in control, the major questions that need to be considered can be concluded as the following: 1) **Where** to add the compensation value in control; 2) **How** to identify the ripple and calculate the compensation value.

For the **first** question, since the output torque of PMSM is normally modelled as a function of current, many existing methods achieve constant torque by introducing current ripple for the cancellation of the original torque ripple. Based on the control structure of the conventional field orientated control (FOC), introducing such current ripple can be implemented either **in the speed loop or in the current loop**. The location of the compensation is discussed in 5.2.1.

For the **second** question, in order to identify the torque ripple, a torque sensor can offer a direct measurement. However, if a torque sensor is absent, the torque ripple

can be found either **online or offline from current and/or position measurements**. Following the literature review in 2.5, three existing torque ripple reduction methods are discussed in details in this Chapter: The first one is the Holtz' compensation method, which is discussed in 5.1.1. The second method is the Choi's compensation scheme, which is discussed in 5.1.2. The third method includes repetitive control, which is discussed in 5.1.3.

Considering the key pros of the aforementioned methods, the combination of repetitive control (RC) and deadbeat current control (DBCC) is proposed in 5.2 where the proposed method is further developed into an angle-based repetitive control for the sake of torque ripple reduction in variable speed drives. Moreover, three aspects of stability enhancement are presented in 5.2.4. Four main innovative aspects in the proposed control scheme are concluded finally in 5.3. The overall proposed control system will be simulated later in Chapter 6.

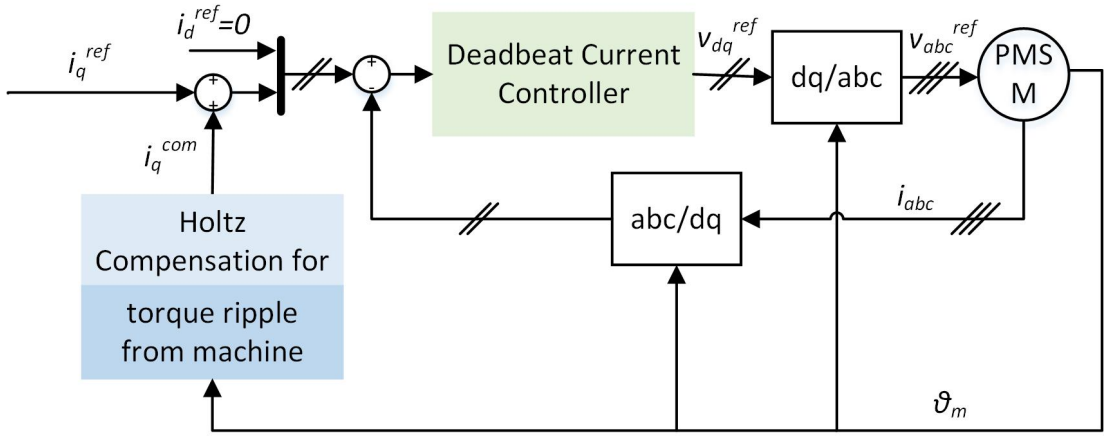
5.1 Existing Techniques

Torque ripple need to be identified before it can be compensated. It is therefore worth taking a closer look at how the ripple can be identified in existing techniques, especially when a torque meter is absent. Three existing compensation techniques are discussed in the following three subsections where the advantages of each method are discussed.

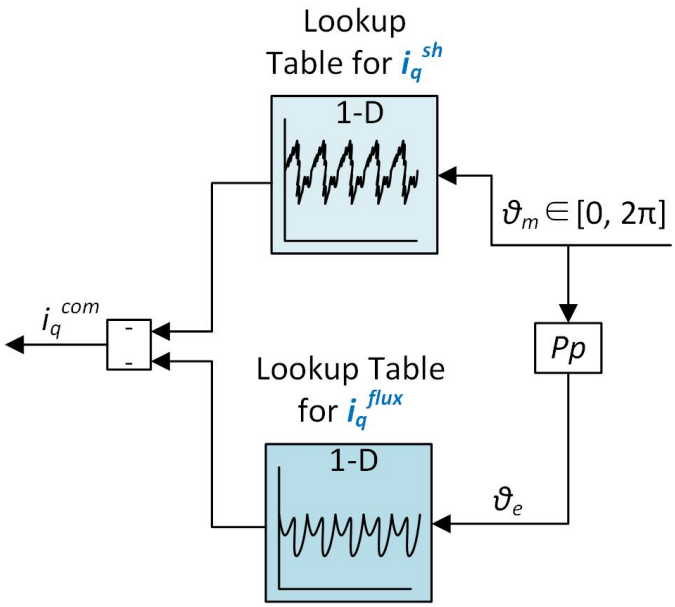
5.1.1 Holtz' Compensation

The method developed by J. Holtz as in [77] and [69] can be a good example for the online learning of the flux harmonics using a machine model, and the offline identification of cogging torque through pre-measurements. A block diagram of the control system in [69] is shown in Fig.5.1. As can be seen, two lookup tables are derived for generating compensation currents for the two types of torque ripple. The compensation current i_q^{sh} is for cancelling the harmonics caused by the slots and teeth, like the cogging torque,

while the compensation current i_q^{flux} targets the harmonics produced by non-sinusoidal flux distribution. The sum of the two compensation currents is fed to the q axis current reference. Deadbeat control is used for the current loop for the sake of high bandwidth.



(a) Overall control system



(b) Holtz compensation block

Figure 5.1: Block diagram for Holtz compensation method

The derivation of the two compensation currents i_q^{sh} and i_q^{flux} are under the assumption that d axis current i_d is always kept zero. In [69], self commission procedures are presented to identify the **flux harmonics online** and the **slot harmonics offline** from experiments; and the compensation currents can be derived accordingly.

It is claimed in the paper that this method contributes the following issues:

+ Through using lookup tables, the torque ripple caused by slot harmonics (cogging effect) is compensated: Some other compensation methods only identify torque ripple through feedback currents and voltages. Since cogging torque is **not** reflected in the induced voltage, such methods are not able to compensate for cogging torque.

+ Online self commissioning of the flux harmonics used in this method is necessary since the back-emf waveforms may differ from machine to machine due to manufacturing tolerances. Also, since permanent magnet characteristics vary with currents and temperature, it is necessary to identify the flux harmonics online.

+ The deadbeat current control is used in order to compensate for high frequency ripple.

Additionally, another advantage of this method can be that:

+ The method works for torque ripple caused by the flux harmonics and the slot harmonics not only at constant speed, but also at variable speed, since the lookup tables are all “angle-based” (rotor position based).

Nevertheless, the main drawbacks of this strategy are:

– The method works only for torque ripple caused by the flux harmonics and the slot harmonics, in other words, the ripple generated by the machine itself. Other torque ripple caused by, for example, mechanical unbalance due to manufacturing tolerance is not considered.

– Online self commissioning of the flux harmonics assumes that the average voltage applied to the motor equals the reference voltage. However, such an assumption may be wrong due to inverter non-linearities. Consequently, the flux harmonics identification may not be accurate.

5.1.2 Choi's Compensation

Choi's compensation scheme refers to the one proposed in [74]. It is an online compensation method in which torque ripple is identified through the rotor position feedback (from where the rotor speed can be calculated). From speed ripple, it is possible to identify the overall torque ripple. Based on a Fourier transformation, any speed waveform can be broken down into a series of sine and cos signals from fundamental frequency to high frequencies. Once the frequency of the ripple is known, the amplitude of ripple at that frequency can be extracted by applying the multiple-angle formulae (5.1).

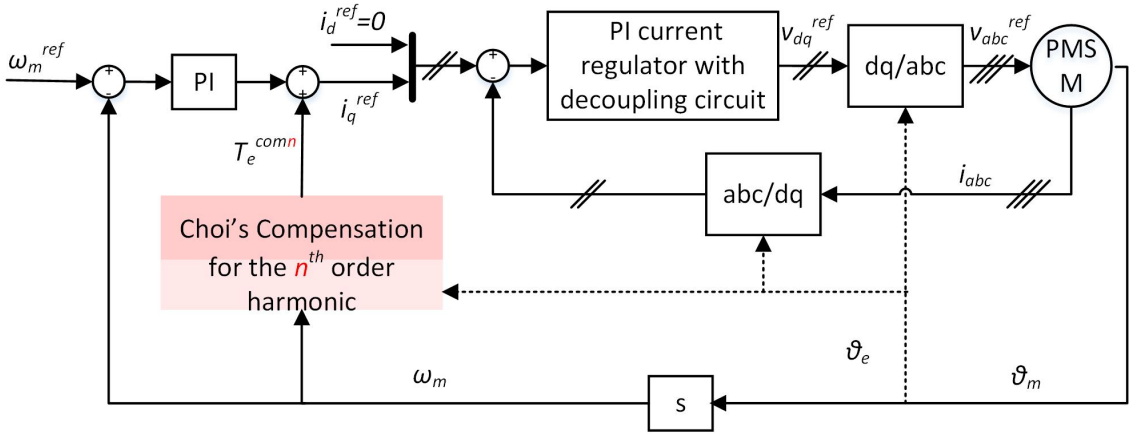
$$\begin{aligned} sin^2\theta &= \frac{1}{2} - \frac{cos(2\theta)}{2} \\ cos^2\theta &= \frac{1}{2} + \frac{cos(2\theta)}{2} \end{aligned} \tag{5.1}$$

For example, the rotor mechanical speed ω_m can be represented as the sum of harmonics from the 1^{st} order to the n^{th}. It can be seen from (5.2) that the amplitude of the n^{th} harmonic becomes a d.c. value after multiplying the speed by $sin(n\theta_e)$ or $cos(n\theta_e)$ and substituting (5.1).

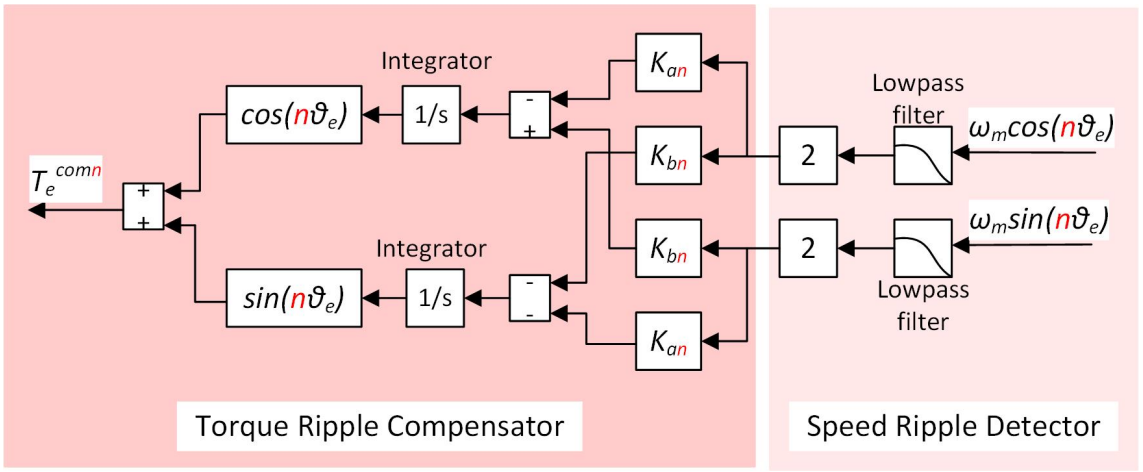
$$\begin{aligned} &\omega_m sin(n\theta_e) = A_1 sin(\theta_e)sin(n\theta_e) + B_1 cos(\theta_e)sin(n\theta_e) + A_2 sin(2\theta_e)sin(n\theta_e) \\ &+ B_2 cos(2\theta_e)sin(n\theta_e) + \cdots + A_n sin^2(n\theta_e) + B_n cos(n\theta_e)sin(n\theta_e) \\ &= \cdots + \frac{A_n}{2} - \frac{A_n}{2}cos(2n\theta_e) + \ldots \\ &\omega_m cos(n\theta_e) = A_1 sin(\theta_e)cos(n\theta_e) + B_1 cos(\theta_e)cos(n\theta_e) + A_2 sin(2\theta_e)cos(n\theta_e) \\ &+ B_2 cos(2\theta_e)cos(n\theta_e) + \cdots + A_n sin(n\theta_e)cos(n\theta_e) + B_n cos^2(n\theta_e) \\ &= \cdots + \frac{B_n}{2} + \frac{B_n}{2}cos(2n\theta_e) + \ldots \end{aligned} \tag{5.2}$$

A block diagram is shown in Fig.5.2, where a compensation block is drawn only for the n^{th} harmonic. **An individual compensation block is required** for compensating ripple of a different order. Correspondingly, the control gains K_{an} and K_{bn} need to be tuned individually. The control gain tuning method has been presented in [74]. Further-

more, as it can be seen from fig.5.2(b), two lowpass filters are used to extract the d.c. amplitude of the n^{th} order harmonic.



(a) Overall control system



(b) Choi's compensation block for the n^{th} harmonics

Figure 5.2: Block diagram for Choi's compensation scheme

Some advantages of the Choi's compensation scheme are:

+ No machine parameters or pre-measurements are required for identifying torque ripple.

+ This strategy can work for reducing the overall torque ripple caused by different sources.

+ Each compensation block works to reduce a fixed order of ripple rather than a fixed frequency of ripple. Therefore, it works also for variable speed. For example, a compensation block has been designed for the 6^{th} order harmonic. Since no matter

how the speed and frequency varies, the 6^{th} order harmonic would always present, this block would work at different speed.

Whereas, drawbacks are:

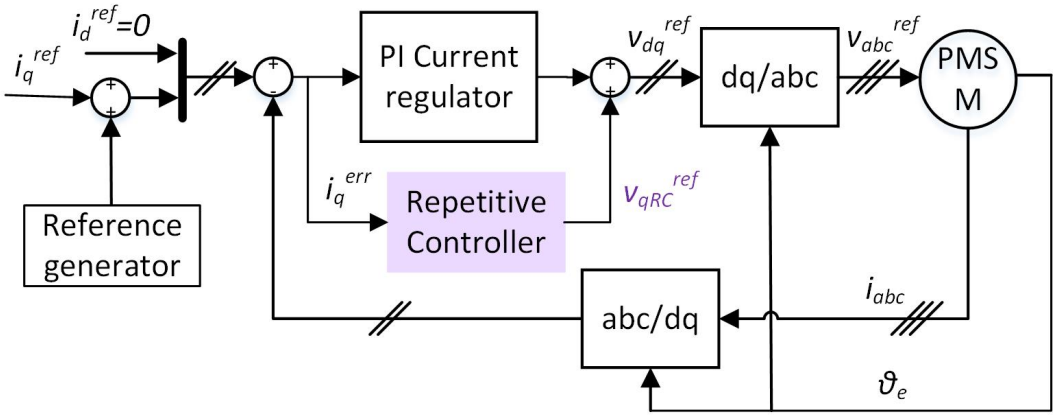
- This strategy works only for selective frequencies. Individual controller need to be developed for each harmonic. So the control would be complex.
- The gains must be manually tuned and their selection is practically dependent on the torque ripple characteristics. So even if machine parameters do not appear directly, the right gains depend on them. If torque ripple is not due to the machine itself, then the tuning must be repeated over time to account for wear, mechanical unbalance etc.
- This strategy works at a relatively low frequency range, limited by the bandwidth of the PI current control loop. High bandwidth current control loop can be used for increasing the frequency range.

5.1.3 Repetitive Control

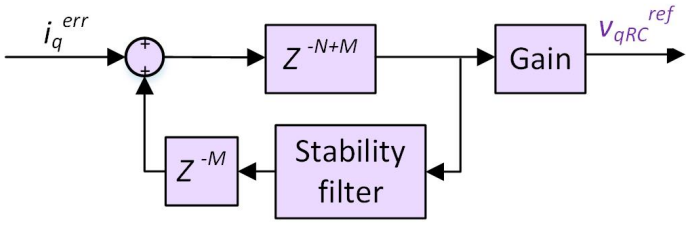
RC is frequently mentioned for torque ripple reduction, since torque ripple is mainly periodic (repetitive). For example, P. Mattavelli has proposed a method using repetitive controller in parallel with conventional PI controller for reducing torque ripple of PMSMs in [65]. The block diagram is as shown in Fig.5.3. External periodic current reference for torque ripple cancellation is yielded by the reference generator.

Due to the limitation of bandwidth, it may be difficult for the PI current regulator to track potential high frequency current references from the reference generator. Therefore, the **principle of repetitive control is to learn** the remaining ripple in current reference that cannot be followed by the PI current regulator, **and to convert** these current tracking errors into the voltage reference as shown in Fig.5.3(a).

The **learning capability** of RC is achieved by iteration through using a delay



(a) Overall control system



(b) Repetitive controller block

Figure 5.3: Block diagram for Mattavelli's repetitive control

chain of N samples as can be seen from Fig.5.3(b). N is chosen to be the closest integer number of samples required for covering the whole period of the fundamental ripple component in i_q^{err}. In other words, NT_s is approximately equal to the period of the fundamental ripple in i_q^{err}. A stability filter is also at the feedback path to remove some unwanted part from the learning.

For **converting** the current tracking error i_q^{err} into the voltage reference V_{qRC}^{ref} as shown in 5.3(b), a **gain** is therefore used for adjusting the amplitude, and the **phase lead compensator** [78], z^M, is also used for adjusting the phase. Since the signal is periodic, delaying the error signal by $N - M$ samples is equivalent to "moving ahead" the signal for M samples.

The phase lead compensator and the stability filter both appear for stability reasons, of which the influences will be discussed later in 5.2.4.

Regarding the advantages of using repetitive control for torque ripple reduction, It can be seen that:

+ No machine parameters or pre-measurements are required for designing the repetitive controller itself. Only foreknowledge of the frequency of torque ripple is required.

+ Comparing to Choi's compensation scheme, repetitive control approach can work for the torque ripple at fundamental frequency and its integer multiple frequencies all-in-one (instead of having individual compensation block for individual harmonics).

Besides, there are also some modifications that can be done for the specific structure in [65] to potentially add more advantages:

√ The approach now requires a current reference generator for generating the periodic current reference for cancelling the torque ripple, and the repetitive controller is responsible for tracking such reference. However, It is possible to **use the repetitive controller to generate such periodic current reference**. Hence, the current reference generator in Fig.5.3(a) or the lookup tables in the Holtz' compensation method in Fig.5.1(a) can be avoided.

√ Consequently, if a repetitive controller is used for generating periodic current reference, a **high bandwidth current controller** like the deadbeat current controller used in the Holtz' compensation method would be required for fast tracking of a high frequency current reference. It is also possible to add an additional RC for high frequency current reference tracking, however, its learning process requires some period and therefore, using another RC would not be a good choice for current control because the resulting response would be slow (compared with the electrical time constants). A deadbeat controller is much faster.

√ An additional benefit of using the repetitive controller for generating the aforementioned current reference according to **the feedback speed signal** is that torque ripple from all sources can be identified and reduced.

√ The repetitive controller in Fig.5.3 is a time-domain controller that is designed for cancelling torque ripple of one fixed frequency (thus fixed speed) and its integer

multiple frequencies. Since the torque ripple is actually periodic with position, by using the **angle-based repetitive control**, it is possible to reduce torque ripple for variable speed. The details will be presented later in 5.2.3.

Overall, inspired by the main advantages of the Holtz' compensation method (5.1.1), the Choi's compensation scheme (5.1.2) and the Mattavelli's repetitive control (5.1.3), the idea is to use repetitive controller for learning the ripple from speed feedback, generate a periodic current reference, while using a high bandwidth deadbeat current controller for tracking the current reference. Therefore, the lookup tables in Holtz' method and the reference generator in Mattavelli's method can be replaced by a repetitive controller. Meanwhile, the frequency range for torque ripple reduction can be expanded due to the high bandwidth current control and if angle-based RC is used.

The control scheme for torque ripple reduction which combines the repetitive control and deadbeat current control will be developed in the following sections.

5.2 Combination of Repetitive Control and Deadbeat Current Control

The inspiration of using the combination of RC and DBCC has been discussed in 5.1.3. As the first step of developing the desired control scheme for torque ripple reduction, possible topologies of the control scheme are discussed in 5.2.1. After the topology has been chosen, the whole control scheme is gradually developed from section 5.2.2 to 5.2.4.

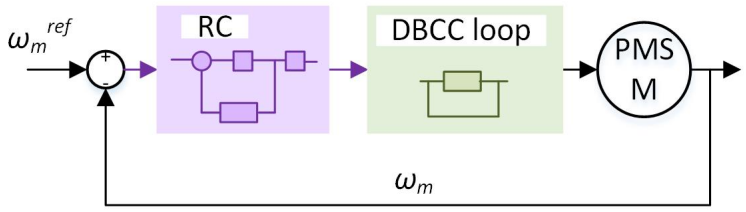
5.2.1 Possible Topologies

A desired topology can be chosen by considering two questions: 1) the possible locations for a repetitive controller in the whole control system; 2) the possible structures of the repetitive controller itself.

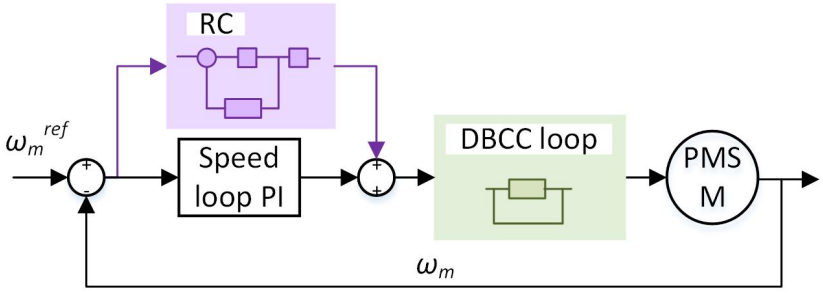
Regarding the first question, following the discussions of using the repetitive controller for generating current reference, it should be located outside of the deadbeat current control loop, thus **in the speed loop**. For answering the second question, three possible block diagrams for the combined repetitive control and deadbeat control system are illustrated in Fig.5.4, considering the three structures [79] (the **direct**, **parallel** and **plug-in** structures) of a repetitive controller available in published literature.

The three structures are all effective for steady state. Therefore, the main consideration for choosing the most suitable structure among the above three is the capability of working during transient. Unfortunately, the repetitive control itself, as it can be seen from Fig.5.3(b), has a delay chain of N samples. Hence, it is naturally more suitable for situations like steady state, when long delay in controller is acceptable. Consequently, it may be better to avoid replacing totally the speed loop PI with the repetitive controller as in Fig.5.4(a). For this reason, the **direct** structure is not preferred. In case of the **parallel** structure as shown in Fig.5.4(b), both the RC and PI would respond to the transient speed error. However, the RC would respond to the speed error after a long delay, whereas the PI would take action immediately, and by the time the RC starts to take action, the PI should already have removed the speed error. Therefore, the RC needs to be disabled during transients, otherwise, it may generate unnecessary actions, and may lead to oscillations. The interaction between RC and PI seems to be the least in case of the **plug-in** structure as shown in Fig.5.4(c), and therefore, may be **the best option** among the three.

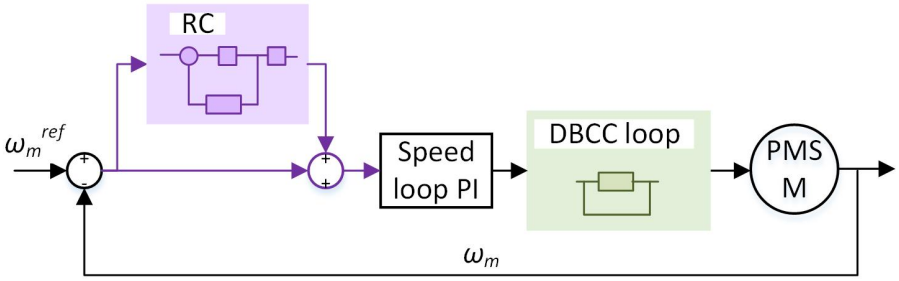
Further considerations are taken in terms of the delays. In Fig.5.4(c), the output of the RC will be delayed by the speed loop PI, the DBCC loop, and the mechanical plant of PMSM, and it would be difficult for the RC to compensate for all these delays. Therefore, the following structure including a **torque estimator** is proposed as shown in Fig.5.5. The equation for torque estimator will be given in 5.2.3. In this way, the overall delay of the mechanical plant of PMSM and the torque estimator can become a fixed time. Also, the delay of the DBCC loop is a fixed time of $2T_s$. Such fixed time delays can then be compensated inside the RC using a simple phase lead compensator z^M. Further discussions will be given in 5.2.2.



(a) Direct structure



(b) Parallel structure



(c) Plug-in structure

Figure 5.4: Three available structures for repetitive controller with deadbeat current control loop

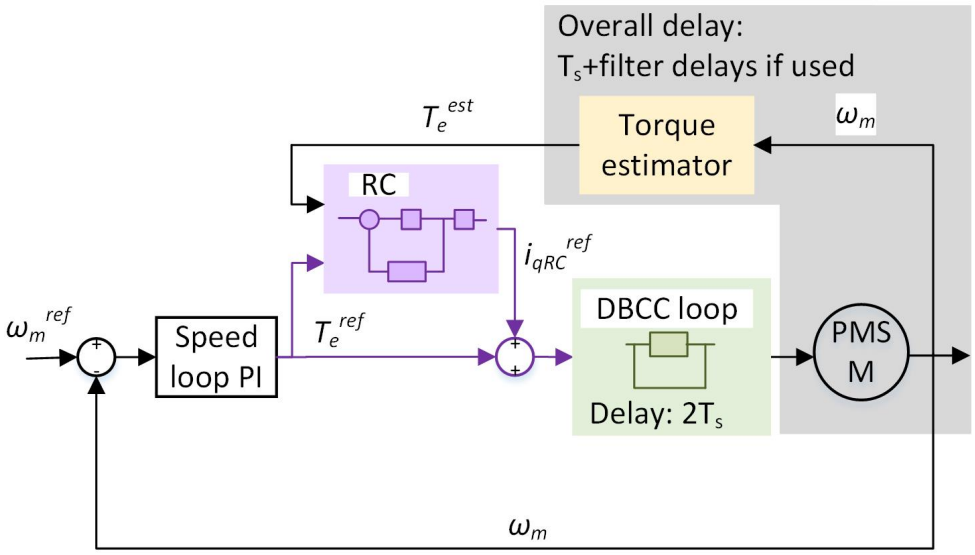


Figure 5.5: The proposed plug-in RC with torque estimator

As may have been noticed, the locations of the plug-in RC and PI are swapped compared to Fig.5.4(c). This is because the speed loop PI, as the outer loop controller, is normally designed to be low in bandwidth, it will be therefore difficult to track the potential high frequency reference generated by the RC.

5.2.2 Time-based Repetitive Control

Based on the conclusions in Chapter 2 and 3, the majority of the ripple torque is periodic with respect to rotor mechanical or electrical position. In either case, the repetitive controller needs to be implemented in the angle-domain for being able to work for variable speed. The principle of RC has been described in 5.1. Before the equations of the angle-based repetitive controller are given in 5.2.3, it is now necessary to look at the equations of the conventional time-based repetitive controller.

Similar to the repetitive controller in Fig.5.3(b) and following the diagram in Fig.5.5, this work considers the repetitive controller with a structure as shown in Fig.5.6. It is worthy to point out that the angle-based and the time-based repetitive controller share exactly the same structure. The time-to-angle and angle-to-time conversion strategies will be presented in 5.2.3.

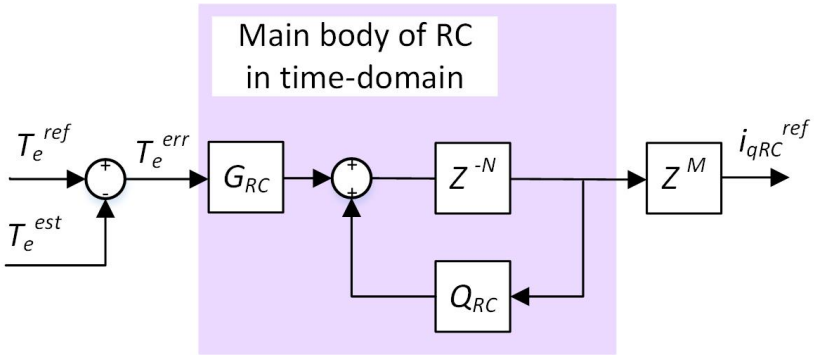


Figure 5.6: Block diagram of the repetitive controller in time-domain

The gain of the repetitive controller is represented by G_{RC}. The forgetting factor (i.e. a constant value within [0, 1]) is represented by Q_{RC}, which is a simple type of stability filter as shown in Fig.5.3(b). N is the closest integer number of the ratio between the sampling frequency f_s and the fundamental frequency of the target ripple torque

(which can be the motor speed in Hz). M is chosen according to the overall delays seen by the repetitive controller. For example, if $i_{qRC}^{ref}(t_k)$ is the output of RC at t_k, then it takes $2T_s$ for the DBCC to finally reach this reference at t_{k+2}, therefore, the torque error at t_{k+2} need to be calculated to know if the compensation is successful. However, this calculation may only be achieved at t_{k+3} or even later due to system delays (delay of the torque estimator or phase shift caused by filters). For instance, the instantaneous torque error at t_{k+2} is only available at t_{k+3}, M should be 3, which is the overall delay samples from the output $i_{qRC}^{ref}(t_k)$ till the torque error acting on this current has been fed back to the repetitive controller. RC can be described using the following equation:

$$i_{qRC}^{ref}(z) = T_e^{err}(z)\frac{G_{RC}z^{-N+M}}{1-Q_{RC}z^{-N}} \tag{5.3}$$

Rearranging (5.3), the following equation can be acquired for calculating the output of repetitive controller at t_k, $i_{qRC}^{ref}(t_k)$.

$$i_{qRC}^{ref}(t_k) = Q_{RC}i_{qRC}^{ref}(t_{k-N}) + G_{RC}T_e^{err}(t_{k-N+M}) \tag{5.4}$$

As can be seen from (5.4), the output of the repetitive controller at t_k depends on the output N samples (one period) ago and the torque error at t_{k-N+M}.

5.2.3 Angle-based Repetitive Control

The angle-based repetitive controller shares the same principle with the time-domain RC, however, the time variables $T_e^{err}(t)$ and $i_{qRC}^{ref}(t)$ are all converted into rotor mechanical position dependent variables $T_e^{err}(\theta_m)$, $i_{qRC}^{ref}(\theta_m)$. Similar to (5.4), the output for when the rotor position is $\theta_m(t_k)$ would depend on the output when the rotor passed the same position in the last period and the remaining ripple torque at this position. In details, the proposed angle-based repetitive controller can be illustrated using Fig.5.7:

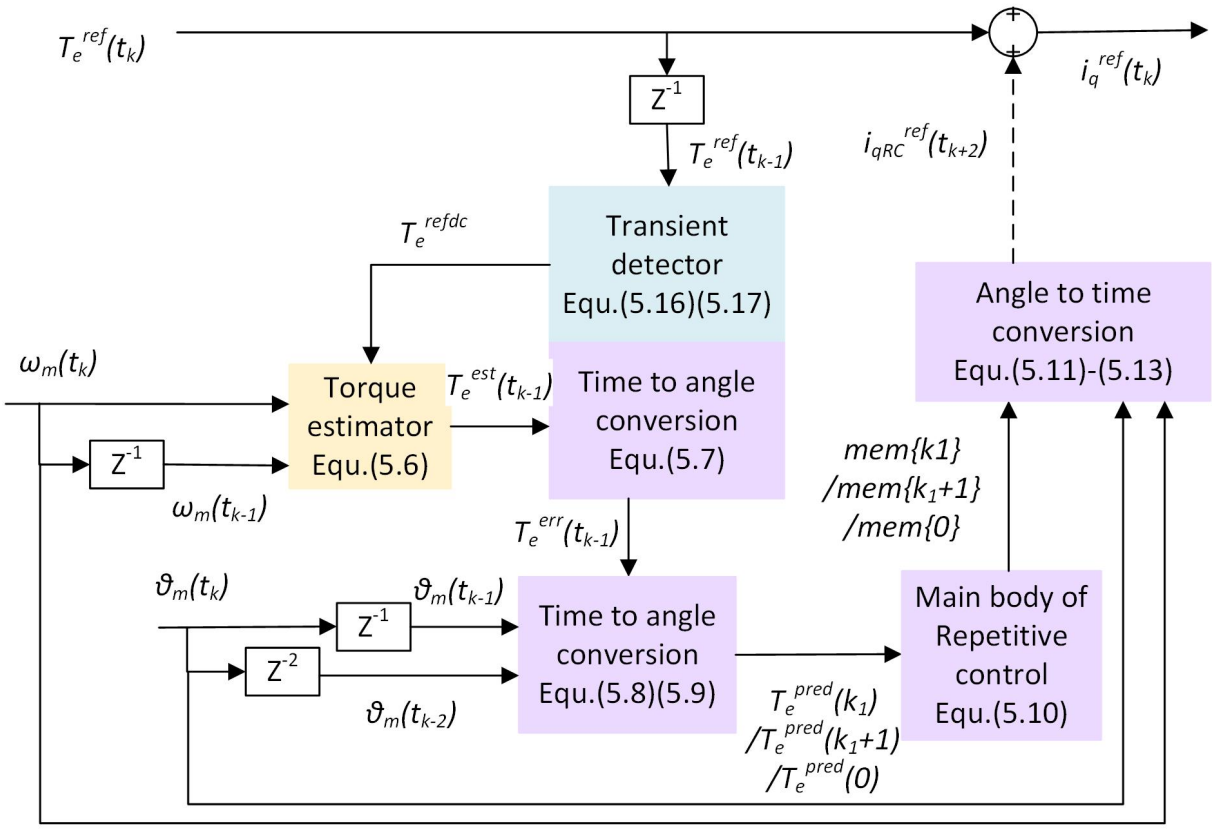


Figure 5.7: Block diagram of the proposed angle-based repetitive control system during the k^{th} sampling period

Each block in Fig.5.7 will be introduced in the following subsections apart from the **transient detector** which will be introduced in 5.2.4. For clarity, **all the equations given are with respect to the calculations carried out in the k^{th} sampling period.**

5.2.3.1 Torque Estimator

The discretized mechanical plant of the motor can be described as in (5.5). The forward rectangular method is used for the discretization. Under this method, the continuous system will be stable as long as the discrete system is stable [80].

$$G_{sp}(z) = \frac{T_s}{Jz - (J - BT_s)} = \frac{\omega_m(z)}{T_e(z) - T_l} \tag{5.5}$$

Where, T_s is the sampling period, J is the motor and load inertia, B is friction

factor, and T_l is the load torque.

