Master's thesis

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

Kristel Storvik Einarsen

FPGA-based real-time modeling of active front end (AFE)

Master's thesis in Electric Power Engineering
Supervisor: Roy Nilsen

June 2023

NTNU
Norwegian University of
Science and Technology

Kristel Storvik Einarsen

FPGA-based real-time modeling of active front end (AFE)

Master's thesis in Electric Power Engineering
Supervisor: Roy Nilsen
June 2023

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

NTNU
Norwegian University of
Science and Technology

Preface and acknowledgements

This report is the final work of my Master's degree at "Electric Power engineering" at the department of Electrical Energy at NTNU, completed in spring 2023. The topic was provided by the resource group, PESC, and it immediately captured my interest, and I was fully prepared to tackle the assigned task. The background information provided by the study program was limited, making the process of acquainting myself with a completely new topic quite challenging.

During the specialization project in autumn 2022, I conducted a comprehensive literature review on the subject of a virtual synchronous machine. However, as I transitioned to my Master's thesis, a new supervisor took on the role, providing me with guidance. This resulted in a change of research topic and the need to delve into an entirely new theme. Throughout the process, I received assistance from my new supervisor, Professor Roy Nilsen.

I would like to express my sincere gratitude to my supervisor for his patience and expertise. Without him, this thesis would not have been possible. I would also like to extend my thanks to Thomas S. Haugan, who developed the necessary programming for the FPGA, and to Professor Roy Nilsen for sharing his valuable Simulink models. Lastly, I would like to express my special appreciation to the members of the Corner Office, with whom I shared both joyous and challenging moments. Through laughter and motivational words, they have been a source of support throughout my journey.

Abstract

In the emerging field of renewable energy, new control methods are being considered and have been researched for quite some time. Many studies can be found on the converter control techniques. This thesis is considering the AFE connected to a LCL-filter. For the development on these converters, the traditional PC computer simulations of the models take quite some time, and therefore the emulated real time simulation have been used for research and design. By employing FPGA implementation, tasks can be executed in parallel manner, enabling real-time simulations. This thesis is a part of a project at the Department of Electrical Energy at NTNU to develop a real-time FPGA-based emulator of a AFE by using the PESC control platform developed by NTNU.

During the development of the Emulated Real-Time Simulator, a simulation process is employed using Matlab, Simulink to analyze and evaluate how a AFE connected to a LCL-filter behaves. First, a simplified model where the AFE is represented by an ideal current source is studied. Followed by a more realistic model, where the converter is represented the power electronic insulated gate bipolar resistors (IGBT) switches, the result is a more realistic and complex model. To further enhance the PESC control platform, an emulated model of the LCL filter is built using the mathematical model derived from the equations. The emulated model developed aims to capture the dynamic reactions of the LCL filter accurately. Further the emulated mathematical model is discretezied. These discretized equations serve as the foundation for constructing the FPGA-based emulator, where the FPGA logic is developed in the Xilinx system generator for DSP add on in the simulink library. Finally, the emulated model and a project file is configured with the PESC control platform, where the emulated model is ready to be tested. Simulation of different scenarios is done within the PESC control platform. The effect of the Active dampening is discussed, where some oscillation can be observed. The frequency of the oscillations is measured to 819.1 Hz. This frequency is compared to the resonance frequency of the LCL filter. The resonance frequency is calculated to 992.4 Hz. Despite the slight discrepancy in frequencies, it can be concluded that the frequencies of the harmonics resulting from the absence of active dampening closely resemble the resonance frequency of the LCL filter. The AFF plot of the exact model and the PESC simulations are compared. The difference of the model is the fundamental frequency, it can be seen that the PESC platform operates with 78.8 Hz and the exact model operated with 60 Hz. f_f and ζ need to be scaled down to get the same K_p and T_i.

Further analyses with different grid forming control scheme need to be done for the model in the PESC platform, ass well as a model using PWM controller.

Sammendrag

I den voksende sektoren for fornybar energi har nye innovative kontrollmetoder blitt forsket på i lang tid. Denne rapporten tar for seg aktiv likeretter koblet til et kraftnett via et LCL-filter. I utviklingen av kraftelektroniske omformere tar tradisjonelle PC-simuleringer lang tid, og derfor har emulerte sanntidssimuleringer blitt brukt til forskning og design. Ved å bruke FPGA-implementering kan oppgaver utføres parallelt, noe som muliggjør sanntidssimuleringer.

Denne avhandlingen er en del av et prosjekt ved institutt for elektrisk energi ved NTNU for å utvikle en sanntids-FPGA-basert emulator av en aktiv likeretter ved bruk av PESC-kontrollplattformen utviklet av NTNU.

Under utviklingen av den emulerte sanntidssimulatoren er simuleringer i MATLAB Simulink gjennomført for å analysere og evaluere oppførselen til den aktive likeretteren. Først simuleres en forenklet modell der aktiv likeretteren representeres av en ideell strømkilde studert. Deretter følger en mer realistisk modell der kraftomformeren representeres av IGBT-brytere, noe som resulterer i en mer realistisk og kompleks modell.

For å forbedre PESC-kontrollplattformen ytterligere, bygges en emulert modell av LCL-filteret ved hjelp en matematisk modell som er utledet fra dynamikken til LCL-filteret. Målet med den utviklede emulerte modellen er å fange opp de dynamiske egenskapene til LCL-filteret.

Den emulerte matematiske modellen blir deretter diskretisert. Disse ligningene fungerer som grunnlaget for konstruksjonen av FPGA-basert emulator, der FPGA-logikken utvikles i Simulink-biblioteket Xilinx System Generator for DSP.

Til slutt blir den emulerte modellen integrert og kompilert med PESC-kontrollplattformen, der den emulerte modellen er klar til å bli testet. Simulering av ulike scenarier utføres innenfor PESC-kontrollplattformen.

Effekten av aktiv demping blir diskutert, der man kan observere hvordan dette påvirker resonansen mellom kraftnett og den aktive likeretteren. Frekvensen til oscillasjonene måles til 819,1 Hz. Denne frekvensen blir sammenlignet med resonansfrekvensen til LCL-filteret, som beregnes til 992,4 Hz. Til tross for den lille forskjellen i frekvensene, kan det konkluderes med at frekvensene til harmoniske svingninger som oppstår på grunn av fravær av aktiv demping i stor grad ligner på resonansfrekvensen til LCL-filteret.

Table of Contents

List of Tables

List of Figures

Abbreviations

- **AC** Alternate Current
- **AFE** Active Front End
- **CPU** Central Processing Unit
- **DC** Direct Current
- **DSP** Digital Signal Processing
- **ERTS** emulated real-time simulations
- **FPGA** Field Programmable Gate Array
- **IGBT** Insulated Gate Bipolar resistors
- **PEC** Power electronics converters
- **PLL** Phase Locked Loop
- **pu** Per Unit
- **PWM** Pulse Width Modulation
- **VSM** Virtual Synchronous Machine

Chapter 1

Introduction

1.1 Background and motivation

Due to the increase in renewable energy sources in the power grid, the natural control of the frequency and voltage coming from the big synchronous machines usually exist in the coal plants and fossile sources. The distributed power generation system, DPG, is required to actively take part in the power regulation of the grid [1]. [2] and [3] study different grid forming schemes and its performance on the power grid. This thesis focuses on the Active Front End (AFE) connected to an LCL-filter, which serves as a vital component in such control systems. Traditional PC-based simulations for converter development are time-consuming, necessitating the use of emulated real-time simulations (ERTS) for research and design purposes. By employing FPGA implementation, tasks can be executed concurrently, enabling real-time simulations. This methodology offers significant advantages, as it allows for the design and evaluation of control algorithms during the design process itself. The present thesis is part of a project undertaken at the Department of Electrical Energy at NTNU, aimed at developing a real-time FPGA-based emulator of an AFE using the PESC control platform developed by NTNU. Drawing upon the ERTS approach presented in [4], which focuses on a Permanent Magnet Synchronous Generator (PMSG), this thesis aims to emulate an AFE converter within a similar framework.

1.2 Scope of Work

During the development of the Emulated Real-Time Simulator, a comprehensive simulation process is employed using Matlab, Simulink to analyze and evaluate how an AFE connected to a LCL-filter behaves. First, a simplified model where the AFE is represented by an ideal current source is studied. Next, a more realistic model is studied, where the ideal current sources are replaced by a realistic model of the converter with power electronic insulated gate bipolar resistors (IGBT) switches, accounting for the influence of switching frequencies and associated ripple effects. The purpose of this simplified model is to gain a better understanding of the system's dynamics and functionality. The control method is a hysteresis controller using a grid-following scheme. The exact AFE model is simulated to study the behavior of the phase-locked loop (PLL). Further, a mathematical model of the LCL-filter connected to the AFE is developed, which aims to capture the dynamic reactions of the LCL-filter accurately. The emulated model is calculating the inverter current, grid current and capacitor voltage with mathematical equations. Further, the FPGA is developed with the Xilinx system generator add on blocks in Simulink. For implementation of the IP cores in the FPGA, the design is exported from System Generator to Vivado, where the IP core design can be synthesized into HDL code to configure with the FPGA-module. However, this is out of the scope for this thesis, as the IP cores will only be developed and tested in System Generator.

Finally, The emulated model and a project file are configured with the PESC control platform, where the emulated model is ready to be tested. Simulation of different scenarios is done within the PESC control platform. These scenarios include simulating the starting state with the breaker open, simulating the starting state with the breaker closed with and without active damping, and simulating the effect of a load increase. The simulations help evaluate the performance of the control platform and analyze its response to different operating conditions. Here different functions are tested, including the Phase locked loop (PLL) and the effect of active dampening.

1.3 Structure

This thesis report is divided into six main chapters. Their contents are as follows:

Chapter 2 presents the background information for this thesis. Starting with describing AFE and its usage. Further, describing the two main control methods, grid-following scheme and grid-forming scheme. The mathematical model of the LCL-filter connected to the AFE is derived. Additionally, fundamental concepts such as Euler's method, the Park-Clark transformation, the logic of the emulator, the PLL, active damping, and the hysteresis controller are explained.

Chapter 3 briefly describes the PESC control platform developed at NTNU. Including The implementation of the FPGA firmware.

Chapter 4 provides the simulations of a simple Simulink model, and a more exact model, as well as the simulation of the PESC controller.

Chapter 5 discusses the results provided from the Simulink simulaions, including a discussion on the PLL technique and active dampening.

Chapter 6 presents the conclusion and proposed further work.

Chapter 2

Background Theory

2.1 Active front-end with LCL-filter

Active front end (AFE), typically as Two level 3-phase converter, offers the advantage of controlling both the DC-link voltage and the power factor. The converter is built from three half-bridges, which consists of two insulated gate bipolar resistors (IGBTs) in each half bridge. This topology is dominant for the three-phase applications for line voltage below 1000 V. The AFE can function as a converter, from AC to DC, or as an inverter, converting DC to AC. The two main methods to control the power electronic IGBT switches, are either using a pulse-width modulation (PWM), which is out of the scope of the present thesis, or a hysteresis controller. In this master thesis, the AFE is implemented using the hysteresis controller with a grid-following control scheme.

However, due to the switching frequency of the IGBTs, the output signal produced may contain high-order harmonics. To mitigate this issue, an LCL-filter is integrated and connected to the AFE to reduce these undesired harmonics. The LCL-filter filters and adsorbs the harmonics to achieve the desired sinusoidal voltage waveform.

In the context of this thesis, the model consists of the AFE connected to an LCL-filter which is connected to the grid via a circuit break.

2.2 Control methods

Power systems generally require the balance and regulation of four variables: voltage, frequency, active power and reactive power. These variables play crucial roles in ensuring the stability and optimal functioning of the power grid.

In a conventional power system, the stability and control of voltage and frequency are achieved by interconnected synchronous machines. These machines, equipped with inherent inertia and kinetic energy, help regulate the system by responding to fluctuations and disturbances. By dynamically adjusting their output, they assist in maintaining a steady voltage and frequency within acceptable limits. The voltage and frequency in the power system have a direct link with the reactive and active power respectively. Thus, grid-following control enables power electric converters (PECs) to effectively manage their active and reactive powers based on the system's voltage and frequency conditions.

In converter-rich power systems, the reliance on PECs has grown significantly. Consequently, PECs are now being called upon to actively contribute to the stabilization of the power system. This shift is necessitated by the irregular nature of renewable energy sources and the decentralized nature of power generation. As renewable energy sources become more dominant, such as wind and solar, the power system looses its natural stabilisation from the synchronous generators. This change in the power system poses challenges in maintaining a stable grid with traditional control methods alone.

To address these emerging challenges, researchers and engineers are actively exploring and studying control schemes that can enhance the robustness and resilience of power systems. The focus is on developing strategies that enable power electronic converters to take on a more active role in frequency and voltage stabilization. These strategies, collectively known as grid-forming techniques, aim to provide PECs with the capability to establish and maintain the grid's fundamental parameters, such as voltage and frequency[2].

In this paper, the grid-forming method with the virtual synchronous machine scheme with hysteresis controller is applied. A PLL is also included in the model.

Grid-following converters operate as a current source, where the grid-following regulates the power injection and voltage by controlling the injected current (current from converter into grid)[3].

As the grid-following scheme controls power by adjusting the current, the grid-forming scheme regulates the voltage directly at its output terminal. There are several techniques to perform a grid-forming control scheme [2]. Examples of grid-forming methods that are discussed in literature are droop control, virtual synchronous machine, power synchronization control and distributed PLL-based. Where the fault ride through characteristics are studied in [2][3].

2.3 Mathematical model with LCL-filter

In order to implement the discrete model in the Field-Programmable Gate Array (FPGA), it is necessary to represent the LCL-filter, which is connected to the converter, using mathematical equations. The single-line diagram of the LCL-filter is depicted in Figure 2.1.

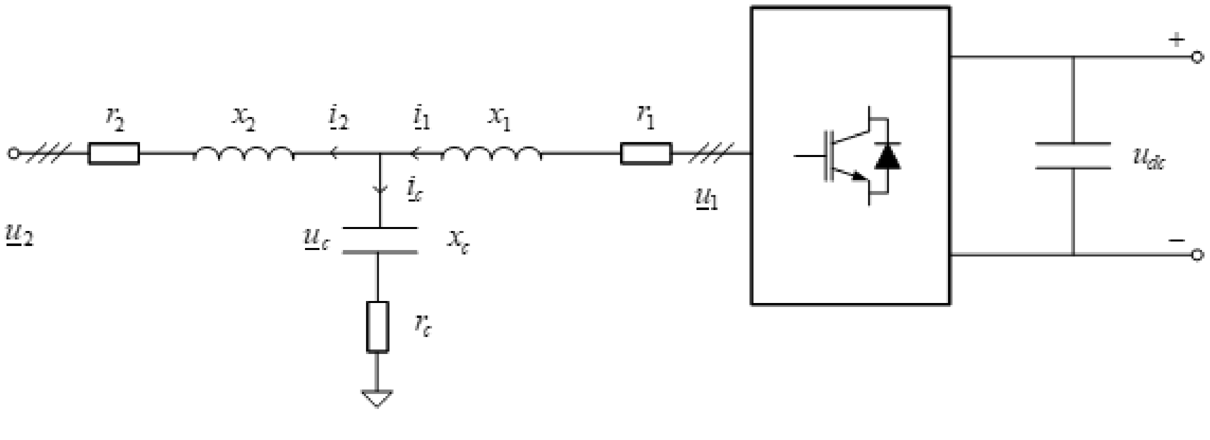


Figure 2.1: Simplified AFE-model[5]

For simplification purposes the capacitance resistance, r_C, is neglected, assuming $r_C = 0$. If a higher capacitance resistance is desired, it can be intercepted in the inverter resistance, r_1, or in the grid resistance,r_2[5].

The voltage at the converter for each phase, u_{1a}, u_{1b} and u_{1c}, is expressed by the equations below.

$$u_{1a} = r_1 \cdot i_{1a} + \frac{x_1}{\omega_N}\frac{di_{1a}}{dt} + u_{Ca} \tag{2.1}$$

$$u_{1b} = r_1 \cdot i_{1b} + \frac{x_1}{\omega_N}\frac{di_{1b}}{dt} + u_{Cb} \tag{2.2}$$

$$u_{1c} = r_1 \cdot i_{1c} + \frac{x_1}{\omega_N}\frac{di_{1c}}{dt} + u_{Cc} \tag{2.3}$$

In the given context, ω_N represents the nominal angular frequency. Furthermore, r_1 and x_1 denote the resistance and inductance on the converter side, respectively. The variable i_1 represents the current on the inverter side running through r_1 and x_1. Similarly, r_2 and x_2 is the resistance and inductance on the grid side, respectively, with i_2 indicating the current on the grid side running through r_2 and x_2. Then, the capacitance voltage for each phase, u_{Ca}, u_{Cb} and u_{Cc}, are derived as follows:

$$u_{Ca} = r_2 \cdot i_{2a} + \frac{x_2}{\omega_N}\frac{di_{2a}}{dt} + u_{2a}$$

$$u_{Cb} = r_2 \cdot i_{2b} + \frac{x_2}{\omega_N}\frac{di_{2b}}{dt} + u_{2b}$$

$$u_{Cc} = r_2 \cdot i_{2c} + \frac{x_2}{\omega_N}\frac{di_{2c}}{dt} + u_{2c}$$

Here, the parameters mentioned above yields, and u_2 represents the voltage on the grid side. Additionally, the current running through the capacitance is given in Equation 2.4.

$$i_C = \frac{1}{C} \cdot \frac{du_C}{dt} \tag{2.4}$$

It is known that $i_C = i_1 - i_2$ and $x_C = \frac{1}{C \cdot \omega_N}$. Equation 2.4 can then be rearranged for the three-phases as follows:

$$\frac{du_{Ca}}{dt} = \omega_N \cdot x_C \cdot (i_{1a} - i_{2a}) \tag{2.5}$$

$$\frac{du_{Cb}}{dt} = \omega_N \cdot x_C \cdot (i_{1b} - i_{2b}) \tag{2.6}$$

$$\frac{du_{Cc}}{dt} = \omega_N \cdot x_C \cdot (i_{1c} - i_{2c}) \tag{2.7}$$

For simplification, the equations for the three-phases is written in vector form, presented as follows:

$$\underline{u_1} = r_1 \cdot \underline{i_1} + \frac{x_1}{\omega_N}\frac{d\underline{i_1}}{dt} + \underline{u_C} \tag{2.8}$$

$$\underline{u_C} = r_2 \cdot \underline{i_2} + \frac{x_2}{\omega_N}\frac{d\underline{i_2}}{dt} + \underline{u_C} \tag{2.9}$$

$$\frac{d\underline{u_C}}{dt} = \omega_N \cdot x_C \cdot (\underline{i_1} - \underline{i_2}) \tag{2.10}$$

where the vectors are as stated below:

$$\underline{u_1} = \begin{bmatrix} u_{1a} \\ u_{1b} \\ u_{1c} \end{bmatrix}, \underline{u_2} = \begin{bmatrix} u_{2a} \\ u_{2b} \\ u_{2c} \end{bmatrix}, \underline{u_C} = \begin{bmatrix} u_{Ca} \\ u_{Cb} \\ u_{Cc} \end{bmatrix}, \underline{i_1} = \begin{bmatrix} i_{1a} \\ i_{1b} \\ i_{1c} \end{bmatrix}, \underline{i_2} = \begin{bmatrix} i_{2a} \\ i_{2b} \\ i_{2c} \end{bmatrix} \tag{2.11}$$

2.3.1 Three-phase to two-phase transformation

To simplify the model of the converter, the filter and grid components, the reference frame is changed from three-phase to three-phase. These are also known as $\alpha\beta$- or Clarke-transformation.

Transformation matrix

The Clarke-transformation converts three-phase abc-quantities to $\alpha\beta$-reference frame, where the transformation matrix is shown in Equation 2.12. Since the three-phase system is considered balanced, the γ - component is considered zero, and is neglected. The result of two-phase representation are scaled to be equal in magnitude to the original three-phases.

$$\begin{bmatrix} u_\alpha \\ u_\beta \end{bmatrix} = \frac{2}{3} \cdot \begin{bmatrix} 1 & \frac{-1}{2} & \frac{-1}{2} \\ 0 & \frac{\sqrt{3}}{2} & \frac{-\sqrt{3}}{2} \end{bmatrix} \cdot \begin{bmatrix} u_a \\ u_b \\ u_c \end{bmatrix} \tag{2.12}$$

For the amplitude-invariant transformation for a balanced three-phase system, the resulting system gives a two-phase system which are orthogonal to each other. It also follows that the α-component is aligned to phase a, whilst β-axis is leading by 90-degree from α-axis. Thus, the beta component is the only component that needs to be calculated. This saves some computational resources in the transformation[6]. The alpha and beta components can then be expressed as:

$$u_\alpha = u_a, u_\beta = \frac{1}{\sqrt{3}} \cdot (u_b - u_c) \tag{2.13}$$

Inverted Clarke-transformation

When the signal processing is finished, the three-phase calculation needs to be transformed back to three-phase before entering the machine. This is done by using the inverse Clarke-transformation. Equation 2.14 shows the inverted Clarke matrix.

$$\begin{bmatrix} u_a \\ u_b \\ u_c \end{bmatrix} = \frac{2}{3} \cdot \begin{bmatrix} 1 & 0 \\ -\frac{1}{2} & \frac{\sqrt{3}}{2} \\ -\frac{1}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix} \cdot \begin{bmatrix} u_\alpha \\ u_\beta \end{bmatrix} \tag{2.14}$$

Park-transformation

In the IP cores development, the reference framework used is the dq-reference. To achieve the dq-reference, a com-bination of the Clarke- and Park-transformation is done. The Park-transformation is represented in Equation 2.15.

$$\begin{bmatrix} i_d \\ i_q \end{bmatrix} = \frac{2}{2} \cdot \begin{bmatrix} cos(\omega t) & sin(\omega t) \\ -sin(\omega t) & cos(\omega t) \end{bmatrix} \cdot \begin{bmatrix} i_\alpha \\ i_\beta \end{bmatrix} \tag{2.15}$$

The corresponding inverse matrix is given in Equation 2.16.

$$\begin{bmatrix} i_\alpha \\ i_\beta \end{bmatrix} = \frac{2}{2} \cdot \begin{bmatrix} cos(\omega t) & -sin(\omega t) \\ sin(\omega t) & cos(\omega t) \end{bmatrix} \cdot \begin{bmatrix} i_d \\ i_q \end{bmatrix} \tag{2.16}$$

By applying the inverse Park-transformation of the controller, it becomes possible to calculate the current references using the dq-system and then transform it into the stator-oriented coordinates. This means calculating the AC-quantities from the DC-quantities[7].

Transformation of system model

It is assumed that there is no magnetic coupling between the phases, then the sum of current in the filter capacitors is assumed zero. This assumption applies to all vectors in the system, which is a balanced three-phase system. Consequently, the capacitor vector will not have a zero component in this scenario.

$$\begin{gathered} i_{10} = i_{1a} + i_{1b} + i_{1c} = 0 \\ i_{20} = i_{2a} + i_{2b} + i_{2c} = 0 \\ i_{C0} = i_{10} - i_{20} = 0 \\ u_{C0} = 0 \end{gathered} \tag{2.17}$$

Therefore, it is sufficient to only evaluate two components, d- and q- component or the α- and β-components as suggested.

2.3.2 Per-Unit Scaling

The emulated model employs scaling of parameters using the per-unit system. Utilizing the per-unit system offers several advantages. Firstly, it facilitates the detection of significant variations beyond the rated values, making it easier to identify deviations. Additionally, the same system can be utilized with different values, allowing the emulated system to be employed with smaller values in laboratory settings and larger values with larger components in practical applications.

The base values used are;

$$U_{base} = \hat{U}_{fn} = \sqrt{\frac{2}{3}} \cdot U_{N,rms,l-l} \tag{2.18}$$

$$I_{base} = \sqrt{2} \cdot i_{N,rms} \tag{2.19}$$

$$\tag{2.20}$$

The impedance base value can be calculated from the U_{base} and I_{base}.

$$Z_{base} = \frac{U_{base}}{I_{base}} = \frac{U_{N,rms,l-l}}{\sqrt{3} \cdot i_{N,rms}} \tag{2.21}$$

Then, the per-unit values of the parameters are calculated using the base values.

$$r_1 = \frac{R_1}{Z_{base}} \tag{2.22}$$

$$r_2 = \frac{R_2}{Z_{base}} \tag{2.23}$$

$$x_1 = \frac{\omega_N \cdot L_1}{Z_{base}} \tag{2.24}$$

$$x_2 = \frac{\omega_N \cdot L_2}{Z_{base}} \tag{2.25}$$

$$x_c = \frac{1}{\omega_N \cdot C \cdot Z_{base}} \tag{2.26}$$

2.3.3 Circuit-breaker and state-variables

When implementing the logic in the FPGA, two scenarios are considered for modeling the circuit-breaker connection to the grid. The first scenario involves the AFE being connected to the grid, then the closed. The second scenario includes the AFE being disconnected from the grid, then the breaker is open.

Initially, the breaker is assumed to be open. In this state, the current flows from the converter and passes through the capacitor, resulting in no current flowing to the grid, $i_2 = 0$. Since the voltage across the breaker is measured on the grid side, we can establish the condition $u_{breaker} = u_2$, leading to the following set of conditions.

$$u_{breaker} = u_2, i_2 = 0 \tag{2.27}$$

In addition, Equation 2.8, providing an equation for the voltage u_1, is derive presenting an expression for the current flowing from the converter in Equation 2.28. Furthermore, the scenario with an open breaker also includes the equation denoting the capacitor voltage depicted in Equation 2.29.

$$u_1 = r_1 \cdot i_1 + \frac{x_1}{\omega_N}\frac{di_2}{dt} + u_C$$

$$\frac{di_1}{dt} = \frac{\omega_N}{x_1} \cdot (-r_1 \cdot i_1 + u_1 - u_C) \tag{2.28}$$

$$\frac{du_C}{dt} = \omega_N \cdot x_C \cdot (i_1 - i_2) \tag{2.29}$$

For the second scenario, when the breaker is closed, the following condition applies:

$$u_{breaker} = u_C$$

Similarly to the first scenario where the breaker is open, an expression for the current flowing from the converter, denoted as i_1, is derived based on the equation for the voltage u_1 given in Equation 2.8. This expression is presented in Equation 2.30. Additionally, the equation denoting the capacitor voltage Equation 2.31 remains unchanged.

$$u_1 = r_1 \cdot i_1 + \frac{x_1}{\omega_N}\frac{di_1}{dt} + u_C$$

$$\frac{di_1}{dt} = \frac{\omega_N}{x_1} \cdot (r_1 \cdot i_1 + u_1 - u_C) \tag{2.30}$$

$$\frac{du_C}{dt} = \omega_N \cdot x_C \cdot (i_1 - i_2) \tag{2.31}$$

From Equation 2.9, an expression for the current passing through the breaker, denoted as i_2, is derived. This expression is presented in Equation 2.32.

$$u_C = r_2 \cdot i_2 + \frac{x_2}{\omega_N}\frac{di_2}{dt} + u_2$$

$$\frac{di_2}{dt} = \frac{\omega_N}{x_2} \cdot (-r_2 \cdot i_2 + u_C - u_2) \tag{2.32}$$

Breaker position	$u_{breaker}$	i_2
open	u_C	0
closed	u_2	nonzero

Table 2.1: Logic of the emulated model

2.3.4 Discrete model

Due to the inherent complexity of solving complex differential equations, a numerical method for computing the approximate solutions is required. The model is discretized, meaning that the continuous equations are transformed into a set of discrete equations that can be solved iteratively using numerical techniques.

2.3.4.1 Eulers method

One of the simplest numerical method is Euler's method. Euler's method is a one-step method. Given the ODE and the initial values, (t_0, y_0), some timestep, T_{step}, is chosen, and $t_1 = t_0 + T_{step}$. Based on this, an approximation for $y_1 = y(t_1)$ is calculated. Based on the new calculated values, (t_1, y_1), the process is repeated, where $t_2 = t_1 + T_{step}$.

Eulers method is derived from the Taylor expansion:

$$y(t_0 + T_{step}) = y(t_0) + T_{step} \cdot \frac{dy(t_0)}{dt} + \frac{1}{2}T_{step}^2 \frac{d^2y(t_0)}{dt^2} + ... \tag{2.33}$$

Assume that the step size, T_{step} is small, and the first two terms is dominating in the expression. The Forward Euler method is derived and given as:

$$y(t_0 + T_{step}) \approx y(t_0) + T_{step} \cdot f(t_0, y_0) + O(T_{step}^2) \tag{2.34}$$

$$y_{n+1} = y_n + T_{step} \cdot f(t_0, y_0) + O(T_{step}^2) \tag{2.35}$$

By employing the Forward Euler's method, the equations derived in subsection 2.3.3 for the matematical model of the LCL-filter is discreteized.

For the scenario when the breaker is open, Equation 2.36 - 2.38 shows the discreteized version of Equation 2.28 and 2.29.

$$u_{breaker}[k] = u_2[k], i_2 = 0 \tag{2.36}$$

$$i_1[k+1] = i_1[k] + \frac{\omega_N \cdot T_{step}}{x_1} \cdot (-r_1 \cdot i_1[k] + u_1[k] - u_C[k]) \tag{2.37}$$

$$u_C[k+1] = u_C[k] + \omega_N \cdot T_{step} \cdot x_C \cdot (i_1[k] - i_2[k]) \tag{2.38}$$

For the scenario where the breaker is closed, Equation 2.39 - 2.42 shows the discretized version of Equation 2.30, 2.31 and 2.32.

$$u_{breaker}[k] = u_C[k] \tag{2.39}$$

$$i_1[k+1] = i_1[k] + \frac{\omega_N \cdot T_{step}}{x_1} \cdot (-r_1 \cdot i_1[k] + u_1[k] - u_C[k]) \tag{2.40}$$

$$i_2[k+1] = i_2[k] + \frac{\omega_N \cdot T_{step}}{x_2} \cdot (-r_2 \cdot i_2[k] + u_{breaker}[k] - u_2[k]) \tag{2.41}$$

$$u_C[k+1] = u_C[k] + \omega_N \cdot T_{step} \cdot x_C \cdot (i_1[k] - i_2[k]) \tag{2.42}$$

Due to the assumption of a small time step, denoted as T_{step}, and the exclusion of second-order and higher terms, a local truncation error (LTE) is introduced at each time step during the numerical solution process [8]. For small values of T_{step}, the error incurred in each step is approximately proportional to T_{step}^2, as the largest term is multiplied by T_{step}^2 and the subsequent terms decrease with a factor of T_{step}^2. This relationship is specific to first-order differential equations. In general, a method of $O(T_{step}^{k+1})$ is considered to be of order k. Another significant error is the global error, which represents the absolute difference between the true solution and the calculated solution. However, if the true solution is unknown, the exact error cannot be determined. Nevertheless, neglecting round-off errors, the global error at the n-th time step is proportional to the LTE multiplied by n. Since n is inversely proportional to T_{step}, the global error can be expressed as $\frac{LTE}{T_{step}}$, indicating that the global error scales with T_{step}^k.

The Forward Euler's method is considered an explicit method, as it calculates the next step, y_{n+1}, based on the known quantities of the current state, y_n, and $f(t_n, y_n)$. While it is straightforward to implement, this method can be numerically unstable if the time step is too large. However, to address the stability issue, implicit methods can be used instead. The Backward Euler's method serves as the counterpart to the Forward Euler's method.

$$y_{n+1} = y_n + T_{samp} \cdot f(y_{n+1}, t_{n+1}) + O(T_{samp}^2) \tag{2.43}$$

Since $f(y_{n+1}, t_{n+1})$ is not known in advance and needs to be determined using root-finding algorithms like Newton-Raphson, the implicit method becomes more time-consuming and costly to implement.

2.3.4.2 Numerical stability

As previously mentioned, explicit methods can be numerically unstable. To further investigate and explain this issue, we consider the linear initial value problem (IVP):

$$\frac{dy}{dt} = -\lambda y, y(0) = 1 \tag{2.44}$$

with $\lambda > 0$ the solution is known to be:

$$y(t) = e^{-\lambda t} \tag{2.45}$$

This is a stable and smooth solution, starting at 1 for $t = 0$, and goes to 0 for $t = \infty$. Lets look at the discrete form, with the forwards Euler's method[8].

$$y_{n+1} = y_n - \lambda T_{step} y_n = (1 - \lambda T_{step}) y_n = (1 - \lambda T_{step})^2 y_{n-1} = ... = y_0 (1 - \lambda T_{step})^{n+1} \tag{2.46}$$

Using the aforementioned expression, an upper boundary is given for managing the numeric stability.

2.3.5 Phase-locked Loop

Phase-locked loop (PLL) is a phase-tracking subsystem useful for synchronizing power electronic converters to the power grid. It plays a crucial role in ensuring that the frequency, amplitude, and phase angle align with the certain requirements when synchronizing with rest of the power grid. The PLL assists in establishing a reliable connection to the grid when the specified criteria are met.

Design rules[9]:

$$h_0(s) = \left(k_p + \frac{K_p}{T_i} \cdot \frac{1}{s}\right) \cdot \frac{1}{s} = \left(2 \cdot \zeta \cdot \omega_f + \omega_f^2 \cdot \frac{1}{s}\right) \cdot \frac{1}{s} = \frac{2 \cdot \zeta \omega_f}{s} + \left(\frac{\omega_f}{s}\right)^2 \tag{2.47}$$

$$m(s) = \frac{1 + 2\zeta \cdot \frac{s}{\omega_f}}{1 + 2\zeta \cdot \frac{s}{\omega_f} + \left(\frac{s}{\omega_f}\right)^2} = \frac{1 + \frac{2\zeta}{\omega_f} \cdot s}{1 + 2\zeta \cdot \frac{s}{\omega_f} + \left(\frac{s}{\omega_f}\right)^2} \tag{2.48}$$

For our topology: The error input for the C++ function in the PESC controller platform for the PLL is the measured angle for $u_{breaker}$, $e = \zeta - \zeta_{filtered}$.

$$h_0(s) = \left(k_p + \frac{K_p}{T_i} \cdot \frac{1}{s}\right) \cdot \frac{\omega_n}{s} = \frac{\omega_n \cdot k_p}{s} + \left(\frac{\sqrt{\frac{\omega_n \cdot k_p}{T_i}}}{s}\right)^2 \tag{2.49}$$

$$m(s) = \frac{h_0(s)}{1 + h_0(s)} = \frac{\frac{\omega_n \cdot k_p}{s} + \left(\frac{\sqrt{\frac{\omega_n \cdot k_p}{T_i}}}{s}\right)^2}{1 + \frac{\omega_n \cdot k_p}{s} + \left(\frac{\sqrt{\frac{\omega_n \cdot k_p}{T_i}}}{s}\right)^2} = \frac{1 + T_i \cdot s}{1 + T_i \cdot s + \frac{T_i}{\omega_n \cdot k_p} \cdot s^2} \tag{2.50}$$

following

$$m(s) = \frac{1 + 2\zeta \cdot \frac{s}{\omega_f}}{1 + 2\zeta \cdot \frac{s}{\omega_f} + \left(\frac{s}{\omega_f}\right)^2} = \frac{1 + \frac{2\zeta}{\omega_f} \cdot s}{1 + 2\zeta \cdot \frac{s}{\omega_f} + \left(\frac{s}{\omega_f}\right)^2} \tag{2.51}$$

$$k_p = 2\zeta \cdot \frac{\omega_f}{\omega_n} = 2\zeta \cdot \frac{f_f}{f_n}, T_i = \frac{2\zeta}{\omega_f} = \frac{\zeta}{\pi \cdot f_f} \tag{2.52}$$

The damping term in the transfer function represents the system's ability to attenuate or suppress oscillations in the output frequency. It is usually controlled by adjusting the PLL's loop filter parameters, such as the gain, K_p, and the time constant, T_i. Damping ratio is a dimensionless quantity that measures the rate at which oscillations decay in a power system following a disturbance. It is usually denoted by the symbol ζ. A higher damping ratio indicates stronger energy dissipation and faster decay of oscillations, leading to improved system stability[10].

2.3.6 Active dampening

To mitigate the harmful effects of harmonics originating from the grid and protect the components from damage, the control system incorporates active damping. The primary objective of active damping is to store the current in the capacitor within the converter.

The transfer function for the output of the LCL-filter including the active dampening is derived as follows: Initially, the voltage equation governing the inductor, which aligns with the equation presented in Equation 2.8, is introduced. By transforming Equation 2.8 into the s domain, the following equation is obtained.

$$u_1(s) = (r_1 + \frac{x_1}{\omega_n} \cdot s) \cdot i_1(s) + u_C(s) \tag{2.53}$$

To incorporate active damping, the current sourced from the converter, denoted as i_1, is subtracted by an imaginary current flowing through a resistance connected in parallel with the capacitor [9]. Unlike passive damping, which involves the inclusion of a resistance in parallel with the capacitor, active damping stores the current within the converter. This principle of active damping not only prevents heat dissipation in the resistance, but also safeguards against component overheating. Furthermore, an expression representing the current flowing through the capacitor is provided:

$$i_C = \frac{1}{\omega_n \cdot x_C} \cdot \frac{du_C}{dt} = i_1 - i_D - i_2 = i_1 - \frac{u_C}{r_D} - i_2 \tag{2.54}$$

Where i_D represents the active current, r_D is the value of the resistance that is included to achieve the active dampening. In the s-domain, the derivation of the equation proceeds as follows:

$$(\frac{1}{r_D} + \frac{s}{\omega_n \cdot x_C}) \cdot u_C(s) = i_1 - i_2$$

$$u_C(s) = \frac{i_1 - i_2}{\frac{1}{r_D} + \frac{s}{\omega_n \cdot x_C}} \tag{2.55}$$

Then, by substituting Equation 2.55 into Equation 2.53, The transferfunction for the current controller of i_1 can be derived and obtained:

$$u_1(s) = \left(r_1 + \frac{x_1}{\omega_n} \cdot s + \frac{1}{\frac{1}{r_D} + \frac{s}{\omega_n \cdot x_C}} \right) \cdot i_1(s) = z_1(s) \cdot i_1(s) \tag{2.56}$$

$$\frac{i_1}{u_1} = \frac{1}{z_1(s)} = \frac{1}{r_1 + \frac{x_1}{\omega_n} \cdot s + \frac{1}{\frac{1}{r_D}+\frac{s}{\omega_n \cdot x_C}}} = \frac{\frac{1}{r_D} + \frac{s}{\omega_n \cdot x_C}}{1 + \frac{r_1}{r_D} + \frac{r_1 \cdot s}{\omega_n \cdot x_C} + \frac{x_1 \cdot s}{\omega_n \cdot r_D} + \frac{x_1}{\omega_n^2 \cdot x_C} \cdot s^2}$$

x_C is changed to the admittance, where $y_C = \frac{1}{x_C}$.

$$= \frac{1}{r_1 + r_D} \cdot \frac{1 + r_D \cdot y_C \cdot \frac{s}{\omega_n}}{1 + \left(\frac{r_1 \cdot r_D}{r_1 + r_D} \cdot y_C + \frac{x_1}{r_1 + r_D} \right) \cdot \frac{s}{\omega_n} + \frac{1}{1 + \frac{r_1}{r_D}} \cdot x_1 \cdot y_C \cdot \left(\frac{s}{\omega_n} \right)^2} \tag{2.57}$$

Under the assumption that r_D is significantly larger than r_1, the sum $r_D + r_1$ can be approximated as r_D. Consequently, the equation can be simplified accordingly.

$$\approx \frac{1}{r_D} \cdot \frac{1 + r_D \cdot y_C \cdot \frac{s}{\omega_n}}{\left(r_1 \cdot y_C + \frac{x_1}{r_D} \right) \cdot \frac{s}{\omega_n} + x_1 \cdot y_C \cdot \left(\frac{s}{\omega_n} \right)^2} \tag{2.58}$$

Equation 2.58 shows the final expression for the current controller of i_1 with active dampening.

