

Development and Integration of a Speed Observer for the Direct Torque Control of an Electric Propulsion System for an Electric Vehicle

João de Bruges Toledo Soares Galamba

Thesis to obtain the Master of Science Degree in

Electrical and Computer Engineering

Supervisor(s): Professor Paulo José da Costa Branco

IDMEC Researcher Francisco Ferreira da Silva

Examination Committee

Chairperson: Prof. Célia Maria Santos Cardoso de Jesus

Supervisor: Prof. Paulo José da Costa Branco

Member of the Committee: Prof. Alexandra Bento Moutinho

January 2021

Declaration

I declare that this document is an original work of my own authorship and that it fulfills all the requirements of the Code of Conduct and Good Practices of the Universidade de Lisboa.

Acknowledgments

A special thanks to my supervisors, Professor Paulo Branco and Master Francisco Silva, for their dedication and support, and for all the time spent pointing me in the right direction.

Thank you to my friends, who were always there for me. To you, my friend Nuno, for all those days spent in that little office room working on our theses and for the help soldering those tiny SMD components. To my great friend Denert for always being there for me and for all the great times we've had over the years.

Thank you to my family for giving me this great opportunity. To my parents for all the love and support I was given throughout my life. To everyone that believed in me and helped me along the way.

Thank you.

Resumo

Esta dissertação surge com a necessidade do projeto VIENA (Veículo Inteligente Elétrico de Navegação Autônoma) de implementar controlo de binário num motor de indução sem a instalação de um sensor de velocidade no eixo, resultando no desenvolvimento de um observador de velocidade. Este tem o propósito de ser implementado no veículo elétrico do projeto, um modelo Fiat Seicento Elettra.

Para aquele efeito, obteve-se o modelo em espaço de estados da máquina de indução, a partir do qual é desenvolvido um estimador de corrente. Finalmente é desenvolvido um controlador Pi que utiliza o resultado desta estimação em conjunto com o valor real da corrente para estimar a velocidade mecânica do motor. Duas situações foram consideradas nos testes do observador relativamente ao funcionamento do motor de indução: inicialmente considerou-se uma ligação direta do motor à rede elétrica de forma a realizar os ensaios iniciais do observador, seguido pela introdução de um inversor de tensão.

Após a análise do observador de velocidade, seguiu-se o projeto e desenvolvimento do sistema eletrónico para a sua implementação, o que requereu a aquisição dos valores de tensão e corrente do motor. Para este efeito foram delineados dois sistemas distintos. O primeiro contendo equipamento de medição integrado no circuito. O segundo fazendo um pedido de dados à interface do inversor do veículo tirando partido da funcionalidade da comunicação CAN.

O segundo sistema foi escolhido devido à sua facilidade de instalação, sendo testada a receção e o envio de mensagens CAN, utilizando *hardware* e *software* dedicados para o efeito, de modo a verificar a viabilidade da utilização de mensagens CAN para transmitir os dados do e para o observador.

Palavras-chave: Motor de indução; Controlo; Observador de velocidade; Tração elétrica; Comunicação CAN

Abstract

This Master thesis arises from the need for the VIENA (Intelligent Electric Vehicle for Autonomous Navigation) project to implement direct torque control on an induction motor without the installation of a speed sensor on the motor axis, resulting in the development of a speed observer. This is to be implemented in the project's electric vehicle, a Fiat Seicento Elettra.

This need led to the development of the space state model of the induction machine, which was used to develop a current estimator. Finally, to get the speed estimation a PI-controller was developed. This uses the estimated currents in conjunction with the real current values to estimate the motor's mechanical speed. This was followed by an array of tests on the observer's performance that included direct on-line startup and inverter integration.

Following this validation and in order to implement the observer, two distinct approaches were developed to acquire the motor's current and voltage data. The first contains integrated sensors to measure these quantities, while the second takes advantage of the CAN communication protocol to request the data directly from the inverter.

The second design was ultimately chosen due to its ease of installation, and tested using dedicated hardware and software, which allows for the tracking of incoming and outgoing CAN messages. This was done to verify the use of CAN messages to transmit the speed observer's input and output data.

Keywords: Induction motor; Control; Speed observer; Electric traction; CAN

Contents

- Declarationii
- Acknowledgmentsiii
- Resumoiv
- Abstractv
- List of Tablesviii
- List of Figuresix
- Glossaryxi
- Nomenclaturexii
- 1 Introduction 1
 - 1.1 Topic Overview 2
 - 1.2 Objectives 5
 - 1.3 Thesis Outline 5
- 2 Theoretical Background 6
 - 2.1 Induction Machine 6
 - 2.1.1 Mathematical Representation 8
 - 2.1.2 Scalar Command 10
 - 2.1.3 Dynamic operation 12
 - 2.2 $dq0$ Model 15
 - 2.2.1 Park Transformation 16
 - 2.2.2 Induction machine equations 16
- 3 Full Order State Observer 19
 - 3.1 Observer Development 19
 - 3.2 Simulation 26
 - 3.2.1 Pure Sinusoidal System 26
 - 3.2.2 Soft Starter Implementation 30
 - 3.2.3 Inverter Implementation 34
- 4 Hardware Implementation 38
 - 4.1 External Acquisition Design 38
 - 4.1.1 Voltage Circuit 39
 - 4.1.2 Current circuit 41
 - 4.2 CAN-HAT Design 44
 - 4.2.1 Hardware and Circuit Design 45
 - 4.2.2 Configuration 50

4.2.3	Testing	51
5	Results	54
5.1	Constant Load.....	54
5.2	Load Disturbance.....	57
6	Conclusions.....	60
6.1	Future work	61
	Bibliography.....	62
	Annex A - PCB Schematics	63
	External Acquisition Design.....	63
	CAN-HAT Design	68

List of Tables

TABLE 1.1 - MOTOR NAMEPLATE DATA 3

TABLE 3.1 - INDUCTION MACHINE ELECTRICAL SIMULATION PARAMETERS..... 26

TABLE 3.2 - INDUCTION MACHINE MECHANICAL SIMULATION PARAMETERS 27

TABLE 3.3 - OBSERVER SIMULATION PARAMETERS (DOL) 27

TABLE 3.4 - OBSERVER SIMULATION PARAMETERS (V/F)..... 31

TABLE 3.5 - OBSERVER SIMULATION PARAMETERS (INVERTER + V/F) 35

List of Figures

FIGURE 1.1 - DRAWING OF A FIAT SEICENTO ELETTRA [17] AND THE REAL VEHICLE IN THE ELECTRICAL MACHINES LABORATORY	2
FIGURE 1.2 - GENERIC ELECTRIC VEHICLE POWERTRAIN [16]	2
FIGURE 1.3 - SEVCON GEN4 SIZE 4 CONTROLLER [2]	3
FIGURE 1.4 - BLOCK DIAGRAM OF A USUAL VEHICLE TORQUE CONTROL SYSTEM	4
FIGURE 1.5 - BLOCK DIAGRAM OF A SENSORLESS VEHICLE TORQUE CONTROL SYSTEM	4
FIGURE 2.1 - SQUIRREL CAGE ROTOR OF AN INDUCTION MACHINE ([5], P.173, PLATE 5.2)	6
FIGURE 2.2 - TYPICAL INDUCTION MOTOR TORQUE - SPEED CURVE ([3], P.310, FIGURE 6.4)	7
FIGURE 2.3 - MODEL OF AN INDUCTION MACHINE [6].....	8
FIGURE 2.4 - PER-PHASE EQUIVALENT CIRCUIT OF AN INDUCTION MACHINE.....	8
FIGURE 2.5 - TORQUE-SPEED CURVES WITH CONSTANT V/F RATIO ([5], P.286, FIGURE 8.3)	10
FIGURE 2.6 - TORQUE-SPEED CURVES WITH VOLTAGE BOOSTING AT LOW SPEEDS ([5], P.287, FIGURE 8.4)	11
FIGURE 2.7 - CONSTANT TORQUE, CONSTANT POWER AND HIGH-SPEED REGIONS ([5], P.289, FIGURE 8.5).....	11
FIGURE 2.8 - PER-COMPONENT REPRESENTATION OF THE INDUCTION MACHINE INDUCTANCE MATRIX	13
FIGURE 2.9 - ELECTRICAL ANGLE RELATIONSHIP BETWEEN STATOR, ROTOR AND DQ-REFERENCE FRAME [6].....	15
FIGURE 3.1 – MOTOR AND SPEED OBSERVER SYSTEMS.....	25
FIGURE 3.2- IM POLES CHANGING WITH SPEED.....	25
FIGURE 3.3 - OBSERVER POLES CHANGING WITH SPEED	25
FIGURE 3.4 - OVERVIEW OF THE SIMULATED SYSTEM	26
FIGURE 3.5 - ADAPTIVE SPEED OBSERVER.....	27
FIGURE 3.6 - PERFORMANCE OF SPEED ESTIMATION (DOL).....	28
FIGURE 3.7 - PERFORMANCE OF TORQUE ESTIMATION (DOL).....	29
FIGURE 3.8 - PERFORMANCE OF DIRECT CURRENT ESTIMATION (DOL)	29
FIGURE 3.9 - PERFORMANCE OF QUADRATURE CURRENT ESTIMATION (DOL)	30
FIGURE 3.10 - SCALAR COMMAND IMPLEMENTATION (V/F)	30
FIGURE 3.11 - SINGLE-PHASE VOLTAGE WITH SCALAR COMMAND (V/F).....	31
FIGURE 3.12 - PERFORMANCE OF SPEED ESTIMATION (V/F)	32
FIGURE 3.13 - PERFORMANCE OF TORQUE ESTIMATION (V/F)	32
FIGURE 3.14 - PERFORMANCE OF DIRECT CURRENT ESTIMATION (V/F)	33
FIGURE 3.15 - PERFORMANCE OF QUADRATURE CURRENT ESTIMATION (V/F)	33
FIGURE 3.16 – BLOCK-DIAGRAM OF A THREE-PHASE PWM INVERTER WITH V/F	34
FIGURE 3.17 - SINGLE-PHASE VOLTAGE WITH INVERTER AND SCALAR COMMAND (V/F)	34
FIGURE 3.18 - OBSERVER POLES CHANGING WITH SPEED ($k=1.1$).....	35
FIGURE 3.19 - IM POLES CHANGING WITH SPEED ($k=1.1$)	35
FIGURE 3.20 - PERFORMANCE OF SPEED ESTIMATION (INV+V/F)	36
FIGURE 3.21 - PERFORMANCE OF TORQUE ESTIMATION (INV+V/F)	36

FIGURE 3.22 - PERFORMANCE OF DIRECT CURRENT ESTIMATION (INV+V/F).....	37
FIGURE 3.23 - PERFORMANCE OF QUADRATURE CURRENT ESTIMATION (INV+V/F).....	37
FIGURE 4.1 - 3D RENDER OF THE FINAL PCB DESIGN FOR THE EXTERNAL ACQUISITION PROTOTYPE	38
FIGURE 4.2 - VOLTAGE DIVIDER	39
FIGURE 4.3 - VOLTAGE ACQUISITION CIRCUIT FOR PHASE A.....	39
FIGURE 4.4 - INPUT LOW PASS FILTER.....	40
FIGURE 4.5 - EQUIVALENT CIRCUIT OF INPUT LOW PASS FILTER.....	40
FIGURE 4.6 - OUTPUT LOW PASS FILTER	41
FIGURE 4.7 - HALL-EFFECT MEASURING PRINCIPLE [15].....	41
FIGURE 4.8 - MAIN ELEMENTS OF A CURRENT TRANSDUCER [11]	42
FIGURE 4.9 - CURRENT ACQUISITION CIRCUIT FOR PHASE A.....	43
FIGURE 4.10 - COMMUNICATION FUNCTIONS OF CAN AND CANOPEN	44
FIGURE 4.11 - CANOPEN MESSAGE LAYOUT	44
FIGURE 4.12 - RASPBERRY PI3 (LEFT) AND CAN-HAT PROTOTYPE BOARD, BEFORE (RIGHT) AND AFTER (CENTER) ASSEMBLY.....	45
FIGURE 4.13 - BLOCK DIAGRAM OF THE CAN-HAT PCB DESIGN.....	45
FIGURE 4.14 - RASPBERRY PI GPIO PIN LAYOUT [12]	46
FIGURE 4.15 - DC/DC STEP-DOWN CONVERTER	47
FIGURE 4.16 - ELECTRICAL CIRCUIT OF THE MCP2515 CAN CONTROLLER	48
FIGURE 4.17 - ELECTRICAL CIRCUIT OF THE MCP2562 CAN TRANSCEIVER.....	49
FIGURE 4.18 - ELECTRICAL CIRCUIT OF THE DS3231MZ REAL-TIME CLOCK (RTC)	49
FIGURE 4.19 - ARP -A COMMAND OUTPUT EXAMPLE.....	50
FIGURE 4.20 - CAN-HAT PROTOTYPE CONNECTED TO A CAN ANALYZER	51
FIGURE 4.21 - CAN ANALYZER PINOUT	52
FIGURE 4.22 - TRANSMISSION AND BUS MONITORING WINDOWS OF THE ANALYZER SOFTWARE	52
FIGURE 4.23 - CANDUMP CAN0 OUTPUT EXAMPLE	53
FIGURE 4.24 - ANALYZER ROLLING TRACE WINDOW AFTER MESSAGE RECEPTION.....	53
FIGURE 5.1 - PERFORMANCE OF SPEED ESTIMATION IN THE TRANSITION TO STEADY STATE.....	54
FIGURE 5.2 - PERFORMANCE OF TORQUE ESTIMATION IN THE TRANSITION TO STEADY STATE	55
FIGURE 5.3 - PERFORMANCE OF DIRECT CURRENT ESTIMATION IN THE TRANSITION TO STEADY STATE	56
FIGURE 5.4 - PERFORMANCE OF QUADRATURE CURRENT ESTIMATION IN THE TRANSITION TO STEADY STATE	56
FIGURE 5.5 - PERFORMANCE OF SPEED ESTIMATION WITH A LOAD DISRUPTION.....	57
FIGURE 5.6 - PERFORMANCE OF TORQUE ESTIMATION WITH A LOAD DISRUPTION.....	58
FIGURE 5.7 - PERFORMANCE OF DIRECT CURRENT ESTIMATION WITH A LOAD DISRUPTION.....	59
FIGURE 5.8 - PERFORMANCE OF QUADRATURE CURRENT ESTIMATION WITH A LOAD DISRUPTION.....	59

Glossary

<i>CAN</i>	Controller Area Network
<i>CLK</i>	Serial Clock
<i>CS</i>	Chip Select
<i>I2C</i>	Inter-Integrated Circuit Communication
<i>COB – ID</i>	Communication Object Identifier
<i>DOL</i>	Direct On-Line
<i>GPIO</i>	General Purpose Input/Output
<i>ICE</i>	Internal Combustion Engine
<i>MAC</i>	Media Access Control
<i>MISO</i>	Master In Slave Out
<i>MOSI</i>	Master Out Slave In
<i>OPAMP</i>	Operational Amplifier
<i>PCB</i>	Printed Circuit Board
<i>PI</i>	Proportional Integral
<i>PWM</i>	Pulse Width Modulation
<i>RMS</i>	Root Mean Square
<i>RTC</i>	Real-Time Clock
<i>RTR</i>	Remote Transmission Request
<i>SCIM</i>	Squirrel Cage Induction Machine
<i>SPI</i>	Serial Peripheral Interface
<i>VIENA</i>	Veículo Inteligente Elétrico de Navegação Autônoma (Intelligent Electric Vehicle for Autonomous Navigation)

Nomenclature

Greek symbols

ω_m	Mechanical angular velocity
ω_S	Stator electrical angular frequency
ω_R	Rotor electrical angular frequency
ϕ	Flux
θ_m	Mechanical angle between stator and rotor
θ_{me}	Electrical angle between stator and rotor
θ_R	Electrical angle between rotor and dq -reference frame
θ_S	Electrical angle between stator and dq -reference frame

Latin symbols

N_S	Synchronous speed
N_R	Rotor speed
f	Frequency
f_e	Electrical frequency
f_{cut}	Cutoff frequency
f_{samp}	Sampling frequency
pp	Number of pole pairs
s	Slip
R	Electrical resistance
X	Electrical reactance
X_m	Magnetizing reactance
$L_{S,R}$	Self-inductance
L_M	Mutual Inductance
$L_{1,2}$	Flux leakage
P_{elect}	Electrical power
P_{field}	Magnetic power
P_{loss}	Power losses
P_{mech}	Mechanical power
T_{mech}	Mechanical torque

T_{el}	Electromagnetic torque
T_{load}	Load torque
I	Electrical current
U	Electrical Voltage
t	Time
T_P	Park transformation
J_{tot}	Total Inertia
W_{elect}	Electrical energy
W_{mech}	Mechanical energy
W_{field}	Magnetic energy
$\hat{}$	Estimated value
e	Estimation error
K_P	Proportional gain
K_I	Integral gain

Subscripts

a, b, c	Natural reference frame
R	Rotor component
S	Stator component
$d, q, 0$	Direct-quadrature reference frame

Superscripts

k	Incremental value
-----	-------------------

1 Introduction

For the last few years, concerns over climate change and the health risks of excessive pollution have led to the electric vehicles' rapid development, specifically electric city cars. Unlike their counterparts from the early 2000s, whose range on electric propulsion was measured in dozens of kilometers, newer models have more than ten times the range while also rivaling their internal combustion engine (ICE) counterparts in terms of performance.

This evolution is due, in part, to advances in battery technology, but an often-overlooked aspect is the development in the control of electric motors. This accounts for lower battery consumption and other features such as regenerative braking and independent wheel torque control.

In this Master thesis, a sensorless speed observer is developed to allow for an induction motor's direct torque control, which will be part of an electric vehicle's powertrain. The sensorless design will ensure fewer points of failure and will reduce the overall weight and complexity of the system when compared to the traditional implementation of a physical speed sensor. Another factor for the reduced complexity is that the control module developed in this thesis was designed to interface with the vehicle's CAN bus. This bus is the standard for communication between the vehicle's systems, such as the engine, brakes, and traction control, and is present in all modern vehicles, electrical and ICE alike.

1.1 Topic Overview

The vehicle used for the project is a 1998 Fiat Seicento Elettra, seen in Figure 1.1. An electric car built from the chassis of the regular gasoline-powered Fiat Seicento, the Elettra is driven by a three-phase squirrel cage induction motor.

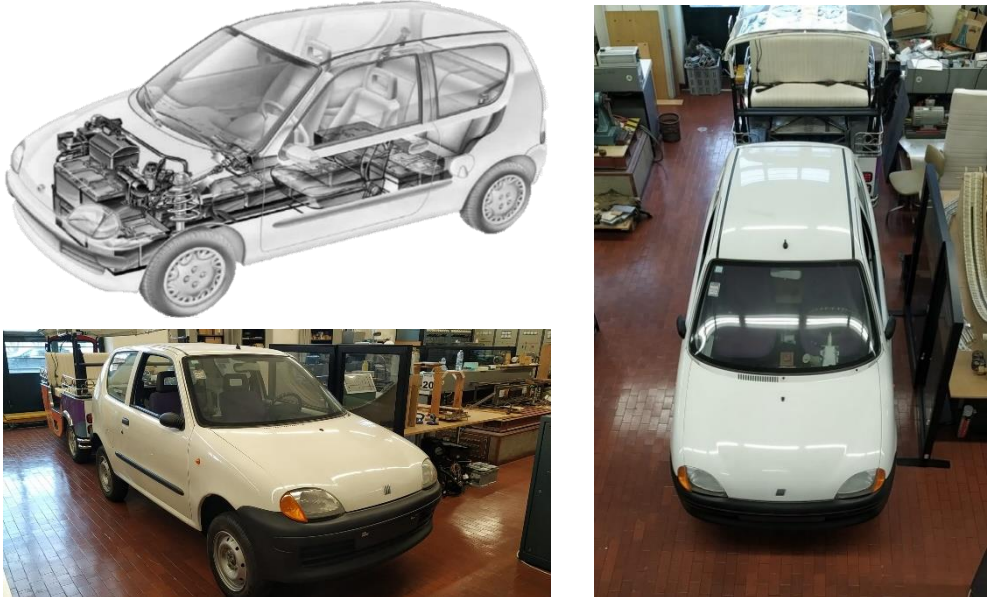


Figure 1.1 - Drawing of a Fiat Seicento Elettra [17] and the Real Vehicle in the Electrical Machines Laboratory

The drivetrain of an electric vehicle can be reduced to a few basic elements, as shown in Figure 1.2. This consists of a motor to move the vehicle, a battery pack to store electric energy, an inverter to feed and control the motor, and, finally, a mechanical transmission to convert the motor’s rotation speed to the wheels. Although all of these components will be briefly touched on, this thesis will mainly focus on the electric motor, its characteristics, and the development and integration of the speed observer.

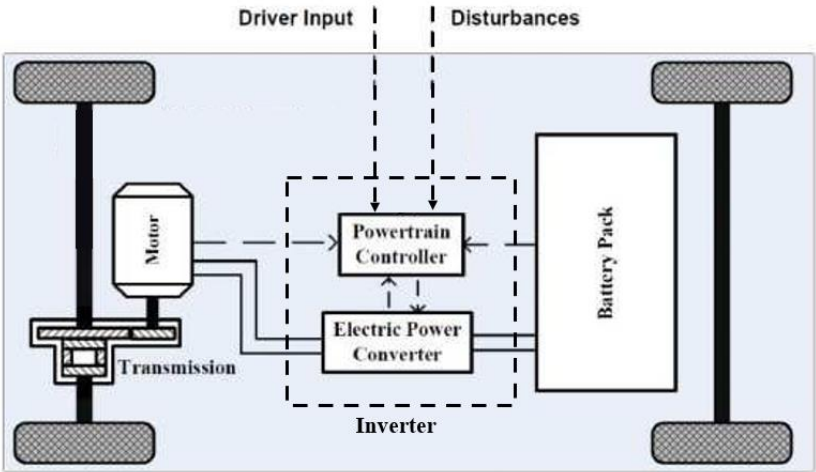


Figure 1.2 - Generic Electric Vehicle Powertrain [16]

The motor, a three-phase induction machine, has its nameplate data displayed in Table 1.1. The data looks quite different from traditional induction machines, as it has several nominal operating points at different speeds. This

is explained by the fact that the motor is part of a vehicle’s powertrain. A car is expected to run at different speeds, from stop-and-go traffic to highway speeds, while maintaining a reasonable energy consumption and driving range. Having multiple nominal operating points, i.e., speeds at which the motor is running at maximum efficiency, allows the vehicle to function without disproportional energy needs.

Table 1.1 - Motor Nameplate Data

Speed [rpm]	Voltage [V]	Frequency [Hz]	Current [A]	Power [kW]	Torque [Nm]
2200	75	76	157	15	65
6500	121	220	90	15	22
9000	121	305	88	15	16

For energy storage, the car originally used eighteen 12 V lead-acid batteries amounting to 13 kWh [1]. This type of battery is no longer used to power modern electric vehicles, as it was replaced with technologies allowing for greater energy density, such as Lithium-Ion.

As an induction machine drives the car, AC power is required to feed the motor. Thus, an inverter is needed to convert the battery’s DC power into AC. The original inverter was recently replaced by a new SEVCON gen4 controller, as seen in Figure 1.3.

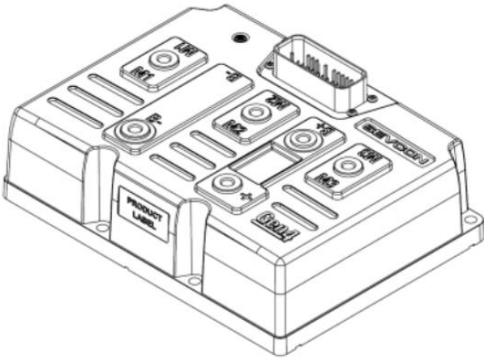


Figure 1.3 - SEVCON Gen4 Size 4 controller [2]

It is important to point out that the inverter is also responsible for controlling the motor’s mechanical outputs, i.e., its rotational speed and mechanical torque, making sure it is able to respond adequately to the driver’s inputs, such as a press of the pedal to accelerate for an overtake. This is done in the DC/AC conversion by adequately regulating the motor’s input voltage and frequency.

The new SEVCON inverter allows its control interface to be accessed using the integrated CAN bus [2], giving the user access to real-time electrical data without the need for additional sensors and measuring devices that could be difficult to implement. This data, in the form of voltage and current values, will then be used as the input for the sensorless, adaptive speed observer, which is the main purpose of this Master thesis.

The usual way to develop a motor's speed control, seen in Figure 1.4, uses a dedicated speed sensor. This requires its fixing on one or several of the vehicle's wheels to determine their rotational speed. This value is then converted, through the available gear ratios of the motor and transmission, into the motor's actual speed, which is then used for the control.

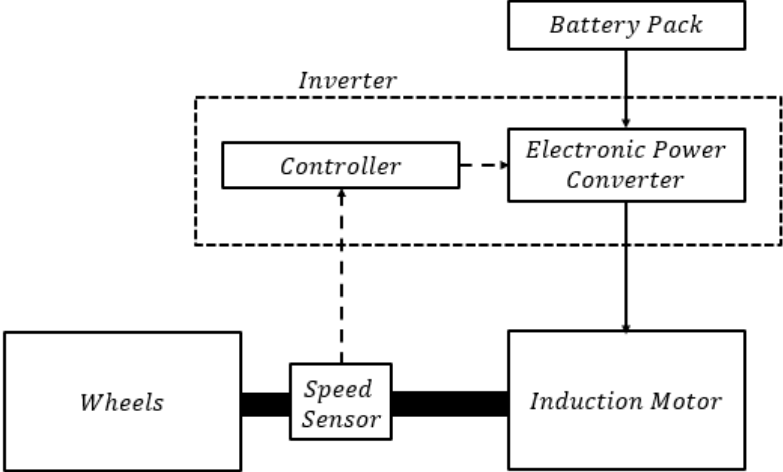


Figure 1.4 - Block Diagram of a Usual Vehicle Torque Control System

Figure 1.5 shows that sensorless speed control abandons this additional equipment, taking advantage of the AC electric drives' extensively known characteristics. The electrical quantities outputted by the inverter, i.e., the amplitude and frequency of voltage and current, are used in a model to estimate, with a high degree of accuracy, the mechanical speed that the motor will output when fed by these quantities. This model is referred to as the *motor speed observer*.

