

2008

Adaptive control of sinusoidal brushless DC motor actuators

Liangtao Zhu
Michigan Technological University

Follow this and additional works at: <https://digitalcommons.mtu.edu/etds>

 Part of the [Mechanical Engineering Commons](#)


Copyright 2008 Liangtao Zhu

Recommended Citation

Zhu, Liangtao, "Adaptive control of sinusoidal brushless DC motor actuators", Dissertation, Michigan Technological University, 2008.

<https://digitalcommons.mtu.edu/etds/427>

Follow this and additional works at: <https://digitalcommons.mtu.edu/etds>

 Part of the [Mechanical Engineering Commons](#)

Adaptive control of
Sinusoidal Brushless DC Motor actuators

BY
LIANGTAO ZHU

A DISSERTATION

Submitted in partial fulfillment of the requirements

for the degree of

DOCTOR OF PHILOSOPHY

(Mechanical Engineering-Engineering Mechanics)

MICHIGAN TECHNOLOGICAL UNIVERSITY

2008

Copyright © Liangtao Zhu 2008

This dissertation, "Adaptive Control of Sinusoidal Brushless DC Motor Actuators," is hereby approved in partial fulfillment of the requirements for the degree of DOCTOR OF PHILOSOPHY in the field of Mechanical Engineering-Engineering Mechanics.

DEPARTMENT:

Mechanical Engineering-Engineering Mechanics

Signatures:

Dissertation Advisor

Typewritten Name: Gordon G. Parker

Date

Department Chair

Typewritten Name: William W. Predebon

Date

ACKNOWLEDGEMENT

I would like to express my gratitude to all those who gave me the possibility to complete this thesis. I am deeply indebted to my advisors Prof. Gordon Parker and Dr. Ravindra Patankar, whose help, stimulating suggestions and encouragement helped me in all the time of research for and writing of this thesis.

The members of my dissertation committee, Dr. Roshan D'Souza, Prof. Charles Van Karsen, and Prof. Clark Givens, have generously given their time and expertise to better my work. I thank them for their contribution and support.

Especially, I would like to give my special thanks to my wife Nadia whose patient love enabled me to complete this work. I would also like to thank my mom Yufen for her unwavering faith in me.

THIS DISSERTATION IS DEDICATED TO

MY WIFE NADIA AND MY DAUGHTER KIMBERLIE

TABLE OF CONTENTS

Table of Contents	v
List of Figures.....	ix
List of Tables.....	xiii
Abstract.....	xiv
Chapter 1 Introduction.....	1
1.1 Electrical Power-Assisted Steering System	1
1.2 Electric Motor Actuator For EPAS.....	3
1.3 BLDC Motors and Control Overview.....	4
1.4 Literature Review on BLDC Motor Control.....	11
1.5 Adaptive Control for Sinusoidal BLDC Motors.....	16
1.6 Publications.....	22
1.7 Summary	23
Chapter 2 Sinusoidal Brushless DC Motor Modeling.....	25
2.1 Permanent Magnet Synchronous Motor Modeling.....	25
2.1.1 The three-phase model.....	26
2.1.2 The direct quadrature coordinate model	28
2.1.3 Implementation of dq coordinate controllers	36
2.2 Torque Control of Sinusoidal BLDC Motors	38
2.2.1 PI controller	39
2.2.2 Feedforward inverse dynamics controller.....	41
2.2.3 Controller selection.....	43

2.3	Summary	45
Chapter 3	Adaptive Parameter Estimation	47
3.1	Motor Inverse Controller Model	47
3.2	Single Parameter Estimation	49
3.2.1	Single parameter estimation	49
3.2.2	Stability of the single parameter estimation scheme	51
3.2.3	Simulation results	55
3.3	Multiple Parameter Estimation	57
3.3.1	Multiple parameter estimation	57
3.3.2	Proof of stability	59
3.4	Improving The Dynamic Performance Of The Adaptive Algorithm	62
3.4.1	Motor electrical dynamics	63
3.4.2	Motor speed sampling delay	68
3.5	Multiple Parameter Estimation Using Gram-Schmidt Orthonormalization	73
3.5.1	Multi-parameter estimation using Gram-Schmidt orthonormalization	74
3.5.2	Proof of stability	76
3.6	Multi-Parameter Estimation Using q and d Axis Current	80
3.6.1	Parameter estimation	80
3.6.2	Proof of unbiased estimation	83
3.7	Conclusion	91
Chapter 4	Recursive Parameter Identification	92
4.1	Discrete Model of the BLDC Motor Dynamics	93

4.2	Recursive Least Square Estimation	96
4.3	RLS Simulation Results	98
4.4	Extended Kalman Filter	101
4.5	EKF Simulation Results.....	106
4.6	Summary	109
Chapter 5	Model Reference Adaptive Control Design.....	111
5.1	Model Reference Adaptive Control	112
5.1.1	The BLDC motor reference model	113
5.1.2	The control law	114
5.1.3	The adaptation law	115
5.2	Simulation Results	119
5.3	Conclusion	121
Chapter 6	Closed loop Simulation and Controller Performance Evaluation.....	123
6.1	EPAS closed loop simulation.....	123
6.1.1	EPAS close loop system model	124
6.1.2	EPAS simulation results and analysis.....	126
6.2	Closed Loop Simulation Of a Motor Speed Control Application.....	135
6.3	Conclusion	139
Chapter 7	An Operation Simulation Model for BLDC Motors.....	146
7.1	A Quasi-Physical BLDC Motor Model	147
7.1.1	The Line-to-Line PMSM model	148
7.1.2	The H-bridge PWM inverter model.....	150
7.1.3	The rotor angular displacement transducer.....	152

7.1.4	The Hall effect sensor	153
7.1.5	SVPWM.....	154
7.1.6	Simulation configuration	157
7.2	Study of Controller Implementation Issues in Simulation.....	159
7.2.1	Initialization	160
7.2.2	Incremental encoder resolution.....	164
7.3	Limit of the Quasi-Physical BLDC Motor Model	165
7.4	Summary	167
Chapter 8	Conclusion and Future Works	168
8.1	Conclusion	168
8.2	Future Works	173
Appendix I	The probability density function derivation.....	175
Appendix II	Solution of the Integration Terms in Equation (4.6).....	177
Appendix III	Continuous Extended Kalman Filter for BLDC Motors.....	179
References	182

LIST OF FIGURES

Figure 1.1 Schematic diagram of a three-phase BLDC motor with one pair of rotor permanent magnet poles.	5
Figure 1.2 Typical trapezoidal and sinusoidal back EMF waveform	6
Figure 1.3 Schematic diagram of a typical BLDC motor control system.....	7
Figure 1.4 BLDC motor torque waveform when operated with back EMF zero crossing sensing mechanism.	9
Figure 1.5 Typical adaptive controller block diagram.....	17
Figure 2.1 The three stator phase, the <i>qd</i> coordinates and the <i>ab</i> coordinates	29
Figure 2.2 Implementation of a generic controller designed in <i>dq</i> coordinates.....	37
Figure 2.3 Step response of the BLDC motor with the PI controller and the feedforward controller at $\omega_{e0} = 0$, assuming full knowledge of the plant parameters.	43
Figure 2.4 Step response of the BLDC motor with the PI controller and the feedforward controller at $\omega_{e0} = 0$, with 10% error in <i>Ke</i> and 50% error in <i>R</i>	44
Figure 3.1 The schematic diagram of the motor test system in simulation.	56
Figure 3.2 Single parameter estimation performance in simulation.....	57
Figure 3.3 Two-parameter estimation performance in simulation.....	62
Figure 3.4 Single parameter estimation transient performances: non-dynamic motor inverse controller vs. approximated dynamic inverse controller.	66

Figure 3.5 Multi-parameter estimation using non-dynamic motor inverse controller vs. dynamic motor inverse controller	67
Figure 3.6 Single parameter estimation performances: comparison of the basic estimation program and the performance improvements.....	71
Figure 3.7 Performance comparison of estimation scheme in open loop simulation: case 1 – the basic scheme; case 3– with approximated $\Delta\omega_m$ compensation.	72
Figure 3.8 Performance comparison of 4 different estimation schemes in open loop simulation: two-parameter estimation.....	73
Figure 3.9 Schematic diagram of Gram Schmidt Orthonormalization	75
Figure 3.10 Parameter estimation performance with Gram-Schmidt orthonormalization: the initial parameter error + 50% R and –15% Ke	79
Figure 3.11 The distribution of $y = 1/x$ given $x \sim N(\mu(x), 1)$ for several values of $\mu(x)$	86
Figure 3.12 The mean of $y = (x_r - x)/x$ given $x \sim N(\mu(x), 1)$ compares to the algebraic function $y = (x_r - \mu(x))/\mu(x)$	87
Figure 3.13 The mean of $y = 1/x$ compares to $1/\mu(x)$	88
Figure 3.14 The qd -solver simulation results: + 50% R and –15% Ke	90
Figure 4.1 RLS estimation simulation results of $a_1 \sim a_5$ & $b_1 \sim b_5$: constant $\omega_m = 100rad / s$	99
Figure 4.2 RLS estimation simulation results of $a_1 \sim a_5$ & $b_1 \sim b_5$: variant velocity.....	101
Figure 4.3 EKF estimation simulation: $R_h(0) = 1.25R$, $Ke_h(0) = 0.9Ke$	107
Figure 4.4 EKF estimation simulation: $R_h(0) = 1.4R$, $Ke_h(0) = 1.2Ke$	107

Figure 4.5 EKF estimation simulation: $R_h(0)=0.8R$, $Ke_h(0)=1.2Ke$	108
Figure 4.6 EKF estimation simulation: $R_h(0)=0.8R$, $Ke_h(0)= 0.8Ke$	108
Figure 4.7 EKF estimation simulation: $R_h(0)=R$, $Ke_h(0)= Ke$	109
Figure 5.1 A schematic diagram of a typical MRAC controller.....	112
Fig 5.2 MRAC reference current tracking performance in simulation: $R_c(0) = 0.5R$, $Ke_c(0) = 1.2Ke$	120
Fig 5.3 MRAC parameter estimation performance in simulation, case 1: $R_c(0) = 0.5R$, $Ke_c(0) = 1.2Ke$	120
Fig 5.4 MRAC parameter estimation performance in simulation, case 2: $R_c(0) = 0.5R$, $Ke_c(0) = 0.8Ke$	121
Figure 6.1 Schematic diagram of a typical EPAS.....	124
Figure 6.2 Comparison of the adaptive BLDC motor controllers in EPAS closed loop simulation: $\Delta R(0) = +50\%R$ and $\Delta K_e(0) = +10\%K_e$	129
Figure 6.3 Comparison of the adaptive BLDC motor controllers in EPAS closed loop simulation: $\Delta R(0) = +50\%R$ and $\Delta K_e(0) = -15\%K_e$	130
Figure 6.4 Comparison of the adaptive BLDC motor controllers in EPAS closed loop simulation: $\Delta R(0) = -50\%R$ and $\Delta K_e(0) = -15\%K_e$	131
Figure 6.5 Comparison of the adaptive BLDC motor controllers in EPAS closed loop simulation: $\Delta R(0) = +50\%R$ and $\Delta K_e(0) = +10\%K_e$	132
Figure 6.6 The block diagram of the motor shaft dynamics Simulink model.	136
Figure 6.7 The block diagram of the anti-windup PI controller Simulink model.....	137

Figure 6.8 Comparison of the adaptive BLDC motor controllers in the motor speed control closed loop simulation: $\Delta R(0) = +50\%R$ and $\Delta K_e(0) = +10\%K_e$	140
Figure 6.9 Comparison of the adaptive BLDC motor controllers in the motor speed control closed loop simulation: $\Delta R(0) = +50\%R$ and $\Delta K_e(0) = -15\%K_e$	141
Figure 6.10 Comparison of the adaptive BLDC motor controllers in the motor speed control closed loop simulation: $\Delta R(0) = -50\%R$ and $\Delta K_e(0) = -15\%K_e$	142
Figure 6.11 Comparison of the adaptive BLDC motor controllers in the motor speed control closed loop simulation: $\Delta R(0) = -50\%R$ and $\Delta K_e(0) = +10\%K_e$	143
Figure 7.1 A typical three-phase BLDC motor system block diagram.....	148
Figure 7.2 The line-to-line PMSM model in Simulink.....	150
Figure 7.3 A simplified circuit of the BLDC motor system.	151
Figure 7.4 The H-bridge inverter model.	152
Figure 7.5 The BLDC motor components block diagram.	153
Figure 7.6 The voltage vectors in the space vector modulation.	155
Figure 7.7 Transistors' on-off timing in the six sectors.....	158
Figure 7.8 SVPWM function implementation in Simulink	159
Figure 7.9 The Simulink model of a closed loop BLDC motor system	160
Figure 7.10 The practical controller model in Simulink.....	161
Figure 7.11 The starting scheme Simulink model	162
Figure 7.12 The BLDC motor physical model closed loop simulation	163
Figure 7.13 The BLDC motor simulation with a low-resolution encoder ($cpr=36$)...165	
Figure 7.14 The BLDC motor simulation with a high-resolution encoder ($cpr=4096$)	166

LIST OF TABLES

Table 2.1 PI controller gains.....	41
Table 3.1 Bound of errors in open loop simulation: two-parameter estimation.....	72
Table 4.1 EKF parameter estimation mean and variance for different noise level.....	107
Table 6.1 Steady-state performances in EPAS closed loop simulation.....	133
Table 6.2 Steady-state performances in speed control closed loop simulation.....	137
Table 7.1 The eight basic voltage vectors in the SVPWM.....	155
Table 7.2 Summary of the transistor on-off timing calculation in each sector.....	157

ABSTRACT

Electrical Power Assisted Steering system (EPAS) will likely be used on future automotive power steering systems. The sinusoidal brushless DC (BLDC) motor has been identified as one of the most suitable actuators for the EPAS application. Motor characteristic variations, which can be indicated by variations of the motor parameters such as the coil resistance and the torque constant, directly impart inaccuracies in the control scheme based on the nominal values of parameters and thus the whole system performance suffers. The motor controller must address the time-varying motor characteristics problem and maintain the performance in its long service life.

In this dissertation, four adaptive control algorithms for brushless DC (BLDC) motors are explored. The first algorithm engages a simplified inverse dq -coordinate dynamics controller and solves for the parameter errors with the q -axis current (i_q) feedback from several past sampling steps. The controller parameter values are updated by slow integration of the parameter errors. Improvement such as dynamic approximation, speed approximation and Gram-Schmidt orthonormalization are discussed for better estimation performance. The second algorithm is proposed to use both the d -axis current (i_d) and the q -axis current (i_q) feedback for parameter estimation since i_d always accompanies i_q . Stochastic conditions for unbiased estimation are shown through Monte Carlo simulations. Study of the first two adaptive algorithms indicates that the parameter estimation performance can be achieved by using more history data. The Extended Kalman Filter (EKF), a representative recursive estimation algorithm, is then investigated for the BLDC motor application.

Simulation results validated the superior estimation performance with the EKF. However, the computation complexity and stability may be barriers for practical implementation of the EKF. The fourth algorithm is a model reference adaptive control (MRAC) that utilizes the desired motor characteristics as a reference model. Its stability is guaranteed by Lyapunov's direct method. Simulation shows superior performance in terms of the convergence speed and current tracking. These algorithms are compared in closed loop simulation with an EPAS model and a motor speed control application. The MRAC is identified as the most promising candidate controller because of its combination of superior performance and low computational complexity.

A BLDC motor controller developed with the *dq*-coordinate model cannot be implemented without several supplemental functions such as the coordinate transformation and a DC-to-AC current encoding scheme. A quasi-physical BLDC motor model is developed to study the practical implementation issues of the *dq*-coordinate control strategy, such as the initialization and rotor angle transducer resolution. This model can also be beneficial during first stage development in automotive BLDC motor applications.

Chapter 1 INTRODUCTION

Electrical power assisted steering system (EPAS) will likely be used for future power steering systems and thus they are on intense target of technology innovation. Brushless DC (BLDC) motors are often selected as the actuator for EPAS in many publications. The characteristics of individual BLDC motors vary with environmental factors, service life, and manufacturing process. A BLDC motor controller that maintains the actuator performance regardless of these variations is highly desired in EPAS applications and is the focus of this research.

1.1 Electrical Power-Assisted Steering System

Automotive power-assisted steering systems have traditionally been dominated by hydraulic pumps. The “always-on” hydraulic pump consumes power from the engine whenever the engine is running. This results in unwanted fuel consumption if the driver is not moving the steering wheel. In contrast, electric motors only consume power when the driver demands assisting force. This has been recognized as an effective way to improve fuel economy.

Kluger and Harris (Kluger and Harris 2007) compared engine brake specific fuel consumption (BSFC) effects of off-engine accessories on a heavy-duty truck and a light duty mini van. Through various driving cycle simulations, it was found that configuring vehicles with off-engine accessories is a method that provides fuel savings ranging from 3% to 15%. For heavy-duty trucks the improvements ranged from 3-11%, and 8-15% for

minivans. Power steering is one of the major contributors for its low operational cycle (20%).

An EPAS offers about 80% lower energy consumption than the hydraulic pump, and the omission of the hydraulic fluid decreases the environmental impact (Harter 2000). Similarly, Burton (Burton 2002) showed an average 3.0% reduction of fuel consumption on a 1.6L car in combined driving cycle, and up to 3.5% reduction in the city driving cycle. In today's automotive market, the advantage in fuel economy alone can make the EPAS an attractive option over the traditional hydraulic power steering system.

In addition, the EPAS is more compact and therefore easier for assembly and configuration. It can be adapted to a wide variety of vehicle applications with small changes to hardware. Furthermore, the EPAS is more flexible in terms of function. Its software control strategies can be modified to provide a wide range of handling characteristics without changing hardware. For example, it is desirable that steering systems give high assistance to the driver at low vehicle speeds while low assistance at high speeds. Cetin et al. (Cetin 2005) showed that this could not be achieved by conventional HPAS systems in which a torsion bar with constant mechanical properties (inertia, stiffness and damping) was used to determine the assistance amount. A compliant controller for the EPAS was proposed in this work and it was shown that this could be achieved by changing the virtual system dynamic parameters and the control gains. The flexibility of the EPAS also provides more opportunity for better vehicle handling performance. Tanaka et al. (Tanaka 2007) developed an active steering control algorithm for an EPAS. This algorithm increased the steering return torque in a region where the alignment torque was saturated due to the driver's excessive steering maneuver

on a slippery road. Kubota et al. (Kubota 2007) proposed an algorithm by using the EPAS to counteract steering-pull caused by lateral disturbances such as road contour, suspension alignment error and tire properties. The disturbance was estimated by finding the mean assisting torque in straight line driving and canceled by the EPAS.

Due to these advantages and progress in electric motor technology, battery and microcontrollers, EPAS will become the main stream of the future power steering system and an intense target of technology innovation.

1.2 Electric Motor Actuator For EPAS

The electric motor is the heart of the EPAS. Electric motors are the most commonly used actuators in mechatronic systems. To name a few advantages, they have higher efficiency and higher power density (power/mass) compared to their mechanical or hydraulic counterparts. Traditionally, brushed DC motors and asynchronous or synchronous AC motors are the most widely used due to their low costs. However, the reliability and maintenance cost introduced by the brush has always been a limiting factor for brushed DC motors' application to the EPAS. Traditional AC motors are rarely used in automotive applications because of unavailability of the AC power supply and their undesirable low speed characteristics.

In the past two decades, there has been significant progress on rare-earth permanent magnet synchronous (PMS) motors. These motors are equipped with permanent magnet rotors. Switching devices, such as brushes, are no longer needed for their operation. Together with an inverter, a switching logic controller and some rotor position feedback mechanism, a PMS motor can be operated in self-controlled mode on a DC current source. Such a motor is often called a brushless DC (BLDC) motor. Due to the absence of

brushes and a commutator, BLDC motors have a number of advantages in maintenance, reliability and efficiency compared to conventional DC or AC motors.

Iles-Klumpner (Iles-Klumpner 2005) reviewed electric actuator candidates for the electrical power steering application, including brushed and brushless drive systems based on permanent magnet brushed DC (DC), induction (IM), permanent magnet trapezoidal and sinusoidal synchronous, switched-reluctance (SR), and reluctance synchronous (RS) machines. A wide range of factors were considered in evaluation, including but not limited to torque density, peak to continuous torque capability, variable speed control, torque pulsations, temperature sensitivity, acoustic noise, power converter requirements, manufacturing, reliability, customer acceptance, cost. The sinusoidal and trapezoidal BLDC motors were identified as the most suitable actuators for the EPAS application. The trapezoidal BLDC motors have been implemented in production EPAS by Delphi Corporation since 2004.

Though the BLDC motors have been accepted as the most suitable candidate actuators for the EPAS application, it is a fact that the BLDC motors require more sophisticated control than brushed DC motors, especially sinusoidal BLDC motors. This is the background and motivation of this research.

1.3 BLDC Motors and Control Overview

A BLDC motor is an AC synchronous motor with permanent magnets on the rotor and windings on the stator. Most BLDC motors have three phase stator windings, while their rotors can have several pairs of rotor magnet poles. Figure 1.1 is a schematic diagram of a three-phase BLDC motor with one pair of rotor magnet poles. The energized stator windings create an electromagnetic field, and the rotor is attracted to align with the stator

field. When current is supplied to the stators in an appropriate sequence, the stator electromagnetic field rotates and drives the rotor magnets. The stator electromagnetic field and the rotor usually rotate at the same speed, and the phase lead between the stator field and the rotor needs to be maintained to generate constant torque. Measurement of the rotor position is needed for a BLDC motor's operation to properly sequence the stator current.

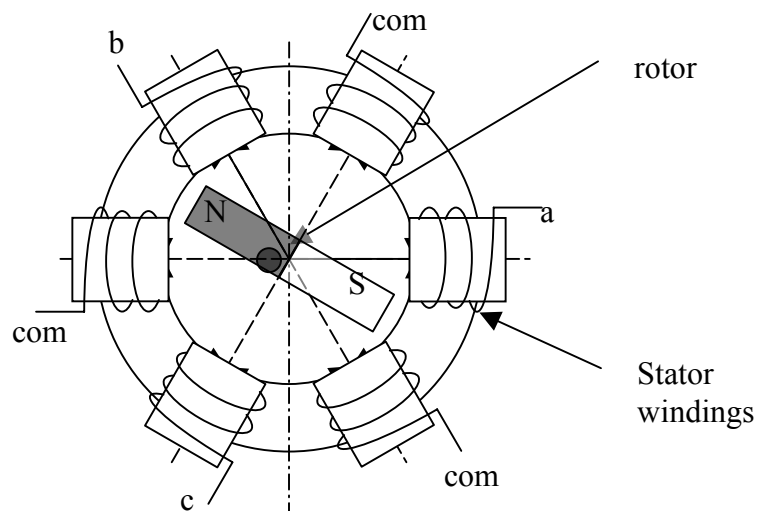


Figure 1.1 Schematic diagram of a three-phase BLDC motor with one pair of rotor permanent magnet poles.

“com” indicate common lines; “a”, “b”, and “c” indicate stator terminals.

BLDC motors can be categorized as trapezoidal or sinusoidal according to the waveform of their back electromotive force (EMF), as shown in Figure 1.2. Structurally, these two types of BLDC motors are different in the way that their stator slots and coils distribute long the stator inner periphery, which creates different waveforms of back EMF in stator coils as the magnets rotating with the rotor. Both types of BLDC motors

require the current and back EMF in each stator phase to be synchronized so as to generate constant torque. The generated torque is proportional to the phase current value on each phase, and the total rotor shaft torque output is the summation of torque generated on all phases.

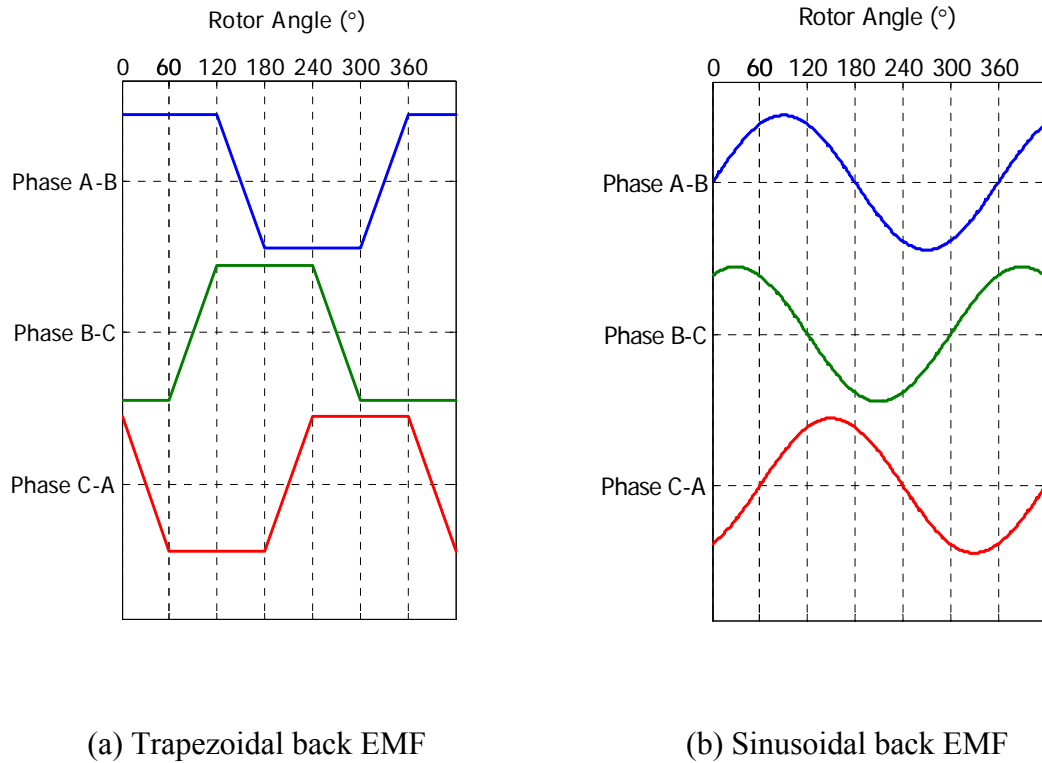


Figure 1.2 Typical trapezoidal and sinusoidal back EMF waveform

Figure 1.3 is a schematic diagram of a typical three-phase BLDC motor control system. It usually consists of an inverter, a micro-controller that controls the switching logic, a rotor position feedback mechanism and a permanent magnet synchronous motor. In the outer loop control system, the BLDC motor control system plays the role of its actuator,

so the objective of BLDC motor control is to generate appropriate desired torque for the outer loop system.

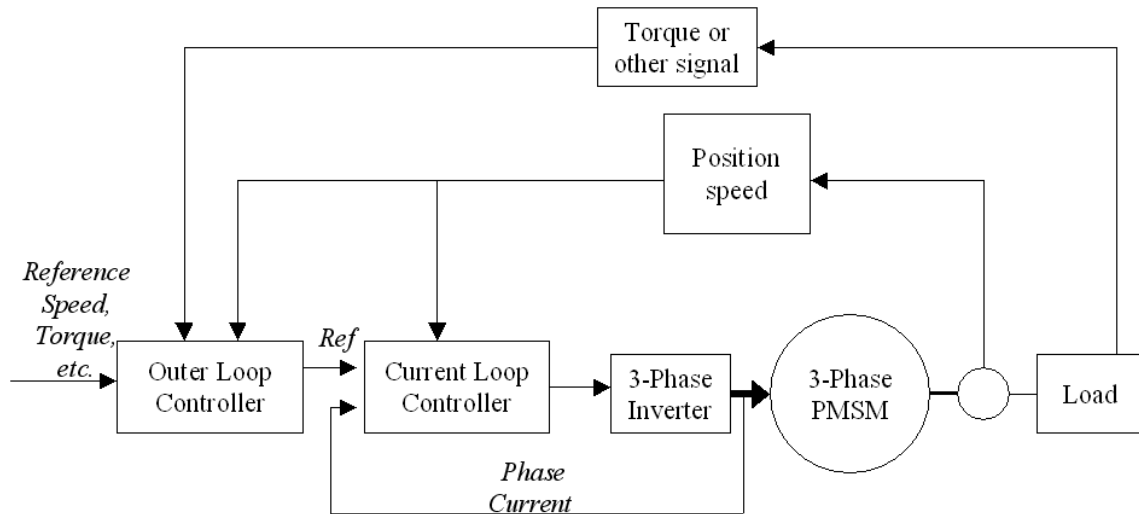


Figure 1.3 Schematic diagram of a typical BLDC motor control system.

PMSM is permanent magnet synchronous motor.

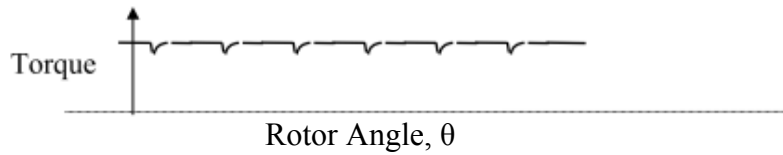
Similar to brushed DC motors, the generated torque from a BLDC motor is generally proportional to the phase currents, so the torque control is basically the phase current control. However, for the BLDC motor, current control consists of two sub-tasks: stator and rotor flux synchronization, and control of the phase current values. The former task ensures consistent torque generation, while the latter determines the magnitude of the generated torque. Both tasks are accomplished through the three-phase inverter of Figure 1.3. To maintain the synchronization, the controller dynamically decides that a certain set of gates is to be turned on and the remaining gates to be off based on the rotor position. The phase current value control is usually achieved by adjusting the timing of those gates

to be turned on. Typically this is done by either hysteresis current control or pulse width modulation (PWM) control. In the hysteresis current control, the phase branch gates are switched on if the feedback current is outside a preset band of their corresponding reference values, and vice versa. Apparently, the gate switching frequency varies as the current error varies. In the PWM control mode, the gates are switched at a fixed frequency, and the current value is controlled through adjusting the PWM duty cycle.

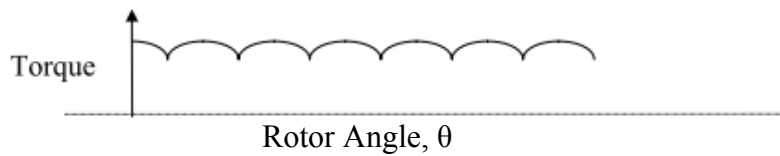
The control principle for the trapezoidal BLDC motors is that current should flow in only two of the three phases at a time (Texas Instruments Europe, 1997). There should be no torque production during the region of back EMF zero crossing for each individual stator phase. Trapezoidal BLDC motors are often equipped with transducers to detect the back EMF zero-crossing regions. The inverter gate switching logic can be obtained through a truth table based on the status of a set of Hall effect sensor outputs. Theoretically, constant torque can be generated with the rotor position feedback, as the back EMF is constant when the phases are switched on. However, due to the phase inductance, the stator phase current cannot be established instantaneously, thus torque ripple is inevitable at every phase commutation as shown in Figure 1.4(a). Sinusoidal BLDC motor can also operated in this way, but the torque ripple will be in sinusoidal shape due to the sinusoidal back EMF and phase commutation, as shown in Figure 1.4(b).

In most trapezoidal BLDC motors, due to the fact that only two phases are on at a time and a single current flow through them, it is possible to control the current with one current sensor on the inverter input line from the DC power supply. Since the switching logic is fixed for each individual motor and is often programmed in hardware like a truth

table, the current controller only needs to control the current value. In this aspect, it is quite similar to the brushed DC motors.



(a) Torque generated by a Trapezoidal BLDC motor



(b) Torque generated by a Sinusoidal BLDC motor

Figure 1.4 BLDC motor torque waveform when operated with back EMF zero crossing sensing mechanism.

Sinusoidal BLDC motors are capable of generating constant torque if the phase currents are controlled to be sinusoidal and in-phase with their corresponding back EMF. This usually requires high resolution of rotor position feedback, especially in the applications where motor speed and load vary significantly. In addition, the inverter switching schemes are more sophisticated as all stator phases are on during the operation of the motor. One of the most widely used methods for constant torque generation in sinusoidal BLDC motors is the Space Vector Pulse Width Modulation (SVPWM) or Field Oriented Control. In the SVPWM scheme, several base voltage vectors are defined in the stator magnetic field coordinates that are determined by the inverter gate on/off states. In each of the PWM periods the controller sends a set of gate on/off commands that correspond to specific base voltage vectors. Through adjusting the duty cycles of

each of these base voltage vectors, any voltage vector in this coordinate can be approximated by a linear combination of the base vectors.

Three-phase sinusoidal BLDC motors, when controlled to achieve constant torque generation, need all three-phase currents to be sinusoidal and in phase with their corresponding back EMF. Unlike the single current control in trapezoidal BLDC motors, all phase currents of the sinusoidal BLDC motor contribute to the total torque output at all times. This usually requires two or three current sensors for feedback control to achieve reference torque tracking. The direct-quadrature (**dq**) model (1.1), which is obtained from the three-phase model through two coordinate transformations, is often implemented for the motor dynamics analysis and control development.

$$\begin{aligned}
 L_q \frac{di_q}{dt} &= -Ri_q - n_p \omega L_d i_d - K_e \omega + v_q, \\
 L_d \frac{di_d}{dt} &= -Ri_d + n_p \omega L_q i_q + v_d, \\
 \tau &= K_e i_q
 \end{aligned} \tag{1.1}$$

where subscripts d and q indicate the direct and quadrature coordinated variables, i is the current, v is the control voltage, R is the coil resistance, L is the coil inductance, ω is the rotor velocity, and n_p is the number of rotor permanent magnet poles pairs.

With the **dq** model, the three phase currents can be transformed into two independent virtual currents (i_d, i_q) in the direct and quadrature coordinates. If the motor is operated under its rated speed, the generated torque is approximately proportional to the quadrature current component (i_q). Therefore, given a reference torque from the outer loop controller, the reference current i_q can be calculated. Since the direct axis current (i_d) does not contribute to the torque generation, it is often desirable to keep it as close to zero

as possible to improve energy efficiency. Control techniques such as proportional integral differential (PID) control can be implemented to achieve reference current tracking.

1.4 Literature Review on BLDC Motor Control

Though its application to the EPAS is relatively new, BLDC motor control has been an active subject in technical publications since the 1980s. Control techniques found in publications addressed speed/current-tracking performance, robustness to parameter variation, torque ripple, saturation and other issues related to BLDC motors. Below is an overview of references most closely related to this work.

Pillay and Krishnan (Pillay 1989) presented a *dq* coordinate model for an industrial sinusoidal PMSM drive and a third order state space model for its speed control application. Simulation results were shown for a pulse width modulation (PWM) current control and hysteresis current control. Reference phase currents were transformed from the desired currents in *dq* coordinates, which in turn were determined by the speed error and torque command.

Pelczewski and Kunz (Pelczewski 1990) designed an optimal controller to address the voltage saturation of BLDC motors. Instead of using a *dq* coordinate model, a 4th order state space model was used for the BLDC motor dynamics including rotor mechanical dynamics and stator coil electrical dynamics. The stator current dynamics were simplified as one of the state equations.

Matsui and Ohashi (Matsui 1992) developed a digital signal processor (DSP) based adaptive controller for a BLDC motor. Space Vector Pulse Width Modulation (SVPWM) was shown to be superior to the Hall-sensor based vector selection PWM in terms of current control performance. The DSP controller was running at 200 μ s. The rotor

position feedback was from a resolver updated at every 800 μ s and interpolated every 100 μ s. In this work, the PWM frequency was set to 10kHz.

Hoang et al. (Hoang 1994) directly used the inverse of the dq coordinate model of the BLDC motor to calculate command voltage. The derivative was approximated by backwards-finite difference model. In addition, an integral of feedback current error was implemented for control correction.

Kim et al. (Kim 1995) proposed an adaptive current controller for the PMSM. It was shown that parameter estimation was robust to other un-estimated parameter error, and this controller could be used to estimate motor speed if sensorless control was desired.

Low et al. (Low 1996) defined a motor identity based on the BLDC current frequency content. It was shown that an optimal drive current could be determined in this way, which gave smooth and maximal torque.

Sozer et al. (Sozer 1997) compared direct model reference adaptive control (DMRAC) with indirect model reference adaptive control (IMRAC). DMRAC gave good results on disturbance rejection in load, set point and parameters. IMRAC had difficulties when parameters changed fast.

Chen et al. (Chen 2006) designed a two-degree-of-freedom controller for the BLDC motor current tracking. This controller showed better disturbance rejection than proportional integral controller. However, since the d axis dynamics and back EMF were defined as disturbances, the compensation through filtering the feedback would be later than direct estimation of the back EMF and d axis current. Therefore, it suffered the same problem as the PI controller, though to a lower extent.

Rahman et al. (Rahman 2003) proposed an adaptive backstepping control for a PMS motor, and showed it was globally asymptotically stable using Lyapunov's direct method.

Cauet et al. (Cauet 2001) derived a robust controller for a class of nonlinear system with parameter variation based on the linear matrix inequality (LMI) approach and polytopic model. Stability was analyzed using a parameter-dependent Lyapunov function and the global stability was proved in the presence of the nonlinearity that was ignored during the linearization process. This controller was implemented on an induction motor and simulation showed asymptotic tracking of the speed trajectory.

Forrai et al. (Forrai 2001) studied robust control for BLDC motors in the presence of control voltage saturation. A BLDC motor model was identified as a 2nd order transfer function (from reference current to speed output) by using the Auto-Regressive eXogenous (ARX) method. Gain scheduling, based on the speed error, was used for the control strategy.

Chen et al. (Chen 2000) proposed a combination of a PI controller, feedforward controller and a robust controller based on direct disturbance cancellation for the BLDC motor phase current control. This controller yielded fast phase current response for both trapezoidal and sinusoidal BLDC motors.

Rubaii and Kotaru (Rubaii 2000) studied a three-layer feedforward artificial neural network (FANN) and a dynamic back propagation (DBP) neural network controller for a BLDC motor application. An adaptive online training strategy was proposed for the FANN. It converged much faster than the DBP learning algorithm with a constant learning rate. The stability of the neural network controllers was not discussed.

Petrovic et al. (Petrovic 2000) studied the 6th and 12th order harmonics of the permanent magnet flux in BLDC motors and their effect on the torque ripple. Since the coefficients of these harmonic components were changing with time, the authors proposed a Lyapunov stable adaptive estimation algorithm. Simulation showed ideal cancellation of the torque ripple. However, they pointed out the practical performance would be limited by the sampling of the controller inputs and outputs. A lower speed limit occurred with increased delay in the speed measurement. Other hardware issues, like dead time and switch voltage drop compensation, had to be solved for successful controller implementation. Their controller was running on a TMS320C31 floating point DSP at a sampling interval of 500 μ s.

Chen and Tang (Chen 1999) proposed a sliding-mode controller for BLDC motors. This controller was implemented as hardware logic circuit, and therefore had essentially no sampling delay problems. Test data showed fast current step response with controller implemented on a FPGA.

Lin and Lin (Lin 1999) proposed a robust controller by combining an integral proportional tracking controller and an adaptive uncertainty observer. A lumped uncertainty was defined to capture the parametric and nonparametric model uncertainty. A Lyapunov function of the tracking error was used to derive the adaptation law.

Sensorless control is one of the latest trends in BLDC motor control publications (Takeshita 1994, Rahman 2003, Haque 2004, Lee 2004, Bolognani 2002, Kim 2003, Johnson 1999 etc.). Here the word “sensorless” means that no rotor position sensors are used. The rotor position and speed are estimated using the back EMF measurement or other indirect methods. However, most of these publications indicated unreliable speed

and angle estimation at low rotor speed, which is an inherent problem of sensorless estimation algorithms. This is undesirable for automotive steering system applications, considering the situation that assisting torque is required while the driver holds the steering wheel at some certain angle.

Many of the BLDC motor control references above were based on the dq coordinate model. The dq coordinate model has been well established for AC motors including induction, synchronous, and PMS motors (Krause 1986, Pillay and Krishnan 1989, Rahman and Zhou 1994, Yang et al. 2003, and etc). These AC motors usually employ a sinusoidal AC power supply at a fixed frequency that is equivalent to the rotor speed. The coordinate transformations from the phase model to the dq coordinate model cancel the rotor angle involved in the dynamic equations (1.1), thus the complexity of analysis is reduced. BLDC motors do not have such dq -model-friendly power supply. Thus, control developers must program the inverter to approximate the AC power supply using the rotor angle measurement. This usually requires knowledge of the inverter and power electronics. In some situations, such as in the application to EPAS, it is often desirable to have a full model of the BLDC motor including the inverter and the PMS motor so that the control program can be tested in simulation. Unfortunately, this type of model only appears in very limited publications. For example, Hossain and Deshpande (Hossain and Deshpande 2003) developed a detailed BLDC motor model in Simulink with thermal degradation, cogging torque, friction loss and digital controller quantization phenomena. Simulation results are shown for 1000rpm and 2Nm load torque. The simulation with non-idealities matched well with test bench measurements. Urasaki et al. (Urasaki 2000)

analyzed power loss factors in a BLDC motor, and identified the corresponding torque loss equations.

In summary, many control techniques have been proven effective in BLDC motor control applications. It is not uncommon that multiple solutions exist for the same problem. However, it is noteworthy that adaptive control has been a popular choice for BLDC motors, especially sinusoidal BLDC motors.

1.5 Adaptive Control for Sinusoidal BLDC Motors

Sinusoidal BLDC motors are one of the most suitable actuators for the EPAS applications because of their high reliability, low maintenance cost and close-to-DC-motor dynamic performance. Like many other mechanical or electrical systems, electric motor characteristics vary among individuals in the same model and from the same manufacturing process. Characteristics may also change with service life and environmental factors such as temperature. This is a serious potential problem for the EPAS application considering the automotive mass production, the expected long service life, and the harsh working environment. Therefore, it is critical for the EPAS to be equipped with a controller that is able to achieve the assisting torque generation task and be robust to motor parameter variation.

The motor parameters most likely to vary include coil resistance, coil inductance, torque constant. The variations directly impart inaccuracies to the model-based control scheme due to its use of the nominal parameter values. In many practical applications of motor control, it is often the case that cost and design considerations prohibit the use of sensors placed directly on the motor windings or the magnets to monitor parameter variations. To ensure adequate torque control and acceptable frequency domain

performance, it is desirable to compensate the controller for variations in motor parameters. Usually, the parameters change orders of magnitude slower than the motor electrical dynamics. Adaptive control appears to be a favorable choice for the sinusoidal BLDC motor application in EPAS.

In general, an adaptive controller consists of a control law that is designed using nominal plant parameters, and an adaptation law that estimates and updates the parameters using plant states and/or output feedback, as shown in Figure 1.5. The adaptation keeps the control law updated for the varying characteristics of the plant, and maintains desirable closed loop performance even when the plant is changing. Adaptive control is a well-developed branch of control theory. There exist a large number of references covering the topics of parameter estimation and adaptive control (Ioannou and Sun 1996, Tao 2004, Ljung 1998, etc). Limited space in this dissertation would not allow a survey of all these adaptive control techniques. However, some basic concepts will be discussed, and the focus will be on some specific adaptive techniques that are potential candidates for the sinusoidal BLDC motor control in EPAS applications.

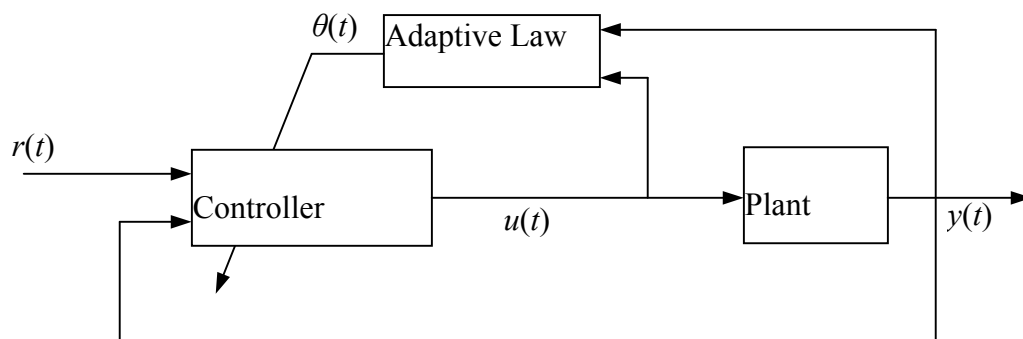


Figure 1.5 Typical adaptive controller block diagram

Depending on how the control law and adaptation law are designed, adaptive control can be classified as direct or indirect. In the indirect adaptive control approach, the plant parameters are used explicitly to calculate the controller parameters. The adaptation law estimates the plant parameters, and therefore indirectly updates the controller. In contrast, the direct adaptive control engages the plant model in the controller parameters implicitly. The adaptation adjusts the control law parameters without calculating the plant parameters. In automotive control applications, it is often desirable to have on-line plant model information for control and diagnostics purposes. The indirect adaptive control obviously fits this requirement better.

Model reference adaptive control is one of the main approaches to adaptive control. A reference model is designed for ideal performance of the closed loop system with consideration of the plant dynamics. The controller drives the plant output to track the output from the reference model. The adaptation law uses the controller command, plant output and the reference model trajectory tracking error to update the controller parameters. In the EPAS application, the ideal case is that the BLDC motor performs like a brushed DC motor. This indicates that the MRAC is a likely a candidate for the BLDC motor control, and the reference model is that of a brushed DC motor.

The sinusoidal BLDC motor is a multi-input multi-output (MIMO) nonlinear system and time varying as shown in its dq coordinate model (1.1). Among the adaptive techniques, Lyapunov's Direct method was often used to derive the adaptation law for nonlinear systems and to prove the stability of the closed loop system. The Extended Kalman Filter (EKF) has been an effective way for both state and parameter estimation in nonlinear systems. It calculates the optimal Kalman gains on-line for the linearization of

the system, and is relatively optimal at steady state. These two methods are candidates for the adaptation law for the sinusoidal BLDC motor adaptive control. In recent years, neural networks were presented in numerous publications for their application in adaptive control (Rovithakis 1999, Patino and Liu 2000). Various types of neural networks were engaged to approximate the unknown plant dynamics and the network weight factors were adaptively updated on-line. However, the sinusoidal BLDC motor model (1.1) has been widely used and proved effective for modeling its dynamics. It is not necessary to use such a “black box” model, therefore this method is not considered in this research.

In the following paragraphs, a few references published in the past decade will be reviewed, which engaged the adaptive technique candidates for the sinusoidal BLDC motor application.

Zhang et al. (Zhang et al. 2000) studied a class of first-order nonlinearly parameterized systems. By utilizing a special property of the systems considered, an integral-type Lyapunov function was introduced to construct a Lyapunov-based controller and parameter updating laws. It was shown that globally asymptotic tracking could be achieved, and explicit transient bounds on the tracking error were provided for different choices of Lyapunov functions.

Hotzel and Karsenti (Hotzel and Karsenti 1998) presented an adaptive feedback tracking strategy for a class of uncertain single-input/single-output systems in strict parametric feedback form with nonlinear time-varying parameterization. The tracking scheme was based on a backstepping design. A local stability result was obtained via Lyapunov arguments.

An adaptive control scheme was proposed (Marino and Tomei 1999) to solve the asymptotic tracking output feedback problem for a class of observable, minimum phase, nonlinear systems with output dependent nonlinearities multiplying time-varying parameters. Proof of asymptotical stability was achieved by showing Lyapunov stability of the system.

Loh et al. (Loh et al. 2003) proposed an adaptive controller that involved two tuning functions that were determined by a mini-max optimization approach. The proposed algorithm was shown to be Lyapunov stable and capable of achieving zero tracking error in steady state.

Zhang et al. (Zhang et al. 2003) proposed a backstepping controller for linear time varying (LTV) systems with known and unknown parameters. The controller was derived by a series of Lyapunov candidate functions, global stability was guaranteed by choosing certain design parameters properly.

Zhang and Ioannou (Zhang and Ioannou 2000) presented a new certainty equivalence based adaptive controller by a combining backstepping based control law with a normalized adaptive law. The new adaptive controller guaranteed stability and performance, as well as parametric robustness for the non-adaptive controller without the use of higher order nonlinearities.

Liao and Chien (Liao and Chien 2000) presented an exponentially stable adaptive compensation for Coulomb friction in a simple servo control system. Stability was proved using the Lyapunov stability theorem. The proposed scheme provided exponential convergence for the Coulomb friction coefficient estimation and state tracking errors even without persistency of excitation.

Kosmatopoulos and Ioannou (Kosmatopoulos and Ioannou 2002) proposed a switching adaptive controller for multi-input nonlinear systems whose dynamics were nonlinearly affected by external input disturbances. By making use of the notion of robust control Lyapunov functions and a modified version of the switching adaptive controller it was shown that the proposed controller guaranteed bounded closed loop signals and convergence of the state to a residual set.

Limanond and Tsakalis (Limanond and Tsakalis 2000) addressed the model reference adaptive control problem of linear time-varying plants. A gradient-based adaptive law with projection and normalization was derived to estimate the unknown controller parameters. It was shown that, for a class of possibly fast time-varying plants, boundedness of the closed loop signals and small tracking errors in the mean-square normalized sense could be achieved, provided that only the unstructured part of the desired controller was slowly time-varying.

Qu (Qu 2002) proposed that, despite of their nonlinearity and time variance, uncertainties or their bounding functions could be estimated as long as they were generated by exosystems whose models were either known or partially known. This was realized by finding a control algorithm that satisfied $\dot{Q} < 0$ of the Lyapunov candidate function, Q .

Milman and Bortoff (Milman and Bortoff 1999) presented an observer based adaptive control through backstepping control approach, which ensured asymptotical stability. Experimental comparison with a full state feedback controller showed better transient performance and smaller steady state error of this observer-based controller.

Jiang and Hill (Jiang and Hill 1999) presented a constructive robust adaptive nonlinear control scheme that could be classified as a robustification of the adaptive backstepping algorithm. Simulations of a simple pendulum with unknown parameters and without velocity measurement illustrated the performance of the controller.

Gobbo et al. (Gobbo et al. 2001) proposed a sensor failure detection and identification scheme by using an EKF to estimate the fault-related parameters, which were processed by a decision algorithm to detect possible failures. Experimental results, by applying different types of failures on the sensors of the inverted pendulum, validated the effectiveness of the approach. Zein et al. (Zein et al.) presented an efficient discrete-time second-order model of an induction motor for the rotor flux and real-time parameter estimation using an EKF. Experimental results showed great accuracy and fast convergence of the estimated parameters.

In conclusion, Lyapunov's Direct method is still one of the most commonly used ways for on-line parameter estimation in various latest adaptive control applications. The Extended Kalman Filter appears to be an option for deriving indirect adaptation law for many nonlinear systems. MRAC has been one of the main approaches to adaptive control. These methods will be evaluated for the indirect adaptive control application on the sinusoidal BLDC motor.

