NTNU
Norwegian University of
Science and Technology

Circulating Current Control for the Modular Multilevel Converter

Gard Rødal

Master of Energy and Environmental Engineering
Submission date: November 2017
Supervisor: Lars Einar Norum, IEL
Co-supervisor: Anirudh Budnar Acharya, IEL

Norwegian University of Science and Technology
Department of Electric Power Engineering

NTNU
Norwegian University of
Science and Technology

Circulating Current Control of the Modular Multilevel Converter

Gard Lyng Rødal

Master of Energy and Environmental Engineering

Submission date: November 2017

Supervisor: Lars Einar Norum

Co – supervisor: Anirudh Budnar Acharya

Norwegian University of Science and Technology

Department of Electric Power Engineering

Preface

This master thesis contains the work of my final semester at the Department of Electric Power Engineering at Norwegian University of Science and Technology (NTNU).

My interests within the scientific field has only grown during my 5 years at NTNU. My passion has focused on electric power engineering, and the journey from early introduction to the fundamentals of electromagnetism to complex control of power electronics has been a challenge, but most of all a great pleasure.

I would like to thank professor Lars Einar Norum. His experience and knowledge within the fields of power electronics has been of great help. PhD. candidate and co – supervisor Anirudh Acharya has provided great assistance with guiding the direction of the thesis and providing academic support. He has been available whenever I have asked and his pedagogical abilities has been indispensable. They have both opened new worlds and increased my interest within digital control of power electronics. I thank them for that.

I would like to thank my parents for opening the door to the world of science. They have provided immense support through challenging times at NTNU. Together with my brother and sister, they have encouraged and motivated me throughout my complete studies at NTNU. I have gotten great friends, which have made my study years fun and unforgettable. Finally, I would like to thank my girlfriend for providing support on several fields through intense master thesis work.

Trondheim, November 7th 2017.

Gard Lyng Rødal

1. Introduction 1
1.2. Problem Description 2
1.3. Objective and Outline of Thesis 2
Acronyms 3
2. Theory of Operation 4
2.1. Mathematical Model 5
2.2.1. Analytical Derivation of the Steady State Sum-Capacitor Voltage Ripple 9
2.2. The Circulating Current 12
2.3. MMC Modulation Techniques 14
2.3.1. Phase – Disposition PWM 15
2.3.1.1. Harmonic Generation with PD – PWM 16
3. Control Techniques 18
3.1. Discrete Modelling 18
3.1.1. Analog to Digital Conversion (ADC) 19
3.1.2. Digital to Analog Conversion (DAC) 20
3.1.3. Continuous to Discrete Conversion 20
3.1.3.1. Zero – Order Hold 21
3.1.3.2. Tustin – transform 21
3.1.4. Selection of sampling time 22
3.2. System Delays 23
3.3. Voltage Control 27
3.3.1. Direct Voltage Control 27
3.3.2. Indirect Voltage Control 27
3.3.3. Hybrid Voltage Control 28
3.4. Submodule Capacitor Voltage Sorting 29
3.5. Circulating Current Control for Direct Voltage Control 31
3.5.1. Proportional Integral (PI) Control 32
3.5.2. Proportional Resonant (PR) Control 33
3.5.2.1. Discrete – Time Realization of the PR Controller 37
3.5.3. Repetitive Control 39
3.5.3.1. Internal Model Principle 39
3.5.3.2. Repetitive Controller Structures 42
3.5.3.2.1. Plug – In Repetitive Control with series PI control (RC System 1) 42
3.5.3.2.2. Plug – In Repetitive Control with parallel PI control (RC System 2) 46
3.5.3.3. Error Convergence of the Repetitive Controller 47

3.5.3.4. Repetitive Controller Elements 48
3.5.3.5. Design of the Nominal Controller 50
3.6. Output Current Control 55
4. Simulation 57
4.1. Series Repetitive Control (RC System 1) 57
4.2. Parallel Repetitive Controller (System 2) 63
4.3. Proportional – Resonance Control 64
4.4. Simulation Results 66
4.4.1. Series Repetitive Control (RC System 1) 66
4.4.2. Varying the Repetitive Gain 72
4.4.3. Steady – State RMS Error with varying Parameters 73
4.4.4. Series vs. Parallel Structure 75
4.4.5. Repetitive Controller vs. Proportional Resonant Controller 76
4.4.6. Optimal Design of Series Repetitive Controller 79
4.4.6.1. Output Data 81
4.4.7. Frequency Content 82
4.4.7.1. Circulating Current 82
4.4.7.2. Submodule Capacitor Voltages 84
4.4.7.3. Output Data 86
5. Algorithm Implementation 88
5.1. Software Implementation 90
5.1.1. Processor – in – the – Loop (PIL) 90
5.2. Hardware Implementation 92
5.2.1. Fixed – Point Implementation 93
5.2.2. HDL Code Generation 95
5.3. Xilinx Vivado Design Suite – Design Synthesis and Implementation 97
5.3.1. Proportional – Resonant Controller Implementation 99
5.4. Floating – Point Implementation 101
6. Hardware Design and Laboratory Set Up 104
6.1. The Modular Multilevel Converter 104
6.1.1. MicroZed 104
6.1.2. Interface Card 108
7. Conclusions 109
8. Future Work 111
References 113
Appendix 116

A. Simulink Simulation Block Diagram 116
A. 1. Electrical System 116
A.2. Control and Modulation System 117
A.3. Fixed – point HDL Compatible Simulink Algorithms 118
A.3.1. Control System Overview 118
A.3.2. Triangular Carrier Generation 118
A.3.3. Sinusoidal Reference Generation 120
A.3.4. The Modulator 122
A.3.5. The Submodule Capacitor Voltage Sorting 123
B. Eagle CAD Schematics 124
B.1. Submodule Board and Interface Card Eagle Schematics 124
B.1.1. Submodule Board File 124
B.1.2. Submodule Schematics 125
B.1.2.1. Sheet 1 125
B.1.2.2. Sheet 2 125
B.1.2.3. Sheet 3 126
B.1.2.4. Sheet 4 126
B.1.2.5. Sheet 5 127
B.1.2.6. Sheet 6 127
B.1.3. Interface Board 128
B.1.3.1. Sheet 1 128
B.1.3.2. Sheet 2 129
B.1.3.3. Sheet 3 130
B.1.3.4. Sheet 4 131
B.1.3.4.5. Sheet 5 132
C. IC Specifications 133
D. Mathematical Expressions 134
D.1. Harmonic Content of Phase Voltage using PD – PWM 134

1. Introduction

The demand for energy from renewable sources is rapidly increasing, due to both increasing energy demand worldwide and an increasing need for this energy to be produced with low carbon emissions. To transmit large power quantities from off – shore windfarms and remote solar power plants, HVDC transmission grids is anticipated to increase in use due to higher efficiency at longer distances than HVAC transmission. Self – commutated Voltage Source Converters (VSC) are increasingly used over current sourced Line Commutated Converters (LCC) for the conversion between DC and AC due to superior flexibility, independent active and reactive power control ability, black start capability and smaller converter station foot – print. As the power handling capability of VSC's increase, higher voltage levels are often required to keep losses down. This is a big challenge for conventional two – and three – level VSC due to series connection of power switches causing valve stresses, EMI problems and high harmonic distortion in the output voltage. This increase the need of bulky and costly passive filters and high switching frequencies to meet performance requirements [6].

The Modular Multilevel Converter (MMC) was proposed by Marquardt and Lesnicar in 2003 [7] and has become one of the most attractive topology for voltage source converters, especially for medium/high – voltage and power applications. Its modular design brings several advantages, including increased number of output voltage levels, avoidance of series connection of the power switches in high – voltage applications and the possibility of avoiding DC – link capacitors. This introduce high improvements in the output voltage harmonic distortion, reducing the need for bulky and costly passive filters, and introduce the possibility of keeping a low voltage level over the power switches, reducing EMI and switching losses while increasing efficiency, longevity and redundancy. Furthermore, the topology shows promising results handling faults like DC – link short circuit, by choosing a proper sub – module topology [8, 9].

However, high number of submodules introduce high number of power switches and storage capacitors. The increased component count introduce challenges with regards to reliability, lifetime and cost. The dynamic characteristics introduced by unequal sharing of the DC – link voltage amongst the submodules creates voltage ripples in the capacitors, which also is the origin of the circulating current. This inner dynamic demand higher control efforts and special considerations, and have great impact on converter performance, efficiency and reliability [6, 8].

In this thesis, mathematical descriptions of the MMC working principles, mechanism of the submodule capacitor voltage ripples and how the circulating current is generated, with the control challenges arising from this, are discussed. Focus is put on suppression of the circulating current by means of *repetitive control*. The control algorithm is analyzed and simulated to verify performance, and implemented as software on embedded processor and hardware logic on FPGA to look at execution time, resource use and critical path estimations. The algorithm is compared to the more established control techniques proportional resonant (PR) and proportional integral (PI) control.

1.2. Problem Description

Imbalances between the upper and lower arm voltages give rise to a current comprising mainly low, even multiples of the fundamental frequency harmonic components. This current is termed *circulating current* and circulate within each phase of the converter, not affecting the output voltage and current. If the current is not properly controlled, it will increase peak and RMS values of the phase arm currents, consequently increasing converter power losses and affecting life – time and reliability of power switches and passive components. As the current is sinusoidal in nature and comprising several, low order harmonics, this pose additional challenges of the selected control algorithms.

1.3. Objective and Outline of Thesis

This thesis aims to address the challenge of suppressing the circulating current. Principle of operation and mathematical derivations, common control methods for inner dynamics and software/hardware algorithm implementation is presented, with focus on the control algorithm *repetitive control* for suppression of the circulating current. This algorithm utilizes the internal model principle for suppression of the periodic circulating current. Algorithm derivation and stability is analyzed in the discrete time – domain and performance confirmed with MATLAB Simulink®. Algorithms are investigated and discussed both for sequential software implementation on embedded processor and as hardware logic implementation on FPGA.

A model – based design approach is used in MATLAB Simulink® to define system requirements, create the analyzed algorithms, simulate and verify the design, and automatically generate code for the system. MATLAB Embedded Coder® is used to generate C/C++ code for embedded processor implementation, while MATLAB HDL Coder® is used to generate synthesizable HDL code for IP core generation. The third – party synthesis tool Xilinx Vivado Design Suite is used to interface the designed IP core into a complete processing system and implement the IP core onto a FPGA. Processor – in – the Loop (PIL) simulations is used to look at algorithm execution time in processor, while MATLAB and Vavado generate reports on hardware logic resource utilization and critical path estimations.

Principle of operation and mathematical derivations of inner and outer dynamics are presented in *Theory of Operation.* Derivations and analysis of control algorithms are presented in *Control Techniques*, while their performance is investigated in *Simulation*. The model – based design approach for hardware/software co – simulation, algorithm verification and implementation on SoC is presented in *Algorithm Implementation.* A description of a hardware experimental platform for experimental verification presented in *Hardware Design and Laboratory Set Up.*

Acronyms

ADC	Analog to Digital Converter
APOD – PWM	Alternate Phase Opposition Disposition – PWM
ASIC	Application Specific Integrated Circuit
DAC	Digital to Analog Converter
DFIIt	Direct Form II transposed
DSP	Digital Signal Processing
EMI	Electromagnetic Interference
FPGA	Field Programmable Gate Array
HDL	Hardware Description Language
HVAC	High Voltage Alternate Current
HVDC	High Voltage Direct Current
IEEE	Institute of Electrical and Electronics Engineers
IC	Integrated Circuit
IP	Intellectual Property
LCC	Line Commutated Converter
PD – PWM	Phase Disposition – PWM
PL	Programmable Logic
PS	Processing System
PWM	Pulse Width Modulation
RMS	Root Mean Square
SoC	System on Chip
THD	Total Harmonic Distortion
VHDL	Very high speed integrated circuit HDL
VSC	Voltage Source Converter

2. Theory of Operation

Figure 1 shows a three phase MMC. The converter consists of a number of phase legs coupled to the equal number of phases on the AC – side. Each phase leg consists of an upper and lower arm, where each arm consists of N submodules each. Each submodule can obtain several different topologies, where the most common is the half – bridge topology due to the low number of switches resulting in lower power losses and control complexity compared to other submodule topologies. However, the half – bridge does not provide DC – fault handling capabilities, which is a major challenge with HVDC – MMC. This is possible with a full – bridge topology, but since it increase the number of switches, it increase losses and control complexity [8]. Other topologies, like clamp – double circuit, three – level converter circuit or five – level cross – connected circuit are possible, and are reviewed in [8]. Further in this thesis, the half – bridge submodule topology is considered.

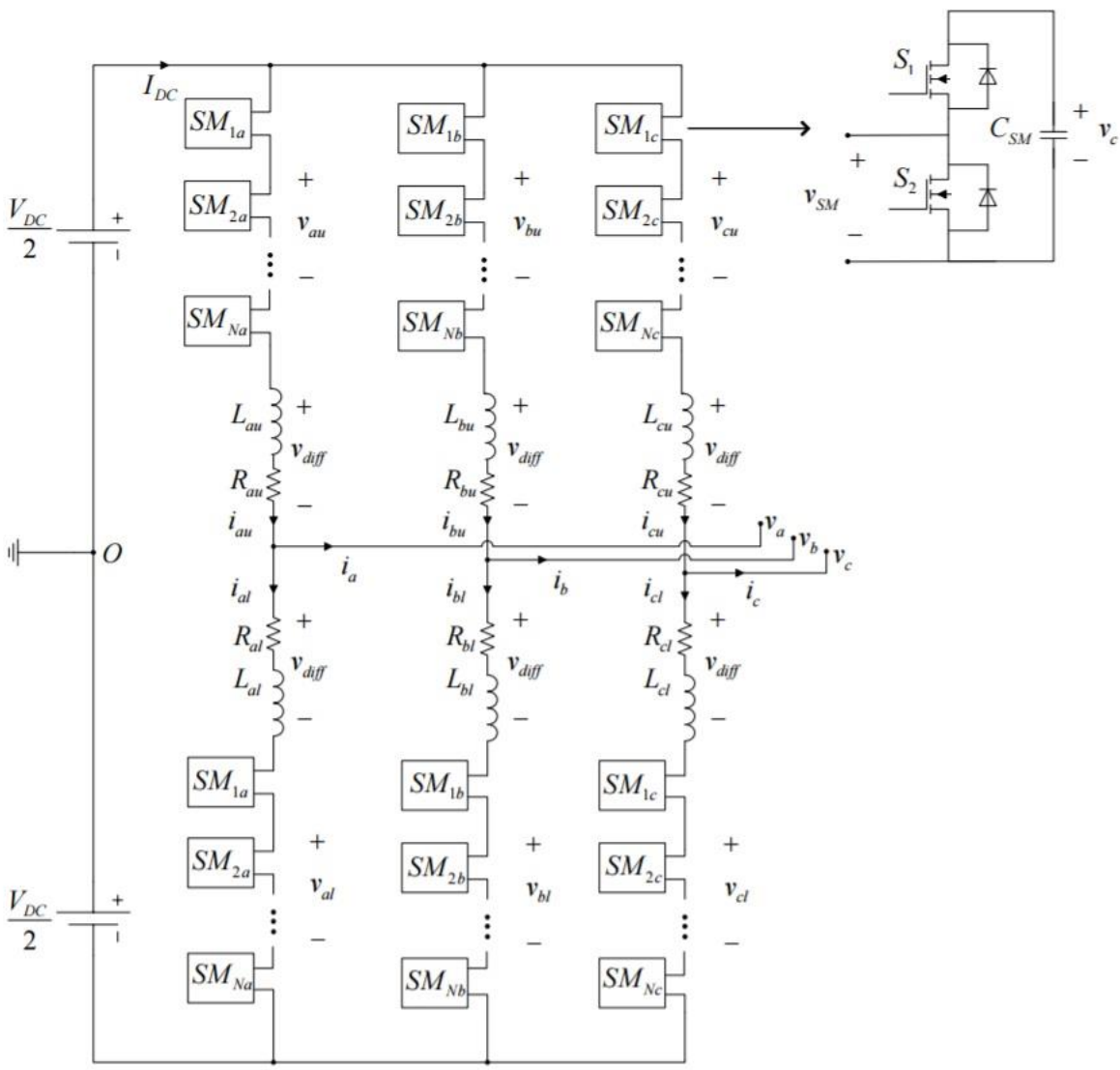


Figure 1- Electrical schematic of a three - phase MMC.

Each submodule capacitor C_{sm} is loaded with the voltage $v_c(t)$ which is influenced by the current flowing through that capacitor (upper arm current for the upper arm submodules and lower arm current for the lower arm submodules). With the half – bridge topology considered here, each submodule can be toggled between two states which is summarized in Table 1.

$v_{sm}(t)$	S_1	S_2
$v_c(t)$	ON	OFF
0	OFF	ON

Table 1 - Submodule switching states.

When a submodule is inserted (S_1 ON) the upper or lower arm current flows through the capacitor, influencing its voltage according the current flowing through the capacitor. When a submodule is bypassed (S_1 OFF), the current flows through S_2 without influencing the capacitor, hence the capacitor voltage remains constant. The N series connected submodule in each arm can be used as a discrete level voltage source with the voltage step of $v_{sm}(t)$.

2.1. Mathematical Model

The following paragraph gives a description of the mathematical equations governing the dynamic behavior of the MMC, based on derivations in [9]. This is done to show the mechanisms behind the submodule capacitor voltage ripples and how the circulating current is generated. The subscripts describing the phase (*a, b, c*) are dropped and the AC – side phase current takes the subscript *s*.

By choosing at each instant the appropriate number of submodules to insert, the inserted voltages can be varied as desired between zero and the sum capacitor voltage of the arm

$$0 \le v_{u,l} \le v_{u,l}^{\Sigma} \tag{1}$$

Where $v_{u,l} = 0$ when all submodules are bypassed and $v_{u,l} = v_{u,l}^{\Sigma}$ when all submodules are inserted. From Figure 1, we have the AC – side current

$$i_s = i_u - i_l \tag{2}$$

The differential current i_{diff} of the converter is defined as the current flowing through each phase. This current comprises of a DC – current component i_0, which is the DC side current I_{DC} divided equally (ideally) amongst each phase, and a circulating current component i_{circ}, a current component circulating within each phase of the converter without affecting the output current i_s

$$i_{diff} = i_0 + i_{circ} \tag{3}$$

The mean values of the arm currents must add up to the DC – bus current for the DC – bus voltage V_{DC} to remain constant

$$\sum_{k=1}^{x} \overline{i_{u,l}^{k}} = i_0 \tag{4}$$

Which implies that unless $i_0 = 0$, there will be a DC – component in the arm currents. Under balanced AC – side conditions, i.e. when the phase voltages and currents have identical, but time shifted waveforms, this DC – component is given by

$$\overline{i_u} = \overline{i_l} = \overline{i_{diff}} = \frac{i_0}{x} \tag{5}$$

To keep the RMS current, and hence power losses and component strain, to a minimum, it is desired that the differential current i_{diff} is pure DC, i.e. that the circulating current i_{circ} is zero. The upper (subscript *u*) and the lower (subscript *l*) currents are defined as

$$\begin{aligned} i_u &= i_{diff} + \frac{i_s}{2} \\ i_l &= i_{diff} - \frac{i_s}{2} \end{aligned} \tag{6}$$

The dynamic relations of the voltages of the upper and lower arms of one phase can be expressed by applying Kirchhoff's voltage law as

$$\begin{aligned} \frac{V_{DC}}{2} - v_u - Ri_u - L\frac{di_u}{dt} &= v_a \\ -\frac{V_{DC}}{2} + v_l + Ri_l + L\frac{di_l}{dt} &= v_a \end{aligned} \tag{7}$$

This gives the dynamic relationship between the output phase current and the controllable voltages as

$$\frac{L}{2}\frac{di_s}{dt} = \underbrace{\frac{-v_u + v_l}{2}}_{v_s} - v_a - \frac{R}{2}i_s \tag{8}$$

and the dynamic relations between the circulating current and the controllable voltages as

$$L\frac{di_{diff}}{dt} = \frac{V_{DC}}{2} - \underbrace{\frac{v_u + v_l}{2}}_{v_c} - Ri_{diff} \tag{9}$$

The voltage v_s is seen to be driving the output current i_s, while the voltage v_c can be seen to be driving the circulating current. For i_{diff} to be pure DC (as desired), the converter must be controlled in such a way that

$$v_c = V_{DC}/2 - Ri_{diff} \approx V_{DC}/2 \tag{10}$$

This give $L\left(di_{diff}/dt\right) = 0$, hence no voltage over the arm inductor capable of inducing time – varying circulating current.

With the definitions of v_s and v_c in equation (8) and (9), the maximum output voltage is obtained by bypassing all submodules in the upper arm and inserting all submodules in the lower arm

$$\left.\begin{aligned} v_u &= 0 \\ v_l &= v_{cl}^{\Sigma} \end{aligned}\right\} \rightarrow v_s^{\max} = \frac{v_{cl}^{\Sigma}}{2} \tag{11}$$

The minimum output voltage obtained by the opposite operation, by inserting all submodules in the upper arm and bypassing all submodules in the lower arm

$$\left.\begin{array}{l} v_u = v_{cl}^{\Sigma} \\ v_l = 0 \end{array}\right\} \rightarrow v_s^{\min} = -\frac{v_{cu}^{\Sigma}}{2} \tag{12}$$

To further obtain a dynamic model of the MMC of relative simplicity, [8] use an averaging principle. The inserted submodules can be expressed as

$$v_{u,l} = \sum_{i=1}^{N} n_{u,l}^{i} v_{c,u,l}^{i} \approx \sum_{i=1}^{N} n_{u,l}^{i} \frac{v_{c,u,l}^{\Sigma}}{N} = \frac{v_{c,u,l}^{\Sigma}}{N} \sum_{i=1}^{N} n_{u,l}^{i} \tag{13}$$

Where $v_{c,u,l}^{\Sigma}$ is the common value of the sum – capacitor voltages, which should be controlled to

$$\overline{v_{c,u,l}^{\Sigma}} = V_{DC} \tag{14}$$

for (10) to be achieved, and $v_{c,u,l}^{\Sigma} / N$ is the approximated average value of the individual submodule capacitor voltages, which then ideally should be charged to

$$v_{c,u,l}^{i} = \frac{\overline{v_{c,u,l}^{\Sigma}}}{N} = \frac{V_{DC}}{N} \tag{15}$$

Mean values in (14) and (15) is used. Since, when inserting a submodule, each capacitor will be charged with the instantaneous current in the arm where it is placed, a ripple will appear in the capacitor voltage. The ripple of each submodule capacitor adds up, creating a sum – capacitor ripple of the arm, hence only the mean value in (14) and (15) can equal V_{DC} and V_{DC} / N .

The approximation of (13) depends on making the submodule balancing (described in *3.4. Submodule Capacitor Voltage Sorting*) accurate enough to allow individual voltage differences between the capacitor voltages to be neglected. $n_{u,l}^{i}$ is the submodule insertion indexes, which only can attain two discrete values: $n_{u,l}^{i} = 0$ implies that the ith submodule in the upper/lower arm is bypassed, whereas $n_{u,l}^{i} = 1$ means that the submodule is inserted. The upper and lower arm voltages (v_u and v_l) can then further be written by approximating a continuous value of the per – arm insertion indexes as

$$n_{u,l} = \frac{1}{N} \sum_{i=1}^{N} n_{u,l}^{i} \rightarrow v_{u,l} = n_{u,l} v_{c,u,l}^{\Sigma} \tag{16}$$

The insertion indexes can obtain $N+1$ discrete values: $0,\ 1/N,\ 2/N,\ \ldots,\ 1,$ where $n_{u,l} = 0$ corresponds to all submodule in either upper (u) or lower (l) arm are bypassed, and $n_{u,l} = 1$ corresponds to all inserted. It is further assumed that the number of submodules N is high enough to allow approximating the insertion indexes as continuous on $[0,1]$. This assumption together with the approximation in (13) forms the basis for the average model of the MMC. (8) and (9) can now be written as the average dynamic model for the currents

$$\frac{L}{2}\frac{di_s}{dt} = \underbrace{\frac{-n_u v_{c,u}^{\Sigma} + n_l v_{c,l}^{\Sigma}}{2}}_{v_s} - v_a - \frac{R}{2} i_s \tag{17}$$

$$L\frac{di_{diff}}{dt}=\frac{V_{DC}}{2}-\underbrace{\frac{n_u v_{c,u}^{\Sigma}+n_l v_{c,l}^{\Sigma}}{2}}_{v_c}-Ri_{diff} \tag{18}$$

The average model infers that the switching dynamics of the MMC is disregarded. The discrete switching will add harmonics content in the output voltage and in the currents of the converter, depending on the switching frequency and the specific modulation strategy.

The averaging principle can further be applied to the dynamics of the capacitor voltages. With the only approximation of regarding $n_{u,l}$ as continuous on $[0,1]$, the governing equations for each submodule capacitor can be written as

$$C_{sm}\frac{dv_{c,u,l}^{i}}{dt}=n_{u,l}^{i}i_{u,l} \tag{19}$$

Summing all the capacitor voltages yields

$$C_{sm}\underbrace{\sum_{i=1}^{N}\frac{dv_{c,u,l}^{i}}{dt}}_{dv_{c,u,l}^{\Sigma}/dt}=\sum_{i=1}^{N}n_{u,l}^{i}i_{u,l}=i_{u,l}\underbrace{\sum_{i=1}^{N}n_{u,l}^{i}}_{Nn_{u,l}} \tag{20}$$

Which can be simplified to

$$\frac{C_{sm}}{N}\frac{dv_{c,u,l}^{\Sigma}}{dt}=n_{u,l}i_{u,l} \tag{21}$$

Expressing the upper and lower arm currents with the phase output current and the differential current yields

$$\begin{aligned}\frac{C_{sm}}{N}\frac{dv_{c,u}^{\Sigma}}{dt}&=n_u\left(\frac{i_s}{2}+i_{diff}\right)\\ \frac{C_{sm}}{N}\frac{dv_{c,l}^{\Sigma}}{dt}&=n_l\left(-\frac{i_s}{2}+i_{diff}\right)\end{aligned} \tag{22}$$

Appropriate selection of the insertion indexes n_u and n_l is important to properly design control and modulation system of the MMC. Reference generation of v_s^* and v_c^* are generally set up by the output and circulating current controllers. Using the expressions for v_s in (17) and v_c in (18) and substituting them with their references, the insertion indexes can be written in terms of the two reference voltages as

$$\begin{aligned}n_u&=\frac{v_c^*-v_s^*}{v_{c,u}^{\Sigma}}\\ n_l&=\frac{v_c^*+v_s^*}{v_{c,l}^{\Sigma}}\end{aligned} \tag{23}$$

Further, the voltage references are given as

$$v_c^*=\frac{V_{DC}}{2}-Ri_{diff}\approx\frac{V_{DC}}{2},\qquad v_s^*=\hat{V}_s\cos(\omega_0 t),\qquad i_s=\hat{I}_s\cos\left(\omega_0 t-\varphi_0\right) \tag{24}$$

Since both the insertion index and the upper and lower arm currents are periodic with the fundamental frequency, the product of them (right hand side of (21)) produce ripples in the sum – capacitor voltages. This can be quantified by integrating (21) scaled by N / C_{sm}, hence the ripples become inversely proportional to C_{sm}.

2.2.1. Analytical Derivation of the Steady State Sum-Capacitor Voltage Ripple

Inserting the expressions of the insertion indexes from (24) into (22) yields the following expressions

$$\begin{aligned} \frac{C_{sm}}{N}\frac{dv_{c,u}^{\Sigma}}{dt} &= \frac{v_c^* - v_s^*}{v_{c,u}^{\Sigma}}\left(\frac{i_s}{2} + i_{diff}\right) \rightarrow v_{c,u}^{\Sigma}\frac{C_{sm}}{N}\frac{dv_{c,u}^{\Sigma}}{dt} = \left(v_c^* - v_s^*\right)\left(\frac{i_s}{2} + i_{diff}\right) \\ \frac{C_{sm}}{N}\frac{dv_{c,l}^{\Sigma}}{dt} &= \frac{v_c^* + v_s^*}{v_{c,l}^{\Sigma}}\left(-\frac{i_s}{2} + i_{diff}\right) \rightarrow v_{c,l}^{\Sigma}\frac{C_{sm}}{N}\frac{dv_{c,l}^{\Sigma}}{dt} = \left(v_c^* + v_s^*\right)\left(-\frac{i_s}{2} + i_{diff}\right) \end{aligned} \tag{25}$$

By noting that

$$\frac{1}{2}\frac{d(v_{c,l}^{\Sigma})^2}{dt} = v_{c,l}^{\Sigma}\frac{dv_{c,l}^{\Sigma}}{dt} \tag{26}$$

and

$$\frac{C_{sm}}{2N}(v_{c,l}^{\Sigma})^2 = W_{u,l} \tag{27}$$

(25) can be rearranged as

$$\begin{aligned} \frac{dW_u}{dt} &= (v_c^* - v_s^*)\left(\frac{i_s}{2} + i_{diff}\right) \\ \frac{dW_l}{dt} &= (v_c^* + v_s^*)\left(-\frac{i_s}{2} + i_{diff}\right) \end{aligned} \tag{28}$$

Introducing the per phase and imbalance energies as $W_{\Sigma} = W_u + W_l$ and $W_{\Delta} = W_u - W_l$, the following relations can be made

$$\begin{aligned} \frac{dW_{\Sigma}}{dt} &= 2v_c^* i_{diff} - v_s^* i_s \\ \frac{dW_{\Delta}}{dt} &= v_c^* i_s - 2v_s^* i_{diff} \end{aligned} \tag{29}$$

Given the ideal values in (24), (29) can be written as

$$\begin{aligned} \frac{dW_{\Sigma}}{dt} &= V_{DC} i_{diff} - \frac{\hat{V}_s \hat{I}_s}{2}\cos\varphi_0 - \frac{\hat{V}_s \hat{I}_s}{2}\cos(2\omega_0 t - \varphi_0) \\ \frac{dW_{\Delta}}{dt} &= \frac{V_{DC}\hat{I}_s}{2}\cos(\omega_0 t - \varphi_0) - 2\hat{V}_s i_{diff}\cos(\omega_0 t) \end{aligned} \tag{30}$$

The term $-\left(\hat{V}_s\hat{I}_s/2\right)\cos\varphi_0$ equals the mean active power input per phase, i.e. $-P/x$. In order for the mean value of W_Σ to be constant in steady state, the relation $V_{DC}i_{diff}-\left(\hat{V}_s\hat{I}_s/2\right)\cos\varphi_0$ must equal zero to avoid an increase or decrease in the mean arm energy. With the ideal constraint of the differential current consisting of a pure DC current equal to $i_{diff}=I_{DC}/x$, giving $i_{diff}=\left(P/xV_{DC}\right)$. The AC output power must equal the DC input power if the losses are neglected. The other terms of (30) are of zero mean. Integration of (30) yields

$$\begin{aligned} W_\Sigma &= W_{\Sigma 0}\underbrace{-\frac{\hat{V}_s\hat{I}_s}{4\omega_0}\sin(2\omega_0 t-\varphi_0)}_{\Delta W_\Sigma} \\ W_\Delta &= W_{\Delta 0}+\underbrace{\frac{V_{DC}\hat{I}_s}{2\omega_0}\sin(\omega_0 t-\varphi_0)-\frac{2\hat{V}_s i_{diff}}{\omega_0}\sin(\omega_0 t)}_{\Delta W_\Delta} \end{aligned} \tag{31}$$

Where the integration constants $W_{\Sigma 0}$ and $W_{\Delta 0}$ are the mean values. (31) shows the total leg energy ripple to be of twice the fundamental frequency, while the imbalance leg energy is of the fundamental frequency. As the average value of $\overline{v_{u,l}}=V_{DC}$, the stored energy per arm is normally

$$\frac{C_{sm}}{2N}(V_{DC})^2 \tag{32}$$

The total leg energy is twice this value and its balanced among the arms, giving

$$\begin{aligned} W_{\Sigma 0} &= \frac{C_{sm}V_{DC}{}^2}{N} \\ W_{\Delta 0} &= 0 \end{aligned} \tag{33}$$

The sum – capacitor voltages can further be obtained by using the sum and difference energy as

$$\begin{aligned} v_{c,u}^{\Sigma} &= \sqrt{\frac{2N}{C_{sm}}W_u} = \sqrt{\frac{N}{C_{sm}}\left(W_{\Sigma 0}+\Delta W_\Sigma+\Delta W_\Delta\right)} = \sqrt{V_{DC}^2+\frac{N}{C_{sm}}\left(\Delta W_\Sigma+\Delta W_\Delta\right)} \\ v_{c,l}^{\Sigma} &= \sqrt{\frac{2N}{C_{sm}}W_l} = \sqrt{\frac{N}{C_{sm}}\left(W_{\Sigma 0}+\Delta W_\Sigma-\Delta W_\Delta\right)} = \sqrt{V_{DC}^2+\frac{N}{C_{sm}}\left(\Delta W_\Sigma-\Delta W_\Delta\right)} \end{aligned} \tag{34}$$

Generally, the capacitor voltage ripples are much smaller than the mean value V_{DC}, hence the following approximation can be made

$$\begin{aligned} v_{c,u}^{\Sigma} &= V_{DC}\sqrt{1+\frac{N}{V_{DC}^2C_{sm}}\left(\Delta W_\Sigma+\Delta W_\Delta\right)} \approx V_{DC}+\underbrace{\frac{N}{2C_{sm}V_{DC}}\left(\Delta W_\Sigma+\Delta W_\Delta\right)}_{\Delta v_{c,u}^{\Sigma}} \\ v_{c,l}^{\Sigma} &= V_{DC}\sqrt{1+\frac{N}{V_{DC}^2C_{sm}}\left(\Delta W_\Sigma-\Delta W_\Delta\right)} \approx V_{DC}+\underbrace{\frac{N}{2C_{sm}V_{DC}}\left(\Delta W_\Sigma-\Delta W_\Delta\right)}_{\Delta v_{c,l}^{\Sigma}} \end{aligned} \tag{35}$$

From (35), the following observation can be made

1.1. The amplitude of the sum – capacitor voltage ripples are inversely proportional to C_{sm}.

1.2. The ripples consist of two components in both arms

i) One component of the fundamental frequency ω_0 (resulting from W_Δ, eq. (31))

ii) One component of twice the fundamental frequency $2\omega_0$ (resulting from W_Σ, eq. (31))

1.3. The twice – the fundamental frequency components of the upper and lower arms are *in phase*, whereas the fundamental – frequency components are 180° phase shifted.

By adding and subtracting the upper and lower sum – capacitor voltages of (35), we obtain

$$\begin{aligned} v_c^\Sigma &= v_{c,u}^\Sigma + v_{c,l}^\Sigma \approx 2V_{DC} + \frac{N}{C_{sm}V_{DC}}\Delta W_\Sigma \approx 2V_{DC} - \frac{N}{C_{sm}V_{DC}}\frac{\hat{V}_s\hat{I}_s}{4\omega_0}\sin(2\omega_0 t - \varphi_0) \\ v_c^\Delta &= v_{c,u}^\Sigma - v_{c,l}^\Sigma \approx \frac{N}{C_{sm}V_{DC}}\Delta W_\Delta \approx \frac{N}{C_{sm}V_{DC}}\left(\frac{V_{DC}\hat{I}_s}{2\omega_0}\sin(\omega_0 t - \varphi_0) - \frac{2\hat{V}_s i_{diff}}{\omega_0}\sin(\omega_0 t)\right) \end{aligned} \tag{36}$$

(36) can be used to accurately predict the leg voltages. Further, from (36) and (31) the specific arm voltage ripples can be written as

$$\begin{aligned} \Delta v_{c,u}^\Sigma &= \frac{N}{2C_{sm}V_{DC}}\left(-\frac{\hat{V}_s\hat{I}_s}{4\omega_0}\sin(2\omega_0 t - \varphi_0) + \frac{V_{DC}\hat{I}_s}{2\omega_0}\sin(\omega_0 t - \varphi_0) - \frac{2\hat{V}_s i_{diff}}{\omega_0}\sin(\omega_0 t)\right) \\ \Delta v_{c,l}^\Sigma &= \frac{N}{2C_{sm}V_{DC}}\left(-\frac{\hat{V}_s\hat{I}_s}{4\omega_0}\sin(2\omega_0 t - \varphi_0) - \frac{V_{DC}\hat{I}_s}{2\omega_0}\sin(\omega_0 t - \varphi_0) + \frac{2\hat{V}_s i_{diff}}{\omega_0}\sin(\omega_0 t)\right) \end{aligned} \tag{37}$$

Summing the expressions for the upper and lower capacitor voltage ripple yields

$$\Delta v_c^\Sigma = \Delta v_{c,u}^\Sigma + \Delta v_{c,l}^\Sigma = -\frac{NV_s\hat{I}_s}{4\omega_0 C_{sm}V_{DC}}\sin\left(2\omega_0 - \varphi_0\right) \tag{38}$$

The 180^0 out – of – phase fundamental frequency voltage ripple components cancel each other out, whereas the 180° in – phase twice – the – fundamental frequency voltage ripple components adds up, resulting in a negative sequence, double fundamental frequency voltage component.

The insertion indexes $n_{u,l}$ will not be purely sinusoidal and continuous on $[0,1]$ as assumed in the previous derivations. Dependent on the specific modulation strategy, the insertion indexes will generate harmonic content in the output voltage and currents of the converter due to the discrete switching.

2.2. The Circulating Current

The differential current is responsible for transferring energy into and out of the arms. Looking at Figure 1, the voltage balance within one phase can be written as

$$n_u v_{c,u}^{\Sigma} + n_l v_{c,l}^{\Sigma} + L\left(\frac{di_u}{dt} + \frac{di_l}{dt}\right) + R\left(i_u + i_l\right) = V_{DC} \tag{39}$$

By inserting (6) in (39), the following expression for the differential current is obtained

$$\begin{gathered} n_u v_{c,u}^{\Sigma} + n_l v_{c,l}^{\Sigma} + L\left(\frac{d\left(i_{diff} + i_s/2\right)}{dt} + \frac{d\left(i_{diff} - i_s/2\right)}{dt}\right) + R\left(\left(i_{diff} + i_s/2\right) + \left(i_{diff} - i_s/2\right)\right) = V_{DC} \\ n_u v_{c,u}^{\Sigma} + n_l v_{c,l}^{\Sigma} + 2L\frac{di_{diff}}{dt} + 2Ri_{diff} = V_{DC} \end{gathered} \tag{40}$$

With the differential current consisting of a DC – component and an AC – component, the dynamic expression governing the circulating current can be expressed as

$$\begin{gathered} n_u v_{c,u}^{\Sigma} + n_l v_{c,l}^{\Sigma} + 2L\frac{d\left(i_0 + i_{circ}\right)}{dt} + 2R\left(i_0 + i_{circ}\right) = V_{DC} \\ \frac{di_0}{dt} = 0 \\ 2L\frac{di_{circ}}{dt} + 2Ri_{circ} = V_{DC} - \left(n_u v_{c,u}^{\Sigma} + n_l v_{c,l}^{\Sigma}\right) - \left(2Ri_0\right) \end{gathered} \tag{41}$$

With the voltage drop $2Ri_0$ generally being way smaller than the DC – link voltage, (41) can be approximated to

$$2L\frac{di_{circ}}{dt} + 2Ri_{circ} = V_{DC} - \left(n_u v_{c,u}^{\Sigma} + n_l v_{c,l}^{\Sigma}\right) = v_{diff} \tag{42}$$

With the Laplace – transform, (42) can be transformed from the time – domain to the frequency domain, yielding simpler system analysis.

$$2L\frac{di_{circ}}{dt} + 2Ri_{circ} = V_{DC} - \left(n_u v_{c,u}^{\Sigma} + n_l v_{c,l}^{\Sigma}\right) \xrightarrow{\mathcal{L}} 2Lsi_{circ} + 2Ri_{circ} = V_{DC} - \underbrace{\left(n_u v_{c,u}^{\Sigma} + n_l v_{c,l}^{\Sigma}\right)}_{2v_c} \tag{43}$$

With the impedance plant of circulating current written as

$$\begin{gathered} G_p(s) = \frac{1}{2R + 2Ls} = \frac{1}{2R}\left(\frac{1}{1 + T_p s}\right) \\ T_p = \frac{L}{R} \end{gathered} \tag{44}$$

In the upper and lower arms respectively, the ripples consist of components of even multiples of the fundamental frequency *in phase* and components of odd multiples of the fundamental frequency *180 degrees out of phase.* Adding the upper and lower arm voltages results in cancellation of the odd frequency multiples and addition of the even. As seen from (42), the voltage inducing the circulating current arise from the difference between the DC – link voltage and sum of the number of instantaneous inserted submodules in the upper and lower arm $V_{DC} - \left(n_u v_{c,u}^{\Sigma} + n_l v_{c,l}^{\Sigma} \right) = v_{diff}$. Assuming the switching frequency to be infinite (i.e. insertion indexes continuous on $[0,1]$) and the DC – link voltage constant, the circulating current will contain low order harmonics of even order, with the amplitude after the second order component rapidly decreasing [10]. With the approximation given in (36), the total leg voltage consists of its DC – component which equals twice the total DC – link voltage and a component oscillating at double fundamental frequency. This approximation holds if the ripples are much smaller than the mean capacitor voltage value V_{DC} .

Furthermore, the discrete nature of the insertion indexes will generate high order frequency components in v_{diff} and hence the circulating current. These higher order harmonics are strictly dependent on the specific modulation strategy used. For sine – triangular PWM modulation schemes, level shifted schemes generate harmonics in the circulating current dependent on whether they are classified as $N+1$ or $2N+1$ modulation scheme, referring to the number of voltage levels in the output voltage, see chapter *MMC Modulation Techniques* for further explanation. The circulating current contains both lower order harmonics due to voltage variations in the sum – capacitor voltage in the leg and higher order harmonics due to the specific modulation strategy used.

2.3. MMC Modulation Techniques

There are several modulation techniques available for multilevel converter topologies. The most common modulation schemes can be summarized in Figure 2.

The goal of modulating the MMC is to insert the correct number of submodules at each time instant to provide the voltage required for the operation of the converter. As the MMC is a switching voltage source converter, it cannot provide a continuous output, instead the output can only obtain discrete levels. The switching instances should be chosen such that the short – term average of the AC – side voltage coincides with the reference signal over a switching cycle. In addition to the desired low – frequency reference voltage, the switching process also generate high – order harmonics, which are undesirable. The frequency range and amplitude of these higher order harmonics are dependent on the specific modulation strategy. Furthermore, the specific modulation strategy will have different impacts on the capacitor voltage ripples and the circulating current [11, 12].

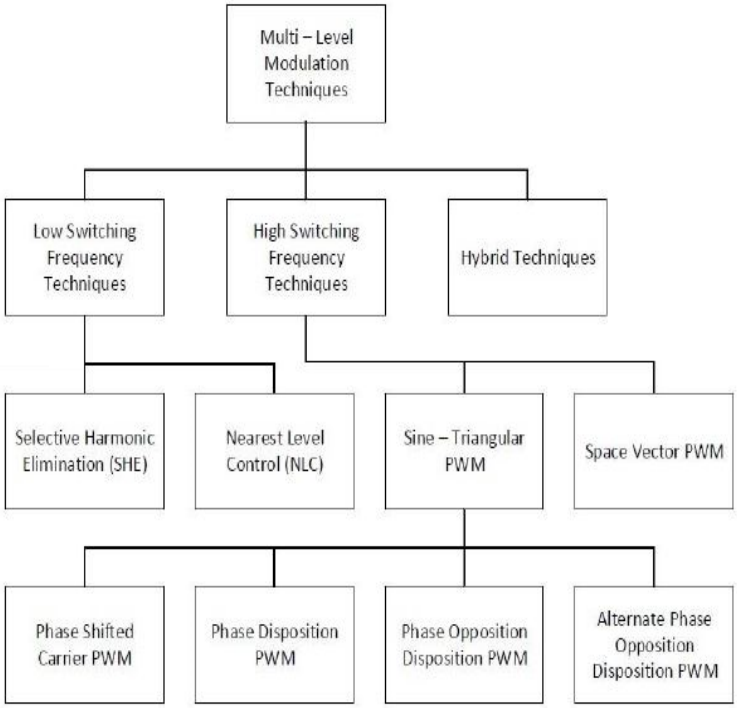


Figure 2 - Most common MMC modulation techniques.

In medium voltage – applications, such as medium voltage motor drives and MMC – STATCOM, the number of submodules *N* is usually relatively low. Sine – Triangular Pulse – Width – Modulation (PWM) is commonly applied to obtain good enough output voltage waveform with low THD. On the other hand, PWM methods occupy larger hardware resources, require more processing power and will produce higher switching losses than low – switching frequency methods such as SHE or NLC. For high voltage applications, where the number of submodules are high (often in the order of hundreds), a low switching frequency modulation method should be used to reduce the switching losses and reduce the implementation complexity, which increase with the number of submodules which needs to be controlled[13].

There are several ways of adapting carrier – based modulation techniques, where a triangular carrier is compared with the sinusoidal reference, to multilevel converter structures. The methods are based on splitting the multilevel waveform into a sum of a number of two – level PWM waveforms, each with its own carrier, while using a single reference voltage. Depending on how the carriers and reference are arranged, different methods have been developed, where there are mainly two categories which are commonly used: *Phase – Shifted Carriers* and *Level – Shifted Carriers [9]*.

In this thesis, a medium voltage and power MMC is considered. The resulting voltage level over each submodule switch will be kept at a relatively low level, as well as the current flowing through them. The generated power in the semiconductor switch due to switching activity is proportional to the switching frequency and the voltage over and current through the device at the switching instant. The low voltage and current level enables a higher switching frequency without reaching power loss and heating limits. Furthermore, the number of submodules are kept relatively low, such that the computational complexity will be kept low with the control system used. These factors enable the use of high – frequency PWM modulation.

2.3.1. Phase – Disposition PWM

This modulation technique is used in the simulations of this thesis. In this technique, the carriers are laid on top of each other, level shifted by V_{DC} / N, dividing the available direct voltage range between them. For a converter that can provide *N+1* equidistant levels, *N* carriers corresponding to the same number of two – level waveforms are required. All carriers have the same frequency, amplitude and phase and are compared to a common reference voltage

$$v_u^* = \frac{V_{DC}}{2}\left(1 - M\cos\left(\omega_0 t\right)\right) \tag{45}$$

$$v_l^* = \frac{V_{DC}}{2}\left(1 + M\cos\left(\omega_0 t\right)\right) \tag{46}$$

With the upper arm voltage reference generating switching pulses for the upper arm submodules, and the lower arm voltage reference generating switching pulses for the lower arm submodules. In Figure 3, the upper arm voltage reference (45) is compared with the triangular carriers and the upper arm switch pulse generation is shown. A similar figure will be the case for the lower arm, only with a 180° phase shifted reference waveform.

Two other commonly used level – shifted modulation techniques are

- *Phase Opposition Disposition (POD) – PWM:* Carriers above zero level are π rad phase shifted with respect to those below zero.
- *Alternate Phase Opposition Disposition – PWM:* Altering phase shifts of zero and π rad are used so that adjacent carriers will be in anti – phase.

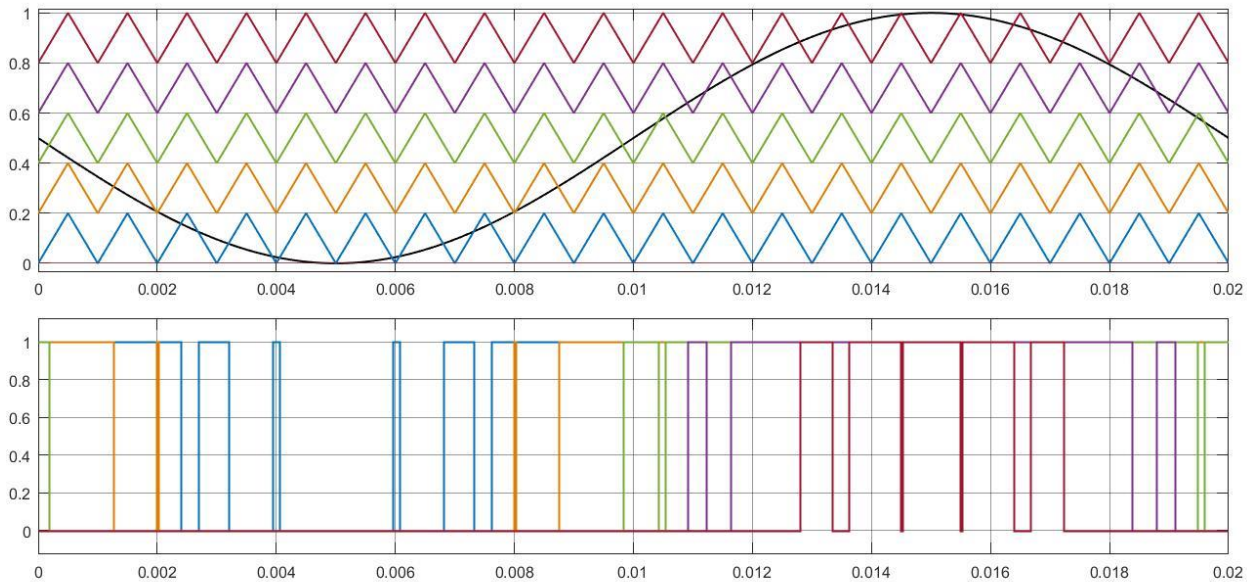


Figure 3 - Phase Disposition PWM for N=5.

2.3.1.1. Harmonic Generation with PD – PWM

The harmonic content of the pulse pattern can be analytically analyzed with the Fourier Series Expansion. One method treats the periodicity of the carrier and reference signals separately, where the total modulated signal is considered a function of the running phase angles $\omega_o t$ (corresponding to the reference signal) and $\omega_c t$ (corresponding to the carrier signal). The modulated signal will thus be a two – dimensional function $f(x,y)$, where

$$\begin{aligned} x &= \omega_0 t + \theta_0 \\ y &= \omega_c t + \theta_c \end{aligned} \tag{47}$$

The modulated signal is thus periodic in both *x* and *y* with a period of 2π rad. The signal can be expanded into a *double Fourier series expansion* written as

$$f(x,y) = \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} C_{mn} e^{j(mx+ny)} \tag{48}$$

With the Fourier coefficients given as the double integral

$$C_{mn} = \frac{1}{4\pi^2} \int_{-\pi}^{\pi} \int_{-\pi}^{\pi} f(x,y) e^{-j(mx+ny)} dxdy \tag{49}$$

With this analysis, analytical expression of the harmonic content of the modulated waveform can be obtained [9, 14]

[14] derives the harmonic contents of the phase and line – voltages for different multilevel PWM strategies, based on the double Fourier series mythology. The derived equations for the harmonic content of the phase voltage is given in *Appendix* for a five – level inverter modulated with PD – PWM. Even sideband harmonics *(2n)* will be present around odd carrier frequency groups *(2m-1)* and odd sideband harmonics *(2n-1)* present around even carrier frequency groups *(2m)*.

For the circulating current, harmonic expression for the arm voltages is given in [12, 15]. With the arm voltages inserted into (42), the harmonic expression for v_{diff} and hence i_{circ} derived, giving the following conclusions for APOD – and PD – PWM can be made

- *PD – PWM* is classified as $2N+1$. The number of inserted submodules in each phase is not always equal to *N* as is the case for APOD. Whenever $n_u + n_l = N \pm 1$, an additional intermediate voltage level $V_{DC}/2N$ will be generated in the output voltage. The voltage across the arm inductors will be increased during the instants when $N+1$ or $N-1$ submodules are inserted into the phase. Circulating current will be induced at these time instances, hence PD – PWM will generate circulating current containing high order harmonics. PD – PWM will, however, have better THD performance in output voltage due to higher number of voltage levels[12].

- *APOD – PWM* is classified as $N+1$. $n_u + n_l$ will always equal to *N*, resulting in the right – hand term of (42) equaling zero. This modulation technique results in complete harmonic cancelation between upper and lower arm voltage, hence the circulating current will be relatively free of high order harmonics caused by the PWM modulation. $N+1$ voltage levels can be generated in the output voltage[12].

3. Control Techniques

The inner dynamics, i.e. the circulating current and submodule capacitor voltages, should be controlled in order to operate the MMC in a satisfying manner. The focus will be on suppressing the circulating current with *repetitive* control and compare the performance[1] of this technique with other, common control techniques through simulations in Simulink. An algorithm for balancing the submodule capacitor voltages is also presented.

3.1. Discrete Modelling

A digital processing system is used to control the MMC. Since digital processing systems operate on discrete, quantized data, all control structures are simulated and implemented in the discrete *z* – domain in MATLAB Simulink. This allows for capturing effects of the sampling process imposed by the digital control structure and directly transform control algorithms into implementable code for both software and hardware[2]. The equivalent system representation can be viewed as in Figure 4.

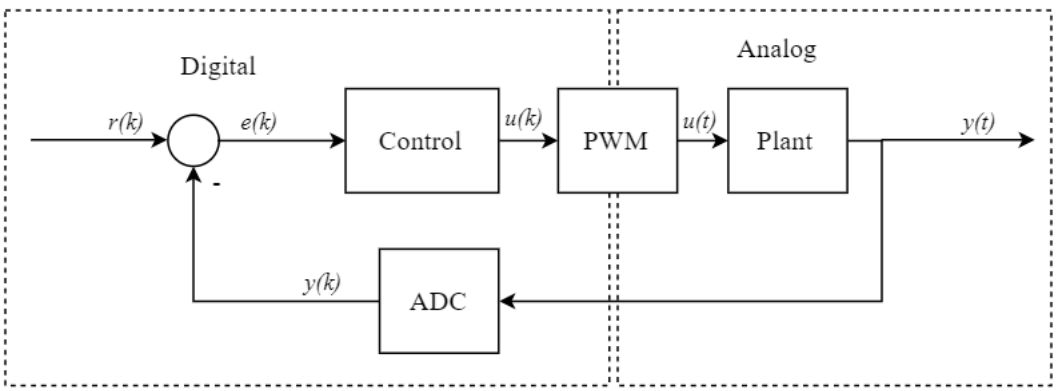


Figure 4 - System overview block diagram.

Tuning parameters for the nominal PI controller used in combination with the repetitive controller and the PR controller used in comparison with the repetitive controller, are analyzed in the continuous frequency (using the variable $s\ (=j\omega)$) domain. To capture the effects of sampling time on total system stability and performance, and allow for code generation and implementation, the controller structures are transformed into discrete frequency domain. The plant, which the digital processing system controls, is also discretized such that the plant can be included in the discrete time analysis of the controller structures.