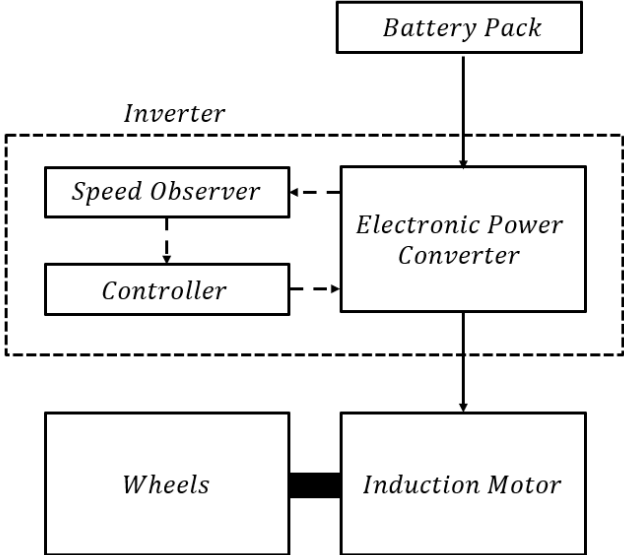


Figure 1.5 - Block diagram of a Sensorless Vehicle Torque Control System

1.2 Objectives

This Master's thesis aims to design and implement an adaptive speed observer on the Fiat Elettra owned by the VIENA project and located in the Electrical Machines Laboratory of the Energy scientific area. This is developed using the induction machine's dynamic $dq0$ model [3], requiring the study and understanding of its mechanical and electrical characteristics. To implement such a speed observer, the instantaneous values of the motor's electrical quantities are needed. Therefore, prototype printed circuit boards (PCBs) will be developed to acquire these electrical quantities through two approaches such as direct measurement or communication with the inverter.

1.3 Thesis Outline

Chapter 2 contains the necessary background for the development of the speed observer. It begins with an overview of the induction machine and its characteristics, followed by its equivalent circuit and overall mathematical representation. We then dive into the motor's dynamic operation characteristics, required due to its application in vehicle traction, and transition into the $dq0$ model and Park Transform, rewriting the machine equations in this new reference frame. Finally, for this chapter, the speed observer model is developed and tested, using a simulated motor. These tests simulate various scenarios, from a direct connection to the power grid to a full system, now including an inverter, which allows command of the voltage amplitude and frequency.

Chapter 3 focuses on developing the prototypes to be implemented on the physical system and the hardware choices that were made for this. The chapter is divided into two sections. The first describes the more conventional approach of a circuit with integrated sensors to measure the motor's voltage and current values and feed this data into the observer. The second approach communicates with the inverter's interface and uses the electrical data directly from its source. The chapter ends with testing of the message reception and transmission, using the fully built prototype.

Chapter 4 presents some more detailed results of the simulations, using the absolute error values. It goes through the speed and torque estimation results, together with the direct and quadrature current values used in their computation.

Finally, chapter 5 summarizes the work done during the master thesis while suggesting different approaches to optimize future observer performance and proposes some hardware changes that could be required after the physical implementation.

2 Theoretical Background

An induction machine, or asynchronous machine, is an electrical machine where alternating voltages are directly supplied to the stator and, indirectly, through induction, to the rotor. Induction motors, in particular, traditionally operate under constant speed, where their steady-state characteristics are more relevant [4]. In our case, though the motor is part of a vehicle's powertrain, making variable-speed operation is a necessity. Hence, its dynamic performance must be considered. This is a considerably more complex subject in induction machines than in other electric motor types. Therefore, a two-axis d-q model will be used to study its dynamic characteristics and design the speed controller.

2.1 Induction Machine

An induction machine's stator is like a synchronous machine, meaning it consists of three distributed windings electrically displaced by 120° . When powered by a balanced three-phase power source, where each phase is displaced by 120° , a rotating magnetic field at a synchronous speed of N_S rpm will be produced in the airgap. This speed is determined by the number of pole pairs (pp) in the machine, as well as its stator electrical frequency (f_e) [4], as

$$N_S = \frac{60 \cdot f_e}{pp}. \quad (2.1)$$

As for the rotor, there are two possible configurations. The first with a wound rotor has distributed windings similar to those of the stator and should have the same number of pole pairs. The terminals of these rotors are usually connected to brushes to allow a connection for command purposes. This type of rotor was popular before the advent of variable frequency inverters for this reason.



Figure 2.1 - Squirrel cage rotor of an Induction Machine ([5], p.173, Plate 5.2)

The second configuration is the squirrel-cage rotor, as seen in Figure 2.1. This consists of conducting bars, short-circuited by design at the ends, giving the rotor a cage-like appearance. As stated by Fitzgerald [3], the construction's simplicity and ruggedness are the main advantages that make this design the most commonly-used electric motor type.

Assuming the rotor of an induction motor is rotating at a speed of N_R rpm, the difference between synchronous and rotor speed is referred to as slip (s) [3], given by

$$s = \frac{N_S - N_R}{N_S} \tag{2.2}$$

When starting, the rotor is stationary ($N_R = 0$), thus the slip is unitary ($s = 1$). As the motor starts to turn and the speed increases, we can see that the motor torque increases with the slip up to a maximum value and then decreases to zero ($N_R = N_S$), as exemplified in Figure 2.2.

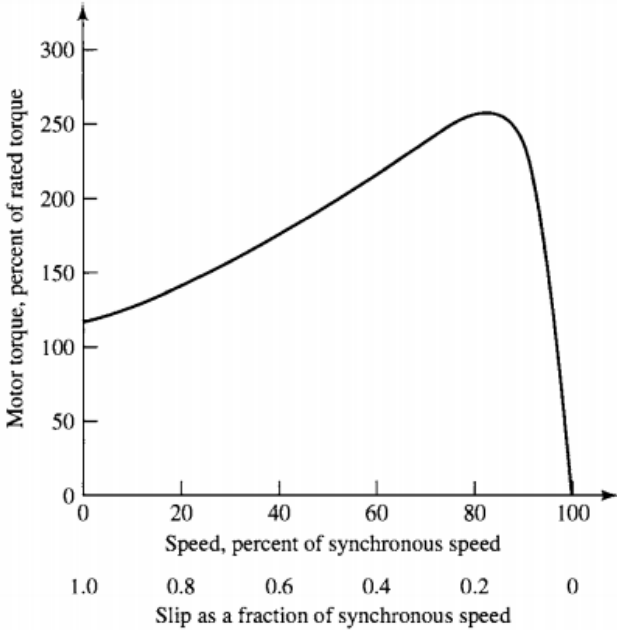


Figure 2.2 - Typical Induction motor Torque - Speed curve ([3], p.310, Figure 6.4)

This is an important characteristic of the induction machine, as the rotor speed (N_R) cannot be the same as the synchronous speed (N_S), i.e., the motor cannot operate with a slip equal to zero. This would mean that the rotor conductors would be stationary concerning the stator field, meaning there would be no induced currents, and thus, no torque would be produced by the motor.

2.1.1 Mathematical Representation

For the mathematical representation of the induction machine, some considerations must be made. Firstly, it is assumed that the motor is a 3-phase balanced and sinusoidal system, with a stator made of three identical windings electrically displaced by 120° . Secondly, although the machine contains a squirrel cage rotor, three windings can be represented with the same arrangement as the stator, connected in star and short-linked as seen in Figure 2.3.

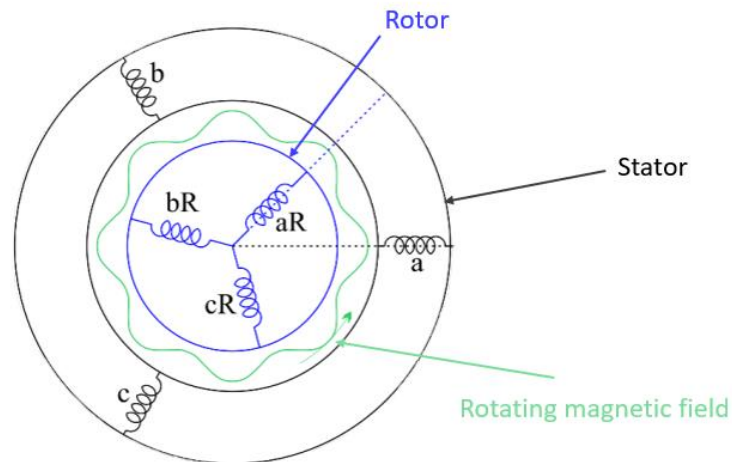


Figure 2.3 - Model of an induction machine [6]

Under these conditions and assuming steady-state operation, a per-phase equivalent circuit, shown in Figure 2.4, is created.

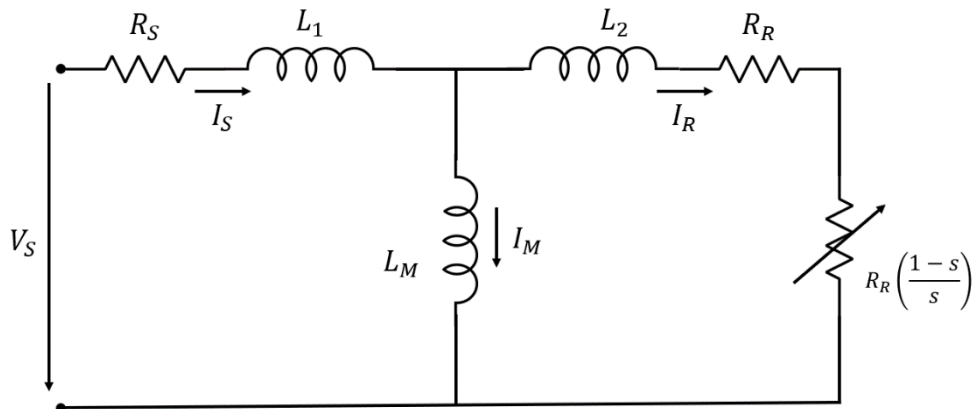


Figure 2.4 - Per-phase equivalent circuit of an Induction machine

Electrical components with a similar behavior represent the physical phenomena occurring in the induction machine. Therefore R_S and R_R represent the copper losses in the stator and rotor, respectively, while L_S and L_R are the self-inductance coefficients of stator and rotor. These are defined as

$$L_S = L_1 + L_M \quad (2.3)$$

$$L_R = L_2 + L_M, \quad (2.4)$$

with L_M as the mutual inductance coefficient and L_1 and L_2 as the leakage inductances.

As for the last component, $R_R \left(\frac{1-s}{s}\right)$, it corresponds to the electrical power that is converted into mechanical power by the motor, which varies with the slip. This means the available mechanical power is then given by

$$P_{mech} = 3R_R \left(\frac{1-s}{s}\right) I_R^2, \quad (2.5)$$

as the single-phase value is multiplied by the number of phases.

The mechanical power can then be used to define the torque of the induction machine, which is defined as

$$T_{mech} = \frac{P_{mech}}{\omega_m} \quad (2.6)$$

with

$$\omega_m = (1-s)\omega_s \quad (2.7)$$

and

$$\omega_s = \frac{2\pi f_e}{pp}, \quad (2.8)$$

where ω_m is the mechanical angular velocity of the rotor and ω_s the synchronous angular velocity.

2.1.2 Scalar Command

As previously mentioned, this specific motor is used for vehicle traction, meaning variable-speed operation is necessary. By nature, induction motors can only function efficiently near their synchronous speed (slip near zero), meaning the speed control must provide a smooth variation of the synchronous speed.

A common speed control method is to supply the motor with a variable frequency supply, using an inverter. This is usually done while also controlling the input voltage and maintaining the ratio between frequency and voltage (V/f). This type of control can be explained with the stator voltage equation

$$V_s = R_s I_s + \frac{d\phi}{dt} \quad (2.9)$$

The voltage drop in the stator can be neglected, meaning

$$V_s \approx \frac{d\phi}{dt}, \quad (2.10)$$

and thus,

$$\phi \propto \frac{V}{f}. \quad (2.11)$$

This results in the family of torque-speed curves shown in Figure 2.5.

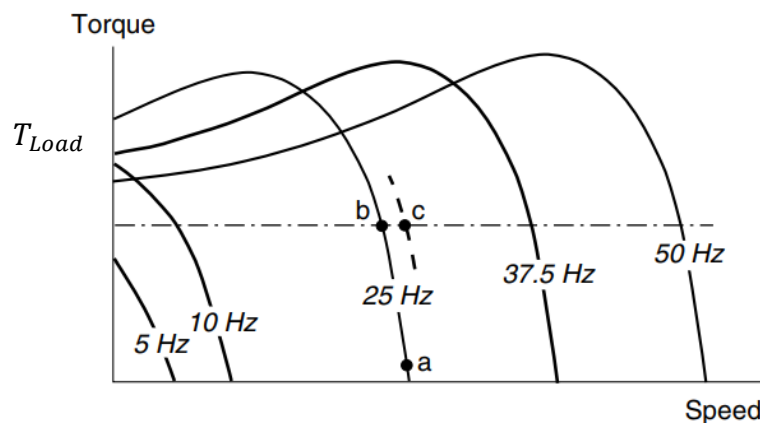


Figure 2.5 - Torque-Speed Curves with Constant V/f ratio ([5], p.286, Figure 8.3)

This does not mean however, that the V/f ratio must remain constant. An increase in voltage while maintaining the frequency, can be used to compensate for the lack of torque at very low speeds, as shown in Figure 2.6. This is commonly referred to as *low-speed voltage boosting*.

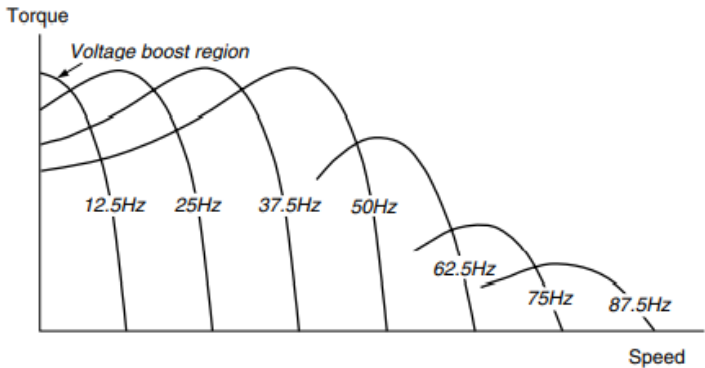


Figure 2.6 - Torque-Speed Curves with Voltage Boosting at Low Speeds ([5], p.287, Figure 8.4)

Such an approach is required for low frequencies, as the voltage drop in the stator cannot be neglected. The opposite is also true. If the motor needs to operate at a very high speed, i.e., higher than the nominal value, the frequency can be increased while maintaining the voltage value. This cannot be done with constant V/f as the motor would overheat.

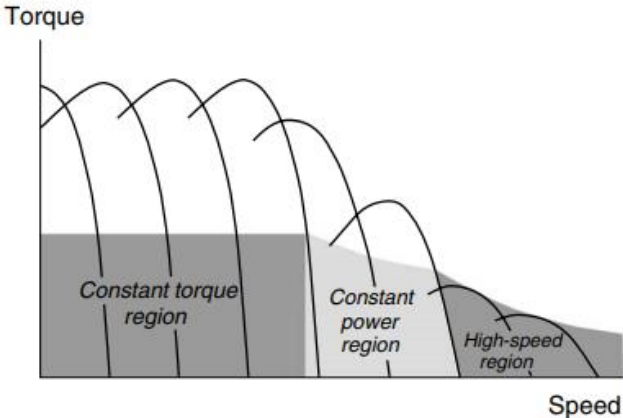


Figure 2.7 - Constant Torque, constant Power and High-Speed regions ([5], p.289, Figure 8.5)

This means the torque-speed curves can be divided into three regions, highlighted in Figure 2.7. The first, the constant torque region, goes from zero up to nominal speed. Here, as the name suggests, the maximum motor torque can be delivered regardless of speed (if low-speed voltage boosting is used).

Above the nominal speed the currents are limited, and thus, the flux is reduced inversely with frequency, making the motor operate with higher slip values. The maximum allowed torque is also reduced with speed, which usually goes to twice its nominal value [5]. This area is called the constant power region. Operation above this value is still possible in the high-speed region, even if constant power is no longer available as the torque is limited to the pull-out value, i.e., the maximum value.

2.1.3 Dynamic operation

When approaching the induction machine's dynamic operation, the first components one should focus on are the machine's self and mutual inductances as they are a key part of the motor flux, described as

$$\phi = LI \quad (2.12)$$

Assuming a three-phase induction machine with windings on both stator and rotor, the motor inductances are given by

$$[L(\theta_{me})] = \begin{bmatrix} \mathcal{L}_{aa} & \mathcal{L}_{ab} & \mathcal{L}_{ac} & \mathcal{L}_{aaR} & \mathcal{L}_{abR} & \mathcal{L}_{acR} \\ \mathcal{L}_{ba} & \mathcal{L}_{bb} & \mathcal{L}_{bc} & \mathcal{L}_{baR} & \mathcal{L}_{bbR} & \mathcal{L}_{bcR} \\ \mathcal{L}_{ca} & \mathcal{L}_{cb} & \mathcal{L}_{cc} & \mathcal{L}_{caR} & \mathcal{L}_{cbR} & \mathcal{L}_{ccR} \\ \mathcal{L}_{aRa} & \mathcal{L}_{aRb} & \mathcal{L}_{aRc} & \mathcal{L}_{aRaR} & \mathcal{L}_{aRbR} & \mathcal{L}_{aRcR} \\ \mathcal{L}_{bRa} & \mathcal{L}_{bRb} & \mathcal{L}_{bRc} & \mathcal{L}_{bRaR} & \mathcal{L}_{bRbR} & \mathcal{L}_{bRcR} \\ \mathcal{L}_{cRa} & \mathcal{L}_{cRb} & \mathcal{L}_{cRc} & \mathcal{L}_{cRaR} & \mathcal{L}_{cRbR} & \mathcal{L}_{cRcR} \end{bmatrix}, \quad (2.13)$$

with (a, b, c) referring to the stator quantities and (aR, bR, cR) to the rotor quantities.

Note that the notation (θ_{me}) implies that the machine inductances are written as a function of the electrical angle between stator and rotor (between stator phase a and rotor phase aR), meaning they are position-dependent.

By further analyzing each inductance matrix position, we can identify the interactions between phases and those between stator and rotor. To this effect, stator self-inductances are described as

$$\mathcal{L}_{aa} = \mathcal{L}_{bb} = \mathcal{L}_{cc} = L_{MS} + L_{al} = L_S, \quad (2.14)$$

where L_{MS} is the air-gap component of the stator self-inductance and L_{al} is the leakage component.

As for the rotor self-inductances, they are described as

$$\mathcal{L}_{aRaR} = \mathcal{L}_{bRbR} = \mathcal{L}_{cRcR} = L_{MR} + L_{arI} = L_R, \quad (2.15)$$

where L_{MR} is the air-gap component of the rotor self-inductance and L_{arI} is the leakage component.

These stator and rotor self-inductances (L_S, L_R) are the ones portrayed in the equivalent circuit of Figure 2.4, as they represent the single-phase, self-inductance component that is relevant for that analysis.

Now for the interactions between each phase of the stator and rotor, the stator to stator mutual inductances are

$$\mathcal{L}_{ab} = \mathcal{L}_{ba} = \mathcal{L}_{ac} = \mathcal{L}_{ca} = \mathcal{L}_{bc} = \mathcal{L}_{cb} = -\frac{1}{2}L_{MS} \quad (2.16)$$

and rotor to rotor mutual inductances

$$\mathcal{L}_{aRbR} = \mathcal{L}_{bRaR} = \mathcal{L}_{aRcR} = \mathcal{L}_{cRaR} = \mathcal{L}_{bRcR} = \mathcal{L}_{cRbR} = -\frac{1}{2}L_{MR} \quad (2.17)$$

Finally, the stator to rotor mutual inductances are

$$\mathcal{L}_{aaR} = \mathcal{L}_{aRa} = \mathcal{L}_{bbR} = \mathcal{L}_{bRb} = \mathcal{L}_{ccR} = \mathcal{L}_{cRc} = L_M \cos(\theta_{me}) \quad (2.18)$$

$$\mathcal{L}_{baR} = \mathcal{L}_{aRb} = \mathcal{L}_{cbR} = \mathcal{L}_{bRc} = \mathcal{L}_{acR} = \mathcal{L}_{cRa} = L_M \cos\left(\theta_{me} - \frac{2\pi}{3}\right) \quad (2.19)$$

$$\mathcal{L}_{caR} = \mathcal{L}_{aRc} = \mathcal{L}_{abR} = \mathcal{L}_{bRa} = \mathcal{L}_{bcR} = \mathcal{L}_{cRb} = L_M \cos\left(\theta_{me} + \frac{2\pi}{3}\right). \quad (2.20)$$

This interaction between stator and rotor inductances results in the positional dependence of the system. Taking the results from Equations (2.14) to (2.20), the inductance matrix can be described as

		Stator			Rotor		
Stator	[L_S	$-\frac{1}{2}L_{MS}$	$-\frac{1}{2}L_{MS}$	$L_M \cos(\theta_{me})$	$L_M \cos\left(\theta_{me} + \frac{2\pi}{3}\right)$	$L_M \cos\left(\theta_{me} - \frac{2\pi}{3}\right)$
	$-\frac{1}{2}L_{MS}$	L_S	$-\frac{1}{2}L_{MS}$	$L_M \cos\left(\theta_{me} - \frac{2\pi}{3}\right)$	$L_M \cos(\theta_{me})$	$L_M \cos\left(\theta_{me} + \frac{2\pi}{3}\right)$	
	$-\frac{1}{2}L_{MS}$	$-\frac{1}{2}L_{MS}$	L_S	$L_M \cos\left(\theta_{me} + \frac{2\pi}{3}\right)$	$L_M \cos\left(\theta_{me} - \frac{2\pi}{3}\right)$	$L_M \cos(\theta_{me})$	
Rotor	[$L_M \cos(\theta_{me})$	$L_M \cos\left(\theta_{me} + \frac{2\pi}{3}\right)$	$L_M \cos\left(\theta_{me} - \frac{2\pi}{3}\right)$	L_R	$-\frac{1}{2}L_{MR}$	$-\frac{1}{2}L_{MR}$
	$L_M \cos\left(\theta_{me} - \frac{2\pi}{3}\right)$	$L_M \cos(\theta_{me})$	$L_M \cos\left(\theta_{me} + \frac{2\pi}{3}\right)$	$-\frac{1}{2}L_{MR}$	L_R	$-\frac{1}{2}L_{MR}$	
	$L_M \cos\left(\theta_{me} + \frac{2\pi}{3}\right)$	$L_M \cos\left(\theta_{me} - \frac{2\pi}{3}\right)$	$L_M \cos(\theta_{me})$	$-\frac{1}{2}L_{MR}$	$-\frac{1}{2}L_{MR}$	L_R	
					$]$		

Figure 2.8 - Per-component representation of the induction machine inductance matrix

As viewed in Figure 2.8 the inductance matrix can be divided into stator and rotor components, top left and bottom right, respectively, as well as the position-dependent interaction between both.

Now that the inductance matrix has been defined the three-phase stator and rotor voltage equations of the induction machine can be described as:

$$[U_{abc}] = [R_{abc}][i_{abc}] + \frac{d}{dt}[\phi_{abc}], \quad (2.21)$$

with

$$[\phi_{abc}] = [L(\theta_{me})][i_{abc}]$$

These are then split into per-phase components for rotor and stator

$$U_a = R_a i_a + \frac{d\phi_a}{dt} \quad (2.22)$$

$$U_b = R_b i_b + \frac{d\phi_b}{dt} \quad (2.23)$$

$$U_c = R_c i_c + \frac{d\phi_c}{dt} \quad (2.24)$$

and

$$0 = U_{aR} = R_{aR} i_{aR} + \frac{d\phi_{aR}}{dt} \quad (2.25)$$

$$0 = U_{bR} = R_{bR} i_{bR} + \frac{d\phi_{bR}}{dt} \quad (2.26)$$

$$0 = U_{cR} = R_{cR} i_{cR} + \frac{d\phi_{cR}}{dt} \quad (2.27)$$

The rotor voltages (U_{aR} , U_{bR} and U_{cR}) are equal to zero, as the windings on an induction machine are short-circuited by design.

As seen in Figure 2.8, voltage's flux component depends on the rotor's relative position due to the inductance matrix. This contributes to the further complexity of the computations and must be addressed. Therefore, the next section will address this positional-dependence along with the system's time-dependence using the Clarke and Park transformation (dq -transformation) [3] [6].

2.2 dq0 Model

To simplify the computations, the Clarke and Park transformation will be used. This turns a three-phase (a, b and c), time-varying system into a two-variable, direct (d) and quadrature (q), non-time dependent system. [3] [6]

This is done by transforming the stator and rotor quantities into equivalent quantities that rotate at the synchronous angular velocity, meaning they are stationary to each other. In an induction machine, converting both stator and rotor quantities is necessary because it does not rotate at synchronous speed. If, for example, the rotor was chosen as the reference, the direct and quadrature values of current and flux would still be time-varying, and the transformation would have no advantages over something like a simple Clarke transform ($\alpha\beta$ -reference frame).

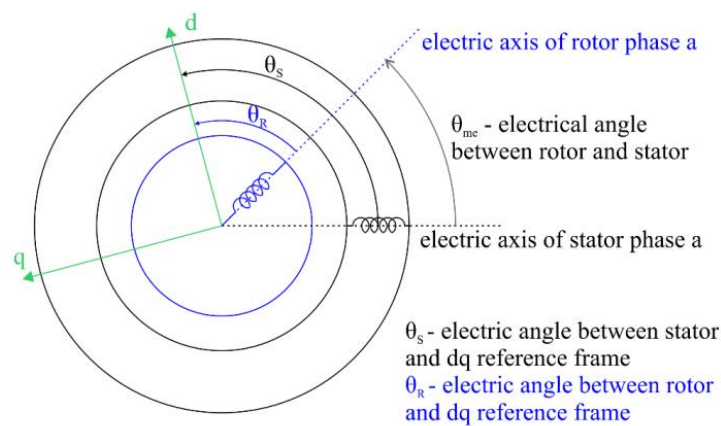


Figure 2.9 - Electrical angle relationship between stator, rotor and dq-reference frame [6]

The relationship between stator, rotor, and dq-reference frame, which can be seen in Figure 2.9, is described as

$$\theta_s = \theta_r + \theta_{me}, \quad (2.28)$$

where θ_s and θ_r refer to the electrical angle between stator, rotor and the dq-reference frame. θ_{me} refers to the electrical angle between the rotor and stator. The relationship between the mechanical angle θ_m and electrical angle θ_{me} can be described as

$$\theta_{me} = pp \cdot \theta_m, \quad (2.29)$$

with pp referring to the number of pole pairs of the induction machine.

2.2.1 Park Transformation

The Park transformation is used to get the machine's input voltage into the dq -reference frame, through

$$[U_{dq0}] = \frac{2}{3} \cdot [T_P] \cdot [U_{abc}], \quad (2.30)$$

where the Park transformation matrix is described as:

$$[T_P] = \begin{bmatrix} \cos(\theta_S) & \cos(\theta_S - \frac{2\pi}{3}) & \cos(\theta_S + \frac{2\pi}{3}) \\ -\sin(\theta_S) & -\sin(\theta_S - \frac{2\pi}{3}) & -\sin(\theta_S + \frac{2\pi}{3}) \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix} \quad (2.31)$$

2.2.2 Induction machine equations

After the transformation into the dq -reference frame, the electrical equations of the induction machine can be described as following.

