UNIVERSITÀ DEGLI STUDI DELL'AQUILA
DIPARTIMENTO DI
INGEGNERIA E SCIENZE DELL'INFORMAZIONE E MATEMATICA

Dottorato di Ricerca in Information and Communication Technology
Curriculum Systems Engineering, telecommunications and HW/SW platforms

XXXVI ciclo

Direct Model Predictive Control of Cascaded H-Bridge Inverters

SSD ING-IND/32

Dottorando
Francesco Simonetti

Coordinatore del corso
Ch.mo Prof. Vittorio Cortellessa

Tutor
Ch.mo Prof. Carlo Cecati

Co-Tutor
Prof. Alessandro D'Innocenzo

a.a. 2022/2023

Summary

Model predictive control (MPC) has gained increasing attention in the last decades in the power electronics field. Among the different versions of MPC, the most popular one is the so-called finite control set MPC (FCS-MPC or direct MPC), which does not require a modulation algorithm and is characterized by a fast dynamic performance. However, it suffers from high computational cost when applied to multilevel converters, and it requires a large amount of calculations that grows when the number of levels increases, which affects the practical real-time implementation and performance.
This thesis addresses the challenge of reducing the computational burden of the FSC-MPC for cascaded H-bridge (CHB) multilevel inverters.
Neural networks were investigated to significantly speed up the computation of the control law while guaranteeing satisfactory performance.
An analytical solution was also developed to dramatically decrease the number of calculations needed to compute the optimal control.
A CHB static synchronous compensator (CHB-STATCOM) was employed as a test bench for the presented studies. Theoretical discussions of the proposed methodologies are provided and comparisons among them and existing methods are given, underlying that the presented methods overcome the state-of-the-art approaches. The proposed solutions were analyzed in a simulation environment by using Matlab/Simulink, through hardware in the loop and, finally, implemented on a 5-level CHB-STATCOM prototype. The control techniques were fully implemented on FPGA, while the digital signal processors on the individual H-bridges were programmed to sample voltages and currents.
The thesis describes the firmware implementation and the most critical encountered problems and the related solutions are discussed. Specifically, the use of current transformer sensors to sense the three-phase currents was the cause of a DC offset in the output currents. A software solution was developed to solve this problem by adding an MPC optimization problem to the overall algorithm. Practical issues of the serial peripheral interface protocol for transferring the measurements to the master FPGA are presented and a solution for minimizing the communication delays is presented.

Contents

List of Algorithms

List of Figures

List of Tables

List of Acronyms

ADC Analog to digital converter

ASM Algorithmic state machine

CCS-MPC Continuous control set model predictive control

CHB Cascaded H-bridge

CT Current transformer

DSP Digital signal processor

FACT Flexible alternating current transmission system

FCS-MPC Finite control set model predictive control

FIFO First in first out

FPGA Field programmable gate array

HIL Hardware in the loop

IGBT Insulated-bipolar gate transistor

IP Intellectual property

ISR Interrupt service routine

LSB Less significant bit

LUT Look-up table

MAE Mean absolute error

ML Machine learning

MPC Model predictive control

MSB Most significant bit

NN Neural network

PCC Point of common coupling

PI Proportional-integral

PLL Phase-locked loop

PV Photovoltaic

PWM Pulse width modulation

RTL Register transfer level

SOC Start of communication

SPI Serial peripheral interface

SPIDAT Transmission data register

SPITXBUF Transmission buffer register

SPI-CLK SPI clock

SPI-CS SPI chip select

SPI-MISO SPI master input slave output

SPI-MOSI SPI master output slave input

STATCOM Static synchronous compensator

THD Total harmonic distortion

TX-FIFO Transmission FIFO buffer

UART Universal asynchronous receiver-transmitter

VHDL Very high speed integrated circuits hardware description language

List of Symbols

ω	Angular speed of grid voltage, $2\pi 50$ rad/s
n	Number of H-bridges per phase
R	Internal resistance of the filter inductor
L	Inductance of the filter inductor
T_s	Sampling interval period
C	Capacitance of the DC-link capacitors
$C_{eq} = C/n$	Equivalent capacitance of the cluster
$\mathbf{i}_{\alpha,\beta} = [i_\alpha, i_\beta]^T$	Output current vector in α, β coordinates
$\mathbf{i}^*_{\alpha,\beta} = [i^*_\alpha, i^*_\beta]^T$	Reference output current vector in α, β coordinates
$\mathbf{i}^*_{d,q} = [i^*_d, i^*_q]^T$	Reference output current vector in d, q coordinates
$\mathbf{v}_{s(\alpha,\beta)} = [v_{s\alpha}, v_{s\beta}]^T$	Grid voltage vector in α, β coordinates
$\mathbf{v}_{\alpha,\beta} = [v_\alpha, v_\beta]^T$	Output voltage vector in α, β coordinates
V_{DC}	DC-link rated voltage
$\mathbf{Q} = q_i \times \mathbf{I}_2$	Weigth matrix for currents vector in current control
$\mathbf{P} = p_i \times \mathbf{I}_2$	Weigth matrix for switching vector in current control
$\mathbf{S}_{\alpha,\beta} = [S_\alpha, S_\beta]^T$	Switching vector in α, β coordinates
$\mathbf{S}^*_{\alpha,\beta} = [S^*_\alpha, S^*_\beta]^T$	Optimal switching vector in α, β coordinates
$\mathbf{V}_{\alpha,\beta}$	Set of feasible switching vectors in α, β coordinates
v_{Cpi}	Individual DC-link voltage of i-th capacitor of phase p, with $i = 1, ..., n$ and $p \in \{a, b, c\}$
s_{pi}	Switching variable of i-th H-bridge of phase p, with $i = 1, ..., n$ and $p \in \{a, b, c\}$

s^*_{pi}	Optimal switching variable of i-th H-bridge of phase p, with $i = 1, ..., n$ and $p \in \{a, b, c\}$
i_p	Output current of phase $p \in \{a, b, c\}$
v_{sp}	Grid voltage of phase $p \in \{a, b, c\}$
v_p	Output voltage of phase $p \in \{a, b, c\}$
S_p	Switching vector of phase $p \in \{a, b, c\}$
S^*_p	Optimal switching vector of phase $p \in \{a, b, c\}$
$\mathbf{V}^{DC} = V_{DC} \cdot \mathbf{1}_n$	$n \times 1$ vector containing V_{DC} in each element
$\mathbf{v}_{Cp} = [v_{Cp1}, ..., v_{Cpn}]^T$	DC-link capacitor voltages vector of phase $p \in \{a, b, c\}$
$\mathbf{s}_p = [s_{p1}, ..., s_{pn}]^T$	Vector of switching variables of phase $p \in \{a, b, c\}$
$\mathbf{s}^*_p = [s^*_{p1}, ..., s^*_{pn}]^T$	Vector of optimal switching variables of phase $p \in \{a, b, c\}$
$\mathbf{Q}_{ib} = q_{ib} \times \mathbf{I}_n$	Weigth matrix for voltages vector in individual voltages balance
$\mathbf{P}_{ib} = p_{ib} \times \mathbf{I}_n$	Weigth matrix for switching variables in individual voltages balance
$\mathbf{S}_{a,b,c} = [S_a, S_b, S_c]^T$	Switching vector in a, b, c coordinates
$\mathbf{S}^*_{a,b,c} = [S^*_a, S^*_b, S^*_c]^T$	Optimal switching vector in a, b, c coordinates
$\mathbf{S}^0_{a,b,c} = [S^0_a, S^0_b, S^0_c]^T$	Switching vector in a, b, c coordinates with zero homopolar component
$\mathbf{V}^{DC}_{a,b,c} = [V_{DC}, V_{DC}, V_{DC}]^T$	3×1 vector containing V_{DC} in each element
$\mathbf{v}_{C(a,b,c)} = [v_{Ca}, v_{Cb}, v_{Cc}]^T$	Cluster voltages vector
$\mathbf{Q}_{cb} = q_{cb} \times \mathbf{I}_3$	Weigth matrix for voltages vector in cluster voltages balance
$\mathbf{P}_{cb} = p_{cb} \times \mathbf{I}_3$	Weigth matrix for switching vector in cluster voltages balance
w_{cb}	Weigth coefficient for homopolar component in cluster voltages balance
$\mathbf{V}_{a,b,c}$	Set of feasible switching vectors in a, b, c coordinates
$\mathbf{T}_{2\times3}$	2×3 Clarke transformation matrix
$\mathbf{T}^{-1}_{2\times3}$	Pseudo-inverse of the 2×3 Clarke transformation matrix
$\mathbf{out}_{\mathrm{NN}_{\alpha,\beta}} = [out_{\mathrm{NN}_\alpha}, out_{\mathrm{NN}_\beta}]^T$	Output of the NN currents controller

$\mathbf{S}_{\alpha,\beta}^{60}$	Neural network currents controller output tranformed into the auxiliar reference frame
$\mathbf{T}_{60}$	Transformation matrix for the auxiliar reference frame
$\mathbf{S}_{\alpha,\beta}^{c}$	Solution to the consintuous unconstrained currents control problem
$\mathbf{S}_{\alpha,\beta}^{proj}$	Solution to the consintuous constrained currents control problem
η	Number of hidden neurons of the NN
h	Number of prediction horizons of the currents MPC
$\mathbf{C}_{cb}$	Costs array of the clusters voltages balance
$\mathbf{C}_{ib,p}$	Costs array of the individual voltages balance of phase $p \in \{a.b.c\}$
$\mathbf{C}_{ib,p}^{*}$	Sorted costs array of the individual voltages balance of phase $p \in \{a.b.c\}$
$\mathbf{I}_{p}$	Indexes of the sorted costs array of the individual voltages balance of phase $p \in \{a.b.c\}$
i_{Cp}	Current on the DC side of cluster $p \in \{a, b, c\}$
i_{lp}	Current loss on the DC side of cluster $p \in \{a, b, c\}$
$\tilde{S}$	Amplitude of the line frequency component of S_p
ϕ_S	Phase of the line frequency component of S_p
$\overline{S}$	Amplitude of the constant component of S_p
$\tilde{I}_S$	Amplitude of the line frequency component of i_p
ϕ_I	Phase of the line frequency component of i_p
$\overline{I}_S$	Amplitude of the constant component of i_p
$\tilde{I}_C$	Amplitude of the line frequency component of i_{Cp}
ϕ_{I_C}	Phase of the line frequency component of i_{Cp}
$\overline{v}_C$	Average capacitors voltage
$K_P^{i_d}$	Proportional gain i_d reference generator
$K_I^{i_d}$	Integral gain i_d reference generator

Chapter 1

Introduction

1.1 Background

1.1.1 Model Predictive Control

Model predictive control (MPC) has established itself as one of the most powerful and versatile advanced control techniques in many engineering fields [1, 2]. The MPC idea is to formulate the control problem as an optimization problem. It employs the dynamical model of the system to predict its future behavior and it computes the optimal control action by taking into account the physical constraint of the system. It is capable to provide enhanced performance compared to traditional methods and other advanced control techniques and it is characterized by the following features.

- It relies on the use of a dynamical model of the system. Compared to the traditional proportional-integral-derivative regulator or other linear control techniques, MPC can handle complex, nonlinear, and time-varying systems, often encountered in many engineering fields. Furthermore, it inherently allows multi-variable control, unlike traditional linear controllers.

- It computes the control action by solving an optimization problem. The MPC involves the minimization of a cost function, which can include various performance metrics. Different cost function can be designed for specific requirements, allowing for a great flexibility and making the MPC a powerful method for a wide range of applications.

- It can predict the future behavior of the system. One of the key features of the MPC is the ability to predict the future output variables in order to optimize the performance of the system over a time window.

- It can handle the physical constraints of the system. The formulation of the control problem as an optimization problem makes it easy to handle constraints. This ensures that the control action computed by MPC not only optimizes the performance metrics but also respects the physical limitations of the system, unlike traditional control strategies that struggle to deal with constraints in a systematic way.

1.1.2 Power Electronics and Power Electronic Converters

Power electronics is a pivotal field of modern electrical engineering. It comprises all the electronics technologies that aim to manipulate the electrical energy. It is an extremely interdisciplinary field that merges concepts from many engineering areas such as electronics, control systems, chemistry, informatics. The core of power electronics is the use of semiconductor devices that can be electrically turned ON and OFF to close or open the electrical circuit. The opportune control of these switching devices allows to start or interrupt the power supplied to a load, to connect/disconnect electrical components to/from the circuit and, in general, to control the energy flow.
The proper arrangement of solid-state switching elements with passive electrical components allows the realization of the power electronic converters, devices specifically designed to control energy, capable of converting electrical energy from one form to another. Different converter topologies were designed to convert direct voltage into alternating voltage or vice-versa, or to increase or decrease the voltage value to meet the system requirements. By integrating an appropriate control system, they are able to dynamically regulate the power flow and adjust voltage and current magnitudes and frequencies to cope with system disturbances and continuously satisfy the application needs regardless of the operating conditions. They are used in :

- power distribution and transmission networks, to increase the robustness of the grid by regulating the voltage level of the node, to increase the efficiency of the energy transfer by adjusting the power factor;
- renewable energy systems, to convert the unpredictable energy produced by solar or wind power plants into stable energy and to interface the power plant with the grid by supplying the expected voltage values of magnitude and frequency;
- electric vehicles, to convert the DC voltage supplied by the batteries to the AC three-phase voltage required by the motor and to precisely control the current profile to reach the desired speed and torque;
- industrial automation, to control the motor drives to improve efficiency and precision of production processes;
- consumer electronics, to convert the AC voltage available from the wall plug to the specific voltage needed by the everyday life devices, such as smartphones, personal computers, home appliances, etc.

1.1.3 Multilevel Power Electronic Converters

Multilevel power electronic converters are the state-of-the-art evolution of traditional converters. The key feature is the ability to produce multiple voltage levels in the output waveform, overcoming the limitations of the two or three voltage levels of classical power converters, at the expense of a more complex topology, a higher number of components and the need for more sophisticated control algorithms. The main features of multilevel converters are the following.

- They can achieve multiple voltage levels by employing multiple voltage sources and switching devices, resulting in a greater granularity in the output voltage waveform,

which leads to a reduced harmonic distortion and a higher power conversion efficiency. The multiple levels offer the possibility to reach higher total voltage levels compared to traditional converters, resulting in a reduction in the size of the step-up transformer or the possibility to eliminate the transformer, thereby reducing costs and improving efficiency.

- They lead to a reduced voltage stress on the circuit elements, since the total voltage is spit among different sources and the semiconductor devices and passive elements operate at a fraction of the overall voltage, resulting in improved reliability and longer life of the components. It leads to smaller size and lower cost of the components, at the expense of a higher number of needed elements. A trade-off between the number of components and their size is required to optimize the overall cost.

- The use of multiple devices leads to a reduction of the switching frequency of the individual component, a decrease of the switching losses and a higher efficiency of the overall converter.

Many multilevel topologies were developed over the last decades and the most popular ones are the so-called neutral point clamped, the flying capacitors, and cascaded H-bridge (CHB) inverters [3, 4].
This thesis is focuses on the CHB inverter topology that, compared to the others, presents the key advantage of an extreme modularity, since it is composed of several elementary cells and where the number of electronic components grows linearly when the number of levels increases, unlike the aforementioned alternatives, and it represents the best topology for large numbers of levels.

1.1.4 Model Predictive Control in Power Electronics

In the last decades, the MPC has proven to be an effective control strategy also in the field of power electronics.
Many versions of MPC have been developed to control power converters and motor drives by following different paradigms with distinct advantages and disadvantages [5, 6].
The MPC can be divided into two main families: the continuous control set MPC (CCS-MPC), also called modulated MPC (MMPC or indirect MPC), and the finite control set MPC (FCS-MPC or direct MPC), with different advantages and disadvantages [5].
In the CCS-MPC approach, the control problem is formulated as a continuous variable optimization problem. The continuous input variable is computed through a quadratic programming solver and, then, a modulation technique is applied in order to convert them into the discrete switching signals to drive the switches [7, 8]. The FCS-MPC, on the other hand, is formulated as a discrete optimization problem by considering the discrete nature of the switches, which can only be turned ON and OFF and the possible switching combinations constitute a finite and countable set of inputs [9]. The controller selects the switching pattern among those physically realizable by the inverter and the input is directly applied to the converter without need for a modulation algorithm. The first consequence of directly controlling the switching signals is a fast dynamic response and the possibility to limit the switching frequency, thus the losses, of the converter.

1.1.5 Model Predictive Control for Multilevel Power Electronic Converters

FCS-MPC has proven to be one of the most effective and promising versions of MPC for power electronic converters. However, it has the main drawback of being computationally demanding. This is especially emphasized when controlling multilevel converters, since the number of possible switching patterns grows exponentially with the number of levels. In addition, due to the fast dynamics of the electrical quantities, the control algorithms for power electronic converters typically require sampling intervals of tens of microseconds to accurately track the current reference
These two factors make it challenging to implement the FCS-MPC on a multilevel converter because the controller has to perform a large amount of computations in a relatively short time interval.
Nowadays, there are extremely powerful platforms used to compute the control law of these devices, megahertz clock frequency of parallel computing capabilities. However, even these platforms may not be able to meet the stringent timing requirements of the system for the high computational load of the existing FCS-MPC solving algorithms.

1.2 Thesis Motivation

In the literature, there are many works about reducing the computational burden of the FCS-MPC for CHB inverters. The most straightforward and popular strategy is to limit the search space of the control inputs, such to reduce the number of possible combinations to be explored by the optimization algorithm [10, 11, 12, 13]. However, this solution dramatically decrease the dynamic response of the controller, removing the original main advantage of the FCS-MPC.
Despite the effort in reducing the computations needed by the algorithm [14, 15, 16], the literature lacks for an efficient way to solve the FCS-MPC for CHB inverters to makes the algorithm implementable in tens of microseconds without negatively affect the performance of the controller.

1.3 Thesis Objectives

This thesis focuses on the development of algorithms to efficiently compute the FCS-MPC for CHB inverters.
Simulative and experimental validations were carried out on a CHB Static Synchronous Compensator (CHB-STATCOM).

1.3.1 Cascaded H-Bridge Static Synchronous Compensator as Test Bench

A STATCOM is an inverter shunt connected to the grid to regulate the node voltage, aiming to increase the stability of the power system. Since the STATCOM is designed to be connected to medium voltage grids (from 1 kV to 100 kV), the CHB is a typical solution for this application because it is a suitable topology for a large number of levels. Therefore, a CHB-STATCOM can reach the value of the grid voltage without the need

for a step-up transformer, reducing cost, bulk, harmonic distortion, and nonlinearity.
Like every CHB inverter aiming to track the output currents, the CHB-STATCOM requires a currents controller and, hence, it resulted a suitable application for testing the control algorithms developed during this research work.
Because of the inherent challenges of the this application, this thesis also proposes some methodologies specifically designed for the CHB-STATCOM control. However, it can be underlined that the developed currents controllers described in this thesis have a general formulation and they can be applied to every CHB inverter.

1.3.2 Overview of Presented Works

Two approaches were developed to reduce the computational burden of the FCS-MPC for CHB inverters.

- The first explored approach employs machine learning techniques to approximate the optimal control law, such that the computations performed by solving the optimization problems are embedded in the inference of the neural networks.
- Then, based on mathematical considerations on the specific problem of controlling a CHB inverter, an analytical method was derived to find a simple explicit solution to the optimal control problem.

During the implementation fo the control techniques on the CHB-STATCOM prototype, many implementation problems were faced and the two key problems were the following.

- The main problem was due to the use of current transformer sensors to sense the output currents, which led to a DC current in the output terminals. This thesis proposes an algorithm to eliminate the DC component, which is simple to implement and without the need for extra hardware. This problem was never faced in the literature for CHB-STATCOM driven by FCS-MPC.
- Other problems related to the sampling and communication of the measurements through the different peripherals composing the system are described. The effects of these problems on the output currents are discussed and solutions are presented.

1.4 Thesis Outline

This thesis is organized as follows.
Part I describes the background of this research work, presenting the circuit diagram, dynamical model and the FCS-MPC for CHB inverters (Chapter 2) and CHB-STATCOMs (Chapter 3).
Part II presents the two proposed methods to reduce the computational burden of the FCS-MPC by using neural networks (Chapter 4) and the developed analytical method (Chapter 5).
Part III is about the validation of the proposed techniques. The neural network method was tested in simulative environment (Chapter 6) and through hardware in the loop (Chapter 7). The analytical solution was tested on a 5-level CHB-STATCOM prototype (Chapter 8).
Part IV describes the main implementation problems faced during the experiments. The

DC current injection problem and the developed solution are explained in Chapter 9, and the measurements sampling and transmission problems are discussed in Chapter 10. The list of published papers is presented in Chapter 11.

References

[1] F. Borrelli, A. Bemporad, and M. Morari, *Predictive control for linear and hybrid systems*. Cambridge University Press, 2017.

[2] E. F. Camacho and C. B. Alba, *Model predictive control*. Springer science & business media, 2013.

[3] M. Malinowski, K. Gopakumar, J. Rodriguez, and M. A. Perez, "A survey on cascaded multilevel inverters," *IEEE Transactions on Industrial Electronics*, vol. 57, no. 7, pp. 2197–2206, 2010.

[4] H. Akagi, "Multilevel converters: Fundamental circuits and systems," *Proceedings of the IEEE*, vol. 105, no. 11, pp. 2048–2065, 2017.

[5] S. Vazquez, J. Rodriguez, M. Rivera, L. G. Franquelo, and M. Norambuena, "Model predictive control for power converters and drives: Advances and trends," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 2, pp. 935–947, 2017.

[6] 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.

[7] Q. Xiao, H. Jia, Y. Tang, Y. Jin, Y. Mu, R. Teodorescu, and F. Blaabjerg, "Dual-layer modulated model predictive control scheme for the cascaded h-bridge converter," *IEEE Transactions on Industrial Electronics*, pp. 1–12, 2022.

[8] X. Pan, L. Zhang, Y. Li, K. Li, and H. Huang, "Modulated model predictive control with branch and band scheme for unbalanced load compensation by mmcc-statcom," *IEEE Transactions on Power Electronics*, vol. 37, no. 8, pp. 8948–8962, 2022.

[9] T. Geyer and D. E. Quevedo, "Performance of multistep finite control set model predictive control for power electronics," *IEEE Transactions on power electronics*, vol. 30, no. 3, pp. 1633–1644, 2014.

[10] B. Gutierrez and S.-S. Kwak, "Modular multilevel converters (mmcs) controlled by model predictive control with reduced calculation burden," *IEEE Transactions on Power Electronics*, vol. 33, no. 11, pp. 9176–9187, 2018.

[11] C. D. Townsend, T. J. Summers, and R. E. Betz, "Multigoal heuristic model predictive control technique applied to a cascaded h-bridge statcom," *IEEE Transactions on Power Electronics*, vol. 27, no. 3, pp. 1191–1200, 2012.

[12] Z. Ni, A. H. Abuelnaga, Y. Pan, A. Elezab, O. Zayed, M. Narimani, and J. Rodriguez, "A new mpc formulation based on suboptimal voltage vectors for multilevel inverters," *IEEE Journal of Emerging and Selected Topics in Power Electronics*, vol. 10, no. 6, pp. 7261–7270, 2022.

[13] P. Cortés, A. Wilson, S. Kouro, J. Rodriguez, and H. Abu-Rub, "Model predictive control of multilevel cascaded h-bridge inverters," *IEEE Transactions on Industrial Electronics*, vol. 57, no. 8, pp. 2691–2699, 2010.

[14] Y. Zhang, X. Wu, X. Yuan, Y. Wang, and P. Dai, "Fast model predictive control for multilevel cascaded h-bridge statcom with polynomial computation time," *IEEE Transactions on Industrial Electronics*, vol. 63, no. 8, pp. 5231–5243, 2016.

[15] A. Dekka, B. Wu, V. Yaramasu, and N. R. Zargari, "Dual-stage model predictive control with improved harmonic performance for modular multilevel converter," *IEEE Transactions on Industrial Electronics*, vol. 63, no. 10, pp. 6010–6019, 2016.

[16] J. Huang, B. Yang, F. Guo, Z. Wang, X. Tong, A. Zhang, and J. Xiao, "Priority sorting approach for modular multilevel converter based on simplified model predictive control," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 6, pp. 4819–4830, 2018.

Part I

Cascaded-H Bridge Inverter and Static Synchronous Compensator

Chapter 2

Cascaded H-Bridge Inverter: Modeling and Control

Power electronic converters are key components in modern electrical systems, being the core technology for the manipulation of the electrical energy and its regulation for supplying thousands of devices, from everyday life to industrial equipments.
From renewable energy sources and grid-connected systems to electric vehicles, power electronic converters have become the key enabling technology for advanced energy conversion and control. The increasing demand for energy efficiency, reduced environmental impact, and enhanced reliability has driven significant advancements in power electronics, making it a dynamic and vital research domain.
A power electronic converter is a solid-state device aiming to convert electrical energy from one form to another of electrical energy. A DC-AC converter, also referred as inverter, can transform a constant voltage to an alternating voltage while a DC-DC converter takes an input DC voltage and generate an amplified or attenuated DC output voltage.
Power converters are used in all the applications that needs an intelligent control of the energy. Inverters, for instance, are used to control the energy flowing from a photovoltaic (PV) panel or a wind turbine into the grid or to control the speed and torque of al electric motor drive. There exist several converter topologies for each functionality and application.
The building block of the more complex inverter topology this thesis is focused on is the so-called H-bridge inverter.

2.1 H-Bridge Inverter

The basic circuit diagram of the H-bridge inverter consists of an input DC voltage that is connected to the load through four power switches, as drown in Fig. 2.1. By turning the switches ON and OFF, the resulting power circuit changes and the input voltage v_{DC} is connected or disconnected to the load. It results in three possible output voltages, i.e., v_{DC} when the switching transistors T_1 and T_3 are ON while T_2 and T_4 are OFF; $-v_{DC}$ when T_2 and T_4 are ON while T_1 and T_3 are OFF; 0 voltage when T_1 and T_4 are ON

while T_2 and T_3 are OFF or vice versa, as shown in Fig. 2.2. The control circuit must absolutely avoid the other switching combinations to prevent short-circuiting the input voltage.
The H-bridge is referred to as a three-level converter, i.e., capable of producing three different voltage levels at the output terminals.

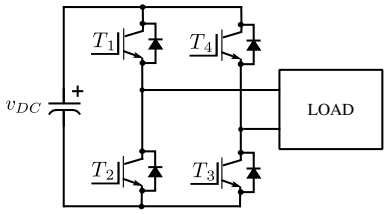


Figure 2.1: H-bridge schematics.

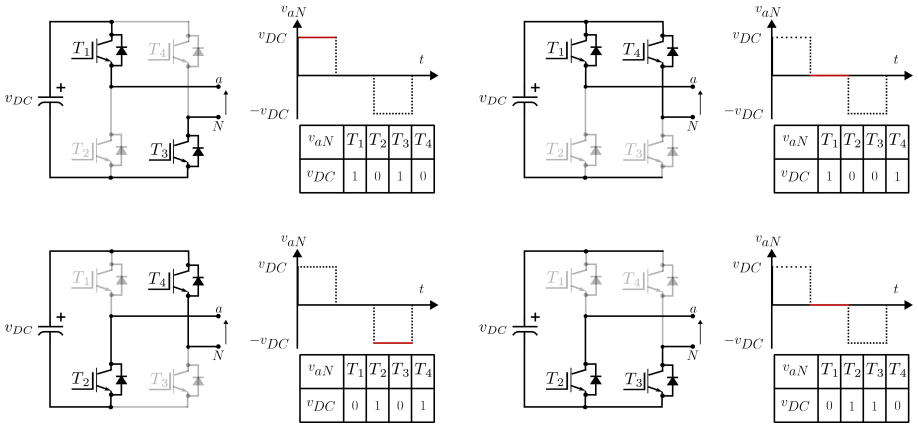


Figure 2.2: H-bridge three voltage levels.

2.2 Cascaded H-Bridge Inverter

A multilevel inverter is a power electronics device used for industrial applications consisting of a technological advancement of the standard inverter in high-power and medium-voltage applications. In order to produce different output voltage levels, it uses multiple voltage sources, each one with a fraction of the total input voltage, resulting in less voltage stress (often referred as dv/dt) on the load and the circuit elements, which leads to a reduction in the size of the converter components [3, 4, 6]. One of the most popular and promising multilevel converter topologies is the so-called cascaded H-bridge (CHB) inverter. It consists of multiple H-bridges connected in series to produce in output the sum of the outputs of the individual H-bridges. One of the main advantages of this topology over the others is its inherent modularity. In fact, each voltage level is generated by an H-bridge of the CHB without the need for additional hardware and the number of components (switches, drivers, sensing and control circuits, ...) grows linearly when the

number of levels increases. Because of this high scalability, the CHB inverter is a suitable topology for a large number of levels and it's an attractive solution for medium-voltage applications since it allows to have a large number of levels able to reach the desired voltage without the need for a step-up transformer, which increases the size, cost and nonlinearity of the system. Fig. 2.3 shows the basic schematics of a three-phase CHB inverter.

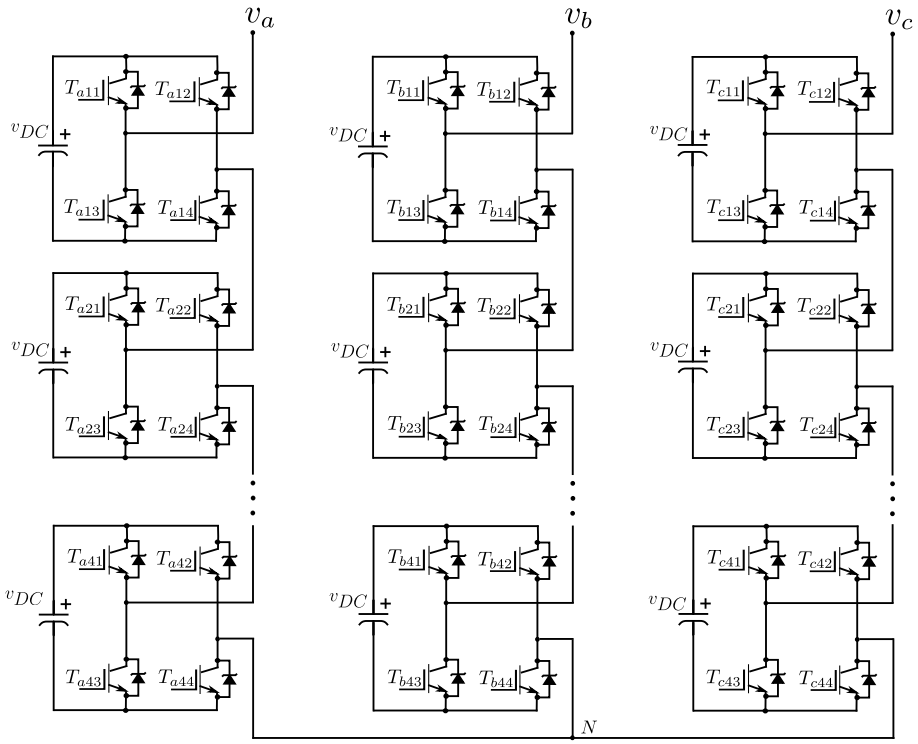


Figure 2.3: Cascaded H-bridge inverter schematics.

2.3 Dynamical Model of a CHB Inverter

Let us consider a three-phase CHB inverter composed of n H-bridges connected in series for each phase $p \in \{a, b, c\}$. Let us consider that each H-bridge is supplied by a DC-link voltage source with rated voltage V_{DC} and the converter is connected to a general RL load and a three-phase voltage source as in Fig. 2.4. Depending on the application, this schematic represents a grid-tied inverter connected to the grid through a filter inductor with its internal resistance, such as an inverter connected to a motor drive, where the voltage sources represent the electromotive forces [7]. The dynamical equations that govern the currents are given by the Kirchhoff's laws as follows:

$$L\frac{di_p}{dt} = v_{sp} - Ri_p - v_p \ \ , p \in \{a, b, c\} \tag{2.1}$$

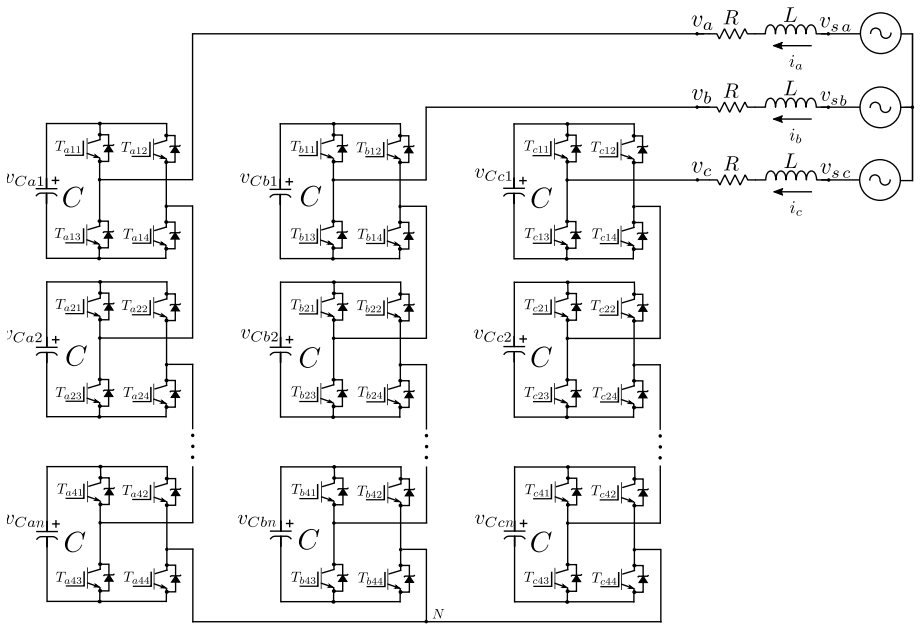


Figure 2.4: Cascaded H-bridge inverter connected to an RL load.

where i_p is the phase p current, v_p is the phase p voltage at the output terminals of the inverter, v_{sp} is the phase p voltage at the load side, R and L are the resistance and inductance of the load. The equations are transformed into the α, β reference frame through Clarke transformation (in the Appendix), obtaining:

$$\begin{cases} L\frac{di_\alpha}{dt} = & v_{s\alpha} - Ri_\alpha - v_\alpha \\ L\frac{di_\beta}{dt} = & v_{s\beta} - Ri_\beta - v_\beta. \end{cases} \tag{2.2}$$

The continuous time dynamical equations are discretized via forward Euler approximation and the obtained dynamical model is the following:

$$\begin{cases} i_\alpha(k+1) = & i_\alpha(k) + \frac{T_s}{L}\left(-Ri_\alpha(k) + v_{s\alpha}(k) - v_\alpha(k)\right) \\ i_\beta(k+1) = & i_\beta(k) + \frac{T_s}{L}\left(-Ri_\beta(k) + v_{s\beta}(k) - v_\beta(k)\right), \end{cases} \tag{2.3}$$

where T_s is the sampling interval and k is the sampling instant.
Since the currents dynamical model is linear, it is convenient to expressed them in matrix form as follows:

$$\mathbf{i}_{\alpha,\beta}(k+1) = \begin{bmatrix} 1-\frac{T_s}{L}R & 0 \\ 0 & 1-\frac{T_s}{L}R \end{bmatrix} \mathbf{i}_{\alpha,\beta}(k) + \begin{bmatrix} \frac{T_s}{L} & 0 \\ 0 & \frac{T_s}{L} \end{bmatrix} \left(\mathbf{v}_{s(\alpha,\beta)}(k) - \mathbf{v}_{\alpha,\beta}(k)\right), \tag{2.4}$$

where $\mathbf{i}_{\alpha,\beta} = [i_\alpha, i_\beta]^T$ is the current vector, $\mathbf{v}_{s(\alpha,\beta)} = [v_{s\alpha}, v_{s\beta}]^T$ is the voltage source vector and $\mathbf{v}_{\alpha,\beta} = [v_\alpha, v_\beta]^T$ is the output voltage vector at the inverter terminals. The inverter output voltage can be expressed as $\mathbf{v}_{\alpha,\beta}(k) = \mathbf{S}_{\alpha,\beta}(k) V_{DC}$, where $\mathbf{S}_{\alpha,\beta}(k) = [S_\alpha(k), S_\beta(k)]^T$ is the switching vector at time k, which is a discrete variable corresponding to the control input applied to the inverter. Eq. (2.4) can be formulated as follows:

$$\mathbf{i}_{\alpha,\beta}(k+1) = \mathbf{A}\ \mathbf{i}_{\alpha,\beta}(k) + \mathbf{B}\ \mathbf{S}_{\alpha,\beta}(k) + \mathbf{F}\ \mathbf{v}_{s(\alpha,\beta)}(k) \tag{2.5}$$

where:

$$\mathbf{A} = \begin{bmatrix} 1-\frac{T_s}{L}R & 0 \\ 0 & 1-\frac{T_s}{L}R \end{bmatrix}$$

$$\mathbf{B} = \begin{bmatrix} -\frac{T_s}{L} & 0 \\ 0 & -\frac{T_s}{L} \end{bmatrix} V_{DC}, \quad \mathbf{F} = \begin{bmatrix} \frac{T_s}{L} & 0 \\ 0 & \frac{T_s}{L} \end{bmatrix}.$$

2.4 Model Predictive Control for CHB Inverters

During the last decade, model predictive control (MPC) has become one of the most popular control techniques in many engineering fields in both research and industry and has been attracting increasing attention also in the field of power converters [1, 2].
The main idea of the MPC is to formulate the control problem objective as an optimization problem. A cost function is designed to minimize the error between the reference and the predicted state variables at the future instants, which are computed by employing the dynamical model of the system. A second term is usually added into the cost function to weight the control effort and to make a trade-off with the control performance. Constraints can be added to the minimization problem to handle physical limitations or to ensure a desired behavior of the controlled system.
The FCS-MPC has demonstrated to be one of the most effective and promising versions of MPC for power electronic converters, for its extremely fast dynamic response and the ability to limit the switching frequency, thus the losses, of the converter [5].
The FCS-MPC current control problem for a CHB inverter is usually expressed as the

following discrete optimization problem:

$$\begin{aligned} \mathbf{S}^*_{\alpha,\beta}(k) = \min_{\mathbf{S}_{\alpha,\beta}(k)} \quad & \left\| \mathbf{i}^{ref}_{\alpha,\beta}(k+1) - \mathbf{i}_{\alpha,\beta}(k+1) \right\|_{\mathbf{Q}} + \\ & + \left\| \mathbf{S}_{\alpha,\beta}(k) - \mathbf{S}_{\alpha,\beta}(k-1) \right\|_{\mathbf{P}} \\ s.t. \quad & \mathbf{i}_{\alpha,\beta}(k+1) = \mathbf{A}\ \mathbf{i}_{\alpha,\beta}(k) + \mathbf{B}\ \mathbf{S}_{\alpha,\beta}(k) + \mathbf{F}\ \mathbf{v}_{s(\alpha,\beta)}(k), \\ & \mathbf{S}_{\alpha,\beta}(k) \in \mathbf{V}_{\alpha,\beta}, \end{aligned} \tag{2.6}$$

where the first term is the weighted norm of the error between the $\mathbf{i}^{ref}_{\alpha,\beta}(k+1)$ and the predicted current $\mathbf{i}_{\alpha,\beta}(k+1)$, and the second term aims to reduce the variation of the switching vector, in order to limit the switching losses of the converter. The 2×2 weighting matrices $\mathbf{Q}$ and $\mathbf{P}$ are diagonal matrices and can be expressed as $\mathbf{Q} = q_i \times \mathbf{I}_2$ and $\mathbf{P} = p_i \times \mathbf{I}_2$, where q_i and p_i are scalar positive parameters. The optimal switching vector $\mathbf{S}_{\alpha,\beta}(k)$ must belong to $\mathbf{V}_{\alpha,\beta}$, the set of all the vectors that the inverter can physically generate.

Depending on the application, the control problem can be extended to predict more future steps (as will be shown in Chapter 4), to take into account for the computation delay (as will be presented in Chapter 5) or to minimize a different cost function.

The most straightforward and popular way to solve this discrete optimization problem is through an exhaustive search algorithm. It relies on computing the currents predictions for every possible switching vector in $\mathbf{V}_{\alpha,\beta}$ by employing the dynamical model in (2.5). Then, the cost function is computed for every prediction and the vector associated with the minimum cost is selected, as summarized in Algorithm 2.1. Despite its simplicity, this approach is not a scalable algorithm, since the number of combinations, which is $12n^2 + 6n + 1$, grows quadratically with respect to the number of levels of the converter. In this work, different solutions were developed to solve the currents control problem of the FCS-MPC with the aim of decreasing the computational burden of the control method by employing both machine learning and analytical strategies.

Algorithm 2.1 FCS-MPC currents controller through exhaustive search algorithm.

1: initialize $c^*_{cc} \leftarrow \infty$, $\mathbf{S}^*_{\alpha,\beta}(k) \leftarrow [0,0]^T$
2: **for all** $\mathbf{S}_{\alpha,\beta}(k) \in \mathbf{V}_{\alpha,\beta}$ **do**
3: $$\mathbf{i}_{\alpha,\beta}(k+1) = \mathbf{A}\ \mathbf{i}_{\alpha,\beta}(k) + \mathbf{B}\ \mathbf{S}_{\alpha,\beta}(k) + \mathbf{F}\ \mathbf{v}_{s(\alpha,\beta)}(k)$$
4: $$\begin{aligned} c_{cc} \leftarrow \quad & \left\| \mathbf{i}^{ref}_{\alpha,\beta}(k+1) - \mathbf{i}_{\alpha,\beta}(k+1) \right\|_{\mathbf{Q}} + \\ & + \left\| \mathbf{S}_{\alpha,\beta}(k) - \mathbf{S}_{\alpha,\beta}(k-1) \right\|_{\mathbf{P}} \end{aligned}$$
5: **if** $c_{cc} < c^*_{cc}$ **then**
6: $c^*_{cc} \leftarrow c_{cc}$
7: $\mathbf{S}^*_{\alpha,\beta}(k) \leftarrow \mathbf{S}_{\alpha,\beta}(k)$
8: **end if**
9: **end for**

References

[1] F. Borrelli, A. Bemporad, and M. Morari, *Predictive control for linear and hybrid systems*. Cambridge University Press, 2017.

