THESIS
2
2001

This is to certify that the

dissertation entitled

A NEW OVERMODULATION METHOD BASE ON A
PROPORTIONAL REFERENCE SIGNAL GENERATOR

presented by

Andres J. Diaz-Castillo

has been accepted towards fulfillment
of the requirements for

Ph.D degree in Electrical Eng

Elias G Str
Major professor

Date 22 Aug 2000

PLACE IN RETURN BOX to remove this checkout from your record.
TO AVOID FINES return on or before date due.
MAY BE RECALLED with earlier due date if requested.

DATE DUE	DATE DUE	DATE DUE

A New Overmodulation Method Based on a Proportional Reference Signal Generator

By

Andres J. Diaz-Castillo

A DISSERTATION

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

Department of Electrical and Computer Engineering

2000

I

inve

the r

prop

limi

mem

outp

can h

the n

matio

digita

the pr

ABSTRACT

A New Overmodulation Method Based on a Proportional Reference Signal Generator

By

Andres J. Diaz-Castillo

In order to obtain the maximum fundamental output voltage from a three phase inverter, a transition from PWM to six-step mode needs to be done. Because of the nonlinear behavior of this overmodulation range, different methods have been proposed to linearize it. Those methods use complex equations and look-up tables, limiting their application to powerful microprocessors with considerable amount of memory. In this thesis an overmodulation method is proposed, which has a perfect output linearity and requires only linear equations for implementation. This method can be applied to both sinusoidal and space vector PWM. A new method to calculate the modulation index from the instantaneous phase values is also proposed. Mathematical analysis, simulations, real microprocessor implementation, and a stand-alone digital design are presented here to demonstrate the effectiveness and simplicity of the proposed method.

To my wife, Angeles

I would l

his support a

of my comm.

A special w

Zanardelli f

to the depar

to my friend

Thanks to m

goals in my

Emilio and

ACKNOWLEDGMENTS

I would like to express my gratitude to my advisor Professor Elias Strangas for his support and excellent guidance. I would also like to thank to the other members of my committee for their time and effort in being in my committee.

A special words of thanks to my lab colleagues in special, Ali Khurram and Wes Zanardelli for their friendship and their help with LaTex and computers. Many thanks to the department staff in special to Marilyn Shriver for her assistance. Special thanks to my friends Walter Zuluaga and Mary Sherby for their help with the corrections.

Thanks to my parents for always being my guide and inspiration in obtaining the big goals in my live. Finally, thanks to my wife Maria de los Angeles and my kids: Diego, Emilio and Leticia for their love, support and sacrifice.

TABLE OF CONTENTS

LIST OF FIGURES

Introd

Pulse Width M

This techniqu

distortion. Ho

capacity of th

of the three p

than the ma

voltage fun

the load.

The tra

the only

gradually

of the fu

reference

should ha

Linear

the *modula*

linearity is a re

overmodulation t

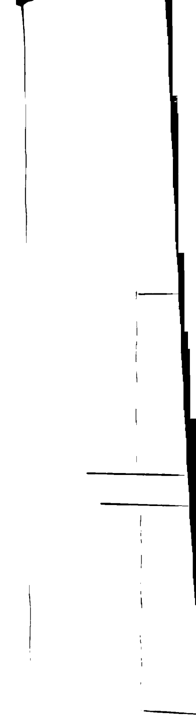
CHAPTER 1

Introduction

Pulse Width Modulation [PWM] has been used for decades in voltage source inverters. This technique produces a two-level output voltage with significantly low harmonic distortion. However, this technique does not permit utilizing the maximum voltage capacity of the DC bus. On the other hand, by applying a pure square wave to each of the three phases, we can obtain an output voltage fundamental of 1.27 times higher than the maximum obtainable with PWM. This 27% additional gain in the output voltage fundamental is obtained by allowing almost 30% of harmonic distortion at the load.

The tradeoff between harmonic distortion and output voltage fundamental is not the only issue here. It is necessary to design an over-modulation technique that gradually converts the sinusoidal voltage reference to a square wave as the amplitude of the fundamental increases. That means the modulation technique should provide a reference signal for each desired modulation index between 1 and 1.27. This technique should have the following three characteristics.

Linearity: The output voltage fundamental should have a linear relationship with the modulation index command. This is the most important characteristic, since linearity is a requirement of high performance control drives. Authors who proposed overmodulation techniques have sometimes focused on linearity at the expense of



simplicity or l

Simplicit

which do not

fore the over-

tributes an a

ware could m

sampling peri

Low total

have been a

each modula

known that b

to obtain a n

possible to o

output, by in

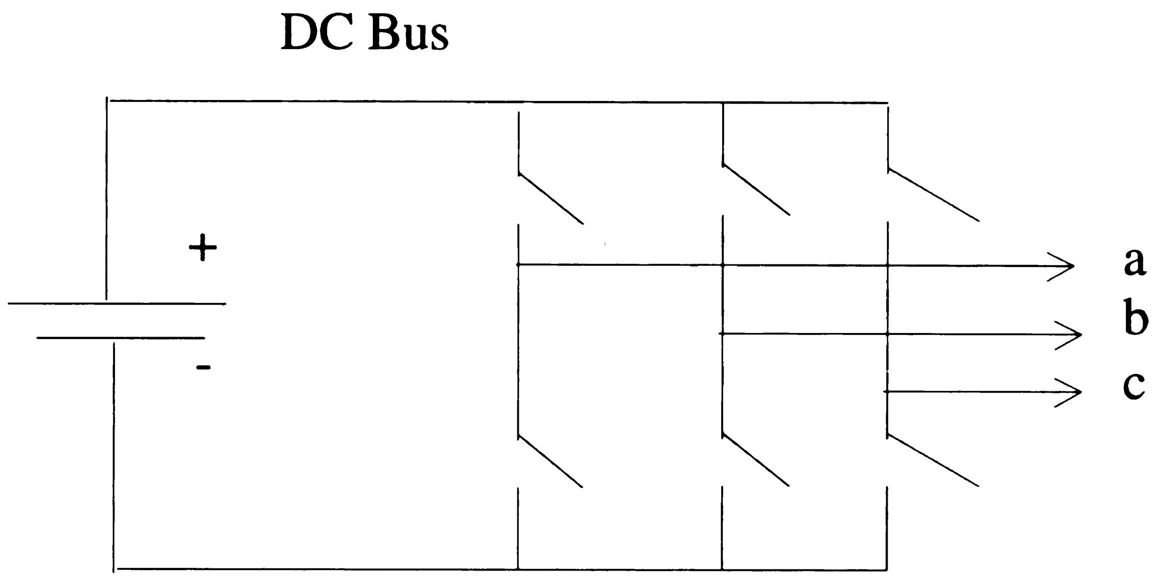


Figure 1.1. Voltage source inverter.

simplicity or harmonic distortion.

Simplicity: Many techniques are available to operate PWM in the linear range, which do not require complex hardware or software in their implementation. Therefore the over-modulation technique should retain this simplicity, since it only contributes an additional 27% extension to the linear range. Complex solution in hardware could mean additional cost. A complex solution in software could mean longer sampling period which will degrade the dynamics of the system.

Low total harmonic distortion (THD) path: Once the first two characteristics have been achieved it is desirable to obtain the minimum harmonic distortion for each modulation index. The THD of a square wave is about 30%. However, it is known that by dividing the overmodulation range into three subranges, it is possible to obtain a minimum low THD path. In the first subrange ($1 < m_i < 1.15$) it is possible to obtain higher fundamental output without increasing the THD at the output, by injecting third harmonics or using space vector technique. In the second sub-range

modified s

1.21 (the n

THD (1.6

the referen

In the n

analyzed. M

Some of the

are so comple

produce differ

into different

methods used

method which

sub-range ($1.15 < m_i < 1.21$) low harmonic distortion is obtained by converting the modified sinusoidal reference to a trapezoidal one. A maximum modulation index of 1.21 (the middle of the range where $1.15 < m_i < 1.27$) is obtained, with only 5% of THD (1/6 of the maximum 30%) at the load. In the third range ($1.21 < m_i < 1.27$) the reference signal is converted gradually into a square wave.

In the next chapter, different methods proposed for the overmodulation range are analyzed. Most provide a technique to extend the linearity beyond the linear range. Some of them provide analytical solutions, using Fourier series expansion, but some are so complex that they need to be implemented using look-up tables. These methods produce different harmonic distortions. Those that divide the overmodulation range into different subranges produce better results. After analyzing the more important methods used so far, their disadvantages and limitations are discussed. Finally, a method which satisfies the three characteristics mentioned previously is proposed.

Previ

In Pulse Wid

signal is com

period of the

of these signa

The output

to the modula

fundamental p

V_{dc} is the

when a square

maximum fund

This value is

CHAPTER 2

Previous Work

In Pulse Width Modulation (triangular comparison method) a sinusoidal reference signal is compared with a triangular signal in order to determine the conduction period of the switching devices. The modulation index is the ratio of the amplitude of these signals.

$$m_i = \frac{V_{p(sinusoidal)}}{V_{p(triangular)}} \tag{2.1}$$

The output voltage fundamental obtained for the two-level PWM is proportional to the modulation index when it is in the range of 0-1 (linear range). The maximum fundamental peak voltage in the linear range is

$$V_{lmax} = m_{imax}\frac{V_{dc}}{2} = \frac{V_{dc}}{2}. \tag{2.2}$$

V_{dc} is the voltage of the DC bus and m_{imax} is the modulation index. However, when a square wave is applied to each one of the three phases (six-step mode) the maximum fundamental peak voltage is

$$V_{max} = \frac{2}{\pi}\int \frac{V_{dc}}{2}\sin t dt = \frac{2V_{dc}}{\pi}. \tag{2.3}$$

This value is the maximum obtained for a two level voltage source inverter. When

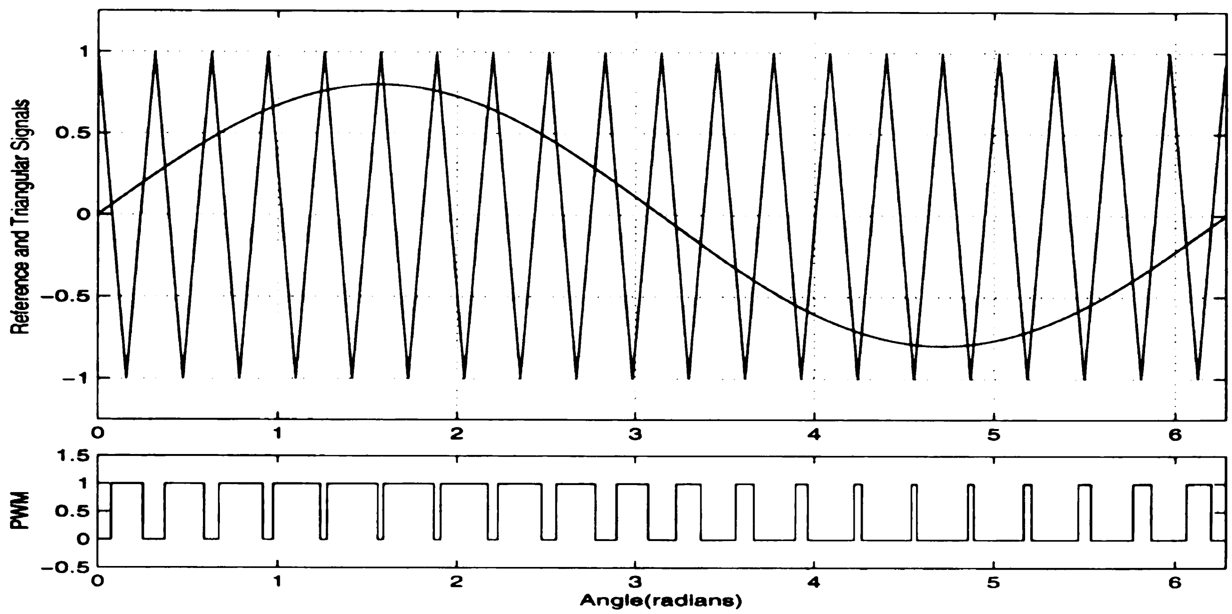


Figure 2.1. Pulse width modulation.

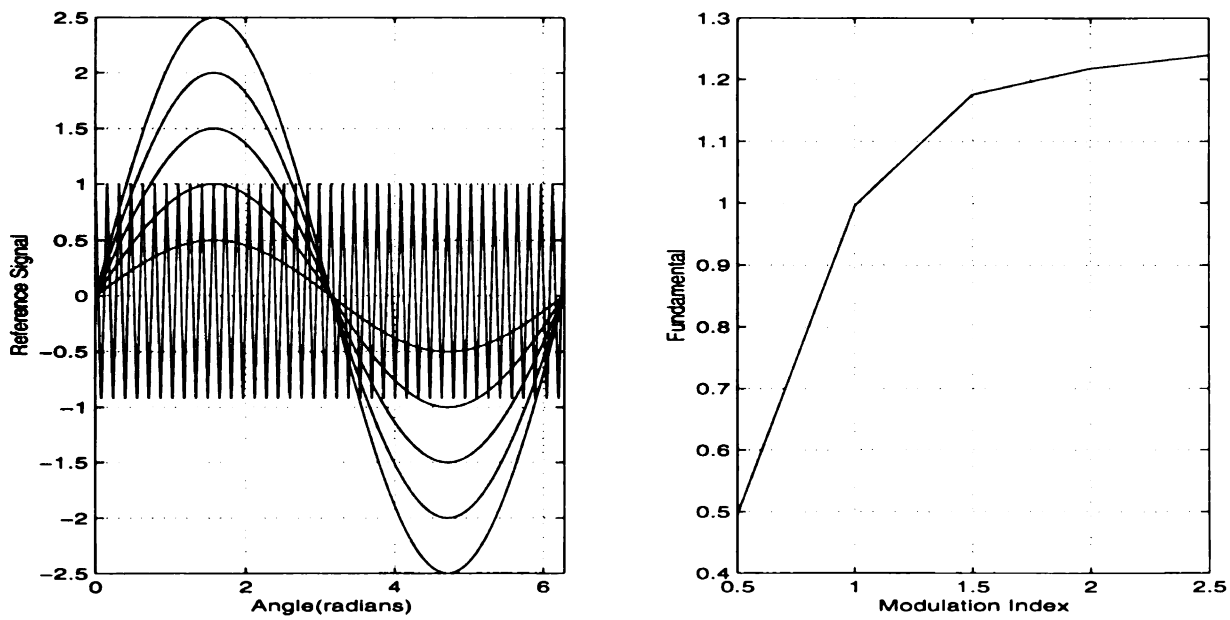


Figure 2.2. Increasing reference signal beyond one.

this value r

in fundame

Therefor

forming the

is called the c

PWM into si

2.1 Co

The most ob

the sinusoida

problems wi

linearly wit

an infinite a

equation to

index comm

index m_i,

produce th

Even if

a large am

this value is compared with the maximum obtained with PWM, the additional gain in fundamental output voltage becomes

$$(gain) = \frac{\frac{2V_{dc}}{\pi}}{\frac{V_{dc}}{2}} = \frac{4}{\pi} = 1.27. \tag{2.4}$$

Therefore, an additional 27% of fundamental voltage can be obtained by transforming the PWM into six-step mode. This modulation index range from 1 to 1.27 is called the overmodulation range, and the method used to smoothly transform the PWM into six-step mode is known as an overmodulation method.

2.1 Compensating Modulation Index

The most obvious overmodulation method is to continue increasing the amplitude of the sinusoidal reference beyond the amplitude of the triangular peak. There are two problems with this method. First, the output voltage fundamental does not change linearly with respect to the amplitude of the sinusoidal reference signal. Second, an infinite amplitude sinusoidal reference is needed to reach the six-step mode. An equation to linearize the output voltage fundamental with respect to the modulation index command was proposed by Kerman *et al.* [9]. For a given desired modulation index m_i, the equation provides a compensated modulation index m_{icp}, which will produce the desired output voltage

$$m_i = \frac{V_o}{V_{bus}/2} = \frac{2}{\pi}\left[\sin^{-1}(\frac{1}{m_{icp}}) + \frac{1}{m_{icp}}\sqrt{1 - \frac{1}{m_{icp}}}\right]. \tag{2.5}$$

Even if a look-up table is used to implement this equation, a reference signal with a large amplitude is impractical to implement.

2.2 T

Another a

this work.

lower the p

wave detern

signal witho

the third ha

that the opti

The equation

With this

Houldsworth a

contributes to

output voltag

increment in t

range, since it

2.3 Spa

Another PWM

Pulse Width M

for each phase

the desired vo

linear range of

The switching s

2.2 Third Harmonic Injection

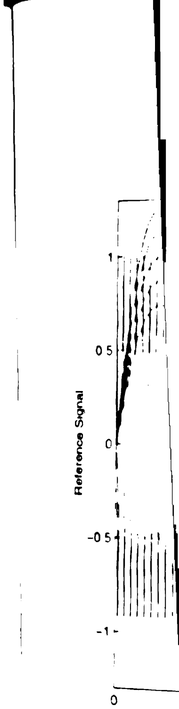
Another approach to correct this non-linearity was proposed by Houldsworth [6]. In this work, third harmonics are added to the sinusoidal reference signal in order to lower the peak of the fundamental. This action makes both signal fit inside the square wave determined for the DC source. Third harmonic can be added to the reference signal without increasing the total harmonics distortion of the output voltage, since the third harmonic currents are eliminated in the three phases load. It was shown that the optimal value of third harmonics is one 6th of the fundamental amplitude. The equation for the new reference signal is therefore

$$v_a = V \sin \omega t + \frac{1}{6} V \sin 3\omega t. \tag{2.6}$$

With this method, the linearity is extended from $m_i = 1$ to $m_i = 1.155$. Houldsworth also shows that only the third harmonic, not other triplen harmonics, contributes to increase the fundamental. This increment of 15% in the fundamental output voltage is obtained without increasing the harmonic distortion. However, this increment in the output voltage fundamental does not cover the entire modulation range, since it can not reach $m_i = 1.27$.

2.3 Space Vector Modulation

Another PWM method with an important role in overmodulation is the Space Vector Pulse Width Modulation (SVPWM). This method determines the state of switches for each phase of a voltage source inverter, based on the module and the angle of the desired voltage. No carrier (triangular wave) is needed with this method. The linear range of SVPWM is 15% greater than Sinusoidal Pulse Width Modulation. The switching states of the three phases are determined by selecting one of the eight



possible space

is connected t

bus) do not pr

vectors. The

and the rest e

Those six

Given a desire

to which sect

two adjacent

The two r

plying them b

for the zero v

The result

However, the

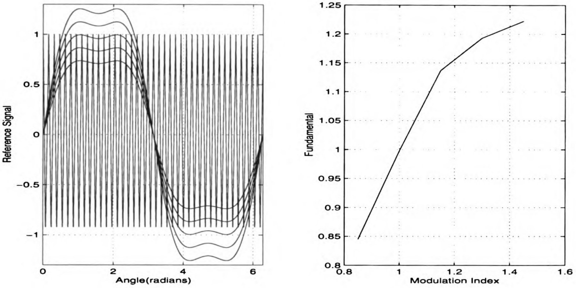


Figure 2.3. Third harmonic injection.

possible space vectors (0-7). Space Vectors 0="000" and 7="111" (0 means the phase is connected to the negative DC bus, and 1 means it is connected to the positive DC bus) do not produce any effective voltage on the three-phase load, and are called zero vectors. The other six vectors (1-6) have at least one of the three phases equal to 1 and the rest equal to 0.

Those six vectors are located 60° apart, dividing the circle into six regions (I-VI). Given a desired vector voltage, V_r (module and angle), the SVPWM algorithm finds to which sector (I-VI) it corresponds. Then the desired vector is projected into the two adjacent space vectors that define the region.

The two resulting vectors, V_a and V_b, are converted to time t_a and t_b, by multiplying them by the switching period T. The remaining time $t_o = T - t_a - t_b$ is used for the zero vectors 0 and 7.

The resulting voltage is the vector summation of V_a and V_b, which are 60° apart. However, the corresponding times t_a and t_b are added arithmetically and restricted

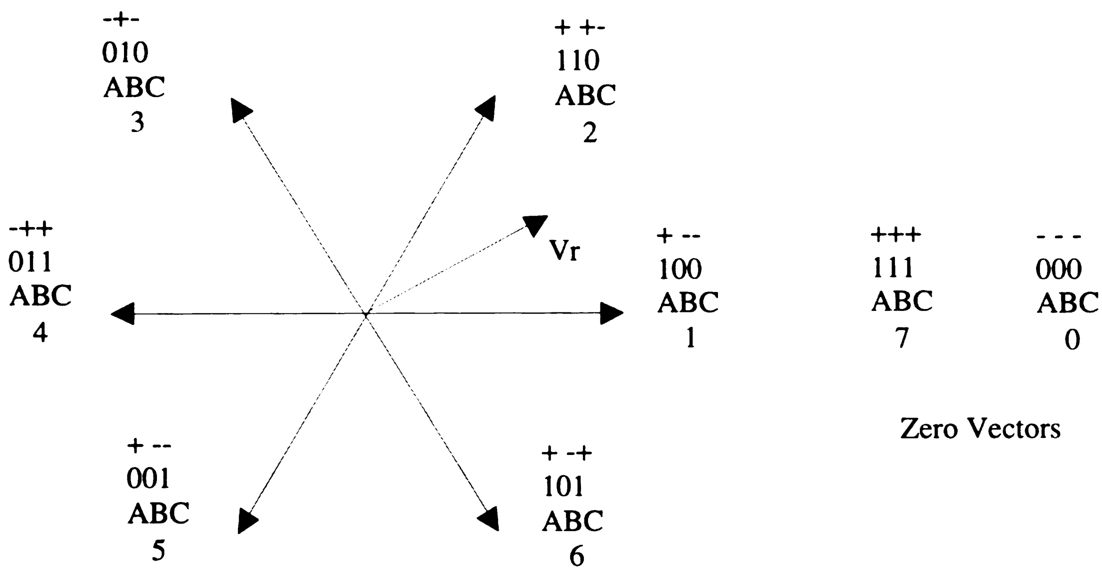


Figure 2.4. Space vectors.

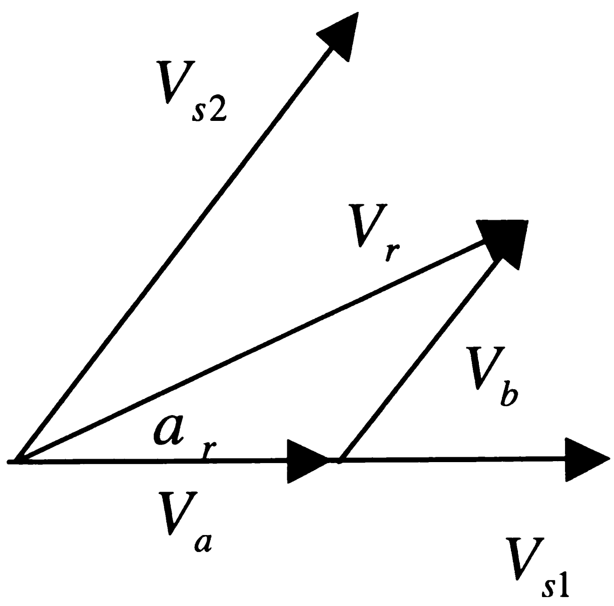


Figure 2.5. Space vector projection.

Vect
V_a =
V_a =

to be less or

the angle. Fo

obtained is e

the sector (3

The set o

angle, forms

with straigh

in order to h

middle of th

corresponds

Surprisin

ies [6]. Ho

SVPWM c

and the pur

fundamenta

Vector Projection	Times	Definitions
$V_a = \frac{2}{\sqrt{3}} V_r \sin\alpha_r$	$t_a = \frac{V_a}{V_s} T$	T =switching period
$V_a = \frac{2}{\sqrt{3}} V_r \sin(\frac{\pi}{3} - \alpha_r)$	$t_a = \frac{V_b}{V_s} T$	V_s = Space-Vector. amp.=$\frac{2}{3} V_{dc}$
	$T_o = T - t_a - t_b$	(V_r, α_r) = reference voltage
		(V_c, α_c) = Command voltage
		$(V_r, \alpha_r) = (V_c, \alpha_c)$ Linear range

Table 2.1. Space Vector Equations

to be less or equal than T. Therefore, the maximum obtainable voltage varies with the angle. For example, in the border of the sector (0° or 60°), the maximum voltage obtained is equal to that of the space vectors ($V_{max} = 2V_{dc}/3$), while in the middle of the sector (30°), the maximum voltage is

$$V_{rmax} = \frac{\sqrt{3}}{2} V_s = \frac{V_{dc}}{\sqrt{3}}. \tag{2.7}$$

The set of points, which are described by the maximum possible voltage for each angle, forms a hexagon. This hexagon can be drawn by joining the six space vectors with straight lines. The trajectory of the resulting vector V_r should describe a circle, in order to have zero harmonics distortion. Therefore, the value of the radius in the middle of the chord is the maximum voltage obtained in the linear range. This value corresponds to a modulation index of

$$m_i = \frac{\frac{\sqrt{3}}{2} V_{max}}{\frac{V_{dc}}{2}} = \frac{\sqrt{3}\frac{2}{3} V_{dc}}{V_{dc}} = \frac{2}{\sqrt{3}} = 1.155. \tag{2.8}$$

Surprisingly, this is the same value obtained with the injection of third harmonics [6]. However, when analyzed, the equivalent reference waveform generated by SVPWM contains more than third harmonics. The difference between this waveform and the pure sinusoidal looks like a triangular waveform with a frequency 3 times the fundamental. Therefore, all harmonics of this injected signal will be triplen harmonies of the fu

the output v

Blasko [2] a

that SVPW

load neutra

overmodul

complemen

which mod

range of

reach grad

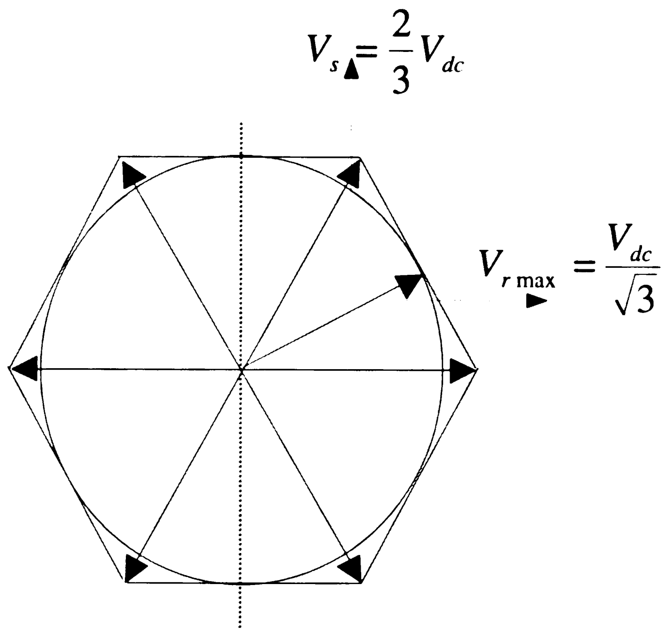


Figure 2.6. Limit of the linear range in SVPWM.

ics of the fundamental. As Houldsworth [6] shows, these harmonics do not appear at the output but neither do they contribute to the fundamental (except the third one). Blasko [2] and Chung [3] explain this increment of 15% in the linear range by showing that SVPWM inherently compensated the movement of the voltage of the three-phase load neutral that occurs because of non-sinusoidal values of the three phases. The overmodulation range of SVPWM is ($1.15 \leq m_i \leq 1.27$). Holtz [5] proposed two complementary techniques to obtain the remaining 12% of the fundamental: Mode I, which modifies the module of the desired voltage to fit in the hexagon and covers the range of ($1.15 < m_i \leq 1.21$). Mode II, which modifies modulus and angle, in order to reach gradually the six-step mode and covers the range of ($1.21 < m_i \leq 1.27$).