For the stator voltages,

$$U_d = R_S I_d + \frac{d\phi_d}{dt} - \omega_S \phi_q \quad (2.32)$$

$$U_q = R_S I_q + \frac{d\phi_q}{dt} + \omega_S \phi_d \quad (2.33)$$

As was explained in Section 2.1.3, as this is a squirrel cage induction machine, the rotor voltages are equal to zero because of the short-circuiting, resulting in

$$0 = U_{dR} = R_R I_{dR} + \frac{d\phi_{dR}}{dt} - \omega_R \phi_{qR} \quad (2.34)$$

$$0 = U_{qR} = R_R I_{qR} + \frac{d\phi_{qR}}{dt} + \omega_R \phi_{dR}. \quad (2.35)$$

For the stator flux

$$\phi_d = L_S I_d + L_M I_{dR} \quad (2.36)$$

$$\phi_q = L_S I_q + L_M I_{qR} \quad (2.37)$$

and rotor flux

$$\phi_{dR} = L_R I_{dR} + L_M I_d \quad (2.38)$$

$$\phi_{qR} = L_R I_{qR} + L_M I_q \quad (2.39)$$

For ease of computation, the electrical equations can be written in matricial form as

$$\begin{bmatrix} U_d \\ U_q \\ U_{dR} \\ U_{qR} \end{bmatrix} = \begin{bmatrix} R_S & 0 & 0 & 0 \\ 0 & R_S & 0 & 0 \\ 0 & 0 & R_R & 0 \\ 0 & 0 & 0 & R_R \end{bmatrix} \begin{bmatrix} I_d \\ I_q \\ I_{dR} \\ I_{qR} \end{bmatrix} + \frac{d}{dt} \begin{bmatrix} \phi_d \\ \phi_q \\ \phi_{dR} \\ \phi_{qR} \end{bmatrix} + \begin{bmatrix} 0 & -\omega_s & 0 & 0 \\ \omega_s & 0 & 0 & 0 \\ 0 & 0 & 0 & -\omega_R \\ 0 & 0 & \omega_R & 0 \end{bmatrix} \begin{bmatrix} \phi_d \\ \phi_q \\ \phi_{dR} \\ \phi_{qR} \end{bmatrix} \quad (2.40)$$

and

$$\begin{bmatrix} \phi_d \\ \phi_q \\ \phi_{dR} \\ \phi_{qR} \end{bmatrix} = \begin{bmatrix} L_S & 0 & L_M & 0 \\ 0 & L_S & 0 & L_M \\ L_M & 0 & L_R & 0 \\ 0 & L_M & 0 & L_R \end{bmatrix} \begin{bmatrix} I_d \\ I_q \\ I_{dR} \\ I_{qR} \end{bmatrix}. \quad (2.41)$$

Now, equations (2.40) and (2.41) are arranged to build the state-space model of the induction machine, with the flux, ϕ , as state variable, resulting in

$$\frac{d}{dt} [\phi] = [A][\phi] + [U], \quad (2.42)$$

with

$$[A] = -[R][L]^{-1} - [\omega]. \quad (2.43)$$

Finally, for the mechanical equations of the induction machine, we have the torque equilibrium, described as

$$J_{tot} \frac{d\omega_m}{dt} = T_{el} - T_{load}, \quad (2.44)$$

where the electromagnetic torque, T_{el} , can be deduced from some energy considerations. To do this, we start with the following equation

$$\frac{dW_{elect}}{dt} = \frac{dW_{mech}}{dt} + \frac{dW_{field}}{dt}. \quad (2.45)$$

From here, the energies can be written as powers whose expressions are known, hence

$$P_{mech} = P_{elect} - P_{loss} - P_{field}. \quad (2.46)$$

Using equation (2.6) for the mechanical power, as well as the corresponding equations for electrical power, power losses, and magnetic power, we arrive at

$$\frac{2}{3} \cdot T_{el} \cdot \omega_r = [U_{dRqR}][I_{dRqR}] - R_R[I_{dRqR}^2] - \frac{d[\phi_{dRqR}]}{dt}[I_{dRqR}]. \quad (2.47)$$

The $\frac{2}{3}$ stems from the power balance in the transformation from a three-axis system into a two-axis system. Now expressions (2.34) and (2.35) are used for the rotor voltage, which results in

$$T_{el} \cdot \omega_r = -\frac{3}{2} \cdot \omega_r \cdot [\phi_{qRdR}] \cdot [I_{dRqR}], \quad (2.48)$$

which is developed into

$$T_{el} = \frac{3}{2} (\phi_{dR} I_{qR} - \phi_{qR} I_{dR}). \quad (2.49)$$

Finally, adapting the equation for any given number of poles we arrive at the desired equation, as

$$T_{el} = \frac{3}{2} \cdot pp \cdot (\phi_{dR} I_{qR} - \phi_{qR} I_{dR}). \quad (2.50)$$

Note that using the relations between the direct and quadrature components of the current and flux, shown in equations (2.36) to (2.39), it is possible to write the torque equation in several different ways, using either flux, current values or a combination of both.

3 Full Order State Observer

An estimator is a dynamic system whose state variables are estimates from another system, which here corresponds to an induction motor. These can be divided into two groups, open-loop, or closed-loop estimators. In this case, a closed-loop estimator, also commonly referred to as an observer, will be used. The main difference between them is that observers contain a correction term that involves the estimation error and can thus adjust their response. [7]

3.1 Observer Development

The observer will be used to estimate the motor speed in real-time from the electrical quantities. This is done by first developing a state-estimator that can estimate the stator and rotor currents of the induction machine. These estimated currents are then used, together with the real current values, in a PI-controller that was designed to output the estimated mechanical speed of the motor, thus creating an adaptive speed observer.

To develop a speed observer, the state-space model of the induction motor is required. This comes in the shape of

$$\frac{dx}{dt} = Ax + Bu \quad (3.1)$$

and

$$I_s = Cx \quad (3.2)$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix},$$

with the flux as the state variable (x), as shown in equation (2.42).

A state variable change is then required to a more easily measured quantity, i.e. the motor current, taking equation (2.42) and changing from the flux to the stator and rotor currents.

This results in

$$\frac{d}{dt}I = AI + BU, \quad (3.3)$$

with

$$A = L^{-1}(R + \omega L) \quad (3.4)$$

$$R = \begin{bmatrix} R_S & 0 & 0 & 0 \\ 0 & R_S & 0 & 0 \\ 0 & 0 & R_R & 0 \\ 0 & 0 & 0 & R_R \end{bmatrix} \quad L = \begin{bmatrix} L_S & 0 & L_M & 0 \\ 0 & L_S & 0 & L_M \\ L_M & 0 & L_R & 0 \\ 0 & L_M & 0 & L_R \end{bmatrix}$$

$$\omega = \begin{bmatrix} 0 & -\omega_S & 0 & 0 \\ \omega_S & 0 & 0 & 0 \\ 0 & 0 & 0 & -\omega_R \\ 0 & 0 & \omega_R & 0 \end{bmatrix}$$

and

$$B = L^{-1}. \quad (3.5)$$

Hence, equation (3.4) can be written as

$$\frac{d}{dt} \begin{bmatrix} I_s^d \\ I_s^q \\ I_r^d \\ I_r^q \end{bmatrix} = \quad (3.6)$$

$$\begin{bmatrix} (L_S L_R)\sigma & (L_M^2 \omega_R - L_S L_R \omega_S)\sigma & -(R_R L_M)\sigma & -(L_M L_R \Delta\omega)\sigma \\ -(L_M^2 \omega_R - L_S L_R \omega_S)\sigma & (L_S L_R)\sigma & -(L_M L_R \Delta\omega)\sigma & -(R_R L_M)\sigma \\ -(R_S L_M)\sigma & -(L_M L_S \Delta\omega)\sigma & -(R_R L_S)\sigma & (L_M^2 \omega_S - L_S L_R \omega_R)\sigma \\ (L_M L_S \Delta\omega)\sigma & -(R_S L_M)\sigma & (L_M^2 \omega_S - L_S L_R \omega_R)\sigma & -(R_R L_S)\sigma \end{bmatrix} \begin{bmatrix} I_s^d \\ I_s^q \\ I_r^d \\ I_r^q \end{bmatrix} \\ + \begin{bmatrix} -(L_R U_d)\sigma \\ -(L_R U_q)\sigma \\ (L_M U_d)\sigma \\ (L_M U_q)\sigma \end{bmatrix}$$

with

$$\sigma = \frac{1}{L_M^2 - L_R L_S}$$

and

$$\Delta\omega = \omega_R - \omega_S.$$

Now, by using the previous state-space equation (3.1) and implementing a term that consists of the difference between real and estimated states, i.e., the estimation error, one can implement an observer that estimates stator and rotor currents, in the form of

$$\frac{d\hat{x}}{dt} = \hat{A}\hat{x} + Bu + G(I_s - \hat{I}_s), \quad (3.7)$$

with

$$\hat{A} = \begin{bmatrix} (L_S L_R)\sigma & (L_M^2 \hat{\omega}_R - L_S L_R \omega_S)\sigma & -(R_R L_M)\sigma & -(L_M L_R \Delta \hat{\omega})\sigma \\ -(L_M^2 \hat{\omega}_R - L_S L_R \omega_S)\sigma & (L_S L_R)\sigma & -(L_M L_R \Delta \hat{\omega})\sigma & -(R_R L_M)\sigma \\ -(R_S L_M)\sigma & -(L_M L_S \Delta \hat{\omega})\sigma & -(R_R L_S)\sigma & (L_M^2 \omega_S - L_S L_R \hat{\omega}_R)\sigma \\ (L_M L_S \Delta \hat{\omega})\sigma & -(R_S L_M)\sigma & (L_M^2 \omega_S - L_S L_R \hat{\omega}_R)\sigma & -(R_R L_S)\sigma \end{bmatrix}$$

$$\Delta \hat{\omega} = \hat{\omega}_R - \omega_S$$

and

$$\hat{I}_S = C \hat{x}$$

$$\hat{x} = \begin{bmatrix} \hat{I}_S^d \\ \hat{I}_S^q \\ \hat{I}_r^d \\ \hat{I}_r^q \end{bmatrix}$$

Here, the symbol ^ represents an estimated value, while G is an observer gain matrix that is selected so that the system is stable. Its computation will be addressed further ahead.

Now that the current estimation is completed, it is necessary to ensure its observability. Observability is described as the possibility of reconstructing a system's full trajectory using data obtained from its input and output measurements [8]. This is done by checking if the rank of the observability matrix O is equal to the rank of the state matrix A , where

$$O = \begin{bmatrix} CA^0 \\ CA^1 \\ CA^2 \\ \vdots \\ CA^{k-1} \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

and

$$k = \text{rank}(A) = 4.$$

Thus, the system is observable if

$$\text{rank}(O) = 4.$$

Now that the stator and rotor currents can be estimated, we move to the development of the PI-controller that will use these quantities to output the induction machine's estimated mechanical speed ($\hat{\omega}_m$).

For this purpose, we will take the induction machine and observer state-space equations (3.1) and (3.7), as well as the expression for the dynamic estimation error (3.8) together with Lyapunov's stability theorem [7].

$$e = x - \hat{x} \quad (3.8)$$

Note that the estimation error does not include the measured rotor currents, as they are not accessible, and thus their real value (I_R) is not known to the system and cannot be used as a reference.

By deriving the error expression, we get

$$\frac{de}{dt} = \frac{dx}{dt} - \frac{d\hat{x}}{dt} \quad (3.9)$$

Replacing these values using the results from (3.1) and (3.7), then results in

$$\begin{aligned} \frac{de}{dt} &= Ax + Bu - \hat{A}\hat{x} - Bu - G(I_s - \hat{I}_s) \quad (3.10) \\ &= Ax - \hat{A}\hat{x} - GCx + GC\hat{x} \\ &= Ax - A\hat{x} + A\hat{x} - \hat{A}\hat{x} - GC(x - \hat{x}) \\ &= (A - GC)(x - \hat{x}) + (A - \hat{A})\hat{x} \\ &= (A - GC)e - \Delta A\hat{x}, \end{aligned}$$

with

$$\Delta A = \hat{A} - A$$

$$\Delta A = \begin{bmatrix} 0 & \sigma L m^2 \Delta \hat{\omega}_R & 0 & \sigma L m L r \Delta \hat{\omega}_R \\ -\sigma L m^2 \Delta \hat{\omega}_R & 0 & -\sigma L m L r \Delta \hat{\omega}_R & 0 \\ 0 & -\sigma L m L s \Delta \hat{\omega}_R & 0 & -\sigma L r L s \Delta \hat{\omega}_R \\ \sigma L m L s \Delta \hat{\omega}_R & 0 & \sigma L r L s \Delta \hat{\omega}_R & 0 \end{bmatrix} \quad (3.11)$$

$$\Delta \hat{\omega}_R = \hat{\omega}_R - \omega_R$$

To achieve the required speed estimation, our objective is to arrive at the equation of a PI-controller in the shape of

$$\hat{\omega}_m = K_p(\dots) + K_i \int (\dots) dt,$$

where $\hat{\omega}_m$ is the estimated mechanical rotor speed of the induction machine. To develop the PI-controller, a candidate Lyapunov function in the form of

$$V = e^T e + \frac{(\hat{\omega}_m - \omega_m)}{c} ; c > 0 \quad (3.12)$$

is chosen [7]. This function is then derived, resulting in

$$\frac{dV}{dt} = e \left[\frac{d(e^T)}{dt} \right] + e^T \left[\frac{de}{dt} \right] + 2 \frac{d\hat{\omega}_m}{dt} \frac{(\hat{\omega}_m - \omega_m)}{c}. \quad (3.13)$$

Combining (3.13) with (3.10), we arrive at

$$\begin{aligned} \frac{dV}{dt} = & \quad (3.14) \\ e^T [(A - GC)^T + (A - GC)] e - (\hat{x}^T \Delta A^T e + e^T \Delta A \hat{x}) + 2 \frac{d\hat{\omega}_m}{dt} \frac{(\hat{\omega}_m - \omega_m)}{c}. \end{aligned}$$

The Observer Gain Matrix (G) is chosen so that the equation's first term is negative definite, meaning the system is stable [7] [9]. Therefore, the sum of the last two terms must be zero to ensure that equation (3.14) is negative definite, resulting in

$$(\hat{x}^T \Delta A^T e + e^T \Delta A \hat{x}) = 2 \frac{d\hat{\omega}_m}{dt} \frac{(\hat{\omega}_m - \omega_m)}{c}.$$

Now by using (3.11) and (3.8), we get

$$2\sigma(\hat{\omega}_m - \omega_m)[Lm(\hat{i}_s^q e_s^d - \hat{i}_s^d e_s^q) + Lr(\hat{i}_r^q e_s^d - \hat{i}_r^d e_s^q)] = 2 \frac{d\hat{\omega}_m}{dt} \frac{(\hat{\omega}_m - \omega_m)}{c}$$

and finally,

$$\frac{d\hat{\omega}_m}{dt} = K_I [Lm(\hat{i}_s^q e_s^d - \hat{i}_s^d e_s^q) + Lr(\hat{i}_r^q e_s^d - \hat{i}_r^d e_s^q)].$$

Now, by adding a proportional gain, resulting in a decrease in the observer's response time, we arrive at the desired equation for the induction machine's estimated mechanical speed.

$$\begin{aligned}\hat{\omega}_m = & K_p [Lm(\hat{i}_s^q e_s^d - \hat{i}_s^d e_s^q) + Lr(\hat{i}_r^q e_s^d - \hat{i}_r^d e_s^q)] \\ & + \int K_I [Lm(\hat{i}_s^q e_s^d - \hat{i}_s^d e_s^q) + Lr(\hat{i}_r^q e_s^d - \hat{i}_r^d e_s^q)]\end{aligned}\quad (3.15)$$

K_p and K_I are positive gains that should be adjusted according to the needs of the system. These are usually kept high to increase response speed.

As for the values of the gain matrix (G), they were chosen so that the observer poles are proportional those of the induction machine, according to [7].

$$G = - \begin{bmatrix} g_1 \cdot I_2 \cdot g_2 \cdot J \\ g_3 \cdot I_2 \cdot g_4 \cdot J \end{bmatrix}, \quad (3.16)$$

with

$$I_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad J = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}.$$

This results in a two-by-four Matrix with the following values:

$$g_1 = -(k-1) \left(\frac{1}{T_S} + \frac{1}{T_R} \right)$$

$$g_2 = (k-1) \hat{\omega}_m$$

$$g_3 = (k^2 - 1) \left\{ - \left[\frac{1}{T_S} + \frac{(1-\delta)}{T_R} \right] \cdot \frac{L_S L_M}{L_R} + \frac{L_M}{T_R} \right\} + L_S \frac{L_M}{L_R} \cdot (k-1) \cdot \left(\frac{1}{T_S} + \frac{1}{T_R} \right)$$

$$g_4 = -(k-1) \hat{\omega}_m \cdot \frac{L_S L_M}{L_R}$$

Here, k , is the proportionality constant ($k \geq 1$). This parameter makes the observer faster than the induction machine but is generally kept small to reduce noise sensitivity.

$$\text{Also, } T_S = \frac{L_S}{R_S}, T_R = \frac{L_R}{R_R} \text{ and } \delta = 1 - \frac{L_M^2}{L_S L_R}.$$

The fully developed speed observer system can be seen in Figure 3.1, connected to the state space model of an induction motor.

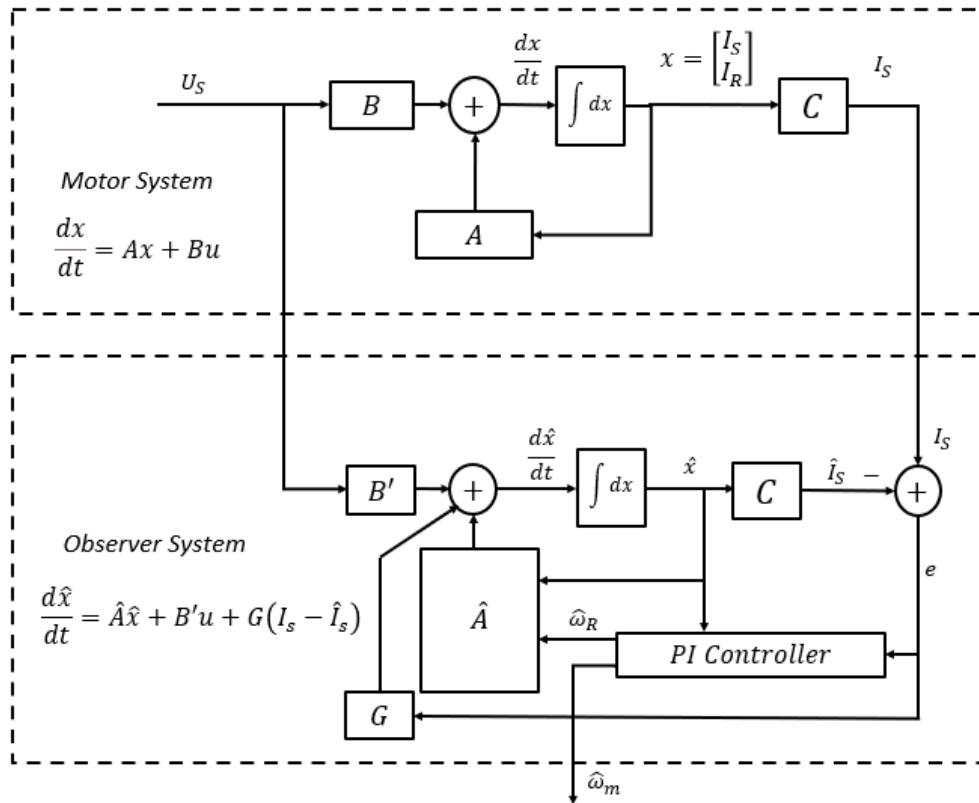


Figure 3.1 – Motor and Speed Observer Systems

The Induction machine poles, shown in Figure 3.2, are then determined by computing the eigenvalues of the matrix $[A]$, while the observer poles in Figure 3.3 can be computed by solving the eigenvalues of $[A - GC]$ [7]. Both these matrixes are dependent on the value of motor speed ω_m , and thus the poles will be displayed from an initial state (mechanical speed = 0) up to steady state (mechanical speed = 2200 rpm).

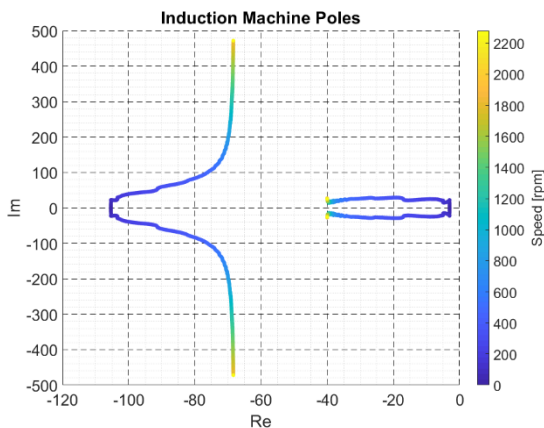


Figure 3.2- IM Poles changing with Speed

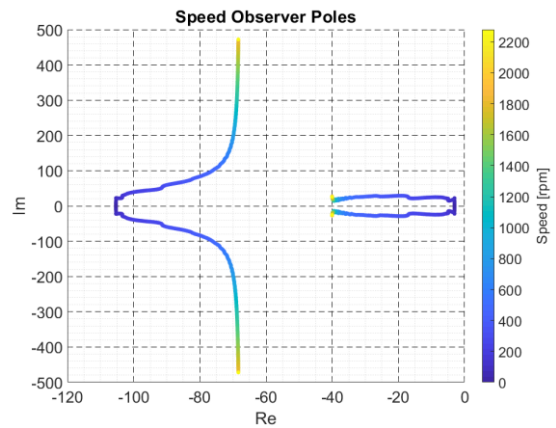


Figure 3.3 - Observer Poles changing with Speed

3.2 Simulation

The simulations will start with the simplest case, i.e., without an inverter, and progress to a scenario with an implemented inverter and V/f (scalar command). The simulated induction motor is based on the motor from the VIENA electric car, which has had its nameplate data detailed in Table 1.1.

Figure 3.4 gives an overview of the simulated system. Firstly, the state-space equation (2.42), as well as the mechanical equations of the induction machine (2.44) and (2.50), are used to simulate an induction motor. The current from this simulated motor is then compared to the one coming from the estimator, and the error between them is again fed to the observer for correction. The speed observer's performance is then evaluated by comparing the electrical and mechanical quantities from its outputs with those of the simulated induction machine.

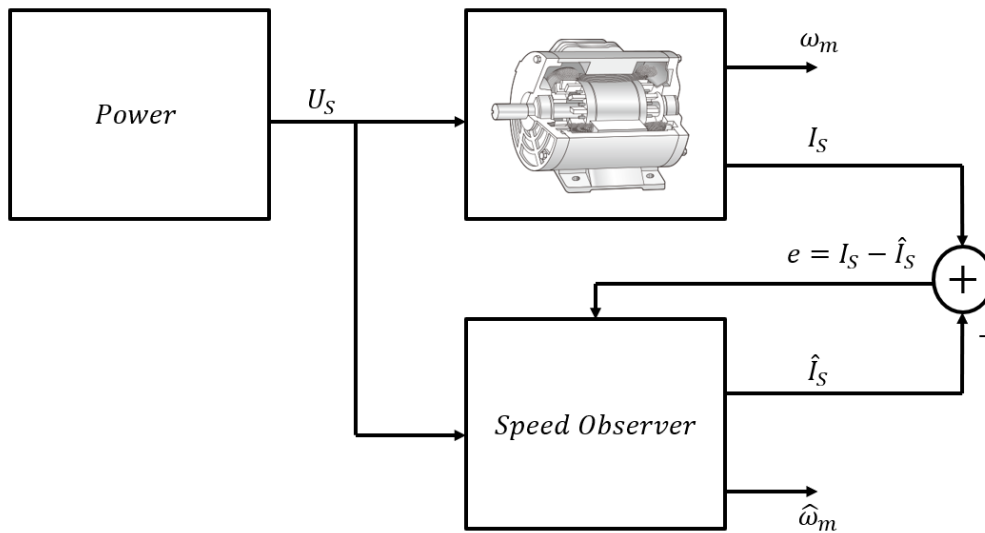


Figure 3.4 - Overview of the simulated system

3.2.1 Pure Sinusoidal System

Firstly, the system was fed direct-on-line (DOL) with a perfect sinusoidal three-phase voltage signal. This portrays an induction machine directly connected to its power source, without any inverter or soft starting device. The simulation parameters can be found in Tables 3.1 and 3.2. It is important to point out that, as shown in Table 3.2, a load of 65 Nm is being applied to the motor for the simulation.

Table 3.1 - Induction Machine Electrical Simulation Parameters

Electrical		Motor				
V	$freq$	R_s	L_1	R_r	L_2	L_M
75 V	76 Hz	8.56 mΩ	0.06292 mH	5.10 mΩ	0.06709 mH	1.0122 mH

Table 3.2 - Induction Machine Mechanical Simulation Parameters

Mechanical		
pp	T_{Load}	$Inertia$
2	65 Nm	0.025 Kg/m ²

Figure 3.5 shows the working principle of the speed observer. It takes the input voltage and current in the dq -reference frame from the simulated induction motor and estimates the stator current (3.2). By comparing the estimated value of the current with the real one, it is then possible to estimate the mechanical speed of the motor ($\hat{\omega}_m$) as shown in (3.9). This value is then converted into the estimated electrical rotor angular frequency ($\hat{\omega}_R$) and used to update matrix $[A]$. This cycle then continues in a closed loop system, outputting updated mechanical speed values.

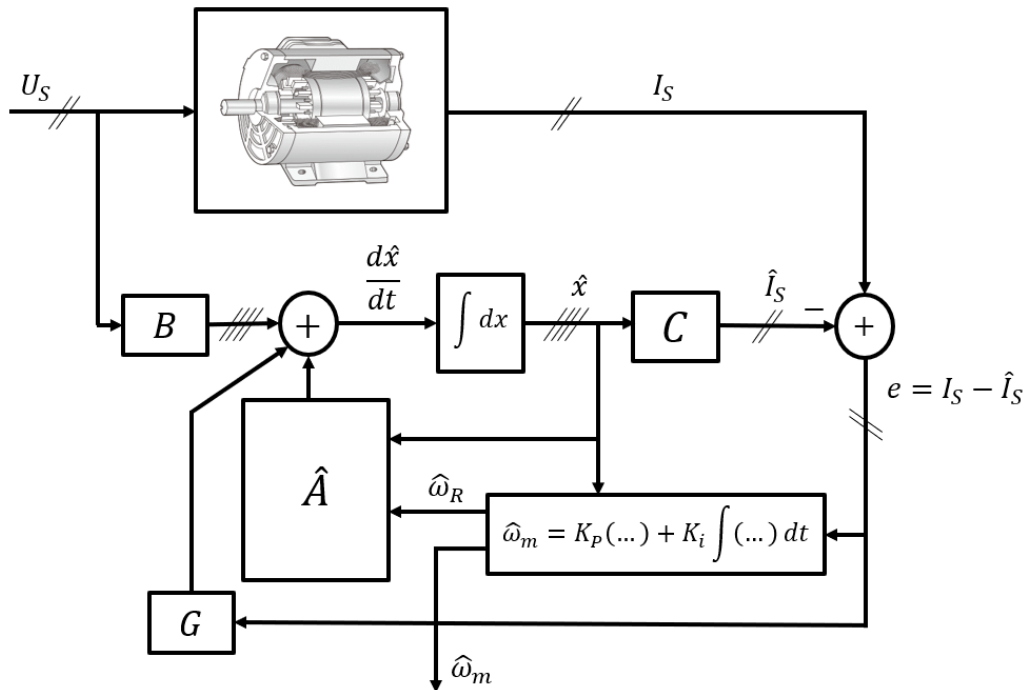


Figure 3.5 - Adaptive speed observer

The electrical parameters of the observer can be seen in Table 3.3.

