Tampere University

Tuukka Salminen

REAL-TIME IMPLEMENTATION OF A DIRECT MODEL PREDICTIVE CONTROL ALGORITHM ON AN INDUSTRIAL CONTROL PLATFORM

Master of Science Thesis
Faculty of Information Technology and Communication Sciences
Examiners: Assoc. Prof. Petros Karamanakos
Dr. Mattia Rossi
Dr. Arto Sankala
May 2023

ABSTRACT

Tuukka Salminen: Real-Time implementation of a direct model predictive control algorithm on an industrial control platform
Master of Science Thesis
Tampere University
Master's degree programme in electrical engineering
May 2023

In recent years, model predictive control (MPC) has been an increasingly researched subject in power electronics. Its ability to handle complex dynamics of higher-order systems, apply constraints on the state, input, and output variables, and simultaneously control multiple control objectives within a single control cycle makes it well suited to power electronics. The challenge of real-time implementation of model predictive control algorithms stems from its high computational burden. The algorithm should be able to solve the optimization problem within tens of microseconds to ensure good performance.

This thesis presents and implements a direct MPC control scheme for a grid-connected two-level voltage source converter with an LCL filter on industrial control hardware. The control objective of the algorithm is to control currents and voltages in the LCL filter simultaneously. Due to control hardware limitations, the LCL filter's grid-side current and capacitor voltage is estimated using an observer. The feasible set of switch positions the MPC considers was reduced through sector identification due to the high computational burden. The results are validated using hardware-in-loop real-time simulations.

The controller shows good tracking of state references during steady state. MPC with a single-step horizon could handle the MPC algorithm's computational burden within a 50μs time window. Two-step horizon MPC could not reach the desired execution time and required more time to enumerate through feasible switch positions. The results show that reducing the feasible set of considered switch positions affects the execution time of the algorithm considerably with horizons larger than one.

Keywords: model predictive control, grid-connected, LCL filter

TIIVISTELMÄ

Tuukka Salminen: Suoran mallipohjaisen ennustavan ohjausalgoritmin käyttöönotto teolliselle ohjausalustalle
Diplomityö
Tampereen yliopisto
Sähkötekniikan diplomi-insinööri tutkinto-ohjelma
Toukokuu 2023

MPC (eng. model predictive control) on yhä enemmän tutkittu aihe tehoelektroniikkaan liittyvässä käytössä viime vuosien aikana. Sen kyky hallita korkeamman asteen järjestelmien dynamiikkaa, rajoittaa tila-, sisääntulo- ja ulostulomuuttujia sekä hallita useita ohjaustavoitteita yhden ohjausjakson aikana tekevät siitä hyvin soveltuvan tehoelektroniikan laitteisiin. MPC:n reaaliaikaisen käyttöönoton haaste on sen huomattavan suuri laskentakuorma. Algoritmin tulisi kyetä ratkaisemaan optimointiongelma kymmenissä mikrosekunteissa taatakseen hyvän suorituskyvyn.

Tässä työssä esitetään ja toteutetaan suora MPC-säätöalgoritmi kaksitasoiselle LCL-suodattimen kautta verkkoon kiinnitetylle teolliselle ohjausalustalle. Algoritmin ohjaustavoitteena on LCL-suodattimen virtojen ja jännitteiden samanaikainen ohjaus. Ohjausalustan rajoitteiden vuoksi LCL suodattimen verkon puoleinen virta ja kondensaattorin jännite on estimoitu laskennallisella arvioinnilla. Laskentatehon rajoittamiseksi algoritmin harkitsemia kytkinohjeita vähennetään sektoritunnistuksella. Tulokset validoidaan hardware-in-the-loop reaaliaikaisella simulaatiolla.

Ohjausalgoritmi onnistuu seuraamaan tilareferenssejä vakaassa tilassa. Yhden aikajakson horisontin MPC onnistui ylläpitämään algoritmin laskenta-ajan alle 50μs aikaikkunassa. Kahden aikajakson horisontin MPC ei pystynyt saavuttamaan haluttua laskenta-aikaa, ja vaati enemmän laskenta-aikaa läpikäymään kaikki mahdolliset kytkinohjeet. Tulokset osoittavat, että mahdollisten kytkinohjeiden rajoittaminen vaikuttaa algoritmin laskenta-aikaan huomattavasti, kun ennustushorisontin pituus on enemmän kuin yksi.

Avainsanat: mallipohjainen ennustava ohjaus, verkkokytkentä, LCL suodatin

PREFACE

This Master of Science thesis was written in Danfoss Drives in Tampere. The examiners of the thesis are Petros Karamanakos and Mattia Rossi from Tampere University and Arto Sankala from Danfoss Drives

Foremost, I would like to thank Arto Sankala and Mattia Rossi for their guidance in implementation of the algorithm on the control platform. Their insight helped me get over many problems during the implementation and testing phase. I would also like to thank my other colleagues, who during the thesis provided me with helpful insights.

I would like to thank Petros Karamanakos for his guidance during the writing of the thesis. His guidance and feedback greatly helped during the writing process.

Tampere, 3rd May 2023

Tuukka Salminen

CONTENTS

LIST OF FIGURES

SYMBOLS

C_f	filter capacitance
J	objective function output
L_1	converter-side inductance
L_2	grid-side inductance
N_p	length of prediction horizon
T_s	sampling interval
V_x	voltage vector
X_C	LCL filter capacitor reactance
X_{fc}	LCL filter converter-side reactance
X_{fg}	LCL filter grid-side reactance
$\tilde{\boldsymbol{K}}(\phi)$	reduced Park transformation matrix
$\tilde{\boldsymbol{K}}$	reduced Clarke transformation matrix
$\tilde{\boldsymbol{K}}^{-}1(\phi)$	reduced inverse Park transformation matrix
$\tilde{\boldsymbol{K}}^{-1}$	reduced inverse Clarke transformation matrix
$\boldsymbol{A}$	discrete-time system matrix
$\boldsymbol{B}$	discrete-time input matrix
$\boldsymbol{C}$	output matrix
$\boldsymbol{F}$	continuous-time system matrix
$\boldsymbol{G}$,$\boldsymbol{G_1}$, $\boldsymbol{G_2}$	continuous-time input matrix
$\boldsymbol{I}$	identity matrix
$\boldsymbol{K}$	Clarke transformation matrix
$\boldsymbol{Q}$	penalty matrix
$\boldsymbol{R}(\phi)$	rotational matrix
$\boldsymbol{U}_{opt}$	optimal switching plan
$\boldsymbol{Y}_{ref}$	output reference trajectory
$\boldsymbol{\xi}_{ab0}$	$\alpha\beta0$-quantity
$\boldsymbol{\xi}_{abc}$	three-phase quantity

$\boldsymbol{i}_{c,ref}$	converter current reference vector
$\boldsymbol{i}_{g,ref}$	grid current reference vector
$\boldsymbol{u}$	input vector
$\boldsymbol{u}_{abc}$	three-phase switch positions
$\boldsymbol{v}_{c,ref}$	converter output voltage reference vector
$\boldsymbol{v}_{f,ref}$	filter capacitor reference vector
$\boldsymbol{x}$	state vector
$\boldsymbol{y}$	output vector
$\boldsymbol{y}_{ref}$	output reference vector
λ_u	switching penalty
$\mathbb{N}$	natural numbers
$\mathbb{U}$	admissible control actions
ω_{fr}	dq-reference frame angular speed
ω_g	angular frequency of grid voltage
e	matrix exponential
f_g	grid frequency
f_B	base frequency
f_{res}	filter resonance frequency
f_{sw}	switching frequency
i_c	converter current
i_g	grid current
k	time step
n_u	number of inputs
n_x	number of states
n_y	number of outputs
u_x	switch position
v_f	LCL filter capacitor voltage
v_g	grid Voltage
v_{dc}	dc-link voltage
w_b	base angular frequency

ACRONYMS

CPU	central processing unit
FCS	Finite control set
FPGA	field-programmable gate array
MIMO	multiple-input multiple-output
MPC	Model predictive control
p.u.	per unit
PCC	point of common coupling
PI	proportinal-integral
PLL	phase-locked-loop
PWM	pulse-width modulation
SoC	system-on-chip
THD	total harmonic distortion
VSC	voltage source converter

1. INTRODUCTION

Power electronics converters deal with converting and controlling electrical power by controlling a set of semiconductor switches [1]. This conversion and transfer of power can happen at very high efficiencies, but for high-powered applications, even slight inefficiency may cause significant heating losses within the semiconductor switches. One possible method to reduce these losses is to reduce the switching frequency of the semiconductor devices. However, the lowered switching frequency will cause the power quality fed to the load to deteriorate. Thus, a compromise between losses and power quality must be found.

Model predictive control (MPC) [2] is a control method finely suited to manage the tradeoff between switching losses and power quality. Finite control set MPC (FCS-MPC), also known as direct model predictive control with reference tracking, is a control method that directly controls the switch states of semiconductor switches [3]. This means that the control algorithm addresses the modulation and current control of the converter in a single computational stage, making it a promising alternative to traditional pulse-width modulation (PWM) and proportional-integral (PI) control. One of the main challenges in the real-time implementation of model predictive control algorithms is handling the potentially high computational burden with suitably low sampling interval [3]. To this end, analyzing the computational power of the chosen control platform and identifying the delays present in the system is critical.

This thesis introduces the real-time implementation of a direct MPC algorithm that controls the converter switches directly without a modulator. The algorithm controls a grid-connected two-level converter with an LCL filter. The algorithm is implemented using a model-based design on a commercial industrial control platform. Due to the existing architecture of the control platform, the MPC algorithm is implemented on a CPU. Typically FCS-MPC algorithms are favored on FPGA-based systems due to their ability to perform calculations in a pipelined and parallelized manner [4],[5]. This thesis focuses on testing the computational capability and limitations of the CPU on an industrial control platform.

Only a short prediction horizon approach to MPC is implemented to lessen the computational effort required. Hardware-in-the-loop system by Typhoon HIL is used to test the FCS-MPC algorithm on the control platform. Delays and execution times of FCS-MPC and supporting control software components are identified through the execution time profiling method implemented on the control hardware. Based on the results, future research direction will be determined.

This thesis relates to a joint research project between Danfoss Drives and Tampere University called Optimal Control for Maximizing the Effectiveness of Power Electronic Systems (OPT4MAX). The research project aims to implement an FCS-MPC algorithm with power loss constraints on a Danfoss ic7-Automation system module. This thesis provides preliminary information about system delays, performance, and control architecture, which can be used in the research project.

The thesis consists of seven chapters. Chapter 2 covers the basic theory behind the operation of a two-level voltage source converter and the prerequisites for the mathematical modeling of the system. LCL filter concepts are introduced. Chapter 3 introduces the theory of direct model predictive control. Chapter 4 outlines the controlled system and the controller structure. Chapter 5 outlines the real-time implementation of the FCS-MPC algorithm and the testing methods used to verify the results. Chapter 6 demonstrates the steady-state performance of the algorithm based on testing and analyzes the capability of the control hardware. Chapter 7 concludes and proposes possible future research for the subject.

2. CASE STUDY: GRID-CONNECTED CONVERTER VIA LCL-FILTER

This chapter presents the theoretical background of the system modeling utilized in this thesis. Prerequisite concepts for modeling the system, namely stationary and rotational reference frames, are introduced. A typical grid-connected converter is presented. Two-level (2L) voltage source converter (VSC) topology based on insulated-gate bipolar transistors (IGBT) is presented. The concept of LCL filter resonance is introduced, and its effects on the controlled system are discussed.

2.1 Orthogonal reference frames

2.1.1 Stationary orthogonal reference frame

The stationary orthogonal reference frame is also known as $\alpha\beta0$-coordinate system. The axes of the system are fixed and perpendicular to each other. $\alpha\beta0$-coordinate system and abc-coordinate system are visualized in Fig. 2.1. Three-phase quantities in the abc-reference frame $\boldsymbol{\xi}_{abc} = [\xi_a\ \xi_b\ \xi_c]^T$ can be expressed in $\alpha\beta0$-reference frame without loss of information, and vice-versa. Clarke transformation transforms three-phase quantities to $\alpha\beta0$-frame.

$$\boldsymbol{\xi}_{\alpha\beta0} = \boldsymbol{K}\boldsymbol{\xi}_{abc}, \tag{2.1}$$

where $\boldsymbol{\xi}_{\alpha\beta0}$ are the equivalent quantities in $\alpha\beta0$-reference frame and $\boldsymbol{K}$ is the Clarke transformation matrix

$$\boldsymbol{K} = \frac{2}{3}\begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix}. \tag{2.2}$$

The factor 2/3 in $\boldsymbol{K}$ preserves the amplitude of the three-phase signals during transformation. The zero component of $\alpha\beta0$-reference frame denotes the common-mode component, which is only present if there is asymmetry within the system or the star point

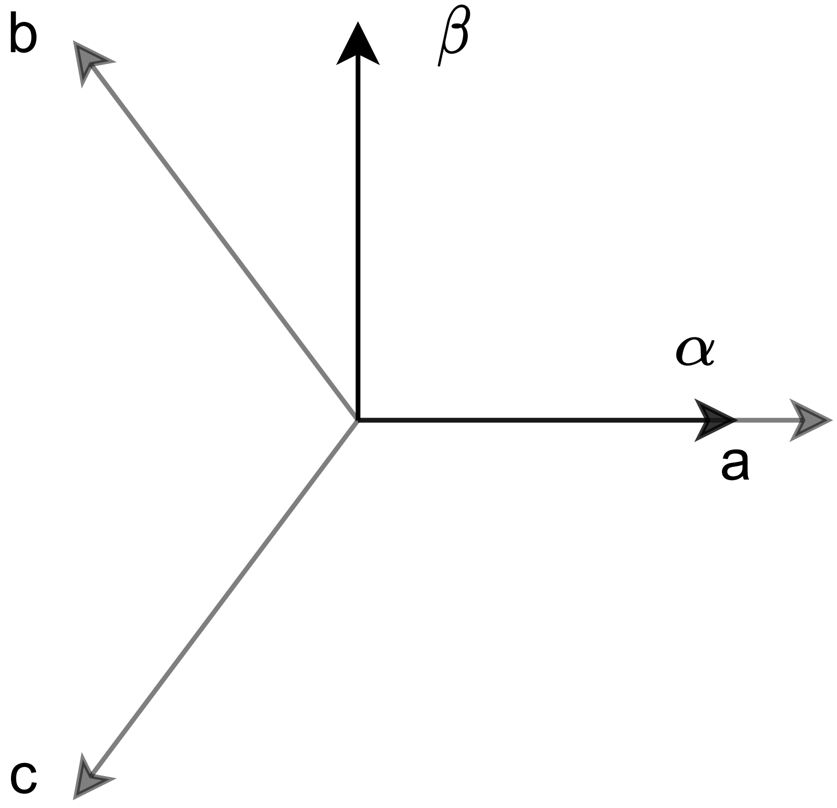


Figure 2.1. *$\alpha\beta0$ and abc-reference frame*

of the load is not floating. Due to this, in a balanced three-phase system, the Clarke transformation matrix can be reduced to:

$$\tilde{\boldsymbol{K}} = \frac{2}{3}\begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix}. \tag{2.3}$$

Transforming a quantity from $\alpha\beta0$-reference frame to abc-reference frame is done using inverse Clarke transformation. The reduced inverse Clarke transformation matrix is given by

$$\tilde{\boldsymbol{K}}^{-1} = \begin{bmatrix} 1 & 0 \\ -\frac{1}{2} & \frac{\sqrt{3}}{2} \\ -\frac{1}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix}. \tag{2.4}$$

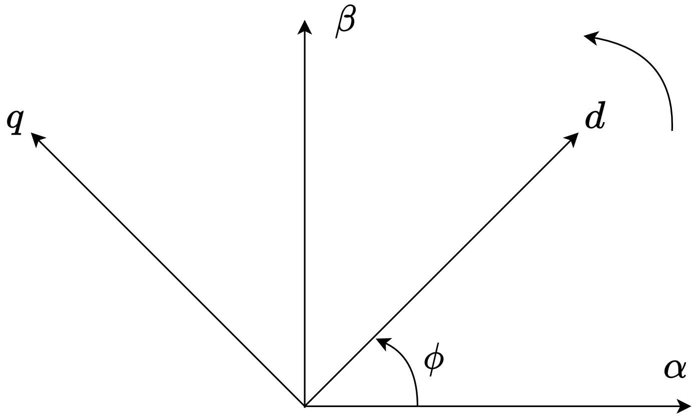


Figure 2.2. *Rotating reference frame*

2.1.2 Rotating orthogonal reference frame

Rotating orthogonal reference frame is also known as $dq0$-coordinate system. The axes of the coordinate system are perpendicular to each other and rotate counter-clockwise at an angular speed ω_{fr}. Three-phase quantities rotating at the same angular frequency will appear constant when the quantities are transformed into a rotating reference frame. $\alpha\beta$-reference frame transformation is a special case of $dq0$-transformation, where $\omega_{fr} = 0$