[1] Performance meaning steady – state error, overshoot and settling time.
[2] Special care must be taken when algorithms are to be implemented on hardware such as FPGA: the Simulink model representation of the algorithm must be *HDL – compatible*.

3.1.1. Analog to Digital Conversion (ADC)

The continuous time signals from analog systems needs to be sampled and quantized before a binary computer system can use the values, a process performed by an analog to digital converter (ADC). At every time – instant *k*, the analog waveform is multiplied with the impulse function (Dirac delta function $\delta(t)$), forming an impulse train with the magnitude given by the value of the analog waveform at sampling instant *k*.

$$e(t)^* = \sum_{k=0}^{\infty} e(t)\delta(t-kT) \tag{50}$$

The value is then held over the sampling time period *T*, a process termed zero – order hold (ZOH). The ZOH is implemented physically as a Track and Hold (T/H)[3] circuit (also termed Sample and Hold (S/H)), as seen in Figure 5. The circuit is in *track* mode, where it tracks the input signal, when the switch is closed. When the switch opens, the last instantaneous value of the input is held over the capacitor C_{HOLD}. This can be seen as the T/H amplifier after the anti – aliasing low – pass filter in the functional block diagram of the AD7606 ADC of Figure 89 used for voltage measurement in the hardware design.

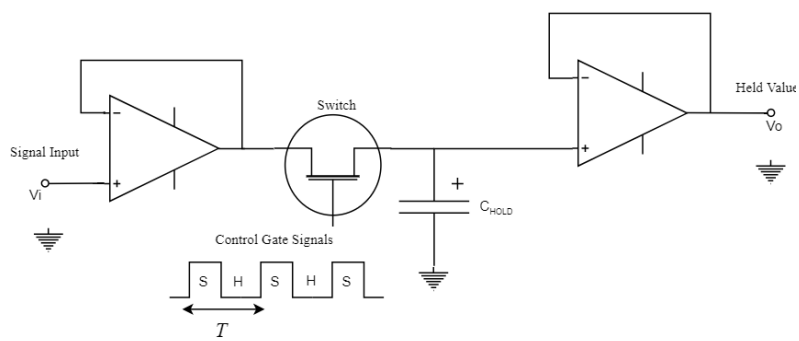


Figure 5 - Analog T/H circuit.

The held value is then quantized into a binary number with word – length dependent on the specific ADC.

To avoid anti – aliasing , the analog, measured input signal must be low – pass filtered through an anti – aliasing filter[4][16] to restrict the bandwidth of the signal to satisfy the sampling theorem over the frequency band of interest. This is implemented within the analog circuitry of the voltage measurement (see *Sheet 6* Eagle schematics in *Appendix*) and as an integrated filter in AD7606. For $\pm 5\ [\text{V}]$ input range, the filter bandwidth ($-3\ [\text{dB}]$ frequency) is typically at $15\ [\text{kHz}]$ (refer to [3] for details).

Furthermore, the sampled signal $e(kT)$ contains both the frequency components of the continuous time signal $e(t)$ and high – frequency complementary components resulting from the sampling process. In order to accurately reconstruct the continuous time – signal from the sampled data, these high – frequency components must also be removed, as they may adversely affect the system characteristics [17].

[3] The hold capacitor takes time to charge/discharge. This is the *track* or *sample* time which is needed to obtain the correct input value.

[4] Nyquist sampling theorem states that a band – limited, analog signal that has been sampled, can perfectly be reconstructed from an infinite sequence of samples if the sampling frequency exceeds the double of the highest frequency in the original signal. Thus, an anti – aliasing filter is used to satisfy this theorem by limiting the signal bandwidth for the given sampling frequency.

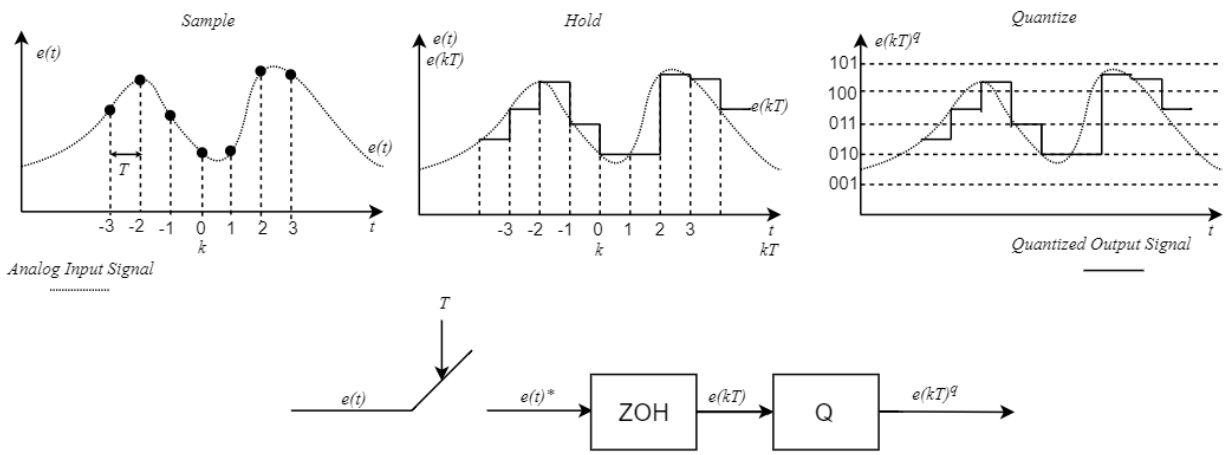


Figure 6 - Analog to quantized, discrete binary signal conversion process.

The quantization leads to error in the digital output dependent on the bit – resolution of the ADC. With samples coded with 12 bits or more (12 – bit resolution), allows one to consider the quantization effects as neglectable [18]. With the 16 – bit AD7606 ADC, the quantization error can thus be neglected.

3.1.2. Digital to Analog Conversion (DAC)

The pulse – width modulation can be viewed as the digital to analog (DAC) conversion process. Whether the PWM is implemented as hardware logic on FPGA[5] or in software on processor, the high/low output gate signals resulting from the PWM must be held at its required value during the required switch on – time. For PWM implemented on FPGA, this is accomplished with D – latch logic synchronized with the clock of the FPGA, which can be viewed as ZOH.

3.1.3. Continuous to Discrete Conversion

To represent continuous systems as discrete, a conversion process must take place. Several methods may be used to achieve discrete system representation, that is converting continuous – time signal to discrete – time signal. Methods supported by MATLAB built in function *c2d* include ZOH, First – Order Hold (FOH), Impulse – Invariant Mapping, Tustin Approximation and Zero – Pole Matching Equivalent. ZOH and Tustin approximation will further be used in this thesis.

[5] This has advantages in applications like MMC which contains high number of power switches. Complex modulation logic resulting from multilevel modulation techniques is required to be parallelly handled to obtain precise and fast handling of gate signals.

3.1.3.1. Zero – Order Hold

ZOH converts the input impulse sample train to a rectangular wave, equal to the sample time difference of two unit – step functions [17]

$$g_{ZOH}(t) = u(t) - u(t-T) \tag{51}$$

The transfer function representing the ZOH can be obtained by taking the Laplace transform of (51), yielding

$$G_{ZOH}(s) = \frac{1-e^{-sT}}{s} \tag{52}$$

The equivalent discrete transfer functions of the continuous – time transfer functions are represented in the z – domain. With the substitution

$$\begin{aligned} z &\triangleq e^{sT} \\ z^{-1} &\triangleq e^{-sT} \end{aligned} \tag{53}$$

The z – transform of the continuous transfer function is written as

$$Z\left(G(s)\right) = \sum_{k=0}^{\infty} e(kT) z^{-k} \tag{54}$$

By utilizing the zero – order – hold method, the relationship between the continuous and discrete time domains becomes

$$G(z) = (1-z^{-1}) Z\left(\frac{G(s)}{s}\right) \tag{55}$$

Using ZOH as the discretization method yields the use of the relationship of (55).

3.1.3.2. Tustin – transform

The Tustin – transform (also referred to as Bilinear transform) gives the best frequency domain match between continuous and discretized systems. If the system to be discretized also contains important dynamics at a specific frequency which the discretization must preserve, as resonant filters do, the prewarped Tustin method gives exact matching at the prewarped frequency. The Tustin transform performs the following relationship between discrete domain and frequency domain

$$s \rightarrow \frac{2}{T}\frac{z-1}{z+1} \tag{56}$$

and for prewarped Tustin yields

$$s \rightarrow \frac{h\omega}{\tan\left(\frac{h\omega T}{2}\right)}\frac{z-1}{z+1} \tag{57}$$

T is the numerical integration time step of the trapezoidal rule, which is used in the Tustin – transform derivation.

Both ZOH and Tustin approximation introduce inaccuracies from the conversion process due to the sampling time. However, when the sampling period *T* is set small enough, which it is assumed done in this thesis, the error between continuous and discrete domains can be neglected.

Tustin – transform is used to discretize the PR controller, while ZOH is used for all other discretization methods in this thesis. The design and analysis of the repetitive control is done directly in the discrete z domain, hence no transformation method is used[6].

3.1.4. Selection of sampling time

The modulator is implemented on a digital processor, and regular sampling is thus utilized. Regular sampled schemes are further classified into symmetrical and asymmetrical PWM, based on the symmetry of the generated semiconductor switching signals. The choice of modulator is dependent on the converter topology, maximum allowed switching losses, harmonic performance due to power switching transition and execution time of control algorithms. Choice of modulation method, switching frequency, control algorithm, software/hardware implementation and control platform is a compromise between performance and cost.

The system used in this thesis will utilize *Asymmetrical PWM With Sampling at the Start and in the Middle of Each Carrier Period (a-PWM- double)*. The modulation function (reference voltage) is updated twice per carrier signal period, once at the beginning and once in the middle of the carrier period. For a – PWM – double, the carrier signal period time T_c is thus doubled in relation to the sampling time period T_s. This leads to an overall dead – time or delay T_d introduced by the sampling, calculation and PWM update routines of [19]

$$T_d = \frac{T_c}{2} = T_s \tag{58}$$

With a carrier frequency of f_c the sampling time becomes $T_s = 1/(2f_c)$. The calculation time available for processing the control algorithm and updating the modulation function will then be $T_{calc} = 1/(2f_c)$ for *a – PWM – double.* A further consideration when choosing the sampling time is that discretized algorithms can only handle frequency components below the Nyquist frequency $\omega = \omega_s / 2 = \pi / T_{s.}$

The selection of sampling time should be viewed in the light of the complete system. Performance requirements with regards to switching frequency must be viewed against the required control algorithm execution time. It is a design compromise between high switching frequency yielding low allowable calculation time and sophisticated and complex control algorithm which may require long execution time.

[6] That is for the internal model and the accompanying filter *Q(z).* See *Internal Model Principle*.

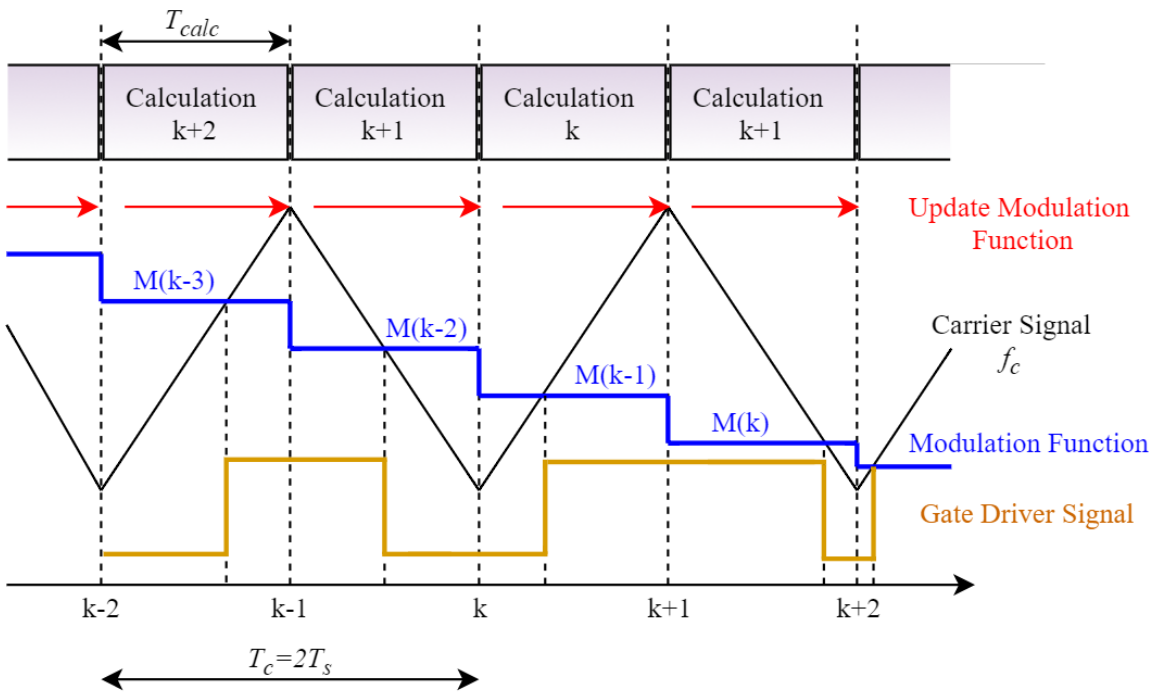


Figure 7 - a – PWM – double.

3.2. System Delays

Delays in the control system may cause instability if not accounted for, and should be incorporated into control paths when analyzing and designing the control system. Delays may be caused by factors like signal transmission delays, propagation delays in IC, sensing delays etc. For the system used in this thesis, the following paragraphs describes factors which may cause delay and how they may be accounted for in analysis. The delays are based on the hardware platform design described in *Hardware Design and Laboratory Set Up.*

1. Feedback time delay $T_{A/D}$

   i) Transducer delay. The current measurement transducer used for this hardware platform design is the LEM LA – 55P. The reaction time is $t_{ra} < 500$ [ns] and the step response time to reach 90% of the measured primary current is $t_a < 1$ [μs], giving the total worst – case time delay introduced by the current measurement $t_{\mathrm{TD}} = 1.5$ [μs].

   ii) ADC delay. The ADC used is the AD7606. The delay introduced by this IC depends on how it is operated:

   Digital Interface mode. The AD7606 provides serial, parallel or parallel byte interface possibilities, each resulting in different cycle delay. Reading data *during* or *after* the conversion process also yields different delays. Serial interfacing with data read after conversion yields the cycle delay $t_{\mathrm{SR}} = 9.4$ [μs].

Oversampling mode with digital MAF[7]. No oversampling yields a maximum conversion delay of $t_{0OS} = 4.15\ [\mu s]$, while the maximum oversampling ratio of $N = 64$ yields a conversion delay of $t_{64OS} = 315\ [\mu s]$. As seen, there are great differences in conversion time depending on the oversampling mode.

Group delay of input anti – aliasing filter is $t_G = 15\ [\mu s]$.

There is a compromise between high SNR performance, and low throughput time and frequency range of input signal, which should be analyzed with the total system in mind.

For further analysis, serial data read after conversion and no oversampling modes are assumed. The total delay from the AD7606 at these settings yields $t_{ADC} = t_{SR} + t_{0OS} + t_G = 28.55\ [\mu s]$. As these are selectable mode options which easily is changeable through bit command signals, different options should be explored while performing experiments to yield best performance.

iii) Signal conditioning and other analog circuitry. All IC units provides some delay. The isolation op – amp ACPL – 7900 yields a propagation delay (90% input to 90% output) of maximum $t_{ACPL} = 3.3\ [\mu s]$. The op – amp LM358D provides a slew – rate of SR=0.3 [V/μs], yielding neglectable distortion for the signals of this application.

These delays summed gives the approximated total input time delay $T_{A/D} = 28.55 + 3.3 = 31.85 \approx 30\ [\mu s]$.

2. Controller output time delay $T_{D/A}$

i) The digital to analog conversion is performed by the PWM, taking digital controller gate signals and providing analog gate signals to the power switches. As discussed in *Selection of sampling time* chapter, the worst-case delay imposed by the modulation – method (a – PWM – double) before the converter can act on a change in the reference voltage is modelled as half of the switching period.

ii) Gate driver IC used is the HCPL – 316J. It provides input pulse to high/low level output propagation delay of $t_{GD} = 0.5\ [\mu s]$. This is neglectable compared to the modelled PWM delay at the given PWM sampling frequency.

The digital to analog delay is thus approximated to be the worst – case delay posed by the PWM, $T_{D/A} = T_s$.

The delay in the ADuM7640 digital isolator is in the ns range and is neglected. Other IC delays and intrinsic circuit delay is assumed low enough to be neglected. For further details regarding the properties of the used ICs, refer to the data sheets

[7] Moving Average Filter, used to improve Signal – to – Noise (SNR) ratio. Dependent on the chosen oversampling ratio, the filter takes a number of samples *N* and averaging them to yield a better accuracy of the read input value. With an oversampling ratio of *N*=64, 64 samples are averages per conversion. High oversampling rates yields good accuracy, but require the ratio multiple higher conversion time. With no oversampling, the maximum throughput rate is 200 kSPS (kilo samples per second), while maximum oversampling rate of 64 yields 3.125 kSPS for all 8 channels. The SNR further depends on the number of quantization levels in the conversion process: more levels yields smaller quantization noise. The AD7606 use 16 – bit successive approximation conversion. The digital filter profiles caused by the MAF and further details regarding the AD7606, refer to data sheet.

When using open – loop transfer function for tuning gain parameters in systems with delays in the feedback path, the open – loop is broken after the feedback delay [20]. This is done to include the effect of the feedback transfer function while using the open – loop for tuning. The Laplace transform of a time – domain function delayed with the time unit T_d is

$$f\left(t-T_d\right)u\left(t-T_d\right)\xleftrightarrow{\mathcal{L}}e^{-sT_d}F\left(s\right) \tag{59}$$

Writing the D/A delay as the transfer function $G_d\left(s\right)=e^{-sT_{D/A}}$ and the A/D delay as $H_f\left(s\right)=e^{-sT_{D/A}}$, a general system representation is as in Figure 8.

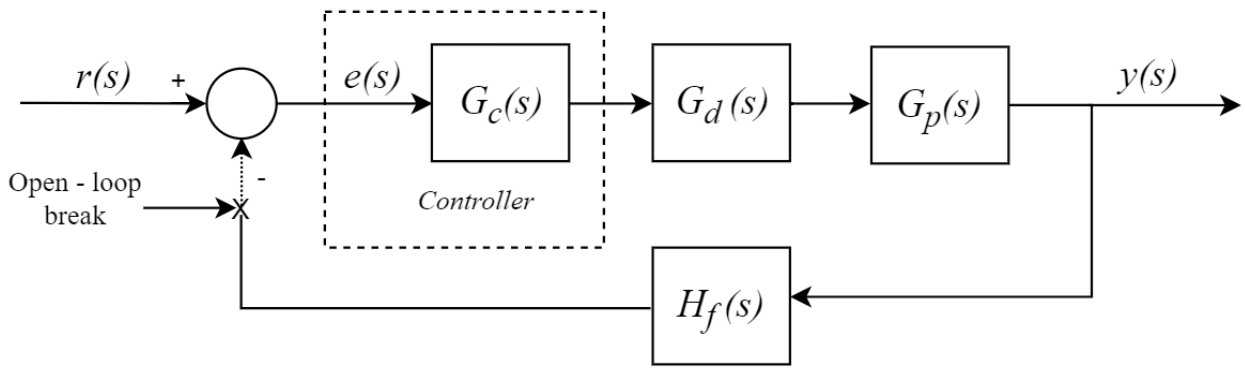


Figure 8 - Block diagram of a system including time delays.

The open – loop indicated by the break point in is written as

$$G_{ol}^{'}\left(s\right)=G_c\left(s\right)G_p\left(s\right)G_d\left(s\right)H_f\left(s\right)=G_c\left(s\right)G_p\left(s\right)e^{-s\left(T_{D/A}+T_{A/D}\right)}=G_c\left(s\right)G_p\left(s\right)e^{-sT_d} \tag{60}$$

With $T_d=T_{D/A}+T_{A/D}$ being the total system delay. By including the feedback delay in the open – loop as in (60) would give the equivalent closed – loop transfer function

$$G_{cl}^{'}\left(s\right)=\frac{G_{ol}^{'}\left(s\right)}{1+G_{ol}^{'}\left(s\right)}=\frac{G_c\left(s\right)G_p\left(s\right)e^{-sT_d}}{1+G_c\left(s\right)G_p\left(s\right)e^{-sT_d}} \tag{61}$$

Which would deviate from the *actual* closed – loop transfer function $G_{cl}\left(s\right)$

$$G_{cl}\left(s\right)=\frac{G_c\left(s\right)G_p\left(s\right)e^{-sT_{D/A}}}{1+G_c\left(s\right)G_p\left(s\right)e^{-sT_d}}=G_{cl}^{'}\left(s\right)e^{sT_{A/D}}=G_{cl}^{'}\left(s\right)\left(H_f\left(s\right)\right)^{-1} \tag{62}$$

Thus, the open – loop tuning with the transfer function in (60) assumes the closed – loop transfer function of (61), which yields some difference in phase due to the delay of the feedback path[8] compared to the actual closed – loop transfer function of (62).

[8] If there are transfer function with other characteristics than pure phase as e^{sT} present in the feedback, for example if an approximation as in (63) is used, differences are seen in the magnitude as well.

As the delay of (59) is irrational and cannot be expressed as a ratio of polynomials[9], including such delay yields a non – analytical expression when analyzing stability with standard methods (root – locus, Routh – Hurwitz etc.)[21]. The delay is therefore approximated with Padé – approximation, where first order often is sufficient [20]

$$e^{-sT_d} \approx \frac{1}{1+sT_d} \tag{63}$$

For discrete – time systems, a time series at sample number *k* represented as $x[k] = x[0] + x[1] + x[2]...$ could then be represented using Laplace transform and using the delay theorem of (59) as $X(s) = x[0] + x[1]e^{-sT_s} + x[2]e^{-2sT_s}...$ with the sample period T_s. However, for periodically sampled signals, an infinite number of insertions would be required. Thus, sampled system is more conveniently represented with the *z* – operator, defined as in (53) [22]. Discrete, sampled system can thus be equivalently described as in Figure 9.

With a simulation time step of T_{sim} for fixed – step solvers, the delays can be discretely implemented in the simulation model and accounted for in analysis and controller tuning by delaying the feedback path with $N_{A/D} = T_{A/D} / T_{sim}$ and forward path $N_{D/A} = T_{D/A} / T_{sim}$ number of samples[10], as in Figure 9. This requires the Simulink block to inherit the sample time of the fixed – step solver, i.e. setting -1 as sample time for the block. Alternatively, the blocks which cause the delay can obtain the sample time equivalent to the delay time and set the delay integer to 1 sample delay.

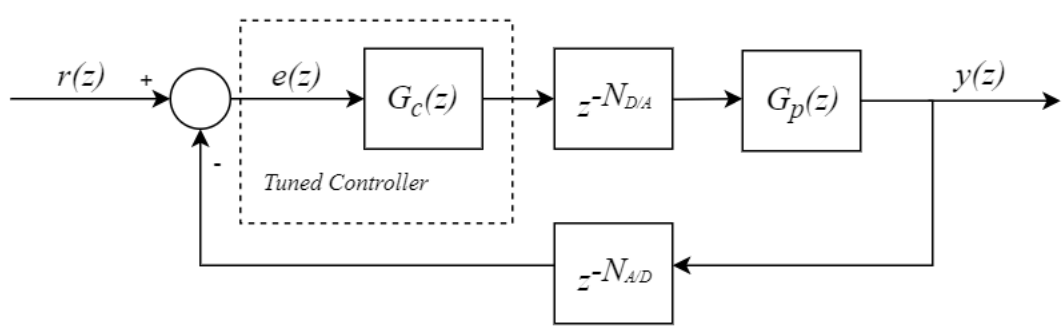


Figure 9 - Block diagram in discrete frequency domain with delays accounted for in the system paths.

Simulation results showing the effects of including these delays in simulation is shown in Figure 24 in *Design of the Nominal* Controller.

[9] Expressed as $g(s) = n(s)/d(s)$, where $n(s)$ and $d(s)$ are polynomials.

[10] Requires $T_{A/D}$ and $T_{D/A}$ to be an integer multiple of T_{sim}.

3.3. Voltage Control

Together with the output current (also termed outer dynamics control), there are three state variables per phase of the averaged dynamic model which needs to be controlled. These variables are the circulating current and the two sum – capacitor voltages (also termed inner dynamics control).

The goal of the voltage control is for the mean values of v_{cu}^{Σ} and v_{cl}^{Σ} to converge to V_{DC}. This control is based upon the proper selection of insertion indexes, as it is seen from (22) that there are these input variables which are available for manipulation. The specific class of voltage control is based on how the reference waveform is generated. The following describe the most commonly used voltage control techniques, based on descriptions in [8, 9].

3.3.1. Direct Voltage Control

This voltage control technique calculates the ideal insertion indexes based on (23), where the sum capacitor voltages v_{cu}^{Σ} and v_{cl}^{Σ} are substituted with their common mean value V_{DC}, such that

$$\begin{aligned} n_u &= \frac{v_c^* - v_s^*}{V_{DC}} \\ n_u &= \frac{v_c^* + v_s^*}{V_{DC}} \end{aligned} \tag{64}$$

This gives inherently asymptotic stable system and a low computational complexity. On the other hand, parasitic components appear in the output and internal voltages, which contributes to harmonics in the circulating current as well as output voltages. The circulating current may however be suppressed by circulating current controllers. This technique will be utilized further in the thesis.

3.3.2. Indirect Voltage Control

In this technique, the upper and lower arm insertion indexes are given by

$$\begin{aligned} n_u &= \frac{v_c^* - v_s^* - v_{reg}^{\Sigma} - v_{reg}^{circ}}{v_{c,u}^{\Sigma}} \\ n_l &= \frac{v_c^* + v_s^* - v_{reg}^{\Sigma} - v_{reg}^{circ}}{v_{c,l}^{\Sigma}} \end{aligned} \tag{65}$$

Where v_{reg}^{Σ} and v_{reg}^{circ} are used to control the total energy in one phase and balance the energy between the arms, respectively.

The Indirect Voltage Control can further be classified into

- *Closed – loop control:* v_{cu}^{Σ} and v_{cl}^{Σ} are calculated based on the actual *measured* values of the submodule capacitor voltages. v_{reg}^{Σ} and v_{reg}^{circ} are obtained from closed – loop control of the total energy stored in the arm capacitors and the energy balance between the arms, respectively. Both these arm – energy controllers are needed to obtain asymptotically stable system. The advantages of this technique lie in the control of the average submodule capacitor voltage, which allows for operation under low output voltage with high number of voltage levels, and the control of the energy imbalance between the upper and lower arms.

- *Open – loop control:* v_{cu}^{Σ} and v_{cl}^{Σ} are calculated based on *estimated* values of the submodule capacitor voltages. Furthermore, $v_{reg}^{\Sigma} = 0$ and v_{reg}^{circ} is estimated to eliminate the harmonics in the circulating current. As for *Direct Voltage Control*, inherent asymptotic stability is obtained and only small parasitic voltage components appear. Advantage of this technique lies in the absence of voltage sensors (such sensor will introduce delays, increase computational effort and increase the use of hardware resources). However, accurate estimation of the real parameters necessary to describe the dynamics of the system is always a big challenge.

3.3.3. Hybrid Voltage Control

In this technique, the insertion indexes are calculated as follows

$$\begin{aligned} n_u &= \frac{v_c^* - v_s^*}{V_{DC} + BPF\left(v_{cu}^{\Sigma}\right)} \\ n_l &= \frac{v_c^* + v_s^*}{V_{DC} + BPF\left(v_{cl}^{\Sigma}\right)} \end{aligned} \tag{66}$$

v_{cu}^{Σ} and v_{cl}^{Σ} in the denominator are band – pass filtered (*BPF)* such that their mean values vanish, but the ripple components remain. The desired mean value V_{DC} is added. The system becomes inherently asymptotically stable.

3.4. Submodule Capacitor Voltage Sorting

Submodule voltage sorting is needed as each submodule is inserted into the arm at different time instants. The arm current flowing through each submodule capacitor will be different, depending on which time instant the submodule is inserted, which can be understood from

$$i_{c,u,l}^{i} = n_{u,l}^{i} i_{u,l} \tag{67}$$

and (19). The individual submodule capacitor voltage depends on the current flowing through them, thus each submodule capacitor voltage will diverge to different voltage levels if not a proper voltage control is implemented. In the inserted submodule, the capacitor will charge if the current flowing through it is positive and discharge if it is negative. For bypassed submodules, the voltage will remain unchanged.

One of the most widely accepted and used methods for submodule voltage balancing is based on a sorting strategy [8]. It is based on measuring each individual submodule capacitor voltage and sorting the submodule voltages in either ascending or descending order depending on the respective direction of the arm current. The number of needed inserted submodules to produce the target waveform are calculated first, before the following situation may apply [9]

- *Positive arm current and a submodule should be inserted:* the bypassed submodule with the lowest voltage is inserted as this will increase (charge) the capacitor voltage.
- *Negative arm current and a submodule should be inserted:* the bypassed submodule with the highest voltage is inserted as this will decrease (discharge) the capacitor voltage.
- *Positive arm current and a submodule should be bypassed:* the inserted submodule with the highest voltage is bypassed as this will keep the submodule voltage to further increase.
- *Negative arm current and a submodule should be bypassed:* the inserted submodule with the lowest voltage is bypassed as this will keep the submodule voltage to further decrease.

The modulator's only task is to calculate the required number of submodules to be inserted at any given time step, and the sorting algorithm takes care of which specific submodule to insert to keep the submodule capacitor voltage as constant as possible. Although the sorting method guarantees capacitor voltage sorting under all operating conditions (provided that the frequency of the sorting is high enough), the method produce unnecessary switching transitions amongst the submodules [8]. Even if the number of required on – state submodules within two consecutive control periods are not changed, the submodule insertion/bypassing may still happen, resulting in increased switching frequency and further power losses. Methods for avoiding these problems are further summarized in [8].

Another problem this algorithm change is the inherent problem of the sudden arm current direction change. The arm current may change in a shorter time period than the time period control cycle of the algorithm (t_0 to t_1 in Figure 10). If the arm current is considered negative at the instant of sorting, a submodule with too high voltage will be inserted. Since the arm current may change direction in a shorter time period than the sorting time period, the current may become positive, even further charging the

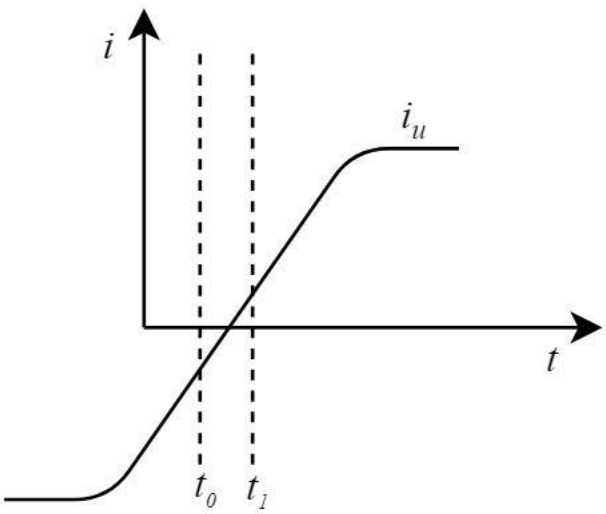


Figure 10 - Problem with submodule capacitor voltages sorting technique.

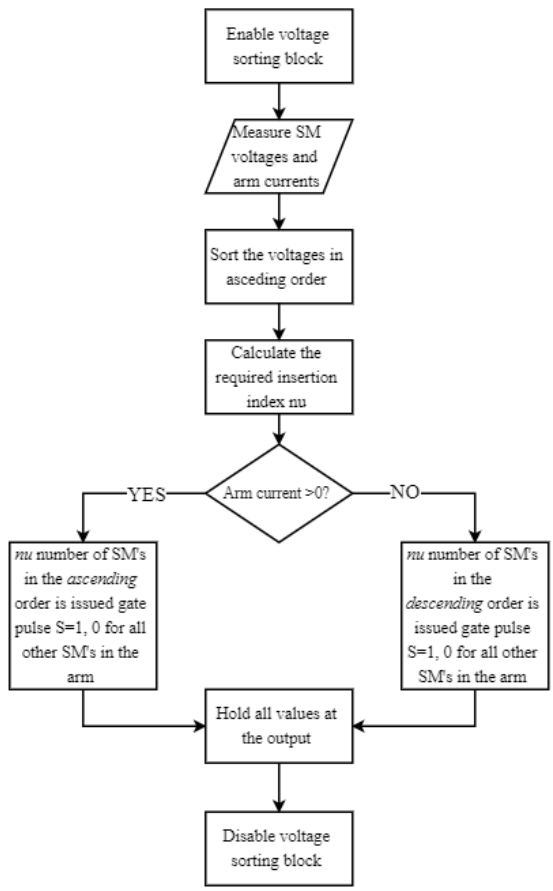


Figure 11 - Voltage sorting algorithm flow chart.

already above – limit inserted submodule capacitor voltage. Evidently, if the sorting is not updated with a low enough time period, it may lead to significant voltage peaks in the capacitor voltage ripples [11].

The algorithm is used in the Simulink simulations to control the submodule voltages. The algorithm is implemented with Simulink blocks is shown in the *Appendix*, Figure 87. Figure 12 shows the submodule capacitor voltages with and without voltage sorting enabled.

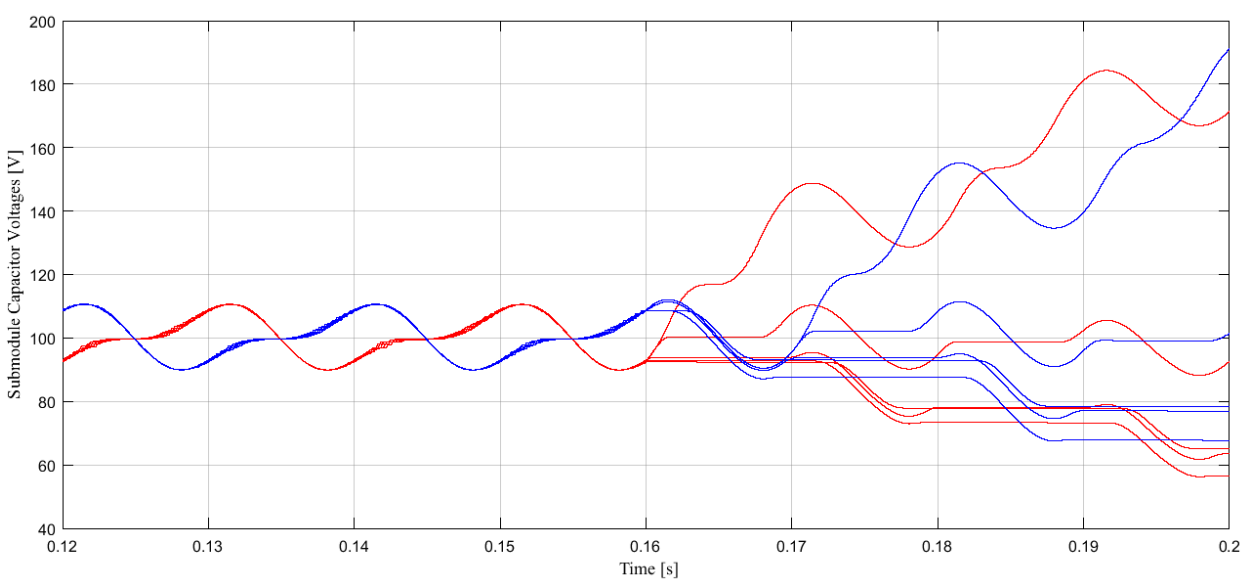


Figure 12 -Upper (red) and lower (blue) arm submodule capacitor voltages with circulating current control and voltage sorting ENABLED at t<0.16 (s). Voltage sorting DISABLED at t=0.16 (s).

3.5. Circulating Current Control for Direct Voltage Control

The circulating current control is linked to the voltage control, and should be adapted to the specific voltage control scheme utilized. As described in the Voltage Control chapter, the different voltage control strategies lead to different consequences for the differential current. With Direct Voltage Control, low even order harmonic components and high order harmonic components due to the PWM modulation is present in the disturbance, and give rise to these components in the circulating current. These components can however be suppressed by proper control strategies which will be discussed in the following. The disturbance contains lower order harmonics of even multiples of the fundamental frequency and higher order harmonics centered around the switching frequency. Of the lower order harmonics, the second order harmonic (at $2\omega_0$) is highest in amplitude, with the following harmonic components rapidly decreasing in amplitude. Switching frequency harmonics are dependent on the specific modulation strategy and will unavoidably contribute to high frequency noise in the circulating current. For high enough switching frequencies, this disturbance can accurately be modelled to be containing double fundamental frequency components and switching harmonics [9].

The disturbance signal should therefore be modelled containing sinusoidal frequency components at even multiples of the fundamental frequency, with the amplitudes rapidly decaying as the frequency increase. The high order switching harmonics will be outside the bandwidth of the controller and can only be influenced by the specific modulation strategy. (68) is the frequency domain disturbance signal

$$d(s) = a_2 \frac{(2\omega_0)^2}{s^2 + (2\omega_0)^2} + a_4 \frac{(4\omega_0)^2}{s^2 + (4\omega_0)^2} + a_6 \frac{(6\omega_0)^2}{s^2 + (6\omega_0)^2} + \ldots \qquad (68)$$
$$a_2 > a_4 > a_6 \ldots$$

A basic block diagram of the system to be controlled is shown in Figure 13.

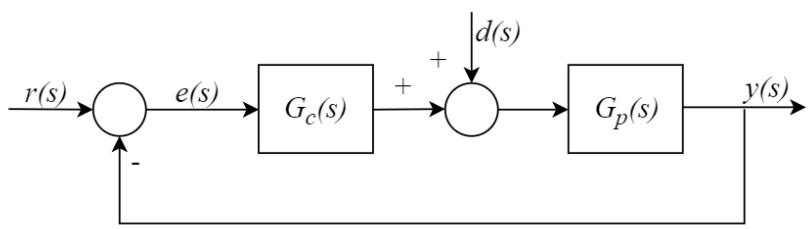


Figure 13 - Block diagram of system.

3.5.1. Proportional Integral (PI) Control

PI controller will only be able to provide zero steady – state error for signals with zero frequency, as the integrator will provide infinite gain at DC. Thus, for a PI controller to accurately track/suppress an oscillating signal without steady – state error, the signal which is to be tracked/suppressed must be transformed into a non – oscillating signal. This may be done by transforming the three – phase, oscillating *abc* – quantities into the synchronous frame in order to obtain DC, *direct – quadrature (dq)* – quantities. The *d* and *q* component can then be controlled with their individual PI controller. For current – regulated, balanced, three – phase converter systems, one PI controller for the *d* and one for the *q* – current component is needed, resulting in two PI controllers. For an unbalanced, three – phase systems, two PI controllers are needed for both the positive sequence and negative sequence currents, resulting in a total of four PI controllers [23].

These structures pose additional computational efforts and accuracy challenges. For *n* harmonics in the circulating current to be suppressed, *n* reference – frame transformations need to be performed, increasing computational efforts. Furthermore, current frequencies need to be estimated through phase – locked – loops (PLL) in order to accurately transform the quantities, introducing accuracy challenges.

These challenges of reference – frame transformations are avoided with control algorithms which directly can act in an effective manner within each phase, that is accurately track/suppress signals with frequency components. Their control algorithms are, however, more complex and require more resources than the PI controller algorithms. Methods for controlling sinusoidal signals is discussed in the following chapters.

3.5.2. Proportional Resonant (PR) Control

To cope with sinusoidal tracking/rejection without *dq* – transformation, a Proportional Resonant (PR) controller can be utilized. For the resonance part of the controller, a resonant filter is used. With the dynamics of the circulating current described with (43), the following control law can be obtained for the circulating current with PR – control

$$2v_c^* = V_{DC} - 2Ri_{diff}^* - \left(k_p + G_h(s)\right)\left(i_{diff} - i_{diff}^*\right) \tag{69}$$

The current i_{diff}, as in (40), is used rather than i_{circ}, as in (43). The difference is only in the presence of the DC – component i_0 in i_{diff}, which will be filtered with a DC – block – filter when implemented in Simulink. Where v_c equals its reference v_c^* delayed with T_d, plus the disturbance voltage *d*. This voltage disturbance component d represent the switching harmonics and, for direct voltage control, components of even – multiples of ω_0 A feedforward term $V_{DC} - 2Ri_{diff}^*$ is added to compensate for the resistive voltage drop in the arms and the DC – link voltage. The block diagram of the control structure is shown in Figure 14.

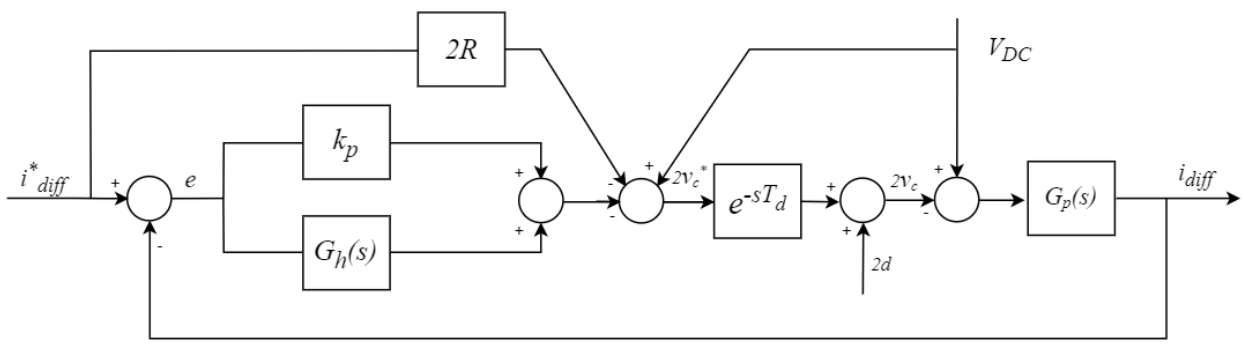


Figure 14 - Block diagram of PR control block control structure.

In practical implementation, R would need to be estimated, where the accuracy of the estimation might be a challenge. The error signal $e = i_{diff}^* - i_{diff}$ is feed through the PR controller consisting of a proportional gain k_{pr} and a resonant filter $G_h(s)$. k_{pr} effectively increase the resistance form R to $R + k_{pr}$ and can be looked upon as virtual or active resistance. The PR controller should be designed for good disturbance rejection within the system stability area. The filter transfer function is [9]

$$G_h(s) = \frac{k_h \cdot \left(s\cos\phi_h - h\omega_0 \sin\phi_h\right)}{s^2 + \alpha_h s + \left(h\omega_0\right)^2} \tag{70}$$

Where k_h is the gain, α_h is the bandwidth of the passband and ϕ_h is the phase compensation angle. The filter acts as a bandpass filter (BPF) about the angular frequency $h\omega_0$. With $s = jh\omega_0$, we get

$$G_h\left(jh\omega_0\right) = \frac{k_h \cdot \left(\cos\phi_h + j\sin\phi_h\right)}{\alpha_h} = \frac{k_h \cdot e^{j\phi_h}}{\alpha_h} \tag{71}$$

With a pure BPF, $e^{j\phi_h}$ can be used to compensate for any forward path time delay e^{-sT_d} by setting $\phi_h = \omega_0 T_d$. Setting $k_h = \alpha_h$ and $\phi_h = 0$ gives $G_h(j\omega_1) = 1$, a frequency component at $h\omega_0$ is admitted without phase shift or amplitude. By setting $\alpha_h = 0$ and $\phi_h = 0$ yields infinite gain at $h\omega_0$, resulting in an *ideal* resonant controller, just like the integrator $(1/s)$ yields infinite gain at $s = 0$ (DC). The ideal resonant controller, also termed the second order generalized integrator (SOGI) have the transfer function

$$G_{SOGI}(s) = \frac{k_h s}{s^2 + (h\omega_0)^2} \tag{72}$$

A PR controller with a certain bandwidth is often preferred for a better digital realization and noise rejection, especially when using fixed point representation [24]. By using the resonant filter summed with a proportional gain, the PR controller transfer function is obtained as

$$G_{pr}(s) = k_{pr} + \frac{k_h s}{s^2 + \alpha_h s + (h\omega_0)^2} \tag{73}$$

To ease the analysis, the PR transfer function is written as [24]

$$\begin{gathered} G_{pr}(s) = k_{pr}\left(1 + \frac{s}{T_h\left(s^2 + \alpha_h s + (h\omega_0)^2\right)}\right) \\ T_h = \frac{k_{pr}}{k_h} \end{gathered} \tag{74}$$

The goal of the tuning is to maximize the proportional gain k_{pr} and minimize the resonant integral (general integrator) time T_h (maximizing the resonant gain k_h), while taking into account the delay time T_d and still keeping a sufficient phase margin φ_{pm}. The maximum proportional gain is achieved at a highest possible crossover frequency α_{co} (frequency where unity gain of the open – loop frequency response occurs with the required phase margin). Minimizing the resonant integral time is then done accordingly. The following parameter tuning is based on [9, 24].

The open loop transfer function including the controller output delay T_d [11] is

$$G_{pr}(s) \cdot G_p(s) = \frac{k_{pr}}{2R}\left(1 + \frac{s}{T_h\left(s^2 + \alpha_h s + (h\omega_0)^2\right)}\right) \cdot \left(\frac{1}{1 + T_p s}\right) \cdot e^{-sT_d} \tag{75}$$

[11] Delay including both *digital to analog* delay in the forward path and *analog to digital* delay in the feedback path (see 3.2).

The phase angle of the forward path at the crossover frequency α_{co} is

$$\angle G_{pr}\left(j\alpha_{co}\right)G_p\left(j\alpha_{co}\right)=\angle\left(\frac{k_{pr}}{2R}\left(1+\frac{j\alpha_{co}}{T_h\left(\left(\left(h\omega_0\right)^2-\alpha_{co}^2\right)+j\alpha_{co}\alpha_h\right)}\right)\left(\frac{e^{-j\alpha_{co}T_d}}{1+j\alpha_{co}T_p}\right)\right)$$
$$\varphi_{ol}=\frac{\pi}{2}-\arctan\left(\frac{\alpha_{co}\alpha_h}{T_h\left(\left(h\omega_0\right)^2-\alpha_{co}^2\right)}\right)-\arctan\left(\alpha_{co}T_p\right)-\alpha_{co}T_d \tag{76}$$

For sufficiently large α_{co}, such that

$$\left|\frac{j\alpha_{co}}{T_h\left(\left(\left(h\omega_1\right)^2-\alpha_{co}^2\right)+j\alpha_{co}\alpha_h\right)}\right|\approx\left|\frac{1}{j\alpha_{co}T_h}\right|, \tag{77}$$

the phase can be approximated as

$$\varphi_{ol}=\angle\left(\frac{k_{pr}}{2R}\left(1+\frac{1}{j\alpha_{co}T_h}\right)\left(\frac{e^{-j\alpha_{co}T_d}}{1+j\alpha_{co}T_p}\right)\right)$$
$$\varphi_{ol}=-\frac{\pi}{2}+\arctan\left(\alpha_{co}T_h\right)-\arctan\left(\alpha_{co}T_p\right)-\alpha_{co}T_d \tag{78}$$

Almost invariably, the crossover frequency will be well above the plant pole frequency, thus the angular contribution from $\arctan\left(\alpha_{co}T_p\right)$ can be approximately $\pi/2$, thus the open loop phase margin φ_{pm} can be approximated as

$$\varphi_{pm}\approx\pi-\left(\pi-\arctan\left(\alpha_{co}T_h\right)+\alpha_{co}T_d\right)$$
$$\varphi_{pm}\approx\arctan\left(\alpha_{co}T_h\right)-\alpha_{co}T_d \tag{79}$$

And the crossover frequency

$$\frac{\arctan\left(\alpha_{co}T_h\right)-\varphi_{pm}}{T_d}=\alpha_{co} \tag{80}$$

A maximum value of α_{co} for a given φ_{pm} occurs for $\arctan\left(\alpha_{co}T_h\right)=\frac{\pi}{2}$, yielding

$$\alpha_{co,\max}=\frac{\pi/2-\varphi_{pm}}{T_d} \tag{81}$$

Looking at the gain of the open loop transfer function at $s = j\alpha_{co,\max}$ with the approximation of (77), we get

$$\left|G_{ol}\left(j\alpha_{co,\max}\right)\right| = \frac{k_{pr}}{2R\cdot\alpha_{co,\max}T_h}\cdot\frac{\sqrt{1+\left(\alpha_{co,\max}T_h\right)^2}}{\sqrt{1+\left(\alpha_{co,\max}T_p\right)^2}} = 1$$
$$k_{pr} = 2R\cdot\alpha_{co,\max}T_h\frac{\sqrt{1+\left(\alpha_{co,\max}T_p\right)^2}}{\sqrt{1+\left(\alpha_{c,\max}T_h\right)^2}} \tag{82}$$

With the assumption of $\alpha_{co,\max}T_p \gg 1$ and $\alpha_{co,\max}T_h \gg 1$ the following expression for the proportional gain is achieved

$$k_{pr} \approx 2R\alpha_{co,\max}T_h\cdot\frac{\alpha_{co,\max}T_p}{\alpha_{co,\max}T_h} = 2R\alpha_{co,\max}T_p = 2L\alpha_{co,\max} \tag{83}$$

Setting $\arctan\left(\alpha_{co,\max}T_h\right) = 85^\circ$ (close to $\pi/2$) further yields

$$T_h \approx \frac{10}{\alpha_{co,\max}} \tag{84}$$

For proper system stability, the cross – over frequency is often set to be [9]

$$\alpha_{co,\max} \approx \frac{2\pi}{10\cdot T_s} \tag{85}$$

Further, the following expression for the resonance bandwidth with $i = 2$ is commonly used [9]

$$\alpha_h = \frac{1}{i\cdot T_h} \tag{86}$$

How the resonance bandwidth affects the frequency response of $G_{pr}(s)$ is shown in Figure 15. Increasing the bandwidth decrease the gain at resonance frequency as well as widen the frequency range of higher gain than $20\log\left(k_{pr}\right)$.

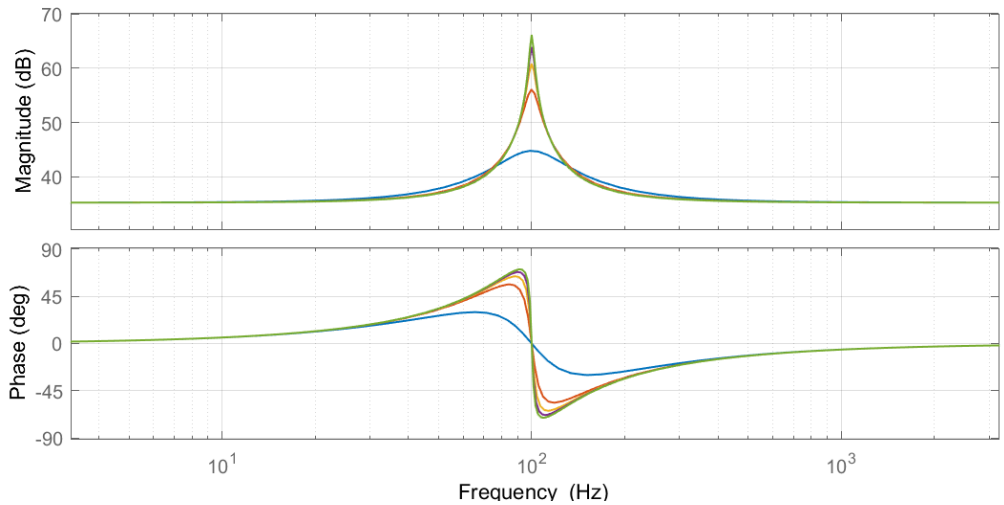


Figure 15 - Frequency response of Gpr(s) for varying values of the resonance bandwidth.

3.5.2.1. Discrete – Time Realization of the PR Controller

The continuous time transfer function of (74) is discretized in order to implement the control logic digitally. Since the controller inhibit important dynamics at the tuned frequency, care is needed in this process of discretization. Besides the discretization stability issue of ensuring that the left half s – plane poles must be mapped inside the unity circle of the z – plane in order to preserve the stability properties of $G_{pr}(s)$, further cautions must be made [9]

- The resonant frequency $(h\omega_1)$ should not be shifted due to the discretization. The prewarped Tustin method, which involves the substitution

$$s \rightarrow \frac{h\omega_1}{\tan\left(\frac{h\omega_1 T_s}{2}\right)} \frac{z-1}{z+1} \tag{87}$$

  has the desired mapping and frequency properties.

- For *fixed point* implementation, realization of (87) should be done using a structure which have good numerical properties. For sampling rates much higher than the r esonant frequency, the poles of a discretized resonator (controllers and filters which have poles in the s – plane close to $s = \pm jh\omega$) cluster near z=1 in the z – plane, resulting in poor numerical properties for the standard direct – form structure based on the z^{-1} (one sample delay) operator of (87). The direct – form II transposed (DFIIt) structure based on the delta - operator

$$\delta = z - 1 \tag{88}$$

  is shown to have better numerical properties regarding quantification noise. DFIIt is currently the preferred structure for fixed point resonator implementation [9, 25].

Applying equation (87) and (88) into (70) yields the following transfer function for the resonant filter [9]

$$G_h(z) = \frac{b_0\delta^2 + b_1\delta + b_2}{\delta^2 + a_1\delta + a_2} \tag{89}$$

With the coefficients

$$r_h = \frac{1}{1 + \frac{\alpha_h}{2h\omega_o}\sin(h\omega_0 T_s)} \tag{90}$$

$$k_h^{'} = \frac{k_h r_h}{2h\omega_0} \tag{91}$$

$$b_0 = k_h^{'}\left(\sin(h\omega_0 T_s + \phi_h) - \sin(\phi_h)\right) \tag{92}$$

$$b_1 = k_h^{'}\left(3\sin(h\omega_0 T_s + \phi_h) - 4\sin(\phi_h) - \sin(h\omega_0 T_s - \phi_h)\right) \tag{93}$$

$$b_2 = k_h^{'}\left(2\sin(h\omega_0 T_s + \phi_h) - 4\sin(\phi_h) - 2\sin(h\omega_0 T_s - \phi_h)\right) \tag{94}$$

$$a_1 = 2\left(1 - r_h\cos(h\omega_0 T_s)\right) \tag{95}$$

$$a_2 = 2r_h\left(1 - \cos(h\omega_0 T_s)\right) \tag{96}$$

The realization of is obtained by using the delayed numerical integrator as $\delta^{-1} = 1/(z-1)$.

