

國立交通大學

電機資訊學院 電機與控制學程

碩士論文

卡曼濾波器應用於永磁同步馬達無感測控制之研究

Sensorless Control of Permanent Magnet Synchronous Motors Using
Kalman Filter Theory

研究生：葛育中

指導教授：鄒應嶼 博士

中華民國九十三年六月

卡曼濾波器應用於永磁同步馬達無感測控制之研究
Sensorless Control of Permanent Magnet Synchronous Motors Using
Kalman Filter Theory

研究生：葛育中

Student: Yu-Chung Ko

指導教授：鄒應嶼 博士

Advisor: Dr. Ying-Yu Tzou



A Thesis

Submitted to Degree Program of Electrical Engineering Computer
Science

College of Electrical Engineering and Computer Science

National Chiao Tung University

in Partial Fulfillment of the Requirements

for the Degree of

Master of Science

in

Electrical and Control Engineering

June 2004

Hsinchu, Taiwan, Republic of China

中華民國九十三年六月

授權書

(博碩士論文)

本授權書所授權之論文為本人在 交通大學 大學(學院) 電機與控制 系所

組 九十二 學年度第 二 學期取得 碩 士學位之論文。

論文名稱： 卡曼濾波器應用於永磁同步馬達無感測控制之研究

1. 同意 不同意

本人具有著作財產權之論文全文資料，授予行政院國家科學委員會科學技術資料中心、國家圖書館及本人畢業學校圖書館，得不限地域、時間與次數以微縮、光碟或數位化等各種方式重製後散布發行或上載網路。

本論文為本人向經濟部智慧財產局申請專利的附件之一，請將全文資料延後兩年後再公開。(請註明文號:)

2. 同意 不同意

本人具有著作財產權之論文全文資料，授予教育部指定送繳之圖書館及本人畢業學校圖書館，為學術研究之目的以各種方法重製，或為上述目的再授權他人以各種方法重製，不限地域與時間，惟每人以一份為限。

上述授權內容均無須訂立讓與及授權契約書。依本授權之發行權為非專屬性發行權利。依本授權所為之收錄、重製、發行及學術研發利用均為無償。上述同意與不同意之欄位若未鈎選，本人同意視同授權。

指導教授姓名：

研究生簽名：
(親筆正楷)

學號：
(務必填寫)

日期：民國 九十三 年 月 日

2本授權書請以黑筆撰寫並影印裝訂於書名頁之次頁。

3授權第一項者，所繳的論文本將由註冊組彙總寄交國科會科學技術資料中心。

4本授權書已於民國 85 年 4 月 10 日送請內政部著作權委員會(現為經濟部智慧財產局)修正定稿。

5本案依據教育部國家圖書館 85.4.19 台(85)圖編字第 712 號函辦理。

國立交通大學

論文口試委員會審定書

本校 電機資訊學院專班 電機與控制 組 葛育中 君

所提論文

合於碩士資格水準、業經本委員會評審認可。



口試委員： _____

指導教授： _____

班主任： _____

中華民國 _____ 年 _____ 月 _____ 日

卡曼濾波器應用於永磁同步馬達無感測控制之研究

學生：葛育中

指導教授：鄒應嶼 博士

國立交通大學電機資訊學院 電機與控制學程（研究所）碩士班

摘 要

永磁同步馬達近年來廣泛應用於資訊家電、自動控制、電動車輛等各種領域，是各種馬達中成長最快速的，其原因主要是具有高效率，體積小及高扭力。在某些應用如壓縮機，環境極差無法使用位置感測器(光學編碼器或霍爾元件)及負載較固定且較不要求快速響應時，無感測器的控制器便有機會使用，可以省下許多感測器的費用，更可以增加系統的信賴度。本論文利用目前已量產且廉價(低於10塊美金)的TI TMS320LF2407A 16位元DSP及搭配愛德利科技的內插式永磁同步馬達來實現無感測控制。無感測控制的方法很多，以方波馬達來說感測其反電動勢已行之有年且市場已有現成專門的微處理器及IC，而本論文是研究此最佳估測器Kalman Filter應用於弦波馬達無感測控制之可行性。

在模擬及實驗中採用了馬達於靜止座標系的動態方程式作為extended Kalman filter (EKF)使用之模型。實驗結果顯示在EKF開路時轉子位置估測值與實際值最大誤差約為20度，而速度估測可藉由調整反電勢參數得到。實驗結果也顯示了使用TMS320LF2407A執行整個估測所花費的時間不到100us，証實此顆定點DSP足以用來實現EKF於永磁弦波同步馬達無感測控制。

Sensorless Control of Permanent Magnet Synchronous Motors Using Kalman Filter Theory

Student : Yu-Chung Ko

Advisors : Dr. Ying-Yu Tzou

Degree Program of Electrical Engineering Computer Science
National Chiao Tung University

ABSTRACT

The number of permanent magnet synchronous motors (PMSM) is constantly growing in the industrial and automotive fields annually largely owing to their high efficiency, small size and high torque. In some applications like compressors or air conditioners, using PMSM with position sensors is extremely difficult. In other applications such as the rigorous environments where the fast response is not required, the sensorless control can be applied. TI TMS320LF2407A is adapted in this investigation owing to its low cost (i.e., less than US\$10), rapid calculation (up to 40 MIPS) and the fact that it is already in mass production. The motor for the experiment comes from the AM series motor of Adlee Powertronics Co., LTD. Among the many sensorless control methods available include, the brushless DC motors (BLDCM), utilizing the zero crossing of its back EMF are the easiest and least expensive way of achieving sensorless control. In this thesis, researching the possibility of using Kalman filter with PMSM sine-wave motors is the major study issue. This investigation examines the feasibility of using Kalman filters with PMSM sine-wave motors.

The motor dynamic equations in the stationary frame have been adopted for the EKF in the simulations and experiments. The experimental results reveal that the maximum error between the estimated rotor position and actual one in the steady state is 20 degrees in EKF opened-loop mode, and the estimated speed can be derived by adjusting the back EMF parameter of the dynamic equations. The results also shows that the total program execution period is less than 100us using TMS320LF2407A, and it has revealed this fixed-point DSP is capable of implementing the sinusoidal-excited sensorless control of the PMSM using the EKF theory.

誌 謝

真是一段漫長的歲月啊! 五年的光陰中經歷了許多困難與挑戰，但一切都值得。雖然沒有完成預期的目標但從實驗的過程中倒是獲益良多。許多人告訴我幹麻虐待自己，隨隨便便混過就好了，其實怎知會有如此多的挑戰，當初只是好奇想了解罷了。試想若當初不作此決定現在可能會覺得碩士的歷練不過爾爾。回憶起來過程永遠是最甜美的。如同當兩年大頭兵，當時會覺得真不是人過的日子，但現在想起來會覺得那是一段值得一提甚至是可炫耀的回憶。

公司在我求學過程中提供了不少的幫助，尤其是實驗的設備，但實務的經驗與驗證更幫助我在實現理論時有更高的成功機會，我想這是在職生的優勢。同伴是不可或缺的一環，實驗室的栩永，光耀及振宇的熱心協助讓我銘記於心。

畫龍點睛的部份要歸功於指導老師應嶼教授了。他在口試前最後的幾個月中讓我體驗到如何做好研究，那就是你不需要一次解決所有問題，只要專注於一個部份並將其徹底研究便是一份好的論文，如果我能更早體認這點相信現在的論文會更有水準。最後感謝口試委員們的指正與建議讓我能順利完成本論文。

Table of Contents

Abstract (Chinese).....	i
Abstract (English).....	ii
Acknowledgement.....	iii
Table of Contents.....	iv
List of Tables.....	vi
List of Figures.....	vii
Abbreviation and Symbols.....	x
Chapter 1 Introduction.....	1
1.1. Research motivation.....	1
1.2. Description of experimental testing bench.....	2
1.3. The research goals and system overview.....	4
1.4. Dissertation organization.....	5
Chapter 2 Modeling and Parameters Measurement of A Permanent Magnet Synchronous Motor.....	7
2.1. Machine model in arbitrary d - q reference frame.....	7
2.2. L_d and L_q measurement.....	15
Chapter 3 Kalman Filter Estimator for A PMSM.....	16
3.1. Kalman filter on rotor reference frame for a PMSM.....	17
3.2. Kalman filter on stator reference frame for a PMSM.....	20
3.3. Selection of the time-domain machine model.....	23
Chapter 4 Simulations of EKF.....	24
4.1. The steps of the discretized EKF algorithm on stator reference frame.....	26
4.2. Matlab/simulink simulation of the EKF.....	29
4.2.1 Description of the function blocks.....	29
4.2.2 Description of the motor simulink model in stator reference frame.....	31
4.2.3 Description of the EKF simulink model in stator reference frame.....	34
4.3. Simulation results.....	37
Chapter 5 Software Implementation of Kalman Filter Theory Estimating Speed and Position.....	51
5.1. Scaling for variables.....	51
5.1.1. Scaling for current, speed and position.....	52

5.1.2. Scaling for EKF models.....	54
5.1.3. Scaling for covariance matrices and Kalman gain K	56
5.2. Software flowchart of the control algorithm.....	57
5.3. Software flow chart of EKF.....	59
5.4. Assembly language implementation.....	60
5.5. Quasi floating point method.....	62
5.6. Timing analysis of the EKF.....	64
5.7. Memory requirements of the codes.....	66
Chapter 6 Experimental Results.....	67
6.1. Method 1: Flux saturation.....	67
6.2. Method 2: Rotor alignment.....	73
6.3. Experimental results of EKF.....	75
Chapter 7 Conclusions.....	82
Appendix.....	83
References.....	84
Author Resume.....	86



List of Tables

Table 5.1 Software execution time.....	64
Table 5.2 EKF execution time.....	65
Table 5.3 Memory requirements of the codes.....	66



List of Figures

Fig. 1.1	Experimental testing bench	3
Fig. 1.2	Block diagram of the sensorless control	4
Fig. 1.3	EKF open loop test	5
Fig. 2.1	Arbitrary d - q reference frame	7
Fig. 2.2	Equivalent circuit of a PMSM	13
Fig. 2.3	Interior magnet type rotor $L_d \neq L_q$	14
Fig. 2.4	Surface mount type rotor $L_d=L_q=L_{ss}$	14
Fig. 2.5	L_d and L_q measurement	15
Fig. 3.1	Structure of EKF	16
Fig. 3.2	Rotor reference frame	17
Fig. 3.3	Stator reference frame	20
Fig. 4.1	PMSM with EKF simulink model	29
Fig. 4.2	PMSM Simulink model in stator reference frame	31
Fig. 4.3	The EKF model on stator reference frame	34
Fig. 4.4	White noises	34
Fig. 4.5	Speed reference = 200 rad/s with EKF open	37
Fig. 4.6	Speed reference = 200 rad/s with EKF closed	37
Fig. 4.7	Speed reference = 100 rad/s with EKF open	38
Fig. 4.8	Speed reference = 100 rad/s with EKF closed	38
Fig. 4.9	Speed reference = 50 rad/s with EKF open	39
Fig. 4.10	Speed reference = 50 rad/s with EKF open	39
Fig. 4.11	Speed reference	40
Fig. 4.12	Speed response and rotor position without load	40
Fig. 4.13	Speed response and rotor position with load 0.1N-m at $t = 1.5$ seconds	41
Fig. 4.14	Torque response with load 0.1N-m at $t = 1.5$ seconds	41
Fig. 4.15	Current response with load of 1.5 N-m at $t = 1.5$ seconds	42
Fig. 4.16	Speed response and rotor position using EKF	43
Fig. 4.17	Current response using EKF in no load condition	45
Fig. 4.18	Torque response without load using EKF	45
Fig. 4.19	Rotor position accumulated error and speed error	46

Fig. 4.20	Speed response and rotor position with load 0.1 N-m using EKF	47
Fig. 4.21	Torque response using EKF with load 0.1 N-m	47
Fig. 4.22	Current response using EKF with load 0.1 N-m	48
Fig. 4.23	Rotor position accumulated error and speed error using EKF with load 0.1 N-m	48
Fig. 4.24	Speed and rotor position response without load using EKF	49
Fig. 4.25	Rotor position and speed error without load using EKF	49
Fig. 5.1	Current scaling	52
Fig. 5.2	Speed scaling for a 2000 PPR encoder	53
Fig. 5.3	Position scaling of a 4 pole motor for a 2000 PPR encoder	54
Fig. 5.4	Software flowchart of the control algorithm	58
Fig. 5.5	Software flowchart of EKF	59
Fig. 5.6	Software flowchart of EKF	63
Fig. 6.1	2 poles PMSM rotor position	67
Fig. 6.2	Inductance curve	68
Fig. 6.3	Detecting rotor step 1	68
Fig. 6.4	Detecting currents	68
Fig. 6.5	Six power switches	69
Fig. 6.6	Detecting rotor step 2	69
Fig. 6.7	Detecting rotor step 3	70
Fig. 6.8	Detecting current wave forms	70
Fig. 6.9	Rotor position “5”	71
Fig. 6.10	Rotor position “1”	71
Fig. 6.11	Rotor position “3”	71
Fig. 6.12	Rotor position “2”	71
Fig. 6.13	Rotor position “6”	72
Fig. 6.14	Rotor position “4”	72
Fig. 6.15	Detecting current wave forms on DC bus	72
Fig. 6.16	Detecting current wave forms on motor phases	73
Fig. 6.17	Time scaled aligning current wave forms	74
Fig. 6.18	Aligning current wave forms	74
Fig. 6.19	Host communication software	75
Fig. 6.20	Actual rotor position	76
Fig. 6.21	Estimated rotor position	76

Fig. 6.22	Error between actual rotor position and estimated one	77
Fig. 6.23	Enlarged wave form of Fig. 6.22.....	78
Fig. 6.24	Actual $\sin(\theta)$	79
Fig. 6.25	Estimated $\sin(\theta)$	79
Fig. 6.26	Error between actual $\sin(\theta)$ and estimated one	79
Fig. 6.27	Estimated speed $\omega_e = 550$ RPM. $Q(3, 3) = 0.06$	79
Fig. 6.28	Estimated speed $\omega_e = 550$ RPM. $Q(3, 3) = 2$	79
Fig. 6.29	Actual I_α	80
Fig. 6.30	Estimated I_α	80
Fig. 6.31	Actual I_α – estimated one	80
Fig. 6.32	Theta error after ω_e direction has been corrected.....	80



Abbreviation and Symbols

ω	Angular frequency of q or d -axis
ω_r	Angular frequency of rotor
θ	Angle between q -axis and stator a axis
θ_r	Angle between rotor a axis and stator a axis
p	Differential operator, $\frac{d}{dt}$
V_s^{abc}	Stator three phase voltages
V_s^{qd0}	Stator voltages on d - q reference frame
$V_s^{\alpha\beta}$	Stator voltages on $\alpha\beta$ reference frame
λ_s^{abc}	Stator three phase flux linkage
λ_s^{qd0}	Stator flux linkage on d - q reference frame
L_{ss}^{abc}	Stator three phase self inductance
L_{sr}^{abc}	Mutual inductance between stator and rotor
L_{ls}	Leakage inductance
L_s	Stator magnetizing inductance
r_s^{abc}	Stator three phase resistance
r_r^{abc}	Rotor three phase resistance
i_s^{abc}	Stator three phase current
$i_s^{\alpha\beta}$	Stator current on $\alpha\beta$ reference frame
i_r^{abc}	Three phase rotor current
$\theta_{e_{k k-1}}$	The estimated k th θ due to the $k-1$ stage
$\omega_{e_{k k-1}}$	The estimated k th ω due to the $k-1$ stage

Chapter 1

Introduction

In this thesis, the Kalman filter theory is used to estimate the rotor position and speed of a sinusoidal-excited permanent magnet synchronous motor (PMSM) instead of a real position sensor such as incremental encoder or Hall sensor. A sine wave synchronous motor with interior-magnet-rotor-type motor is chosen for the experiment and the developed sensorless PMSM control scheme has been realized by using a single-chip fixed-point DSP controller TMS320F2407A. This thesis shows that although Kalman filter theory is complicated in mathematical manipulation and time-consuming in calculation for the sensorless estimation of a PMSM motor, however, it is still feasible to be implemented by using a single-chip fixed-point DSP with 40 MIPS execution speed. This thesis focuses on the design of an extended Kalman filter in estimation of the rotor position of a PMSM motor and realization issues of the Kalman filter based sensorless control scheme.

1.1. Research motivation

Energy conservation is becoming increasingly important. Motor power consumption occupies a large part of industrial energy use, explaining why most industrialized countries must establish strict laws to save energies. In the automotive field, increasing numbers of motors are used in vehicles for mechanisms, like water pumps, steering, brake systems. Important reasons for this practice include the need to be environmentally friendly by increasing the ratio of distance per liter of fuel and the desire to keep costs down.

From above, a permanent magnet synchronous motor (PMSM) is the best choice for the application. Although some motor performance may be lost, sensorless control remains practical for applications such as air conditioners, refrigerators, fans and water pumps. While most sensorless control relies heavily on computation, in this decade digital signal processors have made it possible at low cost option. The cheapest 16-bit DSPs are the series of TI's TMS320LF240x which deliver a good performance up to 40 MIPS. This thesis proposes a way of using TMS320LF2407A to achieve sensorless vector control for PMSMs.

Bolognani, Oboe and Zigliotto [1] demonstrated that the extended Kalman filter (EKF)

can function efficiently for the sensorless control of a PMSM using the floating-point DSP, WEDSP-32C. The advantages are that the phase voltage transducers are eliminated, and thus simplifying the drive and reducing the cost. In [2], the Texas Instrument (TI) reveals how the EKF works for sensorless vector control of an induction machine. The EKF appears to be a universal solution for a sensorless control of any motor just to modify the motor state equations. According to Tatematsu, Hamad and Uchida [3], this method requires all the information from the PMSM including rotor resistances and d - q inductances with measured phase voltages and currents to estimate speed and position, which could be heavily affected due to the motor parameters and measuring noises especially at low speed. A reduced order observer which uses two special inputs to linearize the non-linear model of a PMSM is introduced [4]. The load torque is set to zero for simplification and no load information is mentioned for the experiment.

Extensive applications have been found by using the EKF algorithms in estimation of rotor position and speed in various sensorless motor drives [5]-[8]. As well known the Kalman filter is a recursive optimum-state estimator which means that as long as the required mathematical model is available, the optimum solution can be achieved. The modeling and parameters measurement techniques of sinusoidal PM synchronous motors can be found in [9]-[14], therefore, the state equations can be derived without difficulty, and this is a key step in employing EKF theory in sensorless motor control.

1.2. Description of experimental testing bench

RS232 interface has been established to communicate with a personal computer to get inputs like speed, torque, PI gain, covariance matrices for the EKF and aligning the rotor. RS232 can also output motor and drive status like speed response, current response and software parameters which makes debugging and tuning easier. The power source is a transformer with adjustable output and the motor for testing is attached with a 2000 pulse incremental encoder. Before the EKF is applied, the field-oriented control (FOC) should be rigorously tested with a real encoder. Another motor is used as a resistive load instead of a servo motor to reduce the noises. JTAG is used for downloading the codes and debugging. The taco generator can be used to observe the speed response in real time with 30V/1000 rpm.

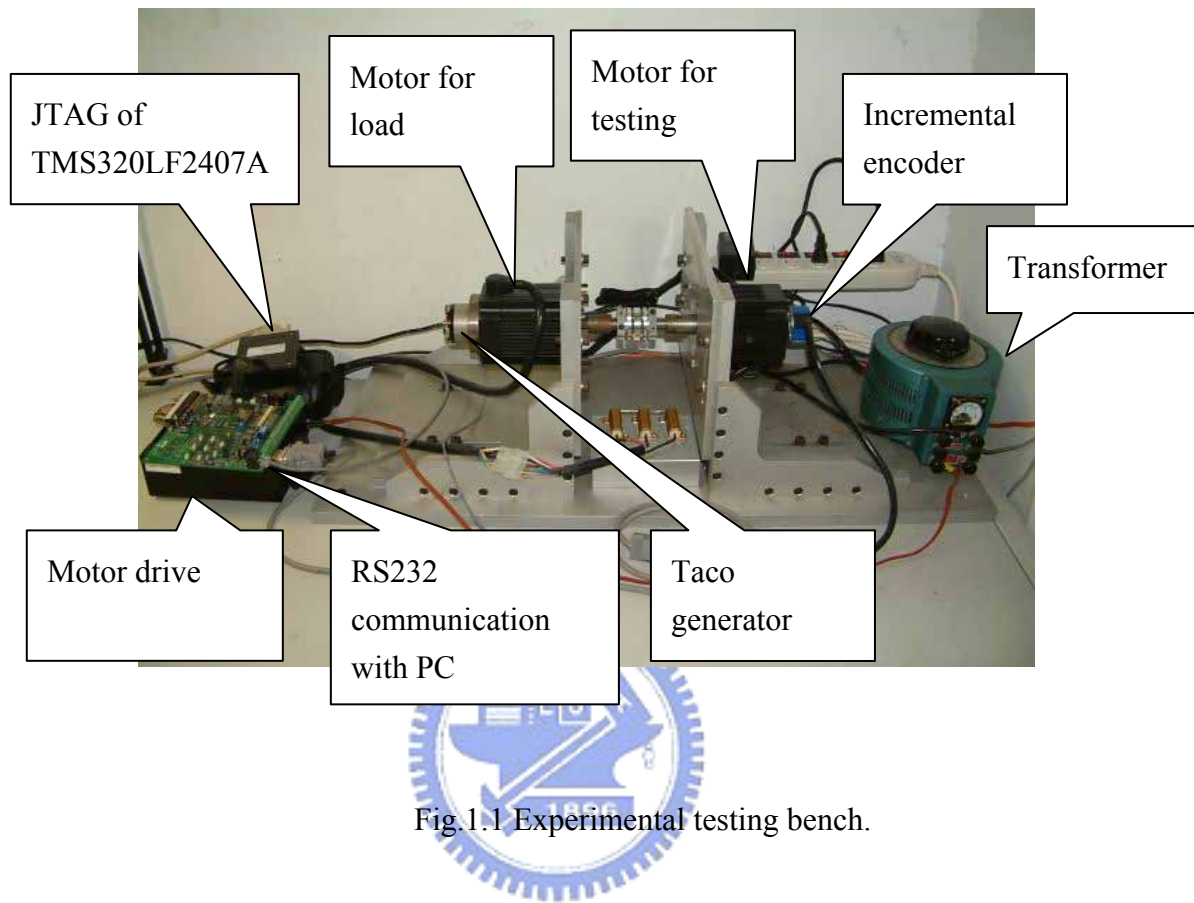


Fig.1.1 Experimental testing bench.

1.3. The research goals and system overview

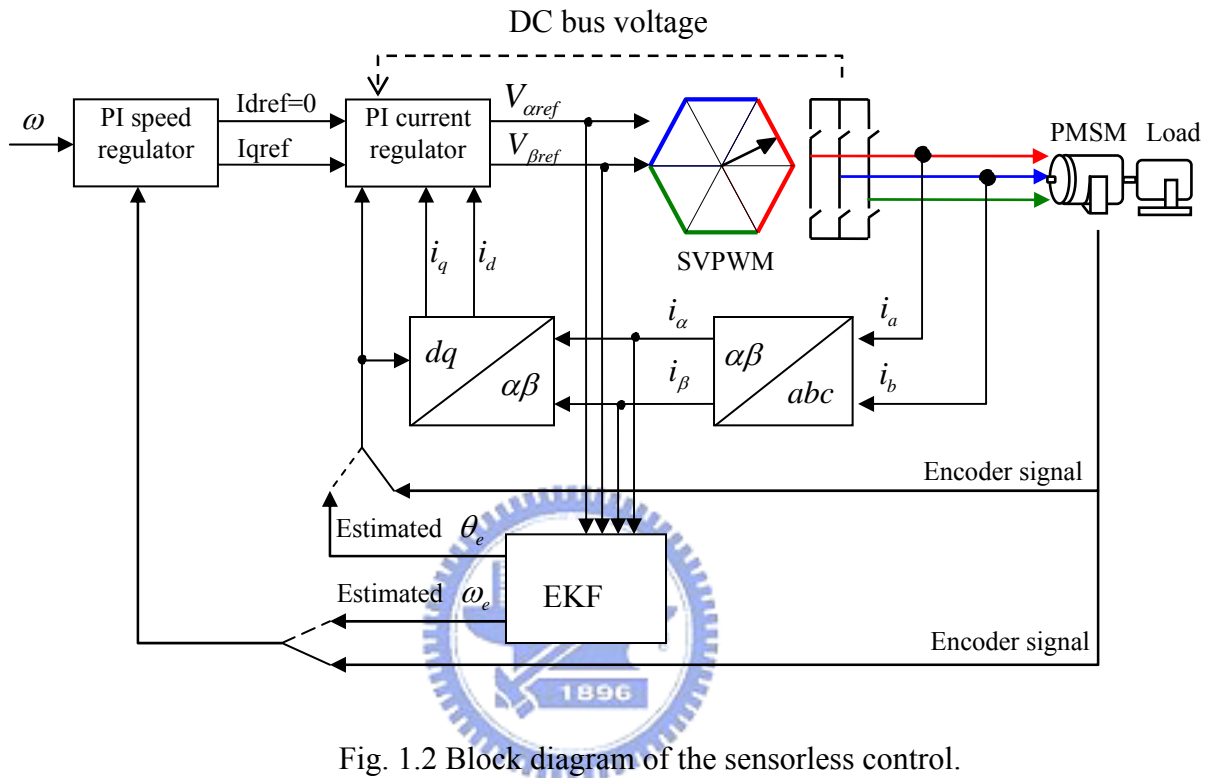


Fig. 1.2 Block diagram of the sensorless control.

The EKF is based on field-oriented control as indicated in Fig. 1.2. I_{dref} is preset to zero for the maximum torque output. There are four inputs for the EKF, that are voltage and current feedbacks expressed on $\alpha\beta$ frame. $V_{\alpha ref}$ and $V_{\beta ref}$ are used instead of the real feedbacks. In fact, voltage measurements are affected by the modulation noise and there must be phase lag due to filtering and the PWM carrier frequency is small with respect to the electrical time constant of the motor so the reference voltage $V_{\alpha ref}$ and $V_{\beta ref}$ generated by the current regulator are used instead. The voltage transducers for measuring motor phase voltages can be eliminated as well.