Table 3.3 - Observer Simulation Parameters (DOL)

Motor					Observer			
R_s	L_1	R_r	L_2	L_M	k	K_I	K_P	f_{samp}
8.56 mΩ	0.06292 mH	5.10 mΩ	0.06709 mH	1.0122 mH	1	20000	20	10 kHz

Comparing Table 3.3 to Table 3.1, it can be seen that the observer was tuned specifically to the parameters of the simulated induction machine, meaning that some factors such as the change of the stator and rotor resistance with temperature, which influences the physical system, will not be taken into account here. As for

the sampling frequency, the lowest possible value was chosen without compromising the observer performance. This was done so that the simulated values could be realistically implemented at a hardware level.

For the observer performance analysis, the induction machine's estimated values, i.e., the mechanical speed,

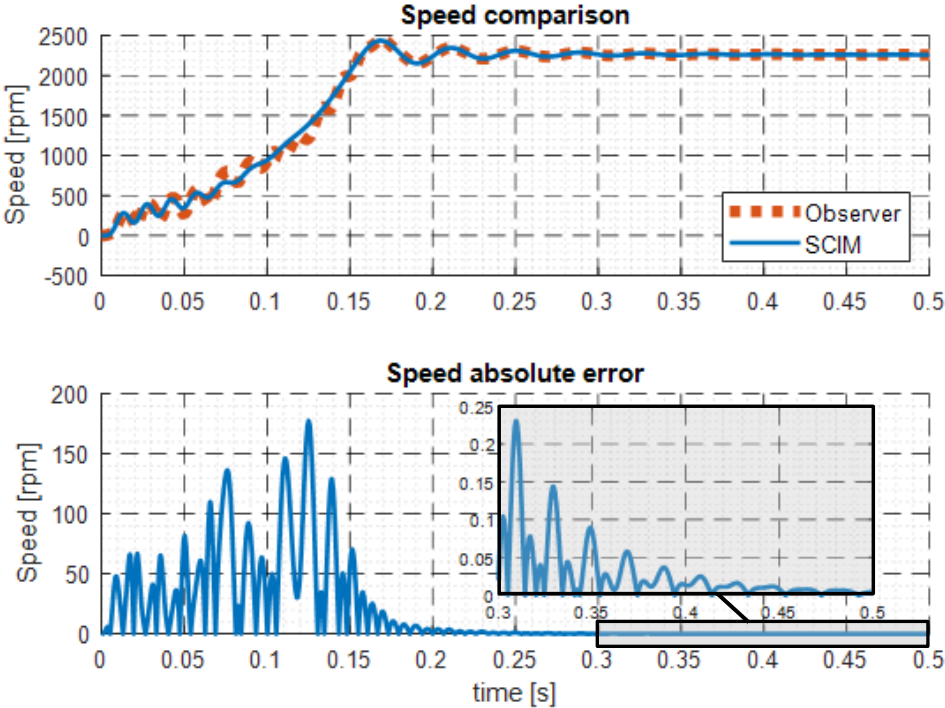


Figure 3.6 - Performance of Speed Estimation (DOL)

torque, and stator currents, will be compared to those coming from the simulated induction motor. This is done using absolute error values, which leads to the following results:

Firstly, for the speed estimation itself, shown in Figure 3.6, we see that the estimator struggles to keep up with speed oscillations common to the transitory period of a direct on-line (DOL) startup. Despite this, the observer concept is solid, as it adapts well to the slower variations and converges with practically no error in steady-state.

The estimated torque values, shown in Figure 3.7, are computed by adapting equation (2.50) to use solely the stator current components shown in Figures 3.8 and 3.9, as these are the state variables chosen for the state-space model of the induction motor.

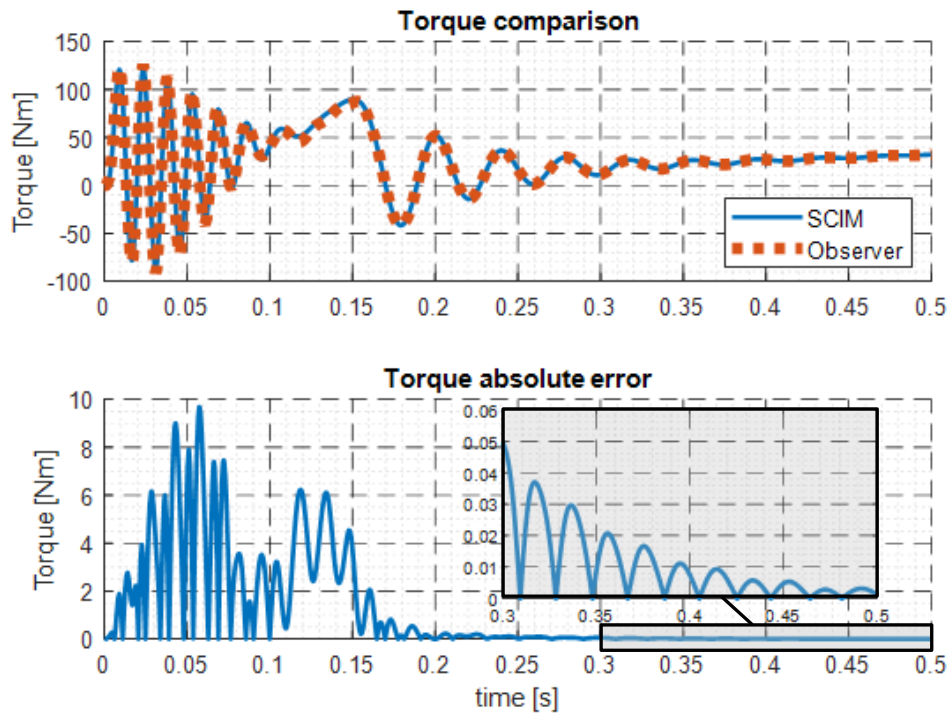


Figure 3.7 - Performance of Torque Estimation (DOL)

Both direct and quadrature current components show current spikes up to ten times the nominal value during startup. Although the observer can somewhat mirror this behavior, this is not a likely scenario. The motor is

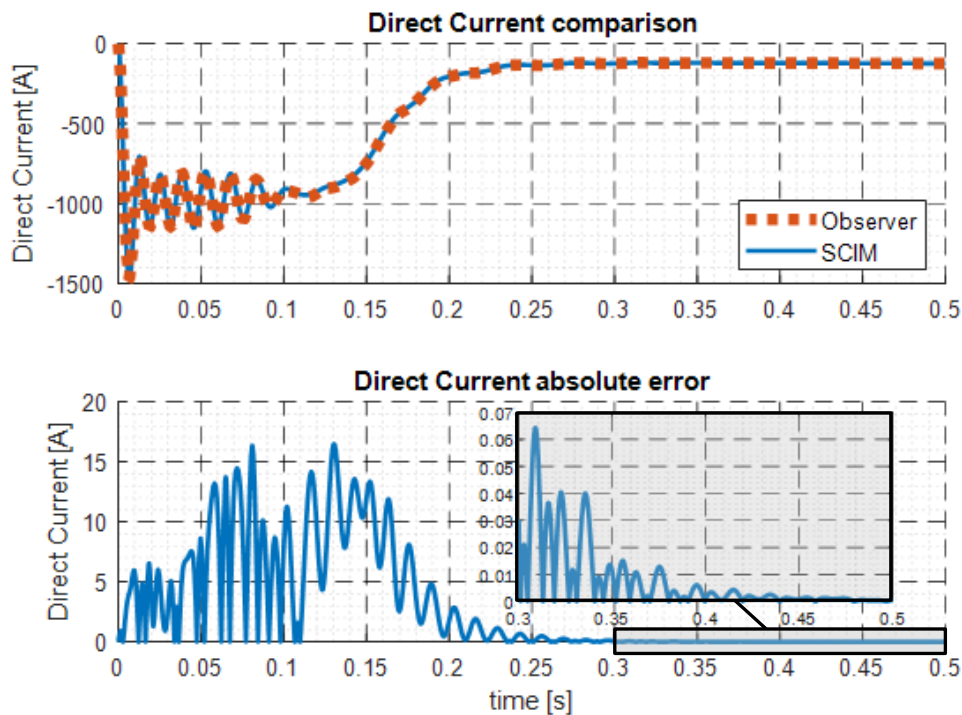


Figure 3.8 - Performance of Direct Current Estimation (DOL)

designed to be part of a vehicle's powertrain and, thus, will not be connected directly to the power grid, as shown in this example.

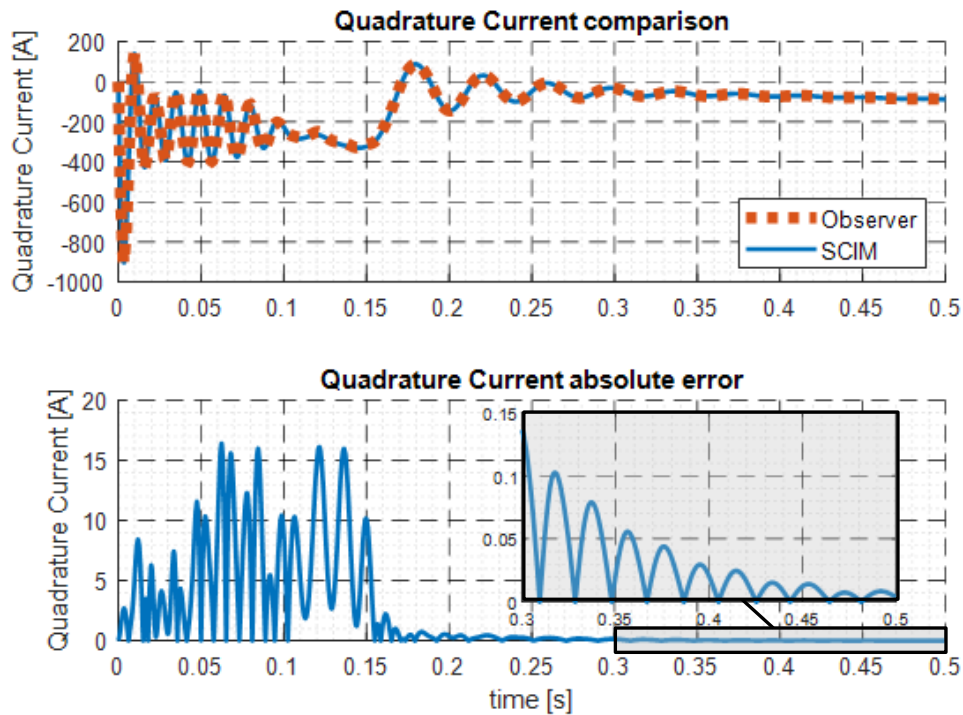


Figure 3.9 - Performance of Quadrature Current Estimation (DOL)

3.2.2 Soft Starter Implementation

As a second approach, the perfect sinusoidal system is maintained, but a scalar command (V/f), shown in Figure 3.10, is added to avoid the massive current spikes seen on startup.

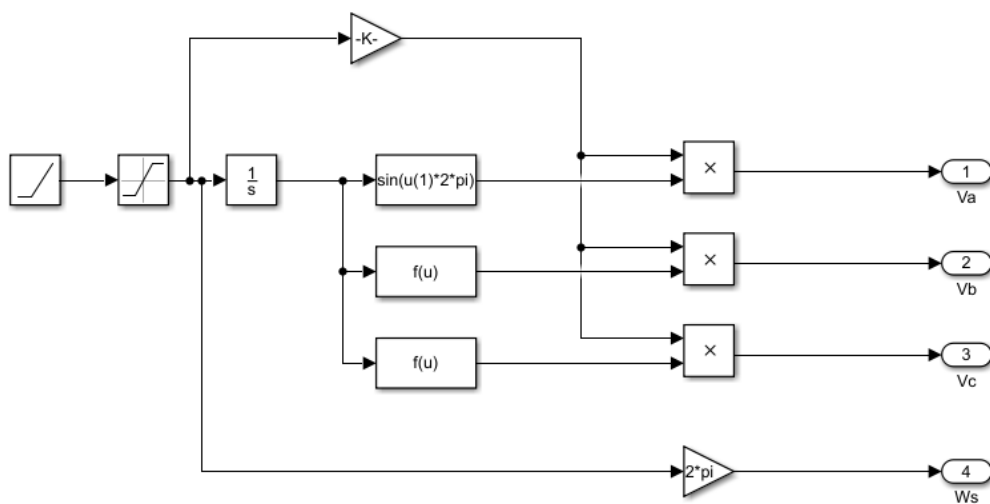


Figure 3.10 - Scalar Command Implementation (V/f)

This is done by slowly increasing the amplitude and frequency of the voltage supplying the motor while maintaining the relation between them. This is done for two seconds until steady-state is reached, resulting in the per-phase input voltage shown in Figure 3.11.

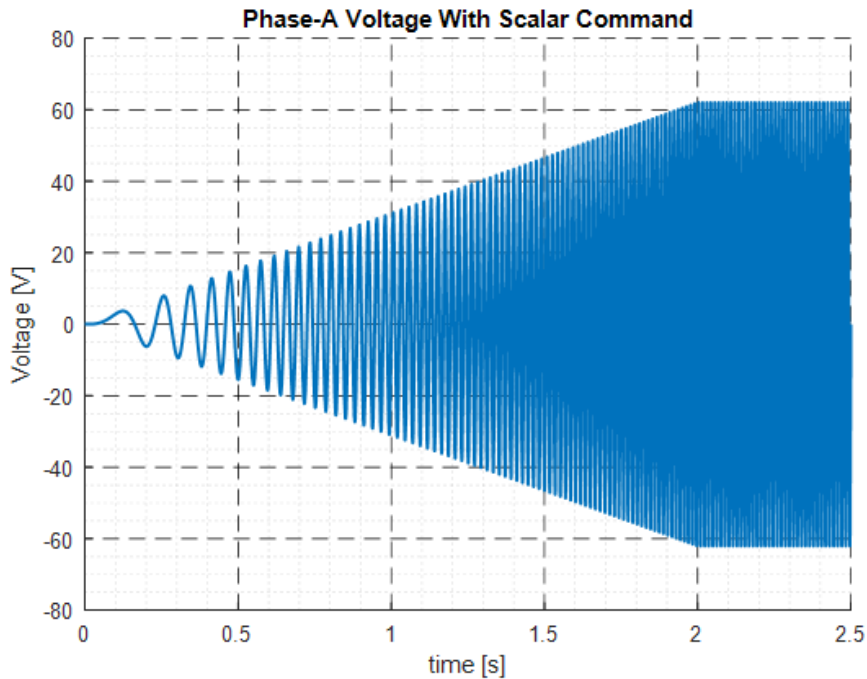


Figure 3.11 - Single-Phase Voltage with Scalar Command (V/f)

With this addition, the observer parameters seen in Table 3.4 are unchanged from the previous section to verify the effectiveness of the V/f implementation. The mechanical simulation parameters seen in Table 3.2 also remain unchanged.

Table 3.4 - Observer Simulation Parameters (V/f)

Motor					Observer			
R_s	L_1	R_r	L_2	L_M	k	K_I	K_P	f_{samp}
8.56 mΩ	0.06292 mH	5.10 mΩ	0.06709 mH	1.0122 mH	1	20000	20	10 kHz

The same electrical and mechanical quantities, i.e., speed, mechanical torque, and both direct and quadrature current components, are tested. This leads to the results shown in Figures 3.12 to 3.15.

As expected, now that the scalar command has been implemented, the observer has no issues estimating the motor’s startup speed and does so with minimal error. As for steady-state, the results are identical to those shown in Figure 3.6, as the V/f implementation has no effect after the two-second mark. This means the observer converges to the desired speed value.

The slight distortion seen in the zoomed-in window of Figure 3.12, is made up of oscillations around the true speed value, as the observer overcorrects itself repeatedly. This is due to the high values of proportional and integral gains used to speed up the observer.

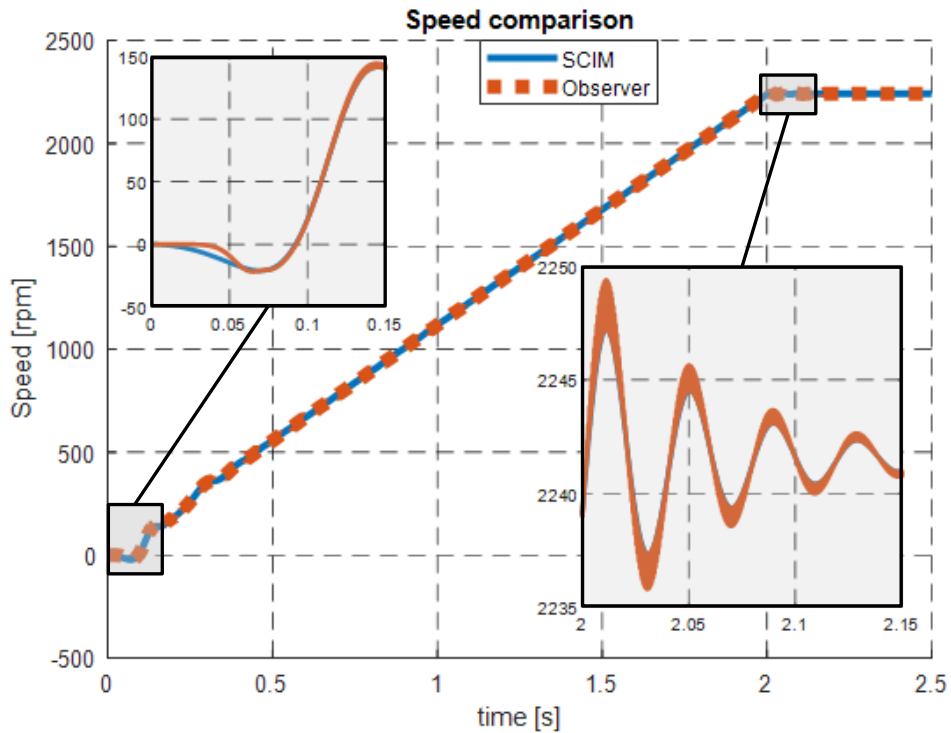


Figure 3.12 - Performance of Speed Estimation (V/f)

As for the torque estimation, seen in Figure 3.13, the observer follows the real values without any issues. A behaviour that was expected due to the lack of disturbances in the signal, aside from the transition to steady state.

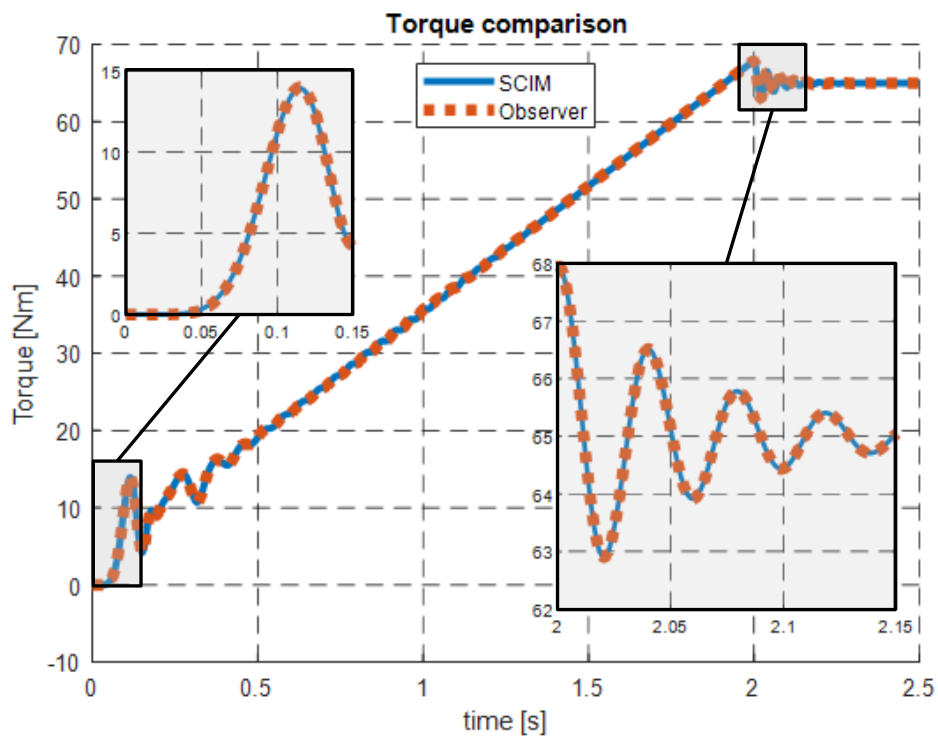


Figure 3.13 - Performance of Torque Estimation (V/f)

Figures 3.14 and 3.15 show the direct and quadrature current components when starting using the scalar command (V/f). Compared to DOL starting, it's seen that the current spikes are now only about 30% higher than the nominal value, an acceptable result as it occurs for a very short amount of time.

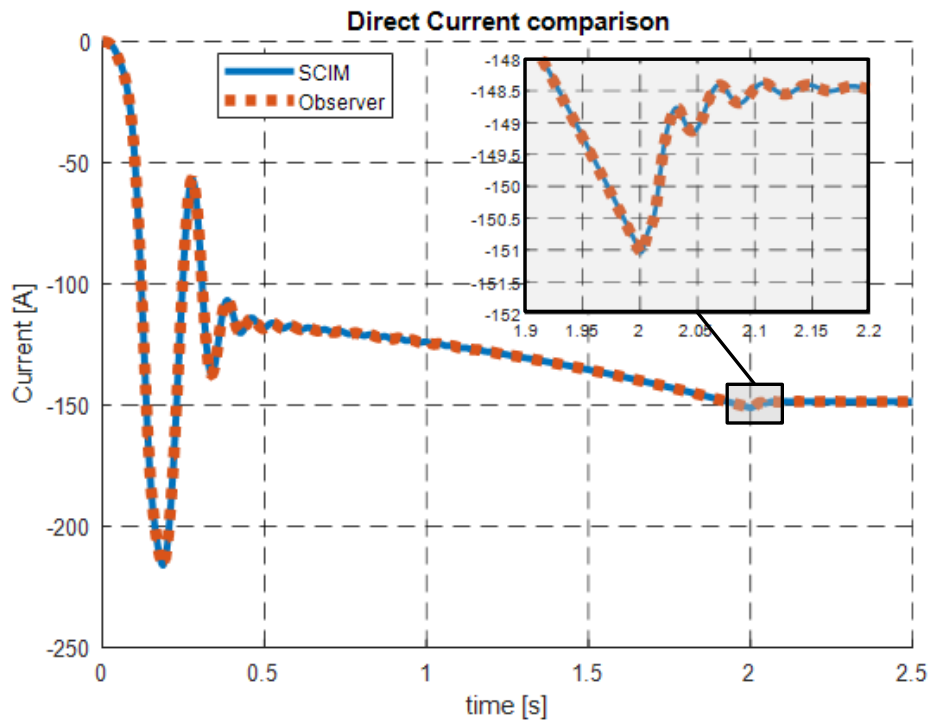


Figure 3.14 - Performance of Direct Current Estimation (V/f)

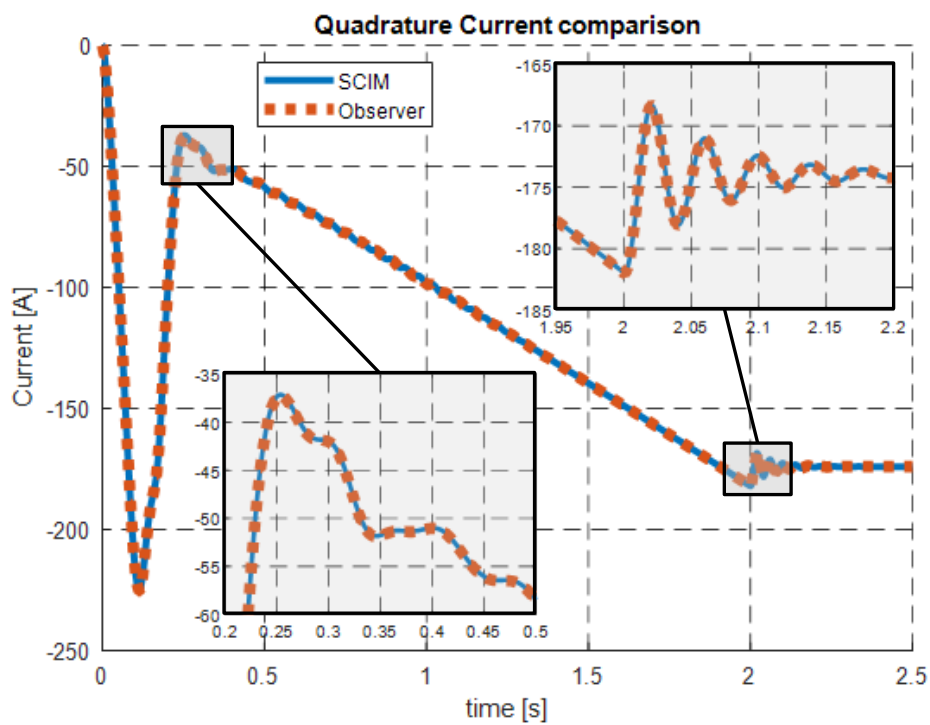


Figure 3.15 - Performance of Quadrature Current Estimation (V/f)

3.2.3 Inverter Implementation

Finally, an inverter is added to account for the ripple present in a real-life application, thus providing a more realistic overview of the constraints that arise from physical implementation. The V/f functionality from the last section was built into the inverter, with a PWM-generator [10] slowly increasing the frequency to the nominal value. Simultaneously, the same is done for the voltage amplitude in the DC power source, maintaining the V/f ratio. This system is shown in the block diagram in Figure 3.16.

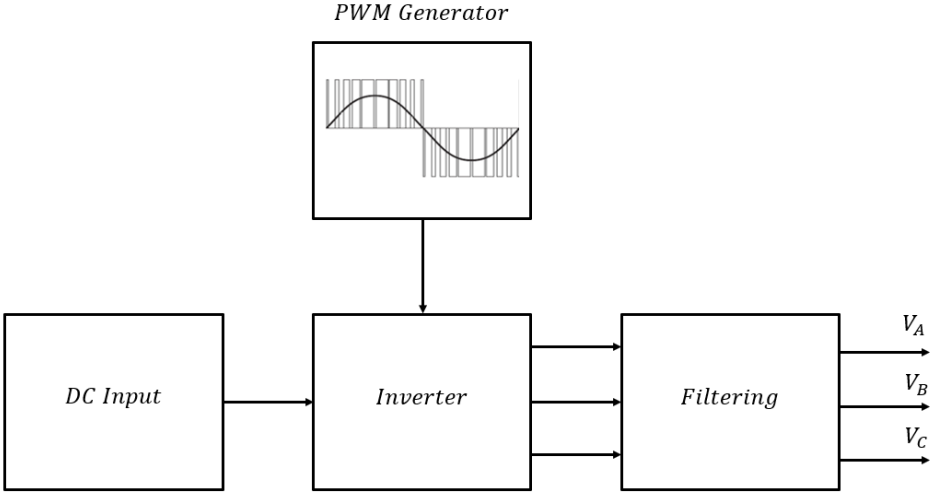


Figure 3.16 – Block-diagram of a Three-phase PWM Inverter with V/f

The inverter is a 3-level, 3-phase PWM system. This was chosen in order to reduce the harmonic content of the input voltage, which can be seen in Figure 3.17.