For fixed point implementation in hardware such as FPGA, ASICs or fixed – point DSPs, the mentioned quantification issue is important. In this thesis, the PR controller is only used to compare Simulink simulation results between PR and repetitive control in a floating – point environment. Only the resource use and estimated critical path of the PR controller tuned to $2\omega_0$ and $4\omega_0$ is investigated to some degree. The PR controller is not simulated with fixed point data types, but if the controller structure is to be implemented in fixed – point using the methods described in this thesis, care must be taken during quantization to obtain proper performance.

3.5.3. Repetitive Control

Repetitive control can be considered as a family of algorithms especially useful for rejecting periodic disturbances or tracking periodic signals with multiple major frequency components. Methods like PR control needs to be designed to a specific frequency: the frequency component which is to be suppressed. A PI controller needs a multiple – of – three phase system in order to transform the oscillating quantities into DC quantities for a proper control without steady state error. This requirement, which needs *dq* – transform and Phase Locked Loops (PLL), increase complexity for an integrated solution [26]. The repetitive control scheme is able to act on sinusoidal signals and is effective for multi – harmonic suppression.

3.5.3.1. Internal Model Principle

The repetitive controller is based on the Internal Model Principle, and is widely used to provide precise tracking of periodical and highly complex waveforms [26]. The Internal Model Principle of control theory states that for an algorithm to be able to perfectly reject a signal, it must contain a model of that signal. The controlled output will track a set of reference inputs without steady state error if the model which generate these references is included in the stable, closed loop system. For example, no steady state error will appear for a step reference command $r(s) = (1/s)$ in a type – 1 stable feedback system which has an integrator $(1/s)$ in the loop, as the this is the *generator* of the step function. Likewise, the resonance frequency response defined in (72) is a representation of the sine – wave, thus including this signal generator in the controller should result in perfect tracking/rejection of sine signals [27]. If one wants to obtain *zero* difference between the command reference and the measured signal (the error), the control algorithm must be able to self – generate a signal in the *absence* of any error signal. Thus, any input that repeats with a known period, can be addressed by the controller with a periodic signal generator. These types of control are termed repetitive control.

A periodic signal generator in the feedback of the controller algorithm satisfies the Internal Model Principle. To obtain the periodic signal generator, a simple delay element can be placed in a positive feedback loop such as in Figure 16.

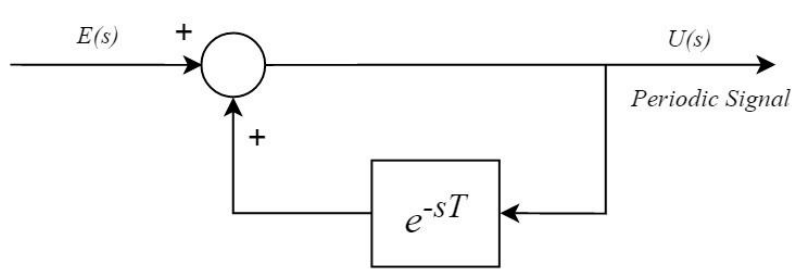


Figure 16 - Block diagram of continuous internal model.

This closed loop system will in the continuous time domain result in an infinite number of marginally – stable poles. A signal with very fast transitions in the time domain requires a high bandwidth signal generator capable of creating this high frequency content. Stabilizing this system is challenging, as the high frequency bandwidth may interact with other high frequency elements within the system, such as the converter modulator. The controller must therefore be design with the complete system in mind [28].

The internal model is often depicted with the delay inserted in the forward path. With a specified sampling period $T_s = 1/f_s$ and the period of the periodic disturbance which is to be rejected as $T_p = 1/f_p$. The required length of the delay can be obtained as $N_s = T_p / T_s$, with $T_p = N_s T_s$, where N_s being the number of sample delays. Following the analysis of [29, 30], the internal model is given as

$$I(s) = \frac{e^{-sT_p}}{1 - e^{-sT_p}} \tag{97}$$

Which by inserting $T_p = N_s T_s$ gives

$$I(s) = \frac{e^{-sN_sT_s}}{1 - e^{-sN_sT_s}} \xrightarrow{s=j\omega} I(j\omega) = \frac{e^{-j\omega N_sT_s}}{1 - e^{-j\omega N_sT_s}} \tag{98}$$

With Euler's formula, (98) can be written as

$$I(j\omega) = \frac{\cos(2\pi f T_s N_s) - j\sin(2\pi f T_s N_s)}{1 - \cos(2\pi f T_s N_s) + j\sin(2\pi f T_s N_s)} \tag{99}$$

(99) shows that when (fT_sN_s) is an integer, the cosine term equals 1 and the sine term equals zero, yielding infinite gain at integer multiplies of (fT_sN_s). This is also shown by the poles of (97), which is located at $s = \pm jk/T_p,\ k \in \mathbb{N}$, theoretically giving infinite gain at frequencies k/T_p, as seen in Figure 17 [29].

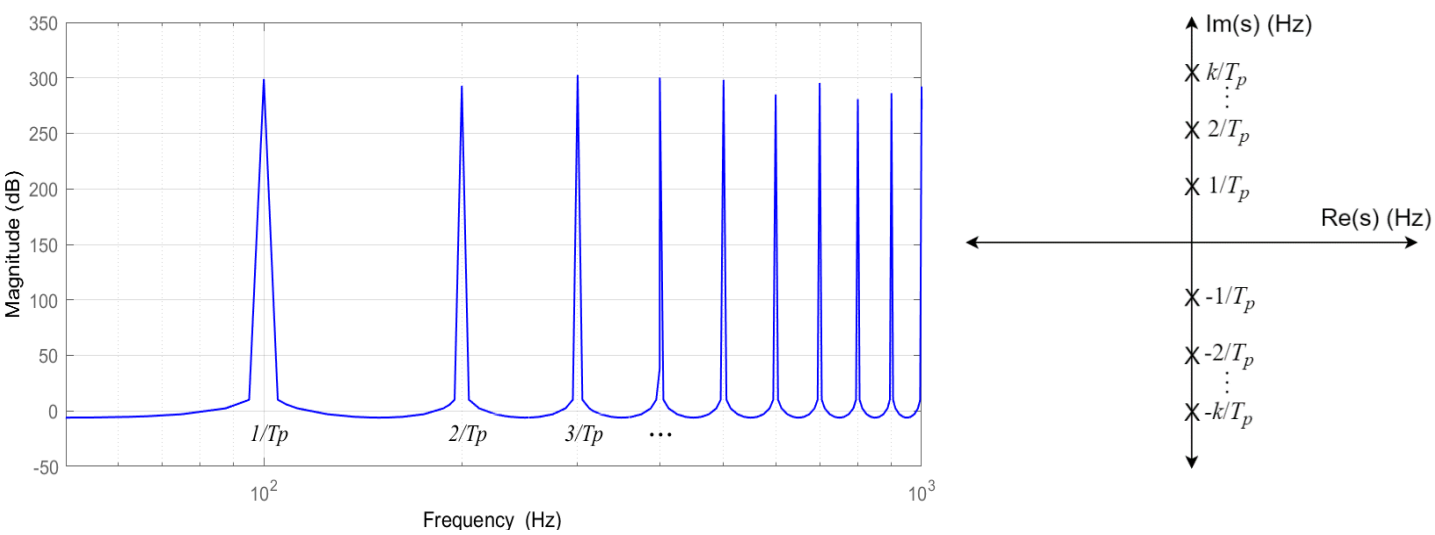


Figure 17 - Bode plot of the internal model frequency response (left) and pole placement (right).

Due to the nature of the exponential function, (97) can be rewritten with $\omega_p = 2\pi / T_p$ as [31]

$$G_I(s) = k_r \cdot I(s) = -\frac{k_r}{2} + \frac{k_r}{sT_p} + \frac{2k_r}{T_p}\sum_{n=1}^{\infty}\frac{s}{s^2 + (n\omega_p)^2} \tag{100}$$

Where k_r is the repetitive controller gain. (100) shows that the repetitive controller is equivalent to a parallel combination of a negative proportional part, integral part and an infinite sum of resonance controllers at frequency – multiples of ω_p. The repetitive controller thus contains internal models of every sinusoidal signal at the frequencies of interest.

The implementation and realization of repetitive control in the continuous time domain is a complicated matter [30]. Fortunately, the implementation in discrete time domain is easier, and have advantages in engineering applications where the controller often is to be implemented on DSP or SoC's. The following relationship $z = e^{sT}$ between the continuous s – domain and the discrete z – domain can be used to create a discrete internal model. By substituting e^{-sT} with z^{-N_s}, the feedback will hold the output signal for the required amount such that a periodic signal of the required period will be generated by the internal model. The discrete internal model that should be included in the loop is [29]

$$I(z) = \frac{z^{-N_s}}{1 - z^{-N_s}} = \frac{1}{z^{N_s} - 1} \tag{101}$$

3.5.3.2. Repetitive Controller Structures

Several repetitive control structures have been proposed in literature in order to track or suppress periodic references or disturbances [26, 29-33]. Two common structures are presented.

3.5.3.2.1. Plug – In Repetitive Control with series PI control (RC System 1)

The system depicted in Figure 18 is termed *Plug– in Repetitive Control* and is the most commonly used control topology when utilizing repetitive control [29, 30, 34, 35]. This structure has a controller $G_c(z)$ in *series* (rather than in the parallel forward path presented in 3.5.3.2.2), and will be termed *series RC* or *RC system 1*. The controller $G_c(z)$ is the *nominal controller* of the system, i.e. a controller for the system *without* the additional plug – in RC. The RC algorithms is to act as additional controller action to handle periodic disturbances which the nominal controller handle less effectively. The feedforward path for addressing the resistive voltage drop over the arm resistance and DC – link voltage, and the converter delay are ignored in the block schematic and further analysis due to simplicity. Equation (69) describing the control dynamics is applied for this controller as well.

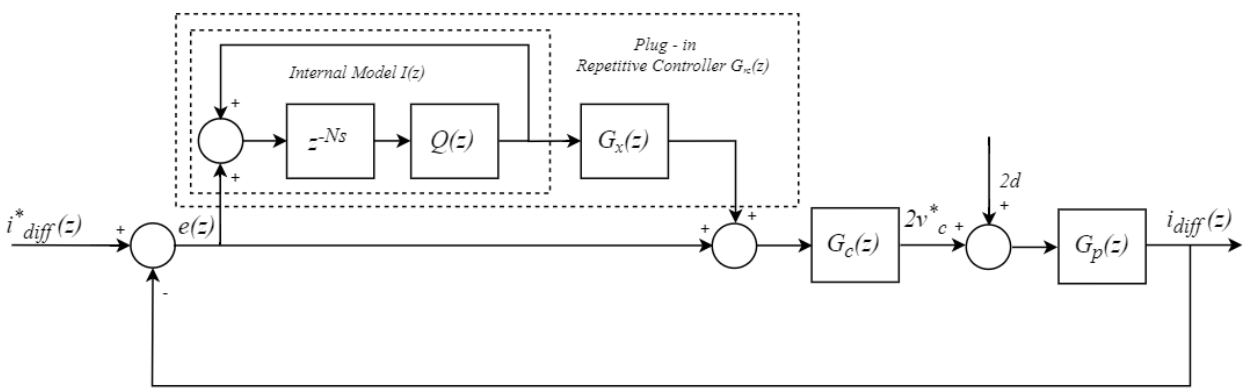


Figure 18 - Block diagram for the series repetitive controller.

The repetitive controller is paralleled with a unity forward path and have a nominal controller in series. With the system in Figure 18, the following transfer function for the internal model $I(z)$ and the repetitive controller $G_{rc,1}(z)$ are obtained

$$I(z) = \frac{z^{-N_s}Q(z)}{1 - z^{-N_s}Q(z)} = \frac{Q(z)}{z^{N_s} - Q(z)} \tag{102}$$

$$G_{rc,1}(z) = G_{x,1}(z) \cdot I(z) = G_{x,1}(z) \cdot \frac{Q(z)}{z^{N_s} - Q(z)} \tag{103}$$

With the subscript 1 signaling RC system 1.

The sensitivity transfer function $S_1(z)$ from the error $e(z)$ to the reference differential current $i_{diff}^*(z)$ is

$$S_1(z)=\frac{e(z)}{i_{diff}^*(z)}=\frac{\left(1-z^{-N_s}Q(z)\right)\frac{1}{\left(1+G_y(z)\right)}}{1-z^{-N_s}Q(z)\left(1-\frac{G_{x,1}(z)G_y(z)}{1+G_y(z)}\right)}$$

$$G_y(z)=G_c(z)G_p(z) \tag{104}$$

Which further can be written as

$$S_1(z)=S_y(z)S_{\text{mod},1}(z) \tag{105}$$

where $S_y(z)$ is the sensitivity function of the system *without* the repetitive controller

$$S_y(z)=\frac{1}{1+G_y(z)} \tag{106}$$

and $S_{\text{mod},1}(z)$ is the modifying sensitivity function

$$S_{\text{mod},1}(z)=\frac{\left(1-z^{-N_s}Q(z)\right)}{\left(1-z^{-N_s}Q(z)\right)\left(1-\frac{G_{x,1}(z)G_y(z)}{1+G_y(z)}\right)} \tag{107}$$

The closed – loop system poles are the poles of $S_1(z)$, or the poles of $S_y(z)$ and $S_{\text{mod},1}(z)$, i.e. the zeros of the characteristic equation

$$\left(1-z^{-N_s}Q(z)\right)\left(1-\left(G_{x,1}(z)G_y(z)/1+G_y(z)\right)\right) \tag{108}$$

The closed loop system of Figure 18 is stable if the following conditions are fulfilled [29, 30]

1. The closed loop system without the repetitive controller $G_0(z)$ is stable, where

$$G_o(z)=\frac{G_y(z)}{1+G_y(z)} \tag{109}$$

2. The modifying sensitivity function can be regarded as a closed loop system with the positive feedback term $\left(1-z^{-N_s}Q(z)\right)\left(1-\left(G_{x,1}(z)G_y(z)/1+G_y(z)\right)\right)$, which gives a sufficient condition for stability as

$$\left\|z^{-N_s}Q(z)\left(1-G_{x,1}(z)G_o(z)\right)\right\|_\infty<1 \tag{110}$$

This condition can further be interpreted in the context of Small Gain Theorem, and can be split into two additional conditions

i)

$$\left\| Q(z) \right\|_{\infty} < 1 \tag{111}$$

Since $Q(z)$ is a design filter, this condition poses constraint on the filter design.

ii)

$$\left\| \left(1 - G_{x,1}(z) G_o(z)\right) \right\|_{\infty} < 1 \tag{112}$$

Where in 2., $G(z)$ is the discrete transfer function discretized with the sampling time T_s and its H_∞ norm is defined as

$$\left\| G(z) \right\|_{\infty} = \max\left\{ \left\| G(e^{j\omega T_s}) \right\|, \ \ \forall \omega \in \left[0, \frac{\pi}{T_s} \right] \right\} \tag{113}$$

In the case where $G_{x,1}(z) = k_r \left(G_o(z)\right)^{-1}$ and $Q(z) = 1,$ the closed loop system poles is [29]

$$z = \sqrt[N_s]{\left|1 - k_r\right|} e^{\frac{2k\pi j}{N_s}}, \ k = 0, ..., N_s - 1 \tag{114}$$

The poles are uniformly distributed over a circle of radius $\sqrt[N_s]{\left|1 - k_r\right|}$ together with the poles of $1 + G_y(z),$ showing why the system without the repetitive controller needs to be stable before enabling the RC. For a discrete time – system to be asymptotically stable, all the roots of the system closed – loop transfer function must be inside the unit circle.

$$1 + a_1 z^{-1} + ... + a_n z^{-n} = 0 \rightarrow \left|z\right| < 1 \tag{115}$$

which gives the following restrictions on the repetitive gain

$$\forall k_r \in \left[0, 2\right] \tag{116}$$

For values outside this range, the system poles will migrate outside the unit circle and induce instability, which is illustrated in Figure 19 and Figure 20.

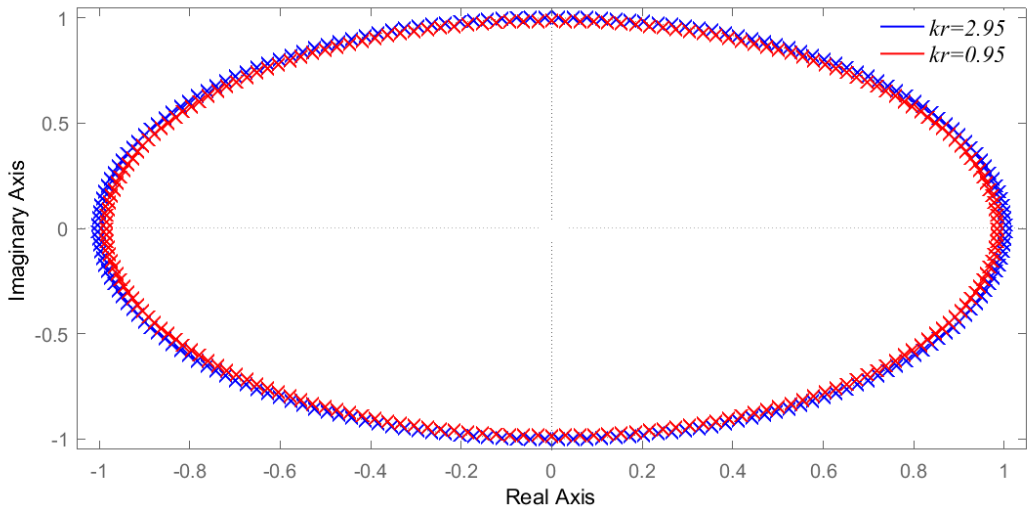


Figure 19 - Pole - zero map showing pole migration depending on kr.

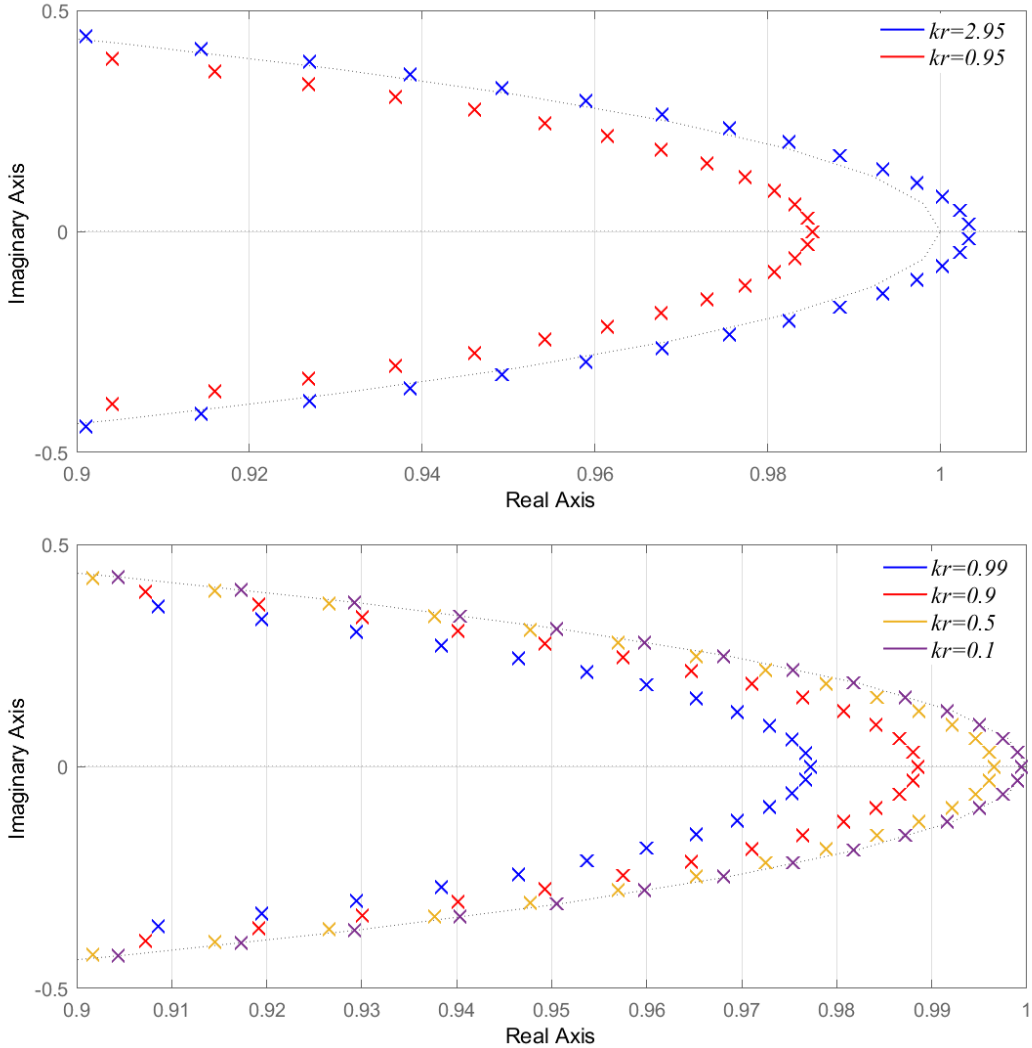


Figure 20 – Zoomed pole - zero map showing pole migration depending on kr.

3.5.3.2.2. Plug – In Repetitive Control with parallel PI control (RC System 2)

The system depicted in Figure 21 have a plug – in type repetitive controller in *parallel* with the nominal controller $G_c(z)$. For this system, the subscript 2 is used to distinguish it from RC system 1, and it will be termed *parallel RC* or *RC system 2*.

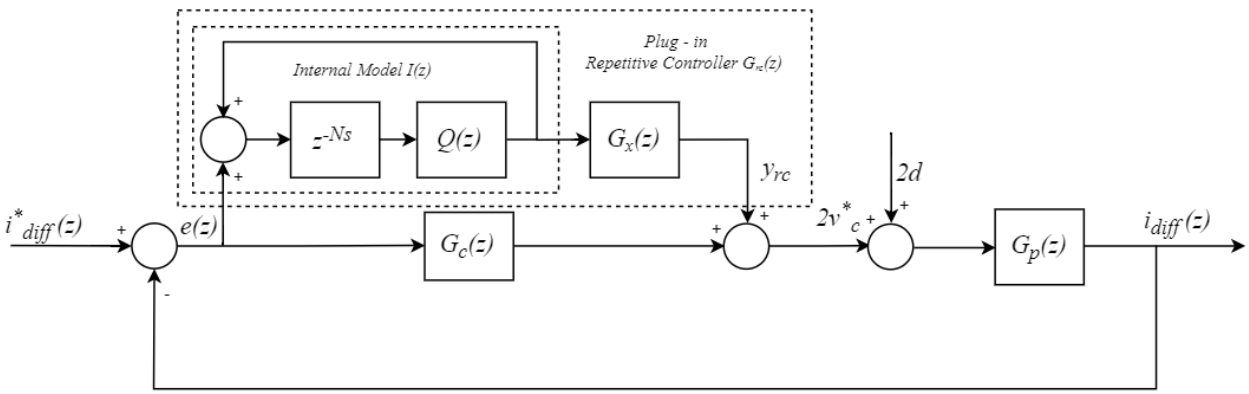


Figure 21 - Plug - in repetitive control with nominal control in parallel.

By introducing the repetitive controller in parallel to the nominal controller, different system transfer functions arise. Following the same transfer function derivations as in the previous section, the internal model $I(z)$ and the repetitive controller $G_{rc,2}(z)$ are similar as in (102) and (103), with only the filter $G_x(z)$ changed. Further, the transfer function from the error $e(z)$ to the reference current $i^*_{diff}(z)$, the sensitivity function $S_2(z)$, becomes

$$S_2(z) = \frac{\left(1 - z^{-N_s}Q(z)\right)\left(\dfrac{1}{1+G_y(z)}\right)}{1 - z^{-N_s}Q(z)\left(1 - \dfrac{G_{x,2}(z)G_p(z)}{1+G_y(z)}\right)} \tag{117}$$

And the modifying sensitivity function $S_{\text{mod},2}(z)$ becomes

$$S_{\text{mod},2}(z) = \frac{\left(1 - z^{-N_s}Q(z)\right)}{1 - z^{-N_s}Q(z)\left(1 - \dfrac{G_{x,2}(z)G_p(z)}{1+G_y(z)}\right)} \tag{118}$$

The same stability conditions of RC system 1 applies to system 2. Due to the differences in the system transfer functions, stability condition (110) is changed to

$$\left\| z^{-N_s}Q(z)\left(1 - \frac{G_{x,2}(z)G_p(z)}{1+G_y(z)}\right)\right\|_\infty < 1 \tag{119}$$

And (112) changed to

$$\left\| \left(1 - \frac{G_{x,2}(z) G_p(z)}{1 + G_y(z)} \right) \right\|_\infty < 1 \tag{120}$$

Applying the filter

$$G_{x,2}(z) = k_r \left(\frac{G_p(z)}{1 + G_y(z)} \right)^{-1} \tag{121}$$

And $Q(z) = 1$ gives the same system poles as (114) and the same restrictions on the repetitive gain as in (116).

3.5.3.3. Error Convergence of the Repetitive Controller

The transfer function between the disturbance $d(z)$ and the error $e(z)$ with the reference set to zero can for RC system 1,2 be written as

$$e(z) = -G_p(z) \cdot S_{1,2}(z) \cdot d(z) = \frac{-G_p(z)\left(1 - z^{-N_s} Q\right)}{1 - z^{-N_s} Q(z) \left(1 - \frac{G_{x,1,2}(z) G_{y,p}(z)}{1 + G_y(z)} \right)} \cdot d(z) \tag{122}$$

With the filter for both systems implemented as discussed, (122) is reduced to

$$e(z) = \frac{-G_p(z)\left(1 - z^{-N_s} Q(z)\right)}{1 - z^{-N_s} Q(z)(1 - k_r)} \cdot d(z) \tag{123}$$

Supposing $|Q(z)| = 1$ and the disturbance to be purely repetitive with period T_p, meaning its value is the same at $(t - T_p)$ and t such that $d(z) = z^{-N_s} d(z)$, the error magnitude can be seen to be [36]

$$e(z) = \left| \frac{G_p(z)\left(1 - z^{-N_s} Q(z)\right)}{1 - z^{-N_s} Q(z)(1 - k_r)} \right| \cdot \left| d(z) \cdot z^{-N_s} \right| \tag{124}$$

For a frequency signal at $\omega = \omega_p = 2\pi / T_p$ and re – addressing the fact that $z^{-N_s} = e^{-sT_s N_s} = e^{-sT_p}$

$= e^{-j\omega_p T_p} = \underbrace{e^{-j2\pi} = 1}_{\text{Euler's Identity}}$, the steady – state error magnitude becomes for frequency components at ω_p

$$\left| e\left(e^{j\omega_p T_s}\right) \right| = \frac{\left| G_p\left(e^{j\omega_p T_s}\right) \right| \left(1 - 1 \cdot \left| Q\left(e^{j\omega_p T_s}\right) \right| \right)}{1 - 1 \cdot \left| Q\left(e^{j\omega_p T_s}\right) \right| \left| (1 - k_r) \right|} \cdot \left| d\left(e^{j\omega_p T_s}\right) \right| = 0 \tag{125}$$

3.5.3.4. Repetitive Controller Elements

The control system comprises of the following elements

- *Stability filter* $G_x(z)$

  The filter $G_x(z)$ is used to fulfill the second stability condition. The fundamental issue is to provide enough leading phase to cancel out the phase of $G_o(z)$. In the case where $G_o(z)$ is minimum phase, $G_x(z)$ is implemented as the inverse of the closed loop system seen by the repetitive controller [29]. This filter structure will be used further in this thesis. For non – minimum phase systems, different strategies need to be used to ensure total system stability, which can be found in [29, 37].

  In [35], the stability filter comprises of a conventional low pass filter $L(z)$ with the cut – off frequency $\omega_{co} = 10 \cdot \omega_0$, which provides magnitude compensation for $G_o(z)$, and a time – advance – unit z^k, which provides phase compensation for $L(z)G_o(z)$. This filter structure will not be used further in this thesis, but is mentioned to show that the filter is implemented differently throughout the literature.

- *Robustness filter* $Q(z)$

  The cancellation performed by $G_x(z)$ is not ideal, and will not be able to completely suppress high order frequency components present in the error signal. The MMC inverter is a harmonic generator in itself, where the circulating current component consist of mainly low even order multiples of the fundamental frequency and higher order components dependent on the specific modulation strategy. In addition, conversion between digital and analog signals, and delays in the processing system will cause high frequency components in the error signal. The internal model must thus avoid integrating the high order components which the system is unable to suppress to avoid saturation/integrator wind – up. This can be, and often is, realized by a zero – phase, low pass FIR filter [29, 35, 38], as long as the filter fulfill (111). Conventional low pass filter may also be used, but it will be necessary to compensate for the phase shift they provoke [29]. Since such filters are symmetrical, they depend on future inputs and is thus non – causal. This is however not a problem when used with repetitive controller, as the filter is used in the positive feedback path of the internal model which is delayed with exactly one signal period, and thus intrinsically exhibits knowledge of the future error [33].

- *Gain* k_r

  The repetitive controller gain k_r ultimately decides the amplitude of the controller output. The design of is a trade-of between stability robustness and steady – state performance. Appropriate selection of k_r also decrease the error caused by non – harmonic components (not affected by the high gain of the internal model)[29].

- *Nominal controller* $G_c(z)$

The nominal controller is commonly chosen to be either pure proportional or proportional – integral. For a PI controller, the transfer function in the continuous *s* – domain is

$$G_c(s) = k_p + \frac{k_i}{s} \tag{126}$$

with the proportional gain k_p and integral gain k_i. For pure proportional, (126) is reduced to k_p. The discrete transfer function discretized by ZOH can be written as

$$G_c(z) = \frac{zk_p + (Tk_i - k_p)}{z-1} \tag{127}$$

With the Tustin (Bilinear) approximation, the transfer function is written as

$$k_p + k_i\left(\frac{T_s}{2}\frac{z+1}{z-1}\right) \tag{128}$$

- *The plant* $G_p(z)$

The plant transfer function governing the generation of circulating current is described in The Circulating Current chapter. It is comprised by the arm inductor and resistance, yielding the continuous transfer function presented in (44), which is given here again for convenience

$$G_p(s) = \frac{1}{2sL + 2R} = \frac{1}{2R}\left(\frac{1}{1+sT_p}\right),\ T_p = \frac{L}{R} \tag{129}$$

The discrete plant transfer function is thus described as

$$G_p(z) = \left(1 - z^{-1}\right) Z\left(\frac{1}{s}\frac{1}{2R}\left(\frac{1}{1+sT_p}\right)\right) = \frac{1}{2R}\left(\frac{1-e^{-T_s/T_p}}{z-e^{-T_s/T_p}}\right) \tag{130}$$

discretized with ZOH and with the Bilinear (Tustin) transform, the discrete transfer function becomes

$$G_p(z) = \left(\frac{1}{2sL+2R}\right)\Bigg|_{s=\frac{2}{T_s}\frac{z-1}{z+1}} = \frac{z+1}{z\left(2R + \frac{4L}{T_s}\right) + \left(2R - \frac{4L}{T_s}\right)} \tag{131}$$

3.5.3.5. Design of the Nominal Controller

The nominal controller should first and foremost ensure the stability of (109). A PI nominal control structure is frequently used in literature [33, 35, 39], but when used with a repetitive control structure, the integral part may not necessary. From chapter 2.2, it is clear that the differential current comprises of a pure DC – component and oscillating component at even multiples of the fundamental frequency and high order harmonics due to the modulation. As the differential current should settle on its DC – value, the error entering the control structure is purely sinusoidal in nature. This sinusoidal quantity should ideally be suppressed to zero. As stated in chapter 3.5.3.1 to perfectly reject or track a signal, the controller should contain a generator of that particular signal.

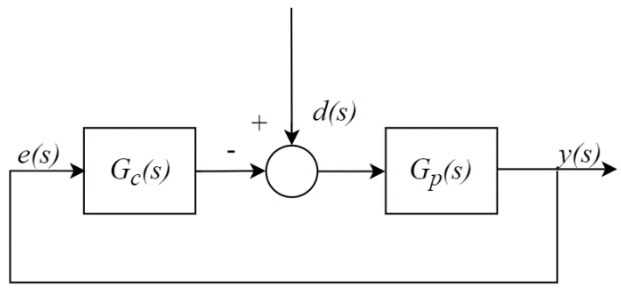


Figure 22 - Error block diagram with reference set to zero.

Considering the block diagram of Figure 22 with the reference set to zero and the disturbance signal $d(s)$, the transfer function from the disturbance to the error yields

$$e(s) = \frac{G_p(s)}{1+G_p(s)G_c(s)} \cdot d(s) = \frac{G_p(s)}{1+G_y(s)} \cdot d(s) \tag{132}$$

For a sinusoidal disturbance with the poles $s = \pm j\omega_p$, any system containing a model of these poles would yield zero error at ω_p. With the plant transfer function $G_p(s)$ defined in (129), the error function can be written as

$$\begin{gathered} e(s) = \frac{1}{2R(1+sT_p)+G_c(s)} \cdot d(s) = \frac{1}{2R(1+sT_p)+\frac{N_c(s)}{D_c(s)}} \cdot d(s) \\ \frac{D_c(s)}{2R(1+sT_p)\cdot D_c(s)+N_c(s)} \cdot d(s) \end{gathered} \tag{133}$$

For a disturbance signal which contain sinusoidal signals as in (68), having poles at $s = \pm jn\omega_p$ and $G_c(s)$ containing a signal generator of (68) having the poles at $s = \pm jn\omega_p$ as defined in (100), such that $D_c(s) = \prod_{n=1}^{k}\left(s^2 + (n\omega_p)^2\right)$, the error $e(s)$ will be zero for frequencies at $n\omega_p$ up to the chosen design frequency $k\omega_p$ as long as the closed loop system is stable.

Furthermore, as the repetitive controller includes both an integral term and proportional term, there should not be necessary to add an integral term in the nominal controller whenever the repetitive controller is enabled to remove sinusoidal disturbances. A proportional term may be added to adjust the settling time of the differential current, as this is affected by the system bandwidth.

The design of the nominal controller is, however, performed assuming the use of an integral part. This is done to make the system *without* the repetitive controller perform optimally. The resulting system *with* the repetitive controller should not need an integral term as discussed, and will function optimally as will be shown in simulations.

The PI controller is designed to be robust to account for unmodeled aspects, model uncertainties and modelling errors, i.e. with enough phase – and gain margin to ensure system stability over a wide frequency range. To avoid interference from the high order harmonics in the circulating current due to the PWM, the closed – loop bandwidth α_b [r/s] is chosen to be one tenth of the carrier frequency f_c. The closed loop bandwidth is approximated by the open – loop cross – over frequency α_{co} in the design, giving

$$\alpha_{co} = \frac{\omega_c}{10} = \frac{2\pi f_c}{10} \tag{134}$$

With the converter delay introduced in the open – loop, the open loop transfer function is written as

$$G_{ol}(s) = G_c(s)G_p(s) = G_y(s) = \frac{k_p}{2R}\left(1+\frac{1}{sT_i}\right)\left(\frac{1}{1+sT_p}\right)e^{-sT_d} \tag{135}$$

With $T_i = k_p / k_i$ being the integral time constant. The tuning of these parameters is based on [24] and the the same concepts as the PR controller tuning. The parameters are obtained as follows

$$\begin{gathered} \varphi_{pm} - \pi = \arctan(\alpha_{co}T_i) - \frac{\pi}{2} - \underbrace{\arctan(\alpha_{co}T_p)}_{\approx \pi/2} - \alpha_{co}T_d \\ \varphi_{pm} \approx \arctan(\alpha_{co}T_i) - \alpha_{co}T_d \end{gathered} \tag{136}$$

Setting $\arctan(\alpha_{co}T_i)$ as close to $\pi/2$ as possible, say around 85°, gives

$$\begin{gathered} T_i = \frac{10}{\alpha_{co}} \\ k_i = \frac{k_p}{T_i} \end{gathered} \tag{137}$$

For small R, the proportional gain will mainly be decided by pole placed by the plant inductance. Setting (135) at α_{co} equal to unity, yields the following approximation for k_p

$$k_p \approx 2L\alpha_{co} \tag{138}$$

The continuous controller is discretized with ZOH and verified through simulation in Simulink with the system block diagram as shown in Figure 9, with the PI controller as the tuned controller. The tuned parameters for a phase margin $\varphi_{pm} = 60^{\circ}$ and crossover frequency $\alpha_{ca} = f_c / 10 = 1000$ [Hz] gives $k_p = 57.8$, $k_i = 36500$ and $T_i = 27.4$ [μs]. Figure 23 and Figure 24 shows plots of system characteristics and closed – loop system step response.

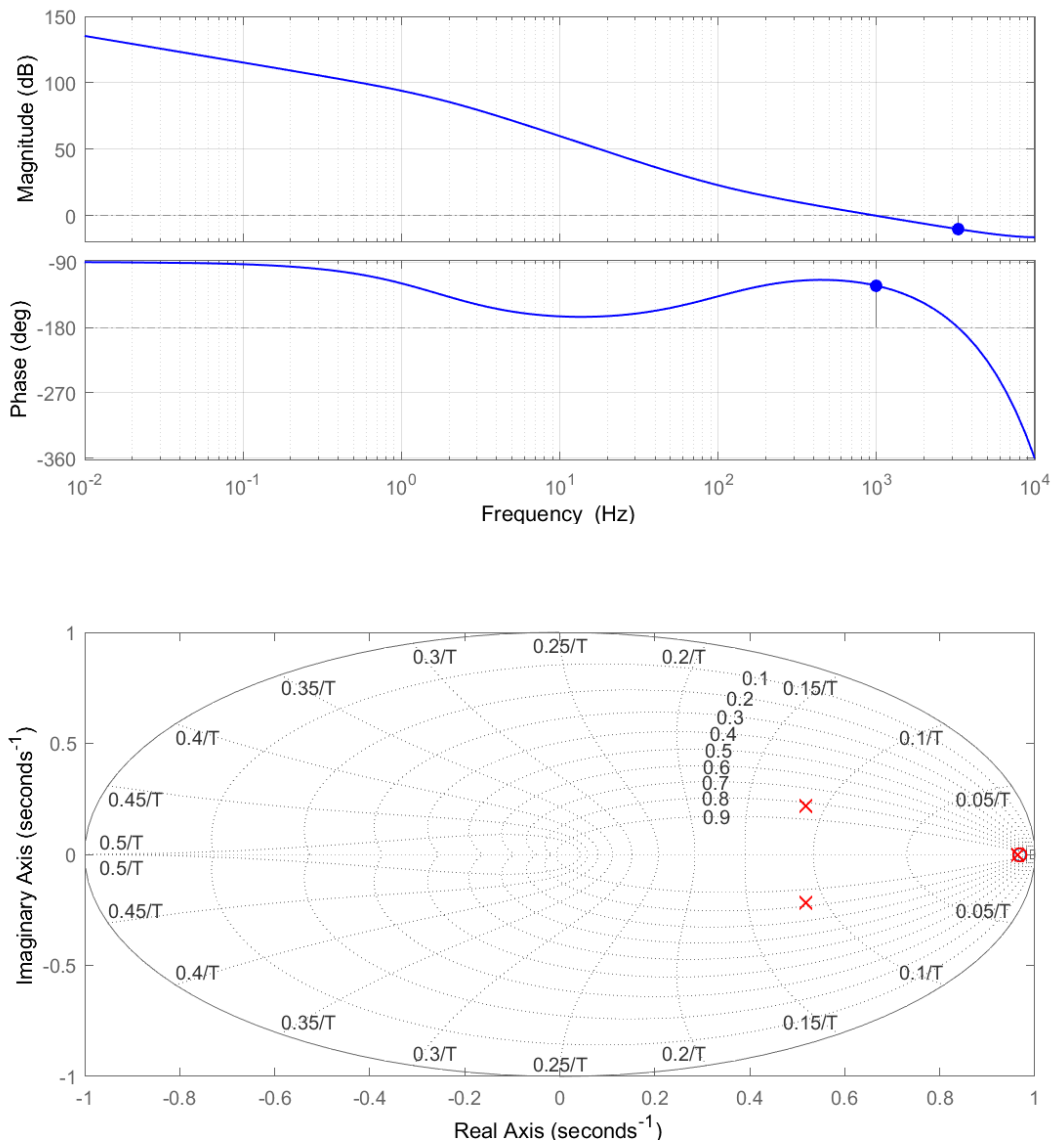


Figure 23 - Bode plot of open loop frequency response Gy (upper) and pole - zero map of the closed loop transfer function Go (lower).

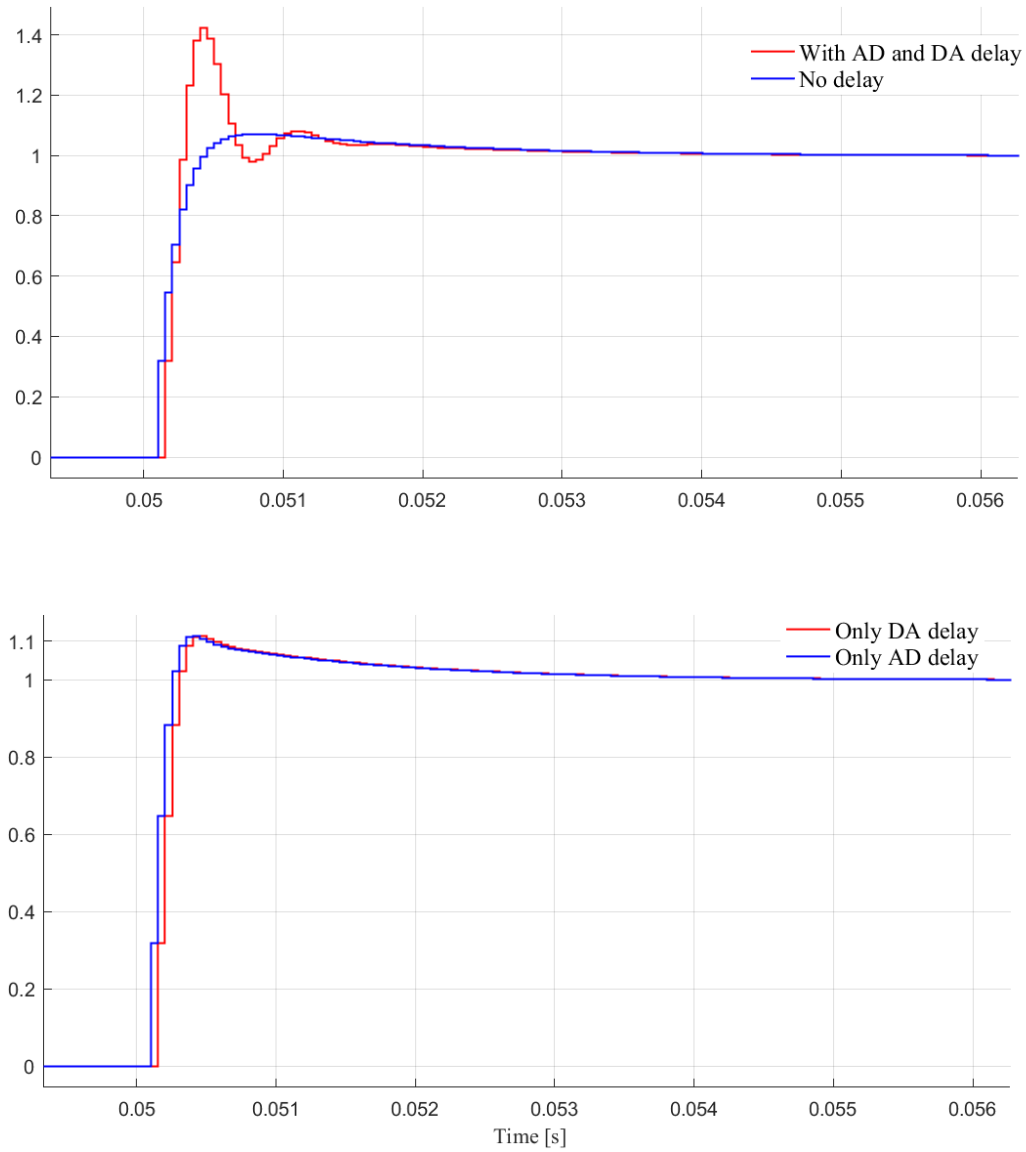


Figure 24 - Step response of closed – loop system for different setting of feedback delay (AD) and controller output delay (DA).

This tuning provides an aggressive controller with high bandwidth within the robustness restriction posed on phase and gain margin. Such aggressiveness may not be necessary when used *with* a repetitive controller to eliminate the error in a system which regulates the differential current. This is confirmed through Simulink simulations *with* repetitive control. Thus, different values of k_p and k_i providing different controller bandwidths are investigated and compared to look at the error the nominal controller generates alone and with the repetitive controller structures. The maximum parameter values are chosen to be the ones designed in this chapter. The parameters of Table 2 are analyzed.

Crossover frequency f_c [Hz]	$k_{p,pi}$
100	5.8
200	11.6
300	17.3
400	23.1
500	28.9
600	34.7
700	40.5
800	46.2
900	52.0
1000	57.8

Table 2: Proportional gains at different crossover frequencies.

k_i is varied between 0 and 36500. How the varying gain is affecting the behavior of the differential current is shown in chapter *4.4. Simulation Results*. The simulation model can be found in *Appendix A. Simulink Simulation Block Diagram.*

3.6. Output Current Control

The output current has great effects on the total converter system performance, and the output current control should be designed with care to avoid system deterioration. The dynamic expression governing the output current was presented in (chapter) and is given again here as

$$\frac{L}{2}\frac{di_s}{dt}+\frac{R}{2}i_s=v_s-v_a \tag{139}$$

Applying the Laplace transform, the time domain equation (139) can be expressed in the frequency domain as

$$\begin{gathered}\frac{L}{2}\frac{di_s}{dt}+\frac{R}{2}i_s=v_s-v_a \xrightarrow{\mathcal{L}} \frac{L}{2}si_s+\frac{R}{2}i_s=v_s-v_a\\ i_s=\frac{2}{R+sL}\left(v_s-v_a\right)\end{gathered} \tag{140}$$

The reference i_s^* is given in (24) and is a fundamental frequency sinusoid, with phase shift dependent on the required active and reactive power of the grid. With its sinusoidal nature, perfect tracking can theoretically be achieved with a proportional – resonant controller with peak gain at $\left(h\omega_0\right)=\omega_0$ is an appropriate choice [9].

As seen in (140), the output current is driven by the voltage difference $\left(v_s-v_a\right)$, where v_a can be looked upon as the load disturbance. Transients in v_a may occur in grid locations close to the converter, i.e. rapid changes in phase – angle, sags (short duration reduction in rms voltage) and swells (short duration increase in RMS voltage). Since v_a is measurable, the dynamic performance can be improved by a feed forward, that is including a term in the control law which objective is to cancel v_a. To avoid passing through high frequency components in v_a, v_a should be passed through a BPF similar to (70). The following control law can now be obtained as [9]

$$\begin{gathered}v_s^*=k_{po}e+\underbrace{\frac{k_{ho}s}{s^2+\alpha_{ho}s+\omega_0{}^2}}_{v_r}e+\underbrace{\frac{\alpha_f\left(s\cos(\phi_0)-\omega_1\sin(\phi_0)\right)}{s^2+\alpha_f s+\omega_0{}^2}}_{v_a^{ff}}\\ e=i_s-i_s^*\\ v_s^*=k_{po}e+v_r+v_a^{ff}\end{gathered} \tag{141}$$

Where $\phi_0=\omega_0 T_d$ can be used to compensate the time delay if deemed necessary. Assuming the parasitic arm resistance is sufficiently small so it may be ignored, the parameter selection can be done in a similar way as for the circulating control PR – control parameter selection, though $2L\rightarrow L/2$, giving

$$
\begin{aligned}
k_{po} &= \frac{R\alpha_{co,\max}}{2} \cdot T_p = \frac{L\alpha_{co,\max}}{2} \\
k_{ho} &= L\alpha_o \alpha_{co,\max} \\
\alpha_o &\approx \frac{\alpha_{co,\max}}{20} \\
\alpha_h &= \frac{1}{2T_h}
\end{aligned}
\tag{142}
$$

The total control system can be summarized with Figure 25.

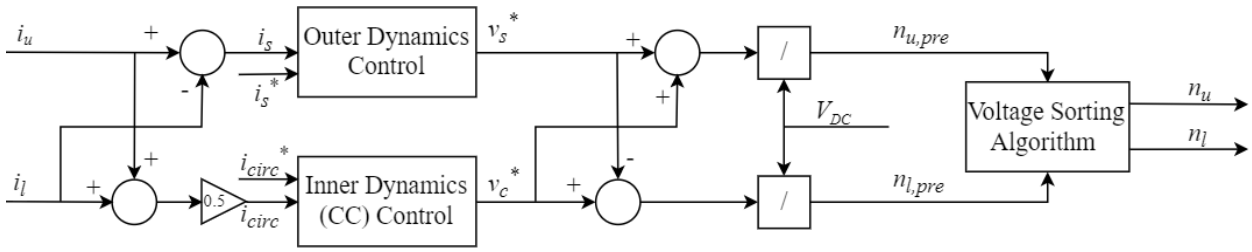


Figure 25 - Total control overview. Outer Dynamics Control governing the output current, Inner Dynamics (Circulating Current) Control governing the circulating current, Voltage Sorting Algorithm governing insertion indexes.

4. Simulation

The repetitive control is simulated on a single phase MMC in Simulink with the given model parameters of Table 3. The control algorithms are tested for different control structures and controller parameters. The circulating current is controlled to the reference circulating current $i_{circ}^{*} = 0$ giving $i_{diff}^{*} = 4$ [A].

Item	Values
Load Power	$P_{load} = 2$ [kW]
Modulation Index	$M = 1$
Fundamental Frequency	$f_0 = \omega_0 / 2\pi = 50$ [Hz]
DC – link Voltage	$V_{DC} = 500$ [V]
Arm Inductance	4.6 [mH]
Arm Resistance	$R = 0.05$ [Ω]
Plant Time Constant	$T_p = L / R = 0.092$ [s]
Carrier Frequency	$f_c = 10$ [kHz]
Sampling Frequency	$f_s = 2 f_c = 20$ [kHz]

Table 3 - Simulation Parameters

4.1. Series Repetitive Control (RC System 1)

The controller is constructed for suppression of the circulating current with a frequency of $2\omega_0 = 2\pi \cdot 100$ [r/s]. Under the assumption that the period of the disturbance is constant, the sampling frequency will be constant and set equal to twice the carrier frequency of the PWM carrier, giving the sampling time $\overline{T} = T_s = 1/2 f_c = 1/ f_s = 50$ [μs]. This gives $N_s = 200$. The use of $\overline{T}$ indicates the use of the constant sampling time[12] given. With the assumption of constant disturbance period T_p, following systems will be Linear Time Invariant (LTI).

The plant given in (129) and the nominal controller is discretized using ZOH. The plant transfer functions become

$$G_p\left(z,\overline{T}\right) = \frac{1}{2R}\left(\frac{1-e^{-\overline{T}/T_p}}{z-e^{-\overline{T}/T_p}}\right) = \frac{0.005433}{z-0.9995} \tag{143}$$

The internal model which will be used is

$$I(z) = \frac{Q(z)}{z^{N_s} - Q(z)} = \frac{0.25z^3 + 0.5z^2 + 0.25z}{z^{202} - 0.25z^3 - 0.5z^2 - 0.25z} \tag{144}$$

[12] The sampling time T_s can be adapted to changes in the disturbance period time T_p to maintain a constant value of N_s. This is termed *Varying Sampling Approach* and is a possible technique for addressing changes in T_p. See *Future Work* for further details and reading.

