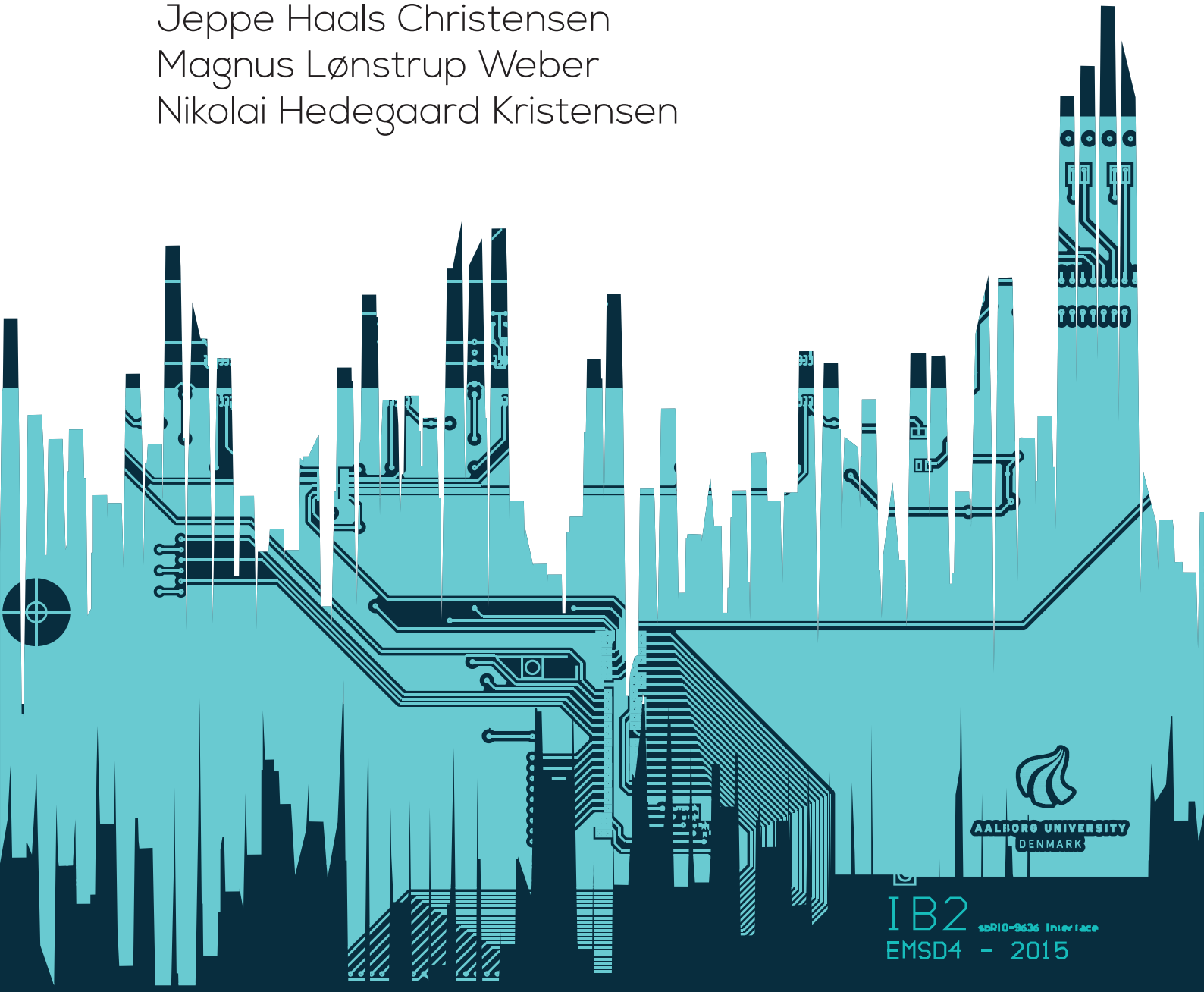
DEVELOPMENT OF AN AUTOMATIC PARAMETER IDENTIFICATION METHOD FOR PMSM DRIVES WITH AN LC-FILTER

Electro-Mechanical System Design
Master's Thesis

Jeppe Haals Christensen
Magnus Lønstrup Weber
Nikolai Hedegaard Kristensen



Study Board of Industry and Global Business Development

Fibigerstraede 16
DK - 9220 Aalborg East
Phone +45 99 40 93 09
lft@m-tech.aau.dk
www.ses.aau.dk

AALBORG UNIVERSITET
STUDENTERRAPPORT

Titel: Development of an Automatic Parameter Identification Method for PMSM Drives with an LC-Filter
Semester: Master Thesis
Project period: 02/02-2015-03/06-2015
ECTS: 30
Supervisor: Michael Møller Bech

Jeppe Haals Christensen

Magnus Lønstrup Weber

Nikolai Hedegaard Kristensen

Synopsis:

This project aims to overcome the difficulties of determining the parameters of a drive system with an inverter, an LC-filter and a motor.

A laboratory setup is established with an inverter, an LC-filter and a permanent magnet synchronous motor. The laboratory setup is investigated and computer models are developed for simulation.

An automatic parameter identification method is designed for a stand alone solution. The method contains system excitation with pulse width modulation, based on a sequence of random binary numbers, frequency response estimation from spectral analysis, and curve fitting with an optimization algorithm.

Custom hardware is design for the purpose of this project. The hardware is designed for control of an inverter and sampling of signals where high precision, high bandwidth, and noise mitigation is key. The designed hardware is manufactured in the laboratory.

The parameter identification method is tested with the simulated system to evaluate the method's performance. Finally the method is used to determine the parameters of the laboratory setup.

The parameter identification method is concluded successful in identifying the parameters of the laboratory setup.

Printed copies: 5 pcs.

Pages: 156

Preface

This report is the result of a Master's Thesis project in the spring of 2015, studying *Electro-Mechanical System Design* at Aalborg University, Denmark.

This report is mainly addressed towards the involved supervisors, peer students or other individuals of the same background knowledge. This report is the second part of a 2-semester project in the field of automatic parameter identification. In the previous project knowledge were gained in the field of parameter identification. In this project focus is on developing an automatic parameter identification method for a laboratory setup, including hardware design and manufacturing.

Attached to the report is an annex CD containing MATLAB scripts, the LabVIEW project folder, Altium Designer circuits and PCB drawings, a copy of the report and other information of interest for the reader. The contents of the annex CD can be found on the last page of the report.

Equations, figures and tables are numbered in accordance with the chapter they appear in. The same goes for references to chapters and sections, followed by the name of the chapter or section. For example: section 3.1 *"Parameter Identification Strategy"*, figure 4.26 and equation (3.10). A full Bibliography is located at the end of this report. Citations contain the last name of the author and the year it was published. For company published manuals, data sheet and applications notes, a brief description of the reference topic follows the author name, for example: (Infineon IGBT, 2014).

Danish Abstract

I dette projekt fokuseres der på, at udvikle en metode til parameter identifikation af et system med en inverter der styrer en elektrisk motor, hvor der er monteret et LC-filter mellem inverter og motor. En inverter styrer en motor ved pulsbreddemodulation, hvilket grundlæggende fremkommer af, at en række kontakter, mellem en DC forsyning og motoren, slukker og tænder systematisk. I systemer der styres med pulsbreddemodulation, kan der opstå høje spændingsgradienter, hvilket kan føre til uønskede egenskaber. Eksempler på uønskede egenskaber er akustisk støj og beskadigelse af motoren. For at reducere de uønskede egenskaber, implementeres et LC-filter mellem inverter og motor. Normalvis estimeres et systems parametre for effektivt at kunne styre motoren, men når et LC-filter implementeres i systemet så øges kompleksiteten af systemet, hvorved identificering af systemets parametre ikke kan ske ved traditionelle metoder. Der ønskes derfor at udvikle en metode til parameteridentifikation, der kan tilsluttes et eksisterende system med LC-filter og derved identifiere systemets parameter.

Et testsystem har været opstillet i laboratoriet i forbindelse med projektet, bestående af inverter, LC-filter og motor. Systemets komponenter er undersøgt og modeller af systemet er opstillet til simulering på PC. Det simulerede system benyttes til udvikling af en metode til parameter identifikation af systemet, som eftervises på laboratorieopstillingen.

Der udvikles en metode til parameteridentifikation, inspireret af artikler på området. Metoden til parameteridentifikation tager udgangspunkt i at excitere systemet med et pulsbreddemoduleret signal, der er baseret på en tilfældig sekvens af binære tal. Systemets spændinger og strømme måles, med egetudviklet hardware, og bruges til, at estimere systemet udfra spektralanalyse. Et analytisk udtryk for systemet opstilles og en optimeringsalgoritme benyttes til, at tilpasse det analytiske udtryk til det estimerede system, hvorved systemets parametre identificeres.

I forbindelse med excitering og måling af systemet, har der været udviklet og fremstillet hardware specifikt til formålet. Ved design af hardware har der været fokus på præcise målinger med høj båndbredde, samt frasortering af støjgradienter. Det designede hardware har været fremstillet i laboratoriet, på Aalborg Universitet, og er fremstillet af projektdeltagerne.

Metoden til parameteridentifikation afprøves på det simulerede system, hvor metodens præstation evalueres, før metoden implementeres på laboratorieopstillingen. Ved implementering i laboratoriet benyttes det egetudviklede hardware, sammen med en indlejret enhed fra National Instruments, til afvikling af det tilhørende software. Det tilhørende software er udviklet til, at kunne fungere, som en stand alone løsning, hvor programmet kan initialiseres og resultater aflæses uden brug af eksternt udstyr.

Afslutningsvis undersøges der, hvilke indstillinger af metoden der medfører en effektiv og præcis identifikation af laboratorieopstillings parametre. Der undersøges hvilken indflydelse forskellige indstillinger har på eksekveringstid og resultat af optimeringsalgoritmen.

Contents

Chapter 1

Introduction

Today, electrical machines are used in a variety of applications. Examples are: the industry, home appliances, vehicles, and medical equipment. The simplicity and high efficiency, exceeding 95% for large synchronous machines, have lead to high adaption of electrical machines in the industry (Mohan et al., 1989). Electric motors, pumps, and fans are some of the most used electrical machines in the industry. In manufacturing industries, 10%-25% of the energy consumption is used in electrical machine drives, and 89% of all motors in the manufacturing industry are electrical (Otis, 2013).

The choice of machine and characteristics vary depending on the application. For fixed speed applications, like fans, pumps, grinders, drills, and hoists, simple induction machines are used due to their simplicity and robustness. For variable speed applications, like automation, robotics, milling, and automotives, the same induction machines can be used with a variable speed drive. Also the similar Permanent Magnet Synchronous Machines (PMSM) are used for variable speed drives, because of its high efficiency and high power density.

In industrial applications three phase electric machines are often used for variable speed drives. These three phase machines are fed by three phase inverters and often controlled by a Digital Signal Processor (DSP). The speed of the machine is controlled by modulating sinusoidal voltages with Pulse Width Modulation (PWM), where changing the frequency of the modulated sinusoidal voltages, will change the speed of the motor. The variable speed drive is illustrated in figure 1.1 where an inverter modulates the voltages, U, V, and W, that drive the motor.

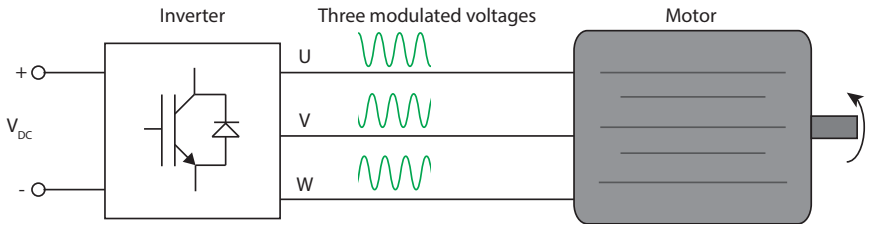


Figure 1.1: Three phase variable speed drive.

1.1 Problems

This project will address problems associated with controlling electrical machines with fast swtiching inverters, therefore the general switching scheme is described together with some of the problems that occur.

An inverter is used to control a motor, by switching the transistors in the inverter in a controlled manner. Figure 1.2 illustrates a three phase inverter where six Insulated-Gate Bipolar Transistors (IGBT) are controlled by a Digital Signal Processor (DSP). A transistor has two stages, either it conducts or it does not. The DSP controls the IGBTs such that the voltage for each phase, U, V, and W, is a square waveform which is the PWM of a reference voltage, as illustrated in figure 1.2.

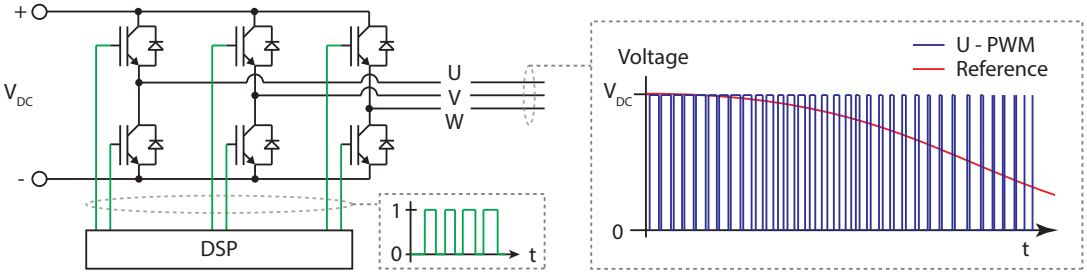


Figure 1.2: Three phase inverter used to pulse width modulate a reference voltage.

The IGBTs are switching with frequencies up to above 20[kHz], which leads to some undesirable behaviors of the system. The high $\frac{dV}{dt}$, that is needed due to the fast switching at each phase, U, V, and W, leads to induced bearing currents in the motor, potentially damaging the bearings and reducing the lifetime of the machine.

In the case when the motor and the inverter is distanced from each other, the high $\frac{dV}{dt}$ may cause voltage oscillations at the motor terminals due to the long cables between the inverter and the motor.

In the following, bearing currents and what happens when long cables are used, are discussed, along with some considerations about acoustic noise in an inverter and motor setup and a solution is proposed to reduce the unwanted behaviors.

1.1.1 Bearing currents

Bearing currents might cause electro-erosion, due to the currents running through the bearings and melting the surface of the runway. A failure analysis of an electric motor, determined the dominant failure mode to be due to electric-erosion of the raceway, with currents > 0.5[A]. It was assessed that a current of 0.01[A] would reduce the lifetime of the bearing with up to 20% (Laboratory Soete, 2004).

Two type of bearring currents exists: non-circulating bearing current and circulating bearing current, which are described further below.

Non-circulating bearing current
The path of the non-circulating bearing current I_{brg} is shown in figure 1.3 for phase U, with the voltage V_U switching between the negative DC bus O and the positive DC. It is shown how the motor frame and the stator are connected to ground, together with the inverter. The ground connection and the inverter DC bus negative O are separated by Z_{in}, which represents the internal impedance of the inverter, which usually consists of parasitic capacitances. C_{wr} and C_{ws} are the parasitic capacitances between winding-rotor and winding-stator, respectively, while C_G is the total capacitance across the bearing (Chen and Lipo, 1998). In figure 1.3 the concept is only illustrated for phase U but the same is true for V and W.

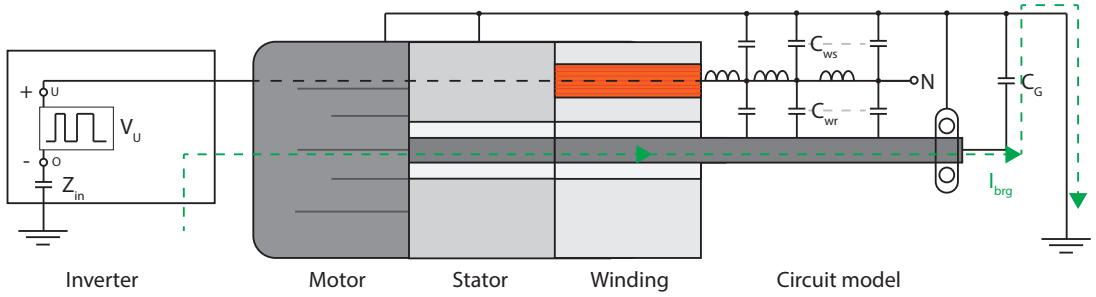
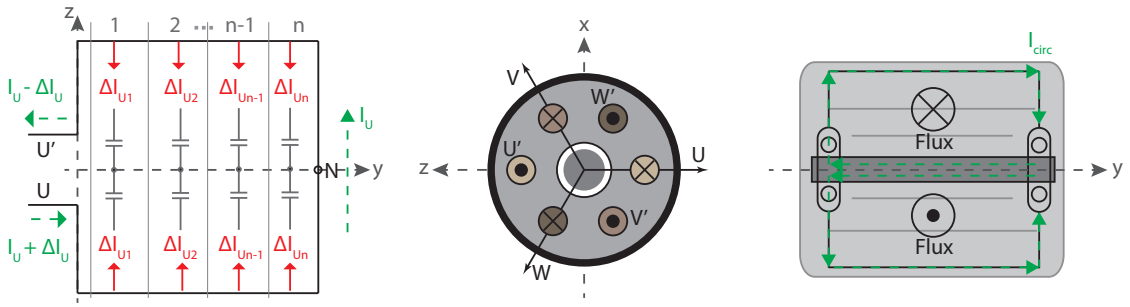


Figure 1.3: Non-circulating bearing current illustrated with equivalent circuit of parasitic capacitance.

Current accumulates in the parasitic capacitance between the windings and the rotor C_{wr}, which discharges through the bearings into the ground-connected motor casing. The conductivity of the bearing might not be continuous, why charge accumulates on C_G until the bearing conducts and produce a current spike, known as discharge mode bearing current. If the bearing is always conducting, the accumulated current will discarge immediately through the bearing and will relate to $\frac{dV_{UO}}{dt}$, known as conduction mode bearing current (Chen and Lipo, 1998).

Circulating bearing current
Figure 1.4 is used to describe the circulating bearing current. Figure 1.4.a shows a simplified winding of phase U. The phase winding is considered as a single coil winding while considering the current flow in this. The stray capacitances C_{wr} from figure 1.3 are assumed uniformly distributed along the y-direction (Chen and Lipo, 1998).



a) Current flow in winding U b) Cross-section of 3-phase motor c) Circulating bearing current flow

Figure 1.4: Description of the circulating bearing current, from the current in the windings to the overall view of the motor. a) is a simplified winding with current flow illustrated. b) is a cross-sectional view of a simplified 3-phase motor. c) shows the circulating bearing current and the pulsating flux field in combination in a simplified motor overview.

The winding is split into smaller parts along the y-direction. For each part a fraction of the current in the winding is running into the stray capacitance, which is illustrated in figure 1.4.a as ΔI_{Ui} , $i = 1,2,..,n$. The current flow in and out of the winding can be described by $I_U + \Delta I_U$ and $I_U - \Delta I_U$, respectively, where ΔI_U is given in equation (1.1) for any cross-section q. Figure 1.4.b shows any cross section q of a simplified 3-phase motor. The input and output currents of the phases V and W are similar to that of phase U, why the total current flow in any cross section is given in equation (1.2). It is clear that for any cross section of the motor $\sum I \neq 0$ why it is concluded that there must be flux linkage around the rotor.

$$\Delta I_U = \sum_{i=q}^{n} I_{Ui} \tag{1.1}$$

$$\begin{aligned} \sum I_q &= [(I_U + \Delta I_U) - (I_U - \Delta I_U)] \\ &+ [(I_V + \Delta I_V) - (I_V - \Delta I_V)] \\ &+ [(I_W + \Delta I_W) - (I_W - \Delta I_W)] \\ &= 2(\Delta I_U + \Delta I_V + \Delta I_W) \end{aligned} \tag{1.2}$$

Figure 1.4.c shows the flux in the motor together with the corresponding bearing currents I_b, which circulates in the motor, from the rotor, through the bearings into the stator, and back through the second bearing to the rotor. The common mode currents are pulses, why the varying flux field induces the circulating bearing current as described above (Chen and Lipo, 1998).

Empirical results

A study (Muetze and Binder, 2007) considered the circulating bearing currents in six different induction motors, all driven by an inverter with fast switching. The motors range from 11 - 500[kW] in power and with a motor frame size from 160 - 400[mm], which is the height from the rotor to the bottom outside of the motor. The induced voltage in the rotor, V_r, is shown to be propertional with the motor frame size, h, cubed

and the square root of the frequency of the common mode voltage, f, as shown in equation (1.3). The experimental results showed the ratio between the common mode current and the circulating bearing current to be within $\frac{I_b}{I_{com}}$ = 0.2 - 0.3, and the theoretical maximum were found to be 0.354 (Muetze and Binder, 2007).

$$\begin{aligned} V_r &\propto h^3 \\ V_r &\propto \sqrt{f} \end{aligned} \tag{1.3}$$

1.1.2 Long cables

Due to parasitic capacitances, long cables between the inverter and the motor are subject to a phenomena called ringing. Ringing is oscillations in the voltage and occurs when voltage switching, from the inverter, is reflected at the motor terminals, backward to the inverter. Ringing can lead to overvoltage at the motor terminals and potentially damage the motor.

When the inverter switches, the cable experience a high $\frac{dV}{dt}$ and because of the parasitic capacitances along the cable, the voltage wave travels through the cable like a shockwave in a pipe. The wave will travel with a speed u_V approximate to half the speed of light, which is $\approx 150\frac{m}{\mu s}$, or as described in equation (1.4), where $L_{\Delta x}$ and $C_{\Delta x}$ are the parasitic inductance and capacitance of an incremental length Δx of the cable, illustrated in figure 1.5.a. The stray capacitances of a cable is illustrated in figure 1.5.b, where it can be seen that the three conductors can be considered either as a delta- or a star-coupling with a stray capacitance C_c between each conductor. For cables with sheath there is also a capacitance between this and the conductor, denoted as C_s. The effective capacitance for each conductor is $C = 3C_c + C_s$ (von Jouanne et al., 1995).

$$u_V = \frac{\Delta x}{\sqrt{L_{\Delta x} C_{\Delta x}}} \tag{1.4}$$

$$\Gamma_L = \frac{R_L - R_c}{R_L + R_c}\ , \quad \Gamma_S = \frac{R_S - R_c}{R_S + R_c}\ , \quad R_c = \sqrt{\frac{L}{C}} \tag{1.5}$$

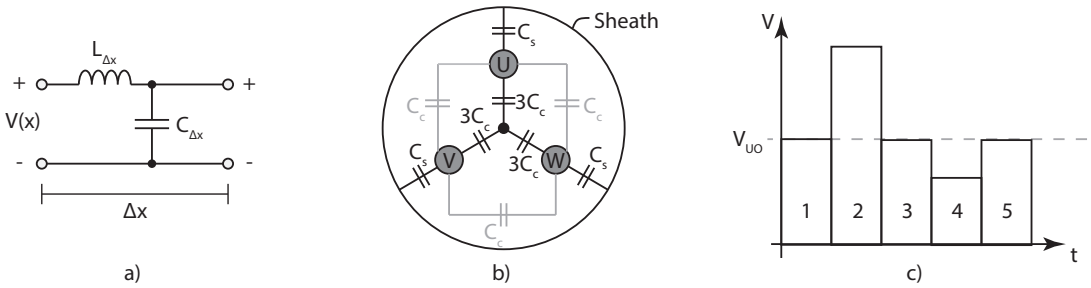


Figure 1.5: a) Equivalent circuit of an incremental length of a cable, b) Illustration of the stray capacitances in a cable with sheath, c) Voltage magnitude sketch at motor terminal, when a wave have passed.

When the wave travels through the cable and reaches the motor terminal it reflects like if it hits a wall. The voltage magnitude of the reflected wave is determined by multiplying the initial voltage with the voltage reflection coefficient Γ_L, which is shown in equation (1.5) together with Γ_S, where R_c is the characteristic impedance of the cable, while R_L is the load impedance and R_S the source impedance. Γ_s is the voltage reflection coefficient on the inverter side of the cable, which reflects the wave back towards the motors. Table 1.1 shows the wave magnitude and voltage at a point on the cable. Figure 1.5.c shows a sketch of the change in voltage, where $\Gamma_s = -1$ which happens when R_s is small and it is clear that the voltage reflection cause a damped oscillation as illustrated in figure 1.6. When the pulse takes longer than half the rise time to travel through the cable, the amplitude can double that of the DC link (von Jouanne et al., 1995).

Voltage magnitude of travelling wave:

Case:	1	2	3	4	5
V_{wave}	0	$V_{UO}\Gamma_L$	$V_{UO}\Gamma_L\Gamma_S$	$V_{UO}\Gamma_L\Gamma_S\Gamma_L$	$V_{UO}\Gamma_L\Gamma_S\Gamma_L\Gamma_S$
V_{line}	V_{UO}	$V_{UO}+V_{UO}\Gamma_L$	$V_{UO}+V_{UO}\Gamma_L+V_{UO}\Gamma_L\Gamma_S$	etc.	...

Table 1.1: Voltage magnitude of the travelling wave, and the line at some point on the cable. The illustration of V_{line} for $\Gamma_S = -1$ is shown in Figure 1.5.c for the 5 cases.

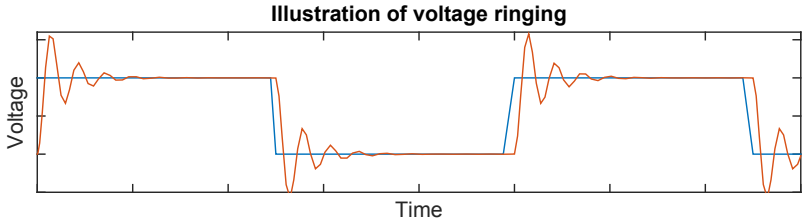


Figure 1.6: Illustration of voltage ringing at the motor terminal (red), together with the ideal switching voltage (blue).

1.1.3 Acoustic noise

Acoustic noise is a combination of different things, such as imbalance, damaged bearings, and the current and voltage waveforms in the machine. Acoustic noise from imbalance and damaged bearings is considered a mechanical design problem and is not within the scope of this project, however, the acoustic noise from the current and voltage waveforms are of interest. The voltage and current waveforms that gets the motor running is also the cause to acoustic noise in the motor, why this is described in the following.

The human ear is sensitive to frequencies up to 20[kHz] but the range 1-5[kHz] are the most sensitive (Mathe, 2010). The frequency of acoustic noise must be considered to

be below 20[kHz] but the noise in the range 1-5[kHz] may be the most noticeable as noise to the human.

The vibrations that cause acoustic noise in an induction machine can be divided into two categories; radial vibrations and tangential vibrations, relative to the rotor axis. The vibrations are due to the pulsating electromagnetic forces introduced by the stator windings. The radial force component is a biproduct of the energy transfer, from electricity to torque, and it does not contribute to the torque of the rotor, but it introduces radial vibrations in the stator frame (Sutthiphornsombat et al., 2010). The tangential force is directly coupled with the generated torque of the rotor and since the magnetic force is directly proportional to the current running in the wire, the tangential vibrations are linked to the current pulsations in the windings (Sutthiphornsombat et al., 2010).

The acoustic spectrum of the noise from the current pulsations will be centered around the switching frequency and two times the switching frequency (Lo et al., 2000). With a switching frequency of 4[kHz] the acoustic noise will be well within the frequency range of the human ear, with frequencies at 4[kHz] and 8[kHz].

In order to reduce the acoustic noise of the motor, the harmonics of the switching must be smoothed out before it reaches the motor terminals.

1.2 Solutions

It is possible the reduce some of the negative behaviors of the drive system by several measures, some of these are described below. A single solution that reduces several of the mentioned problems is discussed.

Bearing current:
There are several methods to reduce the bearing currents, examples are listed (Akagi and Tamura, 2006).

- Install a brush that connects the rotor to the ground, allowing the current to run through the brush instead of the bearing
- Use ceramic bearings or sufficiently insulate the bearings from the stator
- Install passive filters to reduce common mode current and $\frac{dV}{dt}$

The first two methods require modifications to the mechanical design of the motor, why these are not considered further. The last solution with adding passive filters is described further in the following and the filter structure is illustrated in figure 1.7.

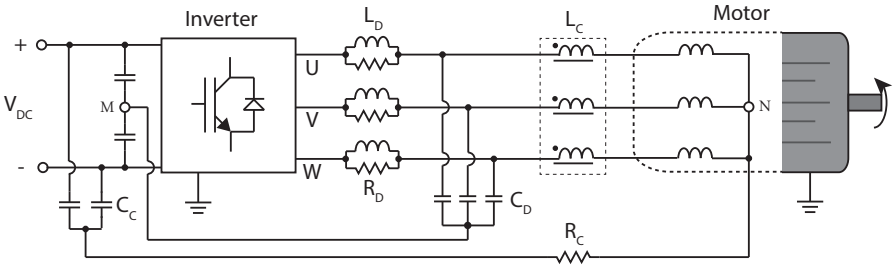


Figure 1.7: Common mode filter and $\frac{dV}{dt}$ filter.

The bearing currents are in general reduced by reducing either the $\frac{dV}{dt}$ or by reducing the common mode current.

In order to reduce the common mode current, a filter is installed that connects the neutral point, N, of the motor windings with the DC-midpoint of the two filter capacitors, C_C. The common mode filter with the components, R_C, L_C, C_C, will remove the bearing currents that relate to the common mode current, while a filter with the L_C only will reduce the bearing currents with more than 50% (Akagi and Tamura, 2006).

The $\frac{dV}{dt}$ filter with the components, R_D, L_D, C_D, acts as a low-pass filter that reduces the $\frac{dV}{dt}$ of the inverter switching - reducing the bearing current. The connection between the DC-midpoint M and the common connection between C_D does not reduce the $\frac{dV}{dt}$ but it reduces the common mode current (Akagi and Tamura, 2006).

In order to totally remove the bearing current both the filters must be applied, as illustrated in figure 1.7. It is not always possible to access N or M, in which cases it is only possible to reduce the bearing currents.

Ringing:
In order to reduce the ringing effects at the motor terminal a second-order shunt filter may be connected at the motor terminals, where the filter components are chosen such that filter impedance matches the cable impedance as shown in equation (1.6) (von Jouanne et al., 1996).

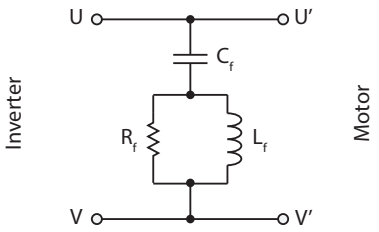


Figure 1.8: Second order shunt filter.

$$Z_{eq,filter} = R_c = \sqrt{\frac{L}{C}} \tag{1.6}$$

Simulation and experimental results shows that the shunt filter successfully reduces the voltage ringing and $\frac{dV}{dt}$ at the motor terminals. A first-order shunt filter, without the impedance L_f, did show even better reductions of the voltage ringing and reduced the $\frac{dV}{dt}$ to half that without filter (von Jouanne et al., 1996).

Acoustic noise:
The acoustic noise from the switching harmonics may be removed by the use of a sine wave filter, as illustrated in figure 1.9. Measurements of reduced acoustic noise using a sine wave filter is shown in figure 1.10. The sine wave filter is designed with a cut-off frequency below the inverter switching frequency, why the motor terminals will not be excited with the harmonics of the switching frequency, thus reducing the acoustic switching noise (Hanigovszki et al., 2007).

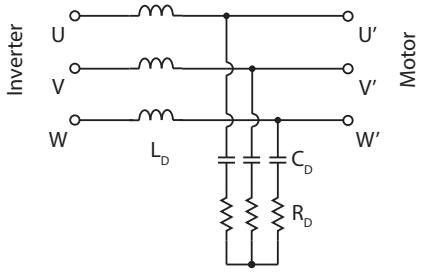


Figure 1.9: Sine wave filter.

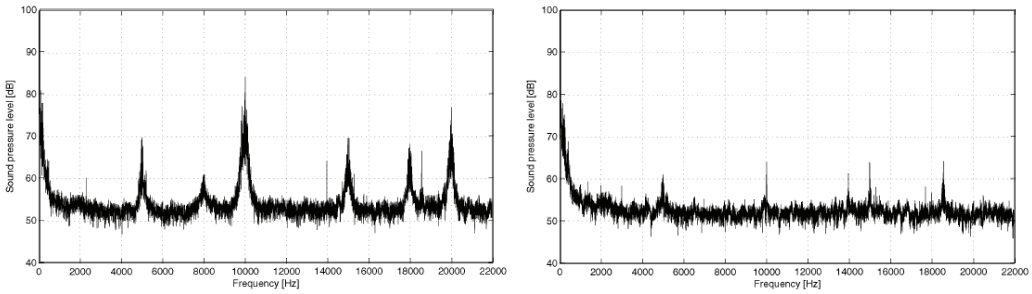


Figure 1.10: Measured acoustic noise radiated by an induction motor (Hanigovszki et al., 2007).

A single filter solution:
In the case that it is sought to reduce all three problems described above with a single filter, then the sine wave filter would be the choice, as shown in figure 1.11. With the cut-off frequency below the switching frequency, the acoustic switching noise is

reduced along with the ringing at the motor terminals, due to a reduction in $\frac{dV}{dt}$. The bearing currents that are due to high $\frac{dV}{dt}$ are reduced with the sine wave filter and the common mode currents may be reduced by connecting the filter with the DC-link.

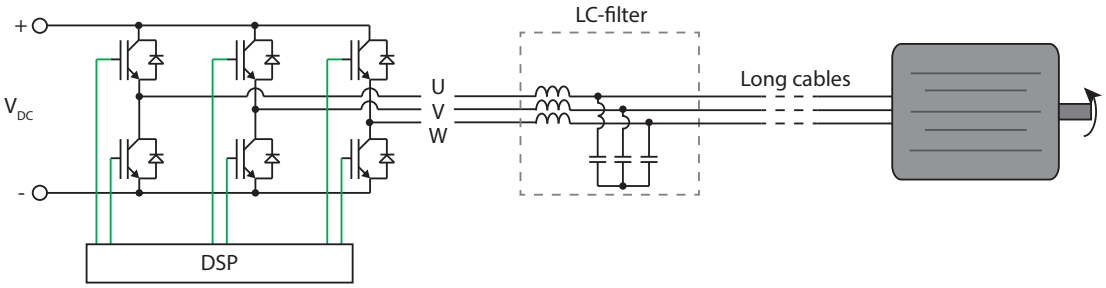


Figure 1.11: Implementation of LC-filter on the inverter output.

A disadvantage of the sine wave filter is that the relative price is more than double the price of the other filter options (Hanigovszki et al., 2007), also the relative size of the sine wave filter can be up to 10-20 times bigger (Danfoss, 2010). Another disadvantage is that when the sine wave filter is used the control algorithms that control the motor has to take into account the filters inductance and capacitance (Hanigovszki et al., 2007).

Parameter identification may be used to determine the parameters of the system, which are used to control of the system. When the LC-filter is added the order of the system is increased and parameters are added to the system, why the common methods for parameter identification do not function properly (Szczupak and Pacas, 2007a).

In a previous project of the participants of this project (Christensen and Weber, 2014), tests were made on a system with an LC-filter on the inverter output, controlling a PMSM. It was concluded that experimental data could be processed, with spectral analysis and an optimization algorithm, and yield satisfying results for parameter identification. The parameter identification was computed on a desktop computer as proof of concept, but showed problems with the optimization algorithm, that are disadvantageous for parameter identification of the system, for industrial application, where automatic and reliable solutions are needed. The strategy of parameter identification was based on voltage and current measurement at the inverter and filter, in order not to add additional sensors to the PMSM.

1.3 Problem Statement

This project will focus on development of a parameter identification method and design of the hardware needed to measure and excite the system in order to answer the main question:

How is a parameter identification method developed, for a drive system with an LC-filter on the inverter output, that is suitable for industrial applications based on current and voltage measurements?

In order to answer the main question, the project will focus on the following three questions:

- How are parameters of a drive system, with an LC-filter on the inverter output, identified?
- What hardware is necessary in order to determine the system parameters?
- How accurately can the parameters be identified?

The aim is to develop an automatic stand alone solution without the need for external interaction or computational power.

Chapter 2

System Models

The following chapter will include the description, analysis and model of the main components. This will include the standard hardware used to test the parameter identification method. The components are simulated using MathWorks Simulink and Plecs simulation software for power electronics. The components used are shown in figure 2.1, and the main specifications are shown in table 2.1.

Figure 2.1: The parts of the experimental setup (Somer), (Danfoss, 2011), (Danfoss, 2010).

Inverter			**Filter**			**PMSM**		
Rated Power	22	[kW]	Cut off freq.	2	[kHz]	Rated Power	22.8	[kW]
Rated Current	61	[A]	Min. freq.	4	[kHz]	Rated Speed	1500	[RPM]
			Rated Current	48	[A]	Weight	79	[kg]

Table 2.1: Selected system specifications.

2.1 Inverter

The inverter is a Danfoss VLT designed for variable speed control of AC-machines, including permanent magnet synchronous motors. The standard interface board of the VLT have been bypassed to allow access directly to the IGBT's gate drivers. The relevant specifications are shown in table 2.2.

Danfoss VLT FC302		
Rated Power	22	[kW]
Rated Output	42.3	[kVA]
Rated Current	61	[A]
Rated Input Current	55	[A]
Efficiency	0.98	[-]

Table 2.2: Inverter specifications.

The inverter has been disassembled to examine the structure of the electrical circuits, and each part has been simulated and verified to view the effect on the complete system. The (Equivalent Electrical Circuit) EEC from the supply to the inverter outputs is shown in figure 1.11.

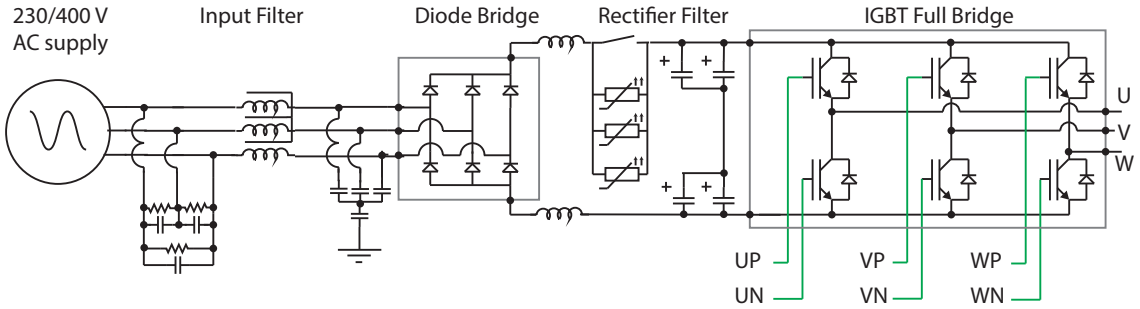


Figure 2.2: Overview of the VLT electrical circuit.

2.1.1 Input Filter

The VLT is connected to the 230/400V mains, shown on the left of the figure, followed by a Electro Magnetic Compatibility (EMC) filter. This is followed by a three phase diode bridge and a rectifier filter, and finally an IGBT full bridge power module, before the output terminals.

The input filter is used to both reduce the Electro Magnetic Interference (EMI) generated from the variable frequency drive itself, but also to protect the inverter from external noise sources. Radiated EMI (RFI), travels through the air and is limited by the grounded metal shielding, but the conducted EMI that travels through the wires,

printed circuits and electronic components is reduced by the filter. RFI is typically defined to range from 30 [MHz] to 1 [GHz], where EMI is defined as being between 150 [kHz] and 30 [MHz]. The EMI is divided into two categories: Common-Mode Noise (CMN) and Differential-Mode Noise (DMN). Common-mode or line-to-ground noise exist on all of the three phases and is in-phase with itself relative to PE, this means the current flows in the same direction and returns via the ground, illustrated in figure 2.3. The transmission of the noise is through parasitic capacitors, and stray electric and magnetic fields. This Common-Mode Noise is suppressed with the toroid inductor, which presents a high impedance to the noise, and by snubber capacitors C_{CMN}. The capacitors are connected from the lines to the ground and thereby suppresses the high-frequency CMN (Mohan et al., 1989).

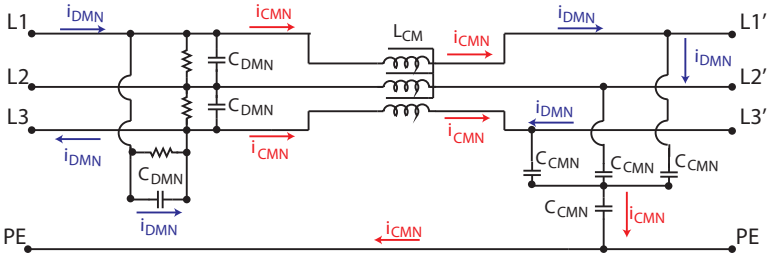


Figure 2.3: Illustration of Common-mode and Differential mode noise current flow in the input filter, shown in figure 2.2.

The differential-mode or line-to-line noise is present between the lines, and flows along one line and returns along another. The transmission of this noise is through the input lines, and from the inverter itself, and again stray capacitors and magnetic coupling contribute to the noise. This out-of-phase noise is suppressed with capacitors C_{DMN} between the phases, and resistors are used to discharge the capacitors when current is not flowing. No DMN current will flow in the ground conductor.

The input filter uses capacitors in the range of a few μF and when the inverter is simulated, the time step of the simulation is dictated by the value of the smallest capacitors or inductors, as the time step must be small enough to capture their charging and discharging. Taking this into account and the fact that a well designed filter should not have any influence on the gain or phase of the input, the input filter is left out.

2.1.2 Rectifier

The three phase rectifier bridge is a single module and the specifications are shown in table 2.3. The output of the rectifier bridge is connected to a filter inductor, or choke, together with a set of smoothing capacitors, used to smooth out the ripple in the rectified voltage. The capacitors are charged using Positive Temperature Coefficient (PTC) ceramic resistors, and their primary use is to limit the current inrush when charging. A PTC resistor will also limit the currents to safe levels in the event of a short circuit.

Diode Bridge - VUB 145-16 NO1		
Rated Voltage	1600	[V]
Rated Current	145	[A]
Forward voltage (@150 A, 25 °C)	1.68	[V]

Table 2.3: Rectifier Bridge specifications (IXYS Rectifier, 2008).

To verify the model of the rectifier, a signal that can be replicated with the model is produced, and used to excite the system. Figure 2.4 shows the DC bus voltage together with the current on phase U. A positive 10 % duty cycle was used on phase U in respect to phase V and W, while the filter and motor was connected. At no load the voltage ripple is $\simeq 1[V]$. When the signal is applied, at 20 [ms] on the figure, the DC shows a $\simeq 2[V]$ voltage ripple, with a $5[V]$ drop. This corresponds to $\sim 1\%$, at $23[A]$ (approximately 1/2 of rated current).

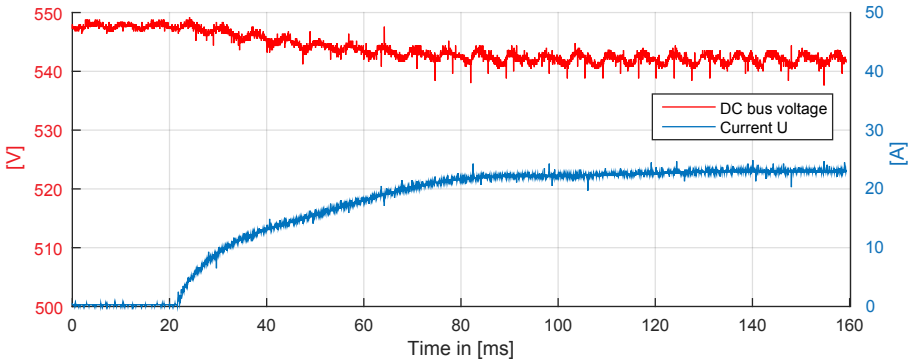


Figure 2.4: DC voltage shown from 500-550 [V], and current on phase U.

When the same scenario is simulated using the model it shows no voltage ripple at no load and a ripple of $\simeq 1[V]$ at $23[A]$. The model also shows a similar $\sim 1\%$ drop in voltage. The model includes the three phase AC supply, a diode bridge, where the diodes are modelled with the forward voltage and zero on-resistance, and the input filter is modelled as the inductors and capacitors. The capacitors are always given time to charge before any data is collected.

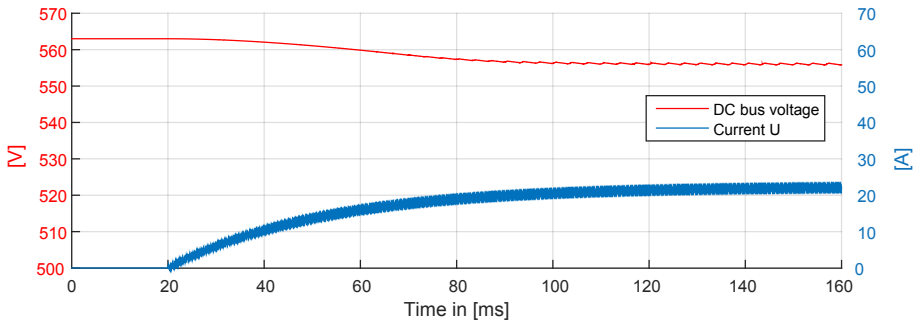


Figure 2.5: Simulated DC voltage shown from 500-570 [V], and simulated current on phase U.

2.1.3 Power Module

The next part of the model is the power module. The VLT uses an Infinion three phase full bridge Insulated Gate Bipolar Transistor (IGBT), module. The specifications of the IGBT module are shown in table 2.4

Trench-Field-Stop IGBT		
Collector-emitter Voltage	1200	[V]
Continous DC collector current	150	[A]
Collector-emitter saturation voltage	1.7	[V]
Anti-parallel Diode		
Repetitive peak reverse voltage	1200	[V]
Continous DC forward current	150	[A]
Forward Voltage	1.65	[V]

Table 2.4: Power Module specifications (Infineon FS150R12KT3, 2014).

First the structure of the IGBTs is examined, to accurately simulate the system, and to find the limitations in regards to current and voltage operating area and limits to switching frequency.

Semiconductor theory
Transistors are power semiconductors, composed of silicon structures. A semiconductor conducts electric current better than insulators, but worse than metals. A material's ability to conduct electricity is dependant on, if there are any charge carriers in the material, which are free to move as a response to an applied electric field. The reason why silicon is used, is its ability to change the free-carrier density, by introducing impurities to the material.

Pure silicon atoms have four electrons in their outer most shell, which allows it to make bonds with four other silicon atoms, creating a lattice structure shown in figure 2.6 a). This configuration has few free electrons, so it will not be a good conductor. The desire is to control the free-carrier density and this is done be doping the semiconductor. There are two types of doping: p-type and n-type, shown in figure 2.6 b). To create n-type, donor elements such as phosphorus is injected. Phosphorous has five electrons in its outer most shell and thereby adds an extra electron, giving it more carriers and it therefore conducts electricity better. To create p-type, acceptor elements such as boron is injected which has just three electrons in its outer most shell. This creates an empty bond or a 'hole' where an electron can move into, and therefore this also increases the semiconductors conductivity. The names of the types are given from the type of charge that can move in the lattice. In n-type it is the negative electron, and in the p-type it is the positively charged 'hole', or lack of electron that moves. Figure 2.6 c) shows this flow of electrons.

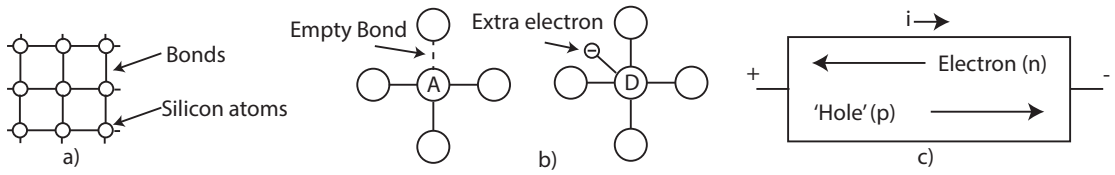


Figure 2.6: a) 2D silicon structure. b) p-type, doped by acceptors and n-type, doped by donors. c) Illustration of electron flow in n-type and p-type.

When a p-type is adjacent to a n-type a depletion layer will form, shown in figure 2.7 b). Here electrons have diffused from the n-type, where electrons are in excess to the p-type where they are missing, thus creating a negatively charged barrier on the p-type, and the opposite for the n-type layer. When a negative voltage potential is applied on the p-doped semiconductor in relation to the n-doped semiconductor, the pn junction will function as a diode, blocking the flow of current, shown in figure 2.7 a). The negative terminal will attract the electrons and the positive terminal will attract the 'holes', thus increase the depletion layer. This is called reverse bias. When the polarity is switched, called forward bias, the positive terminal is connected to the p-type and the negative terminal is connected to the n-type. This will result in shrinking of the depletion layer, and if the applied voltage is over a specified range, the electrons will travel through the depletion layer to the p-type, and the 'holes' will travel through the depletion layer to the n-type, allowing a current to flow.

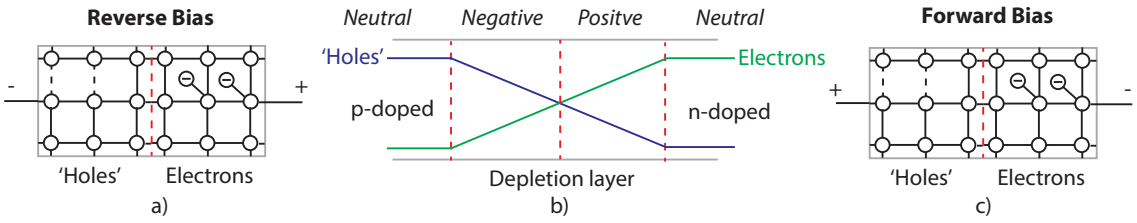


Figure 2.7: Illustration of a pn junction and the depletion layer. The blue curve is the concentration of holes, and the green curve is the concentration of free electrons.

For the description of the IGBTs the conventional naming of the transistors will be used, illustrated in figure 2.8

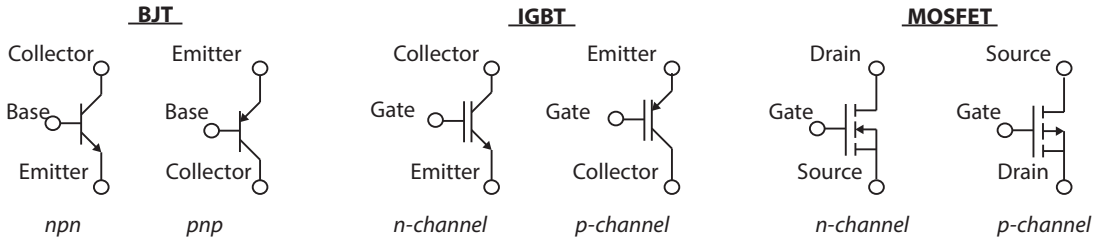


Figure 2.8: Symbols and naming of Bipolar Junction Transistor (BJT), Insulated Gate Bipolar Transistor (IGBT), and Metal-Oxide-Semiconductor Field-Effect Transistor (MOSFET).

A MOSFET is shown in figure 2.9. The transistor has n-type in the ends and p-type in the middle. Electrical contacts are connected at each end, one on the source and one on the drain. A third contact, the gate, is used as the switch to turn on the transistor. The gate is insulated from the semiconductor by an oxide layer. When no charge is applied to the gate the depletion layer will repel all electrons from the n-type, thus preventing the flow of current through the transistor. When a positive charge is applied to the gate this will attract electrons from the n-type and thereby shrinking the depletion layer when the transistors is turned on, creating a conducting channel from the source to the drain. (Baliga, 2010)

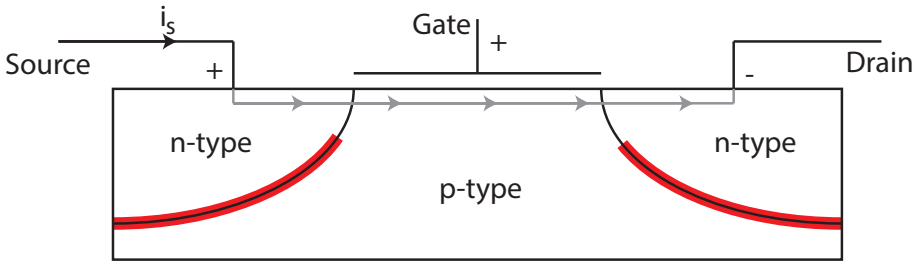


Figure 2.9: Basic n-channel MOSFET.

Trench-Field-Stop IGBT

Figure 2.10 shows the structure of the Trench-Field-Stop IGBT, with an imposed two-transistor equivalent circuit. The concentration of the doping of each layer is denoted by '+' and '-', so a n^+ layer is heavily doped by donors, and n^- is lightly doped. The MOSFET is voltage controlled, like described in the semiconductor section, and drives the intrinsic bipolar pnp transistor.

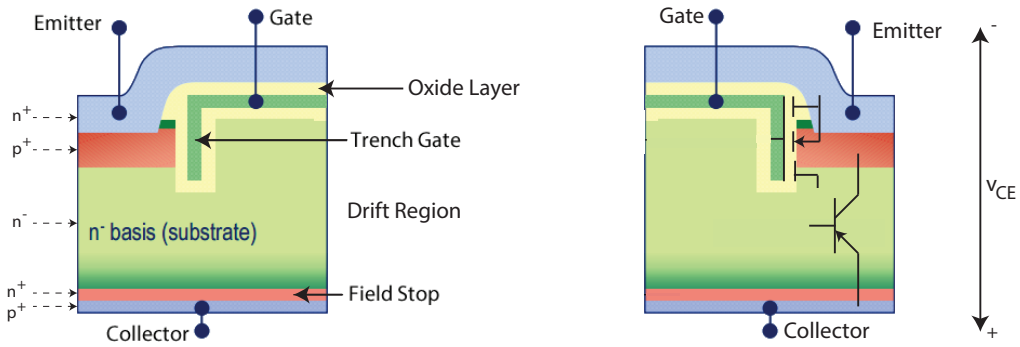


Figure 2.10: IGBT split in the middle illustrating the structure of the Trench-Field-Stop IGBT and the two-transistor equivalent circuit.

The emitter sided pn-junction of the bipolar transistor resembles the IGBT's collector. The structure of the pnp transistor is shown in figure 2.11. When a positive voltage

is applied at the base relative to the collector, electrons will cross the pn-junction between the base and the emitter. The base layer is lightly doped, so the density of free electrons is low, but the collector and emitter are highly doped and therefore the more electrons will flow from the collector to the emitter. This results in a small current between the base and the collector can control a much larger current between the emitter and collector.

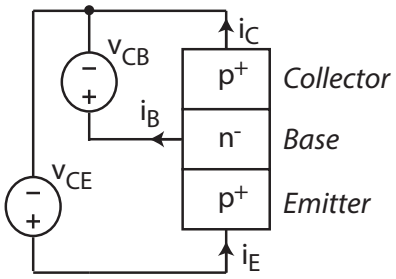


Figure 2.11: Structure of the pnp transistor.

The gate driving characteristics of the IGBT is similar to a power MOSFETs, while the output characteristics has a dominant forward voltage drop, compared to a MOSFETs dominant on-resistance. This forward voltage is due to the pn-junction on the transistor, that like a diode has a characteristic voltage drop when conducting current. The equivalent circuit is useful to understand the workings of the IGBT but a MOSFET driving a diode, connected with the cathode on the drain, would be more accurate in regards to the electrical circuit (Infineon IGBT Modules Explanation, 2013).

The IGBTs used in the power module are Infineon Trench-Field-Stop high speed IGBTs. They are designed to reduce switching losses, and some of the same things that generate losses, sets the limitations for the switching (Infineon IGBT, 2014). The field stop layer is an additional n-doped layer, that enables a thinner base region, with the same blocking voltage. The field stop, or n+ buffer layer makes it possible to have a small n^- drift region, compared to an IGBT without field stop. Carriers stored in the n^- drift region, needs to be removed when the IGBT is turned off, and by having a thinner region this can be done faster, thus reducing the switching time (Rahimo et al., 2001). The n^+ buffer layer on top improves the turn off speed by allowing faster recombination at the emitter pn-junction.

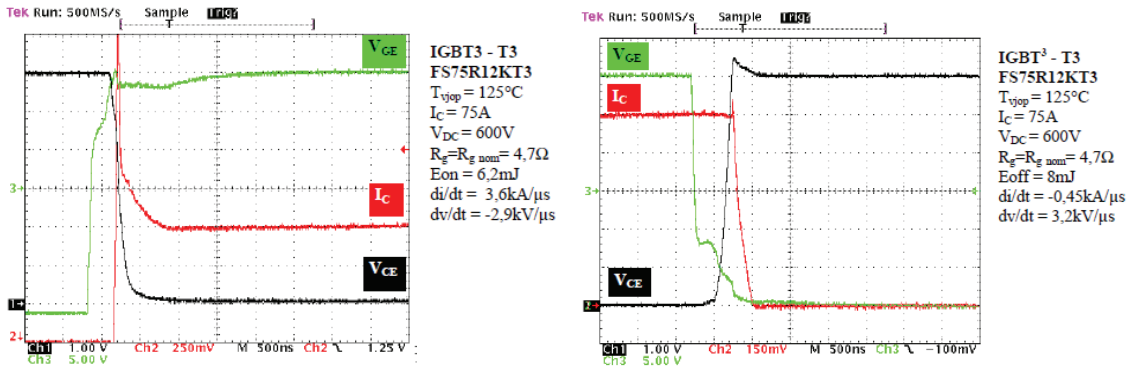


Figure 2.12: a) Turn on characteristics. b) Turn off characteristics (Infineon IGBT3 T3, 2005).

The turn on and turn off characteristics plots from the manufacturer are shown in figure 2.12. These characteristics depend on both the IGBT structure, but also on the anti-parallel diode used, and other factors such as the gate resistance and stray capacitance in the module. The IGBTs have a well-defined blocking capability from collector to emitter, but a weak and undefined blocking capability in the other direction, therefore an anti-parallel diode is used. The inverter uses hard-switching, which is visible on the plots, as the voltage and current overlaps. The spike on the turn on current is the load current plus the reverse recovery current of the anti-parallel diode of a complementary IGBT (International Rectifier, 2013). On the plot of the turn off characteristics it should be noted that current tail is very short due to the Trench-field-stop design. The gate-emitter voltage has an abrupt controlled fall at turn off, and only a short uncontrolled region where recombination is occurring in the n^- drift region. The main advantages of IGBTs are its low on state conduction losses, the relatively fast switching speeds, and the ability to switch high voltages without damage, which makes it ideal for inductive loads. (Infineon IGBT Modules Explanation, 2013)

2.1.4 Model and implementation

In the model the IGBTs are modelled as ideal switches with an ON resistance and forward voltage in series. The anti-parallel diodes are also modelled with an ON resistance and a forward voltage, illustrated in figure 2.13.

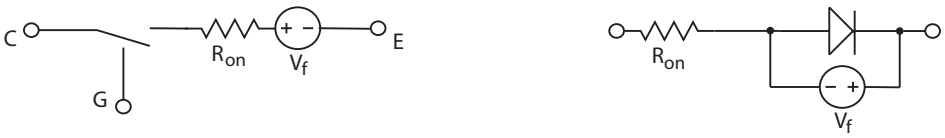


Figure 2.13: Electrical model of IGBT and diode.

The IGBTs and diodes are combined to form a three phase two level IGBT full bridge power module. An example of an Infineon EconoPACK IGBT module is shown in figure 2.14.

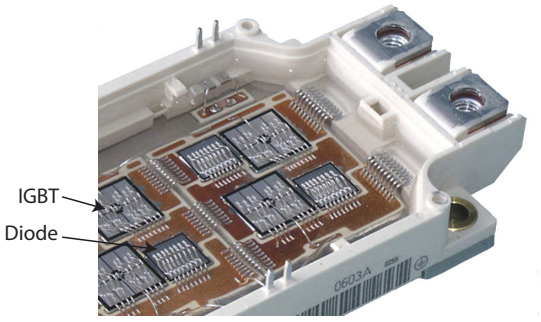


Figure 2.14: Inside of power module, showing the IGBTs, diodes and wire bonds (Power Systems Design).

The complete model of the inverter used in the subsequent simulations of the system is shown in figure 2.15. The IGBTs are controlled by the input signals, where the blue signal is for the upper or positive IGBT, and the red is for the lower or negative IGBT. The control signals for a single phase, and the resulting output are illustrated with deadtime implemented.

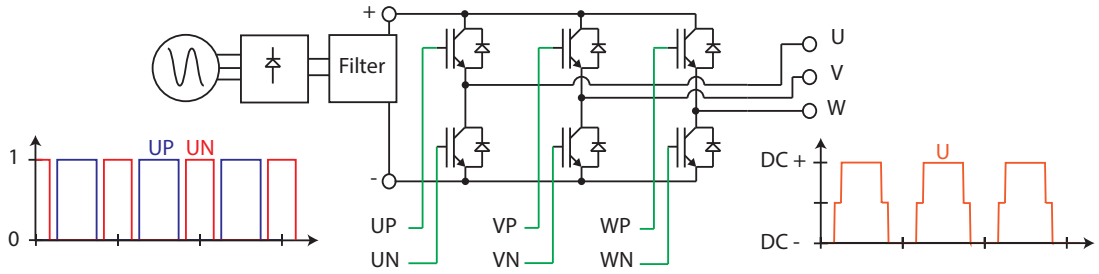


Figure 2.15: Positive and negative control signals to phase U, inverter model, and resulting phase voltage on phase U.

Deadtime

Deadtime is used to make sure a phase-leg of the power module is not conducting on both IGBTs at the same time. The real IGBTs are not ideal switches, so a delay from when one IGBT is turned off to the other is turned on is implemented in the control signal generation for gate drivers. The minimum duration of the delay is calculated using equation (2.1), given in (Infineon Deadtime, 2007), and the specifications listed in table 2.5.

$$t_{dead} = \left((t_{d-off-max} - t_{d-on-min}) + (t_{pdd-max} - t_{pdd-min})\right) \cdot S_f$$
$$t_{dead} = 378[ns] \tag{2.1}$$

Maximal turn off delay time	$t_{d-off-max}$	520	[ns]
Minimum turn on delay time	$t_{d-on-min}$	250	[ns]
Maximal propagation delay of driver	$t_{pdd-max}$	190	[ns]
Minimum propagation delay of driver	$t_{pdd-min}$	145	[ns]
Safety Factor	S_f	1.2	[]

Table 2.5: Timing specifications for IGBTs and gate drivers.

The difference between the maximum turn off time and the minimum turn on time, is the first term of the equation, describing the characteristic of the IGBTs, and the second term is the maximum and minimum propagation delay of the gate drivers used. The rise and fall times of the IGBTs are small compared to the delays and are not included. The definitions of the delay times are shown in the following:

- t_{d-on} : from 10% of v_{GE} to 10% of i_C
- t_{rise} : from 10% of i_C to 90% of i_C
- t_{d-off} : from 90% of v_{GE} to 90% of i_C
- t_{fall} : from 90% of i_C to 10% of i_C

Equation (2.1) is normally used with measured data with this safety factor, and because all the values are listed in the data sheets and are typical values, a larger safety factor should be used. The deadtime used for the system is limited by the interface board, and is chosen to 2 [μs].

2.1.5 Switching and deadtime implementation

The switching and deadtime is implemented in the model. It relates closely to the PWM generation and will be explained in section 4.2.2 "*PWM Strategy and Implementation*". To verify the implementation, the Interface and Protection (IPC3) board is disconnected from the VLT and its outputs are measured. The input to the IPC3 is a 50% duty cycle on phase V and a 55% duty cycle on phase U. The complementary PWM signals, UN and VN are generated from the IPC3 board.

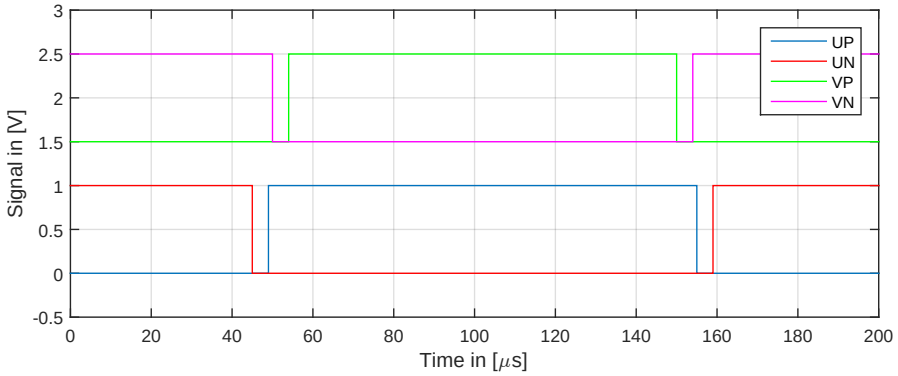


Figure 2.16: Switching control signals for the IGBTs from the model.

The simulated signals are shown in figure 2.16 and the measured signals are shown in figure 2.17.

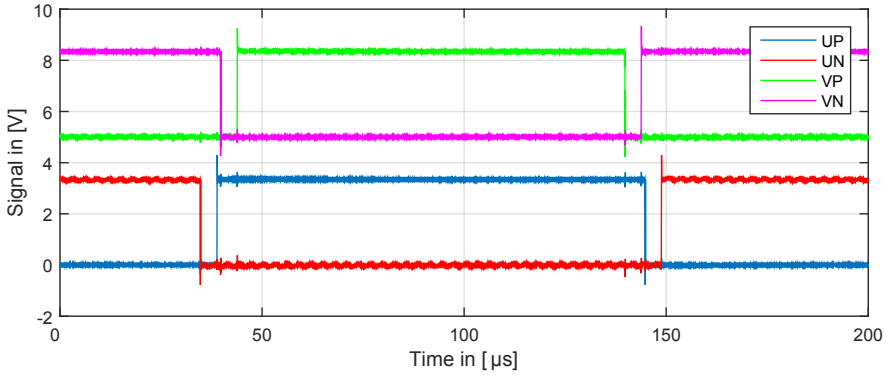


Figure 2.17: Switching control signals for the IGBT drivers measured on the Interface and Protection (IPC3) board.

2.2 Sine Wave Filter

The Danfoss sine wave filter is a low-pass filter designed to attenuate the high frequency components of the inverter output, resulting in a sinusoidal phase to phase voltage and current wave form, as described in section 1 "*Introduction*". The filter will be analyzed and a model will be presented. The specifications for the filter are shown in figure 2.6, which are data sheet values and calculated cut off and resonant frequencies.

Danfoss Sine Wave Filter		
Rated Voltage	200-500	[VAC]
Rated Current @ 50 [Hz]	48	[A]
Rated Current @ 100 [Hz]	36	[A]
Minimum Switching Frequency	4	[kHz]
Maximum Output Frequency	100	[Hz]
Inductance	1.1	[mH]
Capacitance*	14.7	[μF]
Power Loss	260	[W]
Cut off Frequency	2	[kHz]
Resonance Frequency	1.25	[kHz]
Weight	39	[kg]

Table 2.6: Filter specifications.
*Equivalent wye connection value.

2.2.1 Analysis

The filter is composed of a three phase laminated core inductor and a molded three phase capacitor module. From the data sheet the EEC given by the manufacturer is shown in figure 2.18. The input terminals of the filter are denoted U, V, and W, and the output terminals are denoted U', V', and W'.

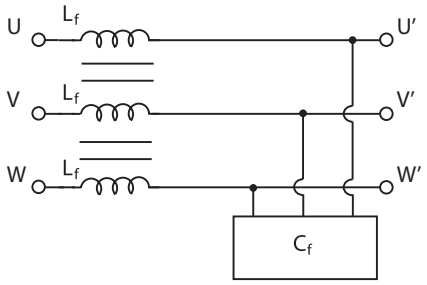


Figure 2.18: Simple electrical circuit of Sine Wave Filter from data sheet.

The EEC is shown without any resistors, and the filter does have a very low resistance in both the connecting cables and in the inductors. The resistance from input terminal to output terminal on a single phase is measured by applying a DC voltage between the terminals and then measure the current, and the results are shown in table 2.7.

Phase	**Voltage**		**Current**		**Resistance**	
U - U'	224.9	[mV]	1.993	[A]	0.113	[Ω]
V - V'	222.4	[mV]	1.994	[A]	0.112	[Ω]
W - W'	223.4	[mV]	1.998	[A]	0.112	[Ω]
Average					0.112	[Ω]

Table 2.7: Resistance in sine wave filter.

The filter without the cover is shown in figure 2.19.

a) b)

Figure 2.19: Picture of the filter capacitor and inductor.

Capacitor Module

The capacitor module is connected on the output terminals of the sine wave filter, shown in figure 2.19 a). To verify the data sheet values of the capacitor the capacitance is measured with a LCR-meter. Since the connection type is not described in the documentation, and the individual capacitors are not accessible, the capacitance is measured over a circuit of capacitors and this value is then converted to an equivalent wye or delta value. Two measurements are made: For the first measurement the LCR-meter is connected with one terminal on phase U', and the other on phase V', while phase W' is left open. The second measurement the LCR meter is connected with one terminal on phase U', and the other terminal is connected to both phase V' and W'. The two possible circuits are shown in figure 2.20.

Figure 2.20: a) Capacitor bank with wye connection. b) Capacitor bank with delta connection.