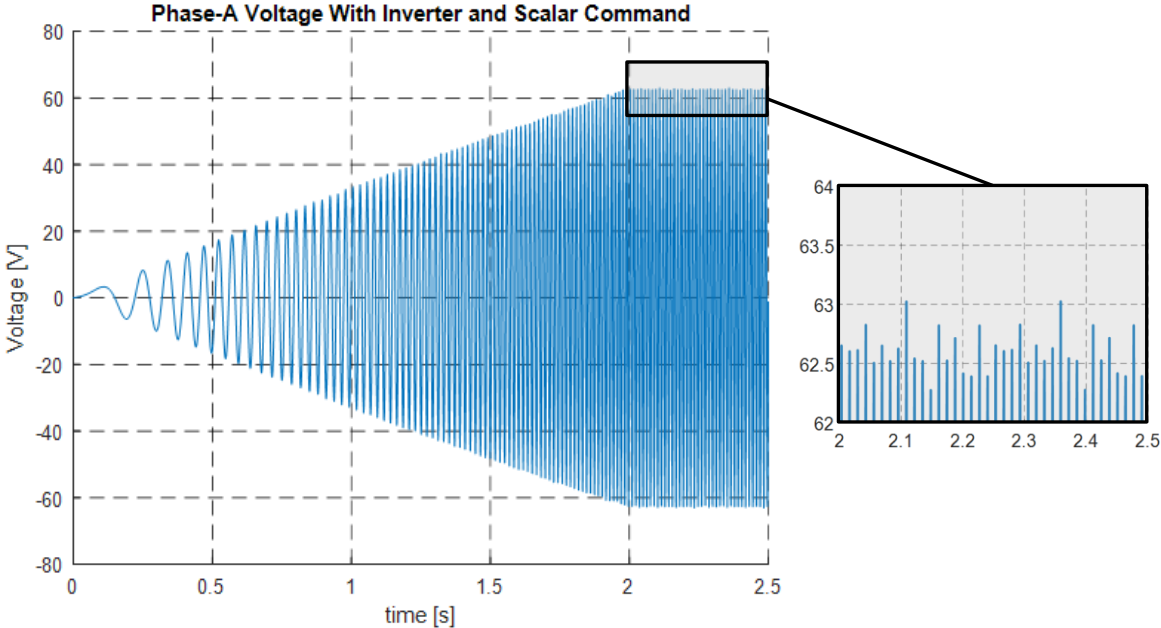


Figure 3.17 - Single-Phase Voltage with Inverter and Scalar Command (V/f)

The inclusion of the inverter led to a new revision of the observer parameters as seen in Table 3.5.

Table 3.5 - Observer Simulation Parameters (Inverter + V/f)

Motor					Observer			
R_s	L_1	R_r	L_2	L_M	k	K_I	K_P	f_{samp}
8.56 mΩ	0.06292 mH	5.10 mΩ	0.06709 mH	1.0122 mH	1.1	50000	5	10 kHz

The integral gain was increased to speed up the observer’s response to the fluctuations caused by the harmonics in the supply voltage. On the other hand, the proportional gain is lowered to reduce the oscillatory patterns seen in the observer values. These were also present in the pure sinusoidal results but are much more prevalent in this case and had to be reduced. As for the mechanical simulation parameters seen in Table 3.2, they again remain unchanged.

In this iteration, the value of the proportionality constant (k) was changed to reduce the observer’s response time. This changes the position of the observer poles, as seen in Figure 3.18. The induction machine poles, shown in Figure 3.19, remain unchanged, as the proportionality constant does not influence its behavior.

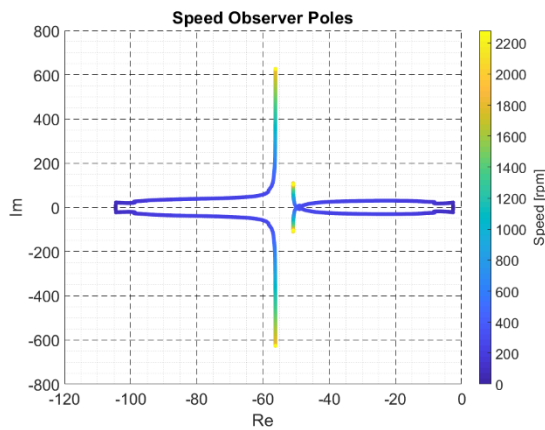


Figure 3.18 - Observer Poles changing with Speed ($k=1.1$)

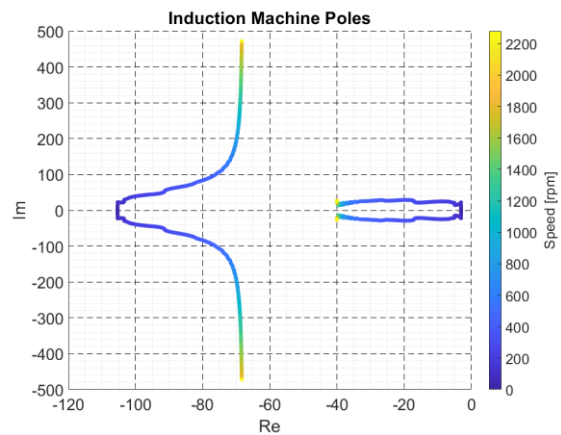


Figure 3.19 - IM Poles changing with Speed ($k=1.1$)

The Observer’s performance is then further analyzed regarding speed, torque, and stator current estimation, leading to the results shown in Figures 3.20 to 3.23. The mechanical parameters shown in Table 3.2 remain in use for this new simulation.

The effect of the harmonics introduced by the inverter is immediately visible in the motor speed plot, displayed in Figure 3.20, as is the observer’s ability to follow its behavior. In this case, the observer has the most trouble in steady-state. The sudden small changes in speed cause the observer to overcompensate, leading to repeated small spikes in the absolute error values.

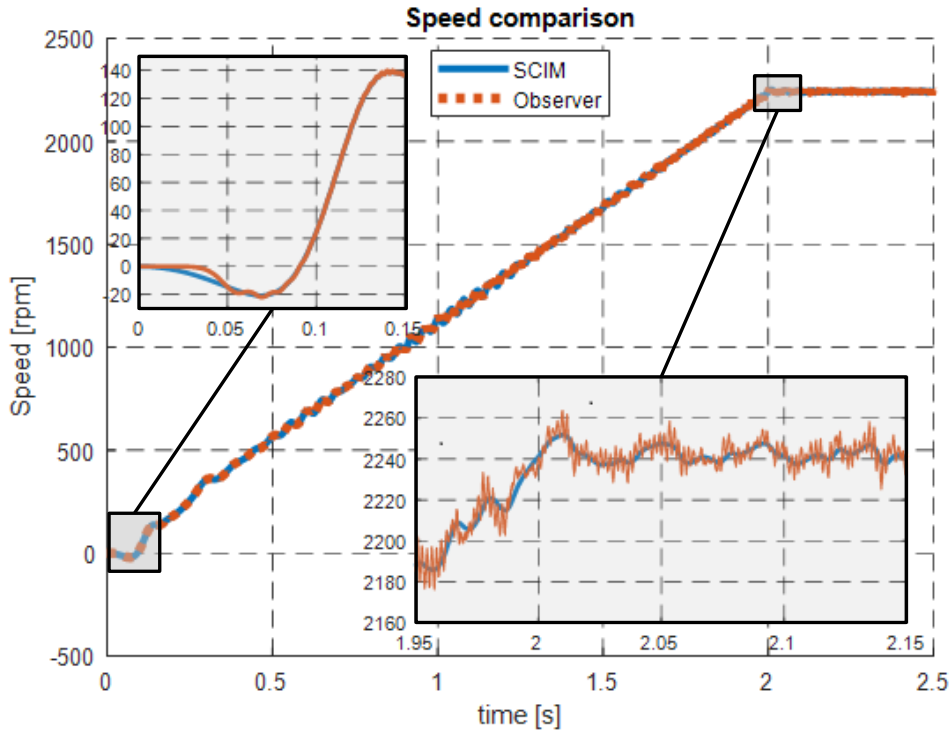


Figure 3.20 - Performance of Speed Estimation (INV+V/f)

The harmonic effect is even more noticeable in the torque, seen in Figure 3.21, as the motor oscillates around the 65 Nm value of the applied load.

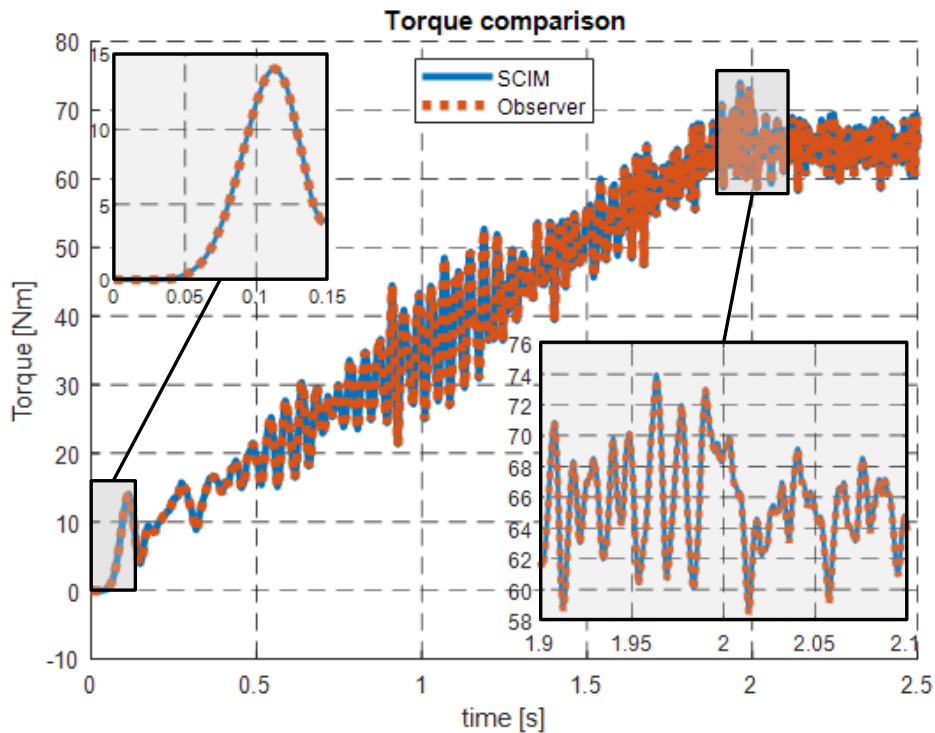


Figure 3.21 - Performance of Torque Estimation (INV+V/f)

Finally, in Figures 3.22 and 3.23, we see the cause for the high torque oscillations, as the machine's direct and quadrature current components also show a large harmonic content, which is mirrored by the observer.

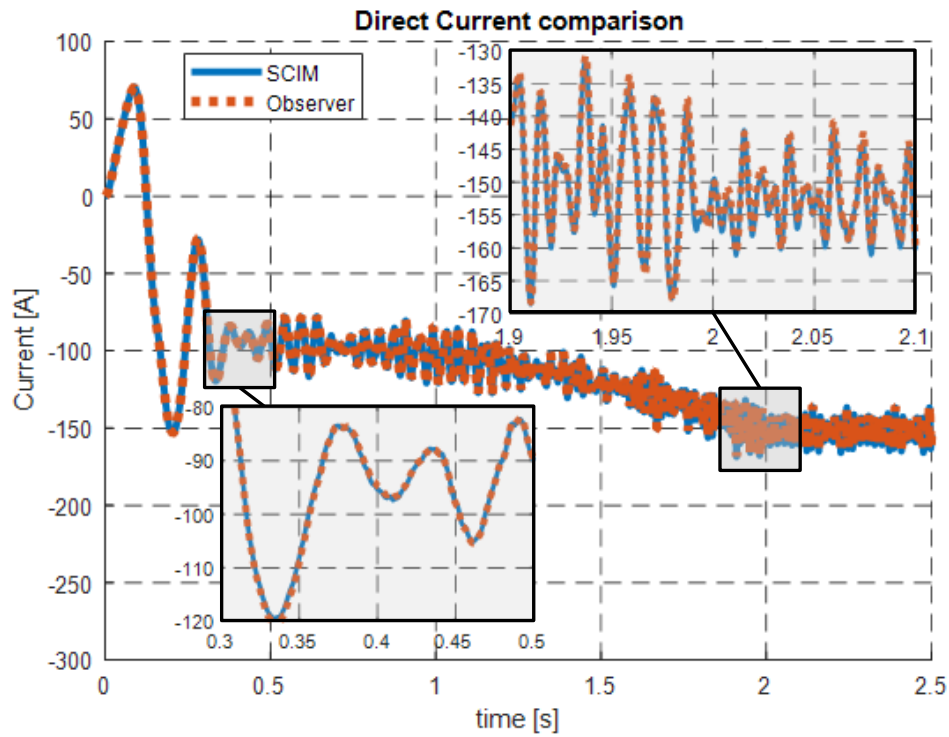


Figure 3.22 - Performance of Direct Current Estimation (INV+V/f)

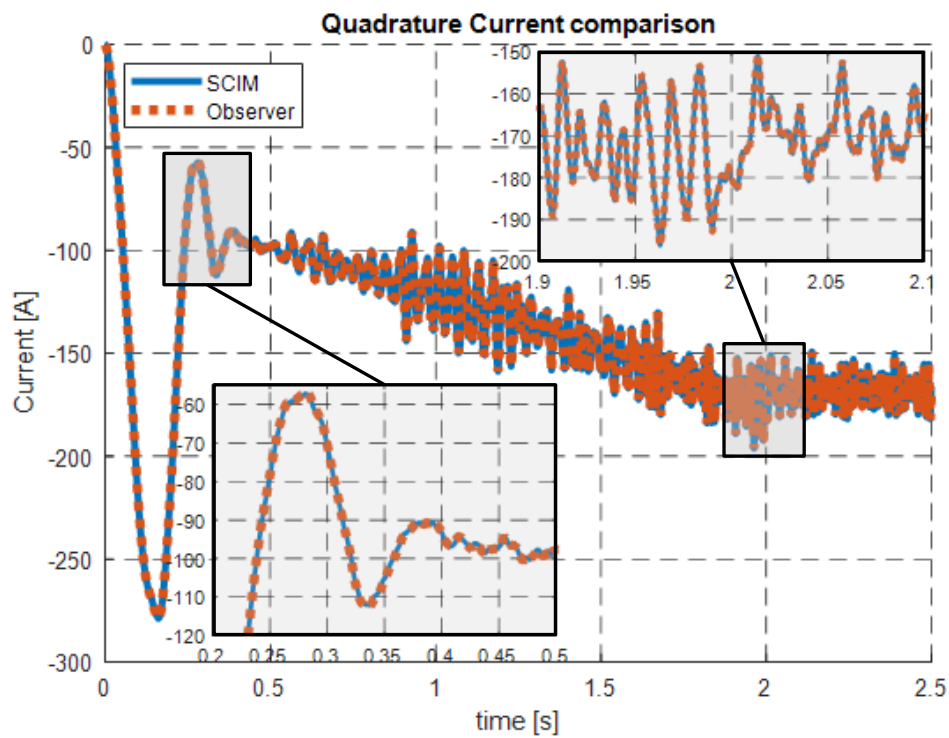


Figure 3.23 - Performance of Quadrature Current Estimation (INV+V/f)

4 Hardware Implementation

Two different board designs were developed to implement the speed observer, utilizing different approaches to acquire the motor's electrical data. The first includes integrated measurement devices that sample the induction machine's stator voltage and current values. This design can either be connected to an Arduino or a Raspberry Pi.

The second design takes advantage of some features of the car's new inverter. Since the inverter supplies the motor with AC power, it already holds all the relevant data. Thus, this design is meant to communicate with the inverter using CAN (Controller Area Network) protocol and send the data to a Raspberry Pi. This board is designed as a shield and is to be directly mounted on the Pi.

4.1 External Acquisition Design

The external acquisition design uses measurement devices present on the PCB and can be divided into two main parts, the voltage acquisition, and the current acquisition circuit. These are portrayed in the render shown in Figure 4.1.

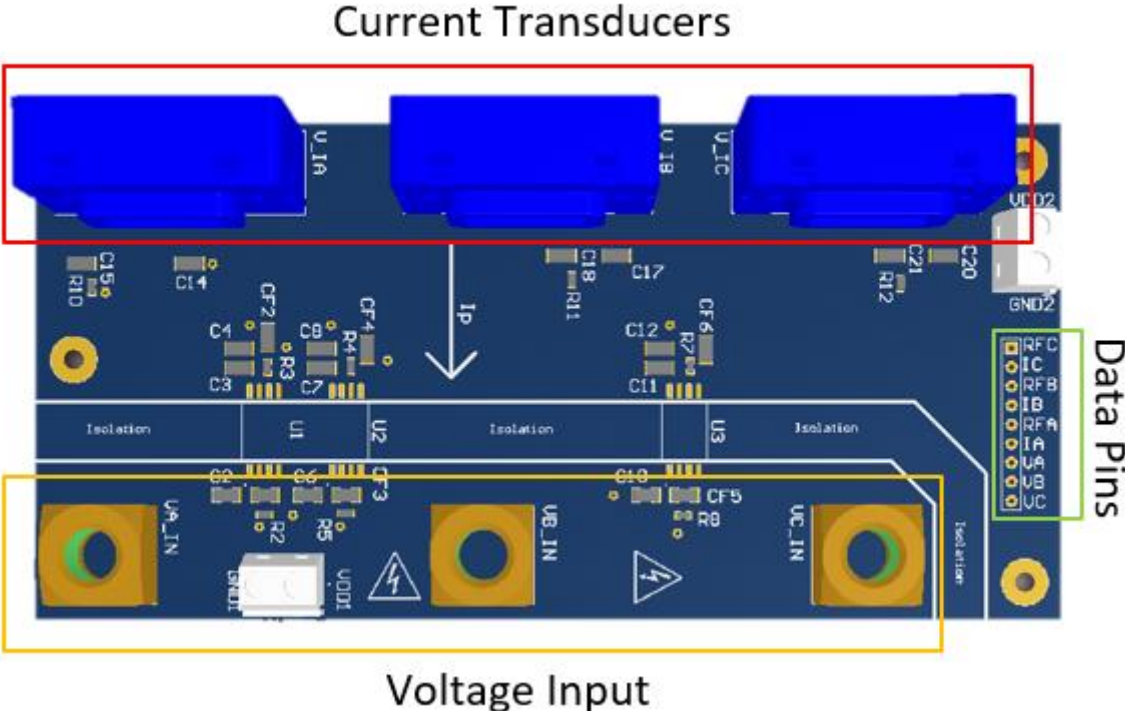
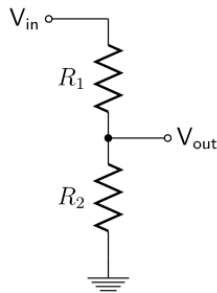


Figure 4.1 - 3D Render of the final PCB Design for the External Acquisition Prototype

4.1.1 Voltage Circuit

For the voltage acquisition the motor's input Voltage of $75 V_{RMS}$, meaning an amplitude of $\pm 75\sqrt{2} \approx \pm 106 V$ passes through resistors R_1 and R_2 , which work as a voltage divider, reducing the voltage to $\pm 212 mV$. This type of circuit is shown in Figure 4.2.



$$V_{out} = \frac{R_2}{R_2 + R_1} V_{in} \quad (4.1)$$

$$\frac{200}{200 + 100000} \cdot 106 = 212 mV$$

Figure 4.2 - Voltage divider

This is done as not to exceed the maximum input of $\pm 300 mV$ of the AMC1301 Isolated amplifier. This IC separates the high voltage part of the board from the rest of the low voltage components as a safety precaution.

This isolation level means the AMC1301 requires separate 5 V sources for its HV and LV side, as can be seen in Figure 4.3, where the schematic for the single-phase voltage acquisition circuit is shown.

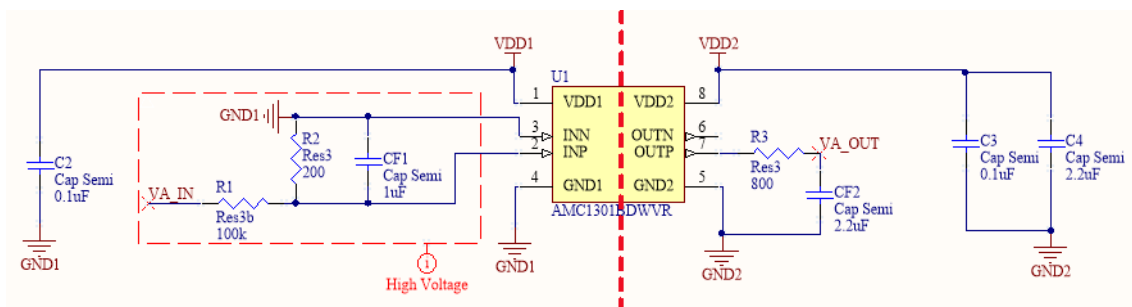


Figure 4.3 - Voltage acquisition circuit for Phase A

Another important factor in the choice of Resistors R_1 and R_2 is the dissipated power.

$$P_{R1} = U \cdot I = \frac{V_{in}^2}{R_1} = 112.36 mW \quad (4.2)$$

$$P_{R2} = \frac{V_{out}^2}{R_2} = 0.225 mW \quad (4.3)$$

This means the value of 100 kΩ for R_1 is required to lower the dissipated power. Capacitors C_2, C_3 and C_4 are decoupling capacitors required by the manufacturer and specified in the datasheet. These are meant to protect the IC from noise fluctuations in the circuit.

Because the motor's voltage supply comes from an inverter, high-frequency harmonics are expected to be part of the acquired signal. Thus capacitors C_{F1} and C_{F2} , will serve as low pass filters to remove them. The first filter, C_{F1} , is placed before the isolation amplifier, while the second, C_{F2} , is placed after.

To get the first filter's cutoff frequency the equivalent resistance of the circuit, R_{eq} , must first be computed. This includes the voltage divider and the OPAMP's internal resistance value, which the manufacturer provides. This circuit is shown in Figure 4.4.

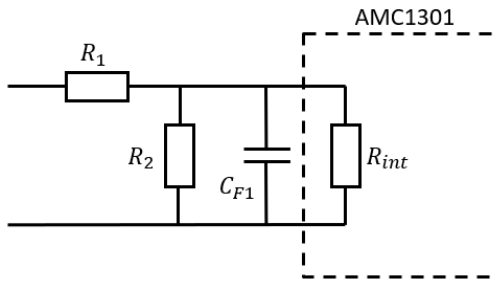


Figure 4.4 - Input low pass filter

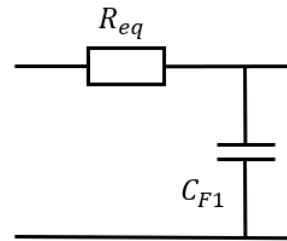
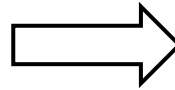


Figure 4.5 - Equivalent circuit of Input low pass filter

Thus, as shown in Figure 4.5, R_{eq} is given by

$$R_1 // R_2 // R_{int} = \frac{1}{\left(\frac{1}{200} + \frac{1}{100000} + \frac{1}{22000}\right)} = 197.8 \Omega. \quad (4.4)$$

After choosing a value of $1\mu F$ for C_{F1} , the cutoff frequency can then be computed, as

$$f_{cut} = \frac{1}{2\pi \cdot R_{eq} \cdot C_{F1}} = \frac{1}{2\pi \cdot 197.8 \cdot (1 \cdot 10^{-6})} = 804.6 \text{ Hz}. \quad (4.5)$$

With an input Voltage of 212 mV, the OPAMP is expected to output a voltage of around 2,3 V. Thus, for the second filter, represented in Figure 4.6, a resistance of 800Ω was chosen to limit the output current to 2,875 mA.

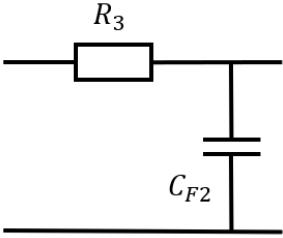


Figure 4.6 - Output low pass filter

A value of 2,2 μF is chosen for CF2, placing the cutoff frequency close to the fundamental harmonic at 76 Hz.

$$f_{cut} = \frac{1}{2\pi \cdot R_3 \cdot C_{F2}} = 90 \text{ Hz} \tag{4.6}$$

4.1.2 Current circuit

As for the current acquisition circuit, a current transducer, HAIS 200-P, was selected as the measurement device. This transducer provides galvanic isolation between the primary circuit, i.e., the motor’s electrical supply and the secondary, low voltage, acquisition board.

The current transducer is based on a Hall-effect measuring principle. This means a magnetic field is measured by applying a current to a thin metal strip. In the presence of a magnetic field the current flow, i.e., the flow of electrons, will be deflected, as seen in Figure 4.7, which creates a potential difference in the strip, a voltage proportional to the applied field.

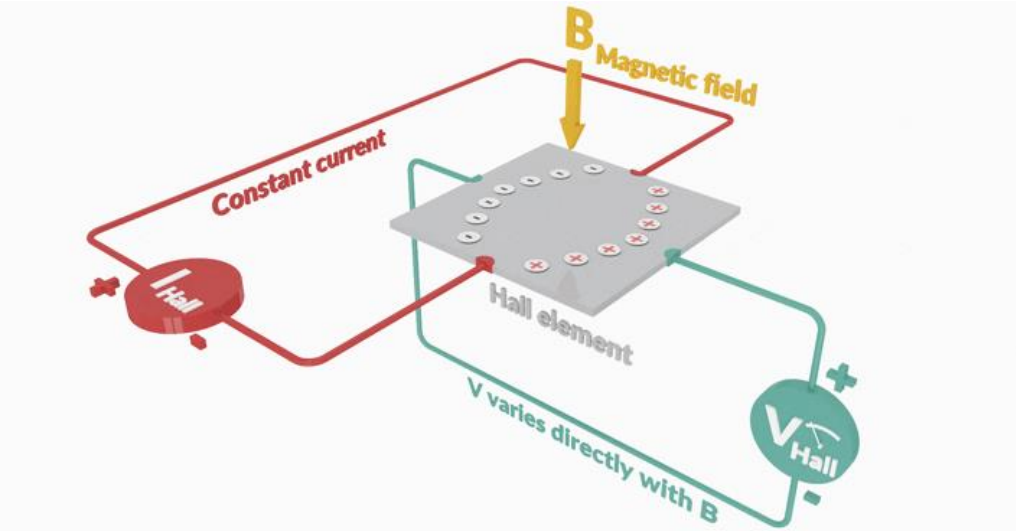


Figure 4.7 - Hall-effect measuring principle [15]

In the case of a current transducer, this magnetic field was created by the cable's current flow going through its core, as shown in Figure 4.8. This means the induced field is proportional to the current that is to be measured, and thus, the current can be computed from the transducer's output voltage.