With the filter $Q(z) = 0.25z + 0.5 + 0.25z^{-1}$ used in further discussions. The plant $G_p\left(z,\overline{T}\right)$ and its inverse

$$\left(G_p\left(z,\overline{T}\right)\right)^{-1} = \frac{z - 0.9995}{0.005433} \tag{145}$$

are both casual and stable, thus the plant is minimum phase. This allows the filter $G_x\left(z,\overline{T}\right)$ to be implemented as

$$G_{x,1}\left(z,\overline{T}\right) = k_r\left(G_o\left(z,\overline{T}\right)\right)^{-1} \tag{146}$$

With the maximum – tuned PI controller gains, the discrete PI nominal controller transfer function becomes

$$G_c\left(z,\overline{T}\right) = \frac{zk_p + \left(\overline{T}k_i - k_p\right)}{z-1} = \frac{57.8z - 55.975}{z-1} \tag{147}$$

and

$$G_y\left(z,\overline{T}\right) = G_c\left(z,\overline{T}\right)G_p\left(z,\overline{T}\right) = \frac{0.314z - 0.3041}{z^2 - 1.999z + 0.9995} \tag{148}$$

With $G_c\left(z,\overline{T}\right)$, the closed loop system *without* the repetitive controller becomes

$$G_o\left(z,\overline{T}\right) = \frac{G_y\left(z,\overline{T}\right)}{1 + G_y\left(z,\overline{T}\right)} = \frac{0.314z - 0.3041}{z^2 - 1.685z + 0.6953} \tag{149}$$

The closed loop system $G_o\left(z,\overline{T}\right)$ has the roots $(z - 0.962942)(z - 0.722058)$, both laying inside the unit circle. Thus, the system is stable, fulfilling stability condition (109). The filter $G_{x,1}\left(z,\overline{T}\right)$ is thus chosen to be

$$G_{x,1}\left(z,\overline{T}\right) = k_r\left(G_o\left(z,\overline{T}\right)\right)^{-1} = k_r\frac{z^2 - 1.685z + 0.6953}{0.314z - 0.3041} \tag{150}$$

for this system. The repetitive controller transfer function becomes

$$G_{rc,1}\left(z,\overline{T}\right) = \frac{0.25z^5 + 0.07865z^4 - 0.4189z^3 - 0.07369z^2 + 0.1738z}{0.314z^{203} - 0.3041z^{202} - 0.07851z^4 - 0.08099z^3 + 0.07355z^2 + 0.076603z} \tag{151}$$

and the complete transfer function for the plug – in repetitive controller with PI control in series can be written as

$$G_{rc,c,1}\left(z,\overline{T}\right)=\left(1+I(z)G_{x,1}\left(z,\overline{T}\right)\right)G_c\left(z,\overline{T}\right)=\left(1+G_{rc,1}\left(z,\overline{T}\right)\right)G_c\left(z,\overline{T}\right)$$
$$=\left(1+k_r\frac{0.25z^5+0.07875z^4-0.4187z^3-0.0736z^2+0.1738z}{0.314z^{203}-0.3041z^{202}-0.0785z^4-0.08098z^3+0.07355z^2+0.07602z}\right)\left(\frac{57.8z-55.975}{z-1}\right)$$
(152)

which for $k_r=1$ results in

$$G_{rc,c,1}\left(z,\overline{T}\right)=\frac{\begin{array}{c}18.15z^{204}-35.16z^{203}+17.02z^{202}+14.45z^6-\\13.99z^5-28.9z^4+27.97z^3+14.45z^2-13.99z\end{array}}{\begin{array}{c}0.314z^{204}-0.6182z^{203}+0.3041z^{202}-0.07851z^5-\\0.002479z^4+0.1545z^3+0.002479z^2-0.07603z\end{array}} \tag{153}$$

The open – loop transfer function becomes

$$G_{ol}\left(z,\overline{T}\right)=\left(1+I\left(z\right)G_{x,1}\left(z,\overline{T}\right)\right)G_y\left(z,\overline{T}\right)=\left(G_y\left(z,\overline{T}\right)+k_r\cdot I\left(z\right)\left(1+G_y\left(z,\overline{T}\right)\right)\right) \tag{154}$$

which for $k_r=1$ results in

$$G_{ol}\left(z,\overline{T}\right)=\frac{\begin{array}{c}0.09862z^{204}-0.191z^{203}+0.09249z^{202}+0.07851z^6-\\0.07599z^5-0.157z^4+0.152z^3+0.07851z^2-0.07599z\end{array}}{\begin{array}{c}0.314z^{205}-0.932z^{204}+0.922z^{203}-0.304z^{202}-0.07851z^6+\\0.07599z^5+0.157z^4-0.152z^3-0.07851z^2+0.07599z\end{array}} \tag{155}$$

With the stability criterion (109) fulfilled, the criterion (110) demands

$$\left\|z^{-N_s}Q\left(z\right)\left(1-G_{x,1}\left(z,\overline{T}\right)G_o\left(z,\overline{T}\right)\right)\right\|_\infty<1$$
$$\left\|z^{-N_s}Q\left(z\right)\left(1-\left(k_r\frac{z^2-1.685z+0.6953}{0.314z-0.3041}\right)\left(\frac{0.314z-0.3041}{z^2-1.685z+0.6953}\right)\right)\right\|_\infty<1$$
$$\left\|z^{-N_s}Q\left(z\right)\left(1-k_r\right)\right\|_\infty<1$$

With $\left|Q\left(z\right)=1\right|$, this gives the poles

$$z=\sqrt[N_s]{\left|\left(1-k_r\right)\right|}e^{\frac{2k\pi j}{N_s}},\ k=0,...,N-1 \tag{156}$$

(156) shows that the poles are uniformly distributed around the unity circle with the radius $\sqrt[N_s]{\left|1-k_r\right|}$. Keeping the repetitive gain range within the limits of

$$\forall k_r\in\left[0,2\right] \tag{157}$$

ensures system stability. Figure 26 shows the pole placement for a repetitive gain of $k_r = 0.99$ and $k_r = 4.99$, where it seen how the poles migrate outside the unit circle for repetitive gains higher than 2. Furthermore, a low – pass, zero – phase FIR filter $Q(z)$ satisfying the condition (111) greatly reduce the impact of the higher frequency gain peaks. Figure 27 shows clearly how the poles of the closed loop transfer function migrate closer to origin as the frequency increase towards the Nyquist frequency $f_n = f_s / 2 = 0.5 / T_s$. In Figure 28, the open – loop transfer function of (154), the sensitivity function defined in (104) and the disturbance error function defined in (122) is shown.

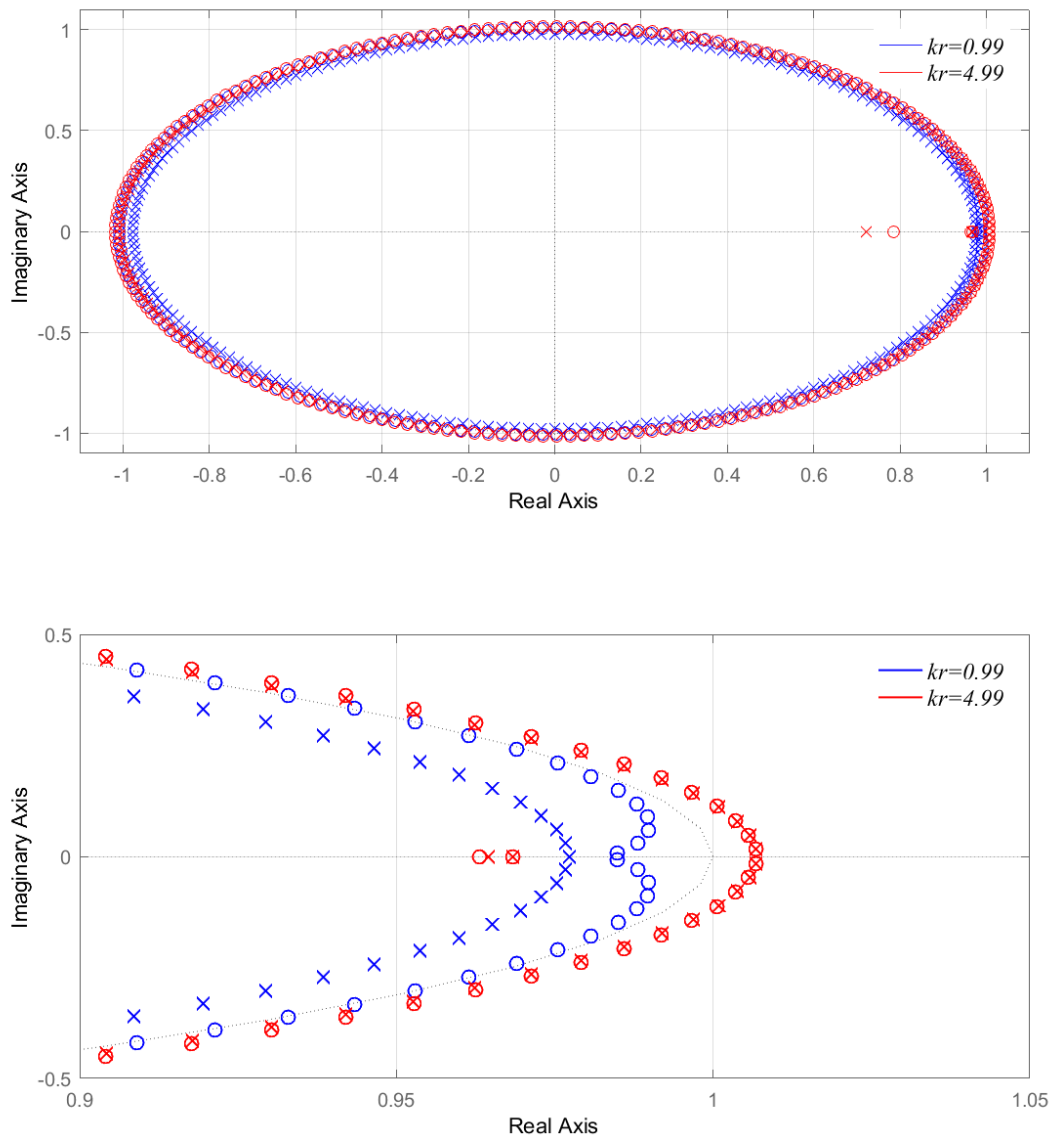
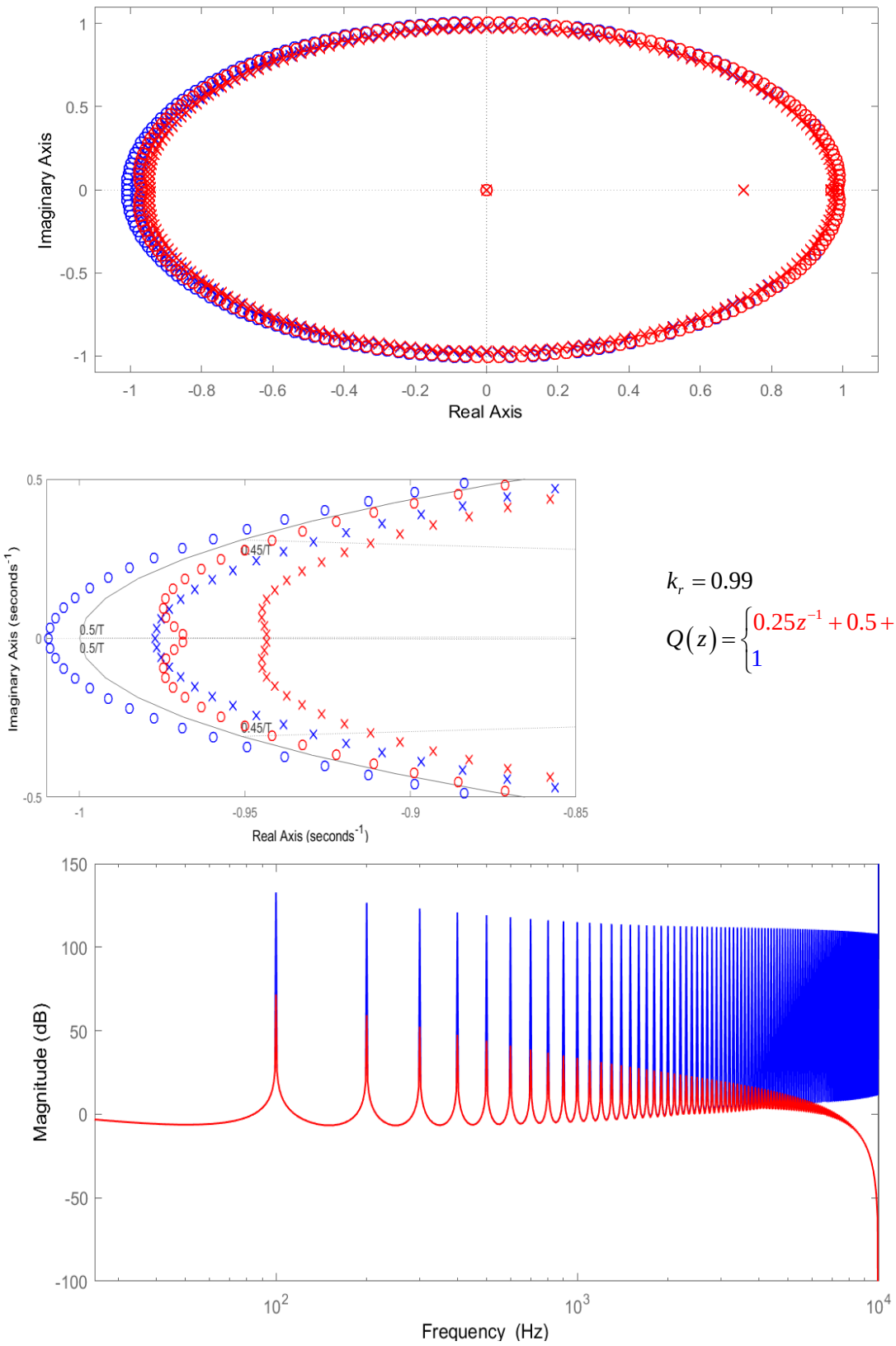


Figure 26 - Pole - zero map of closed loop transfer function showing pole migration for different values of kr (upper), zoomed (lower).



$$k_r = 0.99$$

$$Q(z) = \begin{cases} 0.25z^{-1} + 0.5 + 0.25z \\ 1 \end{cases}$$

Figure 27 – Pole – zero map of the closed loop transfer function (upper), zoomed pole – zero map of the closed loop transfer function (middle) and Bode plot of the repetitive controller transfer function Grc(z) for different settings of Q(z) (lower).

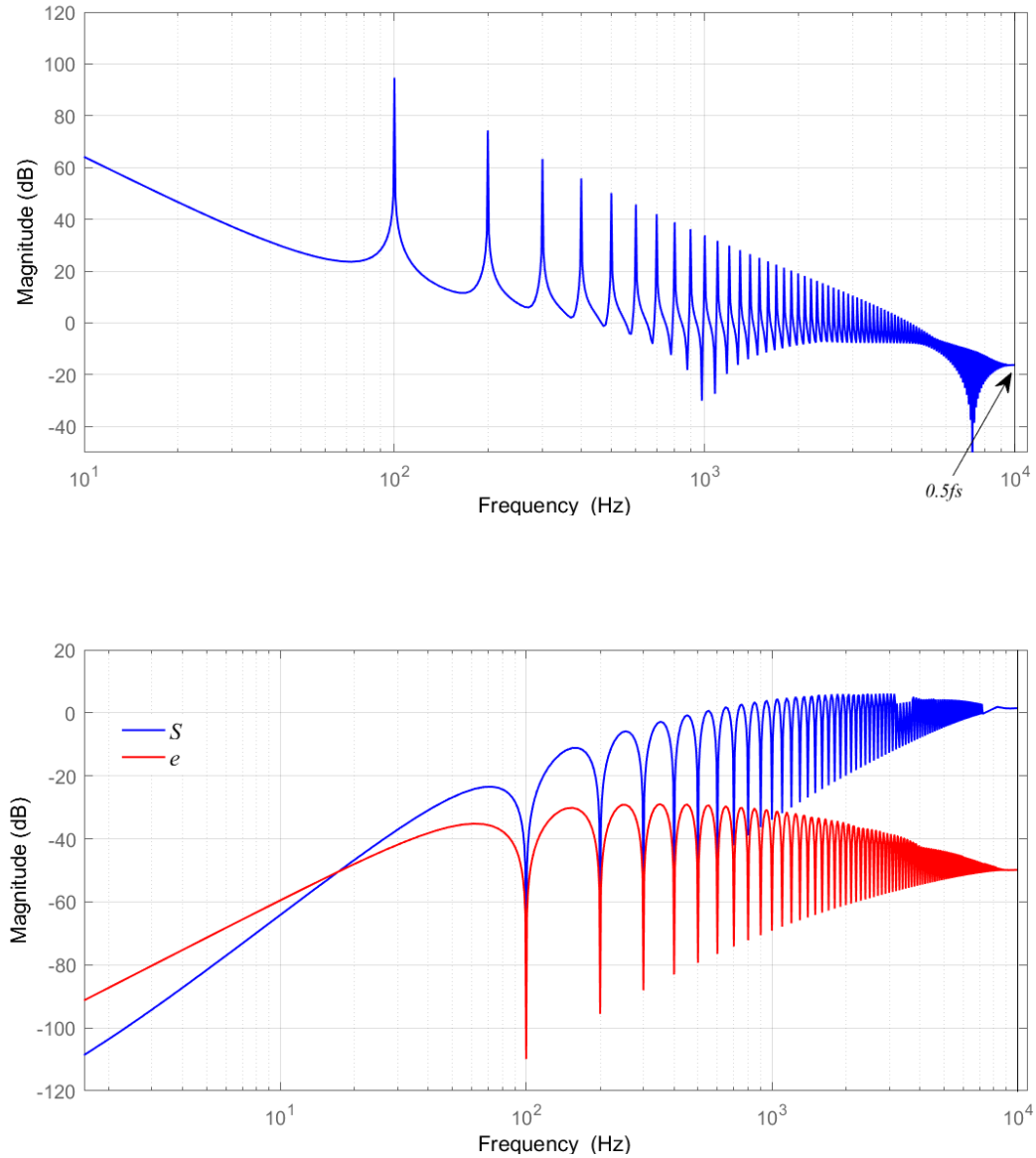


Figure 28 - Frequency response of the open – loop Gol(z) (upper), closed - loop sensitivity function S(z) and closed - loop disturbance error function e(z) (lower).

4.2. Parallel Repetitive Controller (System 2)

From the discussion in *Plug – In Repetitive Control with parallel PI control (RC System 2)*, the changes seen in System 2 with respect to System 1 is in the filter $G_x\left(z,\overline{T}\right)$. The nominal PI controller are no longer in series with the repetitive plug – in structure, but in parallel, thus a different frequency response is seen by the repetitive controller. To provide proper phase cancellation, the filter is now changed according to (121), giving

$$G_{x,2}\left(z,\overline{T}\right)=k_r\left(\frac{G_p(z)}{1+G_y(z)}\right)^{-1}=k_r\frac{z^2-1.685z+0.6953}{0.005433z-0.005433} \tag{158}$$

The complete transfer function of the plug – in repetitive controller with the PI control in parallel and $k_r=1$ now becomes

$$G_{rc,c,2}\left(z,\overline{T}\right)= \frac{\begin{gathered}0.314z^{204}-0.6182z^{203}+0.3041z^{202}-0.25z^6-\\0.2499z^5-0.5z^4+0.4997z^3+0.25z^2-0.2499z\end{gathered}}{0.005433z^{204}-0.01087z^{203}+0.005433z^{202}-0.001358z^5+0.002717z^3-0.001358z} \tag{159}$$

With the filter $G_{x,2}\left(z,\overline{T}\right)$, we get the same open – loop, closed – loop and error transfer functions as for RC System 1.

4.3. Proportional – Resonance Control

The PR controller is discretized with the methods described in *Proportional Resonant (*PR) Control. With the fixed simulation parameters of Table 3, the following tuned parameters are obtained assuming maximum allowable crossover frequency of $f_c = 1000$ [Hz]

Tuned Parameter
$k_{pr} = 57.8$
$T_h = 1.6 \cdot 10^{-3}$
$\alpha_h = 104.72$

Table 4 - Tuned PR Parameters

With $i = 6$ for the resonant bandwidth given by (86). These parameters give the following discretized transfer function

$$G_{pr}\left(z,\overline{T}\right) = \frac{58.71z^2 - 115.2z + 56.59}{z^2 - 1.994z + 0.9948} \tag{160}$$

The open – loop transfer function becomes

$$G_{ol}\left(z\right) = G_{pr}\left(z\right)G_p\left(z\right) = \frac{0.319z^2 - 0.6261z + 0.3075}{z^3 - 2.993z^2 + 2.987z - 0.9942} \tag{161}$$

And with the sensitivity and disturbance error function defined similarly as for the repetitive control structure, the sensitivity, error and open – loop frequency response for the PR control structure is shown in Figure 29.

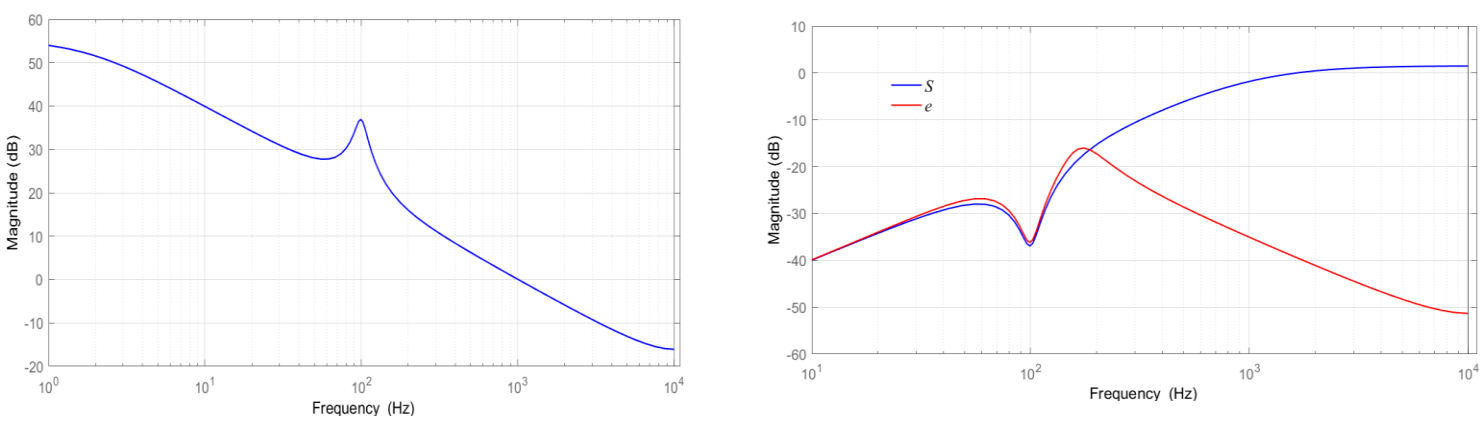


Figure 29 - Frequency response of open - loop transfer function (left) and closed - loop sensitivity function S and closed - loop disturbance error function e (right) for PR control tuned to 2ω.

This structure can be applied to several frequencies of interest. By adding several PR controllers, each tuned to the wanted frequency of suppression, in parallel, a similar frequency response as the RC can be obtained. The total transfer function of the PR controller tuned to multiples frequencies then becomes

$$G_{pr}(s)=\sum_{h=2}^{n}k_{pr,h}\left(1+\frac{s}{T_h\left(s^2+\alpha_h s+(h\omega_0)^2\right)}\right) \qquad (162)$$
$$h=2,4,6...$$

Where n is the max frequency component wanted for suppression. A controller tuned to $2\omega_0$ and $4\omega_0$ with the same parameters as in table Table 4 and $k_{pr,4}=k_{pr,2}/2$ gives the controller transfer function

$$G_{pr}\left(z,\overline{T}\right)=\frac{58.71z^2-115.2z+56.59}{z^2-1.994z+0.9948}+\frac{29.35z^2-57.54z+28.3}{z^2-1.991z+0.9948} \qquad (163)$$

And the open – loop transfer function becomes

$$G_{ol}\left(z,\overline{T}\right)=G_{pr}\left(z,\overline{T}\right)G_p\left(z,\overline{T}\right)=\frac{0.4784z^4-2.892z^3+2.807z^2-1.853z+0.4588}{z^5-4.984z^4+9.941z^3-9.919z^2+4.951z-0.989} \qquad (164)$$

Figure 30 shows the resulting open – loop frequency response of (164). The damped resonant peaks are clearly visible at 100 and 200 [Hz] $\left(2\omega_0 \text{ and } 4\omega_0 \text{ [r/s]}\right)$.

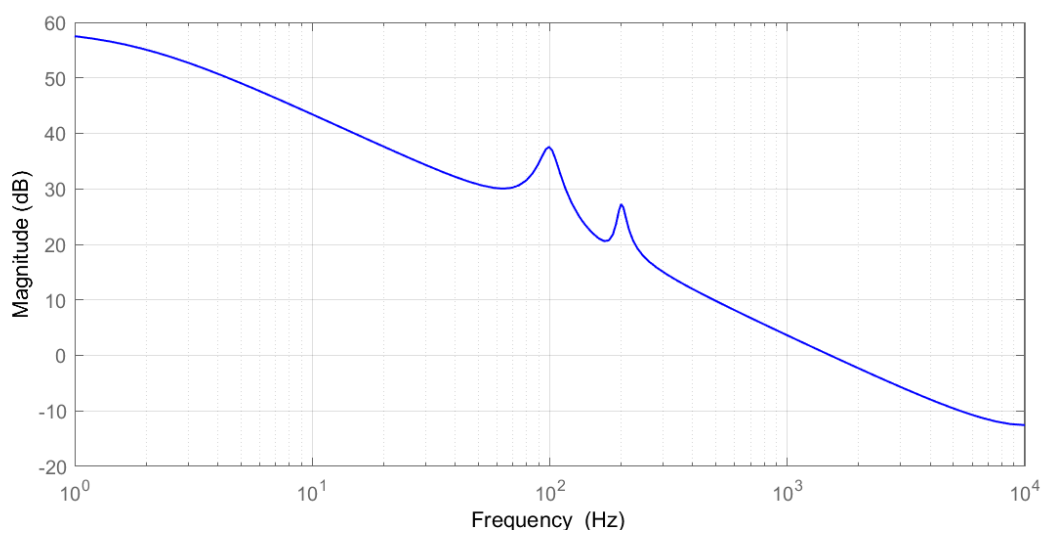


Figure 30 - Frequency response of open - loop transfer function with the controller tuned to 2ω and 4ω.

4.4. Simulation Results

The control structure discussed are simulated using MATLAB Simulink. A figure of the Simulink simulation used is shown in Figure 80 in *Appendix*.

4.4.1. Series Repetitive Control (RC System 1)

The RC system 1 described in 4.1 is simulated. Simulations shows the differential current defined by (3), where the PI controller, with varying values of k_p and k_i, and RC are enabled at a specified time instant as explained in the figure text. k_r is kept constant at 1.

- Maximum design values: $k_p = 57.8$ and $k_i = 36500$. $k_r = 1$.

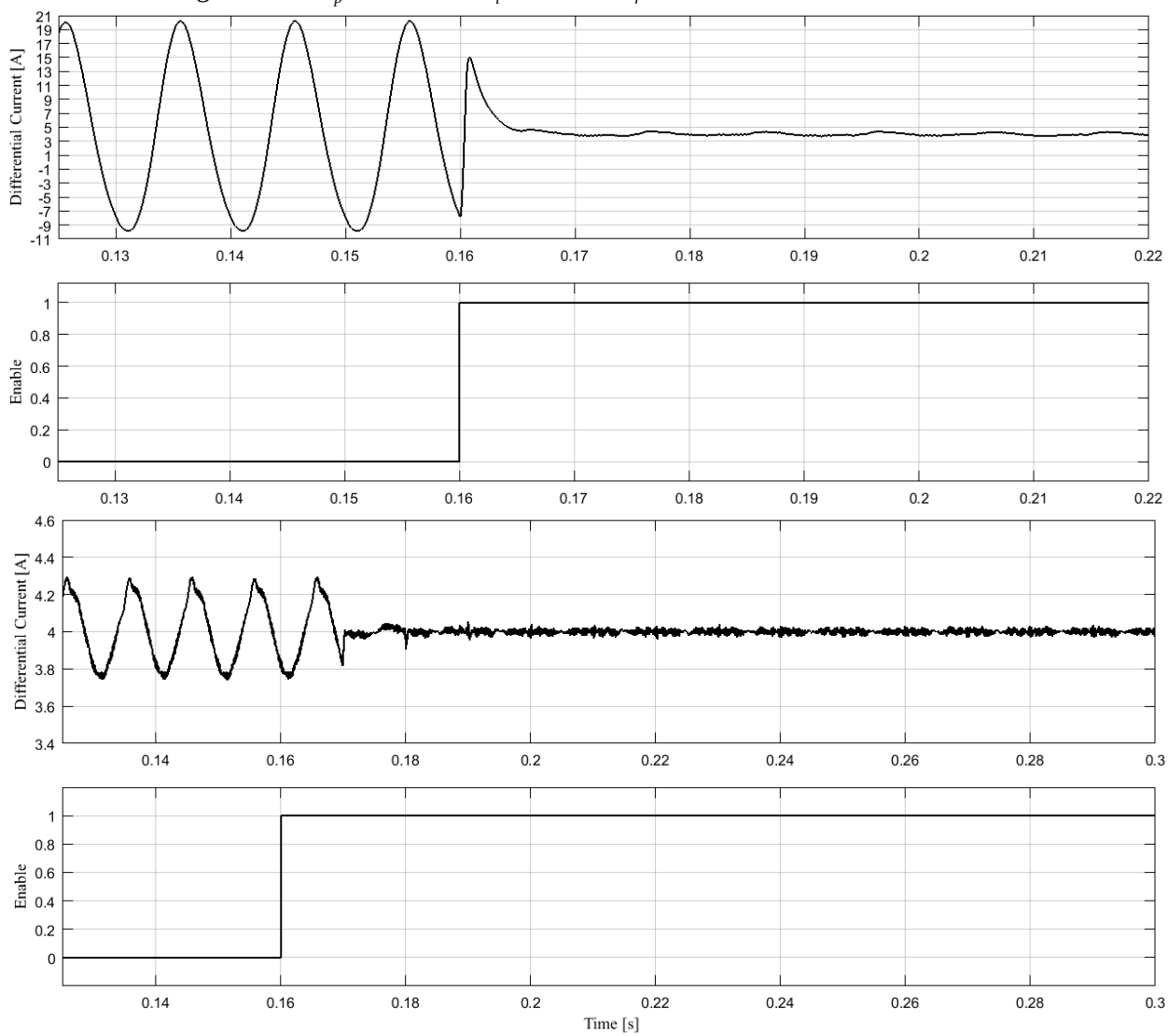


Figure 31 - Upper showing circulating current with a pure PI controller enabling at t=0.16 (s). Lower showing the circulating current with PI controller enabled and series RC enabling at t=0.16 (s).

- $k_p = 57.8$ and $k_i = 9125$. $k_r = 1$.

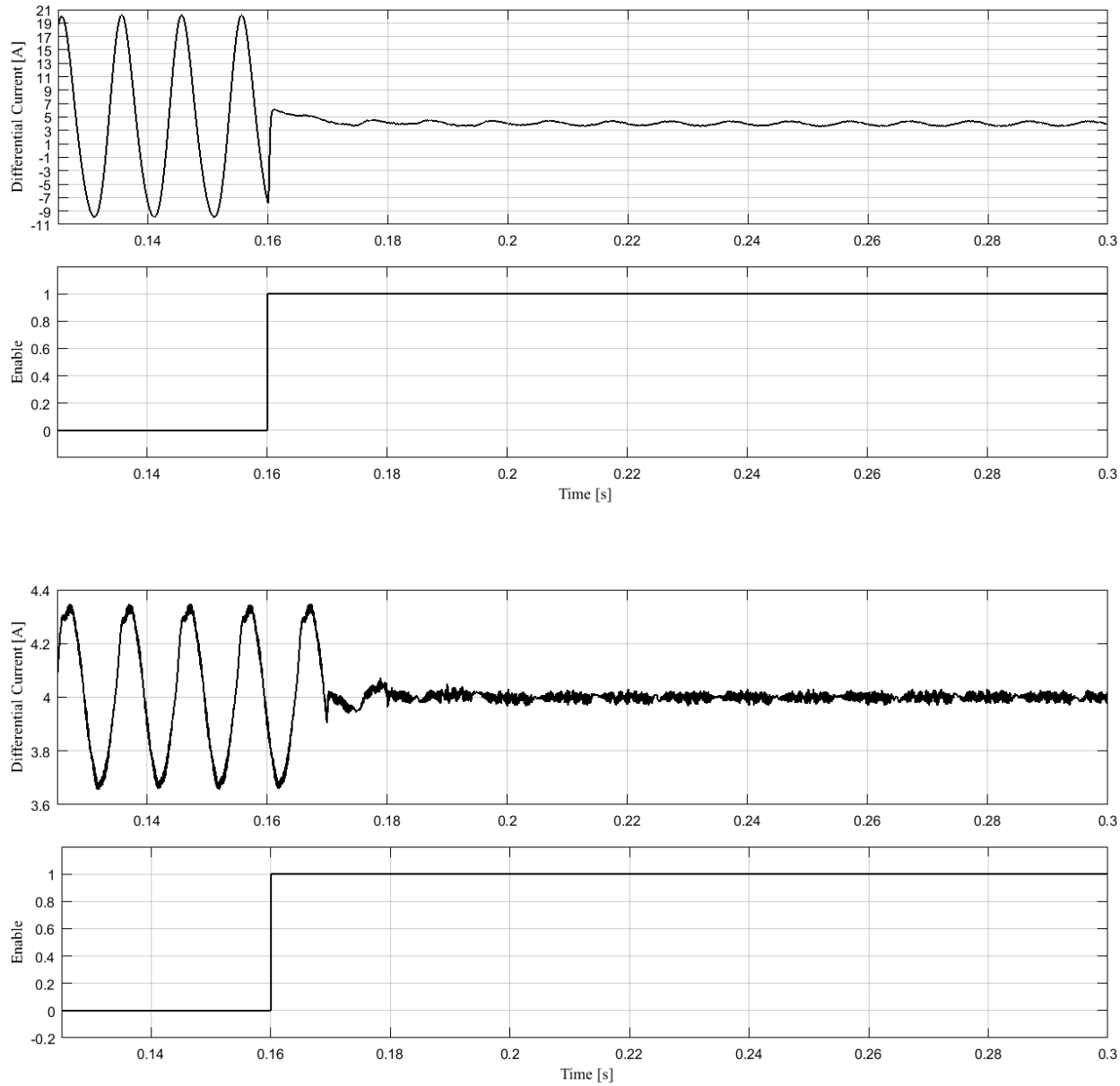


Figure 32 - Upper showing circulating current with a pure PI controller enabling at t=0.16 (s). Lower showing the circulating current with PI controller enabled and series RC enabling at t=0.16 (s).

- $k_p = 57.8$ and $k_i = 0$. $k_r = 1$.

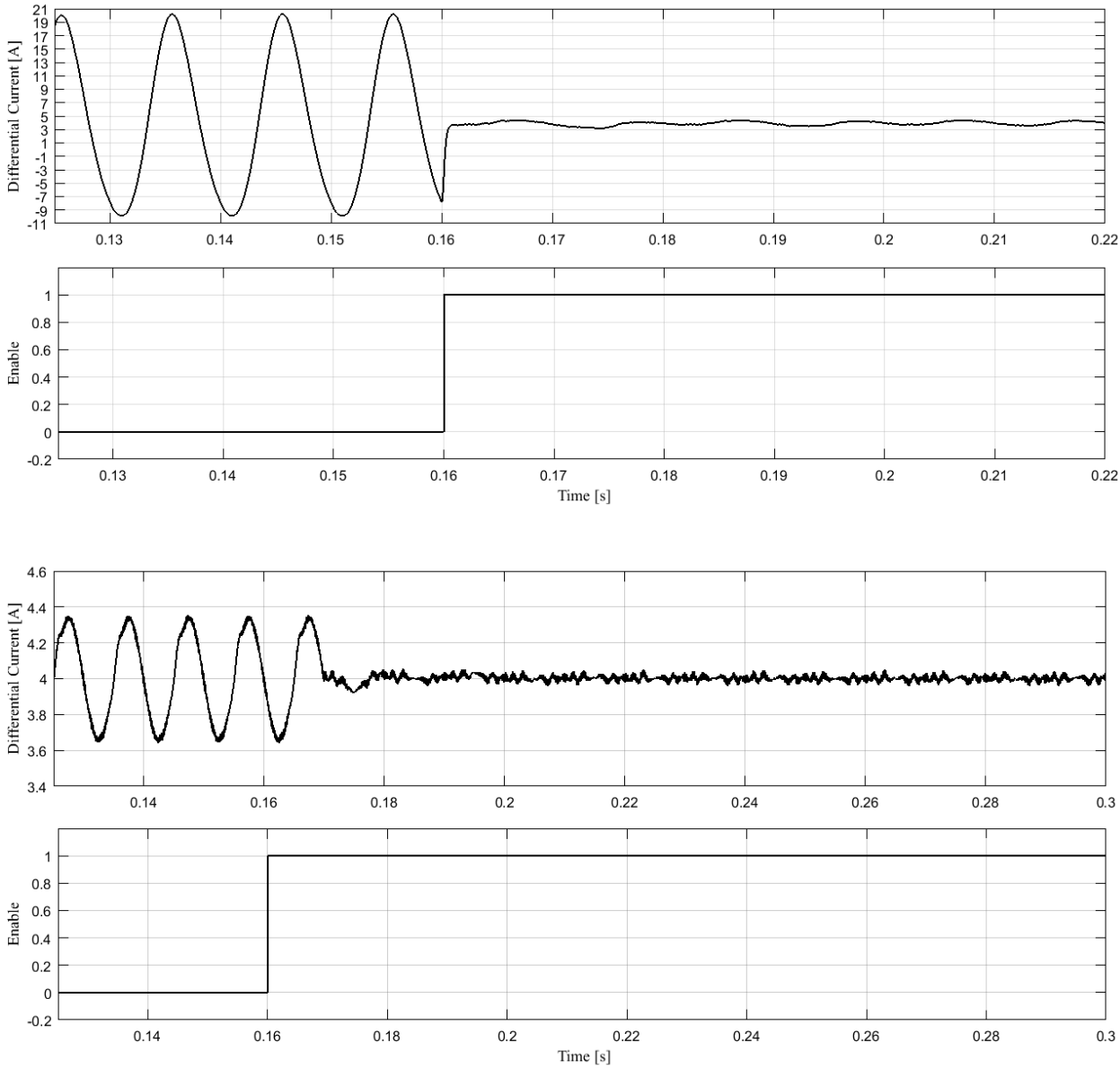


Figure 33 - Upper showing circulating current with a pure PI controller enabling at t=0.16 (s). Lower showing the circulating current with PI controller enabled and series RC enabling at t=0.16 (s).

- $k_p = 11.6$ and $k_i = 36500$. $k_r = 1$.

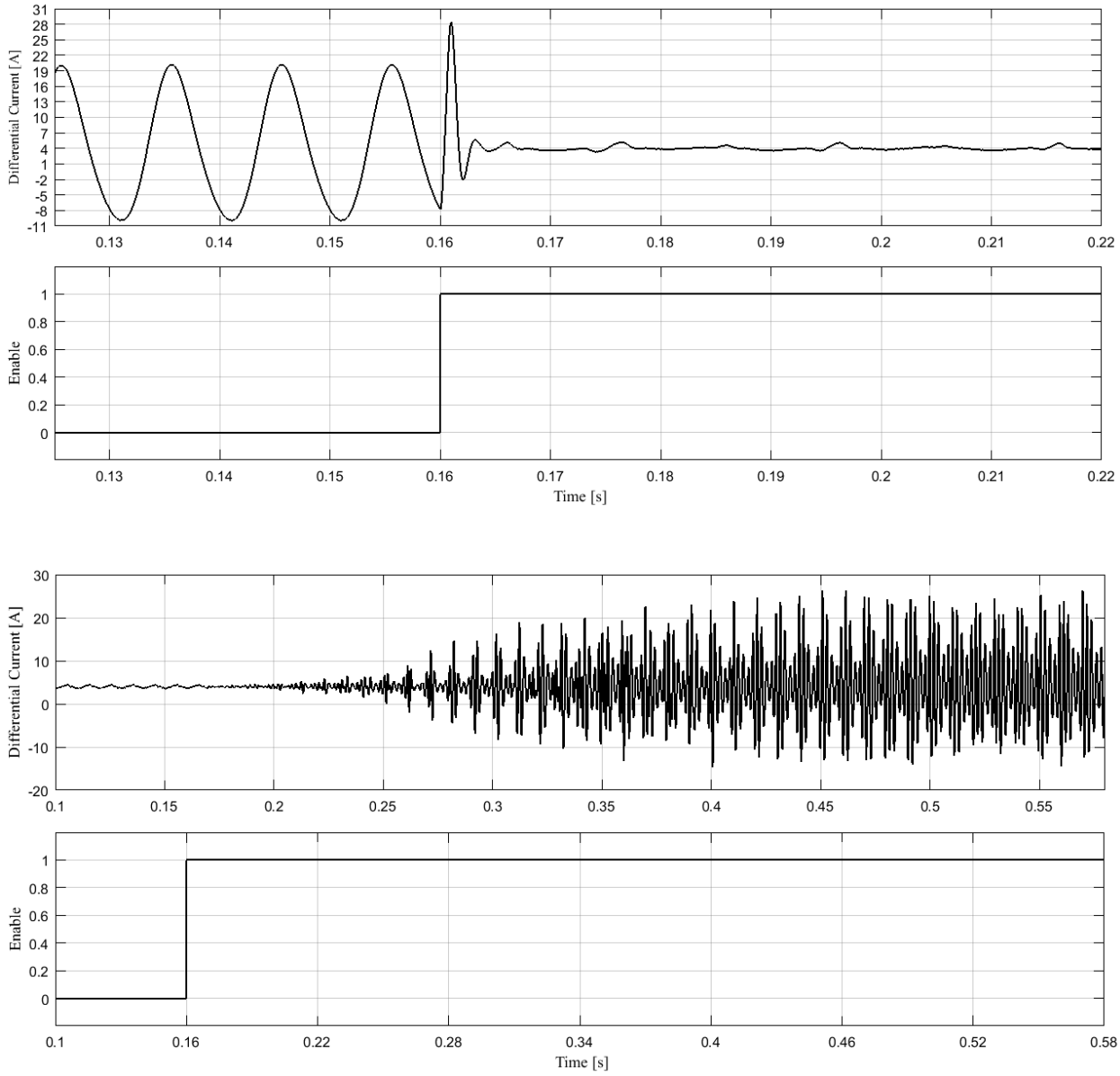


Figure 34 - Upper showing circulating current with a pure PI controller enabling at t=0.16 (s). Lower showing the circulating current with PI controller enabled and series RC enabling at t=0.16 (s).

- $k_p = 11.6$ and $k_i = 0$. $k_r = 1$.

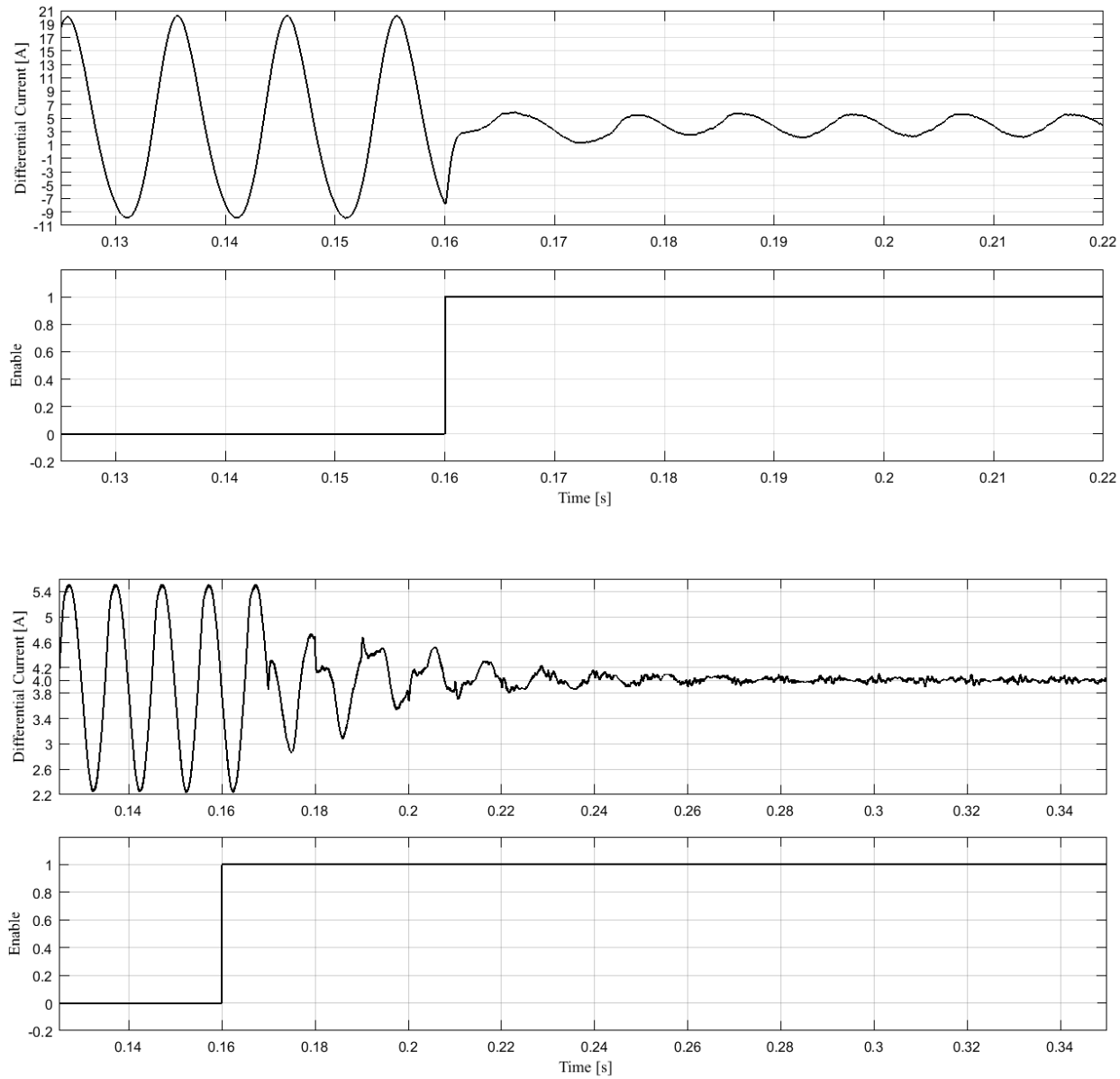


Figure 35 - Upper showing circulating current with a pure PI controller enabling at t=0.16 (s). Lower showing the circulating current with PI controller enabled and RC enabling at t=0.16 (s).

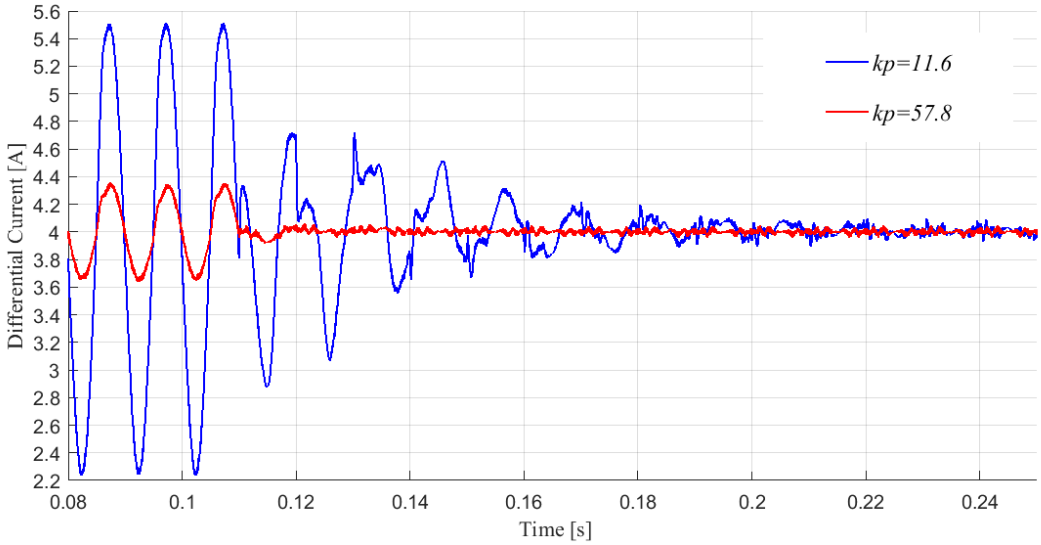


Figure 36 - Differential current settling with different values of proportional gain kp, with RC enabled at t=0.1.

4.4.2. Varying the Repetitive Gain

The repetitive gain k_r is varied to look at the response of the differential current.

- $k_p = 57.8$ and $k_i = 0$.

Time period [s]	< 0.4	0.4 < t < 0.6	0.6 < t < 1.1	>1.1
k_r [Ohm]	1	0.5	0.1	0.05

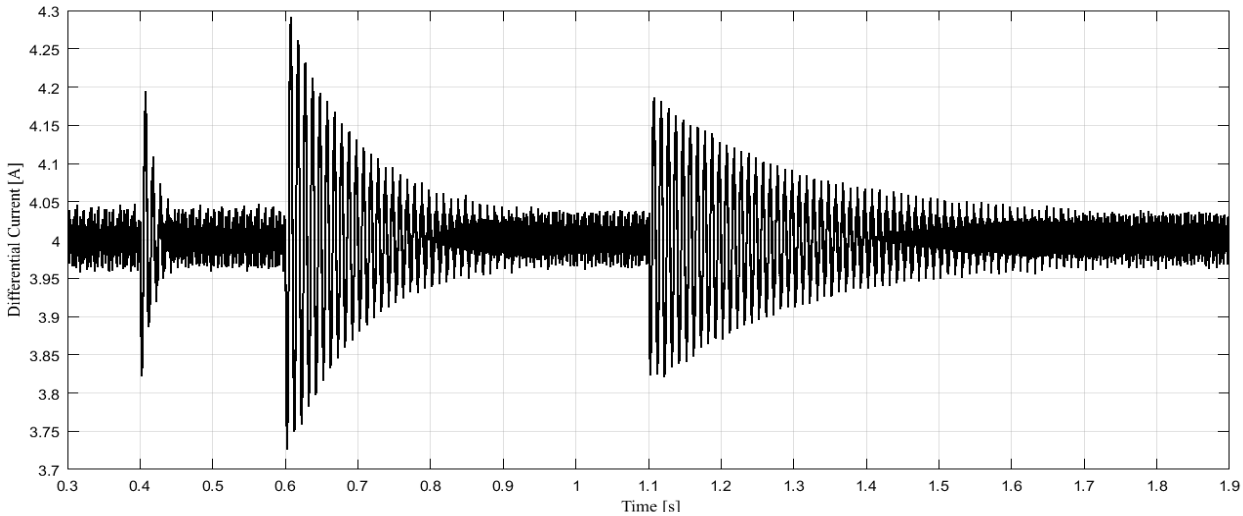


Figure 37 - Series RC with varying gain kr.

- $k_p = 11.6$ and $k_i = 0$.

Time period [s]	< 0.4	0.4 < t < 0.6	0.6 < t < 1.2	>1.2
k_r [Ohm]	1	0.5	0.1	0.05

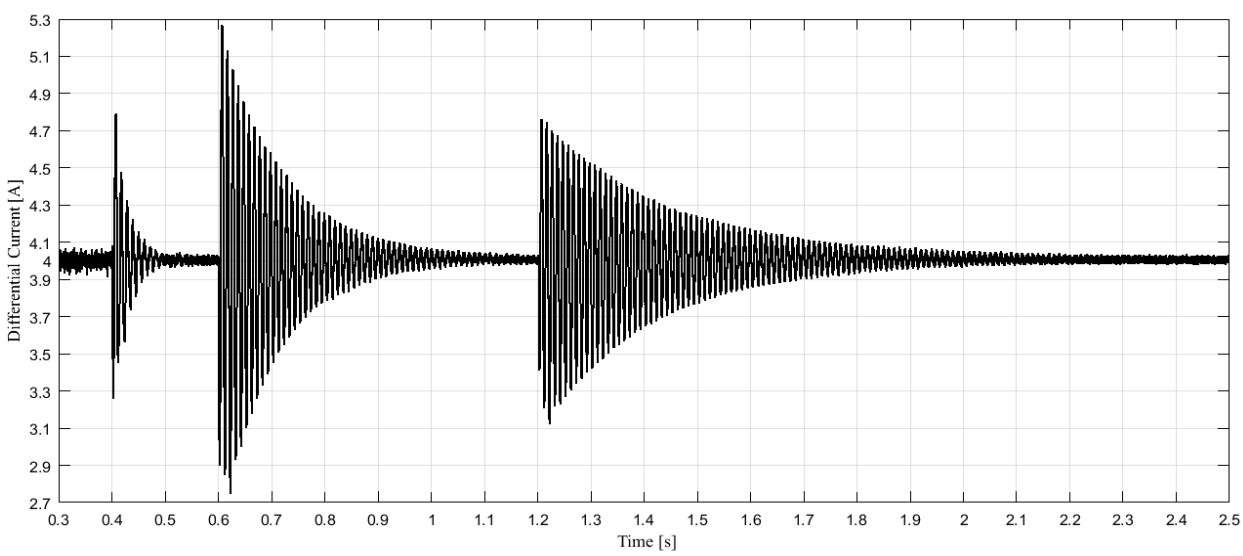


Figure 38 - Series RC with varying gain kr.

4.4.3. Steady – State RMS Error with varying Parameters

Different *series RC* configurations and parameter settings is investigated to look at the steady – state RMS error $|e|_{\text{RMS}} = |i_{diff}^{*} - i_{diff}|$. The error is filtered through a low – pass filter with a cut – off frequency of $f_{co} = 5000\ [\text{Hz}]$ to filter out unavoidable, high frequency harmonics due to PWM.

- $k_i = 0$

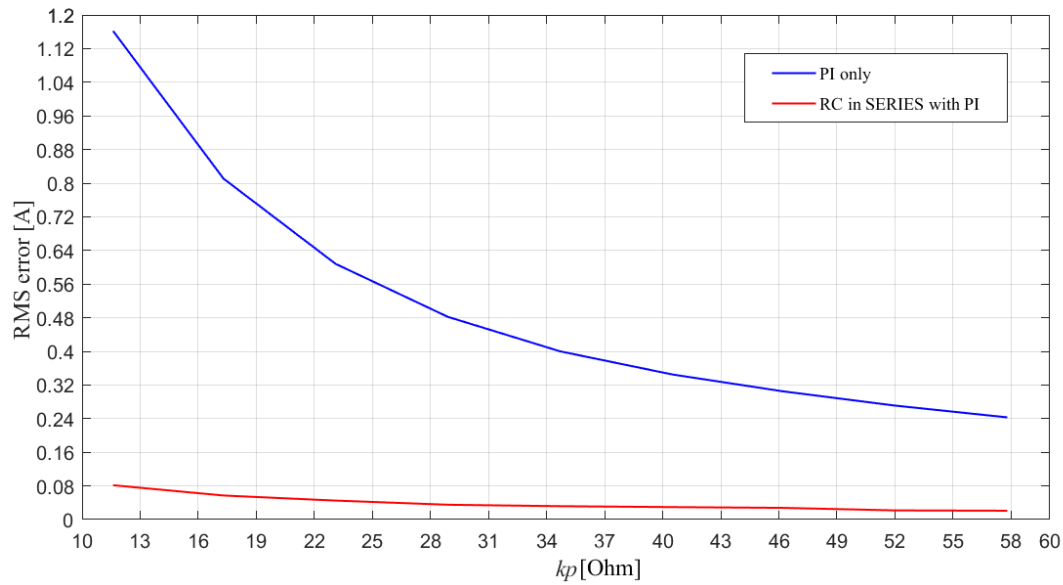


Figure 39 – RMS error for PI (ki=0) control only (blue) and RC System 1 (red) for varying gain.

- $k_i = 36500$

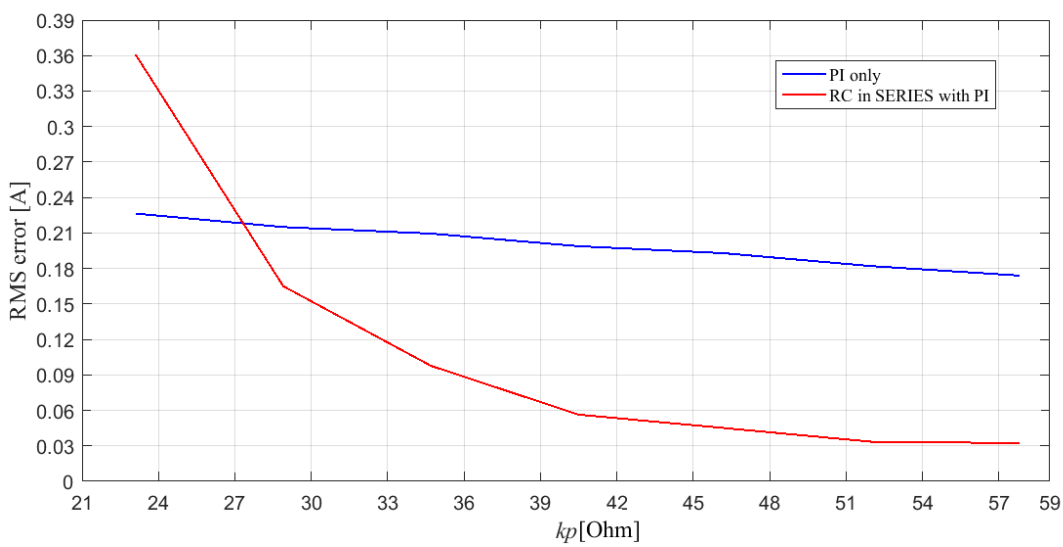


Figure 40 - RMS error for PI control only (blue) and RC System 1 (red) for varying gain.