[2] E. F. Camacho and C. B. Alba, *Model predictive control*. Springer science & business media, 2013.

[3] M. Malinowski, K. Gopakumar, J. Rodriguez, and M. A. Perez, "A survey on cascaded multilevel inverters," *IEEE Transactions on Industrial Electronics*, vol. 57, no. 7, pp. 2197–2206, 2010.

[4] H. Akagi, "Multilevel converters: Fundamental circuits and systems," *Proceedings of the IEEE*, vol. 105, no. 11, pp. 2048–2065, 2017.

[5] S. Vazquez, J. Rodriguez, M. Rivera, L. G. Franquelo, and M. Norambuena, "Model predictive control for power converters and drives: Advances and trends," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 2, pp. 935–947, 2017.

[6] A. M. Saif, C. Buccella, V. Patel, M. Tinari, and C. Cecati, "Design and cost analysis for statcom in low and medium voltage systems," in *IECON 2018 - 44th Annual Conference of the IEEE Industrial Electronics Society*, pp. 3938–3943, 2018.

[7] F. Wang, X. Mei, J. Rodriguez, and R. Kennel, "Model predictive control for electrical drive systems-an overview," *CES Transactions on Electrical Machines and Systems*, vol. 1, no. 3, pp. 219–230, 2017.

Chapter 3

Cascaded H-Bridge Static Synchronous Compensator: Modeling and Control

A CHB static synchronous compensator (CHB-STATCOM) was employed as a test bench for the developed methodologies.
The static synchronous compensator is one of the most advanced flexible alternating current transmission systems (FACTs), a family of technologies aiming to improve the stability of the grid voltage. In medium voltage grids the power line can be considered as a mainly inductive load. Due to this feature, it is possible to regulate the grid voltage by controlling the reactive power flowing through the line. The STATCOM working principle leverages on this property and is used to inject or absorb reactive power to or from the grid enabling the control of the line voltage at the point of common coupling (PCC) and enhancing the stability of the power system.
Unlike other traditional systems, the STATCOM is a solid-state device employing power electronics components, and is a technological advancement of VAR compensators based on passive components. It consists in a power inverter connected to the grid through a filter inductor, used to dynamically control the current flowing between the DC-link and the grid. The STATCOM equips a capacitor at the DC side without any DC voltage source. In fact, it only aims to exchange reactive power and does not provide active power support to the grid. Hence, besides controlling the reactive power, it also has to absorb a little amount of active power to balance the DC-link capacitor.
Since the STATCOMs are designed for medium voltage applications, the CHB multilevel converter topology is an effective alternative, which can lead to removing the step-up transformer and decreasing the size of the components of the system. However, to reach medium voltage values a large number of levels of the converter are needed. In a CHB-STATCOM, the number of components linearly depends on the number of levels, but the computational complexity of the FCS-MPC exponentially grows when the number of levels increases.
Because of the large number of levels used on CHB-STATCOM applications, it was chosen as a suitable application for testing the developed FCS-MPC algorithms.

3.1 Cascaded H-Bridge Static Synchronous Compensator

A multilevel cascaded H-bridge STATCOM is composed of a three-phase CHB inverter with n cells per phase and each H-Bridge is equipped with an input DC-link capacitor C. The converter is connected to the grid through a series inductor L with internal resistance R, as shown in Figure 3.1. Each H-bridge allows four admissible switching combinations

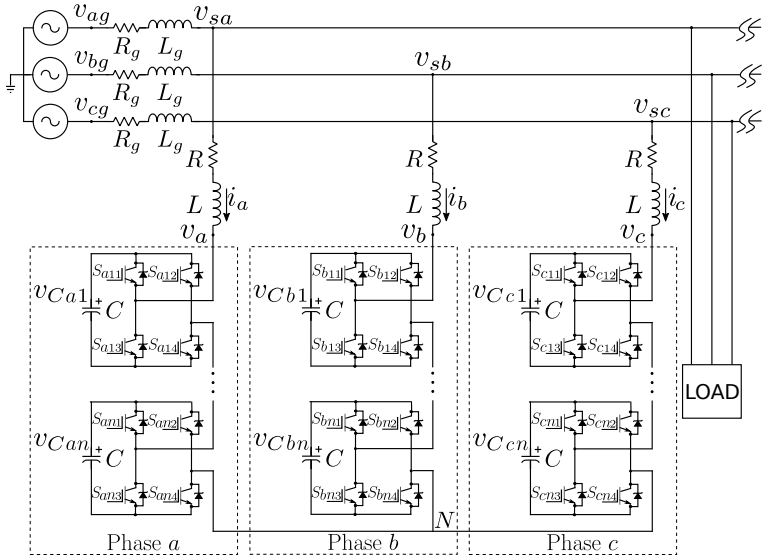


Figure 3.1: Cascaded H-bridge STATCOM schematics.

able to supply a positive, negative or zero output voltage, whose value is equal to that of the input DC-link voltage. According to this, the system can be modeled considering the inputs of the system as discrete variables $s_{pi} \in \{-1, 0, 1\}$, where $i = 1, ..., n$ and $p \in \{a, b, c\}$ indicates that the H-bridge cell i of p phases is outputting negative, zero or positive voltage. The total output voltage v_p of phase p of the STATCOM is the sum of the output voltages of every individual cell of phase p:

$$v_p = \sum_{i=1}^{n} s_{pi} \cdot v_{Cpi} \simeq \sum_{i=1}^{n} s_{pi} \cdot V_{DC} = S_p \cdot V_{DC} \tag{3.1}$$

where v_{Cpi} is the DC-link capacitor voltage of the i-th cell of phase p and S_p is the sum of the individual s_{pi} for $i = 1, ..., n$. The DC-link voltage is assumed to be close to its rated value V_{DC}, which is a common hypothesis since we always assume that the CHB-STATCOM is correctly sized [5, 10] to have a small voltage ripple on the DC side and that the controller is correctly implemented to keep the capacitor balanced [11]. It will be clear later on that, with this assumption, we remove the nonlinearity in the currents model and strongly simplify the controller design.
The dynamic equation of the currents and DC-link capacitor voltages are:

$$\begin{cases} L\frac{di_p}{dt} = v_{sp} - Ri_p - v_p \simeq v_{sp} - Ri_p - S_p \cdot V_{DC} \\ C\frac{dv_{Cpi}}{dt} = s_{pi} \cdot i_p, \quad i = 1, ..., n,\ p \in \{a, b, c\}\ , \end{cases} \tag{3.2}$$

where i_p is the current controlled by the inverter, v_{sp} is the voltage at the PCC and v_p is the output voltage of the inverter for phase p. With the assumption in (3.1), the current equations does not depend on the DC-link voltage of each individual cell, which removes the nonlinear (bilinear) relation in the currents equations, i.e., the product of the state variables i_p times the control inputs s_{pi}. Furthermore, it makes the currents equations independent of the capacitor voltages, which allows us to split the overall control problem into two individual parts: the current control problem and, then, the DC-link voltage balancing problem.
Since we want to design a discrete time controller, we discretize (3.2) and apply the Clarke transformation (in the Appendix), which leads to the following CHB-STATCOM dynamical model:

$$\begin{cases} \mathbf{i}_{\alpha,\beta}\,(k+1) = & \mathbf{A}\ \mathbf{i}_{\alpha,\beta}\,(k) + \mathbf{B}\ \mathbf{S}_{\alpha,\beta}\,(k) + \mathbf{F}\ \mathbf{v}_{s(\alpha,\beta)}\,(k) \\ v_{Cpi}(k+1) = & v_{Cpi}(k) + \frac{T_s}{C}\,(s_{pi}(k) \cdot i_p(k))\,, \quad i = 1, ..., n,\ p \in \{a, b, c\}\ , \end{cases} \tag{3.3}$$

where vectors and matrices are defined as in (3.2).

3.2 Model Predictive Control for CHB-STATCOMs

The overall control problem of FCS-MPC requires to solve an optimal problem that minimizes the deviation of the predicted currents $\mathbf{i}_{\alpha,\beta}(k+1)$ from the reference $\mathbf{i}^{ref}_{\alpha,\beta}\,(k+1)$ and the deviation of the predicted capacitors voltages from the rated value V_{DC}. We introduce the $3n \times 1$ vector of the predicted capacitor voltages $\mathbf{v}_C(k+1) = [\mathbf{v}_{Ca}(k+1), \mathbf{v}_{Cb}(k+1), \mathbf{v}_{Cc}(k+1)]^T$, where $\mathbf{v}_{Cp}(k+1) = [v_{Cp1}\,(k+1)\,, v_{Cp2}\,(k+1)\,, \ldots, v_{Cpn}\,(k+1)]^T$, with $p \in \{a, b, c\}$, and the $3n \times 1$ vector of the rated voltage $\mathbf{V}^{DC}_{3n} = V_{DC} \cdot \mathbf{1}_{3n}$, which is the $3 \cdot n \times 1$ vector containing the value V_{DC} in each element ($\mathbf{1}_{3n}$ is the $3n \times 1$ unitary vector). At the same time, the optimization problem penalizes the switching activity and it computes the $3n \times 1$ switching vector $\mathbf{s}(k) = [\mathbf{s}_a(k), \mathbf{s}_b(k), \mathbf{s}_c(k)]^T$, $\mathbf{s}_p(k) = [s_{p1}\,(k)\,, s_{p2}\,(k)\,, \ldots, s_{pn}\,(k)]^T$, $p \in \{a, b, c\}$.
The optimal problem is formulated as:

$$\begin{aligned} \mathbf{s}^*(k) = \min_{\mathbf{s}(k)} & \left\| \mathbf{i}^{ref}_{\alpha,\beta}\,(k+1) - \mathbf{i}_{\alpha,\beta}\,(k+1) \right\|_{\mathbf{Q}} + \\ & + \left\| \mathbf{V}^{DC}_{3n} - \mathbf{v}_C\,(k+1) \right\|_{\mathbf{Q}_v} + \\ & + \left\| \mathbf{s}\,(k) - \mathbf{s}\,(k-1) \right\|_{\mathbf{P}_s} \\ s.\,t. \quad & \mathbf{i}_{\alpha,\beta}\,(k+1) = \mathbf{A}\ \mathbf{i}_{\alpha,\beta}\,(k) + \mathbf{B}\ \mathbf{S}_{\alpha,\beta}\,(k) + \mathbf{F}\ \mathbf{v}_{s(\alpha,\beta)}\,(k) \\ & v_{Cpi}(k+1) = v_{Cpi}(k) + \frac{T_s}{C}\,(s_{pi}(k) \cdot i_p(k))\,, \quad i = 1, ..., n,\ p \in \{a, b, c\} \\ & s_{pi}\,(k) \in \{-1, 0, 1\},\ i = 1, ..., n,\ p \in \{a, b, c\} \\ & s_{pl} \times s_{pm} \geq 0,\ \forall\, l, m = 1, ..., n : l \neq m, \end{aligned} \tag{3.4}$$

where $\mathbf{Q}_v$ and $\mathbf{P}_s$ are $3n \times 3n$ tuning matrices. The last constraint ensures that the switching states for each phase p have the same polarity, i.e., they must be all non-negative or non-positive. It prevents two H-bridges in the same phase to output a positive and a negative voltage at the same time, which would imply an overall null voltage with unnecessary switching activity.
In this formulation, the discrete optimal switching vector has to be found in a set of $\left(2^{n+1}-1\right)^3$possible inputs. This problem can be solved through a greedy approach by enumerating all the possible solutions only for a small number of levels, due to the combinatorial explosion of the number of iterations.
For this reason, and since we assumed the currents independent of the capacitor voltages, the overall optimization problem is usually split into two, following the so-called partially stratified approach [1, 4, 2, 3, 6, 12]. The first control problem, which is the higher priority optimization layer, aims to only track the currents reference and it computes the optimal switching vector $\mathbf{S}^*_{\alpha,\beta}$ in the α, β frame. The second one aims to find the switching variables s_{pi} that minimize the voltage imbalance between capacitors in the three phases.
The current control problem is the one in Section 2.4, which has to be solved for every CHB inverter driven by FCS-MPC and which is the main focus of this research work. For a CHB-STATCOM, we also have to keep the DC-link capacitors balanced and, hence, to solve the a second optimization problem. However, as it will be clear later on, the currents control problem is the most computationally expensive part of the overall algorithm and the main responsible for the practicability of the FCS-MPC implementation.

3.2.1 Individual Voltage Balance

Once the currents control is solved and the switching vector $\mathbf{S}^*_{\alpha,\beta}$ is computed, it is necessary to choose which H-bridges has to supply voltage among the available ones. The individual voltage balancing problem aims to select the switching variables s_{pi}, $i = 1, \ldots, n$, for the individual H-bridges that keep the voltages of the DC-links balanced and such that to output the overall S^*_p, $p \in \{a, b, c\}$, computed by the higher priority layer.
The optimization problem for the individual balancing of phase $p \in \{a, b, c\}$ is the following:

$$\begin{aligned}
\mathbf{s}^*_p(k) = \min_{\mathbf{s}_p(k)} & \left\| \mathbf{V}^{DC} - \mathbf{v}_{Cp}\left(k+1\right) \right\|_{\mathbf{Q}_{ib}} + \\
& + \left\| \mathbf{s}_p\left(k\right) - \mathbf{s}_p\left(k-1\right) \right\|_{\mathbf{P}_{ib}} \\
s.\,t. \quad & v_{Cpi}(k+1) = v_{Cpi}(k) + \frac{T_s}{C}\left(s_{pi}(k) \cdot i_p(k)\right), \quad i = 1, ..., n \\
& \sum_{i=1}^{n} s_{pi}\left(k\right) = S^*_p\left(k\right), \\
& s_{pl}(k) \times s_{pm}(k) \geq 0, \forall\, l, m = 1, ..., n : l \neq m,
\end{aligned} \tag{3.5}$$

where $\mathbf{V}^{DC} = V_{DC} \cdot \mathbf{1}_n$ is the $n \times 1$ vector with the rated voltage V_{DC} in each element ($\mathbf{1}_n$ is the $n \times 1$ unitary vector), $\mathbf{v}_{Cp}\left(k\right) = \left[v_{Cp1}(k), v_{Cp2}(k), \ldots, v_{Cpn}(k)\right]^T$ is the $n \times 1$ vector containing the capacitor voltages and $\mathbf{s}_p(k) = \left[s_{p1}(k), s_{p2}(k), \ldots, s_{pn}(k)\right]^T$ is the $n \times 1$ vector containing the switching variables at time k of $p \in \{a, b, c\}$ phase. The matrices $\mathbf{Q}_{ib} = q_{ib} \times \mathbf{I}_n$ and $\mathbf{P}_{ib} = p_{ib} \times \mathbf{I}_n$ perform the trade-off between the reference

tracking (which determines the deviation of the capacitor voltages from the rated value) and the switching frequency. A constraint is added to the optimization problem to ensure that the sum of the switching states $s_{pi}(k)$ is equal to the optimal switching vector $S_p^*(k)$ computed by the higher priority optimization layer. This way, the optimal input computed by the currents controller is not affected by the individual voltage balance.
Since each switching variable $s_{pi}(k)$ can have the three different values in $\{-1, 0, 1\}$, the number of possible switching combination for each phase is 3^n, which would make this problem not scalable for a large number of levels because of its combinatorial explosion. The authors in [1] effectively solved this problem by taking advantage of the constraints in (3.5) to reduce the number of computations. In fact, the two constraints impose that all

Algorithm 3.1 Individual voltages balancing algorithm for phase p.

1: **if** $S_p^*(k) \geq 0$ **then**
2: $\quad \overline{s} \leftarrow 1$ ▷ $s_{pi}(k)$ must belong to $\{0, 1\}$
3: **else**
4: $\quad \overline{s} \leftarrow -1$ ▷ $s_{pi}(k)$ must belong to $\{-1, 0\}$
5: **end if**
6: **for** $i \leftarrow 1$ to n **do**
7:

$$v_{Cpi}^{\overline{s}}(k+1) \leftarrow v_{Cpi}(k) + \frac{T_s}{C}(\overline{s} \cdot i_p(k))$$
$$v_{Cpi}^{0}(k+1) \leftarrow v_{Cpi}(k) + \frac{T_s}{C}(0 \cdot i_p(k)) = v_{Cpi}(k)$$

8:

$$\mathbf{C}_{ib,p}(i) \leftarrow \left\| V_{DC} - v_{Cpi}^{\overline{s}}(k+1) \right\|_{q_{ib}} + \left\| \overline{s} - s_{pi}(k-1) \right\|_{p_{ib}} - \left[\left\| V_{DC} - v_{Cpi}^{0}(k+1) \right\|_{q_{ib}} + \left\| 0 - s_{pi}(k-1) \right\|_{p_{ib}} \right]$$

9: **end for**
10: $\mathbf{I}_p \leftarrow$sort($\mathbf{C}_{ib,p}$) ▷ returns $\mathbf{C}_{ib,p}$ indexes sorted in ascending order
11: count$_p \leftarrow 1$
12: **for** $i \leftarrow 1$ to n **do** ▷ asserts the s_{pi} with minimum cost
13: $\quad$ **if** $\mathbf{I}_{\mathrm{p}}(i) \leq |S_p^*(k)|$ **then**
14: $\quad\quad s_{pi} \leftarrow \overline{s}$
15: $\quad$ **else**
16: $\quad\quad s_{pi} \leftarrow 0$
17: $\quad$ **end if**
18: $\quad$ count$_p \leftarrow$ count$_p + 1$
19: **end for**

the element of $\mathbf{s}_p^*(k)$ must be non-negative or non-positive, while satisfying the condition that the sum of the element of $\mathbf{s}_p^*(k)$ must be equal to $S_p^*(k)$. Hence, if $S_p^*(k)$ is non-

negative, only the combinations of non-negative $s_{pi}(k)$ have to be evaluated and, vice versa, if $S_p^*(k)$ is non-positive, only the combinations of non-positive $s_{pi}(k)$ have to be evaluated. Thus, once we have identified the sign of $S_p^*(k)$, we can reduce the problem 3.5 into a $\{0,1\}$ programming problem, which comprises 2^n possible combinations. Moreover, constraint $\sum_{i=1}^{n} s_{pi}(k) = S_p^*(k)$ impose us to select only a certain number of H-bridges to put in conduction among the n available. Specifically, a number equal to $\left\|S_p^*(k)\right\|$ switching variables $s_{pi}(k)$ has to be asserted, i.e., they must be set equal to $\overline{s}$, with $\overline{s} = 1$ if $S_p^*(k) > 0$ or $\overline{s} = -1$ if $S_p^*(k) < 0$, while the others should be set equal to 0.
From the CHB-STATCOM dynamical model in (3.3), we can notice that the switching variable $s_{pi}(k)$ only affects the DC-link voltage $v_{Cpi}(k+1)$. Hence, the overall cost can be considered as the sum of n independent costs given by $\left\|V^{DC} - v_{Cpi}(k+1)\right\|_{q_{ib}} + \left\|s_{pi}(k) - s_{pi}(k-1)\right\|_{p_{ib}}$, with $i = 1, ..., n$. Since $s_{pi}(k)$ can only be equal to the two values in $\{\overline{s}, 0\}$, we can compute the individual increment in the cost function of asserting the individual $s_{pi}(k)$ ($s_{pi}(k) \leftarrow \overline{s}$) with respect to not asserting it ($s_{pi}(k) \leftarrow 0$).
The obtained array of costs increments is sorted in ascending order and the first $\left\|S_p^*(k)\right\|$ elements of the sorted array determine the switching variables to be set to $\overline{s}$, keeping the others equal to 0. This way, we reach the solution by computing n individual costs increments and by sorting the n element costs array, which makes the computational complexity of the individual voltage balance equal to the complexity of the used sorting algorithm, i.e., $\mathcal{O}(n^2)$ for a simple selection sort.

3.2.2 Clusters Voltages Balance

Usually, in order to obtain the optimal switching vector in a, b, c coordinates $\mathbf{S}_{a,b,c}^* = [S_a^*, S_b^*, S_c^*]^T$ for computing the individual voltage balance in (3.5), the optimal vector $\mathbf{S}_{\alpha,\beta}^*$ is transformed into the a, b, c reference frame by setting the homopolar component to zero and applying the inverse Clarke transformation (in the Appendix).
Often, in the literature, we can find works that introduce a common-mode voltage in the STATCOM output aiming to reduce the imbalance between the DC-link voltages among the three phases of the STATCOM, which is referred to as cluster voltage balancing [11]. This thesis introduces the concept of the cluster voltage balance also in the FCS-MPC framework, which was already known in the literature for standard PI regulators [7, 8, 9]. Once the optimal input $\mathbf{S}_{\alpha,\beta}^*$ is computed by the current controller, the $\mathbf{S}_{a,b,c}^*$ is selected among the redundant vectors in order to balance the cluster voltages. From (3.3), the average capacitor voltages v_{Cp} across the three phases are:

$$v_{Cp}(k+1) = v_{Cp}(k) + \frac{T_s}{C_{eq}} S_p(k) \cdot i_p(k),\ p \in \{a, b, c\}\,, \tag{3.6}$$

where $C_{eq} = C/n$ is the equivalent capacitance of one cluster.
The cluster voltage balancing problem is added between the two aforementioned optimization layers and it is formulated as follows:

$$\mathbf{S}^*_{a,b,c}(k) = \min_{\mathbf{S}_{a,b,c}(k)} \quad \left\| \mathbf{V}^{DC}_{a,b,c} - \mathbf{v}_{C(a,b,c)}(k+1) \right\|_{\mathbf{Q}_{cb}} \tag{3.7}$$

$$+ \left\| \mathbf{S}_{a,b,c}(k) - \mathbf{S}_{a,b,c}(k-1) \right\|_{\mathbf{P}_{cb}} \tag{3.8}$$

$$+ \left\| S_a(k) + S_b(k) + S_c(k) \right\|_{w_{cb}} \tag{3.9}$$

$$s.t. \quad v_{Cp}(k+1) = v_{Cp}(k) + \frac{T_s}{C_{eq}} S_p(k) \cdot i_p(k), p \in \{a,b,c\} \tag{3.10}$$

$$\mathbf{S}^*_{\alpha,\beta}(k) = \mathbf{T}_{2\times 3} \cdot \mathbf{S}_{a,b,c}(k) \tag{3.11}$$

$$\mathbf{S}_{a,b,c}(k) \in \mathbf{V}_{a,b,c}, \tag{3.12}$$

where $\mathbf{V}^{DC}_{a,b,c} = [V_{DC}, V_{DC}, V_{DC}]^T$ is the 3×1 vector containing the rated voltage, $\mathbf{v}_{C(a,b,c)} = [v_{Ca}(k+1), v_{Cb}(k+1), v_{Cc}(k+1)]^T$ is the 3×1 vectors of the predicted cluster voltages, $\mathbf{S}_{a,b,c}(k)$ is the switching vector in a,b,c coordinates and $\mathbf{Q}_{cb} = q_{cb} \times \mathbf{I}_3$ and $\mathbf{P}_{cb} = p_{cb} \times \mathbf{I}_3$ are 3×3 tuning matrices. The first term is a weighted norm of the deviation of the three cluster voltages from the reference value and the second term is added to reduce the switching frequency. The optional third term is added in this general formulation to limit the usage of the common-mode voltage, which is weighted by the tuning coefficient w_{cb}. The $\mathbf{T}_{2\times 3}$ matrix is the 2×3 Clarke transformation matrix and the constraint $\mathbf{S}^*_{\alpha,\beta}(k) = \mathbf{T}_{2\times 3} \cdot \mathbf{S}_{a,b,c}(k)$ ensures that the control computed by the currents controller is not affected by this optimization layer, which means, in other words, that we are only injecting a common-mode voltage without modifying the α, β currents controlled by the first optimization layer. Since the number of redundant vectors increases linearly with the number of levels, the algorithm can be simply implemented through an exhaustive search among the $2n+1$ redundant vectors in the a,b,c coordinates, which does not substantially affect the computational burden of the overall control.
Algorithm 3.2 summarizes the exhaustive search implementation for solving the cluster balancing problem.
It starts by computing the switching vector $\mathbf{S}^0_{a,b,c}(k)$ by transforming $\left[S^*_\alpha(k), S^*_\beta(k), 0\right]^T$ and rounding the result. The matrix $\mathbf{T}^{-1}_{2\times 3}$ is introduced to this aim, which is the pseudo-inverse of matrix $\mathbf{T}_{2\times 3}$ and performs the inverse Clarke transformation.
Then, the set $\mathbf{V}_{a,b,c}$ of feasible vectors in a,b,c coordinates is defined as the collection of redundant vectors computed as $\mathbf{S}^0_{a,b,c}(k) + [i,i,i]^T$ by varying i among all the integer values between $-n$ and n, i.e., $i \in \{-n, ..., n\}$.
For each vector belonging to $\mathbf{V}_{a,b,c}$, the costs are computed as in (3.7) and the array of costs $\mathbf{C}_{cb}$ is used to select the vector associated with the minimum cost.
The condition $\left\|\mathbf{S}^i_{a,b,c}(k)\right\|_\infty \leq n$ means that the norm of the elements of $\mathbf{S}^i_{a,b,c}(k) = \left[S^i_a, S^i_b, S^i_c\right]^T$ must have a maximum value equal to n ($-n \leq S^i_p \leq n \; \forall \; p \in \{a,b,c\}$), to ensure the feasibility of the computed switching vector. The clusters voltages balance algorithm has a low computational load compared to the other two sub-problems since it linearly depends on the number of levels, i.e., it is $\mathcal{O}(n)$.

Algorithm 3.2 Cluster voltages balancing control algorithm.

$\mathbf{S}^0_{a,b,c}(k) \leftarrow \mathbf{T}^{-1}_{2\times 3}\left[S(k), S^*_\beta(k), 0\right]^T$
initialize $c^*_{cb} \leftarrow \infty$, $\mathbf{S}^*_{a,b,c}(k) \leftarrow \mathbf{S}^0_{a,b,c}(k)$
for $i \leftarrow -n$ to n **do**
    $\mathbf{S}^i_{a,b,c}(k) \leftarrow \mathbf{S}^0_{a,b,c}(k) + [i,i,i]^T$

$$v^i_{Cp}(k+1) \leftarrow v_{Cp}(k) + \frac{T_s}{C_{eq}} S^i_p(k) \cdot i_p(k), p \in \{a,b,c\}$$
$$\mathbf{v}^i_{C(a,b,c)} \leftarrow \left[v^i_{Ca}(k+1), v^i_{Cb}(k+1), v^i_{Cc}(k+1)\right]^T$$

$$\begin{aligned}\mathbf{C}_{cb}(i) \leftarrow & \left\|\mathbf{V}^{DC}_{a,b,c} - \mathbf{v}^i_{C(a,b,c)}(k+1)\right\|_{\mathbf{Q}_{cb}} + \\ & + \left\|\mathbf{S}^i_{a,b,c}(k) - \mathbf{S}_{a,b,c}(k-1)\right\|_{\mathbf{P}_{cb}} + \\ & + \left\|S^i_a(k) + S^i_b(k) + S^i_c(k)\right\|_{w_{cb}}\end{aligned}$$

    if $\mathbf{C}_{cb}(i) < c^*_{cb}$ **then**
        if $\left\|\mathbf{S}^i_{a,b,c}(k)\right\|_\infty \leq n$ **then**
            $c^*_{cb} \leftarrow \mathbf{C}_{cb}(i)$
            $\mathbf{S}^*_{a,b,c}(k) \leftarrow \mathbf{S}^i_{a,b,c}(k)$
        end if
    end if
end for

3.2.3 Overall CHB-STATCOM Control

The aim of the STATCOM is to balance the voltage at the PCC through reactive power injection. Therefore, the controller has to compute the amount of current in quadrature with the grid voltage to exchange in order to stabilize the PCC at the desired reference voltage. After sampling the three-phase grid voltage $\mathbf{v}_{s(a,b,c)}$, the controller computes the phase angle through a phase-locked loop (PLL) algorithm. Then, it computes the direct voltage v_{sd} and compare it with the reference value v^{ref}_{sd} in order to feed a PI regulator that calculates the needed quadrature current i^{ref}_q to reach the desired voltage at PCC. Since the STATCOM is not supplied by any voltage source, the DC-link capacitors have to be continuously charged in order to compensate for the power losses of the circuit. Hence, a small amount of active power has to be absorbed from the grid. Therefore, the needed direct current i^{ref}_d is computed through a proportional-integral (PI) regulator to stabilize the average voltage of the DC-link capacitors $\overline{v}_C$ to the rated value V_{DC}.
The direct and quadrature currents are projected into the stationary frame α, β through the inverse Park transformation (in the Appendix) to obtain the currents reference $\mathbf{i}^{ref}_{\alpha,\beta}$. Then, the currents controller computes the optimal switching vector in α, β coordinates $\mathbf{S}^*_{\alpha,\beta}$ to track the reference currents by solving (2.6). The $\mathbf{S}^*_{\alpha,\beta}$ vector is transformed into the a, b, c coordinates through the simple Clarke transformation or by solving the cluster

voltage balancing in (3.7) to obtain $\mathbf{S}^*_{a,b,c}$. Finally, the a, b, c vector is used for computing the switching vectors s^*_{pi} for $i = 1, ..., n$ and $p \in \{a, b, c\}$ by solving the individual voltage balancing problem in (3.5) and the switching signals are sent to the drivers of the H-bridges. Fig. 3.2 summarizes the overall control scheme for a CHB-STATCOM controlled with FCS-MPC.

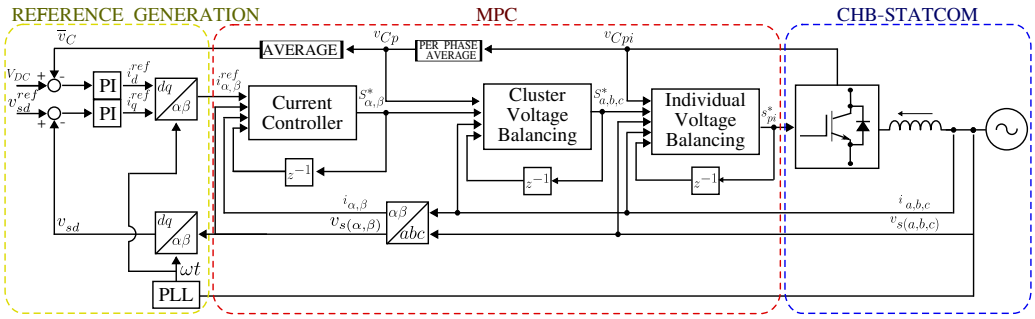


Figure 3.2: Overall CHB-STATCOM FCS-MPC control scheme.

3.2.4 Computational Load Analysis of the FCS-MPC for CHB-STATCOM

The standard currents control, described in Section 2.4, requires computing $12n^2 + 6n + 1$ costs.

The clusters voltages balance in Subsection 3.2.2 (Algorithm 3.2.2) requires computing $2n + 1$ costs in order to find the minimum one.

The individual voltages balance in Subsection 3.2.1 (Algorithm 3.1) requires computing n costs and sorting an array of n element, which is completed in $n^2/2 - n/2$ iterations for a simple selection sort algorithm.

If we just consider the heaviest parts of the individual voltages balance and of the currents control, we can notice that $12n^2$ is 24 times larger that $n^2/2$, while the clusters balance requires a negligible number of computations, i.e., $2n + 1$. Roughly speaking, it means that the overall computational burden is mainly referred to the currents controller, which covers the 96% of the overall computations $(12/(12 + 1/2) = 0.96)$. As it will be clear lather on, this is true also in practice.

Moreover, in a CHB inverter supplied by DC voltage supplies, the clusters and individual voltages balance problems can be theoretically avoided, since the DC-link are assumed to be balanced. The currents control problem, instead, is always needed for every CHB inverter controlling the output currents, irrespective of the specific application, and it is not limited to CHB-STATCOMs.

For these reasons, after these preliminary considerations, the focus of this research was directed to the currents control, which is the main responsible of the computational burden of the FCS-MPC for CHB-STATCOMs and, in general, for CHB inverters.

References

[1] Y. Zhang, X. Wu, X. Yuan, Y. Wang, and P. Dai, "Fast model predictive control for multilevel cascaded h-bridge statcom with polynomial computation time," *IEEE Transactions on Industrial Electronics*, vol. 63, no. 8, pp. 5231–5243, 2016.

[2] A. Dekka, B. Wu, V. Yaramasu, and N. R. Zargari, "Dual-stage model predictive control with improved harmonic performance for modular multilevel converter," *IEEE Transactions on Industrial Electronics*, vol. 63, no. 10, pp. 6010–6019, 2016.

[3] J. Huang, B. Yang, F. Guo, Z. Wang, X. Tong, A. Zhang, and J. Xiao, "Priority sorting approach for modular multilevel converter based on simplified model predictive control," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 6, pp. 4819–4830, 2018.

[4] Y. Zhang, X. Wu, and X. Yuan, "A simplified branch and bound approach for model predictive control of multilevel cascaded h-bridge statcom," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 10, pp. 7634–7644, 2017.

[5] A. M. Saif, C. Buccella, V. Patel, M. Tinari, and C. Cecati, "Design and cost analysis for statcom in low and medium voltage systems," in *IECON 2018 - 44th Annual Conference of the IEEE Industrial Electronics Society*, pp. 3938–3943, 2018.

[6] F. Simonetti, A. D'Innocenzo, and C. Cecati, "Neural network model predictive control for chb converters with fpga implementation," *IEEE Transactions on Industrial Informatics*, pp. 1–12, 2023.

[7] R. Xu, Y. Yu, R. Yang, G. Wang, D. Xu, B. Li, and S. Sui, "A novel control method for transformerless h-bridge cascaded statcom with star configuration," *IEEE Transactions on Power Electronics*, vol. 30, no. 3, pp. 1189–1202, 2015.

[8] D. Lu, J. Wang, J. Yao, S. Wang, J. Zhu, H. Hu, and L. Zhang, "Clustered voltage balancing mechanism and its control strategy for star-connected cascaded h-bridge statcom," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 10, pp. 7623–7633, 2017.

[9] Y. Shi, B. Liu, Y. Shi, and S. Duan, "Individual phase current control based on optimal zero-sequence current separation for a star-connected cascade statcom under unbalanced conditions," *IEEE Transactions on Power Electronics*, vol. 31, no. 3, pp. 2099–2110, 2016.

[10] P. Hu, J. M. Guerrero, and Z. He, "Design and analysis of a transformerless statcom based on hybrid cascaded multilevel converter," *International Journal of Electrical Power & Energy Systems*, vol. 104, pp. 694–704, 2019.

[11] H. Akagi, S. Inoue, and T. Yoshii, "Control and performance of a transformerless cascade pwm statcom with star configuration," *IEEE Transactions on Industry Application*, vol. 43, no. 4, pp. 1041–1049, 2007.

[12] F. Simonetti, A. D'Innocenzo, and C. Cecati, "Simple explicit solution of finite control set model predictive control," *IEEE Transactions on Industrial Electronics*, 2023, unpublished.

Part II

Proposed Techniques to Reduce the Computational Cost of FCS-MPC

Chapter 4

Machine Learning for FCS-MPC

The computational burden of the currents control quadratically depends on the number of levels of the converter, which can result in long execution times for a large number of levels and can force us to increase the length of the sampling interval, leading to a deterioration of the system performance or the impossibility to implement the controller. In recent years, machine learning (ML) techniques have been widely used in many engineering fields and in increasingly different application. Specifically, neural networks (NNs) are well known ML techniques that can be efficiently implemented in real-time applications thanks to their massive parallelism capability.
This chapter presents the neural network based approach developed during this research, aiming to reduce the computational time complexity of conventional FCS-MPC for CHB inverters. The computational cost analysis of the standard algorithm and the proposed method is shown, underlying the significant improvement of the proposed technique [22].

4.1 Introduction

FCS-MPC is commonly computed by searching for the optimal switching pattern among all the possible combinations via an exhaustive search algorithm [10]. However, in multilevel converters, computational complexity rapidly grows with the number of levels of the converter, making the calculation of the optimal input challenging within the sampling interval [12].
Significant efforts can be found in the literature to improve the efficiency of the FCS-MPC algorithm in order to make it implementable in a few tenth microseconds.
In [5, 1], the authors investigate methods that limit the space search of the optimal switching vector, reducing the computations but slowing down the controller dynamic response. In [3, 15], authors proposed control strategies that explicitly exploit the dynamical model of the system. However, they do not compute the control by solving an optimization problem. In [6], the optimization problem is solved by employing a sphere decoding algorithm. However, it requires the determination of an initial radius and involves several iterations.
Due to the large number of signals necessary to drive their power switches, multilevel converters usually employ field programmable gate arrays (FPGAs) because they make available a large amount of I/O pins necessary for driving the signals of the CHB con-

verter. Moreover, they are used for acquiring inputs, for the modulation algorithm and for achieving the highest computational speed [4, 9, 8]. In [4] the authors implemented the polynomial level FCS-MPC algorithm in [2] by employing a digital signal processor (DSP) for the main calculations of the control law, supported by an FPGA for reading the analog input signals, sending the gate signals to the switches and for the hardware acceleration of the most computationally expensive parts of the algorithm. Due to the large number of computations needed by the FCS-MPC approach, an effective solution is to implement the overall control law on the FPGA [7, 13, 14, 11].
In recent years, machine learning techniques have been widely used in many engineering fields. One popular application in the automatic controls field is to use ML to approximate a control law that is too computationally expensive to be implemented in real-time. With offline computations, the ML procedure computes a nonlinear function that embeds the optimization solver. This function can be used in online implementation in the place of the original controller. In [16], a shallow NN is trained to learn the MPC law for a two-level inverter for different circuit parameters and loads. In [19], the same strategy is applied to reduce the computational cost of a two-level inverter when considering horizons equal to one, two and three. In [21], horizons one and two are evaluated on a three-level neutral point clamped topology. In [17], a NN is used for a flying capacitor multilevel topology. A comparison of timings and space complexity of the algorithm is carried out, and the number of output neurons linearly depends on the number of levels. In [18], a comparison between classification and regression NNs for modular multilevel converters is presented, where regression turns out to be more effective than classification. Timings are compared for predictive horizons of lengths one and two. In [20], different ML techniques are compared for approximating MPC on a CHB inverter.
This chapter proposes a methodology to speed-up the FCS-MPC by using neural networks. It presents the study of the computational cost of the neural network MPC (NN-MPC) and discusses the comparison with the FCS-MPC.

4.2 Neural Networks for FCS-MPC of CHB Inverters

The idea of this novel approach is to train a NN to learn the optimal control on the basis of the solution to the optimal problem in Section 2.4, computed offline. This Chapter assumes that the NN can effectively approximate the control law and it only focuses on the computational load aspect. The next Chapters of this thesis validate this assumption.