For the wye connection the ratio from the measured values to the equivalent values are calculated from equation (2.2) and equation (2.3). When the capacitance is measured from U' to V' the circuit is two capacitors in series, and when the capacitance is measured from U' to V' and W', the circuit is a single capacitor in series with two parallel capacitors.

$$C_{u'-v'} = \frac{1}{\frac{1}{C_{f,wye}} + \frac{1}{C_{f,wye}}} \Longrightarrow C_{f,wye} = 2C_{u'-v'} \tag{2.2}$$

$$C_{u'-v'w'} = \frac{1}{\frac{1}{C_{f,wye}} + \frac{1}{C_{f,wye}+C_{f,wye}}} \Longrightarrow C_{f,wye} = \frac{3C_{u'-v'w'}}{2} \tag{2.3}$$

For the delta connection the values are calculated from equation (2.4) and equation (2.5). When the capacitance is measured from U' to V' the circuit is two capacitors in series, parallel with a single capacitor, and from U' to V' and W' the circuit is two parallel capacitors.

$$C_{u'-v'} = C_{f,delta} + \frac{1}{\frac{1}{C_{f,delta}} + \frac{1}{C_{f,delta}}} \Longrightarrow C_{f,delta} = \frac{2C_{u'-v'}}{3} \tag{2.4}$$

$$C_{u'-v'w'} = C_{f,delta} + C_{f,delta} \Longrightarrow C_{f,delta} = \frac{C_{u'-v'w'}}{2} \tag{2.5}$$

When comparing equation (2.2) and equation (2.3) to equation (2.4) and equation (2.5) it can be seen that the ratio between $C_{u'-v'w'}$ and $C_{u'-v'}$ is the same, in both cases: $C_{u'-v'} = 0.75C_{u'-v'w'}$. This means that regardless of the physical structure of the capacitors, mathematically the capacitor module can always be seen as a wye-connection,

which has benefits when setting up the system equations. The measured values are shown in table 2.8, together with the calculated values.

	u'-v'		u'-v'w'	
Measured Values	7.171	[μF]	9.615	[μF]
Calculated Values				
Wye	14.342	[μF]	14.422	[μF]
Delta	4.781	[μF]	4.808	[μF]

Table 2.8: Measured values from capacitor module, and calculated equivalent values.

Inductors

The three coils are mounted on a single core made out of laminated steel sheets. Using a magnetic core increases the inductances of the coils, by containing and concentrate the magnetic field. The magnetic field stores recoverable energy, and by using a single core, more of this energy can be recovered, reducing losses. The steel sheets are made of an iron alloy, called electrical steel, with primarily silicon added. The silicon increases the electrical resistivity, and this reduces eddy current loss. The sheets of the core are laminated for the same reason. Eddy current is a term for the circulating current that will occur when a time-varying magnetic field is applied to a magnetic core. The eddy currents flow in planes perpendicular to the magnetic field, and this circular flow will generate a secondary magnetic field in the opposite direction of the primary magnetic field. The eddy currents generates resistive losses, transforming the electric energy into heat. The laminated sheets creates gaps where electrons cannot cross, so the current cannot circulate in the entire core. The eddy currents will flow around the lines of flux so by placing the sheets parallel to these, the eddy currents are reduced. Any small flow will also be limited due to the high resistance in the thin sheets (Mohan et al., 1989).

Electrical steel has a relative permeability of 2000-30000, but this value varies with the field strength and frequency. The relative permeability is the ratio between the permeability in the material and the permeability of vacuum, and is unitless. The permeability is measured in [H/m], and describes the relation between the magnetizing field H, and the magnetic field B, shown in equation (2.21). The permeability is constant up until the material begins to show saturation. Here any increase in the strength of the applied magnetizing field will not increase the flux density of the core, and the curve levels off.

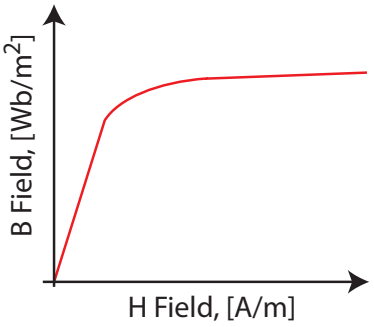


Figure 2.21: Example of typical BH curve for electrical steel.

When the permeability is no longer constant, the flux is no longer independent of the current, and this influences the inductances in the filter. The self-inductance is defined as shown in equation (2.6), where L is the self inductance, N is the number of windings. The inductance L, is independent of the current i, as long as there is a proportional relationship between the flux and the current.

$$L = \frac{N\phi}{i} \tag{2.6}$$

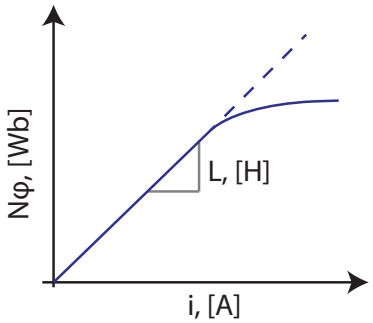


Figure 2.22: Self inductance. It is linear until saturation of the core material. The linearized inductance is shown in the dotted line.

From Faraday's voltage law it follows that:

$$\epsilon = N\frac{d\phi}{dt} \tag{2.7}$$

Where ϵ denotes the Electro Motive Force (EMF), induced by the time-varying flux. This leads to equation (2.8) describing the total induced voltage from the current in the coil itself, and the induced voltage from the adjacent coils. The EMF can be rewritten to equation (2.9), where M is the mutual inductance and its corresponding current.

$$v = L\frac{di}{dt} + \epsilon \tag{2.8}$$

$$\epsilon = \frac{d\phi}{dt} = M\frac{di}{dt} \tag{2.9}$$

When inserting equation (2.9) into equation (2.8) and combining the equations for all three phases it results in equation (2.10).

$$\mathbf{V} = \mathbf{L}_f \frac{d\mathbf{i}}{dt} + \mathbf{M}_f \frac{d\mathbf{i}}{dt} \tag{2.10}$$

In equation (2.10) the self inductance matrix's values are all on the diagonal, and the mutual inductance matrix's values are all off the diagonal. The two matrices can be combined resulting in equation (2.11).

$$\begin{bmatrix} V_U \\ V_V \\ V_W \end{bmatrix} = \begin{bmatrix} L_f & M_f & M_f \\ M_f & L_f & M_f \\ M_f & M_f & L_f \end{bmatrix} \frac{d}{dt} \begin{bmatrix} i_U \\ i_V \\ i_W \end{bmatrix} \tag{2.11}$$

No notations are made to the inductance specified in the data sheet, so a test is carried out to determine the inductance for each inductor. To determine the self inductance a function generator is used to generate a low voltage sine wave. The negative terminal of the function generator is connected to the U' terminal, and the positive terminal is connected to both phase U and phase V', shown in figure 2.23 a). By using the filter in this configuration an LC- or tank circuit is created.

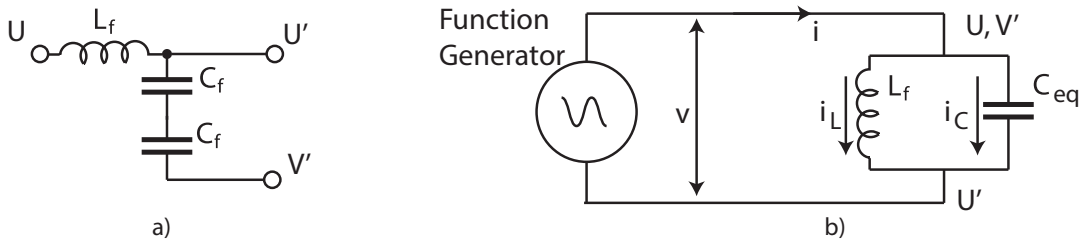


Figure 2.23: a) Section of the electric circuit of the filter. b) LC circuit and function generator.

Figure 2.23 b) shows the LC-circuit where the two capacitors have been replaced with a single equivalent. The voltage across the terminals, v is equal to the voltage over the two passive components, $v = v_L = v_c$. From Kirchhoffs current law it can be seen that that total current equals the sum of the current flowing though the inductor and the capacitor, $i = i_L + i_C$. Equation (2.12) shows the resonance frequency of the circuit.

$$\omega_0 = \sqrt{\frac{1}{L_f C_{eq}}} \tag{2.12}$$

Knowing approximate values for the inductance and capacitance, the function generater was used to swipe in the frequency range close to the resonance frequency. Using an oscilloscope the resonance frequency was found to $2.31[kHz]$. This is converted to

rad/s and inserted in equation (2.12), which is isolated for L_f. The result is shown in equation (2.13).

$$L_f = \frac{1}{4\pi^2 C_{eq} f_0^2} = 0.66mH \tag{2.13}$$

For redundancy the inductance in each phase was measured with a LCR meter, which yielded the same result as calculated in equation (2.13). This value is the self-inductance and does not include any mutual inductances from the adjacent coils. This value is much smaller than the inductance of $1.1mH$, given in the data sheet. A reason for the inductance to be different from the one specified could be that the inductance had been measured at a high load, but this would have resulted in a lower specified inductance relative to the measured. This is illustrated in figure 2.24, where the green line is the gradient at a higher load.

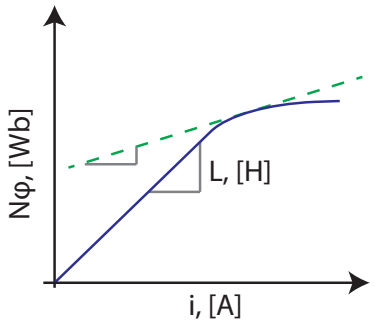


Figure 2.24: Self inductance. Comparison of inductances at different loads.

Another reason for the higher specified inductance could be that the mutual inductance is included in the term. Figure 2.25, shows the EEC of the filter, with the coupled inductors.

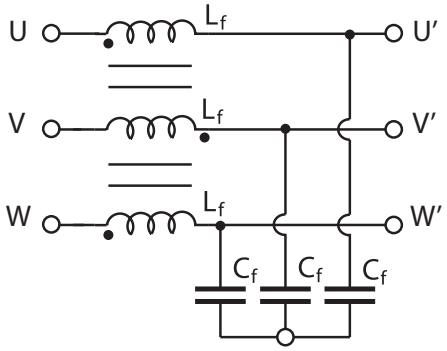


Figure 2.25: EEC of the filter.

To be able to eliminate the off diagonal terms of equation (2.11), it is utilized that the current cannot accumulate in a single point, and thereby equation (2.14) must be true.

$$i_U + i_V + i_W = 0 \Longrightarrow M_f i_U + M_f i_V = -M_f i_W \tag{2.14}$$

This yields equation (2.15). From examining the flux lines, and from this define the polarity of the inductors, the sign of the mutual inductance M_f should be negative.

$$\begin{bmatrix} V_U \\ V_V \\ V_W \end{bmatrix} = \begin{bmatrix} L_f - M_f & 0 & 0 \\ 0 & L_f - M_f & 0 \\ 0 & & L_f - M_f \end{bmatrix} \frac{d}{dt} \begin{bmatrix} i_U \\ i_V \\ i_W \end{bmatrix} \tag{2.15}$$

To analyze the filter, the input-output voltage relations are set up for a single phase. Each phase can be evaluated in relation to the neutral points of the capacitor module. The EEC of a single phase is shown in figure 2.26.

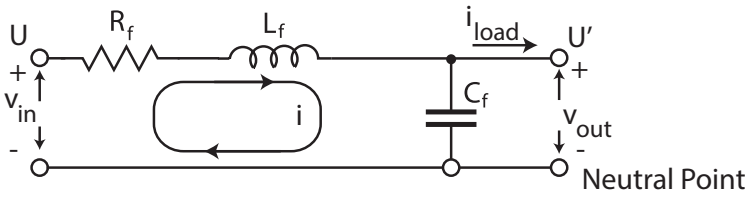


Figure 2.26: EEC of a single phase of the filter.

Equation (2.16) describes the circuit, shown in figure 2.26. When U' is open, i_{load} will be 0, and the current will circulate as illustrated. When using Laplace transformation and substituting equation (2.17) for i, the resulting equation (2.18) appears.

$$v_{in} = R_f \cdot i + L_f \frac{di}{dt} + v_{out} \tag{2.16}$$

$$i = C_f \frac{dv_{out}}{dt} \tag{2.17}$$

$$v_{in} = (C_f R_f s + C_f L_f s^2 + 1) \cdot v_{out} \tag{2.18}$$

Equation (2.19) shows the transfer function for the sine wave filter.

$$G_{SWF} = \frac{v_{out}}{v_{in}} = \frac{1}{C_f L_f s^2 + C_f R_f s + 1} = \frac{\frac{1}{C_f L_f}}{s^2 + \frac{R_f}{L_f} s + \frac{1}{C_f L_f}} \tag{2.19}$$

The bode diagram of the transfer function in shown in figure 2.27, for both the data sheet values and the measured.

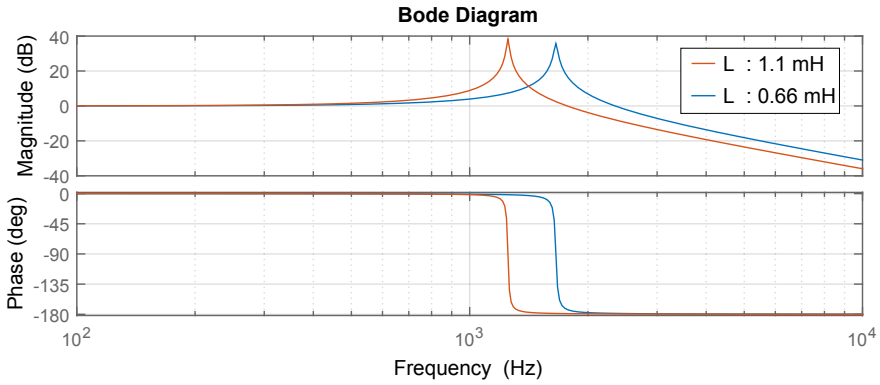


Figure 2.27: Bode diagram of the sine wave filter's voltage transfer function in equation (2.19).

The resonant frequency of the filter with the data sheet values is shown in equation (2.20), and for the filter with the measured in equation (2.21).

$$f_0 = \frac{1}{2\pi}\sqrt{\frac{1}{C_f L_f}} = 1.25[kHz] \tag{2.20}$$

$$f_0 = \frac{1}{2\pi}\sqrt{\frac{1}{C_f L_f}} = 1.66[kHz] \tag{2.21}$$

The large difference indicates that the mutual inductance has a considerable influence. To verify the mutual inductances influence the transfer function of the real filter is estimated. The method used will be explained in section 3.2 "*System Frequency Response Estimation*". The method estimates the transfer function from the output phase to phase voltage on U and V and the phase to phase voltage on the filter output U' and V'.

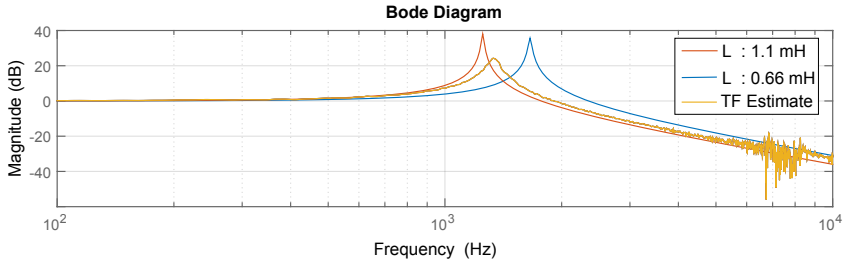


Figure 2.28: Transfer function estimate and analytical function of filter with data sheet inductance and measured inductance.

By reading the resonance frequency of the graph and rewriting equation (2.20) the inductance is calculated, shown in equation (2.22).

$$\begin{aligned} f_0 &= 1.33[kHz] \\ L &= \frac{1}{4\pi^2 C_f f_0^2} = 0.96mH \end{aligned} \tag{2.22}$$

This value is a combination of the self inductance and the mutual inductance. This value is closer to the data sheet value, and thereby better describes the filter response.

2.2.2 Model and implementation

The model is implemented in MathWorks Simulink with PLECS circuit simulator. Simple passive components has been used to simulated the EEC, shown in figure 2.29. In the analysis it was determined that the inductors could be modelled as decoupled inductors where the mutual inductance was included in the self inductance. The capacitor module is modeled as wye connected as it simplifies the equations, and the capacitors values can always be converted to equivalent delta values.

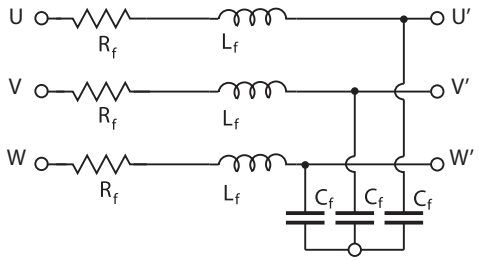


Figure 2.29: EEC of the sine wave filter as implemented in the Plecs Model.

To verify it, the filter model was coupled with the inverter model and a positive 60 % PWM signal was applied phase U in respect to V. The signal was applied as a step to examine both transient and steady state. The output signal of the inverter, and the simulated signal, is shown in figure 2.30, where the the output of the inverter was enabled at t = 0;

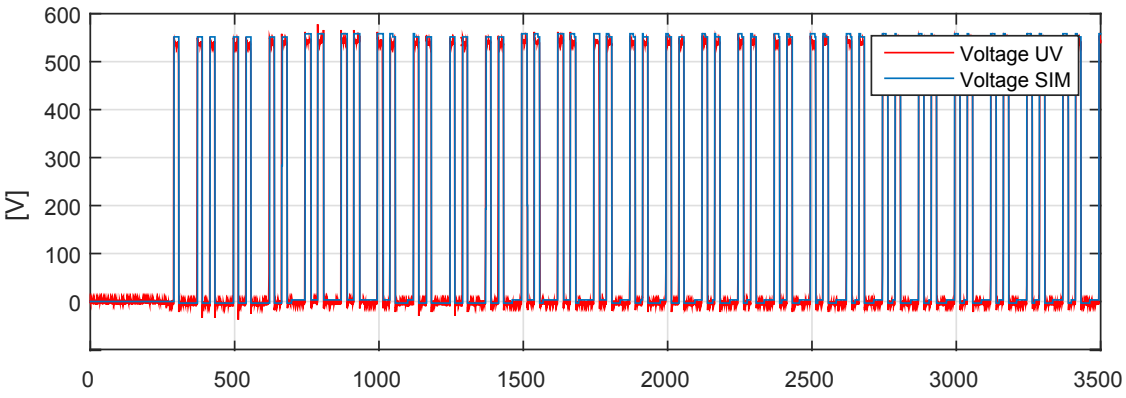


Figure 2.30: 60% positive PWM signal on phase U in respect to V. Simulated PLECS model and voltage measurements on phase U relative to phase V on inverter output.

The current is measured on phase U and compared with the simulated current, shown in figure 2.31.

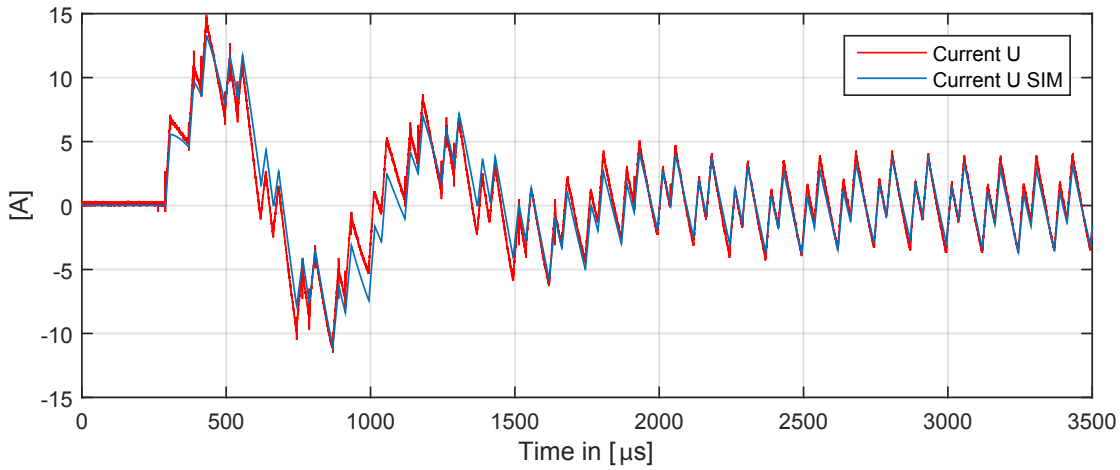


Figure 2.31: 60% positive PWM signal on phase U in respect to V. Simulated PLECS model and current measurements on phase U.

Finally the voltage is measured on the output terminals of the filter, shown in figure 2.32

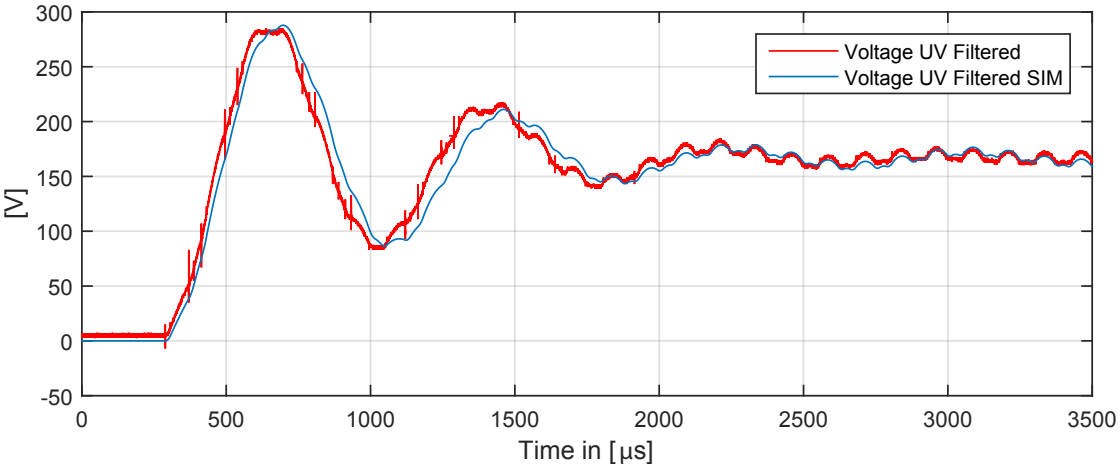


Figure 2.32: 60 % positive PWM signal on phase U in respect to V. Simulated PLECS model and voltage measurements on phase U relative to phase V after filter.

2.3 Permanent Magnet Synchronous Motor

The Permanent Magnet Synchronous Motor (PMSM) used in the experimental setup is a Leroy Somer LSRPM160LR. This type of motor is typically used in industrial applications, and almost always together with variable speed drives, like the Danfoss VLT. The motor is shown in figure 2.33 and the specifications are shown in table 2.9.

Figure 2.33: Picture of the PMSM.

Leroy Somer PMSM		
Rated Speed	1500	[RPM]
Rated Voltage	360	[V]
Frequency	100	[Hz]
Rated Current	43	[A]
Max Current	64.5	[A]
Rated Torque	145	[Nm]
Max Torque	217.5	[Nm]
Pole pairs	4	[-]
Weight	79	[kg]
Efficiency	93.5	[%]

Table 2.9: Motor specifications.

2.3.1 Analysis

The motor is a synchronous AC motor, with an 8 pole permanent magnet rotor. The rare earth permanent magnets are mounted on a core made of electrical steel sheets. When permanent magnets are used the air gap between the stator and the rotor can be large compared to an asynchronous motor because no energy is needed to induce a magnetic field in the rotor. The large air gap lowers the gradient of the BH curve, the permeability, as the magnets permeability is near unity and behaves like an air gap. The stator is coils wound on electrical steel sheets. The EEC of the motor is shown in figure 2.34.

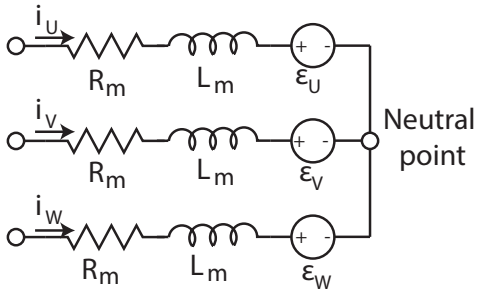


Figure 2.34: EEC of the motor.

The governing equation of the EEC is shown in equation (2.23),

$$\mathbf{u} = \mathbf{R}_m\mathbf{i} + \mathbf{L}_m\frac{d\mathbf{i}}{dt} + \epsilon_{mu} + \epsilon_{pm} \tag{2.23}$$

The resistance and inductance are assumed identical for the three phases and by using the same assumptions as in the derivation of the filter equations the induced volt-

age due to mutual inductance is included in the inductance term, shown in equation (2.24).

$$\mathbf{L}_m \frac{d\mathbf{i}}{dt} + \epsilon_{\mathbf{mu}} = \begin{bmatrix} L_m & M_m & M_m \\ M_m & L_m & M_m \\ M_m & M_m & L_m \end{bmatrix} \frac{d}{dt} \begin{bmatrix} i_U \\ i_V \\ i_W \end{bmatrix} \tag{2.24}$$

When this is utilized and when the EMF from the permanent magnets is written as a back EMF constant and the rotational velocity, the governing equation can be written as equation (2.25)

$$\begin{aligned} \begin{bmatrix} u_U \\ u_V \\ u_W \end{bmatrix} = & R_m \begin{bmatrix} i_U \\ i_V \\ i_W \end{bmatrix} + \begin{bmatrix} L_m - L_{mu} & 0 & 0 \\ 0 & L_m - L_{mu} & 0 \\ 0 & 0 & L_m - L_{mu} \end{bmatrix} \frac{d}{dt} \begin{bmatrix} i_U \\ i_V \\ i_W \end{bmatrix} \dots \\ & + \lambda_{pm} \frac{d}{dt} \begin{bmatrix} cos(\theta_e) \\ cos(\theta_e - \frac{2}{3}\pi) \\ cos(\theta_e - \frac{4}{3}\pi) \end{bmatrix} \end{aligned} \tag{2.25}$$

To determine the parameters a step test is performed on the motor. A function generator was connected with the positive terminal on the phase U, and the negative terminal was connected to phase V and phase W. At standstill the EEC is reduced, shown in figure 2.35 a). This can be simplified to figure 2.35 b), forming a simple RL circuit.

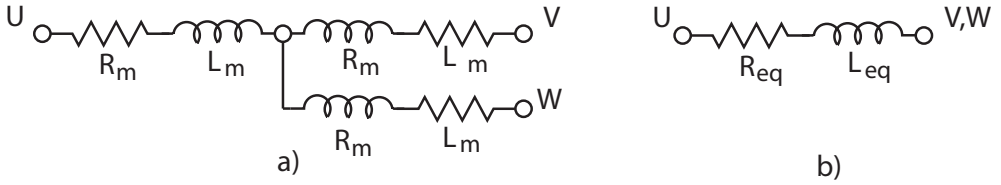


Figure 2.35: a) Connection of phases for step test. b) Equivalent simplified circuit of a)

For the first order transfer function of the RL circuit the time constant is shown in figure 2.26.

$$\tau = \frac{L_{eq}}{R_{eq}} \tag{2.26}$$

From this the equivalent inductance can be calculated, using equation (2.27).

$$L_{eq} = \tau R_{eq} \tag{2.27}$$

For the test an oscilloscope was connected to measure the voltage on phase U in respect to phase V and W, and a the current on phase U. A step on the voltage was applied at time = 0 on figure 2.36.

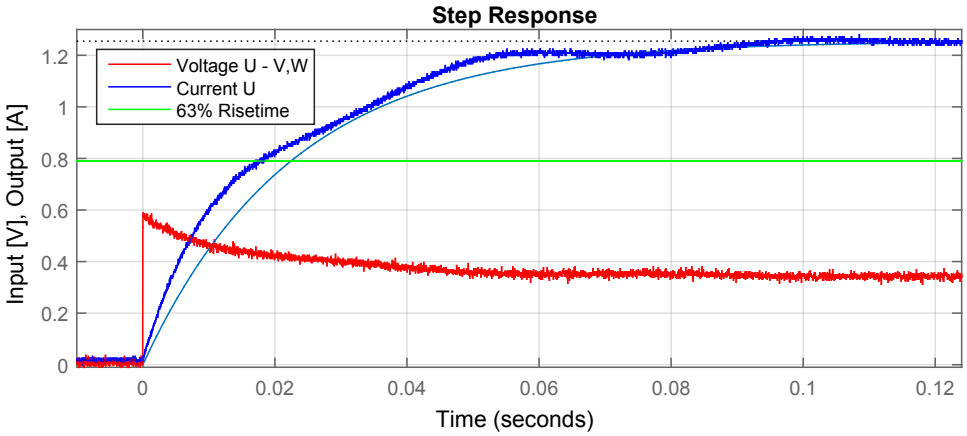


Figure 2.36: Voltage step measured on phase U relative to phase V and W, and the resulting current on phase U. Green line is when the current is at 63%, and the light blue is a step response with the resulting transfer function with the found values.

The results of the step test, converted from equivalent values, are shown in table 2.10.

	Value	Unit
R_m	0.18	[V]
L_m	3.29	[mH]

Table 2.10: Results from step test.

2.3.2 Model and implementation

The PLECS model of the motor is implemented by transforming the governing equation equation (2.28), to an equivalent set of equations in a rotating dq-reference frame, shown in equation (2.29) and equation (2.30).

$$\begin{bmatrix} u_U \\ u_V \\ u_W \end{bmatrix} = R_m \begin{bmatrix} i_U \\ i_V \\ i_W \end{bmatrix} + \begin{bmatrix} L_m - L_{mu} & 0 & 0 \\ 0 & L_m - L_{mu} & 0 \\ 0 & 0 & L_m - L_{mu} \end{bmatrix} \frac{d}{dt} \begin{bmatrix} i_U \\ i_V \\ i_W \end{bmatrix} \dots + \lambda_{pm} \frac{d}{dt} \begin{bmatrix} cos(\theta_e) \\ cos(\theta_e - \frac{2}{3}\pi) \\ cos(\theta_e - \frac{4}{3}\pi) \end{bmatrix} \tag{2.28}$$

The transformation is done using Clarke and Park transformation, and this is also used to transform the phase voltages from the sine wave filter to the motor. The reason for transforming the equations to another reference frame is to simplify the model, as the back-emf from the permanent magnets λ_{pm} is no longer dependant on the velocity,

and because the model was already developed in the previous project (Christensen and Weber, 2014).

$$u_d = R_s i_d + L_d \frac{di_d}{dt} - \omega_e L_q i_q \tag{2.29}$$

$$u_q = R_s i_q + L_q \frac{di_q}{dt} + \omega_e (L_d i_d + \lambda_{pm}) \tag{2.30}$$

The governing equations are shown in equation (2.29), and equation (2.30). The EEC for the motor is shown in figure 2.37.

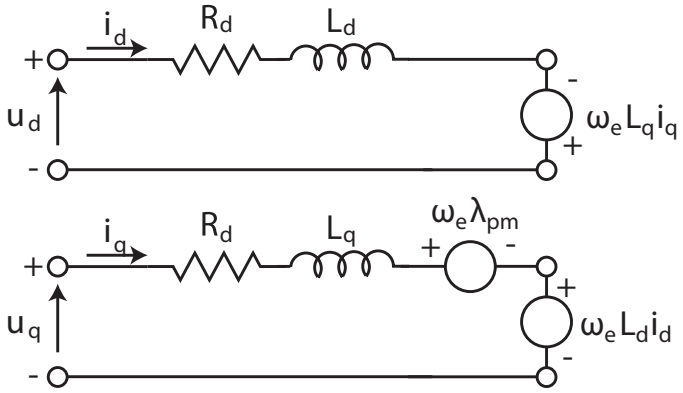


Figure 2.37: EEC of the motor in dq-reference frame.

The model is verified using the same technique as with the sine wave filter. A postive 20% duty cycle is applied on phase U, relative to phase V and W, as a step response and the resulting current and voltage is measured. The measured voltage after the filter on phase U is shown in figure 2.38, and the measured current is shown in figure 2.39.

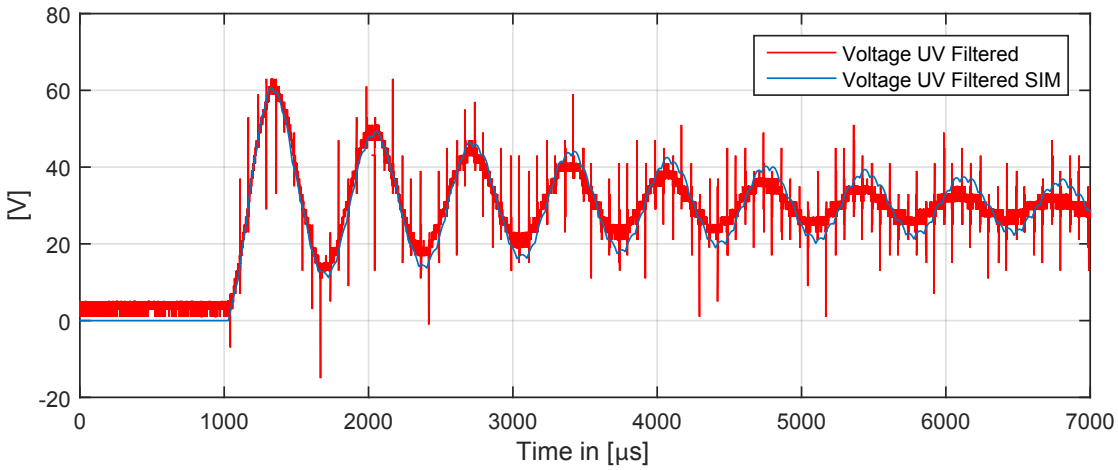


Figure 2.38: Filtered voltage after the sine wave filter, before the motor. 20% positive duty cycle on phase U relative to phase V.

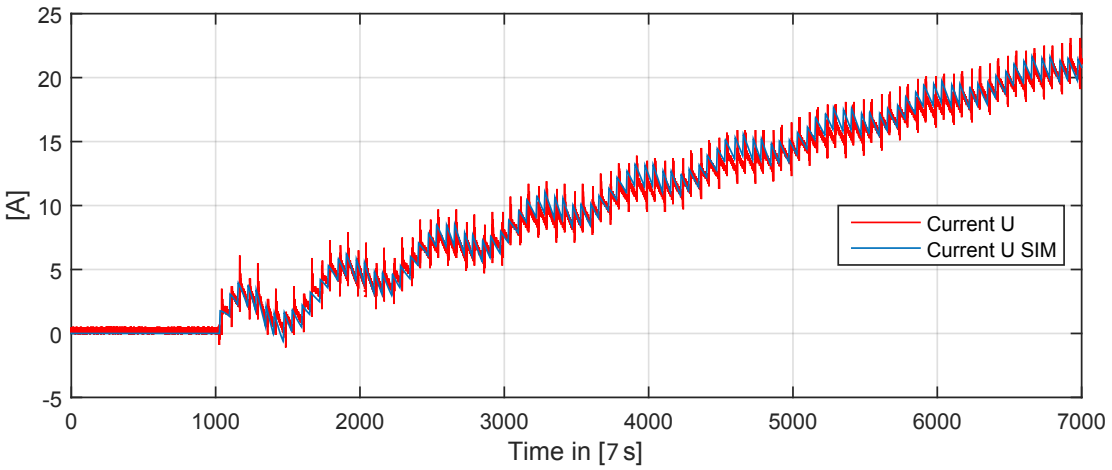


Figure 2.39: Current on phase U. 20% positive duty cycle on phase U relative to phase V.

In the model all the measured values are used together with the data sheet values from the VLT. For the motor the measured values for the resistance and inductance are converted to dq-equivalent values and the rest of the motor parameters are from the data sheet. The model is slightly less damped than the real system. The model is used throughout the report when data is labeled as simulated data. The model is also utilized to determine sensible test parameters for the input signal for the parameter identification.

Chapter 3

Parameter Identification Method

In parameter identification the challenge is to determine the system parameters based on the system response. The general approach is inspired from (Szczupak and Pacas, 2007a) and the strategy is illustrated in figure 3.1.

In general the characteristics of a system may be determined by examining the system response based on different inputs, like impulse, step, ramp, or sinusoidal inputs. In this project it is chosen to use a Pseudo Random Binary Signal (PRBS) to generate the exciting PWM. Voltages and current in the system are measured and used to estimate the systems frequency response. The frequency response is estimated from the Power Spectral Density (PSD) and Cross Power Spectral Density (CPSD) of the system measurements. An analytical expression of the frequency response is curve fitted, to match the estimated frequency response, using an optimization algorithm from which the systems parameters are identified.

In the case of this project it is necessary to choose a method of parameter identification that determines the parameters efficiently, since it has to run as a stand-alone on the sbRIO.

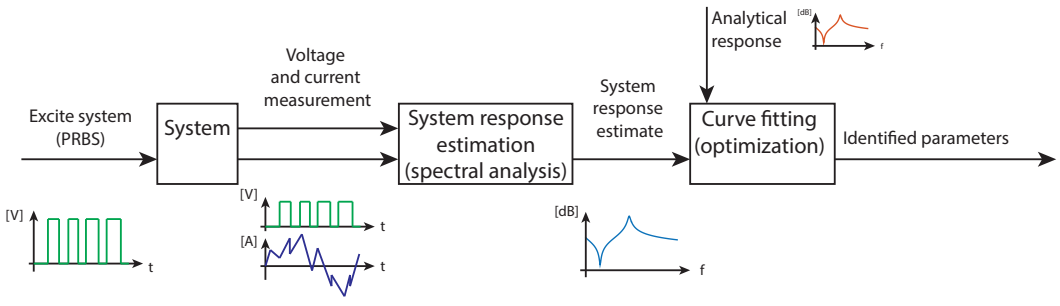


Figure 3.1: Overview of the parameter identification strategy.

The parameter identification is described in the following.

3.1 Parameter Identification Strategy

In the following is described how the parameter identification is utilized in this project. Each step is described in general terms in this section, while further description of each step is described in the following sections of this chapter.

System excitation:
The first step is to excite the system in order to achive a system response. The system is excited with PWM from the VLT output terminals, where the PWM output voltages at terminal *V* and *W* are equal, with a duty cycle of 50%, and the PWM output at terminal *U* is a PRBS based PWM voltage, as shown in figure 3.2.

The PRBS based PWM is chosen because the system needs to be excited at a large range of frequencies in order to accurately describe the frequency response of the system. If the system is excited with a signal at a frequency near the resonance frequency, where the response has a large gain, the current of the system will be high and will possibly damage the system. In order to excite the system at a large range of frequencies the PRBS based signal with white noise characteristics is utilized.

The signal is described further in section 3.3 "*Pseudo Random Binary Signal*", where it is also shown how the signal is improved to better resemble white noise.

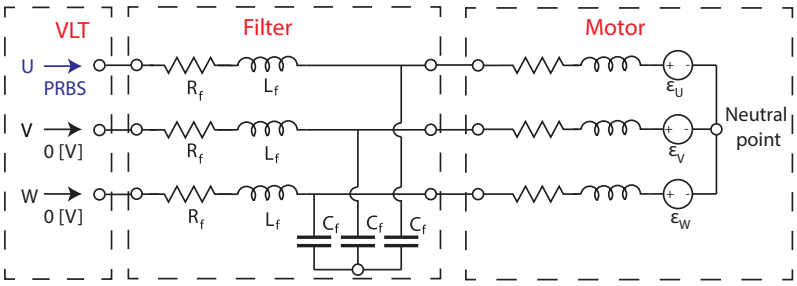


Figure 3.2: Equivalent electrical circuit of the system with PRBS on phase U and 0 voltage signal on phase V and W.

Spectral analysis:
The input voltage is measured at terminal *U* relative to the voltage at terminal *V* and the output current is measured at terminal *U*. The two measurements are used to determine the PSD and CPSD from which the frequency response of the system is estimated. The estimated frequency response will have clear antiresonance- and resonance-peaks as shown in figure 3.3.

More in depth description of the system estimate and spectral analysis methods are found in section 3.2 "*System Frequency Response Estimation*".

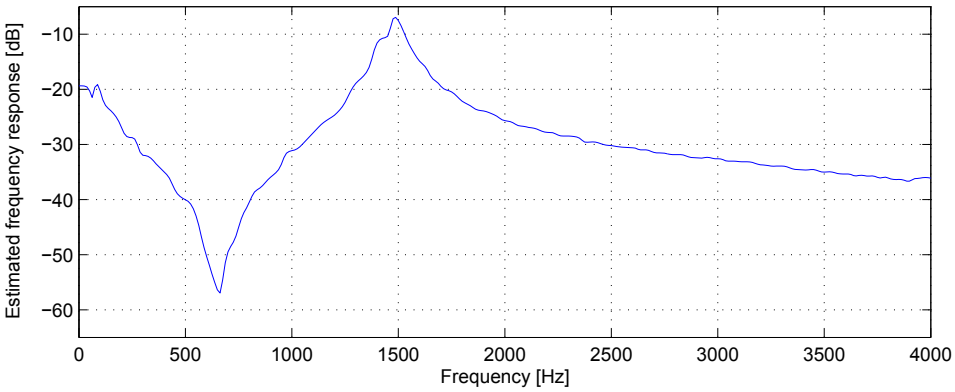


Figure 3.3: Estimated frequency response.

Curve fitting
In order to determine the system parameters an analytical representation of system response is derived. The parameters of the analytical expression are fitted such that the analytical response corresponds to the estimated response. The curve fitting is done numerically with optimization algorithms, which are described further in section 3.4 "*Model Parameter Optimization*".

3.2 System Frequency Response Estimation

In the following is described briefly the generality of discrete Fourier transform, PSD estimate, and system estimation.

3.2.1 Fourier Transform and Signal Spectrum

In order to estimate the system from discrete data the Fast Fourier Transform (FFT) is used as a Discrete Fourier Transform (DFT) to determine the PSD. The DFT, equation (3.2) (Cooley and Tukey, 1964), is based on the Fourier transform of equation (3.1). The DFT is used to determine the frequency spectrum of a signal, $f(x)$, sampled with a fixed sample frequency, $f(x_k)$, $k = 0, \cdots, N-1$, where N is the total number of samples. The FFT is simply the DFT broken into smaller problems for faster computation (Kreyszig, 2011).

$$F(\omega) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} f(x) e^{-i\omega x} dx \tag{3.1}$$

$$F(j) = \sum_{k=0}^{N-1} f(x_k) W^{jk} \quad , \quad W = e^{2\pi i/N} \quad , \quad j = 0, \cdots, N-1 \tag{3.2}$$

3.2.2 Power Spectral Density Estimation

In the following is presented an overview of methods available for PSD estimation, and it is discussed which to use in this project.

Parametric methods calculate the PSD from the output of an assumed linear system of a desired order, which is excited by white noise. Parametric methods result in solving systems of linear equations, resulting in relatively smooth PSD plots, however, mis-specification of system order might lead to errors.

Non-parametric methods estimate the PSD directly from the signal itself, which remove the uncertainty of assumed system order, however, this method usually result in less smooth PSD results.

When calculating the CPSD it should be noted, that using parametric methods is relatively more complex and computational expensive than non-parametric methods, why non-parametric methods are preferred in this project (DSPrelated, 2007).

Examples of non-parametric methods are Periodogram, Welch's, and Multitaper. The three non-parametric methods are used to estimate the PSD of a voltage measurement signal from the previous project with the same setup (Christensen and Weber, 2014), the estimates are shown in figure 3.4 for comparison.

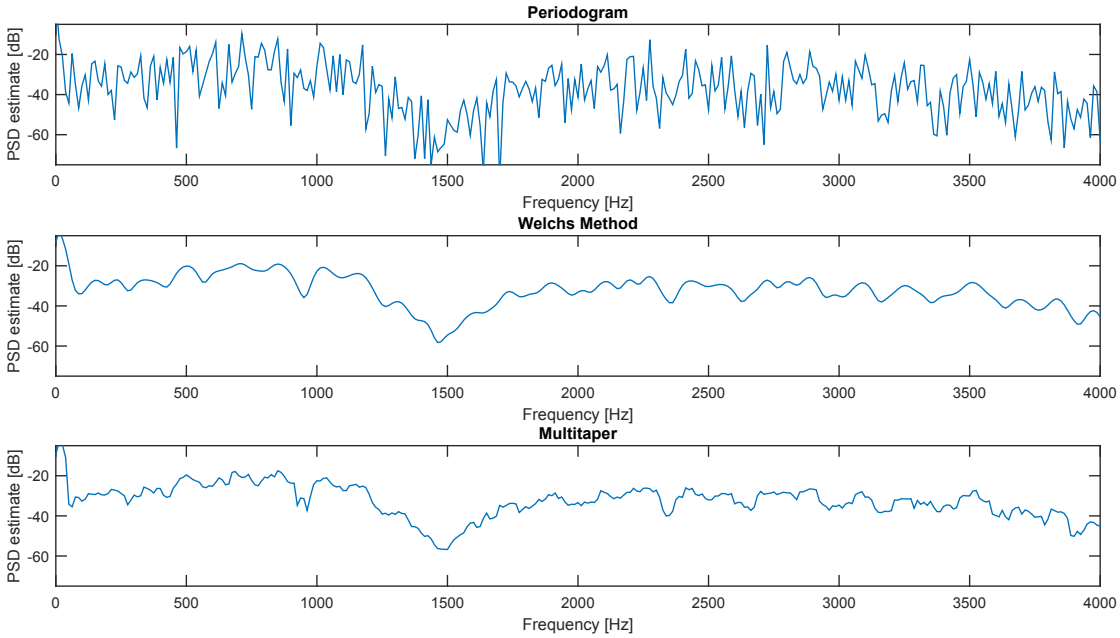


Figure 3.4: Comparison between the three methods to estimate the power spectral density of a voltage measurement; Periodogram, Welch's Method, and Multitaper Method.

In this project Welch's Method is chosen since Welch's Method seems to compute a smoother result than the Multitaper Method, because the multitaper method is more computational expensive, and because Welch's is know to be sufficient for this project, based on experience.

Welch's Method for estimating the PSD is based on computing the FFT of windowed sections of the sampled signal (Welch, 1967). The method is described below:

The signal is sectioned in K sections of length L and is windowed with a window function W such as a hamming. The Fourier transform is calculated for each windowed section in table 3.3.

$$A_k(n) = \frac{1}{L}\sum_{j=0}^{L-1} f_k(j)W(j)e^{-2kijn/L} \tag{3.3}$$

The power spectrum estimate, $\hat{P}_{xx}$, is the average of the modified periodograms, I_k, of equation (3.4).

$$\begin{aligned} I_k(f_n) &= \frac{L}{U}|A_k(n)|^2 \ , \quad k = 1,2,\cdots,K \\ f_n &= \frac{n}{L} \ , \quad n = 0,\cdots,\frac{L}{2} \\ U &= \frac{1}{L}\sum_{j=0}^{L-1} W^2(j) \end{aligned} \tag{3.4}$$

$$\hat{P}_{xx}(f_n) = \frac{1}{K}\sum_{k=1}^{K} I_k(f_n) \tag{3.5}$$

The Welch's PSD estimate of equation (3.5) is a common and efficient way of estimating a signals PSD (Duhamel and Vetterli, 1990). For this project the window function is choosen to be the hamming window, illustrated in figure 3.5, since this is known to yield sufficient results in PSD estimation using Welch's Method.

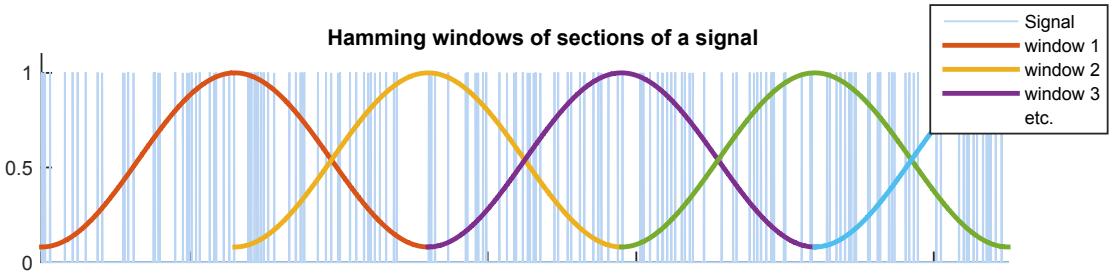


Figure 3.5: Signal with the hamming windows that overlap by 50%.

3.2.3 Frequency Response Estimation

The systems frequency response, $T_{xy}(\omega)$, can be estimated based on the input and output data, where the transfer function is the relationship between the cross power spectral density (CPSD) of input/output, $P_{yx}(\omega)$, and the PSD of the input, $P_{xx}(\omega)$, as in equation (3.6) (Keesman, 2011).

$$T_{xy}(\omega) = \frac{P_{yx}(\omega)}{P_{xx}(\omega)} \tag{3.6}$$

The CPSD and cross periodogram, $I(f_n g_n)$, of the signal samples $f(x)$ and $g(y)$ is estimated in equation (3.7), where $A_x(n)$ and $A_y(n)$ is the Fourier transform of each of the signals, as shown in (Cooley et al., 1969).

$$\begin{aligned} \hat{P}_{xy}(f_n g_n) &= \frac{1}{K} \sum_{k=1}^{K} I_k(f_n g_n) \\ I_k(f_n g_n) &= \frac{N}{2\pi} A_x(n) A_y(n) \end{aligned} \tag{3.7}$$

3.3 Pseudo Random Binary Signal

As mentioned it is important that the exciting PWM resembles white noise, where white noise is characterized by a random signal with a constant PSD. In the following a few steps are taken to come up with a signal that have a constant PSD where the Pseudo Random Binary Signal (PRBS) strategy of (Szczupak and Pacas, 2007b) is used as inspiration.

The PRBS in the following should be considered as the positive reference UP signal used to control the inverter IGBTs. For convenience the PRBS signal is investigated for white noise characteristics rather than the voltage U, however, for completeness the FFT of the system excitation is investigated in the end of this chapter. The FFT of white noise is plotted in figure 3.6 where the constant amplitude is clear.

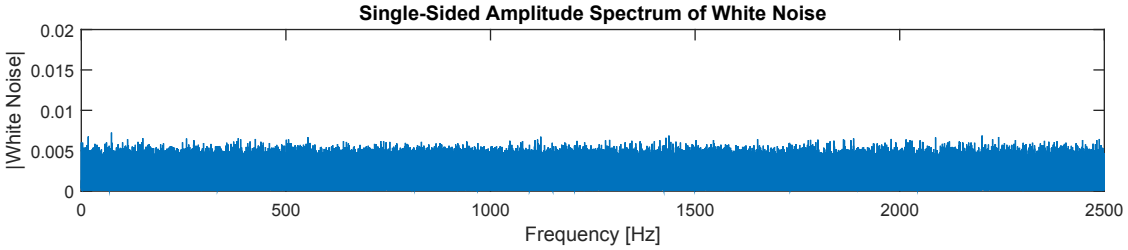


Figure 3.6: FFT of a white noise signal.

A random sequence of binary values is generated in matlab, using the *round(rand())* command to create an array of random binary values. The sequence is used to generate a simulated signal UP with a switching frequency of $5kHz$ and where the n'th entries of the random array defines wether the n'th period is high or low.

The PRBS in the following are considered to be sampled signals and total numbers of samples have been chosen at 1.3×10^6 samples. The simulated sampling frequency is chosen to $31.25MHz$ for comparison, since this is also the sampling frequency of the data from the previous project, used in section 3.2 "*System Frequency Response Estimation*", which corresponds to a total time of $0.0416s$.

A PRBS signal, y_{PRBS}, is generated as described above and the full sequence is shown in figure 3.7, which is the basis of this section. The same sequence of binary values will be used throughout most of this section for better comparison.

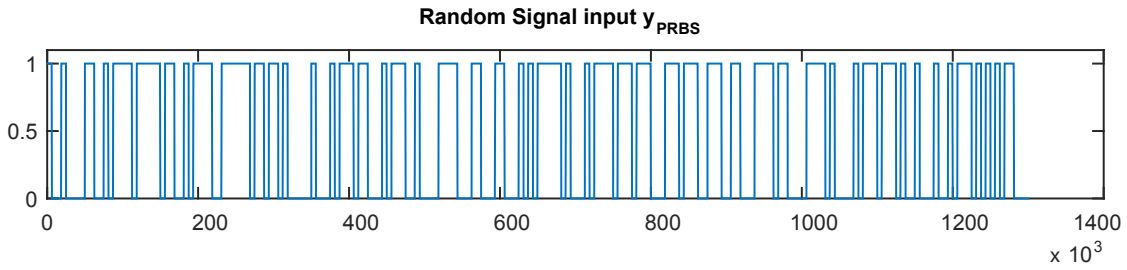


Figure 3.7: Binary signal, based on a sequence of randomly chosen binary values.

A cutout of the random signal y_{PRBS} is shown in figure 3.8 together with the FFT of the signal. The FFT amplitude plot shows a non constant FFT from which it is clear that y_{PRBS} does not resemble white noise.

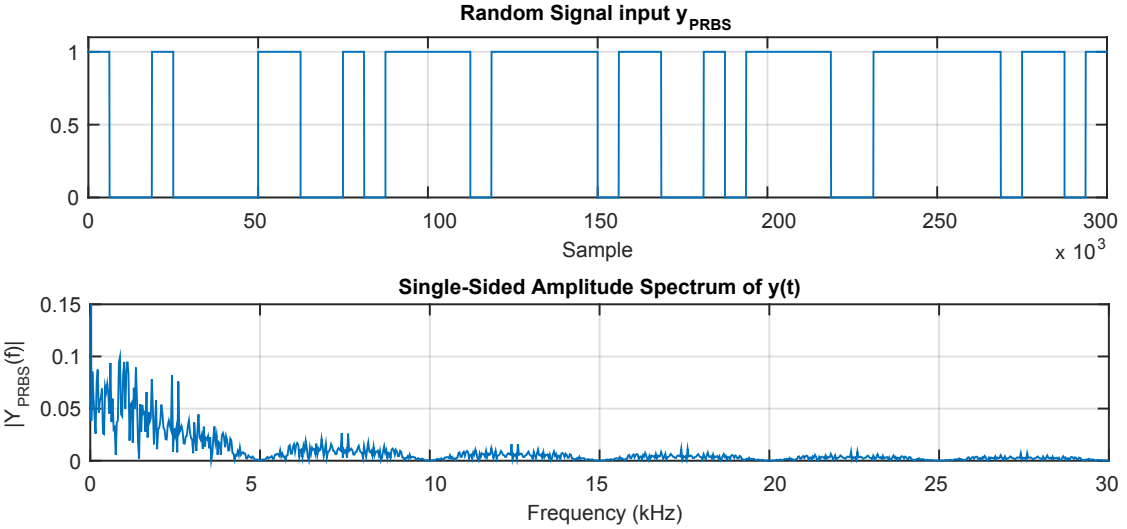


Figure 3.8: A cutout of the PRBS together with a plot of its FFT.

The general PRBS signal need to be modified for it to better resemble white noise. A few methods are available to modify the signal. One method is to change the duty cycle of the binary signal, such that the periods with the binary value 1 only applies to a defined fraction of the switching period. Another method is to randomly change the period length. The two methods are evaluated in the following.

Adding a predefined duty cycle:
A duty cycle of 50% is added to the same random sequence of binary numbers as seen from figure 3.7 and the modified signal y_{Duty} is shown in figure 3.9 together with the FFT of the signal Y_{Duty}.

It is clear that the spectrum is not constant,and it has large components at $5kHz$, $15kHz$, etc. It is concluded that adding a duty cycle only changes the periodics of the signal, instead of eliminating them. It is obvious that the large spikes are due to the switching frequency of $5kHz$ and adding a duty cycle of 50% only makes the switching frequency stand out more.

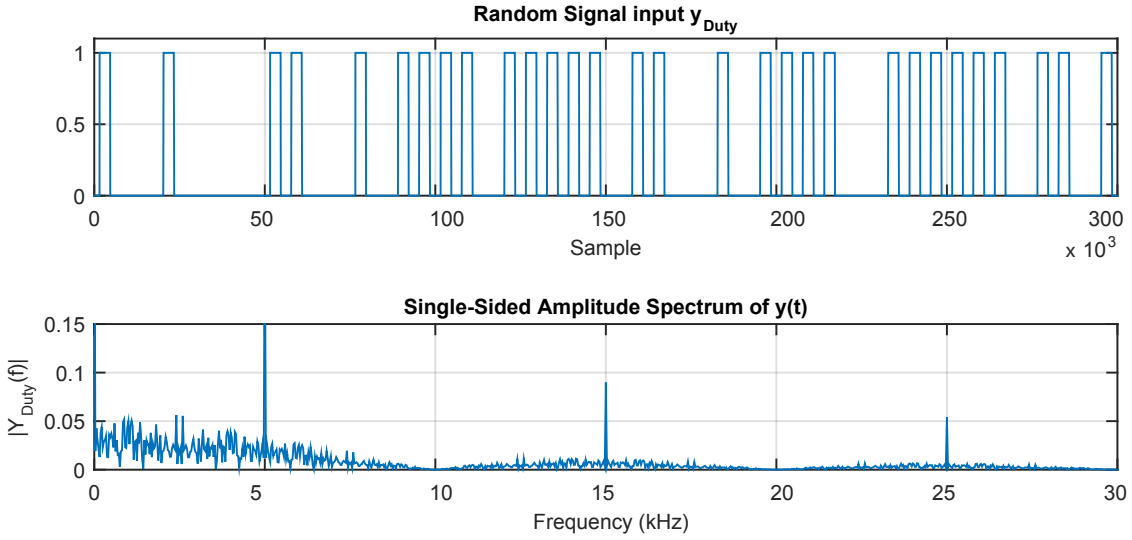


Figure 3.9: A cutout of the PRBS with duty cycle, together with a plot of its FFT.

In order to reduce the harmonics due to the switching frequency it is sought to change the switching frequency in order to remove these harmonics.

Variable Period:
The signal from figure 3.7 is modified such that the sequence of random binary values are the same but instead of a fixed switching frequency, the switching period is random. Another set of random values are generated, ranging from 0-1, these values are used to determine the size of each period. The size of the period is determined with a bias of 6250 points per period, which corresponds to the value 0.5 in the sequence of random values. The range of the period is chosen to be 6250, such that 0 = 3125 and 1 = 9375 which correspond to a range from 100-400μs and a frequency range of 2.5kHz-10.

The signal with variable period is shown in figure 3.10 together with the FFT of the signal. The use of variable switching frequency greatly reduces the harmonics of the signal. The spectrum clearly show that the harmonics due to the switching frequency

have been eliminated and that the spectrum shows great resemblence to white noise from $3kHz$ and up. However, from DC to $3kHz$ the spectrum shows relatively large gains, which should be reduced. In order to lower the gain at the low frequencies a combination of variable switching frequency and adding a duty cycle to the signal is applied.

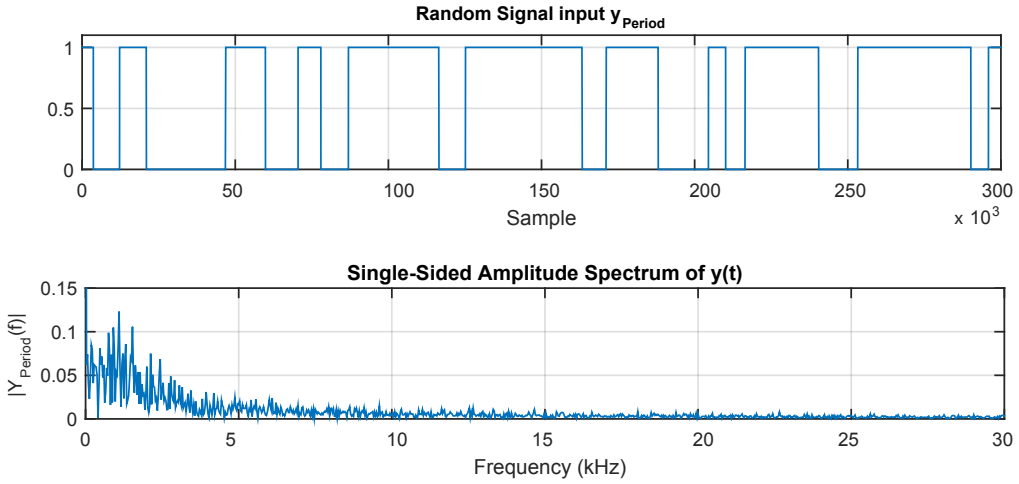


Figure 3.10: A cutout of the PRBS with variable switching frequency, together with a plot of its FFT.

Duty Cycle and Variable Period:
The idea of applying a duty cycle to the PRBS is used on the signal with a variable switching period. The signal is shown in figure 3.11 together with the FFT of the signal. It is clear that the combination of both a duty cycle of 0.5 and a variable switching period greatly improves the resemblance to white noise.

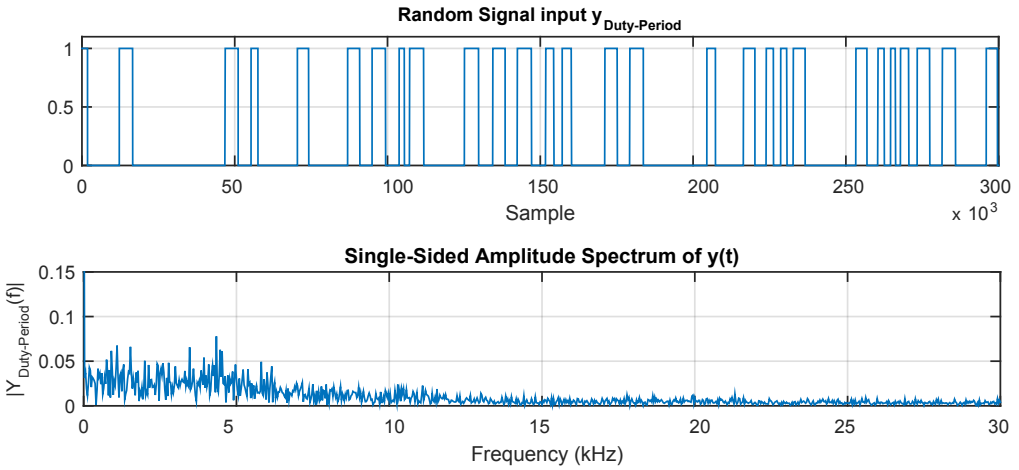


Figure 3.11: A cutout of the PRBS with variable switching frequency and duty cycle, together with a plot of its FFT.

The gain at the low frequencies have been reduced and spectrum is relatively constant, though with a higher amplitude in the low frequencies. The signal with a combination of variable switching frequency and duty cycle is assessed to resemble white noise sufficiently for the purpose of applying the signal to the filter and motor system in order to get a great system response. The effects of different duty cycles and variations of switching frequency have been examined and the results are shown in the following.

Effect of different duty cycles:
From figure 3.12 it is seen how the spectrum amplitudes increase in the low frequencies when the duty cycle is increased. From this observation it should be noted that the duty cycle should be kept small in order to reduce the variation of the spectrum in the low frequencies.

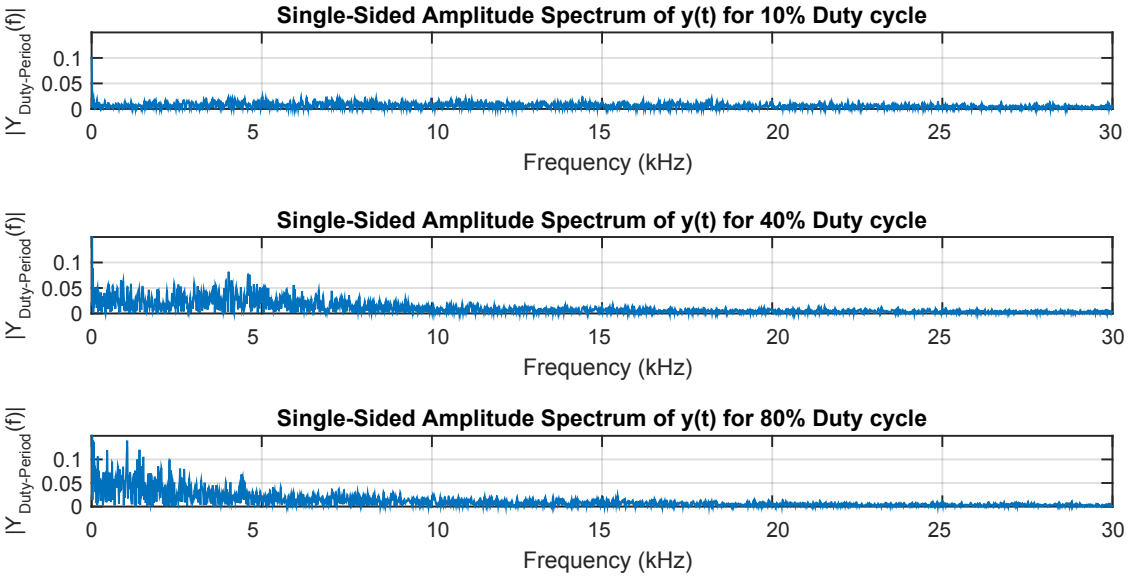


Figure 3.12: FFT of the three different cases used to investigate the effects of different duty cycles. The signals use simulated sample time of $3.2\mu s$ and the switching frequency differs between $2.5kHz$-$7.5kHz$.

Effect of different switching frequency schemes:
In the case with the switching frequency, two variables are considered: The band of allowed frequencies and the position of the band. In order to examine the effects of different variable switching frequency schemes, the following three cases are used. Case 1 has a large band of frequencies, Case 2 a small band but is located at high frequencies, and last Case 3 also with a small band but located at low frequencies.

Case 1: Band of $5kHz$ centered around $7.5kHz$.
Case 2: Band of $2kHz$ centered around $10kHz$.
Case 3: Band of $2kHz$ centered around $5kHz$.

From figure 3.13 it can be seen that the spectrum from the three cases are rather similar, however, it is clear that case 1 is preferable with a lower amplitude in general while case 2 and case 3 show a relatively small increase in amplitude at low frequencies. Since these results are close to each other, a few more simulations were utilized with different random vectors and the power of the signal were calculated utilizing Parseval's Theorem, which calculates the signals RMS value. The results are shown in table 3.1.

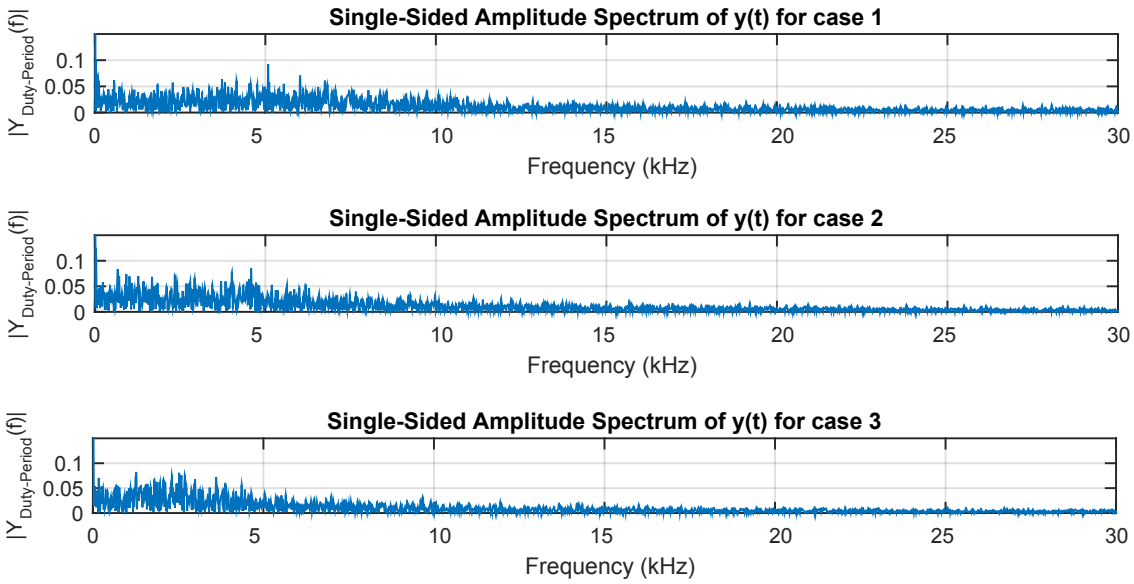


Figure 3.13: FFT of the three different cases used to investigate the effects of different variable switching frequency schemes. The schemes are utilized with a duty cycle of 40% and a simulated sample time of 3.2μs.

RMS values of PSD data:		
Case 1	Duty 40%	120.95
	Duty 10%	35.20
Case 2	Duty 40%	165.98
	Duty 10%	49.85
Case 3	Duty 40%	161.82
	Duty 10%	49.07

Table 3.1: Calculated power of the FFT data ranging from 0-5000Hz.

Based on the above observations it is save to say that a PRBS signal with a small duty cycle together with a variable switching frequency within a large band of frequencies is sufficient for the needs of this project. It is also necessary to look at the inputs on phase V and W of the system in order to evaluate the system excitation entirely, which is discussed in the following.