Assuming pre-knowledge of the mechanical parameters (inertia J^{est} and friction factor B^{est}), the electrical torque can be estimated during the k^{th} sampling interval based on the reverse of (5.5) as follows:

$$T_e^{est}(t_{k-1}) = \frac{J^{est}[\omega_m(t_k) - \omega_m(t_{k-1})]}{T_s} + B^{est}\omega_m(t_{k-1}) + T_e^{refdc} \tag{5.6}$$

$T_e^{est}(t_{k-1})$ is the estimated electrical torque at t_{k-1}. As it can be seen, the estimated torque available during the k^{th} sampling period is $T_e^{est}(t_{k-1})$, **not** $T_e^{est}(t_k)$. This is because **a delay** of T_s is introduced by this torque estimator. $\omega_m(t_k)$ is the rotor mechanical angular speed at t_k. The load torque is approximated using the d.c. value of the torque reference T_e^{refdc} from the output of the speed loop PI controller as shown in Fig.5.5. This load torque approximation is achieved by using the proposed transient detector, and will be introduced later in 5.2.4.3.

The influence of detuned J^{est} and B^{est} will be investigated later in Section 6.3.2. An enhanced torque estimator with FIR filters will also be proposed for the experimental tests later in Section 7.2.

5.2.3.2 Time to Angle Conversion

Since $T_e^{est}(t_k)$ is available no earlier than the $(k+1)^{th}$ sampling period, the calculations taking place during the k^{th} sampling period are with $T_e^{est}(t_{k-1})$. Although the output of speed loop PI controller, $T_e^{ref}(t_k)$ and rotor position feedback from encoder $\theta_m(t_k)$ are already available, $T_e^{ref}(t_{k-1})$ and $\theta_m(t_{k-1})$ are used for 5.7 and 5.8 in order to synchronize with $T_e^{est}(t_{k-1})$.

First of all, the torque error T_e^{err} at t_{k-1} is calculated as in (5.7):

$$T_e^{err}(t_{k-1}) = T_e^{ref}(t_{k-1}) - T_e^{est}(t_{k-1}) \tag{5.7}$$

The time to angle conversion is then executed as follow. An array of length N, named mem, is defined in the memory. The values in the array mem are updated using the torque errors at only N chosen rotor locations out of the whole mechanical resolution (2π). The mechanical revolution is chosen in place of the electrical one in order to be able to compensate torque ripple generated by also mechanical problems. Each of the N values indicates the q axis current required for torque ripple reduction at a specific position, corresponding to one of the chosen rotor locations. For example, as shown in Fig.5.8, the $(i+1)^{th}$ value in the memory array, $mem\{i\}$, is the q axis current required for reducing torque error when the rotor is **exactly** at position $2\pi i/N$. The **rotor location index** k_1 is defined as in (5.8):

$$k_1 = floor\Big[\frac{N\theta_m(t_{k-1})}{2\pi}\Big], k_1 \in [0, N-1] \tag{5.8}$$

where, the value of N is chosen to be the closest integer to the ratio between the sampling frequency f_s and the **rated** rotor speed ω_m in Hz.

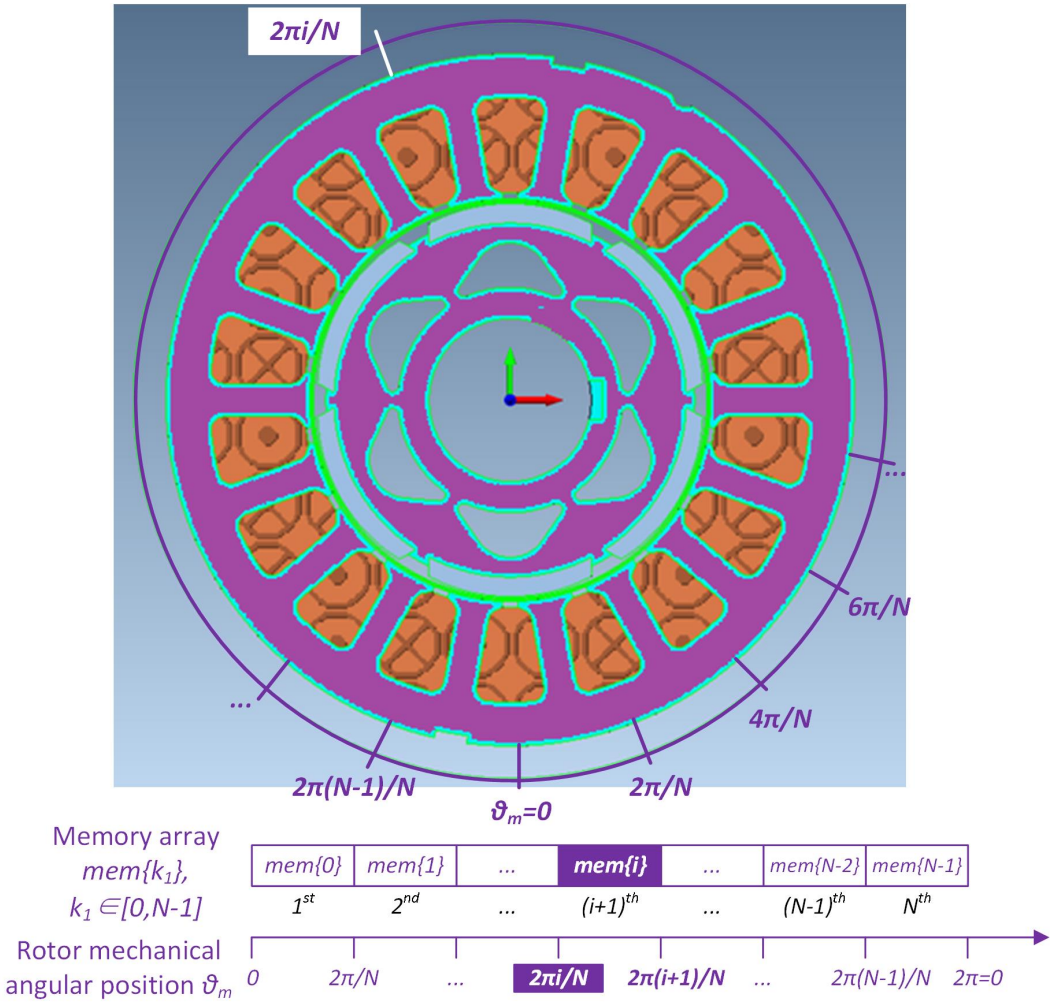


Figure 5.8: The relationship between the rotor location and memory location

Since the rotor position cannot always be measured **exactly** at points coincident with integer multiples of $2\pi/N$, interpolations are necessarily used to predict the torque error at the closest chosen rotor locations ($\theta_m = 0, 2\pi/N, 4\pi/N, ..., 2\pi$). Linear interpolations are executed as in (5.9) **only if** the value of k_1 is different from the one obtained at the previous sampling period. It is worth noting that when the direction of the rotor movement is opposite to the positive direction of the encoder, thus negative rotation, $T_e^{pred}(k_1 + 1)$ is calculated instead of $T_e^{pred}(k_1)$. This is because it is more accurate to estimate the torque error at $\theta_m(k_1 + 1) = 2\pi(k_1 + 1)/N$ (which is in between the position $\theta_m(t_{k-1})$ and its previous position $\theta_m(t_{k-2})$) than to estimate the torque error at $\theta_m(k_1) = 2\pi(k_1)/N$ which has not yet been "reached" (Fig.5.9).

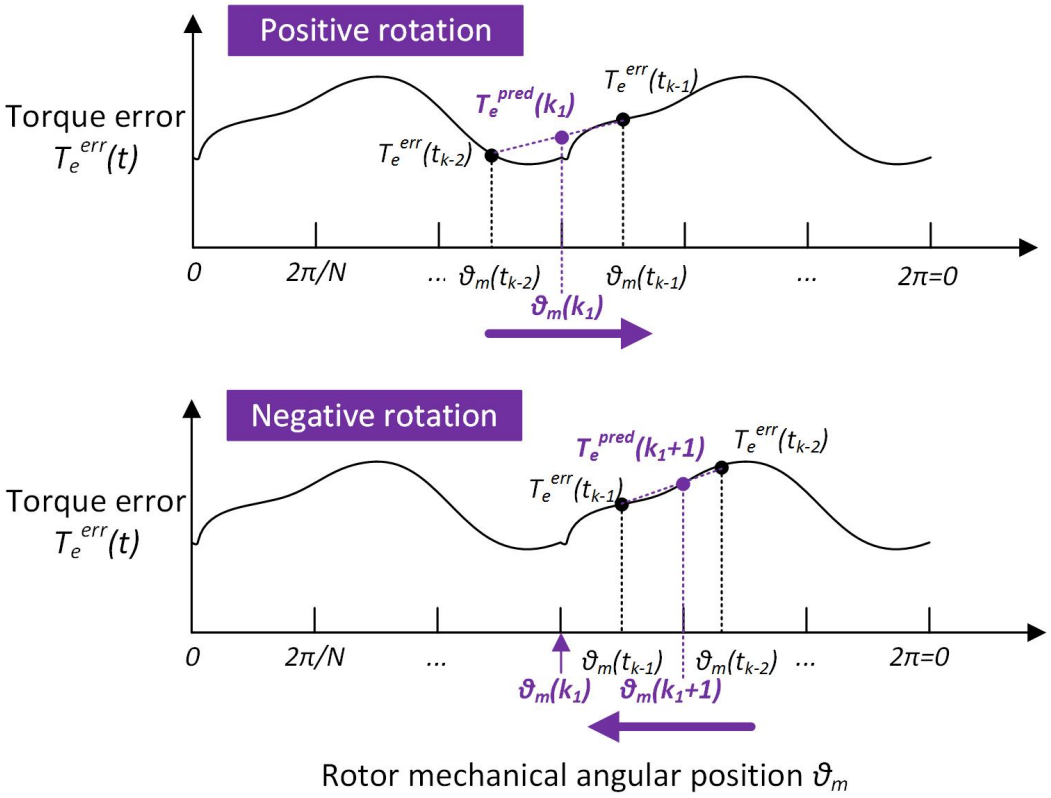


Figure 5.9: Principle of the interpolations in the proposed time to angle conversion

In the case of **positive** rotation:

$$T_e^{pred}(k_1) = T_e^{err}(t_{k-2}) + \frac{T_e^{err}(t_{k-1}) - T_e^{err}(t_{k-2})}{\theta_m(t_{k-1}) - \theta_m(t_{k-2})}[\theta_m(k_1) - \theta_m(t_{k-2})]$$

In the case of **negative** rotation, $k_1 \neq N-1$:

$$T_e^{pred}(k_1+1) = T_e^{err}(t_{k-2}) + \frac{T_e^{err}(t_{k-1}) - T_e^{err}(t_{k-2})}{\theta_m(t_{k-1}) - \theta_m(t_{k-2})}[\theta_m(k_1+1) - \theta_m(t_{k-2})] \tag{5.9}$$

In the case of **negative** rotation, $k_1 = N-1$:

$$T_e^{pred}(0) = T_e^{err}(t_{k-2}) + \frac{T_e^{err}(t_{k-1}) - T_e^{err}(t_{k-2})}{\theta_m(t_{k-1}) - \theta_m(t_{k-2})}[0 - \theta_m(t_{k-2})]$$

For clarity, $T_e^{pred}(k_1)$ is the torque error if the rotor mechanical angular position equals to $\theta_m(k_1) = 2\pi k_1/N$. $T_e^{err}(t_{k-1})$ and $T_e^{err}(t_{k-2})$ are the time-domain torque errors at t_{k-1} and t_{k-2}. $\theta_m(t_{k-1})$ and $\theta_m(t_{k-2})$ are the rotor angular position at t_{k-1} and t_{k-2}.

5.2.3.3 Main Body of Repetitive Control

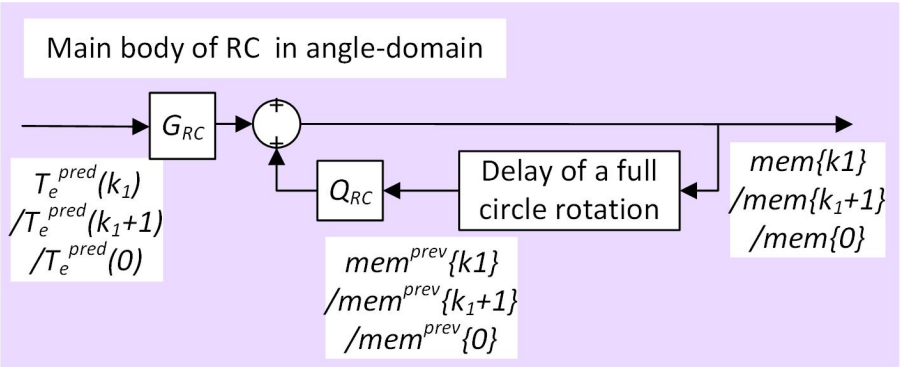


Figure 5.10: Block diagram of the main body of RC in angle-domain

In angle-domain, the main body of RC is as shown in Fig.5.10, whose main difference, when compared to the time-domain one in Fig.5.6, is the delay of N samples has been converted into the delay of a full mechanical cycle rotation. This is achieved through updating the memory array according to the rotor position as follow.

Again, during the k^{th} sampling period, after converting the time domain torque errors $T_e^{err}(t_{k-1})$ and $T_e^{err}(t_{k-2})$ into the angle-domain torque errors $T_e^{pred}(k_1)$, $T_e^{pred}(k_1+1)$ or $T_e^{pred}(0)$ depending on the moving direction, the memory array can now be updated according to the repetitive control equations (5.10). Again, **only when** the rotor location index k_1 is different from the one at the previous sampling period (i.e. the movement is larger than $2\pi/N$), would the memory array need to be updated.

$$
\begin{aligned}
&\text{In the case of } \textbf{positive} \text{ rotation:} \\
&mem\{k_1\} = Q_{RC} \cdot mem^{prev}\{k_1\} + G_{RC} \cdot T_e^{pred}(k_1) \\
&\text{In the case of } \textbf{negative} \text{ rotation}, k_1 \neq N-1: \\
&mem\{k_1+1\} = Q_{RC} \cdot mem^{prev}\{k_1+1\} + G_{RC} \cdot T_e^{pred}(k_1+1) \\
&\text{In the case of } \textbf{negative} \text{ rotation}, k_1 = N-1: \\
&mem\{0\} = Q_{RC} \cdot mem^{prev}\{0\} + G_{RC} \cdot T_e^{pred}(0)
\end{aligned}
\tag{5.10}
$$

Again, Q_{RC} is the forgetting factor of RC, G_{RC} is the gain of RC. $mem\{k_1\}$ is the q axis current for compensating torque ripple at exactly $\theta_m = 2\pi k_1/N$. $mem^{prev}\{k_1\}$ is

the previous value of $mem\{k_1\}$ obtained at the time of one full cycle rotation ago.

5.2.3.4 Angle to Time Conversion

As shown in Fig.5.7, the output of the angle to time conversion acts as an additional q axis current reference, i_{qRC}^{ref}. Therefore, the main functionality of the angle to time conversion block is to select the best i_{qRC}^{ref} from the memory array during the k^{th} sampling period. For this purpose, the following two steps are followed:

Step 1: The current reference i_{qRC}^{ref} is sent to the current loop during the k^{th} sampling period. However, there will be a delay for this reference current to be effectively reached in the motor. Therefore, **delays of the current control loop need to be considered when selecting the correct current reference from the memory array.** For example, in case of the deadbeat current control loop, its delay under normal operating condition is $2T_s$ as concluded in 4.3. This $2T_s$ includes one T_s of computation phase, and one T_s for the current to reach its reference. In another word, the current reference computed during the k^{th} sampling interval will be achieved at t_{k+2}. Hence, for choosing the correct current reference from mem, the future rotor position at t_{k+2} need to be predicted as in (5.11) (assuming the rotor speed is constant from t_k to t_{k+2}):

$$\theta_m^{fut}(t_{k+2}) = \theta_m(t_k) + 2T_s\omega_m(t_k), \theta_m^{fut} \in [0, 2\pi] \tag{5.11}$$

Accordingly, the **future rotor location index** k_2 is defined as in (5.12):

$$k_2 = floor\Big[\frac{N\theta_m^{fut}(t_{k+2})}{2\pi}\Big], k_2 \in [0, N-1] \tag{5.12}$$

Step 2: The current reference can then be selected from the memory array according to the future rotor location index k_2. Similar to the time to angle conversion, **linear interpolations need to be used** for estimating the value between $mem\{k_2\}$ and $mem\{k_2+1\}$ as in (5.13) and Fig.5.11:

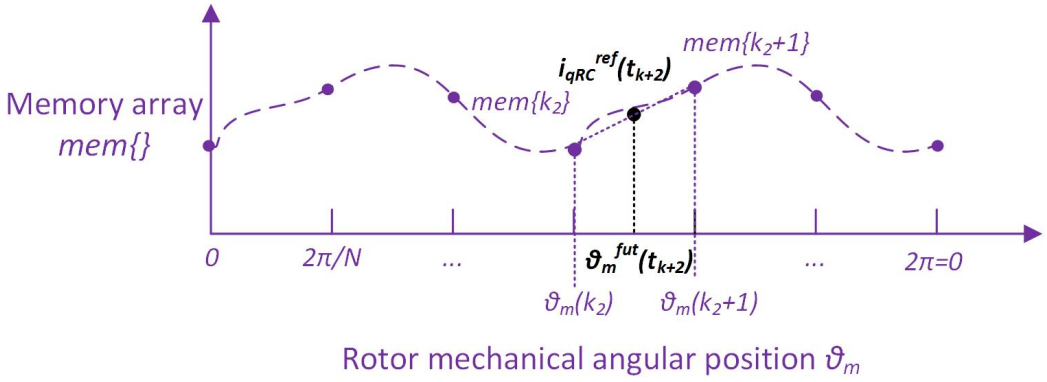


Figure 5.11: Principle of the interpolations in the proposed angle to time conversion

$$\begin{aligned}
&\text{When } k_2 \neq N-1: \\
&i_{qRC}^{ref}(t_{k+2}) = mem\{k_2\} + \frac{N(mem\{k_2+1\} - mem\{k_2\})}{2\pi}\left[\theta_m^{fut}(t_{k+2}) - \frac{2\pi k_2}{N}\right] \\
&\text{When } k_2 = N-1: \\
&i_{qRC}^{ref}(t_{k+2}) = mem\{k_2\} + \frac{N(mem\{0\} - mem\{k_2\})}{2\pi}\left[\theta_m^{fut}(t_{k+2}) - \frac{2\pi k_2}{N}\right]
\end{aligned} \tag{5.13}$$

It can be noticed that through the above two steps, the **phase lead compensator** z^M of the time-domain RC in 5.2.2 has been effectively **implemented in the angle-domain**. It is also worth noting that in time-domain, as presented in 5.2.2, M is chosen according to the overall delay seen by the repetitive controller, including both the current control loop delay and the delay of mechanical system and the torque estimator. However, the position prediction here only compensates for the current loop delay. This is because the delay of the mechanical system, torque estimator and additional filters if used can all be compensated by synchronizing T_e^{err} and θ_m in (5.7) and (5.8).

For example, in case an additional n^{th} order FIR filter is used in the torque estimator (i.e. the overall delay of mechanical system and torque estimator is (n+1)T_s), therefore the estimated torque available during the k^{th} sampling period is $T_e^{est}(t_{k-(n+1)})$, while the newest position feedback during which time is $\theta_m(t_k)$. Obviously, the position feedback and estimated torque is not synchronized. This issue can easily be solved by memorising $\theta_m(t_{k-(n+1)})$ and making sure the correct position is used in (5.7) and (5.8).

5.2.4 Stability Enhancement

Repetitive control cancels periodic torque ripple by generating a compensating current that consequently need to be synchronized with the original ripple. Failure of synchronization can degrade the performance of ripple reduction, and in the worst case, lead to instability.

For the purpose of enhancing the stability, the proposed control method includes **three aspects of enhancements**: 1) the **forgetting factor** as shown in Fig.5.10; 2) the **phase lead compensator** which has been implemented in the angle to time conversion (5.2.3.4); 3) the **transient detector** as shown in Fig.5.7. The function of each enhancement will be introduced in the following subsections.

First of all, summarizing the design of RC in previous sections, the proposed control system is as shown in Fig.5.12, in which the angle-based RC is drawn in Fig.5.7. Neglecting the transient detector, the equivalent diagrams are as shown in Fig.5.13.

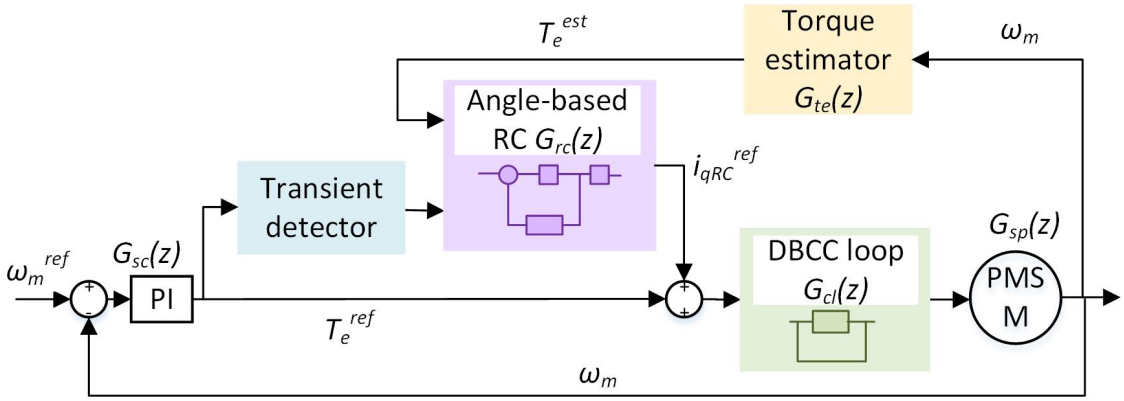
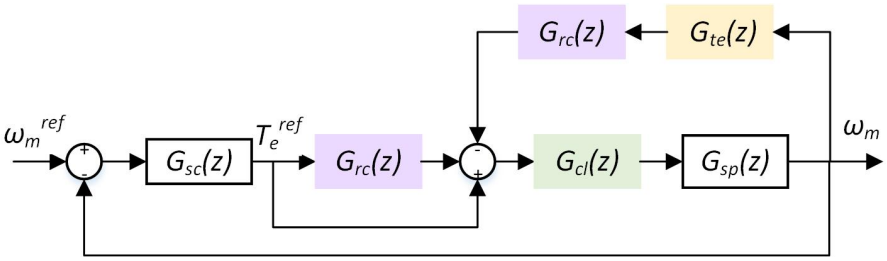
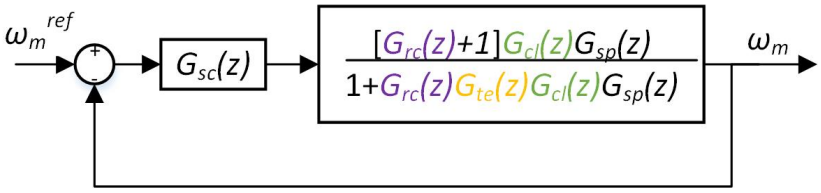


Figure 5.12: Block diagram of the proposed control system



(a) Equivalent to Fig.5.12



(b) Equivalent to (a)

Figure 5.13: Equivalent diagram of the proposed control system

The open-loop and closed-loop transfer function of the proposed control system can be derived as (5.14) and (5.15) respectively.

$$G_{open}(z) = \frac{G_{sc}(z)[G_{rc}(z)+1]G_{cl}(z)G_{sp}(z)}{1+G_{rc}(z)G_{te}(z)G_{cl}(z)G_{sp}(z)} \tag{5.14}$$

$$\begin{aligned} &G_{closed}(z) \\ &= \frac{\omega_m(z)}{\omega_m^{ref}(z)} = \frac{\frac{G_{sc}(z)[G_{rc}(z)+1]G_{cl}(z)G_{sp}(z)}{1+G_{rc}(z)G_{te}(z)G_{cl}(z)G_{sp}(z)}}{1+\frac{G_{sc}(z)[G_{rc}(z)+1]G_{cl}(z)G_{sp}(z)}{1+G_{rc}(z)G_{te}(z)G_{cl}(z)G_{sp}(z)}} \\ &= \frac{G_{sc}(z)[G_{rc}(z)+1]G_{cl}(z)G_{sp}(z)}{1+G_{sc}(z)[G_{rc}(z)+1]G_{cl}(z)G_{sp}(z)+G_{rc}(z)G_{te}(z)G_{cl}(z)G_{sp}(z)} \end{aligned} \tag{5.15}$$

where:

- If the current control loop is working within the physical limit, it follows the conclusion of Chapter 4 that the deadbeat current control loop can be represented by a delay of two sampling periods, $\mathbf{G_{cl}(z) = z^{-2}}$.
- The transfer function of the torque estimator $G_{te}(z)$ can be derived from (5.6), $G_{te}(z) = \frac{J^{est}z-(J^{est}-B^{est}T_s)}{T_s z}$. The speed plant of the PMSM is as (5.5), $G_{sp}(z) = \frac{T_s}{Jz-(J-BT_s)}$. It can be seen that $\mathbf{G_{te}(z)G_{sp}(z) = z^{-1}}$ assuming the estimated inertia J^{est} and friction factor B^{est} equal to the real inertia J and friction factor B in the mechanical model of the PMSM.

- The angle-based repetitive controller is effectively the time-based repetitive control converted into the angle-domain. Hence, the same transfer function can be used for both of them. As can be seen from Fig.5.6 and (5.3), $\mathbf{G_{rc}(z) = \frac{G_{RC}z^{-N+M}}{1-Q_{RC}z^{-N}}}$.

- The speed loop controller $G_{sc}(z)$ is designed for a speed loop bandwidth of 50Hz. For example, assuming $J^{est} = J = 0.0004\text{kg/m}^2$ and $B^{est} = B = 0.001\text{Nms/rad}$, $\mathbf{G_{sc} = \frac{0.1052z-0.1041}{z-1}}$ (damping factor = 0.8).

In the following subsections, the enhancement of stability can be validated using the pole-zero plot of the closed loop system G_{closed} with different setting of the parameter Q_{RC} for the stability filter and M for the phase lead compensator. The results indicate the necessity of using the proposed transient detector.

It is worth pointing out that since the proposed control system combines both angle-domain and time-domain controllers, it would be challenging to derive one universal transfer function of it for all conditions. However, the angle-domain RC can be understood as a time-domain RC with variable sampling frequency. Therefore, in the following analysis, the angle-based RC is represented by the transfer function of the time-domain RC, and **it is believed that converting the time-domain RC into angle-domain would not change the effect of forgetting factor Q_{RC} and phase lead compensator z^M on stability.**

5.2.4.1 Influence of Forgetting Factor Q_{RC} on Stability

Forgetting factor is a simple type of stability filter shown in Fig.5.3(b). Fig.5.14 shows how the pole-zero locations of the closed loop transfer function $G_{closed}(z)$ vary with forgetting factor Q_{RC} varying from 0.01 to 0.99. The control gain G_{RC} is kept at one for this test. N is set to be 20, M is set to be 3.

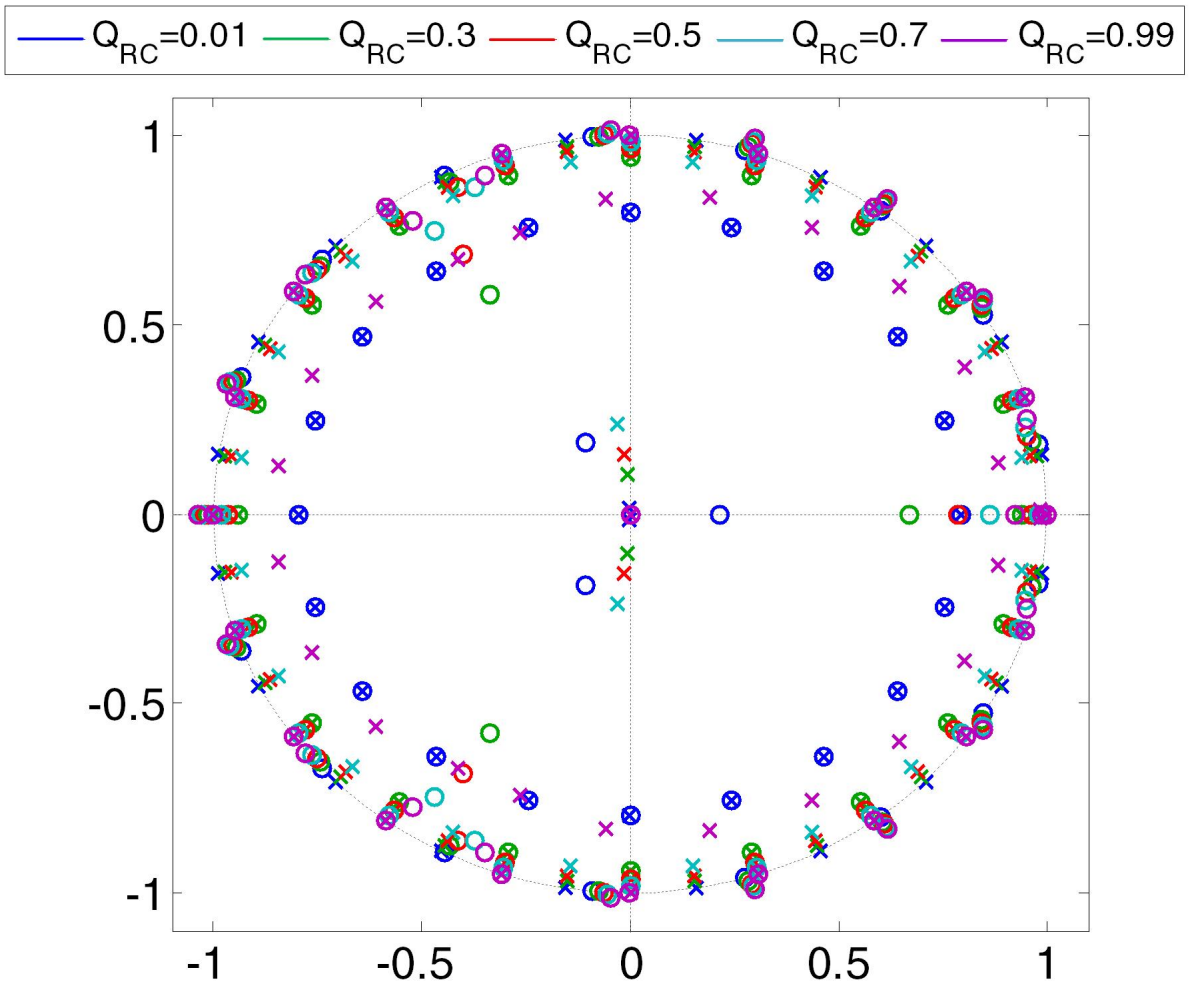


Figure 5.14: Pole-zero plot of closed loop system with forgetting factor Q_{RC} varying from 0.01 to 0.99

It can be seen that by properly choosing Q_{RC}, the closed loop poles and zeros can be located more inside the unity circle. Further discussions about choosing the suitable Q_{RC} will be given in Chapter 6 referring to the simulation tests results. Generally, Q_{RC} is not only related to stability, but also strongly related to the effectiveness of ripple reduction. Therefore, it is **not sufficient** to ensure the stability only by tuning the forget factor Q_{RC}.