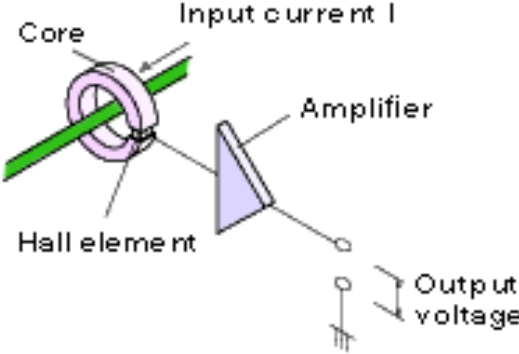


Figure 4.8 - Main elements of a current transducer [11]

In this case the expression is given by the manufacturer in the form of

$$V_{OUT} = V_{REF} \pm \left(0.625 \cdot \frac{I_p}{200} \right), \tag{4.7}$$

where V_{REF} is a reference voltage of 2.5 V and I_p is the current flowing through the sensor.

A relevant factor in the choice of the HAIS 200-P was its low supply voltage of 5 V, meaning it can be easily powered alongside the other low voltage components from an Arduino or Raspberry Pi board.

As can be seen in Figure 4.9, each current transducer requires three decoupling capacitors, C_{13} and C_{14} with 47nF and C_{15} with 4.7nF, as well as a load resistance of 10 kΩ. All these values are specified by the manufacturer in the product's datasheet.

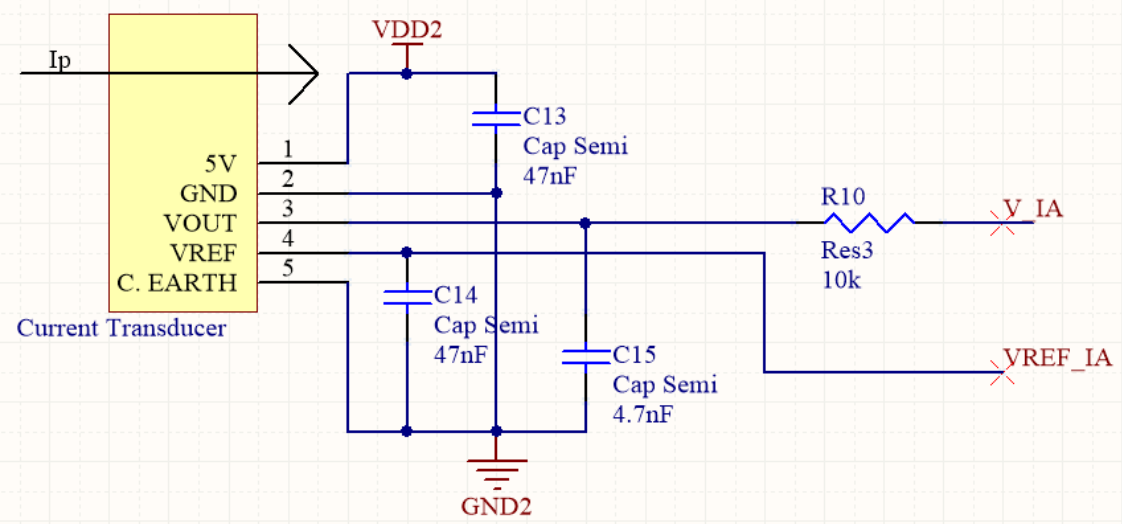


Figure 4.9 - Current acquisition circuit for Phase A

The motor has a nominal current value of 157 A_{RMS}, meaning an amplitude of $\pm 157\sqrt{2} \approx \pm 222$ A. Thus, from equation (4.7), it was computed that the expected output of the transducer would be a maximum of 3.2 V for the positive and 1.8 V for the negative current values, thus staying within the operating range of an Arduino of 0 V to 3.3 V.

The Arduino will acquire this voltage and compute the current I_p passing through the transducer. The current values for each phase would then be used as inputs for the speed observer.

4.2 CAN-HAT Design

The second design is more akin to a communication device, as it bridges the gap between the inverter’s CAN interface and a Raspberry Pi. Controller area network (CAN) is an electronic communication bus consisting of two wires, CAN High and CAN Low. It is a simple and low-cost solution that is widely used in the automotive industry. In this case, specifically, CANopen, a CAN-based communication protocol will be used.

While CAN refers to a physical connection, CANopen builds upon this and provides a higher layer protocol that standardizes the communication between devices and applications from different manufacturers. An overview of this system can be seen in Figure 4.10.

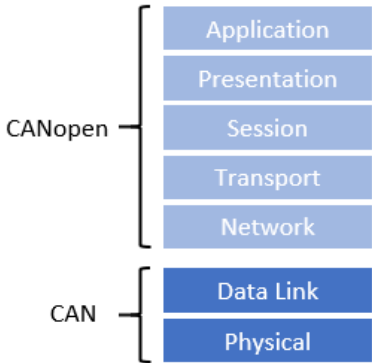


Figure 4.10 - Communication functions of CAN and CANopen

The CAN’s message format can be divided into several parts, represented in Figure 4.11. The first consists of an 11-bit CAN ID commonly referred to as a communication object identifier (COB-ID). This identifier consists of 4 bits designated for the function code, i.e., the nature of the message, and 7 bits for the node ID that identifies the message destination. This is important, as in a CAN bus all messages are available to all nodes in the network instead of being sent directly to a single point.

The rest consists of a control bit (RTR), 4 bits that specify the length of the data being sent, and finally 8 bytes are reserved for the data itself.

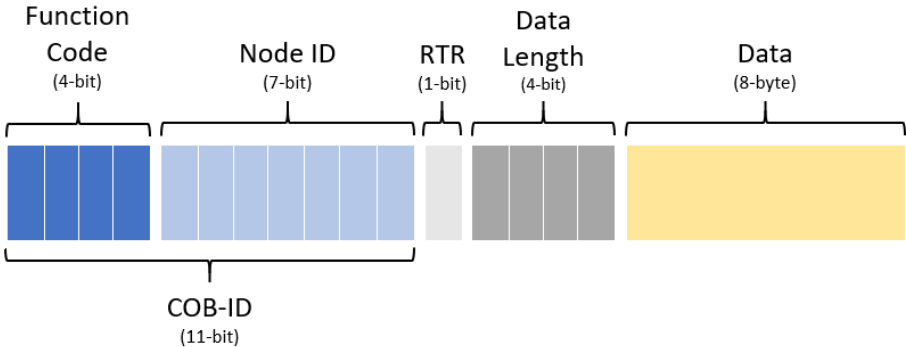


Figure 4.11 - CANopen message layout

Regarding the design of the prototype itself, a choice was made early on that the PCB would be designed as a “hat” meaning it would have the same dimensions and subsequently be mounted on top of the Raspberry Pi, utilizing its GPIO pins for communication.

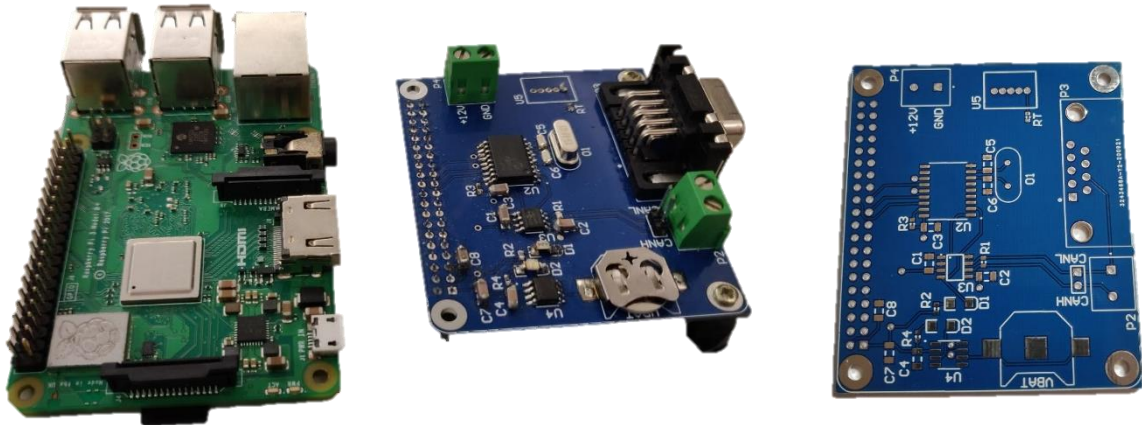


Figure 4.12 - Raspberry Pi3 (left) and CAN-Hat Prototype Board, before (right) and after (center) assembly

4.2.1 Hardware and Circuit Design

The Block diagram in Figure 4.13 shows the basic principles behind this design. The Raspberry Pi board communicates with a CAN controller through its GPIO pins, which then converts the messages according to CAN specifications. These are then sent to the inverter with a CAN transceiver. This process works both ways, with the inverter then sending data back to the controller, which is then decoded and read by the Raspberry Pi.

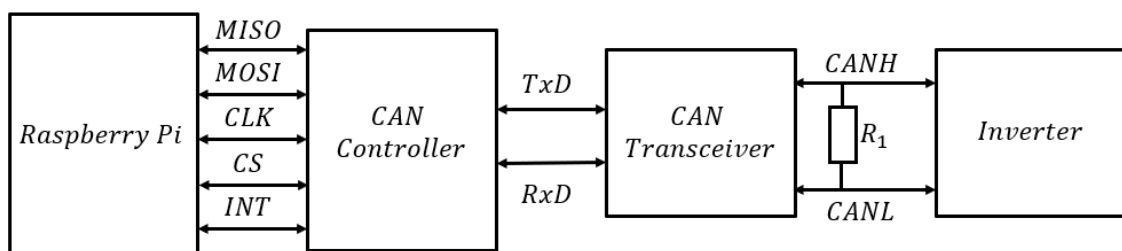


Figure 4.13 - Block diagram of the CAN-HAT PCB Design

The Raspberry Pi 3 was chosen for its large processing capability, versatility, and relatively compact size. The immense processing power was an important factor as the speed observer requires a high sampling frequency to be accurate. It also allows for this prototype to be used in other projects involving the VIENA electric vehicle.

The Raspberry Pi will be powered using the 5V GPIO pins. These are represented in

Figure 4.14 as pin 2 and 4.

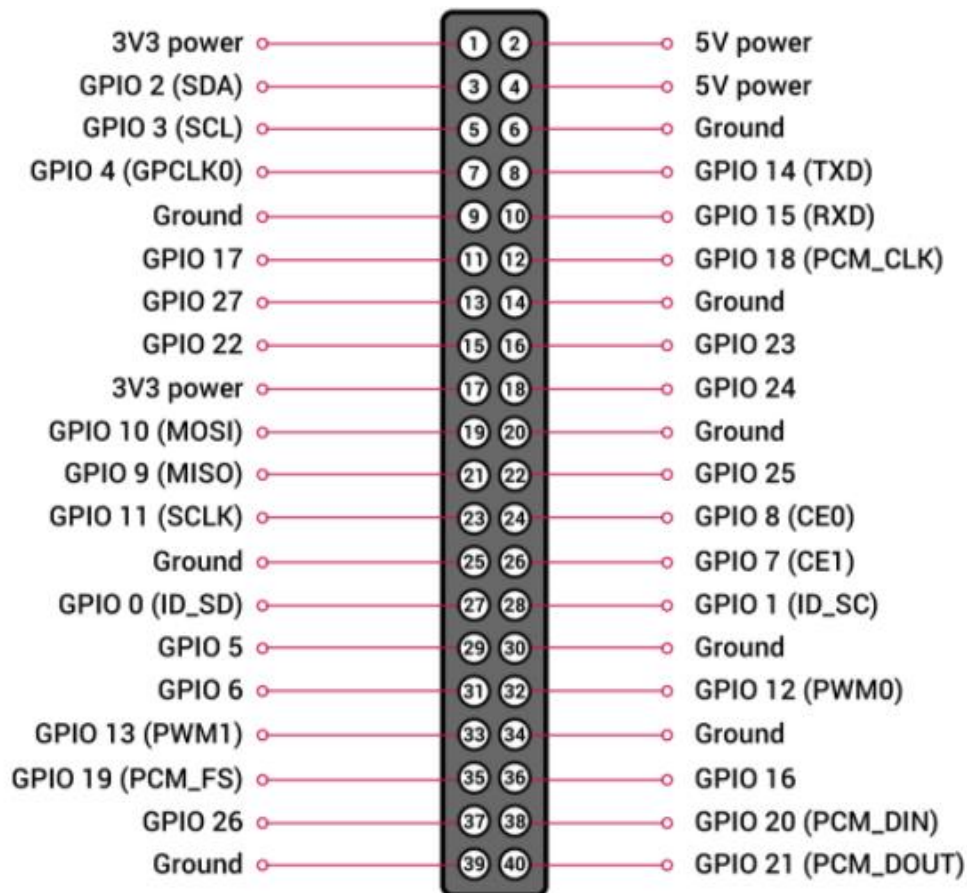


Figure 4.14 - Raspberry Pi GPIO Pin Layout [12]

The supply voltage required to power the ICs will be delivered through the 3V3 pin on the Raspberry Pi (Pin 1). As for the communication with the CAN controller, it will be assured through pins 19, 21, 22, 23, and 24. These are the Raspberry's Serial Peripheral Interface (SPI), an interface that allows the Raspberry to control, or be controlled by external hardware, through a master/slave relationship.

Additionally, pins 3 and 5 are used for Inter-Integrated Circuit Communication (I2C) between the Raspberry and a third IC used as a real-time clock. This interface allows for basic tasks that are integral to the Raspberry to be outsourced to external components, thus freeing computing power that would otherwise be in use.

The vehicle's 12V battery supplies the Raspberry's power, lowered using an OKR-T/3 DC/DC converter to the desired value of 5 Vdc. The 268 Ω precision resistor, R_{Trim} , seen in Figure 4.15, regulates the output voltage of the DC/DC converter.

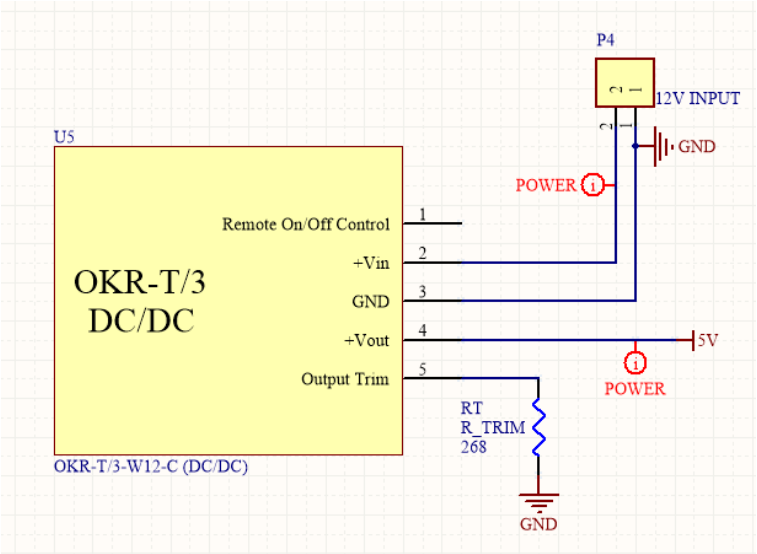


Figure 4.15 - DC/DC Step-down Converter

This value was computed using the expression given by the manufacturer, as

$$R_{TRIM} = \frac{1.182}{(V_{out} - 0.5919)} \tag{4.8}$$

This power supply configuration was developed to allow for use in a moving vehicle where power outlets are not available. Alternatively, the Raspberry Pi can be powered with the included 5 V, 2.5 A charger, as this is enough to support the CAN-Hat's additional power requirements.

The converter was eventually desoldered from the board, as USB-power proved more convenient along the testing process. This was due to the extra cabling and DC power supply required for the DC/DC converter operation, along with the constant coil whine present when it was connected to the circuit.

The design uses the MCP2515 as a CAN network controller. The controller converts data to and from CAN messages, allowing the user to communicate with any network device. In this case, it will be used to request voltage and current data from the vehicle’s inverter, thus allowing for the intended real-time speed estimation. This specific controller was chosen as supported by the Raspberry Pi operating system (Raspbian) and its GPIO (SPI Bus).

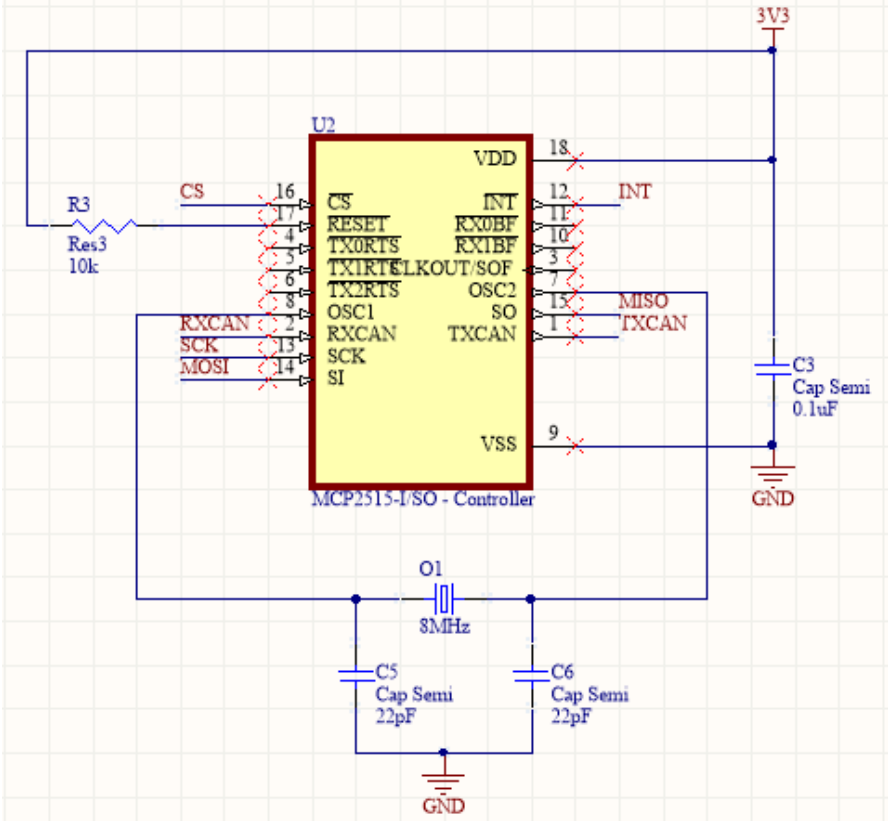


Figure 4.16 - Electrical circuit of the MCP2515 CAN Controller

As seen in Figure 4.16, the controller circuit also includes an oscillator. This is used to set the bitrate of the CAN bus, resulting in a maximum of 500 kbit/s set by the 8 MHz oscillator.

It’s important to note that all nodes on a CAN bus must have the same bitrate, meaning that if another device, for example, a CAN analyzer, were to be connected to the same bus, it would also have to be set to the 500 kbit/s limit to be able to communicate.

The MCP2562 CAN transceiver serves as an interface between the controller, and the physical two-wire CAN bus. This circuit includes the I/O responsible for the connection to the inverter’s CAN interface, either through the two-pin headers designated P2 or a DB9 male connector. Meanwhile, the transceiver also communicates with the controller through the TXCAN (transmit) and RXCAN (receive) pins. An LED was added to the receive pin to indicate if a message is being sent from the Raspberry to the CAN Bus.

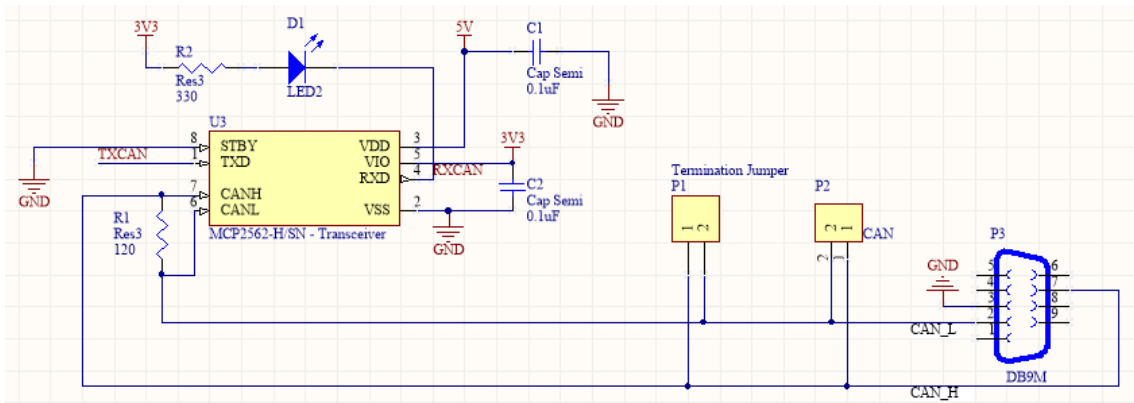


Figure 4.17 - Electrical circuit of the MCP2562 CAN Transceiver

An important component to mention is the termination resistor R_1 , that is placed between the CANH and CANL wires. This is a 120 Ω resistor chosen according to ISO11898 [13] that must be present in all the bus connected nodes to avoid signal reflections.

The design also includes a DS3231MZ real-time clock (RTC), shown in Figure 4.18. This dedicated clock is meant to ensure that the Raspberry is not tasked with timekeeping while running the CAN protocol, which can easily and cheaply be done at the hardware level to conserve processing power. The circuit also holds a coin battery to maintain power to the RTC when powered off, thus maintaining accurate time.

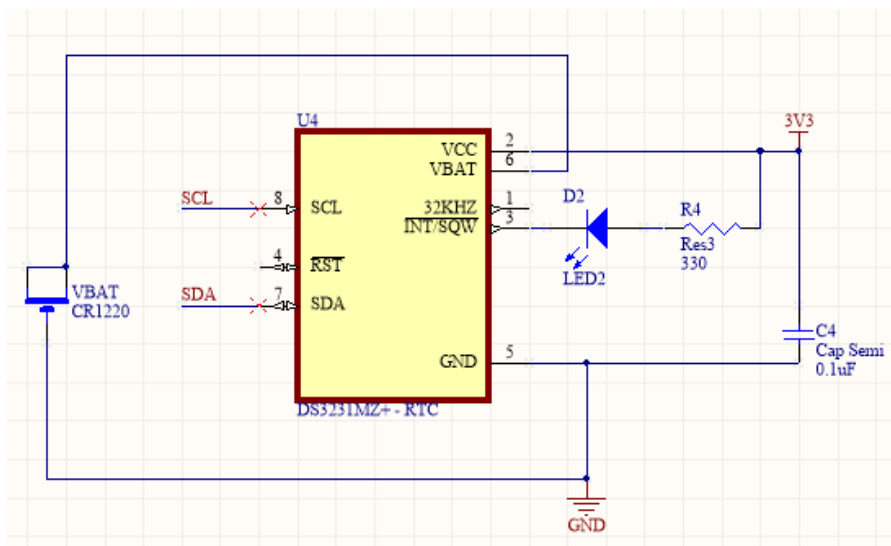


Figure 4.18 - Electrical circuit of the DS3231MZ Real-Time Clock (RTC)

4.2.2 Configuration

With the PCB completed, the setup now requires some software configuration [14]. Firstly, Raspberry Pi OS was installed on the Pi 3 to serve as the operating system, allowing it to work independently if needed. Despite this, normal operation will be conducted by remotely accessing the Raspberry through a laptop. This will be done using VNC viewer, a software that allows for complete control of the Raspberry OS desktop.

To remotely access the Pi 3, we first need to connect it to a network, either through Wi-Fi or ethernet, and find its IP address. This can be done in the Windows command prompt by using the command `arp -a`. This lists all the IP addresses connected to your network and their corresponding mac address, as shown in Figure 4.19.

```
C:\Users\Galamba>arp -a
Interface: 192.168.1.63 --- 0xb
Internet Address      Physical Address      Type
192.168.1.1           00-4f-4e-11-fd-45    dynamic
192.168.1.86          b8-27-eb-d1-83-56    dynamic
192.168.1.110         34-41-5d-d0-66-37    dynamic
192.168.1.255         ff-ff-ff-ff-ff-ff    static
224.0.0.2             01-00-5e-00-00-02    static
224.0.0.22            01-00-5e-00-00-16    static
224.0.0.251           01-00-5e-00-00-fb    static
224.0.0.252           01-00-5e-00-00-fc    static
239.255.255.250       01-00-5e-7f-ff-fa    static
255.255.255.255       ff-ff-ff-ff-ff-ff    static
```

Figure 4.19 - arp -a command output example

The easiest way to recognize the Pi 3 is through its MAC address as the first 3 bytes of every unit are fixed and correspond to the organizationally unique identifier associated with the Raspberry Pi Foundation, B8:27:EB [12]. Note that this is no longer the case in newer Pi 4 models.

With the IP address, we can now access the Raspberry and configure it to enable the SPI and I2C communication described in the last sub-chapter.

For a first-time setup, the installation of the MCP2515 controller driver is also required. This is followed by the configuration of the controller with the desired values. This is done using the `sudo nano /boot/config.txt` command in the Raspberry terminal and adding the following instructions to the file:

```
#CAN bus controllers
dtoverlay=mcp2515-can0,oscillator=8000000,interrupt=25
dtoverlay=spi-bcm2835
```

Here the utilized CAN bus is designated `can0`, and the oscillator frequency is set to the appropriate frequency of 8 MHz.

Another command, `sudo nano /etc/network/interfaces` is then required to set the bus bitrate and some other parameters.

```
auto can0
iface can0 inet manual
    pre-up /sbin/ip link set can0 type can bitrate 250000 \
    triple-sampling on restart-ms 10
    up /sbin/ifconfig can0 up
    down /sbin/ifconfig can0 down
```

Note that in this example, the bitrate was set to 250 kbit/s, despite higher speeds up to 500 kbit/s being possible.

4.2.3 Testing

Now that the configuration process is complete, we move on to testing the designed CAN-Hat prototype. For this purpose, a CAN Bus Analyzer was used. This, together with its proprietary software, serves as the second node on the network, which allows the user to monitor its traffic and communicate with other nodes. This setup is shown in Figure 4.20.