2.4 Hysterises controller

Hysteresis controllers have conventionally been employed as current controllers. This approach involves measuring the actual motor current and comparing it with a predetermined reference value. When the actual current is lower than the reference current, the switch is turned on, and vice versa when the actual current slightly exceeds the reference value. The width of the hysteresis band determines the average switching frequency, in conjunction with the circuit's voltages and inductance. A well-known issue in this context is the decrease in switching frequency as the motor speed increases, resulting from the diminishing disparity between the voltage at the converter's input and the internal induced voltage in the motor. To mitigate this, adaptive hysteresis width techniques are sometimes implemented [7].

Chapter 3

PESC Control Platform

The main-focus the present thesis is to make an emulator based on the AFE, to faciliate tests that can be conducted in the physical lab with real components. The emulated model is built to be interfaced in the PESC software and firmware control platform. An advantage is that the parameters used for the AFE are taken as inputs in the IP cores, instead of hardware components. The user can then easily change the values. To best explain the most important aspects of how the PESC platform works, the hardware and software of the control platform are explained in a simplified manner.

The subsequent sections of this report will provide an explanation of the process involved in implementing the discretized model within the FPGA. These sections will offer detailed insights into the mathematical representation, algorithmic considerations, and the overall design and construction of the FPGA-based emulator.

3.1 Introduction on embedded real-time simulator- ERTS

When developing AFE connected to a LCL-filter, the system is simulated to emulate the model as close as possible. The Embedded real-time simulator makes it possible to simulate faster and achieve a rapid prototyping of the model. To achieve real-time emulation, the power hardware components as LCL-filter is programmed in the Field programmable gate array(FPGA) fabric of the Zynq System-on-Chip (SoC). The main component that makes this possible is that the execution tasks is happening in the FPGA. Matlab Simulink is used for simulation of the system and controller. First the model is implemented using PC- simulations. Then, the model is emulated using mathematical blocks. The mathematical equation used is then discretized using Euler's method. The discretized equations are used building the model in Simulink add on, system generator provided by Xilinx.

As proposed in [4], the ERTS used is mainly divided in three parts. This is allowing separate development, which again corresponds to reusage of common components, ensures uniformity across simulations, minimizes design and simulation failures, and helps diagnose the faults or design errors rapidly. By retaining much of the ERTS modularized, the proposed simulator can be modified to emulate different motor-types and designs, converters or mechanical loads. The proposed simulator is designed using the per-unit system, which makes its main components common across machines, loads and converters irrespective of their ratings.

3.2 Field- programmable gate array

A Field Programmable Gate Array (FPGA) is constructed using configurable logic blocks (CLBs), also known as slices or logic cells. CLBs consists of flip flops and lookup tables, playing crucial roles in the functionality of an FPGA [11]. The flip flops store a logic input, 0 or 1, until the next clock edge, while lookup tables determine the interconnection of the logic blocks. Developing an FPGA entails creating a digital circuit through the utilization of hardware description languages like VHDL or Verilog.[6]. These languages serve as the primary means for defining the logic algorithms executed on the FPGA chip. Following the design phase, it is common practice to verify the FPGA code through rigorous testing. Subsequently, the text-based logic is compiled through a series of complex steps, resulting in a configuration file or bitstream. This file contains information on how the logic components should be interconnected.

One significant advantage of FPGAs is their inherent ability to execute tasks concurrently, enabling faster and more efficient simulation times. This parallelism arises from the allocation of dedicated sections within the chip for each task, allowing for independent functioning without interference from other logic blocks.

Moreover, FPGAs facilitate accelerated development of control algorithms, as they can be rapidly prototyped and tested before deployment in real-world applications. This expedites product development cycles and facilitates the early detection of potential issues, ultimately reducing costs associated with rectifying problems during later stages of development.

3.3 Software and Hardware in the PESC control platform

3.3.1 Hardware

The hardware components of the PESC control platform consist of a PicoZed 7030 control board from Avnet mounted on a process interface board developed by SINTEF. The process interface board facilitates control of two two-level three-phase inverters and provides up to eight input measurement ports. The control board consists of a Zynq-7030 SoC (prosessor) that includes two ARM floating-point processors (CPUs) and one programmable logic FPGA device. One of the CPUs runs a Linux program responsible for programming the remaining CPU and FPGA, as well as enabling the process card to be programmed and monitored using specialized release and monitoring tools. These tools, including the Linux program, release tools, and monitoring tools, are provided by The Switch Marine Drives [12].

3.3.2 Software

Programming of control board involves using three different software, all provided by Xillinx. The code for the remaining CPU is written in C++ using Software Development Kit (SDK). The project simulated on the PESC platform needs to be created and uploaded to the board. The FPGA is programmed by connecting IP-cores together in a program called Vivado and the project to be uploaded in the FPGA is created. The IP-cores are created by using Xilinx's System Generator plugin in Simulink. Vivado also provide a generic IP-core catalogue that can be used.

To execute the emulated model in the FPGA, specific routines need to be performed within the PESC platform. The programming language used in the PESC Control Platform is C++. To upload the software to the hardware equipment, a zip file, the release, is built to be uploaded to the processor board. The ReleaseBuilder tool from The Switch is used to build a software package, which includes the TSW software for CPU0, the NTNU software for CPU1, and the binary file, built in the Vivado software, for the FPGA[13]. A release can be built with or without a database. The database contains various parameters used in the software. In a Master's or PhD work, it may be sufficient to hard-code the parameter values in the ConfigParameters.cpp file. The WatchView program is used for communication of the board and visualizion of the running.

As mentioned, the Switch is providing the release builder. The release builder program is uploading a database to the processor board and rebooting the board. The WatchView-program is used for communicating with the processor. The database has to be configured and created for a specific Drive software, which means that the parameters for the Drive Layer of the database become different for PM and DC motors. However, some parameters are the same. This is typical for the Application layer and the Converter Layer.

During the startup process of the SINTEF board and PicoZed board, the following actions are performed. First, CPU0 unzips the project file, uploaded during the release, and programs the CPUs (CPU0, CPU1) and FPGA. CPU0 writes the project database into the Flash memory and initializes the parameters in the program of CPU1, as specified in the database editor during database generation[13]. Once these steps have been executed, CPU1 initiates the execution of main.cpp, which is programmed by NTNU. The main tasks performed by the program include:

- Setting up the communication channel between CPU0 and CPU1
- Initializing communication routines
- Initializing data loggers
- Running the SW1.Initialize() routine developed by NTNU
- Starting the interrupt routines, including interrupt priorities
- Entering an infinite while loop: Background routine

The next thing to describe is the interrupt routine. The interrupt routine is built as a layer model programmed in C++ code. The interrupt signal is given on the top and bottom of the PWM signals. It is designed in such way that the interrupt signal has enough time to finish before the next signal occurs. The interrupt routine is designed to handle interrupts that immediately need attention from the processor. The routine is run and checks if any of the system states are changed. The routine is calling the function called, DriveRoutineFast(), and the following procedures is executed in the code.

As mentioned earlier the Drive routine function is a layer model, which means that the structure of the code is divided into four classes called application layer, Drive layer, Converter layer and firmware layer.

Then, the interrupts routine are described. First, the measurements from the FPGA is collected, typically currents and voltages. Subsequently, the status is examined by reading the converter's status bits. In the event of over currents, over- or under-voltages, specific fault bits are set, and the converter's status is flagged as ”Fault.” Similar tests are performed in the Drive Layer. If some fault occur, the drive will be tripped, i.e. turned off. The completed drive will be set in the state Malfunction in the Application SW.

The Application function is run in the application layer. It consists of a state machine based on the driveCom-standard. This is an old standard, which has been the basis for several other standards. The states are used for allowing you to go from one state to another like shown in Figure 3.1.

The Drive Layer is controlled by the Application Layer. The Application can request the local state of the Drive Layer, but also send parameters, such as the torque reference and power, limits, etc. The Drive Layer will send actual values and status to the Application Layer. Notably, this communication framework operates on a ”need-to-know” basis, ensuring that the Application Layer remains agnostic to the specific motor type being controlled.

Similarly, the Converter Layer is governed by the Drive Layer. The Drive Layer can request the local state of the Converter Layer. The Drive Layer can also send the required Voltage vector length and angle to the converter layer. The Converter Layer will send actual measurements and status to the Drive Layer. The communication between the layers does not share any unnecessary information.

3.4 Scaling

As the Simulation have switched over to fix point numbers, there are limited storage capacity in the FPGA, and a right scaling is needed. The FPGA-based implementation consists of building blocks that digitally replicate components of the LCL filter, such as resistors (R), inductors (L), and capacitors. The emulated model, developed within the FPGA, is designed to operate in a per-unit system and is parameterized by the application software. The advantages of this emulator is that the scaling block ensures that the generic IP cores don't see a difference between the physical hardware and the emulator when exchanging the parameters. This makes the design procedures valuable for the model design[4].

3.5 Numeric Representation (Binary Design)

The AXI interface in the System-on-Chip (SoC) has a word length of 32 bits, which is why the Extended Real-Time System (ERTS) also maintains a 32-bit word length. This ensures convenient data exchange with the on-chip processor. Additionally, a 32-bit representation guarantees numerical accuracy for real-time simulations[4]. Within the 32-bit representation, the precision allows for a range from +7.99 to -8, with one bit dedicated to representing the sign. As the FPGA operates using fixed-point numbers with a specified time step parameter, it is important to consider the storage requirements when selecting the time step. If the time step is too small, it may lead to challenges in representing extremely small numbers. For example, if a number cannot be expressed with 64 bits, the integral may result in zero. To optimize hardware utilization, the model employs a 32-bit representation.

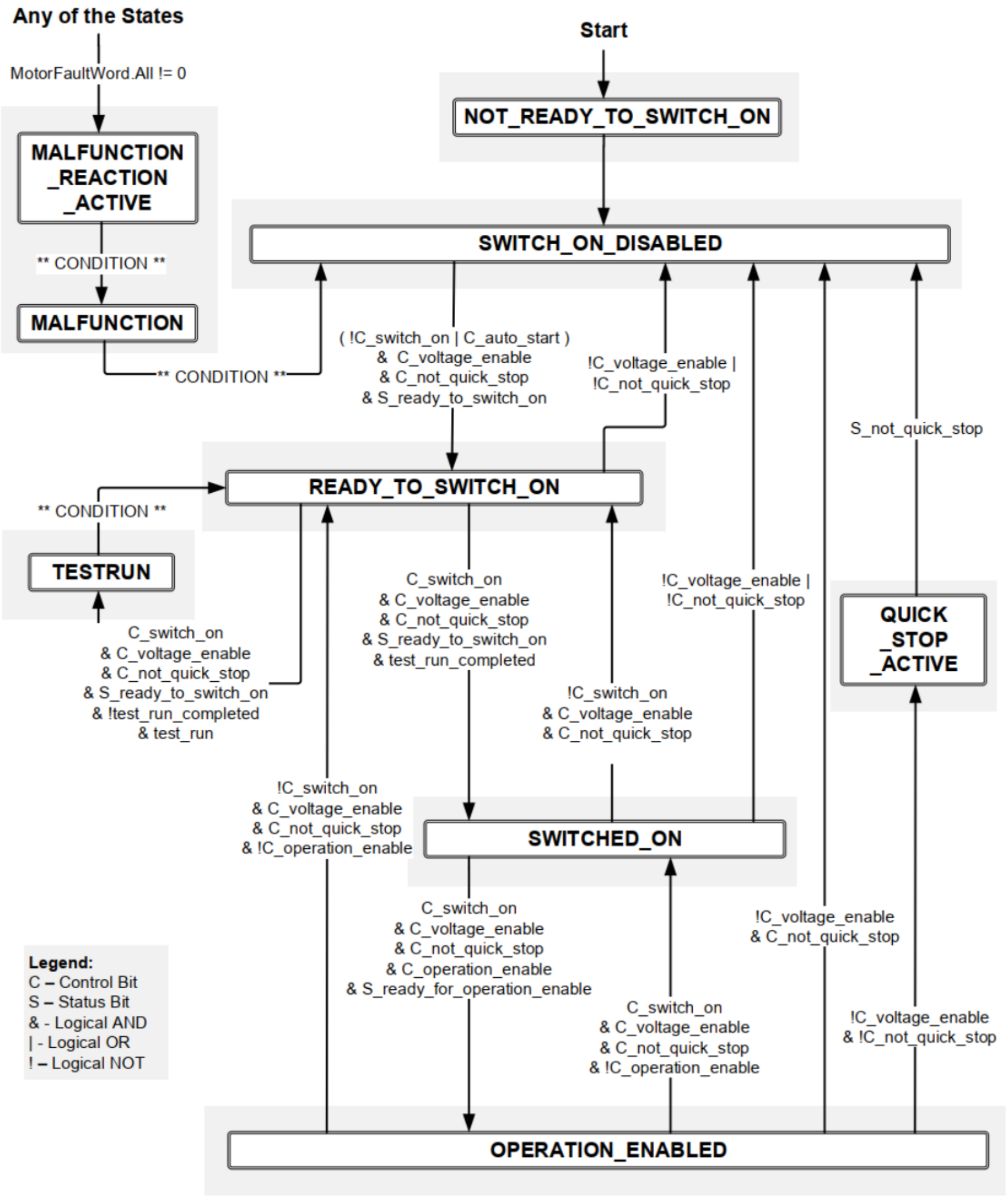

Any of the States
MotorFaultWord.All != 0
MALFUNCTION _REACTION _ACTIVE
** CONDITION **
MALFUNCTION
** CONDITION **
Start
NOT_READY_TO_SWITCH_ON
SWITCH_ON_DISABLED
(!C_switch_on | C_auto_start) & C_voltage_enable & C_not_quick_stop & S_ready_to_switch_on
!C_voltage_enable | !C_not_quick_stop
S_not_quick_stop
READY_TO_SWITCH_ON
** CONDITION **
TESTRUN
C_switch_on & C_voltage_enable & C_not_quick_stop & S_ready_to_switch_on & !test_run_completed & test_run
C_switch_on & C_voltage_enable & C_not_quick_stop & S_ready_to_switch_on & test_run_completed
!C_voltage_enable | !C_not_quick_stop
QUICK _STOP _ACTIVE
!C_switch_on & C_voltage_enable & C_not_quick_stop
!C_switch_on & C_voltage_enable & C_not_quick_stop & !C_operation_enable
SWITCHED_ON
C_switch_on & C_voltage_enable & C_not_quick_stop & C_operation_enable & S_ready_for_operation_enable
!C_voltage_enable & C_not_quick_stop
C_switch_on & C_voltage_enable & C_not_quick_stop & !C_operation_enable
!C_voltage_enable & !C_not_quick_stop
Legend:
C – Control Bit
S – Status Bit
& - Logical AND
| - Logical OR
! – Logical NOT
OPERATION_ENABLED

Figure 3.1: Figure of the DriveCom states in the application layer [13]

3.6 Clock settings

The representation of the physical motor drive system involves the utilization of three distinct clock cycles: the FPGA clock, the processor interrupt cycle, and the solver clock [4]. The FPGA clock, with a duration of 10 ns, is employed for components requiring high processing speed, such as the generic IP cores and the converter block. The processor interrupt cycle, which operates at twice the speed of the pulse-width modulation (PWM), is sufficient for the application software. The solver clock, with a duration of 1 µs, is utilized in the discretized model solver within the hardware emulator. Solver clock is used for the load models in the hardware emulation. T_{step} will determine the integration intervals, thus shorter integration intervals yield more precise results. So, small step size and high numerical representation give precise results.

In the SysGen model, a generator clock signal is employed to synchronize the timing of the emulated RC circuit, necessitating higher-speed operation. However, this increased speed also leads to higher chip temperatures due to the increased number of real switching steps.

When conducting simulations of any type, it is important to consider the duration of the simulation for the different components. The FPGA can operate at a maximum clock speed of 100 MHz, providing the capability for efficient and timely execution of simulations.

3.7 Implementation of emulator in FPGA

The development of an FPGA-based control platform utilizes the graphical design tool System Generator Xilinx in Simulink. Simulink, coupled with the Xilinx library, serves as the programming environment for FPGA implementation, enabling the utilization of blockset libraries containing various blocks for IP core design within Simulink. This approach offers the advantage of block diagram design and HDL code testing in a unified Simulink environment. The two primary IP cores employed in this model are the transformation core and the current core, which have already been developed and utilized in a motor drive application. Therefore, leveraging the pre-existing IP cores significantly reduces unnecessary effort and accelerates the development process of the FPGA for the present model. This section only describes the block setup for building the emulated model of the LCL filter for the AFE. For the development of the IP cores, Thomas S. Haugen made valuable contributions to the implementation of the FPGA and the design of the IP core.

3.7.1 LCL filter model

The emulated model of the LCL filter consists of mathematical equation for the inverter current, grid current and the capacitor voltage. The Xilinx Simulink blockset operates using a dedicated fixed-point signal format. Further, the implementation of the inverter currents, grid current and capacitor voltage consists of the discretized equations from subsection 2.3.4.

The equation below is used for building the representation of the converter current with Xilinx blocks shown in Figure 3.2.

$$i_1[k+1] = i_1[k] + \frac{\omega_N \cdot T_{step}}{x_1} \cdot (-r_1 \cdot i_1[k] + u_1[k] - u_C[k]) \tag{3.1}$$

Further, the following equation is used when building the calculation for the grid current with Xilinx blocks represented in Figure 3.3.

$$i_2[k+1] = i_2[k] + \frac{\omega_N \cdot T_{step}}{x_2} \cdot (-r_2 \cdot i_2[k] + u_{breaker}[k] - u_2[k]) \tag{3.2}$$

Finally, the next equation provides the initial point for the Xilinx representation of the capacitor voltage calculations.

$$u_C[k+1] = u_C[k] + \omega_N \cdot T_{step} \cdot x_C \cdot (i_1[k] - i_2[k]) \tag{3.3}$$

Prior to exiting the FPGA-module, it is necessary to convert the grid current from the dq-reference frame to the three-phase abc coordinates. The transformation process can be represented by a Xilinx block, as illustrated in Figure 3.6.

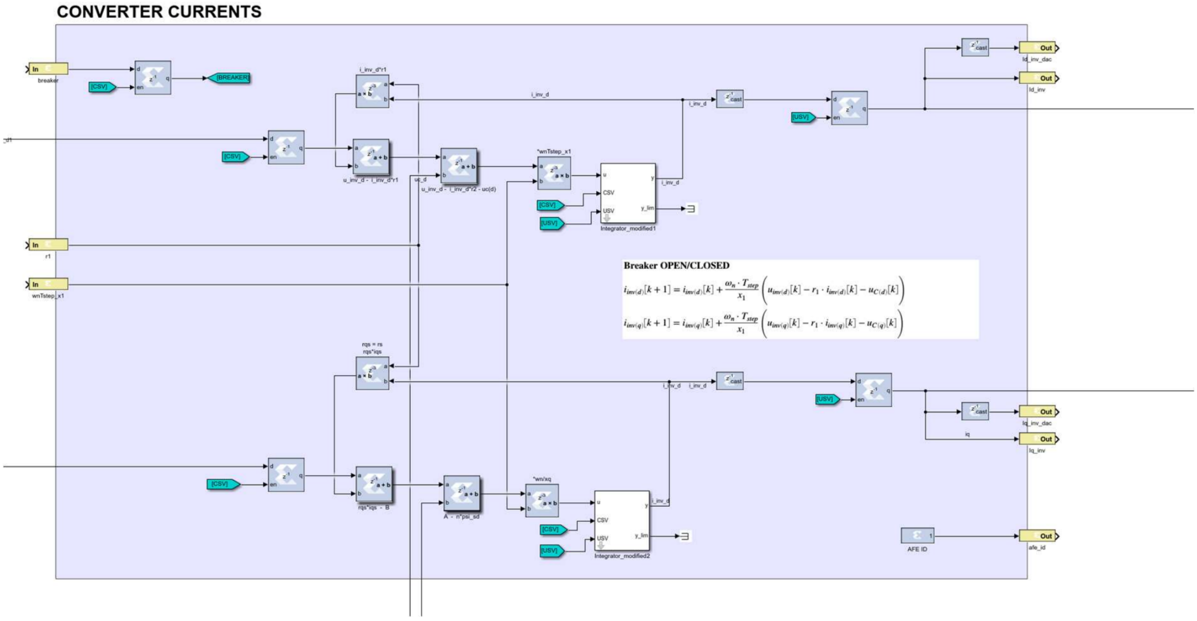
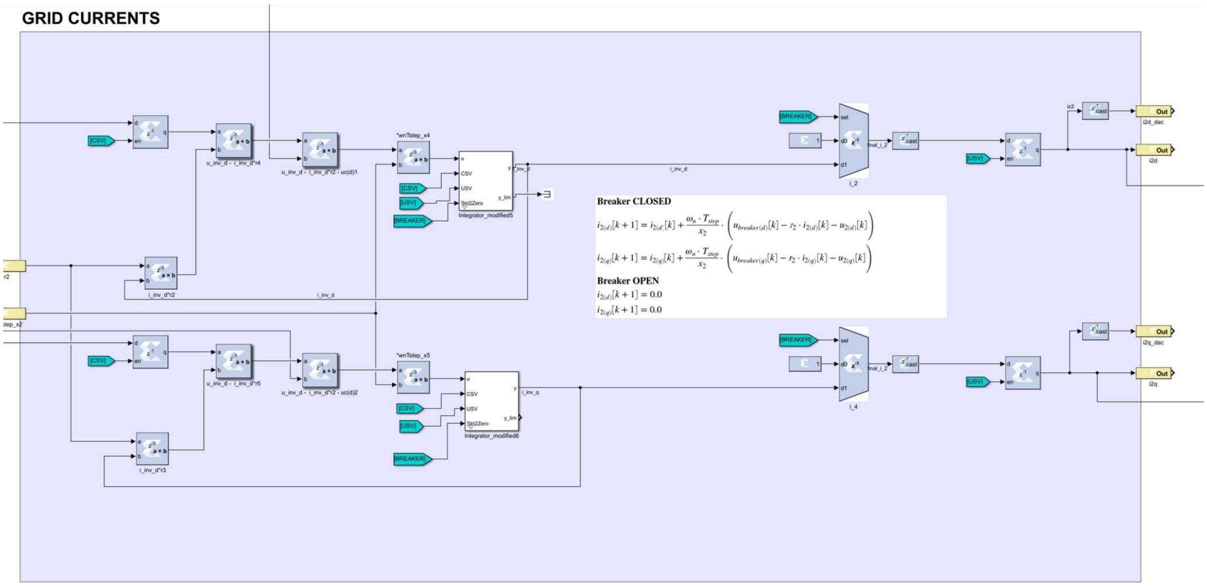

CONVERTER CURRENTS
In
breaker
[CSV]
d
en
z-1
q
[BREAKER]
i_inv_d*r1
a × b
i_inv_d
u_inv_d - i_inv_d*r1
a + b
uc_d
u_inv_d - i_inv_d*r2 - uc(d)
*wnTstep_x1
u
CSV
USV
y
y_lim
Integrator_modified1
[USV]
cast
Out
Id_inv_dac
Id_inv
r1
wnTstep_x1
Breaker OPEN/CLOSED
$i_{inv(d)}[k+1] = i_{inv(d)}[k] + \frac{\omega_n \cdot T_{step}}{x_1}\left(u_{inv(d)}[k] - r_1 \cdot i_{inv(d)}[k] - u_{C(d)}[k]\right)$
$i_{inv(q)}[k+1] = i_{inv(q)}[k] + \frac{\omega_n \cdot T_{step}}{x_1}\left(u_{inv(q)}[k] - r_1 \cdot i_{inv(q)}[k] - u_{C(q)}[k]\right)$
rqs = rs
rqs*iqs
rqs*iqs - B
A - n*psi_sd
*wn/xq
Integrator_modified2
iq
Iq_inv_dac
Iq_inv
AFE ID
afe_id

Figure 3.2: The converter current calculation represented with Xilinx blocks.



Breaker CLOSED

$i_{2(d)}[k+1] = i_{2(d)}[k] + \frac{\omega_n \cdot T_{step}}{x_2} \cdot \left(u_{breaker(d)}[k] - r_2 \cdot i_{2(d)}[k] - u_{2(d)}[k] \right)$

$i_{2(q)}[k+1] = i_{2(q)}[k] + \frac{\omega_n \cdot T_{step}}{x_2} \cdot \left(u_{breaker(q)}[k] - r_2 \cdot i_{2(q)}[k] - u_{2(q)}[k] \right)$

Breaker OPEN

$i_{2(d)}[k+1] = 0.0$

$i_{2(q)}[k+1] = 0.0$

Figure 3.3: The grid current calculation represented with Xilinx blocks.

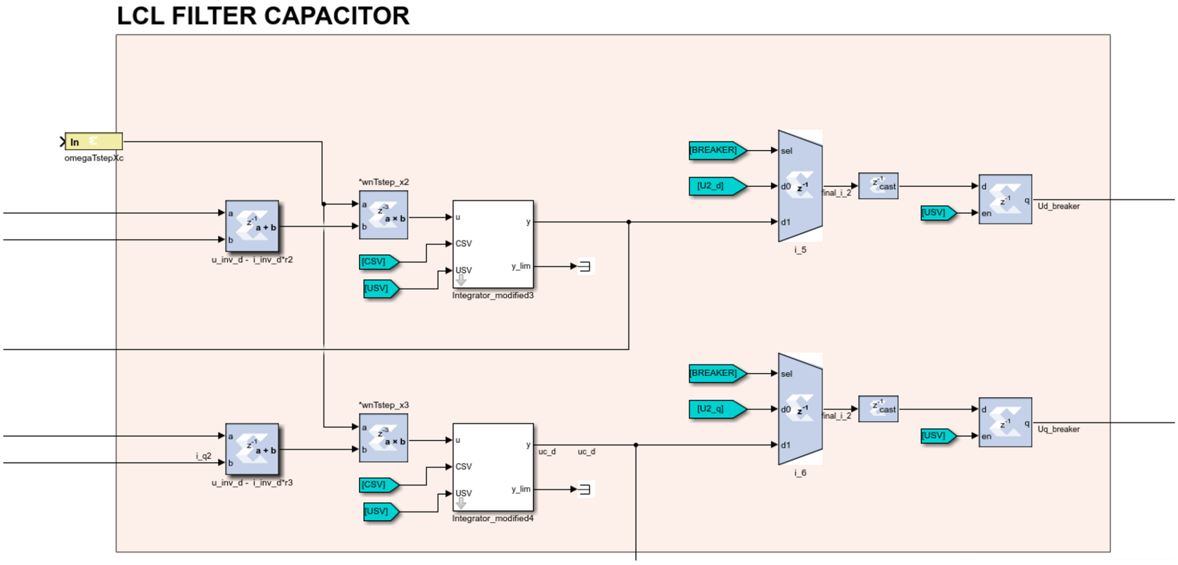


Figure 3.4: The capacitor voltage calculation represented with Xilinx blocks.

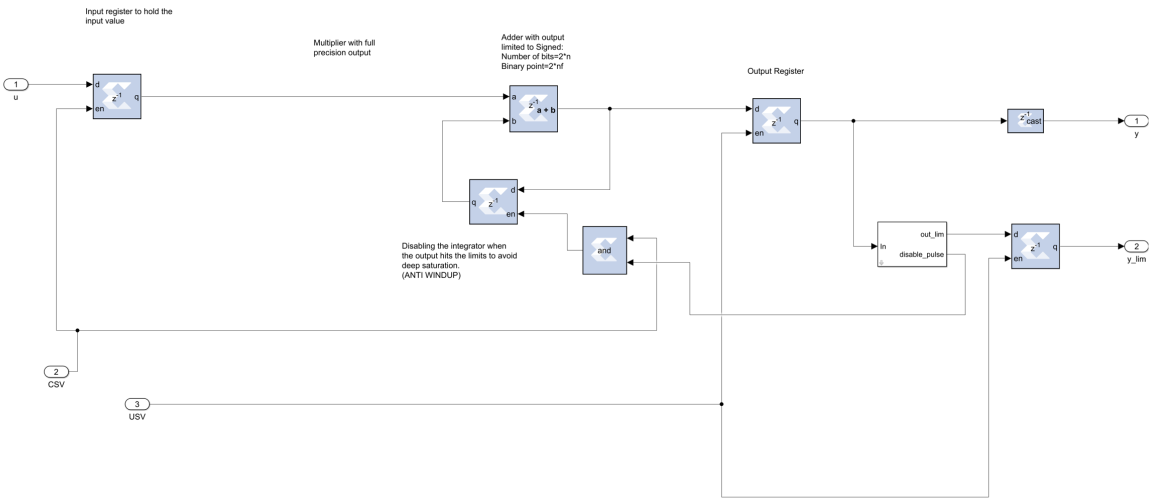
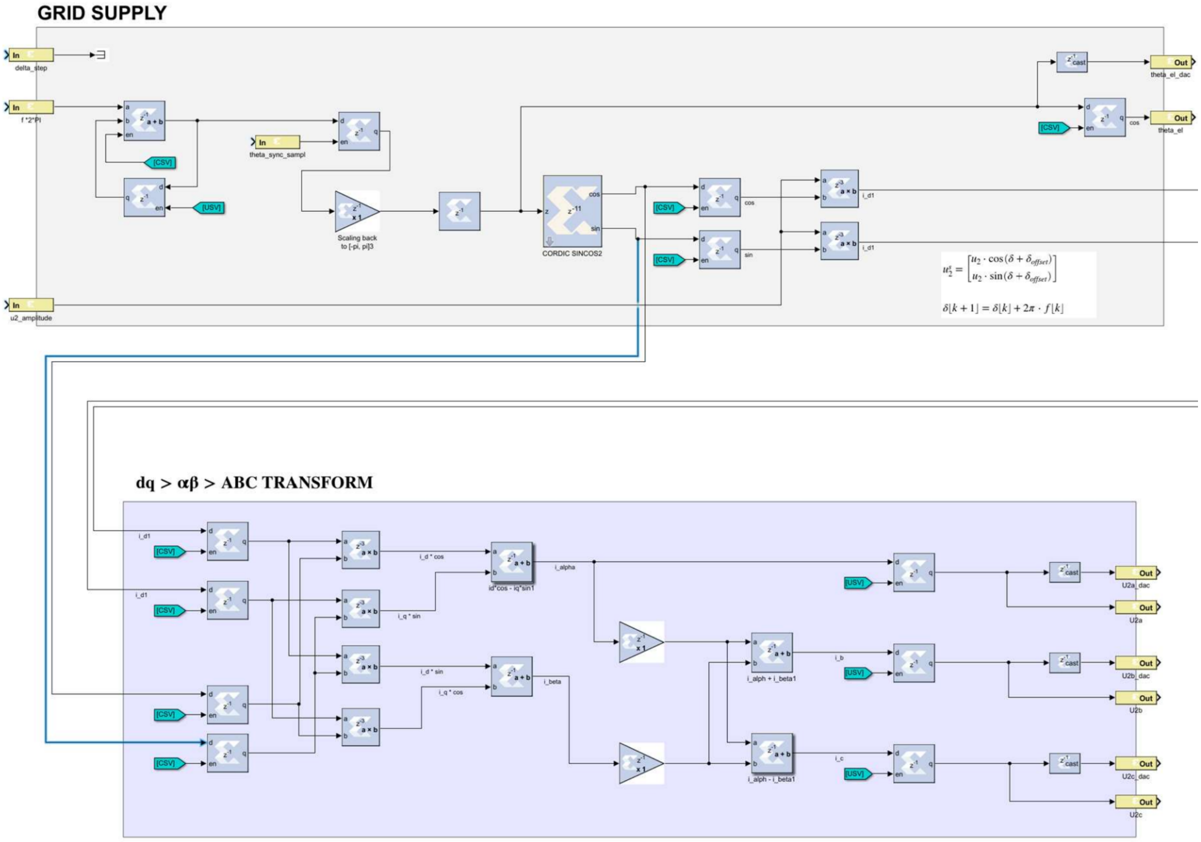

Input register to hold the input value
Multiplier with full precision output
Adder with output limited to Signed: Number of bits=2*n Binary point=2*nf
Output Register
1
u
d
en
q
a
b
a + b
cast
1
y
Disabling the integrator when the output hits the limits to avoid deep saturation. (ANTI WINDUP)
and
In
out_lim
disable_pulse
2
y_lim
2
CSV
3
USV

Figure 3.5: Details of the discretized integrator.



$$u_2^s = \begin{bmatrix} u_2 \cdot \cos(\delta + \delta_{offset}) \\ u_2 \cdot \sin(\delta + \delta_{offset}) \end{bmatrix}$$

$$\delta[k+1] = \delta[k] + 2\pi \cdot f[k]$$

Figure 3.6: The angle and grid voltage step representation with Xilinx blocks.

Chapter 4

Simulations

In this chapter, simulations of an Active Electric Filter (AFE) connected to a LCL-filter is concucted. The purpose is to analyze and evaluate the behavior of the system using different simulation models. The simulations are carried out using MATLAB Simulink and the ERTS is tested using the PESC control platform.

4.1 Simplified Simulink model

In this section, the simulated model represents an active front end (AFE) connected to a LCL-filter, where the converter is implemented as an ideal current source. The purpose of this simplified model is to gain a better understanding of the system's dynamics and functionality, with a particular focus on studying the grid following control scheme. A description of the Simulink-based model can be found in appendix section A. The model is

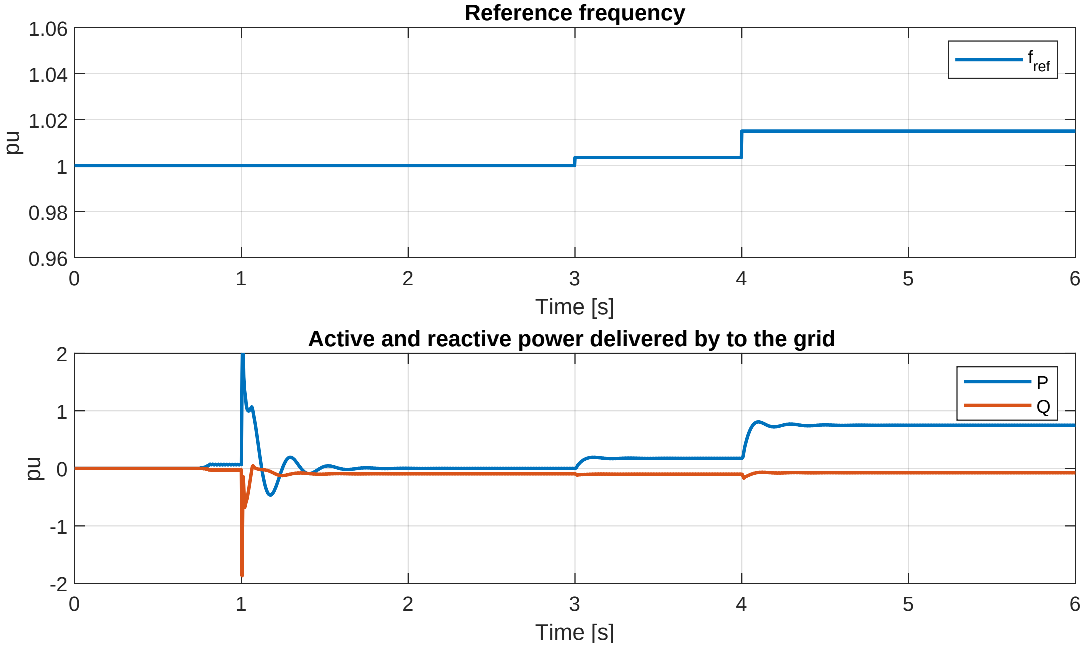


Figure 4.1: Simulation of the frequency cange and the active and reactive power delivered to the grid.

connected to a stiff grid with nominal frequency at 60 Hz and an rms line to line voltage magnitude at 690 V. The power circuit breaker is closed at 1 second, concecently the grid is connected to the AFE converter including filter. To demonstrate the grid following control scheme, the reference frequency in the model is changed. In the time duration from 0 to 3 seconds, the frequency reference equal 1 pu. Then from 3 to 4 seconds the reference frequency is changed from 1 pu to 1.0035 pu. Lastly, at 4 seconds, the frequency reference is altered to 1.015 pu.