The transformation between $\alpha\beta 0$-reference frame and rotating reference frame is accomplished through a rotational matrix $\boldsymbol{R}(\phi)$, which is defined as

$$\boldsymbol{R}(\phi) = \begin{bmatrix} cos(\phi) & sin(\phi) \\ -sin(\phi) & cos(\phi) \end{bmatrix}, \tag{2.5}$$

where ϕ is the angle between the d-axis of the rotational reference frame and the α-axis of the stationary reference frame. Fig. 2.2 shows the rotating reference frame. Three-phase quantities may be transformed to dq-reference frame using the Park transformation. The reduced Park transformation matrix and its inverse are given in (2.6) and (2.7).

$$\tilde{\boldsymbol{K}}(\phi) = \frac{2}{3} \begin{bmatrix} cos(\phi) & cos(\phi - \frac{2\pi}{3}) & cos(\phi + \frac{2\pi}{3}) \\ -sin(\phi) & -sin(\phi - \frac{2\pi}{3}) & -sin(\phi + \frac{2\pi}{3}) \end{bmatrix}, \tag{2.6}$$

$$\tilde{\boldsymbol{K}}(\phi)^{-1} = \begin{bmatrix} cos(\phi) & -sin(\phi) \\ cos(\phi - \frac{2\pi}{3}) & -sin(\phi - \frac{2\pi}{3}) \\ cos(\phi + \frac{2\pi}{3}) & -sin(\phi + \frac{2\pi}{3}) \end{bmatrix}, \tag{2.7}$$

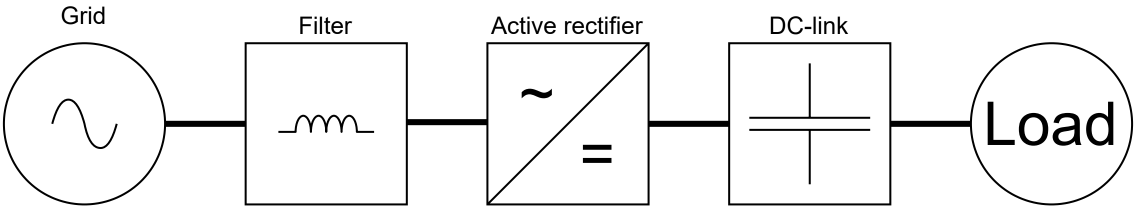


Figure 2.3. *Grid-connected converter*

2.2 Grid-connected three-phase voltage source converter

An illustration of a grid-connected converter is shown in Fig. 2.3. It consists of five main parts: the grid, filter for power quality, rectifier, DC link, and load. Grid-connected converters may also include a transformer between the grid and the filter. In grid-connected applications, LC- or LCL filters are commonly used because of their ability to improve upon the harmonic attenuation caused by the series inductors [6]. The dc link has a capacitor into which the active rectifier can store energy. The load may consist of another converter providing ac power to a motor or microgrid. It is also possible to directly provide power to a dc load through the dc link.

The grid-connected converter is used as an active rectifier. An active rectifier can control the dc voltage of the dc link capacitor v_{dc} and provide bidirectional power flow from the grid to the dc link or from the dc link to the grid. Active rectifier maintains constant the dc-link voltage when the current drawn from the dc-link capacitor changes. This thesis focuses on the control of active rectifier controlling the dc-link voltage. This is achieved by simultaneous control of LCL-filter currents and voltages.

2.2.1 Two-level voltage source converter topology

The converter transforms ac to dc or vice versa through a set of switches that are set on and off at specific time instants. The switches comprise an IGBT and a diode connected in parallel. The output phase voltage is controlled by making the switch conductive. The topology of a two-level three-phase converter is shown in Fig. 2.4.

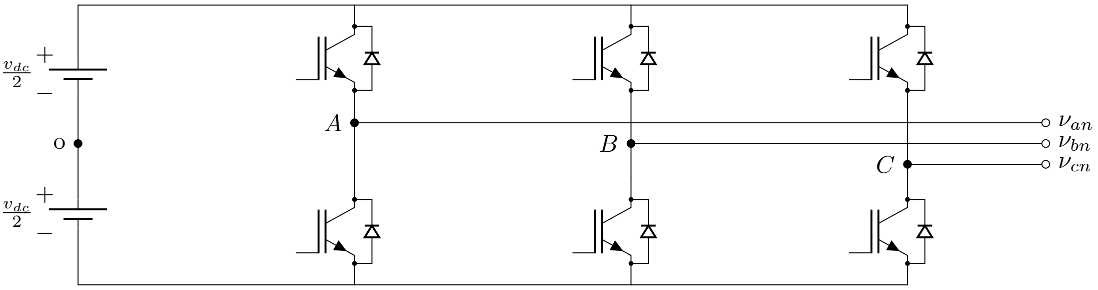


Figure 2.4. *Two-level voltage source converter*

The converter consists of three legs. Each leg has an upper and lower switch, which controls the converter's output voltage. The switches cannot be turned on simultaneously. This would result in short-circuiting the dc-link voltage, possibly resulting in hardware failures and loss of control. Due to this, each phase leg is limited to two possible voltage stages.

Based on the state of the switches, the output phase voltage can be either $\frac{v_{dc}}{2}$ or $-\frac{v_{dc}}{2}$. Therefore, the three-phase output voltage for any given three-phase switch position may be expressed in stationary $\alpha\beta$-reference frame as

$$\boldsymbol{v}_{c,\alpha\beta} = \frac{v_{dc}}{2}\boldsymbol{K}\boldsymbol{u}_{abc} \tag{2.8}$$

$$\boldsymbol{u}_{abc} = \begin{bmatrix} u_a \\ u_b \\ u_c \end{bmatrix}, \tag{2.9}$$

where $u_x \in \{-1, 1\}$, with $x \in \{a, b, c\}$.

The two-level converter can produce a maximum of $2^3 = 8$ possible switch position combinations. Transformation of these combinations to $\alpha\beta$-reference frame will result in seven unique voltage vectors. The vectors can be divided into two groups: the active voltage vectors and the zero vectors. The six active vectors form a hexagon shape around the origin of $\alpha\beta$-coordinate system. They are numbered as $\boldsymbol{V}_{1-6}$. The zero vectors are located in origin, resulting from the upper or lower switches being turned on across the three phases. They are named as $\boldsymbol{V}_0$ and $\boldsymbol{V}_7$. The two zero vectors result in the same output voltage of the converter; thus, only one is unique. Fig. 2.5 illustrates voltage vectors alongside their corresponding switch positions.

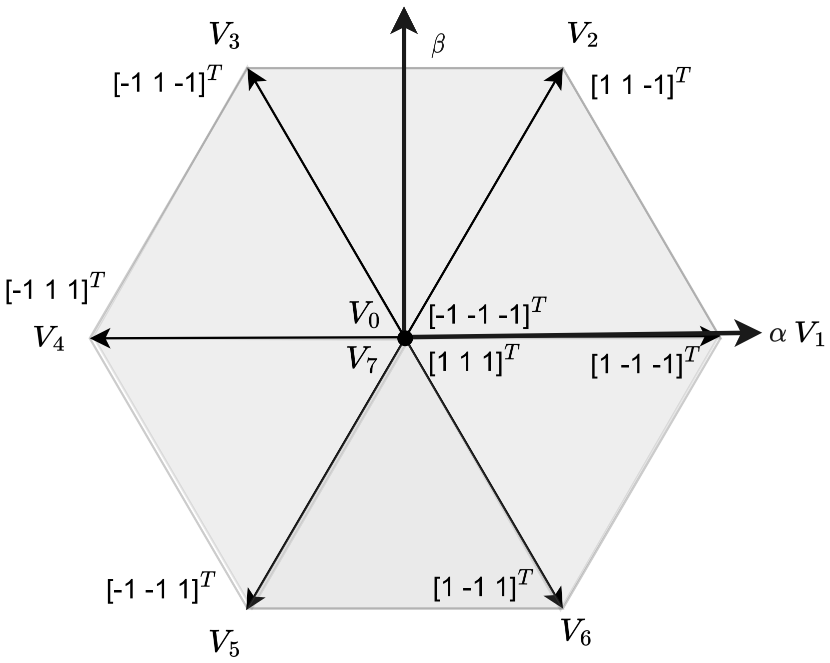


Figure 2.5. *Two-level converter voltage vectors in $\alpha\beta$-reference frame*

2.3 LCL filter

The LCL filter aims to improve the power quality fed to the grid. LCL filters are commonly used in grid-tied applications due to their ability to dampen the harmonics more effectively along with reduced size when compared with L-filter [7]. However, the usage of LCL filters introduces complexity due to the higher order of the controlled system. This means that the control of converter current and LCL-filter capacitor current must be considered alongside the grid current. The harmonic frequency of the LCL filter needs to be accounted for and dampened to avoid the amplification of harmonic currents [8].

The resonance of the LCL filter is caused by the filter capacitor reactance X_C oscillating against the grid-side reactance X_{fg} and the converter-side reactance X_{fc}. Each component has small internal resistance that could provide passive damping of oscillations. However, the passive damping is very negligible, and thus, their effect on the calculation is disregarded. Based on this, the resonance frequencies can be defined as:

$$f_{res} = f_B \frac{1}{\sqrt{X_C \frac{X_{fc} X_{fg}}{X_{fc} + X_{fg}}}}, \qquad \tilde{f}_{res} = f_B \frac{1}{\sqrt{X_C X_{fg}}}, \tag{2.10}$$

where $f_B = \frac{\omega_B}{2\pi}$ is the base frequency in hertz. Base angular frequency ω_B is f_B in rad/s. The base frequency is set to grid frequency, e.g. 50Hz. The first equation defines the dominant resonance frequency f_{res} caused by the interaction between filter capacitor reactance X_c, converter-side reactance X_{fc}, and the total grid-side reactance X_{fg}.

Operating at switching frequencies near the filter resonance frequency can amplify harmonic frequencies in the vicinity of the resonance frequency of the filter. The converter needs to operate at a higher switching frequency or provide active damping to prevent the amplification of these harmonic frequencies. Failing to account for the LCL filter resonance frequency can lead to closed-loop stability issues. The loss of stability may cause hardware failures, e.g., blown capacitors.

3. MODEL PREDICTIVE CONTROL

This chapter discusses the basic principles of MPC in power electronics. The prediction of future states of the system through system modeling is explained. A mathematical basis for the optimization problem for MPC is introduced. The MPC algorithm is extended for prediction horizons larger than one, and concepts related to this, such as receding horizon policy, are explained.

3.1 MPC in power electronics

Model predictive control is a control method first devised in the 1970s. It was developed to be used in the process industry, in which the processes controlled were non-linear and included physical constraints. By the late 1990s, various industries, primarily the refining and petrochemical industries, had adopted models of predictive control to their systems. Other notable industries adopting the control method to their processes included the chemical industry, pulp and paper, aerospace, and automotive industry. [9],[10]

Model predictive control was not initially regarded as useful within power electronics. The computational power within control platforms 40 years ago was insufficient for the short sampling intervals and computational burden the control method required for robust performance. The first investigation on model predictive control in power electronics happened in the 1980s. [11],[12],[3]

Due to increased computational power within modern control platforms, MPC has become increasingly researched as a control method. The control algorithm is exceptionally well suited for high-power, low-switching frequency power electronics applications. Due to the control methods' ability to address multiple control objectives simultaneously, model predictive control is beneficial in multiple-input multiple-output (MIMO) systems. In the last two decades, MPC has been widely adapted to power electronics applications [13]. Applications of FCS-MPC algorithms have been implemented and tested in multiple different power electronics converters [14].

Model predictive control provides multiple advantages over traditional control methods: [15]

- Model predictive control concepts are intuitive and easy to understand
- It can be applied to a variety of systems
- Improved dynamic behavior of the system.
- Multi-variable control is simple
- Inclusion of non-linearities in the model
- Simple treatment and direct inclusion of constraints

Compared to traditional control methods, a significant disadvantage is that MPC is much more computationally demanding. The control platform hardware must be sufficiently powerful to ensure the control algorithm produces good results. A more powerful control platform may lead to increased costs in the production of said hardware. However, reducing the hardware requirements of other parts of the power electronics system becomes possible due to the improved dynamic behavior and the ability to control multiple control objectives simultaneously. LCL filter resonance can be handled much more effectively, allowing cheaper LCL filters to be used.

As the name states, MPC uses a mathematical model of the system to predict future states. Thus, the model of the controlled system must be accurately described, or the performance will deteriorate. As such, estimation or adaptive algorithms must be considered for the model to remain accurate should the system parameters change over time.

3.2 Basic principles of model predictive control

The basic principle behind model predictive control is to use a mathematical model of the system to accurately predict the system's future state and react with optimal control sequence. As such, the algorithm can be divided into two rough categories: the prediction stage and the optimization stage. The prediction stage is done to predict the future system state accurately. Optimization is performed to find the optimal control variables based on the predicted system state and control objectives.

3.2.1 System model

The state model is a mathematical model describing the dynamics of the system. It can then be used to predict the system's future behavior based on the current system state and the effect of manipulated variables in the system. State model in linear form in the continuous-time domain can be expressed as:

$$\frac{\mathrm{d}}{\mathrm{dt}}\boldsymbol{x}(t) = \boldsymbol{F}\boldsymbol{x}(t) + \boldsymbol{G}\boldsymbol{u}(t) \tag{3.1a}$$

$$\boldsymbol{y}(t) = \boldsymbol{C}\boldsymbol{x}(t), \tag{3.1b}$$

with system matrix $\boldsymbol{F}$, input matrix $\boldsymbol{G}$, output matrix $\boldsymbol{C}$, the state vector $\boldsymbol{x}(t)$, input vector $\boldsymbol{u}(t)$ and output vector $\boldsymbol{y}(t)$. The matrix dimensions $\boldsymbol{F}$, $\boldsymbol{G}$ and $\boldsymbol{C}$ are $n_x \times n_x$, $n_x \times n_u$ and $n_y \times n_x$ respectively, where n_x is the number of states, n_u the number of inputs and n_y the number of outputs. System matrix $\boldsymbol{F}$ denotes the effect of the current state of the system on the future behavior of the system, $\boldsymbol{G}$ denotes the effects of inputs on the system, and $\boldsymbol{C}$ determines which states are considered the outputs of the system.

System models in digital control platforms are not calculated in continuous time. The system is sampled in recurring time intervals, and the future state is predicted for the next time step. Thus, the linear model of the system needs to be discretized.

The linear system is updated every discrete time instant kT_s, where T_s is the sampling interval of the system. The manipulated variable stays constant for the control period. It is updated in fixed time instants $t = kT_s$, where $k \in \mathbb{N} = 0, 1, 2, ...$ denotes the time step the calculation is performed on. The continuous-time model presented in (3.1a) can be discretized by integrating it from $t = kT_s$ to $t = (k+1)T_s$. Because the controlled variable $\boldsymbol{u}$ remains constant during the control period, $\boldsymbol{u}(t) = \boldsymbol{u}(k)$ and the system matrices can be rewritten as:

$$\boldsymbol{x}(k+1) = \boldsymbol{A}\boldsymbol{x}(k) + \boldsymbol{B}\boldsymbol{u}(k) \tag{3.2a}$$

$$\boldsymbol{y}(k+1) = \boldsymbol{C}\boldsymbol{x}(k+1), \tag{3.2b}$$

where $\boldsymbol{A}$ is the discretized system matrix and $\boldsymbol{B}$ the discretized input matrix. They are derived from continuous-time matrices with

$$\boldsymbol{A} = e^{\boldsymbol{F}T_s} \tag{3.3a}$$

$$\boldsymbol{B} = \int_0^{T_s} e^{\boldsymbol{F}T_s} d\tau \boldsymbol{G}, \tag{3.3b}$$

where e is the matrix exponential. If $\boldsymbol{F}$ is non-singular, equation (3.3b) can be written as

$$\boldsymbol{B} = \boldsymbol{F}^{-1}(\boldsymbol{A} - \boldsymbol{I})\boldsymbol{G} \tag{3.4}$$

where $\boldsymbol{I}$ is a identity matrix of same dimensions as $\boldsymbol{A}$. This discretization method may be computationally intensive when it is calculated in real-time. The discretized matrices can be assumed to be constant, making it possible to calculate them offline. Assuming constant matrices, however, fails to consider the evolution of system parameters over time. Thus, discretization should be simplified to reduce computational burden if parameter evolution over time is considered. Exact discretization can be approximated through multiple methods, such as backward Euler or forward Euler, which are the most employed methods [3]. However, if the sampling interval of the system is too long, these methods may give inaccurate results. Further information about the approximation method used in this thesis is discussed in Chapter 4.

