UNIVERSITETET I AGDER

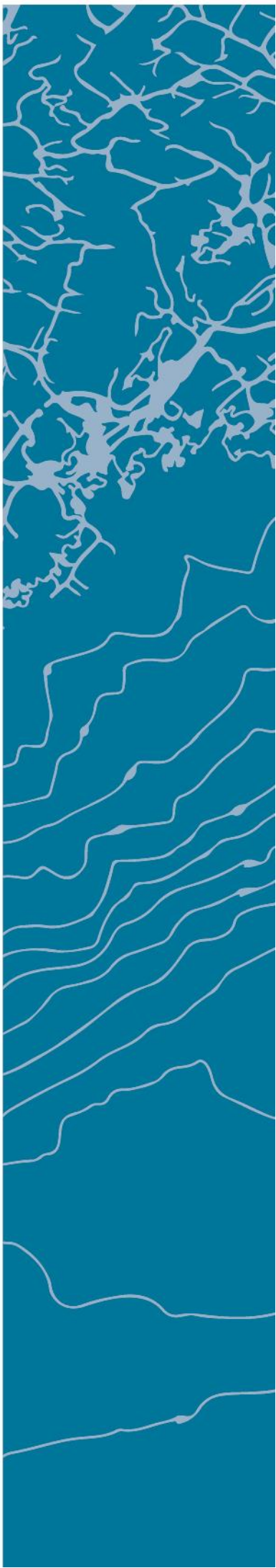
Sensorless Control of Induction Motors

Using an Extended Kalman Filter and Linear Quadratic Tracking

DAVID HÅLAND

SUPERVISORS
Van Khang Huynh
Surya Teja Kandukturi

University of Agder, 2017
Faculty of Engineering and Science
Department of Engineering and Science



Acknowledgements

I would like to give an special thank to Surya Teja Kandukuri for the all the help, non ending endurance and good solutions he has given me; even under my constant questions and emails. I would also like to thank Professor van Khang Huynh for all the help, motivation, extra office lectures and opportunities he has given me.

Lastly, I would like to thank my fiance Maylinn Nøyseth for all the coffee, pizza and patience she have provided in our small flat of 31 m^2 over the past months.

Preface

All internal references for figures, tables, sections and chapters for this report are hyper-linked and is recommended to use to improve the reading experience.

Microsoft Edge is not recommended to use for reading this report, duo to its failure in hyper-linking internal references.

All citing is hyper-linked and coloured in blue.

All figures that are not directly cited and some that are, is made in Latex Tikz, MS Power Point or MS Visio; they are also exported in vector graphics to improve visibility and zooming capability.

Abstract

Conventional sensorless controls for induction motors require two PID regulators and precise gain turning. This thesis presents a sensorless control for induction motors using an extended Kalman filter (EKF) and linear quadratic tracking (LQT). The proposed method requires only a single controller and no physical velocity sensor.

Contents

List of Symbols

- v_{qs}: q-axis stator voltage.
- v_{ds}: d-axis stator voltage.
- v'_{qr}: q-axis rotor voltage referred to stator.
- v'_{dr}: d-axis rotor voltage referred to stator.
- i_{qs}: q-axis stator current.
- i_{ds}: d-axis stator current.
- i'_{qr}: q-axis rotor current referred to stator.
- i'_{dr}: d-axis rotor current referred to stator.
- X_{qs}: q-axis stator reactance.
- X_{ds}: d-axis stator reactance.
- X_{qr}: q-axis rotor reactance.
- X_{dr}: d-axis rotor reactance.
- X_{mq}: q-axis Mutual reactance.
- X_{md}: d-axis Mutual reactance.
- L_{ls}: Leakage inductance for stator.
- L'_{lr}: Leakage inductance for rotor referred to stator.
- L_s: Self inductance for stator.
- L_r: Self inductance for rotor.
- L_{sm}: Mutual inductance for stator.
- L_{rm}: Mutual inductance for rotor.
- L_{ms}: Magnetizing inductance.

- r_s: Stator resistance.
- r_r: Rotor resistance.
- λ_{qs}: q-axis stator flux linkage.
- λ_{ds}: d-axis stator flux linkage.
- λ_{qr}: q-axis rotor flux linkage.
- λ_{dr}: d-axis rotor flux linkage.
- Ψ_{qs}: q-axis stator flux.
- Ψ_{ds}: d-axis stator flux.
- Ψ_{qr}: q-axis rotor flux.
- Ψ_{dr}: q-axis rotor flux.
- θ: Displacement angle of reference frame.
- θ_r: Displacement angle of rotor.
- ω_r: Angular electric velocity of rotor.
- ω: Angular velocity of reference frame.
- ω_m: Angular mechanical velocity of rotor.
- ω_{err}: Error between reference signal and rotor velocity.
- ω_{ref}: Reference velocity of rotor.
- ω_{rc}: Control velocity.
- T_{em}: Electromagnetic torque.
- T_l: Load torque.
- P: Number of poles.
- p: Number of pole pairs.
- J: Rotor inertia.
- x_n: States.
- $\hat{x}$: State estimate.
- $\hat{x}_n^-$: A priori state estimate.
- $\hat{x}_n$: A posteriori state estimate.
- $\hat{T}_l$: Estimated load torque.

- Q_{EKF}: Process noise weights EKF.
- R_{EKF}: Measurement noise weights EKF.
- Q_{LQT}: State weights LQT.
- R_{LQT}: Input weights LQT.
- P_{rated}: Rated power.
- f_{rated}: Rated frequency.
- V_{rated}: Rated voltage.
- I_{rated}: Rated current.
- n_{rated}: Rated velocity.
- τ_{rated}: Rated load torque.

List of Figures

Chapter 1

Introduction

Two PID regulators for speed (or torque) and current controllers are required for conventional motor drives [1]. The performance of the drives depends on PID gains of these controllers. Linear quadratic tracking(LQT) can be used to control electric motors, since this technique is suited for optimal control on nonlinear systems [2]. The gains of the regulator are simply tuned, the proposed method does however have more tuning options than this; this will be explained and shown in this thesis. The objective of this work is to apply LQT(linear quadratic tracking) for sensorless control of induction motors. The sensorless control is developed based on an EKF(extended Kalman filter), which is often employed as an estimation tool for induction motors [1].

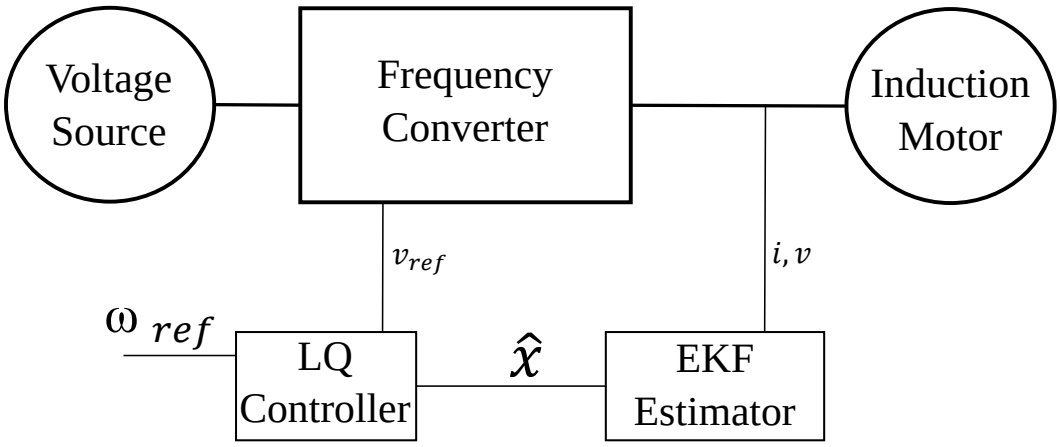


Figure 1.1: Control Scheme.

The overall control structure is shown in Figure 1.1. The LQT get a full state feedback

provided by the EKF. The EKF are get a measurement of stator currents and an input of stator voltages; the EKF then estimates all the required states for this induction motor model.

$$\hat{x} = \Big[\Psi_{qs},\ \Psi_{ds},\ \Psi_{qr},\ \Psi_{dr},\ \omega_r\Big]^T \tag{1.1}$$

$$u = \Big[v_{qs},\ v_{ds},\ T_l]^T \tag{1.2}$$

$$y = \Big[i_{qs},\ i_{ds}]^T \tag{1.3}$$

Estimated states, inputs and measurements are shown in equation 1.1, 1.2 and 1.3, respectively.

The LQT generates voltages based on current state values and the reference signal(r_c) it is given, this is shown in Figure 1.2. The LQT will allways try to drive the error $e = x - r_c$ to zero. For this model the r_c signal will only include a velocity reference signal.

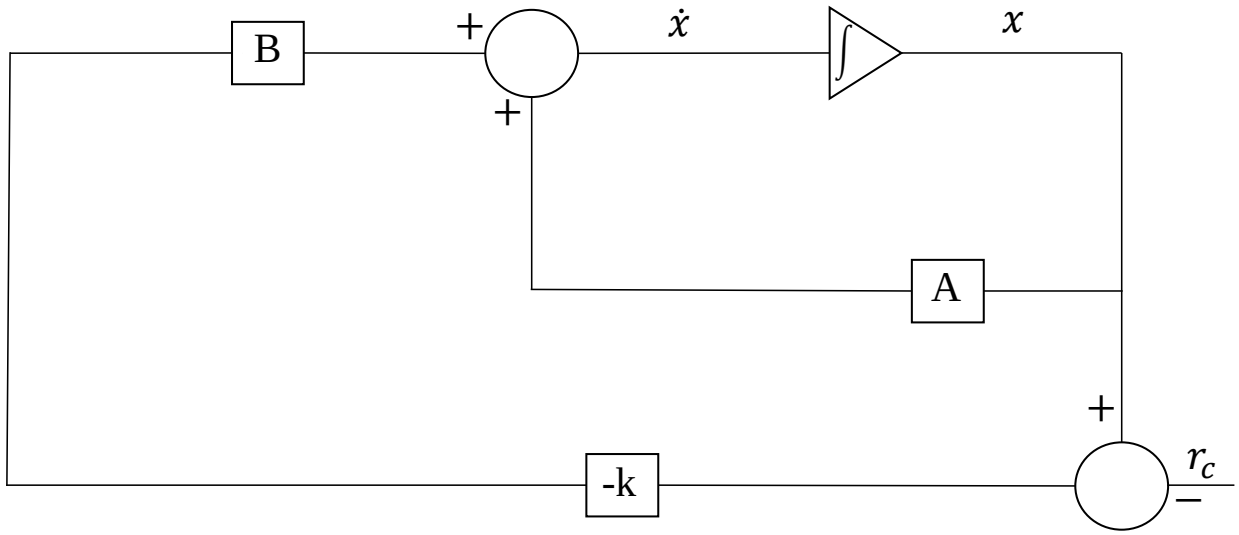


Figure 1.2: Linear Quadratic Tracker.

This thesis will show and explain how this control structure work on a simulated platform using Matlab and Simulink [1]. The estimator will also be tested with real data produced by an physical motor placed in the mechatronics lab in the University of Agder(UIA).

[1]Matlab is a numeric computing tool, Simulink is and extension of this, designed for realistic simulation.

1.1 Problem Statement

1 Successfully implement velocity control for an induction motor in Matlab/Simulink using a Kalman filter and a LQ-optimal controller.

2 Comparing performance in velocity control between EKF-based control method to a PID based control.

1.2 Motivation

The motivation for this project is to get better insight in the subjects listed below:

- State space modelling of non linear systems.
- Kalman Filter estimation.
- Optimal control.
- Induction motor velocity control.
- Greater knowledge and experience using Matlab and Simulink.

1.3 Report Outline

Below is a brief description of the remaining chapters for this report.

Chapter 1 Introduction to this thesis.

Chapter 2 Theoretical subjects used in this thesis.

Chapter 3 Control structure and components used for this thesis.

Chapter 4 Digital induction motor model.

Chapter 5 Results.

Chapter 6 Discussion.

Chapter 7 Conclusion.

Bibliography Reference literature.

Appendix A Matlab EKF filter script.

Appendix B Matlab LQT filter script.

Appendix C Current and voltage plots.

Appendix D Current and voltage plots.

Appendix E ABB catalogue.

Appendix F Torque and velocity plots.

Appendix G Current plots.

Appendix H Motor nameplate and set-up.

Chapter 2

Theoretical Background

In this chapter the theoretical background for this thesis will be explained. The work in this thesis is done in discrete time and all the time-dependent equations will be presented in their discrete form providing a better reading experience.

2.1 Induction Motor

2.1.1 Structure and Principle

The induction machine is the most used electrical machine, it is cheap, rugged and easy to maintain[3]. When directly fed with line voltages (50 or 60 Hz) they operate at almost constant speed, there are however ways to vary this speed by means of power electronic converters [3].

Induction machines consists of many parts, Figure 2.1 shows a basic representation of the induction machines physical appearance and structure.

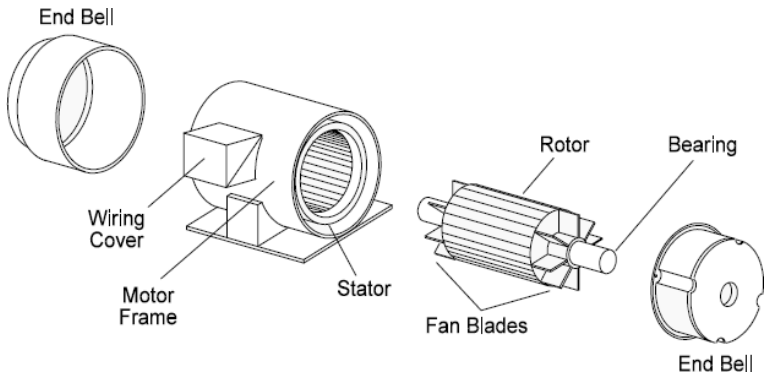


Figure 2.1: Induction Machine.[4]

The two main parts of the induction machine are the stator and the rotor.
Stator
A Stator is made of thin slotted steel laminations and windings inside the motor frame. The reasoning for the thin steel laminations is to reduce eddy current losses. The slots of the steel

laminations is spread out evenly across the inside of the motor. In these slots the windings are coiled with respect to how many phases and poles the motor have [5].

Rotor

There are two types of rotors used for the induction machine, the wound rotor and squirrel cage rotor. The squirrel cage rotor is made up of thin bars arranged in the form of a squirl cage and have its name for that very reason. The rotor bars are usually made from aluminium. At both ends of the rotor the bars are finally connected to each other via shorting rings [5]. Around the bars and between the shorting rings, thin steel laminations are stacked, this is shown in figure 2.2.

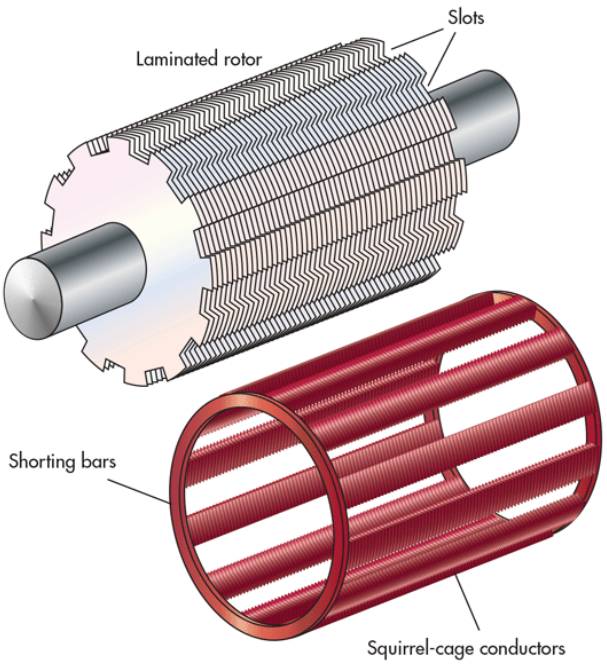


Figure 2.2: Squirrel Cage Rotor.[6]

Working Principle

When alternating voltages are applied to the terminals of the stator a magnetic field is created in the windings. With the physical arrangement of the stator windings the magnetic field is rotating with synchronous velocity.

The rotating magnetic field of the stator induces an electromotive force (EMF) in the rotor bars. Then a current is created in the rotor bars and another magnetic field is induced in the rotor with opposite polarity of the stators field. The magnetic field rotating in the stator will then produces a torque which pulls on the field of the rotor and a rotor rotation is established [7]. This is due to Faraday's law of induction shown in equation 2.1; where: ϵ is EMF, Φ is magnetic flux. This law states that the EMF is given by the rate of change in magnetic flux:

$$\epsilon = -\frac{d\Phi}{dt} \tag{2.1}$$

With this law in mind one can see that if the rotor velocity is equal to the synchronous velocity the change in flux will be zero, hence the EMF will be zero and the rotor will slow down to a point where the change in flux is not equal to zero and the rotor velocity will rise again.

2.1.2 T-Equivalent Circuit

To describe an induction motor mathematically an equivalent circuit is needed. Figure 2.3 shows the equivalent T-circuit of an induction machine, this circuit is the base of the induction motor reference frame used in this thesis; which is explained in section 2.1.4.

Note that all fluxes, currents and voltages in this section are represented by space vectors for modelling purposes. The transformation from $f^{a,b,c}$ to f^s is done to simplify notation where f can be flux, voltage or current.

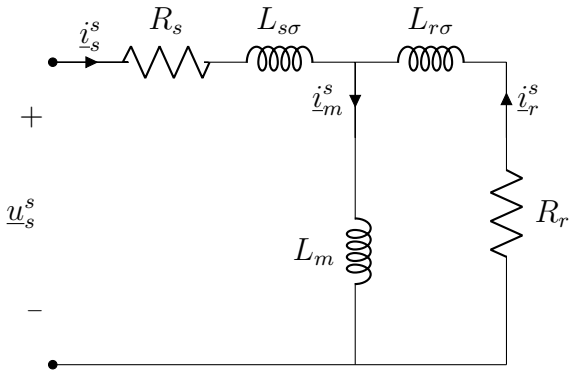


Figure 2.3: T Equivalent Circuit.

where: R_s and R_r are the stator and rotor resistances, $L_{\sigma s}$ and $L_{\sigma r}$ are stator and rotor inductance, respectively. L_M is the magnetizing inductance and $\underline{u}_s^s$ is the stator voltage. $\underline{i}_s^s$, $\underline{i}_r^s$ and $\underline{i}_m^s$ are stator current, rotor current and magnetizing current respectively.

2.1.3 Arbitrary Reference Frames

For induction machine modelling and controlling arbitrary reference frames are used as a tool to simplify the modelling and controlling for the machine.

Figure 2.4 shows a stationary frame, that transforms a three-phase system into a two vector system. This is a mathematical transformation that gives ease to modelling of a multiple-phase systems; with the popular name of the Clarke transformation.

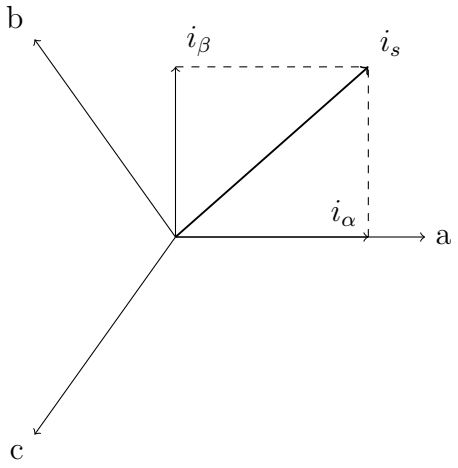


Figure 2.4: $\alpha\beta$-frame.

The frame shown in Figure 2.4 can also be made to rotate at any given velocity, by a Park transformation. This transformation consist of a direct component and a quadrature component, also known as dq-transformation [8], this is shown in Figure 2.5. The displacement angle is often set to match that of an induction machines rotor angle, then the direct component will be aligned with the rotor flux; and give ease to popular methods like field oriented control [8].

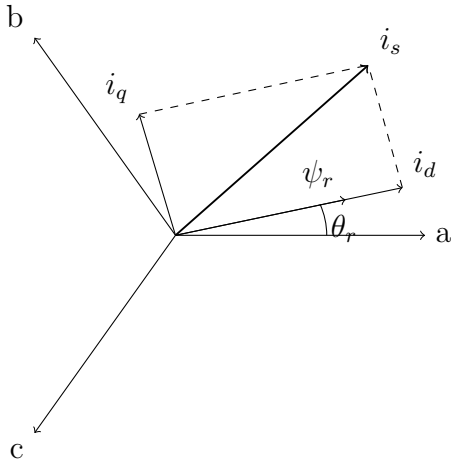


Figure 2.5: dq-frame.

The transformation between a,b and c to dq is shown in the matrix below.

$$\begin{bmatrix} f_{sd} \\ f_{sq} \\ f_{sz} \end{bmatrix} = T(\theta) \cdot \begin{bmatrix} f_{sa} \\ f_{sb} \\ f_{sc} \end{bmatrix}$$

Where:

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

2.1.4 dq-Equivalent Circuit

The circuits and equations shown for this subsection is used for all the modeling done in this thesis and these equations and cirquits are derived by P.C Krause and C.H Thomas [9].

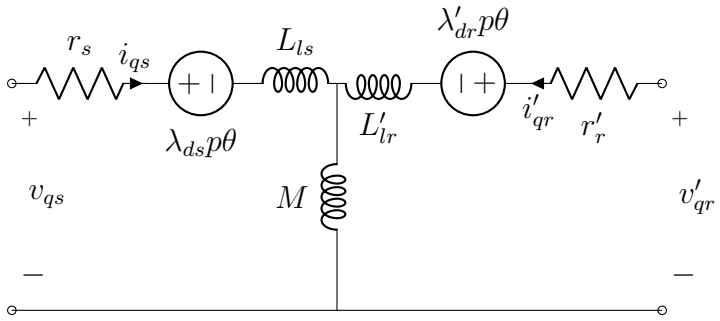


Figure 2.6: dq-Equivalent Circuit, q-axis.

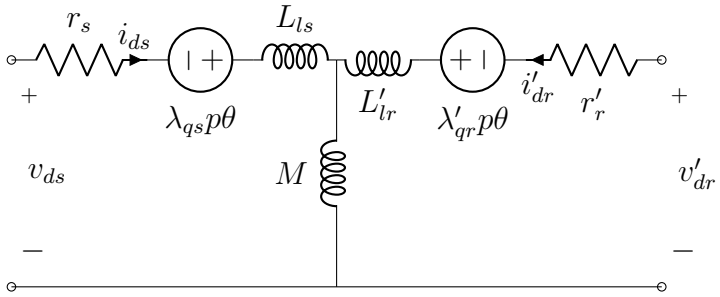


Figure 2.7: dq-Equivalent Circuit, d-axis.

Where: $p = \frac{d}{dt}$, $M = \frac{3}{2}L_{ms}$ and p is and operator for $\frac{d}{dt}$.
The inductances used in Figure 3.2 and Figure 3.3 can be calculated in equation 2.2 to 2.5.

$$L_{ss} = L_s - L_{sm} \tag{2.2}$$

$$L_{sr} = L_r - L_{rm} \tag{2.3}$$

$$L_{ls} = L_{ss} - M \tag{2.4}$$

$$L'_{lr} = L'_{sr} - M \tag{2.5}$$

The fluxes used in Figure 3.2 and Figure 3.3 can be calculated in equation 2.6 to 2.9.

$$\lambda_{qs} = L_{ls} i_{qs} + M(i_{qs} + i'_{qr}) \tag{2.6}$$

$$\lambda_{ds} = L_s i_{ds} + M(i_{ds} + i'_{dr}) \tag{2.7}$$

$$\lambda'_{qr} = L'_{lr} i'_{qr} + M(i_{qs} + i'_{qr}) \tag{2.8}$$

$$\lambda'_{dr} = L'_{lr} i'_{dr} + M(i_{ds} + i'_{dr}) \tag{2.9}$$

2.2 State Space

State space is a modelling method that uses first order differential equations in the form of matrices to describe a system. This method allows for multiple inputs and multiple outputs and gives ease to time integration of big and complex systems.

$$\dot{x} = Ax + Bu \tag{2.10}$$

Where A and B is matrices that describe the dynamics of the system, x is the states of the system and u is the input of the system.

$$y = Cx + Du \tag{2.11}$$

Where y is the output of the system, C is a matrix that describes the output of the system with respect to x, and D is a matrix that describes the output of the system as disturbances due to the input u.