4.2.1 Neural Network for Currents Control

Since the FCS-MPC can be extended to predict more than one future steps, also the possibility to predict different prediction horizons h was investigated .
Hence, the formulation of the currents controller was extended as follows:

$$\begin{aligned} \mathbf{S}^*_{\alpha,\beta}(k) = \min_{\mathbf{S}_{\alpha,\beta}(k)} \quad & \sum_{l=1}^{h} \Big(\left\| \mathbf{i}^{ref}_{\alpha,\beta}(k+l) - \mathbf{i}_{\alpha,\beta}(k+l) \right\|_{\mathbf{Q}} + \\ & + \left\| \mathbf{S}_{\alpha,\beta}(k+l-2) - \mathbf{S}_{\alpha,\beta}(k+l-1) \right\|_{\mathbf{P}} \Big) \\ s.t. \quad & \mathbf{i}_{\alpha,\beta}(k+1) = \mathbf{A}\, \mathbf{i}_{\alpha,\beta}(k) + \mathbf{B}\, \mathbf{S}_{\alpha,\beta}(k) + \mathbf{F}\, \mathbf{v}_{s(\alpha,\beta)}(k) \\ & \mathbf{S}_{\alpha,\beta}(k) \in \mathbf{V}_{\alpha,\beta}, \end{aligned} \tag{4.1}$$

where the currents are predicted from the time step $k+1$ to $k+h$ and the deviation from their reference value are included in the overall cost function, weighted by the matrices $\mathbf{Q} = q_i \times \mathbf{I}_2$ and $\mathbf{P} = p_i \times \mathbf{I}_2$.
For the current control problem, a shallow neural network, namely $\mathrm{NN}_{\alpha,\beta}$, is trained to learn $\mathbf{S}^*_{\alpha,\beta}(k)$ taking as input the same inputs of the minimization problem (4.1), i.e., $\mathbf{i}^{ref}_{\alpha,\beta}(k)$, $\mathbf{i}_{\alpha,\beta}(k)$, $\mathbf{v}_{s(\alpha,\beta)}(k)$, $\mathbf{S}_{\alpha,\beta}(k-1)$. The output $\mathbf{out}_{\mathrm{NN}_{\alpha,\beta}} = \left[out_{\mathrm{NN}_\alpha}, out_{\mathrm{NN}_\beta}\right]^T$ identifies the axis in the α,β frame where the optimal vector should lay, colored in blue and red in Fig. 4.1. Since the three phases can output both positive and negative voltages, considering also the zero, there are $4n+1$ possible values for $S_\alpha(k)$ and $S_\beta(k)$, so: $\mathrm{out}_{\mathrm{NN}_{\alpha,\beta}}, \mathrm{out}_{\mathrm{NN}_{\alpha,\beta}} \in \{-2n, ..., 2n\}$. A regression NN is used in order to obtain continuous values, that has to be processed in order to obtain the discrete switching vector. Due to the approximation of the neural network, the resulting couple $\mathbf{S}_{\alpha,\beta}(k)$ could be an infeasible vector, i.e., it can not lay on the discrete space vector in Fig. 4.1. In fact, the vector is a feasible combination only when the α,β coordinates lay on axes

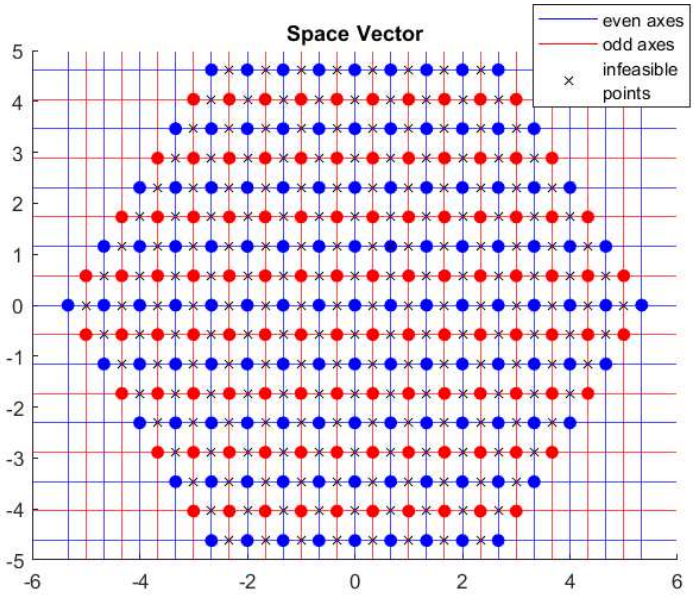


Figure 4.1: Space vector $\mathbf{V}_{\alpha,\beta}$ for a $n = 4$ CHB inverter.

that are both "even" or both "odd", i.e., the full-colored dots in Fig. 4.1, which belong to the set $\mathbf{V}_{\alpha,\beta}$; otherwise, the switching vector has to be reconstructed starting from $\mathbf{out}_{\mathrm{NN}_{\alpha,\beta}}$.
If the vector computed by the NN is not a feasible vector, the feasible point closest to the NN output is chosen. One way is to compute the closest vector in α,β frame, as summarized in in Algorithm 4.1.
An other way is to project the obtained $\mathbf{out}_{\mathrm{NN}_{\alpha,\beta}}$ into a reference frame in which a feasible vector is obtained by simply rounding the result. Hence, $\mathbf{out}_{\mathrm{NN}_{\alpha,\beta}}$ is transformed into a reference frame, in which the α axis is delayed by 60° obtaining the $\mathbf{S}^{60}_{\alpha,\beta}$ vector:

Algorithm 4.1 Neural network for currents tracking algorithm.

procedure $\text{CURRENT}_{\text{NN}_{\alpha,\beta}}(\mathbf{i}^{ref}_{\alpha,\beta}(k), \mathbf{i}_{\alpha,\beta}(k), \mathbf{v}_{s(\alpha,\beta)}(k))$
  $\mathbf{out}_{\text{NN}_{\alpha,\beta}} \leftarrow \text{NN}_{\alpha,\beta}\left(\mathbf{i}^{ref}_{\alpha,\beta}(k), \mathbf{i}_{\alpha,\beta}(k), \mathbf{v}_{s(\alpha,\beta)}(k)\right)$
end procedure
$\mathbf{axis}_{\alpha,\beta} \leftarrow \text{round}\left(\mathbf{out}_{\text{NN}_{\alpha,\beta}}\right)$
$\mathbf{distance}_{\alpha,\beta} \leftarrow \mathbf{out}_{\text{NN}_{\alpha,\beta}} - \mathbf{axis}_{\alpha,\beta}$
if $axis_\alpha, axis_\beta$ even $\vee$ $axis_\alpha, axis_\beta$ odd **then**
  $axis^*_\alpha \leftarrow axis_\alpha$
  $axis^*_\beta \leftarrow axis_\beta$
else
  if $|distance_\alpha| \leq |distance_\beta|$ **then**
    $axis^*_\alpha \leftarrow axis_\alpha$
    $axis^*_\beta \leftarrow axis_\beta + \text{sign}(distance_\beta)$
  else
    $axis^*_\beta \leftarrow axis_\beta$
    $axis^*_\alpha \leftarrow axis_\alpha + \text{sign}(distance_\alpha)$
  end if
end if
$S^*_\alpha \leftarrow axis_\alpha/3$
$S^*_\beta \leftarrow axis_\beta/\sqrt{3}$

$$\mathbf{S}^{60}_{\alpha,\beta} = \sqrt{3}\begin{bmatrix}\cos\left(-\frac{\pi}{3}\right) & \sin\left(-\frac{\pi}{3}\right)\\ 0 & 1\end{bmatrix}\mathbf{out}_{\text{NN}_{\alpha,\beta}} = \begin{bmatrix}\frac{3}{2} & -\frac{\sqrt{3}}{2}\\ 0 & \sqrt{3}\end{bmatrix}\mathbf{out}_{\text{NN}_{\alpha,\beta}} = \mathbf{T}_{60}\,\mathbf{out}_{\text{NN}_{\alpha,\beta}}.$$

In this new reference frame, the feasible points are arranged on a grid with orthogonal axes. The transformation also provides a scale factor, such that the feasible points have integer values. In this way, the feasible switching vector is obtained by rounding $\mathbf{S}^{60}_{\alpha\beta}$. Than, the inverse transformation into the α, β frame leads to:

$$\mathbf{S}^*_{\alpha,\beta} = \begin{bmatrix}\frac{2}{3} & \frac{1}{3}\\ 0 & \frac{1}{\sqrt{3}}\end{bmatrix}\mathbf{S}^{60}_{\alpha,\beta}.$$

The neural network for the currents control with the reference transformation is shown in Fig. 4.2.

4.2.2 Computational Analysis of the Neural Network Approach

In a feed-forward shallow NN (with only one hidden layer), the total number of sums and products computed in order to obtain the result depends on the number of inputs, outputs and hidden neurons. In order to simplify cost analysis, given n_i the number of inputs, it is reasonable to assume that both outputs and neurons are $\mathcal{O}(n_i)$. The shallow NN forward propagation can be modeled as a vector-matrix multiplication and the procedure requires to calculate twice a multiplication between a $n_i \times n_i$ matrix and a $n_i \times 1$ vector

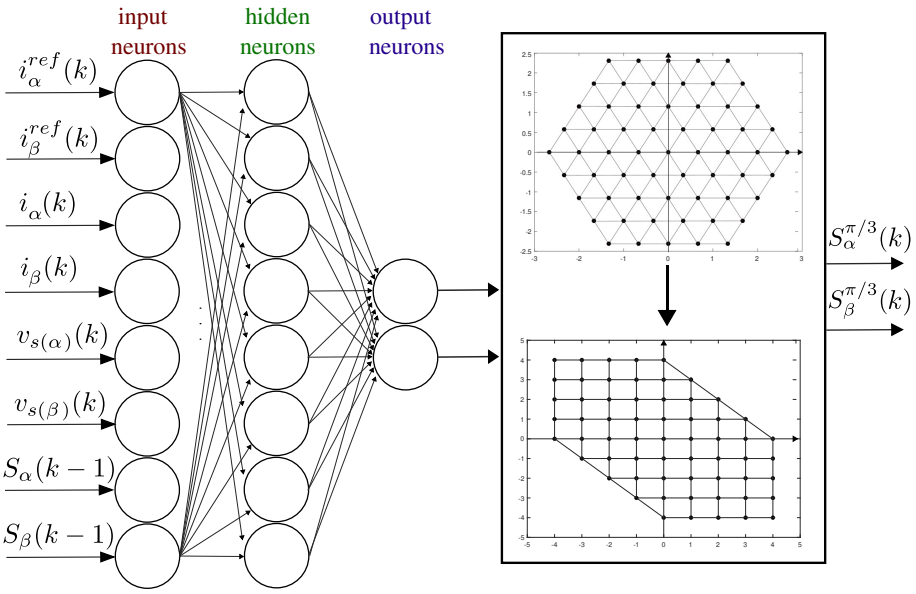


Figure 4.2: Neural network $\text{NN}_{\alpha,\beta}$ for currents control with coordinate transformation.

followed by the n_i evaluations of activation functions, resulting in a cost complexity of $\mathcal{O}\left(n_i^2\right)$.
In the proposed setup, the number of inputs and outputs of $\text{NN}_{\alpha,\beta}$ is fixed to 6 and 1 respectively, and does not directly depend on the number n of H-bridges. Thus, the computational cost of Algorithm 4.1 only depends on the chosen NN architecture and it can be considered as $\mathcal{O}\left(1\right)$.
Moreover, the matrix-vector multiplication and the computation of the activation functions can be fully parallelized, further increasing the efficiency of the NN timing performance.

4.3 Conclusions

This Chapter presents a neural network based method to speed-up the FCS-MPC for CHB inverters. It proposes to train a shallow neural network to learn the optimal control law and to accelerate the computations of the currents controller. The computational cost analysis is presented, showing a qualitative evaluation of the computational load advantages of the proposed methodology compared to the standard solution. The neural network approach has the significant advantage to have a number of computations independent of the number of levels of the converter, overcoming the other methodologies existing in the literature.

References

[1] C. D. Townsend, T. J. Summers, and R. E. Betz, "Multigoal heuristic model predictive control technique applied to a cascaded h-bridge statcom," *IEEE Transactions on Power Electronics*, vol. 27, no. 3, pp. 1191–1200, 2012.

[2] Y. Zhang, X. Wu, and X. Yuan, "A simplified branch and bound approach for model predictive control of multilevel cascaded h-bridge statcom," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 10, pp. 7634–7644, 2017.

[3] M. R. Nasiri, S. Farhangi, and J. Rodriguez, "Model predictive control of a multilevel chb statcom in wind farm application using diophantine equations," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 2, pp. 1213–1223, 2019.

[4] Y. Zhang, X. Yuan, X. Wu, Y. Yuan, and J. Zhou, "Parallel implementation of model predictive control for multilevel cascaded h-bridge statcom with linear complexity," *IEEE Transactions on Industrial Electronics*, vol. 67, no. 2, pp. 832–841, 2020.

[5] P. Cortés, A. Wilson, S. Kouro, J. Rodriguez, and H. Abu-Rub, "Model predictive control of multilevel cascaded h-bridge inverters," *IEEE Transactions on Industrial Electronics*, vol. 57, no. 8, pp. 2691–2699, 2010.

[6] R. Baidya, R. P. Aguilera, P. Acuña, S. Vazquez, and H. d. T. Mouton, "Multistep model predictive control for cascaded h-bridge inverters: Formulation and analysis," *IEEE Transactions on Power Electronics*, vol. 33, no. 1, pp. 876–886, 2018.

[7] Z. Mynar, L. Vesely, and P. Vaclavek, "Pmsm model predictive control with field-weakening implementation," *IEEE Transactions on Industrial Electronics*, vol. 63, no. 8, pp. 5156–5166, 2016.

[8] T. Atalik, M. Deniz, E. Koc, C. Ö. Gercek, B. Gultekin, M. Ermis, and I. Cadirci, "Multi-dsp and -fpga-based fully digital control system for cascaded multilevel converters used in facts applications," *IEEE Transactions on Industrial Informatics*, vol. 8, no. 3, pp. 511–527, 2012.

[9] L. Diao, J. Tang, P. C. Loh, S. Yin, L. Wang, and Z. Liu, "An efficient dsp–fpga-based implementation of hybrid pwm for electric rail traction induction motor control," *IEEE Transactions on Power Electronics*, vol. 33, no. 4, pp. 3276–3288, 2018.

[10] 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.

[11] Z. Zhang, F. Wang, T. Sun, J. Rodriguez, and R. Kennel, "Fpga-based experimental investigation of a quasi-centralized model predictive control for back-to-back converters," *IEEE Transactions on Power Electronics*, vol. 31, no. 1, pp. 662–674, 2016.

[12] P. Cortes, J. Rodriguez, C. Silva, and A. Flores, "Delay compensation in model predictive current control of a three-phase inverter," *IEEE Transactions on Industrial Electronics*, vol. 59, no. 2, pp. 1323–1325, 2012.

[13] O. Gulbudak and E. Santi, "Fpga-based model predictive controller for direct matrix converter," *IEEE Transactions on Industrial Electronics*, vol. 63, no. 7, pp. 4560–4570, 2016.

[14] P. Martin Sanchez, O. Machado, E. J. Peña, F. J. Rodriguez, and F. J. Meca, "Fpga-based implementation of a predictive current controller for power converters," *IEEE Transactions on Industrial Informatics*, vol. 9, no. 3, pp. 1312–1321, 2013.

[15] Z. Ni and M. Narimani, "A new fast formulation of model predictive control for chb statcom," in *IECON 2019 - 45th Annual Conference of the IEEE Industrial Electronics Society*, vol. 1, pp. 3493–3498, 2019.

[16] I. S. Mohamed, S. Rovetta, T. D. Do, T. Dragicević, and A. A. Z. Diab, "A neural-network-based model predictive control of three-phase inverter with an output *lc* filter," *IEEE Access*, vol. 7, pp. 124737–124749, 2019.

[17] D. Wang, Z. J. Shen, X. Yin, S. Tang, X. Liu, C. Zhang, J. Wang, J. Rodriguez, and M. Norambuena, "Model predictive control using artificial neural network for power converters," *IEEE Transactions on Industrial Electronics*, vol. 69, no. 4, pp. 3689–3699, 2022.

[18] S. Wang, T. Dragicevic, Y. Gao, and R. Teodorescu, "Neural network based model predictive controllers for modular multilevel converters," *IEEE Transactions on Energy Conversion*, vol. 36, no. 2, pp. 1562–1571, 2021.

[19] M. Novak and T. Dragicevic, "Supervised imitation learning of finite-set model predictive control systems for power electronics," *IEEE Transactions on Industrial Electronics*, vol. 68, no. 2, pp. 1717–1723, 2020.

[20] F. Simonetti, G. D. Di Girolamo, A. D'Innocenzo, and C. Cecati, "Machine learning for model predictive control of cascaded h-bridge inverters," in *2022 IEEE 21st Mediterranean Electrotechnical Conference (MELECON)*, pp. 1241–1246, IEEE, 2022.

[21] M. Novak and F. Blaabjerg, "Supervised imitation learning of fs-mpc algorithm for multilevel converters," in *2021 23rd European Conference on Power Electronics and Applications (EPE'21 ECCE Europe)*, pp. P–1, IEEE, 2021.

[22] F. Simonetti, A. D'Innocenzo, and C. Cecati, "Neural network model predictive control for chb converters with fpga implementation," *IEEE Transactions on Industrial Informatics*, pp. 1–12, 2023.

Chapter 5

Simple Explicit Solution for FCS-MPC

The neural network controller turned out to be an effective solution for approximating the optimal controller, while ensuring a short computation delay. However, it inherently leads to a sub-optimal solution. For this reason, this research also focused on an analytical solution to solve the optimal currents control problem in real-time. By taking advantage of the intrinsic characteristics of this mathematical problem, a simple solution to find the optimal control input in a few calculations was developed [21].
This Chapter describes the analytical algorithm developed to compute the optimal currents control problem.
In this Chapter, in order to make the methodology more general, we assume that we want to apply the computed control input at the start of next sampling interval, which is a common assumption in digital controllers. It can be underlined that the methods can be easily simplified to the case where we apply the inputs in the same control interval, if we want to neglect the computation delays.
As underlined in Fig. 5.1, the controller was designed to meet the following timing diagram:

- at time k, the optimal control input $u^*(k)$ is applied and the control variable $x(k)$ of the system is sampled;
- by using the value of $x(k)$ and the applied control input $u^*(k)$, the state variable at next time $x(k+1)$ is obtained by employing the dynamical model of the system;
- for each possible discrete control input $u(k+1)$, the two-steps ahead state variables $x(k+2)$ are computed; the cost function is evaluated for each different prediction and the optimal input $u^*(k+1)$ is selected to be the one that minimizes the cost;
- at next time step, it will be applied and the loop starts again.

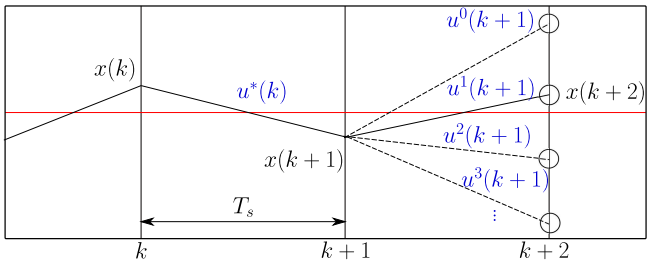


Figure 5.1: Digital controller timing diagram.

5.1 Introduction

As already pointed out, the main disadvantage of FCS-MPC is its large number of computations, which rapidly grows when the converter number of levels increases.
Many works can be found in the literature to reduce the computational burden of the FCS-MPC algorithm for CHB inverters. A simple approach consists of searching the optimum input in a subset of all the possible combinations, as in [4, 2, 1, 3]. For example, the manuscript [4] limits the search set to the vectors that are nearest to the last applied vector, reducing the number of calculations to only seven predictions. It leads to a sub-optimal solution, which is effective in the steady-state but negatively affects the transient performance.
Other approaches explicitly use the dynamical model of the converter without solving an optimization problem [9, 15].
By contrast, model-free predictive control avoids using a nominal model [20]. It uses past data samples to make predictions of future states, creating a data-driven controller that improves the robustness to parametric variations. However, the optimization problem is addressed by limiting the search space, leading to a sub-optimal result.
The sphere decoding algorithm was successfully investigated in [10, 18, 19], reducing the number of computations when using multiple prediction horizons, and it is referred to as "multistep FCS-MPC" or "long-horizon FCS-MPC". One of the major problems of this algorithm is the choice of the initial radius, which strongly affects the computational load during fast transients. Moreover, the computational improvement relies on the fact that the search space is limited to the vectors adjacent to the previous control input, implying a sub-optimal solution.
Despite the improvements when using multiple prediction horizons in motor drives [16, 17], many applications employ the FCS-MPC formulation with one prediction horizon. In grid-tied inverters, for instance, the iterative predictions of the output variable would rely on the prediction of the grid voltage, which is generally not feasible and leads to inaccuracy. In [5, 6, 7], authors applied FCS-MPC to a CHB-STATCOM, dividing the overall optimization problem into sub-problems, strongly reducing the total computational burden. In [5], the authors individually compute the currents and voltages optimization sub-problems, reducing the computational burden from exponential to polynomial level. In [8], the authors further improved the currents optimization, proposing an algorithm with linear complexity. In [11, 12, 14, 13], authors proposed machine learning techniques to speed up the online implementation of the control. However, the obtained control low is a sub-optimal solution.

This Chapter presents the developed analytical method for finding the global optimal solution of the currents control problem of FCS-MPC for CHB inverters. The main contribution of this approach is that it allows computing the global optimal switching vector with a few simple mathematical operations regardless of the number of levels, thus overcoming the existing methodologies.

5.2 Solution of the Currents Control Problem

The two-step ahead currents model predictive control problem is expressed as:

$$\begin{aligned} \min_{\mathbf{S}_{\alpha,\beta}(k+1)} \quad & \left\| \mathbf{i}_{\alpha,\beta}^{ref}(k+2) - \mathbf{i}_{\alpha,\beta}(k+2) \right\|_{\mathbf{Q}} + \\ & + \left\| \mathbf{S}_{\alpha,\beta}(k+1) - \mathbf{S}_{\alpha,\beta}(k) \right\|_{\mathbf{P}} \\ s.t. \quad & \mathbf{i}_{\alpha,\beta}(k+1) = \mathbf{A}\ \mathbf{i}_{\alpha,\beta}(k) + \mathbf{B}\ \mathbf{S}_{\alpha,\beta}(k) + \mathbf{F}\ \mathbf{v}_{s(\alpha,\beta)}(k) \\ & \mathbf{S}_{\alpha,\beta}(k+1) \in \mathbf{V}_{\alpha,\beta}, \end{aligned} \tag{5.1}$$

where $\mathbf{Q}$ and $\mathbf{P}$ are the two weighting matrices and $\mathbf{i}_{\alpha,\beta}^{ref}(k+2)$ is the predicted reference value. The optimal switching vector must belong to $\mathbf{V}_{\alpha,\beta}$, the set of all the vectors that the inverter can physically generate. In order to compute the control at time $k+1$, the currents at time $k+2$ must be computed. Iterating the currents equation (2.5), the following formulation is obtained:

$$\begin{aligned} \mathbf{i}_{\alpha,\beta}(k+2) =& \mathbf{A}\ \mathbf{i}_{\alpha,\beta}(k+1) + \mathbf{B}\ \mathbf{S}_{\alpha,\beta}(k+1) + \mathbf{F}\ \mathbf{v}_{s(\alpha,\beta)}(k+1) \\ =& \mathbf{A}^2\ \mathbf{i}_{\alpha,\beta}(k) + \mathbf{AB}\ \mathbf{S}_{\alpha,\beta}(k) + \mathbf{AF}\ \mathbf{v}_{s(\alpha,\beta)}(k) \\ & + \mathbf{B}\ \mathbf{S}_{\alpha,\beta}(k+1) + \mathbf{F}\ \mathbf{v}_{s(\alpha,\beta)}(k+1), \end{aligned} \tag{5.2}$$

with two unknowns: the control input at next time $\mathbf{S}_{\alpha,\beta}(k+1)$, which is the variable to be computed, and the grid voltage at next time $\mathbf{v}_{s(\alpha,\beta)}(k+1)$, which must be predicted. It is reasonable to approximate the grid voltage at time $k+1$ as the voltage at time k rotated by one step at line frequency ω, as follows:

$$\mathbf{v}_{s(\alpha,\beta)}(k+1) = \mathbf{T}\ \mathbf{v}_{s(\alpha,\beta)}(k),\ \mathbf{T} = \begin{bmatrix} \cos(\omega T_s) & -\sin(\omega T_s) \\ \sin(\omega T_s) & \cos(\omega T_s) \end{bmatrix}.$$

With this assumption, the currents at time $k+2$ are given by:

$$\begin{aligned} \mathbf{i}_{\alpha,\beta}(k+2) =& \mathbf{A}^2\ \mathbf{i}_{\alpha,\beta}(k) + \mathbf{AB}\ \mathbf{S}_{\alpha,\beta}(k) + (\mathbf{AF} + \mathbf{FT})\ \mathbf{v}_{s(\alpha,\beta)}(k) \\ & + \mathbf{B}\ \mathbf{S}_{\alpha,\beta}(k+1) \\ =& \mathbf{A}^2\ \mathbf{i}_{\alpha,\beta}(k) + \mathbf{E}\ \mathbf{S}_{\alpha,\beta}(k) + \mathbf{G}\ \mathbf{v}_{s(\alpha,\beta)}(k) \\ & + \mathbf{B}\ \mathbf{S}_{\alpha,\beta}(k+1). \end{aligned} \tag{5.3}$$

The currents reference at time $k+2$ can be predicted by $\mathbf{i}_{\alpha,\beta}^{ref}(k+2) = \mathbf{T}^2\mathbf{i}_{\alpha,\beta}^{ref}(k)$.
The proposed method aims to solve the discrete optimization problem in (5.1) by solving,

at first, the equivalent continuous problem; then, the discrete solution is obtained based on the continuous solution. Assuming the variables to be continuous, the problem in (5.1) becomes a quadratic programming problem since there is a quadratic cost function and the optimization variables must belong to the hexagonal space vector, which is defined by linear constraints. This problem can be solved by using standard quadratic programming solver [22]. However, since the iteration of the Newton-Raphson algorithm can need several steps, several iterations could be needed, a simple way to solve this quadratic and constrained problem is proposed.
The overall problem is divided into three distinct steps. First, the equivalent continuous unconstrained problem is solved; then, the unconstrained solution is projected into the space vector to find the continuous constrained solution; finally, the discrete optimum solution is found based on the continuous constrained solution.

5.2.1 Continuous Unconstrained Problem

Let's call $\mathbf{S}^{c}_{\alpha,\beta}$ the continuous unconstrained optimum switching vector. The minimum problem of the model predictive control in (2.6) becomes:

$$\begin{aligned}\mathbf{S}^{c}_{\alpha,\beta} = \min_{S_{\alpha,\beta}} &\left(\mathbf{i}^{ref}_{\alpha,\beta}\left(k+2\right) - \mathbf{i}_{\alpha,\beta}\left(k+2\right)\right)^{T}\mathbf{Q}\ \left(\mathbf{i}^{ref}_{\alpha,\beta}\left(k+2\right) - \mathbf{i}_{\alpha,\beta}\left(k+2\right)\right) \\ &+ \left(\mathbf{S}_{\alpha,\beta} - \mathbf{S}_{\alpha,\beta}\left(k\right)\right)^{T}\mathbf{P}\ \left(\mathbf{S}_{\alpha,\beta} - \mathbf{S}_{\alpha,\beta}\left(k\right)\right),\end{aligned} \tag{5.4}$$

where $\mathbf{S}_{\alpha,\beta}$ is the optimization variable and $\mathbf{i}_{\alpha,\beta}\left(k+2\right)$ is predicted by using (5.3). The solution of (5.4) is simply given by computing the derivative of the cost function with respect to $\mathbf{S}_{\alpha,\beta}$ and setting it to zero, which leads to the following result:

$$\begin{aligned}\mathbf{S}^{c}_{\alpha,\beta} = &\left(\mathbf{B}^{T}\mathbf{Q}\ \mathbf{B} + \mathbf{P}\right)^{-1}\left(-\mathbf{B}^{T}\mathbf{Q}\ \mathbf{A}^{2}\mathbf{i}_{\alpha,\beta}\left(k\right) - \left(\mathbf{E}^{T}\mathbf{Q}\ \mathbf{B} + \mathbf{P}\right)\ \mathbf{S}_{\alpha,\beta}\left(k\right)\right. \\ &\left.-\mathbf{B}^{T}\mathbf{Q}\ \mathbf{G}\ \mathbf{v}_{s(\alpha,\beta)}\left(k\right) + \mathbf{B}^{T}\mathbf{Q}\ \mathbf{T}^{2}\ \mathbf{i}^{ref}_{\alpha,\beta}\left(k\right)\right).\end{aligned} \tag{5.5}$$

5.2.2 Continuous Constrained Problem

Once the optimal continuous unconstrained solution is computed, it is needed to compute $\mathbf{S}^{proj}_{\alpha,\beta}$, i.e., the projection of $\mathbf{S}^{c}_{\alpha,\beta}$ into the hexagonal space of feasible solutions. The hexagonal space is a polytope described by six inequalities. Given n H-bridges per phase, the vertices of the polytope, by construction, are:

$$\mathbf{V} = \left\{\begin{bmatrix}\frac{4}{3}\\ \\ 0\end{bmatrix}, \begin{bmatrix}\frac{2}{3}\\ \\ -\frac{2}{\sqrt{(3)}}\end{bmatrix}, \begin{bmatrix}-\frac{2}{3}\\ \\ -\frac{2}{\sqrt{(3)}}\end{bmatrix}, \begin{bmatrix}-\frac{4}{3}\\ \\ 0\end{bmatrix}, \begin{bmatrix}-\frac{2}{3}\\ \\ \frac{2}{\sqrt{(3)}}\end{bmatrix}, \begin{bmatrix}\frac{2}{3}\\ \\ \frac{2}{\sqrt{(3)}}\end{bmatrix}\right\} \cdot n. \tag{5.6}$$

By simply computing the equations of the lines passing through two consecutive vertices, it is possible to obtain the constraints of the hexagon, shown in Fig. 5.2. Given $\overline{m} = \sqrt{3}$ and $\overline{q} = 4n/\sqrt{3}$, the constraints are:

$$\begin{array}{ll} \text{constraint 1: } S_\beta \leq \text{-}\overline{m}S_\alpha + \overline{q} & \text{constraint 2: } S_\beta \geq \overline{m}S_\alpha \text{-} \overline{q} \\ \text{constraint 3: } S_\beta \geq \text{-}\overline{m}S_\alpha \text{-} \overline{q} & \text{constraint 4: } S_\beta \leq \overline{m}S_\alpha + \overline{q} \\ \text{constraint 5: } S_\beta \leq 2\text{n}/\sqrt{3} & \text{constraint 6: } S_\beta \geq \text{-}2\text{n}/\sqrt{3}, \end{array} \tag{5.7}$$

where S_α and S_β are continuous point on the α and β axes. If one of these inequalities

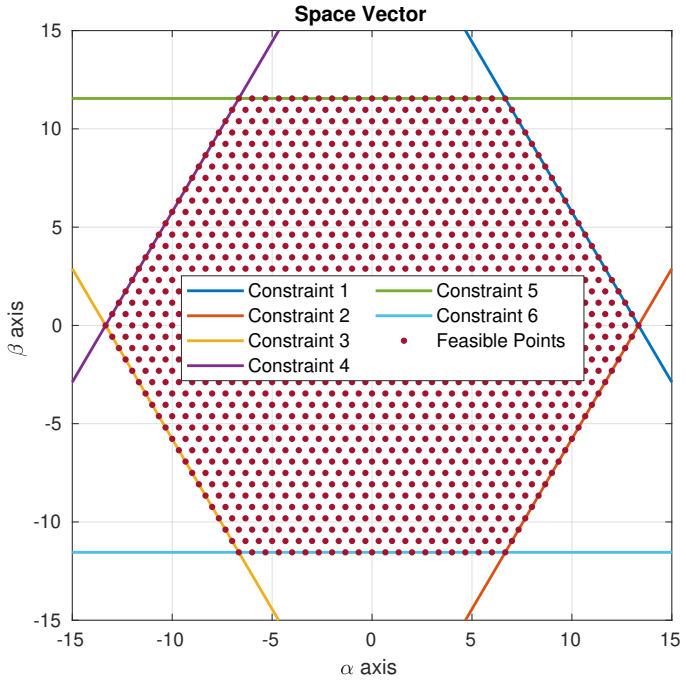


Figure 5.2: Space vector with constraints for a n=10 CHB inverter.

is not satisfied for the computed solution $\mathbf{S}^c_{\alpha,\beta}$, it is necessary to identify in which region outside the hexagon the point lays. Once the region is determined, the constrained solution is computed by projecting the point on the violated constraint, i.e., computing the point on the violated constraint that minimizes the distance between the solution and the constraint. Let's refer to the violated constraint as $S_\beta = mS_\alpha + q$, with m and q as in (5.7). This problem can be expressed as:

$$\begin{aligned} \mathbf{S}^{proj}_{\alpha,\beta} &= \min_{\mathbf{S}_{\alpha,\beta}} \left(\mathbf{S}_{\alpha,\beta} - \mathbf{S}^c_{\alpha,\beta}\right)^T \left(\mathbf{B}^T\mathbf{Q}\,\mathbf{B} + \mathbf{P}\right) \left(\mathbf{S}_{\alpha,\beta} - \mathbf{S}^c_{\alpha,\beta}\right) \\ &\quad \text{s. t.} \quad S_\beta = mS_\alpha + q \\ &= \min_{S_\alpha, S_\beta} \left(\begin{bmatrix} S_\alpha \\ S_\beta \end{bmatrix} - \begin{bmatrix} S^c_\alpha \\ S^c_\beta \end{bmatrix} \right)^T \begin{bmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \end{bmatrix} \left(\begin{bmatrix} S_\alpha \\ S_\beta \end{bmatrix} - \begin{bmatrix} S^c_\alpha \\ S^c_\beta \end{bmatrix} \right) \\ &\quad \text{s. t.} \quad S_\beta = mS_\alpha + q \end{aligned} \tag{5.8}$$

where the matrix $\left(\mathbf{B}^T\mathbf{Q}\,\mathbf{B}+\mathbf{P}\right)$ is the quadratic term of the cost function. In general, it does not make sense to weigh the α coordinate differently from the β coordinate and $\mathbf{Q}$ and $\mathbf{P}$ are diagonal matrices. Consequently, $w_{11}=w_{22}$ and $w_{12}=w_{21}=0$, so the level curves of the cost function are circles and it is possible to find the projection by simply applying the point-to-line projection formula. Indeed, with these assumptions, (5.8) becomes:

$$\begin{aligned}\mathbf{S}_{\alpha,\beta}^{proj} = \min_{S_\alpha,S_\beta} \quad & \left(S_\alpha - S_\alpha^c\right)^2 + \left(S_\beta - S_\beta^c\right)^2 \\ \text{s. t.} \quad & S_\beta = mS_\alpha + q,\end{aligned} \tag{5.9}$$

which leads to

$$S_\alpha^{proj} = \min_{S_\alpha} \quad \left(S_\alpha - S_\alpha^c\right)^2 + \left(mS_\alpha + q - S_\beta^c\right)^2. \tag{5.10}$$

The solution S_α^{proj} is simply computed by setting the derivative with respect to S_α to zero:

$$\begin{cases} S_\alpha^{proj} = & \left(S_\alpha^c + m\left(S_\beta^c - q\right)\right)/\left(1+m^2\right) \\ S_\beta^{proj} = & mS_\alpha + q. \end{cases} \tag{5.11}$$

If the violated constraints are two, the projection is one of the vertices of the hexagon. The constraints that determine the regions are computed by substituting S_α^c, S_β^c with the vertices in (5.6) in the projection formulas in (5.11): in this way, the separation lines between the regions are found.
The constraints of the regions are the following, where $\hat{m}=-1/\overline{m}$ and $\hat{q}=\overline{q}/3$:

$$\begin{array}{ll} \text{constraint 7: } S_\beta \geq \text{-}\hat{m}S_\alpha + \hat{q} & \text{constraint 8: } S_\beta \leq \text{-}\hat{m}S_\alpha \text{-} \hat{q} \\ \text{constraint 9: } S_\beta \geq \hat{m}S_\alpha + \hat{q} & \text{constraint 10: } S_\beta \leq \hat{m}S_\alpha \text{-} \hat{q} \\ \text{constraint 11: } S_\alpha \leq \text{-}2n/\sqrt{3} & \text{constraint 12: } S_\alpha \geq 2n/\sqrt{3}. \end{array} \tag{5.12}$$

Fig. 5.3 shows the different regions outside the hexagonal space vector and underlines the constraints of the regions. The set $\mathbf{V}_{\alpha,\beta}$ of feasible points is computed by transforming in the α,β frame all the a,b,c switching vectors that the inverter can output. The figure also shows an example of a generic unconstrained solution $\mathbf{S}_{\alpha,\beta}^c$ that lays outside the polytope. As evident, the level curves of the quadratic cost function are circles and the optimal constrained solution is the point related to the lower level curve, i.e., the projection $\mathbf{S}_{\alpha,\beta}^{proj}$ of the point into the hexagon. Tab. 5.1 summarizes the projection rule. In the first column the regions are listed; in the second one the constraints that define the regions are shown, where not$(*)$ means that the constraint is violated; the last column is related to the projection rule of the different regions, where the solution can be a projection on an edge or a vertex of the hexagon.

5.2.3 Discrete Problem

Due to the quadratic cost function, the optimal point must be one of the feasible points close to the continuous constrained solution $\mathbf{S}_{\alpha,\beta}^{proj}$. Since the searched point lays on a

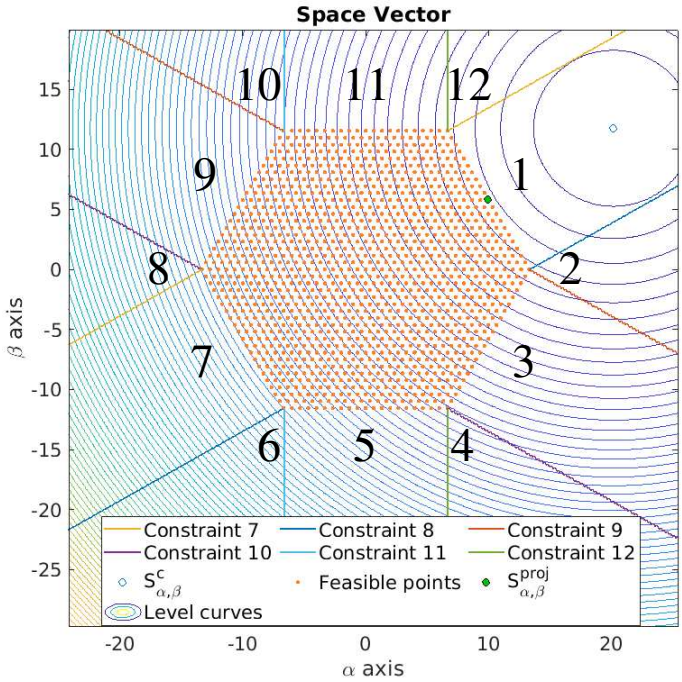


Figure 5.3: Space vector with level curves for a n=10 CHB-inverter.

plane, the candidate points are four. To simplify the computations, the possible points $\mathbf{S}_{\alpha,\beta}$ are scaled as follows:

$$\tilde{\mathbf{S}}_{\alpha,\beta} = \begin{bmatrix} 3 & 0 \\ 0 & \sqrt{3} \end{bmatrix} \mathbf{S}_{\alpha,\beta} = \quad \mathbf{H}\mathbf{S}_{\alpha\beta}. \tag{5.13}$$

The scaled feasible points are equidistant and have integer values. The quadratic part of the cost function, with the previous assumptions, is given by:

$$\begin{aligned} \mathbf{S}_{\alpha,\beta}^T \left(\mathbf{B}^T\mathbf{Q}\ \mathbf{B}+\mathbf{P}\right) \mathbf{S}_{\alpha,\beta} &= \tilde{\mathbf{S}}_{\alpha,\beta}^T \mathbf{H}^{-1} \left(\mathbf{B}^T\mathbf{Q}\ \mathbf{B}+\mathbf{P}\right) \mathbf{H}^{-1} \tilde{\mathbf{S}}_{\alpha,\beta} \\ &= \tilde{\mathbf{S}}_{\alpha,\beta}^T \begin{bmatrix} \frac{1}{3} & 0 \\ 0 & \frac{1}{\sqrt{3}} \end{bmatrix} \begin{bmatrix} w_{11} & 0 \\ & \\ 0 & w_{11} \end{bmatrix} \begin{bmatrix} \frac{1}{3} & 0 \\ 0 & \frac{1}{\sqrt{3}} \end{bmatrix} \tilde{\mathbf{S}}_{\alpha,\beta} \\ &= \tilde{\mathbf{S}}_{\alpha,\beta}^T \begin{bmatrix} \frac{w_{11}}{9} & 0 \\ 0 & \frac{w_{11}}{3} \end{bmatrix} \tilde{\mathbf{S}}_{\alpha,\beta} = \tilde{\mathbf{S}}_{\alpha,\beta}^T \tilde{\mathbf{W}} \tilde{\mathbf{S}}_{\alpha,\beta}, \end{aligned} \tag{5.14}$$

where, in these new variables, $\tilde{\mathbf{W}}$ weights differently α and β directions and the relative level curves are ellipses. Since $\tilde{\mathbf{W}}$ is a diagonal matrix, the axis of the ellipses are aligned with α, β axes and, if all the integer values would be feasible points, it would be sufficient to round the continuous solution $\tilde{\mathbf{S}}_{\alpha,\beta}^c$ to obtain the optimal point. However, only two

Table 5.1: Projection rule.

	Constraint evaluation	Projection rule
Region 1	not $(1-7-8)$	eq. (5.11): $m=\overline{m}, q=\overline{q}$
Region 2	$8-9$	$\mathbf{S}_{\alpha,\beta}^{proj}=\mathbf{V}_1$
Region 3	not $(2-9-10)$	eq. (5.11): $m=\overline{m}, q=-\overline{q}$
Region 4	$10-11$	$\mathbf{S}_{\alpha,\beta}^{proj}=\mathbf{V}_2$
Region 5	not $(6-11-12)$	$\mathbf{S}_{\alpha,\beta}^{proj}=\left[S_\alpha^c, -\frac{2}{\sqrt{(3)}}n\right]^T$
Region 6	$8-11$	$\mathbf{S}_{\alpha,\beta}^{proj}=\mathbf{V}_3$
Region 7	not $(3-7-8)$	eq. (5.11): $m=-\overline{m}, q=\overline{q}$
Region 8	$7-10$	$\mathbf{S}_{\alpha,\beta}^{proj}=\mathbf{V}_4$
Region 9	not $(4-9-10)$	eq. (5.11): $m=-\overline{m}, q=-\overline{q}$
Region 10	$9-11$	$\mathbf{S}_{\alpha,\beta}^{proj}=\mathbf{V}_5$
Region 11	not $(5-11-12)$	$\mathbf{S}_{\alpha,\beta}^{proj}=\left[S_\alpha^c, \frac{2}{\sqrt{(3)}}n\right]^T$
Region 12	$7-12$	$\mathbf{S}_{\alpha,\beta}^{proj}=\mathbf{V}_6$

of the four points around $\tilde{\mathbf{S}}_{\alpha,\beta}^c$ are feasible by construction. Once the rounded point is computed, it is necessary to verify if it is a feasible point. If it is not, the two feasible points around the continuous solution are evaluated.

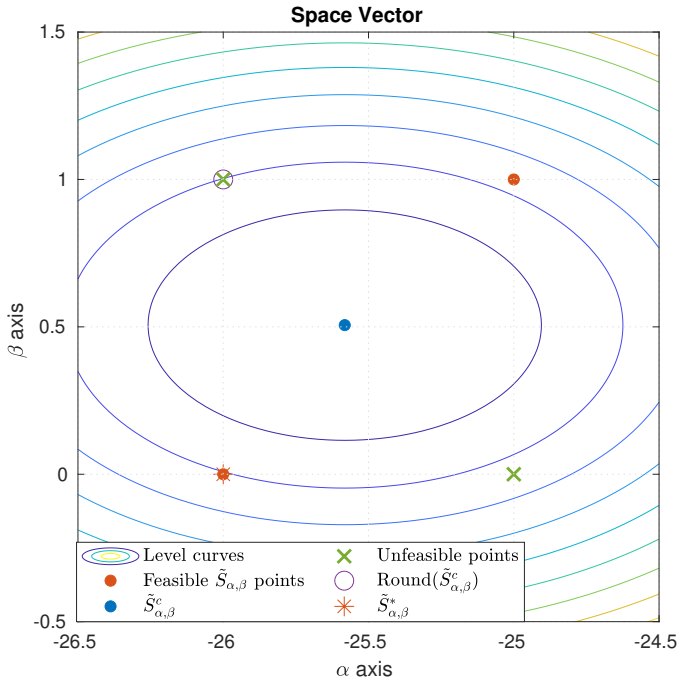


Figure 5.4: Feasible points inside the space vector.

The distances between them and the continuous solution, weighted by $\tilde{\mathbf{W}}$, are computed and the closest one is the solution of the discrete optimization. Fig. 5.4 shows an example of a continuous solution with the four points around it. The rounded solution, in the example, is not a feasible point and, as underlined by the level curves, the optimum point is $[-26, 0]^T$. The flow chart in Fig. 5.5 summarizes the overall procedure.

