

Czech technical university in Prague  
Faculty of Mechanical Engineering  
Department of Instrumentation and Control Engineering



# Sensorless vector control of synchronous permanent magnet motor implemented on DSP

Master's thesis

*Simon Kalombo*

Master's program: Automation and Instrumentation Engineering  
Master's branch: Automation and Industrial Informatics  
Supervisor: Ing. Zdeněk Novák, Ph.D.

Prague, June 2022



# Assignment



## ZADÁNÍ DIPLOMOVÉ PRÁCE

### I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Kalombo** Jméno: **Simon** Osobní číslo: **473687**  
 Fakulta/ústav: **Fakulta strojní**  
 Zadávající katedra/ústav: **Ústav přístrojové a řídicí techniky**  
 Studijní program: **Automatizační a přístrojová technika**  
 Specializace: **Automatizace a průmyslová informatika**

### II. ÚDAJE K DIPLOMOVÉ PRÁCI

Název diplomové práce:

**Bezsenzorové vektorové řízení synchronního motoru s permanentními magnety realizované na DSP**

Název diplomové práce anglicky:

**Sensorless vector control of synchronous permanent magnet motor implemented on DSP**

Pokyny pro vypracování:

- 1) Seznamte se s metodami bezsenzorového vektorového řízení synchronního motoru s permanentními magnety (PMSM). Zaměřte se na pozorovatele stavu, konkrétně na tzv. Rozšířený Kálmánův Filtr (EKF).
- 2) Vytvořte simulační model v prostředí Matlab Simulink, který bude EKF pro bezsenzorové řízení PMSM využívat.
- 3) Využijte získané poznatky a naprogramujte řídicí algoritmus do vhodného digitálního signálového procesoru (DSP).
- 4) Řízení experimentálně ověřte a zhodnoťte dosažené výsledky (např. reálné chování stroje vůči výsledkům simulace). Zhodnoťte rychlost výpočtu řídicí smyčky na DSP.

Seznam doporučené literatury:

- [1] BUSO, Simone a Paolo MATTARELLI. Digital Control in Power Electronics. San Rafael, CA: Morgan & Claypool, 2006. ISBN 9781598291131.  
 [2] GLUMINEAU, Alain a Jesús de León MORALES. Sensorless ac electric motor control. New York, NY: Springer Berlin Heidelberg, 2015. ISBN 978-331-9145-853.  
 [3] N. K. Quang, N. T. Hieu and Q. P. Ha, "FPGA-Based Sensorless PMSM Speed Control Using Reduced-Order Extended Kalman Filters," in IEEE Transactions on Industrial Electronics, vol. 61, no. 12, pp. 6574-6582, Dec. 2014, doi: 10.1109/TIE.2014.2320215.

Jméno a pracoviště vedoucí(ho) diplomové práce:

**Ing. Zdeněk Novák, Ph.D. U12110.1**

Jméno a pracoviště druhého(ho) vedoucí(ho) nebo konzultanta(ky) diplomové práce:

Datum zadání diplomové práce: **29.04.2022**

Termín odevzdání diplomové práce: **09.06.2022**

Platnost zadání diplomové práce:

Ing. Zdeněk Novák, Ph.D.  
podpis vedoucí(ho) práce

podpis vedoucí(ho) ústavu/katedry

doc. Ing. Miroslav Španiel, CSc.  
podpis děkana(ky)

### III. PŘEVZETÍ ZADÁNÍ

Diplomant bere na vědomí, že je povinen vypracovat diplomovou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v diplomové práci.

29.4.2022  
Datum převzetí zadání

Kalombo  
Podpis studenta

**Supervisor:**

Ing. Zdeněk Novák, Ph.D.  
Department of Instrumentation and Control Engineering  
Faculty of Mechanical Engineering  
Czech technical university in Prague  
Technická 4  
160 00 Praha 6  
Czech Republic

# Declaration of Authorship

I hereby confirm that this Master's thesis and the work presented in it is entirely my own. Where I have consulted the work of others this is always clearly stated. All statements taken literally from other writings or referred to by analogy are marked and the source is always given. This paper has not yet been submitted to another examination office, either in the same or similar form. I also agree with the potential publication of the results of the thesis or of its substantial part, provided I will be listed as the co-author

In Prague 2022

.....  
Simon Kalombo

# Abstract

**Abstract** This Master's thesis deals with the design of sensorless vector control of permanent magnet synchronous motors (PMSM). In the introductory part, a research on sensorless algorithms was made. The used sensorless algorithm for position and velocity estimation, which is mainly the focus of this thesis, is called the extended Kalman filter. This algorithm estimates the back electromotive force (back EMF) from which it determines the rotor position and velocity. The knowledge gained from this research was subsequently used to create simulations. Based on the results of the simulations, the strategy was implemented on a digital signal processor (DSP). In the last part of this thesis, the results obtained from testing on a real permanent magnet synchronous motor are evaluated.

**Keywords:** PMSM, sensorless, vector control, extended Kalman filter, DSP, back EMF

**Abstrakt** Tato diplomová práce se zabývá návrhem bezsenzorového vektorového řízení synchronních motorů s permanentními magnety (PMSM). V úvodní části se věnuje rešerši týkající se bezsenzorových algoritmů. Algoritmus bezsenzorového odhadu polohy a rychlosti, kterému se tato diplomová práce převážně věnuje, se nazývá rozšířený Kálmánův filtr. Tento algoritmus odhaduje zpětné elektromotorické napětí, ze kterého určuje polohu a rychlost rotoru. Poznatky získané touto rešerší byly následně využity při vytváření simulací. Na základě výsledků simulací byla strategie implementována na digitální signálový procesor. V poslední části této práce jsou zhodnoceny výsledky získané při testování na reálném synchronním motoru s permanentními magnety.

**Klíčová slova:** synchronní motor s permanentními magnety, bezsenzorové řízení, vektorové řízení, rozšířený Kálmánův filtr, DSP, zpětné elektromotorické napětí

# Acknowledgments

I would like to thank my supervisor Ing. Zdeněk Novák, Ph.D. for excellent guidance during the development of this thesis. I would also like to thank my family and loved ones for their endless support during my studies.

# List of Tables

4.1	EKF computation times . . . . .	36
4.2	Startup constants setup . . . . .	40
5.1	Motor parameters . . . . .	48
5.2	PWM configuration . . . . .	49
5.3	ADC configuration . . . . .	50
5.4	PI regulators' gains . . . . .	50
5.5	PI regulators' saturation . . . . .	51
6.1	Estimated speed mean and standard deviation . . . . .	53



# List of Figures

2.1	PMSM rotor types (a)Surface-mounted magnets (b)Interior magnets [3] . . .	2
2.2	PMSM V/f scalar control . . . . .	3
2.3	Vector sensorless control scheme for SPMSM . . . . .	5
2.4	PMSM reference frames [4] . . . . .	5
3.1	Sensorless estimation methods . . . . .	10
3.2	Back EMF method . . . . .	10
3.3	Flux linkage method . . . . .	12
3.4	Flux angle synthesis [2] . . . . .	13
3.5	Low-pass filters stator flux linkage estimation [2] . . . . .	14
3.6	Saliency-based method . . . . .	15
3.7	Hypothetical reference frame method . . . . .	16
3.8	A hypothetical rotor reference frame in the counterclockwise direction of rotation [2] . . . . .	16
3.9	Observer-based method . . . . .	17
3.10	Sliding-mode observer [16] . . . . .	18
3.11	EKF algorithm flow chart . . . . .	19
3.12	EKF algorithm as part of motor controller . . . . .	21
4.1	PMSM controller . . . . .	23
4.2	Current control with startup . . . . .	23
4.3	Current control system . . . . .	24
4.4	Current controllers . . . . .	24
4.5	PI regulator scheme for $i_d$ current . . . . .	25
4.6	Startup algorithm scheme . . . . .	26
4.7	Startup algorithm state diagram . . . . .	26
4.8	Startup position generator . . . . .	27
4.9	Startup current generator . . . . .	27
4.10	Startup error . . . . .	28
4.11	Speed controller . . . . .	28
4.12	Speed PI regulator . . . . .	29
4.13	SPWM algorithm implementation . . . . .	30
4.14	SPWM comparison . . . . .	31
4.15	Generated PWM signals . . . . .	31
4.16	Dead-time . . . . .	32
4.17	Average-value inverter . . . . .	33
4.18	Variable switching inverter . . . . .	33
4.19	SPMSM state-space model . . . . .	34
4.20	SPMSM Motor control blockset . . . . .	34

4.21	SPMSM high-fidelity model . . . . .	34
4.22	Parallel EKF flow-chart . . . . .	37
4.23	EKF parallel . . . . .	38
4.24	QPLL . . . . .	39
4.25	PLL . . . . .	39
4.26	EKF with QPLL . . . . .	39
4.27	Rotor speed during startup . . . . .	41
4.28	I/f startup position error and currents . . . . .	41
4.29	I/f startup position error (detail) . . . . .	42
4.30	SPMSM sensorless controller - speed and $dq$ currents . . . . .	43
4.31	Error between real and estimated rotor speed . . . . .	43
4.32	SPMSM sensorless controller - abc voltages . . . . .	44
4.33	SPMSM sensorless controller - abc currents . . . . .	44
4.34	SPMSM sensorless controller - high fidelity simulation . . . . .	45
5.1	Validation setup . . . . .	46
5.2	Delfino TMS320F28379D controlCARD . . . . .	47
5.3	PWM scaling . . . . .	49
5.4	PWM implementation . . . . .	49
5.5	ADC measurement . . . . .	50
5.6	Dual-core DSP partitioning . . . . .	51
6.1	Real PMSM rotor speed estimation . . . . .	52
6.2	Estimated speed analysis . . . . .	53
6.3	Measured d-q currents . . . . .	54
6.4	Measured ab currents . . . . .	54
6.5	3-phase reference voltage . . . . .	55
6.6	Real vs. simulated rotor speed estimation . . . . .	55
6.7	Real vs. simulated d-q currents . . . . .	56
6.8	Execution times . . . . .	57
6.9	Validation setup . . . . .	57

# List of acronyms

**AC** Alternating current. 3, 4, 29, 30

**back EMF** Back electromotive force. 1, 10–12, 18, 22, 25, 35, 37–39, 52, 58

**BLDC** Brushless direct current. 10

**CPU** Central processing unit. 47, 51, 56

**DC** Direct current. 1, 4, 10, 13, 23, 29, 32, 50

**DSP** Digital signal processor. x, 1, 22, 35, 46, 47, 51

**EKF** Extended Kalman filter. viii–x, 1, 19, 21, 22, 35–39, 43, 45, 58

**FOC** Field oriented control. 4

**FPGA** Field-programmable gate array. 35

**IGBT** Insulated Gate Bipolar Transistor. 33, 45

**IPC** Internal processor communication. 51

**IPMSM** Interior permanent magnet synchronous motor. 3, 7, 12, 21

**MTPA** Maximum torque per ampere. 7

**PIL** Processor-in-the-loop. 35

**PLL** Phase-locked loop. x, 37–39

**PMSM** Permanent magnet synchronous motor. ix, x, xii, xiii, 1–10, 16–19, 22, 23, 25, 30, 33, 52, 53, 55, 56

**PWM** Pulse width modulation. viii–x, 3, 24, 29–32, 48–50, 54, 56

**QPLL** Quadrature phase-locked loop. x, xiii, 38, 39

**RPM** Revolutions per minute. 30, 52, 53

**SMO** Sliding-mode observer. 18

**SPMSM** Surface-mount permanent magnet synchronous motor. ix, x, xiii, 2, 5, 7, 12, 22, 29, 34, 42–48, 58, 59

**SPWM** Sinusoidal pulse width modulation. ix, 29–31

# Contents

Assignment	iii
Declaration of Authorship	v
Abstract	vi
Acknowledgments	vii
List of Tables	viii
List of Figures	ix
List of Acronyms	xi
<b>1 Introduction</b>	<b>1</b>
<b>2 Vector control</b>	<b>2</b>
2.1 Permanent magnet synchronous motor . . . . .	2
2.2 Scalar V/f control . . . . .	3
2.3 Vector control . . . . .	4
2.3.1 The control scheme of the vector control . . . . .	4
2.3.2 Clarke-Park Transformation . . . . .	5
2.3.3 Torque control . . . . .	6
2.3.4 Mathematical model of PMSM . . . . .	7
2.3.5 The mathematical model in $d, q$ rotating reference frame . . . . .	7
<b>3 Sensorless control</b>	<b>9</b>
3.1 Back EMF method . . . . .	10
3.1.1 Two-axis stationary reference frame . . . . .	11
3.1.2 Machine variables scheme . . . . .	12
3.2 Flux linkage method . . . . .	12
3.2.1 Integration scheme . . . . .	13
3.2.2 Low-pass filter scheme . . . . .	13
3.2.3 Flux linkage speed estimation scheme . . . . .	14
3.2.4 Torque angle estimation scheme . . . . .	15
3.3 Saliency-based method . . . . .	15
3.4 Hypothetical reference frame method . . . . .	16
3.5 Observer-based method . . . . .	17
3.5.1 State observer scheme . . . . .	17
3.5.2 Sliding-mode observer . . . . .	18

3.5.3	Extended Kalman filter . . . . .	19
<b>4</b>	<b>Simulation</b>	<b>22</b>
4.1	Simulation overview . . . . .	22
4.2	Vector control . . . . .	23
4.2.1	Current regulation . . . . .	23
4.2.2	I/f startup algorithm for the PMSM . . . . .	25
4.2.3	Speed regulation . . . . .	28
4.2.4	Pulse width modulation . . . . .	29
4.2.5	Inverter . . . . .	32
4.2.6	PMSM . . . . .	33
4.3	Extended Kalman filter . . . . .	35
4.3.1	Implementation . . . . .	35
4.3.2	Quadrature phase-locked loop (QPLL) . . . . .	38
4.4	Simulation results . . . . .	40
4.4.1	I/f startup method . . . . .	40
4.4.2	Complete SPMSM sensorless controller . . . . .	42
4.4.3	High fidelity simulation . . . . .	45
<b>5</b>	<b>Hardware implementation</b>	<b>46</b>
5.1	Digital signal processor . . . . .	47
5.2	Permanent magnet synchronous motor . . . . .	48
5.3	Single core implementation . . . . .	48
5.3.1	Pulse width modulation . . . . .	48
5.3.2	Analog to digital converter . . . . .	50
5.3.3	PI regulators tuning . . . . .	50
5.4	Dual core implementation . . . . .	51
<b>6</b>	<b>Control strategy validation on real hardware</b>	<b>52</b>
<b>7</b>	<b>Conclusion</b>	<b>58</b>
	<b>Bibliography</b>	<b>62</b>
	<b>List of attachments</b>	<b>63</b>
	<b>List of used software</b>	<b>64</b>

# Chapter 1

## Introduction

Permanent magnet synchronous motors (PMSM) are nowadays one of the most used and researched electric motors. They have high power density and high efficiency. A popular research topic these days is their control. They are not as easily controllable as the DC brushed motors or induction motors. A complex control strategy is needed. Most used modern control strategies for the PMSMs are based on vector control. These control strategies are dependent on the knowledge of the rotor position. This information could be obtained using the position sensor or sensorless algorithms.

The main goal of this Master's thesis is to understand sensorless vector control of permanent magnet synchronous machines. This understanding is crucial to the design of this complex control strategy. The sensorless control is extensively researched nowadays. A huge amount of different strategies and approaches were published. This thesis mainly focuses on strategies based on the back electromotive force (back EMF) estimation and the extended Kalman filter (EKF) observers. The combination of the sensorless speed and position estimation algorithm with the vector control algorithm leads to the complex controller for PMSMs.

To properly validate the functions of the designed PMSM strategy, simulations need to be done. Simulations provide very important information about the controller being developed.

This PMSM controller will be deployed on the hardware. Selection of the right hardware could be crucial and must be taken into account in the implementation. The hardware selected for this thesis was a dual-core digital signal processor (DSP). This multi-core platform provides a lot of implementation options.

The final part of each motor control strategy development is the validation of the real hardware. This contains a fine-tuning to achieve the goal, a perfectly running motor.

# Chapter 2

## Vector control

### 2.1 Permanent magnet synchronous motor

Permanent magnet synchronous motors are one of the most researched electrical machines now. The development of PMSMs started in the 1970s as an evolution of the older electrical machines. Throughout the development, some advantages and also disadvantages emerged. The main vision was to start replacing old induction machines in the industry. This vision has the main issue. The cost of permanent magnet materials. Permanent magnet materials are made from rare earth metals. On the basis of that, the price of a permanent magnet on the rotor could easily reach 80%. [1], [2]

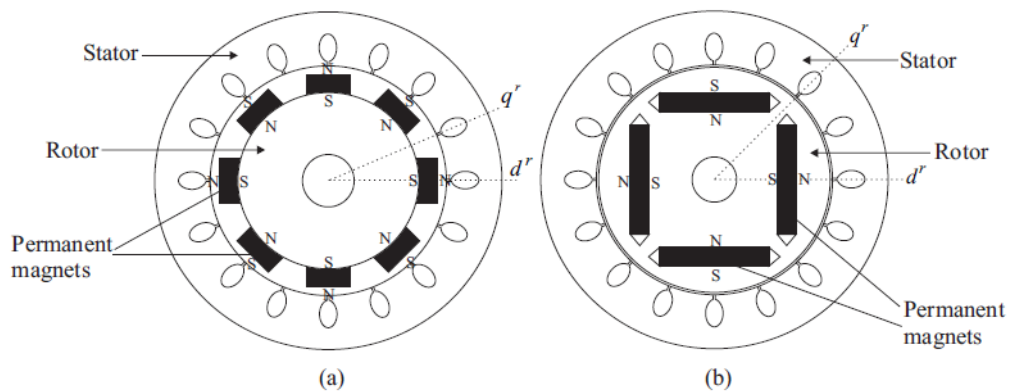


Figure 2.1: PMSM rotor types (a)Surface-mounted magnets (b)Interior magnets [3]

Permanent magnet synchronous motors can be divided into two main groups. These groups depend on the rotor configuration - topology (Figure 2.1). The first rotor configuration has a magnet mounted on the rotor surface. This configuration, called surface-mounted permanent magnet synchronous motor (SPMSM), is widely used from common to high-speed applications. The second type has permanent magnets implemented inside

of the rotor. This configuration is named interior permanent magnet synchronous motor (IPMSM). The main difference between these two configurations follows from the magnet position with respect to the stator winding. In the case of the IPMSM, magnets are not always at the same distance. This results in magnetic asymmetry, i.e., saliency. Saliency is characterized by the motor's ability to produce reluctance torque. The main field of use for salient IPMSM is in traction drive systems. [1]–[3]

## 2.2 Scalar V/f control

Scalar control (figure 2.2) is the simple control strategy for the PMSMs. It lies in keeping the constant ratio between voltage and frequency. This method is mainly used for AC induction motors control. Based on the frequency command, voltage magnitude is set with respect to the constant ratio. PWM algorithm generate 3-phase voltages with the desired frequency and magnitude. Rotor then rotates with the synchronous speed. No current and voltages vectors are controlled. This is why this method is called scalar. This method is suitable for applications, where is no need for the dynamic performance (ventilation, pumps, fans, etc.). PMSM under scalar V/f control will become unstable after exceeding a certain frequency.

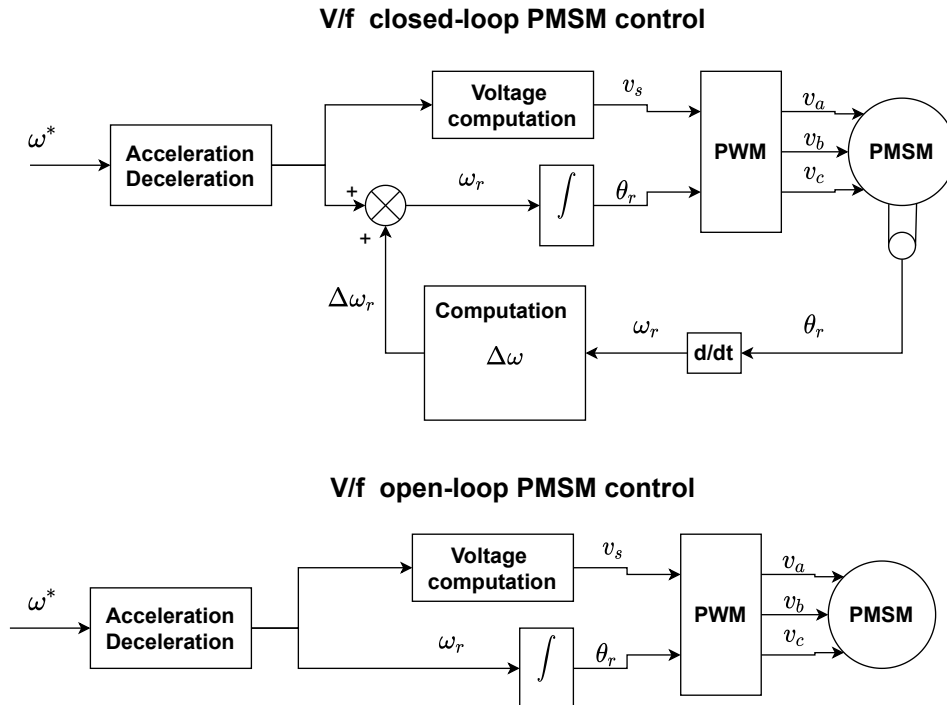


Figure 2.2: PMSM V/f scalar control



## 2.3 Vector control

Torque in any electrical machine is generated by the interaction of two magnetic fields. In other words, the torque is proportional to the outer product of the two magnetic flux linkage vectors. To achieve the maximum torque of the rotary electrical machine, magnetic flux linkage vectors must be perpendicular and independent. Separately excited DC machines reach this goal through their construction. Commutation inside DC machines holds flux linkage vectors perpendicular. On the other side, the AC machine doesn't fulfil this goal by its construction. The magnetic field vectors of AC machines are not perpendicular nor independent by their nature.

Vector control, in some literature called field-oriented control (FOC) [3], maintains both magnetic field vectors of the AC machine-independent and perpendicular. The result of this is maximum torque production. This makes AC vector controlled machines analogous to DC machines.