2.3.1

The circle

two points

points are

is compens

increasing

new angles,

circumference

the limit of

A more

than 0.35

2.8.

Holtz [

with α'_1 or

t_a, t_b and

2.3.1 Mode I

The circle that corresponds to the desired voltage vector intersects the hexagon in two points in each sector as shown in Figure 2.7 . The angles that correspond to those points are called α_1 and α_2. The area between α_1 and α_2 that is outside the hexagon is compensated with an area between the hexagon and the circle. This is done by increasing the voltage reference. The new circle will intersect the hexagon at two new angles, α'_1 and α'_2. The common area, inside the hexagon and between the two circumferences, determines the compensating area. Using the equal area criterion, the limit of this modulation index can be calculated from the geometry:

$$\frac{\sqrt{3}}{4}V_s^2 = \frac{\pi V_r^2}{6} \tag{2.9}$$

$$V_r = \sqrt{\frac{3\sqrt{3}}{2\pi}}V_s = \sqrt{\frac{3\sqrt{3}}{2\pi}\frac{2}{3}}V_{dc} = \sqrt{\frac{2}{\sqrt{3}\pi}}V_{dc} \tag{2.10}$$

$$m_i = \frac{V_r}{V_{max}} = \frac{\sqrt{\frac{2}{\sqrt{3}\pi}}V_{dc}}{\frac{V_{dc}}{2}} = \sqrt{\frac{8}{\pi\sqrt{3}}} = 1.2125. \tag{2.11}$$

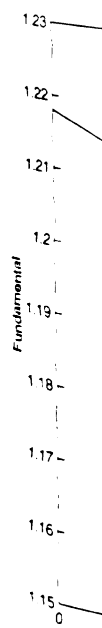
A more precise value of m_i can be obtained with Fourier series, but the error is less than 0.35%. The relation between α'_1 and the modulation index is shown in Figure 2.8.

Holtz [5] does not provide the equations that relate the desired modulation index with α'_1 or with the new reference vector amplitude, but provides the equations for t_a, t_b and t_0 when the reference vector traces the hexagon.

$$t_a = T\frac{\sqrt{3}\cos\alpha - \sin\alpha}{\sqrt{3}\cos\alpha + \sin\alpha} \tag{2.12}$$

$$t_b = T - t_a \tag{2.13}$$

$$t_0 = 0 \tag{2.14}$$


1.23
1.22
1.21
1.2
1.19
1.18
1.17
1.16
1.15
0
Fundamental

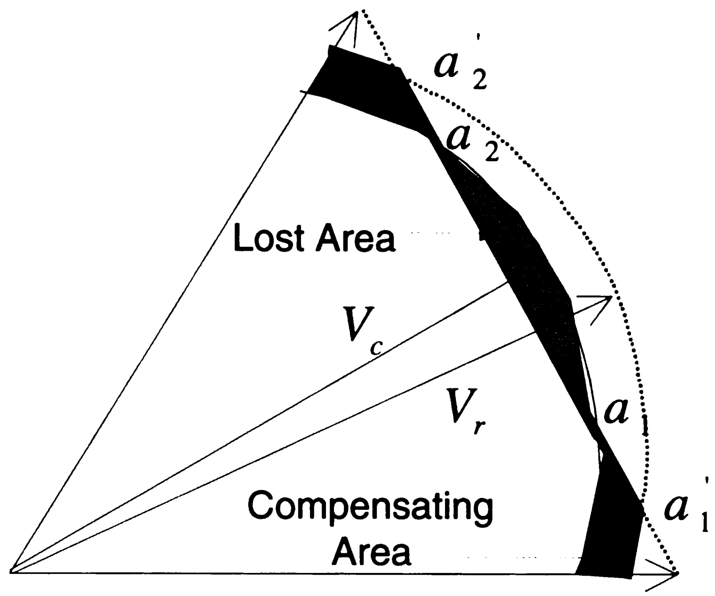


Figure 2.7. Overmodulation Mode I.

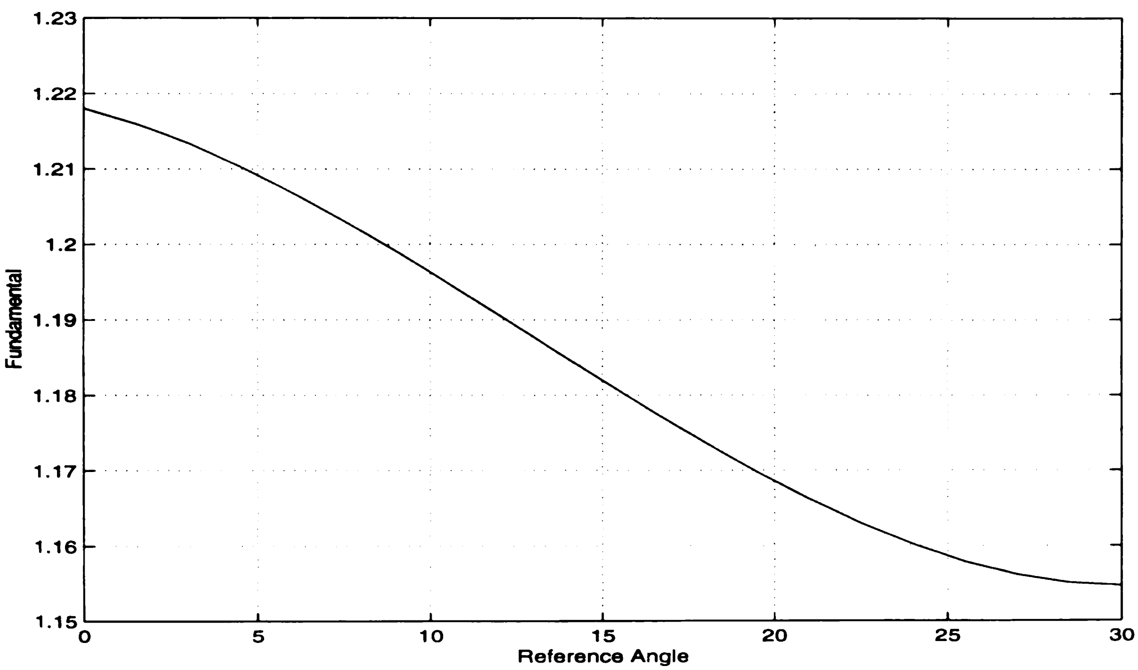


Figure 2.8. Modulation index vs. reference angle in Mode I.

Since t

simplified v

Other a

hexagon. T

table 2.11, a

with a propo

Those eq

that it is need

modulation i

values of m_i

the fundamen

by using the

with the area

m_i and α_1 is

$$m_i =$$

$$V_* =$$

Since the equation for t_a is too difficult to implement, the author suggests a simplified version

$$t_a = T\frac{\alpha}{\pi/3}. \tag{2.15}$$

Other authors suggest another approach to bring the reference voltage within the hexagon. The times t_a and t_b are calculated for the new V_r (using the equations in table 2.1), and the portion of V_r outside the hexagon is brought back to the hexagon with a proportional equation:

$$\begin{aligned}
&for\ \ t_a + t_b > T: \\
t_a^{'} &= \frac{t_a}{t_a + t_b} && (2.16) \\
t_b^{'} &= \frac{t_b}{t_a + t_b} \quad or\ \ t_0 = T - t_a. && (2.17)
\end{aligned}$$

Those equations do not require powerful hardware to implement. Therefore, all that it is needed to implement Mode I is an equation that relates the v_{ref} to the desired modulation index. There are three possible ways to do this. First, by obtaining the values of m_i for each V_r, and using them in a look-up table. This is done by finding the fundamental component of the new signal derived from each value of V_r. Second, by using the equal area criterion, where the area outside the hexagon is compensated with the area inside of the hexagon. Third, by using Fourier series, a relation between m_i and α_1 is found. This method was proposed by Lee [10]. The equation for m_i is

$$\begin{aligned}
m_i &= \frac{2}{\sqrt{3}}\left[\frac{1}{\sqrt{3}}\frac{9}{2\pi}(1+\sqrt{3}\alpha_1)\alpha_1 + (\frac{9}{2\pi}\alpha_1 - 3 - 9\frac{\sqrt{3}}{2\pi})\sin\alpha_1 + \frac{3}{2}\alpha_1\right] && (2.18) \\
V_r &= \frac{V_{dc}}{\sqrt{3}\cos(\frac{\pi}{6} - \alpha_1)}. && (2.19)
\end{aligned}$$

Given

new refer

equations

can give s

not guara

mode I m

implement t

the processi

to the angle

In genera

(1.15 ≤ m_i

The most i

overmodula

six-step mo

2.3.2 M

An additio

mode II. In

index betw

between 0

correspond

angle rema

range (α_h

Figure

Given a modulation index m_i, α_1 can be found using numerical methods. Then the new reference vector V_r is found from equation (2.19). The implementation of both equations is impractical, since they require a lot of computation. Equal area criterion can give simpler equations but perfect linearity between m_i and the fundamental is not guaranteed. Bakhsay *et al.* [1] proposed another approach that works in the mode I modulation range. A classification algorithm (neural network) is used to implement the space vector modulation. Although the authors claim a reduction in the processing time of the algorithms, the equation that relates the modulation index to the angle α_1 is still complex

$$m = \frac{1}{2}\frac{1}{(\pi/6 - \alpha_t)} \ln \frac{1 + \sin(\pi/6 - \alpha_t)}{1 - \sin(\pi/6 - \alpha_t)}. \tag{2.20}$$

In general, Mode I is a method to obtain the modulation index in the range of ($1.15 \leq m_i \leq 1.21$) by converting the modified sinusoidal reference in a trapezoid. The most important characteristic of this method is that it covers about half of the overmodulation range, while the method only produces 20% of the distortion of the six-step mode.

2.3.2 Mode II

An additional 6% of the fundamental output (from 1.21 to 1.27) can be obtained using mode II. In this method the angle and the modulus are modified. Given a modulation index between 1.21 and 1.27, a holding angle α_h is selected. For commanded angles between 0° and α_h, the reference angle remains equal to that of the space vector that corresponds to 0°. For commanded angles between ($60° - \alpha_h$) and 60°, the reference angle remains equal to the space vector that corresponds to 60°. The remaining angle range (α_h, $60° - \alpha_h$) will be expanded to cover the sector (0°,60°).

Figure 2.9 shows how the reference angle is modified, given command angles with

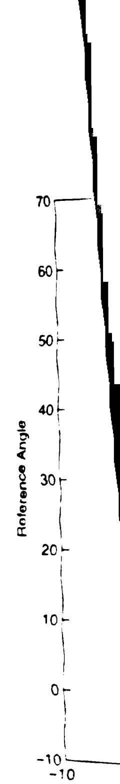


Figure 2

values betv

reference n

modulatio

linear, and

In 199

a function

This ex

mode I. Th

requires th

express th

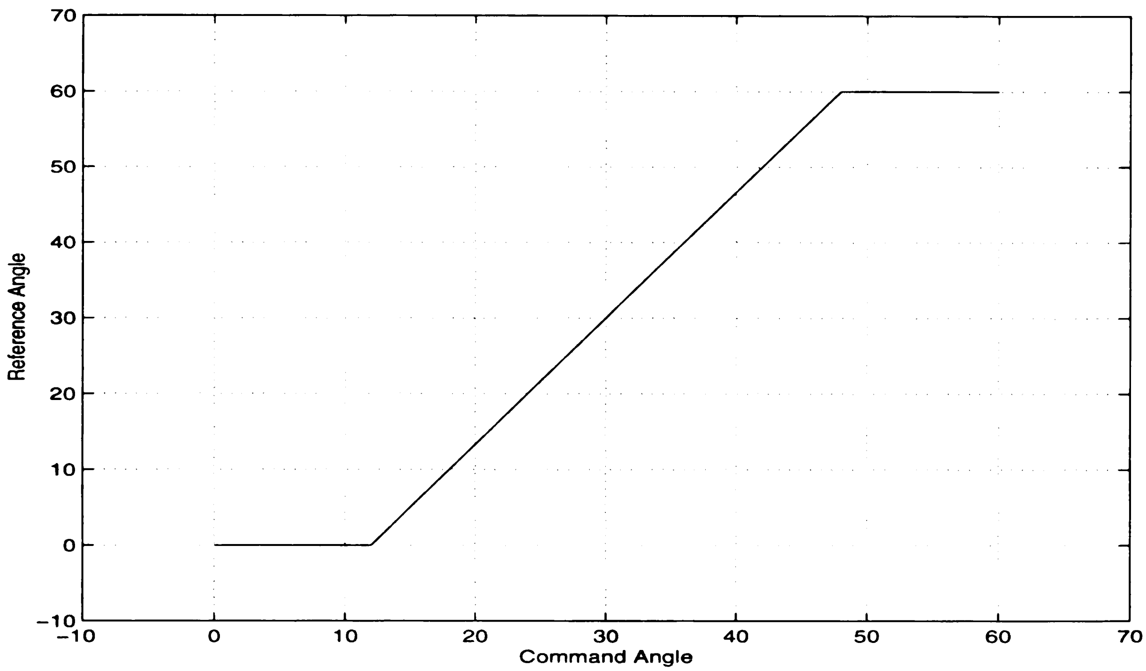


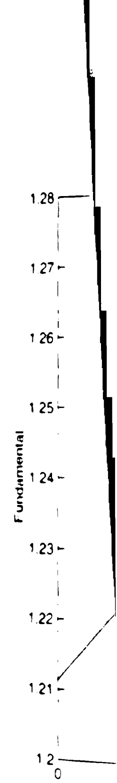
Figure 2.9. Mode II. Reference angle vs. command angle (holding angle=12°).

values between 0° and 60°. The holding angle for this case is 12°. In this range the reference magnitude will be determined by the hexagon. The relation between the modulation index and the holding angle is shown in figure 2.10. This relation is not linear, and was proposed first by Holtz [5] in a form of a graph.

In 1998, Lee [10] presented the analytical expression for the modulation index as a function of the holding angle. This equation is

$$m_i = \frac{\sin(\frac{\pi}{6} - \alpha_h)}{\frac{\pi}{6} - \alpha_h}. \tag{2.21}$$

This expression has the same drawbacks as the one used for the reference angle in mode I. The modulation index appears in the left side of the equation. This equation requires the use of numerical methods or the use of a look-up table, in order to express the holding angle as a function of the modulation index. In his classification



algorithm. B

of the holdin

m

As in th

equation. O

entire overn

by both me

The ref

to trapezoid

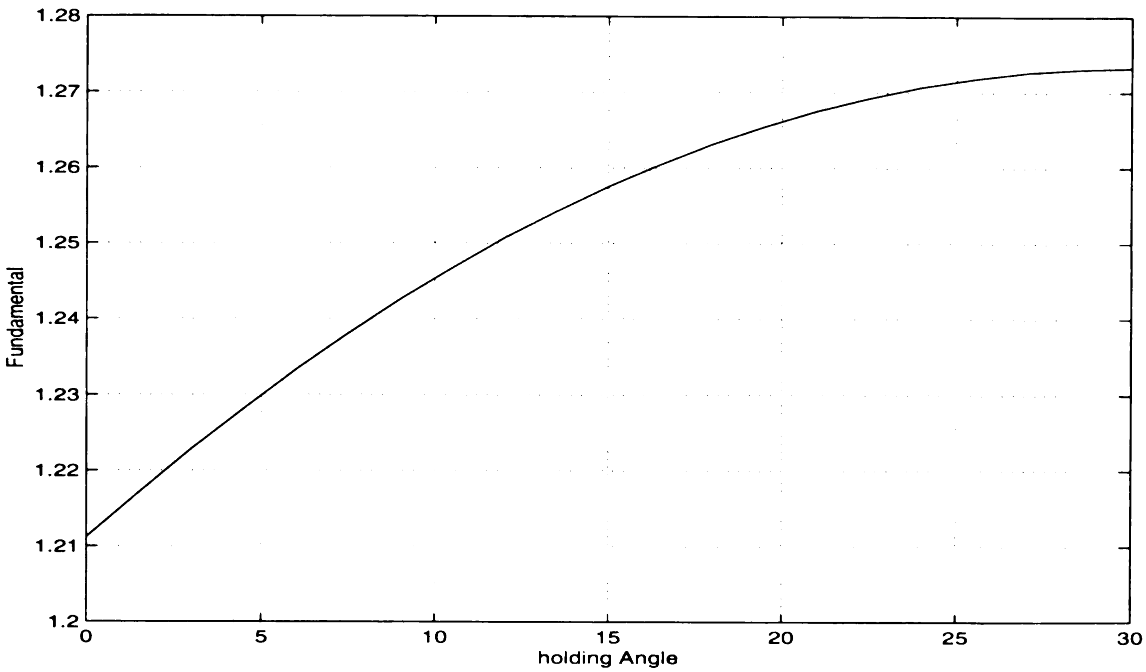


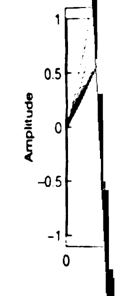
Figure 2.10. Fundamental vs. holding angle in Mode II.

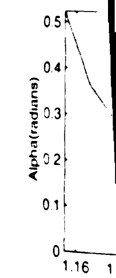
algorithm, Bakhshai [1] proposed another equation for modulation index as a function of the holding angle

$$m = \frac{4\sqrt{3}}{\pi^2}[2\alpha_h \cos(\frac{\pi}{6} - \alpha_h) + 2\sin(\frac{\pi}{6} - \alpha_h) + \alpha_h \cos(\frac{\pi}{6} - \alpha_h)$$
$$- \sin(\frac{\pi}{6} + \alpha_h) + \alpha_h \sin\alpha_h + \cos\alpha_h]. \tag{2.22}$$

As in the previous method, Bakhshai uses a look up table to implement this equation. Overmodulation Mode I and mode II are used together in order to cover the entire overmodulation range in SVPWM (1.15-1.27). The reference signal generated by both methods and their harmonic distortion is plotted in figure 2.11.

The reference signal in Mode I is transformed from modified-sinusoidal waveform to trapezoidal waveform. In Mode II, the signal is transformed from trapezoidal to





square wave

of the meth

II implemen

2.3.3 M

Since Holtz

versions of

additional c

proposed a

on Space Va

tion point of

index) for th

hexagon. Th

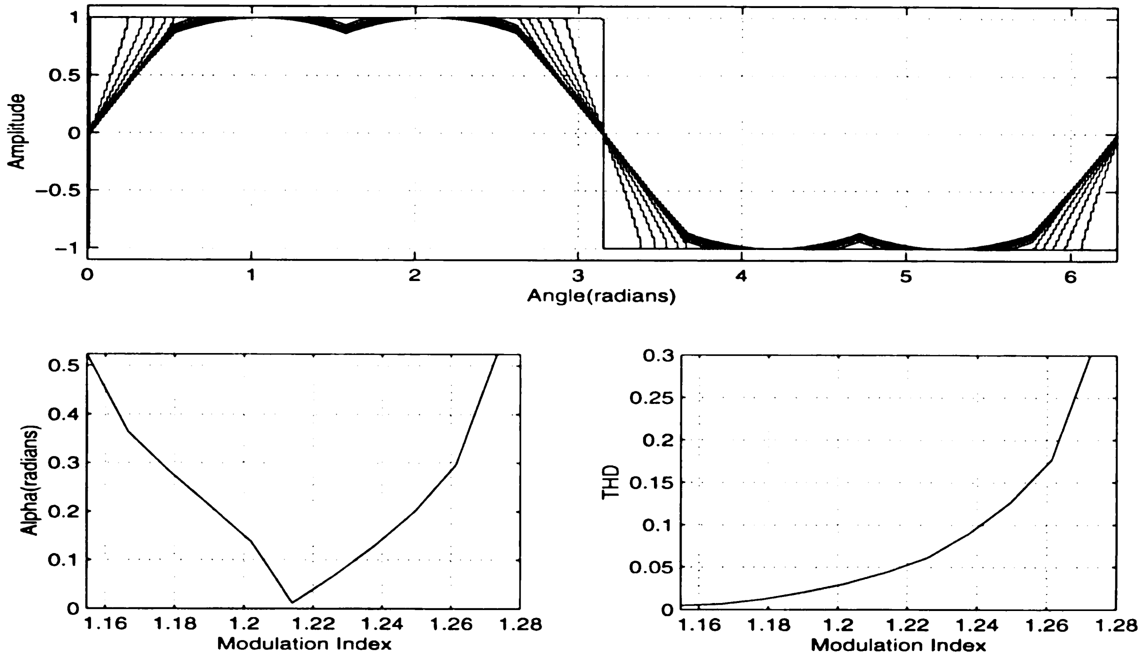


Figure 2.11. Reference signal and THD for mode I-II.

square waveform. The overmodulation method Mode I-Mode II shows the best THD of the methods proposed so far. However, all the proposed versions of Mode I-Mode II implementation require a look-up table because of their complex equations.

2.3.3 Mode III

Since Holtz proposed Mode I and mode II, many authors have proposed different versions of them. However, these two methods require two different coding, and additional code to determine which method should be used. Diaz and Strangas [4] proposed a method for the entire range of the overmodulation ($1 < m_i \leq 1.27$) based on Space Vector Modulation. This method holds the reference angle at the intersection point of the hexagon and the circumference (described for the desired modulation index) for the entire command angle, when the desired vector module is outside the hexagon. This method has significant advantages over previous methods. One advantage is

advantage

index is ex

a powerful

the necessi

The me

one algorit

the process

shown in fi

a powerful

function (a

2.4 A

Kaura and

modulation

it increases

that of the

1. the sign

square wav

cut at the

a reference

equal to 1

the overm

S with m

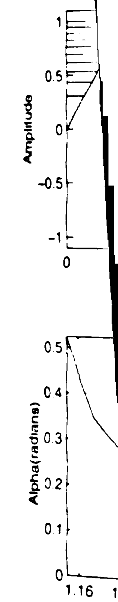
vantage is easy implementation since only the command angle is modified. Another advantage is that the equation of the variable to be changed with the overmodulation index is expressed explicitly. Therefore, this method can be implemented online, if a powerful processor is available, or an easy look-up table can be generated without the necessity of interactive methods. The equation for α_s is

$$\alpha_s = \arcsin(\frac{\sqrt{3}m_i}{2}) - \frac{\pi}{6}. \tag{2.23}$$

The method can be applied to the entire overmodulation range. Therefore, only one algorithm has to be implemented, saving memory code and execution time in the processor. The reference signal, angle α_1, and THD distortion for method III is shown in figure 2.12. Despite all the advantages discussed above, a lookup table or a powerful processor has still to be used, since the equation requires a trigonometric function (arcsin).

2.4 Adding a Square Wave to the Reference

Kaura and Blasko [8] proposed another method to extent the linearity to the overmodulation range. A square wave is added to the original sinusoidal waveform. When it increases beyond the DC bus voltage, the new reference signal is made equal to that of the DC bus. For a sinusoidal reference command with amplitude less than 1, the signal is not altered. For values of modulation index between 1 and 1.27, a square wave of amplitude S is added at the bottom of the reference and the total is cut at the top, making it look like a trapezoidal. The value of S will be equal to 0 for a reference signal less than 1 and will be equal to 1 for a reference signal command equal to 1.27, when the new reference signal will be a square wave. The value of S in the overmodulation range is found using Fourier analysis. The equation that relates S with $m_i(V)$ is



V is the

and an itera

the authors

K_s is a

easier to im

the limit of

in the funda

The refe

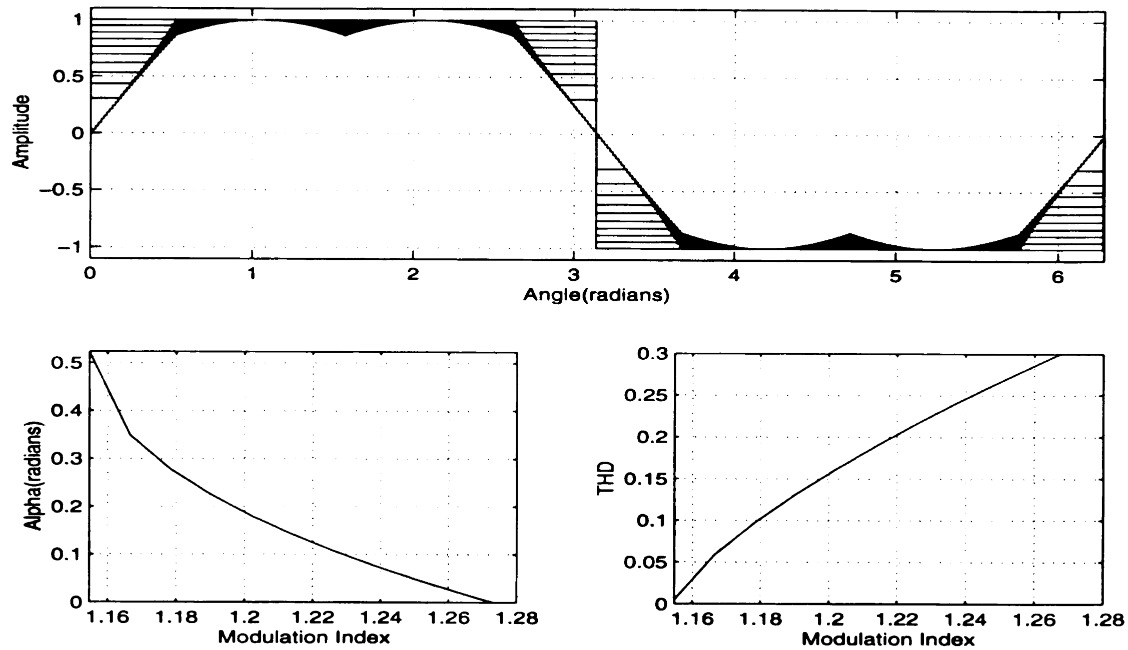


Figure 2.12. Reference signal and THD for Mode III.

$$V = \frac{4}{\pi}S + V\frac{2}{\pi}\left[\sin^{1}\left(\frac{1-S}{V}\right) + \left(\frac{1-S}{V}\right)\sqrt{1-\left(\frac{1-S}{V}\right)}\right]. \tag{2.24}$$

V is the output voltage fundamental. In this equation, S can not be easily isolated and an iterative method has to be used. In order to implement this method online, the authors proposed an approximation

$$S = K_s(m_i - 1)^2. \tag{2.25}$$

K_s is a constant; its optimal value, over the entire range is 5.5. Equation 2.25 is easier to implement in a DSP than the previous equation. However, the error of S at the limit of the range (1.27) using this equation is about 40%. Fortunately, the error in the fundamental is less than that.