1.6 Publications

Three journal papers and two conference papers have been published during the author's doctoral study. The single parameter estimation algorithm using the q -axis current dynamics was presented in a SAE conference paper (Zhu and Patankar 2004). The multi-parameter estimation algorithm using the q -axis current dynamics process was

presented in American Control Conference 2004 (Patankar and Zhu 2004). The multi-parameter estimation algorithm with the Gram-Schmidt process was published in the International Journal of Vehicle Automation Systems (Zhu and Patankar 2006). The parameter estimation algorithms can also be implemented for diagnostics and actuator health monitoring. A paper on this topic was published in the International Journal of Automation and Control (Patankar and Zhu 2007). Another paper about modeling and simulation of a single cylinder internal combustion engine (Chiang, Zhu and Patankar 2007) was published on the Trends in Applied Sciences Research Journal.

1.7 Summary

Electrical Power-Assisted Steering systems will likely be used for the future power steering systems because of its advantage of energy efficiency, flexibility and reliability. The sinusoidal brushless DC motor has been identified as the most suitable candidate actuator for the EPAS. The long service life, harsh working environment and mass production impose motor parameter variation problem for the EPAS actuator controller. Adaptive control is an ideal technique to address this problem while achieving the control goals. Specifically, the indirect adaptive control, the model reference adaptive control, the Lyapunov method, and the Extended Kalman Filter are considered as candidate adaptive techniques for the sinusoidal BLDC motor application. They will be explored in detail in later chapters.

The dissertation is organized as following: Chapter 2 will discuss the dq -model of the sinusoidal BLDC motor in detail. Based on the dq -model, Chapter 3 will present several indirect adaptive algorithms that parameter are estimated by solving algebraic equations formulated by several loops current feedback. In Chapter 4, recursive least square

algorithm and extended Kalman filter will be derived for the parameter estimation problem. Chapter 5 will present a model reference adaptive controller for the BLDC motor application. In Chapter 6, the algorithms developed in Chapter 3 to Chapter 5 are compared in closed loop simulation of an EPAS model and a speed control application. Chapter 7 will discuss some practical control implementation issues through simulation of a quasi-physical model of the BLDC motor system, including components such as the inverter, the space vector pulse width modulation (SVPWM), and etc. Chapter 8 will conclude the dissertation and discuss some future research directions.

Chapter 2 SINUSOIDAL BRUSHLESS DC MOTOR MODELING

A BLDC motor is the combination of a permanent magnet synchronous motor (PMSM), an H-bridge DC-AC inverter, a rotor position feedback mechanism, and a digital controller. From a users perspective, the motor only needs DC power and does not have commutation devices such as brushes, thus it is called a Brushless DC motor. The controller is usually designed in self-controlled mode. Together with the H-bridge inverter, it generates AC current in each phase of the permanent magnet synchronous motor with a DC power supply. Assuming ideal operation of the inverter, a BLDC motor is actually a PMSM motor. The direct quadrature (*dq*) model, which is a well-established model for AC induction motor and synchronous motor, can be used for the BLDC motor dynamics analysis and control design.

Since this research mainly concentrates on the sinusoidal BLDC motor, and considering the fact that most sinusoidal BLDC motors are three-phase Y-connected, a *dq*-axis model for such motors will be derived in this chapter, and torque/current control algorithms based on this model will be discussed.

2.1 Permanent Magnet Synchronous Motor Modeling

The actual motor in a sinusoidal BLDC Motor is a permanent magnet synchronous motor, which consists of a rotor with permanent magnets and several phases of sinusoidally distributed stator windings. A 3-phase PMSM model is presented in this section and a *dq*-coordinate model is derived from the 3-phase model.

2.1.1 The three-phase model

The electrical dynamics of a three-phase PMSM motor can be modeled as (Khorrami 2003, Krause 2002):

$$\mathbf{V}_{123} = R\mathbf{i}_{123} + \frac{d\mathbf{\Phi}_{123}}{dt} \quad (2.1)$$

where $\mathbf{V}_{123} = [V_1, V_2, V_3]^T$, $\mathbf{i}_{123} = [i_1, i_2, i_3]^T$, and $\mathbf{\Phi}_{123} = [\Phi_1, \Phi_2, \Phi_3]^T$ are the vectors of the phase voltages, currents, and fluxes respectively. R is the phase resistance. The flux linkage vector $\mathbf{\Phi}_{123}$ is given as

$$\mathbf{\Phi}_{123} = \mathbf{L}_{123}\mathbf{i}_{123} + \mathbf{L}_{123f}i_f, \quad (2.2)$$

where

$$\mathbf{L}_{123} = \begin{bmatrix} L_{11} & L_{12} & L_{13} \\ L_{21} & L_{22} & L_{23} \\ L_{31} & L_{32} & L_{33} \end{bmatrix} \quad (2.3)$$

is the inductance matrix of the stator coils and

$$\mathbf{L}_{123f} = \begin{bmatrix} L_{1f} \\ L_{2f} \\ L_{3f} \end{bmatrix} \quad (2.4)$$

are the equivalent inductance of the permanent magnet on the rotor. The quantity i_f is the fictitious current due to the permanent magnet. The inductance terms in (2.3) and (2.4) can be calculated as

$$\begin{aligned} L_{11} &= L_a + L_g \cos(2n_p\theta), \\ L_{22} &= L_a + L_g \cos(2n_p\theta + \frac{2\pi}{3}), \\ L_{33} &= L_a + L_g \cos(2n_p\theta - \frac{2\pi}{3}), \end{aligned}$$

$$\begin{aligned}
L_{12} = L_{21} &= -\frac{L_a}{2} + L_g \cos(2n_p \theta - \frac{2\pi}{3}), \\
L_{13} = L_{31} &= -\frac{L_a}{2} + L_g \cos(2n_p \theta + \frac{2\pi}{3}), \\
L_{23} = L_{32} &= -\frac{L_a}{2} + L_g \cos(2n_p \theta), \\
L_{1f} &= L_{m0} + L_{m1} \cos(n_p \theta), \\
L_{2f} &= L_{m0} + L_{m1} \cos(n_p \theta - \frac{2\pi}{3}), \\
L_{3f} &= L_{m0} + L_{m1} \cos(n_p \theta + \frac{2\pi}{3}).
\end{aligned} \tag{2.5}$$

where L_a , L_g , L_{m0} , and L_{m1} are positive constants, n_p is the number of magnet pole pairs, and θ is the rotor position. The first terms of these inductance terms stand for the inductance within the loop formed by the stator iron and air gap between the stator and the rotor, and they are invariant to rotor position. The second terms represent the inductance in the loop formed by the stator iron and the permanent magnets. They are sinusoidal functions of the rotor angle because the permanent magnets are rotating with the rotor.

The electromagnetic torque generated by the motor can be calculated as

$$\tau = \frac{1}{2} \mathbf{i}_{123f}^T \frac{\partial \mathbf{L}}{\partial \theta} \mathbf{i}_{123f} \tag{2.6}$$

where and \mathbf{L} is the inductance matrix

$$\mathbf{L} = \begin{bmatrix} L_{123} & L_{123f} \\ L_{123f}^T & L_{ff} \end{bmatrix} \tag{2.7}$$

and L_{ff} is a positive constant associated to permanent magnets.

It is noteworthy that the inductance matrix in (2.3) and (2.7) are symmetrical, and the inductance terms contain sinusoidal functions of the rotor angle with a phase shift of $\frac{2\pi}{3}$

to each other. This is due to the fact that the stator phases are symmetrically distributed in the stator cylindrical inner surface, and the stator phases are identical in terms of their magnetic and electrical characteristics.

Using (2.5), (2.7) can be expanded as

$$\begin{aligned} \tau = & -n_p L_g \left[i_1^2 \sin(2n_p \theta) + i_2^2 \sin(2n_p \theta - \frac{2\pi}{3}) + i_3^2 \sin(2n_p \theta + \frac{2\pi}{3}) \right] \\ & - 2n_p L_g \left[i_1 i_2 \sin(2n_p \theta - \frac{2\pi}{3}) + i_2 i_3 \sin(2n_p \theta) + i_3 i_1 \sin(2n_p \theta + \frac{2\pi}{3}) \right] \\ & - n_p L_{m1} i_f \left[i_1 \sin(n_p \theta) + i_2 \sin(n_p \theta - \frac{2\pi}{3}) + i_3 \sin(n_p \theta + \frac{2\pi}{3}) \right]. \end{aligned} \quad (2.8)$$

The equations (2.1) and (2.8) can be directly used for AC synchronous motors in which the phase currents and voltages are sinusoidal at a fixed frequency that is equivalent to the rotor magnetic field speed. However, they are not convenient for varying speed applications such as that of BLDC motors, as the rotor angle is explicitly involved in these equations. Therefore, it is desirable to simplify these equations. The most commonly used method is to transform them from the stator phase coordinates into the direct quadrature (dq) coordinates.

2.1.2 The direct quadrature coordinate model

Before performing coordinate transformations, let us see how the dq coordinates are defined. Let us use a three phase single magnet PMSM motor for illustration, as shown in Fig 2.1. The dq coordinates stand for the direct quadrature coordinates, and this coordinate system is fixed on the rotor magnet. The direct (d) axis is aligned with the magnet north pole axis, and the quadrature (q) axis is 90 degree counter-clockwise to the d axis. Similarly, a coordinate system, denoted as the ab coordinates, is defined as the

fixed frame on the stator. The a axis is aligned to the phase 1 axis, and the b axis is 90 degree counter-clockwise to the a axis. The origins in both coordinates are the center of the rotor. In this example, the rotor angle θ is the angular displacement between the dq coordinates and the ab coordinates. If the rotor is equipped with more than one pair of permanent magnets, it can be modeled with an equivalent single pair magnet rotor with rotor angles of $n_p\theta$, where n_p is the number of magnet pairs.

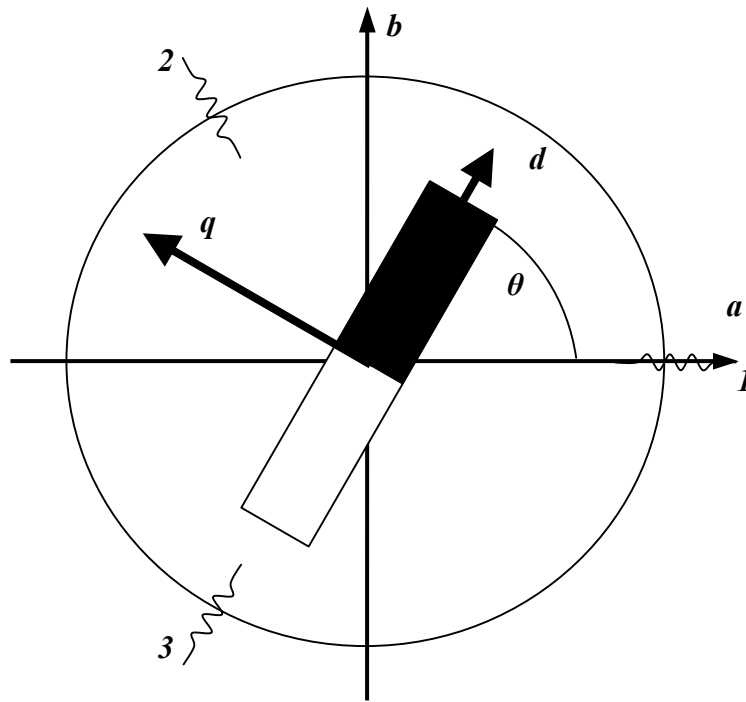


Figure 2.1 The three stator phase, the qd coordinates and the ab coordinates

In the configuration shown in Figure 2.1, the zero angle ($\theta=0$) is the rotor position where the d axis is aligned to the a axis. However, the zero angle rotor position is not unique and can be defined in other configurations. For example, it could be defined when the q and a axis are aligned. The zero-angle position may seem to be trivial for the

control and analysis in the dq coordinates. However, it is essential for the synchronization between the stator the rotor magnetic fields. It determines the phase of almost all the sinusoidal functions involved in the (dq - ab) coordinate transformation and its inverse transformation. The definition of the zero-angle position must be consistent in the coordinate transformations especially when implementing the controller designed in the qd coordinates. In the following sections, the $a-d$ alignment configuration show in Figure 2.1 is adopted.

To simplify the phase model (2.1), let us expand the flux terms by substituting (2.3~2.5) into (2.1),

$$\mathbf{L}_{123} \frac{d\mathbf{i}_{123}}{dt} + \omega \frac{\partial \mathbf{L}_{123}}{\partial \theta} \mathbf{i}_{123} + \omega i_f \frac{\partial \mathbf{L}_{123f}}{\partial \theta} = \mathbf{V}_{123} - R\mathbf{i}_{123}. \quad (2.9)$$

Notice that

$$[1 \ 1 \ 1] \cdot \mathbf{L}_{123} = [L_{11} + L_{21} + L_{31} \quad L_{12} + L_{22} + L_{32} \quad L_{13} + L_{23} + L_{33}] = [0 \ 0 \ 0], \quad (2.10a)$$

and similarly

$$[1 \ 1 \ 1] \cdot \frac{\partial \mathbf{L}_{123f}}{\partial \theta} = 0. \quad (2.10b)$$

In addition, since the neutral point is not accessible in almost all Y-connected BLDC motors, according to Kirchhoff first law, the sum of phase currents must be zero:

$$[1 \ 1 \ 1] \cdot \mathbf{i}_{123} = 0. \quad (2.10c)$$

Apparently, the summation of all three equations in (2.9) gives a trivial equation with zero on both sides. This indicates that only two of the three phase equations are independent, and it is possible to transform them into simpler forms.

The currents, voltages and fluxes in the three-stator phases can be considered as vectors in the stator fixed \mathbf{ab} coordinates. They can be projected to the \mathbf{a} and \mathbf{b} axes with a transformation

$$f_{0ab} = \mathbf{T}_c f_{123}, \quad (2.11)$$

where f standing for i , v , or Φ . $\mathbf{f}_{0ab} = [f_0 \ f_a \ f_b]^T$ is the vector of transformed quantities. The transformation matrix \mathbf{T}_c is given by

$$\mathbf{T}_c = \sqrt{\frac{2}{3}} \begin{bmatrix} \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \\ 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & -\frac{\sqrt{3}}{2} & \frac{\sqrt{3}}{2} \end{bmatrix}. \quad (2.12)$$

Notice that the transformation matrix satisfies $\mathbf{T}_c^T \mathbf{T}_c = \mathbf{I}$. The coefficient $\sqrt{\frac{2}{3}}$ ensures that the transformation maintains energy conservation. Through the transformation (2.11), the inductance matrix \mathbf{L}_{123} becomes

$$\mathbf{L}_{0ab} = \mathbf{T}_c \mathbf{L}_{123} \mathbf{T}_c^T = \frac{3}{2} \begin{bmatrix} 0 & 0 & 0 \\ 0 & L_a + L_g \cos(2n_p \theta) & -L_g \sin(2n_p \theta) \\ 0 & -L_g \sin(2n_p \theta) & L_a - L_g \cos(2n_p \theta) \end{bmatrix}. \quad (2.13)$$

Also,

$$\mathbf{L}_{0abf} = \mathbf{T}_c \mathbf{L}_{123f} = \begin{bmatrix} \sqrt{3} L_{m0} \\ \sqrt{\frac{3}{2}} L_{m1} \cos(n_p \theta) \\ -\sqrt{\frac{3}{2}} L_{m1} \sin(n_p \theta) \end{bmatrix}. \quad (2.14)$$

The generated torque τ can be written in the new coordinates as

$$\begin{aligned}
\tau &= \frac{1}{2} \mathbf{i}_{123}^T \mathbf{T}_c^T \frac{\partial \mathbf{T}_c \mathbf{L}_{123} \mathbf{T}_c^T}{\partial \theta} \mathbf{T}_c \mathbf{i}_{123} + \mathbf{i}_{123}^T \mathbf{T}_c^T \frac{\partial \mathbf{T}_c \mathbf{L}_{123f}}{\partial \theta} i_f \\
&= \frac{1}{2} \mathbf{i}_{0ab}^T \frac{\partial \mathbf{L}_{0ab}}{\partial \theta} \mathbf{i}_{0ab} + \mathbf{i}_{0ab}^T \frac{\partial \mathbf{L}_{0abf}}{\partial \theta} i_f \\
&= \frac{3n_p L_g}{2} \sin(2n_p \theta) (-i_a^2 + i_b^2) - 3n_p L_g i_a i_b \cos(2n_p \theta) \\
&\quad + \frac{\sqrt{6} n_p L_{m1} i_f}{2} [-i_a \sin(n_p \theta) - i_b \cos(n_p \theta)]
\end{aligned} \tag{2.15}$$

The electrical dynamics can be expressed in the $0ab$ coordinates as

$$\mathbf{L}_{0ab} \frac{d\mathbf{i}_{0ab}}{dt} + \omega \frac{\partial \mathbf{L}_{0ab}}{\partial \theta} \mathbf{i}_{0ab} + \omega i_f \frac{\partial \mathbf{L}_{0abf}}{\partial \theta} = \mathbf{V}_{0ab} - R \mathbf{i}_{0ab}. \tag{2.16}$$

The first equation in the 3×1 vector equation (2.16) is algebraic since the elements of the first row of \mathbf{L}_{0ab} are zero, and, as shown in (2.10c), $i_0 = \sqrt{\frac{1}{3}}(i_1 + i_2 + i_3) = 0$.

Therefore, the first equation can be ignored in the analysis of the dynamics and in the control design. The second and third equations provide differential equations that govern the dynamics of i_a and i_b :

$$\mathbf{L}_{ab} \begin{bmatrix} \dot{i}_a \\ \dot{i}_b \end{bmatrix} = \begin{bmatrix} v_a \\ v_b \end{bmatrix} - R \begin{bmatrix} i_a \\ i_b \end{bmatrix} - 3n_p L_g \omega \begin{bmatrix} i_a \sin(2n_p \theta) + i_b \cos(2n_p \theta) \\ i_a \cos(2n_p \theta) - i_b \sin(2n_p \theta) \end{bmatrix} - \frac{\sqrt{6}}{2} n_p L_{m1} \omega i_f \begin{bmatrix} \cos(n_p \theta) \\ -\sin(n_p \theta) \end{bmatrix} \tag{2.17}$$

with

$$\mathbf{L}_{ab} = \frac{3}{2} \begin{bmatrix} L_a + L_g \cos(2n_p \theta) & L_g \sin(2n_p \theta) \\ L_g \sin(2n_p \theta) & L_a - L_g \cos(2n_p \theta) \end{bmatrix}. \tag{2.18}$$

The original three phase model (2.1) and (2.8) are transformed into the simpler forms of (2.15) and (2.17) with only two dynamic equations. However, the rotor angular displacement θ still explicitly presents in the equations. These equations are still inconvenient for analysis and control design.

Recall that the dq coordinate system is fixed on the rotor, and the projection from the ab coordinates to the dq coordinates contain the sinusoidal function of θ . It is possible that the position dependence of the torque expression (2.15) and the electrical dynamics (2.17) can be eliminated through the ab - dq projection, which is given by a transformation

$$\mathbf{f}_{0qd} = \mathbf{T}_p \mathbf{f}_{0ab}. \quad (2.19)$$

where

$$\mathbf{T}_p = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(n_p \theta) & \sin(n_p \theta) \\ 0 & -\sin(n_p \theta) & \cos(n_p \theta) \end{bmatrix}, \quad (2.20)$$

and similar to (2.11).

Combine (2.11) and (2.19), The dq coordinate variables can be obtained from the original three phase variables by

$$\mathbf{f}_{0qd} = \mathbf{T}_{dq} \mathbf{f}_{123}. \quad (2.21)$$

with

$$\mathbf{T}_{dq} = \mathbf{T}_p \mathbf{T}_c = \sqrt{\frac{2}{3}} \begin{bmatrix} \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \\ \cos(n_p \theta) & \cos(n_p \theta - \frac{2\pi}{3}) & \cos(n_p \theta + \frac{2\pi}{3}) \\ -\sin(n_p \theta) & -\sin(n_p \theta - \frac{2\pi}{3}) & -\sin(n_p \theta + \frac{2\pi}{3}) \end{bmatrix}. \quad (2.22)$$

Notice that $\mathbf{T}_{dq}^T \mathbf{T}_{dq} = \mathbf{I}$, which means its inverse is equal to its transpose. Using the transformation (2.22), the inductance matrix in the dq coordinates can be obtained as

$$\mathbf{L}_{0dq} = \mathbf{T}_{dq} \mathbf{L}_{123} \mathbf{T}_{dq}^T = \begin{bmatrix} 0 & 0 & 0 \\ 0 & \frac{3}{2}(L_a + L_g) & 0 \\ 0 & 0 & \frac{3}{2}(L_a - L_g) \end{bmatrix}, \quad (2.23)$$

and the inductance terms for the rotor permanent magnet in the dq coordinates become

$$\mathbf{L}_{0dqf} = \mathbf{T}_{dq} \mathbf{L}_{123f} = \sqrt{\frac{2}{3}} \begin{bmatrix} \frac{3\sqrt{2}}{2} L_{m0} \\ \frac{3}{2} L_{m1} \\ 0 \end{bmatrix}. \quad (2.24)$$

Notice that both \mathbf{L}_{0dq} and \mathbf{L}_{0dqf} do not explicitly depend on the rotor position, and they are much simpler than their counterparts in the phase model as shown in (2.3) and (2.4).

To obtain the dynamic equations in the dq coordinates, care must be taken since the transformation matrix \mathbf{T}_{dq} consists of functions of the rotor angle. Note that

$$\frac{d\Phi_{123}}{dt} = \frac{d(\mathbf{L}_{123}\mathbf{i}_{123} + \mathbf{L}_{123f}i_f)}{dt} = \frac{d\mathbf{T}_{dq}^T (\mathbf{L}_{0qd}\mathbf{i}_{0qd} + \mathbf{L}_{0qdf}i_f)}{dt}, \quad (2.25)$$

and recall that i_f is a constant, the electrical dynamics in the dq coordinates can be derived from (2.1) as

$$\mathbf{L}_{0qd} \frac{d\mathbf{i}_{0qd}}{dt} + \mathbf{T}_{dq} \frac{d\mathbf{T}_{dq}^T}{dt} (\mathbf{L}_{0qd}\mathbf{i}_{0qd} + \mathbf{L}_{0qdf}i_f) = \mathbf{V}_{0qd} - R\mathbf{i}_{0qd}. \quad (2.26)$$

Notice that

$$\mathbf{T}_{dq} \frac{d\mathbf{T}_{dq}^T}{dt} = n_p \omega \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix}. \quad (2.27)$$

Again, the first equation in the 3×1 vector equations (2.26) is algebraic, and therefore can be ignored. The dynamic equations in the dq coordinates can be found as

$$\begin{aligned} L_q \frac{di_q}{dt} &= -Ri_q - n_p \omega L_d i_d - K_e \omega_m + v_q, \\ L_d \frac{di_d}{dt} &= -Ri_d + n_p \omega L_q i_q + v_d. \end{aligned} \quad (2.28)$$

Similarly, the torque expression in (2.15) can be transformed into the dq coordinates as

$$\tau = 3n_p L_g i_d i_q + \frac{\sqrt{6}}{2} n_p L_{m1} i_f i_q. \quad (2.29)$$

Introducing the notation

$$\begin{aligned} L_d &= \frac{3}{2}(L_a + L_g), \\ L_q &= \frac{3}{2}(L_a - L_g), \\ K_\tau &= 3n_p L_g, \\ K_e &= \sqrt{\frac{3}{2}} n_p L_{m1} i_f, \end{aligned} \quad (2.30)$$

the electrical dynamic model of the BLDC motor transformed into the dq coordinates as

$$\begin{aligned} L_q \frac{di_q}{dt} &= -R i_q - n_p \omega L_d i_d - K_e \omega_m + v_q, \\ L_d \frac{di_d}{dt} &= -R i_d + n_p \omega L_q i_q + v_d, \\ \tau &= K_\tau i_d i_q + K_e i_q. \end{aligned} \quad (2.31)$$

In the torque equation, the first term $K_\tau i_d i_q$ indicates the contribution of the stator induction flux, while the second term $K_e i_q$ is the contribution of the permanent magnet flux. In the normal motor speed range, the permanent magnets play the dominant role in the flux linkage. Hence, in many applications the term $K_\tau i_d i_q$ is much lower than the $K_e i_q$, and therefore can be neglected. The torque equation becomes

$$\tau = K_e i_q. \quad (2.32)$$

The dq coordinate model of (2.31) is position independent and therefore more convenient than the 3-phase model of (2.1) for analysis and control purpose. It has been used in many references for BLDC motor control design. Similarly, all control algorithms developed in later chapters will be base on this dq -coordinate model.

2.1.3 Implementation of dq coordinate controllers

Before moving onto the control design topics, the practical implementation of a controller designed on the dq coordinate model deserves a little bit more attention.

The dq coordinate model in (2.31) is position independent, thus it is friendly for control development. However, all physically accessible variables from a motor, such as current, voltage and back EMF, are in the stator phase domain. To practically implement a controller designed on the dq coordinate model, the coordinate transformations in (2.12) and (2.20) must be done in real time. Thus, an actual BLDC motor control system needs a few more functions than the dq coordinate controller alone, as shown in Figure 2.2. The measured phase currents must be transformed into currents in dq coordinates (i_d, i_q) through two coordinate transformations. Then the controller output voltages (v_d, v_q) must be transformed back to phase voltage for the real motor. The phase voltages are sinusoidal functions of the rotor angular position. Usually some special modulation method such as the space vector pulse width modulation (SVPWM) is involved to implement these alternative voltages from the DC power supply. The modulation requires well-coordinated software logic and power electronics hardware operation.

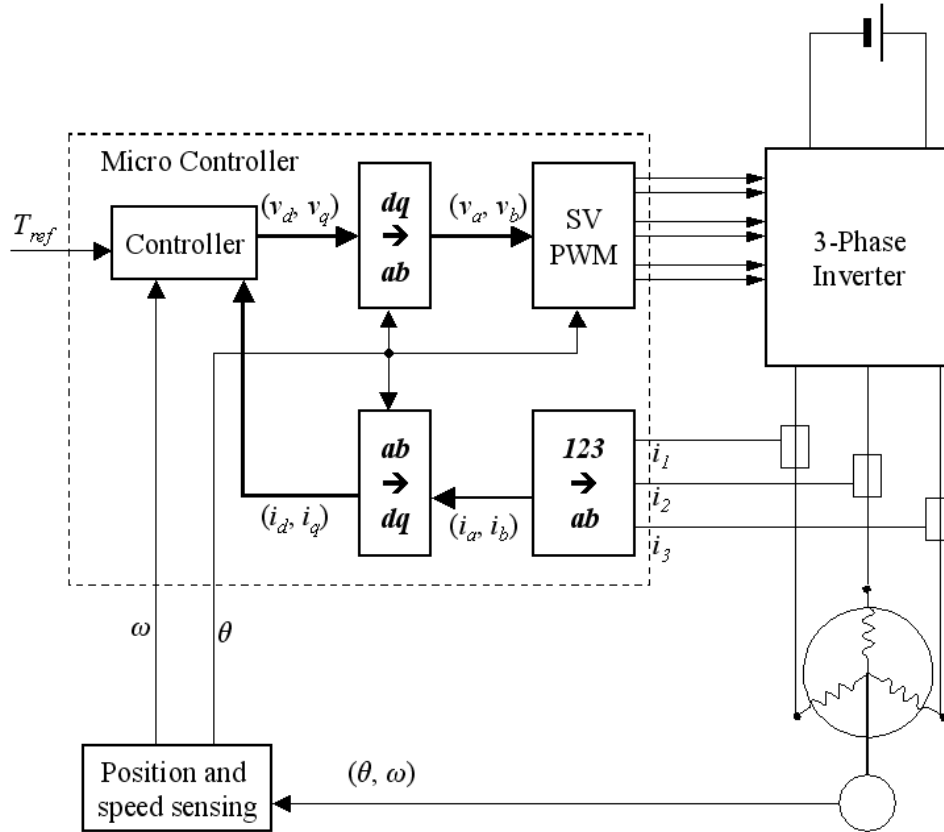


Figure 2.2 Implementation of a generic controller designed in dq coordinates

Even though the dq coordinate model (2.31) does not explicitly depend on the rotor position, almost all the supplement functions require the rotor angle (θ) measurement in real time. Therefore, accurate θ measurement is critical for operation of the BLDC motor and implementation of controllers designed on the dq coordinate model. Usually, θ is measured by using transducers such as optical encoders or resolvers. The measure angular position can be absolute or relative, but care must be taken when calculating the initial value of θ . Recall that the coordinate transformations in (2.12) and (2.20) are based on the zero-angle configuration in which the d and a axis are aligned. When implementing those transformation calculations in the controller, the initial value of the

angle feedback must be consistent with this zero-angle configuration. Transducer resolution is another factor for the implementation of controllers due to the real time coordinate transformations. Only if the transducer resolution is high enough, the coordinate transformations can be considered ideal and transparent to the controller.

For the control design purpose, we shall assume that these supplemental functions for controller implementation are ideal. However, we need to be aware of the uniqueness of the BLDC motor controller implementation comparing to regular implementation of a digital controller to an analog system.

2.2 Torque Control of Sinusoidal BLDC Motors

When a motor is used as the actuator in a control system, usually the input to the motor controller is a command torque from the outer loop. With the dq coordinate BLDC motor model, the q -axis current determines the motor torque shown in (2.32). Given a command torque τ_{cmd} , the desired q -axis current can be calculated as $i_{q,cmd} = \frac{\tau_{cmd}}{K_e}$.

In many applications, it is a common practice to force i_d to as low as possible in amplitude. In another words, the desired d -axis current should be zero, i.e. $i_{d,cmd} \equiv 0$. In some applications, since the q -axis current dominates the motor torque, the desired d -axis current can be used to serve other purposes such as a random excitation signal for identification purpose.

Thus, the torque control problem is equivalent to the current control problem where the desired currents are determined the outer loop torque command and other factors. Various control techniques can be utilized to achieve the current tracking tasks. A PI controller is probably the most popular candidate, especially in industrial applications.

2.2.1 PI controller

The performance of a Proportional and Integral (PI) feedback controller mainly depends on its parameters, namely the proportional gain (k_p) and the integral gain (k_i). The gains are usually designed on a linear time invariant (LTI) model of the plant. If the plant is a nonlinear system, a PI controller may be designed for the linearized system model around some equilibrium states, and the PI gains can be programmed as outputs from lookup tables, which are driven by system states. In addition, a feedforward term is often engaged to account for the nonlinearity of the plant system.

To design a PI controller, the BLDC motor model is restated here for the control design purpose,

$$L_q \frac{di_q}{dt} = -Ri_q - n_p \omega L_d i_d - K_e \omega + v_q,$$

$$L_d \frac{di_d}{dt} = -Ri_d + n_p \omega L_q i_q + v_d.$$

The above equations are nonlinear since the second term on the right hand side of each equation involves product of states ω , i_d and i_q . The variables of interests are i_d and i_q .

Linearize the above equations with respect to the i_d and i_q ,

$$L_q \frac{d(i_{q0} + \Delta i_q)}{dt} = -R(i_{q0} + \Delta i_q) - n_p \omega_0 L_d (i_{d0} + \Delta i_d) - K_e \omega_0 + (v_{q0} + \Delta v_q),$$

$$L_d \frac{d(i_{d0} + \Delta i_d)}{dt} = -R(i_{d0} + \Delta i_d) + n_p \omega_0 L_q (i_{q0} + \Delta i_q) + (v_{d0} + \Delta v_d).$$

The equilibrium point is

$$v_{q0} = Ri_{q0} + n_p \omega_0 L_d i_{d0} + K_e \omega_0, \quad (2.34a)$$

$$v_{d0} = Ri_{d0} - n_p \omega_0 L_q i_{q0}. \quad (2.34b)$$

The PI controller is designed with respect to

$$\begin{aligned} L_q \frac{d\Delta i_q}{dt} &= -R\Delta i_q - n_p \omega_0 L_d \Delta i_d + \Delta v_q, \\ L_d \frac{d\Delta i_d}{dt} &= -R\Delta i_d + n_p \omega_0 L_q \Delta i_q + \Delta v_d. \end{aligned} \quad (2.35)$$

Given desired currents $i_{q,cmd}$ and $i_{d,cmd}$, the PI controller shall stabilize the motor current states around these value. In another words, the desired equilibrium point shall be $(i_{q,cmd}, i_{d,cmd})$, and the proportional and integral gains shall be designed to achieve certain dynamic performances such as overshoot, response time etc. The final control law is therefore proposed as

$$v_q = R i_{q,cmd} + n_p \omega_0 L_d i_{d,cmd} + K_e \omega_0 + k_p (i_{q,cmd} - i_q) + \int k_i (i_{q,cmd} - i_q) dt, \quad (2.36a)$$

$$v_d = R i_{d,cmd} - n_p \omega_0 L_q i_{q,cmd} + k_p (i_{d,cmd} - i_d) + \int k_i (i_{d,cmd} - i_d) dt. \quad (2.36b)$$

An example PI controller is designed for a motor ($R = 0.05\Omega$, $L_d = L_q = 0.0001H$, $K_e = 0.05 Nm/A$). The PI gains for the rotor magnetic field rotating speed at $\omega_{e0} = 0$ rad/s, $\omega_{e0} = 50$ rad/s, $\omega_{e0} = 100$ rad/s, $\omega_{e0} = 200$ rad/s are designed using the MATLAB root locus tool.

The PI gains can be implemented with lookup tables that are driven by the motor speed, and the PI controller is often called as a gain-scheduling controller (Ioannou 1996). It provides some adaptation for performance loss caused by system nonlinearities and changing states. The PI controllers are simple and robust to system uncertainties. However, it relies on integration of the error between the feedback states and their respective reference to counteract the disturbances. This sometimes means sacrifice of dynamic performance.

Table 2.1 PI controller gains

ω_{e0} (rad/s)	k_p	k_i
0	0.005	0.25
50	0.005	0.5
100	0.005	1
200	0.005	2.5

2.2.2 Feedforward inverse dynamics controller

To use a BLDC motor as an actuator in a mechanical system such as automotive steering system, it is desirable to have the motor controller be the exact inverse of the motor electric dynamics. Thus the motor will produce the torque requested from the outer loop controller. In another words, the controller and motor together become a unitary-gain feed-forward gain in the control loop. In nearly all practical applications, the motor drives a mechanical inertia as a part of its load. Usually the dynamics of the mechanical system are orders of magnitude slower in comparison to the electrical dynamics of the motor. Therefore, for the control purpose, the electrical dynamics of the motor can be neglected in comparison to the overall dynamics of the system, i.e.

$$L_q \frac{di_q}{dt} = -Ri_q - n_p \omega L_d i_d - K_e \omega + v_q \approx 0,$$

$$L_d \frac{di_d}{dt} = -Ri_d + n_p \omega L_q i_q + v_d \approx 0.$$

Given command currents $i_{q,cmd}$ and $i_{d,cmd}$, the inverse dynamics control law is proposed

as

$$\begin{aligned}
v_q &= K_{ec}\omega_m + R_c i_{q,cmd} + \omega_e L_c i_d, \\
v_d &= R_c i_{d,cmd} - \omega_e L_c i_q.
\end{aligned}
\tag{2.37}$$

Assuming ideal knowledge of the plant, and substituting (2.37) back into the plant model, the closed loop system dynamics becomes

$$\begin{aligned}
L_q \frac{di_q}{dt} &= -R i_q + R i_{q,cmd}, \\
L_d \frac{di_d}{dt} &= -R i_d + R i_{d,cmd}.
\end{aligned}
\tag{2.38}$$

The closed loop system dynamics model (2.38) consists of two first order low pass filters with the time constants determined by the stator phase parameters. For the example motor mentioned before ($R = 0.05\Omega$, $L_d = L_q = 0.0001H$, $Ke = 0.05 Nm/A$), the time constants for both the d and q axis current dynamics are 0.002s. This is usually fast enough for most mechanical systems.

Figure 2.3 and 2.4 compared the step response of the two controllers in simulation. The reference current has a step increase of $i_{qcmd} = 2A$ at 0.2 seconds. Zero rotor movement was assumed for the purpose of current dynamics performance comparison. In the first simulation shown in Figure 2.3, it is assumed that the controller parameters matched the plant parameters ideally. Both controllers provided quick response to the step input when full knowledge of the plant was assumed. The PI controller showed slight overshoot due to the integration of current error.

In Figure 2.4, the controller parameters has 10% error in Ke and 50% error in R comparing to the plant. With parameter errors presented, it took about half seconds for the PI controller to counteract the error and achieve zero steady state error, while the feedforward controller could not compensate the steady stator error caused by the

parameter error.

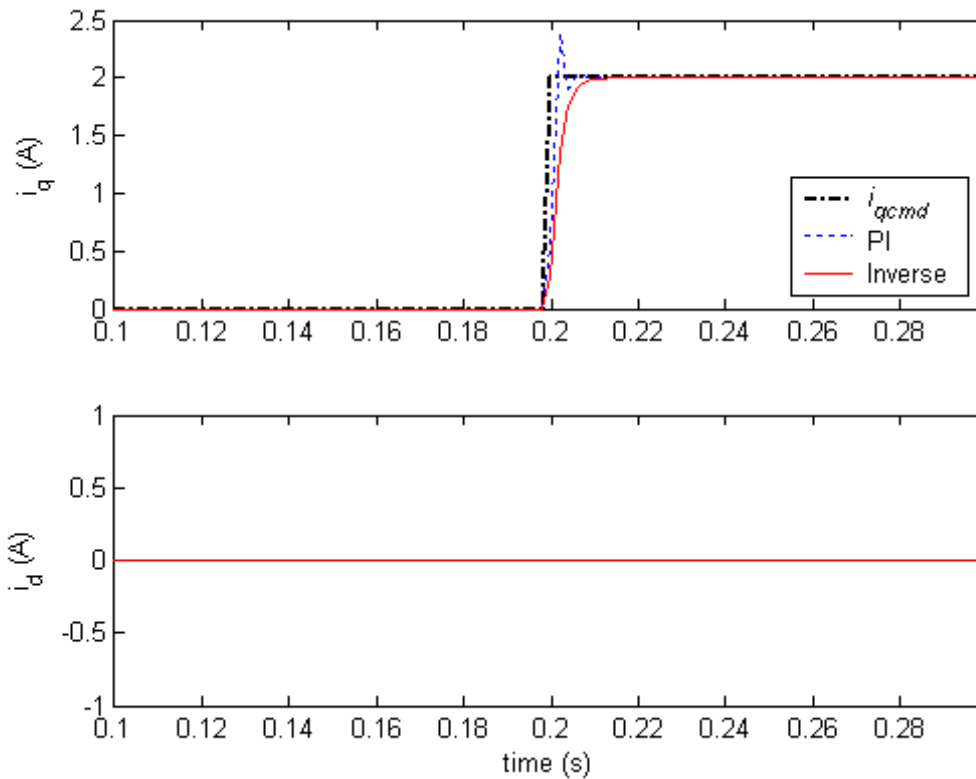


Figure 2.3 Step response of the BLDC motor with the PI controller and the feedforward controller at $\omega_{e0} = 0$, assuming full knowledge of the plant parameters.

2.2.3 Controller selection

The closed loop system performance varied more or less from one controller to another. The controller selection criteria often depend on the specific application. In the case of the automotive power-assisted steering system, it is desirable for the assisting motor to provide a consistent sense of stiffness to the driver. Besides, the chattering effect of assisting torque should be avoided, as human hands are very sensitive to vibrations. Based on the simulation results shown in Figure 2.3 and 2.4, the PI controller may not be

able to provide satisfactory performance in both respects. Given ideal knowledge of the plant, fine-tuning the gains or adding a derivative term may improve the overshoot problem. However, if significant parameter error exist between the controller and the plant, the slow compensation of the PI controller shown in Figure 2.4 would make the driver's feeling of the steering wheel stiffness vary in the scale of seconds. This is highly undesirable.

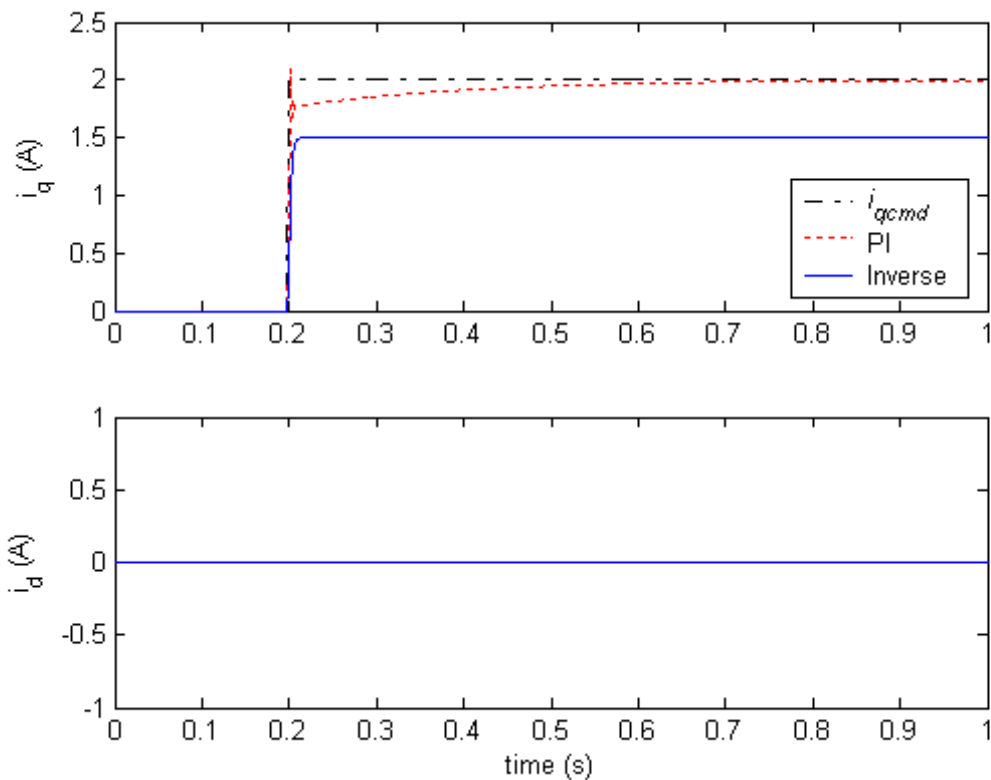


Figure 2.4 Step response of the BLDC motor with the PI controller and the feedforward controller at $\omega_{e0} = 0$, with 10% error in K_e and 50% error in R .

On the other hand, the feedforward motor inverse controller provides favorable

responses for the EPAS application. If there exist parameter value error in the controller, driver may feel that the steering wheel is heavy but it is consistent. In addition, if implemented with some adaptation techniques, where the parameters involved in the control law can be updated with the actual plant, then the feedforward controller provides more desirable performance for EPAS.

Considering the mass production of the EPAS and its expected long service life in such harsh environment, the motor characteristics variation is nearly inevitable across the same product model and over the service life of each individual motor. The adaptive feedforward dynamics inverse controller appears to be a favorable candidate for the BLDC motor in the EPAS application.

2.3 Summary

The *dq*-coordinate model has been widely used for the BLDC motor control design and analysis. One of the main advantages of the *dq*-coordinate model is that the rotor angular position θ is not explicitly involved in the motor dynamics equations. As a result, the control and dynamics analysis using the *dq*-coordinate model are significantly simpler than using the phase model. This chapter re-examined how the *dq*-coordinate model was derived from the three-phase permanent magnet synchronous motor model. The mathematical derivation revealed how θ was cancelled through the coordinate transformation from the three-phase frame to the *dq*-coordinates. It also explained how the *d* and *q* axis dynamics were obtained and their roles in the principle of the BLDC motor operation. In addition to provide a foundation to the control design, the derivation enlightened implementation issues for the controller designed with the *dq*-coordinate model. For example, though the rotor angular position θ is not explicitly engaged in the

dq-coordinate model, it is indispensable in the coordinate transformations that must be done in real time. Therefore the operation of the BLDC motor requires measurement of the rotor angular position θ . Another example is the zero-angle configuration that seems trivial for the control and analysis in the *dq* coordinates. When implementing a controller designed from the *dq*-coordinate model, the zero-angle configuration is critical to ensure the correct calculation of the coordinate transformation and synchronization of the stator and rotor magnetic fields.

With the *dq*-coordinate model, The BLDC motor torque control problem can be converted into an equivalent current control problem. There exist numerous controller candidates for the motor current control task. While it is a subjective matter as to the selection of control structure, the adaptive feedforward inverse dynamics controller showed desirable performance, and therefore is selected for as the favorable controller for the EPAS application.

Chapter 3 ADAPTIVE PARAMETER ESTIMATION

While designing an inverse motor dynamics controller such as that of (2.37), it is assumed that the motor parameters R , L , and K_e are constant and the parameters used in the controller while executing the motor inverse model are the same as that in the motor. However, aging and changing environmental factors, such as temperature, humidity, etc. will change the values of the parameters R , L , and K_e . This can degrade the overall system performance. Often, it is necessary to take some corrective action so that the value of one or more parameters in the controller is as close as possible to the actual value in the motor.

In this chapter, parameter estimation algorithms for a feedforward motor inverse controller are proposed as a solution to the above problem. From single parameter to multiple parameters, the estimation and compensation schemes are derived and stability of the estimation schemes is proved. Improvements for the parameter estimation are proposed by tightening the bound of error in the schemes.

3.1 Motor Inverse Controller Model

A BLDC motor dynamics inverse controller (3.1) had been proposed for the EPAS application (Klienau et al., 2003),

$$V_q = V \cos \delta = Ri_{q,cmd} + \omega_e Li_{d,cmd} + K_{ec} \omega_m, \quad (3.1a)$$

$$V_d = -V \sin \delta = Ri_{d,cmd} - \omega_e Li_{q,cmd}. \quad (3.1b)$$

where δ is the phase advance angle of the control voltage vector to the q axis in the dq coordinate system, and V is the magnitude of control voltage. The phase lead δ can be

set as a function of the angular velocity of electro-magnetic field as $\delta = \tan^{-1}\left(\frac{\omega_e L}{R}\right)$. The

q axis command current $i_{q,cmd}$ is obtained from the outer loop torque command τ_{cmd} as

$$i_{q,cmd} = \frac{\tau_{cmd}}{K_e}. \text{ Note that}$$

$$\sin \delta = \frac{(\omega_e L)}{\sqrt{R^2 + (\omega_e L)^2}}, \quad (3.2a)$$

and

$$\cos \delta = \frac{R}{\sqrt{R^2 + (\omega_e L)^2}}. \quad (3.2b)$$

The control voltage magnitude V can be solved by (3.1a) $\times \cos \delta$ -(3.1b) $\times \sin \delta$ as

$$V = \frac{R^2 i_{q,cmd} + R K_e \omega_m + \omega_e^2 L^2 i_{q,cmd}}{\sqrt{R^2 + (\omega_e L)^2}}. \quad (3.3)$$

The d axis command current $i_{d,cmd}$ can be solved by (3.1a) $\div \cos \delta$ +(3.1b) $\div \sin \delta$ as

$$i_{d,cmd} = \frac{-K_e \omega_m \omega_e L}{R_c^2 + (\omega_e L)^2}. \quad (3.4)$$

Equation (3.3) involves the motor parameters R , L , and K_e . Usually the nominal values of these parameters are used for the control design. Any variation in these parameter values will result in control voltage offset and therefore system performance suffers. Adaptive parameter estimation algorithms will be developed for this controller in this chapter. Let's start with a simpler case of single parameter estimation.

3.2 Single Parameter Estimation

In many situations, the change of the motor coil resistance R is much higher if compared with other parameters such as L , and K_e . Assume that R is the single parameter that changes and affects the motor performance. Integration of the feedback current error scheme to estimate a motor parameter R in the controller is investigated in this section.

3.2.1 Single parameter estimation

In the following paragraphs, the parameters used in the controller are denoted by the suffix c. For the single parameter estimation purpose, we assume that L , and K_e are known. R_c is the phase resistance used in controller and it may be different from its counterpart in the motor. Due to sampling delay and analog/digital conversion, the measured motor velocity ω_{mc} may be slightly different from the actual rotor velocity ω_m .

With this notation, the control law in (3.1) can be restated as

$$V_q = V \cos \delta = R_c i_{q,cmd} + \omega_{ec} L i_{d,cmd} + K_e \omega_{mc}, \quad (3.5a)$$

$$V_d = -V \sin \delta = R_c i_{d,cmd} - \omega_{ec} L i_{q,cmd}. \quad (3.5b)$$

Applying the control voltages (3.5) to the motor dynamics model, we have

$$\frac{di_q}{dt} = \frac{R_c i_{q,cmd} - R i_q}{L} + (\omega_{ec} i_{d,cmd} - \omega_e i_d) + \frac{K_e (\omega_{mc} - \omega_m)}{L}, \quad (3.6a)$$

$$\frac{di_d}{dt} = \frac{R_c i_{d,cmd} - R i_d}{L} + (\omega_e i_q - \omega_{ec} i_{q,cmd}). \quad (3.6b)$$

Define the parameter and velocity error as

$$\Delta R = R - R_c, \quad (3.7a)$$

$$\Delta\omega_m = \omega_m - \omega_{mc}, \quad (3.7b)$$

$$\Delta\omega_e = \omega_e - \omega_{ec}, \quad (3.7c)$$

and the feedback current error as

$$\Delta i_q = i_{q,cmd} - i_q, \quad (3.7d)$$

$$\Delta i_d = i_{d,cmd} - i_d. \quad (3.7e)$$

Assuming the system is in equilibrium, i.e. $\frac{di_q}{dt} = 0$, $\frac{di_d}{dt} = 0$. Substitute (3.7) into

(3.6a), neglect the high order error terms, and reorganize the equation to

$$\Delta R = \frac{\text{sign}(i_{q,cmd})}{|i_{q,cmd}|} (R\Delta i_q + \omega_e L\Delta i_d - Li_{d,cmd}\Delta\omega_e - K_e\Delta\omega_m). \quad (3.8)$$

In normal BLDC motor operation, the current i_q is significant higher than i_d . The velocity error caused by sampling delay is usually lower than the current error caused by the motor dynamics. Thus the first item on the right side of (3.8) is the dominant factor for ΔR calculation. Therefore, the parameter error estimator is proposed as

$$\Delta\hat{R} = \frac{\text{sign}(i_{q,cmd})}{|i_{q,cmd}|} R\Delta i_q. \quad (3.9)$$

where $\Delta\hat{R}$ is the estimated resistance error. The controller value of the resistance can be updated by integrating the estimated error as

$$R_c(t) = R_c(t_0) + \int_{t_0}^t C_1(t)\Delta\hat{R}(t)dt, \quad (3.10)$$

where $0 \leq C_1(t) < 1$ is an integration weighting factor.