One of the most used approaches for Vector control of PMSMs is control in reference d-q frame. Three-phase currents are transformed into the reference frame  $d, q$ . This reference frame is aligned with the rotor magnetic flux vector and rotates. Currents in the d-q frame then behave like DC currents. The transformation from the a-b-c reference frame to  $d, q$  is done by a Clarke-Park transformation.[2], [3]

### 2.3.1 The control scheme of the vector control

The realisation of the vector control uses cascade control. The inner loop controls torque. Speed of PMSM is controlled by the outer loop. Regulation is done with PI regulators as shown in the figure 2.3.[2]

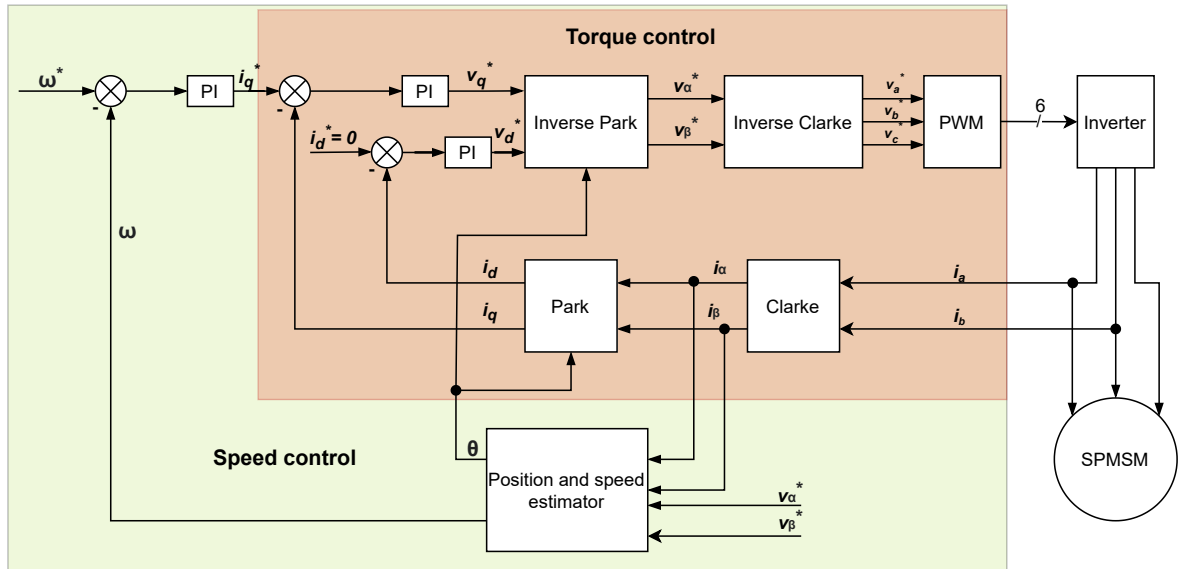


Figure 2.3: Vector sensorless control scheme for SPMSM

### 2.3.2 Clarke-Park Transformation

Current measurements from three-phase balanced systems are represented as  $i_a$ ,  $i_b$  and  $i_c$  with a sinusoidal course. For the controller, which is operating in a  $d, q$  reference frame, it is necessary to transform the measured  $i_a$ ,  $i_b$  and  $i_c$ . This is done by simple mathematical transformations. The same processes are needed in reverse for the transformation of  $i_d$  and  $i_q$  currents from the controller back to  $v_a$ ,  $v_b$  and  $v_c$  for the inverter.

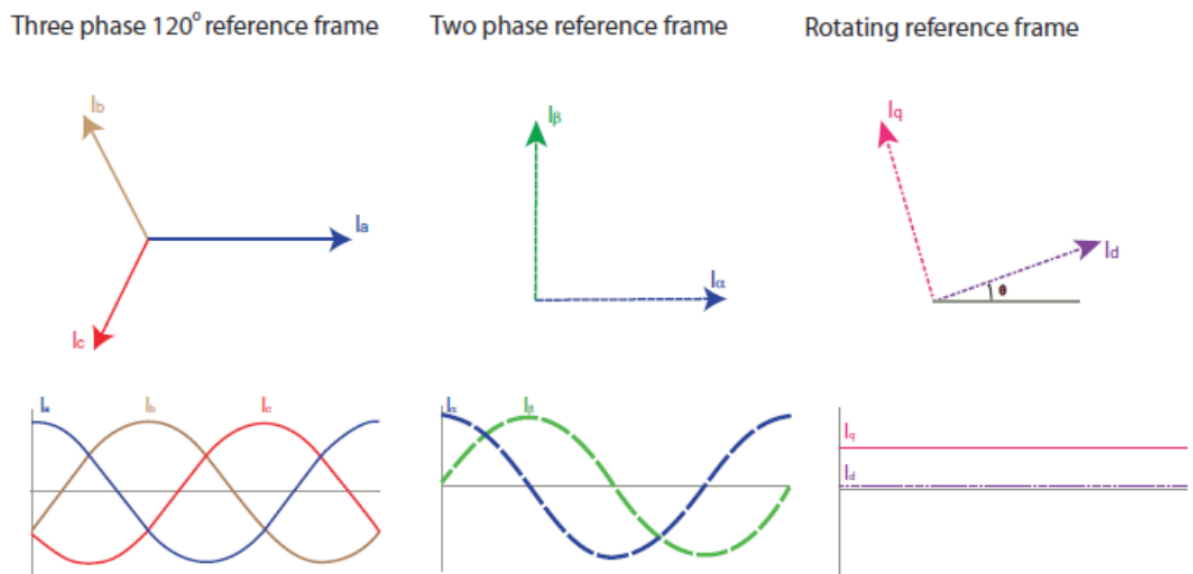


Figure 2.4: PMSM reference frames [4]

**Clarke transformation**

Clarke transformation performs the transition from abc to the  $\alpha - \beta$  reference frame.[2]

$$\begin{bmatrix} \alpha \\ \beta \\ 0 \end{bmatrix} = \frac{2}{3} \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} \quad (2.1)$$

**Inverse Clarke transformation**

Inverse Clarke transformation performs the transition from  $\alpha\beta$  to the abc reference frame.[2]

$$\begin{bmatrix} a \\ b \\ c \end{bmatrix} = \frac{2}{3} \begin{bmatrix} 1 & 0 & 1 \\ -\frac{1}{2} & \frac{\sqrt{3}}{2} & 1 \\ -\frac{1}{2} & -\frac{\sqrt{3}}{2} & 1 \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \\ 0 \end{bmatrix} \quad (2.2)$$

**Park transformation**

Park transformation performs the transition from  $\alpha\beta$  to the  $d, q$  reference frame.[2]

$$\begin{bmatrix} d, q \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix}, \quad (2.3)$$

where  $\theta$  is the electrical angle of the rotor.

**Inverse Park transformation**

Inverse Park transformation performs the transition from  $d, q$  to the  $\alpha\beta$  reference frame.[2]

$$\begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} d \\ q \end{bmatrix}, \quad (2.4)$$

where  $\theta$  is the electrical angle of the rotor.

**2.3.3 Torque control**

As mentioned above, the torque of PMSM is proportional to the outer product of the two magnetic flux linkage vectors. The PMSM torque equation:

$$T_e = T_m + T_r, \quad (2.5)$$

$$T_e = \frac{3}{2} \cdot \frac{P}{2} [\lambda_m i_q + (L_d - L_q) i_d i_q], \quad (2.6)$$

consists of 2 main elements [2]. The first is the electromagnetic torque  $T_m$ :

$$T_m = \frac{3}{2} \cdot \frac{P}{2} [\lambda_m i_q]. \quad (2.7)$$

This torque is produced by the magnetic flux linkage vector of the rotor  $\lambda_m$  and the quadrature axis current  $i_q$ . The second torque, represented by the equation:

$$T_r = \frac{3}{2} \cdot \frac{P}{2} [(L_d - L_q) i_d i_q], \quad (2.8)$$

is called reluctance torque  $T_r$ . This torque depends on the saliency of the rotor. The non-salient rotor (SPMSM) has magnetic asymmetry ( $L_d - L_q \approx 0$ ). On the other hand, the salient rotor motor (IPMSM) has ( $L_d - L_q \neq 0$ ). This results in a slightly different approach for torque control.

In the case of SPMSM, where torque is produced only by the  $i_q$  current, the  $i_d$  current is regulated to 0. Torque control of the IPMSM could be done in the same manner as SPMSM, but with loss of reluctance torque. Manipulation with  $i_d$  is needed to gain the advantage of the additive reluctance torque. One possible approach could be the Maximum torque per ampere strategy (MTPA). [2], [3], [5]

### 2.3.4 Mathematical model of PMSM

Several mathematical models in different reference frames (figure 2.4) can be derived for the different approaches of the PMSM sensorless vector control. For vector control, a mathematical model in rotor rotating reference frame  $d, q$  is used. The derivation is done in [3].

### 2.3.5 The mathematical model in $d, q$ rotating reference frame

Currents equations of the PMSM are

$$\frac{di_d(t)}{dt} = -\frac{R_s}{L_d} i_d + \omega_r \frac{L_q}{L_d} i_q + \frac{v_d}{L_d}, \quad (2.9)$$

$$\frac{di_q(t)}{dt} = -\frac{R_s}{L_q} i_q + \omega_r \frac{L_d}{L_q} i_d - \omega_r \frac{\lambda_m}{L_q} + \frac{v_q}{L_q}, \quad (2.10)$$

where  $i_d, i_q, R_s, L_d, L_q, v_d, v_q, \omega_r, \lambda_m$  represent current in d axis, current in q axis, stator winding resistance (one phase), d axis inductance, q axis inductance, d axis current, q axis current, rotor electrical speed and magnetic flux linkage of the permanent magnets. Rotor electric speed equation is

$$\frac{d\omega_r(t)}{dt} = \frac{P}{J}(T_e - T_L - B\omega_r - T_f), \quad (2.11)$$

where  $P, J, T_e, T_L, B, T_f$  represent number of pole pairs, rotor moment of inertia, electromagnetic torque, load torque, viscous damping coefficient and friction torque. The rotor electrical position  $\theta_r$  is obtained as the time derivation of the rotor electrical speed

$$\frac{d\theta_r(t)}{dt} = \omega_r. \quad (2.12)$$

The electromagnetic and friction torque of the PMSM are

$$T_e = \frac{3P}{2} i_q [\lambda_m + (L_d - L_q) i_d], \quad (2.13)$$

$$T_f = C \operatorname{sgn}(\omega_r), \quad (2.14)$$

where  $C$  represents Coulomb friction coefficient.

# Chapter 3

## Sensorless control

Knowing the rotor position and speed of PMSM is one of the most important requirements of vector control. Position and speed measurements can be realized by a position sensor (resolvers, encoders, tachometers) or a sensorless algorithm. Position sensors always add more hardware complexity to the system. This complexity can lead to reducing system robustness, e.g., for high-speed systems. The cost of the position sensors is also not negligible. The ratio between motor and encoder price depends on the chosen configuration. Resolvers, encoders and tachometers are the most expensive position and speed sensors. Nowadays, induction and magnetic Hall sensors are widely used, because its significant lower price (magnetic sensor can cost about 10 % price compare to the resolver).[6]–[8]

Replacing the position sensor of the electrical rotary machine with algorithm forms one of the most challenging approaches for motor control. Lot of research has been done with different approaches. Few of these strategies are shown in the figure 3.1.

Nowadays, commercially available PMSM drive systems implement some type of sensorless algorithm. No need for a position sensor reduces cost and maintenance. On the other side increases robustness and reliability.

Besides all advantages, a few assumptions need to be taken into concern. Sensorless algorithms can be strongly dependent on motor parameter variations and measurement noise. Few strategies are also not suitable for zero to low-speed applications. [9]–[12] The startup method needs to be implemented for strategies, which are not suitable for the low-speed applications. The startup method starts motor to the speed, where sensorless algorithm starts working properly. [13], [14]

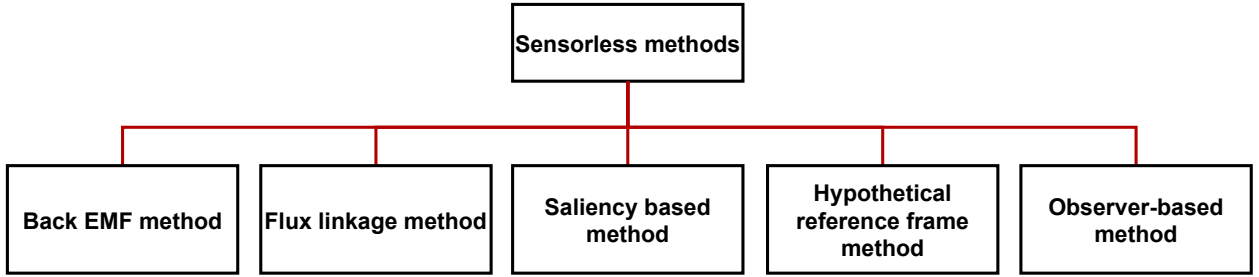


Figure 3.1: Sensorless estimation methods

### 3.1 Back EMF method

When the rotor rotates, it generates a voltage on the stator windings. This phenomenon is called back EMF. Using back EMF for the rotor position and speed estimation is one of the traditional methods. This method works very well with both salient and non-salient PMSM. Because this method depends on generated back EMF, it is not suitable for the zero and low-speed applications [2], [10], [12]. Generated back EMF at low speed may not be distinguishable from measurement noise. Also, back EMF at low speed might not be measurable due to the voltage drop of the stator winding resistance (depends on the current sensor resolution and accuracy).

As mentioned above, the instantaneous phase of back EMF strictly depends on the position between stator and rotor. Based on this, rotor position and speed can be determined. In the case of a permanent magnet DC motor (BLDC), back EMF can be measured at non driven phase by measuring terminal voltage. PMSM is a little bit different from back EMF measuring because all phases are driven. Back EMF is distinguished from current dependent voltages in the winding. Two of the many methods [10], [11] are shown in the figure 3.2. [2], [3]

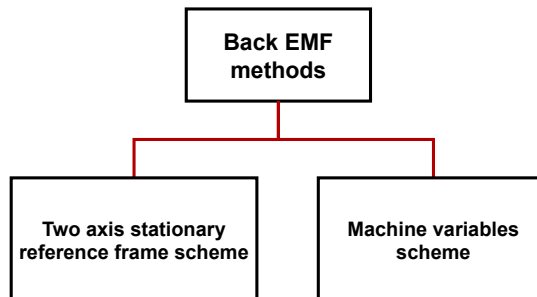


Figure 3.2: Back EMF method

### 3.1.1 Two-axis stationary reference frame

By making an assumption that the rotor magnetic flux linkage vector is oriented with the  $\alpha$ -axis, the magnetic flux in the  $\alpha - \beta$  stator stationary reference frame could be represented by an equation 3.1. [2]

$$\begin{bmatrix} \lambda_\alpha \\ \lambda_\beta \end{bmatrix} = \begin{bmatrix} L_{\alpha\alpha} & L_{\alpha\beta} \\ L_{\beta\alpha} & L_{\beta\beta} \end{bmatrix} \begin{bmatrix} i_\alpha \\ i_\beta \end{bmatrix} + \lambda_m \begin{bmatrix} \sin \theta_r \\ \cos \theta_r \end{bmatrix} \quad (3.1)$$

Mutual inductances represented as a function of the position are

$$L_{\alpha\alpha} = L_1 + L_2 \cos 2\theta_r \quad (3.2)$$

$$L_{\beta\beta} = L_1 - L_2 \cos 2\theta_r \quad (3.3)$$

$$L_1 = \frac{1}{2}(L_\alpha + L_\beta) \quad (3.4)$$

$$L_2 = \frac{1}{2}(L_\alpha - L_\beta) \quad (3.5)$$

$$L_{\alpha\beta} = L_{\beta\alpha} = \frac{1}{2}(L_\alpha - L_\beta) \sin 2\theta_r = -L_2 \sin 2\theta_r \quad (3.6)$$

By making substitution of above equations into stator voltage equation in  $\alpha - \beta$  reference frame 3.7, needed stator voltage equation is obtained as 3.8

$$\begin{bmatrix} v_\alpha \\ v_\beta \end{bmatrix} = \begin{bmatrix} R_\alpha & 0 \\ 0 & R_\beta \end{bmatrix} \begin{bmatrix} i_\alpha \\ i_\beta \end{bmatrix} + \frac{d}{dt} \begin{bmatrix} \lambda_\alpha \\ \lambda_\beta \end{bmatrix} \quad (3.7)$$

$$\begin{bmatrix} v_\alpha \\ v_\beta \end{bmatrix} = R_s \begin{bmatrix} i_\alpha \\ i_\beta \end{bmatrix} + \begin{bmatrix} L_1 & 0 \\ 0 & L_1 \end{bmatrix} \frac{d}{dt} \begin{bmatrix} i_\alpha \\ i_\beta \end{bmatrix} + \lambda_m \omega_r \begin{bmatrix} \cos \theta_r \\ -\sin \theta_r \end{bmatrix} \quad (3.8)$$

The last element of the stator voltage equation 3.8 depends on the magnetic flux of rotor permanent magnets  $\lambda_m$ , rotor speed  $\omega_r$  and rotor position  $\theta_r$ . Equations 3.9 and 3.10 represent back EMF in  $\alpha - \beta$  reference frame.

$$e_\alpha = \omega_r \lambda_m \cos \theta_r \quad (3.9)$$

$$e_\beta = -\omega_r \lambda_m \sin \theta_r \quad (3.10)$$



By knowing  $\alpha$ - and  $\beta$ -axis voltages and currents, back EMF can be calculated online.

$$\begin{bmatrix} e_\alpha \\ e_\beta \end{bmatrix} = \begin{bmatrix} v_\alpha \\ v_\beta \end{bmatrix} - R_s \begin{bmatrix} i_\alpha \\ i_\beta \end{bmatrix} - \begin{bmatrix} L_1 & 0 \\ 0 & L_1 \end{bmatrix} \frac{d}{dt} \begin{bmatrix} i_\alpha \\ i_\beta \end{bmatrix} \quad (3.11)$$

Position can be then determined as:

$$\theta_r = \tan^{-1} \left( \frac{-e_\beta}{e_\alpha} \right) \quad (3.12)$$

Two-axis stationary reference frame method works for SPMSM and IPMSM. This method could be hardly influenced by the motor parameters variation and measurement noise.[2], [3]

### 3.1.2 Machine variables scheme

This method estimates position directly from the motor parameters and measures current and voltages. This method uses no transformation to the two-axis reference frame.[2]

$$\theta_r = \tan^{-1} \frac{v_b - v_c - R_s(i_b - i_c) - L_d \frac{d(i_b - i_c)}{dt} - \sqrt{3}\omega_r(L_d - L_q)i_a}{\sqrt{3}(v_a - R_s i_a - L_d \frac{di_a}{dt}) + \omega_r(L_d - L_q)(i_b - i_c)} \quad (3.13)$$

The equation of the rotor position angle 3.13 can be derived offline from equation 3.12. The purpose of this method is to reduce online calculations. However, its dependence on motor parameters variation and measurement noise still remains. [2]

## 3.2 Flux linkage method

Knowing the angular position of the stator flux linkage vector replaces the rotor position. Vector control can be then used in stator flux reference frames and behaves like a sensorless.

Four methods of flux linkage angle estimation are presented in the figure 3.3.[2], [11]

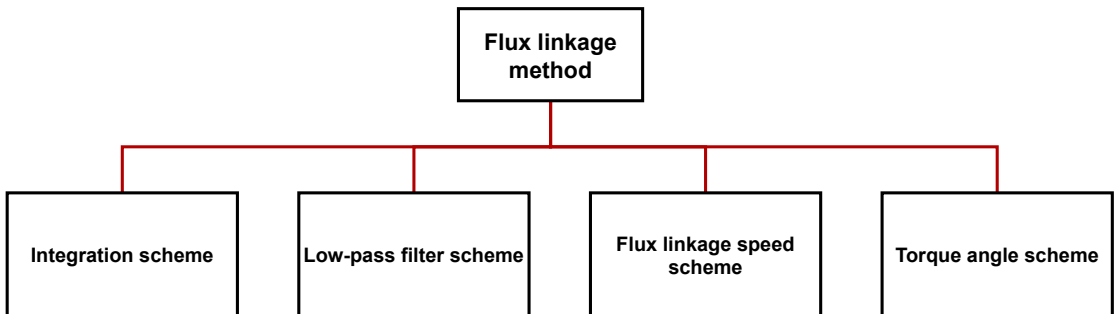


Figure 3.3: Flux linkage method

### 3.2.1 Integration scheme

The principle of the integration scheme is illustrated in the figure 3.4. Currents and voltages are measured and transformed to the  $d, q$  reference frame. Flux linkage components are then calculated as illustrated.

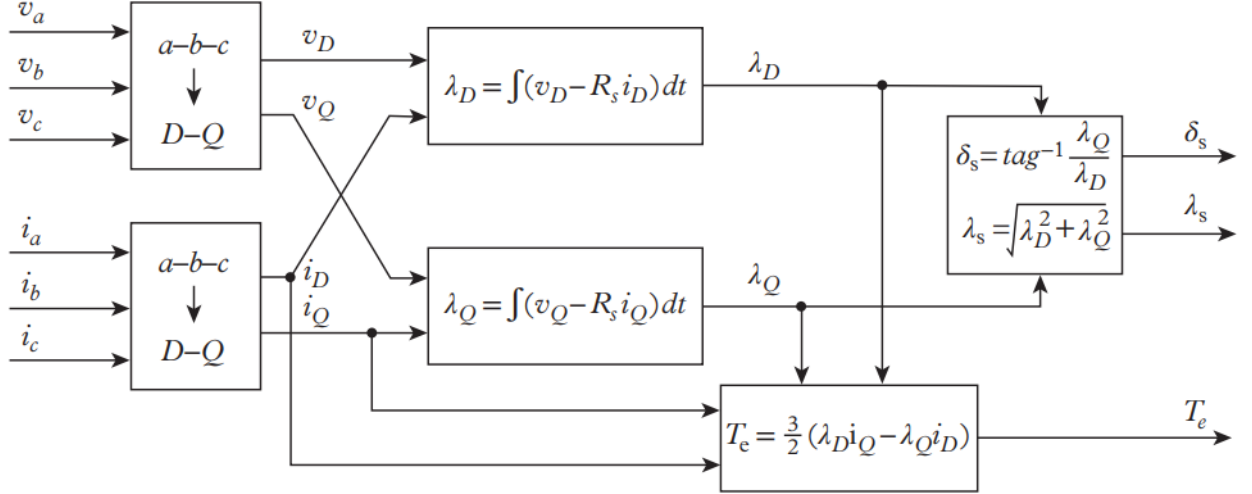


Figure 3.4: Flux angle synthesis [2]

In addition to determining the stator flux linkage angle, it is possible to obtain flux magnitude. Knowledge of flux linkage magnitude is necessary for flux control methods. As mentioned above, the method depends on measurements. Measurements could be affected by noise, DC offset, voltage drop and non-linearities. Also, dependency on the stator winding resistance is non-negligible. Proper resistance measuring and temperature compensation must be done. The integral scheme works, if all conditions are perfect, down to 1-2 Hz. [2]

### 3.2.2 Low-pass filter scheme

Method, represented by the scheme in the figure 3.5, estimates stator flux linkage components. Using low-pass filters instead of integrators removes the effect of initial conditions.