The reference signal and the value of S versus modulation index are shown in the

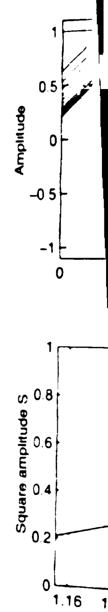


figure 2.13.

the graph. T

approximate

desired one b

change the fa

before, imple

tables or an a

2.5 Res

In this metho

to square wav

modulation in

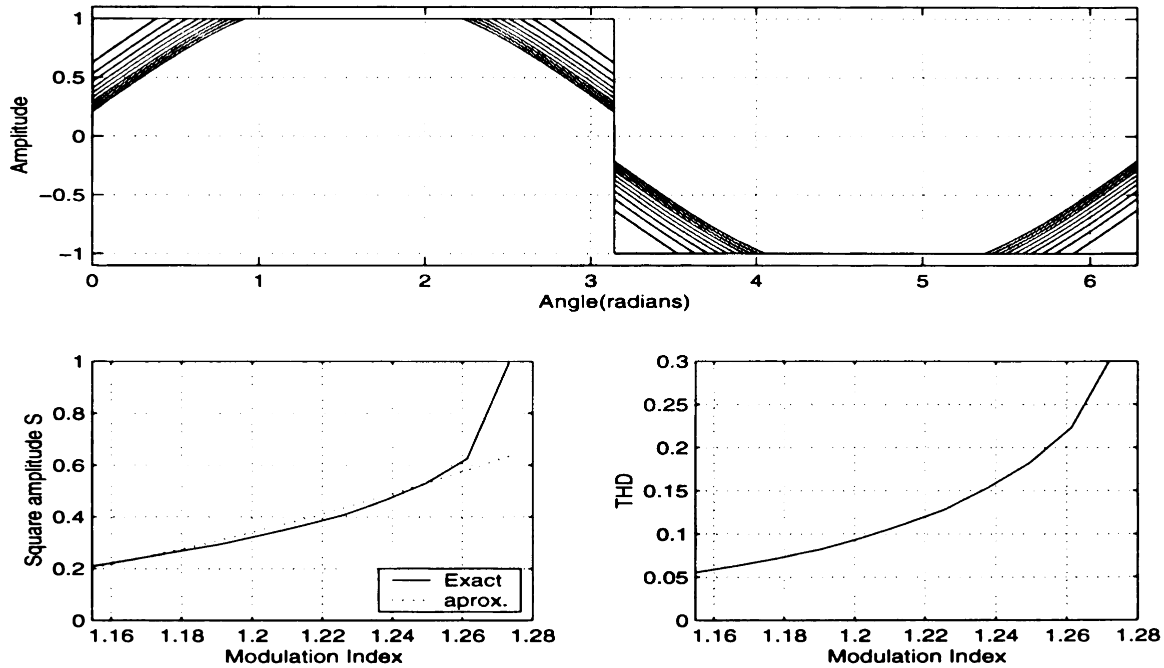
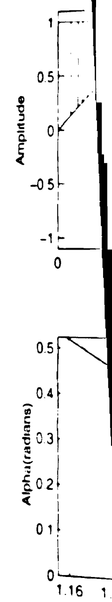


Figure 2.13. Square wave addition method.

figure 2.13. The amount of square wave added is plotted in the left bottom side of the graph. The relation between S and the modulation index is nonlinear. When the approximate equation is used (dotted line), the square wave added differs from the desired one by about 40% at the end of the range (1.27). The authors suggested to change the factor K_s in order to correct that difference. As other methods discussed before, implementation of this method requires selection between the use of look-up tables or an approximate curve (sacrificing linearity).

2.5 Reshaping the Modulation Command.

In this method, proposed by Kaura [7], the reference signal is switched from sinusoidal to square wave for a range of angles determined by the modulation index. Given a modulation index in the range of (1-1.27), an angle α_s is calculated by using Fourier



analysis, an

The refe
in Figure 2
before: it d
to the mod
the impleme

The pro
and sinusoi
range. This

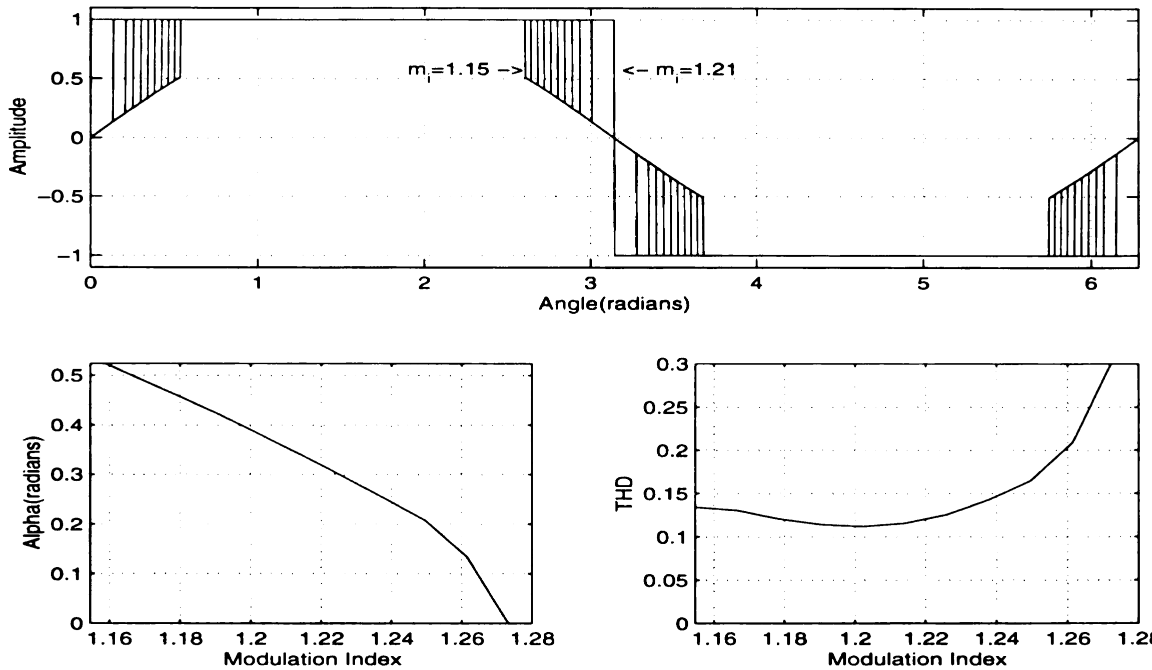


Figure 2.14. Reshaping the modulation command method.

analysis, and the equation is

$$m_i = \frac{4}{\pi}\cos(\alpha_s) + \frac{m_i}{\pi}[2\alpha_s - \sin(2\alpha_s)]. \tag{2.26}$$

The reference signal, α_s, and the THD, for different modulation indexes, are shown in Figure 2.14. This method suffers the same problem than the others discussed before: it does not offer an equation where α_s was expressed explicitly with respect to the modulation index. Therefore iterative methods and lookup tables are used in the implementation of this method.

The previous method and this method suggest the idea of interchanging square and sinusoidal waves in order to obtain the reference signal in the overmodulation range. This idea was used in the technique proposed in this work.

Prop

After anal

any overm

signal into

theory, wh

are shown

the maxim

problem of

this way of

3.1 D

The propos

generating

and square

angle of 30

CHAPTER 3

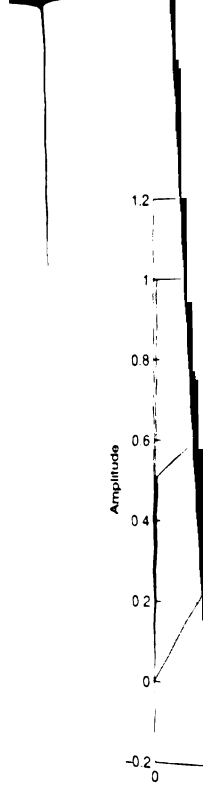
Proposed Method

After analyzing all the methods in the previous section, it can be concluded that any overmodulation method consists of gradually converting a sinusoidal reference signal into a square signal. Even methods originally described as using space vector theory, when they are analyzed as using an equivalent triangular-comparison method, are shown to do the same. This way of reducing a complex problem, obtaining the maximum output voltage from a three-phase inverter, to a simple mathematical problem of waveform transformation, has not been discussed directly so far. However, this way of isolating the problem suggests new method of doing overmodulation.

3.1 Description of the Proposed Method

The proposed method, Proportional Reference Signal Generator (PRSG), consists of generating reference signals by linearly interpolating between the unitary sinusoidal and square waveform values. For example, for a modulation index of 1.13 and an angle of 30°, the value of the proposed reference signal will be

$$v_{ref}(m_i, \theta) = \frac{(Square(\theta) - \sin(\theta))}{m_{isq} - m_{isin}}(m_i - m_{sin}) + \sin(\theta) \tag{3.1}$$



where v_{ref} i

that corresp

in the linea

of $sin(\theta)$, co

signals:

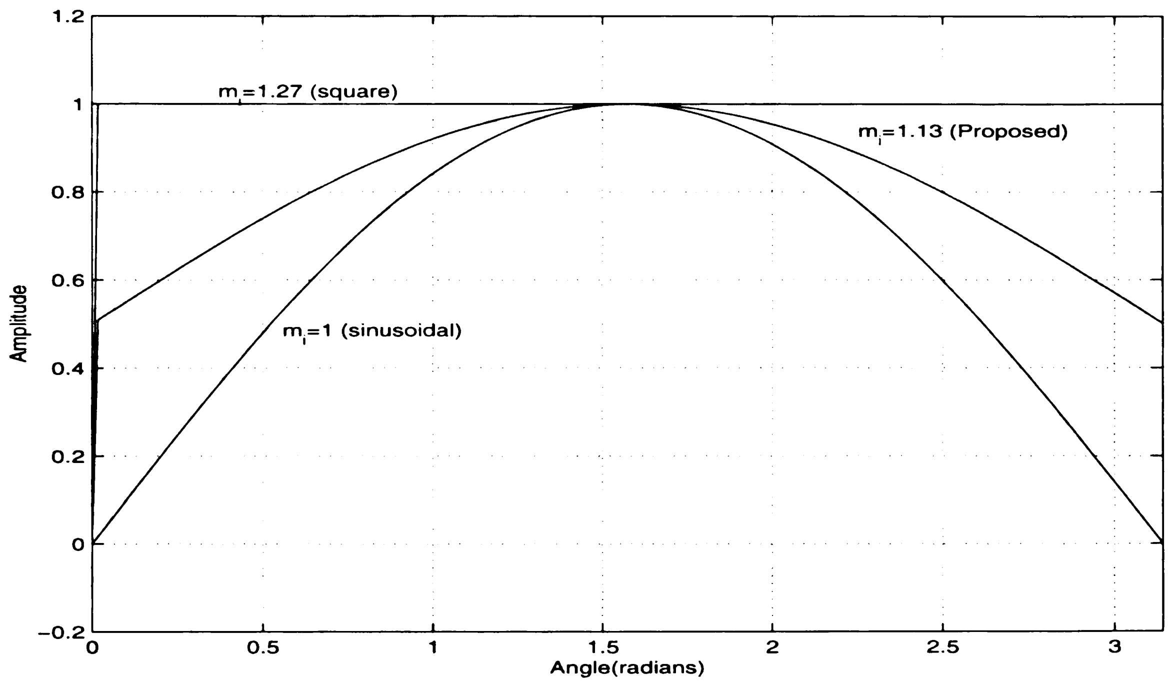


Figure 3.1. Representation of proposed method.

$$v_{ref}(1.13, 30°) = \frac{(1 - .5)}{1.27 - 1}(1.13 - 1) + .5 = 0.74, \tag{3.2}$$

where v_{ref} is the proposed reference signal and m_{isq} is the maximum modulation index that corresponds to the square wave (1.27). m_{isin} is the maximum modulation index in the linear range that corresponds to the unitary sinusoidal ($m_i = 1$). The value of $sin(\theta)$ corresponds to the instantaneous value of one of the three phase reference signals:

$$v_A = m_i \sin(\theta) \tag{3.3}$$

$$v_B = m_i \sin(\theta - 120°) \tag{3.4}$$

$$v_C = m_i \sin(\theta + 120°). \tag{3.5}$$

Since

not in p

from the

Therefor

taneous val

Square θ

function onl

as

or

Since v_a

equation (3.

The sam

Since the input signals are normally given in rectangular coordinates (v_d, v_q) and not in polar (m_i, θ), these three instantaneous voltage values are obtained directly from the input signals with a 2-to-3 transformation:

$$v_A = v_d \tag{3.6}$$
$$v_B = \frac{-1}{2}v_d + \frac{\sqrt{3}}{2}v_q \tag{3.7}$$
$$v_C = \frac{-1}{2}v_d - \frac{\sqrt{3}}{2}v_q. \tag{3.8}$$

Therefore, the value of the unitary sinusoidal can be calculated from the instantaneous values of the three phases

$$v_{au} = \sin(\theta) = \frac{v_A}{m_i}. \tag{3.9}$$

Square(θ) is the value of the unitary square at the desire angle θ. This square function only has two values, 1 and -1, and can be synthesized using the sign function as

$$Square(\theta) = sign(\pi - \theta) \tag{3.10}$$

or

$$Square(\theta) = sign(sin(\theta)) = sign(v_{au}). \tag{3.11}$$

Since v_{au} is available, the second equation is used. Substituting all those terms in equation (3.1) yields

$$v_{ref}(v_d, v_q) = \frac{(sign(v_{au}) - v_{au})}{.27}(m_i - 1) + v_{au}. \tag{3.12}$$

The same equation is obtained for the other three phases by substituting v_{au} by

v_{bi}, v_{ci}

Howev

embedded

tables to b

to obtain a

3.2 A

This techn

phase sim

These two

and

v_1 and

as will be

selected.

and

In o

v_{bu}, v_{cu}. The modulation index, m_i can be calculated from v_d, v_q as

$$m_i = \sqrt{v_d^2 + v_q^2}. \tag{3.13}$$

However, the square root is not a direct instruction in microprocessors used in embedded systems, requiring a Taylor expansion routine or the use of a look-up tables to be implemented. In order to retain the simplicity used so far, an equation to obtain an approximated value of m_i is proposed.

3.2 Approximation of the Modulation Index

This technique takes advantage of the fact that two reference signals of the three phase sinusoidal reference signal can be approximated by a straight line every 30°. These two phases are

$$v_1 = min(abs(v_A, v_B, v_C)) \tag{3.14}$$

and

$$v_2 = mid(abs(v_A, v_B, v_C)). \tag{3.15}$$

v_1 and v_2, are available (calculated for other purposes) in some PWM techniques, as will be shown later. In order to set the linear equation, the region $(0^o - 30^o)$ is selected. In this region

$$v_1 = v_A = m_i \sin(\theta) \tag{3.16}$$

and

$$v_2 = v_C = m_i \sin(\theta + 120°). \tag{3.17}$$

In order to find the linear approximation of each sinusoidal segment, their values at the lin

Using th

become

and

Solving f

an approxin

This equ

However, in

the approxi

reduce this

This can be

at the limits of the range are selected as

$$
\begin{aligned}
v_1(0^\circ) &= 0 && (3.18)\\
v_1(30^\circ) &= \frac{1}{2}m_i && (3.19)\\
v_2(0^\circ) &= \frac{\sqrt(3)}{2}m_i && (3.20)\\
v_2(30^\circ) &= \frac{1}{2}m_i. && (3.21)
\end{aligned}
$$

Using the slope and intersection to approximate these lines, the two equations become

$$v_1 = \frac{3}{\pi}m_i\theta \tag{3.22}$$

and

$$v_2 = \frac{3(1-\sqrt(3))}{\pi}m_i\theta + \frac{\sqrt{3}}{2}m_i. \tag{3.23}$$

Solving for θ in equation (3.22) and substituting it in equation (3.23), a value for an approximate modulation index is found:

$$
\begin{aligned}
m_i &= \frac{2(\sqrt(3)-1)}{\sqrt(3)}v_1 + \frac{2}{\sqrt{3}}v_2 && (3.24)\\
m_i &= k_1v_1 + k_2v_2 = 0.85v_1 + 1.15v_2 && (3.25)
\end{aligned}
$$

This equation coincides with the actual modulation index every 30°, (0°,30°,60°...). However, in the middle of each region (15°,45°,75°.....) it shows an error of 3.4%. Since the approximate value of m_i is always greater than the actual value, it is possible to reduce this error by half (1.7%) by redistributing it below and above the real value. This can be done by changing the values of k_1 and k_2 to produce

F

With th

is equal to

signals asso

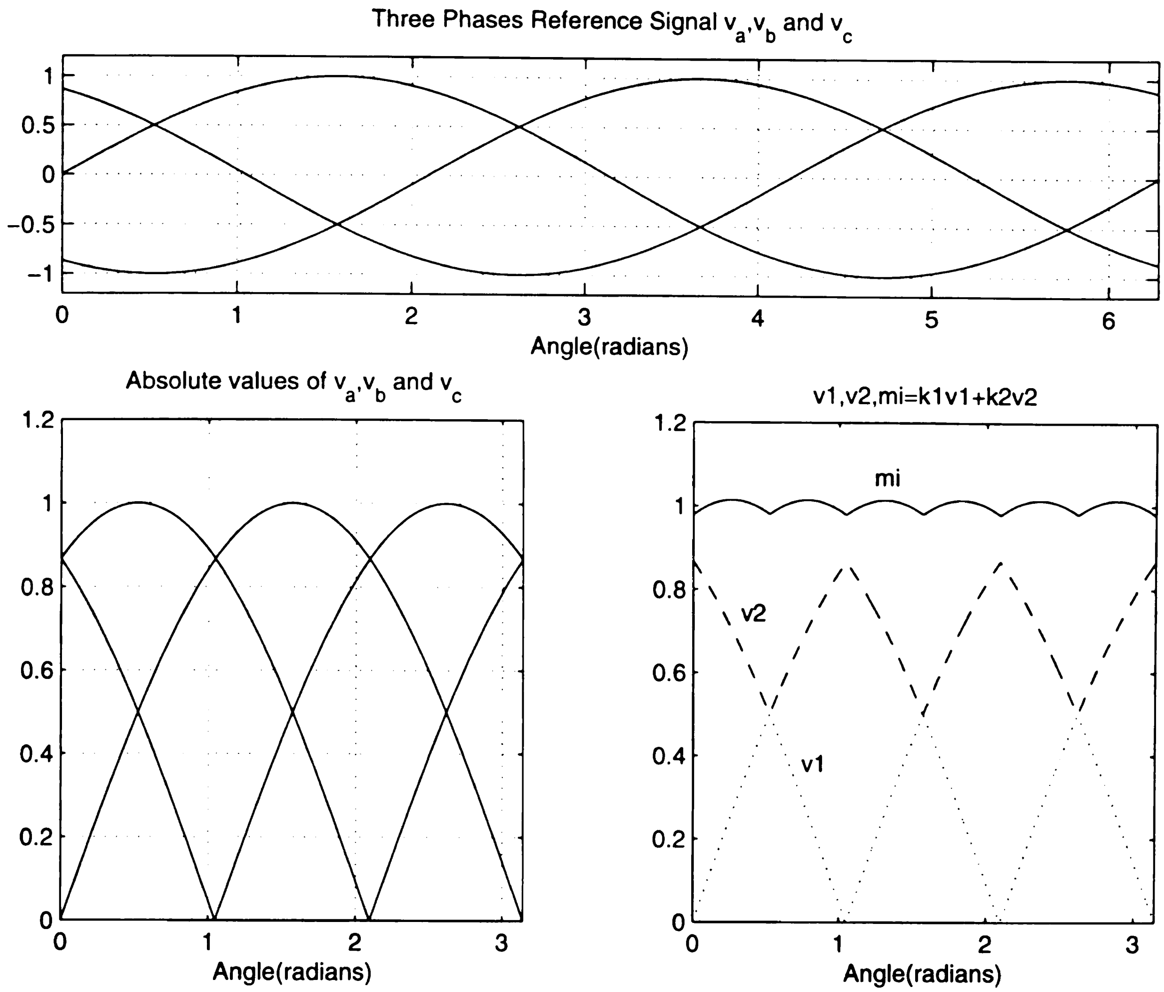


Figure 3.2. Waveforms for calculation of the modulation index .

$$m_i = 0.83v_i + 1.13v_2. \tag{3.26}$$

With these values of k_1 and k_2, the average of the approximated modulation index is equal to the actual modulation index. Figure 3.2 shows the three phases and the signals associated with the modulation index calculations.

3.3 I

The com

block dia

the first b

mum abso

additional

m_i. This n

output equa

modulation

using the ge

used for two

convert v_A in

index:

The nor

wave that w

modulation

signal.

3.4 P

The propor

a square wa

produce a r

allows exte

3.3 PRSG Using Unmodified Reference

The complete procedure to generate the proposed reference signal is shown in the block diagram in figure 3.3. The equations for the 2-to-3 transformation are inside the first block, which converts v_d and v_q into v_a, v_b and v_c. The minimum and maximum absolute values of the three phases are found using logic comparisons, requiring additional blocks. The modulation calculation block uses $v_{max}, v_{mid}, v_{min}$ to produce m_i. This modulation index command is passed through a limiter. That keeps its output equal to 1 for $m_i > 1$. This limiter output is subtracted from the original modulation index in order to produce an output of 0 for $m_i < 1$. This logic allows using the generator in the linear range as well. The output of the addition block is used for two purposes. First, it is added to the value of 1 to form a divisor, which will convert v_A into v_{AU}. Second, it is divided by 0.27 to create the normalized modulation index:

$$m_{in} = \begin{cases} 0 & \text{for } (0 < m_i \leq 1) \\ \frac{m_i - 1}{0.27} & \text{for}(1 < m_i \leq 1.27). \end{cases}$$

The normalized modulation index determines the amount of square or sinusoidal wave that will be present at the output. Finally, the wave mixer uses the normalized modulation index and the unitary sinusoidal value to produce the proposed reference signal.

3.4 PRSG Using Modified Reference

The proportional signal generator is able to convert a sinusoidal reference signal into a square wave. However, some PWM techniques (mentioned in the previous section) produce a modified sinusoidal reference (SVPWM). This modified sinusoidal reference allows extending the linear range (no distortion at the load) to 1.15. This modified

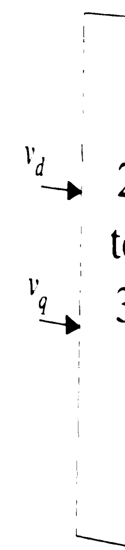


Figure 3.3. I

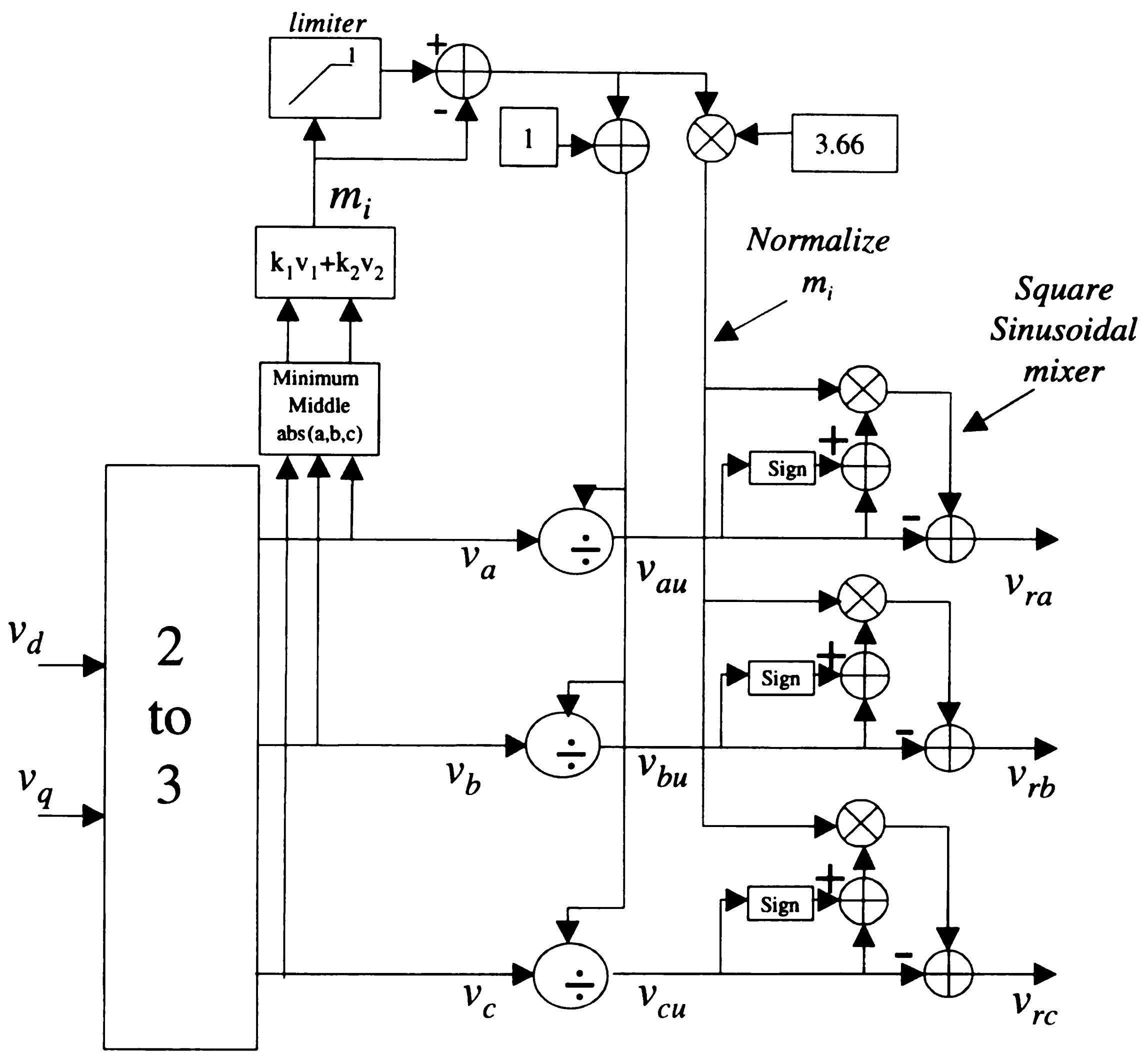


Figure 3.3. Proportional Reference Signal Generator (unmodified reference case).

sinusoi

values

with th

V_{com}) a

In

for mo

is use

neous

the fo

F

The

I

refer

dam

The

lari

inda

mo

mo

sinusoidal is obtained by adding an offset to each of the instantaneous command values of the three phases. This offset is equal to half the voltage value of the phase with the minimum absolute value. The modified sinusoidal references (V_{aum}, V_{bum} and V_{cum}) are defined as

$$V_{aum} = V_{au} + \frac{v_1}{2} \tag{3.27}$$

$$V_{bum} = V_{bu} + \frac{v_1}{2} \tag{3.28}$$

$$V_{cum} = V_{cu} + \frac{v_1}{2}. \tag{3.29}$$

In the process of finding v_1, v_2 is also obtained. Therefore the proposed technique for modulation index approximation does not introduce extra calculations when it is used in conjunction with the modified sinusoidal reference signal. The instantaneous unitary modified sinusoidal reference signals replace the sinusoidal signal, in the formula of the proposed reference signal, equation 3.12 to produce

$$v_{ref}(v_d, v_q) = \frac{(sign(V_{aum}) - V_{aum})}{0.12}(m_i - 1) + V_{aum}. \tag{3.30}$$

Figure 3.4 shows the proportional signal generator for the modified sinusoidal case. The reference signals for the three phases are generated from inputs v_d and v_q.