3.2.2 Stability of the single parameter estimation scheme

Together with the parameter estimation algorithm in (3.9) and (3.10), the feedforward controller in (3.5) becomes an adaptive controller based on the current feedback. The stability of this closed loop control system will be discussed in this subsection.

Assume the motor coil resistance R is stationary, i.e. $\frac{dR}{dt} \approx 0$. Thus, the derivative of

ΔR can be found by differentiating (3.6a) as

$$\frac{d\Delta R}{dt} = \frac{dR}{dt} - \frac{dR_c}{dt} = -\frac{dR_c}{dt}. \quad (3.11)$$

The right side of equation (3.11) can be expanded by differentiating (3.10). Taking the bounded zero-mean additive noise $d(t)$ of the current feedback sensor into account, the derivative of ΔR can be obtained as

$$\frac{d\Delta R}{dt} = -\text{sign}(i_{q,cmd})Rk(t)\Delta i_q + k(t)\text{sign}(i_{q,cmd})(Rd(t) + \omega_e L\Delta i_d - Li_{d,cmd}\Delta\omega_e - K_e\Delta\omega_m), \quad (3.12)$$

where $k(t) = \frac{C_1(t)}{|i_{q,cmd}|}$. Equation (3.12) represents the first order nonlinear differential

equation that defines the dynamics of the error in the estimate of R .

Differentiating the current error in (3.7d) and (3.7e), and substituting $\frac{di_q}{dt}$ and $\frac{di_d}{dt}$ with

(3.6a) and (3.6b), the current error dynamics can be obtained as

$$\frac{d\Delta i_q}{dt} = -\frac{R}{L}\Delta i_q - \omega_e\Delta i_d + \frac{i_{q,cmd}}{L}\Delta R + \Delta\omega_e i_{d,cmd} - \frac{K_e\Delta\omega_m}{L} + \frac{di_{q,cmd}}{dt}, \quad (3.13)$$

$$\frac{d\Delta i_d}{dt} = -\frac{R}{L}\Delta i_d + \omega_e\Delta i_q + \frac{i_{d,cmd}}{L}\Delta R + \Delta\omega_e i_{q,cmd} + \frac{di_{d,cmd}}{dt}, \quad (3.14)$$

Define a new state vector $\mathbf{x} = [\Delta i_q, \Delta i_d, \Delta R]^T$. The three state equations can be written in terms of state \mathbf{x} as

$$\dot{\mathbf{x}} = \mathbf{f}(t, \mathbf{x}) + \mathbf{g}(t, \mathbf{x}) \quad (3.15)$$

where

$$\mathbf{f}(t, \mathbf{x}) = \begin{bmatrix} -R/L & -\omega_e & i_{q,cmd}/L \\ \omega_e & -R/L & i_{d,cmd}/L \\ -\text{sign}(i_{q,cmd})Rk(t) & \text{sign}(i_{d,cmd})\omega_e Lk(t) & 0 \end{bmatrix} \cdot \mathbf{x}, \quad (3.16)$$

and

$$\mathbf{g}(t, \mathbf{x}) = \begin{bmatrix} \Delta\omega_e i_{d,cmd} - \frac{K_e \Delta\omega_m}{L} + \frac{di_{q,cmd}}{dt} \\ \Delta\omega_e i_{q,cmd} + \frac{di_{d,cmd}}{dt} \\ \text{sign}(i_{q,cmd}) \left[k(t)Rd(t) - Li_{d,cmd}\Delta\omega_e - K_e \Delta\omega_m \right] \end{bmatrix}. \quad (3.17)$$

$\dot{\mathbf{x}} = \mathbf{f}(t, \mathbf{x})$ is the nominal system and $\mathbf{g}(t, \mathbf{x})$ is the perturbation. Equation (3.15) represents the system with current feedback based motor resistance estimation.

In order to determine conditions for the stability of the nominal state-space model, the following theorem is used (Khalil 1996).

Theorem 1 Let $x=0$ be an equilibrium point for the nonlinear system $\dot{\mathbf{x}} = \mathbf{f}(t, \mathbf{x})$, where $\mathbf{f} : [0, \infty) \times D \rightarrow R^n$ is continuously differentiable, $D = \{\mathbf{x} \in R^n \mid \|\mathbf{x}\|_2 < r\}$, and the Jacobian matrix $\left[\frac{\partial \mathbf{f}}{\partial \mathbf{x}} \right]$ is bounded and Lipschitz on D , uniformly in t . Let

$\mathbf{A}(t) = \left. \frac{\partial \mathbf{f}}{\partial \mathbf{x}}(t, \mathbf{x}) \right|_{\mathbf{x}=0}$. Then the origin is an exponentially stable equilibrium point for the

nonlinear system if and only if it is an exponentially stable equilibrium point for the linear system $\dot{\mathbf{x}} = \mathbf{A}(t) \cdot \mathbf{x}$.

In this case, $\mathbf{f}(t, \mathbf{x})$ in (3.16) is continuously differentiable in \mathbf{x} , which satisfies the conditions of the above theorem. $\mathbf{x} = [0, 0, 0]^T$ is an equilibrium point for $\dot{\mathbf{x}} = \mathbf{f}(t, \mathbf{x})$, and

$$\left[\frac{\partial \mathbf{f}}{\partial \mathbf{x}} \right] = \begin{bmatrix} -R/L & -\omega_e & i_{qcom}/L \\ \omega_e & -R/L & i_{dcom}/L \\ -\text{sign}(i_{qcom}) \cdot R \cdot k(t) & \text{sign}(i_{qcom}) \cdot \omega_e \cdot L \cdot k(t) & 0 \end{bmatrix}. \quad (3.18)$$

Since the motor speed (ω_e) and the motor torque (i_{qcom}) are bounded for any motor driving a non-zero load, $\left[\frac{\partial \mathbf{f}}{\partial \mathbf{x}} \right]$ is bounded and Lipschitz on any domain defined over the operating region of the motor. Applying this theorem, it can be made sure that $\dot{\mathbf{x}} = \mathbf{A}(t) \cdot \mathbf{x}$ is exponentially stable as required by Theorem 1 by setting the value of $k(t)$ as required. When the motor drives a load which is the case in most operations, $\mathbf{A}(t)$ is guaranteed to be Hurwitz if $k(t) > 0$ is chosen. On rare occasions it is possible that the motor is back-driven by the load. Under back-driving conditions, $k(t)$ is selected such that the following condition is satisfied.

$$k(t) \left[|i_{q,cmd}| \left[\left(\frac{R}{L} \right)^2 - \omega_e^2 \right] - 2\text{sign}(i_{q,cmd}) \frac{R\omega_e i_{d,cmd}}{L} \right] > 0 \quad (3.19)$$

Therefore, $\mathbf{x} = [0, 0, 0]^T$ is an exponentially stable equilibrium point for the nominal system $\dot{\mathbf{x}} = \mathbf{f}(t, \mathbf{x})$ assuming $k(t)$ is appropriately scheduled. If back driving is indeed a rare occurrence, $k(t)$ can be made close to zero during back-driving to effectively stop parameter estimation. This will not affect the performance of the estimation scheme.

Given the stability of equilibrium point $x = [0, 0, 0]^T$ for the nominal system $\dot{x} = f(t, x)$, the stability of the error dynamics (3.15) can be shown by using theorem 2 (Khalil 1996).

Theorem 2 Let $D = \{x \in R^n \mid \|x\|_2 < r\}$ and suppose the following assumptions are satisfied for all $(t, x) \in [0, \infty) \times D$:

1) $f(t, x)$ is continuously differentiable and the Jacobian matrix $\left[\frac{\partial f}{\partial x} \right]$ is bounded and

Lipschitz in x , uniformly in t .

2) The origin $x = 0$ is an exponentially stable equilibrium point of the nominal system $\dot{x} = f(t, x)$.

3) The perturbation term $g(t, x)$ is piecewise continuous in t and locally Lipschitz in x , and satisfies the bound $\|g(t, x)\| \leq \rho, \forall t \geq t_0 \geq 0, \forall x \in D$.

Let $\hat{y}(t)$ and $y(t)$ denote solutions of the nominal system and the perturbed system, respectively. Then, there exist positive constants $\beta, \gamma, \eta, \mu, \lambda$ and k , independent of ρ , such that if $\rho < \eta$, $\|y(t_0)\| < \lambda$, and $\|y(t_0) - \hat{y}(t_0)\| < \mu$, then the solutions $\hat{y}(t)$ and $y(t)$ will be uniformly bounded for all $t \geq t_0 \geq 0$ and

$$\|y(t) - \hat{y}(t)\| \leq ke^{-\gamma(t-t_0)} \|y(t_0) - \hat{y}(t_0)\| + \beta\rho. \quad (3.20)$$

In this case, $g(t, x)$ is bounded by

$$\|g(t, x)\| \leq \rho = \max_t \|g(t, x)\|. \quad (3.21)$$

Since the desired motor torque, motor speed (assuming non-zero load) and acceleration are bounded, $\Delta\omega_e, \Delta\omega_m, \omega_e, i_{q,cmd}$, and $\frac{di_{q,cmd}}{dt}$ are bounded in real systems. In addition,

the sensor noise $d(t)$ is also bounded. Therefore, $\rho = \max_t \|\mathbf{g}(t, \mathbf{x})\|$ is finite for all motors, and the conditions of Theorem 2 are satisfied. Thus, the solutions of (3.15) and its nominal system will be uniformly bounded and satisfy (3.20), i.e. the adaptive control scheme for R_c will result in a bounded value of estimated R and the state vector \mathbf{x} in (3.12) is uniformly bounded.

3.2.3 Simulation results

The control law (3.5) and the adaptation rule (3.9) and (3.10) were tested in simulation for a motor with following constant parameters: $R = 0.05\Omega$, $K_e = 0.05V/(rad/s)$, $L = 1 \times 10^{-4}H$. This motor is a prototype BLDC motor for an EPAS application. It will be used as a plant for all adaptive controllers developed in this research. Usually the closed loop motor system (including the motor and the motor controller) is the actuator for an outer loop system, and the outer loop system characteristics often have some effects on the performance of the motor controller. However, since our main interest is the motor controller, we shall focus on the motor performance and assume that the motor operates independently to the outer loop system. In practice, this is similar to a motor bench test in which the motor can be operated with arbitrary speed and torque. Therefore, we assume that the motor subjects to random command torque τ_{cmd} and independently random motor velocity ω_m in simulations. Figure 3.1 shows a schematic diagram of the closed loop motor tests in simulation.

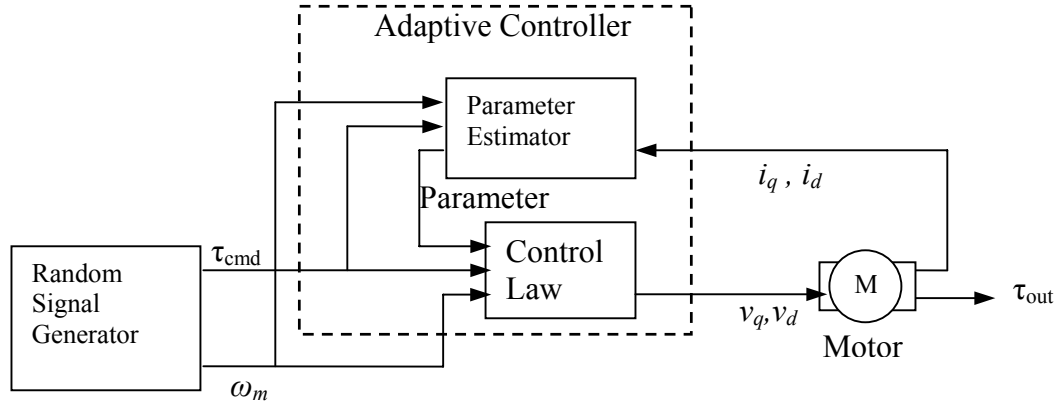


Figure 3.1 The schematic diagram of the motor test system in simulation.

The weighting factor $C_1(t)$ in (3.10) is set to avoid the singular points of $|i_{q,cmd}| = 0$ and to satisfy the stability condition (3.15), as in the following equation

$$C_1(t) = \begin{cases} 0.02 > 0, & \text{sign}(\omega_m) \cdot \text{sign}(\tau_{cmd}) > 0 \\ 0, & \text{sign}(\omega_m) \cdot \text{sign}(\tau_{cmd}) \leq 0 \end{cases} \quad (3.22)$$

To verify the effectiveness of the parameter estimation algorithm, the initial value of R_c is assumed to have 10% error to R . Root mean square (RMS) value of the state variables is used to approximate the bound on the error as

$$b(\Delta) = 6 \cdot RMS(\Delta) \quad (3.23)$$

where Δ represents the error in the variables.

Simulation results of the single parameter estimation are shown in Figure 3.2. Regardless of the sign of initial error, R_c converges to R within 20 seconds and then stays in a bound (max norm) of $1.7228 \times 10^{-4} \Omega (\approx 0.34\%R)$.

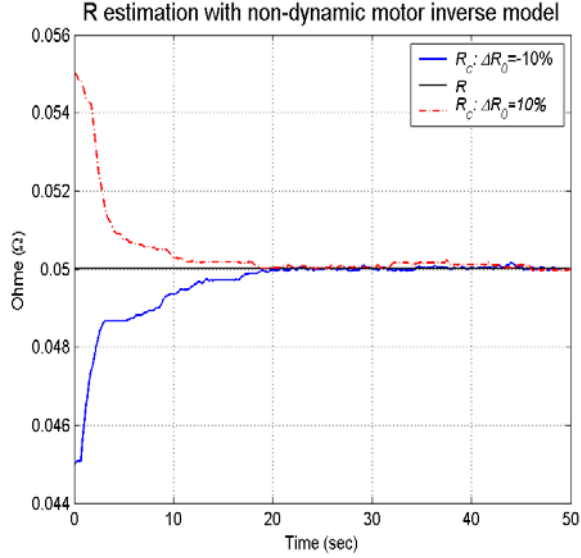


Figure 3.2 Single parameter estimation performance in simulation

3.3 Multiple Parameter Estimation

In many situations, coil resistance R is usually the dominant parameter that is changing in the motor. The single parameter estimation scheme is able to identify this single parameter change in the motor inverse model. However, when there exist errors in more than one parameter, performance of the single-parameter estimation schemes will deteriorate, and the accuracy of the control system will also suffer. In the following subsections, a multi-parameter estimation scheme is proposed and the stability of the scheme is proved.

3.3.1 Multiple parameter estimation

Let's investigate the discrete integration of the estimated parameter error in the controller:

$$R_c(k+1) = R_c(k) + C_1(t)\Delta\hat{R}(k), \quad (3.24)$$

$$K_{ec}(k+1) = K_{ec}(k) + C_1(t)\Delta\hat{K}_e(k), \quad (3.25)$$

where $\Delta\hat{R}(k)$ and $\Delta\hat{K}_e(k)$ are the estimated error, $C_1(k) < 1$ and $C_2(k) < 1$ are integration weighting functions, and k denotes the k th interval. Again, we denote the parameters and variables used for computation in the controller by suffix c . The estimated error $\Delta\hat{R}(k)$ and $\Delta\hat{K}_e(k)$ will be derived below.

Recall the control law

$$V_q = V \cos \delta = R_c i_{q,cmd} + \omega_{ec} L i_{d,cmd} + K_{ec} \omega_{mc}, \quad (3.26a)$$

$$V_d = -V \sin \delta = R_c i_{d,cmd} - \omega_{ec} L i_{q,cmd}. \quad (3.26b)$$

where K_{ec} and R_c may be different from their counterparts in the motor.

Substituting the voltages in (3.26) back into the motor dynamics model, we have

$$\frac{di_q}{dt} = \frac{R_c i_{q,cmd} - R i_q}{L} + (\omega_{ec} i_{d,cmd} - \omega_e i_d) + \frac{K_{ec} \omega_{mc} - K_e \omega_m}{L}, \quad (3.27a)$$

$$\frac{di_d}{dt} = \frac{R_c i_{d,cmd} - R i_d}{L} + (\omega_e i_q - \omega_{ec} i_{q,cmd}). \quad (3.27b)$$

To find the parameter error equation, let's adopt the same definition of errors in (3.7),

$$\Delta R = R - R_c, \quad (3.28a)$$

$$\Delta K_e = K_e - K_{ec}, \quad (3.28b)$$

$$\Delta i_q = i_{q,cmd} - i_q, \quad (3.28c)$$

$$\Delta i_d = i_{d,cmd} - i_d, \quad (3.28d)$$

$$\Delta \omega_e = \omega_e - \omega_{ec}, \quad (3.28e)$$

$$\Delta \omega_m = \omega_m - \omega_{mc}. \quad (3.28f)$$

Assuming the system in equilibrium: $\frac{di_q}{dt} = 0$, $\frac{di_d}{dt} = 0$, substituting (3.28) into

(3.27a), and neglecting the high order error terms such as $\Delta R \Delta i_q$, we get

$$\Delta R i_{q,cmd} + \Delta K_e \omega_m = R \Delta i_q + \omega_e L \Delta i_d - L i_{d,cmd} \Delta \omega_e - K_e \Delta \omega_m. \quad (3.29)$$

It is reasonable to assume that $\Delta R(t)$ and ΔK_e will not change significantly in one sampling period of 0.002 second. In the mean time, for most DC motors, $L \ll R$ and $L \ll K_e$; because of mechanical inertia, $\Delta \omega \ll \Delta i_q$. Therefore, the last 3 terms in the right side of (3.29) can be possibly neglected if compare to the first term, we can find the value of ΔR and ΔK_e by solving following equations

$$\Delta R i_{q,cmd}(k) + \Delta K_e \omega_m(k) = R \Delta i_q(k), \quad (3.30)$$

$$\Delta R i_{q,cmd}(k+1) + \Delta K_e \omega_m(k+1) = R \Delta i_q(k+1). \quad (3.31)$$

Therefore, the estimated error $\Delta \hat{R}(k)$ and $\Delta \hat{K}_e(k)$ for the estimation scheme in (3.24) and (3.25) via feedback current error integration was proposed as

$$\Delta \hat{R}(k) = \frac{R_c \omega_m(k+1) \Delta i_q(k) - R_c \omega_m(k) \Delta i_q(k+1)}{i_{q,cmd}(k) \omega_m(k+1) - i_{q,cmd}(k+1) \omega_m(k)}, \quad (3.32)$$

$$\Delta \hat{K}_e(k) = \frac{R_c i_{q,cmd}(k) \Delta i_q(k+1) - R_c i_{q,cmd}(k+1) \Delta i_q(k)}{i_{q,cmd}(k) \omega_m(k+1) - i_{q,cmd}(k+1) \omega_m(k)}. \quad (3.33)$$

3.3.2 Proof of stability

Subtracting R from both sides of (3.24), the error dynamics in R becomes

$$\Delta R(k+1) = \Delta R(k) - C_1(k) \Delta \hat{R}(k). \quad (3.34)$$

To show the stability of parameter estimation in (3.34), we need to find the difference between the estimated parameter error and the actual parameter error. Recall

(3.29) that approximates the relationship between the actual parameter error and the current error in equilibrium, and include the bounded zero-mean additive noise $d(t)$ of the current feedback sensor and the neglected motor electrical dynamics $\frac{di_q}{dt}$,

$$\Delta R i_{q,cmd} + \Delta K_e \omega_m = R \Delta i_q - K_e \Delta \omega_m + \omega_e L \Delta i_d - L i_{d,cmd} \Delta \omega_e + d(t) - \frac{di_q}{dt}. \quad (3.35)$$

Define a perturbation $g(t)$ as

$$g(t) = -K_e \Delta \omega_m + \omega_e L \Delta i_d - L i_{d,cmd} \Delta \omega_e + d(t) - \frac{di_q}{dt}. \quad (3.36)$$

Solving equation (3.35) at k th interval and $(k+1)$ th interval, the actual parameter error ΔR and ΔK_e can be solved as

$$\Delta R(k) = \frac{[\omega_m(k+1)R\Delta i_q(k) - \omega_m(k)R\Delta i_q(k+1)]}{i_{q,cmd}(k)\omega_m(k+1) - i_{q,cmd}(k+1)\omega_m(k)} + \frac{[\omega_m(k+1)g(k) - \omega_m(k)g(k+1)]}{i_{q,cmd}(k)\omega_m(k+1) - i_{q,cmd}(k+1)\omega_m(k)} \quad (3.37)$$

Note that the first item on the right side of (3.37) is the same as $\Delta \hat{R}(k)$. In another word, ΔR becomes

$$\Delta R(k) = \Delta \hat{R}(k) + \frac{[\omega_m(k+1)g(k) - \omega_m(k)g(k+1)]}{i_{q,cmd}(k)\omega_m(k+1) - i_{q,cmd}(k+1)\omega_m(k)}. \quad (3.38)$$

Let

$$e(k) = \frac{[\omega_m(k+1)g(k) - \omega_m(k)g(k+1)]}{i_{q,cmd}(k)\omega_m(k+1) - i_{q,cmd}(k+1)\omega_m(k)}. \quad (3.39)$$

Substitute the (3.38) and (3.39) into (3.34),

$$\Delta R(k+1) = [1 - C_1(k)]\Delta R(k) + C_1(k)e(k). \quad (3.40)$$

By induction we have

$$\Delta R(k+1) = \left\{ \prod_{i=1}^k [1 - C_1(i)] \right\} \Delta R(0) + E(k+1), \quad (3.41)$$

$$\text{where, } E(k+1) = \sum_{j=0}^k \left\{ \left[\prod_{i=0}^{k-j} (1 - C_1(k-i+1)) \right] C_1(j) e(j) \right\}.$$

Setting $0 \leq C_1(k) < 1$ for each controller computation interval $[kT, (k+1)T]$, the first item on the right of (3.35) will decrease as $k \rightarrow \infty$. In some intervals $e(k)$ in (3.33) is high because of singularity caused by very low angular velocity or command torque input. The integration gain $C_1(k)$ can be set to be zero for these intervals. Otherwise, in a physical system, $\Delta\omega_m$, ω_m , i_{dcom} , $d(t)$, and $\frac{di_q}{dt}$ are all bounded, i.e. $|e(k)| < \delta(k)$ for some positive constant $\delta(k)$. Let

$$C_{\max} = \max_k [1 - C_1(k), C_1(k)] \quad (3.42)$$

and $\delta_{\max} = \max_k [\delta(k)]$, where $k \in \{k | C_1(k) \neq 0, k = 0, 1, \dots, \infty\}$. The second term $E(k+1)$ on the right sides of (3.41) conforms the inequality of

$$E(k+1) \leq \sum_{j=0}^k \left\{ (C_{\max})^j \delta_{\max} \right\} = \frac{1 - C_{\max}^{k+1}}{1 - C_{\max}} \delta_{\max}. \quad (3.43)$$

Let $b(k+1) = \frac{1 - C_{\max}^{k+1}}{1 - C_{\max}} \delta_{\max}$. Since $0 < C_{\max} < 1$, $b(k) \rightarrow \frac{1}{1 - C_{\max}} \delta_{\max}$ as $k \rightarrow \infty$.

Therefore, the second item on the right side of (3.35) will be bounded as $k \rightarrow \infty$. In summary, the estimation scheme of R in (3.24) and (3.32) will be bounded. The boundedness of K_e can be proved similarly.

The multi-parameter estimation algorithm in (3.24) and (3.25) is simulated for the same motor mentioned in section 3.2. Figure 3.3 shows the parameter estimation performance

in simulations. The initial parameter error were set as $\Delta R(t=0) = \pm 10\%R$ and $\Delta K_e(t=0) = \pm 6\%K_e$. To avoid the error caused by singularity points in (3.32) and (3.33), $C_1(k)$ and $C_2(k)$ in (3.24) and (3.25) are set as following:

$$C_1(k) = C_2(k) = \begin{cases} 0.1 & \det(k) \geq 0.01 \\ 0 & \det(k) < 0.01 \end{cases}, \quad (3.44)$$

where $\det(k) = i_{q,cmd}(k)\omega_m(k+1) - i_{q,cmd}(k+1)\omega_m(k)$. In the simulation, the parameter estimation reached the bound of $b(\Delta R) = 7.1656 \times 10^{-4} (\approx 1.43\%R)$ and $b(\Delta K_e) = 2.5125 \times 10^{-4} (\approx 0.5\%K_e)$ within 80 seconds.

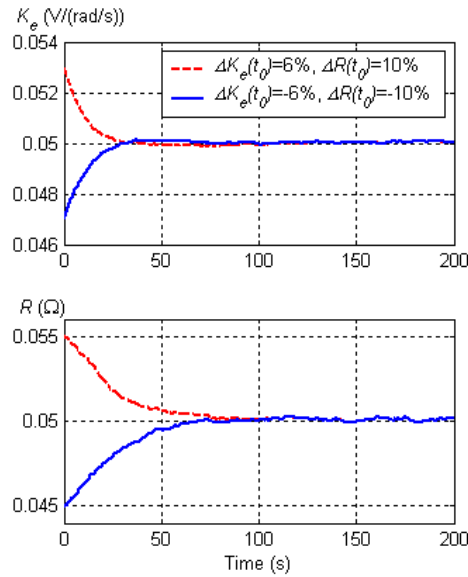


Figure 3.3 Two-parameter estimation performance in simulation

3.4 Improving The Dynamic Performance Of The Adaptive Algorithm

The single and multiple parameter estimation schemes developed in the previous two sections are proved to be stable and validated to be effective in simulations. However,

faster parameter estimation convergence is always welcome for the overall system performance. The convergence speed can be improved by tuning the integration gains $C_1(k)$ and $C_2(k)$ but will result in noisier parameter estimation at steady state. The extent of improvement by gain tuning is limited by the tradeoff between converging speed and the bound of steady state error. A better way, without compromising the dynamic performance would be to reduce the perturbation to the error dynamics model.

Recall that both the single and multiple parameter estimation schemes are based on the motor inverse controller (3.1). The non-dynamic motor inverse model neglected motor electrical dynamics and introduced an error corresponding to neglected dynamics $\frac{di_q}{dt}$ and $\frac{di_d}{dt}$. If the dynamics are not neglected to obtain an algorithm for the applied voltage V , it is possible to reduce errors introduced in the earlier estimation approach. Furthermore, the basic estimation scheme of (3.9), (3.32) and (3.33) neglected the effect of $\Delta\omega_m$ and $\Delta\omega_e$, which were caused by the sampling delay. These terms presented in the perturbation (3.17) and (3.36). If the motor speed error is considered in the estimation scheme, the performance of the adaptive algorithm can be possibly improved. In the next subsections, these two possible improvements for parameter estimation will be explored.

3.4.1 *Motor electrical dynamics*

An analytical solution of the motor electrical dynamics of (2.31) is not available because of the nonlinear items that involve product of states ω_e , i_q and i_d . However, since the motor is driving the inertia, the electrical states can change significantly in duration of the order of the electrical time constant τ , whereas the mechanical states, e.g.

the motor speed ω_m , can hardly change over the same period due to slow dynamics of the mechanical system. It is possible to approximate the motor electrical dynamics by assuming ω_m is a constant during each sampling interval T , which is of the order of the electrical time constant τ . With this assumption, the motor electrical dynamics equation (2.31) becomes a finite dimensional linear time-invariant state equation during the sampling interval T . This equation can be solved exactly via the discrete time state transition matrix (Rugh 1996).

Now consider (2.31) in matrix form as

$$\dot{y} = \frac{1}{\tau} \begin{pmatrix} -1 & -\zeta \\ \zeta & -1 \end{pmatrix} \cdot y + \frac{1}{\tau} \begin{pmatrix} \frac{\cos \delta}{R} & \frac{-K_e}{R} \\ \frac{-\sin \delta}{R} & 0 \end{pmatrix} \cdot u \quad , \quad (3.45)$$

for $kT \leq t \leq (k+1)T$, where $y = \begin{pmatrix} i_q \\ i_d \end{pmatrix}$, $u = \begin{pmatrix} V \\ \omega_m \end{pmatrix}$, $\tau = \frac{L}{R}$, $\zeta = \tan \delta = \frac{\omega_e L}{R}$.

Equations (3.45) can be solved with the state transition matrix:

$$\Phi(kT, t) = e^{-\frac{t-kT}{\tau}} \begin{pmatrix} \cos[\omega_e(t-kT)] & -\sin[\omega_e(t-kT)] \\ \sin[\omega_e(t-kT)] & \cos[\omega_e(t-kT)] \end{pmatrix} \quad (3.46)$$

for $kT \leq t \leq (k+1)T$. The solution is

$$x[(k+1)T] = \Phi[kT, (k+1)T] \cdot x(kT) + \int_{kT}^{(k+1)T} \Phi(kT, \theta) G u d\theta, \quad (3.47)$$

where $G = \begin{pmatrix} \frac{\cos \delta}{R} & \frac{-K_e}{R} \\ \frac{-\sin \delta}{R} & 0 \end{pmatrix}$.

The quantities G and u in the above equation can be treated as constants during each sampling interval because of the zero-order-hold sampling.

Given the command torque $\tau_{cmd}(k)$, which is the expected output at $t = (k+1)T$, the value of the state variable $i_q(k+1)$ is expected to be $i_{q,cmd}(k)$. This value can be obtained from $i_{q,cmd}(k) = \frac{\tau_{cmd}(k)}{K_e}$, and the unknown $V(k)$ and $i_d(k+1)$ can be solved as

$$V(k) = \frac{1}{I \frac{\cos \delta}{R\tau} + J \frac{\sin \delta}{R\tau}} \left\{ i_{q,cmd}(k) + I \frac{K_e}{R\tau} \omega_m - e^{-\frac{T}{\tau}} [i_q(k) \cos(\omega_e T) - i_d(k) \sin(\omega_e T)] \right\}, \quad (3.48a)$$

$$i_d(k+1) = e^{-\frac{T}{\tau}} [i_q(k) \sin(\omega_e T) + i_d(k) \cos(\omega_e T)] + (J \frac{\cos \delta}{R\tau} - I \frac{\sin \delta}{R\tau}) V(k) - J \frac{K_e}{R\tau} \omega_m, \quad (3.48b)$$

where

$$I = \frac{(\omega_e \tau)^2}{(\omega_e \tau)^2 + 1} \left[\frac{1}{\omega_e} e^{-\frac{T}{\tau}} \sin(\omega_e T) + \frac{1}{\omega_e^2 \tau} (1 - e^{-\frac{T}{\tau}} \cos(\omega_e T)) \right], \quad (3.49a)$$

$$J = \frac{(\omega_e \tau)^2}{(\omega_e \tau)^2 + 1} \left[\frac{1}{\omega_e} (1 - e^{-\frac{T}{\tau}} \cos(\omega_e T)) - \frac{1}{\omega_e^2 \tau} e^{-\frac{T}{\tau}} \sin(\omega_e T) \right]. \quad (3.49b)$$

Compared to the non-dynamic motor inverse controller in (3.1), the state transition matrix method approximates the electrical dynamics of the motor. (Hereafter it is called “dynamic motor inverse controller”). Therefore, when the motor speed is constant or changed at a lower frequency compared to the torque, the algorithm with the state transition matrix method can track the required torque with higher fidelity if the parameters in the controller match the actual values in the motor. If there is discrepancy between the parameter value in motor and in the controller, the output current error will reflect this discrepancy. Consequently, the performance of the estimation scheme and the

controller will be improved.

Figure 3.4 compares the performance of single parameter estimation adaptive controller with non-dynamic motor inverse controller in (3.1) and with approximate dynamic motor inverse controller (3.48). The parameter estimation algorithms in these two cases are the same as in (3.9) and (3.10), and the integration gains in both cases are set as

$$C_1(t) = \begin{cases} 0.1, & \text{sign}(\omega_m) \cdot \text{sign}(\tau_{cmd}) > 0 \\ 0, & \text{sign}(\omega_m) \cdot \text{sign}(\tau_{cmd}) \leq 0 \end{cases} \quad (3.50)$$

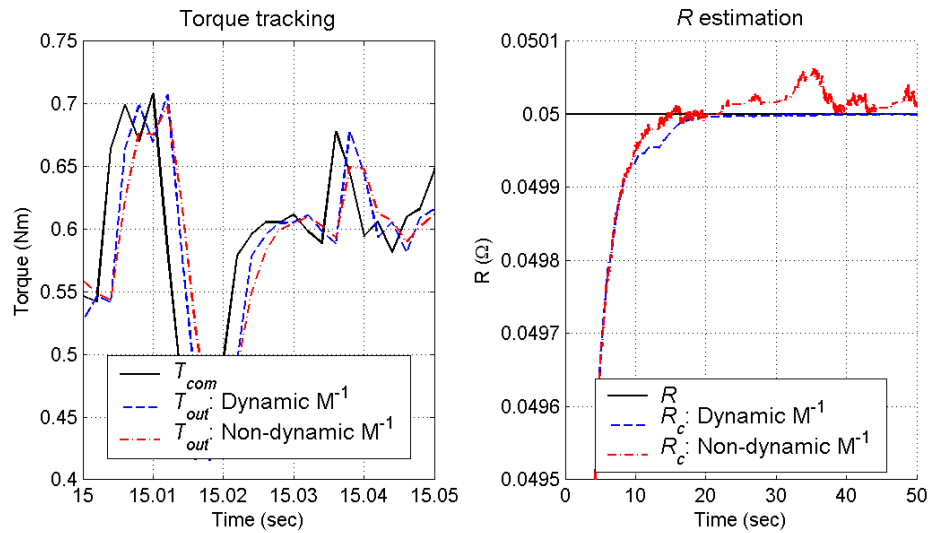


Figure 3.4 Single parameter estimation transient performances: non-dynamic motor inverse controller vs. approximated dynamic inverse controller.

Apparently, the dynamic motor inverse controller improved the accuracy of parameter estimation and current tracking, though the parameter converging speed is generally the same. This also means that, the dynamic motor inverse controller allows faster parameter estimation via higher integration gain given the same requirement on steady state parameter estimation error.

Figure 3.5 compares the non-dynamic motor inverse controller and dynamic motor inverse controller in the case of the multi-parameter estimation. The parameter estimation gains are set as

$$C_1(k) = C_2(k) = \begin{cases} 0.1 & \det(k) \geq 0.01 \\ 0 & \det(k) < 0.01 \end{cases} \quad (3.51)$$

for both controller.

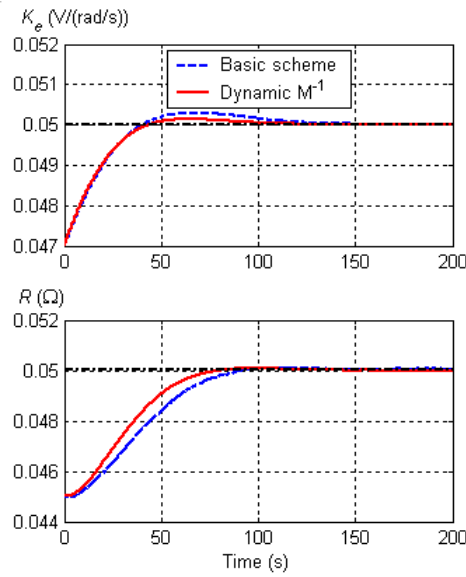


Figure 3.5 Multi-parameter estimation using non-dynamic motor inverse controller vs. dynamic motor inverse controller

With the dynamic motor inverse controller provides, the steady state parameter error bounds (after 150s) are $b(\Delta R) = 4.492 \times 10^{-5}$, $b(\Delta K_e) = 3.365 \times 10^{-6}$, which are about 17.5% and 29.4% of corresponding values in the simulation with the non-dynamic motor inverse controller. Besides, the dynamic motor inverse controller results in lower overshoot of K_{ec} (47% of that of the non-dynamic inverse controller), and the rise time of R_c about 10 seconds shorter.

3.4.2 Motor speed sampling delay

While calculating the voltage using (3.3) or (3.48a), the desired q -axis current during the next sampling period is known but the motor velocity during the next sampling period is not. The error $\Delta\omega_m$ is caused by the use of this “last available” sampled motor velocity. The parameter estimation scheme of (3.9), (3.32) and (3.33) neglected this error caused by the sampling delay. It is possible to reduce the bound of error during estimation if $\Delta\omega_m$ is taken into account.

The motor velocity is not constant during the sampling period $[kT, (k+1)T]$, but it continuously varies from $\omega_m(k)$ to $\omega_m(k+1)$. It is reasonable to approximate the motor velocity error by $\Delta\hat{\omega}_m(k) = \frac{1}{2}[\omega_m(k+1) - \omega_m(k)]$ during this sampling period. In the mean time, the control voltage is calculated based on variable values at the k th sample. If these values are kept in memory for the sampling period $[kT, (k+1)T]$, the approximate velocity error can be used together for parameter error estimation. In short, the parameter estimation algorithm may take advantage of $i_{q,cmd}$, i_q and ω_m that are saved in memory for the past a few samples. The feedforward controller, on the other hand, always uses current values of these variables for control voltage calculation. The physical parameters of the motor are stationary, so using the variable values in the past several samples will not introduce error on parameter estimation.

For the single parameter estimation, an improvement is then proposed as

$$\frac{dR_c}{dt} = k(t)\Delta\hat{R} = \text{sign}(i_{q,cmd})k(t)(R\Delta i_q - Li_{d,cmd}\Delta\hat{\omega}_e - K_e\Delta\hat{\omega}_m). \quad (3.52)$$

With (3.52), $f(t, x)$ and $g(t, x)$ in (3.15) become

$$f(t, x) = \begin{bmatrix} -\frac{R}{L} & -\omega_e & \frac{i_{qcom}}{L} \\ \omega_e & -\frac{R}{L} & \frac{i_{dcom}}{L} \\ -\text{sign}(i_{qcom})Rk(t) & \text{sign}(i_{qcom})\omega_e Lk(t) & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ 0 \\ \text{sign}(i_{qcom})k(t)(Li_{dcom}\Delta\hat{\omega}_e + K_e\Delta\hat{\omega}_m) \end{bmatrix}, \quad (3.53)$$

and,

$$g(t, x) = \begin{bmatrix} \Delta\omega_e i_{d,cmd} - \frac{K_e\Delta\omega_m}{L} + \frac{di_{q,cmd}}{dt} \\ \Delta\omega_e i_{q,cmd} + \frac{di_{d,cmd}}{dt} \\ \text{sign}(i_{q,cmd})k(t)Rd(t) \end{bmatrix}. \quad (3.54)$$

The disturbance function $g(t, x)$ in (3.54) has fewer terms than the $g(t, x)$ in (3.17). Mathematically its bound defined in (3.19) will be lower than or equal to the bound the $g(t, x)$ in (3.17). Physically, it is most likely lower.

When using the approximate motor speed error in the multiple parameter estimation, the parameter error equations of (3.30~31) becomes

$$\Delta R i_{q,cmd}(k) + \Delta K_e \omega_m(k) = R \Delta i_q(k) - K_e \Delta \hat{\omega}_m(k), \quad (3.55)$$

$$\Delta R i_{q,cmd}(k+1) + \Delta K_e \omega_m(k+1) = R \Delta i_q(k+1) - K_e \Delta \hat{\omega}_m(k+1). \quad (3.56)$$

Thus the parameter error estimation scheme is changed to

$$\Delta \hat{R} = \frac{[\omega_m(k+1)\Delta S(k) - \omega_m(k)\Delta S(k+1)]}{i_{q,cmd}(k)\omega_m(k+1) - i_{q,cmd}(k+1)\omega_m(k)}, \quad (3.57)$$

$$\Delta \hat{K}_e = \frac{[-i_{q,cmd}(k+1)\Delta S(k) + i_{q,cmd}(k)\Delta S(k+1)]}{i_{q,cmd}(k)\omega_m(k+1) - i_{q,cmd}(k+1)\omega_m(k)}, \quad (3.58)$$

where $\Delta S(k) = R\Delta i_q(k) - K_e\Delta\omega_m(k)$. The perturbation $g(t)$ becomes

$$g(t) = \omega_e L\Delta i_d - Li_{d,cmd}\Delta\omega_e + d(t) - di_q/dt. \quad (3.59)$$

The bound of the perturbation is therefore lower than that of (3.36).

Since the two ways of performance improvement discussed before are applied to the motor inverse model and parameter estimation algorithm respectively, it is possible to combine them together and tighten the bound of error, thereafter further improve the precision of the system. These improvements are verified in simulations. Figure 3.6 compares the estimation algorithm with improvements and the baseline estimation algorithm for single parameter estimation. The algorithm with both the approximate motor dynamics and the motor velocity error achieved the lowest bound of $b(\Delta R) = 2.83 \times 10^{-5}$ and $b(\Delta i_q) = 0.96$.

Figure 3.7 shows the performance comparison of the estimation scheme with and without the approximated $\Delta\omega_m$ compensation for multi-parameter estimation. The scheme with $\Delta\omega_m$ compensation (case 3) has lower steady state error bounds (after 150s): $b(\Delta R) = 1.638 \times 10^{-4}$, $b(\Delta K_e) = 6.39 \times 10^{-5}$, which are about 19.6% and 42.8% of corresponding values in the basic estimation model (case 1).

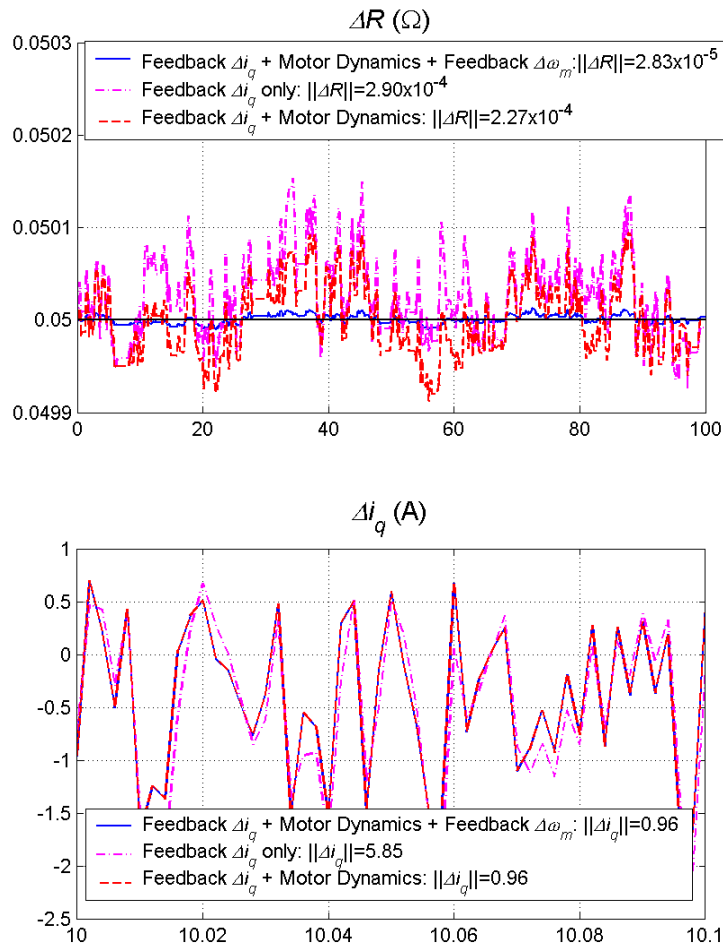


Figure 3.6 Single parameter estimation performances: comparison of the basic estimation program and the performance improvements.

Table 3.1 compares the bounds of estimated parameter error among the four different combinations of motor inverse model and parameter estimation scheme. Figure 3.8 shows the performance comparison of the parameter estimation with the four different algorithms. Apparently, the adaptive control with the dynamic motor inverse model and the $\Delta \omega_m$ compensator achieves highest precision among the 4 cases.

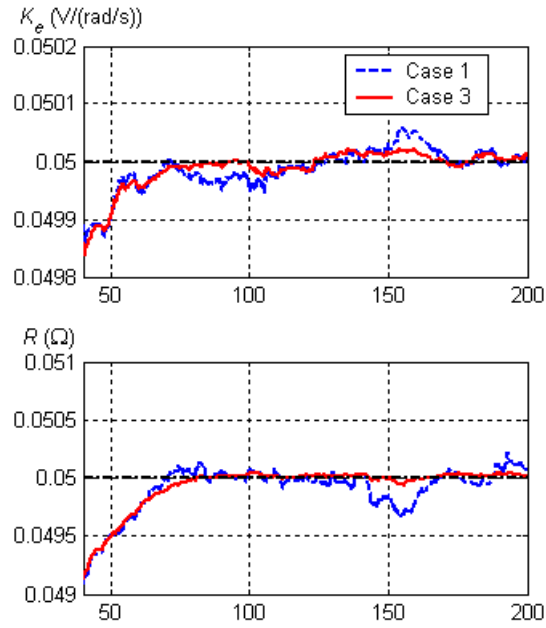


Figure 3.7 Performance comparison of estimation scheme in open loop simulation: case 1 – the basic scheme; case 3– with approximated $\Delta\omega_m$ compensation.

Table 3.1 Bound of errors in open loop simulation: two-parameter estimation

Case	Motor Inverse Model	Estimation scheme	Bound of ΔR $b(\Delta R)$ ($\times 10^{-4}$)	Percentage of $b(\Delta R)$ compared to Case #1	Bound of ΔK_e $b(\Delta K_e)$ ($\times 10^{-4}$)	Percentage of $b(\Delta K_e)$ compared to Case #1
1	Non-Dynamic	Basic scheme	11.39	100%	2.243	100%
2	Dynamic	Basic scheme	5.796	50.89%	1.573	70.16%
3	Non-dynamic	$\Delta\omega_m$ compensator	10.73	94.23%	2.187	97.54%
4	Dynamic	$\Delta\omega_m$ compensator	1.072	9.41%	0.255	11.36%

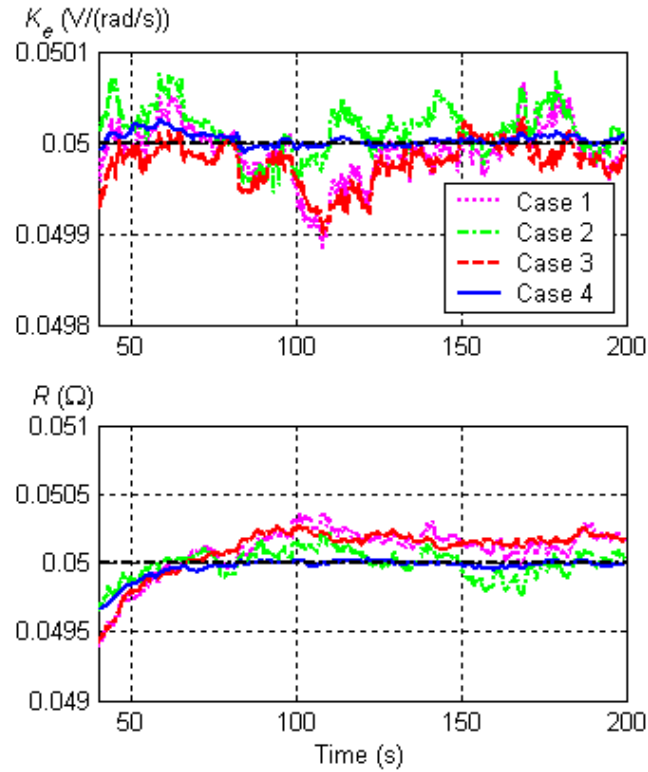


Figure 3.8 Performance comparison of 4 different estimation schemes in open loop simulation: two-parameter estimation

3.5 Multiple Parameter Estimation Using Gram-Schmidt Orthonormalization

In the previous section, single parameter and multi-parameter estimation algorithms were developed for the control law of (3.1). These algorithms were proved to be stable and were validated in simulations. However, it would be more desirable if the parameter estimation algorithms can be faster while maintaining the bound of the steady state parameter error, especially in the multi-parameter case.

Let's re-examine the multi-parameter estimation algorithms (3.32~33, 3.57~58). In each of the parameter error estimators, when the value of the denominator was close to zero at any step, the integration gains in (3.24) and (3.25) were set to zero. By doing this,

the computation noise introduced by singularity can be avoided. On the other hand, useful information hidden in the calculation was discarded. Therefore, if the estimation algorithm can take advantage of the information in these near singularity calculations without adding noise, its performance of parameter estimation will be improved.

Equations (3.32) and (3.33) were obtained by solving (3.29) with data from two sampling intervals. As the actual motor parameters are stationary, we can add more data into the computation. The equation (3.29) can be in a vector format with the parameter error as unknown constants. The parameter errors can then be solved through the Gram-Schmidt orthonormalization process. The singularity problem in (3.32) and (3.33) will be avoided as redundant data are used for solving the parameter error. In addition, today's digital controllers are capable of saving a few steps of data in its memory, so the estimated values can be calculated and updated every several samples. Even though this is not literally real-time, it is fast enough for estimating the physical motor parameter values. This idea will be explored in this section.

3.5.1 Multi-parameter estimation using Gram-Schmidt orthonormalization

Define an inner product $\langle x_1(t), x_2(t) \rangle$ on the function space $f(I) = \{ \text{all continuous functions on the interval } I = [t_0, t_0 + T_{ip}] \}$

$$\langle x_1(t), x_2(t) \rangle = \int_{t_0}^{t_0 + T_{ip}} x_1(t)x_2(t)dt, \quad (3.60)$$

where t_0 may vary with time, T_{ip} is a constant, $x_1(t)$ and $x_2(t)$ are continuous functions in $f(I)$. $i_{q,cmd}(t)$ and $\omega_m(t)$ can be orthonormalized with this inner product. Considering that ΔR and ΔK_e are both slowly developing errors, it is reasonable to assume that ΔR

and ΔK_e are constants within the interval if we choose a short interval T_{ip} for the inner product in (3.60). The estimated value of ΔR and ΔK_e can be computed from (3.29). As

an example, we will find $\Delta \hat{R}$ through the following process.

The projection of $i_{qcom}(t)$ onto $\omega_m(t)$, denoted $i_\omega(t)$ is

$$i_\omega(t) = \frac{\langle i_{qcom}, \omega_m \rangle}{\langle \omega_m, \omega_m \rangle} \omega_m(t). \quad (3.61)$$

Subtracting $i_\omega(t)$ from $i_{q,cmd}(t)$, we get a vector orthogonal to $\omega_m(t)$ in space $f(I)$, as shown in Figure 3.9.