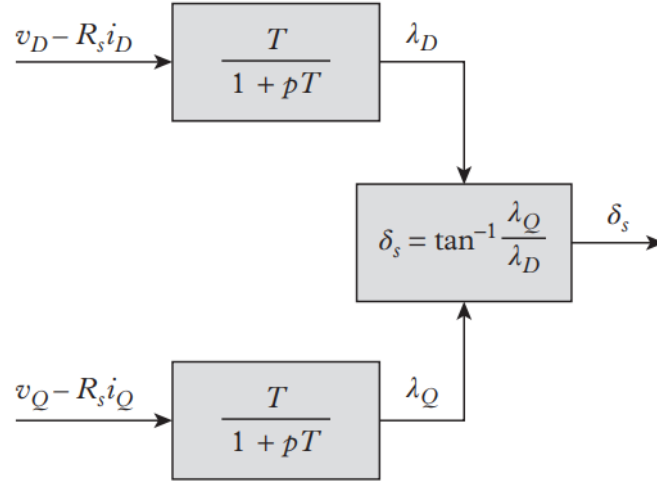


Figure 3.5: Low-pass filters stator flux linkage estimation [2]

The scheme consists of single or cascade-stage low-pass filters. Tuning of the filter time constant  $T$  affects flux linkage components estimation. A large time constant provides accurate estimation at low frequencies. On the other hand, it increases filter damping. The proper time constant  $T$  must be selected to fulfil operational requirements. One possible solution is to make time constant adaptive. [2]

### 3.2.3 Flux linkage speed estimation scheme

The flux linkage speed estimation scheme uses flux linkage components calculated by integration same as in the integral scheme. Speed of the flux linkage vector can be obtained by derivation of the flux linkage vector  $\delta_s$

$$\delta_s = \theta_r + \delta \quad (3.14)$$

,

where  $\theta_r$  is the rotor position and  $\delta$  is the stator flux linkage position of the load angle. Derivation of equation 3.14 and substitution of equation 3.15 result in equation 3.16.

$$\delta_s = \tan^{-1} \frac{\lambda_\alpha}{\lambda_\beta} \quad (3.15)$$

$$\omega_s = \frac{d\delta_s}{dt} = \frac{\lambda_\alpha(v_\beta - R_s i_\beta) - \lambda_\beta(v_\alpha - R_s i_\alpha)}{\lambda_\alpha^2 + \lambda_\beta^2} \quad (3.16)$$

Method struggles at the low-speed region due to the inaccuracy of the flux linkage calculations. [2], [3], [10]

### 3.2.4 Torque angle estimation scheme

This method uses a recursive approach to estimate rotor position angle  $\theta_r$ . Based on the equation 3.14, rotor angle can be presented as

$$\theta_r = \delta_s - \delta \quad (3.17)$$

By having an estimation of the flux linkage angle  $\delta_s$ , the rotor angle could be calculated with the estimate of the torque angle  $\delta$ .

$$\delta = \tan^{-1} \frac{\lambda_q}{\lambda_d} = \frac{L_q i_q}{\lambda_m + L_d i_d} \quad (3.18)$$

This method provides fast and decent results. Drawbacks are the same with all flux methods mentioned above. [2]

## 3.3 Saliency-based method

As the title propose, these methods are using motor saliency to extract position and speed. Several methods are presented by the scheme in the figure 3.6. Methods based on signal injection are nowadays mostly used in this category.

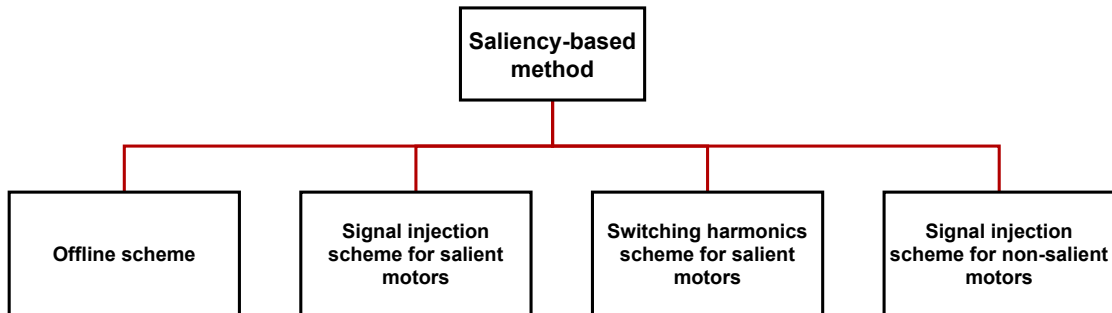


Figure 3.6: Saliency-based method

Signal injection methods are based on injecting a high-frequency carrier signal (voltage or current) on top of the fundamental excitation of the machine. This results in inducted signals containing information. This information is processed with signal processing methods to obtain rotor speed and position. Method hardly depends on processing methods (filters, tracking observers etc.). The main advantage of using this approach is zero-speed position detection. [2], [3], [5]

### 3.4 Hypothetical reference frame method

The principles of these methods are based on the surface-mounted PMSM model in a hypothetical reference frame (figure 3.8). This frame is oriented along with an estimated rotor position. The result of error between estimated and unknown position formulates the voltage component. By measuring motor variables, this voltage component is estimated by a machine model. To reduce position error, this method uses a PI controller. Few approaches to this method are shown in the figure 3.7. [2], [3]

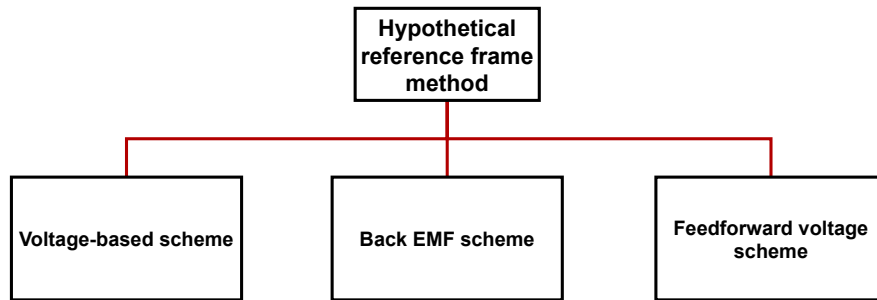


Figure 3.7: Hypothetical reference frame method

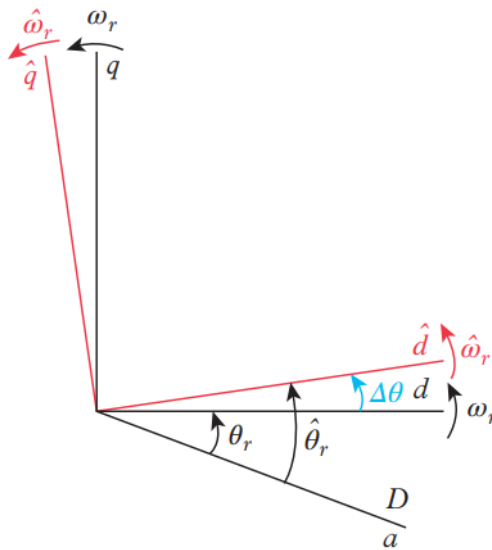


Figure 3.8: A hypothetical rotor reference frame in the counterclockwise direction of rotation [2]

### 3.5 Observer-based method

The use of observer-based methods for position, speed and parameter estimation has been heavily researched and developed for more than 20 years. Three of many methods are presented in figure 3.9. These methods correct their prediction by taking the error between measured and estimated output into the consideration. Observers are based on the dynamic system model in state-space form, both linear and non-linear, and tunable parameters. Tuning of the observer parameters is often done experimentally for the specific case of use. Properly tuned parameters ensure the convergence and the stability [15] of the observer.

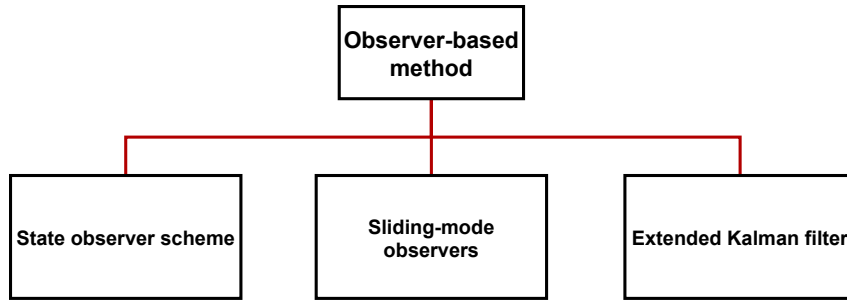


Figure 3.9: Observer-based method

#### 3.5.1 State observer scheme

To describe the form of the state observer, a system dynamic model is needed. Because the PMSM dynamic model is non-linear, derivation of the scheme will be done from it. The general non-linear system model in the state-space form is described by equations

$$\frac{d\mathbf{x}(t)}{dt} = \mathbf{f}[\mathbf{x}(t), \mathbf{u}(t)], \quad (3.19)$$

$$\mathbf{y}(t) = \mathbf{h}[\mathbf{x}(t)], \quad (3.20)$$

where  $\mathbf{x}(t)$  and  $\mathbf{u}(t)$  are current system state and input vectors. For later error calculation, the measured output vector  $\mathbf{y}(t)$  is needed. To adjust the influence of the error between measured and estimated output, observer gain matrix  $\mathbf{G}$  is introduced. The general state observer is presented by the following equation,

$$\frac{d\hat{\mathbf{x}}(t)}{dt} = \mathbf{f}[\hat{\mathbf{x}}(t), \mathbf{u}(t)] + \mathbf{G}[\mathbf{y}(t) - \hat{\mathbf{y}}(t)], \quad (3.21)$$

where  $\hat{\mathbf{x}}(t)$  and  $\hat{\mathbf{y}}(t)$  represent the estimates. The goal of the estimator is to achieve equality between measurement and estimation. Component  $\mathbf{G}[\mathbf{y}(t) - \hat{\mathbf{y}}(t)]$  of the equation

3.21 is by the principle proportional regulator. Elements of the gain matrix  $\mathbf{G}$  are chosen based on the PMSM dynamics. Constant values might not provide good estimates through the whole speed range. [2], [3], [5]

$$\frac{d}{dt} \begin{bmatrix} \hat{i}_d \\ \hat{i}_q \end{bmatrix} = \begin{bmatrix} -\frac{R_s}{L_d} & \omega_r \frac{L_q}{L_d} \\ \omega_r \frac{L_d}{L_q} & -\frac{R_s}{L_q} \end{bmatrix} \begin{bmatrix} \hat{i}_d \\ \hat{i}_q \end{bmatrix} - \hat{\omega}_r \lambda_m \begin{bmatrix} 0 \\ 1 \end{bmatrix} + \begin{bmatrix} \frac{1}{L_d} & 0 \\ 0 & \frac{1}{L_q} \end{bmatrix} \begin{bmatrix} v_d \\ v_q \end{bmatrix} + \begin{bmatrix} g & e \\ e & g \end{bmatrix} \left( \begin{bmatrix} i_d \\ i_q \end{bmatrix} - \begin{bmatrix} \hat{i}_d \\ \hat{i}_q \end{bmatrix} \right) \quad (3.22)$$

$$\begin{aligned} \frac{d\hat{\omega}_r}{dt} = & \begin{bmatrix} \hat{i}_d & \hat{i}_q \end{bmatrix} \frac{3P^2 \lambda_m}{2 J} \begin{bmatrix} 0 \\ 1 \end{bmatrix} + \frac{3P^2 L_d - L_q}{2 J} \begin{bmatrix} \hat{i}_d & \hat{i}_q \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \hat{i}_d \\ \hat{i}_q \end{bmatrix} - \frac{PB\omega_r}{J} - \frac{PC}{J} \text{sgn}(\omega_r) - \\ & - \frac{PT_L}{J} + \frac{3P \lambda_m}{2 J} \begin{bmatrix} m & n \end{bmatrix} \left( \begin{bmatrix} i_d \\ i_q \end{bmatrix} - \begin{bmatrix} \hat{i}_d \\ \hat{i}_q \end{bmatrix} \right) \end{aligned} \quad (3.23)$$

$$\frac{d\hat{\theta}_r}{dt} = \hat{\omega}_r \quad (3.24)$$

### 3.5.2 Sliding-mode observer

Sliding-mode observers (SMO) are based on the Lyapunov function. SMO are often used with the estimation of the back EMF. Feedback of this algorithm is realized with the signum function.

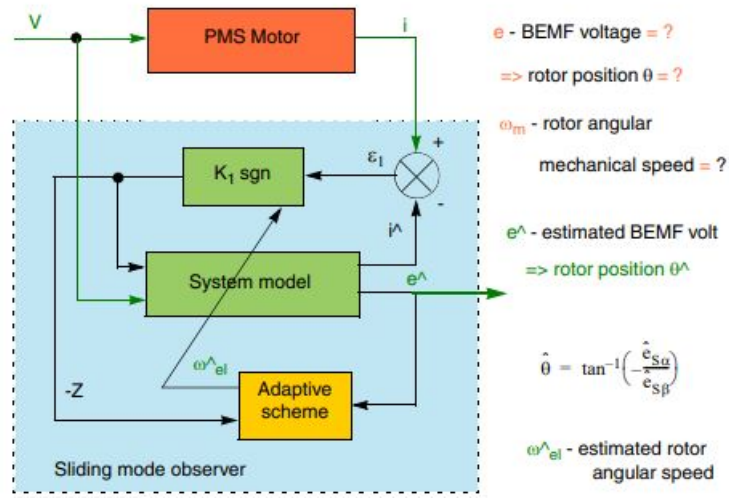


Figure 3.10: Sliding-mode observer [16]

### 3.5.3 Extended Kalman filter

A Kalman filter works as an optimal state and parameter estimator for linear systems. Because of non-linearities in the PMSM model, the algorithm needs to be modified. This upgraded algorithm is called "Extended Kalman filter (EKF)". Kalman filter works by minimizing the mean square error of estimates. Input noise, model inaccuracies and disturbances are also taken into account. EKF works for the non-linear system by linearizing model equations and measurements about the predicted state. [2], [3], [5]

The whole estimation procedure has precondition, that the system model dynamics is known.

The non-linear system model is presented by the following equations,

$$\frac{d\mathbf{x}(t)}{dt} = \mathbf{f}[\mathbf{x}(t)] + \mathbf{B}\mathbf{u}(t) + \boldsymbol{\sigma}(t), \quad (3.25)$$

$$\mathbf{y}(t_k) = \mathbf{H}\mathbf{x}(t_k) + \boldsymbol{\mu}(t_k), \quad (3.26)$$

where  $\mathbf{x}(t)$  is the system state vector with the initial state vector  $\mathbf{x}(t_0)$  described as a Gaussian random vector with mean  $\mathbf{x}_0$  and covariance matrix  $\mathbf{P}_0$ , and  $\mathbf{u}(t)$  is deterministic input vector. Optimal state estimate  $\mathbf{x}(t_k)$  with its covariance  $\mathbf{P}(t_k)$  and measurement output vector  $\mathbf{y}(t_k)$  are defined in the discrete-time domain. Terms  $\boldsymbol{\sigma}(t)$  and  $\boldsymbol{\mu}(t_k)$  mean a zero-mean white Gaussian noise. This white noise needs to be independent. First Gaussian white noise  $\boldsymbol{\sigma}(t)$  is independent of  $\mathbf{x}(t_0)$  and has covariance  $\mathbf{Q}(t)$ . Second Gaussian white noise needs to be independent of  $\mathbf{x}(t_0)$  and  $\mathbf{u}(t)$  with covariance  $\mathbf{R}(t_k)$ . Input and output matrix of the non-linear system are denoted as  $\mathbf{B}$  and  $\mathbf{H}$ . Symbol  $k$  denotes discrete-time step. [2], [5]

The Extended Kalman filter algorithm could be described as a two-step feedback algorithm.

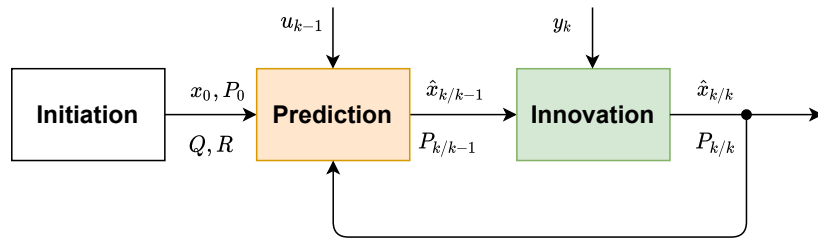


Figure 3.11: EKF algorithm flow chart

Figure 3.11 describes EKF algorithm flow. In the beginning, the algorithm is initialized. Initialisation values are chosen often by the try and error approach. Then prediction step and innovation step are executed.



### 1. Prediction step

In this step, optimal estimated state vector  $\hat{\mathbf{x}}_{k/k-1}$  and state covariance matrix  $\mathbf{P}_{k/k-1}$  are calculated. Calculations are done online from the system mathematical model. To compute estimates at the time  $k$ , most recent system states and inputs from time  $k-1$  are used. Equation 3.27 is obtained by the triangular integration of the equation 3.25 from  $k-1$  to  $k$ . System state white Gaussian noise  $\boldsymbol{\sigma}(0)$  is considered to be 0. Covariance matrix  $\mathbf{Q}$  is set to the constant. [2], [5] Process of prediction step is then set as,

$$\hat{\mathbf{x}}_{k/k-1} = \hat{\mathbf{x}}_{k-1/k-1} + [\mathbf{f}(\hat{\mathbf{x}}_{k-1/k-1}) + \mathbf{u}_{k-1}]T_s, \quad (3.27)$$

$$\mathbf{P}_{k/k-1} = \mathbf{P}_{k-1/k-1} + (\mathbf{F}_{k-1}\mathbf{P}_{k-1/k-1} + \mathbf{P}_{k-1/k-1}\mathbf{F}_{k-1}^T + \mathbf{Q})T_s, \quad (3.28)$$

where

$$\mathbf{F}_{k-1} = \left. \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \right|_{\mathbf{x}=\hat{\mathbf{x}}_{k-1/k-1}} \quad (3.29)$$

### 2. Innovation step

After prediction to state  $k$  from  $k-1$  is done by the prediction step, innovation comes into play. Kalman gain  $\mathbf{K}_k$  is calculated from these estimates.

$$\mathbf{K}_k = \mathbf{P}_{k/k-1}\mathbf{H}^T(\mathbf{H}\mathbf{P}_{k/k-1}\mathbf{H}^T + \mathbf{R})^{-1} \quad (3.30)$$

Estimates at time  $k$  are then innovated with use of Kalman gain as,

$$\hat{\mathbf{x}}_{k/k} = \hat{\mathbf{x}}_{k/k-1} + \mathbf{K}_k(\mathbf{y}_k - \mathbf{H}\hat{\mathbf{x}}_{k/k-1}), \quad (3.31)$$

where  $\mathbf{y}_k$  represents measurements vector at current time  $k$ . Result of that is corrected estimation of state vector for current time. Covariance matrix  $\mathbf{P}$  is also innovated as follows.

$$\mathbf{P}_{k/k} = \mathbf{P}_{k/k-1} - \mathbf{K}_k\mathbf{H}\mathbf{P}_{k/k-1} \quad (3.32)$$

The state vector estimate  $\hat{\mathbf{x}}_{k/k}$  and covariance matrix  $\mathbf{P}_{k/k}$  are then forced as feedback to the prediction step in next the algorithm execution. [2], [5]

### Optimal state-space model for EKF

Because the EKF algorithm is based on matrix calculation, the model needs to be represented in the same manner. This model representation, state-space, needs to be easy to linearize. Stationary two-axis reference frame  $\alpha\beta$  is more suitable for this application.  $d, q$  rotation reference frame includes transformation from stationary to the rotary reference frame. This transformation needs to be observed and that can introduce errors into the estimation process. This is the reason, why stationary reference frame  $\alpha\beta$  is preferred and chosen for the estimator. Implementation of EKF algorithm into motor controller system is shown in the figure 3.12 in  $\alpha\beta$  stationary reference frame. [2]

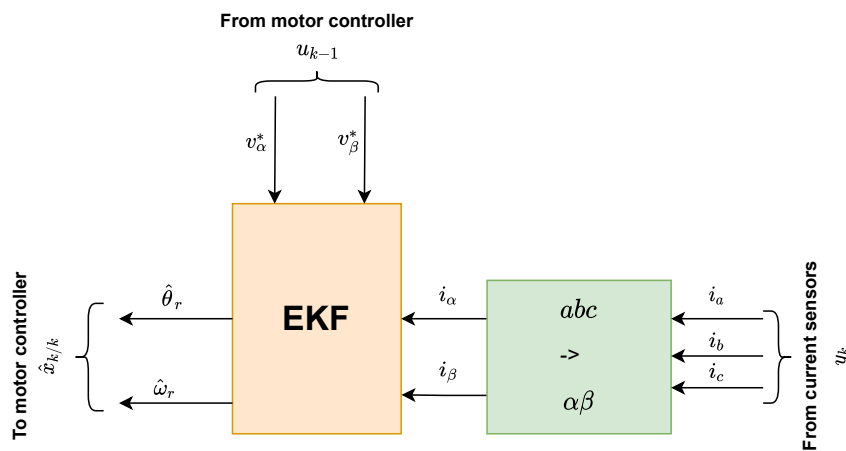


Figure 3.12: EKF algorithm as part of motor controller

The choice of  $\alpha\beta$  stationary reference frame suits very well SPMSMs. Inductances  $L_d$  and  $L_q$  are same. In the case of IPMSMs, the situation is different.  $d, q$  axis inductances are no longer equal. Implementation of the EKF algorithm for the IPMSM is often done in the  $d, q$  reference frame [2]. Design of the used EKF will be presented in the chapter 4.3.

# Chapter 4

## Simulation

### 4.1 Simulation overview

At the beginning of developing the control strategy for the PMSMs, simulation of the whole system comes into play. Simulations provide necessary knowledge and data for further development. The scheme of the complete sensorless vector controller for the PMSM is shown in the figure 4.1. The controller consists of three main blocks. The first one, called "Speed Control" takes care of the speed regulation. This block provides needed current references for the Current Control block. Current Control block implements current regulators, motor startup algorithm and pulse width modulation algorithm. The block responsible for the accurate position and speed estimation based on the extended Kalman filter using back EMF estimation for the position and speed estimation is located below them.

The simulation was done in Matlab Simulink with the use of a few block sets. The first used block set, the Motor control block set, adds a lot of features for the PMSM control. Simulink blocks from it are optimized for later code generation. This was the main reason to implement them. The second block set was used mainly for the DSP implementation. It is called the Embedded software package for the TI C2000 processors. The main use of this tool stays in designing system architecture with all necessary peripherals. The description of the simulation structure of the SPMSM sensorless vector controller (figure 2.3) begins with a description of the current (torque) inner loop, follows by the outer speed loop, the PMSM model and the inverter and ends with a description of the EKF .

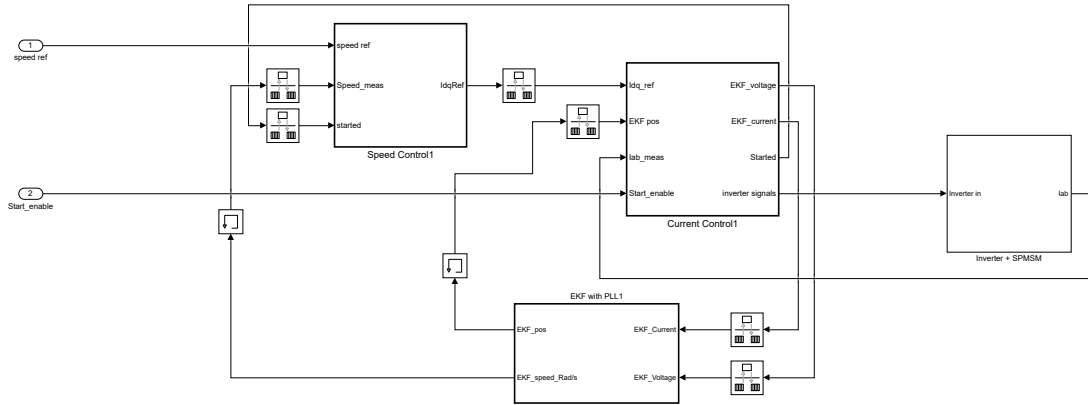


Figure 4.1: PMSM controller

## 4.2 Vector control

### 4.2.1 Current regulation

Block named Current Control implements both currents regulation and startup algorithm, as mentioned above. Current regulation is done in  $d, q$  reference frame. The reason for the  $d, q$  reference frame control is the DC character, instead of the sine wave. Based on that, Clarke and Park's transformations were implemented. Figure 4.2 represents the scheme of Current Control block.