It will be shown in next chapter by, mathematical analysis, that the proportional reference signal generator possesses a perfect linearity. This means that the fundamental component of the new reference signal is equal to the modulation index. The THD of the proposed method varies also linearly with respect to the modulation index. In the unmodified reference case, THD increases as the modulation index increases beyond 1. In the modified sinusoidal case the distortion starts at the modulation index of 1.15, which means lower THD path. If sinusoidal pulse width modulation is used in the linear range, the first version (PRSG-Sinusoidal) is better

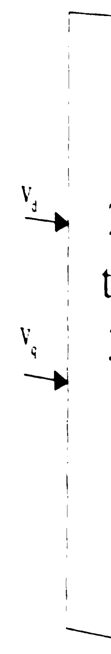


Figure 3.

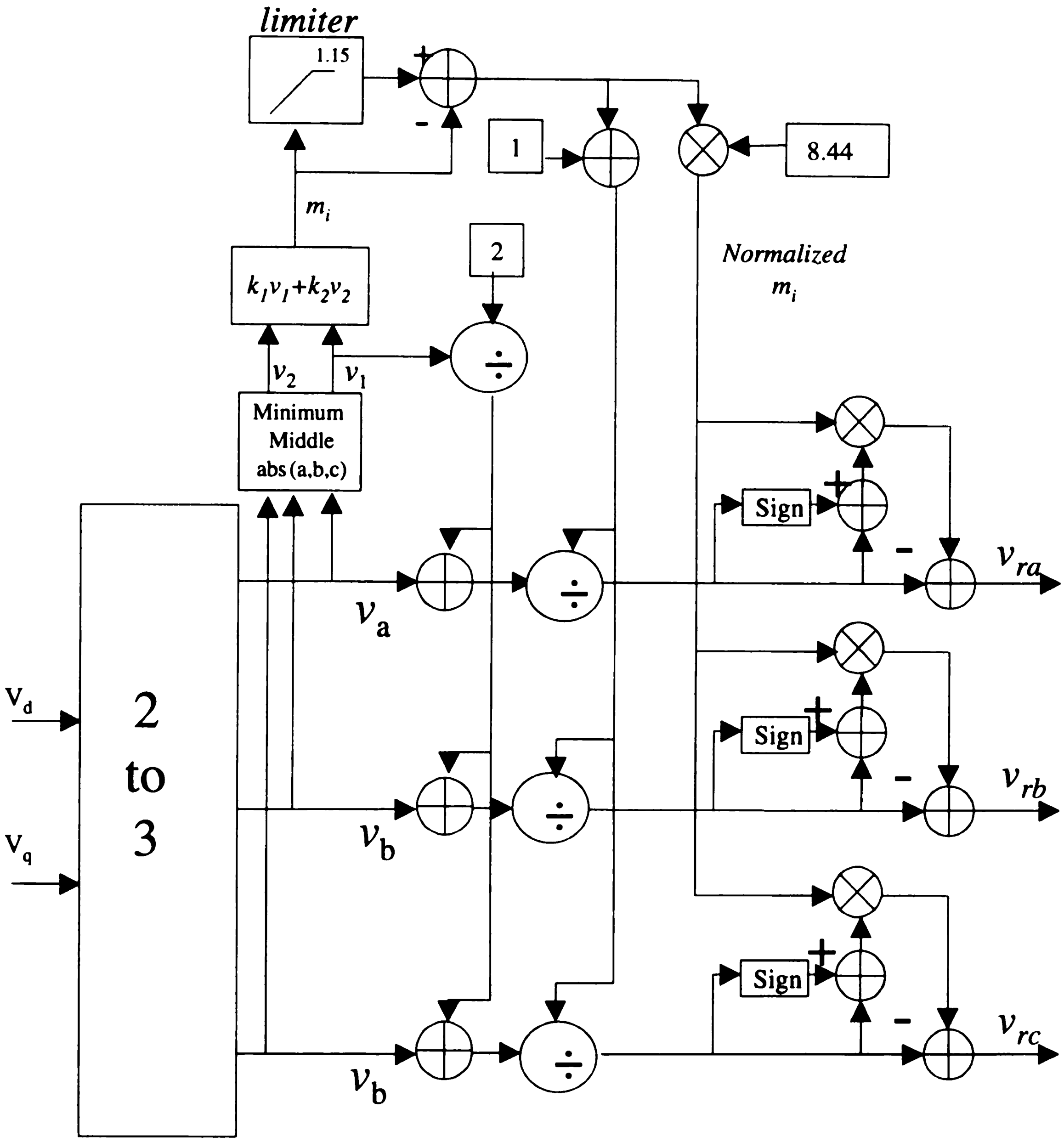
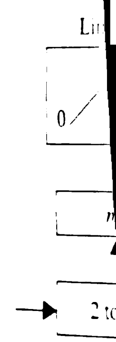


Figure 3.4. Proportional Reference Signal Generator (modified reference case).



Fig

than the se

is used. the

sinusoidal c

3.5 T

The linear

two-step tr

erence into

waveform.

are needed

trapezoida

function.

and two a

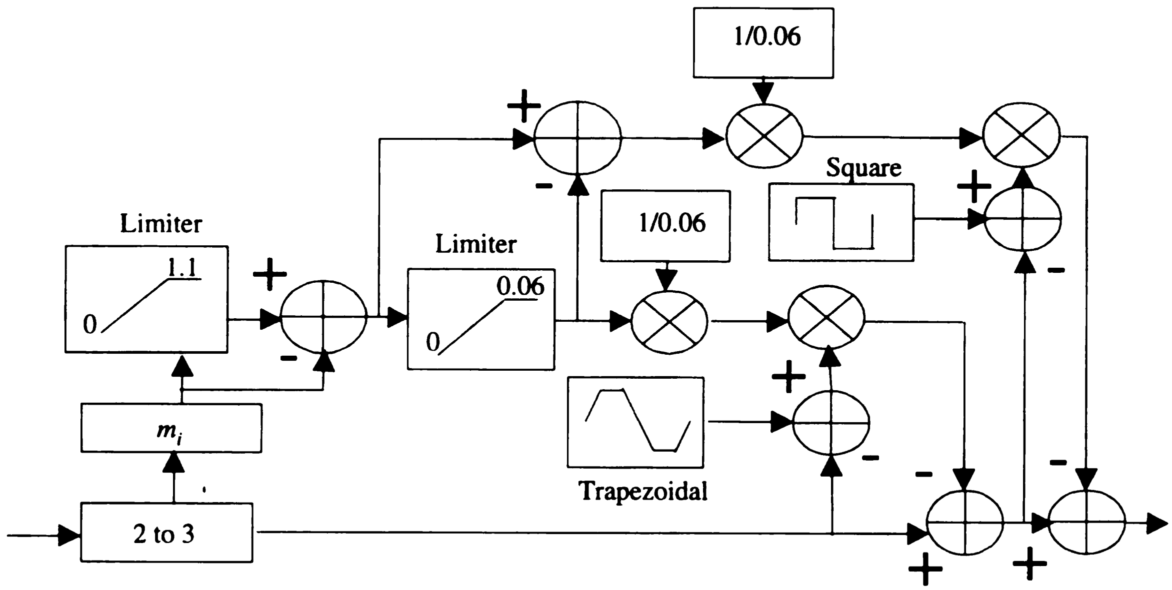


Figure 3.5. Proportional Reference Signal Generator (two-step case).

than the second one. When modified sinusoidal Pulse Width Modulation (SVPWM) is used, the second version is preferable, since the THD distortion is lower than the sinusoidal case, and v_1 and v_2 are already calculated.

3.5 Two-Step Transformation

The linear behavior of the total harmonic distortion can be improved even further if a two-step transformation is used. That means converting the modified sinusoidal reference into a trapezoidal waveform and then converting the trapezoidal into a square waveform. A one-phase version of this method is shown in figure 3.5. Two mixers are needed in this approach. The first mixes the unitary modified sinusoidal with a trapezoidal wave. The trapezoidal wave can be synthesized using the absolute value function. The second mixes the unitary trapezoidal with a square wave. Two limiters and two adders constitute the logic to decide between the two transformations.

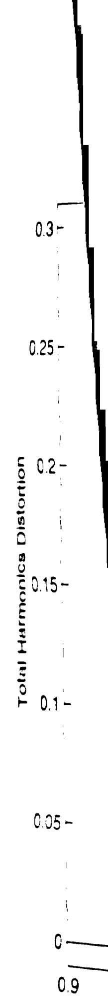


Figure 3

In the

mental of

from 1.21

harmonic

tal. The

the harm

describe a

The t

ure 3.6.

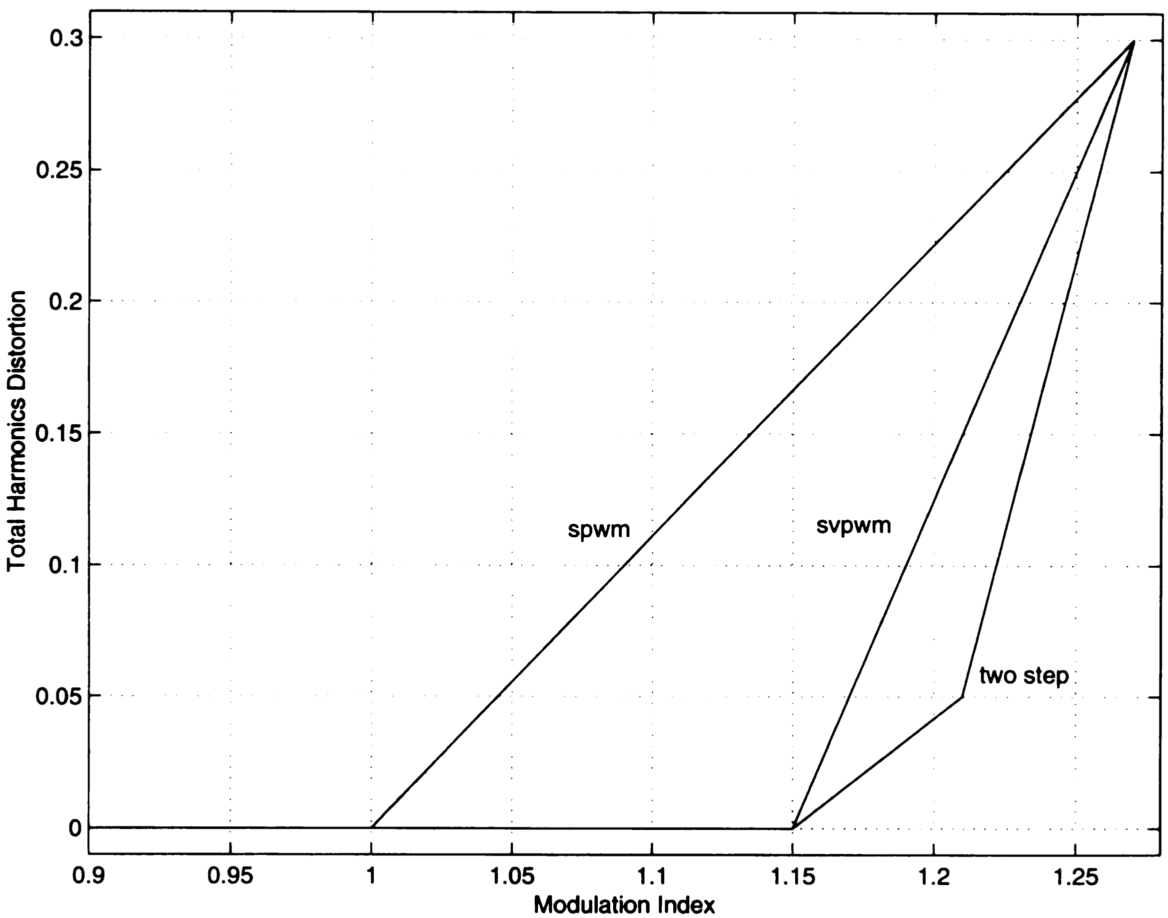


Figure 3.6. Total harmonic distortion for the three proposed versions of PRSG.

In the first step, the modulation index changes from 1.15 to 1.21 since the fundamental of the trapezoidal is 1.21. In the second step the modulation index changes from 1.21 to 1.27. It will be shown later that this two-step transformation reduces the harmonic content, while preserving the linearity of modulation index vs. fundamental. The THD also varies linearly within each one of the sub-ranges. However, since the harmonic distortion of the trapezoidal wave is only 5%, these two straight lines describe a better THD path than a single line that covers the entire range (1.15-1.27).

The total harmonic distortion of the three proposed versions are shown in Figure 3.6. In the simulation chapter, the steps followed to develop this scheme are discussed.

Mat

In this chap

ity is teste

for the mod

(sinusoidal-

tal changes

constants k

4.1 Li

4.1.1 L

When the

sinusoidal F

with a squa

that determ

CHAPTER 4

Mathematical Analysis

In this chapter, the proposed method is mathematically investigated. First, linearity is tested for the sinusoidal case. Second, the same linearity test is performed for the modified sinusoidal case. Third, linearity is examined for general case mixer (sinusoidal-trapezoid and trapezoid-square), which shows that not only the fundamental changes linearly but also the harmonic distortion. Finally, an optimization of the constants k_1 and k_2, which are used for modulation index calculation, is performed.

4.1 Linearity of the PRSG method

4.1.1 Linearity of the PRSG-Unmodified

When the proposed method (PRSG), is used to extend the linearity from the sinusoidal PWM to six-step mode, the unitary sinusoidal reference signal is mixed with a square wave at a ratio determined by the modulation index. The equation that determines the new reference signal is

$$v_{ref}(m_i) = \frac{(sign(V_{AU}) - V_{AU})}{.27}(m_i - 1) + V_{AU}, \tag{4.1}$$

where V

modulati

by its ex

Since t

to m_t. Th

component

Since V

rearranged

sign(V_{AC})

-1 for π <

However, s

the same s

In this

constant th

where V_{AC}

where V_{AU} is the unitary sinusoidal reference signal, $\sin\theta$, and m_i is the reference modulation index. In order to obtain an exact solution, the value of .27 is substituted by its exact value $(4/\pi - 1)$

$$v_{ref}(m_i) = \frac{\pi(m_i - 1)}{4 - \pi}(sign(V_{AU}) - V_{AU}) + V_{AU}. \tag{4.2}$$

Since the reference signal is normalized, the desired output fundamental is equal to m_i. Therefore, a linearity proof for this method consists of showing that the first component of the Fourier series of v_{ref} is equal to the modulation index command

$$F_1 = \frac{1}{\pi}\int_0^{2\pi} v_{ref}\sin\theta\, d\theta = m_i. \tag{4.3}$$

Since V_{AU} is a continuous function of θ while m_i and $sign(\sin\theta)$ are not, v_{ref} is rearranged as

$$v_{ref} = v_{ref\theta} + v_{refk} = \frac{4 - \pi m_i}{4 - \pi}V_{AU} + \frac{\pi(m_i - 1)}{4 - \pi}sign(V_{AU}). \tag{4.4}$$

$sign(V_{AU})$ is a discontinuous function, the values of which are 1 for $0 < \theta \leq \pi$ and -1 for $\pi < \theta \leq 2\pi$. This suggests to divide the integral in these two sub-ranges. However, since $sign(\sin\theta)$ and $\sin\theta$ have half cycle symmetry functions, v_{ref} possess the same symmetry as well, therefore only the first sub-range needs to be integrated.

$$F_1 = \frac{1}{\pi}\int_0^{2\pi} v_{ref}\sin\theta\, d\theta = \frac{2}{\pi}\int_0^{\pi} v_{ref}\sin\theta\, d\theta. \tag{4.5}$$

In this range sign($\sin\theta$) always has a value of 1, which means v_{refk} becomes a constant that depends on m_i. Solving the integral:

$$F_1 = \frac{(4 - \pi\, m_i)}{(4 - \pi)}\,\frac{2}{\pi}\int_0^{\pi}\sin^2\theta\, d\theta + \frac{\pi(m_i - 1)}{(4 - \pi)}\,\frac{2}{\pi}\int_0^{\pi}\sin\theta\, d\theta, \tag{4.6}$$

where V_{AU} is substituted by $\sin\theta$. Solving both integrals we obtain:

which is e

the referen

PRSG (sin

4.1.2 I

In order to

for the mo

of a sinuso

phase with

A ($m_i \sin\theta$

the offset i

For the

fore, the m

Because

sary to con

sinusoidal s

$$F_1 = \frac{(4-\pi m_i)}{(4-\pi)} + \frac{\pi(m_i-1)}{(4-\pi)}\frac{4}{\pi} \tag{4.7}$$

$$F_1 = \frac{(4-\pi m_i + 4m_i - 4)}{4-\pi} = m_i. \tag{4.8}$$

which is exactly what was stated in equation (4.3). The fundamental amplitude of the reference signal is equal to the modulation index command, which shows that PRSG (sinusoidal case) posses a perfect linearity.

4.1.2 Linearity of the PRSG-Modified

In order to test linearity for the PRSG (SVM case), it is necessary to find an expression for the modified sinusoidal reference signal. This signal can be considered as the sum of a sinusoidal signal and an offset. This offset corresponds to the half value of the phase with the lowest absolute value. In the first 30° this phase corresponds to phase A ($m_i \sin\theta$). Using phase A as reference for the modified sinusoidal, the equation for the offset in the first 30°($i.e.\, \pi/6$) is obtained:

$$V_{AM} = m_i \sin\theta + \frac{1}{2} m_i \sin\theta = m_i \frac{3}{2} \sin\theta \tag{4.9}$$

For the next 60° the offset corresponds to phase B (i.e. $m_i \sin(\theta - 2\pi/3)$). Therefore, the modified sinusoidal equation is:

$$V_{AM} = m_i \sin\theta + \frac{1}{2} m_i \sin(\theta + 2\pi/3) = m_i \frac{\sqrt{3}}{2} \sin(\theta + \pi/6). \tag{4.10}$$

Because of the one fourth cycle symmetry of this reference signal, it is not necessary to continue synthesizing it beyond 90°. For the unitary version of the modified sinusoidal signal $m_i = 2/\sqrt{3}$, the equations for this signal become:

Subst

the expres

$F_1 =$

This ec

parts. Fir

is extende

Second, si

to $4/\pi$.

function i

and finall

$$V_{AUM} = \begin{cases} \sqrt{3}\sin\theta & \text{for } 0 \leq \theta < \frac{\pi}{6} \\ \sin(\theta + \pi/6) & \text{for } \frac{\pi}{6} \leq \theta < \frac{\pi}{2} \end{cases} .$$

Substituting V_{AUM} in equation (4.4) and then substituting v_{ref} in equation (4.5) the expression for the fundamental is derived:

$$F_1 = \frac{\sqrt{3}(4 - \pi\, m_i)}{(4\sqrt{3} - 2\pi)} \frac{4}{\pi} \left(\sqrt{3} \int_0^{\pi/6} \sin^2\theta \, d\theta + \int_{\pi/6}^{\pi/2} \sin(\theta + \frac{\pi}{6}) \sin\theta \, d\theta \right) + \frac{\pi(\sqrt{3}m_i - 2)}{(4\sqrt{3} - 2\pi)} \frac{4}{\pi} \int_0^{\pi/2} \sin\theta \, d\theta \qquad (4.11)$$

This equation, compared to the one of the sinusoidal case (4.6), differs in three parts. First, the constant expression outside the integral changes, since m_i in SVM is extended from 1 to $2/\sqrt{3}$ and the factor $4/(4 - \pi)$ becomes $\sqrt{3}\pi/(4\sqrt{3} - 2\pi)$. Second, since only one fourth of the cycle is integrated, the constant $2/\pi$ is changed to $4/\pi$. Third, the integral of V_{AUM}, is divided in two parts, since this modified function is discontinues as was defined above. These three integrals become:

$$\sqrt{3} \int_0^{\pi/6} \sin^2\theta \, d\theta = \frac{2\sqrt{3}\pi - 9}{24} \qquad (4.12)$$

$$\int_{\pi/6}^{\pi/2} \sin(\theta + \frac{\pi}{6}) \sin\theta \, d\theta = \frac{2\sqrt{3}\pi + 9}{24} \qquad (4.13)$$

$$\int_0^{\pi/2} \sin\theta \, d\theta = 1, \qquad (4.14)$$

and finally

$$F_1 = \frac{\sqrt{3}(4-\pi\, m_i)}{(4\sqrt{3}-2\pi)}\frac{4}{\pi}\left(\frac{2\sqrt{3}\pi-9}{24}+\frac{2\sqrt{3}\pi+9}{24}\right)+\frac{\pi(\sqrt{3}m_i-2)}{(4\sqrt{3}-2\pi)}\frac{4}{\pi} \quad (4.15)$$

$$F_1 = \frac{2(4-\pi\, m_i)}{(4\sqrt{3}-2\pi)}+\frac{4(\sqrt{3}m_i-2)}{(4\sqrt{3}-2\pi)} = m_i. \quad (4.16)$$

This proves that the fundamental amplitude of the new reference signal generated with the PRSG has a linear behavior with respect to the modulation index command for the space-vector case as well. It is possible to continue the Fourier analysis for other mixer cases (sinusoidal-trapezoid and trapezoid-square). However, the results of the two previous cases suggest that there should exist a general case proof that justifies the linear behavior of the PRSG method, based on the orthogonality of the Fourier components.

4.1.3 Linearity of the PRSG-General Case

In order to prove that the PRSG has a linear behavior for all cases, including harmonic distortion, a general case test is performed for the PRSG. The heart of the PRSG is the output mixer which takes as inputs two different waveform signals. In some cases, the second signal is generated based in the first one. It also takes as inputs the normalized modulation index, which determines the amount of each input reference signal that will be at the output, and has values between 0 and 1. The general equation for the PRSG mixer is

$$v_{ref}(m_i) = \frac{(v_2 - v_1)}{(m_{imax} - m_{ilim})}(m_i - m_{ilim}) + v_1. \quad (4.17)$$

This linear mixer transforms the output from v_1 for $m_i = m_{ilim}$ to v_2 for $m_i = m_{imax}$, where m_{imax} and m_{ilim} correspond to the normalized fundamental amplitudes of v_2 and v_1 respectively. Defining the normalized modulation index as

$$k = m_{inorm} = \frac{(m_i - m_{ilim})}{m_{imax} - m_{ilim}}, \tag{4.18}$$

the equation for the mixer can be expressed in a linear way

$$v_{out} = kv_1 + (1-k)v_2. \tag{4.19}$$

v_1 and v_2 are odd functions and their Fourier series are

$$v_1 = \sum_{n=1,3,5,..}^{\infty} a_{1n}\sin(n\theta) \qquad v_2 = \sum_{n=1,3,5..}^{\infty} a_{2n}\sin(n\theta). \tag{4.20}$$

Substituting these expressions in the previous equation:

$$\begin{aligned} v_{out} &= k\sum_{n=1,3,5,..}^{\infty} a_{1n}\sin(n\theta) + (1-k)\sum_{n=1,3,5..}^{\infty} a_{2n}\sin(n\theta) \\ v_{out} &= \sum_{n=1,3,5,..}^{\infty} (ka_{1n} + (1-k)a_{2n})\sin(n\theta). \end{aligned} \tag{4.21}$$

Therefore, V_{out} appears here as another Fourier serie, in which each component, n, is a linear function of the corresponding n components of V_1 and V_2. It can be concluded that, because of the linearity of the mixer and the orthogonality of the Fourier components, each harmonic is mixed separately. Therefore, the output fundamental only depends on the fundamental of both input signals and the variable k. If each input is substituted by its fundamental ($v_1 = m_{ilim}$, $v_2 = m_{imax}$, and $v_{out} = F_1$) in the mixer equation (4.17), the linearity proof is trivial:

$$F_1 = \frac{(m_{imax} - m_{ilim})}{(m_{imax} - m_{ilim})}(m_i - m_{ilim}) + m_{ilim} = m_i \tag{4.22}$$

4.2 Harmonic Distortion of the PRSG Method

From the previous analysis and the preliminary Matlab simulations, it appears that the harmonic distortion changes linearly as the modulation index is changed in the overmodulation ranges. However, the following analysis will show that this is not true. The linear behavior of THD in PRSG is due to some special conditions and not to the separate mixing of its components.

The THD is defined as the Euclidean norm of all the components excluding the fundamental. Al the components are divided also by the fundamental. However it can be omitted here without affecting the linearity test. The THD of the two inputs and the output of the PRSG mixer are defined as:

$$THD(v_1) = \sqrt{\sum_{n=5,7,11,13}^{\infty} a_{n1}^2} \tag{4.23}$$

$$THD(v_2) = \sqrt{\sum_{n=5,7,11,13}^{\infty} a_{n1}^2} \tag{4.24}$$

$$THD(v_{out}) = \sqrt{\sum_{n=5,7,11,13}^{\infty} ((1-k)a_{n1} + ka_{n2})^2} \tag{4.25}$$

The THD of the resulting v_{out} is equal to the THD of v_1 (THD_1) for k=0 and is equal to the THD of v_2 (THD_2) for k=1. However, linear behavior also requires that the THD of v_{out} varies linearly between THD_1 and THD_2 as k changes from 0 to 1. The equation for linear THD should be:

$$THD(v_{out}) = (1-k)THD_1 + kTHD2 \tag{4.26}$$

$$\begin{aligned} THD(V_{out}) &= \Bigg[(1-k)\sum_{n=5,7,\ldots}^{\infty} a_{n1}^2 \\ &+ 2k(1-k)\sqrt{\sum_{n=5,7\ldots}^{\infty} a_{n1}^2}\sqrt{\sum_{n=5,7\ldots}^{\infty} a_{n2}^2} + k\sum_{n=5,7\ldots}^{\infty} a_{n2}^2\Bigg]^{\frac{1}{2}}. \end{aligned} \tag{4.27}$$

Doing the same manipulation over the actual THD equation (4.25), it is obtained:

$$THD(v_{out}) = \sqrt{\sum_{n=5,7..}^{\infty} (1-k)^2 a_{n1}^2 + 2k(1-k)a_{n1}a_{n2} + k^2 a_{n2}^2} \tag{4.28}$$

$$THD(v_{out}) = \sqrt{(1-k)^2 \sum_{n=5,7..}^{\infty} a_{n1}^2 + 2k(1-k) \sum_{n=5,7..}^{\infty} a_{n1}a_{n2} + k^2 \sum_{n=5,7..}^{\infty} a_{n2}^2}. \tag{4.29}$$

The equations for the linear THD (4.27) differs from that of the actual THD (4.29) in the middle terms of the expression inside of the square root, which shows that the THD is not linear. Fortunately, it is possible to show that the actual THD is upper bounded by the linear THD.

$$\sum_{n=5,7..}^{\infty} a_{n1}a_{n2} \leq \sum_{n=5,7..}^{\infty} a_{n1} \sum_{n=5,7..}^{\infty} a_{n2} \leq \sqrt{\sum_{n=5,7..}^{\infty} a_{n1}^2} \sqrt{\sum_{n=5,7..}^{\infty} a_{n2}^2} \tag{4.30}$$

It is important to remember here that the optimal THD is not the one that behaves linearly, but the one that has the lowest possible value for each modulation index command. Although it is possible to find vectors for which the two equation analyzed above differ, this will not be the case when the PRSG mixer is used. The reason why THD results (obtained with PRSG) appears so linear is because one of the two input waveforms does not contain harmonics and the THD equation is reduced to

$$THD(V_{out}) = \sqrt{k^2 \sum_{n=5,7..}^{\infty} a_{n2}^2} = kTHD_2. \tag{4.31}$$

The modified sinusoidal waveform contains triplem harmonics, but these harmonics are not considered in this calculation, since as mentioned before, they do not affect the output. The only transformation in which there are some harmonics in the input signal is in the trapezoid-square wave. Even in this case, the difference between the THD curves is negligible. Although the linear behavior of the PRSG THD is not optimal, it helps to predict the THD, which also helps to design the optimal two-step transformation, as it is shown in the next chapter.