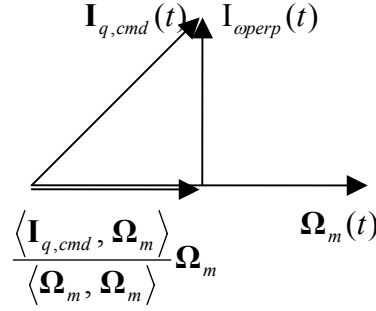


Figure 3.9 Schematic diagram of Gram Schmidt Orthonormalization

$$i_{operp}(t) = i_{q,cmd}(t) - \frac{\langle i_{q,cmd}, \omega_m \rangle}{\langle \omega_m, \omega_m \rangle} \omega_m(t). \quad (3.62)$$

Taking the inner product of $i_{operp}(t)$ to both sides of (3.29) and neglecting the relatively small magnitude of i_{dcom} and Δi_d ,

$$\langle i_{operp}, i_{q,cmd} \Delta R + \omega_m \Delta K_e \rangle = \langle i_{operp}, R_c \Delta i_q - K_{ec} \Delta \omega_m \rangle \quad (3.63)$$

Note that $\langle i_{operp}, \omega_m \rangle = 0$. The estimated value of ΔR is proposed as

$$\Delta \hat{R} = \frac{R_c \langle i_{operp}, \Delta i_q \rangle - K_{ec} \langle i_{operp}, \Delta \omega_m \rangle}{\langle i_{operp}, i_{qcom} \rangle}. \quad (3.64)$$

Similarly, $\Delta \hat{K}_e$ can be proposed as

$$\Delta \hat{K}_e = \frac{R_c \langle \omega_{iperp}, \Delta i_q \rangle - K_{ec} \langle \omega_{iperp}, \Delta \omega_m \rangle}{\langle \omega_{iperp}, \omega_m \rangle}. \quad (3.65)$$

where, $\omega_{iperp} = \omega_m(t) - \frac{\langle i_{q,cmd}, \omega_m \rangle}{\langle i_{q,cmd}, i_{q,cmd} \rangle} i_{q,cmd}(t)$.

The above estimated error $\Delta \hat{R}$ and $\Delta \hat{K}_e$ can then be used to compensate the parameters in the controller as

$$R_c(k+1) = R_c(k) + C_1(k) \Delta \hat{R}(k), \quad (3.66a)$$

$$K_{ec}(k+1) = K_{ec}(k) + C_2(k) \Delta \hat{K}_e(k), \quad (3.66b)$$

where, $R_c(k)$ and $K_{ec}(k)$ are the values of R and K_e in the controller for $kT_{ip} \leq t < (k+1)T_{ip}$, $C_1(k)$ and $C_2(k)$ are the weighting factors and satisfy $0 \leq C_1(k) < 1, 0 \leq C_2(k) < 1$.

3.5.2 Proof of stability

Subtracting R from both sides of (3.55a), the error of ΔR at $(k+1)$ th interval can be found as

$$\Delta R(k+1) = \Delta R(k) - C_1(k) \Delta \hat{R}(k). \quad (3.67)$$

Including the bounded zero-mean additive noise $d(t)$ of the current feedback sensor and

the neglected motor electrical dynamics $\frac{di_q}{dt}$, equation (3.29) becomes

$$\Delta R i_{q,cmd} + \Delta K_e \omega_m = R \Delta i_q - K_e \Delta \omega_m + \omega_e L \Delta i_d - L i_{d,cmd} \Delta \omega_e + d(t) - \frac{di_q}{dt}. \quad (3.68)$$

Define a perturbation $g(t)$ by

$$g(t) = \omega_e L \Delta i_d - L i_{d,cmd} \Delta \omega_e + d(t) - \frac{di_q}{dt}. \quad (3.69)$$

Applying the Gram Schmidt orthonormalization procedure to (3.68),

$$\Delta R(k) = \frac{1}{\langle i_{operp}, i_{q,cmd} \rangle_k} \left\{ R \langle i_{operp}, \Delta i_q \rangle_k - K_e \langle i_{operp}, \Delta \omega_m \rangle_k + \langle i_{operp}, \omega_e L \Delta i_d - L i_{d,cmd} \Delta \omega_e + d(t) - \frac{di_q}{dt} \rangle_k \right\}. \quad (3.70)$$

Substituting (3.64) and (3.69) into (3.70),

$$\Delta R(k) = \Delta \hat{R}(k) + \frac{\langle i_{operp}, g(t) \rangle_k}{\langle i_{operp}, i_{q,cmd} \rangle_k}. \quad (3.71)$$

Substitute (3.71) into (3.67), we get the parameter error dynamics

$$\Delta R(k+1) = [1 - C_1(k)] \Delta R(k) + C_1(k) \frac{\langle i_{operp}, g(t) \rangle_k}{\langle i_{operp}, i_{q,cmd} \rangle_k}. \quad (3.72)$$

By induction we have

$$\Delta R(k+1) = \left\{ \prod_{i=1}^k [1 - C_1(i)] \right\} \Delta R(0) + D(k+1), \quad (3.73)$$

$$\text{where, } D(k+1) = \sum_{j=1}^k \left\{ \left[\prod_{i=1}^{k-j} C_1(i) \right] \frac{\langle i_{operp}, g(t) \rangle_j}{\langle i_{operp}, i_{q,cmd} \rangle_j} \right\}.$$

Setting $0 \leq C_1(k) < 1$ for each interval $[kT_{ip}, (k+1)T_{ip})$, the first item on the right of (3.62) will be decreasing as $k \rightarrow \infty$. Since $\Delta \omega_m$, ω_e , $i_{d,cmd}$, $d(t)$, and $\frac{di_q}{dt}$ are all

bounded, i.e. $\left| \frac{\langle i_{operp}, g(t) \rangle_k}{\langle i_{operp}, i_{q,cmd} \rangle_k} \right| < \delta(k)$ for some positive constant $\delta(k)$.

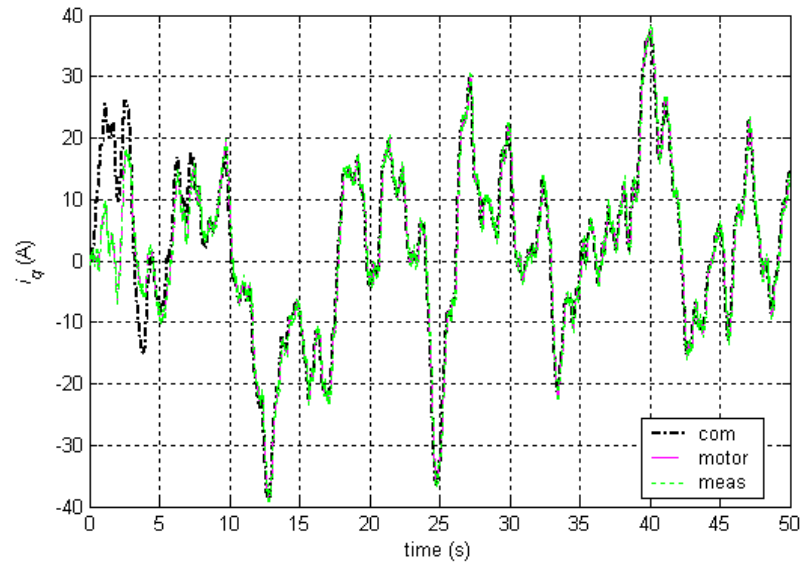
Let $C_{\max} = \max_{k \in \{1, \dots, \infty\}} [C(k)]$, $\delta_{\max} = \max_{k \in \{1, \dots, \infty\}} [\delta(k)]$,

$$D(k+1) \leq \sum_{j=1}^k \left\{ (C_{\max})^j \delta_{\max} \right\} = \frac{1 - C_{\max}^{k+1}}{1 - C_{\max}} \delta_{\max} = b(k+1). \quad (3.74)$$

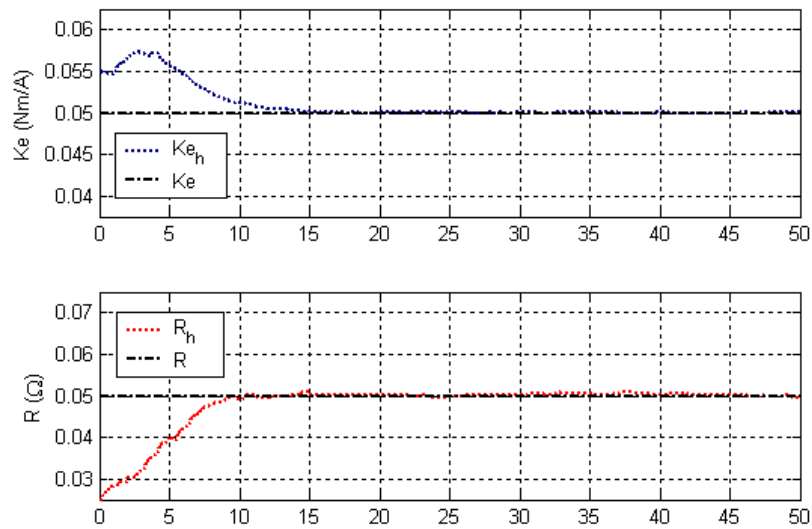
Since $0 \leq C_{\max} < 1$, $b(k) \rightarrow \frac{1}{1 - C_{\max}} \delta_{\max}$ as $k \rightarrow \infty$. Therefore, the second item on the

right side of (3.73) will be bounded as $k \rightarrow \infty$. In summary, the estimation scheme of (3.66) is bounded.

Simulation results of the Gram-Schmidt estimation algorithm are shown in Figure 3.10. In the simulation, the orthonormalization interval was set as 250ms, and the controller-sampling step was set as 2ms. The integration gains were set as $C_1(k) = C_2(k) = 0.1$. Thanks to the orthonormalization process, the algorithms of (3.64~65) had better convergent speed (less than 15 seconds) and higher accuracy than the multi-parameter estimation algorithms developed in section 3.5. Note that the dynamics approximation and speed delay estimator proposed in section 3.4 were implemented in this algorithm for better results.



(a) Current tracking results



(b) Parameter estimation results

Figure 3.10 Parameter estimation performance with Gram-Schmidt orthonormalization: the initial parameter error $+50\%R$ and $-15\%K_e$

3.6 Multi-Parameter Estimation Using q and d Axis Current

In section 3.5, it was shown that the parameter estimation performance could be improved by including more data in the estimator through the Gram-Schmidt normalization process. The performance improvement was achieved at higher computation costs of the inner product (3.60). Examining the parameter estimation algorithms proposed in Sections 3.1 to 3.5, they are all based on the q -axis dynamics of the BLDC motor. It is true that d -axis current, i_d , has little effect on the torque generation, but there may be some information about the parameter difference hidden in the d -axis current dynamics. In addition, the d -axis current, i_d , always accompanies the q -axis current i_q , because they are obtained from the phase currents through the coordinate transformations discussed in Chapter 2. It is possible to take advantage of the i_d dynamics for parameter estimation.

On the other hand, though the boundedness of the proposed single and multiple parameter estimation algorithms are proved, it is still interesting to see the stochastic characteristics of the algorithms. In this section, we will discuss the estimation performance at the presence of zero mean Gaussian noise.

3.6.1 Parameter estimation

Let us re-examine the motor dynamics model

$$\begin{aligned}L \frac{di_q}{dt} &= -Ri_q - \omega_e L i_d - K_e \omega_m + v_q, \\L \frac{di_d}{dt} &= -Ri_d + \omega_e L i_q + v_d, \\ \tau_{out} &= K_e i_q.\end{aligned} \tag{3.75}$$

Notice that the back EMF constant K_e was not explicitly involved in the d -axis current

dynamics equation. It is possible to use the d -axis current dynamics for estimation of the phase resistance R . The back EMF constant K_e can be estimated using the q -axis current feedback and the estimated R . In that case, it is desirable to have a d -axis command current $i_{d,cmd}$ that is independent to the q -axis command current $i_{q,cmd}$. The control law of (3.1) specified the phase of the voltage vector, thus $i_{d,cmd}$ is no longer independent to $i_{q,cmd}$. Therefore, the control law proposed in (2.37) will be used in this section, which is restated here as

$$\begin{aligned} v_q &= R_c i_{q,cmd} + \omega_e L i_{d,cmd} + K_{ec} \omega_m, \\ v_d &= R_c i_{d,cmd} - \omega_e L i_{q,cmd}. \end{aligned} \quad (3.76)$$

Applying the control voltage (3.76) into the motor dynamics model (3.75), and assuming zero order hold sampling, the q -axis closed loop motor dynamics at the k th sample becomes

$$\begin{aligned} L \frac{di_q}{dt}(k) &= [R_c i_{q,cmd}(k) - R i_q(k)] \\ &+ [\omega_e(k-1)L i_{d,cmd}(k) - \omega_e(k)L i_d(k)] + [K_{ec} \omega_m(k-1) - K_e \omega_m(k)] \end{aligned}, \quad (3.77)$$

Assume that the motor mechanical dynamics is much slower than electrical dynamics, and the rotor speed does not change significantly, i.e. $\omega_m(k) \approx \omega_m(k-1)$. Substitute the current error and parameter error definition (3.28) into (3.77). The q -axis closed loop motor dynamics can be written as

$$L \frac{di_q}{dt}(k) = [R_c \Delta i_q(k) - \Delta R i_q(k)] + \omega_e(k)L \Delta i_d(k) - \Delta K_e \omega_m(k). \quad (3.78)$$

It is reasonable to assume that the physical parameters vary significantly slower than the electrical dynamics or mechanical dynamics. Parameter identified by using data

history several samples ago would not introduce significant error. The quantity $\frac{di_q}{dt}(k)$ is not available in real time, but it can be approximated by extrapolation and used in (3.78) for finding parameter difference. Substituting the first order approximation

$$\frac{di_q}{dt}(k) = \frac{i_q(k+1) - i_q(k)}{T} \text{ into (3.78) yields,}$$

$$\frac{L}{T} [i_q(k+1) - i_q(k)] = [R_c \Delta i_q(k) - \Delta R i_q(k)] + \omega_e(k) L_c \Delta i_d(k) - \Delta K_e \omega_m(k). \quad (3.79)$$

Reorganizing (3.79), we have the parameter error equation

$$i_q(k) \Delta R + \omega_m(k) \Delta K_e = R_c \Delta i_q(k) - \omega_e(k) L \Delta i_d(k) - \frac{L}{T} [i_q(k+1) - i_q(k)]. \quad (3.80)$$

Similarly, the d -axis closed loop motor dynamics at the k th sample is

$$L \frac{di_d}{dt}(k) = [R_c \Delta i_d(k) - \Delta R i_d(k)] - \omega_e(k) L \Delta i_q(k), \quad (3.81)$$

Substituting the first order approximation $\frac{di_d}{dt}(k) = \frac{i_d(k+1) - i_d(k)}{T}$ into (3.81), another

parameter error equation is obtained as

$$i_d(k) \Delta R = R_c \Delta i_d(k) - \omega_e(k) L \Delta i_q(k) - \frac{L}{T} [i_d(k+1) - i_d(k)]. \quad (3.82)$$

Once again, at steady state $\frac{di_d}{dt} \approx 0$ and $\frac{di_q}{dt} \approx 0$, and equation (3.80) and (3.82)

become

$$i_q(k) \Delta R + \omega_m(k) \Delta K_e = R_c \Delta i_q(k) - \omega_e L \Delta i_d(k) \quad (3.83)$$

$$i_d(k) \Delta R = R_c \Delta i_d(k) - \omega_e(k) L \Delta i_q(k) \quad (3.84)$$

Solving (3.83) and (3.84) for ΔR and ΔK_e gives

$$\Delta R = \frac{1}{i_d(k)} [R_c \Delta i_d(k) - \omega_e(k) L \Delta i_q(k)], \quad (3.85)$$

$$\Delta K_e = \frac{1}{\omega_m(k)} [R_c \Delta i_q(k) - \omega_e(k) L \Delta i_d(k) - i_q(k) \Delta R]. \quad (3.86)$$

To avoid the singularity and computation noise when $i_d(k), \omega_m(k)$ are close to zero, thresholds are set for the reciprocal calculation of $i_d(k), \omega_m(k)$. In the estimation, the measured i_q, i_d , and ω_m are used to calculate the parameter estimation.

3.6.2 Proof of unbiased estimation

Next, consider using the noisy measured signals for parameter estimation. Since parameters vary much slower than the motor dynamics, and therefore can be treated as constants for any specific motor. Equation (3.82) with the noise terms becomes,

$$[i_d(k) + n_{id}(k)] \Delta R = R_c [\Delta i_d(k) - n_{id}(k)] - n_p [\omega_m(k) + n_\omega(k)] + L [\Delta i_q(k) - n_{iq}(k)] + n_{vd}, \quad (3.87)$$

where the state variables i_q, i_d , and ω_m are deterministic; $n_{id}(k), n_\omega(k), n_{iq}(k)$ and $n_{vd}(k)$ are sensor noise and actuator disturbance. The parameter error ΔR becomes

$$\Delta R = \frac{1}{[i_d(k) + n_{id}(k)]} \left\{ R_c [\Delta i_d(k) - n_{id}(k)] - n_p [\omega_m(k) + n_\omega(k)] + L [\Delta i_q(k) - n_{iq}(k)] + n_{vd} \right\}. \quad (3.88)$$

A constant threshold $\varepsilon > 0$ is set to avoid the singularity when the measured **d**-axis current is near zero, the estimated current error is calculated as

$$\Delta R = \frac{\text{sign}([i_d(k) + n_{id}(k)])}{\max([i_d(k) + n_{id}(k)], \varepsilon)} \left\{ R_c [\Delta i_d(k) - n_{id}(k)] - n_p [\omega_m(k) + n_\omega(k)] + L [\Delta i_q(k) - n_{iq}(k)] + n_{vd} \right\}. \quad (3.89)$$

With the noise terms in (3.89), we need to show that quantity obtained by (3.89) is an unbiased estimation of the deterministic parameter error in (3.85). Assume the noise terms in (3.89) are Gaussian and independent, i.e.

$$E[i_d(k) \cdot n_{id}(k)] = 0, \quad (3.90a)$$

$$E[n_{id}(k) \cdot n_{id}(k)] = S_{id}, \quad (3.90b)$$

$$E[n_\omega(k) \cdot n_\omega(k)] = S_\omega, \quad (3.90c)$$

$$E[n_{id}(k) \cdot n_\omega(k)] = 0. \quad (3.90d)$$

Thus at any instant, the measured d -axis current (denoted as x) is Gaussian, and its probability density function is

$$x \sim f_d(i_d) = N(i_d, S_d) = \frac{1}{\sqrt{2\pi S_d}} \exp\left(-\frac{(x - i_d)^2}{2S_d}\right). \quad (3.91)$$

Let

$$y = \frac{\text{sign}(x)}{\max(|x|, \varepsilon)} = \frac{\text{sign}([i_d(k) + n_{id}(k)])}{\max([i_d(k) + n_{id}(k)], \varepsilon)}, \quad (3.92)$$

and

$$y_1 = R_c [\Delta i_d(k) - n_{id}(k)] - n_p [\omega(k) + n_\omega(k)] + L [\Delta i_q(k) - n_{iq}(k)] + n_{vd}. \quad (3.93)$$

Equation (3.89) is the production of y and y_1 . The variable y_1 is a summation of several Gaussian variables, therefore it is also Gaussian. The variable y is a function of the measured current x , and it can be re-written as

$$y = \begin{cases} \frac{1}{x} & |x| > \varepsilon \\ \frac{1}{\varepsilon} & |x| \leq \varepsilon \end{cases}. \quad (3.93)$$

The probability density function of y (please see Appendix I for detailed derivation) can be found from

$$f_y(y) = \begin{cases} \frac{1}{\left|g'\left(\frac{1}{y}\right)\right|} f_x\left(\frac{1}{y}\right) = \frac{1}{y^2} f_x\left(\frac{1}{y}\right), & -\frac{1}{\varepsilon} \leq y \leq \frac{1}{\varepsilon} \\ 0, & |y| > \frac{1}{\varepsilon} \end{cases} \quad (3.94)$$

where

$$\frac{1}{y^2} f_x\left(\frac{1}{y}\right) = \frac{1}{y^2} \frac{1}{\sqrt{2\pi S_d}} \exp\left(-\frac{i_d^2 \left(y - \frac{1}{i_d}\right)^2}{y^2 2S_d}\right), \quad (3.95)$$

The equation (3.94) can be used to calculate the mean of the first item in the (3.89). Unfortunately, the analytical solution of the mean and the variance would be extremely complicated if not insolvable. Monte Carlo simulation is used to find the mean of the above random signals. In the simulation, 5000 zero mean unit variance random numbers are generated as the noise. A series of numbers μ_x from -10 to 10 are generated as the deterministic component. Actual random variable $x \sim N(\mu_x, 1)$ were the sum of the noise and the deterministic component. For each deterministic component number, the inverse of the 5000 random numbers are calculated and the mean of inverse is then calculated. Figure 3.11 shows the distribution of $y = 1/x$ for four typical deterministic component values μ_x . When μ_x is near zero, the threshold ε limits a significant amount of the $y = 1/x$ values. Thus the distribution curve is heavily distorted. On the other hand, when

the mean of x is relatively far from zero, $y = 1/x$ is close to a normal distribution with mean of $1/\mu(x)$.

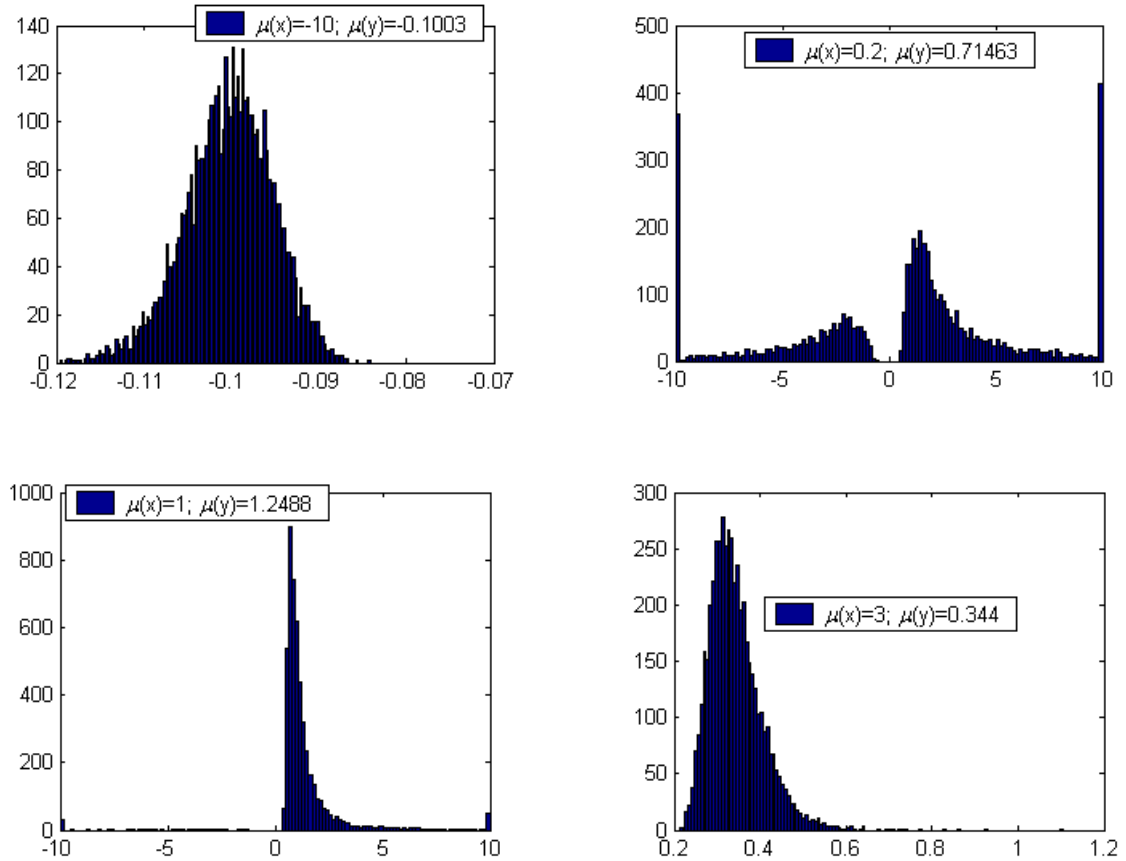


Figure 3.11 The distribution of $y = 1/x$ given $x \sim N(\mu(x), 1)$ for several values of $\mu(x)$.

Figure 3.12 compares the simulated mean of $y = 1/x$ with the algebraic function $y = 1/\mu(x)$ for $\mu(x) \in [-10, 10]$. The simulation indicates that if the deterministic component is above 2 times higher than the noise variance, the mean of the inverse random signal is approximately equal to the inverse of the deterministic component regardless of the value of the threshold ε . However, the value of ε affects the accuracy when the μ_x is close to zero.

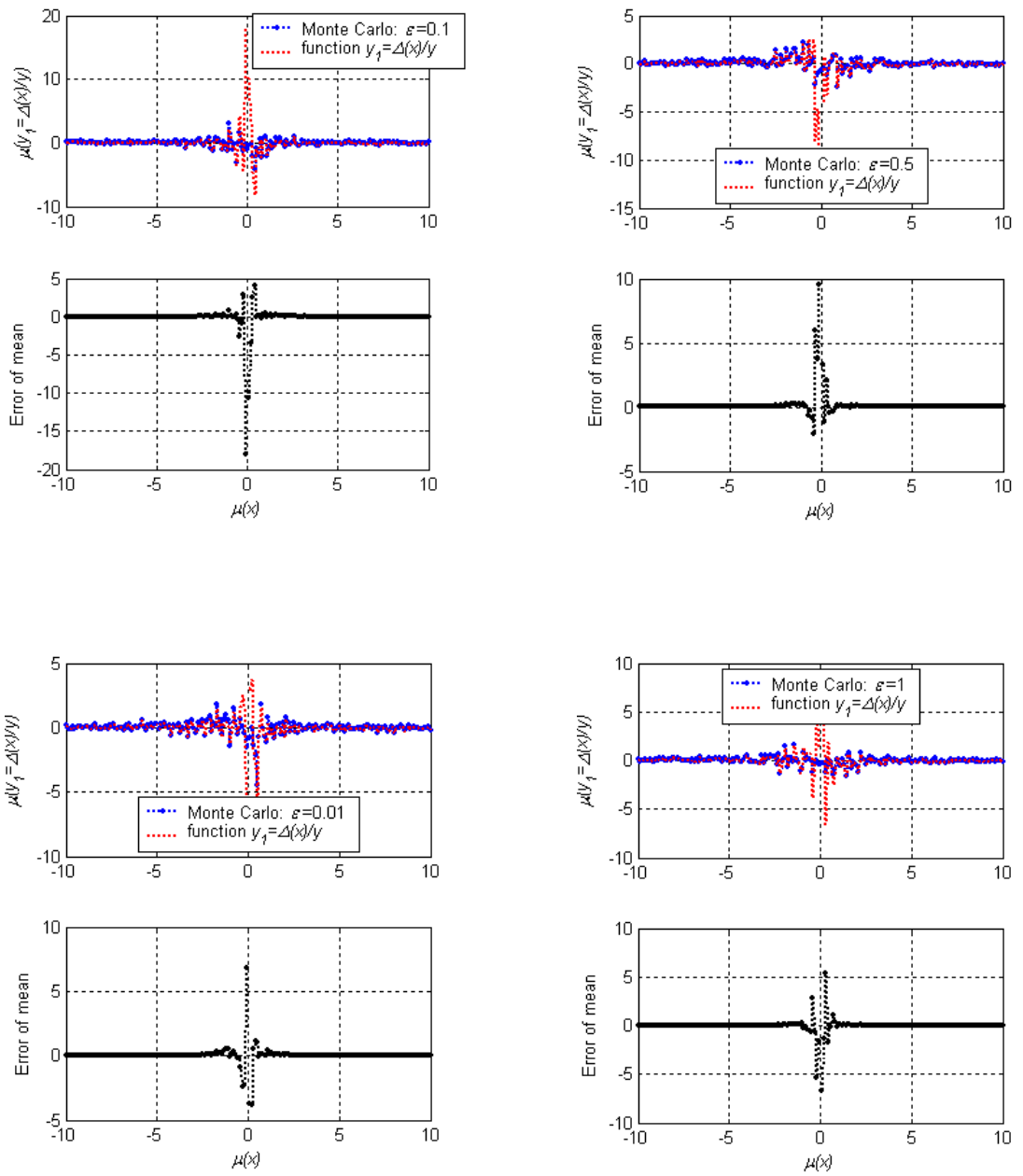


Figure 3.12 The mean of $y = (x_r - x)/x$ given $x \sim N(\mu(x), 1)$ compares to the algebraic function $y = (x_r - \mu(x))/\mu(x)$

Similarly, Monte Carlo simulation is used to verified the consistency of the function,

$$x_1 = \frac{\text{sign}([i_d(k) + n_{id}(k)])}{\max([|i_d(k) + n_{id}(k)|], \varepsilon)} R_c [i_{d,cmd}(k) - i_d(k) - n_{id}(k)].$$

In addition to the inverse of random signal, another deterministic number is generated to simulate the $i_{d,cmd}$. Figure 3.13 indicates that the random signal means are consistent to the algebraic function of the deterministic numbers when the signal to noise ratio (SNR) is higher than 4. This gives the conditions for unbiased estimation of the x_1 .

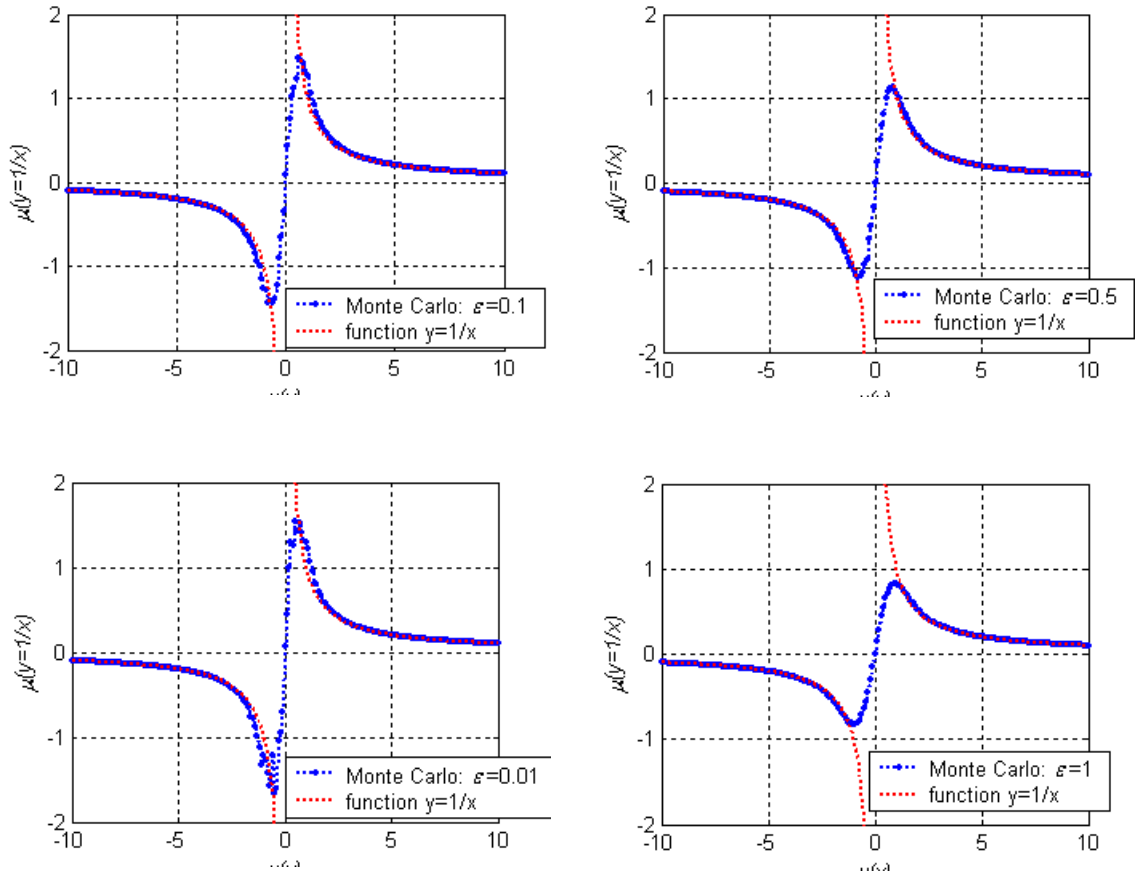


Figure 3.13 The mean of $y = 1/x$ compares to $1/\mu(x)$.

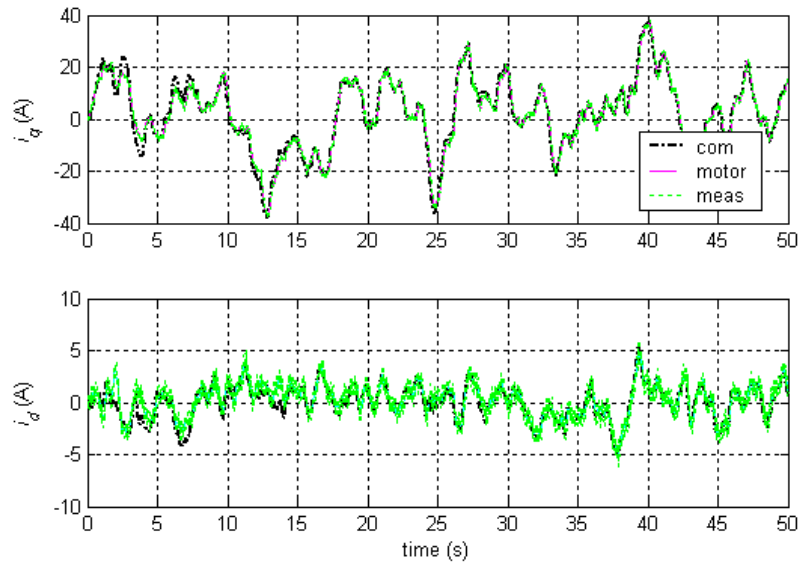
Since $n_o(k)$, n_{vd} and $n_{iq}(k)$ are independent to $n_{id}(k)$, the mean of the remaining two terms on the right hand side of the equation (3.89) will be approximately equal to the

algebraic function value of the means if the inverse of i_d is consistent to the algebraic function value of its deterministic component. This concludes that the ΔR estimation algorithm (3.89) is unbiased when the deterministic signal is selected to ensure a SNR higher than 4. In a similar manner, the unbiased estimation condition can be found for the ΔK_e estimation (3.86).

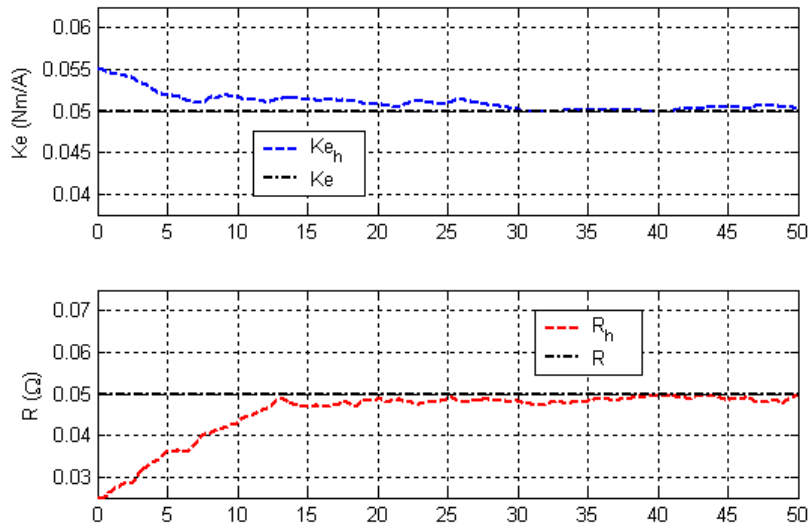
This algorithm is named the ***dq***-solver because it engaged both ***q*** and ***d***-axis current feedback for parameter estimation. The simulation result of the ***dq***-solver estimation algorithm is shown in Figure 3.14. The parameter estimation integration gains were set as

$$C_1(k) = C_2(k) = \begin{cases} 0.1 & i_d(k) \geq 0.1, \omega_m(k) \geq 0.1, \\ 0 & \textit{otherwise}. \end{cases} \quad (3.44)$$

The performance of parameter estimation was not as good as that of the algorithm using Gram Schmidt orthonormalization (as shown in Figure 3.13). The main reason was that the resistance error ΔR calculation only involved the tracking error of i_d , and ΔK_e calculation used ΔR results. Because of the operating principle of the BLDC motor, the magnitude of i_d is usually lower than that of i_q . This would make the parameter estimation slower and noisier than that of the Gram Schmidt orthonormalization algorithm in previous section.



(a) Current tracking results



(b) Parameter estimation results

Figure 3.14 The qd -solver simulation results: +50% R and -15% K_e

3.7 Conclusion

Single parameter estimation and multiple parameter estimation for a feedforward BLDC motor controller have been proposed in this chapter. Stability of these estimation schemes has been proved. The stability proof also indicated that it was possible to improve the performance of estimation schemes by tightening the bound of errors. Thus, the state transition matrix method and approximate motor speed error feedback were proposed to improve the parameter estimation performance. In addition, it was shown that the estimation performance could be improved by engaging more data in the parameter error calculation. Another improvement for multi-parameter estimation using the Gram-Schmidt orthonormalization was proposed to address the data loss in those samples discarded for avoiding singularity. Stability of this method was proved. The *dq*-axis current i_q and i_d are always calculated in pair. So i_d can be used for parameter estimation too. An estimation algorithm using both i_q and i_d feedback was proposed and unbiased estimation condition was shown by Monte Carlo simulation.

Chapter 4 RECURSIVE PARAMETER IDENTIFICATION

In Chapter 3, several parameter estimation algorithms were developed for a feedforward BLDC control law of equation (3.1). These algorithms were based on the same principle of solving parameter errors algebraically from the reference current tracking errors. Among the several performance improvement methods investigated, it was shown that the better accuracy and faster convergence could be achieved by engaging more samples in the parameter error calculation. One of the side effects of engaging more data was that the parameter-updating algorithm lagged a few samples behind the feedforward controller. In addition, it also increased the random access memory (RAM) consumption when implementing the algorithms in a microcontroller. Obviously, there exists a hard limit on such explicit historic data usage. To further improve the parameter estimation performance, we must explore different mechanisms of parameter identification or data history usage, for instance, the recursive algorithms. These algorithms define some state variables (such as covariance matrices) to store useful information from historic data, and update these states at every sample. Thus, when the outputs are calculated in real time from the states, all the past data history are implicitly engaged in these calculations.

The Recursive Least Square (RLS) method (Ljung 2002) and the Extended Kalman Filter (EKF) (Burl 1995, Andrews 2001) are two widely used recursive system identification algorithms. The RLS algorithm assumes a linear discrete model of the subject system, and estimates the coefficients in the linear model by using the redundant input and output signal measurement. The EKF assumes independent Gaussian noises

and disturbances in a nonlinear system, and estimates the states through linearization at every sample. Some system parameters can be estimated as augmented states. In this chapter, the Recursive Least Square (RLS) and the Extended Kalman Filter (EKF) will be investigated for the BLDC motor parameter identification application.

4.1 Discrete Model of the BLDC Motor Dynamics

In applications such as the EPAS, the BLDC motor controllers are almost exclusively implemented by a digital microcontroller, which requires discrete control algorithms. In many references (Ljung 2002, Burl 1995, Andrews 1993), the RLS algorithm and the EKF are designed on the discrete model of the plant. Therefore, an approximate discrete BLDC motor dynamics model will be derived in this section for the estimation algorithm design purpose.

The continuous time model of the BLDC motor electrical dynamics is restated here as

$$\begin{cases} L \frac{di_q}{dt} = -Ri_q - \omega_e L i_d - K_e \omega_m + V_q, \\ L \frac{di_d}{dt} = -Ri_d + \omega_e L i_q + V_d, \\ \tau_{out} = K_e i_q. \end{cases} \quad (4.1)$$

Let $nT \leq t < (n+1)T$, define the state vector $x = [i_q, i_d]^T$ and control input vector $u = [V_q - K_e \omega_m, V_d]^T$. Introduce the time constant of the stator coil $\tau = L/R$. The state space model of the BLDC motor can be obtained as

$$\begin{cases} \dot{x} = \begin{pmatrix} -\frac{1}{\tau} & -\omega_e \\ \omega_e & -\frac{1}{\tau} \end{pmatrix} \cdot x + \frac{1}{L} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \cdot u, \\ y = [K_e \quad 0] \cdot x. \end{cases} \quad (4.2)$$

Mathematical solution of the continuous dynamics (4.2) is hard to obtain due to the fact that the motor velocity ω_m is a function of time. However, ω_m is governed by the mechanical inertia of the motor shaft, therefore it is reasonable to assume that ω_e changes slowly and can be considered as a constant within the typical sampling interval

of 1~5 milliseconds. Let $A = \begin{pmatrix} -\frac{1}{\tau} & -\omega_e \\ \omega_e & -\frac{1}{\tau} \end{pmatrix}$, the characteristic equations of (4.2) can be

found as

$$(sI - A)^{-1} = \begin{pmatrix} s + \frac{1}{\tau} & \omega_e \\ -\omega_e & s + \frac{1}{\tau} \end{pmatrix}^{-1} = \frac{1}{\left(s + \frac{1}{\tau}\right)^2 + \omega_e^2} \begin{pmatrix} s + \frac{1}{\tau} & -\omega_e \\ \omega_e & s + \frac{1}{\tau} \end{pmatrix}. \quad (4.3)$$

The state transition matrix at each sampling moment can be obtained by

$$\Phi = L^{-1}(sI - A)^{-1} = \exp\left(-\frac{t}{\tau}\right) \begin{pmatrix} \cos(\omega_e t) & -\sin(\omega_e t) \\ \sin(\omega_e t) & \cos(\omega_e t) \end{pmatrix}, \quad (4.4)$$

where L^{-1} is the inverse Laplace operator. The state equation (4.2) can be solved as

$$x[(n+1)T] = \Phi[nT, (n+1)T] \cdot x(nT) + \int_{nT}^{(n+1)T} \Phi(nT, t) G u dt, \quad (4.5)$$

$$\text{where } G = \frac{1}{L} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

Let $\theta = (t - kT)$, and substitute (4.4) into (4.5), the solution of (4.2) becomes

$$\begin{aligned} x[(n+1)T] &= \exp\left(-\frac{T}{\tau}\right) \begin{pmatrix} \cos(\omega_e T) & -\sin(\omega_e T) \\ \sin(\omega_e T) & \cos(\omega_e T) \end{pmatrix} x(nT) \\ &+ \frac{1}{L} \left[\int_0^T \exp\left(-\frac{\theta}{\tau}\right) \begin{pmatrix} \cos(\omega_e \theta) & -\sin(\omega_e \theta) \\ \sin(\omega_e \theta) & \cos(\omega_e \theta) \end{pmatrix} d\theta \right] \begin{bmatrix} V_q - K_e \omega_m \\ V_d \end{bmatrix}. \end{aligned} \quad (4.6)$$

The integration term in (4.6) must be solved so as to obtain the discrete dynamics

equation. The integration term can be rewritten as $\begin{bmatrix} I(k) & -J(k) \\ J(k) & I(k) \end{bmatrix}$, where

$$I(k) = \int_0^T \exp\left(-\frac{\theta}{\tau}\right) \cos[\omega_e(k)\theta] d\theta, \quad (4.7a)$$

$$J(k) = \int_0^T \exp\left(-\frac{\theta}{\tau}\right) \sin[\omega_e(k)\theta] d\theta. \quad (4.7b)$$

With the assumption that ω_e is constant within the k th sampling interval, $I(k)$ and $J(k)$ in (4.7) can be solved through some mathematical manipulation (see Appendix II for details) as

$$\begin{pmatrix} I(k) \\ J(k) \end{pmatrix} = \frac{1}{[\omega_e(k)\tau]^2 + 1} \begin{bmatrix} \omega_e(k)\tau & 1 \\ -1 & \omega_e(k)\tau \end{bmatrix} \begin{pmatrix} \tau \cdot \exp\left(-\frac{T}{\tau}\right) \sin[\omega_e(k)T] \\ -\tau \left\{ \exp\left(-\frac{T}{\tau}\right) \cos[\omega_e(k)T] - 1 \right\} \end{pmatrix}. \quad (4.8)$$

Let $H(k) = \begin{pmatrix} \cos[\omega_e(k)T] & -\sin[\omega_e(k)T] \\ \sin[\omega_e(k)T] & \cos[\omega_e(k)T] \end{pmatrix}$, Substituting (4.8) back into the integration

term in (4.6), the approximate discrete motor dynamics is obtained as

$$\begin{pmatrix} i_q(k+1) \\ i_d(k+1) \end{pmatrix} = e^{-\frac{T}{\tau}} H(k) \begin{pmatrix} i_q(k) \\ i_d(k) \end{pmatrix} - \frac{1}{L} \begin{pmatrix} I(k) \\ J(k) \end{pmatrix} K_e \omega_m(k) + \frac{1}{L} \begin{bmatrix} I(k) & -J(k) \\ J(k) & I(k) \end{bmatrix} \begin{pmatrix} V_q(k) \\ V_d(k) \end{pmatrix}. \quad (4.9)$$

The discrete dynamics equation (4.9) approximates the continuous dynamics in (4.1) with the assumption that the control voltage and motor velocity hold constant within each sampling interval. Since an accurate discrete solution of (4.1) is not available, equation (4.9) will be used as the discrete model of the BLDC motor for system identification purpose.

4.2 Recursive Least Square Estimation

Given the approximate discrete dynamics model of (4.9), we will investigate if the motor parameters can be identified through the recursive least square (RLS) method in this section.

The Least Square method assumes that the object system conforms a linear discrete relationship between the measurement y and the input u (Ljung 2002) as

$$y(k) + a_1 y(k-1) + \dots + a_p y(k-p) = b_1 u(k-1) + \dots + b_q u(k-q) + \varepsilon, \quad (4.10)$$

where the coefficients $a_1, \dots, a_p, b_1, \dots, b_q$ are system characteristics constants, p and q are positive integers, and ε is the modeling error. Note that both the measurement y and the input u can be vectors, and the coefficients will be in matrix form in this case. Rewrite (4.10) in the form of

$$y(k) = \mathbf{x}(k)\boldsymbol{\theta} + \varepsilon, \quad (4.11)$$

where $\mathbf{x}(k) = [y(k-1), \dots, y(k-p), u(k-1), \dots, u(k-q)]$, $\boldsymbol{\theta} = [-a_1, \dots, -a_p, b_1, \dots, b_q]^T$.

The optimal estimation of the coefficient vector $\boldsymbol{\theta}$ minimizes the modeling error ε . A typical way to find the optimal estimation of $\boldsymbol{\theta}$ is to define a quadratic cost function of the modeling error as

$$Q = \frac{1}{n} \sum_{k=1}^n [y(k) - \mathbf{x}(k)\boldsymbol{\theta}]^T [y(k) - \mathbf{x}(k)\boldsymbol{\theta}] \quad (4.12)$$

Denote $\hat{\boldsymbol{\theta}}$ as the optimal estimation of $\boldsymbol{\theta}$ that minimizes the quadratic cost Q . Since Q

is quadratic, $\hat{\boldsymbol{\theta}}$ can be found by setting partial derivative $\frac{\partial Q}{\partial \boldsymbol{\theta}}$ to zero as

$$\left. \frac{\partial Q}{\partial \boldsymbol{\theta}} \right|_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}} = -\frac{2}{n} \sum_{k=1}^n [\mathbf{x}(k)^T y(k) - \mathbf{x}(k)^T \mathbf{x}(k)\hat{\boldsymbol{\theta}}] = 0, \quad (4.13)$$

which gives

$$\hat{\boldsymbol{\theta}} = \sum_{k=1}^n [\mathbf{x}(k)^T y(k)] \left\{ \sum_{k=1}^n [\mathbf{x}(k)^T \mathbf{x}(k)] \right\}^{-1}. \quad (4.14)$$

The inverse calculation in (4.14) is computationally expensive and hard to implement in real time. Fortunately, there exists a recursive solution to the matrix inverse calculation.

To simplify the notation, let's denote $\mathbf{X}_n^T \mathbf{X}_n = \sum_{k=1}^n [\mathbf{x}(k)^T \mathbf{x}(k)]$, and denote its inverse as

$\mathbf{P}(n) = [\mathbf{X}_n^T \mathbf{X}_n]^{-1}$. The inverse matrix at step $n+1$ can be expanded as

$$\mathbf{P}(n+1) = [\mathbf{X}_{n+1}^T \mathbf{X}_{n+1}]^{-1} = \left\{ \sum_{k=1}^n [\mathbf{x}(k)^T \mathbf{x}(k)] + \mathbf{x}(n+1)^T \mathbf{x}(n+1) \right\}^{-1},$$

Or

$$\mathbf{P}(n+1) = [\mathbf{P}(n)^{-1} + \mathbf{x}_{n+1}^T \mathbf{x}_{n+1}]^{-1}, \quad (4.15)$$

where $\mathbf{x}_{n+1}^T \mathbf{x}_{n+1} = \mathbf{x}(n+1)^T \mathbf{x}(n+1)$. The matrix inverse in (4.15) can be solved by the

Woolbury Matrix Identity (Golub and Van Loan 1996)

$$(\mathbf{A} + \mathbf{BCD})^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1} \mathbf{B} (\mathbf{C} + \mathbf{DA}^{-1} \mathbf{B})^{-1} \mathbf{DA}^{-1}. \quad (4.16)$$

Comparing (4.15) and (4.16), let $\mathbf{A} = \mathbf{P}(n)^{-1}$, $\mathbf{B} = \mathbf{x}_{n+1}^T$, $\mathbf{C} = I$, $\mathbf{D} = \mathbf{x}_{n+1}$, where I is the unitary matrix. The inverse matrix at $(n+1)$ th step can be solved recursively as

$$\mathbf{P}(n+1) = \mathbf{P}(n) - \mathbf{P}(n) \mathbf{x}_{n+1}^T \left[\mathbf{I} + \mathbf{x}_{n+1} \mathbf{P}(n) \mathbf{x}_{n+1}^T \right]^{-1} \mathbf{x}_{n+1} \mathbf{P}(n). \quad (4.17)$$

Substituting (4.17) into the parameter estimation equation (4.14), the parameter estimation at $(n+1)$ th step becomes

$$\hat{\boldsymbol{\theta}}(n+1) = \hat{\boldsymbol{\theta}}(n) + \mathbf{P}(n) \mathbf{x}(n+1)^T \left[\mathbf{I} + \mathbf{x}(n+1) \mathbf{P}(n) \mathbf{x}(n+1)^T \right]^{-1} [\mathbf{y}(n+1) - \mathbf{x}(n+1) \hat{\boldsymbol{\theta}}(n)]. \quad (4.18)$$

Let

$$\mathbf{K}(n+1) = \mathbf{P}(n)\mathbf{x}(n+1)^T [1 + \mathbf{x}(n+1)\mathbf{P}(n)\mathbf{x}(n+1)^T]^{-1}, \quad (4.19)$$

the recursive least square (RLS) parameter estimation is given as

$$\hat{\boldsymbol{\theta}}(n+1) = \hat{\boldsymbol{\theta}}(n) + \mathbf{K}(n+1)[\mathbf{y}(n+1) - \mathbf{x}(n+1)\hat{\boldsymbol{\theta}}(n)]. \quad (4.20)$$

In a BLDC motor, the measurement vector \mathbf{y} consists of the q -axis current i_q and the d -axis current i_d , i.e. $\mathbf{y}(k) = [i_q(k), i_d(k)]^T$. Comparing (4.9) to (4.10), the discrete state input vector is selected as $\mathbf{x}(k) = [i_q(k-1), i_d(k-1), V_q(k-1), V_d(k-1), \omega_m(k-1)]$. Substituting $\mathbf{y}(k)$ and $\mathbf{x}(k)$ into (4.17), (4.19) and (4.20), the RLS parameter estimation algorithm can be implemented for the BLDC motor application. The inverse matrix $\mathbf{P}(n)$ is the state matrix (5×5) in the RLS estimation, and it keeps information obtained from historic data. The estimated parameters $\hat{\boldsymbol{\theta}}$ is a 5×2 matrix, and its steady state value is a least square estimation of the linear discrete model between $\mathbf{y}(k)$ and $\mathbf{x}(k)$.