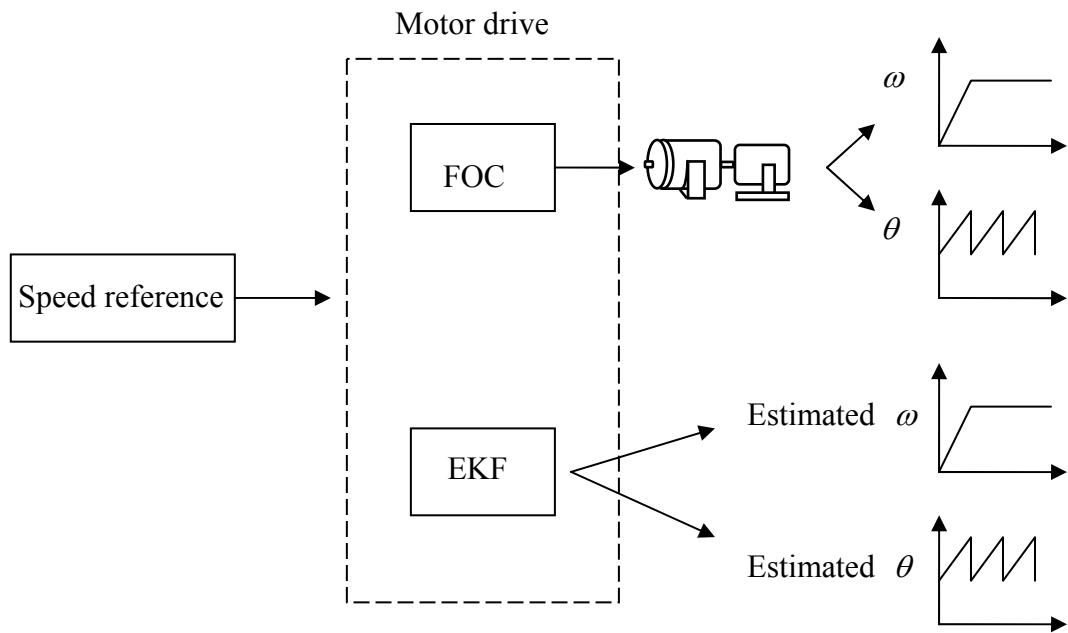


Fig. 1.3 EKF open loop test.

This research focuses on the development of a fully-digital sensorless PMSM control scheme which is feasible by using low-cost single-chip fixed-point DSP controllers. To realize the sensorless control of a PMSM using EKF, FOC with a sensor should be realized first. In FOC, the PI control algorithms, digital current samplings, Park transform, Clark transform and sine look-up table would be the main tasks. Here the fast speed response is not mainly required but to verify the FOC model. This can be done by the simulation of Matlab/simulink and experiment. Next the EKF can be included based on the FOC. In EKF algorithm, the nonlinear model of the motor, matrices calculation using a fixed-point DSP which is the most complex part would be the primary tasks. An open loop test as indicated in Fig.1.3 is performed first to observe the estimated speed and position. After the actual and estimated shaft speed and position have been matched, the EKF can supplant the incremental encoder to achieve the sensorless control for a PMSM.

1.4. Dissertation organization

Chapter 1 introduces the motor applications in the industrial and automotive fields

particularly the PMSM together with the advantages provided by sensorless control.

Chapter 2 derives the dynamic model of a PMSM on the arbitrary rotating $d-q$ frame. The dynamic model is easily mounted on the stationary frame via the rotating $d-q$ frame. Understanding the FOC, EKF and $d-q$ inductance measurement is of fundamental importance.

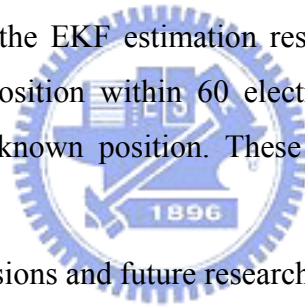
Chapter 3 introduces the EKF algorithm as applied to the parameter estimation of a PMSM. This chapter includes the nonlinear model for the EKF and the recursive steps for calculating the optimum solution.

Chapter 4 summarizes the simulation results for both FOC and the EKF using Matlab/simulink. The function blocks are illustrated in detail.

Chapter 5 illustrates the software realization using assembly language including how the macros reduce the execution time and simplify the software structures. Scaling, software flowchart and timing analysis are also demonstrated clearly.

Chapter 6 details the experiment results including the detection of the initial rotor position, rotor alignment and the EKF estimation results. The detection of the initial rotor position identifies the rotor position within 60 electrical degrees and the rotor alignment forces the rotor to stay at a known position. These two approaches help start the motor successfully.

Chapter 7 states the conclusions and future research tasks.



Chapter 2

Modeling and Parameters Measurement of A Permanent Magnet Synchronous Motor

In the control of a sinusoidal PMSM, the most commonly adopted control schemes are the field-oriented vector control (FOC) and direct torque control (DTC). These decoupling control schemes can decouple those nonlinear coupled physical variables to linear decoupled pseudo variables. A PMSM under well decoupled FOC control can behave just like an external excited dc motor, within base speed control range, the developed torque is proportional to the phase current. It decomposes complicated 3-phase stator current vectors into q -axis and d -axis, and the rotor only have d -axis (magnet axis). The Kalman filter technique will use these mathematical models to estimate the parameters of motors. This chapter examines two kinds of the motor state equations and one of them has been chosen for the EKF estimation.

2.1. Machine model in arbitrary d - q reference frame

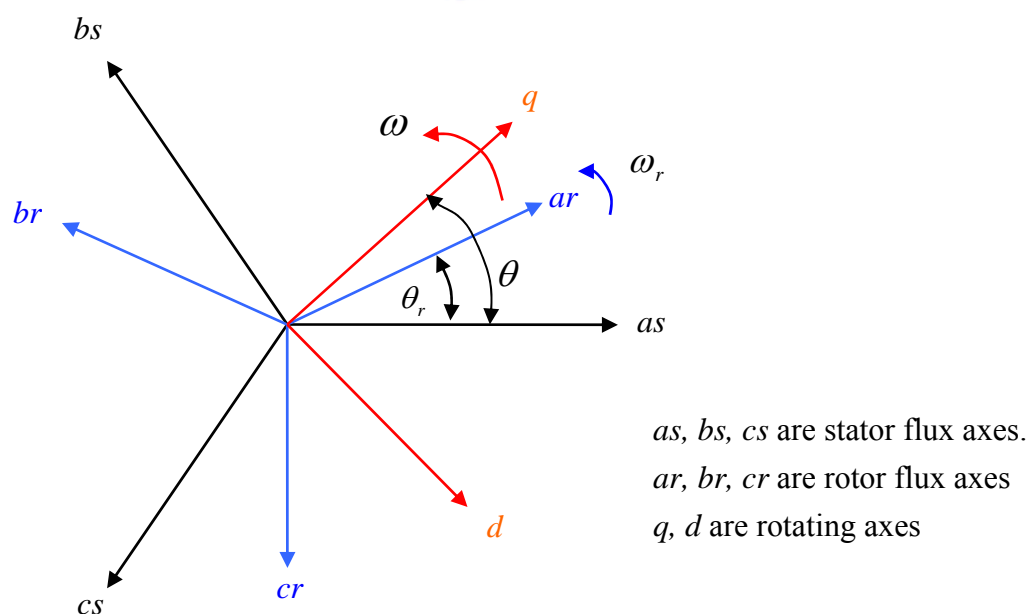


Fig. 2.1 Arbitrary d - q reference frame

To realize how the field-oriented control works, here below are the motor stator and rotor phasers. “ as ”, “ bs ” and “ cs ” mean a , b and c phase of a motor stator and “ ar ”, “ br ” and “ cr ” mean a , b and c phase of a motor rotor respectively. Each of them means the force vector caused by the stator current and rotor current (magnets). “ q ” and “ d ” axes are orthogonal and at arbitrary position. To control a PMSM as a DC motor, those phasers need to project onto the arbitrary d - q frame, that is, three-phase to two-phase transformation.

To simplify mathematical model deriving, assume that

- (1) The air gap is even,
- (2) The magnetic circuit is linear,
- (3) The stator winding is arranged to generate sinusoidal inductive voltage.

Now, project those phasers onto d - q reference frame, we need three phase to two phase transformation as below:

$$T = \frac{2}{3} \begin{bmatrix} \cos \theta & \cos\left(\theta - \frac{2}{3}\pi\right) & \cos\left(\theta + \frac{2}{3}\pi\right) \\ \sin \theta & \sin\left(\theta - \frac{2}{3}\pi\right) & \sin\left(\theta + \frac{2}{3}\pi\right) \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix}, T^{-1} = \begin{bmatrix} \cos \theta & \sin \theta & 1 \\ \cos\left(\theta - \frac{2}{3}\pi\right) & \sin\left(\theta - \frac{2}{3}\pi\right) & 1 \\ \cos\left(\theta + \frac{2}{3}\pi\right) & \sin\left(\theta + \frac{2}{3}\pi\right) & 1 \end{bmatrix}. \quad (2.1)$$

The stator phase voltage equation:

$$V_s^{abc} = p\lambda_s^{abc} + r_s^{abc}i_s^{abc}. \quad (2.2)$$

Use (2.1) to transfer abc coordinate to d - q coordinate:

$$V_s^{qd0} = T(p\lambda_s^{abc} + r_s^{abc}i_s^{abc}) = T \cdot p(T^{-1}\lambda_s^{qd0}) + T \cdot r_s^{abc}(T^{-1}i_s^{qd0}), \text{ where } p = \frac{d}{dt}, \quad (2.3)$$

$$p(T^{-1}\lambda_s^{qd0}) = pT^{-1}\lambda_s^{qd0} + T^{-1}p\lambda_s^{qd0} = \omega \begin{bmatrix} -\sin \theta & \cos \theta & 0 \\ -\sin\left(\theta - \frac{2}{3}\pi\right) & \cos\left(\theta - \frac{2}{3}\pi\right) & 0 \\ -\sin\left(\theta + \frac{2}{3}\pi\right) & \cos\left(\theta + \frac{2}{3}\pi\right) & 0 \end{bmatrix} \lambda_s^{qd0} + T^{-1}p\lambda_s^{qd0}. \quad (2.4)$$

Substituting (2.4) into (2.3), we get

$$\begin{aligned}
 V_s^{qd0} &= \frac{2}{3} \begin{bmatrix} \cos \theta & \cos\left(\theta - \frac{2}{3}\pi\right) & \cos\left(\theta + \frac{2}{3}\pi\right) \\ \sin \theta & \sin\left(\theta - \frac{2}{3}\pi\right) & \sin\left(\theta + \frac{2}{3}\pi\right) \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix} \\
 & \left(\omega \begin{bmatrix} -\sin \theta & \cos \theta & 0 \\ -\sin\left(\theta - \frac{2}{3}\pi\right) & \cos\left(\theta - \frac{2}{3}\pi\right) & 0 \\ -\sin\left(\theta + \frac{2}{3}\pi\right) & \cos\left(\theta + \frac{2}{3}\pi\right) & 0 \end{bmatrix} \lambda_s^{qd0} + T^{-1} p \lambda_s^{qd0} \right) + T r_s^{abc} T^{-1} i_s^{qd0} \\
 &= \frac{2}{3} \omega \begin{bmatrix} 0 & \frac{3}{2} & 0 \\ -\frac{3}{2} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \lambda_s^{qd0} + p \lambda_s^{qd0} + r_s^{qd0} i_s^{qd0}. \tag{2.5}
 \end{aligned}$$

We can get rotor phase voltage in d - q coordinate in similar way:

$$V_r^{qd0} = (\omega - \omega_r) \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \lambda_r^{qd0} + p \lambda_r^{qd0} + r_r^{qd0} i_r^{qd0}. \tag{2.6}$$

Stator flux linkage equation:

$$\lambda_s^{qd0} = T \left(L_{ss}^{abc} i_s^{abc} + L_{sr}^{abc} i_r^{abc} \right), \tag{2.7}$$

L_{ss} : stator phase inductance including self inductance and mutual inductance between phases

$$= L_{ls} + L_s$$

L_{sr} : mutual inductance between stator and rotor

Stator inductance:

$$L_{ss}^{abc} = \begin{bmatrix} \lambda_a \\ \lambda_b \\ \lambda_c \end{bmatrix} = \begin{bmatrix} L_{ss} & L_{sm} & L_{sm} \\ L_{sm} & L_{ss} & L_{sm} \\ L_{sm} & L_{sm} & L_{ss} \end{bmatrix} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} = \begin{bmatrix} L_{ls} + \frac{3}{2} L_s & 0 & 0 \\ 0 & L_{ls} + \frac{3}{2} L_s & 0 \\ 0 & 0 & L_{ls} + \frac{3}{2} L_s \end{bmatrix} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix}. \tag{2.8}$$

$L_{aa} = L_{bb} = L_{cc} = L_{ss}$ = stator coil self inductance

$L_{ab} = L_{ac} = L_{bc} = L_{sm}$ = mutual inductance between stators

L_{ls} = stator leakage inductance

L_s = stator magnetizing inductance

$$\lambda_a = L_{ss}i_a + L_{sm}i_b + L_{sm}i_c, \text{ where } i_a + i_b + i_c = 0, \quad (2.9)$$

$$\lambda_a = (i_b + i_c)(L_{sm} - L_{ss}) = i_a(L_{ss} - L_{sm}) = i_a\left(L_{ls} + \frac{3}{2}L_s\right). \quad (2.10)$$

Mutual inductance between stator and rotor:

$$L_{sr}^{abc} = L_{sr} \begin{bmatrix} \cos\theta_r & \cos\left(\theta_r + \frac{2}{3}\pi\right) & \cos\left(\theta_r - \frac{2}{3}\pi\right) \\ \cos\left(\frac{2}{3}\pi - \theta_r\right) & \cos\theta_r & \cos\left(\theta_r + \frac{2}{3}\pi\right) \\ \cos\left(\frac{4}{3}\pi - \theta_r\right) & \cos\left(\frac{2}{3}\pi - \theta_r\right) & \cos\theta_r \end{bmatrix}, \quad (2.11)$$

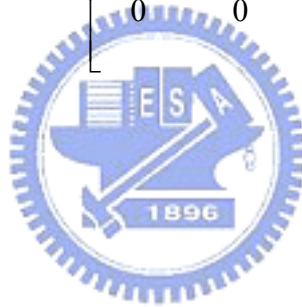
$$TL_{sr}^{abc}i_r^{abc} = T(\theta)L_{sr}^{abc}T^{-1}(\theta - \theta_r)i_r^{qd0}$$

$$= \frac{2}{3} \begin{bmatrix} \cos\theta & \cos\left(\theta - \frac{2}{3}\pi\right) & \cos\left(\theta + \frac{2}{3}\pi\right) \\ \sin\theta & \sin\left(\theta - \frac{2}{3}\pi\right) & \sin\left(\theta + \frac{2}{3}\pi\right) \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix} L_{sr}$$

$$\begin{bmatrix} \cos\theta_r & \cos\left(\theta_r + \frac{2}{3}\pi\right) & \cos\left(\theta_r - \frac{2}{3}\pi\right) \\ \cos\left(\frac{2}{3}\pi - \theta_r\right) & \cos\theta_r & \cos\left(\theta_r + \frac{2}{3}\pi\right) \\ \cos\left(\frac{2}{3}\pi + \theta_r\right) & \cos\left(\frac{2}{3}\pi - \theta_r\right) & \cos\theta_r \end{bmatrix}$$

$$\begin{bmatrix} \cos(\theta - \theta_r) & \sin(\theta - \theta_r) & 1 \\ \cos\left(\theta - \theta_r - \frac{2}{3}\pi\right) & \sin\left(\theta - \theta_r - \frac{2}{3}\pi\right) & 1 \\ \cos\left(\theta - \theta_r + \frac{2}{3}\pi\right) & \sin\left(\theta - \theta_r + \frac{2}{3}\pi\right) & 1 \end{bmatrix} i_r^{qd0}$$

$$\begin{aligned}
&= \frac{2}{3} L_{sr} \begin{bmatrix} \frac{3}{2} \cos(\theta - \theta_r) & \frac{3}{2} \cos\left(\theta - \theta_r - \frac{2}{3} \pi\right) & \frac{3}{2} \cos\left(\theta - \theta_r + \frac{2}{3} \pi\right) \\ \frac{3}{2} \sin(\theta - \theta_r) & \frac{3}{2} \sin\left(\theta - \theta_r - \frac{2}{3} \pi\right) & \frac{3}{2} \sin\left(\theta - \theta_r + \frac{2}{3} \pi\right) \\ 0 & 0 & 0 \end{bmatrix} \\
&\quad \begin{bmatrix} \cos(\theta - \theta_r) & \sin(\theta - \theta_r) & 1 \\ \cos\left(\theta - \theta_r - \frac{2}{3} \pi\right) & \sin\left(\theta - \theta_r - \frac{2}{3} \pi\right) & 1 \\ \cos\left(\theta - \theta_r + \frac{2}{3} \pi\right) & \sin\left(\theta - \theta_r + \frac{2}{3} \pi\right) & 1 \end{bmatrix} i_r^{qd0} \\
&= \begin{bmatrix} \frac{3}{2} L_{sr} & 0 & 0 \\ 0 & \frac{3}{2} L_{sr} & 0 \\ 0 & 0 & 0 \end{bmatrix} i_r^{qd0}. \tag{2.12}
\end{aligned}$$



$$\begin{aligned}
\lambda_s^{qd0} &= T(L_{ss}^{abc} i_s^{abc} + L_{sr}^{abc} i_r^{abc}) = TL_{ss}^{abc} T^{-1} i_s^{qd0} + TL_{sr}^{abc} T^{-1} (\theta - \theta_r) i_r^{qd0} \\
&= L_{ss}^{abc} i_s^{qd0} + TL_{sr}^{abc} T^{-1} (\theta - \theta_r) i_r^{qd0} \\
&= \begin{bmatrix} L_{ls} + \frac{3}{2} L_s & 0 & 0 \\ 0 & L_{ls} + \frac{3}{2} L_s & 0 \\ 0 & 0 & L_{ls} + \frac{3}{2} L_s \end{bmatrix} i_s^{qd0} + \begin{bmatrix} \frac{3}{2} L_{sr} & 0 & 0 \\ 0 & \frac{3}{2} L_{sr} & 0 \\ 0 & 0 & 0 \end{bmatrix} i_r^{qd0}. \quad (2.13)
\end{aligned}$$

It's similar to get

$$\lambda_r^{qd0} = \begin{bmatrix} L_{lr} + \frac{3}{2} L_r & 0 & 0 \\ 0 & L_{lr} + \frac{3}{2} L_r & 0 \\ 0 & 0 & L_{lr} + \frac{3}{2} L_r \end{bmatrix} i_r^{qd0} + \begin{bmatrix} \frac{3}{2} L_{sr} & 0 & 0 \\ 0 & \frac{3}{2} L_{sr} & 0 \\ 0 & 0 & 0 \end{bmatrix} i_s^{qd0}, \quad (2.14)$$

$$\begin{aligned}
V_s^{qd0} &= \begin{bmatrix} V_q \\ V_d \\ V_0 \end{bmatrix} = \omega \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \lambda_s^{qd0} + p \lambda_s^{qd0} + r_s^{qd0} i_s^{qd0} \\
&= \omega \left(\begin{bmatrix} 0 & L_{ss} & 0 \\ -L_{ss} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} i_s^{qd0} + \begin{bmatrix} \frac{3}{2} L_{sr} & 0 & 0 \\ -\frac{3}{2} L_{sr} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} i_r^{qd0} \right) + \begin{bmatrix} pL_{ss} & 0 & 0 \\ 0 & pL_{ss} & 0 \\ 0 & 0 & pL_{ss} \end{bmatrix} i_s^{qd0} \\
&+ \begin{bmatrix} \frac{3}{2} pL_{sr} & 0 & 0 \\ 0 & \frac{3}{2} pL_{sr} & 0 \\ 0 & 0 & 0 \end{bmatrix} + r_s \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} i_s^{qd0}. \quad (2.15)
\end{aligned}$$

It follows that

$$\begin{bmatrix} V_q \\ V_d \\ V_0 \end{bmatrix} = \begin{bmatrix} r_s + pL_{ss} & \omega L_{ss} & 0 \\ -\omega L_{ss} & r_s + pL_{ss} & 0 \\ 0 & 0 & r_s + pL_{ss} \end{bmatrix} \begin{bmatrix} i_s^q \\ i_s^d \\ i_s^0 \end{bmatrix} + \begin{bmatrix} \frac{3}{2} pL_{sr} & \frac{3}{2} \omega L_{sr} & 0 \\ -\frac{3}{2} \omega L_{sr} & \frac{3}{2} pL_{sr} & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} i_r^q \\ i_r^d \\ i_r^0 \end{bmatrix}. \quad (2.16)$$

For BLDC or PM motors, $i_r^q = 0$ and $p i_r^d = 0$,

$$\begin{bmatrix} V_q \\ V_d \end{bmatrix} = \begin{bmatrix} r_s + pL_{ss} & \omega L_{ss} \\ -\omega L_{ss} & r_s + pL_{ss} \end{bmatrix} \begin{bmatrix} i_s^q \\ i_s^d \end{bmatrix} + \begin{bmatrix} \frac{3}{2} \omega L_{sr} i_r^d \\ 0 \end{bmatrix} \quad (2.17)$$

or

$$\begin{bmatrix} V_q \\ V_d \end{bmatrix} = \begin{bmatrix} r_s + pL_{ss} & \omega L_{ss} \\ -\omega L_{ss} & r_s + pL_{ss} \end{bmatrix} \begin{bmatrix} i_s^q \\ i_s^d \end{bmatrix} + \omega \begin{bmatrix} \lambda \\ 0 \end{bmatrix}, \text{ where } \lambda = \frac{3}{2} L_{sr} i_r^d. \quad (2.18)$$

According to (2.18), we get an equivalent circuit as below:

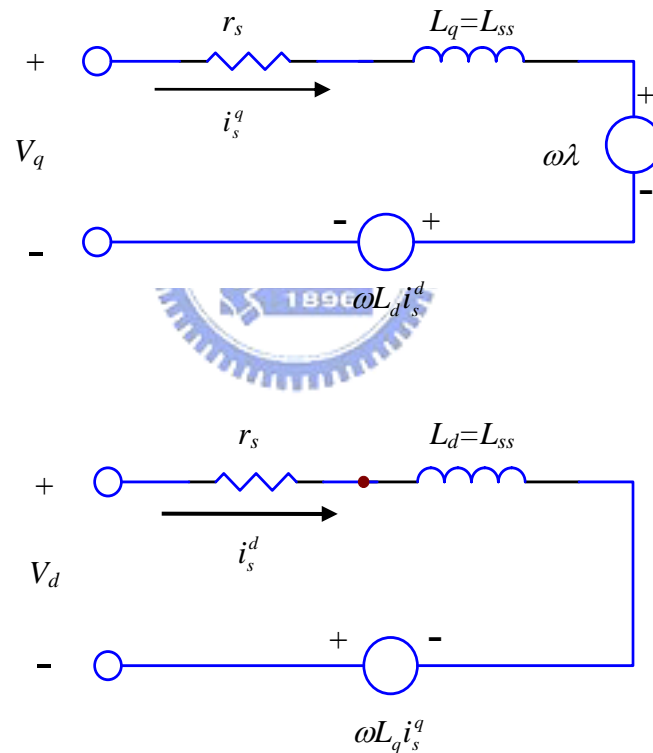


Fig. 2.2 Equivalent circuit of a PMSM.

The air gap is not even for an interior magnet synchronous motor, so L_d is not equal to L_q . See the rotor structure below.



Fig. 2.3 Interior magnet type rotor $L_d \neq L_q$.



Fig. 2.4 Surface mount type rotor $L_d = L_q = L_{ss}$.

2.2. L_d and L_q measurement

The d -axis and q -axis inductance, L_d and L_q , of a PMSM in which the magnets are buried in the rotor as shown in Fig. 2.3 can be measured with a single phase ac voltage source, see the connection below.

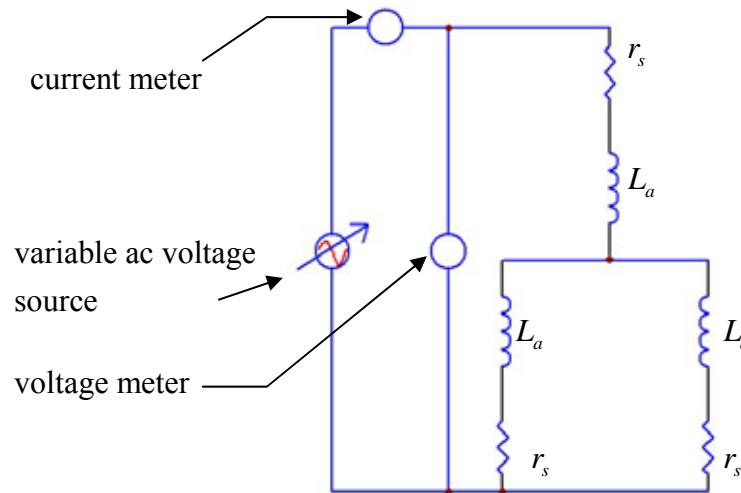


Fig. 2.5 L_d and L_q measurement.

The applied voltage and current are recorded to calculate the inductance. The d - q axis inductance can be calculated as the equation shown below by varying the applied voltage from 0 to 100 volts approximately:

$$L_d \text{ or } L_q = \frac{1}{2\pi 60} \frac{1}{i_s} \sqrt{\left(\frac{2}{3} v_s\right)^2 - (r_s i_s)^2} \quad (2.19)$$

where i_s is the measured current and v_s is the measured voltage.

Chapter 3

Kalman Filter Estimator for A PMSM

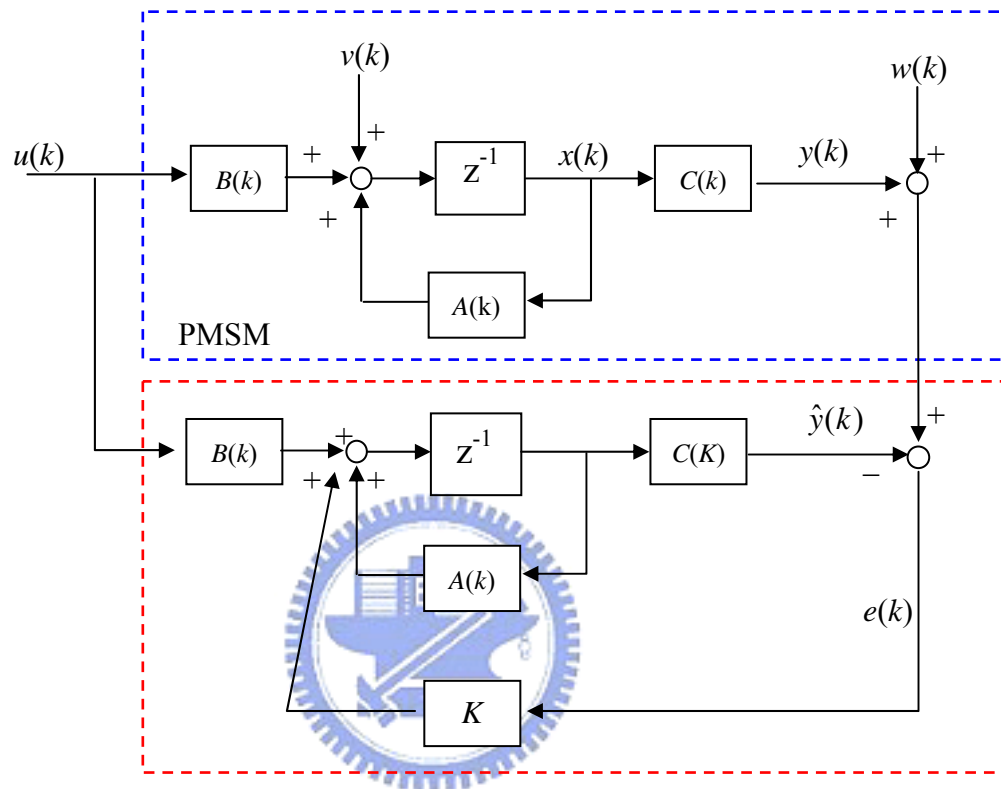


Fig. 3.1 Structure of EKF.

