Czech Technical University in Prague
Faculty of Electrical Engineering
Department of Electric Drives and Traction

Master Thesis

Modulation Strategies for Voltage Converters

Bc. Jakub Kučka

Supervisor: Ing. Jan Bauer

Study Program: Electrical Engineering, Power Engineering and Management, Master

Study Branch: Electrical Machines, Apparatus and Drives

May 1, 2014

Aknowledgements

I would like to thank to my supervisor Ing. Jan Bauer for a support, during working on this thesis and Ing. Vít Hlinovský for giving me his time during an absence of my supervisor.

I would also like to thank company `teste` for lending us `Tectronix Power Analyzer`.

Declaration

I hereby declare that I have completed this thesis independently and that I have listed all the literature and publications used.
I have no objection to usage of this work in compliance with the act §60 Zákon č. 121/2000Sb. (copyright law), and with the rights connected with the copyright act including the changes in the act.

Prague, May 1, 2014 ....

Abstract

This work presents modulation theories for voltage source inverter and matrix converter. It proposes one new modulation method for voltage source inverter, optimized for switching losses minimization. To verify the methods, a modulator design is implemented in VHDL for both converter topologies.

Abstrakt

Táto práca prezentuje modulačné teórie pre klasický napäťový striedač a pre maticový menič. Navrhuje novú modulačnú metódu pre napäťový striedač, optimalizovanú na spínacie straty. K overeniu popísaných metód boli modulátory pre oba meniče implementované vo VHDL.

Contents

List of Figures

List of Tables

Used Shortcuts

3RDM	Modulation with a 3rd Harmonic Injection
AC	Alternating Current
A/D	Analog to Digital
AM	Amplitude Modulation
CPU	Central Processing Unit
DC	Direct Current
DSVM	Direct Space Vector Modulation
FPGA	Field Programmable Grid Array
FSM	Final State Machine
GND	Ground
IGBT	Insulated Gate Bipolar Transistor
IO	Input Output
ISA	Industry Standard Architecture
ISVM	Indirect Space Vector Modulation
MC	Matrix Converter
MS DOS	Microsoft Disk Operating System
NC	not connected
PCI	Peripheral Component Interconnect
PLL	Phase Locked Loop
PWM	Pulse Width Modulation
R	readable
RGIGBT	Recessed-Gate IGBT
SM	Sinus Modulation
SVM	Space Vector Modulation
THD	Total Harmonic Distortion
UI	User Interface
VHDL	VHSIC Hardware Description Language
VSI	Voltage Source Inverter
W	writable

Used Physical Variables

E	energy
E_{loss}	dissipated energy
f	frequency
i, I	current
k	linearization coefficient
M	modulation degree
p, P	power
P_{loss}	dissipated power
t	time
v, V	voltage
ϕ	angle between voltage and current
ω	angular speed

Explanation of Variable Symbols

$x(t)$	dependency of variable x on time
$x_{,1}(t)$	first harmonic of variable $x(t)$
X	value of DC variable
$\vec{x}_{\alpha\beta 0}$	space vector of $x_{1,2,3}$ in $\alpha, \beta, 0$ coordinates
X_{RMS}	effective value of variable $x(t)$
$\bar{x}$	mean value of variable $x(t)$
$\hat{x}$	amplitude of value $x(t)$ (only for periodic functions)
$\underline{X}$	phasor of variable $x(t)$
x_{max}	maximal value of $x(t)$
x_{min}	minimal value of $x(t)$
x_{ref}	reference value for $x(t)$

Chapter 1

Introduction

In this work, two different converter topologies will be presented. The task of this work is to describe their modulation methods and then implement them in VHDL to verify their functionality.

The Voltage Source Inverter and Matrix Converter were chosen to be presented. The intention is to present modern converters, which are also available at the Department of Electric Drives and Traction, so the that the modulators can be practically verified on prototypes.

Voltage Source Inverter (VSI) is a well known converter, which is very often used in industry and in electrical traction drives. This work explains its modulation strategies and is proposing a new modulation strategy. The task of the new modulation strategy is to lower switching losses of the converter.

The switching losses are currently forming one of the biggest parts of the dissipated power. The amount of dissipated power negatively influences the efficiency of the converter and a cooling system size. The act of lowering this losses, might cause a lowering of a price of the converter.

Matrix Converter (MC) is a relatively novel converter type, which is not presently market-common. The main advantage of DC storage part absence is not that significant in comparison to its overall complexity and high amount of needed active components. Although MC is not economically applicable at the moment, it might be useful in future and is still a topic of a recent research. This work explains its functionality and then focuses on Indirect Space Vector Modulation (ISVM) theory.

To verify functionality of modulation methods, modulators for the both converters mentioned are implemented in VHDL and tested using their converter prototypes at the university department.

Chapter 2

Theory

2.1 Voltage Source Inverter

VSI or **"Voltage Source Inverter"** is a controlled voltage converter. It transfers energy between a DC voltage circuit and an AC current circuit. It is capable of a four-quadrant work, which makes it useful for drive applications.

Since transistors with possible high switching frequency were presented, it is the most common type of converter in drive applications. It is well known and useful because of its simplicity and low THD.

Most of the VSI functionality and modulation principles are going to be explained here. Some more information on this topic can be found at [1] and [2].

2.1.1 Functionality

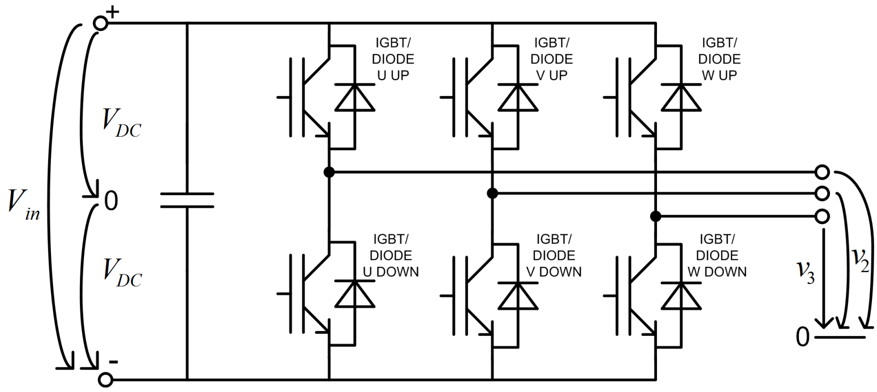


Figure 2.1: Schemes of Voltage Source Inverter

To understand VSI functionality, we have to define virtual zero "0" as the middle of the input DC voltage V_{in}. Therefore $+V_{DC}$ is at the same potential as + terminal of input voltage and $-V_{DC}$ is at the same potential as $-$ terminal. It is obvious that:

$$V_{DC} = \frac{1}{2} V_{in} \tag{2.1}$$

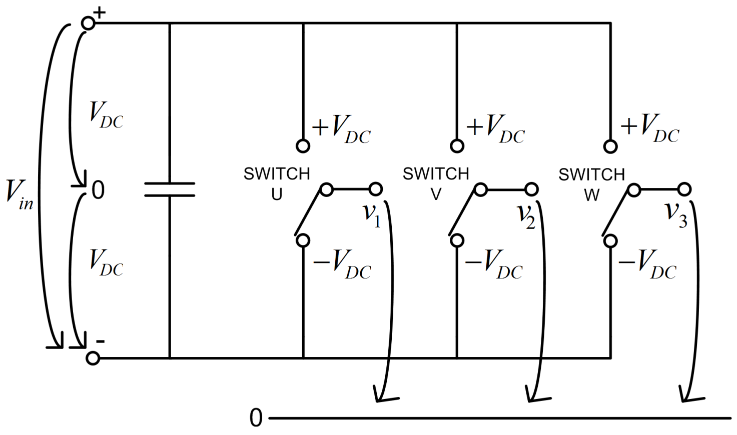


Figure 2.2: Simplified Schema of Voltage Source Inverter

On principle, every IGBT and diode bridge works as an ideal switch (fig. 2.2). It is possible to see, that every output terminal can be connected to either $+V_{DC}$ or $-V_{DC}$ in every moment. This gives us eight switching states, which are to be seen in table. 2.1. In this table there are also displayed line to line voltage and value of common voltage v_0 for every switching state.

State	v_1	v_2	v_3	v_{12}	v_{23}	v_{31}	v_0
ST0	$-V_{DC}$	$-V_{DC}$	$-V_{DC}$	0	0	0	$-V_{DC}$
ST1	$+V_{DC}$	$-V_{DC}$	$-V_{DC}$	$+2V_{DC}$	0	$-2V_{DC}$	$-\frac{1}{3}V_{DC}$
ST2	$+V_{DC}$	$+V_{DC}$	$-V_{DC}$	0	$+2V_{DC}$	$-2V_{DC}$	$+\frac{1}{3}V_{DC}$
ST3	$-V_{DC}$	$+V_{DC}$	$-V_{DC}$	$-2V_{DC}$	$+2V_{DC}$	0	$-\frac{1}{3}V_{DC}$
ST4	$-V_{DC}$	$+V_{DC}$	$+V_{DC}$	$-2V_{DC}$	0	$+2V_{DC}$	$+\frac{1}{3}V_{DC}$
ST5	$-V_{DC}$	$-V_{DC}$	$+V_{DC}$	0	$-2V_{DC}$	$+2V_{DC}$	$-\frac{1}{3}V_{DC}$
ST6	$+V_{DC}$	$-V_{DC}$	$+V_{DC}$	$+2V_{DC}$	$-2V_{DC}$	0	$+\frac{1}{3}V_{DC}$
ST7	$+V_{DC}$	$+V_{DC}$	$+V_{DC}$	0	0	0	$+V_{DC}$

Table 2.1: Table of Switching States for VSI

By choosing a combination of these eight switching states, we are able to get reference value of voltage on the output as mean value during one switching period. Common methods providing this are PWM or duty-cycle calculation (explained in 2.2.2.3).

2.1.2 Restrictions

Instead of ideal switches, IGBTs with diodes are used. As IGBTs do not turn off immediately, it could lead to short circuiting of the input capacitor, what would cause overcurrent and possible converter destruction. Thus, a short period of time is needed between every switching, when both IGBTs are turned off. This time period is called "dead time" and it will have to be included in a modulator design.

2.1.3 Space Vector and $\alpha, \beta, 0$ - Transformation

For modulation to be explained it has to be $\alpha, \beta, 0$ - transformation defined. A power invariant transformation (2.2) was chosen to be used.

$$\vec{x}_{\alpha\beta 0} = \begin{bmatrix} x_\alpha \\ x_\beta \\ x_0 \end{bmatrix} = \sqrt{\frac{2}{3}} \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \end{bmatrix} \tag{2.2}$$

Vector $[x_\alpha \; x_\beta \;\; x_0]$ is called space vector $\vec{x}_{\alpha\beta 0}$. To hold a sinusoidal line to line output voltage, a projection of space vector to α, β plane ("zero plane") has to be a vector rotating with constant angular speed on a circle (to keep the amplitude constant).

2.1.4 Known Modulation Methods

To introduce modulator functionality, some of known modulation methods for VSI are presented.

2.1.4.1 Amplitude Modulation

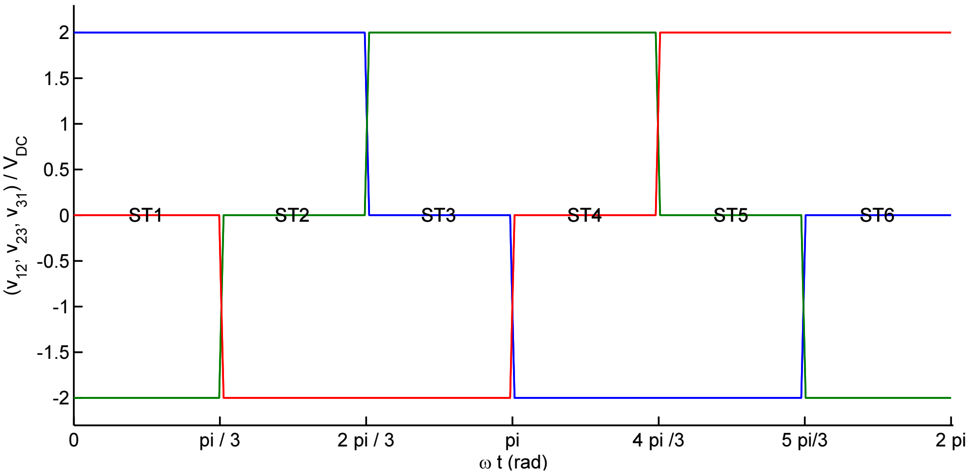


Figure 2.3: Line to Line Voltages During One Period and Switching Function.

AM or "Amplitude Modulation" is the oldest known modulation method. It is the simplest method, with a switching frequency matching the frequency of a generated function.

An important disadvantage of this method is a very high THD factor. Another disadvantage is that output voltage amplitude depends on an input voltage level. It was used frequently in past, when a low switching frequency was a significant advantage. With use of IGBT transistors, this method is very uncommon.

The principle of this method is based on a simple switching between the switching states ST1 to ST6, where every switching state is active for one sixth of period. As it was already mentioned before, the amplitude of output voltage is controlled by adjusting of an input DC voltage level. Therefore, a switching function for this method is influenced just by a period length. Switching function is shown in fig. 2.3.

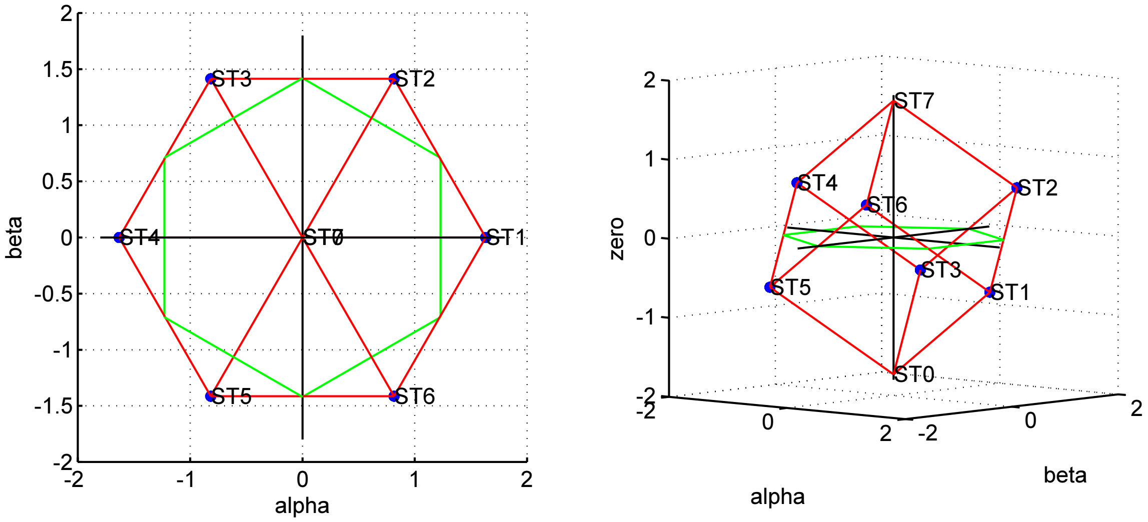


Figure 2.4: Space Vector of Amplitude Modulation in $\alpha, \beta, 0$ Coordinates.

By observing the position of the space vector for this modulation, it is possible to see that it switches discontinuously between all of the switching states. The space vector is represented by the six blue points in fig. 2.4.

In all of the modulation figures in this chapter, the red lines represent the switching cube inside of which all possible space vector states are to be found. The vertexes of the cube are the basic switching states. Green lines represent an intersection of the cube space with the “zero plane”. The resulting plane represents an area, where mean value of common voltage during one switching cycle is held at zero. Blue lines or points represent a trajectory of space vector, while using a chosen modulation method.

2.1.4.2 Sinus Modulation

SM or “Sinus Modulation” is practically the simplest method to be understood. It is a method, where the switching function is generated by applying PWM on a sine reference signal. Generation of this signal is explained in fig. 2.5.

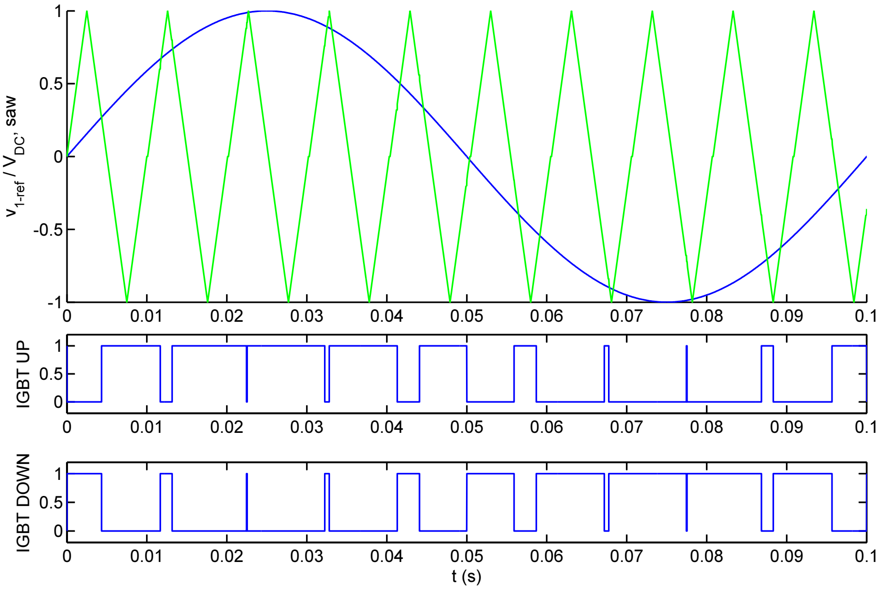


Figure 2.5: Generation of Switching Functions for IGBT Gates using Sinus Modulation

The amplitude is controlled by adjusting an PWM reference signal. The switching frequency does not depend on the output function frequency. An adjustment of modulated signal is performed using modulation degree M.

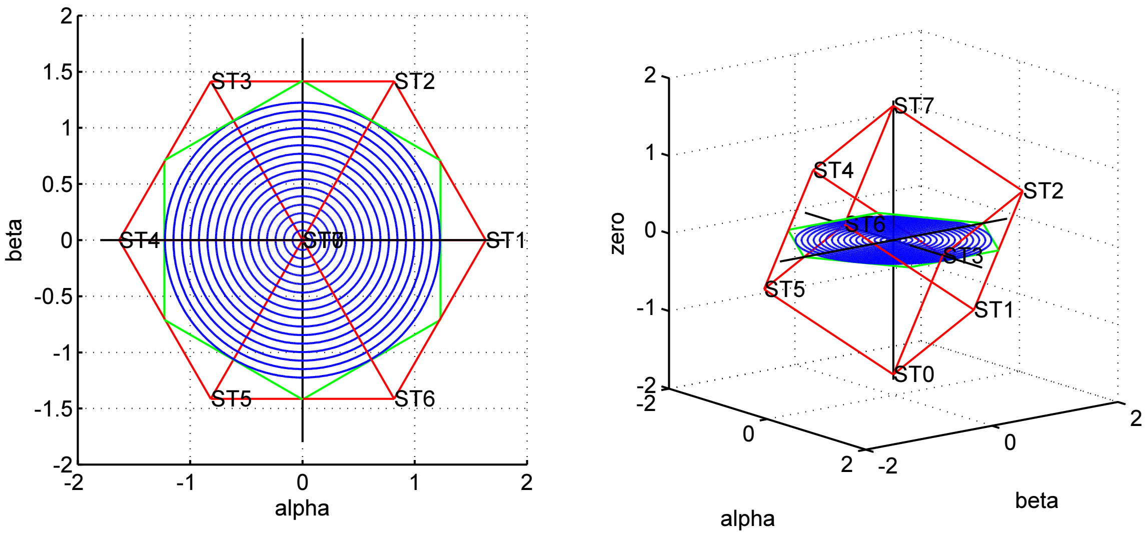


Figure 2.6: Space Vector of Sinus Modulation in $\alpha, \beta, 0$ Coordinates.

$$M = \frac{\hat{v}_{ref}}{V_{DC}} \tag{2.3}$$

Voltage $\hat{v}_{ref}$ is a reference voltage amplitude.

As it is displayed in fig. 2.6, the sinus modulation holds the common voltage $\overline{v}_0 = 0$ as a mean value during one switching cycle. Every blue line represents a space vector trajectory for different modulation degrees. Maximal modulation degree M_{max} is limited to value 1, otherwise the vector would exceed the switching cube space and it would come to overmodulation. The low maximal modulation degree is the reason why it is not the only used method (this method is chosen just for small frequencies in electrical drives, where there is not a need for higher modulation degree).

2.1.4.3 Modulation with 3rd Harmonic Injection

3RDM or "Modulation with 3rd Harmonic Injection" is a modulation, which solves the maximal modulation degree disadvantage of SM. Its main principle is to inject a common voltage function $v_{0,ref}$ in all reference voltages. This function is chosen as one sixth of common 3rd harmonic of $v_{1,ref}$, $v_{2,ref}$ and $v_{3,ref}$. As this is a common part of voltage, which is the same in all of the phases, and the zero clamp is not grounded, it does not have any influence on output currents.

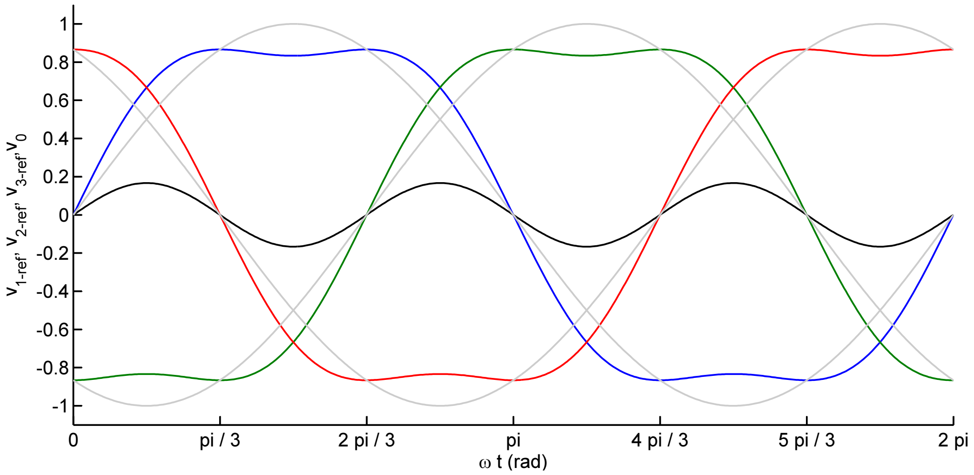


Figure 2.7: Modulation with 3rd Harmonic Injection Reference Function

For reference voltages

$$\begin{bmatrix} v_{1,ref}(t) \\ v_{2,ref}(t) \\ v_{3,ref}(t) \end{bmatrix} = MV_{DC} \begin{bmatrix} \sin(\omega t + \Psi) \\ \sin(\omega t + \Psi + \frac{2\pi}{3}) \\ \sin(\omega t + \Psi + \frac{4\pi}{3}) \end{bmatrix} + v_{0,ref} \tag{2.4}$$

where Ψ is an initial angle, $v_{0,ref}$ is defined as:

$$v_{0,ref}(t) = \frac{1}{6} M V_{DC} \sin(3\omega t + 3\Psi) \tag{2.5}$$

With 3rd harmonic injection, it is possible to see (fig. 2.7), that a maximum of the reference function $v_{ref,max}$ is lower then amplitude $\hat{v}_{ref}$. From fig. 2.8 left we can determine, that:

$$M_{max} = \frac{1}{\cos 30^o} = \frac{2}{\sqrt{3}} = 1.15470... \tag{2.6}$$

This is because, the space vector is not sticking to the "zero plane". This is observable in fig. 2.8.

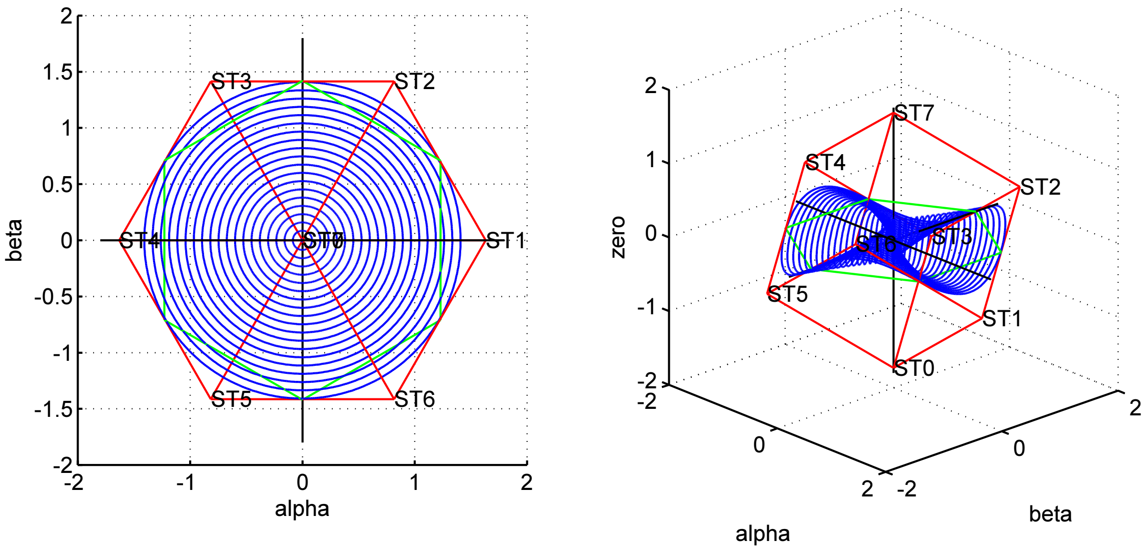


Figure 2.8: Space Vector of 3RDM in $\alpha, \beta, 0$ Coordinates.

2.1.4.4 Space Vector Modulation

SVM or "Space Vector Modulation" is a modulation strategy, which improves 3RDM. The great disadvantage of 3RDM is that an angular speed is needed to calculate the common voltage $v_{0,ref}$. An angular speed is not precisely directly measurable and therefore some kind of PLL is needed in a modulator design. This has a relatively great impact on a control stability. SVM, on the other hand, is a method which depends only on actual reference values, what makes it significantly advantageous.

For reference voltages system (2.4) is the common reference voltage defined as:

$$v_{0,ref}(t) = -\frac{\max\left(v_{1,ref}(t), v_{2,ref}(t), v_{3,ref}(t)\right) + \min\left(v_{1,ref}(t), v_{2,ref}(t), v_{3,ref}(t)\right)}{2} \tag{2.7}$$

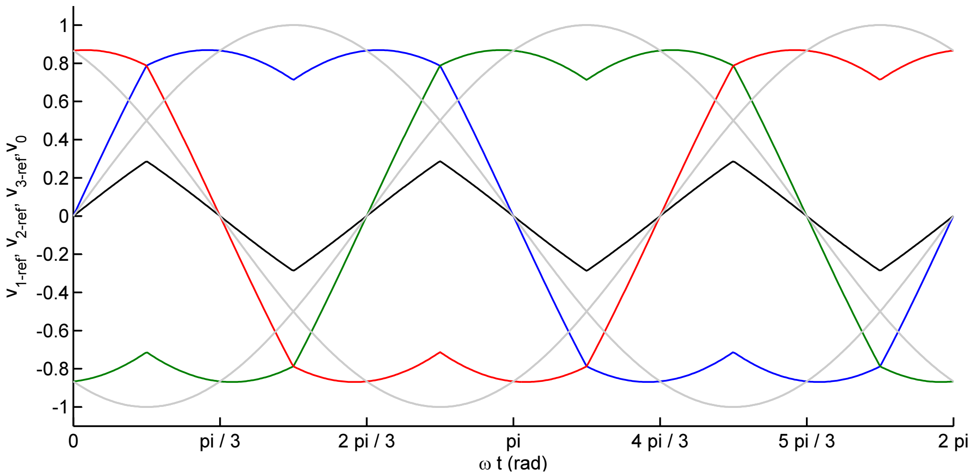


Figure 2.9: Space Vector Modulation Reference Function

From fig. 2.9 and fig. 2.10, we can state, that this method has the same advantages as 3RDM, thou it is much easier to implement this method.

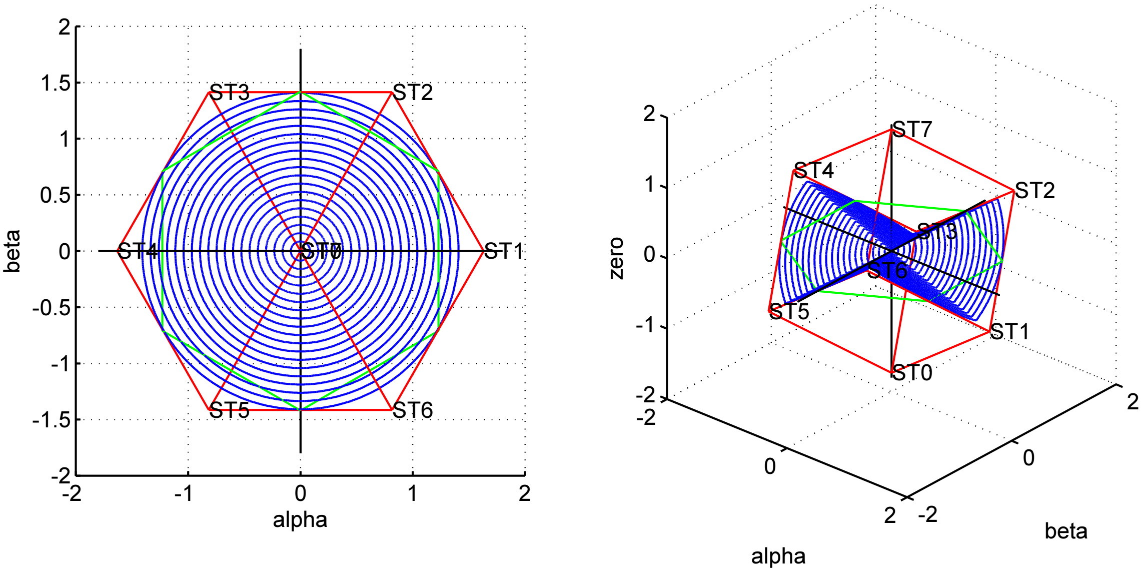


Figure 2.10: Space Vector of Space Vector Modulation in $\alpha, \beta, 0$ Coordinates.

