Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Electric Power Engineering

Isak Nordeng Jensen

Position-Sensorless Indirect Torque Control of Permanent Magnet Synchronous Machines

Master's thesis in Energy and Environmental Engineering
Supervisor: Roy Nilsen
Co-supervisor: Alexey Matveev
June 2021

NTNU
Norwegian University of
Science and Technology

Isak Nordeng Jensen

Position-Sensorless Indirect Torque Control of Permanent Magnet Synchronous Machines

Master's thesis in Energy and Environmental Engineering
Supervisor: Roy Nilsen
Co-supervisor: Alexey Matveev
June 2021

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Electric Power Engineering

NTNU
Norwegian University of
Science and Technology

Abstract

Position-sensorless permanent magnet synchronous machine (PMSM) drives are emerging as the state-of-the-art in high-performance safety-critical and weight-sensitive applications due to advantages such as high power and torque density, high efficiency and reliability and reduced hardware complexity, cost and size.

In this thesis, a state-of-the-art system-on-chip (SoC) based embedded controller is used for the development and implementation of C++ software that enables position-sensorless indirect torque control (ITC) of PMSMs. Two fundamental-excitation based methods for sensorless control are developed and programmed on the SoC central processing unit (CPU) and verified using an embedded real-time simulator (ERTS) that is housed by the SoC field-programmable gate array (FPGA). Both methods utilize the active flux concept for rotor position estimation, but the means of which the active flux vector is obtained differs between the two: The first method uses the Niemelä-corrected voltage model, while the second method is based on the voltage-current model.

It is shown that an inaccurately estimated stator resistance results in a flux estimate that drifts and becomes increasingly inaccurate over time, which makes position-sensorless operation without the implementation of specialized flux models impossible. The proposed flux models both prove to be highly effective for sensorless operation with an erroneous resistance estimate in the medium and high speed region. Moreover, both models enable crossing of the zero-speed region and persistent operation at very-low speeds below 0.1 per unit. Simulations are initially performed with a speed-dependent, quadratic load function that is valid for the modelling of pumps and fans, but the models are also verified for a scenario where a constant load torque is applied, suggesting that they are applicable for constant-load applications such as cranes and hoists.

The ERTS proves to be an immensely useful tool in the development of control software for PMSM drives. The control software on the SoC processor is intended to drive either a physical PMSM drive in a laboratory setup or the ERTS, and the latter enables the verification of the sensorless control strategies without accessing an actual PMSM drive test bench. Owing to the real-time nature of the ERTS, due to the high processing capability of the FPGA, simulations can be drastically accelerated and control strategy verification is made a less time-consuming effort than if conventional simulation tools were to be used.

Sammendrag

Sensorløse permanentmagnet synkronmotordrifter er i ferd med etablere seg som det foretrukne valget for anvendelser som krever stor grad av sikkerhet, pålitelighet og lav vekt. Dette kan tilskrives fordeler som høy effekt- og momenttetthet, virkningsgrad og pålitelighet og redusert maskinvare-kompleksitet, kostnad og størrelse.

I denne avhandlingen blir en styringsplattform basert på en "system on chip" (SoC) brukt for utvikling og implementering av C++-programvare som muliggjør sensorløs momentstyring av permanentmagnet synkronmaskiner. To metoder for sensorløs styring er implemenert og programmert på SoC-prosessoren og verifisert ved hjelp av en sanntidssimulator som er integrert i FPGAen på SoCen. Begge metodene benytter seg av aktiv fluks-konseptet for rotorposisjon-estimering, men fremgangsmåten som benyttes for å estimere aktiv fluksvektoren er ulik: Den første metoden benytter seg av den Niemelä-kompenserte spenningsmodellen, mens den andre er basert på spenning- og strøm-modellen.

Det blir demonstert at en unøyaktig estimert stator-resistans resulterer i et fluksestimat som "drifter" og blir unøyaktig over tid, noe som gjør sensorløs styring vanskelig. Begge fluksmodellene som benyttes viser seg å være høyst effektive for sensorløs styring med et unøyaktig resistans-estimat i medium- og høy-hastighetsområdet. Begge modellene muliggjør kryssing av null-turtallsområdet, samt vedvarende kjøring ved lave turtall under 0.1 per unit. Simuleringene blir i utgangspunktet gjennomført med en hastighetsavhenging, kvadratisk lastfunksjon som brukes for modellering av pumper og vifter. Modellene blir imidlertid også verifsert for et scenario hvor en last med konstant moment blir benyttet. Dette indikerer at metodene også er gyldige for anvendelser som heiser og kraner.

Sanntidssimulatoren som har blitt benyttet har vist seg å være et nyttig verktøy i utviklingen og verifiseringen av styringsprogramvare for permanentmagnet synkronmotordrifter. Programvaren som har blitt utviklet kan anvendes på en fysisk motordrift i et laboratorie eller på sanntidssimulatoren, og sistnevnte muliggjør testing av styringsprogrammvaren uten tilgang på en fysisk testrigg. Som en følge av FPGAens høye prosesseringshastighet kan simulatoren gjennomføre simuleringer i sanntid, noe som gjør verifisering av styringsstrategier en adskillig mindre tidkrevende prosess enn hvis konvesjoneller simuleringsverktøy hadde blitt brukt.

Preface

This thesis concludes my Master of Science degree in Energy and Environmental Engineering at the Norwegian University of Science and Technology, and is submitted to the Department of Electric Power Engineering. The thesis has been written in co-operation with Alva Industries AS.

Working with this thesis has exposed me to a broad range of exciting fields within electric power engineering, and has provided me with invaluable experience in conducting research and independently solving complex problems. There is however still much to learn, and I am eager to keep exploring these realms as I enter the next stage of life and start my professional career.

I would like to thank my supervisor, Professor Roy Nilsen, for his guidance and support during the last academic year. His knowledge and vast experience within the field of electric motor drives have been an immense resource throughout this project. I also thank my co-supervisor, Dr. Alexey Matveev at Alva Industries AS, for providing valuable insight into the world of electric motor manufacturing and the practical challenges that are encountered here.

During the course of this project, I have been fortunate enough to receive guidance and support from people who have chosen to share from their time despite lack of formal obligation to do so. Specifically, I extend my sincere thanks to Thomas Haugan for his help with setting up and becoming familiar with the software that has been used in this thesis, and Aravinda Perera, who in his doctoral thesis is working on a topic similar to the one that is investigated in this master's project. He has provided me with countless advice and shared from his knowledge across many fields, and in doing so, been a truly great support during the last months.

Last, but not least, I would like to thank my friends for making my time as a student at NTNU a memorable experience, and my family for their tireless support.

Trondheim, June 14, 2021

Isak Nordeng Jensen

Table of Contents

List of Figures

Nomenclature and Abbreviations

List of Symbols

U	=	voltage	[V]
u	=	voltage	[per unit]
I	=	current	[A]
i	=	current	[per unit]
R	=	resistance	[Ω]
r	=	resistance	[per unit]
Ψ	=	magnetic flux linkage	[Wb]
ψ	=	magnetic flux linkage	[per unit]
L	=	inductance	[H]
T	=	torque	[Nm]
τ	=	torque	[per unit]
T	=	time constant	[-]
T	=	period	[s]
$\boldsymbol{T}$	=	transformation matrix	[-]
J	=	moment of inertia	[kgm^2]
θ	=	electrical rotor position	[°] or [radians]
N	=	speed	[rpm]
n	=	speed	[per unit]
p	=	number of pole pairs	[-]
X	=	reactance	[Ω]
x	=	reactance	[per unit]
F	=	MMF	[A]
S	=	apparent power	[VA]
f	=	frequency	[Hz]
j	=	$\sqrt{-1}$	[-]
k	=	constant	[-]
H	=	transfer function	[-]

Subscripts

s	=	stator quantity
r	=	rotor quantity
a	=	phase a quantity, a-axis quantity, armature winding quantity
b	=	phase b quantity, b-axis quantity
c	=	phase c quantity, c-axis quantity
α	=	α-axis quantity
β	=	β-axis quantity
d	=	d-axis quantity
q	=	q-axis quantity
m	=	magnet
e	=	electromagnetic
l	=	load
n	=	nominal
u	=	voltage model quantity
i	=	current model quantity
samp	=	sampling
s	=	switch
ref	=	reference
0	=	initial value
LPF	=	low-pass filter
PI	=	proportional-integral controller
f	=	field, filter
filtered	=	low-pass filtered quantity
corr	=	correction

Superscripts

s	=	quantity referred to stator
r	=	quantity referred to rotor
T	=	transpose

Abbreviations

VFD	=	variable-frequency drive
PMSM	=	permanent magnet synchronous machine
IPMSM	=	interior permanent magnet synchronous machine
SPMSM	=	surface permanent magnet synchronous machine
AC	=	alternating current
DC	=	direct current
FE	=	fundamental excitation
HFSI	=	high-frequency signal injection
PWM	=	pulse width modulation
MMF	=	magnetomotive force
EMF	=	electromotive force
DTC	=	direct torque control
ITC	=	indirect torque control
VSC	=	voltage source converter
MTPA	=	maximum torque per ampere
SoC	=	system-on-chip
SoM	=	system-on-module
FPGA	=	field-programmable gate-array
CPU	=	central processing unit
ERTS	=	embedded real-time simulator
PESC	=	power electronic systems and components
PLL	=	phase-locked loop
PI	=	proportional-integral

Chapter 1

Introduction

1.1 Background

The world is currently undergoing a period of rapid change. The global energy demand is set to increase drastically over the course of the next years, and to mitigate climate change as a result of greenhouse gas emissions, large-scale deployment of renewable energy technologies is of the utmost importance [22]. The need for increased renewable energy penetration is currently leading to a massive increase in the utilization of electric AC machines across many sectors of society, including industrial applications, power generation, traction and transportation [6, 15]. Variable frequency drives (VFDs) play a major role in the shift towards a renewable energy-based society, as they are capable of providing large energy savings through optimization of the operating speed of the AC machine [11, 7]. Permanent magnet synchronous machines (PMSMs) are currently becoming the preferred machine type in many high-performance and weight-sensitive VFD applications due to their high power and torque density, high efficiency, reliability, simple construction and high fault-tolerance [30, 23].

Vector control is normally used in the control of modern PMSM drives in order to convert the machine into an equivalent DC machine which has highly desirable control characteristics. Such a transformation requires accurate information about the position of the machine rotor at all times [47, 28]. Traditionally, high-performance vector control of PMSMs have required the use of position sensors for rotor position extraction, but in recent years, position-sensorless PMSM drives have emerged as the state-of-the-art in safety-critical traction and automation applications. This is due to advantages such as increased reliability, reduced hardware complexity and cost, reduced size and lower maintenance require-

ments compared to conventional sensor-based PMSM drives [44, 19].

The available methods for position sensorless control of PMSMs, and variable-frequency drives in general, can be divided into two main categories. These are fundamental-excitation (FE) based methods, which rely on the estimation of the back-EMF and stator flux linkage of the machine for position estimation, and saliency-tracking based methods that estimate the rotor position through tracking of the spatial saliency of the machine [8, 52, 28]. The fundamental-excitation based methods are in general best suited for medium and high-speed operation when the induced back-EMF is large, and during zero and low-speed operation, such methods usually fail due to the lack of induced back-EMF. The saliency-tracking based methods, often referred to as high-frequency signal-injection (HFSI) based methods, work by injecting a high-frequency voltage signal on the top of the voltage fundamental, while measuring the stator current response which contains information about the spatial position of the rotor. The performance of such methods is thus independent of the motor speed and can hence be applied at standstill and very-low speeds. The accuracy of HFSI methods are usually related to the degree of saliency of the machine, however, and may as such not be a viable option for low-saliency PMSM designs [31]. Drawbacks of signal-injection based methods are increased current and torque ripple, that may cause undesirable vibration and acoustic noise, as well as increased complexity of the motor drive control system [28]. Strategies where FE-based schemes and HFSI methods are combined to achieve high performance across the complete speed range are presented in [51] and [44].

The development of motor drive control software has traditionally been done using physical motor drive setups for software verification early in the design process [14]. Such approaches are often tedious and cumbersome, as they require access to a laboratory environment with a fully functioning test bench. Following the recent improvements in FPGA technology, the use of hardware emulators is emerging as a viable alternative to the physical lab setups for control strategy verification [1, 21, 10]. Through the development of such emulators, the physical behaviour of the motor drive can be replicated and control software can be verified by letting it drive the hardware emulator instead of the physical motor drive. Such an FPGA-based emulator has recently been developed by the Power Electronics Systems and Components (PESC) research group in the Department of Electric Power Engineering at NTNU, and has been integrated in the department's state-of-the-art system-on-chip based embedded controller. The control platform also houses a processing system where motor control software that drives either a physical motor drive or the hardware emulator can be implemented. The control platform is meant to provide a common foundation for further research within the field of electric motor drives and to serve as a

starting point for the thesis work of future master's and PhD students.

1.2 Objective and Approach

The main aim of this thesis is to develop C++ subroutines that enable position-sensorless control a PMSM drive.

The following sub-objectives have been defined:

- Develop software that implements two separate fundamental-excitation based sensorless control strategies.
- Verify both strategies and control software using a PMSM drive hardware emulator.
- Provide a comparison of the aforementioned sensorless control strategies.

The thesis objectives will be accomplished using an approach based on the following sequential steps:

1. Initially, a literature review that covers PMSM drive theory and sensorless control concepts are studied with the intention of becoming familiar with the theory that is relevant for this project.
2. Next, sensorless control strategies will be developed and implemented in the embedded controller using the C++ programming language.
3. Finally, real-time simulation of the PMSM drive during position-sensorless operation will be performed using the hardware emulator in order to validate the viability of the proposed methods.

1.3 Limitations

The work that is performed in this thesis is put under the following constraints:

- Only fundamental-excitation based methods for sensorless control will be considered.
- Standstill operation of the motor will not be examined.

- The drive will be investigated during torque controlled operation only.
- The control software will not be verified on a physical PMSM motor drive, and the validation process will as such rely solely on the FPGA-based hardware emulator.

1.4 Structure of Thesis

This thesis consists of a total of 6 chapters:

Chapter 1 - Introduction contextualizes the work that is performed in this project, and contains a presentation of the main thesis objectives, the approach that will be used and the constraints the work will be put under.

Chapter 2 - Theoretical Background provides relevant background theory, including permanent magnet synchronous machine modelling, control principles and an introduction to the control platform that will be used for implementation and verification of control software.

Chapter 3 - Position-Sensorless Indirect Torque Control of the PMSM is dedicated to the description and discussion of the sensorless control strategies that are implemented in this thesis and the software development process.

Chapter 4 - Real-Time Simulation Results contains the real-time simulation results that are obtained in order to verify the control software.

Chapter 5 - Discussion highlights the most important findings from the real-time simulations and discusses their implications.

Chapter 6 - Conclusion and Further Work presents the most important conclusions that can be drawn from the work that is conducted in the thesis, as well as suggestions for further work.

Chapter 2

Theoretical Background

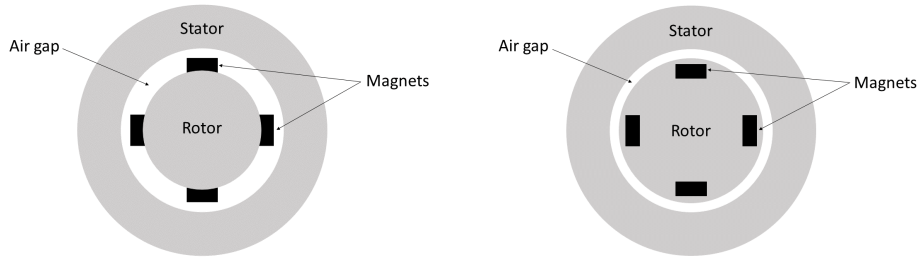
This chapter presents the fundamentals of the theory that the work performed in this thesis is founded upon. A mathematical model of the ironless PMSM will be presented, along with the basics of the control strategy that will be implemented. Finally, the hardware and software that will be used for control implementation and verification will be introduced.

The theory presented in this chapter is to some extent based on the theory presented in the author's specialization project that was written in the fall of 2020 [26]. It has however been revised, and significant modifications have been made to obtain the current version.

2.1 The Permanent Magnet Synchronous Machine

The permanent magnet synchronous machine (PMSM) is a type of synchronous machine in which the magnetic field excitation of the rotor is provided using permanent magnets. This allows for a simple, reliable and highly compact construction. Similarly to conventional syncronous machines, the stator of the machine contains three wye-connected, sinusoidally distributed windings that are identical and symmetrically displaced at 120 degrees. By supplying the stator windings with AC power using a three-phase power electronics converter, the frequency of the stator currents can be varied through pulse-width modulation (PWM). This makes it possible to regulate the speed of the rotating magnetic field in the air gap of the machine, and hence the speed of the rotor. This is the main principle of operation of PMSM-based variable frequency drives.

There are in general two main ways of mounting the permanent magnets on the rotor of the machine, as shown in Figure 2.1. The placement of the permanent magnets will affect the behaviour and the characteristics of the machine. A machine with surface-mounted permanent magnets yields the simplest construction, but the rotor magnets will in this case be prone to mechanical stress. If the permanent magnets are interior-mounted, the mechanical stress on the magnets will reduce and durability of the machine may increase. A consequence of interior-mounted magnets is that the machine becomes salient, meaning that the d- and q- axis inductances are unequal, which results in the generation of reluctance torque. This is due to the fact that the permanent magnets have a relative permeability that for all practical purposes can be assumed to be equal that of air. The concept of the dq coordinate system will be explained in detail later in this chapter. From Figure 2.1, it is evident that the air gap of the machine becomes larger in the case of surface-mounted magnets. This results in lower machine inductances, which gives a machine with fast speed response, but also large current ripple which may cause undesirable torque and speed pulsations.



(a) SPMSM **(b)** IPMSM

Figure 2.1: Permanent magnet synchronous machines with surface-mounted and interior-mounted permanent magnets.

2.1.1 The Space-Vector Concept

Modelling and control of electrical machines usually requires utilization of space vectors. In 1959, the space vector concept for multi-phase machines was developed by Kovács and Rácz and presented in [29]. They proposed to represent multi-phase machine quantities as a single vector variable, and in doing so, they laid the foundation of vector control of variable-frequency drives.

The concept can be demonstrated by considering the magnetomotive force (MMF) in the air-gap of the PMSM resulting from the stator phase currents. The MMF is a function of the instantaneous value of the phase currents $i_a(t)$, $i_b(t)$ and $i_c(t)$, as well as the angle

displacement with respect to the a axis, θ, and is given by the following expression [36]:

$$F(\theta, t) = N\left[i_a(t)\cos(\theta) + i_b(t)\cos\left(\theta - \frac{2\pi}{3}\right) + i_c(t)\cos\left(\theta + \frac{2\pi}{3}\right)\right] \tag{2.1}$$

Here, N is the effective number of turns per winding. If one defines the unit vectors $\underline{a}$, $\underline{b}$ and $\underline{c}$ to point in the direction of the winding axes, the MMF can be rewritten as a vector quantity, or a *space vector*, as shown below:

$$\begin{aligned}\underline{F}(t) &= N\left[i_a(t)\cdot\underline{a} + i_b(t)\cdot\underline{b} + i_c(t)\cdot\underline{c}\right] \\ &= N\left[\underline{i}_a(t) + \underline{i}_b(t) + \underline{i}_c(t)\right] \\ &= \underline{F}_a(t) + \underline{F}_b(t) + \underline{F}_c(t)\end{aligned} \tag{2.2}$$

In Equation 2.2, the phase current space vectors $\underline{i}_a(t)$, $\underline{i}_b(t)$ and $\underline{i}_c(t)$ have been introduced and the resultant air-gap MMF space vector has been expressed as the sum of the MMF space vector for each of the phases. Similarly, the resultant stator current space vector can be defined as

$$\underline{i}_s^s(t) = \underline{i}_a(t) + \underline{i}_b(t) + \underline{i}_c(t) = \begin{bmatrix} i_a(t) \\ i_b(t) \\ i_c(t) \end{bmatrix} \tag{2.3}$$

Space vectors for stator voltages and flux linkages can be defined in a similar manner, and all three-phase machine variables can hence be represented as vector quantities in the two-dimensional vector space. This concept is fundamental for the development of the mathematical model of the machine that will be presented in the next section, and forms a basis for the concept of vector control that will be presented later in this thesis.

Figure 2.2 illustrates the space-vector concept. Here, the resulting current space vector when the phase a current is at its maximum value is displayed.

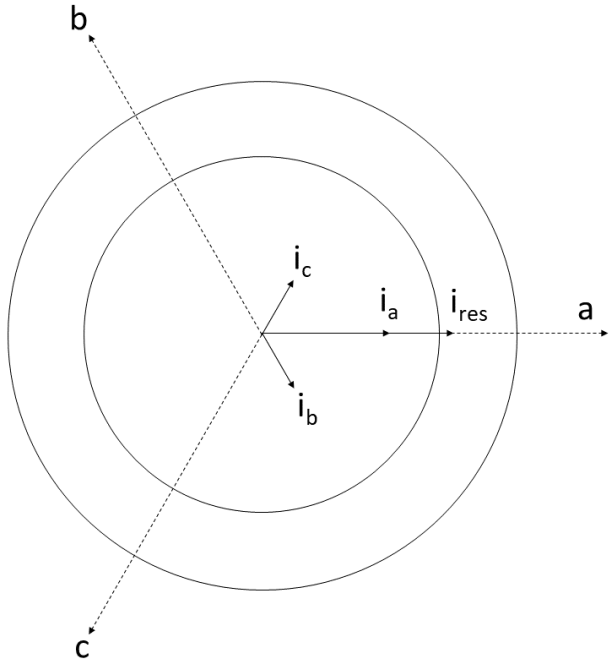


Figure 2.2: Resulting current space-vector when the phase a current is at its maximum value.

2.1.2 Dynamic Motor Model

The stator voltages of the three-phase permanent magnet synchronous machine can be expressed using space vectors as

$$\underline{U}_s^s = R_s \underline{I}_s^s + \frac{d\underline{\Psi}_s^s}{dt} \tag{2.4}$$

where

$$\underline{U}_s^s = \begin{bmatrix} U_a & U_b & U_c \end{bmatrix}^T \tag{2.5}$$

$$\underline{I}_s^s = \begin{bmatrix} I_a & I_b & I_c \end{bmatrix}^T \tag{2.6}$$

Here, the stator flux linkage vector is given as the product of the inductance matrix and the current vector plus the permanent magnet flux vector:

$$\underline{\Psi}_s^s = \begin{bmatrix} \Psi_a & \Psi_b & \Psi_c \end{bmatrix}^T = \boldsymbol{L}_s^s \underline{I}_s^s + \underline{\Psi}_m^s \tag{2.7}$$

Both the inductance matrix, $\boldsymbol{L}_s^s$, and the permanent magnet flux linkage are dependent on the rotor position. The inductance matrix is presented in Appendix A, while the permanent magnet flux linkage vector is given as

$$\underline{\Psi}_m^s = \Psi_m \begin{bmatrix} \cos(\theta) \\ \cos(\theta - \frac{2\pi}{3}) \\ cos(\theta + \frac{2\pi}{3}) \end{bmatrix} \tag{2.8}$$

Coordinate Transformations

In this thesis, two coordinate transformations will be used to represent the machine model given by Equation 2.4 in two alternative coordinate systems. In the first one, the stator variables are represented in the $\alpha\beta$ coordinate system, where the α-axis is aligned with the a-axis of the stator and the β-axis leads the α-axis with $90°$. Assuming that the three-phase system is balanced, the machine variables can be represented in the $\alpha\beta$ coordinate system by pre-multiplying the vector variable in question with the transformation matrix $\boldsymbol{T}_s^s$:

$$\boldsymbol{T}_s^s = \frac{2}{3} \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix} \tag{2.9}$$

As an example, the stator current space vector in the stationary, stator-oriented $\alpha\beta$ coordinate system, $\underline{I}_{s,\alpha\beta}^s$, can be obtained using

$$\underline{I}_{s,\alpha\beta}^s = \boldsymbol{T}_s^s \underline{I}_{s,abc}^s \tag{2.10}$$

The factor $\frac{2}{3}$ in Equation 2.9 ensures that the transformation is amplitude invariant, meaning that the length of the space-vector in the $\alpha\beta$ coordinate system will be equal to the amplitude of the stator phase current. Alternatively, the $\frac{2}{3}$ factor could be replaced with a factor equal to $\sqrt{\frac{2}{3}}$ if a power invariant transformation is preferred.

The second coordinate transformation that will be used in this thesis, and arguably the most important, is referred to as the dq transformation. In the dq coordinate system, the d-axis is aligned with the magnetic axis of the rotor of machine, with the q-axis $90°$ ahead. The dq reference-frame will hence rotate with a frequency equal to the electrical frequency of the PMSM rotor. Assuming a perfectly balanced three-phase system once again, the vector variables of the machine can be represented in the dq coordinate system using the following transformation matrix:

$$\boldsymbol{T}_s^r = \frac{2}{3} \begin{bmatrix} \cos(\theta) & \cos(\theta - \frac{2\pi}{3}) & \cos(\theta + \frac{2\pi}{3}) \\ -\sin(\theta) & -\sin(\theta - \frac{2\pi}{3}) & -\sin(\theta + \frac{2\pi}{3}) \end{bmatrix} \tag{2.11}$$

The stator current space vector represented in the rotating, rotor-oriented dq coordinate system, $\underline{I}_s^r$, is now given by

$$\underline{I}_s^r = \boldsymbol{T}_s^r \underline{I}_s^s \tag{2.12}$$

Figure 2.3 illustrates the relationship between the three different coordinate systems in the case of a two-pole PMSM with interior-mounted magnets. The $\alpha\beta$-coordinate system is represented by the red axes, while the dq-coordinate system is represented by the blue axes. The α-axis is seen to correspond with the a-axis of the stator, while the d-axis in the rotor-oriented coordinate systems coincides with the magnetic axis of the rotor. The electrical rotor position, θ, is equal to the displacement of the d-axis with respect to the a-axis.

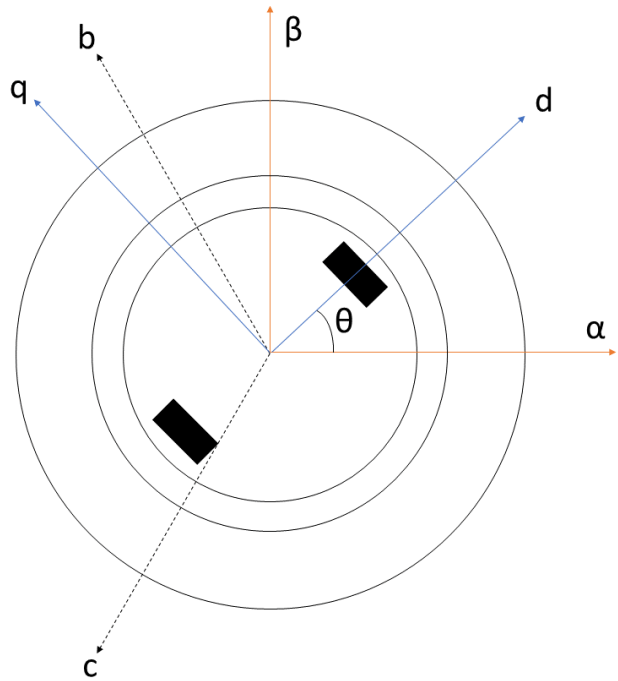


Figure 2.3: Relationship between the different coordinate systems in a two-pole IPMSM.

The Transformed, per unit Scaled Motor Model

If the motor model given by Equation 2.4 is transformed to the dq coordinate system, a simplified motor model is obtained. The PMSM can now be modelled as a two-phase motor with DC stator quantities in steady state and a simplified motor inductance matrix that has become independent of the rotor position. To ensure that any overloading of the

machine can be easily detected, as well as to make it easier to compare individual machines and to reuse motor control software, the motor model quantities are scaled by their base values to obtain an equivalent per unit model. The dynamic per unit model of the IPMSM in the rotor-oriented dq coordinate system that will be used to implement vector control of the motor is given by

$$\underline{u}_s^r = r_s \underline{i}_s^r + \frac{1}{\omega_n} \frac{d\underline{\psi}_s^r}{dt} + n\boldsymbol{j}\underline{\psi}_s^r \tag{2.13}$$

where

$$\underline{u}_s^r = \begin{bmatrix} u_d & u_q \end{bmatrix}^T \tag{2.14}$$

$$\underline{i}_s^r = \begin{bmatrix} i_d & i_q \end{bmatrix}^T \tag{2.15}$$

$$\boldsymbol{j} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \tag{2.16}$$

The stator flux linkage vector is now given by

$$\underline{\psi}_s^r = \begin{bmatrix} \Psi_d & \Psi_q \end{bmatrix}^T = \boldsymbol{x}_s^r \underline{i}_s^r + \underline{\psi}_m^r \tag{2.17}$$

where

$$\underline{\boldsymbol{x}}_s^r = \begin{bmatrix} x_d & 0 \\ 0 & x_q \end{bmatrix} \tag{2.18}$$

$$\underline{\psi}_m^r = \begin{bmatrix} \psi_m & 0 \end{bmatrix}^T \tag{2.19}$$

In Equation 2.13, n is the per unit electrical speed and ω_n is the nominal angular frequency.