3.2.2 Objective function

The optimization stage is performed by formulating an objective function that captures the control objectives and analyzing the function's output values. The system states at time instant kT_s are fed into a discretized state model, which calculates the system's future states based on inputs and the system's current state. The future system output can then be compared with its reference. A finite number of inputs in FCS-MPC enables calculating the objective function for each input signal. Accounting for each possible input allows the MPC to choose the most suitable control set, which is the one that produces the lowest value of the objective function.

The objective function used in the optimization stage is presented in function (3.5).

$$J = ||\boldsymbol{y}_{ref}(k+1) - \boldsymbol{y}(k+1)||^2_{\boldsymbol{Q}}, \tag{3.5}$$

where $\boldsymbol{y}_{ref}(k+1)$ is the output reference vector for the next time step, $\boldsymbol{y}(k+1)$ the predicted output and $\boldsymbol{Q}$ a matrix containing weighing factors for each controlled state. The weighing factors affect how severely deviation from reference is penalized in the objective function. The function outputs higher values if state variables deviate from their reference values.

Control effort can be considered in the objective function to penalize the change in the control signal. When control effort in FCS-MPC is taken into account, the objective function can be formulated as

$$J = ||\boldsymbol{y}_{ref}(k+1) - \boldsymbol{y}(k+1)||^2_{Q} + \lambda_u||\Delta\boldsymbol{u}(k)||, \tag{3.6}$$

where $\Delta\boldsymbol{u}(k) = \boldsymbol{u}(k) - \boldsymbol{u}(k-1)$ and λ_u is a weighing factor for penalizing control effort. Tuning of λ_u directly affects the average switching frequency. If switching is penalized heavily, only sufficiently large deviations from the reference will cause the input to change.

The output of the objective function should always be non-negative.

It should be noted that the function uses the sum of squares, i.e., 2-norm, to penalize the tracking error. To minimize the computational complexity of the objective function, the 1-norm, i.e., the sum of absolute values, might seem preferable. However, using 1-norm has been noted to cause instability in applications where direct model predictive control with reference tracking has been implemented. The 2-norm approach improves closed-loop stability by ensuring that sufficiently large deviation from the reference will always dominate over the control effort [16]

The optimal control input is solved by finding the switch position that produces the minimum objective function value:

$$U_{opt}(k) = \arg\operatorname{minimize} J \tag{3.7a}$$

$$\text{subject to } \; x(k+1) = Ax(l) + Bu(k) \tag{3.7b}$$

$$y(k+1) = Cx(k+1) \tag{3.7c}$$

$$u(k) \in \mathbb{U} \in \{-1, 1\}^{n_u}, \tag{3.7d}$$

The system model and admissible control actions constrain the objective function.

3.2.3 Constraints

Model predictive control can easily consider different constraints of the system in the optimization stage. Constraints can be divided into two groups: hard and soft constraints. Hard constraints represent physical limitations of the system that cannot be violated in any way. For direct control problems in power electronics, the amount of switch positions limits the feasible set of input signals as shown in Eq. (2.9). The limitation of switch signals $u_x \in \{-1, 1\}$ is an example of a hard constraint.

Soft constraints, on the other hand, can be interpreted as mechanisms to protect the system from operating outside its safety limits. Soft constraints are imposed on the system, and while they do not entirely prevent violations from happening, effort should be made to reduce the degree of these violations.

User-defined soft constraints may be added to be considered within the objective function. These may include maximum and minimum values for controlled variables to operate within the safety limits of the control platform. The optimization stage may penalize the violation of the soft constraints with increasing severity; the more prominent the deviation from normal operating conditions, the more it is penalized.

3.3 Prediction Horizon of longer than one step

The trajectory of state signals can be predicted further for any given sequence of input signals. The prediction horizon is a time interval defined by the number of time steps the model predicts and the sampling interval of the system. The length of the prediction horizon is defined as N_pT_s, where N_p is the amount of time steps predicted. By utilizing longer horizons to predict the system behavior over a longer time interval in the future, the system performance and stability can be significantly increased [17].

With a prediction horizon of longer than one step, the objective function can be reformulated as

$$J = \sum_{l=k}^{k+N_p-1} ||\boldsymbol{y}_{ref}(l+1) - \boldsymbol{y}(l+1)||^2_{\boldsymbol{Q}} + \lambda_u ||\Delta \boldsymbol{u}(l)||, \tag{3.8}$$

The function will determine the optimal switching plan $\boldsymbol{U}_{opt}(k) = [\boldsymbol{u}^T(k) \quad \boldsymbol{u}^T(k+1) \quad \ldots \quad \boldsymbol{u}^T(k+N_p-1)]^T$. Increasing the horizon length increases the number of possible solutions for $\boldsymbol{U}_{opt}(k)$ exponentially with FCS-MPC. For this reason, the prediction horizon is kept short in this thesis to keep the computational burden modest. Single-step prediction horizons and two-step prediction horizons are implemented and tested.

Prediction horizon length strongly affects the closed-loop performance, especially when considering higher-order systems such as converters with LC- and LCL-filters [18]. A longer prediction horizon allows the controller to take better control actions. For this reason, even a relatively short prediction horizon can improve the performance and should be adopted [19].

3.3.1 Receding horizon policy

When the switching sequence is calculated over the prediction horizon, only the first element of the optimal sequence $\boldsymbol{U}_{opt}(k)$ is applied to the system at time step k. During the next time step $k+1$, a new optimal switching sequence $\boldsymbol{U}_{opt}(k+1)$ is calculated based on the state information $\boldsymbol{x}(k+1)$ using the objective function (3.8). Receding horizon policy is illustrated in Fig. 3.1, where $\boldsymbol{Y}_{ref}(k) = [\boldsymbol{y}_{ref}(k+1) \quad \boldsymbol{y}_{ref}(k+2) \quad \ldots \quad \boldsymbol{y}_{ref}(k+N_p)]$ is the matrix containing the output reference vectors for the duration of the prediction horizon and $\boldsymbol{Y}$ is the calculated trajectory based on the optimal switching sequence.

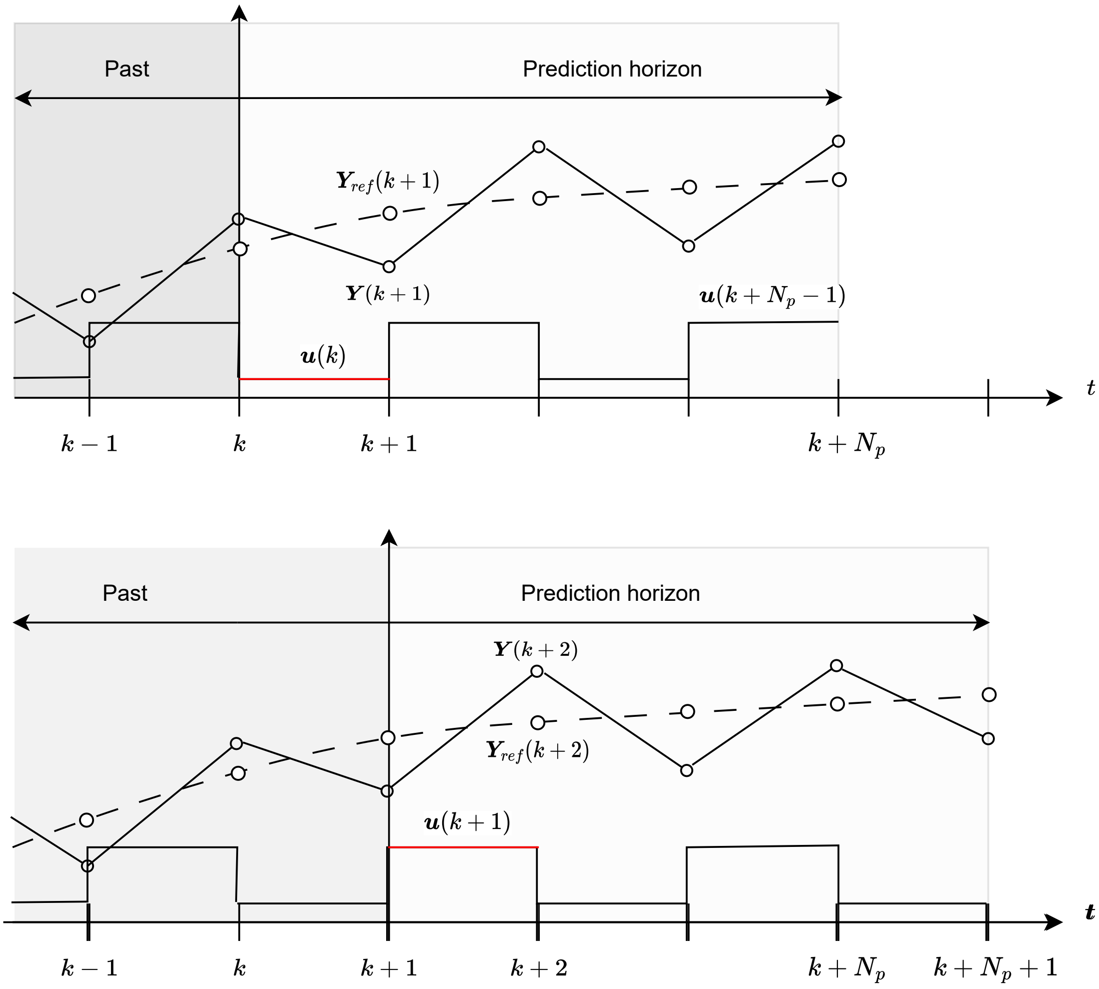


Figure 3.1. *Receding horizon policy [11]*

4. CONTROL OF GRID-CONNECTED INVERTER WITH LCL FILTER

The controlled model is a grid-connected three-phase two-level inverter with an LCL filter connected between the converter and the grid. The main goal of the converter is to act as an active rectifier and regulate the dc-link voltage of the converter. The FCS-MPC algorithm simultaneously controls the installed LCL filter's capacitor voltage alongside converter-side and grid-side current to achieve regulation of the dc-link voltage.

This chapter proposes an FCS-MPC algorithm for the controlled model. The control objectives are clarified, and the controller structure is explained. System and input matrices are created for the controlled model. LCL observer model used to estimate grid-side filter current and filter capacitor voltage is briefly explained. A method to reduce the computational complexity of the developed FCS-MPC algorithm is proposed.

4.1 Control objectives

The control objective of the controller is maintaining a constant dc-link voltage when a load is applied to it. The control has two distinct control loops: the outer dc voltage control loop and the system state control loop. The dc-link voltage is controlled through PI control. The dc voltage control loop will generate a grid current d-axis reference based on the output of the PI controller. The q-axis current reference is set to zero in the context of this thesis. However, the user can manually set it to control the system's reactive power. The output references for filter capacitor voltage and converter current are calculated based on steady-state equations of the system.

An FCS-MPC algorithm is implemented to act as a state control loop. FCS-MPC simultaneously controls converter current, filter capacitor voltage, and grid current. Grid current reference tracking is prioritized within the control to ensure low total harmonic distortion (THD) of the grid current at the point of common coupling (PCC).

The control structure is shown in Fig 4.1. Converter current is measured at converter output, and grid voltage is measured at PCC. The feedbacks are processed and transformed into $\alpha\beta$-reference frame. An LCL observer estimates the instantaneous values for filter capacitor voltage and grid current. A phase-locked-loop (PLL) determines grid voltage

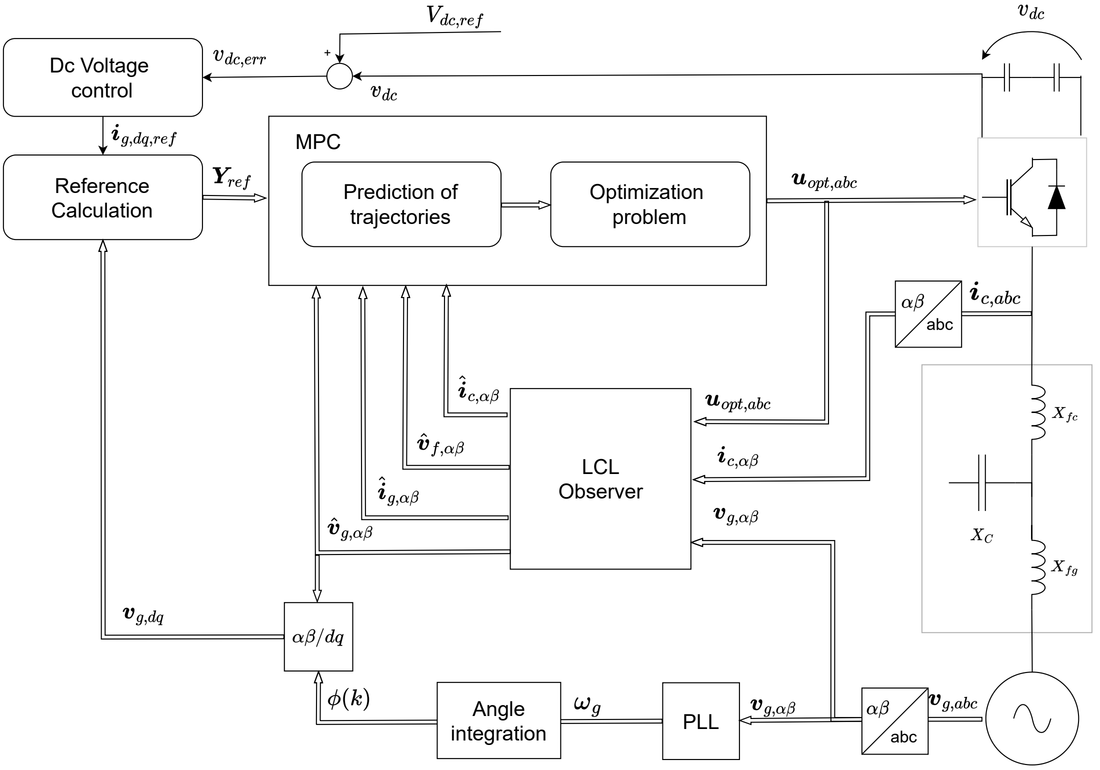


Figure 4.1. *Controller structure*

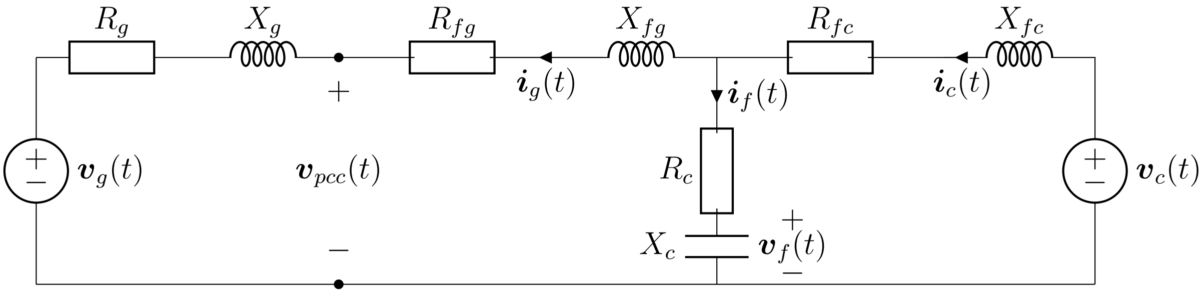
angular frequency based on grid voltage feedback. The angular frequency is integrated to get the grid voltage angle information at time step k for dq-transformation.

4.2 Control of grid-connected converter

4.2.1 Controller model

The equivalent circuit of the system in the $\alpha\beta$-reference frame is shown in Fig. 4.2. Each current and voltage has an α- and β-component. The system dynamics can be described in continuous-time differential equations using Kirchoff's voltage and current laws. The equations are presented in Eq. (4.1).