4.3 RLS Simulation Results

The RLS algorithm is implemented for the BLDC motor application in Simulink®. The following motor parameter values were used in the simulation: $R = 0.05\Omega$, $Ke = 0.05 \text{ Nm/A}$, $L = 1 \times 10^{-4} \text{ H}$, $N_p = 3$. In the first simulation, the motor was running at a constant velocity $\omega_m = 100 \text{ rad/s}$. The simulation ran at a fixed step length of 2ms. Figure 4.1 showed the RLS parameter estimation results. The steady state values of the coefficients were shown as legends in these figures. The parameter estimation converged quickly (less than 2 seconds). The motor parameter R , Ke and L are not directly shown in the estimation results. But the analytical values of the estimated parameters ($a_1 \sim a_5$),

$b_1 \sim b_5$) can be calculated with the given parameter values of R , Ke and L .

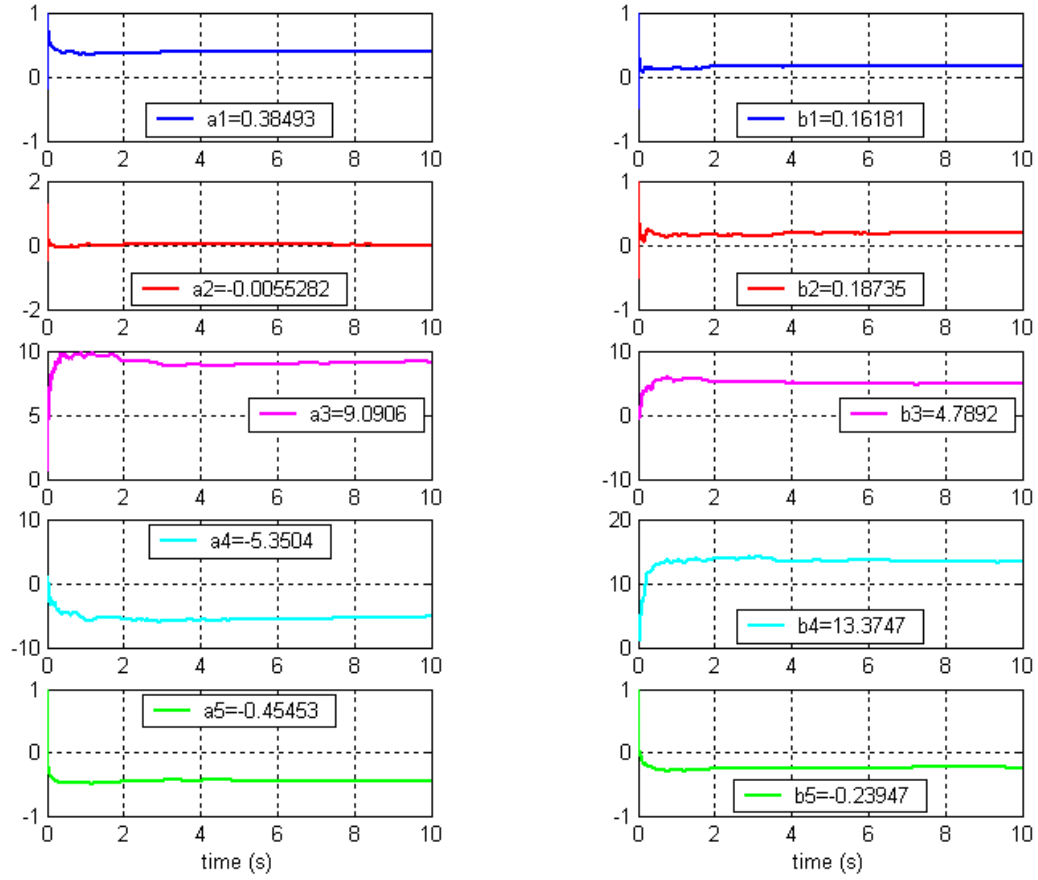


Figure 4.1 RLS estimation simulation results of $a_1 \sim a_5$ & $b_1 \sim b_5$: constant $\omega_m = 100 \text{ rad/s}$.

In each plot the steady state value is shown.

Substitute $R = 0.05 \Omega$, $Ke = 0.05 \text{ Nm/A}$, $L = 1 \times 10^{-4} \text{ H}$, $N_p = 3$ and $\omega_m = 100 \text{ rad/s}$

into (4.8) and (4.9), the coefficient matrices in (4.9) can be calculated as

$$\begin{bmatrix} a_1 & a_2 \\ b_1 & b_2 \end{bmatrix} = e^{-\frac{T}{\tau}} H(k) = \begin{bmatrix} 0.3036 & -0.2077 \\ 0.2077 & 0.3036 \end{bmatrix},$$

$$\begin{bmatrix} a_3 & a_4 \\ b_3 & b_4 \end{bmatrix} = \frac{1}{L} \begin{bmatrix} I(k) & -J(k) \\ J(k) & I(k) \end{bmatrix} = \begin{bmatrix} 12.0737 & -3.0898 \\ 3.0898 & 12.0737 \end{bmatrix},$$

$$\begin{bmatrix} a_5 \\ b_5 \end{bmatrix} = -\frac{1}{L} \begin{pmatrix} I(k) \\ J(k) \end{pmatrix} K_e = \begin{bmatrix} -0.6037 \\ -0.1545 \end{bmatrix}.$$

Comparing the steady state simulation results with the analytical solutions, the RLS estimation approximated the analytical discrete model (4.9) closely, but there were about 20%~60% error in the estimation results.

Figure 4.2 showed the RLS estimation simulation results with random rotor velocity. It took longer time (about 7 seconds) for the estimated parameters to converge to their steady state values. Since the velocity was not longer constant, the parameter matrices in (4.9) could not be solved analytically. However, the RLS algorithm converged to a linear approximate of the nonlinear motor dynamics. The steady state values of $a_1 = 0.93168$ and $b_2 = 0.99908$ implied the current measurements are mostly correlated to their previous value.

In summary, the RLS parameter estimation algorithm was capable of finding a linear approximate of the motor dynamics. The estimated parameters were close to their analytical values if the motor was running at speed mode (i.e. motor velocity was constant). However, the RLS algorithm would not be very directly helpful if the BLDC motor was running at varying velocity, as the linear discrete model assumption was not valid. In addition, the motor parameters of interests R , Ke and L are not directly accessible in the RLS estimation results. This would be inconvenient to use the RLS algorithm with the feedforward motor inverse dynamics controllers.

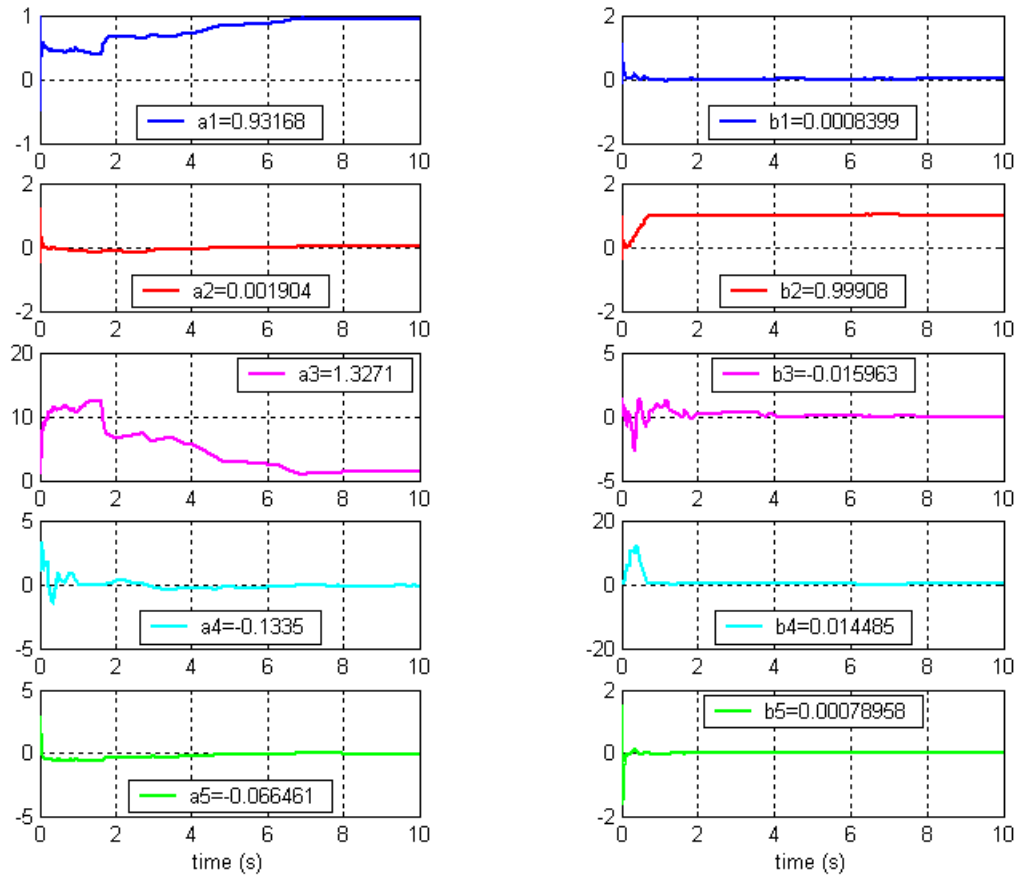


Figure 4.2 RLS estimation simulation results of $a_1 \sim a_5$ & $b_1 \sim b_5$: variant velocity.

In each plot the steady state value is shown.

4.4 Extended Kalman Filter

The Extended Kalman Filter (EKF), as indicated by its name, is an extension of the linear Kalman Filter to nonlinear systems. It is a commonly used algorithm for parameter estimation problems. Computationally, the EKF has similar steps as that of the RLS algorithm. It estimates the covariance matrix recursively in a way similar to how the RLS algorithm solves the inverse matrix in (4.17), and then it updates the Kalman gains from the covariance matrix, similar to the gain calculation in (4.19). The states are updated

from the error between the measurement and the model prediction based on the previous step state values. However, the EKF takes advantage of the existing information of the nonlinear plant model, and involves the model linearization in the covariance matrix and state calculation. As a result, the EKF is usually a good solution for nonlinear system estimation including the parameter identification problems. When the EKF is implemented for system identification problems, the parameters are often treated as new state variables subjected to some stochastic dynamics. The original plant model is augmented with the parameters as new state variables. Using the same recursive procedure, the parameter can be estimated with the original state variables of the system. In this section the EKF will be implemented to the BLDC motor state and parameter estimation problems.

The EKF has a standard implementation procedure (Burl 1995, Andrews 2001). First of all, the plant model will be reformulated to include Gaussian actuator perturbation and measurement noise. Adding the actuator perturbation w_1, w_2 to the approximate BLDC motor discrete model (4.9), the current state equations become

$$\begin{pmatrix} i_q(k+1) \\ i_d(k+1) \end{pmatrix} = e^{-\frac{T}{\tau}} H(k) \begin{pmatrix} i_q(k) \\ i_d(k) \end{pmatrix} - \begin{pmatrix} I(k) \\ J(k) \end{pmatrix} K_e \omega_m(k) + \begin{bmatrix} I(k) & -J(k) \\ J(k) & I(k) \end{bmatrix} \begin{pmatrix} V_q(k) \\ V_d(k) \end{pmatrix} + T \begin{pmatrix} w_1(k) \\ w_2(k) \end{pmatrix}. \quad (4.21a)$$

Assuming parameters K_e and τ are constants subjected to random perturbation w_3 and w_4 , the discrete dynamics of the parameters can be modeled as

$$K_e(k+1) = K_e(k) + T w_3(k), \quad (4.21b)$$

$$\tau(k+1) = \tau(k) + T w_4(k). \quad (4.21c)$$

Let the augmented state be $\mathbf{x} = (i_q, i_d, K_e, \tau)^T$. Equations (4.21a), (4.21b) and (4.21c) formulated the nonlinear state dynamics model of the augmented system. For simplicity, let us denote the nonlinear state model as

$$\mathbf{x}(k+1) = \mathbf{f}(\mathbf{x}(k), \mathbf{u}(k), \mathbf{w}(k)). \quad (4.22)$$

The measurement vector with measurement noise $\mathbf{v} = (v_1, v_2)^T$ becomes

$$\mathbf{y}(k) = \begin{pmatrix} i_q(k) \\ i_d(k) \end{pmatrix} = \mathbf{g}(\mathbf{x}(k), \mathbf{v}(k)) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \mathbf{x}(k) + \begin{pmatrix} v_1(k) \\ v_2(k) \end{pmatrix}. \quad (4.23)$$

It is usually reasonable to assume that the plant perturbation $\mathbf{w} = (w_1, w_2, w_3, w_4)^T$ and measurement noise $\mathbf{v} = (v_1, v_2)^T$ are Gaussian and independent:

$$E[\mathbf{w}(k)\mathbf{w}^T(k+p)] = \Sigma_w \delta(p), \quad (4.24a)$$

$$E[\mathbf{v}(k)\mathbf{v}^T(k+p)] = \Sigma_v \delta(p), \quad (4.24b)$$

$$E[\mathbf{w}(k)] = 0, \quad (4.24c)$$

$$E[\mathbf{v}(k)] = 0, \quad (4.24d)$$

where $\Sigma_w = \begin{bmatrix} S_{w1} & & & 0 \\ & S_{w2} & & \\ & & S_{w3} & \\ 0 & & & S_{w4} \end{bmatrix}$ and $\Sigma_v = \begin{bmatrix} S_{v1} & 0 \\ 0 & S_{v2} \end{bmatrix}$ are the spectral density matrix

of the actuation perturbation and measurement noise, and $\delta(p)$ is the Dirac delta function, and $E[\mathbf{w}(k)\mathbf{w}^T(k+p)]$ is the mathematical expectation operation.

The nonlinear system dynamics model has been formulated in (4.21~24). With the nonlinear state model, we can find its linear approximation functions $\hat{\Phi}(k) = \left. \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \right|_{\mathbf{x}=\hat{\mathbf{x}}(k|k)}$,

$$\hat{\Gamma}(k) = \left. \frac{\partial \mathbf{f}}{\partial \mathbf{w}} \right|_{\mathbf{x}=\hat{\mathbf{x}}(k|k)} \quad \text{and} \quad \hat{\mathbf{C}}(k) = \left. \frac{\partial \mathbf{g}}{\partial \mathbf{x}} \right|_{\mathbf{x}=\hat{\mathbf{x}}(k|k)} .$$

$$\hat{\Phi}(k) = \left. \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \right|_{\mathbf{x}=\hat{\mathbf{x}}(k|k)} = \begin{bmatrix} \left[e^{-\frac{T}{\tau(k|k)}} H(k|k) \right]_{2 \times 2} & -I(k|k)\omega_m(k|k) & \frac{\partial f_1}{\partial \tau}(k|k) \\ & -J(k|k)\omega_m(k|k) & \frac{\partial f_2}{\partial \tau}(k|k) \\ & & 1 & 0 \\ & & 0 & 1 \end{bmatrix}_{4 \times 4} \quad (4.25)$$

where

$$\begin{pmatrix} \frac{\partial f_1}{\partial \tau} \\ \frac{\partial f_2}{\partial \tau} \end{pmatrix} = \frac{T}{\tau^2} e^{-\frac{T}{\tau}} H(k) \begin{pmatrix} i_q(k) \\ i_d(k) \end{pmatrix} - \begin{pmatrix} \frac{\partial I(k)}{\partial \tau} \\ \frac{\partial J(k)}{\partial \tau} \end{pmatrix} K_e \omega_m(k) + \begin{bmatrix} \frac{\partial I(k)}{\partial \tau} & -\frac{\partial J(k)}{\partial \tau} \\ \frac{\partial J(k)}{\partial \tau} & \frac{\partial I(k)}{\partial \tau} \end{bmatrix} \begin{pmatrix} V_q(k) \\ V_d(k) \end{pmatrix},$$

$$\begin{aligned} \frac{\partial I(k)}{\partial \tau} &= -\frac{2\omega_e(k)\tau}{\left\{ [\omega_e(k)\tau]^2 + 1 \right\}^2} \left\{ \omega_e(k)\tau^2 e^{-\frac{T}{\tau}} \sin[\omega_e(k)T] + \tau \left(1 - e^{-\frac{T}{\tau}} \cos[\omega_e(k)T] \right) \right\} \\ &+ \frac{1}{[\omega_e(k)\tau]^2 + 1} \left\{ 2\omega_e(k)\tau \cdot e^{-\frac{T}{\tau}} \sin[\omega_e(k)T] + \left(1 - e^{-\frac{T}{\tau}} \cos[\omega_e(k)T] \right) \right\}, \\ &+ \frac{1}{[\omega_e(k)\tau]^2 + 1} \left\{ \omega_e(k)T e^{-\frac{T}{\tau}} \sin[\omega_e(k)T] - \frac{T}{\tau} e^{-\frac{T}{\tau}} \cos[\omega_e(k)T] \right\} \\ \frac{\partial J(k)}{\partial \tau} &= -\frac{2\omega_e(k)\tau}{\left\{ [\omega_e(k)\tau]^2 + 1 \right\}^2} \left\{ -\tau \cdot e^{-\frac{T}{\tau}} \sin[\omega_e(k)T] + \omega_e(k)\tau^2 \left(1 - e^{-\frac{T}{\tau}} \cos[\omega_e(k)T] \right) \right\} \\ &+ \frac{1}{[\omega_e(k)\tau]^2 + 1} \left\{ -e^{-\frac{T}{\tau}} \sin[\omega_e(k)T] + 2\omega_e(k)\tau \cdot \left(1 - e^{-\frac{T}{\tau}} \cos[\omega_e(k)T] \right) \right\}, \\ &+ \frac{1}{[\omega_e(k)\tau]^2 + 1} \left\{ -\frac{T}{\tau} e^{-\frac{T}{\tau}} \sin[\omega_e(k)T] - \omega_e(k)T e^{-\frac{T}{\tau}} \cos[\omega_e(k)T] \right\} \end{aligned}$$

and T is the sampling time.

$$\hat{\mathbf{\Gamma}}(k) = \left. \frac{\partial \mathbf{f}}{\partial \mathbf{w}} \right|_{\mathbf{x}=\hat{\mathbf{x}}(k|k)} = \begin{bmatrix} T & & 0 \\ & T & \\ 0 & & T \end{bmatrix} \quad (4.26)$$

$$\hat{\mathbf{C}}(k) = \left. \frac{\partial \mathbf{g}}{\partial \mathbf{x}} \right|_{\mathbf{x}=\hat{\mathbf{x}}(k|k)} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}. \quad (4.27)$$

With the linear approximation functions, the a priori covariance matrix is computed recursively from the a posteriori covariance matrix at the previous step

$$\mathbf{\Sigma}_e(\mathbf{k} + 1 | \mathbf{k}) = \hat{\mathbf{\Phi}}(k) \mathbf{\Sigma}_e(\mathbf{k} | \mathbf{k}) \hat{\mathbf{\Phi}}(k)^T + \hat{\mathbf{\Gamma}}(k) \mathbf{\Sigma}_w \hat{\mathbf{\Gamma}}(k)^T. \quad (4.28)$$

The Kalman gain can be computed as

$$\mathbf{G}(\mathbf{k} + 1) = \mathbf{\Sigma}_e(\mathbf{k} + 1 | \mathbf{k}) \hat{\mathbf{C}}(k)^T \left[\hat{\mathbf{C}}(k) \mathbf{\Sigma}_e(\mathbf{k} + 1 | \mathbf{k}) \hat{\mathbf{C}}(k)^T + \mathbf{\Sigma}_v \right]^{-1}, \quad (4.29)$$

and a posteriori covariance matrix is updated as

$$\mathbf{\Sigma}_e(\mathbf{k} + 1 | \mathbf{k} + 1) = \left[\mathbf{I} - \mathbf{G}(\mathbf{k} + 1) \hat{\mathbf{C}}(k) \right] \mathbf{\Sigma}_e(\mathbf{k} + 1 | \mathbf{k}). \quad (4.30)$$

Given the state dynamics model (4.22), we can predict the states at the next step

$$\hat{\mathbf{x}}(k + 1 | k) = \mathbf{f}(\hat{\mathbf{x}}(k | k), \mathbf{u}(k)), \quad (4.31)$$

and predict the measurement as

$$\hat{\mathbf{y}}(k + 1 | k) = \mathbf{g}(\hat{\mathbf{x}}(k + 1 | k)). \quad (4.32)$$

The EKF estimate the state from the measurement by the following equation

$$\hat{\mathbf{x}}(k + 1 | k + 1) = \hat{\mathbf{x}}(k + 1 | k) + \mathbf{G}(\mathbf{k} + 1) [\mathbf{y}(k + 1) - \hat{\mathbf{y}}(k + 1 | k)]. \quad (4.33)$$

The discrete EKF consists of the equations (4.28~30) and (4.33). Besides, the linearization function $\hat{\mathbf{\Phi}}(k)$ involved in equations (4.28) must be computed at every step.

Obviously, the computation cost of the linearization function at every step would be a

challenge for the real time applications. For research purpose, let's assume that the computation power is not a problem.

The EKF can be implemented to the continuous BLDC motor dynamics model. For detail of the continuous EKF, please see Appendix III.

4.5 EKF Simulation Results

The EKF estimation algorithm for the BLDC motor application is implemented in SIMULINK® for simulation. In simulations, the BLDC motor plant model has the following parameter values: $R = 0.05\Omega$, $K_e = 0.05 Nm/A$, $L = 1 \times 10^{-4} H$, $N_p = 3$. In some situations, the effect of parameter error in R and K_e may cancel each other. For example, positive ΔR and negative ΔK_e may end up with the same control voltage v_q calculated with zero parameter errors. It may be harder for the parameter estimation algorithms. Therefore, different combinations of initial parameter errors in R and K_e were simulated to compare the performance of parameter estimation, as shown in Figure 4.3~4.7. Due to some stability issue, the simulation sampling interval had to be less than or equal to 0.5 millisecond. In the figures, the $i_{q,meas}$ $i_{d,meas}$ were the measured currents, which were contaminated by simulated Gaussian noise; the i_q and i_d are the motor plant model calculated current, while the i_{qh} and i_{dh} are the EKF estimated currents, R_h and K_{e_h} are the EKF estimated parameter values. Regardless of initial parameter error combinations, EKF estimation converges to the actual parameter values within 1 second in all simulations. Table 4.1 compares the parameter estimation accuracy of the EKF at steady state, given different value of the Gaussian noise power density Σ_v .

Table 4.1 EKF parameter estimation mean and variance for different noise level.

Noise Power Σ_v	$\mu(\hat{R})$	$\text{var}(\hat{R})$	$\mu(\hat{K}e)$	$\text{var}(\hat{K}e)$
2	1.637×10^{-4}	2.386×10^{-7}	1.514×10^{-4}	1.076×10^{-6}
0.5	1.101×10^{-4}	1.387×10^{-7}	-7.753×10^{-6}	2.707×10^{-8}
0.1	5.341×10^{-5}	9.179×10^{-8}	-1.99×10^{-6}	1.486×10^{-8}

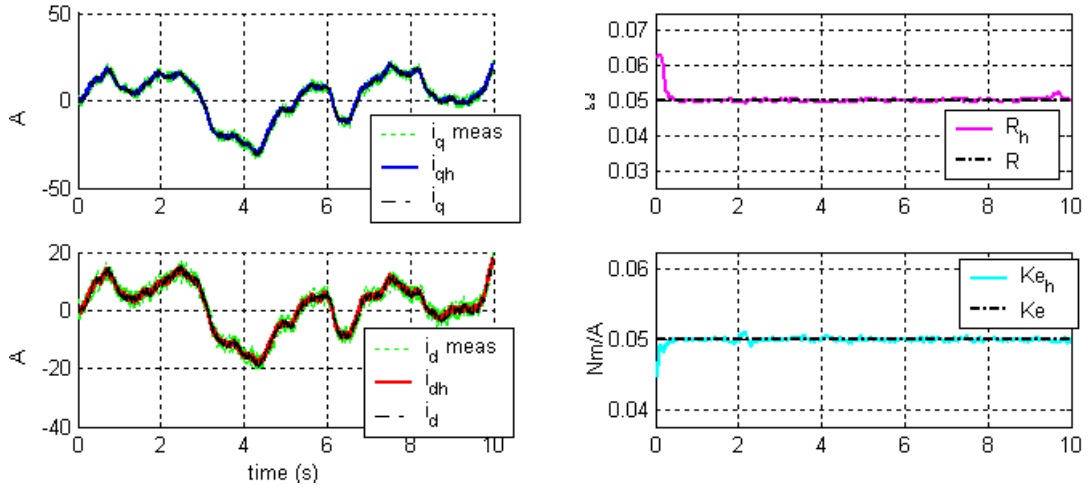


Figure 4.3 EKF estimation simulation: $R_h(0)=1.25R$, $K_{e_h}(0)=0.9K_e$

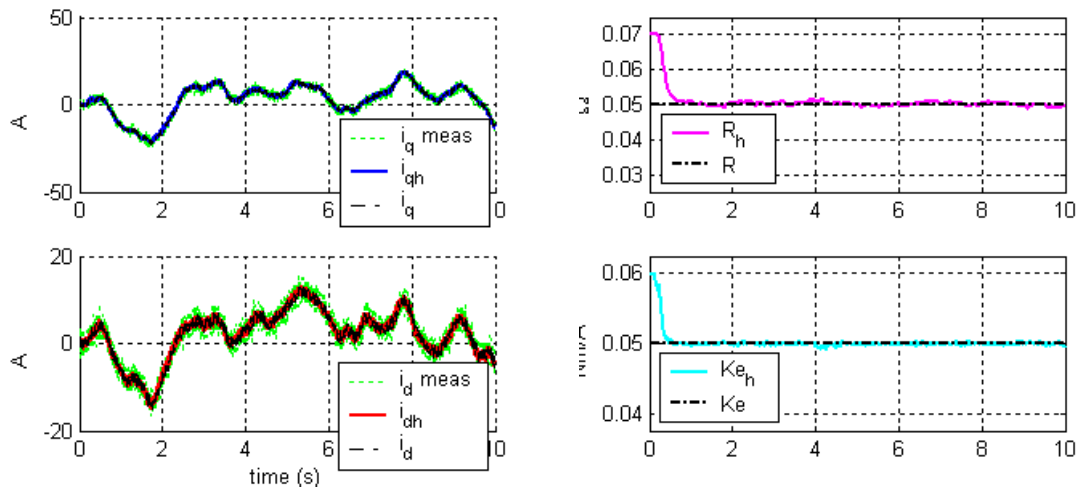


Figure 4.4 EKF estimation simulation: $R_h(0)=1.4R$, $K_{e_h}(0)=1.2K_e$

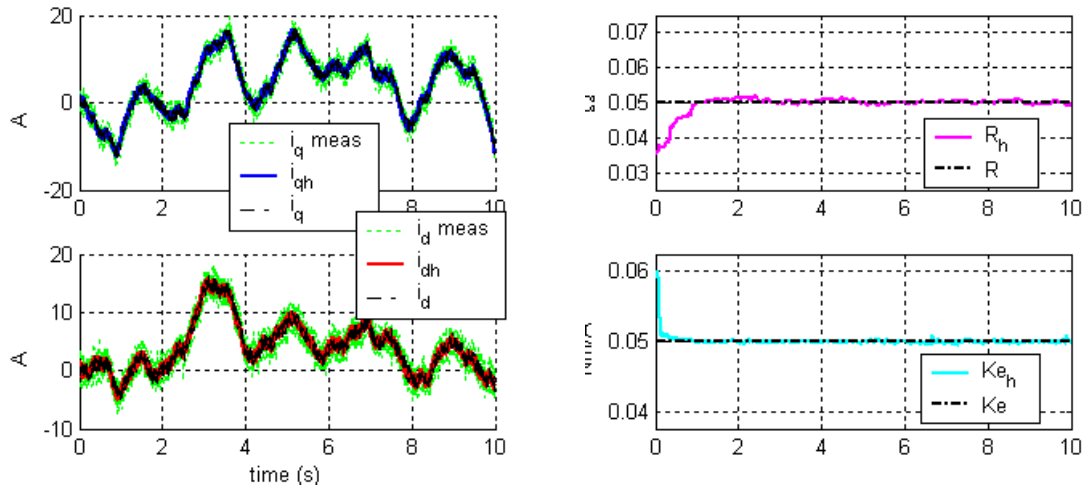


Figure 4.5 EKF estimation simulation: $R_h(0)=0.8R$, $K_{e_h}(0)=1.2K_e$

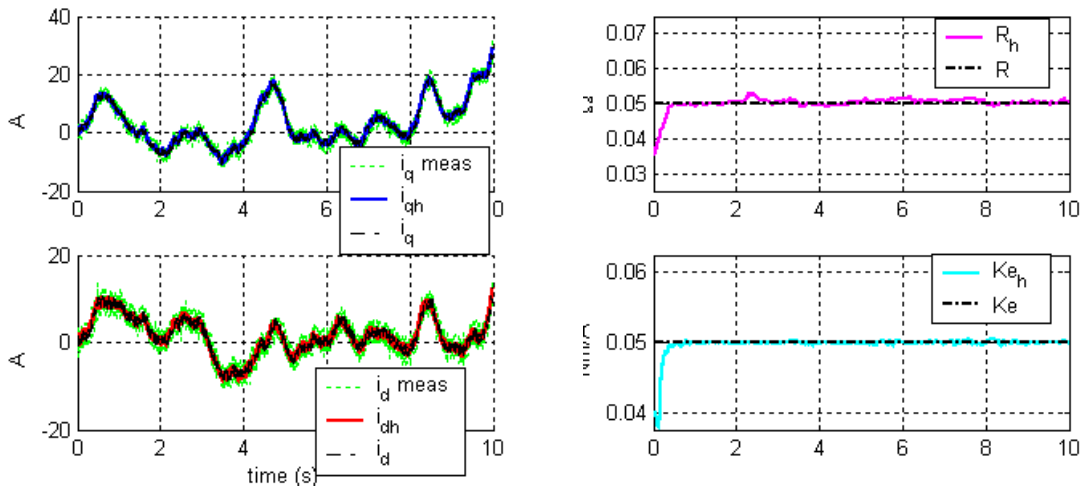


Figure 4.6 EKF estimation simulation: $R_h(0)=0.8R$, $K_{e_h}(0)=0.8K_e$

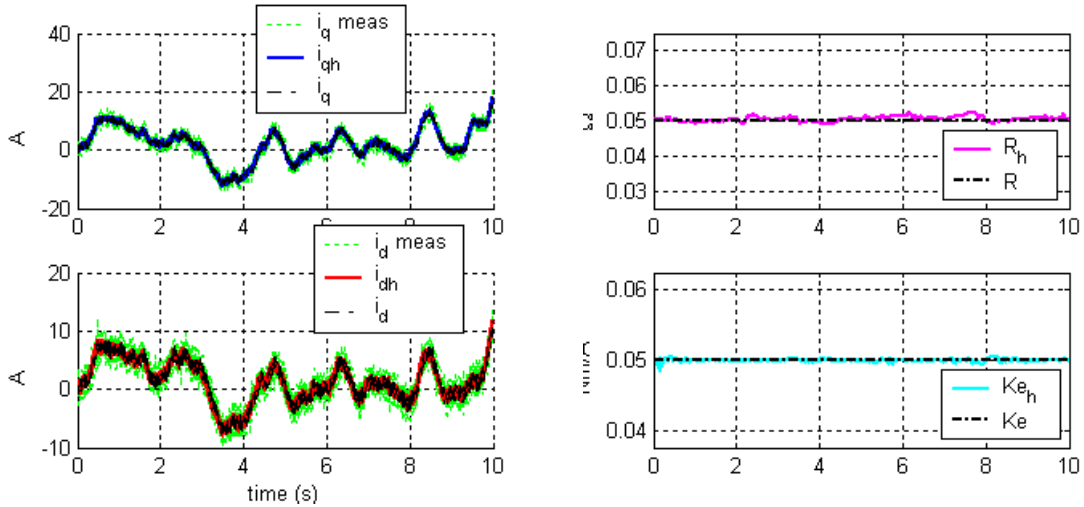


Figure 4.7 EKF estimation simulation: $R_h(0)=R$, $K_{e_h}(0)=K_e$

The EKF parameter identification algorithm demonstrated desirable performance in simulations. The estimated parameters converged to the reference value very quickly and stayed in a tight bound at steady state. From the performance perspective, the EKF is superior to the algorithms developed in Chapter 3. However, the EKF is computationally expensive due to the linearization computation in real time. It also has stability issues in simulation, and no mathematical proof of stability is available.

4.6 Summary

In the chapter, two recursive parameter estimation algorithms, the RLS and the EKF, were investigated for the BLDC motor parameter estimation. The RLS method, which is usually used for linear systems, was shown capable of finding a linear approximate model for the nonlinear BLDC motor dynamics. In simulations, when the motor was running at constant speed, the estimated parameters were close to their analytical values in the plant. But when the motor was running at varying speed, the RLS estimation results were not very meaningful for identifying the motor dynamics model. The EKF took advantage of

the knowledge of the nonlinear plant model and computed linearization functions for state prediction and covariance matrix in real time. As a result, it demonstrated superior performance in both parameter and state estimation with fast convergence and decent accuracy in BLDC motor application simulations. However, the EKF was computationally expensive due to the linearization and the covariance matrix computation in real time. In addition, the EKF, which engaged linearization of the nonlinear dynamics, might have stability issues in practice. In the BLDC motor simulations, the sampling interval had to be set to 0.5 milliseconds so as to avoid instability in the simulation. In summary, the EKF algorithm demonstrated highly desirable parameter identification performance for the BLDC motor application, but the computation cost and the stability issue were barriers for implementing the EKF in practice.

Chapter 5 MODEL REFERENCE ADAPTIVE CONTROL DESIGN

We have discussed several different parameter estimation methods for the BLDC motor application in Chapter 3 and Chapter 4. The EKF algorithm provided most desirable performance, but had some issues with computation cost and stability. The algorithms developed in Chapter 3 were simpler than the EKF and stable, but their parameter estimation performance lagged far behind the EKF. Ideally, we would like to have a parameter estimation algorithm with performance of the EKF and simplicity and stability of the algorithms in Chapter 3.

Let's re-examine the algorithms in Chapter 3 that solved the parameter error algebraically from the current errors. Firstly, these algorithms assumed steady state and neglected the current derivatives when deriving the parameter error equations. In addition, in order to maintain stability, these algorithms discarded some data points that might cause singularity in the parameter error calculation, and therefore discarded useful information hiding in these data points. If we can make full use of the discarded information for parameter estimation, it is highly possible that the performance will be improved.

The model reference adaptive control (MRAC) is a potential solution for the BLDC motor control problem with a good balance between computation complexity, stability and performance. Given a properly selected reference dynamics model, the MRAC can take advantage of the current derivatives in control and parameter estimation. Moreover, the MRAC is usually designed by finding the non-positive derivative of a Lyapunov cost function. Therefore its global stability is guaranteed, and all data points can be engaged in

the adaptation law. In this chapter, we will explore the MRAC application to the sinusoidal BLDC motors.

5.1 Model Reference Adaptive Control

Model reference adaptive control has been a well-developed approach of the adaptive control (Ioannou and Sun 1996, Tao 2003). A typical MRAC controller consists of a reference model, a control law, and an adaptive mechanism that updates the controller parameters by using the feedback error between the reference model and actual plant, as shown in Figure 5.1.

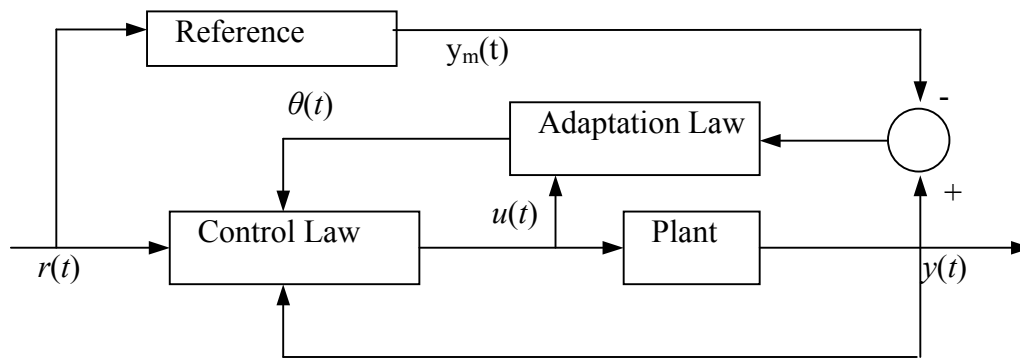


Figure 5.1 A schematic diagram of a typical MRAC controller

To design a MRAC, we need to find a suitable reference model for the closed loop system dynamics, and then we need to find a control law for the plant for achieve the primary control objective such as tracking the reference input $r(t)$. The controller may not be an ideal match for the plant if the parameters of the plant are unknown. The adaptation law will update the parameters in the control law so that the plant output tracks the

reference model output. In the next a few subsections, we will follow these steps to design a MRAC controller for the BLDC motor application.

5.1.1 The BLDC motor reference model

Recall that, as we found in Chapter 2, if the BLDC motor controller matches the plant ideally, the BLDC motor actuator will have a closed loop dynamics as that of (2.38). For convenience, the closed loop dynamics is restated here as

$$\begin{cases} L \frac{di_q}{dt} = -R i_q + R i_{q,cmd}, \\ L \frac{di_d}{dt} = -R i_d + R i_{d,cmd}. \end{cases} \quad (5.1)$$

The dynamics model in (5.1) was shown to be desirable for the BLDC actuator in Chapter 2. Therefore, it may be an ideal candidate reference model in the MRAC, except that the parameters are unknown. Let's substitute the parameters with their nominal value in controller, and rewrite the reference model as

$$\begin{pmatrix} \frac{di_{qm}}{dt} \\ \frac{di_{dm}}{dt} \end{pmatrix} = \begin{pmatrix} -\frac{1}{\tau_c} & 0 \\ 0 & -\frac{1}{\tau_c} \end{pmatrix} \begin{bmatrix} i_{qm} \\ i_{dm} \end{bmatrix} - \begin{pmatrix} i_{q,cmd} \\ i_{d,cmd} \end{pmatrix}, \quad (5.2)$$

where $\tau_c = \frac{L_c}{R_c}$ is the nominal time constant and $(i_{qm}, i_{dm})^T$ are the reference model state current. The reference model actually consists of two independent first order low pass filters, which are similar to that of a DC motor. If the closed loop BLDC motor dynamics tracks the reference model, the BLDC motor would have similar performance as that of a DC motor.

Let's compare the motor dynamics model

$$\begin{pmatrix} \frac{di_q}{dt} \\ \frac{di_d}{dt} \end{pmatrix} = \begin{pmatrix} -\frac{1}{\tau} & -\omega_e(t) \\ \omega_e(t) & -\frac{1}{\tau} \end{pmatrix} \begin{pmatrix} i_q \\ i_d \end{pmatrix} - \frac{1}{L} \begin{pmatrix} K_e \\ 0 \end{pmatrix} \omega_m(t) + \frac{1}{L} \begin{pmatrix} u_q \\ u_d \end{pmatrix} \quad (5.3)$$

to the reference model in (5.2). The nonlinear term $\omega_e(t)i_d$ and $\omega_e(t)i_q$, and the back EMF term $K_e\omega_m(t)$ are not presented in the reference model. To track the reference model, the control law must be designed to cancel these terms. In addition, the parameters in the motor dynamics model may be different from their nominal values used in the reference model. The adaptation law must be designed to identify the motor parameter values.

5.1.2 The control law

In order to track the reference model, let's rewrite the motor dynamics model as

$$\begin{pmatrix} \frac{di_q}{dt} \\ \frac{di_d}{dt} \end{pmatrix} = \begin{pmatrix} -\frac{1}{\tau} & 0 \\ 0 & -\frac{1}{\tau} \end{pmatrix} \begin{bmatrix} \begin{pmatrix} i_{qm} \\ i_{dm} \end{pmatrix} - \begin{pmatrix} i_{q,cmd} \\ i_{d,cmd} \end{pmatrix} \end{bmatrix} + \frac{1}{L} \cdot \begin{bmatrix} \begin{pmatrix} u_q \\ u_d \end{pmatrix} - \begin{pmatrix} K_e\omega_m + Ri_{q,cmd} + \omega_e Li_d \\ Ri_{d,cmd} - \omega_e Li_q \end{pmatrix} \end{bmatrix} \quad (5.4)$$

If the second term on the right hand side of (5.4) were zero, then the actual BLDC model would have the same form of the reference model. As the motor parameters are not known, their value in the controller will be used instead. Thus, the control law is proposed as

$$\begin{cases} u_q = K_{ec}\omega_m + R_c i_{q,cmd} + \omega_e L_c i_d, \\ u_d = R_c i_{d,cmd} - \omega_e L_c i_q. \end{cases} \quad (5.5)$$

where R_c , L_c , and K_{ec} are the parameter values in the controller.

If the controller parameters R_c , L_c , and K_{ec} match their respective counterparts in the motor plant, the controller law (5.5) will be able to drive the closed loop motor dynamics

to track the reference model perfectly. However, since the parameter values in the plant are not known exactly, we need to find some way to estimate their values.

5.1.3 The adaptation law

Let's consider using the feedback error between the reference model and the motor plant to estimate the plant parameter values. Define the reference current error vector as

$$\begin{pmatrix} e_q \\ e_d \end{pmatrix} = \begin{pmatrix} i_q \\ i_d \end{pmatrix} - \begin{pmatrix} i_{qm} \\ i_{dm} \end{pmatrix}. \quad (5.6)$$

We need to find out how the reference current errors are related to the parameter differences. Substituting (5.5) into (5.3), and subtracting the resulting equations by the reference model (5.2), the reference current error dynamics can be found as

$$\begin{aligned} \begin{pmatrix} \frac{de_q}{dt} \\ \frac{de_d}{dt} \end{pmatrix} &= \begin{pmatrix} -\frac{1}{\tau_c} & 0 \\ 0 & -\frac{1}{\tau_c} \end{pmatrix} \begin{pmatrix} e_q \\ e_d \end{pmatrix} + \begin{pmatrix} \frac{1}{\tau_c} - \frac{1}{\tau} & 0 \\ 0 & \frac{1}{\tau_c} - \frac{1}{\tau} \end{pmatrix} \begin{pmatrix} i_q \\ i_d \end{pmatrix} \\ &+ \omega_e \begin{pmatrix} 1 - \frac{L_c}{L} \\ 0 \end{pmatrix} \begin{pmatrix} -i_d \\ i_q \end{pmatrix} + \frac{1}{L_c} \begin{pmatrix} K_{ec} \\ 0 \end{pmatrix} \omega_m - \frac{1}{L} \begin{pmatrix} K_e \\ 0 \end{pmatrix} \omega_m. \end{aligned} \quad (5.7)$$

To simplify the notation in (5.7), we define parameter errors as

$$\tilde{k}_\tau = \begin{pmatrix} \frac{1}{\tau} - \frac{1}{\tau_c} \\ \frac{1}{\tau} - \frac{1}{\tau_c} \end{pmatrix}, \quad (5.8a)$$

$$\tilde{k}_e = \frac{K_e}{L} - \frac{K_{ec}}{L_c}, \quad (5.8b)$$

$$\tilde{k}_L = \frac{1}{L} (L - L_c). \quad (5.8c)$$

Substituting the parameter errors back into (5.7), the reference current error dynamics model becomes

$$\begin{pmatrix} \frac{de_q}{dt} \\ \frac{de_d}{dt} \end{pmatrix} = \begin{pmatrix} -\frac{1}{\tau_c} & 0 \\ 0 & -\frac{1}{\tau_c} \end{pmatrix} \begin{pmatrix} e_q \\ e_d \end{pmatrix} - \tilde{k}_\tau \begin{pmatrix} i_q \\ i_d \end{pmatrix} - \omega_e(t) \tilde{k}_L \begin{pmatrix} i_d \\ -i_q \end{pmatrix} - \begin{pmatrix} \tilde{k}_e \\ 0 \end{pmatrix} \omega_m(t). \quad (5.9)$$

We want the closed loop motor dynamics to track the reference the reference model as close as possible. In another word, we want the reference current error to be as low as possible in magnitude. In addition, if the parameter values in controller match their equivalent in the plant motor, the control law can make the closed loop motor dynamics model exactly the same as that of the reference model. We also want the parameter errors to be as close to zero as possible. To find the minimal current errors and parameter errors, let's define a Lyapunov candidate function of the reference current errors and parameter errors as

$$Q(e_q, e_d, \tilde{k}_\tau, \tilde{k}_e, \tilde{k}_L) = \frac{1}{2} \left(e_q^2 + e_d^2 + \frac{1}{\gamma_1} \tilde{k}_\tau^2 + \frac{1}{\gamma_2} \tilde{k}_e^2 + \frac{1}{\gamma_3} \tilde{k}_L^2 \right), \quad (5.10)$$

where γ_1 , γ_2 , and γ_3 are positive constants.

Q is a non-negative quadratic function of the current errors and the parameter errors. We can get the minimal current errors and parameter errors by minimizing the value of Q . The minimum of Q can be achieved by driving its time domain derivative to be negative or zero. Differentiating (5.1) with respect to time, we have

$$\frac{dQ}{dt} = e_q \dot{e}_q + e_d \dot{e}_d + \frac{1}{\gamma_1} \tilde{k}_\tau \dot{\tilde{k}}_\tau + \frac{1}{\gamma_2} \tilde{k}_e \dot{\tilde{k}}_e + \frac{1}{\gamma_3} \tilde{k}_L \dot{\tilde{k}}_L. \quad (5.11)$$

Substituting the reference current error dynamics (5.7) into (5.11), the derivative of Q becomes

$$\begin{aligned}
\frac{dQ}{dt} &= -\frac{1}{\tau_c} e_q^2 - \frac{1}{\tau_c} e_d^2 \\
&+ e_q \left[-\tilde{k}_\tau i_q - \tilde{k}_e \omega_m - \omega_e(t) \tilde{k}_L i_d \right] + e_d \left[-\tilde{k}_\tau i_d + \omega_e(t) \tilde{k}_L i_q \right] + \frac{1}{\gamma_1} \tilde{k}_\tau \dot{\tilde{k}}_\tau + \frac{1}{\gamma_2} \tilde{k}_e \dot{\tilde{k}}_e + \frac{1}{\gamma_3} \tilde{k}_L \dot{\tilde{k}}_L,
\end{aligned} \tag{5.12}$$

The first two items on the right hand side of (5.12) are non-positive. If we can make the remaining items of (5.12) to be zero, then we can guarantee non-positive time domain derivative of the Lyapunov cost functional, which will then approach its minimum asymptotically. Re-organize (5.12) as

$$\begin{aligned}
\frac{dQ}{dt} &= -\frac{1}{\tau_c} e_q^2 - \frac{1}{\tau_c} e_d^2 \\
&+ \tilde{k}_\tau \left[\frac{1}{\gamma_1} \dot{\tilde{k}}_\tau - (e_q i_q + e_d i_d) \right] + \tilde{k}_e \left(\frac{1}{\gamma_2} \dot{\tilde{k}}_e - e_q \omega_m \right) + \frac{1}{\gamma_3} \left[\dot{\tilde{k}}_L - \omega_e(t) (-e_q i_d + e_d i_q) \right].
\end{aligned} \tag{5.13}$$

Propose the adaptation laws for the controller parameters as

$$\begin{cases} \dot{\tilde{k}}_\tau = \gamma_1 (e_q i_q + e_d i_d) \\ \dot{\tilde{k}}_e = \gamma_2 e_q \omega_m \\ \dot{\tilde{k}}_L = \gamma_3 \omega_e(t) (-e_q i_d + e_d i_q) \end{cases}. \tag{5.14}$$

Substitute the adaptation law (5.14) into (5.13), the time domain derivative of the Lyapunov cost function becomes non-positive as

$$\frac{dQ}{dt} = -\frac{1}{\tau_c} e_q^2 - \frac{1}{\tau_c} e_d^2 \leq 0. \tag{5.15}$$

This guarantees that the global asymptotic stability of the reference current tracking error dynamics model. The reference model is globally stable as it consists of two first order linear systems with poles $-\frac{1}{\tau_c}$ in the left half plane (LHP). Therefore the MRAC

controller proposed in (5.5) and (5.14) guarantees the global asymptotically stability of the closed loop BLDC motor dynamics.