The torque output of the machine can be found using

$$\tau = \psi_d i_q - \psi_q i_d = \psi_m i_q - (x_q - x_d) i_d i_q \tag{2.20}$$

2.2 Control Principles

In this section, an overview over some commonly used methods for control of AC machines will be presented, along with a more detailed discussion of the Indirect Torque Control (ITC) method that will be employed in this thesis.

2.2.1 Existing Methods for Variable-Frequency Drives

The control of variable-frequency drives can generally be divided into two main categories. These are scalar control methods, in which scalar motor quantities are being controlled, and vector control methods, that are based on controlling both the magnitude and angle of space-vector motor quantities. The chart in Figure 2.4 gives an overview over a few commonly used control methods for variable-frequency motor drives.

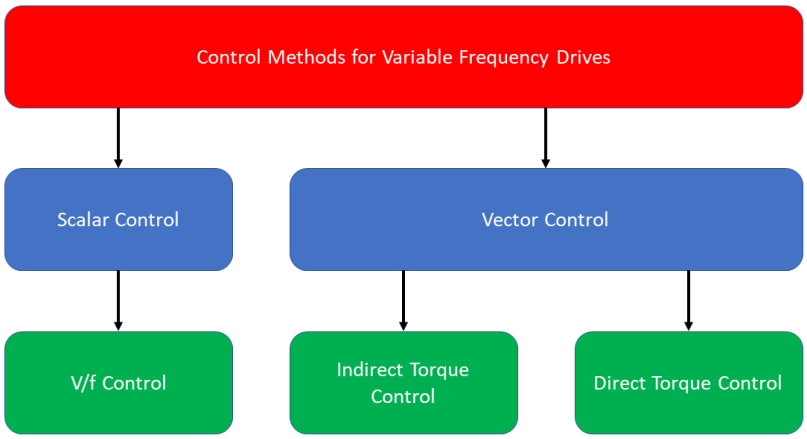


Figure 2.4: Classification of a few commonly used control methods for variable frequency drives.

The most common scalar control method is the volts-per-hertz (V/f) control method. Here, the ratio between the magnitudes of the stator voltage and the frequency of the machine is kept constant, resulting in the stator flux linkage, and hence motor torque, being maintained at rated value across the whole speed range. This method is mostly used to control induction motors, but has also been shown to be effective in conventional PMSM drives [12, 46]. The V/f method is an open-loop control method, which makes high-performance control difficult. It is however very simple, making implementation uncomplicated and control hardware requirements low.

With the increasing computational power of microprocessors, vector control methods have become dominant in variable-frequency drives applications. In the 1960s, the Indirect Torque Control (ITC) method, often referred to as vector control or field-oriented control, was introduced as a result of the desire to control three-phase machines in a way similar to the DC machine. In the DC machine, the flux and the torque can be controlled inde-

pendently by changing the field and the armature current, respectively. In [5], Blaschke suggested a control method based on the transformed model of the synchronous machine that was presented in Section 2.1.2, in which the d-axis of the the $d-q$ reference frame is aligned with the rotor magnetic axis. Using this approach, the stator current space vector was decomposed into a field-producing component, i_d, and a torque-producing component, i_q, that corresponded with the field and armature current of the DC machine, respectively. By changing the d and q axis currents of the machine, the output torque of the machine could be controlled indirectly. The ITC method will be employed in this theses, and a more thorough discussion of this method is therefore presented in Section 2.2.2.

In the mid-1980s, a new method for controlling AC machines was developed. In [48], Takahashi and Noguchi proposed a control method in which the torque and stator flux of the machine were controlled directly using hysteresis controllers. This method was later termed the Direct Torque Control (DTC) method. Almost simultaneously, Depenbrock presented a similar method in [13]. Depenbrock called his method Direct Self Control (DSC), and today, the DTC control method is usually credited to all three individuals. The DTC used the voltage model that will be presented in Section 3.1.1 for estimation of the stator flux, while the current model that is discussed in Section 3.1.2 was used for correction of the voltage model flux estimate. Once the stator flux was estimated, the torque of the machine could be calculated as the cross product of the estimated flux and the stator current. The estimated stator flux and torque were then compared to their reference values using hysteresis controllers, and the outputs of the controllers were used to obtain a voltage reference vector that determined the switching order of the converter switches.

2.2.2 Indirect Torque Control of the PMSM

As mentioned in the previous section, the goal of the Indirect Torque Control method was to enable control of AC machines in a way that resembled the control of DC machines. In the DC machine, the electromagnetic torque, τ, is proportional to the product of the field winding flux linkage, ψ_f, and the armature current, i_a, as shown in Equation 2.21. If the field current, and hence also field flux, is held constant, the electromagnetic torque can easily be controlled by changing the armature current of the machine.

$$\tau \propto \psi_f i_a \tag{2.21}$$

A similar expression for the electromagnetic torque output of the PMSM was presented in Equation 2.20, and is repeated below:

$$\tau = (\psi_m - (x_q - x_d)i_d)i_q \tag{2.22}$$

By comparing Equation 2.21 and Equation 2.22, it can be seen that the PMSM torque equation is analogous to the DC machine torque equation. Here, the field flux given by $\psi_m - (x_q - x_d)i_d$ corresponds to ψ_f in the DC machine, and can be controlled by regulating the d-axis current. If the value of this term is held constant, the torque output of the machine can easily be controlled by regulating i_q, which corresponds to i_a in the DC machine.

An overview of the indirect torque-controlled PMSM drive is displayed in Figure 2.5. The d- and q-axis reference currents are calculated based on the torque reference and compared with the measured stator currents. The difference between reference values and measured values is passed to a PI controller that is tuned using modulus optimum, as explained in [37]. A feed-forward decoupling term is added to the output of the PI controller in order to eliminate the d- and q-axes cross-coupling and obtain the converter voltage references in the rotor oriented reference frame. Next, voltage references in the stator-oriented reference frame are obtained by performing an $\alpha\beta$-transformation. These are passed to the pulse-width modulator that generates the gating signals for the voltage source converter that feeds the stator windings of the PMSM. The $\alpha\beta$ and dq transformations require the rotor position, θ, and the decoupling term calculator requires the rotor speed, n. Methods for estimation of these quantities will be presented later in the thesis.

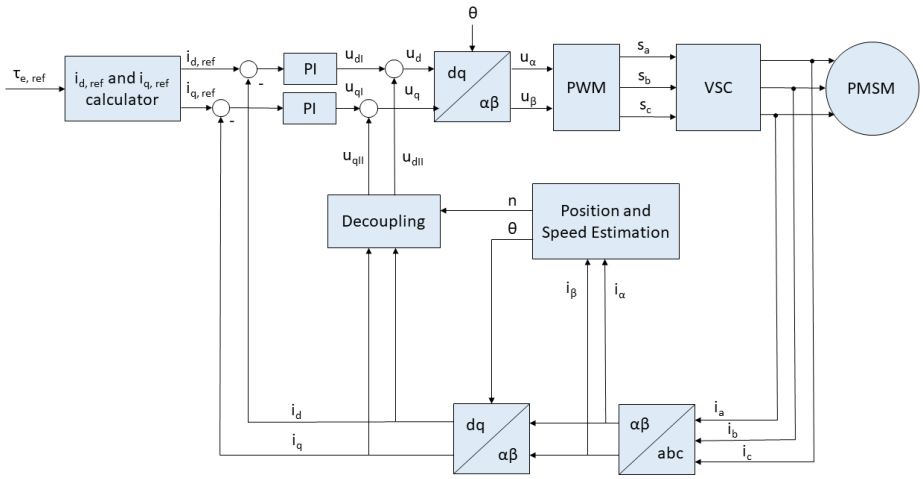


Figure 2.5: Block diagram showing the general structure of the torque-controlled PMSM drive.

Maximum-Torque-Per-Ampere Control

A common ITC strategy is to orient the stator current vector in a way that maximizes the output torque of the machine for the given current magnitude. If the PMSM is non-salient, the reluctance along the flux paths of the d and q axis are equal, resulting in equal d and q axis inductances. The torque equation of the machine then reduces to

$$\tau = \psi_m i_q \tag{2.23}$$

From Equation 2.23, it can be seen that the output torque is directly proportional to the q-axis component of the stator current, and the maximum amount of torque for a given current magnitude is therefore generated if the current space vector is aligned with the q-axis of the dq reference system. This is done by applying the following current references:

$$i_{d,ref} = 0 \tag{2.24}$$

$$i_{q,ref} = \frac{\tau_{e,ref}}{\psi_m} \tag{2.25}$$

Here, $\tau_{e,ref}$ is the torque reference. If the PMSM has interior-mounted permanent magnets, however, the output torque will be given by Equation 2.22. Since magnets have approximately the same relative permeability as air, the inductance along the d-axis will be lower than along the q-axis of the motor. From Equation 2.22 it can be seen that if a negative d axis current is applied, this difference in inductance can be used to generate an additional torque component, usually referred to as reluctance torque. For PMSMs with interior-mounted magnets, it is therefore desirable to operate the machine with a negative d-axis current in order to utilize this reluctance torque. As there is an infinite number of d- and q-axis currents that will produce the same torque, finding the combination that gives the maximum torque output per ampere for this machine type becomes an optimization problem. This problem was first analyzed in [24], where Jahns et al. developed analytical expressions that can be used to obtain the optimal combination of d- and q-axis current. The current references in this thesis will be calculated based on the expressions in Equation 2.26 and Equation 2.27 [45].

$$i_{d,ref} = \frac{\frac{\psi_m}{3} - \sqrt[3]{(\frac{\psi_m}{3})^3 + \frac{(x_q - x_d)^2 \tau_{e,ref}^2}{3\psi_m}}}{x_q - x_d} \tag{2.26}$$

$$i_{q,ref} = \frac{\tau_{e,ref}}{\psi_m - (x_q - x_d) i_d} \tag{2.27}$$

With the maximum torque per ampere control method properly implemented, the stator

current that is necessary for a given torque output is minimized. As the copper losses of the machine are proportional to the square of the stator current, this control strategy minimizes copper losses and maximizes the efficiency of the machine. Additionally, by reducing the stator current magnitude, the size of the converter can be reduced. Hence, the cheapest converter for a given torque is obtained when maximum-torque-per-ampere control is implemented.

Field-Weakening

When the motor operates at rated speed, the back-EMF of the machine, given as the product of stator flux linkage and motor speed, will have become equal to the rated stator voltage. The rated stator voltage corresponds to the maximum output voltage of the converter, which means that the converter has saturated and is not able to increase its output voltage further. At this operating point, no current is flowing into the stator windings, making acceleration beyond the rated speed impossible. To enable operation above speeds of 1 per unit, one must therefore start to decrease the stator flux as the motor approaches rated speed. This way, the back-EMF is not exceeding the rated stator voltage, allowing the machine to accelerate further.

The per unit stator flux linkage in the dq-coordinate system is given by

$$\psi_d = x_d i_d + \psi_m \tag{2.28a}$$

$$\psi_q = x_q i_q \tag{2.28b}$$

As previously discussed, it is desirable to maintain the q-axis current as large as possible, and field-weakening should therefore be performed by applying a negative d-axis current that counteracts the magnetic field from the permanent magnets. From the equations above, it is however obvious that the stator flux linkage reduction that can be obtained depends on the magnitude of the d-axis reactance. If a large field-weakening region is required, the inductance along the d-axis flux path of the machine must therefore be large. For a PMSM with saliency, this will reduce the reluctance torque generation, and hence come at the cost of the torque production capability of the machine. For low-inductance machines, a large negative d-axis current must be applied in order to achieve a notable stator flux reduction. The cost of applying a d-axis current is a corresponding reduction of the q-axis component, as a constant current space vector magnitude must be maintained. Due to the magnitude of the d-axis current that must be applied, field-weakening in low-inductance machines will therefore lead to a large reduction in the q-axis current and hence also the torque output of the machine. Field-weakening in such machines might therefore not be justifiable.

2.3 Digital Control Implementation and Verification

The sensorless control algorithms that are developed in this thesis will be implemented and verified using the PicoZed 7030 system-on-module (SoM) from Avnet. The SoM is based on the Zynq 7030 system on chip (SoC), that contains two ARM Cortex-A9 central processing units (CPUs) and one field-programmable gate array (FPGA). One of the on-chip processors runs a Linux program that is used for monitoring and programming of the remaining processor and the FPGA, while the second on-chip processor can be programmed with the C++ programming language, using the Xilinx Software Development Kit (XSKD). This is where the control strategies and algorithms that are developed in this thesis will be implemented.

The hardware of a complete permanent magnet synchronous motor drive has been programmed on the FPGA on the SoC to produce an embedded real-time simulator (ERTS) that is capable of emulating the PMSM drive. The control software that is implemented in the processor is intented to drive either the emulated drive in the ERTS or a physical PMSM drive. In this thesis, the control software that is developed will be verified using the ERTS. The development of the ERTS and the processor control algorithms has been an ongoing effort in the Power Electronic Systems and Components (PESC) research group in the Department of Electric Power Engineering over the last few years, and at least two master's theses have been dedicated to this purpose [16, 35]. A detailed description of the control platform and an analysis of the performance of the ERTS is provided by Perera et al. in [45]. The embedded controller is pictured in Figure 2.8.

2.3.1 Control Software Implementation: CPU Programming

The on-chip processor of the control platform contains the majority of the control software that is necessary to operate a PMSM drive when this thesis is initiated, with functionality such as per unit scaling, reference frame transformations, maximum-torque-per-ampere and field-weakening strategies and indirect torque control algorithms already implemented [45]. However, the control software relies on the measuring of the rotor position and speed, and is as such not capable of driving the PMSM without the use of a position-sensor. As stated in Chapter 1, the main objective of this thesis is to develop sensorless control subroutines which shall be added to the existing control software library that is programmed on the on-chip processor.

The control software is written using the C++ programming language, which due to its

object-oriented nature ensures scalability, flexibility and structure of the control software. The control software is structured in different layers, as shown in Figure 2.6, where information is passed only between neighbouring layers. The flux modelling and position and speed estimation software that this thesis contributes with will be written in the drive software layer (layer 2).

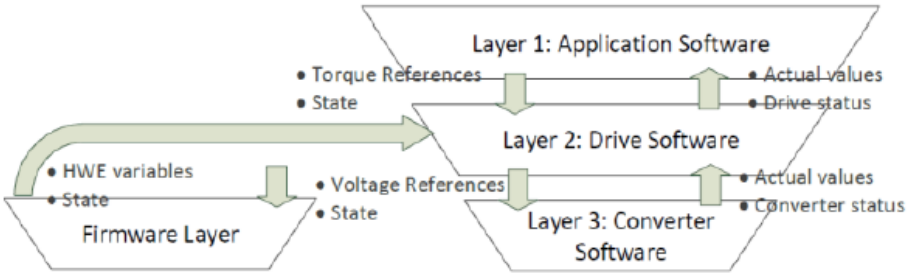


Figure 2.6: The structure of the PMSM drive control software. Retrieved from [45].

2.3.2 Control Software Verification: The Embedded Real-Time Simulator (ERTS)

The control software development process in this thesis will rely on the use of the ERTS for verification and validation. In the past, motor control strategies have usually been developed and tested on a physical motor drive in a lab environment, but the recent trend is to perform software testing and verification using simulated motor drive models [14, 1]. This is referred to as software-in-the-loop (SIL) testing, and requires accurate emulation of the physical behaviour of the motor drive.

The ERTS that is used in this thesis is implemented through the integration of a series of building blocks that are programmed on the FPGA and emulate the different hardware components of the drive, such as the voltage-source converter, the PMSM, the mechanical load etc. Due to the large processing capability of the FPGA, the motor model equations can be solved during time-steps that is equal to the real-world clock, yielding a simulator that is capable of emulating the motor drive in real-time [39]. This means that a simulation with a duration of 1 second can be performed over a time period of 1 second, and simulation results can be obtained much faster than if conventional simulation tools such as MATLAB Simulink are used. The emulator is interfaced with the CPU on the SoC, which means that the control software that is implemented here can be used to drive the ERTS if a physical motor drive is not available. There are hence two main advantages of using the ERTS in the control algorithm development process: Control strategies can be tested

through real-time simulation, making it a significantly less time-consuming effort than if conventional simulation tools are used, and the ERTS enables software-in-the-loop testing, which makes a physical motor drive redundant in the control software development process.

A schematic of the general structure of the control platform is displayed in Figure 2.7. Here, it is shown how the processing system can drive either the ERTS or a physical motor drive. The general control functionality, as well as the flux models and position and speed estimators that will be developed in this thesis, runs on the CPU in the SoC. The ERTS and the pulse-width-modulator (PWM) is implemented on the FPGA.

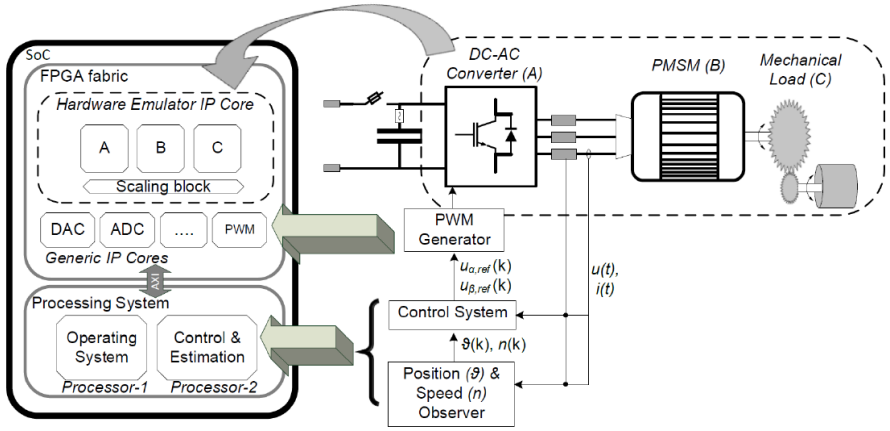


Figure 2.7: The general structure of the SoC-based embedded controller. Retrieved from [45].

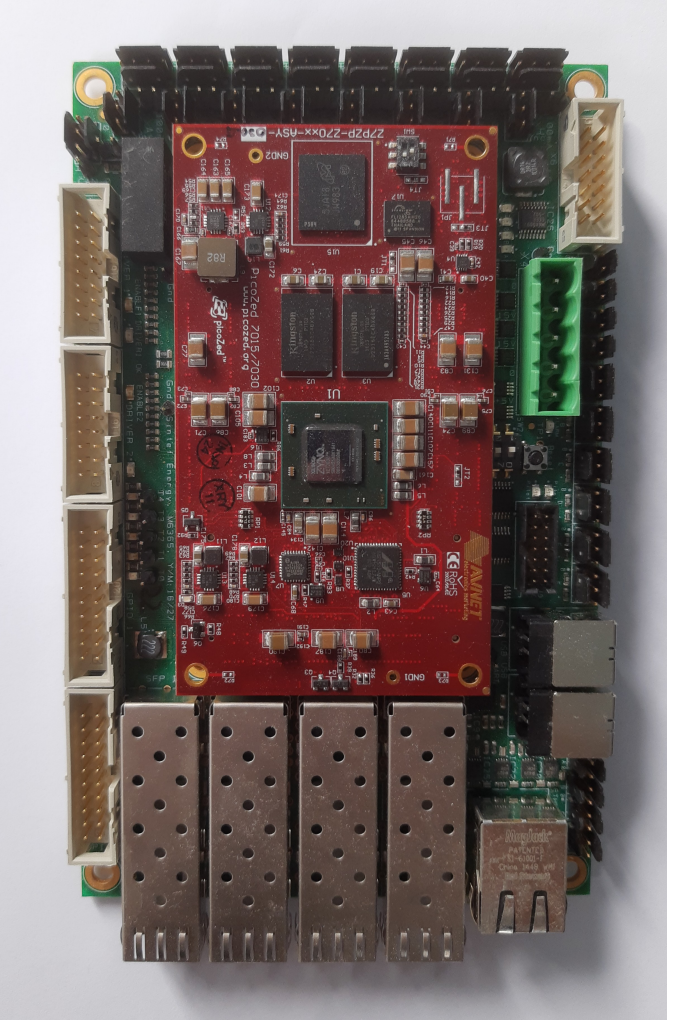


Figure 2.8: The embedded controller.

Chapter 3

Position-Sensorless Indirect Torque Control of the PMSM

Indirect torque-control of electric machines is accomplished by controlling the stator currents in the rotor-oriented reference frame, and knowledge about the rotor position is therefore necessary for its implementation. Traditionally, the rotor position has been obtained using position sensors such as encoders or resolvers. However, the current trend in electric machine drives is to avoid any position or speed sensors by obtaining these quantities through estimation. By removing position and speed sensors, the cost and complexity of the drive can be greatly reduced, resulting in increased reliability and lower maintenance requirements. Additional benefits are reduced weight and size.

With sensors removed, the main obstacle in any sensorless motor drive is obtaining sufficiently accurate estimates of the rotor position and speed. Over the years, various estimation methods have been proposed. Generally, these methods fall within one of of the following categories: Fundamental-excitation based methods, that relies on the back-EMF of the machine for position estimation, and saliency tracking methods, that work by injecting high-frequency voltage components in the stator and extracting the rotor position from the response in the stator current. In this thesis, fundamental-excitation based sensorless control methods will be considered. These estimate the rotor position and speed through a series of two steps: First, the stator flux linkage space vector is estimated using the back-EMF of the machine. Then, the flux estimate is used to obtain estimates of the rotor position.

3.1 Stator Flux Linkage Estimation

Rotor position estimation in fundamental-excitation based sensorless motor drives starts with obtaining the stator flux linkage of the machine, and accurate models for flux estimation thus becomes essential in any high-performance motor drive. There are two main models that can be used to identify the stator flux vector based on the PMSM motor model. These are referred to as the voltage model and the current model, and will be presented in this section.

3.1.1 The Voltage Model

The voltage model for flux estimation is based on the stator voltage balance equations that was presented in Section 2.1. By re-writing the per unit equivalent of Equation 2.4, the back-EMF of the motor in the stationary stator-oriented reference can be written as

$$\frac{d\underline{\psi}_s^s}{dt} = \underline{u}_s^s - r_s \underline{i}_s^s \tag{3.1}$$

The voltage model provides an estimate for the stator flux linkage in the stator-oriented $\alpha\beta$ reference frame by integrating the stator back-EMF according to Faraday's law of induction, as shown in Equation 3.2

$$\underline{\psi}_s^s = \underline{\psi}_{s0}^s + \omega_n \int_0^t (\underline{u}_s^s - r_s \underline{i}_s^s) dt \tag{3.2}$$

Here, $\underline{\psi}_{s0}$ is the initial value of the stator flux. The discrete-time equivalent of Equation 3.2 is given by

$$\underline{\psi}_s^s[k] = \underline{\psi}_s^s[k-1] + \omega_n \cdot T_{samp} \cdot (\underline{u}_s^s[k] - r_s \underline{i}_s^s[k]) \tag{3.3}$$

Generally, the voltage model does not perform well at low speeds. When the speed is zero or close to zero, the back-EMF of the machine becomes very small, which will result in inaccurate flux estimates. While sensorless operation based on the voltage model is a viable option if the low-frequency range is exceeded rapidly, continuous operation at zero or close to zero speed is therefore impossible. The voltage model is thus best suited for flux estimation in the medium and high speed range. It will later be shown that the flux estimates might become inaccurate even for high frequencies if the inputs to the voltage model deviate from their ideal values, and methods that compensate these inaccuracies and stabilize the flux estimate are therefore always necessary in practical applications.

3.1.2 The Current Model

The second model for flux identification that will be considered in this thesis is referred to as the current model. In this model, the measured stator currents along with the machine inductances are used to provide estimates of the stator flux, as shown in Equation 3.4.

$$\underline{\psi}_s^r = \boldsymbol{x}_s^r \underline{i}_s^r + \underline{\psi}_m^r \tag{3.4}$$

The discrete-time equivalent of the current model becomes

$$\underline{\psi}_s^r[k] = \boldsymbol{x}_s^r \underline{i}_s^r[k] + \underline{\psi}_m^r \tag{3.5}$$

In contrary to the voltage model, flux estimation using the current model is done in the rotor-oriented dq reference frame. This means that the rotor position is required as an input to the model, and the current model alone can therefore not be used to estimate the rotor position during sensorless operation. Assuming that an accurate position measurement is available and that the stator inductances and permanent magnet flux linkage are known, the current model will produce an accurate flux estimate across the whole speed range. In this thesis, the current model will be used to initialize the voltage model flux estimate once the rotor of the machine has been rotated into a known initial position before start-up. It will also be showed that the current model can be used in tandem with the voltage model to produce a stable, closed-loop flux observer.

3.1.3 Steady-State Sensitivity of the Flux Models

The flux models that were presented in the preceding sections are highly sensitive to variations in the values of the motor parameters r_s, x_d, x_q and ψ_m. Although motor parameter values usually are provided by the manufacturer, these are only valid in certain operating points and should only be considered estimates. In practice the exact values of the motor parameters will depend on current operating conditions, such as loading, ambient temperature, frequency and magnetic saturation. Motor inductances are mainly dependent on magnetic saturation, and will hence be influenced by the motor loading. ψ_m and r_s are on the other hand to a greater extent temperature dependent, making them vulnerable to variations in loading as well as ambient temperature. [42, 43]

In addition to erroneously estimated motor parameters, inaccurate information about the stator voltage and current may also contribute to imperfect flux estimation. Usually, the stator currents of the motor are measured using current sensors, and in this thesis, it will be assumed that they are accurately measured at all times. The stator voltage, on the other hand, is often estimated. The most common way of doing so is to use the reference voltage

of the converter. If the non-linear voltage drops across the converter are modelled, they can be subtracted from the reference voltage and an estimate for the actual stator voltage can be obtained. This method will be analyzed in more detail in Section 3.3.

Analysis of how inaccuracies in motor parameter and voltage estimates influence the flux models that form the basis for position-sensorless control is important. To facilitate such sensitivity analyses, the MATLAB scripts that are presented in Appendix D have been developed, inspired by a method presented by Bolstad in [9]. The scripts calculate and compare actual stator flux with estimated flux for speeds and torque references between -1 and 1 per unit, and can be used to investigate how erroneously estimated motor parameters and stator voltage may influence the flux models that have been presented. In the following sections, the impact of an underestimated stator resistance and a DC offset in the stator voltage on the voltage model flux estimate during steady-state operation will be demonstrated and discussed. The voltage model is the most important flux model in this thesis, and it is thus natural to dedicate efforts to analysis of this flux model. A similar analysis is however also performed for the current model, and the results are presented in Appendix E.

In the remainder of this thesis, the hat symbol will be used to denote estimated quantities.

Voltage Model Sensitivity Analysis Method

Assuming that the stator currents are accurately measured, the sources of error in the flux estimate provided by the voltage model are erroneously estimated stator resistance, $\hat{r}_s$, and stator voltage, $\underline{\hat{u}}_s^s$. The voltage model can be represented in the frequency domain as

$$jn\underline{\hat{\psi}}_s^s = \underline{\hat{u}}_s^s - \hat{r}_s\underline{i}_s^s \tag{3.6}$$

The magnitude of the flux estimate can then be obtained using

$$\underline{\hat{\psi}}_s^s = \frac{\underline{\hat{u}}_s^s - \hat{r}_s\underline{i}_s^s}{jn} \tag{3.7}$$

With the stator currents given by Equation 2.26 and Equation 2.27, the above expression can be used to obtain the stator flux. The errors in the flux estimate amplitude and angle are calculated as $|\underline{\psi}_s^s| - |\underline{\hat{\psi}}_s^s|$ and $\angle\underline{\psi}_s^s - \angle\underline{\hat{\psi}}_s^s$, respectively. $\underline{\psi}_s^s$ will be obtained by assuming accurate estimation of $\hat{r}_s$ and $\underline{\hat{u}}_s^s$, while $\underline{\hat{\psi}}_s^s$ is found by assuming erroneous estimation of

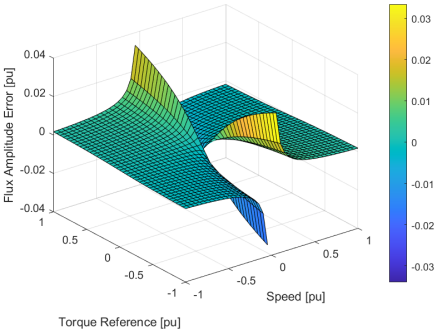
either $\hat{r}_s$ or $\underline{\hat{u}}_s^s$.