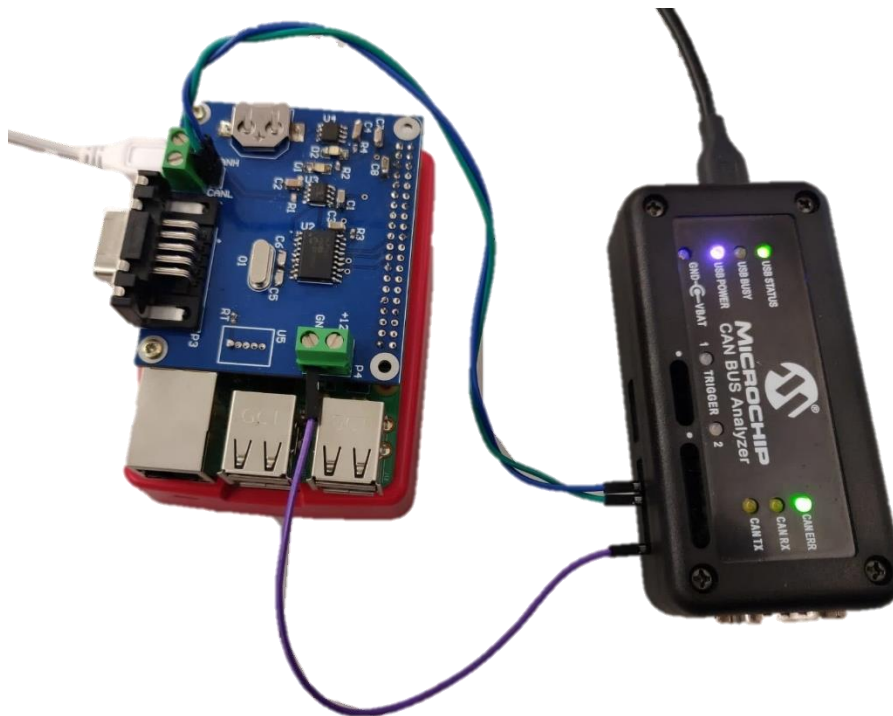


Figure 4.20 - CAN-Hat Prototype connected to a CAN Analyzer

As was mentioned in subsection 4.2.1, all nodes in a CAN bus need to be configured to the same bitrate. This is also true for the analyzer, which was also set to 250 kbit/s.

As for the connections between the analyzer and the CAN-Hat, only the pins CANH and CANL, in addition to common ground are required, as the analyzer is USB-powered. The analyzer's pinout is shown in Figure 4.21.

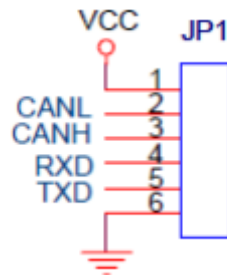


Figure 4.21 - CAN Analyzer Pinout

The first step for the testing was to send a message from the analyzer to the CAN bus and check if this was received and correctly interpreted by the CAN-Hat prototype and if the message readable in the Raspberry.

This test was originally unsuccessful, as neither side of the communication pair detected either transmission or reception. After careful examination of the circuit schematics and PCB layout, it was found that the CANL and CANH pins that connect to the analyzer interface of Figure 4.21 had been mislabeled and were switched. This was promptly addressed, and after the change, the communication tests were rerun.

Thus, for the reception test, a repeated message was sent three times by the CAN analyzer software, as seen in the Transmit window in Figure 4.22. This shows a simplified message layout, when compared to the one in Figure 4.11, as only the message ID, data-length (DLC), and the message itself are shown.

Transmit													
FORMAT	ID	DLC	DATA 0	DATA 1	DATA 2	DATA 3	DATA 4	DATA 5	DATA 6	DATA 7	PERIOD (msec)	REPEAT	TRANSMIT
HEX	123	8	11	22	33	44	55	66	77	88	100	3	Send
HEX											0	0	Send

Rolling Trace												
TRACE	ID	DLC	DATA 0	DATA 1	DATA 2	DATA 3	DATA 4	DATA 5	DATA 6	DATA 7	TIME STAMP (sec)	TIME DELTA (sec)
TX	0x123	8	0x11	0x22	0x33	0x44	0x55	0x66	0x77	0x88	1.0880	0.000
TX	0x123	8	0x11	0x22	0x33	0x44	0x55	0x66	0x77	0x88	1.0880	0.000
TX	0x123	8	0x11	0x22	0x33	0x44	0x55	0x66	0x77	0x88	1.0880	0.000

Figure 4.22 - Transmission and Bus Monitoring Windows of the Analyzer Software

The result can be seen in the rolling trace window of the CAN analyzer software. Three messages with ID 123 were detected in the bus with the trace TX, meaning they were successfully transmitted by the node that is detecting them.

To check if the Raspberry received these messages, the command `candump can0` is used. This command shows all the traffic present in the selected bus, in this case bus `can0`, which results in the output seen in Figure 4.23. This, again, shows the message ID, the data-length and the message.

```
pi@raspberrypi:~ $ candump can0
can0 123 [8] 11 22 33 44 55 66 77 88
can0 123 [8] 11 22 33 44 55 66 77 88
can0 123 [8] 11 22 33 44 55 66 77 88
```

Figure 4.23 - `candump can0` output example

As expected, the messages were correctly received and are accessible in the Raspberry, meaning data sent in this manner could be used for the desired observer computations.

Since the messages were received successfully, we move onto the transmission test. This is done using the `cansend can0` command as follows:

```
pi@raspberrypi:~ $ cansend can0 127#DEADBEEF
```

This sends a message with ID 127 to the chosen CAN bus, bus `can0`. This message is then picked up by the analyzer and highlighted in the rolling trace window seen in Figure 4.24. The message holds the RX trace, meaning it detected a message sent from a different node in the network.

TRACE	ID	DLC	DATA 0	DATA 1	DATA 2	DATA 3	DATA 4	DATA 5	DATA 6	DATA 7	TIME STAMP (sec)	TIME DELTA (sec)
RX	0x127	4	0xDE	0xAD	0xBE	0xEF					48.2160	47.128
TX	0x123	8	0x11	0x22	0x33	0x44	0x55	0x66	0x77	0x88	1.0880	0.000
TX	0x123	8	0x11	0x22	0x33	0x44	0x55	0x66	0x77	0x88	1.0880	0.000
TX	0x123	8	0x11	0x22	0x33	0x44	0x55	0x66	0x77	0x88	1.0880	0.000

Figure 4.24 - Analyzer Rolling Trace Window after message reception

This result means the CAN-Hat prototype successfully sent a message in the correct format to the CAN bus and could make data requests to the inverter, using the instruction set specified in the user manual [2]. This means that these two successful tests assured that the CAN-Hat could fulfill its purpose of making a Raspberry Pi into a CAN bus compatible device.

5 Results

This chapter will analyze the observer’s behavior in the full system simulation, including an inverter and scalar command. This is the closest scenario to the physical implementation that was initially intended. The results are divided into two sub-chapters. The first reiterates the results already presented in section 3.2.3, zooming in on the area with the biggest error values to further examine the observer performance. The second shows these same methods being used to analyze a scenario with a sudden load disturbance.

5.1 Constant Load

The focus will be placed on the transition zone between the V/f startup ramp and steady-state operation in this scenario, as this is where the largest discrepancies between real and estimated values are found.

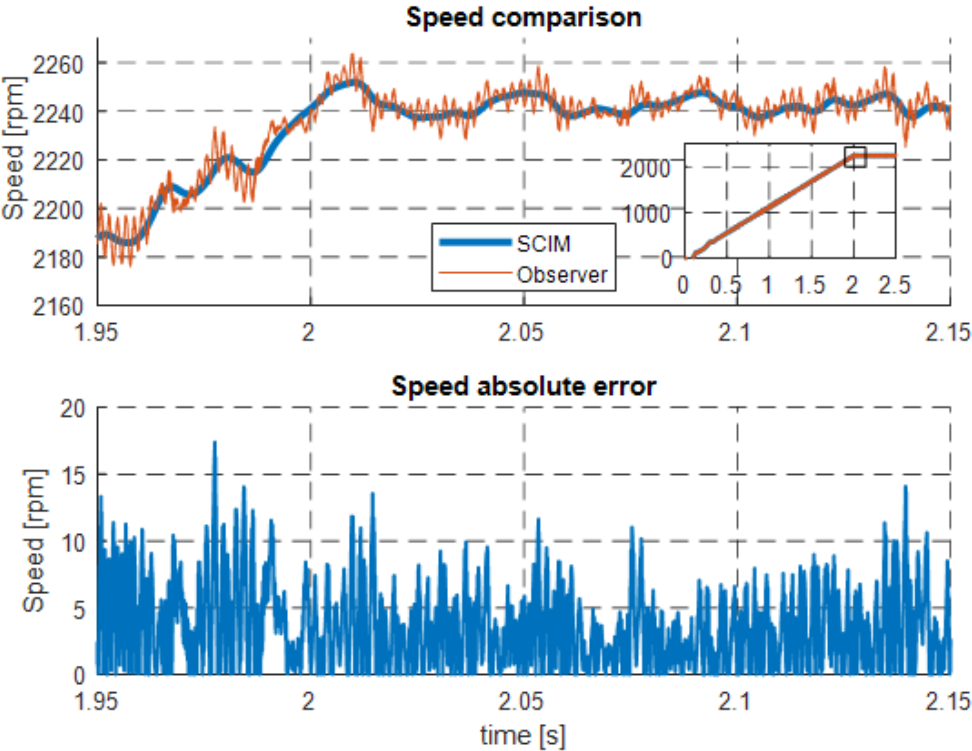


Figure 5.1 - Performance of Speed Estimation in the Transition to Steady State

Firstly, in the speed estimation found in Figure 5.1, we see that the estimated speed oscillates around the true speed value. The estimation error in this scenario never exceeds 1%, with its highest value just before the end of the transition to steady-state. Here, it reaches an absolute value of 18 rpm, which amounts to a relative error of 0.8%.

The torque estimation, seen in Figure 5.2, shows similar results. Looking at the absolute error values, the error never rises above 0.5 Nm. This corresponds to a relative error value of 0.7%.

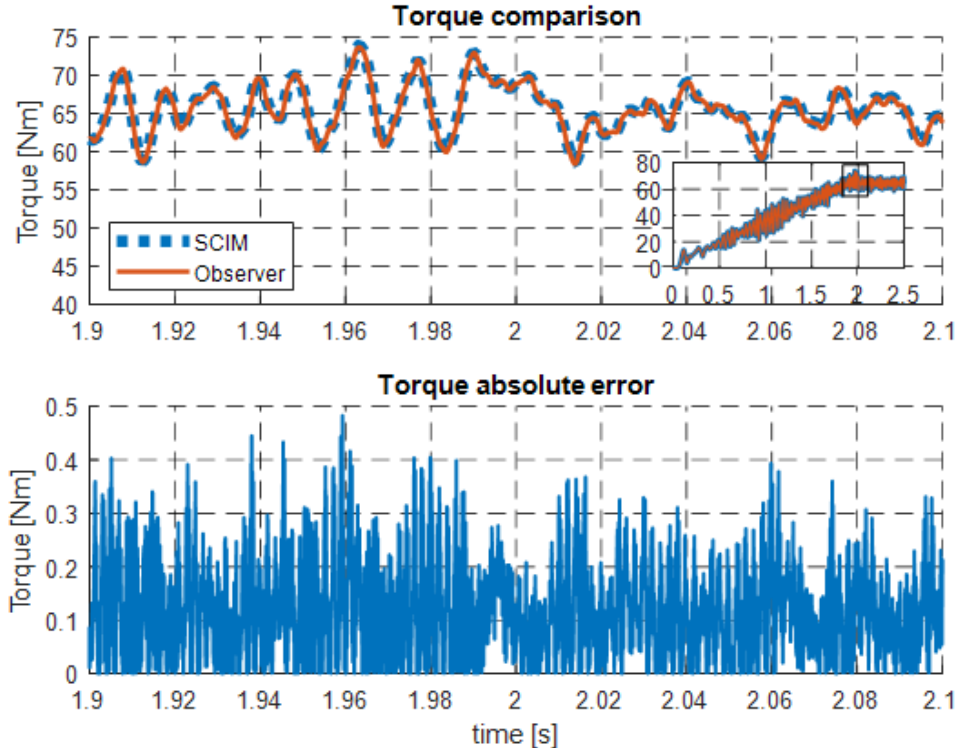


Figure 5.2 - Performance of Torque Estimation in the Transition to Steady State

The direct current component in Figure 5.3 shows some of the system’s worst results, reaching an error of 4 A, or in relative terms, about a 3% error concerning the real current value.

This is problematic, as both estimated current components are present in the computation of speed and torque values. This means that lowering the error between the direct current component’s real and estimated values would greatly improve the speed and torque estimation results.

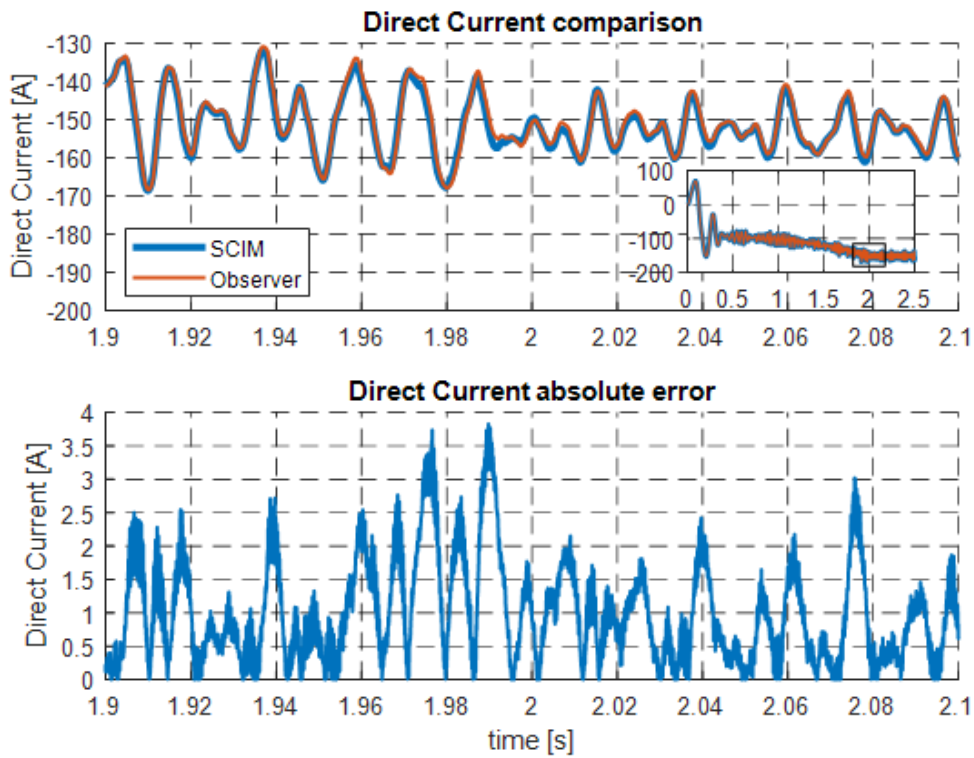


Figure 5.3 - Performance of Direct Current Estimation in the Transition to Steady State

Finally, in Figure 5.4, the quadrature current shows a maximum error of 1.5 A, which amounts to less than 1% relative error. A positive result, as the current components are part of the computation of all estimated mechanical quantities.

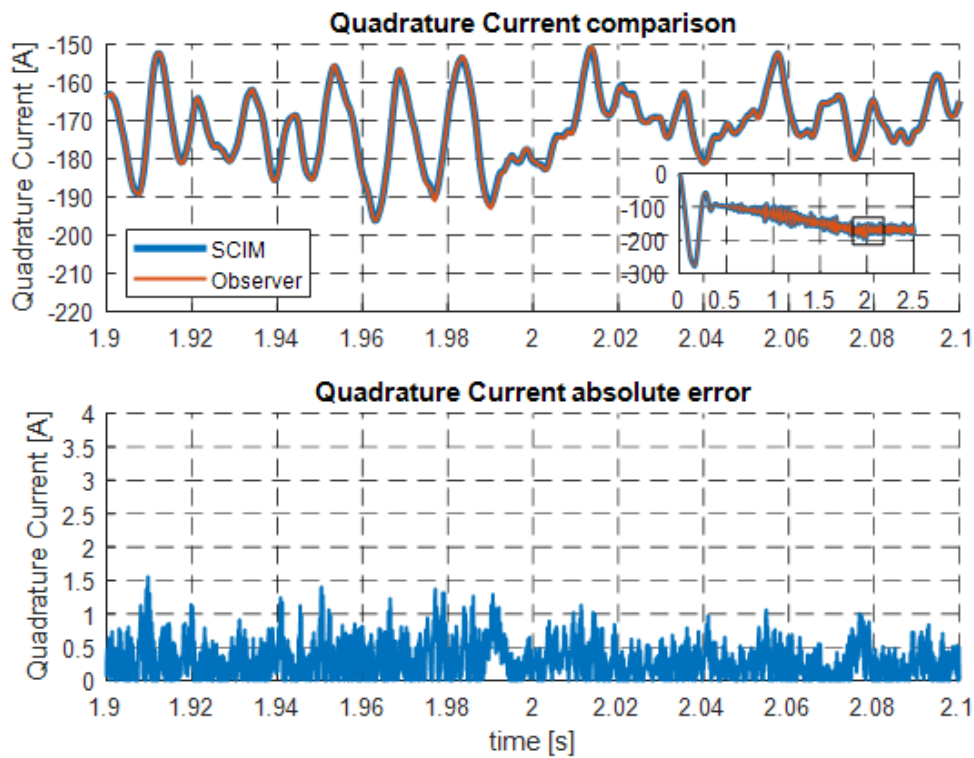


Figure 5.4 - Performance of Quadrature Current Estimation in the Transition to Steady State

5.2 Load Disturbance

This scenario adds a disturbance in the load torque in the shape of a step that reduces the load by a value of 30 Nm. This lasts for one second until the load torque returns to its original value of 65 Nm. The observer performance is then evaluated around this disturbance zone, again using absolute error values.

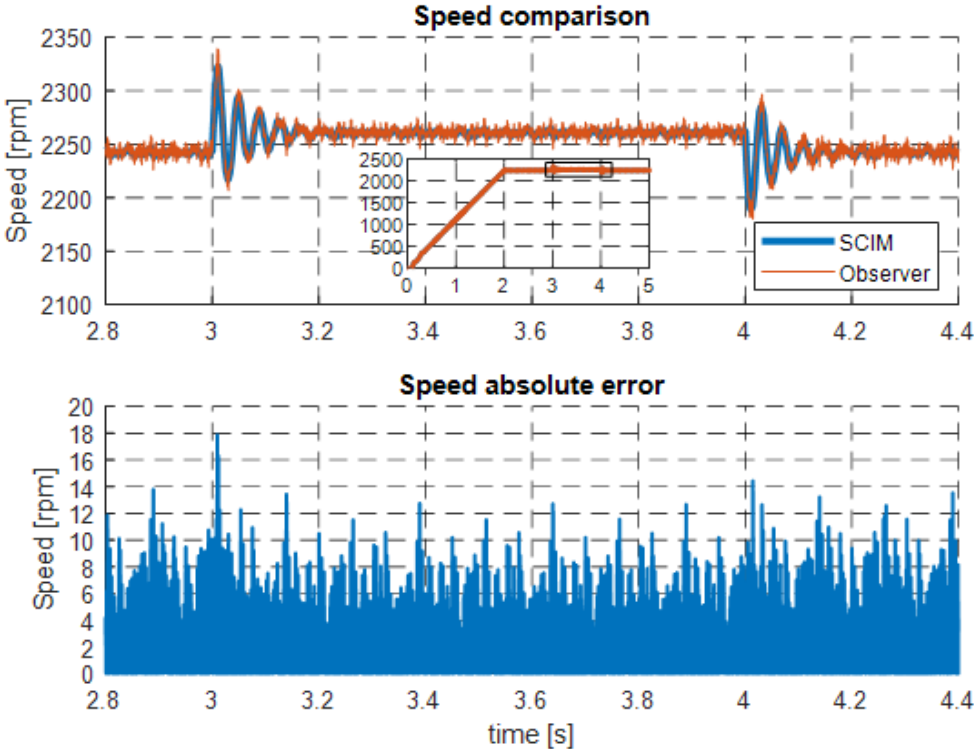


Figure 5.5 - Performance of Speed Estimation with a Load Disruption

The speed response to the disturbance is shown in Figure 5.5. Here we see that the sudden removal of about half of the load causes oscillations in the speed and that, after some time, it stabilizes at a slightly higher velocity. The opposite effect occurs at four seconds when the load is re-applied. The speed oscillates and then returns to its original value.

As for the performance of speed estimation, we see a spike at the three-second mark where the disturbance occurs. Here the error reaches 18 rpm, a 4-rpm increase over the steady state error value. The error increase for the return to full load is lower, as it reaches a total error of less than 15 rpm. An important observation is that the error spikes only occur at the disturbance's exact moment, immediately returning to the values seen in steady-state operation.

The disturbance's effect is visible in Figure 5.6, as the torque drops from 65 Nm to 35 Nm and then returns to its original value. In both cases, the error spikes amount to an increase of 0.2 Nm over the base error value of 0.4, meaning an increase of 33%.

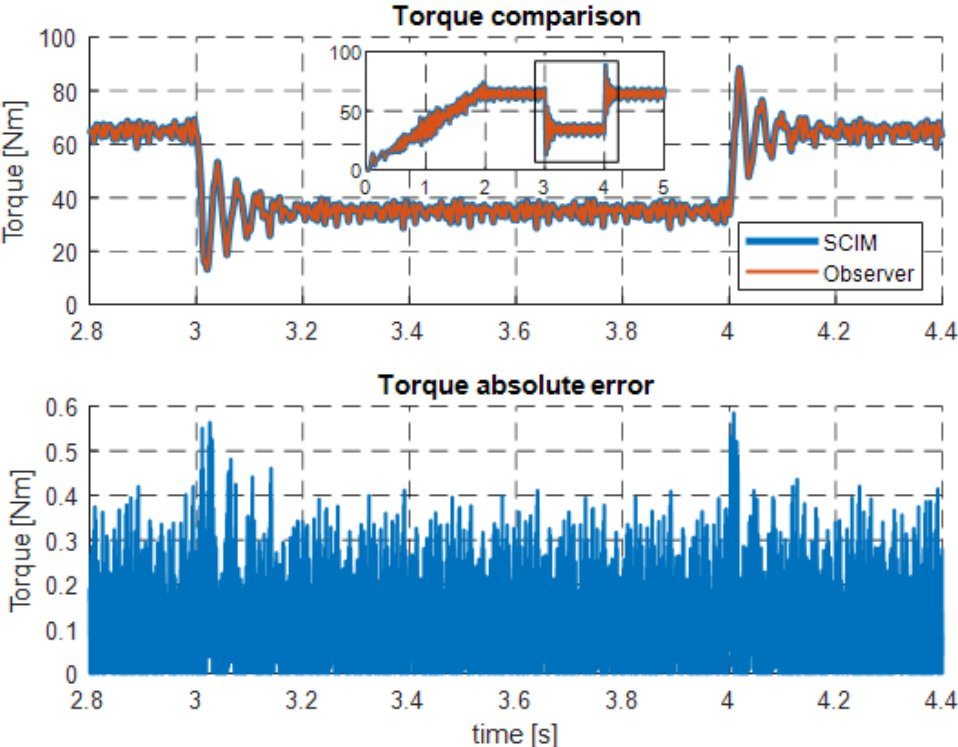


Figure 5.6 - Performance of Torque Estimation with a Load Disruption

The effects on the direct current's error values, seen in Figure 5.7, are some of the smallest. The first disturbance only increases the error by 200 mA, whereas the second worsened the error by 400 mA.

On the contrary, the quadrature component, shown in Figure 5.8, doubled its error value on the first disturbance, while the second caused an increase of 50%. This error increase explains the rise in the torque error that was previously mentioned.

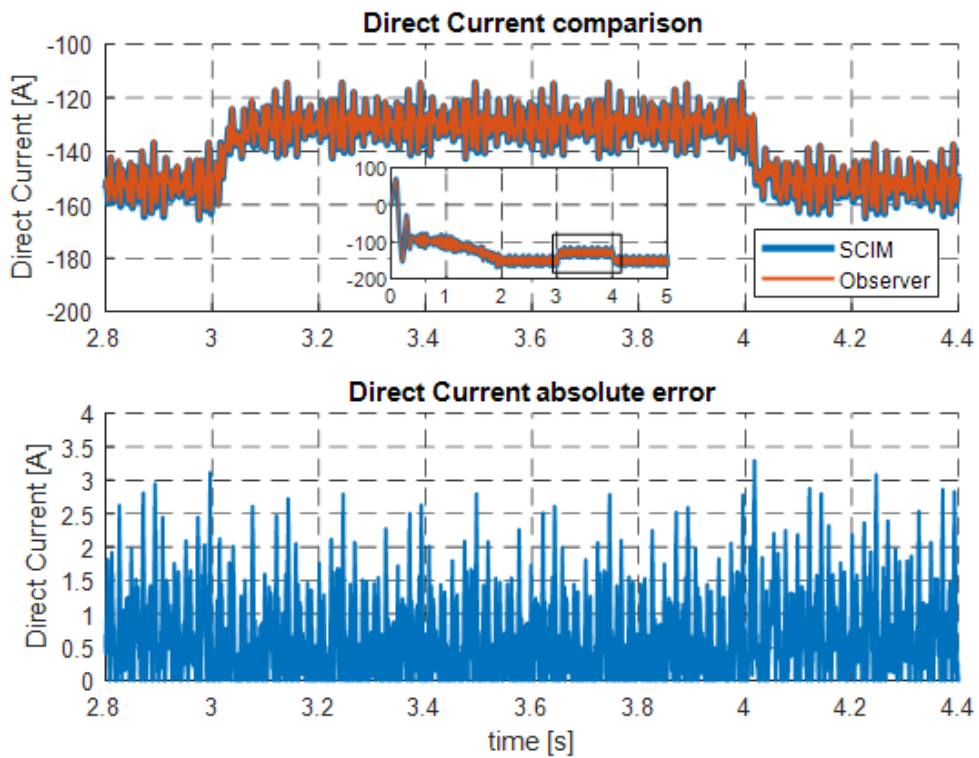


Figure 5.7 - Performance of Direct Current Estimation with a Load Disruption

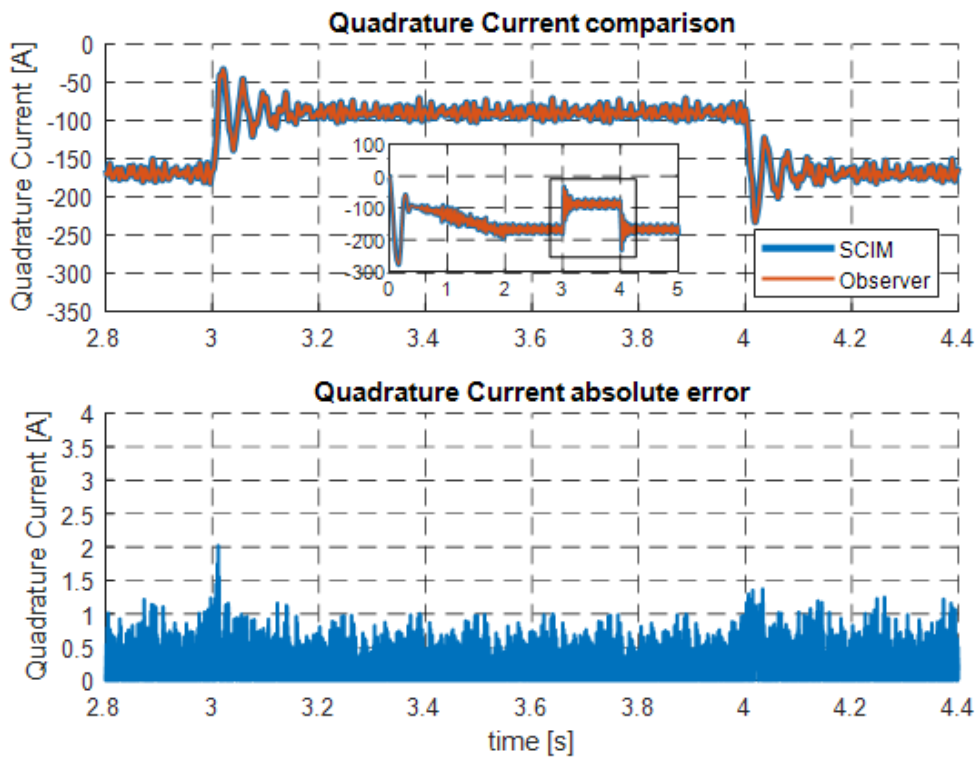


Figure 5.8 - Performance of Quadrature Current Estimation with a Load Disruption

6 Conclusions

The squirrel cage induction machine's characteristics and working principles were studied in detail, including its behavior in a dynamic system. These characteristics were then converted into a stationary $dq0$ -reference frame and used to create a full order speed observer.