Figure 4.2. *Equivalent circuit*



$$
\begin{aligned}
X_{fc}\frac{\mathrm{d}\boldsymbol{i}_c(t)}{\mathrm{d}t} &= -(R_{fc}+R_c)\boldsymbol{i}_c(t)-\boldsymbol{v}_f(t)++R_c\boldsymbol{i}_g(t)+\boldsymbol{v}_c(t) && \text{(4.1a)}\\
X_c\frac{\mathrm{d}\boldsymbol{v}_f(t)}{\mathrm{d}t} &= \boldsymbol{i}_f(t)=\boldsymbol{i}_c(t)-\boldsymbol{i}_g(t) && \text{(4.1b)}\\
(X_{fg}+X_g)\frac{\mathrm{d}\boldsymbol{i}_g(t)}{\mathrm{d}t} &= -R_c\boldsymbol{i}_c(t)+\boldsymbol{v}_f(t)-(R_{fg}+R_g-R_c)\boldsymbol{i}_g(t)-\boldsymbol{v}_g(t). && \text{(4.1c)}
\end{aligned}
$$

State vector is defined as $\boldsymbol{x}(t) = [\boldsymbol{i}_c^T(t)\ \boldsymbol{v}_f^T(t)\ \boldsymbol{i}_g^T(t)]^T$ and similarly output vector is defined as $\boldsymbol{y}(t) = [\boldsymbol{i}_c^T(t)\ \boldsymbol{v}_f^T(t)\ \boldsymbol{i}_g^T(t)]^T$. The two input vectors are as $\boldsymbol{u}_1(t) = \boldsymbol{u}_{abc}(t)$ and $\boldsymbol{u}_2(t) = \boldsymbol{v}_g^T(t)$. Based on these vectors, state-time matrices can be created based on 3.1a and (4.1). The following notation is taken into account to simplify the calculations: $R_1 = R_{fc} + R_c$, $R_2 = R_{fg} + R_g - R_c$ and $X = X_{fg} + X_g$.

$$
\mathbf{F} = \begin{bmatrix}
-\frac{R_1}{X_{fc}} & 0 & -\frac{1}{X_{fc}} & 0 & \frac{R_c}{X_{fc}} & 0\\
0 & -\frac{R_1}{X_{fc}} & 0 & -\frac{1}{X_{fc}} & 0 & \frac{R_c}{X_{fc}}\\
\frac{1}{X_C} & 0 & 0 & 0 & -\frac{1}{X_C} & 0\\
0 & \frac{1}{X_C} & 0 & 0 & 0 & -\frac{1}{X_C}\\
-\frac{R_c}{X} & 0 & \frac{1}{X} & 0 & -\frac{R_2}{X} & 0\\
0 & -\frac{R_c}{X} & 0 & \frac{1}{X} & 0 & -\frac{R_2}{X}
\end{bmatrix}
$$

$$
\boldsymbol{G}_1 = \frac{V_{dc}}{2X_{fc}}[\tilde{\boldsymbol{K}}^T\ \mathbf{0}_{3\times 4}]^T
$$

$$
\boldsymbol{G}_2 = -\frac{1}{X}[\mathbf{0}_{2\times 4}\ \boldsymbol{I}_{\mathbf{2}}]^T
$$

$$
\boldsymbol{C} = \boldsymbol{I}_{\mathbf{6}}
$$

The two input vectors are three-phase switch positions and grid voltage that are taken into account with two input matrices, $\boldsymbol{G}_1$ and $\boldsymbol{G}_1$. The zero matrix $\mathbf{0}$ is a matrix containing zeros. The size of the zero matrices are denoted in the subscript. Switch positions are described in abc-plane; thus, reduced Clarke matrix $\tilde{K}$ is present within $\boldsymbol{G}_1$.

MPC requires discrete system matrices for operation. The Forward Euler method was tested to limit the computational burden. However, it did not yield satisfactory results during the simulation testing of the algorithm. Due to this, a more accurate 4th-order approximation of exact discretization was used to ensure good performance:

$$\boldsymbol{A} = \boldsymbol{F}T_s + \frac{1}{2}(\boldsymbol{F}T_s)^2 + \frac{1}{3!}(\boldsymbol{F}T_s)^3 + \frac{1}{4!}(\boldsymbol{F}T_s)^4 \tag{4.2a}$$

$$\boldsymbol{B} = \boldsymbol{F}^{-1}(\boldsymbol{A} - \boldsymbol{I})\boldsymbol{G}. \tag{4.2b}$$

.

A flowchart describing the operation of single-step horizon FCS-MPC algorithm is shown in Fig. 4.3. The control model first initializes input for the FCS-MPC algorithm. Current system state $\boldsymbol{x}(k)$ is created based on the feedback and estimated signals. An output reference vector is created.

A scalar variable called J_{opt} is created and initialized to infinity. The purpose of J_{opt} is to determine the optimal switch position. During the enumeration of switch positions, the value of this variable is overwritten based on the output of the objective function.

The FCS-MPC algorithm can limit the number of switch positions in the enumeration. The feasible set of switch positions is determined through sector restriction, described in Section 4.3. The algorithm enumerates all eight possible switch positions if the feasible set is not reduced.

The algorithm always starts and ends the enumeration with zero voltage vectors. This means that the enumeration starts with switch position $\boldsymbol{u}_{abc} = [-1\,-1\,-1]^T$ and ends with $\boldsymbol{u}_{abc} = [1\,1\,1]^T$. After the prediction of states based on the switch position is completed, the objective function is calculated. Its output J is then compared to J_{opt}. If J_{opt} is greater than the output of the objective function, it will get overwritten with J, and the switch position associated with this value gets written to parameter $\boldsymbol{u}_{opt,abc}$. If the output of the objective function output is greater than J_{opt}, the algorithm will proceed to the next switch position.

Once the FCS-MPC algorithm has enumerated each feasible switch position, it exits the enumeration loop. The optimal switch position $\boldsymbol{u}_{opt,abc}$ is actuated.

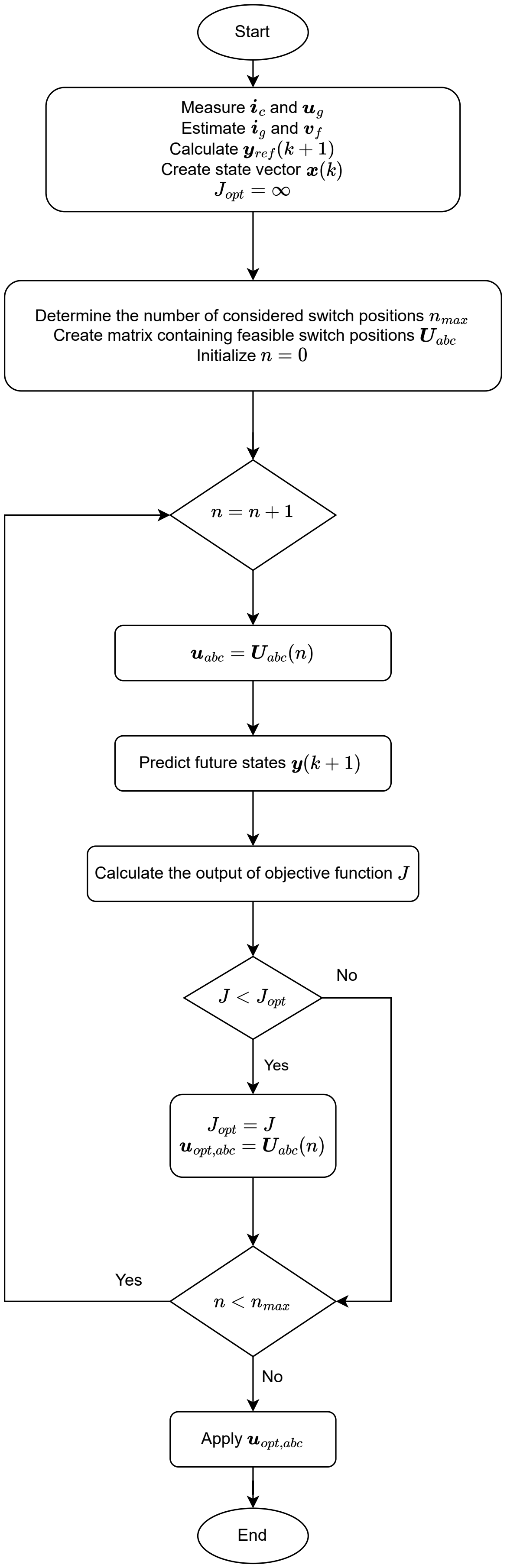

Start
Measure $\boldsymbol{i}_c$ and $\boldsymbol{u}_g$
Estimate $\boldsymbol{i}_g$ and $\boldsymbol{v}_f$
Calculate $\boldsymbol{y}_{ref}(k+1)$
Create state vector $\boldsymbol{x}(k)$
$J_{opt} = \infty$
Determine the number of considered switch positions n_{max}
Create matrix containing feasible switch positions $\boldsymbol{U}_{abc}$
Initialize $n = 0$
$n = n + 1$
$\boldsymbol{u}_{abc} = \boldsymbol{U}_{abc}(n)$
Predict future states $\boldsymbol{y}(k+1)$
Calculate the output of objective function J
$J < J_{opt}$
No
Yes
$J_{opt} = J$
$\boldsymbol{u}_{opt,abc} = \boldsymbol{U}_{abc}(n)$
Yes
$n < n_{max}$
No
Apply $\boldsymbol{u}_{opt,abc}$
End

Figure 4.3. *FCS-MPC algorithm flowchart*

4.2.2 Reference values

The grid current d-axis reference generated by the dc voltage control loop and the measured grid voltage $\boldsymbol{v}_{g,dq}$ are used to generate the reference trajectory $\boldsymbol{Y}_{ref}$. The reference calculation is based on steady-state equations of the system. They are calculated in the dq-reference frame, assuming that the calculated reference values stay constant through the prediction horizon N_pT_s. The reference values are calculated according to the following:

$$\boldsymbol{v}_{f,ref,dq} = \boldsymbol{v}_{g,dq} + j\omega_g X_{fg} i_{g,ref,dq} \tag{4.3a}$$

$$\boldsymbol{i}_{c,ref,dq} = \boldsymbol{i}_{g,ref,dq} + j\omega_g X_C \boldsymbol{v}_{f,ref,dq} \tag{4.3b}$$

$$\boldsymbol{v}_{c,ref,dq} = \boldsymbol{v}_{f,ref,dq} + j\omega_g X_{fc} \boldsymbol{i}_{c,ref,dq}, \tag{4.3c}$$

where $\boldsymbol{v}_{f,ref,dq}$, $\boldsymbol{i}_{c,ref,dq}$ and $\boldsymbol{v}_{c,ref,dq}$ are the reference values for filter capacitor voltage, converter current, and converter output voltage. The converter output voltage reference will limit the feasible set of input signals and approximate the desired output voltage. Constant reference values are then translated to $\alpha\beta$-reference frame through inverse rotational matrix (2.7). Angle information $\phi(k)$ for transformation is gained from the PLL.

Grid angular frequency and sampling interval are assumed to stay constant during operation. Due to this, the angle $\phi(k)$ used in the dq-transformation changes in constant increments of $\Delta\phi$ between time steps. This enables the transformation from dq-reference frame to $\alpha\beta$-reference frame for future time steps e.g. $\phi(k+1) = \phi(k) + \Delta\phi$.

4.2.3 Objective function

The state-space model in discrete time predicts the system output trajectory $\boldsymbol{Y}(k) = [\boldsymbol{y}(k+1)^T \ \boldsymbol{y}(k+2)^T \ ... \ \boldsymbol{y}(k+N_p)^T]$ for each possible switch position plan $\boldsymbol{U}(k) = [\boldsymbol{u}_1(k)^T \ \boldsymbol{u}_1(k+1)^T \ ... \ \boldsymbol{u}_1(k+N_p-1)^T]$. The optimal switching plan is chosen based on the output of Eq. (4.4).

$$J = \sum_{l=k}^{k+N_p+1} ||\boldsymbol{y}_{ref}(l+1) - \boldsymbol{y}(l+1)||_{\boldsymbol{Q}}^2 + \lambda_u||\Delta\boldsymbol{u}_1(l)||. \tag{4.4}$$

$\boldsymbol{Q}$-matrix contains a corresponding weighting factor for each controlled state variable. The weighting factors determine the degree to which reference tracking is prioritized for any given variable. The $\boldsymbol{Q}$-matrix is an 6×6 diagonal matrix:

$$\boldsymbol{Q} = \begin{bmatrix} q_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & q_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & q_2 & 0 & 0 & 0 \\ 0 & 0 & 0 & q_2 & 0 & 0 \\ 0 & 0 & 0 & 0 & q_3 & 0 \\ 0 & 0 & 0 & 0 & 0 & q_3 \end{bmatrix},$$

where q_1 is the converter current weighting factor, q_2 the filter capacitor voltage weighting factor and q_3 the grid current weighting factor. The main goal of the converter tuning is to reduce the harmonic content of the grid current $I_{g,THD}$. For this reason, the values will be chosen in a manner that prioritizes grid current reference tracking. Hence, a much bigger penalty should be imposed on the associated entries of the matrix $\boldsymbol{Q}$ [7].

4.3 LCL Observer

A full-order observer estimates the instantaneous value for grid current and LCL filter capacitor voltage. The observer uses converter current, grid voltage measurement v_g, and previous switch position $\boldsymbol{u}(k-1)$ as inputs. LCL observer uses the matrices described in section 4.2.1. The observer model is expressed in equation (4.5).

$$\hat{\boldsymbol{x}}(k) = \boldsymbol{F}\hat{\boldsymbol{x}}(k-1) + \boldsymbol{G}_1\boldsymbol{u}_1(k-1) + \boldsymbol{G}_2\boldsymbol{u}_2(k-1) + \boldsymbol{L}_{LCL}(i_c(k-1) - \hat{i_c}(k-1)), \quad (4.5)$$

where $\hat{\boldsymbol{x}}$ is the estimated state values, $\hat{i_c}$ is the estimated converter current and $\boldsymbol{L}_{LCL}$ is a matrix containing observer gain. The matrix $\hat{\boldsymbol{x}}(k-1)$ contains the converter current feedback and the estimated filter voltage and grid current from the last control cycle:
$\hat{\boldsymbol{x}}(k-1) = [\boldsymbol{i}_c(k-1)^T \ \hat{\boldsymbol{v}}_f(k-1)^T \ \hat{\boldsymbol{i}}_g(k-1)^T]^T]$

The LCL observer compensates for the feedback delay in state feedback. Due to delays in the control hardware, feedback is received for time step $k-1$. The observer performs initial state prediction by estimating state values at time step k based on the feedback received at $k-1$.

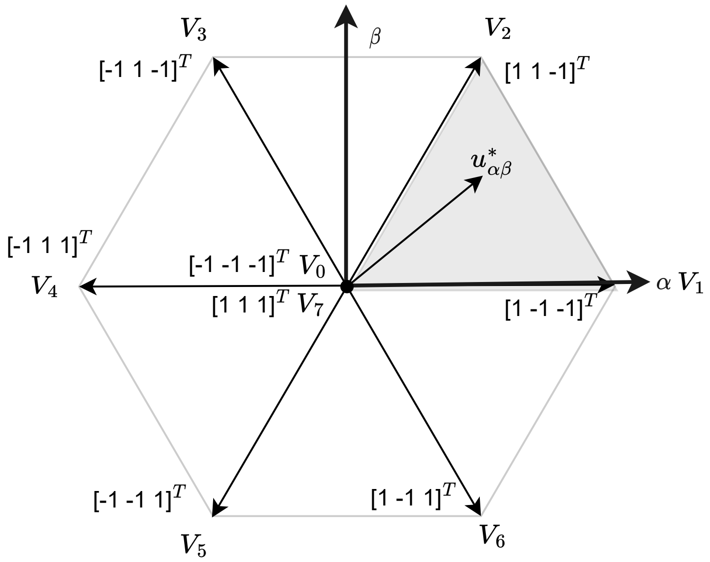


Figure 4.4. *Sector restriction*

4.4 Sector restriction

Several ways exist to reduce the computational burden the model predictive control algorithm requires. This thesis implements sector restriction, which restricts the feasible set of switch positions to be considered in the algorithm [19]. Sector restriction enables the MPC optimization stage to only evaluate the most probable switch positions to be chosen out of all the possibilities. Certain studies have shown that restricting the feasible set reduces the computational effort required significantly [20] [21]. However, the reduced feasible set can lead to suboptimality due to the reduced amount of considered switch positions [3].

Sector restriction will utilize the converter voltage reference calculated in (4.3c). The algorithm will first identify the angle of the voltage reference vector and identify which sector the most feasible solution exists in. Sector restriction is illustrated in Fig. 4.4.

The most probable switch positions to be chosen by the algorithm are the switch positions that correspond to the voltage vectors of the surrounding sector. In the case of sector one, these would be $\boldsymbol{V}_x$, where $x \in [0, 1, 2, 7]$. Only four out of eight switch positions are considered for a single-step horizon. Considering a two-step prediction horizon without sector restriction, the possible switching states amount to $8^2 = 64$. The sector restriction will reduce this amount to $4^2 = 16$.

It should be noted that the zero voltage vectors result in the same output voltage of the converter. Due to this, the amount of switch positions could be reduced to only three. The second zero voltage vector will affect the switching frequency due to the FCS-MPC algorithm considering the change between the last switch position and the considered one. This thesis, however, takes into account both of the zero vectors.