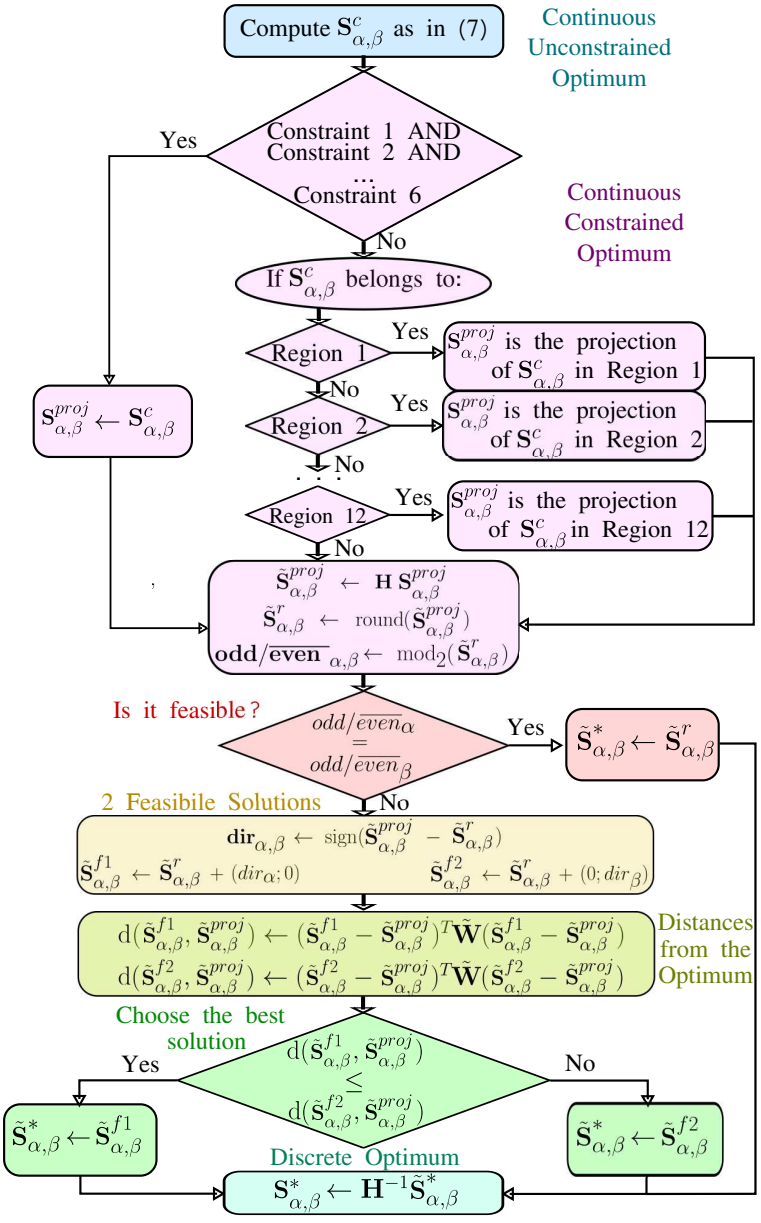


Figure 5.5: Simple explicit solution flow chart.

5.3 Computational Analysis of the Proposed Algorithm

The first part of the algorithm in Subsection 5.2.1 requires the computation of (5.5), which can be expressed as:

$$\mathbf{S}^{c}_{\alpha,\beta} = \mathbf{U}_0\ \mathbf{i}^{ref}_{\alpha,\beta}(k) + \mathbf{U}_1\ \mathbf{i}_{\alpha,\beta}(k) + \mathbf{U}_2\ \mathbf{S}(k) + \mathbf{U}_3\ \mathbf{v}_{s(\alpha,\beta)}(k), \tag{5.15}$$

where

$$\mathbf{U}_0 = \left(\mathbf{B}^T\mathbf{Q}\mathbf{B}+\mathbf{P}\right)^{-1}\mathbf{B}^T\mathbf{Q}\mathbf{T}^2 \qquad \mathbf{U}_1 = -\left(\mathbf{B}^T\mathbf{Q}\mathbf{B}+\mathbf{P}\right)^{-1}\mathbf{B}^T\mathbf{Q}\mathbf{A}^2$$
$$\mathbf{U}_2 = -\left(\mathbf{B}^T\mathbf{Q}\mathbf{B}+\mathbf{P}\right)^{-1}\left(\mathbf{E}^T\mathbf{Q}\mathbf{B}+\mathbf{P}\right) \qquad \mathbf{U}_3 = -\left(\mathbf{B}^T\mathbf{Q}\mathbf{B}+\mathbf{P}\right)^{-1}\mathbf{B}^T\mathbf{Q}\mathbf{G}$$

are 2×2 matrices computed offline. The online computations are the sum of four 2×2 matrices times 2×1 vectors. The total computations are 16 multiplications and 14 sums. In order to compute the constrained solution in (5.2.2), the constraints in (5.7) must be evaluated, which requires 4 multiplications, 4 sums and 6 inequality operators. If one of them is violated, the projection must be computed and one if-than-else is needed. In order to identify the region outside the hexagon, the evaluation of constraints in (5.12) requires 4 multiplications, 4 sums and 6 inequality operators. The projection rule in Tab. 5.1 requires 12 comparisons and the most involved projection rule is in (5.11), which requires 6 online multiplications and sums. The total computations of this phase are: 14 multiplications, 14 sums and 13 comparisons.
In the discretization of the continuous solution in Subsection 5.2.3, the scaling operation in (5.13) and the subsequent inverse scaling to compute $\tilde{\mathbf{S}}^{proj}_{\alpha,\beta}$ and $\mathbf{S}^{*}_{\alpha,\beta}$ in the flow chart in Fig. 5.5 require 4 multiplications. The $\mathbf{odd}/\overline{\mathbf{even}}_{\alpha,\beta}$ in Fig. 5.5 is obtained by taking the first bit of $\tilde{S}^{r}_{\alpha}$ and $\tilde{S}^{r}_{\beta}$, which is the $\mathrm{mod}_2(*)$ operator. To compute the vector $\tilde{\mathbf{S}}^{r}_{\alpha,\beta}$, the round$(*)$ operator is needed. The $\mathbf{dir}_{\alpha,\beta}$ in Fig. 5.5 is the vector containing the sign bits of the result of 2 subtractions. Computing the feasible solutions $\tilde{\mathbf{S}}^{f1}_{\alpha,\beta}$ and $\tilde{\mathbf{S}}^{f2}_{\alpha,\beta}$ requires 2 sums, while the 2 distance operators d$(*)$ require 4 subtractions, 8 multiplications, 2 sums. The total operations are: 12 multiplications, 4 sums, 6 subtractions (or 6 two's complement calculations and 6 sums), 2 rounding and 2 comparisons. Tab. 5.2

Table 5.2: Computational analysis of the algorithm for the simple explicit solution of FCA-MPC.

	5.2.1	5.2.2	5.2.3	Overall Procedure
Sums	14	14	10	38
Multiplications	16	14	12	42
Two's complements	0	0	6	6
Comparisons	0	13	2	15
Rounds	0	0	2	2

summarizes the basic operations needed by the overall procedure. It is clear that, with this proposed method, only a few number of basic operations are needed to solve the

discrete optimization problem irrespective of the number of levels, which is a great result, allowing us to compute the global optimal solution with a small computational burden.

5.4 Conclusions

This Chapter presents a simple analytical explicit solution for FCS-MPC of a CHB converter. The computational cost analysis was carried out and the number of individual operations was computed. The developed algorithm computes the global optimal solution with an extremely low computational burden, which is independent of the number of levels.

References

[1] B. Gutierrez and S.-S. Kwak, "Modular multilevel converters (mmcs) controlled by model predictive control with reduced calculation burden," *IEEE Transactions on Power Electronics*, vol. 33, no. 11, pp. 9176–9187, 2018.

[2] C. D. Townsend, T. J. Summers, and R. E. Betz, "Multigoal heuristic model predictive control technique applied to a cascaded h-bridge statcom," *IEEE Transactions on Power Electronics*, vol. 27, no. 3, pp. 1191–1200, 2012.

[3] Z. Ni, A. H. Abuelnaga, Y. Pan, A. Elezab, O. Zayed, M. Narimani, and J. Rodriguez, "A new mpc formulation based on suboptimal voltage vectors for multilevel inverters," *IEEE Journal of Emerging and Selected Topics in Power Electronics*, vol. 10, no. 6, pp. 7261–7270, 2022.

[4] P. Cortés, A. Wilson, S. Kouro, J. Rodriguez, and H. Abu-Rub, "Model predictive control of multilevel cascaded h-bridge inverters," *IEEE Transactions on Industrial Electronics*, vol. 57, no. 8, pp. 2691–2699, 2010.

[5] Y. Zhang, X. Wu, X. Yuan, Y. Wang, and P. Dai, "Fast model predictive control for multilevel cascaded h-bridge statcom with polynomial computation time," *IEEE Transactions on Industrial Electronics*, vol. 63, no. 8, pp. 5231–5243, 2016.

[6] A. Dekka, B. Wu, V. Yaramasu, and N. R. Zargari, "Dual-stage model predictive control with improved harmonic performance for modular multilevel converter," *IEEE Transactions on Industrial Electronics*, vol. 63, no. 10, pp. 6010–6019, 2016.

[7] J. Huang, B. Yang, F. Guo, Z. Wang, X. Tong, A. Zhang, and J. Xiao, "Priority sorting approach for modular multilevel converter based on simplified model predictive control," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 6, pp. 4819–4830, 2018.

[8] Y. Zhang, X. Wu, and X. Yuan, "A simplified branch and bound approach for model predictive control of multilevel cascaded h-bridge statcom," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 10, pp. 7634–7644, 2017.

[9] M. R. Nasiri, S. Farhangi, and J. Rodriguez, "Model predictive control of a multilevel chb statcom in wind farm application using diophantine equations," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 2, pp. 1213–1223, 2019.

[10] R. Baidya, R. P. Aguilera, P. Acuña, S. Vazquez, and H. d. T. Mouton, "Multistep model predictive control for cascaded h-bridge inverters: Formulation and analysis," *IEEE Transactions on Power Electronics*, vol. 33, no. 1, pp. 876–886, 2018.

[11] F. Simonetti, G. D. Girolamo, A. D'Innocenzo, and C. Cecati, "A neural network approach for efficient finite control set mpc of cascaded h-bridge statcom," in *IECON 2021 – 47th Annual Conference of the IEEE Industrial Electronics Society*, pp. 1–6, 2021.

[12] I. S. Mohamed, S. Rovetta, T. D. Do, T. Dragicević, and A. A. Z. Diab, "A neural-network-based model predictive control of three-phase inverter with an output *lc* filter," *IEEE Access*, vol. 7, pp. 124737–124749, 2019.

[13] S. Wang, T. Dragicevic, Y. Gao, and R. Teodorescu, "Neural network based model predictive controllers for modular multilevel converters," *IEEE Transactions on Energy Conversion*, vol. 36, no. 2, pp. 1562–1571, 2021.

[14] M. Novak and T. Dragicevic, "Supervised imitation learning of finite-set model predictive control systems for power electronics," *IEEE Transactions on Industrial Electronics*, vol. 68, no. 2, pp. 1717–1723, 2020.

[15] Z. Ni, M. Narimani, and J. Rodriguez, "Model predictive control of a four-level t-nnpc inverter without weighting factors," in *2021 IEEE APEC*, pp. 2133–2138, 2021.

[16] A. Tregubov, P. Karamanakos, and L. Ortombina, "Long-horizon robust direct model predictive control for medium-voltage induction motor drives with reduced computational complexity," *IEEE Transactions on Industry Application*, vol. 59, no. 2, pp. 1775–1787, 2023.

[17] P. Karamanakos, T. Geyer, and R. P. Aguilera, "Long-horizon direct model predictive control: Modified sphere decoding for transient operation," *IEEE Transactions on Industry Application*, vol. 54, no. 6, pp. 6060–6070, 2018.

[18] 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.

[19] R. Baidya, R. P. Aguilera, P. Acuña, T. Geyer, R. A. Delgado, D. E. Quevedo, and H. d. T. Mouton, "Enabling multistep model predictive control for transient operation of power converters," *IEEE Open J. Ind. Electron. Soc.*, vol. 1, pp. 284–297, 2020.

[20] P. G. Ipoum-Ngome, D. L. Mon-Nzongo, R. C. C. Flesch, J. Song-Manguelle, M. Wang, and T. Jin, "Model-free predictive current control for multilevel voltage source inverters," *IEEE Transactions on Industrial Electronics*, vol. 68, no. 10, pp. 9984–9997, 2021.

[21] F. Simonetti, A. D'Innocenzo, and C. Cecati, "Simple explicit solution of finite control set model predictive control," *IEEE Transactions on Industrial Electronics*, 2023, unpublished.

[22] A. Bemporad and P. Patrinos, "Simple and certifiable quadratic programming algorithms for embedded linear model predictive control," *IFAC Proceedings Volumes*, vol. 45, no. 17, pp. 14–20, 2012. 4th IFAC Conference on Nonlinear Model Predictive Control.

Part III

Validation of the Proposed Methodologies

Chapter 6

Simulation Results of the Neural Network Approach

This Chapter describes the dataset generation and the train procedure of the neural network controller presented in Chapter 4 and analyzed in terms of computational cost complexity. The theoretical analysis underlined a significant reduction in the overall computational cost of the proposed method with respect to existing ones. Now the question is: assuming that we can replace the FCS-MPC with the neural network controller, does it guarantee the same performance of the original optimal controller?
This Chapter answers to this question, presenting the performance analysis of the NN-MPC and comparing it with the original FCS-MPC in Matlab/Simulink environment [10, 2, 5].
After describing the training procedure of the ML algorithms, it presents the analysis on the approximation capability of the neural networks when the main parameters of the controller and the system change, i.e., number of level of the converter, tuning parameters in the cost function, number of hidden neurons in the NNs, and prediction horizons.

6.1 Power System for the CHB-STATCOM Tests

To verify the proposed FCS-MPC, the power system referenced in [4] was as a test bench. The power system is the 25 kV distribution line shown in the one-line diagram in Fig. 6.1 and comprises a 25 kV 100 MW voltage generator, a power line, a load connected to the 25 kV medium voltage line and a load connected to the grid through a 25 kV/600kV step-up transformer. A $\pm$10 MVAR distribution STATCOM is connected in the node between the two loads, able to compensate power surge up to 20 % of the rated 25 kV voltage at the PCC and it comprises a two-level three–phase inverter connected to the grid through a step-up transformer.In order to verify the proposed techniques, the original two-level STATCOM was replaced by the multilevel CHB-STATCOM.
In this work, the power grid was modified to have a more standard value for medium voltage grid. All the parameters of the benchmark were scaled to obtain a 10 kV 6 MW power system by keeping all the per unit values equal to the original test bench.
One of the advantages of the multilevel inverter is the possibility to reach an higher voltage

by increasing the number of levels, such to avoid the use of a step-up transformer. For this reason, the CHB-STATCOM was designed to be connected to the 10 kV PCC and capable to account for grid voltage variations of ±20 % of the rated voltage, delivering ±600 kVAR.

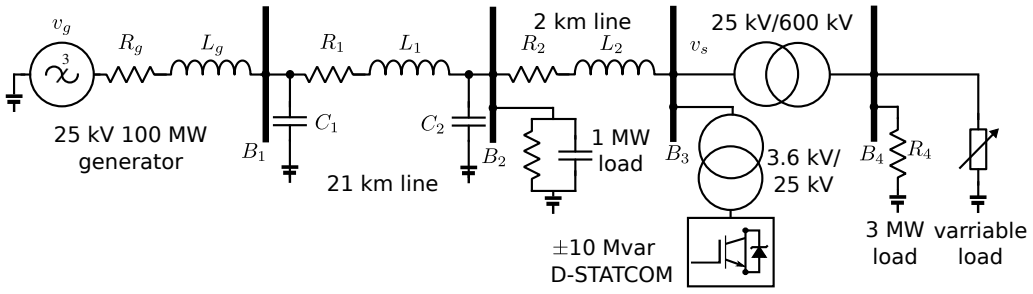


Figure 6.1: One-line diagram of the test bench with the distribution STATCOM.

6.2 CHB Inverters Case Studies

In order to understand the generalization ability of the neural network to deal with the problem of approximating the optimal currents control, three different case studies were considered, i.e., three CHB-STATCOM configurations with $n = 5$, 10, 20 H-bridges, respectively. The inductor was 44 mH, while the capacitors for the three cases were 2600 V - 250 μF, 1300 V - 500 μF and 650 V - 1000 μF, according to the design procedures in [9, 3], which are values that correspond to commercial components.
In order to understand the effectiveness of the NN to approximate the currents controller, as well as changing the number of levels, also the weights of the cost function in (4.1) q_i and p_i, the number of neurons η, and the prediction horizon h were varied.
A set of configurations of n, η,(q_i, p_i) and h were considered and, for each of them, the data were collected from the simulation, by following the procedure described in the previous Section 6.3.
Once the dataset was collected, several NNs were trained in Matlab, changing the activation functions of hidden and output neurons, training algorithms and performance indexes.
It turned out that the NN with the best performance had hyperbolic tangent function (called "tansig" in the Matlab toolbox) in the hidden neurons, linear activation function (called "purelin" in the Matlab toolbox) in the output neurons, while the Bayesian regularization (called "trainbr" in the Matlab toolbox) was used for the learning procedure, considering as performance index the sum squared errors (called "sse" in the Matlab toolbox).
A shallow NN was employed since it was the simplest architecture and it resulted accurate enough for the scope of this work.

6.3 Neural Network Training Procedure

Once the CHB-STATCOM was added in the simulated power system, the standard FCS-MPC was implemented in order to stabilize the CHB-STATCOM currents, DC-link voltages and the grid voltage al PCC. The controllers were tuned by following standard empirical methodologies [7, 8, 6]. Then, the test bench was used in order to excite the system with a sufficiently large set of different inputs, ensuring that the ML algorithms would have sufficient data to effectively learn the optimal control law. Specifically, the test bench was used to simulate changes in the voltage source, which propagates through the grid elements until they reach the node where the STATCOM is connected. Several step changes in the source voltage were simulated in the interval of $[0.8, 1.2]$ pu voltage. In this way, the STATCOM, controlled by FCS-MPC, reacted by providing reactive power in all the range of the rated STATCOM power.
The NN was trained to imitate the currents controller as follows.
The optimization problem for the currents controller, in 2.6, takes as input, at each time step k, the variables $\mathbf{i}^{ref}_{\alpha,\beta}(k)$, $\mathbf{i}_{\alpha,\beta}(k)$, $\mathbf{v}_{s(\alpha,\beta)}(k)$, $\mathbf{S}_{\alpha,\beta}(k-1)$ and computes $\mathbf{S}^{*}_{\alpha,\beta}(k)$. During the simulations, the inputs and output of the optimization problem were collected and stored. Then, the NN was used to map the stored input $\mathbf{i}^{ref}_{\alpha,\beta}(k)$, $\mathbf{i}_{\alpha,\beta}(k)$, $\mathbf{v}_{s(\alpha,\beta)}(k)$, $\mathbf{S}_{\alpha,\beta}(k-1)$ to the computed output $\mathbf{S}^{*}_{\alpha,\beta}(k)$, such that to emulate the optimal current controller.
For each carried out test, the data were collected from a simulation of 8 seconds with 80 step changes. Since the controller sampling time was set to 40 µs, the overall dataset was composed of 200.000 samples. Starting from this dataset, a further dataset was created by perturbing the inputs of the optimization problems (which are $\mathbf{i}^{ref}_{\alpha,\beta}(k)$, $\mathbf{i}_{\alpha,\beta}(k)$, $\mathbf{v}_{s(\alpha,\beta)}(k)$, $\mathbf{S}_{\alpha,\beta}(k-1)$) with random Gaussian noise of ± 20 % of the original value and solving again the control problems for these new data points (hence, computing $\mathbf{S}^{*}_{\alpha,\beta}(k)$). The original dataset and the perturbed one were merged into one dataset: this procedure gives robustness to the learning procedure and improves the inference performance.
Once the dataset was collected, the training dataset was build by collecting the 70% of the overall set, while the validation and the test datasets were 15% of the overall dataset. The three sets (training, validation, and test) were built by randomly collecting the points from the overall set. In this way, every NN learning procedure had a different division of the datasets and computed a slightly different mapping, even if the overall dataset was the same. Hence, from the same dataset, different NNs were trained and it was possible to select the one which provided the best closed loop performance.
This procedure was done to reduce the effect of dataset division on the ML performance. In this work, 10 NNs were trained for the currents controller for each parameter combination, then the NN with best performance was selected for each configuration.
The NNs were trained using machine learning toolbox in Matlab, while the controlled inverter was simulated using the Simscape Electrical library in Matlab/Simulink.

6.4 NN-MPC Performance Evaluation

The performance of the different neural networks was compared considering the mean absolute error (MAE) between current and reference, the switching frequency, the total harmonic distortion (THD) on steady-state, and evaluating the transient response.

6.4.1 Different Numbers of Levels

To study the generalization of the NN approach for different levels, three CHB inverters were considered with $n = 5$, 10, 20 H-bridges per phase.
The weighting coefficients (q_i, p_i) were fixed to $(1, 0.1)$ and $\eta = 8$ hidden neurons were considered with prediction horizon $h = 1$.

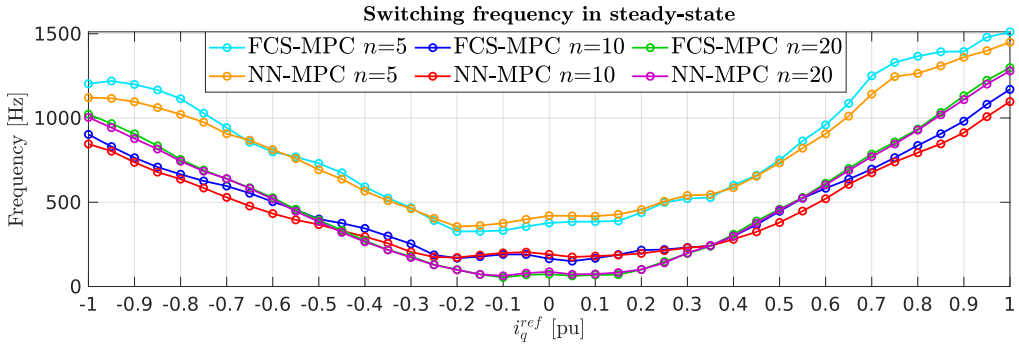


Figure 6.2: Steady-state switching frequency $n = 5$, 10, 20.

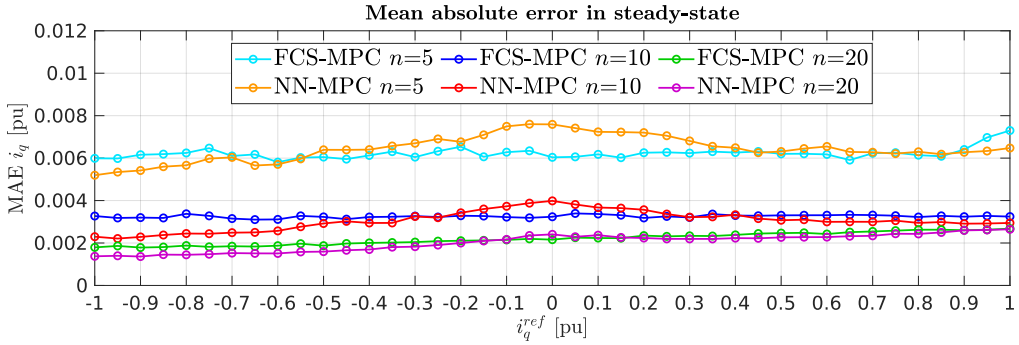


Figure 6.3: Steady-state MAE $n = 5$, 10, 20.

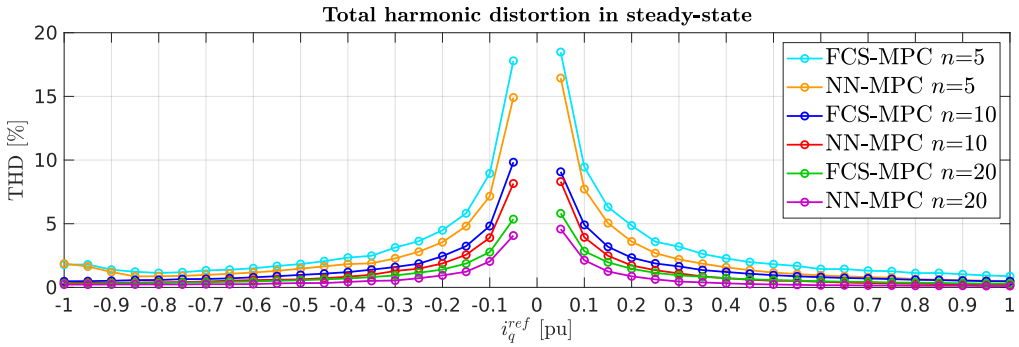


Figure 6.4: Steady-state THD $n = 5$, 10, 20.

Fig. 6.2 shows the switching frequency of the controllers for different steady-state conditions of the quadrature current reference. The frequency is not fixed due to the absence of a modulator and varies according to reference changes.

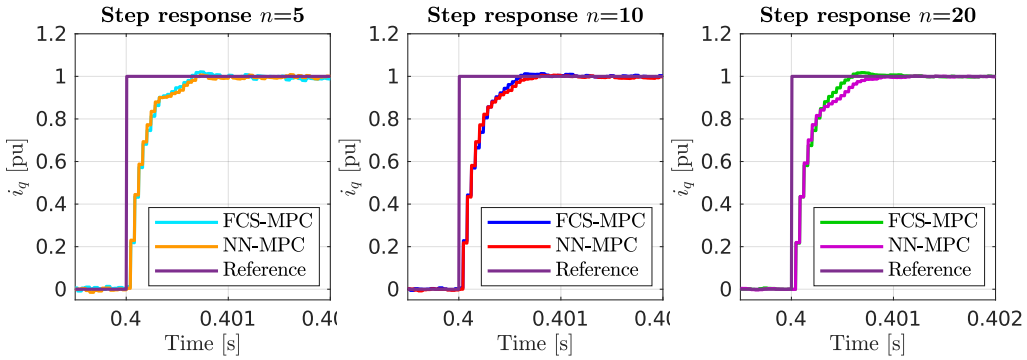


Figure 6.5: Transient state for $n = 5$, 10, 20.

Fig. 6.3 presents the MAE, while Fig. 6.4 shows the THD.
Fig. 6.5 presents the transient state, showing the dynamic response to a $\Delta i_q = 1$ pu step. It turned out that the NN-MPC controller provides a slightly lower frequency and harmonic distortion, compatible reference errors in steady-state and a slightly slower dynamic response compared to FCS-MPC. Moreover, it follows the optimal controller trend in the three considered cases, confirming that the NN approximation does not depends on n and underlying that the NN controllers is a general solution and a promising alternative for real-time implementation irrespective of the number of levels of the converter.
Tab. 6.1 summarized the average performances.

Table 6.1: Average performances for different levels.

	Frequency	MEA	THD	Transient MAE
FCS $N = 5$	809 Hz	0.0062 pu	3.3445	0.0715 pu
NN $N = 5$	783 Hz	0.0064 pu	2.6287	0.0704 pu
FCS $N = 10$	489 Hz	0.0033 pu	1.7212	0.0685 pu
NN $N = 10$	457 Hz	0.0031 pu	1.2918	0.0698 pu
FCS $N = 20$	498 Hz	0.0022 pu	1.0205	0.0682 pu
NN $N = 20$	491 Hz	0.0020 pu	0.6540	0.0725 pu

Further tests were carried out to analyze the system outside the rated conditions to evaluate the performance of the NN controller in a region outside the training dataset.
In particular, a voltage 20 % higher than the nominal value was applied and a current 20 % higher than the rated value was supplied. The grid voltage was 1.4 pu and the reference quadrature current step was 1.2 pu.
Fig. 6.6 shows the step response for the $n = 10$ inverter. A zoom of the α, β currents is shown highlighting the satisfactory dynamic response.
Fig. 6.7 shows the switching frequencies when applying the two controllers. The NN was

able to generalize the control law also for operational conditions not considered in the training dataset. This result is still valid for $n = 5$ and $n = 20$ CHB-STATCOMs.

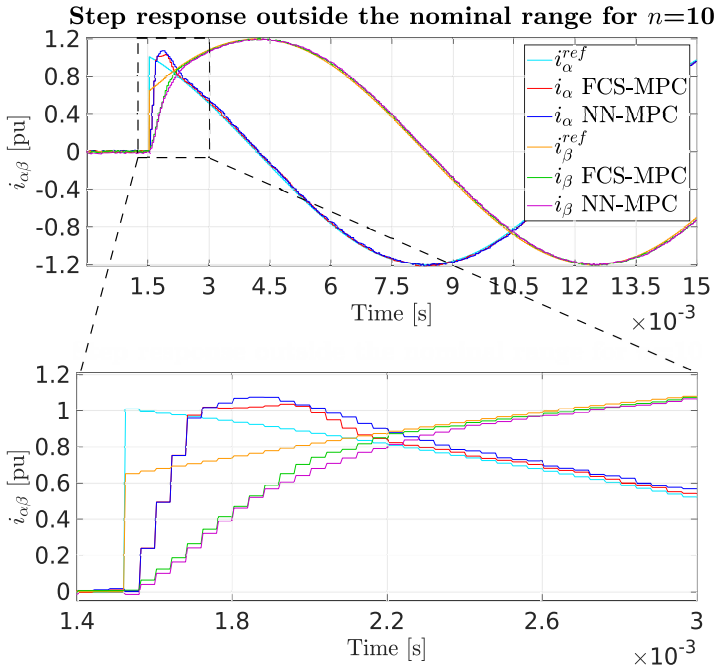


Figure 6.6: Step response outside the nominal range, $n = 10$.

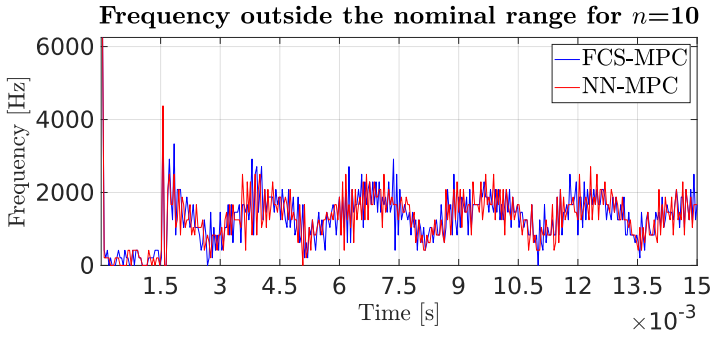


Figure 6.7: Step response outside the nominal range, $n = 10$, frequency.

6.4.2 Different Weighting Coefficients (q_i, p_i)

The weighting factors of the FCS-MPC cost function in (4.1) allow a trade-off between tracking performances and switching losses. By increasing the ratio q_i/p_i the controller

gives higher priority to reference error minimization at the expense of the switching frequency. Once the weighting factors are suitably tuned, it is possible to train the NN that will approximate the desired controller behavior.
In order to evaluate the NN performance under different tuning parameters, tests were carried out by varying the FCS-MPC weighting coefficients (q_i, p_i) between $(1, 0)$, $(1, 0.1)$, $(1, 0.5)$, $(1, 1)$, $(0.5, 1)$.
The following tests refer to the $n = 10$ case, with $\eta = 8$ and $h = 1$.

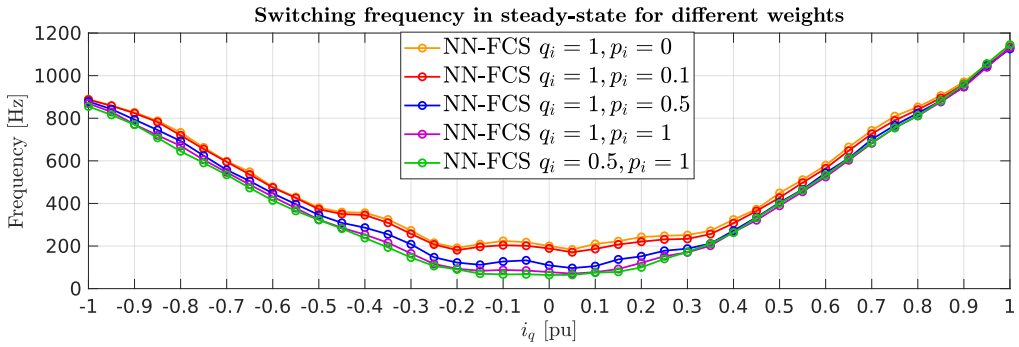


Figure 6.8: Steady-state switching frequency for different weights.

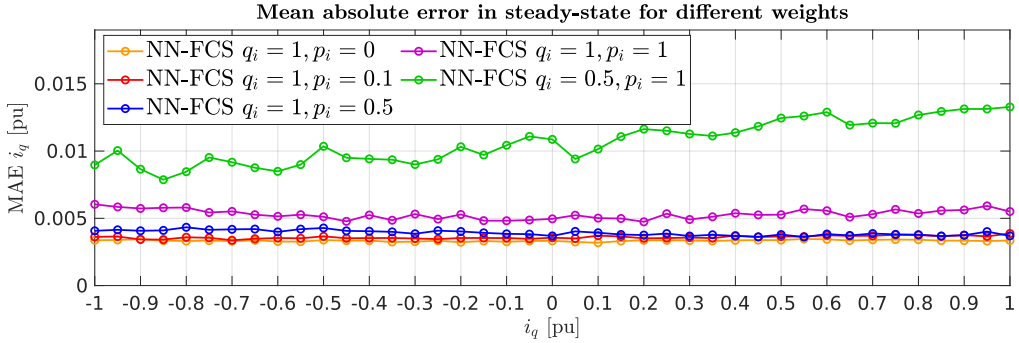


Figure 6.9: Steady-state MAE for different weights.

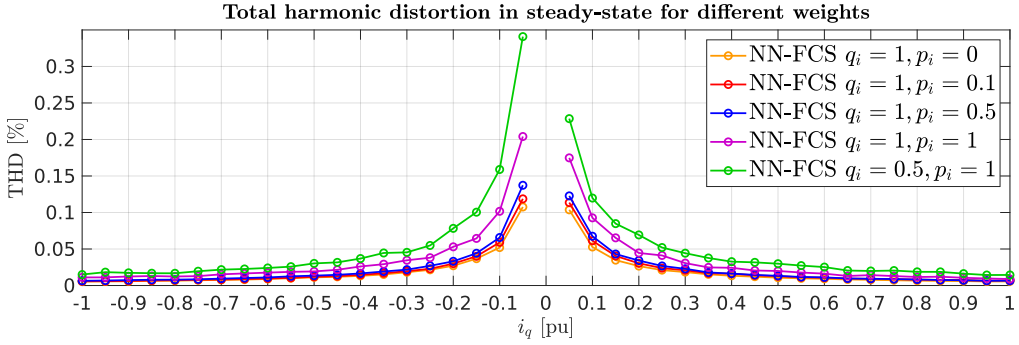


Figure 6.10: Steady-state THD for different weights.

Figs. 6.8, 6.9 and 6.10 show the switching frequencies, the MAE and the THD at steady-state for the different control tunings. By varying the weighting ratio, the performance gradually changes, as happened in the original, which underlines that the NN were correctly approximating the FCS-MPC.
As the ratio increases, the MAE and the THD decrease at the expense of a higher switching frequency. Conversely, as the ratio decreases, the switching frequency becomes lower but MAE and THD increase. The step response in Fig. 6.11 shows that a low ratio also

impacts the transient state, increasing both settling time and overshoot, as evident in the case $(q_i, p_i) = (0.5, 1)$.

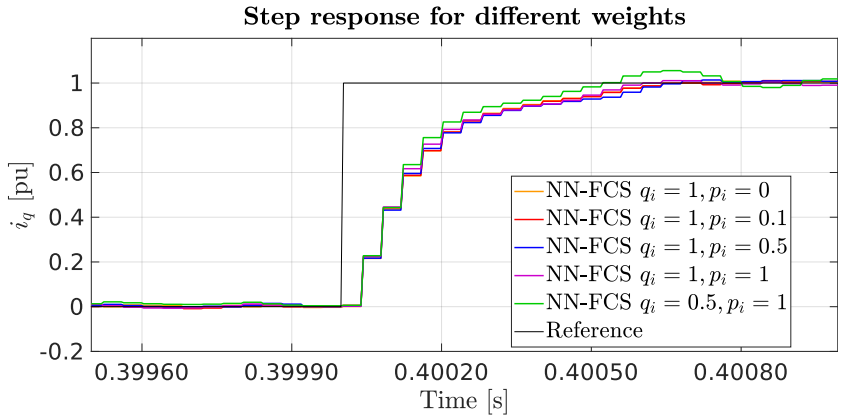


Figure 6.11: Transient operations for different weights.

6.4.3 Different Hidden Neurons η

The trade-off between performance and complexity of the neural network is a key factor for the on-line implementation. In order to find the best trade-off, tests were carried out by training NNs with a different number of hidden neurons.
Starting from the data collected for $n = 10$, $(q_i, p_i) = (1, 0.1)$, $h = 1$ case, different NNs were trained with hidden neurons η equal to 2, 4, 8, 16, 32.
Figs. 6.12, 6.13 and 6.14 show the performance of the different NN architectures, while Fig. 6.15 presents the transient state.
The tests above showed that even a NN with just two hidden neurons was enough to stabilize the system. However, its performance was poor if compared to NNs with more neurons. The performance analysis suggested that $\eta = 4$ was sufficient to guarantee satisfactory results. At the same time, $\eta = 8$ was the best option since it generated the lowest switching frequency and its performance was compatible in term of THD and MAE with respect to $\eta = 4, 16, 32$.

6.4.4 Different Prediction Horizons h

By increasing the prediction horizon of the FCS-MPC, it is possible to improve the overall controller's performance [1]. In order to test the ability of the NN approach to learn multiple horizons, FCS-MPC tests were carried out by changing the prediction horizon. Since the increase of the control horizon led to a dramatic increment in computations, horizons larger than three were not tested.

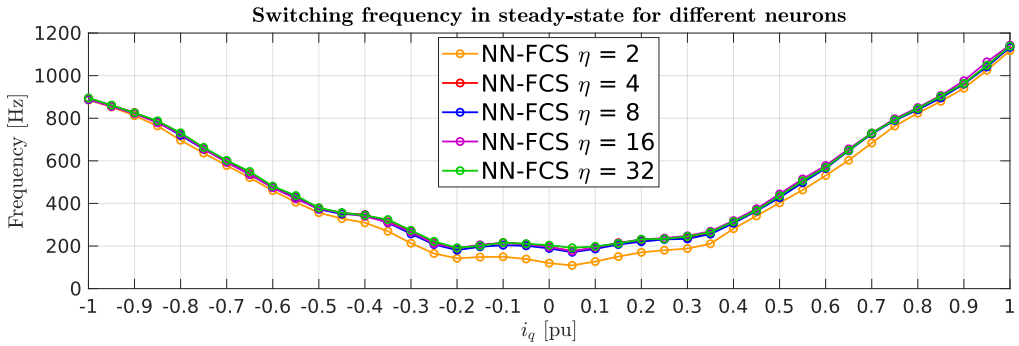


Figure 6.12: Steady-state switching frequency for different hidden neurons.

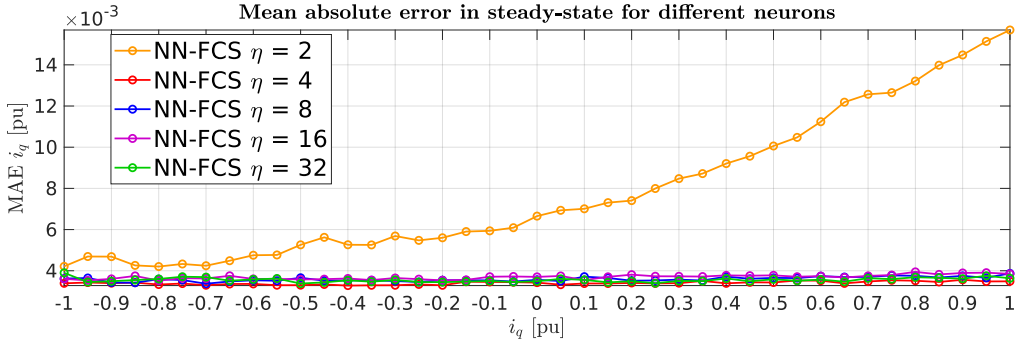


Figure 6.13: Steady-state MAE for different hidden neurons.

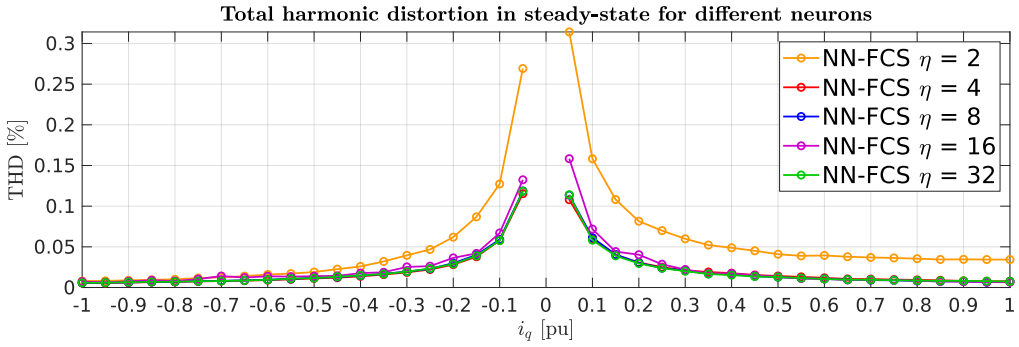


Figure 6.14: Steady-state THD for different hidden neurons.

It turned out that horizon $h = 2$ led to a slightly better performance with respect to the case $h = 1$, but no significant improvements were found by changing h from 2 to 3. For this reason, NNs approximations were tested just for $h = 1$ and 2.
Figs. 6.16, 6.17 and 6.18 show the steady-state switching frequency, MAE and THD for the case $n = 5$, $(q_i, p_i) = (1, 0.5)$, approximated with $\eta = 8$ NNs for horizon $h = 1, 2$.
The NN-MPC tended to have a slightly lower switching frequency and a slightly higher THD when compared with the original FCS-MPC.