5.2.4.2 Influence of Phase Lead Compensator z^M on Stability

The stability of the proposed RC control system is in fact strongly affected by the delays in the system. As shown in Fig.5.15, some closed loop poles (the green case) moved outside

of the unity circle, when the delay T_d of the current loop increases for **only** one more sampling period ($\mathbf{G_{cl}(z) = z^{-3}}$ instead of $\mathbf{G_{cl}(z) = z^{-2}}$). The phase lead compensator is proposed for compensating the system delays. As can be seen that by increasing M from 3 to 4. The system becomes stable again. ($G_{RC} = 1$, $Q_{RC} = 0.5$)

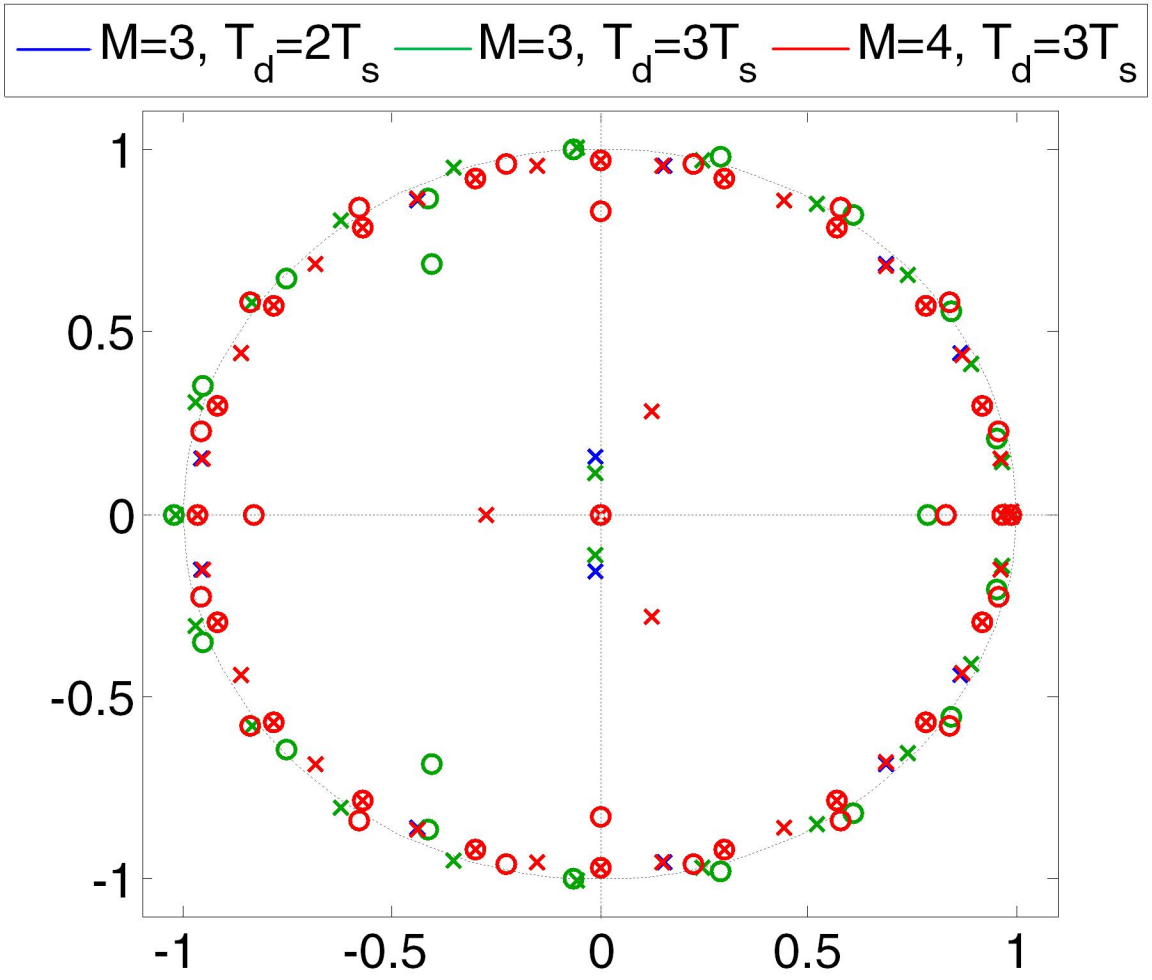


Figure 5.15: Pole-zero plot of closed loop system with the delay of current loop T_d increased from $2T_s$ to $3T_s$

In the proposed angle-based RC control system, The phase lead compensator **has been implemented in the angle to time conversion** (refer to section 5.2.3.4) by predicting the future rotor position, and interpolating the best compensation current reference according to this future rotor position. However, this compensation can **not** ensure the stability once the system delay has changed as normally happens during large current transients when the reference current cannot be reached anymore in two sampling periods. Therefore, it is also necessary to, on one hand, **limit the output of RC** in

order to prevent the RC from generating exceedingly wrong current references, and on the other hand, to block the input of RC if the torque error is too large for any reasons. These two functions can be realized by using the transient detector as proposed below.

5.2.4.3 Transient Detector

The main goal of using a transient detector is to allow torque ripple reduction even during speed and load transients. In fact, the learning process, after a transient, only needs to be updated with a small amplitude change in the torque ripple due to the changed conditions (saturation of the machine for example), whereas the frequency and phase of the torque ripple remain unchanged since related to the rotor position. However, the learning process has to be temporarily disabled during each transient, to avoid the RC outputting an excessive compensating reference current and cause the oscillations as a consequence of the non-existent large torque ripple estimated during a sudden change in speed. Therefore, a transient detection routine is designed to detect speed/load changes and suspend temporarily the learning process.

In details, speed change leads to a sudden change in the torque reference T_e^{ref} generated by the speed loop PI controller. Therefore, a speed transient can be identified if T_e^{ref} varies more than a chosen **threshold** δT_e^{max}, comparing with its value at the previous sampling period. As in (5.16), the speed transient signal S_{speed}^{ena} is set to be zero once a speed transient is detected.

$$\begin{aligned} &\text{If } |T_e^{ref}(t_k) - T_e^{ref}(t_{k-1})| > \delta T_e^{max} : \\ &S_{speed} = 0 \\ &\text{Otherwise: } S_{speed} = 1 \end{aligned} \tag{5.16}$$

Similarly, since load change leads to a gradual change in the torque reference T_e^{ref}, a load change can be identified if T_e^{ref} varies more than the same **threshold** δT_e^{max}, comparing with its value a certain time ago. The certain time is named as **inspection duration** δt in this thesis. The load transient signal S_{load} is set to be zero once a load

transient is detected, as in (5.17).

$$\begin{aligned} &\text{If } |T_e^{ref}(t_k) - T_e^{ref}(t_k - \delta t)| > \delta T_e^{max} : \\ &S_{load} = 0 \\ &\text{Otherwise: } S_{load} = 1 \end{aligned} \tag{5.17}$$

The system is recognized as in steady state only if both the transient signals S_{speed} and S_{load} have been maintained to be one for a certain period. This certain period is named as **minimum steady state time** t^{ste} in this thesis. S^{ena} is the enable signal, S^{ena}=0 indicates the system to be in transient, and S^{ena}=1 indicates the system to be at steady state. Its update law can be seen as in (5.18).

$$\begin{aligned} &\text{If } S_{speed} = S_{load} = 1 \text{ is true for } t^{ste}, \text{ then } : S^{ena} = 1 \\ &\text{Otherwise: } S^{ena} = 0 \end{aligned} \tag{5.18}$$

The enable signal is used for the following **three functionalities**:

Function 1: **Disable the learning process of RC during transients**

Only during steady state the torque error T_e^{err} equals to $T_e^{ref}(t_k) - T_e^{est}(t_k)$ (as in (5.7)), otherwise, torque error T_e^{err} is set to be zero (5.19) so that the memory array becomes independent from the input, and consequently its output would only depend on previously memorized values. Hence, the learning process is effectively disabled, while the compensation is still active.

$$\begin{aligned} &\text{If } S^{ena} = 1 \text{ is true for } t^{ste}, \text{ then } : T_e^{err}(t_k) = T_e^{ref}(t_k) - T_e^{est}(t_k) \\ &\text{Otherwise: } T_e^{err}(t_k) = 0 \end{aligned} \tag{5.19}$$

Function 2: **Approximate the load torque**

As in (5.6), torque estimation requires knowledge of the load torque. The load torque can be recognized as the d.c. value of the torque reference, which can be approximated using the instantaneous torque reference at the moment when steady state starts. Therefore, the T_e^{refdc} in (5.6) is updated as in (5.20) once a new steady state has

been reached. S_{prev}^{ena} is the value of S^{ena} in the previous sampling period. It is worth pointing out that a small offset error would present in the estimated torque due to this implementation. This offset can be compensated by the Function 3 below.

$$\begin{aligned} &\text{If } S^{ena} = 1 \text{ and } S_{prev}^{ena} = 0 : T_e^{refdc} = T_e^{ref}(t_k) \\ &\text{Otherwise, do not update } T_e^{refdc} \end{aligned} \tag{5.20}$$

Function 3: **Limit the output of RC**

As discussed in the previous subsection, the output of the RC needs to be limited to prevent it from sending any exceedingly wrong current references to the current controller. It can be limited from two aspects: 1) the output of RC should be within reasonable limits at all times; 2) ideally, RC is responsible for generating any a.c. part of the current reference, whereas the speed loop PI controller takes care of the d.c. part. However, due to the offset error between the estimated torque and the real electrical torque of the machine, RC would also contain a small d.c. offset. In fact, this small d.c. offset will barely affect the performance. However, if the load torque varies, the d.c. offset may accumulate, and eventually reach the limit of the reasonable value set in 1). For safety reasons, this offset is removed from the RC memory once a new transient starts.

5.3 Conclusions

This chapter focused on the development of a desired control scheme for torque ripple reduction not only at steady state, but also during transients. A map has been drawn in Fig.5.16 to shown the key points of how the proposed scheme are developed. **Four main innovative aspects** of the proposed method can be concluded as the following:

1) Taking benefit from the use of a **deadbeat current control** loop (which has been verified in Chapter 5 that its delay is fixed to be $2T_s$ under normal operating conditions), it is possible to simply use a phase lead compensator (5.2.4.2) to **compensate the delay of the current loop**. This chapter has implemented such a

time-domain compensator in the angle-domain by proposing the **future position prediction** as in 5.2.3.4.

2) Due to the low-pass filter effect of the mechanical system, high frequency components in the machine output torque would be delayed more than the low frequency components. Since a simple phase lead compensator is not able to compensate such "frequency-varying" delays, it is better to compensate this delay outside the RC. The **torque estimator** (5.2.3.1) is consequently used to identify the torque, as well as to **compensate the phase shift between the speed ripple signal and the original torque ripple**. The effect of using wrong parameters in the estimator will be simulated in Chapter 6.

3) The **linear interpolations** presented in the **time to angle and angle to time conversions** (in 5.2.3.2 and 5.2.3.4 respectively) **can improve the learning and compensating accuracy** of RC, and additionally allow the controller to be able to work even when the sampling frequency is not integer multiple of the motor speed (known as **fractional delay issue** in [81]). Further discussions regarding this issue will be given in Chapter 6. It is worth pointing out that with the proposed interpolations, the control is able to **work at a wider frequency range** since the fractional delay issue is solved.

4) Another improvement for using the proposed control at variable speed is to use the **transient detector** presented in 5.2.4.3. Benefiting from the transient detector, the proposed control is able to **maintain the suppression of torque ripple** not only in steady state, but also **during transients**.

The proposed control scheme will then be simulated in Chapter 6 using the PMSM model built in Chapter 3 and the DBCC proposed in Chapter 4.

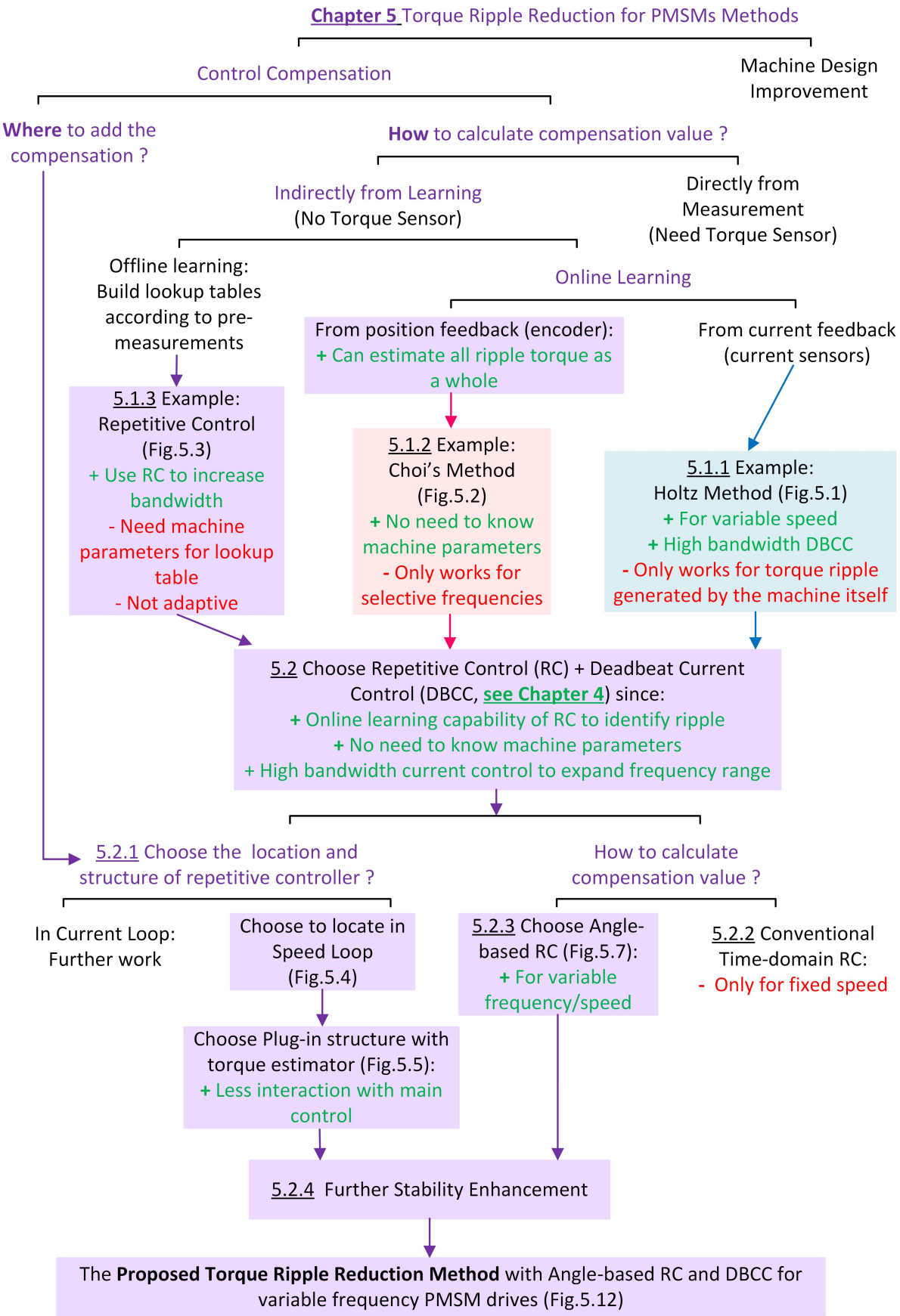


Figure 5.16: The development of the proposed control scheme in Chapter 5

Chapter 6

The Proposed Torque Ripple Reduction Method in Simulation

Four main innovative aspects (deadbeat control, interpolations, torque estimator, transient detector) have been concluded in the conclusion of Chapter 5, these four strategies are specially developed to cope with the angle-based repetitive controller, in order to allow the repetitive controller to give a full play to its ripple rejection effect.

As concluded at the end of Chapter 5, each of the four strategies benefits either the performance, or the stability of the whole control system. Therefore, it is the **aim** of this Chapter to simulatively analyse how each strategy may benefit the performance or stability of the whole control system, taking into account potential uncertainties in the plant. A brief review about the benefits of each strategy as claimed in 5.3, as well as some potential **challenges** are shown in Fig.6.1.

This Chapter is arranged as the following: The setup of the simulation models is presented in 6.1. Then, the repetitive controller is tuned in 6.2 considering the best performance in ideal condition. Additionally, the influence of four non-ideal conditions corresponding to the four innovation strategies are analysed in 6.3. Finally, the key findings from the simulation results are concluded in 6.4.

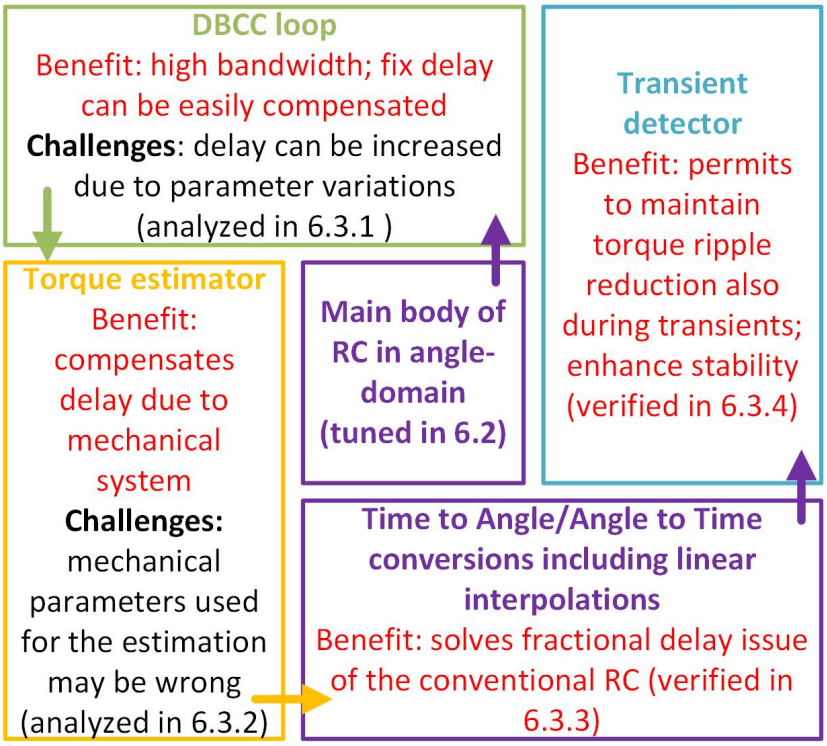


Figure 6.1: Benefits and uncertainties of the four innovative strategies

6.1 Simulation Setup

A simulation model consisting of the proposed PMSM model, an ideal inverter model, the proposed deadbeat current control loop, a normal PI speed control loop, and the proposed plug-in angle-based repetitive controller has been built using Matlab/Simulink. Among them, the PMSM model is the model presented in Chapter 3; the proposed deadbeat current control and normal PI speed control loop are the same as shown in Fig.4.6 (Chapter 4); the plug-in angle-based repetitive controller is the one illustrated in Fig.5.7 and Fig.5.12 (Chapter 5). The inverter non-linearity is not considered in the simulation.

It is necessary to point out that electrical parameters such as the dq axis inductances L_{dq} and the magnetic flux F_m are represented by lookup tables in the PMSM motor, whereas these electrical parameters set in DBCC are constant values. Hence, the L_d^{est} for DBCC is chosen to be the L_d value in PMSM model when $i_{dq} = 0$; L_q^{est} is chosen to be the smallest possible value in the PMSM model to avoid oscillation in the current response based on the discussions in 4.2.1.1. The full list of all the simulation settings for each part of the model are shown in Table. 6.1. Other nominal values about the PMSM

model are the same as in Table. 3.1.

Table 6.1: Parameters for simulation tests

Model		parameters	values [Units]	Remark
PMSM (Chapter 3)	Electrical parameters	d axis inductance $L_d(i_d, 0)$	0.0045∼0.0054 [H]	Lookup table as shown in Fig.3.4
		q axis inductance $L_q(i_d, i_q)$	0.0071∼0.0077 [H]	Lookup table as shown in Fig.3.4
		d axis magnetic flux $F_{md}(0, i_q)$	0.2691∼0.2732 [Wb]	Lookup table as shown in Fig.3.3
		q axis magnetic flux F_{mq}	0 [Wb]	
		d axis ripple flux $F_d^{ac}(\theta_e, i_d, i_q)$	-0.0019∼0.0018 [Wb]	Lookup table as shown in Fig.3.2
		q axis ripple flux $F_q^{ac}(\theta_e, i_d, i_q)$	-0.0041∼0.0039 [Wb]	Lookup table as shown in Fig.3.2
		Error electrical torque $T_e^{err}(\theta_e, i_d, i_q)$	-1.5312∼1.3279 [Nm]	Lookup table as shown in Fig.3.5
	Mechanical parameters	Moment of inertia J	0.00078 [kg/m^2]	
		Friction factor B	0.001 [Nm·s/rad]	
DBCC (Chapter 4)		d axis inductance L_d^{est}	0.0048 [H]	$= L_d(0, 0)$
		q axis inductance L_q^{est}	0.0071 [H]	= the smallest value of $L_q(i_d, i_q)$
		Magnetic flux F_m^{est}	0.2732 [Wb]	$= F_{md}(0, 0)$
		Stator resistance R_s^{est}	1.4 [Ω]	
Angle-based RC (Chapter 5)	Torque etimator	Moment of inertia J^{est}	0.00078 [kg/m^2]	
		Friction factor B^{est}	0.001 [Nm·s/rad]	
	Main body of RC	Gain of RC G_{RC}	0.3	tuned in 6.2
		Forgetting factor of RC Q_{RC}	0.999	tuned in 6.2
		Length of memory N	200	calculated in 6.2
	Angle to time conversion	M for phase lead compensator	2	tuned in 6.2
	Transient detector	Threshold for transient δT_e^{max}	0.4 [Nm]	See 5.2.4.3 for definitions
		Inspection duration for transient δt	$30T_s$	
		Minimum steady state time t^{ste}	0.1 [s]	
Global parameters		Number of pole pairs Pp	3	
		Sampling frequency f_s	10 [kHz]	

6.2 Performance of the Proposed Control in Ideal Conditions

Before the simulation results are presented, it is worth declaring the "**ideal condition**". The ideal condition includes all the following conditions:

Condition 1: All the electrical parameters of DBCC are correct, and the inverter is also an ideal voltage source with no voltage distortions. Therefore, the delay of the current loop is fixed to be $2T_s$.

Condition 2: All the mechanical parameters in the torque estimator are correct. Therefore, the torque can be perfectly estimated.

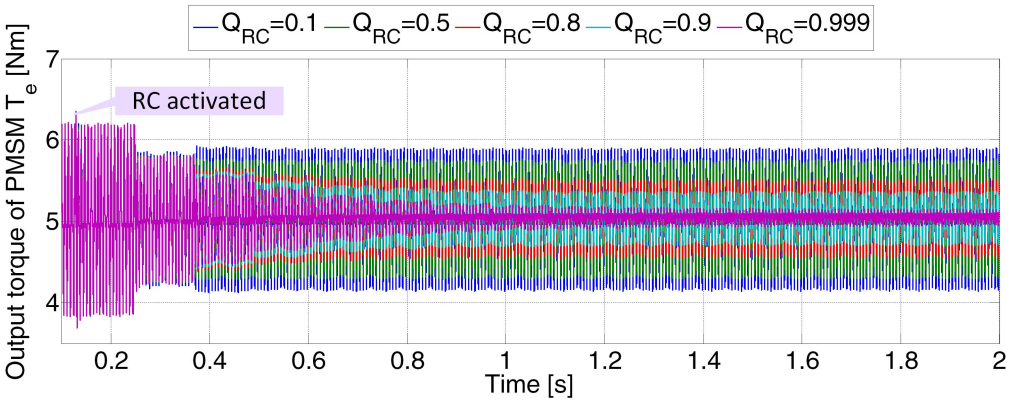
Condition 3: Due to the limitation of memory, the proposed angle-based repetitive controller only memorize torque ripple for N chosen rotor locations, $2\pi/N, 4\pi/N, ..., 2(N-1)\pi/N$. And $N = 60f_s/N_r^{rat}$ (where, N_r^{rat} is the rated speed of the machine in rpm; f_s is the sampling frequency. For this work, $N = 60 \times 10k/3000 = 200$). The moment when the measurements are taken is assumed to be exactly when the rotor passes one of these positions. One necessary but not sufficient condition for achieving this is that the rated speed N_r^{rat} is always a integer multiple of the reference speed N_r^{ref}.

Condition 4: Normally, the conventional time-based RC is disabled during the transient. Therefore, the ideal working condition for RC is at steady state.

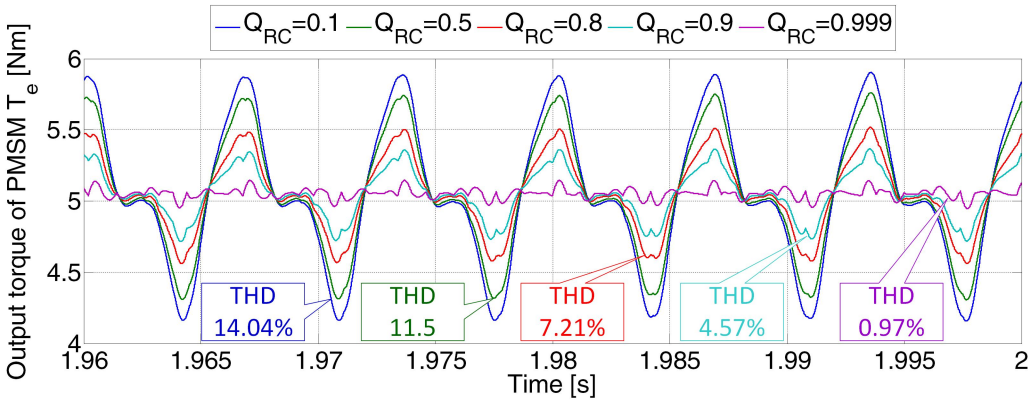
Under all the four ideal conditions, the gain and forgetting factor of the proposed RC are tuned in the following subsections.

6.2.1 Influence of Forgetting Factor Q_{RC} on Performance

The output electrical torque T_e of the proposed PMSM model with different settings of forgetting factor Q_{RC} can be seen from Fig.6.2. During the test, gain G_{RC} is kept at 0.3. The speed reference N_r^{ref} is kept at 500 rpm.



(a)



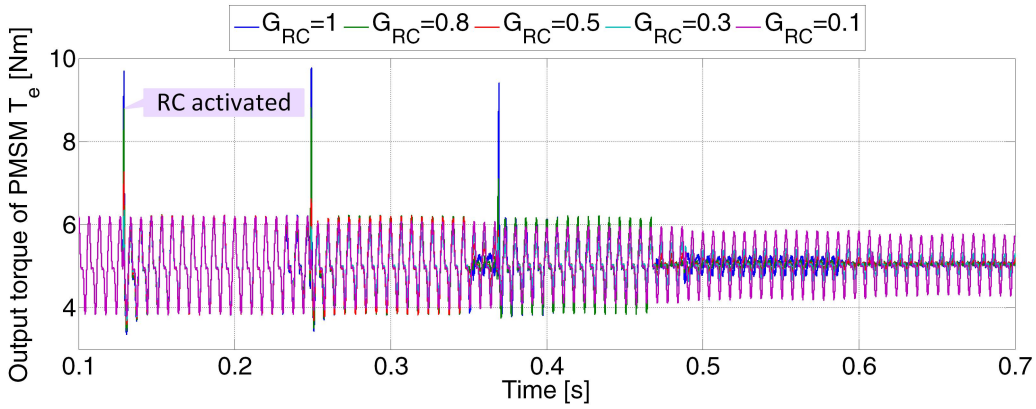
(b) Zoom in of (a)

Figure 6.2: The resultant output torque of PMSM with different forgetting factor Q_{RC}

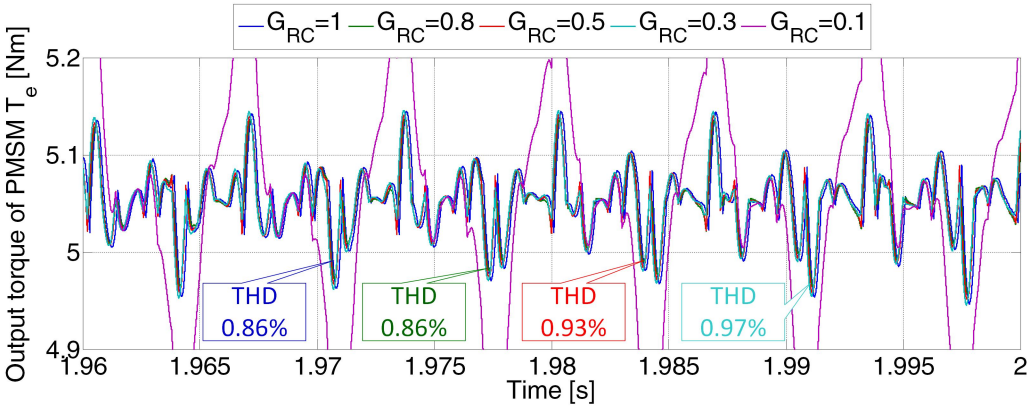
Fig.6.2 shows that the rejection of torque ripple is strongly related to the setting of forgetting factor Q_{RC}. As can be seen, the THD of torque is worse with smaller forgetting factor. This can be easily understood since more torque ripple may remain in the system if the repetitive controller "forgets" more. Combining the discussions after Fig.5.14, it shows that the control system can be more stable by choosing a smaller Q_{RC} with the cost of performance. For achieving the best possible performance, Q_{RC} **is set to be 0.999** for this work.

6.2.2 Influence of Gain G_{RC} on Performance

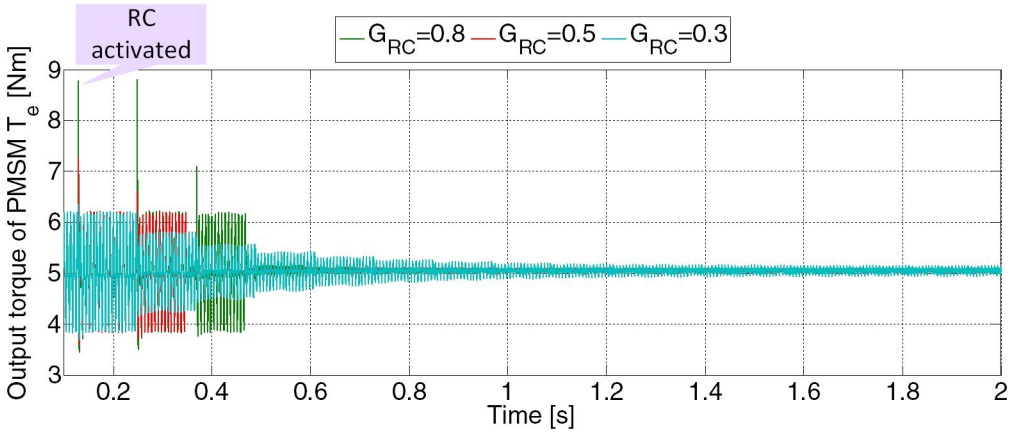
The output electrical torque T_e of the proposed PMSM model with different settings of gain G_{RC} can be seen from Fig.6.3. During the test, forgetting factor Q_{RC} is kept at 0.999. The speed reference N_r^{ref} is kept at 500 rpm.



(a)



(b) Zoom in of (a)



(c) The best three among the five

Figure 6.3: The resultant output torque of PMSM with different gain G_{RC}

The influence of gain G_{RC} can be concluded in the following two aspects: 1) By looking at Fig.6.3(a), it can be seen that the spikes are bigger with larger gain. This is because the output of RC will be larger with larger gain. When RC is initially activated, the larger the output of RC, the stronger the initial overshoot can be. This initial overshoot leads to an overshoot in the RC memory, and consequently causes an overshoot every time the rotor reaches the same position. As can be seen, the RC is able to reject these spikes after three rotations. 2) By looking at Fig.6.3(b) and Fig.6.3(c), it can be seen that the THD of output torque can be comparable eventually with gain equal to 0.3, 0.5 or 0.8. However, the convergence time when $G_{RC} = 0.5$ or $G_{RC} = 0.8$ seems to be shorter than when $G_{RC} = 0.3$. This can be translated as fast or slow RC will be able to "learn" the correct amplitude of ripple eventually. The gain setting affects the learning speed.