2.2.1 Discretization

Discretization is a transformation from a continuous system to a discrete system [1].

$$\dot{x} = Ax + Bu \tag{2.12}$$

$$x_{k+1} = A_n x_k + B_n u_k \tag{2.13}$$

Equation 2.12 shows the system in its continuous form, while equation 2.13 shows the system in its discrete form. The matrices A_n and B_n are now converted to its discrete from shown in equation 2.14 and 2.15, respectively.

$$A_n = I + AT_s + \frac{A^2T_s^2}{2!} + \frac{A^3T_s^3}{3!} + \cdots \approx I + A_nT_s \tag{2.14}$$

$$B_n = BT_s + \frac{B^2T_s^2}{2!} + \frac{B^3T_s^3}{3!} + \cdots \approx B_nT_s \tag{2.15}$$

Where:
I is an identity matrix, and T_s is the sampling time of the system.

2.3 Observability and Controllability

Controllability and observability are two important concepts used in modern control theory. These concepts were introduced by R. Kalman [10], [11]. This thesis will only define these consents for linear systems.

2.3.1 Observability

Refering to Kalmans's obervabillty theory, a system of states are said to be observable at time t_0 if, with the system in state $x(t_0)$, it is possible to determine this state from the observations of the output over a finite time interval. Moreso, a system of states is fully observable if and only if the observabillty matrix $(n \times m)$ matrix has n linearly independent column vectors [12]. This is much like as in linear algebra with n number of solutions for n number of equations, but if there is n-1 number of equations and n number of solutions the entirety if the system is not solvable.

The matrix shown in 2.16 is called the observability matrix and determine the rank of the system [12]. The rank of the system is a value made to determine how observable the system is. If a system has an A matrix of four rows and four columns and a rank of four, the system is fully observable; any less then this the system is not or partially observable.

$$\mathcal{O}(A, C) = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{n-1} \end{bmatrix} \tag{2.16}$$

2.3.2 Controllability

Referring to Kalmans's controllability theory, a system of states are said to be controllable at time t_0 if it is possible by means of an unconstrained control vector to transfer the system from any initial state $x(t_0)$ to any other state in a finite interval of time. Full controllability is obtained when the input of a system can change all the states of the system from any initial value to any given final value. In other words, any desired state value can be reached with the correct input value form any initial state value.

The matrix in 2.17 is called the controllability matrix [12]. This matrix determine the rank of the system.

$$\mathcal{C}(A, B) = \begin{bmatrix} B & AB & A^2B & \cdots & A^{n-1}B \end{bmatrix} \tag{2.17}$$

2.4 Kalman Filter

The Kalman filter is an estimation tool used in many branches of engineering and science.

$$x_k = Ax_{k-1} + Bu_{k-1} + w_{k-1} \tag{2.18}$$

A and B are matrices corresponding the the process to be estimated, x_k is the actual states of the process.

$$z_k = Hx_k + v_k \tag{2.19}$$

H is the output matrix which determine what state that is used as measurement, z_k is the measurement.

w_k and v_k is the process and measurement noise, respectively. The noise covariances Q and R can be updated at every time step or just kept constant, depending on the process to be estimated. The probability of $p(w)$ $N(0, Q)$ and $p(v)$ $N(0, R)$ are white Gaussian normal noise distributions with zero mean[13].

$$e_k^- = x_k - \hat{x}_k^- \tag{2.20}$$

$$e_k = x_k - \hat{x} \tag{2.21}$$

e_k^- and e_k are the a priori and a posteriori errors, respectively.

$$P_k^- = E[e_k^- e_k^{-T}] \tag{2.22}$$

$$P_k = E[e_k e_k^T] \tag{2.23}$$

P_k^- and P_k are the a priori and a posteriori error covariances, respectively.

K_k is the Kalman gain and decides how much gain the measurement should have, or in other words, how much the measurement can be trusted.

$$K_k = \frac{P_k^- H^T}{HP_k^- H^T + R} \tag{2.24}$$

$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - H\hat{x}_k^-) \tag{2.25}$$

If measurement error covariance (R) goes to zero, $\lim_{R\to 0} K_k = \frac{1}{H}$. That means that the measurement is trusted completely [13].

Prediction	Correction
(1) Prior State Estimate $\hat{x}_k^- = A\hat{x}_{k\text{^}-1} + Bu_{k-1}$ (2) Prior Error Covariance $P_k^- = \mathrm{A}_k P_{k-1} A_k^T + Q$	(3) Kalman Gain $\mathrm{K}_k = P_k^- H_k^T \left(H_k\ P_k^- H_k^T + R\right)^{-1}$ (4) Poserior State Estimate $\hat{x}_k = \hat{x}_k^- + \mathrm{K_k}(\mathrm{z_k} - H_k\hat{x}_k^-)$ (5) Posterior Error Covariance $P_k = (I - K_k H_k)P_k^-$

Figure 2.8: Discrete Kalman Filter Algorithm. [13]

2.4.1 Extended Kalman Filter

The extended Kalman filter is an extension of the normal Kalman filter algorithm. This extension is designed to make the Kalman filter work for non-linear systems. This is is done by linearising the system matrices via partial derivation at every iteration in discrete time, this linearisation is shown in equation 2.26 to 2.29 [13].

$$F = \frac{\partial f(\hat{x}, u, 0)}{\partial x} \tag{2.26}$$

$$W = \frac{\partial f(\hat{x}, u, 0)}{\partial w} \tag{2.27}$$

$$H = \frac{\partial h(\tilde{x}, 0)}{\partial x} \tag{2.28}$$

$$V = \frac{\partial h(\tilde{x}, 0)}{\partial v} \tag{2.29}$$

Where: $x_k = f(\partial\hat{x}_{k-1}, u_{k-1}, w_{k-1})$ and is the actual value of the states, $\tilde{x}_k = f(\partial\hat{x}_{k-1}, u_{k-1}, 0)$ and is the approximate value of the states, and $\hat{x}_k$ is the estimated posterior value of the states [13].

Prediction	Correction
(1) Prior State Estimate $\hat{x}_k^- = \mathrm{f}(\hat{x}_{k-1}, u_{k-1})$ (2) Prior Error Covariance $P_k^- = \mathrm{F}_k P_{k-1} F_k^T + \mathrm{W}_k \mathrm{Q}_{k-1} W_k^T$	(3) Kalman Gain $\mathrm{K}_k = P_k^- H_k^T \left(H_k \, P_k^- H_k^T + V_k R_K V_k^T \right)^{-1}$ (4) Poserior State Estimate $\hat{x}_k = \hat{x}_k^- + \mathrm{K}_\mathrm{k}(\mathrm{z}_\mathrm{k} - H_k \hat{x}_k^-)$ (5) Posterior Error Covariance $P_k = (I - K_k H_k) P_k^-$

Figure 2.9: Discrete Extended Kalman Filter Algorithm. [13]

2.5 Linear Quadratic Regulator

The Linear Quadratic Regulator(LQR) algorithm is in some ways similar to the Kalman filter algorithm. The name descreibes it very well, linear system equations with a quadratic cost function. This regulator creates an error covariance that is used in an error correrction gain, or a feedback gain. This gain is then used to produce the required system input according to the given states, this is shown in equation 2.30 to 2.32 [14]. The control scheme for this regulator is shown in figure 2.10.

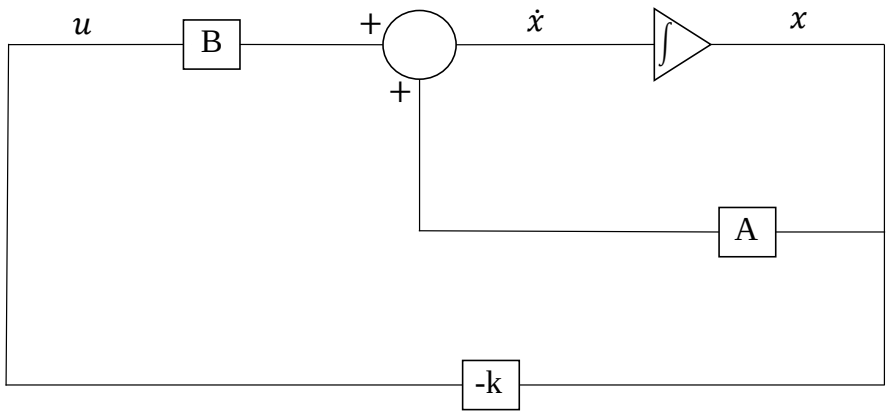


Figure 2.10: LQR Control Scheme.

$$P_{LQR} = A^T P_{LQR} A - A^T P_{LQR} B (R_{LQR} + B^T P_{LQR} B)^{-1} B^T A + Q_{LQR} \tag{2.30}$$

$$K_{LQR} = (R_{LQR} + B^T P_{LQR} B)^{-1} B^T P_{LQR} A \tag{2.31}$$

$$u_{LQR} = -k_{LQR} x \tag{2.32}$$

Q_{LQR} is a matrix that decides how much weight is put on each state, R_{LQR} does the same thing for input values. These values are used to change the response of the system, much like the tuning of a PID-regulator, where one can change speed and accuracy of the system. Equation 2.33 shows how the quadratic function for a discrete infinite horizon linear system.

$$J = \sum_{0}^{\infty} \left(x_k^T Q x_k + u_K^T R u_k \right) \tag{2.33}$$

2.5.1 Linear Quadratic Tracking

Linear Quadratic Tracking(LQT) have the same properties as the LQR, but with the option for setting reference state values. The reference state values are denoted as r_c and are shown in figure 2.11 and in equations 2.34 to 2.36. The purpose of the LQR is to drive the states to zero, with LQT the error of the states are driven to zero [15].

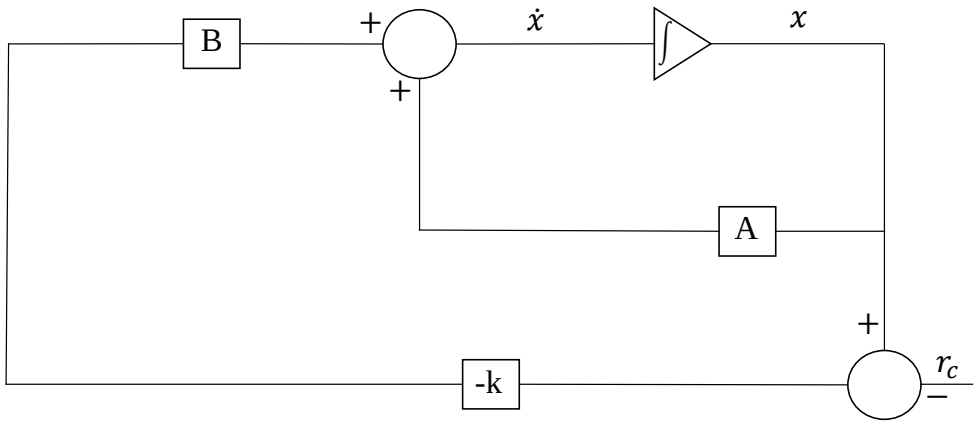


Figure 2.11: LQT Control Scheme.

$$P_{LQT} = A^T P_{LQT} A - A^T P_{LQT} B (R_{LQT} + B^T P_{LQT} B)^{-1} B^T A + Q_{LQT} \tag{2.34}$$

$$K_{LQT} = (R_{LQT} + B^T P_{LQT} B)^{-1} B^T P_{LQT} A \tag{2.35}$$

$$u_{LQT} = -k_{LQT}(x - r_c) \tag{2.36}$$

Chapter 3

Control Structure and Components

In this chapter an in-depth look of the control structure and its components for this thesis will be explained.

3.1 Control Structure

The control structure presented in this thesis uses two main components to achieve sensorless control with no cascaded PID-controllers. These components are, Extended Kalman filter and a Linear Quadratic Tracker and their implementation will be explained in this chapter.

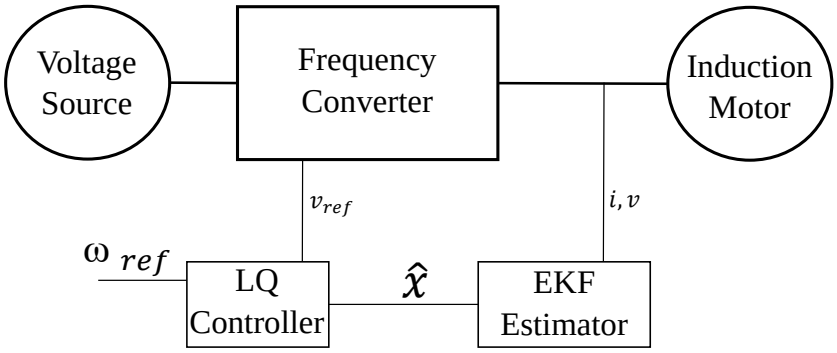


Figure 3.1: Overall Control Scheme.

The proposed control structure is shown in Figure 3.1 where motor currents and -voltages are collected by sensors and fed into the estimator. The estimator produces an estimated full state feedback of all the system states ($\hat{x}$) which are then fed into the controller. The controller produces the corresponding voltages appropriate to the given system states and

reference values. These voltage signals are fed into the frequency converted to give the induction motor the correct voltages.

3.2 Induction Motor Model

The induction motor model used for this thesis is modeled in the dq-reference frame but with the displacement angle of the reference frame equal to zero ($\theta = 0$)[9]. This angle is chosen to prevent faulty results when estimating states, as the all the states depend heavily on this angle and a wrong estimate of this angle may give faulty results.

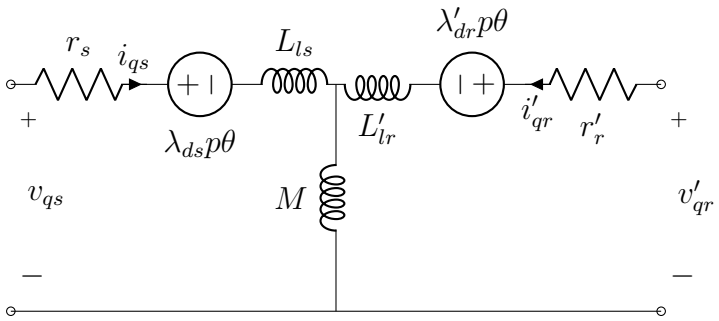


Figure 3.2: dq-Equivalent Circuit, q-axis.

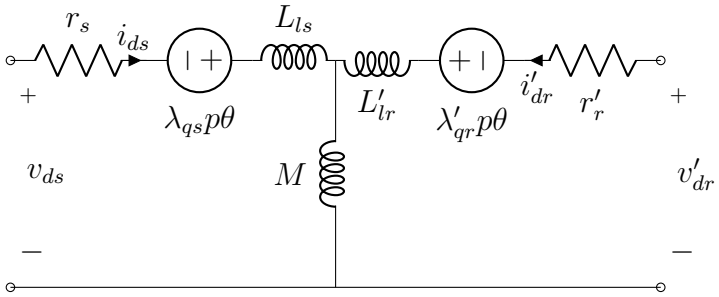


Figure 3.3: dq-Equivalent Circuit, d-axis.

Where: $p = \frac{d}{dt}$, $M = \frac{3}{2}L_{ms}$, $\Psi_{qs} = \lambda_{qs}\omega_e$, etc.

3.2.1 Flux and Current Equations

The equations in this subsection is from a paper by P.C Krause and C.H Thomas [9].

Equations 3.1 to 3.4 show the rate of change in flux for both rotor and stator denoted in the dq reference frame.

$$\dot{\Psi}_{qs} = \omega_e\left[v_{qs} - \frac{\omega}{\omega_e}\Psi_{ds} + \frac{r_s}{X_{ls}}(\Psi_{mq} - \Psi_{qs})\right] \tag{3.1}$$

$$\dot{\Psi}_{ds} = \omega_e\left[v_{ds} + \frac{\omega}{\omega_e}\Psi_{qs} + \frac{r_s}{X_{ls}}(\Psi_{md} - \Psi_{ds})\right] \tag{3.2}$$

$$\dot{\Psi}_{qr} = \omega_e\left[v_{qr} - \frac{\omega - \omega_r}{\omega_e}\Psi_{dr} + \frac{r_r}{X_{lr}}(\Psi_{mq} - \Psi_{qr})\right] \tag{3.3}$$

$$\dot{\Psi}_{sr} = \omega_e\left[v_{dr} + \frac{\omega - \omega_r}{\omega_e}\Psi_{qr} + \frac{r_r}{X_{lr}}(\Psi_{md} - \Psi_{dr})\right] \tag{3.4}$$

Equation 3.5 to 3.8 show the currents for both rotor and stator denoted in the dq reference frame.

$$i_{qs} = \frac{1}{X_{ls}}(\Psi_{qs} - \Psi_{mq}) \tag{3.5}$$

$$i_{ds} = \frac{1}{X_{ls}}(\Psi_{ds} - \Psi_{md}) \tag{3.6}$$

$$i_{qt} = \frac{1}{X_{lr}}(\Psi_{qr} - \Psi_{mq}) \tag{3.7}$$

$$i_{dr} = \frac{1}{X_{lr}}(\Psi_{ds} - \Psi_{md}) \tag{3.8}$$

Where:

$$X_{mq} = X_{md} = \frac{1}{\frac{1}{X_m} + \frac{1}{X_{ls}} + \frac{1}{X_{lr}}} \tag{3.9}$$

$$\Psi_{mq} = X_{mq}\left[\frac{\Psi_{qs}}{X_{ls}} + \frac{\Psi_{qr}}{X_{lr}}\right] \tag{3.10}$$

$$\Psi_{md} = X_{md}\left[\frac{\Psi_{ds}}{X_{ls}} + \frac{\Psi_{dr}}{X_{lr}}\right] \tag{3.11}$$

3.2.2 Torque and Velocity Equations

The equations in this subsection is also from the paper by P.C Krause and C.H Thomas [9].

$$T_{em} = \left(\frac{n}{2}\right)\left(\frac{P}{2}\right)(\lambda_{qr} i_{dr} - \lambda_{dr} i_{qr}) \tag{3.12}$$

$$\dot{\omega}_r = \left(\frac{P}{2}\right)\frac{T_{em} - T_l}{J} \tag{3.13}$$

3.3 Induction Motor States Space Model

The state equations are made for a induction motor model and consist of stator and rotor fluxes together with the angular velocity of the rotor.

$$\dot{x} = Ax + Bu \tag{3.14}$$

The states, inputs and outputs are shown in equation 3.15, 3.16 and 3.17, respectively.

$$x = \begin{bmatrix} \Psi_{qs},\ \Psi_{ds},\ \Psi_{qr},\ \Psi_{dr},\ \omega_r \end{bmatrix}^T \tag{3.15}$$

$$u = \begin{bmatrix} v_{qs},\ v_{ds},\ T_l \end{bmatrix}^T \tag{3.16}$$

$$y = \begin{bmatrix} i_{qs},\ i_{ds} \end{bmatrix}^T \tag{3.17}$$

Currents is chosen for the output y, to give the estimator a solid input to generate its full state feedback. Others have had success with using currents as outputs [1].

The A matrix contains the state dependent elements of the state equations.

$$A = \begin{bmatrix} \omega_e \frac{r_s\left(\frac{X_{mq}}{X_{ls}}-1\right)}{X_{ls}} T_s & 0 & \omega_e \frac{r_s X_{mq}}{X_{ls}X_{lr}} T_s & 0 & 0 \\ 0 & \omega_e \frac{r_s\left(\frac{X_{md}}{X_{ls}}-1\right)}{X_{ls}} T_s & 0 & \omega_e \frac{r_s X_{md}}{X_{ls}X_{lr}} T_s & 0 \\ \omega_e \frac{r_r X_{mq}}{X_{ls}X_{lr}} T_s & 0 & \omega_e \frac{r_r\left(\frac{X_{mq}}{X_{lr}}-1\right)}{X_{lr}} T_s & \omega_e \frac{\omega_r}{\omega_e} T_s & 0 \\ 0 & \omega_e \frac{r_r X_{md}}{X_{ls}X_{lr}} T_s & -\omega_e \frac{\omega_r}{\omega_e} T_s & \omega_e \frac{r_r\left(\frac{X_{md}}{X_{lr}}-1\omega_e\right)}{X_{lr}} T_s & 0 \\ A_{qs}T_s & -A_{ds}T_s & A_{qr}T_s & -A_{dr}T_s & 0 \end{bmatrix} \tag{3.18}$$

Where:

$$A_{qs} = \left(\frac{P}{2}\right)\left(\frac{nP}{4J}\right)\frac{X_{lr}X_{mq}\Psi_{dr}}{X_{lr}^2 X_{ls}\omega_e} \tag{3.19}$$

$$A_{ds} = \left(\frac{P}{2}\right)\left(\frac{nP}{4J}\right)\frac{X_{lr}X_{md}\Psi_{qr}}{X_{lr}^2 X_{ls}\omega_e} \tag{3.20}$$

$$A_{qr} = \left(\frac{P}{2}\right)\left(\frac{nP}{4J}\right)\frac{X_{ls}X_{mq}\Psi_{dr}}{X_{lr}^2 X_{ls}\omega_e} \tag{3.21}$$

$$A_{dr} = \left(\frac{P}{2}\right)\left(\frac{nP}{4J}\right)\frac{X_{ls}X_{md}\Psi_{qr}}{X_{lr}^2 X_{ls}\omega_e} \tag{3.22}$$

The B matrix contains the nonstate dependent elements of the state equations.

$$B = \begin{bmatrix} \omega_e & 0 & 0 \\ 0 & \omega_e & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -\left(\frac{P}{2J}\right) \end{bmatrix} \tag{3.23}$$

$$C = \begin{bmatrix} \frac{1 - \frac{X_{mq}}{X_{ls}}}{X_{ls}} & 0 & -\frac{X_{mq}}{X_{ls}X_{lr}} & 0 & 0 \\ 0 & \frac{1 - \frac{X_{md}}{X_{ls}}}{X_{ls}} & 0 & -\frac{X_{md}}{X_{ls}X_{lr}} & 0 \end{bmatrix} \tag{3.24}$$

3.3.1 Discretization of State Space Model

To convert the continuous versions of the matrices A and B to discrete versions A_n and B_n; discretization is needed.

$$\dot{x} = Ax + Bu \tag{3.25}$$

$$x_{k+1} = A_n x_k + B_n u_k \tag{3.26}$$

A_n and B_n are matrices used for the discrete sate equations 3.26, and are calculated in equation 3.27 and 3.28, respectively.

$$A_n \approx I + A_n T_s \tag{3.27}$$

$$B_n \approx B_n T_s \tag{3.28}$$

$$A_n = \begin{bmatrix} 1 + \omega_e \frac{r_s\left(\frac{X_{mq}}{X_{ls}} - 1\right)}{X_{ls}} T_s & 0 & \omega_e \frac{r_s X_{mq}}{X_{ls}X_{lr}} T_s & 0 & 0 \\ 0 & 1 + \omega_e \frac{r_s\left(\frac{X_{md}}{X_{ls}} - 1\right)}{X_{ls}} T_s & 0 & \omega_e \frac{r_s X_{md}}{X_{ls}X_{lr}} T_s & 0 \\ \omega_e \frac{r_r X_{mq}}{X_{ls}X_{lr}} T_s & 0 & 1 + \omega_e \frac{r_r\left(\frac{X_{mq}}{X_{lr}} - 1\right)}{X_{lr}} T_s & \omega_e \frac{\omega_r}{\omega_e} T_s & 0 \\ 0 & \omega_e \frac{r_r X_{md}}{X_{ls}X_{lr}} T_s & -\omega_e \frac{\omega_r}{\omega_e} T_s & 1 + \omega_e \frac{r_r\left(\frac{X_{md}}{X_{lr}} - 1\omega_e\right)}{X_{lr}} T_s & 0 \\ A_{qs} T_s & -A_{ds} T_s & A_{qr} T_s & -A_{dr} T_s & 1 \end{bmatrix} \tag{3.29}$$

Where:

$$A_{qs} = \left(\frac{P}{2}\right)\left(\frac{nP}{4J}\right)\frac{X_{lr}X_{mq}\Psi_{dr}}{X_{lr}^2 X_{ls}\omega_e} \tag{3.30}$$

$$A_{ds} = \left(\frac{P}{2}\right)\left(\frac{nP}{4J}\right)\frac{X_{lr}X_{md}\Psi_{qr}}{X_{lr}^2 X_{ls}\omega_e} \tag{3.31}$$

$$A_{qr} = \left(\frac{P}{2}\right)\left(\frac{nP}{4J}\right)\frac{X_{ls}X_{mq}\Psi_{dr}}{X_{lr}^2 X_{ls}\omega_e} \tag{3.32}$$

$$A_{dr} = \left(\frac{P}{2}\right)\left(\frac{nP}{4J}\right)\frac{X_{ls}X_{md}\Psi_{qr}}{X_{lr}^2 X_{ls}\omega_e} \tag{3.33}$$

$$B_n = \begin{bmatrix} \omega_e T_s & 0 & 0 \\ 0 & \omega_e T_s & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -\left(\frac{PT_s}{2J}\right) \end{bmatrix} \tag{3.34}$$

Where:
T_s is the sampling time for the given system.