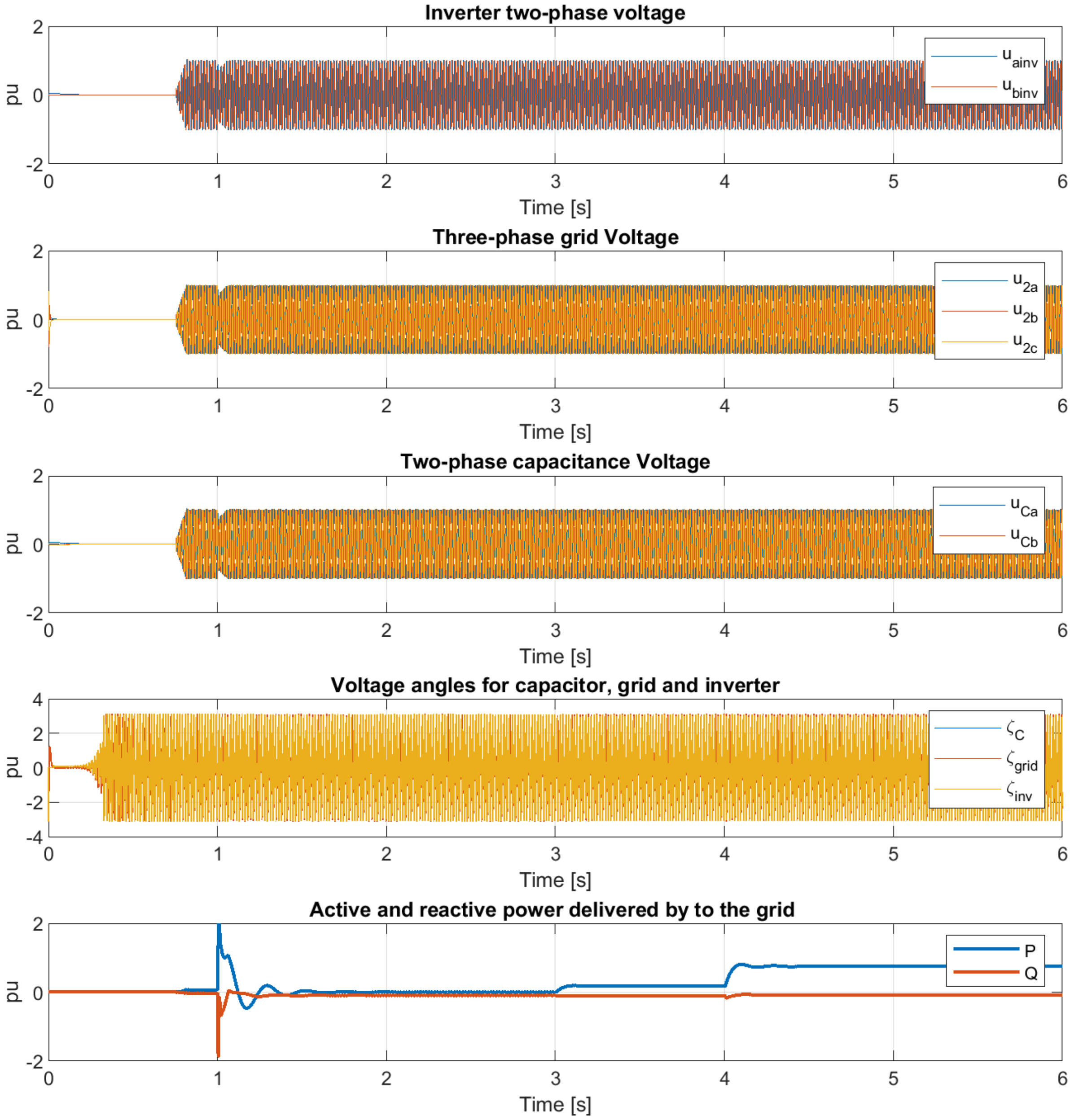


Figure 4.2: Simulation with PESC controller of the starting state with the breaker open.

From top to bottom, Figure 4.2 shows the inverter voltage, grid voltage, capacitor voltage, voltage angles and delivered active and reactive power to the grid. Regardless of the change in when the grid is connected, there are no variations in the system voltages.

It is evident that the model adjusts its active and reactive power to accommodate the system changes. Figure 4.1

illustrates the changes of P and Q delivered to the grid according to the frequency change in the grid.

It is observed that when the frequency change, the active and reactive power change within the system to adapt the system change, as expected when using the grid-following control scheme. When the converter is connected to the system, transient responses can be observed, leading to variations in the respective power system values.

4.1.1 Exact model

In this section, the AFE model is upgraded to provide a more precise representation of the converter system. The current source, that initially represented the converter, is replaced with a real converter featuring the power electronic IGBT switches, resulting in a more realistic and complex model. This enhancement takes into account the influence of the fast switching frequencies of the half bridges. As a consequence, disturbances such as voltage fluctuations become more apparent. Control of the converter is achieved through the implementation of a virtual synchronous machine (VSM) and a phase-locked loop unit (PLL) also including the hysteresis current controller. The input of the emulator model will be the measured inverter voltage, and the grid is a stiff three-phase voltage of adjustable voltage amplitude and frequency.

Furthermore, to achieve the goal of the fix-point model implemented in the FPGA, a emulated model of the LCL-filter is tested in Simulink. The mathematical model of the LCL-filter described in section 2.3 is built with the floating point data type Simulink blocks to ensure its capability in capturing the desired dynamics of the system. To check if the model gives the correct dynamic reactions, the filter capacitor current and voltage from the emulated model are compared with the filter capacitor current and voltage of the Simulink model with the realistic converter.

Parameter	Value
$V_{gn,L-L}$	690 V
I_n	1500 A
fn	60 Hz
capasitance, C_c	990e-6 F
inverter reactance, X_1	83e-6 H
Grid reactance, X_2	106e-6 H
inverter resistance, R_1	2.77e-3 Ω
Grid reactance, R_2	8.08e-4 Ω

Table 4.1: The parameters used in the accurate Simulink model

Parameter	Value
capasitance, y_c	0.10 pu
inverter reactance, x_1	0.11782 pu
Grid reactance, x_2	0.15047 pu
inverter resistance, r_1	0.01043 pu
Grid reactance, r_2	3.0424e-3 pu

Table 4.2: The parameters presented in Table 4.1 converted to the per unit system.

To ensure a meaningful comparison between the emulated mathematical model and the exact model, it is crucial to align their parameter settings. Consequently, the parameters used in the simulation of the exact model are presented in Table 4.1. However, it should be noted that the emulated model is scaled in the per-unit system to fit into the 32-bit fixed-point number system. Therefore, the parameters have been converted to per-unit values and are presented in Table 4.2. This conversion enables a consistent framework for evaluating and contrasting the emulated and accurate models, taking into account their specific characteristics and requirements. The emulated model, built with the floating point data type Simulink blocks, can be seen in the Appendix A.2.

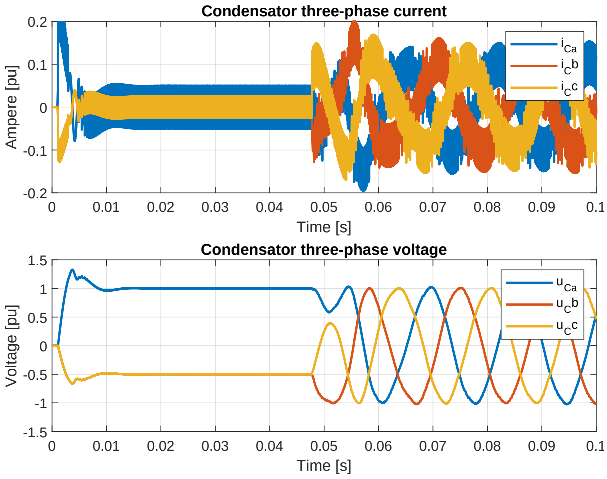


Figure 4.3: Plot of filter capacitor current and voltage in the exact simulink model

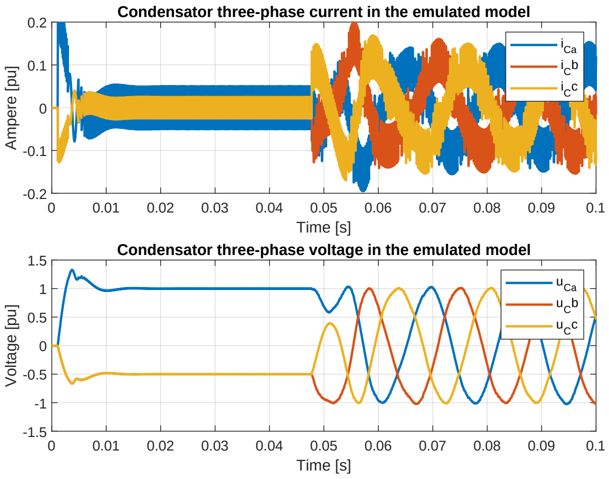


Figure 4.4: Plot of filter capacitor current and voltage in mathematical emulated model

Figure 4.3 and 4.4 represents the time-varying behavior of the current flowing through the filter capacitor and the voltage over the capacitance in each phase of the three-phase system for the Simulink model and the emulated model respectively.

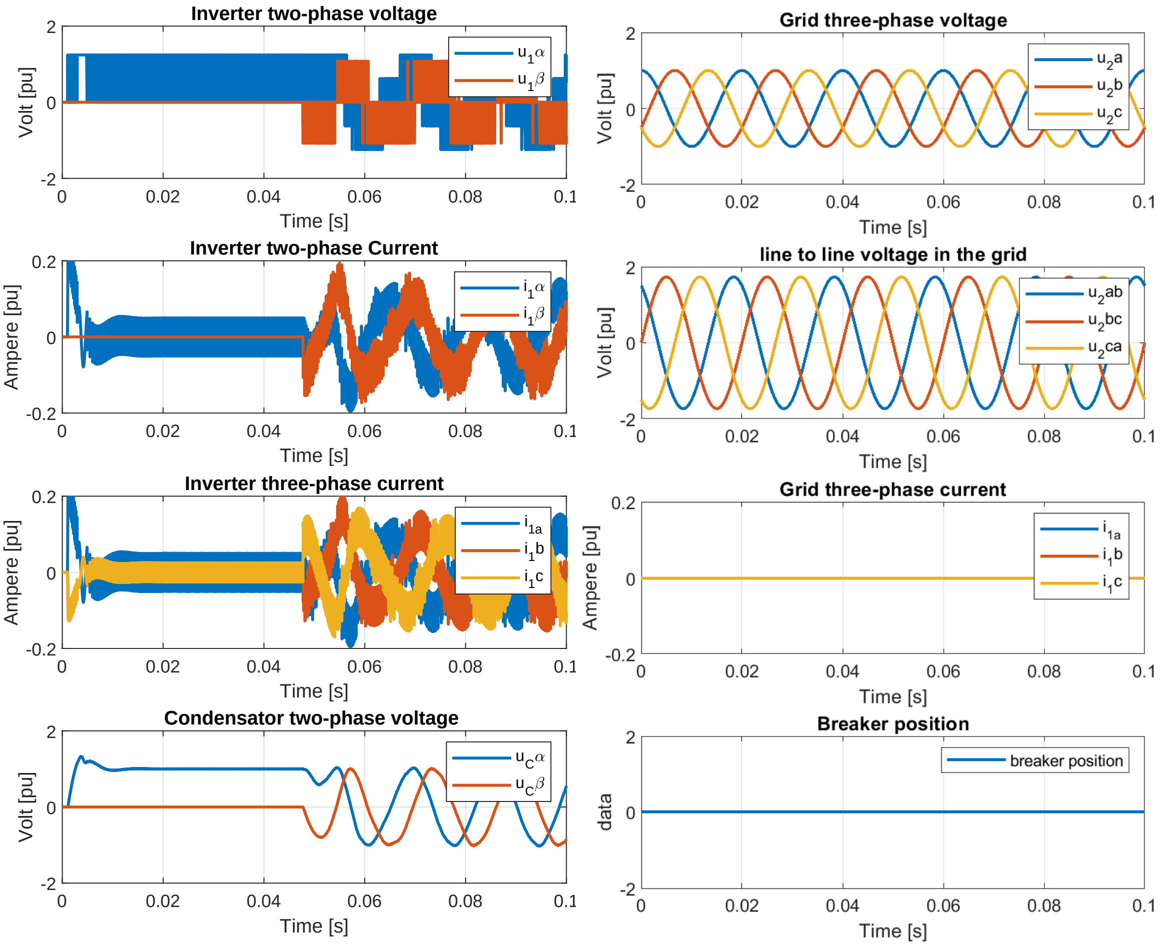


Figure 4.5: The above plots shows the grid voltage and current, v_2 and i_2, The inverter voltage and current, v_1 and i_1, the filter capacitor voltage, v_C, and the breaker position.

By comparing Figure 4.3 and 4.4 it can be seen that the mathematical model gives the same output current and voltage and that the two plots are identical.

Figure 4.5 shows a plot of some values of the emulated model. The simulation performed before the breaker is closed to the grid, because the exact model does not stabilize after the grid is connected to the converter. The active and reactive effect is drifting off after some time, which can be observed from Figure 4.6. Therefore the breaker position is displayed as zero, which indicates that the breaker between the grid and converter is open. Another indication of the open breaker is the grid current, shown to be zero. These aspects imply that it does not flow any current through to the grid when the breaker is open.

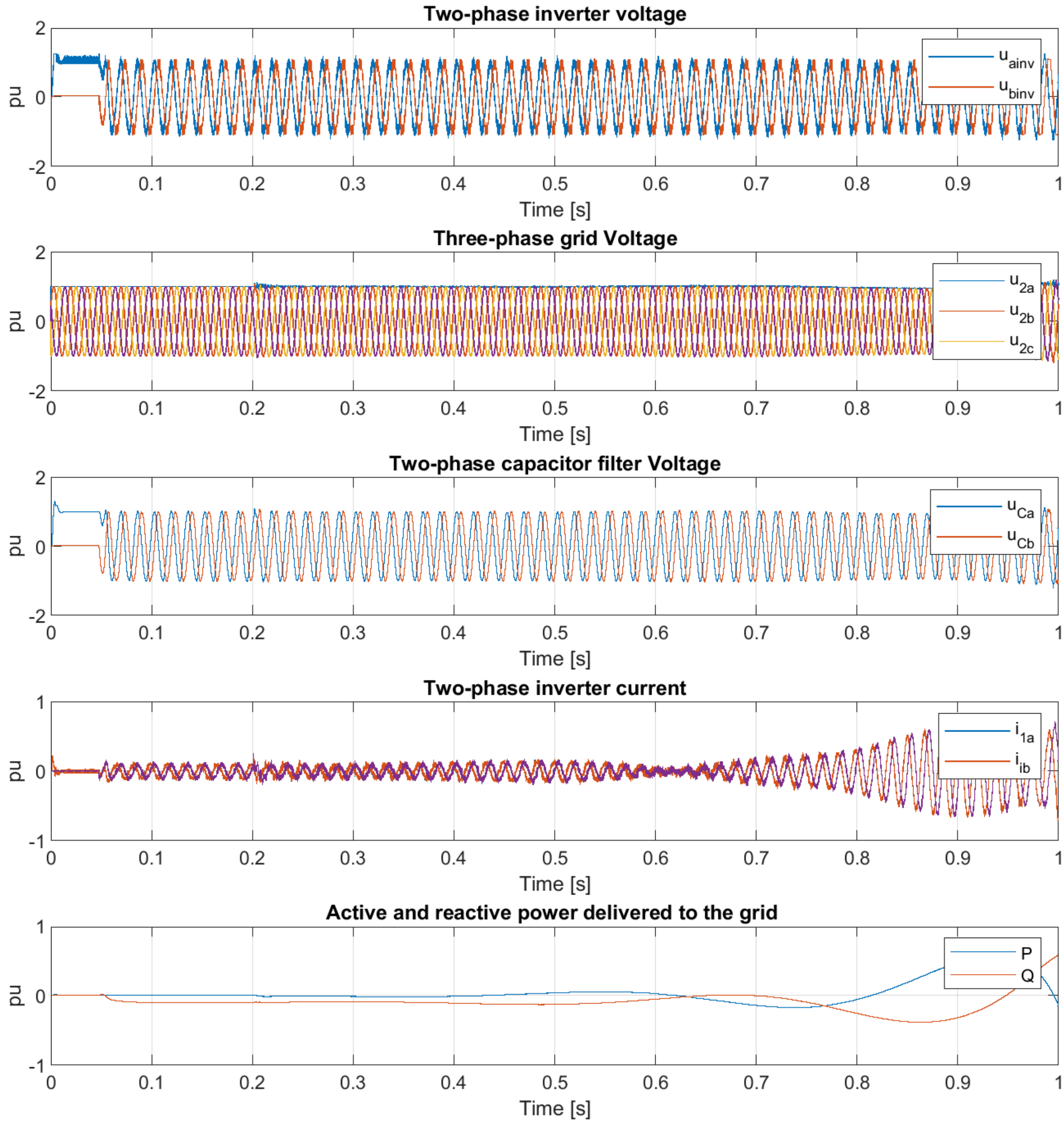


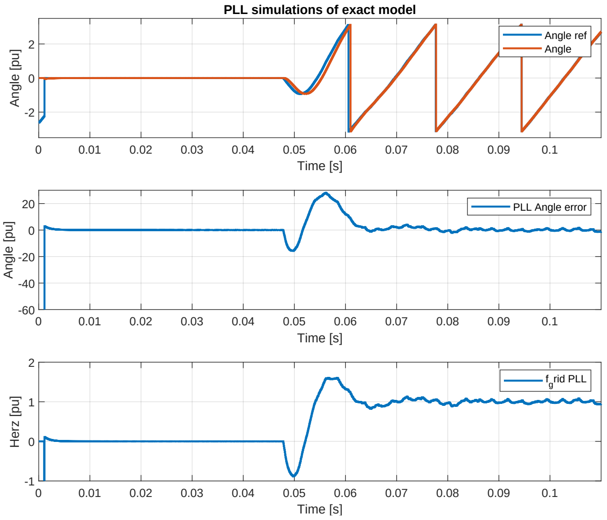
Figure 4.6: Simulation from the exact model showing the off drifting of the active and reactive power.

4.1.2 Testing of PLL

In this section, the PLL dynamics for the exact simulink model is tested. Four different sets of values were simulated for the PLL, as outlined in Table 4.3. The parameters ζ, f_n, and f_f were selected, while K_p and T_i were computed based on the expressions provided in Equation 2.52. Figure 4.7, 4.8,4.9 shows the dynamic of PLL with the parameters displayed in Table 4.3.

	ζ	f_n[Hz]	f_f[Hz]	K_p	T_i[s]
Figure 4.7	1.4	60	58	2.71	0.0077
Figure 4.8	1.4	60	35	1.63	0.0127
Figure 4.9	1.4	60	20	0.93	0.0223

Table 4.3: Table of the simulated ζ, f_n, f_f, Kp and Ti values for the PLL in the exact Simulink model.



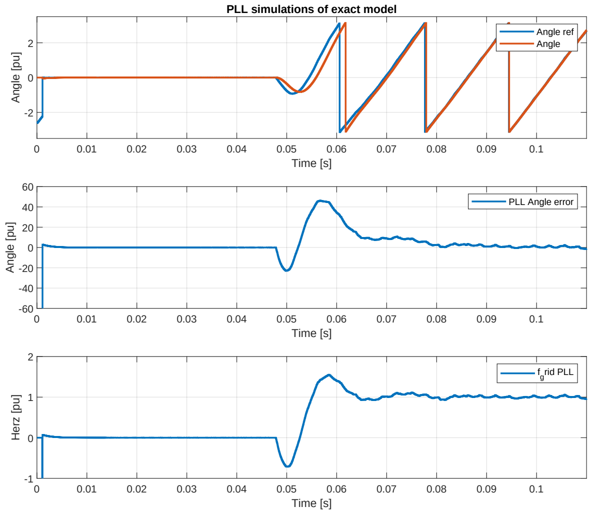


Figure 4.7: Simulation with the exact model for testing PLL, with $f_f = 58Hz$ and $\zeta = 1.4$, which gives $T_i = 7.7ms$ and $K_p = 2.71$

Figure 4.8: Simulation with the exact model for testing PLL, with $f_f = 35Hz$ and $\zeta = 1.4$, which gives $T_i = 12.7ms$ and $K_p = 1.63$

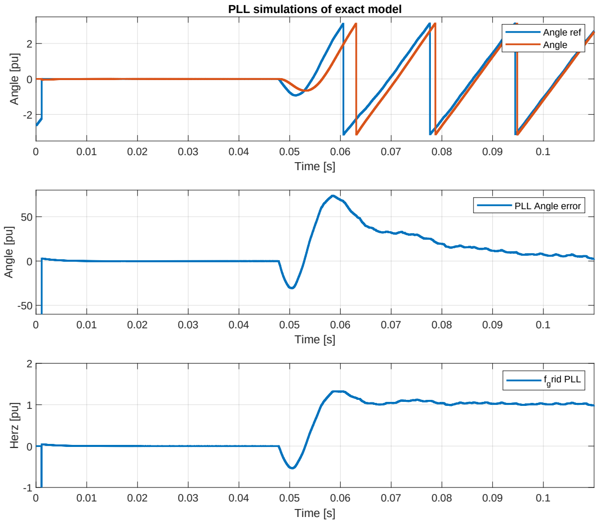


Figure 4.9: Simulation with the exact model for testing PLL, with $f_f = 20Hz$ and $\zeta = 1.4$, which gives $T_i = 22.3ms$ and $K_p = 0.93$

It is observed with decreasing value of K_p the angle error gets bigger. Also the highest angle error occurs at the values $K_p = 0.93$ and $T_i = 0.0223$.

4.2 Simulation of model in PESC platform

Finally, the discretized equations for implementing the PESC platform on the FPGA-module is derived by discretizing the mathematical model employed in the emulated continuous-time Simulink model. This discretized equations serve as the foundation for constructing the FPGA-based emulator, PESC platform.

In this section, the emulated model have been developed and put together for the FPGA to receive the binary compositions for the model. The release builder tool is used for zipping a project file and reboot the processor board for correct project running. The phase-locked loop for the model is mainly investigated. Further, a series of diverse scenarios have been tested in the PESC control platform to investigate their respective system behaviors. These scenarios include: (1) the starting state with the breaker open, (2) the starting state with the breaker closed with and without active dampening, and (3) the simulation involving a load increase. The first and second scenario is without a connected load which can be observed in the plots. The affect of the Active dampening is investigated. A simulation with a load change is also performed to observe the behavior of the model.

4.2.1 Parameter representation

Parameter	Value
V_{sN}	140 V
I_{sN}	11.5 A
S_{base}	2788.6 kVA
Z_{base}	7.0286 A
fn	78.5 Hz
capasitance addmittance, y_c	0.1 pu
inverter reactance, x_1	0.1 pu
Grid reactance, x_2	0.1 pu
inverter resistance, r_1	0.0001 pu
Grid reactance, r_2	0.0001 pu
Active dampening resistance, r_D	2.0 pu

Table 4.4: This table shows the base values used to calculate the per unit values of the passive components, and the Per unit values of the passive components of the LCL-filter used in the PESC control platform during the simulations.

Parameter	Value
capasitance addmittance, y_c	0.70286 F-1
capasitance addmittance, C	1.425 mF
inverter reactance, x_1	0.70286 ø
inverter inductance, L_1	1.425 mH
Grid reactance, x_2	0.70286 ■
Grid inductance, L_2	1.425 mH
inverter resistance, r_1	0.70286 mΩ
Grid reactance, r_2	0.70286 mΩ
Active dampening resistance, r_D	14.0572 Ω

Table 4.5: This table shows the passive components and their real values used in the PESC control platform during the simulations.

Table 4.4 shows the base values used to calculate the per-unit values of the passive components, and the per-unit values of the passive components of the LCL-filter used in the PESC control platform during the simulations. To be able to calculate the resonance frequency, the actual values is converted from the pu values. The following equation is used to convert the per-unit values, as depicted in Table 4.4, into their corresponding actual values. The resulting actual parameters are displayed in Table 4.5.

$$Actual\ value = Pu\ value \cdot Base\ value$$

4.2.2 Tuning the PLL

Initially, the optimal parameters for the PLL is determined by conducting experiments with various parameters using three distinct parameter sets specifically designed for the PLL. The parameters utilized in this evaluation are presented in Table 4.6. Similar to the exact model in Simulink, $zeta$, f_f and f_n are chosen where K_p and T_i is calculating using the equations shown in Equation 2.52. Subsequently, the simulation results of the PLL within the PESC control platform are depicted in Figure 4.10, 4.11 and 4.12. The fundamental frequency, f_n, is different from the PLL simulation in the Simulink model, where the fundamental frequency was set to 60 Hz instead of 78.7 HZ, which is used here.

	ζ	f_n[Hz]	f_f[Hz]	K_p	T_i[s]
Figure 4.10	1.6	78.7	70	2.85	0.00727
Figure 4.11	1.6	78.7	40	1.63	0.01273
Figure 4.12	1.6	78.7	23	0.93	0.02214

Table 4.6: Table of the simulated ζ, nominal frequency, f_n, fault frequency, f_f, Kp and Ti values for the PLL in the PESC platform.

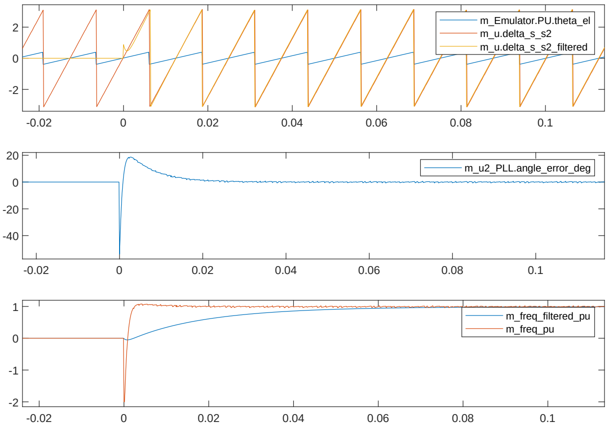


Figure 4.10: Simulation with PESC controller for testing PLL, with $f_f = 70Hz$ and $\zeta = 1.6$, which gives $T_i = 7.27ms$ and $K_p = 2.85$

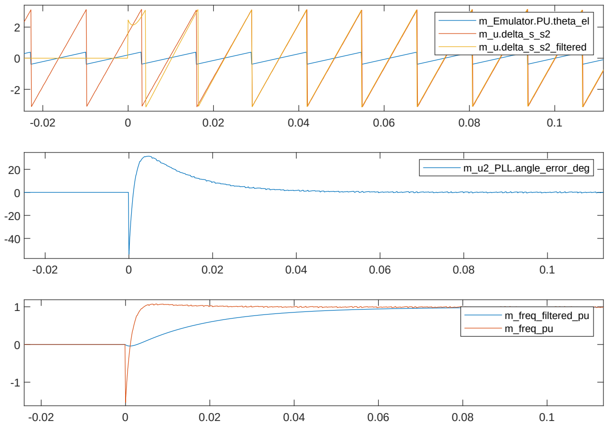


Figure 4.11: Simulation with PESC controller for testing PLL, with $f_f = 40Hz$ and $\zeta = 1.6.$, which gives $T_i = 12.73ms$ and $K_p = 1.63$

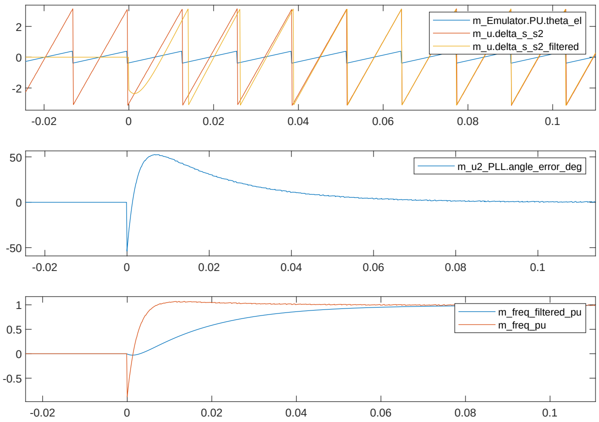


Figure 4.12: Simulation with PESC controller for testing PLL, with $f_f = 23Hz$ and $\zeta = 1.6.$, which gives $T_i = 22.14ms$ and $K_p = 0.93$

All the three plots illustrate the performance of the PLL under different parameter configurations. In the uppermost plot, the red line represents the angle reference, while the yellow line represents the angle of the capacitor voltage. A smaller angle error, indicated by a reduced discrepancy between the yellow and red lines, indicates a more desirable performance of the PLL. The lower plot showcases the angle error represented by the blue curve. Specifically, Figure 4.10 demonstrates the simulation results of the PLL with $K_p = 2.85$ and $T_i = 7.27$ ms. Similarly, Figure 4.11 exhibits the simulation outcomes with $K_p = 1.63$ and $T_i = 12.73$ ms. Finally, Figure 4.12 presents the simulation outcomes achieved with $K_p = 0.93$ and $T_i = 22.14$ ms

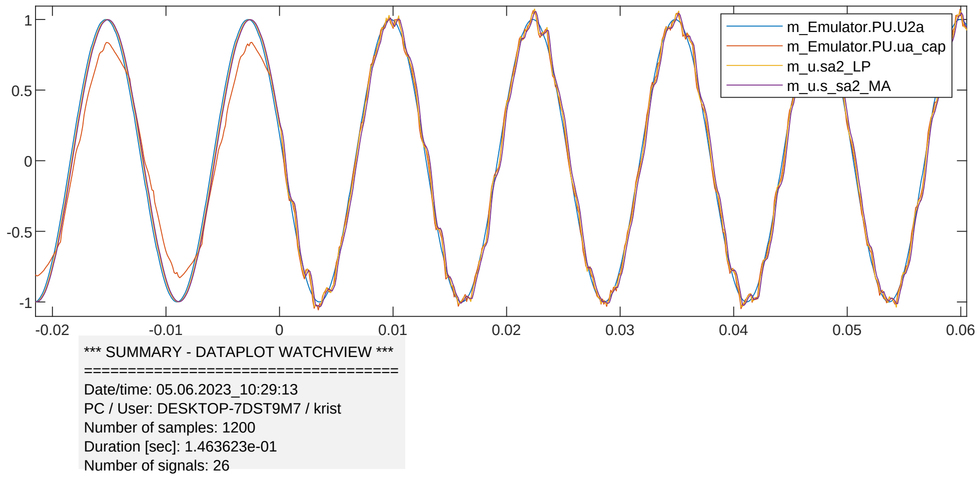
The analysis of the simulation results depicted in Figure 4.10, Figure 4.11, and Figure 4.12 reveals notable observations. Firstly, Figure 4.10 exhibits the most rapid response time among the three cases. As the proportional gain K_p decreases and the integral time constant T_i increases, the response time noticeably increases. This phenomenon is particularly evident in Figure 4.11, which displays the next best response time and angle difference. However, it is worth noting that Figure 4.12 demonstrates the least desirable performance in terms of the angle difference, with results indicating more pronounced tracking errors.

4.2.3 Testing the model in PESC control platform

To begin, the simulation results for the starting state with the breaker open are presented in Figure 4.15. For each scenario, a consistent set of parameters is observed and recorded. These parameters, visible in Figure 4.15 and applicable to all plots, are presented in the following order from top to bottom: The reference angle and the angle of the capacitor voltage, u_C , The grid voltage, u_2, the converter voltage, u_1, the converter current in the direct axes, i_{1d}, the converter current in the quadrant axes, i_{1q}, the converter current in abc-three-phase representation, i_{1a}, i_{1b} and i_{1c}, the active dampning current in abc-three-phase representation, the active dampening and the angle deviation.

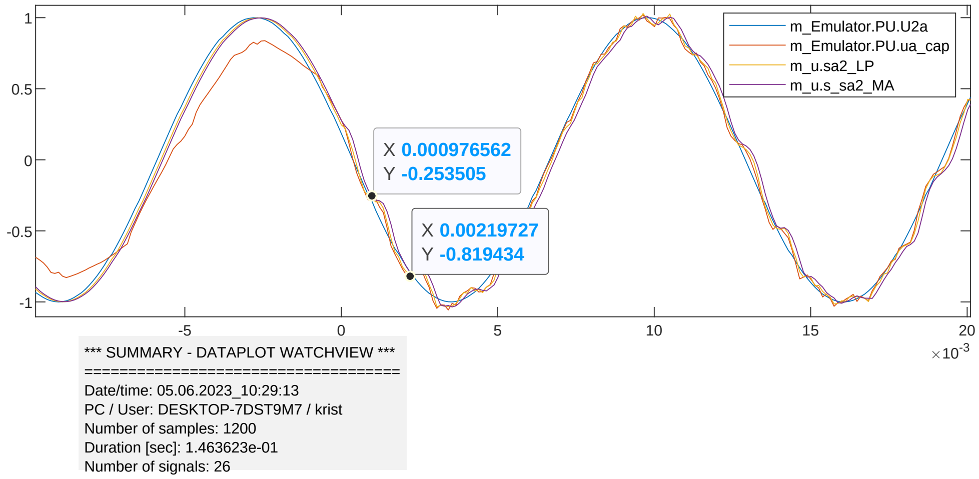
The next simulation investigates the impact of closing the breaker and is depicted in Figure 4.16. It is evident that the closure of the breaker introduces some disturbances to the current and voltages within the system. Consequently, a necessary increase in active dampening is required to mitigate these disturbances effectively. Notably, the angle deviation diminishes to approximately zero, indicating a successful connection between the grid and capacitor. In this simulation, the active dampening is configured with a value of rD = 2 p.u., ensuring enhanced stability and improved performance.

To evaluate the impact of active dampening, a simulation is conducted without the utilization of active dampening, as presented in Figure 4.17. The absence of active dampening makes the small deviations more noticeable, thereby highlighting their significance within the system.



```
*** SUMMARY - DATAPLOT WATCHVIEW ***
===================================
Date/time: 05.06.2023_10:29:13
PC / User: DESKTOP-7DST9M7 / krist
Number of samples: 1200
Duration [sec]: 1.463623e-01
Number of signals: 26
```

Figure 4.13: a zoom up picture of the grid voltage and a closeup of the oscilation occuring after the breaker is closed.



```
*** SUMMARY - DATAPLOT WATCHVIEW ***
===================================
Date/time: 05.06.2023_10:29:13
PC / User: DESKTOP-7DST9M7 / krist
Number of samples: 1200
Duration [sec]: 1.463623e-01
Number of signals: 26
```

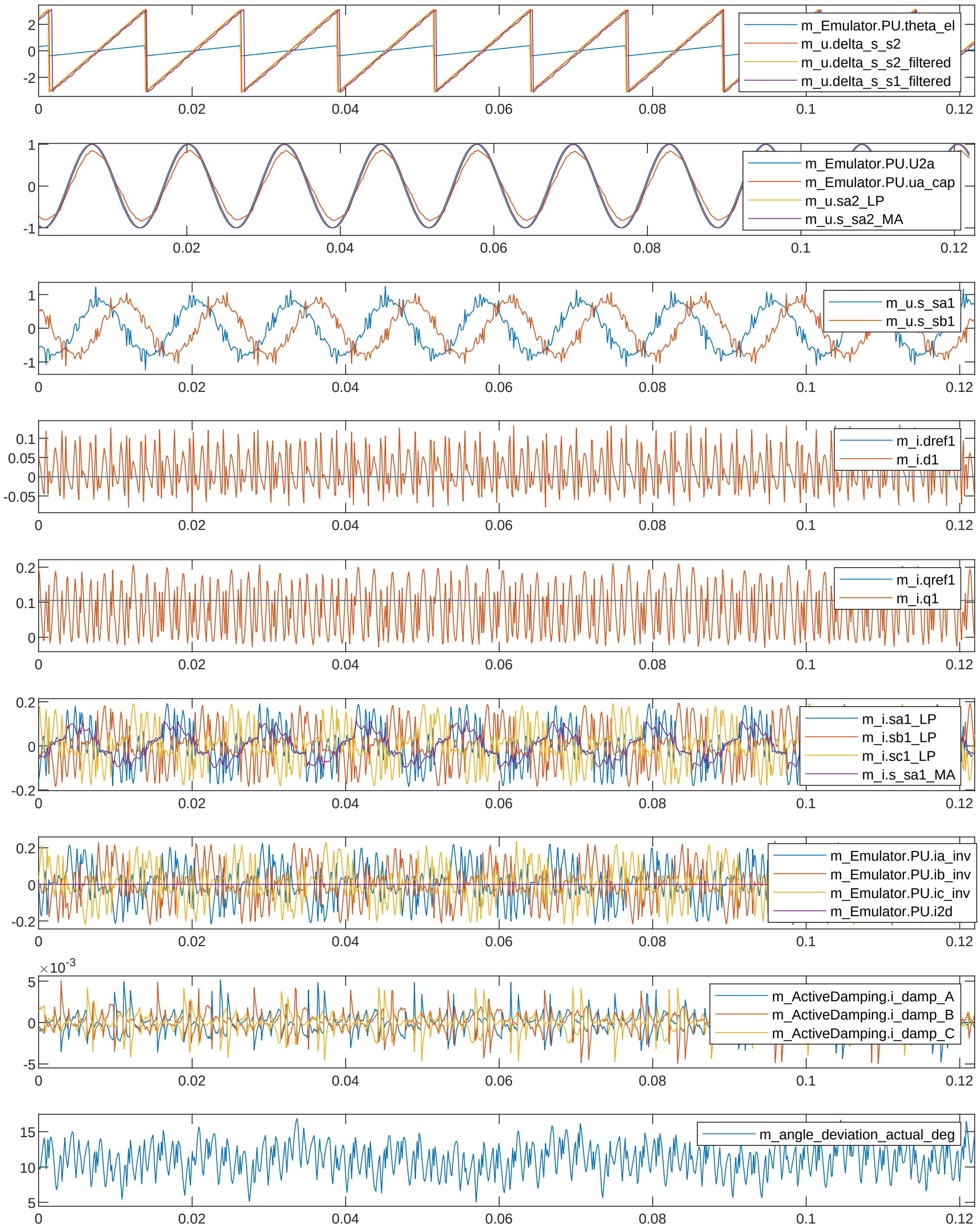
Figure 4.14: A closeup on how the frequency of the oscillation is measured.

The biggest difference observed is the presence of oscillations in the grid voltage after closing the breaker. These oscillations are attributable to frequencies within the system. Given that this system is not interconnected with a large and complex network, it is likely that the frequency arises from the switches in the converter. Figure 4.13 shows a closeup of the grid voltage were the oscillations are more significant. The frequency of the oscillation is measured in Figure 4.14 where it is calculated:

$$\frac{1}{0.00219727 - 0.000976562} = 819.1967Hz$$

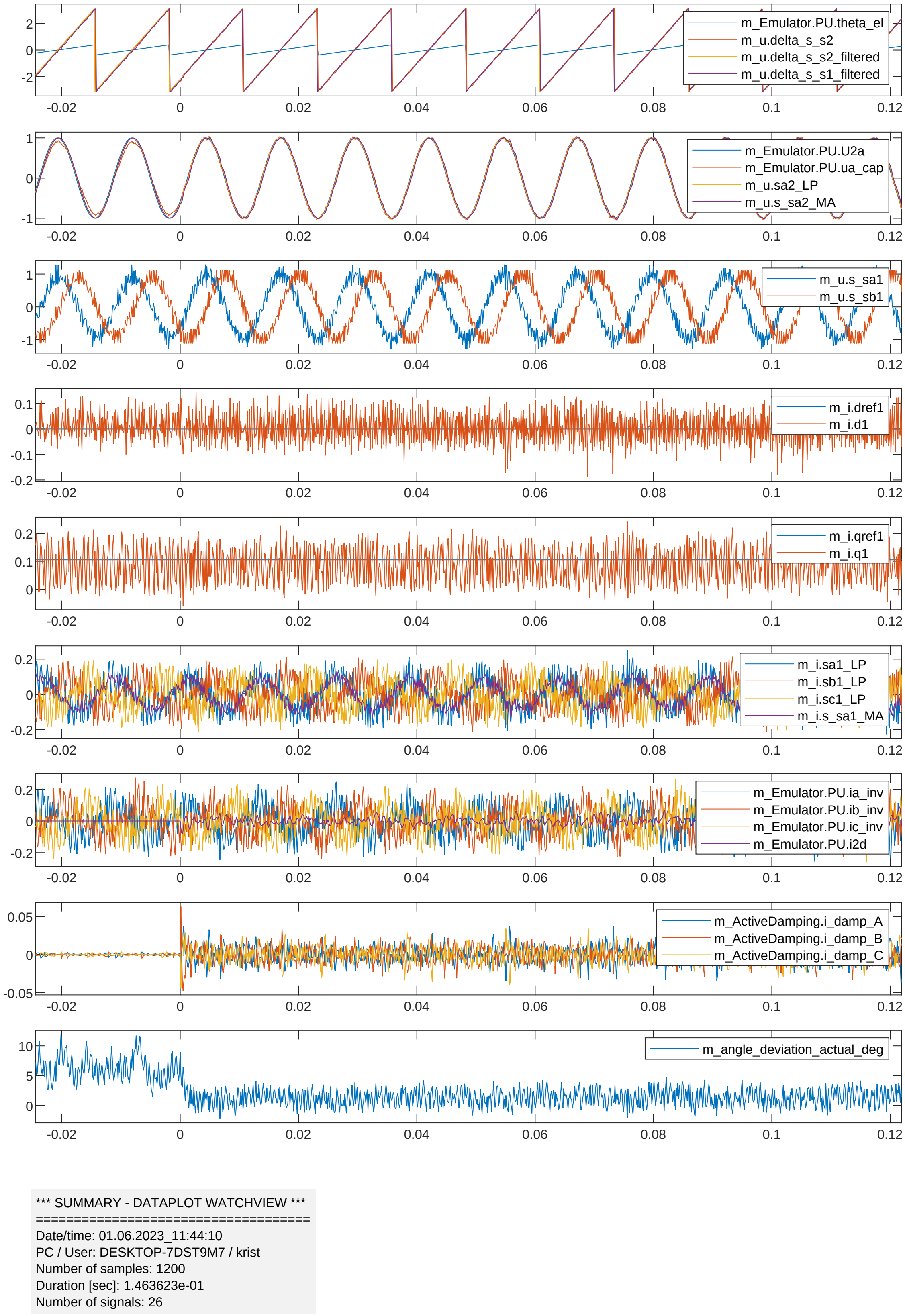
This is discussed more later in section 5.2.