Impact of an Underestimated $\hat{r}_s$

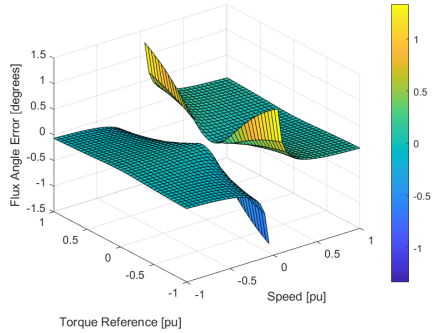
To demonstrate the impact of an erroneously estimated stator resistance, the resistance is assumed to be underestimated with 20%, such that $\hat{r}_s = 0.8r_s$. This could correspond to a scenario in which the stator windings for some reason are experiencing high temperatures, such that the actual resistance is higher than the resistance that is given by the manufacturer. This could be a result of high ambient temperatures, loading or frequency.

Figure 3.1 shows the error in the magnitude and angle of the flux estimate due to the underestimated stator resistance during steady-state. Two important observations can be made: The error in the flux estimate due to an erroneous $\hat{r}_s$ increases with the magnitude of the load torque and it also increases as the speed approaches zero. This can be understood by considering Equation 3.7. A large torque reference means that the stator current magnitude becomes large, and erroneous stator resistance will as such have a larger impact on the flux estimate, since the current and resistance are multiplied. The polarity of the magnitude error depends on whether the machine operates as generator or motor. In motor operation, the error between the real flux and the flux estimate is negative. This is because the stator voltage and the stator current will have the same polarity, and the underestimated resistance will hence lead to the flux magnitude being estimated too large. Similarly, when the machine operates as a generator and the voltage and current have different polarities, the underestimated resistance will result in a flux estimate that is smaller than its real value, leading to a positive flux estimate error. When the speed is low, the stator voltage magnitude is small and the relative importance of the stator resistance in the flux estimate is larger. Additionally, the denominator in Equation 3.7 will then also be small, and the impact of the inaccurate resistance estimate will be amplified.

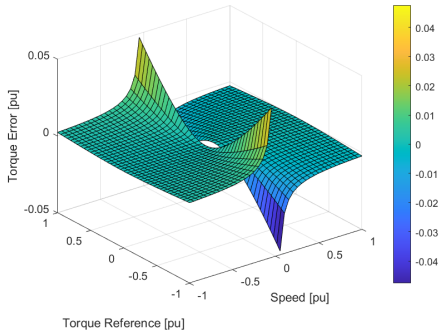
An underestimated resistance $\hat{r}_s = 0.8r_s$ is seen to produce a steady-state amplitude error of around 0.03 per unit and an angle error in excess of 1 degree. This results in an error in the torque output of the machine that approaches 0.05 per unit when the torque reference becomes large and the speed becomes low. The torque error is positive for negative speeds, meaning that the actual motor torque is less than in the ideal case. For positive speeds, the error is negative, which means that the motor will output a greater torque than what it would if the resistance was accurately estimated.



(a) Amplitude error of flux estimate space vector.



(b) Angle error of flux estimate space vector.



(c) Error in the torque output of the motor.

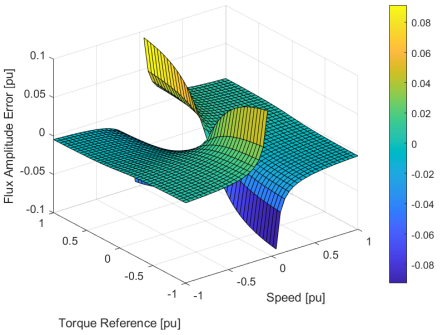
Figure 3.1: Error in flux estimate magnitude and angle and output torque due to an underestimated stator resistance.

Impact of a DC offset in $\underline{\hat{u}}_s^s$

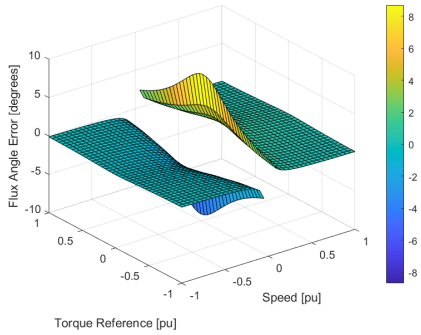
To illustrate the effect of an erroneous stator voltage estimation, the α-axis stator voltage estimate is assumed to contain a DC offset equal to 0.005 per unit. This could for example be a result of inaccurate modelling of the inverter non-linearities. The resulting error in the magnitude and angle of the stator flux estimate is displayed in Figure 3.2.

From the plots below, it is concluded that the impact of the DC offset in the voltage estimate is largest at low speed when the torque reference is large. Low speed means that the stator voltage is low, and that the relative magnitude of the DC offset will be larger. Additionally, a low value of the denominator in Equation 3.2 will amplify the error in the voltage estimate. A large torque reference and the following increase in stator current magnitude will result in the nominator in Equation 3.7 becoming small, such that the impact of the DC offset in the stator voltage becomes prominent. The polarity of the amplitude error

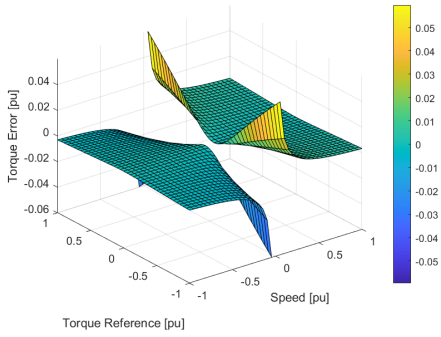
is dependent on whether the machine operates as a motor or a generator. During motor operation, the DC offset will lead to the flux being estimated too small, such that the flux estimate magnitude error becomes positive. Similarly, during generator operation, the DC offset makes the flux estimate magnitude become larger than the real flux, such that the estimate error becomes negative. The voltage model seems to be highly sensitive to even small offsets in the stator voltage. In this case, a maximum amplitude error of around 0.1 per unit and a angle error of about 8 degrees is observed. The maximum torque error occurs during low-speed and high-torque operation, and the error magnitude appears to be in the range of 0.05 per unit. The torque error is negative for negative speeds, which means that the torque output of the machine becomes greater than what it would be in the case of a perfectly estimated stator voltage. For positive speeds, the voltage estimate error results in a reduction of the motor torque output, and the torque error becomes positive.



(a) Amplitude error of flux estimate space vector.



(b) Angle error of flux estimate space vector.



(c) Error in the torque output of the motor.

Figure 3.2: Error in flux estimate magnitude and angle and output torque due to a DC offset in the stator voltage estimate.

3.2 Rotor Position and Speed Estimation

The operation of the sensorless motor drive is divided into two main categories: Finding initial position and "operation enabled". During the former, the initial position of the rotor is obtained and the stator flux estimate is initialized using the current model. In the latter, the rotor position is estimated in real time using the flux estimate from the voltage model.

3.2.1 The Initialization Procedure

In this project, freedom is given to rotate the rotor into a known initial position before the motor is started. This will be done by applying a DC current in the a-axis of the motor during an initialization period of 3 seconds. During this period, the interaction between the flux from the stator windings and the permanent magnet flux linkage will cause the magnetic axis of the rotor (i.e. the d-axis) to align with the a-axis of the stator. It should be noted that this method is not an acceptable option in all cases, as the direction of motion of the rotor will be unpredictable, which may have detrimental consequences in some applications.

Once the initialization procedure is performed, the rotor angle will be 0 degrees and the current model can be used to obtain the initial value of the stator flux estimate, $\underline{\psi}^s_{s0}$. The initial value produced by the current model is passed to the voltage model when the motor is started.

3.2.2 Operation Enabled: The Active Flux Concept

Once the initialization routine is completed, the motor drive enters the "operation enabled" state. Here, the drive will rely on the voltage model flux estimate for position estimation. Once the stator flux linkage is estimated, an estimate for the rotor position and speed can be obtained by introducing the active flux concept that transforms all salient-pole machines into fictitious, non-salient machines [40, 8, 7]. The active, or torque-producing, flux vector is the flux that multiplies the q-axis current in the rotor-oriented torque expression in Equation 2.22. The active flux vector in the rotor-oriented reference frame, Ψ^r_T, is given by

$$\underline{\psi}^r_T = \underline{\psi}^r_m + (x_d - x_q)i_d \tag{3.8}$$

Equation 3.8 can be further modified, and it can be shown that an expression for the active flux vector that is valid in any reference frame is given by

$$\underline{\psi}_T = \underline{\psi}_s - x_q\underline{i}_s \tag{3.9}$$

When the estimate for the stator flux linkage is found, an estimate for the active flux vector can be obtained by using Equation 3.9. From Equation 3.8, it is concluded that $\underline{\Psi}_T$ points in the same direction as the permanent magnet flux linkage and hence the rotor. The angle of $\underline{\Psi}_T$ with respect to the phase a axis thus corresponds to the angle of the rotor. This is can be visualized by considering Figure 3.3.

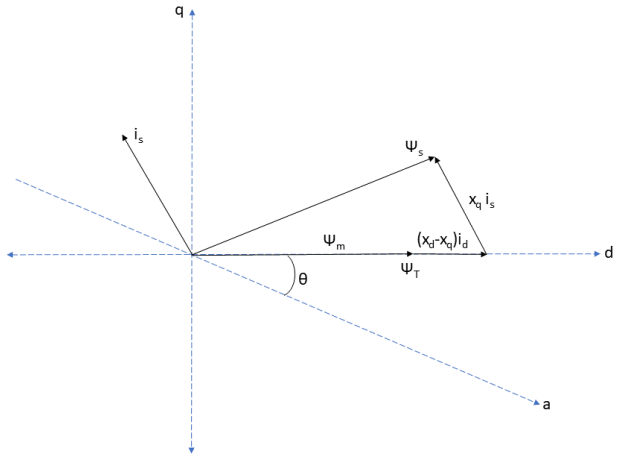


Figure 3.3: The active flux space vector. Reproduced from [40].

An estimate for the active flux vector in stator coordinates, $\underline{\hat{\psi}}_T^s$, can be obtained using the voltage model flux estimate, $\underline{\hat{\psi}}_{s,u}^s$, as shown in Equation 3.10.

$$\underline{\hat{\psi}}_T^s = \underline{\hat{\psi}}_{s,u}^s - x_q \underline{i}_s^s \tag{3.10}$$

The angle of $\underline{\hat{\psi}}_s^T$ is equal to the rotor angle position, θ, and can be calculated by using the inverse tangent function, as shown in Equation 3.11.

$$\hat{\theta}[k] = \arctan\left(\frac{\hat{\psi}_{T,\beta}[k]}{\hat{\psi}_{T,\alpha}[k]}\right) \tag{3.11}$$

Here, $\psi_{T,\alpha}$ and $\psi_{T,\beta}$ are the α and β components of the active flux vector, respectively. Once an estimate for θ is obtained, an approximate value for the per unit rotor speed can be found as the time derivative of the position estimate as shown in Equation 3.12.

$$\hat{n}[k] = \frac{\hat{\theta}[k] - \hat{\theta}[k-1]}{T_{samp}} \tag{3.12}$$

An equivalent expression for the speed estimate is given by Equation 3.13 [40]. This is the expression that will be used for speed estimation in this thesis.

$$\hat{n}[k] = \frac{\hat{\psi}_{T,\alpha}[k-1]\hat{\psi}_{T,\beta}[k] - \hat{\psi}_{T,\beta}[k-1]\hat{\psi}_{T,\alpha}[k]}{T_{samp}(\hat{\psi}^2_{T,\alpha}[k] + \hat{\psi}^2_{T,\beta}[k])} \tag{3.13}$$

Without additional filtering, the speed estimate produced by the equation above may become very noisy, which may cause problems for the control system. A low-pass filter is therefore applied to the speed estimate in order to reduce the noise in the estimate, with the filter time constant being set equal to 0.005 seconds. The time constant value is obtained through trial and error, and the chosen value was the smallest value that resulted in adequate filtering of the speed estimate. As low-pass filtering will introduce a delay between the filtered speed estimate and the unfiltered estimate that increases for increasing filter time constant magnitude, the time constant value should be kept as small as possible. In [8], a value of similar magnitude was deemed appropriate, and thus supports the filter design that is chosen in this thesis.

3.3 On-Line Stator Voltage Estimation

To achieve further reductions in drive hardware complexity, cost, size and weight, as well as increased reliability, the stator voltage sensors can be removed, and the stator voltage can be estimated on-line. Such an approach will be taken in this thesis, and is presented in this section.

A common technique for stator voltage estimation is to use the reference voltages provided by the controllers as a means of obtaining the output voltage of the inverter. Ideally, the inverter's output voltage would be identical to the reference voltage of the converter. In reality, however, the non-ideal properties of the inverter will lead to discrepancies between the reference voltage and the actual stator voltage, as non-linear voltage drops across the converter produces distortion in its output voltage. Especially during low speed operation when the stator voltage is low, the voltage might become highly distorted.

If the non-linear voltage drops across the inverter are accurately modelled, they can be subtracted from the reference voltage in order to compensate for the voltage drop across the inverter and achieve a realistic estimate of the output voltage. This will make it possible to operate the motor drive without using voltage sensors. In the following sections, this approach is described. A thorough analysis of how on-line stator voltage estimation can be performed was conducted by Fjellanger in his master's thesis, and the interested

reader is referred to [16] for a more comprehensive discussion.

3.3.1 Non-Ideal Converters: Influence of Dead Time Effects

The three-phase, two-level converter that is used in the PMSM drive has three bridge legs, with each leg having two transistors and two free-wheeling diodes, as shown in Figure 3.4. Ideally, the transistors are switched such that when one of them is in its ON state, the other one is off. In practice, there will be a short period of time, referred to as the blanking time T_{BT}, in which both transistors in a converter leg are off simultaneously [34]. This is done to avoid "shoot-through", or short-circuiting of the converter leg, as the turn-on delay time ensures that the device is completely turned off and does not turn back on again when the complementary transistor is turned on. If shoot-through were to occur, it is likely that the switches would be destroyed by the high short-circuit current.

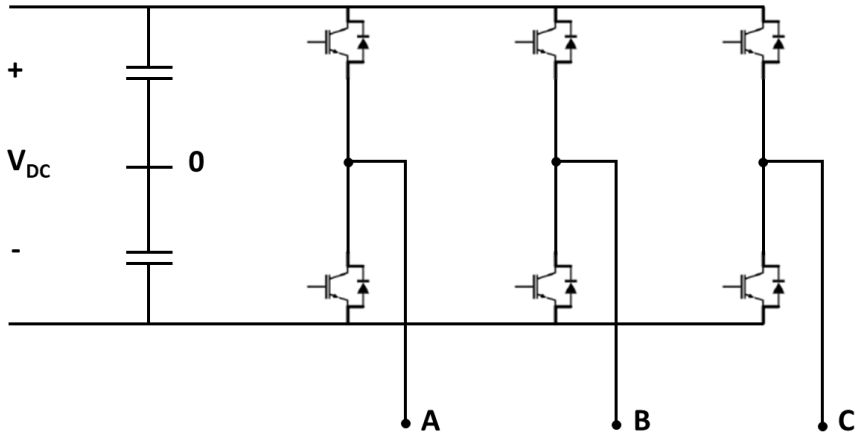


Figure 3.4: Generic three-phase, two-level voltage source converter. Reproduced from [34].

In addition to the blanking time, there will be a delay between the time of the turn on command and the instant when the device actually starts conducting. When a gate pulse is applied to the switch, the voltage across it will start to rise and the switch does not start to conduct before a certain voltage is reached. The period of time it takes for the voltage to increase above its threshold value and start conducting is referred to as the turn-on time of the transistor. Similarly, when the turn-off command is applied, there will be a period of time when the voltage across the device decreases, and when the voltage drops below a given value, the switch will stop conducting. This period of time is referred to as the

transistor's turn-off time. The turn-on and turn off-times of the transistor can generally be incorporated into the blanking time.

As described in the paragraph above, there will be a constant voltage drop across the transistor, u_{PN}, that is necessary for current to flow across the PN junction of the transistor. This also applies to the freewheeling diode. Additionally, both the transistor and the diode will cause an ohmic voltage drop due to their internal resistances, r_T. The combined effects of blanking time, turn-on and turn-off time and the voltage drops across the switches and diodes are usually referred to as the dead time effects of the converter. If these are accurately modelled, the real stator voltage can be estimated with high precision.

3.3.2 Estimating the Stator Voltage

Analytical expressions for the approximate value of the voltage drop across the converter as a result of the dead time effects in the converter, $\Delta u_{s,DT}$, were developed by Weichbold and von Raumer in [49] and are presented in Equation 3.14.

$$\Delta u_{s,DT} = sign(i_s) \left(u_{TD}(|i_s|) + \frac{T_d + T_{on} - T_{off}}{T_s} \cdot u_{DC} \right) \tag{3.14}$$

Here, u_{TD} is the averaged on-state voltage drop of transistor and diode, which is dependent on the current magnitude. T_{BT} is the blanking-time of the transistor, while T_{on} and T_{off} are the the turn-on and turn-off time, respectively. If the on-state voltage drop across the converter switches and diodes are neglected, a simple approximation of the voltage drops across the converter due to the dead-time effects is given by

$$\Delta u_{s,DT} = -sign(i_s) \frac{T_d + T_{on} - T_{off}}{T_s} \cdot u_{DC} \tag{3.15}$$

This is a reasonable approximation, since the magnitude of the DC link voltage is much larger than the on state voltage drop of the diode and transistor [16]. Once the value of Δu_{DT} is obtained for each phase, the output stator voltage, $\underline{\hat{u}}_s^s$ can be estimated using

$$\underline{\hat{u}}_s^s = \underline{u}_{s,ref}^s + \Delta \underline{u}_{s,DT}^s \tag{3.16}$$

where $\underline{u}_{s,ref}^s$ represents the reference voltage to the converter.

As seen from the expressions above, the voltage drop across the semiconductors is dependent on the direction of the current flowing out of the converter. When the current is positive, the value of $\Delta u_{s,DT}$ becomes negative, which leads to the output voltage becoming smaller than the reference voltage. When the current is negative, $\Delta u_{s,DT}$ becomes

positive and the output voltage becomes larger than the reference voltage. In Figure 3.5, the reference voltage of the converter, which corresponds to the ideal stator voltage, and the estimated output stator voltage, which is an approximation of the actual output voltage of the converter, are presented. The stator current is also included in the plot. Here, it has been assumed that the combined turn-on, turn-off and blanking time amount to 10% of the sampling period. This is an unrealistically high value, but allows for easy visualization of the dead-time effects on the output voltage of the converter. In the figure it can be seen that there is an instantaneous change in the output voltage when the current crosses the zero-line.

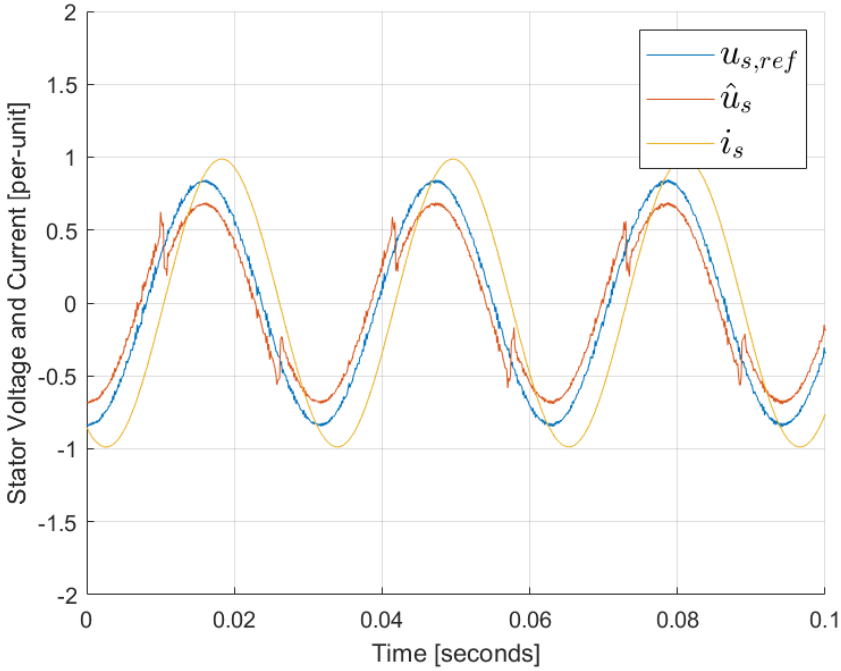


Figure 3.5: Ideal stator voltage, estimated stator voltage and stator current of phase a of the motor.

3.4 Sensorless Operation Based on the Voltage Model

In Section 3.1, the voltage model and the current model were presented, and it was shown how they can be used to estimate the stator flux linkage of the motor. Once a flux estimate is obtained, the rotor position of the machine can be estimated using the active flux concept that was presented in Section 3.2. Since the current model estimates the flux in the dq plane, it alone cannot be used for flux estimation in a position-sensorless motor drive, and

fundamental excitation-based methods for position-sensorless operation therefore rely on the use of the voltage model. The voltage model was presented in Equation 3.2, and is repeated below, now using the hat symbol to denote estimated quantities.

$$\underline{\hat{\psi}}_s^s = \underline{\hat{\psi}}_{s0}^s + \omega_n \int_0^t (\underline{\hat{u}}_s^s - \hat{r}_s \underline{i}_s^s) dt \tag{3.17}$$

The accuracy of the voltage model flux estimate is very sensitive to errors in any of its inputs, as indicated in Section 3.1.3. Wrong $\underline{\hat{\psi}}_{s0}^s$ will lead to a constant offset in the flux estimate that is not removed due to the estimator being an open integrator without feedback. Inaccuracies in $\hat{r}_s$ or $\underline{\hat{u}}_s^s$ will lead to an error in the flux estimate that increases with time. Any of the above scenarios will lead to inaccurate rotor position estimation that makes sensorless control an impossibility. As errors in $\underline{\hat{\psi}}_{s0}^s$, $\hat{r}_s$ or $\underline{\hat{u}}_s^s$ are very likely to occur in real-life scenarios, the voltage model will have limited practical use in sensorless motor drives if it is not further modified.

In the subsequent sections, it will be shown how the voltage model can be modified and corrected in order to obtain a flux model that estimates the stator flux with high accuracy even if any of its inputs are erroneously estimated. The corrected voltage model will as such become a viable means of accomplishing sensorless operation. It should however be emphasized that even with the implementation of the proposed modification, the voltage model will still struggle at low speeds, as the integration in Equation 3.17 will approach an open integrator as the speed approaches zero.

3.4.1 Stator Flux Linkage Drifting

If all inputs to the the voltage model are accurately estimated or measured, the resulting stator flux linkage estimate will in theory be a perfect representation of the actual stator flux of the machine. In steady state, this flux estimate will be a rotating space vector with constant amplitude, whose trajectory will form an origin-centered circle. In Figure 3.6, the trajectories of two stator flux linkage estimates are shown. The blue stator flux estimate is produced with $\underline{\hat{\psi}}_{s0}$, $\hat{r}_s$ and $\underline{\hat{u}}_s^s$ perfectly estimated, and this flux estimate should as such correspond with the actual stator flux. This flux estimate is denoted $\underline{\psi}_s^s$. The red flux estimate has been given an initial value with a DC offset equal to 0.3 per unit in the α-axis of its initial value, and is denoted $\underline{\hat{\psi}}_s^s$. The figure shows the trajectories of the stator flux space vectors in the first 2.5 seconds after a torque reference of 1 per unit has been applied.

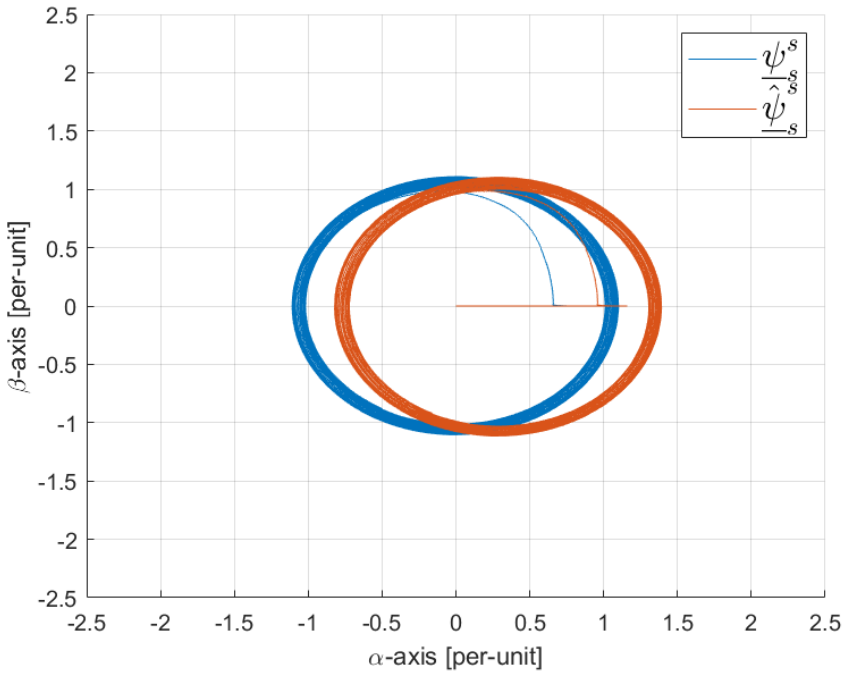
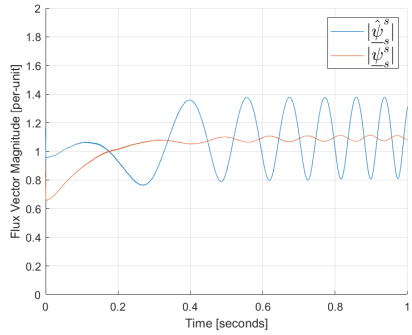
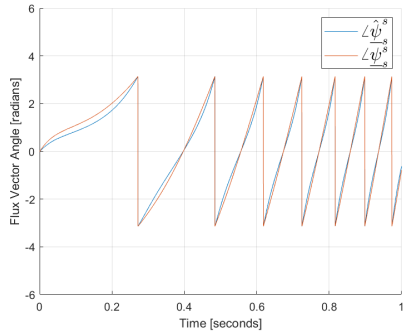


Figure 3.6: Real stator flux and flux estimate trajectories with erroneous initialization of the voltage model.

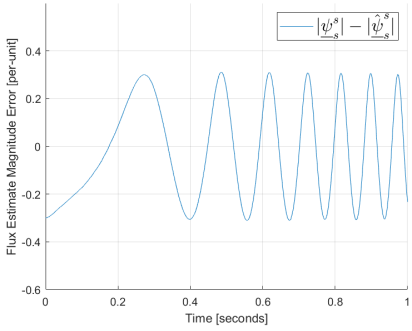
In Figure 3.6, it appears that the stator flux estimate has "drifted" away from the origin and become non-origin centered due to the erroneous initialization of the flux vector, demonstrating how the flux estimate vector may behave if it is erroneously estimated with the voltage model. In Figure 3.7, the time-series of the magnitudes and angles of the estimated and the real flux, as well as the magnitude error and the angle error of the flux estimate, are presented for the same scenario. The magnitude and angle error of the flux estimate are calculated as $|\underline{\psi}_s^s| - |\underline{\hat{\psi}}_s^s|$ and $\angle\underline{\psi}_s^s - \angle\underline{\hat{\psi}}_s^s$, respectively. Here, it can be seen that when the flux estimate drifts, the result is oscillating errors in both the magnitude and the angle of the flux estimate. This can be understood by considering Figure 3.6. When $\underline{\hat{\psi}}_s^s$ is pointing in the positive α-axis direction, its magnitude is larger than that of $\underline{\psi}_s^s$, while it becomes smaller after half electric period when the space vector has rotated $180°$. It is worth mentioning that the magnitude of $\underline{\hat{\psi}}_s^s$ in theory should be constant, given that it is estimated with no errors added to $\underline{\hat{\psi}}_{s0}$, $\hat{r}_s$ or $\underline{\hat{u}}_s^s$, but a small oscillation in the real stator flux' magnitude is observed as well in Figure 3.7a. This is most likely due to inherent inaccuracies in the current and voltage readings in the embedded real-time simulator, e.g. round-off errors, which results in some drifting of $\underline{\psi}_s^s$ as well.



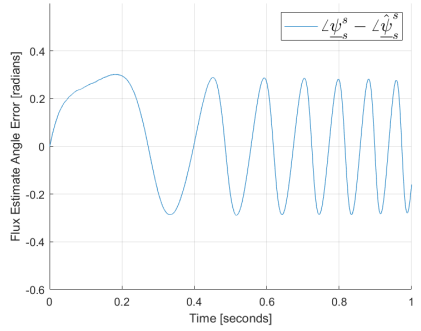
(a) Magnitude of the stator flux estimates.



(b) Angle of the stator flux estimates.