The physical parameter values of the motor can be derived from the adaptation law (5.14). Assuming that R , L and K_e are constants, and substituting (5.8) into (5.14), we have

$$\dot{\tilde{k}}_r = \frac{d}{dt} \left(\frac{R}{L} - \frac{R_c}{L_c} \right) = -\frac{d}{dt} \left(\frac{R_c}{L_c} \right) = \gamma_1 (e_q i_q + e_d i_d), \quad (5.16a)$$

$$\dot{\tilde{k}}_e = \frac{d}{dt} \left(\frac{K_e}{L} - \frac{K_{ec}}{L_c} \right) = -\frac{d}{dt} \left(\frac{K_{ec}}{L_c} \right) = \gamma_2 e_q \omega_m, \quad (5.16b)$$

$$\dot{\tilde{k}}_L = \frac{d}{dt} \left(1 - \frac{L_c}{L} \right) = -\frac{1}{L} \frac{dL_c}{dt} = \gamma_3 \omega_e(t) (-e_q i_d + e_d i_q). \quad (5.16c)$$

The controller parameter estimators can be found as

$$\frac{d}{dt} \left(\frac{R_c}{L_c} \right) = -\gamma_1 (e_q i_q + e_d i_d), \quad (5.17a)$$

$$\frac{d}{dt} \left(\frac{K_{ec}}{L_c} \right) = -\gamma_2 e_q \omega_m, \quad (5.17b)$$

$$\frac{dL_c}{dt} = -\gamma_3 \omega_e(t) L (e_q i_d - e_d i_q). \quad (5.17c)$$

If the inductance L is assumed to be a constant and $L \approx L_c$, the adaptation scheme becomes

$$\frac{dR_c}{dt} = -\gamma_1 L_c (e_q i_q + e_d i_d), \quad (5.18)$$

$$\frac{dK_{ec}}{dt} = -\gamma_2 L_c e_q \omega_m. \quad (5.19)$$

5.2 Simulation Results

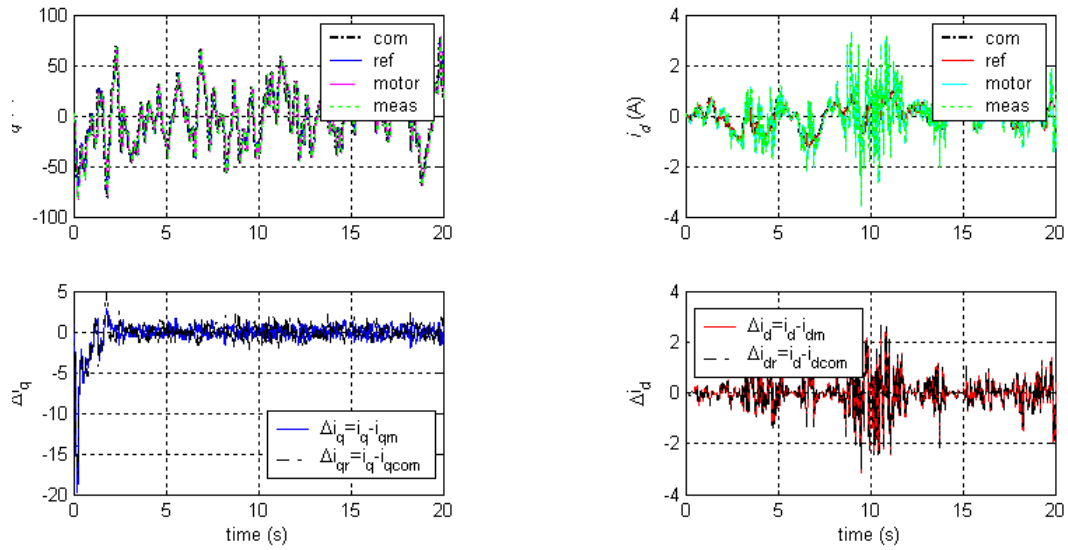
The MRAC controller with the control law of (5.5) and the adaptation law of (5.18~19) was implemented in Simulink® and was tested in simulation for the current tracking and parameter identification performance. The motor parameter values were as the following: $R = 0.05\Omega$, $K_e = 0.05 \text{ Nm/A}$, $L = 1 \times 10^{-4} \text{ h}$, $n_p = 3$. The adaptation gains in (5.18~19) were set as $\gamma_1 = \gamma_2 = 1$. The simulation sampling interval was set to 2 ms. The controller was assumed to have 50% error in R and 20% error in K_e initially. Figure 5.2 shows the reference current tracking simulation results. The reference current tracking error converged to zero within 2 seconds (approximately 1000 steps) as shown in Figure 5.2. Figure 5.3 and 5.4 are simulation results of parameter estimation with two typical cases of initial parameter errors. In both cases, the estimated controller parameters also converged to the their corresponding values within 2 seconds. At steady state, the mean and standard deviation of the parameter estimation errors were:

$$\mu(K_{ec} - K_e) = -2.4864 \times 10^{-5} = -0.0497\% K_e,$$

$$\mu(R_c - R) = 1.7544 \times 10^{-5} = 0.35\% R,$$

$$\|(K_{ec} - K_e)\|_2 = 0.0223,$$

$$\|(R_c - R)\|_2 = 0.0363.$$



(a) q -axis current tracking performance

(b) d -axis current tracking performance

Fig 5.2 MRAC reference current tracking performance in simulation: $R_c(0) = 0.5R$,

$$K_{e_c}(0) = 1.2K_e$$

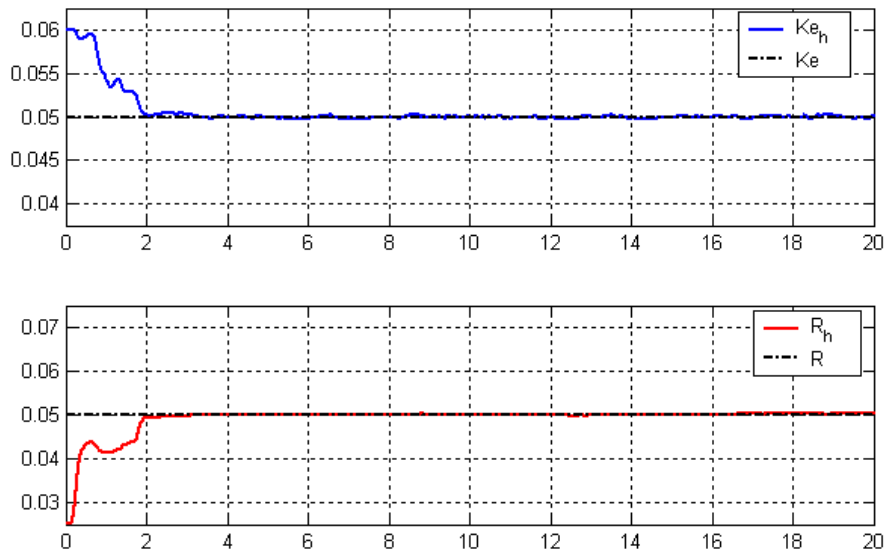


Fig 5.3 MRAC parameter estimation performance in simulation, case 1: $R_c(0) = 0.5R$,

$$K_{e_c}(0) = 1.2K_e$$

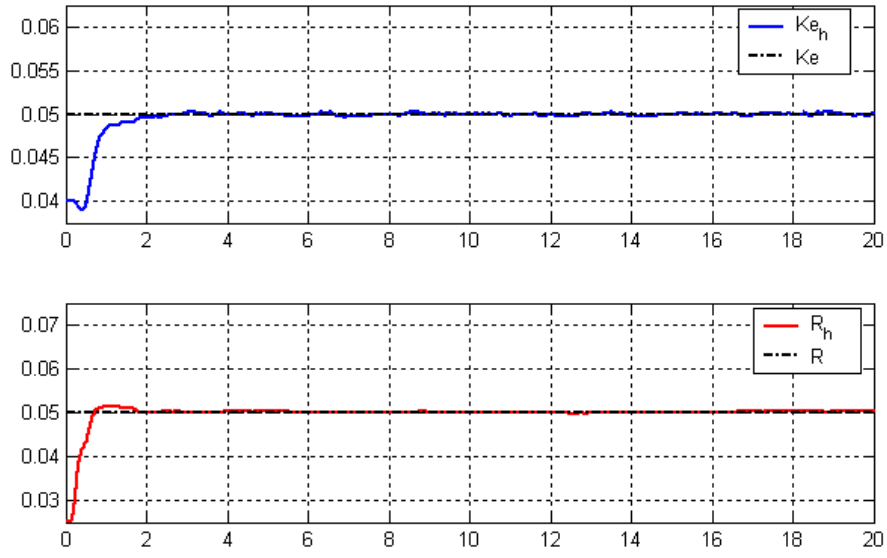


Fig 5.4 MRAC parameter estimation performance in simulation, case 2: $R_c(0) = 0.5R$, $Ke_c(0) = 0.8Ke$

5.3 Conclusion

A model reference adaptive control algorithm has been derived for the BLDC motor current control application in this chapter. The actuator dynamics model of equation (2.38) was selected as the reference model for the closed loop BLDC motor system. A feedback control law of equation (5.7) was proposed to achieve current tracking to the reference model outputs using the controller parameter values. To reduce tracking error introduced by parameter error between the controller and the plant, a Lyapunov cost function (5.10) was defined on the current tracking errors and the parameter errors between the plant and the controller. A controller adaptation law was designed by finding non-positive time-domain derivative of the quadratic Lyapunov function. The adaptation law guaranteed the global asymptotical stability of the MRAC algorithm. Comparing to

the algorithms developed in Chapter 3, the MRAC algorithm engaged all feedback data and derivative of the motor currents for parameter adaptation; therefore it demonstrated superior performance in simulations. Both the parameter estimation and the reference model current tracking converged quickly, and the steady state current tracking error was very low. Comparing to the EKF, the MRAC algorithm was much simpler and thus faster and required less computation resource.

Chapter 6 CLOSED LOOP SIMULATION AND CONTROLLER PERFORMANCE EVALUATION

Several different BLDC motor adaptive control algorithms have been developed in Chapter 3~5. All these algorithms were validated in the motor bench test simulations where the command torque τ_{cmd} and the motor velocity ω_m were assumed to be two independent random signals. In real applications, BLDC motors are usually the actuator of some outer loop systems; therefore the command torque τ_{cmd} and the motor velocity ω_m are partly correlated through the outer loop plant and controller. In this chapter, we will evaluate the adaptive control algorithms from the previous chapters in closed loop simulations with outer loop controller and plant models. The first closed loop system consists of the EPAS plant model together with an assisting torque controller, and the second is a motor speed control application with an anti-windup PI controller.

6.1 EPAS closed loop simulation

In the motor bench test simulations, the command torque τ_{cmd} and the motor velocity ω_m , which were the inputs to the motor controller, were assumed to be two independent random signals. However, in the application of the closed loop control of a system, ω_m and τ_{cmd} cannot be completely independent because the plant output ω_m is partly affected by plant input τ_{cmd} . In order to demonstrate the performance of the adaptive control algorithms in a closed loop setting, the example of an electric power assisted steering system (EPAS) is considered.

6.1.1 EPAS close loop system model

As shown in Figure 6.1, the closed loop automotive EPAS system consists of an electric control unit (ECU), a brushless DC motor, a torque sensor, a steering hand wheel, and a set of rack and pinion. The motor torque is transferred to the steering column via a worm and worm gear assembly. The torque transducer measures the torque on the steering column (τ_s). The motor velocity ω_m is measured by a tachometer. These signals along with motor position are collected by the ECU, which generates τ_{cmd} accordingly and calculates the voltages to be applied to the motor.

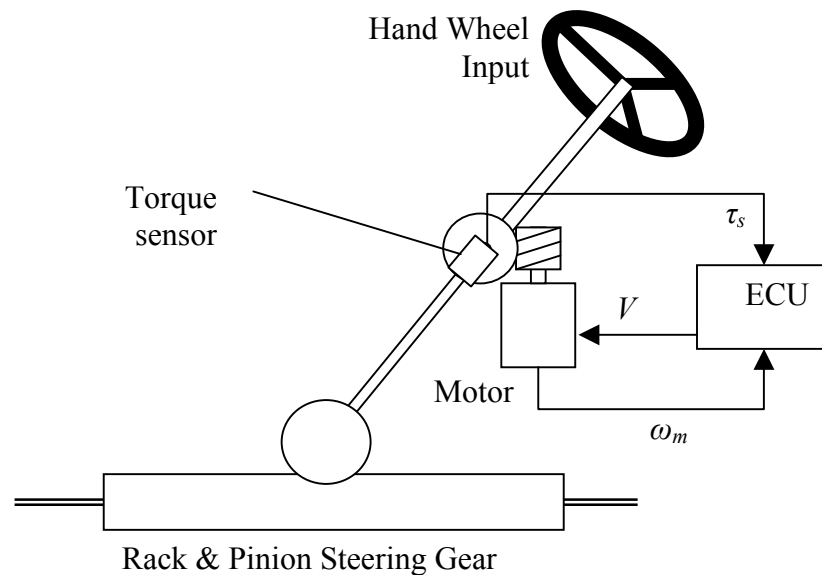


Figure 6.1 Schematic diagram of a typical EPAS.

The steering system was modeled as a two-mass mechanical system with viscous friction (Badawy et al. 1999). It was assumed that the rotor shaft of the BLDC motor was rigidly connected to the steering column. A Simulink model of the EPAS was used for

the closed loop simulation. In this model, the driver hand-wheel torque input was modeled as a Gaussian random signal passed through an anti-aliasing filter, which simulated typical human driver frequency response characteristics. In addition, two independent Gaussian random signals were implemented in the model to simulate the road resistance and other perturbation torque on the steering column. In the simulations, the same set of random inputs, including the hand-wheel torque, the road resistance and other perturbation torque, were fed into the EPAS plant model for each of the adaptive BLDC control algorithms in order to make their current tracking and parameter estimation performance comparable.

The ECU is embedded with the assisting torque controller and the BLDC motor controller. The former is required in the EPAS to achieve the desirable assisting torque and steering wheel feedback for the driver. The latter will be the adaptive controller developed in this research. The assisting torque controller developed by Patankar (Patankar 2003) was adopted in the close loop simulations.

We will compare four type of adaptive BLDC motor controller in the closed loop simulations. The extended Kalman filter (EKF) in section 4.4 was selected as the benchmark parameter estimation algorithm. The current controller of equation (2.37) was chosen as the motor controller to form a closed loop adaptive controller using the EKF, and the controller parameters were updated with the EKF estimated values. The Gram-Schmidt orthonormalization algorithm in section 3.5 showed the best performance among the algorithms that solved the parameter errors algebraically from the q axis current dynamics alone. We denote this class of algorithms as the q -solver algorithms in the simulations. The Gram-Schmidt orthonormalization algorithm was selected as the

representative for the q -solver algorithms for the performance comparison. The third adaptive algorithm included in the evaluation was the one developed in section 3.6 that solved the parameter algebraically from both the q and the d axis current dynamics. It is named as the qd -solver here. The MRAC algorithm developed in Chapter 5 was the fourth algorithm for performance comparison in closed loop simulations. Notice that the EKF had to run at simulation step length of 0.5 milliseconds for stability in simulation, while the simulation step length was 2 milliseconds for the other three algorithms. The EKF was expected to be approximately 4 times faster than other algorithms in term of the parameter estimation convergence speed given the same conditions.

The BLDC motor model of equation (2.38) was used as the motor plant in the closed loop simulation. The parameter values in the BLDC motor plant model were assumed to be constant as the following: $R = 0.05\Omega$, $K_e = 0.05 Nm/A$, $L = 1 \times 10^{-4} h$, $n_p = 3$.

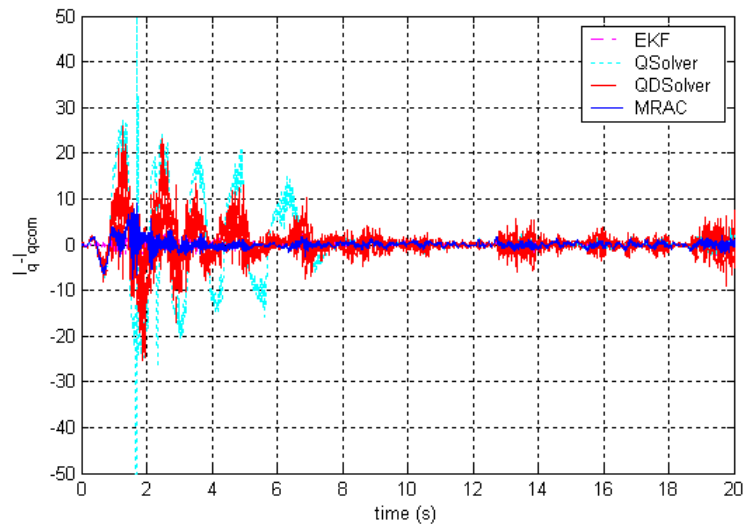
6.1.2 EPAS simulation results and analysis

In some situations, the effect of parameter error in R and K_e may cancel each other. For example, positive ΔR and negative ΔK_e may end up with the same control voltage v_q as that of zero parameter errors. In addition, the assisting torque controller may have different responses to positive and negative parameter errors. It may be harder for the parameter estimation algorithms in some initial parameter errors combinations. Therefore, several different combinations of initial controller parameter errors in R and K_e were simulated. For each set of initial parameter errors, the current tracking error and parameter estimation results from the four algorithms are overlapped in two figures respectively for comparison of the converging speed and steady state parameter

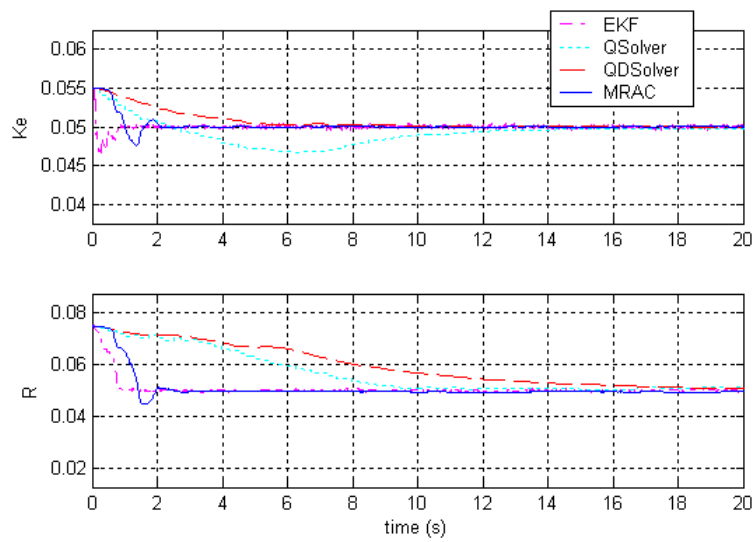
estimation error. Figure 6.2 shows the simulation results of $\Delta R(0) = +50\%R$ and $\Delta K_e(0) = +10\%K_e$. Figure 6.3 shows the simulation results of $\Delta R(0) = +50\%R$ and $\Delta K_e(0) = -15\%K_e$. Figure 6.4 shows the simulation results of $\Delta R(0) = -50\%R$ and $\Delta K_e(0) = -15\%K_e$. Figure 6.5 shows the simulation results of $\Delta R(0) = +50\%R$ and $\Delta K_e(0) = +10\%K_e$. Table 6.1 compares the steady state mean and standard deviation of the current and parameter estimation error. In all simulations, the parameter estimation and current tracking are stable and converge to steady state values. Positive initial error in K_e ($\Delta K_e(0) = +10\%K_e$) appears to be harder than negative initial error $\Delta K_e(0)$ for both current tracking and parameter estimation. The reason was that the positive ΔK_e would result in higher motor control voltages, and therefore higher motor torque than with negative ΔK_e at high motor velocity. Consequently, the assisting torque controller would respond more to high motor velocity, and the command torque might be slightly more correlated to the motor velocity.

Comparing the current tracking error and parameter estimation results in all simulations, the EKF demonstrated the best parameter estimation and the current tracking performance among the four algorithms. This is reasonable as the EKF is optimal if the system is at an equilibrium state with Gaussian noises. The Kalman gains are updated online with the motor states based on the sensitivity of output currents to the parameters values. In addition, the EKF is an observer that takes the control signals (V_d, V_q) and motor states (i_d, i_q) as inputs. It is relatively independent to the outer loop controller. The sampling step length of 0.5 ms also contributes to the faster convergence of EKF, as the other three algorithms were running at 2ms.

The MRAC had the lowest standard deviation of the current tracking error at steady state. This was due to the fact that the adaptive controller was designed to drive the current tracking error to zero. The slightly higher mean of current tracking error $\Delta i_q = i_q - i_{q,cmd}$ in the simulations was because that the MRAC current tracking errors were defined as the difference between the reference model outputs and motor currents. The reference model dynamics introduced some extra error into the command current tracking. Considering the factor of slower sampling speed, the MRAC showed comparable convergence speed in parameter estimation as that of the EKF. The fast convergence speed resulted from the fact that all data were used for the parameter estimation. However, the MRAC showed some steady state parameter estimation error in R . In this algorithm, the parameter estimation was just an intermediate means for achieving reference current tracking. Thus the estimation computation required rich frequency contents of the excitation signals, including $i_{q,cmd}$, $i_{d,cmd}$ and ω_m , to achieve ideal parameter estimation. One possible reason was that the outer loop controller was designed to provide some certain frequency characteristics for driver hand-wheel feedback. Thus the frequency contents of $i_{q,cmd}$ and ω_m were relatively limited. $i_{d,cmd}$ had rich frequency contents as it was set as a Gaussian random signal in the simulations, but the magnitude of $i_{d,cmd}$ was significantly lower than that of $i_{q,cmd}$ because of the operating principle of the BLDC motor.

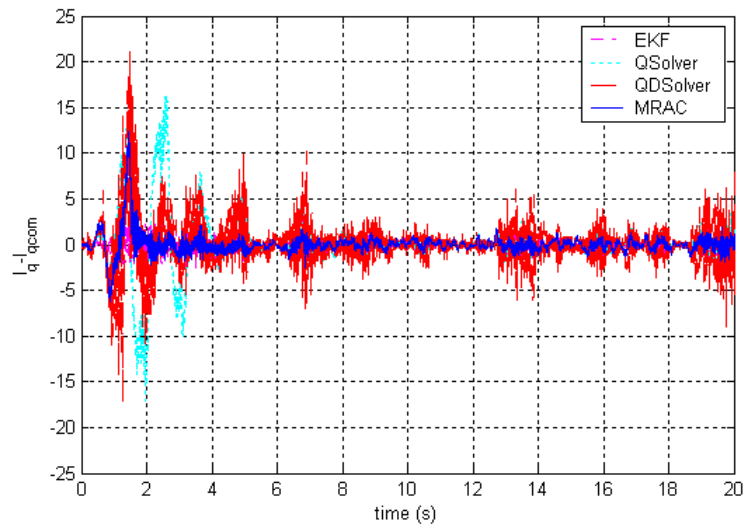


(a) Current tracking results

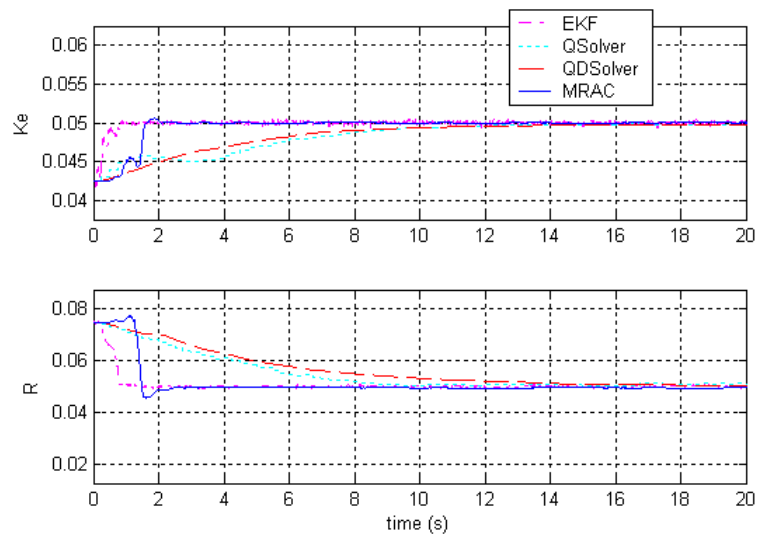


(b) Parameter estimation results

Figure 6.2 Comparison of the adaptive BLDC motor controllers in EPAS closed loop simulation: $\Delta R(0) = +50\%R$ and $\Delta K_e(0) = +10\%K_e$

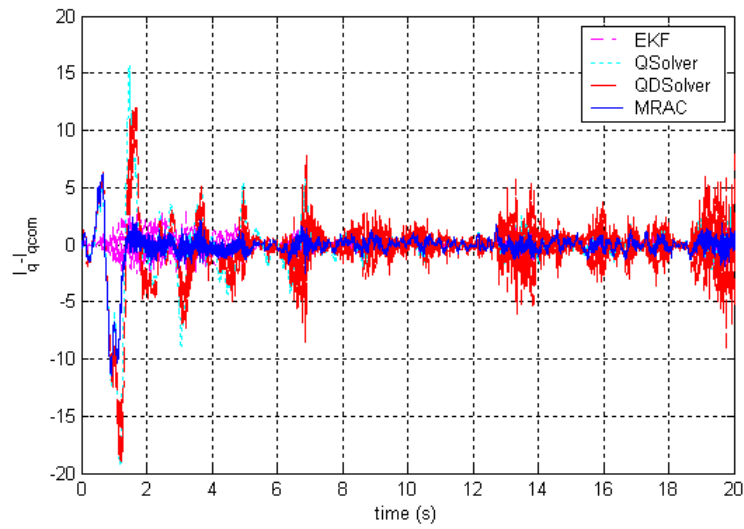


(a) Current tracking results

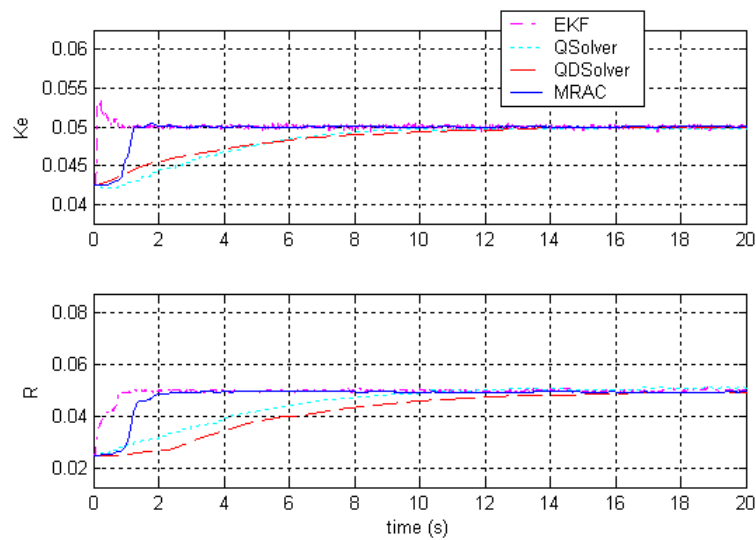


(b) Parameter estimation results

Figure 6.3 Comparison of the adaptive BLDC motor controllers in EPAS closed loop simulation: $\Delta R(0) = +50\%R$ and $\Delta K_e(0) = -15\%K_e$

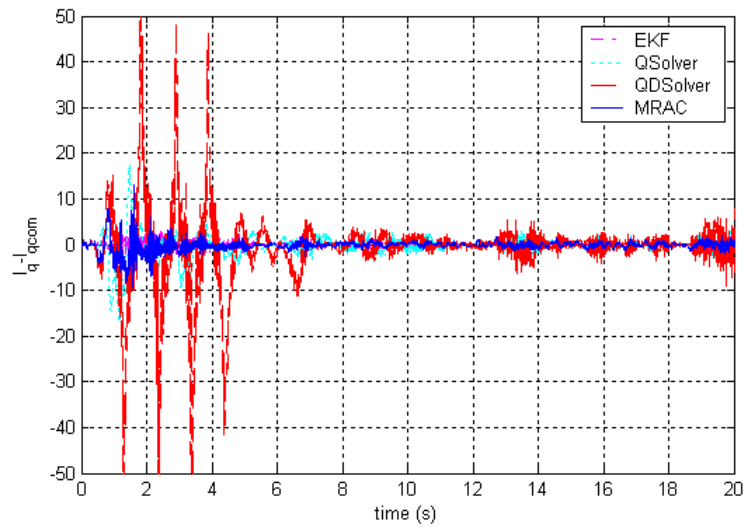


(a) Current tracking results

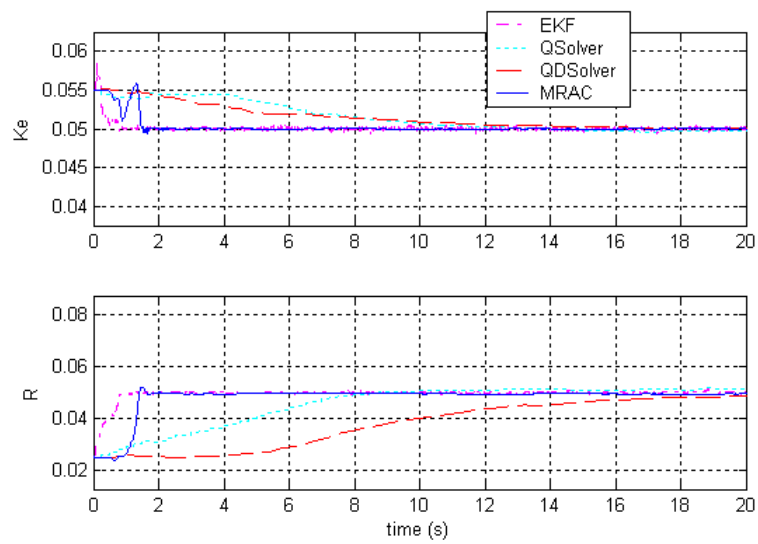


(b) Parameter estimation results

Figure 6.4 Comparison of the adaptive BLDC motor controllers in EPAS closed loop simulation: $\Delta R(0) = -50\%R$ and $\Delta K_e(0) = -15\%K_e$



(a) Current tracking results



(b) Parameter estimation results

Figure 6.5 Comparison of the adaptive BLDC motor controllers in EPAS closed loop simulation: $\Delta R(0) = +50\%R$ and $\Delta K_e(0) = +10\%K_e$

Table 6.1 Steady-state performances in EPAS closed loop simulation

Case	Algorithm	$\mu(\Delta K_e)$	$std(\Delta K_e)$	$\mu(\Delta R)$	$std(\Delta R)$	$\mu(\Delta I_q)$	$std(\Delta I_q)$
1	EKF	2.71×10^{-5}	9.6×10^{-5}	-1.23×10^{-4}	2.22×10^{-4}	-0.032	0.590
2	q -solver	-2.41×10^{-4}	3.57×10^{-5}	8.06×10^{-4}	3.21×10^{-4}	0.0438	0.743
3	qd -Solver	-8.15×10^{-6}	6.39×10^{-5}	-0.0014	1×10^{-3}	-0.0397	1.344
4	MRAC	2.11×10^{-6}	9.59×10^{-5}	-5.37×10^{-4}	3.23×10^{-4}	-0.173	0.492

The **qd**-solver and the **q**-solver were based on similar principle of parameter error estimation. Their parameter estimation results were slower and smoother than that of the EKF and the MRAC. The **qd**-solver and the **q**-solver were similar to the MRAC in the sense of using the correlation of current error and current for parameter estimation. Actually, the **qd**-solver and the **q**-solver algorithms could be considered as a special case of model reference control in which the reference model would be a unit gain. However, in both algorithms, some data points were discarded during the parameter error estimation to avoid noise from singularity (especially matrices inversion). This procedure helped to maintain stability, but ruled out useful data and therefore reduced parameter estimation convergence speed.

The **qd**-solver had better estimation performance in K_e than the **q**-solver regardless of initial parameter errors. On the other hand, the **qd**-solver did not perform as well in the R estimation. It showed similar steady state error in R estimation as that of the MRAC. The main reason was that the **qd**-solver estimated R mainly on the i_d dynamics and K_e on the i_q dynamics with both i_d and i_q feedback in every loop, while the **q**-solver estimated both parameters on the i_q dynamics and needed i_q measurement from two or more loops.

The *qd*-solver had more data points than the *q*-solver, so it performed better in K_e estimation. On the other hand, the magnitude of i_d was significantly lower than that of i_q , which resulted in slightly slower R estimation of the *qd*-solver.

Another factor for the algorithms comparison is the computation costs. Among the four algorithms, the EKF demands most computing power for two reasons: it requires shorter sampling step length to achieve stability of the Extended Kalman Filter; the linearization calculation and the sixteen variance states in the EKF needs significantly higher amount of computation in each step. The orthonormalization in the *q*-solver requires storage of some intermediate calculation results online for a certain period (250ms in the simulation). Therefore, the *q*-solver algorithm requires most random access memory (RAM) in implementation. The approximate inverse motor dynamics and the inner product calculation in the *q*-solver consumed the second highest computation power. The MRAC and the *qd*-solver have the lowest requirements on the computation resource, as they need least historic data and their parameter error calculations are relatively simpler than the EKF and the *q*-solver.

In summary, the EKF provided the most desirable parameter estimation performance in the closed loop EPS simulation at the cost of high computation power. Its stability is not guaranteed. The *q*-solver had the slowest parameter estimation convergence speed, but the steady state accuracy was close to that of the EKF. Its computation costs were second highest due to the orthonormalization and the inverse motor dynamics approximation calculation. The *qd*-solver required less computation resource, but its parameter estimation performance in R was the lowest among the four algorithms. The MRAC had the second highest parameter estimation convergence speed though the EKF was running

at four times faster. It also had the tightest standard deviation in current tracking error. It was economic on computation. However, due to the characteristics of the outer loop controller, the MRAC showed steady state parameter estimation error, as it was an intermediate means to achieve low current tracking error.

The outer loop controller, in the case the assisting torque controller, appeared to have some impact on the performance of the adaptive BLDC motor controllers. From the motor controller point of view, the outer loop controller and the EPAS plant together formed a feedback loop from the motor output torque/velocity to the command torque. Obviously, the performance of the motor controller would subject to the overall system feedback. The outer loop controller had some specific frequency response design, and narrowed the frequency contents of the excitation signals to the adaptive controllers. As a result, the performance of the adaptive controllers reduced slightly when compared to those motor bench test simulations. To further study the outer loop controller's impact on the performance of the adaptive motor controllers, a closed loop motor speed control application will be investigated in the next section.

6.2 Closed Loop Simulation Of a Motor Speed Control Application

We have discussed the performance of the adaptive BLDC motor controllers in the EPAS closed loop simulations. The adaptive control algorithms suffered slightly due to the specific frequency response of the assisting torque controller. In this section, we will see how the adaptive algorithms performed in the closed loop simulations of a typical speed control application. In this case, the outer loop speed controller has significantly different frequency response than that of the EPAS assisting torque controller.

In a typical motor angular speed/position control application, the outer loop plant can be modeled as a second order motor shaft dynamics as shown in Figure 6.6. The load torque on the motor shaft is assumed to be a band-limited Gaussian random signal. To make sure the simulation results are comparable, a random load torque time history was created offline and then was used in the closed loop simulations of all motor control algorithms.

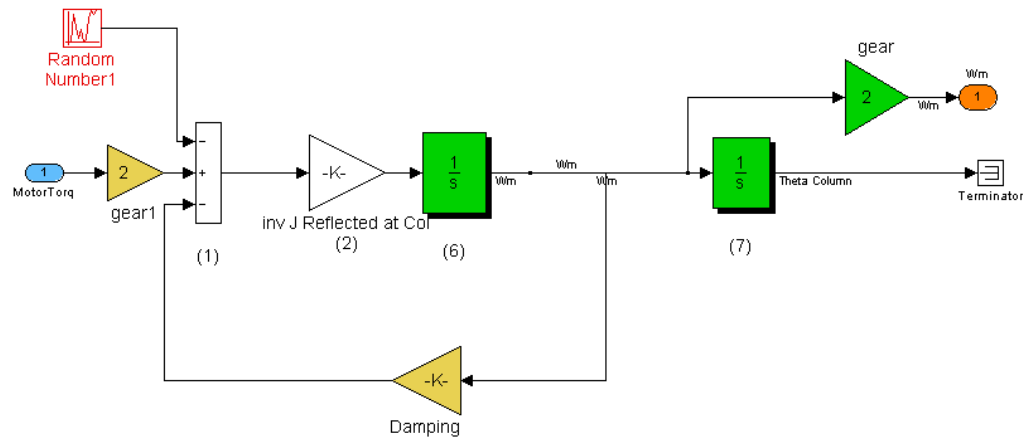


Figure 6.6 The block diagram of the motor shaft dynamics Simulink model.

The outer loop controller is an anti-windup PI controller with a reference speed input, as shown in Figure 6.7. For performance comparison purpose, we use the same BLDC motor ($R = 0.05\Omega$, $K_e = 0.05 Nm/A$, $L = 1 \times 10^{-4} h$, $n_p = 3$) in the speed control simulations. The closed loop simulation results for the EPAS system showed that the initial parameter errors had some effects on the controller performances. Thus, we run the closed loop simulation with different combinations of initial parameter errors for each of

the adaptive control algorithms. The closed loop simulation results are compared in Figure 6.8 ~ 6.11, and the corresponding steady state current tracking and parameter estimation results are compared in Table 6. 2.

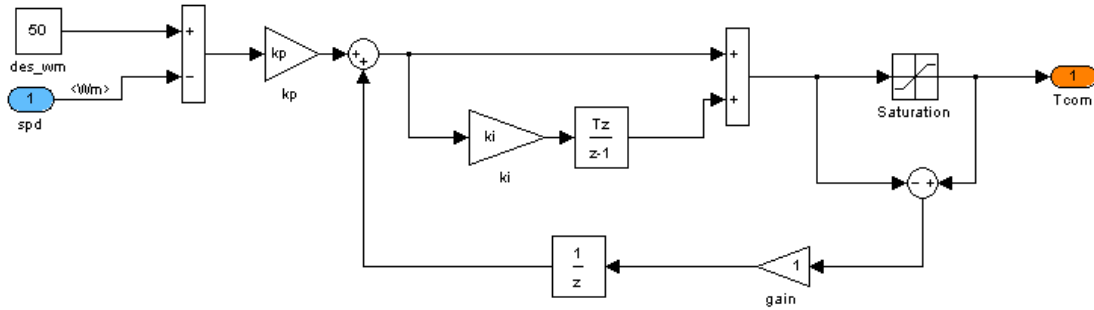


Figure 6.7 The block diagram of the anti-windup PI controller Simulink model.

Table 6.2 Steady-state performances in speed control closed loop simulation

Case	Algorithm	$\mu(\Delta Ke)$	$std(\Delta Ke)$	$\mu(\Delta R)$	$std(\Delta R)$	$\mu(\Delta I_q)$	$std(\Delta I_q)$
1	EKF	5.25×10^{-5}	10.8×10^{-5}	-3.0×10^{-4}	3.07×10^{-4}	-0.022	0.125
2	<i>q</i> -solver	-6.61×10^{-5}	2.99×10^{-5}	-8.5×10^{-4}	1.50×10^{-4}	-0.015	0.059
3	<i>qd</i> -solver	1.54×10^{-4}	1.23×10^{-4}	-9.2×10^{-4}	4.31×10^{-4}	0.044	0.300
4	MRAC	-4.31×10^{-5}	7.88×10^{-5}	2.89×10^{-4}	7.69×10^{-5}	-0.017	0.23

In general, the motor speed control closed loop simulation showed similar results as that of the EPAS closed loop simulation. The EKF had the fastest parameter estimation convergence and the fastest current tracking. The MRAC was the second fastest in terms of parameter estimation and current tracking. The *qd*-solver showed slightly faster parameter convergence than the *q*-solver, while the *q*-solver had the tightest standard

deviation in parameter estimation errors at steady state. The *qd*-solver had the highest parameter estimation error standard deviation especially in R , which was because of the relatively lower magnitude of i_d . Overall, the MRAC was the best combination of parameter estimation and reference current tracking convergence speed, steady state tracking error, and computation cost.

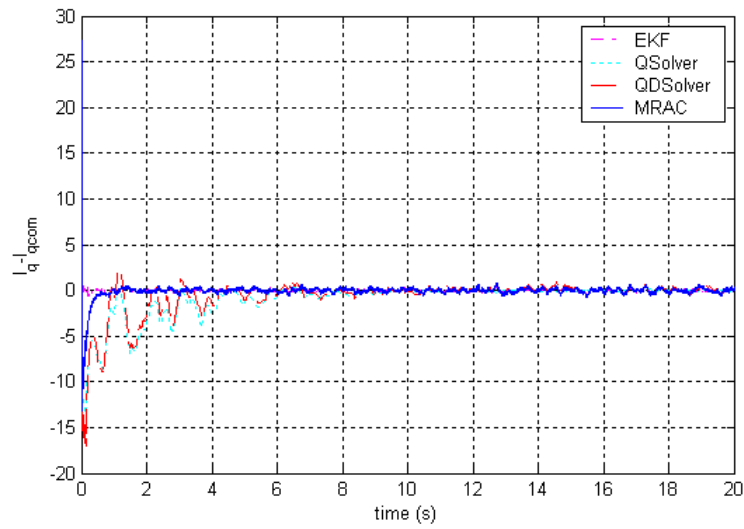
However, the speed control closed loop simulation revealed some different dynamics in the interaction between the adaptive controller and the outer loop controller. Unlike in the EPAS closed loop simulations, none of the adaptive controller showed steady state parameter estimation error in the motor speed control simulations. This was due to the wider frequency contents of the anti-windup PI controller than that of the assisting torque controller in the EPAS system. In addition, the initial parameter errors seemed to have more effects on the parameter estimation, especially for the MRAC. Again, this was because that the anti-windup PI controller responded differently to the current error introduced by the initial parameter error. The reference speed was set to be a constant (50rad/s), thus the ω_m was positive in the simulation. Re-examining the parameter estimation equations (5.18~5.19), the parameter estimation would be mainly determined by the current tracking error. When the initial parameter errors $\Delta R(0)$ and $\Delta K_e(0)$ had different sign, they cancelled out each other to some extent when calculating the control voltages. The outer loop anti-windup PI controller integrated the speed error, thus delayed the effect of current error introduced by opposite sign of $\Delta R(0)$ and $\Delta K_e(0)$. The ended up with slower estimation of parameter error, and consequently the current tracking was also slower when the $\Delta R(0)$ and the $\Delta K_e(0)$ had the same sign. Among the

four adaptive control algorithms, the EKF was the least sensitive one to the outer-loop controller's response to the initial parameter errors. This might be due to its observer structure and the control voltages were part of its input signals to the parameter estimation functions.

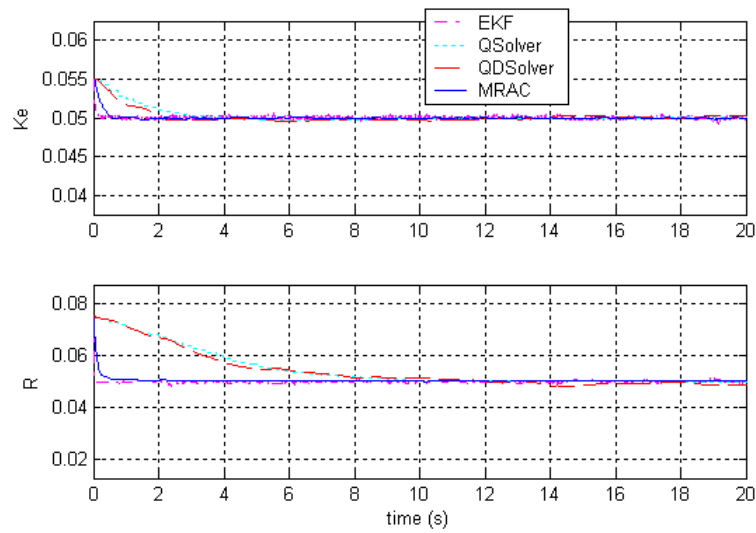
In summary, the motor speed control closed loop simulation showed similar conclusion to the EPAS simulation in terms of the adaptive control algorithms comparison. The MRAC algorithm appeared to be the best combination of performance and computation costs. However, the performance of the MRAC algorithm was slightly affected by the outer loop controller's characteristics.

6.3 Conclusion

Four representative adaptive BLDC motor control algorithms from Chapter 3~5, namely the EKF, the q -solver, the qd -solver, and the MRAC, were compared in closed loop simulation of an EPAS system and a speed control application. Generally, the simulation results revealed a trade-off between the performance and algorithm complexity (or computation costs). Better performance was obtained at the cost of higher complexity of the control algorithm. Moreover, the simulation results indicated that the outer loop controller and plant dynamics would reduce the performance of the adaptive controllers if the outer loop controller had relatively narrow frequency response or if the outer loop system dynamics responded differently to the different parameter error combinations.

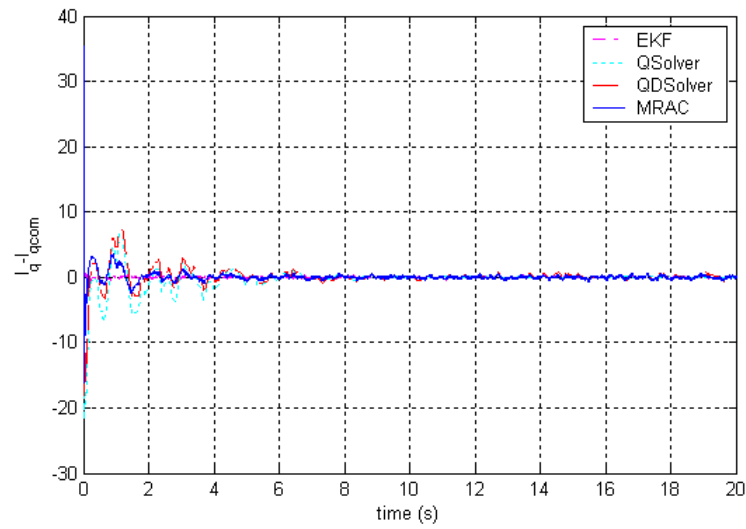


(a) Current tracking results

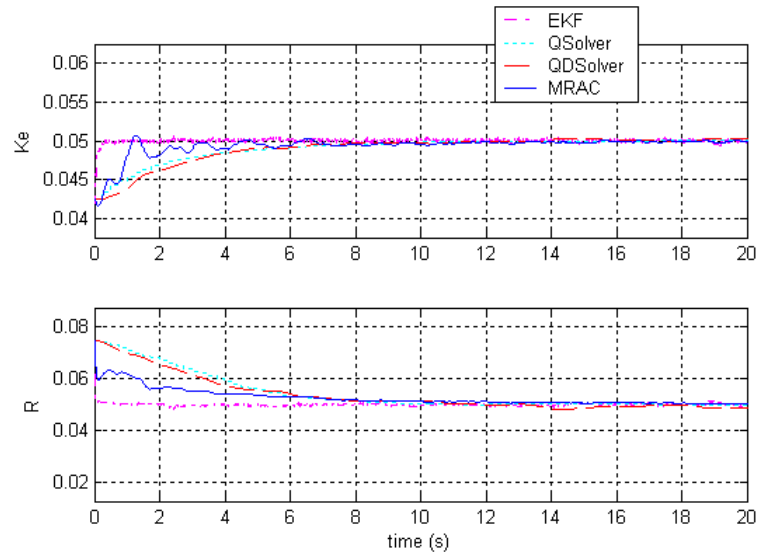


(b) Parameter estimation results

Figure 6.8 Comparison of the adaptive BLDC motor controllers in the motor speed control closed loop simulation: $\Delta R(0) = +50\%R$ and $\Delta K_e(0) = +10\%K_e$

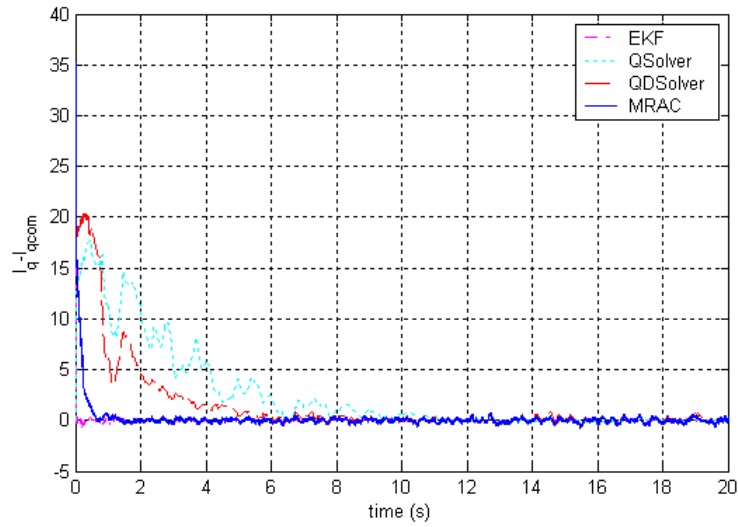


(a) Current tracking results

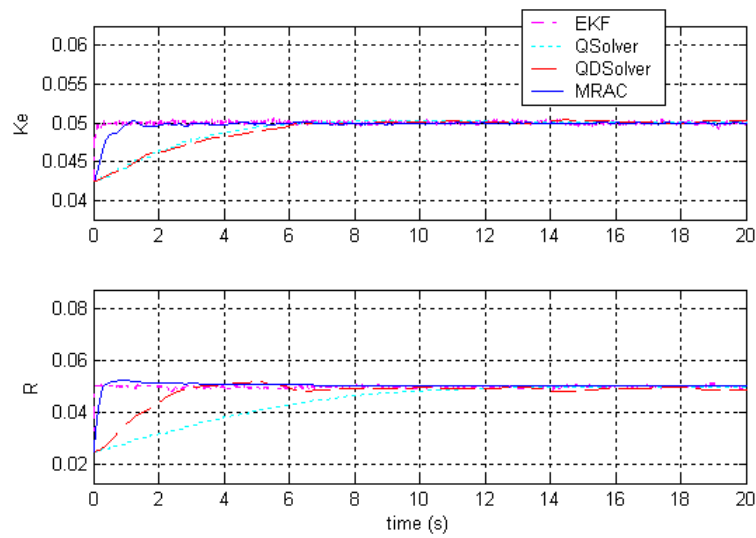


(b) Parameter estimation results

Figure 6.9 Comparison of the adaptive BLDC motor controllers in the motor speed control closed loop simulation: $\Delta R(0) = +50\%R$ and $\Delta K_e(0) = -15\%K_e$

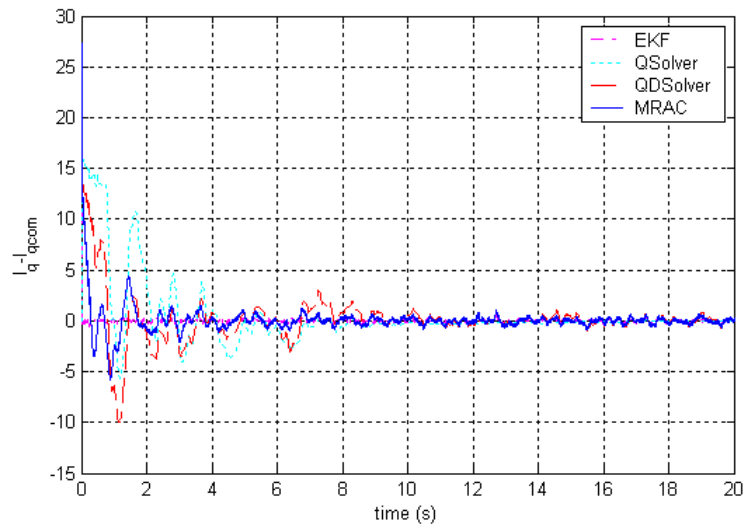


(a) Current tracking results

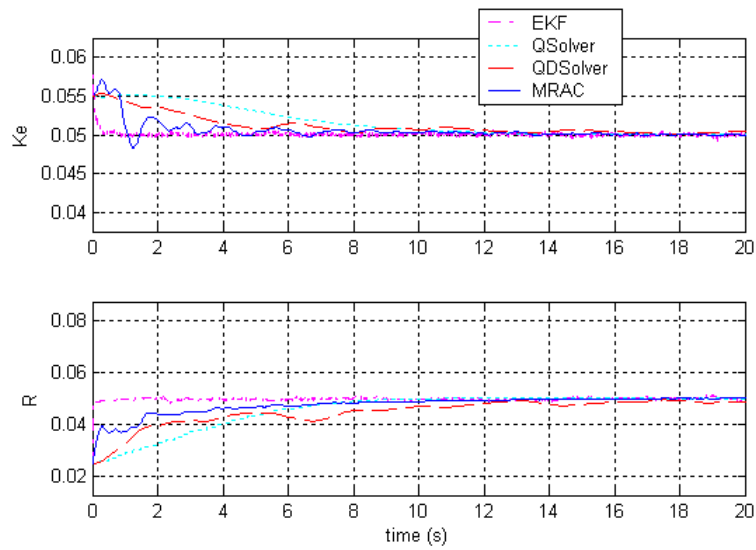


(b) Parameter estimation results

Figure 6.10 Comparison of the adaptive BLDC motor controllers in the motor speed control closed loop simulation: $\Delta R(0) = -50\%R$ and $\Delta K_e(0) = -15\%K_e$



(a) Current tracking results



(b) Parameter estimation results

Figure 6.11 Comparison of the adaptive BLDC motor controllers in the motor speed control closed loop simulation: $\Delta R(0) = -50\%R$ and $\Delta K_e(0) = +10\%K_e$

Among the four algorithms, the EKF showed the best parameter estimation performance in the closed loop simulations. It was also the least sensitive one to the reference signals from the outer loop system. These advantages of the EKF were due to the real time linearization of the nonlinear model and the optimal Kalman filtering based on the linearized model. On the other hand, the desirable performance of the EKF came at the highest computing cost. With the EKF algorithm, the controller needed to compute the linearization function and to update a 4x4 covariance matrix for the 4 augmented states at every sample. In addition, the stability of the EKF was not guaranteed due to the online linearization. In the closed loop simulation, it required about 4 times higher sampling frequency than other algorithms did so as to maintain stability.