3.4 Estimator

The estimator used for this thesis is an Extended Kalman Filter(EKF) which allows for state estimation for nonliniar systems, see chapter 2.4 for more information regarding the Kalman filter algorithm.

Since the load torque is not known, a full representation of the acceleration cannot be fed into the estimator; with that in mind an alternative method is used where the motor velocity is added by an augmented matrix [1].

$$x = \begin{bmatrix} \Psi_{qs},\ \Psi_{ds},\ \Psi_{qr},\ \Psi_{dr},\ \omega_r \end{bmatrix}^T \tag{3.35}$$

$$u = \begin{bmatrix} v_{qs},\ v_{ds} \end{bmatrix}^T \tag{3.36}$$

$$y = \begin{bmatrix} i_{qs},\ i_{ds} \end{bmatrix}^T \tag{3.37}$$

$$A_n = \begin{bmatrix} 1+\omega_e \frac{r_s\left(\frac{X_{mq}}{X_{ls}}-1\right)}{X_{ls}} T_s & 0 & \omega_e \frac{r_s X_{mq}}{X_{ls}X_{lr}} T_s & 0 & 0 \\ 0 & 1+\omega_e \frac{r_s\left(\frac{X_{md}}{X_{ls}}-1\right)}{X_{ls}} T_s & 0 & \omega_e \frac{r_s X_{md}}{X_{ls}X_{lr}} T_s & 0 \\ \omega_e \frac{r_r X_{mq}}{X_{ls}X_{lr}} T_s & 0 & 1+\omega_e \frac{r_r\left(\frac{X_{mq}}{X_{lr}}-1\right)}{X_{lr}} T_s & \omega_e \frac{\omega_r}{\omega_e} T_s & 0 \\ 0 & \omega_e \frac{r_r X_{md}}{X_{ls}X_{lr}} T_s & -\omega_e \frac{\omega_r}{\omega_e} T_s & 1+\omega_e \frac{r_r\left(\frac{X_{md}}{X_{lr}}-1\omega_e\right)}{X_{lr}} T_s & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.38}$$

$$B = \begin{bmatrix} \omega_e T_s & 0 \\ 0 & \omega_e T_s \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \tag{3.39}$$

$$C = \begin{bmatrix} \dfrac{1-\dfrac{X_{mq}}{X_{ls}}}{X_{ls}} & 0 & -\dfrac{X_{mq}}{X_{ls}X_{lr}} & 0 & 0 \\ 0 & \dfrac{1-\dfrac{X_{md}}{X_{ls}}}{X_{ls}} & 0 & -\dfrac{X_{md}}{X_{ls}X_{lr}} & 0 \end{bmatrix} \tag{3.40}$$

3.4.1 Noise

Q_{EKF} and R_{EKF} represent the covariance matrices of process and measurement noise, respectively; these are shown in the equations below:

$$Q_{EKF} = cov(w) = E\{ww^T\} \tag{3.41}$$

$$R_{EKF} = cov(v) = E\{vv^T\} \tag{3.42}$$

Process and measurement noise weights are chosen to:

$$Q_{EKF} = \begin{bmatrix} \zeta_{EKF} & 0 & 0 & 0 & 0 \\ 0 & \zeta_{EKF} & 0 & 0 & 0 \\ 0 & 0 & \zeta_{EKF} & 0 & 0 \\ 0 & 0 & 0 & \zeta_{EKF} & 0 \\ 0 & 0 & 0 & 0 & \gamma_{EKF} \end{bmatrix}, \; R_{EKF} = \begin{bmatrix} \varepsilon_{EKF} & 0 \\ 0 & \varepsilon_{EKF} \end{bmatrix}$$

Where:
$\zeta_{EKF} = 10^{-6}$, $\gamma_{EKF} = 20$ and $\varepsilon_{EKF} = 10^{-6}$

3.4.2 Linearisation

Extended kalman filter linearisation of A_n and C are shown as $F(k)$ and $H(k)$ in equation 3.43 and 3.44, respectively; more information regarding this subject can be found in chapter 2.4.1.

$$F(k) = \begin{bmatrix} 1 + \frac{\omega_e r_s \left(\frac{X_{mq}}{X_{ls}} - 1\right)}{X_{ls}} T_s & 0 & \frac{\omega_e r_s X_{mq}}{X_{ls} X_{lr}} T_s & 0 & 0 \\ 0 & 1 + \frac{\omega_e r_s \left(\frac{X_{md}}{X_{ls}} - 1\right)}{X_{ls}} T_s & 0 & \frac{\omega_e r_s X_{md}}{X_{ls} X_{lr}} T_s & 0 \\ \frac{\omega_e r_r X_{mq}}{X_{ls} X_{lr}} T_s & 0 & 1 + \frac{\omega_e r_r \left(\frac{X_{mq}}{X_{lr}} - 1\right)}{X_{lr}} T_s & \omega_r T_s & \frac{\omega_e \Psi_{dr}}{\omega_e} T_s \\ 0 & \frac{\omega_e r_r X_{md}}{X_{ls} X_{lr}} T_s & -\omega_r T_s & 1 + \frac{\omega_e r_r \left(\frac{X_{md}}{X_{lr}} - 1\right)}{X_{lr}} T_s & -\frac{\omega_e \Psi_{qr}}{\omega_e} T_s \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.43}$$

$$H(k) = \begin{bmatrix} \frac{1 - \frac{X_{mq}}{X_{ls}}}{X_{ls}} & 0 & -\frac{X_{mq}}{X_{ls} X_{lr}} & 0 & 0 \\ 0 & \frac{1 - \frac{X_{md}}{X_{ls}}}{X_{ls}} & 0 & -\frac{X_{md}}{X_{ls} X_{lr}} & 0 \end{bmatrix} \tag{3.44}$$

3.4.3 Estimated Torque

From the estimated states provided by extended Kalman filter, electromagnetic torque are calculated.

$$\hat{T}_{em} = \left(\frac{nP}{4}\right)\left[\frac{X_{mq}\Psi_{dr}}{X_{lr}^2 X_{ls}\omega_e}(X_{lr}\Psi_{qs} + X_{ls}\Psi_{qr}) - \frac{X_{md}\Psi_{qr}}{X_{lr}^2 X_{ls}\omega_e}(X_{lr}\Psi_{ds} + X_{ls}\Psi_{dr})\right] \tag{3.45}$$

$$\hat{\dot{\omega}}_r = \left(\frac{P}{2}\right)\frac{\hat{T}_{em} - \hat{T}_l}{J} \tag{3.46}$$

With equation 3.45 and 3.46 the load torque $\hat{T}_l$ is calculated in equation 3.47.

$$\hat{T}_l = \hat{\dot{\omega}}_r\left(\frac{2J}{P}\right) - \hat{T}_{em} \tag{3.47}$$

The estimated load torque have no practical purpose for this controller, it can however be beneficial to observe what loads the motor is undergoing.

3.4.4 Observability

Observability for non-linear systems can be complex, more complex then what is covered for linear systems in chapter 2.3.1; non-linear system observability will not be covered in this thesis. In this thesis it is assumed that the system have full observability rank, due to the fact that this have been done before with success [1], [16].

3.5 Controller

The controller used in this thesis is an linear quadratic tracker.

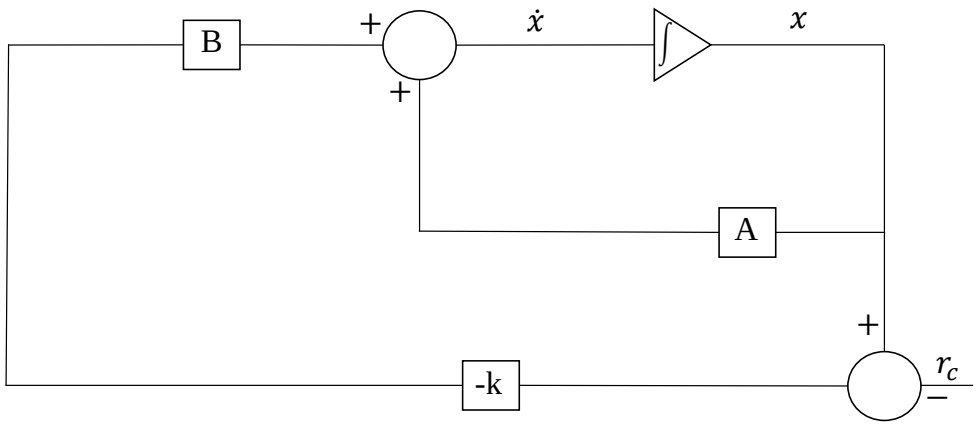


Figure 3.4: LQT Control Scheme.

This controller requires full state feedback, which it gets from the estimator. To get the best results possible, this controller is a linearized full model of the induction motor.

$$A_n = \begin{bmatrix} 1+\frac{\omega_e r_s\left(\frac{X_{mq}}{X_{ls}}-1\right)}{X_{ls}}T_s & 0 & \frac{\omega_e r_s X_{mq}}{X_{ls}X_{lr}}T_s & 0 & 0 \\ 0 & 1+\frac{\omega_e r_s\left(\frac{X_{md}}{X_{ls}}-1\right)}{X_{ls}}T_s & 0 & \frac{\omega_e r_s X_{md}}{X_{ls}X_{lr}}T_s & 0 \\ \frac{\omega_e r_r X_{mq}}{X_{ls}X_{lr}}T_s & 0 & 1+\frac{\omega_e r_r\left(\frac{X_{mq}}{X_{lr}}-1\right)}{X_{lr}}T_s & \omega_r T_s & \frac{\omega_e \Psi_{dr}}{\omega_e}T_s \\ 0 & \frac{\omega_e r_r X_{md}}{X_{ls}X_{lr}}T_s & -\omega_r T_s & 1+\frac{\omega_e r_r\left(\frac{X_{md}}{X_{lr}}-1\right)}{X_{lr}}T_s & -\frac{\omega_e \Psi_{qr}}{\omega_e}T_s \\ A_{qs}T_s & -A_{ds}T_s & -A_{qr}T_s & A_{dr}T_s & 1 \end{bmatrix} \tag{3.48}$$

Where:

$$A_{qs} = \left(\frac{P}{2}\right)\left(\frac{nP}{4J}\right)\frac{X_{mq}\Psi_{dr}}{X_{lr}X_{ls}\omega_e} \tag{3.49}$$

$$A_{ds} = \left(\frac{P}{2}\right)\left(\frac{nP}{4J}\right)\frac{X_{md}\Psi_{qr}}{X_{lr}X_{ls}\omega_e} \tag{3.50}$$

$$A_{qr} = \left(\frac{P}{2}\right)\left(\frac{nP}{4J}\right)\left(\frac{X_{lr}X_{md}\Psi_{ds} + X_{ls}X_{md}\Psi_{dr} - X_{ls}X_{mq}\Psi_{dr}}{X_{lr}^2 X_{ls}\omega_e}\right) \tag{3.51}$$

$$A_{dr} = \left(\frac{P}{2}\right)\left(\frac{nP}{4J}\right)\left(\frac{X_{lr}X_{mq}\Psi_{qs} - X_{ls}X_{md}\Psi_{qr} + X_{ls}X_{mq}\Psi_{qr}}{X_{lr}^2 X_{ls}\omega_e}\right) \tag{3.52}$$

Referring to section 3.4.3 this control system offers estimated load torque $\hat{T}_l$ which can be used to give the controller a complete picture of the motor.
There is however a problem with the implementation of $\hat{T}_l$, the problem is that it cannot be directly implimented in the B_n matrix because then $\hat{T}_l$ would be a input. This model have no way of utilizing this input and therefore it will not be used as a control input.

The B_n matrix is chosen without any load torque.

$$B_n = \begin{bmatrix} \omega_e T_s & 0 \\ 0 & \omega_e T_s \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \tag{3.53}$$

3.5.1 Weighing Values

State weights are chosen to here, whereas how the input weights are chosen is explained in chapter 4.1.3.

$$Q_{LQT} = \begin{bmatrix} \zeta_{LQT} & 0 & 0 & 0 & 0 \\ 0 & \zeta_{LQT} & 0 & 0 & 0 \\ 0 & 0 & \zeta_{LQT} & 0 & 0 \\ 0 & 0 & 0 & \zeta_{LQT} & 0 \\ 0 & 0 & 0 & 0 & \gamma_{LQT} \end{bmatrix}, \; R_{LQT} = \begin{bmatrix} \varepsilon_{LQT} & 0 \\ 0 & \varepsilon_{LQT} \end{bmatrix}$$

Where:
$\zeta_{LQT} = \gamma_{LQT} = 1$.

3.5.2 Reference Values

The reference signal is implemented as in figure 3.4, where r_c is the desired reference value. Since this is an five state system the reference value also have five references. The velocity is the only state with an desired reference value, this is shown in the matrix below.

$$r_c = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \omega_{rc} \end{bmatrix} \tag{3.54}$$

Where:

ω_{rc} is the control velocity.

This r_c is added the the control scheme and shown in figure 3.5. The reference signal velocity(ω_{ref}) is added with together with the error between the reference velocity(ω_{ref}) signal and the estimated velocity($\hat{\omega}_r$); this is shown in equation 3.55 and 3.56.

$$\omega_{err} = \omega_{ref} - \hat{\omega}_r \tag{3.55}$$

$$\omega_{rc} = \omega_{ref} + \omega_{err}(P + I) \tag{3.56}$$

Where:
P and I are proportional and integral control gains; $P = 7$ and $I = 10$.

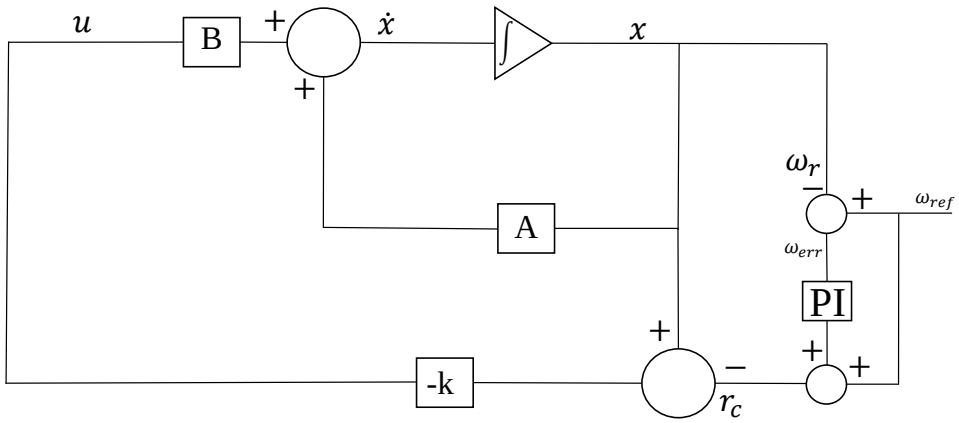


Figure 3.5: LQT Control Scheme with r_c.

This addition to the LQT control scheme is done due to the controller not responding well without it; without this addition the controller only corrects the speed error in no load situations which is not practical. The added error ω_{err} can be looked at as a feed forwarded signal within the closed loop of the controller.

3.5.3 Controllability

Like the observability section 3.4.4, this section will not cover non-linear controllability for this thesis. It is seen trough extensive testing that the control scheme is controllable with the extra addition shown in section 3.5.2. It is not clear why the system is controllable with this addition and only partial controllable without it.

3.5.4 Observations

Trough trail and error it was found that the LQT worked with the non-linearized full model as well as the linearized full model, this could mean that the non-linearized full model is

somehow linearized; the reasoning behind this is another topic.

This is included in the LQT code, so the LQT code gives the option to switch between non-linearized and linearized full model, this is shown in appendix B. This is done to see what model yielded the best results; the non-linearized model gave the best results and the non-linearized model is therefore used for all control simulations. The linearized and non-linearized full model is shown in chapter 3.5 and 3.3.1, respectively.

Chapter 4

Digital Model

In this chapter the code and simulation representation of the induction motor model, the estimator and the controller will be shown and explained.

4.1 Simulation Models

These models are built in Matlab and Simulink.

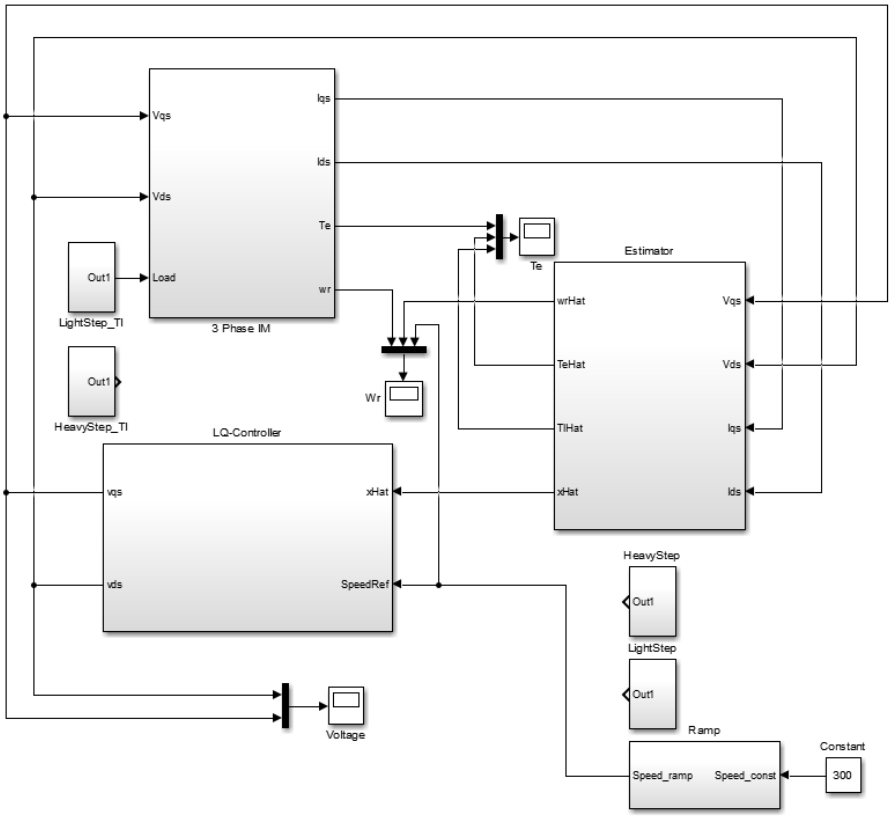


Figure 4.1: Simulink Model.

Figure 4.1 show the entirety of the simulation platform. The block in top left(3 Phase IM) is

the induction motor model. The block to the right(Estimator) is the estimator. The block in the bottom left(LQ-Controller) is the LQT. One can see that both the load torque and reference speed can be set by steps, ramps or constants; where these are the only inputs of the system.

4.1.1 Induction Motor

Figure 4.2 that all the components required to build the induction motor model is shown here in their respective blocks, according to the equation presented in chapter 3.2.

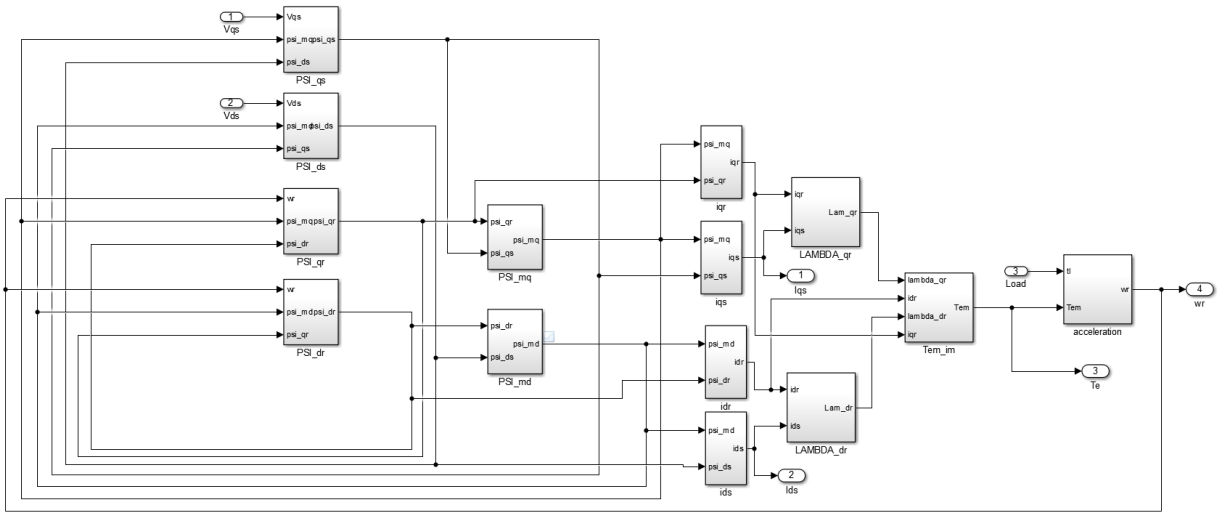


Figure 4.2: IM Model.

There are three inputs to this system, stator dq voltages and load torque(v_{qs}, v_{ds}, T_l). The fluxes (PSI_qs, PSI_ ds etc.) produce the electromagnetic torque(Tem_im) witch together with the load torque(Load) complete the variables need to produce a the rotor velocity (wr).

4.1.2 Estimator

The estimator gets four inputs, stator dq current and voltages(v_{qs}, v_{ds}, i_{qs}, i_{ds}). It produces a total of seven output estimations; all the given states, load- and electromagnetic torque($\hat{\Psi}_{qs}$, $\hat{\Psi}_{ds}$, $\hat{\Psi}_{qr}$, $\hat{\Psi}_{dr}$, $\hat{\omega}_r$, $\hat{T}_l$ and $\hat{T}_{em}$). Its respective Matlab code can be found in appendix A.