The last simulation shows an increase in the load. To see how the model act with a load, Figure 4.18 shows the simulation. The load change is a change of i_{dref} from 0 to 0.6.



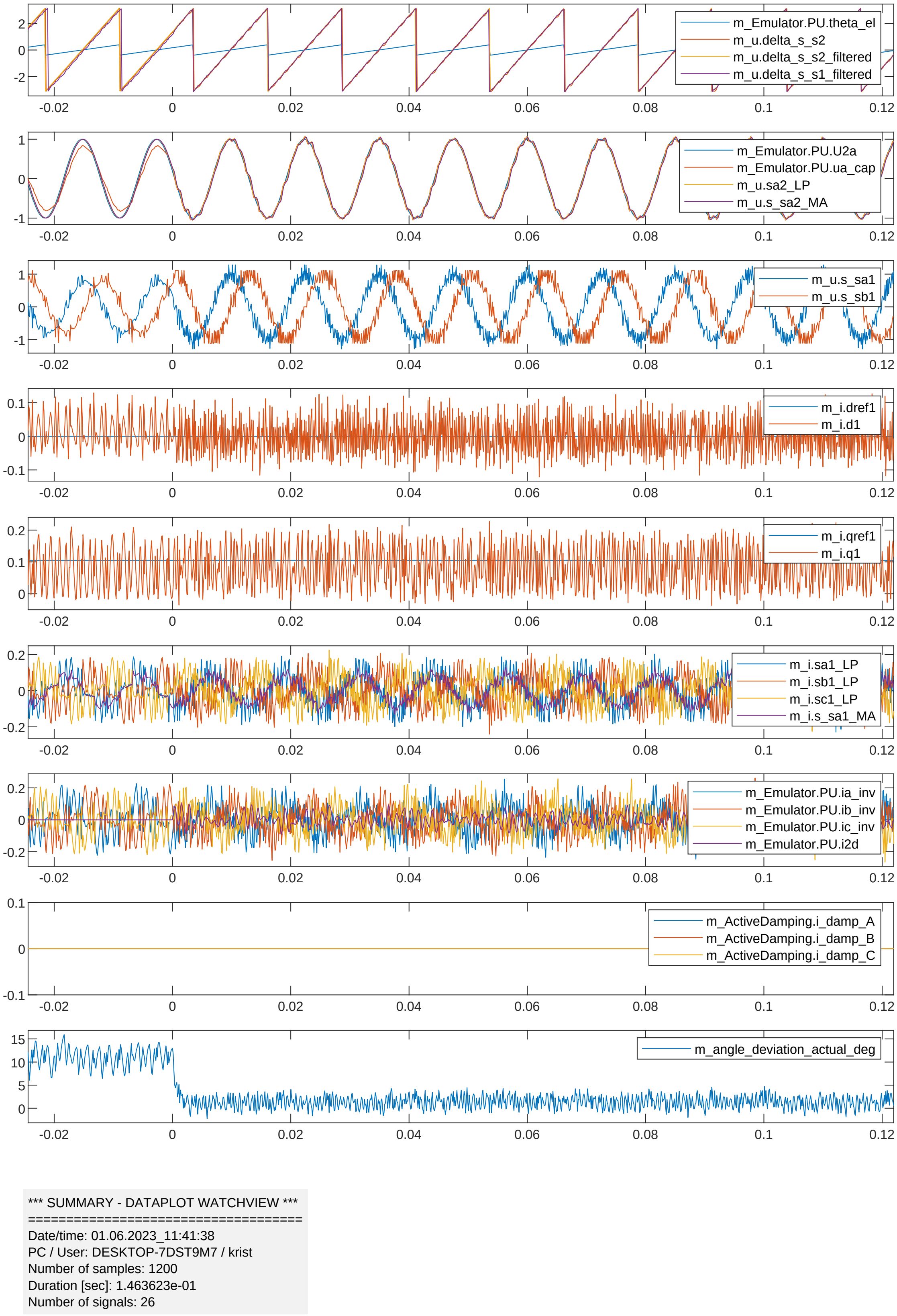
```
*** SUMMARY - DATAPLOT WATCHVIEW ***
===================================
Date/time: 01.06.2023_11:34:48
PC / User: DESKTOP-7DST9M7 / krist
Number of samples: 1000
Duration [sec]: 1.219482e-01
Number of signals: 26
```

Figure 4.15: Simulation with PESC controller of the starting state with the breaker open.



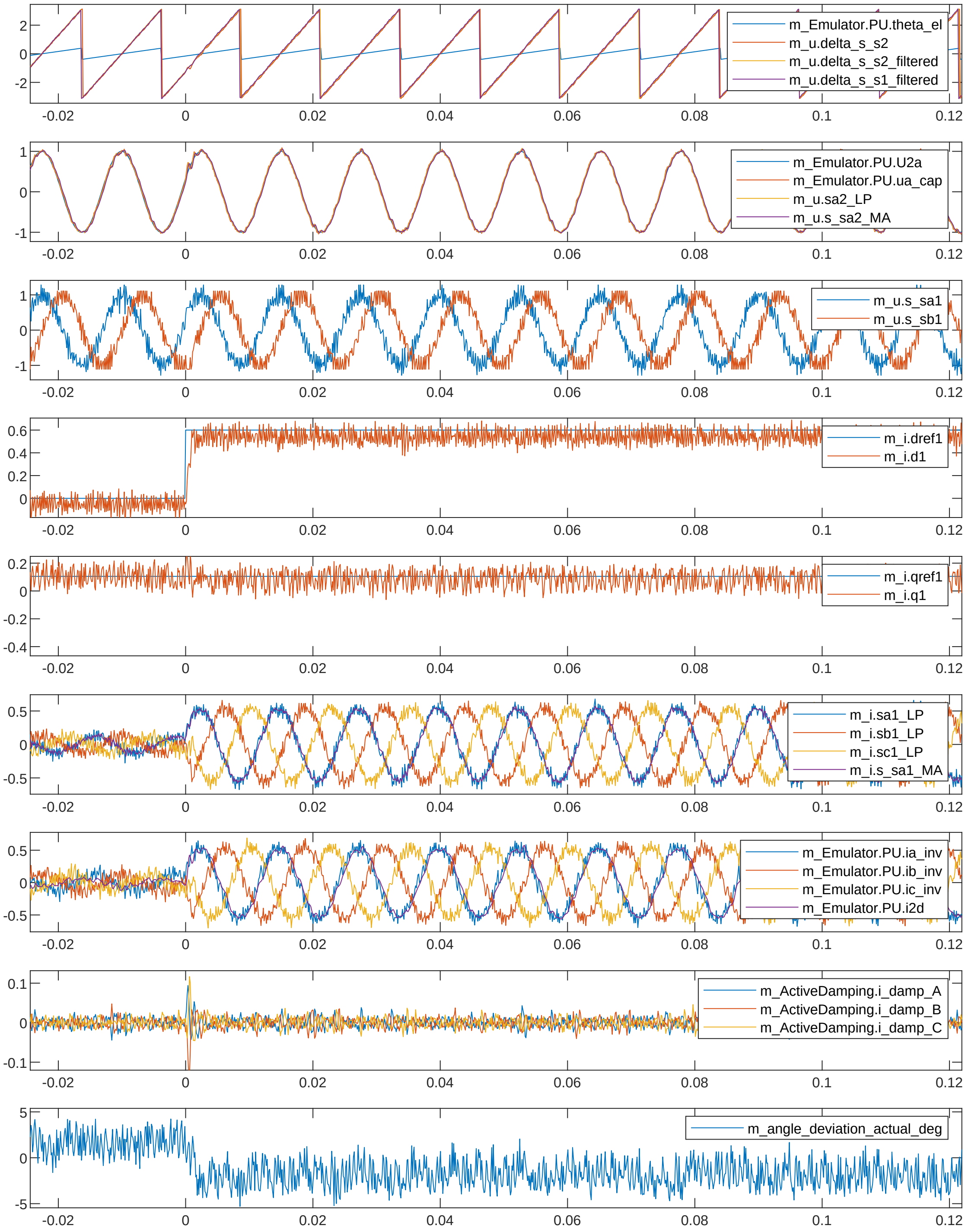
```
*** SUMMARY - DATAPLOT WATCHVIEW ***
==================================
Date/time: 01.06.2023_11:44:10
PC / User: DESKTOP-7DST9M7 / krist
Number of samples: 1200
Duration [sec]: 1.463623e-01
Number of signals: 26
```

Figure 4.16: Simulation with PESC controller of the starting state when the breaker close including the active dampening with a $r_D = 2 p.u.$ and Iqref = yc=1/xc=0.1 pu.



```
*** SUMMARY - DATAPLOT WATCHVIEW ***
==================================
Date/time: 01.06.2023_11:41:38
PC / User: DESKTOP-7DST9M7 / krist
Number of samples: 1200
Duration [sec]: 1.463623e-01
Number of signals: 26
```

Figure 4.17: Simulation with PESC controller of the starting state when the breaker close without active dampening, Iqref = yc=1/xc=0.1 pu.



```
*** SUMMARY - DATAPLOT WATCHVIEW ***
===================================
Date/time: 07.06.2023_15:32:12
PC / User: DESKTOP-7DST9M7 / krist
Number of samples: 1200
Duration [sec]: 1.463623e-01
Number of signals: 26
```

Figure 4.18: Simulation with PESC controller, the breaker is closed. Active dampening is included and a load change for Idref from 0 to 0.6 occurs at t=0.

Chapter 5

Discussion of results

5.1 Bode-plot

In this section, the Bode-plot for the PLL in the exact model and the PESC control platform are analysed. The transferfunction used for the Bodeplots is described in the theory given in Equation 2.50.

5.1.1 Exact Simulink model

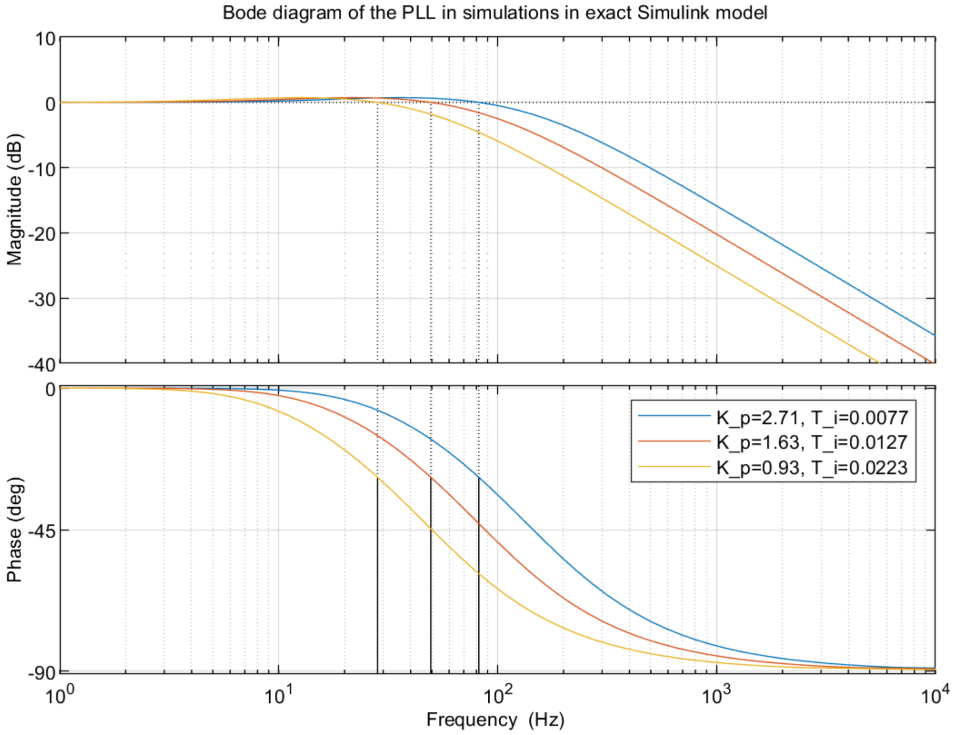


Figure 5.1: Bode-plot of the values in PLL used for the exact model.

To obtain a clearer understanding of how well the parameters fit the PLL, the Bode plot is analyzed. The process begins by plotting the Bode plot of the transferfunction for the exact model, as depicted in Figure 5.1. The corresponding parameter values are displayed in Table 4.3. In the power system, the nominal frequency is set to 60 Hz.

The graph consists of three lines: blue, red, and yellow. The blue line represents the values with $K_p = 2.71$ and $T_i = 7.7$ ms, the red line corresponds to $K_p = 1.63$ and $T_i = 17.7$ ms, and the yellow line represents $K_p = 0.93$ and $T_i = 22.3$ ms. From the simulations it was observed that a higher value of K_p leads to better performance. The Bode plot demonstrates that the cutoff frequency increases with an increasing value of K_p. The PLL functions as a first-order filter, and the crossover frequency marks the point where the incoming signal's amplitude no longer rises or falls significantly in higher frequencies. In this case, the crossover frequency cannot be lower than 60 Hz, which is the nominal frequency representing the fundamental frequency of the grid. Conversely, if the crossover frequency is too high, unwanted frequencies can pass through and potentially damage the components.

When plotting the transfer functions for the PLL in the Exact model, it was observed that the slowest response occurred with the parameters $K_p = 0.93$ and $T_i = 0.0223$. Referring to the Bode diagram in Figure 5.1, it is observed that parameters for the PLL must be chosen carefully. This is due to the cutoff frequency being lower than 60 Hz, the nominal frequency. Unfortunately, filtering out the nominal frequency is an undesirable outcome.

5.1.2 PESC control platform simulations

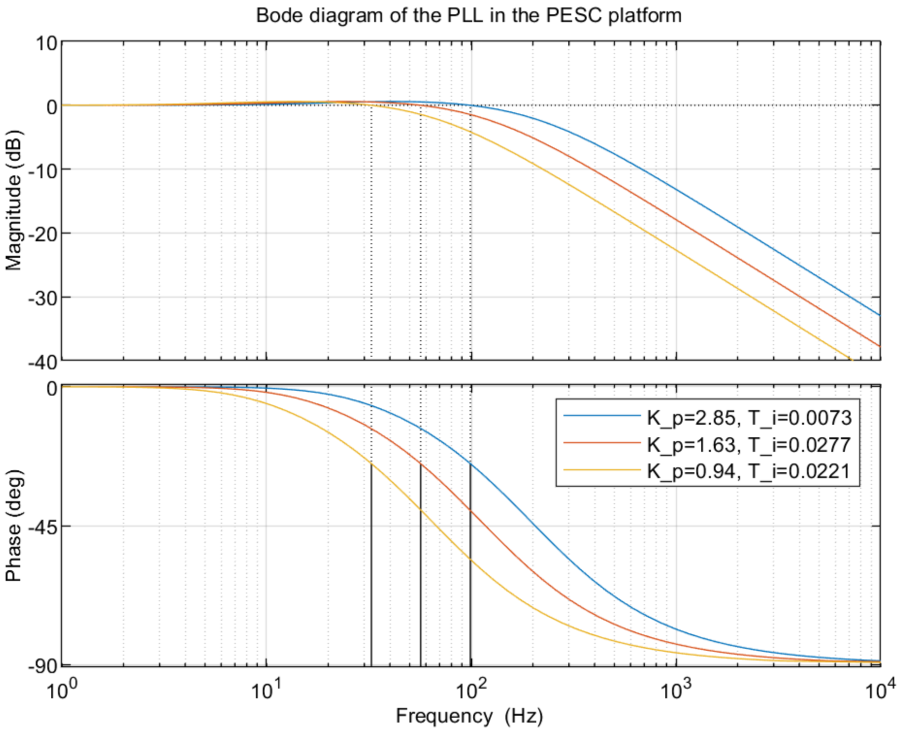


Figure 5.2: AFF plot of the values in PLL used for the PESC controller.

The Bode diagram in Figure 5.2 illustrates the simulation results in the PESC control platform, showcasing various parameter values. The blue graph corresponds to $K_p = 2.85$ and $T_i = 7.3$ ms, the red graph represents $K_p = 1.63$

and $T_i = 12.8$ ms, and the yellow graph signifies $K_p = 0.94$ and $T_i = 22.1$ ms.

Similar to the exact model, the simulation with the highest K_p value gives fastest response time. Analysing Figure 5.2, it is noticeable that these parameters yield the highest cutoff frequency, approximately 100 Hz. The red graph has a cutoff at 58 Hz, and the yellow graph has an approximate cutoff of 31 Hz. Considering that the nominal frequency for the grid is 78.8 Hz, it is crucial to ensure that the cutoff frequency remains within this range. Thus, the chosen parameters for the rest of the simulations are $K_p = 2.85$ and $T_i = 7.3$ ms.

In the case of the exact model, the fundamental frequency is lower than for the PESC control platform. The f_f and ζ need to be chosen to be lower is to get the same K_p and T_i as in the PESC simulations. This makes sense, since with higher frequency, higher dampening ratio is required to stabilize the frequency. With the higher frequency as for the PESC control platform, it is achieved a higher cut off frequency.

If the cutoff frequency is too high, higher harmonics have a chance to pass through as unwanted frequencies. On the other hand, if the cutoff frequency is too low, it filters out the nominal frequency, which is also undesired as power swings are intended. Hence, the conclusion is that the K_p value is desired to be as high as possible, but it is crucial to ensure that unwanted frequencies do not slip through, since the power electronics are vulnerable to disturbances.

5.2 Active dampening

By conducting a comparison between the simulations with and without active dampening, it becomes evident that the inclusion of active dampening eliminates the voltage oscillation. When the converter is connected to the grid, numerous disturbances can arise, influencing the frequencies, output voltages, and currents. Switching breakers, among other factors, can generate transients in voltage or current. To mitigate the harmonics originating from the grid and prevent component damage, active dampening is incorporated into the model.

However, if some of the frequencies coincide with the resonance frequency, the oscillation persists. Figure 4.16 illustrates the impact of active dampening. At time = 0, the breaker connected to the grid is closed, and the converter is linked to the grid. A noticeable oscillation is observed in the grid-side voltage. The simulation done in the PESC controller platform is not connecting to a real-world power network consisting of a lot of different components and a complex system. It is likely that the observed oscillation arises from the converter's switching frequency. If the frequency aligns with the resonance frequency of the LCL filter, an oscillation occurs. If this occurs only once, the oscillation will eventually cease. However, as the oscillation persists, it is probable that the frequency regularly coincides with the resonance frequency. To verify if the oscillation corresponds with the resonance frequency, a comparison is made between the ressonance frequency of the LCL-filter and the frequency of the oscillation occurred on the voltage. From Figure 4.14 the frequency of the oscillation is measured and calculated to 819 Hz. The resonance frequency of the LCL filter is calculated using Equation 5.1[14].

$$\omega_{res} = \sqrt{\frac{(L_1 + L_2)}{L_1 L_2 C_f}} \tag{5.1}$$

where L_1 represents the inductance on the inverter side, L_2 represents the inductance on the grid side, and C_f

represents the capacitance of the LCL-filter. The parameters used in the PESC control platform are displayed in Table 4.5. Substituting the values into Equation 5.1, the resonance frequency of the LCL-filter is calculated as 992 Hz. The deviation of 173 Hz could be attributed to measurement errors. The accuracy of the measurements and the imprecise determination of the bottom of the wave could contribute to the measurement errors. Despite the slight discrepancy in frequencies, it can be concluded that the frequencies of the harmonics resulting from the absence of active dampening closely resemble the resonance frequency of the LCL filter.
Reference frequency and actual frequency are also dependent on dampening[10].

Chapter 6

Conclusions

The AFE, connected to an LCL filter, is simulated in Smulink and analyzed. Two main methods for controlling the power electronic IGBT switches are pulse-width modulation (PWM) and hysteresis control. In this Master's thesis, the AFE is implemented using a hysteresis controller with a grid-following control scheme. The frequency and voltage are regulated by conventional synchronous machines, making the grid-following scheme adjusting for the active and reactive power based on the given voltage and frequency in the power system. Additionally, it is convenient to test the controller for grid-forming scheme and with pulse-width modulation to align with the modern paradigm of renewable energy sources. Furthermore, the PLL in the system was also tested. The emulated model for the PESC control platform has been successfully developed. Simulations play a crucial role in evaluating the performance of the control platform and analyzing its response under different operating conditions. Functions such as the phase-locked loop (PLL) and the effect of active damping are tested. Hence, the conclusion for the PLL is that the K_p value is desired to be as high as possible, but it is crucial to ensure that unwanted frequencies do not slip through, since the power electronics are vulnerable to disturbances. To delete the harmonics coming from the grid and avoid destroying the components, active dampening is included in the model.

Ultimately, the successful implementation of this FPGA-based control platform will pave the way for improved motor drive applications and potentially other domains that require real-time control and high-performance processing. The FPGA simulation model provide with a fast simulation to achieve real-time. This is highly required in design procedures, for neglecting faults, big costs etc. An important thing to consider for FPGA is the numeric representation and the scaling. The advantage of the Pu model is the broader application for the connected converter representation, as there is no need for different models for the higher rated voltage application and the lower voltage applications when the pu modelling is used.

6.1 Further work

Based on the work presented in this future work is proposed:

- In this work, a hysteresis current controller on firmware was used for the inner control loop. An alternative approach could be to move the current controllers into software domain, and replace the hysteresis controller with voltage-controlled PWM-based operation.
- The control system should be complemented with more functionality in the outer control-loop such as DC-link voltage regulation and droop control.
- When the basic functionalities are in place, looking into auxiliary services such as selective harmonic elimination is relevant.
- After the AFE firmware and software control system has been sufficiently tested on the PESC control platform, the next step is further testing using a physical experimental setup.

Bibliography

[1] Frede Blaabjerg et al. “Overview of Control and Grid Synchronization for Distributed Power Generation Systems”. In: *IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS* 53.5 (2006).

[2] Amir Arasteh, Lorenzo Zeni, and Nicolaos A.Cutululis. “Fault ride through capability of grid forming wind turbines:A comparison of three control schemes”. In: *IET Renew. Power Gener.* 16.1866-1881 (2022).

[3] Robert Rosso et al. “Grid-forming Converters: Control Approaches Grid-synchronization, and Future Trends-A Review”. In: *IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS* 2 (2021).

[4] Aravinda Perera et al. *A Design Method of an Embedded Real-Time Simulator for Electric Drives using Low-Cost System-on-Chip Platform*. 2021.

[5] *Conversation with Professor Roy Nilsen*. Took place 21 march 2023.

[6] Anders Moldskred. “FPGA-Based Real-Time Emulator of Permanent Magnet Synchronous Motor”. Master Thesis. Norwegian University of Science and Technology, Dept. of Electric Power Engineering, June 2020.

[7] Roy Nilsen. *TET4120 Electric Drives*. NTNU, Deparment of Electric Energy. Spring 2018.

[8] R. Sureshkumar. *10.001: Numerical Solution of ordinary Differential Equations*. Accessed: 07.05.2023. January 15, 1997. URL: http://web.mit.edu/10.001/Web/Course_Notes/lec24.pdf.

[9] *Conversation with Professor Roy Nilsen*. Took place 25 of may 2023.

[10] Jens G. Balchen, Trond Andresen, and Bjarne A. Foss. *Reguleringsteknikk*. 6th ed. Institutt for teknisk kybernetikk, NTNU, 2016.

[11] *FPGA Fundamentals: Basics of Field-Programmable Gate Arrays*. Last updated: 03.05.2023 Accessed: 04.06.2023. URL: https://www.ni.com/en-no/shop/electronic-test-instrumentation/add-ons-for-electronic-test-and-instrumentation/whatis-labview-fpga-module/fpga-fundamentals.html.

[12] Viljar Fjellanger. “On-line voltage estimation during sensorless control of an induction machine drive”. Master Thesis. Norwegian University of Science and Technology, Dept. of Electric Power Engineering, June 2019.

[13] Roy Nilsen. *Technical report: System Introduction*. May 25, 2021.

[14] M. Ben Sa¨ıd-Romdhanea et al. “Simple and systematic LCL filter design for three-phase grid-connected power converters”. In: *Mathematics and Computers in Simulation* 130 (2016), pp. 181–193. URL: https://www.sciencedirect.com/science/article/pii/S0378475415002116.

Appendices

A Appendix A-simulink models

A.1 Simplified model in simulink

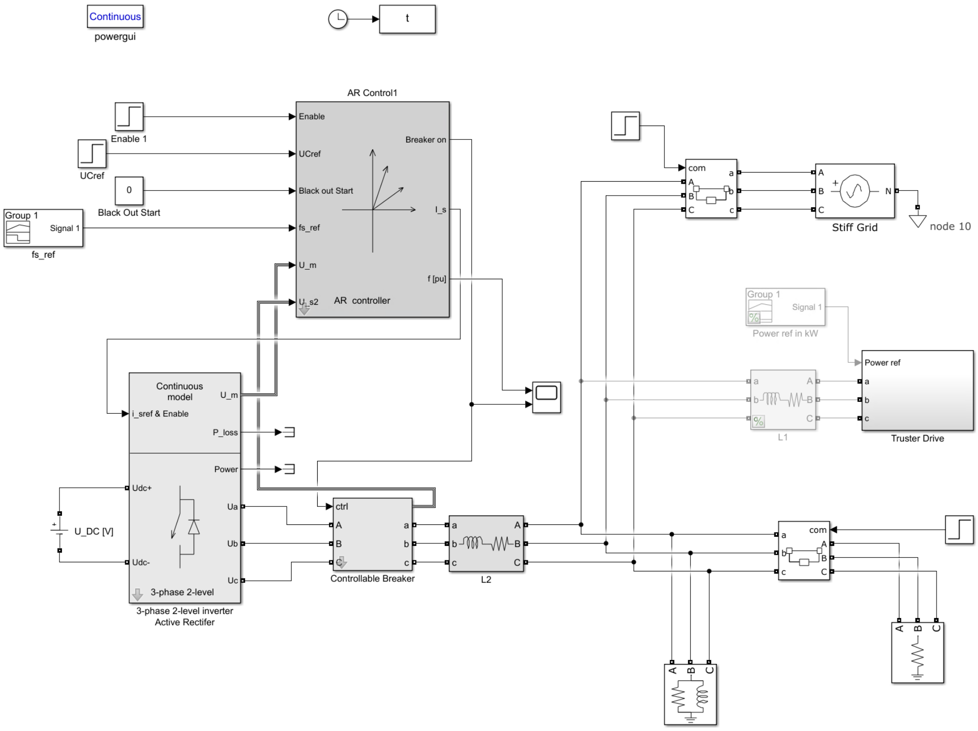


Figure A.1: Top level of the simplified model of the AEF

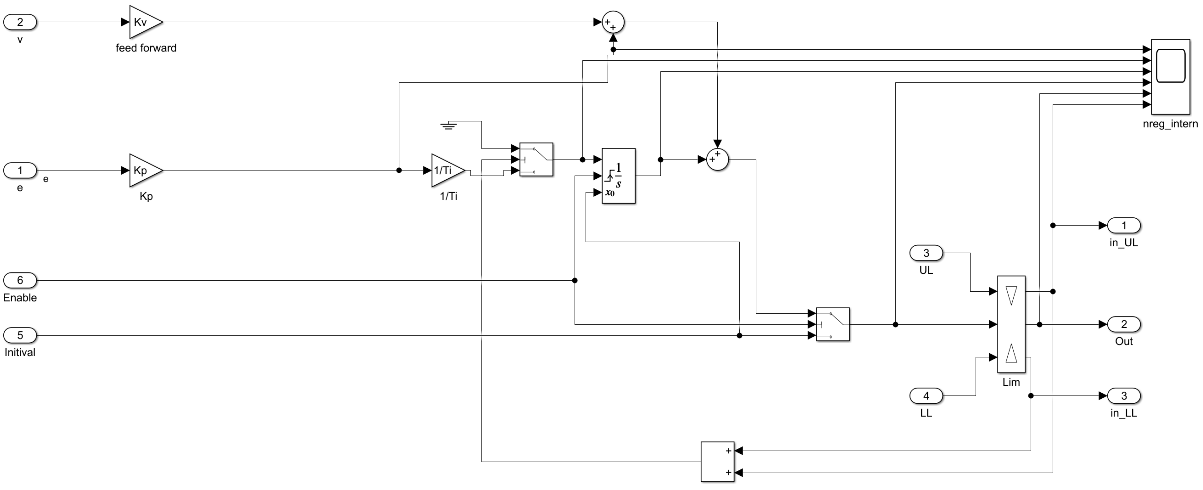


Figure A.2: the PLL of the AEF

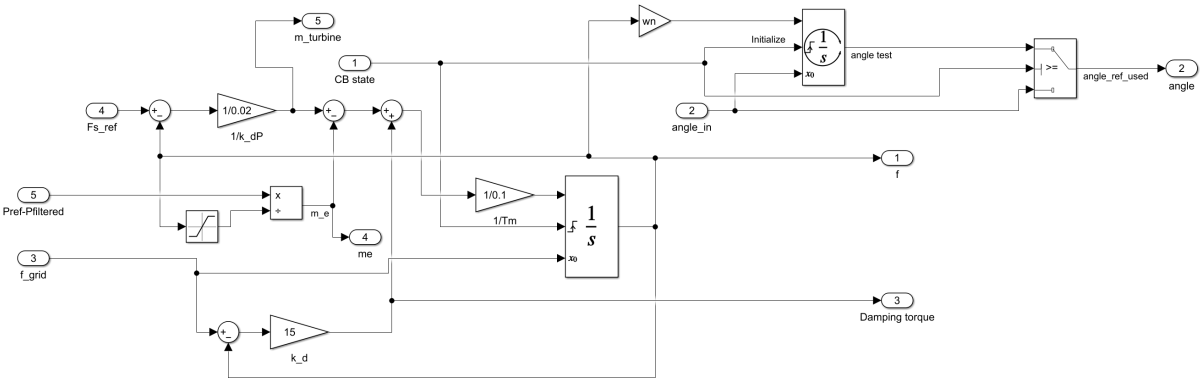


Figure A.3: The Controller of the AEF implemented as the virtual synchrounous machine

A.2 emulated continious model of the LCL filter

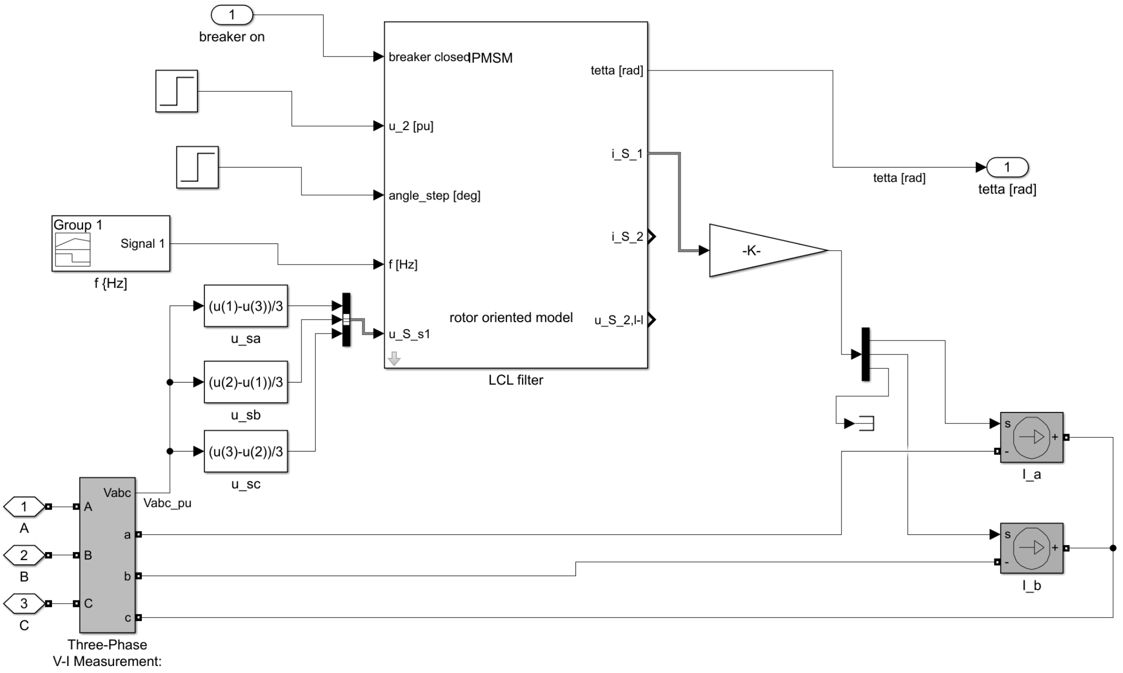


Figure A.4: Top level of the simplified model of the AEF

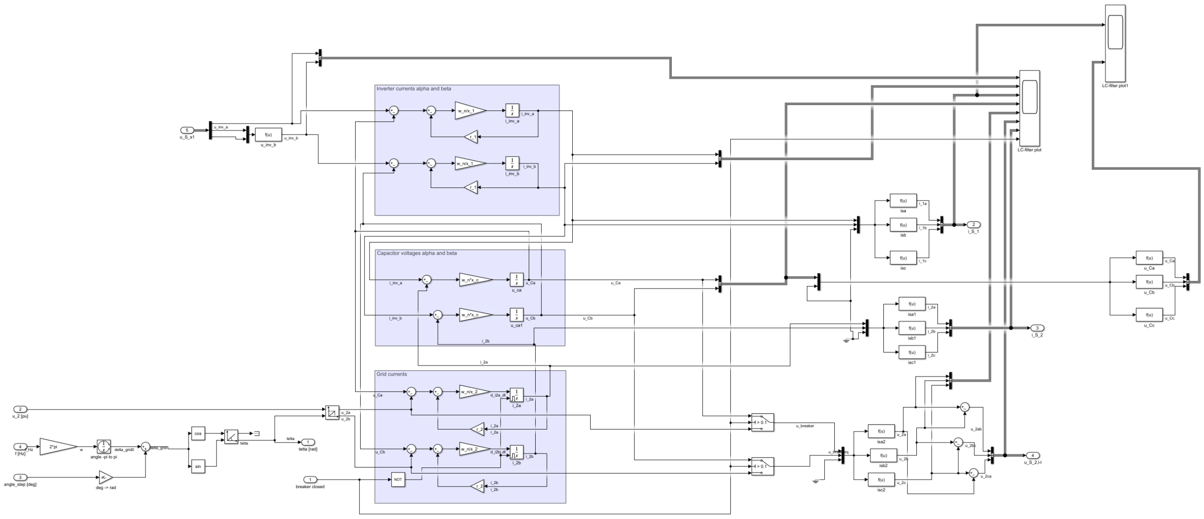

Inverter currents alpha and beta
Capacitor voltages alpha and beta
Grid currents
LC-filter plot
LC-filter plot1

Figure A.5: simulink blocks of the LCL filter

A.3 Nøyaktig modell

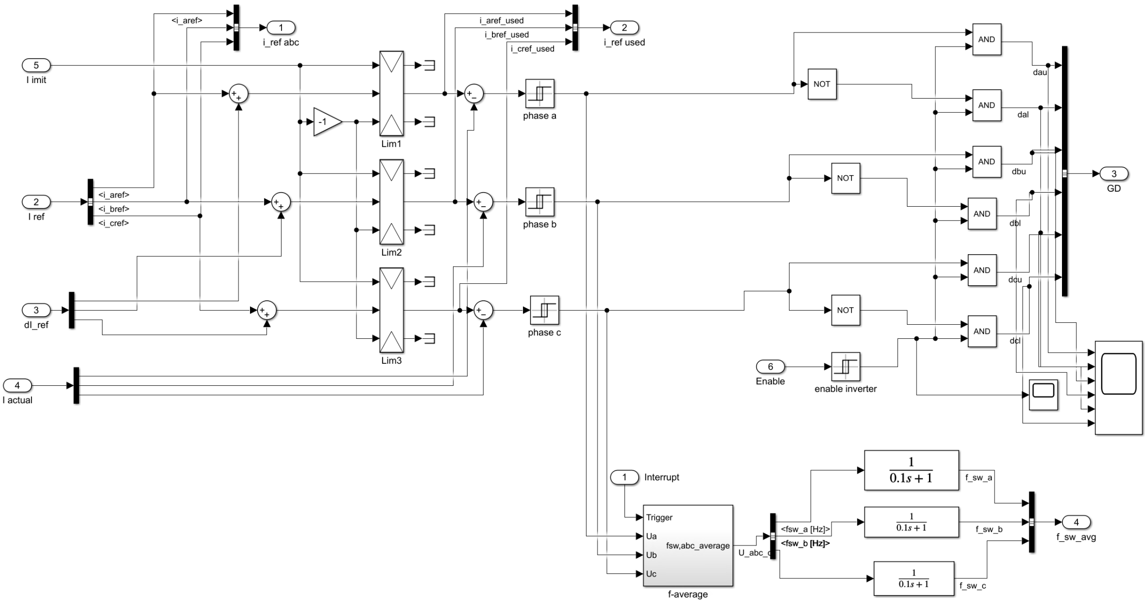


Figure A.6: Hysterises controller

Preface and acknowledgements

This report is the final work of my Master's degree at ”Electric Power engineering” at the department of Electrical Energy at NTNU, completed in spring 2023. The topic was provided by the resource group, PESC, and it immediately captured my interest, and I was fully prepared to tackle the assigned task. The background information provided by the study program was limited, making the process of acquainting myself with a completely new topic quite challenging.