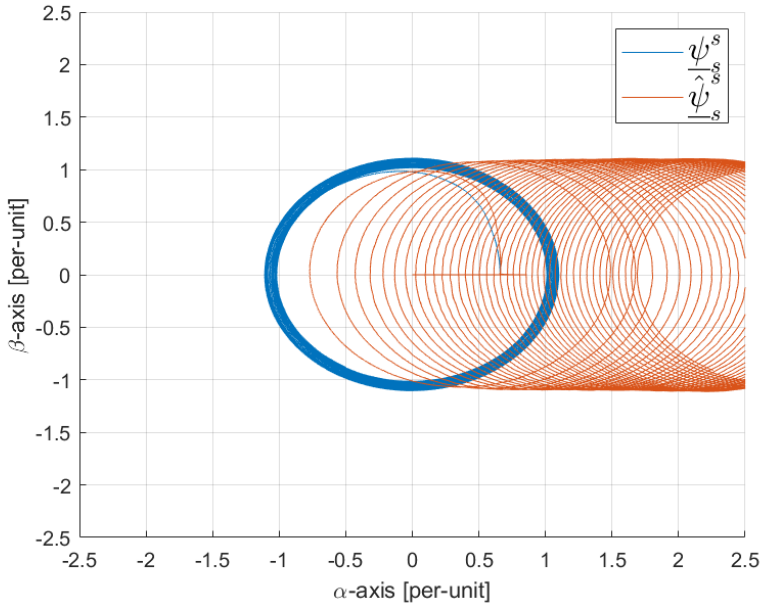
(c) Flux estimate magnitude error.



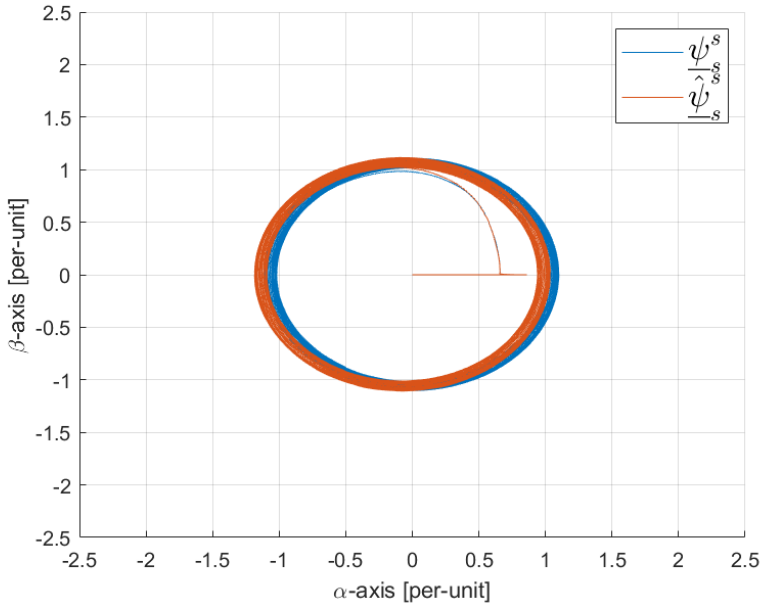
(d) Flux estimate angle error.

Figure 3.7: Voltage model flux estimate with erroneous initialization and voltage model flux estimate with all quantities accurately estimated.

Similar drifting of the flux estimate can be observed if there are offsets in the estimated stator voltage or if the stator resistance is erroneously estimated. These two scenarios are presented in Figure 3.8, where the blue and red flux estimates once again represent the accurate flux estimate and the erroneous flux estimate, respectively. In Figure 3.8a the stator voltage estimate contains a DC offset equal to 0.005 per unit in the α-axis, while in Figure 3.8b, the resistance is assumed to be underestimated with 20%, such that $\hat{r}_s = 0.8r_s$. While an erroneous flux initial value will remain in the flux estimate as a constant error, any errors in $\hat{\underline{u}}_s^s$ and $\hat{r}_s$ will be integrated in the voltage model for each time step, producing an flux estimate that drifts further and further away from the origin with time. Such errors therefore present a more severe problem than erroneous initialization of the flux estimate.



(a) Real stator flux and stator flux estimate with a DC offset in the voltage estimate.



(b) Real stator flux and stator flux estimate with underestimated resistance.

Figure 3.8: Stator flux linkage drifting due to erroneous stator voltage and resistance estimation.

3.4.2 Stator Flux Linkage Drift Correction: The Niemelä Method

To ensure that the flux estimate remains sufficiently accurate for the drive to operate without a position sensor, methods that prevent drifting due to errors in the inputs to the voltage model must be implemented.

Several methods for compensation of the drifting phenomenon have been proposed over the years. Wu and Slemon presented in [50] a method for drift compensation in which the extreme points of the flux estimate locus in the complex plane are used to generate separate correction terms for the real and imaginary components of the flux estimate. Another common approach to dealing with the drift problem is replacing the open integrator with a low-pass filter. However, the low-pass filter introduces an error in the phase angle and the magnitude of the flux estimate. Hu and Wu suggested in [27] three methods for modifying the low-pass filter by adding feedback compensation signals in order to reduce the flux estimate errors the low-pass filter causes.

Niemelä presented in [36] an alternative method for generating flux estimate correction terms that relies on the estimated flux vector itself, which will be adopted in this thesis. This method was shown to be capable of correcting flux drift for frequencies above 0.5 Hz in a 50 Hz synchronous machine, and even worked sufficiently for frequencies as low as 0.1 Hz. In the Niemelä method, the magnitude of the erroneous flux estimate is squared and filtered using a low-pass filter, and the difference between the filtered, squared estimate and the squared estimate is used to obtain an error, ϵ. That is,

$$\epsilon = |\hat{\underline{\psi}}_s^s|^2_{filtered} - |\hat{\underline{\psi}}_s^s|^2 \tag{3.18}$$

The error, ϵ, is multiplied with an correction coefficient, $k_{\psi,corr}$, and added to the original flux estimate, such that the after-correction estimate, $\tilde{\underline{\psi}}_s^s[k]$, becomes

$$\tilde{\underline{\psi}}_s^s[k] = (1 + k_{\psi,corr}[k] \cdot \epsilon[k]) \cdot \hat{\underline{\psi}}_s^s[k] = \hat{\underline{\psi}}_s^s + \underline{\psi}_{s,corr}^s \tag{3.19}$$

The drift-compensated flux estimate can hence be considered the sum of the original flux estimate and a correction component.

The low-pass filter that is used has the transfer function

$$H_{LPF}(s) = \frac{1}{1 + T_f s} \tag{3.20}$$

where T_f is the filter time constant. The filtering process of $|\hat{\underline{\psi}}_s^s|^2$, can hence be written as

$$|\hat{\underline{\psi}}_s^s|^2_{filtered}(s) = \frac{1}{1+T_f s}|\hat{\underline{\psi}}_s^s|^2(s) \tag{3.21}$$

which in discrete-time can be expressed as

$$|\hat{\underline{\psi}}_s^s|^2_{filtered}[k+1] = |\hat{\underline{\psi}}_s^s|^2_{filtered}[k] + \frac{T_{samp}}{T_f}(|\hat{\underline{\psi}}_s^s|^2[k] - |\hat{\underline{\psi}}_s^s|^2_{filtered}[k]) \tag{3.22}$$

The validity of the Niemelä drift correction concept can be demonstrated by considering Figure 3.9. Here, the flux estimate drifts due to an underestimated resistance, $\hat{r}_s = 0.8 r_s$. When the flux estimate drifts, the square of the magnitude of the flux will oscillate around its filtered value and produce a time-varying error signal. The error signal that is generated carries information about the instantaneous magnitude and direction of the drift: From Figure 3.9 it can be seen that when the error signal is positive, the square of the flux magnitude is smaller than the filtered value, meaning that positive correction flux must be added to produce the after-correction estimate. The magnitude of the correction component is proportional to the magnitude of ϵ, such that if the magnitude of ϵ is large, the magnitude of $\underline{\psi}^s_{s,corr}$ will increase. Similarly, a negative ϵ indicates that a negative correction flux should be added to produce the after-correction estimate, with the magnitude of the correction being proportional to the magnitude of ϵ.

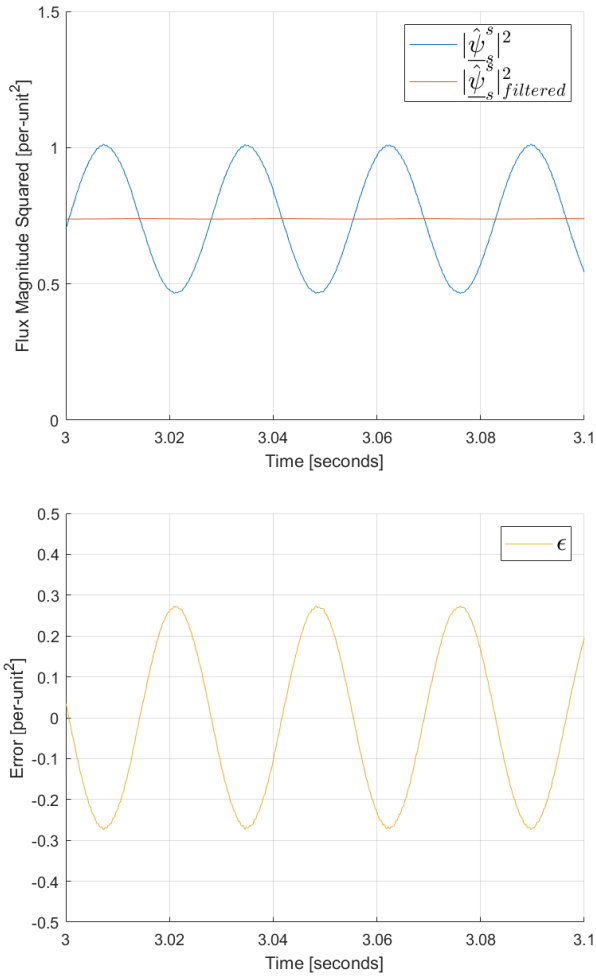


Figure 3.9: Flux estimate magnitude squared, flux estimate magnitude squared and filtered and the error signal.

The speed of rotation of the flux estimate space-vector will be equal to the frequency of the stator currents, meaning that the frequency of the drift-induced ripple in the squared flux estimate also will be equal to the stator frequency. To achieve effective filtering of the ripple in the flux estimate due to drifting, the time constant of the low-pass filter should therefore be made a function of the stator frequency. At low frequencies, the frequency of the ripple in $|\hat{\underline{\psi}}_s^s|^2$ is low, meaning that a large filter time constant will be necessary. Similarly, when the stator frequency is large, the frequency of the ripple in the $|\hat{\underline{\psi}}_s^s|^2$ will be large, and a smaller filter time constant is then sufficient to filter it out. To ensure that

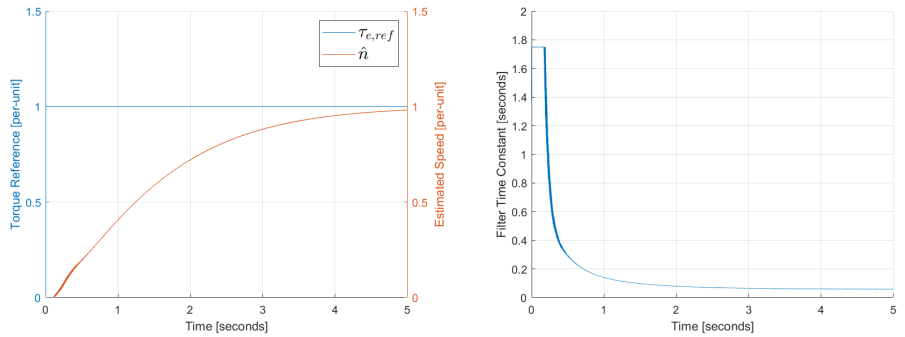
the filter time constant is adequately larger than the electric time period of the drift-induced ripple, the following formula will be used to obtain the filter time constant:

$$T_f = \frac{2}{\hat{n} \cdot f_n} \tag{3.23}$$

Here, $\hat{n}$ and f_n are the estimated motor speed and the nominal frequency, respectively. This formula yields a filter time constant that is approximately twice the electric period. Looking at the suggested formula, however, it is apparent that its value will grow very large during low operating speeds. When the time constant becomes very large, any variations in the stator flux linkage estimate due to changes in the torque reference could be eliminated by the filter, resulting in an inaccurate flux estimate. To avoid this, the time constant will be limited to a maximum value of 1.75, as proposed by [17]. The final formula for the filter time constant then becomes

$$T_f = \min\left(\frac{2}{\hat{n} \cdot f_n}, 1.75\right) \tag{3.24}$$

In Figure 3.10, the torque reference, the estimated speed and the value of the low-pass filter time constant during start-up of the motor are shown. During the initial moments of the start-up transient when the estimated speed is low, the filter time constant exceeds, and is therefore limited to, its maximum limit of 1.75. As the speed increases, the value of the filter time constant decreases until it reaches its steady-state value of approximately 0.04 seconds. This is twice the electric time period of the nominal frequency of 50 Hz.



(a) Torque reference and estimated speed.

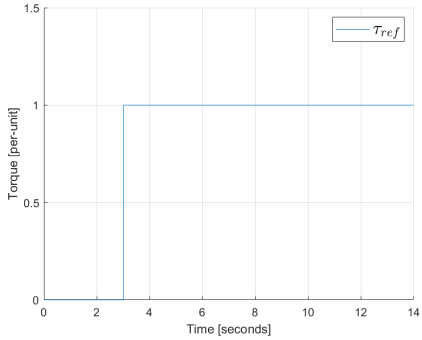
(b) Low-pass filter time constant.

Figure 3.10: Low-pass filter time constant during start-up.

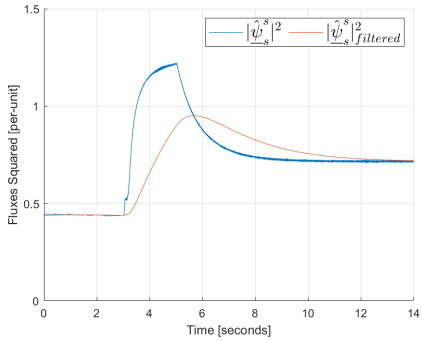
3.4.3 Modifying the Low-Pass Filter

As discussed by Luukko in [32], the dynamic behaviour of the low-pass filter that was proposed by Niemelä is not sufficient during rapid torque transients. During a step in the torque reference, the real stator flux squared will change as a result of the stator currents responding to a change in the reference values that are passed to the control system. The filtered square of the flux that is employed in the Niemelä method will however change considerably slower than the non-filtered value, which leads to the error vector, and hence the drift correction term, becoming non-zero. As a result, the flux estimate will be corrected during the torque transient, preventing the flux estimate from following the actual flux.

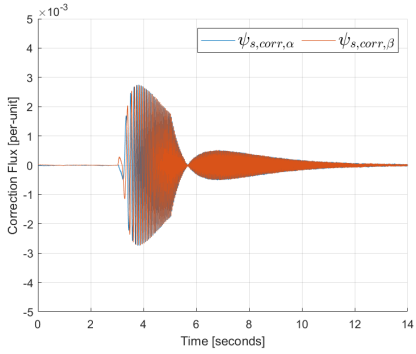
In Figure 3.11, this concept is demonstrated. Here, the response in $(\underline{\hat{\psi}}_s^s)^2$ and $(\underline{\hat{\psi}}_s^s)^2_{filtered}$ to a step in the torque reference can be seen. All inputs to the voltage model are assumed to be perfectly measured, which means that the flux estimate will not drift and correction of the estimate should as such not be necessary. The machine is at a standstill when the torque reference step is applied, meaning that the time constant of the low-pass filter is large. The filtered square of the flux therefore responds considerably slower to the torque reference step than the non-filtered value, creating a difference between the two that leads to a non-zero ϵ. As a result, the change in flux during the torque reference transient is recognized as drifting that needs correction, and an undesirable correction flux component $\underline{\psi}^s_{s,corr}$ is generated. The α and β components of the correction flux is shown in the bottom plot in Figure 3.11.



(a) Torque reference.



(b) Flux estimate magnitude squared and flux estimate magnitude squared and filtered.



(c) α and β components of the correction flux.

Figure 3.11: Dynamics of the squared flux estimate, the squared and filtered flux estimate and the correction flux during a torque reference transient.

In [32], Luukko proposed to modify the low-pass filter that is used to filter the square of the flux estimate to avoid drift correction during rapid torque reference transients. If the filter acts as a unity gain during the transient, the error vector is forced to zero and no correction is done. This is done by adding a derivative term to the previously used low-pass filter, such that it can be written as

$$H_{LPF}(s) = \frac{1 + T_d \cdot s}{1 + T_f \cdot s} \tag{3.25}$$

The relationship between the input and the filtered output of the filter in discrete time is now given by

$$|\hat{\underline{\psi}}_s^s|^2_{filtered}[k+1] = |\hat{\underline{\psi}}_s^s|^2_{filtered}[k] + \frac{T_{samp}}{T_f}(|\hat{\underline{\psi}}_s^s|^2[k] - |\hat{\underline{\psi}}_s^s|^2_{filtered}[k]) + \frac{T_d}{T_f}(|\hat{\underline{\psi}}_s^s|^2[k] - |\hat{\underline{\psi}}_s^s|^2[k-1]) \tag{3.26}$$

where

$$\frac{T_d}{T_f} = k_T \tag{3.27}$$

From Equation 3.25, it is seen that if k_T is equal to 1, the modified low-pass filter becomes a unity gain and no drift correction is performed. If k_T is equal to 0 on the other hand, the derivative term in the modified low-pass filter is eliminated and correction is performed. The value of k_T will be calculated based on Equation 3.28:

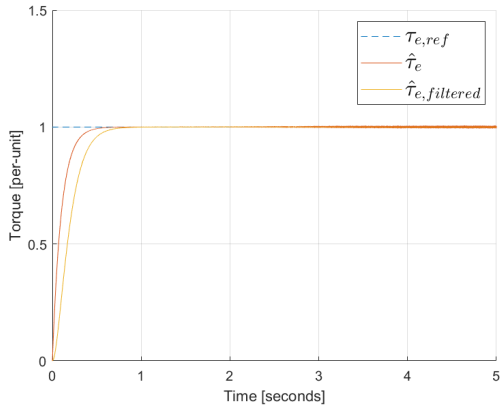
$$k_T = \min\left(1, k_{T0}|\hat{\tau}_e - \hat{\tau}_{e,filtered}|\right) \tag{3.28}$$

Here, $\hat{\tau}_e$ is the estimated value of the electromagnetic torque, which is found using Equation 2.20. The d and q axis currents in the torque equation are obtained using the estimated rotor position from the voltage model. $\hat{\tau}_{e,filtered}$ is a low-pass filtered value of the torque estimate, that is obtained using a low-pass filter with a filter time constant equal to 0.1 seconds, and the difference between $\hat{\tau}_e$ and $\hat{\tau}_{e,filtered}$ will be approximately equal to the per unit value of the torque step when it occurs. The value of the coefficient k_{T0} can be used to modify the behaviour of k_T: E.g., to disable drift correction for torque steps equal to or larger than 0.5 pu, k_{T0} should be equal to 2. The correction coefficient $k_{\psi,corr}$ can then be forced to zero during torque reference steps by defining

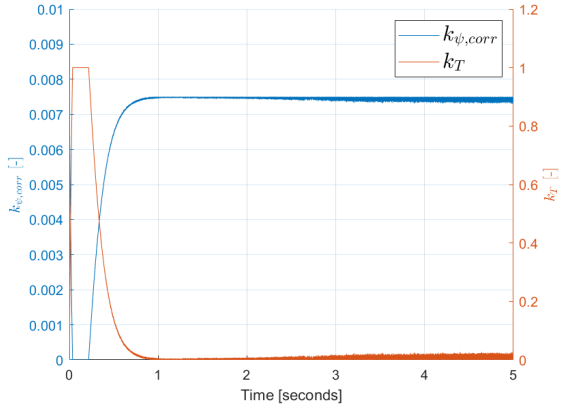
$$k_{\psi,corr} = (1 - k_T)k_{\psi 0} \tag{3.29}$$

According to [38], a typical value of $k_{\psi 0}$ is 0.0075, and this is the value that will be used here.

In Figure 3.12, the torque estimate, the filtered torque estimate, k_T and $k_{\Psi,corr}$ are shown in the seconds after a step in the torque reference equal to 1 per unit has been applied. Here, $k_{T0} = 4$ and $k_{\psi 0} = 0.0075$ have been used, which in theory should disable drifting completely during torque reference steps equal to and larger than 0.25. When the torque reference is applied, an error between τ_e and $\tau_{e,filtered}$ quickly establishes. Multiplying the error with the value of k_{T0}, a k_T that exceeds unity in the moments after the torque reference step is obtained. According to Equation 3.29, the result is $k_{\psi,corr}$ being set equal to zero, which leads to $\underline{\psi}^s_{s,corr}$ becoming zero and drift correction being disabled.



(a) Torque reference, estimated torque and estimated, filtered torque.



(b) The correction coefficients k_T and $k_{\psi,corr}$.

Figure 3.12: The values of k_T and $k_{\psi,corr}$ during a torque reference step transient.

With the Niemelä-method for drift correction and the modified low-pass filter for drift disabling during torque reference steps properly implemented, the corrected voltage model should be able to provide accurate flux estimation even in the presence of inaccuracies in $\underline{\hat{\psi}}^s_{s0}$, $\hat{r}_s$ or $\underline{\hat{u}}^s_s$. Sensorless operation based on the Niemelä-corrected voltage model will be investigated in Chapter 4.

3.5 Sensorless Operation Based on the Voltage-Current Model

In the previous section, the Niemelä-corrected voltage model was presented as a means of obtaining accurate flux estimation in scenarios where the uncorrected voltage model will fail. In this section, a closed-loop flux observer that combines both the voltage model and the current model for flux estimation is presented as an alternative method for flux estimation during sensorless operation.

As discussed in the previous section, the voltage model provides inaccurate flux estimates when the stator frequency is low, and it relies on some sort of drift compensation to correct for inaccurate model inputs. The current model, on the other hand, is accurate across the whole speed range if the model parameters are accurate and the rotor position is known. Bauer and Heining and Jansen and Lorenz suggested in [4] and [25], respectively, to combine the voltage and the current models with the goal of obtaining an improved flux model. In the proposed solution, the flux estimate from the current model acts as a reference value for the flux estimate from the voltage model to follow. The error between the two flux estimates is fed back to the voltage model via a controller, usually a proportional or a proportional plus integral controller, and hence acts as a stabilizing feedback term. The error between the reference flux and the output flux, $\underline{\epsilon}_s$, is then given by

$$\underline{\epsilon}_s = \underline{\hat{\psi}}^s_{s,i} - \underline{\hat{\psi}}^s_{s,u} \tag{3.30}$$

This error signal is passed to the controller in the feedback loop, which outputs a compensation voltage, $\underline{u}^s_{s,comp}$, that is added to voltage model integral. The flux estimate from the voltage model now becomes

$$\underline{\hat{\psi}}^s_s = \underline{\hat{\psi}}^s_{s0} + \omega_n \int_0^t (\underline{\hat{u}}^s_s - \hat{r}_s \underline{i}^s_s + \underline{u}^s_{s,comp}) dt \tag{3.31}$$

Separately, the voltage model and the current model are open-loop estimators, but by combining them as suggested, a closed-loop observer is obtained [25]. The structure of the voltage-current model is presented in Figure 3.13. Here, a proportional controller is used in the feedback-loop and the active flux concept is utilized for rotor position and speed estimation.

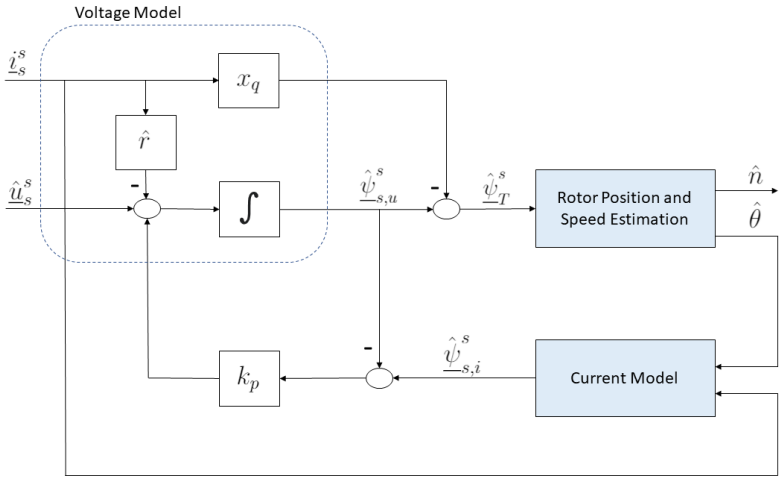


Figure 3.13: The voltage-current model with a proportional controller in the feedback loop. Reproduced from [33].

The closed-loop observer combines the advantages of the current model during low-speed operation with those of the voltage model during medium-high speed operation to provide an improved flux estimate. In general, the current model in the closed-loop observer can use the estimated or measured rotor position in order to transform stator-frame quantities into rotor-oriented quantities. For sensorless control applications, such as the one that is currently being considered, the input to the current model will be the estimated rotor angle from the voltage model. As a result, the closed-loop observer will still struggle during operation around zero speed.

3.5.1 Using a Proportional Controller in the Feedback Loop

In the method presented in [4], a proportional controller with a gain of k_p was used in the feedback loop in order to determine for which frequency the observer would transition from being current model reliant, to becoming reliant on the voltage model. The significance of the value of the feedback gain can be shown mathematically by considering the closed-loop transfer function from the current model reference flux estimate, $\underline{\hat{\psi}}_{s,i}^s$, to the

voltage model output flux estimate, $\underline{\hat{\psi}}^s_{s,u}$. The stator voltage, $\underline{\hat{u}}^s_s$, and the product of the stator resistance and the stator current, $\hat{r}_s\underline{\hat{\imath}}^s_s$, that are inputs to the voltage model can be considered disturbances that act on the control loop that is the closed-loop observer. Using the principle of superposition, these can be set equal to zero, and the impact of $\underline{\hat{\psi}}^s_{s,i}$ on $\underline{\hat{\psi}}^s_{s,u}$ can be investigated. The transfer function from the reference flux to the output flux becomes

$$\frac{\underline{\hat{\psi}}^s_{s,u}}{\underline{\hat{\psi}}^s_{s,i}} = \frac{k_p}{s + k_p} \tag{3.32}$$

For small frequencies $s \ll k_p$, the transfer function magnitude approaches unity, indicating that the flux estimate from the closed-loop observer follows the current model estimate closely in the low-speed region. When $s \gg k_p$, the transfer function magnitude approaches zero, which means that the flux estimate is now given by the voltage model. When $s = k_p$, the magnitude of the transfer function becomes $\frac{1}{2}$, meaning that the voltage model and the current model are trusted equally when producing the output flux estimate. Hence, by choosing the value of k_p, the speed for when the closed-loop observer transitions from being current-model reliant to becoming voltage-model reliant can be set.

The impact of the disturbances, $\underline{\hat{u}}^s_s$ and $\hat{r}_s\underline{\hat{\imath}}^s_s$, on the flux estimate can be exemplified by considering the transfer function from the stator voltage estimate $\underline{\hat{u}}^s_s$ to the voltage model flux estimate, $\underline{\hat{\psi}}^s_{s,u}$. As before, using the principle of superposition, the effect of the voltage on the voltage model flux estimate can be found by setting the current model flux estimate, $\underline{\hat{\psi}}^s_{s,i}$, equal to zero. The following transfer function is obtained:

$$\frac{\underline{\hat{\psi}}^s_{s,u}}{\underline{\hat{u}}^s_s} = \frac{1}{s + k_p} \tag{3.33}$$

The effect of a DC offset in $\underline{\hat{u}}^s_s$ equal to $\Delta\underline{\hat{u}}^s_s$ can be found using the final value theorem as shown below:

$$\lim_{t\to\infty} \underline{\hat{\psi}}^s_{s,u}(t) = \lim_{s\to 0} s\left(\frac{1}{s + k_p}\right)\frac{\Delta\underline{\hat{u}}^s_s}{s} = \Delta\underline{\hat{u}}^s_s \tag{3.34}$$

This shows that potential DC offsets in $\underline{\hat{u}}^s_s$ and $\hat{r}_s\underline{\hat{\imath}}^s_s$ will remain drift-inducing sources of error in the flux estimate that will not be removed over time. This can be verified by considering Figure 3.14. Here, a DC offset equal to 0.005 per unit is present in the α-axis stator voltage, and the response in the α-axis stator flux estimate is shown. The proportional controller is tuned with $k_p = 0.1$. From the figure, it is evident that the amplitude of the response in the α-axis flux estimate to the DC offset has a steady-state value equal to 0.05 per unit.