Fig. 3.1 shows the basic block diagram of the extended Kalman filter (EKF) [9]. It shows that the EKF is based on the state equations of a PMSM. The upper block shows the PMSM state equations which looks like the lower block except the Kalman gain K and noise vectors $v(k)$ and $w(k)$. The actual output $y(k)$ and estimated output $\hat{y}(k)$ would be the same with the same input $u(k)$ neglecting $v(k)$ and $w(k)$ which results $e(k)$ to zero. In fact, this would not happen in reality and $e(k)$ is not zero due to $v(k)$ and $w(k)$ which represent system and measurement noises respectively. In the experiment, $u(k)$ and $y(k)$ are voltage commands and motor phase currents decoupled on stationary frame respectively, see Fig. 1.2. The states

vector $x(k)$ is defined as $[i_s^\alpha, i_s^\beta, \theta_r, \omega]$ in which the unknown states, rotor position θ_r and speed ω , are to be estimated according to the known currents i_s^α and i_s^β . The Kalman gain K will try to minimize $e(k)$ to estimate the speeds and positions with the covariance matrices P , Q , R of the system state vector, system noise vector, and measurement noise vector respectively.

3.1. Kalman filter on rotor reference frame for a PMSM

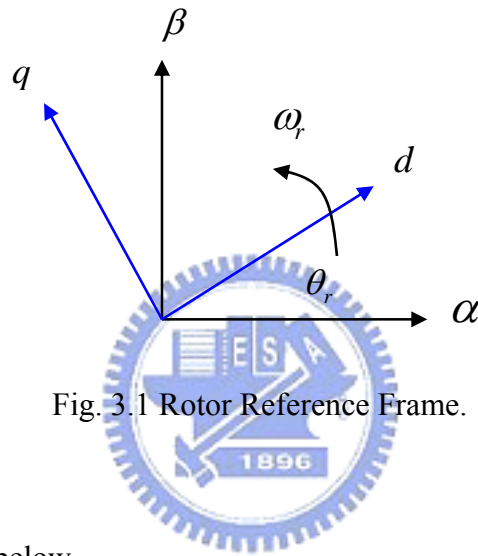


Fig. 3.1 Rotor Reference Frame.

Rewrite (2.18) as below

$$\begin{bmatrix} V_q \\ V_d \end{bmatrix} = \begin{bmatrix} r_s + pL_{ss} & \omega L_{ss} \\ -\omega L_{ss} & r_s + pL_{ss} \end{bmatrix} \begin{bmatrix} i_s^q \\ i_s^d \end{bmatrix} + \omega \begin{bmatrix} \lambda \\ 0 \end{bmatrix}. \quad (3.1)$$

The equation above is derived from an arbitrary $d-q$ reference frame. To apply EKF algorithm, (3.1) needs to be present in differential equations which the state variables should be defined first. Let the state variables be $x = [i_s^d, i_s^q, \omega_r, \theta_r]^t$, where i_s^d and i_s^q are stator current projecting to arbitrary $d-q$ frame, ω_r is rotor rotation speed and θ_r is rotor angle referring to α axis fixed on stator “a” (as) axis. The output is $y = [i_\alpha, i_\beta]^t$, where i_α and i_β are the stator current referring to $\alpha\beta$ reference frame. The input is $u = [V_\alpha, V_\beta, u_p]^t$, where V_α and V_β are stator input voltage referring to $\alpha\beta$ reference frame and $u_p = \omega\lambda$.

There are two different coordinates, $d-q$ and $\alpha\beta$ frames. It's also necessary to know the relation between these two frames:

$$\begin{bmatrix} d \\ q \end{bmatrix} = \begin{bmatrix} \cos \theta_r & \sin \theta_r \\ -\sin \theta_r & \cos \theta_r \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix}. \quad (3.2)$$

State equation forms:

$$\begin{aligned} \frac{d}{dt}x &= Ax + Bu, \\ y &= Cx. \end{aligned} \quad (3.3)$$

Represent (3.1) in state equations:

$$x = \begin{bmatrix} i_s^d \\ i_s^q \\ \omega_r \\ \theta_r \end{bmatrix}, \quad u = \begin{bmatrix} V_\alpha \\ V_\beta \\ u_p \end{bmatrix}, \quad y = \begin{bmatrix} i_\alpha \\ i_\beta \end{bmatrix}. \quad (3.4)$$

$$\begin{aligned} V_d &= r_s i_s^d + L_s \frac{d}{dt} i_s^d - \omega_r L_s i_s^q, \\ \frac{d}{dt} i_s^d &= \frac{-r_s}{L_s} i_s^d + \omega_r i_s^q + \frac{V_d}{L_s}, \end{aligned} \quad (3.5)$$

$$\begin{aligned} V_q &= r_s i_s^q + L_s \frac{d}{dt} i_s^q + \omega_r L_s i_s^d + \omega_r \lambda, \\ \frac{d}{dt} i_s^q &= -\omega_r i_s^d - \frac{r_s}{L_s} i_s^q - \frac{u_p}{L_s} + \frac{V_q}{L_s}, \end{aligned} \quad (3.6)$$

where $u_p = \omega_r \lambda$.

$$\frac{d}{dt} \begin{bmatrix} i_s^d \\ i_s^q \\ \omega_r \\ \theta_r \end{bmatrix} = \begin{bmatrix} \frac{-r_s}{L_s} & \omega_r & 0 & 0 \\ -\omega_r & -\frac{r_s}{L_s} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} i_s^d \\ i_s^q \\ \omega_r \\ \theta_r \end{bmatrix} + \begin{bmatrix} \frac{1}{L_s} & 0 & 0 \\ 0 & \frac{1}{L_s} & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} V_d \\ V_q \\ u_p \end{bmatrix}. \quad (3.7)$$

Rewriting the differential equations (3.5) and (3.6) in the form

$$\frac{d}{dt}x = Ax + Bu,$$

$$y = Cx .$$

and applying (3.2) in (3.7) yields

$$\frac{d}{dt} \begin{bmatrix} i_s^d \\ i_s^q \\ \omega_r \\ \theta_r \end{bmatrix} = \begin{bmatrix} -\frac{r_s}{L_s} & \omega_r & 0 & 0 \\ -\omega_r & -\frac{r_s}{L_s} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} i_s^d \\ i_s^q \\ \omega_r \\ \theta_r \end{bmatrix} + \begin{bmatrix} \frac{\cos \theta_r}{L_s} & \sin \theta_r & 0 \\ -\sin \theta_r & \frac{\cos \theta_r}{L_s} & -\frac{1}{L_s} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} V_\alpha \\ V_\beta \\ u_p \end{bmatrix} . \quad (3.8)$$

The output equation is

$$\begin{bmatrix} i_\alpha \\ i_\beta \end{bmatrix} = \begin{bmatrix} \cos \theta_r & -\sin \theta_r & 0 & 0 \\ \sin \theta_r & \cos \theta_r & 0 & 0 \end{bmatrix} \begin{bmatrix} i_s^d \\ i_s^q \\ \omega_r \\ \theta_r \end{bmatrix} . \quad (3.9)$$

Now the complete state equations are

$$A = \begin{bmatrix} -\frac{r_s}{L_s} & \omega_r & 0 & 0 \\ -\omega_r & -\frac{r_s}{L_s} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} , \quad (3.10)$$

$$B = \begin{bmatrix} \frac{\cos \theta_r}{L_s} & \sin \theta_r & 0 \\ -\sin \theta_r & \frac{\cos \theta_r}{L_s} & -\frac{1}{L_s} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} , \quad (3.11)$$

$$\text{and } C = \begin{bmatrix} \cos \theta_r & -\sin \theta_r & 0 & 0 \\ \sin \theta_r & \cos \theta_r & 0 & 0 \end{bmatrix} . \quad (3.12)$$

3.2. Kalman filter on stator reference frame for a PMSM

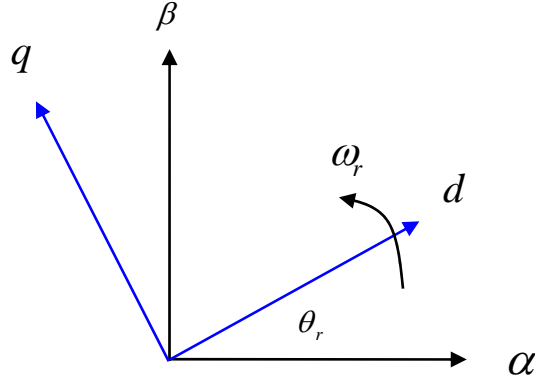
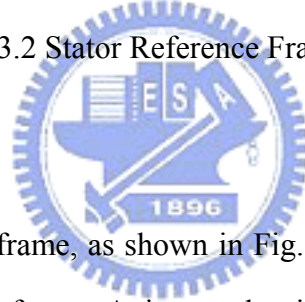


Fig. 3.2 Stator Reference Frame.



In the arbitrary rotating frame, as shown in Fig. 2.1, θ is set to 90 electrical degrees to get a fixed coordinate $\alpha\beta$ frame. Axis q and axis d are rotor quadrant and direct axes respectively. The (2.5) is presented below for quick reference ,

$$V_s^{qd0} = \frac{2}{3} \omega \begin{bmatrix} 0 & \frac{3}{2} & 0 \\ -\frac{3}{2} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \lambda_s^{qd0} + p \lambda_s^{qd0} + r_s^{qd0} i_s^{qd0} \quad (3.13)$$

where ω would be 0 in the fixed coordinate. Then (3.13) becomes

$$V_s^{\beta\alpha 0} = p \lambda_s^{\beta\alpha 0} + r_s^{\beta\alpha 0} i_s^{\beta\alpha 0}, \quad (3.14)$$

$$p \lambda_s^{\beta\alpha 0} = p(T \lambda_s^{abc}) = (pT) \lambda_s^{abc} + Tp \lambda_s^{abc} = Tp \lambda_s^{abc} = T p(L_s^{abc} i_s^{abc} + L_{sr}^{abc} i_r^{abc}), \quad (3.15)$$

$$TL_s^{abc;abc} i_s^{abc} = TL_s^{abc} T^{-1} i_s^{\beta\alpha} = L_s^{abc} i_s^{\beta\alpha}. \quad (3.16)$$

In (2.12), we replace qd with $\beta\alpha$, it becomes

$$T(L_{sr}^{abc} i_r^{abc}) = \begin{bmatrix} \frac{3}{2} L_{sr} & 0 & 0 \\ 0 & \frac{3}{2} L_{sr} & 0 \\ 0 & 0 & 0 \end{bmatrix} i_r^{\beta\alpha}, \quad (3.17)$$

$$\begin{bmatrix} \lambda_{sr}^\beta \\ \lambda_{sr}^\alpha \end{bmatrix} = \begin{bmatrix} \frac{3}{2} L_{sr} & 0 \\ 0 & \frac{3}{2} L_{sr} \end{bmatrix} \begin{bmatrix} \cos \theta_r & \sin \theta_r \\ -\sin \theta_r & \cos \theta_r \end{bmatrix} \begin{bmatrix} i_r^q \\ i_r^d \end{bmatrix}. \quad (3.18)$$

For a PMSM, we set i_r^q to 0, then (3.18) yields

$$\begin{bmatrix} \frac{3}{2} L_{sr} \cos \theta_r & \frac{3}{2} L_{sr} \sin \theta_r \\ -\frac{3}{2} L_{sr} \sin \theta_r & \frac{3}{2} L_{sr} \cos \theta_r \end{bmatrix} \begin{bmatrix} i_r^q \\ i_r^d \end{bmatrix} = \begin{bmatrix} \frac{3}{2} L_{sr} \sin \theta_r i_r^d \\ \frac{3}{2} L_{sr} \cos \theta_r i_r^d \end{bmatrix}. \quad (3.19)$$

Rewrite (3.14), we get

$$V_s^{\beta\alpha} = r_s^{\beta\alpha} i_s^{\beta\alpha} + \omega_r \begin{bmatrix} \frac{3}{2} L_{sr} \cos \theta_r i_r^d \\ -\frac{3}{2} L_{sr} \sin \theta_r i_r^d \end{bmatrix} + L_s^{abc} \frac{d}{dt} i_s^{\beta\alpha}. \quad (3.20)$$

In (3.20), the ω_r term comes from the differentiation of (3.19)

Represent (3.20) as below,

$$V_s^\beta = r_s i_s^\beta + L_s i_s^\beta + \omega_r \lambda \cos \theta_r, \quad (3.21)$$

$$V_s^\alpha = r_s i_s^\alpha + L_s i_s^\alpha - \omega_r \lambda \sin \theta_r. \quad (3.22)$$

Transfer (3.21) and (3.22) into state equation form:

$$\frac{d}{dt} i_s^\beta = -\frac{r_s}{L_s} i_s^\beta - \frac{\lambda}{L_s} \omega_r \cos \theta_r + \frac{V_s^\beta}{L_s}, \quad (3.23)$$

$$\frac{d}{dt} i_s^\alpha = -\frac{r_s}{L_s} i_s^\alpha + \frac{\lambda}{L_s} \omega_r \sin \theta_r + \frac{V_s^\alpha}{L_s}. \quad (3.24)$$

Another differential equation form, $\frac{d}{dt} x = f(x) + Bu$, is introduced with state vector $x = [i_s^\alpha \ i_s^\beta \ \omega_r \ \theta_r]^t$, input $u = [V_\alpha \ V_\beta]^t$ and output state variables $y = [i_s^\alpha \ i_s^\beta]^t$, then the complete state equations are

$$f(x) = \begin{bmatrix} -\frac{r_s}{L_s} i_s^\alpha + \frac{\lambda}{L_s} \omega_r \sin \theta_r \\ -\frac{r_s}{L_s} i_s^\beta - \frac{\lambda}{L_s} \omega_r \cos \theta_r \\ 0 \\ \omega_r \end{bmatrix}, \quad (3.25)$$

$$B = \begin{bmatrix} \frac{1}{L_s} & 0 \\ 0 & \frac{1}{L_s} \\ 0 & 0 \\ 0 & 0 \end{bmatrix}, \quad (3.26)$$

$$\text{and } C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, \quad (3.27)$$

These matrices will be discussed in detail in chapter 4.

3.3. Selection of the time-domain machine model

Comparing (3.10), (3.11), and (3.12) with (3.25), (3.26), and (3.27) we can observe some differences. It is possible to have EKF implementations using time-domain machine models expressed in the stationary or rotor reference frame. Obviously the selection of the reference frame has a great effect for the execution time in real time implementation using a DSP. In (3.26) and (3.27), the elements of these matrices are constants which can make coding easier and reduce calculation time needed, especially in view of the computational intensity of the EKF.

As discussed above, the EKF model would be based on (3.25), (3.26) and (3.27) derived from the stationary frame for the purpose of faster calculation.



Chapter 4

Simulations of EKF

The Matlab/simulink has been used throughout all the analysis, design and simulations. It's easier to simulate the EKF using Matlab language files (M files) instead of transfer functions in Matlab/simulink. Besides, the advantage of the M file is that it is easily converted into an assembly program. So the PMSM models of stator reference frame are adopted for the S-function and so are the EKF models.

As mentioned in section 3.2, the state vector, $x = [i_s^\alpha \ i_s^\beta \ \omega_r \ \theta_r]^t$, have to be estimated. The input is defined as $u = [V_\alpha \ V_\beta]^t$ and the output vector is $y = [i_s^\alpha \ i_s^\beta]^t$. The state variables form of the equations will be obtained by assuming that the rotor inertia has an infinite value (thus the derivative of the rotor speed is negligible compared with the other system variables, and any mechanical load parameter, as well as the load torque, $d\omega_r / dt = 0$) and $\omega_r = d\theta_r / dt$. Although in practice the rotor inertia is not infinite, the required correction is performed by the Kalman filter algorithm. Thus the following state-variable equation is obtained:

$$\frac{d}{dt} x = f(x) + Bu \quad (4.1)$$

and the output vector is

$$y = Cx. \quad (4.2)$$

In section 3.2., $f(x)$ is defined as:

$$f(x) = \begin{bmatrix} -\frac{r_s}{L_s} i_s^\alpha + \frac{\lambda}{L_s} \omega_r \sin \theta_r \\ -\frac{r_s}{L_s} i_s^\beta - \frac{\lambda}{L_s} \omega_r \cos \theta_r \\ 0 \\ \omega_r \end{bmatrix}. \quad (4.3)$$

Furthermore, the input matrix B is defined as

$$B = \begin{bmatrix} \frac{1}{L_s} & 0 \\ 0 & \frac{1}{L_s} \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \quad (4.4)$$

and the output transformation matrix is

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}. \quad (4.5)$$

As mentioned in section 3.3, in contrast to the transformation matrix of the model expressed in the rotor reference frame, see (3.10), the matrix as shown in (4.4) and (4.5) contains only constant elements, and it results in simplifications in the implementation of the EKF. The system described by (4.3) is a non-linear system. It is this non-linearity which ensures that the EKF has to be used, and not the conventional, linearized Kalman filter. It should be noted that the time-discrete model is now put into the following form where the noise vectors have also been added to obtain a system model required by the EKF:

$$x(k) = f[x(k), k] + B(k)u(k) + v(k), \quad (4.6)$$

$$y(k) = C(k)x(k) + w(k). \quad (4.7)$$

The covariance matrices can be chosen to be diagonal, and Q is a 4 by 4 matrix, R is a 2 by 2 matrix, and $P_0=P(0)$ is a 4 by 4 matrix. In general they are assumed to take the form

$$Q = Q_0 = \text{diag}(a, a, b, c), \quad (4.8)$$

$$P_0 = \text{diag}(e, e, f, g), \quad (4.9)$$

and

$$R = R_0 = \text{diag}(m, n). \quad (4.10)$$

The motor behavior with real position feedback and the EKF will be illustrated in simulation and experiments.

4.1. The steps of the discretized EKF algorithm on stator reference frame

The steps of the discretized EKF algorithm are as follows:

Step 1: Initialization of the state vector and covariance matrices

The initial values of the state vector $x_0=x(t_0)$ and noise covariance matrices Q_0 and R_0 are set, together with the starting value of the state covariance matrix P_0 . It is important to note that in general, if incorrect initial values are used, the EKF algorithm will not converge to the correct values. For the initial position of the PMSM, θ_r , which can be obtained using the method of aligning the rotor by applying stator current at “as” axis, then the rotor will be aligned at “as” (or α) axis. For a brushless DC motor, the rotor initial position can be obtained by examining the stator currents applied by a certain duty cycle of the output phase voltages, this method will be discussed in detail in chapter 4.

Step 2: Prediction of the state vector

Prediction of the state vector at sampling time $(k+1)$ from the input $u(k)$, the state vector at previous sampling time $x(k)$, by using f and B , is obtained by performing

$$x(k+1/k) = x(k+1) = x(k/k) + T [f_x(k/k) + Bu(k)] = x(k) + T [f_x(k) + Bu(k)]. \quad (4.11)$$

On the right side equation, the simplified notation has been used, and a simple rectangular integration technique is used by using the previous state estimate $x(k|k)$, and also the mean voltage vector $u(k)$ which is applied to the motor in the period t_k to t_{k+1} .

Step 3: Covariance estimation of prediction

The covariance matrix of prediction is estimated as

$$P(k+1/k) = P(k/k) + T[F(k)P(k+1/k) + P(k+1/k)F^T(k)] + Q, \quad (4.12)$$

where F is the gradient matrix:

$$F(k) = \frac{\partial f(x)}{\partial x}, \quad (4.13)$$

$$= \begin{bmatrix} -\frac{r_s}{L_s} & 0 & \frac{\lambda}{L_s} \sin \theta_r & \frac{\omega\lambda}{L_s} \cos \theta_r \\ 0 & -\frac{r_s}{L_s} & -\frac{\lambda}{L_s} \cos \theta_r & \frac{\omega\lambda}{L_s} \sin \theta_r \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}. \quad (4.14)$$

Step 4: Kalman filter gain computation

$$K(k+1) = P(k+1/k)C^T [CP(k+1/k)C^T + R]^{-1}. \quad (4.15)$$

Step 5: State vector estimation

The state vector estimation at time $(k+1)$ is performed through the feedback correction scheme, which makes use of the actual measured quantities (y):

$$\hat{x}(k+1/k) = x(k+1/k) + k(k+1)[y(k+1) - Cx(k+1/k)]. \quad (4.16)$$

Step 6: Covariance matrix of estimation error

$$\hat{P}(k+1/k+1) = P(k+1/k) - K(k+1)CP(k+1/k). \quad (4.17)$$

Step 7: $k=k+1$, $x(k)=x(k-1)$, $P(k)=P(k-1)$ and go to Step 1.

It should be noted that this scheme requires the measurement of the stator voltages and currents. In a PWM voltage-source inverter, it is possible to reconstruct the stator voltage according to the DC bus voltage and switching states of the power switches. If so, we need a voltage sensor to detect the DC bus voltage and connect it to A/D interface of the DSP. As we know, PWM type inverter can be a noisy source which means you will pick up noises when you measure the motor phase voltages. To avoid the noises, using some filtering is necessary

but it will lead to another problem, phase lag, and this would cause some inaccuracies and convergence problem during the calculations of EKF algorithm. Another possible way is to use stator voltage references instead of the real phase voltages, thus we can get rid of the voltage sensor and simply the software and hardware.



4.2. Matlab/simulink simulation of the EKF

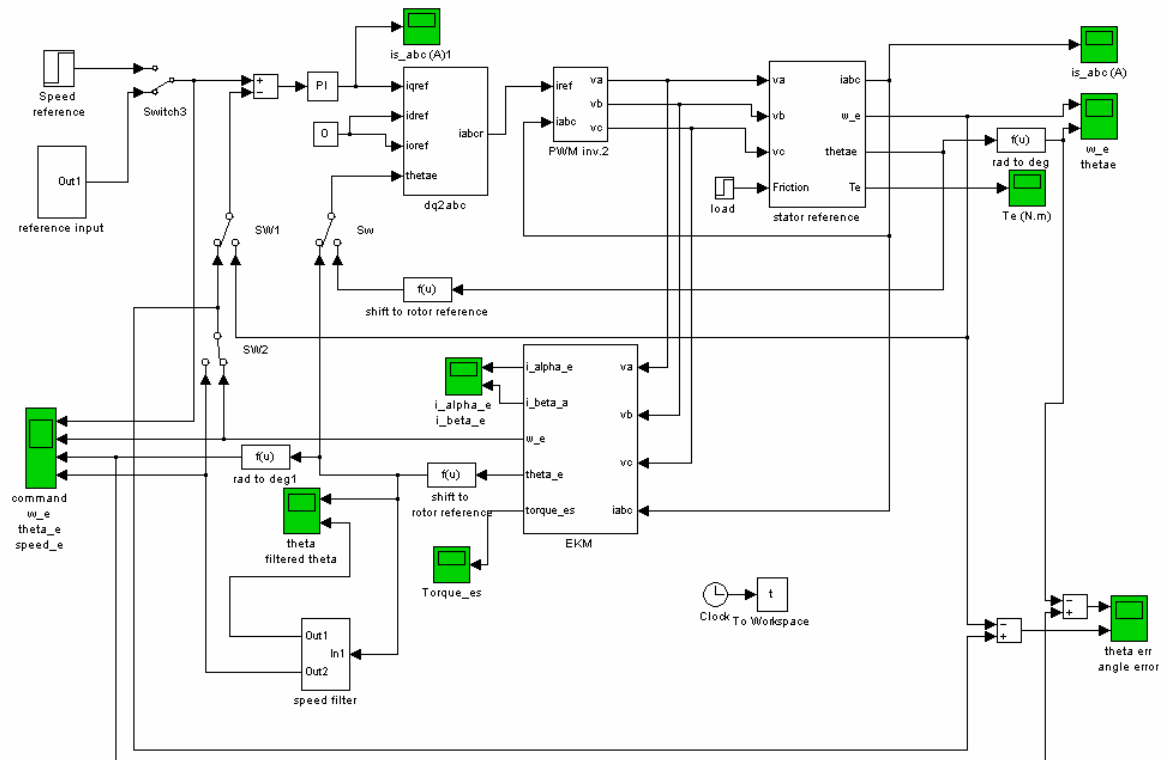


Fig. 4.1 PMSM with EKF simulink model.

4.2.1. Description of the function blocks

In Fig. 4.1, the descriptions of the blocks are as below:

dq2abc: Transform d - q current references to abc current references.

INPUT:

- iqref: q-axis current reference coming from PI speed regulator.
- idref: d-axis current reference, set to 0.
- ioref: Set to 0.
- thetai: Rotor position, it is needed for coordinates transformation.

OUTPUT:

- iabc: Motor three phase current references.

PWM inv 2: Current regulator.

INPUT:

- i_{ref} : Three phase currents reference input.
- i_{abc} : Three phase currents feedback coming from the motor.

OUTPUT:

- v_a, v_b, v_c : Motor three phase voltages.

Stator reference: Motor model in stator reference (alpha-beta), which is based on s functions.

INPUT:

- v_a, v_b, v_c : Motor three phase voltages input.
- Friction: Friction input coming from mechanical parts like bearings and loads.

OUTPUT:

- i_{abc} : Motor three phase current outputs which can be used as monitor and feedback.
- w_e : Motor speed in electric degrees.
- θ_e : Rotor position in electric degrees.
- T_e : Motor output torque.

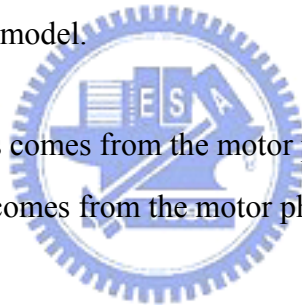
EKM: Extended Kalman filter model.

INPUT :

- v_a, v_b, v_c : The inputs comes from the motor phase voltages.
- i_a, i_b, i_c : The inputs comes from the motor phase currents.

OUTPUT :

- i_{α_e} : Estimated alpha axis current output from EKM.
- i_{β_e} : Estimated beta axis current output from EKM.
- w_e : Estimated rotor speed in electric degrees from EKM.
- θ_e : Estimated rotor position in electric degrees from EKM.
- torque_{es} : Estimated torque from EKM.



4.2.2. Description of the motor simulink model in stator reference frame

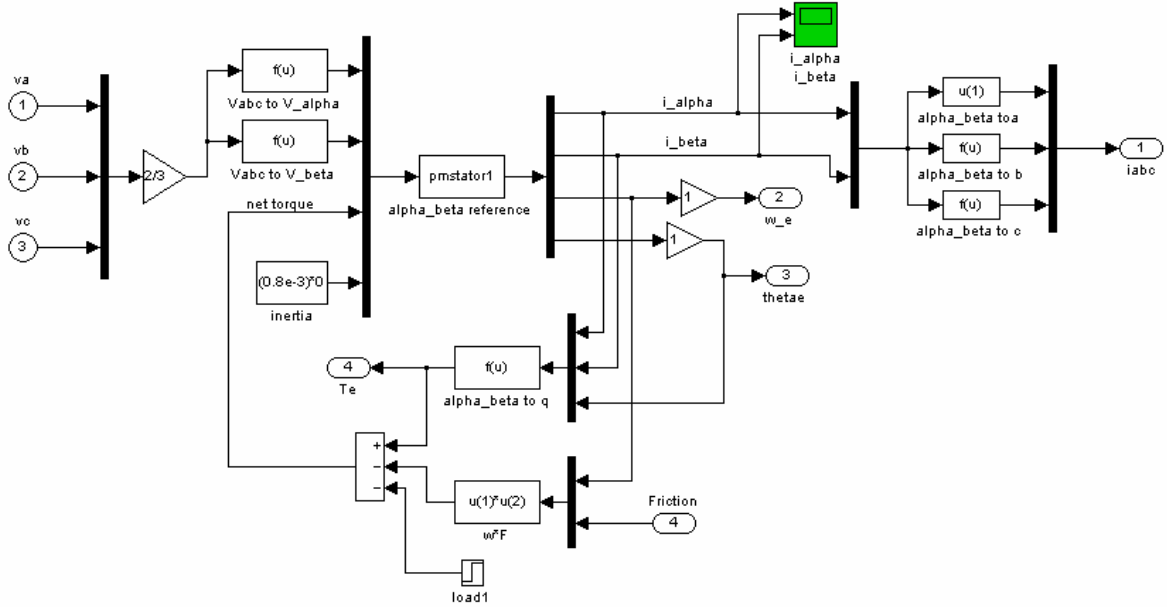


Fig. 4.2 A PMSM Simulink model in stator reference frame.