2.2 Matrix Converter

MC or **"Matrix Converter"** is a direct converter. It transfers the energy from an AC voltage circuit to an AC current circuit with sine currents on both sides and with a low THD. Usually, it transfers energy between mains and an AC motor or a power grid with another frequency.

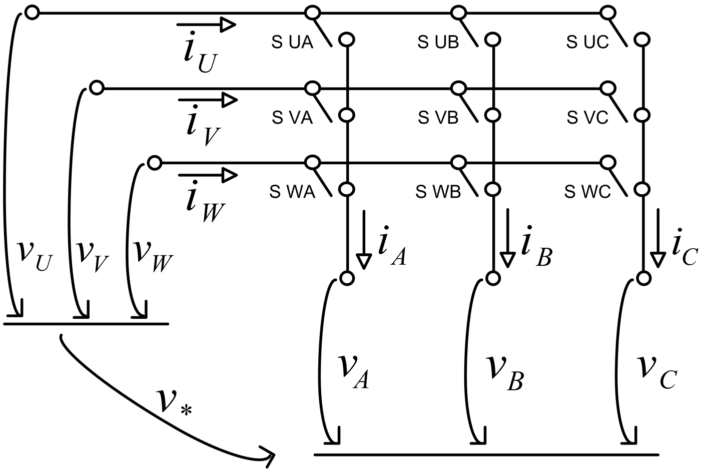


Figure 2.11: Principle Scheme of Matrix Converter

This converter provides operation in all four quadrants, with optional frequency on the both sides (the frequency is limited only by switching frequency). Nevertheless the input power factor is often selected to 1, to keep the maximal transfer ratio between the output and input voltage amplitude of 0.866.

The main benefit of this converter is that there is no DC electrical storage part demand that might make the whole design more compact and it is also possible to start the converter quicker after connecting to mains (there is no need to load the DC capacitor) [6].

There are also few modifications of Matrix Converter topology available and they will not be presented here. If you need this information, please refer to [5].

2.2.1 Functionality

The main idea of this converter topology (fig. 2.11) is to convert energy directly by switching freely one of the input phases (U, V, W) to the output phases (A, B, C).

There are two important restrictions, which determine the count of possible switching states:

- Because of the voltage input source which must not be short circuited, an output phase can be connected just to one input phase at the same time.

- Because of the current output circuit which must not be opened, an output phase has to be constantly connected to some of the input phases.

Therefore every output phase has to be connected to exactly one input phase during the whole time and therefore 3^3 of switching states are available.

An overview of switching states is summarized in tab. 2.2. There are three main groups of switching states. The rotating switching states, labeled as `Rot`, are representing states, when an every output phase is connected to a different input phase. As it will be mentioned later, these switching states cannot be used for Indirect Space Vector Modulation. The second group labeled as `UV/VU`, `VW/WV` and `WU/UW` is a group of active switching states. The third group labeled as `zero` is a group of zero sequence switching states. There is also an ISVM interpretation included in the table.

	UA	UB	UC	VA	VB	VC	WA	WB	WC	ISVM interpretation
Rot	1	0	0	0	1	0	0	0	1	-
	0	1	0	0	0	1	1	0	0	-
	0	0	1	1	0	0	0	1	0	-
	1	0	0	0	0	1	0	1	0	-
	0	0	1	0	1	0	1	0	0	-
	0	1	0	1	0	0	0	0	1	-
UV/VU	0	0	1	1	1	0	0	0	0	UV 001, VU 110
	0	1	0	1	0	1	0	0	0	UV 010, VU 101
	0	1	1	1	0	0	0	0	0	UV 011, VU 100
	1	0	0	0	1	1	0	0	0	UV 100, VU 011
	1	0	1	0	1	0	0	0	0	UV 101, VU 010
	1	1	0	0	0	1	0	0	0	UV 110, VU 001
VW/WV	0	0	0	0	0	1	1	1	0	VW 001, WV 110
	0	0	0	0	1	0	1	0	1	VW 010, WV 101
	0	0	0	0	1	1	1	0	0	VW 011, WV 100
	0	0	0	1	0	0	0	1	1	VW 100, WV 011
	0	0	0	1	0	1	0	1	0	VW 101, WV 010
	0	0	0	1	1	0	0	0	1	VW 110, WV 001
WU/UW	1	1	0	0	0	0	0	0	1	WU 001, UW 110
	1	0	1	0	0	0	0	1	0	WU 010, UW 101
	1	0	0	0	0	0	0	1	1	WU 011, UW 100
	0	1	1	0	0	0	1	0	0	WU 100, UW 011
	0	1	0	0	0	0	1	0	1	WU 101, UW 010
	0	0	1	0	0	0	1	1	0	WU 110, UW 001
zero	1	1	1	0	0	0	0	0	0	UV 111, VU 000, UW 111, WU 000
	0	0	0	1	1	1	0	0	0	VU 111, UV 000, VW 111, WV 000
	0	0	0	0	0	0	1	1	1	WU 111, UW 000, WV 111, VW 000

Table 2.2: Table of Matrix Converter Switching States

With these switching states, using a modulation strategy, a required operation of MC can be obtained.

2.2.2 Modulation Methods

To ensure the expected converter operation a modulation has to be present. There are a few possible modulation methods suitable for MC.

Generally, all modulation methods expect symmetric sine voltage and current at the input and the output. In the following equations, input values are labeled with an index *i* and output values with an index *o*.

$$\begin{bmatrix} v_U(t) \\ v_V(t) \\ v_W(t) \end{bmatrix} = \hat{v}_i \begin{bmatrix} \cos(\omega_i t) \\ \cos\left(\omega_i t - \frac{2\pi}{3}\right) \\ \cos\left(\omega_i t - \frac{4\pi}{3}\right) \end{bmatrix} \tag{2.8}$$

$$\begin{bmatrix} v_A(t) \\ v_B(t) \\ v_C(t) \end{bmatrix} = \hat{v}_o \begin{bmatrix} \cos(\omega_o t + \Psi) \\ \cos\left(\omega_o t + \Psi - \frac{2\pi}{3}\right) \\ \cos\left(\omega_o t + \Psi - \frac{4\pi}{3}\right) \end{bmatrix} = A \cdot \hat{v}_i \begin{bmatrix} \cos(\omega_o t + \Psi) \\ \cos\left(\omega_o t + \Psi - \frac{2\pi}{3}\right) \\ \cos\left(\omega_o t + \Psi - \frac{4\pi}{3}\right) \end{bmatrix} \tag{2.9}$$

$$\begin{bmatrix} i_U(t) \\ i_V(t) \\ i_W(t) \end{bmatrix} = \hat{i}_i \begin{bmatrix} \cos(\omega_i t + \varphi_i) \\ \cos\left(\omega_i t + \varphi_i - \frac{2\pi}{3}\right) \\ \cos\left(\omega_i t + \varphi_i - \frac{4\pi}{3}\right) \end{bmatrix} = A \cdot \hat{i}_o \begin{bmatrix} \cos(\omega_i t + \varphi_i) \\ \cos\left(\omega_i t + \varphi_i - \frac{2\pi}{3}\right) \\ \cos\left(\omega_i t + \varphi_i - \frac{4\pi}{3}\right) \end{bmatrix} \tag{2.10}$$

$$\begin{bmatrix} i_A(t) \\ i_B(t) \\ i_C(t) \end{bmatrix} = \hat{i}_i \begin{bmatrix} \cos(\omega_o t + \varphi_o + \Psi) \\ \cos\left(\omega_o t + \varphi_o + \Psi - \frac{2\pi}{3}\right) \\ \cos\left(\omega_o t + \varphi_o + \Psi - \frac{4\pi}{3}\right) \end{bmatrix} \tag{2.11}$$

A is a voltage transfer ratio. It is defined as:

$$A = \frac{\hat{v}_o}{\hat{v}_i} \tag{2.12}$$

Every switching state from tab. 2.2 can be written into a matrix form, forming switching state matrix $\mathbb{S}$.

$$\mathbb{S} = \begin{bmatrix} S_{UA} & S_{UB} & S_{UC} \\ S_{VA} & S_{VB} & S_{VC} \\ S_{WA} & S_{WB} & S_{WC} \end{bmatrix} \tag{2.13}$$

To meet the mentioned restrictions (every output phase has to be connected to exactly one input phase during the whole operation), the following equations have to be met.

$$\begin{bmatrix} S_{UA} + S_{UB} + S_{UC} \\ S_{VA} + S_{VB} + S_{VC} \\ S_{WA} + S_{WB} + S_{WC} \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \tag{2.14}$$

Because a voltage input and a current output are expected, we can determine the input currents and output voltages of the converter using switching state matrix.

$$\begin{bmatrix} u_A \\ u_B \\ u_C \end{bmatrix} = \mathbb{S} \cdot \begin{bmatrix} v_U \\ v_V \\ v_W \end{bmatrix} = \begin{bmatrix} S_{UA} & S_{UB} & S_{UC} \\ S_{VA} & S_{VB} & S_{VC} \\ S_{WA} & S_{WB} & S_{WC} \end{bmatrix} \cdot \begin{bmatrix} v_U \\ v_V \\ v_W \end{bmatrix} \tag{2.15}$$

$$\begin{bmatrix} i_U \\ i_V \\ i_W \end{bmatrix} = \mathbb{S}^{\mathrm{T}} \cdot \begin{bmatrix} i_A \\ i_B \\ i_C \end{bmatrix} = \begin{bmatrix} S_{UA} & S_{VA} & S_{WA} \\ S_{UB} & S_{VB} & S_{WB} \\ S_{UC} & S_{VC} & S_{WC} \end{bmatrix} \cdot \begin{bmatrix} i_A \\ i_B \\ i_C \end{bmatrix} \tag{2.16}$$

To obtain reference values of voltage and current, more switching states $\mathbb{S}_i$ have to be used during one switching cycle. To each of the switching states, there is related a duty cycle value d_i. The values of input current and output voltage are then synthesized after (2.17) and (2.18) as mean values during the whole switching period.

$$\begin{bmatrix} v_A \\ v_B \\ v_C \end{bmatrix} = \sum_{i=1}^{27} d_i \cdot \mathbb{S}_i \begin{bmatrix} v_U \\ v_V \\ v_W \end{bmatrix} \tag{2.17}$$

$$\begin{bmatrix} i_U \\ i_V \\ i_W \end{bmatrix} = \sum_{i=1}^{27} d_i \cdot \mathbb{S}_i^{\mathrm{T}} \begin{bmatrix} i_A \\ i_B \\ i_C \end{bmatrix} \tag{2.18}$$

The sequence of individual switching states during the switching cycle is called a switching pattern. There are many possibilities of choosing a pattern and there is also a degree of freedom by choosing which of `zero` switching states will be used. The selection of the pattern might influence THD, common mode voltage and switching losses.

As following, few modulation methods will be explained. The focus will be concerned mainly on Indirect Space Vector Modulation method. More information on the other modulation methods can be obtained at [6].

2.2.2.1 Modulation Duty-Cycle Matrix Strategies

The principle of Modulation Duty-Cycle Matrix Strategies is to calculate a duty cycle matrix, which defines a duty cycle period for every switch separately and then to combine switching states with according switching-state duty-cycle periods to relate to the duty-cycle matrix. Duty-cycle matrix $\mathbb{D}$ is derived from (2.17) and (2.18).

$$\begin{bmatrix} v_A \\ v_B \\ v_C \end{bmatrix} = \sum_{i=1}^{27} d_i \cdot \mathbb{S}_i \begin{bmatrix} v_U \\ v_V \\ v_W \end{bmatrix} = \begin{bmatrix} d_{UA} & d_{UB} & d_{UC} \\ d_{VA} & d_{VB} & d_{VC} \\ d_{WA} & d_{WB} & d_{WC} \end{bmatrix} \begin{bmatrix} v_U \\ v_V \\ v_W \end{bmatrix} = \mathbb{D} \cdot \begin{bmatrix} v_U \\ v_V \\ v_W \end{bmatrix} \tag{2.19}$$

$$\begin{bmatrix} i_U \\ i_V \\ i_W \end{bmatrix} = \sum_{i=1}^{27} d_i \cdot \mathbb{S}_i^{\mathrm{T}} \begin{bmatrix} i_A \\ i_B \\ i_C \end{bmatrix} = \begin{bmatrix} d_{UA} & d_{VA} & d_{WA} \\ d_{UB} & d_{VB} & d_{WB} \\ d_{UC} & d_{VC} & d_{WC} \end{bmatrix} \begin{bmatrix} i_A \\ i_B \\ i_C \end{bmatrix} = \mathbb{D}^{\mathrm{T}} \cdot \begin{bmatrix} i_A \\ i_B \\ i_C \end{bmatrix} \tag{2.20}$$

There are two main methods using this strategy: Alesina-Venturini, Alesina-Venturini Optimum. There were proposed more methods, also called scalar, which are though just modified Alesina-Venturini Optimum Method [6].

The **Alesina-Venturini** method calculates the solution directly without injecting any common reference voltage (according to (2.9)). This method's main disadvantage is the low voltage transfer ratio $A_{max} = 0.5$. The construction of voltage reference functions can be seen in fig. 2.12.

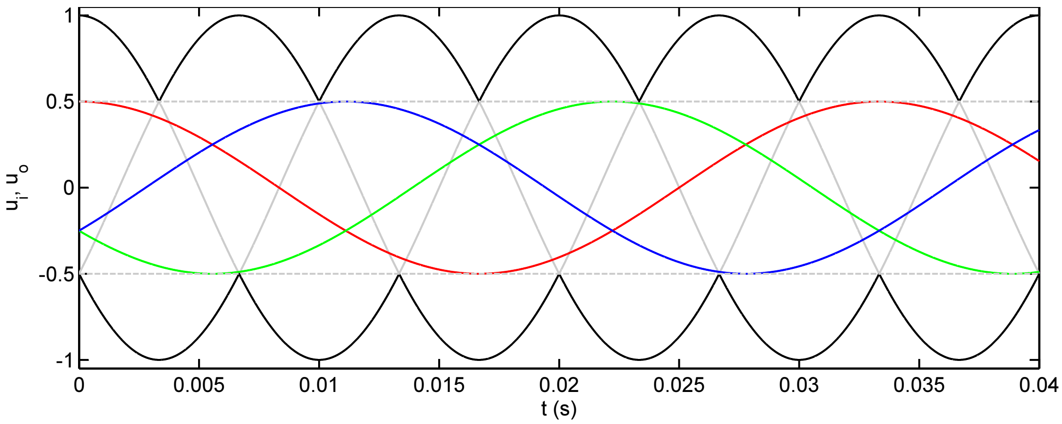


Figure 2.12: Alesina-Venturini Reference Function Example, $f_1 = 50$ Hz, $f_2 = 30$ Hz, $A = 0.5$

In the figure, the reference input voltage functions are colored gray, the reference output voltage functions are colorful.

The **Alesina-Venturini Optimum** method increases the voltage transfer ratio to 0.866 using an injection of common mode voltage into output voltage reference function. The reference function is then:

$$\begin{bmatrix} v_A(t) \\ v_B(t) \\ v_C(t) \end{bmatrix} = A \cdot \hat{v}_i \begin{bmatrix} \cos\left(\omega_o t + \Psi\right) \\ \cos\left(\omega_o t + \Psi - \frac{2\pi}{3}\right) \\ \cos\left(\omega_o t + \Psi - \frac{4\pi}{3}\right) \end{bmatrix} - \frac{A \cdot \hat{v}_i}{6} \cos(3\omega_o t + 3\Psi) + \frac{\hat{v}_i}{4} \cos(3\omega_i t) \tag{2.21}$$

The construction of voltage reference functions of Alesina-Venturini Optimum method can be seen in fig. 2.13.

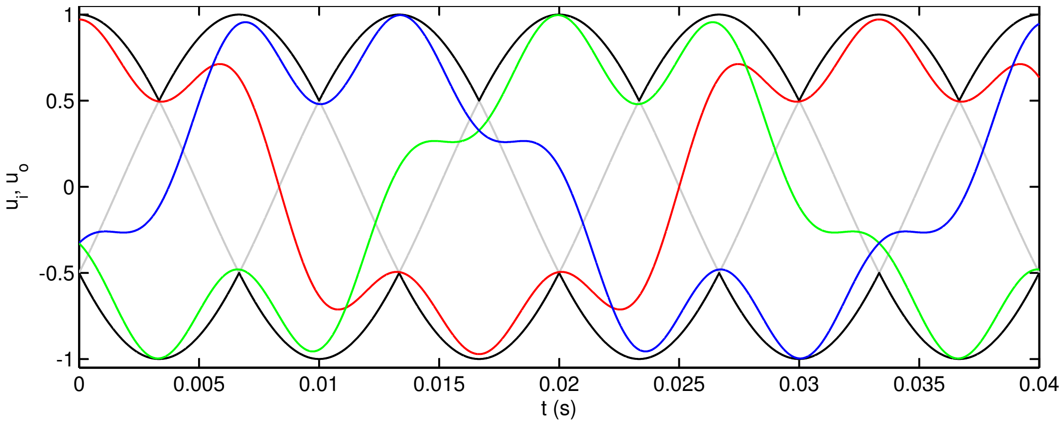
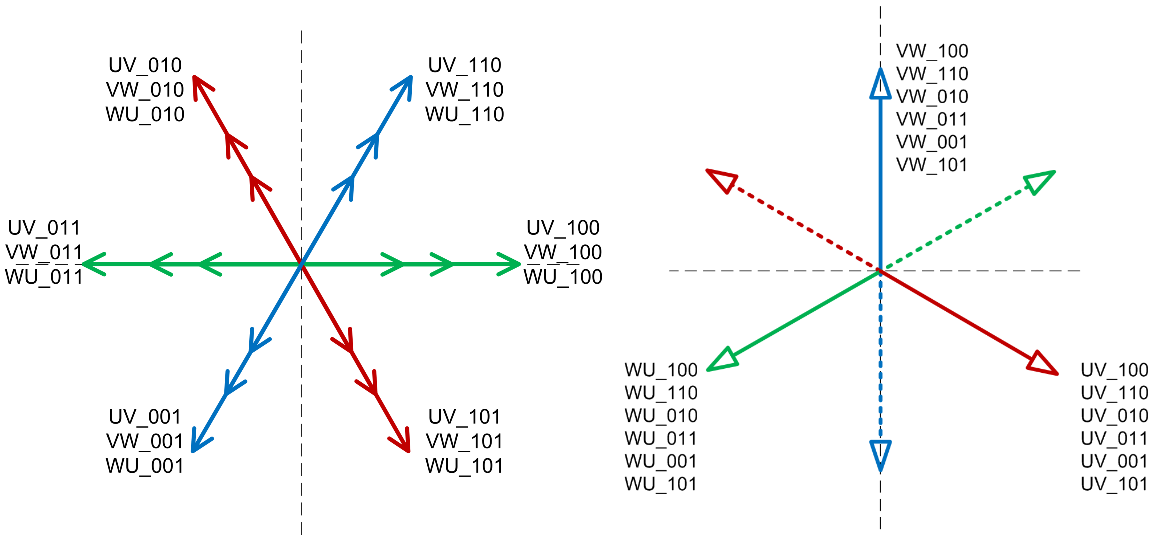


Figure 2.13: Alesina-Venturini Optimum Reference Function Example, $f_1 = 50$ Hz, $f_2 = 30$ Hz, $A = 0.866$

2.2.2.2 Direct Space Vector Modulation

The principle of Direct Space Vector Modulation is using Space Vector Modulation Theory originally researched for VSI. The main difference to the VSI theory is the larger amount of switching state options. Also, the length of particular voltage vectors varies during the time.

Similar to VSI, there is a rotating output reference voltage vector, which is synthesized using switching states. By MC DSVM, there are always four active vectors and one zero vector used. In fig. 2.14a, there is a DSVM diagram for output voltage. It is possible to see that in every sector six active vectors are available. Which four of them will be used is selected by input current modulation.



(a) Output Voltage SVM

(b) Input Current SVM

Figure 2.14: Direct Space Vector Modulation

In fig. 2.14b, there is a DSVM diagram for an input current. According to the output reference vector direction, current vectors direction is determined (full line in figure marks positive direction). The suitable vectors for synthesizing of an input current reference vector are then selected. By excluding two non passing output voltage vectors, four switching state vectors are obtained. Combining these four vectors with zero vector, duty cycles of particular switching states are calculated and the input current reference vector and the output voltage reference vector are synthesized.

This modulation method is not going to be used in implementation; therefore deeper calculations are not included. Deeper explanation of this method is to be found in [6].

2.2.2.3 Indirect Space Vector Modulation

The principle of ISVM is based on dividing the converter to two virtual stages. The first virtual stage is a virtual rectifier stage, where duty cycles for an input current reference vector are calculated. The second stage is a virtual inverter stage, where duty cycles for an output voltage reference vector are calculated. The both stages are interconnected with virtual DC circuit. The principle schema is in fig. 2.15.

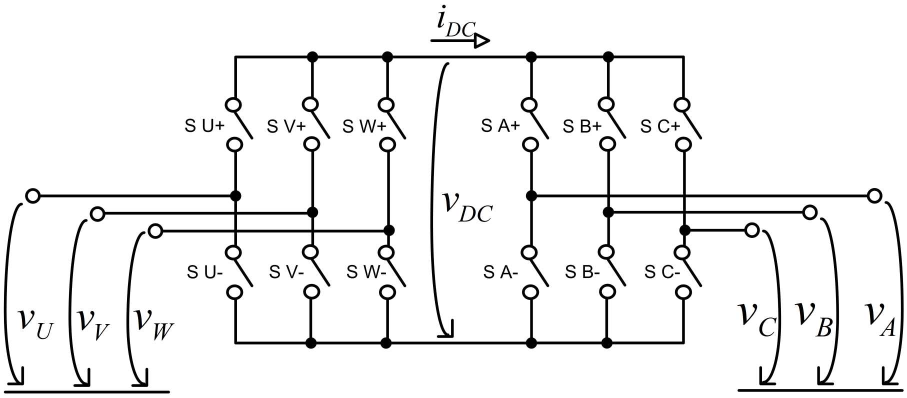


Figure 2.15: Principle Schema of Matrix Converter Indirect Modulation Stages

For ISVM, two new switching states matrices are defined: $\mathbb{S}_R$ for virtual rectifier stage and $\mathbb{S}_I$ for virtual inverter stage. The turned on switch signifies '1' in the switching state matrices and turned off is '0'. For the possible switching states, the standard VSI restrictions (every phase has to be connected either to $+\frac{1}{2}v_{DC}$ or to $-\frac{1}{2}v_{DC}$) and Current Source Rectifier restrictions (one phase has to be connected to $+\frac{1}{2}v_{DC}$ and one to $-\frac{1}{2}v_{DC}$) are relevant.

The virtual rectifier stage voltage equations can be written using virtual rectifier switching state matrix $\mathbb{S}_R$.

$$\begin{bmatrix} v_U \\ v_V \\ v_W \end{bmatrix} = \begin{bmatrix} S_{U+} & S_{U-} \\ S_{V+} & S_{V-} \\ S_{W+} & S_{W-} \end{bmatrix} \begin{bmatrix} \frac{1}{2}v_{DC} \\ -\frac{1}{2}v_{DC} \end{bmatrix} = \mathbb{S}_R \cdot \begin{bmatrix} \frac{1}{2}v_{DC} \\ -\frac{1}{2}v_{DC} \end{bmatrix} \tag{2.22}$$

Similar, we can derive virtual inverter stage voltage equations using $\mathbb{S}_I$ as:

$$\begin{bmatrix} v_A \\ v_B \\ v_C \end{bmatrix} = \begin{bmatrix} S_{A+} & S_{A-} \\ S_{B+} & S_{B-} \\ S_{C+} & S_{C-} \end{bmatrix} \begin{bmatrix} \frac{1}{2}v_{DC} \\ -\frac{1}{2}v_{DC} \end{bmatrix} = \mathbb{S}_I \cdot \begin{bmatrix} \frac{1}{2}v_{DC} \\ -\frac{1}{2}v_{DC} \end{bmatrix} \tag{2.23}$$

By merging two equations into one, we get:

$$\mathbb{S}_I^{\mathrm{T}} \cdot \begin{bmatrix} v_A \\ v_B \\ v_C \end{bmatrix} = \mathbb{S}_R^{\mathrm{T}} \cdot \begin{bmatrix} v_U \\ v_V \\ v_W \end{bmatrix} \tag{2.24}$$

$$\begin{bmatrix} v_A \\ v_B \\ v_C \end{bmatrix} = \mathbb{S}_I \cdot \mathbb{S}_R^{\mathrm{T}} \cdot \begin{bmatrix} v_U \\ v_V \\ v_W \end{bmatrix} \tag{2.25}$$

$$\begin{bmatrix} v_A \\ v_B \\ v_C \end{bmatrix} = \begin{bmatrix} S_{A+} & S_{A-} \\ S_{B+} & S_{B-} \\ S_{C+} & S_{C-} \end{bmatrix} \cdot \begin{bmatrix} S_{U+} & S_{V+} & S_{W+} \\ S_{U-} & S_{V-} & S_{W-} \end{bmatrix} \cdot \begin{bmatrix} v_U \\ v_V \\ v_W \end{bmatrix} \tag{2.26}$$

$$\begin{bmatrix} v_A \\ v_B \\ v_C \end{bmatrix} = \begin{bmatrix} S_{A+}S_{U+} + S_{A-}S_{U-} & S_{A+}S_{V+} + S_{A-}S_{V-} & S_{A+}S_{W+} + S_{A-}S_{W-} \\ S_{B+}S_{U+} + S_{B-}S_{U-} & S_{B+}S_{V+} + S_{B-}S_{V-} & S_{B+}S_{W+} + S_{B-}S_{W-} \\ S_{C+}S_{U+} + S_{C-}S_{U-} & S_{C+}S_{V+} + S_{C-}S_{V-} & S_{C+}S_{W+} + S_{C-}S_{W-} \end{bmatrix} \cdot \begin{bmatrix} v_U \\ v_V \\ v_W \end{bmatrix} \tag{2.27}$$

Using equation (2.15), we can determine the transformation from the combination of virtual inverter and rectifier switching states to the matrix converter switching state.

$$\mathbb{S} = \begin{bmatrix} S_{A+}S_{U+} + S_{A-}S_{U-} & S_{A+}S_{V+} + S_{A-}S_{V-} & S_{A+}S_{W+} + S_{A-}S_{W-} \\ S_{B+}S_{U+} + S_{B-}S_{U-} & S_{B+}S_{V+} + S_{B-}S_{V-} & S_{B+}S_{W+} + S_{B-}S_{W-} \\ S_{C+}S_{U+} + S_{C-}S_{U-} & S_{C+}S_{V+} + S_{C-}S_{V-} & S_{C+}S_{W+} + S_{C-}S_{W-} \end{bmatrix} \tag{2.28}$$

As next, we will focus on the space vector modulation. The virtual inverter is modulated as a standard VSI. The virtual rectifier is modulated as a Current Source Rectifier. For following calculations power invariant α, β-transformation will be used:

$$\vec{x}_{\alpha\beta} = \begin{bmatrix} x_\alpha \\ x_\beta \end{bmatrix} = \sqrt{\frac{2}{3}} \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \end{bmatrix} \tag{2.29}$$

Using α, β-transformation on equations (2.9) and (2.10), we can obtain relative reference vectors in α, β axes.

$$\vec{v}_{o,ref} = \begin{bmatrix} v_{o,ref,\alpha} \\ v_{o,ref,\beta} \end{bmatrix} = \frac{1}{\frac{1}{2}v_{DC}} \sqrt{\frac{2}{3}} \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix} \begin{bmatrix} v_A(t) \\ v_B(t) \\ v_C(t) \end{bmatrix} \tag{2.30}$$

$$\vec{v}_{o,ref} = M_o \sqrt{\frac{2}{3}} \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix} \begin{bmatrix} \cos(\omega_o t + \Psi) \\ \cos\left(\omega_o t + \Psi - \frac{2\pi}{3}\right) \\ \cos\left(\omega_o t + \Psi - \frac{4\pi}{3}\right) \end{bmatrix} \tag{2.31}$$

$$\vec{i}_{i,ref} = \begin{bmatrix} i_{i,ref,\alpha} \\ i_{i,ref,\beta} \end{bmatrix} = \frac{1}{i_{DC}} \sqrt{\frac{2}{3}} \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix} \begin{bmatrix} i_U(t) \\ i_V(t) \\ i_W(t) \end{bmatrix} \tag{2.32}$$

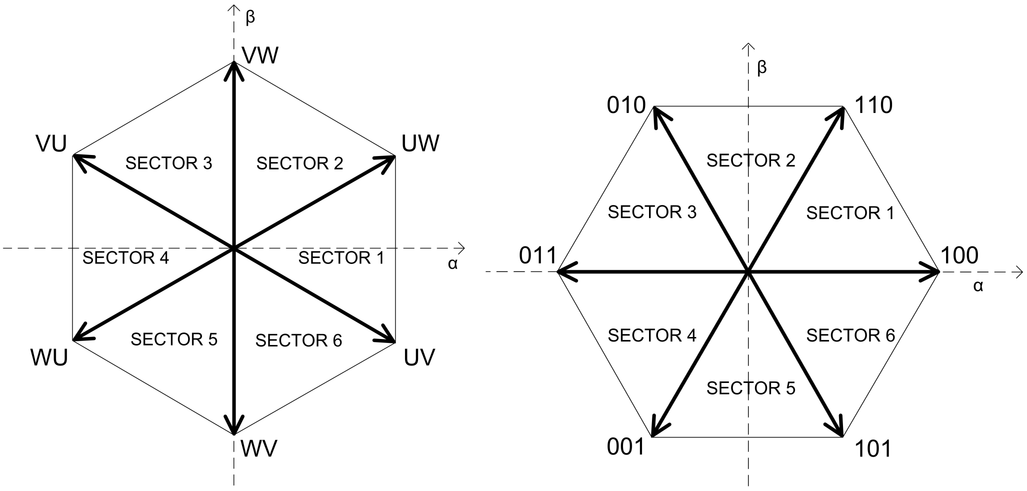
$$\vec{i}_{i,ref} = M_i \sqrt{\frac{2}{3}} \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix} \begin{bmatrix} \cos(\omega_i t + \varphi_i) \\ \cos\left(\omega_i t + \varphi_i - \frac{2\pi}{3}\right) \\ \cos\left(\omega_i t + \varphi_i - \frac{4\pi}{3}\right) \end{bmatrix} \tag{2.33}$$

Here, M_i and M_o are modulation degrees for input current and output voltage modulation. Because space vector modulation is used for the virtual inverter, maximal available modulation degree is $M_{o,max} = 1.15470...$. The maximal available modulation degree for the virtual rectifier is $M_{i,max} = 1$.

$$M_o = \frac{\hat{v}_o}{\frac{1}{2}v_{DC}} \tag{2.34}$$

$$M_i = \frac{\hat{i}_i}{i_{DC}} \tag{2.35}$$

These mentioned reference vectors can be then synthesized using the virtual rectifier and virtual inverter switching state vectors. The switching state vectors for the virtual rectifier are in fig. 2.16a and tab. 2.3. The switching state vectors for the virtual inverter are in fig. 2.16b and tab. 2.4.