System excitation:
The developed PRBS is used to excite the simulated system and the FFT of the system output voltage, U relative to V is used to calculate the FFT. The PRBS is applied on phase U and a 0 voltage PWM is applied to phase V and W which correspond to a duty cycle of 50%. The switching of all three phases are synchronized, why it is obvious that the system excitation is, as shown in figure 3.14, the difference between the two signals. The duty cycles of figure 3.14 are 50% for V and W and 75% for the PRBS. To fully understand the system excitation it is advantageous to study figure 3.2 in section 3.1 "*Parameter Identification Strategy*".

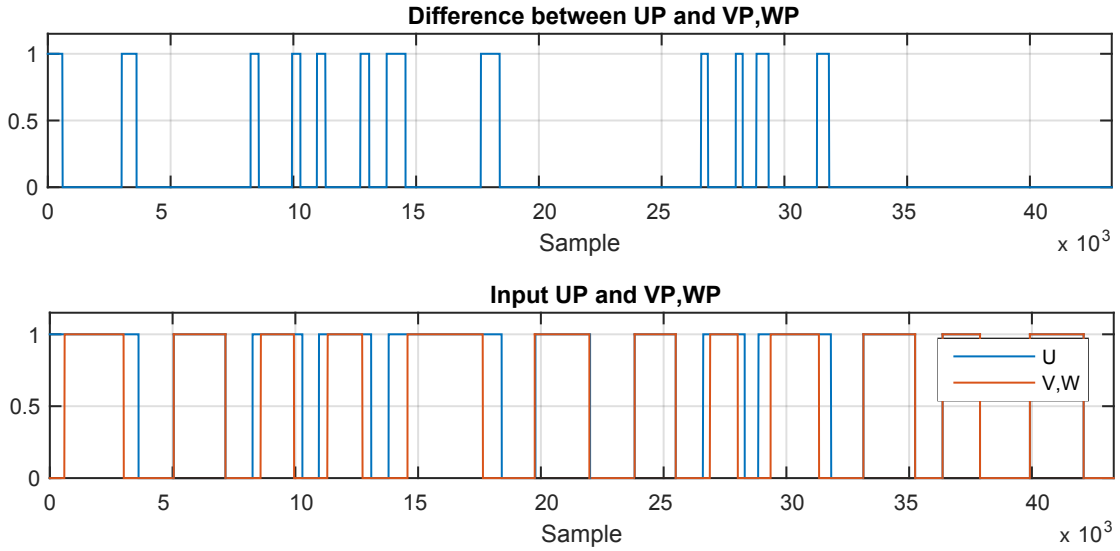


Figure 3.14: The simulated system inputs.

The FFT of the simulated input signals is shown in figure 3.14 and the FFT of the simulated systems voltage at phase U relative to phase V is shown in figure 3.16. The relative higher spectrum aplitude is due to the fact that the PRBS is only of magnitude 1 while the system voltages is 560[V], however, the input is assessed to be sufficiently resembling white noise.

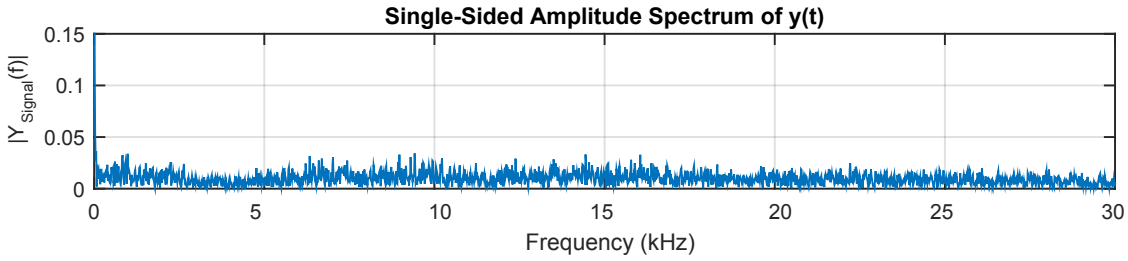


Figure 3.15: FFT of the simulation input signals - the difference between UP and VP.

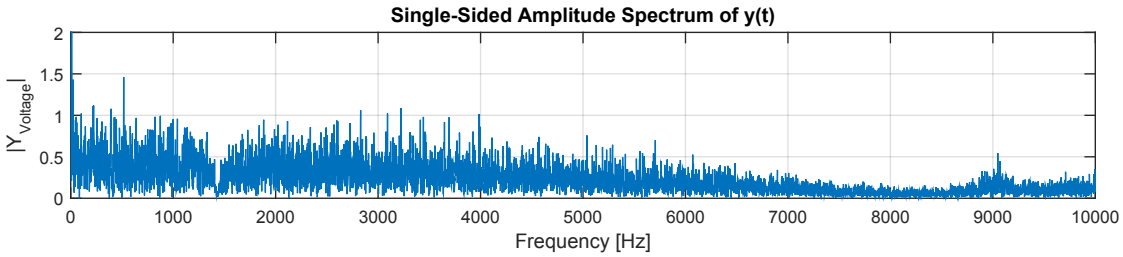


Figure 3.16: FFT of simulated system input voltage.

3.4 Model Parameter Optimization

In the following, two optimizations strategies are discussed for curve fitting, the first is Levenberg-Marquardt and the second is the particle swarm optimization. Both strategies are implemented in MATLAB and evaluated for similar conditions in order to compare the two. Firstly the analytical expression used for curve fitting is derived.

3.4.1 Derivation of System Admittance

The system is configured as shown in figure 3.17 and the system is excited with the PRBS based PWM on terminal U. The input of the system is the voltage at U relative to V or W and the output is the current at U. The system is seen as a connection of admittances since the magnitude of the admittance can be described as $|Y| = \frac{|I|}{|V|}$.

Figure 3.17 shows the EEC of the system and an overview of how the admittance system is derived. In the derivation it is assumed that the filter and motor parameters are equal for the three phases and that the two neutral points can be assumed to have the same voltage potential. It should also be noted that EMF of the motor is assumed zero since the method is utilized at stand still.

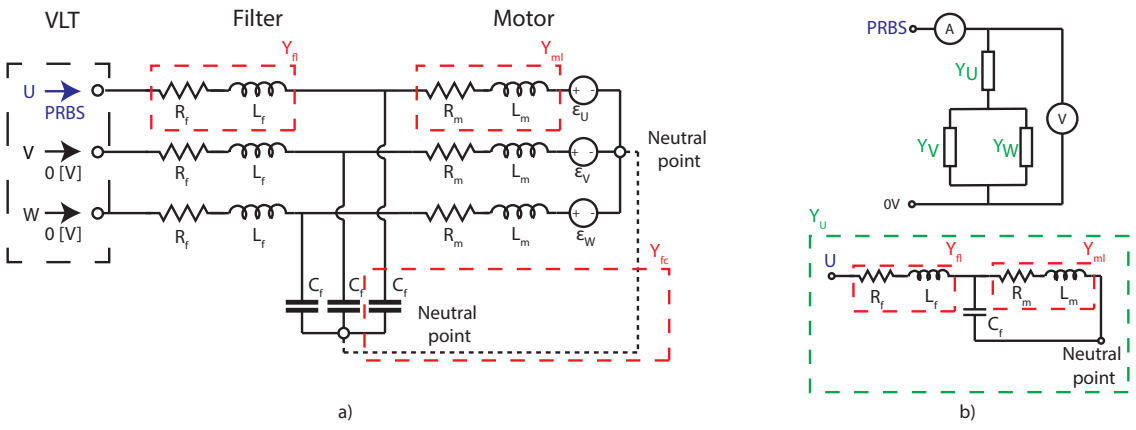


Figure 3.17: a) EEC of the system, b) system of admittance.

The admittance of each phase in the system is modelled as shown for Y_U in figure 3.17.b and equation (3.8).

$$\begin{aligned} Y_U(j\omega) = Y_V(j\omega) = Y_W(j\omega) &= \frac{Y_{ml}Y_{fl} + Y_{fc}Y_{fl}}{Y_{ml} + Y_{fc} + Y_{fl}} \\ Y_{fc} = j\omega C_f \ , \quad Y_{fl} = \frac{1}{R_f + j\omega L_f} \ , \quad Y_{ml} &= \frac{1}{R_m + j\omega L_m} \end{aligned} \tag{3.8}$$

Since the parameters of the three phases are equal, the admittances of V are W equals that of U, why equation (3.9) is true and writing out this expression leads to the magnitude of the system admittance to be that of equation (3.10).

$$Y(j\omega) = \frac{1}{\frac{1}{Y_U(j\omega)} + \frac{1}{2Y_U(j\omega)}} = \frac{1}{\frac{2}{2Y_U(j\omega)} + \frac{1}{2Y_U(j\omega)}} = \frac{2}{3}Y_U(j\omega) \tag{3.9}$$

$$|Y_{(j\omega)}| = \left| \frac{2}{3} \cdot \frac{(j\omega)^2 a_2 + j\omega a_1 + 1}{(j\omega)^3 b_3 + (j\omega)^2 b_2 + j\omega b_1 + b_0} \right| \tag{3.10}$$

Where:

$$\begin{aligned} a_1 &= R_m C_f & a_2 &= L_m C_f \\ b_0 &= R_m + R_f & b_1 &= L_m + L_f + R_m R_f C_f \\ b_2 &= R_m L_f C_f + R_f L_m C_f & b_3 &= L_m L_f C_f \end{aligned}$$

3.4.2 Levenberg-Marquardt Damped Newton Method

For curve fitting of non-linear models the Levenberg-Marquardt optimization algorithm is know to yield good results, which is also concluded in (Szczupak and Pacas, 2007a). Levenberg-Marquardt is based on Newton's method with a damped newton step, where the damping coefficient λ is self regulative. The cost function is determined from the least square of a residual and the Jacobian and the Hessian of the residual is used to determine step size and direction of the algorithm. The cost function $f(\mathbf{x})$, step $\mathbf{s}$, and Hessian estimate $\mathbf{H}$ is shown in equation (3.11) and the residual function in equation (3.12) where Y_{exp} is the experimental data of the system and $Y_{model}(\mathbf{x})$ the modelled system from the parameters $\mathbf{x}$ as a vector (Endelt, 2011).

$$\begin{aligned} f(\mathbf{x}) &= \frac{1}{2}\mathbf{r}(\mathbf{x})^T \mathbf{r}(\mathbf{x}) \\ \mathbf{s} &= -(\mathbf{H} + \lambda \mathbf{I})^{-1} \nabla f(\mathbf{x}) \\ \nabla f(\mathbf{x}) &= \mathbf{J}(\mathbf{x})^T \mathbf{r}(\mathbf{x}) \\ \mathbf{H} &= \mathbf{J}(\mathbf{x})^T \mathbf{J}(\mathbf{x}) \end{aligned} \tag{3.11}$$

$$\mathbf{r}(\mathbf{x}) = \frac{Y_{exp}}{Y_{model}(\mathbf{x})} - 1 \tag{3.12}$$

The cost function, $f(\mathbf{x})$, is the least square of the residual, where the residual, $\mathbf{r}(\mathbf{x})$, is the difference between the system estimated based on experimental data, Y_{exp}, and the modelled system, Y_{model}.

The Levenberg-Marquardt optimization were implemented in MATLAB with analytical expressions for the residual, the jacobian, and the hessian, based on the admittance of equation (3.10). The experimental data used is that of previous work and the results are illustrated in figure 3.18 with conditions listed in table 3.2 and the script is written in appendix A.1 "*Levenberg-Marquardt Algorithm*".

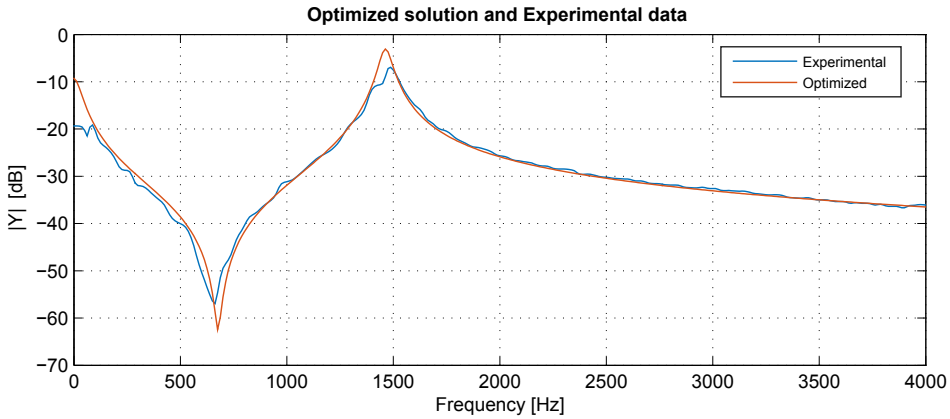


Figure 3.18: Optimized result of the Levenberg-Marquardt with the experimental data.

Levenberg-Marquardt:		
Initial Guess:		
R_m	0.3	$[\Omega]$
L_m	5.5	$[mH]$
R_f	0.1	$[\Omega]$
L_f	1.1	$[mH]$
C_f	14.7	$[\mu F]$

Table 3.2: Initial guess used in Levenberg-Marquardt.

The use of symbolic expressions in MATLAB is very computational expensive, why the algorithm should be modified to use numerical differentiation in order to determine the Jacobian and the Hessian. Also the fact that the algorithm has to be implemented onto the sbRIO means that non-analytical algorithms are needed in order to make the algorithm more computational effective.

Numerical differentiation is implemented as a first order Taylor series, also an initial step is chosen in order to utilize numerical differentiation for the first iteration of the algorithm. By implementing numerical integration, the algorithm tends to explode

and rarely yield a sufficient result, even with a starting guess close to the solution. The damping parameter λ is iteratively increased, which reduces the step size, in order to search for a step that reduces the cost function. It is assessed that the estimates of the Jacobian and the Hessian are not sufficient for this non-linear system and more precise estimates must be calculated to yield a sufficient solution.

Due to the problems with the numerical differentiation it is sought to use an optimization algorithm without the use of differentiation, which lead to the use of particle swarm optimization.

3.4.3 Particle Swarm Optimization

It is chosen to use a standard PSO algorithm, as proof of concept, and compare it with Levenberg-Marquardt. The algorithm is implemented in MATLAB, as described in (Arora, 2012). In this case the cost function of equation (3.13) is the least square of residual, similar to the Levenberg-Marquardt algorithm.

$$f(\mathbf{x}) = \frac{1}{2}\mathbf{r}(\mathbf{x})^T\mathbf{r}(\mathbf{x}) \tag{3.13}$$

Where the residual is calculated as:

$$\mathbf{r}(\mathbf{x}) = \frac{Y_{exp}}{Y_{model}(\mathbf{x})} - 1 \tag{3.14}$$

The idea behind the PSO strategy is inspired from how a flock of birds or a school of fish move. A bird decides not only on its own, but is also affected by the movement of the flock. In PSO, each 'bird' is called a particle, which represents a solution to the system. In this case, a solution, or particle, is a set of the parameters $\mathbf{x} = [R_m\ L_m\ R_f\ L_f\ C_f]^T$. By creating a series of N_p number of particles, one gets a swarm, which is the idea behind PSO, where the swarm is going to search for the optimal solution.

From equation (3.15) and equation (3.16) the primary part of the iterative procedure of PSO is shown.

$$\mathbf{v}_{k+1} = \mathbf{v}_k + c_1 r_1(\mathbf{x}_{P,k} - \mathbf{x}_k) + c_2 r_2(\mathbf{x}_G - \mathbf{x}_k) \tag{3.15}$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{v}_{k+1} \tag{3.16}$$

The vector $\mathbf{v}$ is the velocity of each particle, equivalent to step size of a Newton based algortihm. The vector $\mathbf{x}$ is the current particle position, containing the design points of the particle, $\mathbf{x}_P$ is the particles personal best of all time and $\mathbf{x}_G$ is the swarm's best of all time. The parameters c_1 and c_2 are chosen between 0 and 4, but usually 2, and these are used to determine the cognitive and social behaviour of the particle, respectively. The parameters r_1 and r_2 are random numbers, generated at each iteration.

For every iteration $\mathbf{x}_P$ and $\mathbf{x}_G$ are updated if necessary. The starting guess for each particle is chosen randomly, within predefined boundaries, for each design parameter. The same boundaries are used to limit the particle solution, which is enforced for every iteration. Each iteration is structured as described below:

Step 1 Calculate particle velocity of each particle - equation (3.15)

The inital velocity is choosen to $\mathbf{v}_0 = [0\ 0\ 0\ 0\ 0]$ *while the initial particle positions are chosen randomly, based on the predefined parameter boundaries.*

Step 2 Calculate new particle positions for each particle - equation (3.16)

The new particle position is the summation of the position and the velocity, why the velocity is comparable to a step.

Step 3 Check and enforce bounds

If the design parameter exceeds the predefined parameter boundaries, the parameter is simply set to equal the boundary value.

Step 4 Calculate cost for each particle - equation (3.13) and (3.14)

Calculate the analytical solution of equation (3.10) for each particle and calculate the residual and cost based on this.

Step 5 Update best solution for each particle and for the swarm in total

Compare new and old cost and update x_P *and* x_G *if improved.*

A MATLAB script of the PSO is found in appendix A.2 "*Particle Swarm Optimization - Proof of Concept*". Table 3.3 shows the parameter boundaries and the constants of the algorithm, chosen for this case of proof of concept. N_p is the chosen number of particles in the swarm, which are usually chosen to be between $5n$-$10n$, where n is the number of design variables in $\mathbf{x}$.

Particle Swarm:				
Parameter boundaries:			**Constants:**	
R_m	0.05 - 2	[Ω]	c_1	2
L_m	1.00 - 10	$[mH]$	c_2	2
R_f	0.05 - 2	[Ω]	N_p	35
L_f	0.50 - 5	$[mH]$	n	5
C_f	5.00 - 20	$[\mu F]$		

Table 3.3: Particle Swarm optimization conditions, boundaries and constants.

Figure 3.19 shows the result of an optimization cycle with the parameters as described in table 3.3. The plot of all the inital guesses, figure 3.19(top), clearly show one of the advantages with the PSO. All the initial particles are chosen randomly, why the design solutions are spread all over the design space, which helps to determine a good initial direction, toward the right solution. The best solution plot, figure 3.19(middle), shows

the best solution of each iteration, from which it is clear that the algorithm, gets on the track to the right solution. quite fast The last plot shows the optimal solution, where the cost is minimized. The results are listed in table 3.4

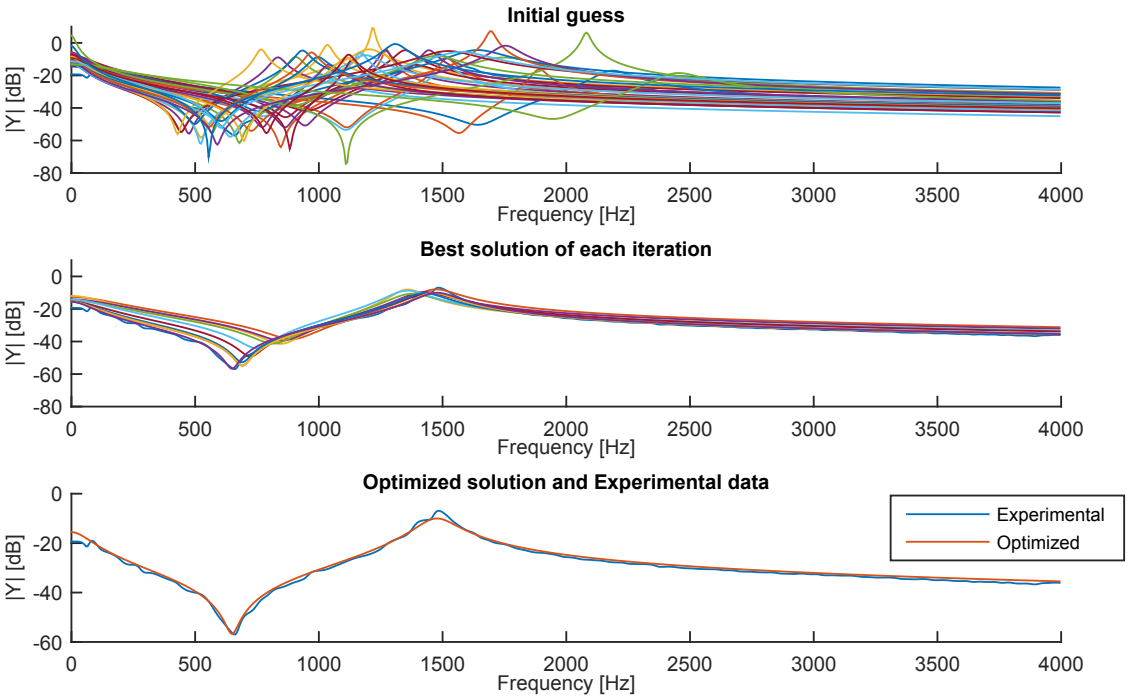


Figure 3.19: PSO based on laboratory data. (top) Shows the responses of all the initial particle solutions, (middle) shows the best solution of each iteration, (bottom) shows the final solution of the PSO.

Particle Swarm - Results:			
Optimized parameters:			**Iterations:**
R_m	2.0	$[\Omega]$	9
L_m	7.4	$[mH]$	**Cost limit:**
R_f	2.0	$[\Omega]$	5
L_f	1.8	$[mH]$	**Iteration-time:**
C_f	8.1	$[\mu F]$	0.03-0.05 $[s]$

Table 3.4: Particle Swarm Optimization results.

Chapter 4

Experimental Setup

With the goal for the project and the system and parameter identification method described, the following chapter describes the experimental setup. This includes a detailed description of the hardware and software designed and developed for the parameter identification, and external hardware necessary for control and data acquisition.

4.1 Experimental Setup

The experimental setup consists of the hardware and software necessary to perform system identification on the filter and motor described in Chapter 2 "*System Models*", using the Danfoss VLT. The experimental setup is based on experience gathered from the previous project "Automatic Parameter Identification for an AC-motor with an LC-filter" (Christensen and Weber, 2014). The main components are meant to be connected as shown in figure 4.1.

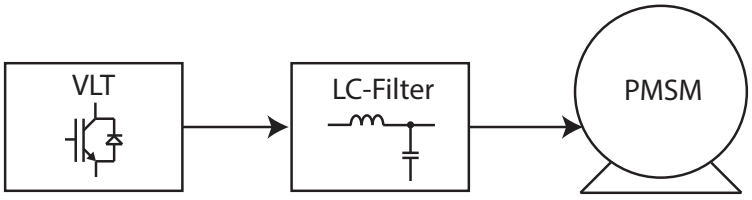


Figure 4.1: Block diagram of the main component connections.

However, modifications and additional hardware is applied in order to perform the parameter identification method, described in Chapter 3 "*Parameter Identification Method*". The Danfoss VLT is equipped with hardware and software to program and control a motor mounted on the output. It is wished, however, to have direct control over the IGBTs in order to execute the PRBS signal. Therefore, the control hardware in the VLT is replaced by an interface board developed at Aalborg University, the IPC3. The IPC3 allows for direct access of the VLT's IGBTs, through external control. The setup will

be controlled with a National Instruments Single Board RIO (sbRIO) running LabVIEW (**?**). It is sought to create an embedded solution, so the parameter identification will be executed on the sbRIO aswell. Therefore the current and phase voltage at the VLT output must be known, in order to estimate the admittance of the system. In "Automatic Parameter Identification for an AC-motor with an LC-filter" it was initially assumed that the voltage at the VLT output could be estimated from the input PWM-signals. However, it proved that dynamics in the system caused voltage spikes that were not included, by estimating it from the input PWM-signals. A possible solution to this problem, is to measure the voltage at the VLT output. Therefore the admittance is estimated by measuring the "input" voltage and the resulting "output current. For the parameter identification method, described in Chapter 3 "*Parameter Identification Method*", it is only necessary to measure one phase for voltage and current. However, as it is sought to create a complete solution, that can also control the motor, the currents and voltages of all three phases are measured.

A block diagram of the experimental setup with the additional components is shown in figure 4.2.

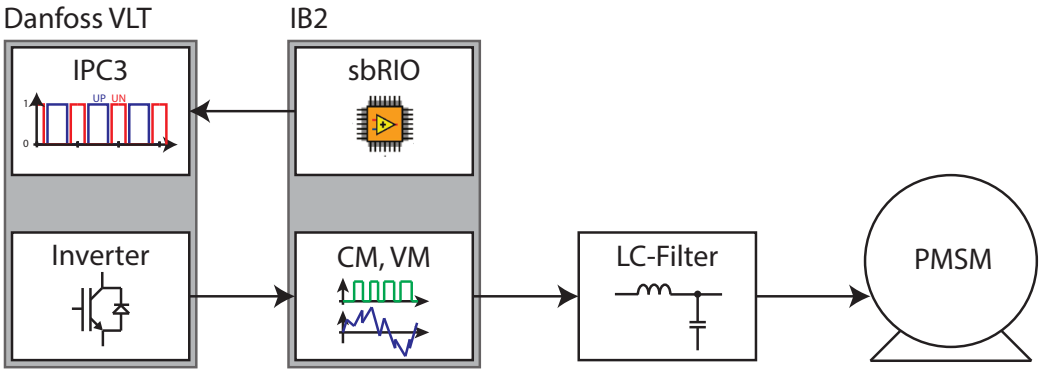


Figure 4.2: Block diagram of the expanded hardware setup for parameter identification. CM refers to current measurement and VM refers to voltage measurement.

The laboratory setup is shown in figure 4.3, with the VLT, custom hardware for measuring the voltages and currents at the VLT output, the LC-filter and the motor.

In the following the control and monitoring system is first described, consisting of the sbRIO and the associated program, followed by the PWM strategy applied to control the IGBTs and a description of the IPC3. Thereafter, the custom hardware, designed for current and voltage measurements and communication between the sbRIO and the IPC3, is described in detail.

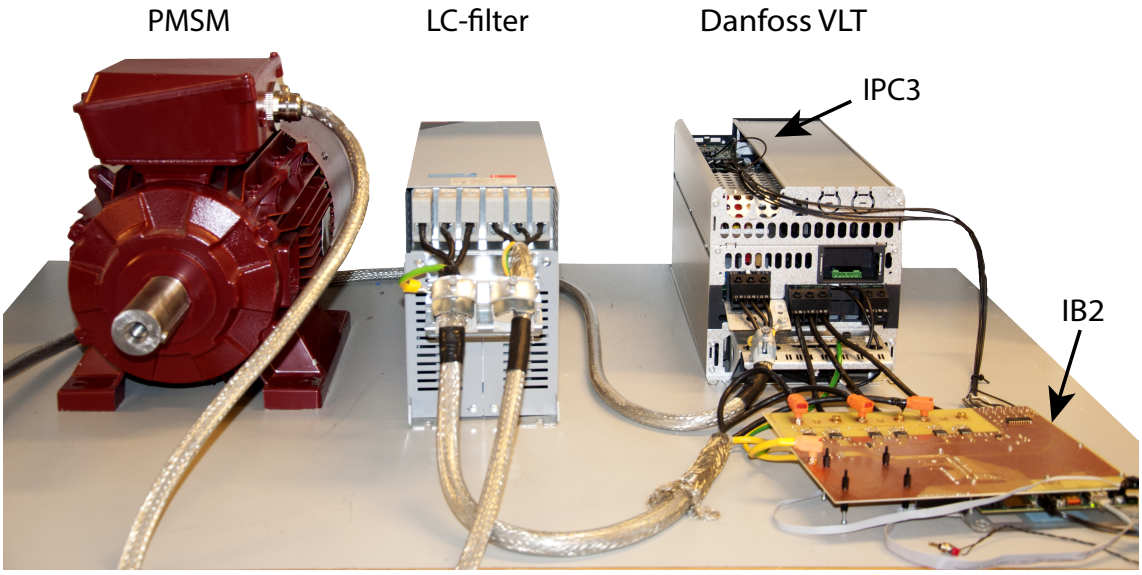


Figure 4.3: Laboratory setup.

4.2 Control and Monitoring System

The hardware used for control and data acquisition is a National Instruments Single Board Reconfigurable I/O (sbRIO), shown in figure 4.4. It is a single board embedded device, containing a Real-Time (RT) processor, a FPGA and digital and analog I/O. The main specifications are shown in table 4.1.

Figure 4.4: sbRIO. (National Instruments - FPGA, 2013)

NI sbRIO-9636
400 MHz Processor
512 MB Memory
256 MB RAM
Xilinx Spartan-6 LX45 FPGA
40 MHz FPGA Top-level Clock
2x 50 pin I/O Header

Table 4.1: Relevant specifications. (National Instruments - FPGA, 2014)

For the parameter identification the main functions are shown in figure 4.5. A signal sequence designed to excite the system is generated and applied on the system using the PWM generator. When the sequence is running, the voltage and current are measured, the data is filtered, and finally scaled from the raw data to physical values. The

PSD and CPSD is used to estimate the frequency response, and a set of parameters are found using the optimization algorithm.

Figure 4.5: The main functions in the parameter identification method.

Each of the main functions includes a number of sub functions, and the whole program is separated onto different parts of the embedded device. Each part of the embedded device is optimized for different tasks. The RT-processor is essential for the solution to work as a stand-alone system. The software is developed on a host-PC, and written to the non-volatile memory of the board. The software is composed of subprograms called Virtual Instruments (VIs), and each VI is made to either run on the Real Time Operating System (RTOS) or on the FPGA. The different functions needed for the system to work as a embedded solution is illustrated in figure 4.6. The host PC is used to program the sbRIO. The code is written in a graphical programming language called 'G'. For the RT VIs the code is compiled to work on the LabVIEW RTOS, and for the FPGA VIs the code is compiled to VHDL, which will be described further in the following FPGA section. When the code is running on the sbRIO, the PC can be used to inspect the signals on the VIs running on the RTOS.

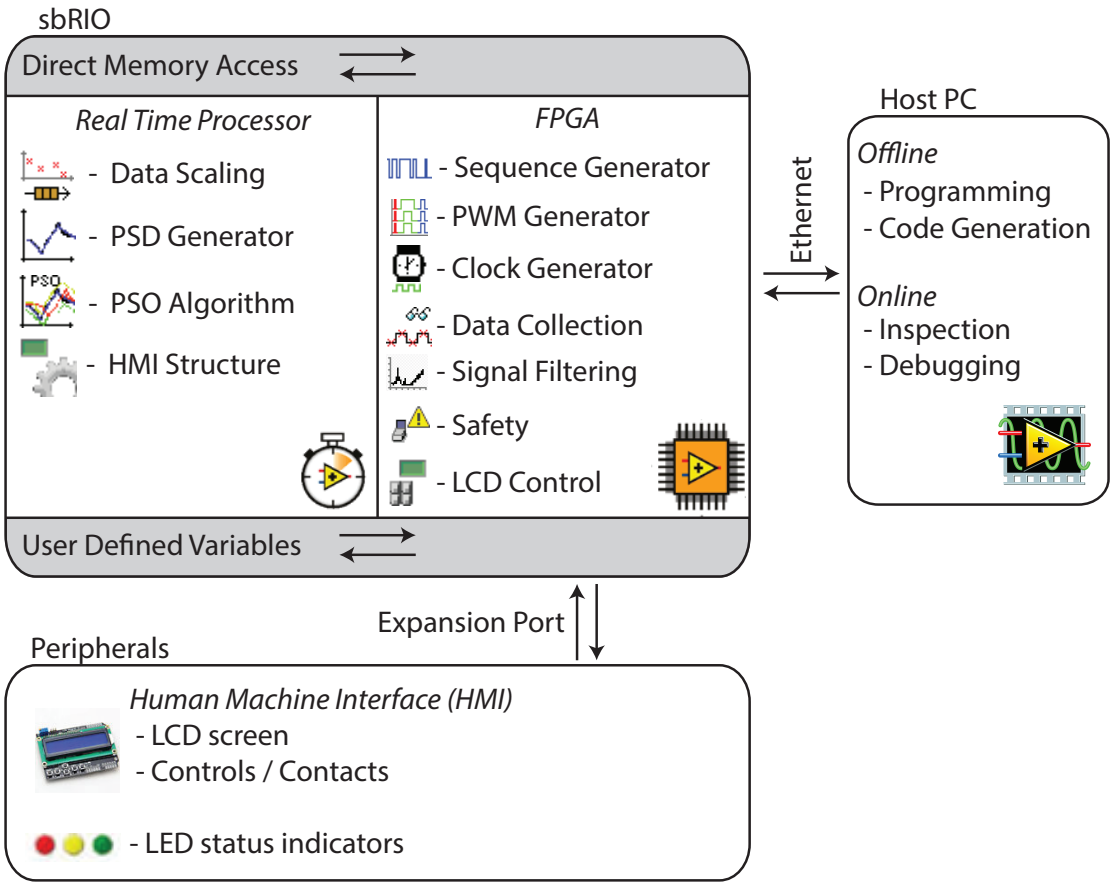


Figure 4.6: Overview of the complete program functions. The host PC communicates with the sbRIO via ethernet, and the peripherals are connected via the expansion port. On the sbRIO the low speed signals (< 100 [Hz]) and controls are using UDV, and the collected data sets are transferred using DMA.

The communication between the RTOS and the FPGA is separated between transfer of collected data and other signals. When transferring the collected data the Direct Memory Access (DMA) channels are used. The channels are made of block RAM and are unidirectional. They consists of two FIFO buffers: one on the FPGA and one on the RT target. When transferring large data sets it is beneficial to use DMA as it saves resources, which is important for data logging. The DMA channels are also used to synchronize the user defined variables between the FPGA and the RTOS, making deterministic communication possible (National Instruments -Scan Clock, 2013). A single DMA channel is also used for for the Human Machine Interface (HMI) control from the RTOS to the FPGA. All other communication between the FPGA and the RTOS is done using User Defined Variables (UDV). For the system to work as stand alone, a combined control and LCD screen module is connected to the expansion port on the Interface Board (IB2). The LED status indicators are used as a ”heartbeat” indicator for the communication VI on the FPGA, and to indicate when the data has been collected.

The final LED is used to indicate if the software over-current safety has been triggered. Both the RTOS and the FPGA have a main VI, where all the functions are controlled from. The functions shown on the RTOS and the FPGA are split into a number of sub-VIs. Before development of these VIs the basics of the sbRIO hardware is introduced.

4.2.1 FPGA

The sbRIO uses a Xilinx Spartan-6 LX45 Field-Programmable Gate Array (FPGA). A FPGA is an integrated circuit designed to be reconfigurable. The FPGA is clocked at a lower frequency than the CPU but because of its structure and configuration with Very high speed integrated circuit Hardware Description Language (VHDL) it is possible to perform multiple, parallel, and sequential operations within a single clock cycle (National Instruments - FPGA Guide, 2014). The FPGA specifications are shown in table 4.2

Xilinx Spartan-6 LX45 FPGA		
Total Slices	6,822	
Number of flip-flops	54,576	
Number of 6-input LUTs	27,288	
Clock management tiles	1080	[MHz]
Number of DSP48s	58	
DSP48s clock frequency	390	[MHz]
Available block RAM	2088	kbits
Block RAM clock frequency	320	[MHz]
DMA channels	5	

Table 4.2: FPGA specifications (National Instruments - FPGA, 2014).

The FPGA consists of configurable logic blocks, or slices, and specialized blocks such as I/O or RAM. These are connected with programmable interconnects, illustrated in figure 4.7.

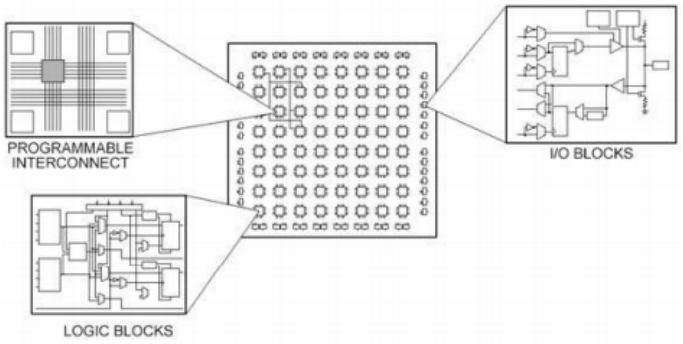


Figure 4.7: Illustration of FPGA Resources (National Instruments - FPGA Guide, 2014).

Each slice contains a number of logic resources, and the programmable interconnects connects the blocks making advanced processing and control tasks possible at hardware speeds. The logic resources include:

- Look Up Tables (LUTs) have predefined list of outputs for every combination of inputs, which is a very efficient way to perform logic operations.
- Multiplexers (Muxs) are used to select between two or more signals, and outputs the selected input.
- Flip-flops are the building blocks of digital electronics systems. Flip-flops have two stable states and represent a single bit, or can be grouped into registers to store bit patterns, such as numeric values. The flip-flops are also implemented as shift registers and feedback nodes.

The specialized blocks perform specific functions, such as clock management, larger data storage (RAM), I/O, or Digital Signal Processing (DSP). The RAM blocks are present throughout the FPGA for storing data, and are used when using memory blocks or FIFOs.

Even though FPGAs have followed Moore's law they are still limited, and consideration must be taken to fit large program designs. When compiling a program on the FPGA the compiler can fail because it runs out of specific resources or, when close to fully mapped, because parts of a slice is unusable due to the way other parts of the slice is configured.

Figure 4.8 shows a small design compiled to the FPGA, and a design that uses most of the FPGA resources, where the red parts are the used resources.

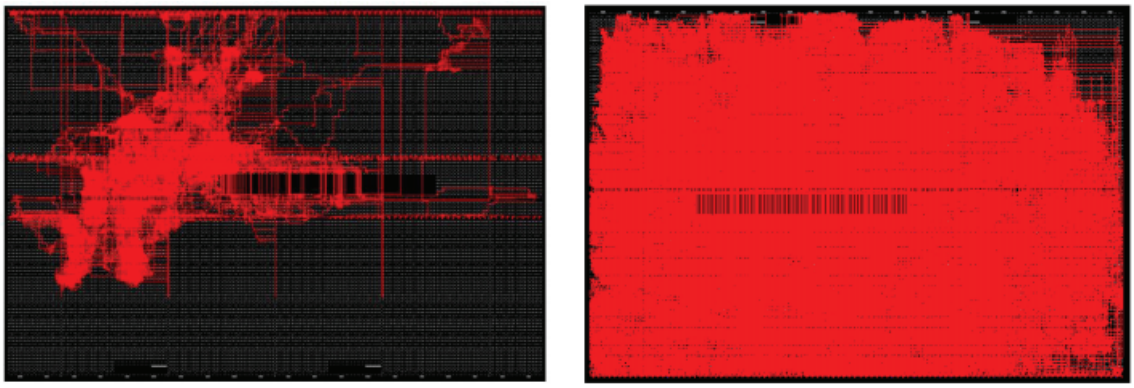


Figure 4.8: Example of small and large design compiled on the FPGA (National Instruments - FPGA Guide, 2014).

When most of the resources are used, the compiler can have problems connecting components, or the propagation and logic delays can prevent the required compilation clock rates.

To achieve higher clock rates the FPGA has specialized DSP slices called DSP48 nodes. These can be used for arithmetic operations or logic operations, and several DSP48 slices can be cascaded to implement complex functions such as advanced digital filters (XILINX - DSP48E, 2014).

When the code is compiled to VHDL the compiler can use different design strategies, depending on the program demands. Effort can be focused on reducing the area of the design or optimizing for speed. Optimizing for area can be used if the VIs are close to filling up the FPGA. The total device utilization is shown in table 4.3.

Device Utilization		
Total Slices	75.0 %	5119 out of 6822
Slice Registers	30.7 %	16756 out of 54576
Slice LUTs	47.8 %	13037 out of 27288
DSP48s	13.8 %	8 out of 58
Block RAMs	6.9 %	8 out of 116

Table 4.3: Device utilization of FPGA with the complete program.

The total program uses 75 % of the available slices, but only 13.8 % of the DSP48s and 6.9 % of the RAM. The digital filters uses the DSP48s, but more of the code could be implemented here to free up other slices, but the DSP48s should be utilized for high speed applications. The code developed for the project is optimized for FPGAs, and uses nearly exclusively the Single Cycle Timed Loop (SCTL). The SCTL is clocked at a frequency derived from the top-level clock. The LCD control to the FPGA is a NI library, and should for future work be rewritten as it often leads to errors in the compilation (National Instruments LCD Driver).

In the program, several different derived clocks are used, shown in table 4.4. The top-level and derived clocks are used by the SCTL structures. The values listed in **"Frequency"** are the requested frequencies. The requested frequency of a SCTL structure is defined directly in the block diagram, while for other structures they are not. The **"Max Frequency"** is the theoretical maximum frequency for each of the clocks. If the required frequency is larger than the maximum frequency the compilation will fail. The difference between the requested and the maximum is a good indication on how 'full' the structures are, and how efficient the code is.

Timing		
Clock name	**Frequency**	**Max Frequency**
BusClk (non-diagram components)	33 [MHz]	70.46 [MHz]
40 MHz - Top-level Clock	40 [MHz]	62.50 [MHz]
80 MHz - 2/1 Derived	80 [MHz]	103.97 [MHz]
60 MHz - 3/2 Derived	60 [MHz]	218.29 [MHz]
20 MHz - 1/2 Derived	20 [MHz]	95.15 [MHz]
2.5 MHz - 1/16 Derived	2.5 [MHz]	47.48 [MHz]

Table 4.4: Timing table of the derived clocks from the top-level clock.

4.2.2 PWM Strategy and Implementation

The requirements for the PWM generator are defined from the analysis made in section 3.3 "*Pseudo Random Binary Signal*", and from the hardware used.

The three PWM signals must be synchronized and the program must be able to maintain a constant duty cycle, while the period is varying. Only the positive PWM signal is required from the FPGA, because the interface board generates the complementary signal and implements dead-time. The switching frequency ranges from four to 11 kHz.

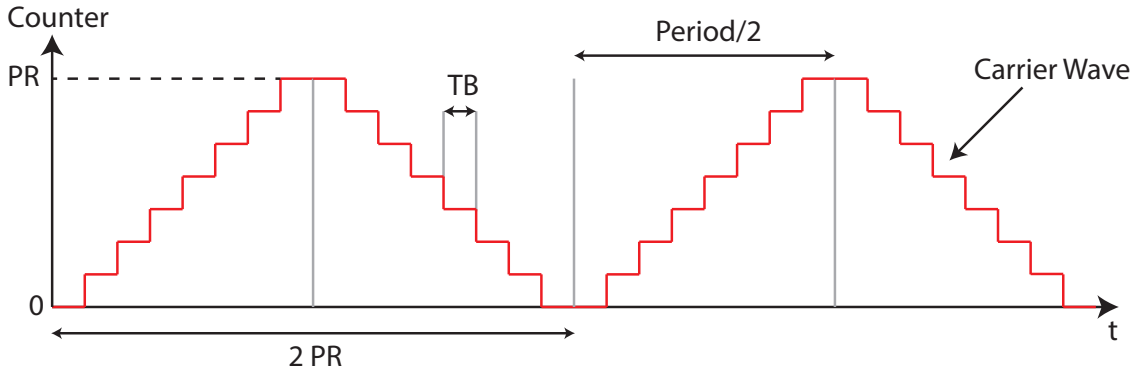


Figure 4.9: Counter based carrier wave.

A counter based carrier wave is used, illustrated in figure 4.9. The counter will increment until it reaches the Period Register (PR), stay high for one extra Time Base (TB) to make sure the PWM signal will be symmetrical, and then decrement until it reaches 0. PR can then be varied to change the switching frequency. The carrier wave is compared to a reference dependent on the desired duty cycle, shown in figure 4.10. The duty cycle is then given by the ratio between the reference and PR. The value of PR and the reference is updated every time PR hits 0, shown in grey dashed lines.

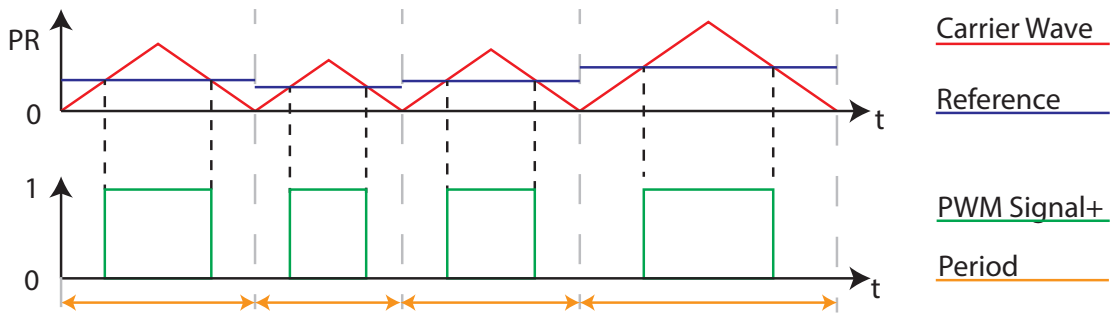


Figure 4.10: Carrier wave, reference, and resulting PWM signal.

For the parameter identification method the switching frequencies (f_s) used ranges from four to 11 kHz. The TB is $1/40MHz = 25ns$. From equation (4.1), this gives a PR value of 4000 at $5kHz$ and 1818 at $11kHz$.

$$\frac{1}{2 \cdot TB \cdot f_s} = PR \tag{4.1}$$

On the FPGA the PWM generator is implemented in a separate VI, shown in figure 4.12. The outmost loop is a Single-Cycle Timed Loop (SCTL) which is a key LabVIEW FPGA structure. Outside of the SCTL each function has registers or small storage elements to clock the data from one function to the next, illustrated in figure 4.11. All functions in the LabVIEW code is a representation of a single or a collection of FPGA circuit components, and all data is transported between these components on the FPGA as electrical signals. Additional control components are used when outside of the SCTL to latch the data on every clock cycle. Each function can take one or more clock cycles to execute. (National Instruments - FPGA Guide, 2014)

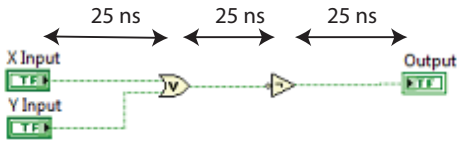


Figure 4.11: Code example of functions running at the 40 [MHz] clock outside of the SCTL.

The SCTL matches the behavior of the FPGA circuits, where a number of registers holds the data, the content of the SCTL is executed in a single clock cycle, and the output data is then written to register elements. When the code is compiled the theoretical maximum frequency the SCTL can execute is calculated to make sure all the code will fit in a single clock cycle. This is an efficient way to implement code on the FPGA because less circuitry is needed to check when the data is ready. The next loop is a flat sequence structure, which executes each frame in sequence, and a frame is finished when all its outputs are updated. In the first frame the counter based PWM generation logic is implemented in case structures. In this frame the register from over-current

safety VI is also updated. In the following frame the values of the logic is written to the digital outputs.

The case structure has multiple cases dependant on the ”state” variable wired to the input terminal of the case structure. In the first frame of the case structure (’1’), all the values are updated from registers located in other sub-VIs. The duty cycle of the PWM signal is a ratio between the PR and the reference (”Low Ticks”), so it is important these values are updated together.

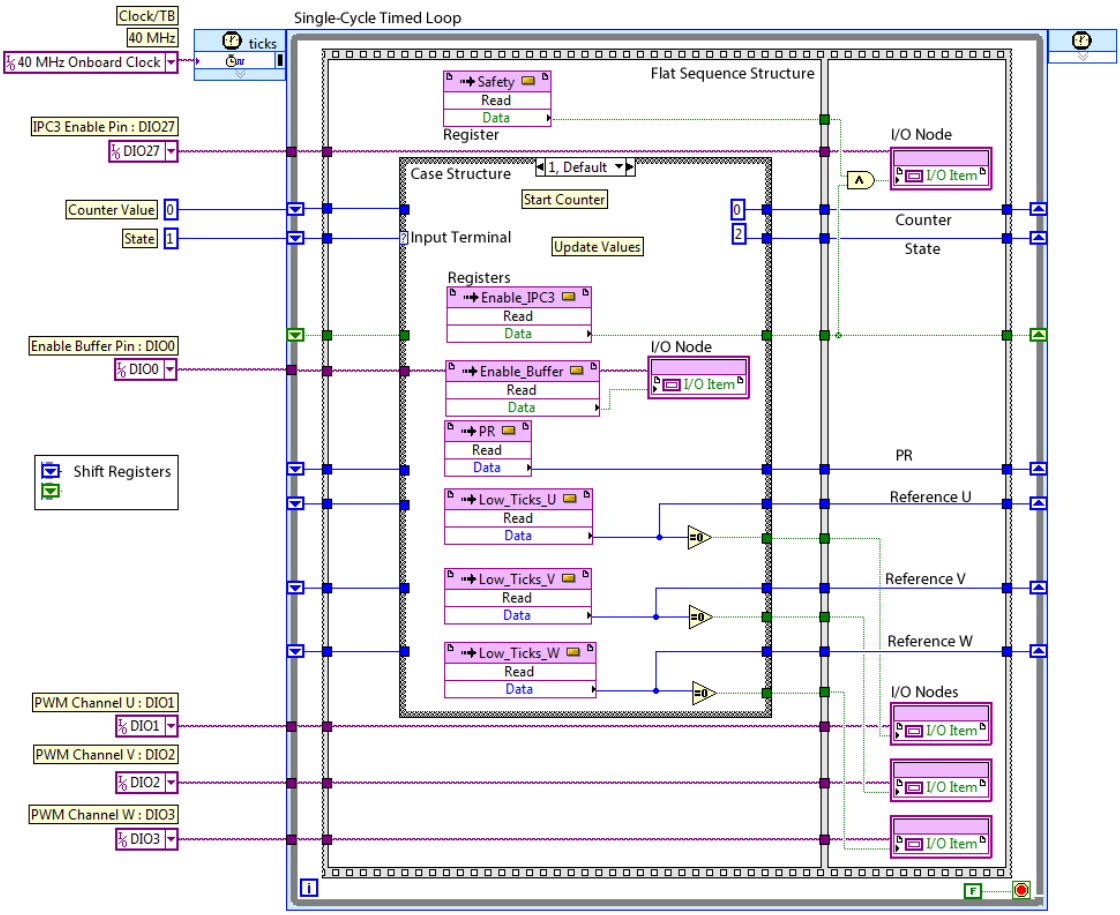


Figure 4.12: Block diagram of PWM generation in LabVIEW.

In total, four cases are used for the counter and the PWM generator logic. The case structure is an easy way to implement counters but are generally not very efficient to use on the FPGA. This is because all the code is written to VHDL so all the possible cases runs in parallel and a mux is used to select the correct outputs. The code could be implemented using select functions which functions as muxs, and thereby matches the way it is implemented, but the benefits of having manageable code is deemed more important.

The first case of the case structure updated all the variables needed, illustrated with a dashed grey line in figure 4.10. Case '2', shown in figure 4.13 counts up to PR, creating the rising side of the carrier wave shown in figure 4.10. Case '3' is the top point of the carrier wave, and the final case '4' is the falling edge, which when the counter value is back at 0, sets the state to '1'. In all the cases the carrier wave is compared with the references to generate the positive PWM signal shown in figure 4.10.

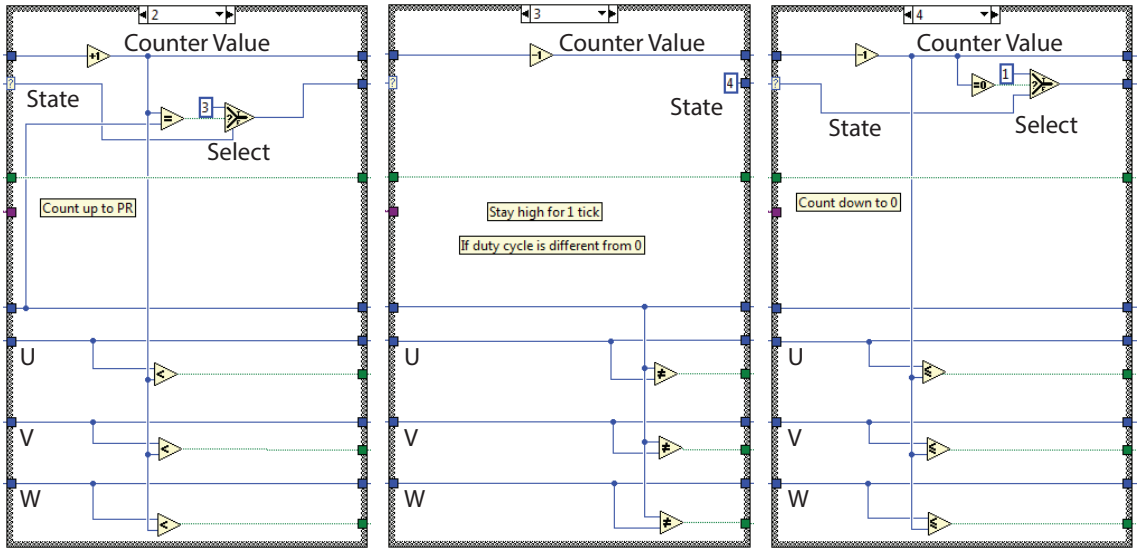


Figure 4.13: Cases from the PWM generator. The variables are updated in case '1', shown in figure 4.12. Case '2' increments up to PR, case '3' is the top point of the carrier wave, and case '4' decrements to 0.

4.2.3 Sequence Generator

The sequence generator is the VI used to generate the PR and references for the PWM generator VI. A functional block diagram of the VI is shown in figure 4.14. PR determines the switching frequency, and the ratio between the reference and the PR determines the duty cycle. The VI is given a switching frequency, which is the center value of the varying switching frequency. The frequency band that the switching frequency can vary is multiplied with a pseudo random number ranging from -1 to 1, and this is then added to the switching frequency. PR is calculated from equation (4.2), where f_s is the switching frequency, f_{sb} is the switching frequency band, and $rand$ is a pseudo random number. In the figure the $2TB$ constant is not shown, but is included in the program.

$$PR = \frac{1}{2 \cdot TB \cdot (f_s + f_{sb} \cdot rand)} \tag{4.2}$$

To calculate the PR the reciprocal needs to be calculated, but dividing inside of the SCTL is very inefficient. Instead it is better to do it in parallel. In the program this is done by transferring the value to a parallel loop, taking the reciprocal, and then transfer it back to the loop. This way the VI can use the SCTL structure, which is important

because the VI needs to be synchronized with the data collection VI. The calculated value for PR is written to a register for communication with the PWM generator VI. The reference for phase V and phase W is a constant 50%. Division by two is possible on the FPGA by a single bit shift to the right.

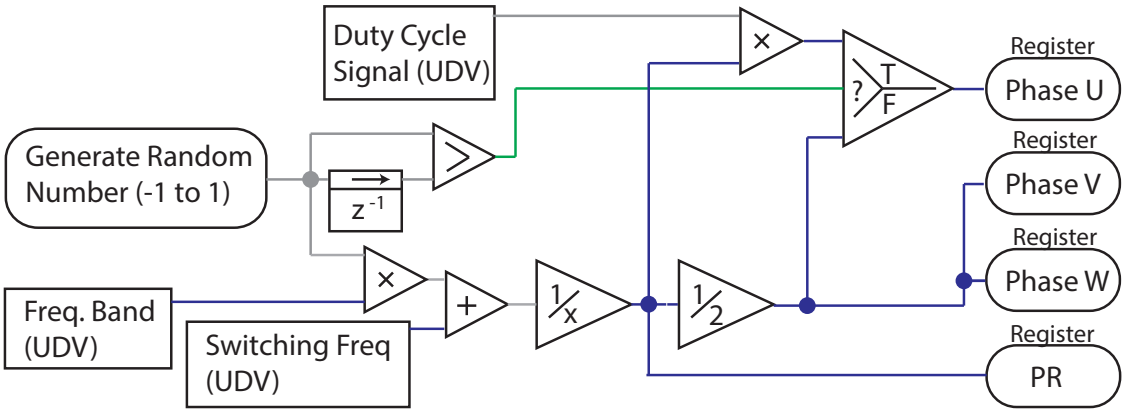


Figure 4.14: Illustration of the sequence generator. The color code of the wires follows LabVIEW syntax. Green is Boolean, grey is fixed point and blue is integers.

The pseudo random number is also used to generate the binary signal. This is done by comparing the number with the previous value, and output a boolean 1 if it is greater and 0 if is less. The boolean value controls the selector and thereby the output of phase U. If the boolean value is 0 the reference for phase U is the same as phase V and phase W. If the boolean value is 1 the reference is dependant on the "Duty Cycle signal", which is a fixed point number ranging from 0-1. The VI is enabled from the main VI and synchronized with the data collection. When the VI is disabled it will output a constant 50% on all three phases.

4.2.4 Pseudo Random Signal Generator

In section 3.3 "*Pseudo Random Binary Signal*" it was determined that the input signal used to excite the system for the identification method, needed to vary in period and alternate randomly between on and off. For the generation of the pseudo random number a software Fibonacci Linear Feedback Shift Register (LFSR) is used. A LFSR is a shift register where the input is linearly dependant on its previous state. A 4-bit Fibonacci LFSR is shown in figure 4.15 b). The LFSR uses XOR gates. The output of the XOR gates are dependant on the values of single bits called taps, numbered from 1 to 4. The inputs of the XOR gate are called A and B, and the logic table for the gate is shown in figure 4.15 a). The LFSR is clocked at a given frequency, and for each clock cycle the Most Significant Bit (MSB) it replaced by the output of the XOR gate and the Least Significant Bit (LSB) is discarded. The 4-bit data can then be interpreted as a number for the output. Three cycles of the 4-bit LFSR are shown infigure 4.15 (Kewal K. Saluja, 1991).

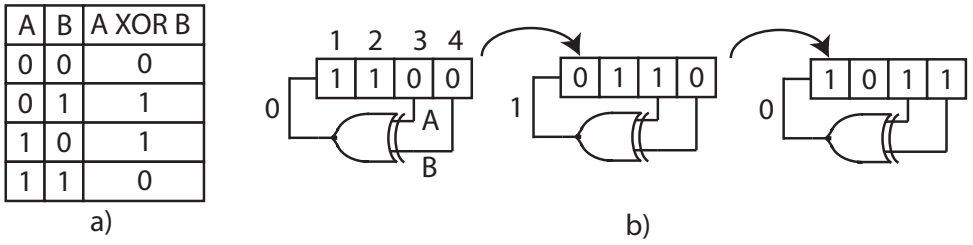


Figure 4.15: Example of 4-bit Fibonacci Linear Feedback Shift Register (LFSR).

The taps chosen as the input for the XOR gate determines the length of the cycle before it repeats. An m-sequence is a sequence with length $2^n - 1$, which includes all the possible states except when all bits are zero. For a 4-bit LFSR there is only one way of achieving this and this is by using taps three and four (New Wave Instruments, 2010). This will result in the cycle shown in figure 4.16.

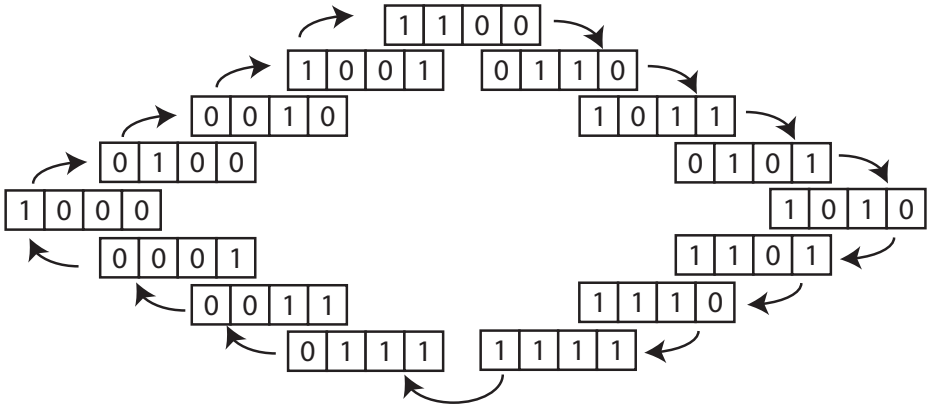


Figure 4.16: Illustration of m-sequence for a 4-bit Fibonacci LFSR.

If this cycle is interpreted as a 4-bit number and the top of the cycle is the seed the cycle would be: 12,6,11,5,10,13,14,15,7,3,1,8,4,2,9 and then repeat. For the pseudo random number generator implemented on the FPGA a 16-bit LFSR is used. This means the cycle will repeat after $2^n - 1 = 65535$ cycles, if the the right taps are chosen. When using four taps, there are 26 configurations, or placement of taps, which will result in a m-sequence. The LFSR is illustrated in figure 4.17 (National Instruments tannerite, 2007).

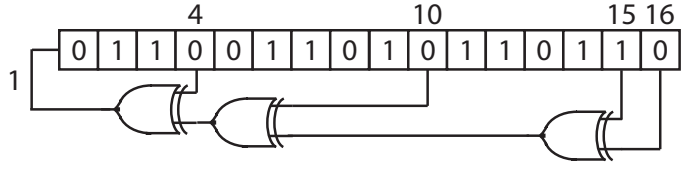


Figure 4.17: 16-bit Fibonacci LFSR with taps 4, 10, 15, and 16.

The output of the LFSR is a 16-bit boolean array, which is interpreted as signed 16-bit fixed point. The integer length of the fixed point is 1, so the resulting pseudo random number ranges from -1 to 0.9999695. The first 500 cycles with the seed shown in figure 4.17.

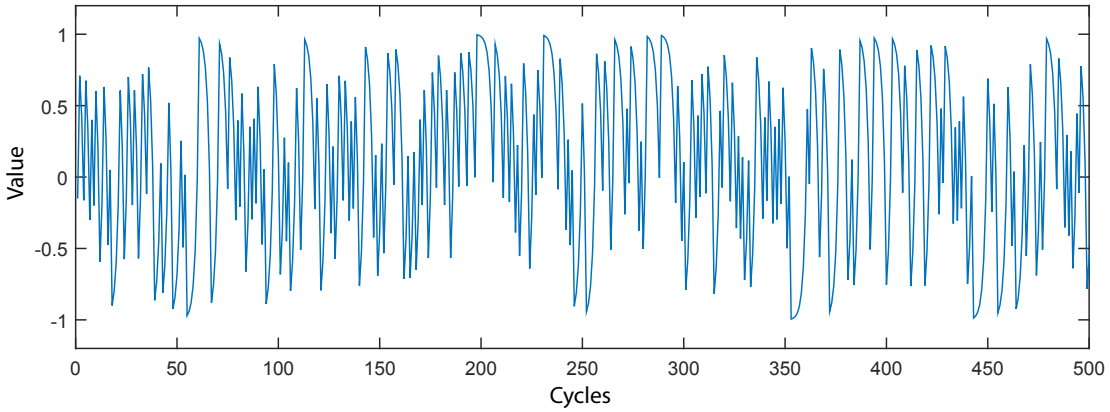


Figure 4.18: First 500 cycles of the sequence with the seed from figure 4.17.

4.3 Inverter Interface and Protection Board

The Interface and Protection Board (IPC3) is a replacement for the original interface card of the VLT. The board provides direct control to the IGBT's gate drivers. The IPC3 is mounted on the front of the VLT, shown in figure 4.19.

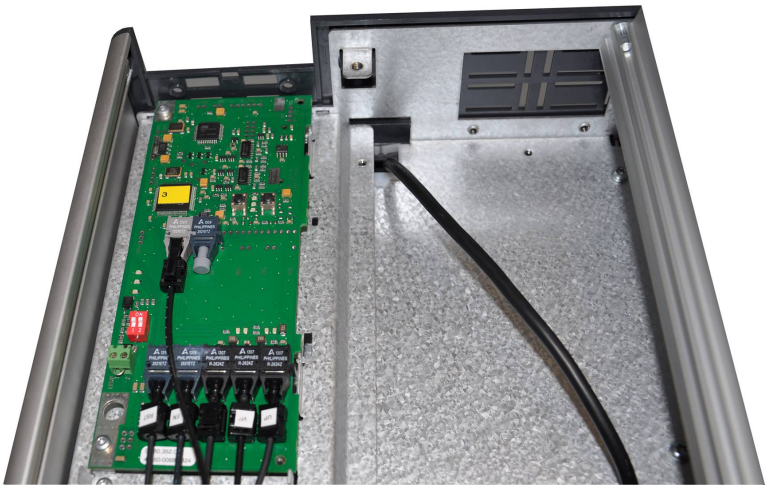


Figure 4.19: The IPC3 board installed in the VLT.

The main function of the IPC3 is to allow access to the IGBT's gate drivers, but it also includes an over-current and temperature protection. The board is controlled through

optical communication. The input for the PWM signal is the signal for the high gate only, and the IPC3 board then takes care of the deadtime and the signal for the low gate. The deadtime is chosen using DIP switches on the front of the board. The board is controlled with a CPLD. Besides the gate signals, the inputs are a RESET signal and an ENABLE signal. The board has a TRIP output to display if the board safety is tripped. If the IPC3 goes into TRIP mode, all the gate signals go low. ENABLE then has to be turned low, RESET needs to be flashed, and ENABLE can then be turned high again to enable the VLT.

4.3.1 Switching and deadtime implementation

When the switching scheme and deadtime is implemented in the model, it uses the same logic as the PWM generator. Therefore the explanation has been placed here and not in the model section. To implement the deadtime in the model an adjusted reference has been used. This offset for the adjusted reference is given by the slope of the carrier wave and the deadtime. The adjusted reference is used every time a transistor is turned on.

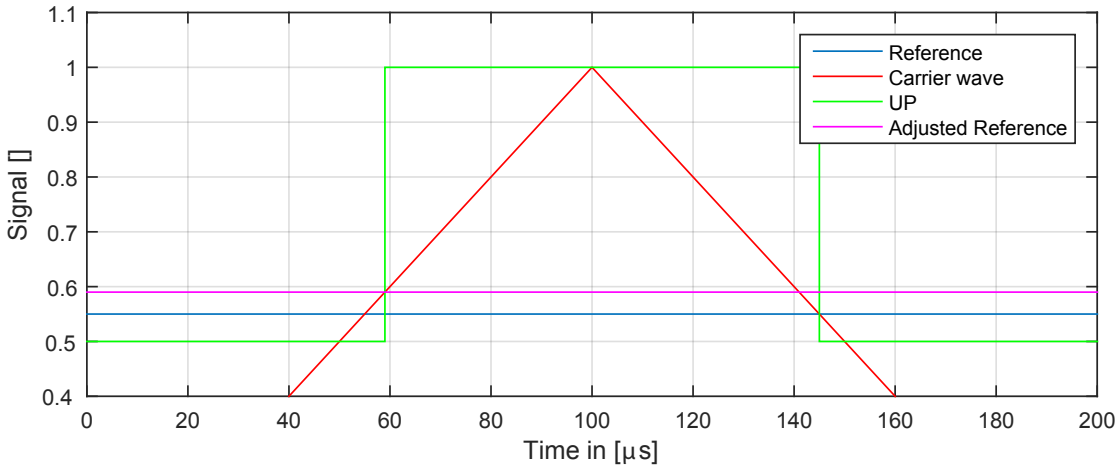


Figure 4.20: Generation of the high gate control signal.

For the control signal to the high gate (UP) the signals are shown in figure 4.20, and for the low gate the signals are shown in figure 4.21. All the signals to turn off the IGBT's follow the reference, and all the signals to turn on the IGBT's follow the adjusted reference.

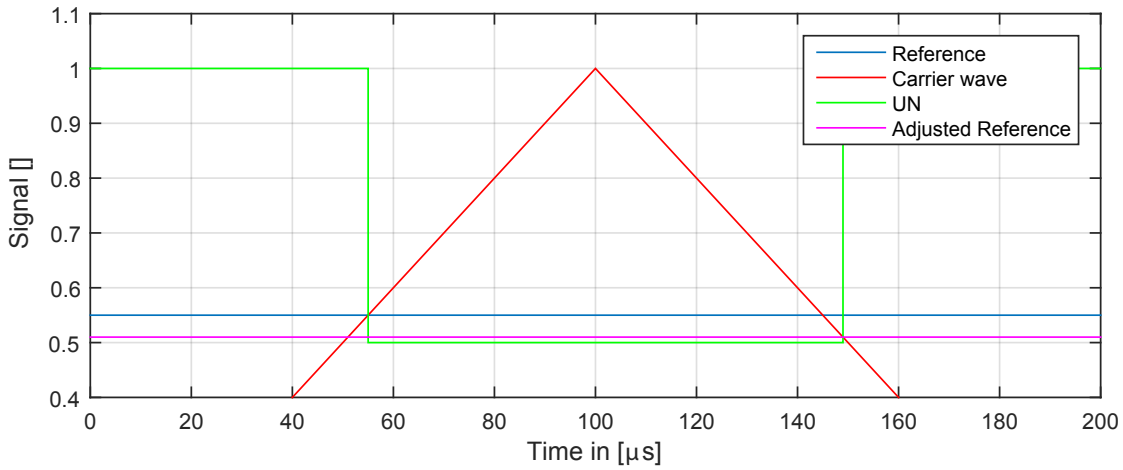


Figure 4.21: Generation of the low gate control signal.

4.4 Custom Parameter Identification Hardware

The following section consists of a description of the hardware designed for the laboratory setup and the thoughts and ideas behind the design. In the previous project "Automatic Parameter Identification for an AC-motor with an LC-filter" (Christensen and Weber, 2014) hardware was designed for the same purpose. The hardware designed for the current project replaces the old hardware, which consisted of two boards, the Interface Board 1 (IB1) and the Current Measurement Board 1 (CM1). Signals were transferred between the boards using Low Voltage Differential Signaling in shielded cat5 Ethernet cables, yet there was troubles with the quality of the high frequency signals because of the cables. The designed hardware has three main functions.