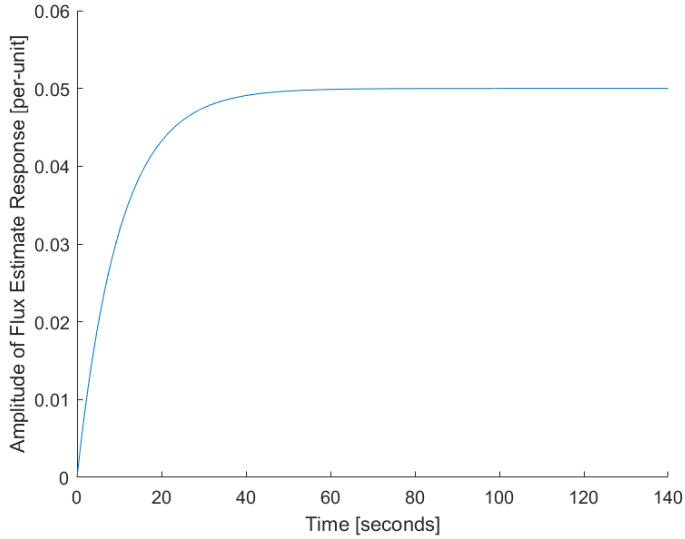


Figure 3.14: Response in the α-axis flux magnitude from the proportional controller based voltage-current model to a DC offset in the stator voltage.

3.5.2 Using a Proportional-Integral Controller in the Feedback Loop

To eliminate the steady-state error in the flux estimate due to errors in the inputs to the voltage model that were observed in the previous section, integral effect can be added to the controller in the observer's feedback loop. The combination of the PI-controller based voltage-current model and the active flux concept for rotor position and speed extraction in PMSM drives has been investigated in [40, 41, 8, 7, 8, 44]. The structure of this configuration is presented in Figure 3.15. The only difference compared to the voltage-current model from the previous section is that a PI controller is now used in the feedback loop.

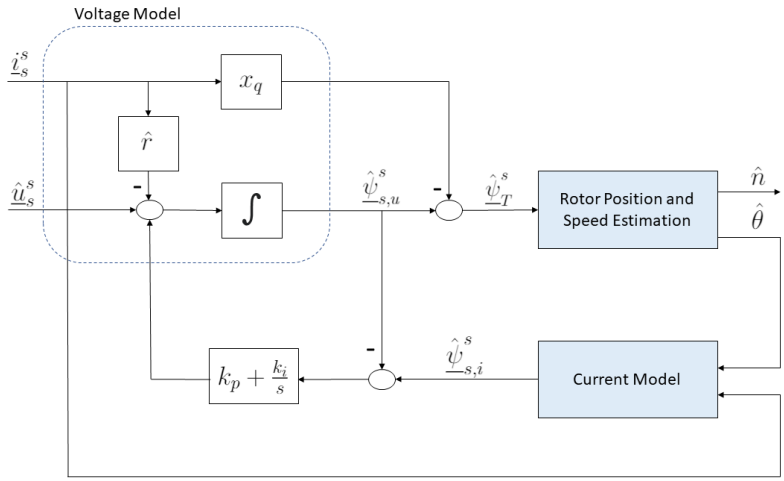


Figure 3.15: The voltage-current model with a proportional-integral controller in the feedback loop.

The PI controller transfer function can be written as

$$H_{PI}(s) = k_p + \frac{k_i}{s} \tag{3.35}$$

where k_p and k_i are the proportional and the integral gains of the controller, respectively. When replacing the proportional controller in the feedback loop of the voltage-current model with a PI controller, the closed-loop transfer function becomes

$$\frac{\underline{\hat{\psi}}_{s,u}^s}{\underline{\hat{\psi}}_{s,i}^s} = \frac{\psi_s}{\psi_{si}} = \frac{k_p s + k_i}{s^2 + k_p s + k_i} \tag{3.36}$$

while the transfer function from the stator voltage estimate to the flux estimate becomes

$$\frac{\hat{\underline{\psi}}^s_{s,u}}{\hat{\underline{u}}^s_s} = \frac{s}{s^2 + k_p s + k_i} \tag{3.37}$$

As before, the effect of a DC offset in the estimated stator voltage on the flux estimate can be investigated using the final value theorem:

$$\lim_{t\to\infty} \hat{\underline{\psi}}^s_{s,u}(t) = \lim_{s\to 0} s \left(\frac{s}{s^2 + k_p s + k_i} \right) \frac{\Delta\hat{\underline{u}}^s_s}{s} = 0 \tag{3.38}$$

Equation 3.38 shows that by adding integral effect to the controller in the feedback loop, the controller will eliminate the effect of errors in the input to the voltage model that would otherwise cause drifting of the flux estimate. Figure 3.16 shows the response of the α-axis flux estimate to a DC offset in the α-axis of the stator voltage estimate equal to 0.005 per unit with the PI controller implemented in the feedback loop. The controller has been tuned using $k_p = 0.1$ and $k_i = 0.1$. From the plot it is evident that by adding integral effect to the feedback controller, the impact of the DC offset in the estimated stator voltage on the α-axis flux estimate magnitude is eliminated over time.

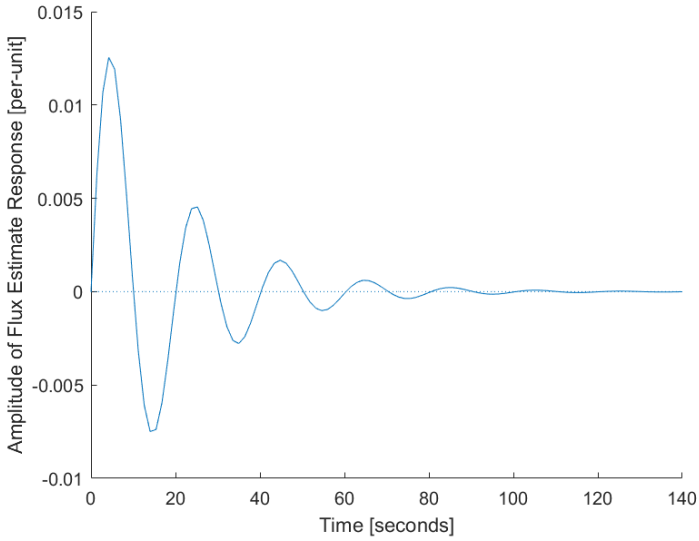


Figure 3.16: Response in the α-axis flux magnitude from the proportional-integral controller based voltage-current model to a DC offset in the stator voltage.

When proportional control was applied in the feedback loop, the relative importance of the voltage model and the current model in the closed-loop observer could be determined

simply by choosing the value of the proportional gain of the controller. When using a PI controller, choosing appropriate values of the controller gains becomes a bit more complex, and the tuning process usually involves a degree of trial and error. An educated guess of the controller gains can however be made using the pole placement method that is described next. The closed-loop observer's characteristic equation is given by the denominator polynomial of the closed-loop transfer function in Equation 3.36. This equation can also be expressed as a function of the system poles, λ_1 and λ_2, with $\lambda_1 \leq \lambda_2$, yielding the following relationship:

$$s^2 + k_p s + k_i = (s - \lambda_1)(s - \lambda_2) \tag{3.39}$$

The PI controller gains can hence be written as a function of the closed-loop observer poles, as shown in Equation 3.40.

$$k_p = -(\lambda_1 + \lambda_2) \tag{3.40a}$$

$$k_i = \lambda_1 \lambda_2 \tag{3.40b}$$

As described in [2], the system poles will characterize the observer's frequency behaviour in the following manner: The current model prevails for low frequencies when $s < |\lambda_1|$, while the voltage model will prevail for $s > |\lambda_1|$. A smooth transition between the two will take place for $|\lambda_1| < s < |\lambda_2|$. The method of pole placement can provide a starting point for the tuning of the PI controller, but in practice, trial and error will be necessary to obtain PI controller gains that results in an accurate stator flux estimate. In this thesis, the PI controller will be tuned using $k_p = 0.1$ and $k_i = 0.1$. These values produce an accurate stator flux estimate, and will be used in the remainder of this thesis.

The voltage-current model with a PI controller in the feedback loop is the configuration that will be used for the remainder of this thesis.

3.6 Possible Improvements of the Sensorless Control Schemes

This section contains a brief description of two concepts that can be used to improve the performance of the sensorless motor drive. Both concepts were attempted in this thesis, but not successfully implemented, and remain strong candidates for future work that can serve to boost the performance of the sensorless control schemes that have been proposed.

3.6.1 Phase-Locked Loop for Rotor Position Estimate Filtering

The inverse tangent function that is used in Equation 3.11 to extract the rotor position from the active flux vector might produce a noisy position estimate if there is significant amount of harmonics present in the active flux vector estimate [20]. The differentiation of the position to obtain the speed estimate may as a consequence become very polluted with noise. Especially when driving through zero speed, this effect may become prevalent as a consequence of inaccurate flux vector estimates from the voltage model around zero-speed. To diminish the potential noise in the position and speed estimate, a phase-locked loop (PLL) can be used to filter the estimated rotor position, as suggested in [38]. This approach was investigated by Bolstad in [9], and resulted in increased accuracy of the closed-loop observer, assuming erroneous stator resistance values. A description of the phase-locked loop, the proposed method and the results that were obtained are presented in Appendix F.

3.6.2 High Frequency Signal Injection for Standstill and Low-Speed Operation

In order to provide a comparison of the proposed fundamental-excitation based methods and saliency-tracking based methods for sensorless operation during low-speed operation, attempts were made to implement the high-frequency signal-injection method for saliency tracking that is presented in [44]. This method is based on the work done by [52] and [28], and works by adding a square-wave voltage component with small amplitude and frequency equal to the switching frequency to the d-axis stator voltage. The ripple in the stator current due to the high-frequency component in the stator voltage will be modulated with information about the rotor position, and by applying the correct demodulation technique, the position of the rotor can be obtained [28]. In the proposed method, an attempt to extract the rotor position using a phase-locked loop was made, but it was not successful. A more detailed description of the approach can be found in Appendix G.

Chapter 4

Real-Time Simulation Results

In this chapter, the sensorless control schemes that were presented in the previous chapter will be validated through simulation. The C++ sub-routines that are necessary for sensorless operation are added to existing control software in the programmable CPU of the SoC-based embedded controller that was presented in Section 2.3. Condensed versions of the most essential sub-routines are presented in Appendix H. These include subroutines for rotor position estimation using the active flux concept, stator voltage estimation and flux estimation using the Niemelä-corrected voltage model and the voltage-current model with a PI controller in the feedback loop.

The control software will in this chapter be used to drive the ERTS that runs on the FPGA of the SoC. The data for the motor drive that is implemented in the ERTS are presented in Appendix B.

Two methods for flux estimation that in combination with the active flux concept enable sensorless operation were presented in the previous chapter, and will be simulated in this chapter. They are:

1. Sensorless operation based on the Niemelä-corrected voltage model.

2. Sensorless operation based on the voltage-current model with a PI controller in the feedback loop. This flux model will be referred to as the ”voltage-current model” in the remainder of this thesis.

The Load

Initially, all simulations will be performed using a quadratic load function that is dependent on the rotor speed. The load torque will be given by

$$\tau_L = kn^2 \tag{4.1}$$

where the constant k is chosen such that the motor operates at nominal power at nominal speed. This load characteristic is valid for the modelling of fans, pumps and compressors [37].

Resistance and Voltage Estimation

To verify the validity of the sensorless control algorithms, the stator resistance will be assumed erroneously estimated in order to create scenarios in which the flux estimate will drift unless the proposed flux models work as intended. The stator resistance will be estimated such that

$$\hat{r}_s = 0.8 r_s \tag{4.2}$$

According to Bolstad, this scenario will occur at $91°$C for copper-based stator windings if the value of $\hat{r}_s$ is accurately estimated at $25°$C and then treated as a constant [9]. It may as such correspond to a situation in which the machine is heavily loaded, subjected to high ambient temperatures or both.

The stator voltage will be estimated according to the method presented in Section 3.3 in all simulations, and the drift correction algorithms will therefore have to correct drifting that results from both the inaccuracy in the resistance estimate and potential inaccuracies in the voltage estimate. All other motor parameters are assumed to be accurate.

Simulation Cases

The following cases will be simulated:

1. Sensorless operation without correction
2. Sensorless operation based on the Niemelä-corrected voltage model
   (a) High-Speed Operation Including Start-Up
   (b) Driving Through Zero-Speed
   (c) Very-Low Speed Operation

i. Start-Up and Very-Low Speed Operation
ii. Very-Low Speed Operation With High Initial Speed

(d) Driving Through Zero-Speed With Constant Load Torque

3. Sensorless operation based on the voltage-current model

(a) High-Speed Operation Including Start-Up

(b) Driving Through Zero-Speed

(c) Very-Low Speed Operation

i. Start-Up and Very-Low Speed Operation
ii. Very-Low Speed Operation With High Initial Speed

(d) Driving Through Zero-Speed With Constant Load Torque

4.1 Sensorless Operation Without Drift Correction

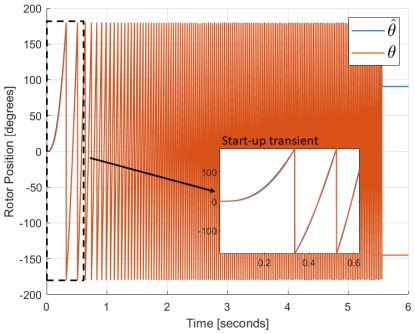
As an initial test, an attempt is made to operate the motor drive using the estimated stator flux from the voltage model without drift correction. This is done to demonstrate the need for specialized flux models when the inputs to the voltage model are inaccurately measured or estimated.

Figure 4.1 shows the true and estimated rotor position and the error between them in the seconds after a torque reference of 1 per unit is applied. The true speed, the estimated speed and the difference between them, along with the torque reference and the estimated torque, are also presented. The estimated torque is obtained using the torque expression in Equation 2.20. With all machine reactances assumed to be accurately measured or estimated, the estimated torque is equal to the actual torque output of the motor.

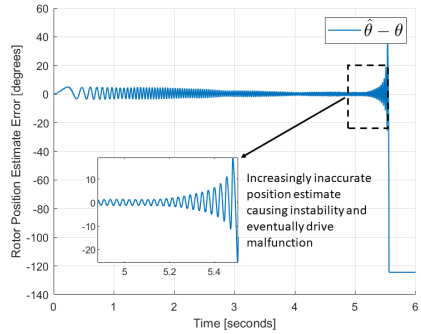
From Figure 4.1 it can be seen that an error in the estimated rotor position is established immediately after the torque reference is applied. During the initial start-up transient when the speed and the stator voltage is low, the flux estimate will be highly sensitive to the erroneous stator resistance estimate. The result is oscillating errors in both flux estimate and rotor position estimate. As the speed increases and approaches nominal speed, the impact of the inaccurate resistance becomes smaller and the magnitude of the position estimate error decreases. However, when the drive approaches its steady-state speed, the errors due to the inaccurate resistance will start to accumulate in the flux estimate, and the result is that the magnitude of the position estimate error will become increasingly large. After approximately 5.5 seconds, the inaccuracy in the rotor position estimate leads to drive

malfunction due to over-currents.

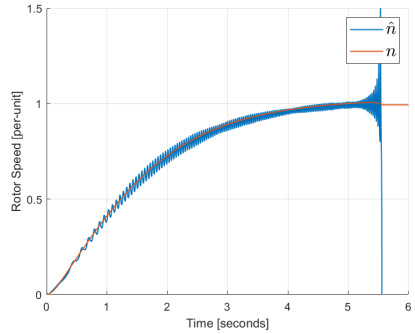
From the above results it is evident that an erroneous resistance estimate and the subsequent drifting of the flux estimate will quickly lead to an inaccurate rotor position estimate, which in turn causes problems for the control system. If the rotor position is inaccurate, the d- and q-axis current references will become erroneous and the performance of the motor drive will degrade. The following is thus clear: Flux models that are capable of estimating the stator flux with high accuracy even when the stator resistance estimate is erroneous are necessary for sensorless operation of the motor drive.



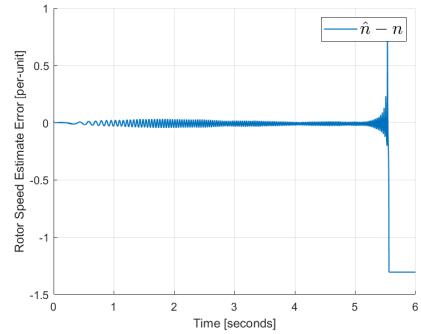
(a) Estimated and measured rotor position.



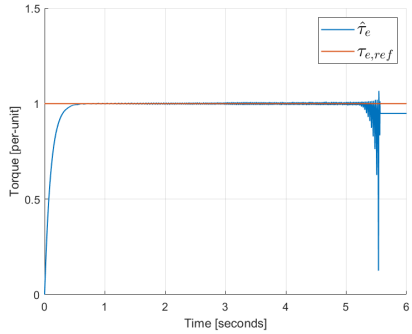
(b) Rotor position estimate error.



(c) Estimated and measured rotor speed.



(d) Rotor speed estimate error.



(e) Estimated output torque and torque reference.

Figure 4.1: Start-up and high speed operation without drift correction.

4.2 Sensorless Operation Based on the Niemelä-Corrected Voltage Model

In the preceding section, the need for correction of the flux estimate from the voltage model with inaccurate inputs was demonstrated. In this section, the Niemelä-corrected voltage model will be used for flux estimation.

Initially, the accuracy of the flux and rotor position estimate coming from the Niemelä-corrected voltage model during nominal operation is investigated. Then, the viability of using this method across the whole speed range will be examined by testing the method during zero-speed drive-through and very-low speed operation. These are challenging operating scenarios for sensorless drives, and will be used to investigate the robustness of the method. As before, the stator resistance has been estimated 20% too low to induce drifting of the stator flux estimate in all simulations.

The initial attempts of sensorless operation revealed that the sensorless drive based on the Niemelä-corrected voltage model is highly sensitive to noise in the rotor speed estimate. The original filter time constant of 0.005 is not sufficient to achieve satisfactory performance of the drive, and a filter time constant as large as 0.06 must be used in order for the drive to remain stable. It will thus be seen that the speed estimate from the Niemelä-corrected voltage model will carry a small amount of noise, but in return, it will struggle to follow the true rotor speed closely during rapid speed transients.

4.2.1 High-Speed Operation Including Start-Up

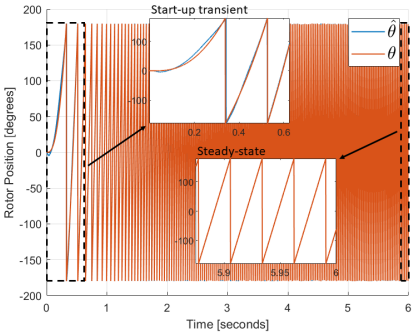
Figure 4.2 shows the true and estimated rotor position and the position estimate error in the seconds after a torque reference of 1 per unit is applied. The true speed, the estimated speed and the speed estimate error, along with the torque reference and the estimated torque are also presented.

During the initial start-up transient, there is an oscillating error in the rotor position that has a maximum value of approximately 15 degrees. The Niemelä correction algorithm does however quickly eliminate this error, and the position estimate locks onto the measured position after approximately 2 seconds, giving a steady-state error in the position estimate of 0 degrees. The Niemelä-corrected voltage model in combination with the active flux concept is thus capable of estimating the rotor position with high accuracy during high-speed operation. In steady-state, the position estimate is polluted by some noise that is resulting from the inverse tangent function that is used on the active flux vector com-

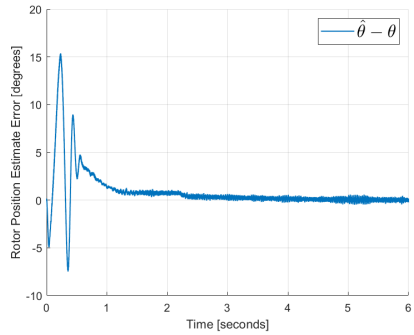
ponents in order to obtain the rotor position. This supports that filtering of the position estimate using a phase-locked loop, as suggested in Section 3.6, could produce a more accurate position estimate.

Oscillations in the speed estimate are also observed during the start-up transient. The maximum magnitude of the error in the speed estimate is approximately 0.04 per unit. As the motor approaches steady-state speed, a steady-state offset in the speed estimate error of slightly less than 0.01 per unit is established. This is due to the low-pass filtering of the speed estimate, that results in a delay between the speed estimate and the speed measurement. The error in the speed estimate is removed relatively slowly as a result of the aggressive tuning of the low-pass filter. As previously mentioned, a large filter time constant has been found to be necessary to operate the drive without a position sensor when the Niemelä-corrected voltage model is used, as the method is highly sensitive to noise in the speed estimate. From the results it can be seen that the large filter time constant leads to a relative noise-free estimate, but slows down its dynamics during transients.

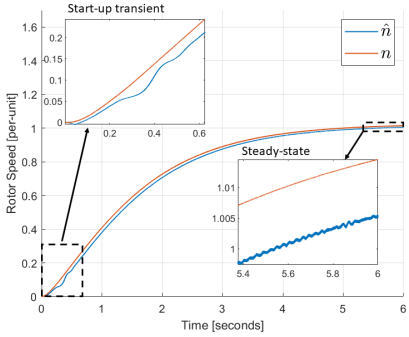
The torque output of the machine quickly reaches its reference value after the torque reference is applied. The distortion of the position estimate does however result in some noise in the torque output that could lead to problematic motor vibrations and noise.



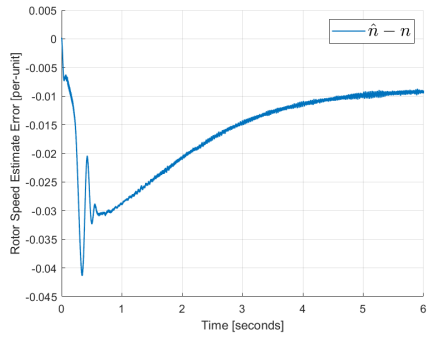
(a) Estimated and measured rotor position.



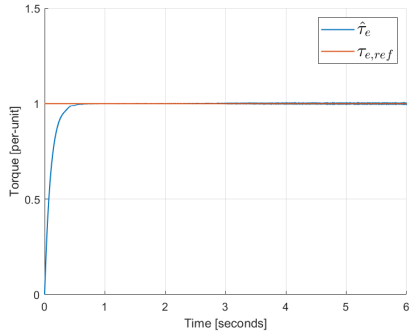
(b) Rotor position estimate error.



(c) Estimated and measured rotor speed.



(d) Rotor speed estimate error.



(e) Estimated output torque and torque reference.

Figure 4.2: Start-up and high speed operation with the Niemelä-corrected voltage model.

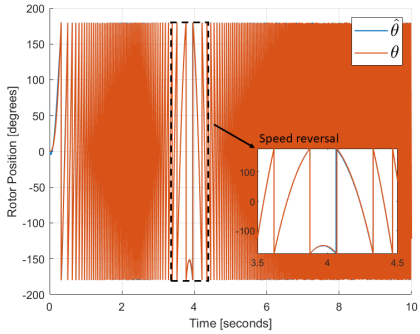
4.2.2 Driving Through Zero-Speed

As has been previously established, the accuracy of the flux estimate from the voltage model worsens when the motor approaches zero speed. While persistent operation at standstill is therefore not an option for a fundamental-excitation based sensorless motor drive, it may be able to cross zero-speed quickly without failure. To investigate the viability of this concept, the control system is subjected to the torque reference signal that is presented in Figure 4.3. During the first 2 seconds while the torque reference is 1 per unit, the motor accelerates and a positive direction of rotation is established. When the torque reference drops to -1 per unit, the motor decelerates and comes to a halt, before it starts accelerating in the negative rotational direction.

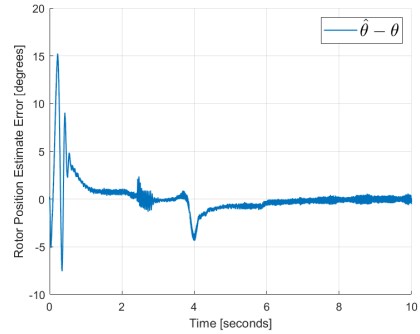
During start-up, the same oscillating error in the position estimate that was observed in the previous case is established. The error has approached zero when the second step in the torque reference occurs, at which point a small increase in the position estimate error occurs. In Figure 4.5, the correction terms k_T and k_ψ during the torque transient are displayed. Here, it is seen that k_T becomes equal to 1 and k_ψ becomes 0 during both torque reference steps. It is thus apparent that the correction method works as intended through the disabling of the drift correction during both positive and negative steps in the torque reference. As the speed approaches zero, there is an increase in the position estimate error. The error magnitude reaches a local maxima of around 5 degrees, before it decreases and is completely eliminated when steady-state speed is reached.

The speed estimate is negative during start-up, and the error magnitude has a local maxima of around 0.04 per unit. When the torque reference is reduced to -1 and the speed starts to decrease, the error becomes positive. This is a result of the filtering of the speed estimate that produces a error between estimated and real speed which is negative for increasing speed and positive for decreasing speed. A small spike in the error is observed when zero-speed is traversed. The torque output of the motor follows its reference closely across the whole speed range.

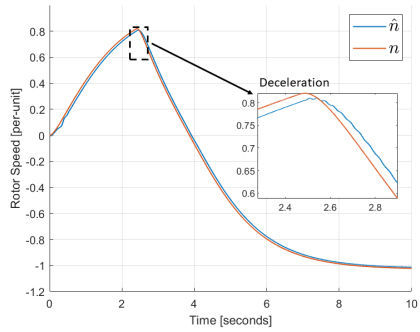
The results obtained in this section verifies that the Niemelä-corrected voltage model provides a flux estimate that enables the position-sensorless motor drive to perform zero-speed traversal without significant difficulties. Since the quadratic load that is employed in this scenario is a function of the square of the motor speed, the load torque is zero when the zero-speed is crossed. This, in combination with the inertia of the rotor, allows the motor to traverse zero-speed quickly. A more challenging scenario will be presented in Section 4.2.4, where driving through zero-speed with constant load torque will be attempted.



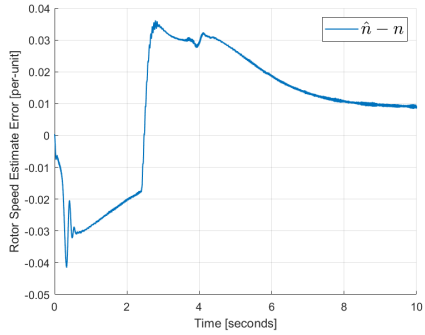
(a) Estimated and measured rotor position.



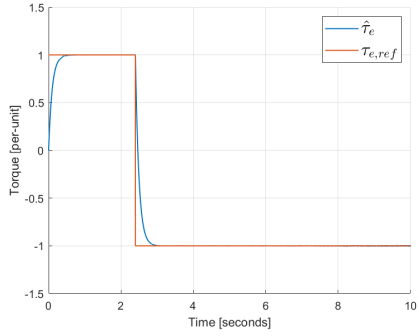
(b) Rotor position estimate error.



(c) Estimated and measured rotor speed.



(d) Rotor speed estimate error.



(e) Estimated output torque and torque reference.

Figure 4.3: Driving through zero-speed with the Niemelä-corrected voltage model.

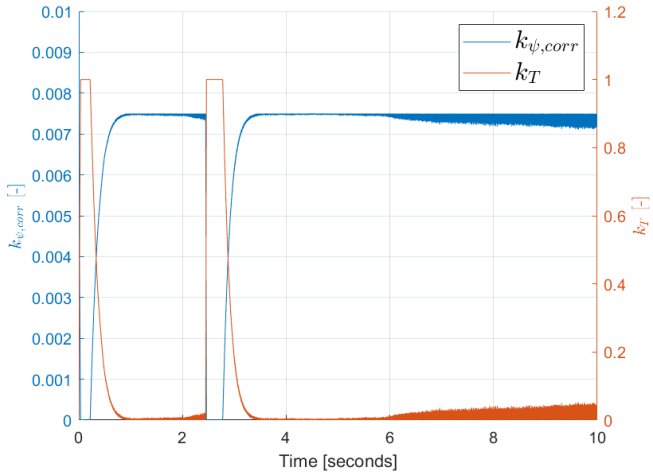


Figure 4.5: Correction terms used in the Niemelä-corrected voltage model when driving through zero-speed.

4.2.3 Very-Low Speed Operation

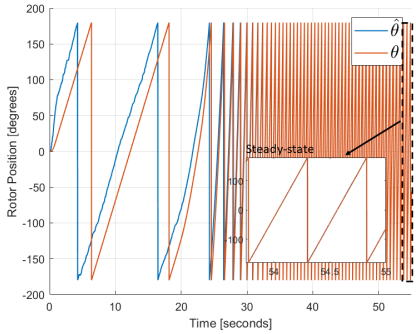
Start-Up and Very-Low Speed Operation

In order to examine the robustness of the current sensorless control method, the performance of the drive is tested during a very-low speed scenario. A torque reference of 0.01 per unit is applied as the motor is started, and the real-time simulator results are displayed in Figure 4.6. The simulation is performed over a period of 55 seconds.