Fig. 4.2 shows the block diagram of PMSM in stator reference frame. The inputs are the motor three phase voltages like the real motor. The outputs are three phase currents, motor speed and rotor position. It's easy to simulate motor behavior using matrix Matlab/simulink S function. According to (3.23) and (3.24), the state equations for simulation in S function is shown below:

$$\frac{d}{dt} \begin{bmatrix} i_s^\alpha \\ i_s^\beta \\ \omega_r \\ \theta_r \end{bmatrix} = \begin{bmatrix} -\frac{r_s}{L_s} & 0 & -\frac{\lambda}{L_s} \sin \theta_r & 0 \\ 0 & -\frac{r_s}{L_s} & -\frac{\lambda}{L_s} \cos \theta_r & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} i_s^\alpha \\ i_s^\beta \\ \omega_r \\ \theta_r \end{bmatrix} + \begin{bmatrix} \frac{1}{L_s} & 0 & 0 \\ 0 & \frac{1}{L_s} & 0 \\ 0 & 0 & \frac{1}{J} \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} V_s^\alpha \\ V_s^\beta \\ T \end{bmatrix}, \quad (4.18)$$

where T means torque, J is the system inertia. The output torque equation is presented as below:

$$T = 1.5P(\lambda i_s^q + (L_d - L_q)i_s^d i_s^q). \quad (4.19)$$

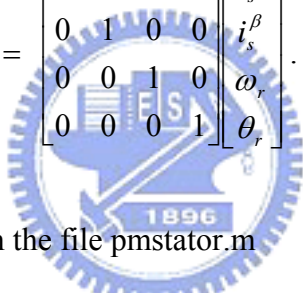
The motor model is based in the stator reference frame, so we need to convert i_s^α and i_s^β to i_s^d and i_s^q . The transfer matrix:

$$\begin{bmatrix} d \\ q \end{bmatrix} = \begin{bmatrix} \cos \theta_r & \sin \theta_r \\ -\sin \theta_r & \cos \theta_r \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix}. \quad (4.20)$$

Apply (4.20) to (4.19), it yields

$$T = 1.5P(\lambda(-i_s^\alpha \sin \theta_r + i_s^\beta \cos \theta_r) + (L_d - L_q)(i_s^\alpha \cos \theta_r + i_s^\beta \sin \theta_r) (-i_s^\alpha \sin \theta_r + i_s^\beta \cos \theta_r)). \quad (4.21)$$

The equation (4.21) will be used to calculate the output torque in the simulation. The output states variables:

$$y = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} i_s^\alpha \\ i_s^\beta \\ \omega_r \\ \theta_r \end{bmatrix}. \quad (4.22)$$


We will use (4.18) and (4.22) in the file pmstator.m

pmstator1.m file

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Motor model in stator reference frame
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [sys,x0,str,ts]=pmstator(t,x,u,flag);
Rs=5;
Ld=10e-3;
Lq=10e-3;
Fm=0.175;
J=0.8e-3;
C=[1 0 0 0;
  0 1 0 0;
  0 0 1 0;
  0 0 0 1];
switch flag
case 0,
    sizes=simsizes;
    sizes.NumContStates=4;

```

```

sizes.NumDiscStates=0;
sizes.NumOutputs=4;
sizes.NumInputs=4;
sizes.DirFeedthrough=0;
sizes.NumSampleTimes=1;
str=[];
ts=[0 0];
x0=[0; 0; 0; 0];
sys =simsizes(sizes);
case 1,
U=[u(1);u(2);u(3)];

A=[-Rs/Ld    0    Fm/Ld*sin(x(4))    0;
    0    -Rs/Lq    -Fm/Lq*cos(x(4))    0;
    0    0    0    0;
    0    0    1    0];

B=[ 1/Ld    0    0;
    0    1/Lq    0;
    0    0    1/(J+u(4));
    0    0    0 ];

sys=A*x+B*U;

case 3,
sys=C*x;

case 4,
sys=[];

case 9,
sys=[];
end

```



4.2.3. Description of the EKF simulink model in stator reference frame

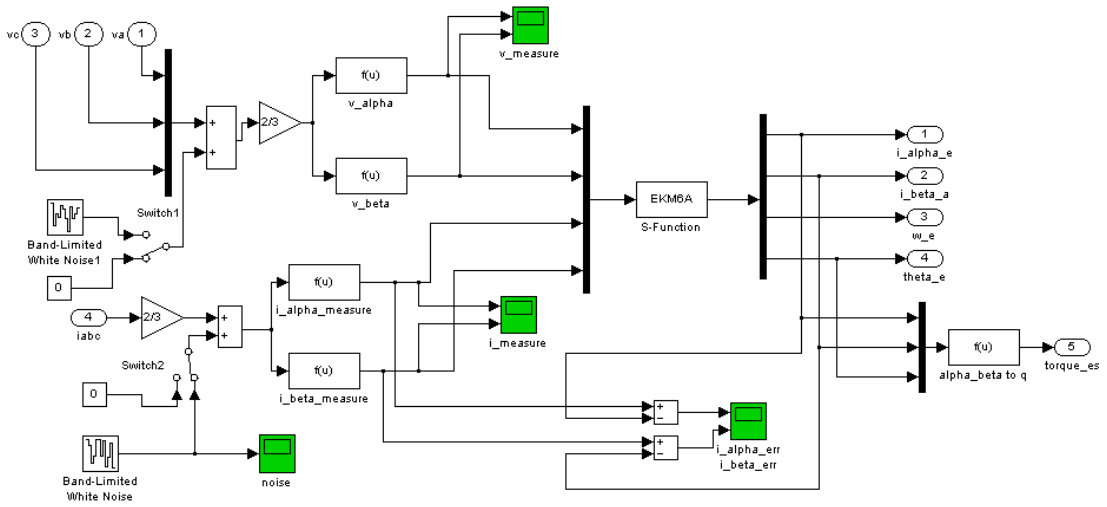


Fig. 4.3 The EKF model on stator reference frame.

Examining Fig. 4.3 finds the structure is quite similar to the model in Fig. 4.2, the input v_{α} , v_{β} and the output i_{α} , i_{β} , w_e , θ_e are the same with Fig. 4.2. The major differences are EKF needs currents feedback to calculate Kalman gain and there are white noises inputs injected to the voltages v_{α} , v_{β} and currents i_{α} , i_{β} . The white noise is appeared as Fig. 4.4.

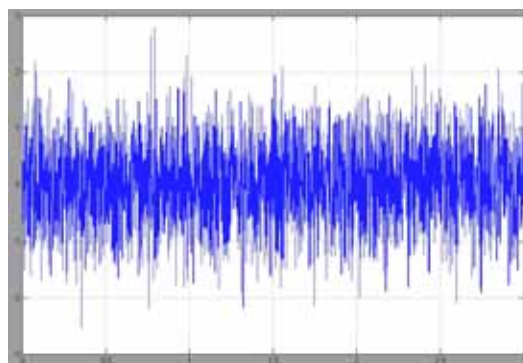


Fig. 4.4 White noises.

According to section 3.1 , we can compose Matlab language as below:

EKM6a.m file

```
function [sys,x0,str,ts]=EKM6a(t,x,u,flag,P,Q,R);
%global Rs Ld Lq Fm v_alpha v_beta i_alpha i_beta;
%global P Q R fid;
%global out fit;
Rs=5;
Ld=10e-3;
Lq=10e-3;
Fm=0.175;
T=5e-4;
%i=0;
```

```
    Bd=[1/Ld    0;
        0      1/Lq;
        0      0;
        0      0];
```

```
    H=[1 0 0 0;
        0 1 0 0];
```

```
switch flag
case 0,
    %ekmini;
    sizes=simsizes;
    sizes.NumContStates=0;
    sizes.NumDiscStates=4;
    sizes.NumOutputs=4;
    sizes.NumInputs=4;
    sizes.DirFeedthrough=0;
    sizes.NumSampleTimes=1;
    str=[];
    ts=[0.0005 0];
    x0=[0; 0; 0; 0];
    sys = simsizes(sizes);
```



case 2,

% Step 1:Initialization of the state vector and covariance matrices

```
Uk=[u(1);u(2)];
Y=[u(3);u(4)];
```

% Step 2:Prediction of the state vector

```
fx=[-Rs/Ld*x(1)+Fm/Ld*x(3)*sin(x(4)) ;
    -Rs/Lq*x(2)-Fm/Lq*x(3)*cos(x(4)) ;
```

```

0
x(3)
];

F=[-Rs/Ld    0    Fm/Ld*sin(x(4))    Fm/Ld*x(3)*cos(x(4));
0    -Rs/Lq    -Fm/Lq*cos(x(4))    Fm/Lq*x(3)*sin(x(4));
0    0    0    0;
0    0    1    0];

```