During the specialization project in autumn 2022, I conducted a comprehensive literature review on the subject of a virtual synchronous machine. However, as I transitioned to my Master's thesis, a new supervisor took on the role, providing me with guidance. This resulted in a change of research topic and the need to delve into an entirely new theme. Throughout the process, I received assistance from my new supervisor, Professor Roy Nilsen.

I would like to express my sincere gratitude to my supervisor for his patience and expertise. Without him, this thesis would not have been possible. I would also like to extend my thanks to Thomas S. Haugan, who developed the necessary programming for the FPGA, and to Professor Roy Nilsen for sharing his valuable Simulink models. Lastly, I would like to express my special appreciation to the members of the Corner Office, with whom I shared both joyous and challenging moments. Through laughter and motivational words, they have been a source of support throughout my journey.

Abstract

In the emerging field of renewable energy, new control methods are being considered and have been researched for quite some time. Many studies can be found on the converter control techniques. This thesis is considering the AFE connected to a LCL-filter. For the development on these converters, the traditional PC computer simulations of the models take quite some time, and therefore the emulated real time simulation have been used for research and design. By employing FPGA implementation, tasks can be executed in parallel manner, enabling real-time simulations. This thesis is a part of a project at the Department of Electrical Energy at NTNU to develop a real-time FPGA-based emulator of a AFE by using the PESC control platform developed by NTNU.

During the development of the Emulated Real-Time Simulator, a simulation process is employed using Matlab, Simulink to analyze and evaluate how a AFE connected to a LCL-filter behaves. First, a simplified model where the AFE is represented by an ideal current source is studied. Followed by a more realistic model, where the converter is represented the power electronic insulated gate bipolar resistors (IGBT) switches, the result is a more realistic and complex model. To further enhance the PESC control platform, an emulated model of the LCL filter is built using the mathematical model derived from the equations. The emulated model developed aims to capture the dynamic reactions of the LCL filter accurately. Further the emulated mathematical model is discretezied. These discretized equations serve as the foundation for constructing the FPGA-based emulator, where the FPGA logic is developed in the Xilinx system generator for DSP add on in the simulink library. Finally, the emulated model and a project file is configured with the PESC control platform, where the emulated model is ready to be tested. Simulation of different scenarios is done within the PESC control platform. The effect of the Active dampening is discussed, where some oscillation can be observed. The frequency of the oscillations is measured to 819.1 Hz. This frequency is compared to the resonance frequency of the LCL filter. The resonance frequency is calculated to 992.4 Hz. Despite the slight discrepancy in frequencies, it can be concluded that the frequencies of the harmonics resulting from the absence of active dampening closely resemble the resonance frequency of the LCL filter. The AFF plot of the exact model and the PESC simulations are compared. The difference of the model is the fundamental frequency, it can be seen that the PESC platform operates with 78.8 Hz and the exact model operated with 60 Hz. f_f and ζ need to be scaled down to get the same K_p and T_i.

Further analyses with different grid forming control scheme need to be done for the model in the PESC platform, ass well as a model using PWM controller.

Sammendrag

I den voksende sektoren for fornybar energi har nye innovative kontrollmetoder blitt forsket på i lang tid. Denne rapporten tar for seg aktiv likeretter koblet til et kraftnett via et LCL-filter. I utviklingen av kraftelektroniske omformere tar tradisjonelle PC-simuleringer lang tid, og derfor har emulerte sanntidssimuleringer blitt brukt til forskning og design. Ved å bruke FPGA-implementering kan oppgaver utføres parallelt, noe som muliggjør sanntidssimuleringer.

Denne avhandlingen er en del av et prosjekt ved institutt for elektrisk energi ved NTNU for å utvikle en sanntids-FPGA-basert emulator av en aktiv likeretter ved bruk av PESC-kontrollplattformen utviklet av NTNU.

Under utviklingen av den emulerte sanntidssimulatoren er simuleringer i MATLAB Simulink gjennomført for å analysere og evaluere oppførselen til den aktive likeretteren. Først simuleres en forenklet modell der aktiv likeretteren representeres av en ideell strømkilde studert. Deretter følger en mer realistisk modell der kraftomformeren representeres av IGBT-brytere, noe som resulterer i en mer realistisk og kompleks modell.

For å forbedre PESC-kontrollplattformen ytterligere, bygges en emulert modell av LCL-filteret ved hjelp en matematisk modell som er utledet fra dynamikken til LCL-filteret. Målet med den utviklede emulerte modellen er å fange opp de dynamiske egenskapene til LCL-filteret.

Den emulerte matematiske modellen blir deretter diskretisert. Disse ligningene fungerer som grunnlaget for konstruksjonen av FPGA-basert emulator, der FPGA-logikken utvikles i Simulink-biblioteket Xilinx System Generator for DSP.

Til slutt blir den emulerte modellen integrert og kompilert med PESC-kontrollplattformen, der den emulerte modellen er klar til å bli testet. Simulering av ulike scenarier utføres innenfor PESC-kontrollplattformen.

Effekten av aktiv demping blir diskutert, der man kan observere hvordan dette påvirker resonansen mellom kraftnett og den aktive likeretteren. Frekvensen til oscillasjonene måles til 819,1 Hz. Denne frekvensen blir sammenlignet med resonansfrekvensen til LCL-filteret, som beregnes til 992,4 Hz. Til tross for den lille forskjellen i frekvensene, kan det konkluderes med at frekvensene til harmoniske svingninger som oppstår på grunn av fravær av aktiv demping i stor grad ligner på resonansfrekvensen til LCL-filteret.

Table of Contents

List of Tables

List of Figures

Abbreviations

- **AC** Alternate Current
- **AFE** Active Front End
- **CPU** Central Processing Unit
- **DC** Direct Current
- **DSP** Digital Signal Processing
- **ERTS** emulated real-time simulations
- **FPGA** Field Programmable Gate Array
- **IGBT** Insulated Gate Bipolar resistors
- **PEC** Power electronics converters
- **PLL** Phase Locked Loop
- **pu** Per Unit
- **PWM** Pulse Width Modulation
- **VSM** Virtual Synchronous Machine

Chapter 1

Introduction

1.1 Background and motivation

Due to the increase in renewable energy sources in the power grid, the natural control of the frequency and voltage coming from the big synchronous machines usually exist in the coal plants and fossile sources. The distributed power generation system, DPG, is required to actively take part in the power regulation of the grid [1]. [2] and [3] study different grid forming schemes and its performance on the power grid. This thesis focuses on the Active Front End (AFE) connected to an LCL-filter, which serves as a vital component in such control systems. Traditional PC-based simulations for converter development are time-consuming, necessitating the use of emulated real-time simulations (ERTS) for research and design purposes. By employing FPGA implementation, tasks can be executed concurrently, enabling real-time simulations. This methodology offers significant advantages, as it allows for the design and evaluation of control algorithms during the design process itself. The present thesis is part of a project undertaken at the Department of Electrical Energy at NTNU, aimed at developing a real-time FPGA-based emulator of an AFE using the PESC control platform developed by NTNU. Drawing upon the ERTS approach presented in [4], which focuses on a Permanent Magnet Synchronous Generator (PMSG), this thesis aims to emulate an AFE converter within a similar framework.

1.2 Scope of Work

During the development of the Emulated Real-Time Simulator, a comprehensive simulation process is employed using Matlab, Simulink to analyze and evaluate how an AFE connected to a LCL-filter behaves. First, a simplified model where the AFE is represented by an ideal current source is studied. Next, a more realistic model is studied, where the ideal current sources are replaced by a realistic model of the converter with power electronic insulated gate bipolar resistors (IGBT) switches, accounting for the influence of switching frequencies and associated ripple effects. The purpose of this simplified model is to gain a better understanding of the system's dynamics and functionality. The control method is a hysteresis controller using a grid-following scheme. The exact AFE model is simulated to study the behavior of the phase-locked loop (PLL). Further, a mathematical model of the LCL-filter connected to the AFE is developed, which aims to capture the dynamic reactions of the LCL-filter accurately. The emulated model is calculating the inverter current, grid current and capacitor voltage with mathematical equations. Further, the FPGA is developed with the Xilinx system generator add on blocks in Simulink. For implementation of the IP cores in the FPGA, the design is exported from System Generator to Vivado, where the IP core design can be synthesized into HDL code to configure with the FPGA-module. However, this is out of the scope for this thesis, as the IP cores will only be developed and tested in System Generator.

Finally, The emulated model and a project file are configured with the PESC control platform, where the emulated model is ready to be tested. Simulation of different scenarios is done within the PESC control platform. These scenarios include simulating the starting state with the breaker open, simulating the starting state with the breaker closed with and without active damping, and simulating the effect of a load increase. The simulations help evaluate the performance of the control platform and analyze its response to different operating conditions. Here different functions are tested, including the Phase locked loop (PLL) and the effect of active dampening.

1.3 Structure

This thesis report is divided into six main chapters. Their contents are as follows:

Chapter 2 presents the background information for this thesis. Starting with describing AFE and its usage. Further, describing the two main control methods, grid-following scheme and grid-forming scheme. The mathematical model of the LCL-filter connected to the AFE is derived. Additionally, fundamental concepts such as Euler's method, the Park-Clark transformation, the logic of the emulator, the PLL, active damping, and the hysteresis controller are explained.

Chapter 3 briefly describes the PESC control platform developed at NTNU. Including The implementation of the FPGA firmware.

Chapter 4 provides the simulations of a simple Simulink model, and a more exact model, as well as the simulation of the PESC controller.

Chapter 5 discusses the results provided from the Simulink simulaions, including a discussion on the PLL technique and active dampening.

Chapter 6 presents the conclusion and proposed further work.

Chapter 2

Background Theory

2.1 Active front-end with LCL-filter

Active front end (AFE), typically as Two level 3-phase converter, offers the advantage of controlling both the DC-link voltage and the power factor. The converter is built from three half-bridges, which consists of two insulated gate bipolar resistors (IGBTs) in each half bridge. This topology is dominant for the three-phase applications for line voltage below 1000 V. The AFE can function as a converter, from AC to DC, or as an inverter, converting DC to AC. The two main methods to control the power electronic IGBT switches, are either using a pulse-width modulation (PWM), which is out of the scope of the present thesis, or a hysteresis controller. In this master thesis, the AFE is implemented using the hysteresis controller with a grid-following control scheme.

However, due to the switching frequency of the IGBTs, the output signal produced may contain high-order harmonics. To mitigate this issue, an LCL-filter is integrated and connected to the AFE to reduce these undesired harmonics. The LCL-filter filters and adsorbs the harmonics to achieve the desired sinusoidal voltage waveform.

In the context of this thesis, the model consists of the AFE connected to an LCL-filter which is connected to the grid via a circuit break.

2.2 Control methods

Power systems generally require the balance and regulation of four variables: voltage, frequency, active power and reactive power. These variables play crucial roles in ensuring the stability and optimal functioning of the power grid.

In a conventional power system, the stability and control of voltage and frequency are achieved by interconnected synchronous machines. These machines, equipped with inherent inertia and kinetic energy, help regulate the system by responding to fluctuations and disturbances. By dynamically adjusting their output, they assist in maintaining a steady voltage and frequency within acceptable limits. The voltage and frequency in the power system have a direct link with the reactive and active power respectively. Thus, grid-following control enables power electric converters (PECs) to effectively manage their active and reactive powers based on the system's voltage and frequency conditions.

In converter-rich power systems, the reliance on PECs has grown significantly. Consequently, PECs are now being called upon to actively contribute to the stabilization of the power system. This shift is necessitated by the irregular nature of renewable energy sources and the decentralized nature of power generation. As renewable energy sources become more dominant, such as wind and solar, the power system looses its natural stabilisation from the synchronous generators. This change in the power system poses challenges in maintaining a stable grid with traditional control methods alone.

To address these emerging challenges, researchers and engineers are actively exploring and studying control schemes that can enhance the robustness and resilience of power systems. The focus is on developing strategies that enable power electronic converters to take on a more active role in frequency and voltage stabilization. These strategies, collectively known as grid-forming techniques, aim to provide PECs with the capability to establish and maintain the grid's fundamental parameters, such as voltage and frequency[2].

In this paper, the grid-forming method with the virtual synchronous machine scheme with hysteresis controller is applied. A PLL is also included in the model.

Grid-following converters operate as a current source, where the grid-following regulates the power injection and voltage by controlling the injected current (current from converter into grid)[3].

As the grid-following scheme controls power by adjusting the current, the grid-forming scheme regulates the voltage directly at its output terminal. There are several techniques to perform a grid-forming control scheme [2]. Examples of grid-forming methods that are discussed in literature are droop control, virtual synchronous machine, power synchronization control and distributed PLL-based. Where the fault ride through characteristics are studied in [2][3].

2.3 Mathematical model with LCL-filter

In order to implement the discrete model in the Field-Programmable Gate Array (FPGA), it is necessary to represent the LCL-filter, which is connected to the converter, using mathematical equations. The single-line diagram of the LCL-filter is depicted in Figure 2.1.

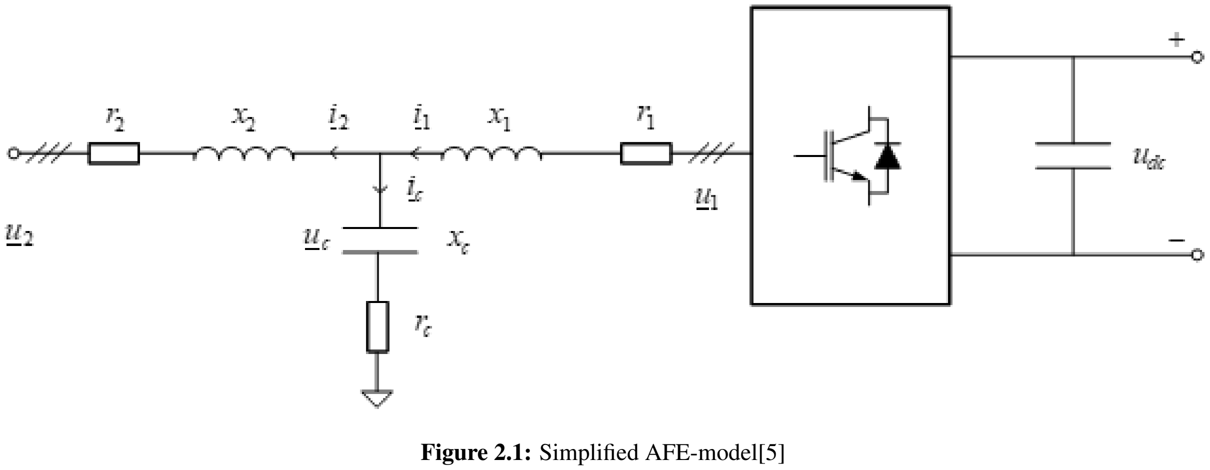


Figure 2.1: Simplified AFE-model[5]

For simplification purposes the capacitance resistance, r_C, is neglected, assuming $r_C = 0$. If a higher capacitance resistance is desired, it can be intercepted in the inverter resistance, r_1, or in the grid resistance,r_2[5].

The voltage at the converter for each phase, u_{1a}, u_{1b} and u_{1c}, is expressed by the equations below.

$$u_{1a} = r_1 \cdot i_{1a} + \frac{x_1}{\omega_N}\frac{di_{1a}}{dt} + u_{Ca} \tag{2.1}$$

$$u_{1b} = r_1 \cdot i_{1b} + \frac{x_1}{\omega_N}\frac{di_{1b}}{dt} + u_{Cb} \tag{2.2}$$

$$u_{1c} = r_1 \cdot i_{1c} + \frac{x_1}{\omega_N}\frac{di_{1c}}{dt} + u_{Cc} \tag{2.3}$$

In the given context, ω_N represents the nominal angular frequency. Furthermore, r_1 and x_1 denote the resistance and inductance on the converter side, respectively. The variable i_1 represents the current on the inverter side running through r_1 and x_1. Similarly, r_2 and x_2 is the resistance and inductance on the grid side, respectively, with i_2 indicating the current on the grid side running through r_2 and x_2. Then, the capacitance voltage for each phase, u_{Ca}, u_{Cb} and u_{Cc}, are derived as follows:

$$u_{Ca} = r_2 \cdot i_{2a} + \frac{x_2}{\omega_N}\frac{di_{2a}}{dt} + u_{2a}$$

$$u_{Cb} = r_2 \cdot i_{2b} + \frac{x_2}{\omega_N}\frac{di_{2b}}{dt} + u_{2b}$$

$$u_{Cc} = r_2 \cdot i_{2c} + \frac{x_2}{\omega_N}\frac{di_{2c}}{dt} + u_{2c}$$

Here, the parameters mentioned above yields, and u_2 represents the voltage on the grid side. Additionally, the current running through the capacitance is given in Equation 2.4.

$$i_C = \frac{1}{C} \cdot \frac{du_C}{dt} \tag{2.4}$$

It is known that $i_C = i_1 - i_2$ and $x_C = \frac{1}{C \cdot \omega_N}$. Equation 2.4 can then be rearranged for the three-phases as follows:

$$\frac{du_{Ca}}{dt} = \omega_N \cdot x_C \cdot (i_{1a} - i_{2a}) \tag{2.5}$$

$$\frac{du_{Cb}}{dt} = \omega_N \cdot x_C \cdot (i_{1b} - i_{2b}) \tag{2.6}$$

$$\frac{du_{Cc}}{dt} = \omega_N \cdot x_C \cdot (i_{1c} - i_{2c}) \tag{2.7}$$

For simplification, the equations for the three-phases is written in vector form, presented as follows:

$$\underline{u_1} = r_1 \cdot \underline{i_1} + \frac{x_1}{\omega_N}\frac{d\underline{i_1}}{dt} + \underline{u_C} \tag{2.8}$$

$$\underline{u_C} = r_2 \cdot \underline{i_2} + \frac{x_2}{\omega_N}\frac{d\underline{i_2}}{dt} + \underline{u_C} \tag{2.9}$$

$$\frac{d\underline{u_C}}{dt} = \omega_N \cdot x_C \cdot (\underline{i_1} - \underline{i_2}) \tag{2.10}$$

where the vectors are as stated below:

$$\underline{u_1} = \begin{bmatrix} u_{1a} \\ u_{1b} \\ u_{1c} \end{bmatrix}, \underline{u_2} = \begin{bmatrix} u_{2a} \\ u_{2b} \\ u_{2c} \end{bmatrix}, \underline{u_C} = \begin{bmatrix} u_{Ca} \\ u_{Cb} \\ u_{Cc} \end{bmatrix}, \underline{i_1} = \begin{bmatrix} i_{1a} \\ i_{1b} \\ i_{1c} \end{bmatrix}, \underline{i_2} = \begin{bmatrix} i_{2a} \\ i_{2b} \\ i_{2c} \end{bmatrix} \tag{2.11}$$

2.3.1 Three-phase to two-phase transformation

To simplify the model of the converter, the filter and grid components, the reference frame is changed from three-phase to three-phase. These are also known as $\alpha\beta$- or Clarke-transformation.

Transformation matrix

The Clarke-transformation converts three-phase abc-quantities to $\alpha\beta$-reference frame, where the transformation matrix is shown in Equation 2.12. Since the three-phase system is considered balanced, the γ - component is considered zero, and is neglected. The result of two-phase representation are scaled to be equal in magnitude to the original three-phases.

$$\begin{bmatrix} u_\alpha \\ u_\beta \end{bmatrix} = \frac{2}{3} \cdot \begin{bmatrix} 1 & \frac{-1}{2} & \frac{-1}{2} \\ 0 & \frac{\sqrt{3}}{2} & \frac{-\sqrt{3}}{2} \end{bmatrix} \cdot \begin{bmatrix} u_a \\ u_b \\ u_c \end{bmatrix} \tag{2.12}$$

For the amplitude-invariant transformation for a balanced three-phase system, the resulting system gives a two-phase system which are orthogonal to each other. It also follows that the α-component is aligned to phase a, whilst β-axis is leading by 90-degree from α-axis. Thus, the beta component is the only component that needs to be calculated. This saves some computational resources in the transformation[6]. The alpha and beta components can then be expressed as:

$$u_\alpha = u_a, u_\beta = \frac{1}{\sqrt{3}} \cdot (u_b - u_c) \tag{2.13}$$

Inverted Clarke-transformation

When the signal processing is finished, the three-phase calculation needs to be transformed back to three-phase before entering the machine. This is done by using the inverse Clarke-transformation. Equation 2.14 shows the inverted Clarke matrix.

$$\begin{bmatrix} u_a \\ u_b \\ u_c \end{bmatrix} = \frac{2}{3} \cdot \begin{bmatrix} 1 & 0 \\ -\frac{1}{2} & \frac{\sqrt{3}}{2} \\ -\frac{1}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix} \cdot \begin{bmatrix} u_\alpha \\ u_\beta \end{bmatrix} \tag{2.14}$$

Park-transformation

In the IP cores development, the reference framework used is the dq-reference. To achieve the dq-reference, a com-bination of the Clarke- and Park-transformation is done. The Park-transformation is represented in Equation 2.15.

$$\begin{bmatrix} i_d \\ i_q \end{bmatrix} = \frac{2}{2} \cdot \begin{bmatrix} cos(\omega t) & sin(\omega t) \\ -sin(\omega t) & cos(\omega t) \end{bmatrix} \cdot \begin{bmatrix} i_\alpha \\ i_\beta \end{bmatrix} \tag{2.15}$$

The corresponding inverse matrix is given in Equation 2.16.

$$\begin{bmatrix} i_\alpha \\ i_\beta \end{bmatrix} = \frac{2}{2} \cdot \begin{bmatrix} cos(\omega t) & -sin(\omega t) \\ sin(\omega t) & cos(\omega t) \end{bmatrix} \cdot \begin{bmatrix} i_d \\ i_q \end{bmatrix} \tag{2.16}$$

By applying the inverse Park-transformation of the controller, it becomes possible to calculate the current references using the dq-system and then transform it into the stator-oriented coordinates. This means calculating the AC-quantities from the DC-quantities[7].

Transformation of system model

It is assumed that there is no magnetic coupling between the phases, then the sum of current in the filter capacitors is assumed zero. This assumption applies to all vectors in the system, which is a balanced three-phase system. Consequently, the capacitor vector will not have a zero component in this scenario.

$$i_{10} = i_{1a} + i_{1b} + i_{1c} = 0$$

$$i_{20} = i_{2a} + i_{2b} + i_{2c} = 0$$

$$i_{C0} = i_{10} - i_{20} = 0$$

$$u_{C0} = 0 \tag{2.17}$$

Therefore, it is sufficient to only evaluate two components, d- and q- component or the α- and β-components as suggested.

2.3.2 Per-Unit Scaling

The emulated model employs scaling of parameters using the per-unit system. Utilizing the per-unit system offers several advantages. Firstly, it facilitates the detection of significant variations beyond the rated values, making it easier to identify deviations. Additionally, the same system can be utilized with different values, allowing the emulated system to be employed with smaller values in laboratory settings and larger values with larger components in practical applications.

The base values used are;

$$U_{base} = \hat{U}_{fn} = \sqrt{\frac{2}{3}} \cdot U_{N,rms,l-l} \tag{2.18}$$

$$I_{base} = \sqrt{2} \cdot i_{N,rms} \tag{2.19}$$

$$\tag{2.20}$$

The impedance base value can be calculated from the U_{base} and I_{base}.

$$Z_{base} = \frac{U_{base}}{I_{base}} = \frac{U_{N,rms,l-l}}{\sqrt{3} \cdot i_{N,rms}} \tag{2.21}$$

Then, the per-unit values of the parameters are calculated using the base values.

$$r_1 = \frac{R_1}{Z_{base}} \tag{2.22}$$

$$r_2 = \frac{R_2}{Z_{base}} \tag{2.23}$$

$$x_1 = \frac{\omega_N \cdot L_1}{Z_{base}} \tag{2.24}$$

$$x_2 = \frac{\omega_N \cdot L_2}{Z_{base}} \tag{2.25}$$

$$x_c = \frac{1}{\omega_N \cdot C \cdot Z_{base}} \tag{2.26}$$

2.3.3 Circuit-breaker and state-variables

When implementing the logic in the FPGA, two scenarios are considered for modeling the circuit-breaker connection to the grid. The first scenario involves the AFE being connected to the grid, then the closed. The second scenario includes the AFE being disconnected from the grid, then the breaker is open.

Initially, the breaker is assumed to be open. In this state, the current flows from the converter and passes through the capacitor, resulting in no current flowing to the grid, $i_2 = 0$. Since the voltage across the breaker is measured on the grid side, we can establish the condition $u_{breaker} = u_2$, leading to the following set of conditions.

$$u_{breaker} = u_2, i_2 = 0 \tag{2.27}$$

In addition, Equation 2.8, providing an equation for the voltage u_1, is derive presenting an expression for the current flowing from the converter in Equation 2.28. Furthermore, the scenario with an open breaker also includes the equation denoting the capacitor voltage depicted in Equation 2.29.

$$u_1 = r_1 \cdot i_1 + \frac{x_1}{\omega_N} \frac{di_2}{dt} + u_C$$

$$\frac{di_1}{dt} = \frac{\omega_N}{x_1} \cdot (-r_1 \cdot i_1 + u_1 - u_C) \tag{2.28}$$

$$\frac{du_C}{dt} = \omega_N \cdot x_C \cdot (i_1 - i_2) \tag{2.29}$$

For the second scenario, when the breaker is closed, the following condition applies:

$$u_{breaker} = u_C$$

Similarly to the first scenario where the breaker is open, an expression for the current flowing from the converter, denoted as i_1, is derived based on the equation for the voltage u_1 given in Equation 2.8. This expression is presented in Equation 2.30. Additionally, the equation denoting the capacitor voltage Equation 2.31 remains unchanged.

$$u_1 = r_1 \cdot i_1 + \frac{x_1}{\omega_N} \frac{di_1}{dt} + u_C$$

$$\frac{di_1}{dt} = \frac{\omega_N}{x_1} \cdot (r_1 \cdot i_1 + u_1 - u_C) \tag{2.30}$$

$$\frac{du_C}{dt} = \omega_N \cdot x_C \cdot (i_1 - i_2) \tag{2.31}$$

From Equation 2.9, an expression for the current passing through the breaker, denoted as i_2, is derived. This expression is presented in Equation 2.32.

$$u_C = r_2 \cdot i_2 + \frac{x_2}{\omega_N} \frac{di_2}{dt} + u_2$$

$$\frac{di_2}{dt} = \frac{\omega_N}{x_2} \cdot (-r_2 \cdot i_2 + u_C - u_2) \tag{2.32}$$

Breaker position	$u_{breaker}$	i_2
open	u_C	0
closed	u_2	nonzero

Table 2.1: Logic of the emulated model

2.3.4 Discrete model

Due to the inherent complexity of solving complex differential equations, a numerical method for computing the approximate solutions is required. The model is discretized, meaning that the continuous equations are transformed into a set of discrete equations that can be solved iteratively using numerical techniques.

2.3.4.1 Eulers method

One of the simplest numerical method is Euler's method. Euler's method is a one-step method. Given the ODE and the initial values, (t_0, y_0), some timestep, T_{step}, is chosen, and $t_1 = t_0 + T_{step}$. Based on this, an approximation for $y_1 = y(t_1)$ is calculated. Based on the new calculated values, (t_1, y_1), the process is repeated, where $t_2 = t_1 + T_{step}$.

Eulers method is derived from the Taylor expansion:

$$y(t_0 + T_{step}) = y(t_0) + T_{step} \cdot \frac{dy(t_0)}{dt} + \frac{1}{2}T_{step}^2 \frac{d^2y(t_0)}{dt^2} + ... \tag{2.33}$$

Assume that the step size, T_{step} is small, and the first two terms is dominating in the expression. The Forward Euler method is derived and given as:

$$y(t_0 + T_{step}) \approx y(t_0) + T_{step} \cdot f(t_0, y_0) + O(T_{step}^2) \tag{2.34}$$

$$y_{n+1} = y_n + T_{step} \cdot f(t_0, y_0) + O(T_{step}^2) \tag{2.35}$$

By employing the Forward Euler's method, the equations derived in subsection 2.3.3 for the matematical model of the LCL-filter is discreteized.

For the scenario when the breaker is open, Equation 2.36 - 2.38 shows the discreteized version of Equation 2.28 and 2.29.

$$u_{breaker}[k] = u_2[k], i_2 = 0 \tag{2.36}$$

$$i_1[k+1] = i_1[k] + \frac{\omega_N \cdot T_{step}}{x_1} \cdot (-r_1 \cdot i_1[k] + u_1[k] - u_C[k]) \tag{2.37}$$

$$u_C[k+1] = u_C[k] + \omega_N \cdot T_{step} \cdot x_C \cdot (i_1[k] - i_2[k]) \tag{2.38}$$

For the scenario where the breaker is closed, Equation 2.39 - 2.42 shows the discretized version of Equation 2.30, 2.31 and 2.32.

$$u_{breaker}[k] = u_C[k] \tag{2.39}$$

$$i_1[k+1] = i_1[k] + \frac{\omega_N \cdot T_{step}}{x_1} \cdot (-r_1 \cdot i_1[k] + u_1[k] - u_C[k]) \tag{2.40}$$

$$i_2[k+1] = i_2[k] + \frac{\omega_N \cdot T_{step}}{x_2} \cdot (-r_2 \cdot i_2[k] + u_{breaker}[k] - u_2[k]) \tag{2.41}$$

$$u_C[k+1] = u_C[k] + \omega_N \cdot T_{step} \cdot x_C \cdot (i_1[k] - i_2[k]) \tag{2.42}$$

Due to the assumption of a small time step, denoted as T_{step}, and the exclusion of second-order and higher terms, a local truncation error (LTE) is introduced at each time step during the numerical solution process [8]. For small values of T_{step}, the error incurred in each step is approximately proportional to T_{step}^2, as the largest term is multiplied by T_{step}^2 and the subsequent terms decrease with a factor of T_{step}^2. This relationship is specific to first-order differential equations. In general, a method of $O(T_{step}^{k+1})$ is considered to be of order k. Another significant error is the global error, which represents the absolute difference between the true solution and the calculated solution. However, if the true solution is unknown, the exact error cannot be determined. Nevertheless, neglecting round-off errors, the global error at the n-th time step is proportional to the LTE multiplied by n. Since n is inversely proportional to T_{step}, the global error can be expressed as $\frac{LTE}{T_{step}}$, indicating that the global error scales with T_{step}^{k}.

The Forward Euler's method is considered an explicit method, as it calculates the next step, y_{n+1}, based on the known quantities of the current state, y_n, and $f(t_n, y_n)$. While it is straightforward to implement, this method can be numerically unstable if the time step is too large. However, to address the stability issue, implicit methods can be used instead. The Backward Euler's method serves as the counterpart to the Forward Euler's method.

$$y_{n+1} = y_n + T_{samp} \cdot f(y_{n+1}, t_{n+1}) + O(T_{samp}^2) \tag{2.43}$$

Since $f(y_{n+1}, t_{n+1})$ is not known in advance and needs to be determined using root-finding algorithms like Newton-Raphson, the implicit method becomes more time-consuming and costly to implement.

2.3.4.2 Numerical stability

As previously mentioned, explicit methods can be numerically unstable. To further investigate and explain this issue, we consider the linear initial value problem (IVP):

$$\frac{dy}{dt} = -\lambda y, y(0) = 1 \tag{2.44}$$

with $\lambda > 0$ the solution is known to be:

$$y(t) = e^{-\lambda t} \tag{2.45}$$

This is a stable and smooth solution, starting at 1 for $t = 0$, and goes to 0 for $t = \infty$. Lets look at the discrete form, with the forwards Euler's method[8].

$$y_{n+1} = y_n - \lambda T_{step} y_n = (1 - \lambda T_{step}) y_n = (1 - \lambda T_{step})^2 y_{n-1} = ... = y_0 (1 - \lambda T_{step})^{n+1} \tag{2.46}$$

Using the aforementioned expression, an upper boundary is given for managing the numeric stability.

2.3.5 Phase-locked Loop

Phase-locked loop (PLL) is a phase-tracking subsystem useful for synchronizing power electronic converters to the power grid. It plays a crucial role in ensuring that the frequency, amplitude, and phase angle align with the certain requirements when synchronizing with rest of the power grid. The PLL assists in establishing a reliable connection to the grid when the specified criteria are met.

Design rules[9]:

$$h_0(s) = \left(k_p + \frac{K_p}{T_i} \cdot \frac{1}{s}\right) \cdot \frac{1}{s} = \left(2 \cdot \zeta \cdot \omega_f + \omega_f^2 \cdot \frac{1}{s}\right) \cdot \frac{1}{s} = \frac{2 \cdot \zeta \omega_f}{s} + \left(\frac{\omega_f}{s}\right)^2 \tag{2.47}$$

$$m(s) = \frac{1 + 2\zeta \cdot \frac{s}{\omega_f}}{1 + 2\zeta \cdot \frac{s}{\omega_f} + \left(\frac{s}{\omega_f}\right)^2} = \frac{1 + \frac{2\zeta}{\omega_f} \cdot s}{1 + 2\zeta \cdot \frac{s}{\omega_f} + \left(\frac{s}{\omega_f}\right)^2} \tag{2.48}$$

For our topology: The error input for the C++ function in the PESC controller platform for the PLL is the measured angle for $u_{breaker}$, $e = \zeta - \zeta_{filtered}$.

$$h_0(s) = \left(k_p + \frac{K_p}{T_i} \cdot \frac{1}{s}\right) \cdot \frac{\omega_n}{s} = \frac{\omega_n \cdot k_p}{s} + \left(\frac{\sqrt{\frac{\omega_n \cdot k_p}{T_i}}}{s}\right)^2 \tag{2.49}$$

$$m(s) = \frac{h_0(s)}{1 + h_0(s)} = \frac{\frac{\omega_n \cdot k_p}{s} + \left(\frac{\sqrt{\frac{\omega_n \cdot k_p}{T_i}}}{s}\right)^2}{1 + \frac{\omega_n \cdot k_p}{s} + \left(\frac{\sqrt{\frac{\omega_n \cdot k_p}{T_i}}}{s}\right)^2} = \frac{1 + T_i \cdot s}{1 + T_i \cdot s + \frac{T_i}{\omega_n \cdot k_p} \cdot s^2} \tag{2.50}$$

following

$$m(s) = \frac{1 + 2\zeta \cdot \frac{s}{\omega_f}}{1 + 2\zeta \cdot \frac{s}{\omega_f} + \left(\frac{s}{\omega_f}\right)^2} = \frac{1 + \frac{2\zeta}{\omega_f} \cdot s}{1 + 2\zeta \cdot \frac{s}{\omega_f} + \left(\frac{s}{\omega_f}\right)^2} \tag{2.51}$$

$$k_p = 2\zeta \cdot \frac{\omega_f}{\omega_n} = 2\zeta \cdot \frac{f_f}{f_n}, T_i = \frac{2\zeta}{\omega_f} = \frac{\zeta}{\pi \cdot f_f} \tag{2.52}$$

The damping term in the transfer function represents the system's ability to attenuate or suppress oscillations in the output frequency. It is usually controlled by adjusting the PLL's loop filter parameters, such as the gain, K_p, and the time constant, T_i. Damping ratio is a dimensionless quantity that measures the rate at which oscillations decay in a power system following a disturbance. It is usually denoted by the symbol ζ. A higher damping ratio indicates stronger energy dissipation and faster decay of oscillations, leading to improved system stability[10].

2.3.6 Active dampening

To mitigate the harmful effects of harmonics originating from the grid and protect the components from damage, the control system incorporates active damping. The primary objective of active damping is to store the current in the capacitor within the converter.

The transfer function for the output of the LCL-filter including the active dampening is derived as follows: Initially, the voltage equation governing the inductor, which aligns with the equation presented in Equation 2.8, is introduced. By transforming Equation 2.8 into the s domain, the following equation is obtained.

$$u_1(s) = (r_1 + \frac{x_1}{\omega_n} \cdot s) \cdot i_1(s) + u_C(s) \tag{2.53}$$

To incorporate active damping, the current sourced from the converter, denoted as i_1, is subtracted by an imaginary current flowing through a resistance connected in parallel with the capacitor [9]. Unlike passive damping, which involves the inclusion of a resistance in parallel with the capacitor, active damping stores the current within the converter. This principle of active damping not only prevents heat dissipation in the resistance, but also safeguards against component overheating. Furthermore, an expression representing the current flowing through the capacitor is provided:

$$i_C = \frac{1}{\omega_n \cdot x_C} \cdot \frac{du_C}{dt} = i_1 - i_D - i_2 = i_1 - \frac{u_C}{r_D} - i_2 \tag{2.54}$$

Where i_D represents the active current, r_D is the value of the resistance that is included to achieve the active dampening. In the s-domain, the derivation of the equation proceeds as follows:

$$(\frac{1}{r_D} + \frac{s}{\omega_n \cdot x_C}) \cdot u_C(s) = i_1 - i_2$$

$$u_C(s) = \frac{i_1 - i_2}{\frac{1}{r_D} + \frac{s}{\omega_n \cdot x_C}} \tag{2.55}$$

Then, by substituting Equation 2.55 into Equation 2.53, The transferfunction for the current controller of i_1 can be derived and obtained:

$$u_1(s) = \left(r_1 + \frac{x_1}{\omega_n} \cdot s + \frac{1}{\frac{1}{r_D} + \frac{s}{\omega_n \cdot x_C}} \right) \cdot i_1(s) = z_1(s) \cdot i_1(s) \tag{2.56}$$

$$\frac{i_1}{u_1} = \frac{1}{z_1(s)} = \frac{1}{r_1 + \frac{x_1}{\omega_n} \cdot s + \frac{1}{\frac{1}{r_D} + \frac{s}{\omega_n \cdot x_C}}} = \frac{\frac{1}{r_D} + \frac{s}{\omega_n \cdot x_C}}{1 + \frac{r_1}{r_D} + \frac{r_1 \cdot s}{\omega_n \cdot x_C} + \frac{x_1 \cdot s}{\omega_n \cdot r_D} + \frac{x_1}{\omega_n^2 \cdot x_C} \cdot s^2}$$

x_C is changed to the admittance, where $y_C = \frac{1}{x_C}$.

$$= \frac{1}{r_1 + r_D} \cdot \frac{1 + r_D \cdot y_C \cdot \frac{s}{\omega_n}}{1 + \left(\frac{r_1 \cdot r_D}{r_1 + r_D} \cdot y_C + \frac{x_1}{r_1 + r_D} \right) \cdot \frac{s}{\omega_n} + \frac{1}{1 + \frac{r_1}{r_D}} \cdot x_1 \cdot y_C \cdot \left(\frac{s}{\omega_n} \right)^2} \tag{2.57}$$

Under the assumption that r_D is significantly larger than r_1, the sum $r_D + r_1$ can be approximated as r_D. Consequently, the equation can be simplified accordingly.

$$\approx \frac{1}{r_D} \cdot \frac{1 + r_D \cdot y_C \cdot \frac{s}{\omega_n}}{\left(r_1 \cdot y_C + \frac{x_1}{r_D} \right) \cdot \frac{s}{\omega_n} + x_1 \cdot y_C \cdot \left(\frac{s}{\omega_n} \right)^2} \tag{2.58}$$

Equation 2.58 shows the final expression for the current controller of i_1 with active dampening.