Fig. 6.19 presents the transient response of the controllers, showing a slightly lower settling time of the NN approximations compared to the FCS-MPC.
Tab. 6.2 summarizes the average performance for horizons $h = 1, 2$.

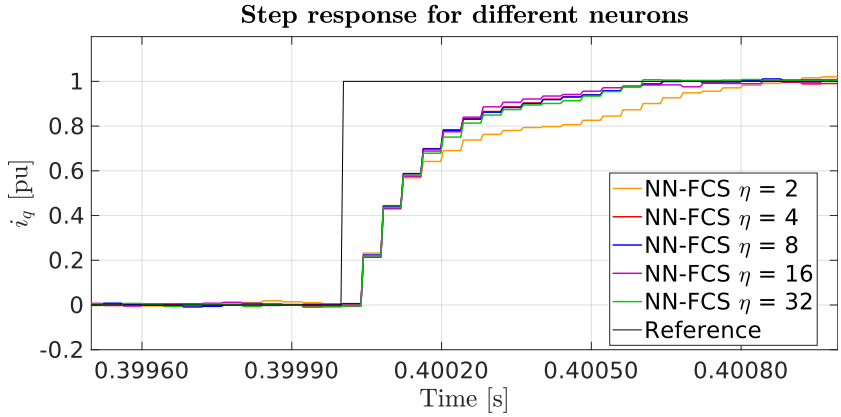


Figure 6.15: Transient operations for different hidden neurons.

Table 6.2: Average performance for different prediction horizons.

	Frequency	MAE	THD	Transient MAE
FCS h=1	760.58 Hz	0.0115 pu	0.0529	0.1042 pu
NN h=1	745.66 Hz	0.012 pu	0.0542	0.1000 pu
FCS h=2	774.75 Hz	0.0096 pu	0.0456	0.1030 pu
NN h=2	766.63 Hz	0.0098 pu	0.0475	0.0921 pu

It can be remarked that, in order to predict the currents of a CHB-STATCOM, the grid voltage is needed, as shown in the dynamical model in (2.5). The grid voltage is sensed at each time step k but, in order to make prediction of future horizon larger than 1, the grid voltages at time $k+1, ..., k+h$ have to be predicted.
The estimation of the grid voltage at time $k+1$ is performed by considering the grid voltage vector at time k and by rotating it by ωT_s, where ω is the nominal voltage frequency, i.e., 50 Hz in this study. This way, we are assuming that the grid voltage has the same magnitude and shift in the next instant, which makes simple the computation of future predictions.
However, making assumptions on the grid voltage leads to inaccuracy, especially in STATCOMs applications, where the inverter must react to quick voltage variations. For this reason, for CHB-STATCOMs, a prediction horizon equal to $h = 3$ leads to worst performance with respect to $h = 2$ and, in general, there is not a valuable improvement by considering prediction horizons greater than $h = 1$.

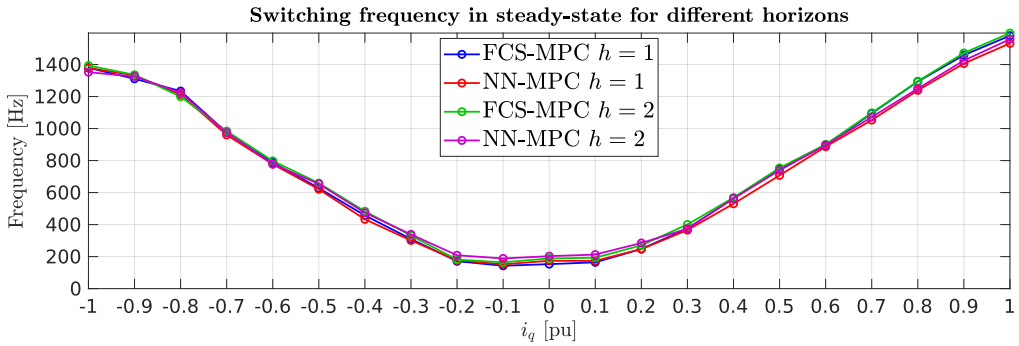


Figure 6.16: Steady-state switching frequency for different prediction horizons.

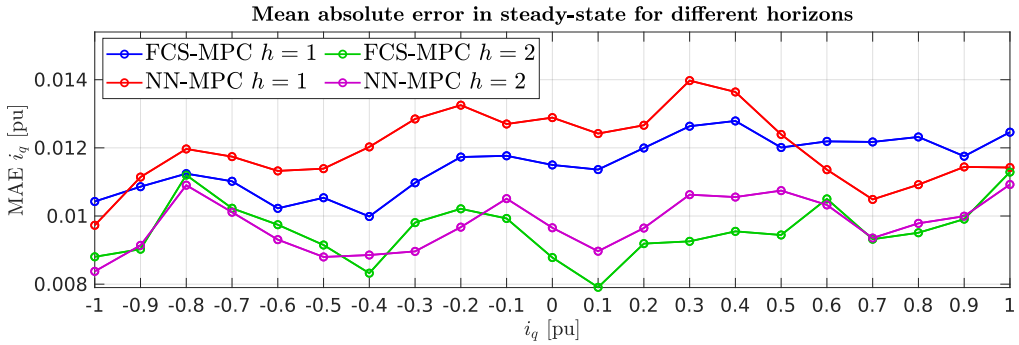


Figure 6.17: Steady-state MAE for different prediction horizons.

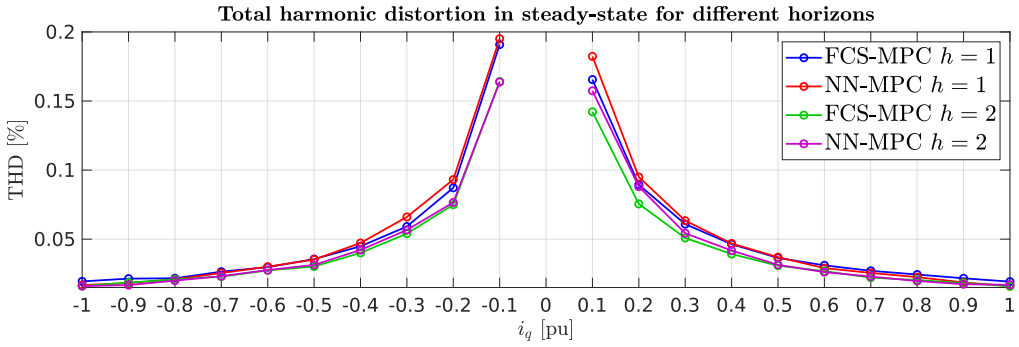


Figure 6.18: Steady-state THD for different prediction horizons.

6.5 Conclusions

This Chapter shows the performance analysis of the neural network controller method described in Chapter 4 in Matlab/Simulink environment and the comparison with the

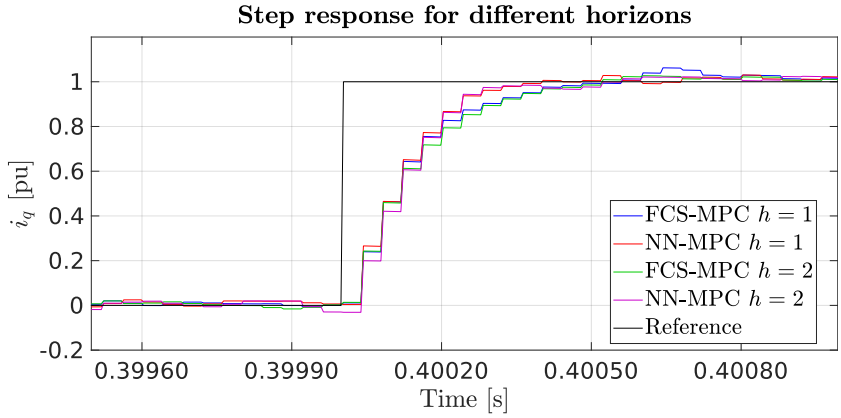


Figure 6.19: Transient operations, different prediction horizons.

standard FCS-MPC.
The generalization of the NN for approximating the FCS-MPC was deeply analyzed by testing the methods for different number of levels of the converter, weighting factors in the MPC cost function, number of hidden neurons in the NN, number of prediction horizons. This study demonstrates that the NN-MPC closely follows the optimal control low behavior and it also proves that it is general solution irrespective of the different parameters of the controller and the converter.

References

[1] T. Geyer and D. E. Quevedo, "Performance of multistep finite control set model predictive control for power electronics," *IEEE Transactions on power electronics*, vol. 30, no. 3, pp. 1633–1644, 2014.

[2] F. Simonetti, G. D. Girolamo, A. D'Innocenzo, and C. Cecati, "A neural network approach for efficient finite control set mpc of cascaded h-bridge statcom," in *IECON 2021 – 47th Annual Conference of the IEEE Industrial Electronics Society*, pp. 1–6, 2021.

[3] A. M. Saif, C. Buccella, V. Patel, M. Tinari, and C. Cecati, "Design and cost analysis for statcom in low and medium voltage systems," in *IECON 2018 - 44th Annual Conference of the IEEE Industrial Electronics Society*, pp. 3938–3943, 2018.

[4] P. Giroux, G. Sybille, and H. Le-Huy, "Modeling and simulation of a distribution statcom using simulink's power system blockset," in *IECON'01. 27th Annual Conference of the IEEE Industrial Electronics Society (Cat. No. 37243)*, vol. 2, pp. 990–994, IEEE, 2001.

[5] F. Simonetti, A. D'Innocenzo, and C. Cecati, "Neural network model predictive control for chb converters with fpga implementation," *IEEE Transactions on Industrial Informatics*, pp. 1–12, 2023.

[6] 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.

[7] P. Cortés, S. Kouro, B. La Rocca, R. Vargas, J. Rodriguez, J. I. Leon, S. Vazquez, and L. G. Franquelo, "Guidelines for weighting factors design in model predictive control of power converters and drives," in *IEEE ICIT 2009*, pp. 1–7, 2009.

[8] D. Shetty and N. Prabhu, "Ziegler nichols method based robust reactive current controller for statcom," *Energy Procedia*, vol. 117, pp. 543–550, 2017.

[9] P. Hu, J. M. Guerrero, and Z. He, "Design and analysis of a transformerless statcom based on hybrid cascaded multilevel converter," *International Journal of Electrical Power & Energy Systems*, vol. 104, pp. 694–704, 2019.

[10] F. Simonetti, G. D. Di Girolamo, A. D'Innocenzo, and C. Cecati, "Machine learning for model predictive control of cascaded h-bridge inverters," in *2022 IEEE 21st Mediterranean Electrotechnical Conference (MELECON)*, pp. 1241–1246, 2022.

Chapter 7

Hardware in the Loop Validation of the Neural Network Approach

This Chapter presents the description at register transfer level (RTL) of the experimental implementation of the NN-MPC on FPGA platform [4], it evaluates the clock cycles needed for the algorithms execution and it discusses the actual time spent in computations on an Intel Cyclone V FPGA (5CSEBA6U23I7) on a Terasic DE-10 Nano board.
This Chapter presents the analysis of the impact of the computation delay on the controller performance via hardware in the loop (HIL) simulation by employing the FPGA, where the controller was implemented, and the system modeled on Matlab/Simulink environment.
The HIL tests were carried out for $n = 5$, 10, 20 cases, the weighting parameters were fixed to $(q_i, p_i) = (1, 0.1)$. The number of hidden neurons was $\eta = 8$, which resulted to be the best trade-off between accuracy and cost complexity for the discussed implementation. The prediction horizon was $h = 1$ to allow a fair comparison with the FCS-MPC, since the classical solution is impractical for larger horizons with large number of levels [3].

7.1 Register Transfer Level Implementation

This Section discusses the register transfer level implementation, presenting the algorithmic state machines (ASMs) that describe the sequential steps to be performed by the control units and showing the combinatorial logic realizing the data path of the designed digital hardware [4].
All the ASMs and digital circuits drawn in this thesis have a univocal translation in hardware, and were coded in VHDL (very high speed integrated circuits hardware description language) in the development environment Intel Quartus Prime software.
In the following ASMs pictures, the notation $a <= b$ indicates a signal assignment, i.e., the signal called a is physically wired to the signal called b; while $a \leftarrow b$ means that the signal b is the input of the register named a, implying that the register a is updated with the value of the signal b when the next rising clock edge arises.

7.1.1 Neural Network Implementation on FPGA

The NN trained with the machine learning Matlab toolbox performs the following computations :

- the pre-processing calculation, which normalizes the input of the NN by performing a subtraction and a multiplication for each input neuron;

- the computation of the the first layer, which involves the matrix-vector multiplication between the 8×1 input vector

$$\mathbf{X} = \left[\mathbf{i}_{\alpha,\beta}^{ref}(k), \mathbf{i}_{\alpha,\beta}(k), \mathbf{v}_{s(\alpha,\beta)}(k), \mathbf{S}_{\alpha,\beta}(k-1)\right]^T$$

  and the $8 \times \eta = 8 \times 8$ weight matrix $\mathbf{W}^1$, followed by the computation of $\eta = 8$ activation functions;

- the computation of the the second layer, which involves the matrix-vector multiplication between the 8×1 output vector of the first layer and the weight matrix $\mathbf{W}^2$ (we do not have to compute an activation function in the second layer for the chosen architecture);

- the post-processing, which normalizes the output of the NN by performing a subtraction and a multiplication for each output neuron.

Fig. 7.1 shows the designed ASM that describes the sequential steps performed by the control to drive the computations of the $\eta = 8$ neural network.
In the ASM, the signals ACC_jA and ACC_jB are the input signals of the j-th combinatorial net performing the sum operation, while ACC_jO is its output signal.
Similarly, MULT_jA and MULT_jB are the input signals of the j-th combinatorial net performing the multiplication, while MULT_jO is its output signal.
A number of 8 accumulator circuits and 8 multiplication circuits were instantiated to perform the computations of the NN, by fully exploiting the parallelism capability of the $\eta = 8$ NN.
The pre-processing and post-processing calculations computed by the NN generated by Matlab were performed in the corresponding states ($\text{PRE-PROSESSING}_{1,2,3}$ and $\text{POST-PROCESSING}_{1,2,3}$ in Fig. 7.1). The computations of the NN layer were implemented by using the 8 adders and 8 multipliers opportunely multiplexed to create the pipeline structure in Fig. 7.2 to speed-up the algorithm: the red lines in the figure underline the different stages of the pipeline.
The activation function of the hidden layer was computed by using a look-up table (LUT in the figure).
The designed digital hardware performed the following computations:

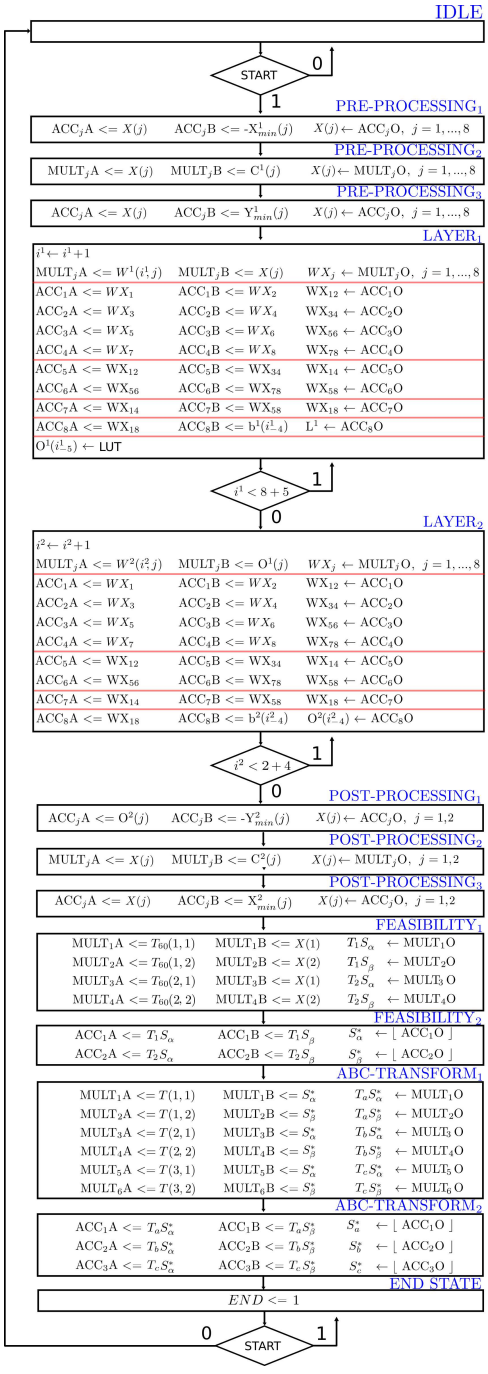


Figure 7.1: Neural network algorithmic state machine.

- Pipeline stage 1: The j-th multiplier performed a multiplication between the ele-

ment in the j-th column and i-th row of the weight matrix $\mathbf{W}^l$ of layer $l \in \{1, 2\}$ (which is the value $\mathbf{W}^l_{ij}$) and the element in the j-th row of the input vector $\mathbf{X}$ (which is the value $\mathbf{X}_j$). The 8 multipliers worked in parallel to compute, in one clock cycle, the 8 individual products that had to be summed up to obtain the matrix-vector multiplication of the i-th row of $\mathbf{W}^l$ and $\mathbf{X}$, i.e., the product $\mathbf{W}^l_i \cdot \mathbf{X}$.

- Pipeline stages 2, 3, 4, 5: The temporary results of the multiplications were summed in pairs in order to obtain the product $\mathbf{W}^l_i \cdot \mathbf{X}$.
- Pipeline stage 6: The result of the linear combination between the input $\mathbf{X}$ and the $\mathbf{W}^l_i$ was fed in the input of the i-th neuron of the l layer. In order to compute the output of the i-th neuron, the input of the i-th neuron was fed into the LUT that implements the activation function.

The designed control unit, described in the ASM in Fig. 7.1, started from the 1-st row of the fist layer weight matrix $\mathbf{W}^1$ and proceeded by feeding the pipeline with a new row of $\mathbf{W}^1$every clock cycle, until it reached the 8-th row. It waited 8 plus 5 clock cycles, as in Fig. 7.1, in order to allow the computations of all the pipeline stages of the last row. After computing the outputs of the 8 hidden neurons, the control unit employed the same pipeline by feeding it with the computed output of the first NN layer.
The weight matrix of the second layer $\mathbf{W}^2$ was employed and the LUT had not to be computed for the NN output layer. Hence, the control unit waited for 2 plus 4 clock cycles to allow the computations of the output of the second layer, which was post-processed to obtain the result of the overall NN.

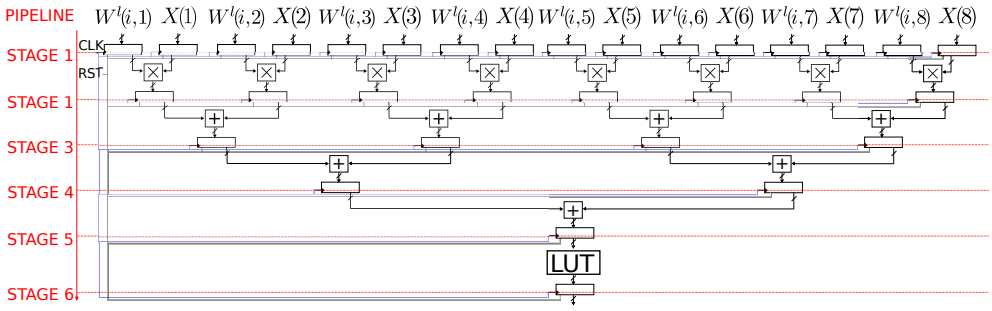


Figure 7.2: Pipeline structure implementing the computations of the NN layer.

The NN output was transformed into $\mathbf{S}^{60}_{\alpha,\beta}(k)$ in order to compute the feasible switching vector and the $\mathbf{S}^*_{\alpha,\beta}(k)$ was obtained as explained in Subsection 4.2.1 (FEASIBILITY$_{1,2}$ states in the ASM picture, which multiplied the NN output by the matrix $\mathbf{T}_{60}$ and rounded the result).
It was, then, transformed into the a, b, c coordinate considering null zero-sequence obtaining $\mathbf{S}^0_{a,b,c}(k)$ (ABC-TRANSFORM$_{1,2}$ states in the ASM picture).

7.1.2 Clusters Voltages Balance Implementation on FPGA

The clusters voltages balancing problem was solved by explicitly computing the cost for all possible a, b, c redundant vectors, as described in the Algorithm 3.2.

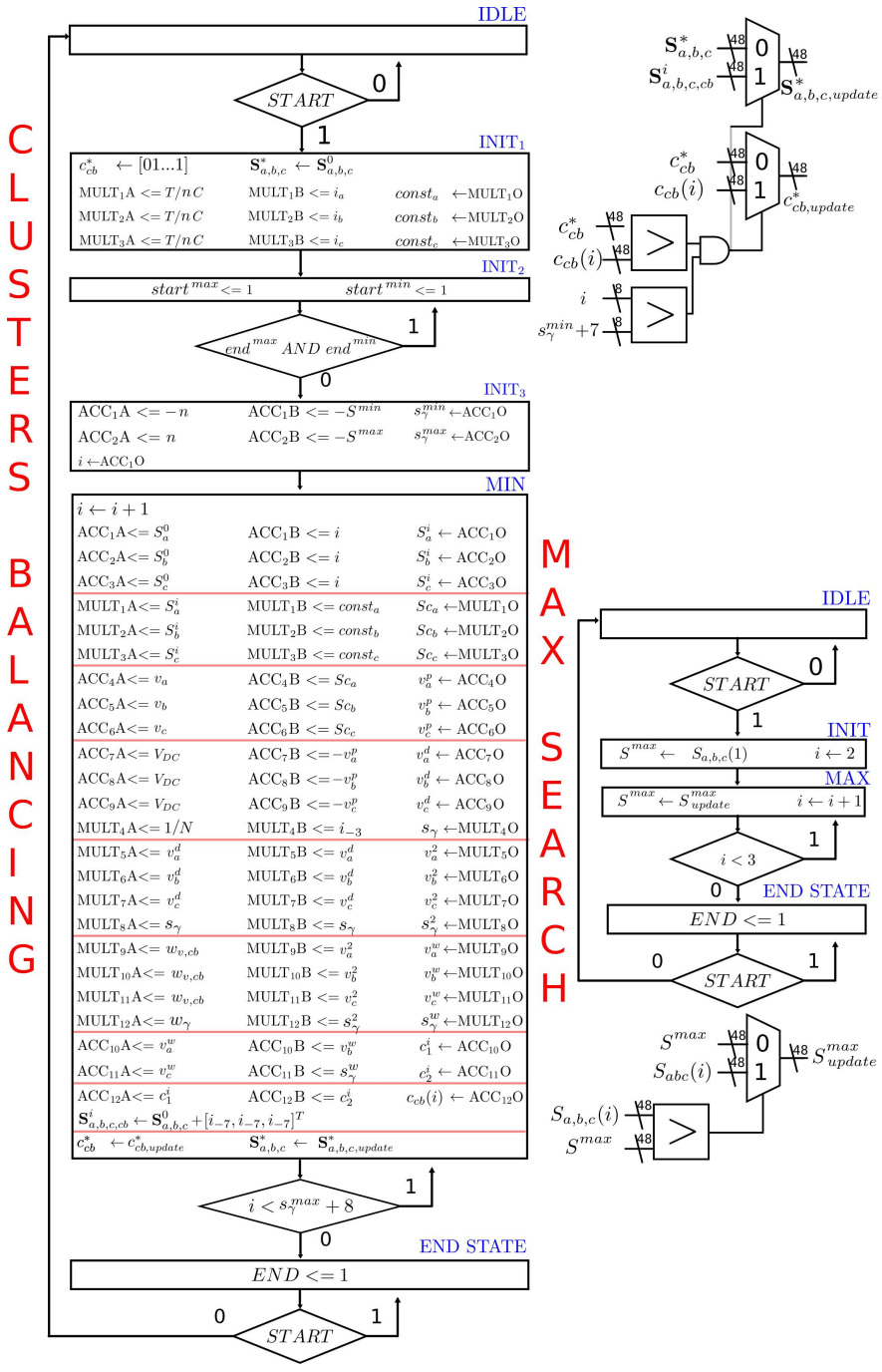


Figure 7.3: Clusters voltages balance algorithmic state machine.

Fig. 7.3 shows the ASM describing the control unit performing the clusters voltages bal-

ance. In the initialization phase ($\text{INIT}_{1,2,3}$ states in the ASM picture) the cost c^*_{cb} was initialized to be the maximum stored value, while the optimum value $\mathbf{S}^*_{a,b,c}$ was initialized to be $\mathbf{S}^0_{a,b,c}$. The $\mathbf{const}_{a,\,b,\,c} = [const_a, const_b, const_c]^T$ values included the term that were constant during the operations of the same sampling interval (INIT_1).
The minimum S^{min} and maximum S^{max} values among S^0_a, S^0_b, S^0_c were evaluated in INIT_2 state, by providing the start operations to the two ASMs performing the minimum and maximum search among the element of the $\mathbf{S}^0_{a,b,c}$ vector.
Fig. 7.3 also shows the ASM for the maximum finding (MAX SEARCH in the picture) and the combinatorial net used for selecting the maximum value between two elements. The minimum search is omitted since it has the same structure, with the only difference that the combinatorial net selects the minimum value between the two elements.
The S^{min} and S^{max} values were used to compute the maximum and minimum zero-sequence admissible voltages, i.e., s^{min}_γ and s^{max}_γ (INIT_3 state in the picture).
Then, the costs were computed for all the admissible values, which are $\mathbf{S}^i_{a,b,c}(k) = \mathbf{S}^0_{a,b,c}(k) + [i,i,i]^T$, $s^{min}_\gamma \leq i \leq s^{max}_\gamma$, by employing a pipeline structure with 12 accumulators and multipliers (the stages of the pipeline are delimited by the red lines in the MIN state in the CLUSTER BALANCING ASM). The control unit scanned all the values between s^{min}_γ and s^{max}_γ and it waited 8 more clock cycles to empty the pipeline. The minimum cost c^*_{cb} and the optimal vector $\mathbf{S}^*_{a,b,c}$ were iteratively updated by employing the comparator circuits in the top-right corner of Fig. 7.3.

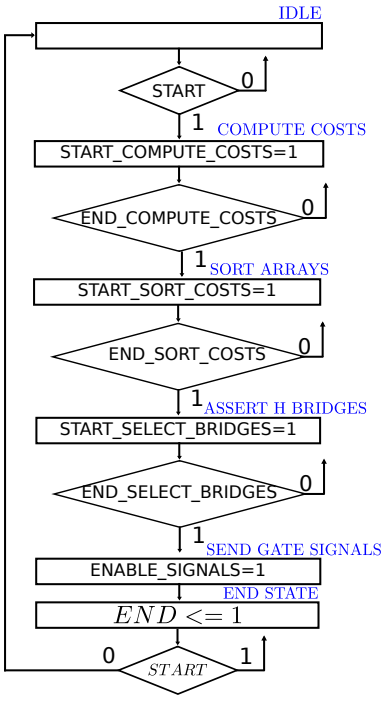


Figure 7.4: Individual voltages balance overall algorithmic state machine.

7.1.3 Individual Voltages Balance Implementation on FPGA

The individual voltages balancing problem was solved for each phase $p \in \{a, b, c\}$ starting from the optimal value S_p^* [1].
As discussed in Subsection 3.2.1, it proceeded as follows: the cost for each H-bridge was computed, then the costs array was sorted in increasing order. The first $\left\|\mathbf{S}^*_{a,b,c}\right\|$ H-bridges were selected to be 1 or -1 according to the sign of $\mathbf{S}^*_{a,b,c}$, while the others were set to 0 and the computed switching variables $s^*_{pi}(k)$ were translated to the gate signals.
Fig. 7.4 shows the overall ASM describing the control unit that scheduled the four different steps of the individual voltage balance. The ASMs implementing the different parts of the algorithm are described as follows.

- The ASM in Fig. 7.5 is related to the computation of the costs array (COMPUTE COSTS in the picture). In INIT_1 and INIT_2 the variables that remain constant during the overall computation (c_1 and c_2 in Fig. 7.5) were calculated depending on the sign of S_p^*, which was computed by the clusters voltages problem. If the most significant bit (MSB) of S_p^* was 0, i.e., S_p^* was positive, the variables $s_{pi}\,(k)$ could not be negative and $\overline{s}$ was set to 1.

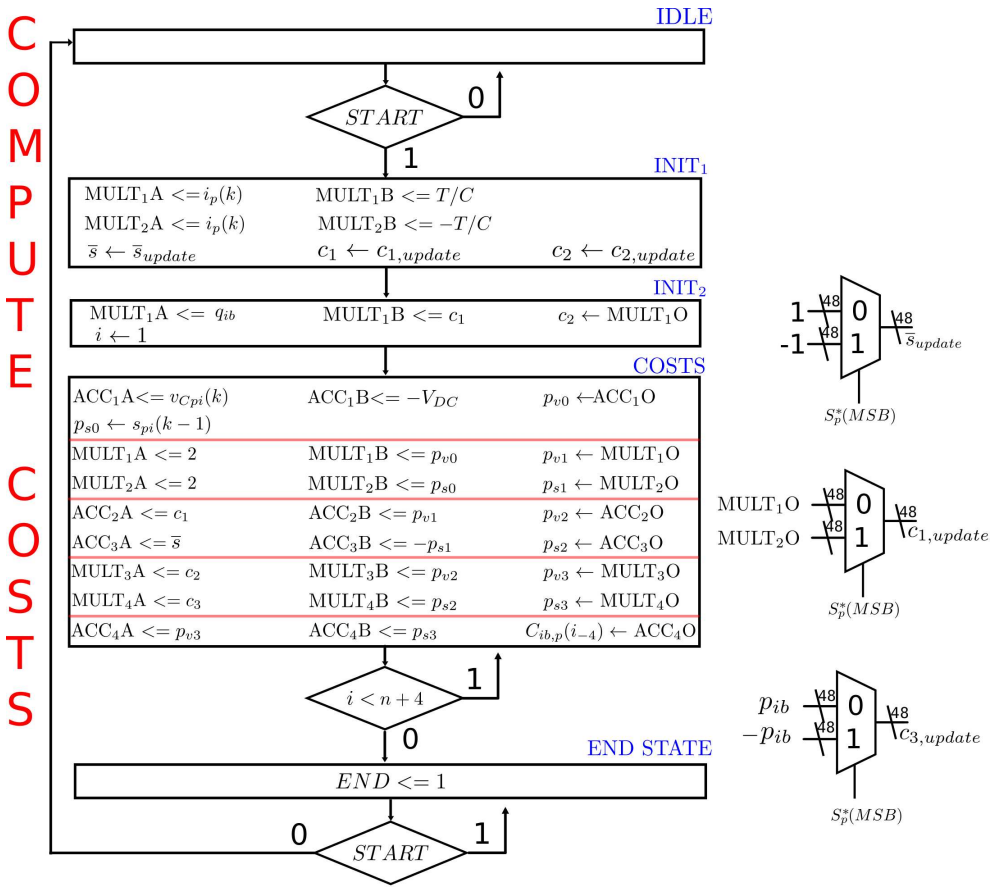


Figure 7.5: Individual voltages balance algorithmic state machine: compute costs.

Otherwise, if the MSB of S_p^* was 1, i.e., S_p^* was negative, the variables $s_{pi}\,(k)$ could not be positive and $\overline{s}$ was set to -1.Then, the n costs were evaluated and stored by employing a five-level pipeline and utilizing 4 adders and 4 multipliers for each phase $p \in \{a, b, c\}$, which made it possible to use the 12 total adders and multipliers, previously used by the cluster balance, for computing the costs $\mathbf{C}_{ib,p}$ of the three phases p in parallel.

- Once the cost array $\mathbf{C}_{ib,p}$ was computed, the array had to be sorted in ascending order to obtain the array $\mathbf{C}_{ib,p}^*$.
  The ASM in Fig. 7.6 (SORT COSTS in the picture) implements the sorting algorithm. For simplicity, a simple selection sort algorithm was implemented. In the INIT state, the ASM initialized the sorted cost array $\mathbf{C}_{ib,p}^*$ equal to the original array $\mathbf{C}_{ib,p}$ and initialized the costs indexes array $\mathbf{I}_p$. Then, it iteratively performed a minimum search until the array was fully sorted in ascending order.

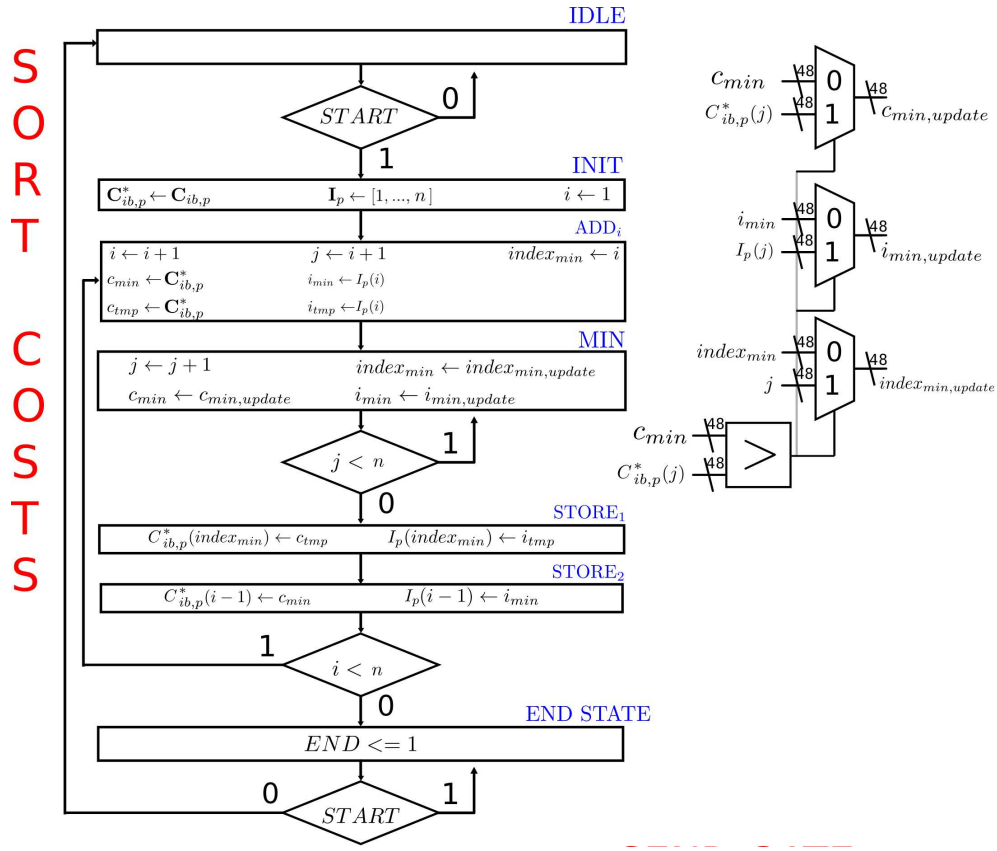


Figure 7.6: Individual voltages balance algorithmic state machine: sort costs.

- The ASM in Fig. 7.7 (SELECT BRIDGES in the picture) scanned the $\mathbf{I}_p$ array. If the i-th element of $\mathbf{I}_p$ was less or equal than the number of H-bridges to assert $\|S_p(k)\|$, the value $s_{pi}(k)$ was set equal to $\overline{s}$, otherwise it was kept to 0, as discussed

in Subsection 3.2.1.
Finally, the switching variable $s_{pi}(k)$ was translated into the switching signals for the four switches of the i-th cell of phase p by the combinatorial functions at the bottom-right corner of Fig. 7.7.

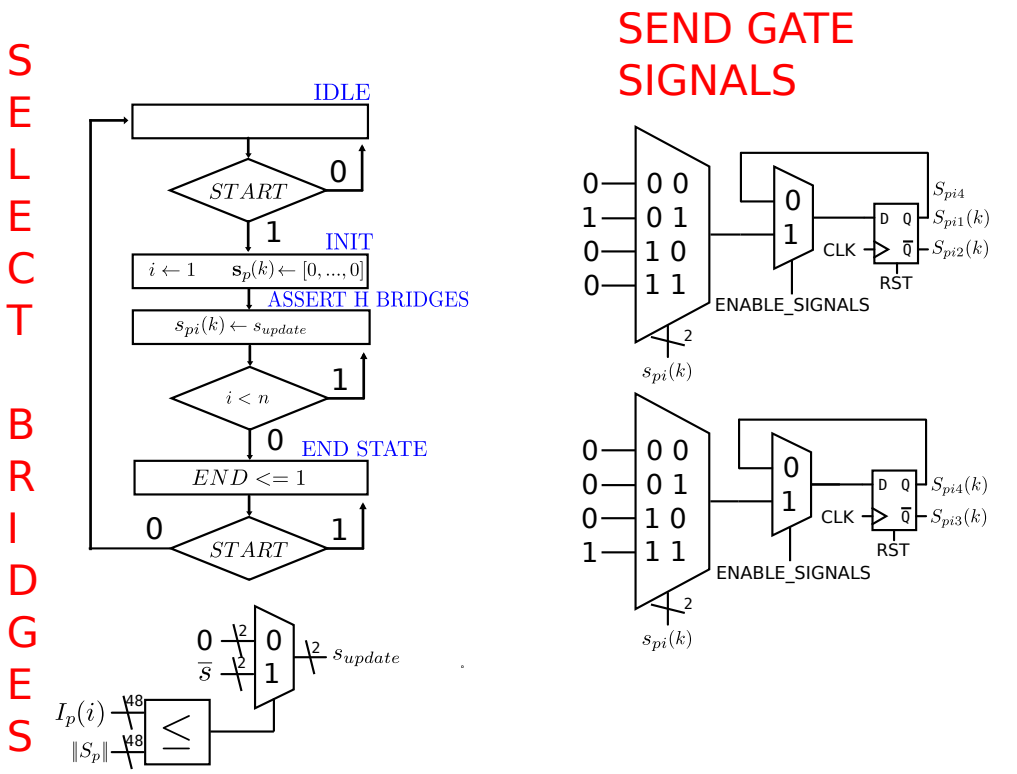


Figure 7.7: Individual voltages balance algorithmic state machine: select H-bridges.

In addition to the neural network controller, the standard FCS-MPC currents controller was also implemented on the FPGA as described in Section 2.4 by exploiting an exhaustive search among all the possible switching combinations in α, β. Since the ASM performing the exhaustive search is straightforward and since it follows the same structure of the clusters balance ASM in Fig. 7.3 (which also performs an exhaustive search among all switching vectors in $\mathbf{V}_{a,b,c}$), the picture of the ASM describing the FCS-MPC current algorithm is not included in the thesis.
To make a fair comparison, the costs computations were realized by using the same number of adders and multipliers used for the NN currents controller, multiplexed in an eight-level pipeline and the computed optimal vector $\mathbf{S}^*_{\alpha,\beta}$ was, finally, transformed by setting the homopolar component to 0, obtaining $\mathbf{S}^0_{a,b,c}$.
Considering the steps needed by the exhaustive search, the pipeline, the initialization and the transformation, the total number of clock cycles needed for the currents control loop were $12n^2 + 6n + 14$.

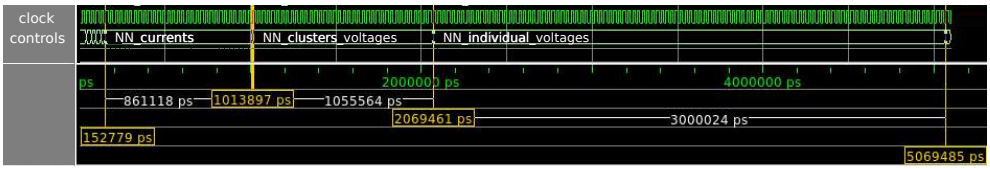


Figure 7.8: NN-MPC simulation in ModelSim-Intel for $n = 10$.

7.2 Hardware in the Loop Results

The two algorithms were implemented on a Terasic DE-10 Nano board equipped with an FPGA SoC Intel Cyclone V (5CSEBA6U23I7). Each control algorithm was tested via hardware in the loop setup, where the CHB-STATCOM was simulated in Simulink environment and the algorithm computations were implemented on the FPGA.

The FPGA in the loop app in Simulink (called "FIL") was used to generate the VHDL files to allow the UART (universal asynchronous receiver-transmitter) communication between the FPGA and Matlab/Simulink and to create a Quartus project that integrated them with the coded VHDL files describing the designed hardware architectures. At each time step of the Simulink simulation, one clock cycle pulse was sent to the FPGA and the ASMs performed one elementary sequential step.

With the described implementation, which employed pipelines, registers sized to 48 bits and the sharing of the arithmetic resources, the Quartus TimeAnalyzer tool suggested to use a clock frequency lower than 36 MHz to ensure the correct execution of the computations. Hence, the simulation step was set equal to the clock time interval required by the FPGA to correctly performs the operations, which was set to 30ns (to ensure a clock frequency less than 36MHz).

The clock cycles needed for the two algorithms and the execution times for the different values of n are reported in Table 7.1.

Fig. 7.8 shows the time spent in the overall NN-MPC computation in the simulation environment ModelSim-Intel for an $n = 10$ CHB-STATCOM.

Table 7.1: Computational cost and execution times of FCS-MPC and NN-MPC.

	clock cycles	$n = 5$	$n = 10$	$n = 20$
Current FCS	$12n^2 + 6n + 14$	9.58 µs	35.42 µs	137.08 µs
Current NN	31	0.86 µs	0.86 µs	0.86 µs
Clusters	$2n + 19$	0.80 µs	1.08 µs	1.64 µs
Voltages	$\frac{1}{2}n^2 + \frac{9}{2}n + 14$	1.36 µs	3.03 µs	8.44 µs
FCS-MPC	$\frac{25}{2}n^2 + \frac{25}{2}n + 39$	11.75 µs	39.52 µs	147.17 µs
NN-MPC	$\frac{1}{2}n^2 + \frac{9}{2}n + 45$	3.03 µs	4.97 µs	10.94 µs

It can be seen that the execution times of the neural networks were the same, since the same NN architecture can be used irrespective of the number of levels, as underlined in the analysis in Subsection 6.4.1. It turned out that the NN-MPC had a much shorter execution time with respect to the standard FCS-MPC.