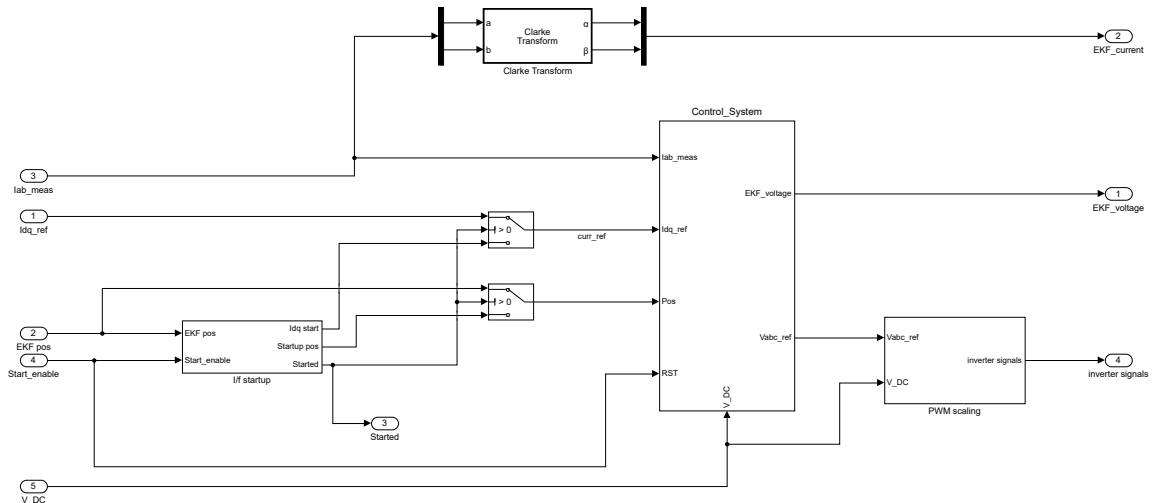


Figure 4.2: Current control with startup

Subsystem Control System handles main current control in  $d, q$  reference frame. Individual necessary transformations are evident from the figure 4.3. Measured currents from PMSM in  $abc$  3-phase reference frame are transformed by the Clarke transformation to the  $\alpha - \beta$  2-phase reference frame. Then Park transformation takes care of the

transformation to  $d, q$  reference frame, needed for the regulators. These  $d, q$  currents work as feedback for regulation. Reference  $d, q$  currents are provided from the Speed Control based on speed command or Startup algorithm.

Output from the Current Controllers, a voltage in  $d, q$  reference frame, is transformed with the use of inverse Park transformation back to the  $\alpha - \beta$  reference frame. There are several ways to proceed. PWM algorithm could be designed for the  $\alpha - \beta$  input as well as for abc input. This control system implements the second approach using the abc voltages input.

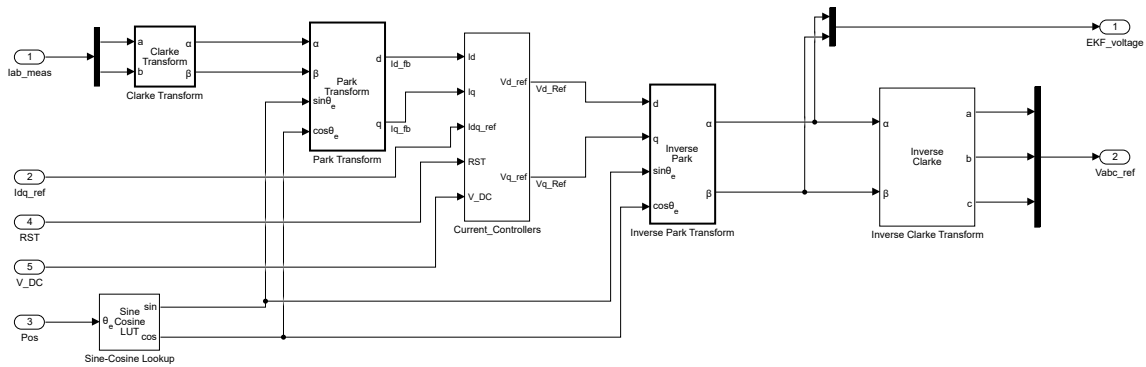


Figure 4.3: Current control system

A detailed scheme of the  $d, q$  current controllers are represented in the figure 4.4. Regulators are the type of PI. Whole blocks, visible in the figure 4.5, were implemented from the Motor control block set. They are discrete type PI controllers with the anti-windup and reset. Reset is mainly used to eliminate current ripple when switching from startup algorithm to vector control.

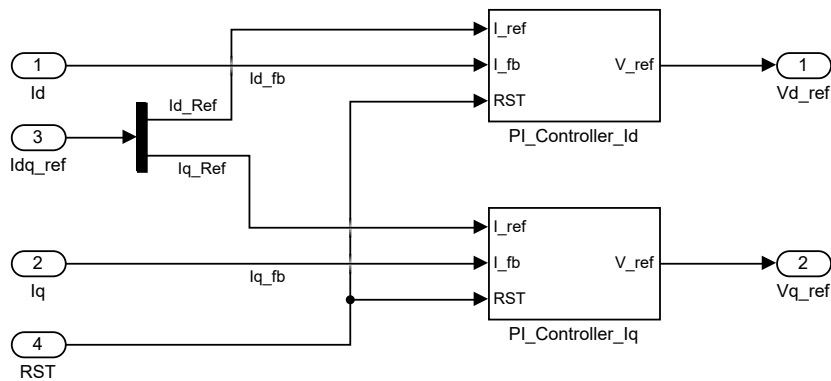
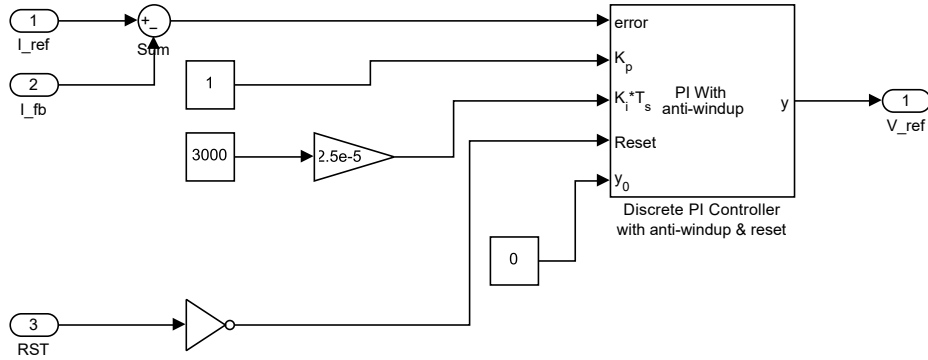


Figure 4.4: Current controllers

Figure 4.5: PI regulator scheme for  $i_d$  current

## 4.2.2 I/f startup algorithm for the PMSM

Because the extended Kalman filter algorithm estimates back EMF, the rotor needs to spin at a specific rate. The magnitude of the back EMF is proportional to the rotor speed. The value of this startup speed depends on the whole system configuration (motor construction, current sensor sensitivity etc.). Often used value is 10% of rated motor speed [13] (in case of the regular speed PMSMs). This statement is only indicative and must be adapted to the specific application. For this application, 10% of rated motor speed works very well.

I/f startup method, e.g., constant current to frequency ratio, works as an open-loop control. The main difference in constant voltage to frequency ratio (V/f) stays in the process of implementation. U/f startup algorithm doesn't use implemented PI regulators of the vector controller and doesn't depend on the rotating reference frame of the vector control. Due to that, switching from U/f startup to vector control needs special procedures to align startup and rotor rotating reference frames. If frames are not aligned properly, switching to vector control could halt the motor.

On the other hand, the I/f starting method uses PI regulators of the vector controller. This results in better switching to the vector control and better startup under load, because torque is controlled. Because of that, the I/f startup method is preferred in the field of PMSM startup algorithms. Implementation of this startup method [13] to the controller is visible in the figure 4.2. The whole sequence of the I/f starting algorithm is described by the state diagram in figure 4.7.

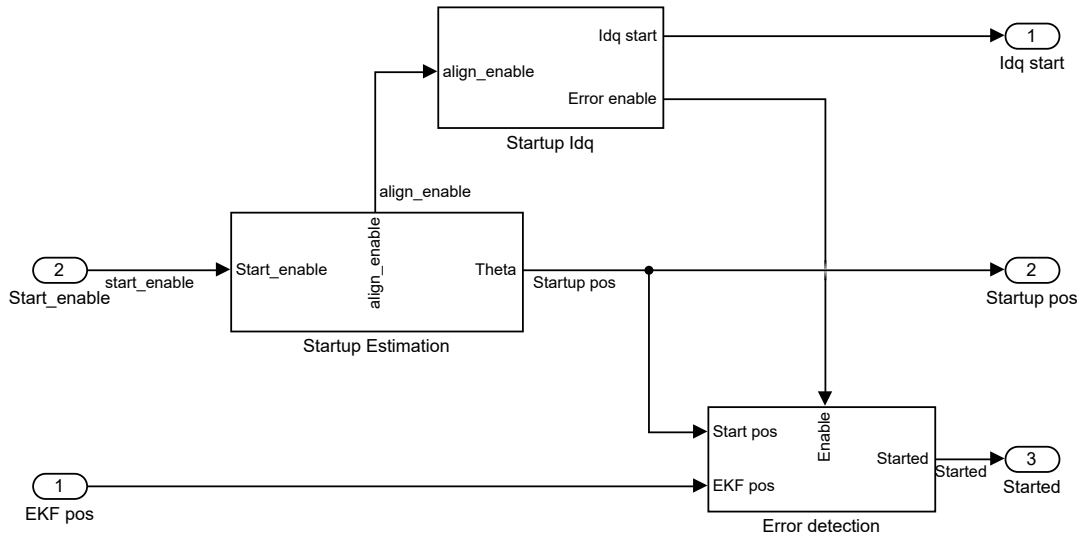


Figure 4.6: Startup algorithm scheme

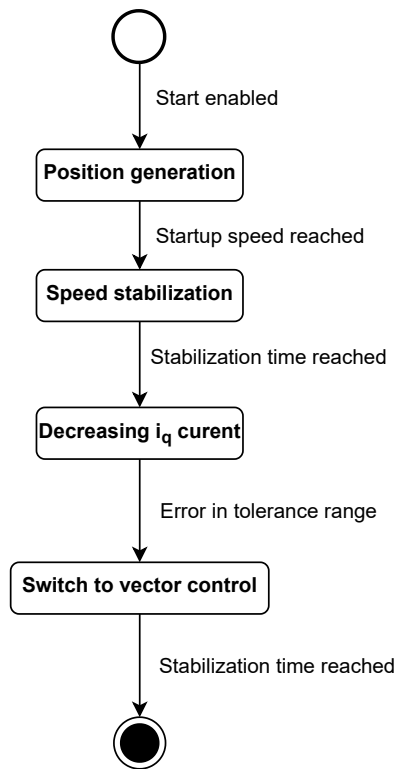


Figure 4.7: Startup algorithm state diagram

The startup algorithm provides reference  $d, q$  currents and rotor angle to the current controllers until the motor reaches desired conditions. These references are generated, so the startup method forms its own rotating reference frame. Individual blocks of this method are represented by the figure 4.6.

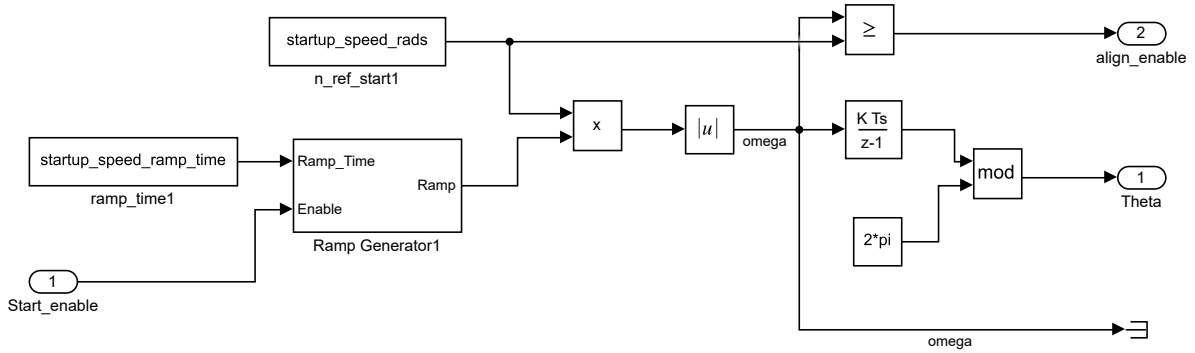


Figure 4.8: Startup position generator

Figure 4.8 shows the principle of the rotor position angle generation. The ramp generator generates a rotor speed ramp which is integrated. The result of that procedure is the required rotor position in the startup rotating reference frame. This position is then fed into the Park transformation inside the current control block.

Reference currents are generated in a similar manner by the ramp generator. Parameters for individual currents need to be set based on the motor parameters. It is important to use the lowest possible current that provides a reliable start of the motor. This results in higher efficiency and less Joule thermal losses. Selected  $i_q = const$  current needs to be high enough so that the startup rotating reference frame is lagging behind real, i.e., estimator rotating reference frame. This means that produced torque is accelerating the machine faster than the startup rotating reference frame. This behaviour self-stabilises the whole system when starting.

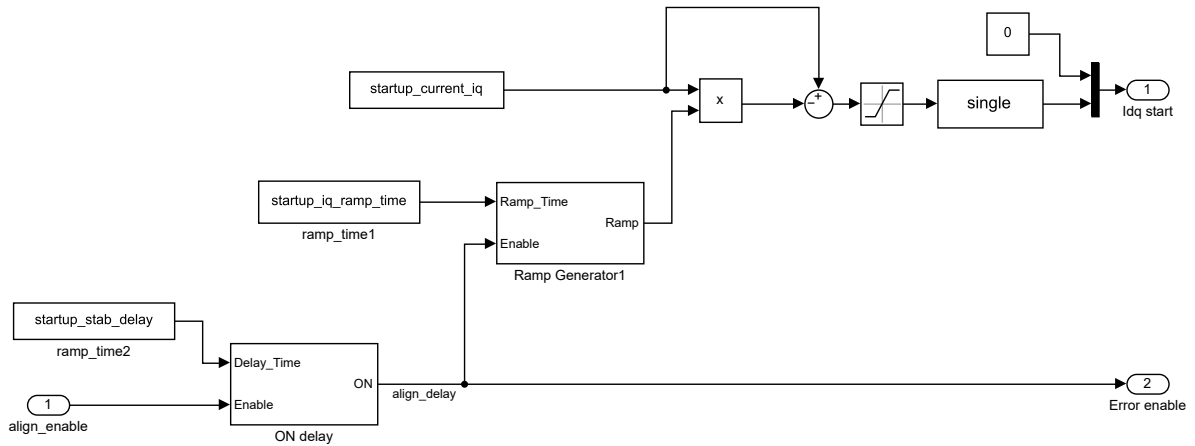


Figure 4.9: Startup current generator



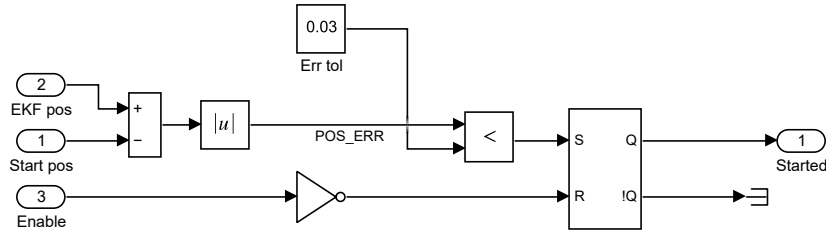


Figure 4.10: Startup error

To achieve a smooth transition from startup to vector control, estimator and startup rotating reference frames need to be aligned properly. This is done by decreasing  $i_q$  current. The real rotating frame starts to decelerate. By measuring the error between both reference frames, an ideal switching point could be found. Implementation of the error detection describes figure 4.10.

### 4.2.3 Speed regulation

The speed controller takes care of providing required  $d, q$  currents based on the speed command. Also implements a few features for a proper startup. The first one is the speed command switch, which depends on the starting procedure. When the startup algorithm switches to the vector control, the same speed reference is held. This is done by the ON delay. This is required for system stabilization. After that vector control can work in user-defined conditions.

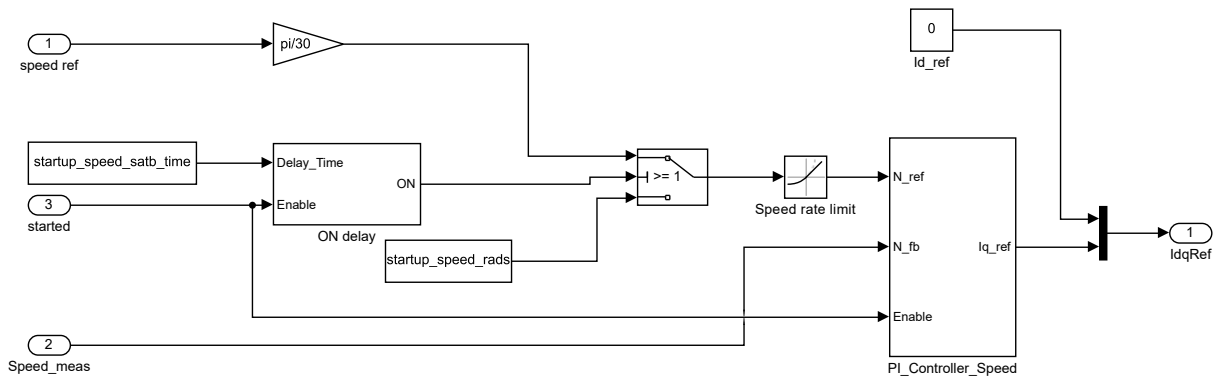


Figure 4.11: Speed controller

A speed control system, represented by the figure 4.11, implements a discrete PI regulator with anti-windup. Also, the reset feature is implemented because of the smooth transition on startup.

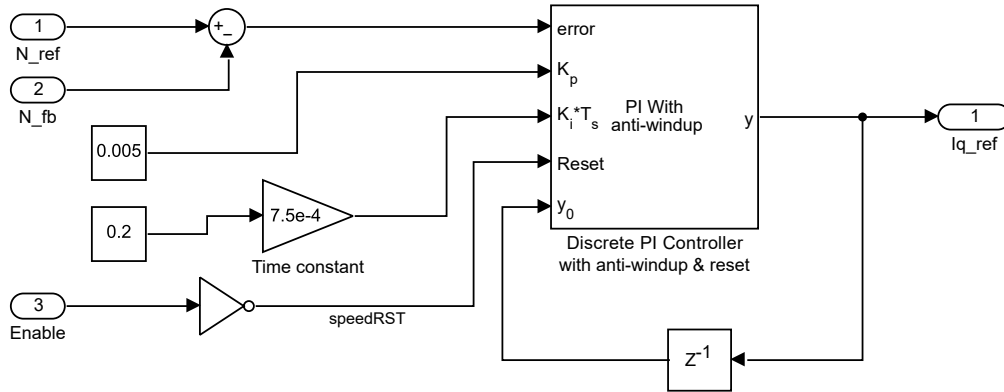


Figure 4.12: Speed PI regulator

Output from the PI regulator is used as reference  $i_q$  current for the current controller. This developed strategy was meant for operation in the base speed region for SPMSM. This is reason why reference current  $i_d = 0$ . This is a typical approach for this rotor topology when there is no need to operate above base speed.

#### 4.2.4 Pulse width modulation

PMSMs are powered with 3-phase AC voltage, but the inverter is powered by the DC power supply. Conversion from DC voltage to 3-phase voltage is done by pulse width modulation (PWM). Method of PWM modulation DC voltage to AC voltage used in this control strategy is called "Sinusoidal pulse width modulation (SPWM)". This method is reliable and easy to implement in its digital form. Simulink implementation of the SPWM method for one phase is shown in the figure 4.13. The output voltage of PI regulators is transformed to abc reference frame. The first step, normalization, converts phase voltage of exact value into normalized one. The base voltage value for normalization is obtained from the DC voltage of the inverter power supply.

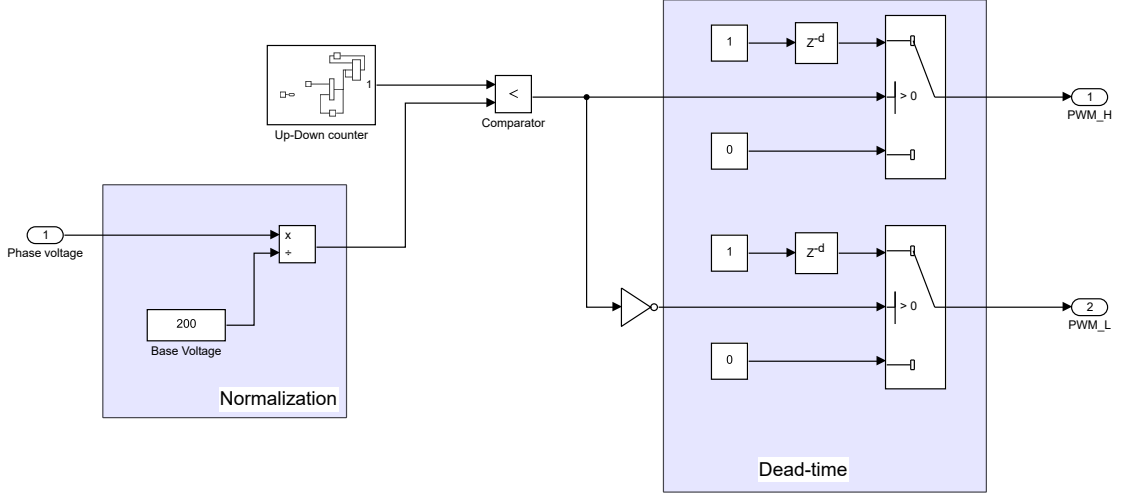


Figure 4.13: SPWM algorithm implementation

Normalized voltage is then compared with the carrier signal. The Carrier signal used here is a triangular wave. To achieve a triangular wave signal, the up-down counter is used for this purpose. The key factor to consider is the frequency of the carrier triangular wave signal. Frequency carrier signal needs to be high enough for the proper sine modulation of the sine voltage signal. Frequency of the carrier signal is usually chosen as high as possible, with respect to the system. Ratio between the carrier and voltage frequencies is called frequency modulation index  $m_f$ , [17]

$$m_f = \frac{f_c}{f_v}, \quad (4.1)$$

where  $f_c$  is the carrier signal frequency and  $f_v$  is the voltage signal frequency.