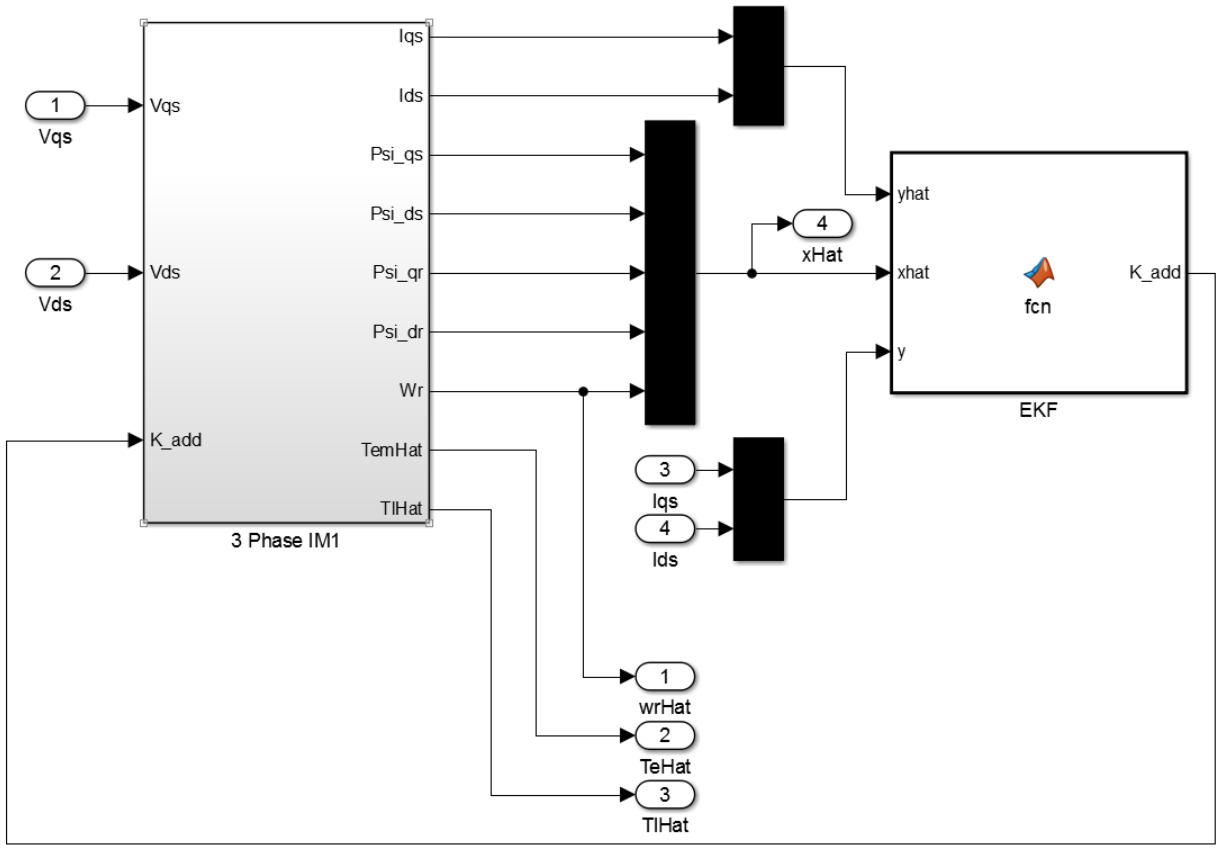


Figure 4.3: EKF Model, view one.

From Figure 4.3, the block to the right(EKF) is the EKF algorithm in code. This EKF block gets a full state update at every time step and produces a Kalman gain which is multiplied with the output error($k_{gain}(y - \hat{y})$) this is noted as K_add in Figure 4.3. This gain is then added to its respective state, this is shown in figure 4.4.

Figure 4.4 show that the model is fairly similar to the model in section 4.1.1. One difference is that velocity is not calculated here but rather estimated with the the kalman gain output error(K_add). Another difference is that the load torque is not an input but rather an output.

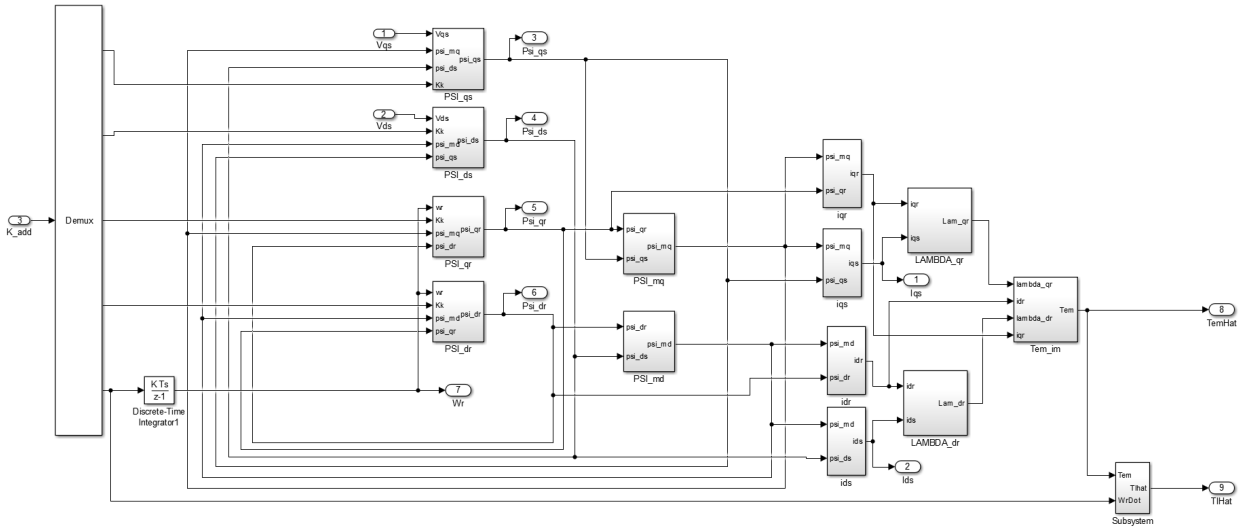


Figure 4.4: EKF Model, view two.

4.1.3 Controller

The controller have a total of six inputs, estimated states and reference speed signal($\hat{\Psi}_{qs}$, $\hat{\Psi}_{ds}$, $\hat{\Psi}_{qr}$, $\hat{\Psi}_{dr}$, $\hat{\omega}_r$, ω_{ref}). It have a total of two outputs, stator dq-voltages(v_{qs}, v_{ds}). Its respective Matlab code can be found in appendix B.

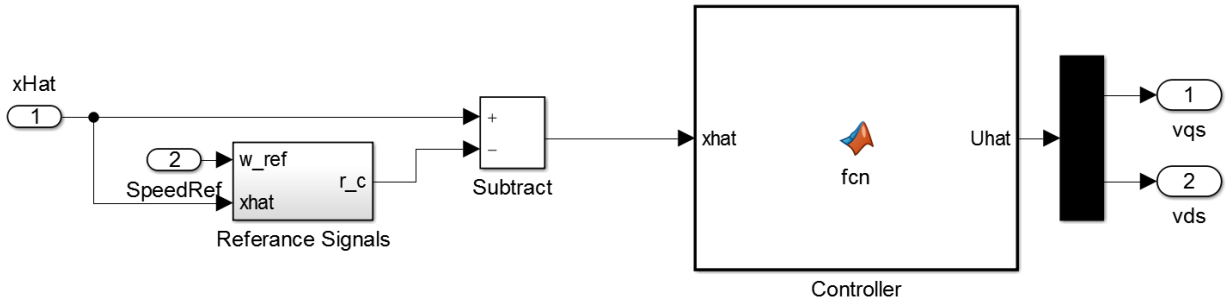


Figure 4.5: LQT Model.

In Figure 4.5 the function block to the right(Controller) is where the actual LQR algorithm is done.
In Figure 4.6 one can see how the reference signal r_c is built, in the same way as shown in chapter 3.5.2. Where the velocity error is multiplied with the given PI gains and added to the reference signal to produce the final value of r_c. One can see that the other states are not getting any reference signals.

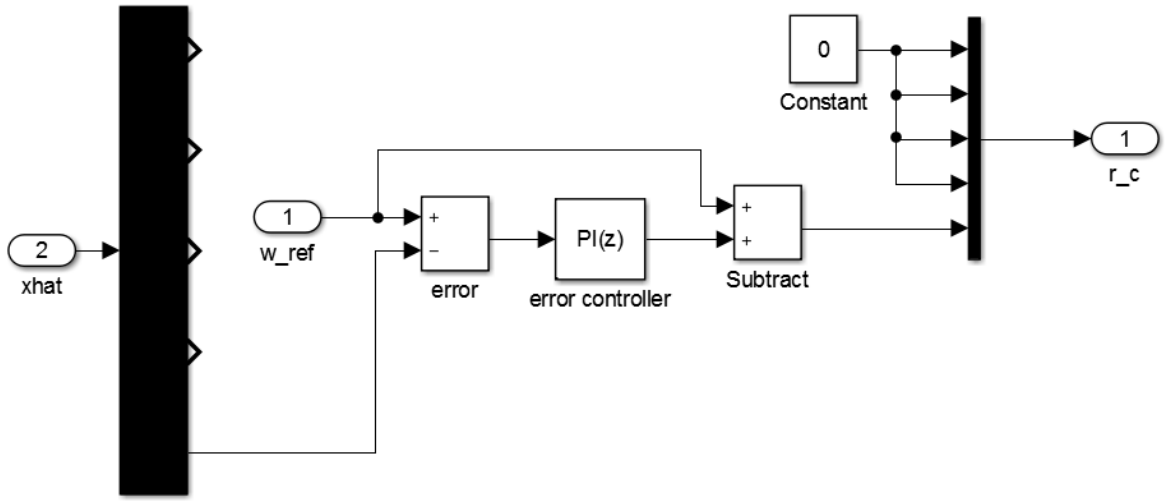
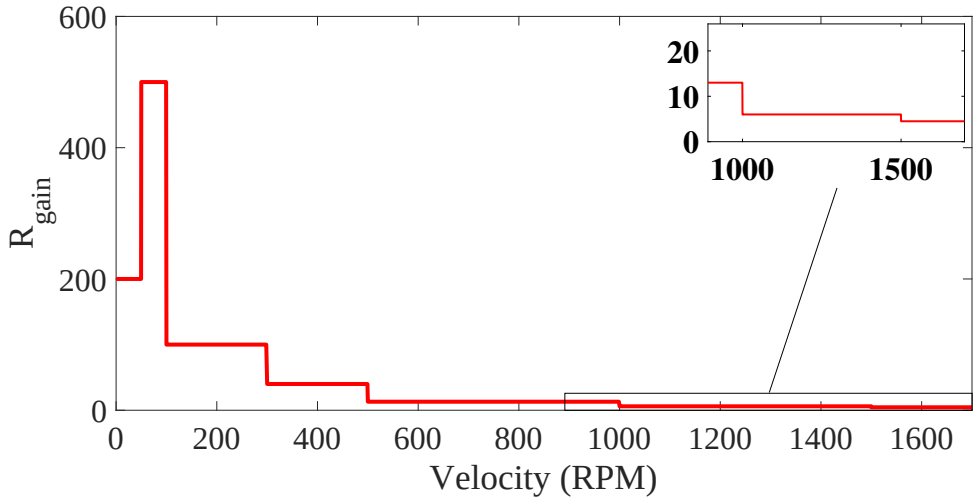


Figure 4.6: Reference Signal r_c.

To get optimal results the input weight matrix R_{LQT} changes with respect to velocity; it was found through trail and error that R_{LQT} does not have one optimal value but rather many optimal values, and this changes with velocity more than anything else. How this gain changes can be found in figure 4.7.



(a) Reference Velocity Ramp.

Figure 4.7: R_{LQT} Gain

The values in Figure 4.7 was found through trail and error; further work could be to optimize this formula even further.

4.1.4 Constant Voltage Source

The voltage source are mainly used for stress testing and comparisons with the LQT voltage. It is built in Simulink and represent the a,b and c to d and q transformation, this is shown in the matrices 4.1.4 and 4.1.4.

$$\begin{bmatrix} V_{sd} \\ V_{sq} \\ V_{sz} \end{bmatrix} = T(\theta) \cdot \begin{bmatrix} V_{sa} \\ V_{sb} \\ V_{sc} \end{bmatrix} \tag{4.1}$$

Where:

$$T(\theta) = \begin{bmatrix} cos(-\theta) & cos(-\theta + \frac{2}{3}\pi) & cos(-\theta - \frac{2}{3}\pi) \\ sin(-\theta) & sin(-\theta + \frac{2}{3}\pi) & sin(-\theta - \frac{2}{3}\pi) \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix} \tag{4.2}$$

The Simulink model is shown in Figure 4.8. By looking at this figure, one can see that the reference frame angle in constructed by integrating the reference frame velocity; for this thesis this is zero, so the reference frame angle is zero. Note that the zero component of the is not used in this thesis, only direct and quadrature components.

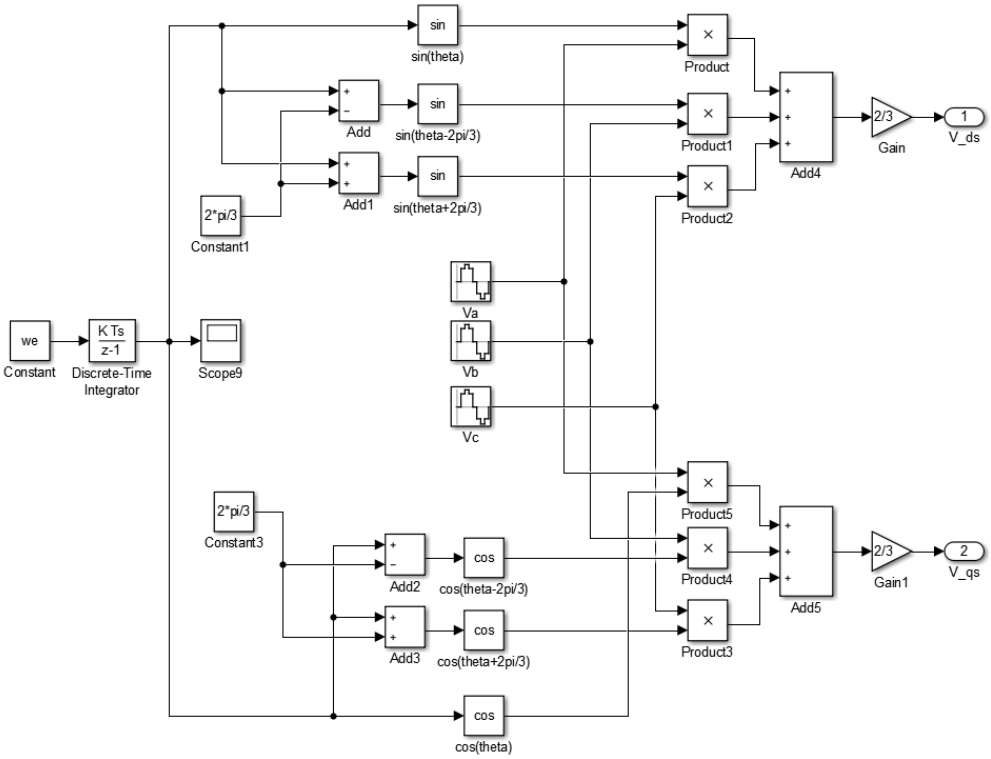


Figure 4.8: Voltage Source.

Chapter 5

Results

This chapter will show how the estimator responds real data and how robust the LQT control structure is. The reasoning behind only testing the estimator with real data is due to lack of time, as I have understood it it is quite complicated to implement advanced control structures from a simulation platform to an actual controller.

The LQT is tested on a simulation platform only and without any friction torque. The real data is collected from an physical motor placed in the mechatronics lab at the University of Agder. The motor parameters used for the digital LQT tests are different than that of the real data tests. This is because the physical motor is different from the motor used for the LQT. Changing the LQT motor parameters to match that of the real motor was not needed in my opinion, since the tests are completely separate. Note that these models does not take change in motor resistance into account; for sensor less control of induction motors this is of great importance at low speed regions [17].

Note that all simulations for this system have a sample time(T_s) of $T_s = 4e^{-5}s$ and $f_s = \frac{1}{T_s} = 25000Hz$ to give good results.

5.1 Simulation Tests

To determine how robust the control structure is, the motor is stress tested under different load and velocity scenarios. The currents and voltages produced by the motor in chapter 5.1.3 and 5.1.4 are shown in appendices C and D, respectively.

5.1.1 Motor Parameters

The motor parameters used for these tests was provided by my supervisors and are shown in table 5.1.1. It is not known how high currents and load torque this motor can handle, but looking at similar sized motors it is chosen an rough estimate for them; since these tests are limited to a digital simulation platform only the exact value the these two are not of great import as long as the controller gives out reasonable values.

P_{rated}	V_{rated}	f_{rated}	J	P	τ_{rated}	n_{rated}	I_{rated}
7.456 [kW]	460 [V]	60 [Hz]	0.05 [kgm^2]	4	35-40 [Nm]	1800 [RPM]	15-18 [A]

r_r	r_s	L_{ls}	L_{lr}	L_{ms}
0.451 [Ω]	0.6837 [Ω]	0.0050 [H]	0.0050 [H]	0.2675 [H]

5.1.2 Currents

The direct and quadrature stator currents are measured with two different voltage sources. This is done to check how high the currents from the LQT controller is; if they are too high or have large spikes it can break the motor.

Voltage Source:

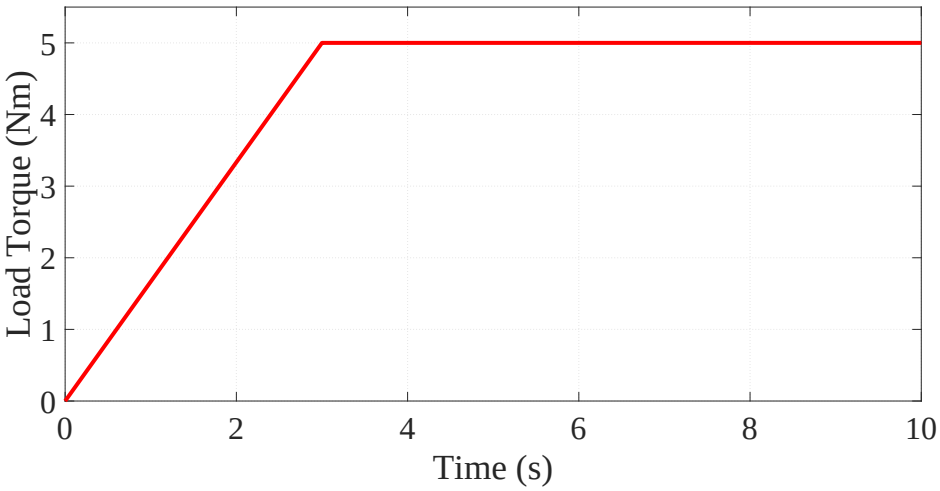
(1) Constant voltage source

(2) LQT voltage source

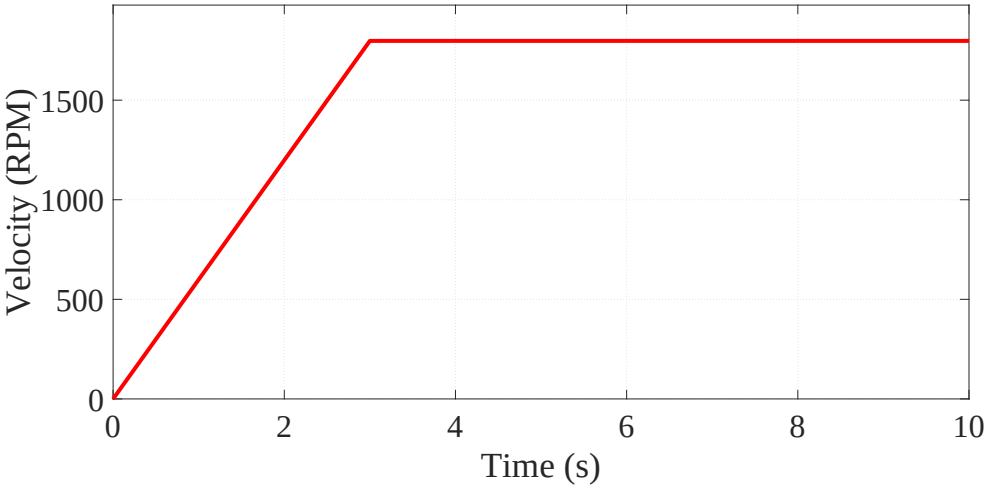
The constant voltage(1) is the maximum voltage of what the motor can handle, so this will quickly go towards the rated speed. The LQT voltage is generated only be setting the reference speed to that of the constant voltage source, this so one can get a good indication of how optimal this controller is.

Load Case One: $T_l = 5$ **Nm**

Figure 5.1 shows ramped load torque for both the voltage sources, it also shows the reference velocity for the LQT. Ramping the load torque and reference velocity is done to give the motor start-up ease compared to a stepped signal.



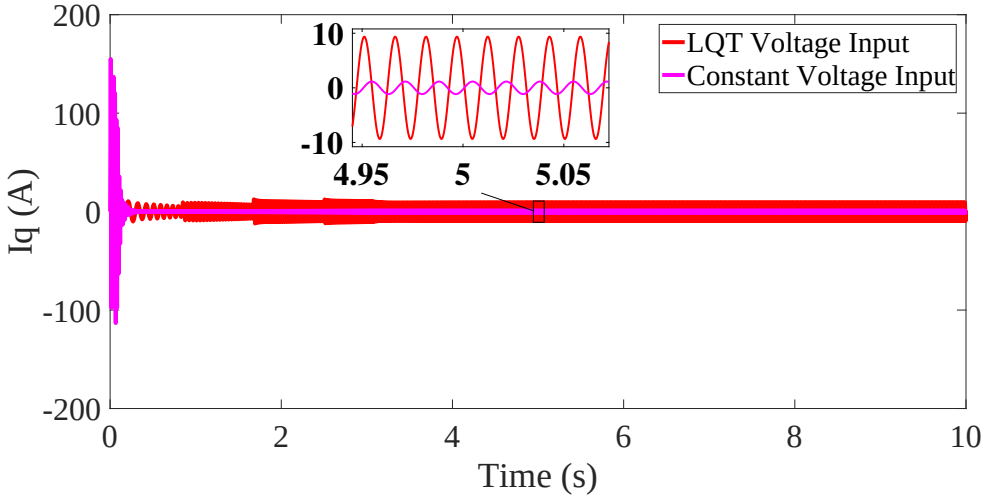
(a) Load Torque Ramp.



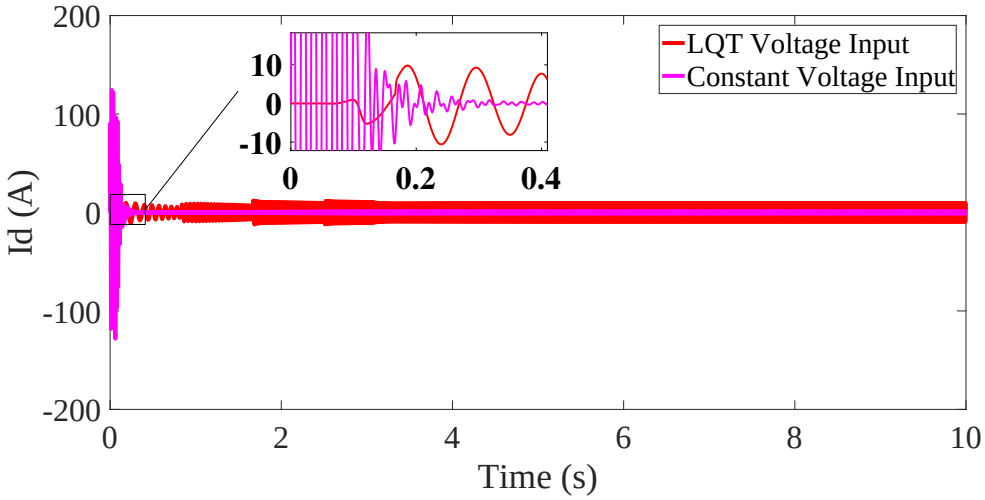
(b) Reference Velocity Ramp.

Figure 5.1: Ramp

Figure 5.2 shows the quadrature and direct stator currents for both the voltage sources. The extended zooming window in figure 5.2 shows that the currents form the LQT voltage source is generally higher than that of the constant voltage source; the currents from the LQT source does however not reach any critical values like that of the constant voltage source.