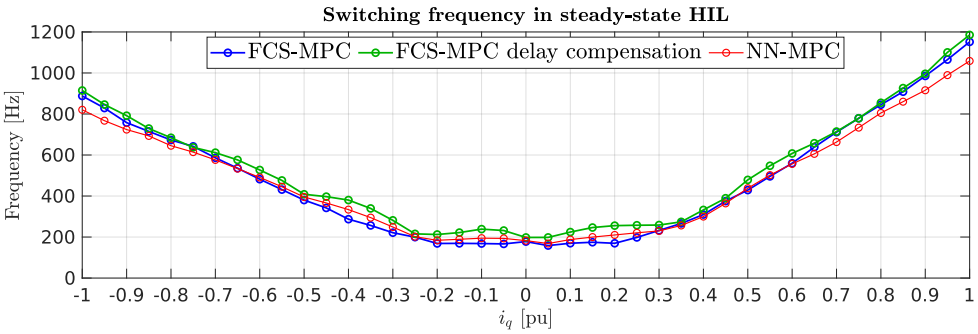


Figure 7.9: Steady-state switching frequency in HIL for $n = 10$.

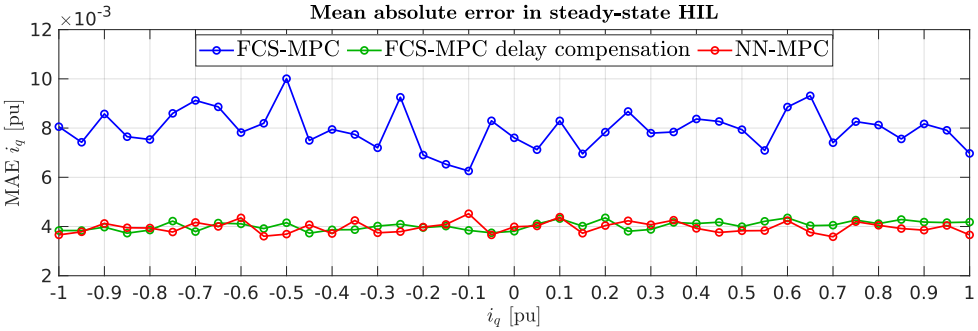


Figure 7.10: Steady-state MAE in HIL for $n = 10$.

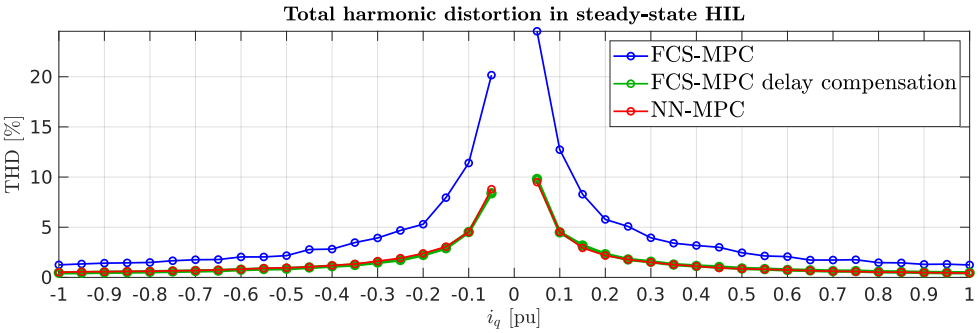


Figure 7.11: Steady-state THD in HIL for $n = 10$.

For the three cases $n = 5$, 10, 20 ,the results were the following.

- For $n = 5$, FCS-MPC needed 11.75 µs while NN-MPC took 3.03 µs. Thus, the classic control is implementable in real-time and it is possible to compute the input within the 40 µs sampling interval. By the way, NN-MPC took an execution time about four times smaller.

- For $n = 10$, FCS-MPC needed 39.52 µs, which is close to the overall sampling interval and it forces us to apply the control input at the next sampling interval. The performance of the standard FCS-MPC is degraded due to the one sampling interval

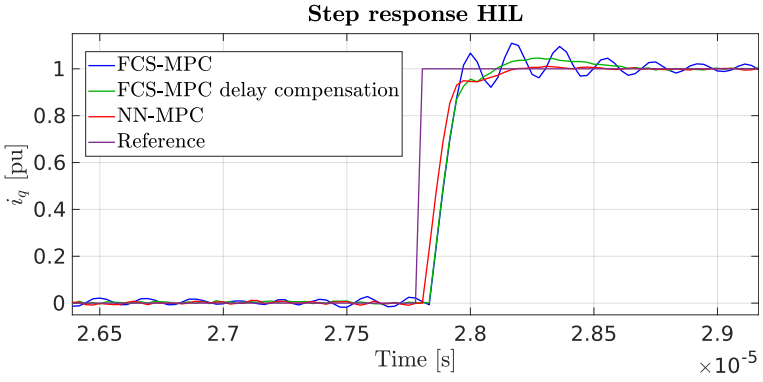


Figure 7.12: Step response in HIL for $n = 10$.

delay. Because of this, it is a common solution to use a delay compensation strategy, as in [2]. It consists of predicting the two-steps ahead prediction of the currents $\mathbf{i}_{\alpha,\beta}(k+2)$ in order to compute the input $\mathbf{S}_{\alpha,\beta}(k+1)$ to apply at the beginning of the next sampling interval, taking into account the one sampling interval delay, as will be discussed in 5.
By using NN-MPC, the control loop was completed in 4.97 µs, which was one order of magnitude less than the conventional FCS-MPC and it made it possible to apply the input in the same sampling interval. Fig. 7.8 shows the algorithm steps simulated via ModelSim, while Figs. 7.9-7.12 shows the HIL simulations for $n = 10$. Compared with the FCS-MPC with delay compensation, NN-MPC had a similar performance on the steady-state error and harmonic distortion, as shown in Figs. 7.10 and 7.11.
However, in Fig. 7.9 the steady-state switching frequency of the NN-MPC was still slightly lower and the dynamic performance of the proposed approach reported in Fig. 7.12 were superior since it had a faster response and it did not suffer from the inaccuracy of the two-step forward prediction that leads, in any case, to a deterioration of the performance.

- For $n = 20$, the standard control was impractical unless the sampling interval was increased. The proposed one took 10.94 µs, which is small enough to permit the real-time implementation.

The times spent for the NN with $\eta = 8, 4, 2$ hidden neurons were similar for the described implementation since 12 adders and multipliers were used in parallel (the cluster voltages balancing requires 4 of them for each phase). In particular $\eta = 4$ required 27 iterations, while $\eta = 2$ required 25 iterations, resulting in 0.75 µs and 0.69 µs.
With the described implementation, without adding extra hardware resources, the NN with $\eta = 16$ required to solve the two loops in the ASM in Fig. 7.1 twice, resulting in 50 iterations. Analogously, the $\eta = 32$ NN required to solve the loops four times and 88 iterations were needed.
The execution times spent were about 1.39 µs and 2,44 µs for the 2 cases. The FCS-MPC for $n = 10$ with horizon $h = 2$ and 3 took about 89.9 µs and 852 µs, while the NN-MPC required the time for the $\eta = 8$ architecture.

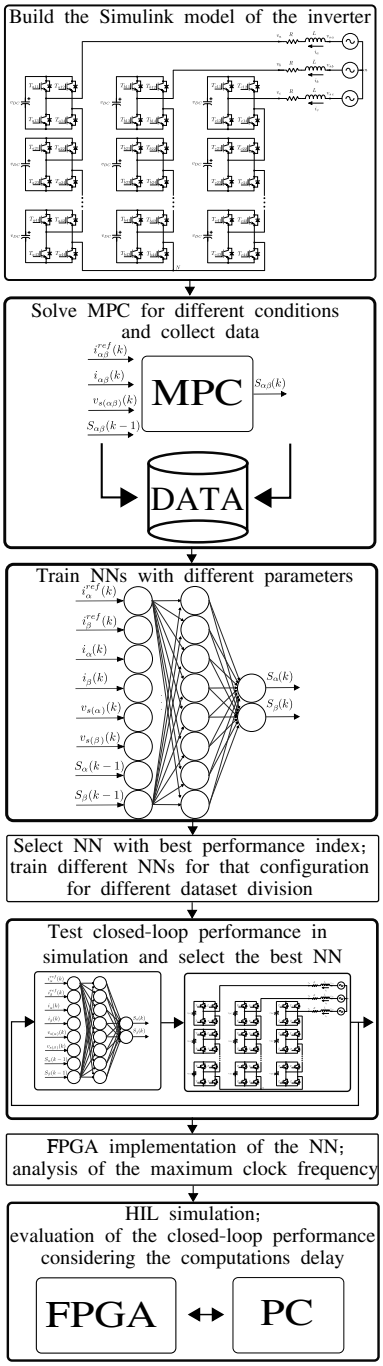


Figure 7.13: Schematic diagram flow of the overall NN-FCS training procedure.

Fig. 7.13 presents a schematic diagram flow that summarizes the whole procedure, which embeds (i) the simulation of the FCS-MPC to control the CHB inverter for collecting data, (ii) the training of the NN, (iii) the selection of the best NN in the closed-loop performance, (iv) the implementation on FPGA and, finally, (v) the HIL simulation used to analyze the effect of computation delay.

7.3 Conclusions

This Chapter presents the HW implementation of the NN-MPC described in Chapters 4 and 6 on FPGA platform. It discusses the performance of the proposed method in terms of clock cycles and time spent in computations and compares the NN approach to the standard FCS-MPC. Hardware in the loop simulations are presented to show the impact of the computations delay on the controller performances for the NN-MPC and the classical FCS-MPC, underlying the superior performance of the proposed technique with respect to the standard approach.

References

[1] Y. Zhang, X. Wu, X. Yuan, Y. Wang, and P. Dai, "Fast model predictive control for multilevel cascaded h-bridge statcom with polynomial computation time," *IEEE Transactions on Industrial Electronics*, vol. 63, no. 8, pp. 5231–5243, 2016.

[2] P. Cortes, J. Rodriguez, C. Silva, and A. Flores, "Delay compensation in model predictive current control of a three-phase inverter," *IEEE Transactions on Industrial Electronics*, vol. 59, no. 2, pp. 1323–1325, 2012.

[3] F. Simonetti, A. D'Innocenzo, and C. Cecati, "Neural network model predictive control for chb converters with fpga implementation," *IEEE Transactions on Industrial Informatics*, pp. 1–12, 2023.

[4] P. P. Chu, *RTL hardware design using VHDL: coding for efficiency, portability, and scalability.* John Wiley & Sons, 2006.

Chapter 8

FCS-MPC on CHB-STATCOM Prototype

This Chapter describes the implementation of the simple explicit solution introduced in Chapter 5. It presents the analysis of the execution time and the comparison of the proposed technique to existent methods for solving the currents FCS-MPC. The explicit solution was experimentally validated by using a 5-level CHB-STATCOM built by DigiPower Ltd [4]. This Chapter also describes the experimental setup and discusses the results obtained on the prototype [8].

8.1 Experimental Setup

The experimental setup is composed of a power circuit that handles the energy flow exchanged among the element of the inverter and the grid and a signal circuit responsible for sensing the state variable of the system, computing the control law and sending the gate signals to the drivers of the power switches.

8.1.1 The Power Circuit

The experimental setup in Fig. 8.2 is a 5-level three-phase CHB inverter, which is composed of $n = 2$ series-connected H-bridges per phase. Each H-bridge equips four insulated-bipolar gate transistors (IGBTs, NGTB30N120L2WG) with their four custom drivers [4]. The DC input buses of the H-bridges are fed by C=0.9 mF DC-link capacitors (AXCT27900J801DA) used to store energy from the grid. The system is connected to the grid through a 230/80 V isolation transformer and an L=0.6 mH filter inductor. Tab. 8.1 summarizes the main parameters of the hardware components.

8.1.2 The Signal Circuit

The system is composed of a main control board equipped with a 50 MHz Intel Cyclone V 5CEBA7F31C8 FPGA, which is responsible for the main calculations and which is

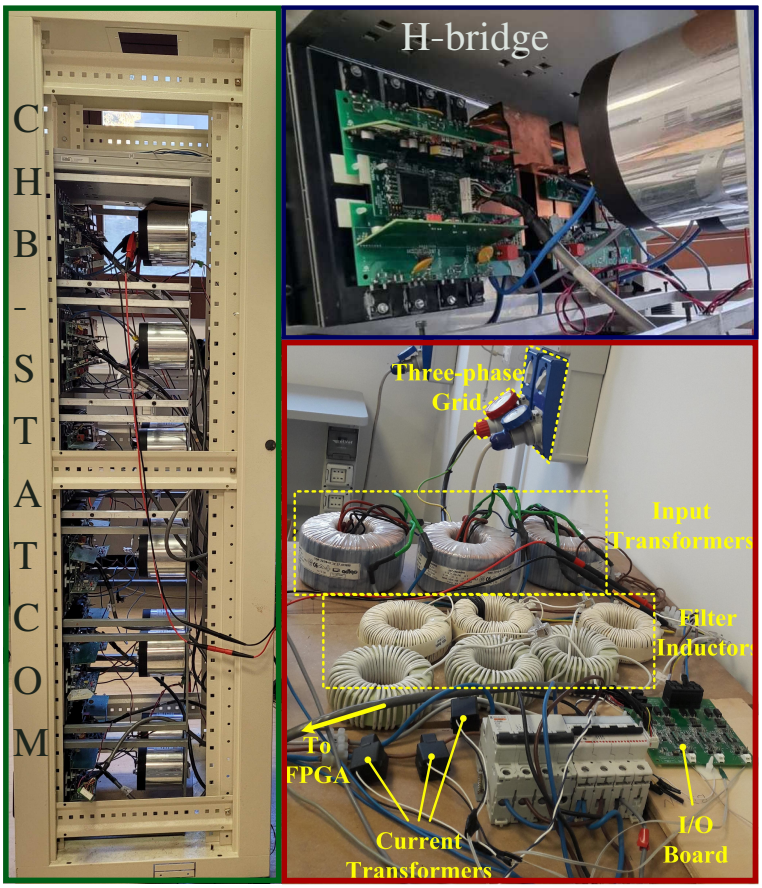


Figure 8.1: 5-level CHB STATCOM experimental setup.

connected to the other peripheral in a master-slave configuration, performing as a master and implementing a centralized control.
Each H-bridge equips a Texas Instrument TMS320 F28377SPTPT DSP (digital signal processor), which can perform distributed control tasks and it is connected to the analog circuitry that acts as an interface between power and signal circuits. The analog to digital converters (ADCs) embedded in the DSP are connected to the analog circuits that measures the DC input voltage (consisting of a voltage divider, an isolation amplifier ACM1301DWVR and an operational amplifier OPA350, from the power circuit to the signal circuit) and the output current of the H-bridge (consisting of an LEM transducer GO 30-SMS and an operational amplifier OPA350).
An external measurement board is used to sample the three-phase grid voltage (equipped with voltage dividers, isolation amplifiers AMC1301DWVR, amplifiers THS2421 and analog to digital converters ADS7254).
Seven master SPI (serial peripheral interface) modules were designed and instantiated in hardware into the FPGA, while the H-bridges and the external measurement board are used as SPI slaves, by using the SPI modules on the DSPs and the ADCs on the

Table 8.1: Parameters of the 5-level CHB-STATCOM.

	Value		Value
PCC voltage RMS	80 V	Capacitor C	0.9 mF
Grid frequency	50 Hz	Inductance L	0.6 mH
Rated power	± 2 kVAR	Sampling time T_s	50 µs
H-bridges per phase n	2	Inductor resistance R	0.5 Ω
DC-link voltage	80 V	Rated current RMS	4 A

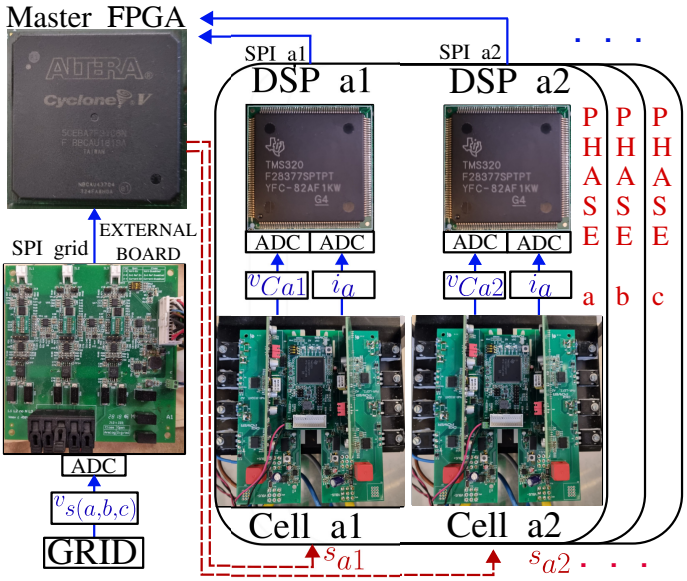


Figure 8.2: Block diagram of the experimental setup.

measurements board, which also implement an SPI-like communication.
Cablings connect the main control board to the DSPs of the six H-bridges and to the external measurement board, enabling the SPI communication, and directly connect the main control board output to the drivers of the H-bridges, such that the switching signals are directly sent from the FPGA.
Fig. 8.2 shows the schematic diagram of the experimental system.

8.2 Implementation on the CHB-STATCOM Prototype

The control algorithm was implemented on the described system by configuring the FPGA and programming the DSPs, in order to analyze the computational burden of the proposed algorithm compared to existing methods and to verify the controller performance.

8.2.1 Hardware-Software Implementation

The simple explicit solution presented in Chapter 5 was implemented on the FPGA platform by describing in VHDL the flow chart in Fig. 5.5, which can be considered as the ASM describing the control unit performing the proposed method.
The Nios II soft-processor provided by Intel Altera was instantiated into the FPGA to handle high-level operations, which were coded in C language. The interval timer core provided by Altera was added to send an interrupt to the Nios II processor at the start of every sampling interval T_s. Once the interrupt was handled, the interrupt service routine (ISR) was executed, which performed the following steps:

- the switching signals computed during the previous sampling interval were sent to the drivers;
- the master FPGA processor started the SPI communication with the seven slave peripherals (the six H-bridges and the external measurement board) to sample the three-phase grid voltage, the DC-link capacitors voltages and the current flowing through the H-bridges, i.e., the three-phase current exchanged with the grid;
- once the grid voltages were acquired, a start operation signal was sent to the HW module performing the PLL, implemented in hardware;
- when the end operation signal was asserted by the the PLL module, i.e., the grid phase voltage was computed, Nios II sent the start operation to the HW module performing the currents controller operations, which computed $\mathbf{S}^*_{\alpha,\beta}(k+1)$;
- when the end operation signal was asserted by the currents controller HW module, the matrix-vector multiplication performing the inverse Clarke transformation (in the Appendix) was computed, obtaining $\mathbf{S}^*_{a,b,c}(k+1)$;
- Nios II sent the start operation to the HW module performing the individual voltages balance, which computed the switching signals $s^*_{pi}(k+1)$;
- when the end operation was asserted by the individual voltages balance HW module, the Nios II processor stored the switching signals, that were applied at the beginning of the next sampling interval, when a new timer interrupt was asserted.

On the other side, the DSPs on the H-bridges, coded in C language, were programmed to sample currents and voltages when the SPI communication starts. A more exhaustive discussion about the FPGA-DSP communication is provided in Chapter 10.

8.2.2 Implemented Overall Control Scheme

The control scheme comprised a reference generator for the reference direct current able to balance the overall capacitor voltage. A PI regulator was employed to compute the reference direct current i_d^{ref} capable of stabilizing the average DC-link voltage $\overline{v}_C$ to the nominal value V_{DC}. The MPC scheme followed the well-known partially stratified approach, where the currents controller is followed by the voltages balancing controller, as in [3, 1, 2]. The reference currents were transformed into α, β frame and the currents controller was computed as in Chapter 5.
This study focused on the currents controller and the optional clusters voltages balance

was not employed, while the individual voltages balance was implemented as described in Subsection 7.1.3.
The PI reference generator was tuned based on [7] and the proportional and integral gains K_P^{id} and K_I^{id} were set to 1 and 100. The MPC parameters tuning was carried out by using the empirical method in [6, 5]. The current control was tuned to obtain a current THD of about 3%, and the voltage balancing was tuned to ensure a maximum 10% voltage ripple. Tab. 8.2 lists the weighting factors, where $\mathbf{I}_m$ is the $m \times m$ identity matrix.

Table 8.2: MPC control parameters.

	K_P^{id}	K_I^{id}	$\mathbf{Q}$	$\mathbf{P}$	$\mathbf{Q}_{ib}$	$\mathbf{P}_{ib}$
Value	1	100	$\mathbf{I}_2$	$10^{-3} \times \mathbf{I}_2$	$\mathbf{I}_n$	$10^{-4} \times \mathbf{I}_n$

Fig. 8.3 shows the diagram of the overall control scheme.

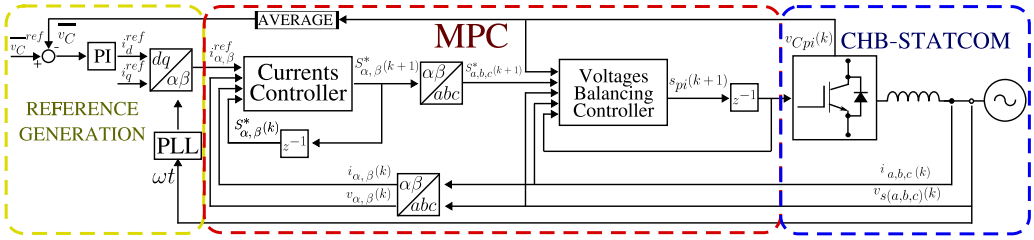


Figure 8.3: Control scheme of FCS-MPC for CHB-STATCOM.

8.2.3 Analysis of the Computational Burden and the Execution Time

In order to analyze the computational advantages of the proposed technique, a comparison with two other methodologies is presented. Tab. 8.3 shows the number of elementary operations for the proposed approach compared with "Fast MPC" in [1] and "B&B Approach" in [2]. The "Fast MPC" algorithm reduces the computational burden from exponential to quadratic level. The "B&B Approach" leads to an algorithm of linear complexity with respect to n, overcoming the previous one. Finally, the proposed approach consists of less computations for every $n > 1$ and the number of operations is constant irrespective of n, further overcoming the other methodologies.

Table 8.3: Computational burden of different approaches.

	Fast MPC [1]	B&B Approach [2]	Explicit Solution
Sums	$12n^2 + 50n + 16$	$16n + 13$	38
Multiplications	$48n + 18$	$22n + 27$	42
Comparisons	$12n^2 + 10n + 2$	$10n + 2$	15
Rounds	0	$2n + 1$	2
Divisions	0	1	0

Tab. 8.4 summarizes the time and the resource utilization of the different parts of

the implementation on the FPGA, i.e., SPI transmission, phase-locked loop, PI of the reference direct current, currents controller (CC) and individual voltages balance (IB).

Table 8.4: Time and space utilization on FPGA of the simple explicit solution.

	SPI	PLL	PI i_d^{ref}	CC	IB
Time spent $[\mu s]$	7.74	1.64	0.52	0.94	0.58
ALMs needed	1178	5620	55.4	2860	2051
Combinational ALUTs	2139	6525	78	4202	3252
Dedicated Logic Registers	1918	918	269	3430	2088
DSP blocks	0	3	0	24	36

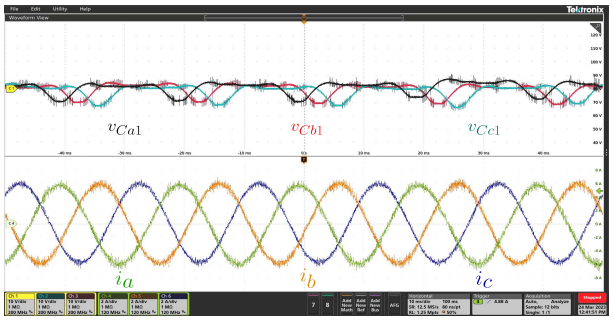
Figure 8.4: Tektronix 5 series MSO oscilloscope used in the experimental setup.

8.2.4 Control Performance

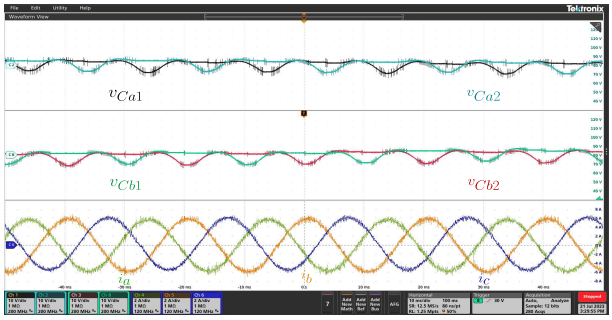
The controller performance of the CHB-STATCOM were analyzed in both capacitive and inductive operational modes for both steady-state and transient conditions. Parametric variations were applied to the dynamical model to verify the robustness of the controller. The experimental analyses were carried out by employing the Tektronix 5 series MSO oscilloscope in Fig. 8.4.

Inductive Mode CHB-STATCOM

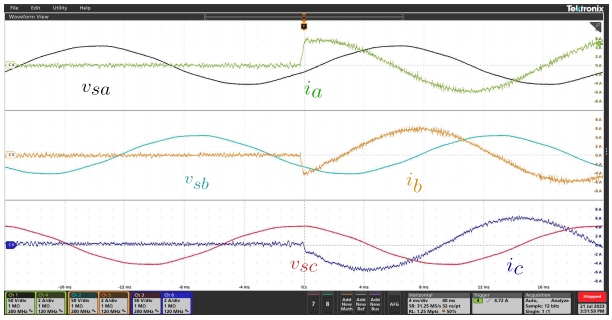
Fig. 8.5a shows the steady-state currents for a 4 A RMS inductive current and the related voltages on capacitors of different phases, i.e., $v_{Ca1}, v_{Cb1}, v_{Cc1}$.



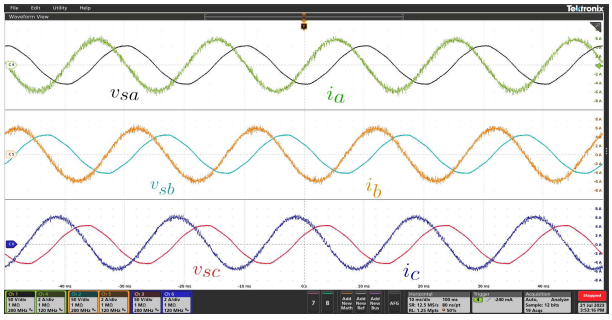
(a) Steady-state inductive mode: $v_{Ca1}, v_{Cb1}, v_{Cc1}$ [10 V/div], i_a, i_b, i_c [2 A/div], time [10 ms/div].



(b) Steady-state inductive mode: $v_{Ca1}, v_{Ca2}, v_{Cb1}, v_{Cb2}$ [10 V/div], i_a, i_b, i_c [2 A/div], time [10 ms/div].



(c) Step response inductive mode: v_{sa}, v_{sb}, v_{sc} [40 V/div], i_a, i_b, i_c [2 A/div], time [4 ms/div].



(d) Steady-state inductive mode: v_{sa}, v_{sb}, v_{sc} [40 V/div], i_a, i_b, i_c [2A /div], time [10 ms/div].

Figure 8.5: CHB-STATCOM inductive mode.

Fig. 8.5b shows the steady-state currents and the DC-link voltages on the same phases, i.e., $v_{Ca1}, v_{Ca2}, v_{Cb1}, v_{Cb2}$. The two figures demonstrate good current reference tracking performance and voltage balancing performance among capacitors on the same phase and among DC-link of the different phases.
Finally, Fig. 8.5c shows the step response for the 4 A RMS inductive current reference, while Fig. 8.5d underlines the phase-shift between grid voltage and currents in the steady-state inductive mode, i.e., the currents are leading the grid voltages by $\pi/2$ radians.
Fig. 8.6 shows the 5-level three-phase CHB-STATCOM output voltage and currents when injecting +2 kVAR into the grid.

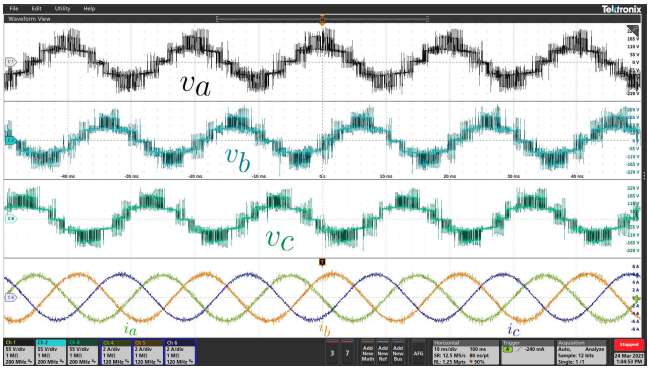


Figure 8.6: Steady-state CHB-STATCOM output: v_a, v_b, v_c [55 V/div], i_a, i_b, i_c [2 A/div], time [10 ms/div].

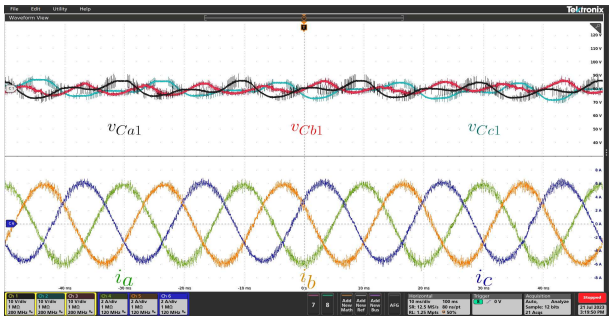
Capacitive Mode CHB-STATCOM

Similarly, the system was tested for the rated current in capacitive mode.
Fig. 8.7a shows the output current underlying the DC-link voltage balance among the three phases. Fig. 8.7b highlights the balance of the different DC-link capacitors in the same phase.
Fig. 8.7c and Fig. 8.7d show the step response and steady-state performance of the system when absorbing 2 kVAR from the grid, i.e., the output currents are lagging the grid voltage by $\pi/2$ radians.
Tab. 8.5 summarizes the harmonic components of the output current for both inductive and capacitive modes when following a 4 A RMS current reference.

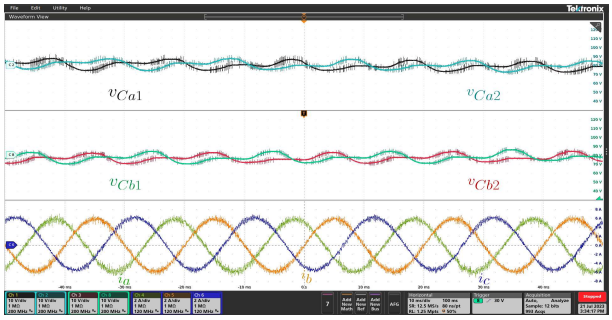
Table 8.5: Harmonic components for a 4A RMS 50Hz current.

	DC	50 Hz	100 Hz	150 Hz	200 Hz	250 Hz
Inductive [A]	0.086	5.77	0.014	0.011	0.008	0.042
Capacitive [A]	0.108	5.88	0.010	0.017	0.013	0.006

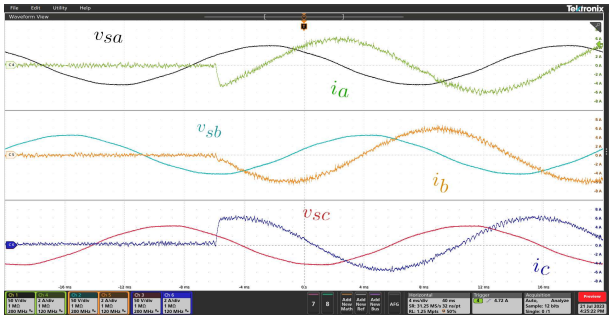
	300 Hz	350 Hz	400 Hz	450 Hz	500 Hz
Inductive [A]	0.014	0.021	0.022	0.009	0.006
Capacitive [A]	0.028	0.023	0.037	0.029	0.038



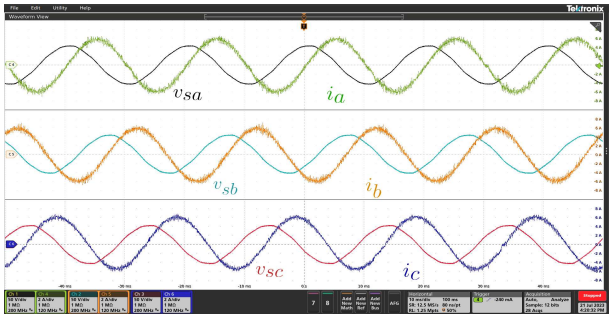
(a) Steady-state capacitive mode: $v_{Ca1}, v_{Cb1}, v_{Cc1}$ [10 V/div], i_a, i_b, i_c [2 A/div], time [10 ms/div].



(b) Steady-state capacitive mode: $v_{Ca1}, v_{Ca2}, v_{Cb1}, v_{Cb2}$ [10 V/div], i_a, i_b, i_c [2 A/div], time [10 ms/div].



(c) Step response capacitive mode: v_{sa}, v_{sb}, v_{sc} [40 V/div], i_a, i_b, i_c [2 A/div], time [4 ms/div].



(d) Steady-state capacitive mode: v_{sa}, v_{sb}, v_{sc} [40 V/div], i_a, i_b, i_c [2 A/div], time [10 ms/div].

Figure 8.7: CHB-STATCOM capacitive mode.

8.2.5 Robustness

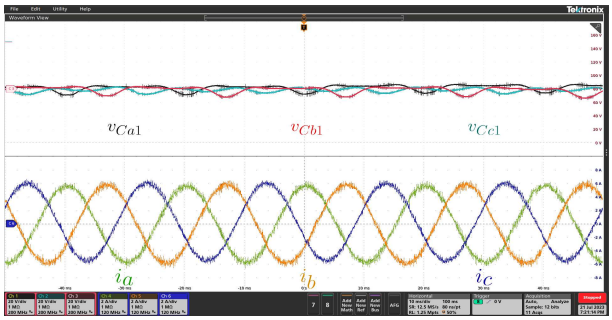
The robustness of the proposed controller was studied by applying parametric variations to the nominal model used to compute the predictions.
Since the aim of this work was to study the currents control performance, only the parameters of the currents model were changed, i.e., R and L, while the individual DC-link voltage balance was kept unchanged.
Figs. 8.8a and 8.8b show the performance when the rated inductance L and resistance R in the mathematical model are 20 % smaller than the actual values.
Conversely, Figs. 8.8c and 8.8d present the performance when the controller considers R and L as 20 % larger than the actual values.
Tab. 8.6 summarizes the total harmonic distortion of the three-phase output currents for both cases and the nominal case, highlighting that the performance of the system is not degraded by these parametric variations.

Table 8.6: Total harmonic distortion parametric variations.

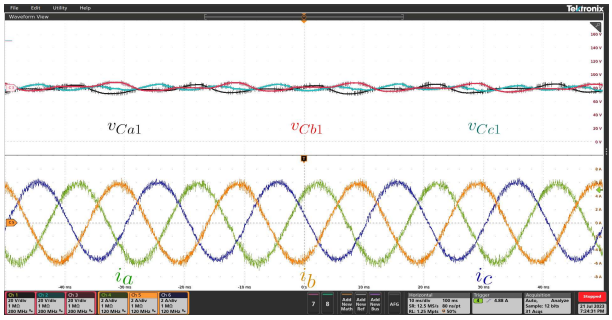
	Nominal Model	$(R, L) - 20$ %	$(R, L) + 20$ %
THD inductive mode	2.965 %	3.146 %	2.984 %
THD capacitive mode	3.287 %	3.252 %	3.498 %

8.2.6 Time Comparison with Existing Approaches

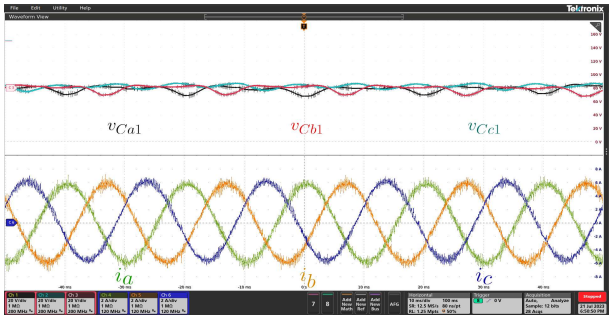
Fig. 8.9a shows the time spent with the proposed approach for different number of levels compared to the "B&B Approach" [2]. The performance of the "Fast MPC" [1] is omitted since it quickly exceeds the 50 µs sampling time for n=3 and 100 µs for n=4.
To make a fair comparison, the methods were implemented on the same FPGA platform with the same technological choices, i.e., the computations of the controllers were fully implemented through HW acceleration on the FPGA, only fixed-point arithmetic operations were used (only the division in the "B&B Approach" in [2] was implemented with floating-point arithmetic since the division is an ill-conditioned operation), the same degree of parallelism was employed by using 12 multipliers and accumulators. Fig. 8.9a also shows the time spent by the overall control in both cases, by including the time for samplings, computing the PLL, i_d^{ref} and the voltage balancing procedure. Even if the proposed currents control requires constant time, the overall time increases due to the voltage balance.
However, it is clear that, with the proposed approach, we can still control an inverter with $n \geq 20$ within the 50 µs sampling interval, while the approach in [2] fails if $n > 10$, forcing the design of a larger sampling time that leads to performance degradation for both steady-state and transient operations.



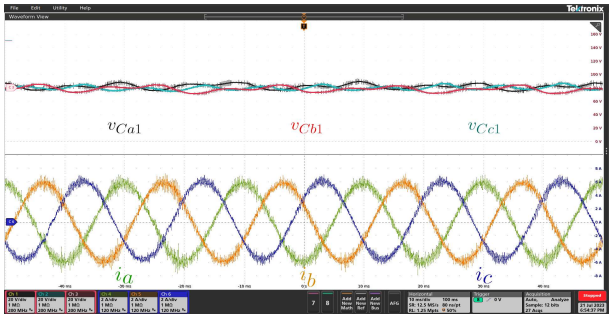
(a) Steady-state inductive mode -20 % R and L: $v_{Ca1}, v_{Cb1}, v_{Cc1}$ [20 V/div], i_a, i_b, i_c [2 A/div], time [10 ms/div].



(b) Steady-state capacitive mode -20 % R and L: $v_{Ca1}, v_{Cb1}, v_{Cc1}$ [20 V/div], i_a, i_b, i_c [2 A/div], time [10 ms/div].

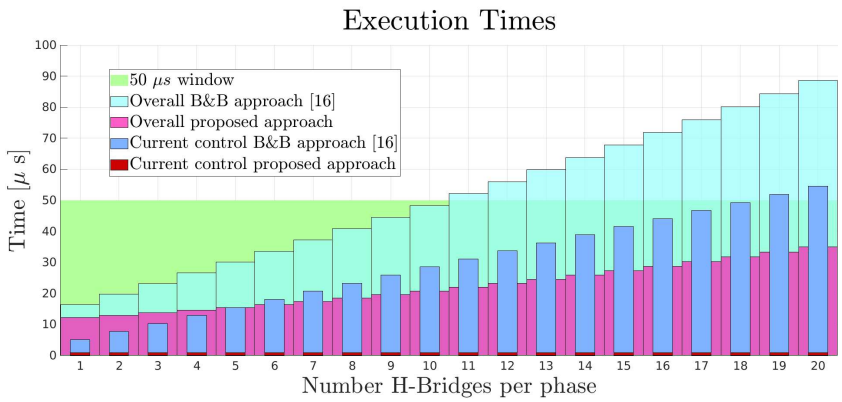


(c) Steady-state inductive mode $+20$ % R and L: $v_{Ca1}, v_{Cb1}, v_{Cc1}$ [20 V/div], i_a, i_b, i_c [2 A/div], time [10 ms/div].

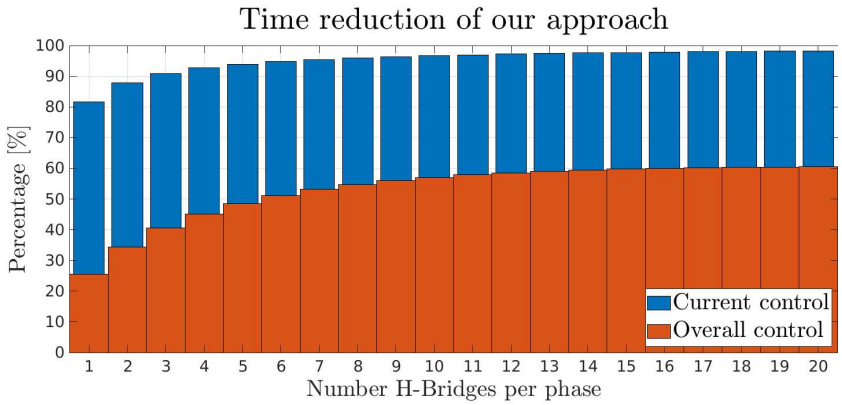


(d) Steady-state capacitive mode $+20$ % R and L: $v_{Ca1}, v_{Cb1}, v_{Cc1}$ [20 V/div], i_a, i_b, i_c [2 A/div], time [10 ms/div].

Figure 8.8: Steady-state CHB-STATCOM parametric variations.



(a) Execution times comparison for different levels.



(b) Time reduction of the proposed approach for different levels.

Figure 8.9: Time comparison with "B&B Approach".

Fig. 8.9b highlights the time reduction of the proposed approach, which shows a time reduction of 97% for the currents control and 57% for the overall control for $n = 10$, which remains stable for $n = 20$.

8.3 Conclusions

This Chapter presents the implementation of the simple explicit solution for FCS-MPC of a CHB converter described in Chapter 5.
The proposed method was compared with two well-known approaches in the literature and tested on a 5-level CHB-STATCOM prototype. The number of elementary operations and the time spent for the control algorithm was determined and the hardware resources utilization on FPGA was presented. The proposed method overcomes the existing ones since it computes the global optimum solution by requiring a few simple operations that do not depend on the number of levels of the converter.