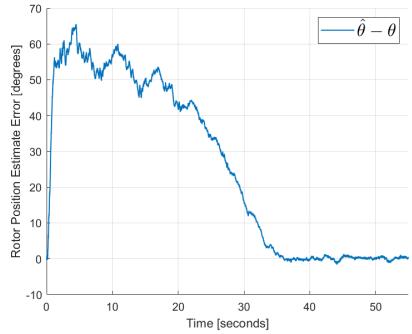
The position estimate error increases rapidly after start-up and reaches a maximum value of approximately 65 degrees before it starts to decrease. It is evident from the figure that the dynamic performance of the motor drive is very poor. The position estimate error is large during the first 35 seconds of simulation, after which it becomes zero. This proves that the rotor estimate takes very long to lock on to the measured rotor position, and this will present difficulties if the application involves rapid torque reference fluctuations during very-low speed. In spite of the position estimate error being large in the 35 initial seconds after start-up, the drive remains stable, and when steady-state speed is reached, the rotor position is estimated with high accuracy.

The speed response of the machine due to the applied torque reference is very slow. There is a small increase in speed during the initial second of the simulation, before the speed remains more or less unchanged throughout the next 20 seconds. As the accuracy of the

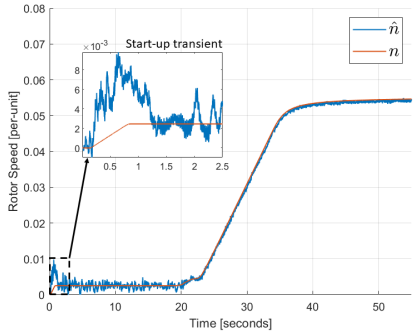
position estimate improves, the speed starts to increase and steady-state speed is reached after approximately 40 seconds of simulation. The inability of the drive to accelerate the motor during the first 20 seconds is interesting, and may be a direct result of the inaccurate position estimate during this period of time. The noise in the speed estimate and torque output of the motor is seen to become substantial during low-speed operation.



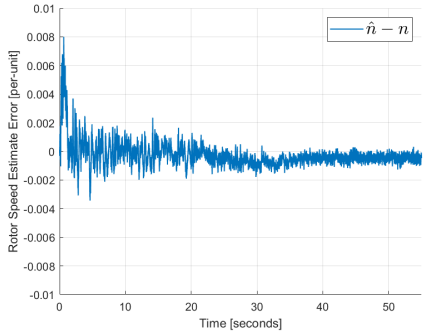
(a) Estimated and measured rotor position.



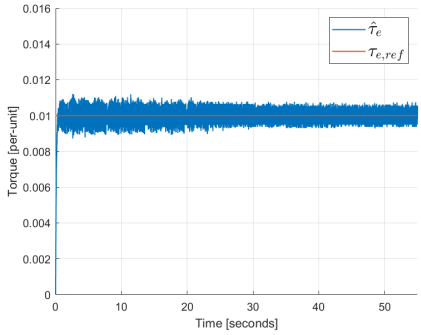
(b) Rotor position estimate error.



(c) Estimated and measured rotor speed.



(d) Rotor speed estimate error.



(e) Estimated output torque and torque reference.

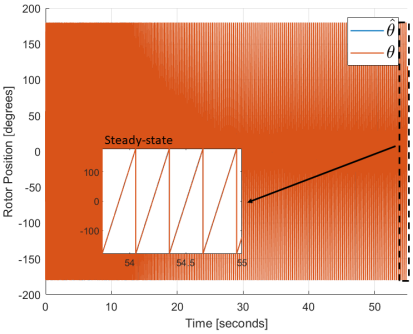
Figure 4.6: Start-up and very-low speed operation using the Niemelä-corrected voltage model.

Very-Low Speed Operation With High Initial Speed

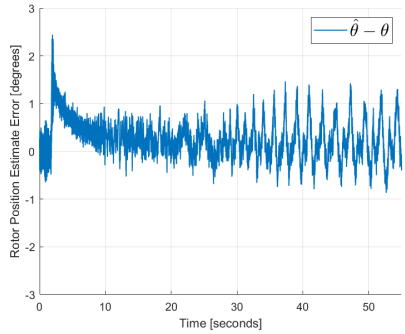
It is evident that the accuracy of the position estimate is poor initially if the drive is given a small torque reference when it is at a standstill. It is hypothesized that the drive's ability to operate at low speed will be better if the motor is already operating at high speed when the torque reference of 0.01 per unit is applied. To investigate this, the motor is brought to a steady-state speed of around 1 per unit by applying a torque reference of 1 per unit, before the torque reference is reduced to 0.01 per unit.

The results from this scenario are displayed in Figure 4.7. Here, it is seen that if the motor speed already is high when the torque reference of 0.01 per unit is given, the accuracy of the position and speed estimation is very good during the transient following the torque reference step. The error in the position estimate has a peak value of around 2 degrees and is quickly reduced to a mean value of 0 degrees as the steady-state speed is approached. There is however considerable noise present in the position estimate as the speed is reduced. The speed estimate error is seen to have an initial value of around 0.01 per unit, which is reduced to almost 0 as steady-state is approached. This is an interesting observation, as it suggests that the the steady-state error in the speed estimate is larger for high speeds than for low speeds.

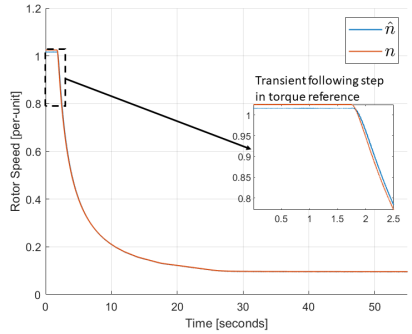
The results in this section indicate that very-low speed operation when using the Niemelä-corrected voltage model for flux estimation is a viable option. The drive seems to struggle if it initially is at a standstill when the small torque reference is applied, but in spite of the large position error during the start-up transient and the very slow speed response following the torque reference step, the motor drive remains stable and performs well during steady-state. If the motor is operated at high speed initially, the accuracy of the method is very good during the transient following the torque reference step, and estimation is done with almost perfect accuracy. Very-low speed operation thus only seems to be problematic if the motor has zero speed initially.



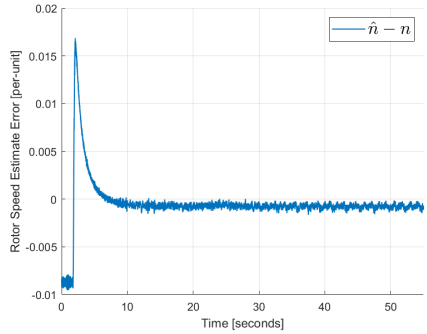
(a) Estimated and measured rotor position.



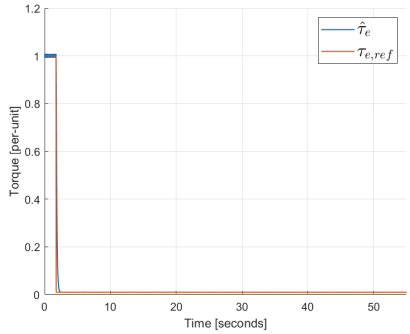
(b) Rotor position estimate error.



(c) Estimated and measured rotor speed.



(d) Rotor speed estimate error.



(e) Estimated output torque and torque reference.

Figure 4.7: Very-low speed operation with high initial speed, using the Niemelä-corrected voltage model.

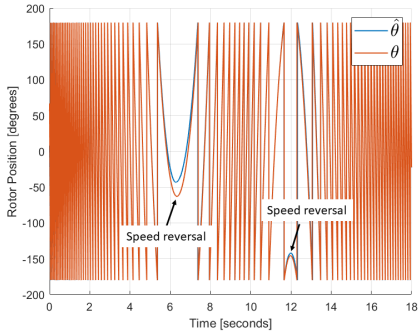
4.2.4 Driving Through Zero-Speed With Constant Load Torque

In Section 4.2.2, it was shown that the motor is capable of crossing zero-speed without noteworthy difficulties when the load torque is a quadratic function of the speed. When the motor is at a standstill, the quadratic load function becomes zero, which enables the motor to traverse the zero-speed region quickly. In this section, a more challenging operating scenario is investigated. Here, a constant load torque will be applied, such that the load torque is non-zero during the zero-speed drive-through. This load characteristics is typical for hoists and cranes, where the load torque will remain the same, independent of speed magnitude and direction [37]. The load torque is set equal to 0.2 per unit throughout the duration of the simulation. The motor torque reference is set equal to 0.3 per unit during the first 9 seconds, before it is reduced to 0.1 for the remainder of the simulation. This means that the difference between load torque and motor torque will be equal to 0.1 per unit at most, which means that the motor will both accelerate and drive through zero-speed slowly.

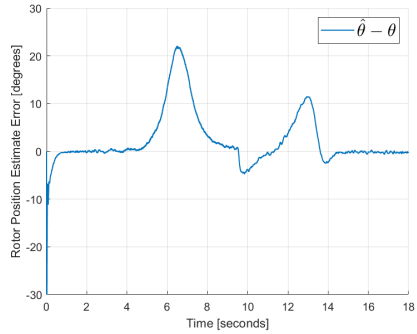
The real-time simulation results can be seen in Figure 4.8. In the moments immediately after start-up, the torque output of the motor is increasing from 0 towards its reference value, while the load torque is equal to 0.3 per unit. This results in the motor speed becoming negative initially, as the load torque is larger than the torque output for a short period of time. When the torque output of the machine exceed the magnitude of the load torque, the motor decelerates and eventually drives through zero-speed after roughly 6 seconds. The speed is now positive. When the torque reference is reduced to 0.1 per unit after 9 seconds, the motor decelerates and drives through zero-speed again, making the speed become negative. In this scenario, the machine operates as a generator when the speed is negative and as a motor when the speed is positive.

The position estimate is inaccurate immediately after start-up, but the error is quickly reduced. During both zero-speed drive-throughs, the estimate becomes inaccurate for a short period of time, with errors becoming in excess of 20 degrees and 10 degrees, respectively. However, the drive perform both speed-reversals with ease, although the position estimate errors are larger than when zero-speed drive-through with a quadratic load function was performed. This verifies that speed-reversals with a constant load torque presents a more challenging scenario.

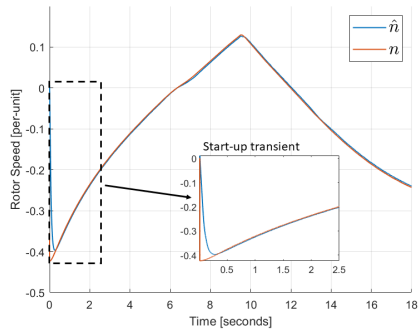
The results in this section indicate that the methods based on the Niemelä-corrected voltage model is an option also for applications that require a constant load torque.



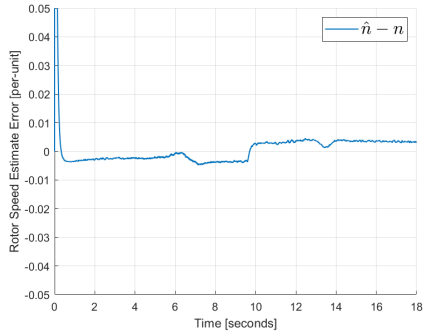
(a) Estimated and measured rotor position.



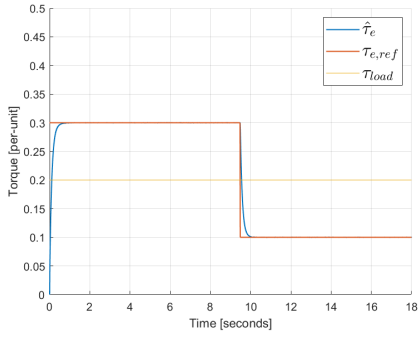
(b) Rotor position estimate error.



(c) Estimated and measured rotor speed.



(d) Rotor speed estimate error.



(e) Estimated output torque and torque reference.

Figure 4.8: Driving through zero-speed with constant load torque, using the Niemelä-corrected voltage model.

4.3 Sensorless Operation Based on the Voltage-Current Model

In the preceding section, it was established that sensorless operation based on the Niemelä-corrected voltage model is a viable option across the majority of the speed-range. In this section, the performance of the voltage-current model that was presented in Section 3.5 will be examined.

The voltage-current model based sensorless drive will be tested in the same operating scenarios as the Niemelä-corrected voltage model was tested in: Nominal operation, driving through zero-speed, very-low speed operation and driving through zero-speed with constant load torque. This way, comparisons between the two methods can be made. As before, the resistance is estimated 20% too low during all simulations to induce drifting of the flux estimate and a quadratic, speed-dependent load torque function will be used initially.

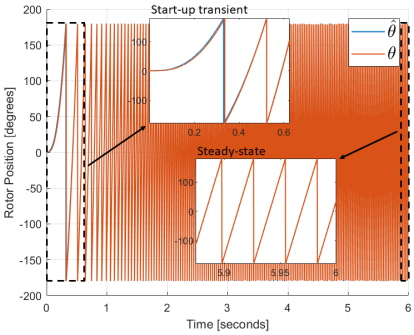
In this section, the original low-pass filter design for filtering of the speed estimate will be used. This means that the filter time constant will be set equal to 0.005. For the Niemelä-corrected voltage model, a large time constant was necessary for the drive to remain stable, but the method based on the voltage-current model does not seem to be very sensitive to noise in the speed estimate, and a smaller time constant can thus be used.

4.3.1 High-Speed Operation Including Start-Up

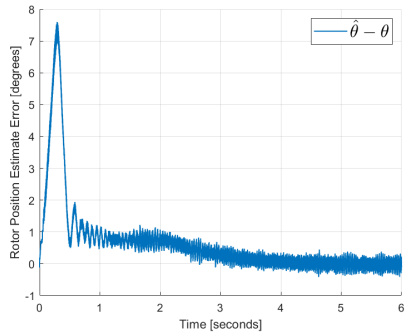
Figure 4.9 shows the result from the first 6 seconds of operation after a torque reference equal to 1 per unit is applied to the control system. The results reveal that the method based on the voltage-current model manages to estimate the rotor position with high accuracy throughout the duration of the simulation, with a peak in the estimate error occurring after start-up. In the moments following start-up, the error in the estimate increases and reaches a maximum value of less than 8 degrees. The error magnitude quickly decays, and as steady-state is approached, the error is eliminated. In comparison with the method based on the Niemelä-corrected voltage model, the start-up error in the position estimate from when using the voltage-current model is considerably smaller. The time necessary for the error to reach its steady-state value of approximately 0 degrees is similar in both cases.

The speed estimate error has small oscillations during start-up, before it reaches its steady-state value that has a small negative offset because of the difference between the measured

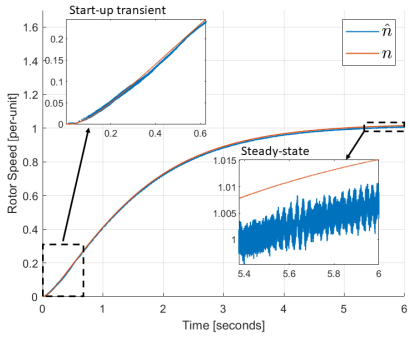
speed and estimated speed due to the low-pass filtering of the latter. The speed estimate is considerably more noisy than when the Niemelä-corrected voltage model was used, as this method required a much larger filter time constant, bu the dynamics of the estimate is however much faster. The torque output of the machine follows its reference value, with a small amount of noise being present in steady-state due to the noise in the position estimate.



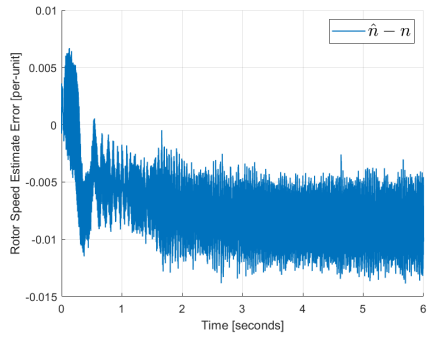
(a) Estimated and measured rotor position.



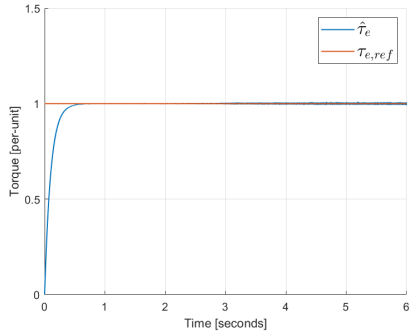
(b) Rotor position estimate error.



(c) Estimated and measured rotor speed.



(d) Rotor speed estimate error.



(e) Estimated output torque and torque reference.

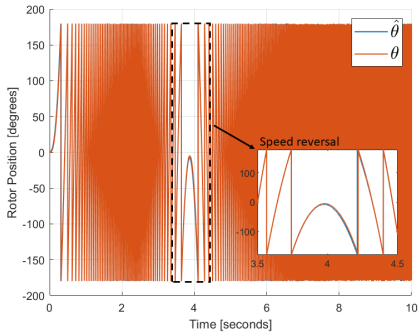
Figure 4.9: Start-up and high speed operation using the voltage-current model.

4.3.2 Driving Through Zero-Speed

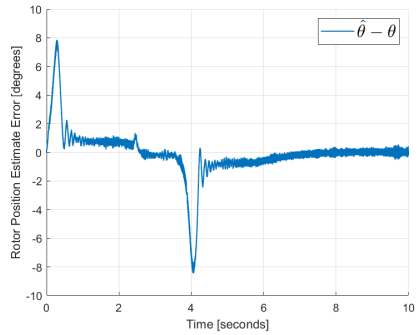
To examine the performance of the voltage-current model around zero speed, the torque reference is set equal to 1 per unit during the first two seconds of simulation, before it is reduced to -1 per unit during the remainder of the simulation.

From the results in Figure 4.10 it is seen that the voltage-current model is capable of driving through zero speed without noteworthy difficulty. The peak error has a magnitude of around 9 degrees during the zero-speed drive through, which is higher than the error when the Niemelä-corrected voltage model was used. The voltage-current model hence seems to provide a more accurate rotor position estimate during start-up, but does perform slightly worse during zero-speed drive through.

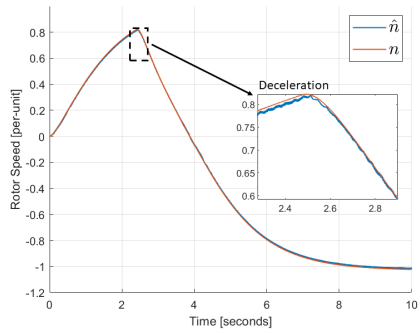
The speed estimate is accurate throughout the complete speed range, with a peak error magnitude of almost 0.015 per unit during the zero-speed drive-through. The torque output of the machine follows its reference closely.



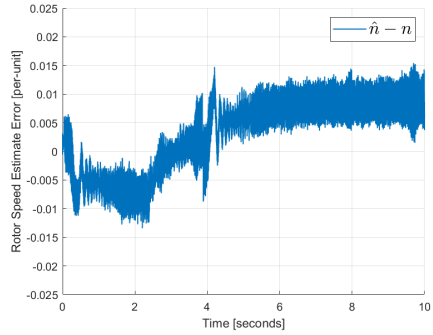
(a) Estimated and measured rotor position.



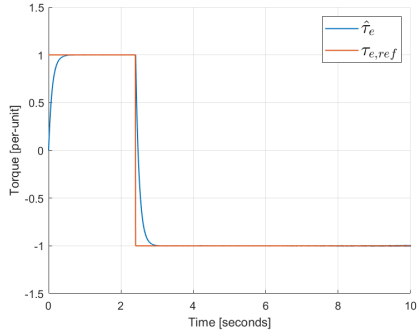
(b) Rotor position estimate error.



(c) Estimated and measured rotor speed.



(d) Rotor speed estimate error.



(e) Estimated output torque and torque reference.

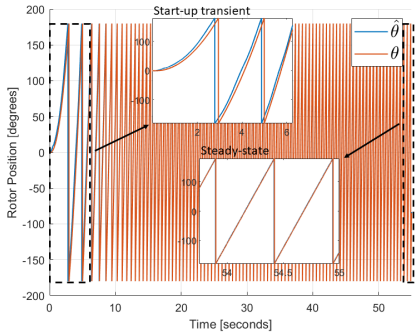
Figure 4.10: Driving through zero-speed using the voltage-current model.

4.3.3 Very-Low Speed Operation

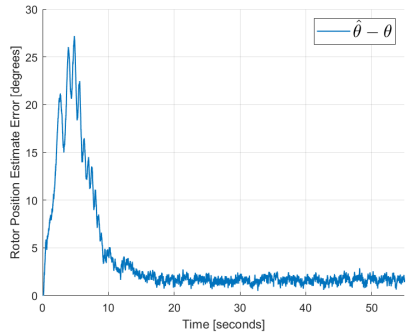
Start-Up and Very-Low Speed Operation

The results from the testing of the voltage-current model during start-up and very-low speed operation after a torque reference equal to 0.01 per unit was applied are shown in Figure 4.11.

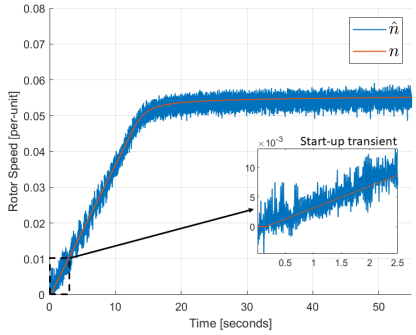
The error in the position estimate is seen to be relatively large during the initial 10 seconds of simulation, reaching a peak value of around 27 degrees. The position estimate is reduced considerably faster than when the method based on the Niemelä-corrected voltage model was used, and after approximately 15 seconds, the position error has been reduced to around 2 degrees. This error does however remain in the position estimate throughout the remainder of the simulation, and the voltage-current model seems to be incapable of removing this steady-state offset with the current tuning of the PI controller in the feedback loop. It is hypothesized that the error might be the result of insufficient integral effect in the PI controller in the feedback loop, and the integral gain of the controller is increased in an attempt to remove the offset. The integral gain is increased to point of the observer response becoming oscillatory and eventually unstable, with no improvement in the performance observed along the way. In an attempt to minimize the magnitude of the offset, the proportional gain of the controller is increased, but this does not yield any improvements in the accuracy either. The original tuning of the PI controller with $k_p = 0.1$ and $k_i = 0.1$ is hence kept. The steady-state offset in the error is to small to have any major impact on the control of the motor, and does therefore not present a significant issue.



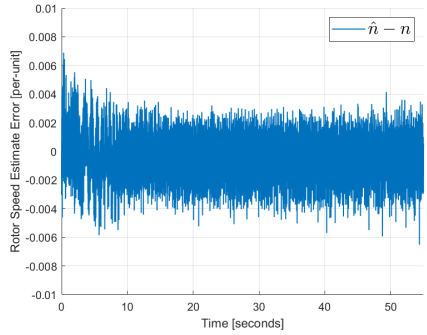
(a) Estimated and measured rotor position.



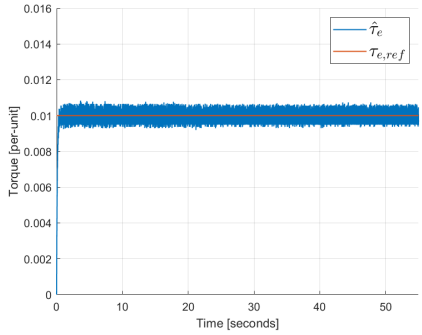
(b) Rotor position estimate error.



(c) Estimated and measured rotor speed.



(d) Rotor speed estimate error.



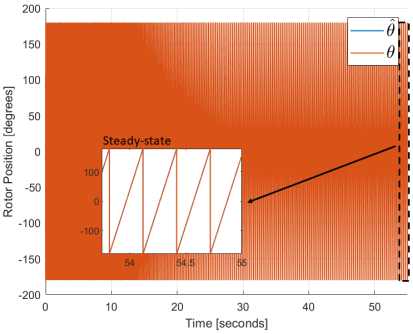
(e) Estimated output torque and torque reference.

Figure 4.11: Start-up and very-low speed operation using the voltage-current model.

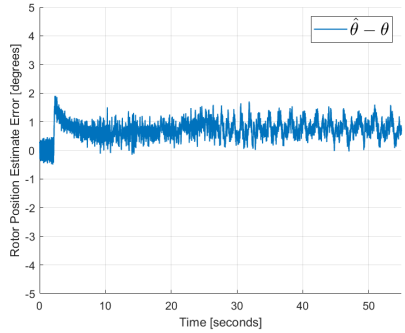
Very-Low Speed Operation With High Initial Speed

As when using the Niemelä-corrected voltage model, to investigate if the voltage-current model performs better during very-low speed if the initial speed is high, the motor is brought to steady-state operation at around 1 per unit speed, before the torque reference is reduced to 0.01 per unit. The results from this simulation are shown in Figure 4.12. As in the case of the Niemelä-corrected voltage model, the method based on the voltage-current model is able to estimate the rotor position with high accuracy when the speed is reduced to below 0.1 per unit. It is however seen that the steady-state offset in the position estimate that was observed previously is established also in this scenario when the speed becomes low. This suggests that the offset in the position estimate is a result of an inherent weakness in the voltage-current model that prevents it from estimating the stator flux perfectly during very-low speed operation, which does not depend on whether the motor is started from a standstill or brought to the very-low speed region from an high initial speed.

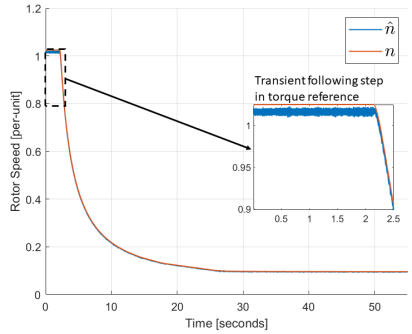
The speed response of the motor during low-speed operation following start-up is considerably better when using the voltage-current model than when the Niemelä-corrected voltage model was used. While the response in the latter case was extremely slow due to the large error in the position estimate during the first half of the simulation, the voltage-current model based motor drive speeds up immediately after the step in the torque reference. Considering the superior dynamic response of the voltage-current model, this method may be a better option than the method based on the Niemelä-corrected voltage model during very-low speed operation, in spite of the former's inability to remove the steady-state error in its position estimate.



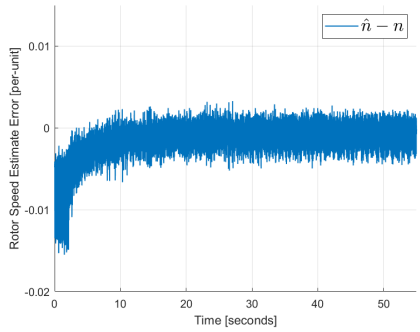
(a) Estimated and measured rotor position.



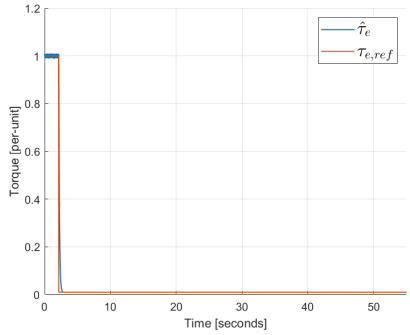
(b) Rotor position estimate error.



(c) Estimated and measured rotor speed.



(d) Rotor speed estimate error.



(e) Estimated output torque and torque reference.

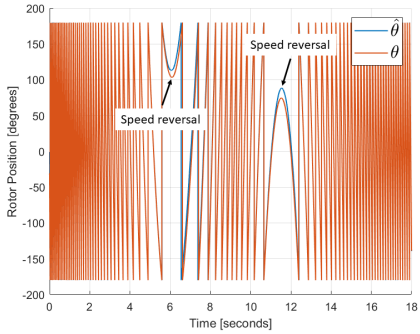
Figure 4.12: Very-low speed operation with high initial speed, using the voltage-current model.

4.3.4 Driving Through Zero-Speed With Constant Load Torque

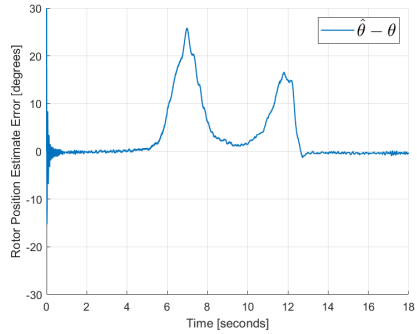
The results from the testing of the zero-speed drive-through capability of the voltage-current model under the influence of a constant load torque are displayed in Figure 4.13.