Sector restriction may affect the performance of the algorithm negatively. The affected performance is especially apparent near the transition between sectors. Due to this negative effect on the performance, it may be beneficial to extend the sector restriction to consider switch positions from two sectors instead of one. Considering more switch positions from another sector does not eliminate the possibility of suboptimal switching actions but will reduce the possibility.

2-sector restriction adds a single switch position to be considered. The two closest sectors are selected based on the converter output voltage reference angle, and their corresponding switch positions are considered. Compared to single-sector restriction, five possible switch positions are now considered. This is illustrated in Fig. 4.5

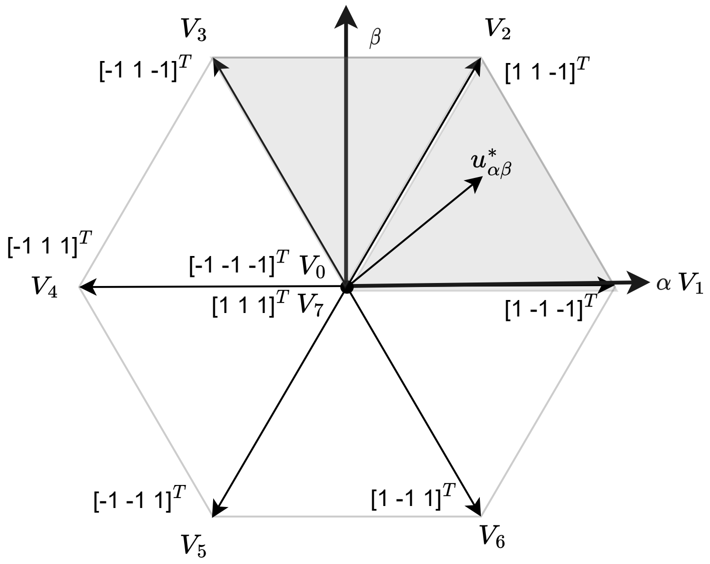


Figure 4.5. *Two-sector restriction*

5. IMPLEMENTATION

Implementation of a control algorithm on real-time applications requires the consideration of multiple factors which are not present in simulation models. Communication delays in the control platform, computation time on the chosen hardware, and the actuation of calculated optimal switch positions must be considered when implementing a control algorithm on a control platform. The combination of these factors limits the minimum execution time and sampling interval of the control platform.

This chapter focuses on implementing the control algorithm and the limitations of the chosen industrial control platform. The testing system used to verify the results is presented. The chapter evaluates the suitability of the control algorithm on the chosen control platform based on existing delays within control hardware. Analysis of the delays present within the control platform is integral for the future development of the algorithm.

5.1 Test setup

The algorithm will be tested with a hardware-in-the-loop real-time simulator. The test system this thesis uses to validate the results is a commercially available Typhoon HIL 604 real-time simulator. It is designed to simulate electrical circuits for power electronics application testing and validation purposes. The HIL model includes a single converter unit operated as an Active front-end unit.

The device under test is a Danfoss ic7-Automation system module operating as an AFE unit. The nominal voltage of the chosen platform is 400V, and the nominal current is 400A. It is operated with a 20kHz sampling frequency, and testing aims for an average switching frequency of 3kHz. An LCL filter with a resonance frequency of 1167Hz accompanies the control platform.

5.1.1 Typhoon HIL

The HIL simulator chosen for testing is a Typhoon HIL 604 real-time simulator. It uses a Zynq-7 System-on-chip (SoC), an integrated circuit with a built-in CPU and FPGA. The FPGA performs real-time electrical circuit simulation in the discrete-time domain. The simulation is performed using a 500ns sampling interval on the FPGA. The short sampling interval allows for accurate real-time simulation of the circuit conditions. The CPU of the HIL SoC is used for simulating slower phenomena within the test system. [22]

Typhoon HIL has separate software for creating, simulating, and verifying electrical circuits. The software used for this thesis is Typhoon HIL schematic editor and Typhoon HIL SCADA. Typhoon HIL schematic editor is a visual design tool for drawing and designing electrical circuits. It is designed to be intuitive and simple to use. The user can use the software to draw and compile the circuit into C code. The C code can then be downloaded into the Typhoon HIL hardware for simulation. [23]

Typhoon HIL SCADA is a user interface for real-time simulation. It enables the user to download a compiled simulation model into HIL hardware, create a customized user interface with built-in widgets and gather simulation data from Typhoon HIL [24]. Analog and digital inputs and outputs of the simulated circuit are defined within the SCADA settings file. The inputs and outputs allow the HIL hardware to communicate with the control hardware. Typhoon HIL 604 supports up to 16 analog inputs, 32 digital inputs, 32 analog outputs, and 32 digital outputs. The HIL simulates in real-time the operation of the grid-connected converter and sends feedback information to the control hardware through analog outputs. Control hardware executes the control software and determines the digital inputs to be sent into the HIL simulation through digital inputs based on calculated switch positions. SCADA allows the user to test and verify the electrical circuit created in the Schematic editor. It also enables the data capture of signals from simulated electrical circuits, e.g., currents and voltages. Data capture can sample the signals with up to 2Mhz sampling frequency. The measured signals used in this thesis are sampled at 1MHz frequency. They are saved in .mat-file format for data post-processing in Matlab.

5.1.2 Plant model

The plant model used for testing was created using Typhoon HIL schematic editor. The purpose of the plant model is to describe the dc link, AFE, LCL filter, and grid dynamics. A complete schematic for the model is presented in Appendix A.

5.1.3 Device under test

The device under test is a Danfoss ic7-Automation system module. The nominal voltage of the control platform is 400V, and the nominal current is 400A. The control platform is an active front-end unit, which will maintain DC-link voltage at referenced value.

The control platform consists of two separate SoC cards: a power card and a control card. The power card actuates the switching pattern and receives state feedback from the HIL simulation. The control card contains the control software and is responsible for the MPC algorithm. The cards are connected through an optical fiber connection.

In addition to these two cards, an external voltage measurement card is used to measure grid voltage. The external voltage measurement card is an option for the Danfoss ic7 product family. It enables galvanically isolated three-phase voltage measurement of either grid voltage at PCC or filter capacitor voltage. The measurement occurs once every control cycle, and the voltage measurement card sends this information to the control card through an optical fiber connection.

The testing platform has a breakout board directly connected to the power card and the HIL system. Its purpose is to communicate between the control platform and the HIL system. The breakout board scales the feedback signals from HIL to be read by the power card and the external voltage measurement card. In this thesis, dc voltage, phase current, and grid voltage are processed in the breakout board and sent to the power card and external voltage measurement card.

The control platform is controlled using a telnet connection from a PC. figure 5.1 shows the communication between devices.

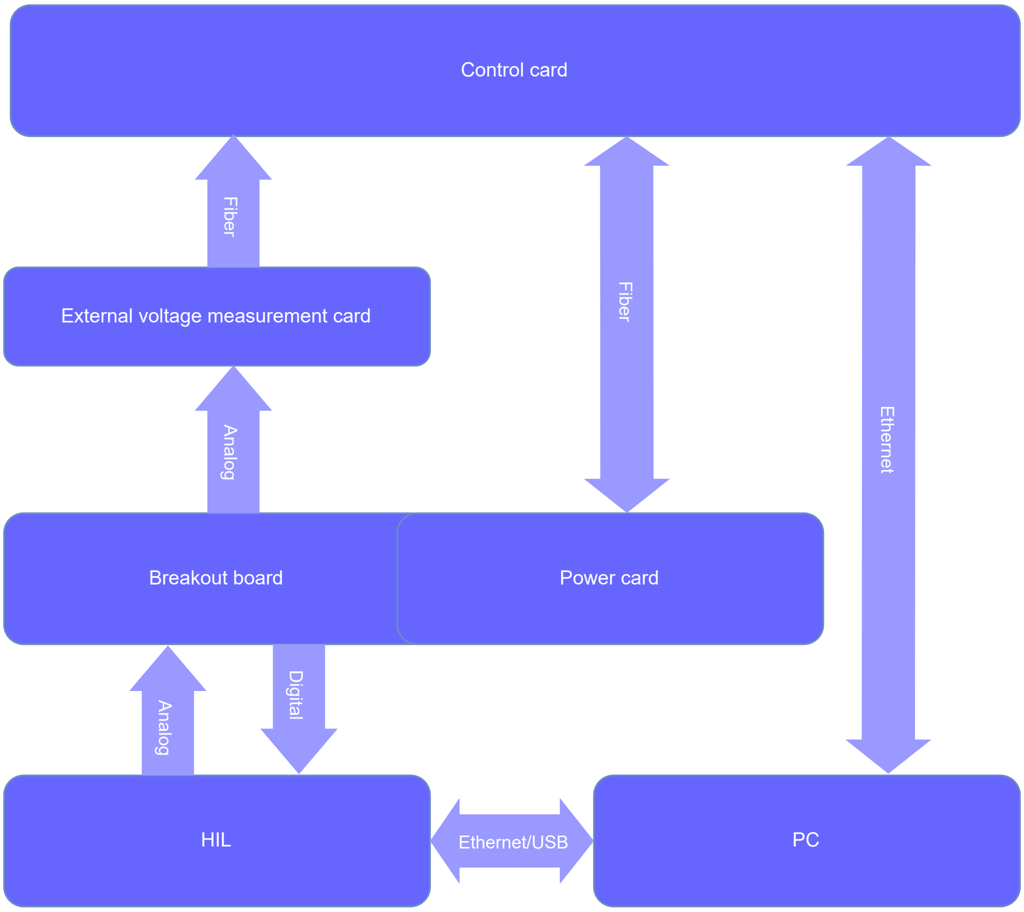


Figure 5.1. *Communication between test system devices*

5.2 Control platform real-time implementation

The development of the control was done using model-based development in Simulink. Model-based development allows the developer to identify and correct errors during the development process continuously through simulation-based testing [25]. Model-based development makes it possible to compile C/C++ code from a Simulink model. The generated code can then be generated into a software package for the control platform.

Existing control platform software was created using Simulink model-based development. A new Simulink library block containing the FCS-MPC functionality was created to implement the FCS-MPC algorithm. The new library block was added to the control software replacing pre-existing AFE current control and modulator. Fig. 4.1 shows the resulting controller structure.

The FCS-MPC Simulink library created for this thesis contained the core functionality of the implemented control method. The control library predicts the trajectory of the system states for each switch position and optimizes the selection based on objective function output. The calculation was performed in $\alpha\beta$-reference frame. Reference value generation was done using a rotating reference frame. The Simulink block uses grid voltage angle information to generate reference values in $\alpha\beta$-reference frame for FCS-MPC algorithms.

A separate Simulink library component was created for parameter calculation and handling. This parameter-handling library component executes its functions only if its input changes. The parameter handling component creates the system, input, and output matrices for FCS-MPC. It also processes user-set weighting factors for switching frequency and converter current, capacitor voltage, and grid current.

5.2.1 CPU Implementation

At each time instant kT_s the converter phase currents $\boldsymbol{i}_{c,abc}$ and grid phase voltages $\boldsymbol{v}_{g,abc}$ are sampled. The sampled signals are processed in a feedback handling process, in which reference frame transformations are performed, and the signals are routed as inputs to functions that use them. Several protection functions ensure that the control platform is not operating under dangerous conditions.

An observer is utilized to estimate the grid currents $\boldsymbol{i}_{g,\alpha\beta}$ and filter capacitor voltage $\boldsymbol{v}_{f,\alpha\beta}$. The estimation is done by utilizing the feedback signals $\boldsymbol{i}_{c,\alpha\beta}$ and $\boldsymbol{v}_{g,\alpha\beta}$ to approximate their values using system modeling. The observer introduces delay compensation to the system and state information $\boldsymbol{x}(k)$.

State information will be input to the FCS-MPC algorithm Simulink library. The block calculates reference values for each state in the dq-reference frame. It is assumed that the reference value in the dq-reference frame stays constant for the period of the prediction horizon. Due to this, the references in $\alpha\beta$-reference frame can be transformed from the dq-reference frame by utilizing the grid voltage vector angle of the corresponding time instant.

The total execution time of the control in the CPU, along with delays present in the system, must not exceed the sampling interval of the MPC. Execution overflows are caused when the control platform does not receive new switching instructions before the end of the control period. Execution overflow causes the control platform to stop modulation. Due to this, it is critical to find an operating time step that is sufficiently short for good performance and long enough that the control platform can compute new switch positions for each time step. Based on the system delays, a preliminary sampling time of 50μs is chosen for testing purposes. This time step will allow the control platform to reach switching frequencies higher than the resonance frequency of the LCL filter, and the effects of this resonance are minimized.

5.2.2 System delay analysis

The existing delays in the system are known, and their effect on the feasibility of the control algorithm can be approximated. Existing system delays are shown in table 5.1

Measurement delay occurs when feedback signals are received from Typhoon HIL to the power card. Analog-to-digital conversion is performed when feedback is received by the sensors in the power card. Measurement delay is a constant value of 8μs. The delay accounts for the effect of digital and analog filters within the measurement circuit. The sample is read once the step response of the filters is finished. Uplink communication begins once the set delay has passed. The measurement delay introduces a static delay to the start of the control cycle and thus shortens the total execution time the control platform has available.

Uplink communication delay is caused by data transfer from the power card to the control card. The uplink data transfer delay is significantly longer than the downlink transfer delay due to the amount of data being transferred. Uplink communication information contains all of the feedback measurements. After the data transfer using fiber optics, the data is communicated through FPGA to the CPU. This data-copying step within FPGA is included within the uplink communication delay time, but it is significantly shorter.

Table 5.1. *System delays*

Delay element	Execution Time
Measurement Delay	8 μs
Uplink communication delay	18.1 μs
Downlink communication delay	4.4 μs
Total delay	30.5 μs

Downlink communication delay is caused by triggering the data transfer from the control to the actuators of the switches—the data transfer triggers when the control has found the desired control action. For actuation, only information on switch positions and their on-time is needed.

6. TESTING RESULTS AND ANALYSIS

This chapter presents the testing results and evaluates the performance of the control hardware. The results are evaluated based on MPC algorithm performance in steady-state using single horizon and two-step horizon. Sector restriction is considered for both horizons, and its effect on steady-state performance and execution time is presented. Hardware capability for the CPU implementation is shown using execution times of the FCS-MPC algorithm, control cycle, and total execution time of the control cycle, including delays.

Steady-state operation is tested. The tested controller frameworks in a single-step horizon are MPC with full enumeration, single-sector restriction, and two-sector restriction. Prediction horizon $N_p = 2$ is only tested with a sector-restricted controller framework due to the computational load.

The MPC control algorithm operates in the per-unit (p.u.) system. The per-unit system is presented in Table 6.1, where subscript B denotes the base value. All testing results are stated in per-unit notation unless stated otherwise.

The system parameters are presented in table 6.2. The LCL filter parameters result in a resonance frequency of 1167Hz. The switching is not penalized heavily, and the switching penalty factor λ_u is set to a low value. This results in the average switching frequency being significantly higher than the resonance frequency of the LCL filter, which improves performance. Setting the switching penalty factor to the same across all controller frame-

Table 6.1. *Per-unit system*

Parameter	Symbol	Definition	Base value
Voltage	V_B	$\sqrt{\frac{2}{3}}V_R$	326.6 V
Current	I_B	$\sqrt{2}I_R$	565.69 A
Frequency	ω_B	$2\pi f_g$	314.15 s^{-1}
Impedance	Z_B	$\frac{V_B}{I_B}$	0.5774 Ω
Capacitance	C_B	$\frac{1}{\omega_B Z_B}$	5.513 mF
Inductance	L_B	$\frac{Z_B}{\omega_B}$	1.838 mH

Table 6.2. *System and controller parameters*

Parameter	Symbol	SI Value	Per Unit Value
Grid inductance	L_g	91.43μH	0.0498
Converter-side filter inductance	L_1	148μH	0.0805
Converter-side filter resistance	R_1	1.5mΩ	0.0026
Grid-side filter inductance	L_2	67μH	0.0365
Grid-side filter resistance	R_2	1.5mΩ	0.0026
Filter capacitance	C_f	400μF	0.0726
DC-link voltage reference	$v_{dc,ref}$	650V	1,99
LCL filter resonance frequency	f_{res}	1.167kHZ	
Sampling time	T_s	50μs	
Switching penalty	λ_u	0.001	
Converter current weighting factor	q_1	10	
Filter voltage weighting factor	q_2	150	
Grid current weighting factor	q_3	600	

works does not guarantee precisely the same average switching frequency. This is especially apparent for MPC with sector restriction, so the switching frequency is expected to be slightly lower. This affects both the state waveforms and the grid THD negatively. The goal of the testing is to verify that the controller framework works and discover the execution time of the MPC algorithm.