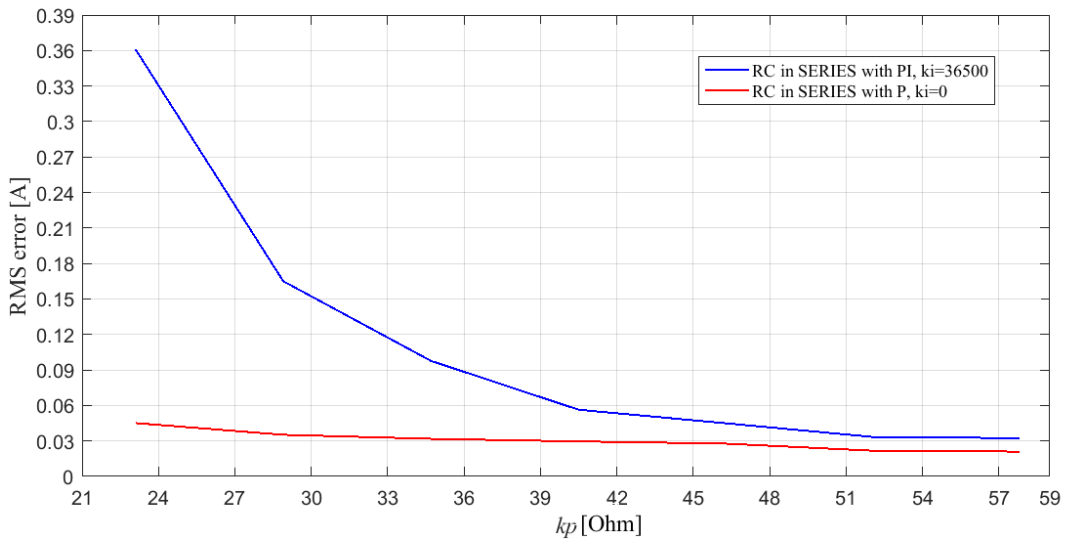


Figure 41 - RMS error vs. different values of kp, with ki=0 and ki=36500.

4.4.4. Series vs. Parallel Structure

Figure 42 shows how the two repetitive control structures cause the differential current to settle at approximately its DC – value, that is, almost completely suppress the frequency content of the differential current. Figure 43 shows the two control structures after the differential current has settled at its stationary value. $k_p = 11.6$ and $k_r = 1$.

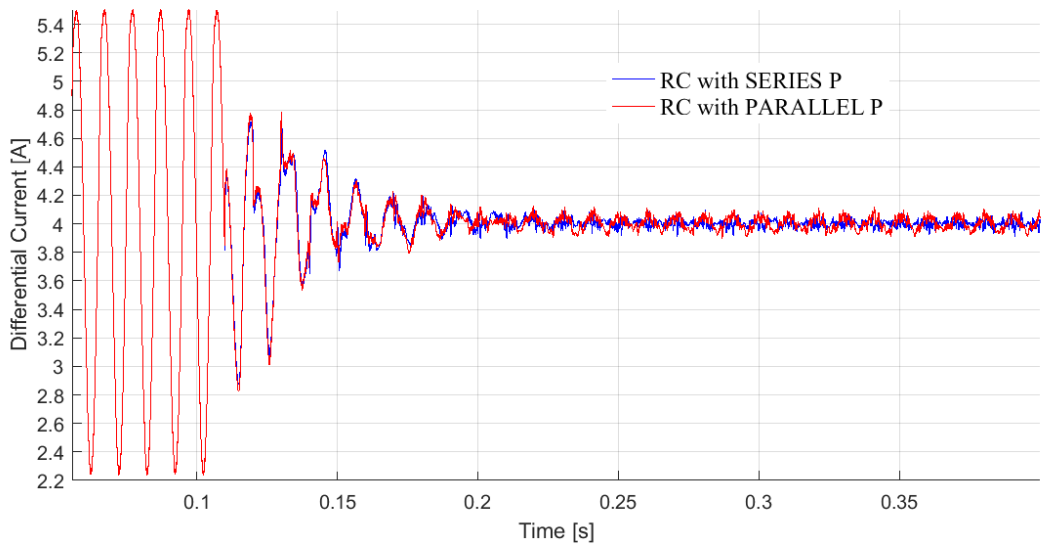


Figure 42 - Repetitive controller with nominal proportional controller in series (blue) and parallel (red) enabled at t=0.1 (s).

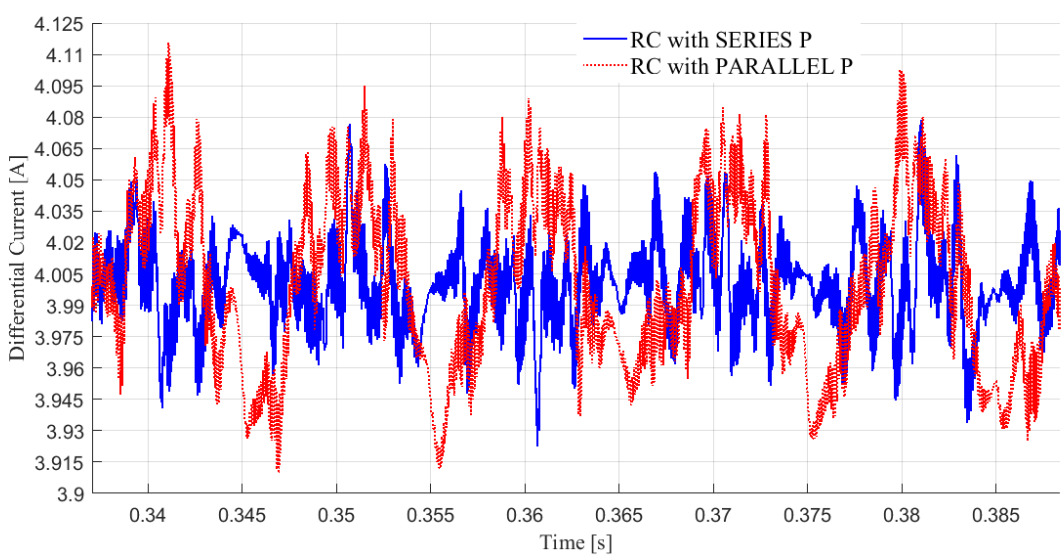


Figure 43 – Steady – state value of series (blue) and parallel (red) RC structure.

4.4.5. Repetitive Controller vs. Proportional Resonant Controller

Series RC with proportional nominal controller is compared with the PR controller tuned to $2\omega_0$. The proportional gain and PR controller gains are equally varied.

- $k_p = 57.8$ and $k_i = 0$. $k_r = 1$. $k_{pr,2} = 57.8$.

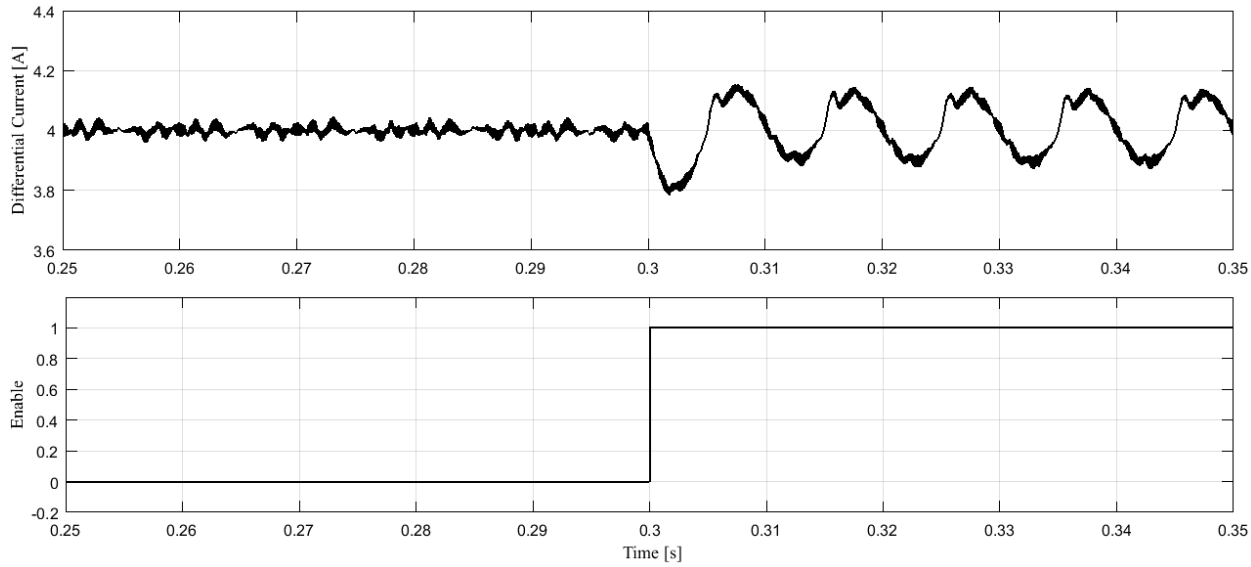


Figure 44 - RC enabled for t<0.3 (s) with PR enabling at t=0.3 (s).

- $k_p = 11.6$ and $k_i = 0$. $k_r = 1$. $k_{pr,2} = 11.6$.

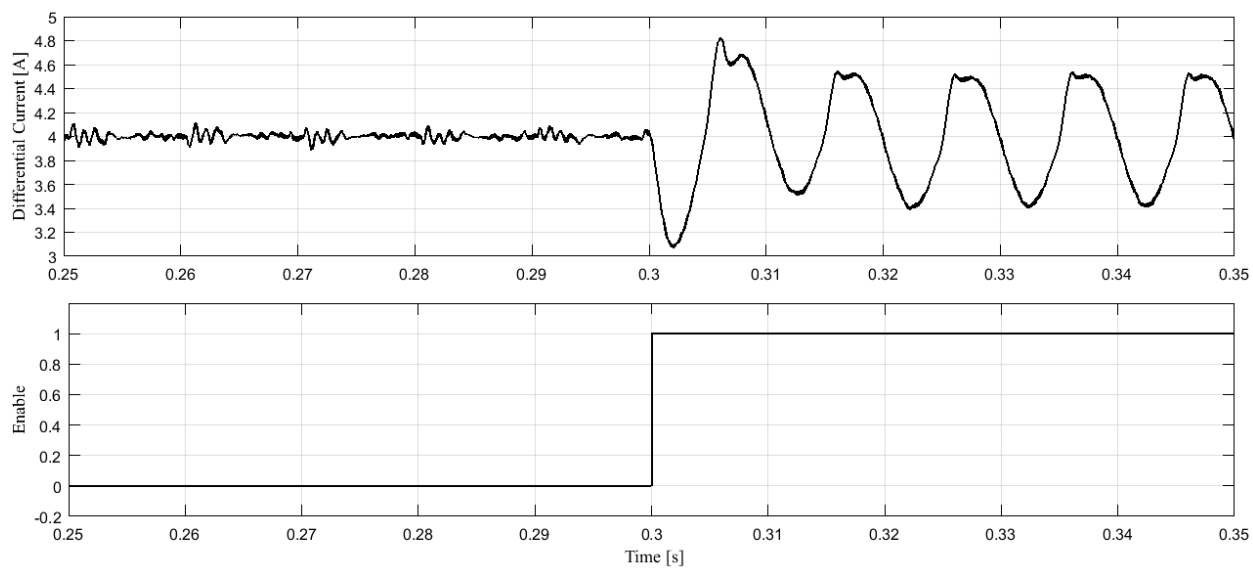


Figure 45 - RC enabled for t<0.3 (s) with PR enabling at t=0.3 (s).

Series RC with proportional nominal controller is compared with the PR controller tuned to $2\omega_0$ and $4\omega_0$. The proportional gain and PR controller gains are equally varied.

- $k_p = 57.8$ and $k_i = 0$. $k_r = 1$. $k_{pr,2} = 57.8$, $k_{pr,4} = 28.9$.

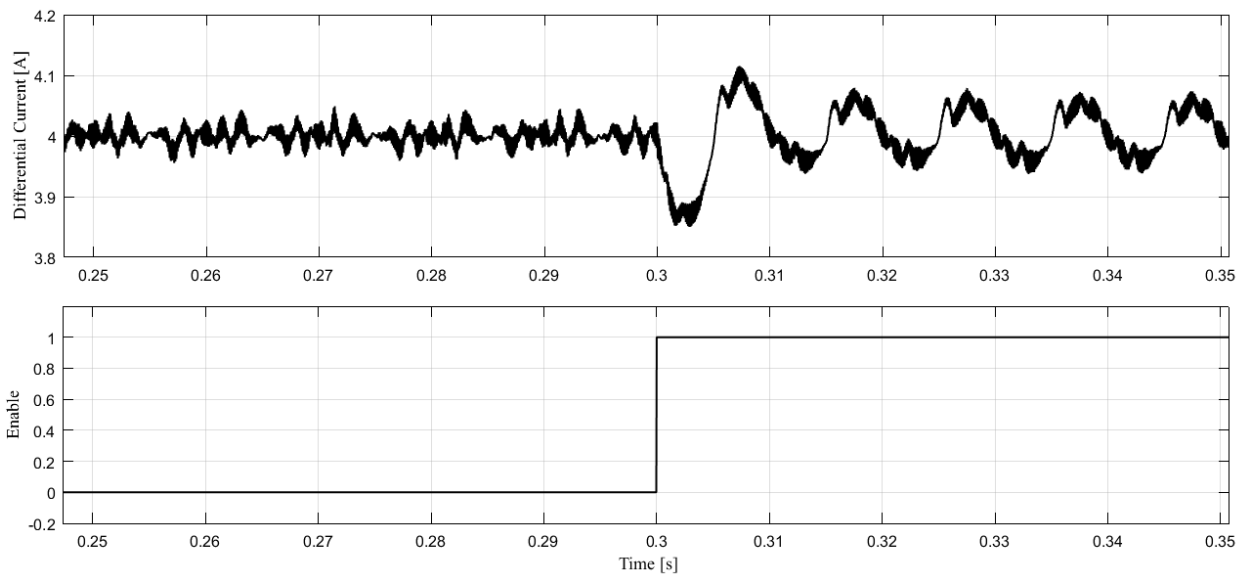


Figure 46 - RC enabled for t<0.3 (s) with PR enabling at t=0.3 (s).

- $k_p = 11.6$ and $k_i = 0$. $k_r = 1$. $k_{pr,2} = 11.6$, $k_{pr,4} = 5.8$.

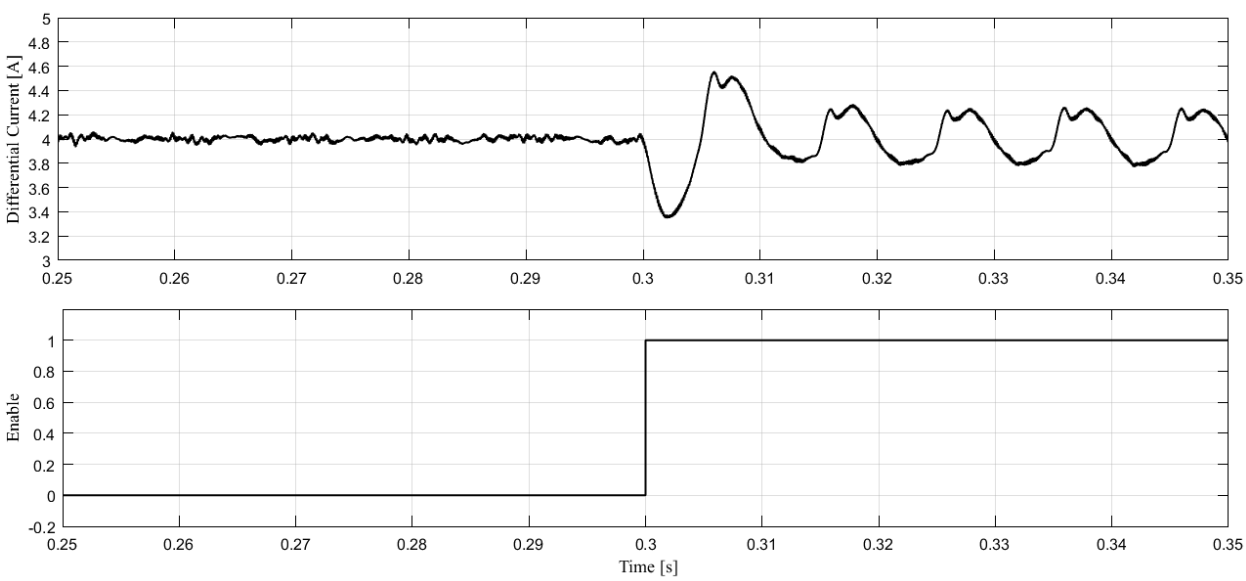


Figure 47 - RC enabled for t<0.3 (s) with PR enabling at t=0.3 (s).

Small differences in steady – state error is observed between highest bandwidth gain $k_p = 57.8$ and lowest bandwidth gain $k_p = 11.6$. However, both the nominal controller gain k_p and the repetitive controller gain k_r has great effect on the settling time of the differential current. Defining the settling time to be when the error $\left| i_{diff} - i_{diff}^* \right|$ falls within 2% of the steady state value of i_{diff}^*, gives the values for settling time in Table 5.

Settling time t_{set} [ms]	Nominal P – gain k_p	RC gain k_r
20 (2 period cycles)	57.8	1
25 (2.5 period cycles)	40.5	1
40 (4 period cycles)	28.9	1
100 (10 period cycles)	11.6	1

Table 5 - Settling time.

Table 5 includes the 10 ms. circulating current period delay introduced by z^{-N_s} required for the repetitive controller to act.

4.4.1 shows significant overshoot present when enabling the nominal PI controller for gain settings of $k_p = 11.6$ and k_i=36500.

4.4.2 shows overshoot present when the repetitive gain k_r is varied while the RC is enabled. A larger value of k_r and k_p shows shorter settling time. The overshoot is larger and settling time higher for lower values of the nominal controller proportional gain k_p.

Chapter 4.4.3 shows the repetitive controller with pure proportional nominal control yields lower steady – state error for a larger gain – range than an integral term added to the nominal controller.

Chapter 4.4.4 shows slightly lower steady – state error for *series* RC (system 1) compared to *parallel* RC (system 2). The settling time is equal.

Chapter 4.4.5 shows the series RC yielding lower steady – state error than both PR tuned to $2\omega_0$ and PR tuned to $2\omega_0$ and $4\omega_0$.

For a fast response with low steady – state error and no overshoot, an optimal control design should comprise of a pure high – gain proportional controller in series with the plug – in repetitive controller, termed *optimal series RC.* This intermittent conclusion is used in the further simulations.

4.4.6. Optimal Design of Series Repetitive Controller

This section show simulation for the optimal designed repetitive control, i.e. the *optimal series RC*, with the proportional gain tuned to maximum, $k_p = 57.8$ and k_r is set to 1.

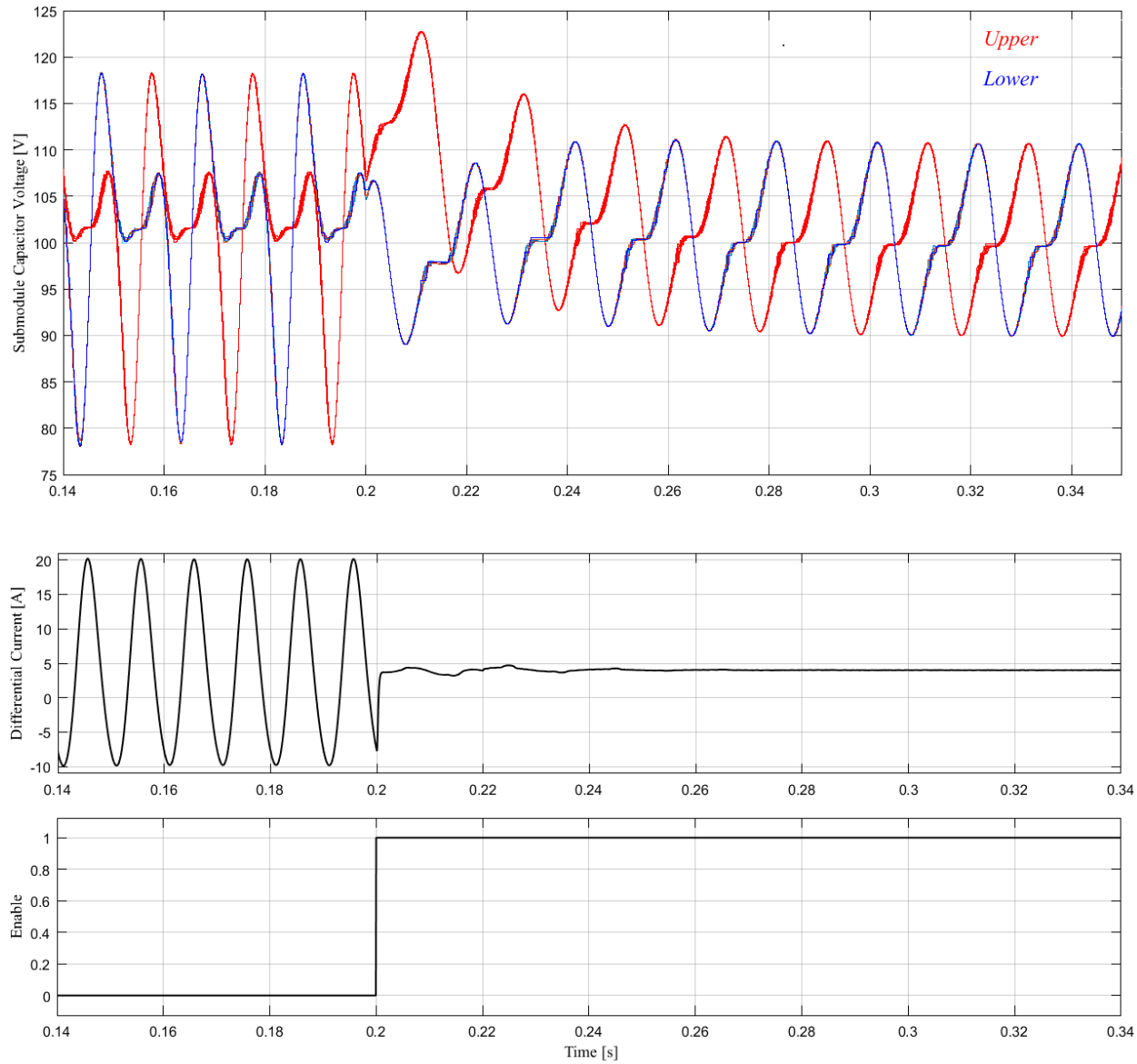


Figure 48 - Submodule capacitor voltages (upper) and circulating current (lower) with series RC enabled at t=0.2 (s).

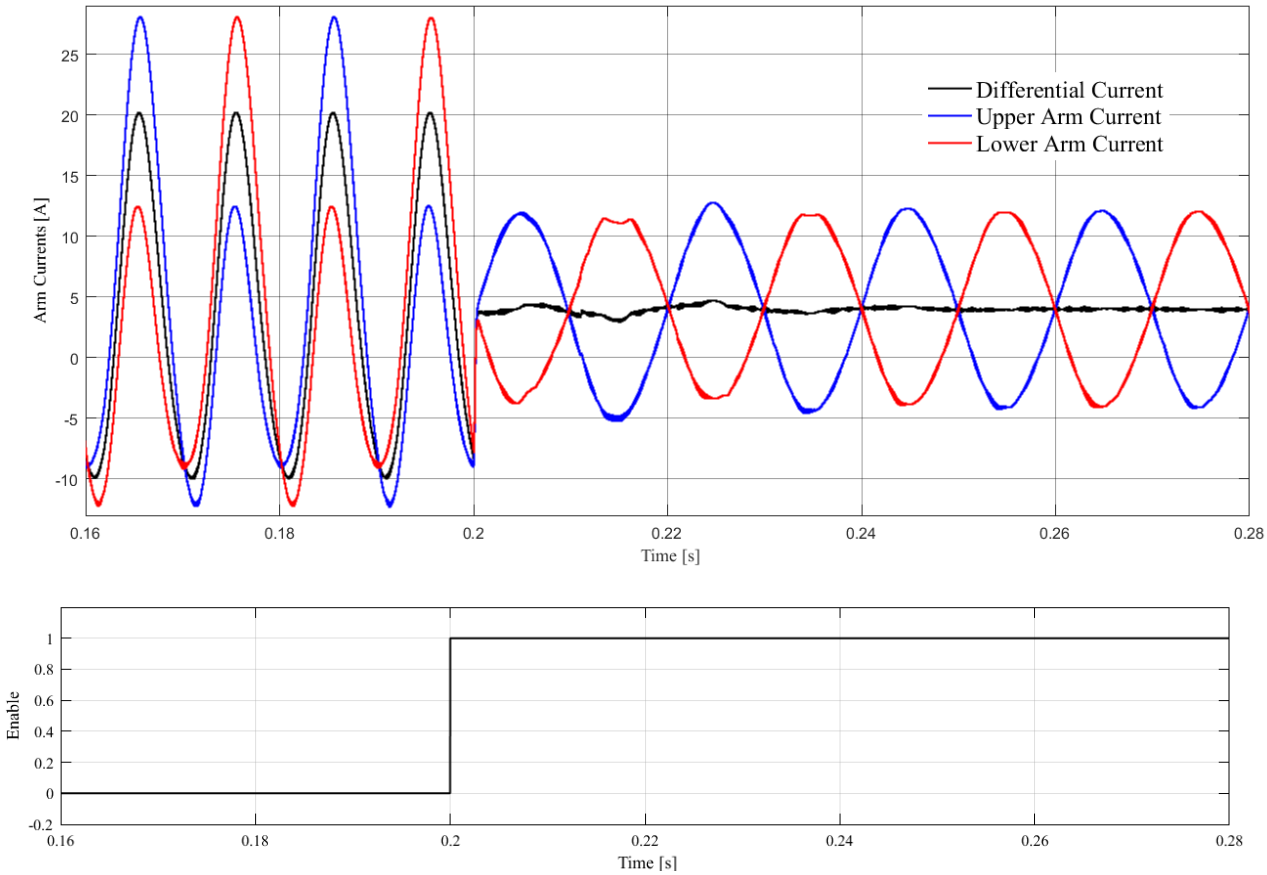


Figure 49 - Arm currents with optimal series RC enabled at 0.2 (s).

4.4.6.1. Output Data

The optimal output data for the simulation, i.e. the theoretically ideal parameter values, is based on the simulation parameters given in Table 3. These values are given in Table 6.

Parameter	Output phase voltage V_a [V]	Output phase current $\hat{I}_s$ [A]	Output phase power P [kW]
Optimal value	250	16	2

Table 6 - Optimal simulation values.

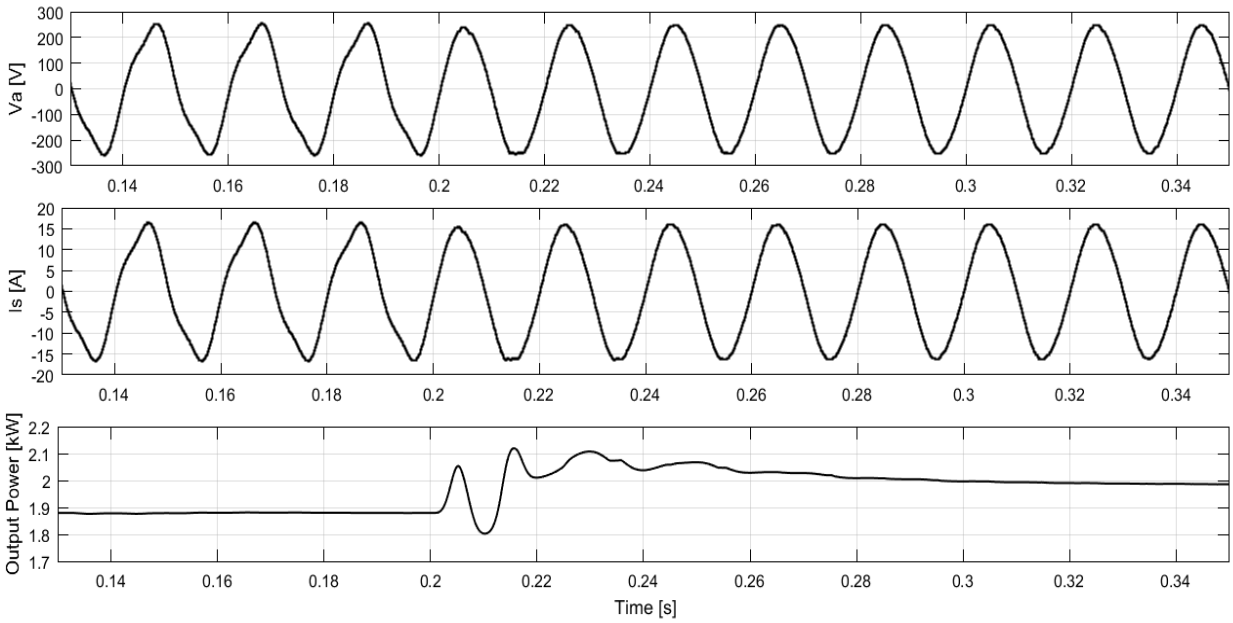


Figure 50 - Output phase voltage (upper), output phase current (middle) and output phase power (lower).

4.4.7. Frequency Content

The Powergui tool in Simulink is used to perform Fast Fourier Transform on the waveform data of interest.

4.4.7.1. Circulating Current

Figure 51 shows the frequency spectrum of the *circulating current* i_{circ} of eq. (3) with circulating current control disabled. Figure 52 shows the frequency spectrum with the *optimal series RC* enabled.

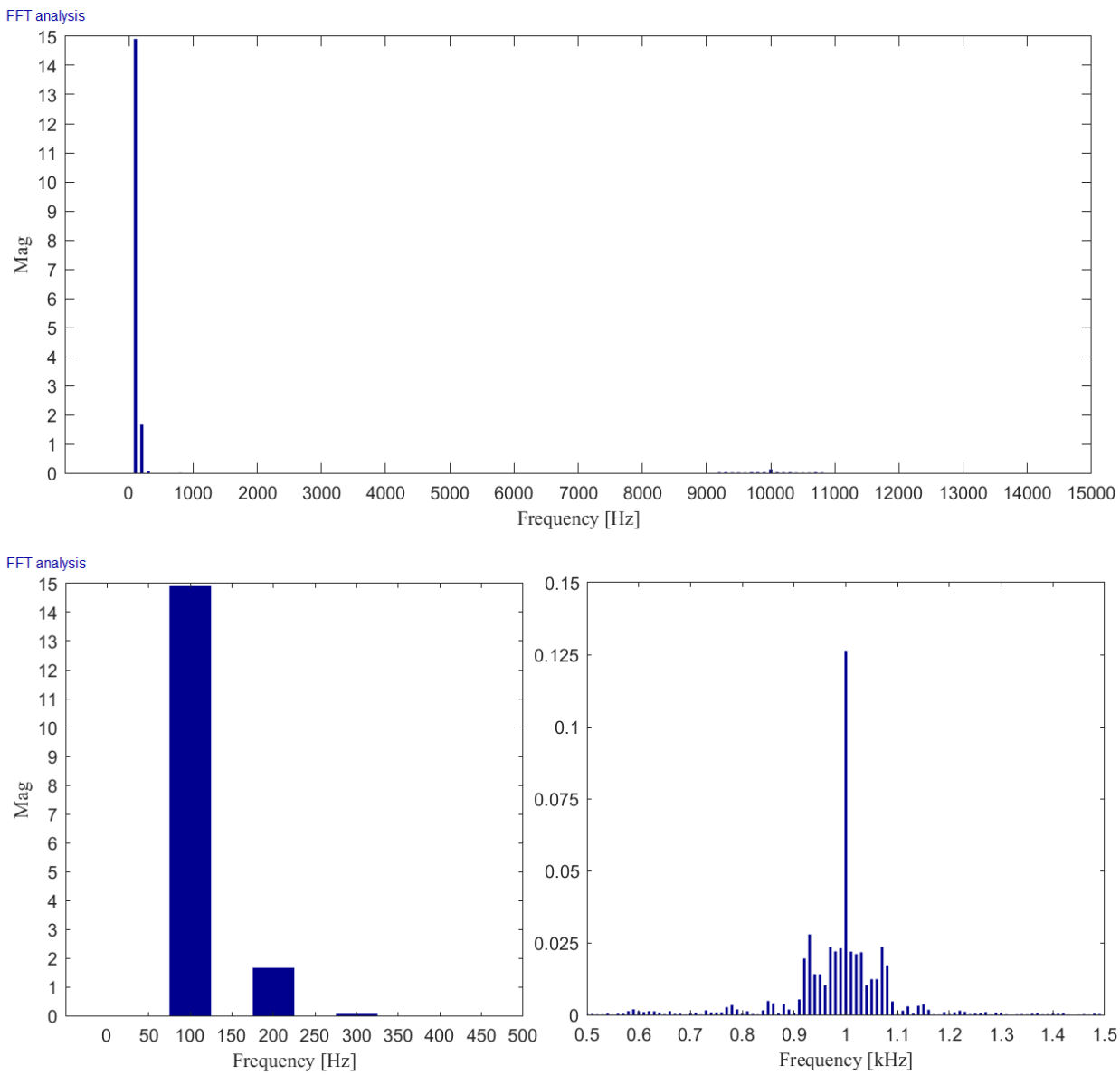


Figure 51 - FFT of the circulating current waveform at different frequency ranges with circulating current control DISABLED.

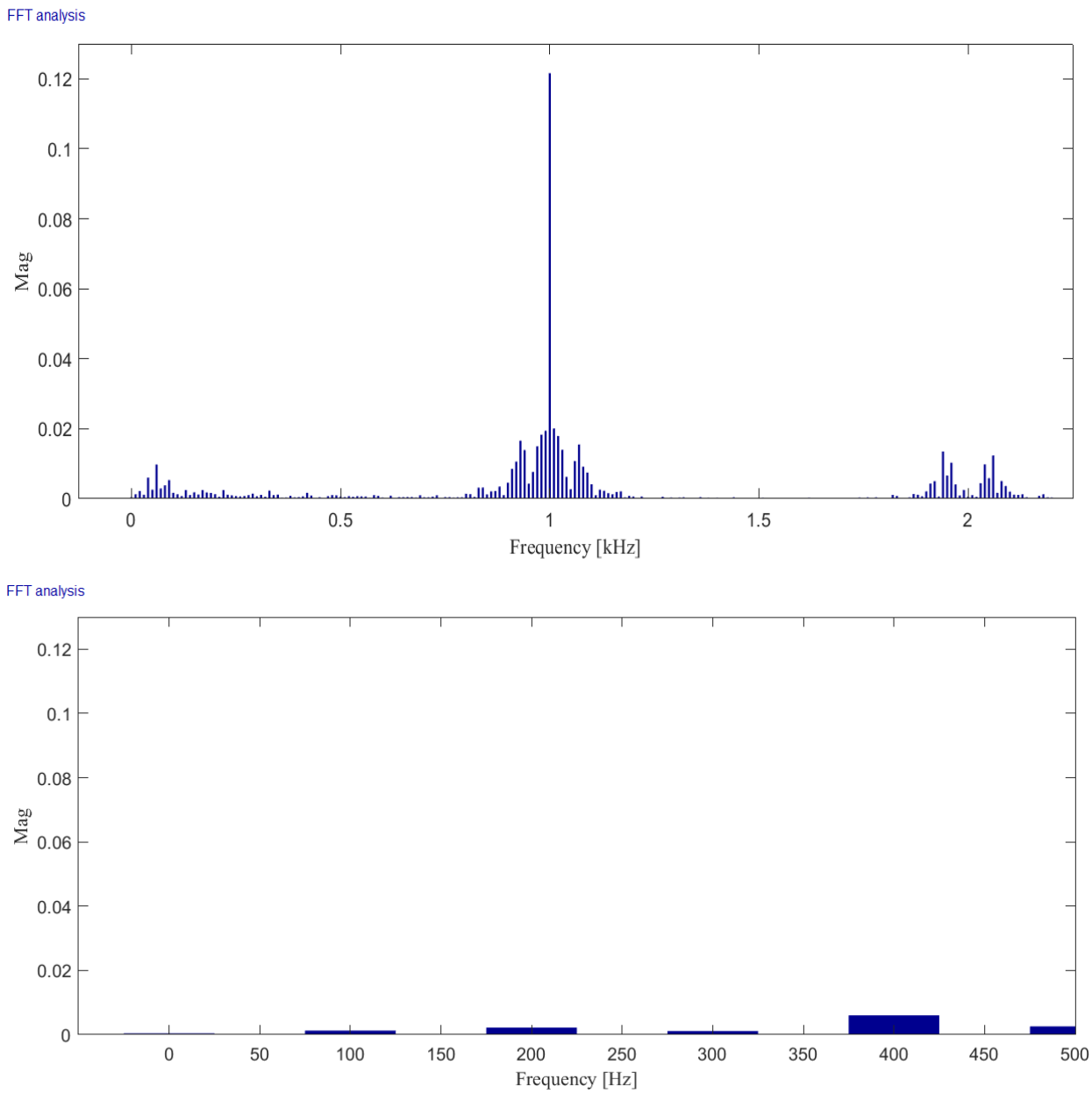


Figure 52 - FFT of the circulating current waveform at different frequency ranges with optimal series RC ENABLED.

4.4.7.2. Submodule Capacitor Voltages

Figure 53 shows the frequency spectrum of the upper and lower capacitor voltages excluding the DC – component $\overline{v_{c,u,l}^{\Sigma}} / N = V_{DC} / N$ as defined in (15). Figure 54 shows the spectrum of the sum – capacitor voltages of the leg excluding the DC – component $2V_{DC}$ and Figure 55 shows the differential – capacitor voltages of the leg, both defined by (36).

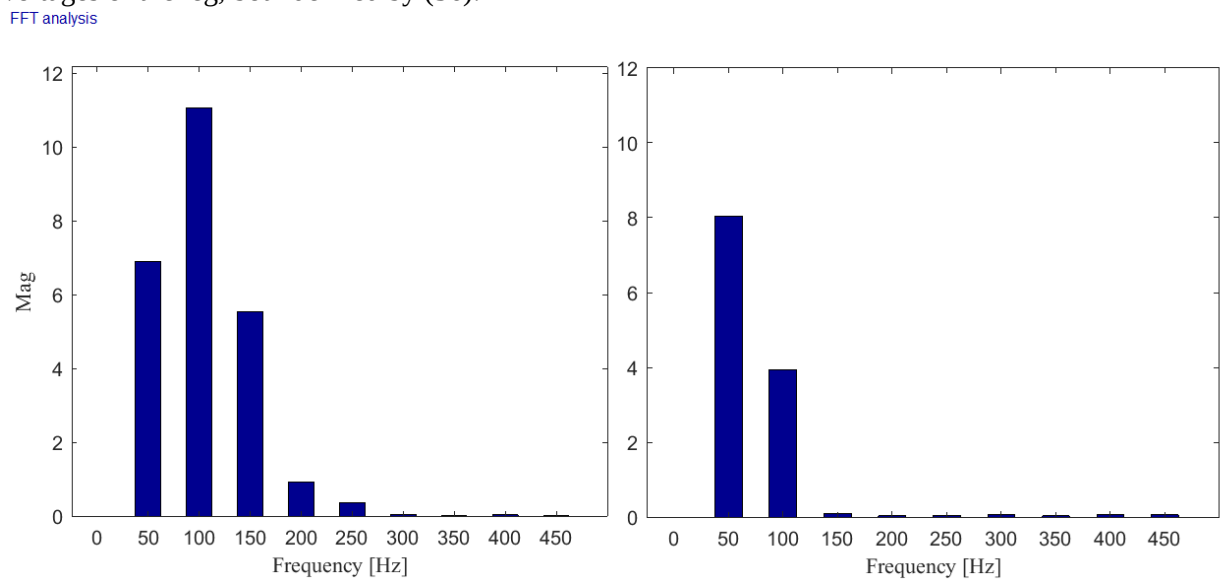


Figure 53 – Frequency spectrum of the upper and lower arm voltages with circulating current control DISABLED (left) and optimal series RC ENABLED (right).

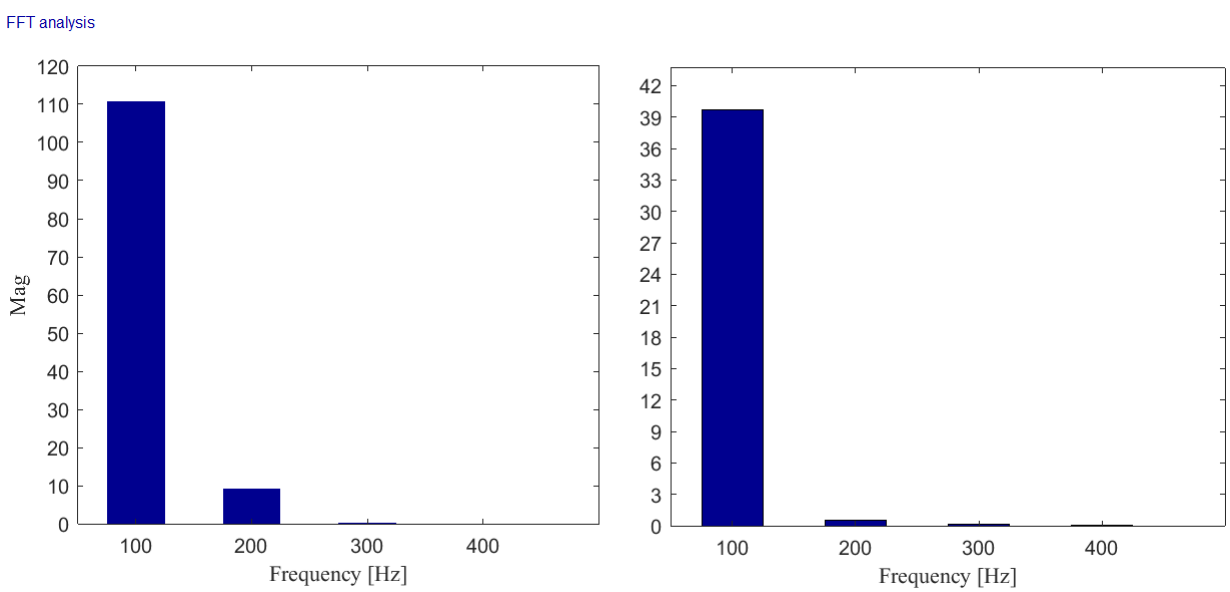


Figure 54 - Frequency spectrum of the sum – capacitor voltage with circulating current control DISABLED (left) and optimal series RC ENABLED (right).

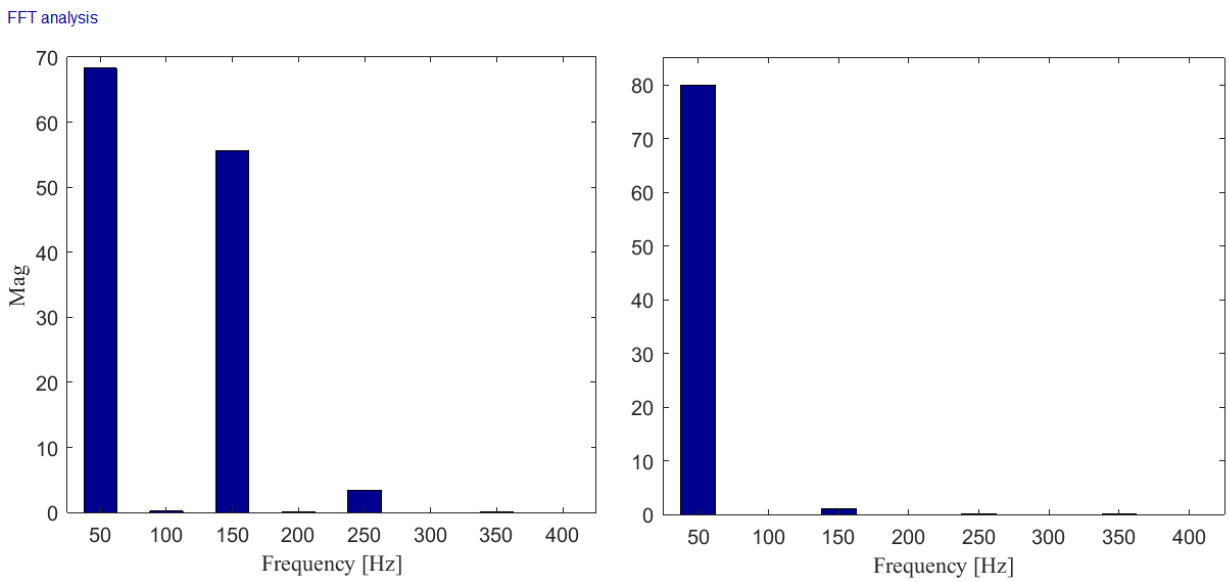


Figure 55 - Frequency spectrum of the differential – capacitor voltage with circulating current control DISABLED (left) and optimal series RC ENABLED (right).

4.4.7.3. Output Data

Figure 56 shows the frequency spectrum of the output phase voltage with no circulating current control enabled. Figure 57 shows the frequency spectrum with the *optimal series RC* enabled. The THD[13] and fundamental frequency components of the output phase voltage $v_a(t)$ and current $i_s(t)$ of (8) with and without optimal repetitive control of the circulating current is given in Table 7.

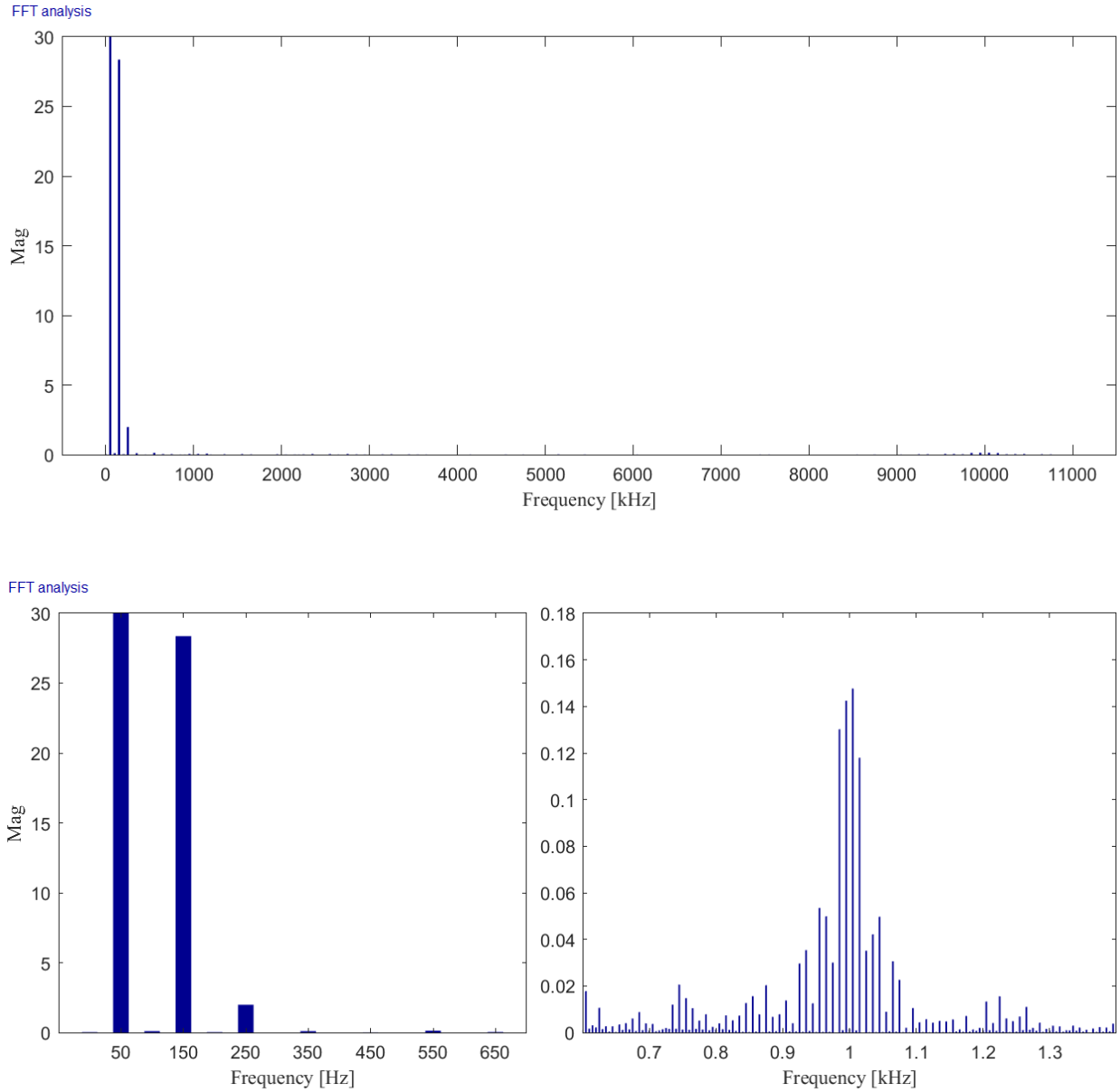


Figure 56 - FFT of the output phase voltage waveform at different frequency ranges with circulating current control DISABLED.

[13] THD – Total Harmonic Distortion – Defined as the ratio of the RMS amplitude of the higher order frequency components to the RMS amplitude of the fundamental frequency component as

$$\text{THD}=\sqrt{\frac{V_2^2+V_3^2+V_4^2+\ldots}{V_1^2}}$$

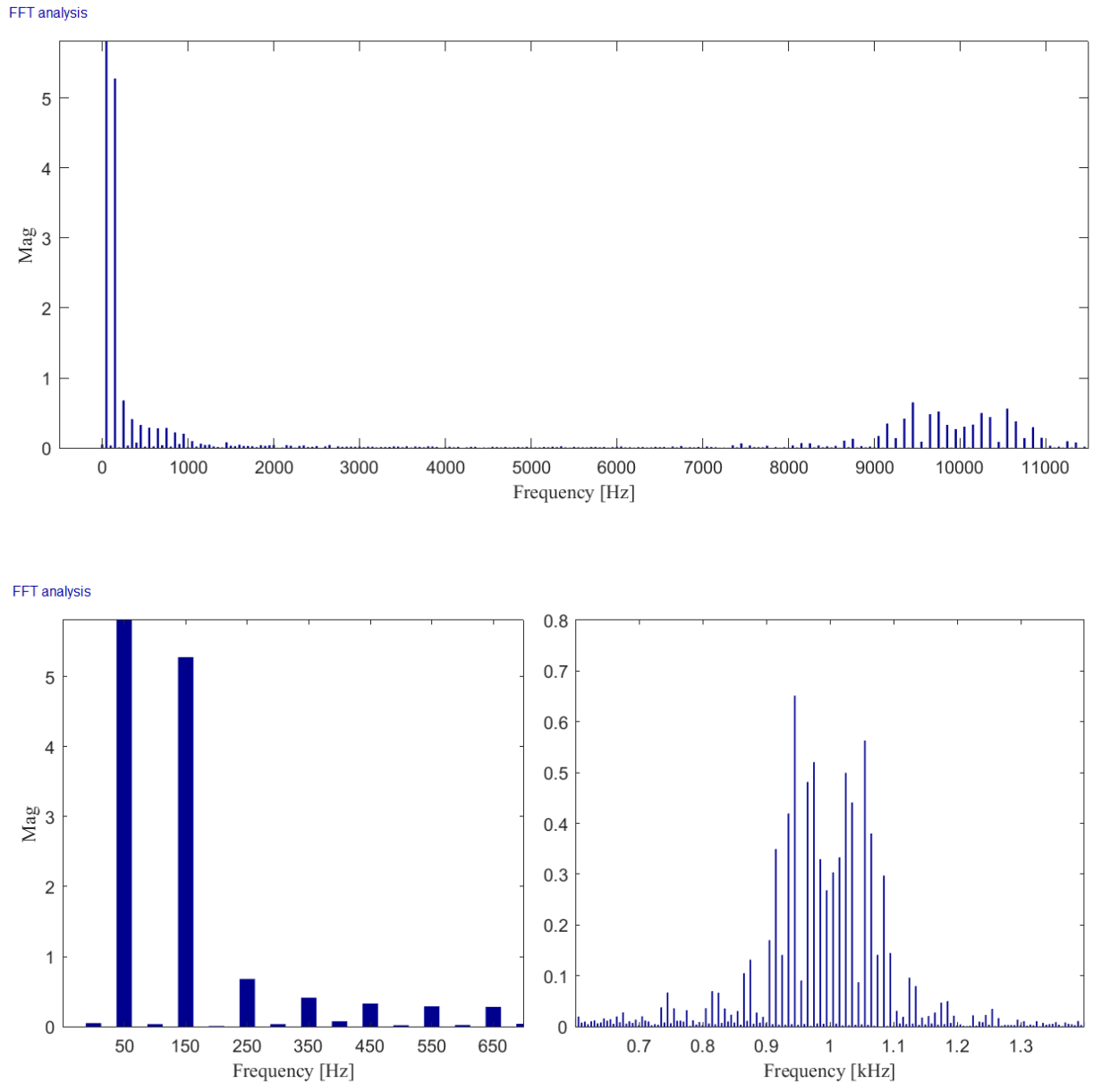


Figure 57 - FFT of the output phase voltage waveform at different frequency ranges with OPTIMAL SERIES RC ENABLED.

	Parameter	Output phase voltage V_a [V]	Output phase current $\hat{I}_s$ [A]
No control	Fundamental	241.3	15.68
	THD [%]	11.79	11.79
Optimal repetitive control	Fundamental	248.0	16.0
	THD [%]	2.3	2.3

Table 7 - THD and fundamental component of output phase voltage and current.

5. Algorithm Implementation

For real – life system implementation, the control algorithms need to be implemented on a digital computing system. In this thesis, the algorithms are investigated for implementation on the Xilinx ® Zynq ® - 7020 All Programmable System on Chip (SoC). This system integrates a dual – core ARM ® Cortex ™ - A9 based processing system (PS) and a 28 nm Xilinx Artix7 programmable logic FPGA (PL) in a single device. This enables combined effects of the software programmability of the ARM processor and the hardware programmability of an FPGA [40].