The error in both position estimate and speed estimate is seen to become very large during the initial start-up transient. These errors are however quickly eliminated, and the speed is estimated accurately for the remainder of the simulation. The position estimate error is quickly eliminated after the start-up transient, but spikes in the error occur during both zero-speed traversals. Similar to when the Niemelä-corrected voltage model was used, the error is largest during the second zero-speed crossing, when the rotational speed of the motor goes from being positive to becoming negative once again. The error reaches a value of around 25 and 15 degrees during the first and second zero-speed traversals, respectively.

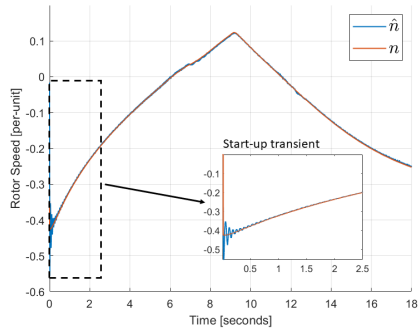
Slow zero-speed drive-throughs with a constant load torque thus seems to be a manageable task for the voltage-current model based method as well.



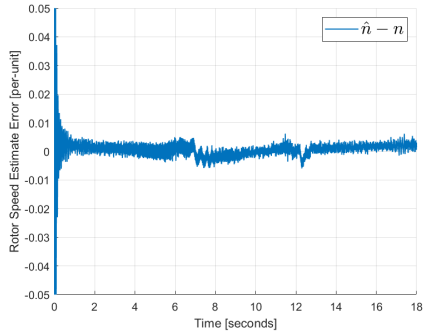
(a) Estimated and measured rotor position.



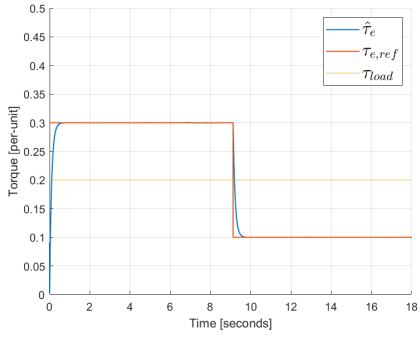
(b) Rotor position estimate error.



(c) Estimated and measured rotor speed.



(d) Rotor speed estimate error.



(e) Estimated output torque and torque reference.

Figure 4.13: Driving through zero-speed with constant load torque, using the voltage-current model.

Chapter 5

Discussion

In Section 3.1.3 the steady-state sensitivity analysis of the voltage model showed that the stator flux estimate is sensitive to errors in the stator resistance and stator voltage estimates when the motor speed is low. In Section 3.4.1 it was shown how such errors will lead to a flux estimate that becomes increasingly inaccurate over time due to the voltage model errors being integrated for each time step. These flux estimate errors cause inaccurate rotor position estimates, which in turn cause inaccurate d- and q-axis current references and a following deterioration of the performance of the motor drive. This was verified by the real-time simulation that was run in Section 4.1, where the motor drive was operated without a position sensor. Here, it was demonstrated that with the stator voltage estimated using the converter reference voltage and the stator resistance being estimated 20% too low, the rotor position estimate becomes increasingly inaccurate over time, eventually causing a malfunction of the motor drive. This shows that modifications of the conventional voltage model must be implemented in order to achieve a sensorless control algorithm that is practically feasible. Two alternatives for an improved flux estimate were presented: The Niemelä-corrected voltage model and the voltage-current model.

5.1 Comparison of the Performance of the Flux Models

Both the methods based on the Niemelä-corrected voltage model and the voltage-current model were capable of estimating the rotor position with high accuracy during start-up and nominal operation when the torque reference was equal to 1 per unit. While both methods eliminated the position estimate error equally fast, the error with the Niemelä-corrected method was considerably larger and more oscillatory during start-up than when the voltage-current model based method was used. Both methods were able to perform

speed-reversals with ease when the torque reference was varied from 1 per unit to -1 per unit, although spikes in the position estimate errors occurred when zero-speed was crossed. The error during the zero-speed traversal was lowest when using the Niemelä-corrected voltage model. The zero-speed traversals in the above scenario were relative fast due to the quadratic load torque function being used: At zero speed, the torque load become zero, allowing the motor to traverse zero-speed very quickly, and to challenge the drive, slower zero-speed drive-throughs were therefore performed by using a constant load torque. The results from these simulations show that both methods struggle more in this scenario, but both methods were still able to cross zero-speed without significant problems. In this scenario, the accuracy of the position estimate was similar for both methods, but the error was slightly smaller when the Niemelä-corrected voltage model was used.

When a very-low torque reference was used, both methods struggled during start-up and it took a significant amount of time before the position estimate errors were reduced. However, the dynamics when using the voltage-current model were superior to the method based on the Niemelä-corrected voltage model, with the former clearing the position estimate error considerably faster. When the very-low torque reference was applied with the motor already operating at high speed, both methods estimated the rotor position almost perfectly. This shows that start-up and subsequent low-speed operation is a much more challenging scenario than when the speed is reduced from high speed to very-low speed. During both low-speed scenarios, there was a small steady-state offset in the position estimate when using the voltage-current model that was not removed even after further tuning of the PI controller in the feedback loop of the voltage-current model was attempted. However, this error seems to be too small to have any practical consequences. The intermediate results from the trial-and-error based tuning procedure that was employed on the PI controller of the observer are not included in this thesis, but the process revealed that the performance of the voltage-current model is highly sensitive to the tuning of the PI controller in the feedback loop, with small adjustments in any of the tuning parameters resulting in inability to estimate the rotor position accurately. The controller gains that were used in this thesis appear to be optimal values for this specific configuration.

As mentioned in Chapter 4, when using the Niemelä-corrected voltage model, the performance of the drive was highly sensitive to noise in the speed estimate. The time constant of the speed estimate filter had to be increased to 0.06 in order for the drive to remain stable, which resulted in a speed estimate with a small amount of noise, but with relative slow dynamics. In comparison, a filter constant as low as 0.005 seconds was used for the voltage-current model. This could present a problem if the drive were to be tested under speed-controlled operation, where the estimated speed would be used as feedback

and compared with the reference speed. If the speed estimate changes considerably slower than the real speed due to the filtering, this could lead to deterioration of the drive's performance and potentially instability.

The simulations in this thesis were performed with a resistance value that was underestimated with 20% in order to emulate a scenario in which flux models with drift-correction capabilities are necessary. Due to the temperature-dependency of the stator resistance, having a real stator resistance value that differs from the estimated value is a very likely scenario, and the simulations in this thesis can therefore be deemed applicable for situations that may be encountered in the real world. All other motor parameters were assumed to be accurately measured, although the estimated values of the permanent magnet flux linkage and the motor inductances in practice also will differ from their real values. This may be particularly problematic for the voltage-current model that relies not only on the voltage model, but also on the current model, in which these parameter values are included. In the case of erroneously estimated inductances and permanent magnet flux linkage, the flux reference provided by the current model in the voltage-current model would differ from the real flux, which would results in an error being fed back to the voltage model even if the voltage model flux estimate was accurate. Such a scenario would most likely worsen the performance of the voltage-current model. The fact that the Niemelä-corrected voltage model only relies on a single motor paramater, namely the stator resistance, while the voltage-current model relies on the stator resistance, stator inductances and permanent magnet flux linkage could be an advantage of the former. This can be verified through simulations in which also the inductances and the permanent magnet flux linkage are erroneously estimated.

5.2 General Remarks About Fundamental-Excitation Based Sensorless Control

The sensorless control methods that have been investigated in this thesis are both based on the voltage model, and do hence fall within the fundamental-excitation based methods category. While such methods typically struggle around zero-speed, the simulation results that are obtained in this thesis indicate that both the method based on the Niemelä-corrected voltage model and the method based on the voltage-current model perform satisfactory during zero-speed traversals, and are even able to remain stable during persistent operation at speeds below 0.1 per unit. For applications that require very high accuracy in

the very-low speed region, saliency tracking methods such as the high-frequency signal-injection method that is presented in Appendix G may have to be implemented. This would add complexity and cost to the drive system, and considering that persistent very-low speed operation may not be a common scenario in most applications, cost-benefit analysis would be necessary to determine whether or not such additional methods should be implemented. The results obtained from the real-time simulations in this thesis demonstrate that the FE-based methods that were implemented in this thesis are satisfactory accurate for scenarios that include start-up, medium and high-speed operation and zero-speed drive-throughs, and are even capable of persistent very-low speed operation. Additionally, the methods were proven to be effective for both speed-dependent and constant load torques, making the proposed methods viable options across most of the speed range for a broad range of applications, including pumps, fans, hoists and cranes.

5.3 The ERTS and Assessment of Result Validity

Ideally, the control software that was developed in this thesis would have been verified experimentally on a physical laboratory setup. The development of a fully functioning PMSM drive test bench is an ongoing effort within the PESC group in the Department of Electric Power Engineering at NTNU, but due to delays caused by the ongoing Covid-19 pandemic, it has not yet been made available. This thesis has hence relied solely on the FPGA-based ERTS for control software verification. The ERTS has proved to be a useful tool for verification of the C++-based control software without having access to a physical motor drive setup. In addition to providing a means of software-in-the loop testing and final verification of the control software, the real-time nature of its simulations, due the high processing capability of the FPGA, has made it possible to investigate the long-time dynamics of the position-sensorless motor drive. This was useful when the performance of the position-sensorless motor drive was investigated during very-low speed operation, when simulations with a duration of 55 seconds were performed. Performing similar simulations with conventional simulation tools such as MATLAB Simulink would be a highly time-consuming effort, making the ERTS tool an invaluable asset in the development and verification of sensorless control methods.

The accuracy of the ERTS was investigated by Perera et. al in [45], where the PMSM drive hardware emulator was compared to an equivalent MATLAB Simulink model. The results from this analysis showed negligible numerical errors in d- and q-axis currents, rotor position estimate and torque output between the ERTS and the MATLAB Simulink

model during a 1 second simulation run. This indicates that the accuracy of the ERTS is high, and it is hence reasonable to trust the results that are obtained in this thesis with a relatively high degree of certainty. The ERTS performed the 1 second simulation in real-time, while the run-time of the Simulink model was in the vicinity of 550 seconds. The usefulness of the ERTS as a tool is thus evident: Using it, simulations can be considerably accelerated at the expense of practically insignificant numerical errors.

Chapter 6

Conclusion and Further Work

6.1 Conclusion

In this thesis, C++ software that enables sensorless indirect torque control of a permanent magnet synchronous machine has been successfully developed and implemented using a state-of-the-art SoC-based embedded control platform. The validity of the control strategies has been verified using an FPGA-based embedded real-time simulator.

Two different flux models have been used in combination with the active flux concept to facilitate position-sensorless operation: The Niemelä-corrected voltage model and the voltage-current model. Both flux estimation methods have enabled sensorless operation across the majority of the speed range when the stator resistance is erroneously estimated, and have performed well during scenarios that include start-up and high speed operation and zero-speed drive-throughs. Moreover, very-low speed operation below 0.1 per unit speed has also proved to be a viable option. By performing tests with both quadratic and constant load functions, the methods have been verified for several applications, such as fans, pumps, hoists and cranes.

In general, the method based on the Niemelä-corrected voltage model and the voltage-current model have performed similarly. The method based on the voltage-current model seems to be capable of removing errors in the position estimate considerably faster than the one based on the Niemelä-corrected voltage model. In return, its steady-state position estimate seems to be less accurate, containing a small steady-state offset of around 1-2 degrees when the speed is low. A potential drawback of the method based on the Niemelä-corrected voltage model is that it requires aggressive filtering of speed estimate to remain

stable. This could present difficulties if the drive were to be speed-controlled, as the large low-pass filter time constant results in a speed estimate that changes notably slower than the actual rotor speed during transients. However, this did not seem to have any significant negative impact on the torque-controlled drive. In conclusion, it can be stated that the Niemelä-corrected voltage model and the voltage-current model in combination with the active flux concept both provide well-performing fundamental-excitation based torque controlled sensorless PMSM drives. Which flux model to use may become a trade-off between fast dynamic response and steady-state accuracy. Both methods are able to operate across the majority of the speed range for a broad range of applications.

The FPGA-based ERTS has proved to be an invaluable tool in the development of motor control software. Using it, software-in-the-loop testing has been made possible, enabling frequent testing of the motor control strategies. In this thesis, it has also been used for final verification of the proposed C++ control software, due to the lack of a fully functioning PMSM drive test bench. Additionally, its real-time nature has made the production of what can be assumed to be highly accurate results a considerably less time-consuming than if conventional simulations tools were to be used.

6.2 Further Work

The following suggestions for further work are made:

6.2.1 Software Verification on a Physical Laboratory Setup

The control software that was developed in this thesis has relied solely on the ERTS for verification. Although the ERTS can be trusted to produce high-accuracy results, testing of the control strategies on a physical laboratory setup would provide final validation of the concepts that have been proposed, and should thus be performed.

As previously discussed, the PMSM drive test bench that is currently being development by the PESC group in the Department of Electric Power Engineering was not made available during the work with this thesis due to delays caused by the Covid-19 pandemic, but it is likely that it will be available for future candidates.

6.2.2 Improving the Performance of the Position-Sensorless Drive

In Section 3.6, two strategies for improvement of the performance of the sensorless motor drive were presented:

- Including a PLL for rotor position filtering
- Implementing high-frequency signal-injection for saliency tracking and improved standstill and low-speed performance

The implementation of the above methods may boost the performance of the PMSM drive. Both of the above strategies were attempted implemented in this thesis without success. The approaches that were used are presented in Appendix F and Appendix G, and future attempts of implementation could use the theory here as a starting point.

Another interesting concept for improved performance of the position-sensorless PMSM drive is:

- Implementing online parameter estimation algorithms in order to obtain more accurate motor parameter estimates in real-time.

6.2.3 Further Testing of the Position-Sensorless Drive

The following scenarios were not investigated in this thesis, and may be tested in the future:

- Simulating the position-sensorless drive during speed-controlled operation.
- Simulating scenarios with erroneously estimated permanent magnet flux linkage and stator inductances to investigate how they affect the performance of the voltage-current model.

Bibliography

[1] F. Alvarez-Gonzalez, A. Griffo, B. Sen, and J. Wang. Real-Time Hardware-in-the-Loop Simulation of Permanent-Magnet Synchronous Motor Drives Under Stator Faults. *IEEE Transactions on Industrial Electronics*, 64(9):6960–6969, 2017.

[2] G.-D. Andreescu, C. I. Pitic, F. Blaabjerg, and I. Boldea. Combined flux observer with signal injection enhancement for wide speed range sensorless direct torque control of ipmsm drives. *IEEE Transactions on Energy Conversion*, 23(2):393–402, 2008.

[3] L. G. Barbosa Rolim, D. Rodrigues da Costa, and M. Aredes. Analysis and software implementation of a robust synchronizing pll circuit based on the pq theory. *IEEE Transactions on Industrial Electronics*, 53(6):1919–1926, 2006.

[4] F. Bauer and H. D. Heining. Quick response space vector control for a high power three-level-inverter drive system. *Archiv für Elektrotechnik*, 74(1):53–59, 1990.

[5] F. Blaschke. The Principle of Field Orientation as Applied to the New Transvector Closed Loop Control System for Rotating Field Machines. *Siemens Review*, 34:217–220, 1972.

[6] I. Boldea. Electric generators and motors: An overview. *CES Transactions on Electrical Machines and Systems*, 1(1):3–14, 2017.

[7] I. Boldea and S. C. Agarlita. The active flux concept for motion-sensorless unified AC drives: A review. In *International Aegean Conference on Electrical Machines and Power Electronics and Electromotion, Joint Conference*, pages 1–16, 2011.

[8] I. Boldea, M. C. Paicu, G. D. Andreescu, and F. Blaabjerg. “Active flux” orienta-

tion vector sensorless control of IPMSM. In *2008 11th International Conference on Optimization of Electrical and Electronic Equipment*, pages 161–168, 2008.

[9] M. Bolstad. Sensorless Control of Synchronous Machines Used in Adjustable Speed Hydro. Master's thesis, Department of Electric Power Engineering, NTNU – Norwegian University of Science and Technology, June 2018.

[10] Y. Chen and V. Dinavahi. Digital Hardware Emulation of Universal Machine and Universal Line Models for Real-Time Electromagnetic Transient Simulation. *IEEE Transactions on Industrial Electronics*, 59(2):1300–1309, 2012.

[11] M. Chinchilla, S. Arnaltes, and J. Burgos. Control of permanent-magnet generators applied to variable-speed wind-energy systems connected to the grid. *IEEE Transactions on Energy Conversion*, 21(1):130–135, 2006.

[12] R. S. Colby and D. W. Novotny. An Efficiency-Optimizing Permanent-Magnet Synchronous Motor Drive. *IEEE Transactions on Industry Applications*, 24(3):462–469, 1988.

[13] M. Depenbrock. Direct Self-Control (DSC) of Inverter-Fed Induction Machine. *IEEE Transactions on Power Electronics*, 3(4):420–429, 1988.

[14] C. Dufour, J. Belanger, S. Abourida, and V. Lapointe. FPGA-Based Real-Time Simulation of Finite-Element Analysis Permanent Magnet Synchronous Machine Drives. In *2007 IEEE Power Electronics Specialists Conference*, pages 909–915, 2007.

[15] M. Ehsani, K. Rahman, and H. Toliyat. Propulsion system design of electric and hybrid vehicles. *IEEE Transactions on Industrial Electronics*, 44(1):19–27, 1997.

[16] V. M. Fjellanger. On-Line Voltage Estimation During Sensorless Control of an Induction Machine Drive. Master's thesis, Department of Electric Power Engineering, NTNU – Norwegian University of Science and Technology, June 2019.

[17] B. Fossen. Modelling and Identification of Multi-Phase Machines. Project report in TET4520, Department of Electric Power Engineering, NTNU – Norwegian University of Science and Technology, December 2016.

[18] F. M. Gardner. *Phaselock Techniques*. John Wiley Sons, 2nd edition, 2005.

[19] J. Holtz. Sensorless control of induction machines—with or without signal injection? *IEEE Transactions on Industrial Electronics*, 53(1):7–30, 2006.

[20] L. I. Iepure, I. Boldea, and F. Blaabjerg. Hybrid I-f Starting and Observer-Based Sensorless Control of Single-Phase BLDC-PM Motor Drives. *IEEE Transactions on Industrial Electronics*, 59(9):3436–3444, 2012.

[21] Y. Inaba, S. Cense, T. O. Bachir, H. Yamashita, and C. Dufour. A Dual High-Speed PMSM Motor Drive Emulator With Finite Element Analysis on FPGA Chip With Full Fault Testing Capability. In *Proceedings of the 2011 14th European Conference on Power Electronics and Applications*, pages 1–10, 2011.

[22] International Energy Agency. *World Energy Outlook 2020*. 2020.

[23] A. Jack, B. Mecrow, and J. Haylock. A comparative study of permanent magnet and switched reluctance motors for high-performance fault-tolerant applications. *IEEE Transactions on Industry Applications*, 32(4):889–895, 1996.

[24] T. M. Jahns, G. B. Kliman, and T. W. Neumann. Interior Permanent-Magnet Synchronous Motors for Adjustable-Speed Drives. *IEEE Transactions on Industry Applications*, IA-22(4):738–747, 1986.

[25] P. Jansen and R. Lorenz. A physically insightful approach to the design and accuracy assessment of flux observers for field oriented induction machine drives. *IEEE Transactions on Industry Applications*, 30(1):101–110, 1994.

[26] I. N. Jensen. Control of an Ironless Permanent Magnet Synchronous Machine. Project report in TET4520, Department of Electric Power Engineering, NTNU – Norwegian University of Science and Technology, December 2020.

[27] Jun Hu and Bin Wu. New integration algorithms for estimating motor flux over a wide speed range. *IEEE Transactions on Power Electronics*, 13(5):969–977, 1998.

[28] S. Kim, J.-I. Ha, and S.-K. Sul. PWM Switching Frequency Signal Injection Sensorless Method in IPMSM. *IEEE Transactions on Industry Applications*, 48(5):1576–1587, 2012.

[29] K. P. Kovàcs and M. Kuzniarski. *Transiente Vorgänge In Wechselstrommaschinen*. Verlag der Ungarischen Akademie der Wissenschaften, 2nd edition, 1959.

[30] C. Lascu, I. Boldea, and F. Blaabjerg. Super-twisting sliding mode control of torque and flux in permanent magnet synchronous machine drives. In *IECON 2013 - 39th Annual Conference of the IEEE Industrial Electronics Society*, pages 3171–3176, 2013.

[31] M. Linke, R. Kennel, and J. Holtz. Sensorless speed and position control of synchronous machines using alternating carrier injection. In *IEEE International Electric Machines and Drives Conference, 2003. IEMDC'03.*, volume 2, pages 1211–1217 vol.2, 2003.

[32] J. Luukko. *Direct Torque Control of Permanent Magnet Synchronous Machines - Analysis and Implementation*. PhD thesis, Lappeenranta University of Technology, 2000.

[33] H. L. Mjell. Sensorless Control of Synchronous Machines Used in ASH. Master's thesis, Department of Electric Power Engineering, NTNU – Norwegian University of Science and Technology, June 2019.

[34] N. Mohan, T. M. Undeland, and W. P. Robbins. *Power Electronics: Converters, Applications, and Design*. John Wiley Sons, 2nd edition, 1995.

[35] A. Moldskred. FPGA-Based Real-Time Emulator of Permanent Magnet Synchronous Motor. Master's thesis, Department of Electric Power Engineering, NTNU – Norwegian University of Science and Technology, June 2020.

[36] M. Niemelä. *Position Sensorless Electrically Excited Synchronous Motor Drive for Industrial Use Based on Direct Flux Linkage and Torque Control*. PhD thesis, Lappeenranta University of Technology, 1999.

[37] R. Nilsen. *Electric Drives*. Norwegian University of Science and Technology, 2018.

[38] R. Nilsen. *Modeling, Identification and Control of Electrical Machines*. Norwegian University of Science and Technology, 2019.

[39] M. D. Omar Faruque, T. Strasser, G. Lauss, V. Jalili-Marandi, P. Forsyth, C. Dufour, V. Dinavahi, A. Monti, P. Kotsampopoulos, J. A. Martinez, K. Strunz, M. Saeedifard, X. Wang, D. Shearer, and M. Paolone. Real-Time Simulation Technologies for Power Systems Design, Testing, and Analysis. *IEEE Power and Energy Technology Systems Journal*, 2(2):63–73, 2015.

[40] M. C. Paicu, I. Boldea, G. . Andreescu, and F. Blaabjerg. Very Low Speed Performance of Active Flux Based Sensorless Control: Interior Permanent Magnet Synchronous Motor Vector Control Versus Direct Torque and Flux Control. *IET Electric Power Applications*, 3(6):551–561, 2009.

[41] M. C. Paicu, L. Tutelea, I. Boldea, G. D. Andreescu, and R. Ancuti. Pm-rsm sensorless vector control: Zero q-axis flux versus approximate maximum torque per current, with experiments. In *2010 12th International Conference on Optimization of Electrical and Electronic Equipment*, pages 460–468, 2010.

[42] A. Perera and R. Nilsen. A Sensorless Control Method for IPMSM with an Open-Loop Predictor for Online Parameter Identification. In *2020 23rd International Conference on Electrical Machines and Systems (ICEMS)*, pages 1983–1988, 2020.

[43] A. Perera and R. Nilsen. Gauss-Newton: A Prediction-Error-Gradient based Algorithm to Track PMSM Parameters Online. In *2020 IEEE International Conference on Power Electronics, Drives and Energy Systems (PEDES)*, pages 1–8, 2020.

[44] A. Perera and R. Nilsen. Full Speed Range Sensorless IPMSM Drive Enhanced with Online Parameter Identification. *IEEE International Electric Machines Drives Conference (IEMDC)*, page 6, 2021.

[45] A. Perera, R. Nilsen, T. Haugan, and K. Ljøkelsøy. A Design Method of an Embedded Real-Time Simulator for Electric Drives using Low-Cost System-on-Chip Platform. *PCIM Europe digital days, In Press*, page 8, 2021.

[46] P. D. C. Perera, F. Blaabjerg, J. K. Pedersen, and P. Thogersen. A Sensorless, Stable V/f Control Method for Permanent-Magnet Synchronous Motor Drives. *IEEE Transactions on Industry Applications*, 39(3):783–791, 2003.

[47] P. Pillay and R. Krishnan. Modeling, simulation, and analysis of permanent-magnet motor drives. i. the permanent-magnet synchronous motor drive. *IEEE Transactions on Industry Applications*, 25(2):265–273, 1989.

[48] I. Takahashi and T. Noguchi. A New Quick-Response and High-Efficiency Control Strategy of an Induction Motor. *IEEE Transactions on Industry Applications*, IA-22(5):820–827, 1986.

[49] P. Weichbold and T. V. Raumer. Influence of Dead Time Effects of PWM-VSI on Current Control. 2001.

[50] R. Wu and G. R. Slemon. A permanent magnet motor drive without a shaft sensor. *IEEE Transactions on Industry Applications*, 27(5):1005–1011, 1991.

[51] S.-C. Yang and Y.-L. Hsu. Full speed region sensorless drive of permanent-magnet machine combining saliency-based and back-emf-based drive. *IEEE Transactions on Industrial Electronics*, 64(2):1092–1101, 2017.

[52] Y.-D. Yoon, S.-K. Sul, S. Morimoto, and K. Ide. High-Bandwidth Sensorless Algorithm for AC Machines Based on Square-Wave-Type Voltage Injection. *IEEE Transactions on Industry Applications*, 47(3):1361–1370, 2011.

Appendix A

The Inductance Matrix

Since the stator flux path changes with rotor position in the IPMSM, the stator inductance matrix becomes a function of the rotor electrical position θ. The inductance matrix can be written as the sum of a leakage inductance matrix, a position independent inductance matrix and a position dependent matrix [38]. The complete stator inductance matrix becomes

$$\boldsymbol{L}_s^s = \begin{bmatrix} L_{s\sigma} + L_{sm} + L_g\cos(2\theta) & -\frac{1}{2}L_{sm} + L_g\cos(2\theta - \frac{2\pi}{3}) & \\ -\frac{1}{2}L_{sm} + L_g\cos(2\theta - \frac{2\pi}{3}) & L_{s\sigma} + L_{sm} + L_g\cos(2\theta + \frac{2\pi}{3}) \dots & \\ -\frac{1}{2}L_{sm} + L_g\cos(2\theta + \frac{2\pi}{3}) & -\frac{1}{2}L_{sm} + L_g\cos(2\theta) & \\ & -\frac{1}{2}L_{sm} + L_g\cos(2\theta + \frac{2\pi}{3}) & \\ \dots & -\frac{1}{2}L_{sm} + L_g\cos(2\theta) & \\ & L_{s\sigma} + L_{sm} + L_g\cos(2\theta - \frac{2\pi}{3}) & \end{bmatrix} \tag{A.1}$$

If the machine is non-salient, the air gap of the machine is constant for every value of θ. In that case, L_g will be equal to zero, and the inductance matrix becomes position independent. The inductance matrix is then

$$\boldsymbol{L}_s^s = \begin{bmatrix} L_{s\sigma} + L_{sm} & -\frac{1}{2}L_{sm} & -\frac{1}{2}L_{sm} \\ -\frac{1}{2}L_{sm} & L_{s\sigma} + L_{sm} & -\frac{1}{2}L_{sm} \\ -\frac{1}{2}L_{sm} & -\frac{1}{2}L_{sm} & L_{s\sigma} + L_{sm} \end{bmatrix} \tag{A.2}$$

Appendix B

Motor Drive Data

	Symbol	Value	Unit
Nominal Voltage	U_n	220	V
Nominal Current	I_n	51	A
Nominal Frequency	f_n	35	Hz
Pole pairs	p	1	-
Nominal Speed	N_n	2100	rpm
Switching Frequency	f_s	4000	Hz
Stator resistance	r_s	0.009	per-unit
d-axis Reactance	x_d	0.4	per-unit
q-axis Reactance	x_q	1.0	per-unit
Permanent Magnet Flux Linkage	ψ_m	0.66	per-unit

Appendix C

Base-Values Used for Per-Unit Conversion

$$U_{basis} = \sqrt{\frac{2}{3}} U_n$$

$$I_{basis} = \sqrt{2} I_n$$

$$\Psi_{basis} = \frac{U_{basis}}{2\pi f_n}$$

$$S_{basis} = \sqrt{3} U_n I_n$$

$$T_{basis} = \frac{S_{basis}}{N_n \cdot \frac{\pi}{30}}$$

Appendix D

Scripts Used for Flux Model Sensitivity Analysis

In the below scripts, the maximum-torque-per-ampere control strategy is implemented by assuming current references according to Equation 2.26 and Equation 2.27. The scripts calculate and compare actual stator flux with estimated flux for speeds and torque references between -1 and 1 per unit. Flux estimation is hence only performed in the maximum-torque-per-ampere region, meaning that the flux models' performance during field weakening operation has not been considered.