(a) Quadrature Stator Currents.

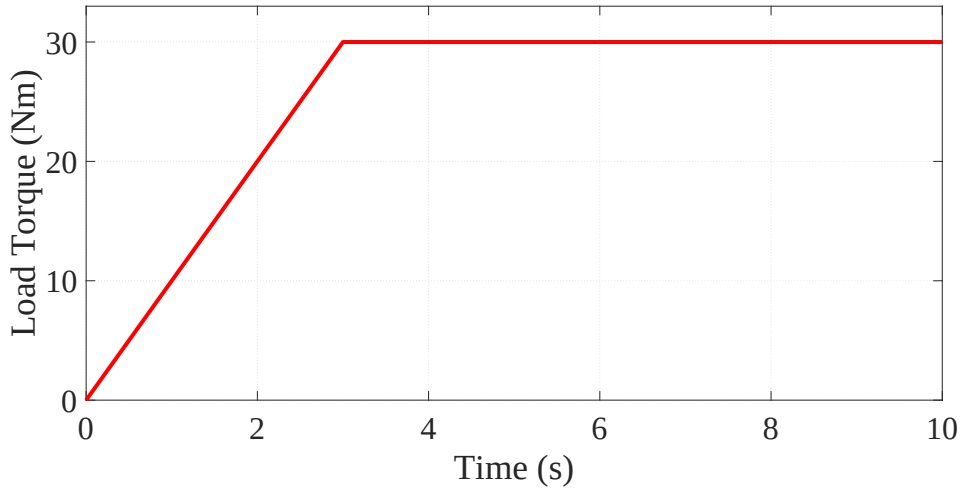


(b) Direct Stator Currents.

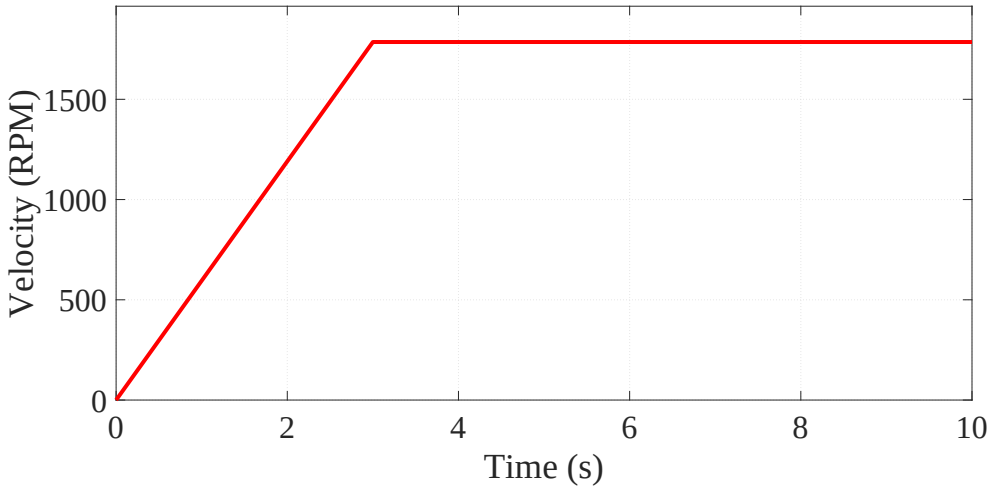
Figure 5.2: T_l =5 Nm

Load Case Two: $T_l = 30$ **Nm**

Figure 5.3 shows the ramping of load torque and reference velocity for load case two.



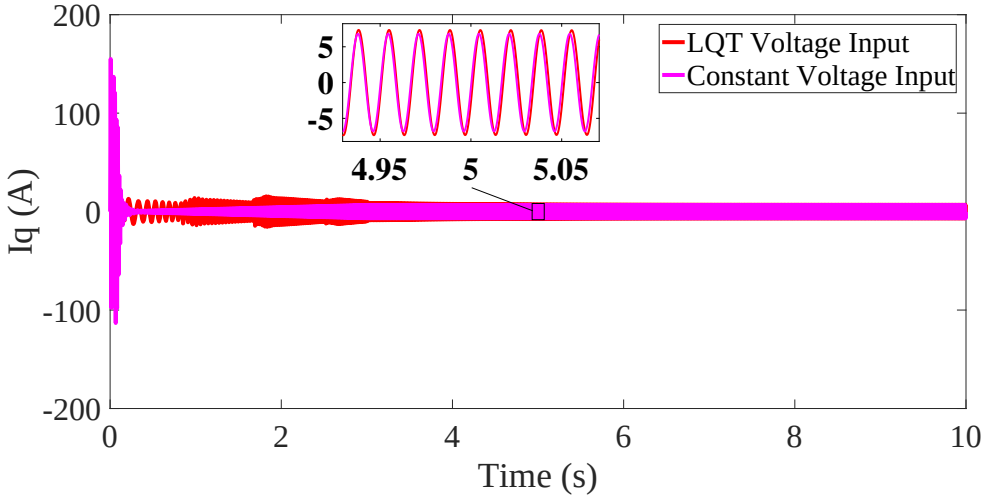
(a) Load Torque Ramp.



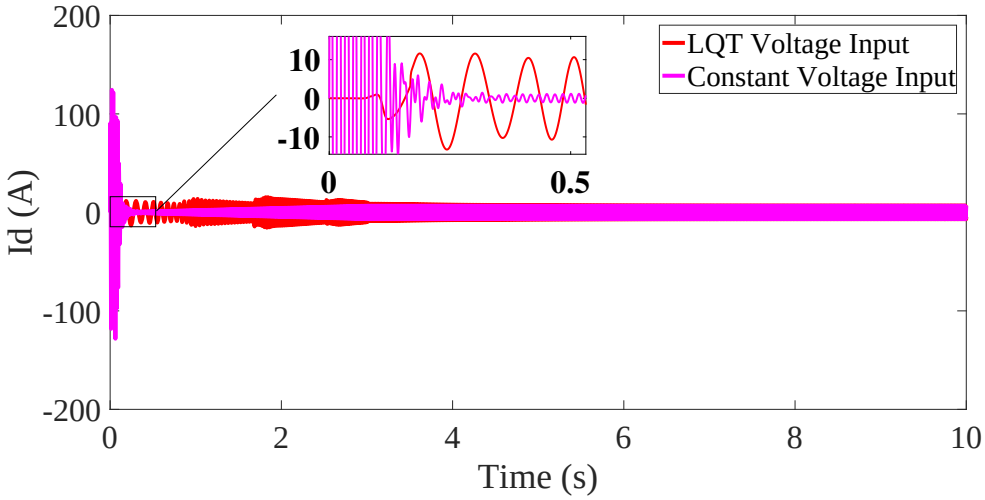
(b) Reference Velocity Ramp.

Figure 5.3: Ramp

Figure 5.4 show the currents produced by the LQT. Even at this load the currents do not reach any critical values, in addition to that they are almost identical to that of the constant voltage source. At this load, the currents produced looks more optimal than on lower loads.



(a) Quadrature Stator Currents.



(b) Direct Stator Currents.

Figure 5.4: T_l =30 NM

5.1.3 Velocity

Velocity stress tests are done in three scenarios; ramping, light steps and heavy steps. The tests are done to show how the motor reacts under stress. The applied load torque for these tests will ramp up too 20 Nm (30 Nm for ramp) in the period of three second; actual and estimated -torques are also added to show what torques the motor produces under these testes, this is shown in figure 5.5, 5.7 and 5.9 for ramp, light step and heavy step, respectively.

Velocity Test 1, Ramp

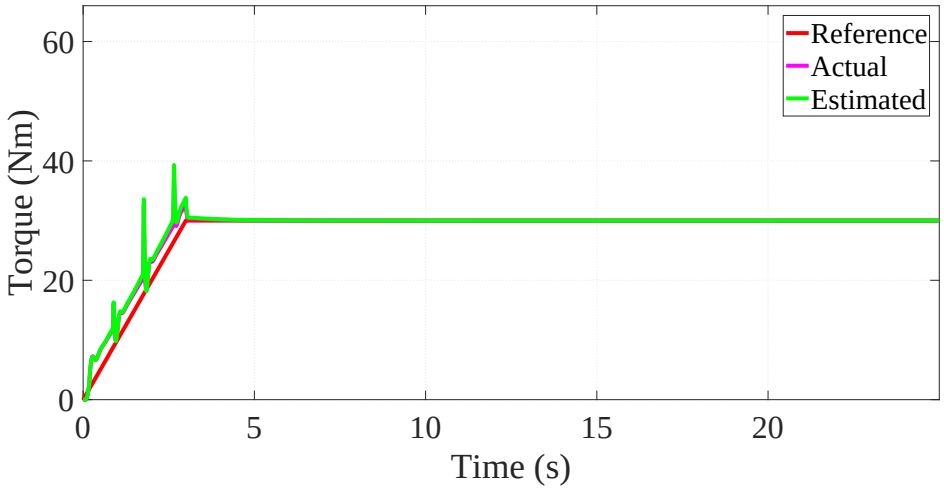


Figure 5.5: Velcoty Test 1, Torque.

A ramp is provided as a reference velocity signal where the motor should reach a final velocity of 1700 RPM over a period of three seconds. This is shown in figure 5.6.

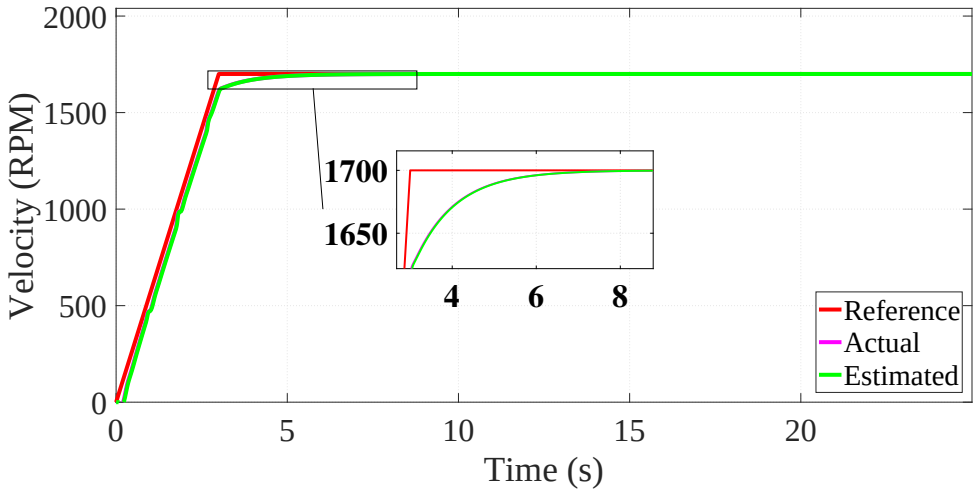


Figure 5.6: Velcoty Test 1, Velocity.

Figure 5.6 shows reference, actual and estimated velocity; one can see that the estimation and the actual velocity is responding quite quickly and settling in a acceptable time.

Velocity Test 2, Light Step

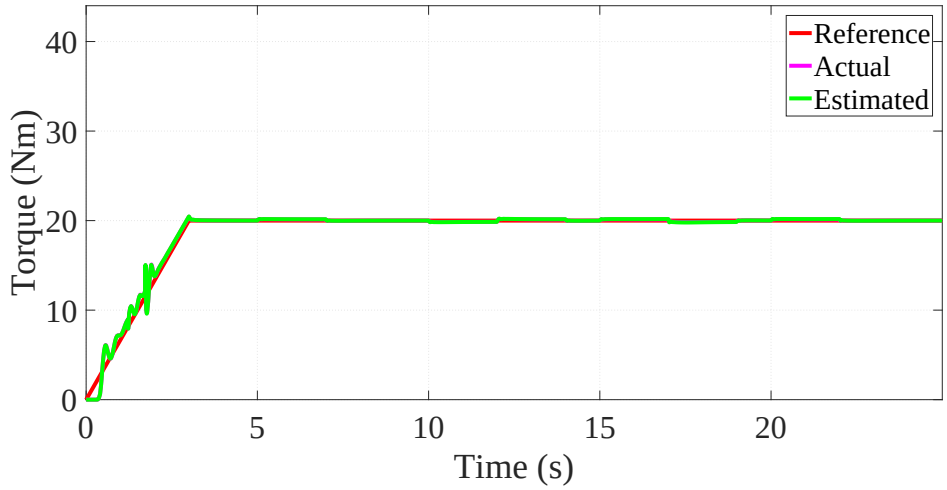


Figure 5.7: Velcoty Test 2, Torque.

Figure 5.8 shows a number of steps as a reference velocity signal where the motor ramps up too a velocity of 1700 RPM over a period of three seconds. For every 2.5 seconds the motor steps up or down with a magnitude of 50 RPM over a period of 25 seconds, where it stabilises on 300 RPM. These steps are meant to show how well the motor behaves under step velocity.

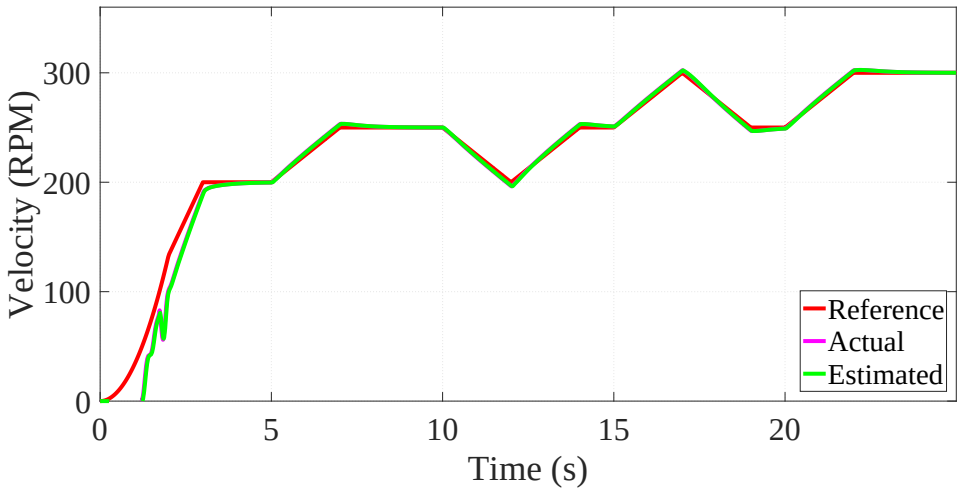


Figure 5.8: Velocity, Light Step.

Figure 5.8 shows that the estimation and the actual velocity is responding quite quickly, settling in a acceptable time and handling the steps well.

Velocity Test 2, Heavy Step

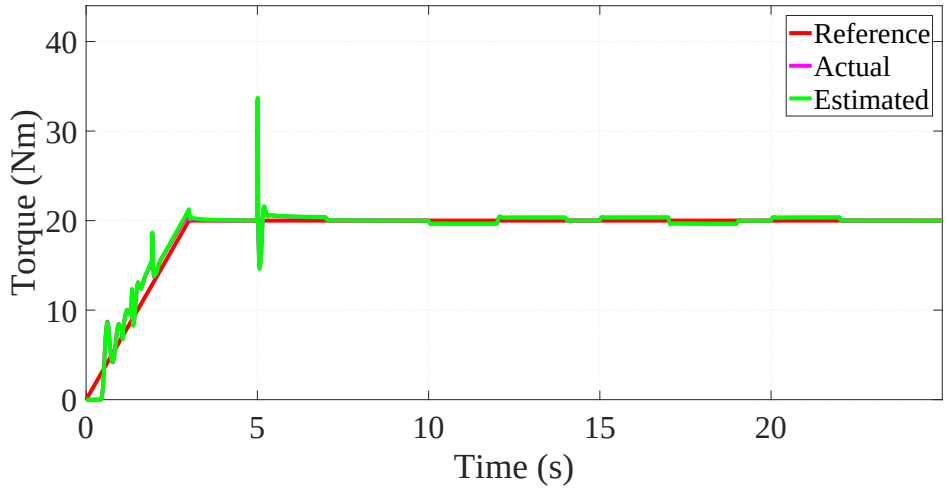


Figure 5.9: Velcoty Test 3, Torque.

Similar as in the light steps, Figure 5.10 shows a number of steps as a reference velocity signal where the motor ramps up too a velocity of 500 RPM over a period of three seconds. For every 2.5 seconds the motor steps up or down with a magnitude of 125 RPM over a period of 25 seconds, where it stabilises on 750 RPM. These steps are meant to show how well the motor behaves under heavy step velocity.

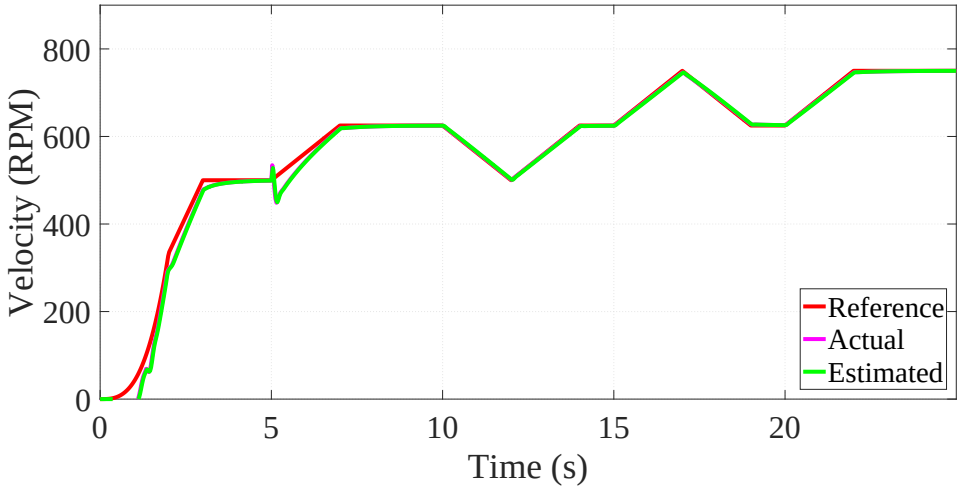


Figure 5.10: Velcoty Test 3, Velocity.

Figure 5.10 shows that the estimation and the actual velocity is responding quite quickly, settling in a acceptable time and handling the heavy steps well just like in figure 5.8.

5.1.4 Torque

Torque stress tests are done in three scenarios; zero load, light steps and heavy steps. The tests are preformed to show how the motor reacts under stress. The velocity for these tests will ramp up too 1700 RPM in the period of three seconds; actual and estimated -velocities are also added to show what velocities the motor produces under these testes, this is shown in Figure 5.11, 5.13 and 5.15 for zero load, light step and heavy step, respectively.

Torque Test 1, Zero Load

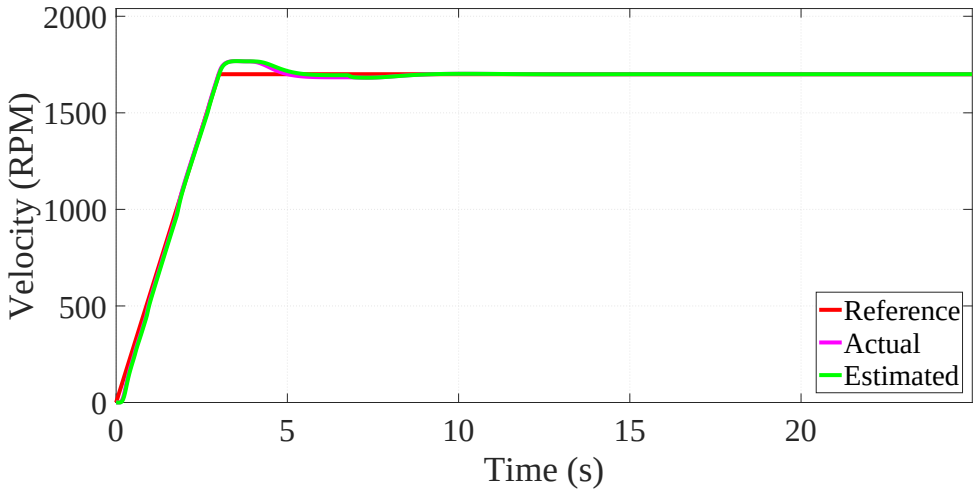


Figure 5.11: Torque Test 1, Velocity.

In this test the motor experiences zero load. This is shown in figure 5.12.

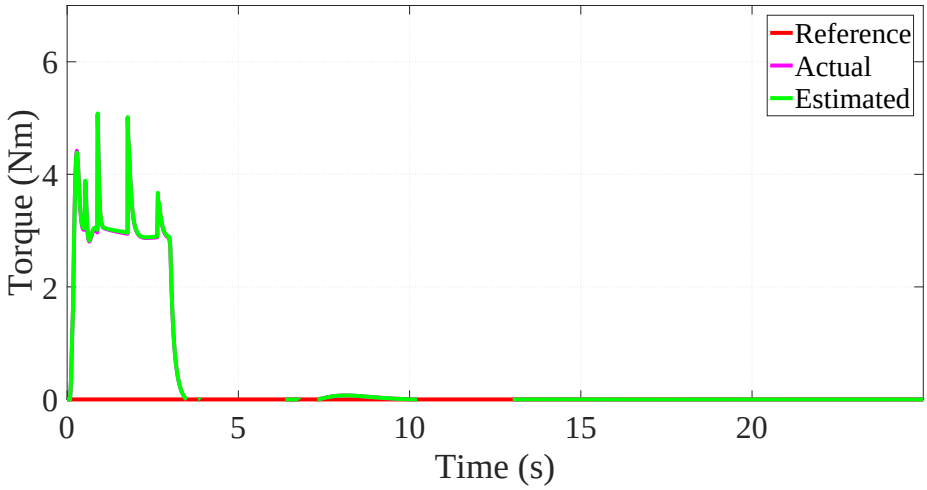


Figure 5.12: Torque Test 1, Torque.

Figure 5.12 and 5.11 shows that the motor is responding well to zero load conditions.

Torque Test 2, Light Step

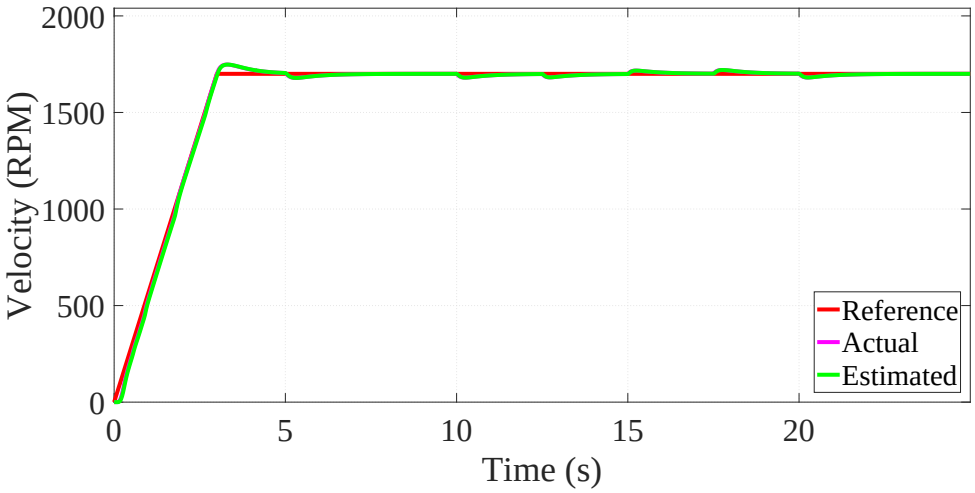


Figure 5.13: Torque Test 2, Velocity.

In this test the motor experiences light load steps; the load torque steps with a magnitude of one Nm. This is shown in Figure 5.14.