```
x1=x+T*(fx+Bd*Uk);
```

```
% Step 3: Covariance estimation of prediction
```

```
P1=P+T*(F*P+P*F')+Q; % P1 is 4x4
```

```
% Step 4: Kalman filter gain computation
```

```
K1=P1*H'*inv(H*P1*H'+R);% K1 is 4x2
```

```
% Step 5: State vector estimation
```

```
h=[x1(1);x1(2)];
sys=x1+K1*(Y-h);
```

```
% Step 6: Covariance matrix of estimation error
```

```
P=P1-K1*H*P1;
```

```
case 3,
```

```
sys=x;
```

```
case 4,
```

```
sys=(round(t/T)+1)*T;
```

```
%sys=[];
```

```
case 9,
```

```
sys=[];
```

```
end
```



4.3. Simulation results

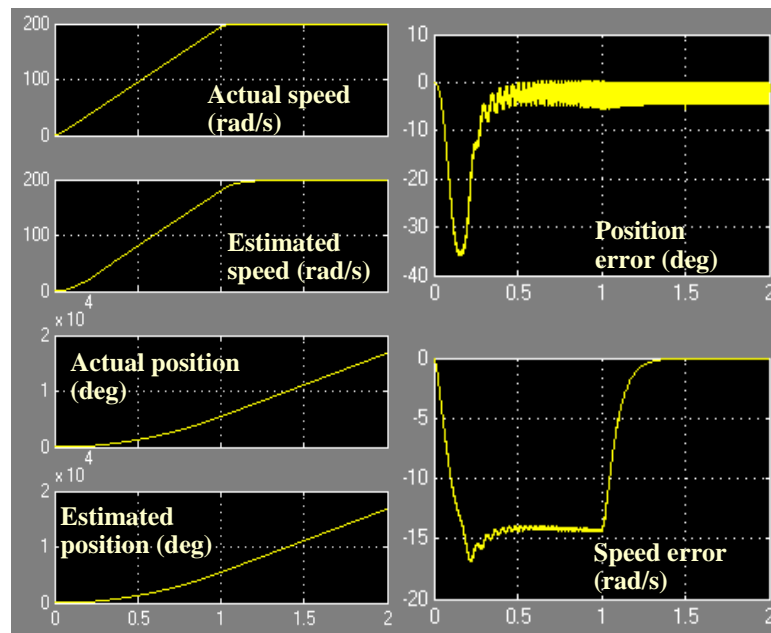


Fig. 4.5 Speed reference = 200 rad/s with EKF open.

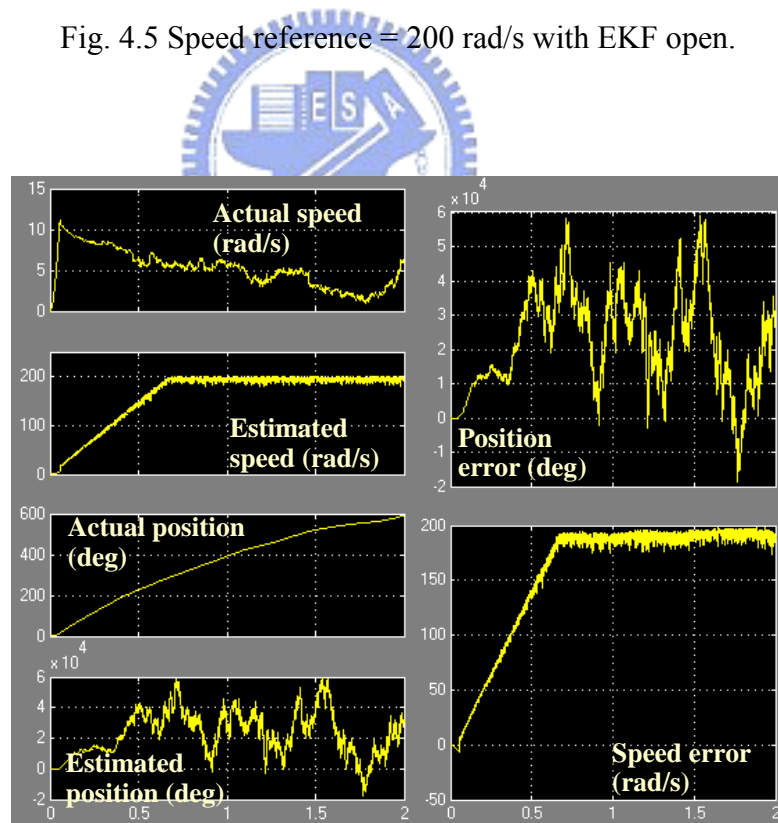


Fig. 4.6 Speed reference = 200 rad/s with EKF closed.

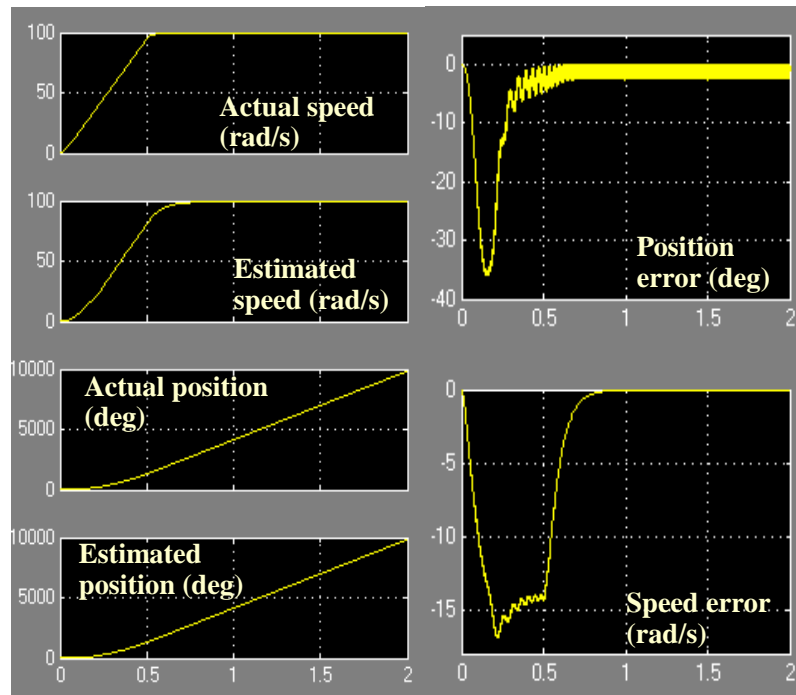


Fig. 4.7 Speed reference = 100 rad/s with EKF open.

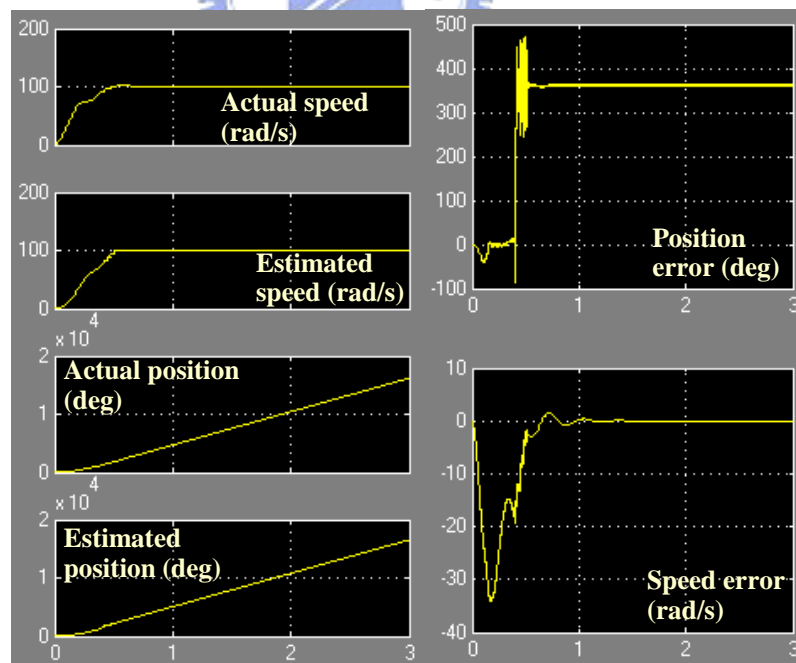


Fig. 4.8 Speed reference = 100 rad/s with EKF closed.

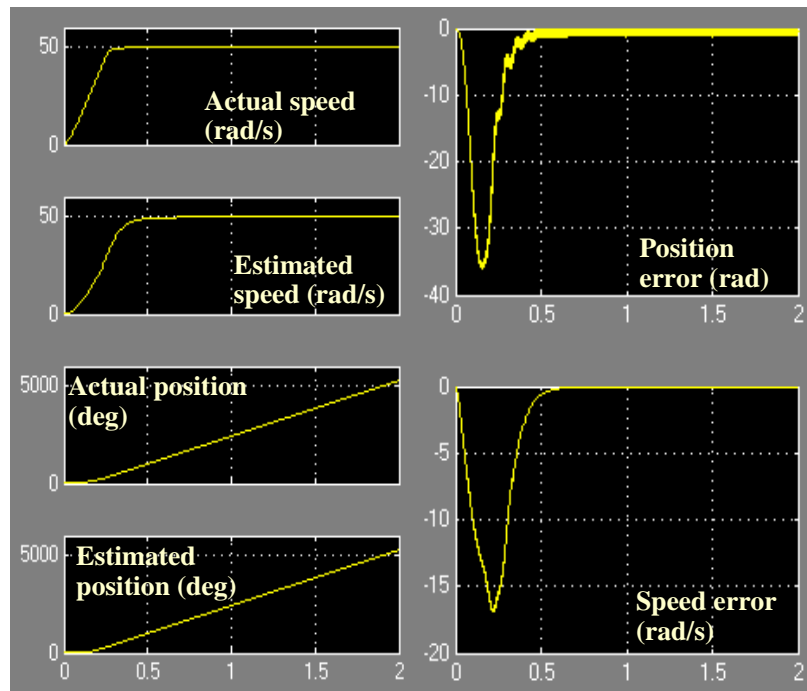


Fig. 4.9 Speed reference = 50 rad/s with EKF open.

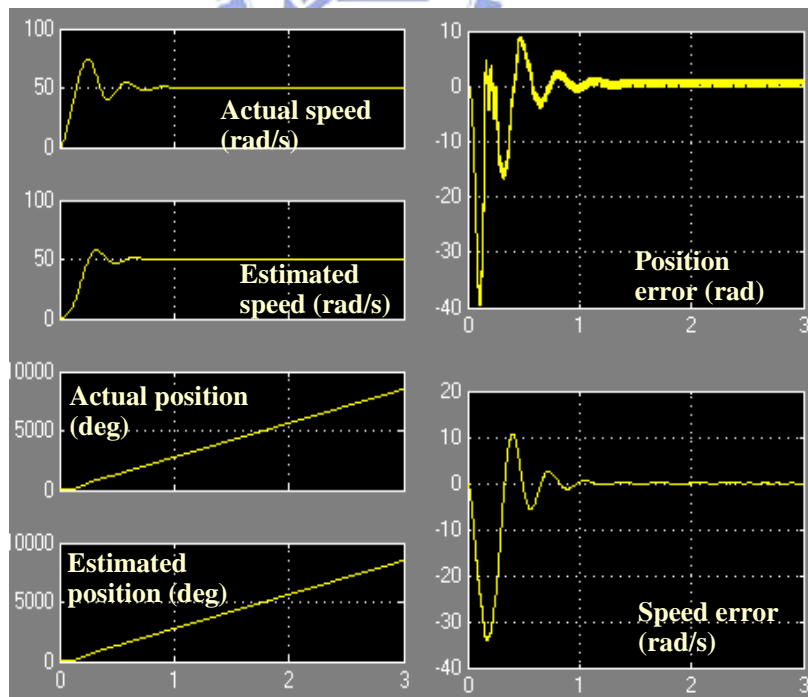


Fig. 4.10 Speed reference = 50 rad/s with EKF closed.

Fig. 4.5 to Fig. 4.10 show the results with EKF open and closed where the input commands come from 50 to 200 rad/s with a ramp of 200 rad/s at the beginning. The speeds and rotor positions can be easily estimated with EKF open for speed references from 50 to 200 rad/s. With EKF closed, which means the encoder feedbacks are replaced with the estimated speeds and positions by EKF, the EKF failed at a reference command of 200 rad/s. Another speed reference is defined as below:

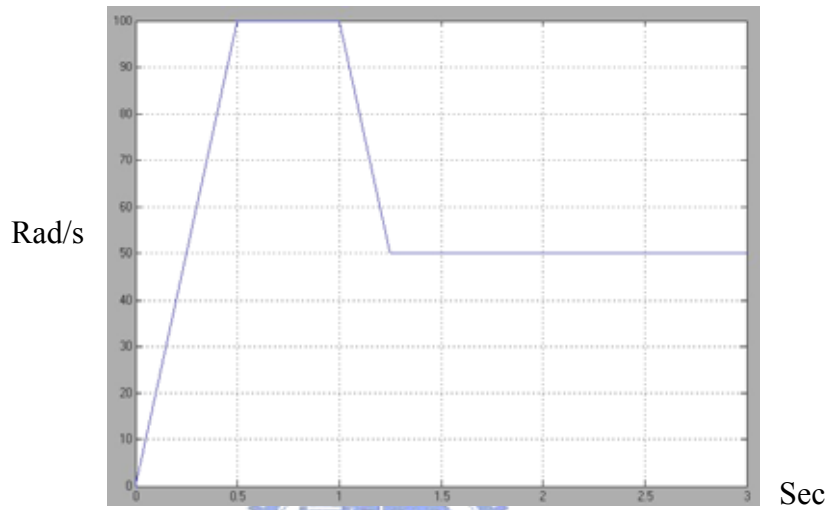


Fig. 4.11 Speed reference.

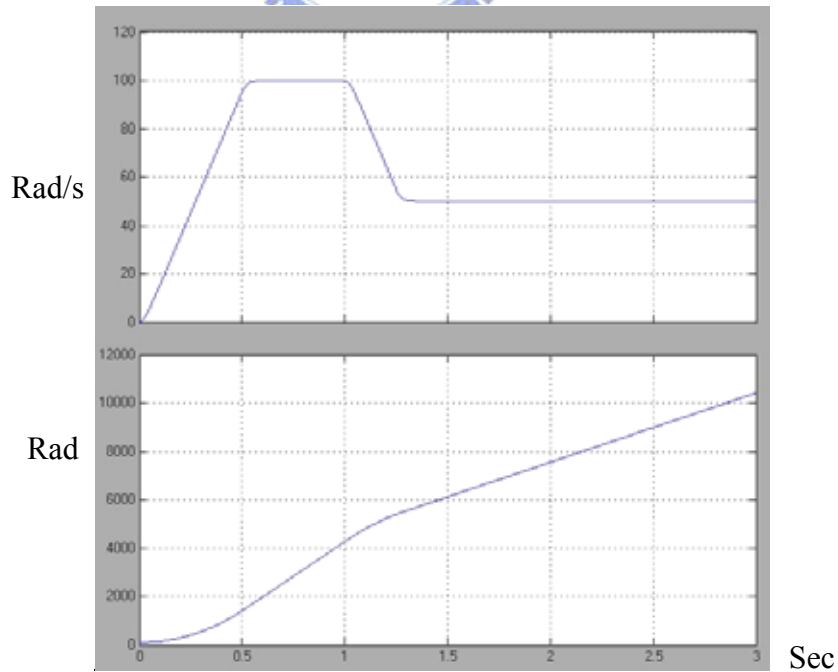


Fig. 4.12 Speed response and rotor position without load.

The speed reference ramps up at a slope of 200 rad/s, and it finally reaches to 50 rad/s. The motor behavior between field-oriented control (FOC) with and without sensor (EKF) are illustrated below. All the response comes out of the same PI controller which means the

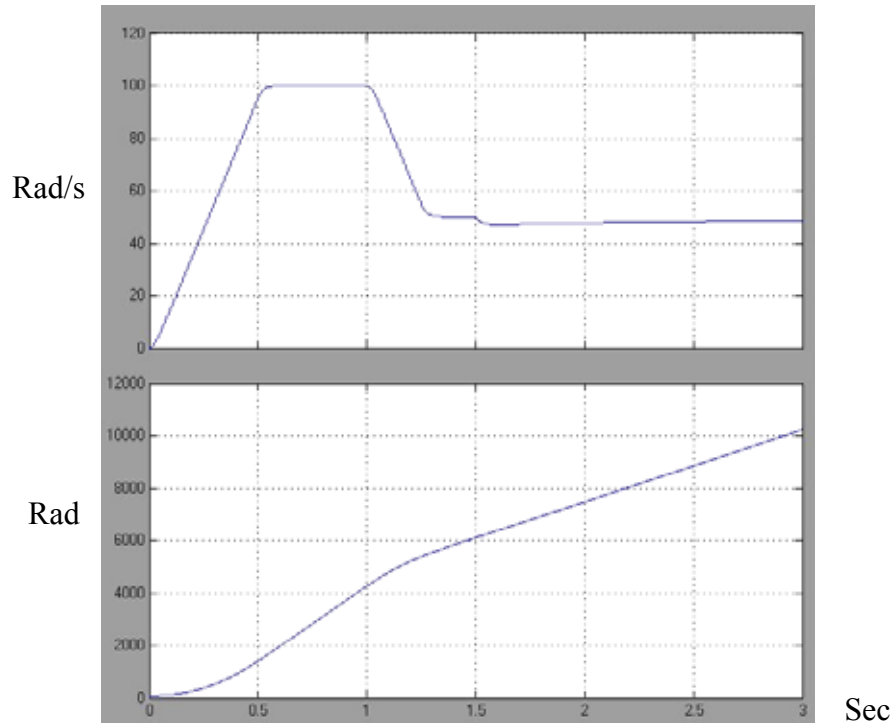


Fig. 4.13 Speed response and rotor position with load 0.1N-m at $t=1.5$ seconds.

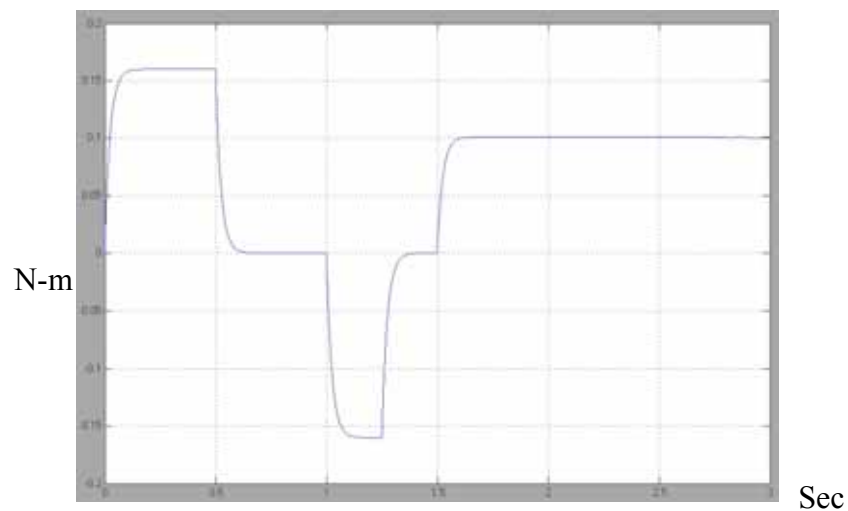


Fig. 4.14 Torque response with load 0.1Nm at $t=1.5$ seconds.

proportional gain and integral gain are the same in order to see the differences if using the EKF instead. Fig. 4.12 is the speed and rotor position responses of FOC without load using

position sensor. We can see the response is almost the same with the reference as shown in Fig 4.12. Next, a little bit loads is applied to the motor at 1.5 seconds which causes the speed response to change, as shown in Fig. 4.13. Actually the speed response can be better but the PI gain was adjusted only for the EKF instead of the FOC in the purpose of stability, so the PI gain is not suitable for the FOC here. As mentioned, the PI gains remain unchanged throughout the entire simulation. The torque response is shown in Fig. 4.14. We can see the applied torque is 0.1 Nm at 1.5 seconds. Fig. 4.15 is the current response.

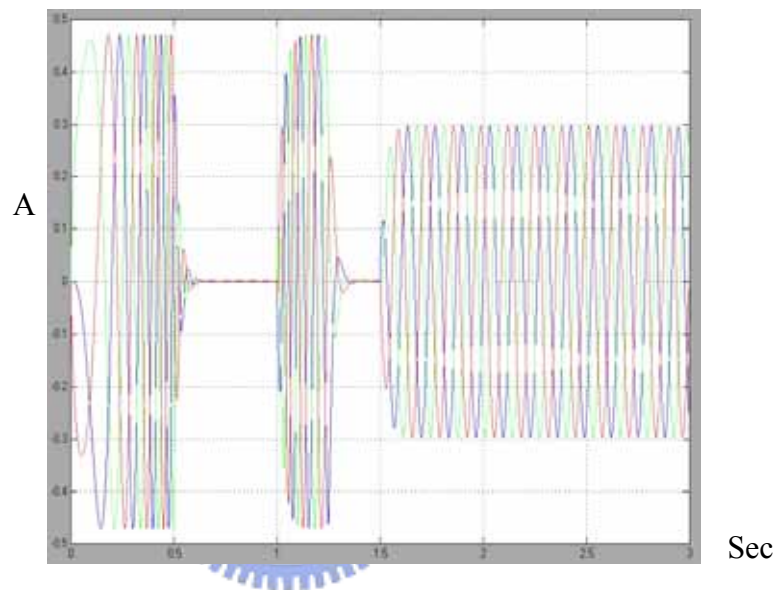


Fig. 4.15 Current response with load of 1.5 N-m at $t = 1.5$ seconds.

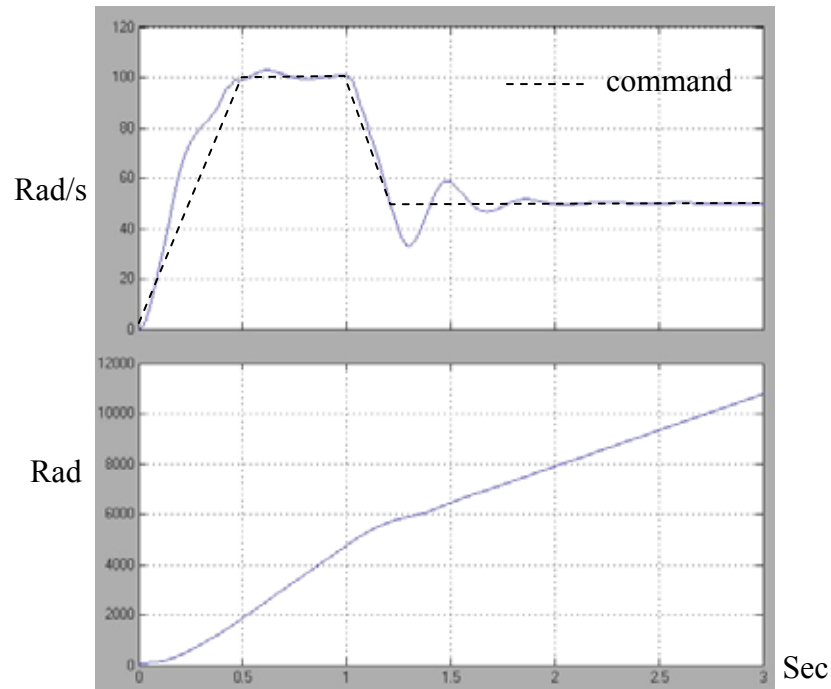


Fig. 4.16 Speed response and rotor position using EKF.

Next is the motor behavior without position feedback which uses the EKF to calculate the rotor position. Fig. 4.16 is the speed response without any load using EKF. The speed response looks not so good compared to Fig. 4.12. Of course the response is poor in EKF, because the actual position sensor was not applied. The EKF will execute and output rotor position and speed every 500 us (2 KHz), because the TMS320LF2407 fixed-point DSP is adopted for the experiment, and a lot of matrices manipulation will be involved for EKF theory in real time. 2 KHz execution rate is acceptable for the DSP. The Kalman gain K will be adapted constantly, see below:

$$t = 0.000000 \text{ second,}$$

$$K = \begin{bmatrix} 0.485420 & 0.000000 \\ 0.000000 & 0.487555 \\ 0.000000 & -0.828697 \\ 0.000000 & 0.000000 \end{bmatrix}.$$

t= 2.000000 seconds,

$$K = \begin{bmatrix} 0.485420 & 0.000000 \\ 0.000000 & 0.487555 \\ 0.631452 & 0.629459 \\ -0.139311 & 0.101825 \end{bmatrix}.$$

t= 3.000000 seconds,

$$K = \begin{bmatrix} 0.485420 & 0.000000 \\ 0.000000 & 0.487555 \\ 0.371937 & -0.765470 \\ 0.172920 & 0.061218 \end{bmatrix}.$$

The initial covariant matrices for the simulations are :

$$P_0 = \begin{bmatrix} 0.1 & 0 & 0 & 0 \\ 0 & 0.1 & 0 & 0 \\ 0 & 0 & 200 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad Q = \begin{bmatrix} 0.4 & 0 & 0 & 0 \\ 0 & 0.4 & 0 & 0 \\ 0 & 0 & 16 & 0 \\ 0 & 0 & 0 & 4 \end{bmatrix}, \quad R = \begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix}. \quad (4.22)$$

We can see Kalman gain changes all the time, especially the gain for rotor position and speed.

Fig. 4.17 and 4.18 are current and torque response of the EKF without loads.

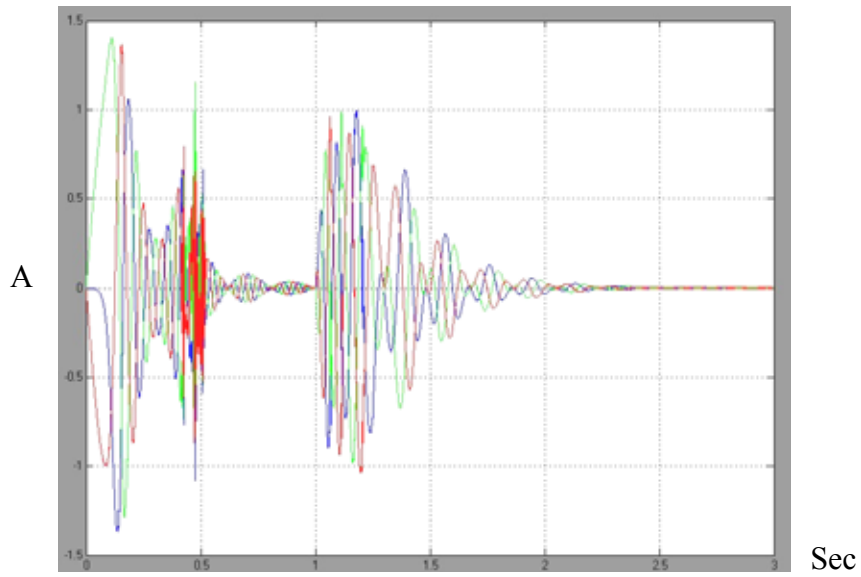


Fig. 4.17 Current response using EKF in no load condition.

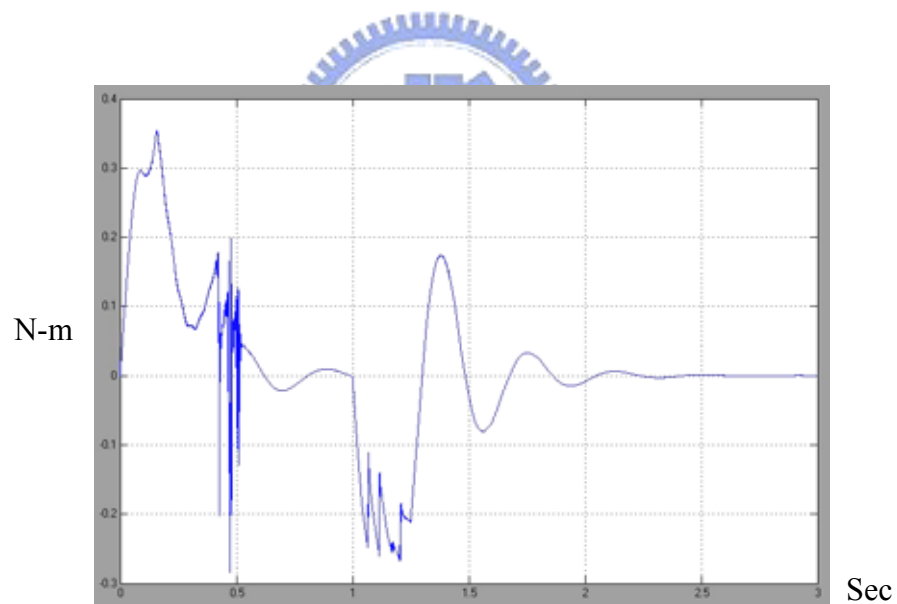


Fig. 4.18 Torque response without load using EKF.

Fig. 4.19 shows the speed and accumulated position errors compared with the actual speed and rotor position. The speed error will decay to zero and the rotor accumulated position error no longer increases after 1.5 seconds. The accumulated position error is about 2π or

6.24 radian. Fig. 4.20 shows that a light load at 1.5 seconds was applied which results the speed to oscillate a little bit compared with the Fig. 4.16. Fig. 4.21, 4.22 and 4.23 shows the same phenomenon due to a light load applied at 1.5 seconds.

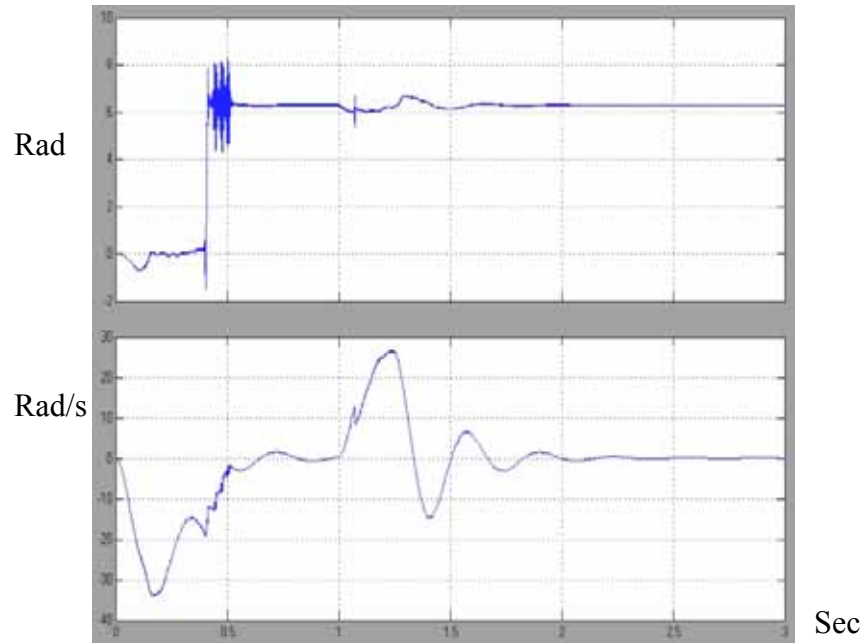


Fig. 4.19 Rotor position accumulated error and speed error without load.



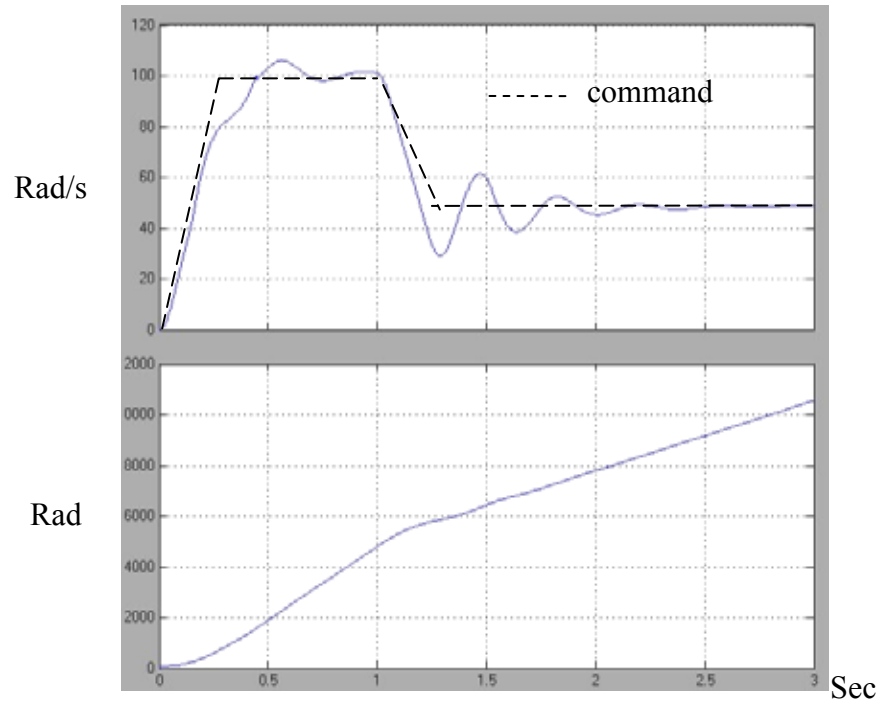


Fig. 4.20 Speed response and rotor position using EKF with load 0.1 N-m at $t=1.5$ seconds..

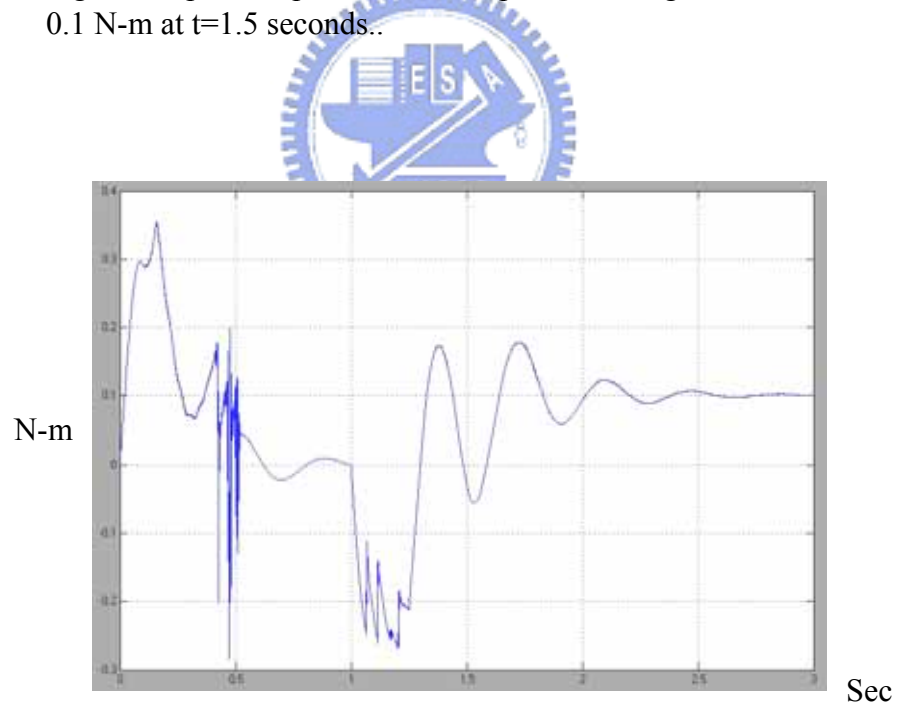


Fig. 4.21 Torque response using EKF with load 0.1 N-m at $t=1.5$ seconds.

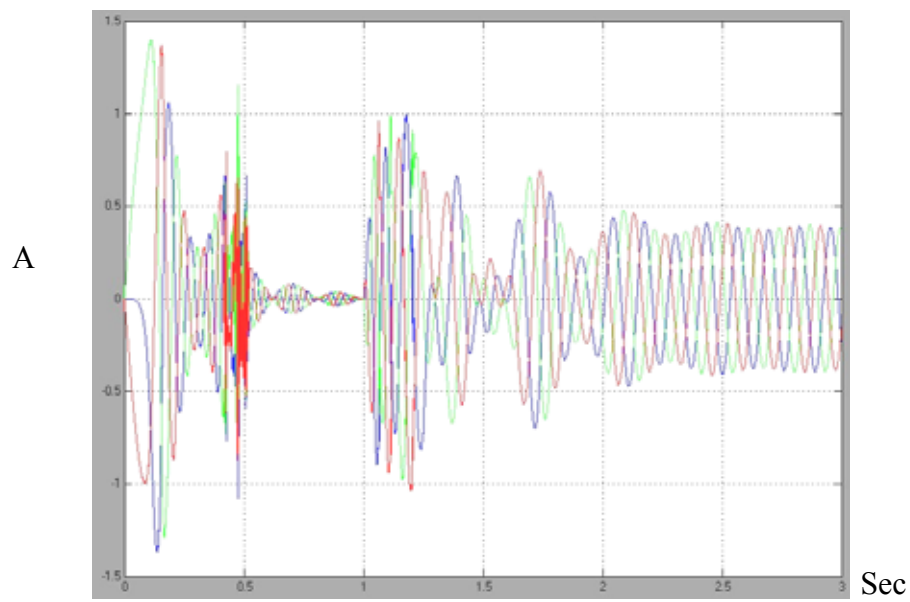


Fig. 4.22 Current response using EKF with load 0.1 N-m.

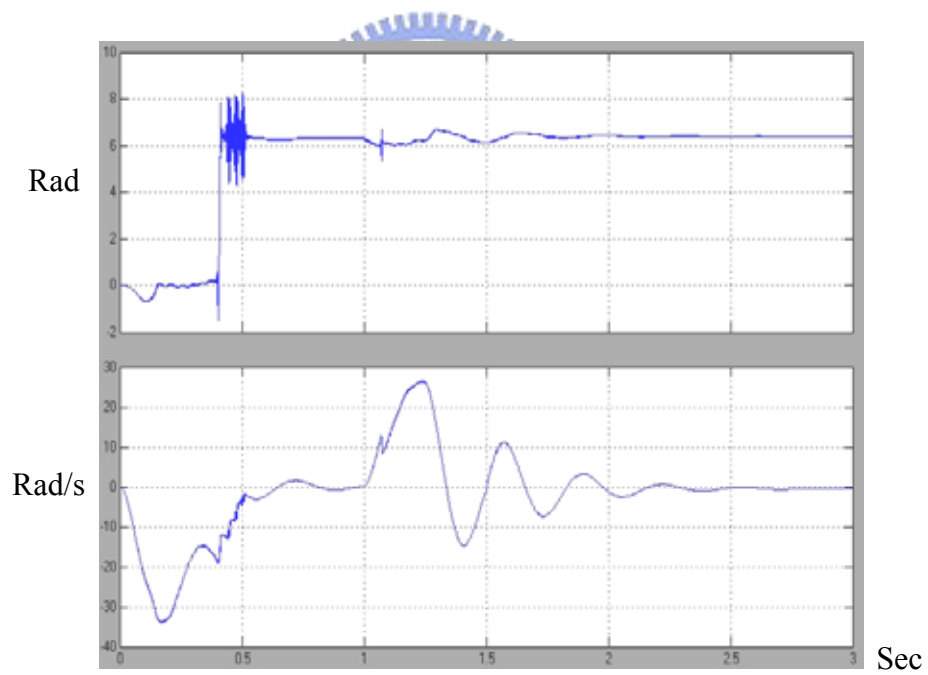


Fig. 4.23 Rotor position accumulated error and speed error using EKF with load 0.1 N-m.

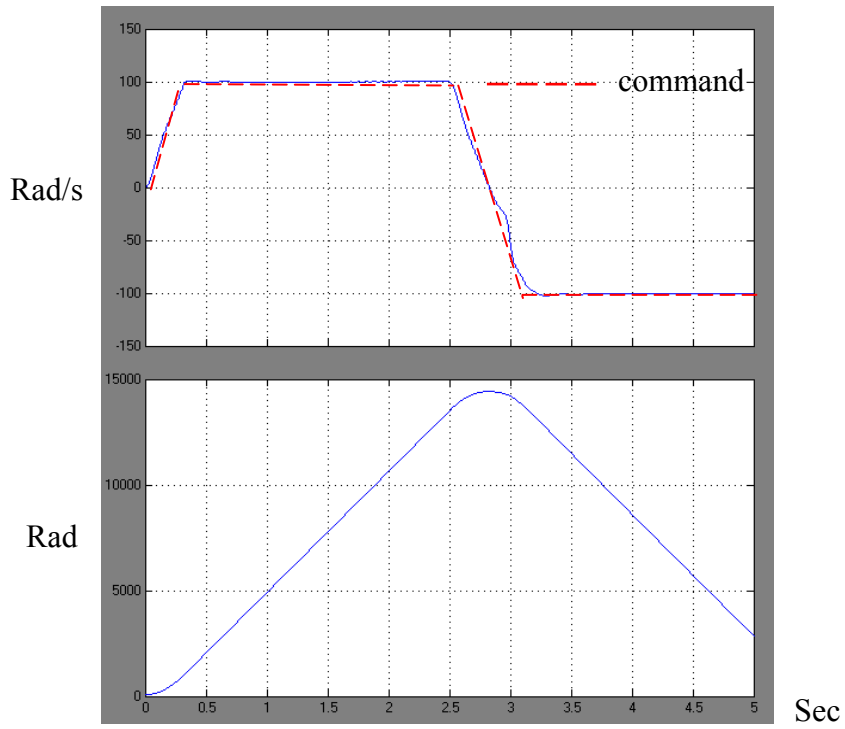


Fig. 4.24 Speed and rotor position using EKF.

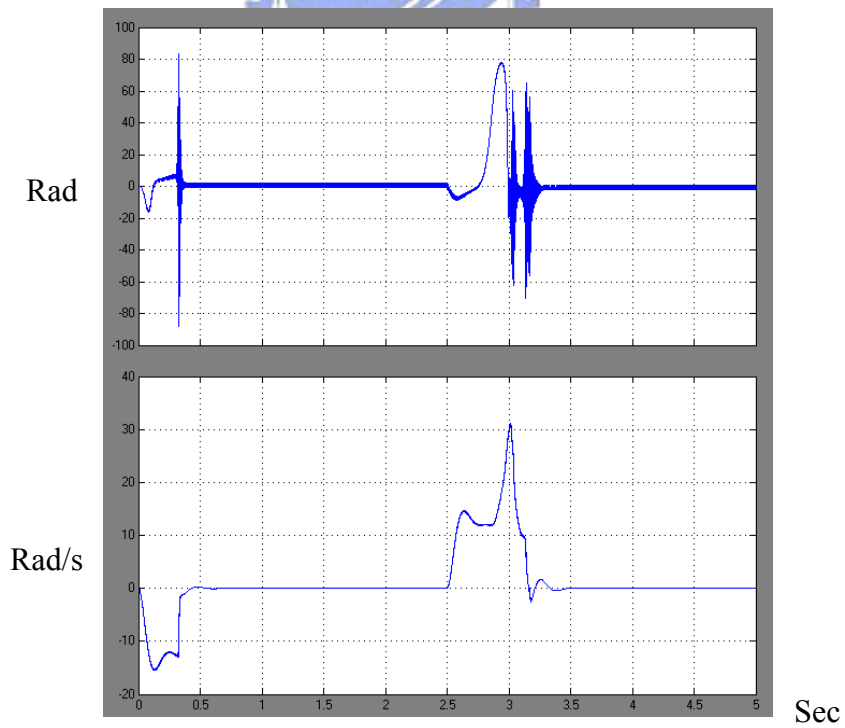


Fig. 4.25 Rotor position and speed error using EKF.

The speed response is not good as shown in Fig. 4.16 due to the values of the covariance matrices P, Q and R. Another set of matrices which was derived by try and error is selected as below,

$$P_0 = \begin{bmatrix} 0.1 & 0 & 0 & 0 \\ 0 & 0.1 & 0 & 0 \\ 0 & 0 & 250 & 0 \\ 0 & 0 & 0 & 2 \end{bmatrix}, \quad Q = \begin{bmatrix} 0.4 & 0 & 0 & 0 \\ 0 & 0.4 & 0 & 0 \\ 0 & 0 & 10 & 0 \\ 0 & 0 & 0 & 2 \end{bmatrix}, \quad R = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}. \quad (4.23)$$

The speed response is much better compared with Fig. 4.16 in no load condition as shown in Fig. 4.24.

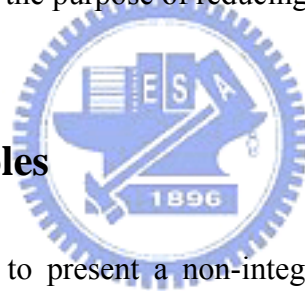


Chapter 5

Software Implementation of Kalman Filter Estimating

Speed and Position

To implement the EKF algorithm in real time estimation by using a fixed-point DSP is not an easy task because there are a lot of numerical calculations involved such as signal conditioning, scaling, quantization noise, precision control, and all these computations must be completed within a very short specified time interval. First the motor parameters like inductances, constant of back EMF and resistances should be known. Next the numerical range of other parameter vectors that are phase currents and voltages, shaft speed and positions should be determined as well to decide the Q format. Then assembly language is adopted for this experiment for the purpose of reducing the calculation time.