1. Manage communication between the IPC3 and the sbRIO
2. Measure the phase currents at the VLT output
3. Measure the phase-to-phase voltages at the VLT output

All functions are gathered on one Interface Board (IB2), with the sbRIO mounted. By only having one board, the negative effects of applying cables, to transfer signals, are bypassed. The IB2 is mounted at the VLT output, where the three phases enter the board for measurements, and exit to be connected to the LC-filter and motor. Circuit diagrams for the IB2 are found in appendix B "*IB2 Circuit diagrams*". The IB2 at the VLT output is shown in figure 4.22.

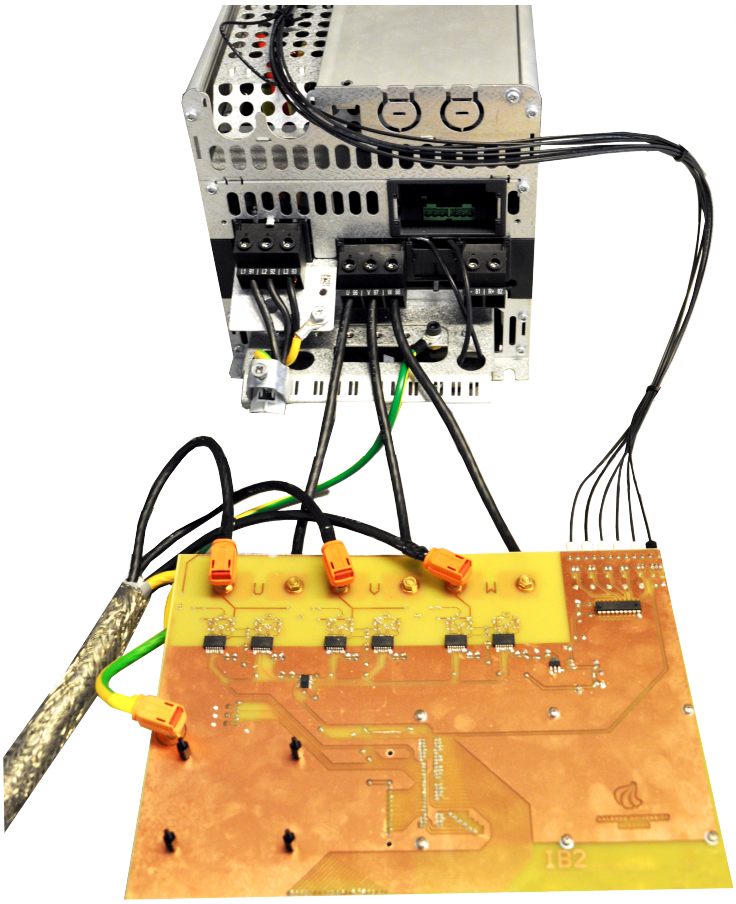


Figure 4.22: IB2 placement at the VLT output.

The bottom side of the IB2, with the sbRIO mounted, is shown in figure 4.23.

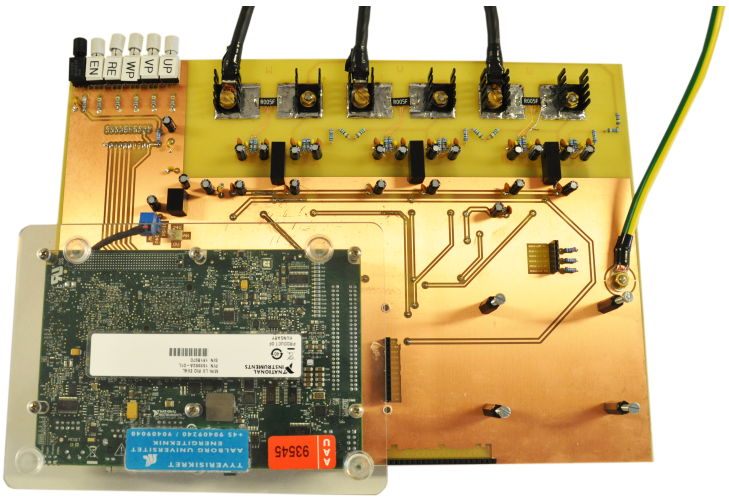


Figure 4.23: Bottom side of the IB2.

The IB2 is designed to be incorporated in a stand alone solution for parameter identification. It is therefore designed to be encased in a grounded metal box, to protect the electronics on the IB2 from noise. The cable shields are to be connected to this box, allowing for a grounded shield to connect the whole laboratory setup.

A block diagram of the functions on the IB2 is shown in figure 4.24.

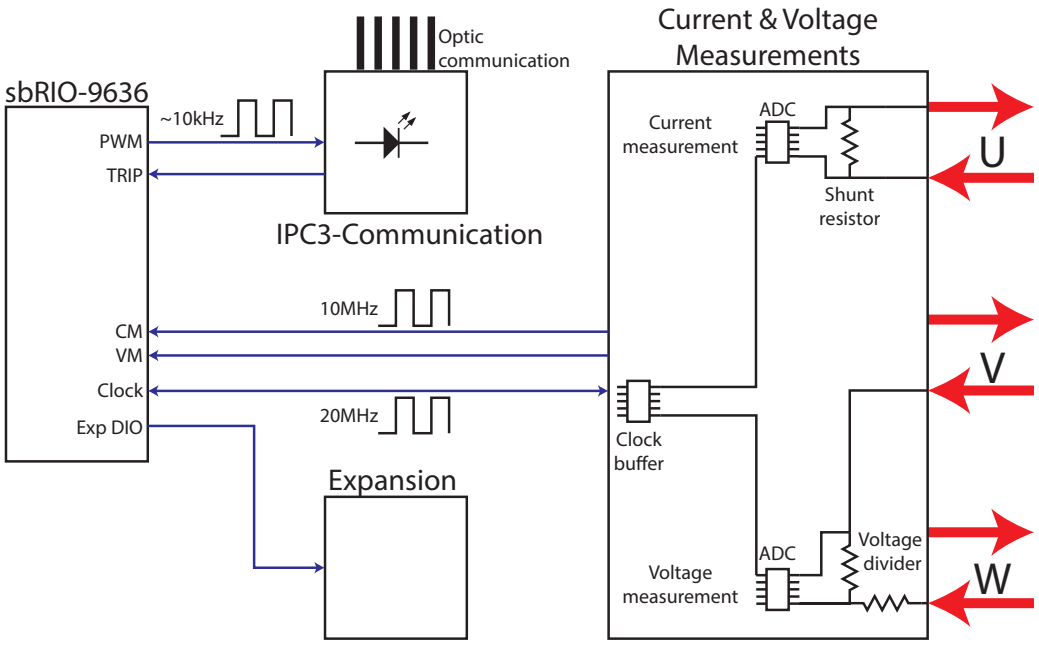


Figure 4.24: Component connection overview of the IB2.

The sbRIO-9636 sends the PWM signal to control the inverter via optical communication, these switching frequency of these signals are of the order 10$[kHz]$. The currents and voltages of the phases out of the inverter are measured and converted to digital signals. The ADCs used require a 20$[MHz]$ input clock and return a 10$[MHz]$ data signal.

Because everything is mounted on one board, there are high differences in voltage potential between circuits, making component and PCB trace placement key when designing the board. The three phases and the measurement of these, are high current and voltages alternating quickly. In order to avoid damage and mitigate noise, it is therefore important to keep these separated from the signals to and from the sbRIO. An illustration of the component placements is shown in figure 4.25.

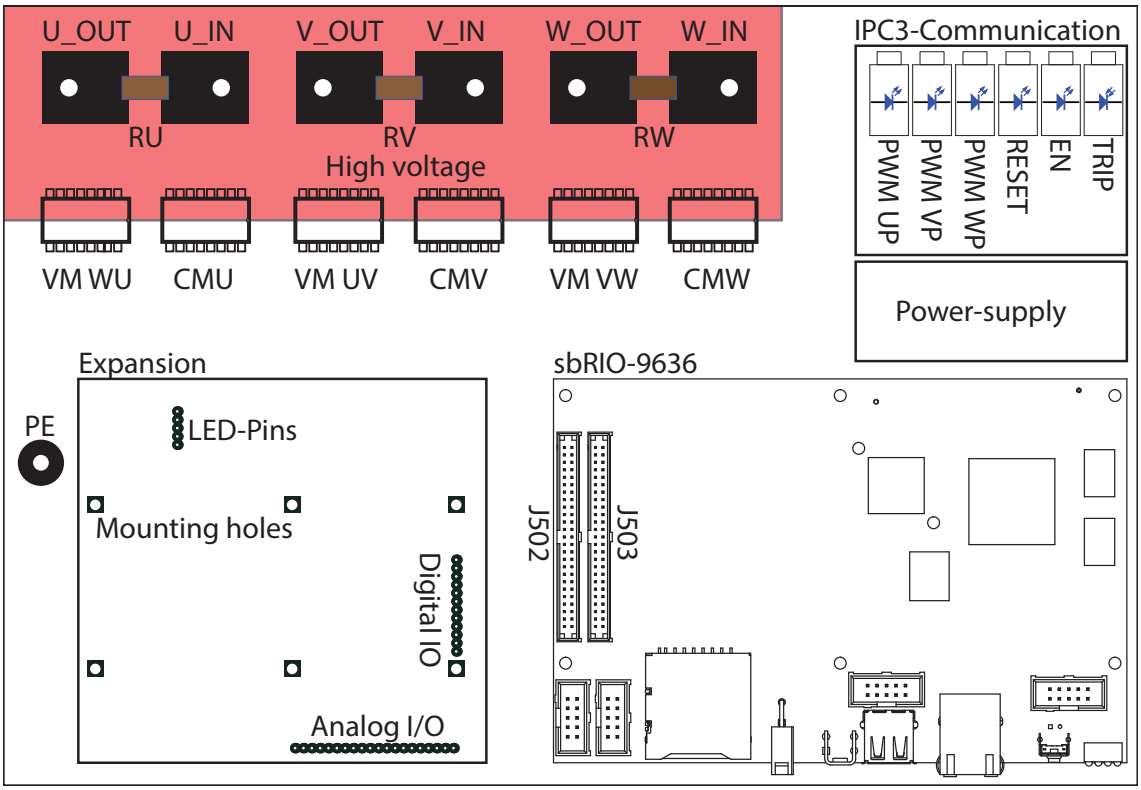


Figure 4.25: IB2 component placement overview. Not to scale.

The red area in the figure represents the area in which high voltages and currents are present. The components on the border to the high voltage area, are analog to digital converters (ADCs) for current and voltage measurements. The rest of the board is electrically isolated from the 3 phases and share $0V$ potential with the sbRIO and Protected Earth (PE). With the exception of a few low-height surface mounted components, all components are placed on the bottom of the board. This allows for connection pins for the phases to be mounted on the topside, giving easy access when connecting the LC-filter. It should be noted that figure 4.25 is not to scale.

In section 4.2 "*Control and Monitoring System*", it was mentioned that a Human Machine Interface (HMI) is created for the experimental setup, such that it can operate as a stand-alone solution. This HMI consists of an LCD display, LED status indicators and control contacts, shown in figure 4.26.

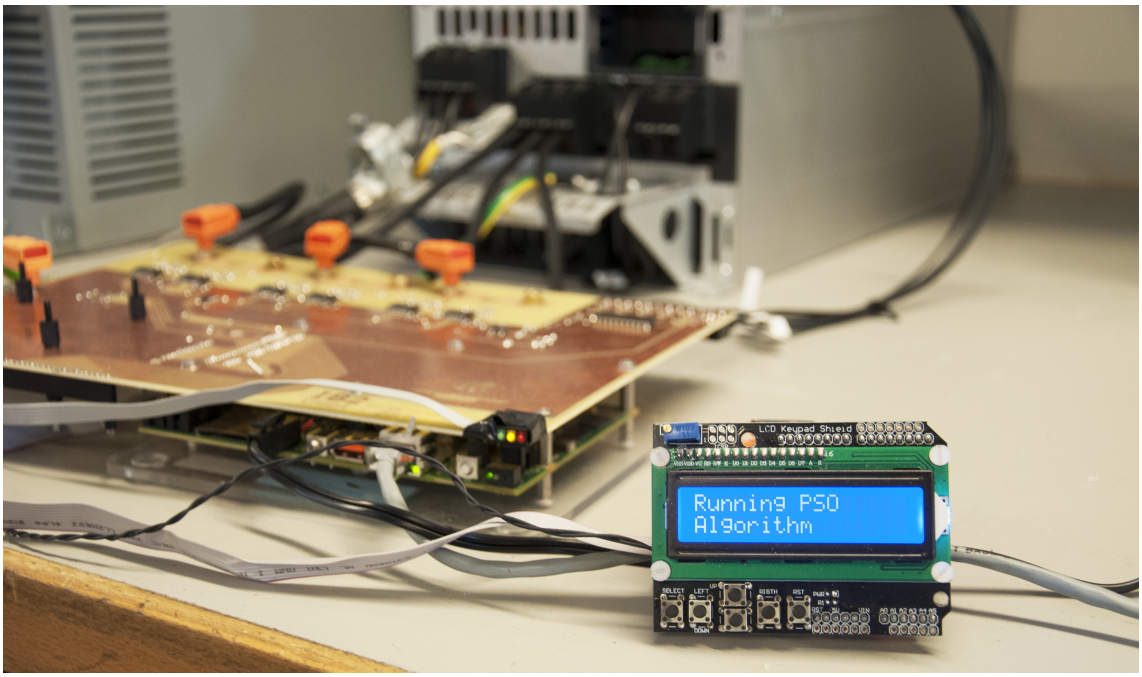


Figure 4.26: HMI to the parameter identification hardware.

The connections, power circuits and general PCB considerations are first explained in the following. Thereafter the current and voltage measurement circuits are explained, followed by how ΔΣ Modulators (ΔΣMs) are used as ADCs.

4.4.1 Connections and Power

The laboratory setup is controlled by the sbRIO, thus all data signals either enter or exit the sbRIO. For this setup, the J502 and J503 connectors on the sbRIO-9636 are used for data transfer. This gives 28 DIO (Digital Input-Output) pins, 18 AI (Analog Input) pins and 4 AO (Analog Output) pins. The sbRIO-9636 can deliver 3$[mA]$ per DIO port with a total of 84$[mA]$ for all ports. Digital pins are used for the following on the IB2:

CM-circuit:

- CM V
- CM U
- CM W

VM-circuit:

- VM WU
- VM UV
- VM VW

IPC3-IO:

- Buffer Enable
- PWM UP
- PWM VP
- PWM WP
- IPC3 Reset
- IPC3 Enable
- IPC3 Trip

Miscellaneous:

- LED1
- LED2
- LED3
- Clock Out
- Clock Returned

Where CM refers to current measurements, VM refers to voltage measurements and PWM UP refers to the positive IGBT gate signal for phase U.

An area of the board, see figure 4.25 is kept clear to leave room for an expansion board. Mounting holes connected to $0V$ and PE are placed, allowing the possibility of mounting a board. Pins, connected to the sbRIO pins, are placed in this area, giving the possibility of expanding the IB2 with 9 DIO, 12 AI, 4 AO and a $24V$ power supply.

Power-supply

The IB2 is powered by the $24[V]$ power supply of the sbRIO-9636. The sbRIO can not supply enough current to drive the optical drivers in the IPC3-IO circuit, so a buffer is applied on the signals going to the optical drivers. The buffer requires $5[V]$ input and a $8[V] - 42[V]$ to $5[V]$ DC/DC converter is placed on the IB2. As shown in table 4.1, the digital ports of the sbRIO have a logic low below $0.8[V]$ and a logic high above $2[V]$. This leaves a midpoint of $1.4[V]$ and in order to obtain a somewhat symmetrical voltage swing across this midpoint, it is chosen to apply $3.3[V]$ logic on all other digital signals on the IB2. Another advantage of using $3.3[V]$, as opposed to $5[V]$, is that many of the signals on the IB2 switch with $10 - 20[MHz]$, where the switching noise is reduced by the lower voltage signals (Devices and Zumbahlen, 2008). A $5V - 12V$ to $3.3V$ voltage regulator, supplied by the $5[V]$ output of the DC/DC converter, supplies the $3.3[V]$ for the rest of the circuits.

On the phase-voltage sides of the $\Delta\Sigma$ Modulators, a $5[V]$ power supply is required. These power supplies must float with their respective phases. A Current Measurement (CM) circuit and a Voltage Measurement (VM) circuit share $0[V]$ potential and pair as follows:

Phase U:	**Phase V:**	**Phase W:**
• CM U	• CM V	• CM W
• VM WU	• VM UV	• VM VW

To supply these, a 24$[V]$ to 5$[V]$ isolating DC-DC converter is placed at each phase.

IPC3 Communication

As described in 4.3, the IPC3 replaces hardware in the VLT and controls the IGBTs. The IPC3 only requires the high-side gate signals (PWM- UP, VP and WP) and generates the low-side gate signals (PWM- UN, VN and WN) by setting the desired deadtime. The communication between the IPC3 and the IB2 is by fiber optic drivers and receivers. The signals are transmitted by drawing a current through a Light Emitting Diode (LED), the light is then channeled through a fiber optic cable, collected by a photodiode and then turned back into a voltage signal. In order to draw the signals as currents, resistors are chosen to match the chosen optic drivers and receivers. A function diagram of the setup for the optic drivers and receivers is shown in figure 4.27.

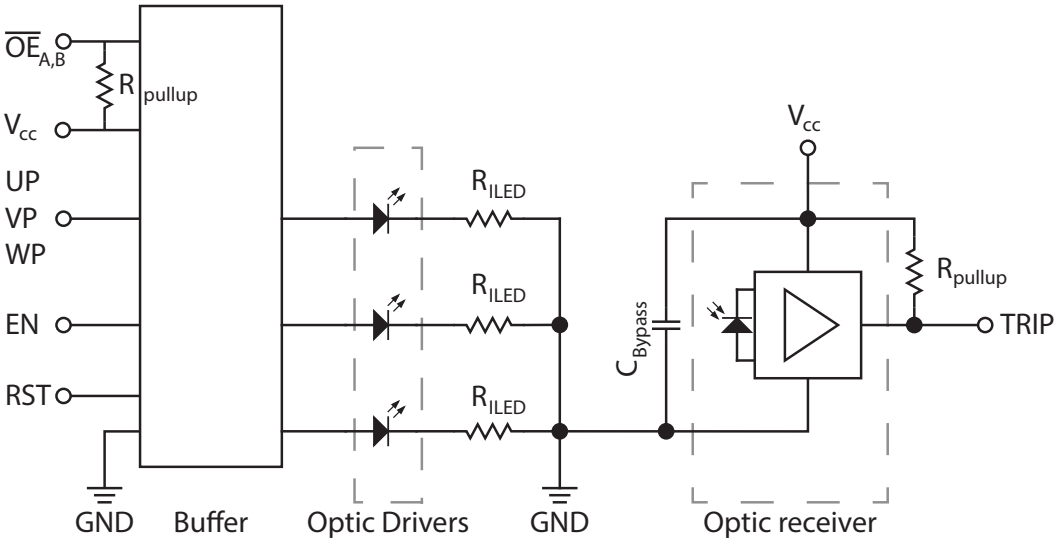


Figure 4.27: Function diagram of the optical driver and receiver circuit on the IB2.

The buffer receives the high-gate PWM signals, along with ENABLE and RESET as 3.3$[V]$ signals from the sbRIO. The buffer is supplied with 5$[V]$, making the output signals 4$[V]$ and able to draw a current of up to 25$[mA]$ (Texas Instruments HCT540, 2004). $\overline{OE}_{A,B}$ are inverted enable ports on the buffer, allowing, a single enable signal from the sbRIO, to set all signals out of the buffer low. When configuring the sbRIO all DIO ports are floating. In order to ensure that the IPC3 does not react to any noise or unknown output from the sbRIO, when it is not active, a pullup resistor is placed on the $\overline{OE}_{A,B}$.

A PWM signal before and after the buffer is shown in figure 4.28, where it is shown that the voltages are as expected and the quality of the signals are acceptable.

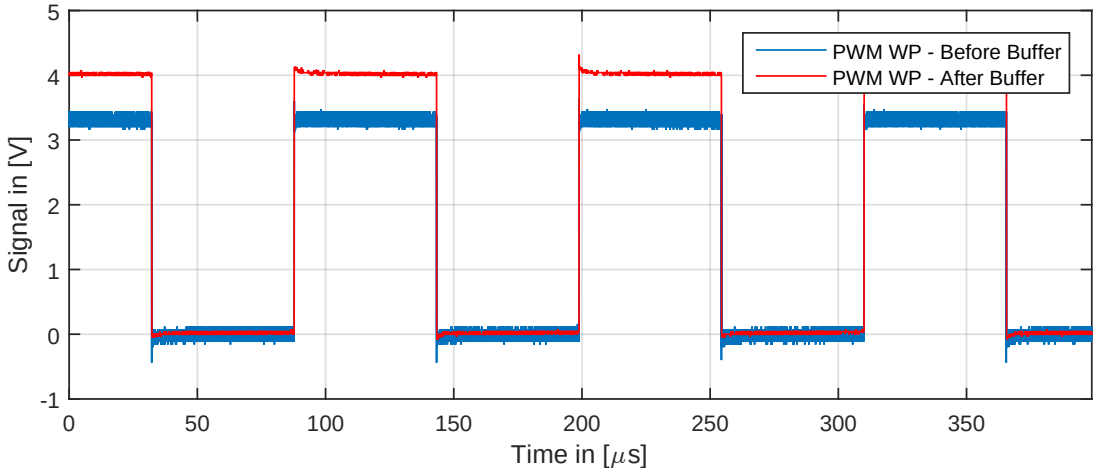


Figure 4.28: PWM WP out of the sbRIO and at the output of the buffer, with a 50% duty cycle at 9$[kHz]$.

The optical receivers on the IPC3 react to the optical power received, measured in decibel-miliwatts (dBm). As a rule of thumb, the higher the optical power, the better. However, the power must be within the receiver's dynamic range, stated as the max power for Logic '0' in table 4.5. (Alwayn, 2004) The logic states for the receivers on the IPC3 are:

	Min	Max	Unit
Logic '0'	-21.6	-8.7	$[dBm]$
Logic '1'	-	-43	$[dBm]$

Table 4.5: Logic states for the optical receivers on the IPC3 (Avago Technologies HFBR-0500ETZ Series, 2014).

The optical output power of the transmitters is dependent on the current drawn through the LED. In order to determine the size of the resistor, to draw the current through the optical transmitter, the voltage drop across the resistor must be known. This voltage drop is dependent on the voltage of the signal and the voltage drop across the diode, known as the forward voltage. The size of the resistor can be calculated by equation (4.3).

$$R_{ILED} = \frac{V_{Signal} - V_{Forward}}{I_{Forward}} \tag{4.3}$$

The forward voltage of the diode is dependent on the forward current, and can be determined from the graph in figure 4.29. The forward current is found by choosing the desired optical output power and reading the graph in figure 4.29.

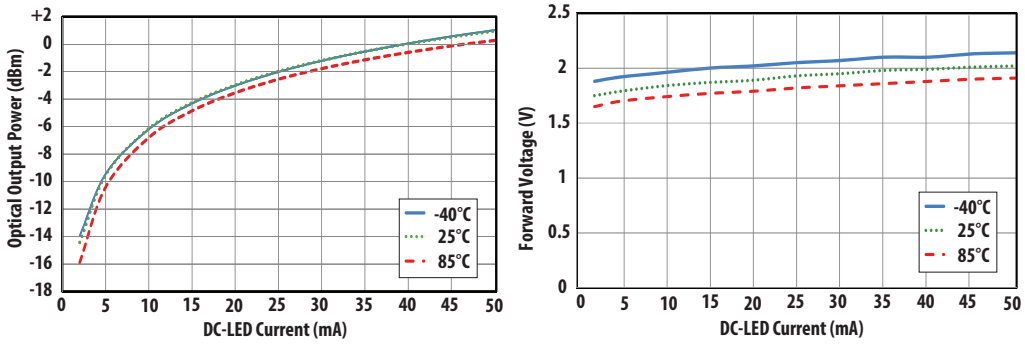


Figure 4.29: Graphs for design of optical transmitter resistors, from Avago Technologies Application Note 5341. (Avago Technologies AN5341, 2012)

The minimum optical power required for a logic '0' is $-21.6[dBm]$, but must not exceed $-8.7[dBm]$ and with loss to expected through the cables, a desired optical power is chosen to be $-10[dBm]$. Using the graphs in figure 4.29, this requires a forward current of approximately $5[mA]$, resulting in a V_f of about $1.8[V]$. Using equation (4.3), a resistor of approximately $440[\Omega]$ is required.

General PCB design considerations

During the design of IB2, several considerations are taken to ensure satisfactory signaling throughout the board. The following sections describes some of the largest factors affecting the design of the IB2. Three of the troublemakers are first clarified, based on (Devices and Zumbahlen, 2008), followed by a description of the issues they cause and finally the solutions to these. The three troublemakers are:

1) Stray Inductance - All conductors are inductive, meaning that all wires and PCB tracks have an inductance. The inductance is a function of the physical characteristics of the conductor, meaning the width, height and length of a PCB track. The inductance of a conductor can become important at high frequency signals, because the inductance becomes and impedance, which will be inspected later in the chapter.

2) Mutual Inductance - Mutual inductance is when unwanted inductance from one source affects other circuits. This unwanted inductance can appear on conductors when a "coil" is formed. This includes PCB traces, where the signal out and the current return path create a coil, shown in figure 4.30.a, or long leads of components creating a coil effect, shown in figure 4.30.b. When current is drawn through an inductor, a voltage can be induced in another, disturbing a signal that might be present.

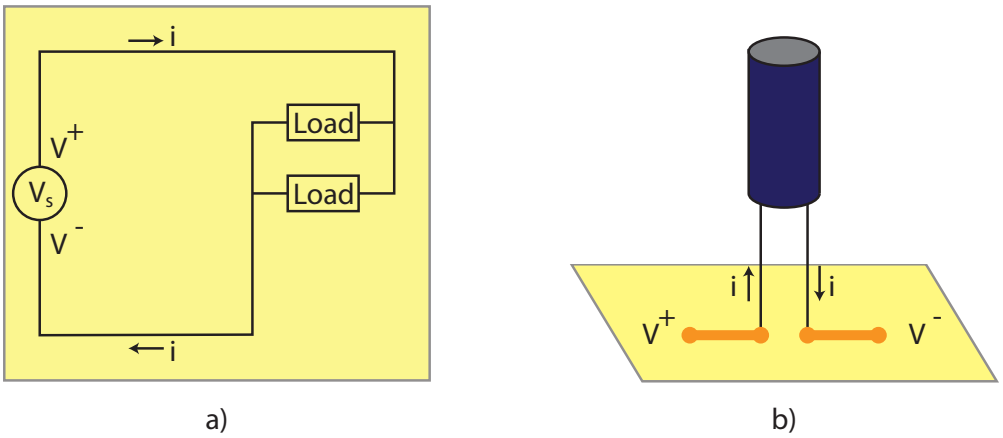


Figure 4.30: Unwanted inductance caused by a coil effect.

3) Stray capacitance - When two conductors are in the vicinity of each other and not short-circuited or screened from each other they act as a capacitor. This includes circuit traces with respect to one another as well as circuit traces with respect to ground, creating a large number of capacitors on a PCB. This can have a high effect on the impedance of PCB traces on high frequency signals. As with the stray inductance, the stray capacitance is a function of the physical characteristics of the conductors, the gap between them and the dielectric constant of the medium separating them. The dielectric constant is the relative permittivity of a material to vacuum. The permittivity is a material's ability to resist an electric field, and is measured in $\frac{farads}{meter}$.

With these three sources of signal degradation explained, the design aspects for the IB2, that are influenced by these, are described. There are two major considerations to take into account:

a) Large differences in voltage potential.

b) High speed digital signals (up to 20$[MHz]$).

a) High voltages:
The three phases out of the inverter enter the IB2 to be measured, and exit again, as shown in figure 4.22 and figure 4.24. The phase voltages, as shown in Chapter 2 "*System Models*", are PWM based voltage pulses of up to 600$[V]$ and the VLT is rated to 61$[A]$. The high currents in the cables between the inverter and the IB2 cause a considerable amount of noise, because of the stray inductance in the cables. The IB2 is designed to be enclosed in a grounded metal box, which functions as a Faraday shield, protecting the electronics on the IB2 from external noise.
Besides producing a lot of EMI, these high switching voltages must be kept separated

on the IB2. When considering spacing between conductors on a PCB, two distances are to be considered. Clearance is the air distance from one conductor to another, while creepage is the minimum distance along the board between two conductors, illustrated in figure 4.31.

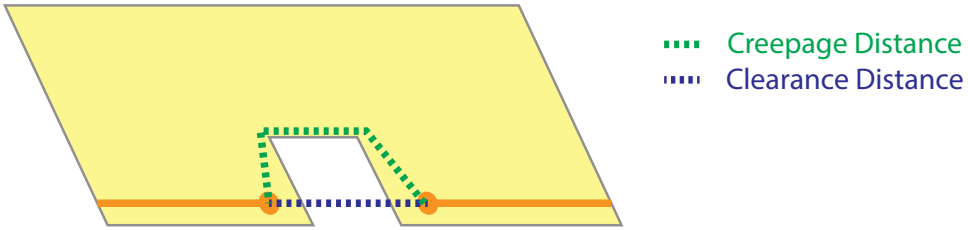


Figure 4.31: Clearance distance and creepage distance.

The purpose of the clearance distance is to prevent a high voltage from creating a spark between two conductors, through the air, and damaging the hardware. The breakdown of a clearance path is a fast process that can occur at a voltage peak. Creepage, on the other hand, is a slow process where environmental conditions and a steady or RMS voltage generate a conductive path over time. The effects of having too low a creepage distance might not necessarily be seen until many hours of operation have been completed. Effects such as temperature, humidity and material characteristics affect creepage. (Brucchi et al., 2012) On the IB2, no isolation barriers or gaps in the board have been applied to increase the creepage distance, making the clearance and creepage distance, from a practical perspective, equal. When deciding the distance between high voltage conductors on a PCB, there a many different standards and guidelines to follow. This includes US, European, German and international standards. These are each divided into categories dependent on the purpose of the hardware to be designed. The international standard IPC-9592 have released a linear equation to determine a conservative creepage distance, to be used as a rule of thumb, shown in equation (4.4).

$$creepage[mm] = 0.6 + V_{peak} \cdot 0.005 \tag{4.4}$$

With a voltage potential difference peak of 600$[V]$ between a phase and PE, this gives a creepage distance of 3.6$[mm]$. Between two phases there can be up to 1200$[V]$ difference, giving a creepage distance of 6.6$[mm]$. The IB2 has been designed such that the distances between the phases are maximized, in regard to what is practically possible, to a minimum of 12$[mm]$. However, the 5$[V]$ DC-DC converters to power the $\Delta\Sigma$ Modulators on the floating side, only have a 3$[mm]$ pin distance between the ground for the high-voltage side and the ground for the low-voltage side (PE). This distance falls 0.6$[mm]$ short of the conservative distance from the equation. Through stress tests with high voltages on the IB2, and with knowledge of the equation being conservative (Brucchi et al., 2012), the 3$[mm]$ creepage distance is assessed acceptable.

b) High speed logic
When applying digital logic signals, there are several effects to consider while designing a board.

The first of which is switching noise, which can occur two different ways. A logic component consists of a pair of transistors switching between V^+ and $Ground$. When the output of the component changes logic state, there is a short period where both transistors are ON and current flows from V^+ to $Ground$. These currents can cause voltage spikes affecting both V^+ and $Ground$ (Horowitz and Hill, 2015). If $Ground$ experiences a voltage spike, this can become a problem for other logic components on the board, as a shift in ground might be interpreted as a logic $HIGH$.

Another negative effect from switching is more linked to high-speed switching. Digital signals, especially high-speed logic, require a fast signal rise and fall time. A clock-buffer, replicating a 20$[Mhz]$ clock signal, used on the IB2 has an output signal rise speed of approximately 1$[V/ns]$, and with 3.3$[V]$ logic the rise time is 3.3$[ns]$. This can be a problem if stray capacitance is present on the routes. At fast switching, a digital output sees the stray capacitance of the PCB traces as an impedance and thus a load, that must either sink or source current (Horowitz and Hill, 2015). This current, in return, can cause voltage spikes, disturbing the signals.

To reduce these effects, all free space on the IB2 has been converted to ground areas, meaning that return currents have as short a travel distance as possible, reducing the inductance and resistance. The ground areas are henceforth referred to as ground planes, although they differ slightly from actual ground planes. Instead of being true ground planes, the ground areas are interrupted by the signals that are traced on the board. A single ground plane is not practical on the IB2, as several signals need to cross one another. Through-holes are used liberally to connect the ground on the two sides, ensuring return currents have as short a path as possible. In addition the ground planes are connected to PE on the inverter, in order for the ground planes also to function as a shield, absorbing noise.

Bypass capacitors have also been placed on all component power supplies, including voltage regulators, serving to reduce voltage spikes, as they function as local voltage sources, able to react to fast switching voltages.

Transmission lines
At high frequencies, or just fast rise times, the capacitance and inductance of cables or PCB traces become a significant impedance. The wavelength of the signal can become significant, in regard to the length of the cable or PCB trace. When dealing with high-frequency circuits, it is often necessary to regard cables and PCB traces as transmission lines. A transmission line is a cable or PCB trace specially designed to carry a switching signal of high frequency, by having uniform dimensions along their length and thus a constant impedance, known as the characteristic impedance Z_0. The characteristic impedance is determined by the geometry and materials of the transmission line. The

problems associated with transmission lines are delay, reflections and crosstalk (Weiler et al., 2006).

Discontinuities in the impedance, between the source of the signal, the transmission line and the destination of the signal, can cause reflections of the signal. Therefore when designing for transmission lines the following three must be considered:

1. The impedance at the signal source
2. The impedance of the transmission line
3. The impedance of the destination

Depending on the sizes of these, for example if the source impedance is lower than the transmission line, different undesired effects can occur. If a signal is propagated in a transmission line directly from a driver and picked up by a receiver, the signal will reflect back and forth, adding to the source signal (Horowitz and Hill, 2015). This can cause the signal at the receiver to contain logic voltage many times the size of the original signal and have logic highs in places they don't belong. This can be solved by the use of impedance matching.

Impedance matching is the act of changing the impedance at the source or destination to match that of the transmission line, in order to mitigate the negative effects of the reflections. Impedance matching, also called termination, can be implemented in several different ways, which include adding a series resistor at the signal source, adding a pull down resistor at the destination or double-ended termination, which is a combination of both. Using a coaxial cable as an example the impedance matching resistors are illustrated in figure 4.32.

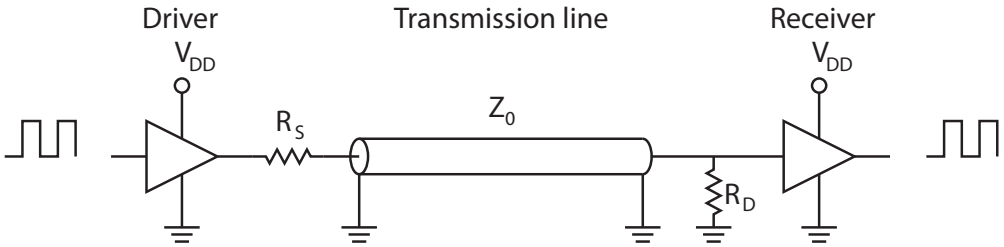


Figure 4.32: Impedance matching options for a transmission line.

Far-end termination is when a pull-down resistor is placed at the receiver. This suppresses all reflections and makes the cable input look like a purely resistive load of $Z_0 = R_D$ (Horowitz and Hill, 2015). However, it is apparent from the figure that the far-end termination results in a current drawn. If the logic voltage is $3.3[V]$ and the characteristic impedance of the transmission line is $50[\Omega]$ (typical value for a coaxial cable), then $66[mA]$ is drawn through R_D.

If a matched series resistor is added at the driver, then the load as seen by the driver is doubled, and the current required is reduced. The added series resistor will also swallow the suppressed reflections, returning from the receiver end. However, the cable's input resistor forms a voltage divider with the series resistor, halving the voltage of the signal at the receiver end. The final way of impedance matching is by only having the series resistor at the driver output. The series resistor and the characteristic impedance of the transmission line still form a voltage divider. However, the reflection at the receiver end is not suppressed and the signal is fully reflected. The reflected signal has the same sign as the incident and the sum becomes one full signal. The reflection is swallowed by the series resistor and does not reflect more than once. In this manner, reflections are suppressed and a full range signal reaches the receiver (Horowitz and Hill, 2015) .This is illustrated in figure 4.33.

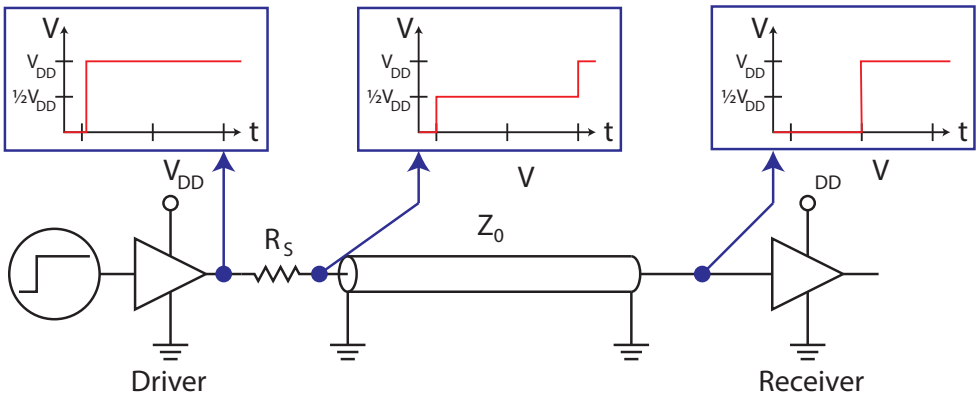


Figure 4.33: Series impedance matching. Figure inspired by (Horowitz and Hill, 2015).

With this method of impedance matching the load seen by the driver is $2Z_0$ and therefore draws the least current. The digital output ports on the sbRIO-9636 are routed with a 55[Ω] characteristic impedance and able to sink or source up to 3$[mA]$. This rules out far-end termination or double-ended termination unless a driver is implemented on the digital signals from the sbRIO-9636, to deliver more current.

First, it is inspected whether it is necessary to consider the PCB traces on the IB2 as transmission lines.

National instruments defines a rule of thumb for when to consider transmission lines as: When the physical length of a wire or trace exceeds 1/6 of the electrical length of the signal propagating on that wire or trace (National Instruments Termination, 2014). The maximum PCB trace length, before the signal should be treated as a transmission line, is thus calculated as:

$$L_{max} = \frac{L_{electric}}{6} \tag{4.5}$$

Where the electric length of a signal is calculated by:

$$L_{electric} = t_{rise} \cdot V_p \tag{4.6}$$

V_p is the velocity of the signal propagating in the transmission line and t_{rise} is the rise/-fall time of the signal. The rise/fall time of the signal is dependent on the component driving the signal, and usually given in the datasheet of the component. The fastest of the two (rise or fall) is the most critical, and should chosen as t_{rise}. The velocity of the signal is calculated as:

$$V_p = \frac{c}{\sqrt{\epsilon_{eff}}} \tag{4.7}$$

Where c is the speed of light and ϵ_{eff} is the effective dielectric constant of the surrounding materials. Because ϵ_{eff} is dependent on many physical factors such as the width and height of the PCB trace, the electric properties of the surrounding materials and their distance to the trace, it can become complicated to calculate.

Avago's free software AppCAD, is used to estimate the ϵ_{eff} of the traces on the IB2. There are several inconsistencies in the physical parameters of the PCB traces that cannot be taken into account. As an example, PCB traces close to component pins run parallel with no room for ground planes areas in between, and at other points on the IB2 are surrounded by ground areas. This is shown in figure 4.34.

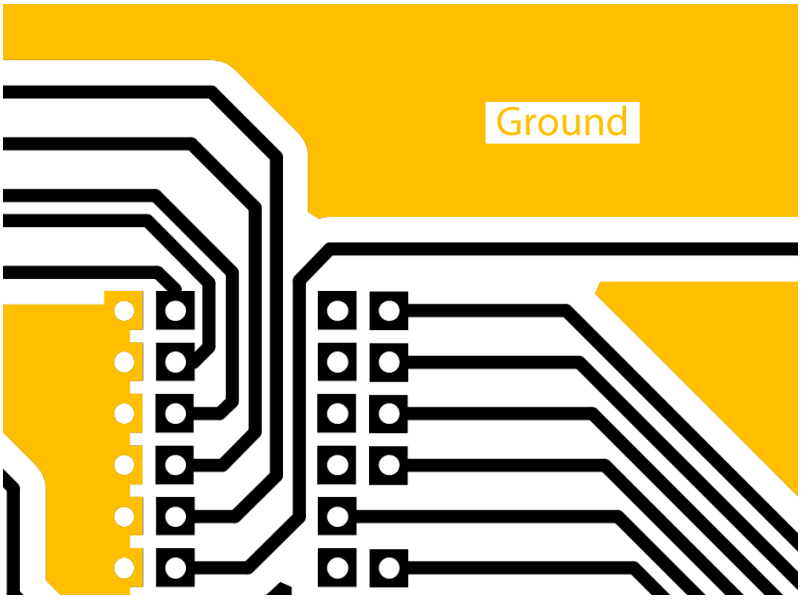


Figure 4.34: Example, from the top layer of the IB2, of trace geometry inconsistencies.

Therefore the ϵ_{eff} is calculated with different parameters to see their influence on ϵ_{eff}. The PCB traces on the IB2 can be described as a microstrip, which is a PCB trace over a ground plane, separated by a dieelectric material. But because the PCB traces also

have ground surrounding the trace at times, it can be seen as a coplanar waveguide, which is a microstrip surrounded by ground. A microstrip and a coplanar waveguide are illustrated in figure 4.35 and figure 4.41 respectively.

The IB2 is handmade from a dual layer copper laminated board, where the dieelectric material FR-4 seperates the two copper layers. The specifications for the board are shown in table 4.6.

Copper thickness	0.035	$[mm]$
Dielectric thickness	1.5	$[mm]$
Dielectric constant	4.6	$[-]$

Table 4.6: Specification for the copper laminated board used for the IB2.

On the IB2 there are several different trace widths and different signal speeds. The combinations of signal frequency and trace widths of interest on the IB2 are:

- $20[MHz]$ Signal with a trace width of $0.5[mm]$
- $20[MHz]$ Signal with a trace width of $0.25[mm]$
- $10[MHz]$ Signal with a trace width of $0.5[mm]$

The lengths of these traces differ, with the shortest distance being $31.5[mm]$ and the longest $160.4[mm]$. Most of the distance are, however, in the range of $100[mm]$. Neither ϵ_{eff} nor Z_0 are dependent on the length of the PCB track or the frequency of the signals, if the geometry is assumed constant. It should be noted that the geometry is in fact not constant, as the tracks turn and change board side using vias. However, for the purpose of this inspection, it is assessed that the geometry can be assumed constant. In order to enforce this, the angles of the trace turns have been reduced, by never making 90° turns.

An example of ϵ_{eff} estimated on a microstrip, using AppCAD, is shown in figure 4.35 with a $20[MHz]$ signal with a trace width of $0.5[mm]$.

With the same geometry the ϵ_{eff} is estimated for a coplanar waveguide in figure 4.41.

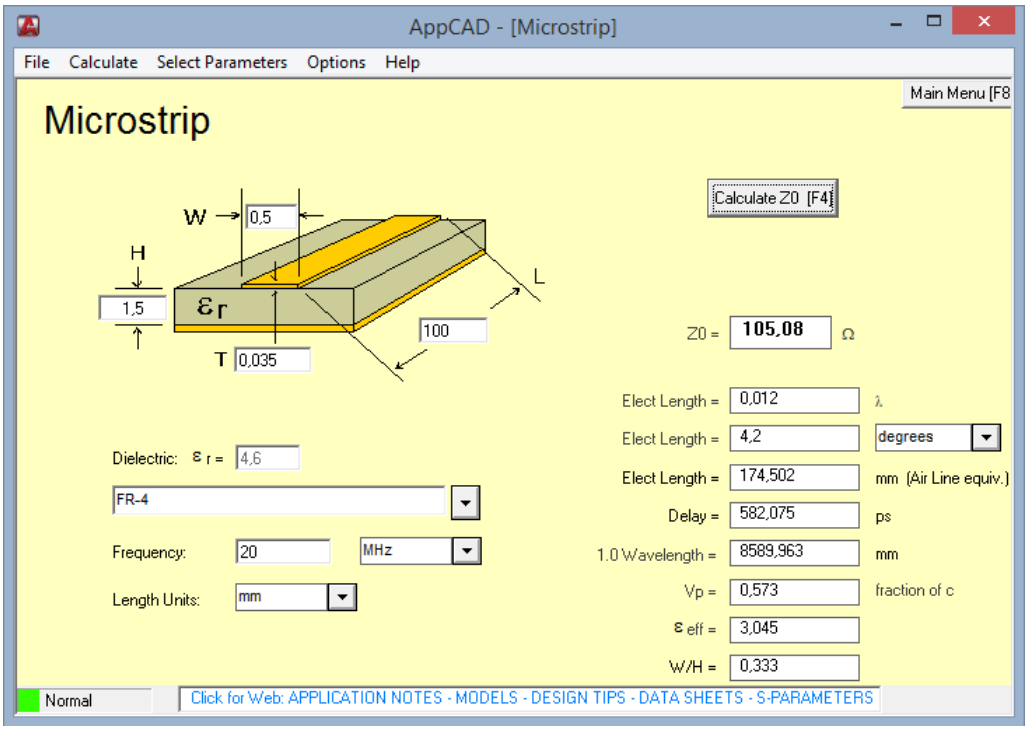


Figure 4.35: AppCAD screenshot for a microstrip PCB trace.

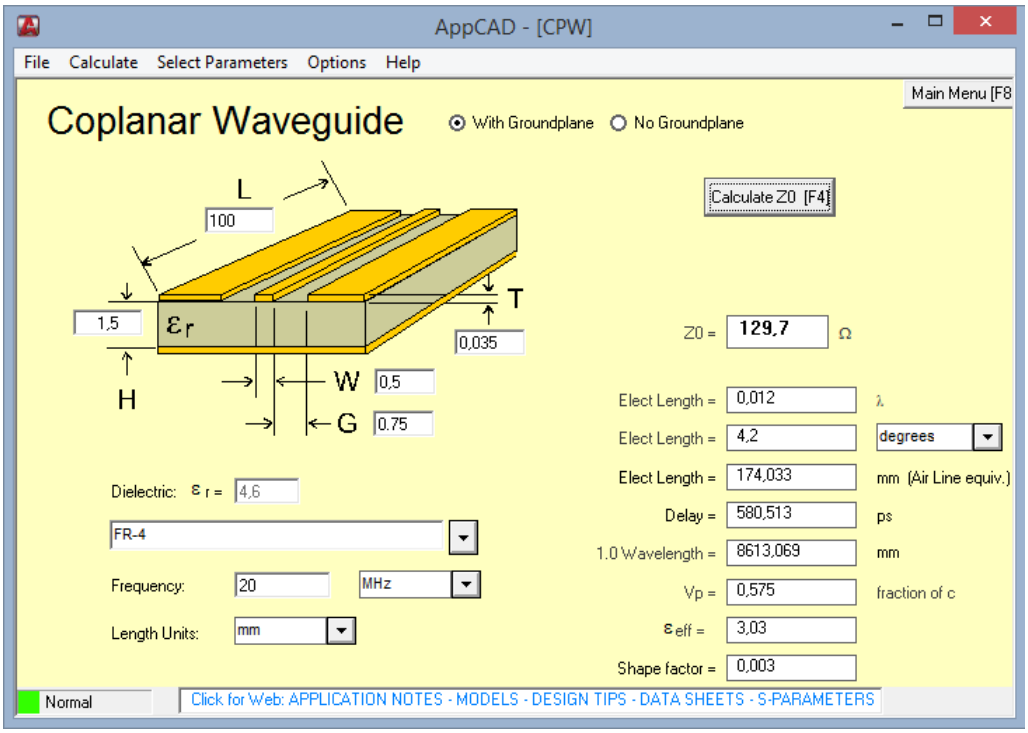


Figure 4.36: AppCAD screenshot for a coplanar PCB trace.

As is seen in the two figures, AppCAD also gives an electrical length (174.502$[mm]$). This value is presumably calculated from the relationship shown in equation (4.8) (Horowitz and Hill, 2015).

$$L_{electric} = L_{physical} \cdot \sqrt{\epsilon_{eff}} \tag{4.8}$$

Where $L_{physical}$ is the physical length of the transmission line. This relationship, is only dependent on the geometry of the transmission line and not the actual signal wave, propagating through the transmission line. The physical length $L_{physical}$ (L in AppCAD) is set to an arbitrary 100$[mm]$, as it has no influence on the two parameters of interest, ϵ_{eff} and Z_0.

The electrical length is instead calculated from the rise time of the signal out of the driver and the velocity of the signal propagating through the transmission line, as shown in equation (4.6) and equation (4.7). There are three different rise times to be determined:

- 20$[MHz]$ signal out of the sbRIO
- 20$[MHz]$ signal out of the clock buffer
- 10$[MHz]$ signal out of the $\Delta\Sigma$ Modulators

The purpose of these signals and components and their sizes are described later in the chapter. Rise/fall time of the output is given for the clock buffer from the datasheet, but not for the other two. They are therefore measured with an oscilloscope, as shown in figure 4.37.

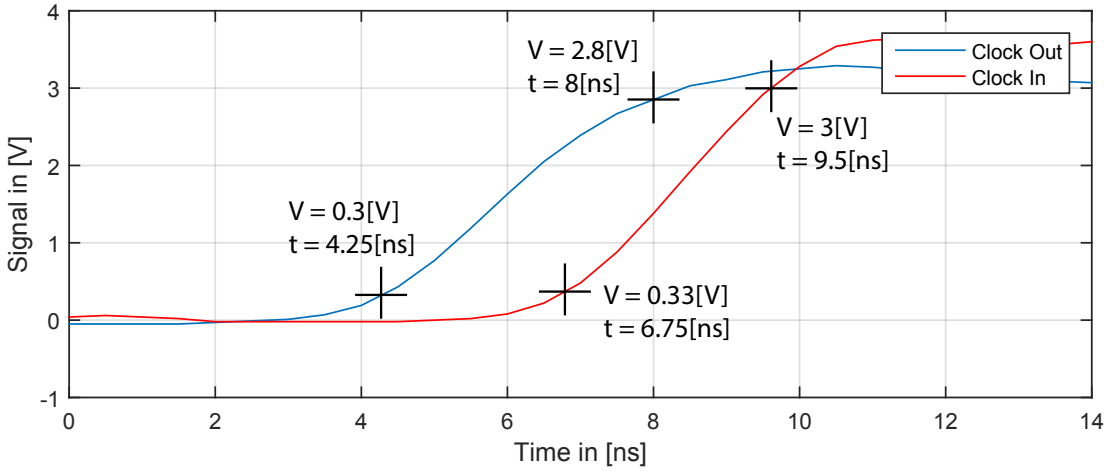


Figure 4.37: Rise time of the 20$[MHz]$ out of the sbRIO (Clock Out) and out of the clock buffer (Clock In).

The rise time is calculated as the time it takes the signal to reach 90% of its final value from 10%. This is done for both rise and fall time of all three signals and shown in table 4.7.

Signal (out of)	Rise time	Fall time	Unit
sbRIO	3.75	3	$[ns]$
Clock buffer	2.75	2.5	$[ns]$
ΔΣ Modulators	15	20	$[ns]$

Table 4.7: Rise and fall times of three fast logic components on the IB2.

The results of the two different PCB trace widths calculated as both a microstrip and as a coplanar waveguide, along with the maximum trace length, before they are to be considered transmission lines, are listed in table 4.8.

Component	Trace Geometry	ϵ_{eff}	$L_{electric}$		L_{max}	
sbRIO	Microstrip W = 0.5$[mm]$	3.045	295.362	$[mm]$	49.227	$[mm]$
	Coplanar W = 0.5$[mm]$	3.03	296.824	$[mm]$	49.471	$[mm]$
Clock buffer	Microstrip W = 0.25$[mm]$	2.949	254.148	$[mm]$	42.358	$[mm]$
	Coplanar W = 0.25$[mm]$	2.47	303.434	$[mm]$	50.572	$[mm]$
ΔΣ Modulators	Microstrip W = 0.5$[mm]$	3.045	1.477	$[m]$	246.135	$[mm]$
	Coplanar W = 0.5$[mm]$	3.03	1.484	$[m]$	247.354	$[mm]$

Table 4.8: ϵ_{eff} and L_{max} calculated for different PCB traces on the IB2.

The actual lengths of the PCB tracks according to the listed categories in table 4.8 are listed in table 4.9.

Component	Trace Name	L_{actual}	
sbRIO	CLKO	109.721	$[mm]$
Clock buffer	CLKI	111.471	$[mm]$
	CLK VMWU	64.013	$[mm]$
	CLK CMU	31.494	$[mm]$
	CLK VMUV	50.443	$[mm]$
	CLK CMV	55.386	$[mm]$
	CLK VMVW	102.981	$[mm]$
	CLK CMW	130.771	$[mm]$
ΔΣ Modulators	DATA VMWU	160.433	$[mm]$
	DATA CMU	131.383	$[mm]$
	DATA VMUV	107.948	$[mm]$
	DATA CMV	95	$[mm]$
	DATA VMVW	144.978	$[mm]$
	DATA CMW	130.771	$[mm]$

Table 4.9: Actual length of the PCB traces on the IB2.

It is apparent from table 4.8 and table 4.9, that according the the guidelines set by (National Instruments Termination, 2014), the $CLKO$ out of the sbRIO should be treated as a transmission line, as well as all Clock buffer outputs, except for one, but it is not necessary for the traces out of the $\Delta\Sigma$ Modulators. The IB2 is initially designed without termination of the transmission lines and modifications to the board, would require a new board to be created. Many of the PCB trace lengths are on the edge of the guideline and the signal itself is inspected for whether it is assessed necessary to apply termination on the traces. In figure 4.38 the signal out of the sbRIO (Clock Out on the figure and CLKO in the table) measured at receiver pin, on the clock buffer, and a signal out of the clock buffer (Clock IN on the figure and CLKI in the table) measured at the receiver pin, on the sbRIO, are shown. According to the guidelines, both these signals should be terminated as they are approximately twice as long as the maximum length before they should be treated as transmission lines.

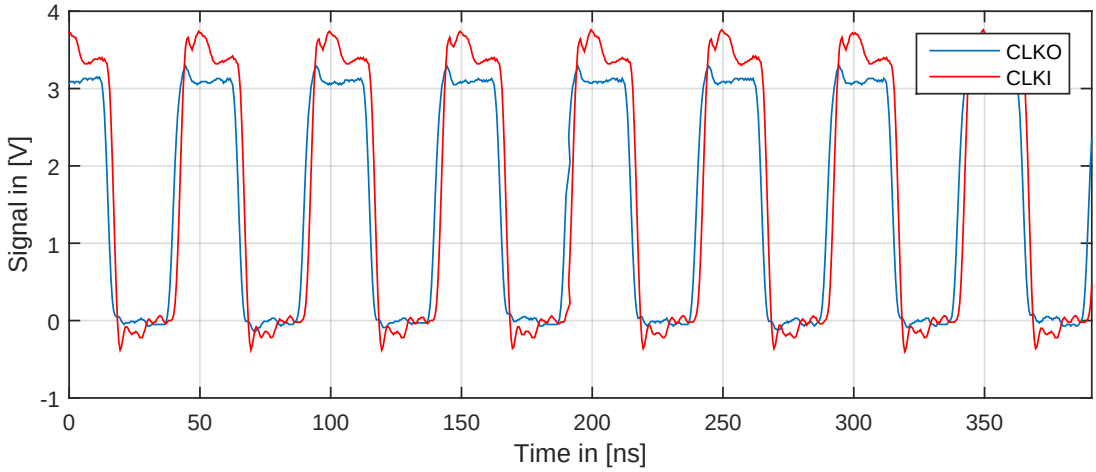


Figure 4.38: Quality of the 20$[MHz]$ clock signal out of the sbRIO (CLKO) and out of the clock buffer (CLKI).

The Clock Out signal is clear, and displays very little ringing, having an overshoot of 10% over its settling voltage of 3$[V]$. This could be due to the sbRIO having partly a matched output, as the pins are terminated to 55[Ω] and the trace has an impedance somewhere in between 105.08[Ω] (figure 4.35) and 129.7[Ω] (figure 4.41). The Clock In signal, however has clear ringing, most likely due to reflections of the signal, as the output is not terminated. The ringing on the Clock In signal is assessed not to pose a significant problem. The ringing only presents an overshoot of the voltage, and no undesired HIGHs are present. In addition the maximum input voltage for the sbRIO is 5.25$[V]$, so the overshoot will not damage the receiver.

To inspect the worst case scenario of not applying transmission lines on the IB2, CLK CMW is measured at its driver and its receiver. CLK CMW has a trace length of 130.771$[mm]$, while it should be no longer than 42.358 – 50.572$[mm]$ before termination should be

applied. In addition, the clock buffer, the driver of this signal, has no internal impedance matching, while AppCad estimated the trace impedance at 116[Ω] for a Coplanar Waveguide and at 127.37[Ω] for a microstrip. The signal measured at the buffer driver and at the ΔΣ Modulator receiver is shown in figure 4.39.

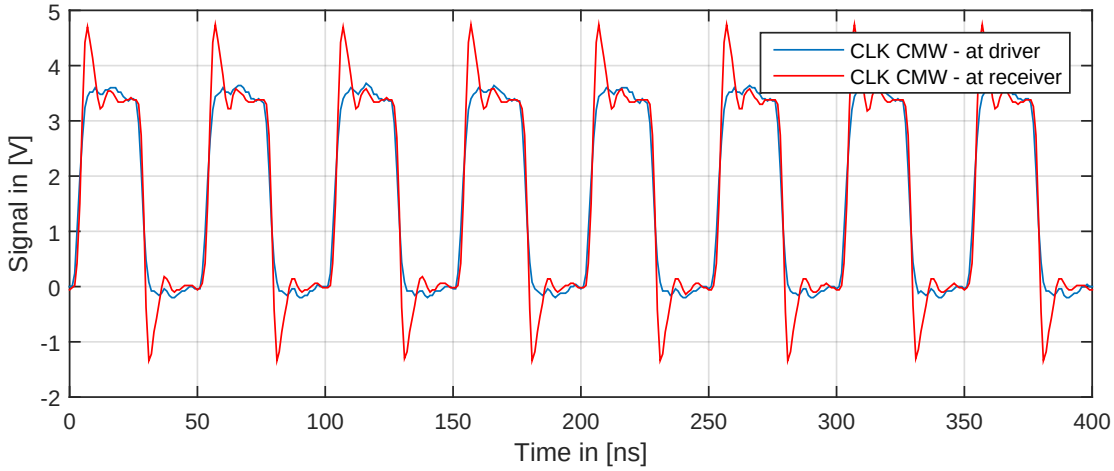


Figure 4.39: Quality of the 20[MHz] clock signal out of the buffer and at the ΔΣ Modulators input.

The effects of not terminating the signal is clearly seen at the receiver end of this signal. The voltage overshoot is up to 40%, reaching 4.7[V], while the maximum input voltage for the ΔΣ Modulator is 3.8[V] (Analog Devices AD7403, 2014). There are no false HIGHs present on the signal, but the voltage overshoot could potentially damage the ΔΣ Modulator over time. For further inspection, the data signal out of the ΔΣ Modulator is shown in figure 4.40.

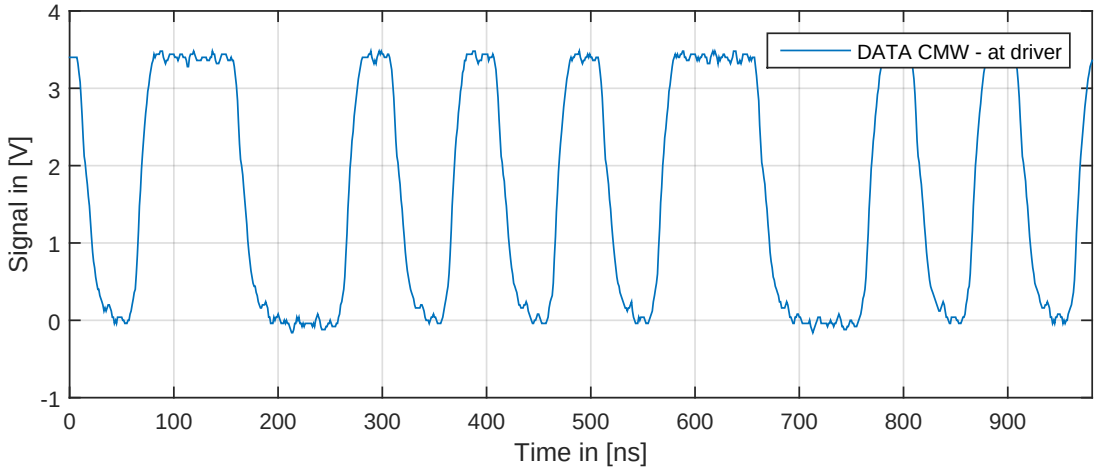


Figure 4.40: Data signal out of the CMW ΔΣ Modulator.

The signal out of the ΔΣ Modulator is a clear clocked digital signal, and does not looked distorted. The double width signals are expected, and are explained in section 4.5 "*Delta Sigma Modulator*". It is assessed however, that this PCB trace needs to be terminated, in order to protect the ΔΣ Modulator from potential damage. Reducing these voltage spikes also reduces the chance of them affecting other circuits on the IB2. Due to the time scope of the project, making a new version of the IB2 is not possible. An alternative would be a "quick fix", which entails breaking the connection of the clock buffer pin to the PCB trace, and solder on a series resistor between the pin and the trace. However, a "quick fix" of this particular trace on the IB2 is problematic, as the trace is 0.25$[mm]$ wide and runs parallel to another 0.25$[mm]$ wide trace, as shown in figure 4.41.

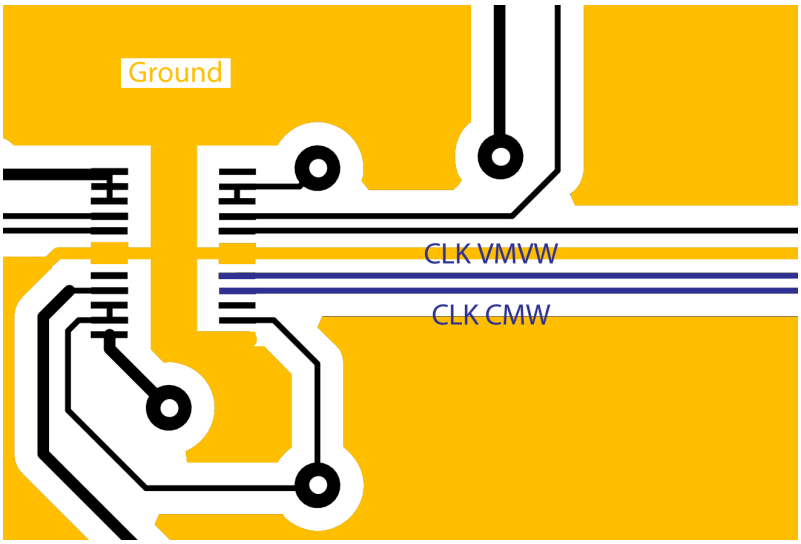


Figure 4.41: Traces out of the clock buffer. CLK CMW and CLK VMVW are both 0.25$[mm]$ wide, and separated by 0.4$[mm]$.

Although control of the motor is out of scope for this project, all three phases are still measured, so that the hardware may still be applied for control of the motor. For the parameter identification method however, it is only necessary to measure the current of phase U and the voltage of phase U, with respect to phase V. The trace length for CMU is below the L_{max} and does not require trace termination. For these reasons it is decided not to alter the IB2, and continue the parameter identification with CMU and VM UV. For future work that involves control of the motor, considerations should be taken in regard to either modifying the IB2 or making a new board. This should also include either a test of whether the 3$[mm]$ distance on the DC-DC converter, separating the high-voltage side to PE, is adequate for creepage or if perhaps new sources of power should be considered.

4.4.2 Current and Voltage Measurements

With the general signal and power circuits explained, the following describes the circuits for current and voltage measurements of the VLT output phases. In order to measure the currents and voltages of phases U,V and W, ΔΣ Modulators are used as ADCs. How the ΔΣ Modulator work, is explained in section 4.5 "*Delta Sigma Modulator*". The ΔΣ Modulators have a linear input range of $\pm 250[mV]$, the rated current of the motor is $43[A]$ and the peak phase voltage (V_{peak}) of the PWM out of the inverter is $566[V]$, thus these must be converted. As Chapter 3 "*Parameter Identification Method*" showed, the frequencies of interest, when performing parameter identification, are up to around $4000[Hz]$. This is thus the bandwidth of the signals of interest.

As the phases enter IB2, the currents are measured as a voltage drop across a shunt resistor, and the voltages are measured one phase with respect to another, using a voltage divider. It is chosen to measure the phase voltages with this delta configuration in order to make a more flexible solution, that can be applied to any three phase inverter, and not require the $DC-$, often not available. The circuit for converting the high voltages and currents to low voltage input for the ΔΣ Modulators is shown in figure 4.42.

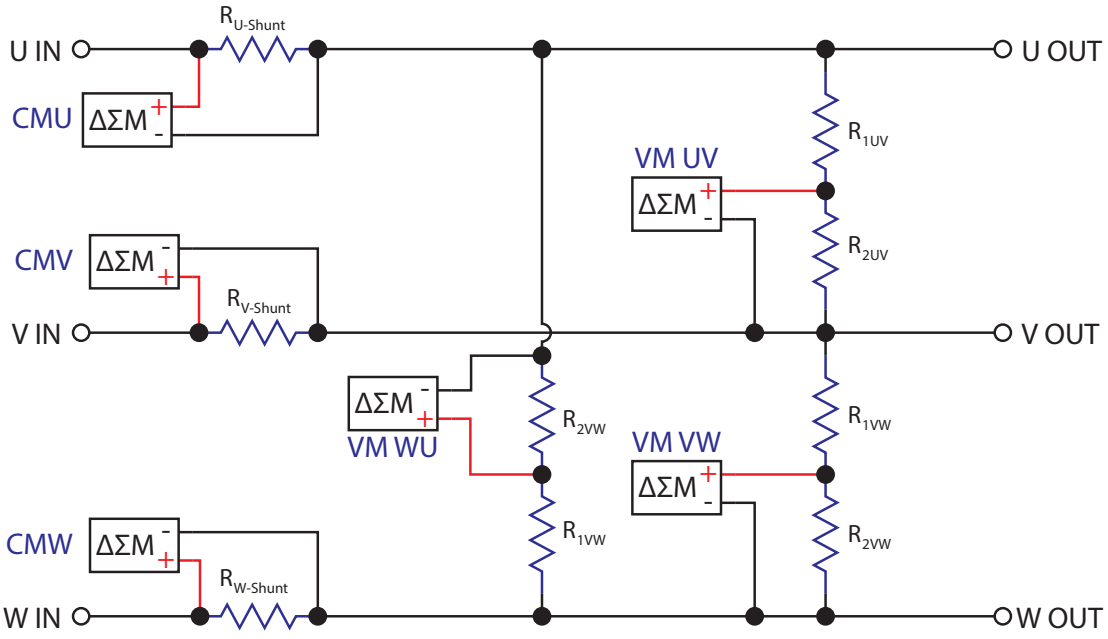


Figure 4.42: Measurement circuit overview, the ΔΣ are the ADCs and in the voltage dividers, R_1 is the larger resistor, while R_2 is the lower.

As the figure illustrates, phases U, V and W enter the board, pass through a shunt resistor and leave the board again. The voltage dividers allow the phases to be measured with respect to one another.

In the following, the current measurement as a voltage drop across the shunt resistor is explained. This is followed up by the explanation of the voltage divider used to down-scale the phase voltages for measurements. Lastly, the ΔΣ Modulator is described and it is explained how it is used as an ADC.

Shunt resistor

The shunt resistor must be chosen such that the voltage drop across the resistor is within the range of $\pm 250[mV]$ for the expected current range. The input to the ΔΣ Modulator can be registered up to $\pm 320[mV]$, but is only linear within $\pm 250[mV]$ (Analog Devices AD7403, 2014). The purpose of the shunt resistor is to measure the current for parameter identification, but also for future control schemes. It is therefore wished to be able to accurately measure currents up to the nominal $43[A]$ of the motor.

A shunt resistor with the value $5[m\Omega]$ is chosen, giving linear range with the ΔΣ Modulator up to $\pm 50[A]$ (ARCOL AP5025). The chosen resistor is an ARCOL AP5025 and relevant specifications are listed in table 4.10.

Resistance	5	$[m\Omega]$
Power Rating	8	$[W]$
TCR	±50	$[\frac{ppm}{°C}]$
Max current (for 2.5[s])	126	$[A]$
Series Inductance	5	$[nH]$
Operating temperature	-55 to 175	$[°C]$

Table 4.10: Specifications for the ARCOL AP5025 shunt resistor (ARCOL AP5025).