The maximum frequency of the AC current provided to PMSM could be obtained from equation 4.2 with respect to maximum RPMs. For example. If one pole pair ( $P = 1$ ) PMSM is rated for  $n_{base} = 100,000$  RPM, current frequency is approx. 1.7 kHz. By choosing frequency modulation index  $m_f = 12$ , carrier signal frequency is approximately 20 kHz.

$$f_{current} = \frac{n_{base}}{120} 2P \quad (4.2)$$

Choosing higher carrier frequency results in an increase in the PWM switching frequency and better sine wave modulation. High PWM switching induces less current ripple, which is wanted. The drawbacks of increasing PWM switching frequency are higher transistors switching losses and the computational power of the hardware that runs this controller.

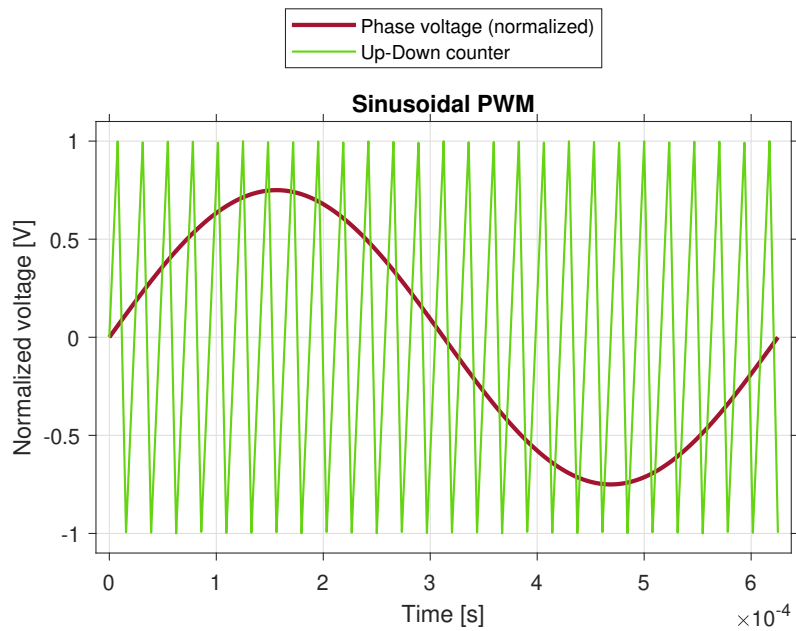


Figure 4.14: SPWM comparison

Figure 4.14 suggests how does comparison between carrier and voltage signal looks like. Implemented comparator checks, when carrier signal is higher or lower in the magnitude. Output of that comparison is then PWM signal for the high-side switching transistor of the inverter. Low-side transistor signal is generated by negation of high-side signal, as seen from above schematic.

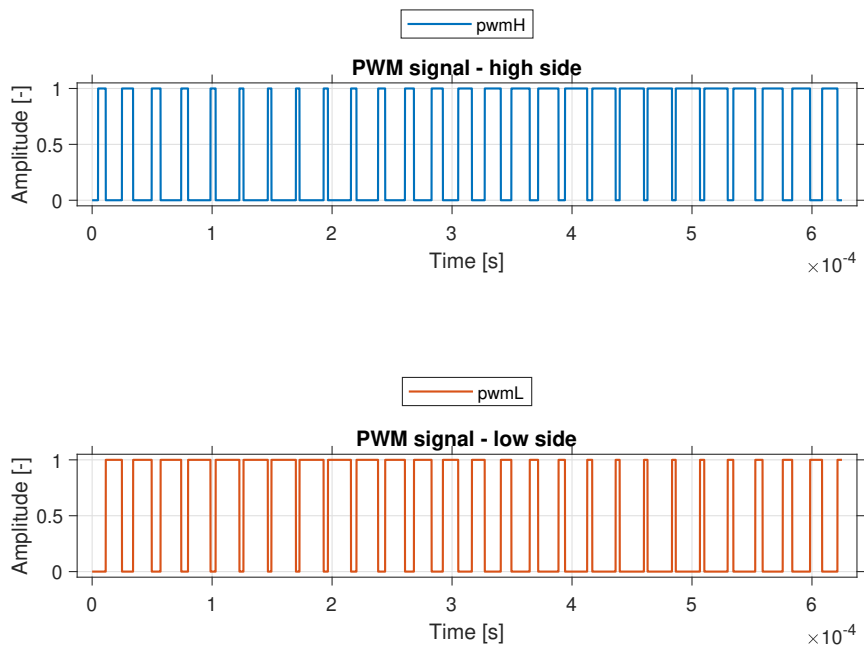


Figure 4.15: Generated PWM signals

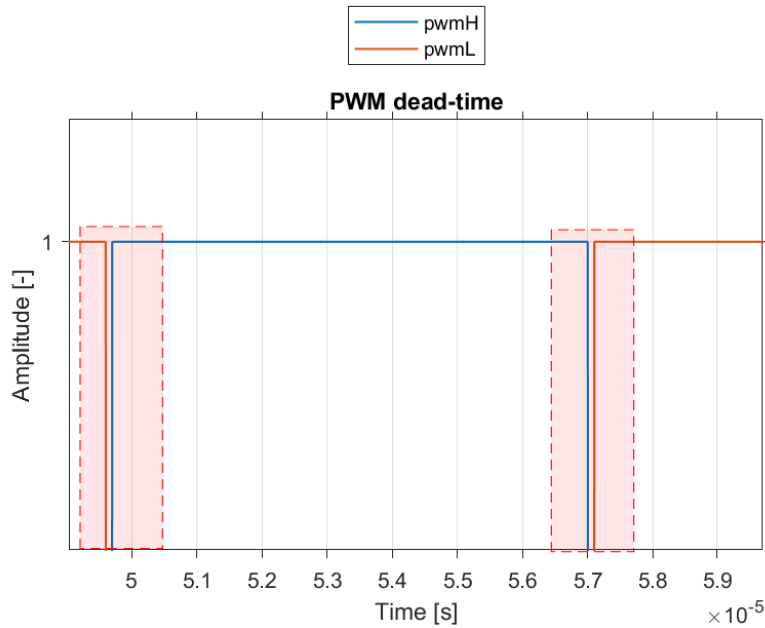


Figure 4.16: Dead-time

When switching from the high-side to low-side transistor and reverse, small delay must be implemented. This delay is called "Dead-time". This prevents the phase shortage of the inverter. Delay must be set accordingly to the on and off times of the transistors. Visualization of the dead-time can be seen on figure 4.16 (highlighted by the red rectangle). Length of the dead-time is calculated as [18]

$$t_{dead} = [(t_{d\_off\_max} - t_{d\_on\_min}) + (t_{pdd\_max} - t_{pdd\_min})]k_d, \quad (4.3)$$

where  $t_{d\_off\_max}$ ,  $t_{d\_on\_min}$ ,  $t_{pdd\_max}$ ,  $t_{pdd\_min}$ ,  $k_d$  represent transistors maximum turn-off delay, minimum turn-on delay, maximum propagation delay of the driver, minimum propagation delay of the driver and the safety factor. Safety factor  $k_d$  is equal to 1.2. [18]

## 4.2.5 Inverter

Several inverter types were used through the simulation making process. In the beginning, no inverter and PWM was used. Inverter adds another complexity to simulations, so the first simulation was mainly concerned with sensorless algorithm tuning. After successful tuning, inverter and PWM were added.

The first inverter used was the Average-value inverter from the Motor control block set. This inverter, represented by the figure 4.17, calculates phase voltages by knowing DC power supply voltage and duty cycles. It is easy to implement and validates the PWM algorithm.

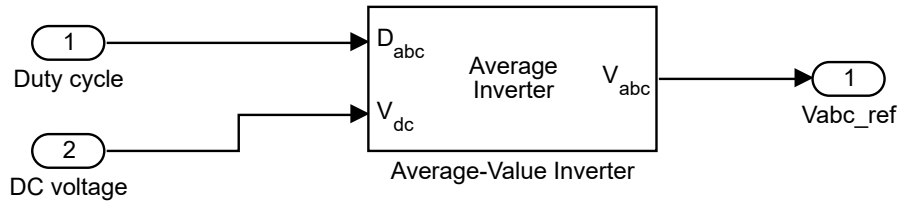


Figure 4.17: Average-value inverter

As mentioned above, the average-value inverter provides a good base start for the first steps of complete system simulations. On the other side, the results are far from real-world behaviour. To make results approach the real-world behaviour, the inverter could be modelled more in detail. With the use of Simulink Simscape, a variable switching inverter can be modelled. For this approach, models of IGBT transistors with diodes were used. Configuration of IGBT parameters matches their datasheet (depends on the selected IGBT transistor). Figure 4.18 represents mentioned variable switching inverter. This forms part of the high-fidelity simulation.

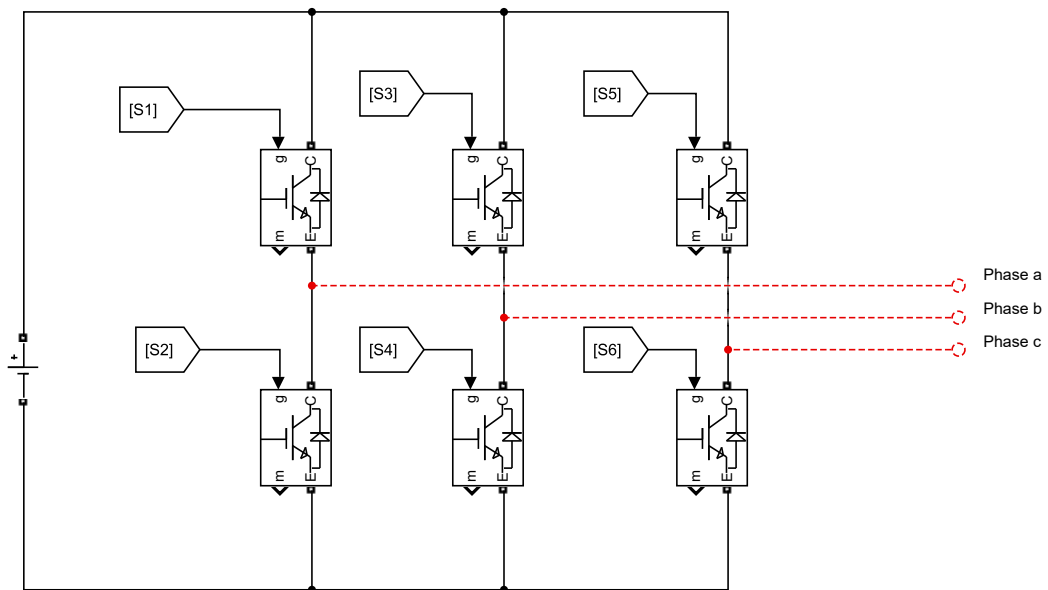


Figure 4.18: Variable switching inverter

## 4.2.6 PMSM

Same approach of using several models was used for the motors. All motors used are surface-mount type. For tuning purposes, linearized state-space model was used. This is a typical procedure for tuning almost all types of systems.

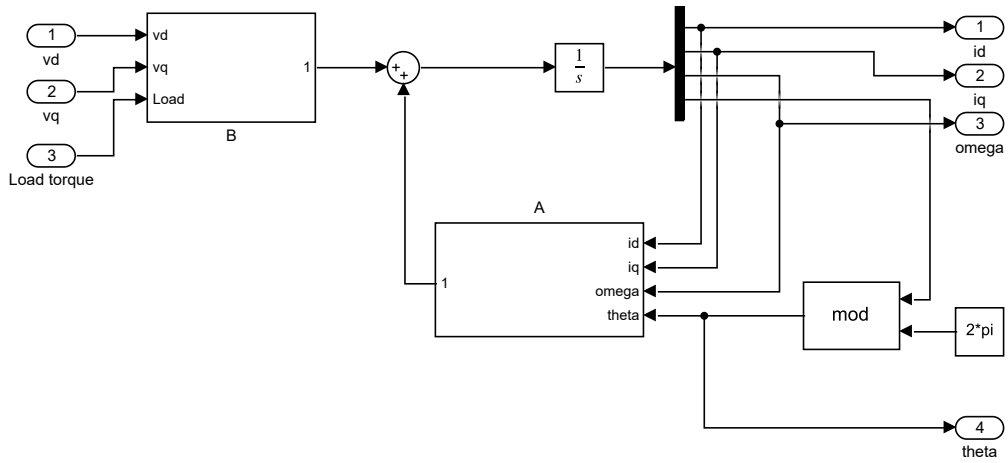


Figure 4.19: SPMSM state-space model

Through the controller strategy development, the SPMSM model from the Motor control block set was tested. One of its features is the easy transition between continuous and discrete models.

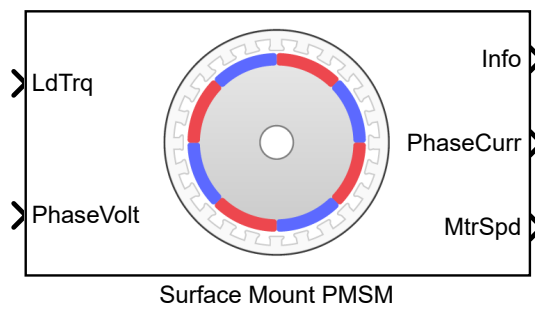


Figure 4.20: SPMSM Motor control blockset

SPMSM motor model on the figure 4.21 comes from the Simscape. This forms the second part of the high-fidelity simulation. In combination with a variable switching inverter, obtained results might better match real word behaviour.

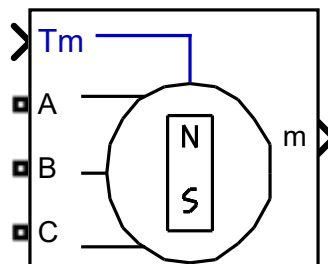


Figure 4.21: SPMSM high-fidelity model

### 4.3 Extended Kalman filter

To fully substitute speed and position sensors, a sensorless algorithm needs to be developed. The approach used by this thesis, called the extended Kalman filter, uses back EMF estimation. Position and speed are then obtained from the estimated back EMF. Derivation of the used algorithm was done in [19]. The main difference between the algorithm used in [19] and this thesis lies in the role of implementation. The algorithm in [19] was designed to operate on a field-programmable gate array (FPGA), but this thesis implements it on the digital signal processor (DSP). It was especially interesting how the chosen implementation will work on the DSP.

#### 4.3.1 Implementation

EKF derived in the [19] shows the process of optimization by reducing EKF order. Full non-optimized system (equation 4.5) is 4<sup>th</sup> order.

$$\dot{x}_{\alpha\beta} = Ax_{\alpha\beta} + Bu_{\alpha\beta} \quad (4.4)$$

$$\frac{d}{dt} \begin{bmatrix} i_\alpha \\ i_\beta \\ e_\alpha \\ e_\beta \end{bmatrix} = \begin{bmatrix} \frac{-R_s}{L_s} & 0 & \frac{-1}{L_s} & 0 \\ 0 & \frac{-R_s}{L_s} & 0 & \frac{-1}{L_s} \\ 0 & 0 & 0 & -\omega_r \\ 0 & 0 & \omega_r & 0 \end{bmatrix} \begin{bmatrix} i_\alpha \\ i_\beta \\ e_\alpha \\ e_\beta \end{bmatrix} + \begin{bmatrix} \frac{1}{L_s} & 0 \\ 0 & \frac{1}{L_s} \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} v_\alpha \\ v_\beta \end{bmatrix} \quad (4.5)$$

System can be divided into two separate 3<sup>th</sup> order systems (equations 4.6 and 4.7). This procedure is named decoupling.

$$\frac{d}{dt} \begin{bmatrix} i_\alpha \\ e_\alpha \\ e_\beta \end{bmatrix} = \begin{bmatrix} \frac{-R_s}{L_s} & \frac{-1}{L_s} & 0 \\ 0 & 0 & -\omega_r \\ 0 & \omega_r & 0 \end{bmatrix} \begin{bmatrix} i_\alpha \\ e_\alpha \\ e_\beta \end{bmatrix} + \begin{bmatrix} \frac{1}{L_s} \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} v_\alpha \end{bmatrix} \quad (4.6)$$

$$\frac{d}{dt} \begin{bmatrix} i_\beta \\ e_\beta \\ e_\alpha \end{bmatrix} = \begin{bmatrix} \frac{-R_s}{L_s} & \frac{-1}{L_s} & 0 \\ 0 & 0 & \omega_r \\ 0 & -\omega_r & 0 \end{bmatrix} \begin{bmatrix} i_\beta \\ e_\beta \\ e_\alpha \end{bmatrix} + \begin{bmatrix} \frac{1}{L_s} \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} v_\beta \end{bmatrix} \quad (4.7)$$

This optimization method was originally designed for FPGA to run in parallel. To properly test the effect of the optimization on the DSP. Processor in the Loop (PIL) simulations were run. Computation time of the EKF algorithm was obtained from the Matlab Simulink.



Task	4th order system	2 parallel 3th order systems
Initiation [ $\mu s$ ]	6.09	4.75
Step [ $\mu s$ ]	82.92	28.88
Terminate [ $\mu s$ ]	0.08	0.06

Table 4.1: EKF computation times

Matrices for the extended Kalman filter are obtained as,

$$F = \left. \frac{\partial f}{\partial x} \right|_{x=x(t)} = A, \quad (4.8)$$

$$H = \left. \frac{\partial h}{\partial x} \right|_{x=x(t)} = C, \quad (4.9)$$

$$\Phi(t_n, t_{n-1}, x(t_n)) \cong I + FT_c = \begin{bmatrix} 1 - \frac{R_s T_c}{L_s} & \frac{-T_c}{L_s} & 0 \\ 0 & 1 & \mp \omega_e T_c \\ 0 & \pm \omega_e T_c & 1 \end{bmatrix}. \quad (4.10)$$

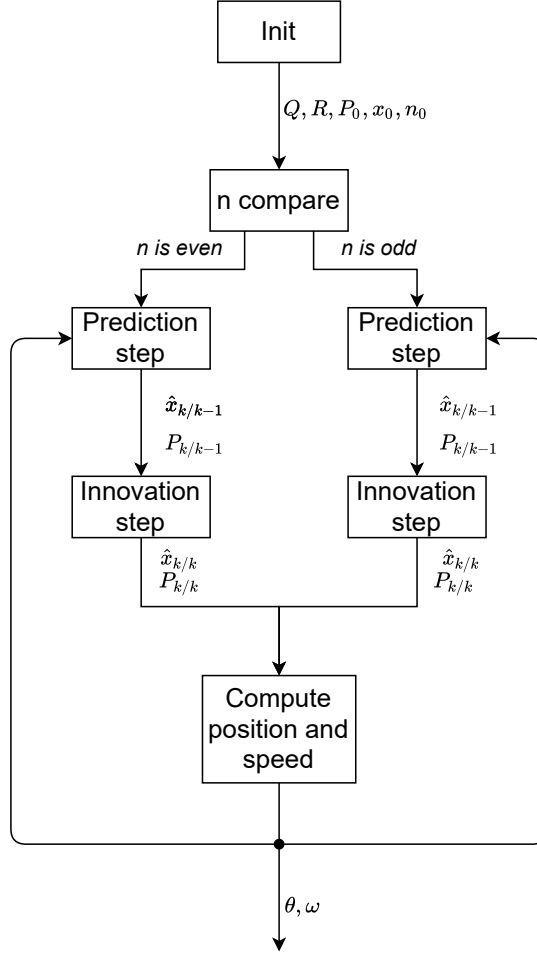


Figure 4.22: Parallel EKF flow-chart

Procedure of the extended Kalman filter, proposed in [19], is visualized by the flow chart diagram (figure 4.22). Block "Compute position and speed" uses equations 4.11 and 4.12 to get position and speed estimates. This calculation is based on estimated back EMF and magnetic flux. For accurate estimates, proper value of  $\lambda_m$  must be known.

$$\hat{\theta}_r(n) = \tan^{-1} \left( \frac{-\hat{e}_\alpha(n)}{\hat{e}_\beta(n)} \right) \quad (4.11)$$

$$\hat{\omega}_r(n) = \frac{1}{\lambda_m} \sqrt{\hat{e}_\alpha^2(n) + \hat{e}_\beta^2(n)} \quad (4.12)$$

To avoid inaccurate estimates, motor parameters dependency needs to be removed. Possible solution, which using only estimated back EMF, is called phase-locked loop (PLL) algorithm.

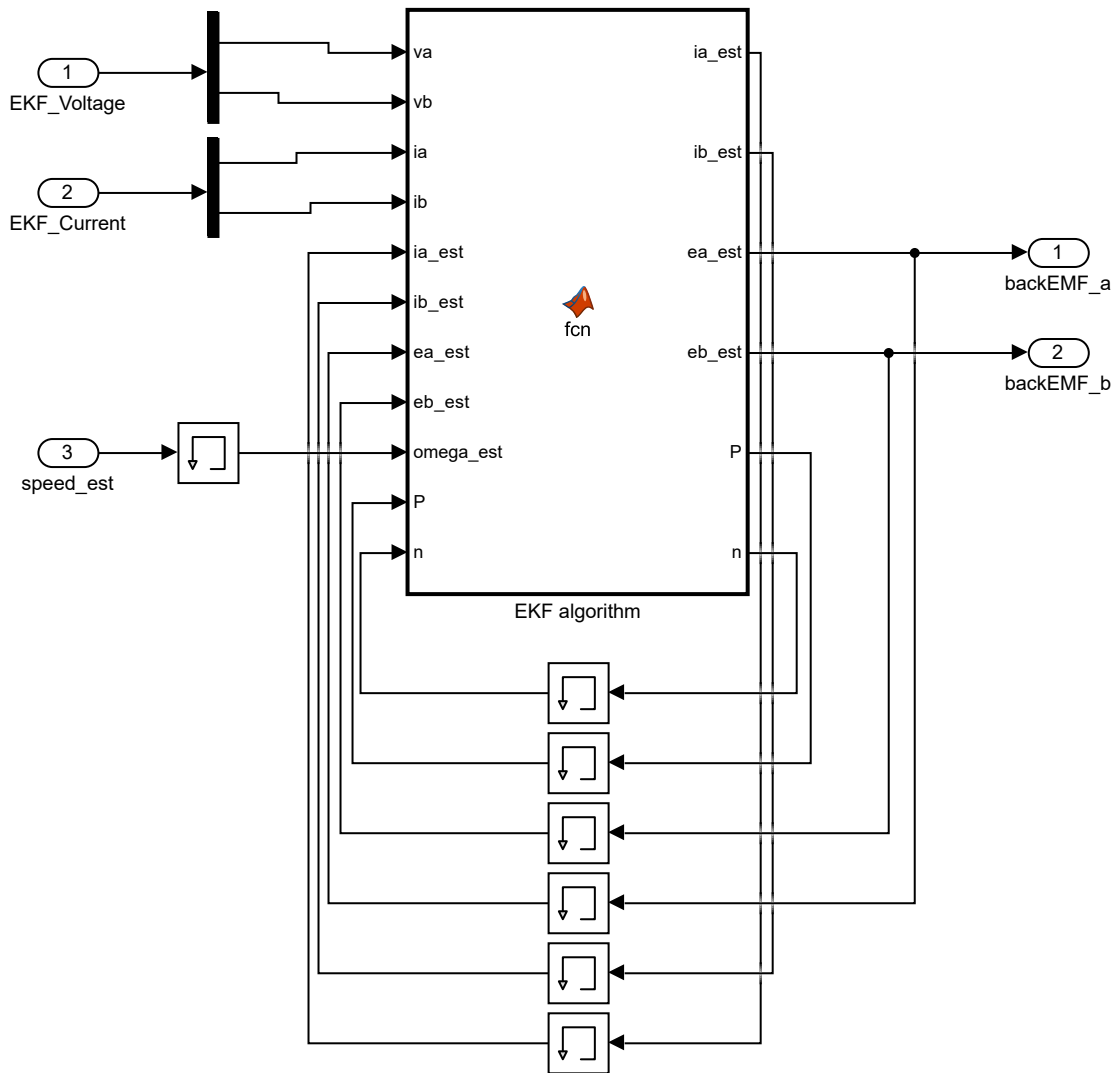


Figure 4.23: EKF parallel

### 4.3.2 Quadrature phase-locked loop (QPLL)

More robust way for extracting position and speed from the back EMF is PLL algorithm. This algorithm track phase of the back EMF and determines speed from its generated frequency. Speed is then integrated (in discrete time domain) to get position information.