The observer's behavior was then analyzed regarding its speed and torque estimation and compared to a simulated induction motor, developed from the machine's electrical and mechanical equations. This analysis consisted of several scenarios, including DOL grid connection, scalar command, and the implementation of a simulated inverter.

This led to two different electrical circuits, devised to acquire the electrical quantities feeding the VIENA's actual motor. These took two different approaches to solve this problem. The first used measuring devices to directly acquire the relevant electrical quantities and send them to an Arduino for data processing. The second design took advantage of the fact that the electrical quantities feeding the motor originate at the vehicle's inverter, meaning that this device already holds all the relevant data for the observer model. This led to developing a circuit that could connect to the inverter using some communication protocol, in this case, the CAN bus, and send this data to a Raspberry Pi, where the needed computations could occur. Both circuits were then designed as fully-fledged PCBs and sent to manufacturing. Unfortunately, only the second design's components would be purchased due to budgetary constraints, meaning only this prototype would be completed and enter the testing phase.

The next step was configuring the Raspberry Pi and familiarization with the CAN interface and its commands. What followed was a series of communication tests sending and receiving CAN messages and monitoring the results with dedicated hardware in the shape of a CAN bus analyzer. The tests were successful, meaning the prototype was ready for implementation in the VIENA with the new inverter.

Unfortunately, due to the ongoing global pandemic resulting in limited access to the laboratory, other systems that do not concern this master thesis but are an integral part of the vehicle's operation were not in working order, meaning the integration with the car and inverter, that was set as one of the objectives, was not possible. Nevertheless, the concept was shown to be promising and will surely be perused further in future works.

6.1 Future work

One aspect that could prove relevant to the improvement of observer performance is implementing an adaptive stator-resistance estimator. Since this quantity varies with temperature and significantly influences the speed estimation, the adopted approach, which consisted of measuring the stator resistance and using a constant value as the observer parameter, was not the most accurate. Such an estimator's development would be similar to that of the speed observer already described in this master thesis.

It's important to point out that, although the stator resistance significantly influences the estimated speed, the rotor resistance can be neglected, as "its influence is independent of the speed, since the speed estimation error and the rotor resistance error cannot be separated from stator variables" [7].

Some additional work would probably arise from implementing such a prototype in the VIENA as the one built. Only then could aspects such as the real sampling frequency or the delay in the inverter's answer to data requests be considered. These factors could lead to changes to the hardware design being necessary to obtain satisfactory results from the observer. One such change that could be easily done if necessary is changing the circuit's oscillator from an 8Mhz model to 16MHz, which would essentially double the CAN bus's maximum bitrate, possibly solving any bottlenecks in the data transmission.

Another interesting study could be done analyzing the differences in observer performance when using the data values transmitted from the inverter compared to direct measurement. Essentially comparing both approaches for data acquisition presented in this master thesis.

Bibliography

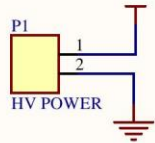
- [1] Fiat Chrysler Automobiles, "FIAT Seicento ELETTRA rev.01," 1998.
- [2] SEVCON, "Gen4 Applications Reference Manual Rev. 3.4".
- [3] A. E. Fitzgerald, C. J. Kingsley and U. S. D., *Electric Machinery*, Sixth ed., McGraw-Hill, 2003, pp. 306-356, 657-667.
- [4] R. Crowder, *Electric Drives and Electromechanical Systems*.
- [5] A. Hughes, *Electric Motors and Drives*, Third ed., Elsevier, 2006, pp. 167-304.
- [6] J. Fernandes, F. Silva and B. Tibério, "Electrical Drives and Electrical Vehicles, Project - Viena #3 - dq Model for the Induction Motor," 2019.
- [7] P. Vas, *Sensorless and Direct Torque Control*, Oxford University Press, 1998, pp. 263-574.
- [8] A. Glumineau and J. de León Morales, *Sensorless AC Electric Motor Control, Robust Advanced Design Techniques and Applications*, Springer International Publishing, 2015, pp. 1-120, 201-231.
- [9] K. Hisao, M. Kouki and N. Takayoshi, "DSP-Based Speed Adaptive Flux," *IEEE TRANSACTIONS ON INDUSTRY APPLICATIONS*, vol. 29, no. 2, pp. 344-348, March/April 1993.
- [10] J. A. Melkebeek, *Electrical Machines and Drives*, Springer International Publishing, 2018, pp. 117-144.
- [11] KOHSHIN ELECTRIC CORPORATION, "Hall Effect Current Sensor," 2013. [Online]. Available: <http://www.kohshin-ele.com/>. [Accessed 15 September 2020].
- [12] Raspberry Pi Foundation, "Raspberry Pi Documentation," [Online]. Available: <https://www.raspberrypi.org/>. [Accessed June 2020].
- [13] International Organization for Standardization, "Road vehicles — Controller area network (CAN) — Part 1: Data link layer and physical signalling," December 2015. [Online]. Available: <https://www.iso.org>.
- [14] B. Tibério, "Viena: Veículo Inteligente Elétrico de Navegação Autônoma, Documentation Guide Revision 4," 2018.
- [15] "Melexis Inspired Engineering," 2019. [Online]. Available: <https://www.melexis.com/>. [Accessed 15 September 2020].
- [16] H. Balaska, S. Ladaci, H. Schulte and A. Djouambi, "Adaptive Cruise Control System for an Electric Vehicle Using a Fractional Order Model Reference Adaptive Strategy," *IFAC PapersOnLine*, 2019.
- [17] D. Pizzo, "Le Fiat Elettra, dalla Panda alla Seicento elettrica," *AUTOMOTO ELETTRICO*, 7 May 2019. [Online]. Available: <https://www.automoto.it/>. [Accessed 19 December 2020].

Annex A - PCB Schematics

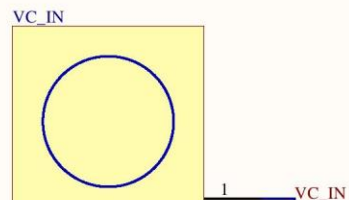
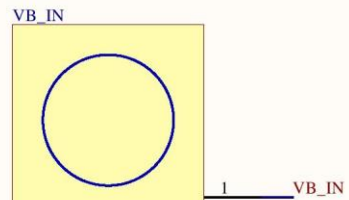
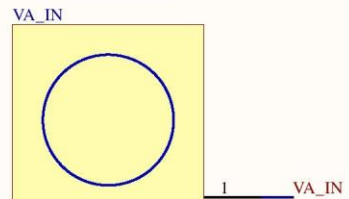
This annex contains the schematics and board layout of the designed PCB prototypes. It is divided into two subsections. The first holds the design with integrated measurement devices and the second, the CAN module for the Raspberry Pi.

External Acquisition Design

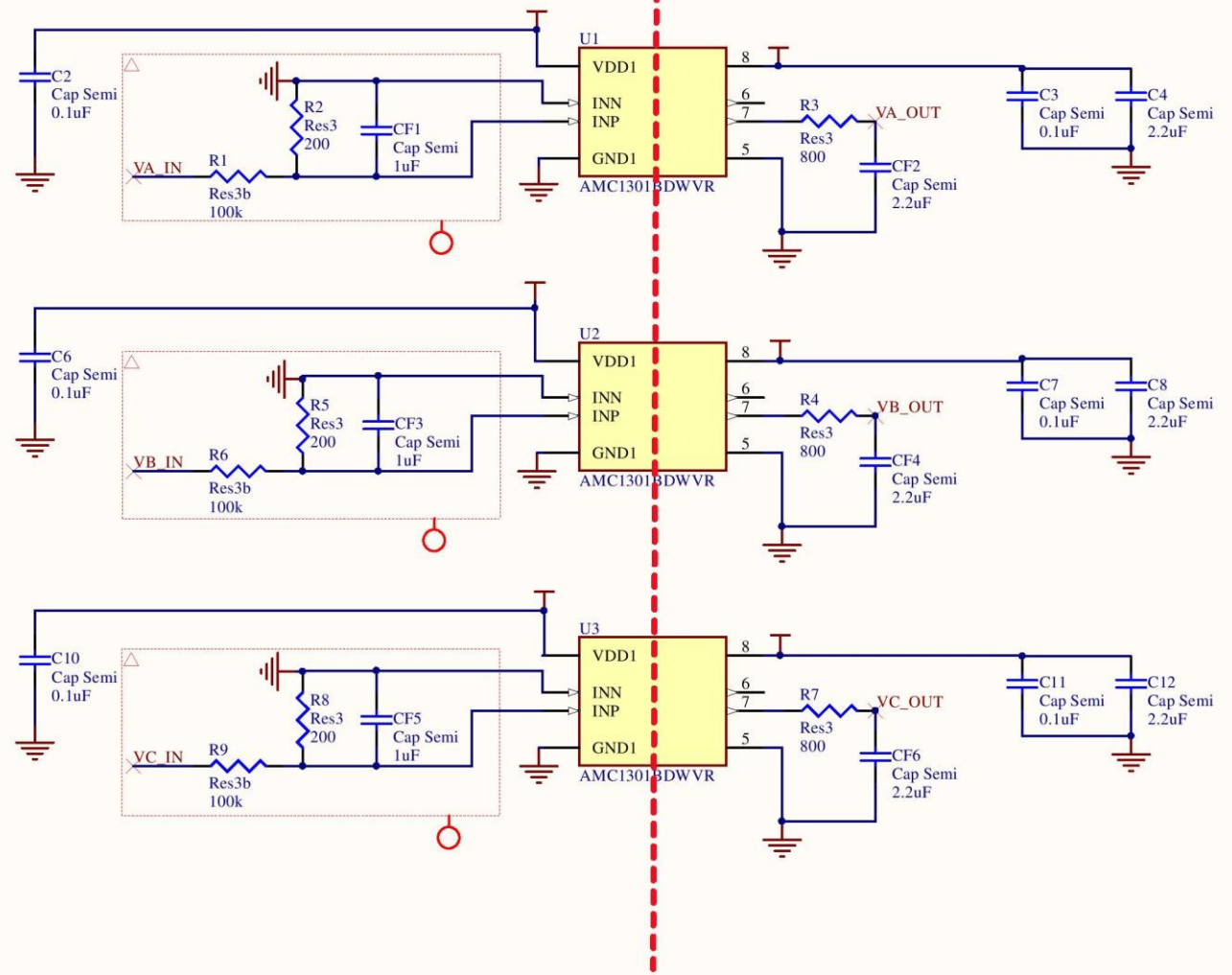
VDD1 is 5Vdc
from the Inverter



VIN is 75Vrms



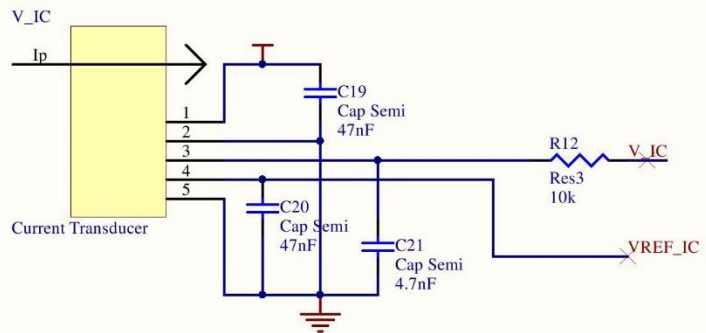
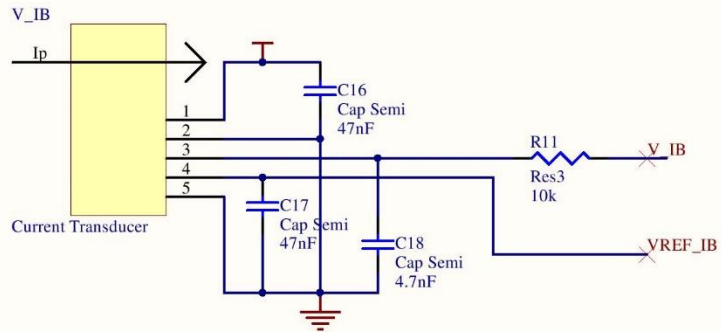
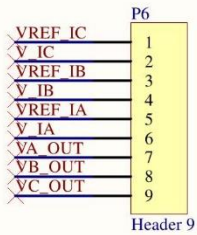
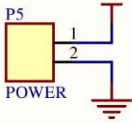
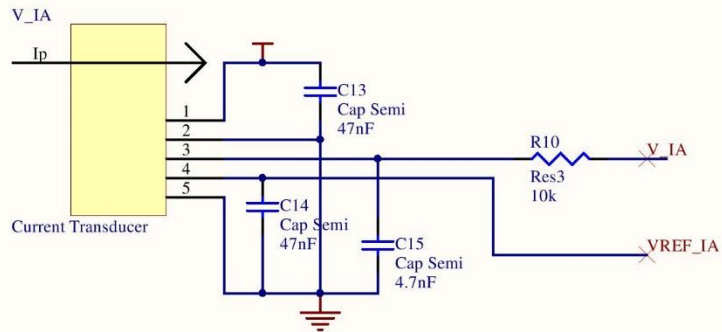
VDD2, GND2 and
VOUT to ARDUINO
VDD2 is 5Vdc



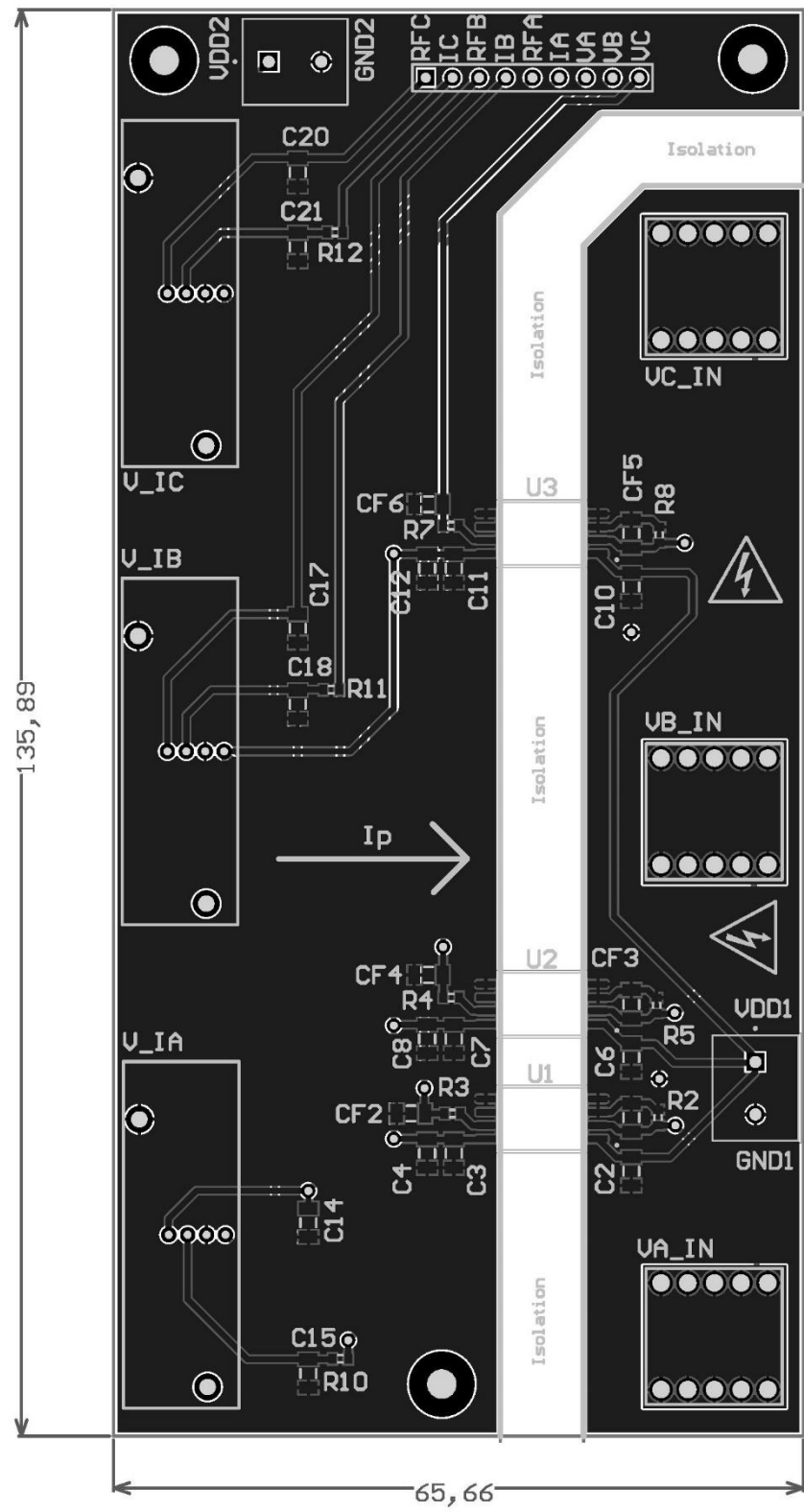
Title		
Size	Number	Revision
A4		
Date:	21/12/2020	Sheet of
File:	D:\Tese\.\Volt_acquisition.SchDoc	Drawn By:

VDD2 is 5Vdc
 VREF is 2.5V Voltage Reference

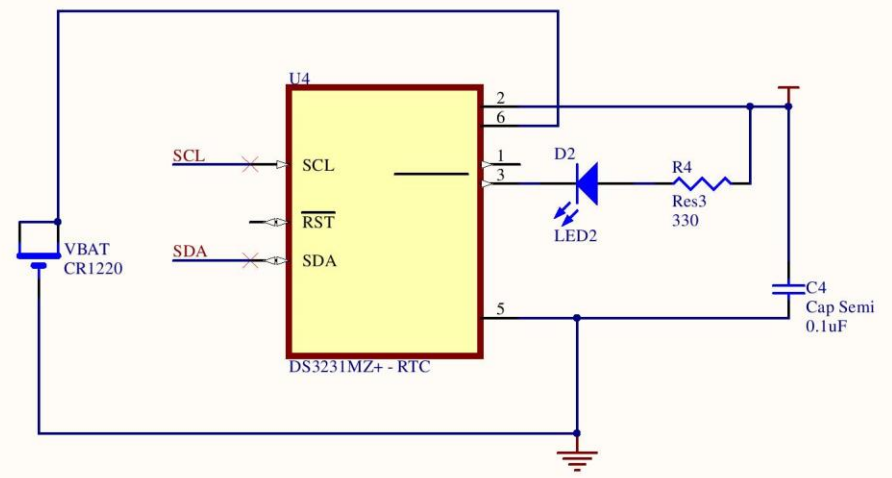
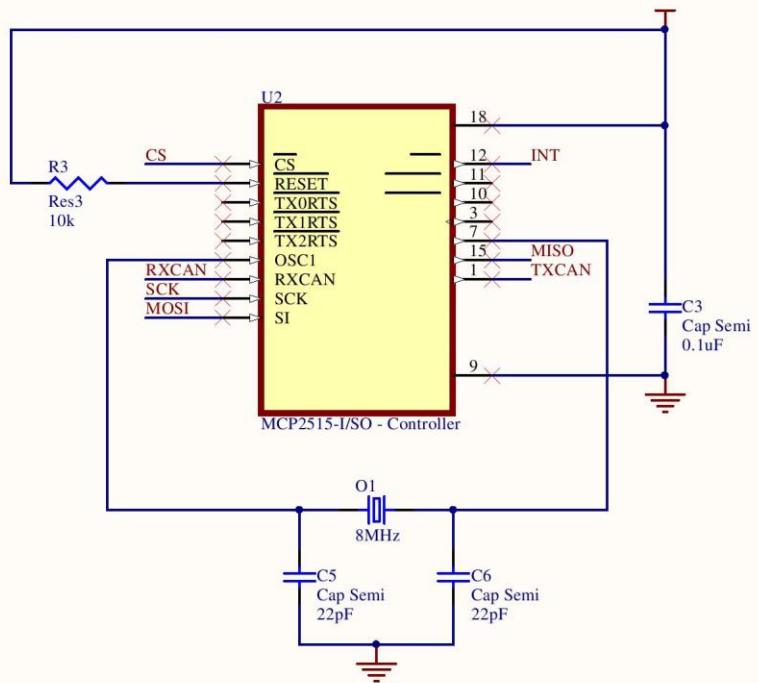
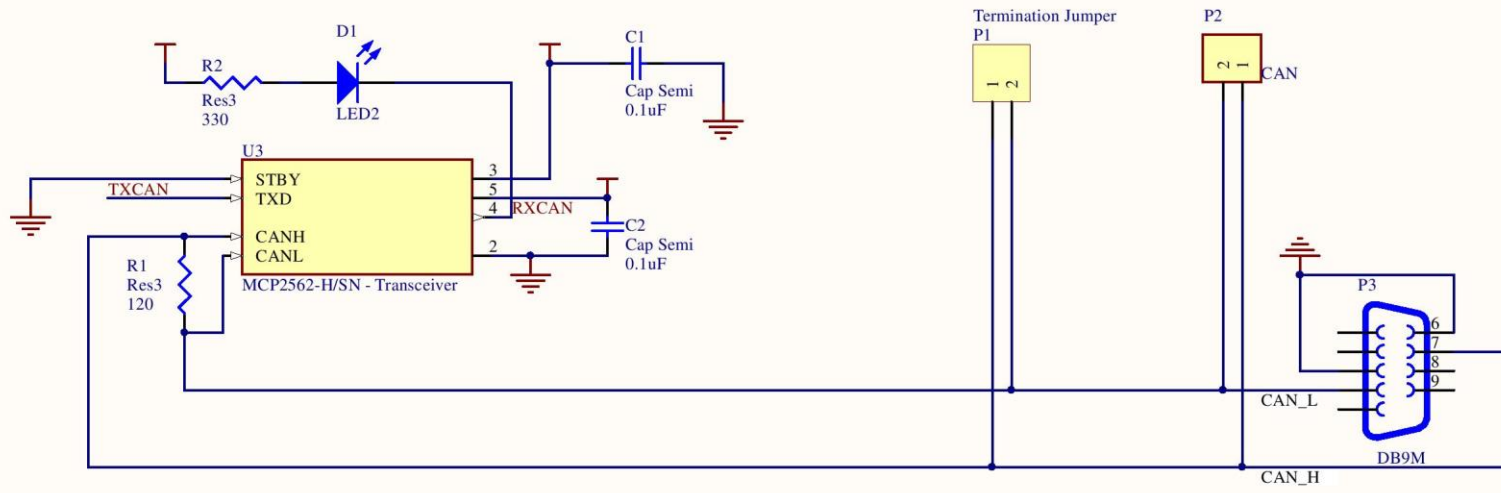
Headers hold all Voltage and Current data



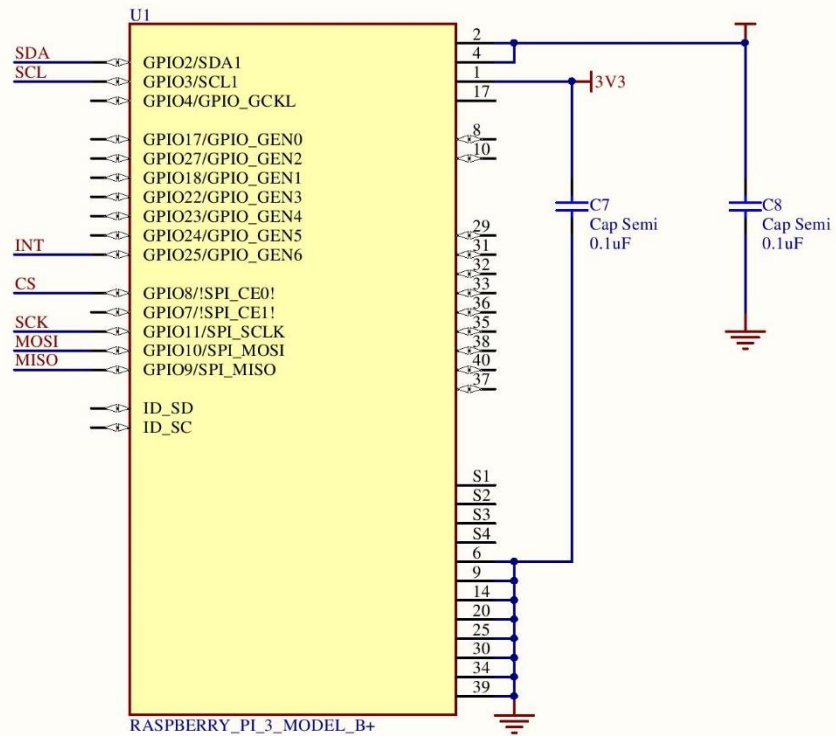
Title		
Size A4	Number	Revision
Date:	21/12/2020	Sheet of
File:	D:\Tese\.\Curr_acquisition.SchDoc	Drawn By:



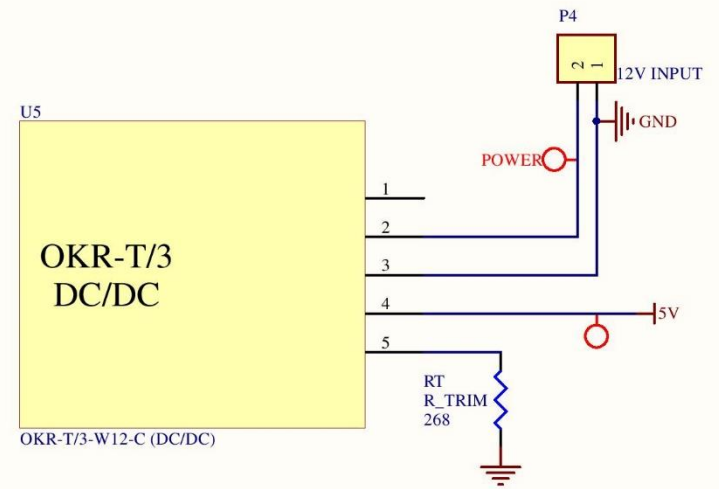
CAN-HAT Design



Title		
Size A4	Number	Revision
Date: 21/12/2020	Sheet of	
File: C:\Users\A\CAN_Shield_Rpi_4_B.SchDoc	Drawn By:	



Step down converter
(12V to 5V) to power
Rpi from existing battery



Title		
Size A4	Number	Revision
Date: 21/12/2020	Sheet of	
File: C:\Users\...Rpi_3_B_conect.SchDoc	Drawn By:	