D.1 MATLAB Script for Voltage Model Sensitivity Analysis

```
clc
clear all

% Defining motor parameters
x_q = 1;
x_d = 0.4;
r_s = 0.009;
psi_m = 0.66;

% Performing calculations for torque references between -1 and 1 pu and
% speeds between -1 and 1 pu
t =[-1:0.05:1];
n_vec =[-1:0.05:1];
```

```
[N,T]= meshgrid(n_vec,t);
for k=1:length(t)
    for l =1:length(n_vec)
        tau_e=t(k);
        n = n_vec(l);

        % Currents that give MTPA
        i_d = ( (psi_m/3) - ( (psi_m/3)^3 + ( (x_q - x_d)^2 * tau_e^2 ) / (3*psi_m) )^(1/3) ) / ( x_q - x_d );
        i_q = tau_e / ( psi_m - (x_q - x_d)*i_d);

        % Calculate real dq fluxes
        psi_d = x_d*i_d + psi_m;
        psi_q = x_q*i_q;

        % Calculate dq stator voltages in steady-state
        u_d = r_s*i_d - n*psi_q;
        u_q = r_s*i_q + n*psi_d;

        % Looking at the instant the rotor is aligned with the a-axis (theta is zero)
        % Transforming into alpha and beta components
        psi_a = psi_d;
        psi_b = psi_q;
        psi_s_s = psi_a +j*psi_b;

        i_a = i_d;
        i_b = i_q;
        i_s_s = i_a + j*i_b;

        u_a = u_d;
        u_b = u_q;
        u_s_s = u_a + j*u_b;

        % Calculate the real active flux vector
        psi_T_a = psi_a - i_a*x_q;
        psi_T_b = psi_b - i_b*x_q;

        % Calculate the real rotor position
        theta = atan2(psi_T_b, psi_T_a);

        % Resistance erroneously estimated
        r_s_est = 0.8*r_s;

        % Voltage erroneously estimated
        u_s_s_est = u_s_s; % + 0.005;

        % Estimate flux using voltage model
```

```
        psi_s_s_est = (u_s_s_est - (r_s_est*i_s_s))/(j*n);

        psi_a_est = real(psi_s_s_est);
        psi_b_est = imag(psi_s_s_est);

        psi_d_est = psi_a_est;
        psi_q_est = psi_b_est;

        % Estimate active flux
        psi_T_a_est = psi_a_est - i_a*x_q;
        psi_T_b_est = psi_b_est - i_b*x_q;

        % Estimate the rotor position
        theta_est = atan2(psi_T_b_est, psi_T_a_est);

        % Theta error
        theta_error(k,l) = theta_est*(180/pi);
        if n==0
            theta_error(k,l) =0/0;
        end

        % Flux error
        flux_mag_error(k,l) = abs(psi_s_s) - abs(psi_s_s_est);
        flux_angle_error(k,l) = angle(psi_s_s)*(180/pi) - angle(
    psi_s_s_est)*(180/pi);
        if n==0
            flux_mag_error(k,l) = 0/0;
            flux_angle_error(k,l) =0/0;
        end

        % Torque error
        torque_error(k,l) = ((psi_d-psi_d_est)*i_q - (psi_q - psi_q_est)*
    i_d);
    end
end

figure(1)
surf(N,T,flux_mag_error)
colorbar
xlabel('Speed [pu]')
ylabel('Torque Reference [pu]')
zlabel ('Flux Amplitude Error [pu]')

figure(2)
surf(N,T,flux_angle_error)
colorbar
xlabel('Speed [pu]')
ylabel('Torque Reference [pu]')
```

```
zlabel ('Flux Angle Error [degrees]')

figure(3)
surf(N,T,torque_error)
colorbar
xlabel('Speed [pu]')
ylabel('Torque Reference [pu]')
zlabel ('Torque Error [pu]')

figure(4)
surf(N,T,theta_error)
colorbar
xlabel('Speed [pu]')
ylabel('Torque Reference [pu]')
zlabel (['Theta Error [degrees]'])

% NB: Field weakening is not considered.
% No limits have been considered (limit current and voltage to within
% maximum values
```

Listing D.1: Script for voltage model sensitivity analysis.

D.2 MATLAB Script for Current Model Sensitivity Analysis

```
clc
clear all

% Defining motor parameters
x_q = 1.0;
x_d = 0.4;
r_s = 0.009;
psi_m = 0.66;

% Performing calculations for torque references between -1 and 1 pu and
% speeds between -1 and 1 pu
t =[-1:0.05:1];
n_vec =[-1:0.05:1];
[N,T]= meshgrid(n_vec,t);
for k=1:length(t)
    for l =1:length(n_vec)
        tau_e=t(k);
        n = n_vec(l);

        % Currents that give MTPA
```

```
        i_d = ( (psi_m/3) - ( (psi_m/3)^3 + ( (x_q - x_d)^2 * tau_e^2 ) /
    (3*psi_m) )^(1/3) ) / ( x_q - x_d );
        i_q = tau_e / ( psi_m - (x_q - x_d)*i_d);

        % Calculate real dq fluxes
        psi_d = x_d*i_d + psi_m;
        psi_q = x_q*i_q;

        % Calculate dq stator voltages in steady-state
        u_d = r_s*i_d - n*psi_q;
        u_q = r_s*i_q + n*psi_d;

        % Looking at the instant the rotor is aligned with the a-axis
        % (theta is zero)
        % Transforming into alpha and beta components
        psi_a = psi_d;
        psi_b = psi_q;
        psi_s_s = psi_a +j*psi_b;

        i_a = i_d;
        i_b = i_q;
        i_s_s = i_a + j*i_b;

        u_a = u_d;
        u_b = u_q;
        u_s_s = u_a + j*u_b;

        % Calculate the real active flux vector
        psi_T_a = psi_a - i_a*x_q;
        psi_T_b = psi_b - i_b*x_q;

        % Calculate the real rotor position
        theta = atan2(psi_T_b, psi_T_a);

        % Reactance estimates
        x_d_est = 1.0*x_d;
        x_q_est = 1.0*x_q;

        % Permanent magnet flux linkage estimate
        psi_m_est = 0.8*psi_m;

        % Estimate flux using current model
        psi_d_est = x_d_est*i_d + psi_m_est;
        psi_q_est = x_q_est*i_q;

        psi_a_est = psi_d_est;
        psi_b_est = psi_q_est;

```

```
        psi_s_s_est = psi_a_est + j*psi_b_est;

        % Estimate active flux
        psi_T_a_est = psi_a_est - i_a*x_q;
        psi_T_b_est = psi_b_est - i_b*x_q;

        % Estimate the rotor position
        theta_est = atan2(psi_T_b_est, psi_T_a_est);

        % Theta error
        theta_error(k,l) = theta_est*(180/pi);

        % Flux error
        flux_mag_error(k,l) = abs(psi_s_s) - abs(psi_s_s_est);
        flux_angle_error(k,l) = angle(psi_s_s)*(180/pi) - angle(
    psi_s_s_est)*(180/pi);

        % Torque error
        torque_error(k,l) = ((psi_d-psi_d_est)*i_q - (psi_q - psi_q_est)*
    i_d);
    end
end

figure(1)
surf(N,T,flux_mag_error)
colorbar
xlabel('Speed [pu]')
ylabel('Torque Reference [pu]')
zlabel ('Flux Amplitude Error [pu]')

figure(2)
surf(N,T,flux_angle_error)
colorbar
xlabel('Speed [pu]')
ylabel('Torque Reference [pu]')
zlabel ('Flux Angle Error [degrees]')

figure(3)
surf(N,T,torque_error)
colorbar
xlabel('Speed [pu]')
ylabel('Torque Reference [pu]')
zlabel ('Torque Error [pu]')

figure(4)
surf(N,T,theta_error)
colorbar
xlabel('Speed [pu]')
```

```
114 ylabel('Torque Reference [pu]')
115 zlabel ('Theta Error [degrees]')
116
117 % NB: Field weakening is not considered.
118 % No limits have been considered (limit current and voltage to within
119 % maximum values
```

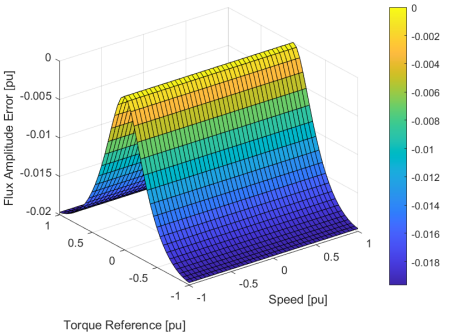
Listing D.2: Script for current model sensitivity analysis.

Appendix E

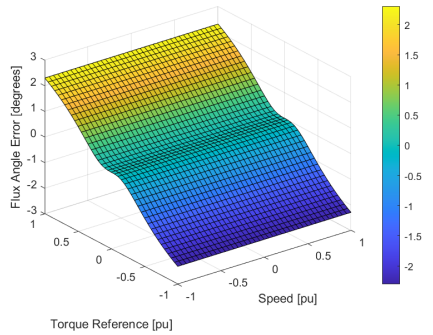
Current Model Parameter Sensitivity Analysis Results

In the following sections, the impact of underestimated motor parameter values on the current model flux estimate and the torque output of the motor is demonstrated.

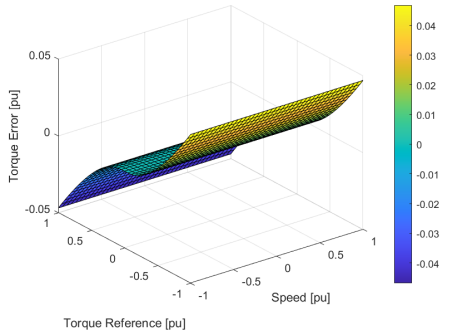
E.1 Impact of $\hat{x}_d = 0.8x_d$



(a) Amplitude error of flux estimate space vector.



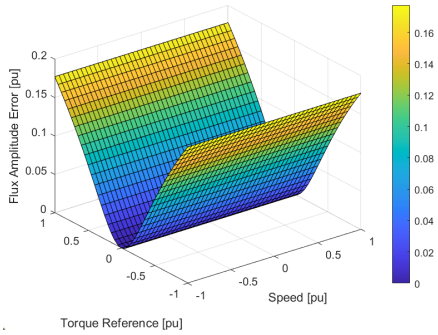
(b) Angle error of flux estimate space vector.



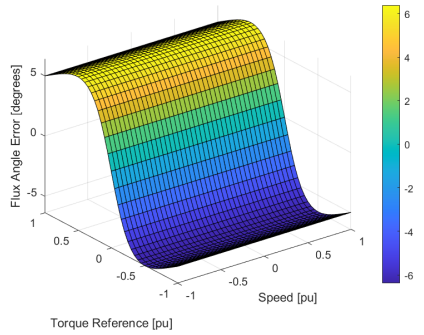
(c) Error in the torque output of the motor.

Figure E.1: Error in flux estimate magnitude and angle and output torque due to an underestimated d-axis reactance.

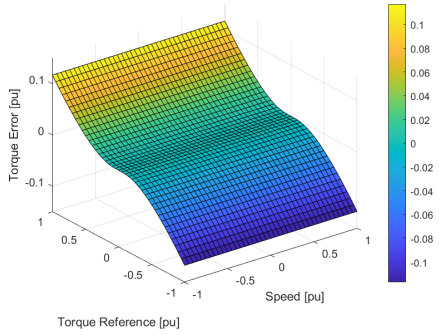
E.2 Impact of $\hat{x}_q = 0.8x_q$



(a) Amplitude error of flux estimate space vector.



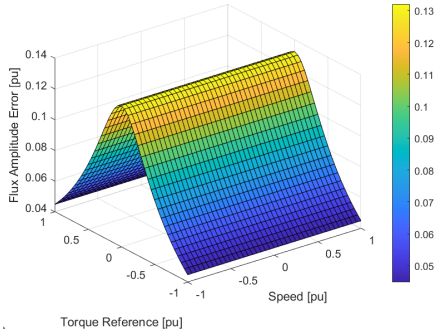
(b) Angle error of flux estimate space vector.



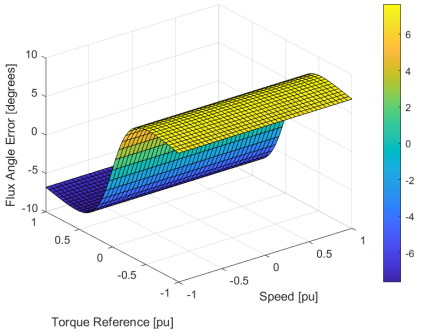
(c) Error in the torque output of the motor.

Figure E.2: Error in flux estimate magnitude and angle and output torque due to an underestimated q-axis reactance.

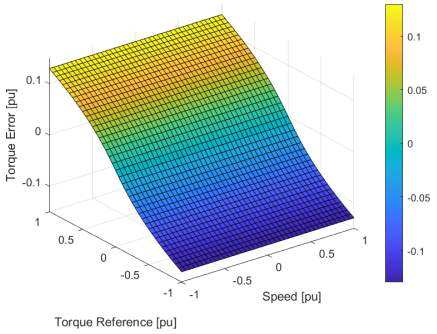
E.3 Impact of $\hat{\psi}_m = 0.8\psi_m$



(a) Amplitude error of flux estimate space vector.



(b) Angle error of flux estimate space vector.



(c) Error in the torque output of the motor.

Figure E.3: Error in flux estimate magnitude and angle and output torque due to an underestimated permanent magnet flux linkage.

Appendix F

Phase-Locked Loop Filtering of the Position Estimate

As explained by Gardner in [18], the phase-locked loop consists of three basic elements, namely a phase detector (PD), a loop filter (LF) and a voltage-controlled oscillator (VCO). The phase detector compares the phase of the input signal against the phase of the VCO signal, and produces an error signal that is passed to the loop filter. The output of the loop filter acts as a control voltage that changes the frequency of the VCO, and hence serves to reduce the phase difference between the input signal and the VCO signal.

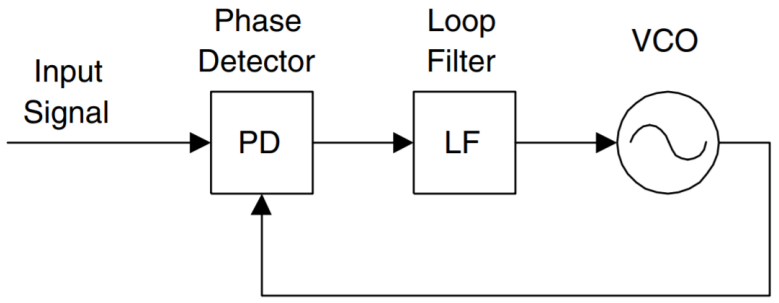


Figure F.1: General structure of a phase-locked loop [18].

If the input to the PLL is a noisy signal, the PLL will act as a filter that rejects the noise in the input signal and passes a cleaned-up version of the signal to its output. By pass-

ing the estimated rotor position angle through a well-designed PLL, it should be possible the eliminate any potential noise in $\hat{\theta}$, hence providing a better rotor position estimate. The specific PLL structure that will be used here for filtering of the position estimate is presented in Figure F.2. Here, the loop filter is a PI controller.

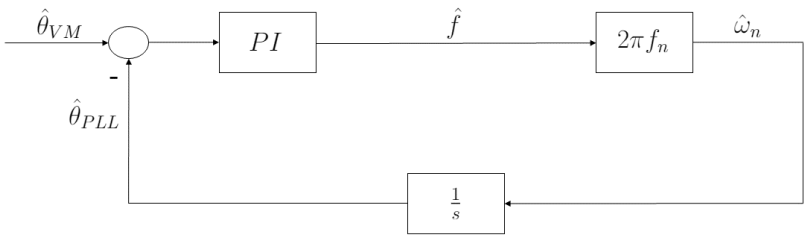


Figure F.2: Structure of the PI-controller based phase-locked loop. Reproduced from [9].

Using $\omega_n = 2\pi f_n$, the transfer function of the phase-locked loop can be written as

$$H_{PLL}(s) = \frac{k_p \omega_n s + k_i \omega_n}{s^2 + k_p \omega_n s + k_i \omega_n} \tag{F.1}$$

The transfer function in Equation F.1 can be rewritten on the following general form:

$$H_{PLL}(s) = \frac{2\zeta\omega_0 s + \omega_0^2}{s^2 + 2\zeta\omega_0 s + \omega_0^2} \tag{F.2}$$

Here, ω_0 is the undamped angular frequency of the system and ζ is the relative damping ratio. Generally, the magnitude of ω_0 will determine the speed of the system response, while the value of ζ determines whether the system has overshoot and an oscillatory response or not. Together, knowledge about the values of ω_0 and ζ will make it possible to predict the characteristics of the response of the closed-loop system. By comparing Equation F.1 and Equation F.2, the PI controller gains can be expressed as functions of ω_0 and ζ as shown below.

$$k_p = 2\zeta\frac{\omega_0}{\omega_n} \tag{F.3}$$

$$k_i = \frac{\omega_0^2}{\omega_n} \tag{F.4}$$

Now, the controller can be tuned by determining which values of these parameters that will give the desired response. As explained in [3], a robust PLL design is achieved if the PI controller is tuned to meet the following criterion:

$$\zeta = \frac{1}{\sqrt{2}} \approx 0.707 \tag{F.5}$$

Next, choosing the value of ω_0 becomes a trade-off between having a fast dynamic response and narrow bandwith for good filtering performance and noise rejection.

The implementation of the PLL for rotor position filtering did not work as intended, suggesting that there are errors in the PLL C++ sub-routine. The PLL actually produced a position estimate that is more polluted by noise than the original position estimate, and did as such have the opposite effect of what was intended.

Appendix G

High-Frequency Signal Injection for Standstill and Low-Speed Operation

Due to the inherent saliency of the rotor of the IPMSM, the inductances of the flux path seen by the stator in the IPMSM will vary with the rotor position. This can be utilized by adding a high-frequency voltage signal to the voltage reference of the converter and measuring the corresponding response in the stator current. The ripple in the stator current due to the high-frequency component in the stator voltage will be modulated with information about the rotor position, and by applying the correct demodulation technique, the position of the rotor can be obtained [28].

The method that was attempted in this thesis is presented in [44], and is based on the work done by [52] and [28]. Here, a short introduction to the estimation scheme is presented.

The square-wave voltage that will be injected in the stator is given by Equation G.1:

$$\begin{bmatrix} u_{\hat{d}h} \\ u_{\hat{d}h} \end{bmatrix} = \begin{bmatrix} u_h(-1)^k \\ 0 \end{bmatrix} \tag{G.1}$$

In the above equation, k is used to denote the number of interrupt signals, which are generated at the top and bottom of the triangular PWM carrier signal. The result is that

a square-wave voltage with a frequency equal the switching frequency and a magnitude equal to u_h is generated and injected in the estimated d-axis, denoted as $\hat{d}$.

The $\hat{q}$-axis current response to the high-frequency voltage component can be approximated using

$$\delta i_{\hat{q}h} \approx \delta i_{\hat{q}} = i_{\hat{q}}[k] - i_{\hat{q}}[k-1] \tag{G.2}$$

This approximation assumes that the current response due to the fundamental excitation is constant withing a sampling period interval. The $\hat{q}$-axis current response can be used to obtain an estimation of the error in the current rotor position estimate, $\tilde{\theta}$, as shown in Equation G.3.

$$\tilde{\theta} = \frac{x_d \cdot x_q}{(x_q - x_d) \cdot u_{\hat{d}h}[k] \cdot \omega_n \cdot T_{samp}} \cdot \delta i_{\hat{q}} \tag{G.3}$$

If this error is driven to zero using a phase-locked loop, as proposed by [44], the position estimate coming from the saliency-tracker will correspond to the true rotor position. This approach was attempted to extract the rotor position extraction, but the method was not successfully implemented.

Appendix H

C++ Control Software

In the following pages, a condensed version of the control software that is implemented in this thesis is presented. Only the most relevant subroutines are included.

H.1 Voltage Estimation

```
void Converter::estimateVoltage(){
  T_blank = 0.0; //0.05*(2*Tsamp); // 5% of switching period. According to
     Gullvik: 0.00000178
  U_PNI = 0.0; //0.3; // From Gullvik
  U_PNII = 0.0; //0.39; // From  Gullvik
  r_T = 0.0; //0.018; // From  Gullvik

  I_sa_mag = abs(m_measured_values.I_sa_A);
  I_sb_mag = abs(m_measured_values.I_sb_A);
  I_sc_mag = abs(m_measured_values.I_sc_A);

  // Finding alpha and beta components of reference voltages
  U_sa_ref = m_Us_peak_V*cos(m_delta_s_s);
  U_sb_ref = m_Us_peak_V*sin(m_delta_s_s);

  // Transforming to three-phase quantities
  U_san_ideal = U_sa_ref;
  U_sbn_ideal = -0.5*U_sa_ref + 0.866*U_sb_ref; // (sqrt(3))/2 = 0.866
  U_scn_ideal = -0.5*U_sa_ref - 0.866*U_sb_ref; // (sqrt(3))/2 = 0.866

  // Phase A
  if (m_measured_values.I_sa_A >= 0) {
      delta_U_DT_a  = -(T_blank/(2*Tsamp))*m_measured_values.U_dc_V;
```

```
  }
  else if (m_measured_values.I_sa_A < 0) {
    delta_U_DT_a = (T_blank/(2*Tsamp))*m_measured_values.U_dc_V;
  }

  // Phase B
  if (m_measured_values.I_sb_A > 0) {
    delta_U_DT_b = -(T_blank/(2*Tsamp))*m_measured_values.U_dc_V;
  }
  else if (m_measured_values.I_sb_A < 0) {
    delta_U_DT_b = (T_blank/(2*Tsamp))*m_measured_values.U_dc_V;
  }

  // Phase C
  if (m_measured_values.I_sc_A > 0) {
      delta_U_DT_c = -(T_blank/(2*Tsamp))*m_measured_values.U_dc_V;
  }
  else if (m_measured_values.I_sc_A < 0) {
      delta_U_DT_c = (T_blank/(2*Tsamp))*m_measured_values.U_dc_V;
  }

  // Output per-unit voltage estimate
  U_sa_est = U_san_ideal + delta_U_DT_a;
  U_sb_est = U_sbn_ideal + delta_U_DT_b;
  U_sc_est = U_scn_ideal + delta_U_DT_c;

  // Alpha-beta components of the estimated voltage
  U_salpha_est = 0.667*U_sa_est - 0.333*U_sb_est - 0.333*U_sc_est;
  U_sbeta_est = 0.577*U_sb_est - 0.577*U_sc_est;
```

Listing H.1: Stator voltage estimation code.

H.2 The Current Model

```
void MotorModel::CurrentModel()
{
  // calculation of stator flux in rotating frame
  m_psi_s_di = m_id*par.xd + par.psi_m;
  m_psi_s_qi = m_iq*par.xq;

  // stator flux transformation to stator frame for comparison with them
    of voltage mode
  m_psi_s_sai = m_psi_s_di*cos(theta_used) - m_psi_s_qi*sin(theta_used);
  m_psi_s_sbi = m_psi_s_di*sin(theta_used) + m_psi_s_qi*cos(theta_used);

```

```
}
```

Listing H.2: The current model.

H.3 The Voltage Model

```
void MotorModel::VoltageModel()
{
  if (runMRAS != 1) {
    // Implementing the voltage model for stator flux linkage estimation.
    Using the Niemel  method for drift compensation.

      // A-priori flux estimate (Isak, modified)
      m_psi_s_sau = m_psi_sa_old + Tsamp*par.wn* (m_u_s_sa - 0.8*par.rs*
    m_i_s_sa_avg); // Assuming resistance is incorrectly estimated
      m_psi_s_sbu = m_psi_sb_old + Tsamp*par.wn* (m_u_s_sb - 0.8*par.rs*
    m_i_s_sb_avg); // Assuming resistance is incorrectly estimated


      // Generate flux estimate correction terms
      //SimpleNiemelaCorrection();
      if (withNiemelaCorrection == 1) {
        NiemelaCorrection();
      }

      // A-posteriori flux estimate
      m_psi_s_sau = m_psi_s_sau + m_psi_s_sau_corr;
      m_psi_s_sbu = m_psi_s_sbu + m_psi_s_sbu_corr;

      m_psi_s_su_mag = sqrt(m_psi_s_sau*m_psi_s_sau + m_psi_s_sbu*
    m_psi_s_sbu);
      m_psi_s_s_angle = atan2(m_psi_s_sbu, m_psi_s_sau);

      // updating the old value, Euler discretization
      m_psi_sa_old = m_psi_s_sau;
      m_psi_sb_old = m_psi_s_sbu;
  }

  if (runMRAS == 1) {

    // Generate compensation voltages
    m_u_comp_a = m_PI_alpha.Run(m_psi_s_sai, m_psi_s_sau, 0, -1.0, 1.0);
    m_u_comp_b = m_PI_alpha.Run(m_psi_s_sbi, m_psi_s_sbu, 0, -1.0, 1.0);


    // Flux estimate from voltage model
```

```
    m_psi_s_sau = m_psi_sa_old + Tsamp*par.wn* (m_u_s_sa - 0.8*par.rs*
    m_i_s_sa_avg + m_u_comp_a);
    m_psi_s_sbu = m_psi_sb_old + Tsamp*par.wn* (m_u_s_sb - 0.8*par.rs*
    m_i_s_sb_avg + m_u_comp_b);

    // updating the old value, Euler discretization
    m_psi_sa_old = m_psi_s_sau;
    m_psi_sb_old = m_psi_s_sbu;
  }
```

Listing H.3: The voltage model.

H.4 Niemelä-Correction

```
void MotorModel::NiemelaCorrection()
{
  // Torque estimation
  m_torque_est = (par.psi_m - (par.xq - par.xd) * m_id) * m_iq;

  // Removing noise in torque estimate
  m_torque_est = m_LPF_tau_est2.Run(m_torque_est);

  // Filtering of the torque estimate
  m_torque_est_filt = m_LPF_tau_est.Run(m_torque_est);

  // Assigning value to k_T
  m_k_T_0 = 4;
  if (abs(k_T_0*(m_torque_est - m_torque_est_filt)) < 1) {
    m_k_T = abs(k_T_0*(m_torque_est - m_torque_est_filt));
  }
  else {
    m_k_T = 1;
  }

  // Assigning correction coefficient value
  m_k_psi_0 = 0.0075;
  m_k_psi_corr = (1 - k_T)*k_psi_0;

  // Limits low-pass filter time constant to a maximum value of 1.75.
  if ((2 / (abs(speed_est) * fn)) < 1.75 && (2 / (abs(speed_est) * fn)) >=
     0) {
    m_Tfilt = 2 / (abs(speed_est) * fn);
  }
  else {
    m_Tfilt = 1.75;
  }
```

```
32
33    // A-priori flux linkage estimate magnitude squared
34    m_psi_s_su_sqr = m_psi_s_sau*m_psi_s_sau + m_psi_s_sbu*m_psi_s_sbu;
35
36    // Modified low-pass filter
37    m_psi_s_su_sqr_filt = m_psi_s_su_sqr_filt + (Tsamp / Tfilt)*(
        m_psi_s_su_sqr - m_psi_s_su_sqr_filt) +
38        k_T * (m_psi_s_su_sqr - m_psi_s_su_sqr_old);
39
40    // Updating old value of the the flux estimate squared
41    m_psi_s_su_sqr_old = m_psi_s_su_sqr;
42
43    // Error
44    m_epsilon = m_psi_s_su_sqr_filt - m_psi_s_su_sqr;
45
46    // Generating correction terms
47    m_psi_s_sau_corr = m_k_psi_corr*m_epsilon*m_psi_s_sau;
48    m_psi_s_sbu_corr = m_k_psi_corr*m_epsilon*m_psi_s_sbu;
49
50  }
```

Listing H.4: The Niemelä-method for drift correction.

H.5 Position Estimation

```
1  void MotorModel::PositionEstimator()
2  {
3    // Active flux calculation
4    m_psi_Ta =  m_psi_s_sau - par.xq*m_i_s_sa_avg;
5    m_psi_Tb =  m_psi_s_sbu - par.xq*m_i_s_sb_avg;
6
7    // Position estimate
8    theta_est = atan2(m_psi_Tb, m_psi_Ta);
9
10   // Per-unit speed estimate
11   help1 = m_psi_Tb * m_psi_Ta_old - m_psi_Ta * m_psi_Tb_old;
12   help2 = 1/((m_psi_Ta*m_psi_Ta + m_psi_Tb*m_psi_Tb)*Tsamp*par.wn);
13   speed_est = help1*help2;
14
15     // Low-pass filtering of the speed estimate
16   m_speed_est = m_LPF_n_est.Run(speed_est);
17
18   m_psi_Ta_old = m_psi_Ta;
19   m_psi_Tb_old = m_psi_Tb;
20 }
```

Listing H.5: Rotor position estimation code.

NTNU

Norwegian University of
Science and Technology