The q -solver and the qd -solver were based on the same principle of solving the parameter error from the current feedback algebraically. The q -solver showed the slowest parameter estimation convergence in the closed loop simulations. Computation of the orthonormalization and approximation of the inverse motor dynamics consumed second highest computation power and memory. The qd -solver engaged both i_q and i_d for parameter error calculation, and therefore had faster estimation than the q -solver. Besides, it required less computation resource as it used more data for the estimation. However, due to the operation principle of the BLDC motor, the d -axis current is usually of significant lower magnitude than the q -axis current. Consequently, the qd -solver had the lowest estimation performance in R among the four algorithms.

The MRAC had second highest fast parameter estimation convergence speed in the closed loop simulations. It also had the tightest variation of current tracking error. It was economic in terms of computation cost. These benefits were due to that the MRAC took

advantage of the motor current dynamics and used all feedback data for estimation and current tracking. However, since the parameter estimation in the MRAC was an intermediate means to achieve low current tracking error, it was more sensitive to the outer loop system dynamics than other algorithms.

From the user perspective, the command torque (current) tracking is the most important criterion for the motor controller selection. The stability of the algorithm plays equally important role for practical control implementation. In addition, the algorithm complexity and computation costs are also important factors for implementing the control algorithm. Considering these factors, the MRAC algorithm appears to be the most promising candidate for the BLDC motor actuator.

Chapter 7 AN OPERATION SIMULATION MODEL FOR BLDC MOTORS

In Chapter 2, we have showed that the dq -model of a sinusoidal BLDC motor is mathematically equivalent to its corresponding stator phase model. Due to its simplicity and convenience, the dq -model has been used for the BLDC motor controller development. We have also pointed out several supplemental functions are needed to practically implement the controllers designed on the dq -model. These supplemental functions may be assumed to be ideal when designing the controller, but they have some impact on the controller implementation. Care must be taken to address some practical issues when implementing a dq -coordinate controller, for example, the rotor angular position measurement and the phase control voltage modulation. Traditionally these factors can be studied through the trial and error method during prototype development in labs. In the EPAS application, due to the high torque requirement and low battery voltage, the BLDC motor phase resistance is very low and the peak current can be over 100 Amp. In this case, a minor error may result in damage of the components such as the inverter. Thus the traditional trial and error method may not be an efficient way for control prototype development. Validation of control software in the simulation before conducting motor tests can significantly reduce the risk of prototype failure. It is often desirable to have the capability of simulating these practical implementation issues. In this chapter, a detailed Simulink model of a typical BLDC motor will be discussed and developed. We will discuss two practical issues of the controller initialization and transducer resolution in simulation with this quasi-physical BLDC model.

7.1 A Quasi-Physical BLDC Motor Model

An operating BLDC motor system consists of a PMSM motor, an inverter, an inverter switching-logic controller, a motor controller, a motor angular displacement/velocity transducer, and motor phase current transducers. A typical BLDC motor system is shown in Figure 7.1. In many cases, the inverter switching-logic control algorithm and the motor controller are programmed in a microcontroller. The angular displacement transducer can be an optical encoder or a resolver. Some BLDC motors are equipped with a set of Hall effect sensors, which can replace the motor angular displacement transducer for the inverter switching logic control purpose. The inverter can be considered as the actuator of the closed loop BLDC motor system. The microcontroller achieves the motor control by adjusting the switching timing and duty cycle of the inverter.

We have discussed the PMSM motor phase model and its *dq*-coordinate motor controller. However, since most BLDC motors are three-phase Y-connected, we will use the Line-to-Line motor PMSM motor model so as to be close to real system. In addition, a complete simulation model of the BLDC motor system must also include other components such as the inverter, the inverter switching logic controller, the rotational displacement transducer, the Hall effect sensors, and the phase current sensors. We will discuss these components in the following subsections.

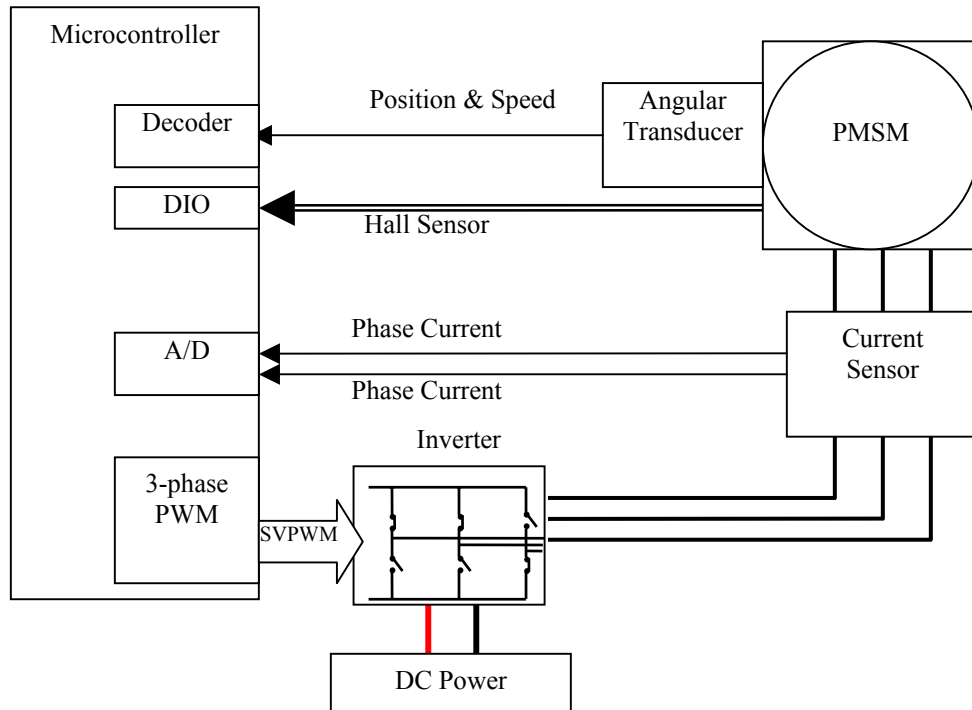


Figure 7.1 A typical three-phase BLDC motor system block diagram.

7.1.1 The Line-to-Line PMSM model

For a Y-connected PMSM motor, the neutral point is usually hidden in the motor housing and therefore not accessible for the controller. Instead, the inverter controls the line-to-line voltages across each pair of the three phases. We will need a line-to-line model of the motor instead of the phase model in equation (2.1). Rewrite the phase model (2.1 or 2.9) as

$$\mathbf{L} \frac{d\mathbf{i}}{dt} = -\mathbf{R}\mathbf{i} + \mathbf{V} - \mathbf{E} \quad (7.1)$$

where $\mathbf{i} = [i_1, i_2, i_3]^T$ is the phase current vector, $\mathbf{V} = [v_1, v_2, v_3]^T$ is the phase voltage

$$\text{vector, } \mathbf{L} = \begin{bmatrix} L_a & -\frac{L_a}{2} & -\frac{L_a}{2} \\ -\frac{L_a}{2} & L_a & -\frac{L_a}{2} \\ -\frac{L_a}{2} & -\frac{L_a}{2} & L_a \end{bmatrix}, \text{ and } \mathbf{E} = [e_1, e_2, e_3]^T \text{ is the phase back EMF vector with}$$

$$\begin{pmatrix} e_1 \\ e_2 \\ e_3 \end{pmatrix} = \begin{bmatrix} -\omega n_p i_f L_{m1} \sin(n_p \theta) \\ -\omega n_p i_f L_{m1} \sin(n_p \theta - \frac{2\pi}{3}) \\ -\omega n_p i_f L_{m1} \sin(n_p \theta + \frac{2\pi}{3}) \end{bmatrix} = \sqrt{\frac{2}{3}} K_e \omega \begin{bmatrix} -\sin(n_p \theta) \\ -\sin(n_p \theta - \frac{2\pi}{3}) \\ -\sin(n_p \theta + \frac{2\pi}{3}) \end{bmatrix}. \quad (7.2)$$

Define line-to-line variable as

$$\begin{pmatrix} f_{12} \\ f_{23} \\ f_{31} \end{pmatrix} = \begin{pmatrix} f_1 - f_2 \\ f_2 - f_3 \\ f_3 - f_1 \end{pmatrix} = \begin{bmatrix} 1 & -1 & 0 \\ 0 & 1 & -1 \\ -1 & 0 & 1 \end{bmatrix} \begin{pmatrix} f_1 \\ f_2 \\ f_3 \end{pmatrix} \quad (7.3)$$

where f can be i , v or e . Multiply both sides of (7.1) by $\begin{bmatrix} 1 & -1 & 0 \\ 0 & 1 & -1 \\ -1 & 0 & 1 \end{bmatrix}$, the line-to-

line model is obtained as

$$\begin{cases} \frac{3}{2} L_a \frac{di_{12}}{dt} + R \cdot i_{12} + e_{12} = V_{12} \\ \frac{3}{2} L_a \frac{di_{23}}{dt} + R \cdot i_{23} + e_{23} = V_{23} \\ \frac{3}{2} L_a \frac{di_{31}}{dt} + R \cdot i_{31} + e_{31} = V_{31} \end{cases} \quad (7.4)$$

With line-to-line control voltage vector $(V_{12}, V_{23}, V_{31})^T$ from the inverter, the line-to-line currents $(i_{12}, i_{23}, i_{31})^T$ will be solved from (7.4). Using the Kirchoff first law, the phase currents are then calculated from the line-to-line currents as

$$\begin{pmatrix} i_1 \\ i_2 \\ i_3 \end{pmatrix} = \begin{bmatrix} 2/3 & 1/3 & 0 \\ -1/3 & 1/3 & 0 \\ -1/3 & -2/3 & 0 \end{bmatrix} \begin{pmatrix} i_{12} \\ i_{23} \\ i_{31} \end{pmatrix} \quad (7.5)$$

The torque generated by the motor can be calculated with

$$\tau = \frac{(e_1 i_1 + e_2 i_2 + e_3 i_3)}{\omega} = -k_e \left[i_1 \sin(\omega t + \theta_0) + i_2 \sin\left(\omega t + \theta_0 - \frac{2\pi}{3}\right) + i_3 \sin\left(\omega t + \theta_0 + \frac{2\pi}{3}\right) \right] \quad (7.6)$$

The line-to-line PMSM motor model is implemented as a state space model in Simulink, as shown in Figure 7.2.

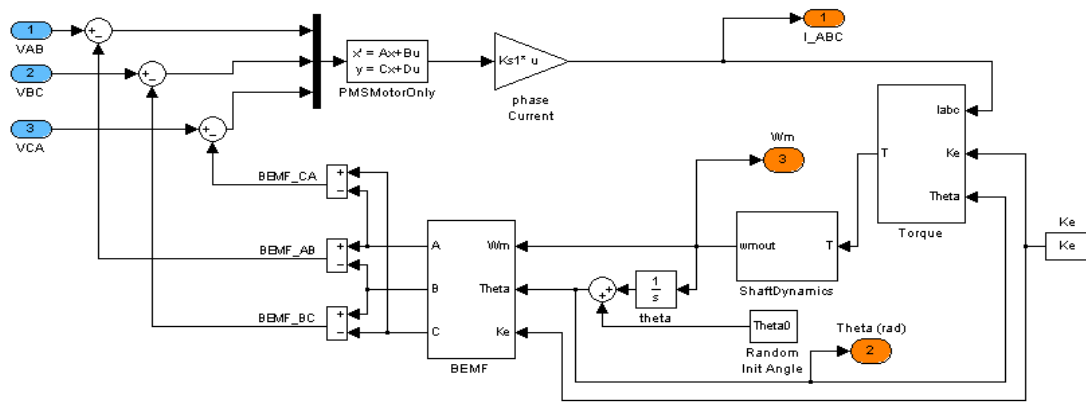


Figure 7.2 The line-to-line PMSM model in Simulink.

7.1.2 The H-bridge PWM inverter model

Typically, a three-phase PMSM motor requires three-phase AC control voltages. The DC power supply to the BLDC motor cannot directly generate the AC control voltages. It is the function of the three-phase H-bridge PWM inverter to generate appropriate AC voltages for the PMSM motor. The H-bridge consists of six or more digitally controlled power transistors such as hexagonal field effect transistors (HEXFET), as shown in

Figure 7.3. Usually the power transistor response time is in the order of nano (10^{-9}) seconds, which is significantly shorter than a cycle of the PWM signal (usually in 10^{-5} seconds). It is reasonable to assume that these trans. They can be simplified as a switch controlled by the PWM pulse. If the upper transistor in a branch of the H-bridge is switched on, the corresponding phase terminal is connected to the positive terminal of the DC power supply. If the lower transistor is switched on, the corresponding phase terminal is connected to the negative terminal of the DC power supply. Obviously, the upper transistor and the lower transistor in any branch cannot be switched on simultaneously.

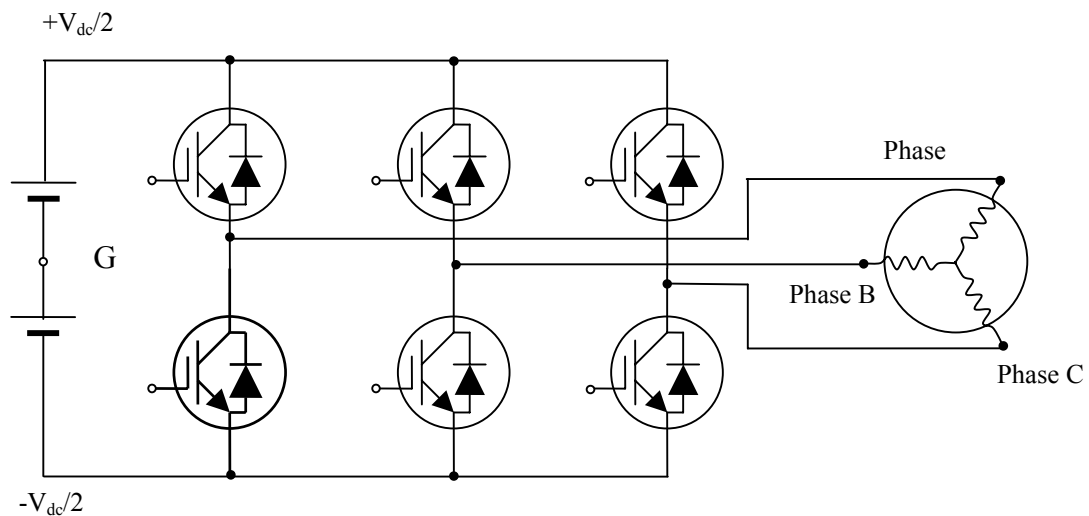


Figure 7.3 A simplified circuit of the BLDC motor system.

To calculate the line-to-line voltages for the PMSM motor, we define the ground reference (zero potential) point as the mid of DC power supply, as shown in Figure 7.3. The H-bridge sets the motor stator pole to either $V_{dc}/2$ or $-V_{dc}/2$, while the line-to-line voltage is either V_{dc} or $-V_{dc}$. Note that the potential at the neutral point of the motor (the

center of Y connection) changes with the different H-bridge switching configuration, therefore the stator phase voltage may be different from $V_{dc}/2$. For each branch of the H-bridge, if the switch control signal is high (1), the corresponding stator phase potential is set to $V_{dc}/2$; otherwise, it is set to $-V_{dc}/2$. The phase pole potential voltages are then used to calculate line-to-line voltage among the three phases. The Simulink model block diagram of the H-bridge is shown in Figure 7.4.

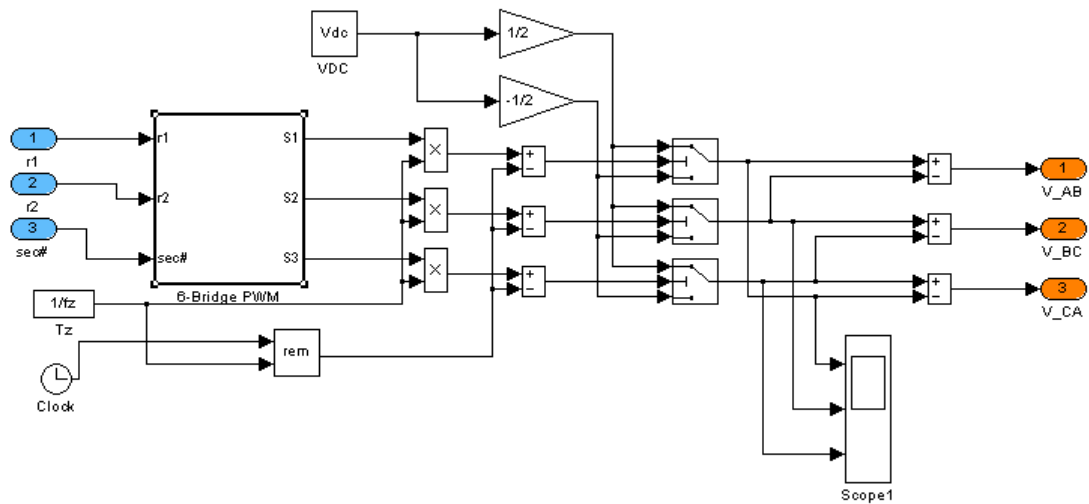


Figure 7.4 The H-bridge inverter model.

7.1.3 The rotor angular displacement transducer

The quadrature incremental encoder is a commonly used rotational displacement transducer in motion control applications. A typical quadrature incremental encoder can be modeled as

$$x = 0.25 \times \text{round}\left(4 \cdot \text{cpr} \cdot \frac{\theta}{2\pi}\right) \quad (7.7)$$

where x is the encoder count output, and cpr is the count per resolution.

The incremental encoder starts counting from zero when the function is enabled. Therefore, an independent integrator is used to simulate the relative input angle. Figure 7.5 shows the Simulink block diagram for the incremental encoder model.

It is worth mentioning that the incremental encoder only measures a relative displacement from the initial position of the rotor. The initial rotor angle is modeled as a random number uniformly distributed on the interval of $[0, 2\pi]$. We have discussed that importance of the initial angle in the coordinate transformation calculations. Therefore, when using a relative rotor angular displacement transducer, the motor controller needs a strategy to find the zero-angle configuration.

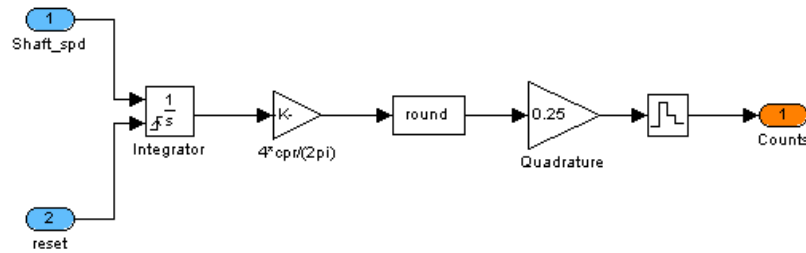


Figure 7.5 The BLDC motor components block diagram.

7.1.4 The Hall effect sensor

Trapezoidal BLDC motors are usually equipped with a set of three Hall effect sensors for the rotor position feedback. Sinusoidal BLDC motors can be operated in the same manner as trapezoidal BLDC motors if they are equipped the Hall effect sensors. These Hall effect sensors provide the angular position of rotor magnetic field and can be used to

synchronize the stator magnetic field to the rotor permanent magnets. Each of the Hall sensors can be simplified as following functions:

$$H_i = \begin{cases} 1, & \theta_i \leq \theta < \theta_i + \pi/n_p \\ 0, & \text{others} \end{cases}$$

where θ is the rotor angle, n_p is the number of permanent magnet pole pairs, and θ_i is the relative angle between the reference point and the angular position of the hall sensor. θ_i can be obtained from the motor manufacturer specification or from a motor phase test.

7.1.5 SVPWM

The Space Vector PWM (SVPWM) is the most widely used inverter switching mechanism for the sinusoidal BLDC motors. It achieves the voltage vector control by adjusting the timing and duty cycle of the eight switching states of the three-phase H-bridge inverter. Assuming that stator coils in the three phases are identical, each switching state of the H-bridge corresponds to a voltage vector in the three-phase stator coil frame. Let's look at the example of the switching state (1, 0, 0) for the branch (a, b, c) of the H-bridge. The upper gate of branch a, the lower gates of branch b and c are turn on. Referring to Figure 7.3, the pole of phase A is connected to $+V_{dc}/2$ and the poles of phase B and C are connected to $-V_{dc}/2$. The phase voltage vector will be

$\left(\frac{2}{3}V_{dc}, -\frac{1}{3}V_{dc}, -\frac{1}{3}V_{dc}\right)^T$. Similarly, the eight basic voltage vectors ($v_0 \sim v_7$) for a Y-connected motor in the three-phase frame are shown in Figure 7.6 and their corresponding switch states are shown in Table 7.1. Notice that v_0 and v_7 are zero vectors.

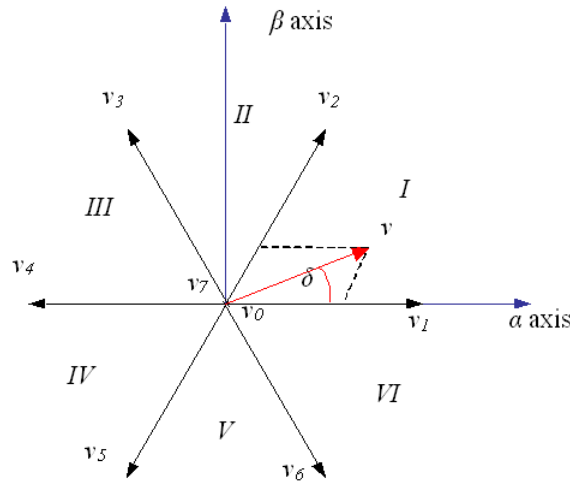


Figure 7.6 The voltage vectors in the space vector modulation.

Table 7.1 The eight basic voltage vectors in the SVPWM

Voltage Vectors	Bridge State			Motor Phase Voltage ($\times V_{dc}$)			Line to Line Voltage ($\times V_{dc}$)		
	a	b	c	V_{AN}	V_{BN}	V_{CN}	V_{AB}	V_{BC}	V_{CA}
v_0	0	0	0	0	0	0	0	0	0
v_1	1	0	0	$2/3$	$-1/3$	$-1/3$	1	0	-1
v_2	1	1	0	$1/3$	$1/3$	$-2/3$	0	1	-1
v_3	0	1	0	$-1/3$	$2/3$	$-1/3$	-1	1	0
v_4	0	1	1	$-2/3$	$1/3$	$1/3$	-1	0	1
v_5	0	0	1	$-1/3$	$-1/3$	$2/3$	0	-1	1
v_6	1	0	1	$1/3$	$-2/3$	$1/3$	1	-1	0
v_7	1	1	1	0	0	0	0	0	0

Given a voltage vector in the three-phase frame, we can find a linear combination the eight basic voltage vectors to approximate the vector by weighing the duty cycle for each inverter state. For example, the vector v shown in Figure 7.6 can be a linear combination of v_1 , v_2 and the zero vectors v_0 , v_7 as

$$v = r_1 v_1 + r_2 v_2 + \frac{1}{2}(1 - r_1 - r_2)v_0 + \frac{1}{2}(1 - r_1 - r_2)v_7, \quad (7.8)$$

where $r_1 = \frac{|\vec{v}| \sin(\pi/3 - \delta)}{|\vec{v}_1| \sin(\pi/3)}$ and $r_2 = \frac{|\vec{v}| \sin(\delta)}{|\vec{v}_1| \sin(\pi/3)}$ are the duty cycles of the vectors v_1, v_2 ,

$0 \leq \delta < \pi/3$ is the angle between the vector v and the vector v_1 . We can separate the 2-D space in to six sectors (I~VI) as shown in Figure 7.6. Any vectors located in the 6 sectors can be expressed as a linear combination of its nearby basic vectors

$$\vec{v} = r_1 \vec{v}_k + r_2 \vec{v}_{k+1} + \frac{1}{2}(1 - r_1 - r_2)\vec{v}_0 + \frac{1}{2}(1 - r_1 - r_2)\vec{v}_7, \quad (7.9)$$

where

$$\begin{cases} r_1 = \frac{|\vec{v}| \sin[\pi/3 - (\delta - k\pi/3)]}{|\vec{v}_k| \sin(\pi/3)}, \\ r_2 = \frac{|\vec{v}| \sin(\delta - k\pi/3)}{|\vec{v}_{k+1}| \sin(\pi/3)}, \end{cases} \quad k\pi/3 \leq \delta < (k+1)\pi/3, k = 0, \dots, 5.$$

$k = 0, 1, \dots, 5$ is the sector number corresponding to the I~VI in Figure 7.6. The transistors on/off timing can be calculated as

$$\begin{cases} T_1 = r_1 \cdot T_z, \\ T_2 = r_2 \cdot T_z, \\ T_0 = (1 - r_1 - r_2) \cdot T_z \end{cases} \quad (7.10)$$

where $T_z = 1/f_z$, and f_z is the carrier PWM signal frequency. The six-bridge transistor on-off timing calculation is summarized in Table 7.2. Figure 7.7 shows the six transistors' on-off timing in one PWM cycle when the voltage vector in each sector.

Table 7.2 Summary of the transistor on-off timing calculation in each sector

Vector	High Switches (S_1, S_2, S_3); Low switches $S_4 = \text{NOT}(S_1), S_5 = \text{NOT}(S_2), S_6 = \text{NOT}(S_3)$;
0	$S_1 = T_1 + T_2 + T_0/2, S_2 = T_2 + T_0/2, S_3 = T_0/2$
1	$S_1 = T_1 + T_0/2, S_2 = T_1 + T_2 + T_0/2, S_3 = T_0/2$
2	$S_1 = T_0/2, S_2 = T_1 + T_2 + T_0/2, S_3 = T_2 + T_0/2$
3	$S_1 = T_0/2, S_2 = T_1 + T_0/2, S_3 = T_1 + T_2 + T_0/2$
4	$S_1 = T_1 + T_0/2, S_2 = T_0/2, S_3 = T_1 + T_2 + T_0/2$
5	$S_1 = T_1 + T_2 + T_0/2, S_2 = T_0/2, S_3 = T_1 + T_0/2$

For the simulation purpose, the physical switching process is simplified as a comparison between the normalized PWM cycle time to the duty cycle ratio:

$$S_i = \begin{cases} 1 & \frac{t}{t_z} < r \\ 0 & \frac{t}{t_z} \geq r \end{cases}, i = 1, 2, 3,$$

where S_i is the upper arm switch input, t_z is the PWM cycle interval, r is the duty cycle ranging from 0 to 1. The SVPWM function takes the sector number and the corresponding base voltage vector timing (r_1, r_2) as input, and outputs the H-bridge branch on timing signal (S_1, S_2, S_3). It is implemented in Simulink as shown in Figure 7.8.

7.1.6 Simulation configuration

With the components model discussed in the previous subsection, a complete model of a sinusoidal BLDC motor system is implemented in Simulink, as shown in Figure 7.9.

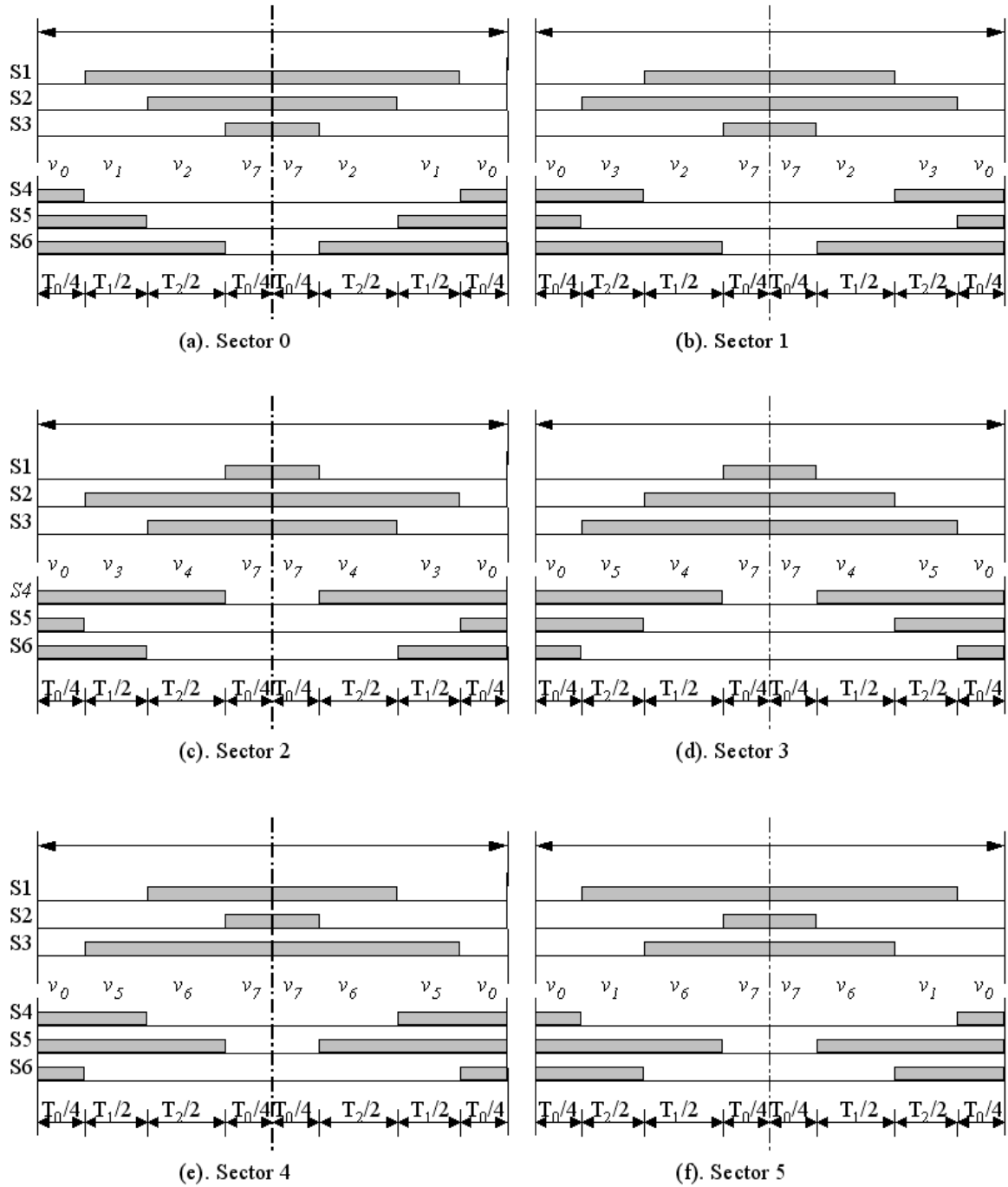


Figure 7.7 Transistors' on-off timing in the six sectors

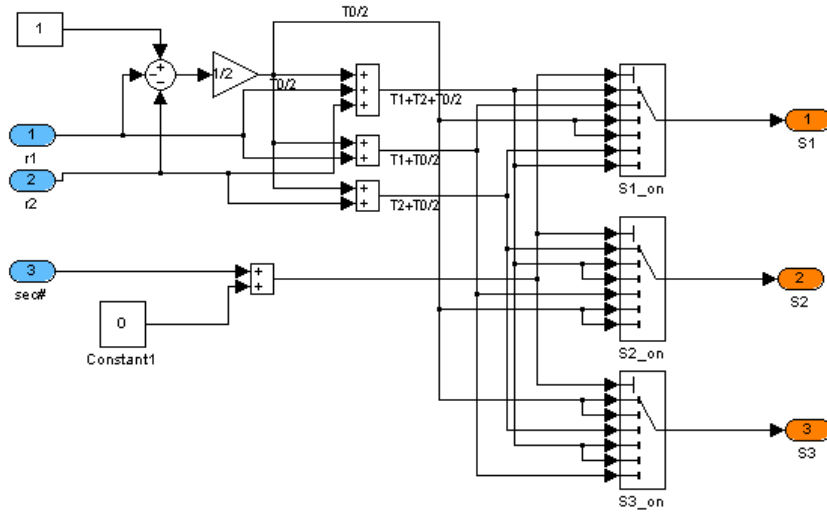


Figure 7.8 SVPWM function implementation in Simulink

To ensure good resolution, the SVPWM function, the inverter and the PMSM motor must be running at very high frequency since the actual PWM signal frequency is usually 10~20kHz. The sampling step for these components was set to $2\mu\text{s}$. The motor controller, Hall-effect sensors and incremental encoder model was running at 2ms. Therefore, the rate-transition and zero-order-hold function have to be used in order to complete the closed loop simulation. In the simulation, the same set of motor parameters will be used as in the previous chapters: $R = 0.05\Omega$, $K_e = 0.05 \text{ Nm/A}$, $L = 1 \times 10^{-4} \text{ h}$, $n_p = 3$.

7.2 Study of Controller Implementation Issues in Simulation

With the BLDC motor system model, the controller implementation issue can be studied in simulation. For control algorithms developed on the dq -coordinate model, the controller needs to convert the command voltage (V_d , V_q) into the three-phase voltages. The phase voltages will be converted into a voltage vector in the stator phase frame, and a sector number and the base voltage vector timing (r_1 , r_2) will be calculated from

equation (7.9) as the final controller output. An example Simulink model of this process is shown in Fig 7.10.

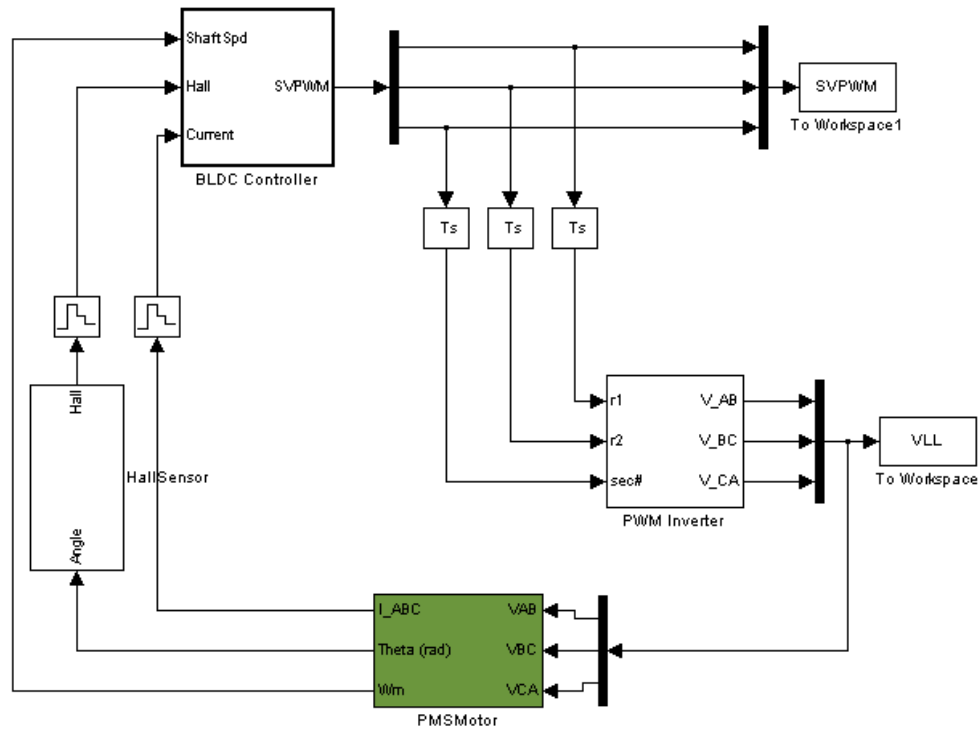


Figure 7.9 The Simulink model of a closed loop BLDC motor system

7.2.1 Initialization

In addition to the coordinate transformation and SVPWM encoding, the controller must deal with the initial angle problems. If the BLDC motor system is equipped with an incremental encoder, the actual rotor angle is unknown when the microcontroller is powered up. The controller needs make sure that the angular position measurement is consistent with the zero-angle configuration defined in the coordinate transformations. One solution is to force the rotor into the zero-angle configuration before normal

operation. For example, the controller outputs the base voltage vector v_1 for long enough time such that the rotor reaches the equilibrium state where the d axis of the rotor magnetic field is aligned to the stator phase 1 axis. This is the exact zero-angle configuration that was defined for the coordinate transformations in Chapter 2. The settling time depends on the duty cycle of the controller output and damping factor of the rotor shafts. It can usually be limited in the order of 0.1s. At the end of the starting scheme, the controller switches to the normal operation model, and the incremental encoder counter is reset to zero. Thus, in the normal operation mode, the rotor angle feedback from the encoder will be consistent with the coordinate transformation calculations. This forced-alignment initialization is implemented in Simulink, as shown in Fig 7.11.

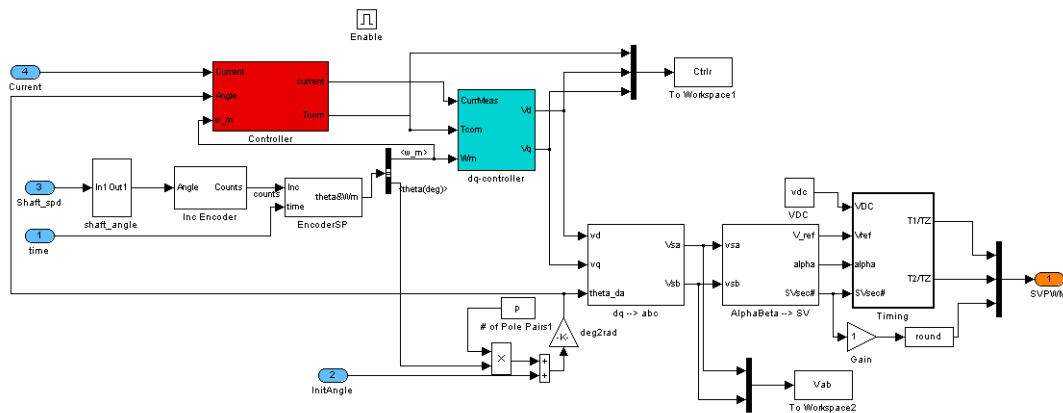


Figure 7.10 The practical controller model in Simulink

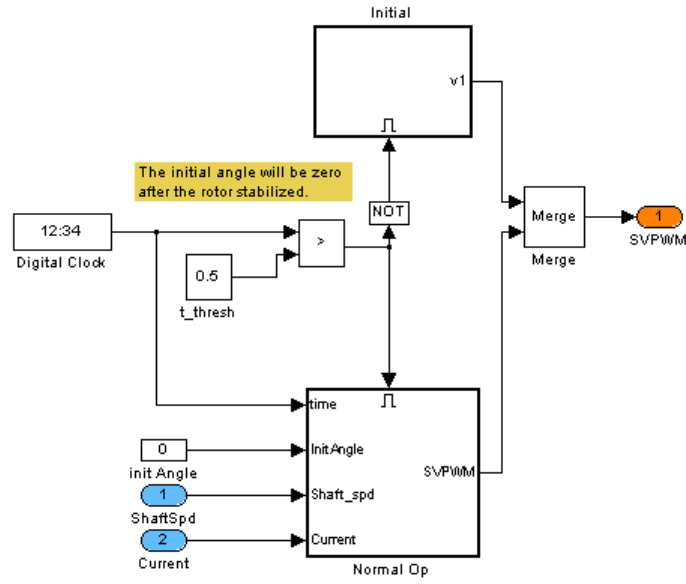


Figure 7.11 The starting scheme Simulink model

The simulation of the initialization strategy with the physical motor model validated the effectiveness of the above solution for real control implementation. In the simulation, the controller commanded an aligning voltage vector in the direction of the basic voltage vector v_1 with a magnitude of $0.1V_{dc}$. The starting mode was set to be 0.5 seconds long. In the normal operation mode, the command torque (τ_{cmd}) was set as 0.5Nm, and a step load torque of 0.4Nm was applied to the rotor shaft at time $t = 1$ s. Figure 7.12 shows simulation results of the phase currents, the actual motor dq currents, the rotor velocity and angular displacement, the generated torque and the line-to-line voltages. In the initialization mode, the aligning voltage drove the rotor from a random initial angle to the zero-angle configuration within 0.2 seconds. At $t=0.5$ s, the controller switched to the normal operation model, the controller used the angle feedback from the encoder for all coordinate transformation calculations. The motor generated the commanded torque

steadily, which indicated that the stator magnetic field was synchronized with rotor magnetic field.

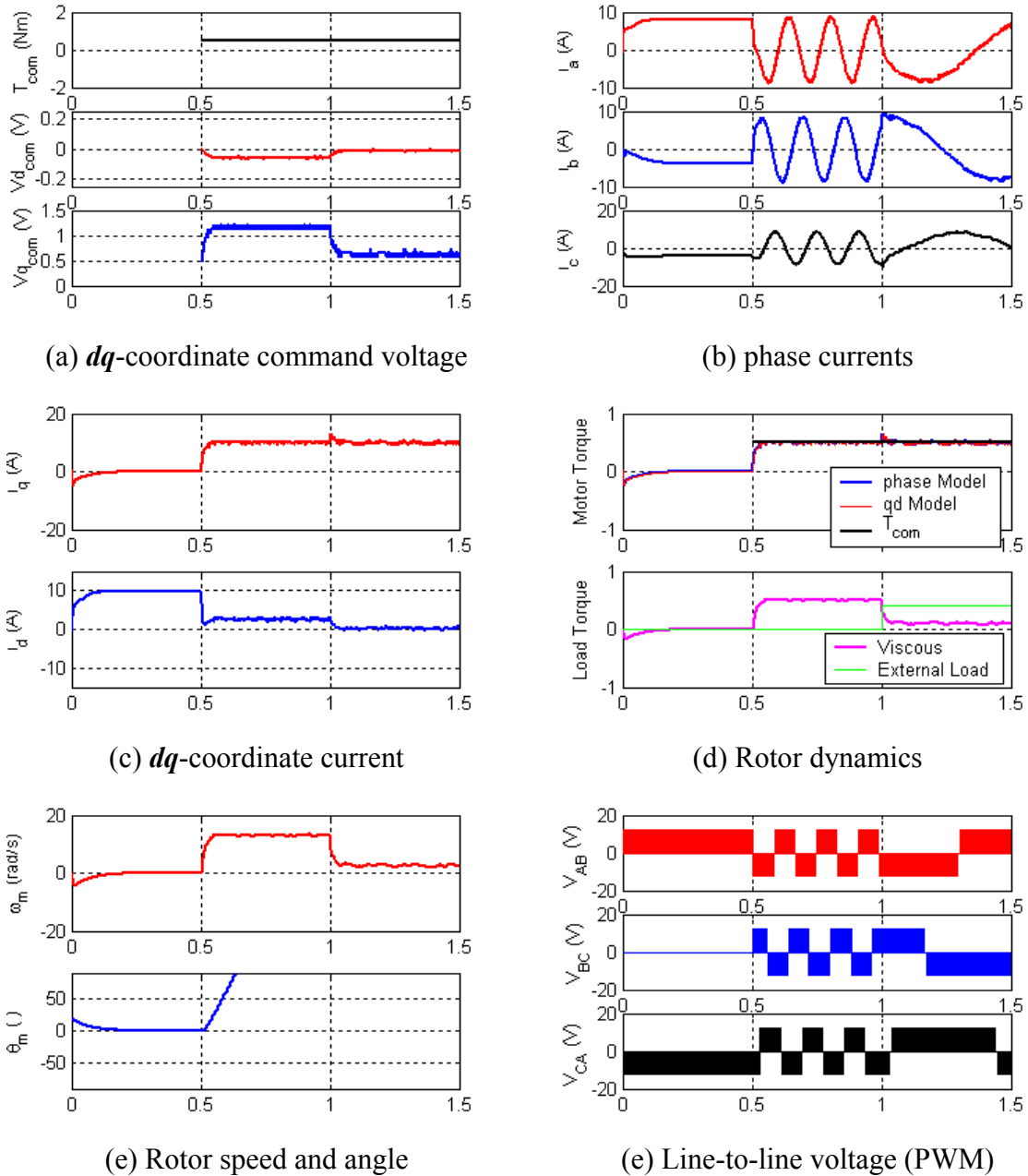


Figure 7.12 The BLDC motor physical model closed loop simulation

7.2.2 Incremental encoder resolution

The quasi-physical BLDC motor model simulation can be used to study some other practical issues, for example, the resolution of the incremental encoder (*cpr*). If the incremental encoder is the only angular displacement transducer, its resolution can be an important factor to the performance of the BLDC motor actuator. As shown in the previous chapters, the BLDC motor controller in the *dq*-coordinate needs motor velocity to calculate the control voltages. In addition, the coordinate transformation functions and the SVPWM depend on the rotor angular position feedback to calculate accurate output for the motor. The accuracy of the rotor angle feedback has an impact on almost all the functions in the motor controller.

A prototype BLDC motor for the EPAS application is equipped with a low-resolution quadrature encoder (36 *cpr*). Simulation results with such an encoder are shown in Figure 7.13. In the simulation, the controller switched to normal operation mode at $t=0.5s$. During the next 0.2 seconds, the motor velocity calculation was very noisy, and the phase currents were not in proper sinusoidal curve shape. The reason was that the low-resolution encoder was not able to measure rotor angular displacement less than 2.5 degree from one sample to the next. This might result in about 21.8rad/s error in motor velocity feedback since the motor controller sampling time was 2ms. The coordinate transformation calculations were inaccurate and caused slight loss of synchronization between the stator and rotor magnetic fields. Therefore the phase currents became noisy and ill shaped. In comparison, Figure 7.14 shows simulation results with a high-resolution encoder (4096 *cpr*). With the high-resolution rotor angle feedback, motor velocity, control voltages and phase currents were smooth and in desirable shape for

proper operation of the BLDC motor. Obviously, the 36-CPR encoder could not provide enough resolution for smooth BLDC motor operation, especially during the low speed operation.

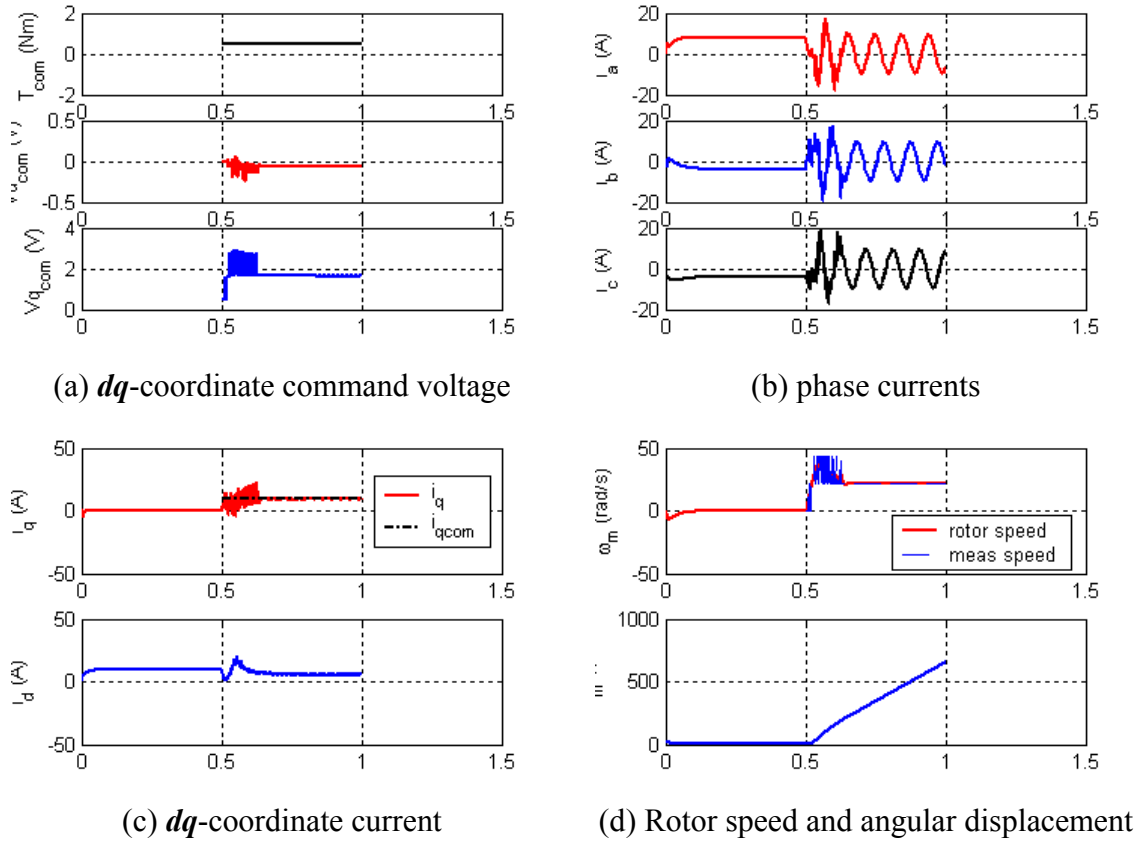


Figure 7.13 The BLDC motor simulation with a low-resolution encoder ($cpr=36$)

7.3 Limit of the Quasi-Physical BLDC Motor Model

The quasi-physical BLDC motor model includes model of all necessary components in a BLDC motor system, therefore it reveals more practical operation characteristics of the BLDC motor. We have shown the benefit of this model in study of initialization and incremental encoder resolution for practical BLDC motor applications. Simulation results

of this model can also be used for education purpose. It reveals more details of the BLDC motor operation than the simpler dq -coordinate model, though the latter is more convenient.