(a) Virtual Rectifier SVM

(b) Virtual Inverter SVM

Figure 2.16: Indirect Space Vector Modulation

Vector Name $\vec{s}$	s_α	s_β	Active Switches
UV	$\sqrt{\frac{3}{2}}$	$-\sqrt{\frac{1}{2}}$	$S_{U+},\ S_{V-}$
UW	$\sqrt{\frac{3}{2}}$	$\sqrt{\frac{1}{2}}$	$S_{U+},\ S_{W-}$
VW	0	$\sqrt{2}$	$S_{V+},\ S_{W-}$
VU	$-\sqrt{\frac{3}{2}}$	$\sqrt{\frac{1}{2}}$	$S_{V+},\ S_{U-}$
WU	$-\sqrt{\frac{3}{2}}$	$-\sqrt{\frac{1}{2}}$	$S_{W+},\ S_{U-}$
WV	0	$-\sqrt{2}$	$S_{W+},\ S_{V-}$

Table 2.3: Table of Virtual Rectifier Switching States

Using a linear combination of two virtual inverter switching state vectors, it is possible to synthesize an output reference voltage vector. The vectors v_{i1} and v_{i2} are chosen according to a virtual inverter modulation sector (tab. 2.5).

$$\vec{v}_{o,ref} = d_{i1} \cdot \vec{v}_{i1} + d_{i2} \cdot \vec{v}_{i2} \tag{2.36}$$

Similarly, using linear combination of two virtual rectifier switching state vectors, it is possible to synthesize an input reference current vector. The vectors i_{r1} and i_{r2} are chosen

Vector Name $\vec{s}$	s_α	s_β	Active Switches
100	$2 \cdot \sqrt{\frac{2}{3}}$	0	$S_{U+},\ S_{V-},\ S_{W-}$
110	$\sqrt{\frac{2}{3}}$	$\sqrt{2}$	$S_{U+},\ S_{V+},\ S_{W-}$
010	$-\sqrt{\frac{2}{3}}$	$\sqrt{2}$	$S_{U-},\ S_{V+},\ S_{W-}$
011	$-2 \cdot \sqrt{\frac{2}{3}}$	0	$S_{U-},\ S_{V+},\ S_{W+}$
001	$-\sqrt{\frac{2}{3}}$	$-\sqrt{2}$	$S_{U-},\ S_{V-},\ S_{W+}$
101	$\sqrt{\frac{2}{3}}$	$-\sqrt{2}$	$S_{U+},\ S_{V-},\ S_{W+}$

Table 2.4: Table of Virtual Inverter Switching States

according to a virtual inverter modulation sector (tab. 2.5).

$$\vec{i}_{i,ref} = d_{r1} \cdot \vec{i}_{r1} + d_{r2} \cdot \vec{i}_{r2} \tag{2.37}$$

Sector	Virtual Inverter		Virtual Rectifier	
	$\vec{v}_{i1}$	$\vec{v}_{i2}$	$\vec{i}_{r1}$	$\vec{i}_{r2}$
1	100	110	UV	UW
2	110	010	UW	VW
3	010	011	VW	VU
4	011	001	VU	WU
5	001	101	WU	WV
6	101	100	WV	UV

Table 2.5: Table of Vector Interpretation According to Particular Sectors.

In contrast with standard Active Front End topology, there is a synchronization needed between the first and the second stage (because there is not a DC storage part present). There are more possible switching patterns. In this work two of them will be implemented according to [6]. A `non-optimized` switching pattern follows:

$\vec{i}_{r1}$		$\vec{i}_{r2}$					$\vec{i}_{r1}$		
$\frac{1}{2} \cdot d'_{r1}$		d'_{r2}					$\frac{1}{2} \cdot d'_{r1}$		
$\vec{v}_{i1}$	$\vec{v}_{i2}$	$\vec{v}_0$	$\vec{v}_{i2}$	$\vec{v}_{i1}$	$\vec{v}_{i1}$	$\vec{v}_{i2}$	$\vec{v}_0$	$\vec{v}_{i2}$	$\vec{v}_{i1}$
$\frac{1}{2} \cdot d_{11}$	$\frac{1}{2} \cdot d_{12}$	$\frac{1}{2} \cdot d_0$	$\frac{1}{2} \cdot d_{22}$	$\frac{1}{2} \cdot d_{21}$	$\frac{1}{2} \cdot d_{21}$	$\frac{1}{2} \cdot d_{22}$	$\frac{1}{2} \cdot d_0$	$\frac{1}{2} \cdot d_{12}$	$\frac{1}{2} \cdot d_{11}$

Table 2.6: Non-optimized Switching Pattern

The pattern table presents a sequence of the switching state vectors of the virtual inverter and virtual rectifier together with duty-cycle period ratios. The duty-cycle ratios are calculated as follows:

$$d_{11} = d_{r1} \cdot d_{i1} \tag{2.38}$$

$$d_{12} = d_{r1} \cdot d_{i2} \tag{2.39}$$

$$d_{21} = d_{r2} \cdot d_{i1} \tag{2.40}$$

$$d_{22} = d_{r2} \cdot d_{i2} \tag{2.41}$$

$$d_0 = 1 - d_{11} - d_{12} - d_{21} - d_{22} \tag{2.42}$$

Because the length of a zero vector is defined by the virtual inverter, it is no need to include a zero vector by the virtual rectifier. Therefore the duty-cycle period is adjusted on the whole period:

$$d'_{r1} = \frac{d_{r1}}{d_{r1} + d_{r2}} \tag{2.43}$$

$$d'_{r2} = \frac{d_{r2}}{d_{r1} + d_{r2}} \tag{2.44}$$

It is possible to see, that while using the non-optimized pattern, there is a switching present during zero vector presence. The main task of the pattern optimizing is to remove this switching. Optimized pattern differs according to a sector sum (input sector number + output sector number). If the sum is even, the pattern is same as when non-optimized. If the sum is odd, the pattern is changed.

$\vec{i}_{r1}$		$\vec{i}_{r2}$					$\vec{i}_{r1}$		
$\frac{1}{2} \cdot d'_{r1}$		d'_{r2}					$\frac{1}{2} \cdot d'_{r1}$		
$\vec{v}_{i2}$	$\vec{v}_{i1}$	$\vec{v}_0$	$\vec{v}_{i1}$	$\vec{v}_{i2}$	$\vec{v}_{i2}$	$\vec{v}_{i1}$	$\vec{v}_0$	$\vec{v}_{i1}$	$\vec{v}_{i2}$
$\frac{1}{2} \cdot d_{12}$	$\frac{1}{2} \cdot d_{11}$	$\frac{1}{2} \cdot d_0$	$\frac{1}{2} \cdot d_{21}$	$\frac{1}{2} \cdot d_{22}$	$\frac{1}{2} \cdot d_{22}$	$\frac{1}{2} \cdot d_{21}$	$\frac{1}{2} \cdot d_0$	$\frac{1}{2} \cdot d_{11}$	$\frac{1}{2} \cdot d_{12}$

Table 2.7: Optimized Switching Pattern for Odd Sum of Sectors

Both, the optimized and the non-optimized pattern lookup tables are to be found in appendix B.

To determine the influences on maximal voltage transfer ratio, we have to determine influences on the maximal u_{DC}, using energy conservation law.

$$P_i = P_{DC} \tag{2.45}$$

$$\frac{3}{2}\hat{i}_i \hat{u}_i \cos\varphi_i = u_{DC} i_{DC} \tag{2.46}$$

$$u_{DC} = \frac{3}{2}\frac{\hat{i}_i}{i_{DC}} \cos\varphi_i = \frac{3}{2} M_i \cos\varphi_i \tag{2.47}$$

From equation (2.47), we can determine that maximal voltage transfer ratio is available, for $M_i = 1$ and $\cos\varphi_i = 1$.

2.2.3 Matrix Converter Restrictions

To build a Matrix Converter few restrictions have to be met. The ideal bidirectional switch has to be replaced with a combination of real switches, the input current distortion, caused by switching, has to be reduced by a filter and additional protection clamp circuit has to be added for a case of converter failure.

The schema of converter built at Czech Technical University at Department of Electric Drives and Traction is in fig. 2.17.

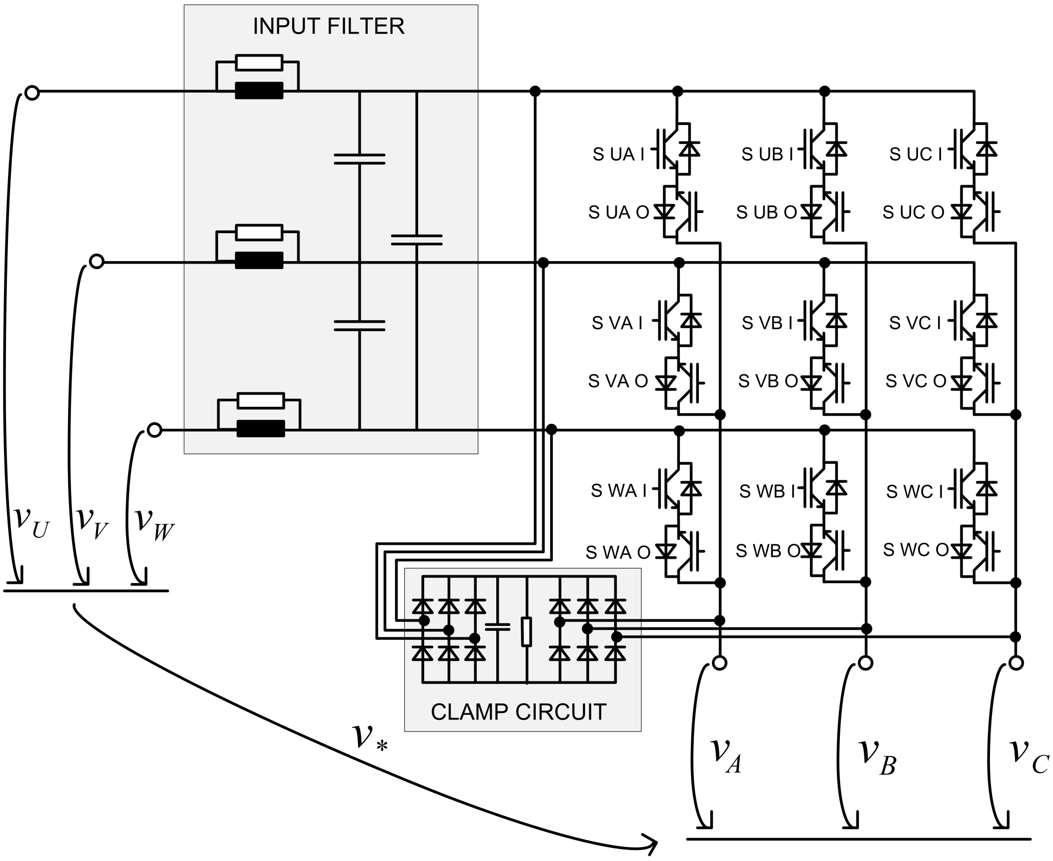


Figure 2.17: Schema of Matrix Converter built at Czech Technical University

2.2.3.1 Input Filter

The main task of the input filter (also drawn in fig. 2.17) is to reduce a current distortion caused by switching. Generally, a LC filter is chosen for this purpose. The converter prototype uses a series damped filter. More information about the input filter can be found in [6] and [5].

2.2.3.2 Protection Clamp Circuit

The main task of the protection clamp circuit is to protect the converter from overvoltages and overcurrents. It consists of two rectifier bridges with a use of fast recovery diodes, which are connected to a storage capacitor. This capacitor should be able to store all energy from parasite inductions in case of an emergency breaking of the converter. More information on this topic can be found at [6] and [5].

2.2.3.3 Bidirectional Switches

There is a number of possibilities of emulating a bidirectional switch. The known opportunities are in fig. 2.18.

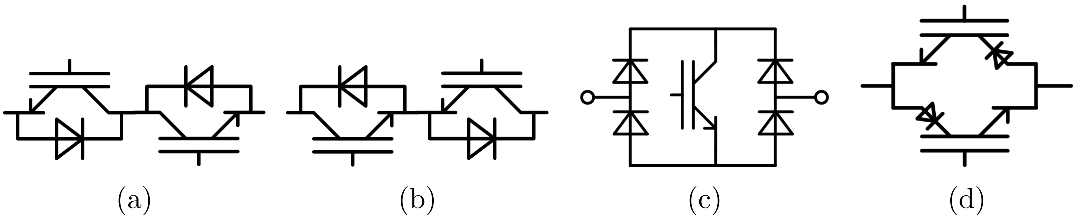


Figure 2.18: Practical configuration of a bidirectional switch: (a) common collector IGBT switch, (b) common emitter IGBT switch, (c) diode bridge switch, (d) antiparallel RGIGBT switch

The most common are configurations (a) and (b). An advantage of a common collector configuration is that drivers for both switches can use a same ground. At the university prototype, there is a common emitter configuration used.

The configuration (c) is not applicable for MC because there is not a safe commutation guaranteed. The configuration (d) is a configuration using novel components RGIGBT. Because of their relatively high price and relatively poor properties, they are not usable at the moment.

To obtain more information about bidirectional switches, please refer to [5].

2.2.4 Commutation Methods

To switch safely from one input phase to another without short circuiting them, there is a commutation demanded.

There are several commutation methods available. The commutation methods differ on the basis of how a commutation switching sequence is decided. There are current, voltage or combined commutation methods [5]. Every single commutation cycle is done mostly in two or four steps, depending on the particular method.

In this part, there will be two commutation methods explained, which will also be implemented into a modulator design in a next chapter. To obtain more information about commutation methods, please refer to [5], [6] and [7].

2.2.4.1 Commutation Schema

For commutation to be explained, a commutation scheme is presented in fig. 2.19.

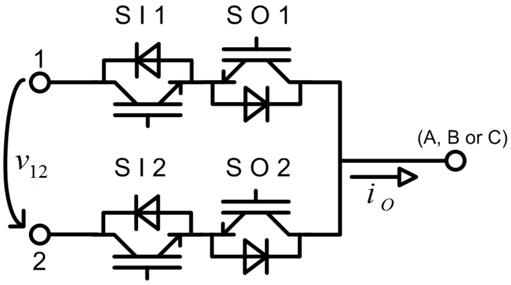


Figure 2.19: Commutation Principle Scheme

The terminal 1 in the figure is a terminal that is initially connected to an output terminal (U,V or W). The terminal 2 is a terminal to which we want the output to be connected. The voltage v_{12} is decisive for voltage commutation methods. The current i_O is decisive for current commutation methods.

2.2.4.2 Four Step Voltage Commutation

The principle of this method is to switch from one input phase to second, using their line-to-line voltage polarity information. It is switched according to the voltage polarity, not to open short-circuiting line. Because the output current polarity is unknown, at least two switches have to be turned on to ensure a current path.

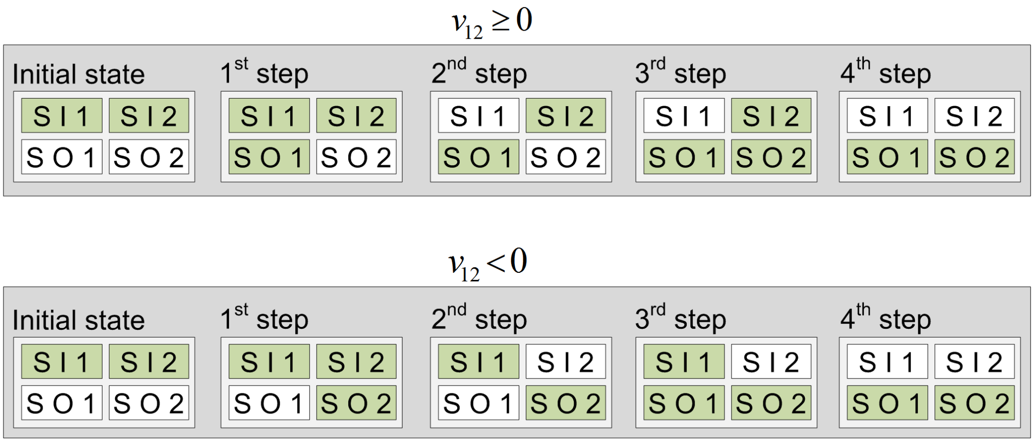

$v_{12} \geq 0$
Initial state: 🟩 S I 1, 🟩 S I 2, S O 1, S O 2
1st step: 🟩 S I 1, 🟩 S I 2, 🟩 S O 1, S O 2
2nd step: S I 1, 🟩 S I 2, 🟩 S O 1, S O 2
3rd step: S I 1, 🟩 S I 2, 🟩 S O 1, 🟩 S O 2
4th step: S I 1, S I 2, 🟩 S O 1, 🟩 S O 2
$v_{12} < 0$
Initial state: 🟩 S I 1, 🟩 S I 2, S O 1, S O 2
1st step: 🟩 S I 1, 🟩 S I 2, S O 1, 🟩 S O 2
2nd step: 🟩 S I 1, S I 2, S O 1, 🟩 S O 2
3rd step: 🟩 S I 1, S I 2, 🟩 S O 1, 🟩 S O 2
4th step: S I 1, S I 2, 🟩 S O 1, 🟩 S O 2

Figure 2.20: Four Step Voltage Commutation Algorithm

When the line-to-line voltage v_{12} is positive, we have to deny a state, when S I 1 and S O 2 are turned on at the same time. Similarly, when the line-to-line voltage v_{12} is negative, we have to deny a state, when S I 2 and S O 1 are turned on at the same time.

Fig. 2.20 shows, the four step voltage commutation method algorithm. The green background of a switch signifies that IGBT is turned on.

2.2.4.3 Two Step Current Commutation

The principle of the two step current commutation method is to switch from one input phase to another, using an output current polarity information. There are always either input or output transistors switched on during commutation, therefore the possibility of short-circuiting of input phases is denied.

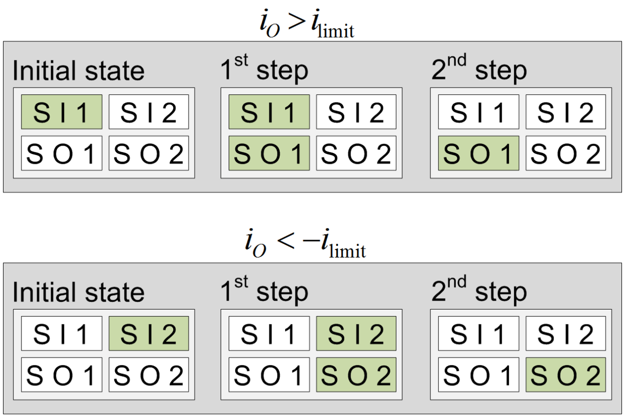


Figure 2.21: Two Step Current Commutation Principle

When output current polarity i_O is positive, S I switches are turned on during commutation. Similarly, when output current polarity i_O is negative, S O switches are turned on during commutation.

Fig. 2.21 shows, the two step current commutation method algorithm. It is clear that there must be some hysteresis in a current polarity measurement to insure an output current path. When output current is in a range from $-i_{limit}$ to i_{limit}, both S I and S O need to be turned on. To provide commutation in this range, either an extra logic is needed [5] or different voltage commutation method is used [7]. A four step voltage commutation was chosen as a supplement for the implementation of this method.

2.2.4.4 Output Current Polarity Measurement

One of the possibilities of measuring the output current polarity is to compare the voltage drop on turned-on IGBT/diode component to the voltage that is slightly higher then the minimal IGBT voltage drop V_{T0}. The principle scheme of comparing is in fig. 2.22.

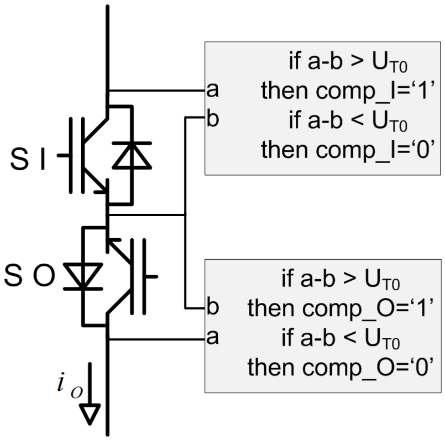


Figure 2.22: Current Polarity Measurement Principle

When the voltage drop on a turned-on component is lower than V_{T0}, the current is too small for polarity to be determined or the current flows through the diode. When the voltage drop on component turned on is higher than V_{T0}, the current flows through the IGBT.

The possible states determined from this measurement are summarized in tab. 2.8.

`comp_I`	`comp_O`	**State**
0	0	i_O polarity is not determined
0	1	i_O is negative
1	0	i_O is positive
1	1	detection error is present

Table 2.8: Current Polarity Interpretation

Chapter 3

New Modulation Method for VSI

In this chapter new VSI modulation method "NewM" will be presented. This method should be able to reduce the converter switching losses that will be analytically examined and then proven using a simulation model in `Matlab/Simulink` using `Plexim Plecs`.

3.1 Principle

As it was already mentioned, the task of this method is to reduce the switching losses. The basic strategy of this method is to specify the reference common voltage $v_{0,ref}$ so that the number of switching during a period is lower, then by the other common methods.

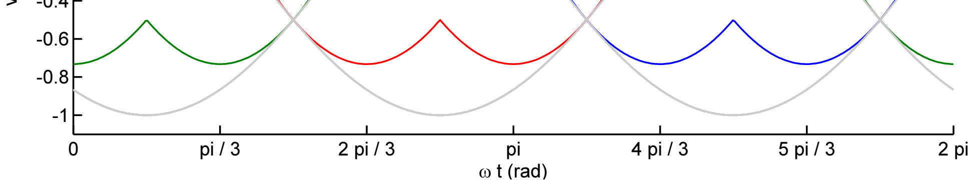


Figure 3.1: New Modulation Reference Function

For the reference output voltage functions, we have decided that non-switching cycles have to be chosen symmetrically. Therefore, every pair of transistors (every half bridge) is not switching during one third of the reference function period. After a short analysis, a switching curve was chosen where there is no switching present during the third of period with maximal values of reference voltage.

For the reference-voltage system (2.4), there was a common-voltage function chosen:

$$v_{0,ref} = V_{DC} - \max(v_{1,ref}, v_{2,ref}, v_{3,ref}) = V_{DC}\left(1 - \max\left(\frac{v_{1,ref}}{V_{DC}}, \frac{v_{2,ref}}{V_{DC}}, \frac{v_{3,ref}}{V_{DC}}\right)\right) \tag{3.1}$$

Using this system, curves in fig. 3.1 were plotted. In the figure, there are curves $v_{1,ref} - v_{0,ref}$, $v_{2,ref} - v_{0,ref}$, $v_{3,ref} - v_{0,ref}$ (grey), curve $v_{0,ref}$ (black) and modulated curves $v_{1,ref}$, $v_{2,ref}$, $v_{3,ref}$ as functions of ωt.

After an examination performed on the other modulation methods, it is possible to state that the maximal modulation degree for this modulation method is the same as for SVM and 3RDM.

$$M_{max} = \frac{1}{\cos 30^o} = \frac{2}{\sqrt{3}} = 1.15470\ldots \tag{3.2}$$

In fig. 3.2 the space vector trajectory in $\alpha, \beta, 0$-coordinates is shown. It is possible to see, that the vector trajectory sticks to the upper walls of “switching cube”.

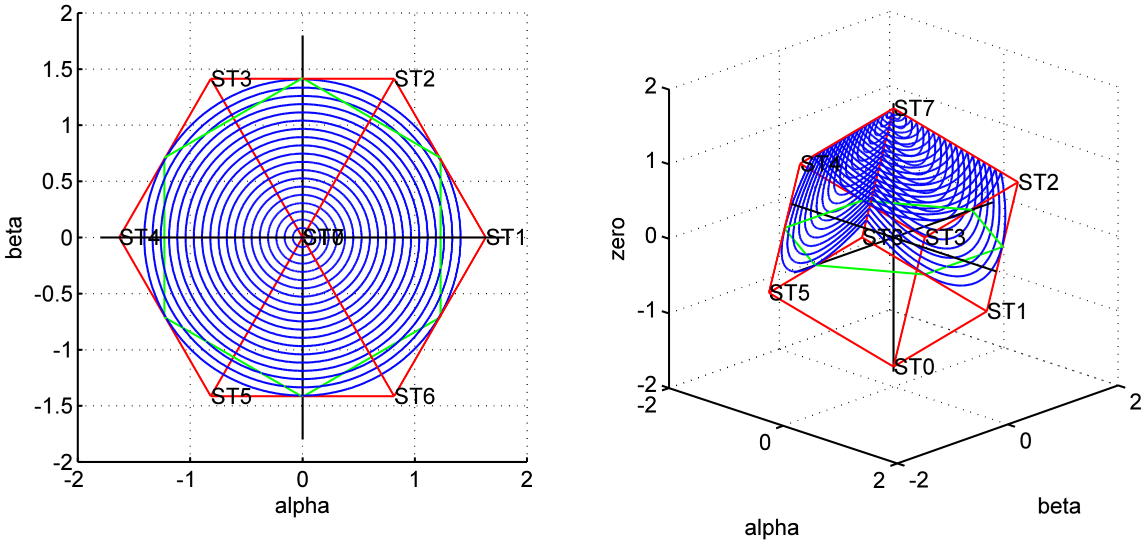


Figure 3.2: Space Vector of New Modulation in $\alpha, \beta, 0$ Coordinates.

3.2 Analytic Derivation of Switching Losses

In general, switching energy loss is a function of an emitter-to-collector voltage, a junction temperature, an emitter current, a load type etc. To calculate switching energy loss, we have to linearize the switching energy loss as a function of an emitter current. To achieve that, we set the voltage constant (the DC input voltage changes during operation just in the case of AM), set the load as inductive and the junction temperature as constant. The dissipated energy in IGBT caused by switching is then:

$$E_{loss,T,ON} = k_{T,ON} \cdot I_E \tag{3.3}$$

$$E_{loss,T,OFF} = k_{T,OFF} \cdot I_E \tag{3.4}$$

Dissipated energy in a diode is caused only by the reverse recovery losses, when the diode is switching off.

$$E_{loss,D,OFF} = k_{D,OFF} \cdot I_E \tag{3.5}$$

We will concentrate just on one phase (one pair of transistors) as the rest is symmetric. One phase output current function can be written as:

$$i(t) = \hat{\imath} \sin(\omega t + \varphi) \tag{3.6}$$

where $\omega = 2\pi f = \frac{2\pi}{T}$.

The dissipated energy calculation was already mentioned at (3.3), (3.4) and (3.5). Because of switching frequency f_{sw} being much higher than output current frequency f, we can approximate emitter current in a moment of switching as a mean value during one switching period.

$$E_{loss} = k\frac{1}{T_{sw}} \int_{t_0}^{t_0+T_{sw}} i_E(t)\mathrm{d}t \tag{3.7}$$

Mean dissipated power during one output current period can be then calculated as:

$$P_{loss} = \frac{1}{T} \sum E_{loss} \tag{3.8}$$

$$P_{loss} = \frac{1}{T}\left(k\frac{1}{T_{sw}} \int_{t_0}^{t_0+T_{sw}} i_E(t)\mathrm{d}t + k\frac{1}{T_{sw}} \int_{t_0+T_{sw}}^{t_0+2T_{sw}} i_E(t)\mathrm{d}t + k\frac{1}{T_{sw}} \int_{t_0+2T_{sw}}^{t_0+3T_{sw}} i_E(t)\mathrm{d}t + \cdots\right) \tag{3.9}$$

$$P_{loss} = k\frac{1}{T}\frac{1}{T_{sw}}\left(\int_{t_0}^{t_0+T_{sw}} i_E(t)\mathrm{d}t + \int_{t_0+T_{sw}}^{t_0+2T_{sw}} i_E(t)\mathrm{d}t + \int_{t_0+2T_{sw}}^{t_0+3T_{sw}} i_E(t)\mathrm{d}t + \cdots\right) \tag{3.10}$$

$$P_{loss} = k\frac{1}{T}\frac{1}{T_{sw}} \int_{t_0}^{t_0+T} i_E(t)\mathrm{d}t \tag{3.11}$$

$$P_{loss} = k f_{sw} f \int_{t_0}^{t_0+T} i_E(t)\mathrm{d}t \tag{3.12}$$

We can apply this to determine any of the switching losses during one period.

3.2.1 Switching Losses by Sinus Modulation

First, we will determine every part of the switching losses for both components. When the current is positive the switching losses are present on upper IGBT and diode down.

$$P_{loss,T,UP} = k_{T,ON} f_{sw} f \int_{\frac{\varphi}{\omega}}^{\frac{\pi+\varphi}{\omega}} i(t)\mathrm{d}t + k_{T,OFF} f_{sw} f \int_{\frac{\varphi}{\omega}}^{\frac{\pi+\varphi}{\omega}} i(t)\mathrm{d}t \tag{3.13}$$

$$P_{loss,T,UP} = (k_{T,ON} + k_{T,OFF}) f_{sw} f \int_{\frac{\varphi}{\omega}}^{\frac{\pi+\varphi}{\omega}} i(t)\mathrm{d}t \tag{3.14}$$

$$P_{loss,D,DOWN} = k_{D,OFF} f_{sw} f \int_{\frac{\varphi}{\omega}}^{\frac{\pi+\varphi}{\omega}} i(t)\mathrm{d}t \tag{3.15}$$

When the current is negative the switching losses are present on the upper diode and the IGBT down.

$$P_{loss,T,DOWN} = (k_{T,ON} + k_{T,OFF}) f_{sw} f \left(\int_{0}^{\frac{\varphi}{\omega}} -i(t)\mathrm{d}t + \int_{\frac{\pi+\varphi}{\omega}}^{\frac{2\pi}{\omega}} -i(t)\mathrm{d}t \right) \tag{3.16}$$

$$P_{loss,D,UP} = k_{D,OFF} f_{sw} f \left(\int_{0}^{\frac{\varphi}{\omega}} -i(t)\mathrm{d}t + \int_{\frac{\pi+\varphi}{\omega}}^{\frac{2\pi}{\omega}} -i(t)\mathrm{d}t \right) \tag{3.17}$$

Solving these equations, we obtain switching losses present on the upper IGBT/DIODE part $P_{SM,loss,UP}$ and on the IGBT/DIODE down $P_{SM,loss,DOWN}$.

$$P_{SM,loss,UP} = \frac{\hat{i} f_{sw} (k_{T,ON} + k_{T,OFF} + k_{D,OFF})}{\pi} \tag{3.18}$$

$$P_{SM,loss,DOWN} = \frac{\hat{i} f_{sw} (k_{T,ON} + k_{T,OFF} + k_{D,OFF})}{\pi} \tag{3.19}$$

It is possible to see, that the mean dissipated switching power is symmetrically split between the individual parts when Sinus Modulation is used. The losses also do not depend on the angle φ.