4.3 Optimization of Constants for Modulation Index Calculation

In the previous chapter, a new way to calculate the modulation index from the v_d and v_q input signals was described. This method uses the information of the instantaneous values of v_a, v_b *and* v_c, to calculate the modulation index, using a linear equation that involves only the constant values k_1 and k_2.

Since there are two constants, and in order to obtain the modulation index equal to the desired one, we established one constraint, a degree of freedom is obtained. This situation can be exploited to minimize the average square error obtained with this method. In order to obtain the first relation between k_1 and k_2, we add an equation, so that the average error between actual m_i and calculated m_i is made zero:

$$\int_0^{\pi/6} (m_i - k_1 v_1 - k_2 v_2) d\theta = 0 \tag{4.32}$$

$$\int_0^{\pi/6} \left(1 - k_1 \sin\theta - k_2 \sin\left(\theta - \frac{2\pi}{3}\right)\right) d\theta = 0. \tag{4.33}$$

In this equation, the integration range covers the first 30^o, since the calculated modulation index curve is repeated every 30^o. In this range, v_1 corresponds to phase A ($sin\theta$) and v_2 corresponds to phase C ($sin(\theta - 2\pi/3)$). To simplify the integration, the modulation index m_i is made equal to 1. Evaluating (4.33):

$$\theta + k_1 \cos\theta + k_2 \cos(\theta - \frac{2\pi}{3})\Big]_0^{\pi/6} = 0 \tag{4.34}$$

$$k_1 \left(\frac{2 - \sqrt{3}}{2}\right) + k_2 \left(\frac{1 + \sqrt{3}}{2}\right) = \frac{\pi}{6} \tag{4.35}$$

$$k_1(2-\sqrt{3}) + k_2(1+\sqrt{3}) \quad = \quad \frac{\pi}{3}. \tag{4.36}$$

In order to get a second equation that relates k_1 and k_2, the integral of the square error over the same angle range ($0^o - 30^o$) is minimized.

$$\min(e_{as}) = \min\left(\int_0^{\pi/6}\left(1 - k_1\sin\theta + k_2\sin\left(\theta+\frac{2\pi}{3}\right)\right)^2 d\theta.\right) \tag{4.37}$$

The expression inside the integral becomes:

$$e_{as} = \int_0^{\pi/6}\left(1 + k_1^2\sin^2\theta + k_2^2\sin^2\left(\theta+\frac{2\pi}{3}\right) - 2k_1\sin(\theta) - 2k_2\sin\left(\theta+\frac{2\pi}{3}\right)\right.$$
$$\left. +2k_1k_2\sin\theta\sin\left(\theta+\frac{2\pi}{2}\right)\right) d\theta. \tag{4.38}$$

Calculating each part of this integral and substituting in equation (4.37)

$$e_{as} = \frac{\pi}{6} + \frac{2\pi - 3\sqrt{3}}{24}k_1^2 + \frac{\pi}{12}k_2^2 + \left((\sqrt{3}-2)k_1 + (1-\sqrt{3})k_2\right) + \frac{3\sqrt{3}-\pi}{12}k_1k_2. \tag{4.39}$$

The expression in parentheses is equal to that in the first equation (4.36), and can be substituted by $-\pi/3$. The last equation becomes

$$e_{as} = -\frac{\pi}{6} + \frac{2\pi - 3\sqrt{3}}{24}k_1^2 + \frac{\pi}{12}k_2^2 + \frac{3\sqrt{3}-\pi}{12}k_1k_2. \tag{4.40}$$

Using equation 4.36and expression for k_2 in terms of k_1 and use it in the 4.40 equation to obtain an expression in terms of k_1. To simplify we introduce constants M, N, Q, R, S:

$$e_{as} \quad = \quad -M + Nk_1^2 + \frac{M}{2}k_2^2 + Qk_1k_2 \tag{4.41}$$

$$k_2 \quad = \quad R - Sk_1, \tag{4.42}$$

where these variables correspond to

$$M = \frac{\pi}{6} \quad N = \frac{2\pi - 3\sqrt{3}}{24} \quad Q = \frac{3\sqrt{3} - \pi}{12} \tag{4.43}$$

$$R = \frac{\pi}{3(\sqrt{3} - 1)} \quad S = \frac{2 - \sqrt{3}}{\sqrt{3} - 1}. \tag{4.44}$$

Substituting equation (4.42) in equation (4.41),

$$e_{as} = -M + Nk_1^2 + \frac{M}{2}R^2 - MRSk_1 + \frac{M}{2}S^2k_1^2 + Qk_1R - QSk_1^2. \tag{4.45}$$

This equation, in terms of k_1, can be rewritten as

$$e_{as} = ak_1^2 + bk_1 + c \tag{4.46}$$

where a, b, c are

$$a = (N + \frac{M}{2}S^2 - QS) \quad b = (Q - MS)R \quad c = -M + \frac{M}{2}R^2 \tag{4.47}$$

In order to minimize the error, its derivative is made equal to 0:

$$\frac{de_{as}}{dk_1} = 2ak_1 + b = 0 \tag{4.48}$$

and finally the values of k_1 and k_2 are found:

$$k_1 = \frac{-b}{2a} = 1.1282 \tag{4.49}$$

$$k_2 = R - Sk_1 = .8245 \tag{4.50}$$

Those values are very similar to the ones proposed in the description chapter (1.13 and 0.83) and were found in a very different way. Those original values were found using Matlab simulations, where k_1 and k_2 were changed in steps of .001 and the minimum error was evaluated. This analytical expression fully verifies the previous results.

CHAPTER 5

Simulation

This chapter shows the test results obtained from the simulation of the proposed method. Block diagrams, state graphs, electric circuits, and equations where used to model the different components of the new method. Simulations for the modified and unmodified cases were performed using block diagrams and equations. Each one of this cases was tested with actual and predicted modulation index, in order to evaluate the effectiveness of the proposed modulation index calculation method. Then, the two-state case and Mode I and Mode II were simulated separately using a state graph design. Finally the proposed method (PRSG-TWO-STEP) and the best of the existing methods (Mode I and II) are compared.

5.1 Proportional Reference Signal Generator (PRSG)

5.1.1 PRSG Components

2-to-3 transformation

One of the advantages of simulations is that they can lead to simplification of the actual implementation. One example of this is the classical 2-to-3 transformation,

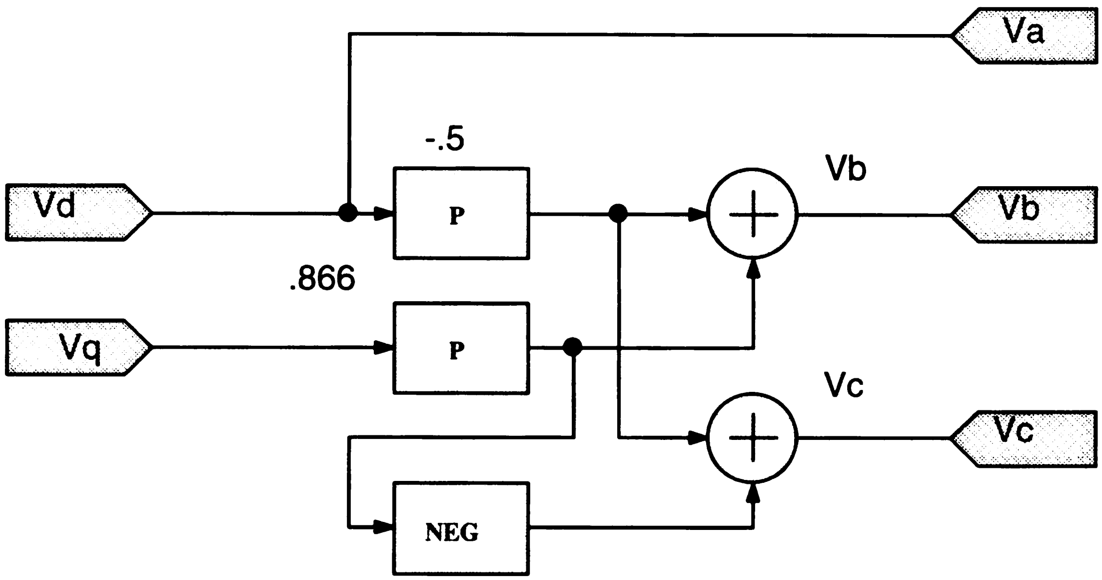


Figure 5.1. 3-to-2 transformation.

which converts the reference vector described by v_d and v_q to the three phase reference signals v_a,v_b,v_c. In this transformation it is required to multiply a 2 by 3 matrix by v_d and v_q. However, seeing the matrix elements in detail, we notice that only one multiplication, a division by two, two additions, and a negation are needed to implement the conversion. The following block diagrams show this simplification.

Phase Ordering block

In order to calculate the offset that it is needed to generate the reference signal for space vector modulation and for the predicted modulation index, it is necessary to know the maximum, middle, and minimum instantaneous values of the reference voltage of the three phases. In the experimental implementation (Intel 80C196 Microcontroller) an algorithm to order the three phases value is used. However, the Simplorer simulation package provides built-in blocks, which calculate maximum and

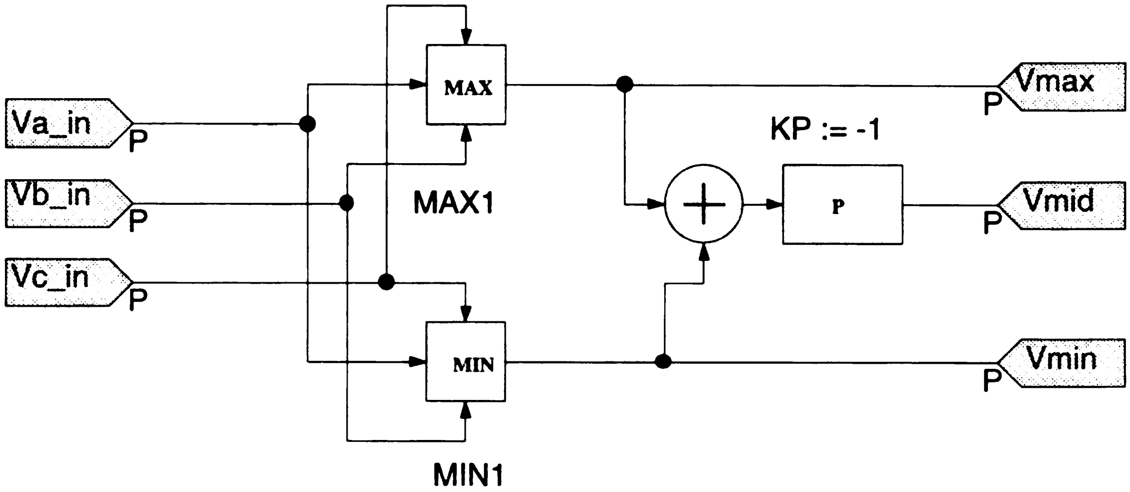


Figure 5.2. Phase ordering block diagram.

minimum values of the three given inputs. These two blocks are used here to simplify this ordering procedure. Since the three instantaneous values add to 0, the middle value can be calculated from the other two (maximum and minimum). In Figure 5.2 the phase ordering blocks are shown.

Offset Calculation

As was explained previously, the offset needed to produce the SVM reference signal can be calculated from the maximum and minimum phase values. When this idea was first proposed, the offset used was

$$Offset = \frac{(T_{max} - T_{min})}{2}. \tag{5.1}$$

Where T_{max} and T_{min} are the equivalent times of the maximum and minimum values of the three reference signals v_a,v_b,v_c. However, if this offset is calculated as voltage, before converting to time, it is equal to

$$Offset = \frac{-(v_{max} + v_{min})}{2} = \frac{v_{mid}}{2}. \tag{5.2}$$

Therefore, it is the same to find v_{max} and v_{min} or to find directly v_{mid}. In terms of algorithms, both procedures have the same difficulty. If the absolute values are used, the offset becomes

$$|Offset| = \frac{v_{mid}}{2} = \frac{\min(|v_a|, |v_b|, |v_c|)}{2}). \tag{5.3}$$

This last equation appears easier to apply, since only a minimum value is needed. However, only the magnitude of the offset is known with this method. In order to know the sign, we use the index of this minimum value, which makes the procedure as complicated as the previous two. An alternative method to calculate the modulation index without using absolute values will be used.

Modulation Index Calculation

Since the sign of the reference signal is lost when the absolute values are calculated, a method to find v_1 and v_2 from the three original (signed) reference signals is proposed here. Assume v_1 is the minimum absolute value of the three reference signals, which it is always the middle value of the signed reference signals. Assume v_2 is the middle value of the absolute reference signals, then v_2 is equal to v_{max} if v_1 is positive and it is equal to v_{min} if v_1 is negative. Hence, the final procedure to be implemented is reduced to the following algorithm

$$v_1 = -(v_{max} + v_{min}) \tag{5.4}$$

$$offset = v_1/2 \tag{5.5}$$

$$\text{if } v_1 > 0 \;\; v_2 = v_{max} \tag{5.6}$$

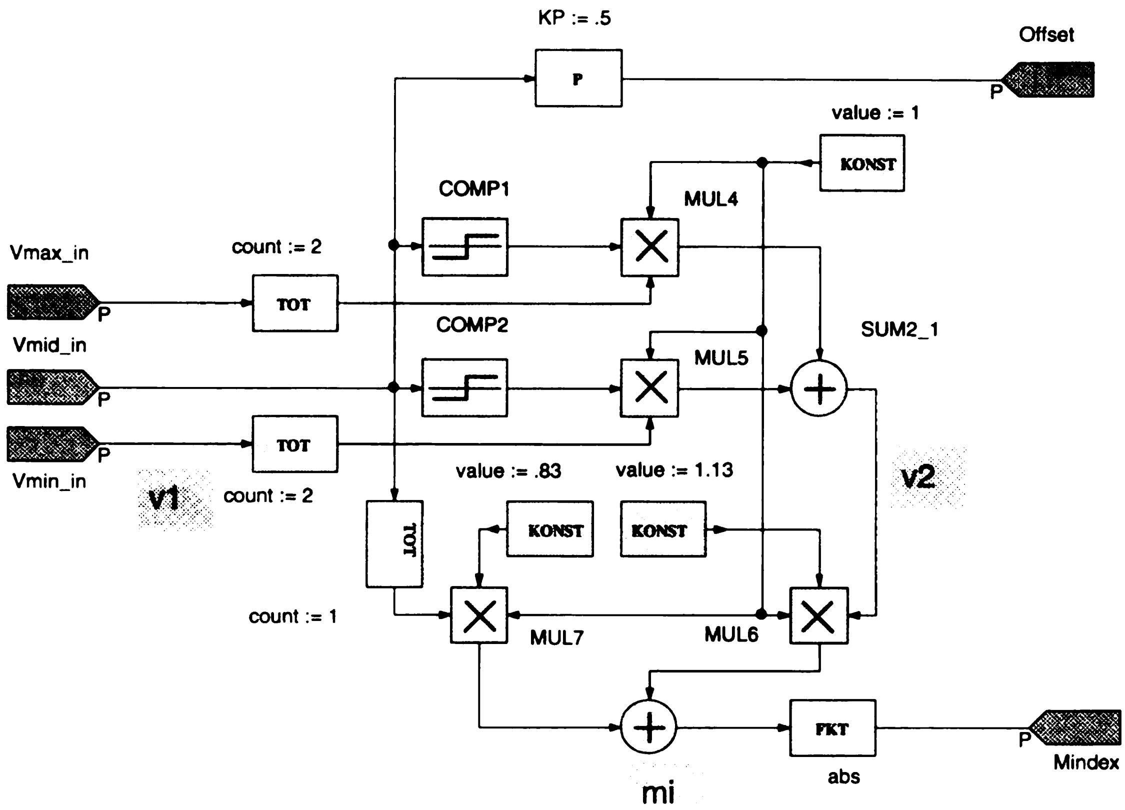


Figure 5.3. Modulation index and offset calculations.

$$\text{if } v_1 < 0 \quad v_2 = v_{min} \tag{5.7}$$

$$m_i = |k_1 * v_1 + k_2 * v_2|. \tag{5.8}$$

With this procedure, the absolute value over each phase, and the procedure to find the sign of the offset are eliminated. Notice that an absolute value has to be calculated only once at the end, since m_i is always positive. The decision to select between v_{max} and v_{min} is implemented with comparators and multipliers. The block diagram of this method is shown in Figure 5.3.

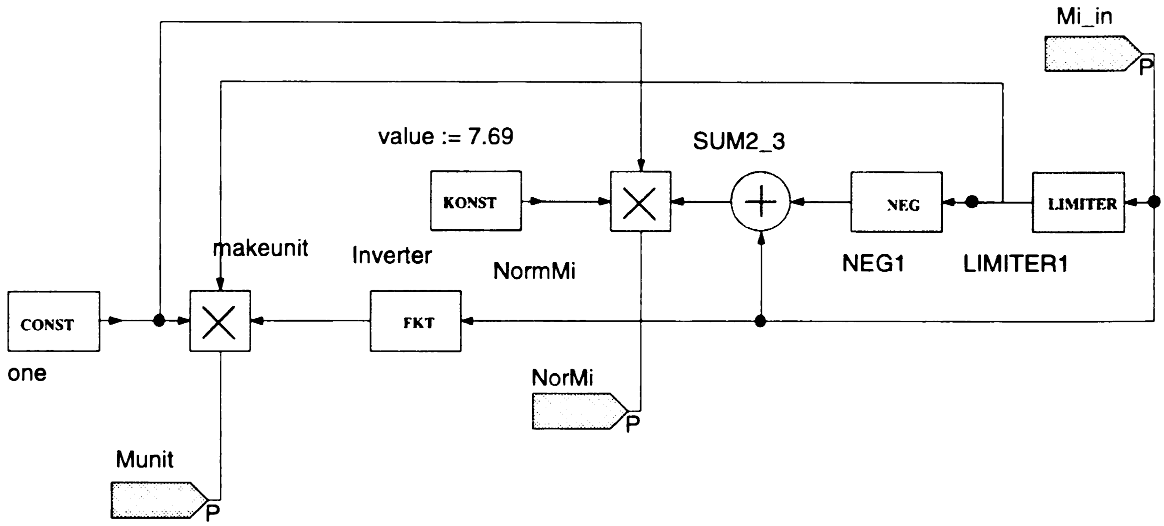


Figure 5.4. Normalized m_i and normalizing factor $Norm$.

Normalized m_i and normalizing factor

The normalized modulation index determines which portion of the square or the sinusoidal reference signal will be present in the output reference signal. The normalized modulation index is calculated by subtracting the reference m_i from the limited m_i and dividing then by 0.27. For $m_i < m_{ilim}$ the $m_{inorm} = 0$ and for $m_i > m_{ilim}$ the modulation index has a value between 0 and 1. The normalizing factor keeps the amplitude of the reference signal through all the overmodulation range. The unitary factor is calculated by dividing the real m_i by the limited m_i. For m_i less than m_{ilim}, m_{iunit} is equal to 1 and for m_i greater than m_{ilim}, m_{iunit} is equal to $(m_{iunit} - 1)/0.27$. The block diagram for the calculation of those two normalizing factors (m_{inorm}, m_{iunit}) is shown in Figure 5.4.

Square Sinusoidal Mixer

This stage is implemented as described in Chapter 2. The sign signal is computed from a comparator with zero reference and output values of -1 and 1. The three-phase mixers follow the addition blocks used to implement the offset and multiplier blocks to keep the reference signal with amplitude 1. The blocks to implement the mixer are shown in Figure 5.5.

Complete System

Each component described above was encapsulated in separated blocks and connected together to form the proposed proportional reference generator. In Figure 5.6 the block diagram for this one-step version of PRSG is shown.

5.1.2 PRSG Using Unmodified Reference signal

In sinusoidal pulse width modulation, SPWM, the reference signal is unmodified for the linear range. The maximum modulation index (m_{ilim}) for the linear range is equal to 1. The PRSG for this mode starts its interpolation between unitary sinusoidal and square waveform at this point, until the maximum modulation index of 1.27. The complete system shown in Figure 5.6 can be used to simulate both sinusoidal PWM and space vector PWM. For space vector PWM the following condition has to be set:

$$Offset = 0 \tag{5.9}$$

$$m_{ilim} = 1 \tag{5.10}$$

$$m_{inorm} = \frac{(m_i - m_{ilim})}{.27}. \tag{5.11}$$

In order to analyze the linearity and total harmonic distortion, in the simulation, the modulation index is changed from 1.0 to 1.27 in increments of 0.03. This mod-

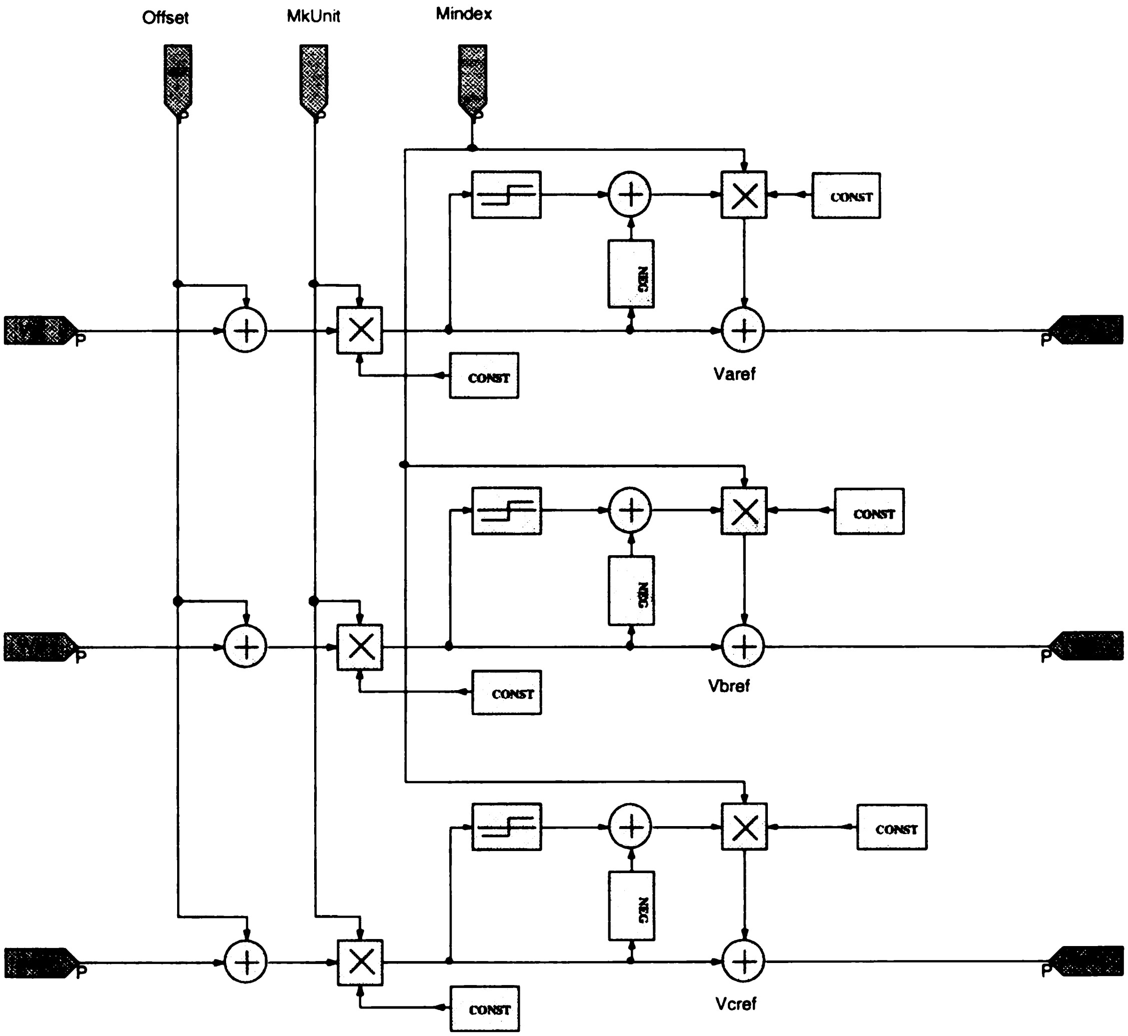


Figure 5.5. Square-sinusoidal mixer.