5.1. Scaling for variables

Q format must be used to present a non-integer value for a fixed-point DSP using assembly language. For a 1024-pulse encoder with 1 KHz sampling frequency, there are 204 counts every millisecond at 3000 RPM. The minimum Q format that can represent the sampled data is Q8 and its resolution is $1/256$ which is better than $1/204$. The recommended resolution of the Q format for calculation should better be at least four times of the sampled data, that is, Q10. But ten times of the resolution of the sampled data is preferred for a better precision which is Q11.

A 10-bit A/D conversion is common in a 16-bit DSP or MCU which means the resolution of the A/D conversion is $1/512$ for a signed value. From above, the minimum Q format would be Q11 (four times of the resolution).

Another consideration for choosing the Q format is the regulator performances which are the transient response and steady state error. To compromise all those factors discussed above, Q12 is mainly used in the assembly language.

5.1.1. Scaling for current, speed and position

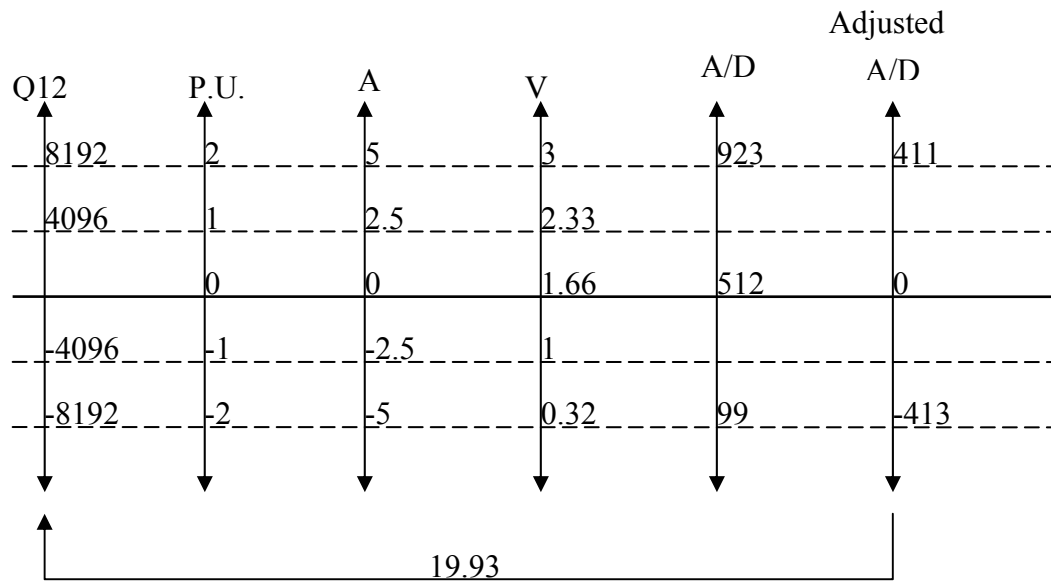


Fig. 5.1 Current scaling.

For the PMSM used in the experiment, the phase current is in a range of 5A which is needed to choose the proper Q format for a fixed-point DSP, as shown in Fig. 5.1. The DSP is supplied by 3.3V that means the A/D input voltage range is bounded form 0 to 3.3V and the current transducer has an output of 4V at 5A, so the output voltage from the transducer has to be scaled and shifted to fit the range. After the A/D is done, it has to be converted to Q format and a Q12 is chosen for the experiment. See below for the calculation and its error.

$$19.93_{\text{dec}} = 5102.08_{\text{Q12}}$$

$$\text{Scaled current at 5 A} = 411_{\text{adjusted A/D}} = 8191_{\text{Q12}} = 4.999 \text{ A.}$$

$$\text{Error} = 5 \text{ A} - 4.999 \text{ A} = 0.0001 \text{ A.}$$

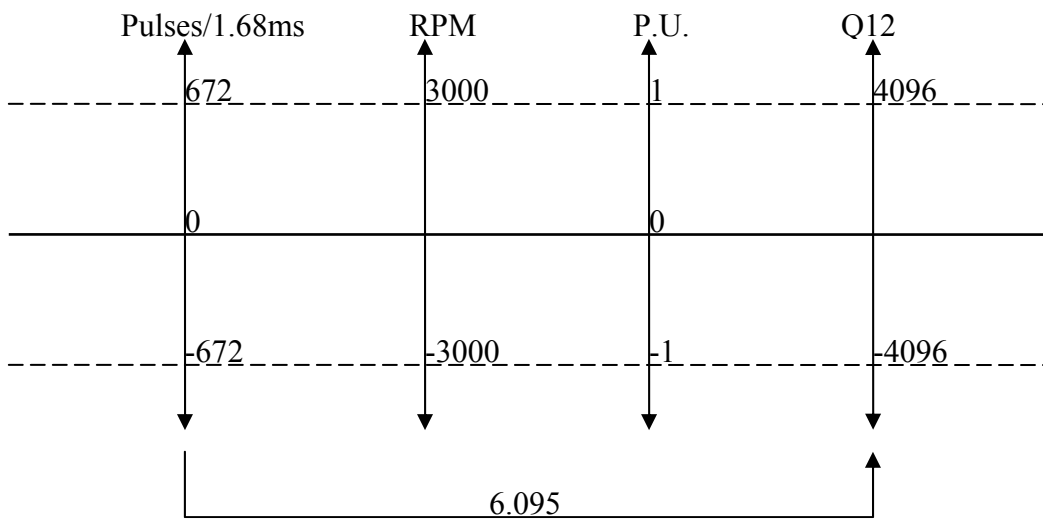


Fig. 5.3 Speed scaling for a 2000 PPR encoder.

$$6.095_{\text{dec}} = 1561_{\text{Q8}}$$

$$\text{Calculated speed at 3000 rpm: } 672 \times 1561 / 256 = 4097_{\text{Q12}} = 3000.7 \text{ rpm.}$$

$$\text{Error} = \text{Real speed} - \text{Calculated speed} = 3000 - 3000.7 = -0.7 \text{ rpm.}$$



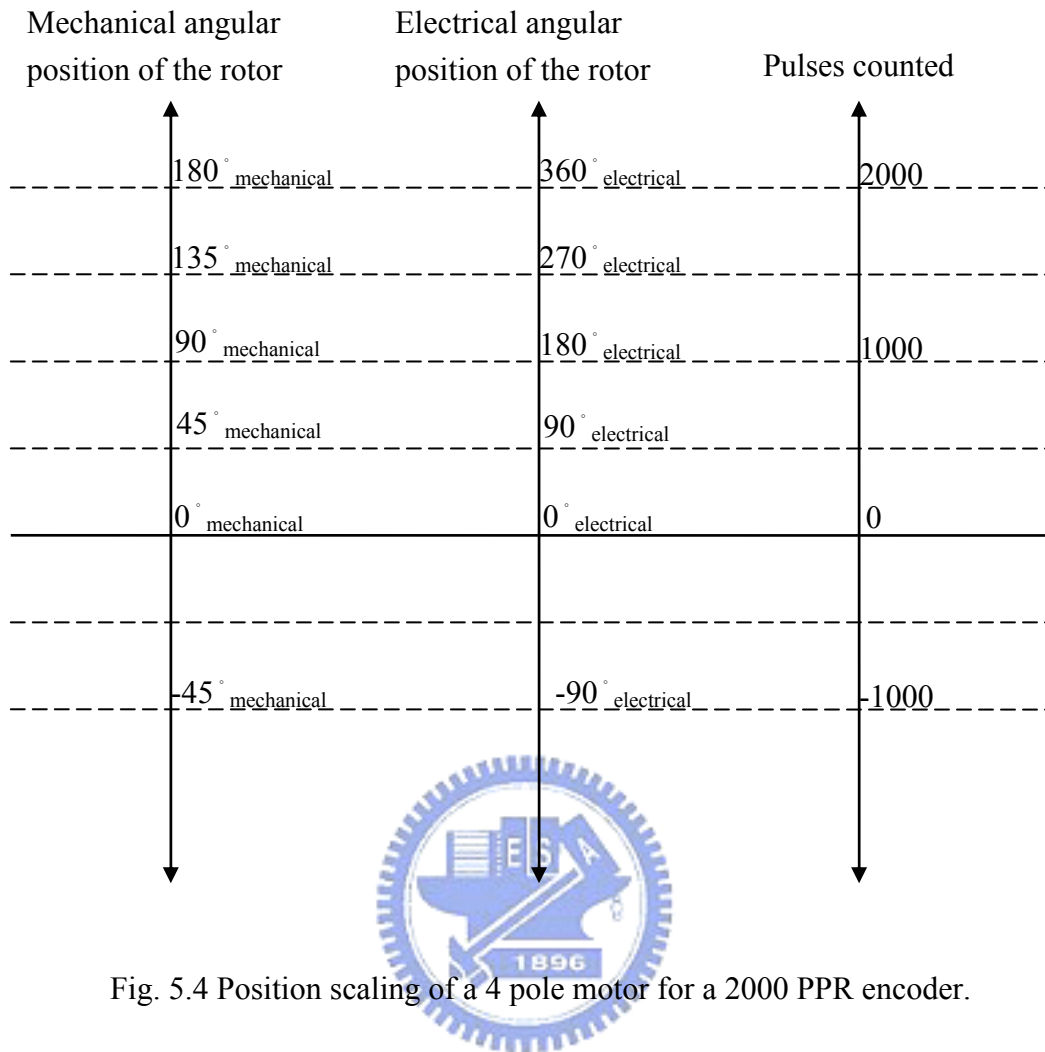


Fig. 5.4 Position scaling of a 4 pole motor for a 2000 PPR encoder.

From above, Q12 format is capable for the fixed-point calculation and most of the other variables use this format as well to simply the software especially in the EKF macro.

5.1.2. Scaling for EKF models

The chosen model is

$$f(x) = \begin{bmatrix} -\frac{r_s}{L_d} i_s^\alpha + \frac{\lambda}{L_d} \omega_r \sin \theta_r \\ -\frac{r_s}{L_q} i_s^\beta - \frac{\lambda}{L_q} \omega_r \cos \theta_r \\ 0 \\ \omega_r \end{bmatrix}. \quad (5.1)$$

Replacing the parameters with a real numerical value, $L_d= 12$ mH, $L_q= 14$ mH, $r_s= 3.4$ ohm and $\lambda = 0.11327$ wb-T into (5.1) yields

$$f(x) = \begin{bmatrix} -283i_s^\alpha + 9.44\omega_r \sin \theta_r \\ -229i_s^\beta - 7.65\omega_r \cos \theta_r \\ 0 \\ \omega_r \end{bmatrix}. \quad (5.2)$$

For the EKF, the integration period T is 0.5 ms, then

$$T \cdot f(x) = 0.0005 f(x) = \begin{bmatrix} -0.1415i_s^\alpha + 0.00472\omega_r \sin \theta_r \\ -0.1145i_s^\beta - 0.003825\omega_r \cos \theta_r \\ 0 \\ 0.0005\omega_r \end{bmatrix} \quad (5.3)$$

and from (4.13) and (4.14),

$$F(K) = \begin{bmatrix} -0.1415 & 0 & 0.00472 \sin \theta_r & 0.00472 \cos \theta_r \\ 0 & -0.1145 & -0.003825 \cos \theta_r & 0.003825 \sin \theta_r \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0.0005 & 0 \end{bmatrix}. \quad (5.4)$$

Converting the constant values in (5.4) into Q12 format yields

$$F(K) = \begin{bmatrix} -580 & 0 & 19.3 \sin \theta_r & 19.3 \cos \theta_r \\ 0 & -469 & -15.6 \cos \theta_r & 15.6 \sin \theta_r \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 \end{bmatrix}_{Q12}. \quad (5.5)$$

In similar way,

$$B = \begin{bmatrix} \frac{1}{L_d} & 0 \\ 0 & \frac{1}{L_q} \\ 0 & 0 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 0.04167 & 0 \\ 0 & 0.03379 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 171 & 0 \\ 0 & 138 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}_{Q12} \quad (5.6)$$

and

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 4096 & 0 & 0 & 0 \\ 0 & 4096 & 0 & 0 \end{bmatrix}_{Q12}. \quad (5.7)$$

5.1.3. Scaling for covariance matrices and Kalman gain K

The covariance matrices and Kalman gain K are not well known as phase currents, shaft speed and rotor position which have a specific numerical range at beginning. To identify the ranges of those matrices and gain K , the proposed method is to check the simulation results. According to the simulations results by the simulink, the matrices P , Q , R and gain K are almost less than one except the elements (3, 3) and (4, 4) in P and Q , respectively, which means a Q12 format could cover most of the variable ranges. Below is the matrices $P(k)$ coming from the simulation of (4.12),

$$P(k) = \begin{bmatrix} 0.4716 & 0 & -0.4915 & -0.1613 \\ 0 & 0.4757 & 0.6901 & -0.0844 \\ -0.4915 & 0.6901 & 216 & 0.1 \\ -0.1613 & -0.0844 & 0.1 & 4.8 \end{bmatrix}. \quad (5.8)$$

Obviously element $P(3, 3)$ is out of the range for Q12. Another covariance matrices Q and R is presented, respectively, as follows:

$$Q = \begin{bmatrix} 0.4 & 0 & 0 & 0 \\ 0 & 0.4 & 0 & 0 \\ 0 & 0 & 16 & 0 \\ 0 & 0 & 0 & 4 \end{bmatrix}, \quad R = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}. \quad (5.9)$$

Unlike matrix $P(k)$ which changes with time, the elements of Q and R are constant and thus Q12 format can be applied except for the elements (3, 3) and (4, 4). In the EKF simulation, elements $P(3, 3)$ could be chosen form 50 to 500 or larger and it comprises the addition of Q , as shown in (4.12), so the $Q(3, 3)$ should be in the same format, that is Q5, as $P(3, 3)$. $Q(4, 4)$ uses Q8 format that has a positive range of 127 which can handle the possible values to choose according to the simulation. The Kalman gain K also changes with time, here below shows one of the matrices K which comes from the result of simulink simulation,

$$K(k) = \begin{bmatrix} 0.4854 & 0 \\ 0 & 0.4875 \\ 0.3719 & -0.7654 \\ 0.1729 & 0.0612 \end{bmatrix}. \quad (5.10)$$

According to the simulations, the elements of $K(k)$ did not change substantially and Q12 can cover all of them.

5.2. Software flowchart of the control algorithm

In this experiment, the EKF subroutine is executed every 200 us due to the precision of Q12 format. It's better to raise the execution frequency above 10 KHz which is synchronized with the current loop.



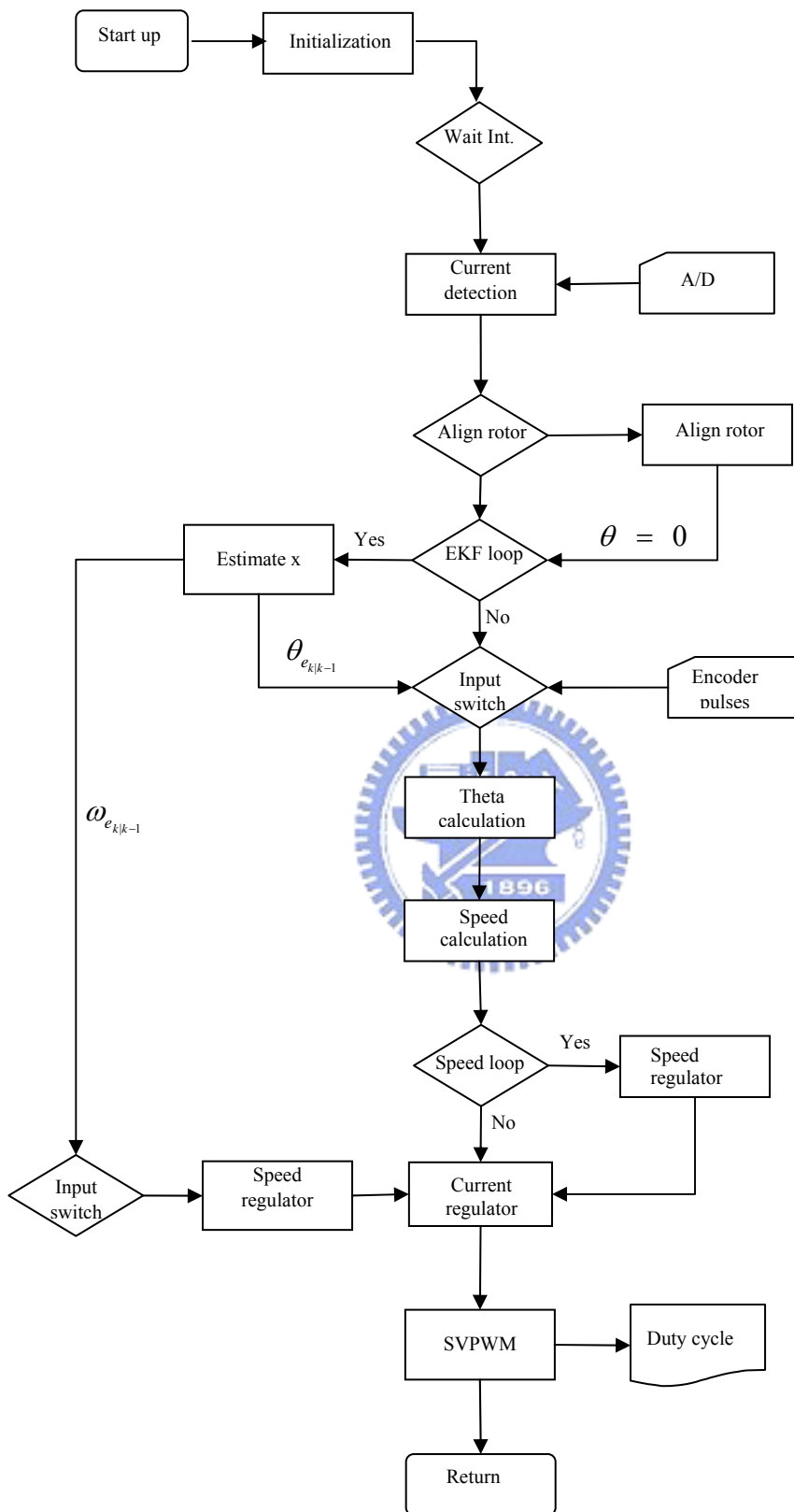


Fig. 5.5 Software flowchart of the control algorithm.

5.3. Software flow chart of EKF

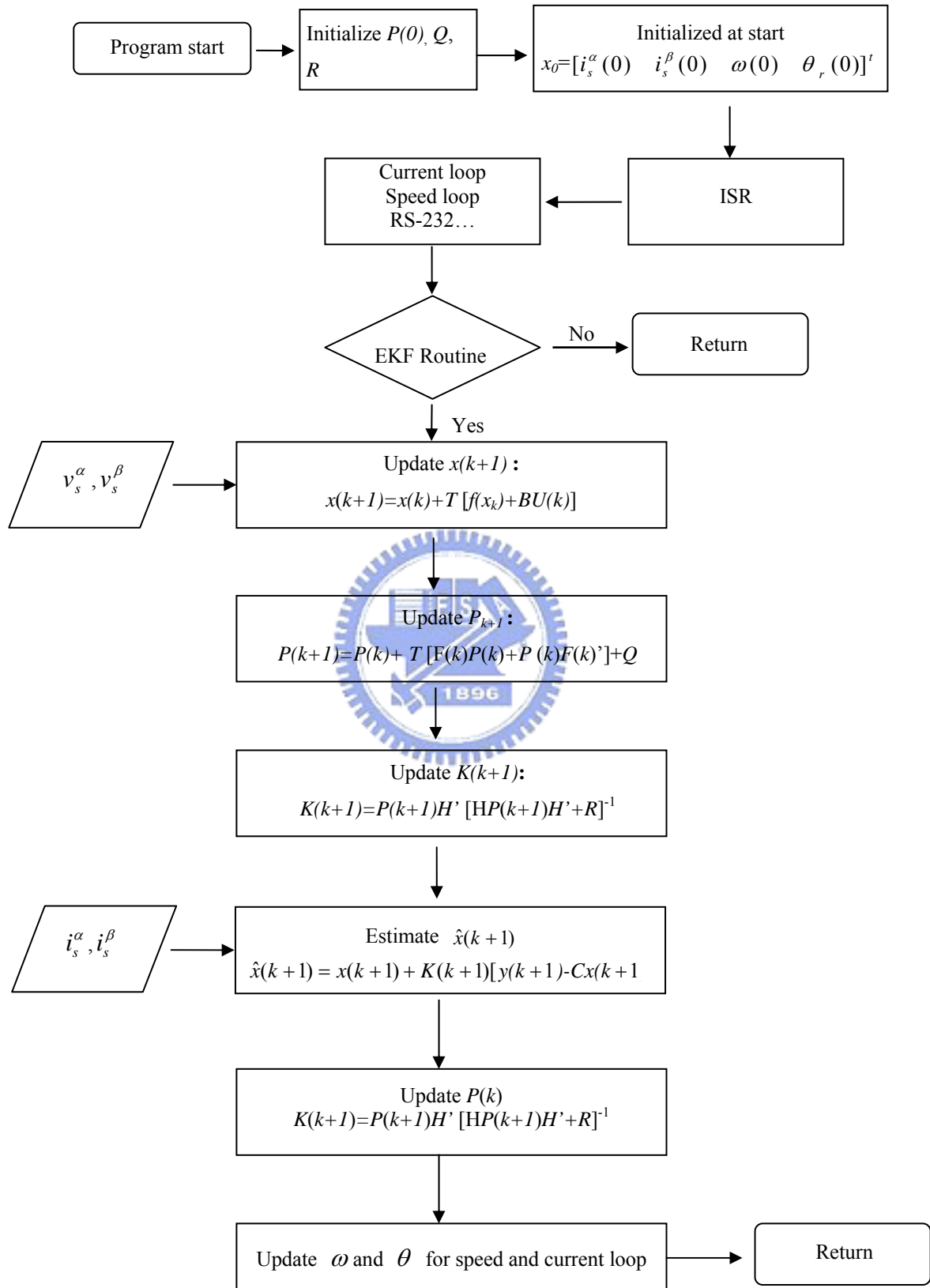


Fig. 5.6 Software flowchart of EKF.

5.4. Assembly language implementation

To reduce the software execution time of a fixed-point DSP in real time, the assembly language is chosen for the implementation. Most of the EKF codes are composed by macro instead of subroutines which would take more computation time. Because calling a subroutine would cause pipeline operation of a DSP low efficient. See below for macro examples. Those macros are “mmf”, “mmtf”, “maddG”, “madi”, “mmtfrag”, “mmfrag”, “madjG”, “detG”, “mdscIG”, and “msub”.