Word quadrature is based on the nature of  $e_\alpha$  and  $e_\beta$  signals that are perpendicular to each other. The phases of the individual signals are shifted by  $90^\circ$  relative to each other. This makes it possible to monitor the rotating vector.

For simplicity, estimated back EMF signals are normalized as

$$\hat{e}_{\alpha n} = \frac{\hat{e}_{\alpha}^2}{\sqrt{\hat{e}_{\alpha}^2 + \hat{e}_{\beta}^2}},$$

$$\hat{e}_{\beta n} = \frac{\hat{e}_{\beta}^2}{\sqrt{\hat{e}_{\alpha}^2 + \hat{e}_{\beta}^2}}.$$
(4.13)

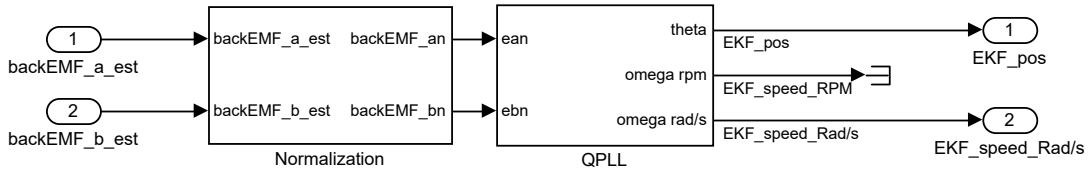


Figure 4.24: QPLL

After normalization, PLL algorithm computes speed and position estimates. Scheme of the whole algorithm is shown in the figure 4.25.

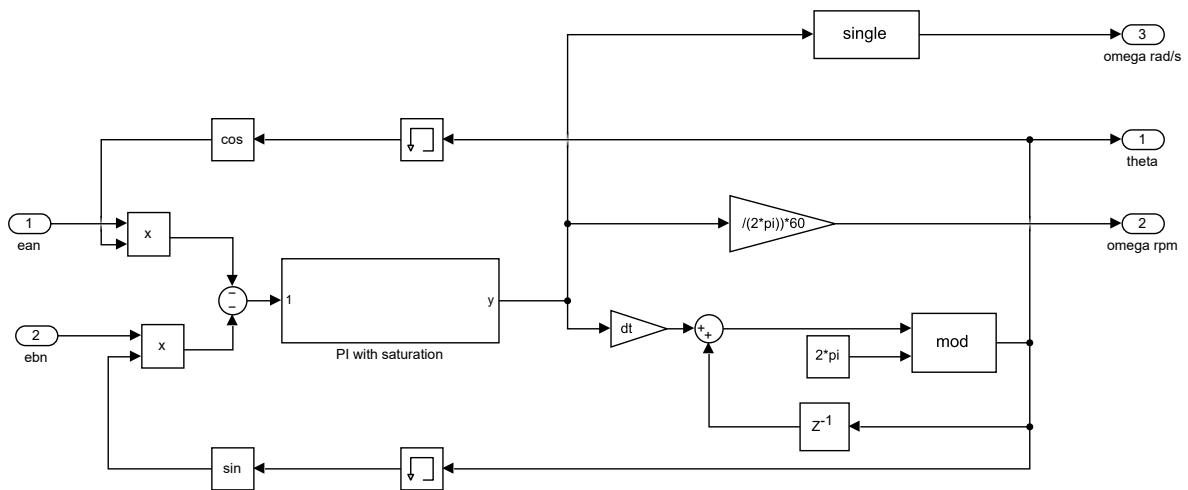


Figure 4.25: PLL

This combination of the EKF estimating back EMF and QPLL algorithm provide robust and reliable estimates used for the vector control.

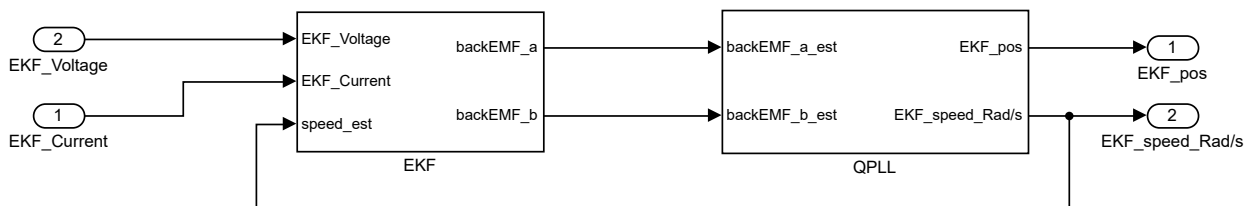


Figure 4.26: EKF with QPLL

## 4.4 Simulation results

### 4.4.1 I/f startup method

The simulation setup for the I/f startup algorithm can be configured for a specific motor. Whole startup process is visible in the figure 4.27. Constants used in this simulation are defined in the table 4.2. The first constant declares the speed of the ramp which generates the speed (position) command. When startup speed is reached, stabilization time is needed (Stabilization time 1) for stabilizing the rotor speed.

Startup speed [rad/s]	1000
Startup current [A]	10
Position error tolerance [rad]	0.02
Speed ramp time [s]	2
Stabilization time 1 [s]	2
$i_q$ decrease ramp time [s]	2
Stabilization time 2 [s]	2

Table 4.2: Startup constants setup

After complete stabilization, the alignment procedure starts. Current  $i_q$  starts decreasing. Constant  $i_q$  decrease ramp time determines decrease time from selected startup current to 0. Position error is detected in this window. Switching to vector control must occur before the current reaches 0. When the startup algorithm does a proper switch to the vector control, another stabilization time is implemented. During this time (Stabilization time 2) speed controller holds speed at the same value as startup speed.

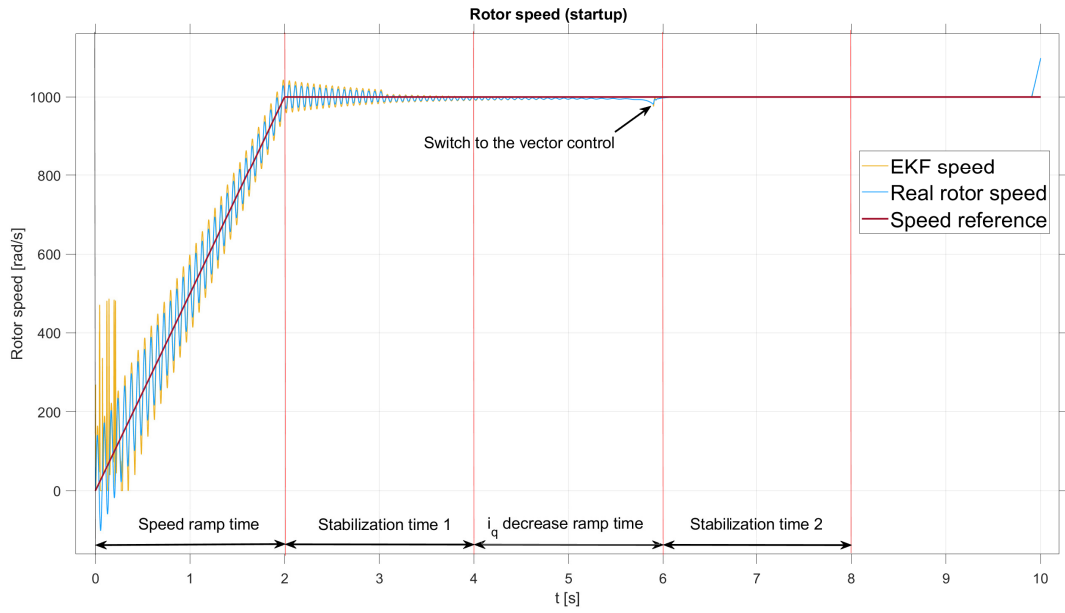


Figure 4.27: Rotor speed during startup

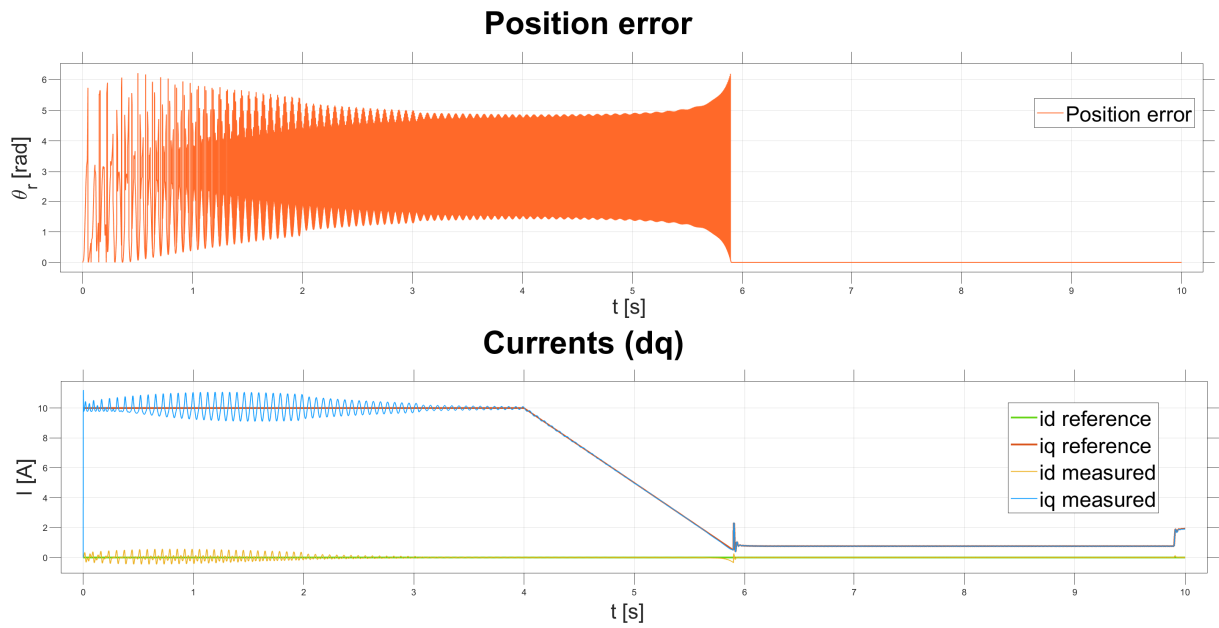


Figure 4.28: I/f startup position error and currents

Position error detection (figure 4.28) starts when motor reaches startup speed. To be sure, that the switch point is reliably detected, the position error tolerance constant is defined. While error falls under this tolerance (figure 4.29), startup complete command is switch on. After this, the controller works in the vector control mode.

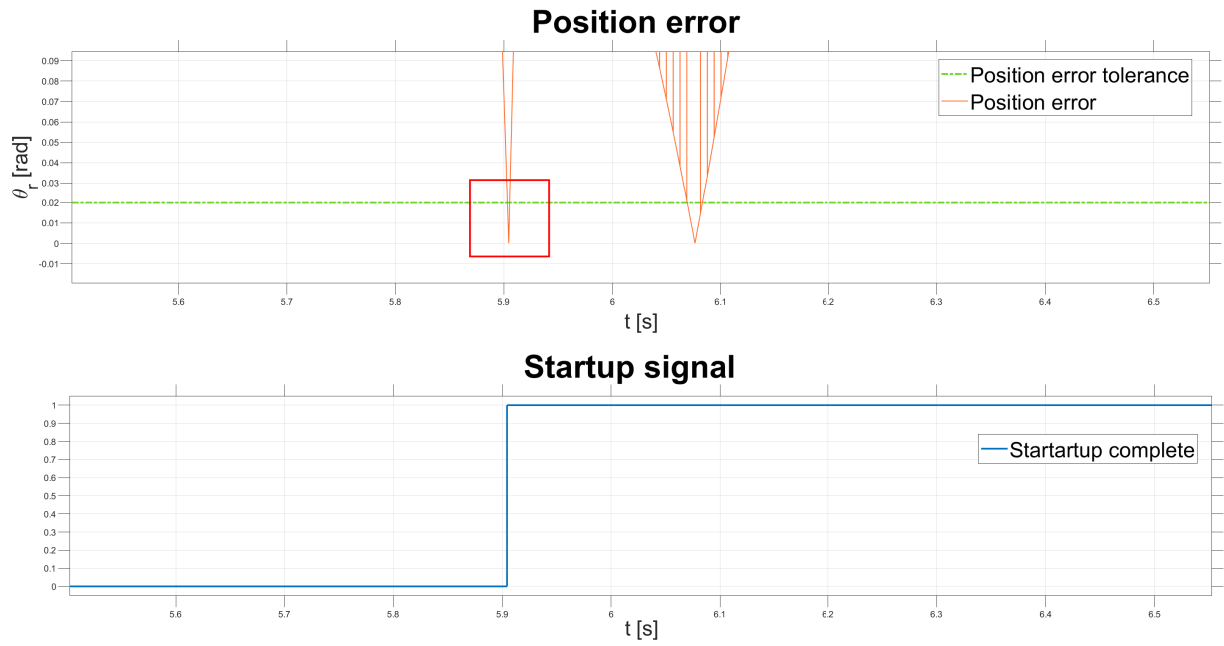


Figure 4.29: I/f startup position error (detail)

#### 4.4.2 Complete SPMSM sensorless controller

After I/f startup algorithm was tuned and simulated, whole system can be executed. Complete sensorless SPMSM controller run, described in the figure (4.30), consist of:

1. I/f startup method
2. Vector control mode (hold startup speed)
3. Vector control mode (Change of the speed reference)

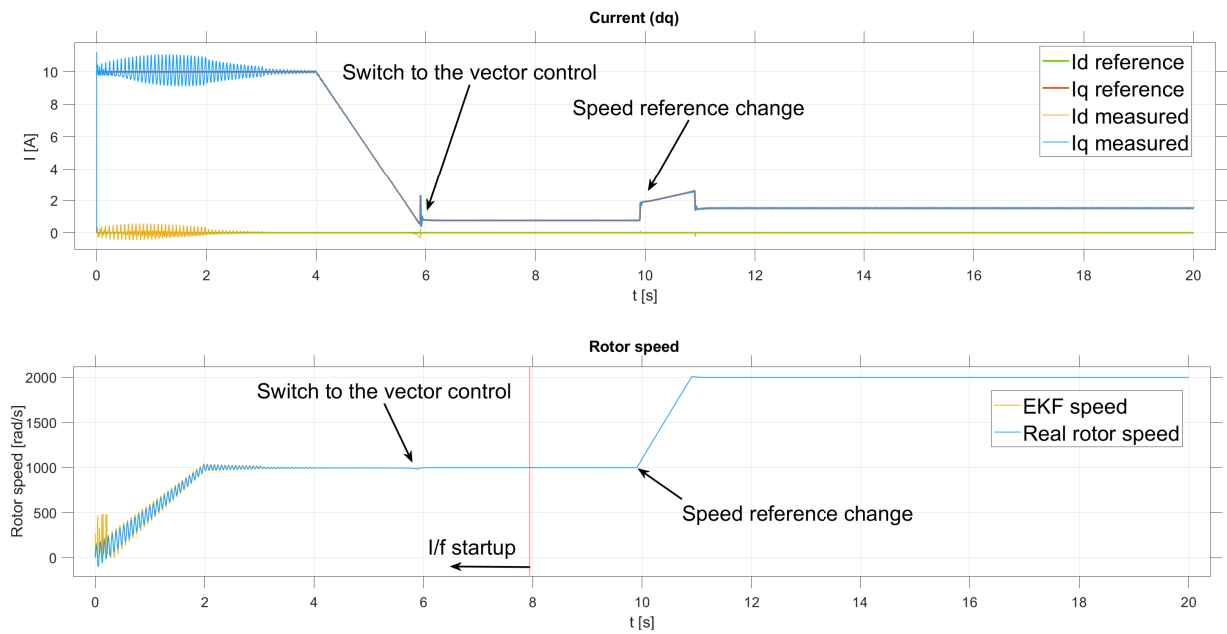


Figure 4.30: SPMSM sensorless controller - speed and  $dq$  currents

The functionality of the developed extended Kalman filter, in the role of position and speed estimator, is well described in the Rotor speed plot in the figure 4.30. EKF initialize during startup. After startup speed is reached, EKF is aligned with the real rotor speed. This shows the proper function of the selected sensorless algorithm. Differences between real rotor speed and estimated EKF speed are almost indistinguishable (figure 4.31).

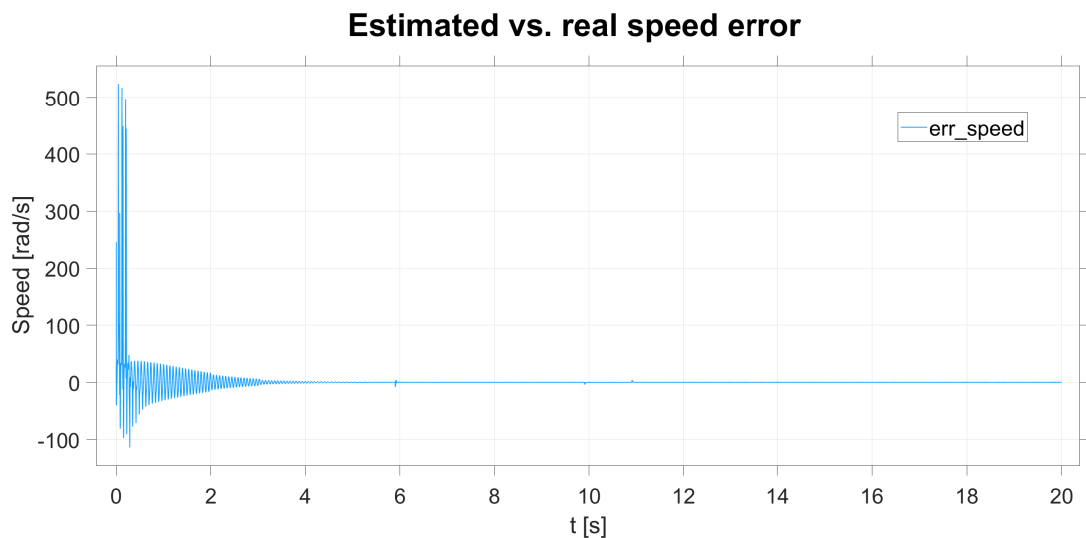


Figure 4.31: Error between real and estimated rotor speed



The voltage plot (figure 4.32) nicely shows the startup ramp and reference change. When looking at the currents plot (figure 4.33) high startup current is clearly visible. After switching to the vector control, the magnitude drops significantly. This shows how beneficial vector control is in comparison with open-loop control, i.e., I/f startup. Speed reference change at 10 seconds is again clearly observable.

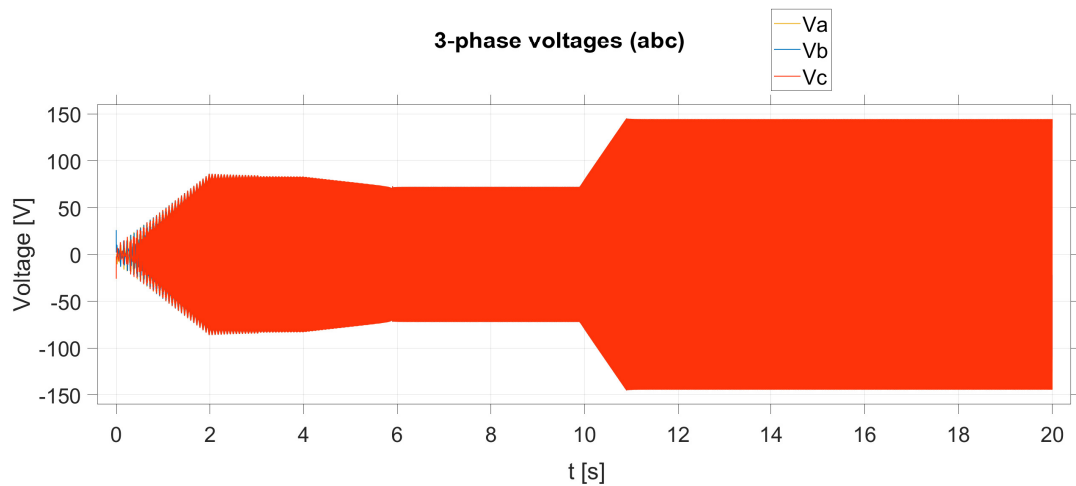


Figure 4.32: SPMSM sensorless controller - abc voltages

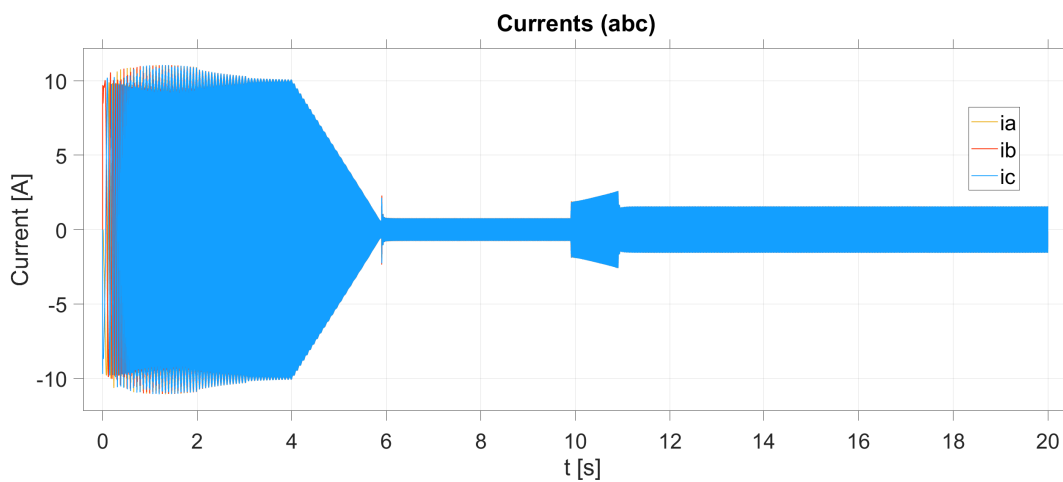


Figure 4.33: SPMSM sensorless controller - abc currents

### 4.4.3 High fidelity simulation

By introducing the previously mentioned high fidelity SPMSM and inverter models, more realistic behaviour could be achieved. This block implements simulation of the real switching IGBT transistors. Also, simulated measurements are affected. This results in a slightly more noisy estimates from EKF (figure 4.34). Even though the estimate is noisier, the real rotor speed still follows the reference speed perfectly.