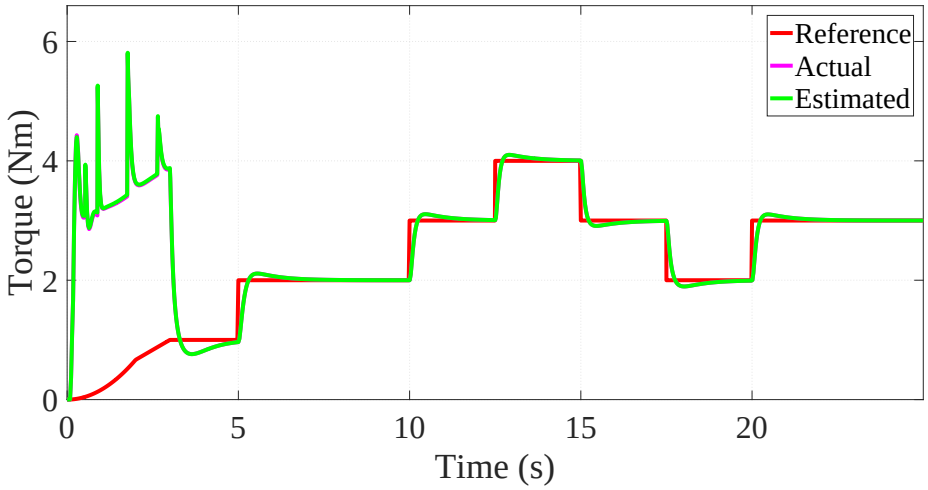


Figure 5.14: Torque Test 2, Torque.

Figure 5.14 and 5.13 shows that the motor is responding well to light load torque steps.

Torque Test 3, Heavy Step

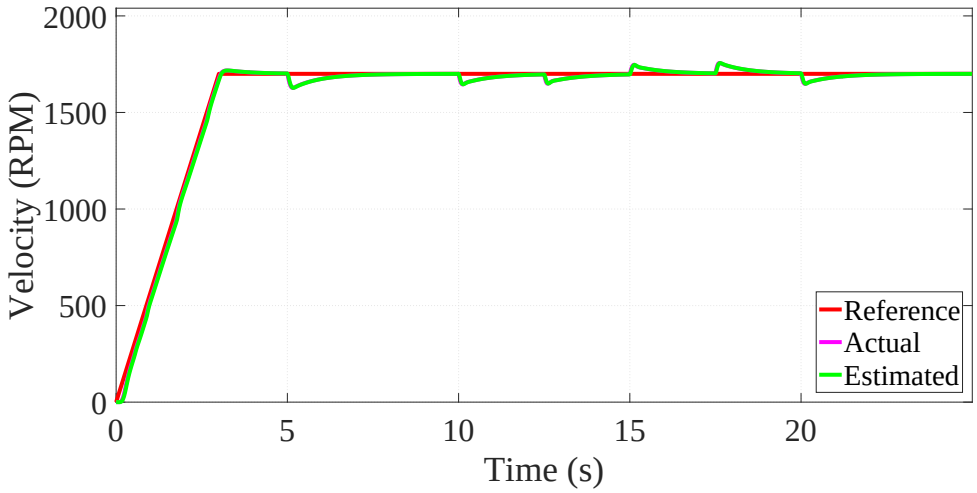


Figure 5.15: Torque Test 3, Velocity.

In this test the motor experiences heavy step; the load torque steps with a magnitude of five Nm. This is shown in Figure 5.16.

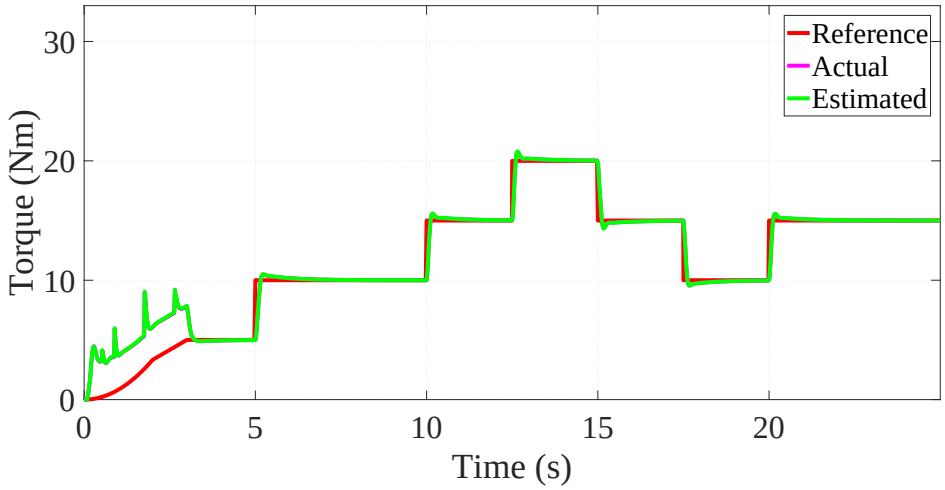


Figure 5.16: Torque Test 3, Torque.

Figure 5.14 and 5.13 shows that the motor is responding well to heavy load torque steps.

5.2 Real Data Tests

The real voltages and currents are extracted by physical sensors. These sensors are attached to the motor and plot data in real time, this data is saved arrays: time, voltage, frequency, currents, velocity and load torque. The data is then saved as time-series blocks and fed into Simulink.

5.2.1 Motor Parameters

The motor used for these tests is called M3ARF 90 S and some of its parameters was provided by my supervisors, the rest was found in ABB catalogues, this is shown in appendix E and in table 5.2.1. Note that the parameters used for this motor is extracted from SimulationX [1]. The motor set-up and the motors nameplate can be found in appendix H.

ABB's service channel is used to get the exact parameters for this motor; they have been very polite and helpful but the information they had came a bit late, so this information was not used due to lack of time.

P_{rated}	V_{rated}	f_{rated}	J	P	τ_{rated}	n_{rated}	I_{rated}
1.1 [kW]	400 [V]	50 [Hz]	0.0034 [kgm^2]	4	7.5 [Nm]	1410 [RPM]	2.59 [A]

r_r	r_s	L_{ls}	L_{lr}	L_{ms}
6.491 [Ω]	6.275 [Ω]	0.0019 [H]	0.0460 [H]	0.4878 [H]

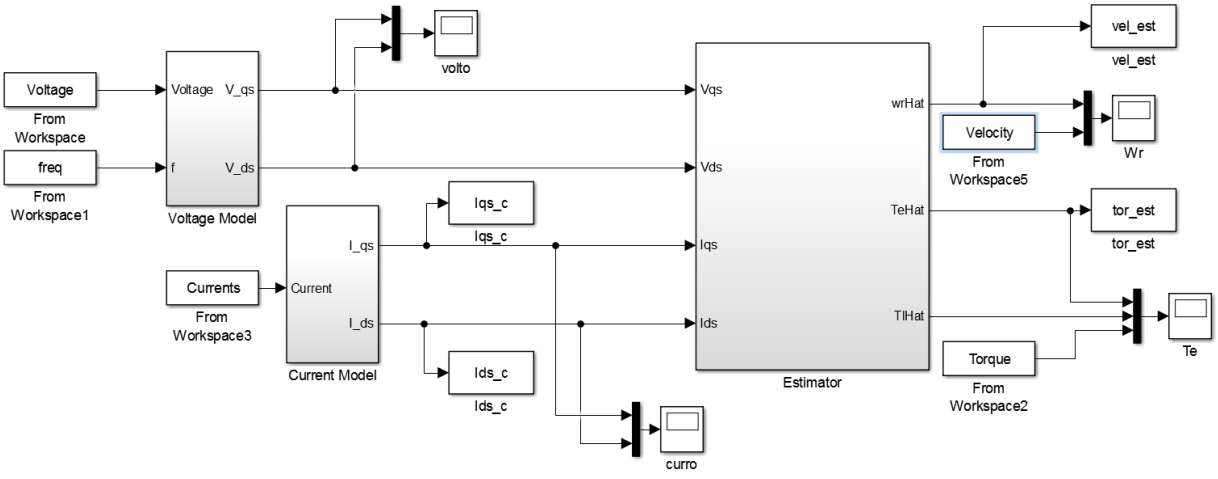


Figure 5.17: Real Data Estimator in Simulink.

Figure 5.17 show that the inputs are: "Voltage", "freq", "Currents", "Velocity" and "Torque" which is operating voltage, electric frequency, angular velocity, and load torque, respectively.

The currents that where collected in the real tests are somehow faulty in my opinion. They

[1] SimulationX is a simulation computer program.

seem too small. These currents are compared to currents generated by the model made in this thesis, where the model got the same supply voltages and load torque as the real motor; this is done too see how high the currents would get on an simulation platform. The simulated currents have an amplitude of around 2.2 A and the real currents have an amplitude of around 0.4 A, this is shown in Figure 5.18.

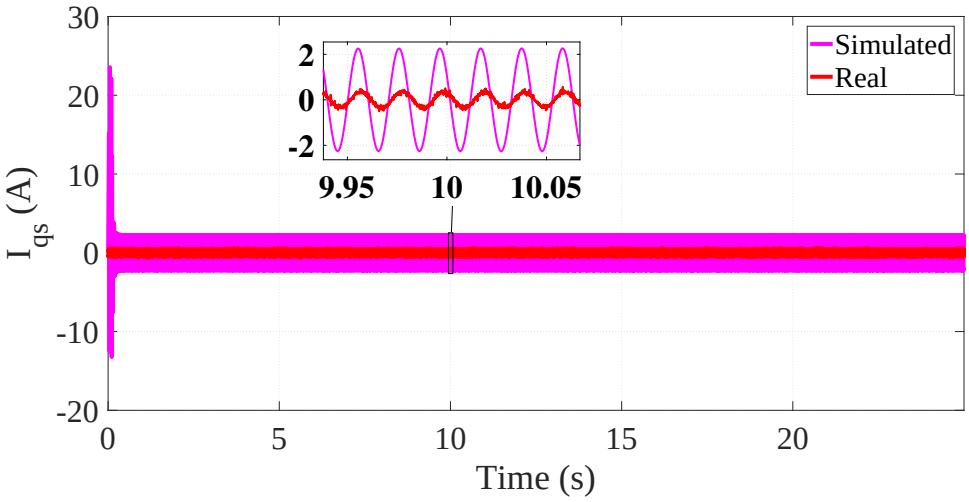


Figure 5.18: Current Comparison of Real Test Data.

It is clear that the currents do not match each other and that the lower currents are wrong due to the fact that the motor are running at around 1420 RPM and has a load torque of around 6.5 Nm which is almost at rated power; this means that the currents should also be at their rated value, which they are not. The real motor velocity and load torque can be found in appendix F.

Since I do not have time to do more tests, I am choosing not to use the faulty currents, but rather generate currents with my model using the same voltages, frequency and load torques as the actual motor got. To get the currents to look more realistic white Gaussian noise is added to them.

5.2.2 Tests

The motor is tested in three scenarios;

- (1) Constant load and velocity at 1420 RPM.
- (2) Constant load and velocity at 1000 RPM.
- (3) Variable load and velocity.

The currents produced by the simulation motor can be found in appendix G.

Test (1)

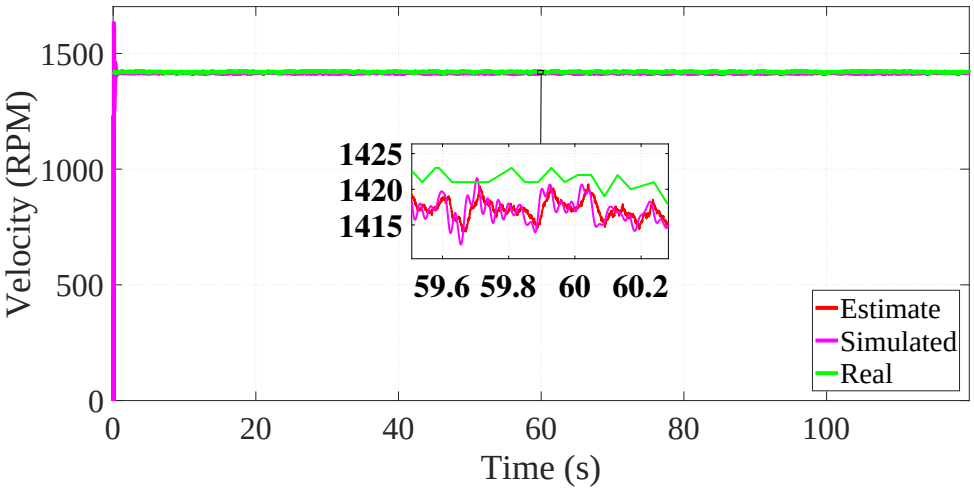


Figure 5.19: Velocity (1).

By looking at Figure 5.19 one can see that the simulated and estimated values are very similar and this indicates that the estimator is working well even with the white Gaussian noise. They do however differ a bit from the real velocity and this may be a result of non optimal motor parameters and temperature changing motor resistances.

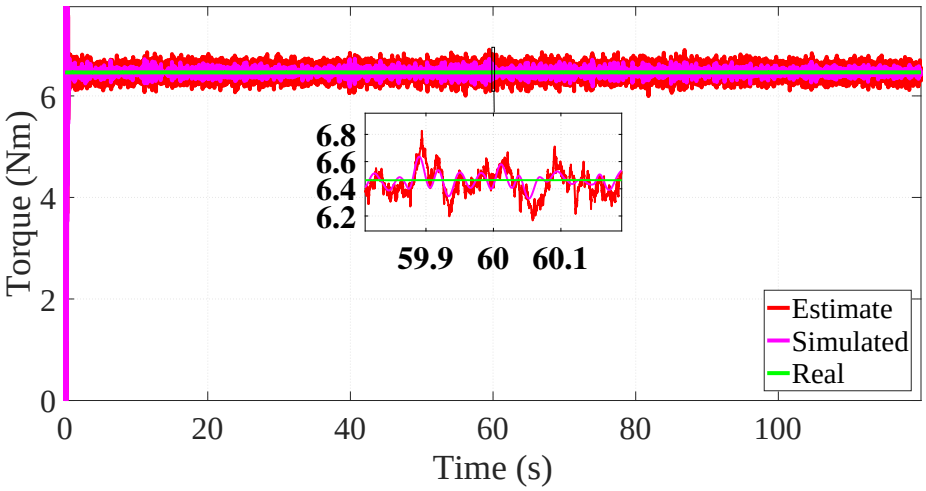


Figure 5.20: Torque (1).

Figure 5.20 shows that the estimated and simulated torques are oscillating more than the real velocity.

Test (2)

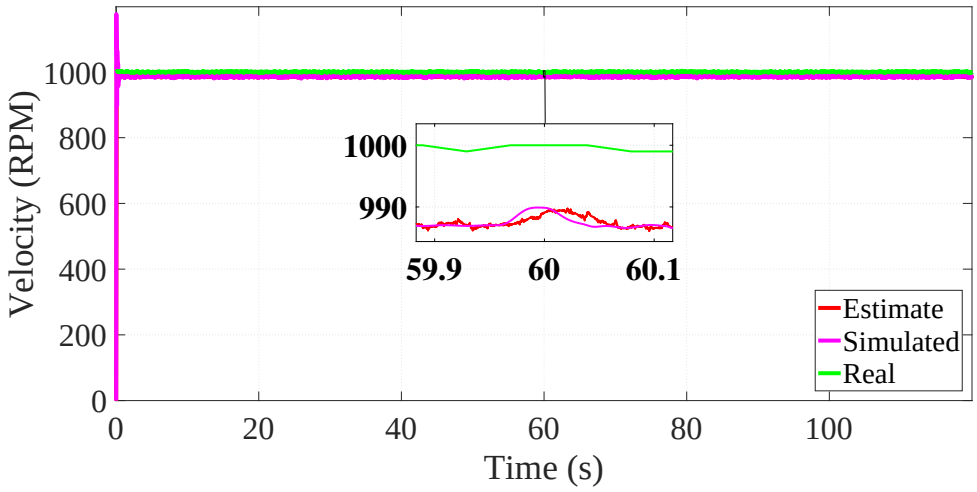


Figure 5.21: Velocity (2).

Figure 5.21 show that the simulated and estimated values are very similar and this indicates that the estimator is working well even with the white Gaussian noise. They do however differ from the real velocity even more-so than that in figure 5.19.

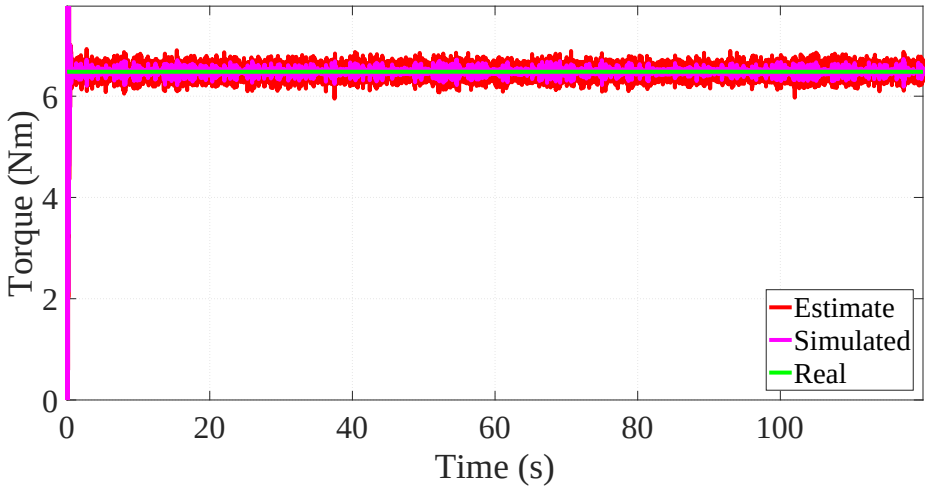


Figure 5.22: Torque (2).

Figure 5.22 show that the estimated and simulated torques are oscillating more than the real velocity.

Test (3)

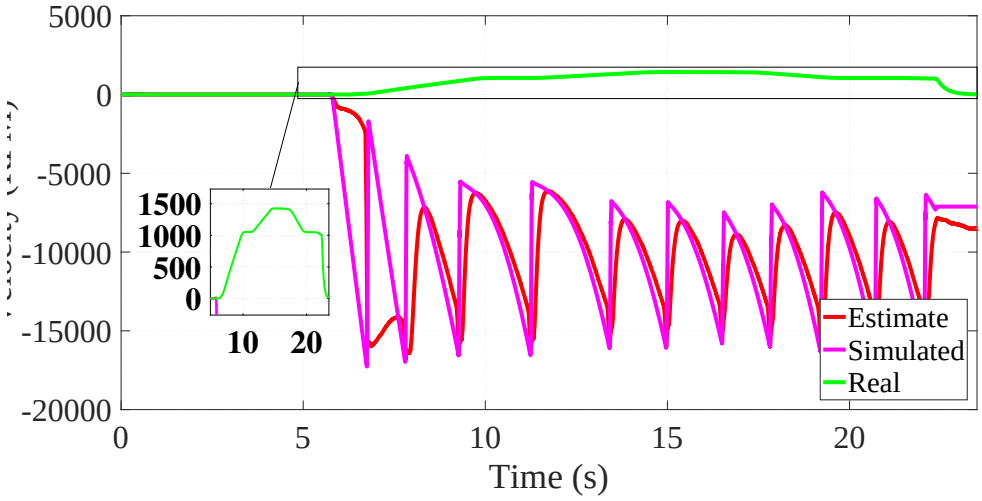


Figure 5.23: Velocity (3).

Figure 5.23 show that the simulated IM-model and the estimator is not working at all. This maybe due to faulty measured inputs or faulty motor parameters.

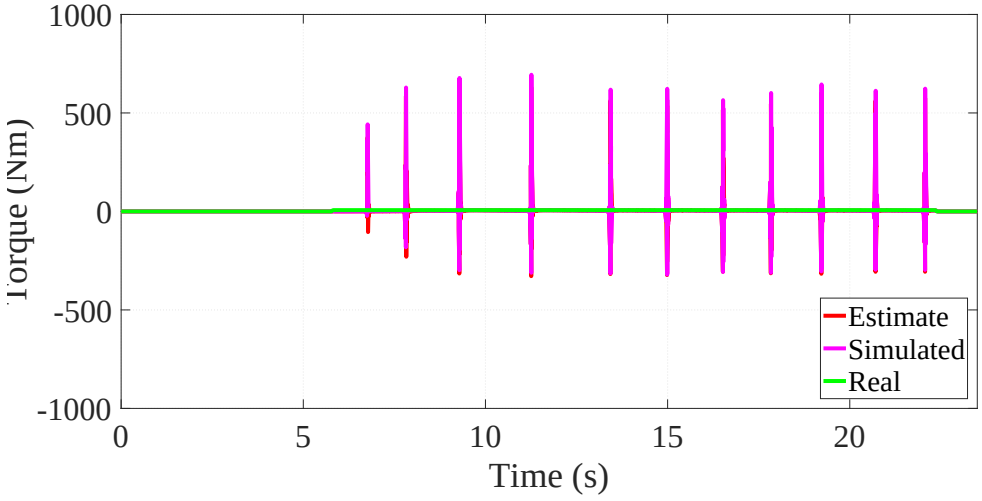


Figure 5.24: Torque (3).

Figure 5.24 have the same observations as in 5.23.

Chapter 6

Discussion

This chapter presents a detailed discussion around the main subjects of this thesis.

6.1 Model

6.1.1 Induction Motor

The IM model seems to behave like a real motor and it also seems give out realistic results by looking at the results from chapter 5.1. Chapter 5.2 show that there are something wrong with the correlation between real and simulated data. I do not think this is a model fault, but rather a fault in motor parameters or logged data. This conclusion is based on the way the real measured currents behaved and that the motor parameters where extracted from SimulationX and not real measurements. It is also noteworthy to mention that the resistances in induction machinery is changing with the state of the motor; since the resistances used for this model where constant, this could also have played a role in the faulty results.

6.1.2 Controller

The control structure of this LQT can be summed as a regular LQT with a velocity correction loop and velocity-dependent input gain weights.

Velocity Correction Loop
The velocity correction loop addition can be viewed in Figure 3.5. This loop is essential to achieve the desired velocity profile, the velocity would not follow the correct path without this loop with an exception of zero load contritions. Why this loop is essential is not clear, in my opinion the LQT should have been able to follow the correct velocity path without it; my guess is that it have something to do with the acceleration equation. The acceleration equation need two sources of information, load torque and electromagnetic torque; this controller get the information form the electromagnetic torque but not the load torque. It does not get the load torque information due to the way the B matrix is built for the controller; this is shown in chapter 3.5.

Velocity-dependent Input Gain Weights
To vary the input gain weights is of great import to control the stator current amplitude of

the motor. If the gain id high at low velocities the currents would behave as they should, but if the same gain value is implemented at high velocities the currents would be too high and potentially harm the motor in an real scenario. So trough trail and error it was found that the input gain weights needed to change with respect to velocity. How this changes is shown in chapter 4.1.3.

Linearization
Like mentioned in chapter 3.5.4 the controller tests showed that the controller worked just as well with a non-linearized full model as a linearized full model. This could mean that the non-linearized model is linearized, but this is another subject, although very interesting.

Tuning
The tuning of this controller is not as simple as first believed. The varying inputs gains, the velocity correction loop PI-gains and the internal state gains creates a lot of possibilities for tuning this controller, this can be both a positive and negative function.

6.1.3 Estimator

The estimator is working very well in my opinion. The tuning is very simple in my opinion, even with distorted data, it is not hard to get on good tuning values and can easily be done manually, atleast for the work done in this thesis. In my opinion, the most important subject regarding how well this estimator is working, is how well the motor parameters is chosen; if the estimator motor parameters do not change with different temperatures or other varying factors the estimator calculations will be a bit off.

6.2 Further Work

6.2.1 Model

A subject of further work is to include varying stator resistances as a function of temperature to get a more accurate state estimation.

6.2.2 Controller

It would have been interesting to see how this controller behaves on a real motor, so a subject of further work could be to implement this control structure into a physical controller and test it on a real motor.

It would also be interesting to learn how the linearization work for this model, so this can also be a subject of further work.