```

;-Step-3-----P1=P+T(F*P+P*F')+Q-----
mmf      F_11,P_11,RLT1_11,4,4,4,EKFtmp1,EKFtmp2,EKFtmp3 ;T(F*P)      -->RLT1
mmtf     P_11,F_11,P_11,4,4,4,EKFtmp1,EKFtmp2,EKFtmp3 ;T(P*F')      -->P1
maddG    RLT1_11,P_11,P_11,4,4 ;T(F*P+P*F') -->P1
madi     P_11,Q_11,4 ;T(F*P+P*F')+Q-->P1
maddG    P_11,P_11,P_11,4,4 ;update P1

;-Step 4-----K =P1*H'*inv(H*P1*H'+R)-----
mmtfraG  P_11,H_11,RLT1_11,4,4,2 ;P1*H' -->RLT1      (4*4)(4*2)=(4*2)
mmfraG   H_11,RLT1_11,K_11,2,4,2 ;H*RLT1 -->K = H*P1*H' (2*4)(4*2)=(2*2)
madi     K_11,R_11,2 ;K+R -->K = H*P1*H'+R result=2*2
madjG    K_11,P_11 ;adj(K) -->P = adj (H*P1*H'+R)
mmtfraG  RLT1_11,P_11,K_11,4,2,2 ;K_11 = P1*H'*adj (H*P1*H'+R) (4*2)(2*2)=(4*2)
detG     P_11,RLT1_11 ;det (adj (H*P1*H'+R)=det (H*P1*H'))
mdscIG   K_11,RLT1_11,K_11,4,2,EKFtmp1,EKFtmp2,EKFtmp3,EKFtmp4

;-Step 5-----X(k+1)=X(k)+K (y-X)-----
msubG    iSalfa,Ialpha,P_11,2,1
mmfraG   K_11,P_11,RLT1_11,4,2,1
maddG    Ialpha,RLT1_11,Ialpha,4,1

;-Step 6-----P=P1-K1*H*P1-----
mmfraG   K_11,H_11,RLT1_11,4,2,4
mmfraG   RLT1_11,P_11,P_11,4,4,4
msub     P_11,P_11,P_11,4,4

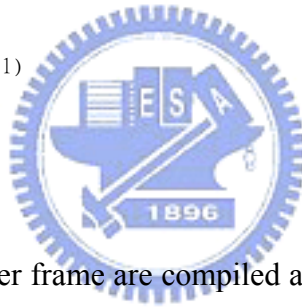
```

To make the source codes reusable, auxiliary register addressing mode should be adopted for the DSP TMS320LF2407A. For the register addressing, it can access the full range of the DSP memory space from 0000H to FFFFH regardless of the data page of those variables used for EKF codes. The macro example “mmfrag” uses AR2 and AR3 for addressing “mat1”, “mat2” and “reslt” rather than direct addressing. Utilizing the macro language like “.loop”, “.asg”, “.eval” and “.break” has significant advantages for the matrices manipulations that can simplify the codes and reduce the time needed for loop operations without interrupting pipeline sequences.