The resistor is only capable of dissipating up to $8[W]$, corresponding to $40[A]$. This means that at close to nominal current of the motor, the resistor reaches its max power rating. The resistor for current measurement lies on a price border. Shunt resistors for current measurements are available in the size range of $2-8[W]$ with a resistance of $0.005-0.01[\Omega]$ and cost around 15 DKK (Mouser Electronics TGHG). From these, there is a large gap, up until the shunt resistors apply a Kelvin Bridge and are able to dissipate $60-100[W]$, and the price rises up to around 170 DKK (Mouser Electronics AP5025). A kelvin bridge resistor is a four terminal resistor, that ensures that any resistance in the solder or connection of the resistor, does not influence the voltage drop, as illustrated in figure 4.43 (Horowitz and Hill, 2015).

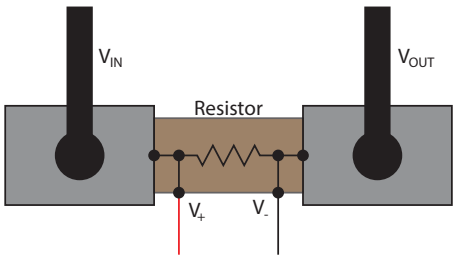


Figure 4.43: Kelvin bridge connection.

The ARCOL AP5025 is chosen, because it is assessed that it is sufficient for the application, and that the benefits of the more expensive resistor, do not outweigh the significantly increased price.

Having chosen the shunt resistor for current measurement, there are two things to be vary of:

1. Frequency response
2. Temperature

The frequency response of the resistor is important when dealing with large $\frac{di}{dt}$. At high enough frequencies, the inductance of the resistor becomes an impedance, and effectively changes the resistance of the resistor. The frequency response for the AP5025 resistor is given in the datasheet and shown in figure 4.44.

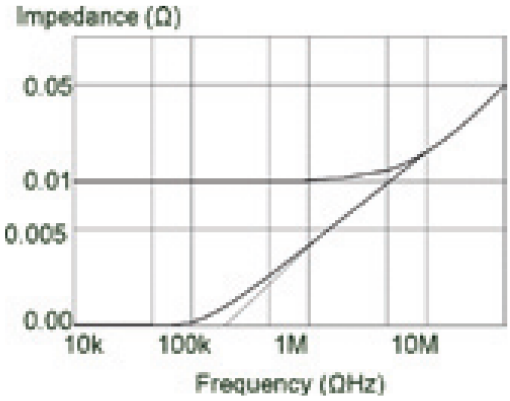


Figure 4.44: The frequency response of the AP5025 resistor from the datasheet (ARCOL AP5025).

The first thing to notice on the graph, is that there are three different lines and that the frequency is written in $[\Omega Hz]$. There is no explanation for the graph, but it assessed reasonable to deduce from the graph that the impedance is constant up to around $100[kHz]$. The switching frequency of the current to be measured is in the order of $10[kHz]$, not nearly high enough to be disturbed by the impedance of the shunt resistor.

At certain temperatures the resistance of the resistor can change value. This is given as the TCR (Temperature Coefficient of Resistance) of the resistor, with the unit $\left[ppm/^{\circ}C\right]$. The TCR for the chosen resistor is given as $\pm 50\left[ppm/^{\circ}C\right]$ at $25[^{\circ}C]$. This means that if the temperature of the resistor changes to for example $70[^{\circ}C]$, the resistance will change to:

$$5[m\Omega] \cdot \left(1 + 50\left[\frac{10^{-6}}{^{\circ}C}\right]\left(70[^{\circ}C] - 25[^{\circ}C]\right)\right) = 5.011[m\Omega] \tag{4.9}$$

If the resistor reaches its max of 175[$^\circ C$], then at 40[A] the voltage error as a result of the change in resistance is 1.5[mV], corresponding to an error of 0.3[A]. This error is minimal and it is assessed that this will not influence the measurements. The operating temperature for the resistor is $-55[^\circ C]$ to 175[$^\circ C$].

The datasheet gives recommended PCB design for dissipating heat from the resistor. The recommended PCB design only applies up to 4.6[W] and requires a copper pad on each side of the resistor to be 2700[mm^2]. This is not enough power, nor is it a practical pad size.

Instead, an active cooling solution has been designed for the resistor. The cooling solution is shown in figure 4.45 on the bottom side of the board. The cables on the picture are the phases from the inverter. On the other side of the board, after passing through the shunt resistor, the phases leave the board again.

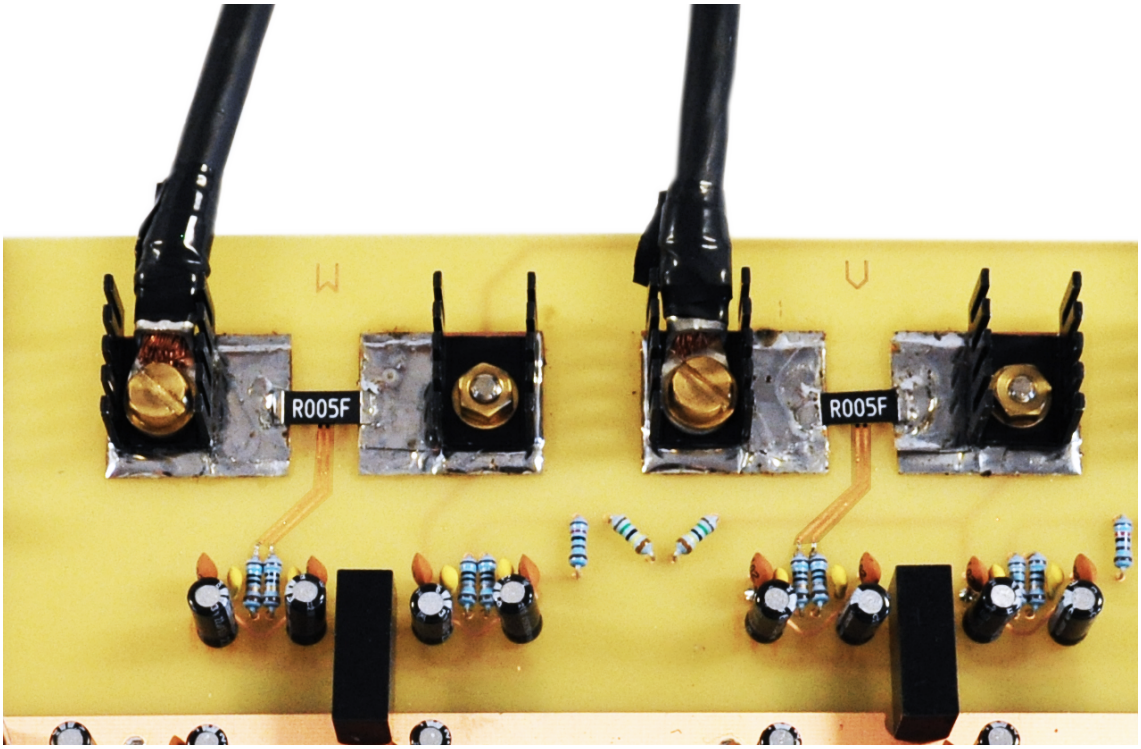


Figure 4.45: Shunt resistor on the IB2, with the cooling solution.

Heat sinks are placed close to the terminals of the resistor, and using a fan to create an airflow across them, it is assessed that there is enough cooling on the resistors, and the bottleneck for dissipating power is to be found in the resistor. The PCB surface area of the pads at each side of the shunt resistor are only 480[mm^2], while the total surface area of each heat sink is approximately 1300[mm^2]. A thermal analysis of the resistors and the cooling solution has not been conducted. For future work, a thorough analysis of the heat dissipation capabilities, of the solution, should be conducted, or perhaps

another solution should be designed, as the solution implemented is expected to be over-dimensioned.

A cross sectional view of the cooling solution is shown in figure 4.46.

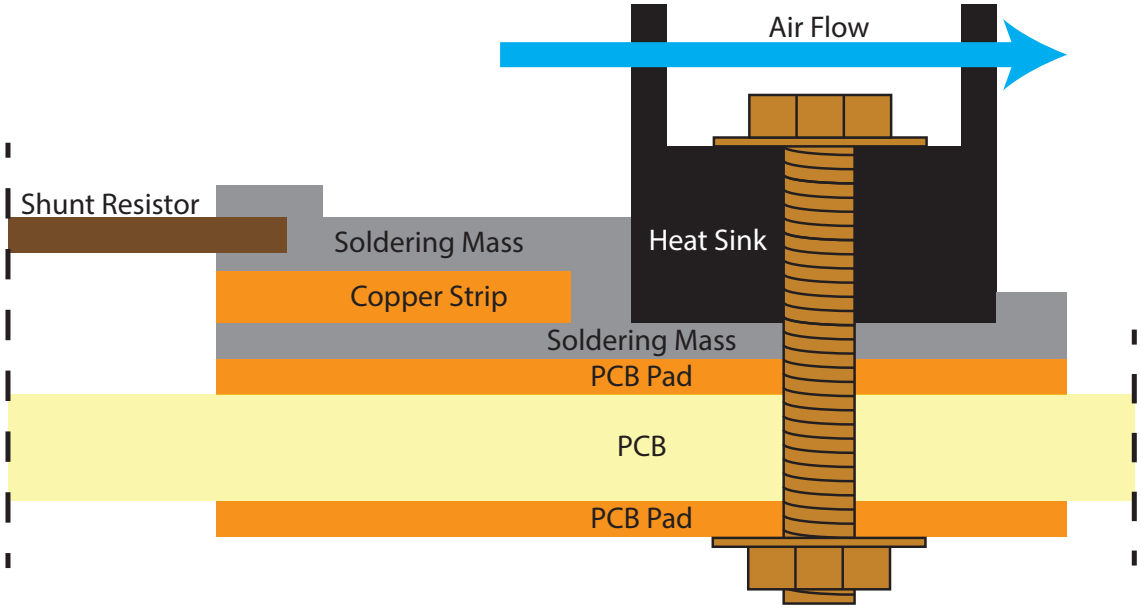


Figure 4.46: Heat dissipation strategy for the shunt resistor. Not to scale.

Before soldering the shunt resistor to the board, a copper strip is soldered onto the pad. This raises the shunt resistor from the dielectric material of the PCB, lowering the stray inductance of the resistor. In addition, there is more conducting material to lead the high currents from the cables, bolted on the heat sink, through the resistor and onto the the pad on the other side of the resistor. PCB traces run beneath the resistor, and although it is laminated, raising the resistor, helps to ensure that it is isolated from the trace. The traces beneath the resistor are shown in figure 4.47.

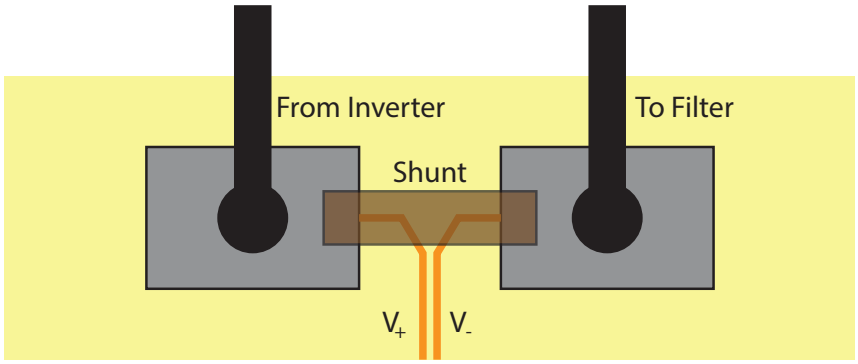


Figure 4.47: PCB traces beneath the shunt resistor, for current measurement.

The traces running beneath the shunt resistor run to the input pins of the analog ΔΣ Modulator. There are several reason for this layout. First of all, the quasi-Kelvin connection provide as close contact as possible to the leads of the shunt resistor. This is done to ensure that the voltage drop is only measured across the shunt resistor, and not the soldering tin. This ensures the low inductance of the resistor and the low TCR (Avago Shunt, 2010). Another advantage of running the two traces very close together, is that it reduces the "coil effect" of the trace and thereby the effect of mutual inductance on the signal.

Voltage divider

As described the voltages are measured phase-to-phase, however these phases are PWM voltages out of the inverter. This means that two phases measured with respect to one another, will either give $\pm V_{peak}$ or $0V$, plus possible effects from the dynamics of the system. The inverter is rated to $600[V]$ output voltage and this is equal to $\pm V_{peak}$. Therefore the voltage divider must be dimensioned such that voltage range of the phases are within the linear voltage range of the ΔΣ Modulator. A single phase voltage divider, from figure 4.42, is shown in figure 4.48, where the input-output relation is described by equation (4.10).

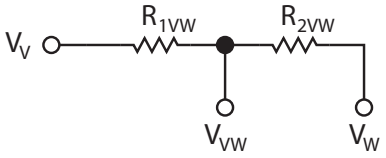


Figure 4.48: Voltage divider for voltage measurement V_{VW}

$$V_{VW} = \frac{R_{2VW}}{R_{1VW} + R_{2VW}} \cdot V_V \tag{4.10}$$

The voltage scaling for this application is relatively high and the size of the resistors have to be considered. If the resistances are too high, there is a risk of V_{VW} floating, but the lower the resistance values, the more power is dissipated in the resistors. The resistors chosen are $R_1 = 3[M\Omega]$ and $R_2 = 1.2[k\Omega]$ making it possible to measure the phase-to-phase voltages linearly up to $\pm 625.25[V]$. The chosen resistors for the voltage divider have a rated power dissipation of $0.6[W]$ and a max operating voltage of $350[V]$. (VISHAY MRS25) R_1 is therefore split into two series coupled resistors of $1.5[M\Omega]$ each, keeping the total resistance the same, while halving the voltage drop for each resistor. At $625.25[V]$ $R_{1,1}$ and $R_{1,2}$ will only be subject to $0.065[W]$ each and R_2 to $0.000052[W]$, both well below the capability of the resistors. The chosen resistor is VISHAY MRSS25 and the relevant data is listed in table 4.11.

Resistance	1.5M and 1.2k	[Ω]
Power Rating	0.6	$[W]$
TCR	±50	$[\frac{ppm}{°C}]$
Max Voltage	350	$[V]$
Resistance tolerance	1	[%]

Table 4.11: Specifications for the VISHAY MRSS25 voltage divider resistors (VISHAY MRS25).

Since the power dissipation of the resistors are so low, a change in resistance due to temperature is not expected. The resistance tolerance makes little difference, since any deviation would be permanent, and the ΔΣ Modulator needs to be calibrated anyway. The worst case deviation would result in a linear voltage measuring range of either $\pm 613[V]$ or $\pm 695[V]$, with the lowest still being within the range of the voltage to be measured. It has not been possible to obtain the frequency response of the chosen resistors, which could potentially influence the voltage measurements, as the voltages have a low rise time and very high $\frac{dV}{dt}$. This could be inspected for future work, but will not be addressed any further in this project.

4.5 Delta Sigma Modulator

With the circuits for current and voltage measurements described, the ΔΣ Modulator and how it is used as an ADC is explained. The ΔΣ Modulator used for both current and voltage measurements is the AD7403 with the specifications listed in table 4.12.

Measuring input max	320	$[mV]$
Measuring input linear	250	$[mV]$
Clock input max	20	$[MHz]$
Resolution	16	$[Bit]$
Effective-Number-of-Bits (ENOB)	14.2	$[Bit]$
Signal-to-Noise-and-Distortion Ratio (SINAD)	87	$[dB]$
Signal-to-Noise Ratio (SNR)	98	$[dB]$
Working isolation voltage	1200	$[V]$

Table 4.12: Specifications for the AD7403 ΔΣ Modulator (Analog Devices AD7403, 2014).

Before explaining how the ΔΣ Modulator works, some general values that describe the performance of an ADC are explained.

The Signal-to-Noise Ratio (SNR) is the ratio between the RMS of the signal and the RMS of the noise at the ADC output, up to half the sampling frequency (Analog Devices AD7403, 2014). The SNR is a function of the resolution of the ADC. Thus for an ideal N-bit converter the SNR is given by equation (4.11).

$$SNR = (6.02N + 1.76)[dB] \tag{4.11}$$

For the 16-bit AD7403 the SNR is thus 98$[dB]$. However, the Effective-Number-of-Bits (ENOB) is a measure of the resolution, taking noise and distortion into account, and is calculated as:

$$ENOB = \frac{(SINAD - 1.76)}{6.02}[bits] \tag{4.12}$$

Where the Signal-to-Noise-and-Distortion Ratio (SINAD) is the SNR taking distortion and harmonics into account. This value is measured during testing by the manufacturer, influenced by the sampling frequency, the frequency of the signal any noise that might be present and harmonics. However, other factors also come into effect, namely the amplitude of the input signal. In figure 4.49, the influence of the signal amplitude on the ENOB for the AN7403 is shown.

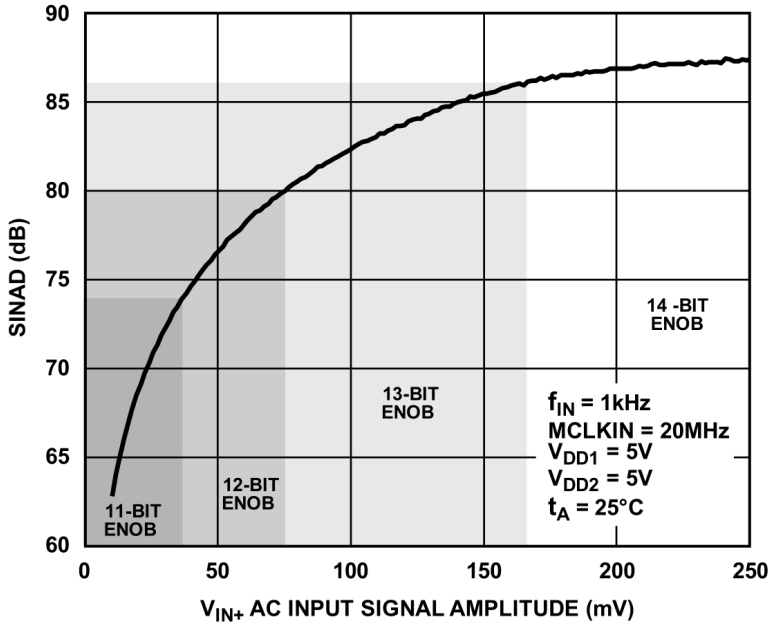


Figure 4.49: Influence of signal amplitude on the ENOB (Analog Devices AD7403, 2014).

Another parameter worth mentioning is the Over-Sampling-Ratio (OSR). The OSR is the ratio between the frequency of the signal to be measured, and the sampling frequency, taking the Nyquist frequency into account. The OSR is thus calculated as:

$$OSR = \frac{f_{sampling}}{2f_{signal}} \tag{4.13}$$

A $\Delta\Sigma$ Modulator uses the concept of oversampling to achieve a high resolution, noise reduction and avoid aliasing. How this is achieved is explained in the following about the $\Delta\Sigma$ Modulator and the associated digital filter.

Theory of operation

An overview of how the ΔΣ Modulator is used as a ADC is shown in figure 4.50

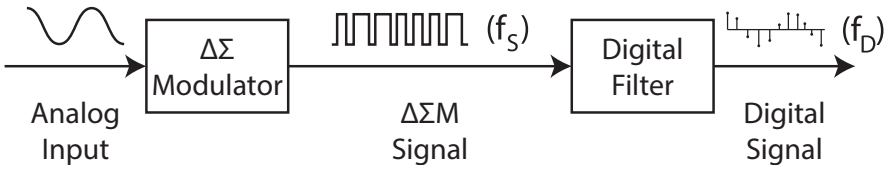


Figure 4.50: Overview of a ΔΣ Modulator converting an analog signal to digital.

The ΔΣ Modulator converts an analog input signal to a 1-bit stream of data. This stream is directly proportional to the input signal, where averaging across them provides a representation of the analog input signal. This is illustrated in figure 4.51.

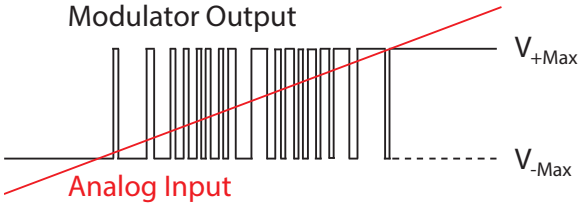


Figure 4.51: 1-Bit output stream of a ΔΣ Modulator, proportional to the analog input. Inspired by a figure from (Analog Devices AD7403, 2014).

This means that a $0[V]$ input, ideally sends an output stream of equally many HIGHs and LOWs, a $-320[mV]$ ideally sends out all LOWs and a $320[mV]$ output ideally sends out all HIGHs. Any input voltage in between, will send out a 1-bit stream with a percentage of LOWs and HIGHs proportional to that voltage. As an example, a constant input voltage of $160[mV]$ will send out a 1-bit stream of 75% HIGHs. The bit stream changes state with an input clock to the ΔΣ Modulator, switching at every clock period. With an input clock of $20[MHz]$, the output stream can thus switch state every $50[ns]$, where the percentage of HIGHs and LOWs come from successive HIGHs or LOWs. At a 50% HIGH/LOW output, the signal thus resembles a $10[MHz]$ square wave. The bit stream is propagated on a rising edge of the clock and valid on the following, resulting in a $50[ns]$ delay (Analog Devices AD7403, 2014).

The chosen ΔΣ Modulator is of 2^{nd} order, but in order to better explain how it works, a 1^{st} order is shown in figure 4.52.

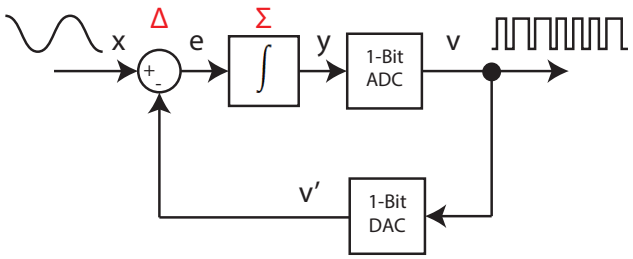


Figure 4.52: 1^{st} Order ΔΣ Modulator.

The name ΔΣ Modulator is apparent, as it consists of a subtractor (Δ), a discrete integrator (thus an accumulator Σ), a comparator (1-bit ADC) and a 1-bit DAC. The ΔΣ Modulator works like a feedback loop, where the bit stream value is converted to a ± analog reference value by the DAC. The subtractor then takes the difference between the input analog signal and the feedback value. The input analog signal is clocked at f_{CLK}, the 20$[MHz]$ input clock, by two highly linear capacitors, shown in figure 4.53 (Analog Devices AD7403, 2014).

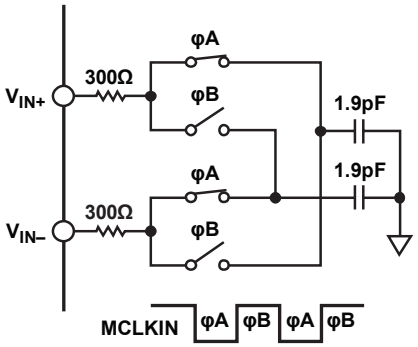


Figure 4.53: Input of the ΔΣ Modulator, figure from (Analog Devices AD7403, 2014).

The integrator works like an accumulator, adding the difference voltage at each clock cycle, which is then converted to a stream of HIGHs and LOWs by the comparator, clocked at f_{CLK}. To illustrate how this works, a simple example is shown in figure 4.54

	x	e	y	v	v'
0	0.2	0.2	0.2	1	1
1	0.2	−0.8	−0.6	0	−1
2	0.2	1.2	0.6	1	1
3	0.2	−0.8	−0.2	0	−1
4	0.2	1.2	1.0	1	1
5	0.2	−0.8	0.2	1	1
6	0.2	−0.8	−0.6	0	−1
7	0.2	1.2	−0.2	1	1

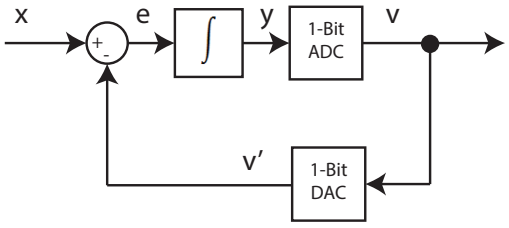


Figure 4.54: Simplified example of how the ΔΣ modulator works.

The example shows the discrete values of each point of the ΔΣ modulator at each time step from 0 to 7. Initial values are set at timestep 0, and at timestep 1 the subtractor outputs the difference $e = -0.8$, the accumulator sums the previous timestep value to the current and $y = -0.6$. The output of the ADC is thus $v = -1$, which is then fed back to the next timestep. This simple example repeats after 5 timesteps and when averaging over them, the output is shown in equation (4.14).

$$\frac{3 \cdot 1 + 2 \cdot (-1)}{5} = 0.2 \tag{4.14}$$

The ΔΣ Modulator can thus be seen as a feedback loop, striving to reduce the error between the analog input signal and the digital output bit stream (Horowitz and Hill, 2015).

However, because the feedback loop jumps between the two extremes, a significant amount of quantization noise is present on the bit stream, from the 1-bit ADC. A common explanation of the benefits of a ΔΣ Modulator, is that it shapes the noise. To inspect how the noise on the ΔΣ Modulator is shaped, a simplified block diagram is shown in figure 4.55.

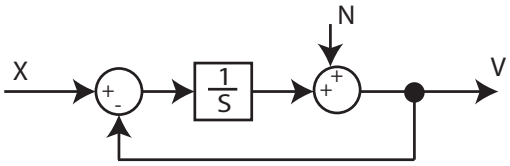


Figure 4.55: Simplified block diagram of a 1^{st} order ΔΣ Modulator.

$$|G_{Sig}| = \frac{V(s)}{X(s)} = \frac{1}{1+s} \tag{4.15}$$

$$|G_{QN}| = \frac{V(s)}{N(s)} = \frac{s}{1+s} \tag{4.16}$$

It should be noted, that in this block diagram of the ΔΣ Modulator, no delays are included. The $50[ns]$ delay has no impact on the explanation of the ΔΣ Modulator and,

as will be shown later, has very little influence on the actual current and voltage measurements. In addition, the delay of the DAC is exempt, for simplification of explaining how the ΔΣ Modulator works. In figure 4.55, the ADC has been replaced by a an added quantization noise N, whose transfer function is shown in equation (4.16), and the transfer function for the signal is shown in equation (4.15). The bode plot for both transfer functions is shown in figure 4.56.

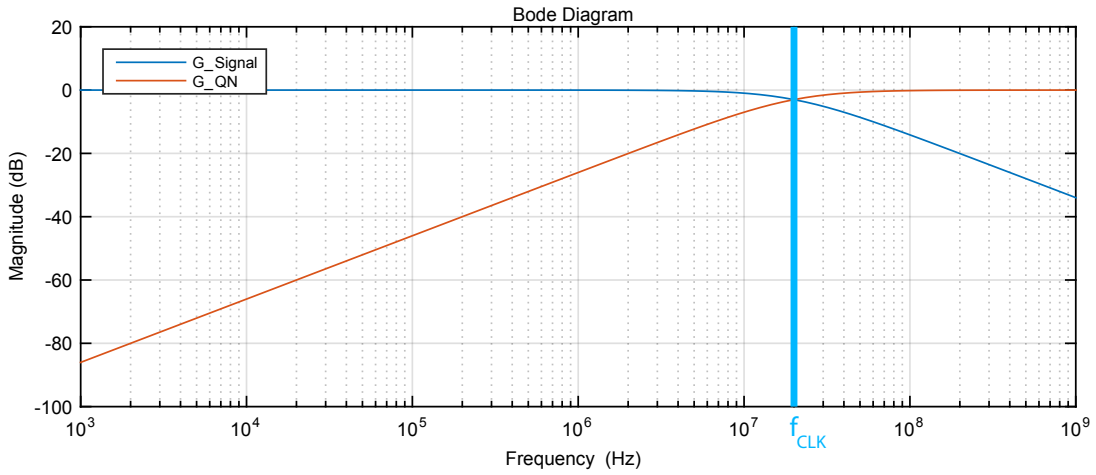


Figure 4.56: Frequency response of the signal and quantization noise transfer functions for a 1^{st} order ΔΣ Modulator.

As shown in the bode plot, the quantization noise acts like a high-pass filter, with a cut-off frequency of $f_{CLK} = f_s$ and the signal acts like a low-pass filter with the same cut-off frequency. Thus, if the signal is oversampled, meaning that the bandwidth of the signal is well below the sampling frequency, then the signal is preserved, while the quantization noise is shaped out of the signal bandwidth. A theoretical equation for determining the ENOB for a ΔΣ Modulator is given as equation (4.17) (Horowitz and Hill, 2015).

$$ENOB_{theo} = log_2(OSR) \cdot \left(m + \frac{1}{2}\right) \tag{4.17}$$

Where m is the order of the ΔΣ Modulator and the OSR is the Over Sampling Ratio. In equation (4.17) it is thus apparent that the ENOB can be increased by either increasing the OSR or the order of the ΔΣ Modulator. The OSR is determined alongside the sampling frequency, when designing the digital filter. Increasing the order of the ΔΣ Modulator has a large influence on the ENOB, and in general it is more common to apply higher order ΔΣ Modulators. The order of a ΔΣ Modulator is increased by cascading additional subtractors and integrators, as shown in figure 4.57.

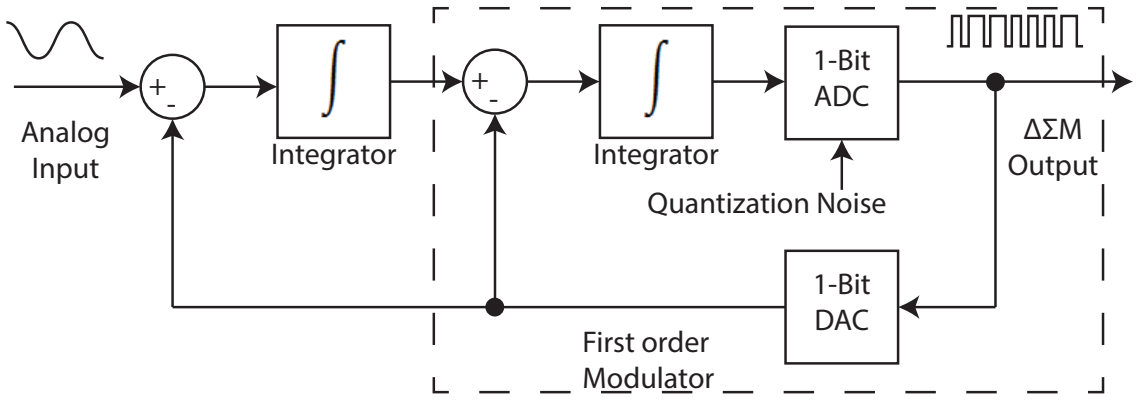


Figure 4.57: 2^{nd} Order ΔΣ Modulator block diagram.

A simplified block diagram with the associated signal and noise transfer functions, under the same circumstances as figure 4.55 is shown in figure 4.58.

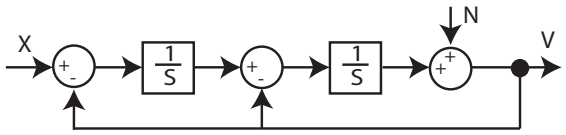


Figure 4.58: Simplified block diagram of a 2^{nd} order ΔΣ Modulator.

$$|G_{Sig}| = \frac{V(s)}{X(s)} = \frac{1}{s^2+s+1} \tag{4.18}$$

$$|G_{QN}| = \frac{V(s)}{N(s)} = \frac{s^2}{s^2+s+1} \tag{4.19}$$

The bode plot for equation (4.18) and equation (4.19) is shown in figure 4.59

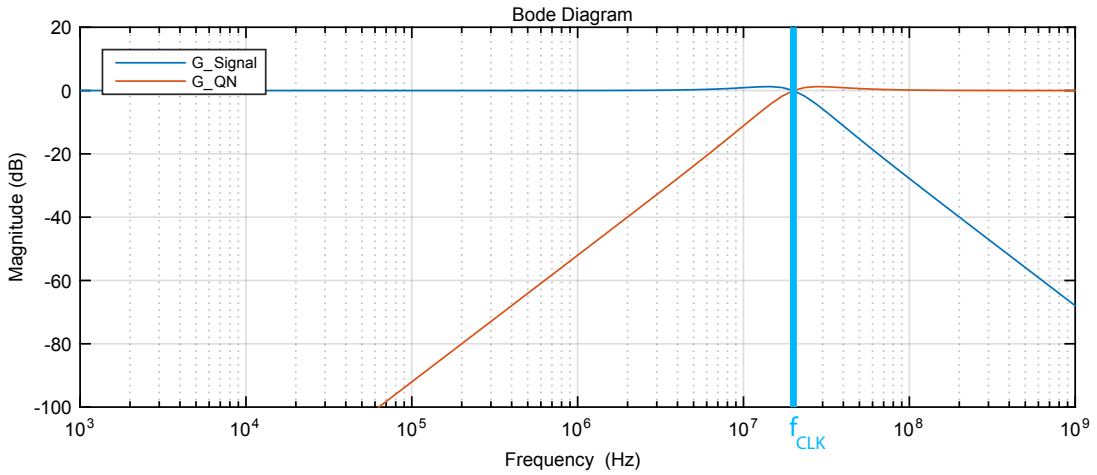


Figure 4.59: Frequency response of the signal and quantization noise transfer functions for a 2^{nd} order ΔΣ Modulator.

The first thing to notice is that the slope of the quantization noise has increased from $20[dB/dec]$ to $40[dB/dec]$, with the same break point, meaning that the noise is even more attenuated at lower frequencies. The performance of a ΔΣ Modulator is dependent on the associated digital filter, that interprets the bit stream. This is explained in section 4.6 "*Digital Filter*", but first the circuit for the AD7403 on the IB2 is described.

4.5.1 ΔΣ Modulator circuits on the IB2

With the currents and voltages of the phases scaled for the ΔΣ Modulator, as described in section 4.4.2 "*Shunt resistor*" and section 4.4.2 "*Voltage divider*", the specific circuit for the AD7403 is shown in figure 4.60, using a current measurement as an example.

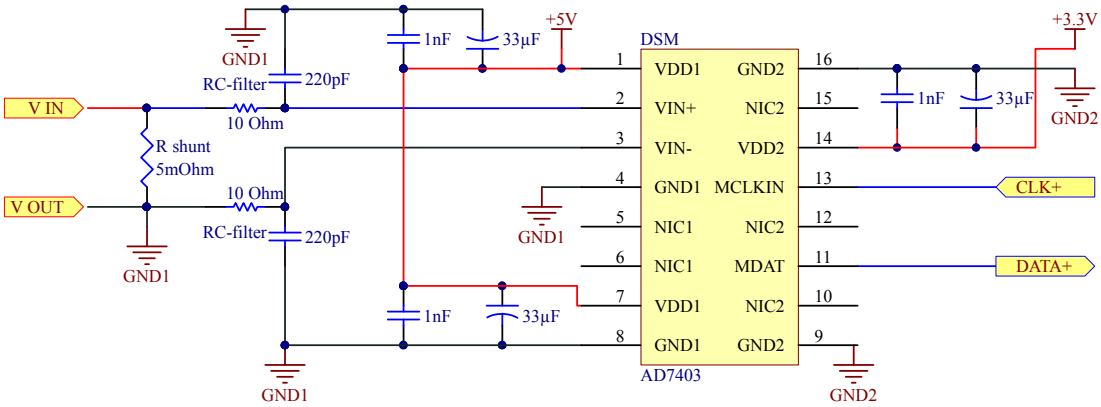


Figure 4.60: ΔΣ Modulator circuit for current measurement.

The AD7403 isolates its analog side from its digital side, where the analog side floats with the high phase voltages. $GND1$ is connected with the output side of the shunt resistor, measuring a positive current as a positive voltage drop. Decoupling capacitors are placed at all three supply pins, to steady the supply voltage. The AD7403 uses a differential input, enforcing the benefits of the quasi-kelvin connection, rather than measuring with respect to $GND1$.

The RC-filters at the analog input ports serve two purposes. The resistors suppress ringing that might be present in the circuit between the resistor and the ΔΣ Modulator (Avago Shunt, 2010). Most importantly the RC-filter functions as an anti-aliasing filter. When following the Nyquist criterion and sampling a signal at twice the signal frequency, undesired signals may be present above the Nyquist frequency. These signals may cause aliasing and therefore a low-pass filter is usually applied to the analog signal, to eliminate components above the signal frequency. This filter is placed before the ADC, making it analog. It would be ideal to cut off all frequencies above the sampling frequency, however because it is analog, there is a transition band, that contains the gradual attenuation of the input frequencies (National Instruments Anti-Aliasing, 2000). In many cases this requires a complicated filter, because the filter must

pass all signal frequencies while attenuating the unwanted frequencies above the signal (Horowitz and Hill, 2015). When heavily oversampling the signal, the Nyquist frequency, is well above the signal that must pass and a much simpler filter can be implemented, without worrying about the transition band. This is a large benefit for the ΔΣ Modulator, as the signal is heavily oversampled. With a sampling frequency of 20$[MHz]$, an analog filter should be implemented to cut off frequencies above 10$[MHz]$. The anti-aliasing filter is a simple RC-filter, with values recommended by (Analog Devices AD7403, 2014), and has the frequency response shown in figure 4.61.

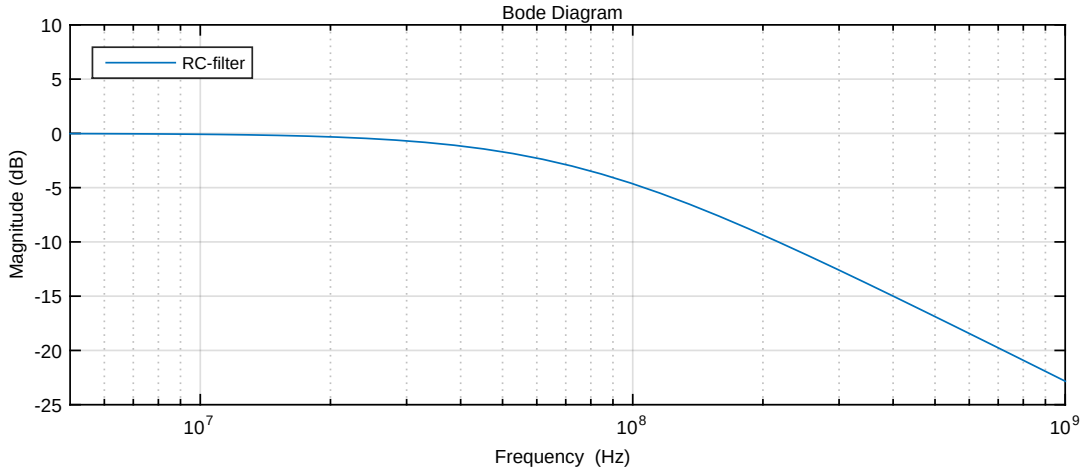


Figure 4.61: Frequency response of the anti-aliasing RC-filter.

The cut-off frequency is 72.343$[MHz]$, but as is seen on the bode plot, the filter starts attenuating frequencies above 10$[MHz]$.

With the ΔΣ Modulator and the circuits to ensure good performance explained, the output of a ΔΣ Modulator on the IB2 is shown in figure 4.62.

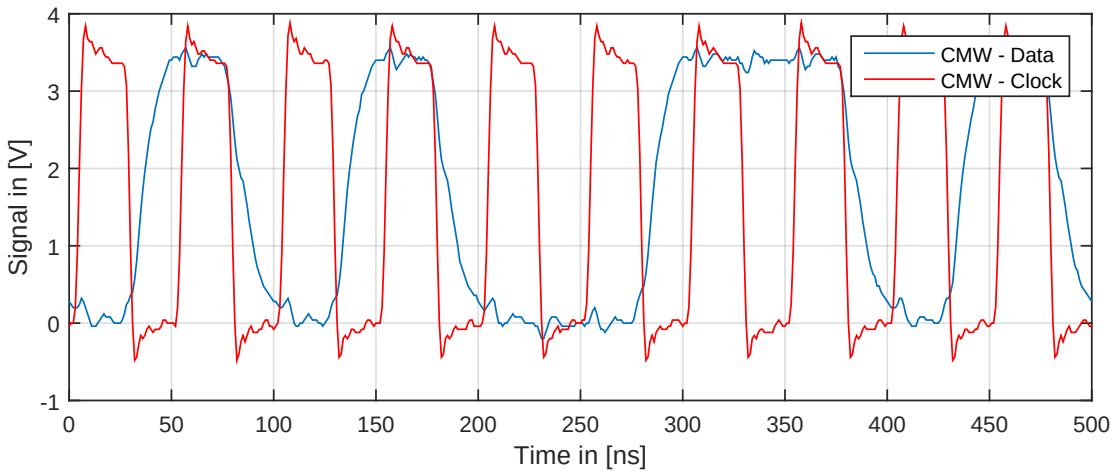


Figure 4.62: 1-bit data stream out of the DSM and the input clock.

The input to the ΔΣ Modulator in figure 4.62 is floating, and it is the signal itself that is of interest. A delay between the clock in and the data out is present, but the signal itself is of satisfactory quality. As described, the data stream switches logic with the clock, and the signal resembles a $10[MHz]$ squarewave. At $275[ns]$ two successive HIGHs are seen, as can be expected if the input to the ΔΣ Modulator is a positive voltage.

In figure 4.63, the 1-bit data stream is shown together with the phase voltage that it is converting.

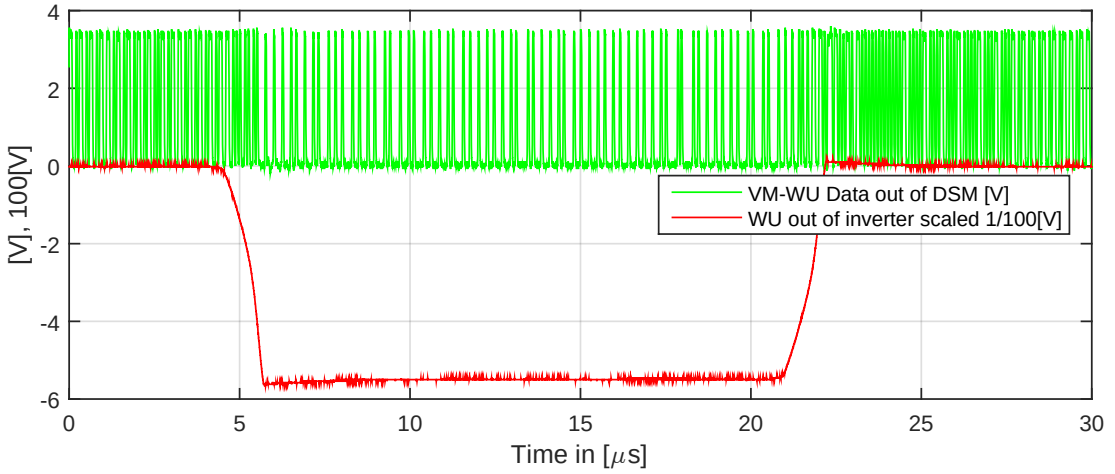


Figure 4.63: 1-bit data stream out of the DSM and the associated phase voltage.

Here it is seen that at $0[V]$ between phase U and W, the output data stream of the ΔΣ Modulator has somewhat equally many LOWs as HIGHs. Applying a filter to this data stream, the output would be a constant value in the middle of the voltage range of the signal. While when the phase voltage goes $-565[V]$, the data stream is mostly LOW, and

a filtering would result in a constant value closer to $0[V]$. Inspection of whether this data stream results in the correct voltage measurements is done in section 4.6 *"Digital Filter"*.

While inspecting the quality of the signals, one phenomenon worth investigation was found. At every second phase-to-phase voltage peak, the output of the data stream goes low for an extraordinary amount of time. This is shown in figure 4.64

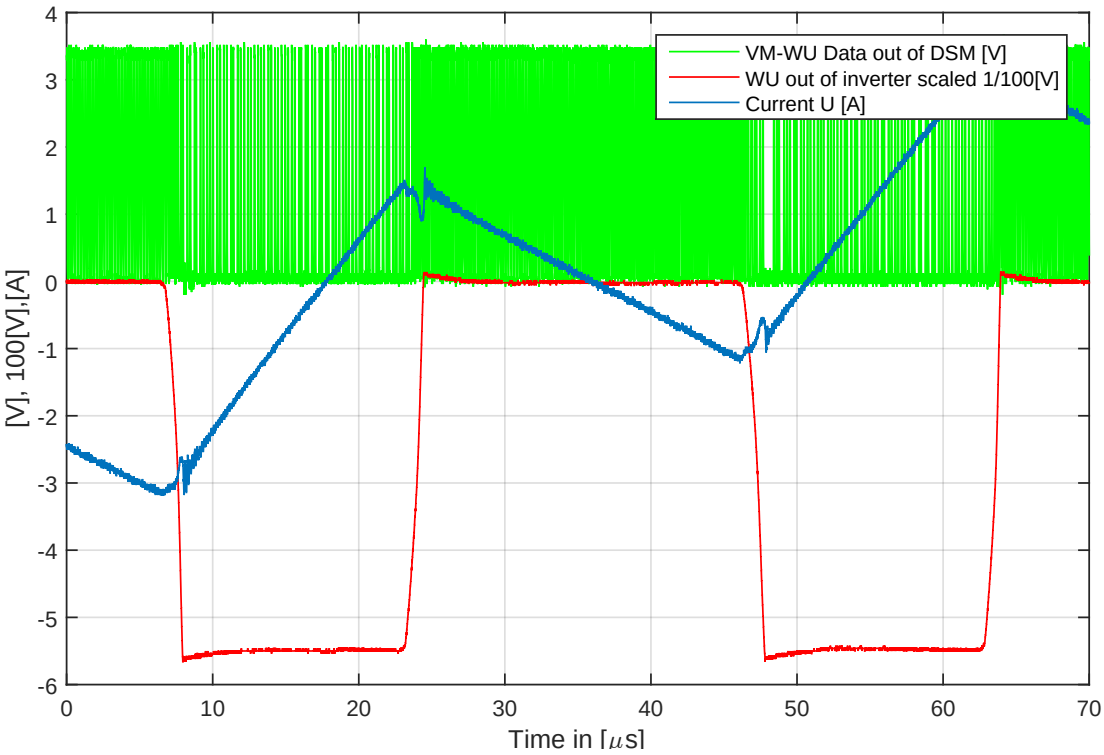


Figure 4.64: 1-bit data stream out of the DSM, the associated phase voltage and phase current.

The measurements are one phase with respect to another, so the two peaks shown in figure 4.64 are part of one PWM cycle. To better understand when this occurs, figure 4.65 illustrates at what times during a PWM cycle this happens.

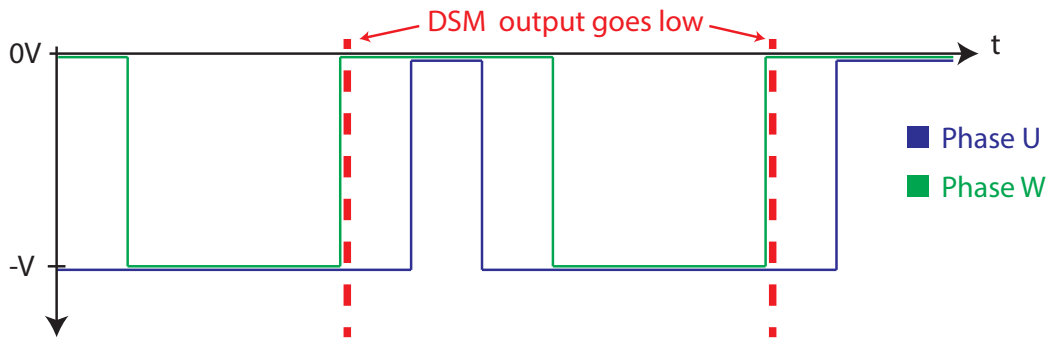


Figure 4.65: Illustration of when the ΔΣ Modulator goes low.

This appears on all voltage measurement bit streams out of the ΔΣ Modulators, when the phase that is measured changes state from a HIGH to a LOW, but does not occur on the current measurements. As this does not occur when the $0[V]$ potential shifts, it is assessed that dynamics in the voltage divider might be the source. This is not inspected any further in this project, and as 4.6 shows, the influence is not noticeable. The voltage divider, should however, be inspected for future work, as there are many unknowns, such as the inductance of the resistors and their associated frequency response.

The next step is designing the digital filter, to interpret the output of the ΔΣ Modulator.

4.6 Digital Filter

The signal from the ΔΣ modulator is a 1-bit data stream, and to retrieve the desired signal from this stream it needs to filtered. The input needs to be sampled at the ΔΣ modulator's output clock of $20[MHz]$. This signal is then filtered to isolate the desired signal, and decimated to output it at a rate that fits the demands for the spectral estimation.

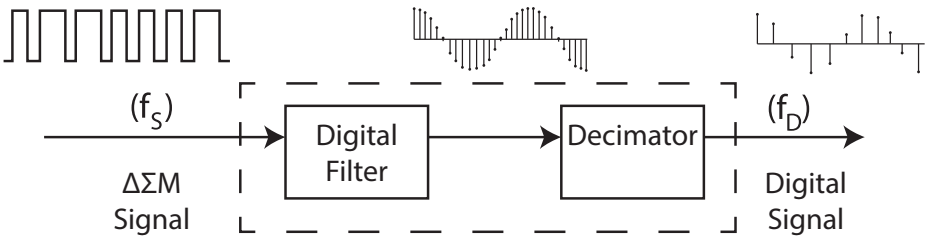


Figure 4.66: The input to the filter is a 1-bit data stream with sampling frequency f_S from the ΔΣ modulator. The digital filter filters and decimates the signal and outputs the signal at data frequency f_D.

The following will include the description of the filters used, and the implementation on the FPGA.

4.6.1 Data Logging

In the process of collecting the data it is essential that the ΔΣ modulators and the FPGA are synchronized. The clock for the ΔΣ modulators is generated on the FPGA, and then returned to the FPGA. This way the bit stream can be synchronized with the data collection VIs. When the FPGA main program is initialized, the data collection VIs are initialized on the first rising edge of the returned clock. After the initial trigger, the data collection VI uses a 20$[MHz]$ clock. The data collection is illustrated in figure 4.67.

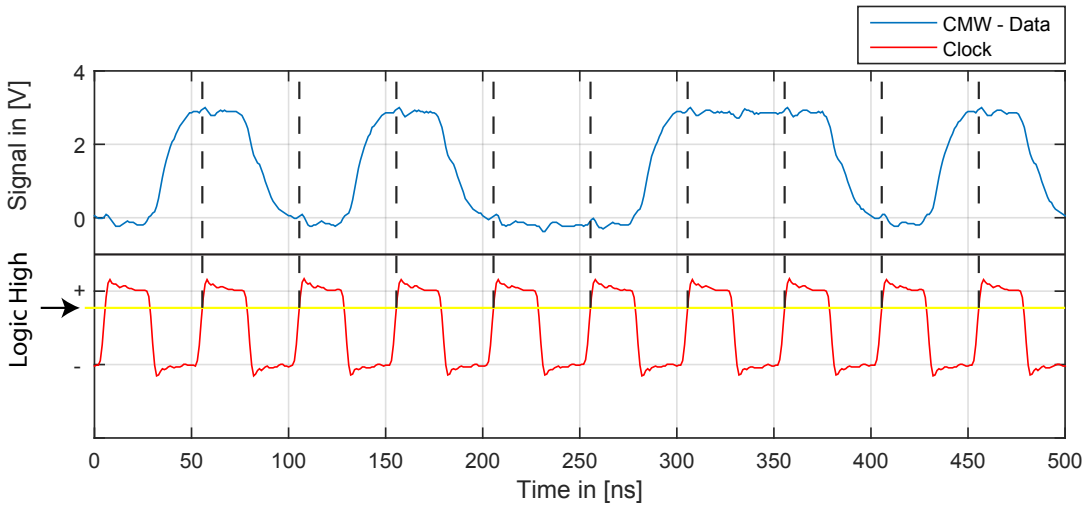


Figure 4.67: The data stream from the ΔΣ modulator is sampled at the rising edge of the returned clock, illustrated with dotted lines.

4.6.2 Sinc3 Filter

In the previous project a Sinc3 filter was determined to meet the requirements for the digital filter (Christensen and Weber, 2014). The Sinc filter is also called a Cascaded Integrator Comb (CIC) filter, from the cascaded integrators and the characteristic comb like characteristic of the filter response. The structure of a Sinc3 filter is shown in figure 4.68.

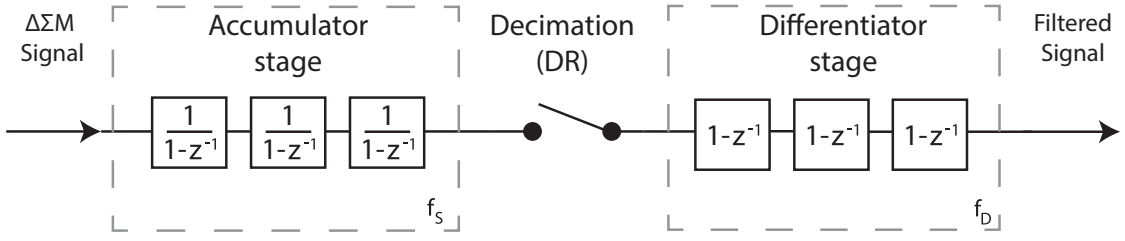


Figure 4.68: The structure of a Sinc3 filter. The input is the signal from the ΔΣ modulator, sampled at the sampling frequency f_S. The accumulator stage consist of three cascaded integrators. The signal is then downsampled or decimated. The ratio between f_S and f_D is the Decimation Ratio (DR). The output of the decimation state is at the data frequency f_D, the same as the differentiator stage.

If the measured voltage is constant, the accumulator stage will reach reach a steady state value. The decimation stage will then sample the accumulated value at f_D. The differentiator or comb stage will also reach a steady state if a constant signal is applied. At DC the resolution of the filtered signal is only limited by the $\Delta\Sigma$ modulator. The filter characteristics are dependent on the decimation ratio as shown in equation (4.20) (Schlichthärle, 2011).

$$H_f(z) = \left(\frac{1}{DR} \cdot \frac{1-z^{DR}}{1-z^{-1}}\right)^3 \tag{4.20}$$

The bode diagram of equation (4.20), with decimation ratio 8, 16, 32, 64, and 256 is shown in figure 4.69.

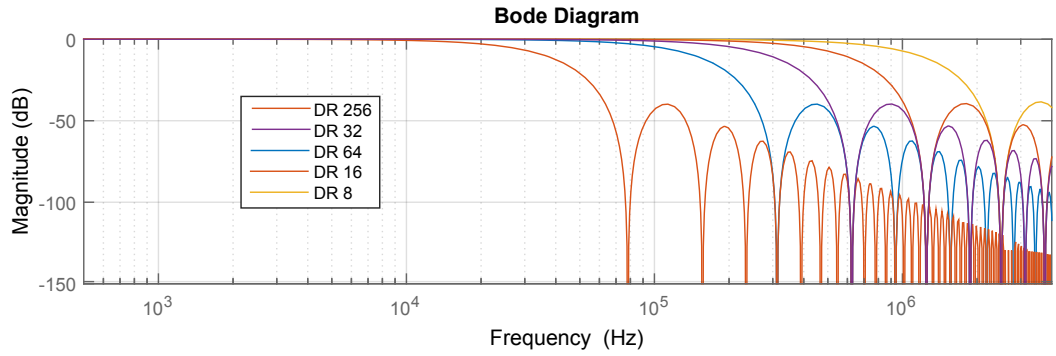


Figure 4.69: Bode diagram of equation (4.20).

When designing the filter for the$\Delta\Sigma$ modulator data, the decimation ratio must be chosen. The signals to be measured are highly oversampled, and by choosing the DR it is possible to make a tradeoff between the resolution of the measurement and the bandwidth. Another factor that weighs in on this particular setup is the sampling time. It would be intuitive to choose a DR of 256, as the cutoff frequency is $20,4[kHz]$, as shown in figure 4.20, which is close to the Nyquist frequency of the phase voltages. However, the throughput frequency, which is the frequency of the data out of the filter, is only $78.125[kHz]$. Comparing to the DR of 8, which has a cutoff frequency of $653[kHz]$ and a throughput of $2.5[MHz]$, the DR of 256 would require a much longer system excitation and sampling time, in order to gather enough points to perform the PSD.

A shorter system excitation and sampling time, may enable the parameter identification to be executed online, in between operation. The effect of changing the DR is inspected on the voltage measurements, as they have a higher bandwidth than the currents. Voltage measurements at different DRs are shown in figure 4.70.

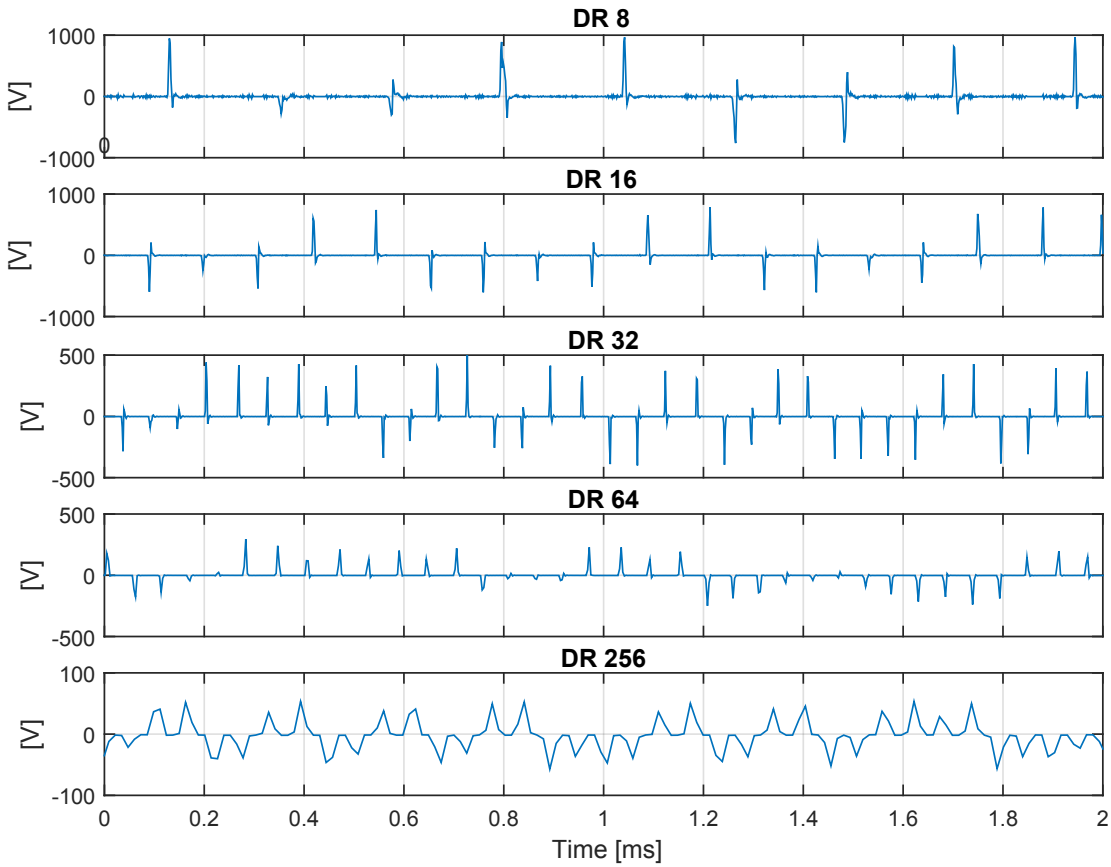


Figure 4.70: Comparison of a pseudo random voltage signal measured with different decimation ratios. (It should be noted that the shown graphs are of five different measurements but with the same duty cycle).

The first thing to notice is that the amplitudes are very different for the different DRs. On the higher DRs, especially 64 and 256, the voltage peaks are filtered out and the amplitude is well below the actual voltage, which we know to be $560[V]$. However, because the voltages are PWM based, as long as the sampling frequency is above the Nyquist frequency of the inverter's switching frequency, no information is lost. This is shown in figure 4.71 and figure 4.72, where the PSD is taken of the voltage measurements at the different DRs.

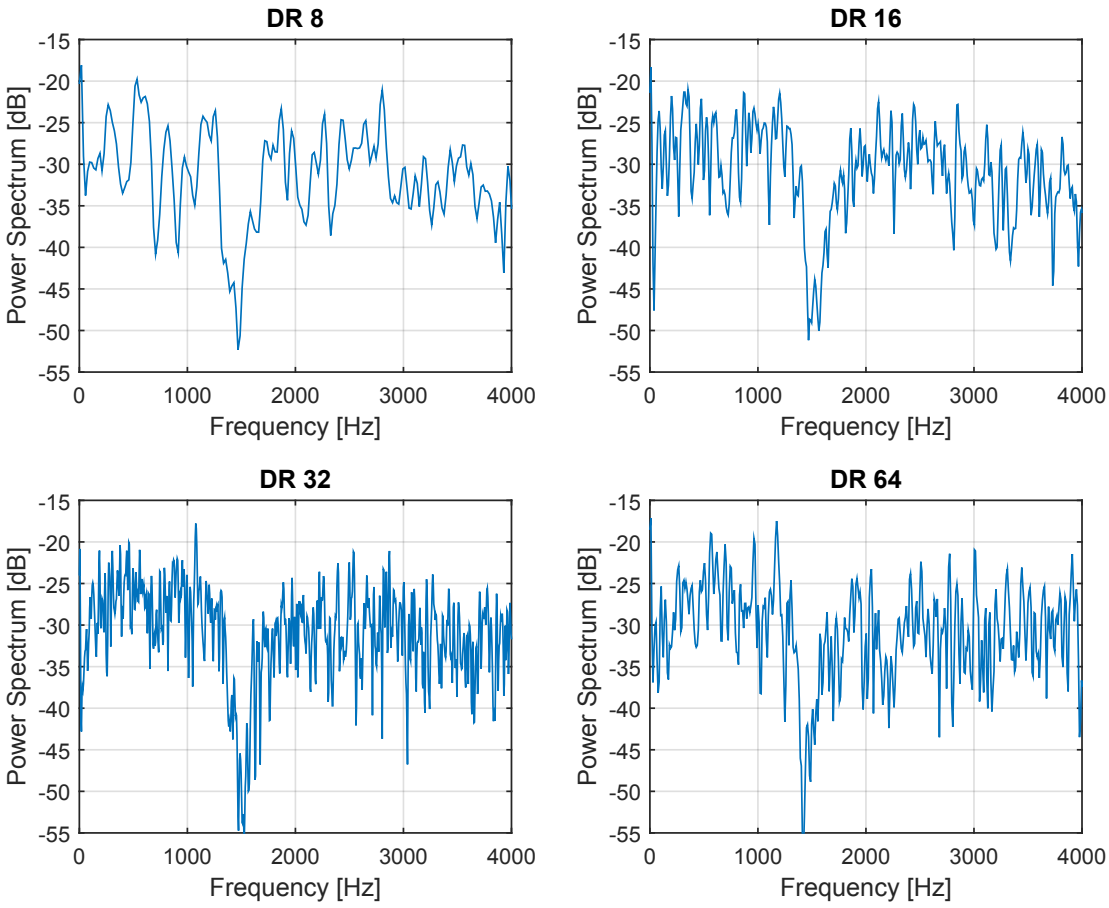


Figure 4.71: Impact of the decimation ratio of the voltage measurement, on the PSD.

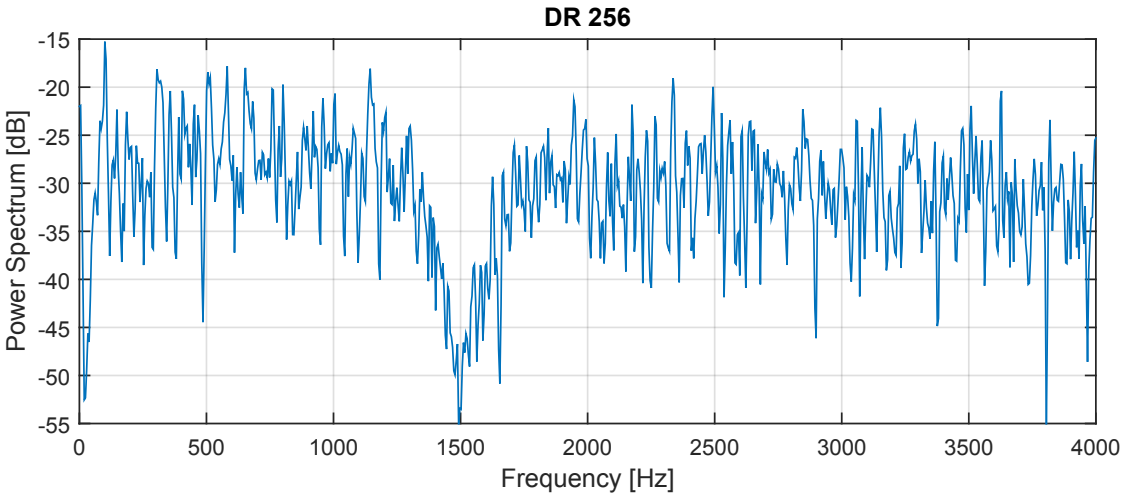


Figure 4.72: .

It is seen that although the magnitude of the measurements are very different in the time domain, the PSD shows the same tendencies and has gains of the same order. For the parameter identification tests for this project, sampling time is not an issue and the following tests will be executed with a DR of 256. However, for future work, one might experiment with the DRs on the stand alone solution, or even look into different digital filters.

Chapter 5

Parameter Identification of Sine Wave Filter

The results of using the method of parameter identification, described in Chapter 3 "*Parameter Identification Method*", to identify the system with only the filter connected to the inverter, is shown in the following. From applying the method to the filter only experience is gained into deciding on the conditions of the Particle Swarm Optimization (PSO), where values such as boundaries, cost value and algorithm constants are considered. The initial values and parameters used in the identification strategy are those of section 3.4.3 "*Particle Swarm Optimization*". The strategy is tested with the filter only, in order to simplify the system and better evaluate the accuracy of the results. The filter admittance is derived analytically, the identification strategy is utilized with simulated data, and lastly the strategy is utilized with data from the laboratory setup.

5.1 Admittance of Filter

Figure 5.1 illustrates how the admittance of the filter is derived. The system is derived from the assumption that the filter terminal U is excited with a signal different than V and W, which are excited with equal signals, as described in section 3.3 "*Pseudo Random Binary Signal*". In the case when nothing is connected to the filter output terminals U', V', and W', an equivalent system of admittances, figure 5.1.c, can be used for derivation of the filter admittance. In the case of figure 5.1 it is assumed that the parameters of the filter are equal for the three phases.

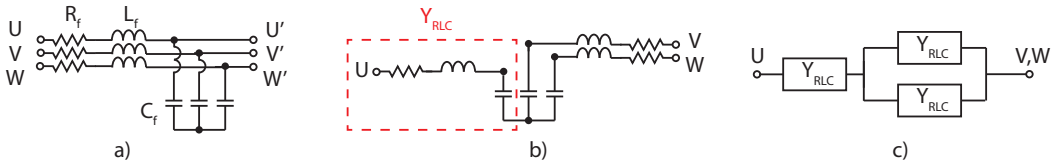


Figure 5.1: a) Diagram of the LC-filter, b) Diagram of the LC-filter reordered to clearify Y_{RLC}, c) Admittance connection of filter components for derivation.

The admittance of each phase in the filter, Y_{RLC}, is derived in equation (5.1) and illustrated in figure 5.1.b. Knowing that the admittance is the inverse of the impedance it is easy to derive Y_{RLC}, as shown in equation (5.1).

$$\begin{aligned} Z_{RL} &= R_f + j\omega L_f \ , \quad Z_C = \frac{1}{j\omega C_f} \\ Z_{RLC} &= Z_{RL} + Z_C = R_f + j\omega L_f + \frac{1}{j\omega C_f} \\ Y_{RLC} &= \frac{1}{Z_{RLC}} = \frac{1}{R_f + j\omega L_f + \frac{1}{j\omega C_f}} \\ &= \frac{j\omega C_f}{j\omega^2 L_f C_f + j\omega R_f C_f + 1} \end{aligned} \tag{5.1}$$

The admittance for the filter, Y_{filter}, is derived in equation (5.2), simply based on the representation of figure 5.1.c.

$$\begin{aligned} Y_{filter} &= \frac{1}{\frac{1}{Y_{RLC}} + \frac{1}{2Y_{RLC}}} = \frac{2}{3} Y_{RLC} \\ &= \frac{j\omega 2C_f}{3(j\omega^2 L_f C_f + j\omega R_f C_f + 1)} \\ &\approx \frac{\omega_0^2}{j\omega^2 + j\omega 2\zeta\omega_0 + \omega_0^2} \end{aligned} \tag{5.2}$$

Equation (5.2) shows a second order system, from which it is possible to determine the natural frequency and the damping coefficient of the system, as shown in equation (5.3) and equation (5.4), respectively, using the parameters of table 5.1.

$$f_0 = \frac{\omega_0}{2\pi} = \frac{1}{2\pi}\sqrt{\frac{1}{L_f C_f}} = 1251.6[Hz] \tag{5.3}$$

$$\zeta = \frac{R_f C_f}{2\omega_0 L_f C_f} = \frac{R_f}{2L_f\sqrt{\frac{1}{L_f C_f}}} = 0.0053 \tag{5.4}$$

It is clear from equation (5.3) and equation (5.4), that only L_f and C_f has an influence on the natural frequency of the system, while R_f is directly proportional to the damping coefficient.