3.2.2 Switching Losses due to New Modulation

To derive switching losses due to New Modulation, we have to define a mask function that masks the moments when there are no switching losses present. This function will be then applied on the current function.

This masking function $\Theta(t)$ is defined using Heaviside Theta function $\mathrm{H}(x)$.

$$\Theta(t) = \mathrm{H}(t) - \mathrm{H}\left(t - \frac{\pi}{6\omega}\right) + \mathrm{H}\left(t - \frac{2\pi}{3\omega} - \frac{\pi}{6\omega}\right) \tag{3.20}$$

Applying this masking function on the equations (3.14), (3.15), (3.16) and (3.17), we obtain following equations.

$$P_{loss,T,UP} = (k_{T,ON} + k_{T,OFF}) f_{sw} f \int_{\frac{\varphi}{\omega}}^{\frac{\pi+\varphi}{\omega}} \Theta(t) i(t)\mathrm{d}t \tag{3.21}$$

$$P_{loss,D,DOWN} = k_{D,OFF} f_{sw} f \int_{\frac{\varphi}{\omega}}^{\frac{\pi+\varphi}{\omega}} \Theta(t) i(t)\mathrm{d}t \tag{3.22}$$

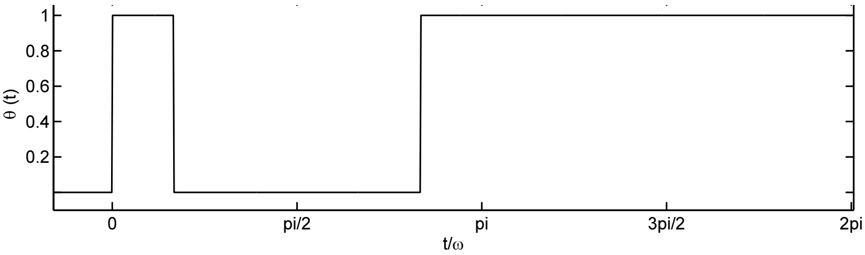


Figure 3.3: Masking Function

$$P_{loss,T,DOWN} = (k_{T,ON} + k_{T,OFF}) f_{sw} f \left(\int_0^{\frac{\varphi}{\omega}} -\Theta(t)i(t)\mathrm{d}t + \int_{\frac{\pi+\varphi}{\omega}}^{\frac{2\pi}{\omega}} -\Theta(t)i(t)\mathrm{d}t \right) \tag{3.23}$$

$$P_{loss,D,UP} = k_{D,OFF} f_{sw} f \left(\int_0^{\frac{\varphi}{\omega}} -\Theta(t)i(t)\mathrm{d}t + \int_{\frac{\pi+\varphi}{\omega}}^{\frac{2\pi}{\omega}} -\Theta(t)i(t)\mathrm{d}t \right) \tag{3.24}$$

By solving these equations, we obtain switching losses present on the upper IGBT/DIODE part $P_{NewM,loss,UP}$ and on the IGBT/DIODE down $P_{NewM,loss,DOWN}$. To simplify these equations a ξ ratio is defined:

$$\xi = \frac{k_{D,OFF}}{k_{T,ON} + k_{T,OFF}} \tag{3.25}$$

The power dissipation equations are:

$$P_{NewM,loss,UP} = (k_{T,ON} + k_{T,OFF}) \hat{i} f_{sw} \cdot \\ \cdot \frac{2 + 2\xi - \sqrt{3}\cos\varphi + (1+\xi)\left(\left(1 + cos\left(\varphi + \frac{\pi}{6}\right)\right) \mathrm{H}(6\varphi - 5\pi) + \left(-1 + \cos\left(\varphi - \frac{\pi}{6}\right)\right) \mathrm{H}(6\varphi - \pi)\right)}{2\pi} \tag{3.26}$$

$$P_{NewM,loss,DOWN} = (k_{T,ON} + k_{T,OFF}) \hat{i} f_{sw} \cdot \\ \cdot \frac{2 + 2\xi - \sqrt{3}\xi\cos\varphi + (1+\xi)\left(\left(1 + cos\left(\varphi + \frac{\pi}{6}\right)\right) \mathrm{H}(6\varphi - 5\pi) + \left(-1 + \cos\left(\varphi - \frac{\pi}{6}\right)\right) \mathrm{H}(6\varphi - \pi)\right)}{2\pi} \tag{3.27}$$

From these equations, it is possible to see that the losses are not split equally between the upper part and the lower part while using New Modulation.

3.2.3 Relative Switching Losses

The relative losses can be defined as a ratio between losses due to New Modulation and losses due to Sinus Modulation.

$$P_{rel,loss,UP} = \\ = \frac{2 + 2\xi - \sqrt{3}\cos\varphi}{2(1+\xi)} + \frac{1}{2}\left(\left(1 + cos\left(\varphi + \frac{\pi}{6}\right)\right) \mathrm{H}(6\varphi - 5\pi) + \left(-1 + \cos\left(\varphi - \frac{\pi}{6}\right)\right) \mathrm{H}(6\varphi - \pi)\right) \tag{3.28}$$

$$P_{rel,loss,DOWN} =$$
$$= \frac{2+2\xi-\sqrt{3}\xi\cos\varphi}{2(1+\xi)} + \frac{1}{2}\left(\left(1+cos\left(\varphi+\frac{\pi}{6}\right)\right)\mathrm{H}(6\varphi-5\pi)+\left(-1+\cos\left(\varphi-\frac{\pi}{6}\right)\right)\mathrm{H}(6\varphi-\pi)\right) \tag{3.29}$$

And total switching losses:

$$P_{rel,loss} =$$
$$= 1-\frac{\sqrt{3}}{4}\cos\varphi+\left(\frac{1}{2}+\frac{1}{2}\cos\left(\varphi+\frac{\pi}{6}\right)\right)\mathrm{H}(6\varphi-5\pi)+\left(-\frac{1}{2}+\frac{1}{2}\cos\left(\varphi-\frac{\pi}{6}\right)\right)\mathrm{H}(6\varphi-\pi) \tag{3.30}$$

This was plotted into a fig. 3.4 for different ξ as a function of $\cos\varphi$.

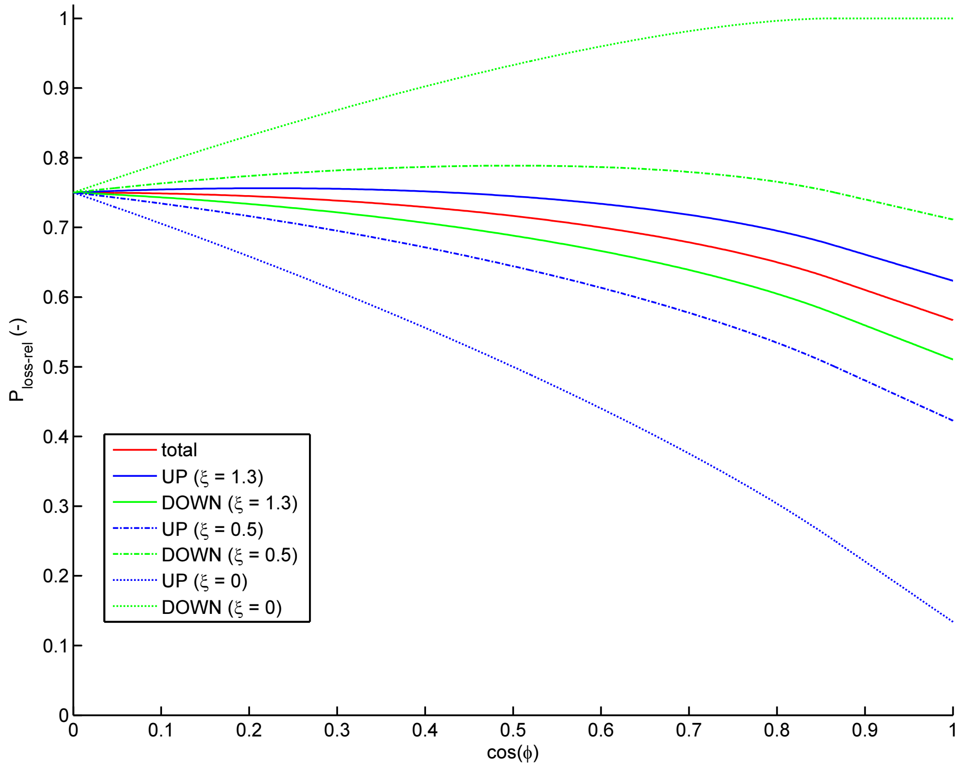


Figure 3.4: Relative Losses Derived Analytically.

In general, it is possible to see a significant reduction of switching losses. In total they are between ca. 60 and 70 percent of switching losses due to Sinus Modulation. It is also possible to see that losses are lowest near $\cos\varphi = 1$.

For $\xi = 1$, the losses are equally distributed. It is the case, when the diode losses are the same as the IGBT losses. In a case, when the IGBT losses are higher ($\xi > 1$) or when the IGBT losses are lower ($\xi < 1$) than the diode losses, we can see that losses are distributed unequally.

3.3 Verification on Simulation

To verify derived results a simulation model was created in `Matlab/Simulink` using `Plexim Plecs`.

A simulation of the converter in a steady state was performed to determine its behavior.

3.3.1 Simulation Parameters

The following simulation parameters were chosen.

Variable	Value	Description
V_{input}	460 V	DC source voltage
M	1	modulation degree
$\hat{i}_{AC}$	70 A	output AC current amplitude
f	50 Hz	output current frequency
f_{sw}	12 kHz	switching frequency
T_j	120^o C	junction temperature
L_f	4 mH	filtering inductor

Table 3.1: Simulation Parameters.

3.3.2 Simulation Model

In principle, the simulation model consists of four parts: a reference function generator, a PWM modulator, a converter model, and a `Matlab` script.

The reference function generator generates a reference function with a use of a particular modulation. This reference function is routed to an input of PWM modulator, where pulses for each IGBT are generated. This pulses signal comes as an input signal into the converter model, where the converter variables are computed. Some important variables are then sent to the `Matlab Workspace`, where they are used for further computation described in the `Matlab` script.

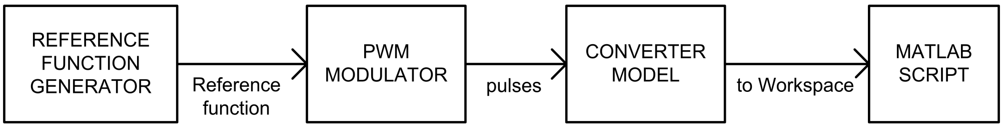


Figure 3.5: Simulation principle.

3.3.2.1 Reference Function Generator

There are three different reference-function generators, each for a different modulation method. A reference function vector is defined as follows:

$$\begin{bmatrix} v_{1,ref}(t) \\ v_{2,ref}(t) \\ v_{3,ref}(t) \end{bmatrix} = MV_{DC} \begin{bmatrix} \sin(\omega t) \\ \sin(\omega t + \frac{2\pi}{3}) \\ \sin(\omega t + \frac{4\pi}{3}) \end{bmatrix} + v_{0,ref} \tag{3.31}$$

For **Sinus Modulation**, the reference common voltage is defined as $v_{0,ref} = 0$ and a reference function generation schema is in fig. 3.6.

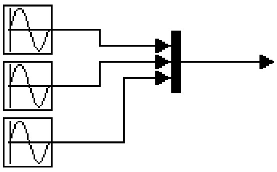


Figure 3.6: SM Reference Function Generator

For **space vector modulation**, the reference common voltage is defined as (3.32) and a reference function generation schema is in fig. 3.7.

$$v_{0,ref}(t) = -\frac{\max(v_{1,ref}(t), v_{2,ref}(t), v_{3,ref}(t)) + \min(v_{1,ref}(t), v_{2,ref}(t), v_{3,ref}(t))}{2} \tag{3.32}$$

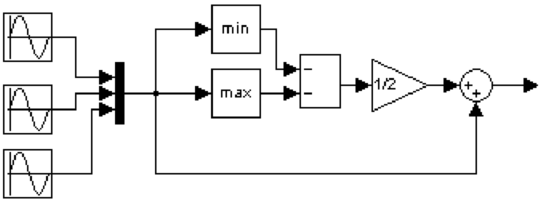


Figure 3.7: SVM Reference Function Generator

For **new modulation**, there is the reference common voltage defined as (3.33) and a reference function generation schema is in fig. 3.8.

$$v_{0,ref} = V_{DC}\left(1 - \max\left(\frac{v_{1,ref}}{V_{DC}}, \frac{v_{2,ref}}{V_{DC}}, \frac{v_{3,ref}}{V_{DC}}\right)\right) \tag{3.33}$$

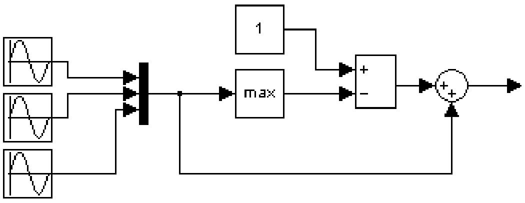


Figure 3.8: NewM Reference Function Generator

3.3.2.2 PWM Modulator

PWM modulator was created as a `Plecs Circuit`. It consists of factory made PWM units and saturation blocks, which create a matched signal for IGBT transistors in the converter model.

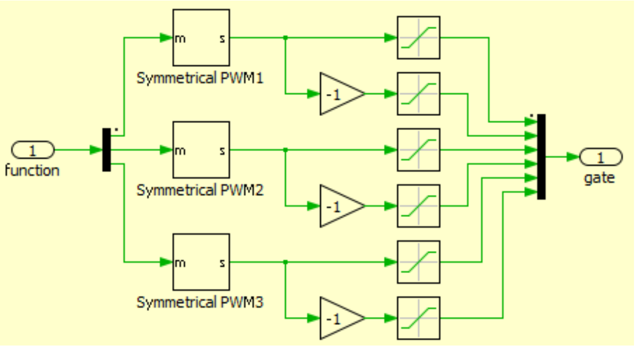


Figure 3.9: Plecs Circuit VSI PWM Modulator.

3.3.2.3 Converter Model

The converter model was also created as a `Plecs Circuit`. IGBT/Diode parts were chosen to emulate real parts. As a load was chosen a 3-phase voltage system with filtering inductors. Output currents and losses on the individual transistors are measured and then sent to `Workspace`.

To simulate losses, a thermal model was also added to the converter model. The thermal sink has a very high thermal capacity, so that the temperature does not change during simulation. There was a losses model attached for IGBT/Diode parts. It was created using a datasheet for the IGBT/Diode part `POWEREX CM100DU-24NFH` [10], which will be used later in practical application.

For steady-state simulation, initial conditions of a current on inductors and a load source voltage function have to be stated. They result as a solution for three phases static

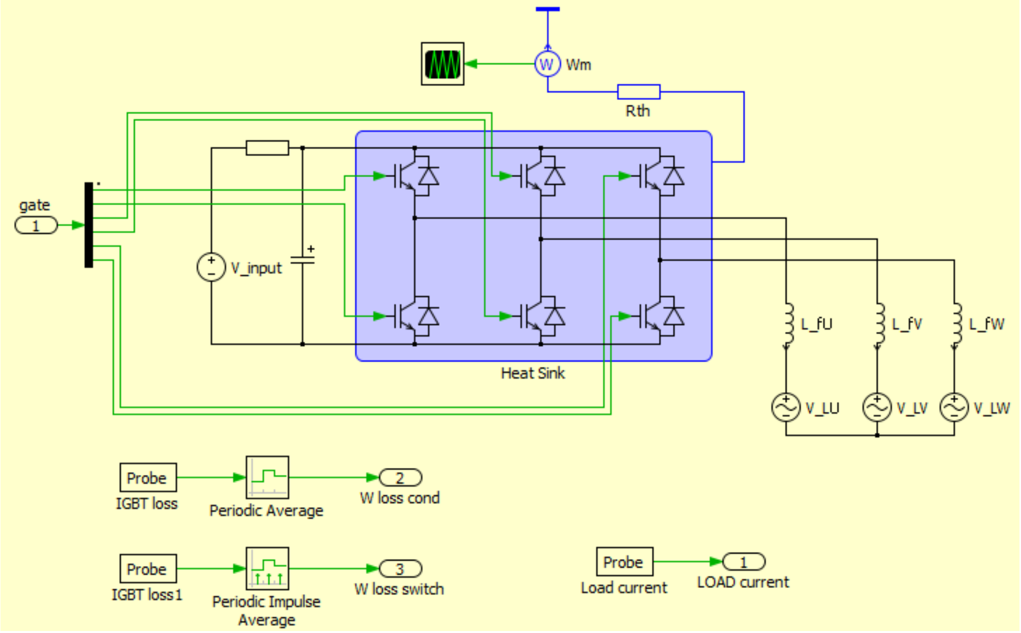


Figure 3.10: Plecs Circuit VSI Converter Model.

system analysis.

The fist harmonic for the converter output voltages phasor can be written as:

$$\vec{\underline{V}}_1 = \begin{bmatrix} \underline{V}_{1,U} \\ \underline{V}_{1,V} \\ \underline{V}_{1,W} \end{bmatrix} = \hat{v}_{,1} \begin{bmatrix} \mathrm{e}^{\ j0} \\ \mathrm{e}^{\ j\frac{2\pi}{3}} \\ \mathrm{e}^{\ j\frac{4\pi}{3}} \end{bmatrix} = \frac{M \cdot V_{input}}{2} \begin{bmatrix} \mathrm{e}^{\ j0} \\ \mathrm{e}^{\ j\frac{2\pi}{3}} \\ \mathrm{e}^{\ j\frac{4\pi}{3}} \end{bmatrix} \tag{3.34}$$

Output current phasor:

$$\vec{\underline{I}} = \begin{bmatrix} \underline{I}_U \\ \underline{I}_V \\ \underline{I}_W \end{bmatrix} = \hat{i}_{AC} \begin{bmatrix} \mathrm{e}^{\ j(0-\varphi)} \\ \mathrm{e}^{\ j\left(\frac{2\pi}{3}-\varphi\right)} \\ \mathrm{e}^{\ j\left(\frac{4\pi}{3}-\varphi\right)} \end{bmatrix} \tag{3.35}$$

From these two equations, we can determine the load voltage phasor.

$$\vec{\underline{V}}_1 - \vec{\underline{V}}_L = j\omega L_f \cdot \vec{\underline{I}} \tag{3.36}$$

$$\vec{\underline{V}}_L = \vec{\underline{V}}_1 - j\omega L_f \cdot \vec{\underline{I}} \tag{3.37}$$

The current initial values can be then set as:

$$\begin{bmatrix} i_{Lf,U}(0) \\ i_{Lf,V}(0) \\ i_{Lf,W}(0) \end{bmatrix} = \Im\left(\vec{\underline{I}}\right) \tag{3.38}$$

The amplitude of the load source can be set as:

$$\hat{v}_L = \mid \underline{V}_L \mid \tag{3.39}$$

The angle Ψ, between the first harmonic of a converter output voltage and a load voltage source, is:

$$\Psi = Arg\left(\underline{V}_{L,U}\right) \tag{3.40}$$

3.3.2.4 Matlab Script

A task of `Matlab` script is to calculate mean value of switching power loss for different conditions. It also does a THD calculation according to (3.41).

$$\mathrm{THD}(i(t)) = \frac{\sqrt{I_{RMS}^2 - I_{1,RMS}^2}}{I_{1,RMS}} \tag{3.41}$$

3.3.3 Results

For the simulation parameters, result plots were collected. Figures are either in absolute variables or relative variables. Relative variable is a ratio of a variable to the same variable for SM.

In fig. 3.11, the switching power losses are plotted as a function of $\cos\varphi$. From the relative values, it is possible to see, that a potential of power loss savings by the new method is up to 40% in comparison to the other methods. In total power loss is around 15% (fig. 3.12). The highest savings are present, when there is just an active power transmitted.

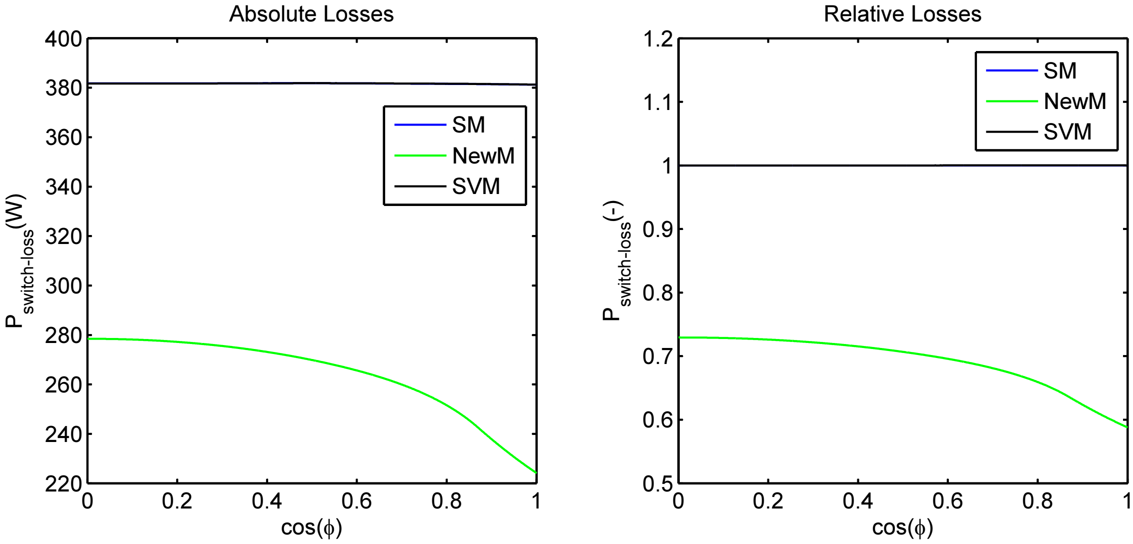


Figure 3.11: Switching Losses of Converter for Different Modulations.

In fig. 3.13 we can see that simulated results are quite matching the analytically derived results. It is also possible to see that the IGBT linearizing coefficient is slightly different from the diode linearizing coefficient. That means that for chosen IGBT/Diode part, there will be losses unequally distributed between part up and part down.

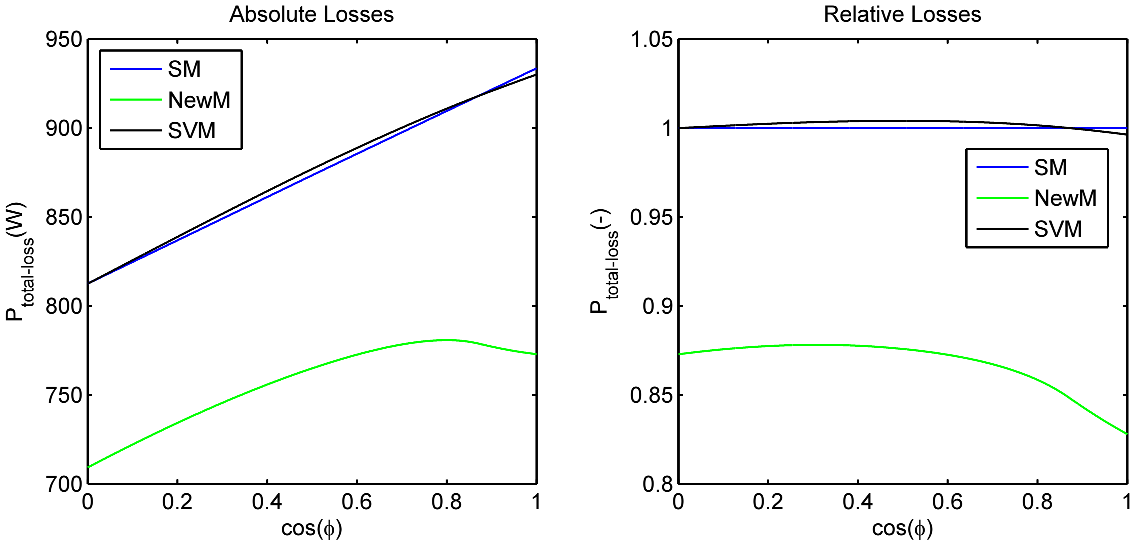


Figure 3.12: Total Losses of Converter for Different Modulations.

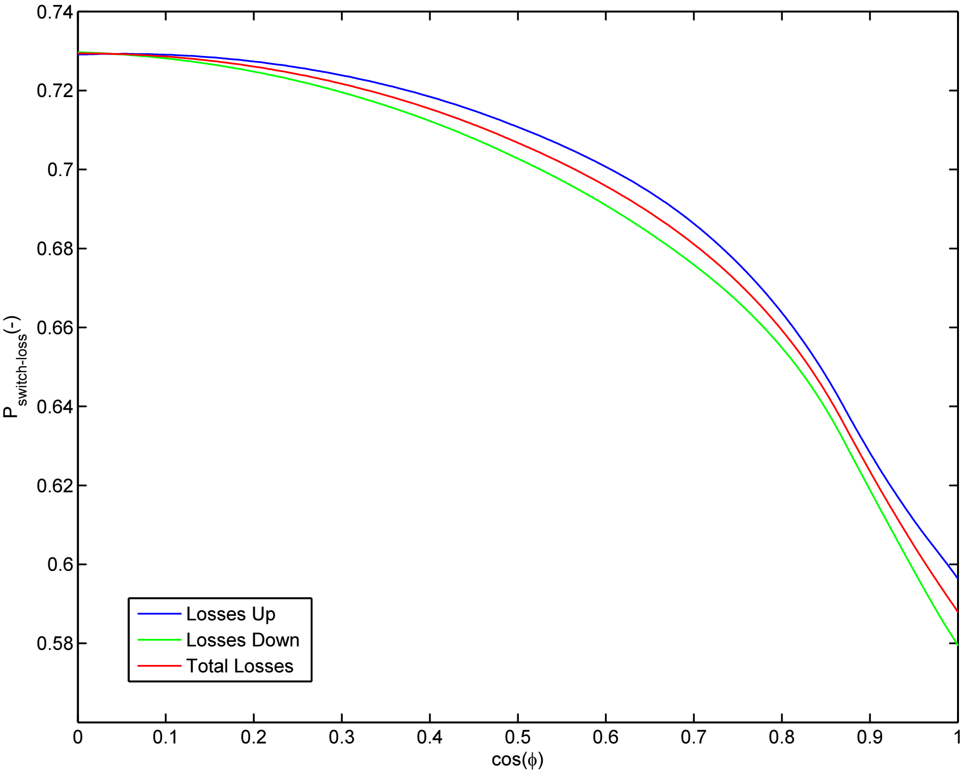


Figure 3.13: Relative Losses and its Distribution using New Modulation.

From the results in fig. 3.14 we can determine, that current THD does not change significantly for different modulation methods.

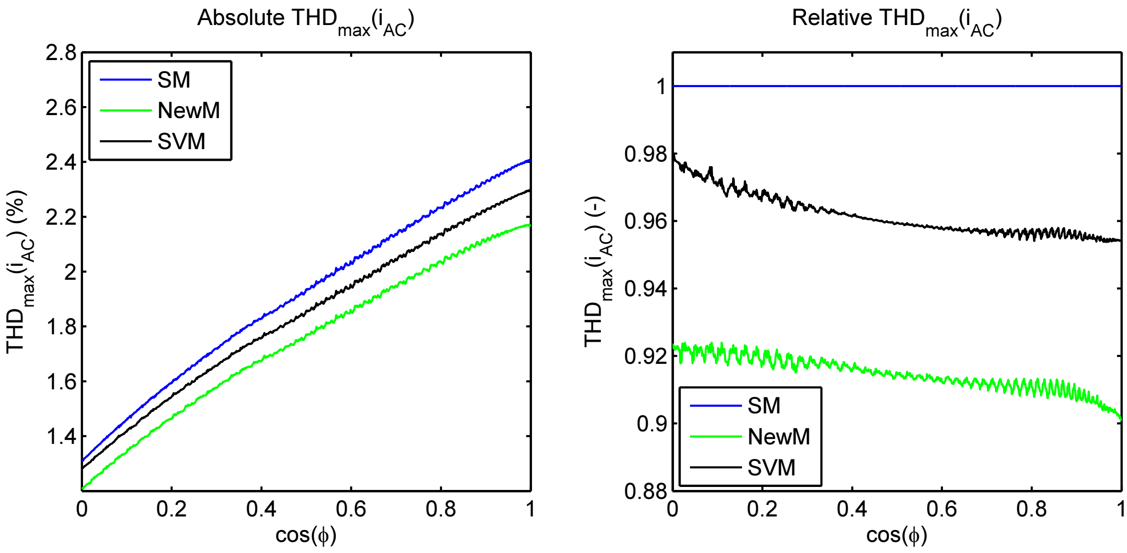


Figure 3.14: Total Harmonic Distortion of Output Current for Different Modulations.

3.4 Conclusion

In this chapter, we have proposed and verified the new modulation method. Verification was done analytically and simulated.

Analytically derived results are fitting results from simulation.

This new method, as expected, delivered significant savings on switching power loss (between 25 and 40%), while other relevant parameters stayed satisfactory (similar THD, maximal modulation degree M_{max} same as by Space Vector Modulation).

The possible difficulty of this method might be non-zero mean value of common voltage, what might cause polarization of oil in motor bearings. This should be examined in a future research.

This modulation method is also applied at the modulator design (in the next chapter).

Chapter 4

Application

To verify and demonstrate functionality of the modulation methods, a practical built of modulators for the converters mentioned was done.

4.1 Control System Hardware

Both of the modulators were described in `VHDL` on `RTD FPGA 6800 Board` [12]. This FPGA board is then connected to a particular Converter sending gate pulses for converter transistors and obtaining error signals. The FPGA board is also connected to CPU board `CMA157886 cpuModules` [13] running `Microsoft DOS` operating system. The CPU board can be also connected to A/D conversion card through a `PCI` port to provide a system for real-time machine control.