References

[1] Y. Zhang, X. Wu, X. Yuan, Y. Wang, and P. Dai, "Fast model predictive control for multilevel cascaded h-bridge statcom with polynomial computation time," *IEEE Transactions on Industrial Electronics*, vol. 63, no. 8, pp. 5231–5243, 2016.

[2] Y. Zhang, X. Wu, and X. Yuan, "A simplified branch and bound approach for model predictive control of multilevel cascaded h-bridge statcom," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 10, pp. 7634–7644, 2017.

[3] F. Simonetti, A. D'Innocenzo, and C. Cecati, "Neural network model predictive control for chb converters with fpga implementation," *IEEE Transactions on Industrial Informatics*, pp. 1–12, 2023.

[4] DigiPower, "DigiPower Website." https://digipower.it, 2023. [Accessed: September 25, 2023].

[5] 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.

[6] P. Cortés, S. Kouro, B. La Rocca, R. Vargas, J. Rodriguez, J. I. Leon, S. Vazquez, and L. G. Franquelo, "Guidelines for weighting factors design in model predictive control of power converters and drives," in *IEEE ICIT 2009*, pp. 1–7, 2009.

[7] D. Shetty and N. Prabhu, "Ziegler nichols method based robust reactive current controller for statcom," *Energy Procedia*, vol. 117, pp. 543–550, 2017.

[8] F. Simonetti, A. D'Innocenzo, and C. Cecati, "Simple explicit solution of finite control set model predictive control," *IEEE Transactions on Industrial Electronics*, 2023, unpublished.

Part IV

Addressed Implementation Problems and Developed Solutions

Chapter 9

DC Current Injection when Using Current Transformer Sensors

During the research work on the experimental setup, many problems were faced related to HW implementation of the algorithms, FPGA debugging process, sampling of measurements, synchronization between the main FPGA and the DSPs, and so on, to name a few.
In the original experimental setup, the three-phase output currents were not sensed through the LEM transducers on the H-bridges mentioned in Subsection 8.1.2. They were sensed by the three current transformer (CTs) sensors TDK CCT27-02 (connected to amplifier THS2421 and ADC ADS7254 of the external measurement board) in Fig. 9.1 and a large amount of DC current appeared in the output terminals of the CHB-STATCOM.
This Chapter presents the problem of the DC current injection of the CHB-STATCOM to the grid caused by the use of CTs. It mathematically analyzes the problem and proposes a simple solution to eliminate the DC current without the need for extra HW [21, 22].

Figure 9.1: Current transformer sensor TDK CCT27-02.

9.1 Introduction

A DC current injection can be caused by asymmetry in gate signals or power transistors outputs, errors in the analog to digital converters, zero-drifting in sensors and signal conditioning circuits [3]. These shortcomings can be attenuated by improving the measuring circuit or by adding a line-frequency transformer between the converter and the grid, but with increasing losses, size, and cost of the overall structure.
The CTs are popular sensors due to their galvanic insulation and electromagnetic capability, their ability to deal with large inrush currents and low cost. However, they largely amplify the DC current injection problem. In fact, since they are based on the electromagnetic coupling principle, they are unable to measure the DC component of the circulating current [15]. As a result, a closed-loop control can only stabilize the AC current, ignoring the DC component.
Grid-tied converters, such as FACTS devices (like STATCOMs) or photovoltaic inverters, inject/absorb AC power to/from the grid [2]. Hence, these applications are regulated by power quality standards, which fix strict limits to the injected DC current because they increase the losses in distribution transformers, corrode network cabling and lead to safety issues and saturation of magnetic cores. The imposed limit varies depending on the country. USA and China tolerate 0.5% of the rated output current, UK and Australia allow 5 mA while Germany accepts 1 A DC current [20].
Adopted solutions for detecting and suppressing the DC current usually require the use of extra hardware [13, 5, 6, 7]. For instance, in [5], an extra current sensor is added in the DC-link of a transformer-less H-Bridge to detect the DC offset in the output phase currents by measuring the DC-link current during the freewheeling intervals of the unipolar PWM scheme. In [6], the authors proposed a combination of hardware, including an isolated RC attenuation circuit, and software, an estimation scheme through a DC component filtering algorithm, for a single-phase PV inverter. In [7], the CT is equipped with a power amplifier, an additional winding, and a commercial current sensor. The output of the extra sensor is filtered to obtain the DC component used in the closed-loop control. In order to reduce the costs of extra hardware, software-based solutions have been presented in the literature [14, 4, 8, 9, 12]. In [4], the DC current injected by a static synchronous compensator [1] employing CTs is estimated by filtering the line frequency ripple on the capacitors with a moving average filter. A PI regulator was added to the standard control scheme [19, 18] to compensate for the detected DC current. In [8], the DC current injected by a grid-tied inverter is compensated by controlling the DC component of the output voltage. A PI-based control scheme is employed to compute d, q currents to compensate for this DC voltage. In [9], a proportional-integral-resonant controller is proposed to eliminate the DC current of a single-phase PV inverter, which is estimated from the line frequency ripple of the DC-link voltage. There is a correlation between the DC current injected by the inverter in the grid and the line-frequency ripple on the DC-link voltage [12].
For CHB-STATCOM linear control schemes, a strategy for balancing the DC-link voltage of the three phase clusters is commonly employed and, as presented in Subsection 3.2.2, this concept can be imported into the FCS-MPC framework [11, 10].
This Chapter proposes the use of a cluster balancing strategy for the FCS-MPC of CHB-STATCOM to suppress the DC current injected into the grid. Despite the topic was already addressed for liner controllers, to the authors' knowledge, it was never faced

for the MPC strategy. Through extensive theoretical and experimental analyses, this Chapter demonstrates that the DC output current can be suppressed by controlling the DC-link voltage of the clusters accomplished by injecting a common-mode voltage. The proposed approach enables the DC current elimination in MPC-based CHB-STATCOM with an algorithm that does not require extra sensors and complex filtering procedures. Moreover, the proposed formulation is simple to implement and computationally efficient. Thanks to the MPC capability to handle constraints, the injected voltage does not create unbalances as in [4] or mismatches with the reference current as in [8].
The 5-level CHB-STATCOM employing CTs is used as a demonstrator for the proposed algorithm, validating the effectiveness of the proposed strategy.
A comparison with a standard PI approach is provided underlying the differences and advantages of the proposed technique.

9.2 DC Current Injection Problem and Solution

This Section describes the relation between the DC output current and the DC input voltage of the cluster. It proves that, by adding the cluster voltage balance, which aims to reduce the ripple on the cluster voltage, the DC output current is automatically eliminated, by introducing a common-mode voltage in the output terminals of the CHB-STATCOM.

9.2.1 Link Between DC Current and Cluster Voltage

It is well–known in the literature that, by using CT sensors, a DC current is generated at the output of the STATCOM [4]. Let us assume that we have a DC current in the output of the cluster p of the CHB-STATCOM, the simplified model of the individual cluster can be drown as in Fig. 9.2. The relationship between the current on the AC side $i_p(t)$ of

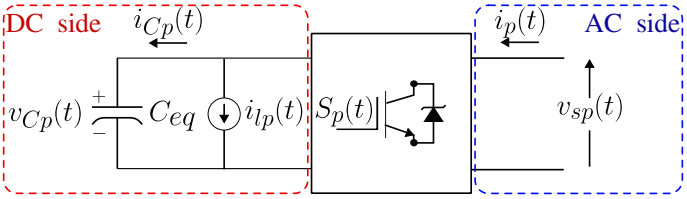


Figure 9.2: Simplified individual cluster of the CHB-STATCOM.

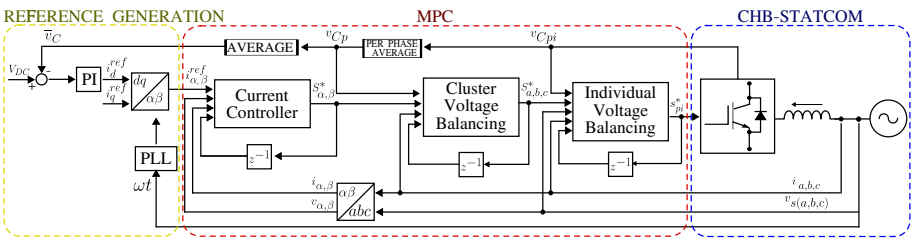


Figure 9.3: CHB-STATCOM overall control scheme with clusters balance.

the cluster and the DC-link voltage $v_{Cp}(t)$ is:

$$C_{eq}\frac{dv_{Cp}(t)}{dt} = i_{Cp}(t) = S_p(t)\cdot i_p(t) - i_{lp}(t), \tag{9.1}$$

where $S_p(t)$ is the switching function, $i_{lp}(t)$ represents the losses in the power circuit and C_{eq} is the equivalent capacitor of the overall cluster p defined as $C_{eq} = C/n$. Since this study is focused on the low frequency behavior of the system, the switching function is assumed to be $S_p(t) = \tilde{S}\cos(\omega t+\phi_S)$, by ignoring high frequency component. When there is a DC current in the AC side, the STATCOM output current is $i_p(t) = \tilde{I}_S\cos(\omega t+\phi_I)+\overline{I}_S$. By substituting $S_p(t)$ and $i_p(t)$ in (9.1), it follows that

$$\begin{aligned} C_{eq}\frac{dv_{Cp}(t)}{dt} &= \tilde{S}\cos(\omega t+\phi_S)\cdot\left[\tilde{I}_S\cos(\omega t+\phi_I)+\overline{I}_S\right] - i_{lp}(t) \\ &= \frac{\tilde{S}\tilde{I}_S}{2}\left[\cos(\phi_S-\phi_I) + \cos(2\omega t+\phi_S+\phi_I)\right] \\ &\quad + \tilde{S}\overline{I}_S\cos(\omega t+\phi_S) - i_{lp}(t), \end{aligned} \tag{9.2}$$

obtaining the equation describing the cluster voltage dynamics when a DC current is in the output, where we can notice that a constant term, a line-frequency and a double line-frequency ripples appear.
Now, let us consider introducing a DC offset in the switching function $S_p(t)$.
Assuming the switching function $S_p(t) = \tilde{S}\cos(\omega t+\phi_S)+\overline{S}$ and substituting it in (9.1), it follows that the new DC-link voltage dynamics is:

$$\begin{aligned} C_{eq}\frac{dv_{Cp}(t)}{dt} &= \left[\tilde{S}\cos(\omega t+\phi_S)+\overline{S}\right]\cdot\left[\tilde{I}_S\cos(\omega t+\phi_I)+\overline{I}_S\right] - i_{lp}(t) \\ &= \frac{\tilde{S}\cdot\tilde{I}_S}{2}\left[\cos(\phi_S-\phi_I) + \cos(2\omega t+\phi_S+\phi_I)\right] \\ &\quad + \tilde{S}\cdot\overline{I}_S\cos(\omega t+\phi_S) + \overline{S}\cdot\tilde{I}_S\cos(\omega t+\phi_I) + \overline{S}\overline{I}_S - i_{lp}(t) \\ &= \frac{\tilde{S}\cdot\tilde{I}_S}{2}\cos(\phi_S-\phi_I) + \overline{S}\cdot\overline{I}_S - i_{lp}(t) \\ &\quad + \tilde{S}\cdot\overline{I}_S\cos(\omega t+\phi_S) + \overline{S}\cdot\tilde{I}_S\cos(\omega t+\phi_I) \\ &\quad + \frac{\tilde{S}\cdot\tilde{I}_S}{2}\cos(2\omega t+\phi_S+\phi_I). \end{aligned} \tag{9.3}$$

By comparing (9.3) and (9.1) it is clear that, by adding the $\overline{S}$ term, the constant term $\overline{S}\cdot\overline{I}_S$ and the line frequency ripple $\overline{S}\cdot\tilde{I}_S\cos(\omega t+\phi_I)$ are added to the cluster voltage dynamics, while the double line-frequency ripple is not affected by $\overline{S}$.
Since this Chapter refers to a STATCOM, which aims to inject quadrature current (reactive power), the relation $\phi_I \simeq \phi_S \pm \pi/2$ holds. By consequence, the line frequency ripple in (9.3) is:

$$\begin{aligned}\tilde{I}_C \sin(\omega t + \phi_{I_C}) &= \tilde{S} \cdot \overline{I}_S \cos(\omega t + \phi_S) + \overline{S} \cdot \tilde{I}_S \cos(\omega t + \phi_I) \\ &\simeq \tilde{S} \cdot \overline{I}_S \cos(\omega t + \phi_S) + \overline{S} \cdot \tilde{I}_S \cos(\omega t + \phi_S \pm \pi/2) \\ &= \tilde{S} \cdot \overline{I}_S \cos(\omega t + \phi_S) \mp \overline{S} \cdot \tilde{I}_S \sin(\omega t + \phi_S) \\ &= \sqrt{(\tilde{S}\overline{I}_S)^2 + (\overline{S}\tilde{I}_S)^2} \sin\left(\omega t + \phi_S \mp \tan^{-1}\left(\frac{\tilde{S} \cdot \overline{I}_S}{\overline{S} \cdot \tilde{I}_S}\right)\right).\end{aligned} \tag{9.4}$$

It results that, by injecting $\overline{S}$, it is not possible to reduce the amplitude of the line frequency ripple. By contrast, by introducing $\overline{S}$, we can reduce the constant part of the voltage dynamics in (9.3).

9.2.2 Minimization of the DC Current

Subsection 9.2.1 shows that an offset $\overline{S}$ in the switching function $S_p(t)$ affects the cluster voltage dynamics. This Subsection shows that minimizing the error between the reference voltage V_{DC} and the cluster voltage $v_{Cp}(t)$ is equivalent to minimizing the DC current $\overline{I}_S$ in steady-state condition. Therefore, the cluster balancing algorithm satisfies two objectives simultaneously: it balances the voltage shared among the clusters and eliminates the DC current at the output current.
Considering the system at steady-state condition, because of the effect of the upper voltage control level (the reference generator in Fig. 9.3), it is reasonable to assume $v_{Cp}(t) \simeq V_{DC}$. Therefore, minimizing the error between V_{DC} and $v_{Cp}(t)$ is equivalent to force the equation in (9.3) to zero, enforcing $dv_{Cp}(t)/dt \to 0$. From the analysis of the ripple in (9.4), it is clear that $\overline{S}$ cannot minimize the oscillating part and can only reduce the constant part of (9.3). Thus, the following holds:

$$\begin{aligned}\min_{\overline{S}} \left(v_{Cp}(t) - V_{DC}\right)^2 &\overset{\text{steady-state}}{\simeq} \min_{\overline{S}} \left(C_{eq} \frac{dv_{Cp}(t)}{dt}\right)^2 \\ &= \min_{\overline{S}} \left(\frac{\tilde{S} \cdot \tilde{I}_S}{2} \cos(\phi_S - \phi_I) + \overline{S} \cdot \overline{I}_S - i_{lp}(t)\right)^2.\end{aligned} \tag{9.5}$$

By solving this optimization problem, the constant part is forced to approach zero, which enforces the law:

$$\overline{S} = \left(i_{lp}(t) - \frac{\tilde{S} \cdot \tilde{I}_S}{2} cos(\phi_S - \phi_I)\right) / \overline{I}_S, \tag{9.6}$$

which, finally, establishes that by increasing the DC offset $\overline{S}$ in the switching function, the DC current $\overline{I}_S$ decreases. The result is that the DC current is attenuated by introducing an offset in the switching function with the aim to minimize the error between reference and cluster voltage.

9.2.3 Clusters Voltages Balance for MPC

The clusters voltages balance optimization layer is described in Subsection 3.2.2. The constraint $\mathbf{S}^{*}_{\alpha,\beta}(k) = \mathbf{T}_{2\times 3} \cdot \mathbf{S}_{a,b,c}(k)$ in (3.7) ensures that the injected common-mode voltage does not affect the α, β currents controlled by the first optimization layer. In other words, the offset $\overline{S}$ is introduced without affecting $\tilde{S}$, ϕ_S, $\tilde{I}_S$, ϕ_I of (9.3). The algorithm can be simply implemented by searching among the $2n{+}1$ redundant vectors in the a, b, c coordinates, as discussed in 7.1.2 and summarized in the flowchart in Fig. 9.4. The overall control scheme is summarized in Fig. 9.3.

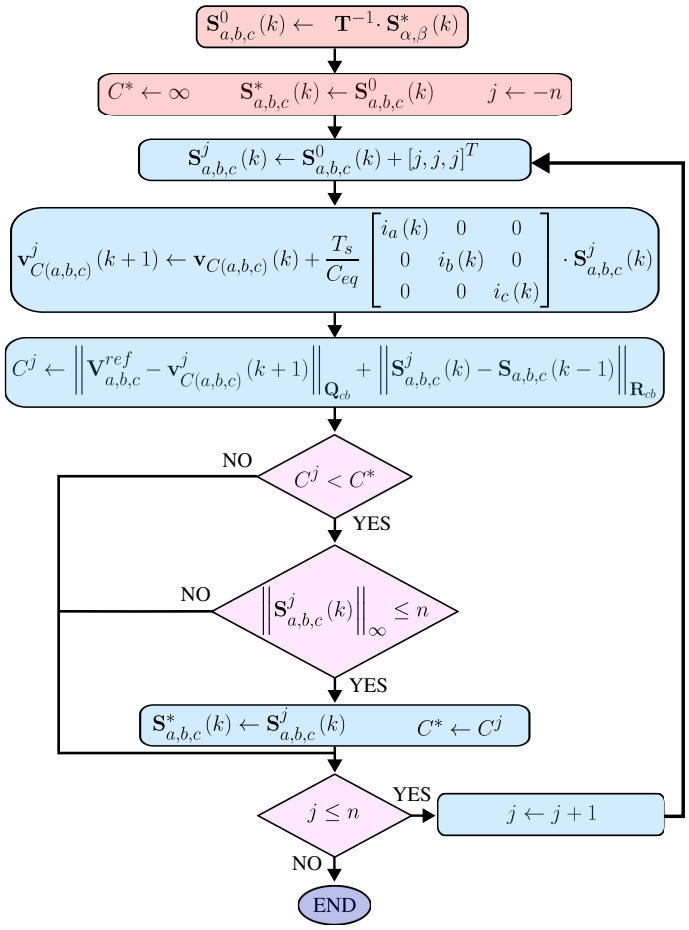


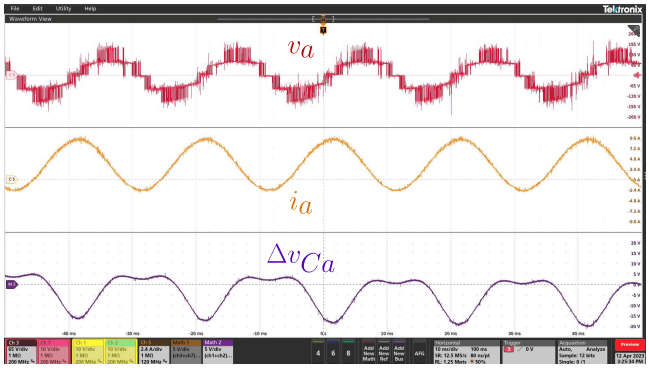
Figure 9.4: Cluster balancing flow chart for DC current elimination.

9.3 Experimental Results

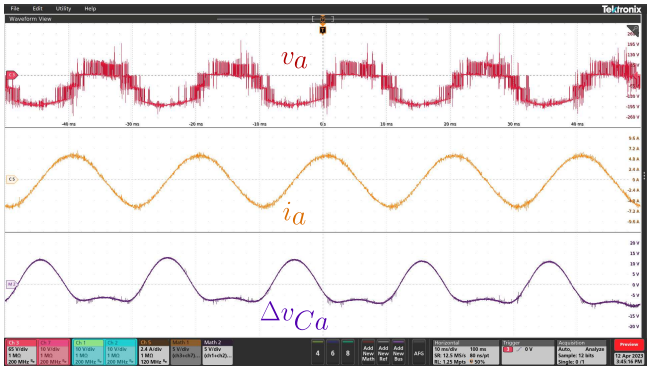
The proposed strategy was tested on the 5-level CHB-STATCOM described in 8.1, where the currents are sensed by three TDK CCT27-02 current transformers. The control tuning parameters of the reference generator, the currents controller and individual voltages

balance were set as in Section 8.1, while the clusters balance parameters were empirically set to guarantee a 0.5 pu DC offset, i.e., 0.3A, which leaded to $\mathbf{Q}_{cb}$=$\mathbf{I}_3$ and $\mathbf{R}_{cb}$ =$10^{-2}\times\mathbf{I}_3$ [18, 17, 16].

9.3.1 Experimental Validation of the Theory



(a) CHB-STATCOM MPC with CTs without cluster balancing: $v_{sa}\,(t)$ [65 V/div], $i_a\,(t)$ [2.4 A/div], $\Delta v_{Ca}\,(t)$ [5 V/div], time [10 ms/div].



(b) CHB-STATCOM MPC with CTs with cluster balancing: $v_{sa}\,(t)$ [65 V/div], $i_a\,(t)$ [2.4 A/div], $\Delta v_{Ca}\,(t)$ [5 V/div], time [10 ms/div].

Figure 9.5: CHB-STATCOM MPC with CTs without and with cluster balancing.

Fig. 9.5a shows the steady-state output voltage $v_{sa}\,(t)$, current $i_a\,(t)$ and the error in cluster voltage $\Delta v_{Ca}\,(t) = (v_{Ca}\,(t) - nV_{DC})$ when a 4 A RMS quadrature current is given as reference without cluster balancing. The output voltage does not presents a DC offset while a DC current appears in the output current. Fig. 9.5b shows the waveforms when using the proposed cluster balancing. As shown, the clusters control layer adds an offset in the output voltage that eliminates the DC current. The fast Fourier transform analysis confirms the discussion in Section 9.2. Figs. 9.6, 9.7 and 9.8 show, respectively, the Fourier expansion of $\Delta v_{Ca}\,(t)$, $i_a\,(t)$ and $v_{sa}\,(t)$ and the amplitude and phase of the low harmonics are summarized in Table 9.1.

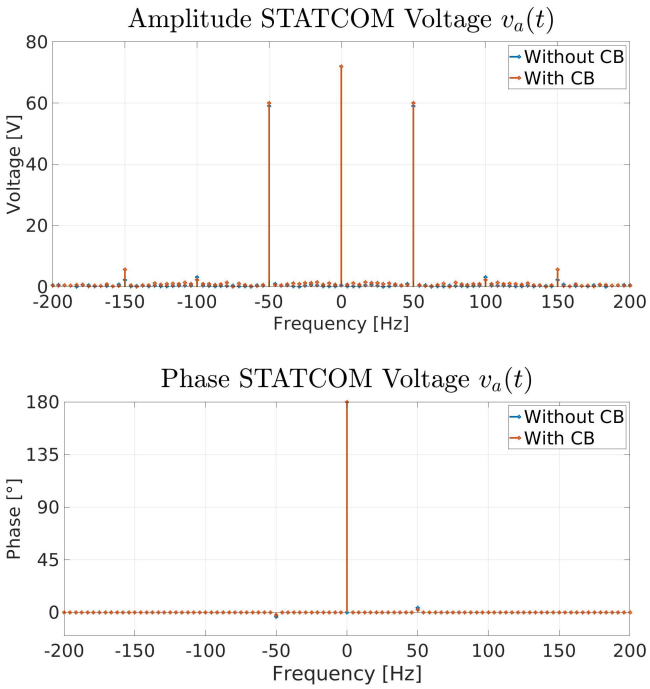


Figure 9.6: Fourier spectrum of the output voltage.

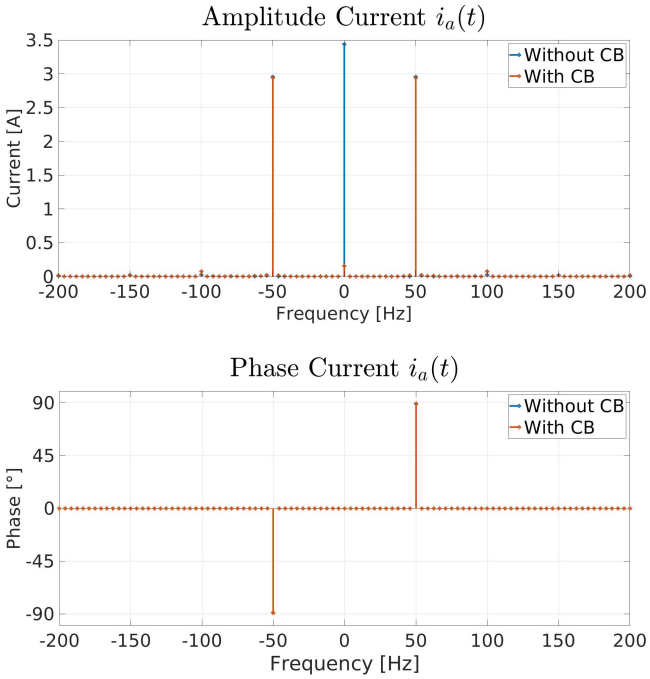


Figure 9.7: Fourier spectrum of output current.

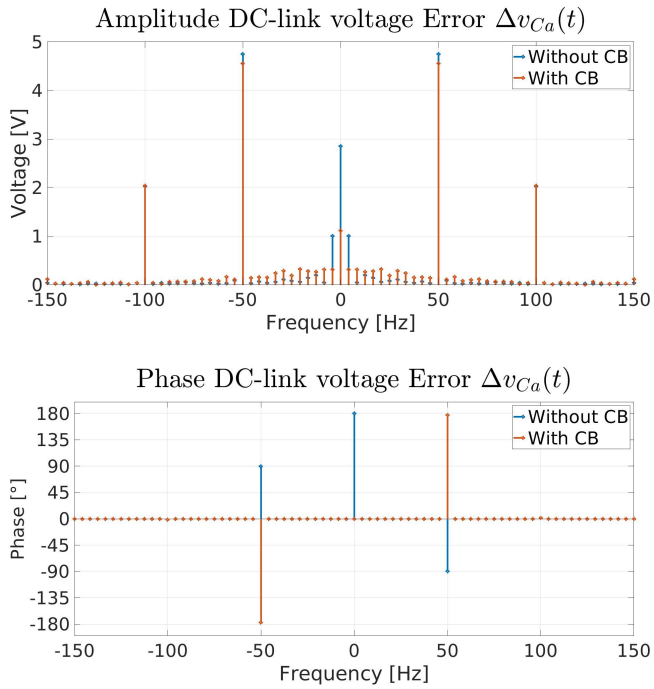


Figure 9.8: Fourier spectrum of DC-link voltage error.

Table 9.1: Fourier analysis of $v_{sa}(t)$, $i_a(t)$ and $\Delta v_{Ca}(t)$.

Without cluster balancing		With cluster balancing	
$v_{sa}(t)$ DC	0.49V	$v_{sa}(t)$ DC	-71.93V
$v_{sa}(t)$ 50Hz	118.08 $\angle 3.82°$V	$v_{sa}(t)$ 50Hz	120.1 $\angle 2.28°$V
$v_{sa}(t)$ 100Hz	6.22 $\angle 0°$V	$v_{sa}(t)$ 100Hz	4.54 $\angle 0°$V
$i_a(t)$ DC	3.44 A	$i_a(t)$ DC	0.156 A
$i_a(t)$ 50Hz	5.9 $\angle 89.09°$A	$i_a(t)$ 50Hz	5.88 $\angle 89.19°$
$i_a(t)$ 100Hz	0.06 $\angle 0°$A	$i_a(t)$ 100Hz	0.15 $\angle 0°$A
$\Delta v_{Ca}(t)$ DC	2.85V	$\Delta v_{Ca}(t)$ DC	$1,12$V
$\Delta v_{Ca}(t)$ 50Hz	9.5 $\angle -89.56°$V	$\Delta v_{Ca}(t)$ 50Hz	9.15 $\angle 177.42°$V
$\Delta v_{Ca}(t)$ 100Hz	4.04 $\angle 0°$V	$\Delta v_{Ca}(t)$ 100Hz	4.07 $\angle 0°$V

By substituting the numerical values extracted from the experiments and shown in Table 9.1 into the formulas presented in the theoretical discussion of Subsection 9.2.1, we can compare the values expected by the theory to the values actually measured as follows.

- Let us extract the current loss without the cluster balance. In this scenario, the cluster dynamics is given by (9.2).
  In steady-state condition, we assume that the cluster is balanced, which leads to the assumption that the constant part of $dv_{Ca}(t)/dt$ is zero.
  By definition, the switching function $S_a(t)$ is equal to the output voltage over the

nominal cluster DC-link voltage, i.e., $S_a(t) = v_{sa}(t) / (n \cdot V_{DC})$.
Hence, by setting the constant part of (9.2) equal to zero, the current loss can be computed as follows:

$$\frac{\tilde{S} \cdot \tilde{I}_S}{2} \cos(\phi_S - \phi_I) = -i_{la}(t) \Rightarrow \quad i_{la}(t) = \frac{\tilde{S} \cdot \tilde{I}_S}{2} \cos(\phi_S - \phi_I)$$

$$i_{la}(t) = \frac{\frac{118.08}{160} \times 5.9 \text{ A}}{2} \cos(3.82^\circ - 89.09^\circ) = 0.179 \text{ A}. \tag{9.7}$$

- The theoretical line frequency ripple of the cluster without the proposed control layer can be extracted from (9.2) as:

$$\begin{aligned} \frac{1}{C_{eq}} \int_0^t \tilde{S} \cdot \overline{I}_S \cos(\omega\tau + \phi_S)\, d\tau &= \frac{\tilde{S} \cdot \overline{I}_S}{\omega C_{eq}} \sin(\omega t + \phi_S) \\ &= \frac{\tilde{S} \cdot \overline{I}_S}{\omega C_{eq}} \cos\left(\omega t + \phi_S - \frac{\pi}{2}\right) \\ &= \frac{\frac{118.08}{160} \times 3.44 \text{ A}}{2 \times \pi \times 50 \times 0.9 \text{ } \mu\text{F}} \cos(\omega t + 3.82^\circ - 90^\circ) \\ &= 9 \cos(\omega t - 86.18^\circ) \text{ V}, \end{aligned} \tag{9.8}$$

  which is coherent with the measured 9.5 $\underline{/-89.56^\circ}$ V.

- Now, we want to study the compensated current loss when the cluster balancing control is introduced. It can be extracted by setting the constant part of the new $dv_{Ca}(t)/dt$ in (9.3) equal to zero and by substituting the values in Table 9.1, as follows:

$$\frac{\tilde{S} \cdot \tilde{I}_S}{2} \cos(\phi_S - \phi_I) + \overline{S} \cdot \overline{I}_S - i_{la}(t) = 0 \Rightarrow i_{la}(t) = \frac{\tilde{S} \cdot \tilde{I}_S}{2} \cos(\phi_S - \phi_I) + \overline{I}_S \cdot \overline{S}$$

$$i_{la}(t) = \frac{\frac{120.1}{160} \times 5.88 \text{ A}}{2} \cos(2.28^\circ - 89.19^\circ) - \frac{71.93}{160} \times 0.156 \text{ A} = 0.17 \text{ A}, \tag{9.9}$$

  where the theoretical loss current $i_{la}(t)$ that is coherent with the one computed in (9.7). Indeed in both scenarios, the DC current produced by the controller compensates the losses, forcing the cluster dynamics to be zero in steady-state, verifying the assumption in (9.5).

- The theoretical line frequency ripple on the cluster is extracted from (9.4), as follows:

$$\begin{aligned} \frac{1}{C_{eq}} \int_0^t \tilde{I}_C \sin(\omega\tau + \phi_{I_C})\, d\tau &= -\frac{\tilde{I}_C}{\omega C_{eq}} \cos(\omega t + \phi_{I_C}) \\ &= 9.34 \cos(\omega t + 179^\circ) \text{ V}, \end{aligned} \tag{9.10}$$

  which is, again, coherent with the experimentally measured 9.15 $\underline{/177.42^\circ}$ V.

- Finally, according to Table 9.1, the double line-frequency ripple in $\Delta v_{Ca}(t)$ does not change by introducing the cluster balancing control, as expected from the theoretical discussion.

Indeed, the Fourier analysis verifies the theoretical reasoning presented in Section 9.2.

9.3.2 Comparison with the Existing Method

The DC current elimination problem for CTs sensed CHB-STATCOM controlled by PI regulator was faced in [4]. The standard PI control strategy in [19] was implemented on the CHB-STATCOM prototype. To compare the proposed methodology with the PI strategy, the DC current elimination method in [4] was implemented and the steady-state, dynamic response and computational burden were analyzed. The parameters of the PI control were tuned based on [4] and are listed in Tab. 9.2. The switching frequency was set to 2 kHz to make a fair comparison with the MPC, which had 2 kHz average switching frequency.

Table 9.2: PI control parameters.

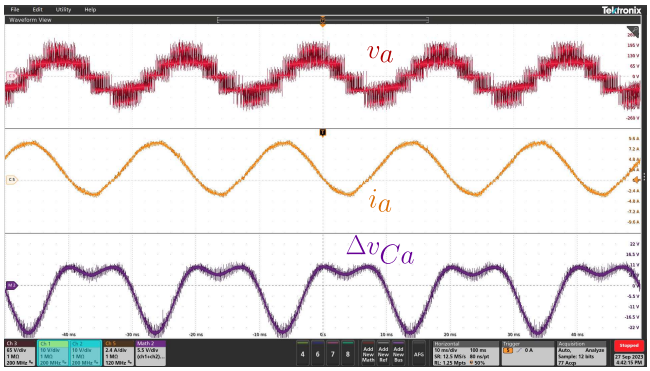
	Value		Value
K_P current control	1	K_I current control	100
K_P voltage control	1	K_I voltage control	0
K_P DC elimination	0.5	K_I DC elimination	10
Sampling time T_s	50 µs	Carrier frequency	2kHz

Steady-State Performance

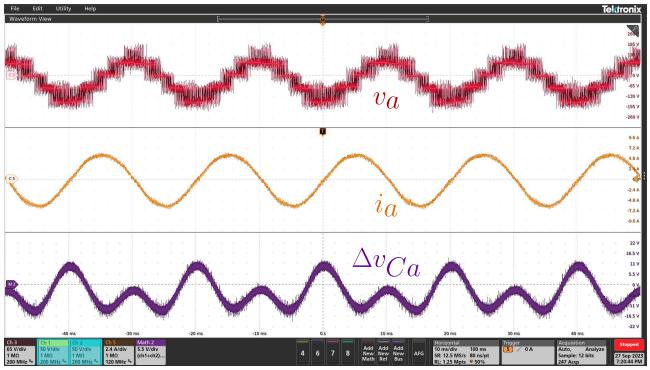
Fig. 9.9a shows the output voltage and current and the cluster ripple of phase a when controlling the CHB-STATCOM with a standard PI regulator. As expected, a DC current component appeared in the output current due to the use of CTs. Fig. 9.9b shows the same waveforms when adding the PI methodology in [4] to compensate the DC current. To analyze the results, Fourier analysis and THD were computed and the main parameters are summarized in Tab. 9.3.
Both the proposed method and [4] were able to suppress the DC current in the two different controllers by introducing a DC offset in the output voltage of the CHB-STATCOM. In the carried out test, the PI and the MPC experienced, respectively, a 2.73 A and a 3,44 A current offset that resulted in a 58.02 % and 47.78 % current THD. By introducing the DC current elimination methods, a DC offset was injected in the output voltage of, respectively, -75 V and -25 V, which resulted in a THD of 61.62 % and 39.17 % for the two controllers, underlying that the output voltage was more distorted for the MPC. Despite this, the DC current was kept to 0.156 A for the MPC, against the 0.23 A for the PI regulator, demonstrating a better DC current suppression in the MPC. Moreover, the THD of the output current was 4.12 % for the MPC and 6.06 % for the PI, which also confirmed the overall better performance of the MPC.

Dynamic Performance

To test the dynamic performance of the proposed method, the cluster balancing control was tested when a DC current was already present in the output current and compared with the PI method.
Fig. 9.10 shows the currents dynamics in a 10 s time window with a zoom before and after the activation of the DC elimination method for MPC (Figs. 9.10a, 9.10c) and PI (Figs. 9.10b, 9.10d).



(a) PI-controlled CHB-STATCOM without DC current elimination method: $v_{sa}\,(t)$ [65 V/div], $i_a\,(t)$ [2.4 A/div], $\Delta v_{Ca}\,(t)$ [5.5 V/div], time [10 ms/div].



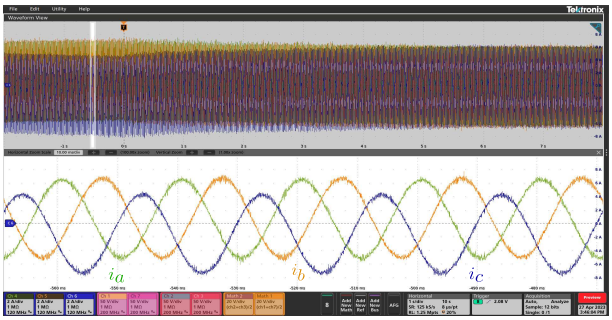
(b) PI-controlled CHB-STATCOM with DC current elimination method: $v_{sa}\,(t)$ [65 V/div], $i_a\,(t)$ [2.4 A/div], $\Delta v_{Ca}\,(t)$ [5.5 V/div], time [10 ms/div].

Figure 9.9: DC current elimination method for PI-controlled CHB-STATCOM.

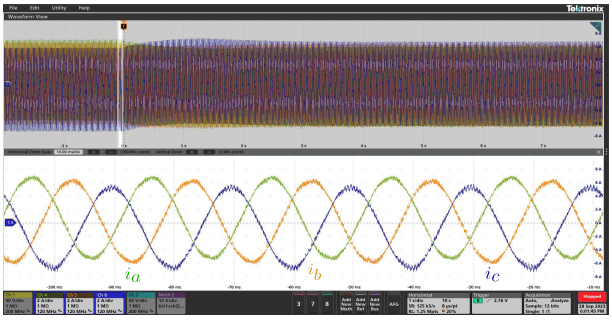
Table 9.3: Comparison between PI and MPC.

MPC without DC current elimination		PI without DC current elimination	
$v_{sa}\,(t)$ DC	0.49 V	$v_{sa}\,(t)$ DC	0.48 V
$v_{sa}\,(t)$ THD	8.79 %	$v_{sa}\,(t)$ THD	35.48 %
$i_a\,(t)$ DC	3.44 A	$i_a\,(t)$ DC	2.73 A
$i_a\,(t)$ THD	58.02 %	$i_a\,(t)$ THD	46.78 %

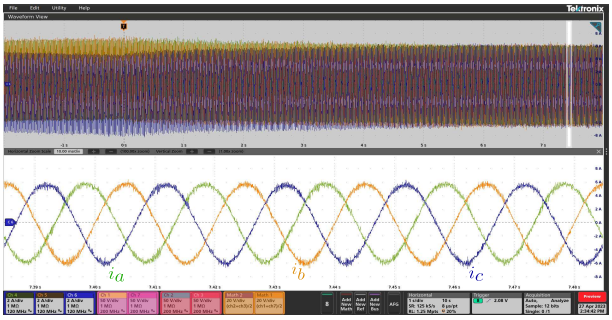
MPC with DC current elimination		PI with DC current elimination	
$v_{sa}\,(t)$ DC	−71.93 V	$v_{sa}\,(t)$ DC	−25 V
$v_{sa}\,(t)$ THD	61.62 %	$v_{sa}\,(t)$ THD	39.17 %
$i_a\,(t)$ DC	0.156 A	$i_a\,(t)$ DC	0.23 A
$i_a\,(t)$ THD	4.12 %	$i_a\,(t)$ THD	6.06 %



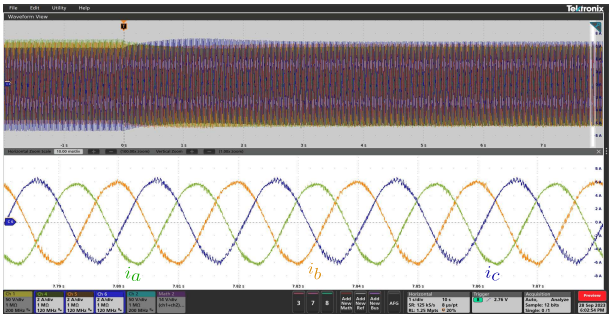
(a) MPC without clusters balance (i_a, i_b, i_c [2 A/div], time [1 s/div] on the top and [10 ms/div] on the bottom of the figure).



(b) PI without DC elimination (i_a, i_b, i_c [2 A/div], time [1 s/div] on the top and [10 ms/div] on the bottom of the figure).



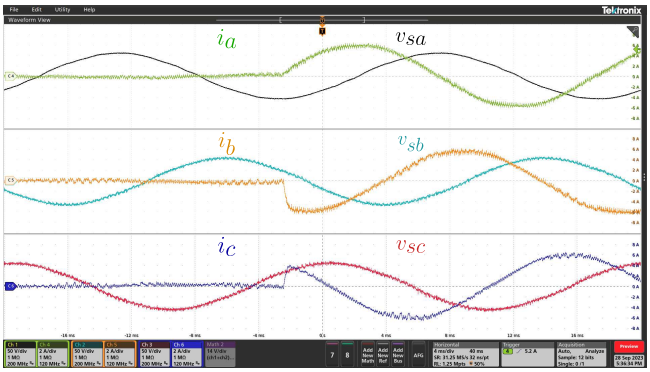
(c) MPC with clusters balance (i_a, i_b, i_c [2 A/div], time [1 s/div] on the top and [10 ms/div] on the bottom of the figure).



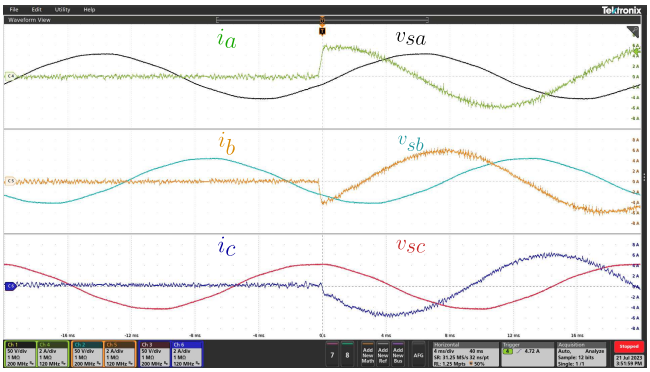
(d) PI with DC elimination (i_a, i_b, i_c [2 A/div], time [1 s/div] on the top and [10 ms/div] on the bottom of the figure).