For the rest of this thesis, the gain of RC G_{RC} **is set to be 0.3** for smooth and fast convergence (less spikes).

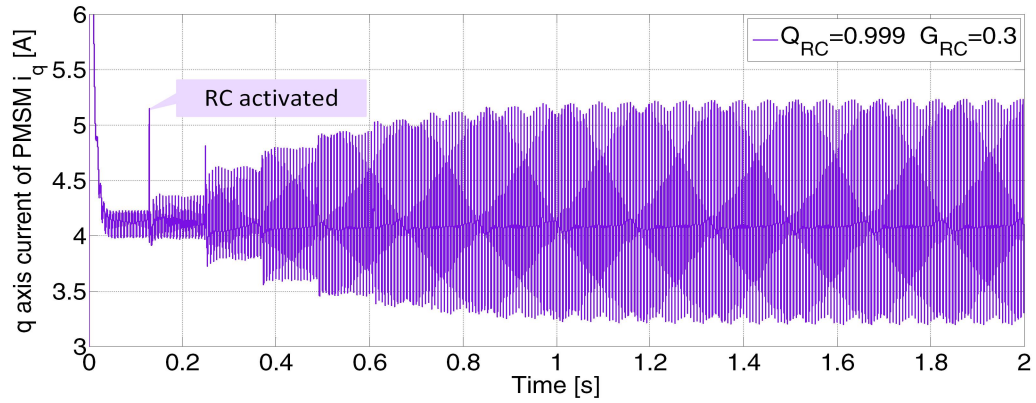
6.2.3 Performance of Proposed RC with Tuned Q_{RC} and G_{RC}

After the forgetting factor and gain of RC have been tuned, the simulation results of the q axis current i_q, output torque T_e, rotor speed in rpm N_r, and the enable signal S^{ena} calculated by the proposed transient detector (as presented in 5.2.4.3) are shown in Fig.6.4. All the parameters are set as shown in Table 6.1. The speed reference N_r^{ref} is 500 rpm.

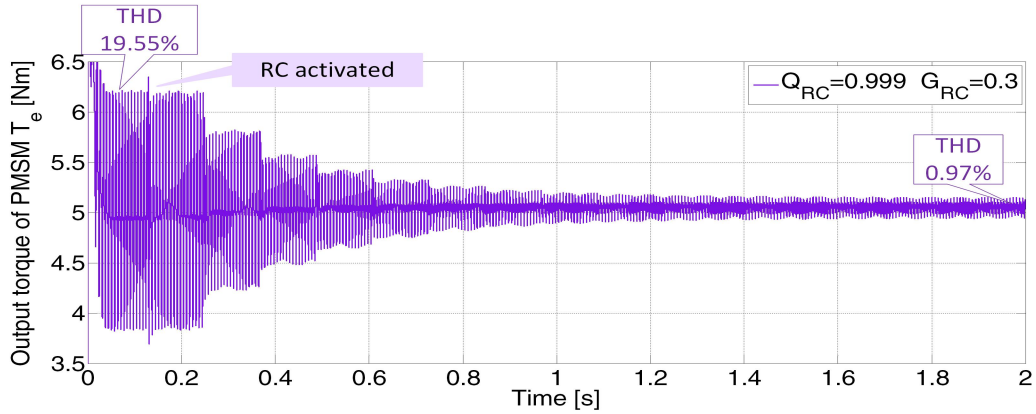
The q axis current shown in Fig.6.4(a) confirms the fact that RC reduces torque ripple through inducing current ripple. As can be seen, the current ripple increases gradually after the RC has been enabled. Fig.6.4(b) and Fig.6.4(c) show the effectiveness of the proposed RC under ideal condition. Both the torque ripple and speed ripple are reduced significantly (**torque THD** reduces from 19.55% to 0.97%, **speed oscillation** reduces from ±13 rpm to ±0.5 rpm). The speed **converges** after about 10 full revolutions' rotation from the moment RC is activated. Fig.6.4(d) shows the **successful detection**

of steady state by the transient detector ($S^{ena} = 0$ indicates the transient, $S^{ena} = 1$ indicates the steady state). Note that the rotor speed was originally 0 rpm at 0 s.

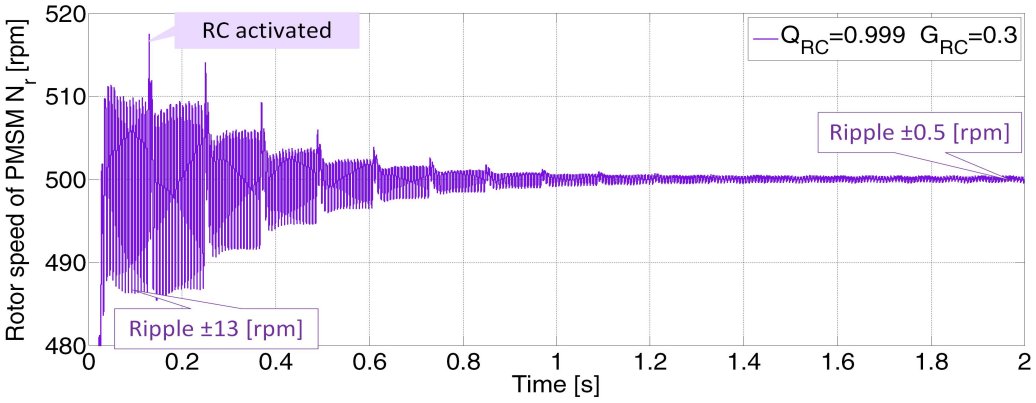
The performance shown in Fig.6.4 for the ideal condition will be used as a benchmark for comparing with the performance under non-ideal conditions in the following section.



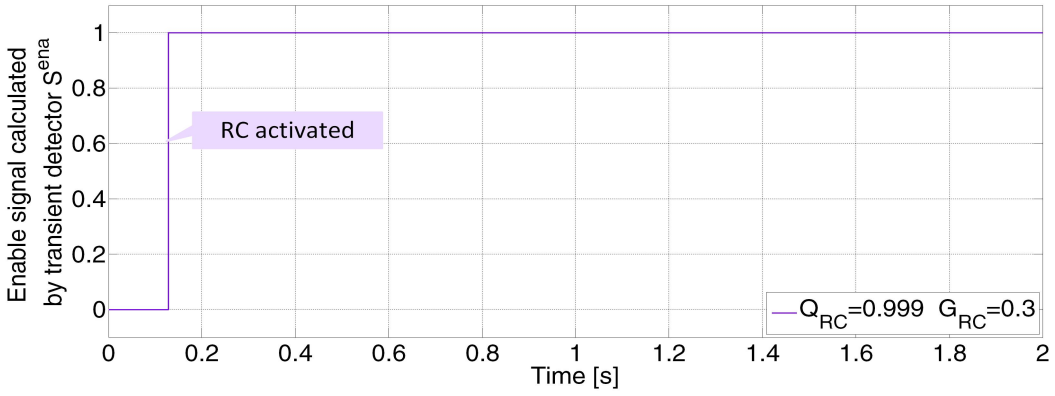
(a)



(b)



(c)



(d)

Figure 6.4: The resultant output torque of PMSM with tuned Q_{RC} and G_{RC}

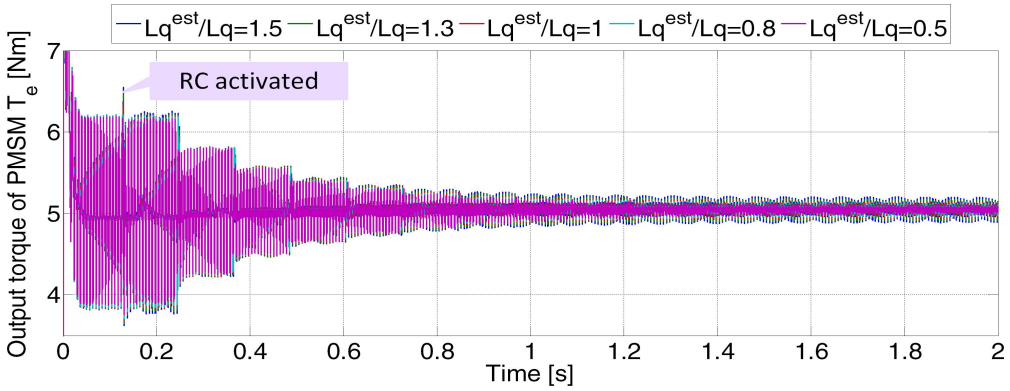
6.3 Performance and Stability of the Proposed Control in Non-ideal Conditions

As may have been noticed, the four "ideal" conditions defined at the beginning of 6.2 correspond to the four innovative aspects in the proposed control concluded at the end of 5.3. As shown in Fig.6.1, in the following four subsections, the performance of the proposed RC with either one of the four ideal conditions becoming non-ideal will be presented.

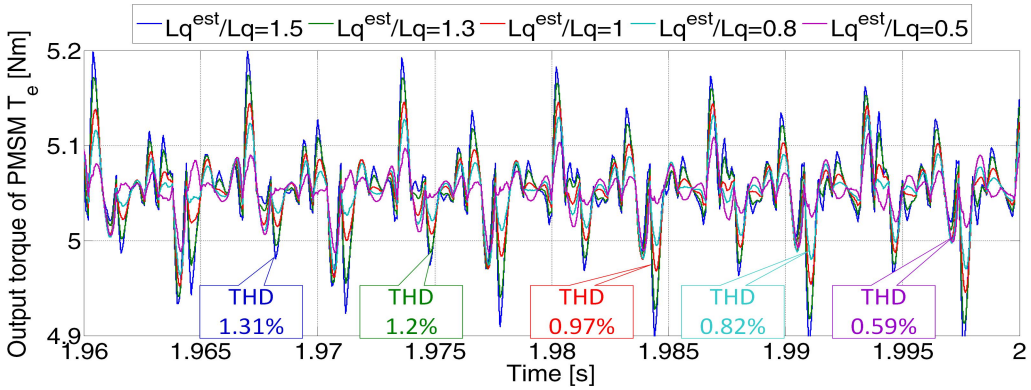
6.3.1 Influence of Uncertainties in Deadbeat Current Control Loop

In *Condition 1* of the four ideal conditions, the deadbeat current control loop is assumed to be perfect, which is true when the machine parameters in the controller match the real parameters of the machine. Therefore, it is worth investigating the performance when there are mismatches in those parameters.

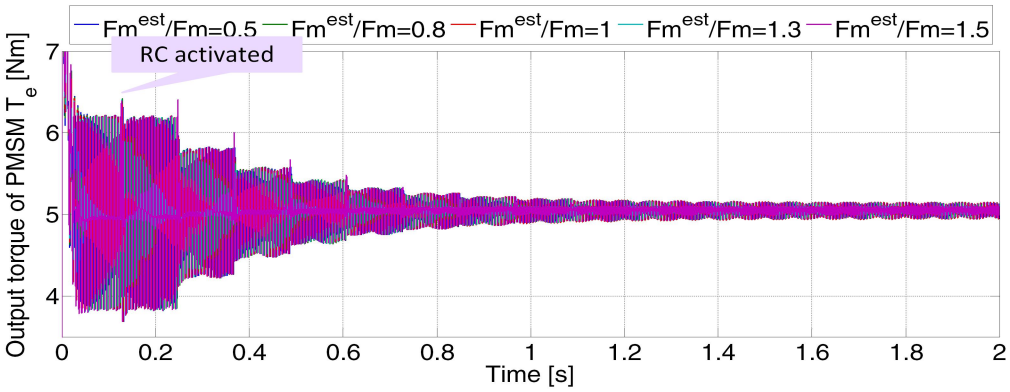
According to the DBCC performance analysis in 4.2.1.1 and 4.2.1.2, we know that the bandwidth and phase response of the proposed DBCC are mainly influenced by the q axis inductance L_q, whereas its steady state error is influenced the most by L_q and magnetic flux F_m. Therefore, the tests below are carried out with L_q^{est} and F_m^{est} in the DBCC controller varies from 50% to 150% of the real value in PMSM. The speed reference N_r^{ref} is again 500 rpm. All the other parameters are as in Table 6.1. The resultant output torque of PMSM is shown in Fig.6.5.



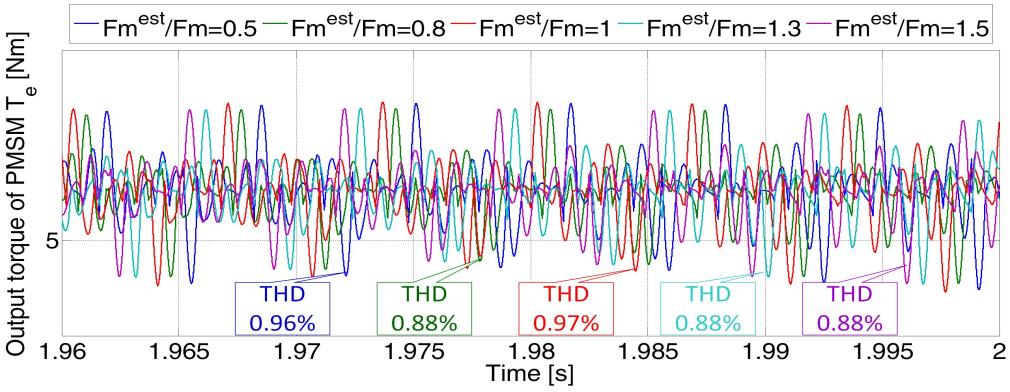
(a) Mismatch of q axis inductance L_q



(b) Zoom in of (a)



(c) Mismatch of magnetic flux F_m



(d) Zoom in of (c)

Figure 6.5: The resultant output torque of PMSM with parameter mismatches in deadbeat current control

Generally, the biggest concern for RC is that the delay of the current loop may increase due to parameter variations. As analysed in 5.2.4.2 (Fig.5.15), the delay of the current loop is critical for the stability of the control system, and the choice of M for the phase lead compensator also depends on this delay.

Therefore, it is positive to see from the results in Fig.6.5(a) and Fig.6.5(b) that, stability is maintained when the q axis inductance in DBCC L_q^{est} varies from 50% to 150% of the real L_q in motor. This confirms indirectly the bode diagrams in 4.2.1.1 that **the influence of parameters on the delay of current loop (i.e. phase response) is small**.

THD of the resultant torque does increase from 0.59% to 1.31% as L_q^{est} increases from 50% to 150%. This matches the suggestions in 4.2.1.1 that **it is better to choose L_q^{est} for DBCC as the smallest possible L_q in machine**. It is worth noting that, according to Fig.4.8(c), the bandwidth of current loop would decrease as L_q^{est} decreases. As a consequence, the machine would produce less current harmonic dependant torque ripple. Therefore, the torque THD when $L_q^{est}/L_q = 0.5$ is better than when $L_q^{est}/L_q = 1$, and this is not because the proposed RC works better, but because the machine produces less torque ripple.

Moreover, Fig.6.5(c) and Fig.6.5(d) show the robustness of the proposed RC against steady state errors in DBCC. As can be seen, the performance of torque ripple reduction is barely affected by mismatch of F_m. This finding is important since it confirms the fact that the proposed RC is adaptive to steady state errors in current response, so that **high performance speed control can be achieved even with an imperfect deadbeat current loop.**

Again, keeping consideration that L_q is the parameter that affects the dynamic characteristics of DBCC the most, and F_m is the parameter that affects the steady state error of DBCC the most, it is therefore reasonable to believe that **the parameter variations of PMSM have limited influence on the performance of the proposed RC.**

6.3.2 Influence of Uncertainties in Torque Estimator

In *Condition 2* of the four ideal conditions, the torque estimator is assumed to be perfect, which is true when the mechanical parameters (such as moment of inertia J^{est}, friction factor B^{est}) in the torque estimator match the real parameter (i.e. J, B) of the mechanical system. The following tests are done for investigating the influence of mechanical parameter mismatches on the performance of the proposed RC.

Mechanical parameters (J^{est} and B^{est}) for the torque estimator are set to be from 0.1 times to 10 times of the real J and B in machine. All the other parameters are as shown in Table 6.1. Speed reference N_r^{ref} is kept to be 500 rpm. The resultant output torque of PMSM is shown in Fig.6.6.

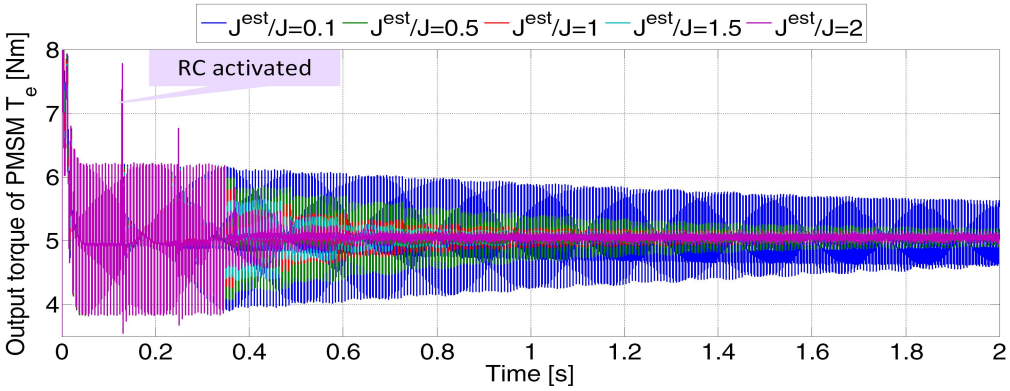
As can be seen from Fig.6.6, it seems that the parameter mismatch of J^{est} may have an impact on the performance of RC when it is too small (0.1 times of the real parameter in the mechanical system). And, the performance seems to be robust when J^{est} is bigger than the real J or when B^{est} changes from 0.1 times to even 10 times of real B.

To explain this phenomenon, it is worth looking at the bode diagrams of $G_{te}(z)G_{sp}(z)$ with parameter mismatches shown in Fig.6.7. (As introduced at the beginning of 5.2.4, $G_{te}(z) = \frac{J^{est}z-(J^{est}-B^{est}T_s)}{T_s z}$ is the transfer function of torque estimator. $G_{sp}(z) = \frac{T_s}{Jz-(J-BT_s)}$ is the real mechanical system function. It can be seen that $\mathbf{G_{te}(z)G_{sp}(z) = z^{-1}}$ when there is no parameter mismatch.) The main ripple torque when the speed is 500 rpm (=8.3Hz) are the 6^{th} and 12^{th} harmonics, thus 50 Hz and 100 Hz. As shown in Fig.6.7(a), in case of the J^{est} mismatch, the worst case happens when $J^{est}/J = 0.1$ (blue line), where both 50 Hz and 100 Hz signals are attenuated for -20 dB. This attenuation prevents the RC from "seeing " the ripple in torque, therefore, the performance degrades as shown in Fig.6.6(b). On contrary, since there are no significant attenuations or phase shifts under all the other test conditions, performance of RC is maintained.

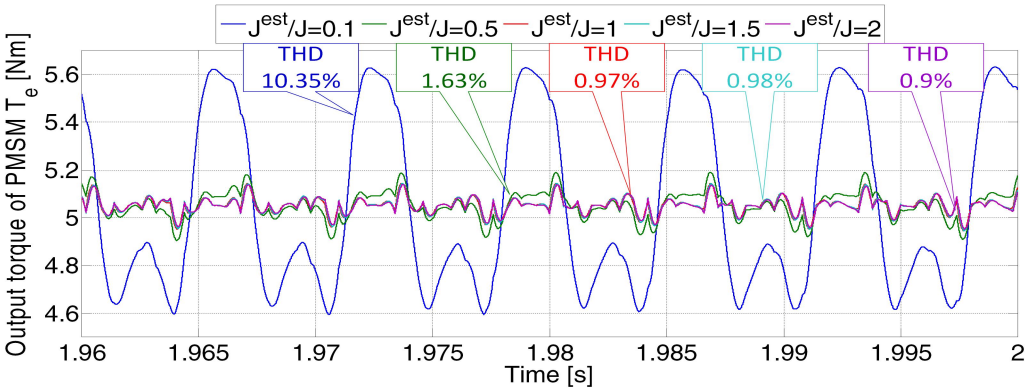
Another phenomenon is that the best steady state THD is achieved when $J^{est}/J = 2$, as shown in Fig.6.6(b). However, compared with the ideal condition as shown in

Fig.6.4(a), the transient behaviour is worse. This is because according to Fig.6.7(a), ripple would be amplified when $J^{est}/J = 2$, therefore greater transient spikes are present in Fig.6.6(a) than in Fig.6.4(a).

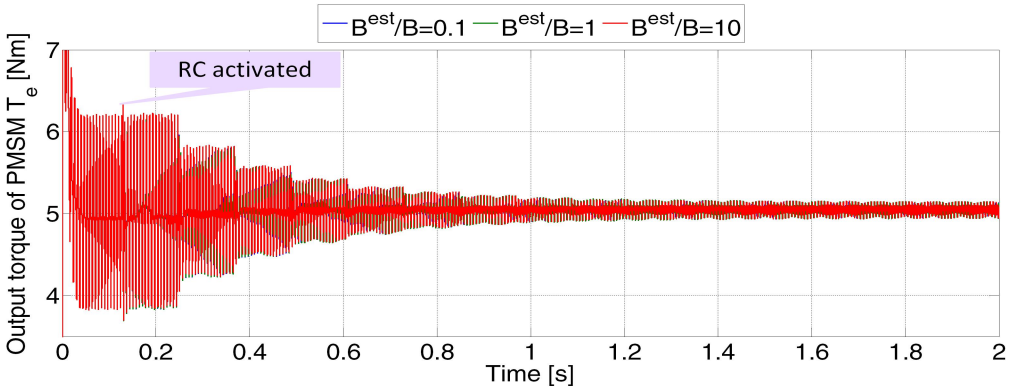
In summary, the mismatch of J^{est} and B^{est} does not necessarily lead to degrade of performance. **The level of its impact on the performance depends on the frequency** of target ripple. Meanwhile, the **stability is maintained for all the tested conditions.**



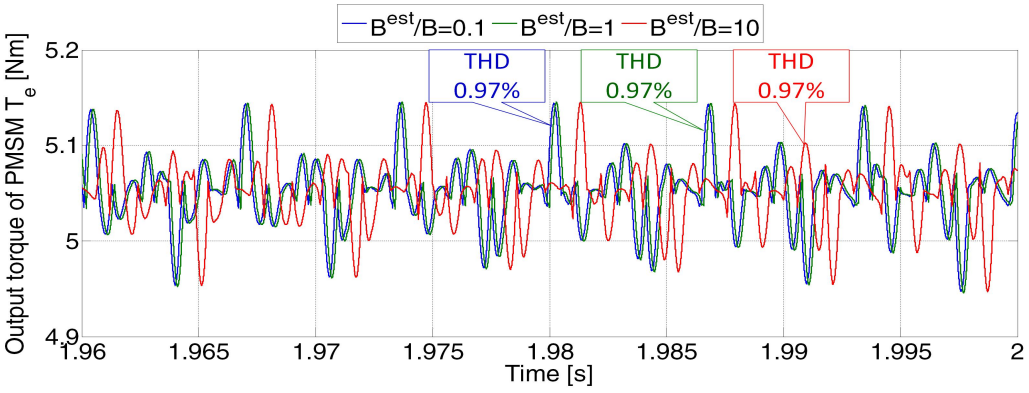
(a) Mismatch of moment of inertia J^{est}



(b) Zoom in of (a)

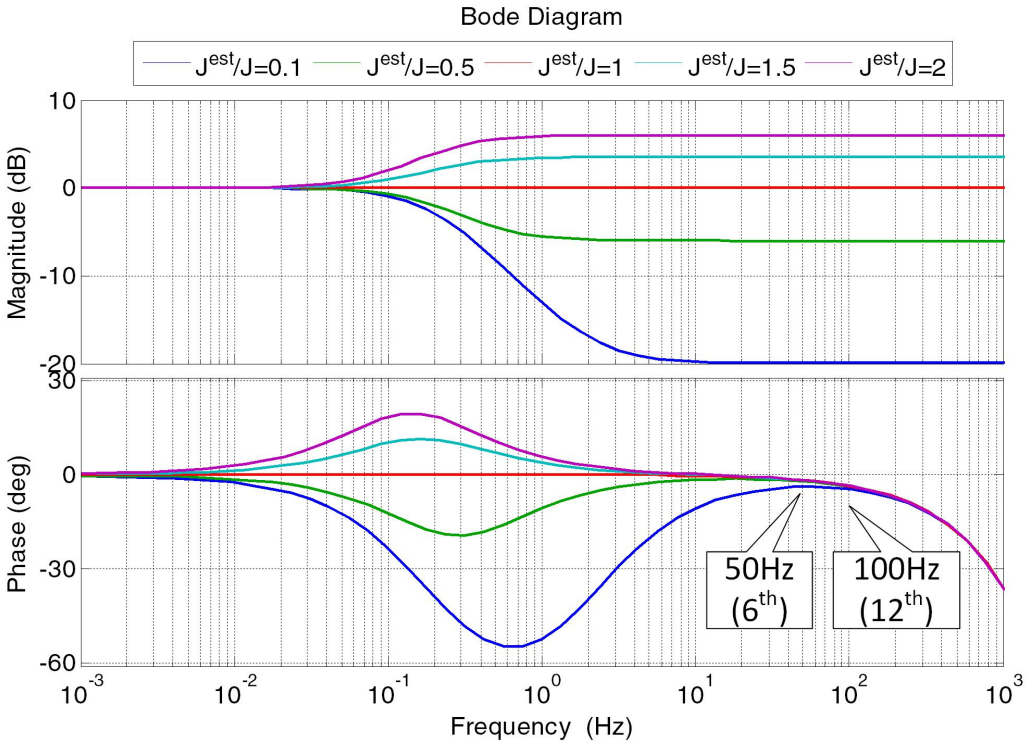


(c) Mismatch of friction factor B^{est}

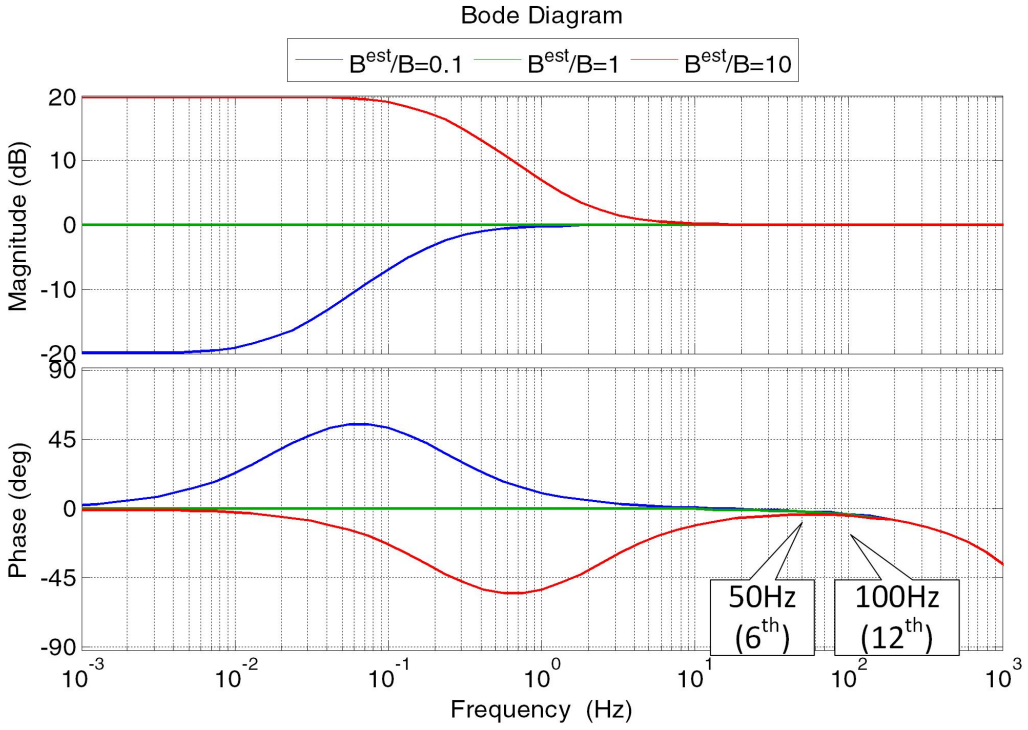


(d) Zoom in of (c)

Figure 6.6: The resultant output torque of PMSM with parameter mismatches in torque estimator



(a) Mismatch of moment of inertia J^{est}



(b) Mismatch of friction factor B^{est}

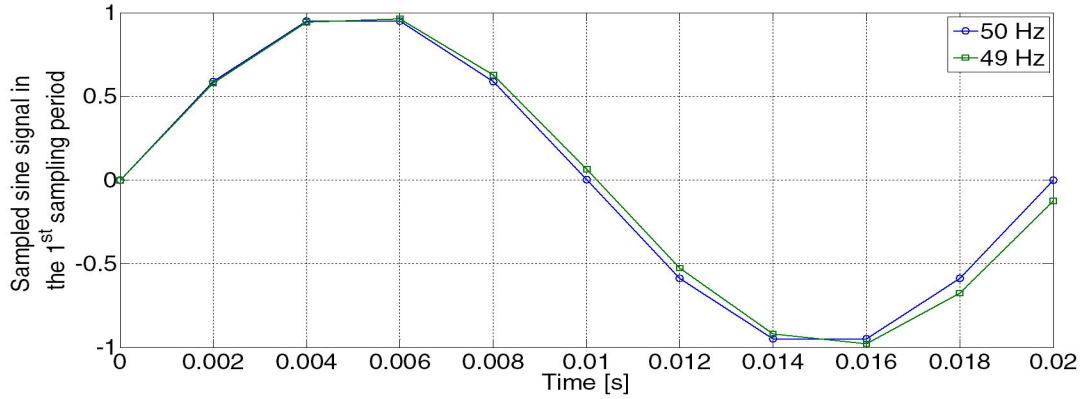
Figure 6.7: Bode diagram of $G_{te}G_{sp}$ with parameter mismatches

6.3.3 Effectiveness of Interpolations for Solving Fractional Delay Issue

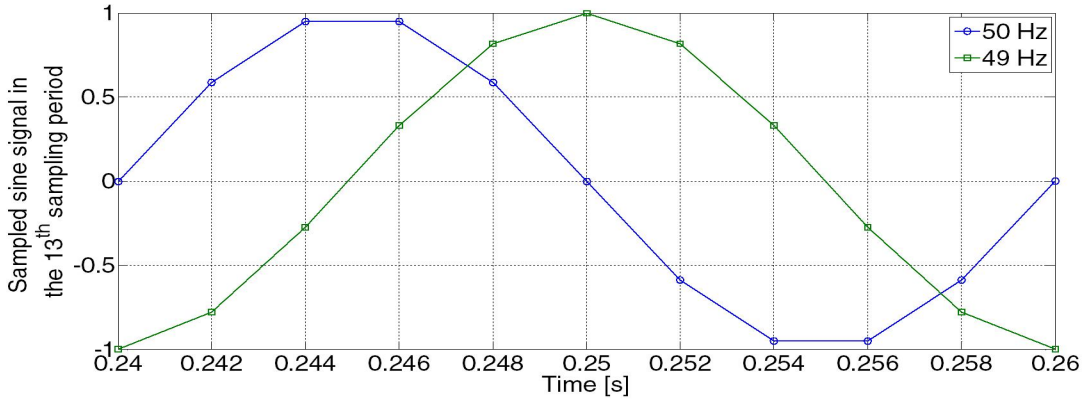
Before starting the simulation, it is necessary to define the **fractional delay issue** for the conventional time-domain RC, and **what would this issue be in the angle-domain**.

Essentially, fractional delay issue is caused by the fact that RC is a discrete controller, whereas the target ripple is a continuous signal. When recording a continuous signal in a discrete form, **phase error** may occur if the sampling frequency T_s can not be evenly divided by the frequency of the target ripple.