The whole control system is displayed in fig. 4.1.

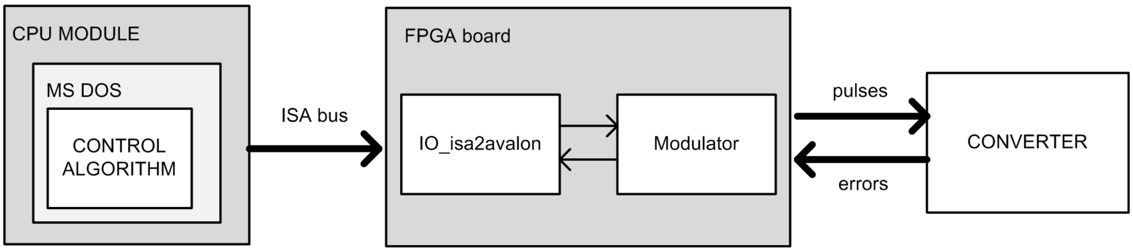


Figure 4.1: Control System Hardware Connection.

The connection via ISA bus might be done using IO space or memory space. For this application, IO space was selected.

4.1.1 isa_io2avalon

All of the modulators are written to communicate via avalon bus to maintain the versatility of modules. To adjust ISA IO bus to avalon bus a bridge unit `isa_io2avalon` was created according to [14]. This unit is displayed in fig. 4.2.

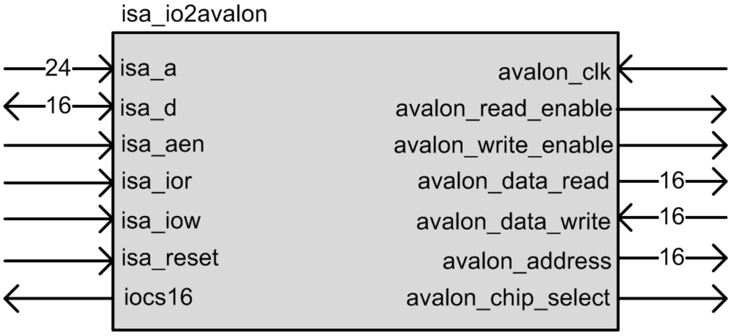


Figure 4.2: Bridge isa_io2avalon.

A transfer cycle begins always with setting the address (`isa_a`) and address enable signal (`isa_aen`). The address is decoded and if it matches the memory hole space, `avalon_chip_select` signal turns to logical `'1'` and `iocs16` is pulled down to evoke the 16-bit data transfer. The lowest five bits of `isa_a` bus are directly connected to the five lowest bits of the avalon address bus. The rest of avalon address bus is directly connected to logical `'0'`. Note that because a 16-bit transfer is present, the lowest address bit is insignificant and should not be connected to a modulator unit.

When a writing cycle is present, the data (`isa_d`) are buffered with a raising edge of signal `isa_iow` and then transfered in next cycle of `avalon_clk` clock to `avalon_data_write` bus and signal `avalon_write_enable` turns to `'1'` for one period of `avalon_clk` clock.

When a reading cycle is present, `isa_ior` signal turns to logical `'0'`, `isa_d` bus is connected with `avalon_data_read` bus and `avalon_read_enable` turns to `'1'`.

To determine a control cycle period length, it is needed to know the reading and writing speed, while using this unit and ISA IO bus. This was determined by a testing script using a 32-bit CPU counter. The resulting times are written in tab. 4.1.

writing cycle	1.6 μs
reading cycle	1.4 μs

Table 4.1: Measured approximate times of writing and reading cycle.

4.2 Voltage Source Inverter

First, a VSI modulator will be presented. It provides three modulation methods (SM, SVM and New Modulation) with a symmetric saw signal and a driver error handling. To provide higher versatility of the modulator, different operation modes are possible. It is possible to operate with or without dead time. With or without using a safe mode, during which a driver error immediately stops the converter.

4.2.1 Converter Loser

4.2.1.1 Description

Converter Loser is a converter created at the Czech Technical University in Prague, Department of Electric Drives and Traction to be used for educational purposes. It consists of eight IGBT pairs with antiparallel diodes [10], a filtering capacitor and SCALE drivers [11]. The whole converter is controlled using `5V TTL` signals for IGBT driving (logical `'1'` = IGBT is turned on) and error handling (logical `'0'` = driver is in error).

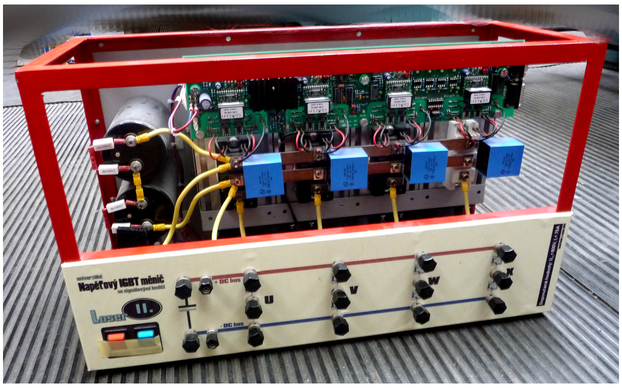


Figure 4.3: Picture of Converter Loser, from [4]

The `SCALE` drivers provide overcurrent and higher IGBT power dissipation protection. In case of failure they automatically wait for a necessary period of time until IGBT is cooled down [11]. The failure flag is removed automatically after the next switching or can be removed manually by confirming the error.

The `SCALE` drivers also provide two modes of operation. A `DEAD TIME MODE`, during which a dead time is generated automatically and `DIRECT MODE` during which a dead time is not generated. Note that it is necessary for this application that `DIRECT MODE` is present.

The important parameters of the converter are summarized in tab. 4.2.

More information about converter Loser can be found in [3] and [4].

4.2.1.2 Connectors

There are two 15 pin `D-SUB` connectors present on the converter. One with a name “PULSES” is a connector, where IGBT control pulses are present. The second one with

nominal input voltage	800 V
nominal input current	70 A
maximal switching frequency	16 kHz

Table 4.2: Nominal Parameters of Converter Loser.

a name "ERROR" is a connector, where error signals are present.

D-SUB pin	signal	BOARD pin
1	IGBT U down	CN6_19
2	IGBT U up	CN6_21
3	IGBT V down	CN6_23
4	IGBT V up	CN6_25
5	IGBT W down	CN6_27
6	IGBT W up	CN6_29
7	IGBT X down	CN6_31
8	IGBT X up	CN6_33
9	GND	CN6_20
10	GND	CN6_22
11	GND	CN6_24
12	GND	CN6_26
13	GND	CN6_28
14	GND	CN6_30
15	GND	CN6_32

Table 4.3: Table of connector "PULSES" pins, their interpretation in DIRECT MODE and connection to FPGA board.

At IGBT signals, gate control pulses are present.

The "ERROR" connector consists of an error flag register, where each bit signalizes, where an error is currently present. The error state of converter can be confirmed turning "error confirmation pin" to logical `'1'`.

If pulses on some transistor are blocked by the driver logic, "pulses blocked" signal turns to logical `'1'`. If at least one transistor from the error flag is in an error state, "at least one error pin" turns to logical `'1'`. These two signals are not connected to the FPGA board.

Note that the `+15V` pins should not be physically connected to FPGA board. Doing so could damage the board.

D-SUB pin	signal	BOARD pin
1	error U down	CN6_01
2	error U up	CN6_03
3	error V down	CN6_05
4	error V up	CN6_07
5	error W down	CN6_09
6	error W up	CN6_11
7	error X down	CN6_13
8	error X up	CN6_15
9	GND	CN6_02
10	+15V	NC
11	pulses blocked	NC
12	at least 1 error	NC
13	error confirmation	CN6_10
14	+15V	NC
15	GND	CN6_14

Table 4.4: Table of connector "ERROR" pins, their interpretation and connection to FPGA board.

4.2.2 Modulator Description

In this section modulator functionality will be explained.

4.2.2.1 VSI_top

`VSI_top` module is a top module of a voltage source inverter modulator entity. This entity interconnects all the others inside. Its simplified routing is displayed in fig. 4.4.

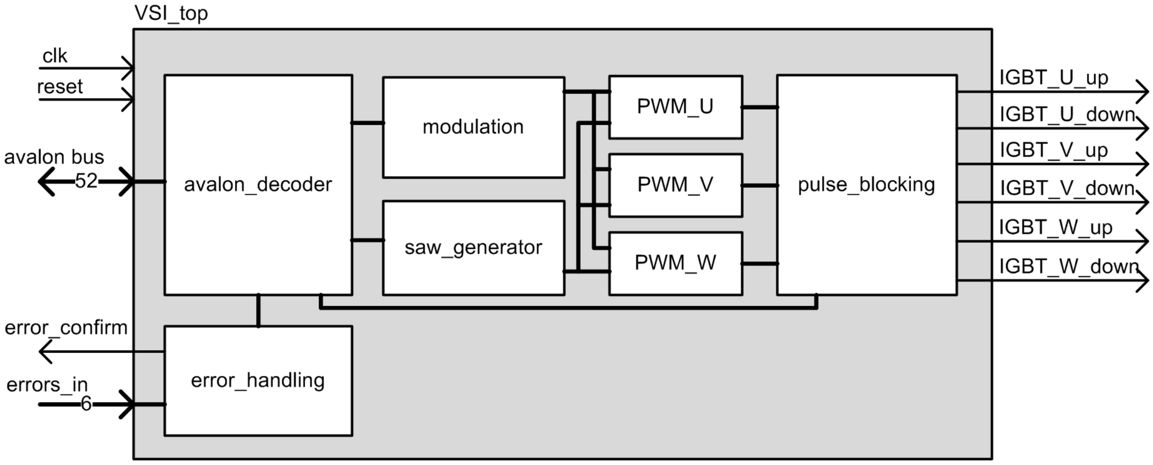


Figure 4.4: Top modulator entity VSI_top.

Input signals of top unit are consisting of `clk`, which is FPGA board clock, `reset`, which resets the entity, avalon bus, which is routed to avalon_decoder and errors_in signal, which stores the information about driver state. Output signals are error_confirm, which confirms the error driver flags and six IGBT gate signals, which can be then routed to three pairs of transistors (U, V, W or X).

Signals clk and reset are routed through the whole design. They will not be drawn in unit schemes.

The whole control is done in unit avalon_decoder, which provides settings for all the other units. It also provides important global signals such as programming_mode (some of the settings apply just in this mode, changes are made immediately - not waiting for saw synchronizing signal, pulses are blocked during this mode) and pulse_blocking signal which simply blocks pulses. The majority of communication in the top level module is done using enable-acknowledgement handshake.

The units modulation, saw_generator and PWM units provide chosen modulation from obtained data.

4.2.2.2 avalon_decoder

As it was already mentioned avalon_decoder is the main control entity in the project.

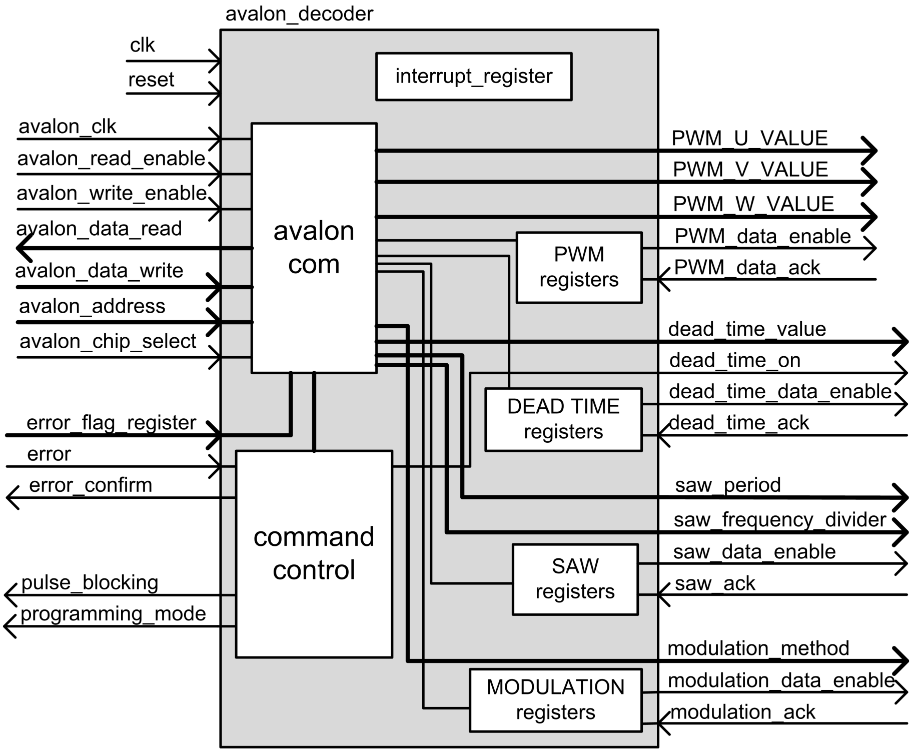


Figure 4.5: Control entity avalon_decoder.

The whole communication on avalon bus is done inside the avalon_com process. This process allows reading from shadow registers and the interrupt register and writing to shadow registers and the command register. When a new shadow register is loaded a *_new signal turns to logical '1' for one clock period. This is then further processed with *_registers process. Some registers (saw data registers and PWM data registers) need an enabling command to inhibit *_new signal.

All actual information about converter modulator and converter is stored in the interrupt_register.

The command_control process is called by command_new signal. Depending on data written into this register, a command is executed. Possible commands are: pulse blocking on (sets '1' in pulse_blocking signal immediately), pulse blocking off (sets '0' in pulse_blocking signal immediately), programming mode on (sets '1' in programming_mode signal immediately), programming mode off (sets '0' in programming_mode signal immediately), saw data enable command (sets '1' in saw_new signal for one clock period), PWM data enable command (sets '1' in PWM_new signal for one clock period), error confirm (confirms error, if error signal is '0' and safe mode is turned on), safe mode on, safe mode off, dead time on (turns dead_time_on to logical '1'), dead time off (turns dead_time_on to logical '0'). Note, that dead_time_on signal applies in PWM units first after programming mode has been turned on.

When *_new signal turns to logical '1', *_registers process sets *_data_enable signal to logical '1' until an acknowledgement signal *_data_ack appears. If you try to write to register, during enable signal is in '1', a writing error occurs in the interrupt register.

4.2.2.3 saw_generator

The main function of the saw_generator unit is to generate a symmetric saw function signal. At the beginning of every period, it produces synchronizing signal saw_sync.

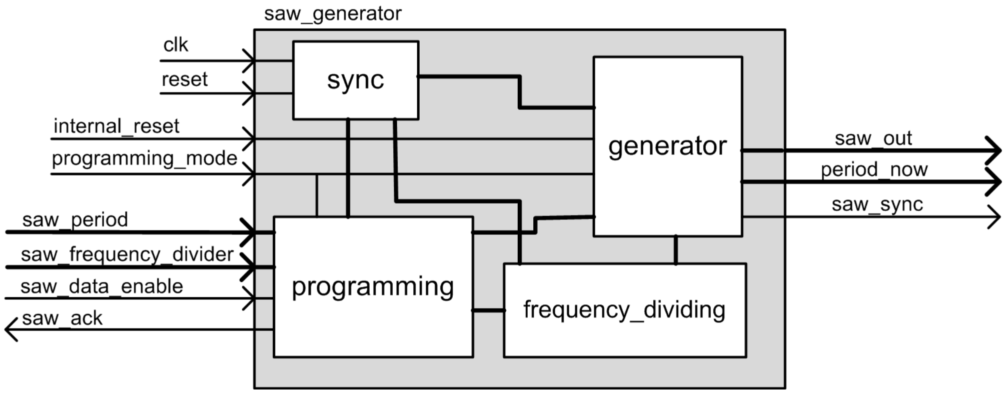


Figure 4.6: Saw function signal generator saw_generator.

Process generator generates symmetric saw function signal saw_out. At beginning of every period it produces synchronizing signal saw_sync. The internal counter is synchronized in process sync.

Process frequency_dividing has a counter counting upto frequency_divider_value. At beginning of every period it generates clk_enable signal, which enables synchronizing of generator's internal counter in process sync.

Process programming is loading new data from shadow registers (from unit avalon_decoder) when the synchronizing signal or the programming mode are present. Actual period information (period_now) is needed to calculate modulation input values. Although it is not recommended to change period value during New Modulation Method presence.

4.2.2.4 modulation

The modulation unit provides modulation calculations from input non modulated data. If input data are already modulated, SM method should be chosen.

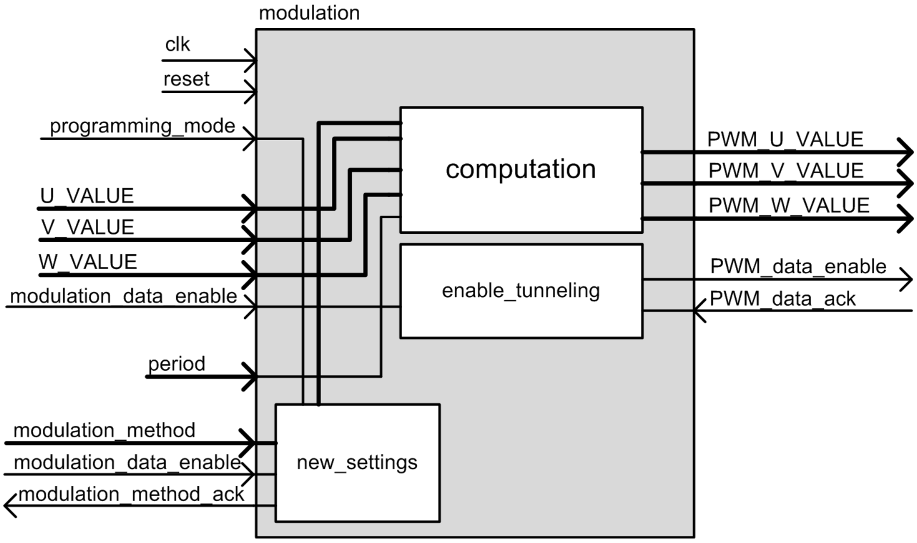


Figure 4.7: Modulation calculations unit modulation.

Process new_settings loads a modulation method value from a shadow register from unit avalon_decoder. A new value is loaded, if modulation_method_enable signal is '1' and programming mode is present or any of new data for modulation are not present (PWM_data_enable = '0').

In process computation, modulation fitted data are calculated according to individual modulation methods equations (SM, SVM, NewM). Depending on a currently chosen modulation method, output values (PWM_*_value) are calculated. The process of the calculation takes two clock cycles, therefore the enable signal has to be delayed. This is provided by the process enable_tunneling.

4.2.2.5 PWM unit

The PWM unit provides Pulse Width Modulation and generates a dead time.

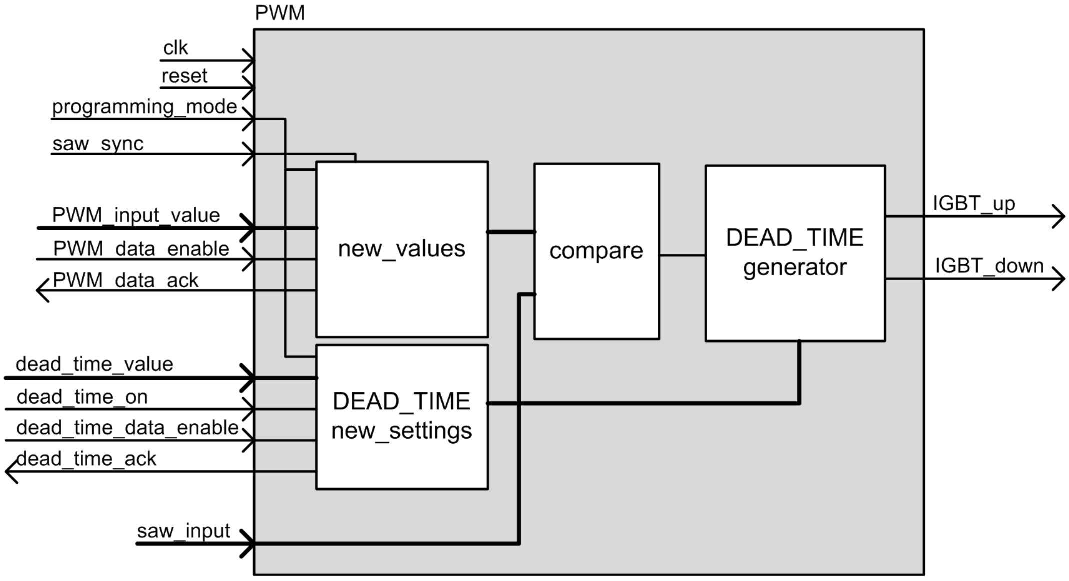


Figure 4.8: Pulse Width Modulation PWM unit.

Processes new_values and DEAD_TIME_new_settings are loading new values from shadow register using enable-acknowledgement handshake. The loading of new values is present only during programming mode or when saw_sync signal turns '1'.

By comparing saw_input signal with PWM_value signal, compared signal is produced. Using this signal the IGBT pulses signals are then generated in the DEAD_TIME_generator process. Dead times are added using FSM and counter.

4.2.2.6 pulse_blocking

pulse_blocking unit blocks gate pulse signals, if programming_mode = '1' or pulse_blocking = '1'.

In the process blocking, an output signal is selected. When the pulse blocking signal or the programming mode are present it sets '0' to output, when not, it pipelines the pulse data through.

4.2.2.7 error_handling

This unit is reading the error input signals. In a case of an error presence, it sets the information about an error for avalon_decoder unit.

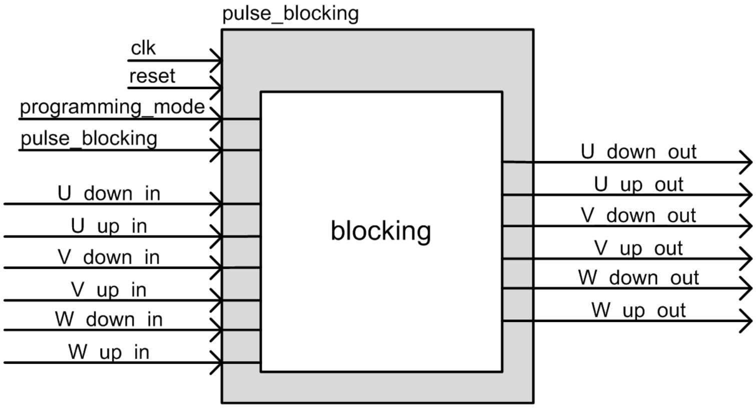


Figure 4.9: Pulse Blocking Unit.

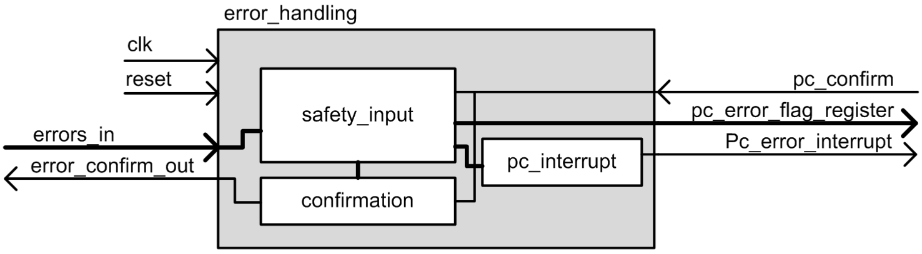


Figure 4.10: Error Handling Unit.

The process `safety_input` is responsible for setting `error_flag_register`. In a case of an error presence at the input, it sets `error_flag_register`. When `PC_confirm` signal from `avalon_decoder` unit is set to `'1'`, it clears the register.

A signal `pc_error_interrupt` includes an information about an error present at some of the drivers. It is set in process `pc_interrupt`.

In the moment, a confirmation request comes from `avalon_decoder`, `error_confirm_out` signal is set to logical `'1'`. When error flag register is cleared, the signal is deasserted.

This unit is working with logical `'1'` for an error being present at the input. In a case, when logical `'0'` states an error present (e.g. converter Loser is used), it is necessary to negate the signal in a top module.

4.2.3 VSI Registers Description

Here, the VSI shadow registers will be presented. These registers are providing the communication with the unit.

Interrupt Register 0x1

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	-	R	R	R	R	R	R	R	R	R	R	R	R	R
-	-	-	DTO	SM	DTE	ME	PVE	SE	DE	PM	PB	ML	PVL	DTL	SL

Interrupt Register basically provides all the necessary actual information about the converter.

SL SAW LOADED
The saw registers are already loaded to operate.

DTL DEAD TIME LOADED
The dead time registers are already loaded to operate.

PVL PWM VALUES LOADED
The PWM values registers are already loaded to operate.

ML MODULATION LOADED
The modulation settings are already loaded to operate.

PB PULSE BLOCKING
Pulse blocking mode is present (might also be a result of an error).

PM PROGRAMMING MODE
Programming mode is present. Pulses are blocked. Some settings apply only in programming mode.

DE DRIVER ERROR
Some driver error is present. In a case of safe mode operation, the blocking mode is turned on.

SE SAW WRITING ERROR
This error occurs when it is tried to write into saw registers, before new values are loaded.

PVE PWM WRITING ERROR
This error occurs when it is tried to write into PWM value registers, before new values are loaded.

ME MODULATION WRITING ERROR
This error occurs when it is tried to write into modulation register, before new values are loaded.

DTE DEAD TIME WRITING ERROR
This error occurs when it is tried to write into dead time register, before new values are loaded.

SM SAFE MODE
Safe mode is present. In case of a driver error, pulses are automatically blocked.

DTO DEAD TIME ON
Indicates that dead time is being generated.
This setting is recommended unless a used converter is capable of dead time generation.

Command Register 0x0

15 -	14 -	13 -	12 -	11 -	10 -	9 -	8 -	7 -	6 -	5 -	4 -	3 W	2 W	1 W	0 W
-	-	-	-	-	-	-	-	-	-	-	-	CR3	CR2	CR1	CR0

CR3-0	**Description**
0x1	BLOCK PULSES Turns pulse blocking on.
0x2	UNBLOCK PULSES Turns pulse blocking off.
0x3	PROGRAMMING MODE ON Turns programming mode on.
0x4	PROGRAMMING MODE OFF Turns programming mode off.
0x5	SAW DATA ENABLE Requests new saw values to be loaded (applies first in new saw period).
0x6	PWM DATA ENABLE Requests new PWM values to be loaded (applies first in new saw period).
0x7	SAFE MODE ON Turns safe mode on (automatic pulse blocking in a case of a driver error).
0x8	SAFE MODE OFF Turns safe mode off.
0x9	ERROR CONFIRMATION Confirms error, in case the source of error was removed.
0xA	DEAD TIME ON Turns dead time generation on (applies first in programming mode).
0xB	DEAD TIME OFF Turns dead time generation off (applies first in programming mode).

Modulation Method Register 0x2

15 -	14 -	13 -	12 -	11 -	10 -	9 -	8 -	7 -	6 -	5 -	4 -	3 -	2 -	1 W/R	0 W/R
-	-	-	-	-	-	-	-	-	-	-	-	-	-	MR1	MR0

MR1-0	**Description**
'00'	SINUS MODULATION PWM signals are not being changed
'01'	SPACE VECTOR MODULATION PWM signals are recalculated according to SVM
'10'	NEW METHOD MODULATION PWM signals are recalculated according to NewM During this method, any changes of the saw signal frequency are not recommended.

Dead Time Value Register 0x3

15 -	14 -	13 -	12 -	11 -	10 -	9 -	8 -	7 W/R	6 W/R	5 W/R	4 W/R	3 W/R	2 W/R	1 W/R	0 W/R
-	-	-	-	-	-	-	-	DT7	DT6	DT5	DT4	DT3	DT2	DT1	DT0

This register determines dead time length, which can be calculated using equation (4.1).

$$T_{DT} = \frac{n+1}{f_{clk}} \tag{4.1}$$

Where n stands for dead time value and f_{clk} for clock frequency.

Changes of this setting apply first after turning programming mode on.

Saw Period Register 0x4

15 W/R	14 W/R	13 W/R	12 W/R	11 W/R	10 W/R	9 W/R	8 W/R	7 W/R	6 W/R	5 W/R	4 W/R	3 W/R	2 W/R	1 W/R	0 W/R
R15	R14	R13	R12	R11	R10	R9	R8	R7	R6	R5	R4	R3	R2	R1	R0

This register contains the period of the periodic symmetric sawtooth signal.

It is being loaded at the a beginning of a new saw period. Note that additional command (`SAW DATA ENABLE`) is needed to request the loading of the new values. The `PWM VALUES LOADED` flag is significant first after the command has been send.

The signal frequency calculation is explained in the next section.

Saw Frequency Divider Register 0x5

15 -	14 -	13 -	12 -	11 -	10 -	9 -	8 -	7 W/R	6 W/R	5 W/R	4 W/R	3 W/R	2 W/R	1 W/R	0 W/R
-	-	-	-	-	-	-	-	R7	R6	R5	R4	R3	R2	R1	R0

This register contains a value of frequency divider for sawtooth signal.

It is being loaded at a beginning of a new saw period. Note that additional command (`SAW DATA ENABLE`) is needed to request the loading of the new values. The `PWM VALUES LOADED` flag is significant first after the command has been sent.

The sawtooth signal frequency calculation follows:

$$f_{saw} = \frac{f_{clk}}{2(p+1)(d+1)} \tag{4.2}$$

Where f_{clk} is the FPGA clock frequency, p is the period value and d is the frequency divider value.

Error Register - 0x6

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	-	-	-	R	R	R	R	R	R
-	-	-	-	-	-	-	-	-	-	E5	E4	E3	E2	E1	E0

This register provides a possibility to locate at which component a driver error was present. After removing the source of the error, it is needed to clear this register with `ERROR CONFIRMATION` command.

Registers `PWM_U_VALUE`, `PWM_V_VALUE`, `PWM_W_VALUE` - 0xA, 0xB, 0xC

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
W/R	W/R	W/R	W/R	W/R	W/R	W/R	W/R	W/R	W/R	W/R	W/R	W/R	W/R	W/R	W/R
R15	R14	R13	R12	R11	R10	R9	R8	R7	R6	R5	R4	R3	R2	R1	R0

These registers store the PWM input values. In a case of Sinus Modulation operation, an absolute value of the values should not exceed the half of the saw period value. In case of SVM or NewM operation, it should not exceed 0.576 of the saw period value.

The values are in the complement integer number format.