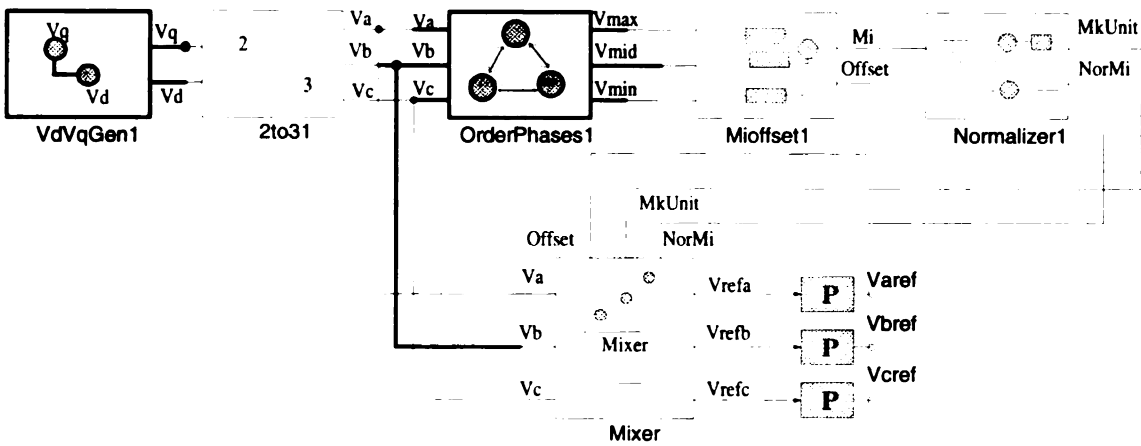


Figure 5.6. Complete system for PRSG.

ulation index is applied to the amplitude of the v_d and v_q generator. This generator has a fixed frequency of 60Hz, and each modulation index is kept constant for each cycle. Therefore, about 1000 points are recorded in each cycle for Fourier analysis. The waveform analyzed is the phase voltage, since it does not contain triple harmonics. The phase voltage can be obtained by applying the three output voltages to a resistive Y configuration, and measuring the voltage of one leg. Using the flexibility of Simplorer we obtain the phase voltage by subtracting the neutral voltage from the reference signal:

$$v_n = v_{aref} + v_{bref} + v_{cref} \tag{5.12}$$

$$v_{an} = v_{aref} - v_n. \tag{5.13}$$

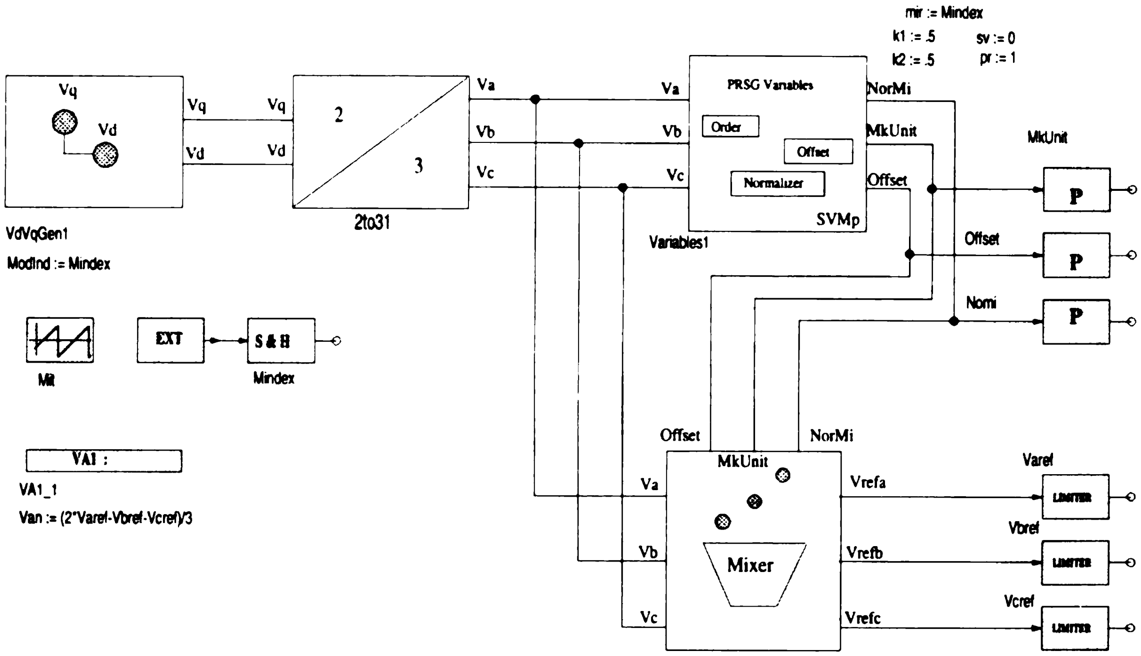


Figure 5.7. PRSG block diagram with parameters.

Actual or aproximate modulation index

The block diagram presented above uses the proposed formula for modulation index calculation $m_i = k_1 * v_1 + k_2 * v_2$. In order to measure the accuracy of the proposed method, the actual modulation index, that it is present in the system, is used. The blocks for phase ordering, modulation index calculation, and normalizing factor were grouped in one. Some components were added to this block to allow the simulation of both sinusoidal and space vector case, each one with predicted and actual modulation index. Figure 5.7 shows this new block diagram.

Therefore, for sinusoidal modulation case, two versions of results are shown: real and predicted modulation index. Figure 5.8 shows the simulation results for PRSG sinusoidal case with calculated modulation index.

In the left-top corner of Figure 5.8 the fundamental output voltage is plotted

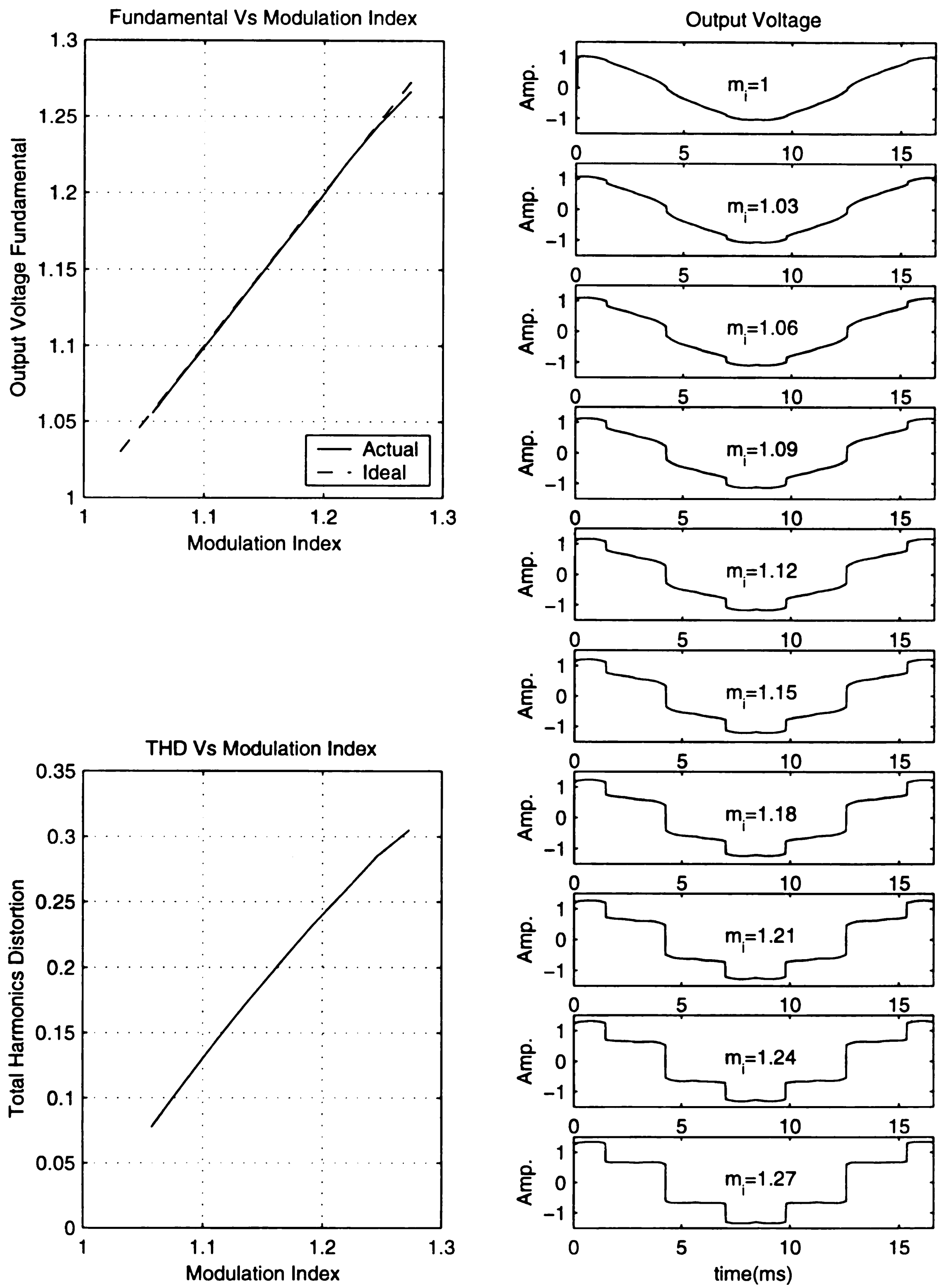


Figure 5.8. Simulations results for PRSG-unmodified using approximate m_i.

against the m_i command. The modulation index it also plotted against itself as a reference for the linearity of the tested method. This graph shows that PRSG posses a perfect linearity. The total harmonic distortion (THD) also changes linearly with respect to m_i. Although this linear behaviour of the THD can be satisfactory for many applications, it can be improved with space vector modulation and two-step transformation. In the right part of the graph, the amplitude of the reference signal is plotted vs. time, for each of 10 selected values of m_i in the overmodulation range. For modulation index of 1.0 the reference signal looks almost sinusoidal. As the modulation index increases, the waveform looks more like the six-step waveform.

In order to evaluate the effectiveness of the modulation index calculation, the same simulation was repeated using actual modulation index. This value could be calculated from v_d and v_q with the formula

$$m_i = \sqrt{v_q^2 + v_d^2}. \tag{5.14}$$

This actual value of the modulation index is taken from the (v_q and v_q) generator which is available. This m_i value corresponds to the stair generator which produce the ten different values of modulation index. This waveform is generated in Simplorer with a sawtooth waveform generator and a sample-and- hold block. The sample-and-hold block keeps the modulation index constant during each period.

Figure 5.10 shows that the output voltage fundamental obtained with PRSG, using the calculated modulation index, is similar to the ones obtained with the actual modulation index. This shows the accuracy of the calculated modulation index.

5.1.3 PRSG Using Modified Reference Signal

The PRSG-SVM case extends the linearity of the linear range of the space vector PWM to six step mode. The main advantage of the SVM is that the linear range is

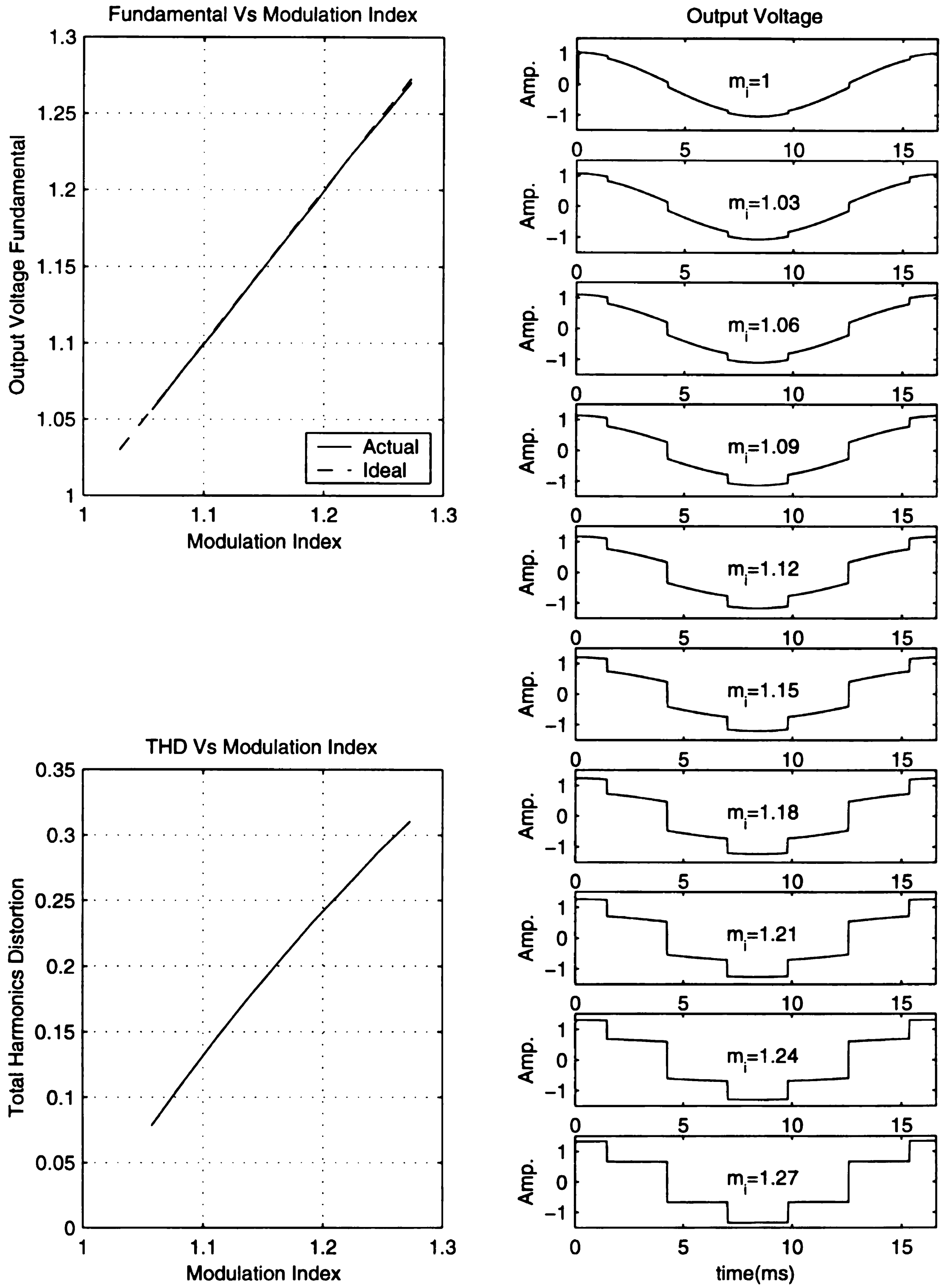


Figure 5.9. Simulations results for PRSG-unmodified using actual m_i.

extended to $m_i = 1.15$ and the THD is practically zero until this limit. The PRSG interpolates between the modified sinusoidal reference signal with the square wave, in a way similar to the sinusoidal case. All the block diagrams remain the same, except the following parameters

$$Offset = \frac{v_{mid}}{2} \tag{5.15}$$

$$m_{ilim} = 1.15 \tag{5.16}$$

$$m_{inorm} = \frac{m_i - m_{imin}}{.12}. \tag{5.17}$$

The simulation results for this case are shown in Figure 5.10. The output voltage fundamental changes linearly with respect to the modulation index. The total harmonic distortion also changes linearly with respect to the modulation index. This fact shows that the sinusoidal-square mixer operates linearly over the Fourier components of the input signal (sinusoidal or space vector reference). Although the THD also changes linearly with respect to the m_i, this case offers low THD output response for each m_i, since it starts later, at $m_i = 1.15$.

The same simulation was performed on the SVM case, using the actual modulation index instead of the aproximate one. As is shown in figure 5.11, the output fundamental is the same as when the estimated modulation index is used. This shows that the effect of the ripple caused for linearazing formula and the amplification of this ripple by the reduction of the overmodulation range is negligible.

5.1.4 PRSG using Two-Step Transformation

Although the THD for modified reference case of the PRSG is lower than the unmodified refernce case, it is still higher than that of the classical Mode I and Mode II. In order to improve the THD, even more the PRSG is applied in two steps. First,

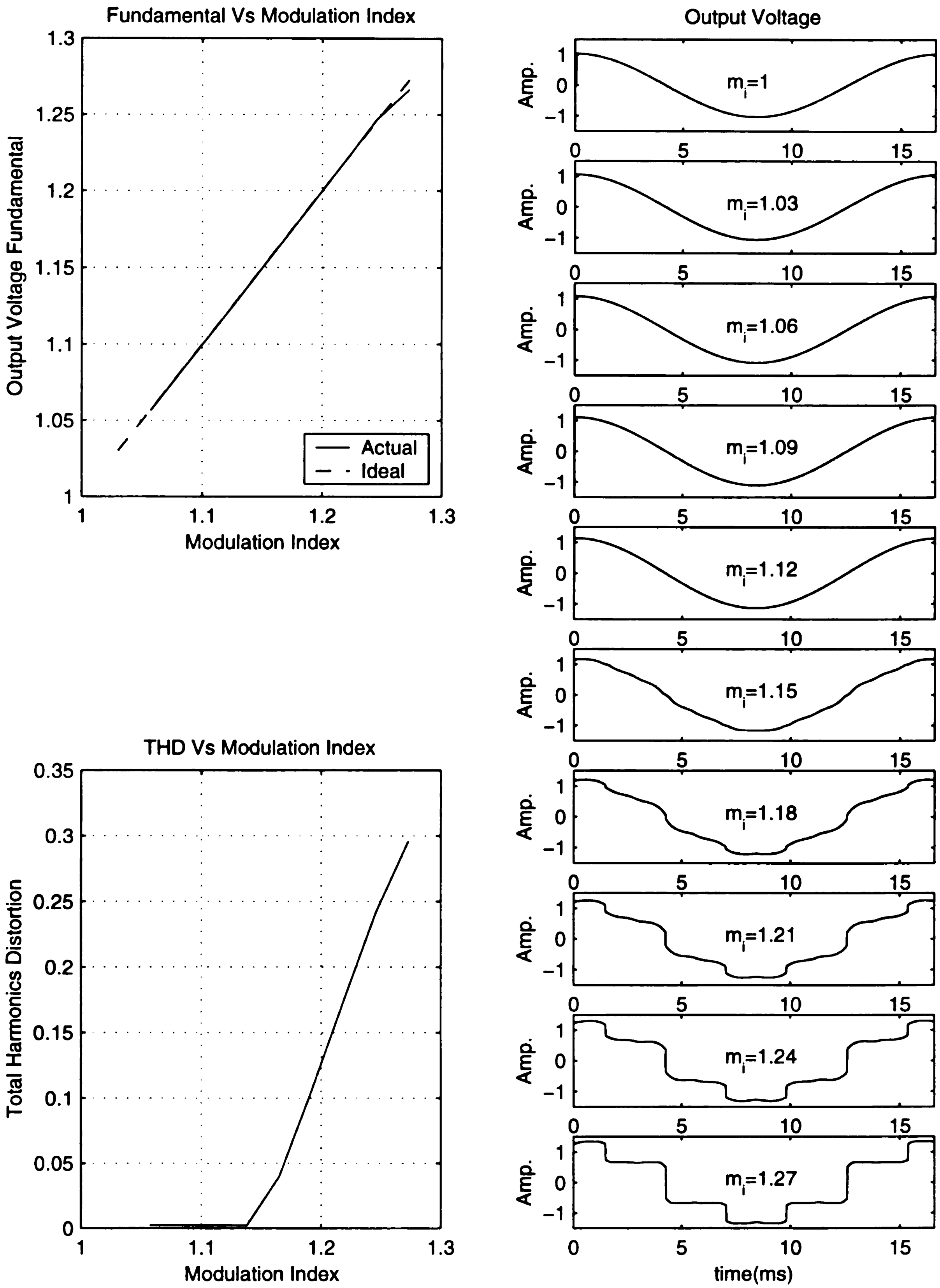


Figure 5.10. Simulation results for PRSG-modified using approximate m_i.

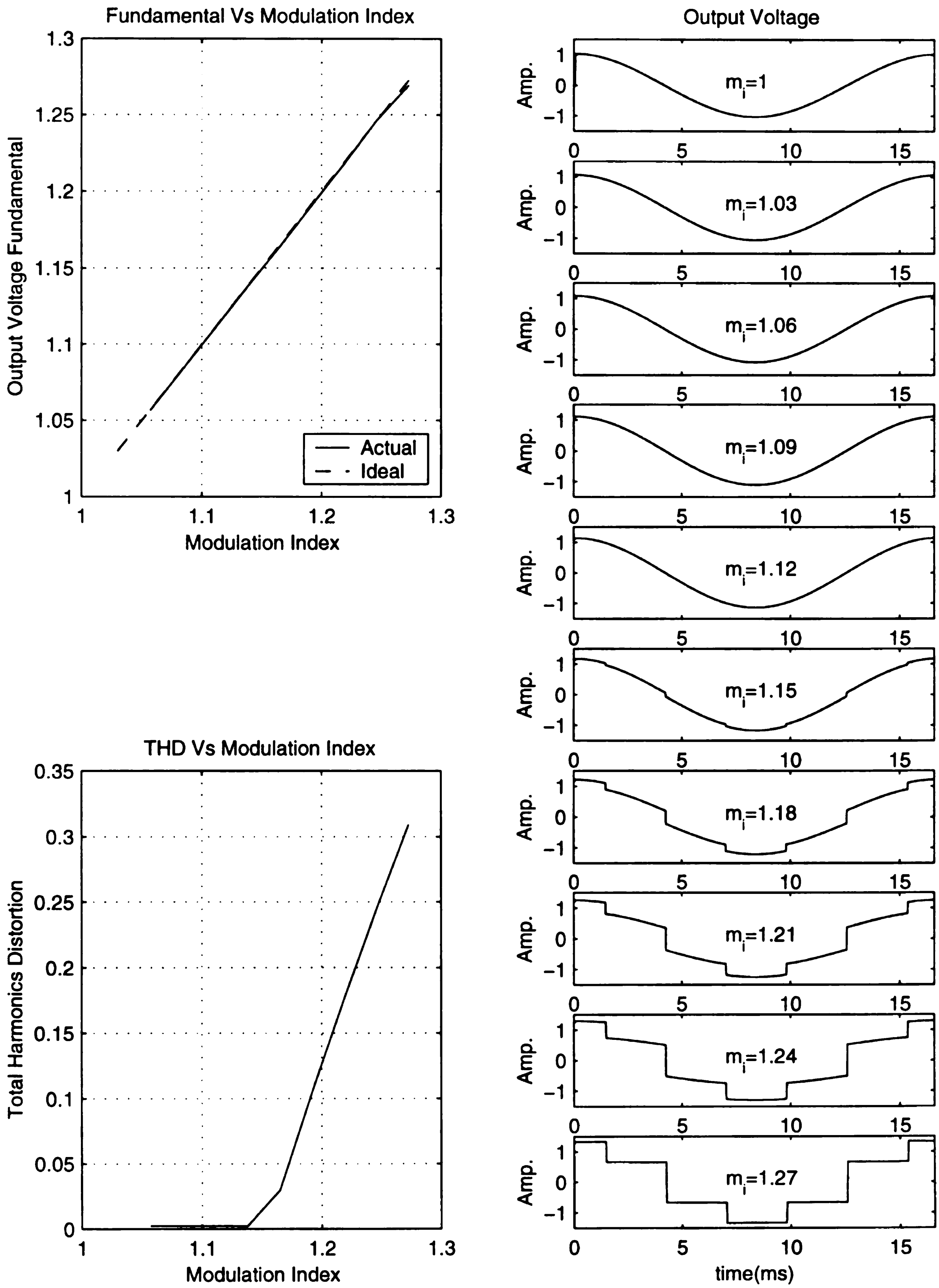


Figure 5.11. Simulation results for PRSG-modified using actual m_i.

the pure or modified sinusoidal (SVM) is converted to a trapezoid. The conversion is done by linearly interpolating between trapezoid and the unitary sinusoidal reference signal. The key of this step is to synthesize the trapezoid waveform with the information of the unitary sinusoidal. A trapezoid can be created with the information of the angle which is not available. Although this method was used in the preliminary simulation in Matlab, a more efficient way to synthesize a trapezoid is required for Simplorer simulation and experimental implementation. This method consists of multiplying the unitary sinusoidal by 2 and then applying a limiter of ± 1. Figure 5.12 shows the two step state diagram and equation used for Simplorer simulation. Figure 5.13 shows the linearity, THD and waveform transformation of this method.

5.2 Mode I and Mode II

In order to compare the proposed overmodulation method with the classical methods Mode I and Mode II, a Simplorer simulation was performed. These two modes work together to cover the entire overmodulation index range. They were implemented using state diagrams and equations. The angle α_1 in Mode I and the angle α_2 in Mode II were implemented using linear equations. Although these methods are normally implemented using complex equations, numerical methods, or look up tables, here they were implemented using linear equation to show that they do not have inherently linearity as the PRSG. Figure 5.14 shows the state diagram and the code needed to implement the Mode I and II in Simplorer. Figure 5.15 shows the linearity, THD and waveform transformation of this method.

5.3 Comparison of PRSG and Existing Methods

Figures 5.12 and 5.15 for the PRSG and existing method are presented in this section in a different way, than they were presented in the previous section, in order to

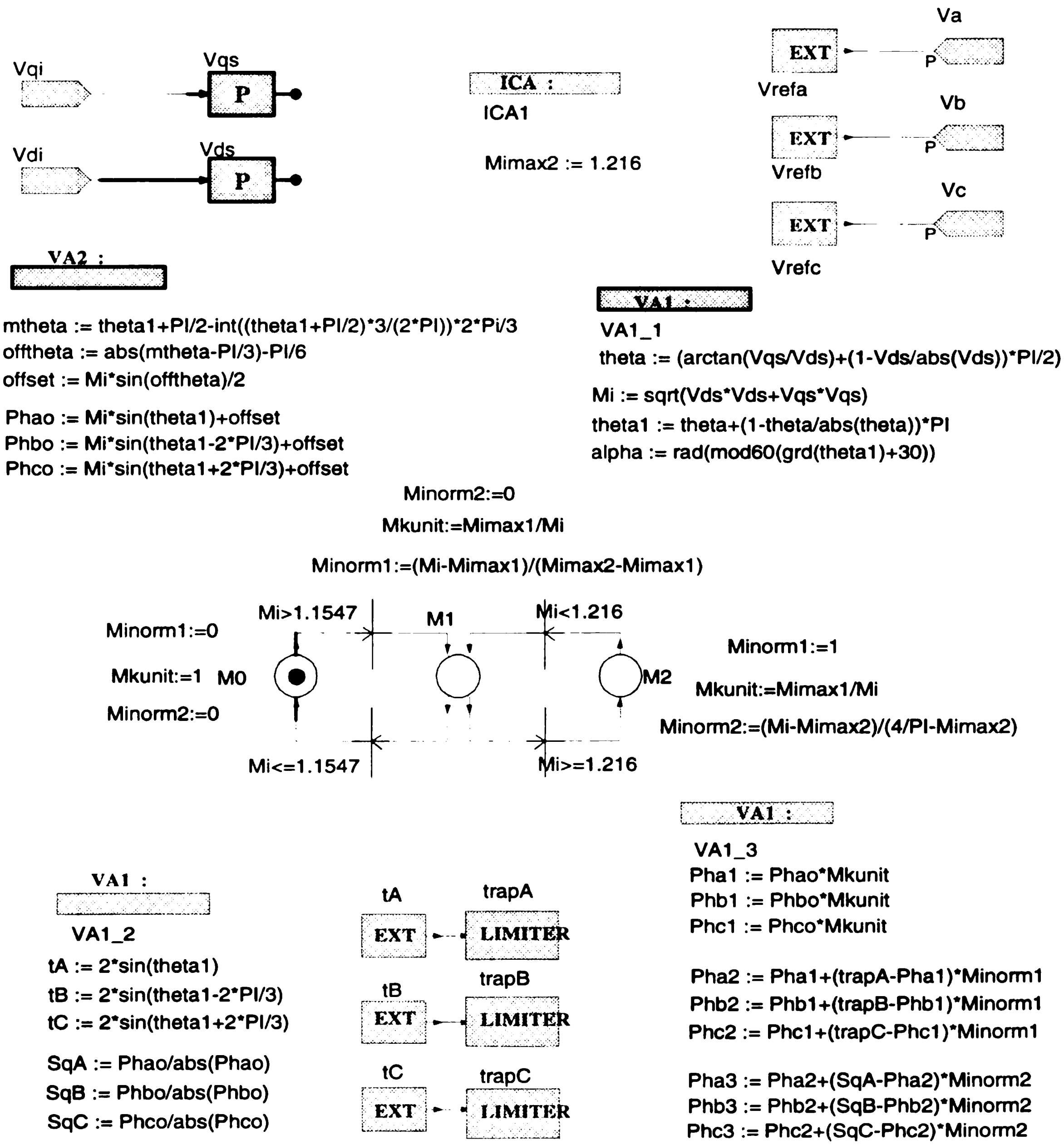


Figure 5.12. Two-step state diagram and equations.