Figure 9.10: CHB-STATCOM currents dynamics: before and after DC elimination.

Fig. 9.10a shows the MPC currents when the cluster balancing control was inactive and a DC component appeared in the output currents of, respectively, 1.22 A, 1.2 A and -2.54 A for the phases a, b and c. At time 0 s, the cluster balancing control was activated and the DC component was gradually reduced and essentially eliminated at time 5 s. Fig. 9.10c shows the output currents once they reached the steady-state. The DC current in the three phases were reduced to 0.12 A, 0.04 A and -0.18 A respectively, which experimentally confirmed the validity of the proposed approach.
The same test was carried out for the PI regulator and Fig. 9.10b shows the three-phase current for a standard PI control with a DC offset of 0.7 A, 0.25 A and -0.84 A for the phases a, b and c. At time 0 s, the DC current elimination method was activated and,



(a) Dynamic response of PI control: v_a, v_b, v_c [50 V/div], i_a, i_b, i_c [2 A/div], time [4 ms/div].



(b) Dynamic response of MPC: v_a, v_b, v_c [50 V/div], i_a, i_b, i_c [2 A/div], time [4 ms/div].

Figure 9.11: Dynamic response of PI and MPC.

after a 3 s transient, the offset was reduced to -0.37 A, -0.13 A and 0.33 A for phases a, b and c, as shown in Fig. 9.10d. It is possible to see an overshoot, typical of PI regulators, before the DC current was suppressed. It turned out that, with the use of the proposed method, the MPC had a better DC current suppression capability with respect to the PI control and, hence, a better current THD.
Fig. 9.11 shows the step response of the two controllers for a 4 A RMS quadrature current

reference. Both the DC current elimination methods did not substantially affect the step performance of the controllers while suppressing the undesired DC component.

Computational Burden

The proposed cluster balancing algorithm in Fig. 9.4 was implemented on FPGA by fully exploiting the parallelism capability on the algorithm using 3 accumulators and 3 multipliers, as described in Subsection 7.1.2. By taking into account the possibility to compute up to three sums and multiplications in parallel and considering the required $2n+1$ iterations, the time spent by the algorithm was about 0.7 µs.
The PI DC current elimination required computing a, b, c to d, q transformation, moving average filter and PI regulator for each phase, which resulted in 1.12 µs with the same number of accumulators and multipliers. Hence, both methods had minor influence on the overall computational burden. The current control of the FCS-MPC was implemented by using the method described in Chapter 5, while the individual voltage balance algorithm was computed as described in Subsection 7.1.3. The overall time spent for the FCS-MPC was 12.12 µs, which was compatible with the 11.4 µs spent by the overall PI control.

9.4 Conclusions

This Chapter addresses the problem of eliminating the DC current in the CHB-STATCOM output terminal when we are controlling the system through FCS-MPC and sensing the currents by using current transformer sensors.
This Chapter demonstrates that, by adding the cluster voltage balance layer in the CHB-STATCOM FCS-MPC, we can reduce the voltage imbalance among the clusters and eliminate the DC current injected by the CHB-STATCOM.
Experimental validation of the proposed solution were carried out with a 5-level CHB-STATCOM employing CTs.
A comparison with a standard PI approach is given in terms of steady-state and dynamic operations and computational burden, demonstrating the superior performance of the proposed technique.

References

[1] A. M. Saif, C. Buccella, V. Patel, M. Tinari, and C. Cecati, "Design and cost analysis for statcom in low and medium voltage systems," in *IECON 2018 - 44th Annual Conference of the IEEE Industrial Electronics Society*, 2018, pp. 3938–3943.

[2] S. Mohamadian, H. Pairo, and A. Ghasemian, "A straightforward quadrature signal generator for single-phase sogi-pll with low susceptibility to grid harmonics," *IEEE Transactions on Industrial Electronics*, vol. 69, no. 7, pp. 6997–7007, 2021.

[3] B. Long, M. Zhang, Y. Liao, L. Huang, and K. T. Chong, "An overview of dc component generation, detection and suppression for grid-connected converter systems," *IEEE Access*, vol. 7, pp. 110 426–110 438, 2019.

[4] Y. Shi, B. Liu, and S. Duan, "Eliminating dc current injection in current-transformer-sensed statcoms," *IEEE Transactions on Power Electronics*, vol. 28, no. 8, pp. 3760–3767, 2012.

[5] M. Armstrong, D. J. Atkinson, C. M. Johnson, and T. D. Abeyasekera, "Auto-calibrating dc link current sensing technique for transformerless, grid connected, h-bridge inverter systems," *IEEE Transactions on Power Electronics*, vol. 21, no. 5, pp. 1385–1393, 2006.

[6] W. Zhang, M. Armstrong, and M. A. Elgendy, "Mitigation of dc current injection in transformer-less grid-connected inverters using a voltage filtering dc extraction approach," *IEEE Transactions on Energy Conversion*, vol. 34, no. 1, pp. 426–434, 2019.

[7] G. Qiu, J. Liao, B. Wu, and Z. Shi, "Suppressing dc current injection in transformerless grid-connected inverter using a customized current sensor," *IEEE Transactions on Power Electronics*, vol. 36, no. 10, pp. 11 003–11 008, 2021.

[8] M. Chen, D. Xu, T. Zhang, K. Shi, G. He, and K. Rajashekara, "A novel dc current injection suppression method for three-phase grid-connected inverter without the isolation transformer," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 11, pp. 8656–8666, 2018.

[9] B. Guo, M. Su, Y. Sun, H. Wang, X. Li, Y. Liao, J. Lin, and Y. Yang, "Cost-effective dc current suppression for single-phase grid-connected pv inverter," *IEEE Journal of Emerging and Selected Topics in Power Electronics*, vol. 9, no. 2, pp. 1808–1823, 2021.

[10] Q. Xiao, H. Jia, Y. Tang, Y. Jin, Y. Mu, R. Teodorescu, and F. Blaabjerg, "Dual-layer modulated model predictive control scheme for the cascaded h-bridge converter," *IEEE Transactions on Industrial Electronics*, vol. 70, no. 10, pp. 9751–9763, 2023.

[11] X. Pan, L. Zhang, Y. Li, K. Li, and H. Huang, "Modulated model predictive control with branch and band scheme for unbalanced load compensation by mmcc-statcom," *IEEE Transactions on Power Electronics*, vol. 37, no. 8, pp. 8948–8962, 2022.

[12] W. Zhang, M. Armstrong, and M. A. Elgendy, "Dc injection suppression in transformer-less grid-connected inverter using a dc-link current sensing and active control approach," *IEEE Transactions on Energy Conversion*, vol. 34, no. 1, pp. 396–404, 2019.

[13] S. N. Vukosavió and L. S. Periċ, "High-precision active suppression of dc bias in ac grids by grid-connected power converters," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 1, pp. 857–865, 2017.

[14] Q. Yan, X. Wu, X. Yuan, Y. Geng, and Q. Zhang, "Minimization of the dc component in transformerless three-phase grid-connected photovoltaic inverters," *IEEE Transactions on Power Electronics*, vol. 30, no. 7, pp. 3984–3997, 2015.

[15] P. Ripka, K. Draxler, and R. Styblikova, "Measurement of dc currents in the power grid by current transformer," *IEEE Transactions on Magnetics*, vol. 49, no. 1, pp. 73–76, 2013.

[16] 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.

[17] P. Cortés, S. Kouro, B. La Rocca, R. Vargas, J. Rodriguez, J. I. Leon, S. Vazquez, and L. G. Franquelo, "Guidelines for weighting factors design in model predictive control of power converters and drives," in *IEEE ICIT 2009*, 2009, pp. 1–7.

[18] D. Shetty and N. Prabhu, "Ziegler nichols method based robust reactive current controller for statcom," *Energy Procedia*, vol. 117, pp. 543–550, 2017.

[19] H. Akagi, S. Inoue, and T. Yoshii, "Control and performance of a transformerless cascade pwm statcom with star configuration," *IEEE Transactions on Industry Application*, vol. 43, no. 4, pp. 1041–1049, 2007.

[20] B. Long, M. Zhang, Y. Liao, L. Huang, and K. T. Chong, "An overview of dc component generation, detection and suppression for grid-connected converter systems," *IEEE Access*, vol. 7, pp. 110 426–110 438, 2019.

[21] F. Simonetti, S. Mohamadian, C. Buccella, and C. Cecati, "Dc current elimination for mpc for chb-statcom employing current transformer sensors," in *2023 IEEE Energy Conversion Congress and Exposition (ECCE)*, 2023, pp. 2831–2836.

[22] ——, "Dc current suppression in chb-statcom with model predictive control employing current transformers," *IEEE Transactions on Industrial Electronics*, pp. 1–11, 2023.

Chapter 10

Communication Delays FPGA-DSP

In the final setup, the current transformer sensors were avoided and the LEM (LEM GO20SMS) sensors embedded on the H-bridges were used to sense the currents. As mentioned in Section 8.1, each H-bridge is equipped with a measurement circuit to sense the input DC-link voltage and a measurement circuitry to sense the current, which are wired to two ADC converters built into the DSP. The DSP is responsible for acquiring the analog signals, converting them to digital signals and sending them to the master FPGA via an SPI communication.
However, several problems related to the transmission of the measurements to the master board were encountered, which resulted in delays in the communication and, thus, a significant deterioration of the controller performance.
This Chapter discusses the synchronization problems in the FPGA-DSP communication, showing the impact on the output waveforms and it presents the developed solutions.

10.1 Introduction

DSPs are used in almost all power converters because they provide analog to digital conversion of sensor measurements (e.g., current, voltage), execution of the control algorithm, implementation of the pulse width modulation (PWM), communication with other devices and so on. On the other hand, FPGAs are very powerful solutions in highly demanding systems with many digital output signals, such as modular multilevel converters, and-or requiring the execution of the control task in extremely short times [6, 11, 12, 10]. In fact, the possibility to design ad-hoc hardware architectures and the intrinsic FPGA parallelism capability, make it an effective platform for complex control systems.
In complex topologies or to implement complex control algorithms, power converters use both DSPs and FPGAs, as in [7]. In this case, a communication strategy must be properly designed to limit the delay between the the acquisition of the physical data by the DSP and their transmission to the master controller, i.e., the FPGA. In [3], a distributed PWM strategy was implemented for a CHB inverter. Each H-bridge comprises a DSP and communicates with a master DSP via SPI protocol in a networked control system.

In [4], multiple DSPs and FPGA boards are employed to control a CHB converter for FACTS applications. The system uses optical fibers for high-speed communications and conventional serial communication where a relatively low speed is acceptable. In [5], a control based on DSP and FPGA with high-speed communication interfaces is applied for grid-tied converters for distributed power generation systems. The communications between DSP and FPGA were implemented using an external memory interface.
On the other hand, MPC is one of the most computationally expensive advanced control techniques and it takes significant advantage of the FPGA high performance and parallelism capability. In [1, 9, 2], authors implement FCS-MPC on a CHB-STATCOM with reduced computations and employing both DSP and FPGA. However no details on inter-processor communications are given.
This Chapter focuses on implementation issues of the signal acquisition through the DSP and the successive transmission of the data to the FPGA via SPI in a the CHB-STATCOM controlled through FCS-MPC. These problems lead to communication delays that affect the accuracy of the predictions and strongly negatively affect the output waveforms. This Chapter considers four different scenarios extracted from practical experience and presents the effects of communication delays on the output waveforms. The discussion starts from the worst scenario and, solving specific issues step by step, it reaches the last scenario, where a practical solution for minimizing the delays is described.

10.2 FPGA-DSP Sampling and Communication

The SPI module on the DSP (slave device in this setup) equips a 16 bit transmission data register (SPIDAT-DSP), in which the DSP can write the data to sent.
The less significant bit (LSB) of SPIDAT-DSP is connected to the SPI-MISO (SPI master input slave output) pin of the master FPGA, which is wired to the most significant bit (MSB) of the data register inside the master SPI module on the FPGA (SPIDAT-FPGA). On the other side, the MSB of SPIDAT-DSP is connected to the SPI-MOSI (SPI master output slave input) pin of the master FPGA, which is wired to the LSB of the receiving register inside the master SPI module on the FPGA (SPIDAT-FPGA). This way, the SPI communication protocol creates a circular bus between the FPGA and the DSP to share the data contained in the two data registers.
The master SPI module also provides a clock signal to the SPI slave through the SPI-CLK pin, to synchronize the transmission of the data: in this design, every bit of the data is transmitted at the rising edge of the SPI-CLK.
Since the SPI protocol allows having multiple slave peripherals, a chip select signal (SPI-CS) is asserted by the master module to select the proper slave among the others. In this setup, one SPI master module is instantiated within the FPGA for each SPI slave module on the DSPs. Therefore, the SPI-CS is only used to enable the communication between the master and the slave modules.
The SPI module on the DSP also provides a transmission FIFO (first in first out) buffer (TX-FIFO) and a transmission buffer register (SPITXBUF). When enabled, the DSP can write multiple data in the TX-FIFO. The first data of the TX-FIFO is transmitted to the SPITXBUF and, if the SPIDAT-DSP is empty, the
data is transmitted to the SPIDAT-DSP. The use of the transmission buffer allows the DSP to acquire more data, that will be forward one by one to the master only when it

starts the next communication. Moreover, despite the SPI modules on the DSP have 16 bit registers, the DSP provides the possibility to establish a 32 bit communication by employing the TX-FIFO.
Two ADCs built into the DSP are used to acquire the input voltage and output current of the H-bridge, which perform a 16 bit conversion and they store the data into two registers. Fig. 10.1 shows the FPGA-DSP communication scheme.

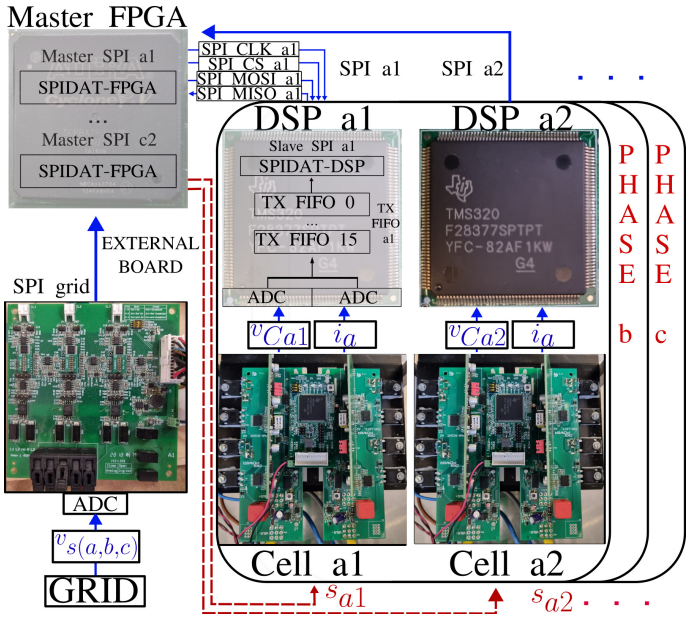


Figure 10.1: FPGA-DSP communication schematic of the experimental setup.

10.2.1 Scenario 1: Continuous Sampling

The SPI master module on the FPGA was implemented by using the 32 bit Altera SPI core intellectual property (IP) [8]. The DSP was programmed to operate the two ADCs in continuous sampling mode.
The SPI slave module in the DSP employed the TX-FIFO to establish a 32 bit transmission to send the two 16 bit data (voltage and current) in a single SPI communication.
The DSP continuously wrote the two data in the TX-FIFO by performing two writing processes and the data was sent by the SPI slave module every time the SPI master module on the FPGA started a new communication.
Due to the continuous sampling mode, the TX-FIFO was always full. When the FPGA asked for the data, the DSP transmitted the last value stored in the buffer, which it turned out to be actually sampled more than one control interval ahead.
The measurements acquired by the FPGA had a delay which affects the predictions computed by the controller. The inaccuracy of the predictions impacted the control performance, resulting in distorted output currents and unbalanced DC-link voltages, as shown in Fig. 10.2.

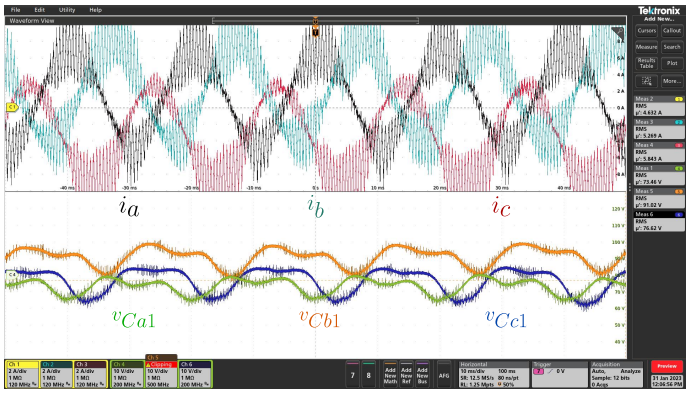


Figure 10.2: Scenario 1: i_a, i_b, i_c [2 A/div], $v_{Ca1}, v_{Cb1}, v_{Cc1}$ [10 V/div], time [10 ms/div].

10.2.2 Scenario 2: One Sample within each Control Interval

Since the continuous sampling and writing in the SPI buffer was the main cause of the delay, a first improvement was introduced by using a one shot sampling and storing. At the beginning of the control interval, the FPGA started the SPI communication.
To reduce the time between the sample and transmission, after the transmission was finished, the DSP waited a certain amount of time before sampling the new data, as underlined by the timeline in Fig. 10.3.

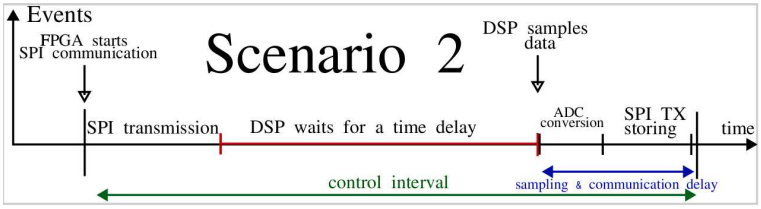


Figure 10.3: Sampling and communications strategies in scenario 2.

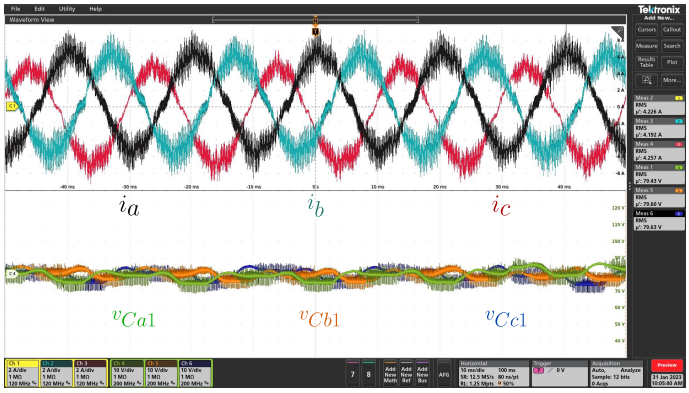


Figure 10.4: Scenario 2: i_a, i_b, i_c [2 A/div], $v_{Ca1}, v_{Cb1}, v_{Cc1}$ [10 V/div], time [10 ms/div].

This way, the time between the sampling instant and when the data is transmitted to the FPGA was reduced by opportunely tuning the delay. Fig. 10.4 shows the output waveforms. Despite the improvement, spikes were still presents. In fact, this communication was still affected by one control interval delay caused by the buffer.

10.2.3 Scenario 3: Elimination of the SPI Buffer

To reduce the delay, the use of the SPI buffer was avoided. Without the buffer, a 32 bit data transmission was no more possible because of the 16 bit SPI data register. Hence, the 32 bit communication was divided into two distinct 16 bit words. Furthermore, the communication scheme was revised to minimize the time between sampling and transmission of the data, as shown in Fig. 10.5.

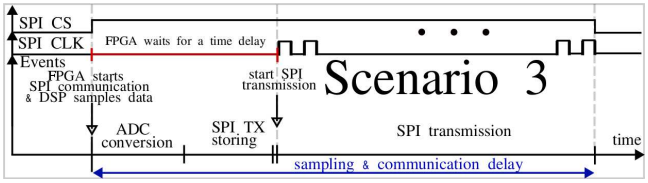


Figure 10.5: Sampling and communications strategies in scenario 3.

The start of conversion signal of the ADC on the DSP, which was previously asserted via software, was directly connected to the SPI-CS of the master FPGA. Therefore, when the FPGA asserted the SPI-CS, which was send to the SPI salve module, starting the SPI communication, it also started the ADC conversion with the same signal.
When the conversion was finished, the ADC asserted an interrupt to the DSP. The called ISR stored the sampled data in the SPIDAT-DSP.
On the FPGA side, the Altera SPI master module was configured to add a delay between the SPI-CS the SPI-CLK to wait for the DSP to correctly convert and store the result in the SPIDAT-DSP.

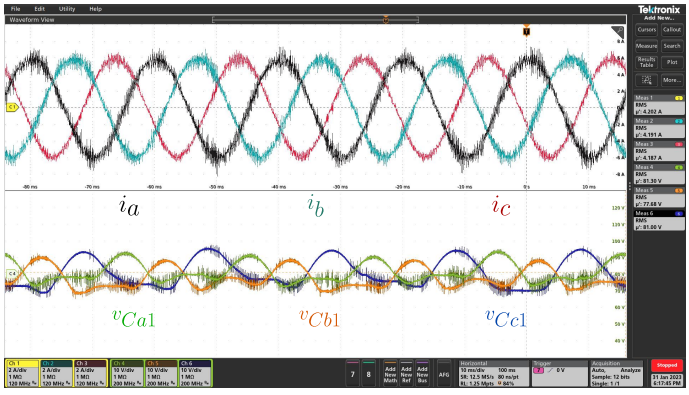


Figure 10.6: Scenario 3: i_a, i_b, i_c [2 A/div], $v_{Ca1}, v_{Cb1}, v_{Cc1}$ [10 V/div], time [10 ms/div].

Then, the FPGA sent the clock pulses through the SPI-CLK signal and the measured data was transmitted from the DSP to the FPGA.
Two SPI communications had to be established to transmit the two data, i.e., voltage and current measurements. Nios II was programmed to write a specific value inside the SPIDAT-FPGA before starting each communication. The value was sent during the next SPI transmission and it was used as an identifier by the DSP, which sent the measured current or the measured voltage at the next SPI communication depending on the specific value.
This strategy solved the delay introduced by the buffer and Fig. 10.6 underlines the performance improvements of the output waveforms.
However, the double 16-bit communication introduced an overhead in the transmission delay compared to a single 32-bit communication. Two 16-bit communications took about 9 µs, while the 32-bit one was about 6 µs.
Moreover, it turned out that the use of the Altera SPI core IP in the FPGA led to a short delay among the different communications. In fact, the transmission of the provided SPI module could only be started via software from Nios II. Hence, there was a single line of code to execute to start the communication for each of them, which started the transmission sequentially, causing a short delay between the previous communication and the next one, of about 600ns. Therefore, also the sampling instants of the different currents and voltages were a slightly different.
For instance, phase a and phase b were not sampled simultaneously, and the delay between the first and sixth communication was about 3 µs, which further increases if the number of H-bridges grows.

10.2.4 Scenario 4: Parallel SPI Communications

To eliminate the delay between the SPI communication, the best solution was to start all the SPI transmission simultaneously. Since the Altera SPI core IP only allows the start the communication via SW, it is impossible to avoid the delay of the sequential executions of the Nios II instructions. Hence, during this work, an ad-hoc SPI master module was designed and, then, described in VHDL to replace the Altera SPI IP.

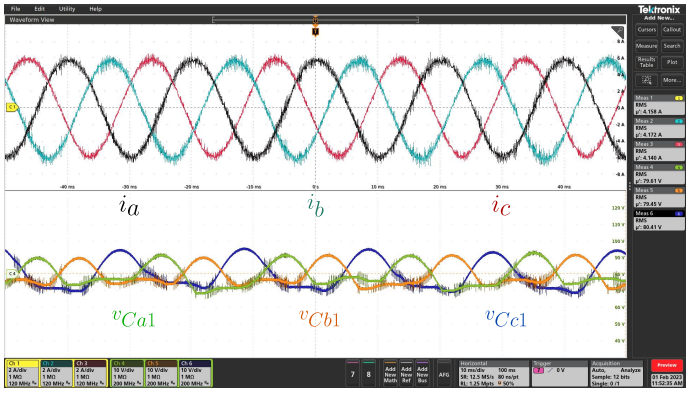


Figure 10.7: Scenario 4: i_a, i_b, i_c [2 A/div], $v_{Ca1}, v_{Cb1}, v_{Cc1}$ [10 V/div], time [10 ms/div].

The main extra feature of the designed module was the possibility to start the communication through a start operation signal, in the place of the SW instruction.
The i-th SPI master module had its own start of communication (SoC_i) signal and all the SoC_i signals were connected together to a unique SoC signal. This way, when the SoC was asserted, the SPI communications started concurrently. Hence, also the sampling from the DSP side were started simultaneously and transmitted in the same time window. The improved communication took about 6 µs and Fig. 10.7 shows the obtained performance.

10.3 Conclusions

This Chapter discusses the practical issues faced on the ADC sampling and the successive FPGA-DSP transmission via SPI communication protocol of the data, i.e., the currents and voltages measurements.
The effects of the communication delays on the MPC performance of the CHB-STATCOM are discussed and the practical solution found to minimize the delays is presented.

References

[1] Y. Zhang, X. Yuan, X. Wu, Y. Yuan, and J. Zhou, "Parallel implementation of model predictive control for multilevel cascaded h-bridge statcom with linear complexity," *IEEE Transactions on Industrial Electronics*, vol. 67, no. 2, pp. 832–841, 2020.

[2] F. Simonetti, A. D'Innocenzo, and C. Cecati, "Neural network model predictive control for chb converters with fpga implementation," *IEEE Transactions on Industrial Informatics*, pp. 1–12, 2023.

[3] P. C. Loh, D. G. Holmes, and T. A. Lipo, "Implementation and control of distributed pwm cascaded multilevel inverters with minimal harmonic distortion and common-mode voltage," *IEEE Transactions on Power Electronics*, vol. 20, no. 1, pp. 90–99, 2005.

[4] T. Atalik, M. Deniz, E. Koc, C. Ö. Gercek, B. Gultekin, M. Ermis, and I. Cadirci, "Multi-dsp and -fpga-based fully digital control system for cascaded multilevel converters used in facts applications," *IEEE Transactions on Industrial nformatics*, vol. 8, no. 3, pp. 511–527, 2012.

[5] E. J. Bueno, A. Hernandez, F. J. Rodriguez, C. Giron, R. Mateos, and S. Cobreces, "A dsp- and fpga-based industrial control with high-speed communication interfaces for grid converters applied to distributed power generation systems," *IEEE Transactions on Industrial Electronics*, vol. 56, no. 3, pp. 654–669, 2009.

[6] C. Buccella, C. Cecati, and H. Latafat, "Digital control of power converters - a survey," *IEEE Transactions on Industrial Informatics*, vol. 8, no. 3, pp. 437–447, 2012.

[7] C. Buccella, M. G. Cimoroni, and C. Cecati, "Mitigation technique for cascaded h-bridge multilevel inverters based on pulse active width modulation," *IEEE Journal of Emerging and Selected Topics in Power Electronics*, vol. 11, no. 1, pp. 999–1008, 2023.

[8] "5. SPI Core — intel.com." https://www.intel.com/content/www/us/en/docs/programmable/683130/21-4/spi-core.html. [Accessed 10-May-2023].

[9] F. Simonetti, G. D. Di Girolamo, A. D'Innocenzo, and C. Cecati, "Machine learning for model predictive control of cascaded h-bridge inverters," in *2022 IEEE 21st Mediterranean Electrotechnical Conference (MELECON)*, pp. 1241–1246, 2022.

[10] S. Mohamadian, H. Pairo, and A. Ghasemian, "A straightforward quadrature signal generator for single-phase sogi-pll with low susceptibility to grid harmonics," *IEEE Transactions on Industrial Electronics*, vol. 69, no. 7, pp. 6997–7007, 2022.

[11] R. Di Fonso, P. Bharadwaj, R. Teodorescu, and C. Cecati, "Internal resistance estimation of li-ion batteries using wavelet analysis," in *2022 IEEE 13th International Symposium on Power Electronics for Distributed Generation Systems (PEDG)*, pp. 1–5, 2022.

[12] M. Dezhbord, P. Mohseni, S. H. Hosseini, D. Mirabbasi, and M. R. Islam, "A high step-up three-port dc-dc converter with reduced voltage stress for hybrid energy systems," *IEEE Journal of Emerging and Selected Topics in Industrial Electronics*, vol. 3, no. 4, pp. 998–1009, 2022.

Chapter 11

Contributions and Conclusions

11.1 Contributions

The research work presented in this thesis provides the following main contributions.

- The application of a shallow neural network for approximating the finite control set model predictive control (FCS-MPC) of a cascaded h-bridge (CHB) inverter was investigated. The obtained neural network model predictive control (NN-MPC) is able to embed the computations of the optimization problem solved online and can be used in place of the original controller in run-time implementation. The performance of the NN-MPC was deeply investigated in Matlab/Simulink environment for different numbers of levels of the converter, weighting coefficients in the cost function, numbers of hidden neurons and prediction horizons. It turned out that the neural network approach can effectively approximate the optimal control law and guarantee compatible performances. The computational complexity of the proposed approach was studied and the algorithm was implemented on a field programmable gate array (FPGA) platform to study the actual time spent in computations. The register transfer level (RTL) implementation was detailed described and an hardware in the loop (HIL) was created by using the FPGA and the CHB static synchronous compensator (CHB-STATCOM) simulated on Simulink to test the impact of the computations delay on the performance on the closed-loop controllers. The short computation delay of the NN-MPC resulted in superior performance with respect to the standard FCS-MPC.

- An analytical method for a simple explicit solution of the FCS-MPC for a CHB inverter was developed. The number of computations needed for the algorithm were derived and compared with state-of-the-art methodologies. The proposed approach requires few computations irrespective of the number of levels of the converter, overcoming the existing techniques. Moreover, it computes the global optimal solution without introducing approximation errors. The proposed method was implemented on a 5-level CHB-STATCOM prototype and the controller performance was deeply studied for different operative conditions, demonstrating the effectiveness of the proposed controller.

- A method to eliminate the DC current injected by a CHB-STATCOM employing current transformer sensors (CTs) was derived in the FCS-MPC framework. The proposed technique can effectively suppress the undesired DC current that appears in the output terminal due to the use of CTs. This problem was never faced for CHB-STATCOM driven by FCS-MPC. The proposed algorithm is simple to implement, computationally efficient and does not require extra hardware. The method was tested on a 5-level CHB-STATCOM prototype for steady-state and transient operations. Its effectiveness was analyzed and compared with the standard PI approach, demonstrating its superior performances.

- Implementation problems related to the sampling of voltages and currents by the digital signal processors (DSPs) on the H-bridges and their transmission to the FPGA via serial peripheral interface (SPI) protocol were summarized. The effect of the sampling and communication delays on the output currents were described and practical solutions to minimize the delay were given, improving and optimizing the controller performance.

11.2 Conclusions

This thesis presented two methods to reduce the computational burden of FCS-MPC for CHB inverters.
The first method consists on training a shallow NN to approximate the FCS-MPC and to use the obtained nonlinear function at run-time, in the place of the original controller. The obtained NN-MPC requires few computations compared to solving the optimization problem at run-time, which does not depend on the number of levels. The control was tested on a CHB-STATCOM in Matlab/Simulink environment to analyze the performance of the proposed controller in comparison to the standard FCS-MPC. The NN was tested for different numbers of levels, weighting parameters of the cost function, numbers of hidden neurons and prediction horizons. The NN well follows the optimal behavior in all cases and this deep analysis proved that the proposed NN-MPC is a general approach to control CHB inverters.
The standard and the proposed algorithm were implemented on FPGA and the time spent by the computations of the controllers was derived, demonstrating the extremely low computational burden of the NN-MPC compared to the standard appraoch. HIL simulations were carried out to analyze the impact of the computations delays on the controllers, demonstrating the superior performance of the proposed NN-MPC compared to the standard FCS-MPC.
The second method is an analytical strategy for a simple explicit solution to the FCS-MPC problem of CHB inverters. The computational cost analysis was carried out and the number of individual operations was computed. The proposed method was compared with two state-of-the-art approaches. The method was tested using a 5-level CHB-STATCOM prototype. The computational time for the control algorithm was determined and the hardware resources utilization on FPGA was presented. The proposed method overcomes the existing ones since it needs a few simple computations irrespective of the number of levels of the converter and it computes the global optimum solution.
The use of CTs in the closed-loop control of a CHB-STATCOM leads to an undesired DC output current. The problem was already addressed in the literature for linear controllers

but it was never faced for FCS-MPC. This thesis proposed a method to eliminate the DC output current by adding a third optimization layer in the overall FCS-MPC optimization problem. Experimental validation of the proposed solution was carried out on a 5-level CHB-STATCOM employing CTs. A comparison with a standard PI approach was given in terms of steady-state, dynamic performance, and computational burden, demonstrating the superior performance of the proposed technique.
Finally, this thesis discussed the practical issues faced in the sampling of the measurements by the DSP and the DSP to FPGA data transmission via SPI. The effects of the introduced delays on the performances of the FCS-MPC for CHB-STATCOM were shown and practical solutions to minimize the delays were described in order to optimize the performance of the controller.

List of Publications

List of Journal Papers

[1] F. Simonetti, A. D'Innocenzo, and C. Cecati, "Neural network model predictive control for chb converters with fpga implementation," *IEEE Transactions on Industrial Informatics*, pp. 1–12, 2023.

[2] S. Mohamadian, M. Modarres, F. Simonetti, and C. Cecati, "Modeling of switching power losses in cascaded h-bridges with unipolar pwm," *IEEE Journal of Emerging and Selected Topics in Power Electronics*, vol. 11, no. 3, pp. 3270–3280, 2023.

[3] F. Simonetti, A. D'Innocenzo, and C. Cecati, "Simple explicit solution of finite control set model predictive control for cascaded h-bridge inverters," *IEEE Transactions on Industrial Electronics*, pp. 1–11, 2023.

[4] F. Simonetti, S. Mohamadian, C. Buccella, and C. Cecati, "Dc current suppression in chb-statcom with model predictive control employing current transformers," *IEEE Transactions on Industrial Electronics*, pp. 1–11, 2023.

List of Conference Papers

[5] F. Simonetti, G. D. Girolamo, A. D'Innocenzo, and C. Cecati, "A neural network approach for efficient finite control set mpc of cascaded h-bridge statcom," in *IECON 2021 – 47th Annual Conference of the IEEE Industrial Electronics Society*, pp. 1–6, 2021.

[6] F. Simonetti, G. D. Di Girolamo, A. D'Innocenzo, and C. Cecati, "Machine learning for model predictive control of cascaded h-bridge inverters," in *2022 IEEE 21st Mediterranean Electrotechnical Conference (MELECON)*, pp. 1241–1246, 2022.

[7] C. Buccella, M. G. Cimoroni, F. Simonetti, and C. Cecati, "Selective harmonic mitigation-pulse amplitude modulation technique for 7-level inverters," in *2023 IEEE 32nd International Symposium on Industrial Electronics (ISIE)*, pp. 1–6, IEEE, 2023.

[8] S. Mohamadian, F. Simonetti, M. Dezhbord, C. Buccella, and C. Cecati, "Accurate modeling of switching losses in h-bridges with unipolar or bipolar pwm," in *IECON 2023- 49th Annual Conference of the IEEE Industrial Electronics Society*, pp. 1–6, 2023.

[9] M. Dezhbord, F. Simonetti, R. Di Fonso, S. Mohamadian, C. Buccella, and C. Cecati, "A soft-switched step-up converter topology with ics capability and lowered npiv," in *IECON 2023- 49th Annual Conference of the IEEE Industrial Electronics Society*, pp. 1–6, 2023.

[10] F. Simonetti, S. Mohamadian, C. Buccella, and C. Cecati, "Dc current elimination for mpc for chb-statcom employing current transformer sensors," in *2023 IEEE Energy Conversion Congress and Exposition (ECCE)*, pp. 2831–2836, 2023.

[11] G. Bossi, C. Buccella, C. Cecati, A. Damiano, A. Floris, and F. Simonetti, "A converter topology for auxiliary power system of an all electric aircraft," in *2023 IEEE Energy Conversion Congress and Exposition (ECCE)*, pp. 1605–1612, 2023.

Appendix

Clarke Transformation

The Clarke transformation is used to express the three-phase vector from the original a, b, c reference frame in the α, β, γ coordinates, where the first two elements α and β describe the rotating component of the original vector and the third term γ is the mean value of the original a, b, c vector.
Given a three-phase vector $\mathbf{v}_{a,b,c} = [v_a, v_b, v_c]^T$ in a, b, c coordinates, the Clarke transformation is performed as follows:

$$\mathbf{v}_{\alpha,\beta,\gamma} = \begin{bmatrix} v_\alpha \\ v_\beta \\ v_\gamma \end{bmatrix} = \frac{2}{3} \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix} \times \begin{bmatrix} v_a \\ v_b \\ v_c \end{bmatrix} = \mathbf{T} \times \mathbf{v}_{a,b,c}.$$

Similarly, the inverse Clarke transformation computes the vector in the a, b, c reference frame starting from a vector in α, β, γ coordinates, and the anti-transformation matrix is simply obtained by inverting the Clarke transformation matrix as follows:

$$\mathbf{v}_{a,b,c} = (\mathbf{T})^{-1} \times \mathbf{v}_{\alpha,\beta,\gamma} = \begin{bmatrix} 1 & 0 & 1 \\ -\frac{1}{2} & \frac{\sqrt{3}}{2} & 1 \\ -\frac{1}{2} & -\frac{\sqrt{3}}{2} & 1 \end{bmatrix} \times \mathbf{v}_{\alpha,\beta,\gamma}.$$

Reduced Clarke Transformation

In a balanced three-phase system, the sum of the three components of the $\mathbf{v}_{a,b,c}$ vector is zero. Hence, the γ component of the transformed $\mathbf{v}_{\alpha,\beta,\gamma}$ is null and it is sufficient to describe the vector by using only the two α, β values, reducing the variables of the system from 3 to 2 and simplifying the mathematical description.
The reduced Clarke transformation is performed as follows:

$$\mathbf{v}_{\alpha,\beta} = \begin{bmatrix} v_\alpha \\ v_\beta \end{bmatrix} = \frac{2}{3} \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix} \times \begin{bmatrix} v_a \\ v_b \\ v_c \end{bmatrix} = \mathbf{T}_{2\times3} \times \mathbf{v}_{a,b,c}.$$

In the same way, the reduced anti-transformation is computed by multiplying the $\mathbf{v}_{\alpha,\beta}$ vector times the pseudo-inverse of the reduced Clarke transformation matrix $\mathbf{T}_{2\times3}$, as follows:

$$\mathbf{v}_{a,b,c} = \mathbf{T}_{2\times3}^{-1} \times \mathbf{v}_{\alpha,\beta} = \begin{bmatrix} 1 & 0 \\ -\frac{1}{2} & \frac{\sqrt{3}}{2} \\ -\frac{1}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix} \times \mathbf{v}_{\alpha,\beta}.$$

Fig. 1shows the transformation of the vector from the a, b, c frame to the α, β frame.

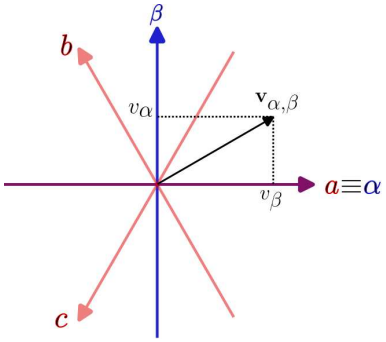


Figure 1: Clarke transformation.

Park Transformation

Since in a three-phase system the voltage and current three-phase vectors rotate at a specific frequency, the Park transformation was introduced to convert the $\mathbf{v}_{\alpha,\beta}$ vector from the stationary reference frame α, β to the d, q frame, which rotates at the steady-state frequency of the three-phase system. A vector rotating at a pulsation ω in the stationary coordinates, is a constant vector in the reference frame rotating at the same pulsation ω, thus, further simplifying the mathematical formulation.
The Park transformation to compute the $\mathbf{v}_{d,q}$ vector at time t starting from $\mathbf{v}_{\alpha,\beta}$ is performed as follows:

$$\mathbf{v}_{d,q} = \begin{bmatrix} v_d \\ v_q \end{bmatrix} = \begin{bmatrix} \cos(\omega t) & \sin(\omega t) \\ -\sin(\omega t) & \cos(\omega t) \end{bmatrix} \times \mathbf{v}_{\alpha,\beta}.$$

Similarly, the Park anti-transformation is computed as:

$$\mathbf{v}_{\alpha,\beta} = \begin{bmatrix} \cos(\omega t) & \sin(\omega t) \\ -\sin(\omega t) & \cos(\omega t) \end{bmatrix}^{-1} \times \mathbf{v}_{d,q} = \begin{bmatrix} \cos(\omega t) & -\sin(\omega t) \\ \sin(\omega t) & \cos(\omega t) \end{bmatrix} \times \mathbf{v}_{d,q}.$$

Fig. 2shows the transformation of the vector from the α, β frame to the d, q frame.

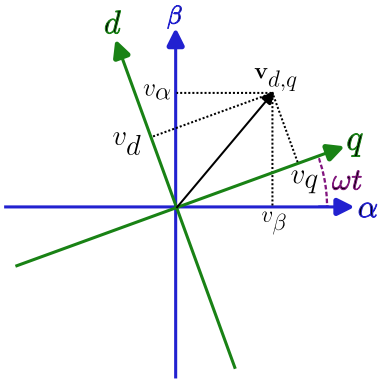


Figure 2: Park transformation.