4.2.4 VSI C Header

To use the VSI converter in a control algorithm, a C Header was written.

There are prepared functions for writing and reading particular registers and writing particular commands. It also stores addresses of registers.

The header is included on the CD. It is also printed in appendix C.

4.2.5 VSI Demo Code

In order to verify function of the converter a demo code was written. The demo generates 50 Hertz three phase sine signal, which is present at output. It is also possible to read and write particular registers using this program.

This program with source codes is included on the CD.

4.3 Matrix Converter

The matrix converter modulator will be presented in this part. It provides indirect space vector modulation with four step voltage commutation and two step current commutation. There is also a current-direction recognition included in the modulator design. To keep a higher versatility of the modulator, there are two optional patterns (optimized and non-optimized), and an option of safety mode, during which the converter stops, if a driver error is detected.

4.3.1 Matrix Converter Prototype

Some parts of the matrix converter prototype were already explained in chapter 2.2.3. The full hardware schema is in fig. 2.17.

4.3.1.1 Description

To verify the functionality of the modulator, matrix converter prototype created at Czech Technical University in Prague Department of Electric Drives and Traction will be used. This prototype was built for research purposes.

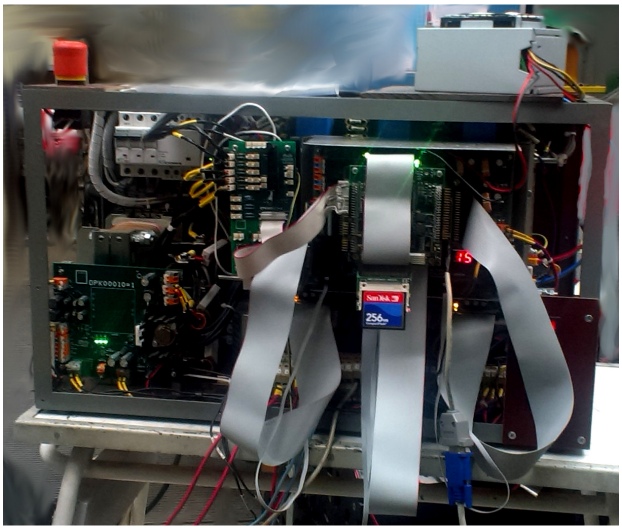


Figure 4.11: Picture of Matrix Converter Prototype

The converter hardware is widely explained at [8]. It consists of 18 IGBT modules `Polovodiče a.s. GD401-70-12`. The nominal properties of the converter are summarized in tab. 4.5.

nominal output current	25 A
nominal phase-to-phase input voltage	400 V
minimal dead time period	850 ns

Table 4.5: Nominal Parameters of Matrix Converter Prototype

4.3.1.2 Connectors

There are three identical driver connectors on the converter prototype (one for every output phase) and one CPU board connector storing information about the input voltage polarity. There are also voltage and current measurement signal connectors wiring signals from LEM probes, which are though irrelevant for the modulator design.

CONNECTOR pin	signal	BOARD pin
01	+5 V	NC
02	+5 V	NC
03	GND	NC
04	+3.3 V	NC
06	not comp_O Y U	CNX_35
08	not comp_I Y U	CNX_33
10	S Y U O	CNX_31
12	S Y U I	CNX_29
14	not error Y U O	CNX_27
16	not error Y U I	CNX_25
18	not comp_O Y V	CNX_23
20	not comp_I Y V	CNX_21
22	S Y V O	CNX_19
24	S Y V I	CNX_17
26	not error Y V O	CNX_15
28	not error Y V I	CNX_13
30	not comp_O Y W	CNX_11
32	not comp_I Y W	CNX_09
34	S Y W O	CNX_07
36	S Y W I	CNX_05
38	not error Y W O	CNX_03
40	not error Y W I	CNX_01
05 - 39 (odd)	GND	CNX_36 - CNX_02(even)

Table 4.6: Matrix Converter Driver Connector Signals and Wiring

In tab. 4.6, the driver connector signal wiring is explained. The connector is a 40-pin flat type connector. There is a same connector on the driver board for every output phase. Depending on the phase (Y), a CNX port of the FPGA board is selected, according to tab. 4.7. The supply pins (+5 V,+3.3 V and GND) have to be connected to some external energy supply.

The input voltage polarity connector wiring is in tab. 4.8. The logical '1' signifies the positive direction of voltage. Note, that the voltage polarity has to be determined in the CPU (using A/D conversion board) and sent to the particular pins of the unit parallel port. This has to be implemented in an application control algorithm.

Y (output phase)	CNX
A	CN4
B	CN5
C	CN6

Table 4.7: FPGA Board Driver Mapping

CPU BOARD PIN	SIGNAL	FPGA BOARD PIN
port 378 (0)	V_UV	CN4_43
port 378 (1)	V_VW	CN4_45
port 378 (2)	V_WU	CN4_47

Table 4.8: Voltage Polarity Measurement Wiring

4.3.2 Modulator Description

In this section modulator functionality will be explained. The modulator unit names and topology were chosen similar to the VSI modulator.

4.3.2.1 MC_top

MC_top module is a top module of matrix converter modulator entity. This entity interconnects all the others inside. Its simplified routing is displayed in fig. 4.12.

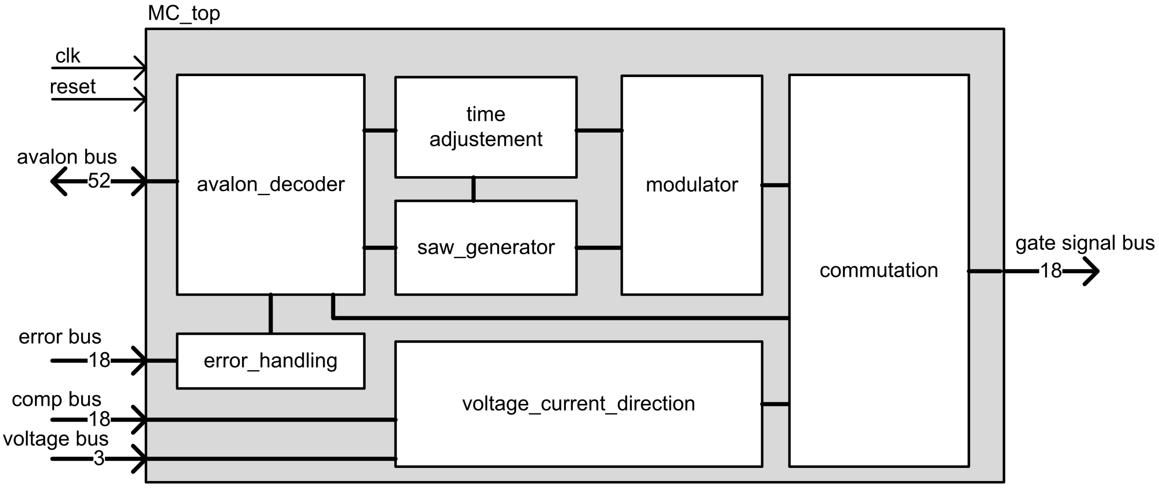


Figure 4.12: Top modulator entity MC_top.

Input signals of top unit are consisting of clk, which is the FPGA board clock, reset, which resets the entity, avalon bus, which is routed to avalon_decoder, error bus (particular signals for every driver) and measurement signals consisting of comp bus (for an output current polarity determination) and voltage measure bus (input voltage polarity information). Output signals are forming gate signal bus. There are particular signals for every transistor gate.

Signals `clk` and `reset` are routed through the whole design. They will not be drawn in the unit schemes.

The whole control is executed in unit `avalon_decoder`, which provides settings for all the other units. It also provides important global signals such as `programming_mode` (some of the settings apply just in this mode, changes are made immediately - not waiting for saw synchronizing signal, pulses are blocked during this mode) and `pulse_blocking` signal which simply blocks pulses. Most of the communication in the top level module is done using enable-acknowledgement handshake.

The units `saw_generator`, `time_adjustement` and `modulator` provide chosen modulation from obtained data. To ensure safe commutation, unit `commutation` is present.

4.3.2.2 avalon_decoder

As it was already mentioned `avalon_decoder` is the main control entity in the project.

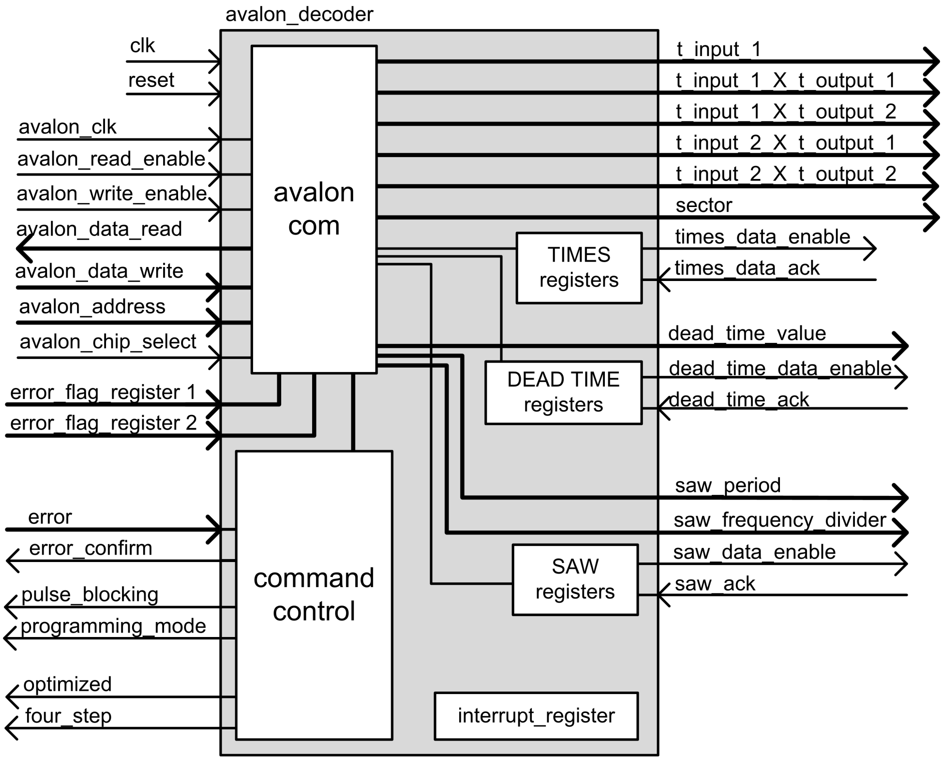


Figure 4.13: Control entity avalon_decoder.

The whole communication on avalon bus is executed inside the `avalon_com` process. This process allows reading from shadow registers and interrupt register and writing to shadow registers and command register. When a new shadow register is loaded a `*_new` signal turns to logical `'1'` for one clock period. This is then further processed with `*_registers` process.

Some registers (saw data registers and times data registers) need an enabling command to inhibit `*_new signal`.

All actual information about converter modulator and converter is stored in `interrupt-_register`.

The `command_control` process is called by `command_new` signal. Depending on data written into this register, a command is executed. Possible commands are: pulse blocking on (sets `'1'` in `pulse_blocking` signal immediately), pulse blocking off (sets `'0'` in `pulse_blocking` signal immediately), programming mode on (sets `'1'` in `programming_mode` signal immediately), programming mode off (sets `'0'` in `programming_mode` signal immediately), saw data enable command (sets `'1'` in `saw_new` signal for one clock period), times data enable command (sets `'1'` in `times_new signal` for one clock period), error confirm (confirms error, if `error` signal is `'0'`, safe mode on, safe mode off, optimized pattern on, optimized pattern off, two step commutation (sets `'0'` in `four_step` signal) and four step commutation (sets `'1'` in `four_step` signal).

When `*_new` signal turns to logical `'1'`, `*_registers` process sets `*_data_enable` signal to logical `'1'` until acknowledgement signal `*_data_ack` is asserted. If you try to write to a register, during an enable signal is in `'1'`, a writing error occurs in interrupt register.

4.3.2.3 saw_generator

Here, a unit adapted from VSI modulator project was used (explained at 4.2.2.3). The functionality of the unit is the same. The only difference is, that the settings of the unit can only be changed in the programming mode.

4.3.2.4 time_adjustement

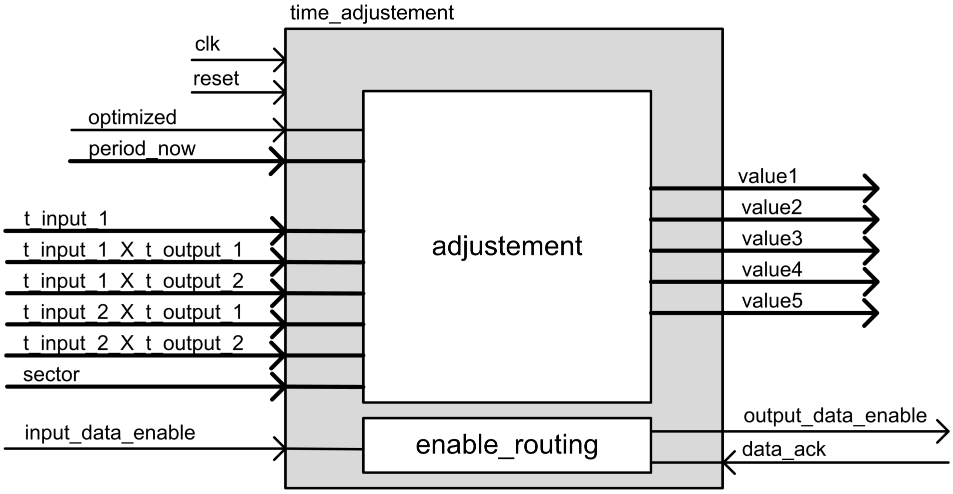


Figure 4.14: Time Period Adjusting Unit time_adjustement.

The `time_adjustement` unit is a unit responsible for preparing data for the modulator comparator, depending on the chosen pattern.

The unit input data are entering the process `adjustement`, where they are prepared according to the chosen switching pattern (if the pattern is optimized, the calculation depends also on the sector sum). The output of this process are data, which are prepared for the `modulator` unit, where they will be compared to the `saw` signal.

The `output_data_enable` signal is asserted by the `input_data_enable` signal in the process `enable_routing`. Because the calculation lasts one clock cycle, the enable signal is delayed by this time period. The signal is deasserted by `data_ack` signal.

4.3.2.5 modulator

The `modulator` unit is providing the switching of particular bidirectional switches in time.

In the process `new_data`, new values are loaded synchronized with a beginning of a saw period (signified by signal `saw_sync`) or during programming mode, if `data_enable` signal is present.

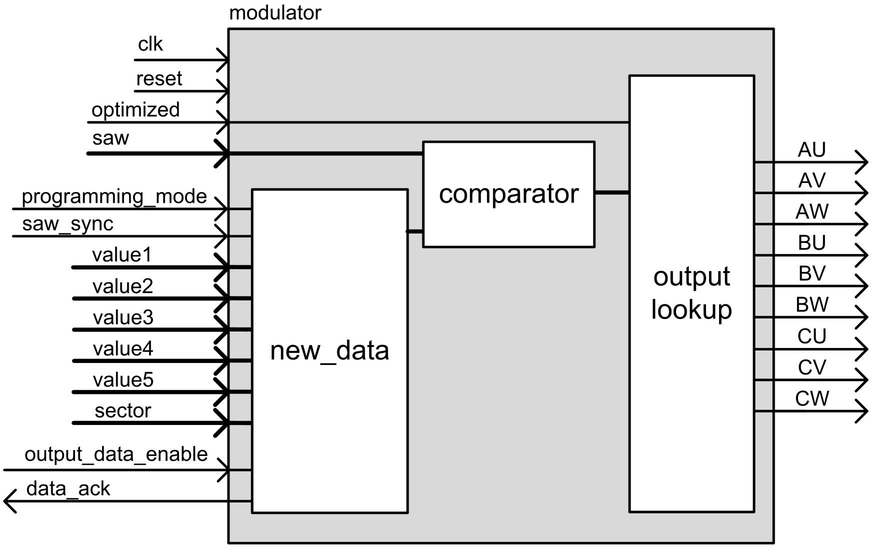


Figure 4.15: Modulation Unit modulator.

In the process `compare`, the loaded data are then compared to `saw` signal and it is determined, which part of pattern is currently present.

Depending on the pattern part information, there is a switching of bidirectional switches generated using lookup table in the process `output_lookup`.

4.3.2.6 commutation

The `commutation` unit provides the commutation (two step or four step) and it blocks pulses if necessary (programming mode or pulse blocking are present).

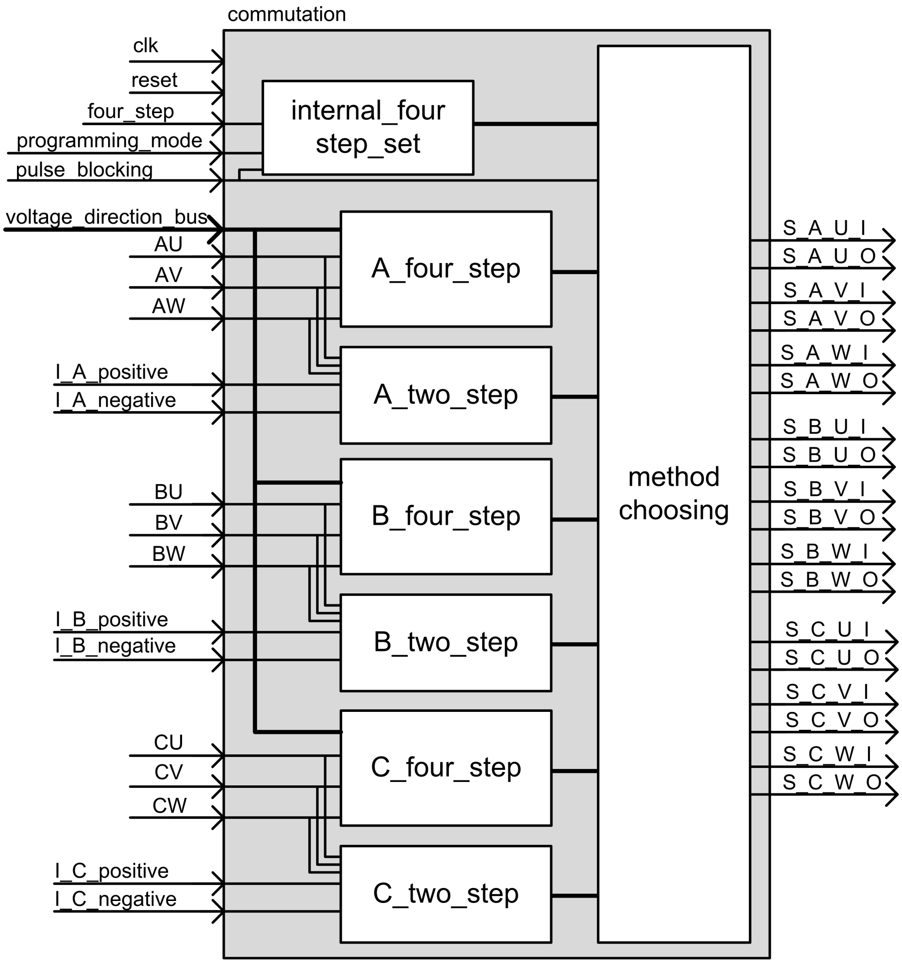


Figure 4.16: Commutation Unit commutation.

The `internal_four_step_set` process is loading new settings of a commutation method, if programming mode is present or there is not any commutation currently present. If the commutation cannot be provided by `*_two_step` unit, `force_4step_*` signal is asserted and this process chooses four step commutation method for the current output phase.

Depending on the commutation method currently chosen by `internal_four_step_set`, the outputs from commutation units (`*_four_step and *_two_step`) are routed to entity output in process `method_choosing`. If programming mode or pulse blocking are present, '0' is routed to the transistor gate signals.

To every `*_four_step` unit, the `voltage direction bus` is connected. It consists of the signals `V_UV`, `V_VW` and `V_WU`, which store information about a voltage polarity. Every *_two_step unit needs information about an output current polarity.

To every commutation unit, dead time signals are routed (dead_time_value, dead_time_enable and dead_time_ack). The dead_time_value signal stores information about the commutation step time period length, these signals are not drawn in the unit scheme figure.

4.3.2.7 four_step_commutation

The four_step_commutation unit provides a commutation using four step voltage modulation method.

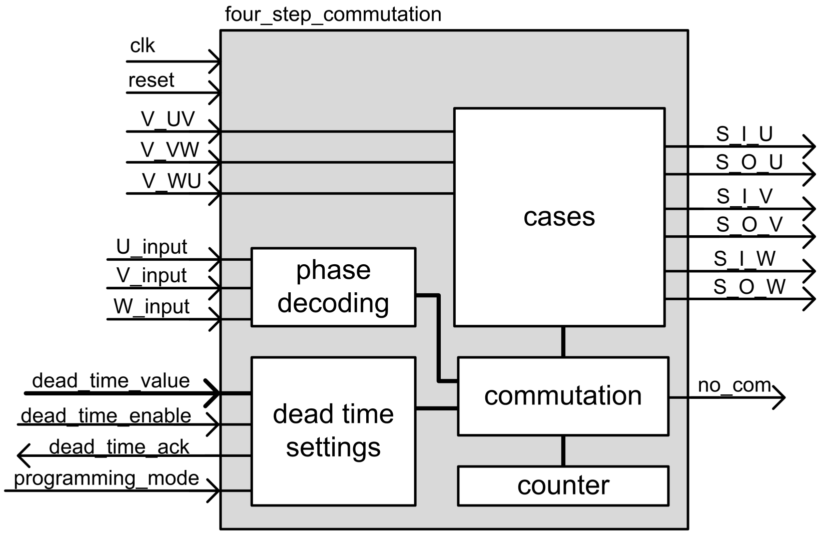


Figure 4.17: Four Step Voltage Commutation Unit four_step_commutation.

Signals input_U, input_V and input_W, signifying the reference output phase, are first decoded into one signal in phase_decoding process. The length of a commutation step determined by dead_time_value is asserted during programming mode by using enable-acknowledgement handshake. To achieve the correct step length a simple resetable counter is used.

In the process commutation, there is a generic four step commutation written using FSM. There are generic signals used in this process (they are adjusted to the versatile case of commutation - see commutation schema at 2.2.4.1). This commutation adjusting for the particular cases is processed in the cases process.

Signal no_com is asserted, when there is no commutation currently present in the unit.

4.3.2.8 two_step_commutation

The two_step_commutation unit provides commutation using two step current modulation method.

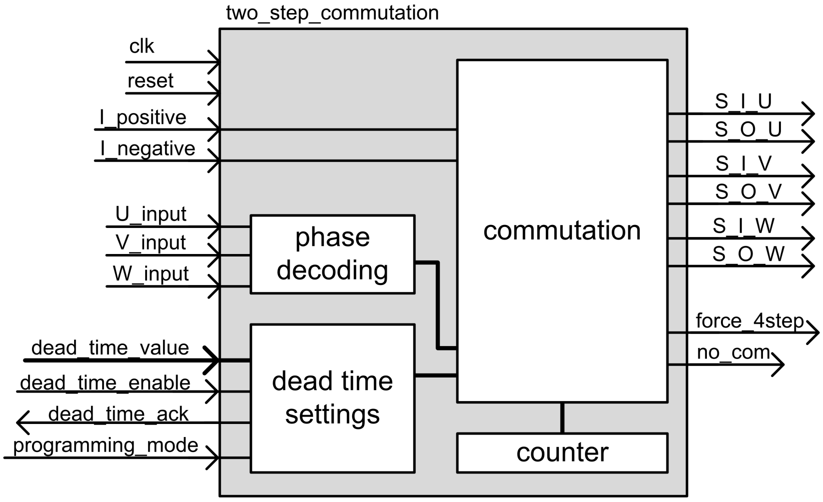


Figure 4.18: Two Step Current Commutation Unit two_step_commutation.

In this unit, the processes phase_decoding, dead_time_settings and counter are identical with those in four_step_commutation unit.

The process commutation is providing the whole commutation using FSM. The signal no_com is asserted, when there is no commutation currently present at the unit. If the current direction cannot be recognized, the signal force_4step, forcing the four step commutation, is asserted.

4.3.2.9 voltage_current_direction

The voltage_current_direction unit prepares the input phase-to-phase voltage polarity signals and voltage signals from the particular transistors (output current polarity detection) to be used in commutation.

The voltage polarity signals (V_UV_input, V_VW_input and V_WU_input) are sampled with a clock frequency. If a signal value did not change during the last three samples, then this value is being loaded into the voltage polarity register (V_UV, V_VW and V_WU).

The actual output phase information is obtained using the output gate signals (using or). Using these signals and the compared voltage signals from the particular transistors, the output current polarity is determined in a unit current_decoder. There are three identical current_decoder units present, each for the particular output phase.

4.3.2.10 current_decoder

The current_decoder unit determines the output current direction from the information about an actual connected input phase and the compared voltages on the particular

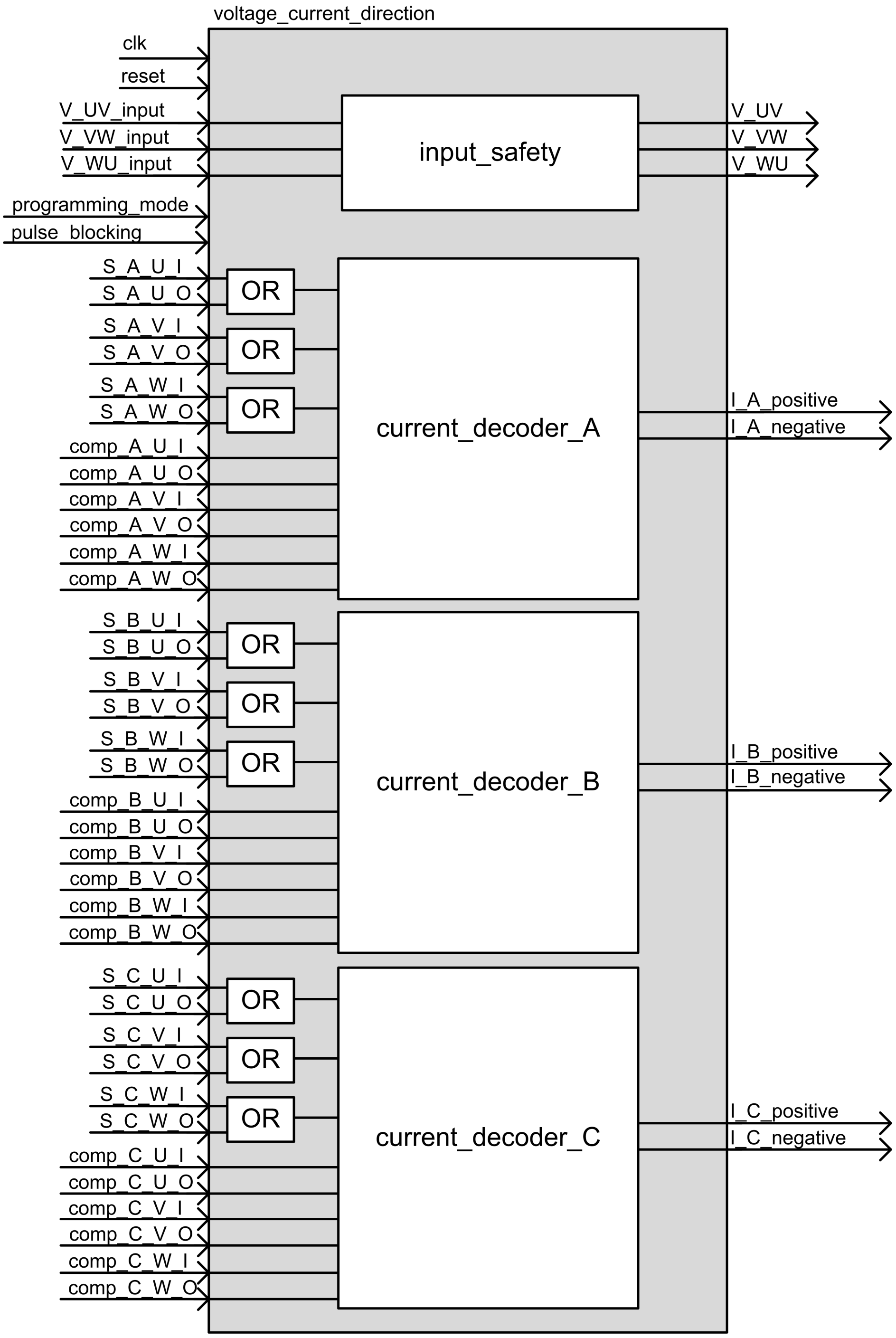


Figure 4.19: Polarity Detection Unit voltage_current_direction.

transistors.

In the process `input_safety`, the compared voltage signals are sampled. If a signal value does not change during three clock cycles, this value is loaded to internal `comp_*_signal` register. Using this register, the output current direction is determined.

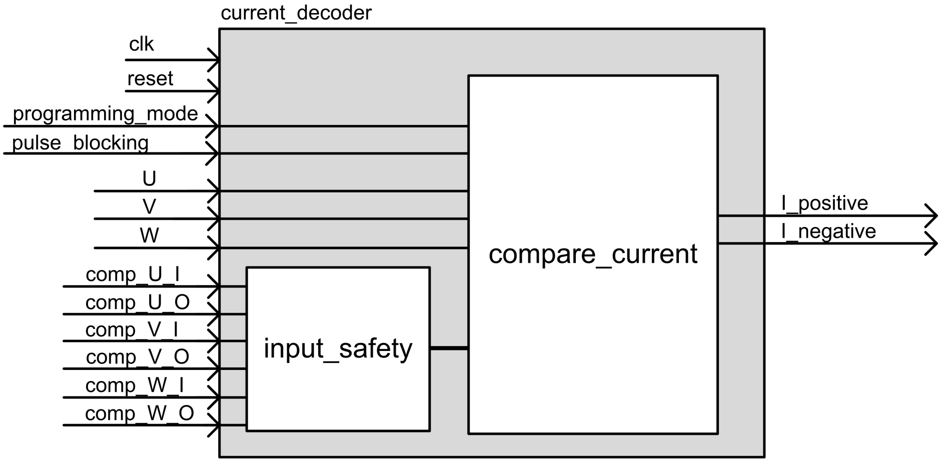


Figure 4.20: Output Current Polarity Detection Unit current_decoder.

In the process compare_unit, the output current polarity is determined using the compared voltage signals and actual connected input phase information according to the 2.2.4.4. This unit cannot determine current polarity during commutation. A detection error is not captured and the output current polarity is set as undecided.