Through model – based design, the algorithms can be synthesized into C/C++ and HDL code to target Xilinx Zynq – 7000 AP SoC devices. The model – based design approach means that from a Simulink model, one can refine and define the system requirements, simulate the design and automatically generate code for the system. The Embedded Coder is used to generate C/C++ code for the software model, while the HDL Coder is used to generate Verilog or VHDL code to produce Intellectual Properties cores (IP cores) to be programmed on to the PL. The IP cores are designed to be connected to an embedded processor through the AXI4 – Lite interface. The processor, running Linux operating system, acts as master and the IP cores as slaves. The software model is generated as an executable, running on the ARM processor under Linux. Furthermore, a software interface model is generated with IP core driver blocks for the processor to be able to interact with the hardware. This software model is essentially AXI4 – Lite write and read register blocks. The processor will read to and write from the registers to interact with and control the IP cores based on the implemented software model.

Blocks outside the Simulink HDL subsystem will be realized as the software model. This introduces the design challenge of what elements will be placed in the model for software implementation and what elements will be placed in the model for hardware implementation. This is an evaluation which depends on the complete system. Algorithms which require fast executions such as high – bandwidth control should be implemented as hardware logic to utilize the speed and parallelization possibilities. For example, systems like MMC, where PWM gate signals should be sent simultaneously to all submodules, the modulation logic should be implemented on hardware rather than sequential – based software implementation on processor. Secondly, for algorithm implementation on hardware such as FPGA, a compromise between hardware resource use and performance based on the data type (fixed point vs. floating point, word layout and quantization) used for the algorithm must be thoroughly analyzed. The Fixed Point Tool in MATAB proposes fixed – point data types that maximize precision and cover the required data range, and can be used for analyzing optimal fixed – point data types [41].

The Simulink model blocks must be adapted to the specific data type which is needed for the design. HDL Coder support both native floating – point code generation and fixed – point code generation, and allows for isolating operations that need floating – point while keeping the rest of the algorithm fixed point for the most optimized implementation. Optimization of fixed – point data types is not discussed further in this thesis.

The model – based workflow (Figure 58) has been used in this thesis to look at execution time and hardware resource use of the discussed algorithms. The algorithms have been tested through simulation in Simulink in both the continuous and discrete time domain, transformed into specific data type models and then synthesized into IP cores and C – code to look at the algorithm resource use and execution time on the processor and FPGA.

Due to time restrictions for this thesis, only execution time of the repetitive controller was investigated for software implementation, while both repetitive control structures and proportional resonant control were implemented on hardware and investigated for resource use and critical path estimations.

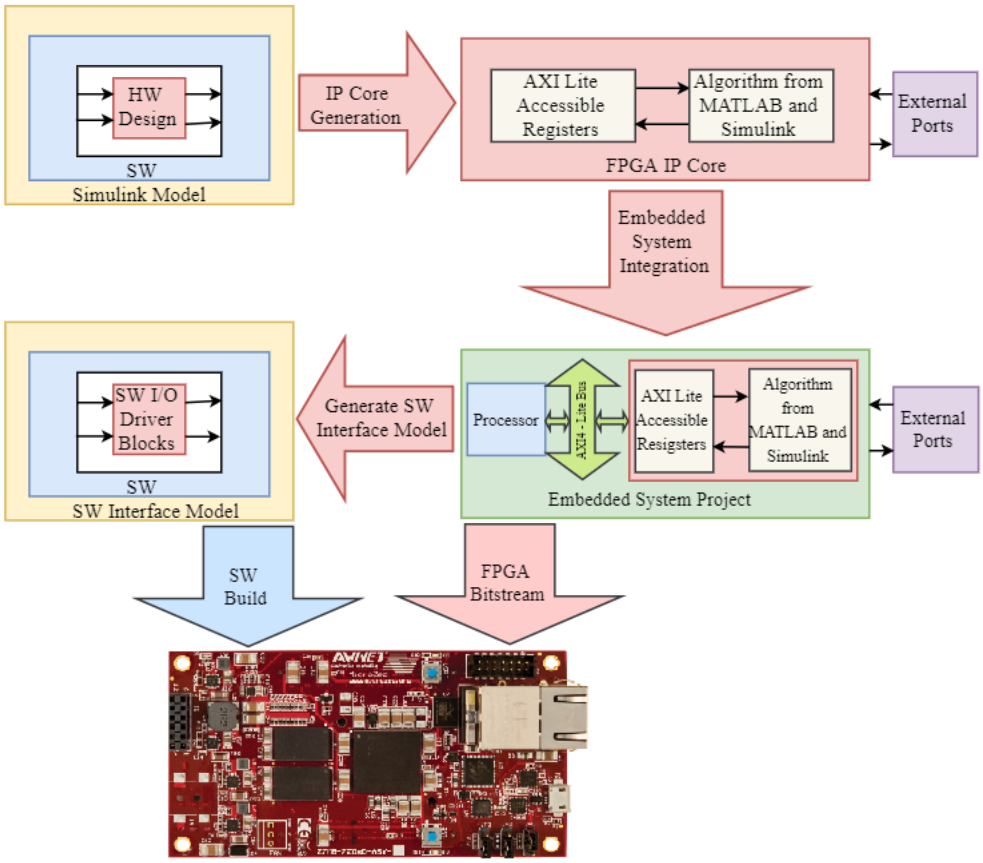


Figure 58 - Model based design workflow for algorithm implementation on MicroZed [5].

The generated C/C++ and HDL codes are automatically generated by the mentioned software tools. Even though the tools allow for advanced optimization and configuration options, further optimization should be investigated by the designer. Code optimization is not investigated to an extensive degree in this thesis. An optimization analysis could reduce code size, execution time and resource use.

A block – diagram showing the MicroZed layout, bank assignment and interconnection between the PS and PL of the Zynq – 7020 can be found in *Appendix - C.* IC Specifications.

5.1. Software Implementation

The Simulink control algorithms can be implemented as embedded C – code on Zynq – 7020. The Simulink block – diagram is validated through simulation and the algorithm transformed into C – code through an automated process using MATLAB Embedded Coder.

5.1.1. Processor – in – the – Loop (PIL)

Processor – in – the – Loop (PIL) is a method used to validate the embedded control algorithm performance and execution time. The control algorithm is run on the Zynq ARM – processor rather than the host processor which is running the simulation. Instead of reading physical sensor values, the sensor values are calculated by the simulation as a result of the specific plant model and used as inputs to the embedded algorithm. The outputs from the embedded control algorithm are fed back into the simulation, contributing to the reference generation.

While running the simulation, execution profiling for the generated code is used to verify different aspects of the code, like [42]

- Determine whether the generated code meets execution time requirements for real – time deployment on the target hardware (the Zynq ARM – processor in this case).
- Identify code sections that require performance improvements.

By configuring the model for code execution profiling, generate the code and run a PIL simulation, the performance can be analyzed through code execution profiling plots and reports. For the optimal *series RC* algorithm, the time used for each execution over the complete simulation period of 0.5 seconds is shown in Figure 59.

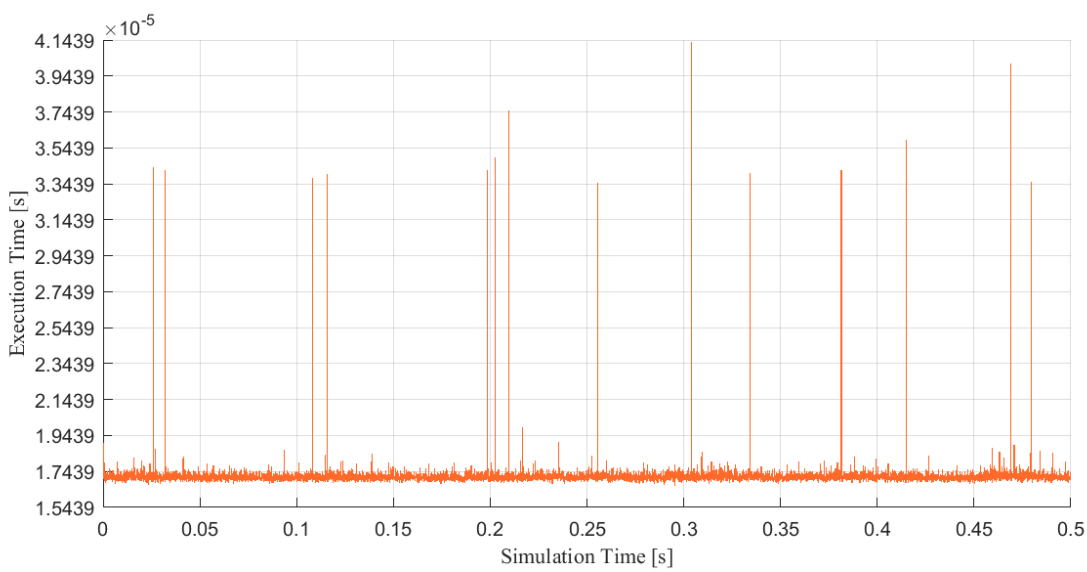


Figure 59 – Execution time of individual executions of the embedded algorithm.

Average and Maximum execution time is shown in Figure 60.

Block: Model

Maximum Execution Time	Average Execution Time	Maximum Self Time	Average Self Time	Calls
rep_controller_p_initialize				
2685	2685	2685	2685	1
rep_controller_profili_Init				
6072	6072	6072	6072	1
rep_controller_profilingTID0 [1e-06 0]				
25380	1423	25380	1423	500001
rep_controller_profilingTID1 [5e-05 0]				
41319	17167	41319	17167	10001

View full code execution profiling report

Figure 60 - Average and maximum execution time.

An average of $t_{ex} = 17.167\ [\mu s]$ and maximum of $t_{\max} = 41.319\ [\mu s]$ is observed. An important remark here is that this execution time is including the time needed for transmitting data between the simulation host PC and the processor where the algorithm is running. Data is transmitted via an ethernet cable, adding significant delay which should not be ignored in this case. To find the delay time, a PIL simulation including only a transmitting block (i.e. no arithmetic operations) is run and profiling executed.

The average transmission delay is observed to be $1.101\ [\mu s]$, yielding the algorithm execution time to be

$$t_{ex} = 17.167 - 1.101 = 16.066\ [\mu s]$$

With the sampling time $T_s = 1/20000 = 50\ [\mu s]$, the algorithm executes well below the max execution time T_{calc} (equal the sampling time step for the given modulation strategy) for these system settings.

5.2. Hardware Implementation

The PL partition can be used for algorithm implementation to offload the processor, increase algorithm execution speed and/or if special considerations like many I/O capabilities are required. Complex algorithms may require significant execution time when running as embedded code on processor due to their *sequential* nature. The processor breaks op the code into sequences and execute each sequence one at a time at each time step. As described in chapter 3.1.4 on selection of sampling time, there is a maximum calculation time T_{calc} allowed for processing the control algorithm before the modulation function needs to be updated. If the control algorithm execution time T_{ex} exceeds T_{calc}, some action must be made in order to either reduce T_{ex} or increase T_{calc}. The latter can be increased by changing the modulation method (doubling T_{calc} by changing modulation method from *a – PWM – double* to for example *s – PWM – start* (see [19]) for fixed f_c) or reducing the carrier frequency f_c. Both will have negative impact on the converter THD performance. Another option is to reduce T_{calc}, which can be done by reducing the control algorithm complexity, or by implementing the control algorithm on programmable logic in for example a FPGA. Due to the option of configuring the processing of the FPGA in a *parallel* manner, the entire control algorithm may be executed within a single clock tick. More realistically, far fewer clock ticks than in the sequential processor, only limited by the algorithm's critical path[14]. Hardware logic design can thus outperform a processor at certain tasks, even at an overall lower clock rate, due to the parallel design ability. Furthermore, large data volumes can be passed through the same algorithm because the data can be distributed across parallel execution units built into the PL, obtaining higher data throughput than with only a processor.

The discussed control algorithm, and Matlab and Simulink models in general, can be implemented on hardware by using MathWorks® HDL Coder™ to generate synthesizable Verilog® and VHDL® code. The HDL Workflow Advisor is used to generate HDL code from the Simulink model and package it into an IP core. The Advisor facilitates RTL[15] code (VHDL or Verilog, VHDL will be used for this system) and testbench generation from a Simulink subsystem, performs synthesis tasks by invoking a supported third – party synthesis tool (Xilinx Vivado Design Suite for this system) and annotates critical path information back to the system. The Advisor generates HDL code, but needs a synthesis tool to synthesize, implement and generate the bitstream from the HDL code for FPGA implementation. The Advisor guides the user through the necessary tasks for full deployment and provides feedback on each task, and information on how to modify the model to complete the task if the task fails [43].

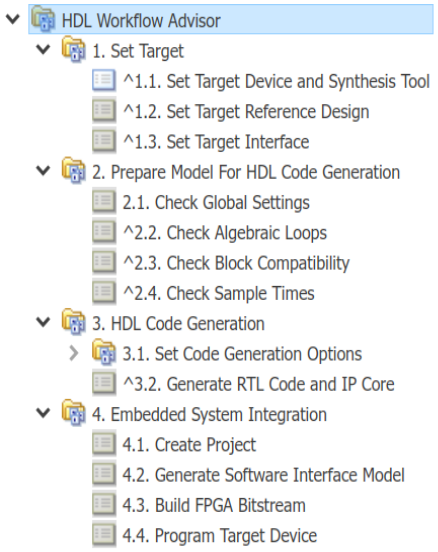


Figure 61 - HDL Workflow Advisor automated tasks.

[14] Critical path – defined as the path between input and output with the maximum time delay.
[15] Register Transfer Language (RTL) is synthesizable HDL code (VHDL or Verilog). HDL code supports constructs which are both synthesizable (implementable on PL) and non – synthesizable. Non – synthesizable HDL can be commands such as *wait*, which is mainly used to make the testbench more powerful.

The Workflow tasks are shown in Figure 61. The algorithm to be implemented is the optimal series RC. After successfully running through tasks 1 – 2, running task 3 generates RTL code and an IP core of the subsystem for the Xilinx Vivado environment. Vivado synthesize and implement the logic design into an IP core and a bitstream[16] generated to be programmed into the hardware. The IP core can now interact with the Zynq – 7020 processing system. HDL code for both fixed - point and floating – point representation of the control algorithm is generated, to look at differences in estimated execution time and required hardware resources.

5.2.1. Fixed – Point Implementation

For fixed point implementation on the FPGA via the HDL Workflow Advisor, an HDL compatible Simulink model must be utilized. Blocks within the HDL Coder library in Simulink can be used, and the model requires further considerations of signal generation and processing. For implementation of transfer functions, the data types need to be changed into the chosen fixed – point format.

Fixed - point data types are represented as *"data type=fixdt(Signed, WordLength, FractionLength)"* in MATLAB/Simulink. The parameters to be decided is

1) *Signed* – 0 (false) for unsigned, 1(true) for signed. Chosen to reflect whether the data needs to be represented with negative numbers. If the *Signed* bit is 1, the first bit of the word is reserved for the two's compliment mathematical operation. For a word with *n* bit *WordLength* and 0 bit *FractionLength*, the possible value which can be obtained are

   i) *Signed:* $[-2^{n-1},\ 2^{n-1}-1]$

   ii) *Unsigned:* $[0,\ 2^{n}-1]$

2) *WordLength* – number of bits *n* in the word.

3) *FractionLength* – number of fraction bits n_{frac} in the word. The remaining number of integer bits after choosing the number of fraction bits then becomes $n_{\text{int}} = n - n_{\text{frac}}$. The integer part is decided by the max integer value which needs to be represented. For a per – unitized system with controller saturation limits set at for example 2 p.u., the required number of integer bits is $n_{\text{int}} = 2$ as two bits are required to represent the decimal value 2. The fraction length is decided by the required precision. The bits to the right of the radix point are fraction bits and the bits to the left are integer bits, as seen in Figure 62. The higher the fraction length, the better precision may be obtained.

Signed bit	$2^{n_{\text{int}}-1}$	$2^{n_{\text{int}}-2}$	$\cdots$	2^{1}	2^{0}	Radix point	2^{-1}	2^{-2}	$\cdots$	$2^{-n_{\text{frac}}}$

Figure 62 – Word bit layout for fixed - point data type representation.

[16] Configuration data describing the logic to be loaded into the FPGA.

Figure 63 shows the discrete Simulink block – diagrams with data type set to 16 – bit. The signal name *sfix16_En_14* described the signed 16 – bit word length with 14 fraction bits. As the error signal will obtain both positive and negative values, the signed bit is set to 1. There is, however, important to be careful when setting data types within the repetitive controller transfer function. With the repetitive controller transfer function on the form

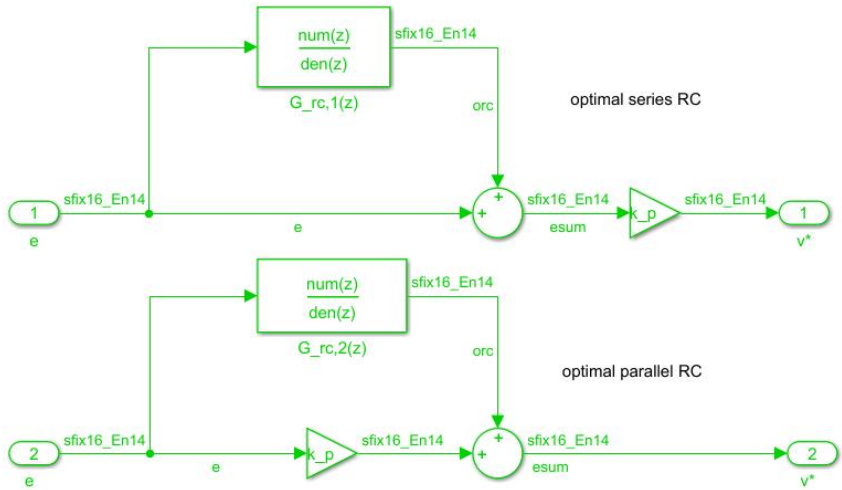


Figure 63 - 16 - bit implementation of the optimal series RC (upper) and optimal parallel RC (lower).

$$G_{rc}(z) = \frac{b_m + b_{m-1}z^{-1} + \ldots + b_0 z^{-m}}{1 + a_{n-1}z^{-1} + \ldots + a_0 z^{-n}} \tag{0.165}$$

and the parameters of the series RC defined in (151)[17], none of the coefficients exceeds 1, hence the integer bit is set to 1. With fixed - point representation, the lowest non – zero number representation possible is $2^{-n_{\text{frac}}}$ Fixed - point operations may produce results which has more bits than the operands, thus information loss is possible. Fixed - point multiplication can potentially result in a number which has more bits than each of the operands, hence the result must be rounded or truncated which can lead to precision loss. Multiplying two numbers with n_{int} integer bits and n_{frac} fraction bits, the result can have up to $2n_{\text{int}}$ integer bits and $2n_{\text{frac}}$ fraction bits. Some fraction precision loss is common in fixed - point, fractional multiplication. If any integer bits are lost, the value may significantly change, hence great care must be taken in choosing the output data type of such an arithmetic operation in order to maintain information [44].

[17] This is the transfer function for series repetitive control with maximum tuned parameters for both the proportional and integral gain.

5.2.2. HDL Code Generation

The HDL Workflow Advisor creates a HDL Code Generation Report through the HDL Coder. The report estimates the high – level resource use and critical path, and gives an optimization report containing information on pipelining, the IP core generation and traceability. The High – Level Resource Report gives an estimate on the logic resources which will be used when the IP core is implemented on the PL. The report is shown in Figure 64 for 16 – and 32 – bit.

WordLength	16 - Bit		32 - Bit	
Structure	Series	Parallel	Series	Parallel
Multipliers	6	7	6	6
Adders/Subtractors	8	8	8	8
Registers	201	201	201	201
Total 1 - Bit Registers	3216	3216	6432	6432
RAMs	0	0	0	0
Multiplexers	1	1	1	1
I/O Bits	36	36	68	68
Static Shift Operators	0	0	0	0
Dynamic Shift Operators	0	0	0	0

Figure 64 - High - Level Resource Report.

The Advisor further gives an estimate on the critical path, as seen in Figure 65 and summarized in Table 8.

Word Length	16 – Bit		32 – Bit	
Algorithm	Series RC	Parallel RC	Series RC	Parallel RC
Critical Path [ns]	11.175	8.724	16.552	12.026

Table 8 - Critical path estimates.

Critical Path Delay : 11.175 ns
Critical Path Begin : s_state_out199
Critical Path End : Saturation

Critical Path Details

Id	Propagation (ns)	Delay (ns)	Block Path
1	0.4600	0.4600	s_state_out199
2	4.3860	3.9260	nume_gain199
3	4.3860	0.0000	s_nume_acc1
4	4.3860	0.0000	s_nume_acc2
5	4.3860	0.0000	s_nume_acc3
6	4.3860	0.0000	output_cast
7	5.7830	1.3970	Sum
8	9.7090	3.9260	Gain
9	11.1745	1.4655	Saturation

Figure 65 - Critical path estimates of series RC 16 - Bit word length.

The Advisor further generates an *after – transformation* Simulink model highlighting the critical path, shown for the series RC in Figure 66 - Critical path Simulink model, highlighting the critical path in blue and displaying time delay for each algorithmic operation and delay element which cause delay. Figure 66. The critical path delay will inherently exist in the programmed hardware structure of the repetitive controller due to arithmetic operations and physical delays like internal cell delay, wire delay or cell input capacitance.

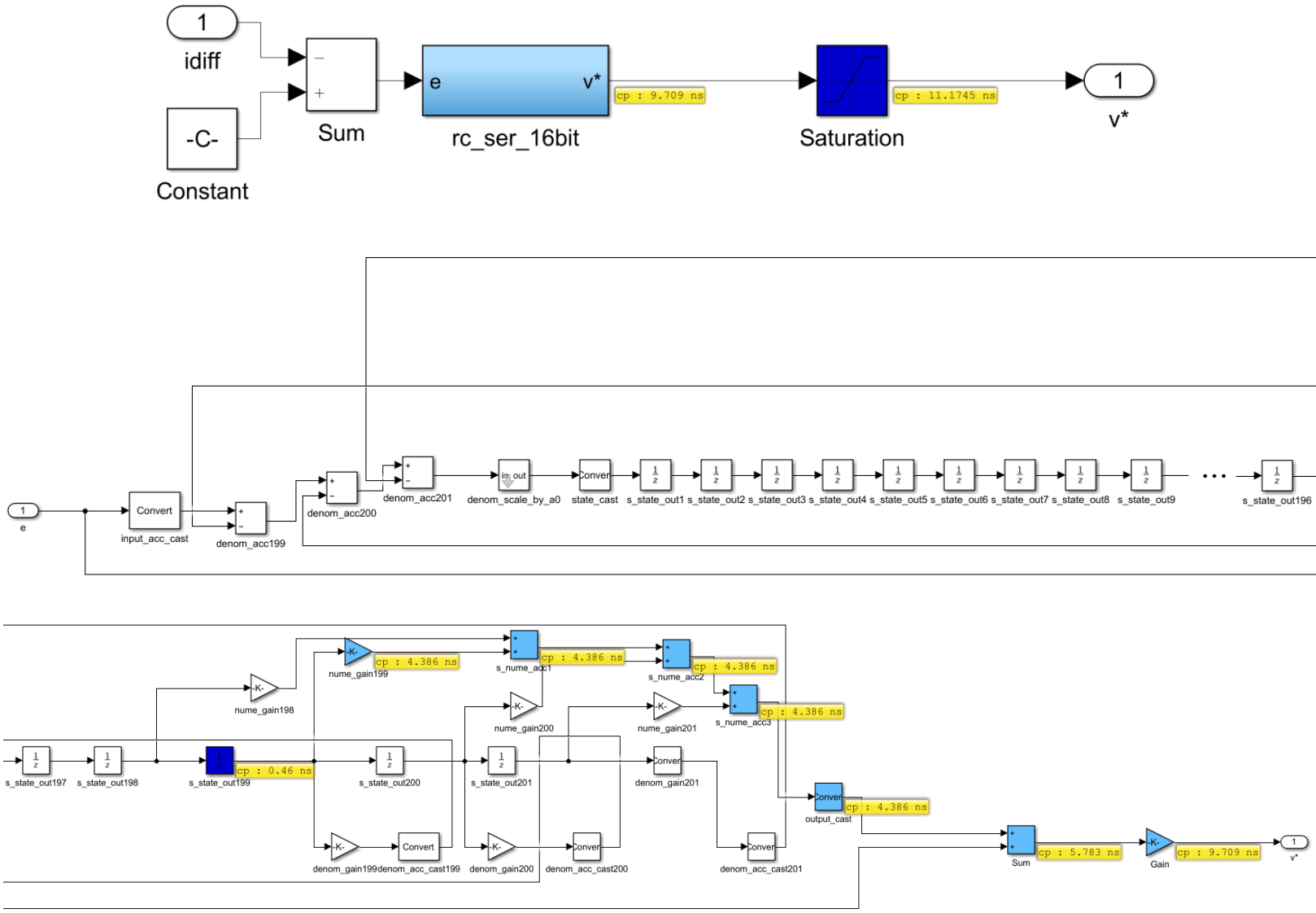


Figure 66 - Critical path Simulink model, highlighting the critical path in blue and displaying time delay for each algorithmic operation and delay element which cause delay. The lower figure is the expanded subsystem of "rc_ser_16bit".

5.3. Xilinx Vivado Design Suite – Design Synthesis and Implementation

The HDL code packaged as an IP core from the HDL Workflow Advisor is implemented into a complete system design with Xilinx Vivado Design Suite. This block design contains IP cores from all operating systems, that is algorithm, processing system and interface IP cores. The designed IP cores of both the series (*rc_ser_#bit_ip*) and parallel (*rc_par_#bit_ip*) RC are connected to the Zynq processor (*Zynq7 Processing System*) IP core through the AXI4 – Lite interconnection (*AXI Interconnect*) IP core, shown in Figure 67.

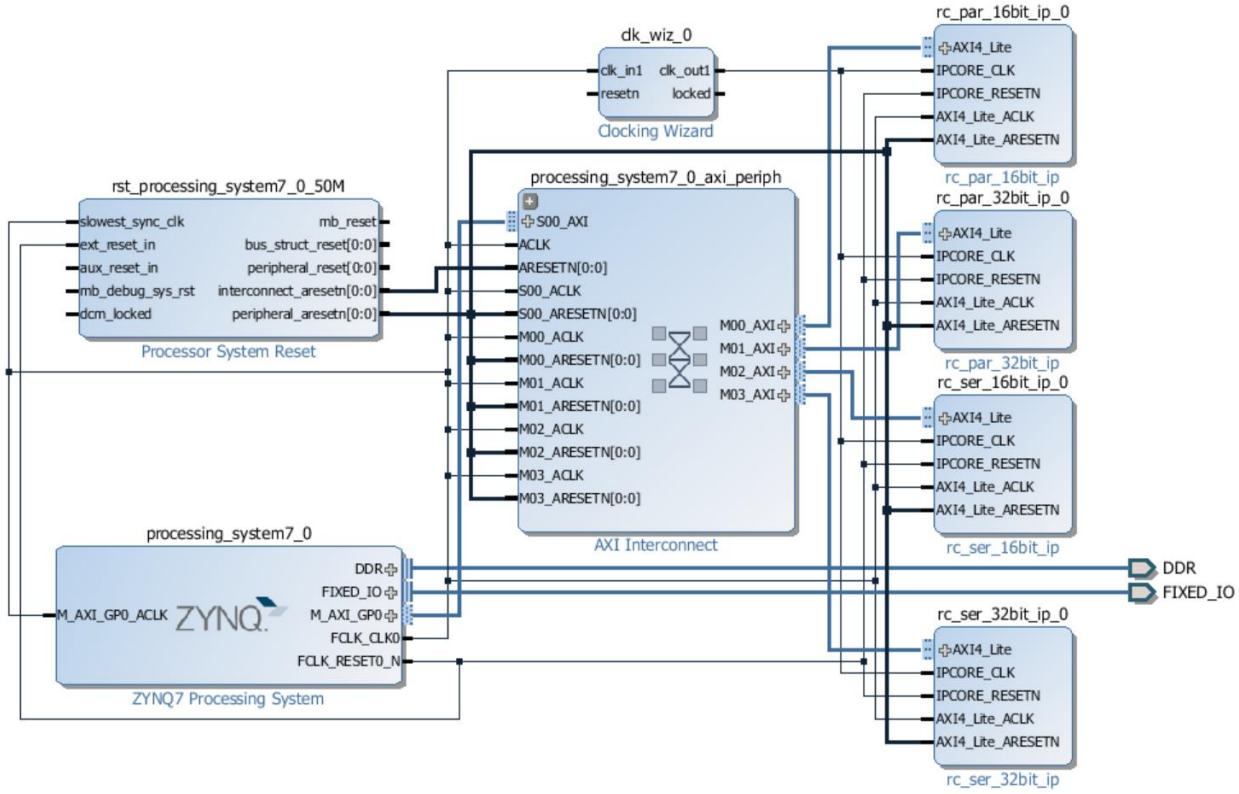


Figure 67 - Xilinx Vivado system block diagram.

The *Clocking Wizard* IP core provides the correct clock frequency for the FPGA, by transforming the processor clock frequency to the designed FPGA clock frequency. The *Processor System Reset* IP cores provides reset signals for the *AXI Interconnect* and the repetitive controller IP cores.

By running Vivado implementation, the IP cores represented in the block diagram are synthesized into a programmable bitstream. The implementation provides a utilization resource report from the implementation – run shown in Figure 68 and Figure 69, giving more in – depth estimates of the resource use of the IP cores in addition to the high – level resource estimation from MATLAB HDL code generation. Exact resource use can, however, not be known until the bitstream is programmed into the FPGA.

Name (Total)	Slice LUTs (53200)	Slice Registers (106400)	Slice (13300)	LUT as Logic (53200)	LUT as Memory (17400)	LUT Flip Flop Pairs (53200)	DSPs (220)
design_rc_imp_wrapper	2925	2490	1152	2213	712	3898	61
design_rc_imp_i	2925	2490	1152	2213	712	3898	61
clk_wiz_0 (Clocking Wizard)	0	0	0	0	0	0	0
processing_system7_0 (Zynq7 Processing System)	0	0	0	0	0	0	0
processing_system7_0_axi_periph (AXI Interconnect)	597	723	243	530	67	782	0
rc_par_16bit_ip_0 (16 - Bit parallel RC IP Core)	257	369	149	152	105	469	7
rc_ser_16bit_ip_0 (16 - Bit series RC IP Core)	860	497	320	643	217	1087	24
rc_par_32bit_ip_0 (32 - Bit parallel RC IP Core)	284	371	159	179	105	529	6
rc_ser_32bit_ip_0 (16 - Bit series RC IP Core)	910	499	324	693	217	1072	24
rst_processing_system7_0_50M (Processing System Reset)	17	31	11	16	1	22	0

Name	Slice LUTs	Slice Registers	Slice	LUT as Logic	LUT as Memory	LUT Flip Flop Pairs
design_rc_imp_wrapper	5.50%	2.34%	8.66%	4.16%	4.09%	7.33%
design_rc_imp_i	5.50%	2.34%	8.66%	4.16%	4.09%	7.33%
clk_wiz_0 (Clocking Wizard)	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
processing_system7_0 (Zynq7 Processing System)	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
processing_system7_0_axi_periph (AXI Interconnect)	1.12%	0.68%	1.83%	1.00%	0.39%	1.47%
rc_par_16bit_ip_0 (16 - Bit parallel RC IP Core)	0.48%	0.35%	1.12%	0.29%	0.60%	0.88%
rc_par_32bit_ip_0 (32 - Bit parallel RC IP Core)	1.62%	0.47%	2.41%	1.21%	1.25%	2.04%
rc_ser_16bit_ip_0 (16 - Bit series RC IP Core)	0.53%	0.35%	1.20%	0.34%	0.60%	0.99%
rc_ser_32bit_ip_0 (16 - Bit series RC IP Core)	1.71%	0.47%	2.44%	1.30%	1.25%	2.02%
rst_processing_system7_0_50M (Processing System Reset)	0.03%	0.03%	0.08%	0.03%	0.01%	0.04%

Figure 68 - Resource utilization of system IP cores.

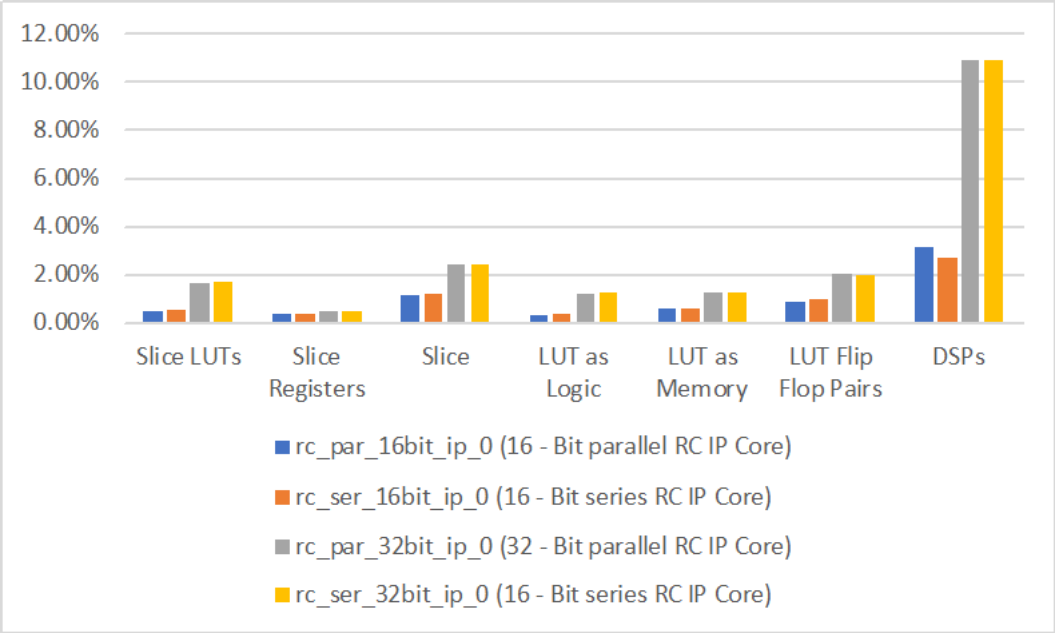


Figure 69 - Resource utilization of the control algorithms at 16 and 32 – bit word lengths.

The main contribution for the difference in the resource use is in the use of integrated DSP[18] slice IP cores (termed DSP48E1 in Zynq, see Figure 70). Increasing the word – length have great impact on the use of DSP slices. This is because the default MACC takes 18 and 25 – bit input word lengths. For any word length greater than this, additional MACC's need to be used corresponding to the word lengths of the data. This is also observed in the High – Level Resource Report generated by MATLAB.

[18] DSP – Digital Signal Processing – a term applying broadly to continuous mathematical processes attempted in real – time, such as digital filtering, convolution and Fast Fourier Transforms. In the case of DSPs for the programmable logic of the Zynq 7020, this refers to 18x25 MACCs (18 bits times 25 bits Multiply/Accumulator) DSP slices.

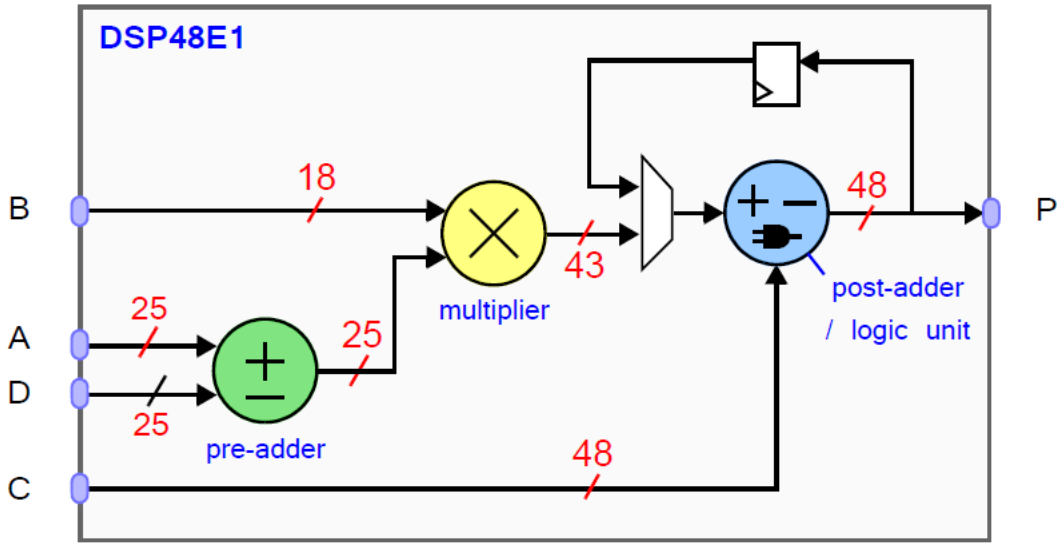


Figure 70 - Arithmetic capabilities of the DSP48E1 slice. Various computations are possible, involving one, two or all three arithmetic operators[4] .

5.3.1. Proportional – Resonant Controller Implementation

The discussed PR controller tuned to $2\omega_0$ and $4\omega_0$ (eq. (163)) is implemented with 16 – and 32 – bit data types with similar procedure as for the repetitive controllers. Critical path of the discussed algorithms is shown in Figure 71.

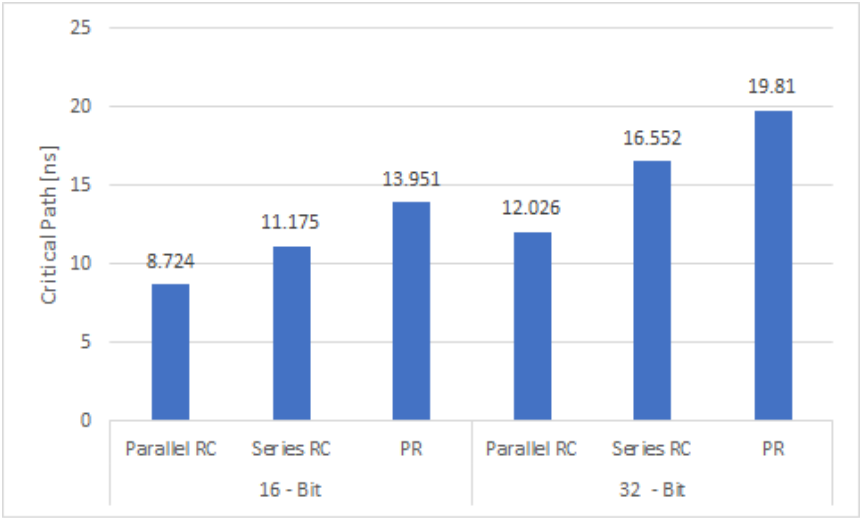


Figure 71 - Critical path summary.

and resource utilization report of the IP cores of RC and PR fixed – point implemented algorithms shown in Figure 72.

Name (Total)	Slice LUTs (53200)	Slice Registers (106400)	Slice (13300)	LUT as Logic (53200)	LUT as Memory (17400)	LUT Flip Flop Pairs (53200)	DSPs (220)
design_rc_imp_wrapper	2925	2490	1152	2213	712	3898	61
design_rc_imp_i	2925	2490	1152	2213	712	3898	61
clk_wiz_0 (Clocking Wizard)	0	0	0	0	0	0	0
processing_system7_0 (Zynq7 Processing System)	0	0	0	0	0	0	0
processing_system7_0_axi_periph (AXI Interconnect)	597	723	243	530	67	782	0
rc_par_16bit_ip_0 (16 - Bit parallel RC IP Core)	257	369	149	152	105	469	7
rc_ser_16bit_ip_0 (16 - Bit series RC IP Core)	860	497	320	643	217	1087	24
rc_par_32bit_ip_0 (32 - Bit parallel RC IP Core)	284	371	159	179	105	529	6
rc_ser_32bit_ip_0 (16 - Bit series RC IP Core)	910	499	324	693	217	1072	24
pr_2_4_32bit_ip_0 (16 - Bit PR)	705	270	248	705	0	824	36
pr_2_4_16bit_ip_0 (32 - Bit PR)	120	138	75	120	0	214	10
rst_processing_system7_0_50M (Processing System Reset)	17	31	11	16	1	22	0

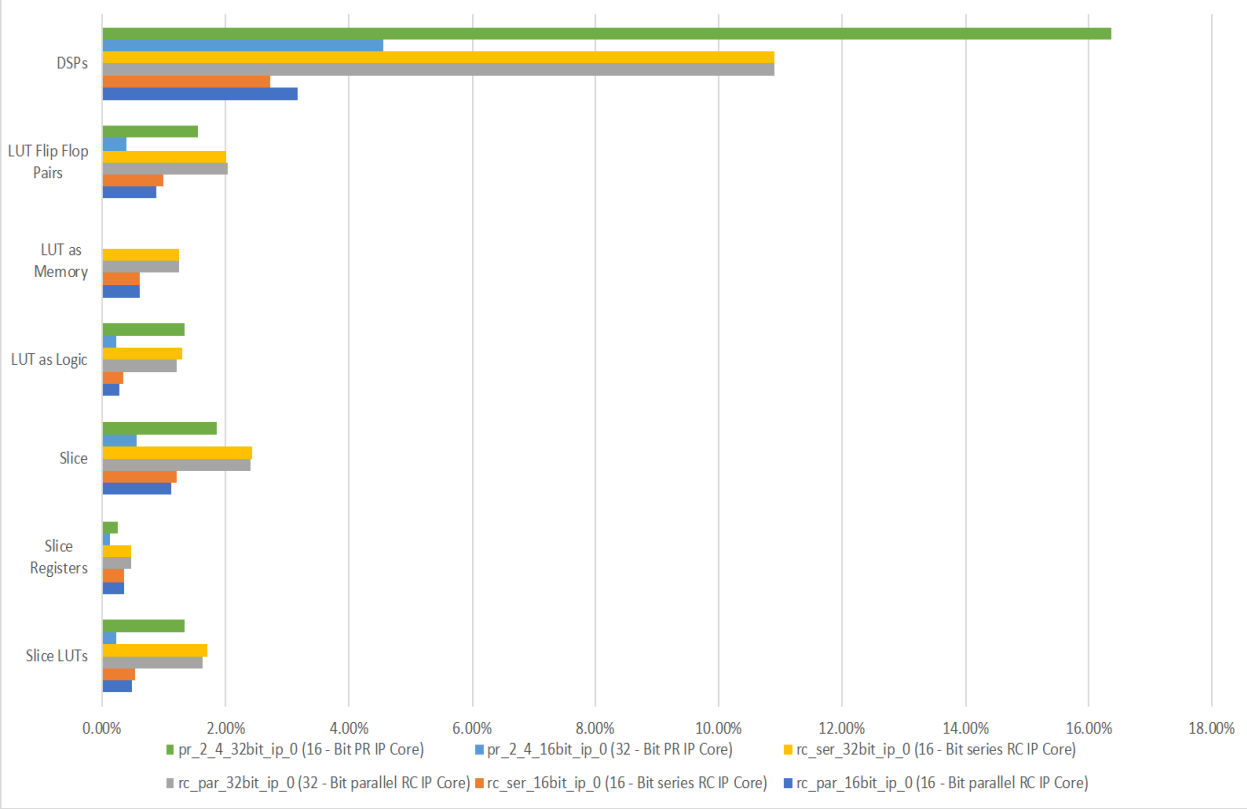


Figure 72 - Resource utilization report number of total available (upper) and percent (lower).

5.4. Floating – Point Implementation

For algorithms which require high dynamic range with good precision, floating – point implementation usually yields better results. However, floating – point arithmetic operations require more logic and clock resources, resulting in longer algorithm execution time and more hardware usage. Instead of reserving specific number of bits for integer and fraction, floating – point representation reserves a specific number of bits for its mantissa and exponent. Fixed - point data type represents a number approximately to a fixed number of significant digits (significand) and scaled using an exponent in some fixed base. According to IEEE Standard 754, the single precision native[19] floating – point requires 32 – bit word length, and is structured as [45]

Signed S	Exponent E	Mantissa M
1 bit	8 bits	23 bits

$$\text{number} = (-1)^{S} \cdot \underbrace{2^{E-127}}_{\text{Base}} \cdot \underbrace{(1.M)}_{\text{Significand}}$$

The maximum and minimum representable IEEE 754 floating – point value is $\pm\left(2-2^{-23}\right)\cdot 2^{127} \approx \pm 3.402823\cdot 10^{38}$ and $\pm 2^{-126}\cdot 1 \approx \pm 1.175494\cdot 10^{-38}$. This gives from 6 – 8 significant decimal digit precision, meaning a decimal number with 6 or fewer significant digits may be represented with single – precision without loss of information. Depending on the precision type, floating – point gives better precision and higher dynamic range than fixed - point representation (for example, a two's compliment 32 – bit fixed – point may give a fraction precision of 2^{-31} with 0 integer bits, trading fraction precision against obtainable magnitude/range).

The resources needed for a hardware implementation of a floating – point algorithm and execution time, on the other hand, is usually much higher for floating – point arithmetic operations and processing than for fixed – point.

The series RC algorithm is implemented in single precision floating – point through similar steps as described for the fixed – point algorithm implementation. The HDL Workflow Advisor creates synthesizable HDL code based on the block diagram showed Figure 75. Similarly to fixed – point implementation, a high – level utilization report is generated by MATLAB based on the HDL code seen in Figure 74.

[19] The term *native* floating – point causes all floating-point data items to use IEEE Standard 754.

Summary Section

Critical Path Delay : 1.700 ns
Critical Path Begin : Saturation_LowerConst
Critical Path End : Data Type Conversion

Critical Path Details

Id	Propagation (ns)	Delay (ns)	Block Path
1	0.0000	0.0000	Saturation_LowerConst
2	1.0580	1.0580	Saturation_Switch1
3	1.7000	0.6420	Saturation_Switch2
4	1.7000	0.0000	Data Type Conversion2

Figure 73 - Critical path - report of floating - point implemented series RC.

The Critical Path – report seen in Figure 73 gives an estimate of the critical path time delay posed by the algorithm. However, not included in the report is the *delay matching*. The delay matching is included in the Simulink model seen in Figure 76 as z^{-3} and z^{-44}. The arithmetic block comprising the repetitive controller transfer function require complex hardware algorithms. For the corresponding hardware implementation to stay cycle – accurate with the Simulink model, the hardware algorithm must compute in the same clock cycle as it does in Simulink. If both the RC transfer function and the unity forward path execute in one clock cycle in Simulink, they should execute in one clock cycle (the hardware clock frequency is different from the Simulink clock cycle) when run on hardware. Depending on the complexity of the hardware implementation of the block, the hardware algorithm will deviate in execution time from what it does in Simulink. MATLAB implements the stated delays to account for this and maintain equal system dynamics as the original Simulink model [46].

The delay added depends on the value set to be the hardware clock frequency[20], which in this case is set to $f_{hw} = 100$ [MHz]. The total critical path is then calculated to be

$$t_{cp} = 1.7 \text{ [ns]} + (3+44) \cdot 10 \text{ [ns]} = 471.7 \text{ [ns]} \tag{166}$$

Multipliers	7
Adders/Subtractors	145
Registers	1370
Total 1-Bit Registers	21739
RAMs	0
Multiplexers	1196
I/O Bits	36
Static Shift operators	23
Dynamic Shift operators	40

Figure 74 - High - Level Resource Report for floating - point implementation.

[20] The Clocking Wizard IP Core is used to configure the clock frequency for the hardware algorithms. It takes the ARM processor clock as input and outputs the wanted hardware clock frequency.

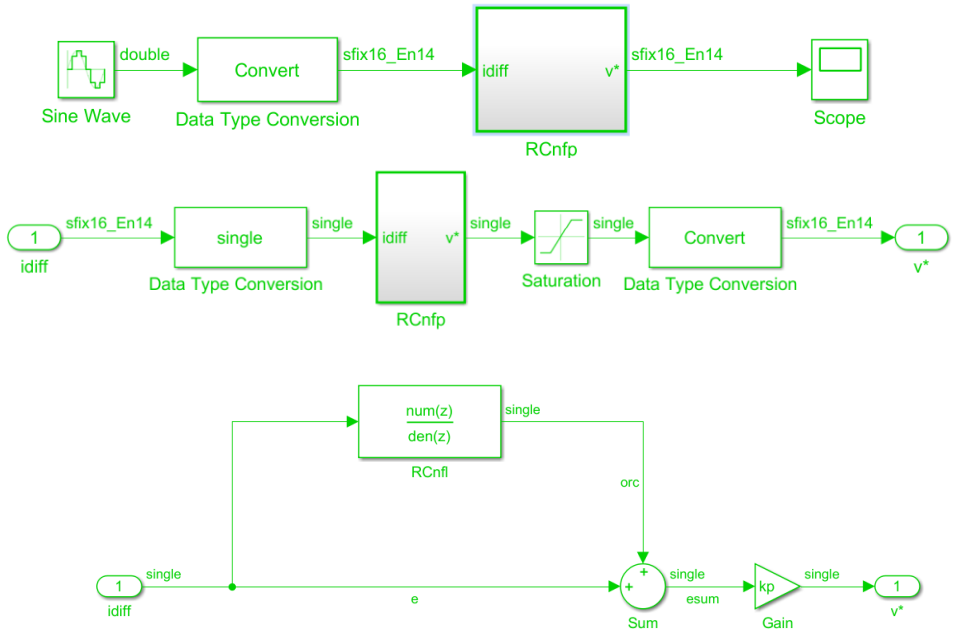


Figure 75 – Simulink block diagram of floating – point data type implementation of series RC. Subsystem RCnfp is expanded from the top to bottom.

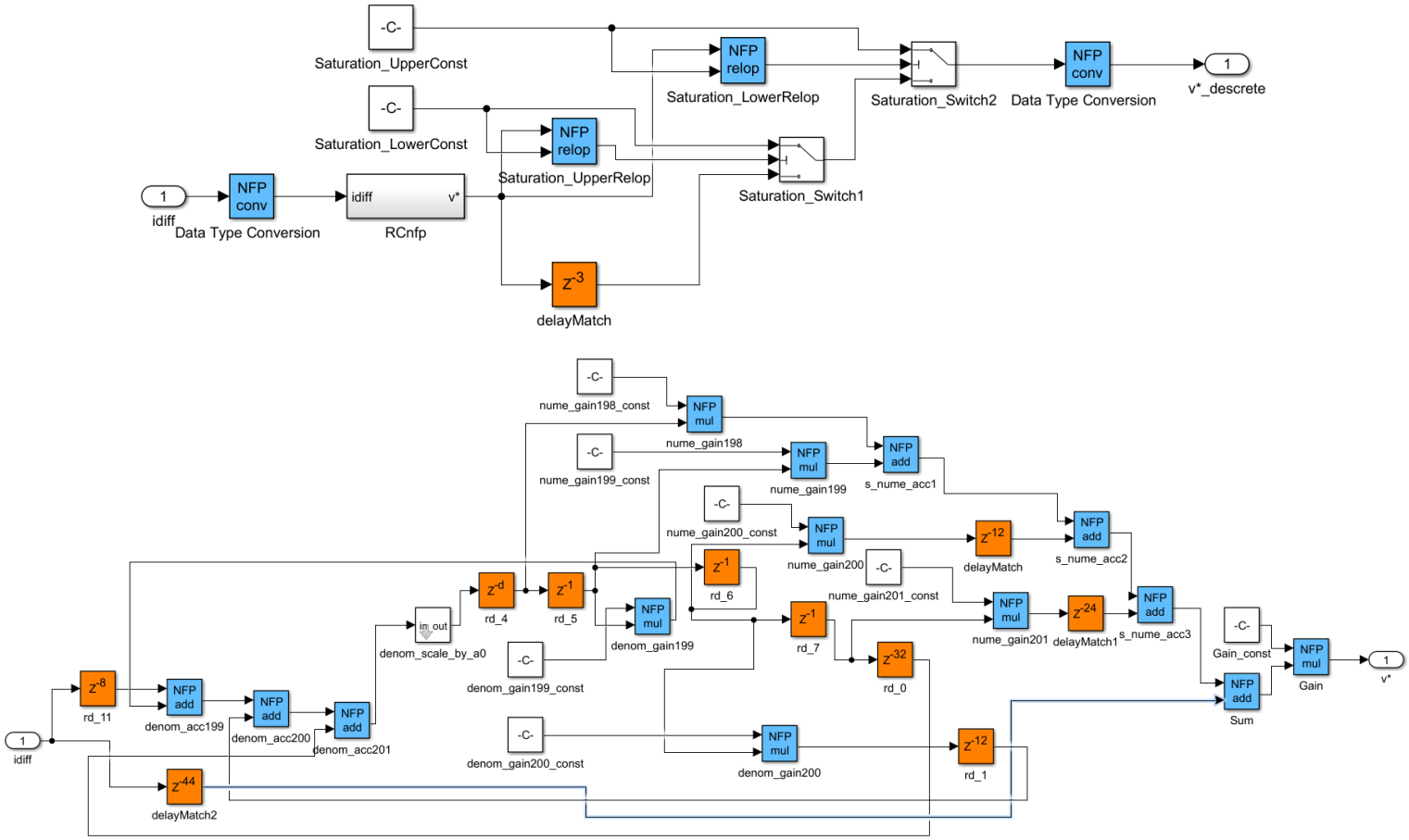


Figure 76 - Equivalent Simulink model of critical path delay. Blue colored block are delay imposed by arithmetic operations, while the orange blocks are delay matching block which are added as delay for the algorithm to work as expected. Lower model is the expanded subsystem RCNFP seen in the upper model.

6. Hardware Design and Laboratory Set Up

This chapter describes the design of the hardware setup of the MMC. Schematic diagrams and board files are designed and created with Eagle PCB design.

6.1. The Modular Multilevel Converter

The configuration of the MMC converter consist of 3 submodules in each leg, comprising two half – bridge submodules per card. The positive pole of the upper arm is connected to the positive DC – link pole and the negative pole of the lower arm connected to the negative DC – link pole. The neutral point of the leg is connected to the load and the load neutral is connected to the neutral of the DC – link, as shown in Figure 1. A system overview is shown in Figure 77.

The submodule is designed as a full – bridge converter and used as half – bridge submodules for MMC applications, yielding 3 submodules per arm and 6 per leg. The advantages for both full – and half – bridge topologies is mentioned in the chapter 2 and can further be reviewed in [8]. The design is made based on approximate electrical characteristics of Table 9.

Parameter	Rated Value
3 – Phase Power	$P_{3\varphi} = 12$ [kW]
Max Submodule Power	$P_{\mathrm{SM,MAX}} = 5$ [kW]
DC – link Voltage	$V_{DC} = 700$ [V]
Submodule Voltage	$V_{SM} = 230$ [V]
Phase Voltage (L-N RMS)	v_{LN} =250 [V]
Submodule Capacitance	$C_{SM} = 1$ [mF]
Arm Inductance	$L_{u,l} = 5$ [mH]
Fundamental Frequency	$f_0 = 50$ [Hz]
Switching Frequency	$f_c = 1000$ [Hz]

Table 9 - MMC electrical characteristics.