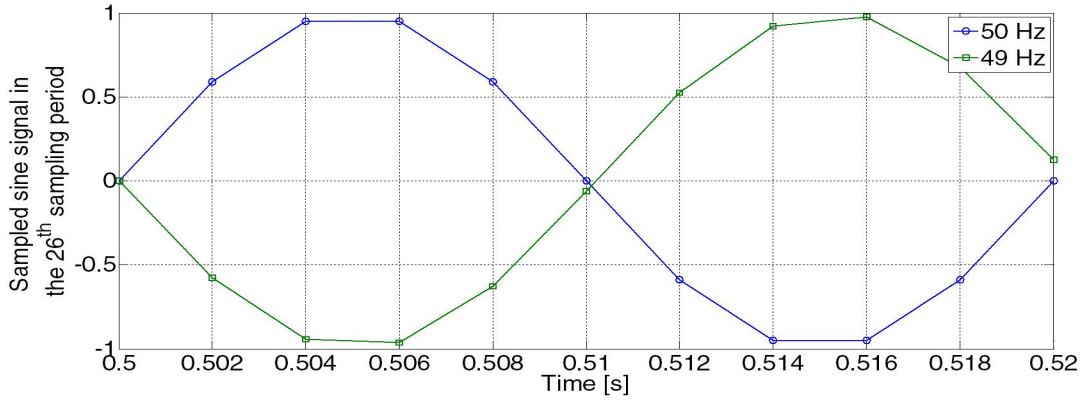
For example, 50 Hz and 49 Hz sinusoidal signals are sampled at the same sampling rate of 500 Hz. For designing the delay chain in a time-domain RC, it requires 10 delays for covering one full period of the 50 Hz signal, and 10.2 delays for the 49 Hz signal. Since the fractional number of delays are impossible, we have to set it 10 delays also for the 49 Hz signal. Consequently, from a scope of 10 samples, as in Fig.6.8, the sampled 50 Hz signals are identical in all the three graphs, whereas the sampled 49 Hz signal **gradually shifts its phase as time**. RC compensates the target ripple by generating the reverse signal according to the signal sampled 10 samples ago. Therefore, if the signal is not repetitive every 10 delays, the waveform recorded through the delay chain will not be always synchronised with the original ripple waveform, and a risk is that wrong compensation actions may be generated. This issue is called the fractional delay issue.



(a) The 1^{st} ten samples



(b) The 13^{th} ten samples

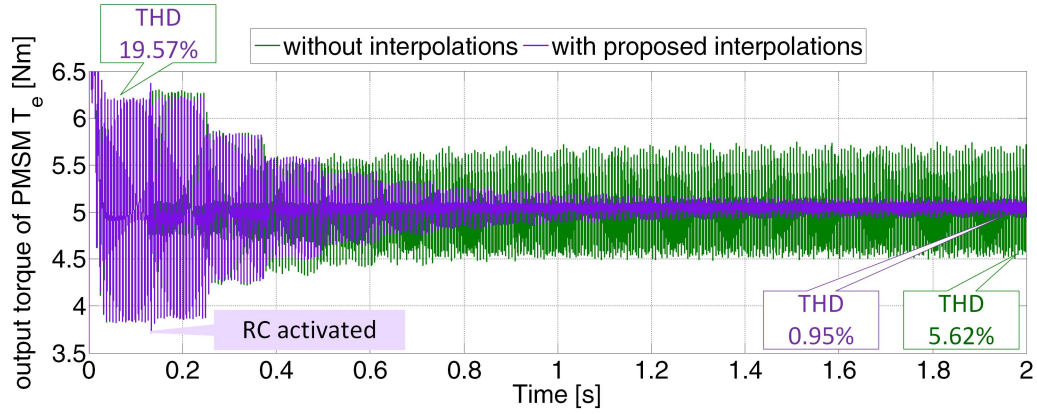


(c) The 26^{th} ten samples

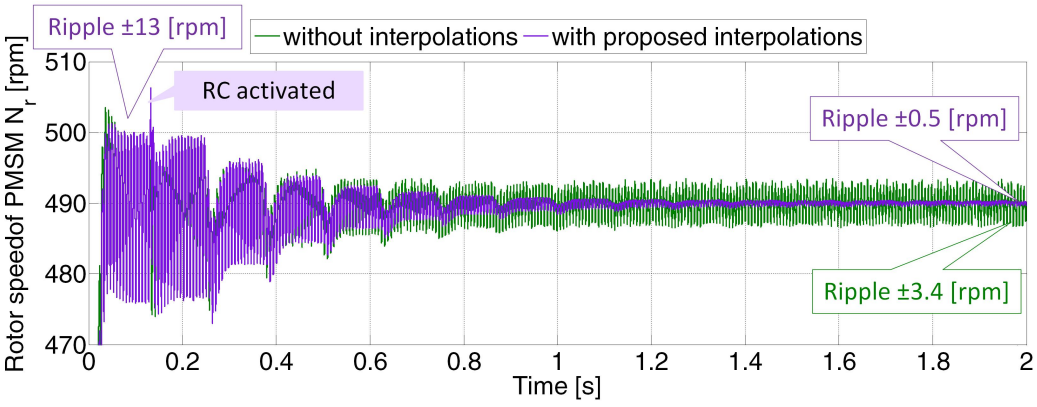
Figure 6.8: Phase error in samples due to fractional delay issue

In the angle-domain, the synchronization problem appears when the sampled ripple waveform is not aligned with the correct rotor position. The angle-based RC only records ripple at N chosen rotor locations ($\theta_m = 0, 2\pi/N, ..., 2(N-1)\pi/N$). However, there is no guarantee that the sampling is taken place exactly when the rotor position equals to a chosen rotor location. Consequently, interpolations in time to angle conversion are proposed to interpolate the ripple at exactly the closest chosen rotor position, so as to increase the resolution of sampling.

In *Condition 3* of the four ideal conditions, this misalignment is minimised since the sampling frequency T_s can be evenly divided by the machine rotation frequency (revolution per second). In the following test, the rotor speed N_r^{ref} is set to be 490 rpm, all the other parameters are the same as in Table. 6.1. The resultant output torque T_e, rotor speed N_r with and without the proposed interpolations in the time-to-angle/angle-to-time conversions are compared in Fig.6.9.



(a)



(b)

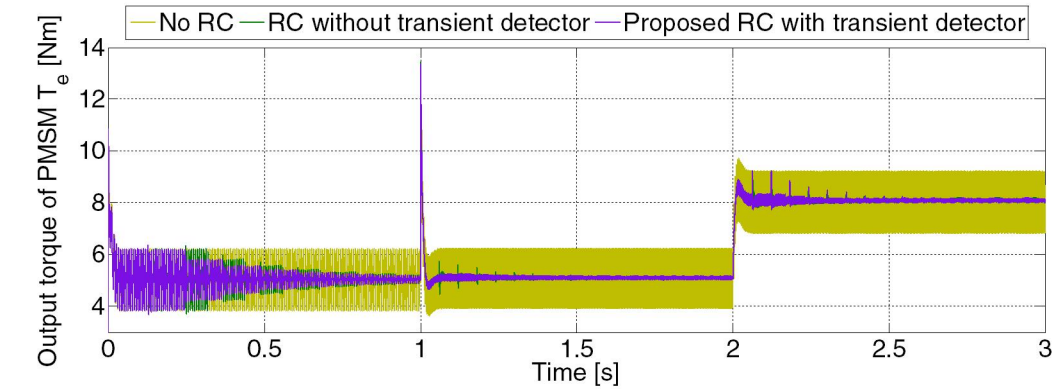
Figure 6.9: The resultant output torque of PMSM with/without proposed interpolations

As can be seen, the performance degrades significantly without the proposed interpolations. With the proposed interpolations, the performance at 490 rpm is the same as the ideal performance at 500 rpm shown in Fig.6.4. It is important to know from this test that, through the usage of interpolations, the proposed angle-based RC is able to maintain high performance for different speed/frequencies, which makes it promising for using this method at variable speed/frequency. The remaining question is how would the proposed RC behave during speed transients? (see the following subsection)

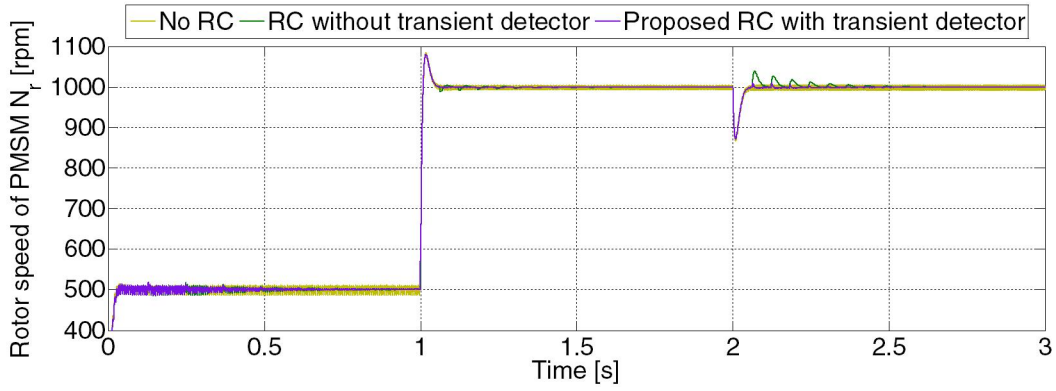
6.3.4 Effectiveness of Transient Detector

In *Condition 4* of the four ideal conditions, the proposed RC works only at steady states. The main concern of keeping RC activated during transients is that, some transient overshoots (spikes) may occur due to the "fake" torque error at a sudden change (as discussed in Fig.6.3(a)). Such transient ripple would naturally disappear as the transient finishes. However, RC is not able to recognize such non-periodic ripple, and consequently will generate wrong compensation actions. Therefore, the principle of the proposed transient detector is to detect the transients. During transients, the input of RC is set to zero, while the output of RC is still activated. Theoretically, there will be no overshoots (spikes) during the transients if the proposed transient detector is working perfectly.

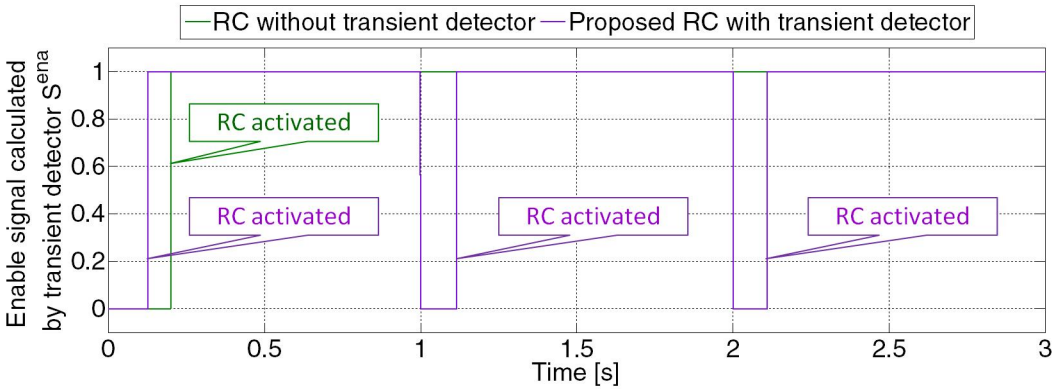
The following tests are carried out with/without the proposed transient detector during a speed and a load transient. All the parameters are set according to Table. 6.1. The speed reference N_r^{ref} steps from 501 rpm to 999 rpm at 1s. Load torque T_l steps from 5 Nm to 8 Nm at 2s. The output torque T_e, the speed N_r, and the enable signal S^{ena} calculated by the proposed transient detector are shown in Fig.6.10. A zoom in of the torque and speed behaviour during the speed step and load step is shown in Fig.6.11.



(a)

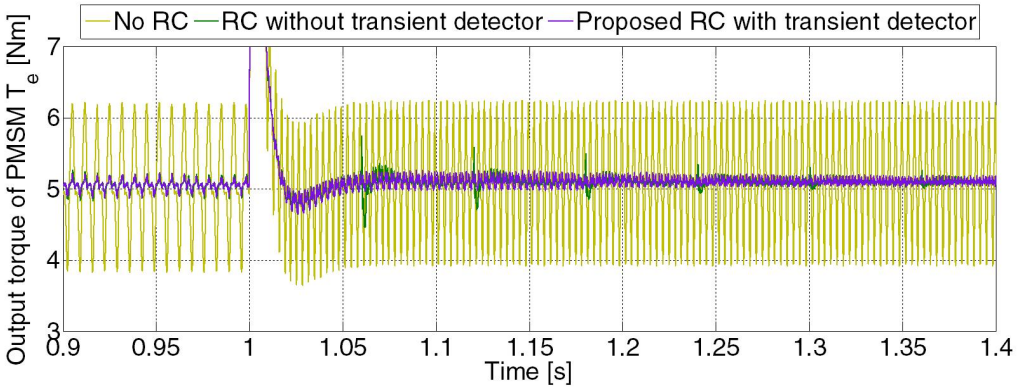


(b)

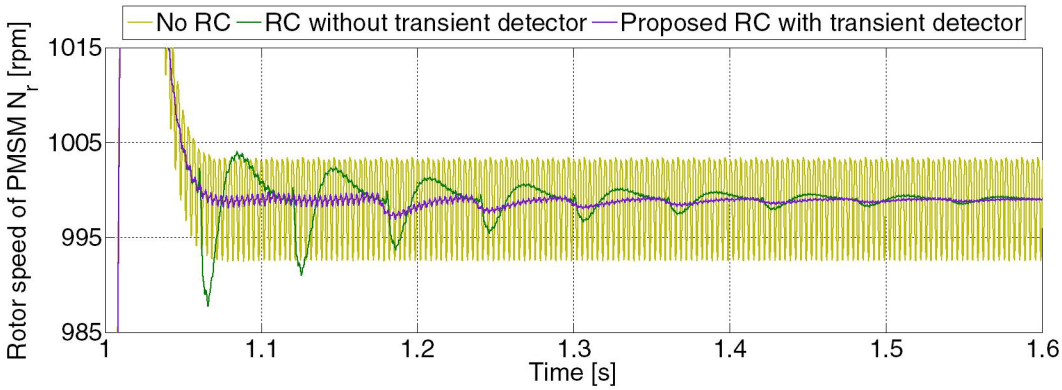


(c)

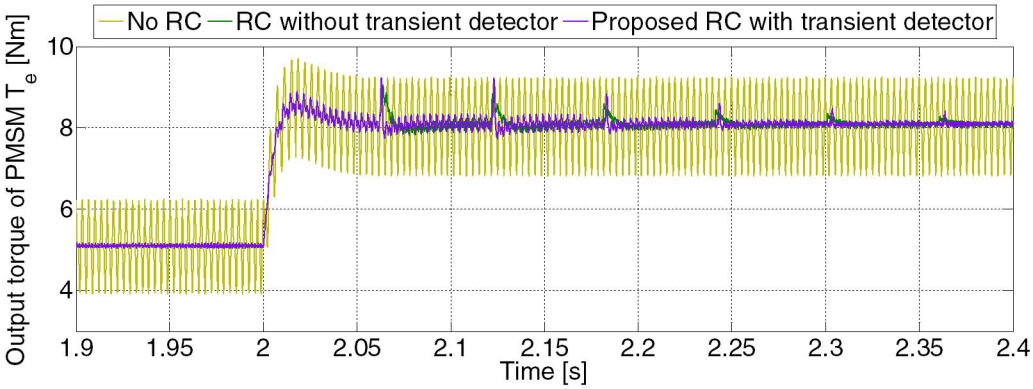
Figure 6.10: The resultant output torque of PMSM with/without the proposed transient detector



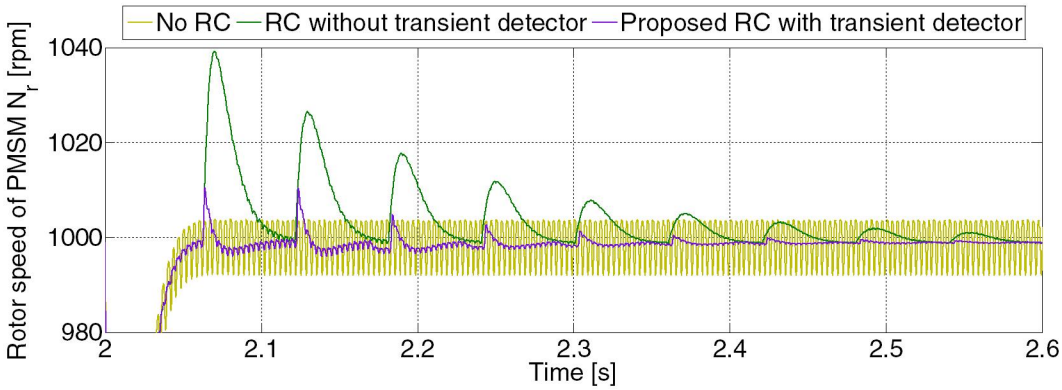
(a) Speed reference steps at 1s



(b) Speed reference steps at 1s



(c) Load torque steps at 2s



(d) Load torque steps at 2s

Figure 6.11: The resultant output torque of PMSM with/without the proposed transient detector (zoom in)

Fig.6.10 and Fig.6.11 confirms that the proposed transient detector is able to detect the speed change at 1s and the load change at 2s. Especially, as shown in Fig.6.10(c), the proposed transient detector can successfully detect the transients by set its output $S^{ena} = 0$ during the transients, and set $S^{ena} = 1$ when a steady state has been reached.

Fig.6.11(b) and 6.11(d) confirms that by using the proposed transient detector, the "spikes" or "overshoots" in the speed waveform during both speed and load transients are significantly reduced. The remaining "spikes" in 6.11(d) for the case of load transient is caused by the small d.c. offset in torque estimator, and it disappears after around 5 full rotations (around 0.5s at 999 rpm).

Overall, the test results confirm that **the proposed RC can maintain torque ripple reduction even during transients**.

6.4 Conclusions

This chapter focuses on verifying the effectiveness of the proposed angle-based RC under both ideal and non-ideal conditions. Four "ideal" conditions have been defined at the beginning of 6.2, which are in fact corresponding to the four innovative aspects discussed at the end of Chapter 5 (5.3). Some key findings from the simulations in this chapter can be concluded as the following:

- Under the four ideal conditions, the proposed is able to achieve **a torque THD reduction from 19.55% to 0.97%, and a speed ripple reduction from** ± 13 **rpm to** ± 0.5 **rpm**. See the results in Fig.6.4.
- The **first** ideal condition describes the ideal response of the proposed **deadbeat current control** loop. Results in Fig.6.5 show that the performance with the q axis inductance L_q and magnetic flux F_m detuned, is comparable to the ideal performance in Fig.6.4.

  Considering that L_q and F_m are the two parameters which influence the bode response and steady state error of current loop the most (according to the analy-

sis shown in Fig.4.8 and Fig.4.10), we can conclude that **the performance of proposed RC is reasonably robust against machine electrical parameter variations**.

Besides, the performance when L_q^{est} in the current controller is smaller than the real L_q in machine, is better than when it is the other way round. Therefore, it is again suggested to **set the inductances in the controller considering the smallest possible value in the machine**, as has been suggested by the results in Fig.4.8.

- The **second** ideal condition describes using accurate mechanical parameters for the proposed **torque estimator**. Simulation results in Fig.6.6 shows that ideal performance can be maintained when the friction factor B^{est} in torque estimator is wrong, and also when the moment of inertia J^{est} is bigger than the real J in mechanical system. Therefore, **it is suggested to set J^{est} for the torque estimator considering the highest possible value.**

- The **third** ideal condition is related to the fractional delay issue as defined at the beginning of 6.3.3. Through using the proposed **linear interpolations** in the **time-to-angle** (5.2.3.2) and **angle-to-time conversions**(5.2.3.4), the ideal performance can be maintained for all speed, which expands widely the operating speed/frequency range of the RC when compared with the traditional time-based RC.

- The **forth** ideal condition is related to the transient behaviour of RC. The conventional RC is normally activated only at steady state. However, the results in Fig.6.11 shows that the proposed **transient detector** is able help reducing the "spikes" or "overshoots" of the speed waveform during the both speed and load transients.

Overall, simulation results have confirmed that **the proposed angle-based RC can achieve ideal or close to ideal performance with detuned electrical or mechanical parameters, for a wide range of speed, not only at steady state but also during transients.** Limitations of the simulation tests carried out in this chapter will be discussed in Chapter 8. The effectiveness of the proposed RC will be confirmed through experimental tests in Chapter 7.

Chapter 7

The Proposed Torque Ripple Reduction Method in Experiment

As demonstrated in Chapter 6, there are four ideal operating conditions for the proposed torque ripple reduction method. Its performance when one of the four conditions becomes non-idea has been simulated to compare with the performance under all ideal conditions. Again, the four ideal operating conditions are related to the four innovative strategies that have been chosen and developed as explained in Chapter 5, in order to better cope with the repetitive controller, so as to maximize the torque ripple reduction effect. The simulative tests in Chapter 6 not only have compared the torque ripple reduction effect under ideal/non-ideal conditions, but also have verified the innovative strategies can effectively enhance the torque ripple reduction performance. The aim of this chapter is to further verify the proposed torque ripple reduction technique through experimental tests.

This Chapter focuses on verifying the effectiveness of torque ripple reduction under both constant speed/load and variable speed/load. Fig.7.1 below shows the layout of this chapter, as well as the correspondence between this chapter and the previous one.

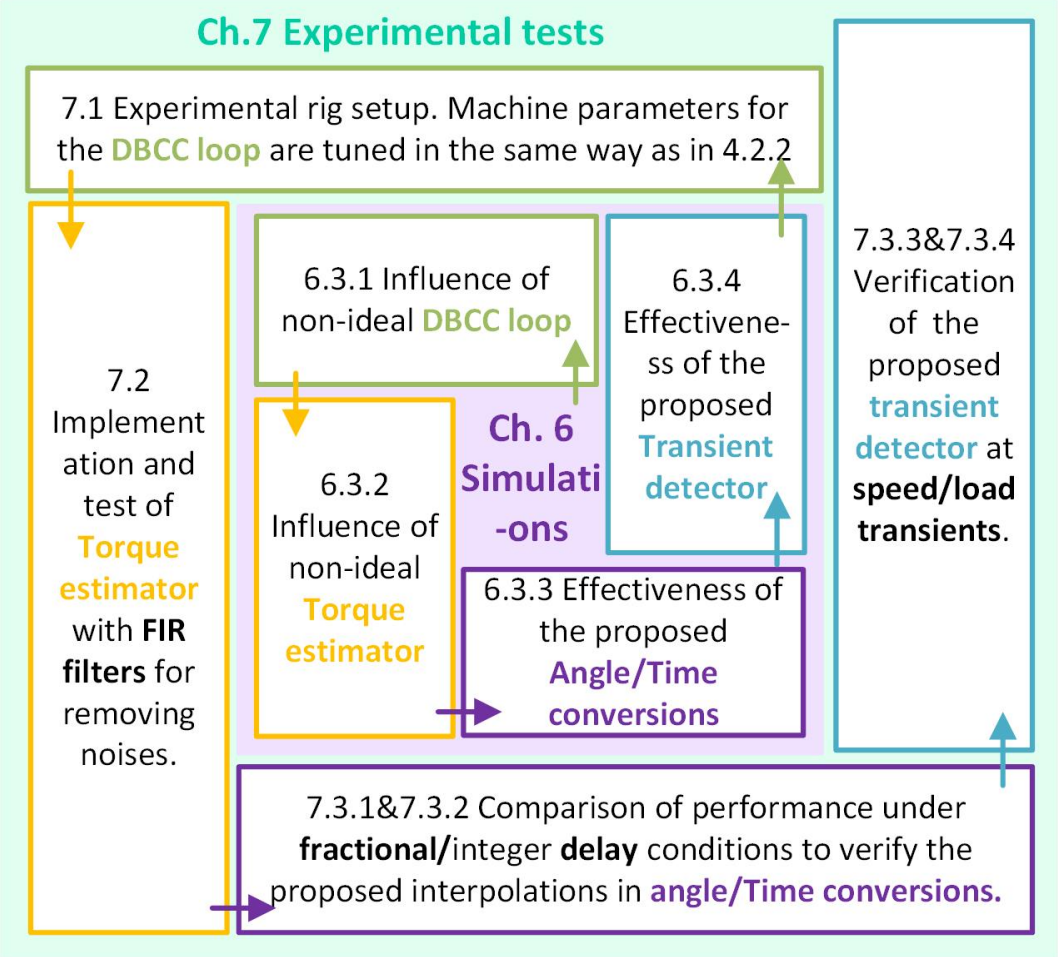
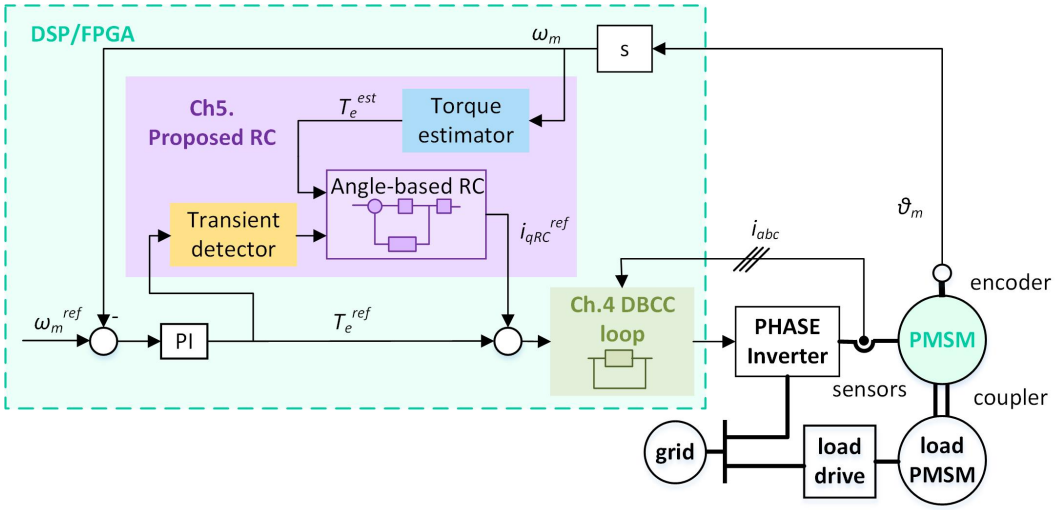


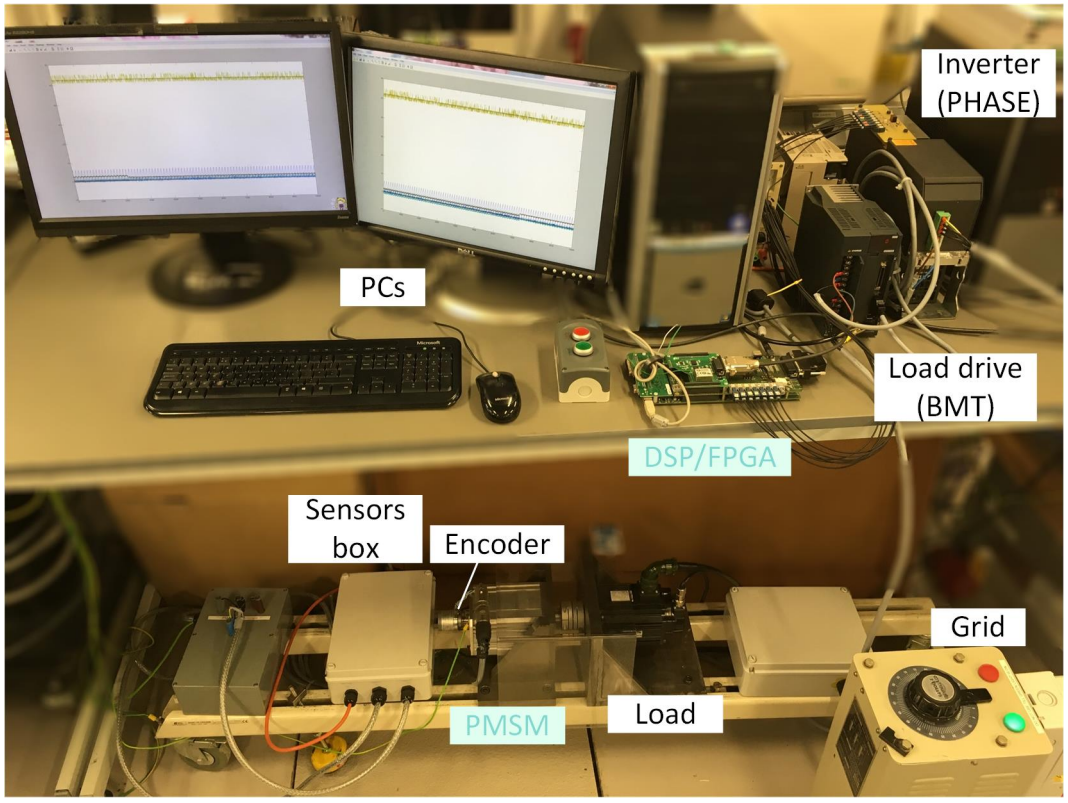
Figure 7.1: Experimental validation of the four innovative aspects

7.1 Experimental Rig Setup

An experimental rig as shown in Fig.7.2 has been set up. The DBCC loop proposed in Chapter 4 and the angle-based RC proposed in Chapter 5 are implemented using DSP/FPGA. The DSP code of the proposed repetitive controller can be found in Appendix C. A commercial inverter is used for supplying the PMSM of which the parameters are as shown in Table. 7.1. The PMSM (University prototype) is controlled under speed control mode. Another PMSM (YASKAWA SGMSV-20ADA61) has been coupled as load (coupler: Miki Pulley SFC-050SA2-24-24). This load PMSM is driven using another motor driver (Best Motion Technology (BMT)) under torque control mode.



(a) Diagram



(b) Photo

Figure 7.2: Experimental rig

The three-phase currents are measured by three current transducers (LEM LAH 25-NP), the DC bus voltage of the motor-side inverter is measured by a voltage transducer (LEM LV 25-P), the rotor position of the PMSM (University prototype) is measured by a 17-bit absolute encoder (Kubler 8.F3673.2413.C712).

The sampling frequency in DSP/FPGA is set to be the same as the switching frequency of the motor-side inverter (f_s=10 kHz). The dq axis inductances L_{dq}^{est}, stator resistance R_s^{est}, magnetic flux linkage F_m^{est} required for DBCC are shown in Table 7.1. The parameters for the torque estimator (J^{est}, B^{est}) are also included in Table. 7.1. All the other parameters for RC are the same as shown in Table. 6.1.

Table 7.1: Parameters (tuned) for experimental validation

Parameter	**Value [Unit]**	**Parameter**	**Value [Unit]**
Rated power	1.5 [kW]	d axis stator inductance L_d^{est}	0.0009 [H]
Rated speed	5000 [rpm]	q axis stator inductance L_q^{est}	0.0012 [H]
Rated torque	2.8 [Nm]	Stator resistance R_s^{est}	0.5 [Ω]
Rated line voltage	400 [V]	Magnetic flux linkage F_m^{est}	0.0566 [wb]
Rated current (RMS)	4.9 [A]	Inertia J^{est}	0.0009 [kg/m^2]
Number of pole pairs	5	Friction factor B^{est}	0.004 [Nms/rad]
Number of slots and tooth	12		

It is worth pointing out that the parameters in Table. 7.1 **have been tuned** according to the regulations discovered from both the dynamic assessment results of DBCC in 4.2.1.1 and the steady state assessment results in 4.2.1.2. During the tuning procedure, the DBCC has been implemented **with inverter non-linearity compensated** (using the compensation method described in 4.1.4). Therefore, the current control performance should be the same as shown in Fig.4.12, Fig.4.16(a), Fig.4.17(a) and Fig.4.17(c). For the rest of the experimental tests in this Chapter, the inverter non-linearity compensation is **removed**.

Another issue arisen in the implementation of the proposed angle-based RC is the noise issue. Since speed is not directly measured, but calculated from the derivation of position measurements in the experiment, and the proposed torque estimator (as in Fig.7.2) estimates torque from the derivation of speed, these two times of derivations would amplify the quantization noise contained in the position measurement. For reducing this noise, the proposed torque estimator is enhanced with Finite impulse response (FIR)

filters as presented in the following section.

The output $\bar{x}$ of a N^{th} order FIR filter during the k^{th} sampling period can be expressed as follows:

$$\bar{x}(k-N) = b_0x(k-2N-1)+b_1x(k-2N)+...+b_Nx(k-N)+...+b_{2N-1}x(k-1)+b_{2N}x(k) \tag{7.1}$$

Where, x is the rough data that needs to be filtered. $\bar{x}$ is the filtered data. For calculating the filtered value, it requires N previous data and N future data. Therefore, the $(k-N)^{th}$ value of $\bar{x}$ is calculated using $x(k-2N-1), x(k-2N), ..., x(k)$. Consequently, there is a delay of N samples between the input of the FIR filter x and its output $\bar{x}$. $b_0, b_1, ...b_{2N}$ are the weight coefficients. These coefficients are computed by using the Matlab command "fir1" in this work.

7.2 Estimation of Torque Ripple from Position Feedback with FIR Filters

Filters are necessarily required in torque estimation mainly for reducing the quantization error in the position feedback. The amplitude of quantization error is determined by the resolution of encoder as demonstrated below.