4.3.2.11 error_handling

The error_handling unit captures the error signals and resets the error register after the confirmation command was called.

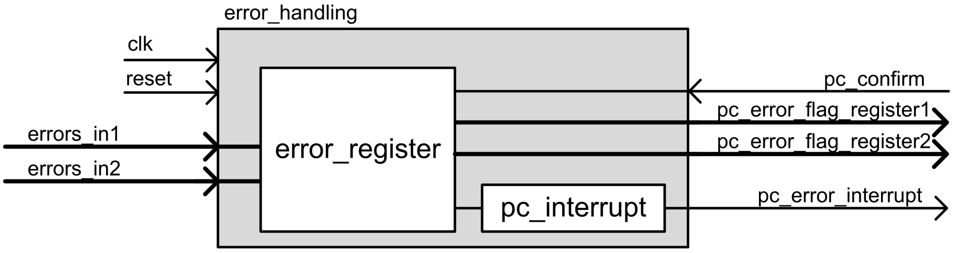


Figure 4.21: Error Capturing Unit error_handling.

In the process error_register, the driver error signals from the converter are sampled. If there is the same non-zero signal during three clock cycles, an internal error register is asserted. If a confirmation signal pc_confirm is asserted, the error registers are set to zero.

In the process pc_interrupt, a driver error signal for interrupt register is decided. If there is some error present at the error register, pc_error_interrupt signal is set to '1'.

4.3.3 Registers Description

In this section a use of registers will be described. In a case of practical use of modulator, this section has a great significance.

`Interrupt Register` 0x1

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	-	R	-	R	R	R	R	R	R	R	R	R	R	R
-	-	-	SM	-	DTE	OO	TE	SE	DE	PM	PB	FS	TL	DTL	SL

`Interrupt Register` basically provides all the necessary actual information about the converter.

SL SAW LOADED
The saw registers are already loaded to operate.

DTL DEAD TIME LOADED
The dead time registers are already loaded to operate.

TL TIMES LOADED
Time registers are already loaded to operate.

FS FOUR STEP ON
When 1, the converter is operating with four step voltage commutation.
When 0, two step current commutation is present.

PB PULSE BLOCKING
Pulse blocking mode is present (might also be a result of an error).

PM PROGRAMMING MODE
Programming mode is present. Pulses are blocked. Some settings apply only in programming mode.

DE DRIVER ERROR
Some driver error is present. In case of safe mode operation, the blocking mode is turned on.

SE SAW WRITING ERROR
This error occurs when it is tried to write into saw registers, before new values are loaded.

TE TIMES WRITING ERROR
This error occurs when it is tried to write into input time and sector registers, before new values are loaded.

OO OPTIMIZED ON
Optimized pattern is used.

DTE DEAD TIME WRITING ERROR
This error occurs when it is tried to write into dead time registers, before new values are loaded.

SM SAFE MODE ON
Safe mode turns on pulse blocking, if a driver error is present.

Command Register 0x0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	-	-	-	-	-	W	W	W	W
-	-	-	-	-	-	-	-	-	-	-	-	CR3	CR2	CR1	CR0

CR3-0	**Description**
0x1	BLOCK PULSES Turns pulse blocking on.
0x2	UNBLOCK PULSES Turns pulse blocking off.
0x3	PROGRAMMING MODE ON Turns programming mode on.
0x4	PROGRAMMING MODE OFF Turns programming mode off.
0x5	SAW DATA ENABLE Requests new saw values to be loaded (applies first in programming mode).
0x6	TIMES DATA ENABLE Requests new time and sector values to be loaded (applies first in new saw period).
0x7	OPTIMIZE ON Turns pattern optimizing mode on (only in programming mode).
0x8	OPTIMIZE OFF Turns pattern optimizing mode off (only in programming mode).
0x9	ERROR CONFIRMATION Confirms a driver error, in case that source of error was removed.
0xA	FOUR STEP COMMAND Turns on four step voltage commutation (applies immediately after last commutation).
0xB	TWO STEP COMMAND Turns on two step current commutation (applies immediately after last commutation).
0xD	SAFE MODE ON Turns on the safe mode. In a case of safe mode, a driver error automatically blocks the pulses.
0xE	SAFE MODE OFF Turns off the safe mode.

Dead Time Value Register 0x3

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	-	W/R	W/R	W/R	W/R	W/R	W/R	W/R	W/R
-	-	-	-	-	-	-	-	DT7	DT6	DT5	DT4	DT3	DT2	DT1	DT0

This register determines a dead time length for a commutation. The length of the dead time can be calculated using equation (4.3).

$$T_{DT} = \frac{n+1}{f_{clk}} \tag{4.3}$$

Where n stands for the dead time value and f_{clk} for the clock frequency.

Changes to this setting apply first after turning programming mode on.

`Saw Period Register` 0x4

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
W/R	W/R	W/R	W/R	W/R	W/R	W/R	W/R	W/R	W/R	W/R	W/R	W/R	W/R	W/R	W/R
R15	R14	R13	R12	R11	R10	R9	R8	R7	R6	R5	R4	R3	R2	R1	R0

This register contains the period of the periodic symmetric sawtooth signal.

It is loaded only in the programming mode. Note that an additional command (`SAW DATA ENABLE`) is needed to request the loading of new values. The `PWM VALUES LOADED` flag is significant first after the command has been sent.

The signal frequency calculation is explained in the next section.

`Saw Frequency Divider Register` 0x5

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	-	W/R	W/R	W/R	W/R	W/R	W/R	W/R	W/R
-	-	-	-	-	-	-	-	R7	R6	R5	R4	R3	R2	R1	R0

This register contains a value of frequency divider for sawtooth signal.

It is loaded only in the programming mode. Note that an additional command (`SAW DATA ENABLE`) is needed to request the loading of the new values. The `PWM VALUES LOADED` flag is significant first after the command has been send.

The sawtooth signal frequency calculation follows:

$$f_{saw} = \frac{f_{clk}}{2(p+1)(d+1)} \tag{4.4}$$

Where f_{clk} is the FPGA clock frequency, p is the period value and d is the frequency divider value.

`Error Register 1, Error Register 2 -` 0x6, 0x7

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	R	R	R	R	R	R	R	R	R
-	-	-	-	-	-	-	AU	BU	CU	AV	BV	CV	AW	BW	CW

These two registers provide a possibility to locate on which component an error was present. After removing the source of an error, it is needed to clear this register with the confirmation command.

The error information about the input-side transistors is in the `Error Register 1`. The error information about the output-side transistors is in the `Error Register 2`.

Registers `T_INPUT_1`, `T_INPUT_1_X_T_OUTPUT_1`, `T_INPUT_1_X_T_OUTPUT_2`, `T_INPUT_2_X_T_OUTPUT_1`, `T_INPUT_2_X_T_OUTPUT_2` - 0xA, 0xB, 0xC, 0xD, 0xE

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
W/R	W/R	W/R	W/R	W/R	W/R	W/R	W/R	W/R	W/R	W/R	W/R	W/R	W/R	W/R	W/R
R15	R14	R13	R12	R11	R10	R9	R8	R7	R6	R5	R4	R3	R2	R1	R0

These registers contain time values calculated from modulation duty-cycle period values d and the saw period register value p.

$$\texttt{T\_INPUT\_1} = \frac{d_{r1}}{d_{r1} + d_{r2}} \cdot p \tag{4.5}$$

$$\texttt{T\_INPUT\_1\_X\_T\_OUTPUT\_1} = d_{r1} \cdot d_{i1} \cdot p \tag{4.6}$$

$$\texttt{T\_INPUT\_1\_X\_T\_OUTPUT\_2} = d_{r1} \cdot d_{i2} \cdot p \tag{4.7}$$

$$\texttt{T\_INPUT\_2\_X\_T\_OUTPUT\_1} = d_{r2} \cdot d_{i1} \cdot p \tag{4.8}$$

$$\texttt{T\_INPUT\_2\_X\_T\_OUTPUT\_2} = d_{r2} \cdot d_{i2} \cdot p \tag{4.9}$$

These registers are being loaded always synchronous with beginning of saw signal period, if the enabling request command was used (`TIMES DATA ENABLE COMMAND`). Note, that this additional command is needed to request loading of new values. The `TIME VALUES LOADED` flag is significant first after the command has been send.

`Sector Register` 0xF

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	-	W/R	W/R	W/R	W/R	W/R	W/R	W/R	W/R
-	-	-	-	-	-	-	-	R3	R2	R1	R0	I3	I2	I1	I0

This register contains the information about the virtual rectifier modulation sector (INPUT SECTOR) and the virtual inverter modulation sector (OUTPUT SECTOR).

Bits **R3 - R0** include information about rectifier modulation sector (0x1 - 0x6).

Bits **I3 - I0** include information about inverter modulation sector (0x1 - 0x6).

Note that the sector information has to be enabled by the enabling command (`TIMES DATA ENABLE COMMAND`) together with times registers.

4.3.4 MC C Header

In order to use the MC converter in a control algorithm, a C Header was written.

There are prepared functions for writing and reading particular registers and writing particular commands. It also stores addresses of registers.

The header is included on the CD. It is also printed in appendix D.

4.3.5 MC Manual UI

In order to make settings to the MC modulator manually, a simple UI program was written. Using this program, it is also possible to read the current settings.

This program with source codes is included on the CD.

4.3.6 MC Demo Code

In order to verify function of the converter a demo code was written. The demo was written with the help of my supervisor using `Real Time Kernel` [9], which is currently being developed at the Department of Electric Drives and Traction.

The demo includes simple PLL for the converter input and generates the output voltage at the reference frequency and output modulation degree. It also provides a voltage polarity information, which is essential for the converter commutation.

This program with source codes is included on the CD.

Chapter 5

Experimental Results

In this chapter, experimental results measured on Voltage Source Inverter and Matrix Converter, using the described modulators, will be presented.

5.1 Voltage Source Inverter

In order to verify the VSI modulator functionality and to obtain the properties of New Modulation Method, there were few measurements performed.

5.1.1 Measurement Stand Description

A measurement stand was built according to the fig. 5.1. There was one analogue ammeter `A` and voltmeter `V` used, in order, to obtain assured information about the converter

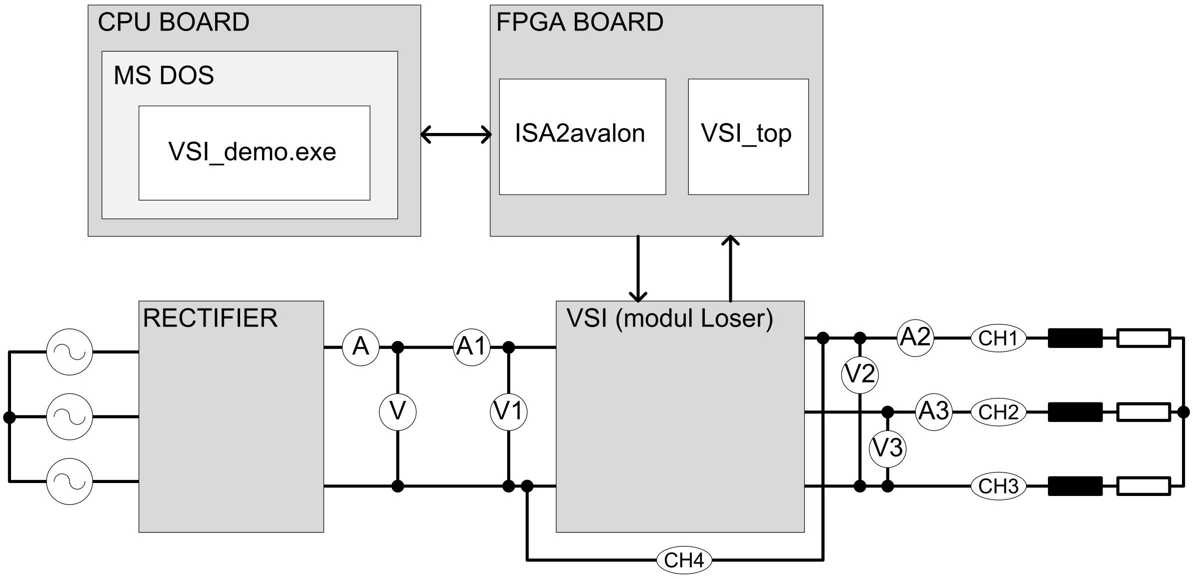


Figure 5.1: VSI Measurement Stand Scheme.

state. Using Tektronix THS 3014 scope, time function curves of the output currents (current probes CH1, CH2 and CH3) and transistor IGBT U down voltage (voltage probe CHR4) were captured. The voltage on the transistor was measured to capture the switching function curve form.

In order to measure an efficiency and THD of the converter, a power analyzer Tektronix PA 4000 was used. Input power was measured on the clamps A1 and V1. The output power was measured on the clamps A2, V2, A3 and V3 using standard Blondel measurement technique. The settings of the power analyzer were set using PWRVIEW software wizard choosing "PWM Drive Efficiency Measurement" option. In order to calculate the THD, the first 100 harmonics were used.

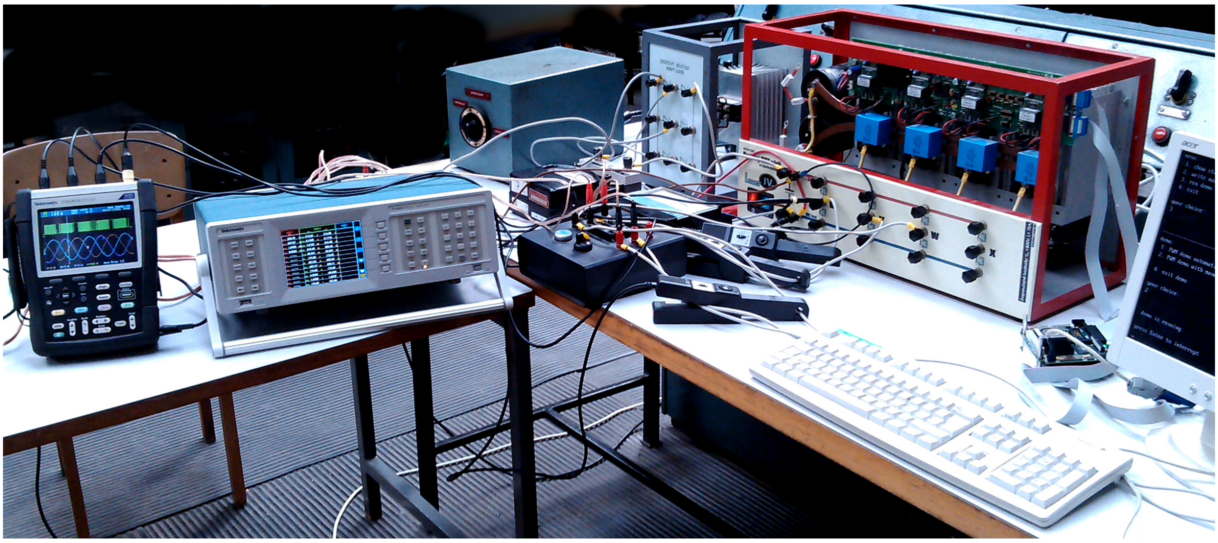


Figure 5.2: VSI Measurement Stand Photo.

As the input DC power supply, a three-phase voltage supply with a diode rectifier were used. As the load a variable three-phase resistor and three-phase inductor, with a variable air gap, were used.

5.1.2 Measured Curves

To verify and show the basic functionality of the modulator and different modulation strategies, there were time-function curves of the output currents (i_U, i_V and i_W) and the voltage on the IGBT U down component ($v_{S,U,down}$) captured.

Different modulation methods were tested, by the same input parameters. The modulation degree was chosen to 0.9. The input DC voltage was chosen 391 V and input power was 3.38 kW.

From the following figures, we can state that all of the implemented methods provide sine-form output currents. On the voltage on the IGBT component, we can see, that there is not any switching present on the IGBT during one third of period, when using New Modulation Method.

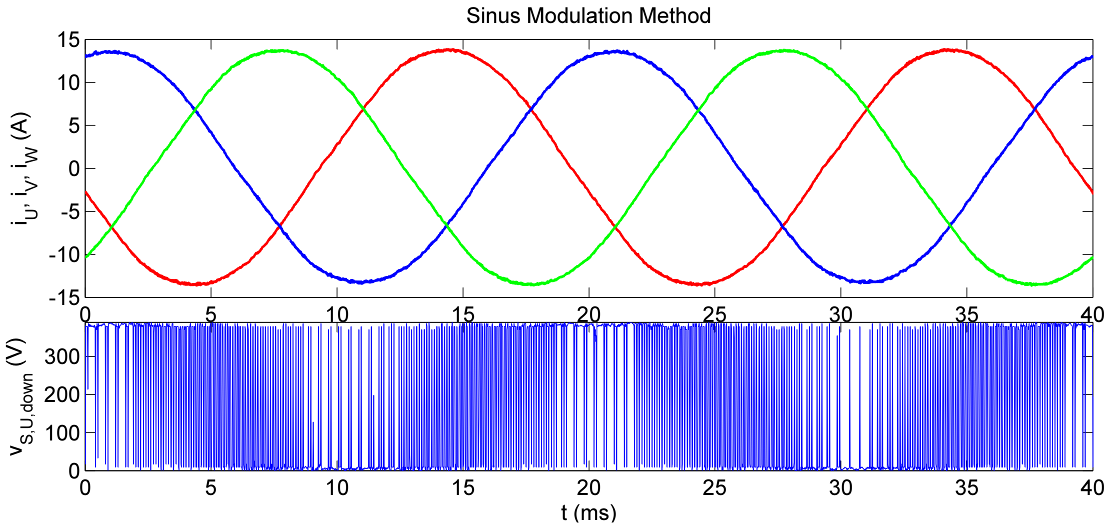


Figure 5.3: Measured Sinus Modulation Time-Function Curves.

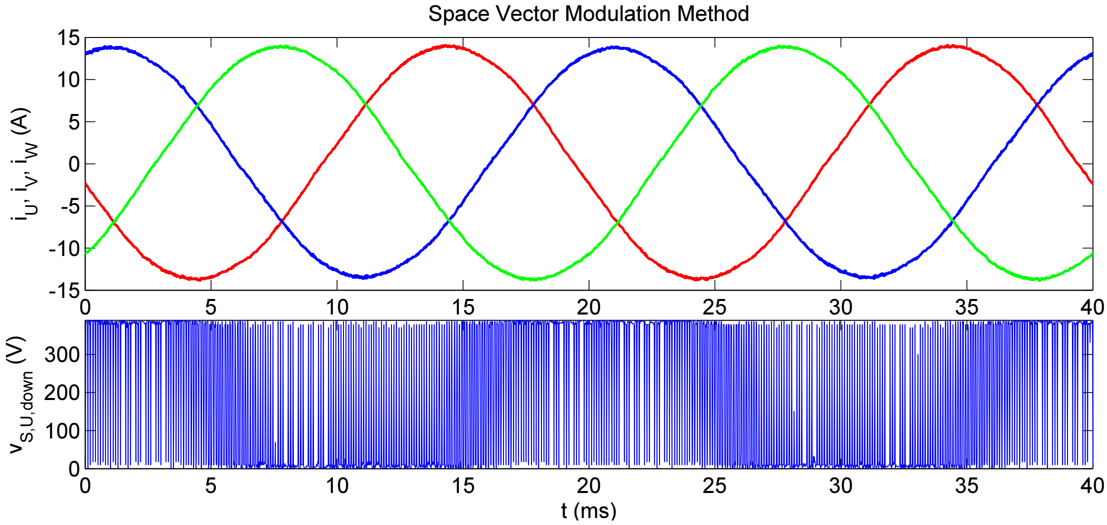


Figure 5.4: Measured Space Vector Modulation Time-Function Curves.

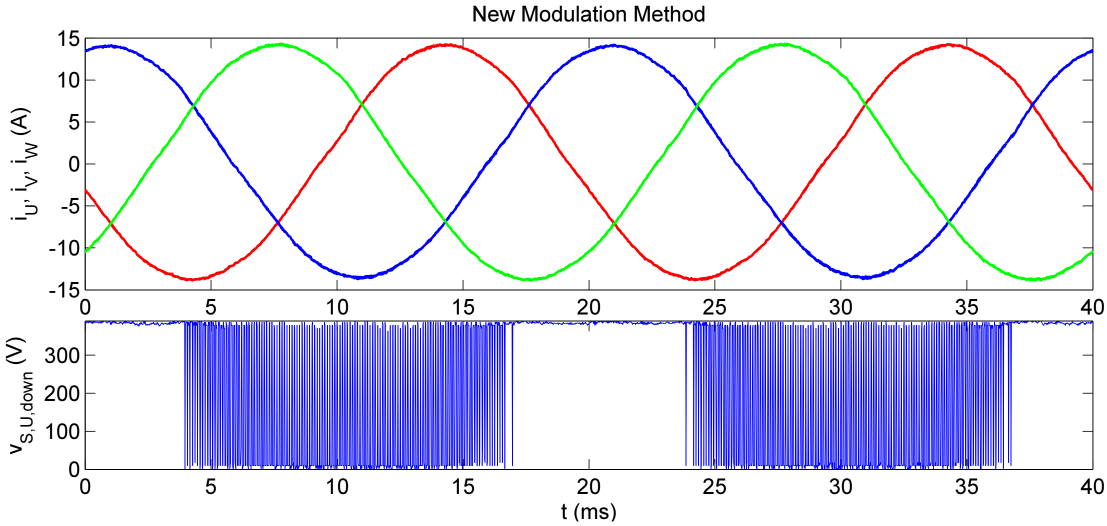


Figure 5.5: Measured New Modulation Time-Function Curves.

5.1.3 Losses Measurements

To compare the different modulation methods, the efficiency was measured for different power factor values. In fig. 5.6, we can see that the shape of measured efficiency curves is similar to the ones simulated. The overall efficiency of the converter Loser is higher then simulated (in the simulation the values from datasheet were used, which are commonly the worst-case values).

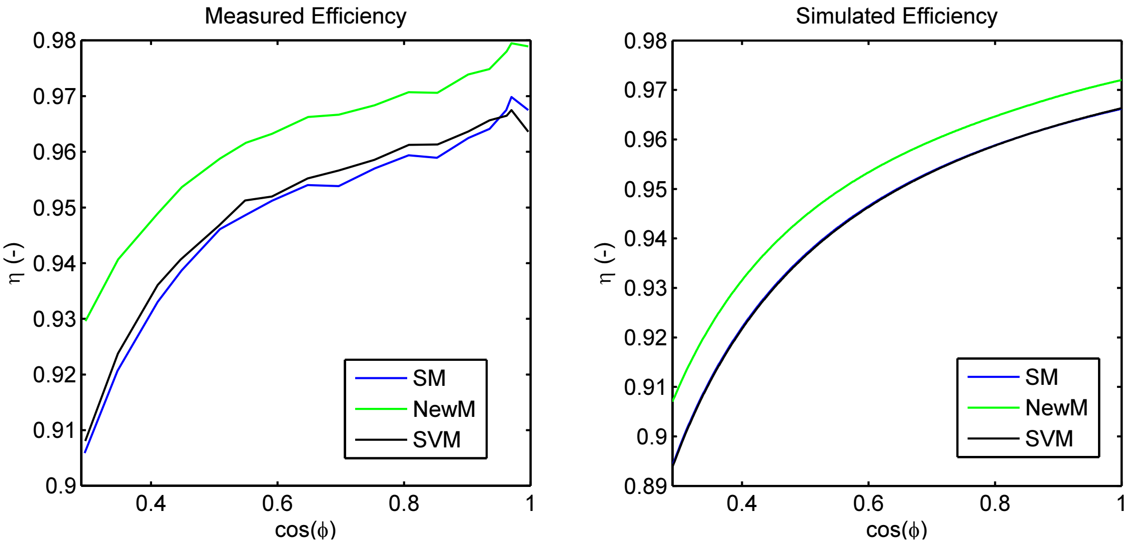


Figure 5.6: Measured VSI Efficiency for Different Modulation Methods as a Function of Power Factor.

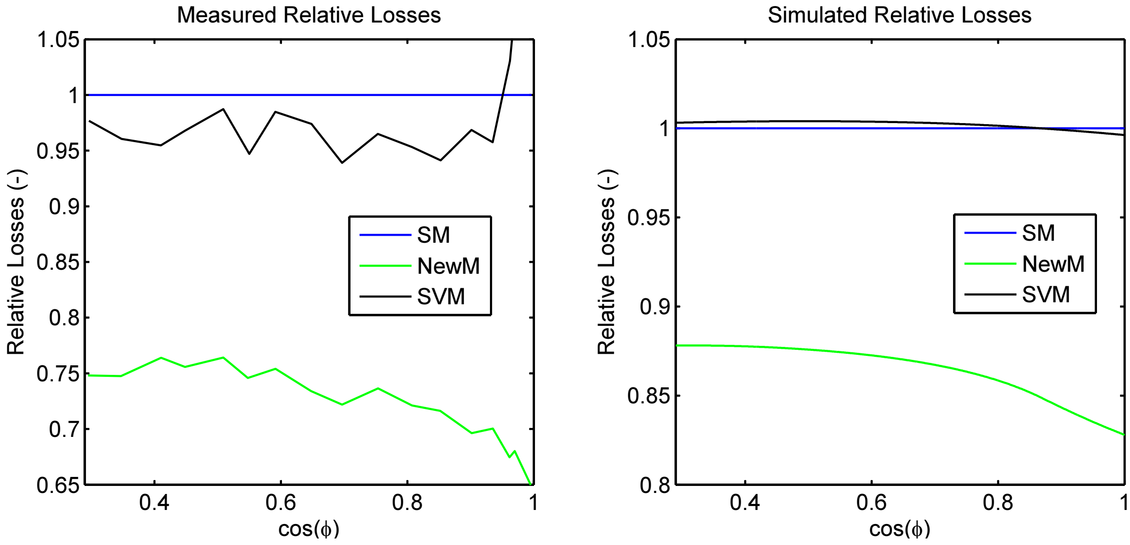


Figure 5.7: Measured VSI Relative Losses for Different Modulation Methods as a Function of Power Factor.

In the fig. 5.7, there are converter relative losses as a ratio towards Sinus Modulation losses. The losses while using Space Vector Modulation Method lie in a 7% interval from Sinus Modulation losses, therefore they do not differ significantly. The losses, while using New Modulation Method, are between 25 and 35 % lower. The gain of this method measured is higher than simulated. This means that by the converter, the switching losses are more significant then the others.

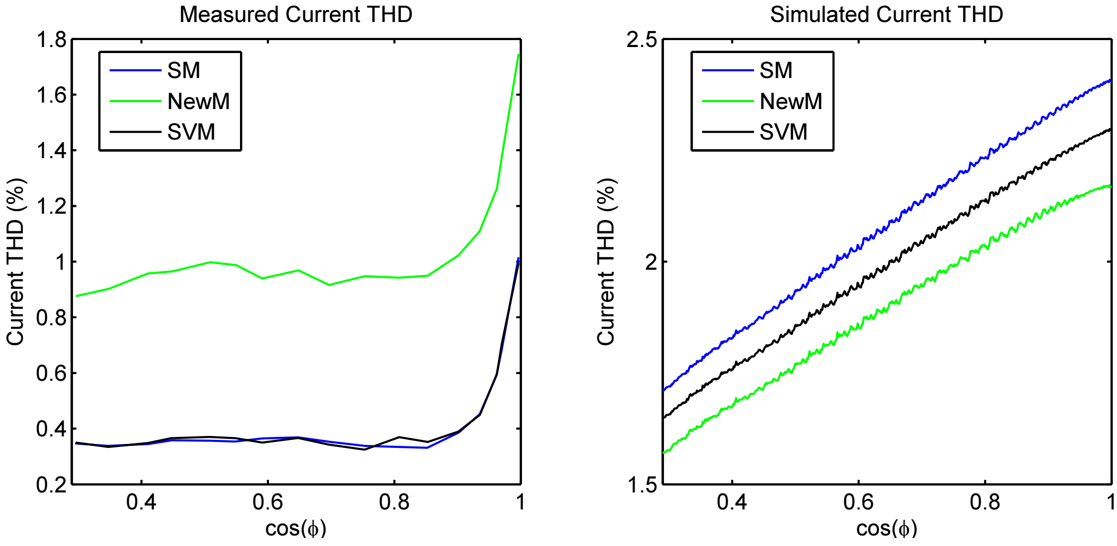


Figure 5.8: Measured VSI current THD for Different Modulation Methods as a Function of Power Factor.

The output current THD factor curves are in fig. 5.8. The THD of the New Modulation Method is negatively influenced by a dead time generation. This causes a distortion of thin pulses in the parts, when the current changes significantly.

5.2 Matrix Converter

In order to verify the MC modulator functionality, there were few measurements performed.

5.2.1 Measurement Stand Description

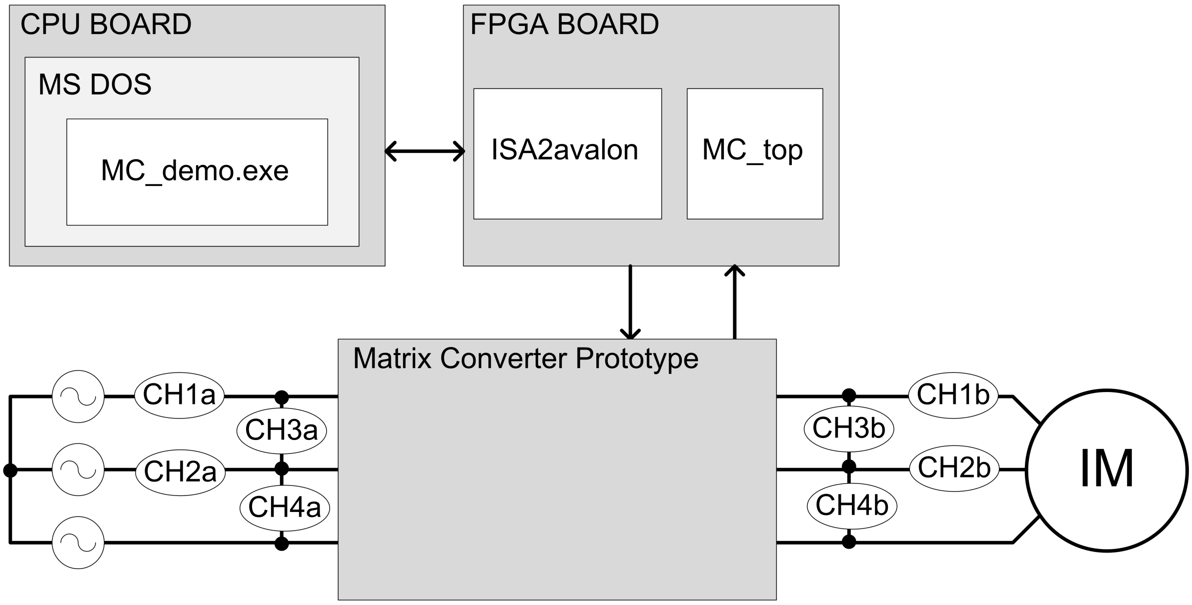


Figure 5.9: MC Measurement Stand Scheme.