2.4 Hysterises controller

Hysteresis controllers have conventionally been employed as current controllers. This approach involves measuring the actual motor current and comparing it with a predetermined reference value. When the actual current is lower than the reference current, the switch is turned on, and vice versa when the actual current slightly exceeds the reference value. The width of the hysteresis band determines the average switching frequency, in conjunction with the circuit's voltages and inductance. A well-known issue in this context is the decrease in switching frequency as the motor speed increases, resulting from the diminishing disparity between the voltage at the converter's input and the internal induced voltage in the motor. To mitigate this, adaptive hysteresis width techniques are sometimes implemented [7].

Chapter 3

PESC Control Platform

The main-focus the present thesis is to make an emulator based on the AFE, to faciliate tests that can be conducted in the physical lab with real components. The emulated model is built to be interfaced in the PESC software and firmware control platform. An advantage is that the parameters used for the AFE are taken as inputs in the IP cores, instead of hardware components. The user can then easily change the values. To best explain the most important aspects of how the PESC platform works, the hardware and software of the control platform are explained in a simplified manner.

The subsequent sections of this report will provide an explanation of the process involved in implementing the discretized model within the FPGA. These sections will offer detailed insights into the mathematical representation, algorithmic considerations, and the overall design and construction of the FPGA-based emulator.

3.1 Introduction on embedded real-time simulator- ERTS

When developing AFE connected to a LCL-filter, the system is simulated to emulate the model as close as possible. The Embedded real-time simulator makes it possible to simulate faster and achieve a rapid prototyping of the model. To achieve real-time emulation, the power hardware components as LCL-filter is programmed in the Field programmable gate array(FPGA) fabric of the Zynq System-on-Chip (SoC). The main component that makes this possible is that the execution tasks is happening in the FPGA. Matlab Simulink is used for simulation of the system and controller. First the model is implemented using PC- simulations. Then, the model is emulated using mathematical blocks. The mathematical equation used is then discretized using Euler's method. The discretized equations are used building the model in Simulink add on, system generator provided by Xilinx.

As proposed in [4], the ERTS used is mainly divided in three parts. This is allowing separate development, which again corresponds to reusage of common components, ensures uniformity across simulations, minimizes design and simulation failures, and helps diagnose the faults or design errors rapidly. By retaining much of the ERTS modularized, the proposed simulator can be modified to emulate different motor-types and designs, converters or mechanical loads. The proposed simulator is designed using the per-unit system, which makes its main components common across machines, loads and converters irrespective of their ratings.

3.2 Field- programmable gate array

A Field Programmable Gate Array (FPGA) is constructed using configurable logic blocks (CLBs), also known as slices or logic cells. CLBs consists of flip flops and lookup tables, playing crucial roles in the functionality of an FPGA [11]. The flip flops store a logic input, 0 or 1, until the next clock edge, while lookup tables determine the interconnection of the logic blocks. Developing an FPGA entails creating a digital circuit through the utilization of hardware description languages like VHDL or Verilog.[6]. These languages serve as the primary means for defining the logic algorithms executed on the FPGA chip. Following the design phase, it is common practice to verify the FPGA code through rigorous testing. Subsequently, the text-based logic is compiled through a series of complex steps, resulting in a configuration file or bitstream. This file contains information on how the logic components should be interconnected.

One significant advantage of FPGAs is their inherent ability to execute tasks concurrently, enabling faster and more efficient simulation times. This parallelism arises from the allocation of dedicated sections within the chip for each task, allowing for independent functioning without interference from other logic blocks.

Moreover, FPGAs facilitate accelerated development of control algorithms, as they can be rapidly prototyped and tested before deployment in real-world applications. This expedites product development cycles and facilitates the early detection of potential issues, ultimately reducing costs associated with rectifying problems during later stages of development.

3.3 Software and Hardware in the PESC control platform

3.3.1 Hardware

The hardware components of the PESC control platform consist of a PicoZed 7030 control board from Avnet mounted on a process interface board developed by SINTEF. The process interface board facilitates control of two two-level three-phase inverters and provides up to eight input measurement ports. The control board consists of a Zynq-7030 SoC (prosessor) that includes two ARM floating-point processors (CPUs) and one programmable logic FPGA device. One of the CPUs runs a Linux program responsible for programming the remaining CPU and FPGA, as well as enabling the process card to be programmed and monitored using specialized release and monitoring tools. These tools, including the Linux program, release tools, and monitoring tools, are provided by The Switch Marine Drives [12].

3.3.2 Software

Programming of control board involves using three different software, all provided by Xillinx. The code for the remaining CPU is written in C++ using Software Development Kit (SDK). The project simulated on the PESC platform needs to be created and uploaded to the board. The FPGA is programmed by connecting IP-cores together in a program called Vivado and the project to be uploaded in the FPGA is created. The IP-cores are created by using Xilinx's System Generator plugin in Simulink. Vivado also provide a generic IP-core catalogue that can be used.

To execute the emulated model in the FPGA, specific routines need to be performed within the PESC platform. The programming language used in the PESC Control Platform is C++. To upload the software to the hardware equipment, a zip file, the release, is built to be uploaded to the processor board. The ReleaseBuilder tool from The Switch is used to build a software package, which includes the TSW software for CPU0, the NTNU software for CPU1, and the binary file, built in the Vivado software, for the FPGA[13]. A release can be built with or without a database. The database contains various parameters used in the software. In a Master's or PhD work, it may be sufficient to hard-code the parameter values in the ConfigParameters.cpp file. The WatchView program is used for communication of the board and visualizion of the running.

As mentioned, the Switch is providing the release builder. The release builder program is uploading a database to the processor board and rebooting the board. The WatchView-program is used for communicating with the processor. The database has to be configured and created for a specific Drive software, which means that the parameters for the Drive Layer of the database become different for PM and DC motors. However, some parameters are the same. This is typical for the Application layer and the Converter Layer.

During the startup process of the SINTEF board and PicoZed board, the following actions are performed. First, CPU0 unzips the project file, uploaded during the release, and programs the CPUs (CPU0, CPU1) and FPGA. CPU0 writes the project database into the Flash memory and initializes the parameters in the program of CPU1, as specified in the database editor during database generation[13]. Once these steps have been executed, CPU1 initiates the execution of main.cpp, which is programmed by NTNU. The main tasks performed by the program include:

- Setting up the communication channel between CPU0 and CPU1
- Initializing communication routines
- Initializing data loggers
- Running the SW1.Initialize() routine developed by NTNU
- Starting the interrupt routines, including interrupt priorities
- Entering an infinite while loop: Background routine

The next thing to describe is the interrupt routine. The interrupt routine is built as a layer model programmed in C++ code. The interrupt signal is given on the top and bottom of the PWM signals. It is designed in such way that the interrupt signal has enough time to finish before the next signal occurs. The interrupt routine is designed to handle interrupts that immediately need attention from the processor. The routine is run and checks if any of the system states are changed. The routine is calling the function called, DriveRoutineFast(), and the following procedures is executed in the code.

As mentioned earlier the Drive routine function is a layer model, which means that the structure of the code is divided into four classes called application layer, Drive layer, Converter layer and firmware layer.

Then, the interrupts routine are described. First, the measurements from the FPGA is collected, typically currents and voltages. Subsequently, the status is examined by reading the converter's status bits. In the event of over currents, over- or under-voltages, specific fault bits are set, and the converter's status is flagged as "Fault." Similar tests are performed in the Drive Layer. If some fault occur, the drive will be tripped, i.e. turned off. The completed drive will be set in the state Malfunction in the Application SW.

The Application function is run in the application layer. It consists of a state machine based on the driveCom-standard. This is an old standard, which has been the basis for several other standards. The states are used for allowing you to go from one state to another like shown in Figure 3.1.

The Drive Layer is controlled by the Application Layer. The Application can request the local state of the Drive Layer, but also send parameters, such as the torque reference and power, limits, etc. The Drive Layer will send actual values and status to the Application Layer. Notably, this communication framework operates on a "need-to-know" basis, ensuring that the Application Layer remains agnostic to the specific motor type being controlled.

Similarly, the Converter Layer is governed by the Drive Layer. The Drive Layer can request the local state of the Converter Layer. The Drive Layer can also send the required Voltage vector length and angle to the converter layer. The Converter Layer will send actual measurements and status to the Drive Layer. The communication between the layers does not share any unnecessary information.

3.4 Scaling

As the Simulation have switched over to fix point numbers, there are limited storage capacity in the FPGA, and a right scaling is needed. The FPGA-based implementation consists of building blocks that digitally replicate components of the LCL filter, such as resistors (R), inductors (L), and capacitors. The emulated model, developed within the FPGA, is designed to operate in a per-unit system and is parameterized by the application software. The advantages of this emulator is that the scaling block ensures that the generic IP cores don't see a difference between the physical hardware and the emulator when exchanging the parameters. This makes the design procedures valuable for the model design[4].

3.5 Numeric Representation (Binary Design)

The AXI interface in the System-on-Chip (SoC) has a word length of 32 bits, which is why the Extended Real-Time System (ERTS) also maintains a 32-bit word length. This ensures convenient data exchange with the on-chip processor. Additionally, a 32-bit representation guarantees numerical accuracy for real-time simulations[4]. Within the 32-bit representation, the precision allows for a range from +7.99 to -8, with one bit dedicated to representing the sign. As the FPGA operates using fixed-point numbers with a specified time step parameter, it is important to consider the storage requirements when selecting the time step. If the time step is too small, it may lead to challenges in representing extremely small numbers. For example, if a number cannot be expressed with 64 bits, the integral may result in zero. To optimize hardware utilization, the model employs a 32-bit representation.

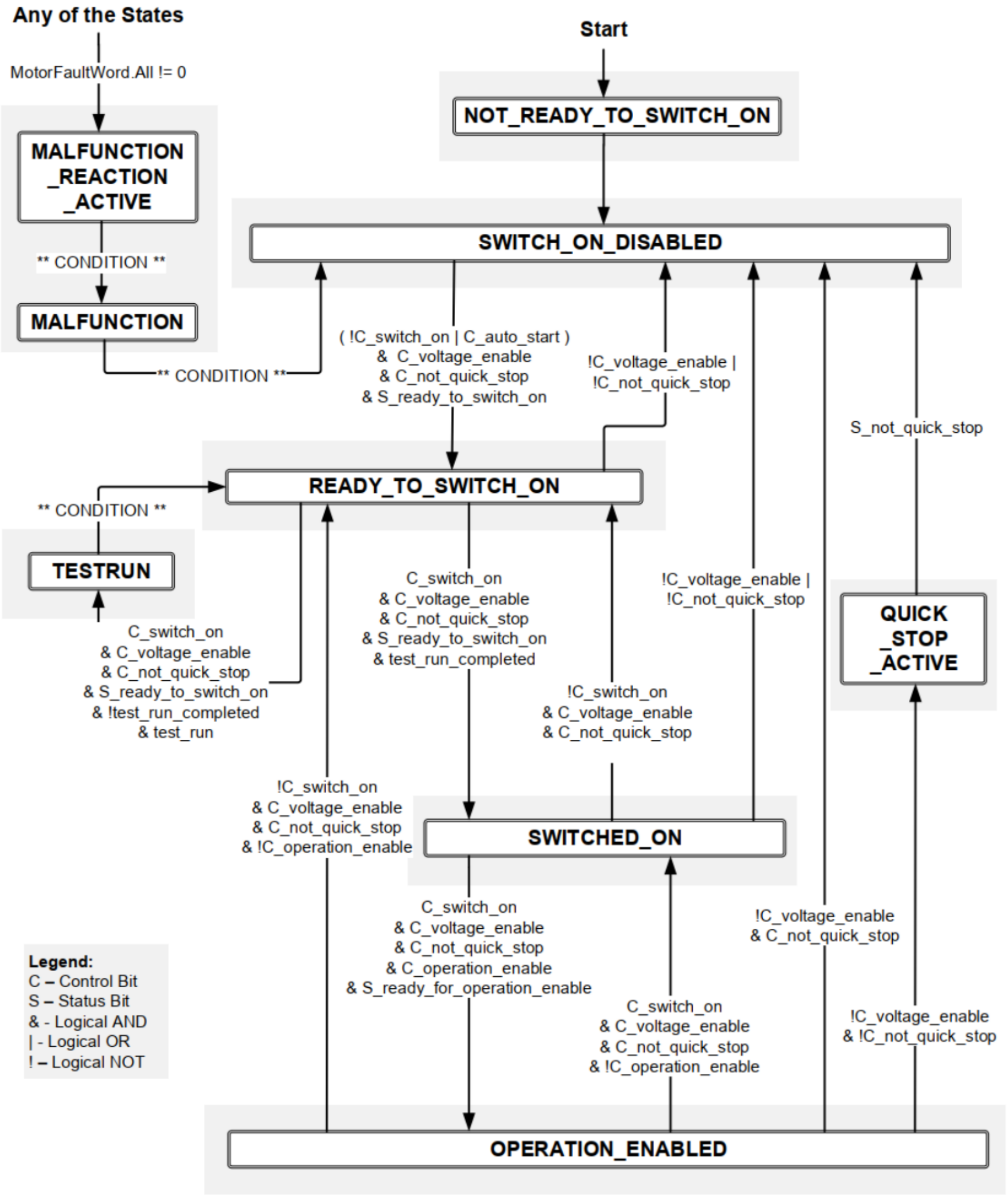

Any of the States
MotorFaultWord.All != 0
MALFUNCTION _REACTION _ACTIVE
** CONDITION **
MALFUNCTION
** CONDITION **
Start
NOT_READY_TO_SWITCH_ON
SWITCH_ON_DISABLED
(!C_switch_on | C_auto_start) & C_voltage_enable & C_not_quick_stop & S_ready_to_switch_on
!C_voltage_enable | !C_not_quick_stop
S_not_quick_stop
READY_TO_SWITCH_ON
** CONDITION **
TESTRUN
C_switch_on & C_voltage_enable & C_not_quick_stop & S_ready_to_switch_on & !test_run_completed & test_run
C_switch_on & C_voltage_enable & C_not_quick_stop & S_ready_to_switch_on & test_run_completed
!C_voltage_enable | !C_not_quick_stop
QUICK _STOP _ACTIVE
!C_switch_on & C_voltage_enable & C_not_quick_stop
!C_switch_on & C_voltage_enable & C_not_quick_stop & !C_operation_enable
SWITCHED_ON
C_switch_on & C_voltage_enable & C_not_quick_stop & C_operation_enable & S_ready_for_operation_enable
!C_voltage_enable & C_not_quick_stop
C_switch_on & C_voltage_enable & C_not_quick_stop & !C_operation_enable
!C_voltage_enable & !C_not_quick_stop
Legend:
C – Control Bit
S – Status Bit
& - Logical AND
| - Logical OR
! – Logical NOT
OPERATION_ENABLED

Figure 3.1: Figure of the DriveCom states in the application layer [13]

3.6 Clock settings

The representation of the physical motor drive system involves the utilization of three distinct clock cycles: the FPGA clock, the processor interrupt cycle, and the solver clock [4]. The FPGA clock, with a duration of 10 ns, is employed for components requiring high processing speed, such as the generic IP cores and the converter block. The processor interrupt cycle, which operates at twice the speed of the pulse-width modulation (PWM), is sufficient for the application software. The solver clock, with a duration of 1 µs, is utilized in the discretized model solver within the hardware emulator. Solver clock is used for the load models in the hardware emulation. T_{step} will determine the integration intervals, thus shorter integration intervals yield more precise results. So, small step size and high numerical representation give precise results.

In the SysGen model, a generator clock signal is employed to synchronize the timing of the emulated RC circuit, necessitating higher-speed operation. However, this increased speed also leads to higher chip temperatures due to the increased number of real switching steps.

When conducting simulations of any type, it is important to consider the duration of the simulation for the different components. The FPGA can operate at a maximum clock speed of 100 MHz, providing the capability for efficient and timely execution of simulations.

3.7 Implementation of emulator in FPGA

The development of an FPGA-based control platform utilizes the graphical design tool System Generator Xilinx in Simulink. Simulink, coupled with the Xilinx library, serves as the programming environment for FPGA implementation, enabling the utilization of blockset libraries containing various blocks for IP core design within Simulink. This approach offers the advantage of block diagram design and HDL code testing in a unified Simulink environment. The two primary IP cores employed in this model are the transformation core and the current core, which have already been developed and utilized in a motor drive application. Therefore, leveraging the pre-existing IP cores significantly reduces unnecessary effort and accelerates the development process of the FPGA for the present model. This section only describes the block setup for building the emulated model of the LCL filter for the AFE. For the development of the IP cores, Thomas S. Haugen made valuable contributions to the implementation of the FPGA and the design of the IP core.

3.7.1 LCL filter model

The emulated model of the LCL filter consists of mathematical equation for the inverter current, grid current and the capacitor voltage. The Xilinx Simulink blockset operates using a dedicated fixed-point signal format. Further, the implementation of the inverter currents, grid current and capacitor voltage consists of the discretized equations from subsection 2.3.4.

The equation below is used for building the representation of the converter current with Xilinx blocks shown in Figure 3.2.

$$i_1[k+1] = i_1[k] + \frac{\omega_N \cdot T_{step}}{x_1} \cdot (-r_1 \cdot i_1[k] + u_1[k] - u_C[k]) \tag{3.1}$$

Further, the following equation is used when building the calculation for the grid current with Xilinx blocks represented in Figure 3.3.

$$i_2[k+1] = i_2[k] + \frac{\omega_N \cdot T_{step}}{x_2} \cdot (-r_2 \cdot i_2[k] + u_{breaker}[k] - u_2[k]) \tag{3.2}$$

Finally, the next equation provides the initial point for the Xilinx representation of the capacitor voltage calculations.

$$u_C[k+1] = u_C[k] + \omega_N \cdot T_{step} \cdot x_C \cdot (i_1[k] - i_2[k]) \tag{3.3}$$

Prior to exiting the FPGA-module, it is necessary to convert the grid current from the dq-reference frame to the three-phase abc coordinates. The transformation process can be represented by a Xilinx block, as illustrated in Figure 3.6.

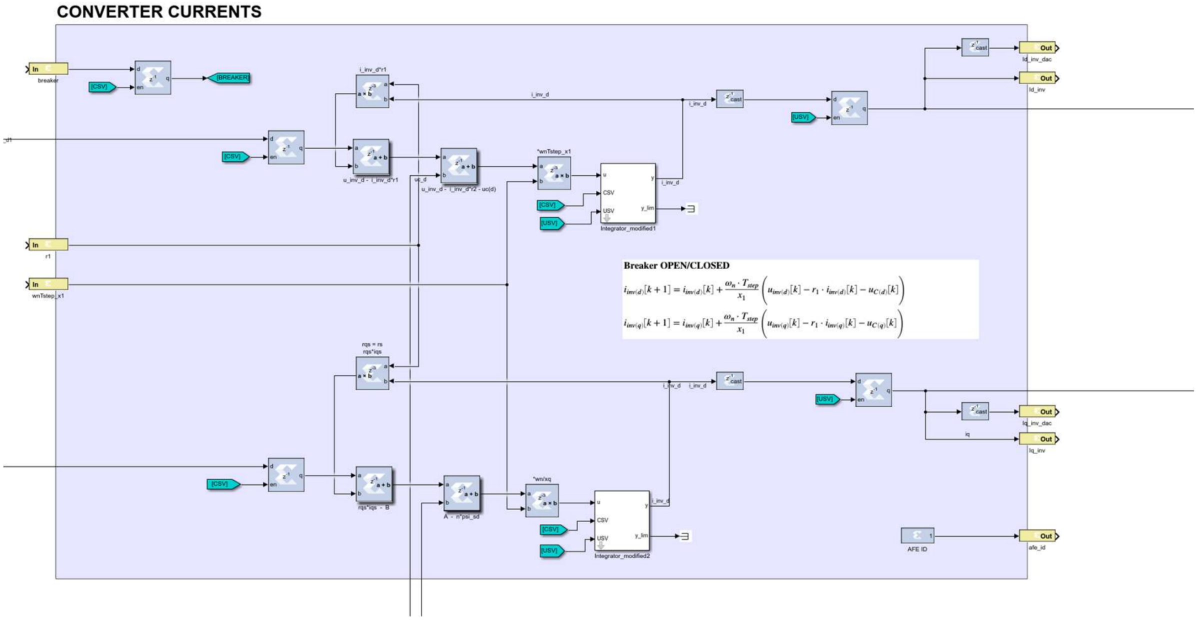

CONVERTER CURRENTS
In
breaker
[CSV]
[BREAKER]
i_inv_d*r1
i_inv_d
u_inv_d - i_inv_d*r1
uc_d
u_inv_d - i_inv_d*r2 - uc(d)
*wnTstep_x1
Integrator_modified1
u
CSV
USV
y
y_lim
[USV]
cast
Out
Id_inv_dac
Id_inv
In
r1
In
wnTstep_x1
Breaker OPEN/CLOSED
$i_{inv(d)}[k+1] = i_{inv(d)}[k] + \frac{\omega_n \cdot T_{step}}{x_1}\left(u_{inv(d)}[k] - r_1 \cdot i_{inv(d)}[k] - u_{C(d)}[k]\right)$
$i_{inv(q)}[k+1] = i_{inv(q)}[k] + \frac{\omega_n \cdot T_{step}}{x_1}\left(u_{inv(q)}[k] - r_1 \cdot i_{inv(q)}[k] - u_{C(q)}[k]\right)$
rqs = rs
rqs*iqs
rqs*iqs - B
A - n*psi_sd
*wn/xq
Integrator_modified2
iq
Iq_inv_dac
Iq_inv
AFE ID
afe_id

Figure 3.2: The converter current calculation represented with Xilinx blocks.

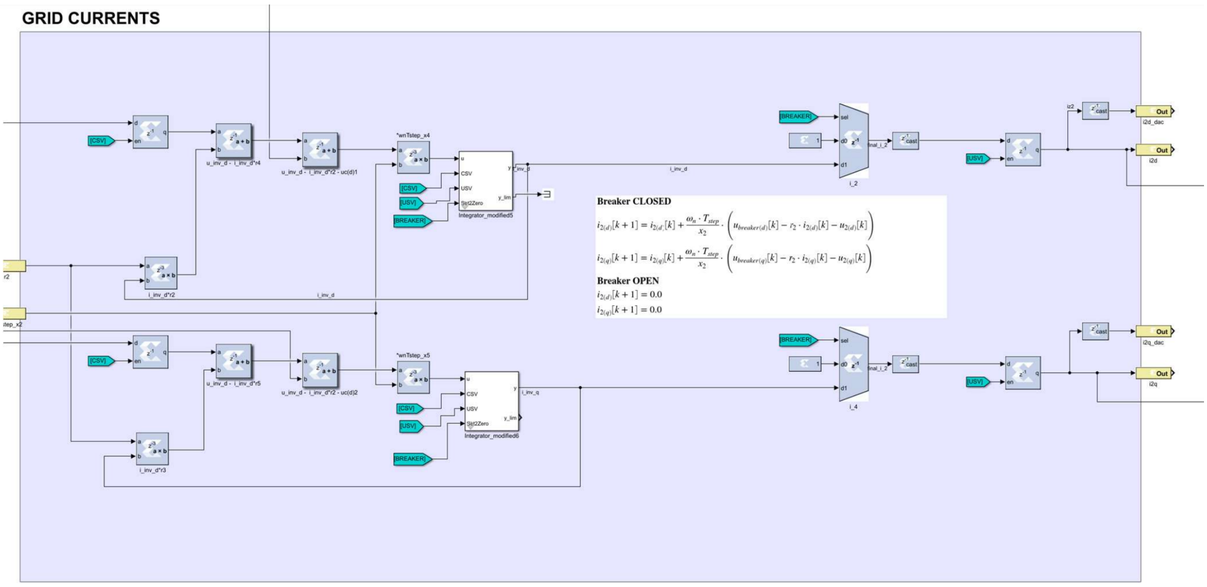

GRID CURRENTS
Breaker CLOSED
$i_{2(d)}[k+1] = i_{2(d)}[k] + \frac{\omega_n \cdot T_{step}}{x_2} \cdot \left(u_{breaker(d)}[k] - r_2 \cdot i_{2(d)}[k] - u_{2(d)}[k]\right)$
$i_{2(q)}[k+1] = i_{2(q)}[k] + \frac{\omega_n \cdot T_{step}}{x_2} \cdot \left(u_{breaker(q)}[k] - r_2 \cdot i_{2(q)}[k] - u_{2(q)}[k]\right)$
Breaker OPEN
$i_{2(d)}[k+1] = 0.0$
$i_{2(q)}[k+1] = 0.0$
[CSV]
[USV]
[BREAKER]
u_inv_d - i_inv_d*r4
u_inv_d - i_inv_d*r2 - uc(d)1
*wnTstep_x4
Integrator_modified5
i_inv_d*r2
i_inv_d
i_2
i_4
final_i_2
cast
Out
i2d_dac
i2d
i2q_dac
i2q
u_inv_d - i_inv_d*r5
u_inv_d - i_inv_d*r2 - uc(d)2
*wnTstep_x5
Integrator_modified6
i_inv_q
i_inv_d*r3
sel
d0
d1
y_lim
Set2Zero

Figure 3.3: The grid current calculation represented with Xilinx blocks.

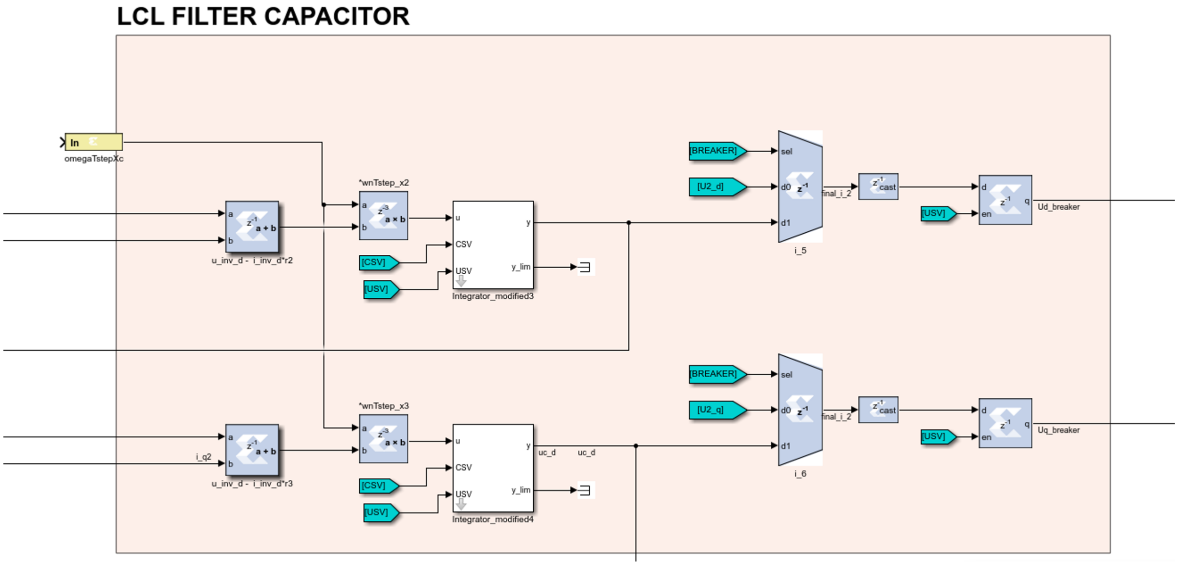


Figure 3.4: The capacitor voltage calculation represented with Xilinx blocks.

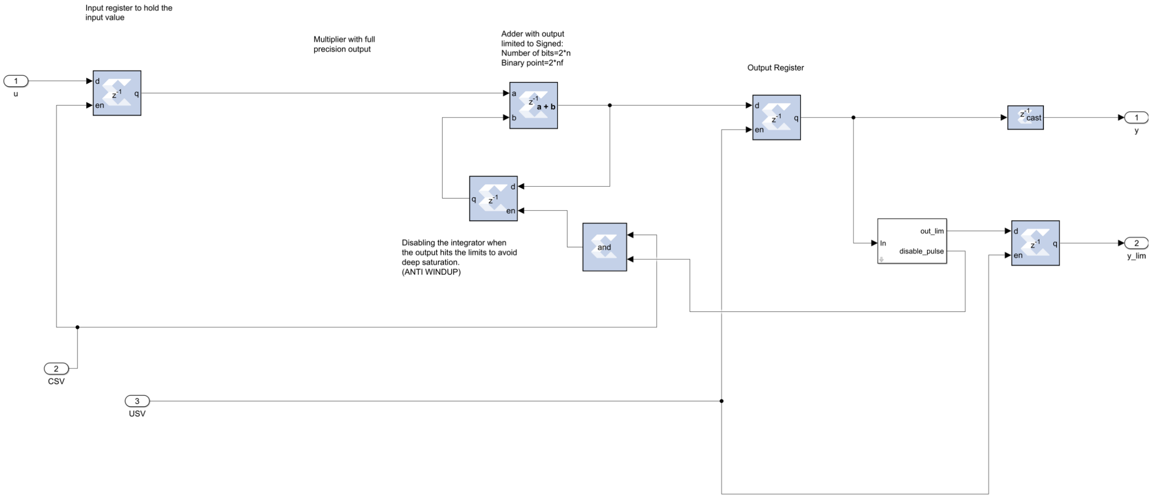

Input register to hold the input value
Multiplier with full precision output
Adder with output limited to Signed: Number of bits=2*n Binary point=2*nf
Output Register
1
u
d
en
q
a
b
a + b
Disabling the integrator when the output hits the limits to avoid deep saturation. (ANTI WINDUP)
and
cast
1
y
In
out_lim
disable_pulse
2
y_lim
2
CSV
3
USV

Figure 3.5: Details of the discretized integrator.

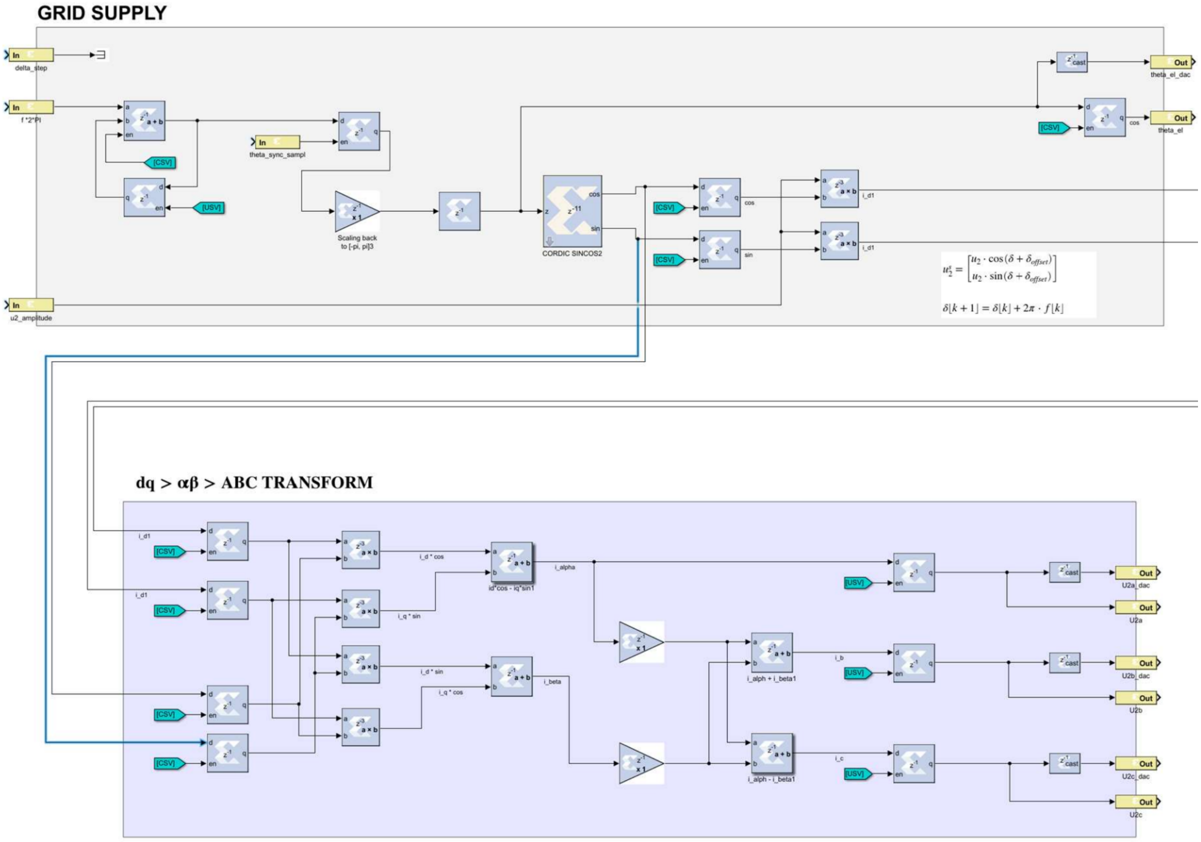


Figure 3.6: The angle and grid voltage step representation with Xilinx blocks.

Chapter 4

Simulations

In this chapter, simulations of an Active Electric Filter (AFE) connected to a LCL-filter is concucted. The purpose is to analyze and evaluate the behavior of the system using different simulation models. The simulations are carried out using MATLAB Simulink and the ERTS is tested using the PESC control platform.

4.1 Simplified Simulink model

In this section, the simulated model represents an active front end (AFE) connected to a LCL-filter, where the converter is implemented as an ideal current source. The purpose of this simplified model is to gain a better understanding of the system's dynamics and functionality, with a particular focus on studying the grid following control scheme. A description of the Simulink-based model can be found in appendix section A. The model is

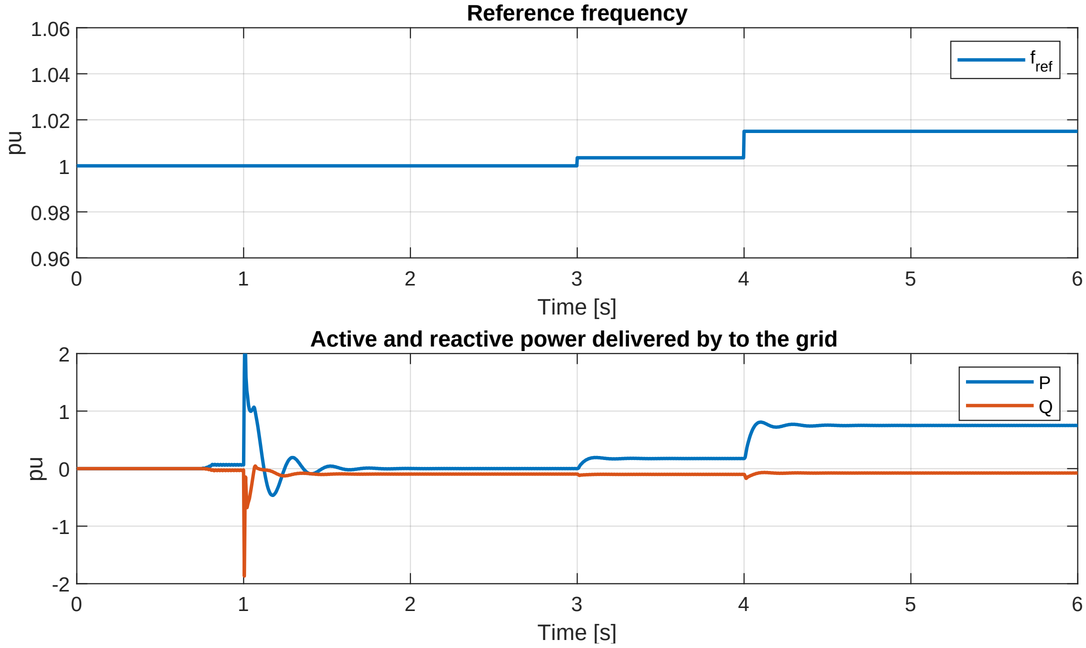
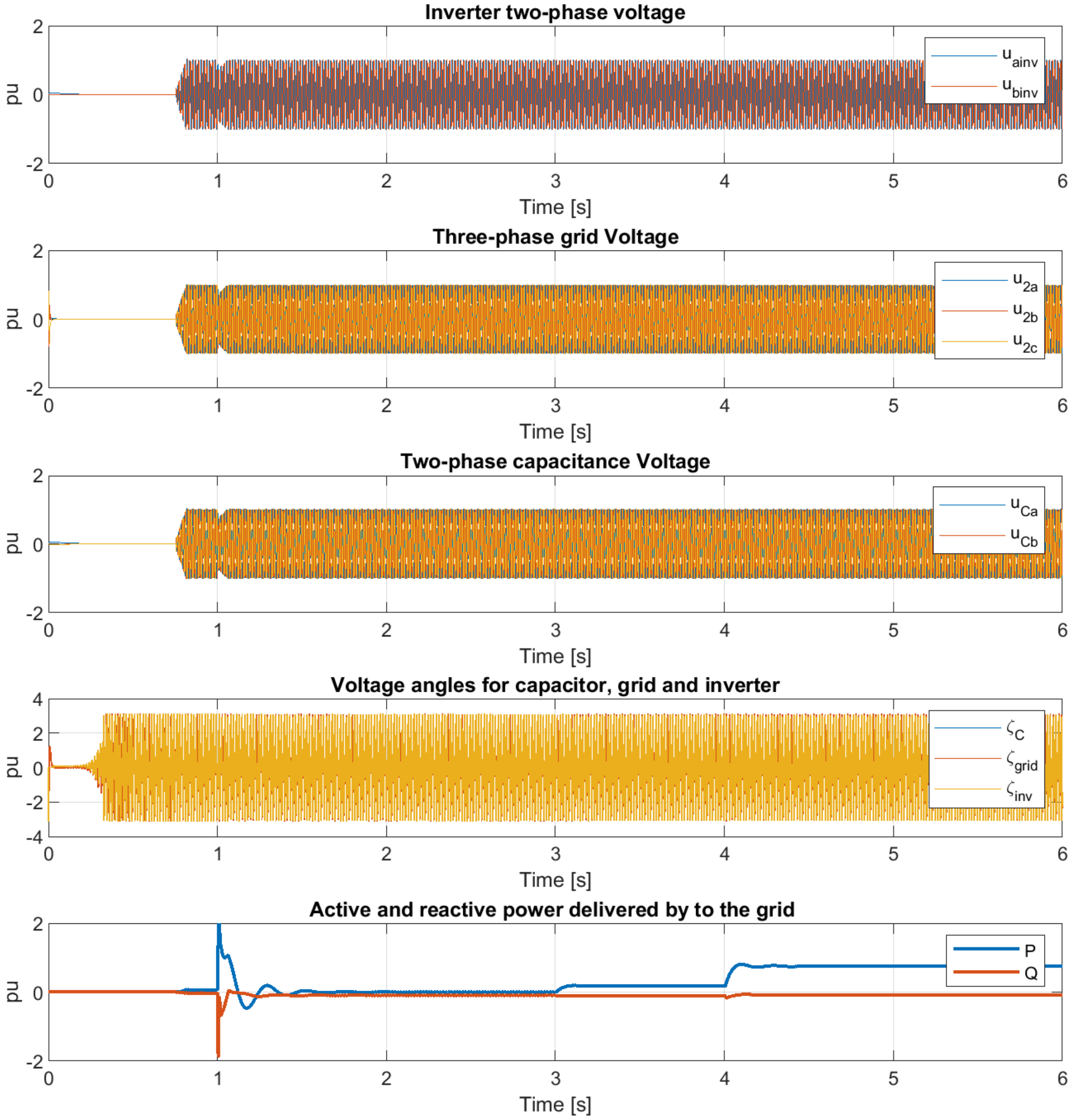


Figure 4.1: Simulation of the frequency cange and the active and reactive power delivered to the grid.

connected to a stiff grid with nominal frequency at 60 Hz and an rms line to line voltage magnitude at 690 V. The power circuit breaker is closed at 1 second, concecently the grid is connected to the AFE converter including filter. To demonstrate the grid following control scheme, the reference frequency in the model is changed. In the time duration from 0 to 3 seconds, the frequency reference equal 1 pu. Then from 3 to 4 seconds the reference frequency is changed from 1 pu to 1.0035 pu. Lastly, at 4 seconds, the frequency reference is altered to 1.015 pu.