6.1.1. MicroZed

The MicroZed™ development board is used to interface the Xilinx Zynq ® -7000 All Programmable SoC to the converter. The MicroZed contains two I/O headers which provides connection to two I/O banks on the PL side of the SoC. An external power supply, provided by the interface card, is needed to activate these banks through MicroHeader 1,2, as seen in the block diagram of Figure 78.

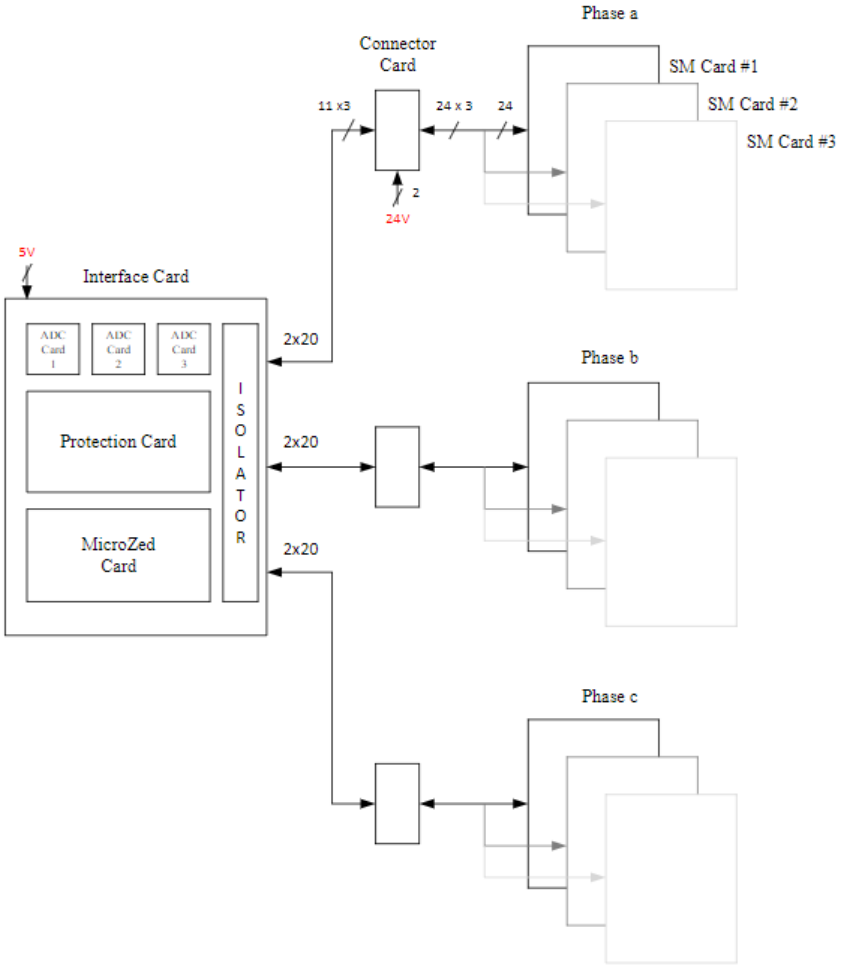


Figure 77 - Schematic overview of laboratory hardware set - up.

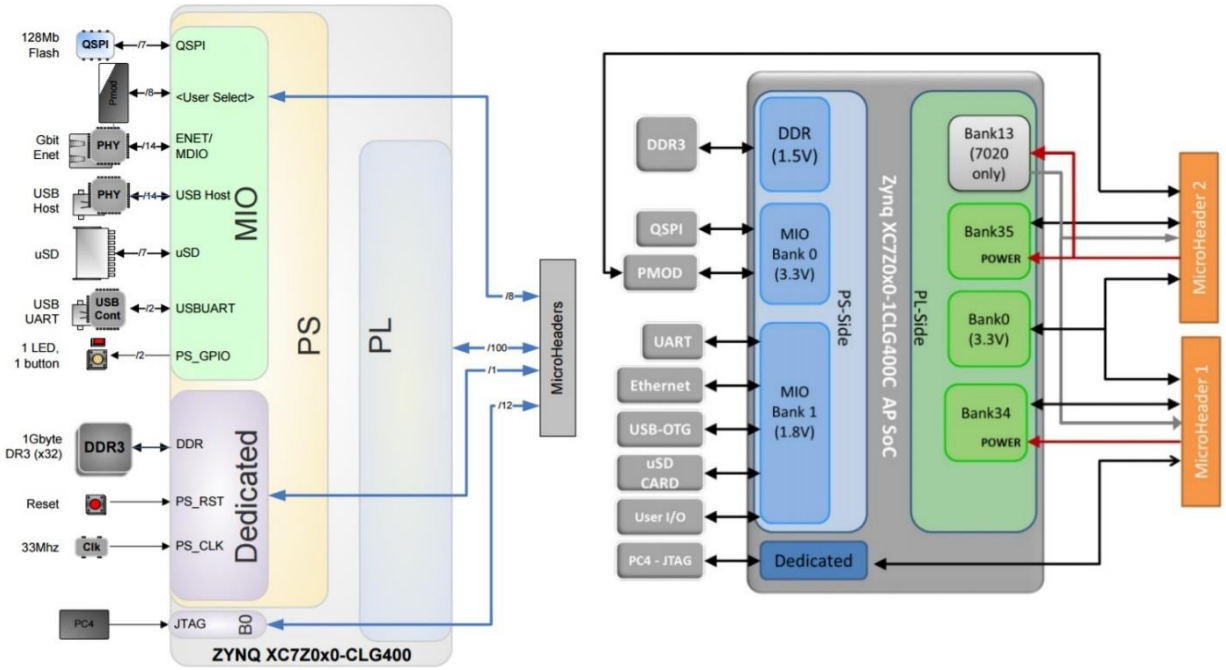


Figure 78 - MicroZed Block Diagram (left) and MicroZed Bank Assignment (right)[2].

A description of the analog circuitry is given in the following section and the Eagle schematics is found in *Appendix*. The schematics is divided in to separate sheets to give a better overview.

- *Sheet 1*

  This sheet shows the submodule input/output pins together with the DC – DC converters for the power supply. There are 24 pins, where 12 of them are connected to analogue ground. The other pins are either output (signals *from* submodule *to* control) or input (signals *to* submodule *from* control) and are explained in Table 10.

Pin Number	Label	Description
1, 3, 5, 7	G*ij* (*i*,*j*=(1-2, 1-2))	Input to gate driver. Gate driver signals (PWM – pulses) received from the controller.
9	$\overline{\text{RESET}}$	Input to gate driver. Resets $\overline{\text{FAULT}}$ output HIGH to enable operation.
11	$\overline{\text{ENABLE}}$	Input. Enables switch pulses (see sheet 2).
13, 15	$\overline{\text{FAULT\_SM}}$	Output from gate driver fault pin. Notifies Zynq about fault condition when goes low.
17, 19	VDC1,2	Output. Submodule capacitor voltage measurement signals.
21, 23	24VCC	24 volts submodule components supply.

Table 10 - Pin description.

  The 24 volt supply is inputted to 3 different converters, the TSR 1 – 2450 providing non – isolated 5V, the TME 2505S providing isolated 5V and the MGJ2D242005SC providing +20, 0 and -5V isolated for the semiconductor switch gate voltages (+20 turn on, -5 turn off). The supply offering isolation is used for IC's requiring isolation from the submodule power supply and the non – isolated is used where this is not needed.

- *Sheet 2*

  The left circuit of the sheet shows the implementation of the dead – time generation for the upper switch pair needed to avoid short circuit the capacitor through the upper and lower switch. Gate signals for the gate driver IC are doubled, where one of the identical signals is delayed with an analog low pass filter before both are AND'ed. Such dead – time delays can also be implemented digitally via code generation either in software or hardware, where it is easier to change the dead – time opposed to the analog circuit described. For changing the dead – time in the analog circuit, the values of *R* and *C* must physically be changed.

  The right circuit shows the enabling function of the gate signals. The gate signals are AND'ed with the enable signal, thus only outputting gate signals when both inputs are high.

- *Sheet 3*

  Sheet 3 shows logic circuit for fault handling. The 74HC04D inverter is used to get the correct normal level of the fault signals. Fault signals are AND'ed using the 74ac08d AND gate IC.

- *Sheet 4*

  Sheet 4 shows the two switch configurations comprising one complete full – bridge topology, or two separate half – bridge topologies. The upper switches (Q1,3) is connected to the positive pole of the submodule capacitor, and the lower switches (Q2,4) to the negative pole. The switches take the gate signals GS11, GS12, GS21, GS22 from the gate driver circuit.

- *Sheet 5*

Sheet 5 shows the schematic for the gate driver circuits. The specific circuit configuration is based upon design recommendations given by the supplier of the gate driver IC. The gate driver IC chosen in here is the HCPL – 316J and the circuit design recommendation for an application circuit can be found in the datasheet. The circuit is configured to provide stable IC supply voltage and stable turn – on/off voltages at the level and timing required by the semiconductor switch. The DSAT pin of HCPL – 316J provides fault protection. The pin monitors the collector – emitter voltage. If the voltage exceeds 7V, the gate voltage VOUT is slowly lowered and the $\overline{\text{FAULT}}$ output goes low.

- *Sheet 6*

This sheet shows the circuit layout for the voltage measurement of the submodule capacitor voltages (VDC1,2). Two IC's are used in this circuit. U$19,20 are ACPL – 7900 – 300E optically isolated operation amplifiers. The ACPL – 7900 provides analog isolation between the submodule capacitor voltage and rest of the signal conditioning circuitry. The isolated measurements are further inputted to LM358D operational amplifier, which provide the proper value of the DC – voltage through its gain for transmission to the Zynq – controller.

The following calculations provides a guideline for the necessary components values. With the components in the schematic layout of the voltage measurement in sheet 6, and the submodule voltage $v_c(t)$ as input, the voltage divider provides the voltage at R_{F1}

$$v_{pre}(t) = \frac{R_{10}}{R_9 + R_{10}} v_c(t)$$

The combination R_{F1} and C_{F1} is an analog RC low – pass, anti – aliasing filter for the ADC, providing the cut – off frequency

$$f_{co,\text{F1}} = \frac{1}{2\pi R_{\text{F1}} C_{\text{F1}}}$$

The gain of the ACPL – 7900 has a nominal gain of $G_{\text{iso}} = 8.2$ with the max/min range $\in (8.16, 8.24)$. The positive $v_{out}^{+}(t)$ and negative $v_{out}^{-}(t)$ output of the ACPL – 7900

$$\left(v_{out}^{+}(t) - v_{out}^{-}(t)\right) = G_{\text{iso}}\left(v_{in}^{+} - v_{in}^{-}\right)$$

is fed to the LM358D op – amp, which is configured to a single – ended, ground – referenced differential amplifier. The output voltage obtained from the LM358D with this configuration then becomes

$$v_0(t) = -v_{out}^{-}(t)\frac{R_{22}}{R_{16}} + v_{out}^{+}(t)\left(\frac{R_{18}}{R_{18} + R_{13}}\right)\left(\frac{R_{16} + R_{22}}{R_{16}}\right) = \frac{R_{18,22}}{R_{16}}\left(v_{out}^{+}(t) - v_{out}^{-}(t)\right)$$

The resistor and capacitance values are design choices to be made to meet the voltage levels. The design should further be evaluated based on application recommendations from data sheets of the specific IC used.

6.1.2. Interface Card

The interface board harbors the ADC's, voltage and current protection circuitry and provides isolation between MicroZed and outer electronic circuit.

- *Sheet 1*

  Sheet 1 shows the schematic for the current transducer LEM LA – 55p, the carrier cards (UZ_BOCC) and three power supplies. The TDK Lambda provides isolated 5V for IC's requiring isolated power supply, the Recom R – 78B provides non – isolated 5V for IC's which don't require isolation and LMR10510 provides isolated 3.3V for the MicroZed. The LMR is non – isolated, thus requires an isolated voltage supply provided by the TDK Lambda.

  In the MMC, the upper and lower arm currents need to be measured. The current transducer LA – 55P from LEM is used. Given in data sheet, the current consumption for the transducer is $I_c = 10(@\pm U_c) + I_s$ [mA], with the supply voltage $U_c = \pm 15$ [V] $(\pm 12 ... \pm 15$ recommended$)$. The conversion ratio is $K_N = 1{:}1000$, giving a secondary current of $I_s = K_N I_P$. The peak power consumption of the LA – 55P should be calculated as

  $$P_{\mathrm{LA}} = U_c I_c$$

  With a measuring range of $I_p = \pm 70$ [A], the peak power consumption is calculated to be $\hat{P}_{LA} = 30 \cdot (0.010 + 0.070) \approx 2.4$ [W]. Thus, a 3 [W] DC/DC converter outputting ± 15 [V] is necessary to allow maximum measurement range. The electrical characteristics of the hardware platform used for the design yields a peak phase current of 11.4 [A]. The complete measuring range is thus not necessary during normal operating conditions. A wide measuring range is, however, advantageous to account for possible overloading and fault conditions.

- *Sheet 2*

  Sheet 2 shows the circuit configuration for the PWM – signal isolation. The ADUM7640 6 – channel, bidirectional digital isolator is used for isolating these signals.

- *Sheet 3*

  Sheet 3 shows the circuit layout for the AD7606 ADC, and the isolation of operational signals and the digital sensor values provided by the ADC. The AD7606 is a 16 bit, 8/6/4 channel, simultaneous sampling Analog-to-Digital Data Acquisition system (DAS) and the isolation IC is the digital isolator ADUM7640.

- *Sheet 4*

  Sheet 4 shows the circuit layout for isolation of the fault signals using the digital isolator ADUM7640

- *Sheet 5*

  Sheet 5 shows the protection circuit. This circuit is used for over – voltage protection. The measured submodule capacitor voltage is compared with an upper and lower limit for which the LM339N comparators trigger. Each submodule capacitor voltage is measured and each comparator trigger – signal is AND'ed such that if either one of the voltage go out of range, a fault signal is given.

7. Conclusions

This thesis investigates methods of suppressing the circulating current generated in the MMC. Significant amplitudes of even multiples of the fundamental frequency component, especially the second, are present in the circulating current with no circulating current control applied. The current does not contribute to energy transfer from DC to AC side. It causes significant power loss and adds additional electrical stress on power switches and passive components.

The main technique investigated for circulating current suppression is plug – in repetitive controller (RC) with PI control. The RC is investigated as plug – in structure with both a PI controller (the system nominal controller) in series (series RC) and parallel (parallel RC). The main emphasis has been on digital implementation of repetitive control structures. Stability analysis has been carried out in the discrete time – domain and tuning – guidelines for the repetitive and nominal PI controller presented. The RC structures are compared to proportional resonant (PR) and pure PI control. It is shown through simulations in Simulink® that both series and parallel RC yields lower steady – state error compared to pure PI and PR tuned $2\omega_0$ and $4\omega_0$. An optimal design for the repetitive controller is concluded, that is plug – in repetitive control with a proportional nominal controller in series, termed optimal series RC.

The optimal series RC yields a RMS steady – state error of 0.03 [A] compared to 0.1 [A] for the PR tuned to $2\omega_0$ and $4\omega_0$, 0.2 [A] for maximum tuned PI and 10 [A] without any circulating current control enabled.

The settling time is decreasing for increasing nominal controller gains k_p and RC gain k_r. The settling time is found to be 20 ms. for the optimal series RC. No overshoot is observed for the investigated RC control structures.

The series RC algorithm is synthesized into C code for ARM processor execution. The algorithm execution time is obtained through Processor – in – the – Loop (PIL) simulation. The execution time of the optimal RC design is below the allowed control calculation time T_{calc} for the given sampling period with good margin. This means that other algorithms, for example the capacitor voltage balancing algorithm, could be implemented on the processor as long as the total execution time does not exceed $T_{calc} = T_s$ with adequate margin. This would better utilize processor resources and free up hardware resources for other algorithm hardware implementations.

The algorithms *series* and *parallel RC*, and *PR* tuned to $2\omega_0$ and $4\omega_0$ is implemented on Artix7 FPGA. This study is performed to look at the utilized hardware resources and to estimate critical paths of the algorithms. The IP core and critical path estimates of the algorithm was obtained from HDL workflow in Simulink®. The algorithm resource utilization was obtained from synthesis and implementation using Xilinx Vivado. The algorithms are implemented as both 16 – and 32 – bit fixed – point, and the *series RC* is implemented as single precision floating – point.

The *PR* control structure demands high hardware resources compared to *series* and *parallel RC* for fixed point implementation. As expected, all hardware entities (that is, LUTs, DSPs etc., see Figure 72) increase their utilization as the word – length increase. Most of the resource utilization, and the largest percent – wise increase when increasing the word - length, lays with the DSPs. This is expected due to the inherent 18x25 MACC DSP configuration of the Zynq Artix7 FPGA (see Figure 70). When inputs go above the word lengths 18 and 25 bit, the number of DSPs need to be increased to accommodate the increased signal width.

The same trend is seen for the estimated critical path. The time delay increases for increasing word – length, and is highest for 32 – bit *PR*. When increasing the word – length, the arithmetic operations gets more frequent as the DSPs increase, hence a higher, estimated delay is seen.

Floating – point hardware implementation results in increased resource utilization and critical path delay, which is expected due to more complicated and frequent arithmetic operations.

Even though the trend for controlling VSC has been towards the use of floating – point DSPs, fixed – point DSPs and FPGAs/ASICs are cheaper, draw less power per arithmetic operation and generally have shorter algorithm execution time [25]. Furthermore, with the use of a combined software/hardware system such as the Zynq – 7000 All Programmable SoC series, it is clear that both fixed – point and floating – point algorithms should be investigated and analyzed to conclude on the best control algorithms for best overall system performance.

8. Future Work

- Quantization and optimized fixed – point design of the algorithms for hardware implementation.

For the performance of fixed – point algorithms to be comparable to floating – point algorithms, the issue of quantization should be properly addressed. The investigated word – lengths for fixed – point were 16 – and 32 – bit for *series/parallel RC* and *PR* implementation. The investigated implementation for single precision floating – point was *series RC*. The investigated parameters were limited to resource utilization and critical path estimations. The fixed – point performance, however, was not investigated. For sampling rates much higher than the resonant frequency, fixed – point implementation of resonators such as the resonant filter and repetitive control, demands special attention to avoid quantization noise. Proper filter realization, such as DFIIt[21] [25], should be investigated further to obtain optimal numerical properties for fixed – point implementation.

The number of fraction bits has great impact on the accuracy and the algorithm performance. A wider range of word – lengths and fraction – bits should be investigated to yield optimal performance for fixed – point algorithm implementation. MATLAB Fixed Point Tool is an accelerating design tool for such processes.

- The repetitive controller under varying frequency conditions

In real systems, variations in the real period time of the circulating current may occur due to for example transients and load variations affecting the fundamental frequency. The repetitive controller has proven performance for $N_s = T_s / T_p$, but any variations in T_p could lead to performance degradation. As seen in Figure 79, small variations in the frequency leads to large variations in the gain of the internal model, and a similar effect is seen in the closed – loop system phase [29]. This can be addressed in several ways, for example *varying sampling approach* or *high order repetitive control* design methods [29, 47, 48].

- Further algorithm investigation

A complete evaluation of controllers for sinusoidal tracking/rejection should be analyzed and investigated for implementation on software/hardware and run on the hardware platform to asses real – life performance. PR control should be tuned to more frequencies to make the control performance more similar to RC. This allows for a more accurate comparison between the two algorithms. Furthermore, different control algorithms for closed – loop voltage control should be investigated. Arm – energy controllers and techniques for open – loop voltage control, such as predictive control and estimation techniques for the submodule voltages ripple, should be investigated.

The voltage balancing algorithm and the chosen modulation technique should be analyzed in similar ways as is done for the circulating current control algorithms. As the PWM ideally should be implemented on hardware logic to ensure fast gate signaling, the modulation algorithm should particularly be investigated for resource utilization and critical path delay. This is to ensure proper system performance and that adequate hardware resources are available for implementation.

[21] Direct Form II transposed.

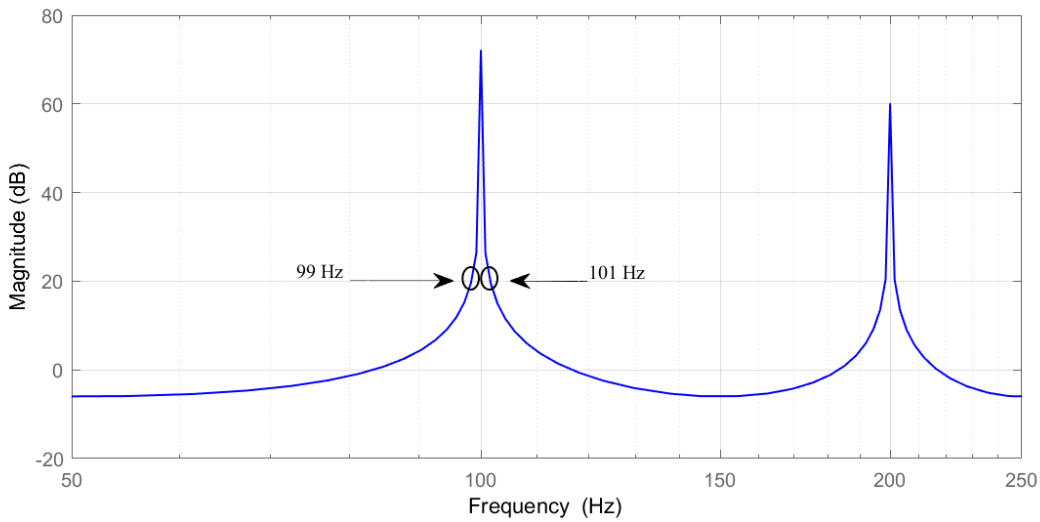


Figure 79 - Frequency response of internal model.

- Hardware platform

The hardware platform should be completed and experiments must be carried out for the investigated control methods to suppress the circulating current. Comparison with other promising control techniques within different scenarios must be investigated.

References

[1] "LEM Current Transducer LA - 55P. Data Sheet. ," 2015.

[2] "MicroZed Zynq Evaluation and Development and System on Module Hardware User Guide," 28.01.2014.

[3] A. Devices, "AD7606 Data Sheet. 8-/6-/4-Channel DAS with 16-Bit, Bipolar Input, Simultaneous Sampling ADC," 2010.

[4] R. E. Louise Crockett, Martin Enderwitz, Robert Stewart, *The Zynq Book - Embedded Processing with the ARM® Cortex®-A9 on the Xilinx® Zynq®-7000 All Programmable SoC*. Strathclyde Academic Media, 2014.

[5] N. Levine, "A Guided Workflow for Zynq Using MATLAB and Simulink," *MathWorks,* 09.10.2013.

[6] N. Ahmed, A. Haider, D. Van Hertem, L. Zhang, and H. P. Nee, *Prospects and challenges of future HVDC SuperGrids with modular multilevel converters*. 2011, pp. 1-10.

[7] R. M. A. Lesnicar, "An innovative modular multilevel converter topology suitable for a wide power range," *Power Tech Conference Proceedings, 2003 IEEE Bologna,* 2004.

[8] S. Debnath, J. Qin, B. Bahrani, M. Saeedifard, and P. Barbosa, "Operation, Control, and Applications of the Modular Multilevel Converter: A Review," *IEEE Transactions on Power Electronics,* vol. 30, no. 1, pp. 37-53, 2015.

[9] L. H. Kamran Sharifabadi, Hans - Peter Nee, Steffan Norrga, Remus Teodorescu, *Design, Control, and Application of Modular Multilevel Converters for HVDC Transmission Systems*. John Wiley & Sons, 2016.

[10] K. Ilves, A. Antonopoulos, S. Norrga, and H. P. Nee, "Steady-State Analysis of Interaction Between Harmonic Components of Arm and Line Quantities of Modular Multilevel Converters," *IEEE Transactions on Power Electronics,* vol. 27, no. 1, pp. 57-68, 2012.

[11] A. Hassanpoor, S. Norrga, H. P. Nee, and L. Ängquist, "Evaluation of different carrier-based PWM methods for modular multilevel converters for HVDC application," in *IECON 2012 - 38th Annual Conference on IEEE Industrial Electronics Society*, 2012, pp. 388-393.

[12] X. Liu, A. Lindemann, and H. Amiri, "Theoretical and experimental comparison of different control strategies for modular multilevel converters," in *2014 IEEE 15th Workshop on Control and Modeling for Power Electronics (COMPEL)*, 2014, pp. 1-9.

[13] L. Lin, Y. Lin, Z. He, Y. Chen, J. Hu, and W. Li, "Improved Nearest-Level Modulation for a Modular Multilevel Converter With a Lower Submodule Number," *IEEE Transactions on Power Electronics,* vol. 31, no. 8, pp. 5369-5377, 2016.

[14] L. Holmes, *Pulse Width Modulation for Power Converters - Principles and Practice* (IEEE Series on Power Engineering). A JOHN WILEY & SONS, INC., PUBLICATION, 2003.

[15] A. Ouerdani, Belkhodja, "Harmonic Analysis of Pulse Width Modulation-Based Strategies for Modular Multilevel Converter," *INTERNATIONAL JOURNAL of RENEWABLE ENERGY RESEARCH,* vol. 6, no. 3, 2016.

[16] S. Gao, Xu Zhang, Songmei Zhang, "Anti-aliasing Filter Circuit Design for Active Power Filter," *Second International Conference on Intelligent Human-Machine Systems and Cybernetics,* 2010.

[17] K. Ogata, *DISCRETE - TIME CONTROL SYSTEMS*. PRENTICE - HALL INTERNATIONAL 1987.

[18] Z. Landau, *Digital Control Systems - Design, Identification and Implementation* (Communication and Control Engineering). Springer - Verlag London Limited, 2006.

[19] N. Hoffmann, F. W. Fuchs, M. P. Kazmierkowski, and D. Schröder, "Digital current control in a rotating reference frame - Part I: System modeling and the discrete time-domain current controller with improved decoupling capabilities," *IEEE Transactions on Power Electronics,* vol. 31, no. 7, pp. 5290-5305, 2016.

[20] R. Nilsen, *Electric Drives. Lecture Notes. 4th edition.* . NTNU Trondheim, 2016.
[21] C.-T. Chen, *Linear System Theory and Design. Third Edition.* . State University of New York at Sony Brook: OXFORD UNIVERSITY PRESS, 1999.
[22] F. J. Taylor, *Digital Filters: Principles and Applications with MATLAB*. Institute of Electrical and Electronics Engineers, Inc.: John Wiley & Sons, Inc., 2012.
[23] H. Kabiri, McGrath, "Double Synchronous Frame Current Regulation of Distributed Generation Systems under Unbalanced Voltage Conditions without Sequence Current Separation," *2015 IEEE Applied Power Electronics Conference and Exposition (APEC),* 2015.
[24] D. G. Holmes, T. A. Lipo, B. P. McGrath, and W. Y. Kong, "Optimized Design of Stationary Frame Three Phase AC Current Regulators," *IEEE Transactions on Power Electronics,* vol. 24, no. 11, pp. 2417-2426, 2009.
[25] L. Harnefors, "Implementation of Resonant Controllers and Filters in Fixed-Point Arithmetic," *IEEE Transactions on Industrial Electronics,* vol. 56, no. 4, pp. 1273-1281, 2009.
[26] L. He, K. Zhang, J. Xiong, and S. Fan, "Repetitive control scheme for circulating current suppression in modular multilevel converters," in *2013 IEEE Energy Conversion Congress and Exposition*, 2013, pp. 963-969.
[27] T. K. Hassan, "A Repetitive-PI Current Controller for Boost Single Phase PFC Converters " *Electrical Engineering Department, University of AL-Mustansiriya, Baghdad, Iraq,* 2011.
[28] S. J. L. a. J. A. Profeta, "A USER'S GUIDE TO REPETITIVE CONTROL SYSTEMS AND THE INTERNAL MODEL PRINCIPLE," *Aerotech, Inc.* .
[29] G. A. R. Fuentes, "Digital Repetitive Control under Varying Frequency."
[30] R. Costa-Castello, J. Nebot, and R. Grino, "Demonstration of the internal model principle by digital repetitive control of an educational laboratory plant," *IEEE Transactions on Education,* vol. 48, no. 1, pp. 73-80, 2005.
[31] Z. LU, YANG, " A General Internal Model Principle Based Control Scheme for CVCF PWM Converters " *2nd IEEE International Symposium on Power Electronics for Distributed Generation Systems,* 2010.
[32] C. Li, D. Zhang, and X. Zhuang, "Theory and applications of the repetitive control," in *SICE 2004 Annual Conference*, 2004, vol. 1, pp. 27-34 vol. 1.
[33] C. H. v. d. B. Philipp Schülting, Rik W. De Doncker, "A Generalized Control Design Approach for a Repetitive Controller on Current Harmonics," *Control and Modeling for Power Electronics (COMPEL), 2015 IEEE 16th Workshop* 2015.
[34] Y. Shin-Chung and T. Ying-Yu, "Adaptive repetitive control of a PWM inverter for AC voltage regulation with low harmonic distortion," in *Power Electronics Specialists Conference, 1995. PESC '95 Record., 26th Annual IEEE*, 1995, vol. 1, pp. 157-163 vol.1.
[35] L. He, K. Zhang, J. Xiong, and S. Fan, "A Repetitive Control Scheme for Harmonic Suppression of Circulating Current in Modular Multilevel Converters," *IEEE Transactions on Power Electronics,* vol. 30, no. 1, pp. 471-481, 2015.
[36] Z. Kai, K. Yong, X. Jian, and C. Jian, "Direct repetitive control of SPWM inverter for UPS purpose," *IEEE Transactions on Power Electronics,* vol. 18, no. 3, pp. 784-792, 2003.
[37] T. Tomizuka, Chew, "Analysis and Synthesis of Discrete - Time Repetitive Controllers," *Journal of Dynamic Systems, Measurement, and Control,* vol. 111, 1989.
[38] S. ÇETİNKAYA, "REPETITIVE CONTROL OF A THREE PHASE UNINTERRUPTIBLE POWER SUPPLY WITH ISOLATION TRANSFORMER " *A THESIS SUBMITTED TO THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES OF MIDDLE EAST TECHNICAL UNIVERSITY,* 2006.
[39] H. Zhang, Yao, Lu, "Circulating Harmonic Current Elimination of a CPS-PWM-Based Modular Multilevel Converter With a Plug-In Repetitive Controller," *IEEE TRANSACTIONS ON POWER ELECTRONICS,,* vol. 29, no. 4, 2014.

[40] XILINX, "Zynq-7000 All Programmable SoC Overview. Product Specification," 2016.
[41] MathWorks. *Fixed-Point Tool*. Available: https://se.mathworks.com/help/fixedpoint/ug/overview-of-the-fixed-point-tool.html
[42] (2016). *MathWorks, Inc. Documentation: Execution Profiling for Generated Code*.
[43] "HDL Workflow Advisor, MathWorks, Inc. ."
[44] *Fixed-point arithmetic*. Available: https://en.wikipedia.org/wiki/Fixed-point_arithmetic
[45] *Floating-point arithmetic*. Available: https://en.wikipedia.org/wiki/Floating-point_arithmetic
[46] *Delay Balancing and Validation Model Workflow In HDL Coder™*. Available: https://se.mathworks.com/help/hdlcoder/examples/delay-balancing-and-validation-model-workflow-in-hdl-coder.html
[47] M. Steinbuch, "Repetitive control for systems with uncertain period-time ," *Automatica 38 (2002) 2103 – 2109,* 2002.
[48] G. Pipeleers, B. Demeulenaere, J. D. Schutter, and J. Swevers, "Robust high-order repetitive control," in *2008 American Control Conference*, 2008, pp. 1080-1085.

Appendix

A. Simulink Simulation Block Diagram

A. 1. Electrical System

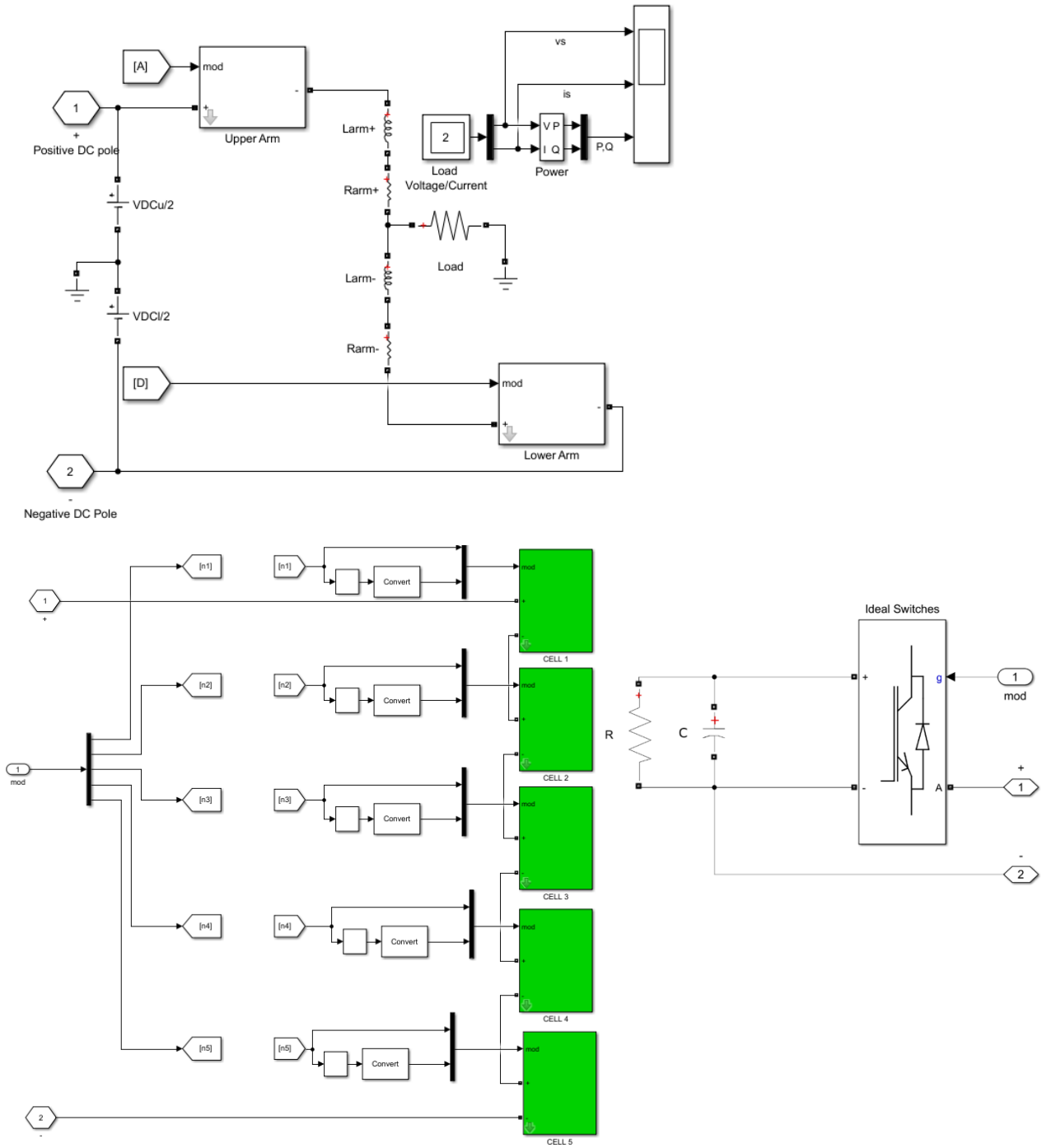
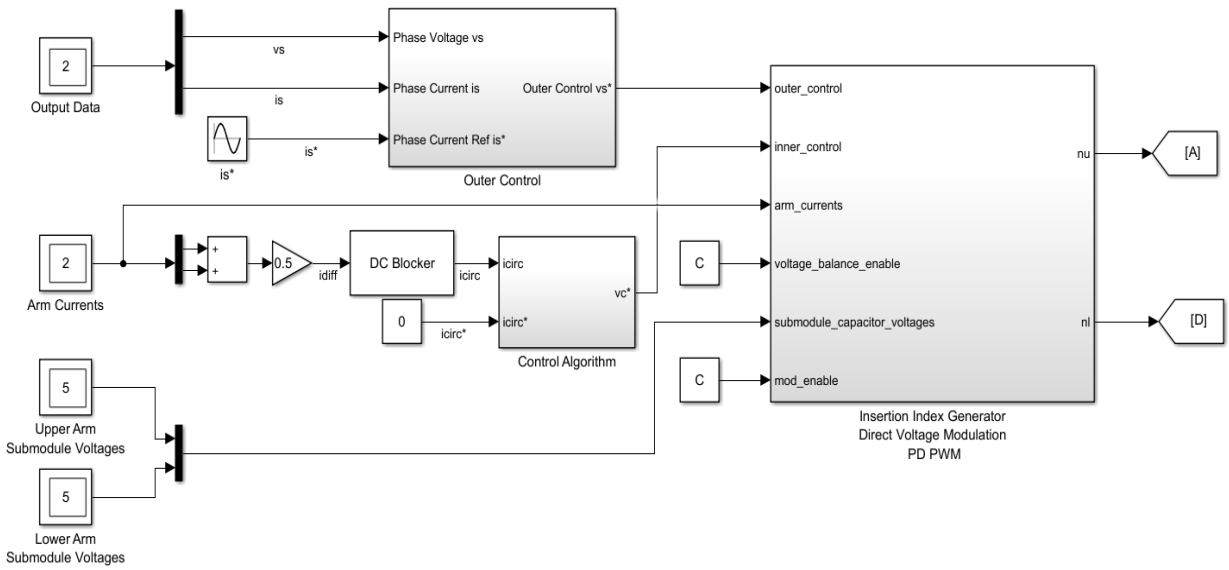


Figure 80 – Top view of Simulink MMC simulation model (upper), expanded upper arm (lower left)) and the half – bridge configuration comprising one submodule (lower right)

A.2. Control and Modulation System



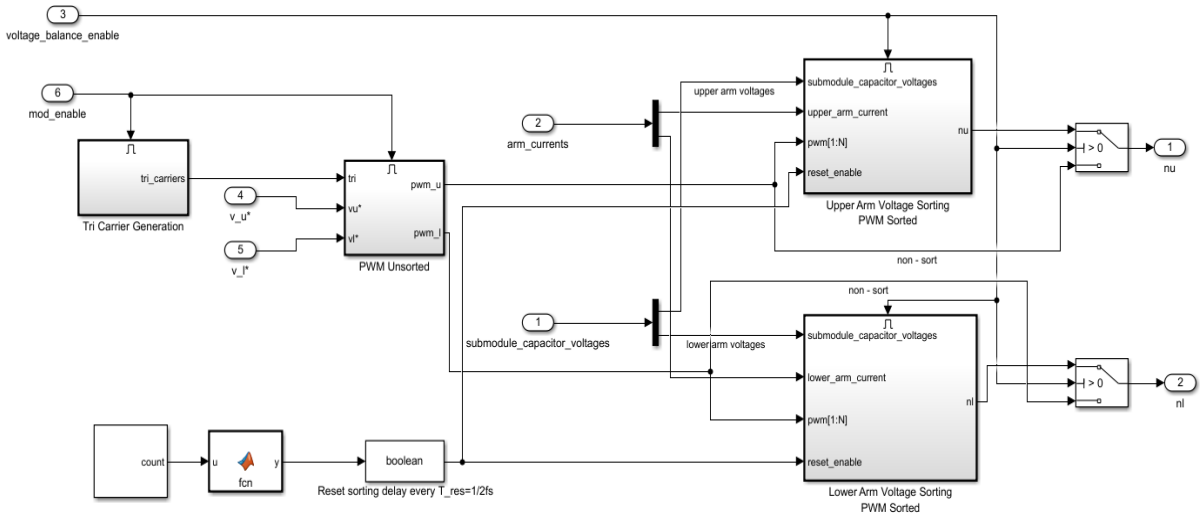


Figure 81 - Top view of control system block diagram (upper). Expanded "Insertion Index Generator Direct Voltage Modulation PD PWM" block diagram in Simulink (lower)

A.3. Fixed – point HDL Compatible Simulink Algorithms

A.3.1. Control System Overview

The Zynq – 7020 is capable of a maximum clock frequency of $f_{\max} = 667$ [MHz] [40]. To avoid high – frequency noise, the clock frequency for the control structure used is chosen to be $f_{clk} = 50$ [MHz]. For the represented algorithms, 16 – bit word lengths are used.

A.3.2. Triangular Carrier Generation

The triangular carrier consists of two sawtooth waves with respectively incrementing and decrementing count values at twice the frequency of the wanted triangular carrier frequency. The decrementing sawtooth is subtracted from the incrementing sawtooth at the time count of one sawtooth waveform period and half the wanted triangular carrier period. With a n bit HDL counter, the counter's maximum count value is $b_{\max} = 2^n - 1$. A built – in HDL counter, which wraps around to zero when the limit – value is reached, is used. With the step size of 1, the counter increments the count value at every clock period time step $T_{clk} = 1/f_{clk} = 2 \cdot 10^{-8}$ [s]. For a specified triangular frequency f_{tri} and clock frequency f_{clk}, the value the counter must count to, to reach half the period of the triangular carrier is

$$b_{sawtooth} = \text{round}\left(\frac{f_{clk}}{2 \cdot f_{tri}}\right) \tag{167}$$

The required counter word – length becomes

$$n = \text{ceil}\left(\log(b)/\log(2)\right) \tag{168}$$

Where "ceil" is a function which rounds the input to nearest integer in the direction of positive infinity. To get the desired triangular wave, a switch is used to output the incremental sawtooth the first half of the wanted period (switch selector is HIGH) and the decremental sawtooth the second half of the wanted period. To achieve this, a pulse generator is pulsating at $2 \cdot f_{tri}$. The desired count value of the pulse – generating counter becomes

$$b_{pulse} = \text{round}\left(\frac{f_{clk}}{f_{tri}}\right) \tag{169}$$

With the necessary word length

$$n_{pulse} = \text{ceil}\left(\log\left(2 \cdot b_{pulse}\right)/\log(2)\right) \tag{170}$$

The counter value is passed through the MATLAB function given in Figure 82.

```
function y  = fcn(u, b_sawtooth)
%#codegen

if u<=cast(b_sawtooth,'uint16')
    y=cast(1,'uint16');
else
    y=cast(0,'uint16');
end
```

Figure 82

When the counter value reaches the value $b_{sawtooth}$, the function outputs 1, else 0. This ensure that the pulse will have the proper high - time

$$T_{high} = b_{sawtooth} \cdot T_{clk} \tag{171}$$

enabling the switch to switch at the correct time. The "cast" function is used to ensure the values outputted and inputted are unsigned 16 – bit integers. (fig) Shows a simulation example with $f_{clk} = 50\ [\text{MHz}]$ and $f_{tri} = 4000\ [\text{Hz}]$.

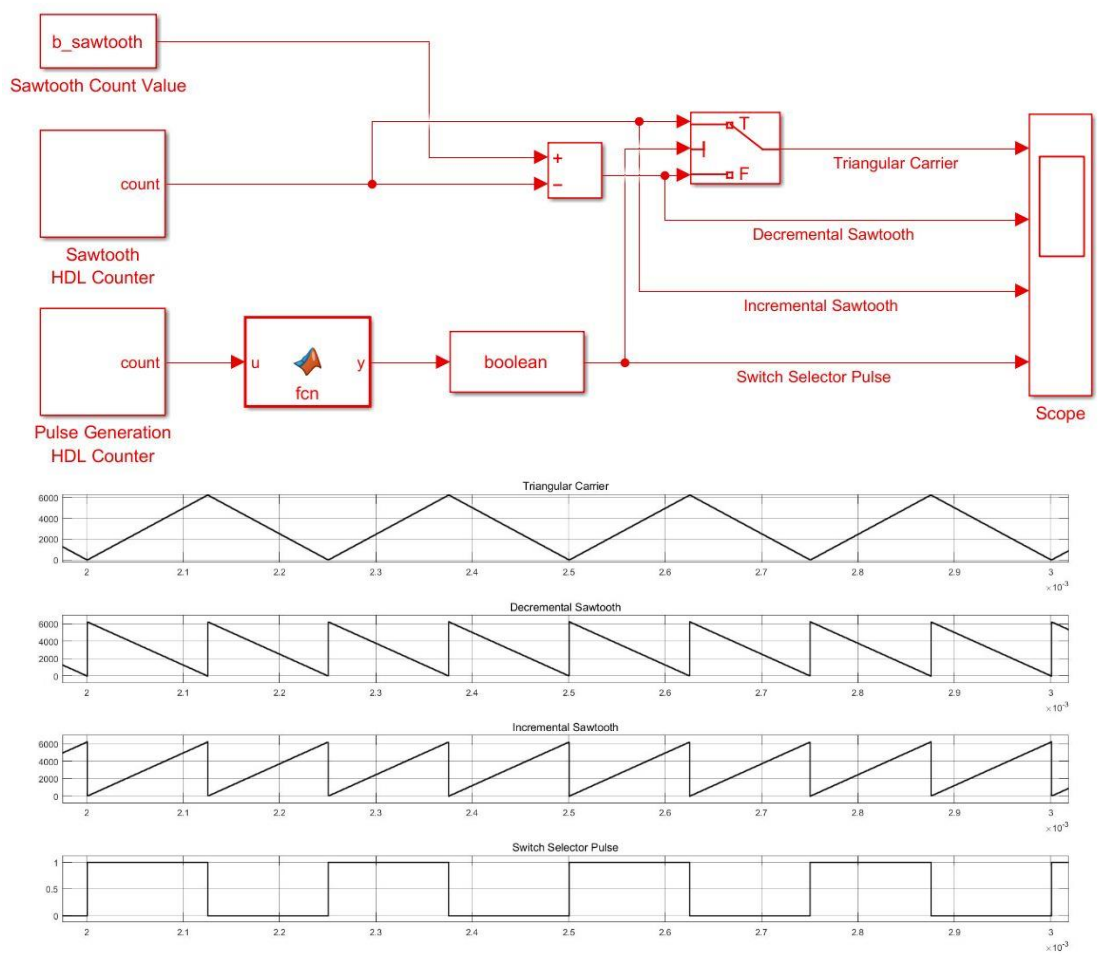


Figure 83 - Simulation model of triangular carrier generation.

A.3.3. Sinusoidal Reference Generation

The sinusoidal references are generated through Look Up Tables (LUT). Address values are inputted into the LUT, via the logic shown in (fig). The LUT use a sawtooth wave as input, where the incremental values of the sawtooth corresponds to the correct output sine value. The addresses are generating by considering the incremental angle theta

$$\begin{aligned} \theta &= \int \omega \cdot dt \\ \Delta\theta &= \omega \cdot \Delta t = 2\pi f \cdot T_s \end{aligned} \tag{172}$$

As the FPGA only comprehends dimensionless data, the values are per – unitized

$$\begin{aligned} \Delta\theta_{pu} \cdot \theta_{base} &= 2\pi f_{pu} \cdot f_{base} \cdot T_s \\ \Delta\theta_{pu} &= \underbrace{\frac{2\pi f_{base} \cdot T_s}{\theta_{base}}}_{C_{frequency}} \cdot f_{pu} \end{aligned} \tag{173}$$

Where the per – unit system is as follows

$$\text{per unit } (pu) = \frac{\text{actual value}}{\text{base value}} \tag{174}$$

By choosing proper base values for the system frequencies of interest and a reasonable sample time ($\theta_{base} = 2\pi$ for sine – waves), the actual values are per – unitized. By choosing the following frequency base value $f_{base} = 50$ [Hz], the following values can be obtained by assuming signed 16 – bit system word lengths (signed n - bit word lengths allow values in the range [$2^{(n-1)} - 1$, $-2^{(n-1)}$]).

frequency [Hz]	per – unit	decimal
100	2	32767
75	1.5	24575
50	1	16383
25	0.5	8192
1	0.02	328

Figure 84 shows the logic of the address generation.

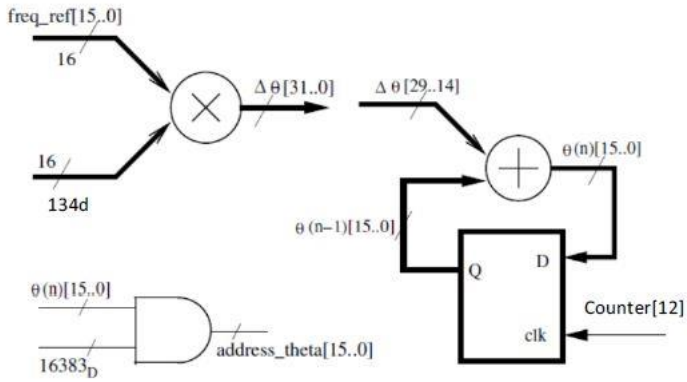


Figure 84 - Address Generation Logic.

Multiplying f_{pu} with the constant $C_{frequency}$ gives the real world per – unit value of $\Delta\theta_{pu}$. By multiplying this by the corresponding decimal value gives $\Delta\theta_{pu}$ in terms of binary representation in real world. Assuming f_{pu} and $C_{frequency}$ to be two 16 bit numbers, multiplying them yields a 32 – bit number. A slice block is used to return a field of consecutive bits from the input, where the numbers correspond to the position of the most significant bit (MSB) and least significant bit (LSB) of the output field. The output word length is computed as (MSB position – LSB position) + 1. The output from the multiplication is sliced down to a 16 – bit number. The number is sliced as to keep to the correct radix point of the output number. The 16 – bit incremental theta address is then inputted to a D – latch which delays the value at the rate given by the clock, thus adding the previous value to itself to increment the address. The address value is then inputted to the LUT which is configured to output the sine – value for the given input address value.

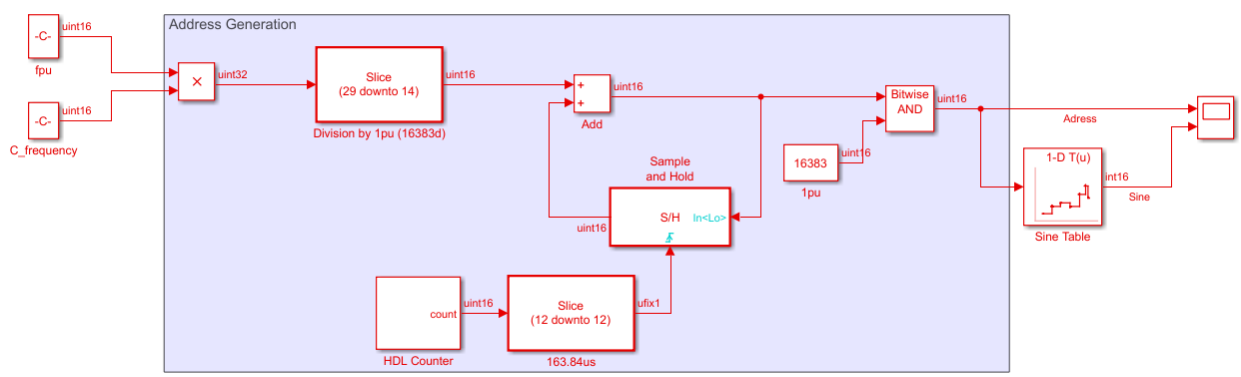


Figure 85 - HDL compatible sine generation.

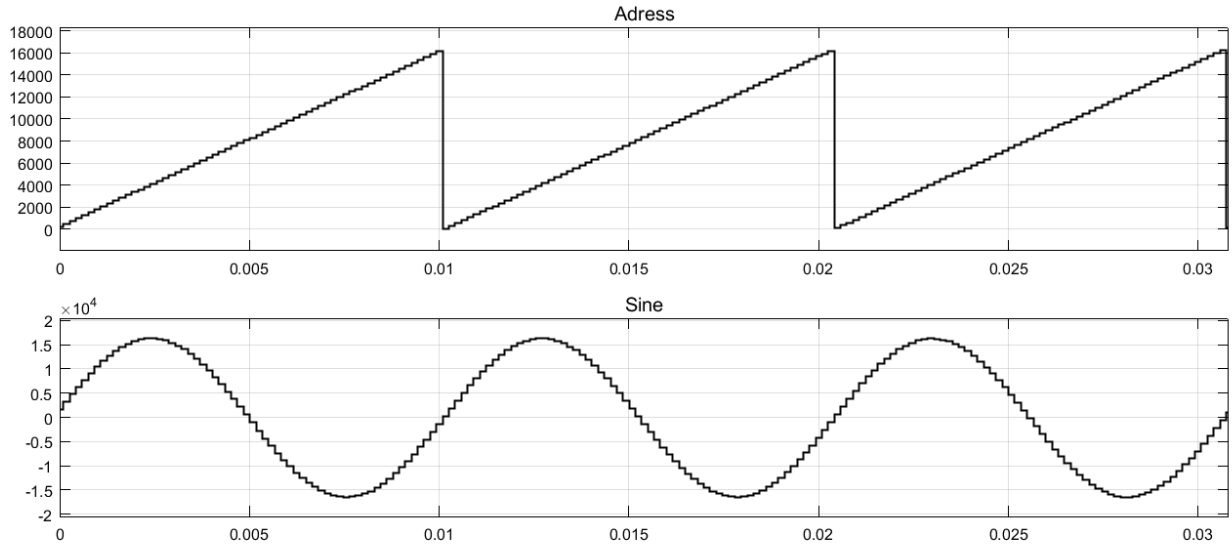


Figure 86 - Simulation run of sine generation.

A.3.4. The Modulator

For this system, the Phase – Disposition (PD) – PWM method is used. The method is explained in *Phase – Disposition PWM* and can be realized as a fixed – point model as shown in Figure 88.