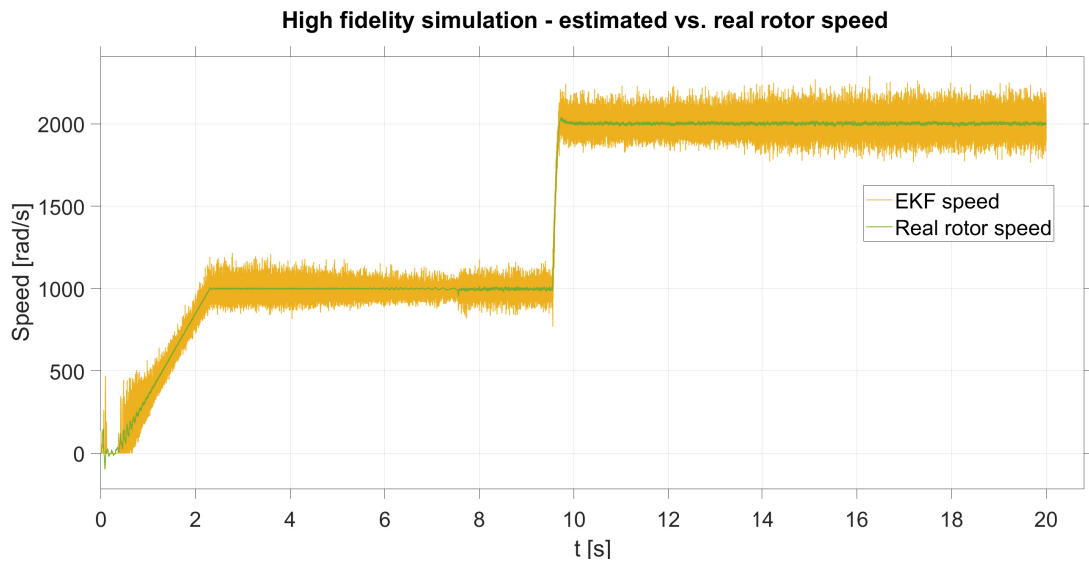


Figure 4.34: SPMSM sensorless controller - high fidelity simulation

# Chapter 5

## Hardware implementation

To successfully implement above mentioned SPMSM sensorless vector control strategy, several assumptions need to be made.

- Important parameters need to be tunable (regulators etc.)
- Data visualization
- Control from the host computer

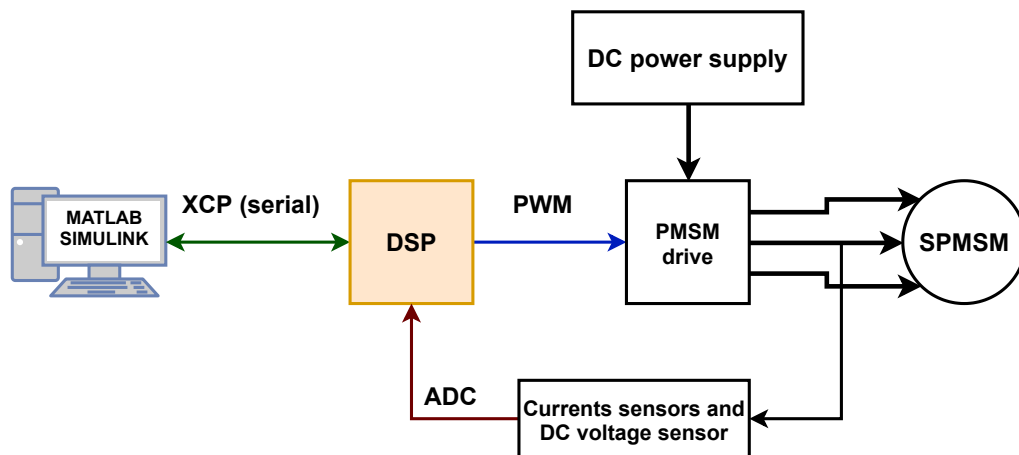


Figure 5.1: Validation setup

To meet all requirements, Matlab Simulink was chosen. Deployed program runs on DSP in the external mode [20]. This simulation mode provides full control over the running program. All parameters can be tuned and data are visualized in the Simulink Data Inspector. Communication with the computer is done with use of the Universal Measurement and Calibration Protocol (XCP). This protocol runs on top of the serial communication between computer (Matlab) and DSP. [20]

## 5.1 Digital signal processor

For the sensorless vector SPMSM controller, digital signal processor (DSP) from the Texas Instruments was selected. This processor, called C2000 TMS320F28379D, has two separate cores. This makes it suitable for high performance applications. Developed strategy can be partitioned to the both cores. This can lead to the higher overall performance [21]. The DSP was used with maximum 200 MHz CPU frequency.

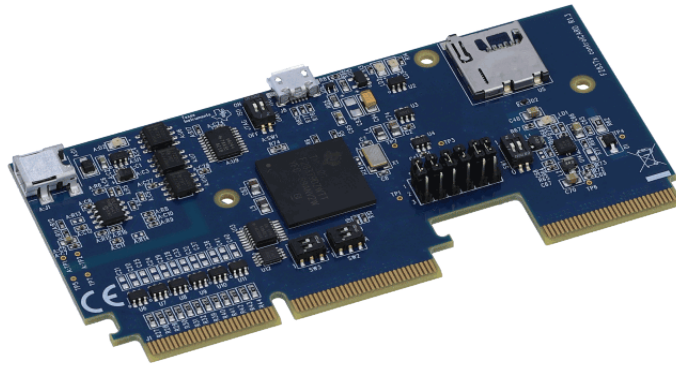


Figure 5.2: Delfino TMS320F28379D controlCARD

## 5.2 Permanent magnet synchronous motor

To complete implementation, motor parameters need to be known (table 5.1). This parameters are implemented into the final software.

Motor manufacturer	VUES Brno s.r.o.
Motor name	2AML406B-S
Motor type	SPMSM
Rated voltage [V]	179
Rated torque [Nm]	1.2
Rated current [A]	12.2
Rated speed [ $min^{-1}$ ]	25 000
Max speed [ $min^{-1}$ ]	40 000
Rated power [W]	3 140
Number of pole pairs [-]	1
Stator winding resistance [ $\Omega$ ]	0.396
D-axis inductance [H]	0.0011
Q-axis inductance [H]	0.0011
Moment of inertia [ $kg \cdot m^2$ ]	0.00011
Magnet constant [Wb]	0.072
Bearing friction coefficient [-]	$8.2 \cdot 10^{-5}$

Table 5.1: Motor parameters

## 5.3 Single core implementation

Unlike the simulation implementation, several blocks have to be added to the hardware implementation. To get measurements and provide inverter signals from the PWM peripherals an ADC (figure 6.9) needs to be configured.

### 5.3.1 Pulse width modulation

All PWM modules are configured to run in synchronous with each other. This is necessary to correctly modulate 3-phase voltages.

Switching frequency	20 kHz
Counting mode	Up-Down
ADC trigger	Middle of the period
PWM synchronization	Yes

Table 5.2: PWM configuration

Before the reference voltages are fed into the PWM comparator, normalization and scaling need to be done (figure 5.3). The measured voltage serves as a normalization value. Normalization and scaling process produces values from 0 to 1. Reference voltages are normalized with half of the measured DC power supply voltage. For practical reasons, saturation block is implemented to limit value from 0.1 to 0.9. This is mainly to ensure that the inverter does not operate at maximum and minimum voltage. This value is then scaled to PWM counter period (figure 5.4), which produces corresponding duty cycle value.

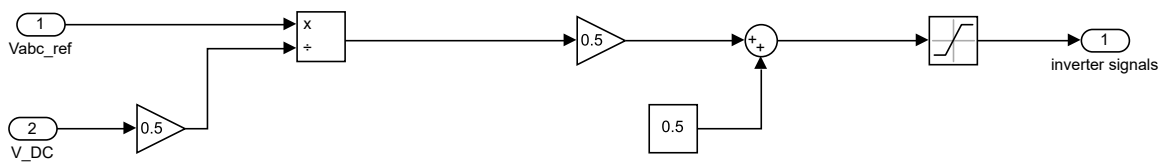


Figure 5.3: PWM scaling

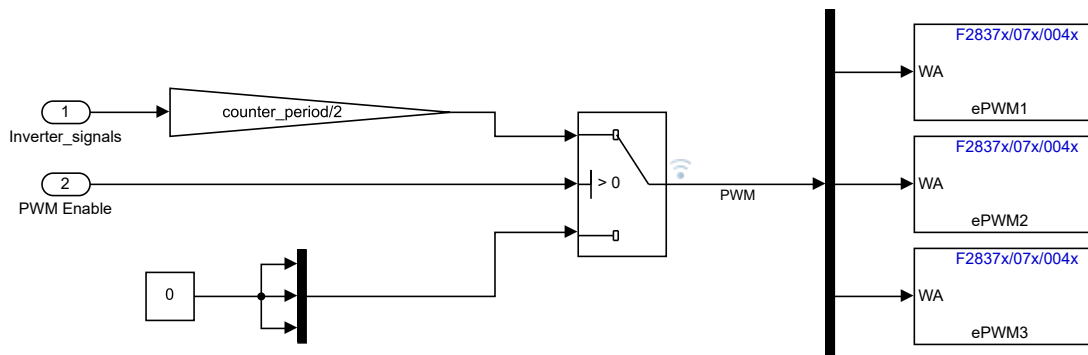


Figure 5.4: PWM implementation

### 5.3.2 Analog to digital converter

DC power supply voltage and two phase currents are measured. All ADC channels use same configuration (table 5.3).

Resolution	16-bit
Type	Differential
Data type	uint16
ADC trigger source	PWM

Table 5.3: ADC configuration

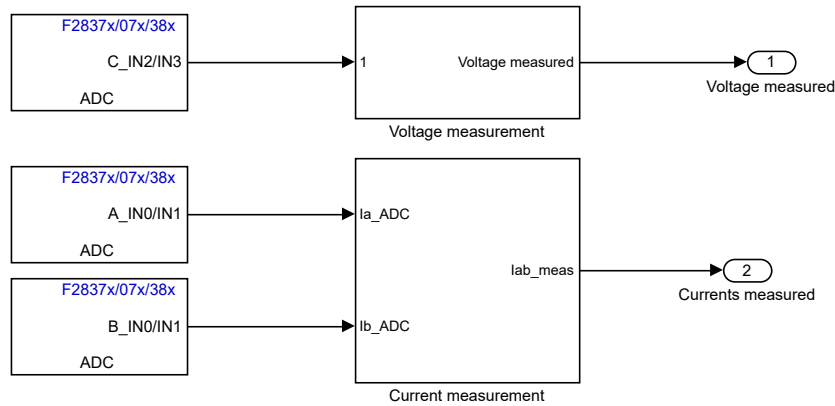


Figure 5.5: ADC measurement

### 5.3.3 PI regulators tuning

PI regulators' gains might differ from the simulation and real hardware. For rough gains estimate, model-based tuning approach [22] was chosen. These estimates were the fine-tuned by observing the motor behaviour. Final gains are shown in the table 5.4

$P(i_d)$	1
$I \cdot T_s(i_d)$	0.075
$P(i_q)$	1
$I \cdot T_s(i_q)$	0.075
$P(speed)$	0.005
$I \cdot T_s(speed)$	0.00015

Table 5.4: PI regulators' gains

The saturation of the current PI regulators must be set with respect to the DC power supply voltage. For this strategy, saturation limits settings are shown in the table 5.5.

To safely run the controller, PI regulators' saturations (table 5.5) need to be set with respect to the real hardware. In case of current regulators, small offset from the maximum value was added. Used PI regulators from the motor control require constant setting of the saturation value. In case of different voltages, PI regulators need to be changed to the type, where the saturation can be updated online.

Current	$\pm \frac{DC}{2} - 5$ [V]
Speed	$\pm 10$ [A]

Table 5.5: PI regulators' saturation

## 5.4 Dual core implementation

To achieve better overall computing performance, dual CPU approach was used. The single core strategy was partitioned based on the scheme in the figure 5.6. Results of this implementation are 2 individual programs (one for each CPU). Programs communicate with each other with the "Internal Processor Communication (IPC)". This communication is very fast and doesn't implement any delay. Behaviour of this implementation is the same as the single core approach, but with partitioned computational power. Based on that, several other algorithms could be added to this control strategy.

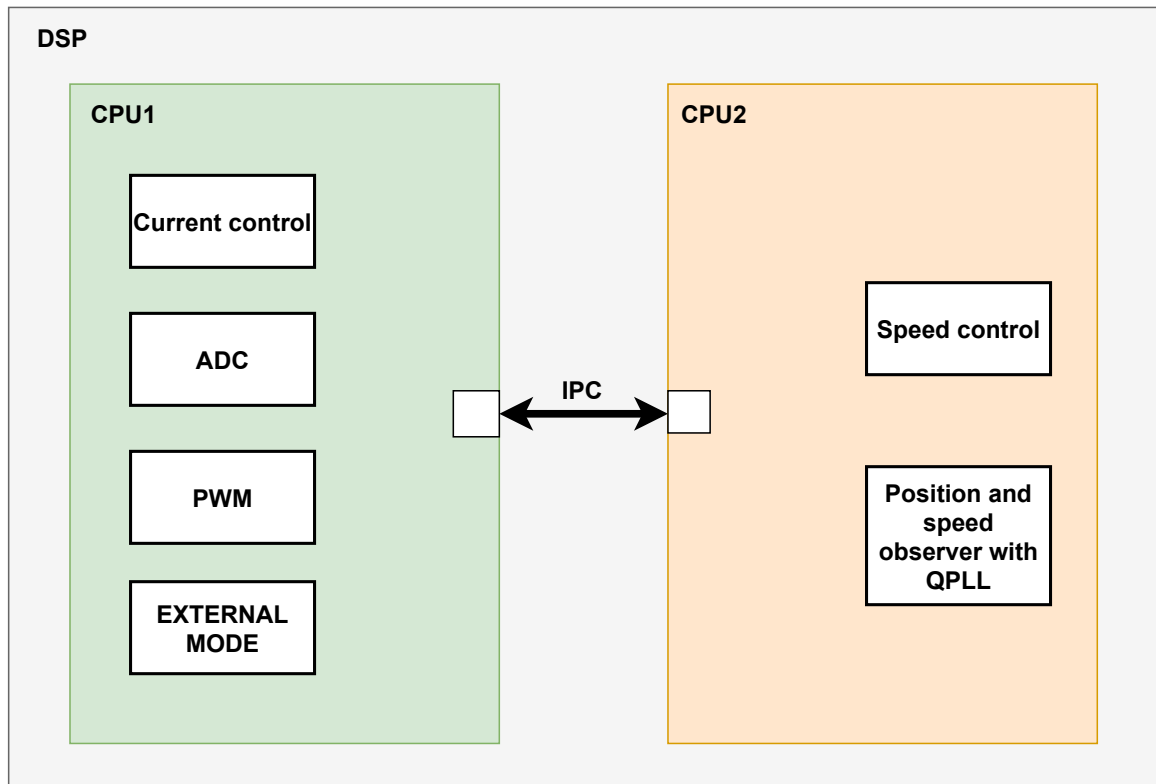


Figure 5.6: Dual-core DSP partitioning



# Chapter 6

## Control strategy validation on real hardware

The implemented sensorless PMSM control strategy successfully fulfill the requirements. PMSM can track speed reference (figure 6.1). The DC voltage of the power supply was set to 80 V.

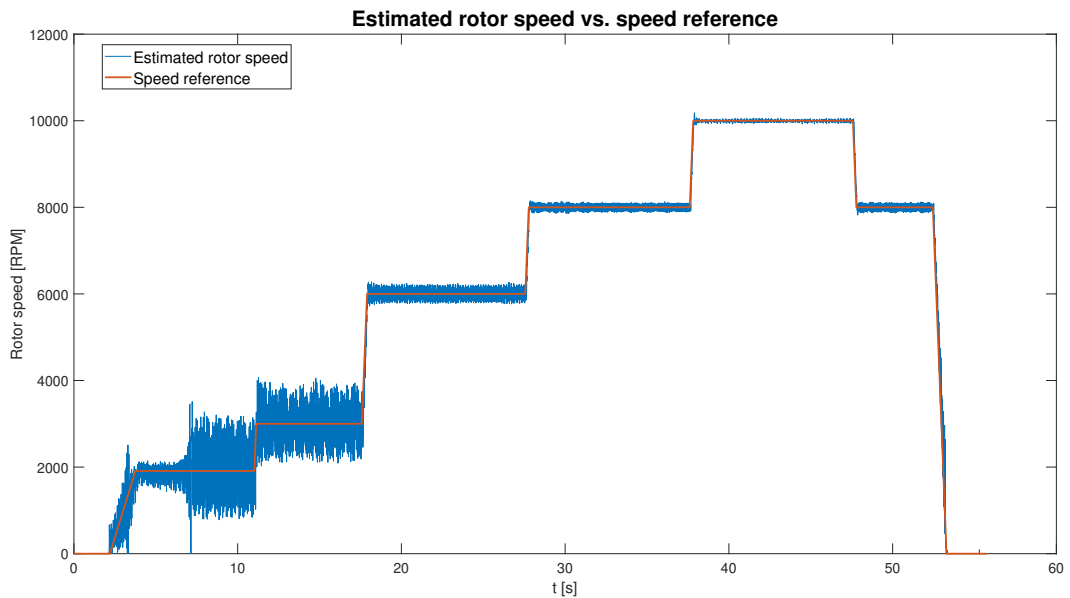


Figure 6.1: Real PMSM rotor speed estimation

By observing the estimated rotor speed signal, different speed levels have different noise amplitude. This is due to the lower estimated back EMF. Used PMSM has low current draw, when no load is applied. Due to that, current sensors are not measuring as precise as needed. With higher RPM, estimated speed noise is much lower. To analyze

the estimated speed quality, analysis with mean and standard deviation was made (figure 6.2 and table 6.1).

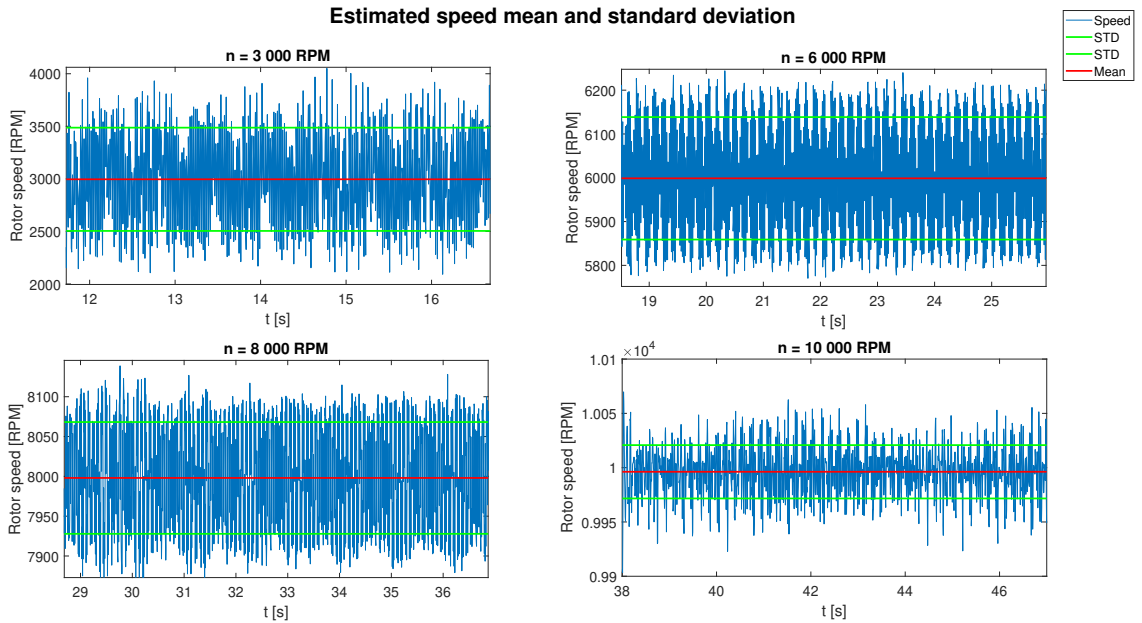


Figure 6.2: Estimated speed analysis

Rotor speed [RPM]	Mean value [RPM]	Standard deviation [RPM]
3000	2995.8	491.2
6000	5998.7	139.8
8000	7998	70.2
10000	9996.2	24.6

Table 6.1: Estimated speed mean and standard deviation

From standard deviation values, significant estimation quality increase with the rising speed is clearly visible. Mean values of the speed estimates are almost equal to the speed reference signal.

Whole current draw through the above showed run is shown in the figure 6.3 for the d-q reference frame. The startup procedure with the  $i_q$  current decreasing is visible at the beginning of this plot. As mentioned above, two phase currents are measured (figure 6.4 as feedback to the PI regulators. Due to the PMSM current draw, current measurements had significant noise. Therefore low-pass filters with the cut-off frequency of 10 kHz were added.