5.2 Simulated Results of System with Filter Only

In the following the parameter identification method is applied on a simulated system of the filter and inverter. The results are used to prove and improve the method of parameter identification.

A plot of the analytical expression of the admittance of the filter is shown in figure 5.4, where the parameters of table 5.1 are used. The resistance R_f was determined from measurements in the laboratory, described in Chapter 2 "*System Models*".

R_f	0.1	[Ω]
L_f	1.1	[mH]
C_f	14.7	[μF]

Table 5.1: Filter parameters from datasheet (Danfoss, 2010). R_f is measured.

System response estimation:
The inverter and filter system is simulated in Plecs in Simulink, described in Chapter 2 "*System Models*", where the simulated data is used to estimate the system response, as described in section 3.2.3 "*Frequency Response Estimation*". The Plecs model uses the parameters of table 5.1 and the simulated system is excited with the PRBS based PWM signal with 55[%] duty cycle at a switching frequency of 9[kHz] and a variable band of 2[kHz]. The input and output data used to estimate the system is shown in figure 5.2 and a zoom of the data in figure 5.3.

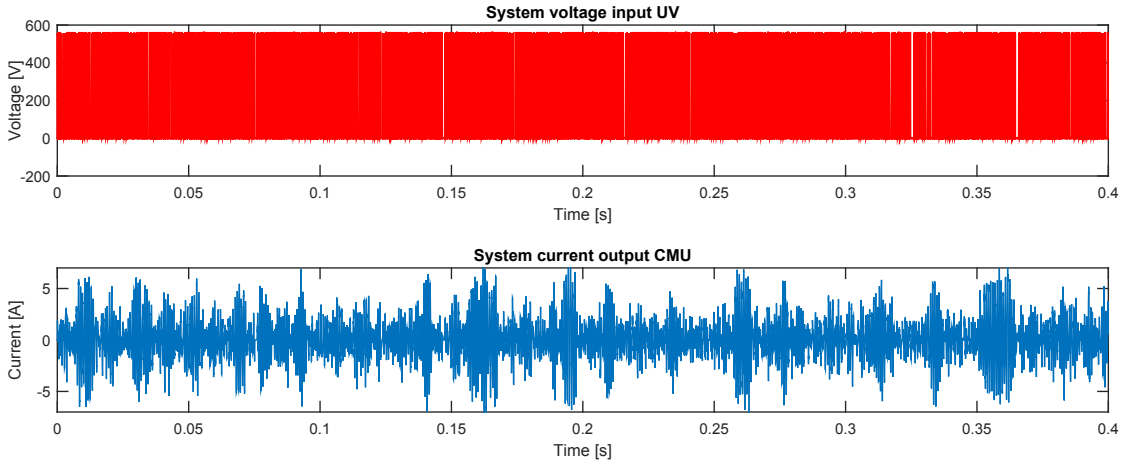


Figure 5.2: Input voltage and output current of Plecs model, with a PRBS signal with 55[%] duty cycle and a switching frequency of 9[kHz] with a variable frequency band of 2[kHz].

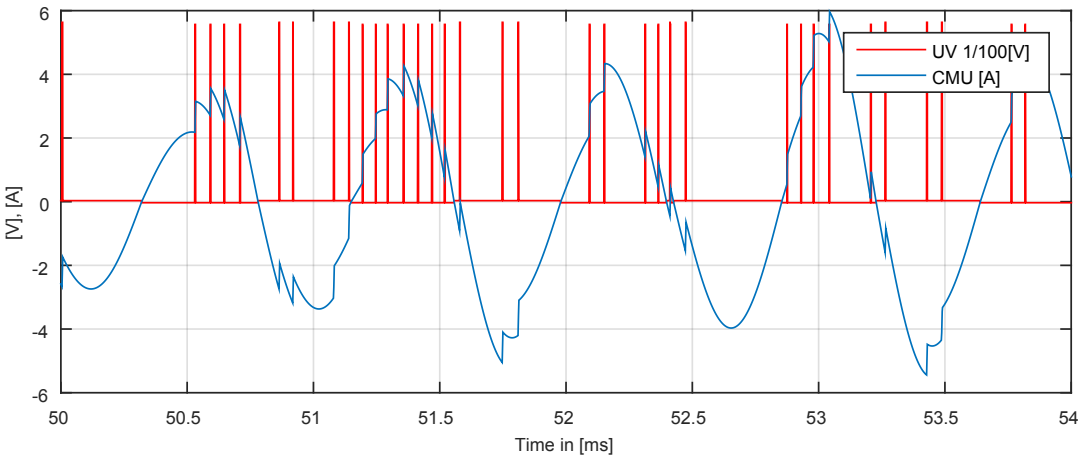


Figure 5.3: Zoomed view of figure 5.2.

The simulated system is estimated with Welch's method and the estimated response is shown in figure 5.4, together with the analytical solution. The estimated response differs from the the analytical at the low frequencies, which is shown in the plot with logarithmic scale, where the analytic solution has a constant inclination, 20[dB] pr. decade, while the estimated solution breaks at around 20 [Hz]. The difference at the low frequencies means that there will always be a difference between the analytical and the estimated response, which will influence the cost function of the PSO. The difference in the low frequencies is discussed in the following while considering the final cost value and the influence to the identified parameters.

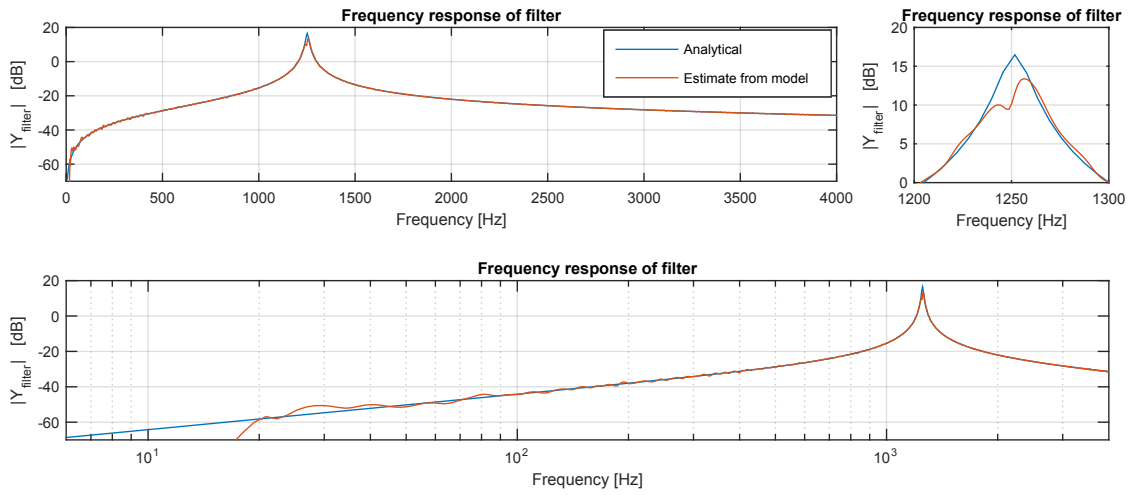


Figure 5.4: Comparison between the estimated system response based on the signals from the simulated model and the analytical expression with the parameters of table 5.1

Particle swarm optimization:
The estimated response is used as the reference in the PSO and the analytical expression is fitted to the estimated response, in order to identify the system parameters, as described in section 3.4.3 "*Particle Swarm Optimization*". Table 5.2 shows the parameters used, to run the PSO algorithm and identify the parameters of the system simulated in Plecs, along with the results. The results of the PSO is plotted in figure 5.5 along with the initital guesses and the best solution of each iteration.

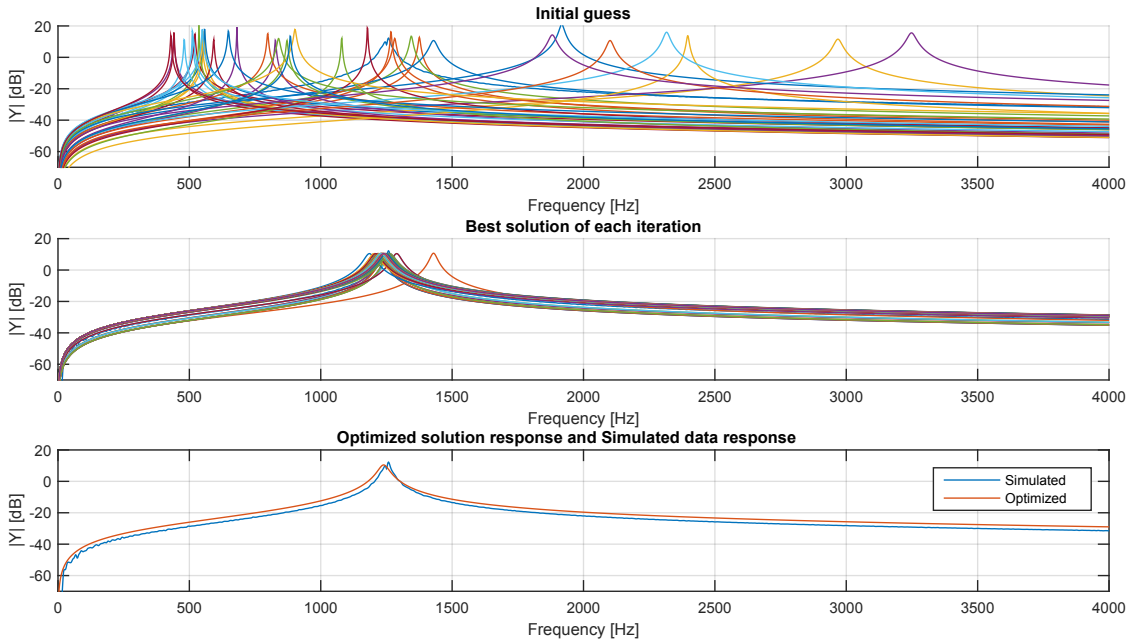


Figure 5.5: PSO based on data from simulated system. (top) Shows the responses of all the initial particle solutions, (middle) shows the best solution of each iteration, (bottom) shows the final solution of the PSO and the analytical response of the expected parameters. Results: $R_f = 0.2[\Omega]$, $L_f = 0.8246[mH]$, $C_f = 20[\mu F]$

Particle Swarm Optimization of Simulated System:

Parameter boundaries:			**Constants:**		**Results:**			**Dif:**
R_f	0.05 - 0.2	$[\Omega]$	c_1	2	R_f	0.2	$[\Omega]$	100 %
L_f	0.1 - 10	$[mH]$	c_2	2	L_f	0.8246	$[mH]$	-25.04 %
C_f	1.0 - 20	$[\mu F]$	N_p	35	C_f	20	$[\mu F]$	36.05%

Table 5.2: Particle Swarm optimization with final cost of 125.3 in 100 iterations.

The PSO does not identify the parameters of the system. The iteration limit is reached at 100 iterations and a cost value of 125.3. The parameter values are too far from the expected values of table 5.1 and modifications are needed to successfully identify the systems parameters.

Since it was observed that the analytical and the estimated system have different characteristics at low frequencies, these low frequencies are cut out before running the PSO again, in order to investigate their influence on the results. It is assessed to be reasonable to cut out these frequencies, since this range is of no importance for the system response and because the difference is most likely due to a lack of sampling points, why the low frequency response can't be determined accurately. The frequencies from 0-100$[Hz]$ are cut out and the results are shown in figure 5.6 and table 5.3 from which it is concluded that it improves the result of the optimization algorithm. The cost value is improved drastically from 125.3 to 2.25, while the total iterations are reduced from the limit of 100 to only 13 iterations. The identified parameters are also improved greatly towards the parameters used to simulate the system.

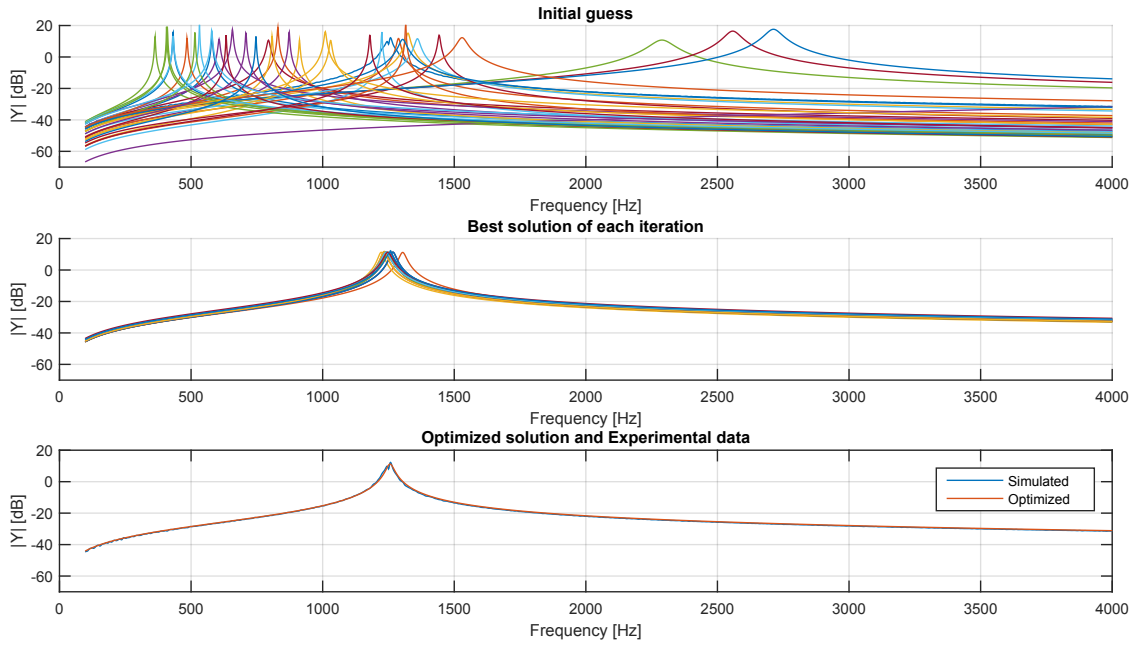


Figure 5.6: PSO based on data from simulated system. (top) Shows the responses of all the initial particle solutions, (middle) shows the best solution of each iteration, (bottom) shows the final solution of the PSO and the analytical response of the expected parameters. Results: $R_f = 0.183$ [Ω], $L_f = 1.064$ [mH], $C_f = 15.087$ [μ F]

Particle Swarm Optimization of Simulated System:

Parameter boundaries:			**Constants:**		**Results:**			**Dif:**
R_f	0.05 - 0.2	[Ω]	c_1	2	R_f	0.183	[Ω]	83.17%
L_f	0.1 - 10	$[mH]$	c_2	2	L_f	1.064	$[mH]$	-3.28 %
C_f	1.0 - 20	$[\mu F]$	N_p	35	C_f	15.087	$[\mu F]$	2.64 %

Table 5.3: Particle Swarm optimization with final cost of 2.25 in 13 iterations.

The method of parameter identification has been proved for the simulated system, why the method is applied to the laboratory setup in the following.

5.3 Laboratory Results of System with Filter Only

The laboratory setup is excited with a PRBS based PWM signal with 55[%] duty cycle at a switching frequency of 9[kHz] and a variable band of 2[kHz]. The input voltage and output current is sampled with the IB2 board, which is described in section 4.4 "*Custom Parameter Identification Hardware*", and stored on the sbRIO for processing on a desktop pc. The system voltage input UV is measured at the inverter output terminal U relative to the voltage at the inverter output terminal V, while the system current output CMU is measured at the inverter output terminal U. The voltage and current is plotted in figure 5.7 and figure 5.8.

System response estimation:

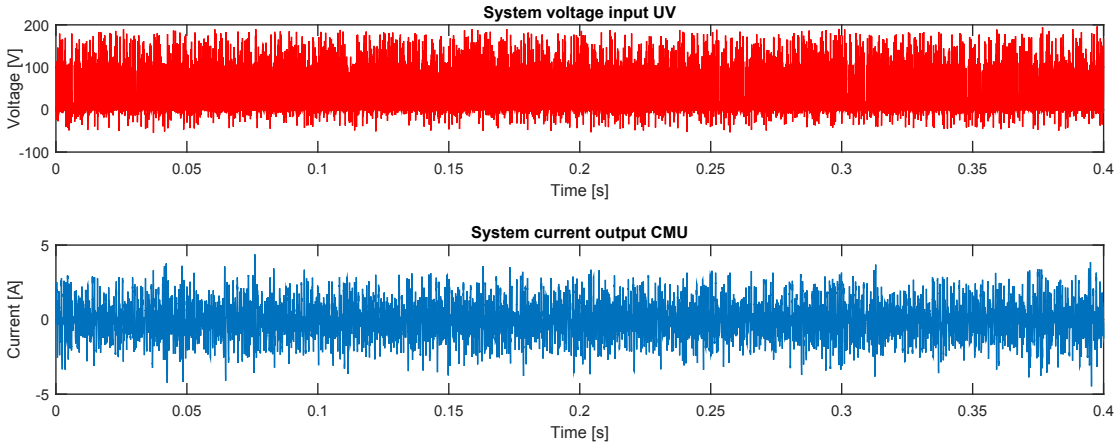


Figure 5.7: Input voltage and output current of laboratory setup, measured with IB2 - PRBS signal with 55[%] duty cycle and a switching frequency of 9[kHz] with a variable frequency band of 2[kHz].

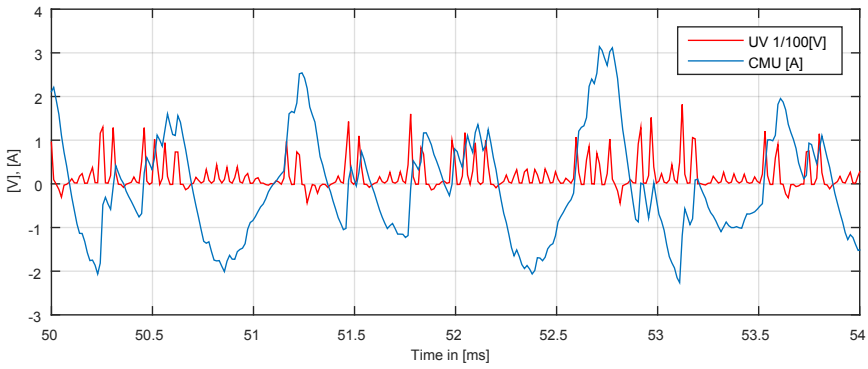


Figure 5.8: Zoom view of figure 5.7.

Figure 5.9 shows the analytical solution along with the estimated response based on experimental data, from the laboratory setup. The estimated response of the labo-

ratory setup is more damped than the analytical response, why R_f is expected to be larger than the measured value $R_f = 0.1[\Omega]$. The resonance frequency of the estimated response is different than that of the analytical solution, which indicates that either C_f, L_f or both are different than the expected values listed in table 5.1. The difference in the two responses indicates that the parameters identified in the following will be different from the expected.

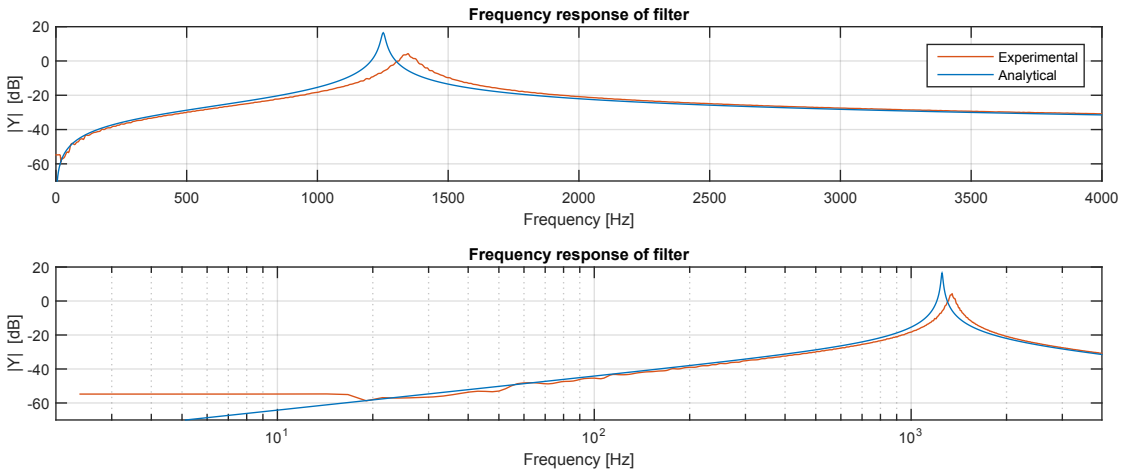


Figure 5.9: Comparison between the estimated system response based on the signals from the laboratory setup and the analytical expression with the parameters of table 5.1.

Particle swarm optimization:
The data for the frequencies 0-100$[Hz]$ are removed to improve the optimization algorithm towards a lower cost value. The results of the PSO are shown in figure 5.10 and table 5.4.

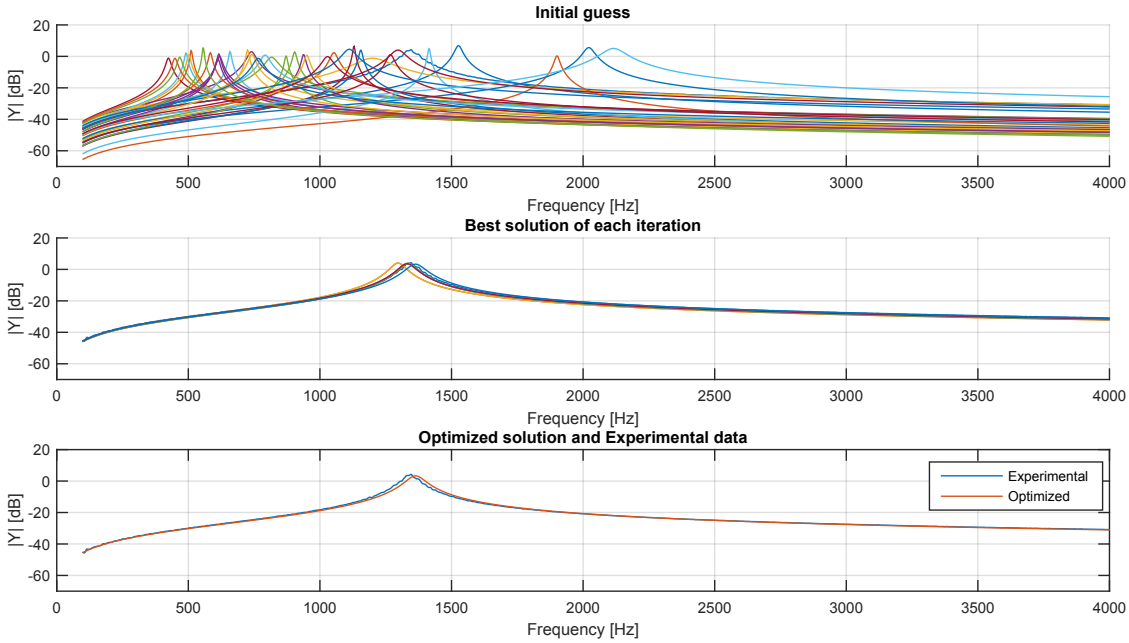


Figure 5.10: PSO based on laboratory data. (top) Shows the responses of all the initial particle solutions, (middle) shows the best solution of each iteration, (bottom) shows the final solution of the PSO and the analytical response of the expected parameters. Results: $R_f = 0.45\Omega$, $L_f = 1.072mH$, $C_f = 13.487\mu F$.

Particle Swarm Optimization of Laboratory Data:

Parameter boundaries:			**Constants:**		**Results:**			**Dif:**
R_f	0.1 - 0.8	$[\Omega]$	c_1	2	R_f	0.45	$[\Omega]$	354.34 %
L_f	0.1 - 10	$[mH]$	c_2	2	L_f	1.072	$[mH]$	-2.59 %
C_f	1.0 - 20	$[\mu F]$	N_p	35	C_f	13.487	$[\mu F]$	-13.48 %

Table 5.4: Particle Swarm optimization with final cost of 4.60 in 6 iterations.

The big difference in the R_f is expected since it was already observed that the estimated system is more damped than the expected analytical solution. The difference in C_f is also expected since the resonance frequency of the estimated response is higher than the expected analytical response. The cost value of 4.60 is higher than the value 2.25 from the simulated case, why the cognitive parameter, c_1, and the social parameter, c_2, are chosen at a smaller value in order to reduce the step size of each iteration. The results with $c_1 = c_2 = 1$ are shown in figure 5.11 and table 5.5.

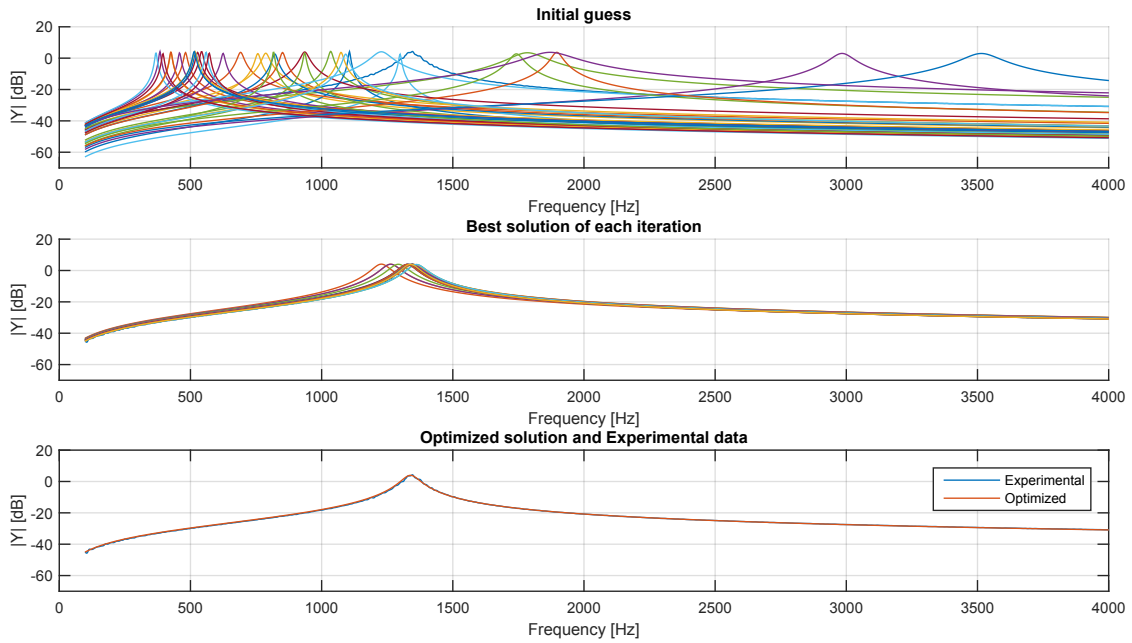


Figure 5.11: PSO based on laboratory data and $c_1 = c_2 = 1$. (top) Shows the responses of all the initial particle solutions, (middle) shows the best solution of each iteration, (bottom) shows the final solution of the PSO and the analytical response of the expected parameters. Results: $R_f = 0.42\Omega$, $L_f = 1.047mH$, $C_f = 13.489\mu F$.

Particle Swarm Optimization of Laboratory Data:

Parameter boundaries:			**Constants:**		**Results:**			**Dif:**
R_f	0.4 - 0.5	$[\Omega]$	c_1	1	R_f	0.42	$[\Omega]$	323.77 %
L_f	0.1 - 10	$[mH]$	c_2	1	L_f	1.047	$[mH]$	-4.74 %
C_f	1.0 - 20	$[\mu F]$	N_p	35	C_f	13.489	$[\mu F]$	-8.24 %

Table 5.5: Particle Swarm optimization with final cost of 0.57 in 15 iterations.

It is concluded that it is possible to sufficiently determine the parameters of the system with the filter alone, based on the described method. L_f and C_f are identified within 10% of the expected value of the parameters, and this is assessed a reasonable benchmark for the filter capacitance and inductance.

It is important to remember that the parameters and constants of the PSO are problem specific why a general understanding of the problem is important. The experience of this chapter is of great importance in order to satisfactorily design the PSO to the laboratory setup. The experience gained is listed below.

- Remove the data from the lower frequencies of the system response in order to improve the convergence of the PSO. The data at the low frequencies 0-100[Hz]

are not important when describing the response of the filter and can be removed without changing the system response.

- Evaluate the location of the resonance frequency and the damping of the estimated response in order to sufficiently determine the parameter boundaries in the PSO algorithm.
- The size of the constants c_1 and c_2 has an influence on the step size and in order to get a cost value < 1 these constants were reduced from 2 to 1.

In the next section the method of parameter identification is applied to the system with filter and motor connected, using the gained experience from the simplified system with the filter only.

Chapter 6

Parameter Identification of PMSM with Sine Wave Filter

Similar to the previous chapter, the parameter identification method described in Chapter 3 "*Parameter Identification Method*" is utilized in the following, with both the filter and the motor connected. First a simulation of the system is used, followed by identification of the laboratory setup.

The system's admittance, which is used for curve fitting in the PSO algorithm, was derived in section 3.4 "*Model Parameter Optimization*" and is shown in equation (6.1).

$$|Y_{(j\omega)}| = \left| \frac{2}{3} \cdot \frac{(j\omega)^2 a_2 + j\omega a_1 + 1}{(j\omega)^3 b_3 + (j\omega)^2 b_2 + j\omega b_1 + b_0} \right| \tag{6.1}$$

$$\begin{aligned} a_1 &= R_m C_f & a_2 &= L_m C_f \\ b_0 &= R_m + R_f & b_1 &= L_m + L_f + R_m R_f C_f \\ b_2 &= R_m L_f C_f + R_f L_m C_f & b_3 &= L_m L_f C_f \end{aligned}$$

The expected system parameters are listed in table 6.1 and these are also the parameters used in the simulated model in the following section.

System parameters:		
R_m	0.18	[Ω]
L_m	3.29	$[mH]$
R_f	0.1	[Ω]
L_f	1.1	$[mH]$
C_f	14.7	$[\mu F]$

Table 6.1: Expected system parameters used to simulate the system and to define PSO boundaries.

6.1 Simulated Results of System with Filter and Motor

The system model is excited as described in section 3.1 "*Parameter Identification Strategy*", with a PRBS based PWM, with 55% duty cycle at a frequency of 9[kHz] and a variable frequency band of 2[kHz]. The simulated voltage and current signals are shown in figure 6.1 and a zoom of the signals in figure 6.2.

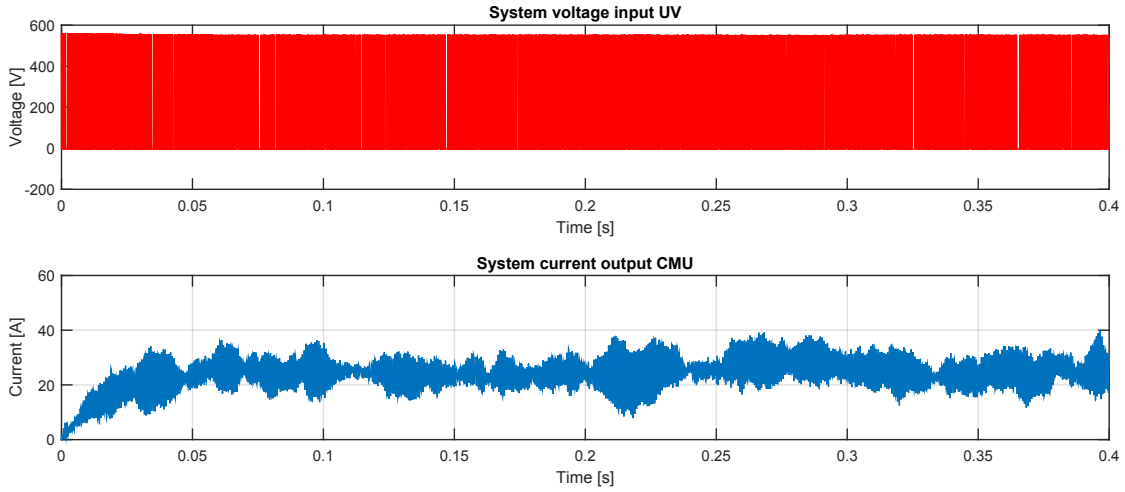


Figure 6.1: The simulated voltage and current signals.

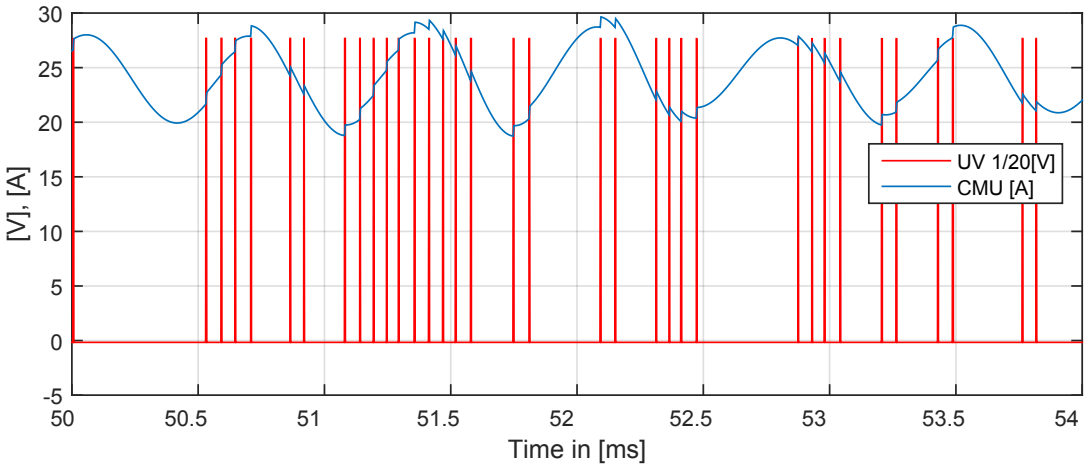


Figure 6.2: Zoomed view of figure 6.1.

The signals are used to estimate the system's frequency response, using the Welch method to estimate the PSD and CPSD of the signals. The estimated response is plotted in figure 6.3 for three different settings. The top plot shows that the estimated response is noisy around the anti-resonance, which leads to utilizing different settings of the Welch estimator. The three estimates are estimated with 4 or 8 windows in the Welch estimator and with 50% and 75% overlap of the windows.

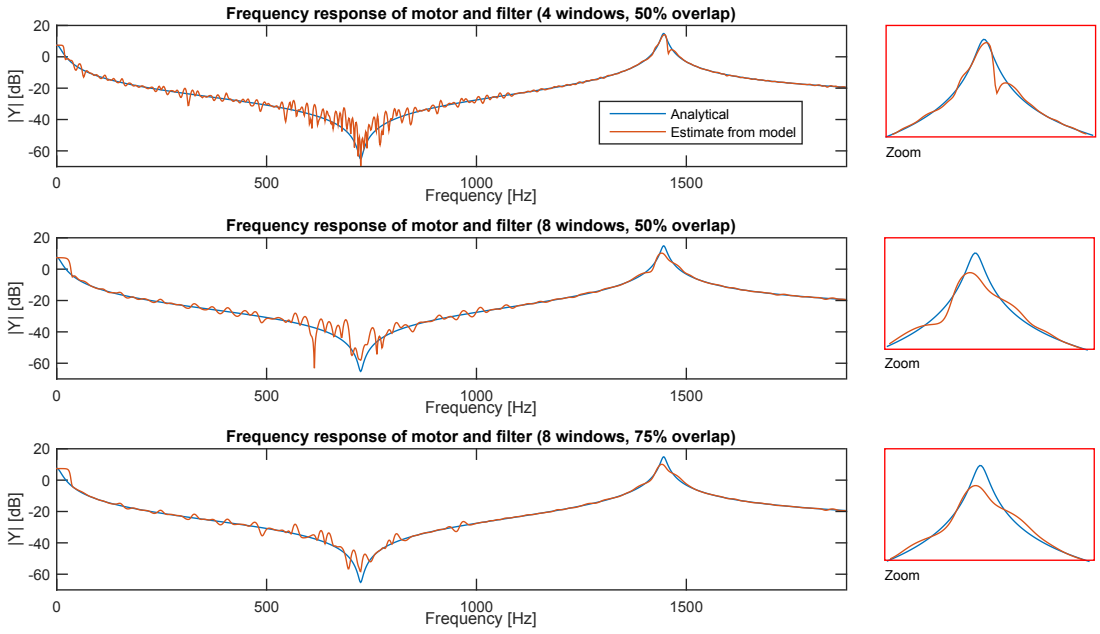


Figure 6.3: System estimates from the simulated voltage and current signals, utilizing different conditions for the Welch estimator. On the right is a zoom of the resonance peaks.

The estimated response is more smoothed out when more windows are used, as seen in figure 6.3. However, the estimate suffers at the resonance peak when the number of windows is increased, from which it is assessed that 8 windows is the highest possible for the given signals. The middle response plot of figure 6.3 is still noisy at the anti-resonance, why the window overlap is increased, and an overlap of 75% seems to reduce the noise, without affecting the peak at the resonance.

From figure 6.3 it is seen that the estimated response based on the simulated signals is close to the expected response of the system.

The Welch estimate with 8 windows and an overlap of 75% is used in the curve fitting algorithm. The conditions and results of the PSO is shown in table 6.2 while the results are illustrated in figure 6.4. The frequency within which the PSO is utilized is, chosen to be 100-2500[Hz], based on experience gained in the previous chapter, and because the relevant dynamics of the system is located around the resonance peaks.

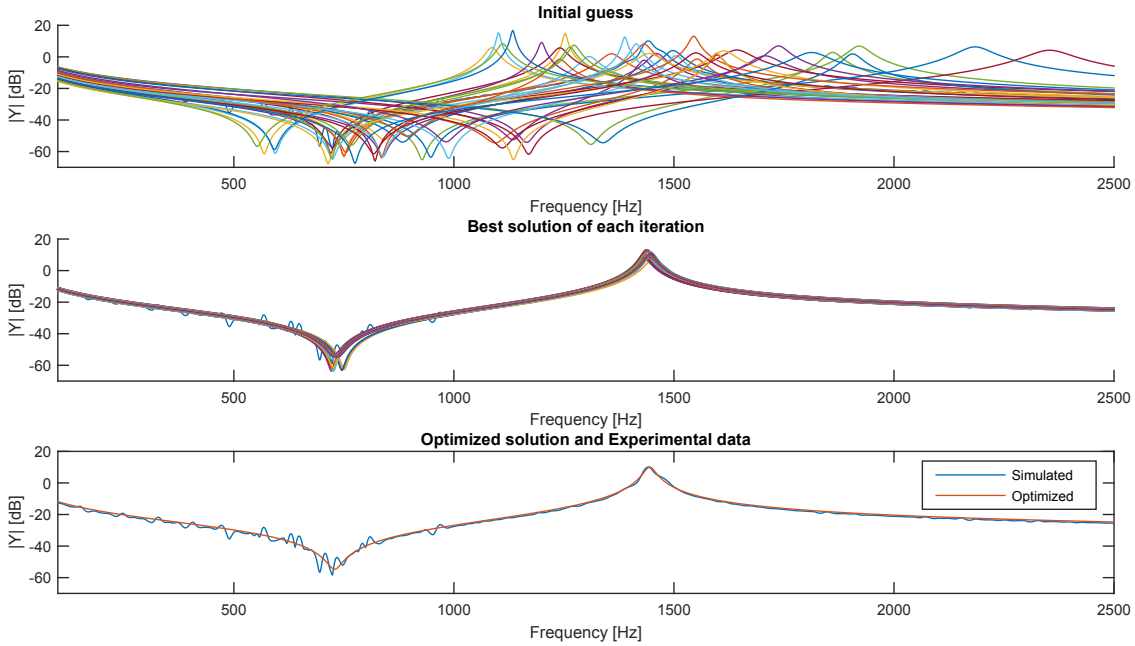


Figure 6.4: Results of the PSO algorithm, with a Welch estimate of simulated signals using 8 windows and 75% overlap. The PSO reached the limit of 100 iterations with a final cost of 20.737.

Particle Swarm Optimization of Simulated System:

Parameter boundaries:			**Constants:**		**Results:**			**Dif:**
R_m	0.10 - 0.5	$[\Omega]$	c_1	0.5	R_m	0.5000	$[\Omega]$	177.78 %
L_m	0.10 - 5.0	$[mH]$	c_2	0.5	L_m	2.9438	$[mH]$	-10.52%
R_f	0.05 - 0.5	$[\Omega]$	N_p	35	R_f	0.1522	$[\Omega]$	52.19%
L_f	0.50 - 2.0	$[mH]$			L_f	1.006	$[mH]$	-8.54%
C_f	10.0 - 20	$[\mu F]$			C_f	16.184	$[\mu F]$	10.09%

Table 6.2: Particle Swarm optimization with final cost of 20.737 in 100 iterations.

The results of table 6.2 show that the identified parameters, based on the 8 windows and 75% overlap estimate, differs from the expected parameters with more than 10%, why the results are not assessed sufficiently accurate. However, in general it is clear that the curve fitting is successful in estimating a fitting solution, why it is necessary to come up with a more smooth PSD estimate, without loosing the dynamics at the resonance peaks.

The problem with increasing the number of windows in the Welch estimator, is that the signals are filtered too much, why the estimate will not sufficiently describe the system dynamics. It was observed from figure 6.3 that increasing the window overlap, resulted in a more smooth estimate and based on this it was found that an estimate with 4

windows and 95% overlap is able to give a smooth response, including the dynamics at the resonance peaks, as is shown in figure 6.5.

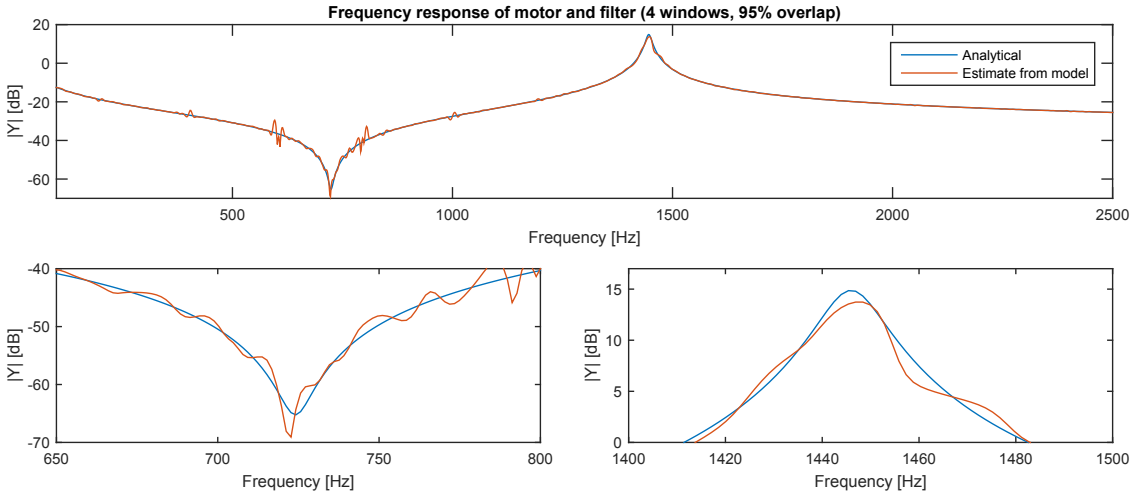


Figure 6.5: System estimate from the simulated voltage and current signals with a Welch estimate of 4 windows and 95% overlap, along with a zoom at the resonance peaks.

The results of parameter identification, based on the Welch estimate with 4 windows and 95% overlap, is shown in figure 6.6 and conditions and results are listed in table 6.2.

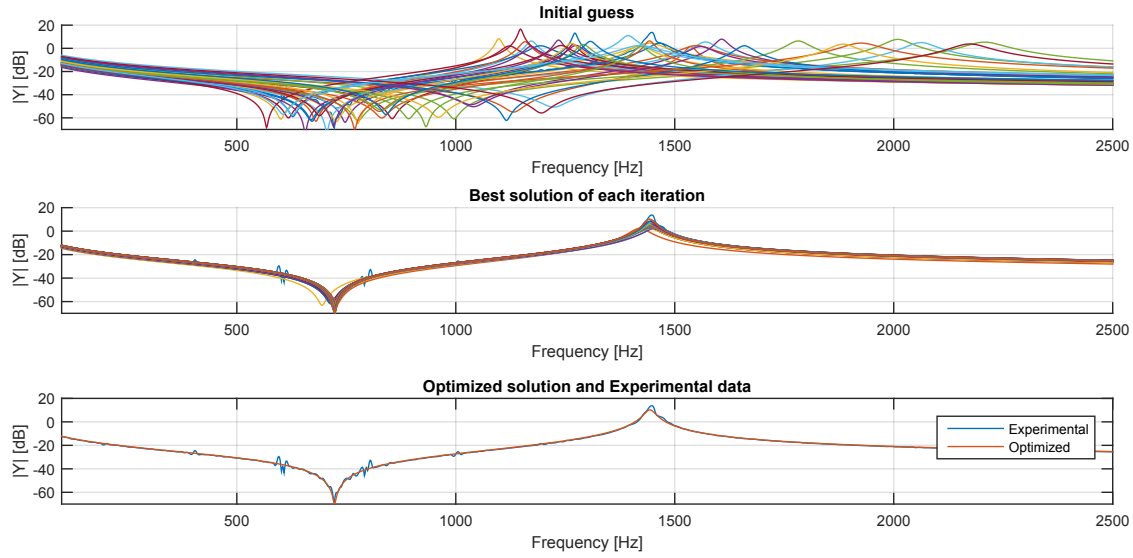


Figure 6.6: Results of the PSO algorithm, with a Welch estimate of simulated signals using 4 windows and 95% overlap. The PSO reached 100 iterations with a final cost of 9.5957.

Particle Swarm Optimization of Simulated System:

Parameter boundaries:			**Constants:**		**Results:**			**Dif:**
R_m	0.10 - 0.5	[Ω]	c_1	0.5	R_m	0.1000	[Ω]	-44.44 %
L_m	0.10 - 5.0	$[mH]$	c_2	0.5	L_m	3.1964	$[mH]$	-2.84%
R_f	0.05 - 0.5	[Ω]	N_p	35	R_f	0.1935	[Ω]	93.51%
L_f	0.50 - 2.0	$[mH]$			L_f	1.0723	$[mH]$	-2.52%
C_f	10.0 - 20	$[\mu F]$			C_f	15.146	$[\mu F]$	3.03%

Table 6.3: Particle Swarm optimization with final cost of 9.5957 in 100 iterations.

The results of table 6.3 show that the identified parameters, based on the 4 windows and 95% overlap estimate, was improved greatly towards the expected parameters, except for the two resistances R_m and R_f. Since it seems difficult to identify the resistances accurately, it has been chosen to fix these parameters to the expected values when running the PSO. The results of the PSO with fixed values for the resistances are listed in table 6.4, which show great improvements toward identifying the remaining three parameters.

Particle Swarm Optimization of Simulated System:

Parameter boundaries:			**Constants:**		**Results:**			**Dif:**
R_m	0.18	[Ω]	c_1	0.5	R_m	0.18	[Ω]	0 %
L_m	0.10 - 5.0	$[mH]$	c_2	0.5	L_m	3.235	$[mH]$	-1.67%
R_f	0.1	[Ω]	N_p	35	R_f	0.1	[Ω]	0%
L_f	0.50 - 2.0	$[mH]$			L_f	1.0811	$[mH]$	-1.71%
C_f	10.0 - 20	$[\mu F]$			C_f	14.9605	$[\mu F]$	1.77%

Table 6.4: Particle Swarm optimization with final cost of 7.1436 in 100 iterations.

It have been shown to be possible to successfully identify the systems parameters based on simulated signals. In the following the method of parameter identification is utilized with signals from the laboratory setup.

6.2 Laboratory Results of System with Filter and Motor

The laboratory setup is excited as described in section 3.1 "*Parameter Identification Strategy*" with a PRBS based PWM with 55% duty cycle at a frequency of 9[kHz] and a variable frequency band of 2[kHz]. The voltage and current signals are shown in figure 6.7 and a zoom of the signals in figure 6.8.

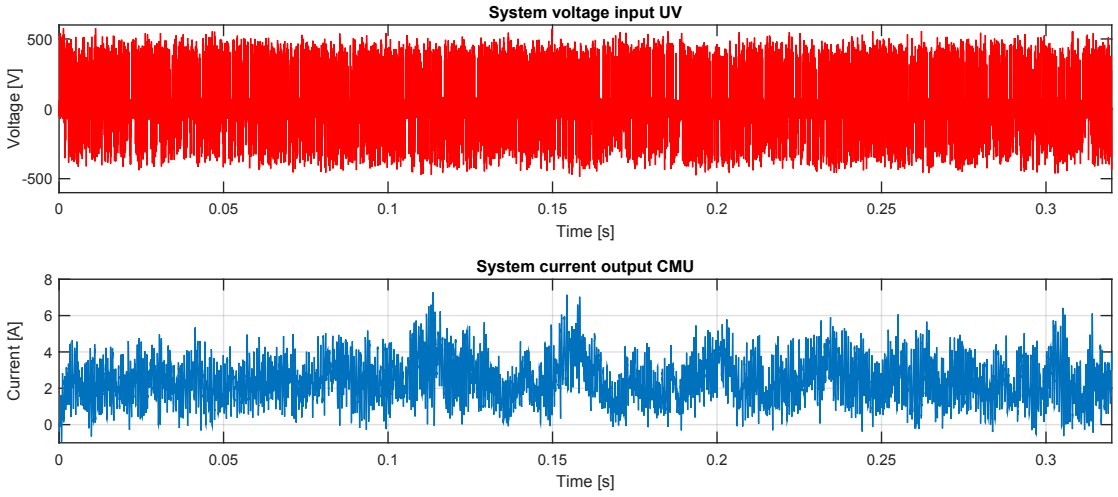


Figure 6.7: Voltage and current signals of the laboratory setup.

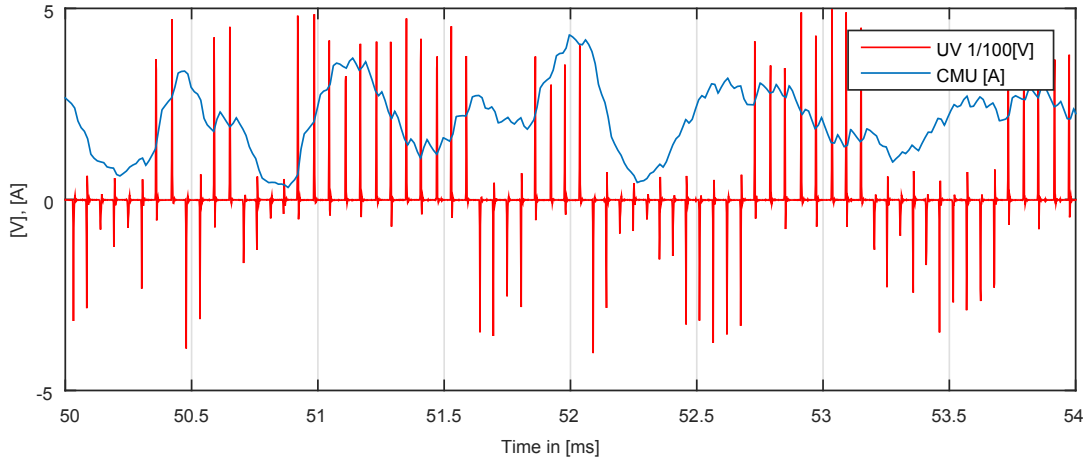


Figure 6.8: Zoom of figure 6.7.

The estimated response of the system is plotted in figure 6.9, which is estimated with 6 hamming windows and 95% overlap. Since the estimated response differs from the expected with both different damping and location of the resonance peaks, another set of data, with a 60% duty cycle PRBS based PWM, was used to estimate a similar response, which shows that it is the actual response of the system.

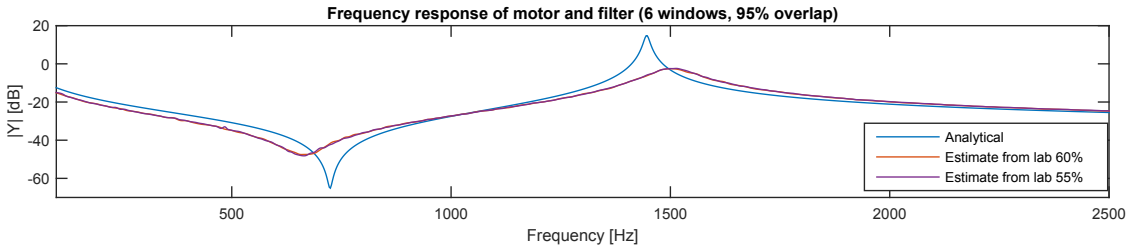


Figure 6.9: Estimated system response based on signals from the laboratory setup, both signals from 55% and 60% duty cycle based PRBS PWM have been used to evaluate the system response.

Since the estimated system response differs significantly from the expected response it is important to consider the boundaries of the parameter, used in the PSO algorithm. If the boundaries are chosen too tight and without the actual parameters with the chosen range, the PSO is not able to converge to a fitting result for the system response. The PSO is utilized with a large range of parameter values and increased number of particles, due to the larger solution range. The results of the PSO is shown in figure 6.10 and conditions and results are listed in table 6.5.

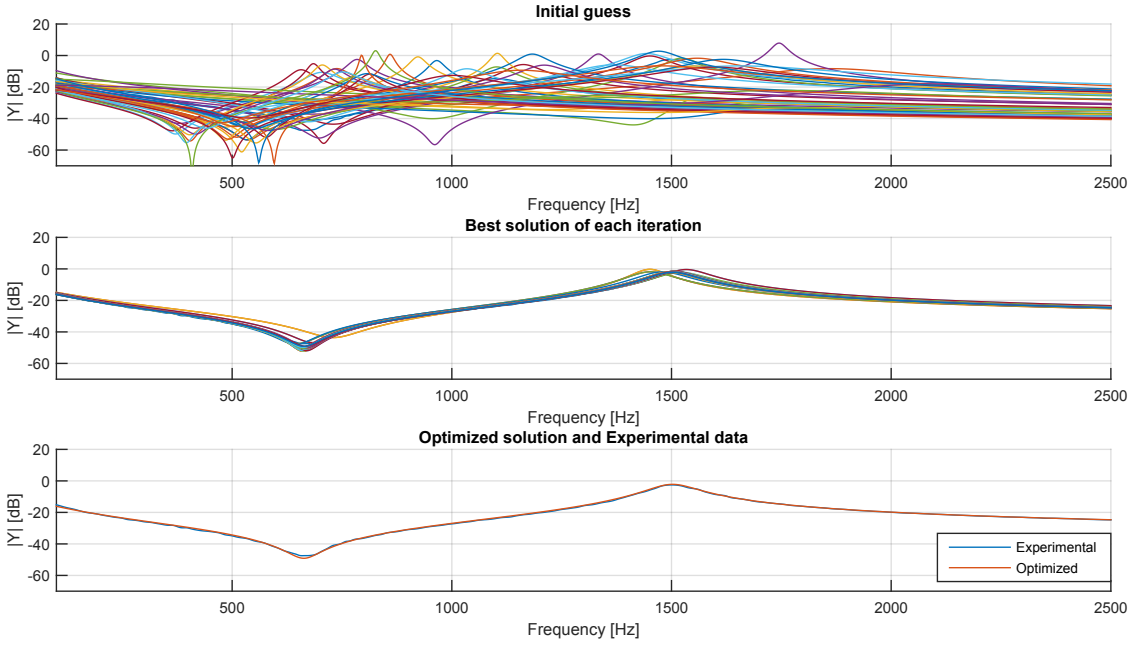


Figure 6.10: Results of the PSO algorithm, with a Welch estimate of simulated signals using 6 windows and 95% overlap. The PSO reached 41 iterations with a final cost of 0.9416.

Particle Swarm Optimization of Simulated System:

Parameter boundaries:			**Constants:**		**Results:**			**Dif:**
R_m	0.10 - 5	[Ω]	c_1	0.5	R_m	1.7242	[Ω]	857.90 %
L_m	0.10 - 10	$[mH]$	c_2	0.5	L_m	4.3366	$[mH]$	31.81%
R_f	0.05 - 2	[Ω]	N_p	50	R_f	0.7555	[Ω]	655.52%
L_f	0.50 - 5	$[mH]$			L_f	1.0537	$[mH]$	-4.21%
C_f	10.0 - 20	$[\mu F]$			C_f	13.2505	$[\mu F]$	-9.86%

Table 6.5: Particle Swarm optimization with final cost of 0.9416 in 41 iterations.

It is obvious, from figure 6.10, that the curve fitting is successful but also the final cost of 0.9416 shows that the response is identified quite accurately. The identified parameters are in general very different from the expected values, however, by relating the filter parameters to the results of section 5.3 "*Laboratory Results of System with Filter Only*" it can be seen that the identified parameters of the filter have similar values, why the results are assessed an accurate identification of the system.

Since the parameter boundaries have been chosen for a large range, it is interesting to investigate whether the PSO is consistent when identifying the systems parameters. The PSO is utilized 20 times with the same conditions listed in table 6.5, the average of all the parameters are determined, and lastly the deviation of each set is calculated. The bar plot of figure 6.11 shows the deviation from the average, for each of the 20 sets.

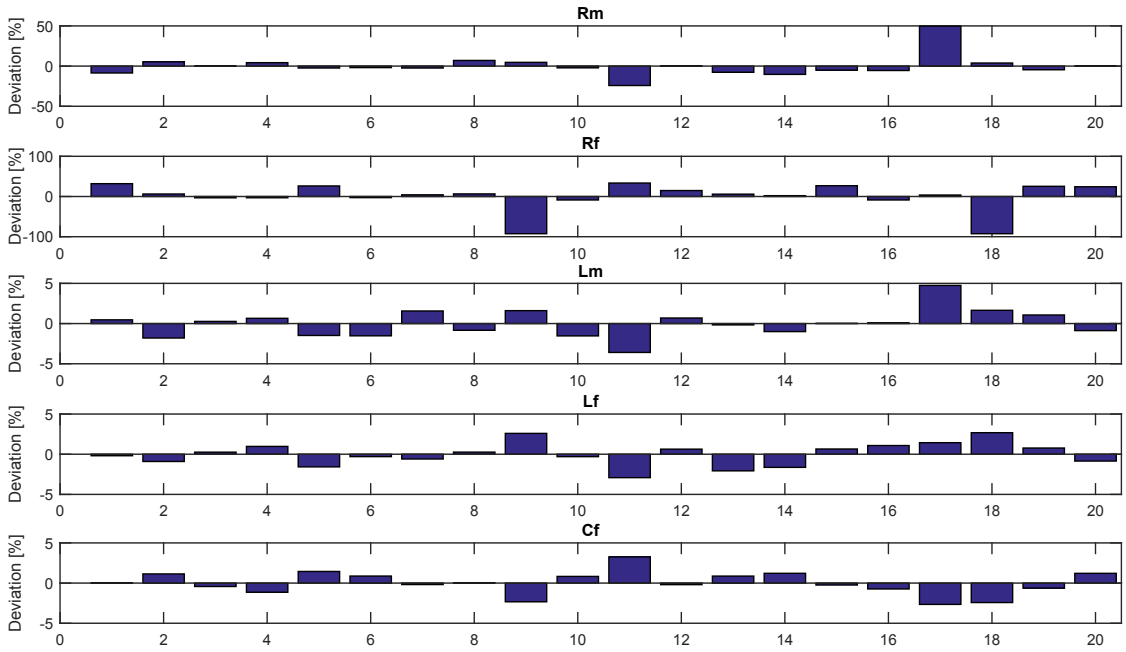


Figure 6.11: Deviation of each parameter over 20 different PSO results based on the same boundaries and conditions of the algorithm. Note the differences in the y-axis span for the different parameters.

The resistances are the only parameters that show significant deviation, which have been a problem in general with this parameter identification method. The remaining parameters are assessed to be determined accurately well, however, the resistances seem to be difficult to determine accurately.

The parameter identification method has been proved to accurately determine the parameters of the system. In the following chapter the implementation of the method on a stand alone solution is described.

Chapter 7

Implementation

The implementation section will include the description of the Real Time part of the LabVIEW program. The main functions of the LabVIEW program will be explained and the remaining code can be found on the enclosed CD. The main VIs from both the FPGA and the RT can be found in appendix C.1 "*FPGA VIs*" and appendix C.2 "*RT VIs*". For the implementation of the complete program on the embedded device, the experience gathered regarding the PSD and PSO algorithm must be considered so the embedded device's limited resources are fully utilized. A discussion of the this will follow the implementation section.

7.1 LabVIEW Real Time Implementation

For the RT program the main structure is shown in figure 7.1. When the system is started, it will indicate when all the VIs are initialized. The LCD screen and contacts (HMI), can be used to change the different parameters for the system excitation, such as switching frequency, duty cycle and the total sampling time. When all the settings are set, the output sequence is started. The collected data is used to estimate the PSD, the PSD and the pre-selected optimization settings are used to initialize the PSO algorithm. The PSO algorithm VI is running in a MathScipt VI, so the developed Matlab code can easily be implemented. The cost is evaluated outside of the MathScript function so the progress can be followed. When the algorithm is completed the result will be shown on the LCD screen.

Figure 7.1: The main structure of the RT program.

Figure 7.2 shows FPGA-RT communication loop. This handles the different boolean controls from the entire Main RT structure. A single VI is used to control the LCD screen, but can be called from other structure in the code. The VIs illustrated with an LCD screen and a '!' on figure 7.3 writes to this function.

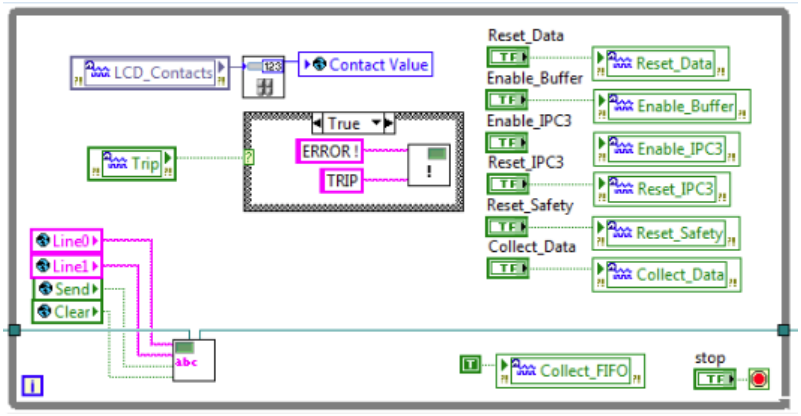


Figure 7.2: Communication sttucture.

The main RT structure, shown in figure 7.3 is responsible for executing the same sequence as illustrated in figure 7.1. The structure is a flat sequence structure and it executes each frame in order. In the first frame all the boolean controls are initialized, and the two safeties are reset. In the following frame the size of the DMA FIFOs are defined, and in the following frame they are started. In the same frame the settings for the parameter identification are set and send to the FPGA.

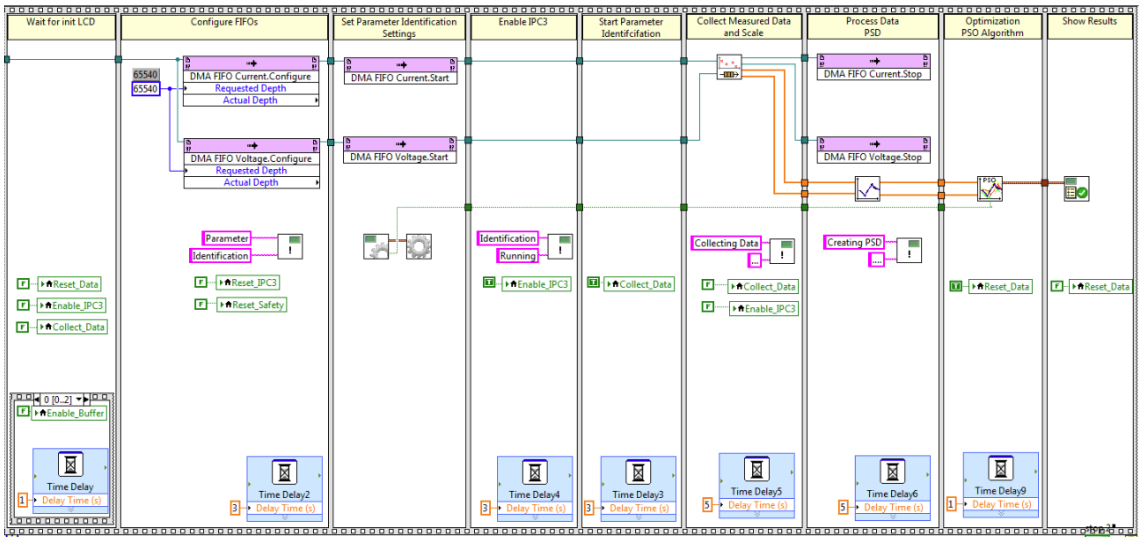


Figure 7.3: Main RT Structure.

The next frame enables the VLT, and then a delay is inserted to make sure the VLT is ready. The following frame starts the data collection and system excitation. The RT starts the sequence, and the FPGA then runs for a pre defined number of samples. The data from the sequence is in the next frame collected and scaled. In this frame the VLT is also disabled. The next frame stops the FIFOs and the VI estimates the PSD. At every step of the RT structure the LCD screen is updated. In the penultimate frame the PSO algorithm is carried out, shown in figure 7.4. In the final frame the results are presented on the LCD.

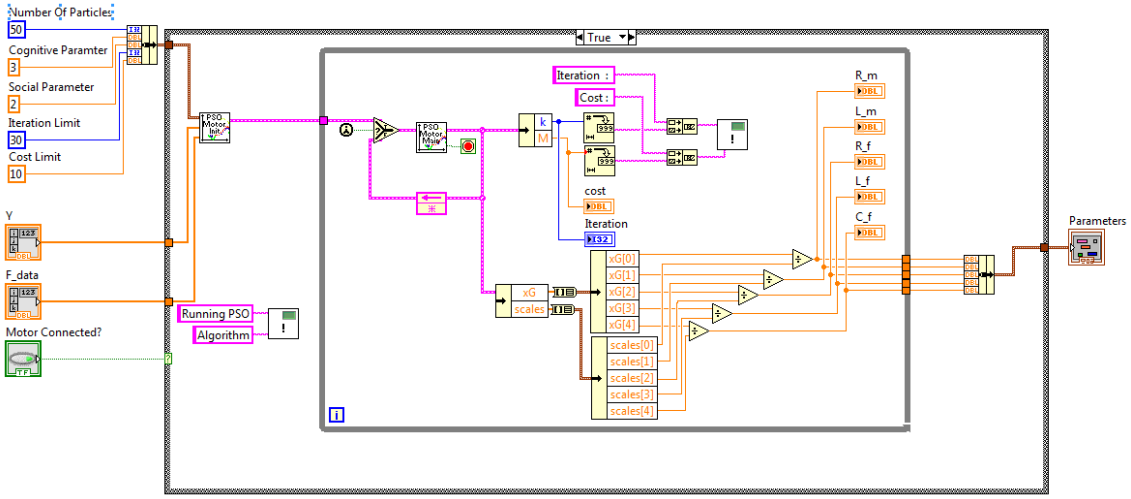


Figure 7.4: Main PSO loop.

The PSO algorithm changes dependent on whether or not the filter is connected alone or the motor is connected with the filter. The PSO is first initialized and then enters a while loop. The while loop will stop when the cost function is lower than the cost limit. The algorithm is constructed to step out of the MathScript function at every time step to follow the progress. When the cost is under the cost limit the while loop will stop and the found parameters will be sent to the final frame as shown in figure 7.3.

7.2 System Estimation and Curve Fitting Constants

In order to determine the settings for the algorithm, a laboratory sample has been used to utilize the method of parameter identification at a range of different settings. The laboratory setup was excited with a PRBS based PWM with a duty cycle of 55[%] at 9[kHz] switching frequency with a 2[kHz] band. System excitation and signal logging is done with the designed hardware for the laboratory setup, but all calculations in the following is computed on a desktop pc for convenience.

The voltage and current signals are used to estimate the system frequency response using the Welch method to estimate the PSD and CPSD of the signals. In the Welch

estimator a number of Hammings windows and the overlap of the windows has to be chosen. To choose the best settings for this case the signals have been used to estimate the response with a range of different settings, from 1-10 windows and 50-95[%] overlap. All the different responses have been used in the PSO algorithm, for curve fitting, to identify the system parameters. The results are shown in figure 7.5 where the final cost and the total time of the curve fitting, is shown for all of the different system responses.