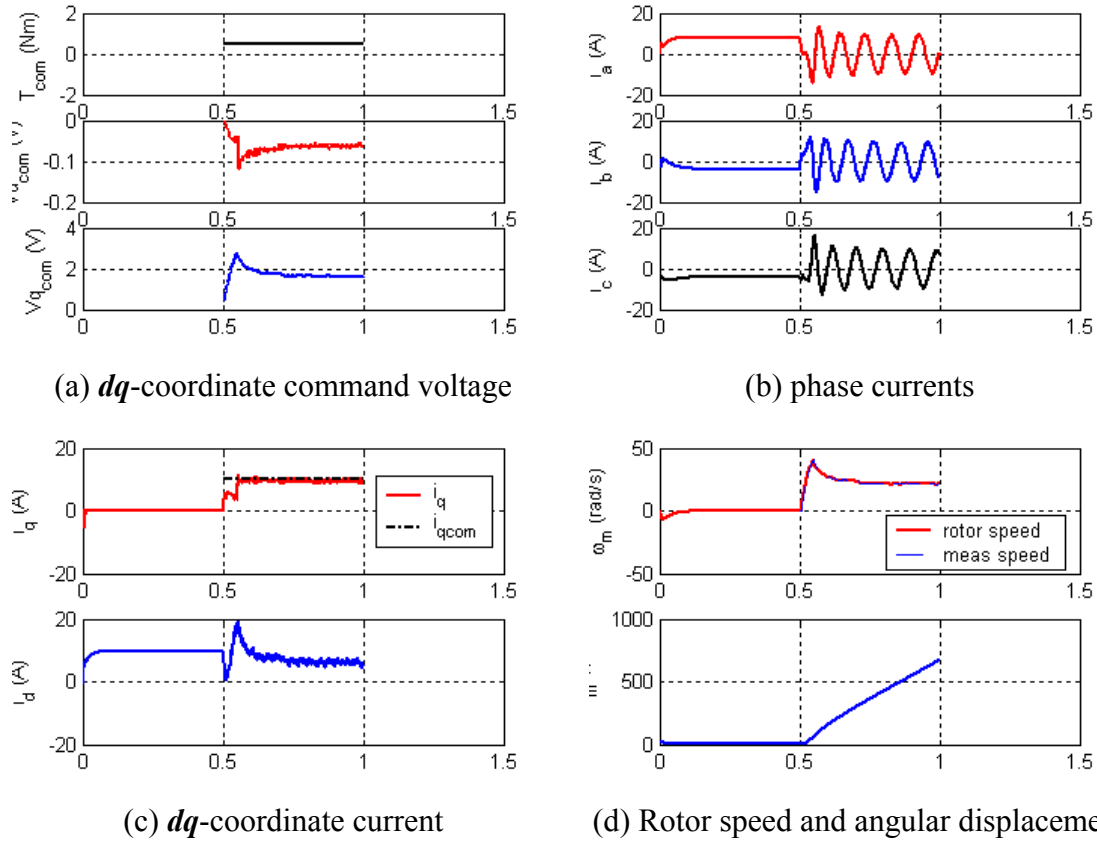


Figure 7.14 The BLDC motor simulation with a high-resolution encoder ($cpr=4096$)

Unfortunately, simulation of the quasi-physical BLDC motor model still cannot replace real motor experiments. One disadvantage of the simulation is its simulation speed. Since the SVPWM function, the inverter model and the PMSM motor model have to run at very short step ($2\mu s$), the simulation cannot run at real time. On a PC equipped with a 1.8GHz CPU and MATLAB 6.5, it took more than 100 seconds to complete a 2-second simulation. In addition, because the PMSM motor model was solved 500,000 times each

second, simulation results was relatively sensitive to the selection of solver. In the Simulink environment, the ode2 solver gave the closest simulation results to that of the *dq*-coordinate model alone.

7.4 Summary

In this chapter, a quasi-physical model was developed for a typical three-phase Y-connected sinusoidal BLDC motor. This model included the rotor position/speed transducers, the inverter, the SVPWM algorithm and etc. A line-to-line PMSM model replaced the phase model. Closed loop simulation of the controllers and the physical model helped to study the implementation issues of the *dq*-coordinate controllers such as the initial rotor angular position and the incremental encoder resolution.

The quasi-physical BLDC model is not a real-time model. It was slow in simulation because very short sampling step ($2\mu\text{s}$) was needed to simulate the interaction between the inverter and the PMSM motor with good resolution. For similar reason, the simulation results were also slightly sensitive to selection of solver. Despite of these limits, the quasi-physical model simulation was proved to be a useful tool to validate the controller functions that were not discussed in the *dq*-coordinate controller model, for example, the SVPWM and the coordinate transformation functions. With the quasi-physical model simulation, the risk of equipment damage by the traditional trial and error method can be significantly reduced.

Chapter 8 CONCLUSION AND FUTURE WORKS

8.1 Conclusion

The Electrical Power Steering system (EPAS) will be the main stream of future automotive power steering system for its advantage of energy efficiency and flexibility. The sinusoidal brushless DC motor has been identified as one of the most suitable candidate actuator for the EPAS. The long service life, harsh working environment and mass production impose motor parameter variation problem for the EPAS actuators. Adaptive control is one of the most suitable candidate techniques for the parameter variation problem.

The *dq*-coordinate model has been widely used for the BLDC motor control and dynamics analysis. We started the adaptive control development with derivation of the *dq*-coordinate model for a typical three-phase sinusoidal BLDC motor. The mathematical derivation revealed how the rotor angular position (θ) was cancelled in the coordinate transformations from the three-phase frame to the *dq*-coordinate, but the θ was indispensable in the implementation of the any controller designed with the *dq*-coordinate model. The derivation also explained the importance of the zero-angle ($\theta=0$) definition. Though it was not shown in the *dq*-coordinate model, the zero-angle configuration determined the phase of almost all sinusoidal functions in the coordinate transformations.

With the *dq*-coordinate model, several adaptive algorithms were studied for the sinusoidal BLDC motor application. The first group of algorithms estimated the motor parameters by integrating the motor parameter errors that were solved algebraically with

q -axis current feedback from one or more loops. These algorithms were denoted as the q -solver algorithms. Certain conditions were set on the solved parameter errors to avoid noise due to singularity in matrix inversion operations. Stability of the algorithms was proved mathematically and improvements were proposed for such algorithms by estimating the motor dynamics and rotor velocity sampling delay. In addition, the Gram-Schmidt orthonormalization process was proposed to improve estimation performance by dealing with the correlation between the rotor velocity and command torque.

The second adaptive algorithm was proposed to estimate the parameter errors by using both the q -axis and d -axis current (i_q, i_d) feedback through the same process as that of the q -solver algorithms, since i_q and i_d always existed in pair. This algorithm was denoted as the qd -solver. Comparing to the q -solver algorithms, the qd -solver was simpler since it only required one loop of current feedback to calculate the parameter error. On the other hand, the qd -solver required arbitrary d -axis command current so as to get desirable estimation performance. The conditions for unbiased estimation of the parameter errors were studied through Monte Carlo simulations. These conditions turned out to be similar to the conditions in the q -solver algorithms that were set on the solved parameter errors to avoid singularity in matrix inversion operations.

The third parameter estimation algorithm was the extended Kalman filter (EKF). The EKF took advantage of the knowledge of the nonlinear plant model. Unlike the previous two algorithms that had a fixed gain for integrating the parameter errors, the EKF computed linearization of the nonlinear plant model and adjusted the Kalman gains in real time. The EKF is close to optimal filtering given steady state motor operation. It demonstrated superior performance in both parameter and state estimation with fast

convergence and decent accuracy in simulations. However, the computation cost and the stability issue were barriers for implementing the EKF in practice.

The model reference adaptive control (MRAC) was the fourth algorithm studied for the BLDC motor parameter variation problem. By taking a reference model for the closed loop motor dynamics, the MRAC algorithm utilized the derivative of i_q and i_d for the parameter estimation and current tracking. Comparing to the q -solver and the qd -solver, the MRAC improved the data efficiency and therefore the convergence speed of parameter estimation. The MRAC was designed to maintain non-positive derivative of a quadratic cost function of the reference model tracking error and the parameter errors. Therefore, the stability of this algorithm was guaranteed. The MRAC was computationally simpler and faster than the EKF algorithm.

For development purpose, all these adaptive control algorithms were validated in motor bench test simulations in which the command torque and the motor velocity were assumed to be Gaussian random signals and independent to each other. In real applications, the command torque and the motor velocity are usually correlated to some extent. To evaluate the performance of these algorithms, they were compared in the closed loop simulation of an EPAS system and a motor speed control application. In general, the closed loop simulation results revealed a trade-off between the performance and algorithm complexity (or computation costs). Better performance was obtained at the cost of higher complexity of the control algorithm. Moreover, the simulation results indicated that the outer loop controller and plant dynamics would reduce the performance of the adaptive controllers if the outer loop controller had relatively narrow frequency

response or if the outer loop system dynamics responded differently to the different parameter error combinations.

In the closed loop simulations, the EKF showed the best parameter estimation performance among the four algorithms. It was also relatively insensitive to the command signals from the outer loop systems. These advantages of the EKF were due to the real time linearization of the nonlinear model and the optimal Kalman filtering based on the linearized model. The desirable performance of the EKF came at the highest computing cost. With the EKF algorithm, the controller needed to compute the linearization function and to update a 4x4 covariance matrix for the 4 augmented states at every loop. This resulted in high requirement on computation resources.

The q -solver with the Gram-Schmidt orthonormalization showed the slowest parameter estimation convergence, but it had almost the tightest steady state parameter estimation error bound. The orthonormalization and approximation of the inverse motor dynamics consumed second highest computation resource. The qd -solver required less computation resource, but its steady state accuracy was the lowest among the four algorithms, especially in R . This was due to the fact that the d -axis current was significantly lower than the q -axis current in magnitude. The slow parameter estimation of these two algorithms was partly because of the neglected current dynamics in parameter error computation and the conditions for singularity.

The MRAC showed the second highest fast parameter estimation convergence speed in the closed loop simulation. It also had the tightest variation of current tracking error partly because of current tracking was its dominant control objective. However, since the parameter estimation was an intermediate means to achieve reference model current

tracking, it was relatively sensitive to the outer loop system dynamics. Comparing to the EKF and the q -solver with the Gram-Schmidt orthonormalization, the MRAC algorithm was simple and required less resource. Overall, the MRAC appeared to be a best combination of accuracy and computing complexity among the four algorithms.

The closed loop simulations also revealed that the outer loop system dynamics appeared to have impact on the performance of the adaptive BLDC motor controllers. From the motor controller's point of view, the outer loop controller and the EPAS plant together formed a feedback loop from the motor output torque/velocity to the command torque. The outer loop system had some specific frequency response due to its design. This usually narrowed the frequency contents of the command torque, which was one of the excitation signals to the adaptive controllers. As a result, the performance of the adaptive controllers reduced slightly when compared to the motor bench test simulations where the command torque and velocity were assumed to be random and independent. The EKF, with its observer structure, appeared to be less sensitive to the outer loop system dynamics. This was because that the EKF took the motor control voltages and current measurements as its input signals, and the command torque was not directly involved in its estimation process.

From the user perspective, the command torque or current tracking is the most important criterion for the motor controller selection. The stability of the algorithm plays equally important role for practical control implementation. In addition, the algorithm complexity and computation costs are also important factors for implementing the control algorithm. Considering these factors, the MRAC algorithm appears to be the most promising control candidate for the BLDC motor actuator in the EPAS application.

One of the unique characteristics of the BLDC motor control problem is that the control algorithms developed in the *dq*-coordinate cannot be directly used for real motor control application. Coordinate transformations are indispensable between the *dq*-coordinate controller and physical motor system. In addition, the control voltage must be encoded in a suitable inverter actuation scheme such as the space vector pulse width modulation. To facilitate the BLDC motor control implementation, a quasi-physical BLDC motor model was developed. Some practical control implementation issues were studied in simulation, including unknown initial angle, and rotor angular displacement feedback resolution. Even though the simulation could not run real-time, and the simulation results were slightly sensitive to solver, the simulation results proved that it was a useful tool to validate the controller functions such as the SVPWM and the coordinate transformations.

8.2 Future Works

Steer-by-Wire is one of the potential candidate technologies for future automotive steering systems. Comparing to the traditional hydraulic or mechanical steering system, the Steer-by-Wire system replaced the hardware link between the steering control (steering wheel) and the road wheels with electronically transmitted command signal. On the one hand, this technology creates greater flexibility of hardware installation and software reconfiguration that enables intelligent driver assistance and vehicle stability control. On the other hand, the electronic signal transmission is one of the main issues for the Steer-by-Wire in terms of fail safety and the robustness of the electronic control system. The sinusoidal BLDC motor is an ideal candidate actuator for the Steer-by-Wire systems. Fault tolerance will be a necessity for the BLDC motor control in such as system.

The BLDC motor operation involves the rotor position/speed transducer, PWM inverter that consists of three-phase rectifier and power transistors, and the PMSM. Typical fault includes position transducer failure, power transistor failure, and motor stator coil open circuit, short to ground, and etc. The parameter estimation algorithm and the EKF can be implemented for the fault diagnosis. Statistical study has to be carried out so as to obtain the fault detection thresholds.

Sensorless control has been a hot research and development direction for BLDC motors. Sensorless means that the PMSM will be operated without a rotor angular position sensor. Usually some certain type of observer is employed for asymptotical estimation of the motor speed and angular position. For the trapezoidal BLDC motors, BEMF zero crossing detection is also proposed to replace the phase Hall sensors. It is a good candidate technology for the fail-safe operation of the BLDC motor in case of position sensor fault. The PWM inverter provides not only a effective way for the PMSM operation with a DC power supply, it also give some flexibility for the BLDC motor fail-save operation in case of stator winding fault such as open circuit and inverter bridge component failure.

Appendix I THE PROBABILITY DENSITY FUNCTION DERIVATION

$$x \sim N(\mu_x, \sigma_x^2) = \frac{1}{\sigma_x \sqrt{2\pi}} \exp\left(-\frac{(x - \mu_x)^2}{2\sigma_x^2}\right) \quad (\text{A1.1})$$

$$y = \begin{cases} \frac{1}{x} & |x| > \varepsilon \\ \frac{1}{\varepsilon} & |x| \leq \varepsilon \end{cases} \quad (\text{A1.2})$$

$F_y(y)$ and $F_x(x)$ are the cumulative distribution function of y and x .

If $0 < y \leq \frac{1}{\varepsilon}$, then $x \geq \varepsilon$

$$F_y(y) = F_x\left(x \geq \frac{1}{y}\right) = \left[1 - F_x\left(\frac{1}{y}\right)\right], \quad 0 < y \leq \frac{1}{\varepsilon} \quad (\text{A1.3})$$

If $y > \frac{1}{\varepsilon}$, then $\varepsilon > x > 0$

$$F_y(y) = F_x(\varepsilon > x > 0) = [F_x(\varepsilon) - F_x(0)], \quad y > \frac{1}{\varepsilon} \quad (\text{A1.4})$$

If $y < -\frac{1}{\varepsilon}$, then $0 > x > -\varepsilon$

$$F_y(y) = F_x(0 > x > -\varepsilon) = [F_x(0) - F_x(-\varepsilon)], \quad y < -\frac{1}{\varepsilon} \quad (\text{A1.5})$$

If $0 > y \geq -\frac{1}{\varepsilon}$, then $x \leq -\varepsilon$

$$F_y(y) = F_x\left(x \leq \frac{1}{y}\right) = \left[F_x\left(\frac{1}{y}\right)\right], \quad 0 > y \geq -\frac{1}{\varepsilon} \quad (\text{A1.6})$$

$$F_y(y) = \begin{cases} \left[1 - F_x\left(\frac{1}{y}\right)\right], & 0 < y \leq \frac{1}{\varepsilon} \\ F_x\left(\frac{1}{y}\right), & 0 > y \geq -\frac{1}{\varepsilon} \\ F_x\left(\frac{1}{\varepsilon}\right) - F_x\left(-\frac{1}{\varepsilon}\right), & |y| > \frac{1}{\varepsilon} \end{cases} \quad (\text{A1.7})$$

$$y = g(x) = \frac{1}{x}, \quad g'(x) = -\frac{1}{x^2}, \quad g'\left(\frac{1}{y}\right) = -y^2 \quad (\text{A1.8})$$

$$f_y(y) = \begin{cases} \frac{1}{\left|g'\left(\frac{1}{y}\right)\right|} f_x\left(\frac{1}{y}\right) = \frac{1}{y^2} f_x\left(\frac{1}{y}\right), & -\frac{1}{\varepsilon} \leq y \leq \frac{1}{\varepsilon} \\ 0, & |y| > \frac{1}{\varepsilon} \end{cases} \quad (\text{A1.9})$$

where

$$\frac{1}{y^2} f_x\left(\frac{1}{y}\right) = \frac{1}{y^2} \frac{1}{\sqrt{2\pi\sigma_x^2}} \exp\left(-\frac{\mu_x^2 \left(y - \frac{1}{\mu_x}\right)^2}{y^2 2\sigma_x^2}\right). \quad (\text{A1.10})$$

Appendix II SOLUTION OF THE INTEGRATION TERMS IN EQUATION (4.6)

Let

$$I = \int_0^T e^{-\frac{\theta}{\tau}} \cos[\omega_e(k)\theta] d\theta, \quad (\text{A2.1a})$$

$$J = \int_0^T e^{-\frac{\theta}{\tau}} \sin[\omega_e(k)\theta] d\theta. \quad (\text{A2.1b})$$

These two terms can be solved by the following manipulation:

$$\begin{aligned} I &= \frac{1}{\omega_e(k)} \int_0^T e^{-\frac{\theta}{\tau}} d \sin[\omega_e(k)\theta] \\ &= \frac{1}{\omega_e(k)} \left\{ e^{-\frac{\theta}{\tau}} \sin[\omega_e(k)\theta] \Big|_0^T - \int_0^T \sin[\omega_e(k)\theta] d e^{-\frac{\theta}{\tau}} \right\} \\ &= \frac{1}{\omega_e(k)} \left\{ e^{-\frac{T}{\tau}} \sin[\omega_e(k)T] + \frac{1}{\tau} \int_0^T e^{-\frac{\theta}{\tau}} \sin[\omega_e(k)\theta] d\theta \right\} \\ &= \frac{1}{\omega_e(k)} \left\{ e^{-\frac{T}{\tau}} \sin[\omega_e(k)T] + \frac{1}{\tau} \int_0^T e^{-\frac{\theta}{\tau}} \sin[\omega_e(k)\theta] d\theta \right\} \\ &= \frac{1}{\omega_e(k)} \left\{ e^{-\frac{T}{\tau}} \sin[\omega_e(k)T] + \frac{1}{\tau} J \right\}, \end{aligned}$$

or

$$I - \frac{1}{\omega_e(k)\tau} J = \frac{1}{\omega_e(k)} e^{-\frac{T}{\tau}} \sin[\omega_e(k)T]. \quad (\text{A2.2})$$

Similarly,

$$J = -\frac{1}{\omega_e(k)} \int_0^T e^{-\frac{\theta}{\tau}} d \cos[\omega_e(k)\theta]$$

$$\begin{aligned}
&= -\frac{1}{\omega_e(k)} \left\{ e^{-\frac{\theta}{\tau}} \cos[\omega_e(k)\theta] \Big|_0^T - \int_0^T \cos[\omega_e(k)\theta] de^{-\frac{\theta}{\tau}} \right\} \\
&= -\frac{1}{\omega_e(k)} \left\{ e^{-\frac{T}{\tau}} \cos[\omega_e(k)T] - 1 + \frac{1}{\tau} \int_0^T e^{-\frac{\theta}{\tau}} \cos[\omega_e(k)\theta] d\theta \right\} \\
&= -\frac{1}{\omega_e(k)} \left\{ e^{-\frac{T}{\tau}} \cos[\omega_e(k)T] - 1 + \frac{1}{\tau} I \right\} \\
J + \frac{1}{\omega_e(k)\tau} I &= -\frac{1}{\omega_e(k)} \left\{ e^{-\frac{T}{\tau}} \cos[\omega_e(k)T] - 1 \right\} \tag{A2.3}
\end{aligned}$$

Write (A2.2) and (A2.3) in matrix form as

$$\begin{bmatrix} \omega_e(k)\tau & -1 \\ 1 & \omega_e(k)\tau \end{bmatrix} \begin{pmatrix} I \\ J \end{pmatrix} = \begin{pmatrix} \tau \cdot e^{-\frac{T}{\tau}} \sin[\omega_e(k)T] \\ -\tau \left\{ e^{-\frac{T}{\tau}} \cos[\omega_e(k)T] - 1 \right\} \end{pmatrix}. \tag{A2.4}$$

The solution of $\begin{pmatrix} I \\ J \end{pmatrix}$ can be obtained as

$$\begin{pmatrix} I(k) \\ J(k) \end{pmatrix} = \frac{1}{[\omega_e(k)\tau]^2 + 1} \begin{bmatrix} \omega_e(k)\tau & 1 \\ -1 & \omega_e(k)\tau \end{bmatrix} \begin{pmatrix} \tau \cdot \exp\left(-\frac{T}{\tau}\right) \sin[\omega_e(k)T] \\ -\tau \left\{ \exp\left(-\frac{T}{\tau}\right) \cos[\omega_e(k)T] - 1 \right\} \end{pmatrix}. \tag{A2.5}$$

Appendix III CONTINUOUS EXTENDED KALMAN FILTER FOR BLDC

MOTORS

A continuous version of the extended Kalman filter for the BLDC motor parameter identification problem is derived below.

The BLDC motor dynamics model can be reformulated with the actuator perturbation w_1, w_2 :

$$\begin{pmatrix} \frac{di_q}{dt} \\ \frac{di_d}{dt} \end{pmatrix} = \begin{pmatrix} -\frac{1}{\tau(t)} & -\omega_e(t) \\ \omega_e(t) & -\frac{1}{\tau(t)} \end{pmatrix} \begin{pmatrix} i_q \\ i_d \end{pmatrix} + \frac{1}{L} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} V_q(t) - K_e(t)\omega_m(t) \\ V_d(t) \end{pmatrix} + \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} w_1(t) \\ w_2(t) \end{pmatrix}, \quad (\text{A3.1a})$$

Assuming parameters K_e and τ are constants subjected to random perturbation w_3 and w_4 , the dynamics of the parameters can be modeled as

$$\frac{d\tau}{dt} = w_3(t) \quad (\text{A3.1b})$$

$$\frac{dK_e}{dt} = w_4(t) \quad (\text{A3.1c})$$

Let the state $\mathbf{x} = (i_q, i_d, K_e, \tau)^T$, then the augmented state equation becomes

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{f}(\mathbf{x}, \mathbf{u}, \mathbf{w}) \\ &= \begin{bmatrix} -\frac{1}{\tau(t)} & -\omega_e(t) & \frac{i_q}{\tau^2(t)} & \frac{1}{L}\omega_m(t) \\ \omega_e(t) & -\frac{1}{\tau(t)} & \frac{i_q}{\tau^2(t)} & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \mathbf{x}(t) + \begin{bmatrix} \frac{1}{L} & 0 \\ 0 & \frac{1}{L} \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{pmatrix} V_q(t) - K_e(t)\omega_m(t) \\ V_d(t) \end{pmatrix} + \begin{pmatrix} w_1(t) \\ w_2(t) \\ w_3(t) \\ w_4(t) \end{pmatrix} \end{aligned}$$

(A3.2)

and the measurement with measurement noise $\mathbf{v} = (v_1, v_2)^T$ becomes

$$\mathbf{y}(t) = \begin{pmatrix} i_q(k) \\ i_d(k) \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \cdot \mathbf{x}(t) + \begin{pmatrix} v_1(t) \\ v_2(t) \end{pmatrix}. \quad (\text{A3.3})$$

It is usually reasonable to assume that the plant perturbation $\mathbf{w} = (w_1, w_2, w_3, w_4)^T$ and measurement noise $\mathbf{v} = (v_1, v_2)^T$ are Gaussian and independent:

$$E[\mathbf{w}(t)\mathbf{w}^T(t + \theta)] = \Sigma_w \delta(\theta), \quad (\text{A3.4a})$$

$$E[\mathbf{v}(t)\mathbf{v}^T(t + \theta)] = \Sigma_v \delta(\theta), \quad (\text{A3.4b})$$

$$E[\mathbf{w}(t)] = 0, \quad (\text{A3.4c})$$

$$E[\mathbf{v}(t)] = 0, \quad (\text{A3.4d})$$

where $\Sigma_w = \begin{bmatrix} S_{w1} & & & 0 \\ & S_{w2} & & \\ & & S_{w3} & \\ 0 & & & S_{w4} \end{bmatrix}$ and $\Sigma_v = \begin{bmatrix} S_{v1} & 0 \\ 0 & S_{v2} \end{bmatrix}$ are the spectral density matrix

of the actuation perturbation and measurement noise, and $\delta(\theta)$ is the Dirac delta function, and $E[\mathbf{w}(t)\mathbf{w}^T(t + \theta)]$ is the mathematical expectation operation.

The linearized model can be obtained as

$$\hat{\mathbf{F}}(t) = \frac{\partial f}{\partial \mathbf{x}} = \begin{bmatrix} -\frac{1}{\tau(t)} & -\omega_e(t) & \frac{i_q}{\tau^2(t)} & \frac{1}{L}\omega_m(t) \\ \omega_e(t) & -\frac{1}{\tau(t)} & \frac{i_q}{\tau^2(t)} & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (\text{A3.5})$$

$$\hat{\mathbf{B}}_u(t) = \frac{\partial f}{\partial u} = \begin{bmatrix} \frac{1}{L} & 0 \\ 0 & \frac{1}{L} \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \quad (\text{A3.6})$$

$$\hat{\mathbf{C}}_m = \frac{\partial g}{\partial x} \Big|_{x=\hat{x}(k|k)} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, \quad (\text{A3.7})$$

With the linear approximation functions, the a priori covariance matrix is computed recursively from the a posteriori covariance matrix at previous step

$$\dot{\Sigma}_e(\mathbf{t}) = \hat{\mathbf{F}}(\mathbf{t})\Sigma_e(\mathbf{t}) + \Sigma_e(\mathbf{t})\hat{\mathbf{F}}^T(\mathbf{t}) + \Sigma_w - \Sigma_e(\mathbf{t})\mathbf{C}_m^T \Sigma_v^{-1} \mathbf{C}_m \Sigma_e(\mathbf{t}) \quad (\text{A3.8})$$

The Kalman gain can be computed as

$$\mathbf{G}(\mathbf{t}) = \Sigma_e(\mathbf{t})\mathbf{C}_m^T \Sigma_v^{-1} \quad (\text{A3.9})$$

And the state can be estimated with the measurement as

$$\dot{\hat{\mathbf{x}}}(t) = \hat{\mathbf{F}}(t)\hat{\mathbf{x}}(t) + \hat{\mathbf{B}}_u(t) \cdot \begin{pmatrix} V_q(t) - K_e(t)\omega_m(t) \\ V_d(t) \end{pmatrix} - \mathbf{G}(t)[m(t) - \hat{\mathbf{C}}_m \mathbf{x}(t)]. \quad (\text{A3.10})$$

The continuous Extended Kalman Filter consists of the equations (A3.8), (A3.9) and (A3.10).

REFERENCES

- Acarman, T., et al, (2002), “A robust controller design for drive by wire hydraulic power steering system”, Proc. 2002 Am. Control Conf., Anchorage, AK, pp. 2522–2527.
- Anwar, S., (2004), “A Parametric Model of an Eddy Current Electric Machine for Automotive Braking Applications”, IEEE Trans. Contr. Sys. Tech., Vol. 12, No. 3.
- Batzel, T. D., Lee K. Y., (2005), “Electric Propulsion With Sensorless Permanent Magnet Synchronus Motor: Implementation and Performance”, IEEE TRANSACTIONS ON ENERGY CONVERSION, Vol. 20, No. 3.
- Bolognani, S., et al, (2002), “EKF-Based Sensorless IPM Synchronus Motor Drive For Flux-Weakening-Applications”, IEEE.
- Burton A. W., (2002), “Control Objectives and Systems Analysis: Innovation Drivers for Electric Power-Assisted Steering”, Proc. American Control Conference, pp 3401 – 3406.
- Cao, C., et al, (2003), “Parameter Convergence in Nonlinearly Parameterized Systems”, IEEE Trans. Automat. Contr., Vol. 48, No. 3.
- Cauet, S., et al, (2001), “Robust Control and Stability Analysis of Linearized System with Parameter Variation: Application to Induction Motors”, Proceedings of the 40th IEEE Conference on Decision and Control, pp 2647-2650.
- Cetin, A., E., et al, (2005), “Compliant Control of Electric PowerAssisted Steering Systems”, SAE Technical Paper 2005-01-3535.

- Chen, H. C., et al, (2000), “Robust Current Control for Brushless DC Motors”, IEE Proc. Electr. Power Appl., Vol. 147, No. 6, pp 503-512.
- Chen, H., et al, (2006), “PMSM Servo Drive System for Electric Power Steering Based on Two-Degree-of-Freedom Torque Control”, International Conference of Industrial TechNology, pp 2901 – 2906.
- Chen, J., Tang P. C., (1999), “A sliding mode current control scheme for PWM brushless DC motor drives”, IEEE Transactions On Power Electronics, Vol. 14, No. 3.
- Chiang, w., Zhu, L., Patankar R., (2007), “Mean Value Engine Modeling and Validation for a 4-stroke Single Cylinder Gasoline Engine”, Trends in Applied Sciences Research, Vol. 2, No. 2.
- Choi, C., et al, (2007), “Multi-domain modeling of Electric Power Steering with PMSM Drive System”, IEMDC '07, pp 1355 – 1360.
- Choi, H., et al, (2005), “On-Center Handling Characteristics of Motor Driven Power Steering System”, Proceedings of the 2005 IEEE International Conference on Mechatronics, July.
- Dubey, G. K., (2001), Fundamentals of Electrical Drives, Alpha Science International Ltd., Pangbourne.
- Fan, J. C., Kobayashi T., (1998), “A Simple Adaptive PI Controller for Linear Systems with Constant Disturbances”, IEEE Trans. Automat. Contr., Vol. 43, No. 5.
- Forrai, A., et al, (2001), “design of robust controllers with constraints on the control signal application for Burshless D C drives”, IECON'01, pp 603-608.

- Ge, S. S., Hong, F., Lee, T. H., (2003), "Adaptive Neural Network Control of Nonlinear Systems With Unknown Time Delays", IEEE Trans. Automat. Contr., Vol. 48, No. 11.
- Gobbo, D. D., et al, (2001), "Experimental Application of Extended Kalman Filtering for Sensor Validation", IEEE Trans. Contr. Sys. Tech., Vol. 9, No. 2.
- Golub, G. H., Van Loan, C. F., (1996), Matrix Computations, 3rd edition, Johns Hopkins University Press, pp.50.
- Grewal, M. S., Andrews, A. P., (2001), Kalman Filtering: Theory and Practice Using MATLAB, 2nd edition, John Wiley & Sons.
- Guinee, R. A., Lyden, C., (2000), "A Novel Application Of The Fast Simulated Diffusion Algorithm For Dynamical Parameter Identification Of Brushless Motor Drive Systems", ISCAS 2000-IEEE International Symposium on Circuits and Systems, Geneva.
- Haque, M. E., et al, (2003), "A Sensorless Initial Rotor Position Estimation Scheme for a Direct Torque Controlled Interior Permanent Magnet Synchronous Motor Drive", IEEE TRANSACTIONS ON POWER ELECTRONICS, Vol. 18, No. 6.
- Harter, W., et al, (2000), "Future Electrical Steering Systems: Realizations with Safety Requirements", SAE Technical Paper 2000-01-0822.
- Hoang L., et al, (1994), "Analysis and implementation of a real-time predictive current controller for permanent-magnet synchronous servo drives", IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS, Vol. 41., No. 1.

- Hossain, S., Deshpande, U., (2003), "Modeling Of Practical Nonidealities In A Permanent Magnet SynchroNous Motor Drive", IEEE Industry Applications Conference, Vol. 3, pp 1941 – 1945.
- Hotzel, R., Karsenti, L., (1998), "Adaptive Tracking Strategy for a Class of Nonlinear Systems", IEEE Trans. Automat. Contr., Vol. 43, No. 9.
- Huang, B., (2001), "Detection of Abrupt Changes of Total Least Squares Models and Application in Fault Detection", IEEE Trans. Contr. Sys. Tech., Vol. 9, No. 2.
- Huang, J., (2002), "An Adaptive Compensator for a Class of Linearly Parameterized Systems", IEEE Trans. Automat. Contr., Vol. 47, No. 3.
- Iles-Klumpner, D., et al, (2005), "Electric Actuation TechNologies for Automotive Steering Systems", SAE Technical Paper 2005-01-1275.
- IoanNou, P. A., Sun, J., (1996), Robust Adaptive Control, Prentice-Hall, New Jersey.
- Isermann, R., Lachmann, K. H., Matko, D., (1991), Adaptive Control Systems, Prentice Hall, Hemel Hempstead.
- Jiang, Z.-P., Hill, D. J., (1999), "A Robust Adaptive Backstepping Scheme for Nonlinear Systems with Unmodeled Dynamics", IEEE Trans. Automat. Contr., Vol. 44, No. 9.
- Johnson, J. P., et al, (1999), "Review of Sensorless Methods for Brushless DC", IEEE.
- Khalil, H. K., (1996), Nonlinear Systems, 2nd edition, Prentice-Hall, New Jersey.
- Khorrami, F., Krishnamurthy, P., Melkote, H., (2003), Modeling and Adaptive Nonlinear Control of Electric Motors., Springer, New York.
- Kim, K. H., et al, (1994), "Parameter estimation and control for permanent magnet synchroNous motor drive using model reference adaptive technique", IEEE.

- Kim, K., et al, (1999), "A Current Control for a Permanent Magnet SynchroNous Motor with a Simple Disturbance Estimation Scheme", IEEE Trans. Contr. Sys. Tech., Vol. 7, No. 5.
- Klienau, J., Patankar, R. P., Collier, S., Chandy, A., (2003), "Feedback parameter estimation for electric machines ", US app #. 10/013932, Pub. # US 2003/0071594 A1.
- Kluger, M., Harris, J., (2007), "Fuel EcoNomy Benefits of Electric and Hydraulic Off Engine Accessories", SAE Technical Paper 2007-01-0268.
- Ko, J. S., (1998), "Asymptotically stable adaptive load torque observer for precision position control of BLDC motor", IEE Proc. Electr. Power Appl., Vol. 145, No. 4., pp 383-386.
- Kosmatopoulos, E. B., IoanNou, P. A., (2002), "Robust Switching Adaptive Control of Multi-Input Nonlinear Systems", IEEE Trans. Automat. Contr., Vol. 47, No. 4.
- Kovudhikulrungsri, L., Koseki, T., (2006), "Precise Speed Estimation From a Low-Resolution Encoder by Dual-Sampling-Rate Observer", IEEE/ASME TRANSACTIONS ON MECHATRONICS, Vol. 11, No. 6.
- Krause, P. C., (2002), Analysis of Electric Machinery and Drive Systems 2nd edition, Wiley-IEEE Press, New York.
- Krstic, M., Kanellakopoulos, I., Kokotovic, P.V., (1995), Nonlinear and Adaptive Control Design, Wiley, New York.
- Lee, C. K., Kwok, N. M., (1995), "Variable Structure Controller with Adaptive Switching Surfaces", Proceedings of the American Control Conference, Vol. 1, pp. 1033-1034.

- Lee, S., et al, (2004), “Improved Velocity Estimation for Low-Speed and Transient Regimes Using Low-Resolution Encoders”, IEEE/ASME TRANSACTIONS ON MECHATRONICS, Vol. 9, No. 3.
- Liao, T., Chien, T., (2000), “An Exponentially Stable Adaptive Friction Compensator”, IEEE Trans. Automat. Contr., Vol. 45, No.5.
- LimaNond, S., Tsakalis, K. S., (2000), “Model Reference Adaptive and Nonadaptive Control of Linear Time-Varying Plants”, IEEE Trans. Automat. Contr., Vol. 45, No. 7.
- Lin, F. J., Lin, Y. S., (1999), “A Robust PM SynchroNous Motor Drive with Adaptive Uncertainty Observer”, IEEE Transactions on Energy Conversion, Vol. 14, No. 4.
- Lin, W., Qian, C., (2002), “Adaptive Control of Nonlinearly Parameterized Systems: A Nonsmooth Feedback Framework”, IEEE Trans. Automat. Contr., Vol. 47, No. 5.
- Lin, W., Qian, C., (2002), “Adaptive Control of Nonlinearly Parameterized Systems: The Smooth Feedback Case”, IEEE Trans. Automat. Contr., Vol. 47, No. 8.
- Liu, Y., Li, X. Y., (2003), “Robust Adaptive Control of Nonlinear Systems Represented by Input–Output Models”, IEEE Trans. Automat. Contr., Vol. 48, No. 6,
- Loh, A. P., Qu, C. Y., Fong, K. F., (2003), “Adaptive Control of Discrete Time Systems With Concave/Convex Parametrizations”, IEEE Trans. Automat. Contr., Vol. 48, No. 6.
- Low T. S., et al, (1996), “Motor identity-a motor model for torque analysis and control”, IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS, Vol.43, No.2, pp 285-291.

- MariNo, R., Peresada, S., and Tomei, P., (1999), "Global Adaptive Output Feedback Control of Induction Motors with uncertain Rotor Resistance", IEEE Trans. Automat. Contr., Vol. 44, No. 5.
- MariNo, R., Tomei, P., (1999), "An Adaptive Output Feedback Control for a Class of Nonlinear Systems with Time-Varying Parameters", IEEE Trans. Automat. Contr., Vol. 44, No. 11.
- Matsui, N., Ohashi, H., (1992), "DSP-based adaptive control of a brushless motor", IEEE transactions on industry applications, Vol. 28, No. 2.
- McCann, R., et al, (1998), "Influence of Motor Drive Parameters on the Robust Stability of Electric Power steering Systems", Power Electronics in Transportation, pp 103 – 108.
- Miller, D. E., (2003), "A New Approach to Model Reference Adaptive Control", IEEE Trans. Automat. Contr., Vol. 48, No. 5.
- Milman, R., Bortoff, B. A., (1999), "Observer-Based Adaptive Control of a Variable Reluctance Motor: Experimental Results", IEEE Trans. Contr. Sys. Tech., Vol. 7, No. 5.
- Mir, S., Elbuluk, M. E., Zinger, D. S., (1998), "PI and Fuzzy Estimators for Tuning the Stator Resistance in Direct Torque Control of Induction Machines", IEEE Transactions on Power Electronics, Vol. 13, No. 2, pp. 279-287.
- Mir, S., Husain, I., Elbuluk, M. E., (1998), "Switched Reluctance Motor Modeling with On-Line Parameter Identification", IEEE Transactions on Industry Applications, Vol. 34, No. 4, pp. 776-782.

- Mohammed, O. A., et al, (2005), “Physical Modeling of PM SynchroNous Motors for Integrated Coupling With Machine Drives”, IEEE TRANSACTIONS ON MAGNETICS, Vol. 41, No. 5.
- Moseler, O., et al, (1999), “Model-Based Fault DiagNosis of an Actuator System Driven by the Brushless DC Motor”, Proceedings of American Control conference.
- Moseler, O., Isermann, R., (2000), “Application of Model-Based Fault Detection to a Brushless DC Motor”, IEEE Transactions on Industrial Electronics, Vol. 47, No. 5, pp.1015-1020.
- Narendra, K. S., Balakrishnan, J., (1997), “Adaptive Control Using Multiple Models”, IEEE Trans. Automat. Contr., Vol. 42, No. 2.
- Narendra, K. S., Xiang, C., (2000), “Adaptive Control of Discrete-Time Systems Using Multiple Models”, IEEE Trans. Automat. Contr., Vol. 45, No. 9.
- Ojo, O., et al, (2002), “Models For The Control And Simulation Of SynchroNous Type Machine Drives Under Various Fault Conditions”, Industry Applications Conference, Vol.3, pp:1533 – 1540.
- Ong, C., (1998), Dynamic Simulation of Electric Machinery Using Matlab®/Simulink, Prentice Hall, New Jersey.
- Park, T. J., Han, C. S., Lee, S. H., (2005), “Development of the electronic control unit for the rack-actuating steer-by-wire using the hardware-in-the-loop simulation system”, Mechatronics, Vol. 15, pp. 899–918.
- Parmar, M., Hung J. Y., (2004), “A Sensorless Optimal Control System for an Automotive Electric Power Assist Steering System”, IEEE Transactions On Industrial Electronics, Vol. 51, No. 2.

- Patankar R., Zhu, L., (2004), "Real-time Multiple Parameter Estimation for Voltage Controlled Brushless DC Motor Actuators", Proc. American Control Conference.
- Patankar, R., Zhu, L., (2007), "Brushless DC motor actuator health monitoring and degradation compensation via real-time multiple parameter estimation", International Journal of Automation and Control, Vol. 1, No. 1, pp. 48-63.
- Pati, Y., Krishnaprasad, P., (1994), "Rational wavelets in model reduction and system identification", Proc. IEEE Conf. on Decision and Control, Lake Buena Vista, FL.
- PatiNo, H. D., Liu D., (2000), "Neural Network-Based Model Reference Adaptive Control System", IEEE Trans. Syst. Man and Cybern., Vol. 30, No. 1.
- Pelczewski, P. M., Kunz, U. H., (1990), "The optimal control of a constrained drive system with brushless DC motor", IEEE Transactions on Industrial Electronics, Vol. 37, No. 5.
- Petrovic, V., et al, (2000), "Design and Implementation of an Adaptive Controller for Torque Ripple Minimization in PM SynchroNous Motors", IEEE Transactions On Power Electronics, Vol. 15, No. 5, pp 871 – 880.
- Pillay, P, Krishnan, R., (1989), "Modeling Simulation And Analysis Of Permanent-Magnet Motor Drives. Part I. The Permanent-Magnet SynchroNous Motor Drive", IEEE TRANSACTIONS ON INDUSTRY APPLICATIONS, Vol. 25, No. 2.
- Pillay, P, Krishnan, R., (1989), "Modeling Simulation and Analysis of Permanent-Magnet Motor Drives Part II: The Brushless DC Motor Drive", IEEE TRANSACTIONS ON INDUSTRY APPLICATIONS, Vol. 25, No. 2.

- Qu, Z., (2002), “Robust Control of Nonlinear Systems by Estimating Time Variant Uncertainties”, IEEE Trans. Automat. Contr., Vol. 47, No. 1.
- Rahman, M. A., et al, (2003), “Nonlinear Control of Interior Permanent-Magnet SynchroNous Motor”, IEEE TRANSACTIONS ON INDUSTRY APPLICATIONS, Vol. 39, No. 2.
- Rahman, M. A., Zhou P., (1996), “Analysis of brushless permanent magnet synchroNous motors”, IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS, Vol.43, No.2.
- Rahman, M. F., et al, (2003), “A Direct Torque-Controlled Interior Permanent-Magnet SynchroNous Motor Drive Without a Speed Sensor”, IEEE TRANSACTIONS ON ENERGY CONVERSION, Vol. 18, No. 1.
- Ravikanth, R., Meyn, S. P., (1999), “Bounds on Achievable Performance in the Identification and Adaptive Control of Time-Varying Systems”, IEEE Trans. Automat. Contr., Vol. 44, No. 4.
- Rovithakis, G. A., (1999), “Robustifying Nonlinear Systems Using High-Order Neural Network Controllers”, IEEE Trans. Automat. Contr., Vol. 44, No. 1.
- Rubaii, A., Kotaru, R., (2000), “Online Identification and Control of a DC Motor Using Learning Adaptation of Neural Networks”, IEEE Transactions On Industry Applications, Vol. 36, No. 3, pp 935 – 942.
- Rugh, W. J., (1996), Linear System Theory 2nd edition, Prentice-Hall, New Jersey.
- Sayed, A. H., (2001), “A Framework for State-Space Estimation with Uncertain Models”, IEEE Trans. Automat. Contr., Vol. 46, No. 7.

- Shyu, K. K., Shieh, H. J., Fu, S. S., (1998), "Model Reference Adaptive Speed Control for Induction Motor Drive Using Neural Networks", IEEE Trans. Ind. Electron., Vol. 39, No. 1, pp180-182.
- Song, J., et al, (2004), "Model development and control methodology of a new electric power steering system", Proc. Inst. Mech. Eng. D. J. Automobile Eng., Vol. 218, No. 9.
- Sozer, Y., et al, (1997), "Direct model reference adaptive control of permanent magnet brushless DC motors", Proceedings of the 1997 IEEE International Conference on Control Applications.
- Sozer, Y., Kaufman, H., Torrey, D. A., (1997), "Direct Model Reference Adaptive Control of Permanent Magnet Brushless DC Motors", Proc. The 1997 IEEE Intl. Conf. On. Contr. Appl., pp 633-638.
- Sugitani, N., et al, (1997), "Electric power steering with H-infinity control designed to obtain road information", Proc. 1997 Amer. Control Conf., Albuquerque, NM, pp. 2935–2939.
- Sureshbabu, N., Farrell J. A., (1999), "Wavelet-Based System Identification for Nonlinear Control", IEEE Trans. Automat. Contr., Vol. 44, No. 2.
- Tai, M., Hingwe, P., Tomizuka, M., (2004), "Modeling and control of steering system of heavy vehicles for automated highway systems", IEEE/ASME Trans. Mechatron., Vol. 9, No. 4, pp. 609–618.
- Takeshita, T., Matsui N., (1994), "Sensorless Brushless DC Motor Drive with EMF constant Identifier", IEEE.

- Tanaka, H., et al, (2007), “Development of Torque Controlled Active Steering with Improving the Vehicle Stability for Brushless EPS”, SAE Technical Paper 2007-01-0268.
- Tao, G., (2003), Adaptive Control Design and Analysis, John Wiley & Sons, New Jersey.
- Tao, G., Joshi, S. M., Ma, X., (2001), “Adaptive State Feedback and Tracking Control of Systems with Actuator Failures”, IEEE Trans. Automat. Contr., Vol. 46, No. 1.
- Telford, D., et al, (2005), “Online Identification of Induction Machine Electrical Parameters for Vector Control Loop Tuning”, IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS, Vol. 50, No. 2.
- Terzic, B., Jadric, M., (2001), “Design and Implementation of the Extended Kalman Filter for the Speed and Rotor Position Estimation of Brushless DC Motor”, IEEE Transactions On Industrial Electronics, Vol. 48, No. 6.
- Tian, D., Yin, G., Xie, G., (2004), “Model and H-inf robust control design for electric-power steering system”, Proc. 2004 Int. Conf. Intell. Mechatron. Autom., Chengdu, P. R. China, pp. 779–783.
- Urasaki, N., et al, (2000), “An Accurate Modeling For Permanent Magnet SynchroNous Motor Drives”, IEEE 2000.
- Wang, J.B., Liaw, C. M., (1997), “Control of induction motor drive for improving operating characteristics and dynamic response”, ASME/IEEE Mechatronics, Vol. 7, No. 7, pp. 641-661.
- Xu, H., IoanNou, P. A., (2003), “Robust Adaptive Control for a Class of MIMO Nonlinear Systems With Guaranteed Error Bounds”, IEEE Trans. Automat. Contr., Vol. 48, No. 5.

- YANG Gui-jie, et al, (2003), “Model of Permanent Magnet Inductor Type SynchroNous Motor”, IEEE.
- Zaremba, A. T., Liubakka, M. K., Stuntz, R. M., (1998), “Control and steering feel issues in the design of an electric power steering system”, Proc. 1998 Am. Control Conf., Philadelphia, PA, pp. 36–40.
- Zein I., Loren L., Forgez C., (2000), “An Extended Kalman Filter and an Appropriate Model for the Real-time Estimation of the Induction Motor Variables and Parameters”, url: http://www.utc.fr/lec/Publications/articles/2001/IASTED_MECO_Zein.pdf.
- Zhang, Q., (2002), “Adaptive Observer for Multiple-Input–Multiple-Output (MIMO) Linear Time-Varying Systems”, IEEE Trans. Automat. Contr., Vol. 47, No. 3.
- Zhang, Q., (1994), “Using wavelets in Nonparametric estimation”, Proc. 33rd IEEE Conf. Decision and Control, Lake Buena Vista, FL.
- Zhang, Q., Benveniste, A., (1992), “Wavelet networks”, IEEE Trans. Neural Networks, Vol. 3.
- Zhang, T., et al, (2000), “Adaptive Control of First-Order Systems with Nonlinear Parameterization”, IEEE Trans. Automat. Contr., Vol. 45, No. 8.
- Zhang, Y., Fidan, B., IoanNou, P. A., (2003), “Backstepping Control of Linear Time-Varying Systems With KNown and UnkNown Parameters”, IEEE Trans. Automat. Contr., Vol. 48, No. 11.
- Zhang, Y., IoanNou P. A., (2000), “A New Linear Adaptive Controller: Design Analysis and Performance”, IEEE Trans. Automat. Contr., Vol. 45, No. 5.

Zhao, J., Kanellakopoulos, I., (2002), “Active Identification for Discrete-Time Nonlinear Control—Part I: Output-Feedback Systems”, IEEE Trans. Automat. Contr., Vol. 47, No. 2.

Zhao, J., Kanellakopoulos, I., (2002), “Active Identification for Discrete-Time Nonlinear Control—Part II: Strict-Feedback Systems”, IEEE Trans. Automat. Contr., Vol. 47, No. 2.

Zhu, L., Patankar R., (2004), “Adaptive Estimation for Brushless DC Motor Actuators in Electric Power Assisted Steering”, SAE technical paper 2004-01-0761.

Zhu, L., Patankar, R., (2006), “Multiple Parameter Estimation Using Gram-Schmidt OrthoNormalization for Brushless DC Motor Actuators”, International Journal of Vehicle Autonomous Systems, Vol. 4, No. 1, pp. 88-101.