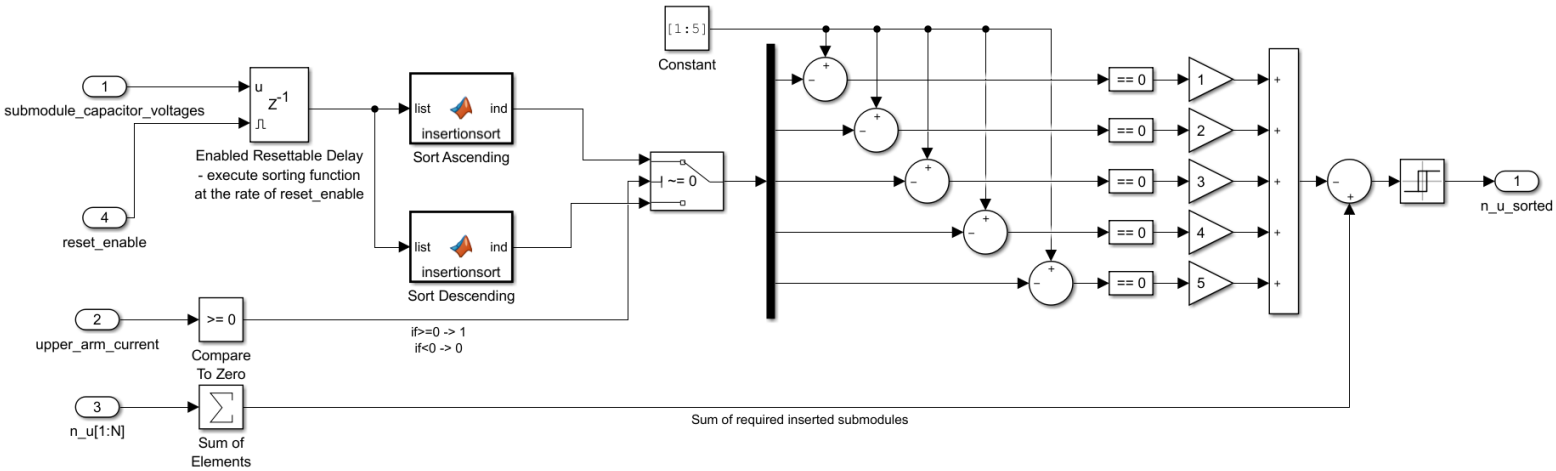


Figure 87 - HDL compatible Simulink model of the voltage sorting algorithm.

A.3.5. The Submodule Capacitor Voltage Sorting

The gate – signals generated for upper and lower arm in Figure 88 are un – balanced and using these insertion indexes will lead to diverging submodule capacitor voltages. One method for balancing the voltages is given in *Submodule Capacitor Voltage Sorting*. This algorithm can be implemented in hardware as the Simulink model is shown in Figure 87.

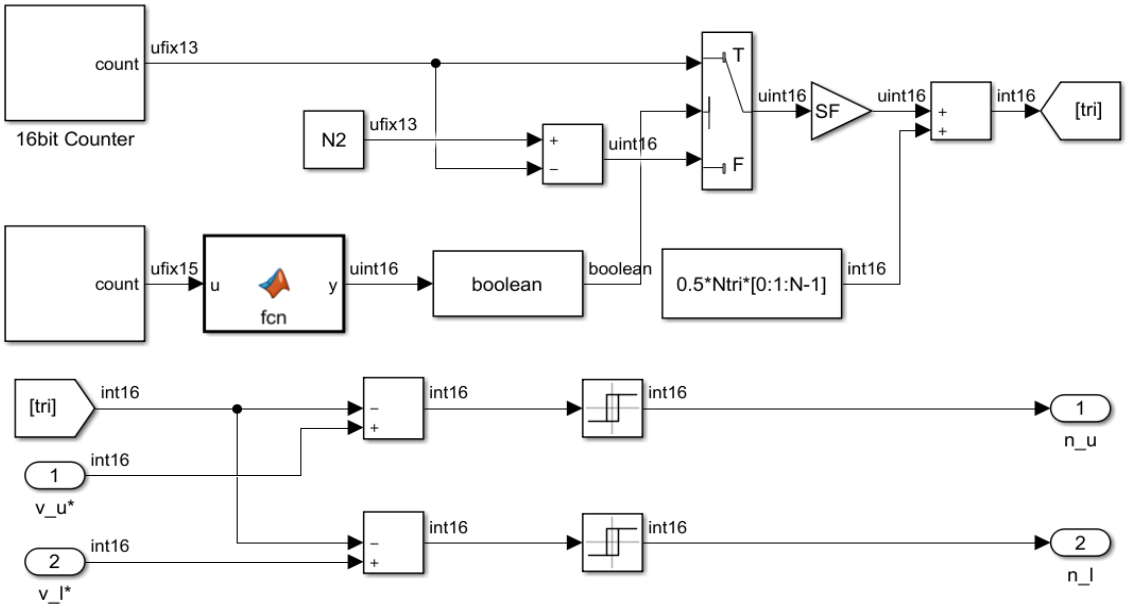


Figure 88 – HDL compatible PD - PWM signal generation for upper n_u and lower n_l arm.

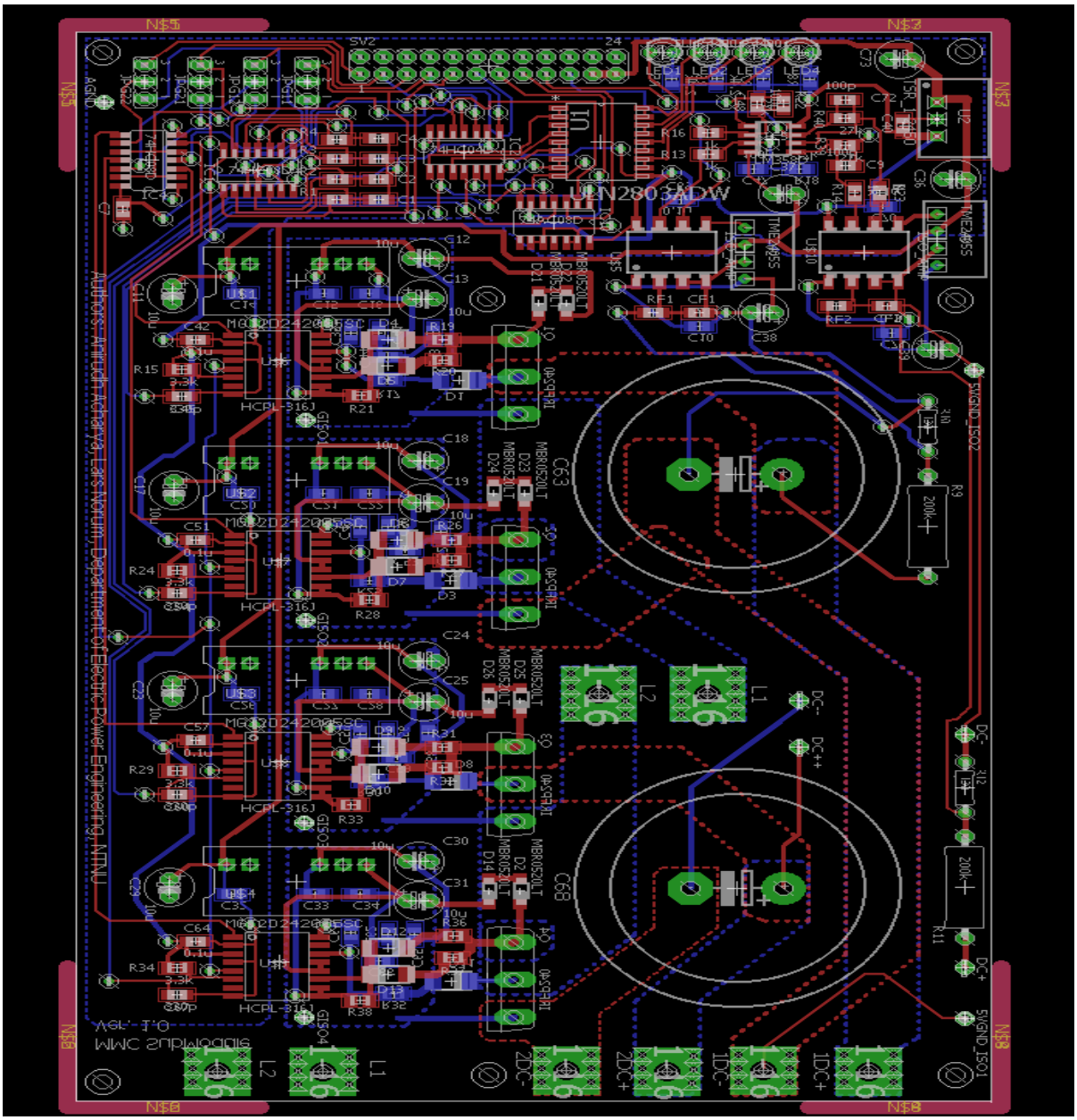
B. Eagle CAD Schematics

The following paragraphs shows schematic layouts comprising the designed electric circuitry for signal processing, IC applications, the power modules of the MMC and interfacing between digital control and the analog plant. The schematics and board file is generated in Eagle CAD.

B.1. Submodule Board and Interface Card Eagle Schematics

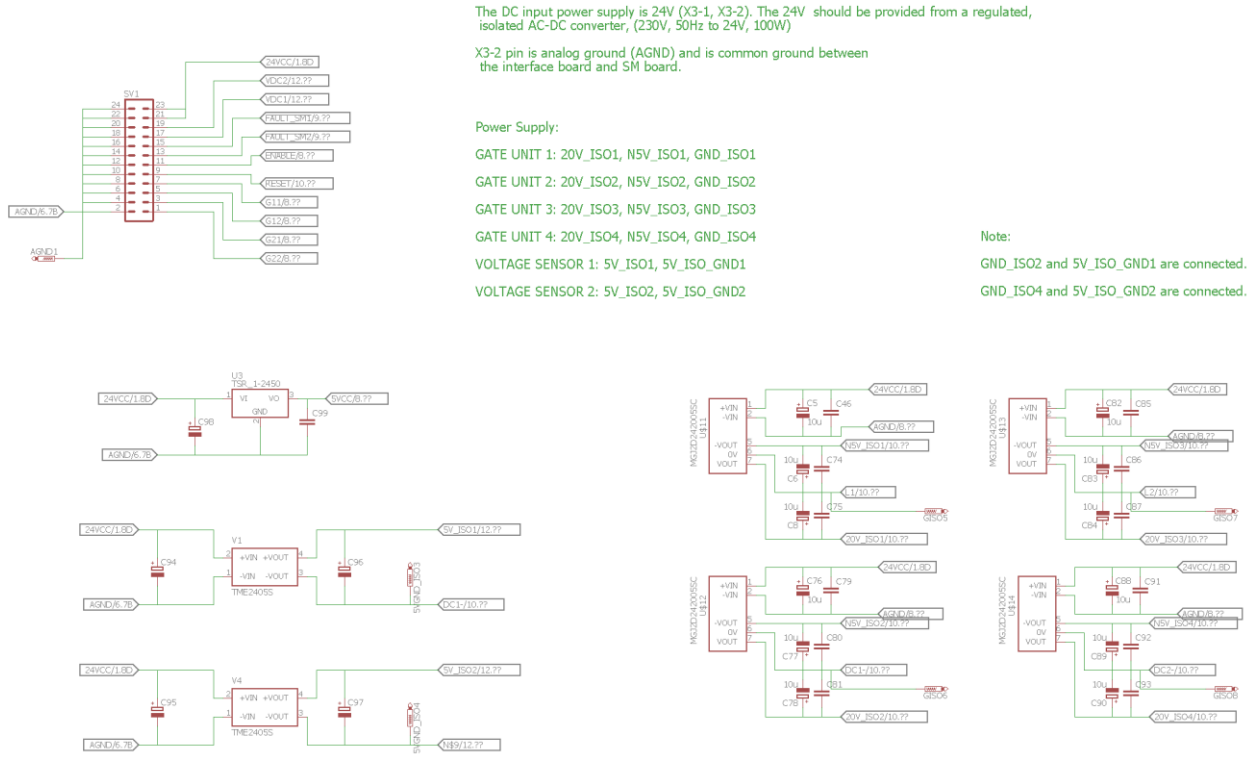
B.1.1. Submodule Board File

The final board file shown is made by professor Lars Norum and Anirudh Acharya and at Department of Electric Power Engineering NTNU. Contribution to the board file has been made by the author.

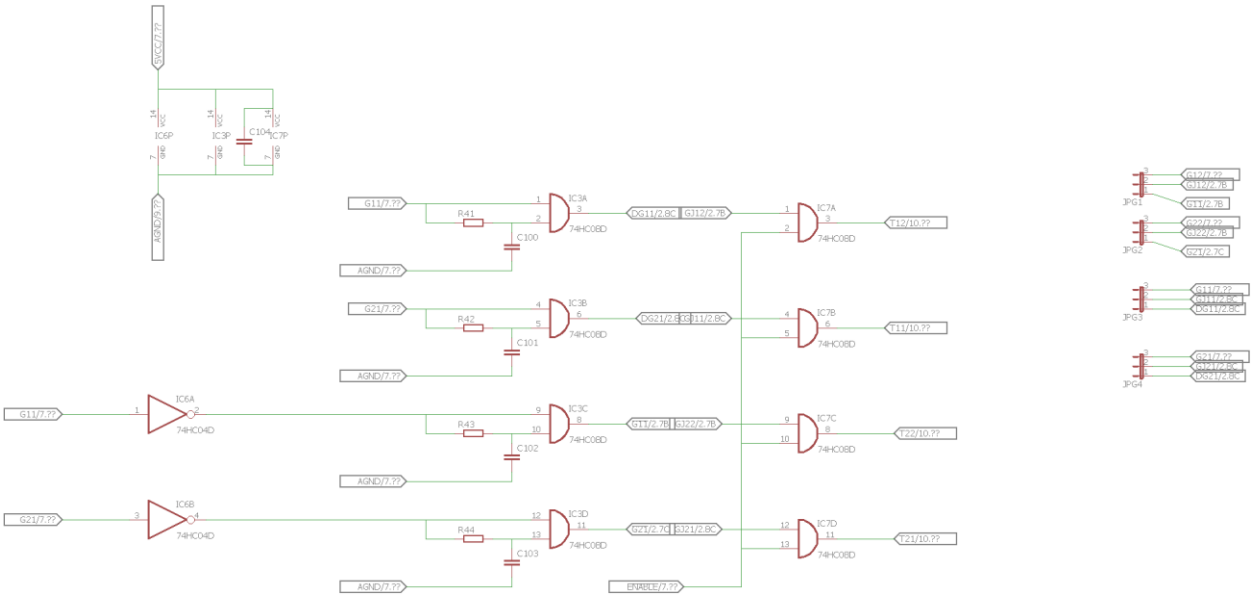


B.1.2. Submodule Schematics

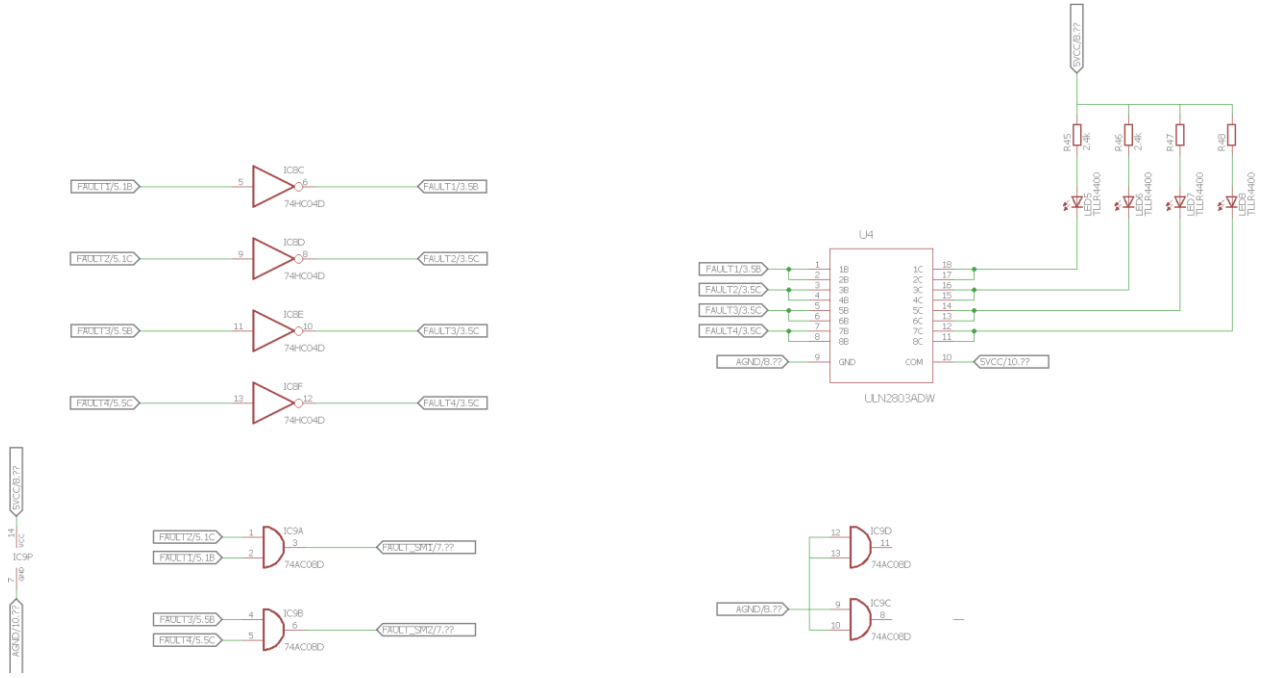
B.1.2.1. Sheet 1



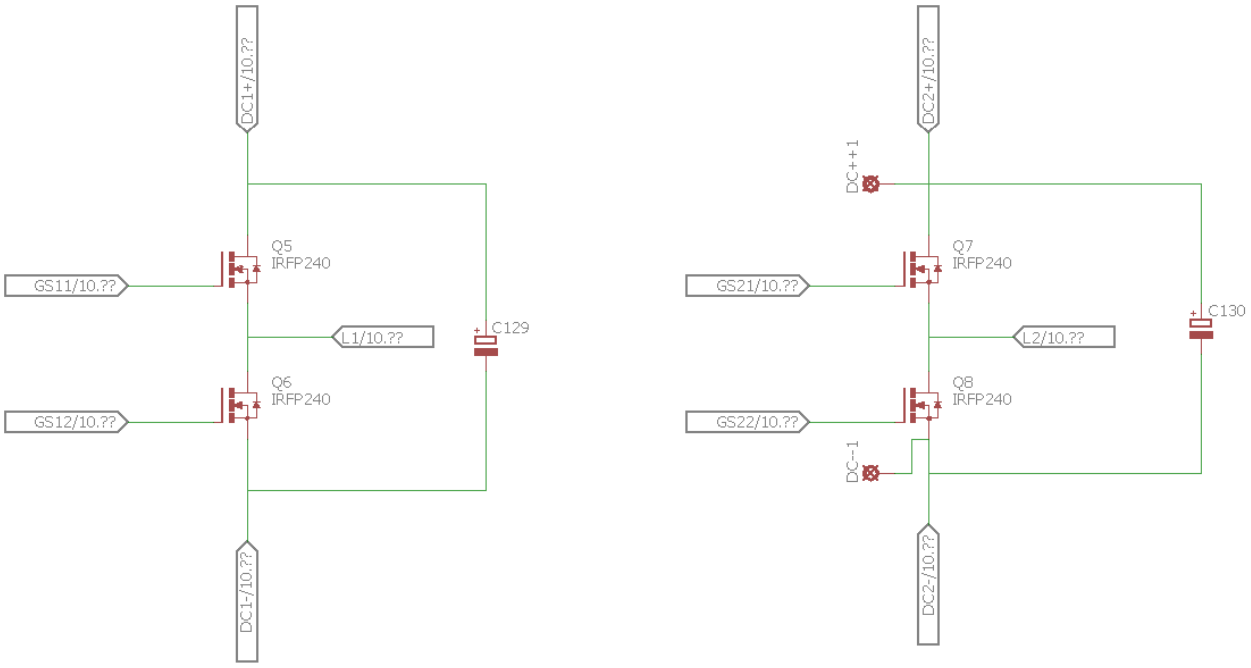
B.1.2.2. Sheet 2



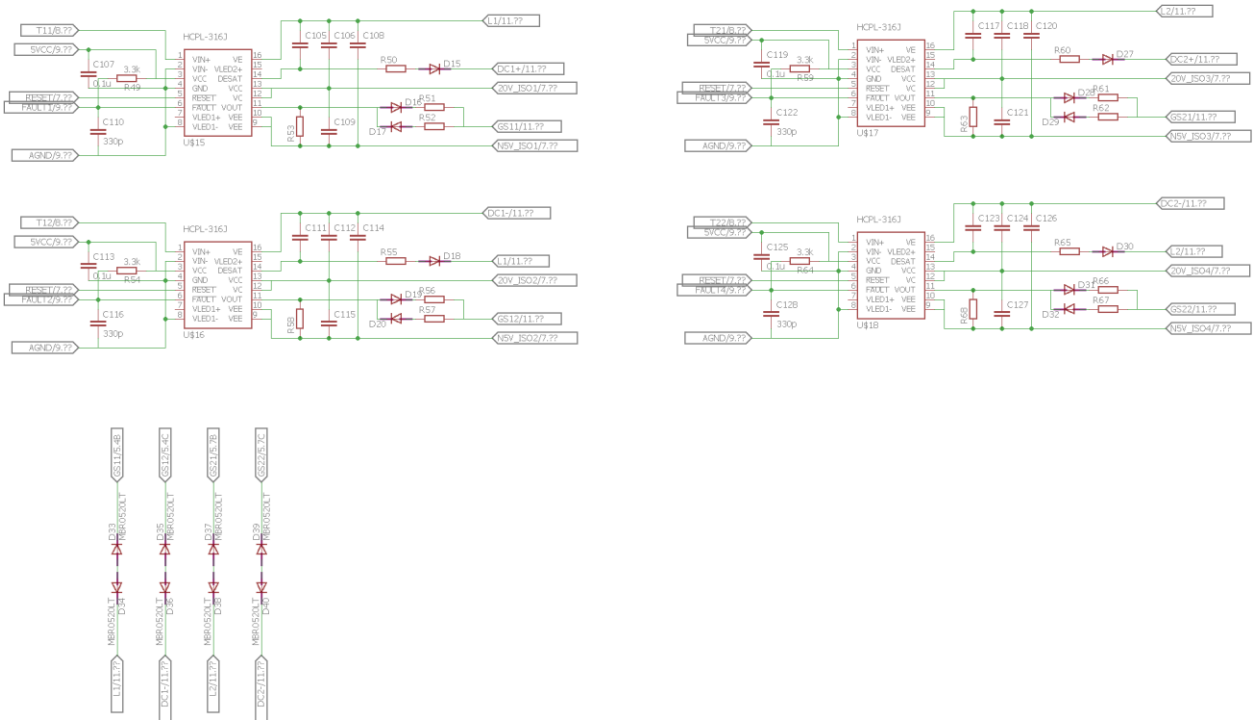
B.1.2.3. Sheet 3



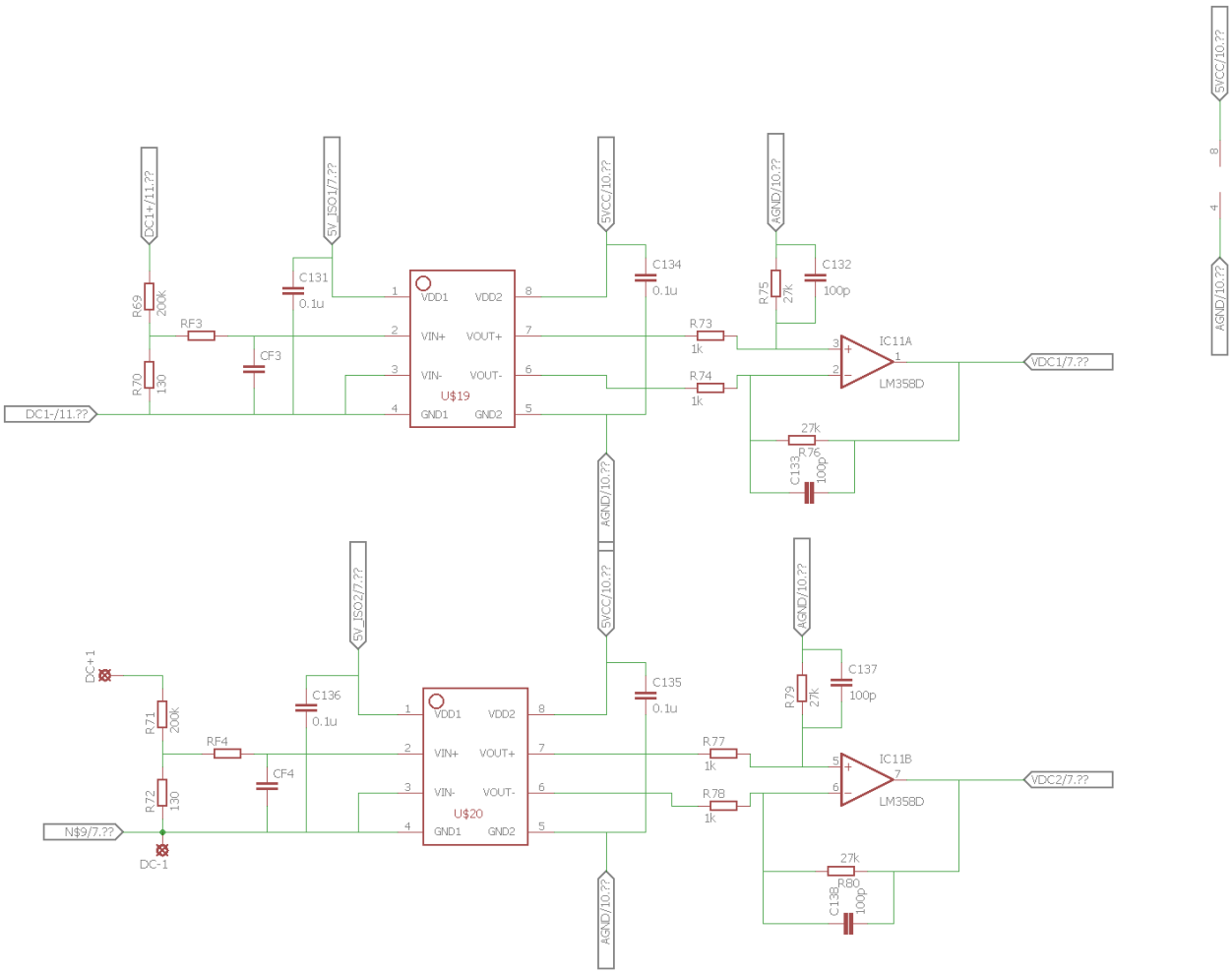
B.1.2.4. Sheet 4



B.1.2.5. Sheet 5

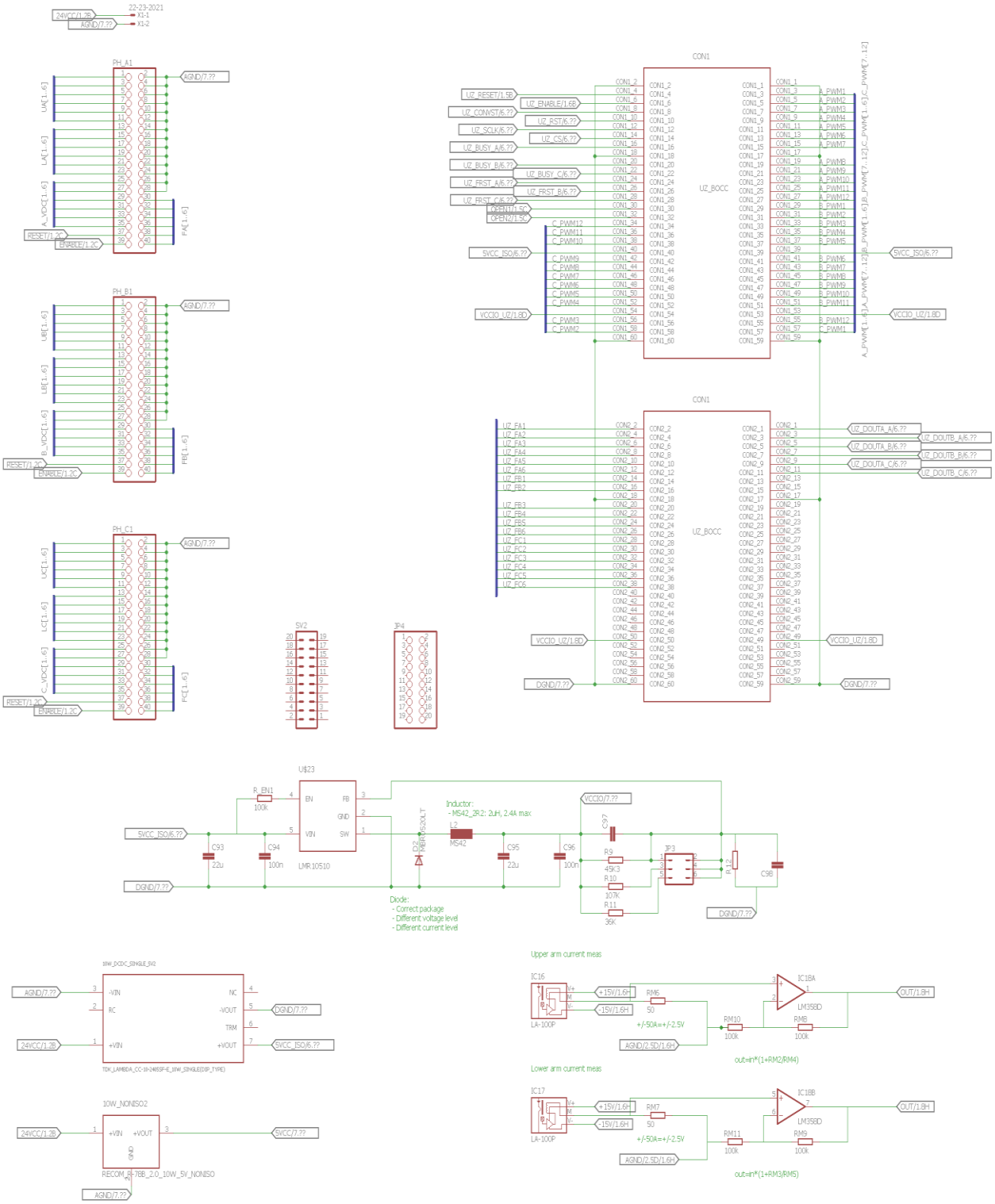


B.1.2.6. Sheet 6

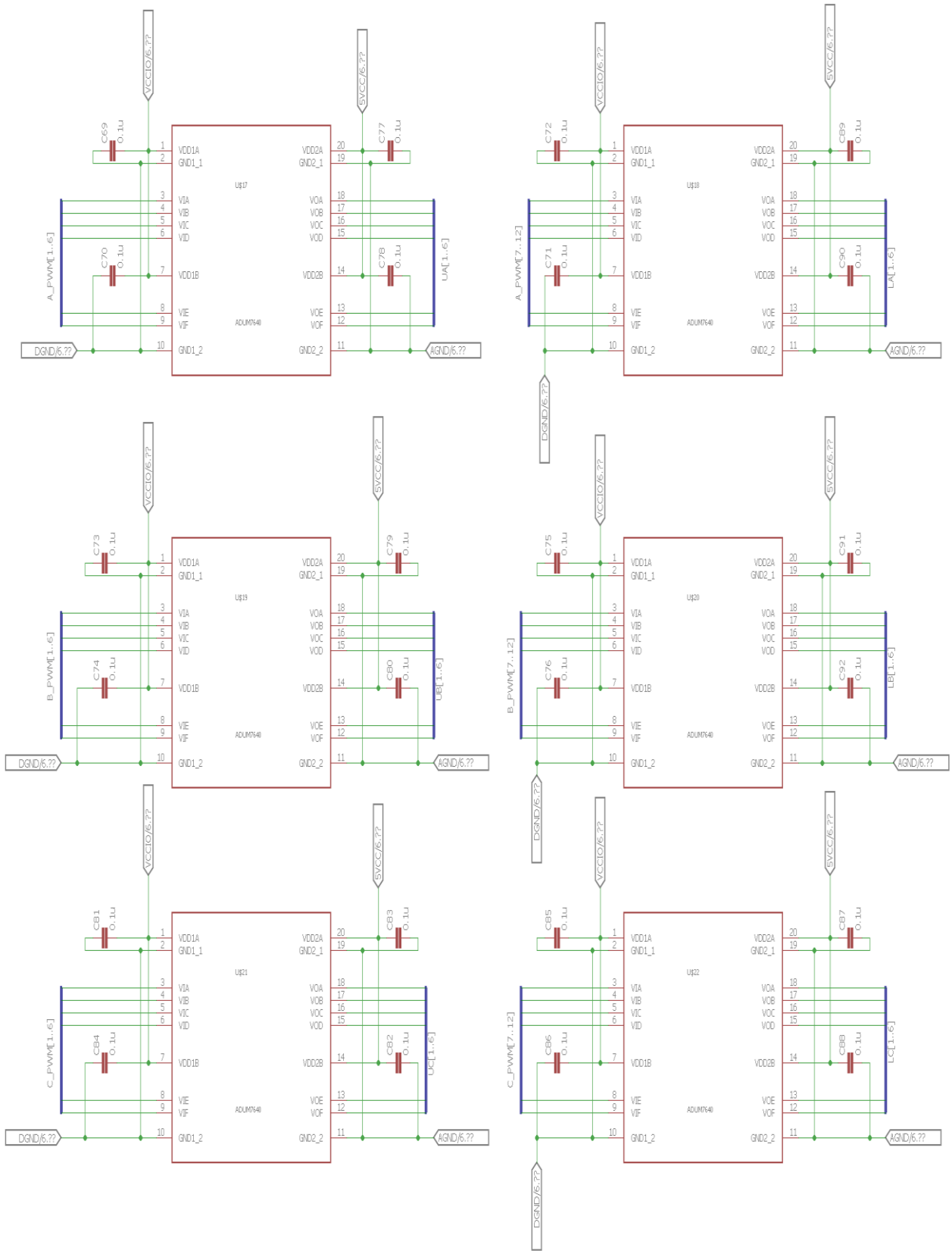


B.1.3. Interface Board

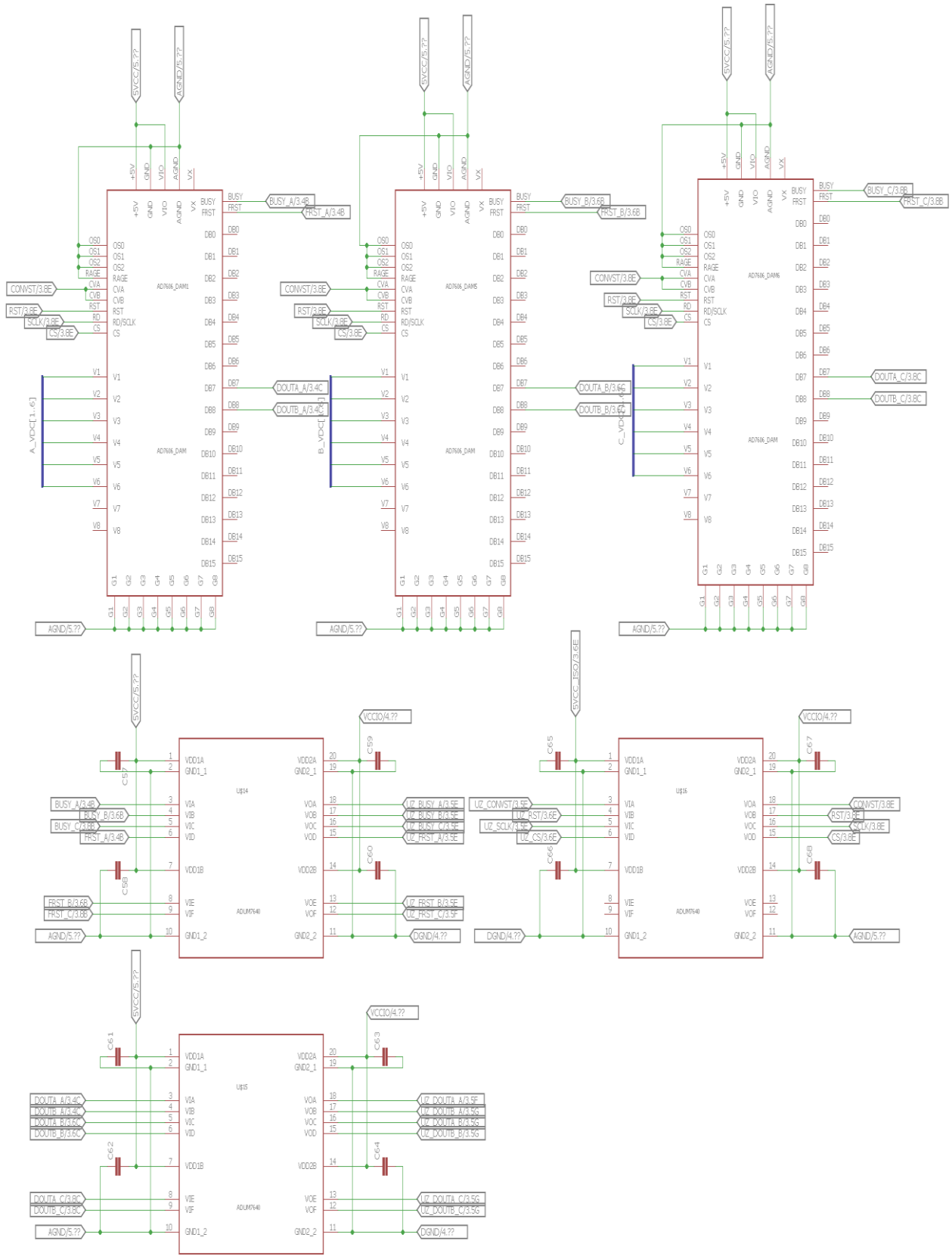
B.1.3.1. Sheet 1



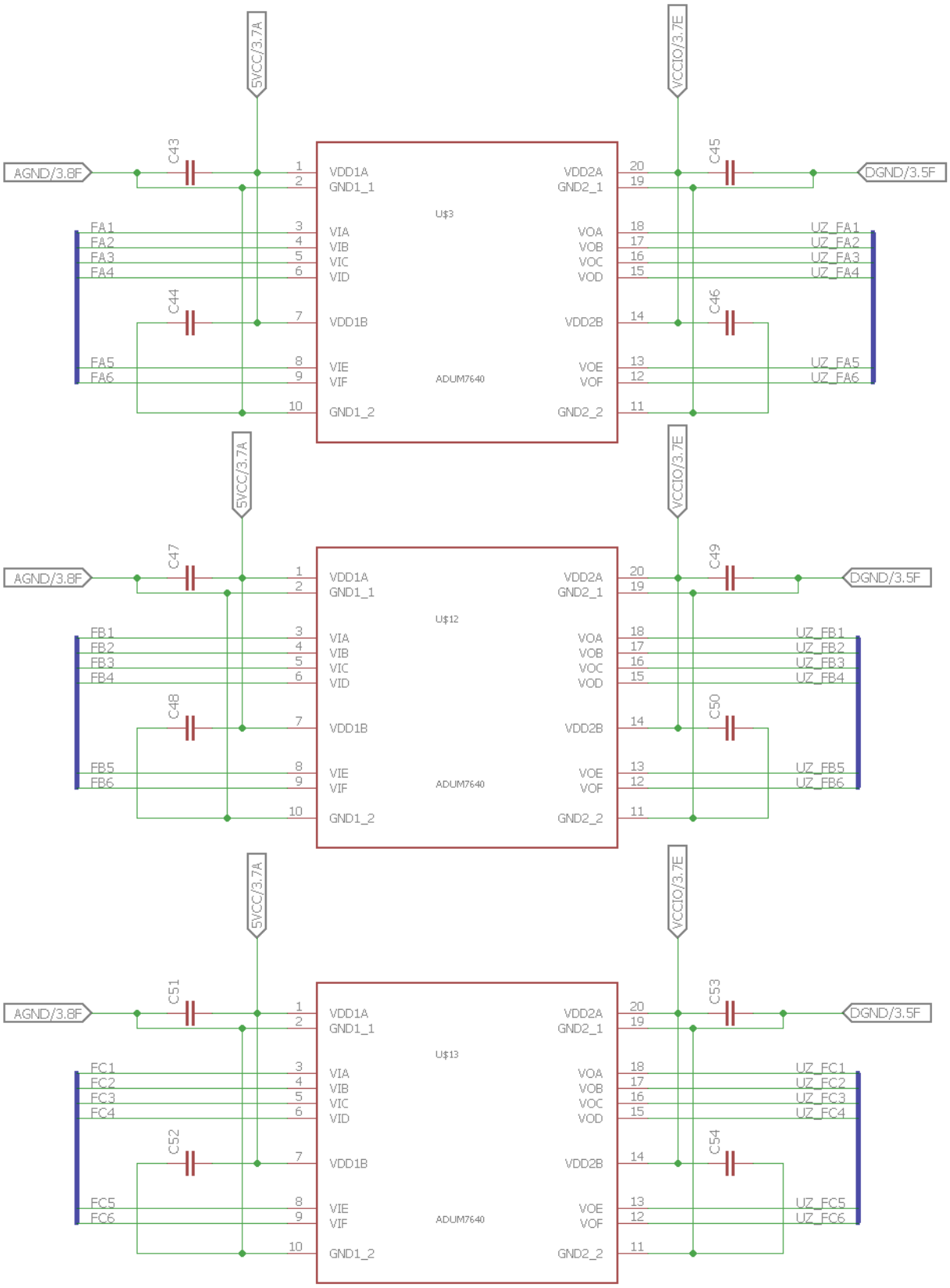
B.1.3.2. Sheet 2


VCCIO/6.??
SVCC/6.??
DGND/6.??
AGND/6.??
A_PWM[1..6]
A_PWM[7..12]
B_PWM[1..6]
B_PWM[7..12]
C_PWM[1..6]
C_PWM[7..12]
UA[1..6]
LA[1..6]
UB[1..6]
LB[1..6]
UC[1..6]
LC[1..6]
U$17
U$18
U$19
U$20
U$21
U$22
ADUM7640
VDD1A
GND1_1
VIA
VIB
VIC
VID
VDD1B
VIE
VIF
GND1_2
VDD2A
GND2_1
VOA
VOB
VOC
VOD
VDD2B
VOE
VOF
GND2_2
C69
C70
C71
C72
C73
C74
C75
C76
C77
C78
C79
C80
C81
C82
C83
C84
C85
C86
C87
C88
C89
C90
C91
C92
0.1u

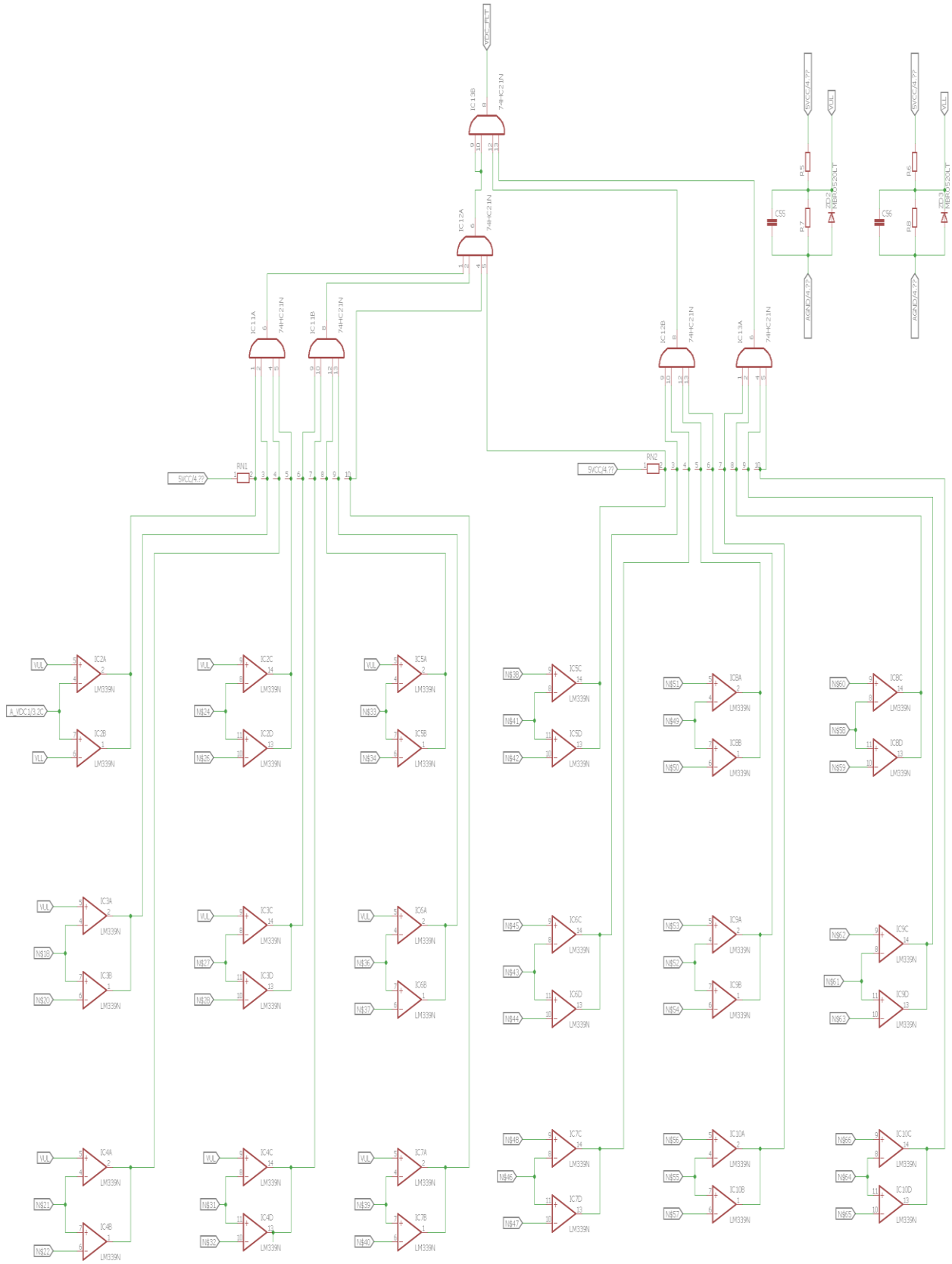
B.1.3.3. Sheet 3


5VCC/5.??
AGND/5.??
AD7606_DAM1
AD7606_DAM5
AD7606_DAM6
AD7606_DAM
A_VDC[1..6]
B_VDC[1..6]
C_VDC[1..6]
CONVST/3.8E
RST/3.8E
SCLK/3.8E
CS/3.8E
BUSY_A/3.4B
FRST_A/3.4B
BUSY_B/3.6B
FRST_B/3.6B
BUSY_C/3.8B
FRST_C/3.8B
DOUTA_A/3.4C
DOUTB_A/3.4C
DOUTA_B/3.6C
DOUTB_B/3.6C
DOUTA_C/3.8C
DOUTB_C/3.8C
5VCC_ISO/3.6E
VCCIO/4.??
DGND/4.??
U$14
U$16
U$15
ADUM7640
C57
C58
C59
C60
C65
C66
C67
C68
C61
C62
C63
C64
UZ_BUSY_A/3.5E
UZ_BUSY_B/3.5E
UZ_BUSY_C/3.5E
UZ_FRST_A/3.5E
UZ_FRST_B/3.5E
UZ_FRST_C/3.5F
UZ_CONVST/3.5E
UZ_RST/3.6E
UZ_SCLK/3.5E
UZ_CS/3.6E
UZ_DOUTA_A/3.5E
UZ_DOUTB_A/3.5G
UZ_DOUTA_B/3.5G
UZ_DOUTB_B/3.5G
UZ_DOUTA_C/3.5G
UZ_DOUTB_C/3.5G

B.1.3.4. Sheet 4


5VCC/3.7A
VCCIO/3.7E
AGND/3.8F
DGND/3.5F
C43
C44
C45
C46
U$3
ADUM7640
VDD1A
GND1_1
VIA
VIB
VIC
VID
VDD1B
VIE
VIF
GND1_2
VDD2A
GND2_1
VOA
VOB
VOC
VOD
VDD2B
VOE
VOF
GND2_2
FA1
FA2
FA3
FA4
FA5
FA6
UZ_FA1
UZ_FA2
UZ_FA3
UZ_FA4
UZ_FA5
UZ_FA6
C47
C48
C49
C50
U$12
FB1
FB2
FB3
FB4
FB5
FB6
UZ_FB1
UZ_FB2
UZ_FB3
UZ_FB4
UZ_FB5
UZ_FB6
C51
C52
C53
C54
U$13
FC1
FC2
FC3
FC4
FC5
FC6
UZ_FC1
UZ_FC2
UZ_FC3
UZ_FC4
UZ_FC5
UZ_FC6

B.1.3.4.5. Sheet 5



C. IC Specifications

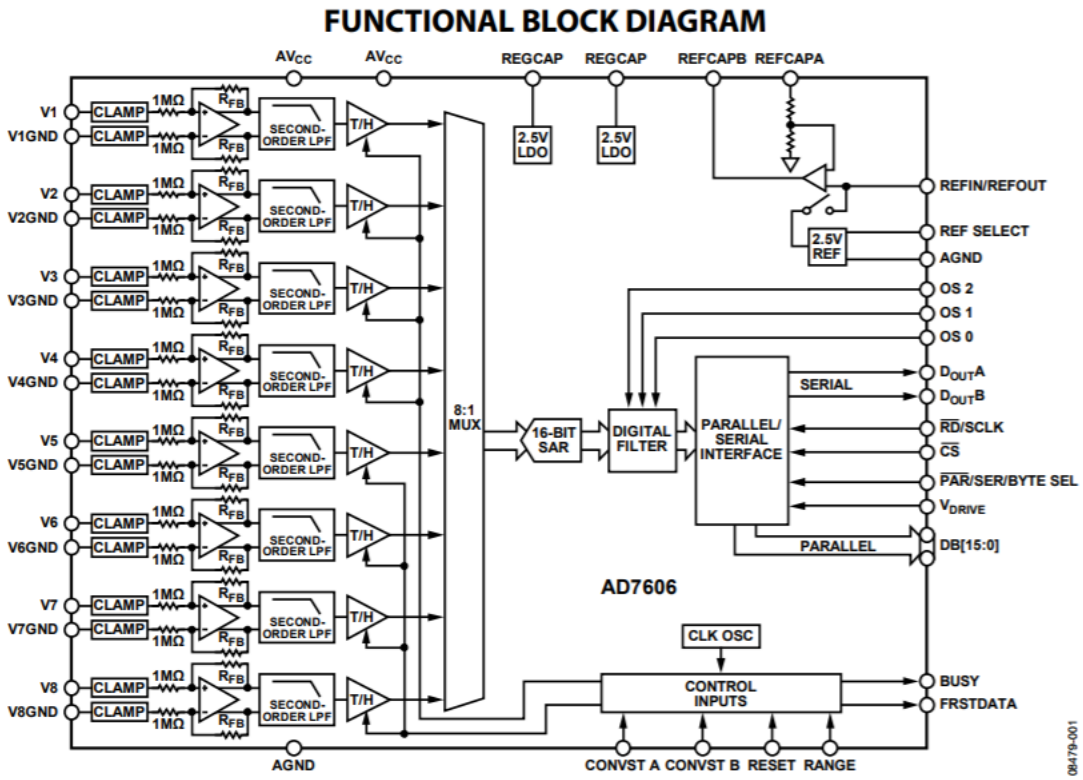


Figure 89 - Functional block diagram of the AD7606 ADC [3].

Electrical data							
I_{PN}	Primary nominal rms current			50			A
I_{PM}	Primary current, measuring range			0 .. ±70			A
R_M	Measuring resistance @		T_A = 70 °C		T_A = 85 °C		
			$R_{M\,min}$	$R_{M\,max}$	$R_{M\,min}$	$R_{M\,max}$	
	with ±12 V	@ ±50 A_{max}	10	100	60	95	Ω
		@ ±70 A_{max}	10	50	60 [1)]	60 [1)]	Ω
	with ±15 V	@ ±50 A_{max}	50	160	135	155	Ω
		@ ±70 A_{max}	50	90	135 [2)]	135 [2)]	Ω
I_{SN}	Secondary nominal rms current			50			mA
K_N	Conversion ratio			1 : 1000			
U_C	Supply voltage (±5 %)			±12 .. 15			V
I_C	Current consumption			10 (@ ±15 V) + I_S			mA
Accuracy - Dynamic performance data							
X	Accuracy @ I_{PN}, T_A = 25 °C	@ ±15 V (±5 %)		±0.65			%
		@ ±12 .. 15 V (±5 %)		±0.90			%
ε_L	Linearity error			<0.15			%
				Typ	Max		
I_O	Offset current @ I_P = 0, T_A = 25 °C				±0.2		mA
I_{OM}	Magnetic offset current [3)] @ I_P = 0 and specified R_M, after an overload of 3 × I_{PN}				±0.3		mA
I_{OT}	Temperature variation of I_O	-25 °C .. +85 °C		±0.1	±0.6		mA
		-40 °C .. -25 °C		±0.2	±1.0		mA
t_{ra}	Reaction time			<500			ns
t_r	Step response time to 90 % of I_{PN}			<1			µs
d*i*/d*t*	d*i*/d*t* accurately followed			>200			A/µs
BW	Frequency bandwidth (-1 dB)			DC .. 200			kHz

Figure 90 - Electrical characteristics and dynamic performance of the LA - 55P current transducer [1].

D. Mathematical Expressions

D.1. Harmonic Content of Phase Voltage using PD – PWM

The harmonic components generated in the phase voltage due to PD – PWM is derived for 3 – and 5 level multilevel inverters in [14]. The harmonic spectrum for a 5 – level multilevel inverter can be analytically described as [14]

$$v_a(t) = \frac{MV_{DC}}{2}\cos(\omega_0 t) + \frac{V_{DC}}{N\pi}\sum_{m=1}^{\infty}\frac{1}{2m}\sum_{n=-\infty}^{\infty} J_{2n+1}(4m\pi M)\cos(n\pi)\cos(2m\omega_c t + (2n+1)\omega_0 t)$$

$$+\frac{2V_{DC}}{N\pi^2}\sum_{m=1}^{\infty}\frac{1}{2m-1}\sum_{n=-\infty}^{\infty}\sum_{k=1}^{\infty}\left(\begin{array}{c} J_{2k-1}((2m-1)2\pi M)\cos(k\pi)\cdot \\ \left[\begin{array}{c} \dfrac{(\cos((n-k)\pi)+2\sin((2k-1-2n)\varphi))}{(2k-1-2n)}+ \\ \dfrac{(\cos((n+k)\pi)+2\sin((2k-1+2n)\varphi))}{(2k-1+2n)} \end{array}\right]\cdot(\cos((2m-1)\omega_c t)+2n\omega_0 t) \end{array}\right)$$

$$+\frac{4V_{DC}}{N\pi^2}\sum_{m=1}^{\infty}\frac{1}{2m-1}\sum_{m=1}^{\infty}\frac{1}{2k-1}J_{2k-1}((2m-1)2\pi M)(1+2\sin((2k-1)\varphi)\cos(k\pi))\cos((2m-1)\omega_c t)$$

$$\varphi = \arccos\left(\frac{1}{2M}\right)$$

(175)

Showing even sideband harmonics $2n\omega_0 t$ be present at odd carrier groups $(2m-1)\omega_c t$ and odd sideband harmonics $(2n+1)\omega_0 t$ present at even carrier groups $2m\omega_c t$.