Inverter two-phase voltage

Three-phase grid Voltage

Two-phase capacitance Voltage

Voltage angles for capacitor, grid and inverter

Active and reactive power delivered by to the grid

Figure 4.2: Simulation with PESC controller of the starting state with the breaker open.

From top to bottom, Figure 4.2 shows the inverter voltage, grid voltage, capacitor voltage, voltage angles and delivered active and reactive power to the grid. Regardless of the change in when the grid is connected, there are no variations in the system voltages.

It is evident that the model adjusts its active and reactive power to accommodate the system changes. Figure 4.1

illustrates the changes of P and Q delivered to the grid according to the frequency change in the grid.

It is observed that when the frequency change, the active and reactive power change within the system to adapt the system change, as expected when using the grid-following control scheme. When the converter is connected to the system, transient responses can be observed, leading to variations in the respective power system values.

4.1.1 Exact model

In this section, the AFE model is upgraded to provide a more precise representation of the converter system. The current source, that initially represented the converter, is replaced with a real converter featuring the power electronic IGBT switches, resulting in a more realistic and complex model. This enhancement takes into account the influence of the fast switching frequencies of the half bridges. As a consequence, disturbances such as voltage fluctuations become more apparent. Control of the converter is achieved through the implementation of a virtual synchronous machine (VSM) and a phase-locked loop unit (PLL) also including the hysteresis current controller. The input of the emulator model will be the measured inverter voltage, and the grid is a stiff three-phase voltage of adjustable voltage amplitude and frequency.

Furthermore, to achieve the goal of the fix-point model implemented in the FPGA, a emulated model of the LCL-filter is tested in Simulink. The mathematical model of the LCL-filter described in section 2.3 is built with the floating point data type Simulink blocks to ensure its capability in capturing the desired dynamics of the system. To check if the model gives the correct dynamic reactions, the filter capacitor current and voltage from the emulated model are compared with the filter capacitor current and voltage of the Simulink model with the realistic converter.

Parameter	Value
$V_{gn,L-L}$	690 V
I_n	1500 A
fn	60 Hz
capasitance, C_c	990e-6 F
inverter reactance, X_1	83e-6 H
Grid reactance, X_2	106e-6 H
inverter resistance, R_1	2.77e-3 Ω
Grid reactance, R_2	8.08e-4 Ω

Table 4.1: The parameters used in the accurate Simulink model

Parameter	Value
capasitance, y_c	0.10 pu
inverter reactance, x_1	0.11782 pu
Grid reactance, x_2	0.15047 pu
inverter resistance, r_1	0.01043 pu
Grid reactance, r_2	3.0424e-3 pu

Table 4.2: The parameters presented in Table 4.1 converted to the per unit system.

To ensure a meaningful comparison between the emulated mathematical model and the exact model, it is crucial to align their parameter settings. Consequently, the parameters used in the simulation of the exact model are presented in Table 4.1. However, it should be noted that the emulated model is scaled in the per-unit system to fit into the 32-bit fixed-point number system. Therefore, the parameters have been converted to per-unit values and are presented in Table 4.2. This conversion enables a consistent framework for evaluating and contrasting the emulated and accurate models, taking into account their specific characteristics and requirements. The emulated model, built with the floating point data type Simulink blocks, can be seen in the Appendix A.2.

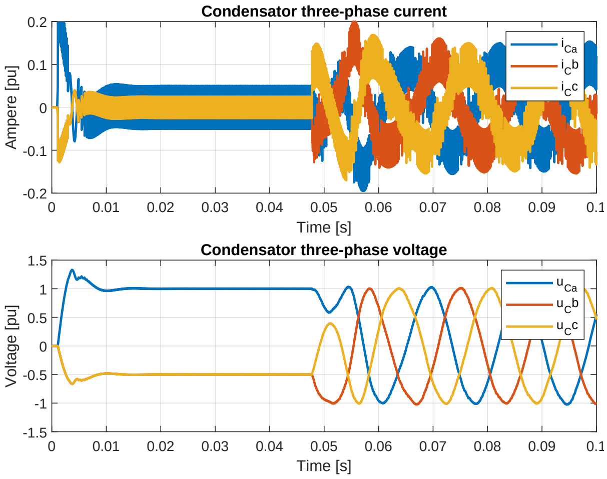


Figure 4.3: Plot of filter capacitor current and voltage in the exact simulink model

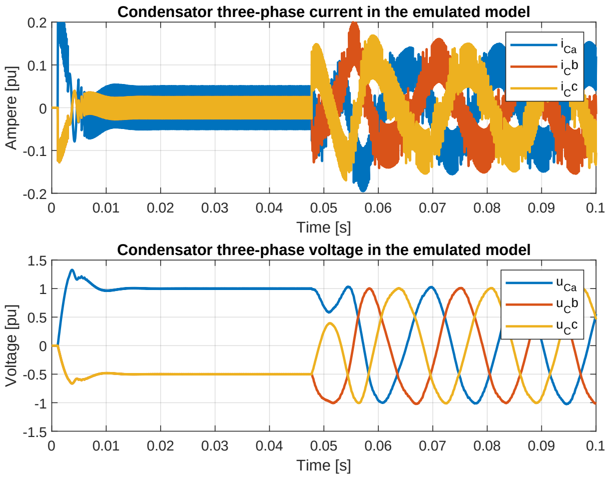


Figure 4.4: Plot of filter capacitor current and voltage in mathematical emulated model

Figure 4.3 and 4.4 represents the time-varying behavior of the current flowing through the filter capacitor and the voltage over the capacitance in each phase of the three-phase system for the Simulink model and the emulated model respectively.

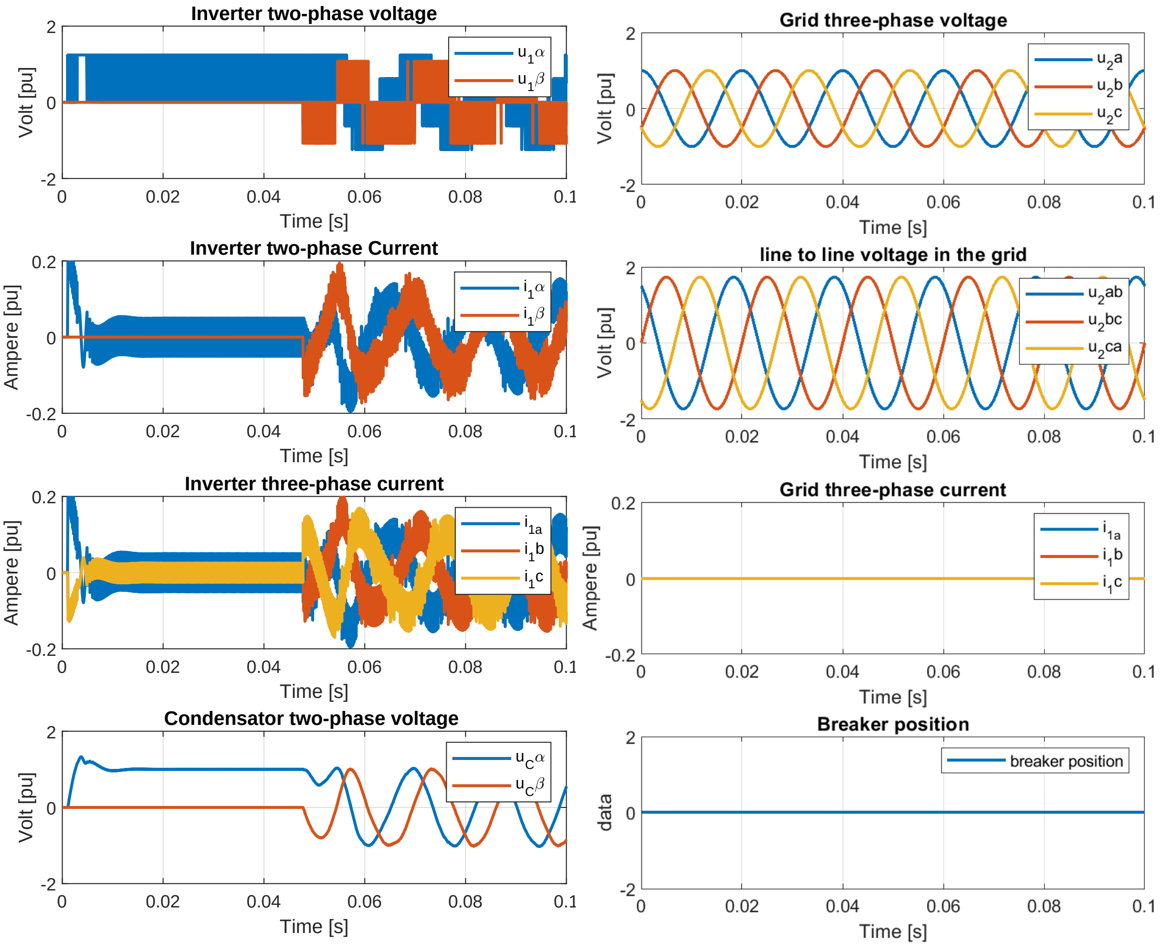


Figure 4.5: The above plots shows the grid voltage and current, v_2 and i_2, The inverter voltage and current, v_1 and i_1, the filter capacitor voltage, v_C, and the breaker position.

By comparing Figure 4.3 and 4.4 it can be seen that the mathematical model gives the same output current and voltage and that the two plots are identical.

Figure 4.5 shows a plot of some values of the emulated model. The simulation performed before the breaker is closed to the grid, because the exact model does not stabilize after the grid is connected to the converter. The active and reactive effect is drifting off after some time, which can be observed from Figure 4.6. Therefore the breaker position is displayed as zero, which indicates that the breaker between the grid and converter is open. Another indication of the open breaker is the grid current, shown to be zero. These aspects imply that it does not flow any current through to the grid when the breaker is open.

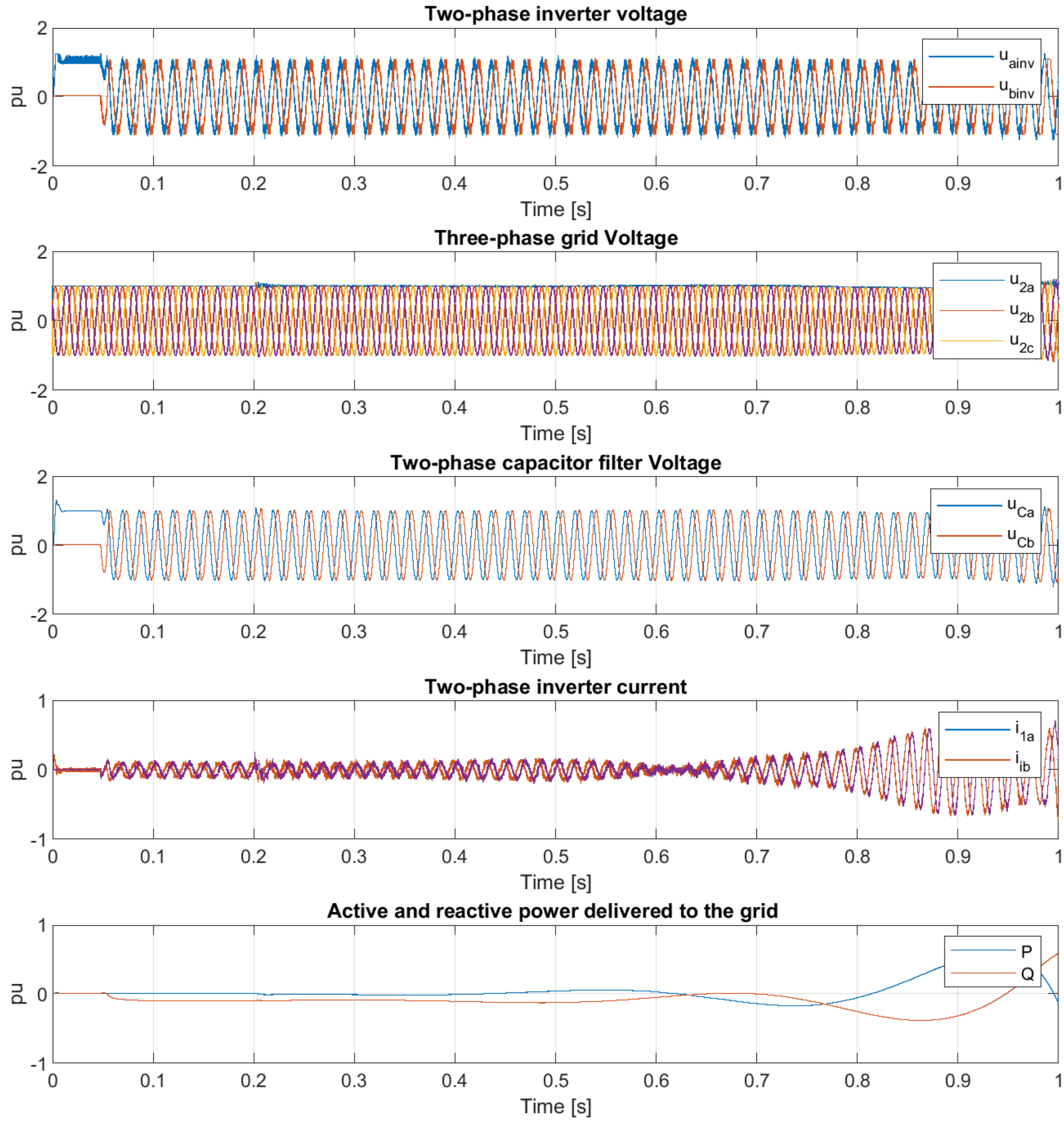


Figure 4.6: Simulation from the exact model showing the off drifting of the active and reactive power.

4.1.2 Testing of PLL

In this section, the PLL dynamics for the exact simulink model is tested. Four different sets of values were simulated for the PLL, as outlined in Table 4.3. The parameters ζ, f_n, and f_f were selected, while K_p and T_i were computed based on the expressions provided in Equation 2.52. Figure 4.7, 4.8,4.9 shows the dynamic of PLL with the parameters displayed in Table 4.3.

	ζ	f_n[Hz]	f_f[Hz]	K_p	T_i[s]
Figure 4.7	1.4	60	58	2.71	0.0077
Figure 4.8	1.4	60	35	1.63	0.0127
Figure 4.9	1.4	60	20	0.93	0.0223

Table 4.3: Table of the simulated ζ, f_n, f_f, Kp and Ti values for the PLL in the exact Simulink model.

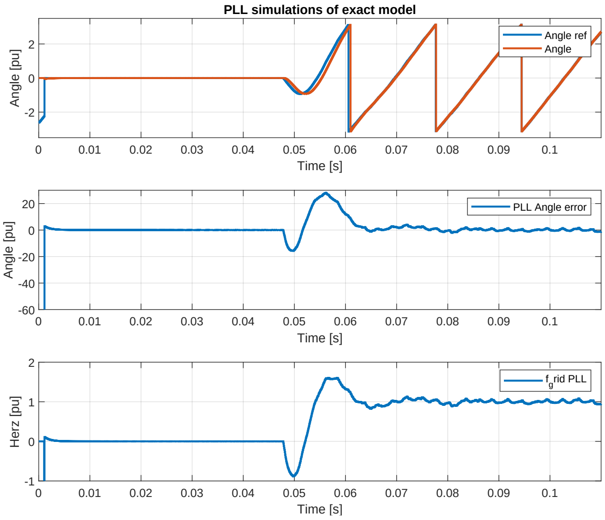


Figure 4.7: Simulation with the exact model for testing PLL, with $f_f = 58Hz$ and $\zeta = 1.4$, which gives $T_i = 7.7ms$ and $K_p = 2.71$

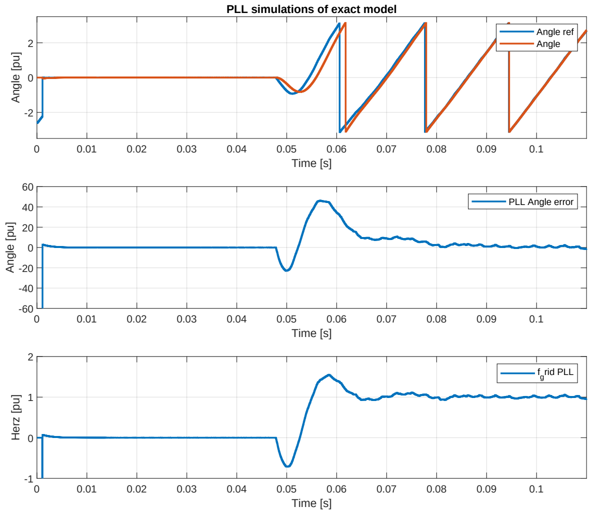


Figure 4.8: Simulation with the exact model for testing PLL, with $f_f = 35Hz$ and $\zeta = 1.4$, which gives $T_i = 12.7ms$ and $K_p = 1.63$

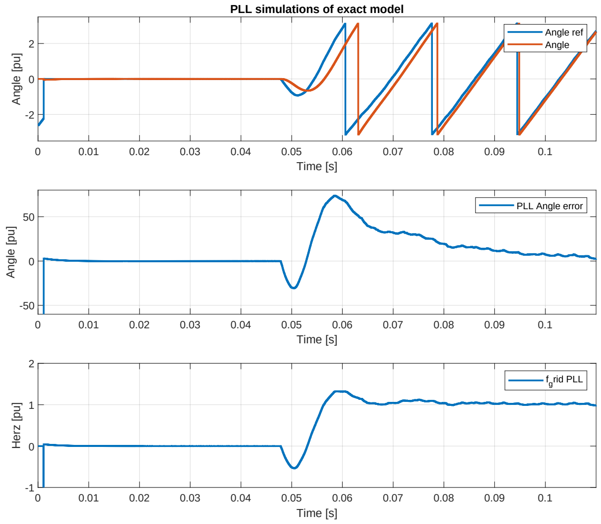


Figure 4.9: Simulation with the exact model for testing PLL, with $f_f = 20Hz$ and $\zeta = 1.4$, which gives $T_i = 22.3ms$ and $K_p = 0.93$

It is observed with decreasing value of K_p the angle error gets bigger. Also the highest angle error occurs at the values $K_p = 0.93$ and $T_i = 0.0223$.

4.2 Simulation of model in PESC platform

Finally, the discretized equations for implementing the PESC platform on the FPGA-module is derived by discretizing the mathematical model employed in the emulated continuous-time Simulink model. This discretized equations serve as the foundation for constructing the FPGA-based emulator, PESC platform.
In this section, the emulated model have been developed and put together for the FPGA to receive the binary compositions for the model. The release builder tool is used for zipping a project file and reboot the processor board for correct project running. The phase-locked loop for the model is mainly investigated. Further, a series of diverse scenarios have been tested in the PESC control platform to investigate their respective system behaviors. These scenarios include: (1) the starting state with the breaker open, (2) the starting state with the breaker closed with and without active dampening, and (3) the simulation involving a load increase. The first and second scenario is without a connected load which can be observed in the plots. The affect of the Active dampening is investigated. A simulation with a load change is also performed to observe the behavior of the model.

4.2.1 Parameter representation

Parameter	Value
V_{sN}	140 V
I_{sN}	11.5 A
S_{base}	2788.6 kVA
Z_{base}	7.0286 A
fn	78.5 Hz
capasitance addmittance, y_c	0.1 pu
inverter reactance, x_1	0.1 pu
Grid reactance, x_2	0.1 pu
inverter resistance, r_1	0.0001 pu
Grid reactance, r_2	0.0001 pu
Active dampening resistance, r_D	2.0 pu

Table 4.4: This table shows the base values used to calculate the per unit values of the passive components, and the Per unit values of the passive components of the LCL-filter used in the PESC control platform during the simulations.

Parameter	Value
capasitance addmittance, y_c	0.70286 F-1
capasitance addmittance, C	1.425 mF
inverter reactance, x_1	0.70286 ø
inverter inductance, L_1	1.425 mH
Grid reactance, x_2	0.70286 ■
Grid inductance, L_2	1.425 mH
inverter resistance, r_1	0.70286 mΩ
Grid reactance, r_2	0.70286 mΩ
Active dampening resistance, r_D	14.0572 Ω

Table 4.5: This table shows the passive components and their real values used in the PESC control platform during the simulations.

Table 4.4 shows the base values used to calculate the per-unit values of the passive components, and the per-unit values of the passive components of the LCL-filter used in the PESC control platform during the simulations. To be able to calculate the resonance frequency, the actual values is converted from the pu values. The following equation is used to convert the per-unit values, as depicted in Table 4.4, into their corresponding actual values. The resulting actual parameters are displayed in Table 4.5.

$$Actual\ value = Pu\ value \cdot Base\ value$$

4.2.2 Tuning the PLL

Initially, the optimal parameters for the PLL is determined by conducting experiments with various parameters using three distinct parameter sets specifically designed for the PLL. The parameters utilized in this evaluation are presented in Table 4.6. Similar to the exact model in Simulink, $zeta$, f_f and f_n are chosen where K_p and T_i is calculating using the equations shown in Equation 2.52. Subsequently, the simulation results of the PLL within the PESC control platform are depicted in Figure 4.10, 4.11 and 4.12. The fundamental frequency, f_n, is different from the PLL simulation in the Simulink model, where the fundamental frequency was set to 60 Hz instead of 78.7 HZ, which is used here.

	ζ	f_n[Hz]	f_f[Hz]	K_p	T_i[s]
Figure 4.10	1.6	78.7	70	2.85	0.00727
Figure 4.11	1.6	78.7	40	1.63	0.01273
Figure 4.12	1.6	78.7	23	0.93	0.02214

Table 4.6: Table of the simulated ζ, nominal frequency, f_n, fault frequency, f_f, Kp and Ti values for the PLL in the PESC platform.

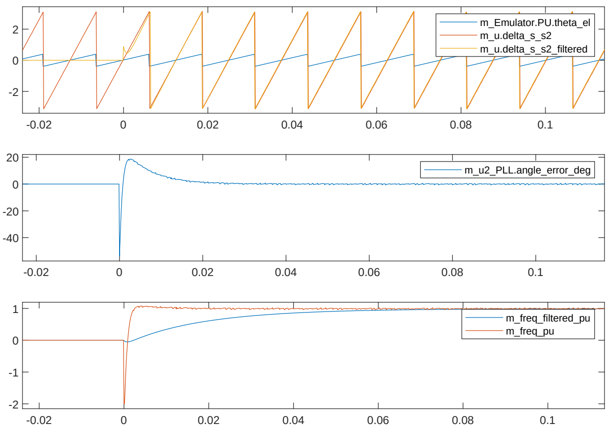


Figure 4.10: Simulation with PESC controller for testing PLL, with $f_f = 70Hz$ and $\zeta = 1.6$, which gives $T_i = 7.27ms$ and $K_p = 2.85$

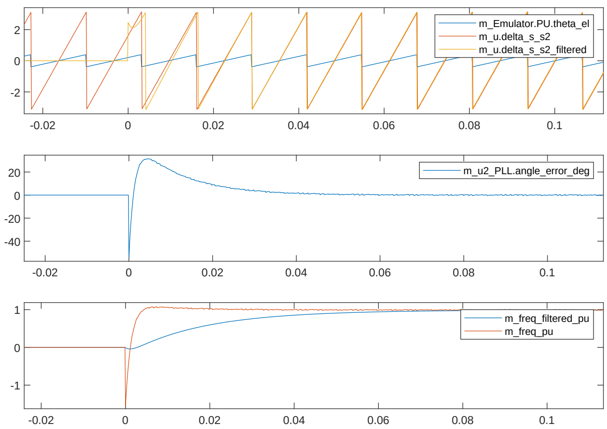


Figure 4.11: Simulation with PESC controller for testing PLL, with $f_f = 40Hz$ and $\zeta = 1.6$., which gives $T_i = 12.73ms$ and $K_p = 1.63$

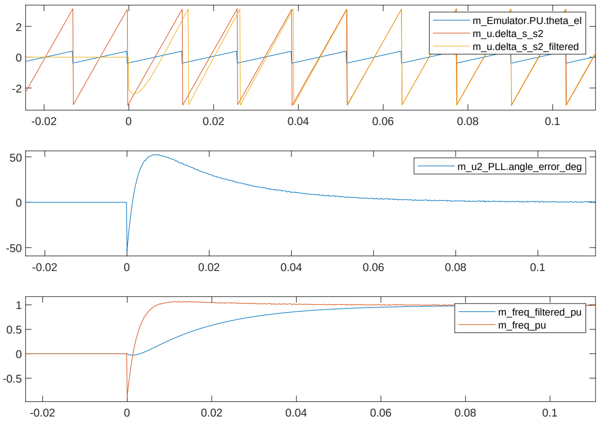


Figure 4.12: Simulation with PESC controller for testing PLL, with $f_f = 23Hz$ and $\zeta = 1.6$., which gives $T_i = 22.14ms$ and $K_p = 0.93$

All the three plots illustrate the performance of the PLL under different parameter configurations. In the uppermost plot, the red line represents the angle reference, while the yellow line represents the angle of the capacitor voltage. A smaller angle error, indicated by a reduced discrepancy between the yellow and red lines, indicates a more desirable performance of the PLL. The lower plot showcases the angle error represented by the blue curve. Specifically, Figure 4.10 demonstrates the simulation results of the PLL with $K_p = 2.85$ and $T_i = 7.27$ ms. Similarly, Figure 4.11 exhibits the simulation outcomes with $K_p = 1.63$ and $T_i = 12.73$ ms. Finally, Figure 4.12 presents the simulation outcomes achieved with $K_p = 0.93$ and $T_i = 22.14$ ms

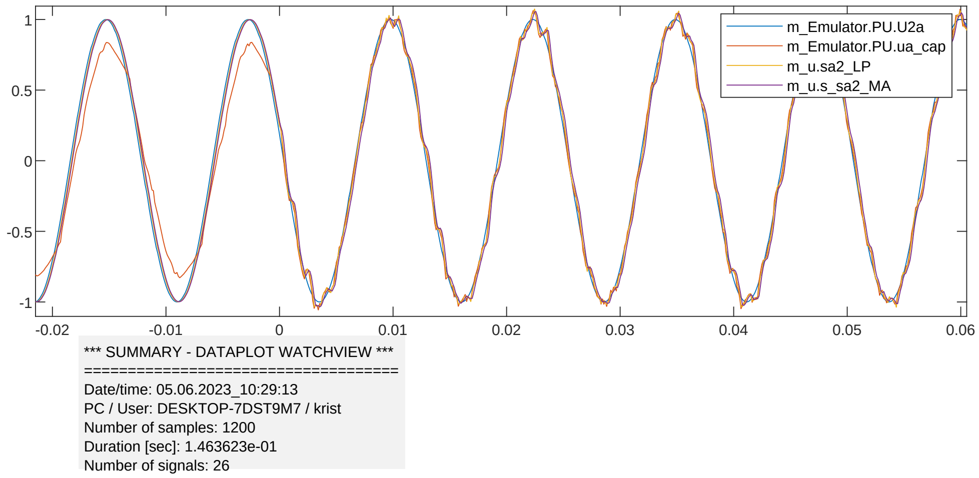
The analysis of the simulation results depicted in Figure 4.10, Figure 4.11, and Figure 4.12 reveals notable observations. Firstly, Figure 4.10 exhibits the most rapid response time among the three cases. As the proportional gain K_p decreases and the integral time constant T_i increases, the response time noticeably increases. This phenomenon is particularly evident in Figure 4.11, which displays the next best response time and angle difference. However, it is worth noting that Figure 4.12 demonstrates the least desirable performance in terms of the angle difference, with results indicating more pronounced tracking errors.

4.2.3 Testing the model in PESC control platform

To begin, the simulation results for the starting state with the breaker open are presented in Figure 4.15. For each scenario, a consistent set of parameters is observed and recorded. These parameters, visible in Figure 4.15 and applicable to all plots, are presented in the following order from top to bottom: The reference angle and the angle of the capacitor voltage, u_C , The grid voltage, u_2, the converter voltage, u_1, the converter current in the direct axes, i_{1d}, the converter current in the quadrant axes, i_{1q}, the converter current in abc-three-phase representation, i_{1a}, i_{1b} and i_{1c}, the active dampning current in abc-three-phase representation, the active dampening and the angle deviation.

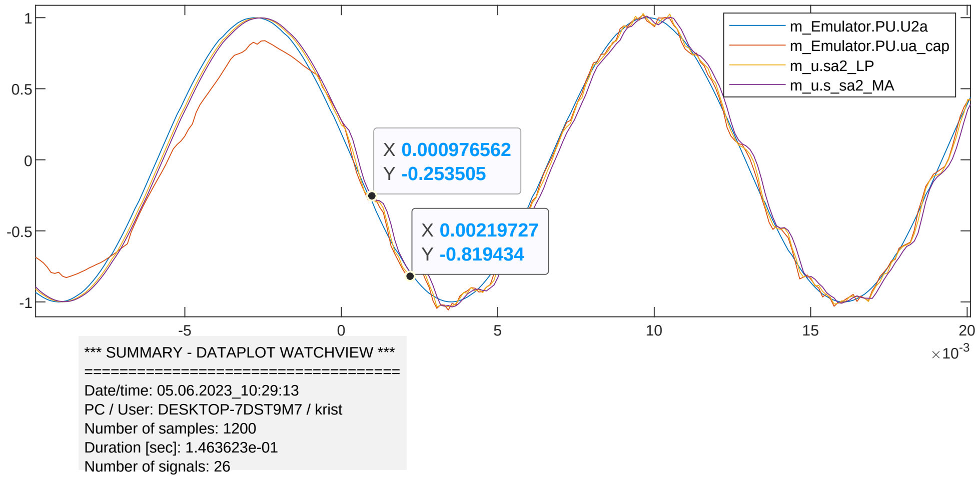
The next simulation investigates the impact of closing the breaker and is depicted in Figure 4.16. It is evident that the closure of the breaker introduces some disturbances to the current and voltages within the system. Consequently, a necessary increase in active dampening is required to mitigate these disturbances effectively. Notably, the angle deviation diminishes to approximately zero, indicating a successful connection between the grid and capacitor. In this simulation, the active dampening is configured with a value of rD = 2 p.u., ensuring enhanced stability and improved performance.

To evaluate the impact of active dampening, a simulation is conducted without the utilization of active dampening, as presented in Figure 4.17. The absence of active dampening makes the small deviations more noticeable, thereby highlighting their significance within the system.



```
*** SUMMARY - DATAPLOT WATCHVIEW ***
====================================
Date/time: 05.06.2023_10:29:13
PC / User: DESKTOP-7DST9M7 / krist
Number of samples: 1200
Duration [sec]: 1.463623e-01
Number of signals: 26
```

Figure 4.13: a zoom up picture of the grid voltage and a closeup of the oscilation occuring after the breaker is closed.



```
*** SUMMARY - DATAPLOT WATCHVIEW ***
====================================
Date/time: 05.06.2023_10:29:13
PC / User: DESKTOP-7DST9M7 / krist
Number of samples: 1200
Duration [sec]: 1.463623e-01
Number of signals: 26
```

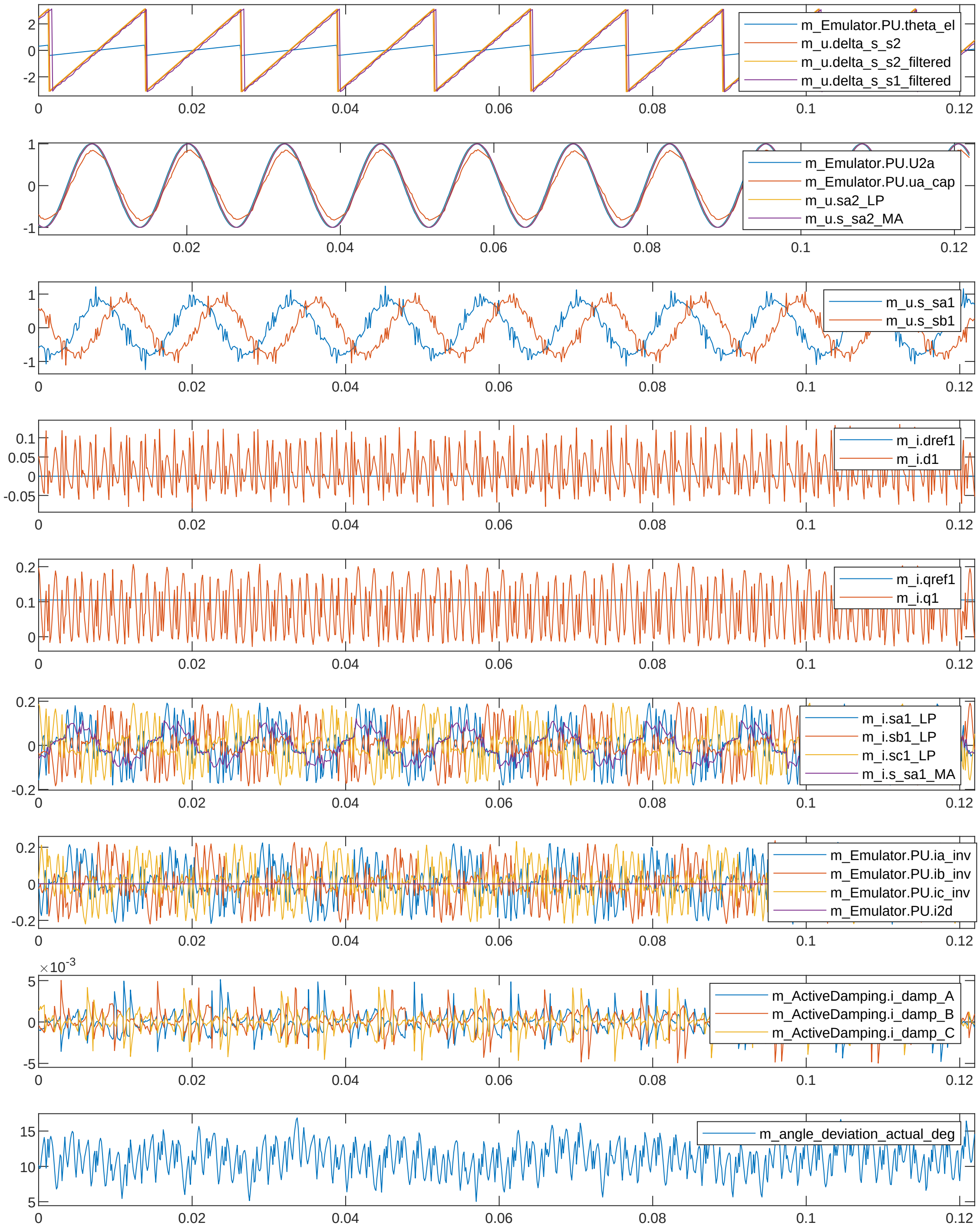
Figure 4.14: A closeup on how the frequency of the oscillation is measured.

The biggest difference observed is the presence of oscillations in the grid voltage after closing the breaker. These oscillations are attributable to frequencies within the system. Given that this system is not interconnected with a large and complex network, it is likely that the frequency arises from the switches in the converter. Figure 4.13 shows a closeup of the grid voltage were the oscillations are more significant. The frequency of the oscillation is measured in Figure 4.14 where it is calculated:

$$\frac{1}{0.00219727 - 0.000976562} = 819.1967Hz$$

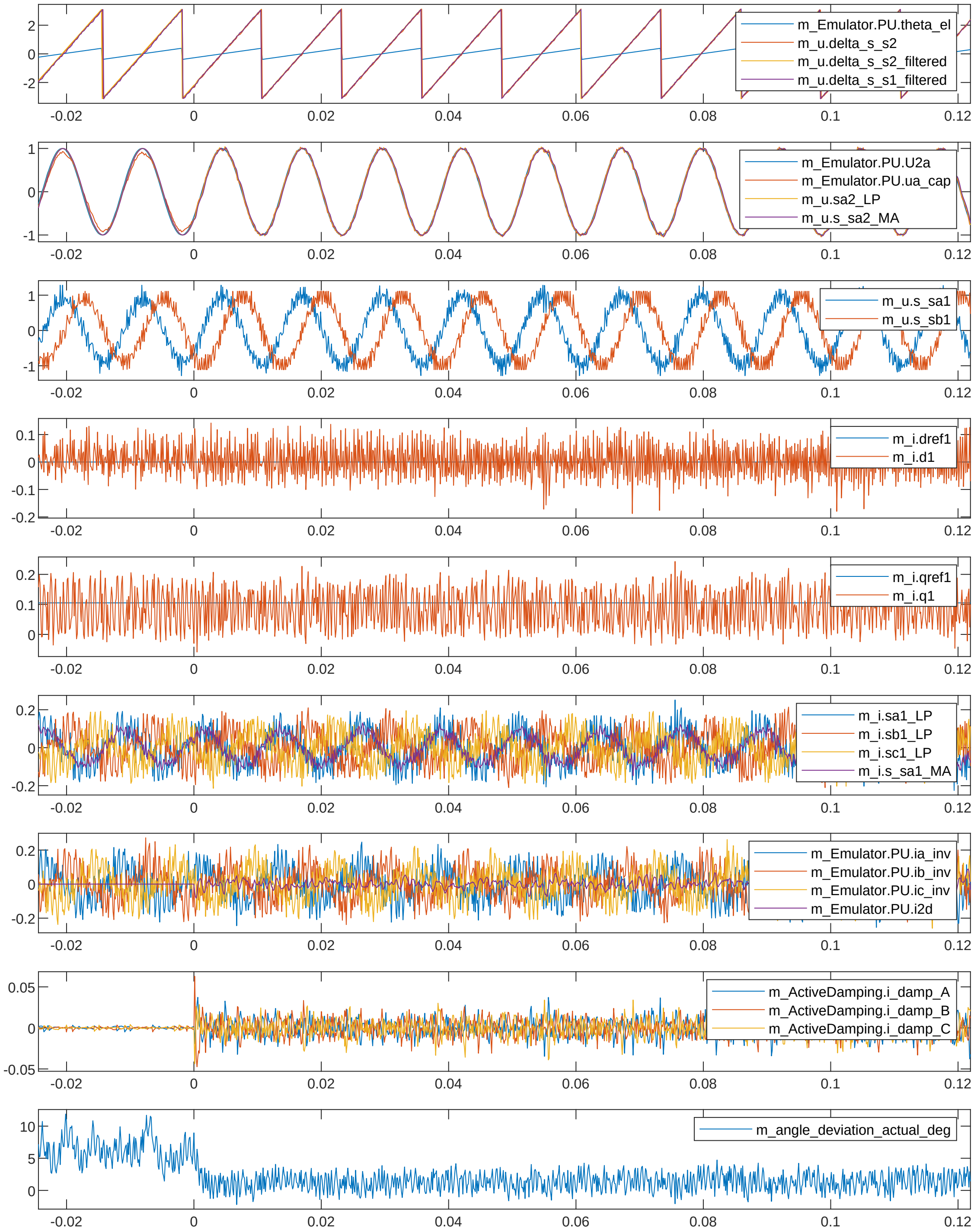
This is discussed more later in section 5.2.