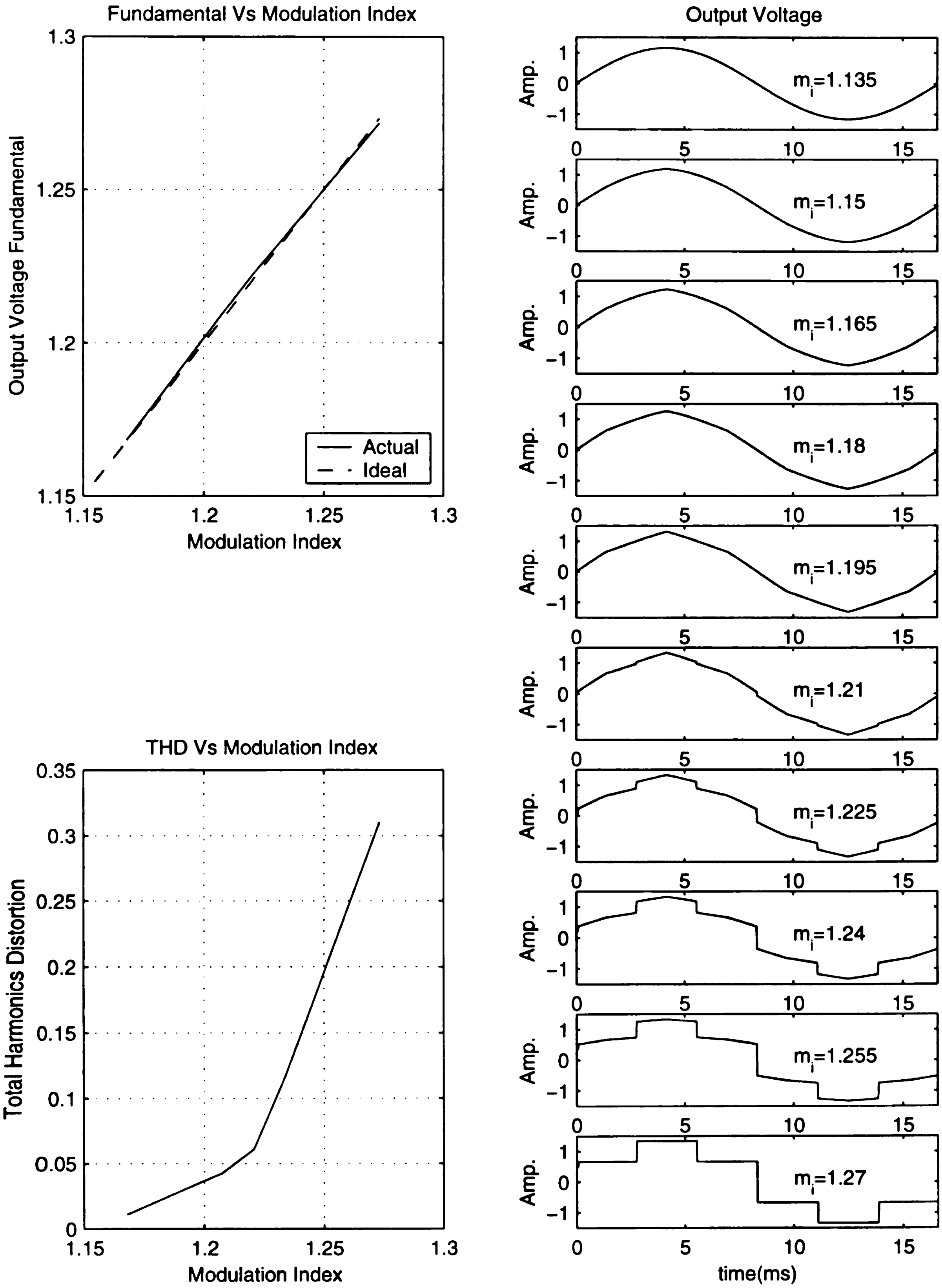


Figure 5.13. Simulation results for PRSG-two-step.

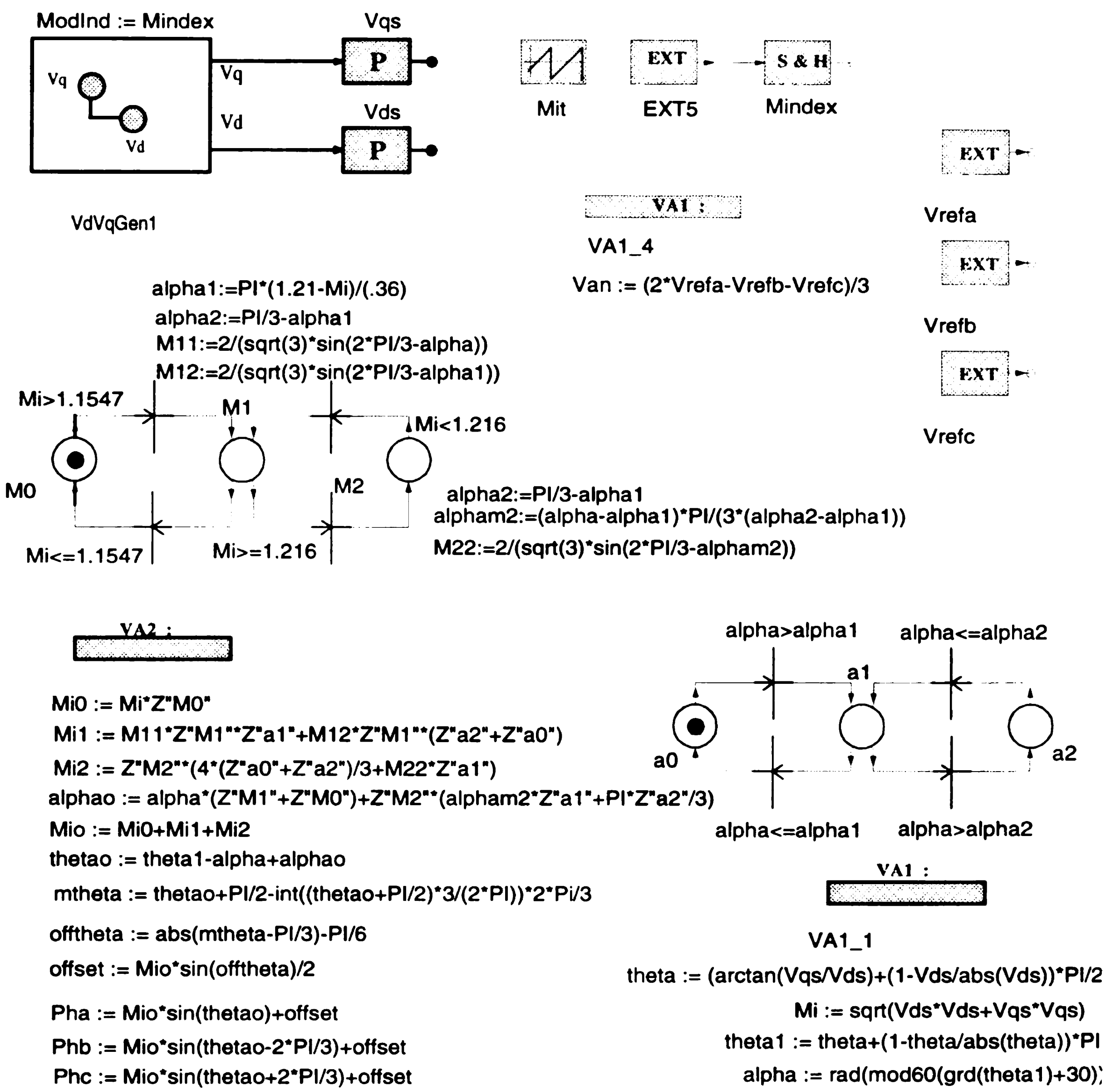


Figure 5.14. Mode I and Mode II state diagrams and equations.

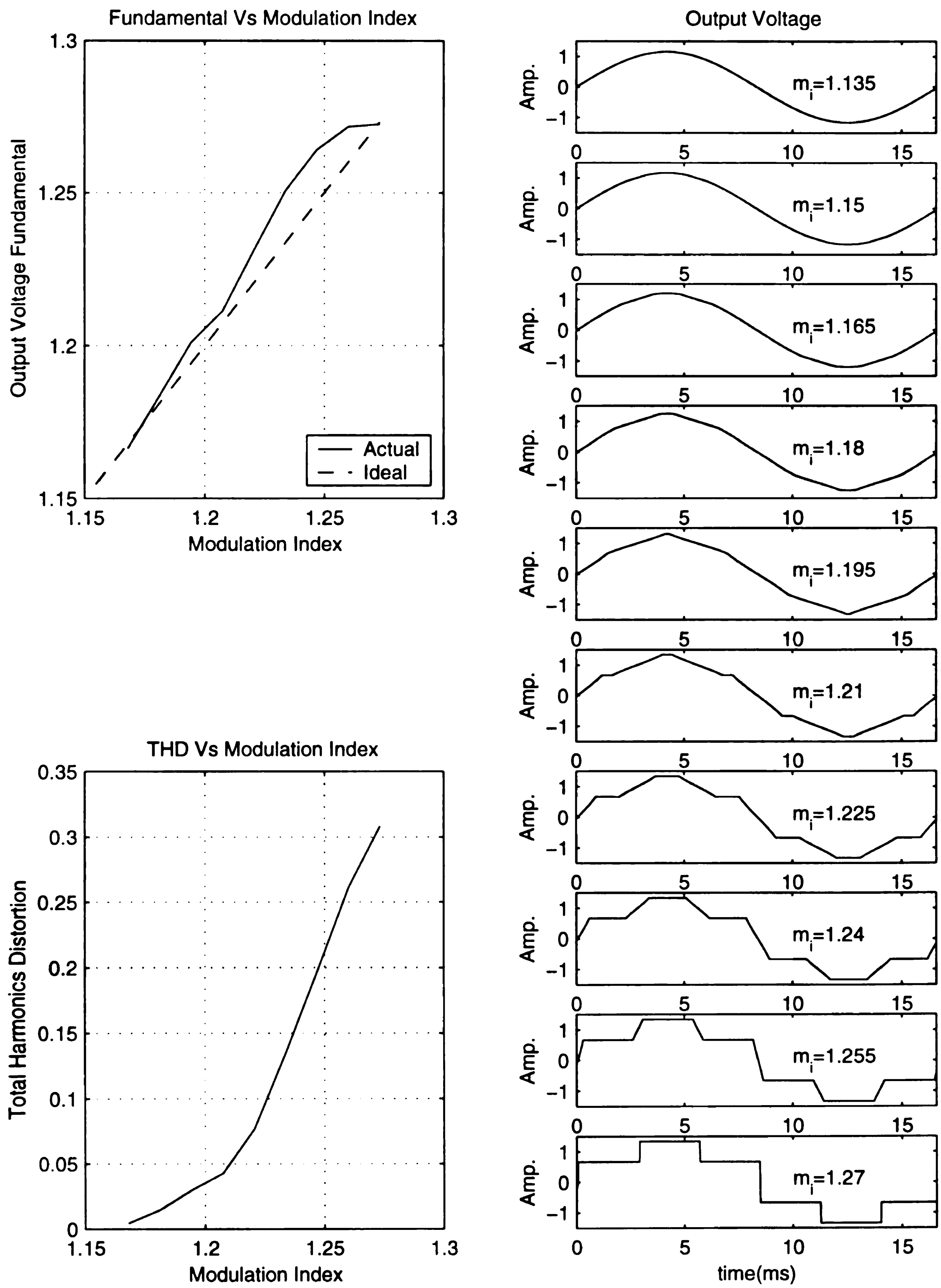


Figure 5.15. Simulation result for Mode I and Mode II.

compare separately each one of the desirable characteristics of the overmodulation method: linearity, simplicity and harmonic distortion. Although other methods were studied in the literature review, and simulated with Matlab files, Mode I and Mode II report the lowest THD of all.

5.3.1 Linearity

Linearity requires that the desired modulation index vs. output voltage fundamental follows an straight line. The overmodulation range has been considered until now as a very nonlinear one, therefore all the existing methods including Mode I and Mode II, use linearizing methods in their implementation. The proposed method claims for the first time that only linear equations are necessary to obtain a perfect linearity at the output.

In order to make a fair comparison between existing methods and the proposed method, the first were implemented using straight line equations. That means for example, that the holding angle, α_h, in Mode II, which changes from 30^0 to 0^0 for m_i from 1.21 to 1.27, is changed using the linear equation below

$$\alpha_h = \frac{0^o - 30^o}{1.27 - 1.21} * m_i + 30^0. \tag{5.18}$$

In a similar way, angle α_1, which changes from 0^0 to 30^0 for m_i between 1.15 to 1.21 in Mode I, is changed using the linear equation below

$$\alpha_1 = \frac{30^o - 0^o}{1.21 - 1.15} * m_i. \tag{5.19}$$

The output voltage fundamental vs. modulation index curve for both the existing and the proposed method is plotted in Figure 5.16. The curve for the proposed method follows a straight line, while the curve for Mode I and II does not. Linearity as was explained in Chapter I, makes the feedback control loops easier to design.

Figure 5.16 shows the linearity comparison between Mode I and Mode II and the PRSG two-step mode.

5.3.2 Harmonic Distortion

The harmonic distortion of the proposed method was compared to that produced by Mode I and Mode II. Both curves were plotted in the same graph. The PRSG curve shows higher harmonic distortion in the first part of the range ($1.15 \leq m_i \leq 1.20$), and it shows lower THD than mode I-II for $m_i > 1.20$. This is because the harmonic distortion of the trapezoid created in the two-step transformation contains less harmonics that the trapezoid resulted from the projection of the hexagon. PRSG shows lower THD in the most important range, where the distortion tends to have higher values. Therefore, PRSG-two-step has better overall THD performance than Mode I and Mode II. Figure 5.17 shows the results of this comparison.

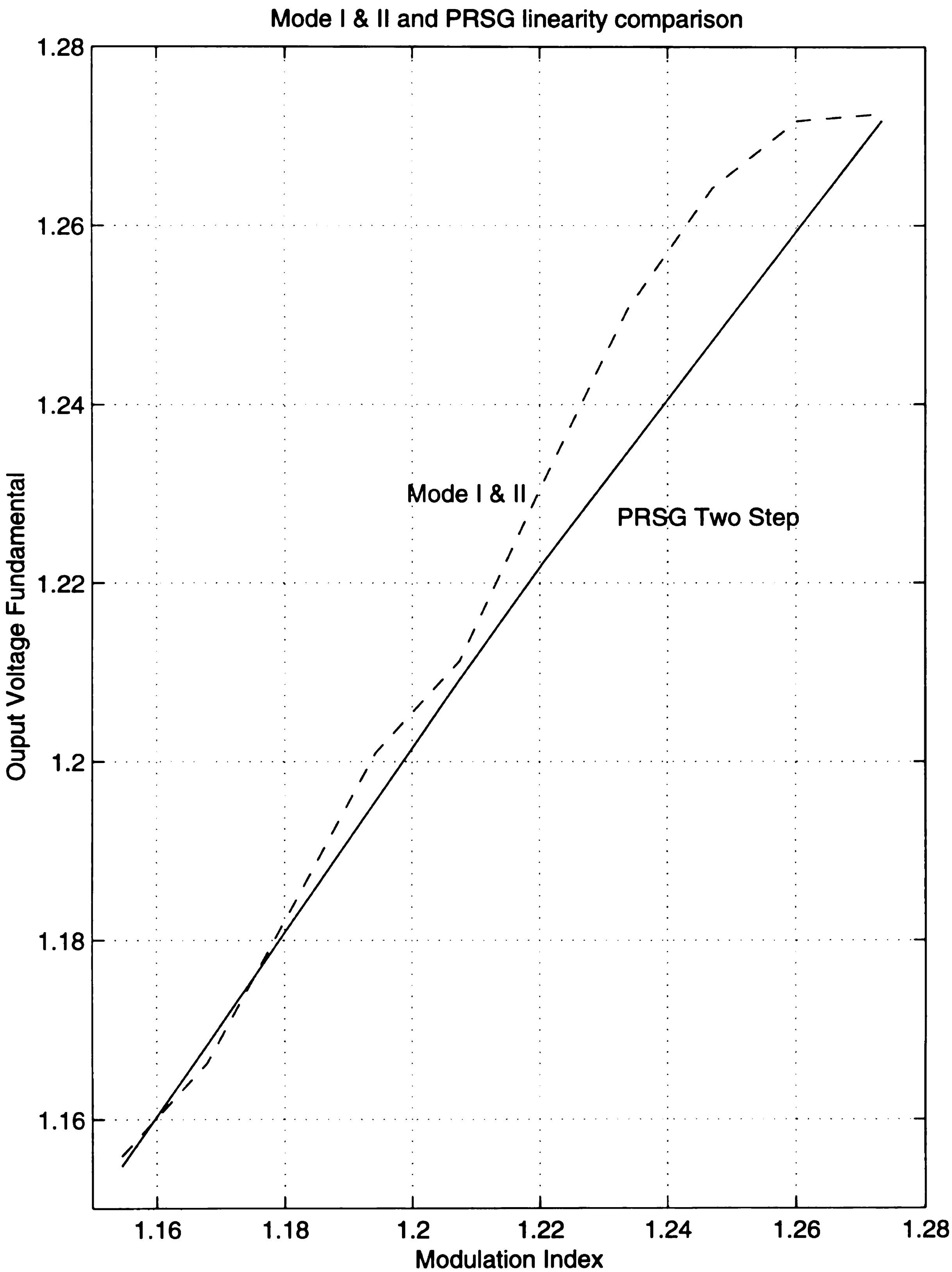


Figure 5.16. PRSG-two-step and Mode I-Mode II Linearity comparison..

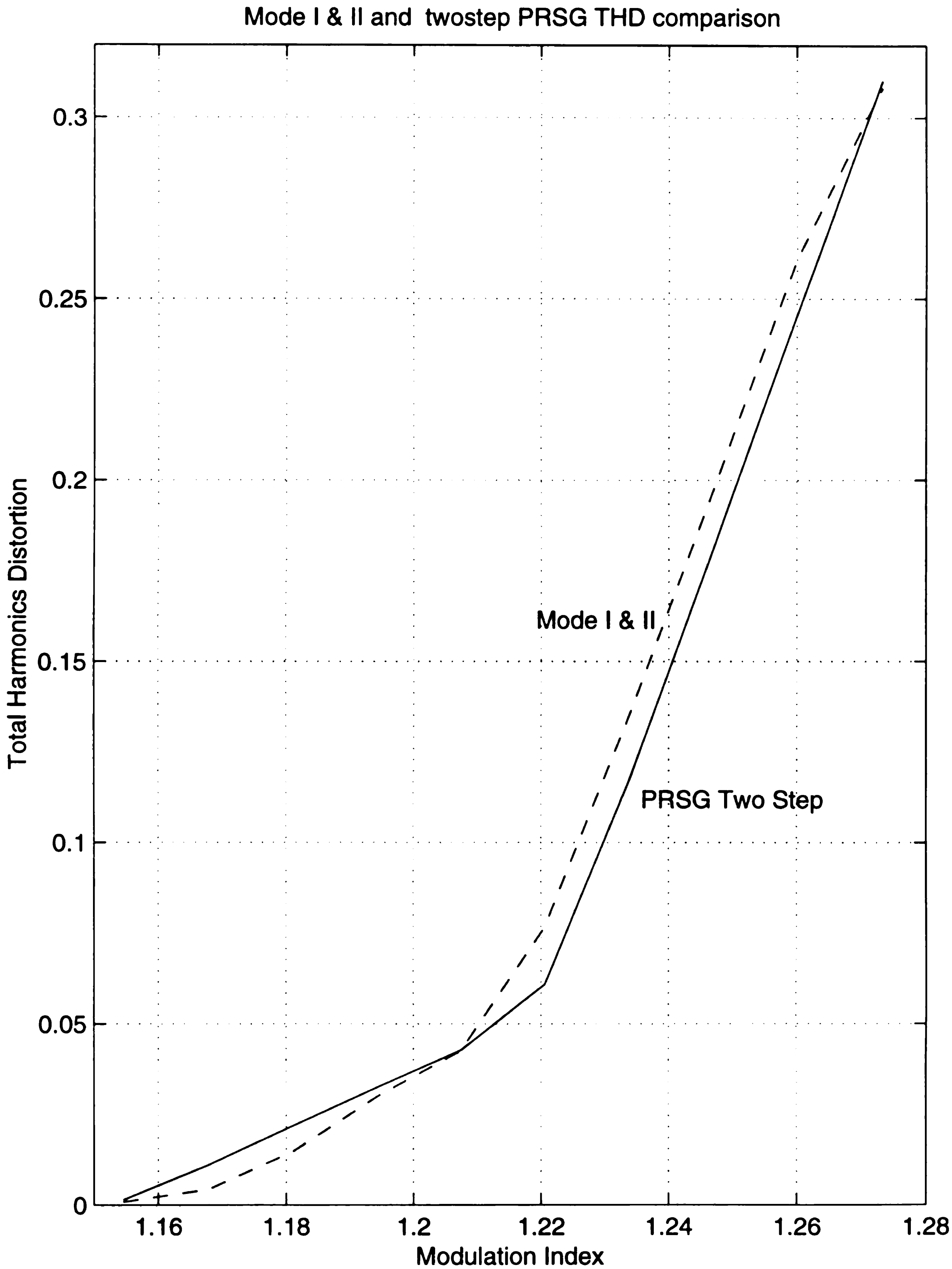


Figure 5.17. PRSG-two-step and Mode I-Mode II THD comparison.

CHAPTER 6

Experimental Results

In this chapter the implementation of the new overmodulation method in a Intel 80c196 microcontroller is described. This 16 bit microcontroller has three 8-bit and one 4-bit port. It also contains Event-compare registers (EPA), which make the implementation of the PWM easier. The modulation index command was sent from a PC to one of the ports of the microcontroller. PRSG was implemented in assembler code and used to control (open loop) a three-phase inverter. The inverter was connected to a three phase induction motor. The phase voltages are recorded to verify the results about linearity and harmonic distortion obtained in the mathematical analysis and simulation chapters.

6.1 General Description

6.1.1 One Byte Resolution

The microcontroller receives the modulation index command from the parallel port of a PC. This is a word of 7 bits, allowing the modulation index word to vary from 0-127. This resolution was selected for many reasons. First, is the same resolution to be used in the proposed chip, therefore it will be a way to test if this resolution is good enough. Second, there is a one-to-one relationship between the modulation index value 0-1.27 and the 7 bit binary resolution (0-127), which makes it unnecessary to scale all the constants used in the theoretical analysis. When this modulation index is converted to amplitude (-127 to 127), it still fits in a byte. Third, with this resolution the counter of the timing section only has to count 200 different values, which allows to reach a switching frequency of $60kHz$ if necessary. Finally, although the 80c196 is a 16 bit microcontroller, it is more convenient to use the extra 8-bits to do the necessary scaling, when working with fixed point operations.

6.1.2 Internal Generation of vd and vq

The value of m_i received through the parallel port is used to calculate the v_d and v_q reference signals, using sine and cosine look-up tables inside the microcontroller. Although it appears a contradiction, since it is claimed earlier than the PRSG method works directly with v_d and v_q and does not need a look-up table, this approach its necessary for several reasons: First, the real modulation index was necessary to compare with the one calculated with the proposed method. Second, although the 80c196 has three and a half 8-bit digital ports, the serial communication is lost when more than two of them are used simultaneously. In the actual set-up one port is used for the modulation index and the other in the output signal. The serial communication is necessary to have available troubleshooting technics. Third, only one parallel port is necessary to send the modulation index, reducing wiring and I/O PC requirements. Finally, if the angle information is generated outside of the microcontroller, the information sent through the parallel port has to be 255 times faster. Since everything is controlled from Matlab involving DSP read and write operation, this speed is too high for the complete cycle of measurement. The modulation index information is not changed at each cycle but one every 10 cycles, since each modulation index is sampled several times. Therefore the difference in speed using this approach is even higher. Since all the plots required for this experiment have the modulation index as a base, it is more convenient to send the modulation index to the microcontroller and to generate the angle automatically inside it. Once the program is running using the generated values of v_d and v_q, the 80c196 port used for m_i can be used to receive v_d and port2 (shared with serial communication), can be used to receive v_q. The code for v_d and v_q generation and the look-up tables can be removed and the program will be ready for vector control operation.

6.1.3 Three Phase Output Signals

The output generated by the microcontroller corresponds to the timing signals of the three phases. These signals feed the voltage source inverter and determine whether each phase will be connected to the positive or negative rail of the DC bus. Although six signals are needed to drive the IGBT bridge, the rest of them are generated inside the inverter. The dead time necessary to avoid short circuit in each phase leg is also generated inside the inverter.

6.1.4 Voltage Source Inverter

The three phase inverter is really an AC-AC converter where the input is a three-phase 60Hz, 208 system. These three AC inputs are rectified by a diode bridge and filtered by a capacitor. In this application the DC bus voltage was controlled by changing the input voltage with a three-phase Variac. This voltage was set to 150V, lower than the normal 300V, since the motor rated voltage is 120Vrms and to prevent saturation when the voltage applied goes into the overmodulation range. The control circuit inside the inverter not only creates the gate signals for the other three IGBTs, and the signal delays, but also the gate voltages necessary to drive the IGBTs.

6.1.5 Data Acquisition Systems

The voltage applied to the motor is measured across two of the phases (V_{ll}) through a voltage sensor and is fed to the DSP. The DSP, which is an internal card inside the PC, takes 123,000 samples per second with a 12 bit analog-to- digital converter. Over 4208 points that correspond to two cycles are taken and saved inside the DSP board. A compiled C program is in charge of taking the data from the DSP board and saving it in a file. This file is then opened by a Matlab program and analyzed. The program finds zero crossing points in order to determine an exact cycle, and the correct phase, before performing the Fourier analysis. Then the fundamental and harmonic distortion is calculated and saved together with the modulation index. After that, a new modulation index is sent to the parallel port starting the cycle again. This procedure is repeated for each modulation index in steps of 0.01 over the entire overmodulation range.

6.2 PRSG Microcontroller Assembler Implementation

In this section the different parts of the assembler program with are relevant to the real implementation are explained. Then the results obtained for each case are shown.

6.2.1 Timing System

After the three reference signals are calculated using the PRSG technique, they are sent to the EPA units. These units compare the data saved in their register (reference signals) with the count of counter 2. Once these values they become equal, the EPA changes the output of one external pin of the microcontroller without intervention of the CPU. However, an interrupt routine has to be executed with the following task:

- load reference signal data in EPA1-3 registers,
- change the direction of the counter,
- load the starting state of the three signals,
- program each EPA1-3 to set or reset the output pin,
- set the count when the next interrupt will be produce.

A fourth EPA(0) is necessary to make the comparison between counter 2 and the count for the next interrupt and causes the interrupt routine be executed. The change of direction of the counter is necessary to produce the triangular wave that produces (through the comparison with the EPA1-3 registers) the PWM. Since the EPAs only change state when the compared values are equal, it is necessary to synchronize each EPA by programing it to set the output pin when the counter is counting up and programing it to reset it when the counter is counting down. Finally, setting the starting state of the three signals is a requirement of the overmodulation, since even when a value high or low enough can be set to avoid the EPA to change the output pin, it has to be set with the right value at the beginning of each cycle.

Another part of the assembler program is dedicated to comparing each of the three reference phases voltages to their maximum and minimum possible values (triangular peaks). In case that the phase values are out of range (overmodulation), another set of values are assigned to them, just to make sure than the counter will never reach this value and a glitch is not produced. The starting signal vector, which is updated during the interrupt routine, is formed in this routine. Therefore, about 24 instruction lines are added to the assembler program to avoid these glitches. This routine and the interrupt routine create some overhead of time and code space that could be avoided in a dedicated chip implementation.