```

mmfraG      .macro  mat1,mat2,result,s11,s21,s22
            setc   ovm                               ; saturation mode on.
            mar    *,AR2
            .asg   0,i
            .loop
            .asg   0,j
            .loop
            zac
            mpy    #0                                ;clear PREG
            lar    AR2, #mat1+i*s21+0
            lar    AR3, #mat2+0*s22+j
            .asg   0,k
            .loop
            lta    *+,AR3
            mpy    *0+,AR2
            .eval  k+1,k
            .break (k = s21)
            .endloop
            apac
            lar    AR2,#result+s22*i+j
            sach   *,4
            .eval  j+1,j
            .break (j = s22)
            .endloop
            .eval  i+1,i
            .break (i = s11)
            .endloop
            clrc  ovm                               ; saturation mode off
            spm   0                                ;shifting off
            .endm

```



The codes inside the upper frame are compiled as below. The macro instructions do not occupy any computation time since no machine codes are built. The “.loop” instruction does not really make the codes running like a circle with “return” and “call” but straight forward instead until “k” equals 4. That is why the macros can reduce the computation time needed.

		.asg	0,k
0812	70ab	lta	*+,AR3
0813	54ea	mpy	*0+,AR2
		.eval	0+1,k
		.break	(1 = 4)
0814	70ab	lta	*+,AR3
0815	54ea	mpy	*0+,AR2
		.eval	1+1,k
		.break	(2 = 4)
0816	70ab	lta	*+,AR3
0817	54ea	mpy	*0+,AR2
		.eval	2+1,k
		.break	(3 = 4)
0818	70ab	lta	*+,AR3
0819	54ea	mpy	*0+,AR2
		.eval	3+1,k
		.break	(4 = 4)
081a	be04	apac	
081b	bf0a	lar	AR2,#K_11+2*0+0
081c	0065-		

5.5. Quasi floating point method

In the real-time implement, a “quasi floating point” method is introduced for the matrix inverse of the EKF. The recursive step, (3.15), is present below again:

$$K(k+1) = P(k+1|k)C^T [CP(k+1|k)C^T + R]^{-1} \quad (5.1)$$

$P(k+1|k)C^T$ is a 4 by 2 matrix, there are 16 elements in it. It will be multiplied by a 2 by 2 matrix $[CP(k+1|k)C^T + R]^{-1}$. The result could be significantly different from the real one because of the calculation precision.

$$\frac{a_{i,j}}{scl} = \frac{0.5}{2^k scl} a_{i,j} \times 2^{k+1} \quad (5.2)$$

In (4.2), $a_{i,j}$ are the elements of the matrix and scl is the divisor. The absolute value of the divisor scl is scaled by 2^k and the range of $2^k scl$ is between (0.5, 1). So the result $\frac{0.5}{2^k scl}$ is in the range of (0.5, 1) as well. All this procedures are to get a better precision for the inverse of the matrix. Q15 format is adopted in the software implement. A care must be taken that the absolute value of scl should be less than one.

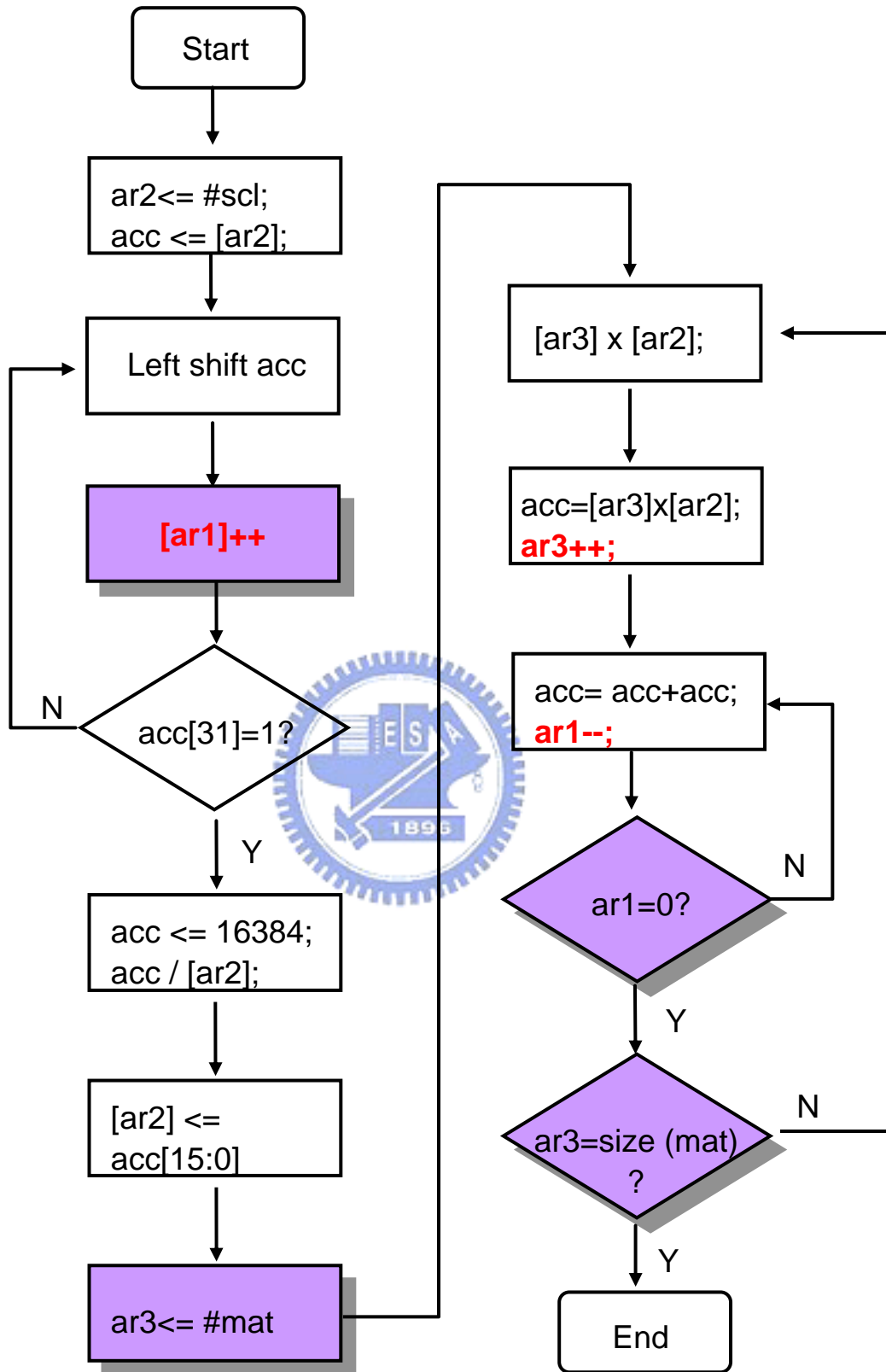


Fig. 5.7 Software flowchart of EKF.

5.6. Timing analysis of the EKF

Table 5.1 Software execution time.

Routine name		Execution time	
1.	Extended Kalman filter subroutine	64.65 us	83.26%
2.	Current sensing	2.55 us	3.28%
3.	Current scaling	0.8 us	1.03%
4.	Park transform	0.22 us	0.28%
5.	Clarke transform	0.35 us	0.45%
6.	Inverse Clark transform	0.35 us	0.45%
7.	Motion sensing	0.5 us	0.64%
8.	Sine, Cosine calculation	0.8 us	1.03%
9.	Speed regulator	2.6 us	3.34%
10.	Space Vector PWM	4.2 us	5.41%
11.	RS232 sampling	0.6 us	0.77%
Total		77.65 us	100%

For an EKF calculation of TMS320LF2407A, there are 8 matrix multiplications, 8 matrix additions and 1 matrix inversion. As shown in table 4.1, most calculation power is consumed by EKF subroutine; it is 83% of the total execution time. The DSP still have enough time capable of executing command inputs and outputs when running at 40 MHz.

Table 5.2 EKF execution time.

EKF Phase	Macro Name	Description	Execution Time (us)	
$x(k+1)=x(k)+T [f(x_k)+BU(k)]$	N.A.	Macro not available	1.875	2.90 %
$P(k+1)=P(k)+ T [F(k)P(k)+P (k)F(k)'] +Q$	mmf	[4 by 4]×[4 by 4]	9.85	15.24 %
	mmtf	[4 by 4]×[4 by 4] with transposition	9.4	15.54 %
	maddG	[4 by 4]+[4 by 4]	1.525	2.36 %
	madi	[4 by 4]+[4 by 4] with only diagonal elements	0.625	0.97 %
	maddG	[4 by 4]+[4 by 4]	1.525	2.36 %
$K(k+1)=P(k+1)H' [HP(k+1)H'+R]^{-1}$	mmtfraG	[4 by 4]×[4 by 2]	3.8	5.88 %
	mmfraG	[2 by 4]×[4 by 2]	2.05	3.17 %
	madi	[2 by 2]+[2 by] with only diagonal elements	0.475	0.73 %
	madjG	Adjunct of [2 by 2]	0.6	0.93 %
	mmfraG	[4 by 2]×[2 by 2]	3.05	4.72 %
	detG	Det of [2 by 2]	0.5	0.77 %
	mdsclG	[4 by 2]/scalar	8.525	13.19 %
$\hat{x}(k+1)=x(k+1)+K(k+1)[y(k+1)-Cx(k+1)]$	msubG	[2 by 1]-[2 by 1]	0.475	0.73 %
	mmfraG	[4 by 2]×[2 by 1]	1.65	2.55 %
	maddG	[4 by 1]+[4 by 1]	0.625	0.97 %
$P(k) = P(k + 1)-K(k + 1)HP(k + 1)$	mmfraG	[4 by 2]×[2 by 4]	5.85	9.05 %
	mmfraG	[4 by 4]×[4 by 4]	7.45	11.52 %
	msub	[4 by 4]-[4 by 4]	1.525	2.36 %
Other			3.275	5.07 %
		Total execution time	64.65	100%

5.7. Memory requirements of the codes

Table 5.3 Memory requirements of the codes.

Program part (TMSLF2407A)	Program size (words)	Program space usage	Data size (words)	DRAM usage
FOC	1612	9.84 %	97	17.83 %
RS232	1507	9.20 %	8	1.47 %
EKF	2317	14.15 %	125	22.98 %
Total	5463	33.34 %	230	42.28 %



Chapter 6

Experimental Results

A major challenge in developing sensorless PMSM control scheme is its dynamic performance when operating at low-speed or even zero-speed under torque disturbances. The EKF sensorless control scheme is limited when operating in low-speed region, this is possibly due to very poor signal-to-noise ratio when operating in low-speed region. Poor controllability at low speed means it would be a problem when starting the motor. In fact, EKF may not converge if a wrong initial position is given. In the experiments, the research shows two methods for detecting the rotor position of a PMSM.

6.1. Method 1: Flux saturation

A 3-phase brushless DC motor (BLDCM) is also belonged to PMSM category. There are only six different rotor positions needed when driving the motor. For a 2-pole BLDC, as shown in Fig. 6.12, the rotor is in one of the six zones. Without Hall effect sensors, the magnetic flux saturation and unsaturation can be utilized due to the different stator currents

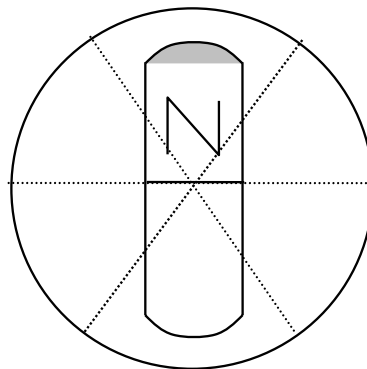


Fig. 6.1 2 poles PMSM rotor position.

which are used to distinguish which zone the rotor stays at. If a sufficient stator current is applied which produces a magnetic flux in the same direction with the N pole of the rotor, then the stator would start to saturate due to the enormous magnetic flux, and thus causes a

higher stator current.

Fig. 6.2 shows the inductance varies due to stator current. The sensed stator current will be higher at the same stator voltage applied with the same duration for a smaller inductance.

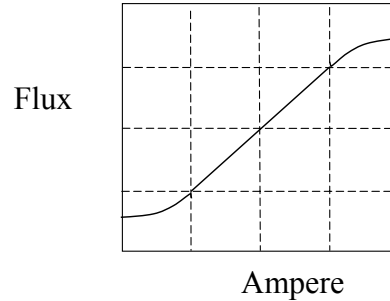


Fig. 6.2 Inductance curve.

Step 1:

At first step, we can narrow down the rotor position range within 180 degrees, as shown in

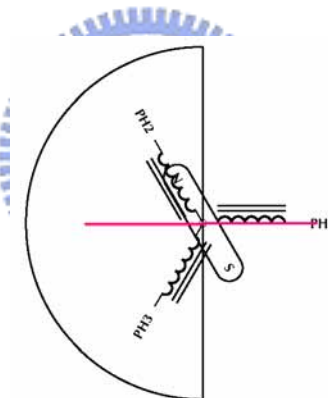


Fig. 6.3 Detecting rotor step 1.

Fig. 4.3. The red line shows the flux direction is aligned with PH1, the stator voltages are applied as shown in Fig. 4.4. We define phase U, V, W as PH1, PH2 and PH3. In Fig. 6.5 the power switches S1, S4 and S6 are switched on and the others off for a certain time (noted

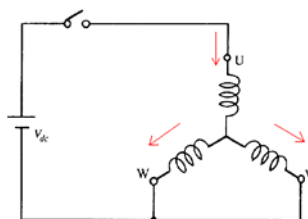


Fig. 6.4 Detecting currents.

as “shoot 1”), then S2, S3, S5 on and the others off (noted as “shoot 2”), which can produce positive and negative flux on the red line in Fig. 6.3. Next compare the stator current amplitude to determine the rotor position. In Fig. 4.3, the rotor N pole is located in the left side of the red line because the stator current “shoot 1” is less than the current “shoot 2” .

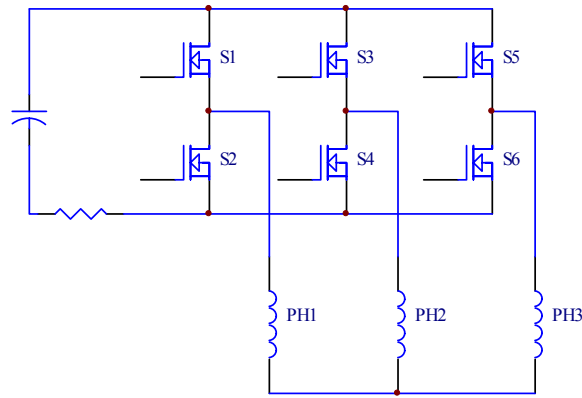


Fig. 6.5 Six power switches.

STEP 2:

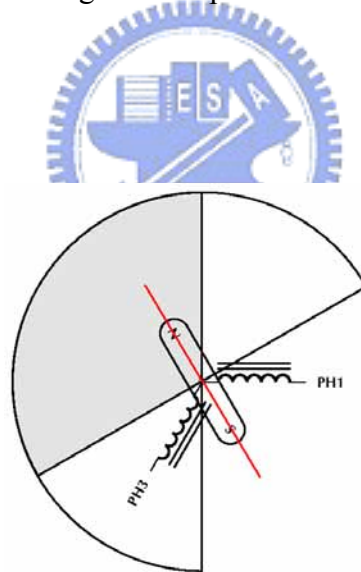


Fig. 6.6 Detecting rotor step 2.

Use the same procedures to turn on S2 S3 S6 and turn off the others (shoot 3), then turn on S1 S4 S5 and turn off the others (shoot 4). The rotor N pole should be in the left side of the red line because of the current of “shoot 3” is higher than “shoot 4”. The gray zone is the overlap of step 1 and step 2. After these two steps, the rotor position is narrowed down to 120 degrees as shown in the gray zone.

Step 3:

Again, turn on S2 S4 S5 and turn off the others (shoot 5), then turn on S1 S3 S6 and turn off the others (shoot 6). The current of "shoot 5" should be less than "shoot 6". Finally we can locate the rotor position within 60 degrees in the black zone.

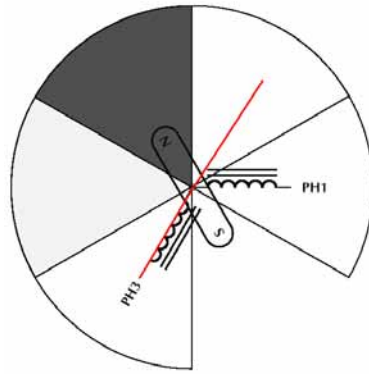


Fig. 6.7 Detecting rotor step 3.

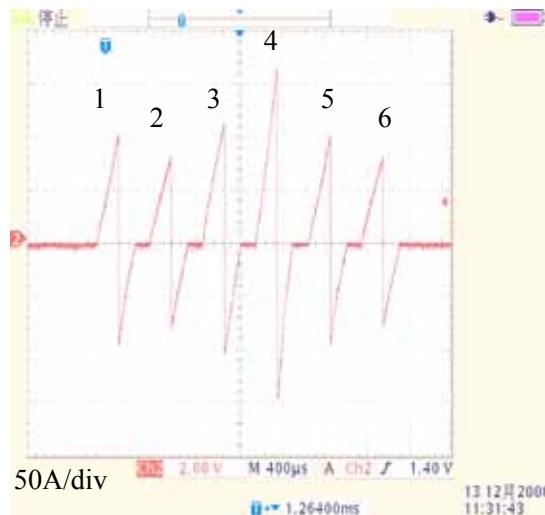


Fig. 6.8 Detecting current wave forms.

Fig. 6.8 shows the typical current wave forms on DC bus of a 48V BLDC motor for scooters. There are six shoots and each shoot has different amplitude. As described in step 1 to step 3, comparing each amplitude results six different states of the rotor position. Fig. 6.8 to Fig. 6.12 are the detected current wave forms on DC bus. The six positions are determined as below:

1. Assign a 3-bit variable which can present 6 rotor positions like 1 to 6.
2. If the amplitude shoot "1" is higher than shoot "2", then set the bit 2 of the variable

as “1”, if not set to “0”.

3. Repeat comparing the amplitudes of shoot “3” to “6” as described in procedure 2, you can get the values of bit 1 and bit 0.

Fig. 6.9 to Fig. 6.14 show the six different rotor positions and the detected currents.

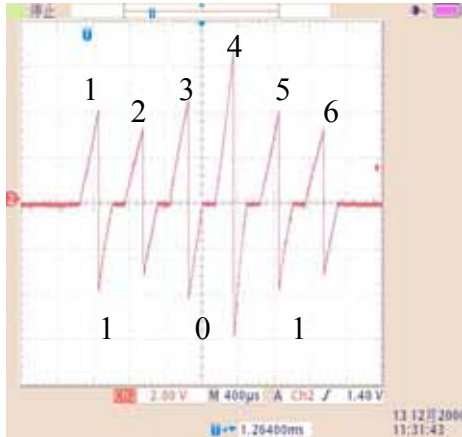


Fig. 6.9 Rotor position “5”.

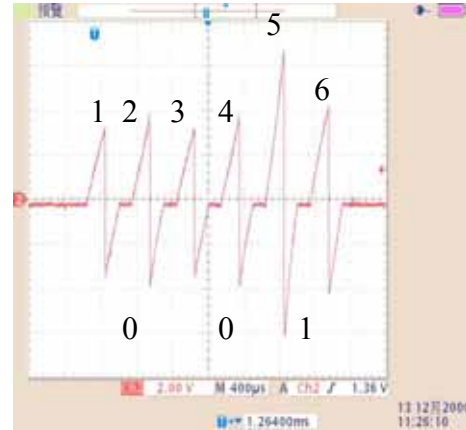


Fig. 6.10 Rotor position “1”.

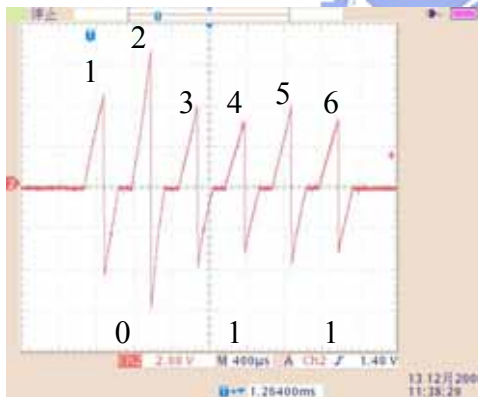


Fig. 6.11 Rotor position “3”.

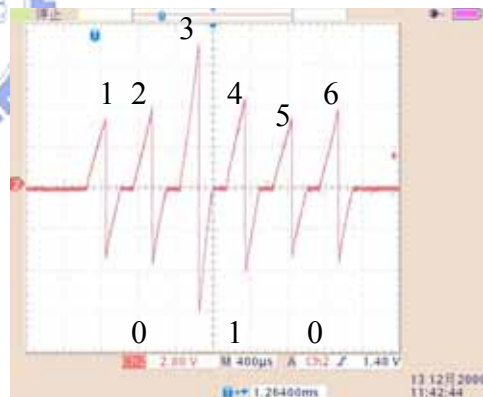


Fig. 6.12 Rotor position “2”.

After getting the rotor positions, we can send the right energizing currents to run the motor as shown in Fig. 6.15 and Fig. 6.16. Fig. 6.15 shows the current wave forms on DC bus when the motor is running, the rotor detecting procedures are executed every 7 ms approximately. This detecting procedure can be only applied for the low speed region and zero speed. In a higher speed region the back EMF detecting method can be applied. Fig. 6.16 shows the motor phase

currents and the detecting currents .

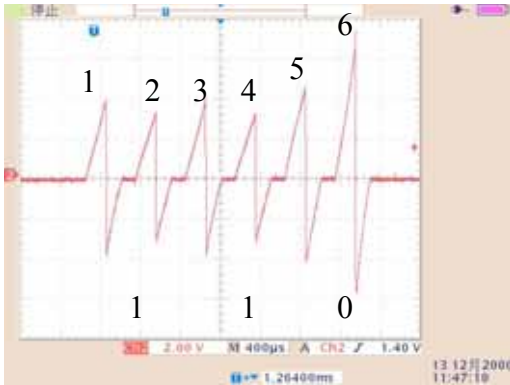


Fig. 6.13 Rotor position "6".

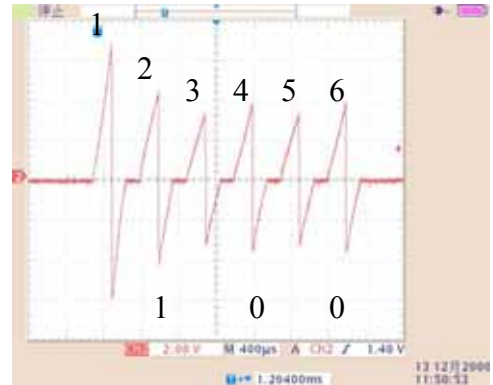


Fig. 6.14 Rotor position "4".

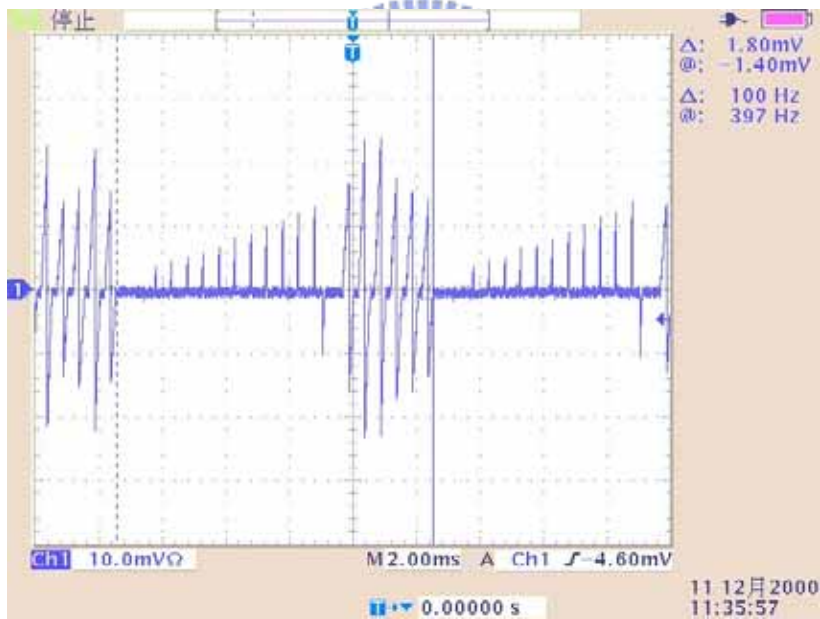


Fig. 6.15 Detecting current wave forms on DC bus.

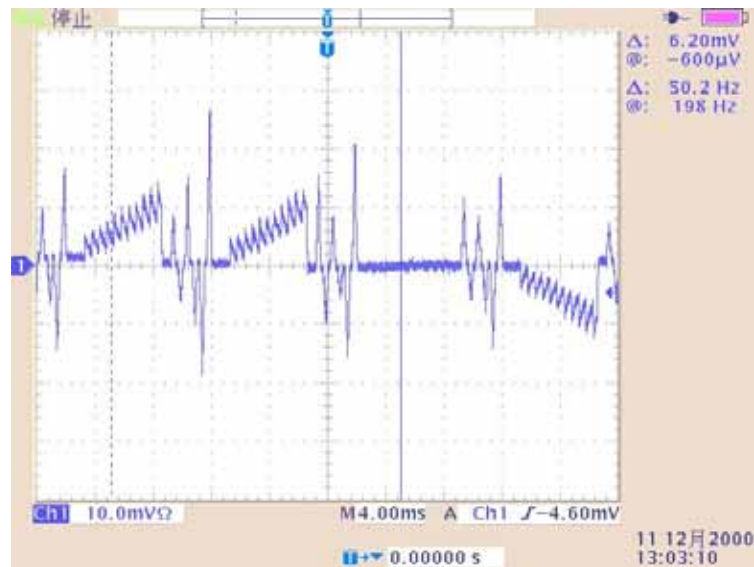


Fig. 6.16 Detecting current wave forms on motor phases.

6.2. Method 2: Rotor alignment

Giving a wrong initial value for the EKF, like the initial rotor position, could lead to convergence problem. To get the correct initial rotor position, we can align rotor d -axis in stator “a” axis (or U axis) as shown in Fig. 6.3. This technique can also be applied for starting a PMSM when using an incremental encoder. Fig. 6.18 shows the applied U and V phase currents and rotor speed response of aligning the rotor. Of course the current on U phase is two times the amplitude of V phase and W phase. The rotor is aligned at U axis after about 0.8 second. Fig. 6.17 is time magnified of Fig. 6.18.

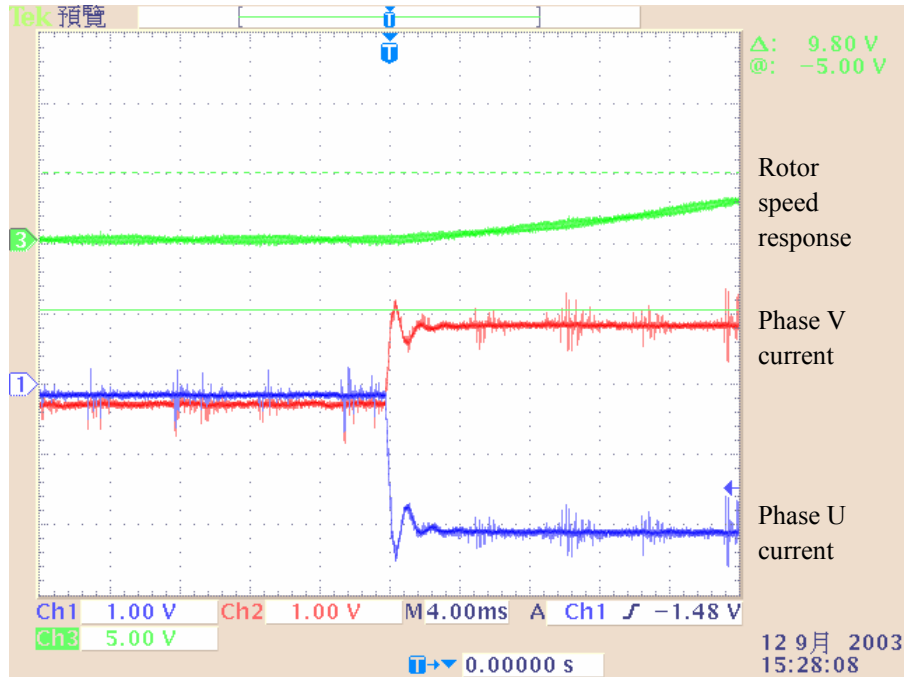


Fig. 6.17 Time scaled aligning current wave forms.

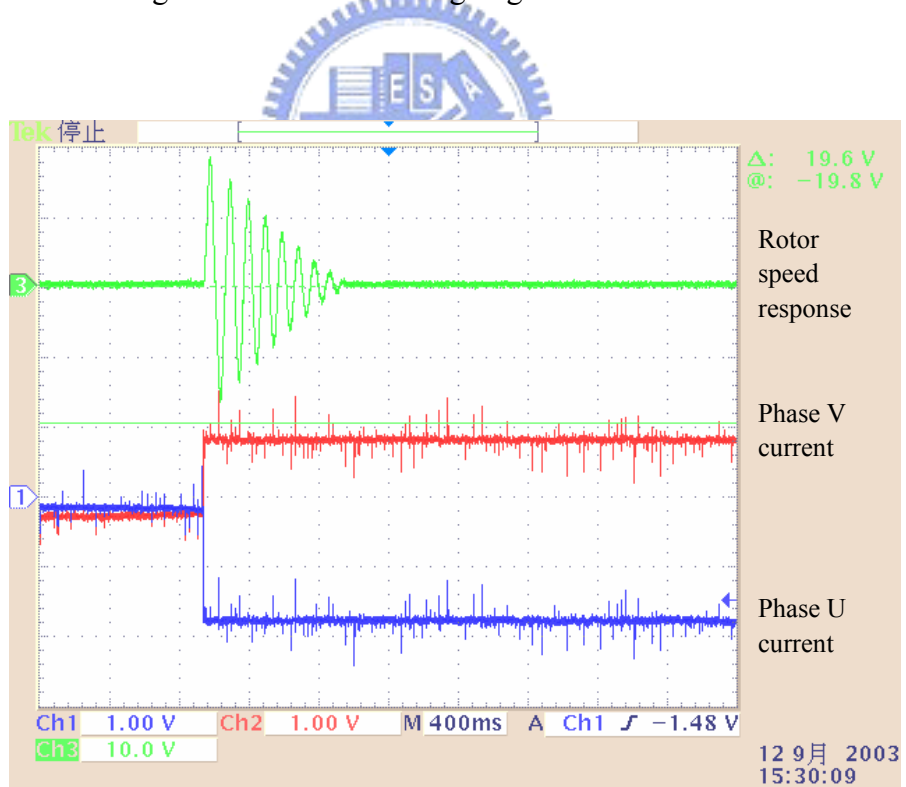


Fig. 6.18 Aligning current wave forms.

6.3. Experimental results of EKF

To implement the EKF algorithm, the TI DSP TMS320LF2407A has been used at its maximum speed, 40MHz operating frequency. It has a fast D/A conversion time up to 500ns and thus helps reduce sampling time for the motor phase currents. For monitoring those variables in the software, an RS232 communication is established. The sampling rate for those variables is nearly 1 KHz and they can be presented in wave forms as shown in Fig. 6.20.

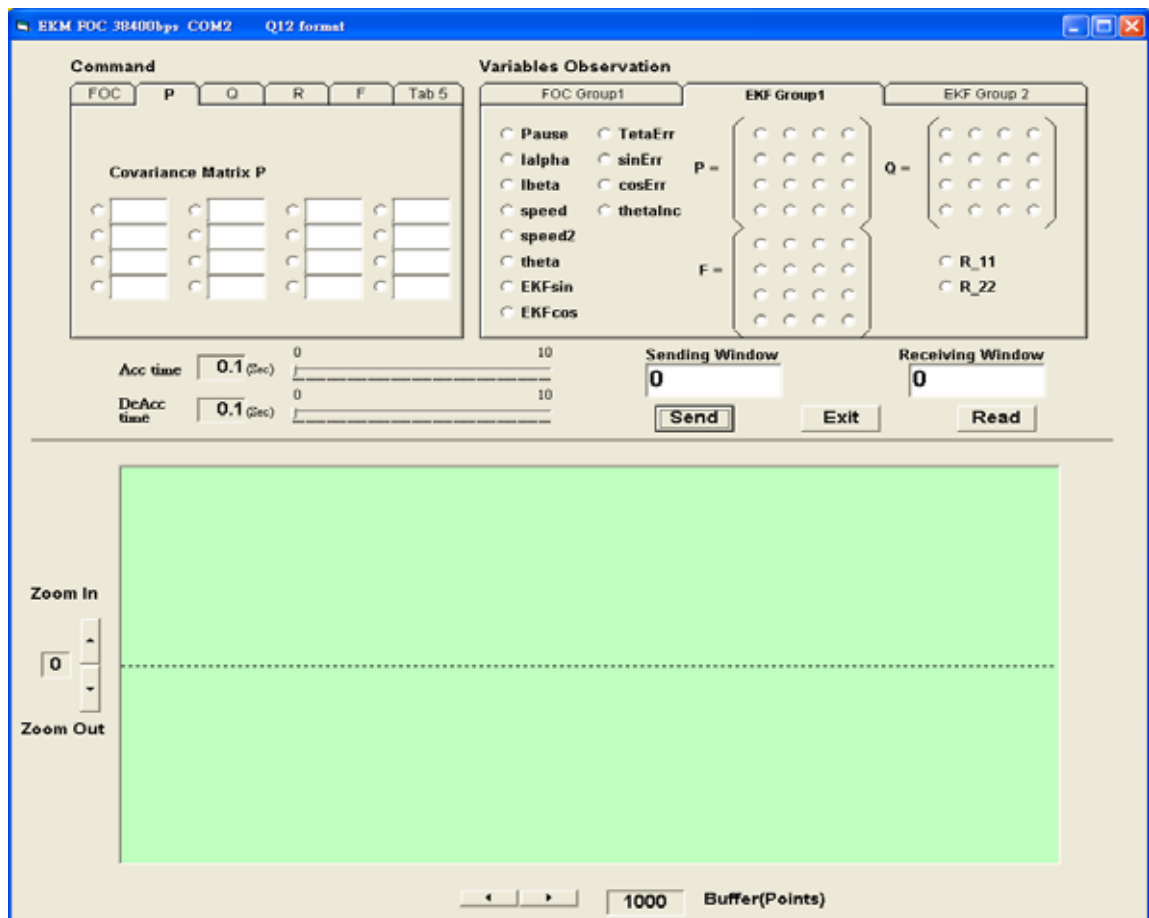


Fig. 6.19 Host communication software.

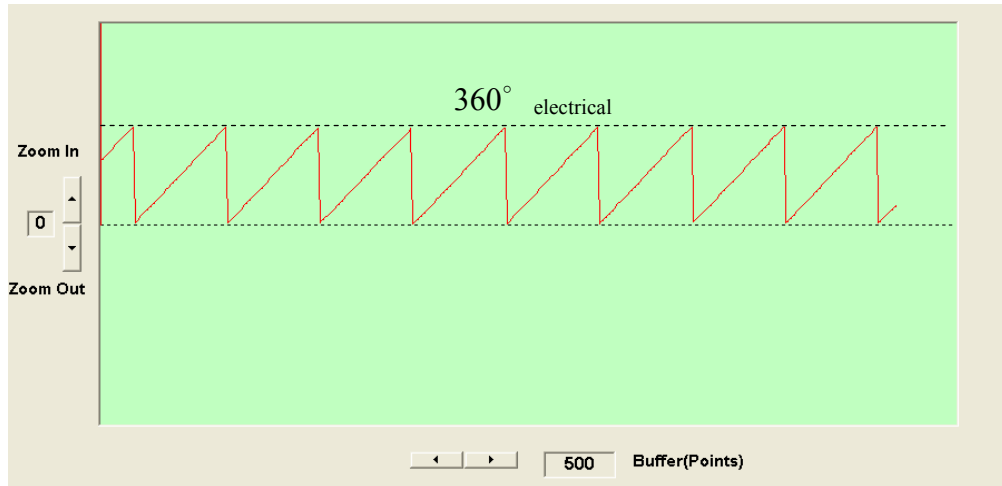


Fig. 6.20 Actual rotor position.

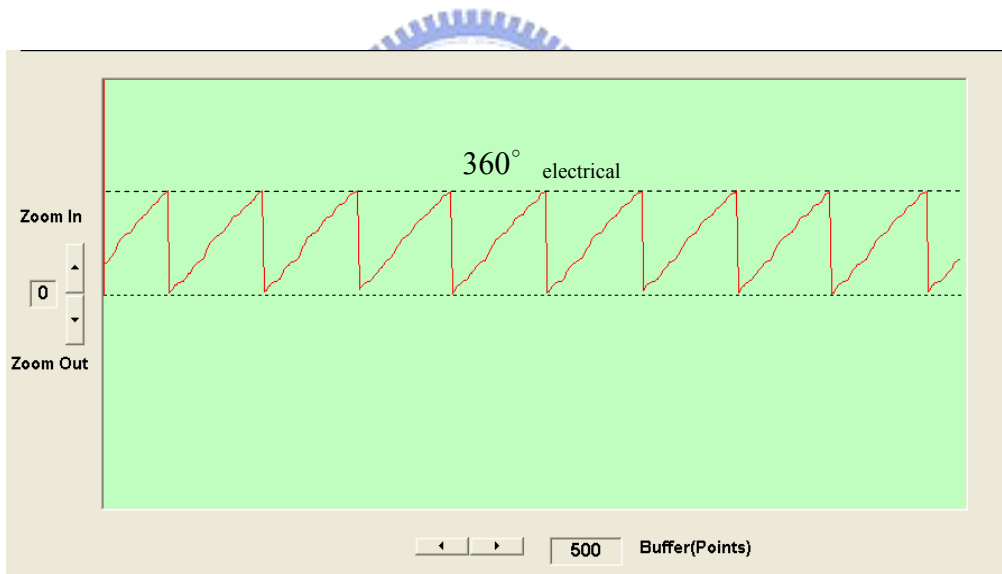


Fig. 6.21 Estimated rotor position.

Fig. 6.21 shows the actual rotor position at quadrant current command (i_{qref}) 0.4A. The EKF subroutine is executed every 200us to estimate rotor positions and speeds. The motor still uses the encoder feedback instead because we need to tune the covariant matrices P , Q and R . After the estimated parameters matches the actual ones then we can use the estimated θ_e and ω_e instead of the encoder feedback.

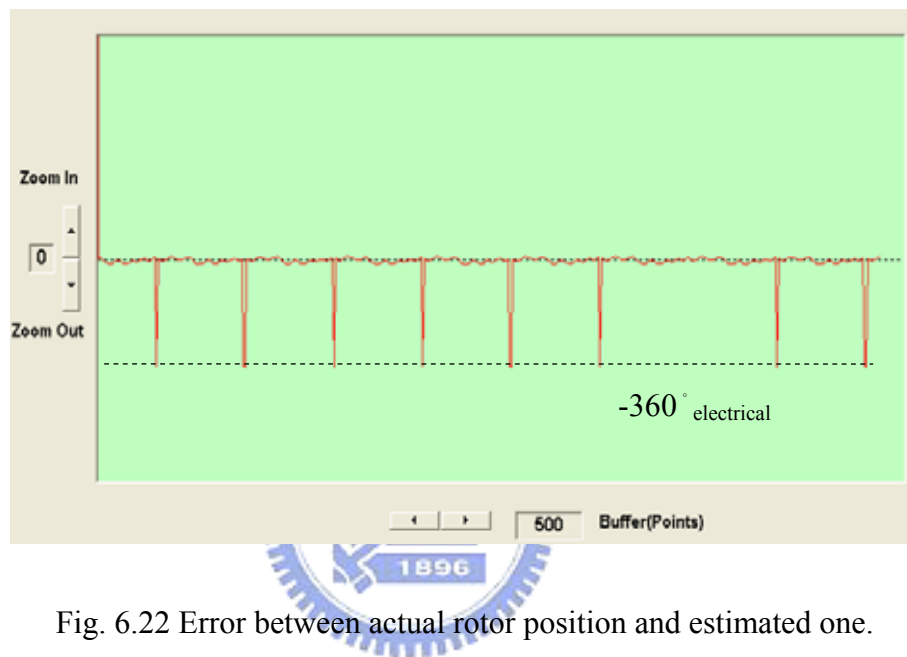


Fig. 6.22 Error between actual rotor position and estimated one.

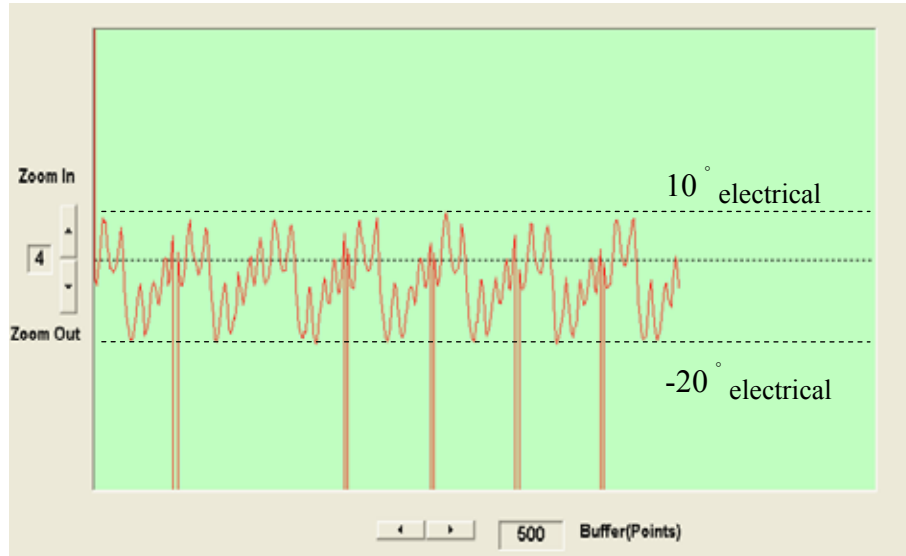


Fig. 6.23 Enlarged wave form of Fig. 6.22.

Tuning those covariance matrices is important and not a easy work for EKF in this experiment. It affects estimated motor speed and rotor position greatly. It needs a trial-and-error procedure to get a tradeoff between stability and convergence time. After the estimated rotor positions have been derived, it will be used to calculate the value of sine and cosine for coordinate transformations as shown in Fig. 6.26. Fig. 6.26 shows the error between actual $\sin(\theta)$ and estimated one. The maximum error is about 0.2 due to the estimated error of rotor position. In the trial-and-error procedure, the covariant matrix P_0 seems not to affect the estimated parameters too much but the covariance matrices Q and R do. Fig. 6.27 and 6.28 show the different responses due to different Q (3, 3). In the experiment, the estimated ω_e sometimes has the opposite direction compared to the actual one. This is due to the inherent ambiguity of the non-linear model. $-\omega_e$ and $+\omega_e$ give the same results.

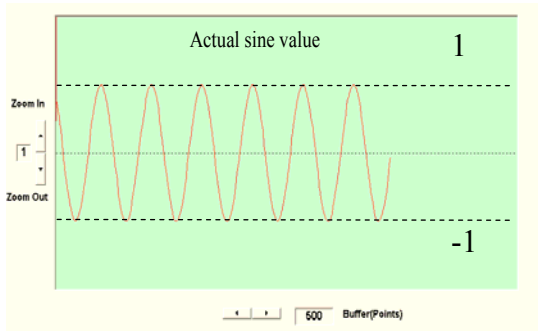


Fig. 6.24 Actual $\sin(\theta)$.

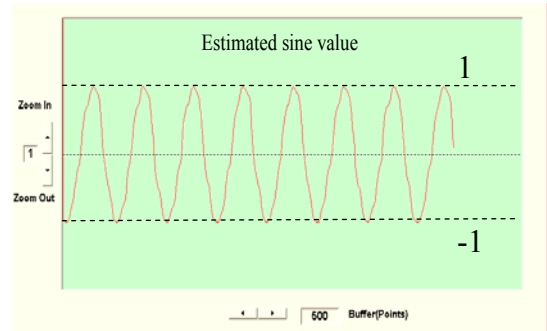


Fig. 6.25 Estimated $\sin(\theta)$.

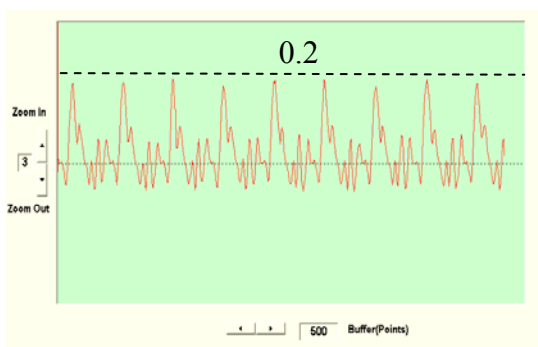


Fig. 6.26 Error between actual $\sin(\theta)$ and estimated one.

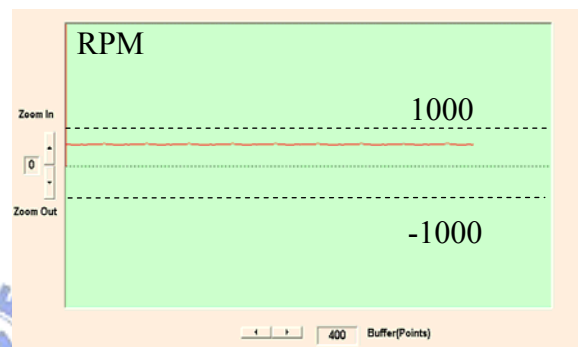


Fig. 6.27 Estimated speed $\omega_e = 550$ RPM.
 $Q(3, 3) = 0.06$.

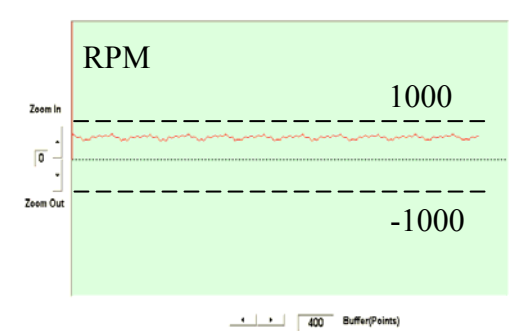


Fig. 6.28 Estimated speed $\omega_e = 550$ RPM
 $Q(3, 3) = 2$.

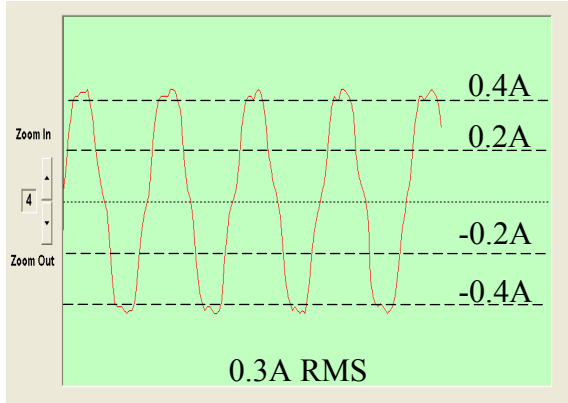


Fig. 6.29 Actual I_α .

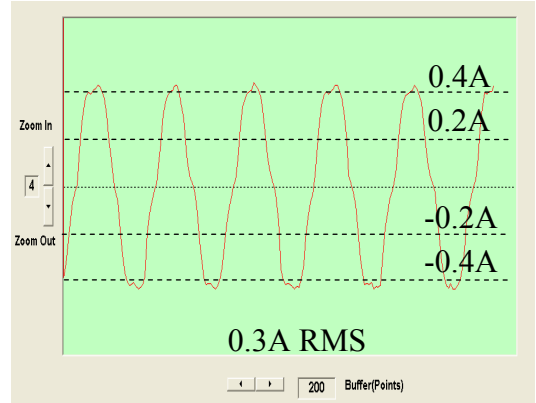


Fig. 6.30 Estimated I_α .

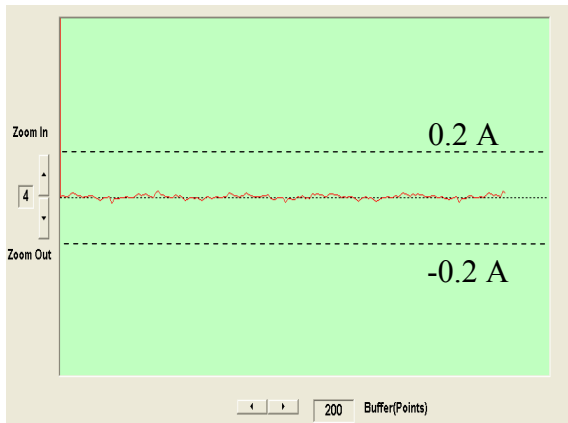


Fig. 6.31 Actual I_α – estimated one.

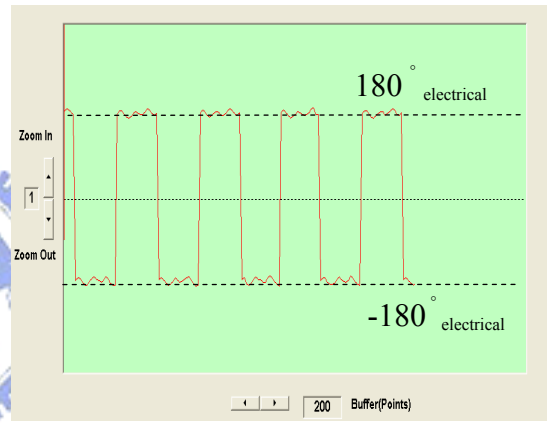


Fig. 6.32 Theta error after ω_e direction has been corrected.

To overcome the wrong estimated speed direction, a strategy has been adopted as shown in (6.1).

$$\text{If } (\theta_{e_{k|k-1}} - \theta_{e_{k-1|k-2}}) \times \omega_{e_{k|k-1}} < 0, \text{ then } \omega_{e_{k|k-1}} = -\omega_{e_{k|k-1}} \text{ and } \theta_{e_{k|k-1}} = \theta_{e_{k|k-1}} - \pi. \quad (6.1)$$

Fig.6.32 shows the result using (6.1), the estimated rotor position is always positive and negative 180 degrees alternately shifted from the actual one. This problem is still under researching. Currently, the EKF is executed every 200us and the current loop is 100us. According to the covariance matrices of the simulation results, Q12 format is chosen for real

time implement using a fixed-point DSP TMS320LF2407A. The sampling time interval is set to be 0.0002 second and that is nearly the resolution for Q12 format. The covariance matrices of the DSP software is

$$P = \begin{bmatrix} 0.1 & 0 & 0 & 0 \\ 0 & 0.1 & 0 & 0 \\ 0 & 0 & 200 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad Q = \begin{bmatrix} 0.4 & 0 & 0 & 0 \\ 0 & 0.4 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 2.5 \end{bmatrix}, \quad R = \begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix}. \quad (6.2)$$

Q15 format is preferred to increase the calculation precision, and care must be taken for the dynamic range of the covariance matrices and saturation problems.



Chapter 7

Conclusions

In this research, we find it is feasible to implement real-time EKF estimation of the rotor position and speed of a sinusoidal PMSM by using a low-cost single-chip DSP controller. The DSP can afford enough power for the matrices calculation of the EKF; it takes less than 100us cycle time by the software. The drive does not use the voltage transducers for the motor phase voltages and thus simplifies the drive and reduces the cost, so the drive is the same with the ordinary servo drives in the market unlike the other methods need the voltage transducers. This means using EKF is competitive in the cost point of view.

Due to the nonlinearity of the state equations, the estimated speed could be in the wrong direction and solving it is kind of tricky. Another consideration is that the covariance matrices could be in a value of big variation, which means we could not use the same Q format throughout all the calculations and this makes the software more complicated. The experiment results show that the maximum estimation error of the rotor position is -20 electrical degrees in steady state which differs from the results of the simulation. In the simulation the motor parameters are the same in the motor block and EKF, and all the calculations are based on the floating point method, but there are parameter errors between the actual motor and software, sampling errors, quantization errors and the characteristic of those covariance matrices P , Q and R are not well known in the experiment. In the future, the PWM will be used instead of the RMS value for the motor three-phase voltages and fixed-point method will be used instead of the floating-point method in the simulation to present the actual circumstances as similar as possible in order to understand the differences which can be a reference to improve the sensorless drive.

Appendix

The technical data for the motor in the experiment are as below:

Rated power:	90W/3000 RPM
Pole number:	4
Stator phase resistance:	3.4 Ω
Stator inductance L_d :	9mH
Stator inductance L_q :	12mH
Flux linkage:	0.11327 Wb-T



References

- [1] Silverio Bolognani, Roberto Oboe, and Mauro Zigliotto, "Sensorless full-digital PMSM drive with EKF estimation of speed and rotor position," *IEEE Trans. Ind. Electron.*, vol. 46, pp. 184-191. Feb. 1999.
- [2] M. A. Jabbar, M. A. Hoque, and M. A. Rahman, "Sensorless permanent magnet synchronous motor drives," *IEEE CCECE'97*, 1997, pp. 878-883.
- [3] Kazutaka Tatematsu, Daisuke Hamada, and Kenji Uchida, "Sensorless control for permanent magnet synchronous motor with reduced order observer," in *Proc. of The 29th IEEE Power Electronics Specialists Conf. (PESC'98)*, 1998, pp. 125-131.
- [4] "Sensorless control with Kalman filter on TMS320 fixed-point DSP (Literature Number: BPRA057)," Texas Instruments Europe, 1997.
- [5] Yuan-Rui Chen, Norbert C. Cheung, and Jie Wu, "Sensorless drive of permanent magnet linear motors using modified Kalman filter," *IEEE Power Electronics Specialist Conf., PESC'2001*, 2001, pp.17-21.
- [6] Mohand A. Ouhrouche, "EKF-Based estimation of rotor flux, speed and rotor resistance in cage induction motor sensorless drive," *Proceeding of the IASTED International Conf. Modeling and Simulation (MS'2000)*, May 15-17, 2000 - Pittsburgh, Pennsylvania, USA.
- [7] Bozo Terzic and Martin Jadric, "Design and implementation of the extended Kalman filter for the speed and rotor position estimation of brushless DC motor," *IEEE Trans. Ind. Electron.*, vol. 48, no. 6, pp. 1065-1073, December 2001.
- [8] A. Bado, S. Bolognani, and M. Zigliotto, "Effective estimation of speed and rotor position of a PM synchronous motor drive by a Kalman filtering technique," *Proc. IEEE-PESC*, vol. 2, pp. 951-957, 1992.
- [9] Peter Vas, *Sensorless Vector and Direct Torque Control*, Oxford, New York, 1998.
- [10] Chee-Mun Ong, *Dynamic Simulation of Electric Machinery Using Matlab/Simulink*, Prentice Hall, New Jersey, 1998.
- [11] *Power System Blockset User's Guide*, TEQSIM International Inc., 2000.
- [12] Yasuhiko Dote and Sakan Kinoshita, *Brushless Servomotors Fundamentals and Applications*, Clarendon Press. Oxford, 1990.
- [13] Duane C. Hanselman, *Brushless Permanent-Magnet Motor Design*, McGraw-Hill

Inc., 1994.

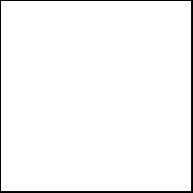
- [14] Peter Vas, *Parameter Estimation, Condition Monitoring, and Diagnosis of Electrical Machines*, Clarendon Press. Oxford, 1993.



Author Resume

姓名: 葛育中
出生: 1972/2/22
求學歷程: 逢甲大學 自動控制工程
 台中一中
經歷: 現任職於愛德利科技股份有限公司(1996 至今)
主要作品: 電動機車無刷馬達控制器
 工業用無刷馬達控制器 D305 系列





2.5 cm	92			92
畢業 民國 年度				
1 cm	碩 士 論 文			碩 士 論 文
2.5 cm	卡 曼 濾 波 器 應 用 於 永 磁 同 步 馬 達 無 感 測 控 之 研 究			卡 曼 濾 波 器 應 用 於 永 磁 同 步 馬 達 無 感 測 控 之 研 究
論 文 題 目				
				
校 院 所 名	交 通 大 學 電 機 資 訊 學 院 電 機 與 控 制 學 程			交 通 大 學 電 機 資 訊 學 院 電 機 與 控 制 學 程
3 cm				
著 者 姓 名	葛 育 中			葛 育 中
1 cm				
2 cm				