For example, the encoder used for this work is a 17-bit encoder. Hence, the minimum rotor position increment can be detected by the encoder is 4.794×10^{-5} rad (i.e. $2\pi/2^{17}$ rad). Consequently, an error of maximum 4.794×10^{-5} rad can present in the position measurement. This error is called **quantization error**. It is the possible error between the real rotor position and the sampled rotor position. As a result, the error in rotor speed ω_m if calculating from the derivation of such sampled rotor position can be 0.4794 rad/s (i.e. $4.794 \times 10^{-5}/T_s = 4.794 \times 10^{-5}/0.0001$ rad/s, where T_s=0.0001s). Furthermore, the error in the estimated torque calculated using (5.6) can be as worse as 1.9176 Nm (i.e. $0.4794 \times J^{est}/T_s = 0.4794 \times 0.0004/0.0001$ Nm).

Noting that the rated torque of the PMSM as shown in Table. 7.1 is only 2.8

Nm, such large error in the estimated torque is unacceptable. Therefore, two filters are designed. One for calculating rotor speed from the sampled rotor position, and the other one for estimating torque from the calculated rotor speed.

Regarding the choice of filters, some popular filters for digital signal processing are the infinite impulse response (IIR) filter, or the finite impulse response (FIR) filter. The definitions of these two types of filters can be found in [82]. In this work, **FIR filters are chosen for two reasons**:

1) The transfer function of a FIR filter contains no poles, but only zeros. And, the time delay between the output of the filter and its input is fixed. Whereas, a IIR filter would introduce different delays for different frequencies. Therefore, FIR filter is preferred since the delay can be easily compensated for. This is also the same reason why DBCC is preferred over PI controller as discussed in Chapter 4.

2) A FIR filter is normally higher in order than a IIR filter, and it would introduce long delays. Such long delays would not be a problem for RC, since RC itself would introduce even longer delays. Since the delay of the FIR filter is fixed and pre-known, it can be easily compensated for.

For this work, it is not compulsory but a 9^{th} order FIR filter is used when calculating rotor speed from the sampled rotor position, and another 10^{th} order FIR filter is used when estimating torque from the calculated rotor speed. (For example, the 9^{th} order FIR filter coefficients are computed using Matlab command "fir1(19,0.01,0.21)". It returns 19 ($= 2 \times 9 + 1$) coefficients for the filter, and its cut-off frequency equals $0.21 f_s/2 = 1.2 \times 10k/2 = 1.2$ kHz. The DSP code for these two filters can be found in Appendix C.) Consequently, some small modifications are done for the following equations:

For the torque estimator, the electrical torque can be estimated during the k^{th} sampling interval is now $T_e^{est}(t_{k-10})$ instead of $T_e^{est}(t_{k-1})$, since the 9^{th} order FIR filter delays the speed feedback by additional nine samples. (5.6) is therefore modified to (7.2). Moreover, after applying the 10^{th} order FIR filter for the estimated torque, the torque

estimated during the k^{th} sampling interval becomes the torque at t_{k-20}, thus $T_e^{est}(t_{k-20})$.

$$T_e^{est}(t_{k-10}) = \frac{J^{est}[\omega_m(t_{k-9}) - \omega_m(t_{k-10})]}{T_s} + B^{est}\omega_m(t_{k-10}) + T_e^{refdc} \tag{7.2}$$

For the time to angle conversion, during the k^{th} sampling interval, (5.7)-(5.9) are modified into (7.3)-(7.5).

$$T_e^{err}(t_{k-20}) = T_e^{ref}(t_{k-20}) - T_e^{est}(t_{k-20}) \tag{7.3}$$

$$k_1 = floor\Big[\frac{N\theta_m(t_{k-20})}{2\pi})\Big], k_1 \in [0, N-1] \tag{7.4}$$

In the case of **positive** rotation:

$$T_e^{pred}(k_1) = T_e^{err}(t_{k-21}) + \frac{T_e^{err}(t_{k-20}) - T_e^{err}(t_{k-21})}{\theta_m(t_{k-20}) - \theta_m(t_{k-21})}[\theta_m(k_1) - \theta_m(t_{k-21})]$$

In the case of **negative** rotation, $k_1 \neq N-1$:

$$T_e^{pred}(k_1+1) = T_e^{err}(t_{k-21}) + \frac{T_e^{err}(t_{k-20}) - T_e^{err}(t_{k-21})}{\theta_m(t_{k-20}) - \theta_m(t_{k-21})}[\theta_m(k_1+1) - \theta_m(t_{k-21})] \tag{7.5}$$

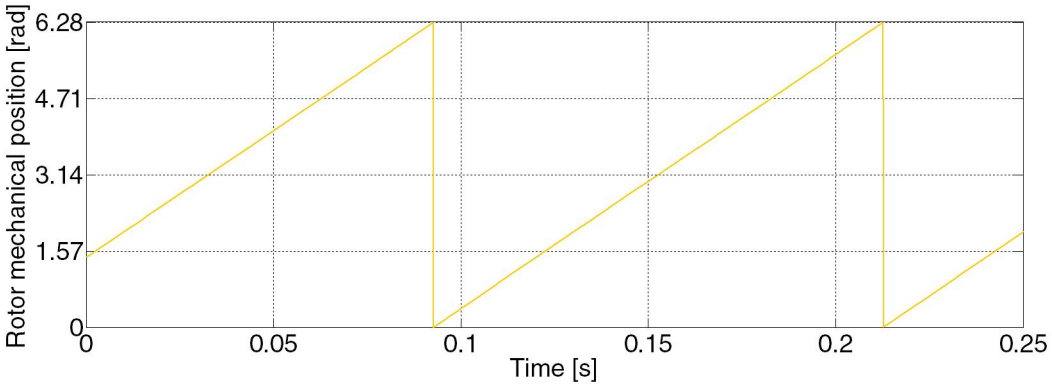
In the case of **negative** rotation, $k_1 = N-1$:

$$T_e^{pred}(0) = T_e^{err}(t_{k-21}) + \frac{T_e^{err}(t_{k-20}) - T_e^{err}(t_{k-21})}{\theta_m(t_{k-20}) - \theta_m(t_{k-21})}[0 - \theta_m(t_{k-21})]$$

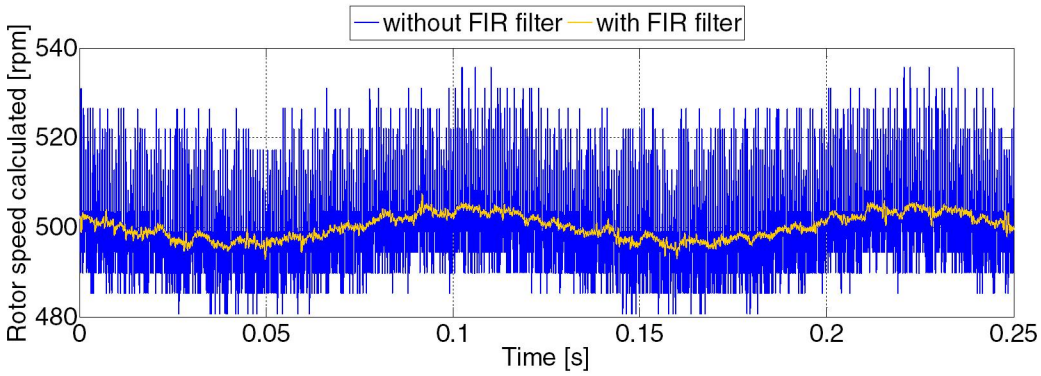
For the main body of repetitive controller, the memory can be updated the same way as in (5.10). For the angle to time conversion, during the k^{th} sampling interval, (5.11)-(5.13) stay the same to predict the rotor position after the $2T_s$ current loop delay and to apply the correct compensation current at the correct rotor position.

In order to show the necessity of applying FIR filters, and their effectiveness, an experimental test has been done by driving the motor under constant speed (500 rpm) control without repetitive controller enabled, and with no load. Fig.7.3(b) shows the resultant rotor mechanical speed ω_m calculated from the position measurement θ_m with and without the 9^{th} order FIR filter. From the FFT analysis results shown in Fig.7.3(c) and Fig.7.3(d), it can be seen that the ripple components above 1kHz have been removed

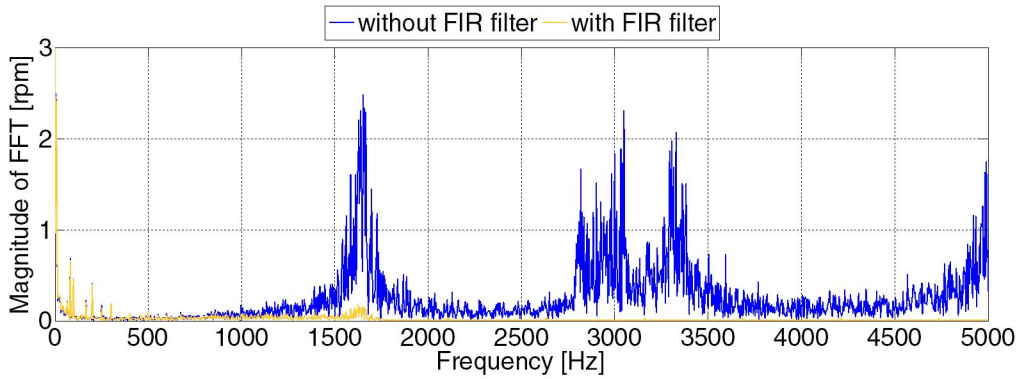
by the FIR filter, whereas the ripple components below 1kHz are kept. This matches the design of the FIR filter. The cut-off frequency of both the two filters have been designed to be 1.2kHz. Fig.7.4 shows the resultant torque estimated from the speed according to (7.2), where mechanical parameters such as J^{est} and B^{est} (as shown in Table 7.1) required for this estimation are acquired from pre-measurements, and should be reasonably correct.



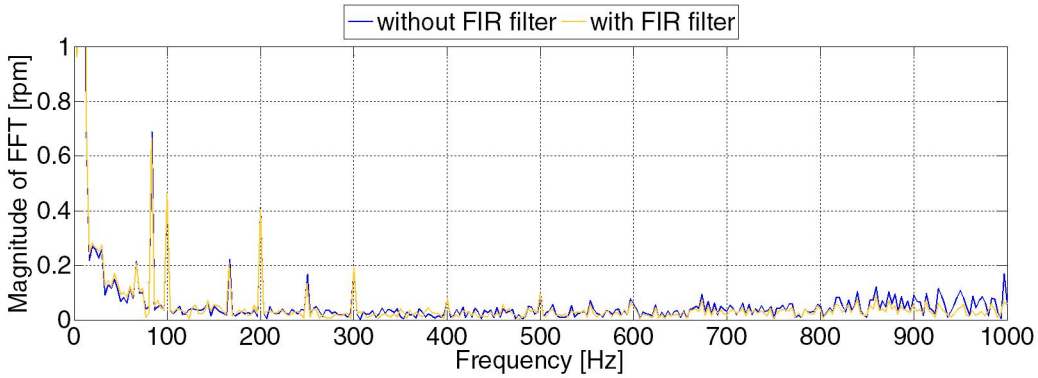
(a) Rotor mechanical position mearsurement θ_m



(b) Calculated rotor mechanical speed ω_m



(c) FFT of the calculated speed ω_m



(d) Zoom in of (c)

Figure 7.3: The calculated speed from encoder feedback (ω_m^{ref}=500 rpm, without repetitive controller)

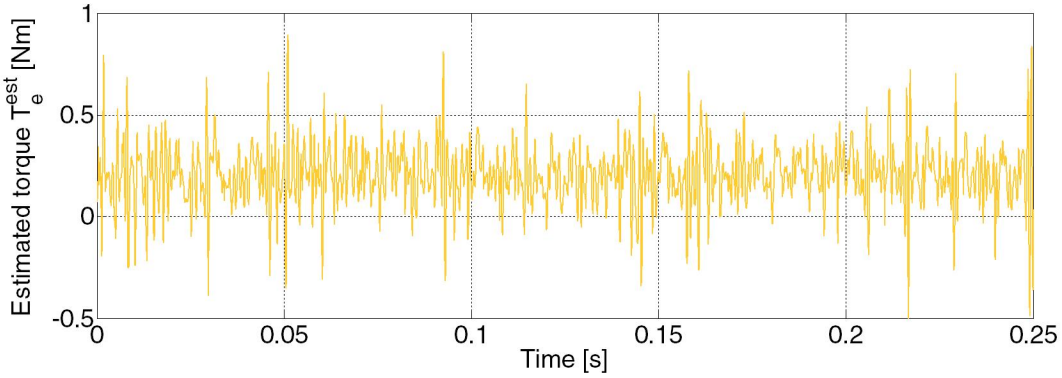


Figure 7.4: The estimated torque T_e^{est} (ω_m^{ref}=500 rpm, without repetitive controller)

7.3 Effectiveness of Torque Ripple Reduction

A host of experimental tests have been done to verify the effectiveness of the proposed torque ripple reduction method under four operating conditions in the following four subsections. All the tests are carried out under speed control mode.

The **integer delay condition** in 7.3.1 indicates the condition when the sampling frequency can be evenly divided by the rotor speed. Since the deadbeat current control loop has been tuned, and the mechanical parameters required for the torque estimator are acquired from pre-measurements, the experimental condition in 7.3.1 is close to the ideal condition defined in Chapter 6.2.

Section 7.3.2 targets the experimental validation of the proposed interpolations for solving the **fractional delay** issue, of which the simulative results have been given in 6.3.3. The fractional delay issue has also been defined in 6.3.3.

Sections 7.3.3 and 7.3.4 are the experimental validations of the simulative results given in 6.3.4. The behaviour of the proposed control system during **speed and load transients** is investigated.

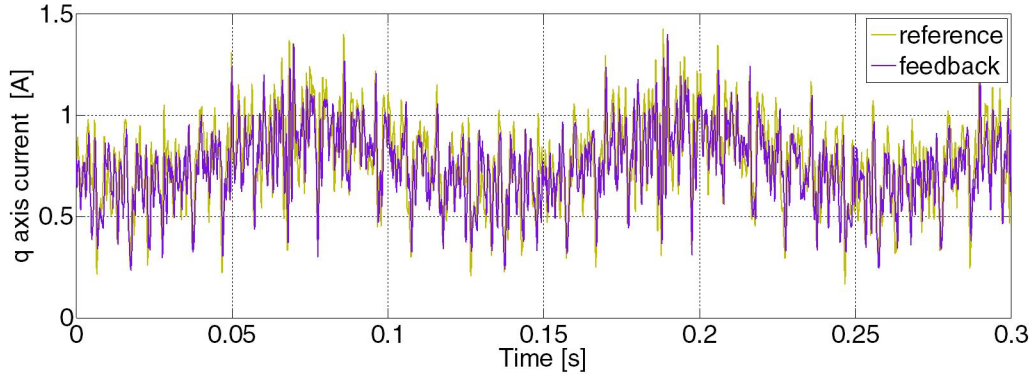
The control parameters for RC in the tests can be found in Table. 6.1.

7.3.1 Effectiveness under Integer Delay Condition

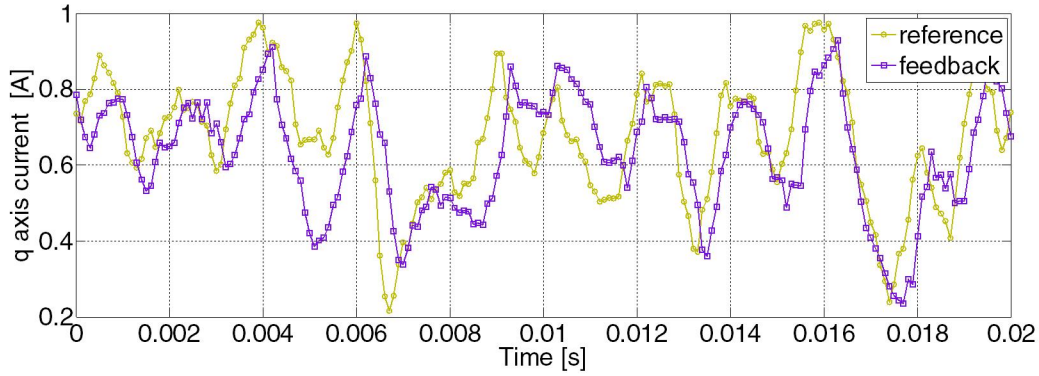
Since the sampling frequency is 10 kHz, the reference rotor speed is set to be 500 rpm (500/60=8.3 Hz). Therefore the speed waveform of a full mechanical rotation can be recorded by the controller using integer numbers of delays (10k/8.3=1200). Such condition is recognized as **integer delay condition** in this work. The ripple reduction effect of the proposed control is shown in Fig.7.5 for the case of 500 rpm. The test has been repeated for 80 rpm, 100 rpm, 200 rpm, 400 rpm. The FFT results of the rotor speed in each case are compared in Fig.7.6.

The current response plot in Fig.7.5(b) proves the normal operation of the DBCC. As can be seen, the feedback current (purple) is able to reach the reference current (yellow) after $2Ts$ (0.0002s). Fig.7.5(c) compares the rotor speed ω_m (calculated with the 9^{th} order FIR filter) with and without the repetitive controller when the motor is commanded to rotate at constant 500 rpm. As it can be seen, the speed ripple before applying the proposed RC is within $\pm$7.4 rpm, whereas $\pm$2.3 rpm after, which is a peak to peak value reduction from 2.96% to 0.92%.

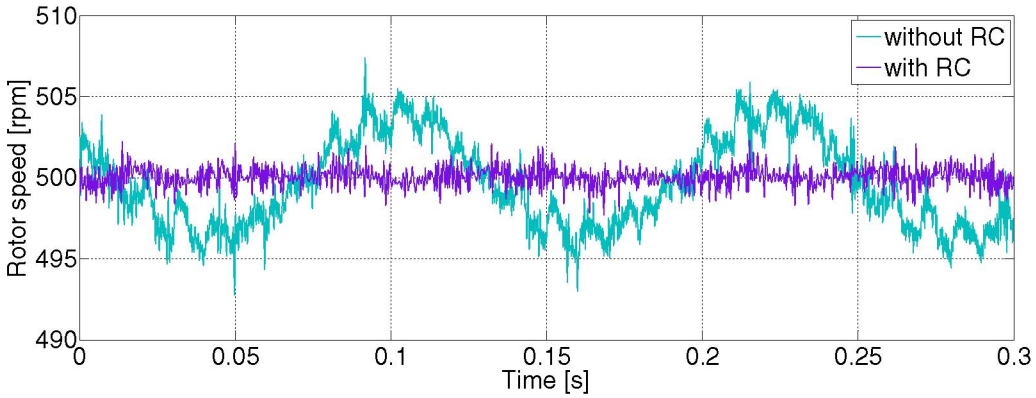
Importantly, the effectiveness of ripple reduction under integer delay condition is confirmed at a range of speed from 80 rpm to 500 rpm. As can be seen from the FFT analysis results shown in Fig.7.6, there is a significant reduction in speed ripple before and after applying the proposed method. Besides, Fig.7.6 also confirms the fact that torque ripple is angle-based and mainly varies with rotor position. Therefore, no matter how the speed command changes, the frequency of speed harmonics changes proportionally, and the order of main harmonics are fixed to be the 10^{th}, 12^{th}, 20^{th}, 24^{th}, 30^{th}, 36^{th}, 48^{th}, 60^{th} for all the speed. Where, the 12^{th} and its multiple order harmonics are mainly due to the cogging effect, since the overall number of slots and teeth is 12.



(a) q axis current response with RC

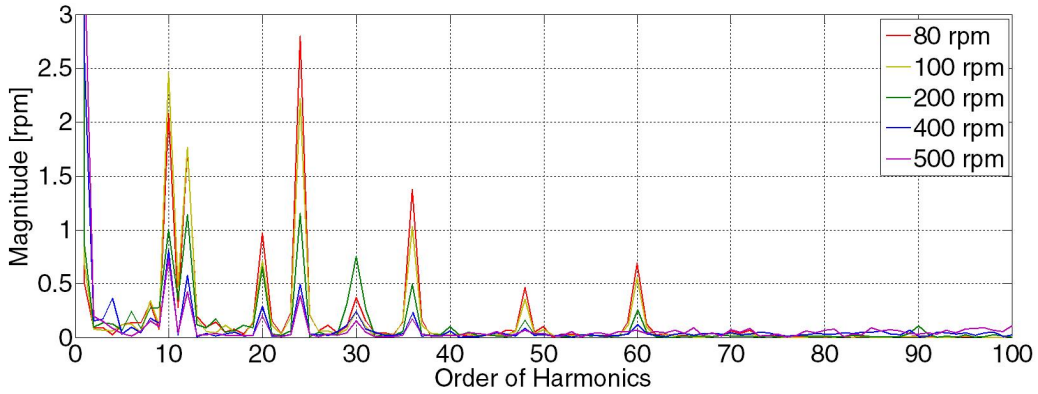


(b) Zoom in of (a)

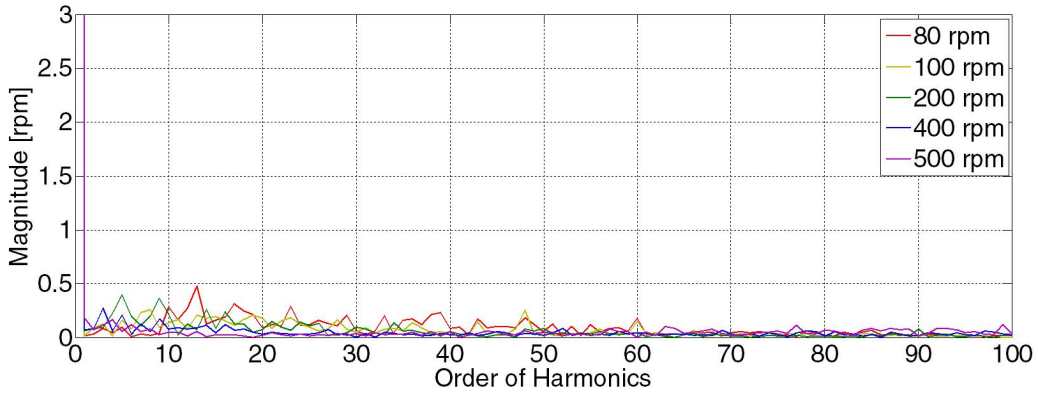


(c) Rotor speed with and without RC

Figure 7.5: Effectiveness of the proposed control at 500 rpm (integer delay condition)



(a) Without the proposed RC

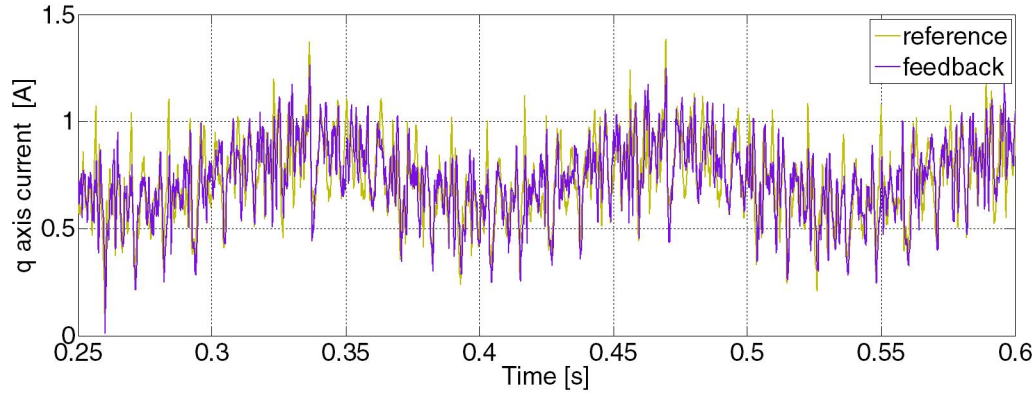


(b) With the proposed RC

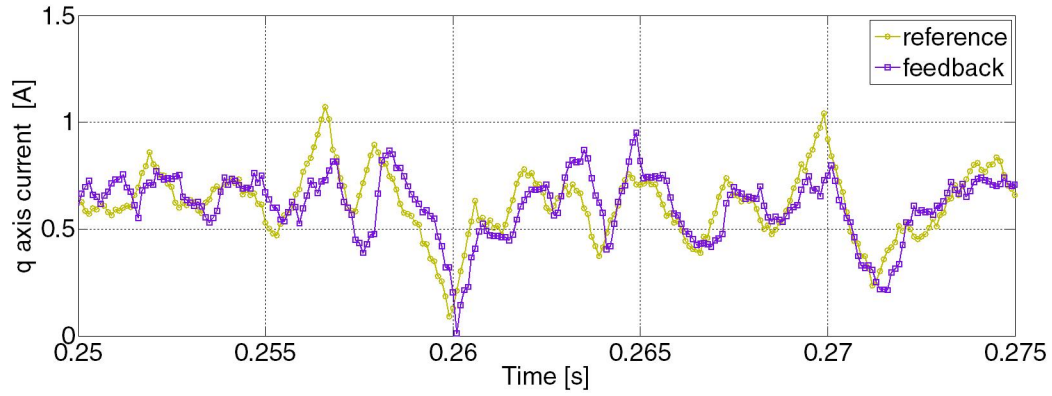
Figure 7.6: FFT of rotor speed under different speed (integer delay condition)

7.3.2 Effectiveness under Fractional Delay Condition

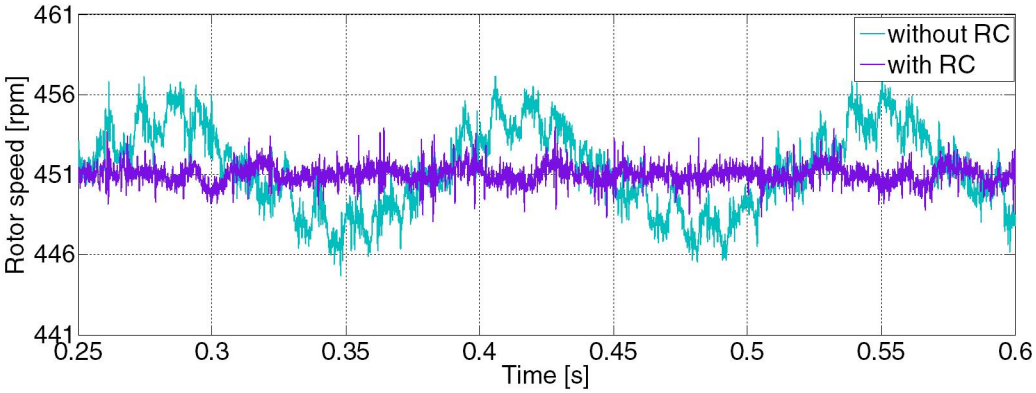
The same speed control test has been carried out for the fractional delay condition (i.e. the condition when the sampling frequency is not integer multiple of the rotor speed frequency, as has been explained in Section 6.3.3), where the speed reference is set to be 123 rpm, 214 rpm, 300 rpm, 349 rpm, 451 rpm. The ripple reduction effect of the proposed control is shown in Fig.7.7 for the case of 451 rpm. The FFT results of the rotor speed at the other speed are compared in Fig.7.8.



(a) q axis current response with RC

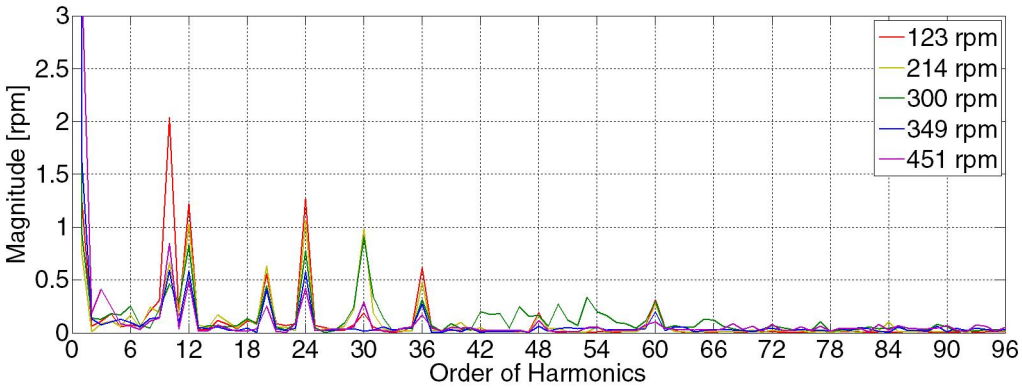


(b) Zoom in of (a)

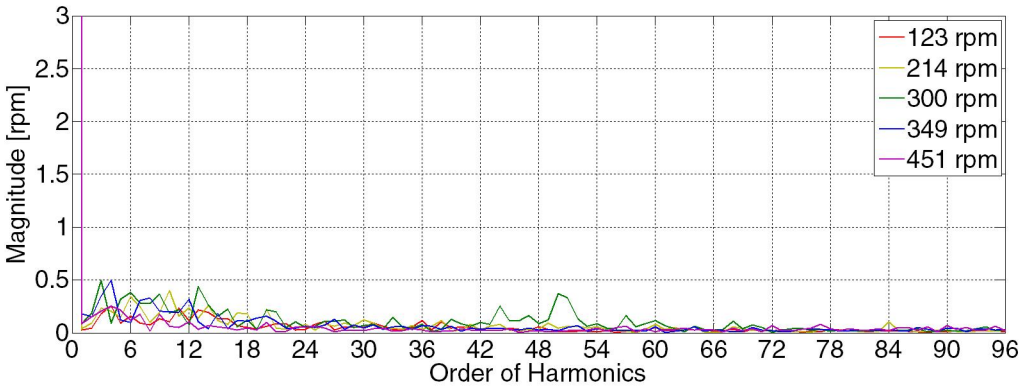


(c) Rotor speed with and without RC

Figure 7.7: Effectiveness of the proposed control at 451 rpm (fractional delay condition)



(a) Without the proposed RC



(b) With the proposed RC

Figure 7.8: FFT of rotor speed under different speed (fractional delay condition)

Similar to Fig.7.5(b), Fig.7.7(b) also proves the normal operation of the deadbeat current loop, since the feedback current (purple) is able to track the reference after a $2T_s$ delay. Fig.7.7(c) shows the rotor speed (calculated with the 9^{th} order FIR filter) before and after applying the proposed RC. It shows that the speed ripple is reduced from ± 6.3 rpm to ± 2.9 rpm.

As can be seen from Fig.7.8, the 10^{th}, 12^{th}, 20^{th}, 24^{th}, 30^{th}, 36^{th}, 48^{th}, and 60^{th} harmonics are still the main ripple components in the speed, which matches the results in Fig.7.6. All the ripple of these frequencies is successfully removed by the proposed RC as shown in Fig.7.8(b).

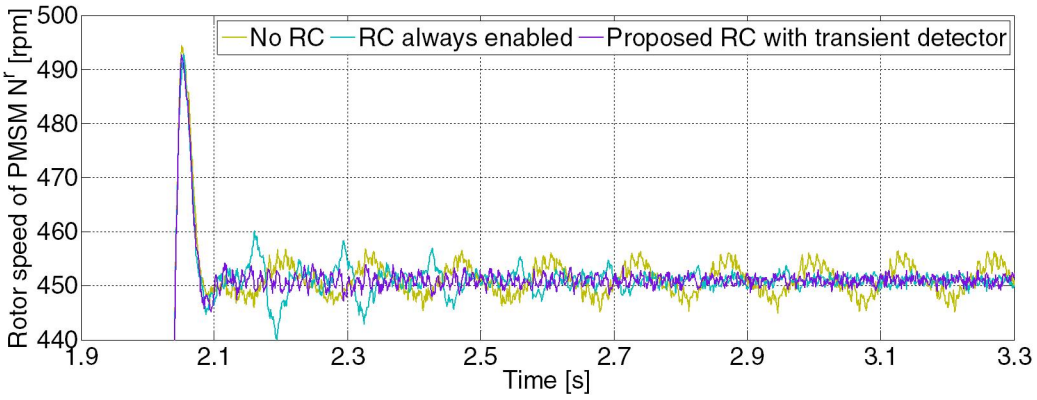
Generally, the results for fractional delay condition are comparable with the results for integer delay condition shown in Fig.7.5 and Fig.7.6.