Another topic of further work could be the investigation regarding the velocity correction loop made for this controller; why it works and why it is essential to achieve velocity control.

Lastly it could also be beneficial to optimise the varying the input gain weight function.

6.2.3 Estimator

One subject of further work could be to measure the measurement noise of the current sensor to achieve even better estimation results.

Chapter 7

Conclusion

This chapter presents an overall conclusion of the work done in this thesis.

Referring to the problem statement in chapter 1.1:

1 Successfully implement velocity control for an induction motor in Matlab/Simulink using a Kalman filter and a LQ-optimal controller.

2 Comparing performance in velocity control between EKF-based control method to a PID based control.

1) It is clear that velocity control of an induction motor have been achieved using a Kalman filter and a LQT, in Simulink and Matlab. It is not only achieved, it is achieved quite successfully in my opinion.

2) The work in this thesis is quite extensive and time consuming, implementing the control structure to a physical controller and then testing it and comparing it to a already existing PID-based controller is not achieved due to lack of time.

Bibliography

[1] W. Pawlus, S. T. Kandukuri, G. Hovland, M. Choux, and M. R. Hansen, "Ekf-based estimation and control of electric drivetrain in offshore pipe racking machine," in *Industrial Technology (ICIT), 2016 IEEE International Conference on*. IEEE, 2016, pp. 153–158.

[2] E. Todorov and W. Li, "A generalized iterative lqg method for locally-optimal feedback control of constrained nonlinear stochastic systems," in *American Control Conference, 2005. Proceedings of the 2005*. IEEE, 2005, pp. 300–306.

[3] N. Mohan, *Electric Machines and Drives*. Wiley Global Education, 2011.

[4] E. Csanyi. Construction of 3-phase ac induction motors. [Online]. Available: http://electrical-engineering-portal.com/construction-of-3-phase-ac-induction-motors

[5] O. I. Elgerd and P. D. van der Puije, "Induction machines," in *Electric Power Engineering*. Springer, 1998, pp. 326–385.

[6] J. Johnson. (2015) Hydraulic-electric analogies: Power sources. [Online]. Available: http://hydraulicspneumatics.com/hydraulic-pumps-motors/hydraulic-electric-analogies-power-sources-part-3

[7] J. Bednarczyk. Induction motor theory. [Online]. Available: http://www.pdhonline.com/courses/e176/e176content.pdf

[8] T. I. Europe, "Field orientated control of 3-phase ac-motors," *Literature Number: BPRA073*, 1998.

[9] P. C. Krause and C. Thomas, "Simulation of symmetrical induction machinery," *IEEE transactions on power apparatus and systems*, vol. 84, no. 11, pp. 1038–1053, 1965.

[10] R. Kalman, "On the general theory of control systems," *IRE Transactions on Automatic Control*, vol. 4, no. 3, pp. 110–110, 1959.

[11] R. E. Kalman *et al.*, "Contributions to the theory of optimal control," *Bol. Soc. Mat. Mexicana*, vol. 5, no. 2, pp. 102–119, 1960.

[12] K. Ogata and Y. Yang, "Modern control engineering fifth edition," 1970.

[13] G. Welch and G. Bishop, "An introduction to the kalman filter," 1995.

[14] D. P. Bertsekas, D. P. Bertsekas, D. P. Bertsekas, and D. P. Bertsekas, *Dynamic programming and optimal control*. Athena Scientific Belmont, MA, 1995, vol. 1, no. 2.

[15] B. D. Anderson and J. B. Moore, *Optimal control: linear quadratic methods*. Courier Corporation, 2007.

[16] R. Gunabalan, V. Subbiah, and B. R. Reddy, "Sensorless control of induction motor with extended kalman filter on tms320f2812 processor," *measurement*, vol. 500, p. 7, 2009.

[17] V. Vasic, S. N. Vukosavic, and E. Levi, "A stator resistance estimation scheme for speed sensorless rotor flux oriented induction motor drives," *IEEE Transactions on Energy Conversion*, vol. 18, no. 4, pp. 476–483, 2003.

Appendices

Appendix A

Appendix A

```
function K_add = fcn(yhat,xhat,y)

persistent FS Pk Kk
if isempty(FS)
    FS = true;
    Kk = zeros(5,5);
    Pk = diag(ones(5,1));
end
x = zeros(5,1);
if ~isempty(FS)
   x = xhat;
end

P_Q=1e-6;
P_R=1e-6;
  Q = diag([P_Q P_Q P_Q P_Q 20]);
  R = diag([P_R P_R]);
Rs = 0.6837; %Ohm, Stator Resistance%
Rr = 0.451; %Ohm, Rotor Resistance%
Llr = 0.004152; %H, Rotor leakage inductance%
Lls = 0.004152; %H, Stator leakage inductance%
Lm = 0.1486; %H, Mutual inductance%

M=3/2*Lm;
P=2;
J = 0.05; %Kg/m2, Inertia%
p = 4; %poles%
f_motor = 60; %Hz, operating frequency%


we=0;
wb      =  2*pi*f_motor;   %electrical velocity
Xls     =wb*Lls;           %Stator leakage impedance
Xlr     =wb*Llr;           %Rotor leakage impedance
Xm      =wb*Lm;            %Magnetizing impedance
Xmq = 1/(1/Xm+1/Xls+1/Xlr);
Xmd = Xmq;

```

```
Fmeasure = 25000;
deltaT = 1/Fmeasure;
 n=3;


phi_qs = x(1); phi_ds = x(2); phi_qr = x(3); phi_dr = x(4); wr = x(5);

%%Linearized Statespace
A_c = zeros(5,5);
A_c = wb*[ Rs/Xls*(Xmq/Xls-1)        0             Rs*Xmq/(Xls*Xlr)
|0                        0;
             0                  Rs/Xls*(Xmd/Xls-1)          0              Rs*Xmd/(Xls*
|0;
             Rr*Xmq/(Xlr*Xls)         0             Rr/Xlr*(Xmq/Xlr-1)
|wr/wb              phi_dr/wb;
             0                  Rr*Xmd/(Xlr*Xls)         -wr/wb             Rr/Xlr*(Xmd/
|-phi_qr/wb;
             0                           0                    0
|0                        0];
 F = zeros(5,5);
 F = eye(5)+A_c*deltaT+(A_c*deltaT)^2/2+(A_c*deltaT)^3/6;
 C= zeros(2,5);

C = [1/Xls*(1-Xmq/Xls) 0 -Xmq/(Xls*Xlr) 0 0;
     0 1/Xls*(1-Xmd/Xls) 0 -Xmd/(Xls*Xlr) 0;];

Pk = F*Pk*F'+Q;

Kk = Pk*C'/(C*Pk*C'+R);

error = y-yhat;

K_add = Kk*error;

Pk = (eye(5)-Kk*C)*Pk;

end
```

Appendix B

Appendix B

```
function [Uhat, R_gain]= fcn(xhat,ref)

linni=0; % if set to 1, the A matrix is linearized, if set to 0, then A is not linear

persistent FS uhat Pl P_R
if isempty(FS)
    Pl=ones(5,5);
    FS = true;
    P_R =100;
end
x = zeros(5,1);
if ~isempty(FS)
   x = xhat;
   P_R =100;
end
if ref<=50
P_R = 200;
elseif ref <=100
P_R=500;
elseif ref <=300
P_R=100;
elseif ref <=500
P_R=40;
elseif ref <=1000
P_R=13;
elseif ref <=1500
P_R=6;
elseif ref <=1800
P_R=4.5;
else
    P_R =1;
end;
  R_gain=P_R;
P_Q=1;
 Q=[P_Q 0 0 0 0;
    0 P_Q 0 0 0;
    0 0 P_Q 0 0;
    0 0 0 P_Q 0;
```

```
    0 0 0 0 1;]; %agressive poleplacement

R=[P_R 0;
   0 P_R;];
Rs = 0.6837; %Ohm, Stator Resistance%
Rr = 0.451; %Ohm, Rotor Resistance%
Llr = 0.004152; %H, Rotor leakage inductance%
Lls = 0.004152; %H, Stator leakage inductance%
Lm = 0.1486; %H, Mutual inductance%

M=3/2*Lm;
P=4;
J = 0.05; %Kg/m2, Inertia%
f = 0.005752; %friction coeff%
p = 4; %poles%
f_motor = 60; %Hz, operating frequency%


we=0;
wb      =  2*pi*f_motor; %2*pi*f_motor;          %Base speed
Xls     =wb*Lls;           %Stator leakage impedance
Xlr     =wb*Llr;           %Rotor leakage impedance
Xm      =wb*Lm;            %Magnetizing impedance
Xmq = 1/(1/Xm+1/Xls+1/Xlr);
Xmd = Xmq;

Fmeasure = 25000;
deltaT = 1/Fmeasure;
n=3;

psi_qs = x(1);
psi_ds = x(2);
psi_qr = x(3);
psi_dr = x(4);
wr = x(5);
k_eq1=n*p/4;
k_eq2=p/(2*J);
k_eq=k_eq1*k_eq2;
Xhat=[x(1), x(2), x(3), x(4), x(5)]';
if linni == 1
%%Linearized Statespace
Psi_qs = (Xmq*k_eq*psi_dr)/(Xlr*Xls*wb); % linearized
Psi_ds =-(Xmd*k_eq*psi_qr)/(Xlr*Xls*wb);
Psi_qr =-(k_eq*(Xlr*Xmd*psi_ds + Xls*Xmd*psi_dr - Xls*Xmq*psi_dr))/(Xlr^2*Xls*wb);
Psi_dr = (k_eq*(Xlr*Xmq*psi_qs - Xls*Xmd*psi_qr + Xls*Xmq*psi_qr))/(Xlr^2*Xls*wb);
A_c = zeros(5,5);
A_c = wb*[ Rs/Xls*(Xmq/Xls-1)        0             Rs*Xmq/(Xls*Xlr)
0                    0;
            0                  Rs/Xls*(Xmd/Xls-1)          0              Rs*Xmd/(Xls*
0;
            Rr*Xmq/(Xlr*Xls)         0             Rr/Xlr*(Xmq/Xlr-1)
wr/wb             psi_dr/wb;
            0                  Rr*Xmd/(Xlr*Xls)         -wr/wb            Rr/Xlr*(Xmd/
-psi_qr/wb;
            Psi_qs                   Psi_ds               Psi_qr
Psi_dr                 0;];
```

```
90
91  else
92  Psi_qs =Xlr*Xmq*psi_dr*k_eq/(Xlr^2*Xls*wb); %non linearized
93  Psi_ds =-Xlr*Xmd*psi_qr*k_eq/(Xlr^2*Xls*wb);
94  Psi_qr =Xls*Xmq*psi_dr*k_eq/(Xlr^2*Xls*wb);
95  Psi_dr =-Xls*Xmd*psi_qr*k_eq/(Xlr^2*Xls*wb);
96
97
98  A_c = zeros(5,5);
99  A_c = wb*[ Rs/Xls*(Xmq/Xls-1)        0             Rs*Xmq/(Xls*Xlr)
    |0                      0;
100             0                  Rs/Xls*(Xmd/Xls-1)          0              Rs*Xmd/(Xls*
    |0;
101             Rr*Xmq/(Xlr*Xls)         0             Rr/Xlr*(Xmq/Xlr-1)
    |wr/wb                    0;
102             0                  Rr*Xmd/(Xlr*Xls)         -wr/wb             Rr/Xlr*(Xmd/
    |0;
103             Psi_qs                    Psi_ds               Psi_qr
    |Psi_dr                  0;];
104 end;
105  A = zeros(5,5);
106  A = eye(5)+A_c*deltaT+(A_c*deltaT)^2/2+(A_c*deltaT)^3/6;
107
108  B_c=[wb 0;
109       0 wb;
110       0  0;
111       0  0;
112       0  0;];
113
114 B=zeros(5,2);
115 B = B_c*deltaT;
116
117 Pl=Q+A'*Pl*A-A'*Pl*B*pinv(R+B'*Pl*B)*(B'*Pl*A);
118 k=pinv(R+B'*Pl*B)*B'*Pl*A;
119
120 uhat=-k*Xhat;
121 Uhat=uhat;
122 end
```

Appendix C

Appendix C

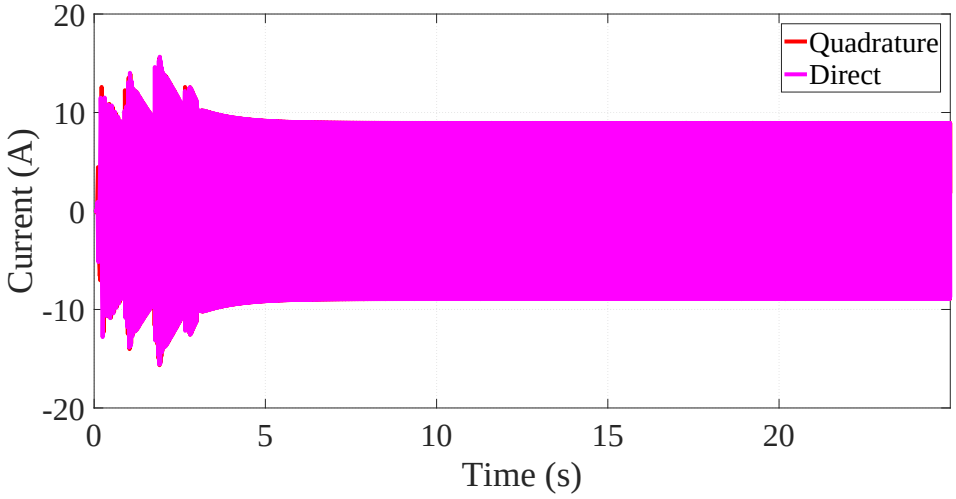


Figure C.1: Velocity Test 1, Current.

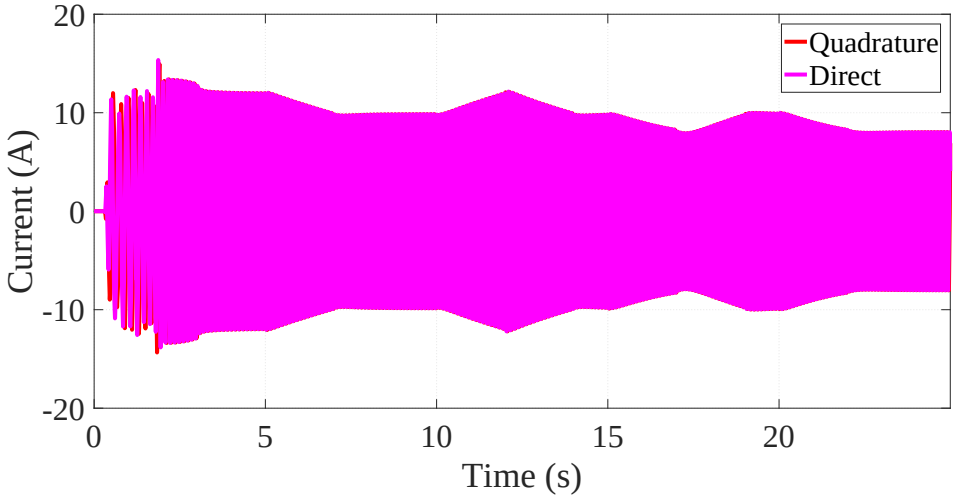


Figure C.2: Velocity Test 2, Current.

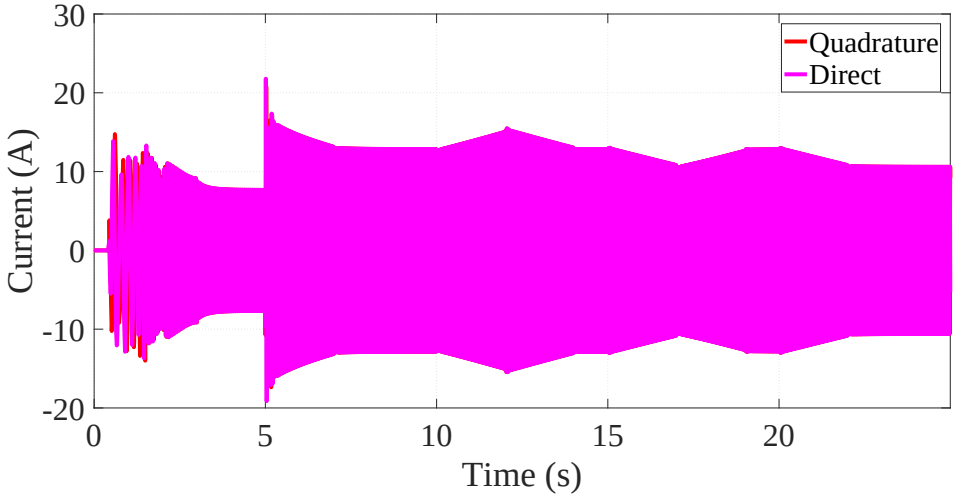


Figure C.3: Velocity Test 3, Current.

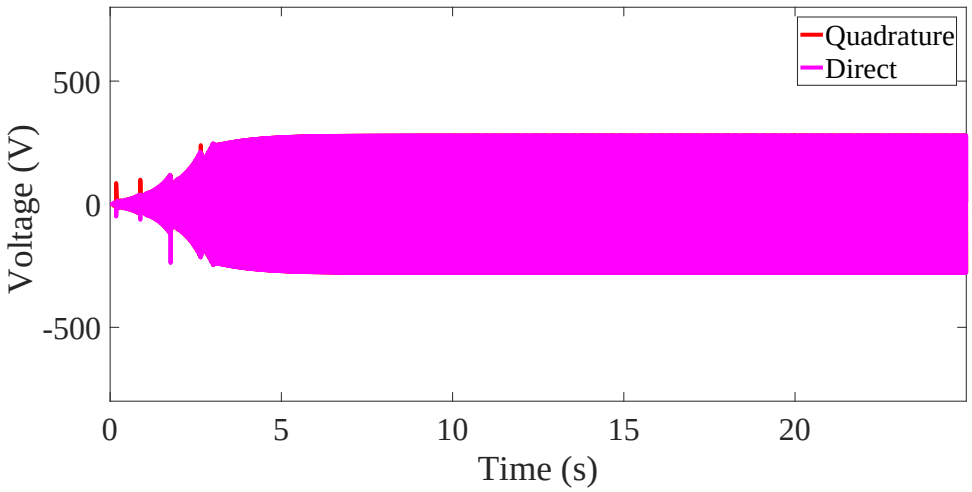


Figure C.4: Velocity Test 1, Voltage.

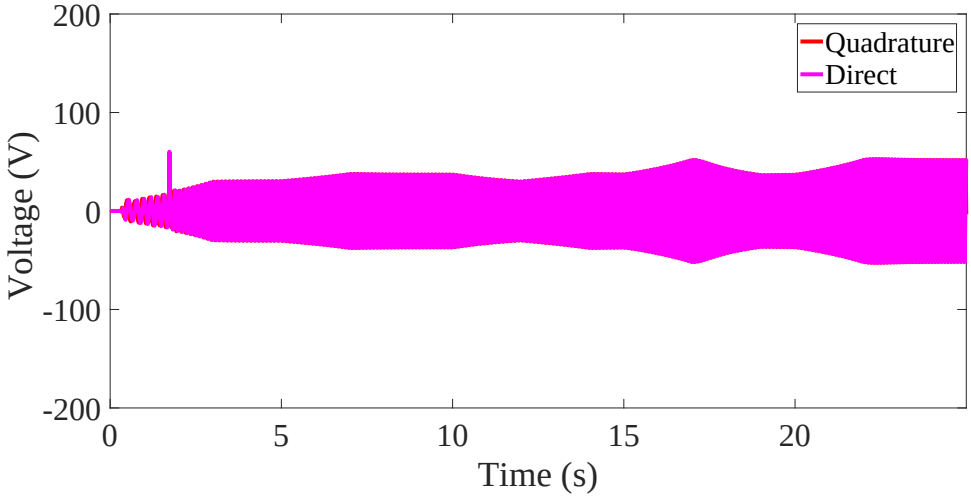


Figure C.5: Velocity Test 2, Voltage.

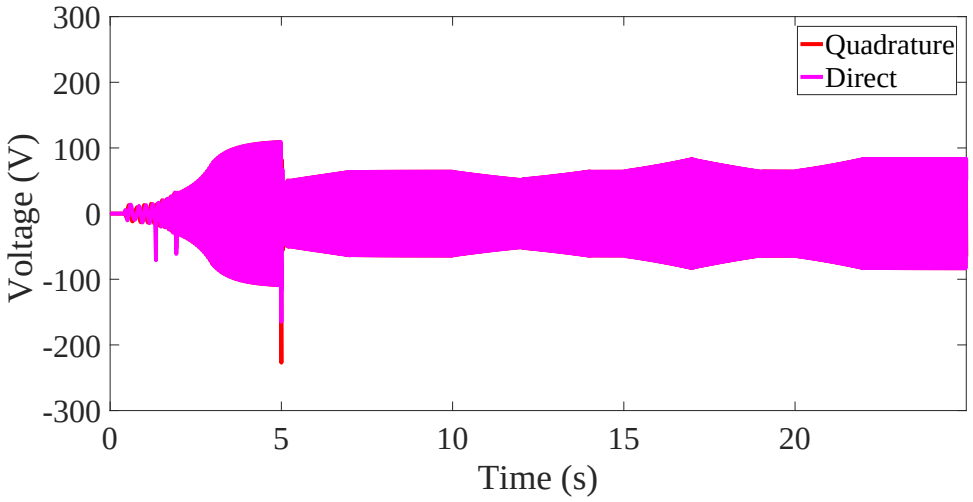


Figure C.6: Velocity Test 3, Voltage.

Appendix D

Appendix D

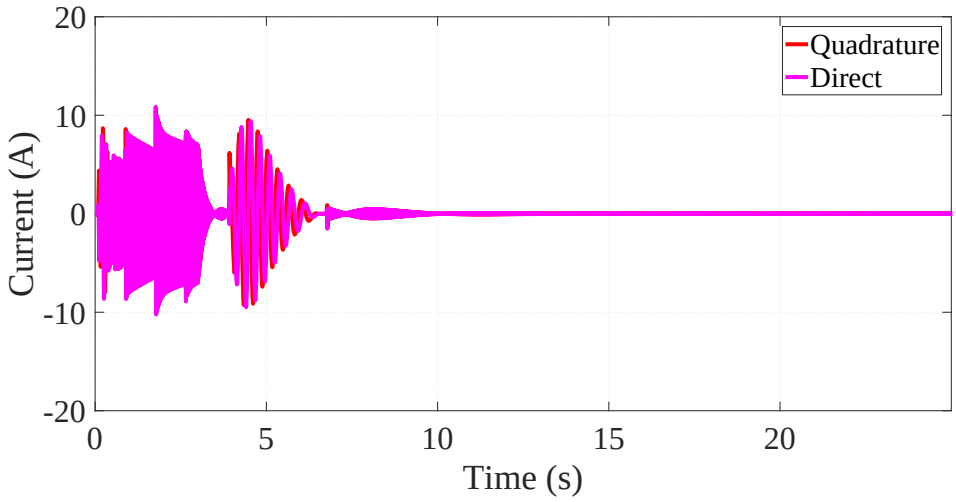


Figure D.1: Torque Test 1, Current.