The weighting factors for MPC are chosen empirically. Grid current reference tracking is prioritized, and thus q_3 is set higher than q_2 and q_1. Converter-side filter current deviates the most from its reference due to switching of the converter. It is prioritized least.

6.1 Steady-state performance

Steady-state performance is evaluated by maintaining the DC voltage at reference. A load corresponding to the nominal power of the control platform is applied to the dc link. The control stabilizes the dc link using an outer voltage control loop, generating a grid current reference using a PI controller. The grid current reference determines state references for filter capacitor voltage and converter current according to 4.3. The control platform's performance is evaluated based on the harmonic contents of the grid current. The average switching frequency of the operation may vary between test cases due to the number of voltage vectors considered. The sampling interval was set to 50μs in a single-step horizon. In the case of prediction horizon $N_p = 2$, the sampling interval had to be increased due to increased computational burden, lowering the average switching frequency.

The waveforms are captured using the Typhoon HIL SCADA capture tool. It samples the signals at a rate of 1MHz. The waveforms are sampled for a more extended period to ensure better accuracy for THD calculation, but two fundamental periods are shown in the figures.

6.1.1 Single-step horizon with full enumeration

The steady-state variable waveforms for MPC with full enumeration are shown in Fig. 6.1. The average switching frequency of the converter is approximately 3.09 kHz. The average total harmonic distortion of the grid current is 2.23%

The waveforms follow the sinusoidal reference, and oscillations caused by switching are significantly dampened within the grid current. LCL filter voltage has very little oscillation. Due to switching, the converter current has very high-frequency oscillations within the waveform.

The harmonic spectrum of the grid current is shown in Fig. 6.2. The figure shows the harmonic spectrum up to a frequency of 4000Hz. It can be observed that the harmonics are not concentrated on any particular harmonic, and the spectrum is even. The highest harmonic components are the fifth, seventh, and eleventh harmonics. Due to the irregular pattern of switching signals, the THD of every phase is not the same but varies slightly. For this reason, the average THD between phases is considered. The switching frequency is much higher than the LCL filter's resonance frequency, which means there is very little harmonic content caused by switching around the resonance frequency.

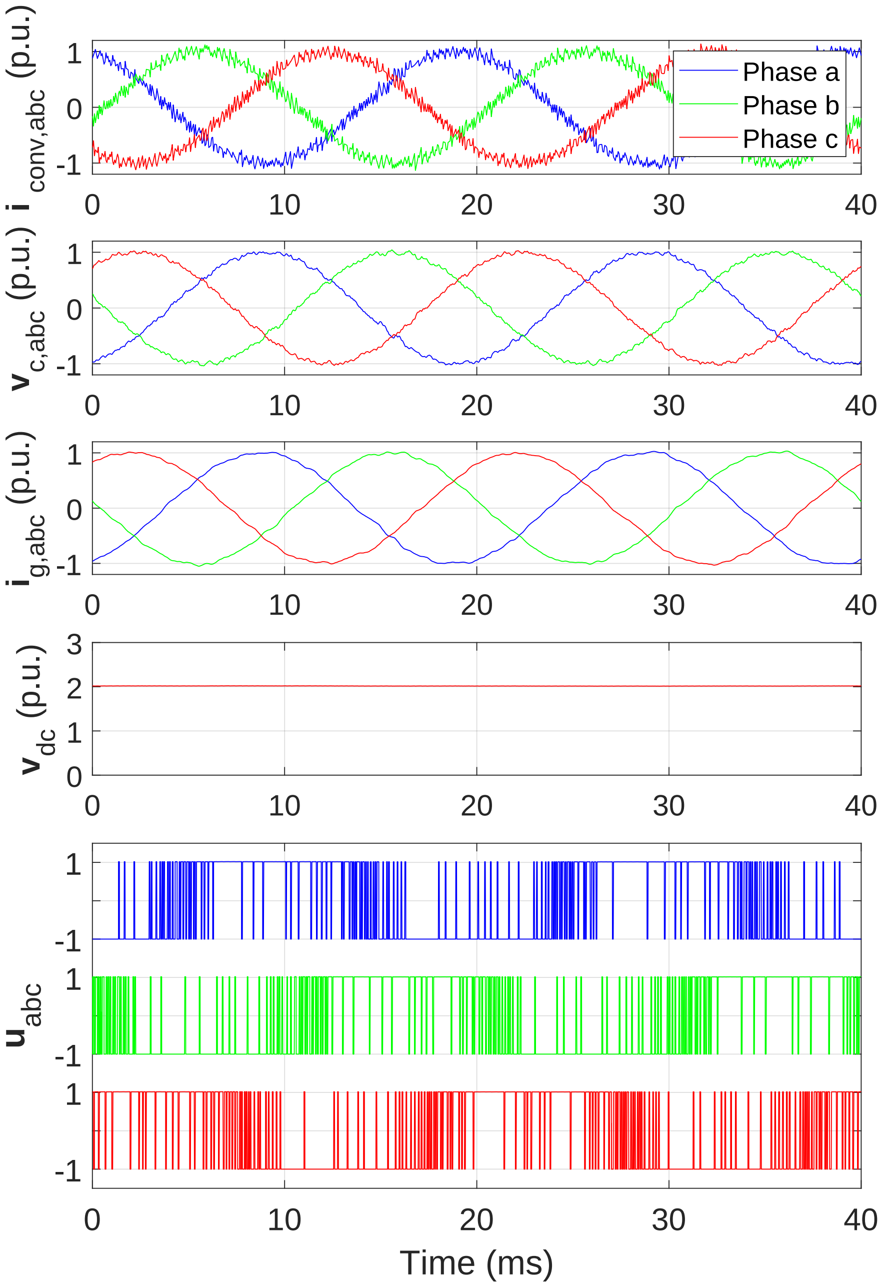


Figure 6.1. *Steady-state with $N_p = 1$, $f_{sw} = 3.09$kHz and full enumeration*

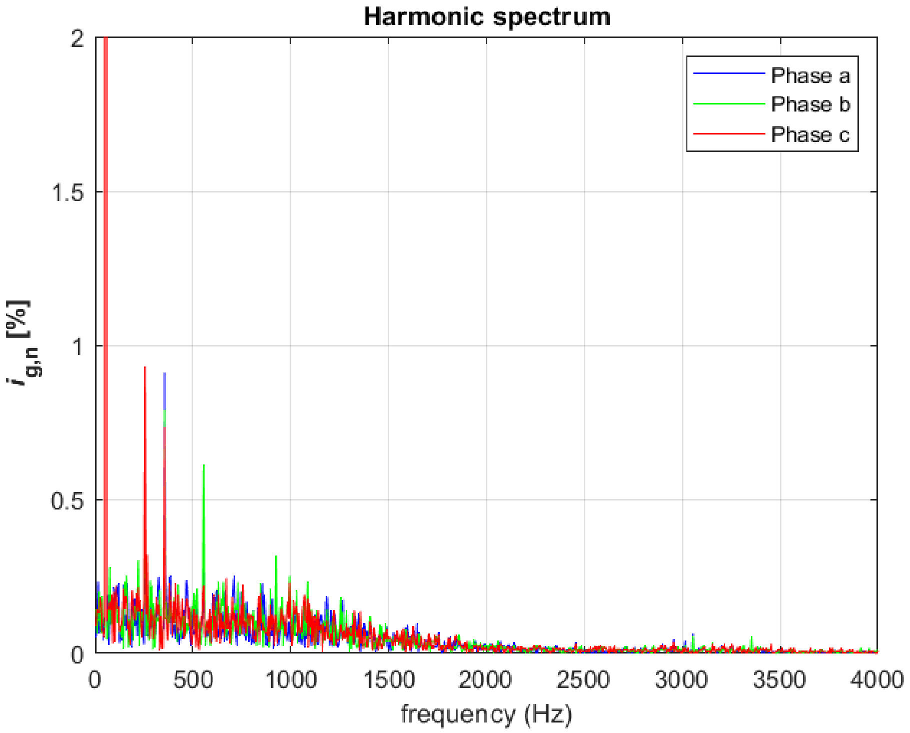


Figure 6.2. *Harmonics of $\boldsymbol{i}_g$ with $N_p = 1$, full enumeration with THD of 2.23%*

6.1.2 Single-step horizon with 1-sector restriction

The steady-state performance of single-sector restricted MPC is shown in Fig. 6.3. The average switching frequency is 2.6kHz, and the THD of the grid current is 3.34%. The controlled variables follow their respective references, but reference tracking is not as accurate compared to MPC with full enumeration. During each fundamental period, there exist six occasions where the algorithm cannot find the optimal solution, resulting in worsened performance. These suboptimal switching patterns are caused when the non-restricted solution to the MPC control problem crosses between sectors. Depending on the situation, the optimal switch position may be one that can not be selected due to sector restriction. This results in a deviation from the reference, causing more harmonic content to be present within the waveforms.

The harmonic spectrum of single sector restricted MPC is presented in Fig. 6.4. The spectrum shows that the magnitude and amount of harmonics within the waveform are much greater than in MPC with full enumeration. Many more spikes exist within the harmonic spectrum, especially at higher frequencies. The switching frequency is still high enough that harmonic components near LCL filter resonance frequency are not seen.

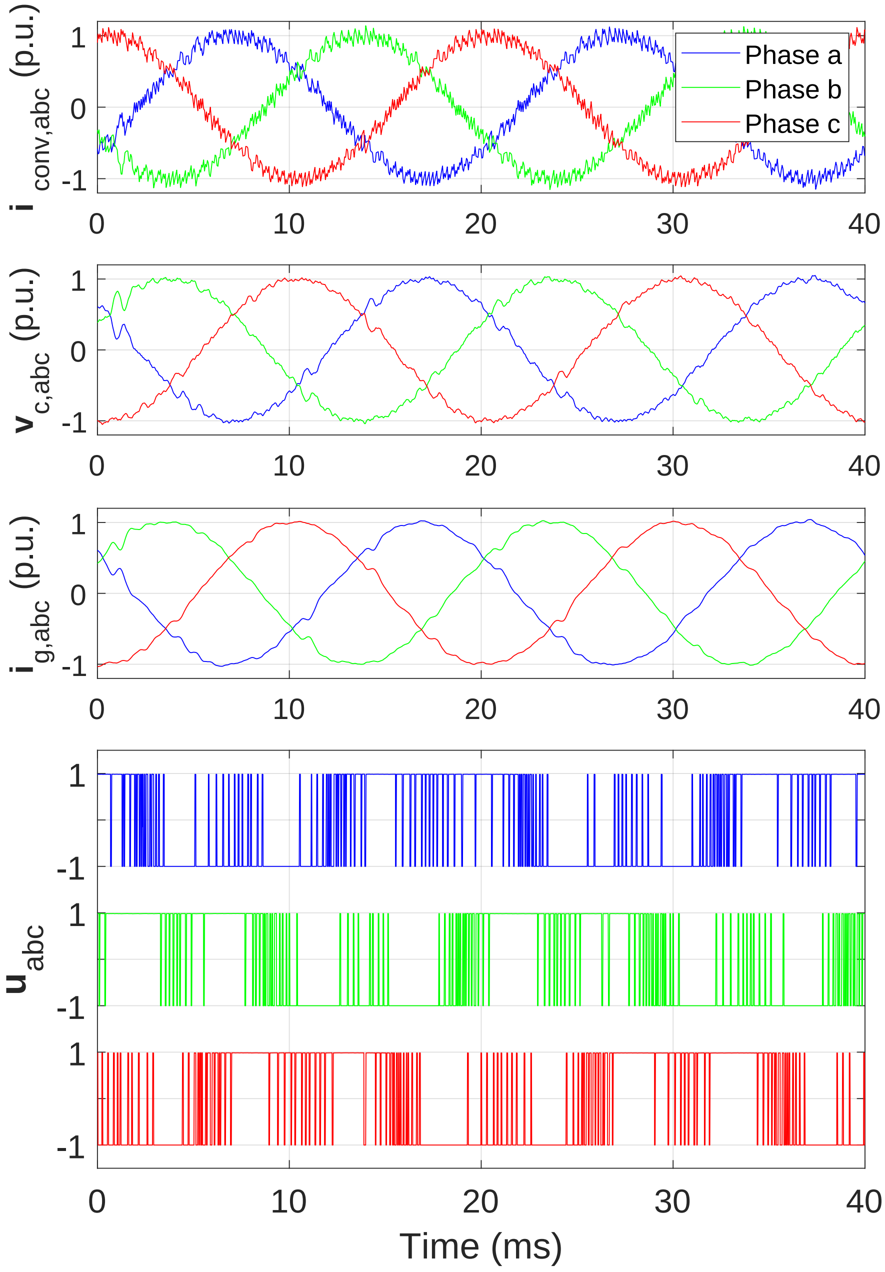


Figure 6.3. *Steady-state with $N_p = 1$, $f_{sw} = 2.6$kHz and 1-sector restricted MPC*

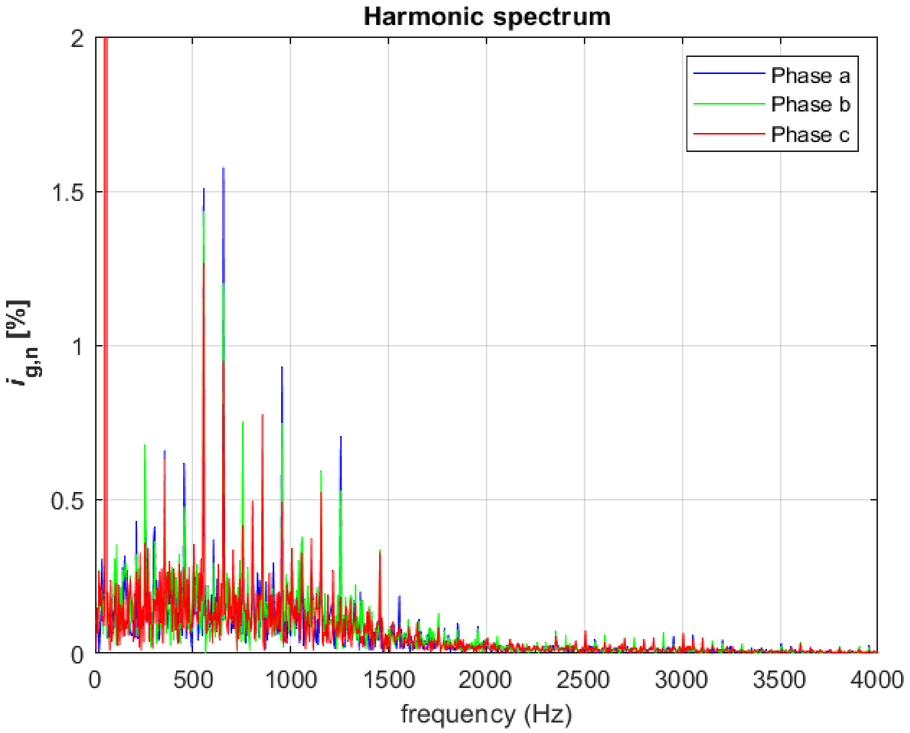


Figure 6.4. *Harmonics of $\boldsymbol{i}_g$ with $N_p = 1$, 1-sector restricted MPC with THD of 3.34%*

6.1.3 Single-step horizon with 2-sector restriction

The steady-state performance of two-sector restricted MPC is shown in Fig. 6.5. The average switching frequency for this controller framework is 3.02kHz and the THD of the grid current is 2.28%. Due to the increased amount of possible switch positions to consider, the algorithm is performing much better when compared to single-sector restricted MPC. The algorithm can choose optimal control action, producing performance comparable to MPC with full enumeration.

The harmonic spectrum for 2-sector restricted MPC is shown in Fig. 6.6. It is very similar when compared with MPC with full enumeration. 5th, 7th, 11th and 13th harmonic have noticeable peaks. The slightly increased THD may be caused by a small degree of suboptimality in the switching. The harmonic components near LCL filter resonance frequency are not amplified.