7.3.3 Effectiveness during Speed Transient

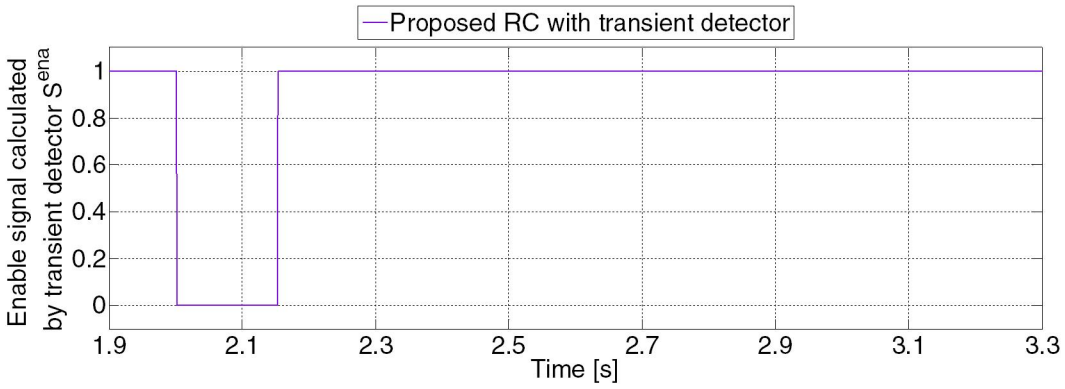
Moving now to the transient behaviour of the proposed repetitive controller. Two types of transients are considered in this work. One type is the speed step transient, of which the results are shown in this subsection. The other type is the load step transient, of which the results will be shown in the next subsection.

For the speed step test, no load is applied. The speed reference steps from 123 rpm to 451 rpm is commanded at 2s. The resultant rotor speed waveform can be seen from 7.9(a). The speed with the proposed repetitive controller (purple), and with the RC but without the proposed transient detector (blue), in which way the RC is always activated, are compared. In both cases, the speed ripple is successfully reduced after around 7 full rotations (counted from Fig.7.9(c)), however, during the first a few revolution of rotations, the purple waveform is considerably smoother and converges faster than the blue waveform.

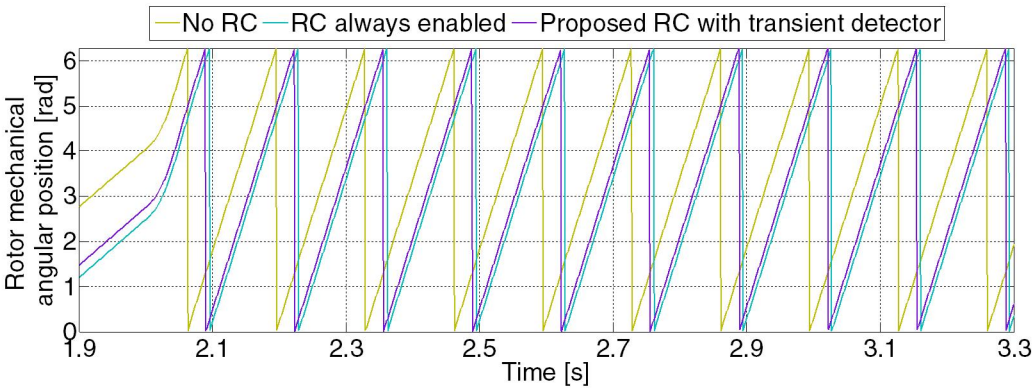
Overall, the results in Fig.7.9 confirm the effectiveness of the proposed transient detector. During the speed transient, the transient detector is able to correctly disable the learning capability of the proposed repetitive controller, while maintaining the ripple reduction effect according what has been previously learned at steady state. When the system reaches the new steady state, the transient detector is able to correctly enable the learning capability, therefore, the proposed RC can update its memory, and thus gradually improves the ripple reduction effect.



(a)



(b)



(c)

Figure 7.9: Effectiveness of the proposed control during a speed transient at 2s

7.3.4 Effectiveness during Load Transient

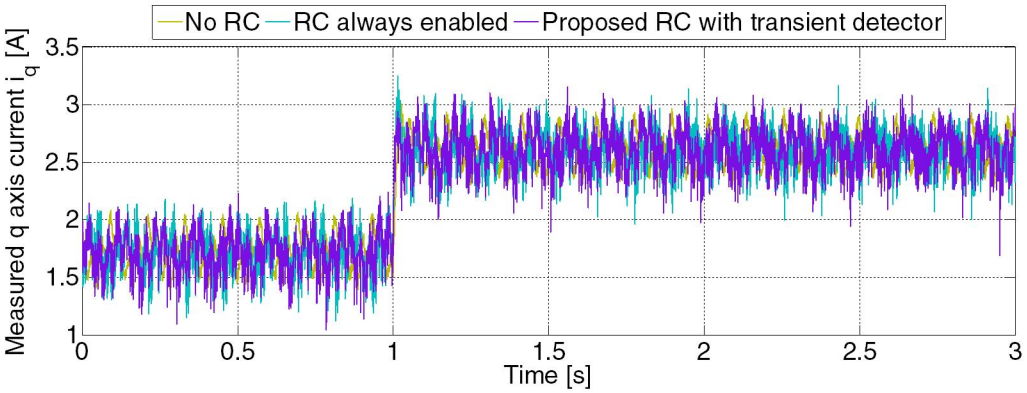
For the purpose of investigating the behaviour of the proposed transient detector during load transients, the motor is still controlled under speed control mode (499 rpm), whereas another load side PMSM shown in Fig.7.2 is controlled under torque control mode. While the motor is spinning, a step change in the load torque is applied at 1s. The corresponding responses are recorded as shown in Fig.7.10.

Fig.7.10(a) shows the measured q axis current i_q is around 1.7A before the load change, and around 2.6A after. This indicates the successful implementation of a load torque change.

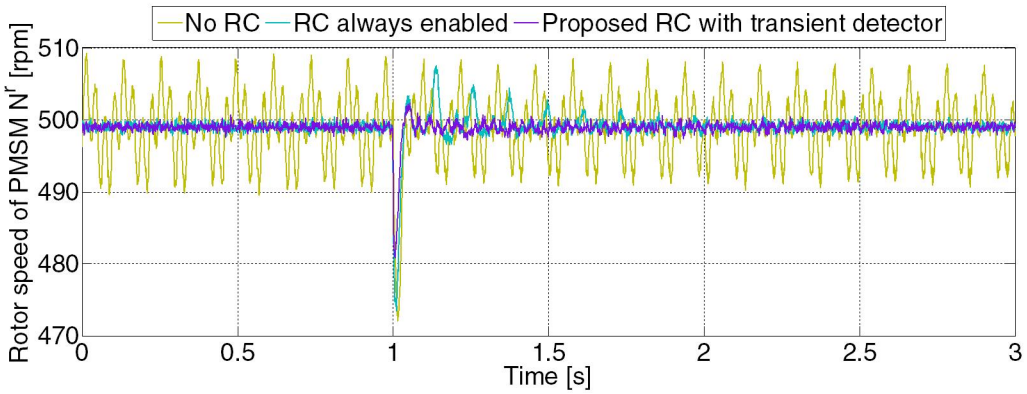
Fig.7.10(b) compares the rotor speed (purple) when with the proposed RC including the transient detector, and the rotor speed (blue) when with RC but excluding the transient detector (i.e. RC always on). Similar behaviours are found when compared with the results acquired for the speed change (Fig.7.9(a)). After the transient, the blue waveform and the purple waveform becomes comparable eventually. However, it takes around 7 full mechanical rotations for the blue line to reach the new steady state, while takes only 2 rotations for the purple line. Less overshoots and spikes are also the adding benefits when looking at the purple waveform.

Fig.7.10(c) proves the correct detection of the load change, as the enable signal S^{ena} calculated by the transient detector drops from 1 to 0 at the transient, and steps back from 0 to 1 when a new steady state has been reached.

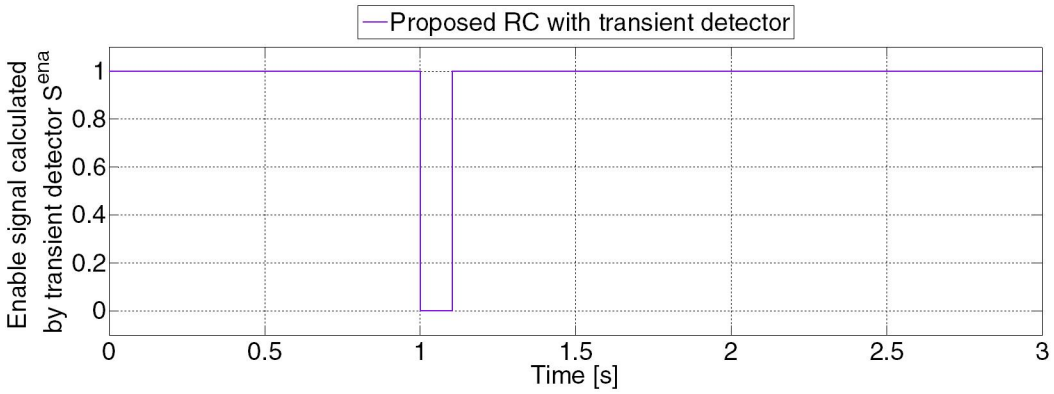
Overall, the results confirm the successfully implementation of the proposed transient detector during a load change. Through the usage of such transient detector, smoother rotor speed and faster ripple rejection can be achieved.



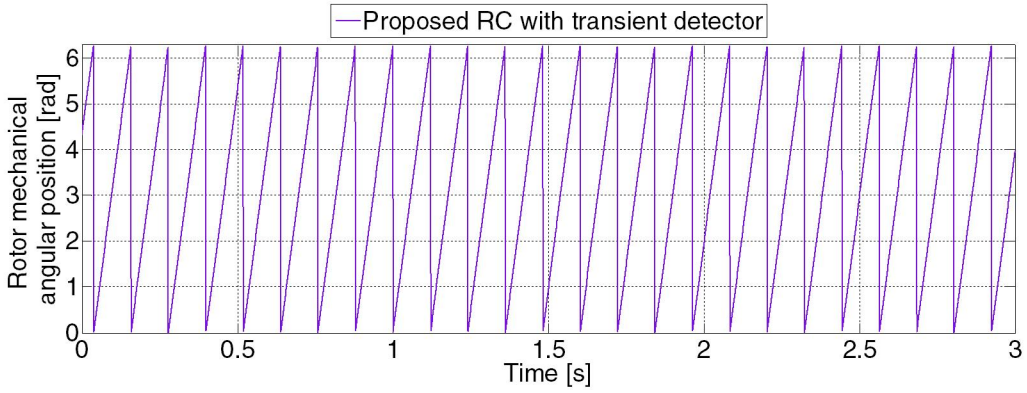
(a)



(b)



(c)



(d)

Figure 7.10: Effectiveness of the proposed control during a load transient at 1s

7.4 Conclusions

This chapter targets at the experimental implementation and validation of the proposed angle-based repetitive control. The layout and the key findings from each part in this chapter can be summarized using the chat below:

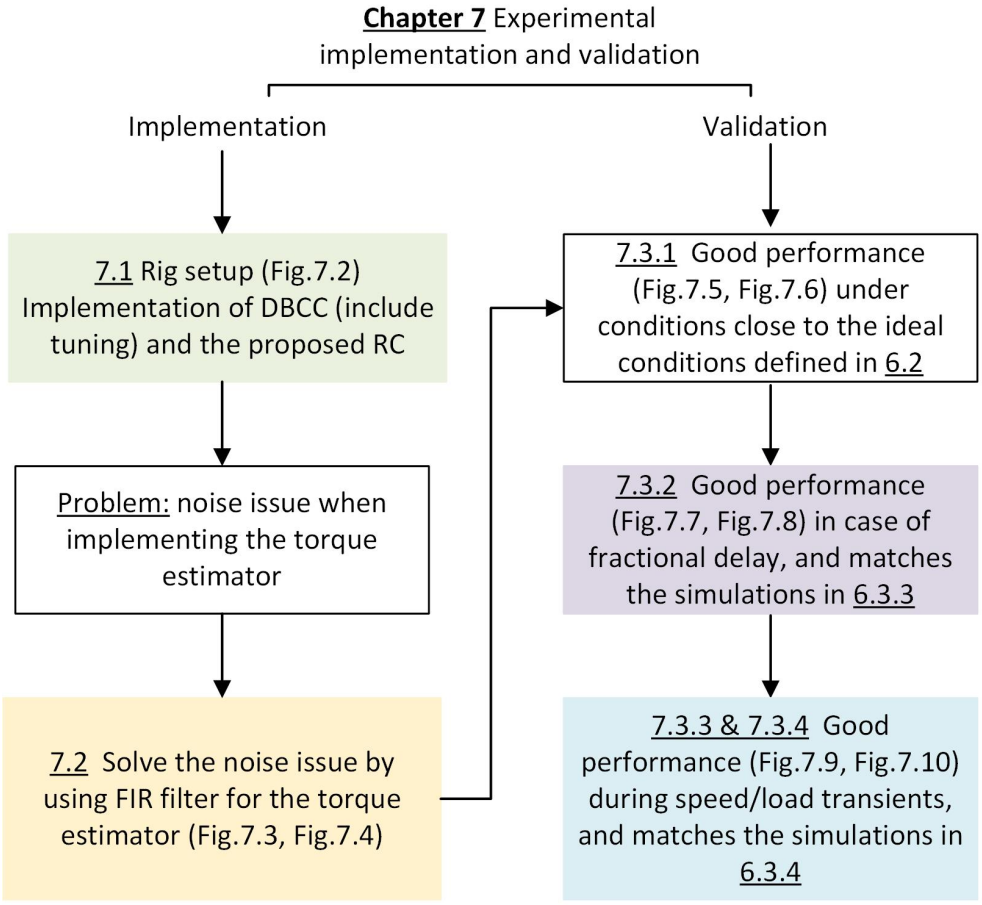


Figure 7.11: Key results and findings in Chapter 7

In summary, this chapter has shown that **the proposed angle-based repetitive controller is able to reduce torque ripple effectively at constant speed, at variable speed/load, and even during speed/load transients.** It is also worth emphasizing **six advanced features** of the proposed method as follow:

1) The proposed control method can **work at different constant speeds without changing any control parameters**. All the tests carried out in this chapter are using the same control parameters.

2) The proposed control method can **achieve comparable performance under**

fractional delay condition (Fig.7.7 and Fig.7.8), as the integer delay condition (Fig.7.5 and Fig.7.6). This feature is an important point which allows the proposed method to cope with variable speed/frequency.

3) Another important design for variable speed operation is to design RC in angle-domain. Fig.7.6 and Fig.7.8 also confirms that the main ripple is angle-based and, therefore, it is necessary to **compensate such ripple based on the rotor position**.

4) By solving the fractional delay issue using linear interpolations in time to angle, angle to time conversions, and through the use of angle-based RC instead of conventional time-based, the proposed control method is able to **maintain high performance under a wide range of variable speed (frequency)**. The result in Fig.7.9 confirms this.

5) An additional advantage of the proposed method is its transient performance. Normally, conventional RC is only activated at steady state. However, with the proposed transient detector, the proposed RC is able to maintain high performance even during speed/load transients. Fig.7.9 and Fig.7.10 prove that with transient detector, the proposed control can **achieve a smoother transient behaviour and converges faster** than when without.

6) What is more, as mentioned in the experimental rig setup (section 7.1), the inverter non-linearities compensation (as discussed in 4.1.4) is removed. In the experimental tests of the DBCC loop (section 4.2), we know that the steady state error of the current response can be very sensitive to the inverter non-linearities, and therefore it has been compensated. However, with the proposed RC, and under speed control mode, RC is able to learn and compensate the steady state error. Hence, there is **no need to compensate the inverter non-linearities** with extra lookup table.

Chapter 8

Conclusions

The aim of developing a novel torque ripple reduction scheme for a.c. permanent magnet servo motor drives has been achieved. The overall control topology can be seen from Fig.5.12. Details about the proposed angle-based repetitive controller is shown in Fig.5.7, and details about the proposed deadbeat current controller is shown in Fig.4.2.

A map of this work is drawn in Fig.8.1, where the relationships between chapters are shown.

The following section 8.1 will further explain this map and concludes the main contributions of this work. Section 8.2 will discuss the limitations of this work and provide guidelines for further research in this direction.

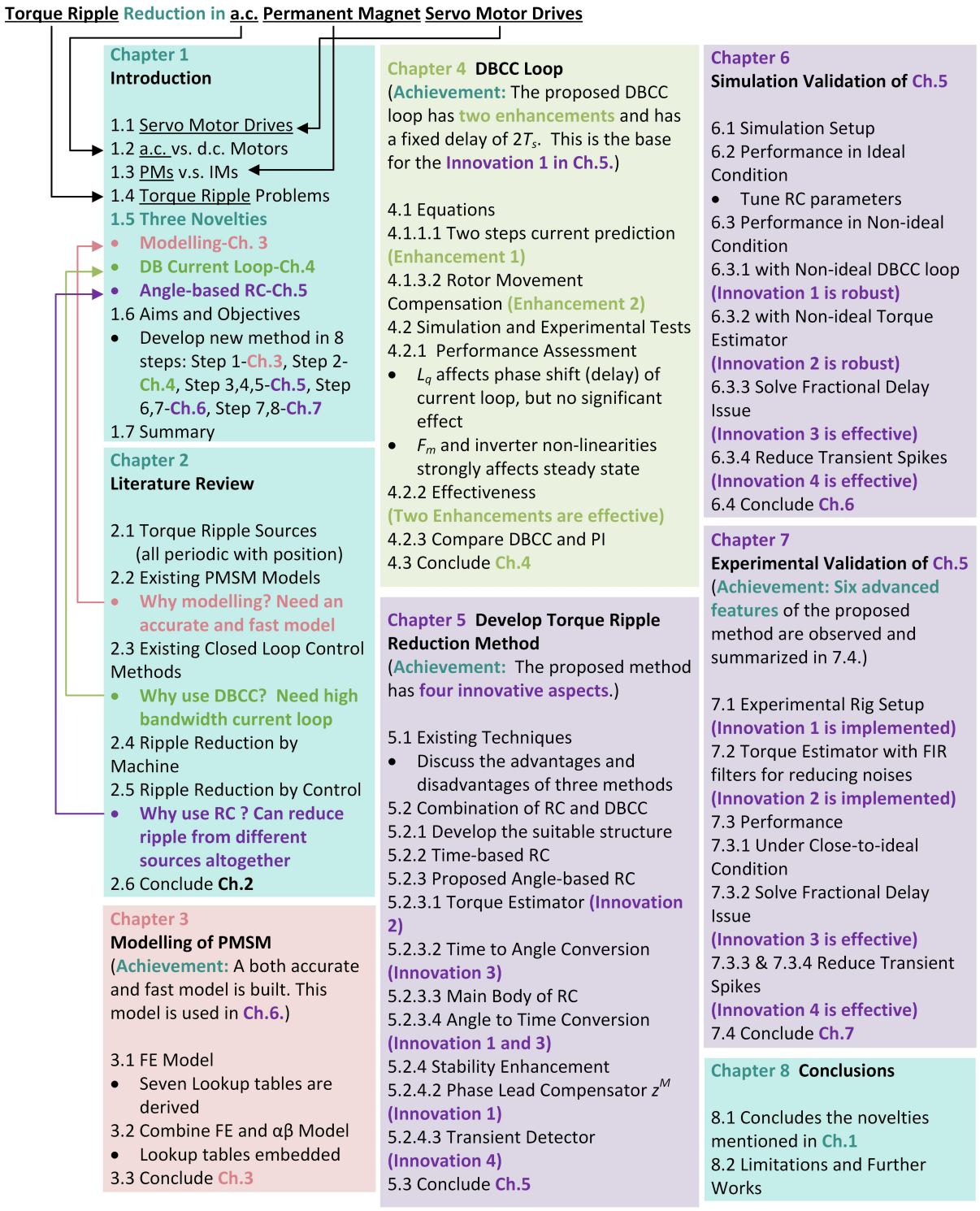


Figure 8.1: Map of this work

8.1 Conclusions

In conclusion, three novelties (as aforementioned in section 1.5) are included in this work. Based on the results of simulation and experimental tests, we can now conclude these **three novelties** as follow:

***Novelty 1:* Modelling of PMSM** (Chapter 3)

Section 2.2 has explained that the existing models for PMSM are either intended for high accuracy or fast computation. Chapter 3 has proposed a PMSM model combining accuracy and reduced simulation times which are typical of finite element (FE) and $\alpha\beta$ frame models. In details:

1) seven novel lookup tables have been derived from the FE analysis results (Section 3.1);

2) the dynamic equations of PMSM in $\alpha\beta$ coordinates have been modified to embed these lookup tables (Section 3.2).

Lookup tables are commonly used for PMSM modelling in literature, however, the way these lookup tables are derived and embedded is novel. Simulation results in 3.2 show that the proposed model agrees with the FE model under the test conditions. Therefore, **accuracy and fast computation can be achieved at the same time**. This model is used for the simulation tests in Chapter 6.

***Novelty 2:* Enhanced deadbeat current control** (Chapter 4)

From the review of the existing closed loop control methods in 2.3 and the existing torque ripple reduction control methods in 2.5, we understand that torque ripple reduction requires high bandwidth current control loop, and DBCC is one of the candidates. The DBCC proposed in Chapter 4 includes **two enhancements**:

1) **two steps current prediction** is used instead of the traditional one step (Section 4.1.1.1). Experimental results in 4.2.2.1 confirm that the two steps current

prediction can effectively improve the accuracy of current prediction.

2) the voltage offset error caused by the rotor movement has been identified in this work, a novel **rotor movement compensation** scheme has been proposed (Section 4.1.3). Experimental results in 4.2.2.2 confirm that the compensation can reduce current steady state error at high speed.

Besides, a very detailed **performance assessment** has been done for the proposed DBCC loop, which has not been done in existing literature. The key findings can be concluded as:

1) The phase shift or delay of the current response is the most sensitive to the variation of the q axis inductance L_q. However, the impact is small. Results in 4.2.1.1 show that the increase or decrease in the current loop delay will be within one T_s, when the L_q^{est} in controller varies from 50% to 150% of the real L_q in machine. **This fixed delay ($2T_s$) nature of DBCC loop set a good base for the further design of the proposed repetitive controller (see later in Novelty 3)**.

2) The steady state error of the current response strongly depends on the correctness of magnetic flux F_m and inverter non-linearities (dead time) (Section 4.2.1.2) .

All these achievements have been published on IET Power Electronics journal in 2016 [83] and the 9^{th} International Conference on Power Electronics and ECCE Asia in 2015 [84].

***Novelty 3:* Proposed angle-based repetitive control (RC)** (Chapter 5)

RC is chosen for this work due to its possibility of reducing ripple from all sources together (Section 2.5 and 5.1). The proposed angle-based RC contains **four innovative aspects**:

1) taking advantage of the $2T_s$ delay nature of DBCC loop, the fixed delay can be easily compensated by a simple phase lead compensator (introduced in 5.1.3). Simulations in 5.2.4.2 show that instability can occur if the delay of DBCC has not been properly compensated. This work has implemented the **phase lead compensator in the**

angle domain through using the future position prediction as in 5.2.3.4. Simulation results in 6.3.1 and experimental results in 7.3 indirectly prove the effectiveness of this compensator, since stable results are achieved.

2) since the delay of the mechanical system is not fixed (varies with frequency), therefore it cannot be compensated by the aforementioned phase lead compensator. A **torque estimator** has been proposed in 5.2.3.1, in order to compensate this delay outside of the RC. Simulations in 6.3.2 prove the robustness against mechanical parameter variations. Also, it has been successfully implemented for the experimental tests in 7.2 with FIR filters included, in order to solve noise issue in the position feedback.

3) the fractional delay issue of conventional time-based RC has been addressed in 6.3.3. Simulation results in 6.3.3 and experimental results in 7.3.2 confirm that the proposed **linear interpolations in time to angle/angle to time conversions** (proposed in 5.2.3.2 and 5.2.3.4 respectively) can solve the fractional delay issue. This is an important feature that allows the proposed control to be used for variable speed/frequency drives.

4) a **transient detector** has been proposed for stability reasons in 5.2.4.3. Its main function is to disable the learning capability of RC during transient, and automatically enable it when a new steady state has been reached. Simulation results in 6.3.4, experimental results in 7.3.3 and 7.3.4 confirm that smoother transient speed and faster ripple reduction are achieved with the help of this transient detector.

The simulation results have been published in the 8^{th} IET International Conference on Power Electronics, Machines and Drives (PEMD 2016) [85]. The experimental results are in submission to Transaction on Industry Application. (Appendix D)

Six advanced features of the overall proposed control system have been concluded in the conclusions of Chapter 7 (section 7.4). Overall, the torque ripple reduction scheme is able to reduce torque ripple effectively at constant speed, at variable speed/load, and even during speed/load transients. The proposed RC can also compensate inverter

non-linearities (Good experimental results are achieved in Chapter 7 without compensating inverter non-linearities).

A full list of **published papers** are given in Appendix D.

8.2 Limitations and Further Works

A new torque ripple reduction scheme for variable speed PMSM servo drives has been proposed in this work, which has also been validated with simulative and experimental tests. However, it is worth clarifying the test conditions to understand the limitations of the tests. Some limitations and further works are summarized as follow:

Limitation 1: The effectiveness of the proposed method under field weakening region is not investigated.

For investigating this, more FE analysis results for all possible combinations of dq axis currents in field weakening region need to be collected from the FE model presented in 3.1, and therefore, more comprehensive lookup tables can be derived. A challenge of using the proposed method under this region is the parameter variation of machine. Since electrical parameters such as inductances may decrease significantly due to the saturation of coil, such variation need to be carefully considered in the tuning of the proposed deadbeat current controller and repetitive controller. Otherwise, it may be a risk for the stability of the overall control system.

Further work 1: Accomplishing the proposed PMSM model for field weakening region. Assessment of the proposed RC under field weakening region.

Limitation 2: The tuning process of the proposed repetitive controller is not developed.

The tests in this work are carried out with the control gain and forgetting factor of the repetitive controller tuned by experience. Although the Nyquist stability criterion (i.e. the closed loop poles should be located inside the unity circle) has been used to guide

the choice, more automatic tuning process can be developed.

Further work 2: Development of the automatic tuning method for the proposed RC.

Limitation 3: The interaction between the proposed torque ripple reduction scheme and the mechanical resonance is not investigated.

Once investigated, the cut-off band of the FIR filters used during the torque estimation (as presented in 7.2) can be designed more wisely.

Further work 3: Investigation of the interaction between the proposed RC and the mechanical resonance.

Limitation 4: The interaction between the proposed repetitive controller and the speed loop PI regulation is not fully investigated.

As mentioned in 5.2.1, the plug-in structure of RC was chosen among the three possible structures(direct, parallel, plug-in) for the least interaction between the speed loop PI regulator and the repetitive controller. However, still, there would be an interaction during the transients, since both the RC and PI would respond the transient error and may "fight" for the d.c. part of torque error. In fact, the transient detector was proposed for this purpose (in 5.2.4.3), which can effectively reduce the interaction between them by blocking the input of RC during transients. Alternatively, the torque estimator can be improved to achieve faster load torque detection during the load transient. Therefore, the load torque error (d.c. part of T_e^{err}) can be removed from the torque error T_e^{err}. In such way, RC can only work to reduce the a.c. part of torque error, and the PI will be responsible for the d.c. part.

Further work 4: Investigation of the interaction between the proposed RC and speed loop PI. Improvement of the tuning of speed loop PI and the design of the proposed torque estimator.

Bibliography

[1] R. a. Markets. Global and china servo system industry report 2016 - forecasts to 2020 for the $10.45 billion market - research and markets. http://www.prnewswire.com/news-releases/global-and-china-servo-system-industry-report-2016.html.

[2] China's servo motor suppliers see light at the end of the tunnel. http://www.gcis.com.cn/china/chinasservomotorsuppliersseelightattheendofthetunnel.htm.

[3] Servo motors and drives market worth 15.92 billion usd by 2022. http://www.marketsandmarkets.com/pressreleases/servo-motors-and-drives.asp.

[4] J. Puranen *et al.*, "Induction motor versus permanent magnet synchronous motor in motion control applications: a comparative study," *Acta Universitatis Lappeenrantaensis*, 2006.

[5] T. A. Lipo, *Vector control and dynamics of AC drives*. Oxford university press, 1996, vol. 41.

[6] I. M. Gottlieb and I. M. Gottlieb, *Practical electric motor handbook*. Newnes, 2003.

[7] M. Jufer, *Electric drives*. ISTE, 2010, ch. 4 Motors, p. 41–85.

[8] J. F. Gieras and M. Wing, *Permanent magnet motor technology: design and applications, Third Edition*. Marcel Dekker, 2010, ch. 5 Permanent Magnet Synchronous Motors, p. 171–214.

[9] T. M. Jahns and W. L. Soong, "Pulsating torque minimization techniques for permanent magnet ac motor drives-a review," *IEEE Transactions on Industrial Electronics*, vol. 43, no. 2, pp. 321–330, 1996.

[10] Torque. https://en.wikipedia.org/wiki/torque.

[11] B. Drury, *The Control Techniques drives and controls handbook*. Institution of Electrical Engineers, 2001, ch. A1.4 A.C. synchronous motors, p. 52–53.

[12] W. Qian, S. K. Panda, and J.-X. Xu, "Torque ripple minimization in pm synchronous motors using iterative learning control," *IEEE Transactions on Power Electronics*, vol. 19, no. 2, pp. 272–279, 2004.

[13] W. Q. Chu and Z. Q. Zhu, "Investigation of torque ripples in permanent magnet synchronous machines with skewing," *IEEE Transactions on Magnetics*, vol. 49, no. 3, pp. 1211–1220, 2013.

[14] B. Drury, *The Control Techniques drives and controls handbook 2nd Edition*. Institution of Electrical Engineers, 2009, ch. A2 Drive converter circuit topologies, p. 71–108.

[15] D. G. Holmes and T. A. Lipo, *Modulation of ThreePhase Voltage Source Inverters*. Wiley-IEEE Press, 2003, pp. 215–258. [Online]. Available: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=5311949

[16] L. C. O. de Oliveira, W. M. Borges, J. B. de Souza, and L. L. Ravagnani, "Dc and ac side harmonics in ccc converters." IEEE, 2007, pp. 1–5.

[17] V. Ruuskanen, J. Nerg, M. Rilla, and J. Pyrhönen, "Iron loss analysis of the permanent-magnet synchronous machine based on finite-element analysis over the electrical vehicle drive cycle," *IEEE Transactions on Industrial Electronics*, vol. 63, no. 7, pp. 4129–4136, 2016.

[18] G. Sawczuk and J. Junak, "Finite element demagnetization analysis of permanent magnet synchronous motor considering magnetization process," 2013, pp. 7–12.

[19] M. van der Geest, H. Polinder, and J. A. Ferreira, "Efficient finite element based rotor loss calculation for permanent magnet synchronous machines," 2014, pp. 1133–1138.

[20] A. Tessarolo, F. Agnolet, F. Luise, and M. Mezzarobba, "Use of time-harmonic finite-element analysis to compute stator winding eddy-current losses due to rotor motion

in surface permanent-magnet machines," *IEEE Transactions on Energy Conversion*, vol. 27, no. 3, pp. 670–679, 2012.

[21] E. Schmidt, M. Kaltenbacher, and A. Wolfschluckner, "Eddy current losses in permanent magnets of permanent magnet synchronous machines - analytical calculation methods and high order finite element analyses," 2014, pp. 1–5.

[22] M. Fitouri, Y. BenSalem, and M. N. Abdelkrim, "Analysis and co-simulation of permanent magnet sychronous motor with short-circuit fault by finite element method," 2016, pp. 472–477.

[23] M. Chaieb, N. B. Hadj, J. K. Kammoun, and R. Neji, "Thermal modeling of permanent magnet motor with finite element method," in *Sciences and Techniques of Automatic Control and Computer Engineering (STA), 2014 15th International Conference on*, 2014, pp. 594–598.

[24] J. K. Tangudu, T. M. Jahns, A. EL-Refaie, and Z. Q. Zhu, "Lumped parameter magnetic circuit model for fractional-slot concentrated-winding interior permanent magnet machines," 2009, pp. 2423–2430.

[25] J.-P. Wang and D. K. Lieu, "Magnetic lumped parameter modeling of rotor eccentricity in brushless permanent-magnet motors," *IEEE Transactions on Magnetics*, vol. 35, no. 5, pp. 4226–4231, 1999.

[26] E. C. Lovelace, T. M. Jahns, and J. H. Lang, "A saturating lumped-parameter model for an interior pm synchronous machine," *IEEE Transactions on Industry Applications*, vol. 38, no. 3, pp. 645–650, 2002.

[27] J. T. Chen, Z. Q. Zhu, and D. Howe, "A dual-lumped parameter magnetic circuit model accounting for the cross-coupling effect, with particular reference to flux-switching permanent magnet machines," 2008, pp. 111–115.

[28] J. T. Chen and Z. Q. Zhu, "Influence of the rotor pole number on optimal parameters in flux-switching pm brushless ac machines by the lumped-parameter magnetic circuit model," *IEEE Transactions on Industry Applications*, vol. 46, no. 4, pp. 1381–1388, 2010.