The last simulation shows an increase in the load. To see how the model act with a load, Figure 4.18 shows the simulation. The load change is a change of i_{dref} from 0 to 0.6.



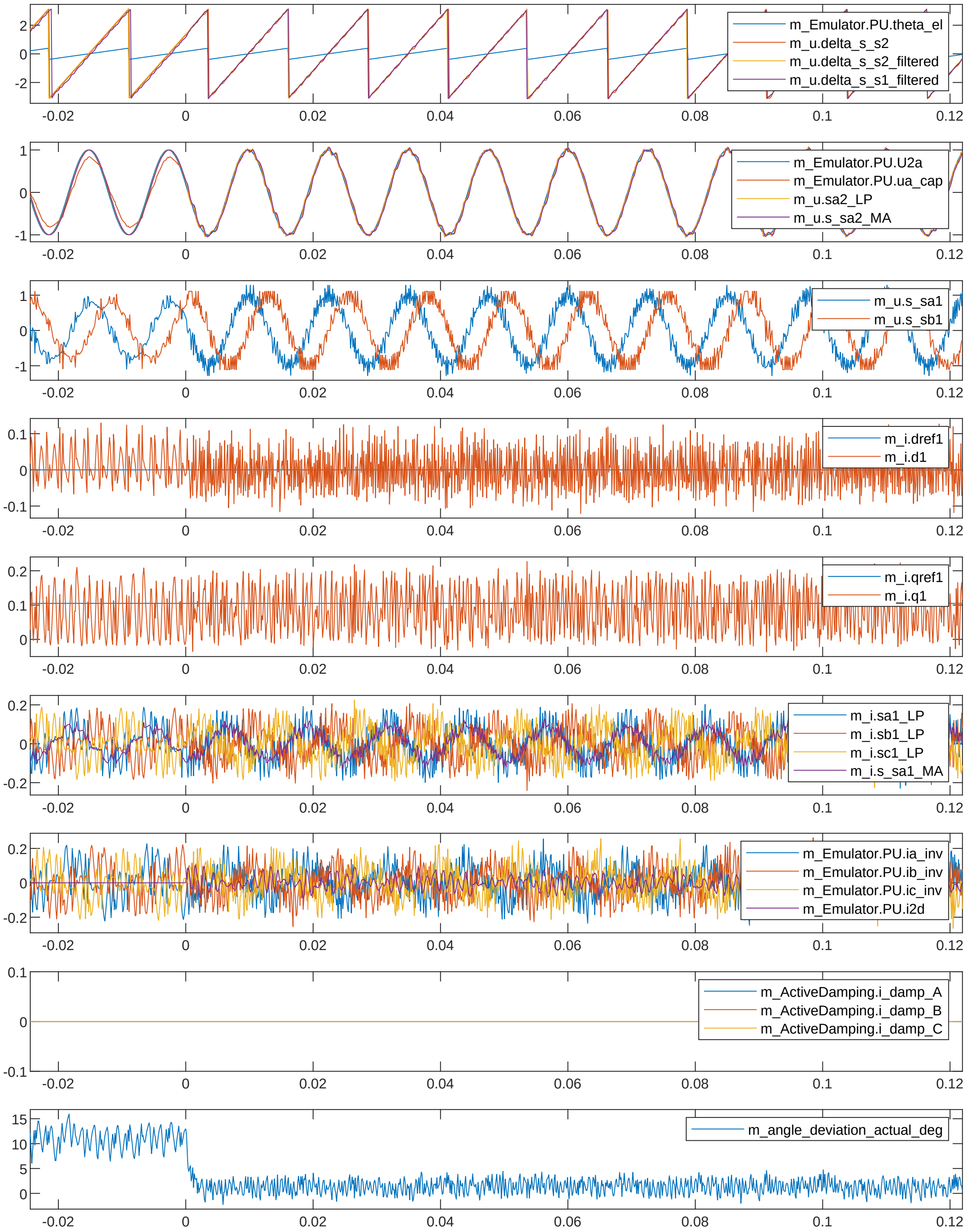
```
*** SUMMARY - DATAPLOT WATCHVIEW ***
===================================
Date/time: 01.06.2023_11:34:48
PC / User: DESKTOP-7DST9M7 / krist
Number of samples: 1000
Duration [sec]: 1.219482e-01
Number of signals: 26
```

Figure 4.15: Simulation with PESC controller of the starting state with the breaker open.



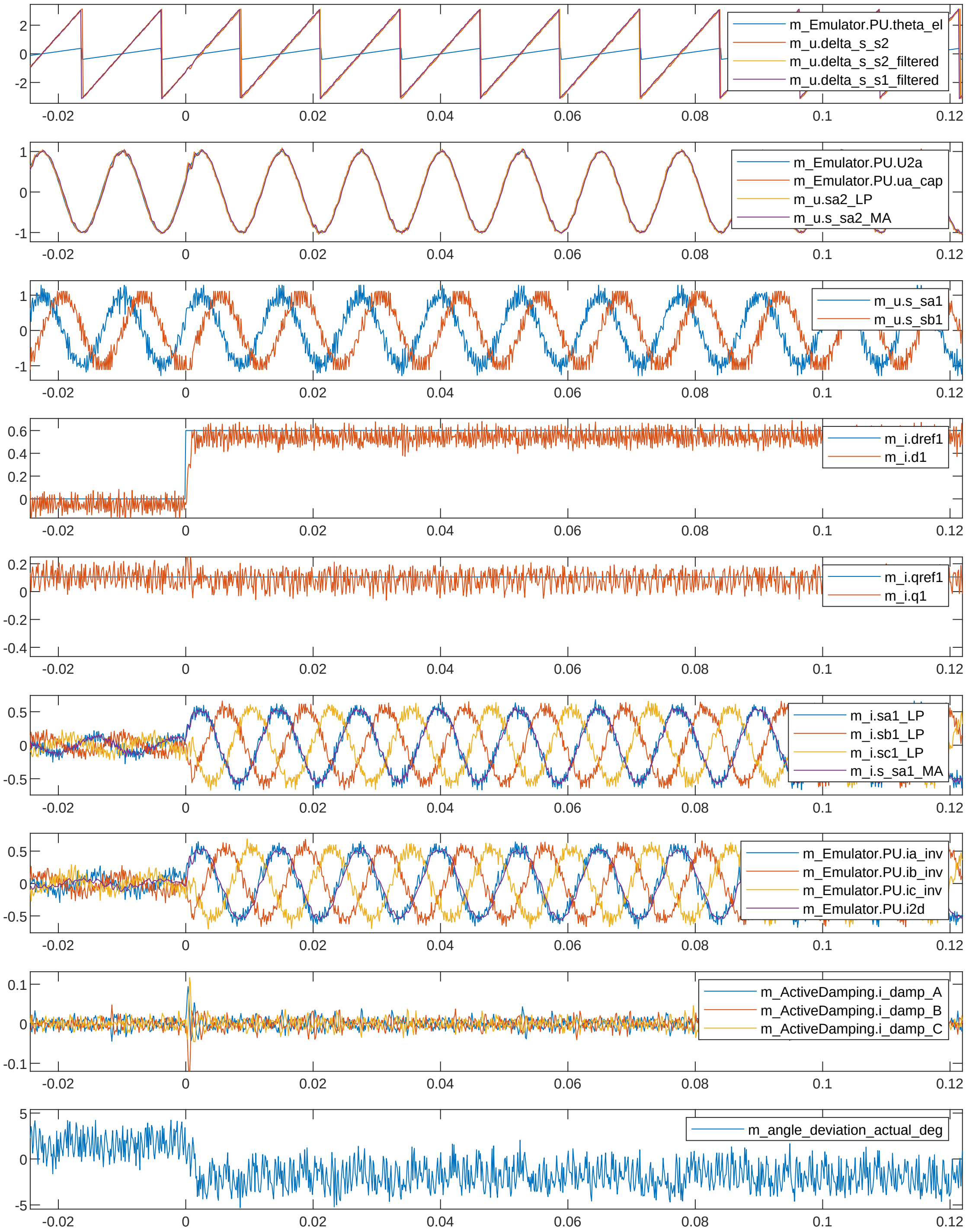
```
*** SUMMARY - DATAPLOT WATCHVIEW ***
==================================
Date/time: 01.06.2023_11:44:10
PC / User: DESKTOP-7DST9M7 / krist
Number of samples: 1200
Duration [sec]: 1.463623e-01
Number of signals: 26
```

Figure 4.16: Simulation with PESC controller of the starting state when the breaker close including the active dampening with a $r_D = 2p.u.$ and Iqref = yc=1/xc=0.1 pu.



```
*** SUMMARY - DATAPLOT WATCHVIEW ***
===================================
Date/time: 01.06.2023_11:41:38
PC / User: DESKTOP-7DST9M7 / krist
Number of samples: 1200
Duration [sec]: 1.463623e-01
Number of signals: 26
```

Figure 4.17: Simulation with PESC controller of the starting state when the breaker close without active dampening, Iqref = yc=1/xc=0.1 pu.



```
*** SUMMARY - DATAPLOT WATCHVIEW ***
===================================
Date/time: 07.06.2023_15:32:12
PC / User: DESKTOP-7DST9M7 / krist
Number of samples: 1200
Duration [sec]: 1.463623e-01
Number of signals: 26
```

Figure 4.18: Simulation with PESC controller, the breaker is closed. Active dampening is included and a load change for Idref from 0 to 0.6 occurs at t=0.

Chapter 5

Discussion of results

5.1 Bode-plot

In this section, the Bode-plot for the PLL in the exact model and the PESC control platform are analysed. The transferfunction used for the Bodeplots is described in the theory given in Equation 2.50.

5.1.1 Exact Simulink model

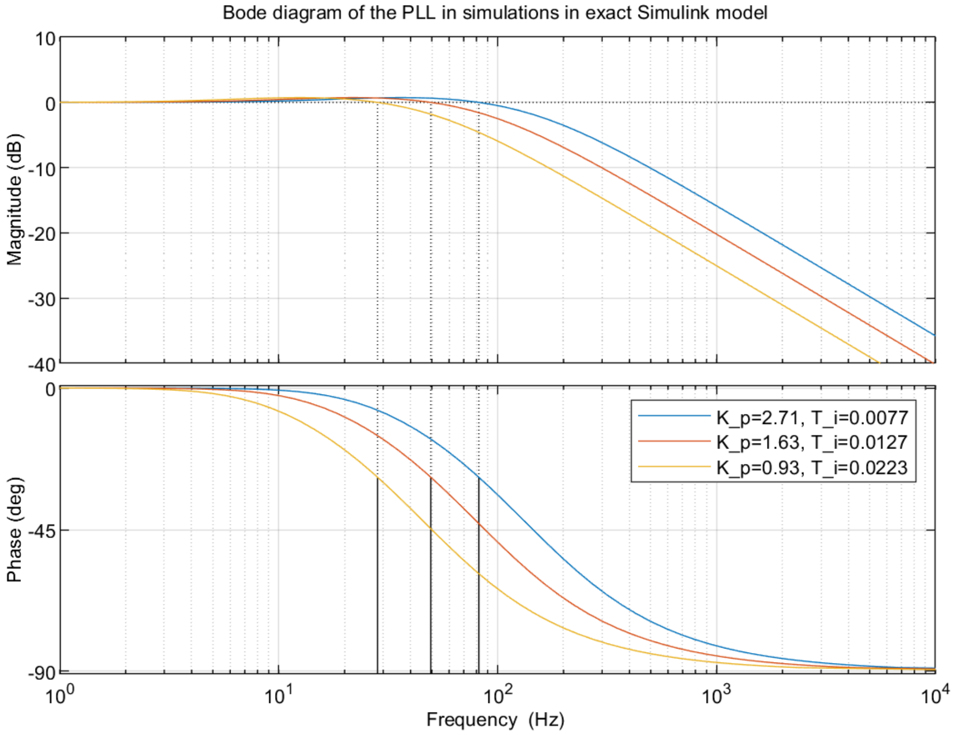


Figure 5.1: Bode-plot of the values in PLL used for the exact model.

To obtain a clearer understanding of how well the parameters fit the PLL, the Bode plot is analyzed. The process begins by plotting the Bode plot of the transferfunction for the exact model, as depicted in Figure 5.1. The corresponding parameter values are displayed in Table 4.3. In the power system, the nominal frequency is set to 60 Hz.

The graph consists of three lines: blue, red, and yellow. The blue line represents the values with $K_p = 2.71$ and $T_i = 7.7$ ms, the red line corresponds to $K_p = 1.63$ and $T_i = 17.7$ ms, and the yellow line represents $K_p = 0.93$ and $T_i = 22.3$ ms. From the simulations it was observed that a higher value of K_p leads to better performance. The Bode plot demonstrates that the cutoff frequency increases with an increasing value of K_p. The PLL functions as a first-order filter, and the crossover frequency marks the point where the incoming signal's amplitude no longer rises or falls significantly in higher frequencies. In this case, the crossover frequency cannot be lower than 60 Hz, which is the nominal frequency representing the fundamental frequency of the grid. Conversely, if the crossover frequency is too high, unwanted frequencies can pass through and potentially damage the components.

When plotting the transfer functions for the PLL in the Exact model, it was observed that the slowest response occurred with the parameters $K_p = 0.93$ and $T_i = 0.0223$. Referring to the Bode diagram in Figure 5.1, it is observed that parameters for the PLL must be chosen carefully. This is due to the cutoff frequency being lower than 60 Hz, the nominal frequency. Unfortunately, filtering out the nominal frequency is an undesirable outcome.

5.1.2 PESC control platform simulations

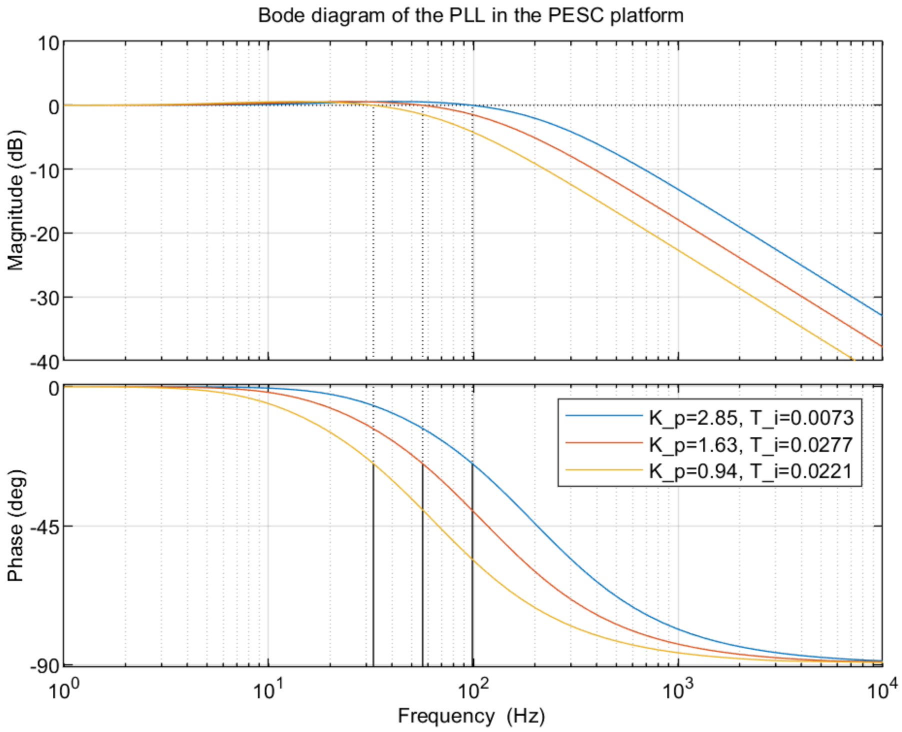


Figure 5.2: AFF plot of the values in PLL used for the PESC controller.

The Bode diagram in Figure 5.2 illustrates the simulation results in the PESC control platform, showcasing various parameter values. The blue graph corresponds to $K_p = 2.85$ and $T_i = 7.3$ ms, the red graph represents $K_p = 1.63$

and $T_i = 12.8$ ms, and the yellow graph signifies $K_p = 0.94$ and $T_i = 22.1$ ms.

Similar to the exact model, the simulation with the highest K_p value gives fastest response time. Analysing Figure 5.2, it is noticeable that these parameters yield the highest cutoff frequency, approximately 100 Hz. The red graph has a cutoff at 58 Hz, and the yellow graph has an approximate cutoff of 31 Hz. Considering that the nominal frequency for the grid is 78.8 Hz, it is crucial to ensure that the cutoff frequency remains within this range. Thus, the chosen parameters for the rest of the simulations are $K_p = 2.85$ and $T_i = 7.3$ ms.

In the case of the exact model, the fundamental frequency is lower than for the PESC control platform. The f_f and ζ need to be chosen to be lower is to get the same K_p and T_i as in the PESC simulations. This makes sense, since with higher frequency, higher dampening ratio is required to stabilize the frequency. With the higher frequency as for the PESC control platform, it is achieved a higher cut off frequency.

If the cutoff frequency is too high, higher harmonics have a chance to pass through as unwanted frequencies. On the other hand, if the cutoff frequency is too low, it filters out the nominal frequency, which is also undesired as power swings are intended. Hence, the conclusion is that the K_p value is desired to be as high as possible, but it is crucial to ensure that unwanted frequencies do not slip through, since the power electronics are vulnerable to disturbances.

5.2 Active dampening

By conducting a comparison between the simulations with and without active dampening, it becomes evident that the inclusion of active dampening eliminates the voltage oscillation. When the converter is connected to the grid, numerous disturbances can arise, influencing the frequencies, output voltages, and currents. Switching breakers, among other factors, can generate transients in voltage or current. To mitigate the harmonics originating from the grid and prevent component damage, active dampening is incorporated into the model.

However, if some of the frequencies coincide with the resonance frequency, the oscillation persists. Figure 4.16 illustrates the impact of active dampening. At time = 0, the breaker connected to the grid is closed, and the converter is linked to the grid. A noticeable oscillation is observed in the grid-side voltage. The simulation done in the PESC controller platform is not connecting to a real-world power network consisting of a lot of different components and a complex system. It is likely that the observed oscillation arises from the converter's switching frequency. If the frequency aligns with the resonance frequency of the LCL filter, an oscillation occurs. If this occurs only once, the oscillation will eventually cease. However, as the oscillation persists, it is probable that the frequency regularly coincides with the resonance frequency. To verify if the oscillation corresponds with the resonance frequency, a comparison is made between the ressonance frequency of the LCL-filter and the frequency of the oscillation occurred on the voltage. From Figure 4.14 the frequency of the oscillation is measured and calculated to 819 Hz. The resonance frequency of the LCL filter is calculated using Equation 5.1[14].

$$\omega_{res} = \sqrt{\frac{(L_1 + L_2)}{L_1 L_2 C_f}} \tag{5.1}$$

where L_1 represents the inductance on the inverter side, L_2 represents the inductance on the grid side, and C_f

represents the capacitance of the LCL-filter. The parameters used in the PESC control platform are displayed in Table 4.5. Substituting the values into Equation 5.1, the resonance frequency of the LCL-filter is calculated as 992 Hz. The deviation of 173 Hz could be attributed to measurement errors. The accuracy of the measurements and the imprecise determination of the bottom of the wave could contribute to the measurement errors. Despite the slight discrepancy in frequencies, it can be concluded that the frequencies of the harmonics resulting from the absence of active dampening closely resemble the resonance frequency of the LCL filter.
Reference frequency and actual frequency are also dependent on dampening[10].

Chapter 6

Conclusions

The AFE, connected to an LCL filter, is simulated in Smulink and analyzed. Two main methods for controlling the power electronic IGBT switches are pulse-width modulation (PWM) and hysteresis control. In this Master's thesis, the AFE is implemented using a hysteresis controller with a grid-following control scheme. The frequency and voltage are regulated by conventional synchronous machines, making the grid-following scheme adjusting for the active and reactive power based on the given voltage and frequency in the power system. Additionally, it is convenient to test the controller for grid-forming scheme and with pulse-width modulation to align with the modern paradigm of renewable energy sources. Furthermore, the PLL in the system was also tested. The emulated model for the PESC control platform has been successfully developed. Simulations play a crucial role in evaluating the performance of the control platform and analyzing its response under different operating conditions. Functions such as the phase-locked loop (PLL) and the effect of active damping are tested. Hence, the conclusion for the PLL is that the K_p value is desired to be as high as possible, but it is crucial to ensure that unwanted frequencies do not slip through, since the power electronics are vulnerable to disturbances. To delete the harmonics coming from the grid and avoid destroying the components, active dampening is included in the model.

Ultimately, the successful implementation of this FPGA-based control platform will pave the way for improved motor drive applications and potentially other domains that require real-time control and high-performance processing. The FPGA simulation model provide with a fast simulation to achieve real-time. This is highly required in design procedures, for neglecting faults, big costs etc. An important thing to consider for FPGA is the numeric representation and the scaling. The advantage of the Pu model is the broader application for the connected converter representation, as there is no need for different models for the higher rated voltage application and the lower voltage applications when the pu modelling is used.

6.1 Further work

Based on the work presented in this future work is proposed:

- In this work, a hysteresis current controller on firmware was used for the inner control loop. An alternative approach could be to move the current controllers into software domain, and replace the hysteresis controller with voltage-controlled PWM-based operation.
- The control system should be complemented with more functionality in the outer control-loop such as DC-link voltage regulation and droop control.
- When the basic functionalities are in place, looking into auxiliary services such as selective harmonic elimination is relevant.
- After the AFE firmware and software control system has been sufficiently tested on the PESC control platform, the next step is further testing using a physical experimental setup.

Bibliography

[1] Frede Blaabjerg et al. “Overview of Control and Grid Synchronization for Distributed Power Generation Systems”. In: *IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS* 53.5 (2006).

[2] Amir Arasteh, Lorenzo Zeni, and Nicolaos A.Cutululis. “Fault ride through capability of grid forming wind turbines:A comparison of three control schemes”. In: *IET Renew. Power Gener.* 16.1866-1881 (2022).

[3] Robert Rosso et al. “Grid-forming Converters: Control Approaches Grid-synchronization, and Future Trends-A Review”. In: *IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS* 2 (2021).

[4] Aravinda Perera et al. *A Design Method of an Embedded Real-Time Simulator for Electric Drives using Low-Cost System-on-Chip Platform*. 2021.

[5] *Conversation with Professor Roy Nilsen*. Took place 21 march 2023.

[6] Anders Moldskred. “FPGA-Based Real-Time Emulator of Permanent Magnet Synchronous Motor”. Master Thesis. Norwegian University of Science and Technology, Dept. of Electric Power Engineering, June 2020.

[7] Roy Nilsen. *TET4120 Electric Drives*. NTNU, Deparment of Electric Energy. Spring 2018.

[8] R. Sureshkumar. *10.001: Numerical Solution of ordinary Differential Equations*. Accessed: 07.05.2023. January 15, 1997. URL: http://web.mit.edu/10.001/Web/Course_Notes/lec24.pdf.

[9] *Conversation with Professor Roy Nilsen*. Took place 25 of may 2023.

[10] Jens G. Balchen, Trond Andresen, and Bjarne A. Foss. *Reguleringsteknikk*. 6th ed. Institutt for teknisk kybernetikk, NTNU, 2016.

[11] *FPGA Fundamentals: Basics of Field-Programmable Gate Arrays*. Last updated: 03.05.2023 Accessed: 04.06.2023. URL: https://www.ni.com/en-no/shop/electronic-test-instrumentation/add-ons-for-electronic-test-and-instrumentation/whatis-labview-fpga-module/fpga-fundamentals.html.

[12] Viljar Fjellanger. “On-line voltage estimation during sensorless control of an induction machine drive”. Master Thesis. Norwegian University of Science and Technology, Dept. of Electric Power Engineering, June 2019.

[13] Roy Nilsen. *Technical report: System Introduction*. May 25, 2021.

[14] M. Ben Sa¨ıd-Romdhanea et al. “Simple and systematic LCL filter design for three-phase grid-connected power converters”. In: *Mathematics and Computers in Simulation* 130 (2016), pp. 181–193. URL: https://www.sciencedirect.com/science/article/pii/S0378475415002116.

Appendices

A Appendix A-simulink models

A.1 Simplified model in simulink

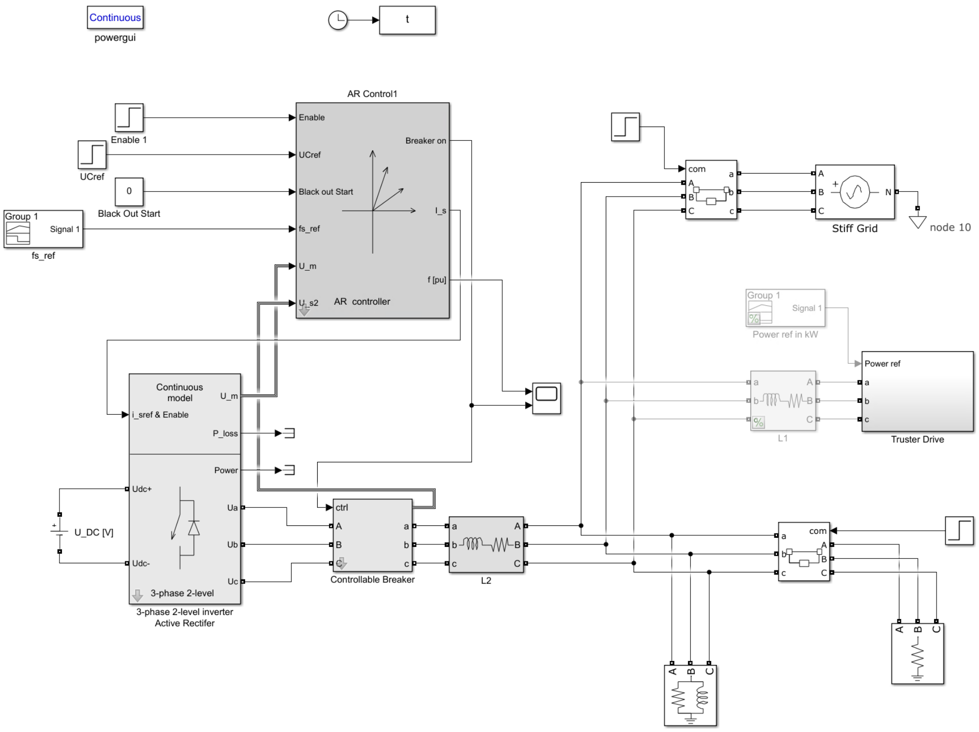


Figure A.1: Top level of the simplified model of the AEF

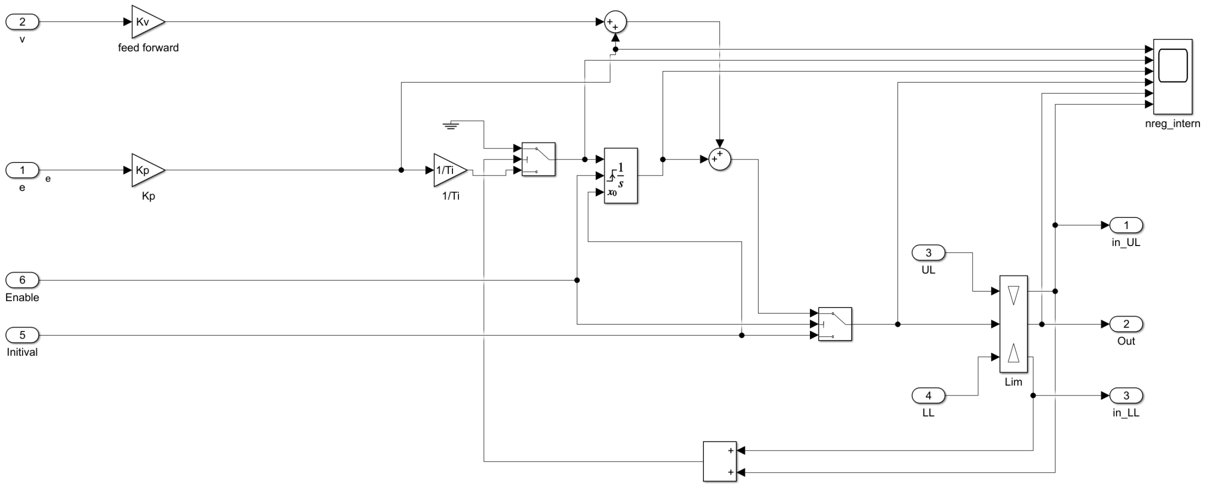


Figure A.2: the PLL of the AEF

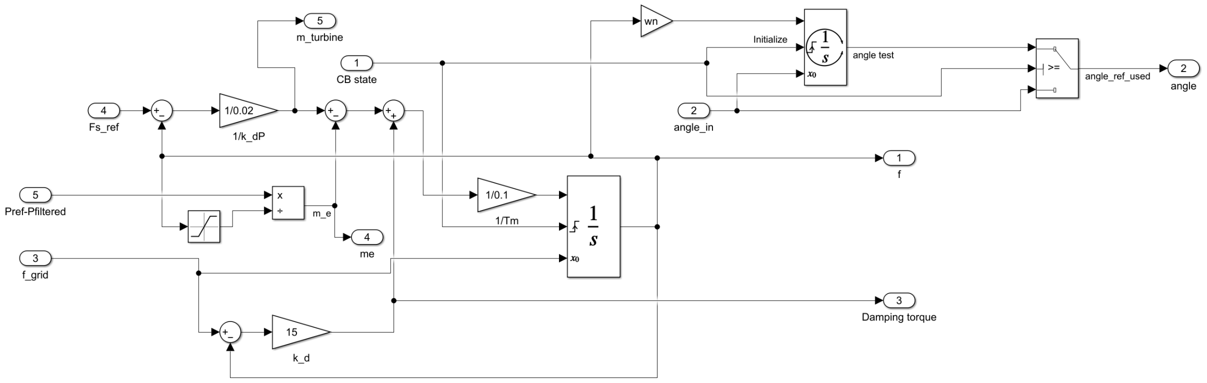


Figure A.3: The Controller of the AEF implemented as the virtual synchrounous machine

A.2 emulated continious model of the LCL filter

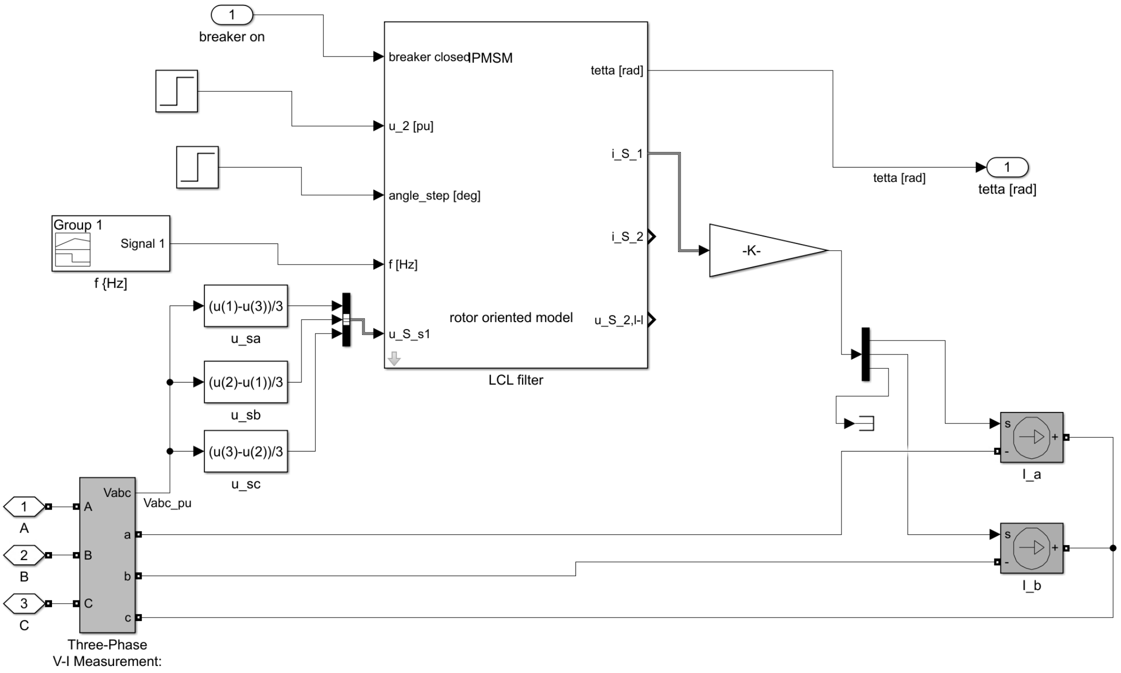
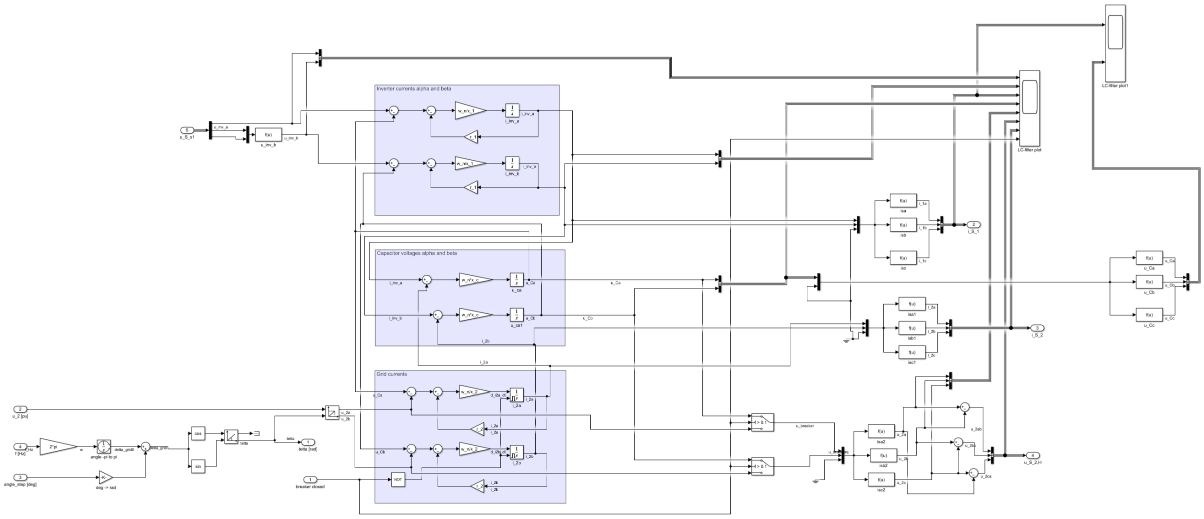


Figure A.4: Top level of the simplified model of the AEF



Inverter currents alpha and beta
Capacitor voltages alpha and beta
Grid currents
LC-filter plot
LC-filter plot1

Figure A.5: simulink blocks of the LCL filter

A.3 Nøyaktig modell

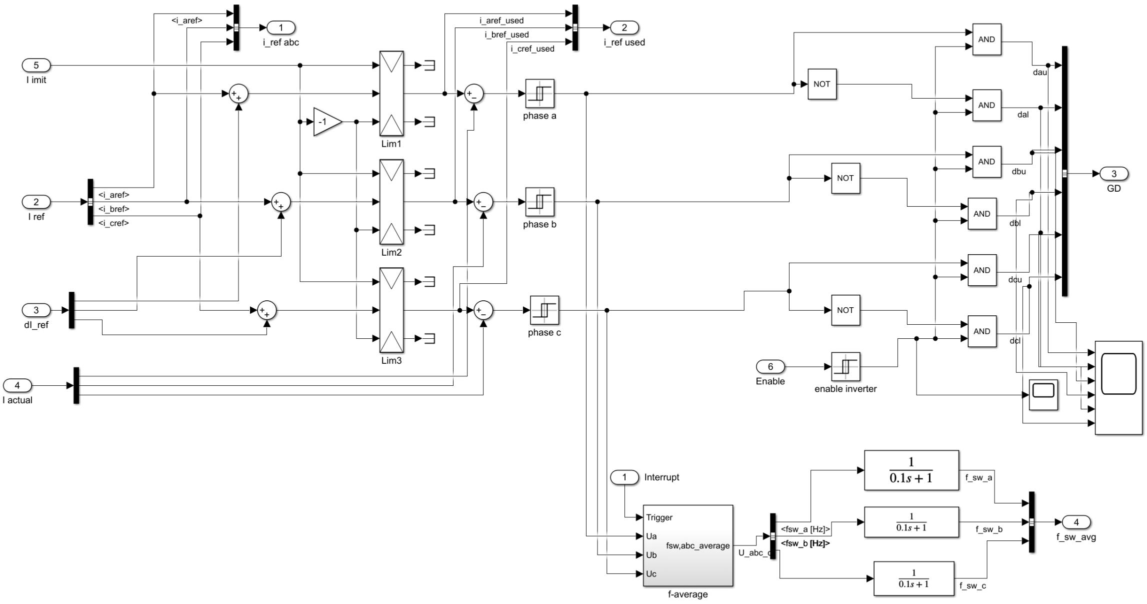


Figure A.6: Hysterises controller

NTNU

Norwegian University of
Science and Technology