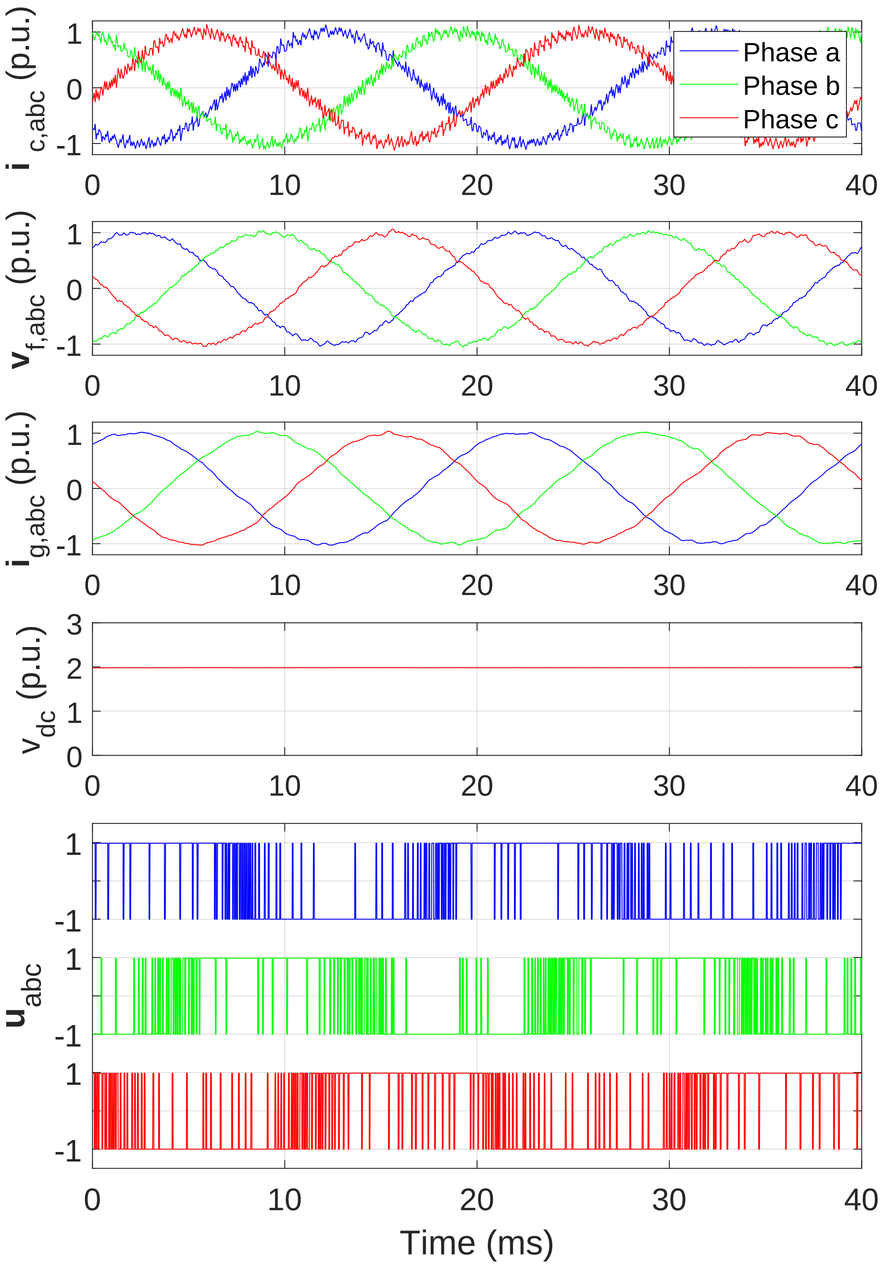


Figure 6.5. *Steady-state with* $N_p = 1$,$f_{sw} = 3.02$*kHZ and 2-sector restricted MPC*

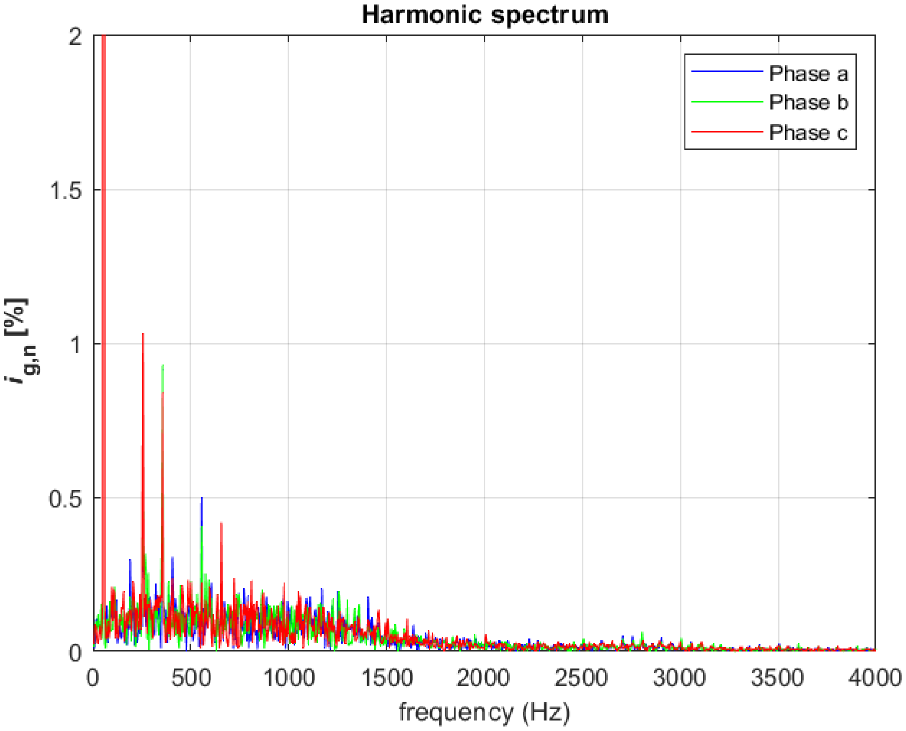


Figure 6.6. *Harmonics of $\boldsymbol{i}_g$ with $N_p = 1$, 2-sector restricted MPC with THD of 2.28%*

6.1.4 Two-step horizon with 1-sector restriction

The steady-state performance of the two-step prediction horizon was evaluated using sector-restricted MPC. Results of both single-sector restricted MPC and two-sector restricted MPC is shown. The computational burden was increased in both cases, resulting in execution time steps longer than the initial 50μs. Due to this, increasing the sampling interval T_s of the system became necessary. The sampling interval was increased to 75μs. The increased sampling interval resulted in lowered switching frequency and worsened reference tracking ability.

Fig. 6.7 displays the performance of single-sector restricted MPC with a two-step prediction horizon. The longer sampling time caused the average switching frequency to drop to around 1.93kHz. Similarly to a single-step horizon, MPC with a single-sector restrictor has suboptimality within its switching. Current oscillations happen at sector crossings.

The harmonic spectrum of sector-restricted MPC operating at $N_p = 2$ is shown in Fig. 6.7. The harmonic spectrum shows that the increased sampling time and lowered switching frequency significantly increased the magnitude of harmonics in the system. The amplitude of higher harmonics rose significantly, especially near LCL filter resonance frequency. It can be seen that harmonic content around LCL filter resonance f_{res} has also been amplified. The THD of the grid current is 5.21%.

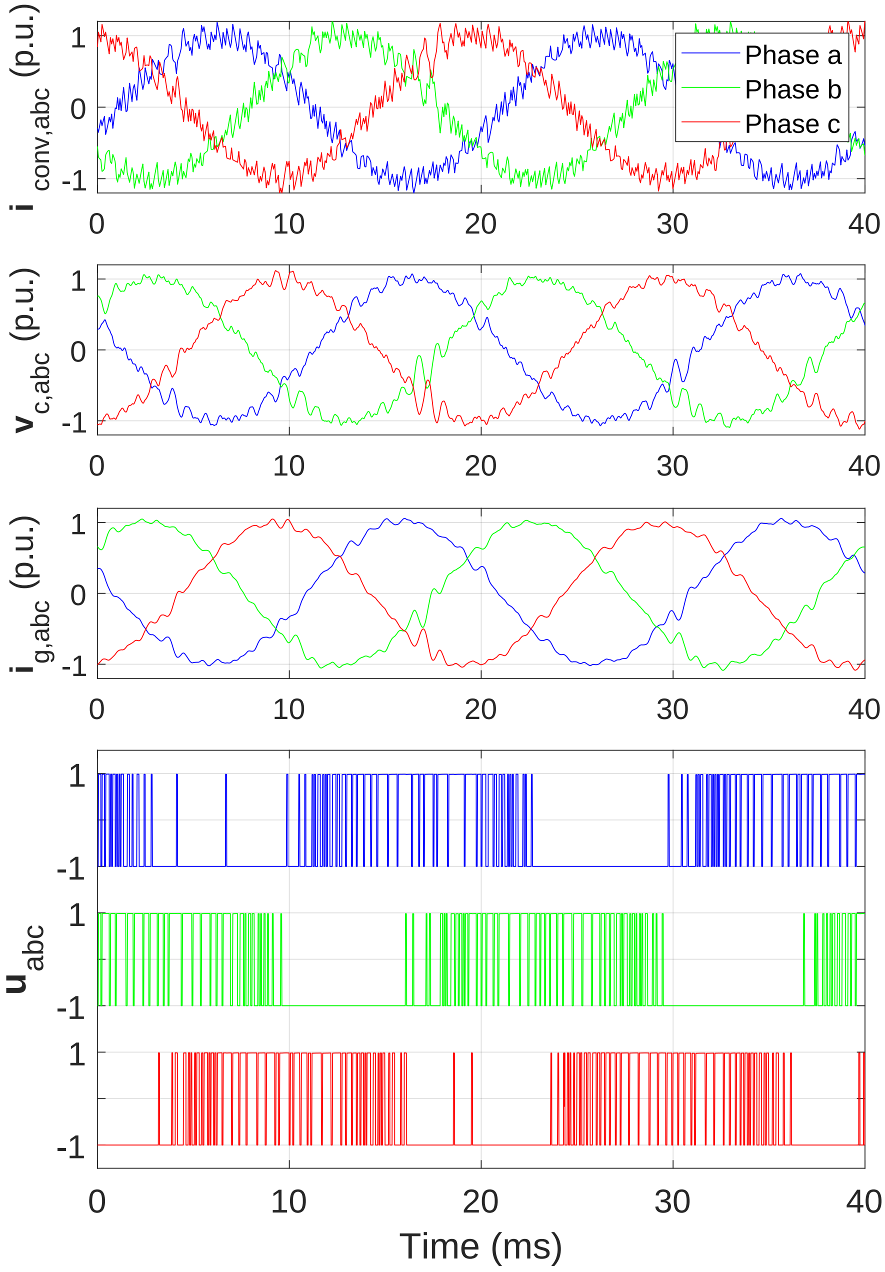


Figure 6.7. *Steady-state with $N_p = 2$,$f_{sw} = 1.93$kHz and 1-sector restricted MPC*

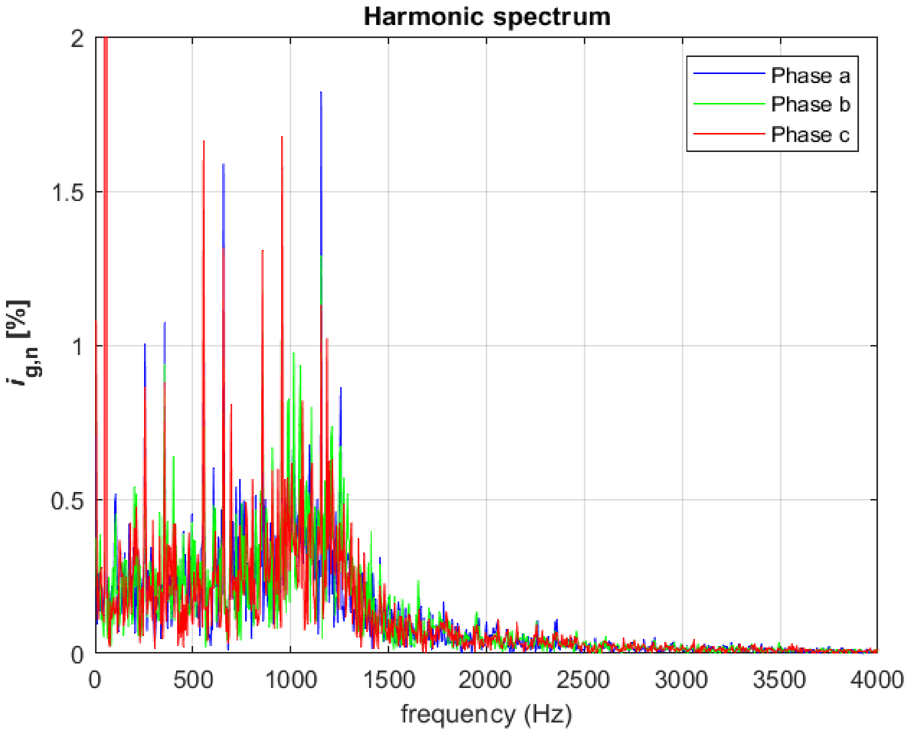


Figure 6.8. *Harmonics of $\boldsymbol{i}_g$ with $N_p = 2$, single-sector restricted MPC with THD 5.21%*

6.1.5 Two-step horizon with 2-sector restriction

The performance of two-sector restricted MPC with prediction horizon $N_p = 2$ is shown in Fig. 6.9. The switching frequency for a two-step horizon with a 2-sector restriction was 1-86kHz. The resulting THD of the operation was 4.44%.

The harmonics for two-sector restricted MPC with prediction horizon $N_p = 2$ are shown in Fig. 6.10. Few noticeable spikes within the harmonic spectrum exist compared to single-sector restricted MPC. Amplification of harmonic components near LCL filter resonance f_{res} can be observed similarly to single-sector restricted MPC with $N_p = 2$. The spectrum is much more spread out, and does not have many individual harmonic spikes that stand out.

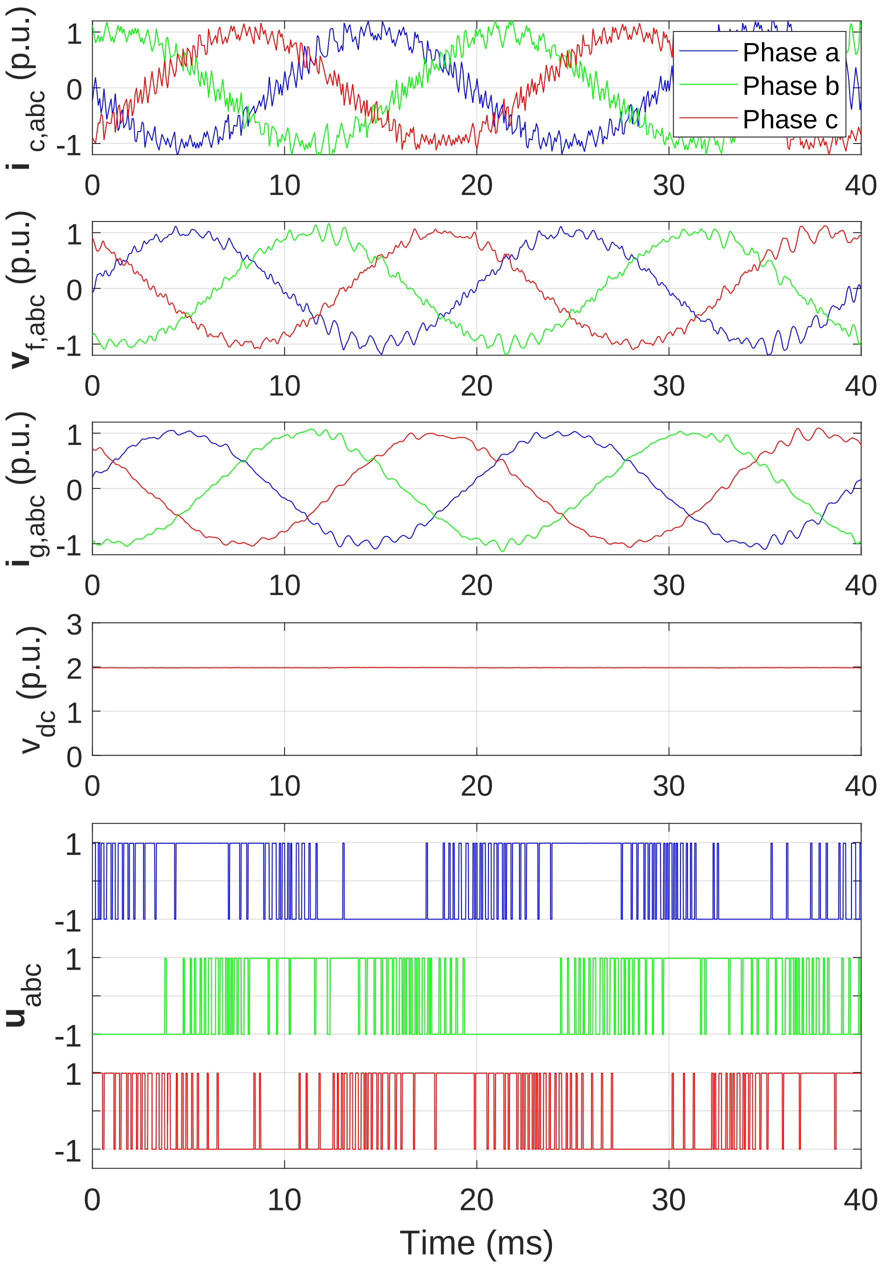


Figure 6.9. *Steady-state with $N_p = 2, f_{sw} = 1.87$kHz and 2-sector enumeration*