[29] N. Bracikowski, M. Hecquet, P. Brochet, and S. V. Shirinskii, "Multiphysics modeling of a permanent magnet synchronous machine by using lumped models," *IEEE Transactions on Industrial Electronics*, vol. 59, no. 6, pp. 2426–2437, 2012.

[30] T. Raminosoa, J. Farooq, A. Djerdir, and A. Miraoui, "Reluctance network modelling of surface permanent magnet motor considering iron nonlinearities," *Energy Conversion and Management*, vol. 50, no. 5, pp. 1356–1361, 2009.

[31] H. Roisse, M. Hecquet, and P. Brochet, "Simulations of synchronous machines using a electric-magnetic coupled network model," *IEEE Transactions on magnetics*, vol. 34, no. 5, pp. 3656–3659, 1998.

[32] B. Asghari and V. Dinavahi, "Permeance network based real-time induction machine model," 2009, pp. 1–6.

[33] H. Mirahki and M. Moallem, "Comparison of saturated parameters of three-layer interior permanent magnet synchronous machine obtained from lumped parameter model and finite element method." IEEE, 2013, pp. 1–5.

[34] I. Boldea and S. A. Nasar, *Electric drives Third Edition*. CRC Press, 2016, ch. 11 PM and Reluctance Synchronous Motor Drives, p. 348–365.

[35] N. Q. Dai, R. Dutta, and M. F. Rahman, "Comparative performance analysis of field-oriented control and direct torque control for a fractional-slot concentrated winding interior permanent magnet synchronous machine," 2012, pp. 879–885.

[36] R. Souad and H. Zeroug, "Comparison between direct torque control and vector control of a permanent magnet synchronous motor drive," 2008, pp. 1209–1214.

[37] Vector control (motor). https://en.wikipedia.org/wiki/vector_control_(motor).

[38] T. M. Jahns, "Flux-weakening regime operation of an interior permanent-magnet synchronous motor drive," *IEEE Transactions on Industry Applications*, vol. IA-23, no. 4, pp. 681–689, 1987.

[39] S. M. Sue and C. T. Pan, "Voltage-constraint-tracking-based field-weakening control of ipm synchronous motor drives," *IEEE Transactions on Industrial Electronics*, vol. 55, no. 1, pp. 340–347, 2008.

[40] A. Consoli, G. Scarcella, G. Scelba, and A. Testa, "Steady-state and transient operation of ipmsms under maximum-torque-per-ampere control," *IEEE Transactions on Industry Applications*, vol. 46, no. 1, pp. 121–129, 2010.

[41] F. F. Mazda, *Power electronics handbook*. Elsevier, 1997.

[42] Z. Mynar, L. Vesely, and P. Vaclavek, "Pmsm model predictive control with field-weakening implementation," *IEEE Transactions on Industrial Electronics*, vol. 63, no. 8, pp. 5156–5166, 2016.

[43] F. Morel, X. Lin-Shi, J. M. Retif, B. Allard, and C. Buttay, "A comparative study of predictive current control schemes for a permanent-magnet synchronous machine drive," *IEEE Transactions on Industrial Electronics*, vol. 56, no. 7, pp. 2715–2728, 2009.

[44] S. J. Qin and T. A. Badgwell, "A survey of industrial model predictive control technology," *Control engineering practice*, vol. 11, no. 7, pp. 733–764, 2003.

[45] S. Vazquez, J. I. Leon, L. G. Franquelo, J. Rodriguez, H. A. Young, A. Marquez, and P. Zanchetta, "Model predictive control: A review of its applications in power electronics," *IEEE Industrial Electronics Magazine*, vol. 8, no. 1, pp. 16–31, 2014.

[46] S. Matsutani, T. Zanma, Y. Sumiyoshi, M. Ishida, A. Imura, and M. Fujitsuna, "Optimal control of pmsms using model predictive control with integrator," 2009, pp. 4847–4852.

[47] M. T. Hagan, H. B. Demuth, and O. D. Jesús, "An introduction to the use of neural networks in control systems," *International Journal of Robust and Nonlinear Control*, vol. 12, no. 11, pp. 959–985, 2002.

[48] F. F. M. El-Sousy, "Robust adaptive h$^{\infty}$ position control via a wavelet-neural-network for a dsp-based permanent-magnet synchronous motor servo drive system," *IET Electric Power Applications*, vol. 4, no. 5, pp. 333–347, 2010.

[49] F. M. El-Sousy, "Hybrid h^{∞} -based wavelet-neural-network tracking control for permanent-magnet synchronous motor servo drives," *IEEE Transactions on Industrial Electronics*, vol. 57, no. 9, pp. 3157–3166, 2010.

[50] F.-J. Lin, C.-H. Lin, and P.-H. Shen, "Self-constructing fuzzy neural network speed controller for permanent-magnet synchronous motor drive," *IEEE Transactions on Fuzzy Systems*, vol. 9, no. 5, pp. 751–759, 2001.

[51] F.-J. Lin and C.-H. Lin, "A permanent-magnet synchronous motor servo drive using self-constructing fuzzy neural network controller," *IEEE Transactions on Energy Conversion*, vol. 19, no. 1, pp. 66–72, 2004.

[52] K. D. Young, V. I. Utkin, and U. Ozguner, "A control engineer's guide to sliding mode control," *IEEE transactions on control systems technology*, vol. 7, no. 3, pp. 328–342, 1999.

[53] X. Zhang, L. Sun, K. Zhao, and L. Sun, "Nonlinear speed control for pmsm system using sliding-mode control and disturbance compensation techniques," *IEEE Transactions on Power Electronics*, vol. 28, no. 3, pp. 1358–1365, 2013.

[54] S. Li, M. Zhou, and X. Yu, "Design and implementation of terminal sliding mode control method for pmsm speed regulation system," *IEEE Transactions on Industrial Informatics*, vol. 9, no. 4, pp. 1879–1891, 2013.

[55] B. Sencer and E. Shamoto, "Effective torque ripple compensation in feed drive systems based on the adaptive sliding-mode controller," *IEEE/ASME Transactions on Mechatronics*, vol. 19, no. 6, pp. 1764–1772, 2014.

[56] N. T. T. Vu, D. Y. Yu, H. H. Choi, and J. W. Jung, "T-s fuzzy-model-based sliding-mode control for surface-mounted permanent-magnet synchronous motors considering uncertainties," *IEEE Transactions on Industrial Electronics*, vol. 60, no. 10, pp. 4281–4291, 2013.

[57] M. Sugeno, "An introductory survey of fuzzy control," *Information sciences*, vol. 36, no. 1, pp. 59–83, 1985.

[58] H. Chaoui and P. Sicard, "Adaptive fuzzy logic control of permanent magnet synchronous machines with nonlinear friction," *IEEE Transactions on Industrial Electronics*, vol. 59, no. 2, pp. 1123–1133, 2012.

[59] J. X. Shen, Z. Q. Zhu, D. Howe, and J. M. Buckley, "Fuzzy logic speed control and current-harmonic reduction in permanent-magnet brushless ac drives," *IEE Proceedings - Electric Power Applications*, vol. 152, no. 3, pp. 437–446, 2005.

[60] J. W. Jung, T. H. Kim, and H. H. Choi, "Speed control of a permanent magnet synchronous motor with a torque observer: a fuzzy approach," *IET Control Theory Applications*, vol. 4, no. 12, pp. 2971–2981, 2010.

[61] M. Uchiyama, "Formulation of high-speed motion pattern of a mechanical arm by trial (in japanese)," *Trans. SICE (Soc. Instrum. Control Eng.)*, vol. 4, p. 706–712, 1978.

[62] W. Qian, S. K. Panda, and J.-X. Xu, "Torque ripple minimization in pm synchronous motors using iterative learning control," *IEEE Transactions on Power Electronics*, vol. 19, no. 2, pp. 272–279, 2004.

[63] W. Qian, S. K. Panda, and J. X. Xu, "Speed ripple minimization in pm synchronous motor using iterative learning control," *IEEE Transactions on Energy Conversion*, vol. 20, no. 1, pp. 53–61, 2005.

[64] T. Inoue, M. Nakano, and S. Iwai, "High accuracy control of servomechanism for repeated contouring," *: Proceedings of the 10th Annual Symposium on Incremental Motion Control Systems and Devices*, p. 258–292, 1981.

[65] P. Mattavelli, L. Tubiana, and M. Zigliotto, "Torque-ripple reduction in pm synchronous motor drives using repetitive current control," *IEEE Transactions on Power Electronics*, vol. 20, no. 6, pp. 1423–1431, 2005.

[66] Y. Wang, F. Gao, and F. J. Doyle, "Survey on iterative learning control, repetitive control, and run-to-run control," *Journal of Process Control*, vol. 19, no. 10, pp. 1589–1600, 2009.

[67] M. S. Islam, S. Mir, and T. Sebastian, "Issues in reducing the cogging torque of mass-produced permanent-magnet brushless dc motor," *IEEE Transactions on Industry Applications*, vol. 40, no. 3, pp. 813–820, 2004.

[68] S. Bolognani, L. Tubiana, and M. Zigliotto, "Sensorless control of pm synchronous motors with non-sinusoidal back emf for home appliance," in *Electric Machines and Drives Conference, 2003. IEMDC'03. IEEE International*, vol. 3. IEEE, 2003, pp. 1882–1888.

[69] J. Holtz and L. Springob, "Identification and compensation of torque ripple in high-precision permanent magnet motor drives," *IEEE Transactions on Industrial Electronics*, vol. 43, no. 2, pp. 309–320, 1996.

[70] Y. A. R. I. Mohamed and E. F. El-Saadany, "A current control scheme with an adaptive internal model for torque ripple minimization and robust current regulation in pmsm drive systems," *IEEE Transactions on Energy Conversion*, vol. 23, no. 1, pp. 92–100, 2008.

[71] J. Richter and M. Doppelbauer, "Control and mitigation of current harmonics in inverter-fed permanent magnet synchronous machines with non-linear magnetics," *IET Power Electronics*, vol. 9, no. 10, pp. 2019–2026, 2016.

[72] G. Heins, M. Thiele, and T. Brown, "Accurate torque ripple measurement for pmsm," *IEEE Transactions on Instrumentation and Measurement*, vol. 60, no. 12, pp. 3868–3874, 2011.

[73] Y. Yuan, F. Auger, L. Loron, S. Moisy, and M. Hubert, "Torque ripple reduction in permanent magnet synchronous machines using angle-based iterative learning control," in *IECON 2012 - 38th Annual Conference on IEEE Industrial Electronics Society*, 2012, pp. 2518–2523.

[74] J.-W. Choi, S.-S. Lee, S.-Y. Yu, and S.-J. Jang, "Novel periodic torque ripple compensation scheme in vector controlled ac motor drives," *Applied Power Electronics Conference and Exposition, 1998. APEC '98. Conference Proceedings 1998., Thirteenth Annual*, vol. 1, pp. 81–85 vol.1, 1998.

[75] H.-T. Moon, H.-S. Kim, and M.-J. Youn, "A discrete-time predictive current control for pmsm," *IEEE Transactions on Power Electronics*, vol. 18, no. 1, pp. 464–472, 2003.

[76] A. Gaeta, P. Zanchetta, F. Tinazzi, and M. Zigliotto, "Advanced self-commissioning and feed-forward compensation of inverter non-linearities," in *Industrial Technology (ICIT), 2015 IEEE International Conference on*, 2015, pp. 610–616.

[77] L. Springob and J. Holtz, "High-bandwidth current control for torque-ripple compensation in pm synchronous machines," *IEEE Transactions on Industrial Electronics*, vol. 45, no. 5, pp. 713–721, 1998.

[78] B. Zhang, D. Wang, K. Zhou, and Y. Wang, "Linear phase lead compensation repetitive control of a cvcf pwm inverter," *IEEE Transactions on Industrial Electronics*, vol. 55, no. 4, pp. 1595–1602, 2008.

[79] C. Ji, "High performance resonant pulsed power supply for radio frequency applications," Ph.D. dissertation, 2012.

[80] P. Gohari. Lecture 6: Discrete equivalents. http://web.cecs.pdx.edu/~tymerski/ece452/6.pdf.

[81] G. Escobar, G. A. Catzin-Contreras, and M. J. Lopez-Sanchez, "Compensation of variable fractional delays in the 6k+-1 repetitive controller," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 10, pp. 6448–6456, 2015.

[82] S. M. Kuo, B. H. Lee, and W. Tian, *Real-time digital signal processing: fundamentals, implementations and applications*. John Wiley & Sons, 2013.

[83] M. Tang, A. Gaeta, A. Formentini, K. Ohyama, P. Zanchetta, and G. Asher, "Enhanced dbcc for high-speed permanent magnet synchronous motor drives," *IET Power Electronics*, 2016.

[84] M. Tang, A. Gaeta, K. Ohyama, P. Zanchehtta, and G. Asher, "Assessments of dead beat current control for high speed permanent magnet synchronous motor drives,"

in *2015 9th International Conference on Power Electronics and ECCE Asia (ICPE-ECCE Asia)*. IEEE, 2015, pp. 1867–1874.

[85] M. Tang, A. Gaeta, A. Formentini, and P. Zanchetta, "A variable frequency angle-based repetitive control for torque ripple reduction in pmsms," in *8th IET International Conference on Power Electronics, Machines and Drives (PEMD 2016)*, April 2016, pp. 1–6.

Appendix A

Fourier Series

Fourier series is used for expressing a waveform in terms of its frequency components. By using the Fourier series, a waveform can be split into a series of sine and cosine waveforms from zero frequency to high frequency.

For example, considering a square wave $W(t)$ (which is a function of variable t) with a period of T, and amplitude of A_0, as shown below:



Figure A.1: An example square wave

The Fourier series representation of $W(t)$ can be seen from (A.1), where $\omega = \frac{2\pi}{T}$, $n = 0, \pm 1, \pm 2, \pm 3, ...$

$$W(t) = \sum_{n=-\infty}^{\infty} c_n e^{jn\omega t}$$

$$\text{where, } c_n = \frac{1}{T}\int_{\frac{-T}{2}}^{\frac{T}{2}} W(t)e^{-jn\omega t}dt = -\frac{1}{T}\int_{\frac{-T}{2}}^{0} Ae^{-jn\omega t}dt + \frac{1}{T}\int_{0}^{\frac{T}{2}} Ae^{-jn\omega t}dt \tag{A.1}$$

$$= j\frac{A}{n\pi}(cosn\pi - 1) = \begin{cases} -\frac{A}{n\pi}(1 - cosn\pi) \text{ when } n \text{ is an odd number} \\ 0 \text{ when } n \text{ is an even number} \end{cases}$$

(A.2) can be acquired by solving (A.1). As can be seen, the square wave only contains odd harmonics.

$$W(t) = \frac{A}{n\pi}(sin\omega t + \frac{1}{3}sin3\omega t + \frac{1}{5}sin5\omega t + ... + \frac{1}{n}sinn\omega t + ...) \tag{A.2}$$

Appendix B

Clarke Transformation and Park Transformation

Many reference frames have been developed for representing the electrical quantities (current, voltage, flux, etc) in different ways. Taking the stator three-phase currents i_a, i_b, i_c of a PMSM for example.

Fig.B.1 shows three commonly used frames for a PMSM:

The abc frame is a static frame fixed with stator, i_a, i_b, i_c denote the three-phase stator currents with their phases lagging 120° from each other, and their amplitudes vary sinusoidally.

Although the three-phase stator currents are physically in three axis, it is actually sufficient to use only two axis for representing the currents. Therefore, the $\alpha\beta$ frame is introduced. In this project, the α axis is chosen to be aligned with the positive a axis, and the β axis is chosen to be 90° lagging the a axis (in between a axis and c axis). The equation for converting i_{abc} into $i_{\alpha\beta}$ is called the **Clarke transformation**. According to (B.1-B.2), the peak value of $i_{\alpha\beta}$ should equal to the peak value of i_{abc}.

The dq frame is the only moving frame among the three. The d axis aligns with the direction of magnetic flux on the rotor. In this project, the q axis is chosen to be

aligned with the α axis at zero position, and the rotor electrical position θ_e equals to the angle between the q axis and α axis. The equation for converting i_{abc} into i_{dq} is called the **Park transformation**. According to (B.3-B.4), the square root of i_{dq} should always equal to the peak value of the three-phase currents.

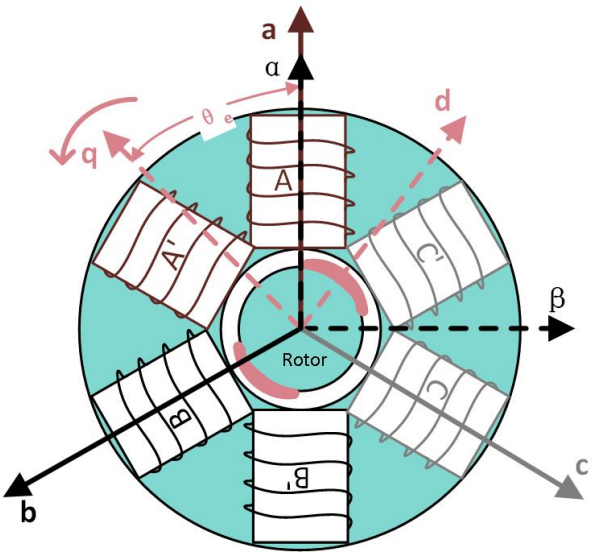


Figure B.1: Three reference frames for a PMSM

Clarke transformation (converting the instantaneous $i_{abc}(t)$ into $i_{\alpha\beta}$):

$$\begin{aligned} i_\alpha &= \frac{2}{3}[i_a(t) - \frac{1}{2}i_b(t) - \frac{1}{2}i_c(t)] \\ i_\beta &= \frac{2}{3}[-\frac{\sqrt{3}}{2}i_b(t) + \frac{\sqrt{3}}{2}i_c(t)] \end{aligned} \tag{B.1}$$

Reverse Clarke transformation (converting $i_{\alpha\beta}$ into the instantaneous $i_{abc}(t)$):

$$\begin{aligned} i_a(t) &= i_\alpha \\ i_b(t) &= -\frac{1}{2}i_\alpha - \frac{\sqrt{3}}{2}i_\beta \\ i_c(t) &= -\frac{1}{2}i_\alpha + \frac{\sqrt{3}}{2}i_\beta \end{aligned} \tag{B.2}$$

Park transformation (converting $i_{abc}(t)$ into i_{dq}):

$$\begin{aligned} i_q &= \frac{2}{3}[i_a(t)cos(\theta_e) + i_b(t)cos(\theta_e - \frac{2\pi}{3}) + i_c(t)cos(\theta_e + \frac{2\pi}{3})] \\ i_d &= \frac{2}{3}[i_a(t)sin(\theta_e) + i_b(t)sin(\theta_e - \frac{2\pi}{3}) + i_c(t)sin(\theta_e + \frac{2\pi}{3})] \end{aligned} \tag{B.3}$$

Reverse Park transformation (converting i_{dq} into $i_{abc}(t)$):

$$\begin{aligned} i_a(t) &= i_q cos(\theta_e) + i_d sin(\theta_e) \\ i_b(t) &= i_q cos(\theta_e - \frac{2\pi}{3}) + i_d sin(\theta_e - \frac{2\pi}{3}) \\ i_c(t) &= i_q cos(\theta_e + \frac{2\pi}{3}) + i_d sin(\theta_e + \frac{2\pi}{3}) \end{aligned} \tag{B.4}$$

The transformation between $i_{\alpha\beta}$ and i_{dq}:

$$\begin{aligned} i_q &= i_\alpha cos(\theta_e) - i_\beta sin(\theta_e) \\ i_d &= i_\alpha sin(\theta_e) + i_\beta cos(\theta_e) \end{aligned} \tag{B.5}$$

$$\begin{aligned} i_\alpha &= i_q cos(\theta_e) + i_d sin(\theta_e) \\ i_\beta &= -i_q sin(\theta_e) + i_d cos(\theta_e) \end{aligned} \tag{B.6}$$

Appendix C

DSP Code for the Proposed Repetitive Controller

Function 1: The transient detector

```
// enable the repetitive control at steady state.Input: torque
   reference from speed loop, sampling period. Output:1 for
   enable
void inline RC_Enable(float torque_ref_PI, float ts, float *
   rc_ena, float *loadchange)
{
        #define L 30
        static int torque_ref_PI_sum=0;
        static int x=0;
        static float torque_ref_PI_prev[L]={0};
        float torque_ref_PI_diff_L=0;
        float torque_ref_PI_diff=0;
        static int trigger=0;
        static int trigger_prev=0;
```

```
// consider it is steady state if the torque reference
   generated by speed loop PI is stable within +-0.3Nm
if (x==L-1)
{
        torque_ref_PI_diff= torque_ref_PI -
           torque_ref_PI_prev[0];
}
else
{
        torque_ref_PI_diff= torque_ref_PI -
           torque_ref_PI_prev[x+1];
}
// load change detection
torque_ref_PI_diff_L= torque_ref_PI - torque_ref_PI_prev
   [x];

// both sudden change and countinous change in iqref is
   within +-0.3Nm for 0.1s ==steady state
if ((fabs(torque_ref_PI_diff) <= 0.3)&&(fabs(
   torque_ref_PI_diff_L)<=0.3))
{
        torque_ref_PI_sum = 1 + torque_ref_PI_sum;
}
else
{
        torque_ref_PI_sum = 0;
}
// torque_ref_PI_sum can not be infinte max 10001
if ( 10001== torque_ref_PI_sum)
    torque_ref_PI_sum=10000;
```

```
        // if the torque reference generated by speed loop is
           stable for 0.1s
        //if ((torque_ref_PI_sum >= 0.1/ts)&&(
           torque_ref_PI_diff_sump<=0.3)&&(
           torque_ref_PI_diff_summ>=-0.3))
            trigger=(torque_ref_PI_sum >= 0.1/ts);
        if (1==trigger)
        {
                *rc_ena= 1.0;
        }
        else
        {
                *rc_ena=0;
        }

        if ((1==trigger) && (0==trigger_prev))
        {
                *loadchange=torque_ref_PI;
        }

        trigger_prev=trigger;
        torque_ref_PI_prev[x]=torque_ref_PI;
        x--;
        if(x<0)
        {
                x=L-1;
        }
}
```

Function 2: Calculate speed from position with the 9^{th} order FIR filter

```
// calculation of speed from position with filter. Input:
```

```
   mechanical position, sampling period. Output: mechanical
   speed
void inline RC_SpeedCalculation(float thetam_rc, float ts, float
    *wm_rc_deriva, float *wm_rc_filt)
{
# define FIR_LENGTH1  19

static float thetam_rc_prev=0;
float thetam_rc_diff=thetam_rc-thetam_rc_prev;

static float mem_wm[FIR_LENGTH1]={0};
float multi=0;
float add=0;

static const float rc_filter[19] = {-0.000956, -0.003610,
   -0.008471, -0.011956, -0.004570, 0.023700, 0.074778,
   0.137195, 0.189172, 0.209435, 0.189172, 0.137195, 0.074778,
   0.023700, -0.004570, -0.011956, -0.008471, -0.003610,
   -0.000956};

static int l=0;
static int x=0;
static int y=0;

if (thetam_rc_diff > M_PI) // if position varies more than pi
{
thetam_rc_diff -= 2*M_PI;
}

else if (thetam_rc_diff < -M_PI) //if position varies more than
```

```
   -pi
{
thetam_rc_diff += 2*M_PI;
}

mem_wm[l]=thetam_rc_diff/ts;
*wm_rc_deriva=mem_wm[l];

thetam_rc_prev=thetam_rc;
// wm_rc_prev=*wm_rc_tustin;
//wm_rc_filted = *wm_rc_deriva * ts / (0.001 + ts) +
   wm_rc_filted * 0.001 / (0.001 + ts);//first order filter with
    time constant=0.0001s

x=l;
y=FIR_LENGTH1-1;
do{ multi=mem_wm[x]*rc_filter[y];
add=add+multi;
y--;
x--;} while(x>=0);
x=FIR_LENGTH1-1;
do{multi=mem_wm[x]*rc_filter[y];
add=add+multi;
y--;
x--;}while(y>=0);

*wm_rc_filt= add;

l++;
if (FIR_LENGTH1==l)
```

```
l=0;
}
```

Function 3: The torque estimation with the 10^{th} order FIR filter

```
// calculation of Te-Tl from speed. Input: speed, Output:Te-Tl
void inline RC_TorqueCalculation(float wm_rc, float ts, float
    J_rc, float B_rc, float *te_rc, float *te_rc_filter)
{

# define FIR_LENGTH2  21

static float wm_rc_prev=0;

static float mem_te[FIR_LENGTH2]={0};
float multi=0;
float add=0;

static const float rc_filter[21] = {0.000783, -0.001222,
    -0.005613, -0.012149, -0.015315, -0.005353, 0.026030,
    0.078547, 0.139960, 0.189818, 0.209027, 0.189818, 0.139960,
    0.078547, 0.026030, -0.005353, -0.015315, -0.012149,
    -0.005613, -0.001222, 0.000783};

static int l=0;
static int x=0;
static int y=0;

mem_te[l]=J_rc*(wm_rc-wm_rc_prev)/ts+B_rc*wm_rc_prev;// net
    torque, without load torque
*te_rc=mem_te[l];
```

```
wm_rc_prev=wm_rc;


x=l;
y=FIR_LENGTH2-1;
do{ multi=mem_te[x]*rc_filter[y];
add=add+multi;
y--;
x--;} while(x>=0);
x=FIR_LENGTH2-1;
do{multi=mem_te[x]*rc_filter[y];
add=add+multi;
y--;
x--;}while(y>=0);

*te_rc_filter= add;

l++;
if (FIR_LENGTH2==l)
l=0;
}
```

Function 4: The proposed repetitive controller including time/angle conversion.

```
void inline RC_MainBody(float rc_ena, float te_dc, float ts,
   float rc_te_est, float speedpi_te_new, float pos, float
   speedmrad, float *rc_iq )
{
# define N 200    // number of locations in memory
# define M 2     // predict the future position after M*TS, for
   DB 2
# define Q_RC 0.999     //forgeting factor
# define GAIN_RC 0.3   // gain
```

```
# define Y 20 //filter delays(19) add torque estimation delay(1)
// during (k+1)^{th} sampling period
///  Input:    pos: mechanical angular position rad, within [0,2
   pi) (t_{k+1})
///  Input:    speedmrad: mechanical speed rad/s (t_{k+1})
///output rc_iq: current reference from repetitive control  (t_{
   k+2})

//variables for time to angle conversion
static float error=0;  //torque error Nm (t_k-20)
static int x=Y;  // for choosing speedpi_te(t_k-20)
static float speedpi_te[Y+1]={0}; //for memorising previous
   torque ref from PI, speedpi_te[x] is the value 21Ts ago,
   speedpi_te[x-1] is 20Ts ago

static float pos_prev[Y+1]={0};//for memorising previous
   positions
static int k1=0; //memory location index
static int k_prev=0;//previous location index in memory

static float error_prev=0; //error (t_k-21)
static float error_diff=0;
static float pos_diff=0;
static float pos_diffK1=0;
float error_pred=0;
//variables for main body of RC
static float mem[N]={0}; ///memery array
//variables for angle to time conversion
float pos_fut=0;
static int k2=0;//location index of predicted future position
```

```
float res2=0;
float output=0;
//variables for clear dc part in mem
static float rc_ena_prev=0;
static int i=0;
static float mem_sum=0;
static float mem_dc=0;

//time to angle conversion
//stop memorizing input when enable=0
if ((0==x)&&(speedpi_te[Y]!=NULL))
{
error=speedpi_te[Y]-rc_te_est;
}
else if ((0!=x)&&(speedpi_te[x-1]!=NULL))
{
error=speedpi_te[x-1]-rc_te_est;
}
else
{
error=0;
}

error_diff=error-error_prev;

if(0==x)
{
k1=(int)(pos_prev[Y]*N/M_2PI); //calculated the corresponding
   location in memory according to current position  [0,N-1];///
   current location index in mermoy
```

```
pos_diff=pos_prev[Y]-pos_prev[x];
}
else
{
k1=(int)(pos_prev[x-1]*N/M_2PI);
pos_diff=pos_prev[x-1]-pos_prev[x];
}
pos_diffK1=k1*M_2PI/N-pos_prev[x];

//////reset mem when the reset trip button is pressed (active
    low)
if (0==(*FPGA_DPR0 & FPGA_DPR0_RTR))
{
memset(mem,0,sizeof(mem));
}

//////////MEMORY UPDATE WITH ERROR PREDICTION///////////////

if (pos_diff>M_PI)
{
pos_diff=pos_diff-M_2PI;
pos_diffK1=pos_diffK1-M_2PI;
}
else if (pos_diff<-M_PI)
{
pos_diff=pos_diff+M_2PI;
pos_diffK1=pos_diffK1+M_2PI;
}

if (k1!=k_prev) // not in the same location,speed change in one
```

```
   Ts is big enough
{
if (pos_diff>0) // spinning in positive direction
{
error_pred=error_prev+error_diff/pos_diff*pos_diffK1;
mem[k1]=mem[k1]*Q_RC+error_pred*GAIN_RC;
}
else if (k1!=(N-1)) // spining in negative direction and (k1+1)
   !=n
{
error_pred=error_prev+error_diff/pos_diff*(pos_diffK1+M_2PI/N);
mem[k1+1]=mem[k1+1]*Q_RC+error_pred*GAIN_RC;
}
else
{
error_pred=error_prev+error_diff/pos_diff*(pos_diffK1+M_2PI/N);
mem[0]=mem[0]*Q_RC+(error_pred)*GAIN_RC;
}
}

/////////OUTPUT WITH POSITION PREDICTION and INTERPOLATION
   ////////////

pos_fut=pos+speedmrad*ts*M+speedmrad*TACQ_ENC;// predict the
   position after M*TS, assuming speed is constant during the
   next M*TS
res2=pos_fut-M_2PI*(int)(pos_fut/M_2PI);//convert pos_fut to
   [0,2pi)
k2=(int)(res2*N/M_2PI);  //the corresponding location of pos_fut
```

```
if((N-1)!=k2) //(k2+1) is from [1,N-1]
{
output =mem[k2]+(mem[k2+1]-mem[k2])/(M_2PI/N)*(res2-k2*M_2PI/N);
    //interpolation, assuming data in memory varies linearly
   within one TS
}
else
{
output =mem[k2]+(mem[0]-mem[k2])/(M_2PI/N)*(res2-k2*M_2PI/N);
}

if(output>6)
output=6;
else if (output<-6)
output=-6;

*rc_iq=output;

////////////update prev values////////////////////
pos_prev[x]=pos;
if (0==rc_ena)
{
speedpi_te[x]=NULL;
}
else
speedpi_te[x]=speedpi_te_new-te_dc;
x--;
if(x<0)
{
x=Y;
```

```
}
error_prev=error;
k_prev=k1;

////////////clear dc value in mem//////
if ((1==rc_ena_prev) && (0==rc_ena))
{
do { mem_sum+=mem[i];
i++;} while(i<N);

mem_dc=mem_sum/N;
do{mem[i]-=mem_dc;
i--;}while(i>0);
}

rc_ena_prev=rc_ena;
}
```

Appendix D

Published Papers

Journals:

1. M. Tang, A. Gaeta, K. Ohyama, P. Zanchehtta and G. Asher, "Enhanced DBCC for high-speed permanent magnet synchronous motor drives," IET Power Electronics, Volume 9, Issue 15, 14 December 2016, p. 2880 – 2890. doi: 10.1049/iet-pel.2015.0232

2. M. Tang, A. Gaeta, A. Formentini and P. Zanchetta, "A Fractional Delay Variable Frequency Repetitive Control for Torque Ripple Reduction in PMSMs", Transactions on Industry Applications, pending of acceptance

---

Conferences:

1. M. Tang, A. Gaeta, K. Ohyama, P. Zanchehtta and G. Asher, "Assessments of dead beat current control for high speed permanent magnet synchronous motor drives," 2015 9th International Conference on Power Electronics and ECCE Asia (ICPE-ECCE Asia), Seoul, 2015, pp. 1867-1874. doi: 10.1109/ICPE.2015.7168033

2. M. Tang, A. Gaeta, A. Formentini and P. Zanchetta, "A variable frequency angle-based repetitive control for torque ripple reduction in PMSMs," 8th IET International Conference on Power Electronics, Machines and Drives (PEMD 2016), Glasgow, 2016, pp. 1-6. doi: 10.1049/cp.2016.0325

3. M. Tang, A. Formentini, S. Odhano and P. Zanchetta, "Design of a repetitive controller as a feed-forward disturbance observer," 42nd Annual Conference of IEEE Industrial Electronics Society (IECON 2016), Florence, 2016