Number of windows: [1 2 3 4 5 6 7 8 9 10]
Percentage overlap: [50 60 70 80 85 90 95]

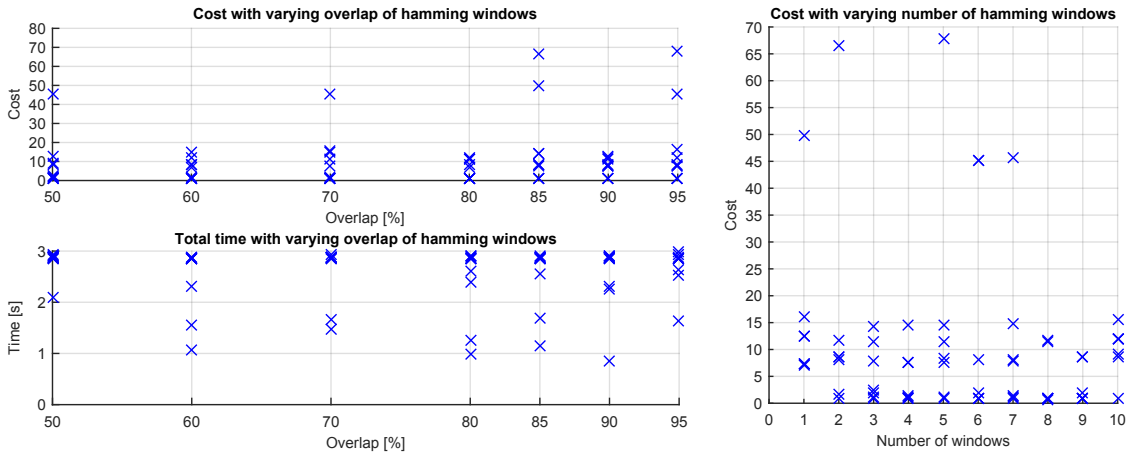


Figure 7.5: Plot of cost and total time of the curve fitting for 70 different combinations of number of hamming windows and windows overlap in the Welch estimator. (left) Cost and total time plotted for varying overlap [%], (right) Cost for varying number of Hamming windows.

Looking at the two plots on the left it is hard to say anything, in general, about the relationship between the window overlap and the final cost, but when looking at the total time of the PSO algorithm it seems to be slightly faster with more overlap. Since there does not seem to be much difference with changing overlap, the response is plotted in figure 7.6 for 50 and 80[%] both with 5 windows. There is only a small difference, showing the 80[%] to be slightly more smooth, but not much. The window overlap does not have much influence in this case, but is assessed to have more influence in cases when the response in general may be more noisy.

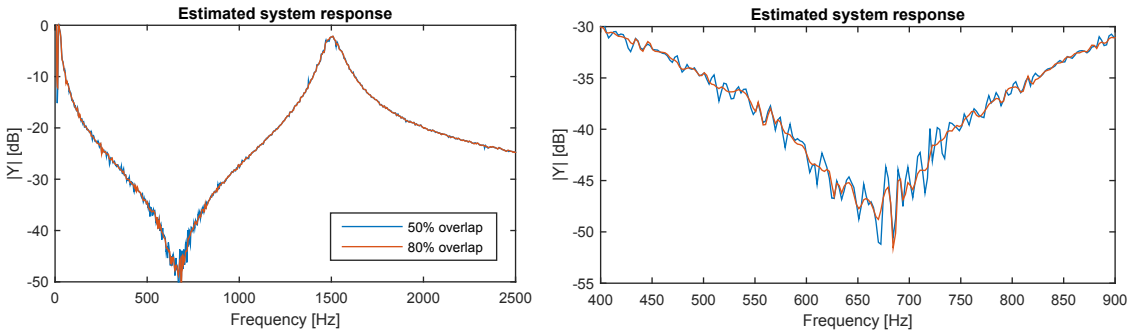


Figure 7.6: Frequency response plots of estimates based on 50 and 80[%] overlap of hamming windows.

With regard to the cost value, a value of 10 is considered sufficiently low to identify the parameters successfully, which is achieved in general. When looking at the plot on the right, of figure 7.5, it can be seen that the number of windows does not have that big an influence on the final cost value, why it would be advantageous to choose a low number of windows in order to reduce the computational efforts in estimating the PSD and CPSD. Since it is hard to tell the difference in the overall view, it is chosen to compare the case with 3 windows and 80-90[%] overlap with the same case but 8 windows. From figure 7.7 it seen that 3 windows is preferred over 8 windows, due to several reasons:

1. The PSO converges faster toward 0 cost in the case with only 3 windows
2. In the case with 3 windows, the cost seem more steady and is probably more robust than the case with 8 windows
3. The case with 3 windows is less computational expensive than with 8 windows, with regard to PSD and CPSD estimation.

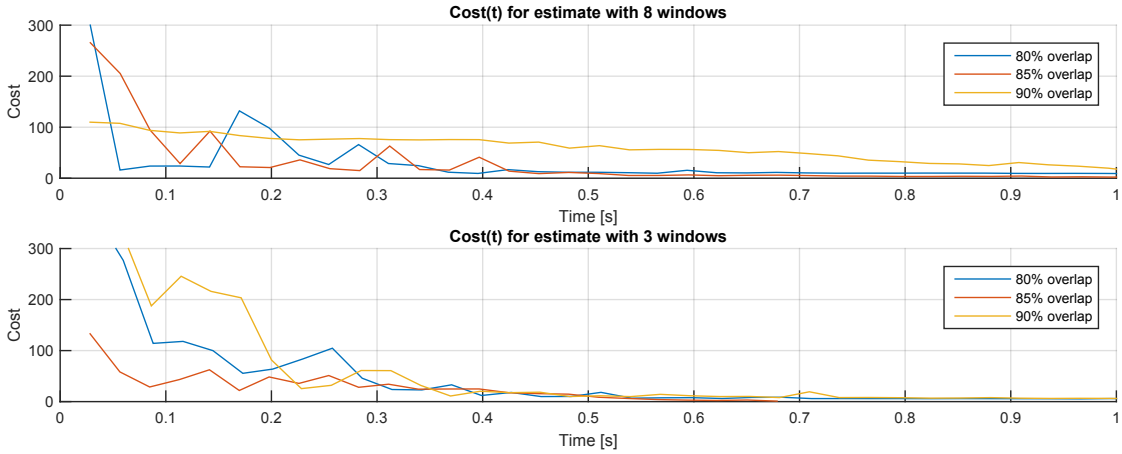


Figure 7.7: Curves showing the cost over time for the two cases with 3 and 8 windows, both with window overlap between 80-90[%]. (top) Cost for estimates with 8 windows, (bottom) Cost for estimates with 3 windows.

Settings of the Welch estimator has been investigated. Next is the settings of the PSO algorithm used for curve fitting. The basic settings of the PSO is to decide on the number of particles in the swarm, N_p, and to decide on the social and cognitive parameters, c_1 and c_2. The influence of these three basic constants are investigated in the follwing, where the response were estimated with 6 Hamming windows and 95[%] overlap. The final cost is decreased when the number of particles is increased, however, from 25 to 75 number of particles there is no big difference in the final cost, as shown in figure 7.8. Increasing the number of particles shows a linear relation to the total time used to run the PSO algorithm. When looking at the influence of c_1 and c_2 it is obvious that these must be chosen to be < 1 in order to yield low costs.

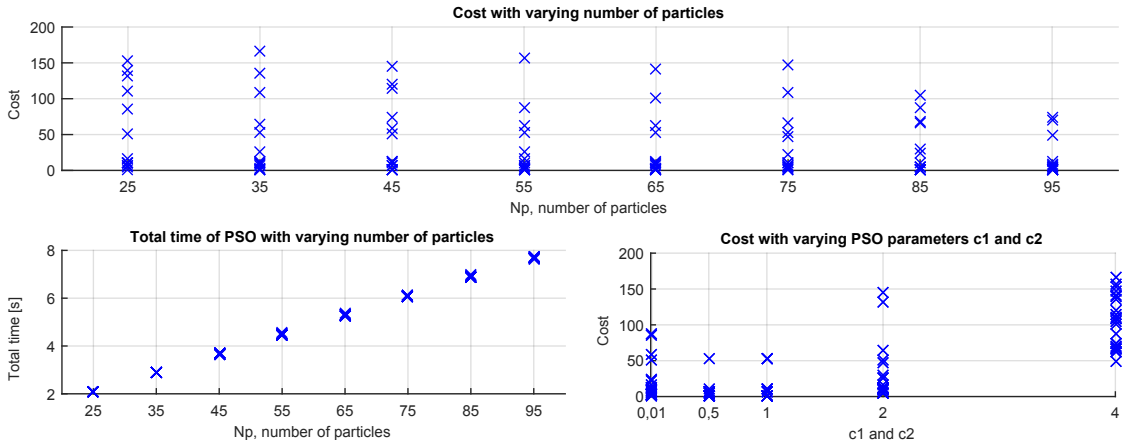


Figure 7.8: Results of a series of PSOs executed with different settings. (top) Shows the final cost with varying number of particles in the swarm, (bottom-left) shows the total time of the PSO with varying number of particles in the swarm, (bottom-right) Shows the final cost with varying c_1 and c_2.

In order to sufficiently visualize the effect of the number of particles the PSO have been executed with 35, 70, and 105 particles, and for each of these a set of $c_1 = c_2$, as can be seen from figure 7.9. When looking at the plot on the right it is observed how the PSO with only 35 particles, in general, uses more iterations to identify the parameters of the system. The top cost plot, shows the cost for the PSO with 35 particles and this shows great results for all $c \leq 0.6$, with a total time just around 0.5[s] for most of the cases. According to the plot of iterations the PSO's with 70 and 105 particles is able to identify the parameters in less interations than with 35 particles, but from studying the cost plots it is observed that the time used is higher in the case with 105 particles. However, looking at the cases with 70 particles, the time needed to identify the parameters is low for all $c \leq 0.4$, why it is concluded that around 70 particles and $c_1 = c_2 \leq 0.4$ should be used to utilize the PSO.

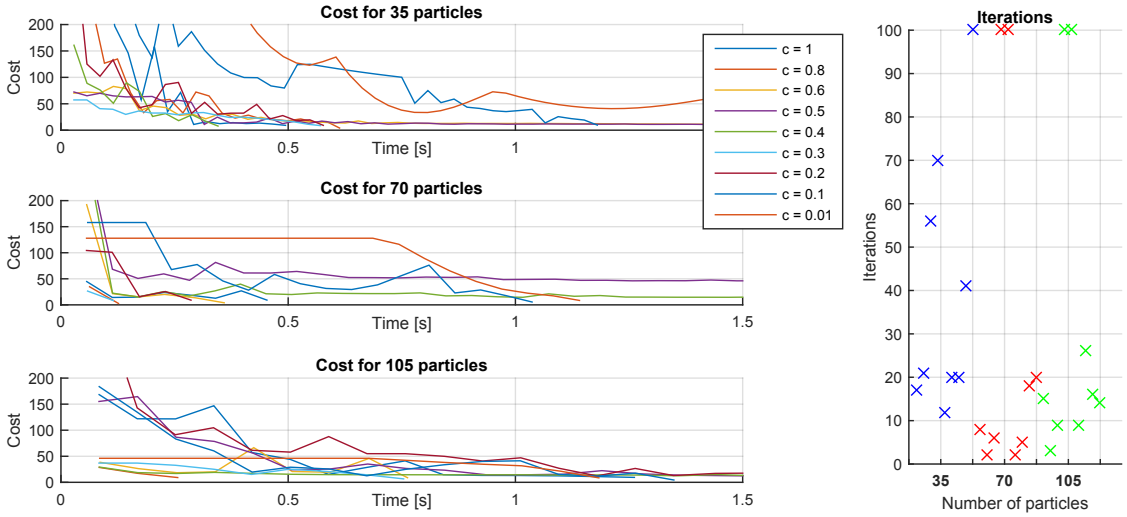


Figure 7.9: Results of varying c_1 and c_2 at different number of particles. (left) Shows the cost over time for the three cases $N_p = 35, 70, 105$, (right) shows the number of iterations, split in *(blue)* $N_p = 35$, *(red)* $N_p = 70$, and *(green)* $N_p = 105$.

Lastly it is investigated what frequencies can be cut away to reduce the computational effort of the algorithm. In figure 7.10 is shown the three cases that are investigated. For every section listed below, the PSO is executed 11 times and the results are plotted in figure 7.11 where the final cost, the total time and total number of iterations are shown for easy comparison.

Section 1: 100-2500[Hz]
Section 2: 300-2250[Hz]
Section 3: 500-2000[Hz]

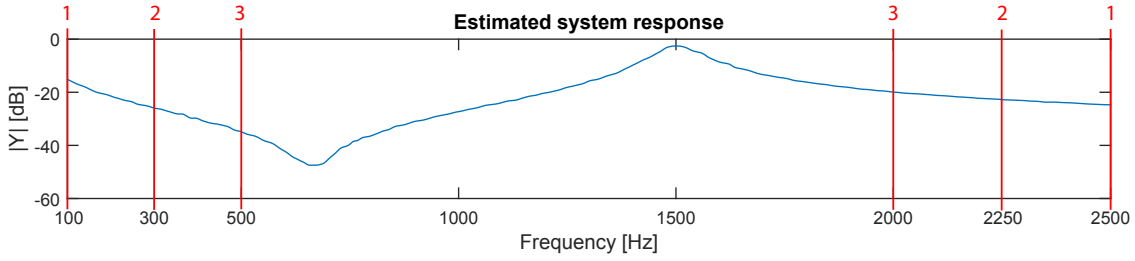


Figure 7.10: Estimated frequency response with lines to show the defined frequency sections.

In general there is no big difference in the final cost, shown in the left of figure 7.11, in all three frequency sections the PSO converges to a cost value below the cost limit of 10. When looking at the total time and the number of iterations it is clear that section 2 and 3 are faster than section 1. By comparing section 2 and 3 it is also possible to tell that section 2 is slightly faster than section 3. With regard to section 3 it is assessed that

the first cut is too close to the anti-resonance peak, why some of the dynamics of the system is lost in the PSO when section 3 is utilized.

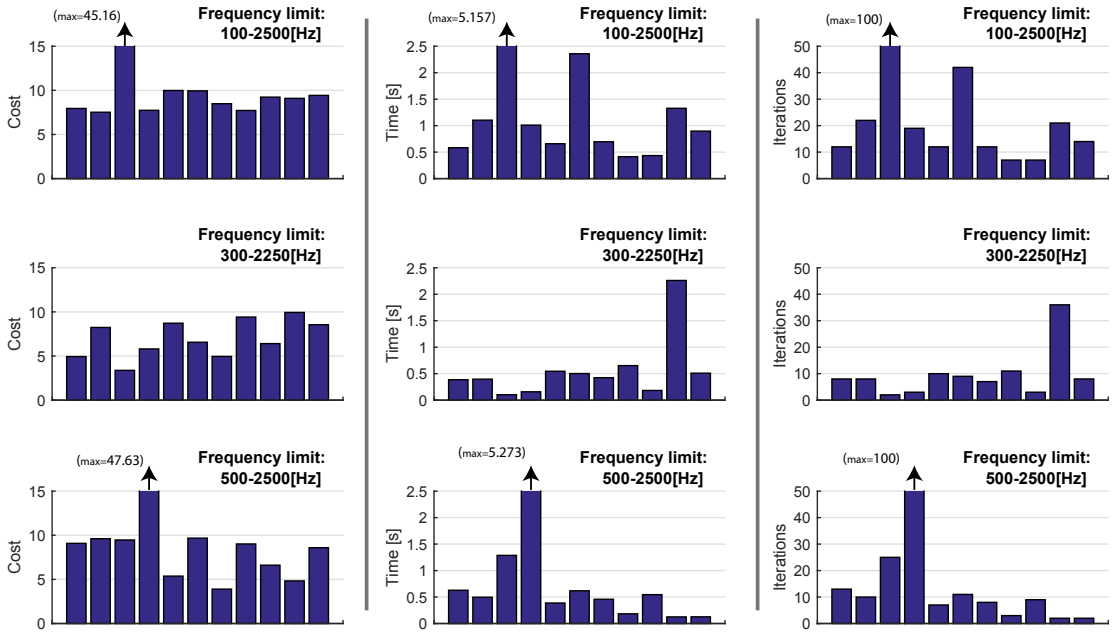


Figure 7.11: Results of different frequency sections. (left) Shows the final cost for the three sections, (middle-column) total time of the three sections, (right) the final number of iterations for the three sections.

7.2.1 Summary

Based on the previous investigation of the parameter identification method the following should be considered when choosing the settings of the method.

Hamming windows:
A low number of windows like 2, 3, or 4 windows, in order to reduce the computation effort in estimating the system response, also the convergence of the curve fitting algorithm were improved with fewer windows used.

Window overlap:
A high percentage of overlap in the range of 80-90[%] is preferred because this is shown to slightly decrease the total time of the PSO algorithm, relative to an overlap of 50[%].

Number of particles:
A swarm of around 70 particles should be used in order to reduce the total number of iterations and to maintain as short a PSO time span as possible, while still accurately determining the system parameters.

Social and cognitive constants:
These should be chosen to ≤ 0.4 in the case when $c_1 = c_2$ and may be considered as an equivalent to a step size in the optimization algorithm.

Frequency evaluation range:
It is found that the range 300-2250[Hz] yield great results with regard to low number of iterations. It is assessed possible to reduce the range further if needed, however the top frequency limit should be considered before reducing the lower frequency limit.

Chapter 8

Conclusion

Through the introduction to the report, it was clarified that the negative effects of controlling an electrical machine by PWM, could be mitigated by the addition of an LC-filter to the system. For the industry, it is highly advantageous for the inverter to automatically determine parameters of the motor is controlling, in order to control if more efficient. This feature is embedded in many inverters, such as the Danfoss VLT used for this project. However, when a filter is mounted in between the inverter and the motor, the system completely changes, and the parameters become more complicated to determine. The project has aimed to develop a stand alone solution, with high flexibility, that may be implemented on a 3-phase inverter, in order to determine the system parameters with an LC-filter connected.

The entire laboratory system was modeled mathematically, allowing the parameter identification to be tested before implementing it on the real system, where high voltages and currents may cause damage. This was especially important for determining how to best excite the system for parameter identification, where it is wanted to include as many dynamics as possible. A pseudo random signal was designed, which allows the system to be excited, without sweeping through the resonance frequency but still maintain a close to constant power spectrum density at all frequencies.

The system parameters are determined by curve-fitting, utilizing a least-squares based optimization algorithm. The system is described as a relationship between the controlled voltage into the system and the resulting current. These two values are measured and mapped to the frequency domain, by the use of spectral analyses. A mathematical model was developed, that describes the relationship between the voltage and current, as a function of the system parameters, in the frequency domain. This model was curve fitted, with an optimization algorithm, to the measured system response. A particle swarm optimization algorithm was designed, that is based on the principle of multiple particles that iteratively move closer and closer towards a minimum.

In order to have a quantifiable indicator of the accuracy of the identified parameters, system parameters from information slates, on the components, were used as the expected values. The estimated parameters from the parameter identification proved to

deviate from the expected, but were, however, very consistent. With large boundaries, a test was conducted to determine the consistency of the results of the optimization algorithm. While the resistances were not very consistent, the other parameters only deviated in the order of $1-5\%$ from their respective mean of 20 tests. From this it was assessed that the parameter identification found the actual parameters of the system, while the precision and validity of the expected values can be questioned. Additionally, the system identification also includes resistance in cables and connections, which can explain why the resistances always had the largest deviations. The parameter identification was concluded a success, however there is still room for fine tuning the parameter settings of both the spectral analysis and the optimization algorithm.

Hardware was designed and developed, capable of functioning as a stand alone solution for parameter identification. The measurements circuits consisted of shunt resistors, voltage dividers and applied $\Delta\Sigma$ Modulators as ADCs, providing cheap but high resolution, high bandwidth and low noise measurements. The IB2 also provides communication to the VLT's IGBT transistors for, not only parameter identification, but also the ability to control the motor, once the parameters are identified. All signals on the IB2 were thoroughly examined, and it showed that, although some connections should be terminated, communication on the board is capable of handling the high voltages and currents of the VLT phases and also the high frequency measurement signals.

The IB2 was designed to harbor the sbRIO-9636, the computational power behind the data acquisition and parameter identification. The sbRIO was programmed both on the FPGA level and the RT level, ensuring that the high frequency measurements can be handled. The IB2 and program include a Human Machine Interface (HMI) for control of the system without the need for an externally connected computer.

In conclusion a successful parameter identification method was developed, with the associated hardware and program to ensure a stand alone solution, that may be implemented on any 3-phase system with an LC-filter.

Chapter 9

Future Work

The project is considered successful as a parameter identification method was developed and the associated hardware, to realize it as an embedded stand alone solution. There are however, still a few loose strings that were neglected due to time limit of the project. If these ”leftovers” were to be investigated or solved, the project could be well on its way to being implemented in the industry. This is a matter for future work, for anyone willing to pursue this method of parameter identification.

The first matter to be considered is the hardware. As concluded in the impedance inspection of the PCB traces, several of the traces should be considered transmission lines and impedance matched. Looking at the signals, this was also evident, as some of the signals were subject to excessive ringing, causing the signal voltage to rise above what the receiving component is built for. Another consideration for the traces, with regard to impedance, is to investigate what trace widths and distance between trace and ground plane, would result in a more suitable impedance, for termination. There is also the matter of the DC-DC converters, supplying the $\Delta\Sigma$ Modulators, not adhering to the minimum creepage distance between the ground plane and the high voltage phase circuits. This could either be solved by finding a different solutions for power supply, a different PCB design or investigating which standard is applicable for the situation and their requirements. In general it would be advisable to create a new generation of the IB2, following the solutions described in this report, as they have proved to generate good results. A box should also be developed, for either the IB2 or eventual new generations, encasing the electronics in a shield, protecting the user and allowing the HMI to be fixated. It should also be noted that according to design standards, the creepage distance may be reduced if the electronics are properly shielded and isolated (Brucchi et al., 2012). Lastly, on the matter of hardware, the resistors may be the source of the voltage measurement output of the $\Delta\Sigma$ Modulator misbehaving. The frequency response of the resistors should be investigated, or some resistors with a known frequency response should replace them. The latter is recommended, as the max voltage of the applied resistors, just barely exceeds the required.

The next point to look into is the parameter identification process. The data measured when the system is excited is filtered by the use of Welch's method. In many cases, the anti-resonance peak contained oscillations, that made it harder for the optimization algorithm to converge. Other filtering methods should be inspected, or a more thorough investigation of the Welch parameters may be conducted. Reducing the higher frequency oscillations of the system response, can reduce that calculating time of the optimization algorithm and may even increase the accuracy of the results. Another point to investigate, is whether curve fitting the entire system response is necessary. An alternative suggestions would be to derive the equations for the resonance and anti-resonance peaks and optimize for these. The resistance values would only act on the dampening on these, and have little influence on the actual frequency. Therefore, the resistance may have to be determined some other way, but seeing as the resistances are already hard to determine, a different approach for determining these may be necessary in any case. This leads to another recommendation for future work. It could be investigated how well the system identification would perform if the parameters for the filter were first determined without the motor connected. Then the parameters of the motor could be determined with the filter parameters locked. This could ease the calculations for the optimization, since the parameters couple. In addition, the system equation for the filter has already been determined in this project.

The final topic for future work is the voltage and current measurements. First of all, the effective number of bits for the chosen $\Delta\Sigma$ Modulator is supplied by the datasheet. However, this value is theoretical and it might be advantageous to look into the actual effective number of bits. This leads to the design of the digital filter. A sinc3 filter was used, as recommended by the datasheet, and the effect of applying different decimation ratios was inspected. However, because the signals are so heavily oversampled, the full bandwidth available and potential resolution are not fully utilized. Different filters should be inspected, in order to see if the measured bit stream could produce better measurement results. Also, by reducing the decimation ratio, the throughput frequency is increased, allowing for a short system excitation and sampling time. If this gets short enough, the parameter identification may even be able to be executed while the system online, in between operation. This could keep the system parameters up to date, even if the system starts degrading over time.

Bibliography

Hirofumi Akagi and Shunsuke Tamura. A passive emi filter for eliminating both bearing current and ground leakage current from an inverter-driven motor. *IEEE*, 2006.

Vivek Alwayn. *Optical Network Design and Implementation*. Cisco Press, 2004. ISBN 978-1587141508.

Analog Devices AD7403. *AD7403. 16-Bit, Isolated Sigma-Delta Modulator*, 2014. URL `http://www.analog.com/media/en/technical-documentation/data-sheets/AD7403.pdf`. Accessed 18 November 2014.

ARCOL AP5025. *AP5025 8Watt Current Sense Chip Resistors*. URL `http://www.arcolresistors.com/wp-content/uploads/2012/08/AP5025-2-12.33.pdf`. Accessed 26 February 2015.

J. S. Arora. *Introduction to Optimum Design*. Elsevier, 2012. ISBN 978-0-12-381375-6.

Avago Shunt. *Design with Avago Technologies Isolation Amplifiers*, 2010. URL `www.avagotech.com/docs/AV02-2633EN`. Accessed 18 November 2014.

Avago Technologies AN5341. *Driver and Receiver Circuits for Avago SFH Series Plastic Fiber Components (PFC)*, 2012. URL `www.avagotech.com/docs/AV02-3323EN`. Accessed 26 February 2015.

Avago Technologies HFBR-0500ETZ Series. *HFBR-0500ETZ Series Versatile Link The Versatile Fiber Optic Connection*, 2014. URL `www.avagotech.com/docs/AV02-3283EN`. Accessed 26 February 2015.

B.J. Baliga. *Fundamentals of Power Semiconductor Devices*. Springer US, 2010. ISBN 9780387473147. URL `https://books.google.com/books?id=UiqrUWrYZXkC`.

Fabio Brucchi, Wolfgang Peinhopf, and Infineon Isolation. *Application Note AN 2012-10 Electrical safety and isolation in high voltage discrete component applications and design hints*, 2012. URL `https://www.infineon.com/dgdl/Infineon+-+Application+Note+-+PowerMOSFETs+-+CoolMOS%E2%84%A2+-+Electrical+Safety+and+Isolation+in+high+voltage+Applications.pdf?fileId=db3a30433d1d0bbe013d20e0cbf017fe`. Accessed 26 February 2015.

Shaotang Chen and Thomas A. Lipo. Circulating type motor bearing current in inverter drives. *IEEE*, 1998.

Jeppe H. Christensen and Magnus L. Weber. Automatic parameter identification for an ac-motor with an lc-filter. Technical report, Department of Energy Technology, Aalborg University, Denmark, 2014.

J. W. Cooley and J. W. Tukey. An algorithm for the machine calculation of complex fourier series. *JSTOR*, 1964.

J. W. Cooley, P. A. W. Lewis, and P. D. Welch. The application of the fast fourier transform algorithm to the estimation of spectra and cross spectra. *Elsevier*, 1969.

Danfoss. *Danfoss - Output Filters Design Guide*, 2010. URL http://www.danfoss.com/NR/rdonlyres/27F81E71-3779-4406-8EA0-849044873F59/0/Output_Filters_Design_Guide.pdf. Accessed 18 November 2014.

Danfoss. *Danfoss AutomationDrive Selection*, 2011. URL http://www.ctiautomation.net/PDF/Danfoss/Danfoss-AutomationDrive-Selection.pdf. Accessed 18 November 2014.

Analog Devices and Hank Zumbahlen. *Linear Circuit Design Handbook Ch.12*, 2008. URL http://www.analog.com/library/analogDialogue/archives/43-09/EDch%2012%20pc%20issues.pdf. Accessed 26 February 2015.

DSPrelated, 2007. Dsprelated - cross power spectral density. URL http://www.dsprelated.com/showmessage/81132/1.php. Accessed 26 February 2015.

P. Duhamel and M. Vetterli. Fast fourier transforms: Atutorial review and a state of the art. *Elsevier*, 1990.

Benny Endelt. Gradient based optimization schemes - nonlinear system identification and inverse modeling, applied newton and trust region methods. Technical report, Alborg University, Department of Mechanical and Manufacturing Engineering, 2011.

Norbert Hanigovszki, Jørn Landkildehus, and Frede Blaabjerg. Output filters for ac adjustable speed drives. *IEEE*, 2007.

Paul Horowitz and Winfield Hill. *The Art of Electronics*. Cambridge University Press, 2015. ISBN 978-0-521-80926-9.

Infineon Deadtime. *How to calculate and minimize the dead time requirement for IGBTs properly*, 2007. URL https://www.infineon.com/dgdl/Infineon-AN2007_04_Deadtime_calculation_for_IGBT_modules-AN-v1.0-en.pdf.

Infineon FS150R12KT3. *IGBT Module, FS150R12KT3*, 2014. URL http://www.infineon.com/dgdl/Infineon-FS150R12KT3-DS-v02_01-en_de.pdf.

Infineon IGBT. *600/1200V IGBTs Set Benchmark Performance in High Switching Speed Applications*, 2014. URL http://www.power-mag.com/pdf/feature_pdf/1274195604_Infineon_Feature_Layout_1.pdf.

Infineon IGBT Modules Explanation. *Industrial IGBT Modules Explanation of Technical Information*, 2013. URL https://www.infineon.com/dgdl/Infineon-AN2011_05_IGBT_Modules_Explanation-AN-v1.1-en.pdf.

Infineon IGBT3 T3. *Characteristic differences between 1200V IGBT3 modules of the E3 and T3 series*, 2005. URL https://www.infineon.com/dgdl/Infineon-AN2005_02_Differences_of_IGBT3_generations-AN-v1.0-en.pdf.

International Rectifier. *IGBT reverse conduction characteristics Hard-switching and soft-switching*, 2013. URL http://www.irf.com/product-info/igbt/switching.pdf.

IXYS Rectifier. *Three Phase Rectifier Bridge with IGBT and Fast Recovery Diode for Braking System*, 2008. URL http://www.europowercomponents.com/media/uploads/VUB145-16no1.pdf.

Karel J. Keesman. *System Identification.* Springer, 2011. ISBN 978-0-85729-521-7.

Kewal K. Saluja. Linear feedback shift registers theory and applications, 1991. URL http://homepages.cae.wisc.edu/~ece553/handouts/LFSR-notes.PDF.

Erwin Kreyszig. *Advanced Engineering Mathmatics - Tenth Edition.* John Wiley & Sons, 2011. ISBN 978-0-471-64613-7.

Laboratory Soete. Failure analysis of deep groove ball bearings of an electric motor. *Elsevier*, 2004.

W. C. Lo, C. C. Chen, Z. Q. Zhu, Lie Xu, David Howe, and K. T. Chau. Acoustic noise radiation by pwm-controlled induction machine drives. *IEEE*, 2000.

Laszlo Mathe. Product sound: Acoustically pleasant motor drives. Technical report, Alborg University, Energy, 2010.

Ned Mohan, Tore M. Undeland, and William P. Robbins. *Power Electronics - Converters, Applications, and Design.* John Wiley & Sons, 1989. ISBN 978-0-471-22693-2.

Mouser Electronics AP5025. Mouser electronics part ap5025 r005 f. URL http://dk.mouser.com/ProductDetail/Ohmite/TGHGCR0100FE/?qs=sGAEpiMZZMtlleCFQhR%2fzUV1A4jD%2fILUqfrrvYttpzg%3d. Accessed 26 February 2015.

Mouser Electronics TGHG. Mouser electronics part tghgcr0100fe. URL http://dk.mouser.com/ProductDetail/Arcol/AP5025-R005-F/?qs=Q8MESGXhlr7o0cXGojCi4g%3D%3D. Accessed 26 February 2015.

Annette Muetze and Andreas Binder. Calculation of circulating bearing currents in machine of inverter-based drive systems. *IEEE*, 2007.

National Instruments - FPGA. *NI sbRIO-9636*, 2013. URL http://sine.ni.com/nips/cds/view/p/lang/en/nid/210421. Accessed 26 February 2015.

National Instruments - FPGA. *OEM OPERATING INSTRUCTIONS AND SPECIFICATIONS NI sbRIO-9605/9606 and NI sbRIO-9623/9626/9633/9636 Single-Board RIO OEM Devices*, 2014. URL http://www.ni.com/pdf/manuals/373378c.pdf. Accessed 26 February 2015.

National Instruments - FPGA Guide. *NI LabVIEW High-Performance FPGA Developers Guide*, 2014. URL download.ni.com/pub/gdc/tut/labview_high-perf_fpga_v1.1.pdf. Accessed 26 February 2015.

National Instruments -Scan Clock. *Using NI CompactRIO Scan Mode with NI LabVIEW Softwares*, 2013. URL http://www.ni.com/white-paper/7338/en/. Accessed 26 February 2015.

National Instruments Anti-Aliasing. *What are Anti-Aliasing Filters and Why are They Used?*, 2000. URL http://digital.ni.com/public.nsf/allkb/68F14E8E26B3D101862569350069E0B9. Accessed 26 February 2015.

National Instruments LCD Driver. Lcd character display driver labview fpga driver for hd44780-compatible displays. URL http://sine.ni.com/nips/cds/view/p/lang/en/nid/210210.

National Instruments tannerite. Lv fpga pseudo random number generator, 2007. URL https://decibel.ni.com/content/docs/DOC-1143/version/4.

National Instruments Termination. *Proper Termination for High-Speed Digital I/O Applications*, 2014. URL http://www.ni.com/white-paper/3854/en/. Accessed 26 February 2015.

New Wave Instruments. Linear feedback shift registers, 2010. URL http://www.newwaveinstruments.com/resources/articles/m_sequence_linear_feedback_shift_register_lfsr.htm.

P. Otis. Energy information administration - eia, 2013. URL http://www.eia.gov/todayinenergy/detail.cfm?id=13431. Accessed 14 February 2015.

Power Systems Design. Optimizing power modules. URL http://www.powersystemsdesign.com/optimizing-power-modules.

M. Rahimo, W. Lukasch, C. von Arx, A. Kopta, R. Schnell, S. Dewar, and S. Linder. Novel soft-punch-through (spt) 1700v igbt sets benchmark on technology curve. *IEEE*, 2001.

D. Schlichthärle. *Digital Filters: Basics and Design*. Springer, 2011. ISBN 9783642143250. URL https://books.google.com/books?id=Crggzv_PqqAC.

Leroy Somer. Leroy somer dyneo - synchronous motors with permanent magnets, techinal catalogue. URL http://www.emersonindustrial.com/en-EN/leroy-somer-motors-drives/Products/Permanent-magnet-synchronous-motors/Pages/LSRPM-Dyneo%C2%AE.aspx. Accessed 19 September 2014.

B. Sutthiphornsombat, A. Khoobroo, and B. Fahimi. Mitigation of acoustic noise and vibration in permanent magnet synchronous machines drive using field reconstruction method. *IEEE*, 2010.

P. Szczupak and J. M. Pacas. Automatic parametric model identification method for ac-drives with lc-filter. *Diagnostics for Electric Machines, Power Electronics and Drives, 2007. SDEMPED 2007. IEEE International Symposium on*, (978-1-4244-1062-0):208–214, 2007a.

P. Szczupak and M. Pacas. Identification of ac-machines with lc-filters by utilising non-parametric models. *IEEE*, 2007b.

Texas Instruments HCT540. *High-Speed CMOS Logic Octal Buffer and Line Drivers, Three-State*, 2004. URL www.ti.com/lit/gpn/cd74hct540. Accessed 26 February 2015.

VISHAY MRS25. *Professional Thin Film Leaded Resistors*. URL http://www.farnell.com/datasheets/1862243.pdf. Accessed 26 February 2015.

A. von Jouanne, P. Enjetti, and W. Gray. The effect of long motor leads on pwm inverter fed ac motor drive systems. *IEEE*, 1995.

A. von Jouanne, P. Enjetti, W. Gray, and D. A. Rendusara. Filtering techniques to minimize the effect of long motor leads on pwm inverter-fed ac motor drive systems. *IEEE*, 1996.

Alexander Weiler, Alexander Pakosta, and Texas Instruments Guidelines. *Application Report SCAA082 High-Speed Layout Guidelines*, 2006. URL www.ti.com/lit/an/scaa082/scaa082.pdf. Accessed 26 February 2015.

P. D. Welch. The use of fast fourier transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms. *IEEE*, 1967.

XILINX - DSP48E. *7 Series DSP48E1 Slice*, 2014. URL http://www.xilinx.com/products/intellectual-property/dsp48_macro.html#overview. Accessed 26 February 2015.

Appendix A

Optimization Algorithms

A.1 Levenberg-Marquardt Algorithm

```
%Using  Levenberg-Marquardt's method for least squares, the ...
    parameters
%of a PMSM motor are determined.
%Jeppe Haals Christensen & Magnus Lønstrup Weber
%Aalborg University EMSD3 2014
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Initial parameter guess:
%x = [0.3; 5.5e-3; 0.1; 1.1e-3; 1.47e-5];
%scaled [R_mx1 L_mx100 R_fx1 L_fx100 C_fx10000]
x = [0.3; 0.55; 0.1; 0.11; 0.147];
%Initial Lambda
Lambda = 100;
%Algorithm options:
epsilon1=1e-5;
epsilon2=1e-10;
k_max=100;
%Counters:
k = 1;
F_count = 1;
%Cost function weight
Weight_value = 1000;
Weight = diag((ones(length(F_data),1)'*Weight_value));
%Loads experimental data:
Experimental_data_osci
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Initial values:
%Function value vector and Jacobian
syms R_m L_m R_f L_f C_f
x_syms = [R_m; L_m; R_f; L_f; C_f];
%Converts the frequency points from the tfestimate, F_data, to ...
    complex
w = 1j*2*pi*F_data;
%Scaling of parameters
Par_scale = [1 1/100 1 1/100 1/10000];
```

```
a1 = (R_m*Par_scale(1,1)) * (C_f*Par_scale(1,5));
a2 = (L_m*Par_scale(1,2)) * (C_f*Par_scale(1,5));
b0 = (R_m*Par_scale(1,1)) + (R_f*Par_scale(1,3));
b1 = (L_m*Par_scale(1,2)) + (L_f*Par_scale(1,4)) + ...
    (R_m*Par_scale(1,1))...
* (R_f*Par_scale(1,3)) * (C_f*Par_scale(1,5));
b2 = (R_m*Par_scale(1,1)) * (L_f*Par_scale(1,4)) * ...
    (C_f*Par_scale(1,5))...
+ (R_f*Par_scale(1,3)) * (L_m*Par_scale(1,2)) * ...
    (C_f*Par_scale(1,5));
b3 = (L_m*Par_scale(1,2)) * (L_f*Par_scale(1,4)) * ...
    (C_f*Par_scale(1,5));
%Calculates the magnitude of admittance vector
Y = sym(zeros(1,length(w)));
for i = 1:length(w)
Y(i) = ...
   abs(2/3*(w(i)^2*a2+w(i)*a1+1)/(w(i)^3*b3+w(i)^2*b2+w(i)*b1+b0));
end
%Residual vector
r_syms = Y_data./transpose(Y)-1;
%Evaluates the residual vector at start guess
r = double(subs(r_syms, x_syms, x));
%Jacobian calculated analytically
J_syms = jacobian(r_syms,x_syms);
%Jacobian evaluated at start guess
J = double(subs(J_syms,x_syms, x));
J = abs(J);
%Sum of residual squares
cost = 0.5*(r'*Weight*r);
%Gradient
g = transpose(J)*r;
% Length of the gradient
l=norm(g);
%Hessian estimate
H = transpose(J)*J;
%Storing original parameters
X(k,:) = x;
F(k) = cost;
G(k,:) = g;
L(k) = Lambda;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Iterating:
while (l≥epsilon1 && cost>epsilon2 && k<k_max)
    %Calculate s_k:
    s = -inv(H+eye(length(x))*Lambda)*g;
    %Calculate x_trial
    x_trial = x+s;
    %Update the residual vector
    r = double(subs(r_syms, x_syms, x_trial));
    %Update sum of squares
    cost_trial=0.5*(r'*Weight*r);
    F_count = F_count + 1;
```

```
80      %Update Lambda if step size is acceptable
81      if (cost_trial≤cost)
82          Lambda=Lambda/2;
83      else
84      % Reduce step size until the step is acceptable
85      while(cost_trial>cost)
86          Lambda=Lambda*2;
87          s = -inv(H+eye(length(x))*Lambda)*g;
88          x_trial = x+s;
89          r = double(subs(r_syms, x_syms, x_trial));
90          cost_trial=0.5*(r'*Weight*r);
91          F_count = F_count + 1;
92      end
93      end
94      %Update variables and parameters
95      k = k + 1;
96      cost = cost_trial;
97      x = x_trial;
98      r = double(subs(r_syms, x_syms, x));
99      J = double(subs(J_syms,x_syms, x));
100     J =abs(J);
101     g = J'*r;
102     H = J'*J;
103     l=norm(g);
104     X(k,:) = x;
105     F(k) = cost;
106     G(k,:) = g;
107     L(k) = Lambda;
108 end
```

A.2 Particle Swarm Optimization - Proof of Concept

```
%%% Particle Swarm Optimization (PSO)
%%% Test of PSO based on experimental data from previous project
clc
clear all
close all

scales = [1;1;1;1;1];

load ExpData.mat

%% Initialization:
Np = 35; %Number of particles, usually 5n to 10n, n: no. design ...
    variables
c1 = 2; %cognitive parameter
c2 = 2; %social parameter
kmax = 100; %limit of iterations
costLim = 5;
k = 0; %counter
v0 = zeros(5,Np); %initial velocity
w = 1j*2*pi*F_data;

%% Initial parameter guess:
% x = [R_m; L_m; R_f; L_f; C_f];
% dx = [low;high] are the range of each variable
% each particle gets a random initial variable set
x = zeros(5,Np);
dx1 = [0.05;2]*scales(1);
for q=1:Np
    x(1,q) = dx1(1)+(rand()*(dx1(2)-dx1(1)));
end
dx2 = [1;10]*1e-3*scales(2);
for q=1:Np
    x(2,q) = dx2(1)+(rand()*(dx2(2)-dx2(1)));
end
dx3 = [0.05;5]*scales(3);
for q=1:Np
    x(3,q) = dx3(1)+(rand()*(dx3(2)-dx3(1)));
end
dx4 = [0.5;5]*1e-3*scales(4);
for q=1:Np
    x(4,q) = dx4(1)+(rand()*(dx4(2)-dx4(1)));
end
dx5 = [5;20]*1e-6*scales(5);
for q=1:Np
    x(5,q) = dx5(1)+(rand()*(dx5(2)-dx5(1)));
end

%% Initial Cost:
cost = zeros(1,Np);
```

```
for q=1:Np
    Y = Y_fun3(x(1,q),x(2,q),x(3,q),x(4,q),x(5,q),scales,F_data,w);
    r(1:length(w),q) = Y_data./transpose(Y)-1;
    cost(q) = 0.5*(r(1:length(w),q)'*Weight*r(1:length(w),q));
end
q=1;

[M,I] = min(cost);

xP = x; % Best solution for each particle
xG = x(1:5,I); % Best solution for swarm
v1 = v0; % velocities

%% Iterations:
for k=1:kmax
    %Calculate particle velocities
    for q=1:Np
       v1(1:5,q) = v1(1:5,q) + rand()*c1*(xP(1:5,q)-x(1:5,q)) + ...
           rand()*c2*(xG-x(1:5,q));
    end
    q=1;

    %Calculate new particel positions
    for q=1:Np
       x(1:5,q) = x(1:5,q) + v1(1:5,q);
    end
    q=1;

    %Check for bounds of each particles design variables
    for q=1:Np
        if x(1,q) < dx1(1)
            x(1,q) = dx1(1);
        elseif x(1,q) > dx1(2)
            x(1,q) = dx1(2);
        end
        if x(2,q) < dx2(1)
            x(2,q) = dx2(1);
        elseif x(2,q) > dx2(2)
            x(2,q) = dx2(2);
        end
        if x(3,q) < dx3(1)
            x(3,q) = dx3(1);
        elseif x(3,q) > dx3(2)
            x(3,q) = dx3(2);
        end
        if x(4,q) < dx4(1)
            x(4,q) = dx4(1);
        elseif x(4,q) > dx4(2)
            x(4,q) = dx4(2);
        end
        if x(5,q) < dx5(1)
            x(5,q) = dx5(1);
```

```
100             elseif x(5,q) > dx5(2)
101                 x(5,q) = dx5(2);
102             end
103         end
104         q=1;
105 
106         costOld = cost;
107         for q=1:Np
108             Y = ...
                    Y_fun3(x(1,q),x(2,q),x(3,q),x(4,q),x(5,q),scales,F_data,w);
109             r(1:length(w),q) = Y_data./transpose(Y)-1;
110             cost(q) = 0.5*(r(1:length(w),q)'*Weight*r(1:length(w),q));
111         end
112         q=1;
113 
114         %Check for best solutions for each particle
115         for q=1:Np
116             if cost(q) > costOld(q)
117                 xP(1:5,q) = xP(1:5,q);
118             elseif cost(q) < costOld(q)
119                 xP(1:5,q) = x(1:5,q);
120             end
121         end
122         q=1;
123 
124         %Check for best solution for swarm
125         [Mold,Iold] = min(costOld);
126         [M,I] = min(cost);
127         if M < Mold
128            xG = x(1:5,I);
129         end
130         hist(k) = cost(I);
131 
132         if cost(I) ≤ costLim
133            iterations = k;
134            k=kmax;
135            break
136         end
137     end
```

A.3 Parameter Identification

A.3.1 PSDEstimate

```
%%% Estimation of systems frequency response
%%% EMSD 4 - Master Thesis 2015

function [F_data,Y_data] = ...
    PSDEstimate(fs,VM,CM,nfft,overlap,window) %VM,CM: array

% System estimation
[Y_data_complex,F_data_full] = ...
    tfestimate(VM,CM,window,overlap,nfft,fs);
Y_data_full = abs(Y_data_complex);
Y_data_full_dB = 20*log10(Y_data_full);

I = int64(4000/(F_data_full(7)-F_data_full(6)));
Y_data = abs(Y_data_full(1:I));
F_data = F_data_full(1:I);
```

A.3.2 PSO

```
%%% Particle Swarm Optimization (PSO)
%%% EMSD 4 - Master Thesis 2015
%%% Magnus, Jeppe, and Nikolai

function [Parameters,costhist,iterations,TimeArray] = ...
    PSO(F_data,Y_data,LowLim,UpLim,Np,c1,c2,kmax,costLim, ...
    scales,dx1,dx2,dx3,dx4,dx5)
Y_data = Y_data+1e-5;

%% Initialization:
k = 0; %counter
v0 = zeros(5,Np); %initial velocity
w = 1j*2*pi*F_data; %frequency vector

Iq = int64(LowLim/(F_data(7)-F_data(6)))+1; % Lower frequency ...
    limit (cutout)
Iqp = int64(UpLim/(F_data(7)-F_data(6)))+1; % Upper frequency ...
    limit (cutout)

%% Initial parameter guess:
% x = [R_m; L_m; R_f; L_f; C_f];
% dx = [low;high] are the range of each variable
% each particle gets a random initial variable set
x = zeros(5,Np);
for q=1:Np
    x(1,q) = dx1(1)+(rand(1)*(dx1(2)-dx1(1)));
    x(2,q) = dx2(1)+(rand(1)*(dx2(2)-dx2(1)));
    x(3,q) = dx3(1)+(rand(1)*(dx3(2)-dx3(1)));
```

```
    x(4,q) = dx4(1)+(rand(1)*(dx4(2)-dx4(1)));
    x(5,q) = dx5(1)+(rand(1)*(dx5(2)-dx5(1)));
end

%% Initial Cost:
cost = zeros(1,Np);
r = zeros((Iqp-Iq),Np);
for q=1:Np
    Y = Y_fun_motor(x(1,q),x(2,q),x(3,q),x(4,q),x(5,q),scales,w);
    r(1:(Iqp-Iq)+1,q) = Y_data(Iq:Iqp)./transpose(Y(Iq:Iqp))-1;
    cost(q) = 0.5*(r(1:(Iqp-Iq)+1,q)'*r(1:(Iqp-Iq)+1,q));
end
q=1;

[M,I] = min(cost);

xP = x; % Best solution for each particle
xG = x(1:5,I); % Best solution for swarm
v1 = v0; % velocities

%% Iterations:
hist = zeros(1,kmax);
YG = Y_fun_motor(xG(1),xG(2),xG(3),xG(4),xG(5),scales,w);

TimeArray = zeros(kmax,1);
tic
for k=1:kmax
    %Calculate particle velocities
    for q=1:Np
       v1(1:5,q) = v1(1:5,q) + rand(1)*c1*(xP(1:5,q)-x(1:5,q)) + ...
           rand(1)*c2*(xG-x(1:5,q));
    end
    q=1;

    %Calculate new particel positions
    for q=1:Np
       x(1:5,q) = x(1:5,q) + v1(1:5,q);
    end
    q=1;

    %Check for bounds of each particles design variables
    for q=1:Np
        if x(1,q) < dx1(1)
            x(1,q) = dx1(1);
        elseif x(1,q) > dx1(2)
            x(1,q) = dx1(2);
        end
        if x(2,q) < dx2(1)
            x(2,q) = dx2(1);
        elseif x(2,q) > dx2(2)
            x(2,q) = dx2(2);
        end
```

```
76          if x(3,q) < dx3(1)
77              x(3,q) = dx3(1);
78          elseif x(3,q) > dx3(2)
79              x(3,q) = dx3(2);
80          end
81          if x(4,q) < dx4(1)
82              x(4,q) = dx4(1);
83          elseif x(4,q) > dx4(2)
84              x(4,q) = dx4(2);
85          end
86          if x(5,q) < dx5(1)
87              x(5,q) = dx5(1);
88          elseif x(5,q) > dx5(2)
89              x(5,q) = dx5(2);
90          end
91      end
92      q=1;
93
94      costOld = cost;
95      for q=1:Np
96          Y = ...
                Y_fun_motor(x(1,q),x(2,q),x(3,q),x(4,q),x(5,q),scales,w);
97          r(1:(Iqp-Iq)+1,q) = Y_data(Iq:Iqp)./transpose(Y(Iq:Iqp))-1;
98          cost(q) = 0.5*(r(1:(Iqp-Iq)+1,q)'*r(1:(Iqp-Iq)+1,q));
99      end
100     q=1;
101
102     %Check for best solutions for each particle
103     for q=1:Np
104         if cost(q) > costOld(q)
105             xP(1:5,q) = xP(1:5,q);
106         elseif cost(q) < costOld(q)
107             xP(1:5,q) = x(1:5,q);
108         end
109     end
110     q=1;
111
112     %Check for best solution for swarm
113     [Mold,Iold] = min(costOld);
114     [M,I] = min(cost);
115     if M < Mold
116        xG = x(1:5,I);
117     end
118     hist(k) = cost(I);
119     YG = Y_fun_motor(xG(1),xG(2),xG(3),xG(4),xG(5),scales,w);
120
121     TimeArray(k,1) = toc;
122
123     if cost(I) ≤ costLim
124        iterations = k;
125        k=kmax;
126        break
```

```
    end
    if k == kmax
    iterations = kmax;
    break
    end
end

costhist = (hist(1:iterations))';
Parameters = xG./scales;
iterations = iterations;
```

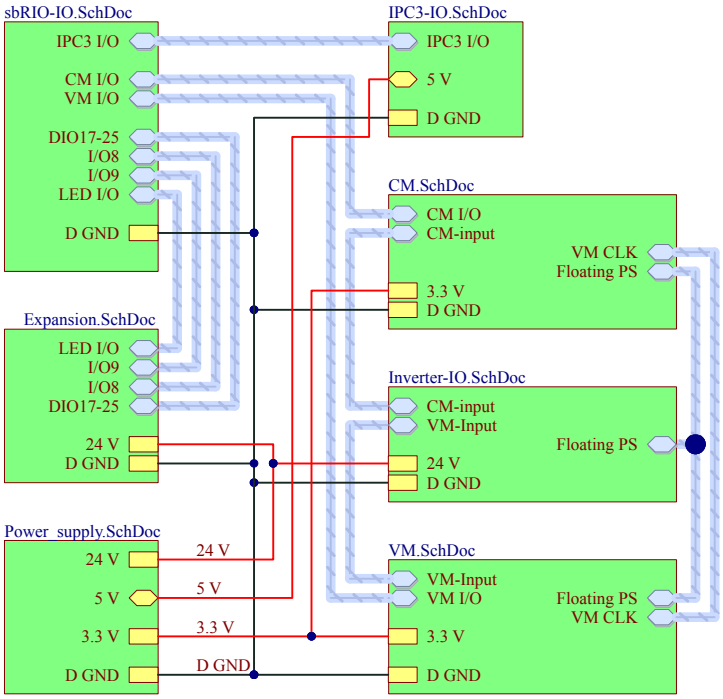
A.3.3 Y

```
% scales = [1e-4;1e-2;1e-4;1e-2;1];
% R_m = 1.08*scales(1);
% L_m = 7.29e-3*scales(2);
% R_f = 0.87*scales(3);
% L_f = 1.98e-3*scales(4);
% C_f = 7.59e-6*scales(5);

function [Y] = Y_fun_motor(R_m,L_m,R_f,L_f,C_f,scales,w)

a1 = R_m/scales(1) * C_f/scales(5);
a2 = L_m/scales(2) * C_f/scales(5);

b0 = R_m/scales(1) + R_f/scales(3);
b1 = L_m/scales(2) + L_f/scales(4) + R_m/scales(1) * ...
    R_f/scales(3) * C_f/scales(5);
b2 = R_m/scales(1) * L_f/scales(4) * C_f/scales(5) + ...
    R_f/scales(3) * L_m/scales(2) * C_f/scales(5);
b3 = L_m/scales(2) * L_f/scales(4) * C_f/scales(5);

Y = zeros(1,length(w));
for i = 1:length(w)
Y(i) = abs(2/3 * (w(i)^2 *a2 + w(i) * a1 +1)/(w(i)^3 * b3 + ...
    w(i)^2 * b2 + w(i) * b1 + b0));
end
Y = Y+1e-5;
```

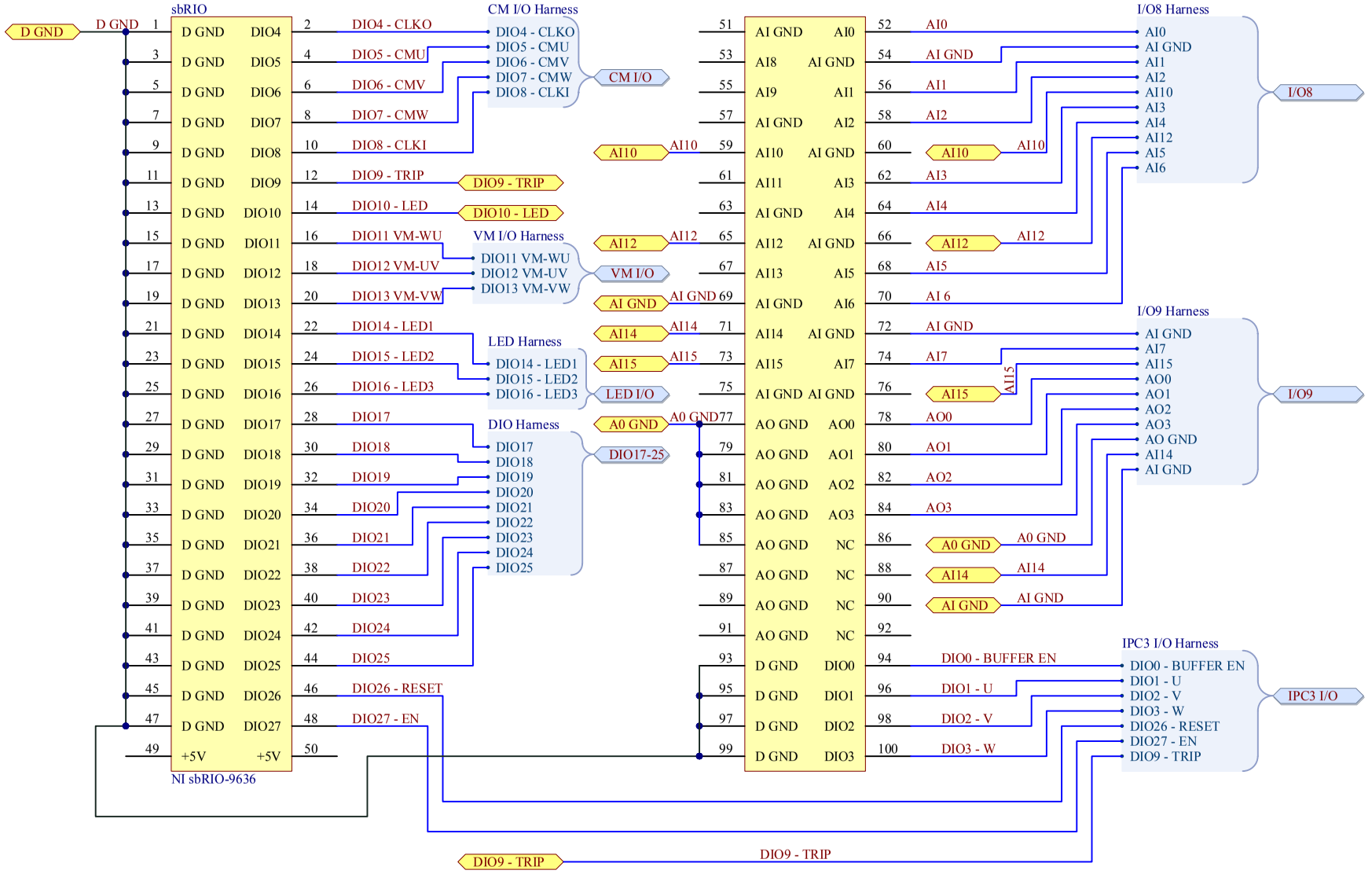
Appendix **B**

IB2 Circuit diagrams

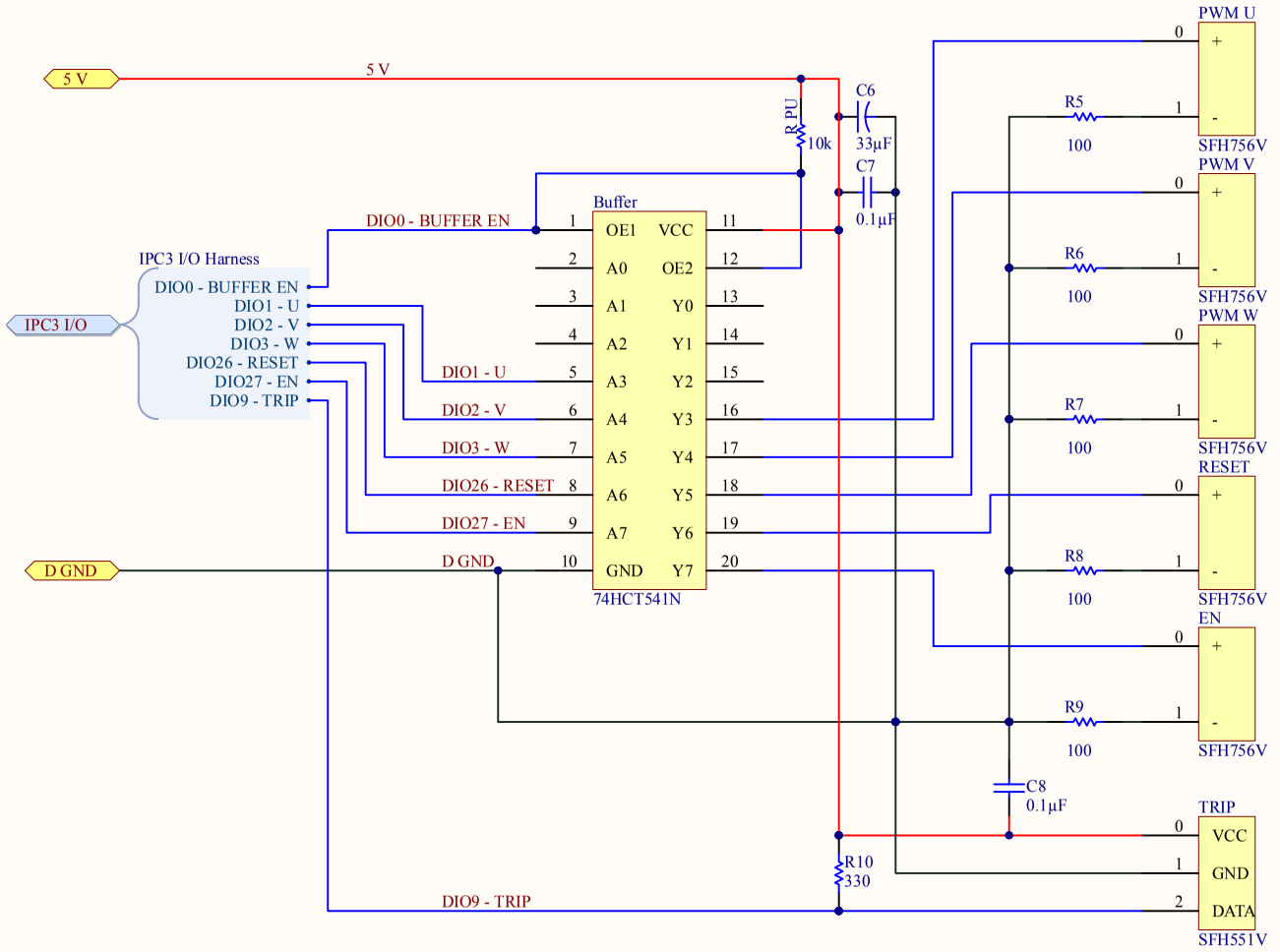
In the following pages the circuit diagrams for the IB2 are found. The circuits are drawn in Altium Designer and the actual files are found on the enclosed CD. The circuits are coupled using sheet symbols, meaning that the first circuit diagram ”IB2 Connections” is a schematic of how they are connected, while the rest are the actual circuit diagrams. The circuit diagrams are found in the following order:



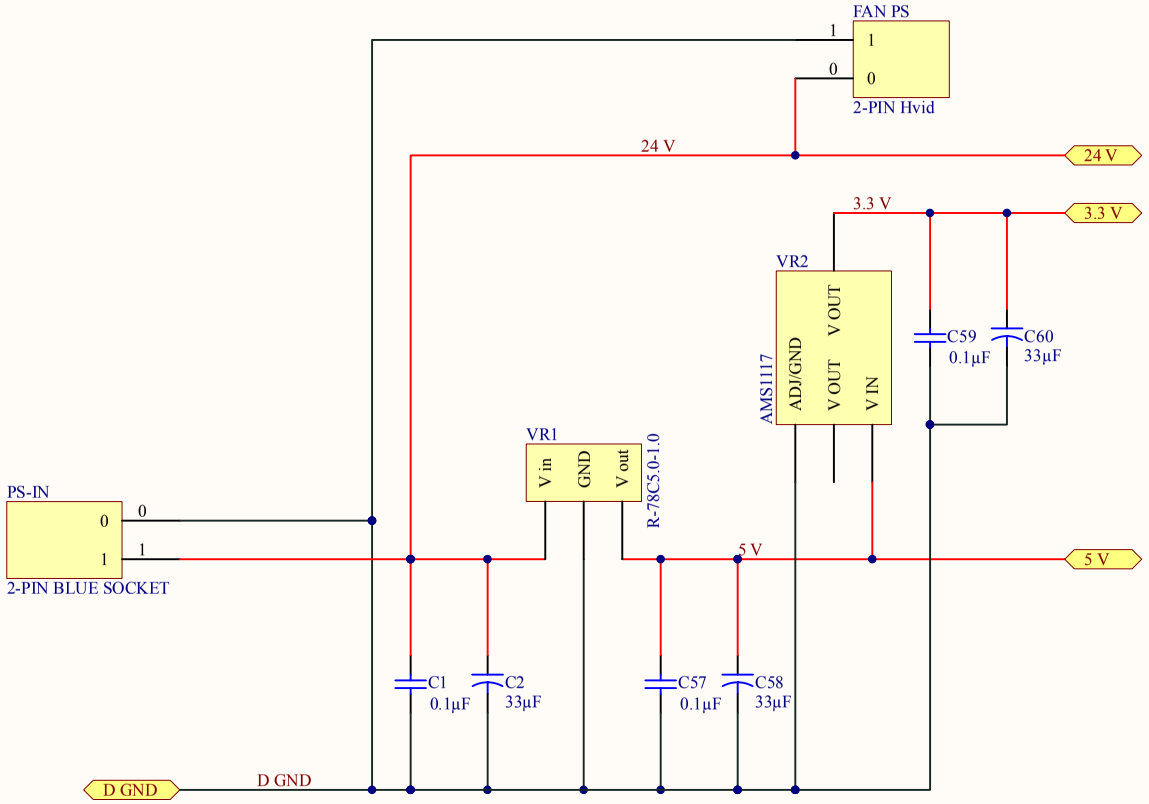
Title		
Size A4	Number	Revision
Date: 18-03-2015		Sheet of
File: C:\Users\..\IB2.0.SchDoc		Drawn By:



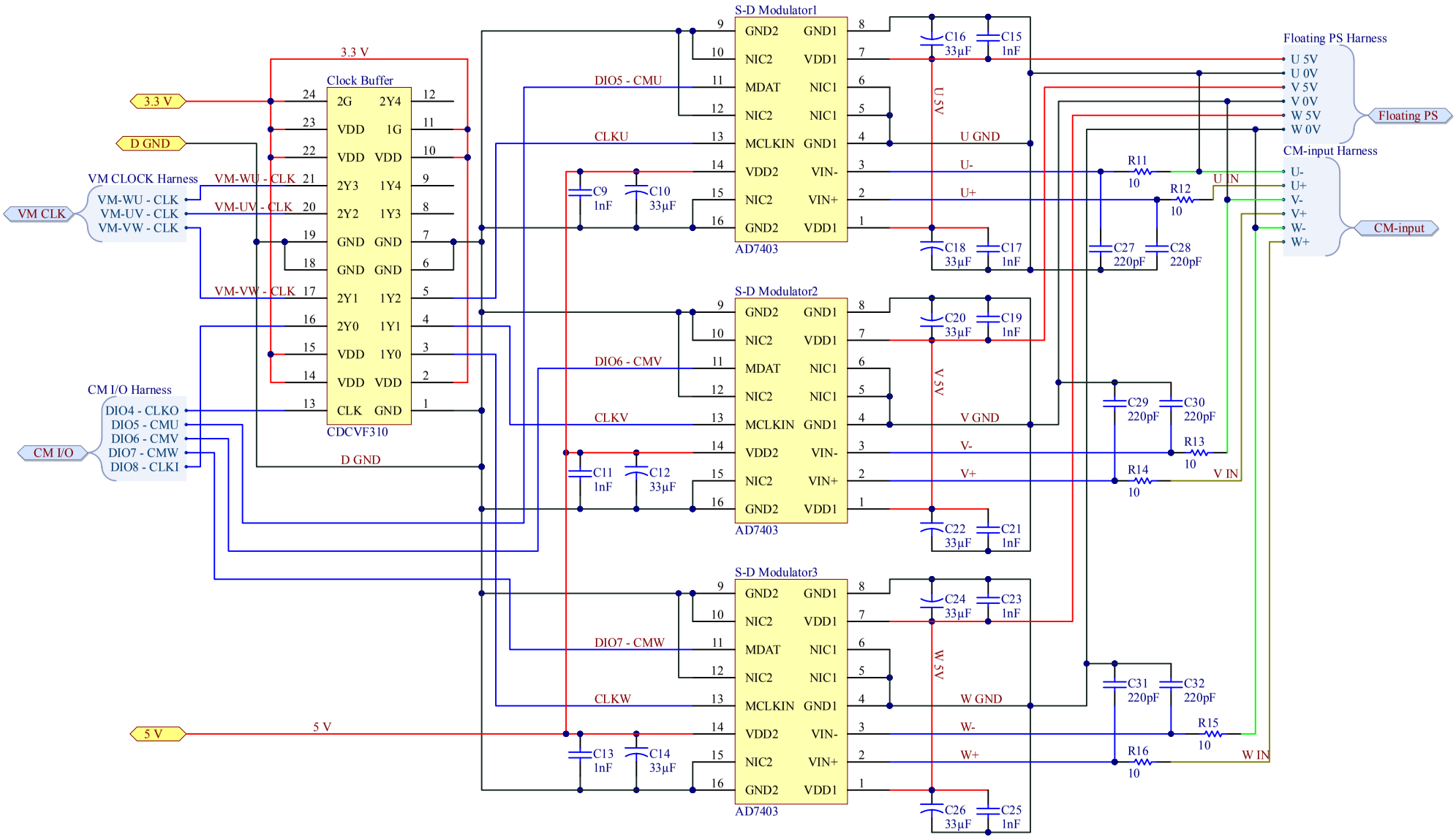
Title		
Size A4	Number	Revision
Date: 16-03-2015		Sheet of
File: C:\Users\..\sbRIO-IO.SchDoc		Drawn By:

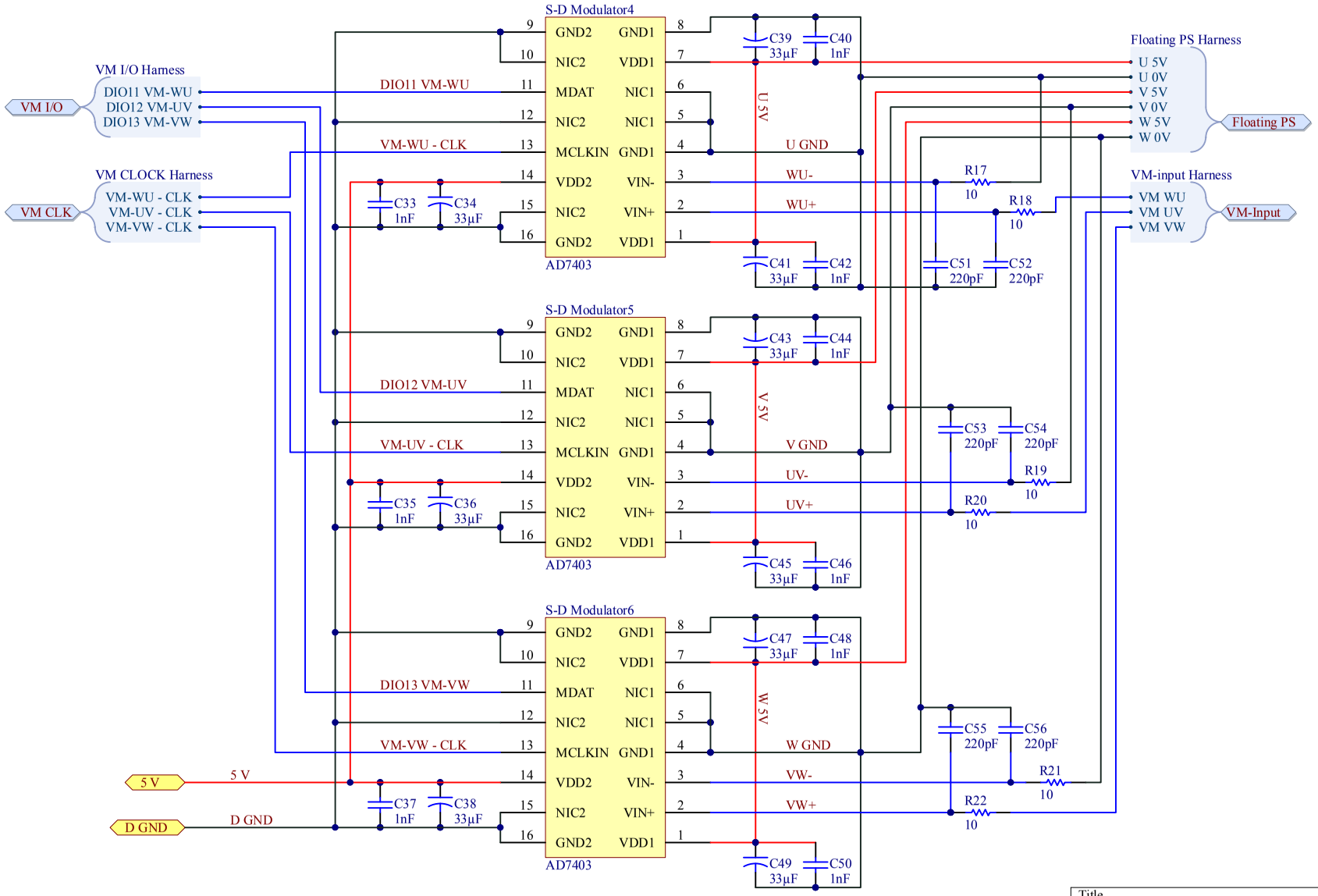


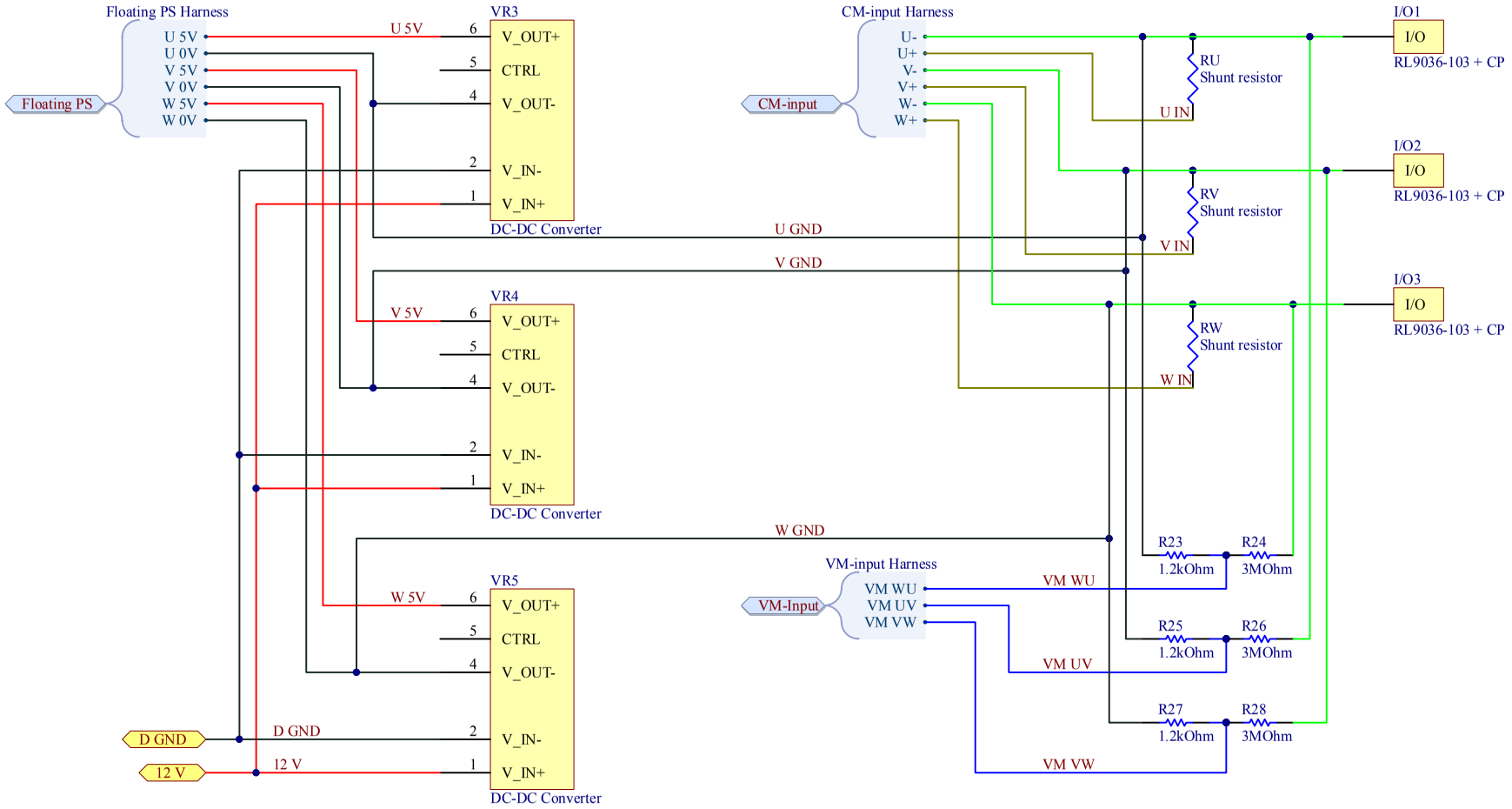
Title		
Size A4	Number	Revision
Date: 31-03-2015		Sheet of
File: C:\Users\..\IPC3-IO.SchDoc		Drawn By:



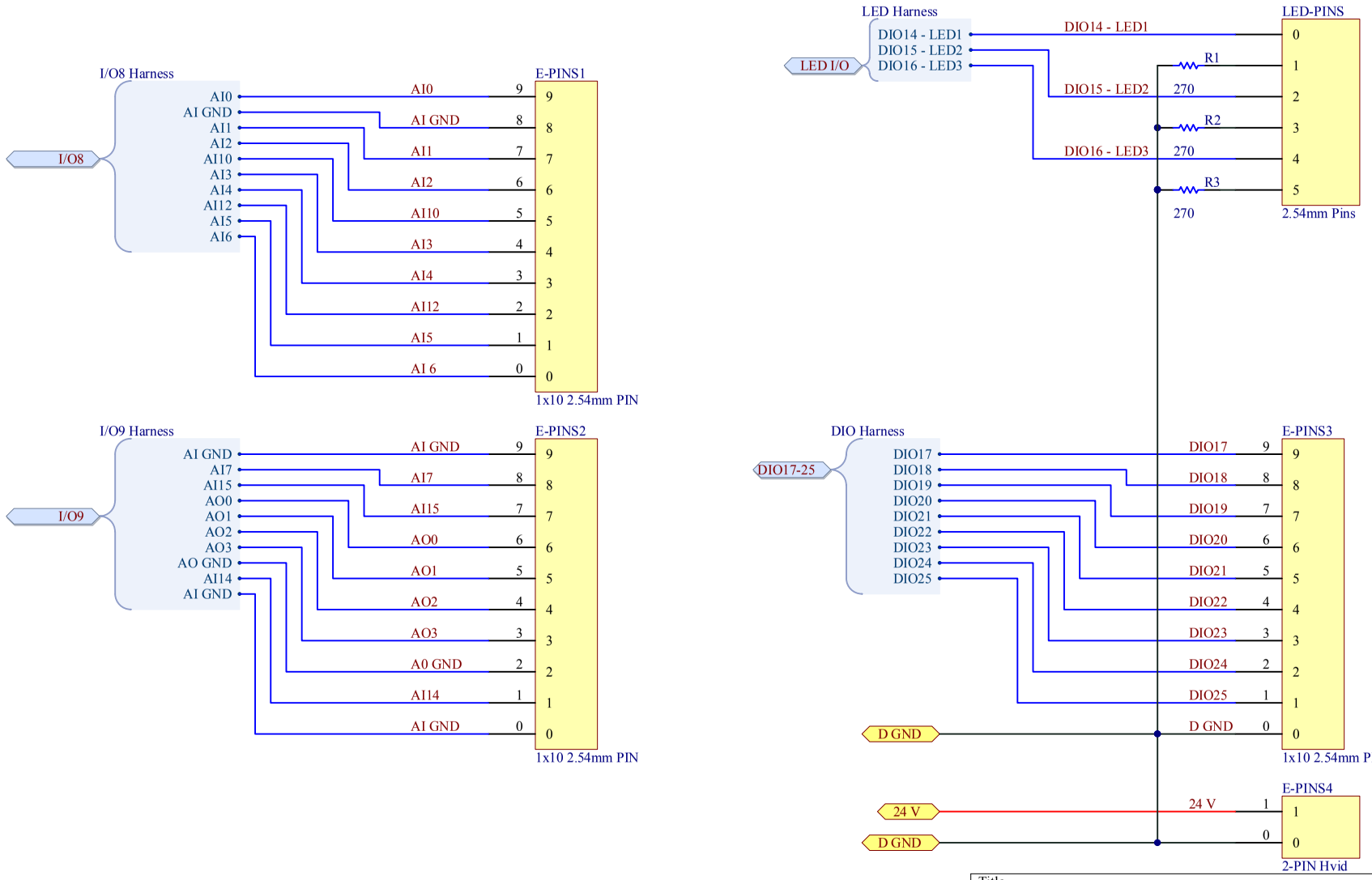
Title		
Size A4	Number	Revision
Date: 31-03-2015		Sheet of
File: C:\Users\..\Power_supply.SchDoc		Drawn By:


3.3 V
D GND
VM CLK
VM CLOCK Harness
VM-WU - CLK
VM-UV - CLK
VM-VW - CLK
CM I/O
CM I/O Harness
DIO4 - CLKO
DIO5 - CMU
DIO6 - CMV
DIO7 - CMW
DIO8 - CLKI
5 V
Clock Buffer
CDCVF310
S-D Modulator1
S-D Modulator2
S-D Modulator3
AD7403
GND2
GND1
NIC2
VDD1
MDAT
NIC1
MCLKIN
VDD2
VIN-
VIN+
U 5V
V 5V
W 5V
U GND
V GND
W GND
U-
U+
V-
V+
W-
W+
U IN
V IN
W IN
Floating PS Harness
U 5V
U 0V
V 5V
V 0V
W 5V
W 0V
Floating PS
CM-input Harness
CM-input
Title
Size
A4
Number
Revision
Date:
19-02-2015
Sheet of
File:
C:\Users\...\CM.SchDoc
Drawn By:


S-D Modulator4
S-D Modulator5
S-D Modulator6
AD7403
GND2
GND1
NIC2
VDD1
MDAT
NIC1
MCLKIN
VIN-
VIN+
VDD2
VM I/O Harness
DIO11 VM-WU
DIO12 VM-UV
DIO13 VM-VW
VM I/O
VM CLOCK Harness
VM-WU - CLK
VM-UV - CLK
VM-VW - CLK
VM CLK
5 V
D GND
U 5V
U GND
V 5V
V GND
W 5V
W GND
WU-
WU+
UV-
UV+
VW-
VW+
Floating PS Harness
U 5V
U 0V
V 5V
V 0V
W 5V
W 0V
Floating PS
VM-input Harness
VM WU
VM UV
VM VW
VM-Input
C33 1nF
C34 33µF
C35 1nF
C36 33µF
C37 1nF
C38 33µF
C39 33µF
C40 1nF
C41 33µF
C42 1nF
C43 33µF
C44 1nF
C45 33µF
C46 1nF
C47 33µF
C48 1nF
C49 33µF
C50 1nF
C51 220pF
C52 220pF
C53 220pF
C54 220pF
C55 220pF
C56 220pF
R17 10
R18 10
R19 10
R20 10
R21 10
R22 10
Title
Size A4
Number
Revision
Date: 19-02-2015
Sheet of
File: C:\Users\..\VM.SchDoc
Drawn By:



Title		
Size A4	Number	Revision
Date: 19-02-2015		Sheet of
File: C:\Users\..\Inverter-IO.SchDoc		Drawn By:


I/O8 Harness
I/O8
AI0
AI GND
AI1
AI2
AI10
AI3
AI4
AI12
AI5
AI6
E-PINS1
AI0 9
AI GND 8
AI1 7
AI2 6
AI10 5
AI3 4
AI4 3
AI12 2
AI5 1
AI 6 0
1x10 2.54mm PIN
I/O9 Harness
I/O9
AI GND
AI7
AI15
AO0
AO1
AO2
AO3
AO GND
AI14
AI GND
E-PINS2
AI GND 9
AI7 8
AI15 7
AO0 6
AO1 5
AO2 4
AO3 3
A0 GND 2
AI14 1
AI GND 0
1x10 2.54mm PIN
LED Harness
LED I/O
DIO14 - LED1
DIO15 - LED2
DIO16 - LED3
DIO14 - LED1
R1
DIO15 - LED2
270
R2
DIO16 - LED3
270
R3
270
LED-PINS
0
1
2
3
4
5
2.54mm Pins
DIO Harness
DIO17-25
DIO17
DIO18
DIO19
DIO20
DIO21
DIO22
DIO23
DIO24
DIO25
D GND
E-PINS3
DIO17 9
DIO18 8
DIO19 7
DIO20 6
DIO21 5
DIO22 4
DIO23 3
DIO24 2
DIO25 1
D GND 0
1x10 2.54mm PIN
24 V
D GND
E-PINS4
24 V 1
0
2-PIN Hvid
Title
Size
A4
Number
Revision
Date: 16-03-2015
Sheet of
File: C:\Users\..\Expansion.SchDoc
Drawn By:

Appendix C

LabVIEW Program

C.1 FPGA VIs

Figure C.1 Main FPGA.

Figure C.2 Data Collection, FIFO communication and Data transfer.

Figure C.3 Sequence generator, reciprocal function, and pseudo random number generator.

Figure C.4 Safety, Trigger for data collection, and ΔΣ modulator clock generator.

Figure C.5 Complete Sinc3 Filter, Integrators, Decimator, Differentiator, and Output Scaling .

Figure C.6 A single integrator and a single differentiator.

Figure C.7Â´User defined variables to and from registers.

Figure C.8 LCD interface to RT, Contacts, Initialize LCD, and Write LCD case 1 of 5.

Figure C.9 LCD Write case 2-5 .

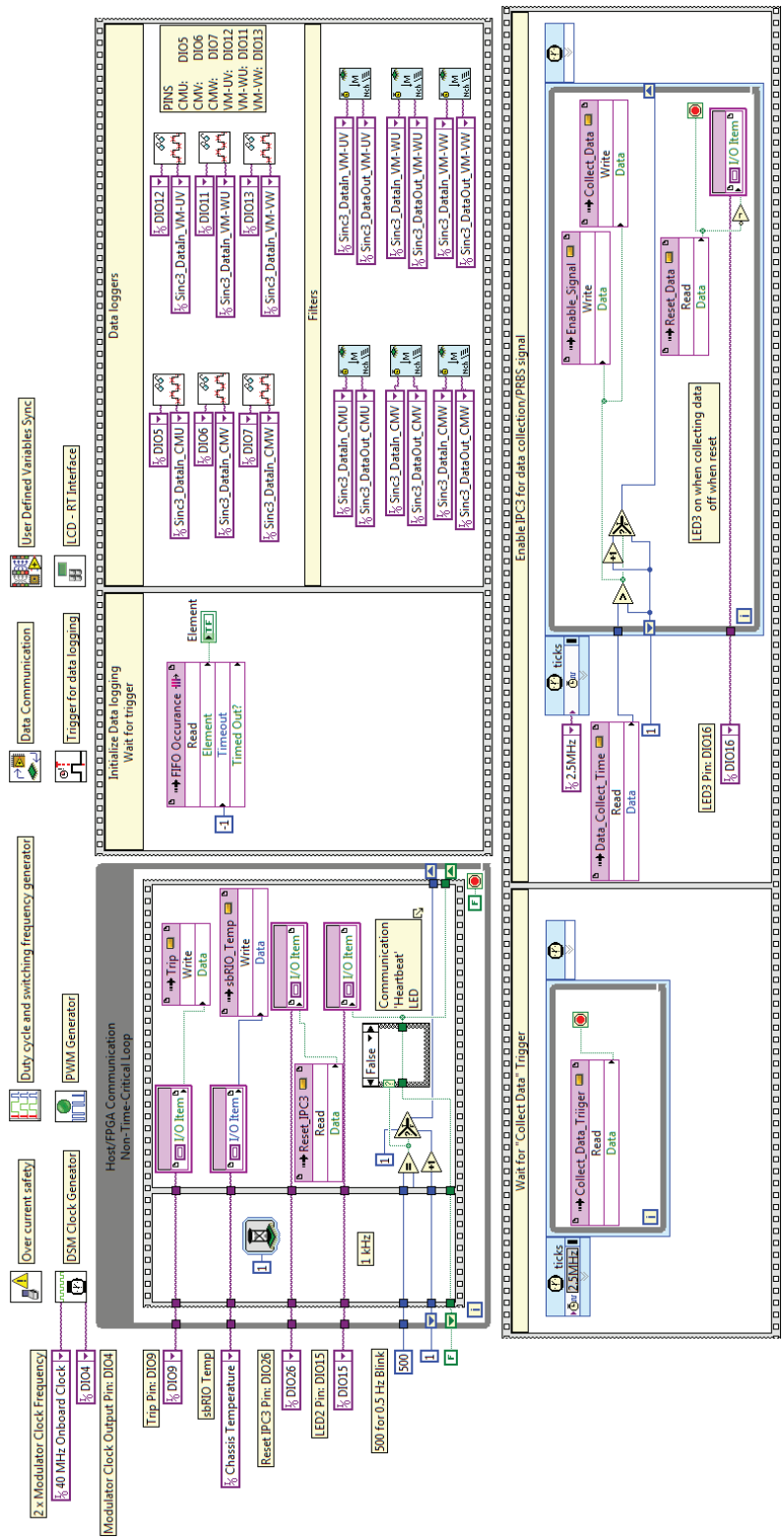


Figure C.1: Main FPGA Program.

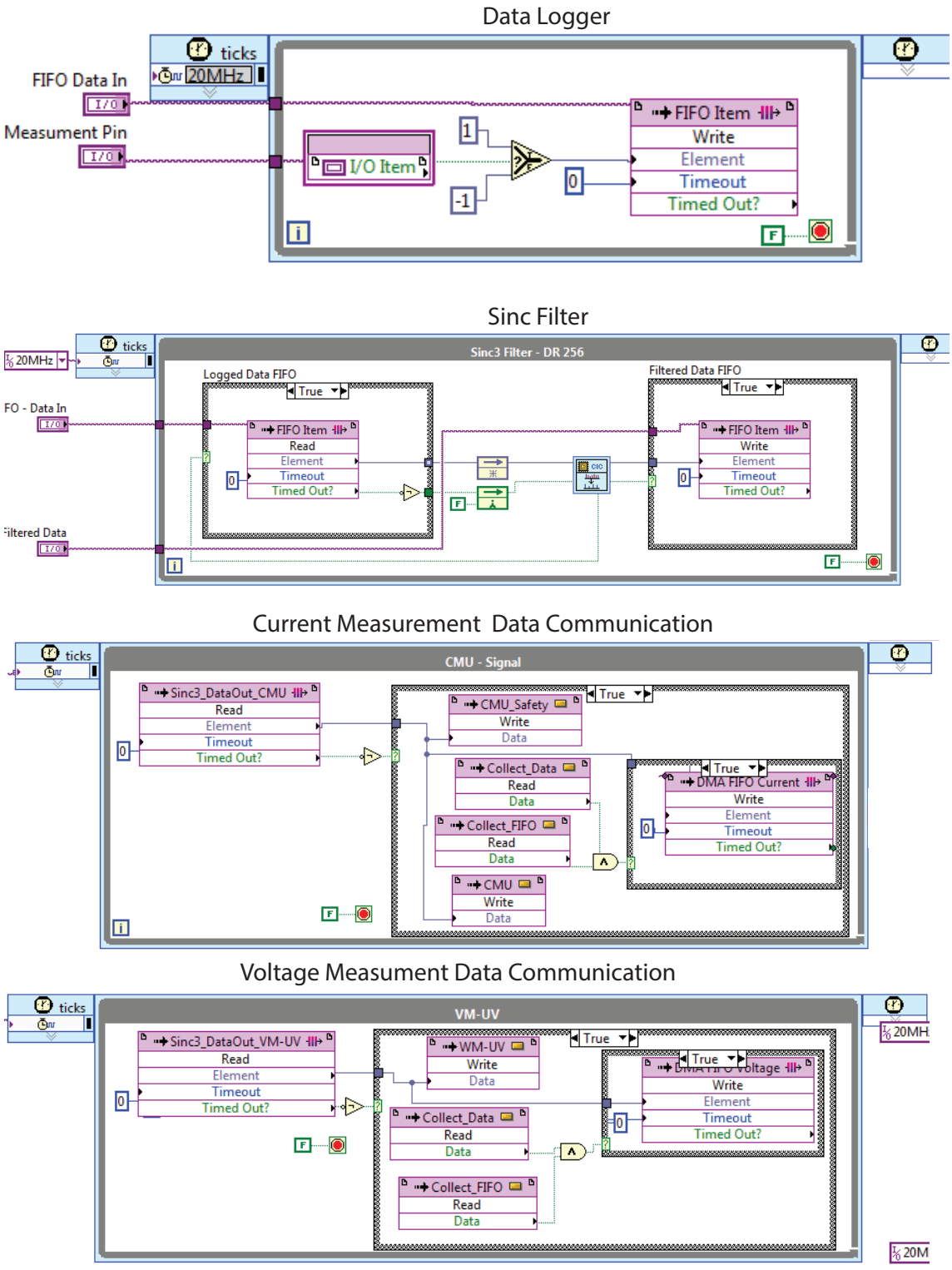


Figure C.2: Data Collection, FIFO communication and Data transfer.

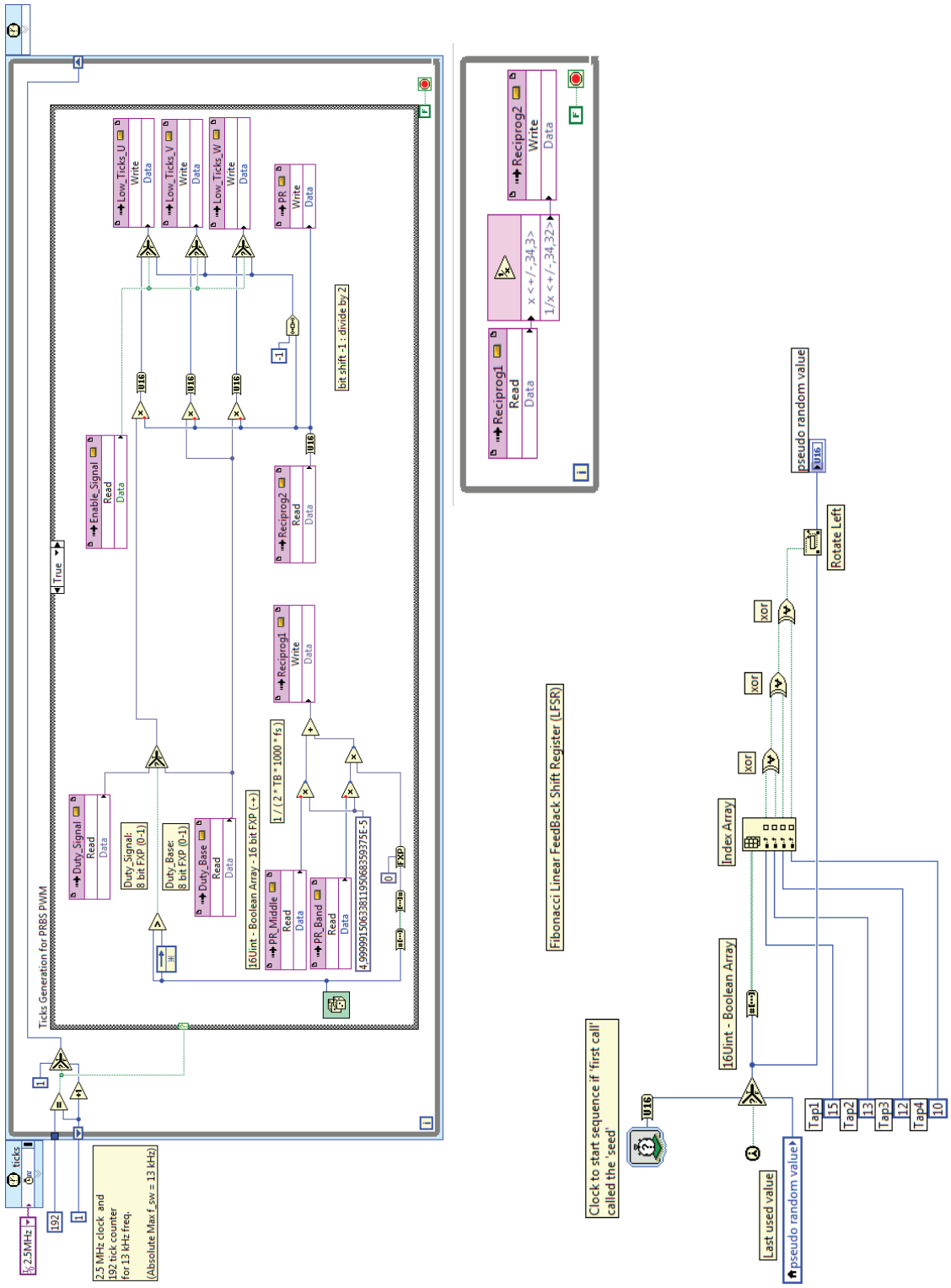


Figure C.3: Sequence generator, reciprocal function, and pseudo random number generator.

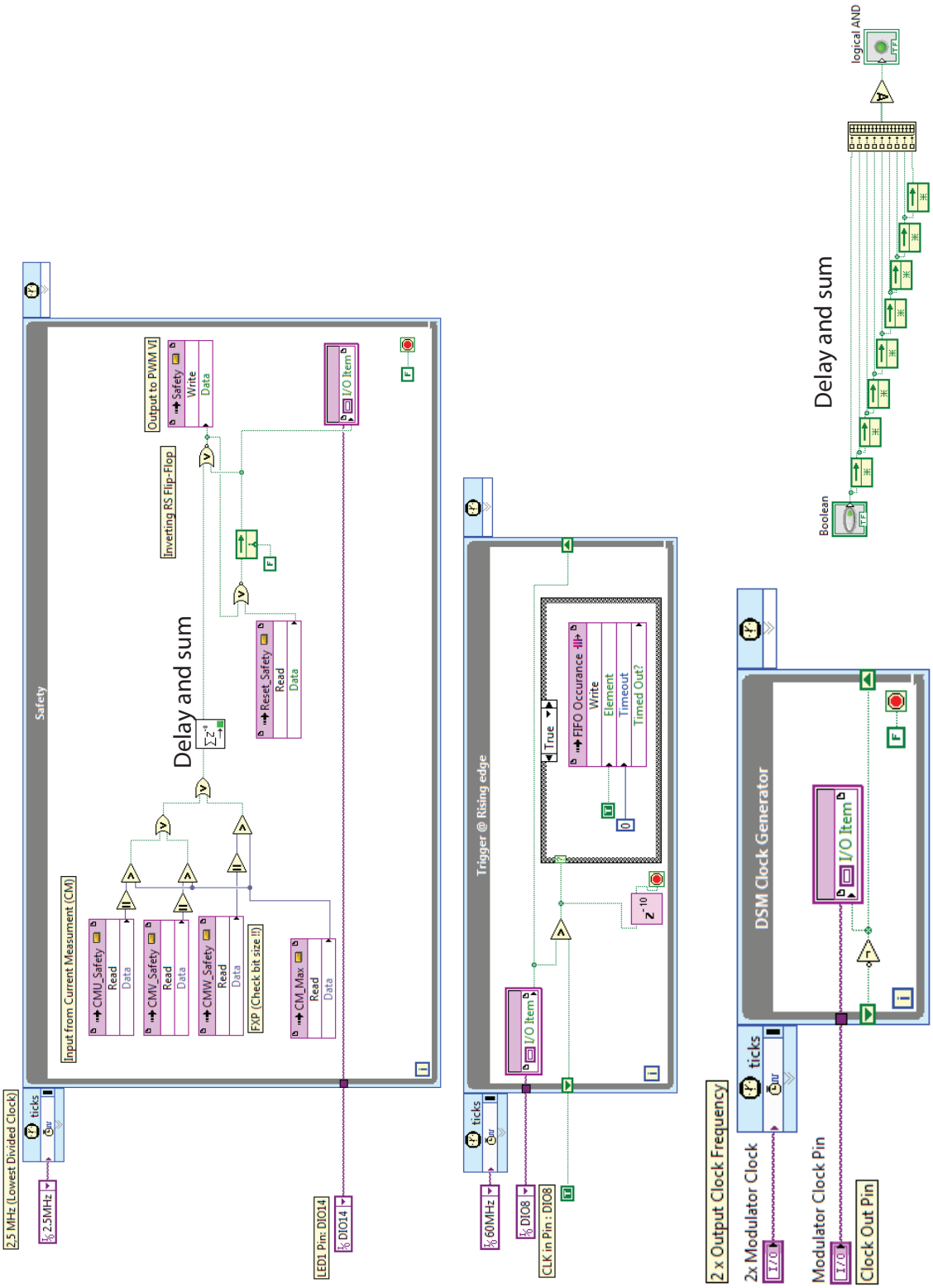


Figure C.4: Safety, Trigger for data collection, and ΔΣ modulator clock generator.

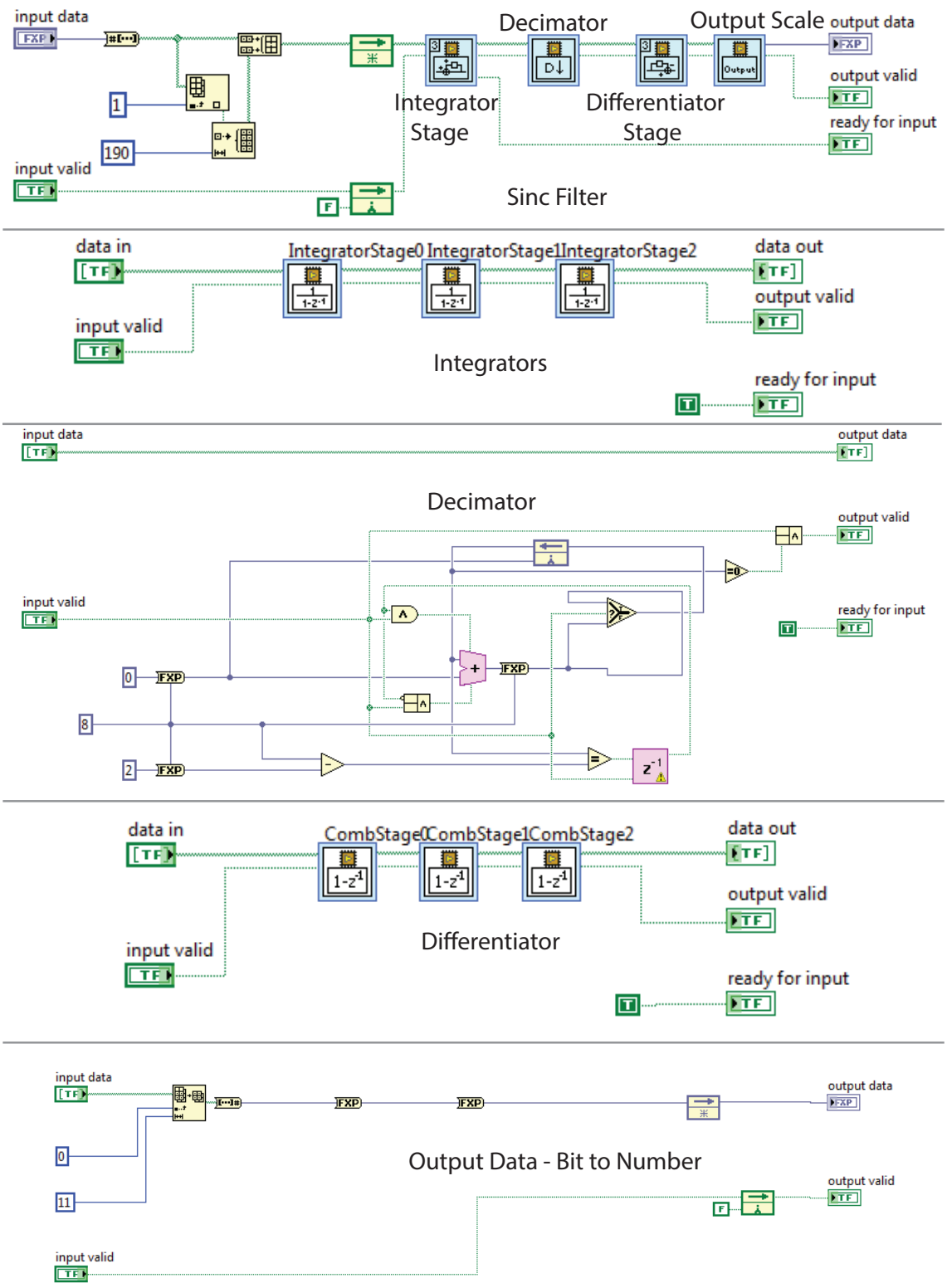


Figure C.5: Complete Sinc3 Filter, Integrators, Decimator, Differentiator, and Output Scaling.

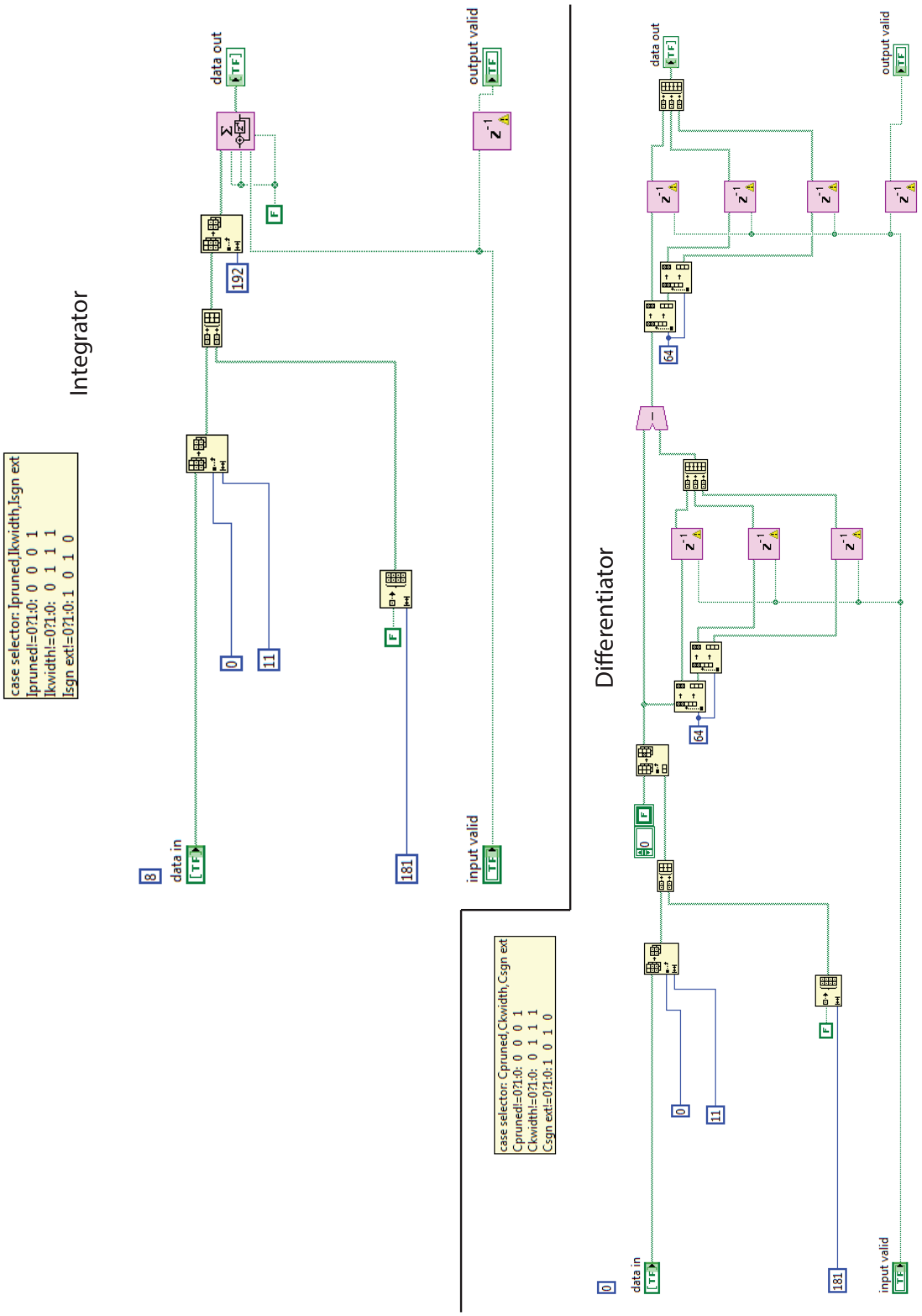


Figure C.6: A single integrator and a single differentiator.

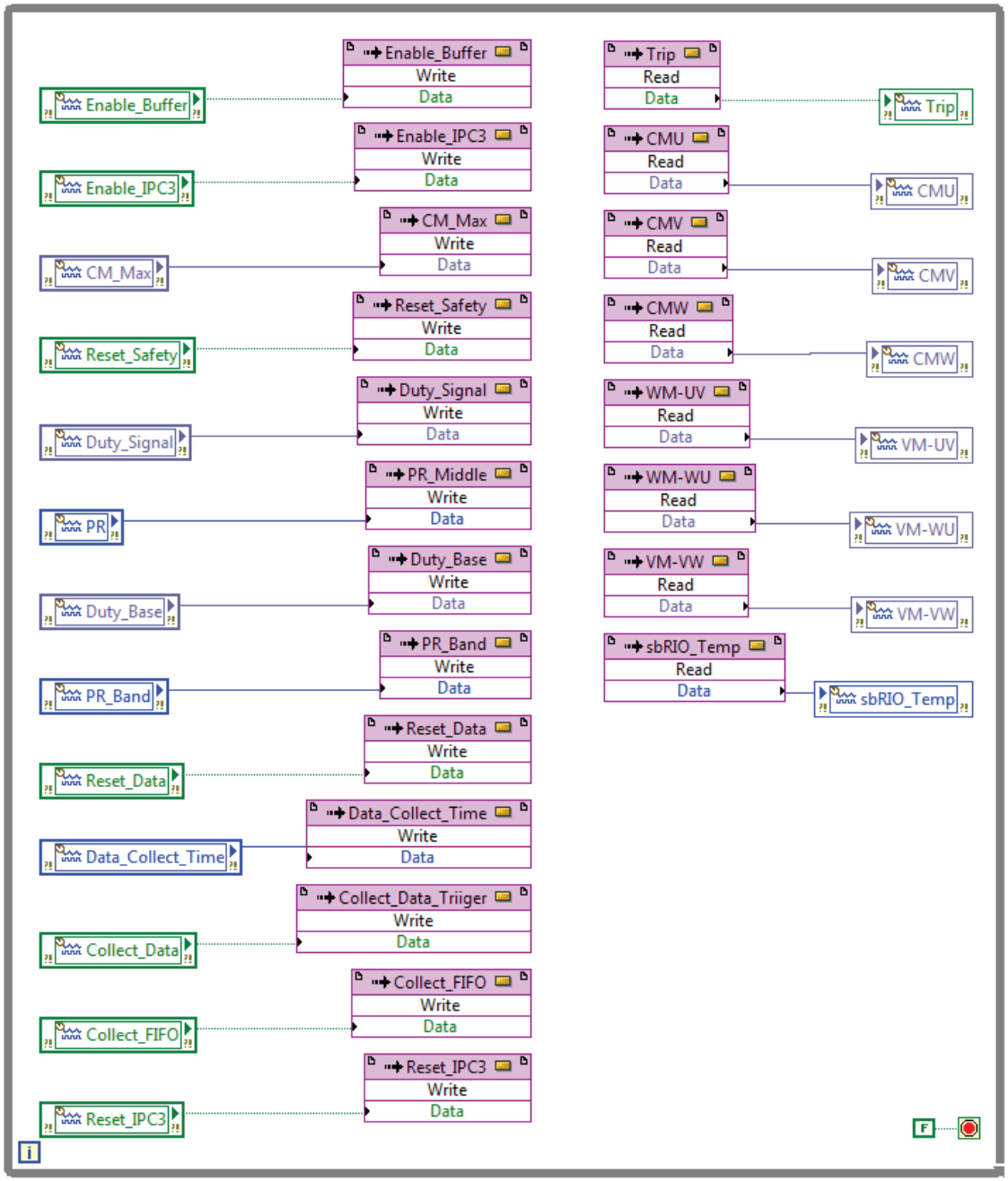
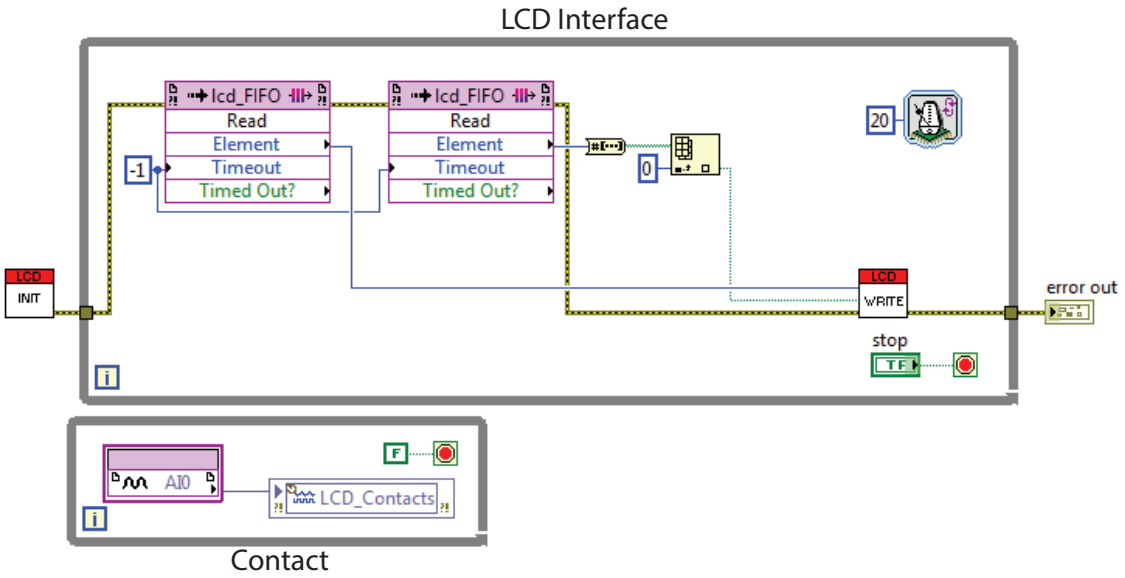
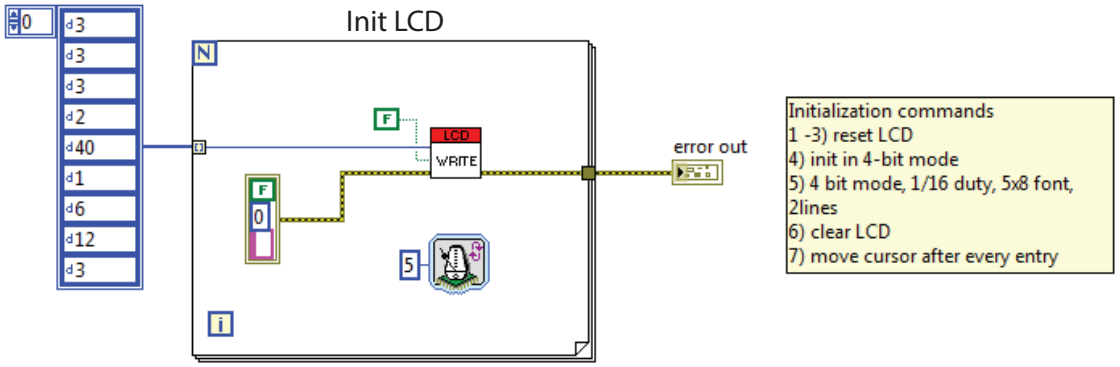


Figure C.7: User defined variables to and from registers.





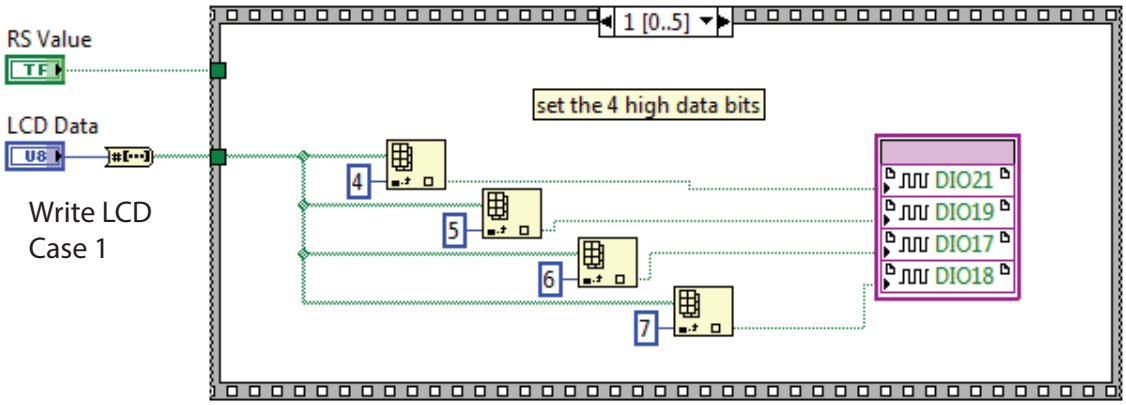


Figure C.8: LCD interface to RT, Contacts, Initialize LCD, and Write LCD case 1 of 5 figure C.8..

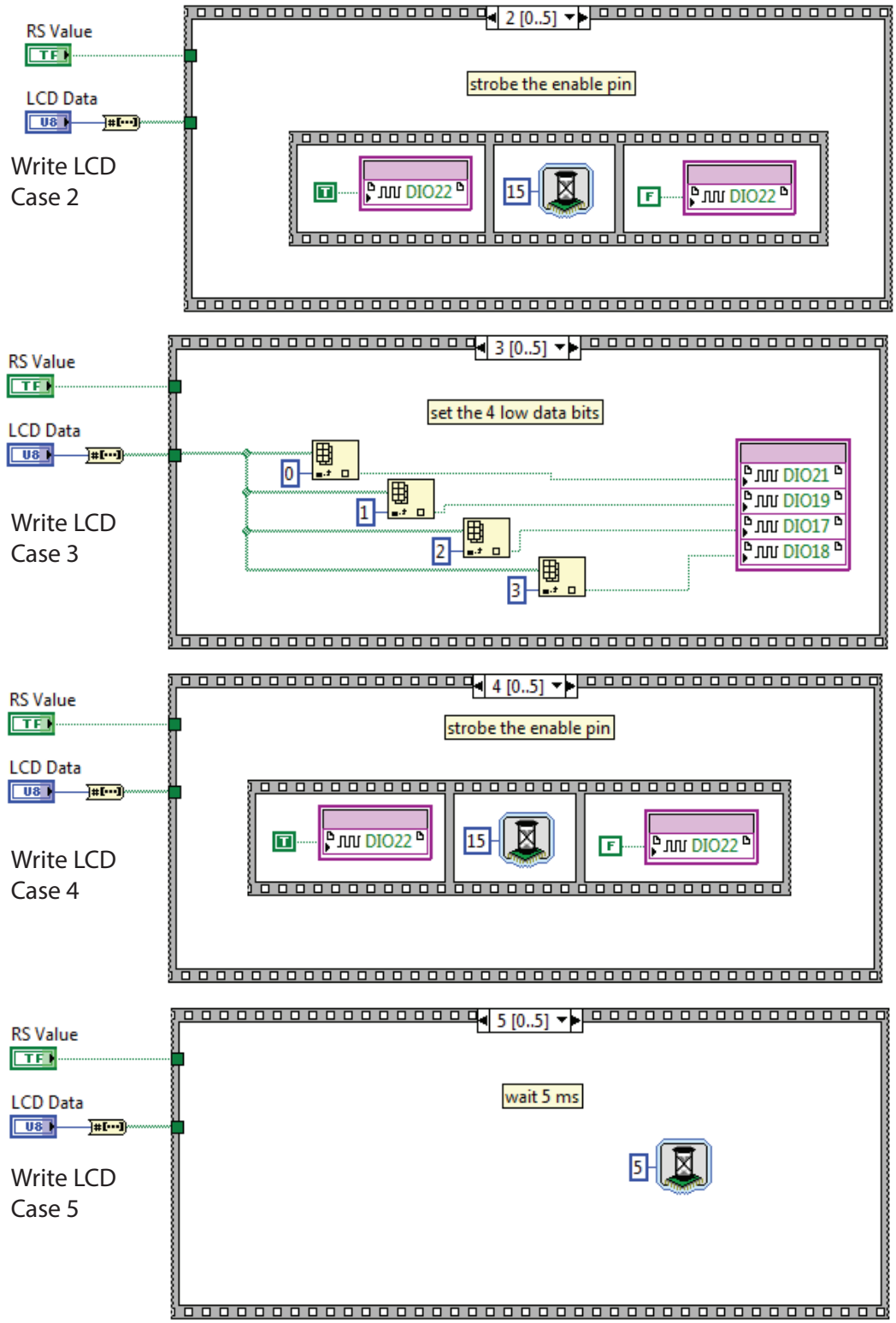


Figure C.9: LCD Write case 2-5.

C.2 RT VIs

Figure C.10 Main RT.

Figure C.11 PSO Algorithm, Configuration cluster, LCD Contacts.

Figure C.12 PSO Main Motor VI, Settings Menu, Results Menu.

Figure C.13 PSO Main Filter VI, and PSD function.

Figure C.13 FIFO collect.

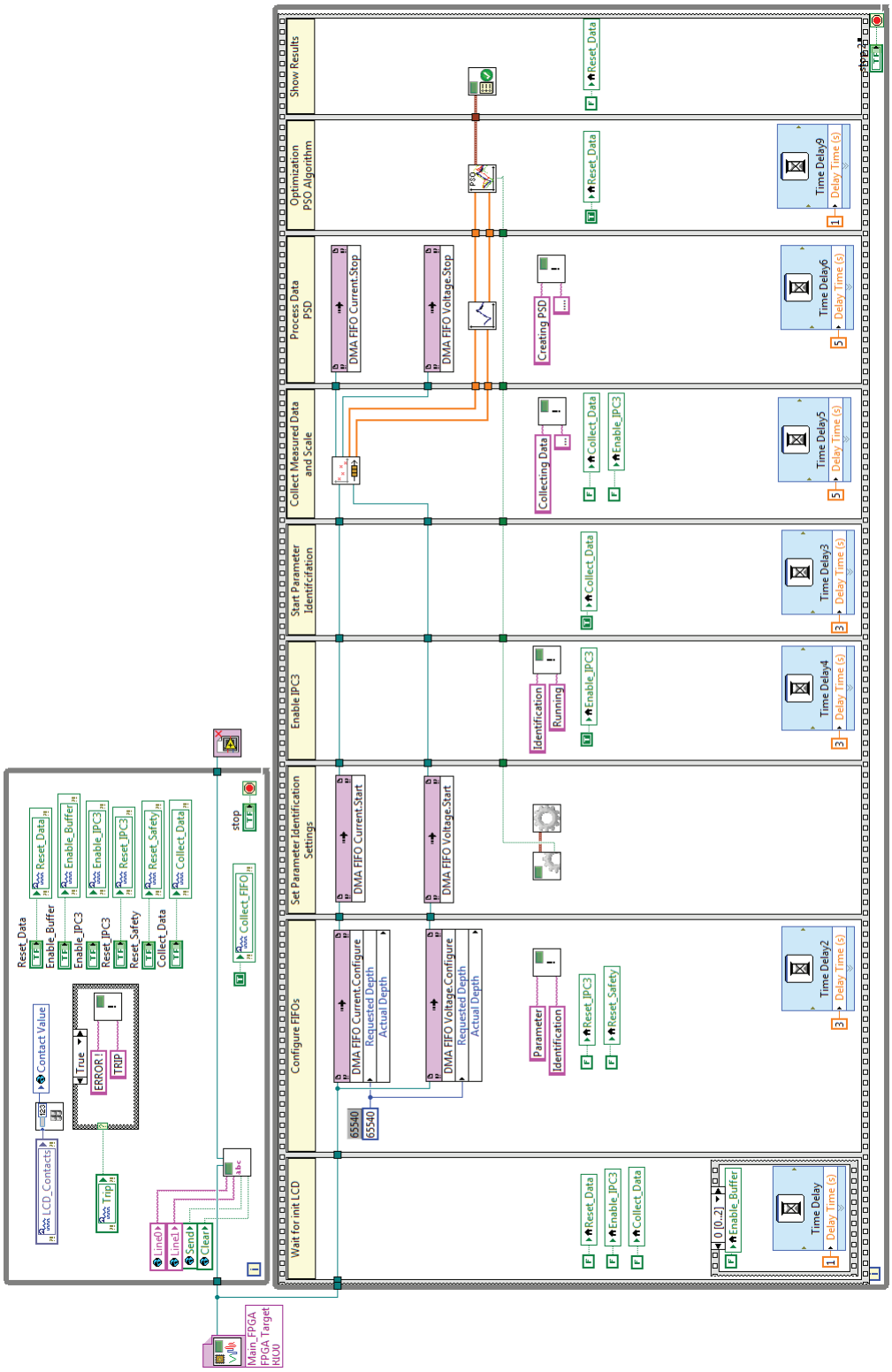


Figure C.10: Main RT.

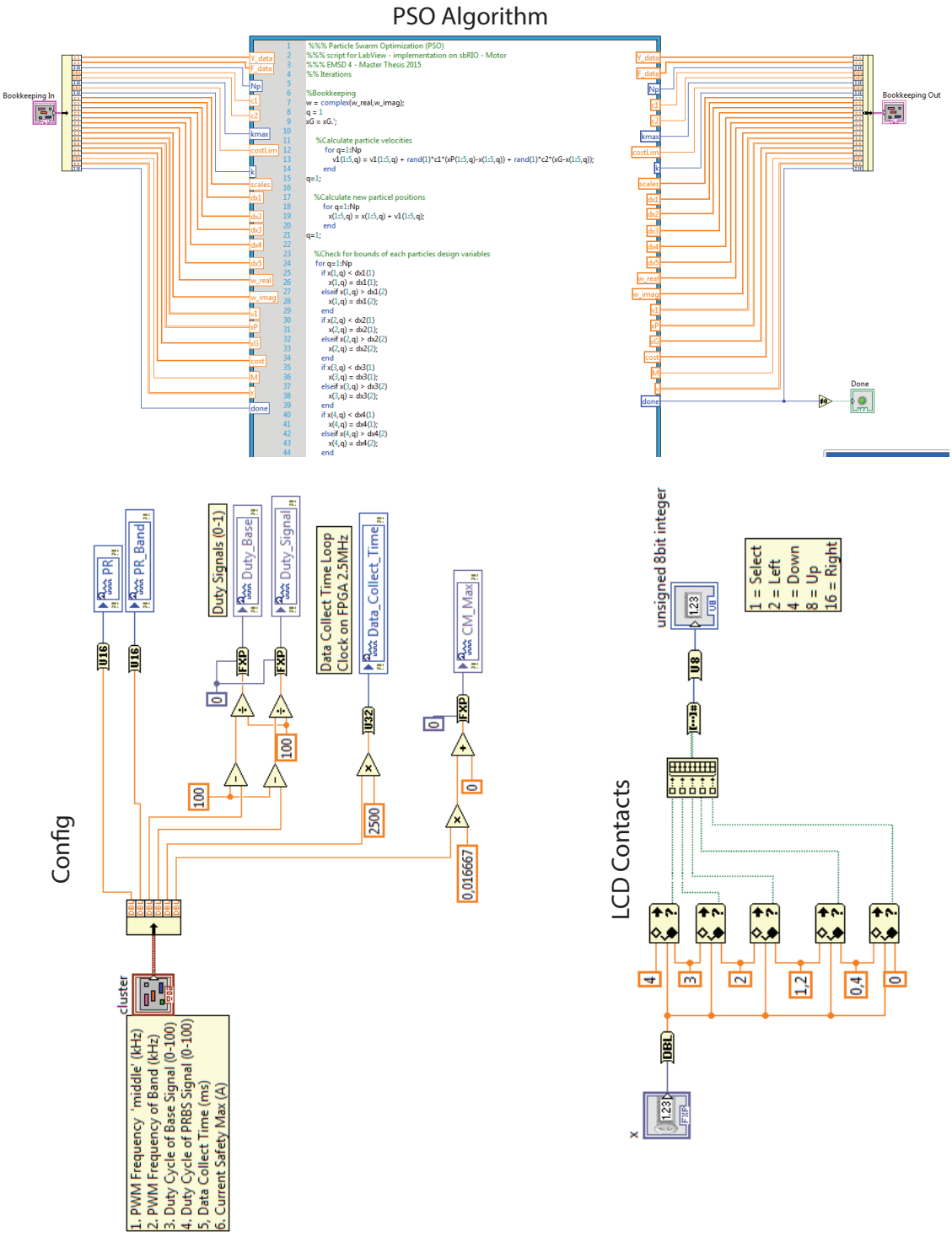


Figure C.11: PSO Algorithm, Configuration cluster, LCD Contacts.

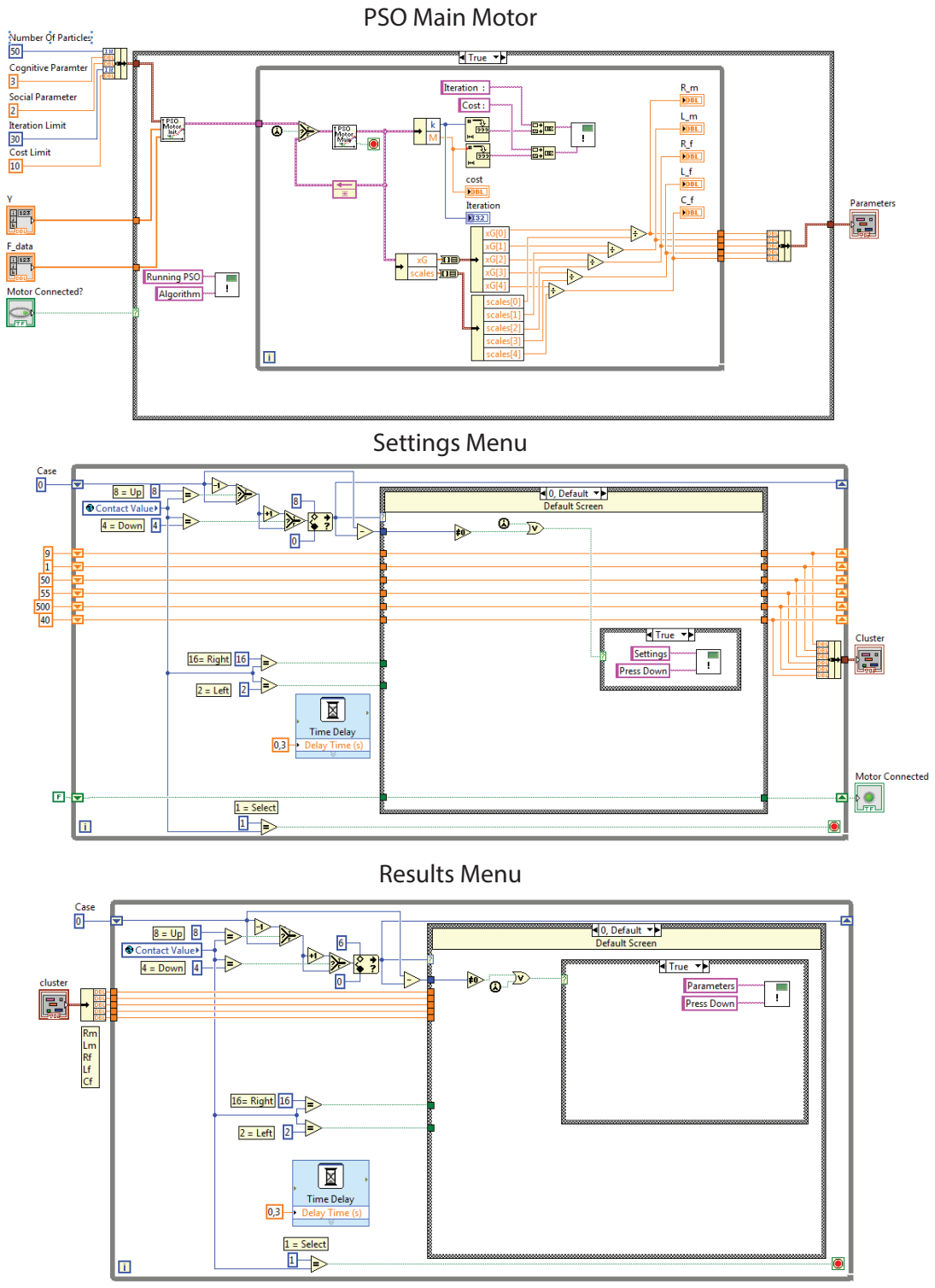


Figure C.12: PSO Main VI, Settings Menu, Results Menu.

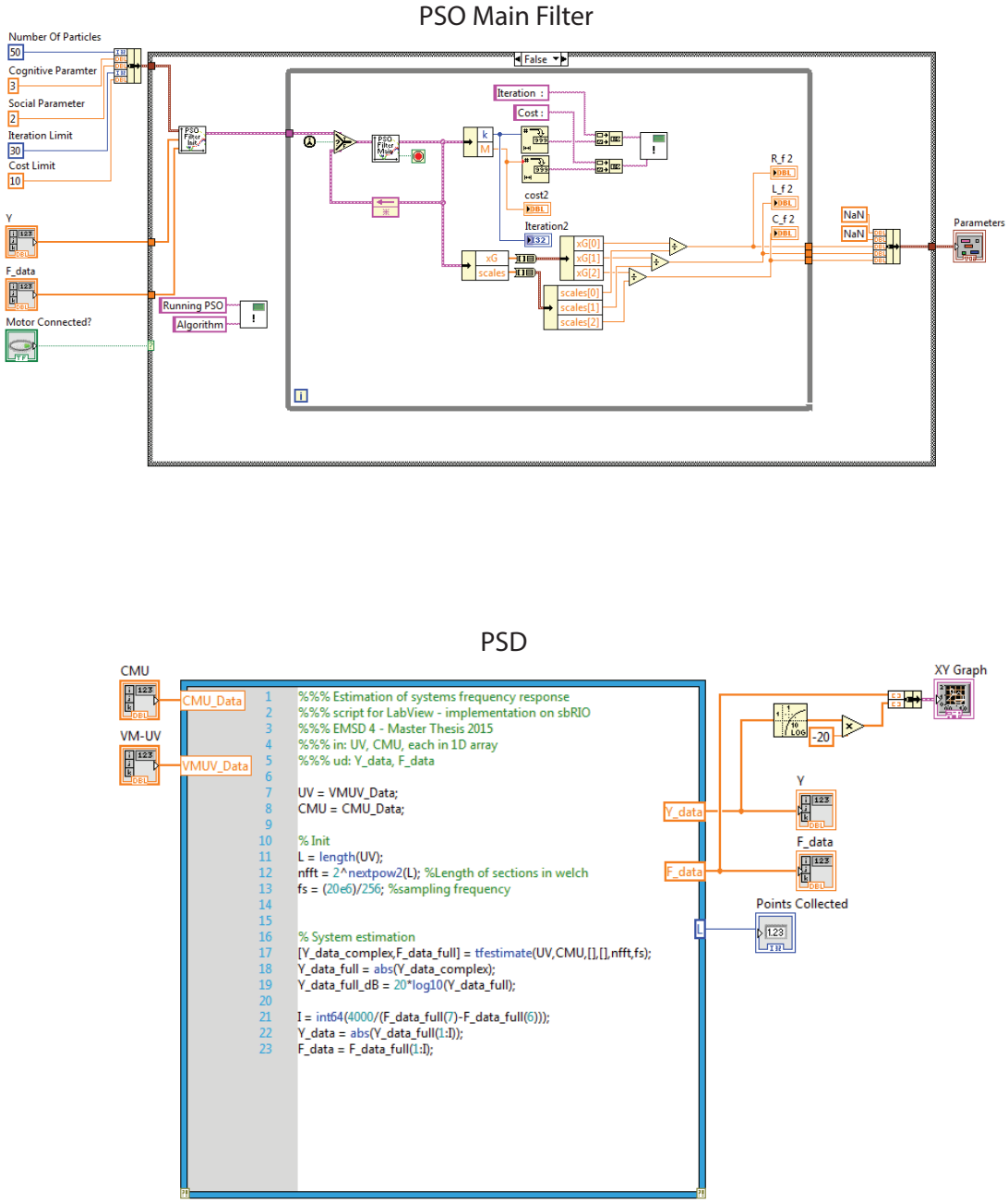


Figure C.13: PSO Main Filter VI, and PSD function.

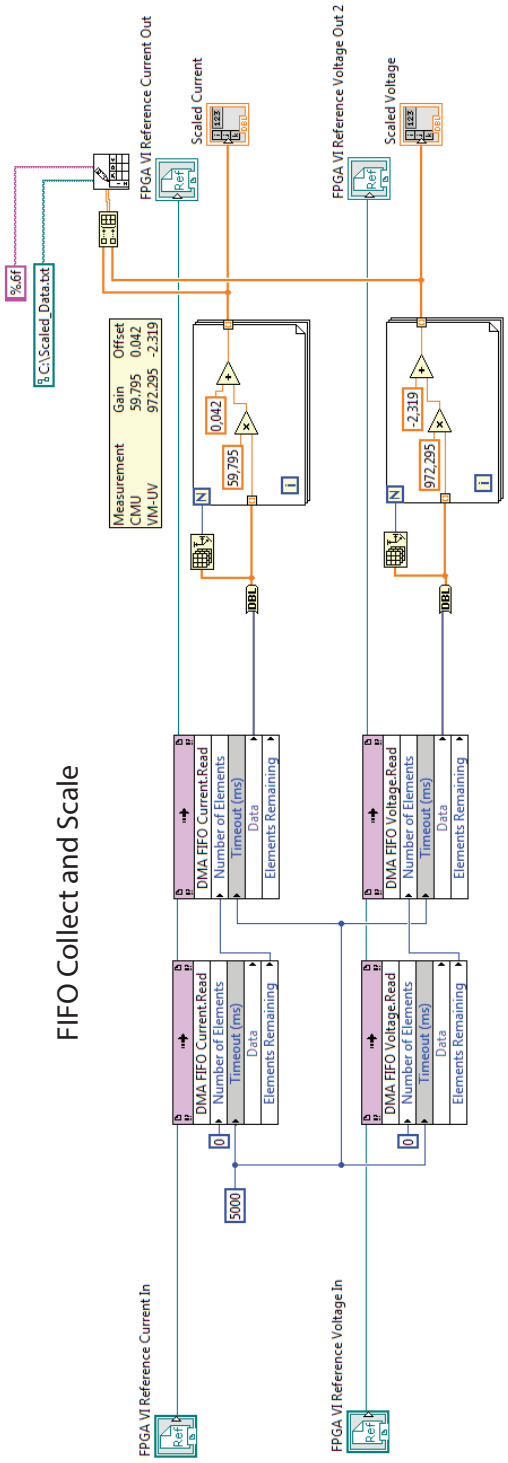


Figure C.14: FIFO collect.

Appendix **D**

Contents of the enclosed CD