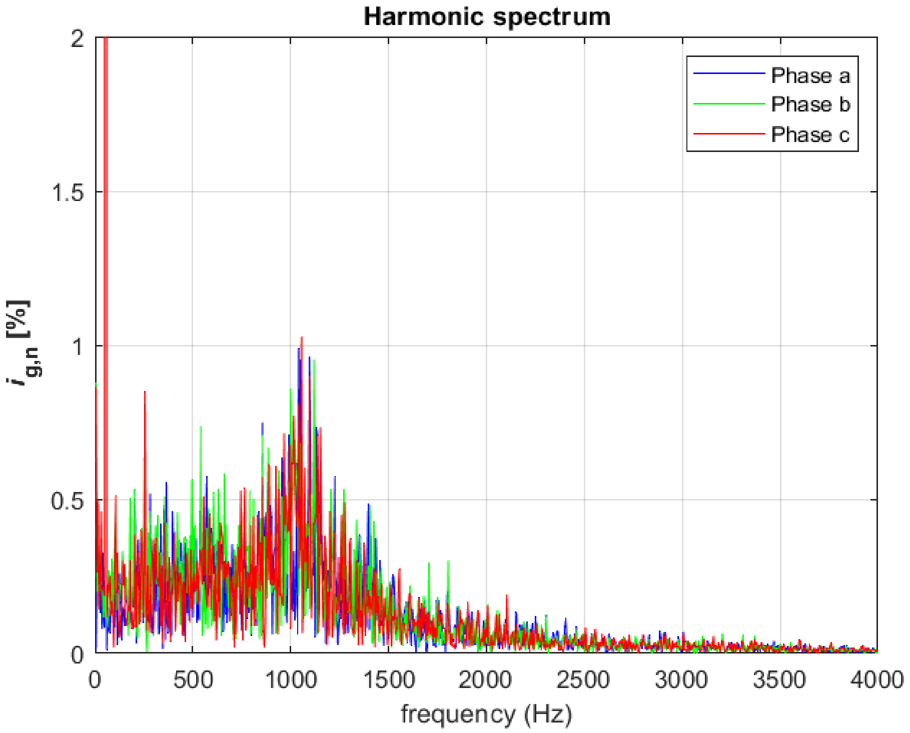


Figure 6.10. *Harmonics of i_g with $N_p = 2$, 2-sector restricted MPC with THD of 4.09%*

6.2 CPU Performance

Computational load is analyzed by identifying the CPU execution time of the various control functions. The control platform has a built-in CPU time analyzer called temporal profiler, which can be used to identify the execution time on the CPU. Using the temporal profiler causes a slight increase in CPU load. This means the actual execution time may be slightly shorter than the identified execution time.

FCS-MPC execution time varies significantly based on how many voltage vectors are considered for any given prediction horizon. In the case of a single horizon, the maximum amount of considered switch positions is 8, with sector restriction bringing this number down to a minimum of 4. The variation between execution times for short-horizon MPC is insignificant due to the limited amount of calculations performed during the control loop.

For Horizons of longer than 1, the execution time of FCS-MPC significantly increases. The considered switch positions making up the switching sequence can increase up to 2^{3N_p} if no restriction on the number of switch positions is considered. Only sector-restricted cases were considered for two-step horizon due to the computational effort solving the optimization problem would require. This requires 16 switch positions to be considered for single-sector restricted MPC and 25 for two-sector restricted MPC.

The execution times for the MPC algorithm, control loop, and total execution time are presented in Table 6.3. The table shows the minimum, maximum, and average execution

	FCS-MPC			Control loop			Total with delays		
FCS-MPC	Min	Max	Avg	Min	Max	Avg	Min	Max	Avg
$N_p = 1$, full enum	9.96	10.84	10.22	15.87	21.93	17.43	45.37	51.45	46.93
$N_p = 1$, 1-sector	5.71	6.71	5.99	12.14	18.26	13.86	41.64	47.76	43.36
$N_p = 1$, 2-sector	7.37	8.74	7.78	13.31	20.41	15.41	42.81	49.31	44.91
$N_p = 2$, 1-sector	24.59	25.87	24.95	30.59	38.73	32.6	60.09	68.73	62.10
$N_p = 2$, 2-sector	36.61	37.96	37.08	43.31	49.67	45.66	72.81	79.17	75.16

Table 6.3. *Execution times of FCS-MPC, control cycle and delays in μs*

time of the MPC control functionality, the sum of the execution time of the control components, and the total execution time of the system, including delays presented in Table 5.1. The control loop time is calculated as a sum of its parts. These include, e.g. feedback handling, protection functions, reference calculation, and observer. Each component has an average, minimum, and maximum execution time. The execution times are combined to compile the theoretical average, maximum and minimum values.

The variation of maximum and minimum execution time for the FCS-MPC algorithm is relatively small. The low variation is expected due to the fixed amount of switch positions considered. The effect can be seen when the results of full enumeration are compared with sector-restricted MPC. Total control loop time has more variation within its execution time. The minimum and maximum values variation may reach an 8μs difference.

The maximum execution time is an important parameter to consider for the viability of the control strategy. The total maximum execution time should not exceed the sampling time T_s set by the user.

In the case of single horizon MPC, the average execution time with delays remained under the user-set $50\mu s$ sample time. The average execution time has a few microsecond margins of error to ensure that the sample time is not exceeded. In comparison, for maximum execution time, the margin between the actual execution time and sample time is significantly reduced, barely fitting within the set sample time. In the case of MPC with full enumeration, it even exceeds the sample time. It should be noted, however, that the actual execution time may be shorter due to execution time profiling affecting the results slightly.

When the 2-step horizon was considered, FCS-MPC algorithm execution time increased significantly. Due to this, the original 50μs time step needed to be longer for the control hardware. The longer time step caused worse THD and harmonic content in the state variables.

In conclusion, the CPU implementation can support MPC with a single horizon. Further optimizations to different areas of CPU implementations could be made, resulting in execution time savings of a few microseconds. However, FCS-MPC execution time largely depends on the number of switch positions considered within the calculation, which increase exponentially when the prediction horizon is extended.

7. CONCLUSIONS

MPC is a control algorithm in which a system model, optimization problem, and receding horizon policy are used to determine the optimal control actions. The system model predicts future system state as a function of switching patterns and previous state information. Optimization is performed using objective function and constraints. The objective function is used to quantify the control objectives into a single scalar value, which can be used to determine the optimal plan of switch positions. Only the first element of the plan is executed, and a new optimal plan is calculated during the next step. The constraints restrict the behavior into a set of rules governing the control algorithm.

An MPC algorithm for controlling a grid-connected two-level converter was proposed and implemented. The control objective of the FCS-MPC algorithm was simultaneous control of LCL filter currents and voltages according to reference generated from dc-link voltage control. The proposed algorithm controlled the switches of the two-level converter directly and did not require a separate modulator.

The implemented control algorithm can control the states of the system using a 50μs sampling interval in a single-horizon operation mode. Due to increased computational burden, the algorithm could not perform the prediction horizon of two steps within 50μs sampling interval. For prediction horizon of $N_p = 2$, the sampling interval had to be increased to 75 μs . Longer sampling intervals directly affected the average switching frequency lowering it. Consequently, this caused worsened performance in operation.

The joint project between Tampere University and Danfoss Drives this thesis is related to, OPT4MAX, which continues the research on MPC on the control platform. Based on the results of this thesis, it has become apparent that for longer prediction horizons, the computational power of the CPU is not enough to solve the optimization problem within a short enough time. However, the system delays alongside other components take a short enough time to consider an FPGA-based implementation on the system-on-chip. Due to its ability to pipeline calculations, the optimization problem can be solved more quickly. Due to this, longer prediction horizons may be achievable in an FPGA-based implementation.

REFERENCES

[1] O. Ellabban, H. Abu-Rub, and F. Blaabjerg, “Renewable energy resources: Current status, future prospects and their enabling technology”, *Renewable & sustainable energy reviews*, vol. 39, pp. 748–764, 2014, ISSN: 1364-0321.

[2] E. F. Camacho, *Model Predictive Control*, 2nd ed. 2007., ser. Advanced Textbooks in Control and Signal Processing. London: Springer London, 2007, ISBN: 0-85729-398-2.

[3] P. Karamanakos, E. Liegmann, T. Geyer, and R. Kennel, “Model predictive control of power electronic systems: Methods, results, and challenges”, *IEEE Open Journal of Industry Applications*, vol. 1, pp. 95–114, 2020. DOI: 10.1109/OJIA.2020.3020184.

[4] B. Stellato, T. Geyer, and P. J. Goulart, “High-speed finite control set model predictive control for power electronics”, *IEEE Transactions on Power Electronics*, vol. 32, no. 5, pp. 4007–4020, 2017. DOI: 10.1109/TPEL.2016.2584678.

[5] T. Dorfling, H. du Toit Mouton, T. Geyer, and P. Karamanakos, “Long-Horizon Finite-Control-Set Model Predictive Control With Nonrecursive Sphere Decoding on an FPGA”, *IEEE Transactions on Power Electronics*, vol. 35, no. 7, pp. 7520–7531, 2020. DOI: 10.1109/TPEL.2019.2956213.

[6] J. Scoltock, T. Geyer, and U. Madawala, “Model Predictive Direct Power Control for a grid-connected converter with an LCL-filter”, in *2013 IEEE International Conference on Industrial Technology (ICIT)*, 2013, pp. 588–593. DOI: 10.1109/ICIT.2013.6505737.

[7] M. Rossi, P. Karamanakos, and F. Castelli-Dezza, “An indirect model predictive control method for grid-connected three-level neutral point clamped converters with lcl filters”, 2022.

[8] “Design and control of an LCL-filter-based three-phase active rectifier”, *IEEE transactions on industry applications*, vol. 41, no. 5, pp. 1281–1291, 2005, ISSN: 0093-9994.

[9] S. Qin and T. A. Badgwell, “A survey of industrial model predictive control technology”, eng, *Control engineering practice*, vol. 11, no. 7, pp. 733–764, 2003, ISSN: 0967-0661.

[10] M. Morari and J. H. Lee, “Model predictive control: Past, present and future”, *Computers & chemical engineering*, vol. 23, no. 4, pp. 667–682, 1999, ISSN: 0098-1354.

[11] T. Geyer, *Model predictive control of high power converters and industrial drives*, eng, First edition. Chichester, West Sussex, United Kingdom: John Wiley & Sons, Incorporated, 2017, ISBN: 1-5231-1478-9.

[12] R. Kennel and D. Schröder, “Predictive control strategy for converters”, *IFAC Proceedings Volumes*, vol. 16, no. 16, pp. 415–422, 1983, 3rd IFAC Symposium on Control in Power Electronics and Electrical Drives, Lausanne, Switzerland, 12-14 September, 1983, ISSN: 1474-6670. DOI: https://doi.org/10.1016/S1474-6670(17)61897-4. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1474667017618974.

[13] S. Vazquez, J. I. Leon, L. G. Franquelo, *et al.*, “Model predictive control: A review of its applications in power electronics”, *IEEE industrial electronics magazine*, vol. 8, no. 1, pp. 16–31, 2014, ISSN: 1932-4529.

[14] J. Rodriguez, M. P. Kazmierkowski, J. R. Espinoza, *et al.*, “State of the art of finite control set model predictive control in power electronics”, *IEEE transactions on industrial informatics*, vol. 9, no. 2, pp. 1003–1016, 2013, ISSN: 1551-3203.

[15] J. Rodríguez Pérez and P. Cortes, *Predictive control of power converters and electrical drives*, eng, 1st edition, ser. Wiley - IEEE ; 44. Hoboken, New Jersey: Wiley, 2012, ISBN: 1-119-94264-0.

[16] P. Karamanakos, T. Geyer, and R. Kennel, “On the choice of norm in finite control set model predictive control”, *IEEE Transactions on Power Electronics*, vol. 33, no. 8, pp. 7105–7117, 2018. DOI: 10.1109/TPEL.2017.2756092.

[17] C. Bordons and C. Montero, “Basic principles of mpc for power converters: Bridging the gap between theory and practice”, *IEEE industrial electronics magazine*, vol. 9, no. 3, pp. 31–43, 2015, ISSN: 1932-4529.

[18] T. Geyer, P. Karamanakos, and R. Kennel, “On the benefit of long-horizon direct model predictive control for drives with LC filters”, in *2014 IEEE Energy Conversion Congress and Exposition (ECCE)*, 2014, 3520R–3527. DOI: 10.1109/ECCE.2014.6953879.

[19] P. Karamanakos and T. Geyer, “Guidelines for the design of finite control set model predictive controllers”, *IEEE Transactions on Power Electronics*, vol. 35, no. 7, pp. 7434–7450, 2020. DOI: 10.1109/TPEL.2019.2954357.

[20] M. Siami, D. A. Khaburi, and J. Rodriguez, “Simplified Finite Control Set-Model Predictive Control for Matrix Converter-Fed PMSM Drives”, *IEEE transactions on power electronics*, vol. 33, no. 3, pp. 2438–2446, 2018, ISSN: 0885-8993.

[21] Z. Zhang, C. M. Hackl, and R. Kennel, “Computationally Efficient DMPC for Three-Level NPC Back-to-Back Converters in Wind Turbine Systems With PMSG”, *IEEE transactions on power electronics*, vol. 32, no. 10, pp. 8018–8034, 2017, ISSN: 0885-8993.

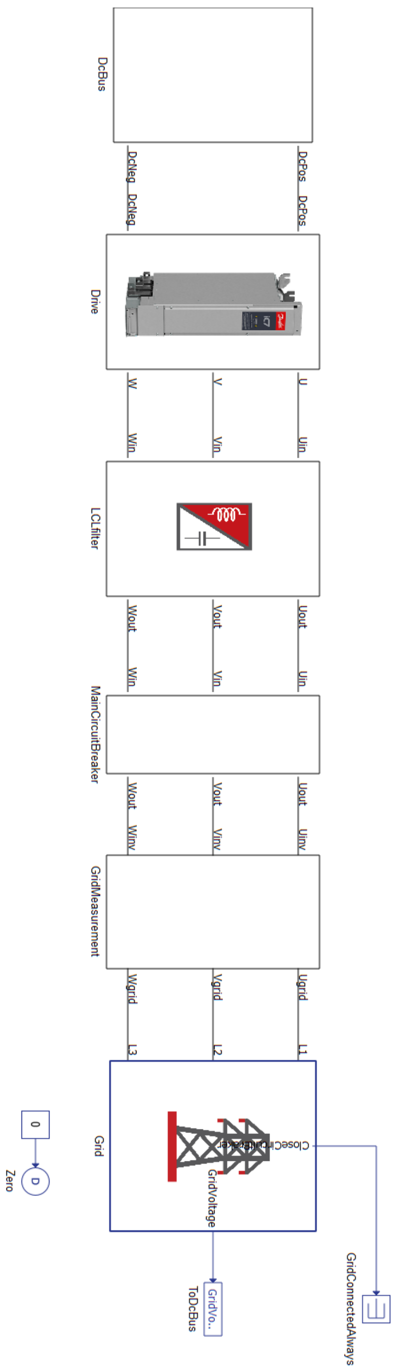
[22] “Typhoon HIL Documentation”. (2023), [Online]. Available: https://www.typhoon-hil.com/documentation/.

[23] “Typhoon HIL schematic editor documentation”. (2023), [Online]. Available: https://www.typhoon-hil.com/documentation/typhoon-hil-software-manual/concepts/schematic_editor.html.

[24] "Typhoon HIL SCADA documentation". (2023), [Online]. Available: https://www.typhoon-hil.com/documentation/typhoon-hil-software-manual/concepts/hil_scada.html.

[25] MathWorks. "Model-based design with Simulink". (2022), [Online]. Available: https://se.mathworks.com/help/simulink/gs/model-based-design.html.

APPENDIX A: HIL MODEL


DcBus
DcPos
DcPos
DcNeg
DcNeg
Drive
U
V
W
Uin
Vin
Win
LCLfilter
Uout
Vout
Wout
Uin
Vin
Win
MainCircuitBreaker
Uout
Vout
Wout
Uinv
Vinv
Winv
GridMeasurement
Ugrid
Vgrid
Wgrid
L1
L2
L3
Grid
CloseCircuitBreaker
GridVoltage
0
D
Zero
GridVo..
ToDcBus
GridConnectedAlways