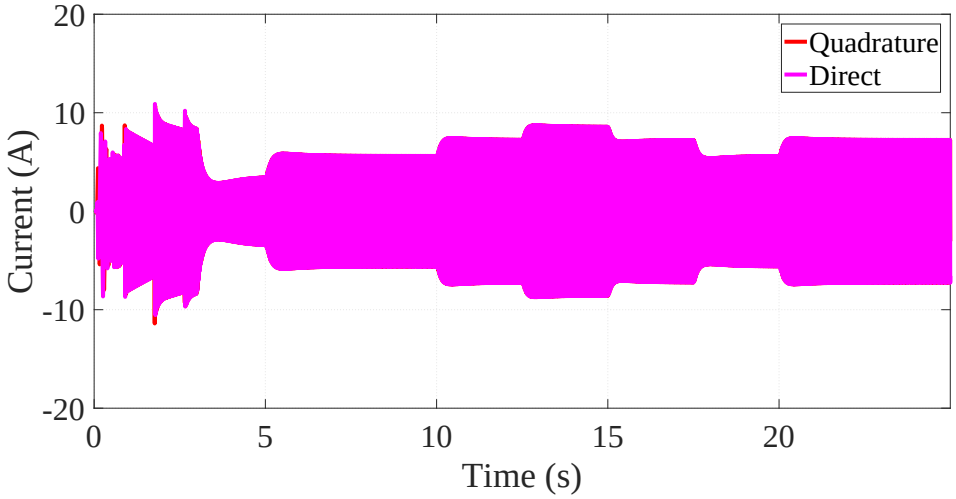


Figure D.2: Torque Test 2, Current.

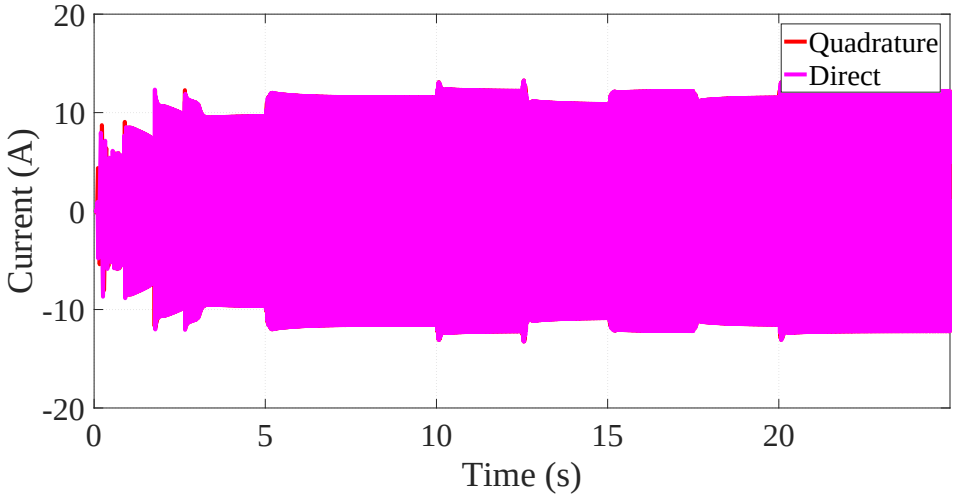


Figure D.3: Torque Test 3, Current.

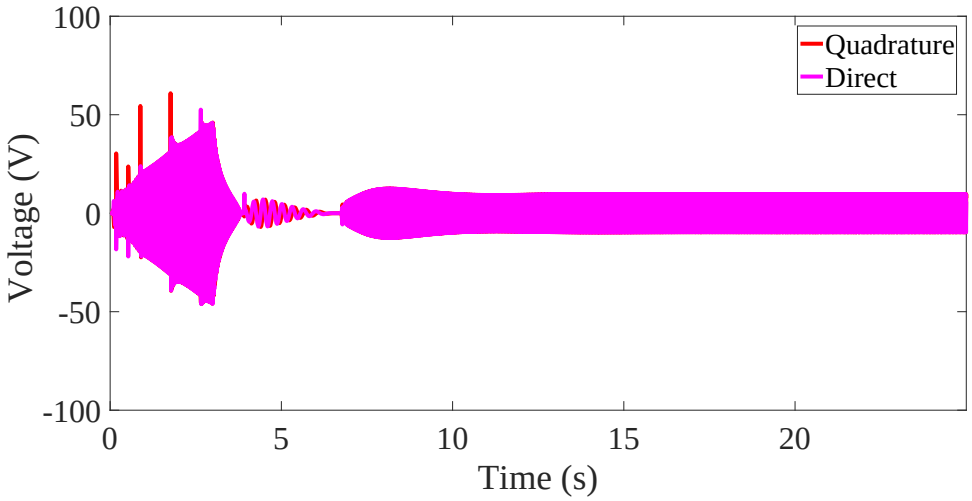


Figure D.4: Torque Test 1, Voltage.

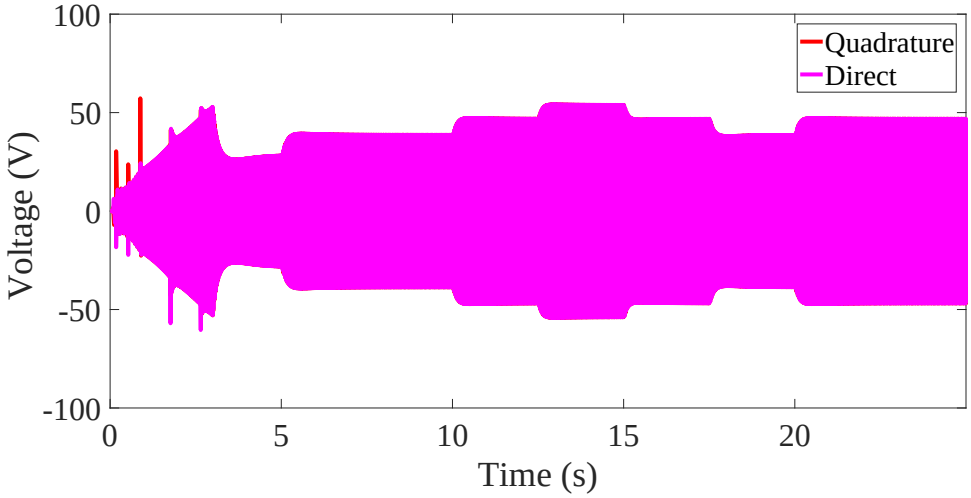


Figure D.5: Torque Test 2, Voltage.

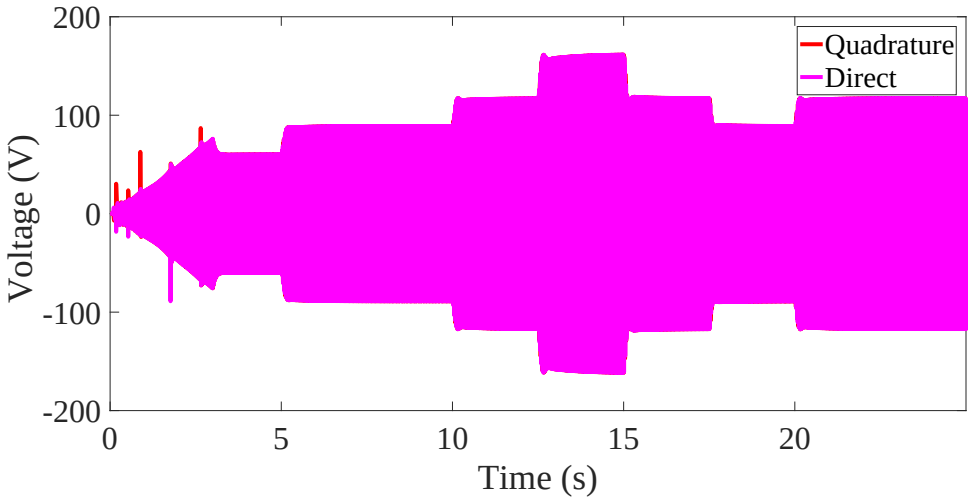


Figure D.6: Torque Test 3, Voltage.

Appendix E

Appendix E

IP 55 – IC 411 – Insulation class F, temperature rise class B – Brake IP 23 S

Output kW	Motor type	Product code	Speed r/min	Torque rated TN Nm	Torque braking TB Nm	TS/TN	K[1]	Efficiency 100%	Efficiency 75%	Power factor cos φ	Current IN A	IS/IN	c/h[2]	Moment of inertia J=1/4GD² kgm²	Weight foot mounted kg
1500 r/min = 4 poles				**400 V 50 Hz**							**Basic design**				
0.12	**M3VRF 63 A**	3GVR 062 401-••C	1400	0.82	7.5	2.6	9.37	63.7	58.4	0.59	0.46	3.1	7100	0.00029	5
0.18	**M3VRF 63 B**	3GVR 062 402-••C	1380	1.25	7.5	2.5	5.76	65.6	62.1	0.64	0.63	3.1	7100	0.00036	5.5
0.25	**M3VRF 71 A**	3GVR 072 401-••E	1410	1.71	10	2.7	5.88	70.4	69.1	0.71	0.74	4.3	6500	0.00081	7
0.37	**M3VRF 71 B**	3GVR 072 402-••E	1420	2.51	10	2.6	4	74.6	72.1	0.69	1.05	4.4	6500	0.00104	8
0.55	**M3VRF 80 A**	3GVR 082 401-••E	1390	3.75	24	2.6	6.31	75.3	73.1	0.76	1.4	4.6	5000	0.00128	11
0.75	**M3VRF 80 B**	3GVR 082 402-••E	1410	5.08	24	3.5	4.7	78.2	75.6	0.74	1.9	4.7	5000	0.00159	12
1.1	**M3ARF 90 S**	3GAR 092 401-••E	1410	7.5	35	2.2	4.66	77.5	76.4	0.81	2.59	5.0	3200	0.0034	19
1.5	**M3ARF 90 L**	3GAR 092 402-••E	1420	10	35	2.4	3.5	80.3	78.1	0.79	3.45	5.0	3200	0.0045	22

Figure E.1: Motor Parameters M3ARF 90 S.

Appendix F

Appendix F

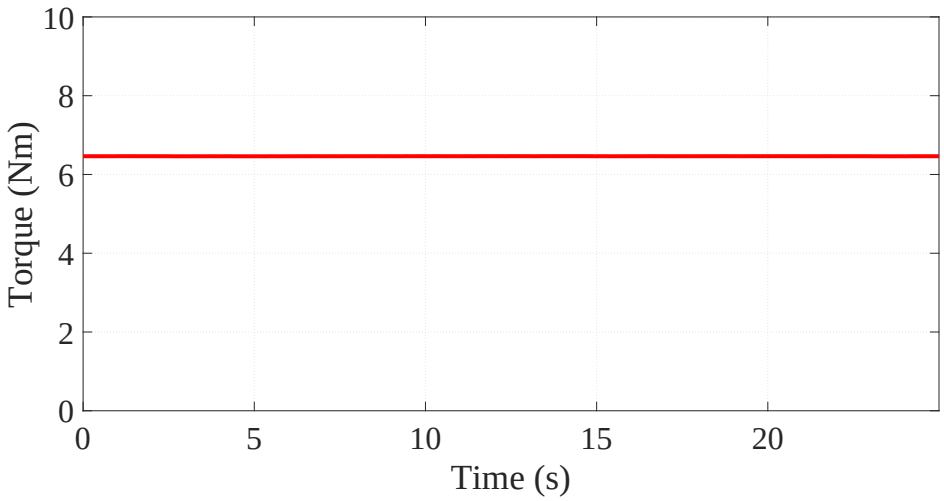


Figure F.1: Torque, Real Data.

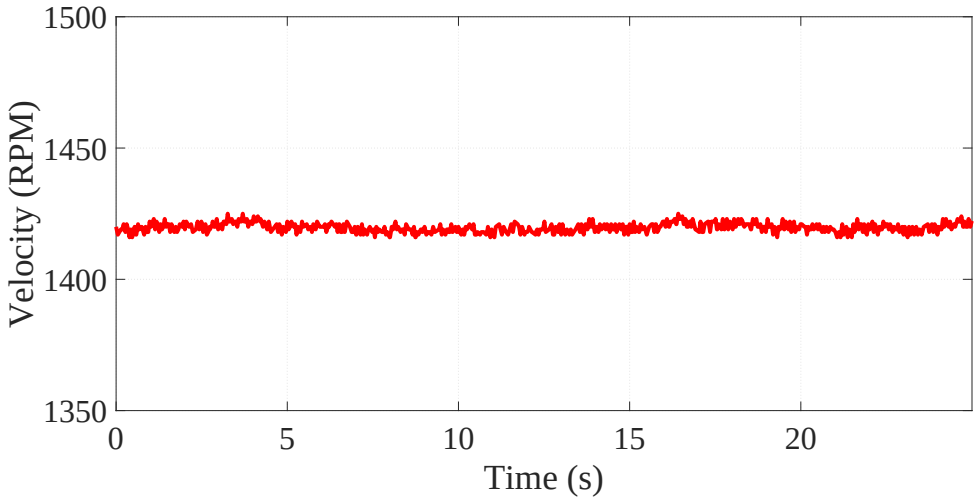


Figure F.2: Velocity, Real Data.

Appendix G

Appendix G

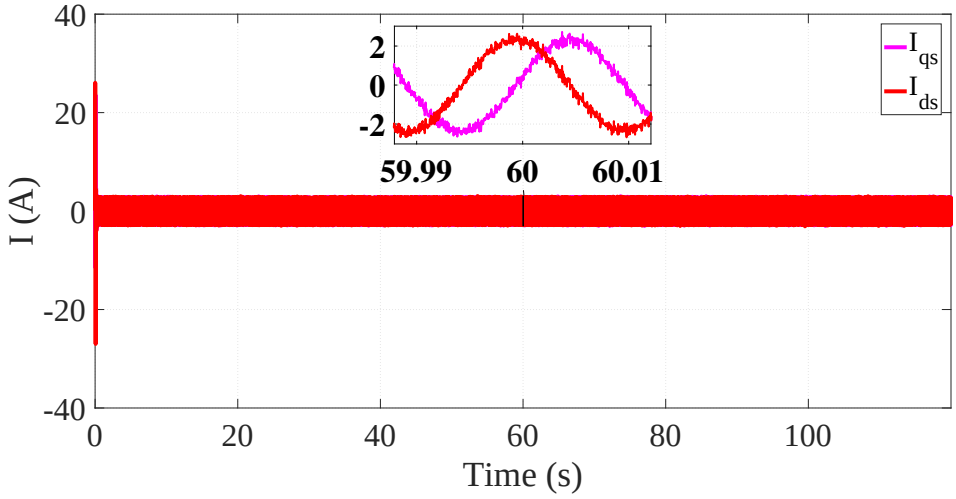


Figure G.1: Currents Test (1).

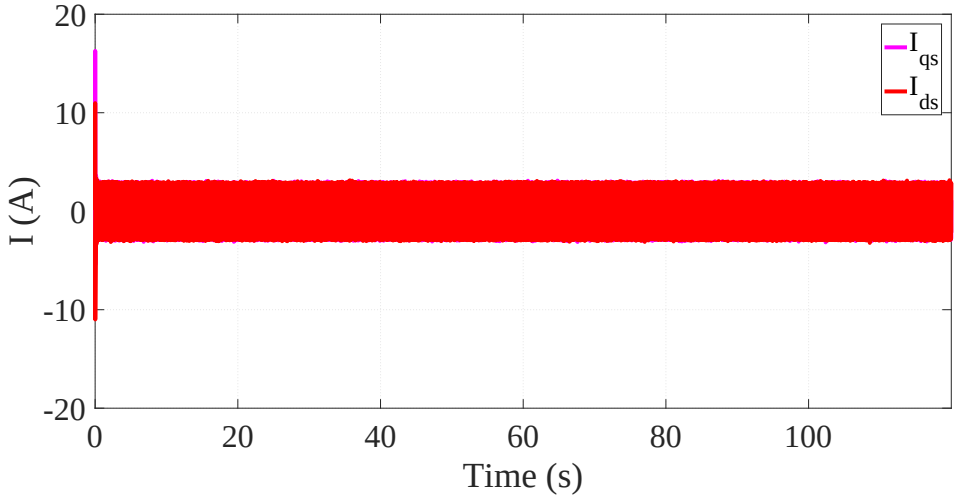


Figure G.2: Currents Test (2).

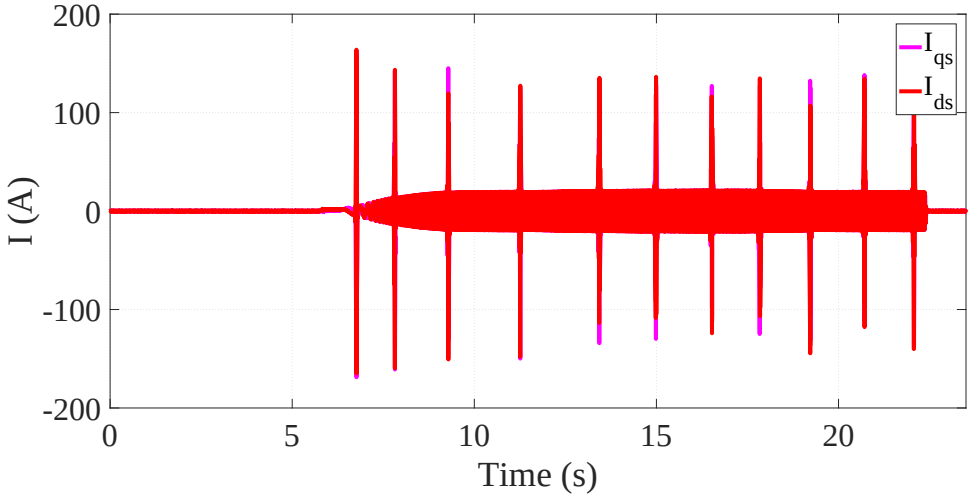


Figure G.3: Currents Test (3).

Appendix H

Appendix H

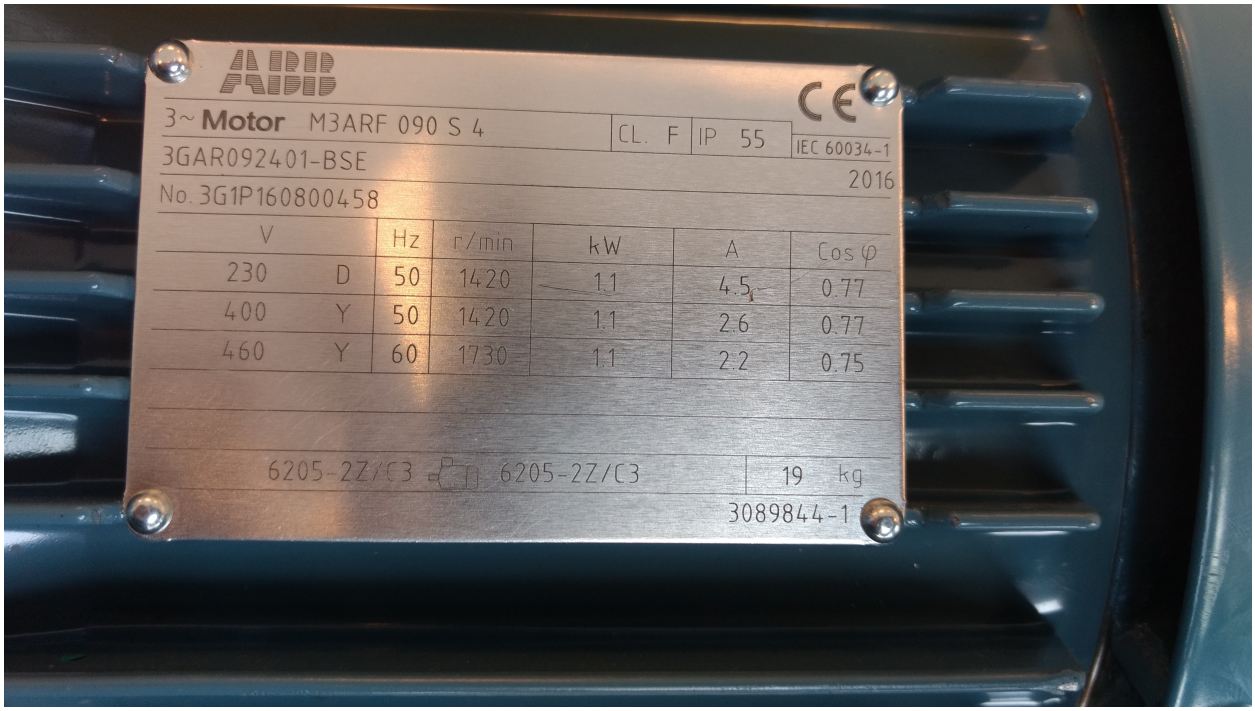


Figure H.1: Motor Name Plate.

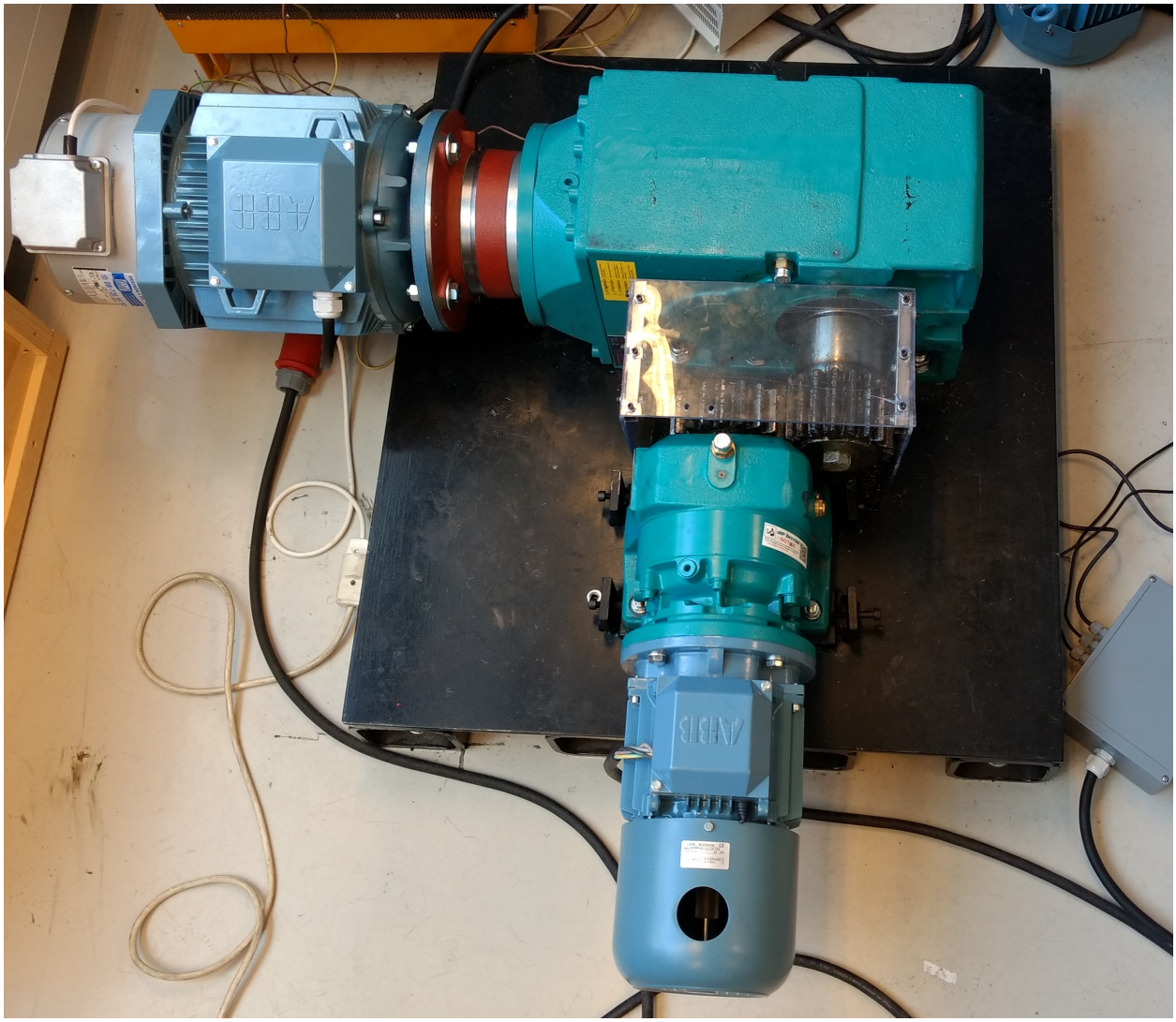


Figure H.2: Motor Set-up.