A measurement stand was built according to the fig. 5.9. Using `Tektronix THS 3014` scope, time-function curves of the converter input currents (current probes `CH1a` and `CH2a`) and output currents (current probes `CH1b` and `CH2b`) and the converter input line-to-line voltages (voltage probes `CH3a` and `CH4a`) and output voltages ((voltage probes `CH3b` and `CH4b`) were captured. The missing voltage and current curves, which were not measured, were calculated assuming that a sum of all three line-to-line voltages or a sum of all three phase currents is equal zero.

There was a three-phase voltage source used for the converter input. As the load, there was an induction machine used.

5.2.2 Measured Curves

In order to prove the functionality of the converter the input currents (i_U, i_V and i_W) and voltages (u_U u_V and u_W) and the output currents (i_A, i_B and i_C) and voltages (u_A, u_B and u_C) were captured, while different settings were chosen.

The measured curves are plotted in fig. 5.10, fig. 5.11 and fig. 5.12. There are three different output frequencies chosen (20, 35 and 45 Hz). The virtual rectifier degree is held at $M_i = 1$ and virtual inverter modulation degree M_o differs from 0.95 to 1.15. The converter does not generate any reactive power at the input to achieve the maximal possible voltage transfer ratio (this is a common practice for Matrix Converter).

From the mentioned figures, it is possible to state that a chosen commutation method or a chosen switching pattern do not influence the measured curve forms. Nevertheless, the switching pattern should influence the star voltage v_*, which was not measured, because there was no common voltage clamp available for the voltage source.

Because it was not possible to measure the currents behind the converter input filter, there is always a compensating current present. This current contains more current 50-Hertz harmonics and compensates the harmonics of the voltage source and is around 3 A RMS. This compensating current is the reason of non-sine-form curves of the input current.

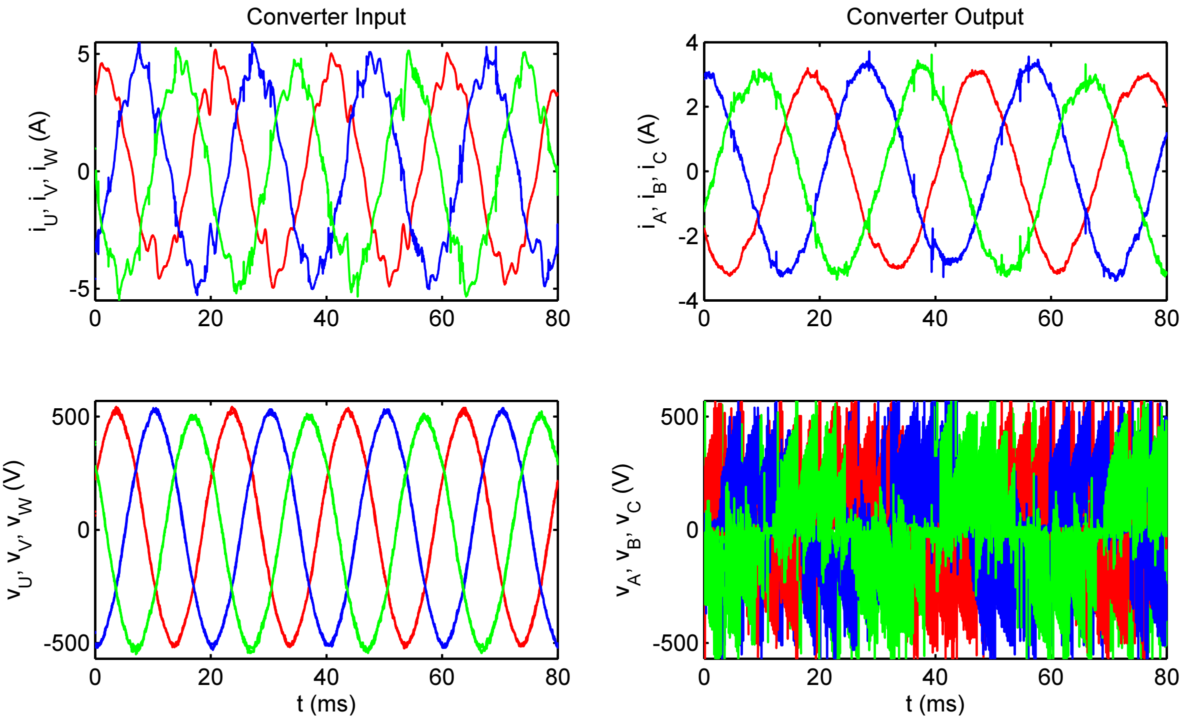


Figure 5.10: Measured Voltages and Currents on Matrix Converter Prototype, $f_{out} = 35$ Hz, $M_o = 1.15$, optimized pattern and four step voltage commutation are present.

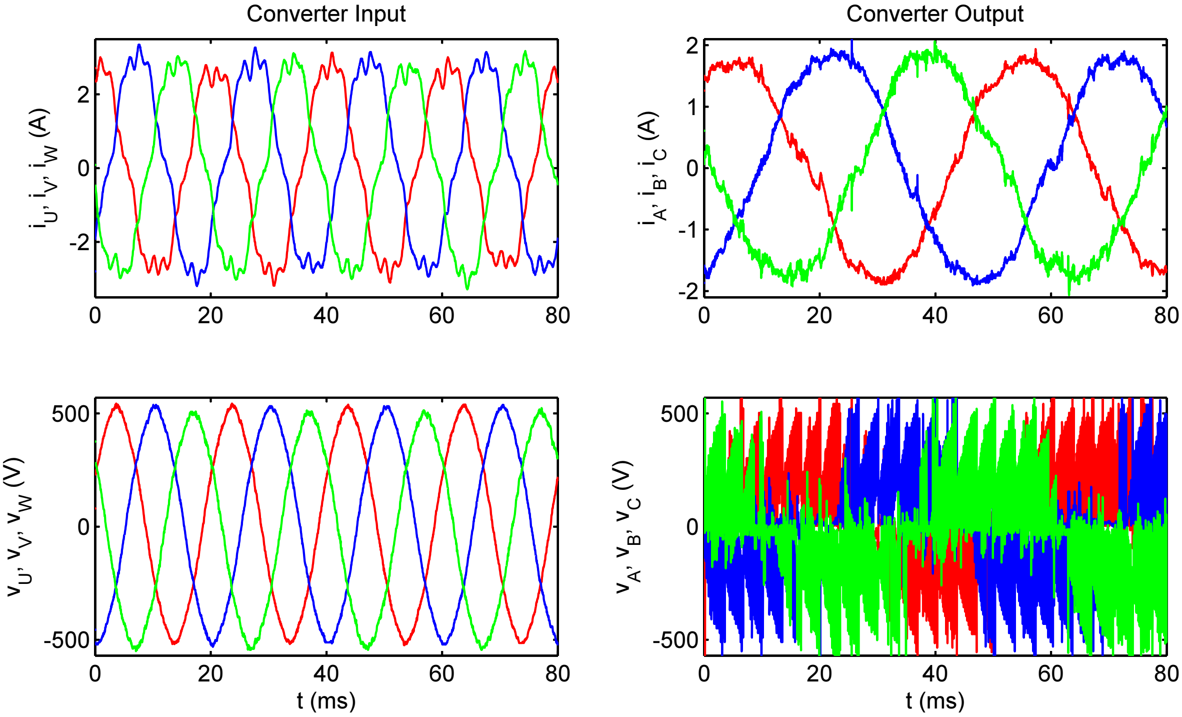


Figure 5.11: Measured Voltages and Currents on Matrix Converter Prototype, $f_{out} = 20$ Hz, $M_o = 1$, optimized pattern and two step current commutation are present.

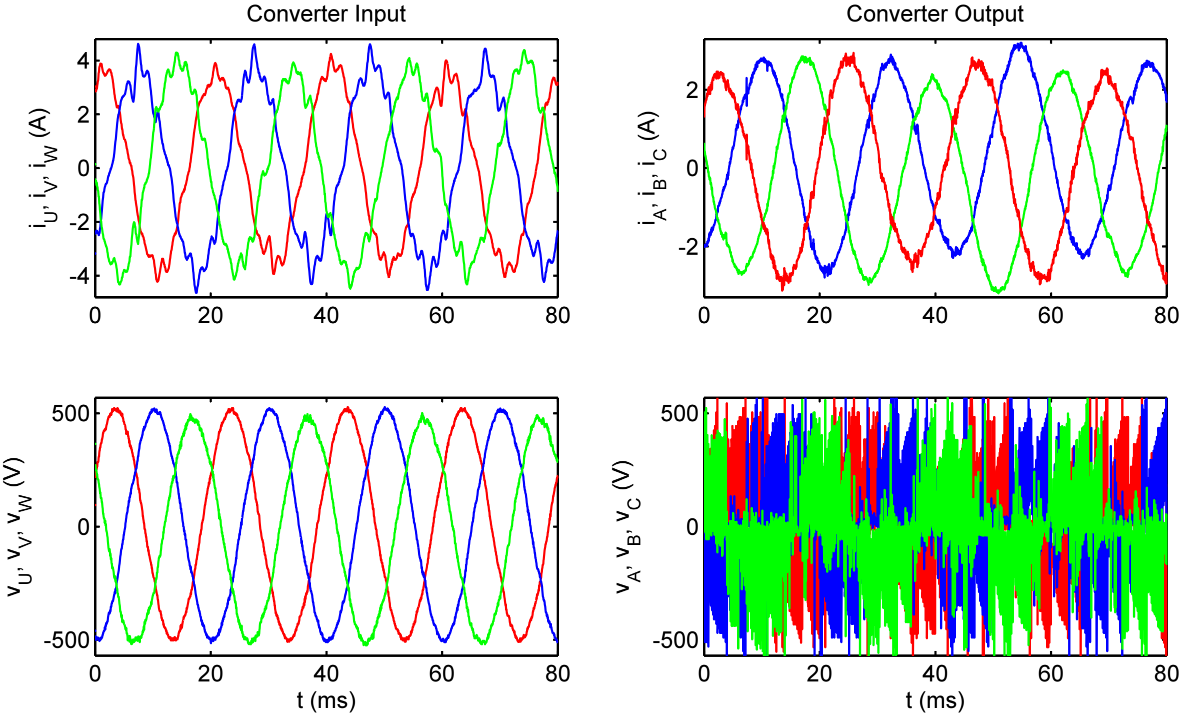


Figure 5.12: Measured Voltages and Currents on Matrix Converter Prototype, $f_{out} = 45$ Hz, $M_o = 0.9$, unoptimized pattern and two step voltage commutation are present.

Chapter 6

Conclusion

In this work, modulation methods for two known converter topologies were presented. For the common Voltage Source Inverter, we dealt with Amplitude Modulation method, Sinus Modulation method, Modulation with an Injection of Third Harmonic and Space Vector Modulation method. One new method with a DC common voltage injection was also proposed. For the Matrix Converter, Modulation Duty-Cycle Matrix Strategies, Direct Space Vector Modulation method and Indirect Space Vector Modulation method were presented. The focus was mainly put on the Indirect Space Vector Modulation method.

The main benefit of New Modulation method is a lowering of switching losses. This was analytically derived, simulated and later measured. All the methods deliver similar results, showing the significant savings on the power loss (by measurements it was shown, that using this method, between 25 and 35 percent of overall losses can be saved). The current THD was not significantly influenced while using this method.

In order to show the functionality of particular modulation methods, modulators were described in VHDL for the both of the mentioned converters. The Voltage Source Inverter modulator was tested with a converter module Loser using Sinus Modulation, Space Vector Modulation and New Method. The Matrix Converter modulator was tested with a converter prototype at the Department of Electric Drives and Traction using Indirect Space Vector Modulation Method (with two possible switching patterns and four step voltage or two step current commutation method).

The functionality was satisfactory proven for all the modulation methods.

The VHDL modulators were written versatile to be used with CPU board for any real time application with practically any converter of the mentioned topologies e.g. direct torque control.

Bibliography

[1] *Prof. Dr.-Ing. Mertens, A.:* **Leistungselektronik II.**
Vorlesungsskript, Leibniz Universität Hannover

[2] *Pavelka, J.; Čeřovský, Z.; Lettl, J.:* **Výkonová elektronika.**
Nakladatelství ČVUT, Prague 2007

[3] *Hájek, J.:* **Elektrický pohon s řídícím a měřícím systémem CompactRIO**
Master Thesis, Prague 2012

[4] *Kučka, J.:* **Návrh DC/DC měniče pro nabíjení superkondenzátoru**
Bachelor Thesis, Prague 2012

[5] *Jussila, M.:* **Comparison of Space-Vector-Modulated Direct and Indirect Matrix Converters in Low-Power Applications**
Thesis for the degree of Doctor of Technology, Tampere 2007

[6] *Kuzmanović, D.:* **Field Oriented Control and Direct Torque Control of Matrix Converter-Fed Induction Motor**
PhD Thesis, Prague 2009

[7] *Pošta, P.:* **Stavový automat modulátoru pro kompaktní maticový měnič**
Master Thesis, Prague 2007

[8] *Bauer, J.:* **Konstrukce kompaktního maticového měniče a jeho dimenzování**
Master Thesis, Prague 2007

[9] *Flígl, S.:* **RT PC-based control system**
Introduction & user manual, Ruhr-Universität Bochum 2003

[10] *POWEREX CM100DU-24NFH Datasheet*
`http://www.pwrx.com/pwrx/docs/cm100du-24nfh.pdf`
[November 25, 2013]

[11] *Description and Application Manual for SCALE Drivers*
`http://www.symmetron.ru/suppliers/concept/scale_e.pdf`
[March 3, 2014]

[12] *RTD FPGA 6800 User's Manual*
`http://www.rtd.com/NEW_manuals/hardware/utilitymodules/FPGA6800_BDM610020061B.pdf`
[November 9, 2013]

[13] *CMA157886 cpuModules User's Manual*
`http://www.rtd.com/NEW_manuals/hardware/cpumodules/CMA157886_BDM610000063F.pdf`
[February 16, 2014]

[14] *ISA Bus Technical Summary*
`http://http://www.techfest.com/hardware/bus/isa.htm`
[March 3, 2014]

Appendix A

CD

```
CD:
 - kucka_master_thesis.pdf     (this thesis)

 + VSI
   + vsi_header                (VSI headers)
     - vsi_def.h
     - vsi_def.c
   + vsi_demo
     - vsi_demo.exe            (VSI demo program)
     - ISA_qual.exe            (modulator with PC communication quality test)
     - ISA_time.exe            (determines length of ISA IO writing and reading
                                cycle period)
   + vsi_modudulator           (modulator in VHDL)

 + MC
   + mc_header                 (MC headers)
     - mc_def.h
     - mc_def.c
   + mc_manual_UI
     - mc_man.exe              (MC manual UI program)
   + mc_demo
     - mc_demo.exe             (MC demo program)
   + mc_modudulator            (modulator in VHDL)
```

Appendix B

Switching Patterns Lookup Table

SECTOR		ISVM VECTORS				NON-OPTIMIZED		OPTIMIZED	
		$\vec{i}_{r1}$	$\vec{i}_{r1}$	$\vec{i}_{r2}$	$\vec{i}_{r2}$	$\vec{i}_{r1}$	$\vec{i}_{r2}$	$\vec{i}_{r1}$	$\vec{i}_{r2}$
INPUT	OUTPUT	$\vec{v}_{i1}$	$\vec{v}_{i2}$	$\vec{v}_{i1}$	$\vec{v}_{i2}$	$\vec{v}_{0}$	$\vec{v}_{0}$	$\vec{v}_{0}$	$\vec{v}_{0}$
1	1	AB_100	AB_110	AC_100	AC_110	AB_111	AC_111	AB_111	AC_111
	2	AB_110	AB_010	AC_110	AC_010	AB_000	AC_000	AB_111	AC_111
	3	AB_010	AB_011	AC_010	AC_011	AB_111	AC_111	AB_111	AC_111
	4	AB_011	AB_001	AC_011	AC_001	AB_000	AC_000	AB_111	AC_111
	5	AB_001	AB_101	AC_001	AC_101	AB_111	AC_111	AB_111	AC_111
	6	AB_101	AB_100	AC_101	AC_100	AB_000	AC_000	AB_111	AC_111
2	1	AC_100	AC_110	BC_100	BC_110	AC_111	BC_111	AC_000	BC_000
	2	AC_110	AC_010	BC_110	BC_010	AC_000	BC_000	AC_000	BC_000
	3	AC_010	AC_011	BC_010	BC_011	AC_111	BC_111	AC_000	BC_000
	4	AC_011	AC_001	BC_011	BC_001	AC_000	BC_000	AC_000	BC_000
	5	AC_001	AC_101	BC_001	BC_101	AC_111	BC_111	AC_000	BC_000
	6	AC_101	AC_100	BC_101	BC_100	AC_000	BC_000	AC_000	BC_000
3	1	BC_100	BC_110	BA_100	BA_110	BC_111	BA_111	BC_111	BA_111
	2	BC_110	BC_010	BA_110	BA_010	BC_000	BA_000	BC_111	BA_111
	3	BC_010	BC_011	BA_010	BA_011	BC_111	BA_111	BC_111	BA_111
	4	BC_011	BC_001	BA_011	BA_001	BC_000	BA_000	BC_111	BA_111
	5	BC_001	BC_101	BA_001	BA_101	BC_111	BA_111	BC_111	BA_111
	6	BC_101	BC_100	BA_101	BA_100	BC_000	BA_000	BC_111	BA_111
4	1	BA_100	BA_110	CA_100	CA_110	BA_111	CA_111	BA_000	CA_000
	2	BA_110	BA_010	CA_110	CA_010	BA_000	CA_000	BA_000	CA_000
	3	BA_010	BA_011	CA_010	CA_011	BA_111	CA_111	BA_000	CA_000
	4	BA_011	BA_001	CA_011	CA_001	BA_000	CA_000	BA_000	CA_000
	5	BA_001	BA_101	CA_001	CA_101	BA_111	CA_111	BA_000	CA_000
	6	BA_101	BA_100	CA_101	CA_100	BA_000	CA_000	BA_000	CA_000
5	1	CA_100	CA_110	CB_100	CB_110	CA_111	CB_111	CA_111	CB_111
	2	CA_110	CA_010	CB_110	CB_010	CA_000	CB_000	CA_111	CB_111
	3	CA_010	CA_011	CB_010	CB_011	CA_111	CB_111	CA_111	CB_111
	4	CA_011	CA_001	CB_011	CB_001	CA_000	CB_000	CA_111	CB_111
	5	CA_001	CA_101	CB_001	CB_101	CA_111	CB_111	CA_111	CB_111
	6	CA_101	CA_100	CB_101	CB_100	CA_000	CB_000	CA_111	CB_111

SECTOR		ISVM VECTORS				NON-OPTIMIZED		OPTIMIZED	
INPUT	OUTPUT	$\vec{i}_{r1}$ $\vec{v}_{i1}$	$\vec{i}_{r1}$ $\vec{v}_{i2}$	$\vec{i}_{r2}$ $\vec{v}_{i1}$	$\vec{i}_{r2}$ $\vec{v}_{i2}$	$\vec{i}_{r1}$ $\vec{v}_{0}$	$\vec{i}_{r2}$ $\vec{v}_{0}$	$\vec{i}_{r1}$ $\vec{v}_{0}$	$\vec{i}_{r2}$ $\vec{v}_{0}$
6	1	CB_100	CB_110	AB_100	AB_110	CB_111	AB_111	CB_000	AB_000
	2	CB_110	CB_010	AB_110	AB_010	CB_000	AB_000	CB_000	AB_000
	3	CB_010	CB_011	AB_010	AB_011	CB_111	AB_111	CB_000	AB_000
	4	CB_011	CB_001	AB_011	AB_001	CB_000	AB_000	CB_000	AB_000
	5	CB_001	CB_101	AB_001	AB_101	CB_111	AB_111	CB_000	AB_000
	6	CB_101	CB_100	AB_101	AB_100	CB_000	AB_000	CB_000	AB_000

Appendix C

VSI C Header Printout

C.1 vsi_def.h

```
/*-------------------------------------------------------------------
Project:           VSI modulator

by                Bc. Jakub Kucka
              (kuckajak@fel.cvut.cz)

component:  vsi_def
description: here, all needed definitions and functions are to be found
             to use VSI modulator this header has to be included
 use of particular registers is described in Thesis

This was created as a part of Master Thesis.
---------------------------------------------------------------------*/

#ifndef __vsi_def__
#define __vsi_def__
//-------REGISTER_ADDRESSES------------------------------------
#define PWM_U_VALUE                    0xAu
#define PWM_V_VALUE                    0xBu
#define PWM_W_VALUE                    0xCu
#define DEAD_TIME_VALUE                0x3u
#define SAW_PERIOD                     0x4u
#define SAW_FREQUENCY_DIVIDER          0x5u
#define MODULATION_METHOD              0x2u
#define ERROR_REGISTER                 0x6u
#define COMMAND_REGISTER               0x0u
#define INTERRUPT_REGISTER             0x1u

//-------COMMANDS-----------------------------------------------
#define BLOCK_PULSES_COMMAND           0x0001u
#define UNBLOCK_PULSES_COMMAND         0x0002u
#define PROGRAMMING_MODE_ON_COMMAND    0x0003u
```

```
#define PROGRAMMING_MODE_OFF_COMMAND   0x0004u
#define SAW_DATA_ENABLE_COMMAND        0x0005u
#define PWM_DATA_ENABLE_COMMAND        0x0006u
#define SAFE_MODE_ON_COMMAND           0x0007u
#define SAFE_MODE_OFF_COMMAND          0x0008u
#define ERROR_CONFIRM_COMMAND          0x0009u
#define DEAD_TIME_ON_COMMAND           0x000Au
#define DEAD_TIME_OFF_COMMAND          0x000Bu

//-------INTERRUPT POSITIONS------------------------------------
#define SAW_LOADED_INTERRUPT_POSITION                 0
#define DEAD_TIME_LOADED_INTERRUPT_POSITION           1
#define PWM_VALUES_LOADED_INTERRUPT_POSITION          2
#define MODULATION_LOADED_INTERRUPT_POSITION          3
#define PULSE_BLOCKING_INTERRUPT_POSITION             4
#define PROGRAMMING_MODE_INTERRUPT_POSITION           5
#define DRIVER_ERROR_INTERRUPT_POSITION               6
#define SAW_WRITING_ERROR_INTERRUPT_POSITION          7
#define PWM_WRITING_ERROR_INTERRUPT_POSITION          8
#define MODULATION_WRITING_ERROR_INTERRUPT_POSITION   9
#define DEAD_TIME_WRITING_ERROR_INTERRUPT_POSITION    10
#define SAFE_MODE_INTERRUPT_POSITION                  11
#define DEAD_TIME_ON_INTERRUPT_POSITION               12

//-------BEGGINING OF MEMORY SPACE FOR ISA----------------------
#define MEM_ADR                       0xf00000u
//-------BEGINNING OF ISA IO ADR--------------------------------
#define IO_ADR                        0x300u
//-------CHIP SELECT OFFSET-------------------------------------
#define chip_sel_offset               0x0000u

void write_vsi(unsigned short int address, unsigned short int data);
/*
 This function writes data to modulator register.
 address - register address - for example: PWM_U_VALUE
 data    - register input data
 It is also possible to write command using this function:
 write_vsi(COMMAND_REGISTER, BLOCK_PULSES_COMMAND);
*/
unsigned short int read_vsi(unsigned short int address);
/*
 This function reads from modulator register.
 address - register address - for example: PWM_U_VALUE
*/
unsigned short int get_bit(unsigned short int value, unsigned short int position);
/*
 This function returns bit value from "value" at position "position".
 It can be used to detect concrete interrupt flag.
 get_bit(read_vsi(INTERRUPT_REGISTER),DEAD_TIME_ON_INTERRUPT_POSITION);
*/
#endif
```

C.2 vsi_def.c

```
/*---------------------------------------------------------------------
Project:           VSI modulator

by                Bc. Jakub Kucka
              (kuckajak@fel.cvut.cz)

component:  vsi_def
description: here, all needed definitions and functions are to be found
             to use VSI modulator this header has to be included
 use of particular registers is described in Thesis

This was created as a part of Master Thesis.
----------------------------------------------------------------------*/

#include "vsi_def.h"
#include <conio.h>
#include <stdint.h>

void write_vsi(unsigned short int address, unsigned short int data)
{
  uint16_t i;

  i = IO_ADR+(2*(uint16_t)address); //address*2 - because of writing 2 bytes
  outpw(i,data);
}

uint16_t read_vsi(unsigned short int address)
{
  uint16_t i;
  i = IO_ADR+(2*(uint16_t)address);  //address*2 - because of writing 2 bytes
  return inpw(i);
}
unsigned short int get_bit(unsigned short int value, unsigned short int position)
{
  return ((value >> position)  & 0x0001);
}
```

Appendix D

MC C Header Printout

D.1 mc_def.h

```
/*-------------------------------------------------------------------
Project:           MC modulator

by                Bc. Jakub Kucka
              (kuckajak@fel.cvut.cz)

component:   mc_def
description: here, all needed definitions and functions are to be found
             to use MC modulator this header has to be included
 use of particular registers is described in Thesis

This was created as a part of Master Thesis.
---------------------------------------------------------------------*/

#ifndef __mc_def__
#define __mc_def__
//-------REGISTER_ADDRESSES------------------------------------
#define T_INPUT_1                        0xAu
#define T_INPUT_1_X_T_OUTPUT_1           0xBu
#define T_INPUT_1_X_T_OUTPUT_2           0xCu
#define T_INPUT_2_X_T_OUTPUT_1           0xDu
#define T_INPUT_2_X_T_OUTPUT_2           0xEu
#define SECTOR                           0xFu
#define DEAD_TIME_VALUE                  0x3u
#define SAW_PERIOD                       0x4u
#define SAW_FREQUENCY_DIVIDER            0x5u
#define ERROR_REGISTER1                  0x6u
#define ERROR_REGISTER2                  0x7u
#define COMMAND_REGISTER                 0x0u
#define INTERRUPT_REGISTER               0x1u

//-------COMMANDS-----------------------------------------------
```

```
#define BLOCK_PULSES_COMMAND                     0x0001u
#define UNBLOCK_PULSES_COMMAND                   0x0002u
#define PROGRAMMING_MODE_ON_COMMAND              0x0003u
#define PROGRAMMING_MODE_OFF_COMMAND             0x0004u
#define SAW_DATA_ENABLE_COMMAND                  0x0005u
#define TIMES_DATA_ENABLE_COMMAND                0x0006u
#define OPTIMIZED_ON_COMMAND                     0x0007u
#define OPTIMIZED_OFF_COMMAND                    0x0008u
#define ERROR_CONFIRM_COMMAND                    0x0009u
#define FOUR_STEP_COMMAND                        0x000Au
#define TWO_STEP_COMMAND                         0x000Bu
#define SAFE_MODE_ON_COMMAND                     0x000Du
#define SAFE_MODE_OFF_COMMAND                    0x000Eu

//-------INTERRUPT POSITIONS------------------------------------
#define SAW_LOADED_INTERRUPT_POSITION                 0
#define DEAD_TIME_LOADED_INTERRUPT_POSITION           1
#define TIMES_LOADED_INTERRUPT_POSITION               2
#define PULSE_BLOCKING_INTERRUPT_POSITION             4
#define PROGRAMMING_MODE_INTERRUPT_POSITION           5
#define DRIVER_ERROR_INTERRUPT_POSITION               6
#define SAW_WRITING_ERROR_INTERRUPT_POSITION          7
#define TIMES_WRITING_ERROR_INTERRUPT_POSITION        8
#define OPTIMIZED_ON_INTERRUPT_POSITION               9
#define DEAD_TIME_WRITING_ERROR_INTERRUPT_POSITION    10
#define FOUR_STEP_ON_INTERRUPT_POSITION               3
#define SAFE_MODE_INTERRUPT_POSITION                  12

//-------BEGGINING OF MEMORY SPACE FOR ISA----------------------
#define MEM_ADR                       0xf00000u
//-------BEGINNING OF ISA IO ADR--------------------------------
#define IO_ADR                        0x300u
//-------CHIP SELECT OFFSET-------------------------------------
#define chip_sel_offset               0x0000u

void write_mc(unsigned short int address, unsigned short int data);
/*
 This function writes data to modulator register.
 address - register address - for example: T_INPUT_1_X_T_OUTPUT_1
 data    - register input data
 It is also possible to write command using this function:
 write_mc(COMMAND_REGISTER, BLOCK_PULSES_COMMAND);
*/
unsigned short int read_mc(unsigned short int address);
/*
 This function reads from modulator register.
 address - register address - for example: T_INPUT_1_X_T_OUTPUT_1
*/
```

```
unsigned short int get_bit(unsigned short int value, unsigned short int position);
/*
 This function returns bit value from "value" at position "position".
 It can be used to detect concrete interrupt flag.
 get_bit(read_mc(INTERRUPT_REGISTER),SAFE_MODE_INTERRUPT_POSITION);
*/
#endif
```

D.2 mc_def.c

```
/*--------------------------------------------------------------------
Project:           MC modulator

by                Bc. Jakub Kucka
              (kuckajak@fel.cvut.cz)

component:  mc_def
description: here, all needed definitions and functions are to be found
             to use MC modulator this header has to be included
 use of particular registers is described in Thesis

This was created as a part of Master Thesis.
--------------------------------------------------------------------*/

#include "mc_def.h"
#include <conio.h>
#include <stdint.h>

void write_mc(unsigned short int address, unsigned short int data)
{
  uint16_t i;

  i = IO_ADR+(2*(uint16_t)address); //address*2 - because of writing 2 bytes
  outpw(i,data);
}

uint16_t read_mc(unsigned short int address)
{
  uint16_t i;

  i = IO_ADR+(2*(uint16_t)address);  //address*2 - because of writing 2 bytes
  return inpw(i);
}

unsigned short int get_bit(unsigned short int value, unsigned short int position)
{
  return ((value >> position)  & 0x0001);
}
```