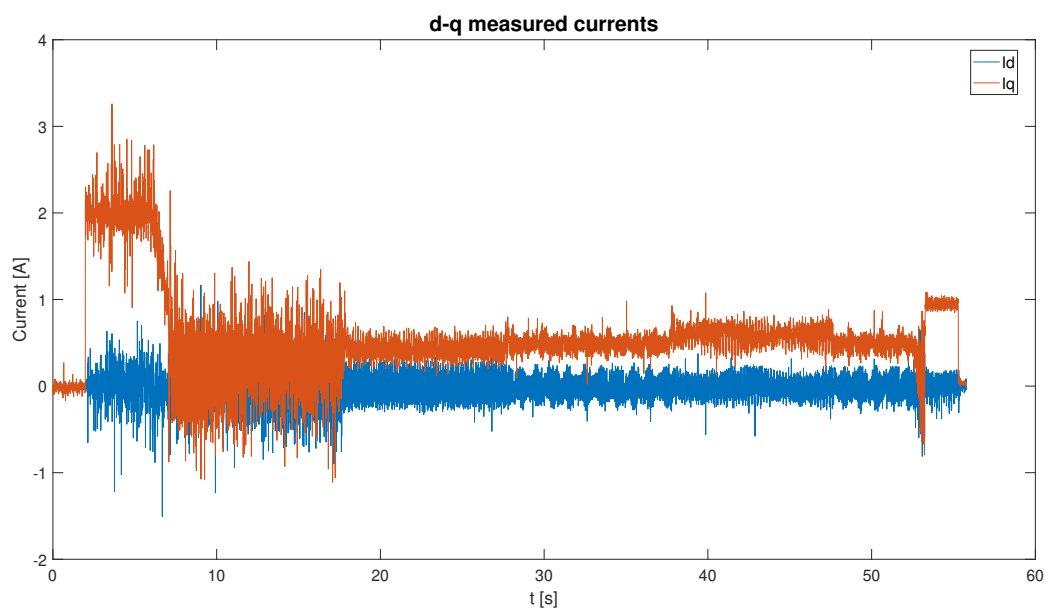


Figure 6.3: Measured d-q currents

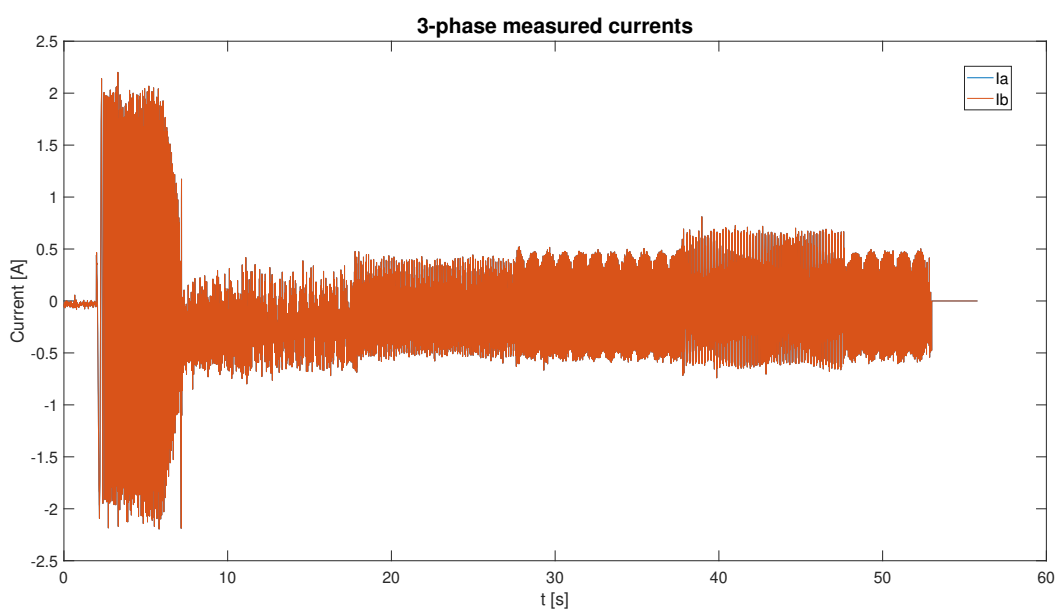


Figure 6.4: Measured ab currents

The output of the sensorless controller, 3-phase abc voltage, is shown in the figure 6.5. This signal is then fed to the PWM algorithm.

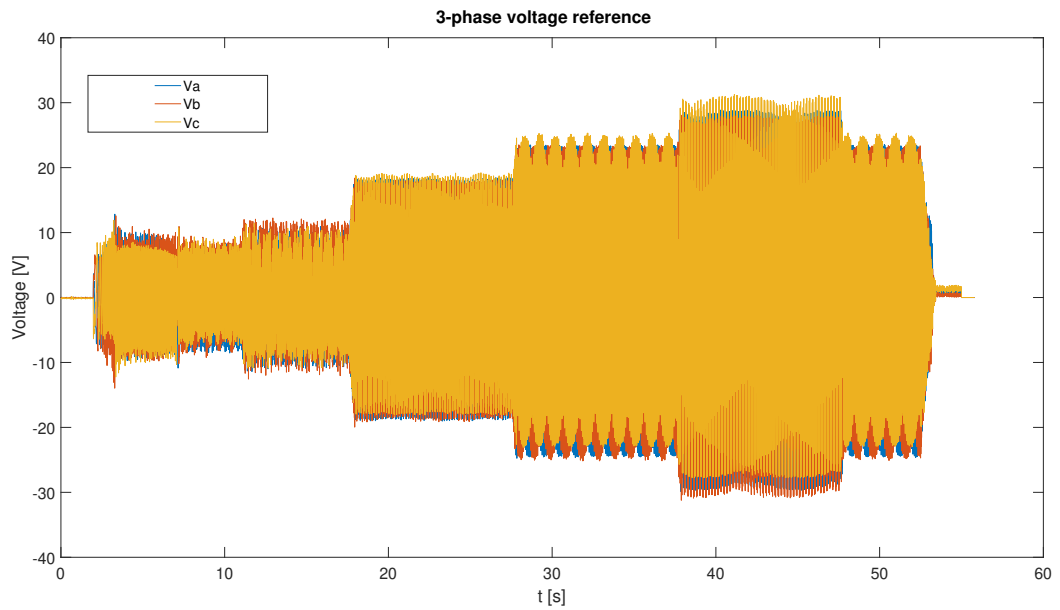


Figure 6.5: 3-phase reference voltage

To validate the real PMSM with respect to the simulated PMSM behaviour, simulation with the same parameter setting was done. From the rotor speed plot (figure 6.6) it is clear that speed reference tracking was successfully achieved. In comparison with the normal mode simulation, noise in speed estimation is negligible. Therefore, high fidelity simulation comes into play. Results of this simulation are much closer to the real run.

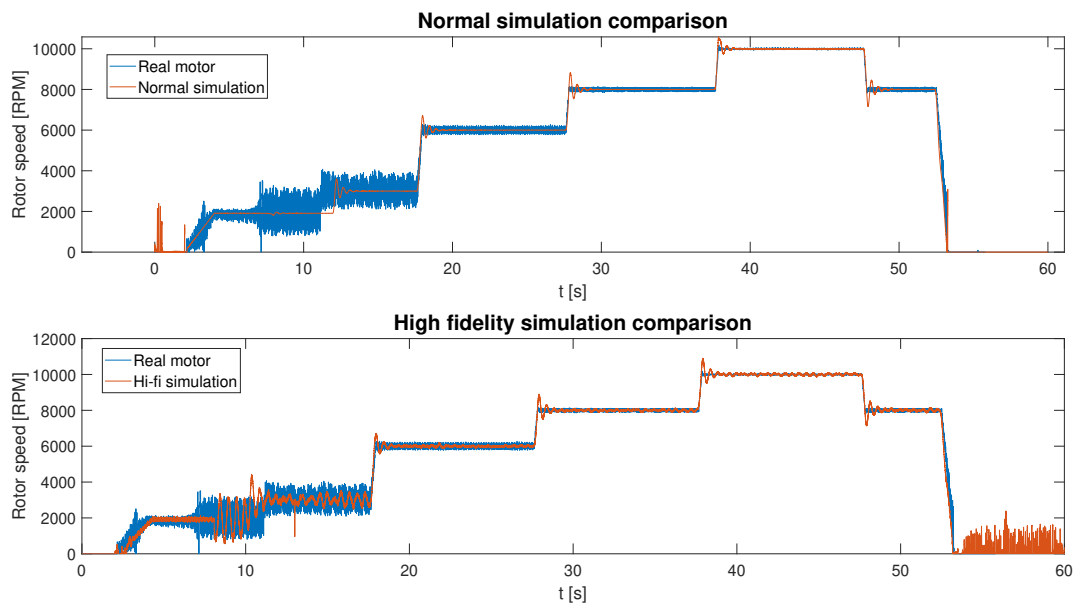


Figure 6.6: Real vs. simulated rotor speed estimation

Currents in d-q reference frame (figure 6.7 have a similar course. The only difference, apart from the noise, are current ripples when speed reference is changed. In the real PMSM, current ripples are suppressed.

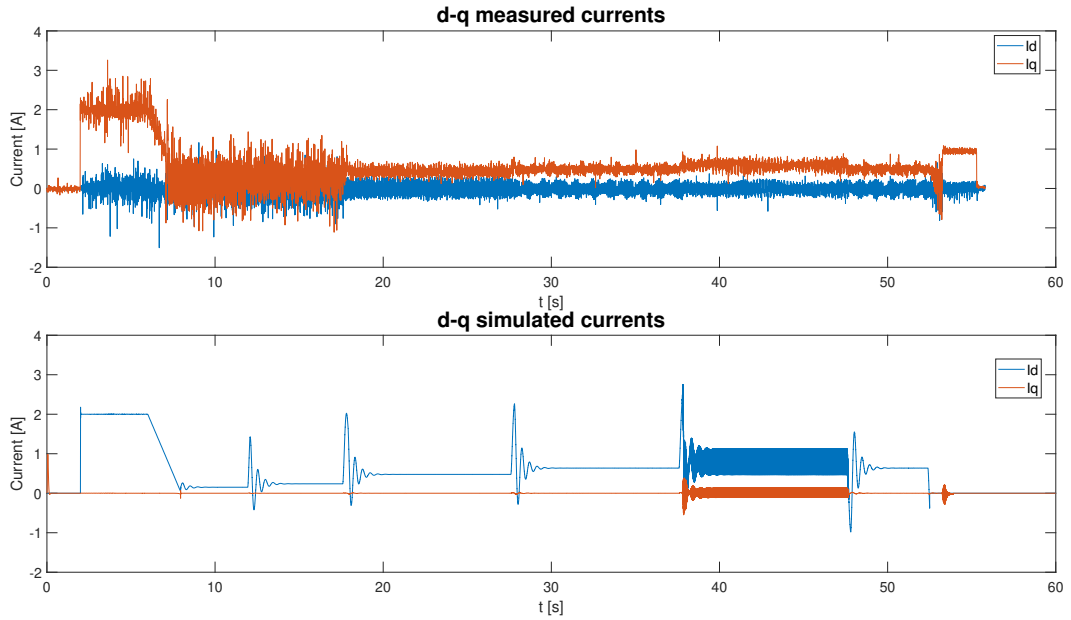


Figure 6.7: Real vs. simulated d-q currents

The last thing to validate is computational performance of the implemented PMSM sensorless controller. Implementation on the single CPU and dual CPU behaves the same. The main reason for the algorithm partitioning stays on the lowering computation times. This allows the addition of other algorithms, such as online parameter monitoring or higher PWM frequency with no loss of the computational performance. Execution times are shown in the figure 6.8. The partitioning was done as shown in the figure 5.6.

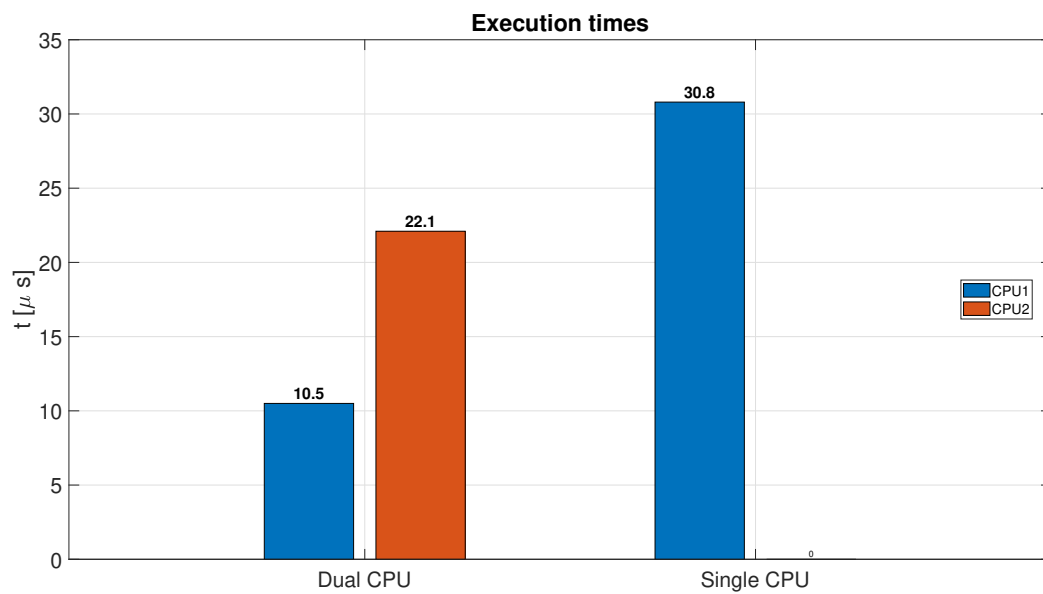


Figure 6.8: Execution times

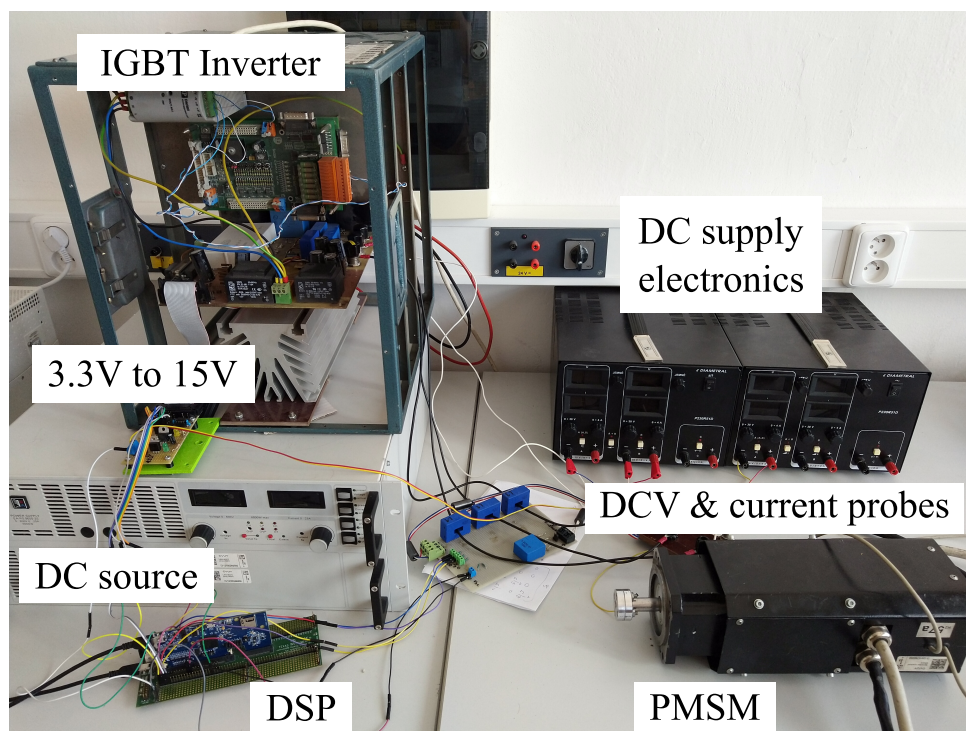


Figure 6.9: Validation setup

# Chapter 7

## Conclusion

The aim of this work was to design a functional sensorless control algorithm for controlling surface-mounted permanent magnet synchronous motors SPMSM. A research on sensorless algorithms was carried out to gain knowledge of the subject. The search mainly focused on the extended Kalman filter EKF estimating the back electromotive force (back EMF). Based on this knowledge, simulations were created in Matlab Simulink software. The EKF was optimized to reduce the computational time. The computational time of the optimized algorithm was reduced approximately 3 times. In such a fast system, such time saving is very noticeable. Since this sensorless algorithm used is based on the generation of back EMF, it is not possible for it to work from zero motor speed. Because of this, the I/f startup method has been implemented. This method uses the already implemented PI controllers of the system and thus there is no abrupt change during the transition from start to vector control.

The simulations served as a building block for implementation on a digital signal processor. Thanks to their results, the first functional versions were created. During the implementation, several problems had to be solved that were not visible in the simulations. Proper setup of the PI regulators, limitations and DSP peripherals was a major challenge. The chosen I/f startup method was susceptible to appropriate setup of the PI regulators. For this reason, the strategy was initially tuned using a open-loop scalar V/f startup method. The implemented SPMSM sensorless vector control strategy was subsequently fine-tuned and validated on a real SPMSM. The obtained measured values were then analyzed and compared against the simulations. The results of the simulations match the results obtained on the real SPMSM. A larger magnitude of noise is observed at lower speeds. This was due to the characteristics of the motor and the EKF itself. Since the motor used has a very low current draw in the unloaded state, the sensors did not measure optimally. The estimated back EMF then has a small amplitude and contains a large amount of noise. When the speed was increased, this estimate was already considerably

better.

The result of this thesis is a functional sensorless vector control algorithm for SPMSM that can run on a single or dual core DSP. From the comparison of computation times, there is a significant saving in computation time for the dual-core implementation. This solution provides opportunities to add additional algorithms used to optimize the SPMSM control or increase the PWM switching frequency.



# Bibliography

- [1] S. Morimoto, “Trend of permanent magnet synchronous machines,” *IEEJ Transactions on Electrical and Electronic Engineering*, vol. 2, no. 2, pp. 101–108, 2007. DOI: <https://doi.org/10.1002/tee.20116>. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1002/tee.20116>.
- [2] S. Vaez-Zadeh, *Control of Permanent Magnet Synchronous Motors*, English. 2018, ISBN: 9780198742968.
- [3] P. Perera, “Sensorless control of permanent-magnet synchronous motor drives,” English, Ph.D. dissertation, 2002, ISBN: 87-89179-41-2.
- [4] I. T. AG, *Permanent magnet synchronous motor (pmsm)*. [Online]. Available: <https://www.infineon.com/cms/en/applications/solutions/motor-control-and-drives/permanent-magnet-synchronous-motor/>.
- [5] A. Glumineau and J. De León-Morales, *Sensorless AC Electric Motor Control*. Jan. 2015, ISBN: 978-3-319-14585-3. DOI: 10.1007/978-3-319-14586-0.
- [6] *Overview of products by maxon: Maxon group*. [Online]. Available: <https://www.maxongroup.com/maxon/view/content/product-overview>.
- [7] *New second-generation triaxis® resolver motor position sensor*, 2017. [Online]. Available: <https://www.melexis.com/en/news/2017/09nov2017-mlx90380-triaxis-resolver-motor-position-sensor>.
- [8] *Mlx90380lgo-bab-103-re melexis: Mouser*. [Online]. Available: <https://cz.mouser.com/ProductDetail/Melexis/>.
- [9] C. J. V. Filho, D. Xiao, R. P. Vieira, and A. Emadi, “Observers for high-speed sensorless pmsm drives: Design methods, tuning challenges and future trends,” *IEEE Access*, vol. 9, pp. 56 397–56 415, 2021. DOI: 10.1109/ACCESS.2021.3072360.
- [10] Z. Zhang, “Sensorless control of synchronous machines using fundamental back-emf voltage - a review,” *IEEE Transactions on Power Electronics*, pp. 1–1, 2022. DOI: 10.1109/TPEL.2022.3162963.

- [11] G. Wang, M. Valla, and J. Solsona, "Position sensorless permanent magnet synchronous machine drives—a review," *IEEE Transactions on Industrial Electronics*, vol. 67, no. 7, pp. 5830–5842, 2020. DOI: 10.1109/TIE.2019.2955409.
- [12] S. Bolognani, S. Calligaro, and R. Petrella, "Design issues and estimation errors analysis of back-emf-based position and speed observer for spm synchronous motors," *IEEE Journal of Emerging and Selected Topics in Power Electronics*, vol. 2, no. 2, pp. 159–170, 2014. DOI: 10.1109/JESTPE.2013.2296974.
- [13] Z. Wang, K. Lu, and F. Blaabjerg, "A simple startup strategy based on current regulation for back-emf-based sensorless control of pmsm," *IEEE Transactions on Power Electronics*, vol. 27, no. 8, pp. 3817–3825, 2012. DOI: 10.1109/TPEL.2012.2186464.
- [14] L. Wang, Y. Zhang, L. Zhao, and G. Chen, "An improved 3-step startup method based on sensorless vector control of pmsm," in *2019 IEEE PES Asia-Pacific Power and Energy Engineering Conference (APPEEC)*, 2019, pp. 1–5. DOI: 10.1109/APPEEC45492.2019.8994460.
- [15] R. A. Walambe and V. A. Joshi, "Closed loop stability of a pmsm-ekf controller-observer structure," *IFAC-PapersOnLine*, vol. 51, no. 1, pp. 249–254, 2018, 5th IFAC Conference on Advances in Control and Optimization of Dynamical Systems ACODS 2018, ISSN: 2405-8963. DOI: <https://doi.org/10.1016/j.ifacol.2018.05.062>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2405896318302234>.
- [16] *Sensorless pmsm vector control with a sliding mode observer for compressors using mc56f8013*. [Online]. Available: <https://www.nxp.com/design/documentation/DOCUMENTATION>.
- [17] M. E. Asker and H. Kilic, "Modulation index and switching frequency effect on symmetric regular sampled spwm," *European Journal of Technique (EJT)*, vol. 7, pp. 102–109, 2017, ISSN: 2536-5010.
- [18] *Calculate and minimize the dead time for igbts*. [Online]. Available: [https://www.infineon.com/dgdl/Infineon-Deadtime\\_calculation\\_for\\_IGBT\\_modules-ApplicationNotes-v01\\_00-EN.pdf?fileId=db3a30431a5c32f2011a5daefc41005b](https://www.infineon.com/dgdl/Infineon-Deadtime_calculation_for_IGBT_modules-ApplicationNotes-v01_00-EN.pdf?fileId=db3a30431a5c32f2011a5daefc41005b).
- [19] N. K. Quang, N. T. Hieu, and Q. P. Ha, "Fpga-based sensorless pmsm speed control using reduced-order extended kalman filters," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 12, pp. 6574–6582, 2014. DOI: 10.1109/TIE.2014.2320215.

- [20] *External mode simulation by using xcp communication*. [Online]. Available: <https://www.mathworks.com/help/rtw/ug/external-mode-simulation-with-xcp-communication.html>.
- [21] *Tms320f2837xd dual-core microcontrollers datasheet*. [Online]. Available: [https://www.ti.com/product/TMS320F28379D?utm\\_source=google&utm\\_medium=cpc&utm\\_campaign=epd-c2x-null-prodfolderdynamic-cpc-pf-google-ww&utm\\_content=prodfolddynamic&ds\\_k=DYNAMIC+SEARCH+ADS&DCM=yes&gclid=Cj0KCQjwvqeUBhCBARIsA0dt45bzLw3eNrB-K0Szphfo\\_geHi068p3LeoBZ13S7-I-6ad2S\\_9xTWO\\_caArTIEALw\\_wcB&gclsrc=aw.ds](https://www.ti.com/product/TMS320F28379D?utm_source=google&utm_medium=cpc&utm_campaign=epd-c2x-null-prodfolderdynamic-cpc-pf-google-ww&utm_content=prodfolddynamic&ds_k=DYNAMIC+SEARCH+ADS&DCM=yes&gclid=Cj0KCQjwvqeUBhCBARIsA0dt45bzLw3eNrB-K0Szphfo_geHi068p3LeoBZ13S7-I-6ad2S_9xTWO_caArTIEALw_wcB&gclsrc=aw.ds).
- [22] K. Odo, E. Sochima Vincent, and C. Ogbuka, "A model-based pi controller tuning and design for field oriented current control of permanent magnet synchronous motor," vol. 14, pp. 35–41, Aug. 2019. DOI: 10.9790/1676-1404023541.

# List of Attachments

- Text of the thesis
- Single-core implementation
- Dual-core implementation
- Simulation
- High fidelity simulation
- Measured data

# List of used software

- MATLAB SIMULINK R2021b
- TI Code Composer Studio
- TI C2000Ware
- draw.io
- GIMP
- Overleaf