6.2.2 Reference Signal Generation

Once v_d and v_q are generated, the 3-to-2 transformation described in the simulation chapter is implemented to obtain $v_a, v_b and v_c$. The most important differences with the simulation is that shift-right arithmetic is used to create the factor $v_d/2$ and that all constants are multiplied by 100 to produce the necessary scaling to carry out the fixed-point operation.

After v_a, v_b and v_c are obtained, the phases are ordered. As in the simulation part, the signed values and not the absolute values are ordered. To implement an ordering routine of three values it is necessary to use decision instructions. An efficient routine uses only two decision instructions and swapping of two variables. Once the phases are ordered, the middle value phase corresponds to v_1 and the other phase whose has the same sign as v_1 becomes v_2. The modulation index is calculated using k_1 and k_2.

Another difference between simulation and microcontroller assembler implementation is that once the modulation index falls within the linear range, all overmodulation procedures are skipped, including all the mixers.

The normalizing factor($Norm$), the normalized modulation index(m_{in}) and the mixers, are implemented in a similar way as in the simulation. However, some differences were introduce here to substitute the absolute values and limiter for decision instructions. The complete block diagrams for the assembler implementation is shown in figure 6.1.

6.2.3 PRSG-Unmodified Reference Case Results

actual m_i case

In the sinusoidal case of the PRSG, the subroutine for offset calculation and addition is skipped. For actual m_i case the ordering of the phases is not needed. Therefore this is the smallest version of the assembler program. The value for m_i is taken from

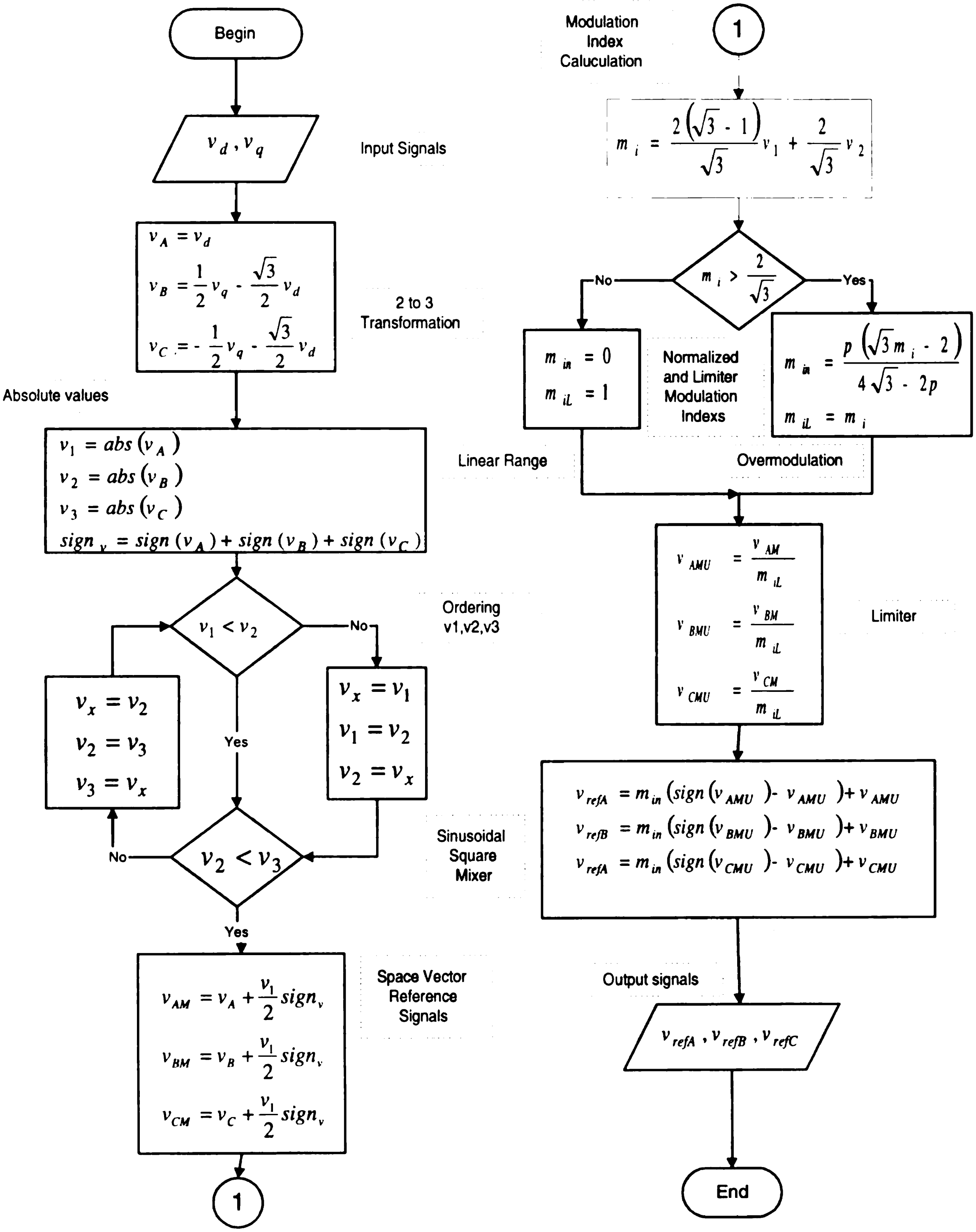


Figure 6.1. Flow diagram of the assembler program.

the input port. When the programs were modified in order to receive v_d and v_q, an extra subroutine was necessary for modulation index calculation. The modulation index is evaluated, and in case of this value fall in the linear range, all overmodulation procedures are skipped and the reference signals are send directly to the EPAs. On the other hand, if m_i is in the overmodulation range, the normalizing factor and normalized modulation index are calculated, and used later in the linear mixers.

Figure 6.2 shows the results obtained for this case. In the left top corner the fundamental vs. modulation index is plotted. Although this graph does not look as smooth as in the simulation, the linear behavior of the fundamental is observed. In the bottom left corner the THD is plotted vs. the modulation index, which shows a linear behavior as predicted. In the right part of the graph the transformation is shown of the voltage applied to the motor, from sinusoidal to six-step mode. This voltage is filtered in the sensor and then in the Matlab program in order to minimize the noise created for the data acquisition system.

Approximate m_i Case

In the approximate modulation index case of the PRSG-sinusoidal, the ordering of the phases, is performed in order to determine v_1 and v_2. Then the modulation index is evaluated to determine whether it is in the overmodulation range. If it is, the normalizing factor is calculated using this estimated modulation index. No effect is noticed in the output fundamental. Only at the end of the range a difficulty is noticed to reach exactly the maximum modulation index of 1.27. The THD varies linearly with m_i but under the imaginary straight line which connects 0 and 30% which is acceptable. In the waveform transformation graph some noise appears in the envelope of the voltage, but because of its high frequency it does not have effect on the load.

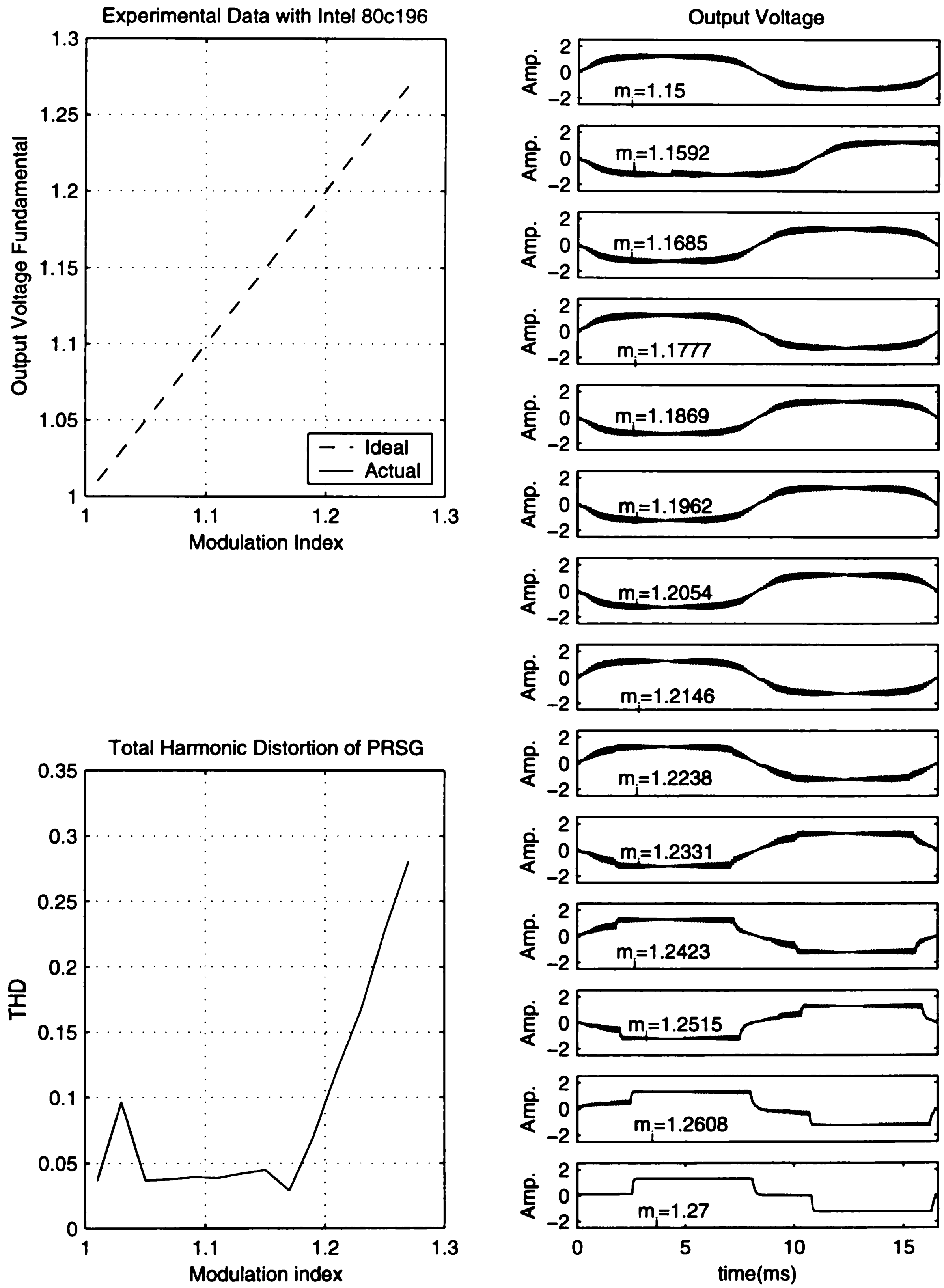


Figure 6.2. PRSG-unmodified experimental results using actual m_i.

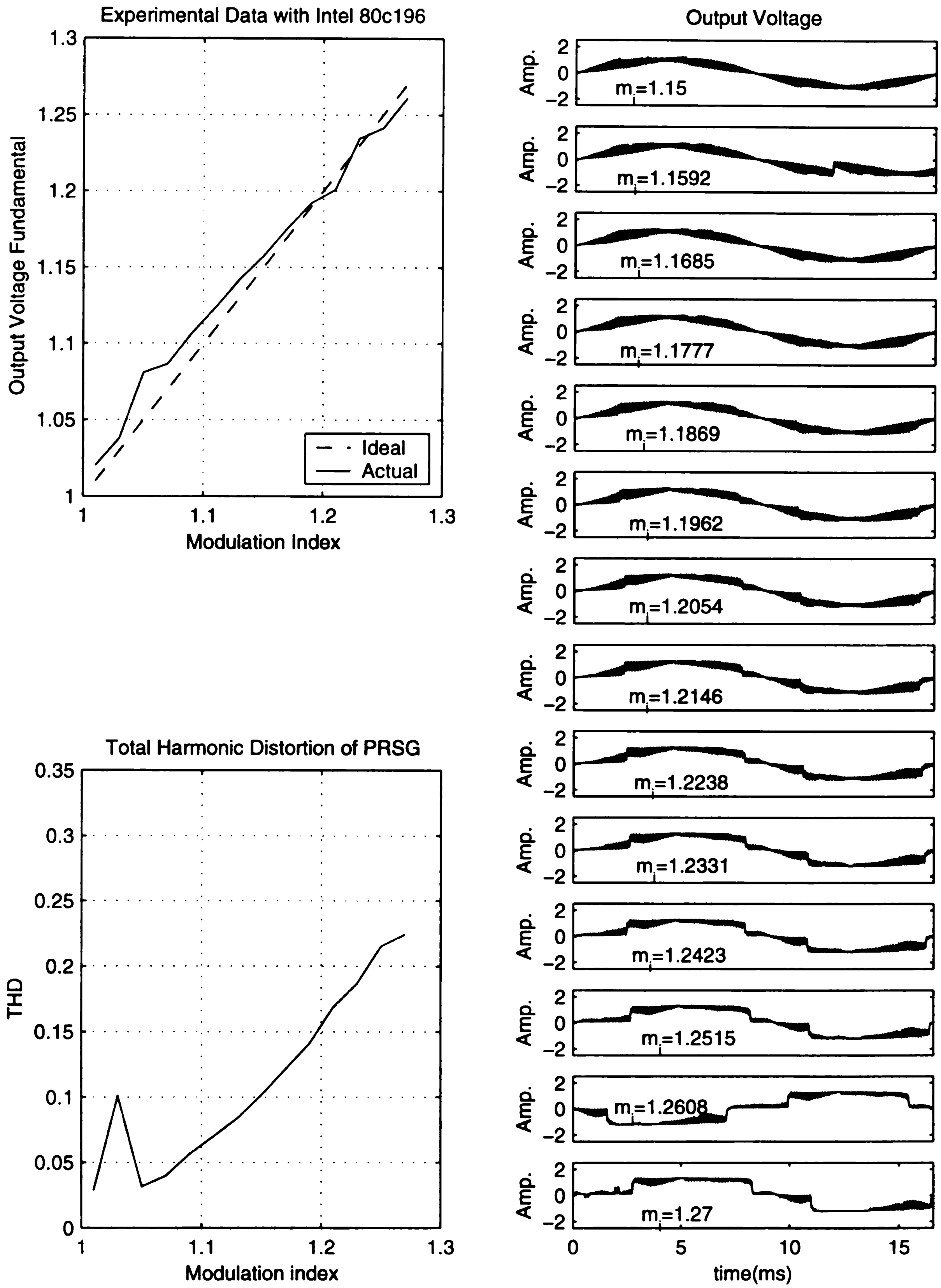


Figure 6.3. PRSG-unmodified experimental results using approximate m_i.

6.2.4 PRSG-Modified Reference Case Results

Actual m_i case

In the Space Vector Reference case, an offset is always added to the three phase signals before they are mixed. Therefore, the ordering subroutine is always executed. However, for the actual m_i sub-case the modulation index is not calculated but taken from the input values. The offset algorithm is the same as was used in the simulation, where the signed phase values are used. Once the range of the overmodulation index is determined, the mixer action is performance.

Figure 6.4 shows that the fundamental changes linearly with respect to the modulation index. It can be observed that the actual and ideal lines are exactly one above the other. The THD curve also shows similar result to those obtained in the theory and simulations. THD remains very low until $m_i = 1.15$, then it behaves linearly from 1 up to 1.27. The waveform transformation also produces satisfactory results, since it shows very clearly how the space vector reference waveform is converted to six-step mode.

aproximate m_i case

For the approximate modulation index sub-case the subroutine for m_i calculation is executed. The graph for this case (figure 6.5) shows an output fundamental with a linear behavior. However at the end of the overmodulation range it was difficult to reach the maximum fundamental output of 1.27. This is because the ripple of the modulation index is amplified by $8(= 1/.12)$ and even when the average error is zero, this ripple is clamped for the limitation of the power supply and can not compensate the part where the calculated m_i is under the average. Even so the imperfection of the data acquisition system and the precision of the assembler program appear to

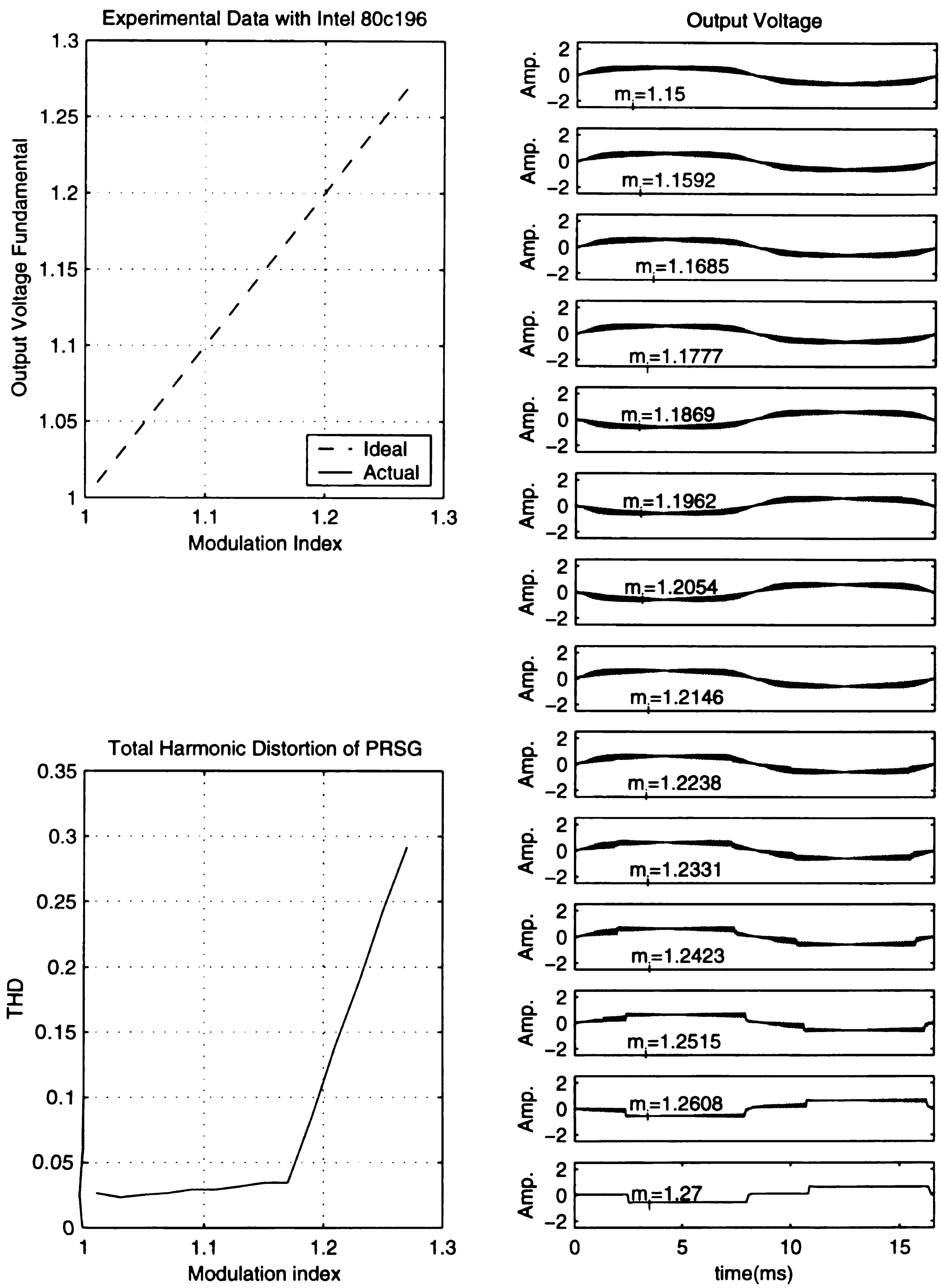


Figure 6.4. PRSG-modified experimental results using actual m_i.

play also some role in the output obtained, since the simulations show better results.

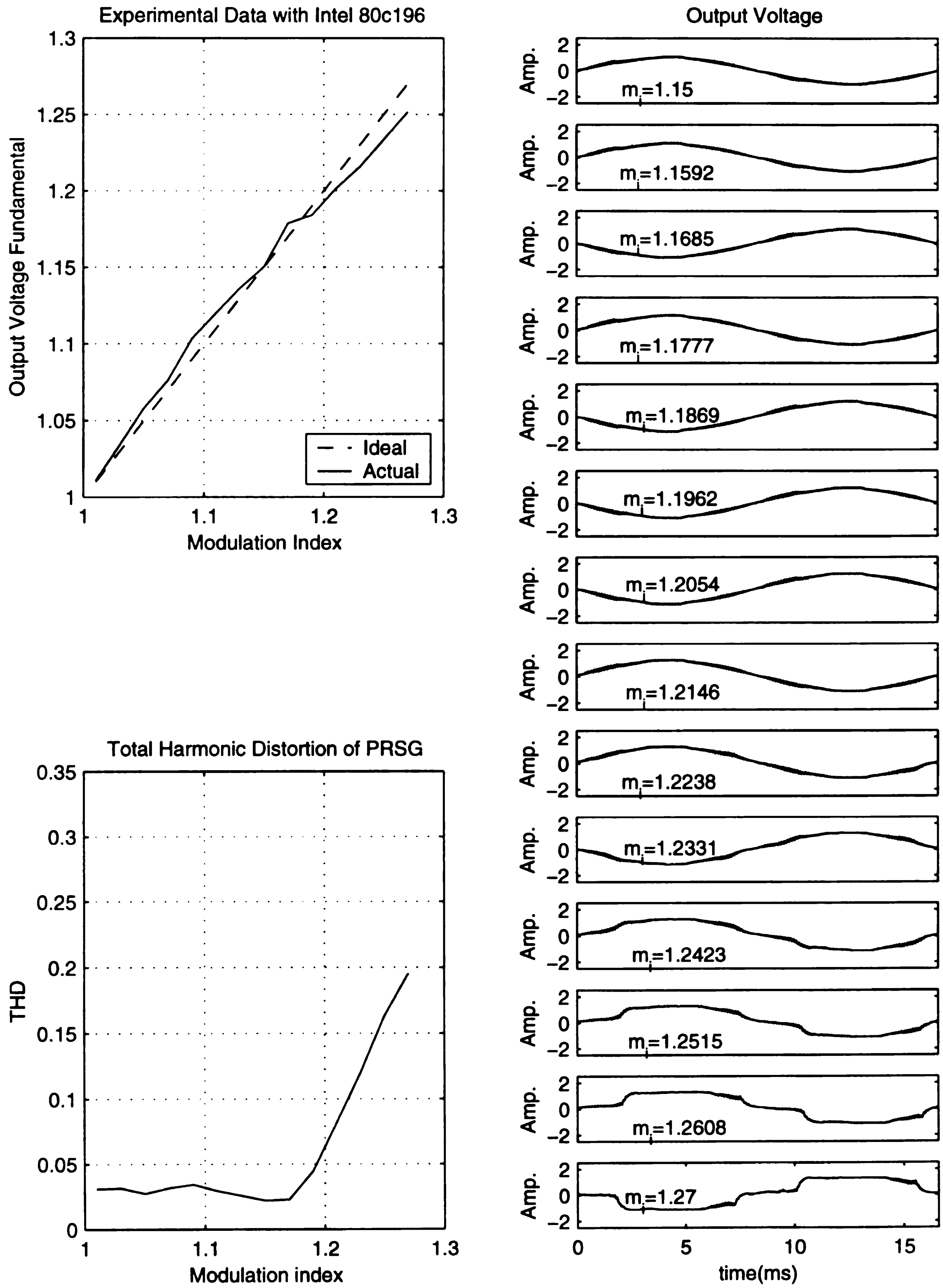


Figure 6.5. PRSG-modified experimental results using approximate m_i.

CHAPTER 7

VHDL Implementation

The Visic Hardware Description Language (VHDL) is a computer language that allows designers to describe and model digital systems at different levels of hierarchy. It is possible for example, to define components based on simple gates. Then, those components can be called and used by just knowing their input and output requirements and the function they perform. That abstraction capability allows designers to build complex digital systems. Furthermore, each component can be defined using behavioral architecture, which is closer to the concept and specification of the component than an structural architecture.

In this chapter, a VHDL implementation of the new overmodulation method is described. The objective is to show that the new overmodulation method is simple enough to be implemented in a separate digital block, saving the cost of a microprocessor. The code in VHDL can be mapped in an FPGA or any other logic programmable device. The system is first described in a block diagram, as it was in the simulation chapter. However, each block is now described based on the logic components needed to accomplish its function. The complete implementation is done using VHDL code. First, a simulation of the complete system using behavioral architecture is done. Then, this code is separated into procedures and functions in order to identify the different blocks. Finally, each procedure is converted to components and simulated again. Some of these components are then described in structural architecture using standard libraries. The rest of them, which require more mathematical calculations, are converted to behavioral architecture that can be synthesized to an FPGA using available packages.

7.1 General Specification

The digital system to be implemented in VHDL is shown in Figure 7.1 and has the following specifications:

- two 8-bit digital input ports for v_d and v_q;
- 3 single-bit digital output for inverter main devices switching state;
- sampling rate of 10kHz (v_d and v_q are read);
- 2MHz frequency clock for triangular generator.

The function of this overmodulation digital system can be divided into two major sub-functions. The first executes the mathematical algorithm to obtain the three new reference signal voltages (8 bits wide) based on the 2 input digital vectors. This part of the system requires multiplication and addition blocks in order to be implemented. The second part converts the eight-bit information to time (PWM). This part consists of counters, comparators and a decoder. This part can be mapped directly using structural architecture since those components already exist in libraries. Figure 7.2 shows these two blocks. The first block is called the calculation block, while the second is called the timing block.

The calculation block has to realize three functions. The first is a 2-to-3 transformation, converting v_d and v_q to the instantaneous three reference voltages v_a, v_b and v_c. The second function is to calculate the normalized modulation index and the normalization constant (m_{in}, $Norm$) to be used in the mixer. The third function is to

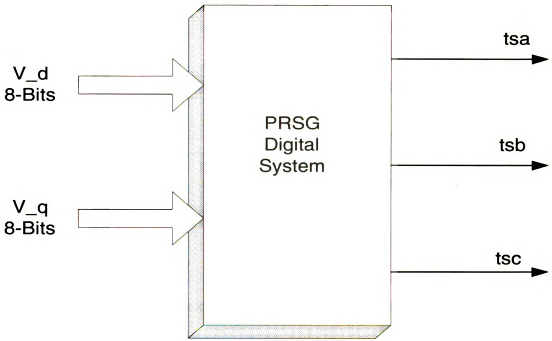


Figure 7.1. Digital system main block.

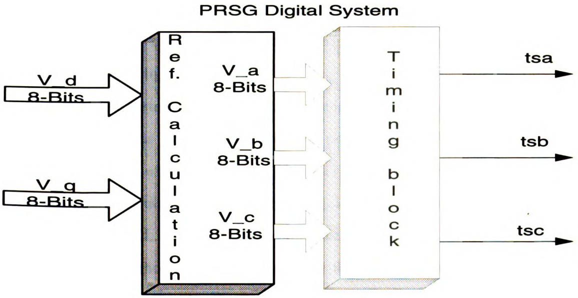


Figure 7.2. Calculation and timing blocks.

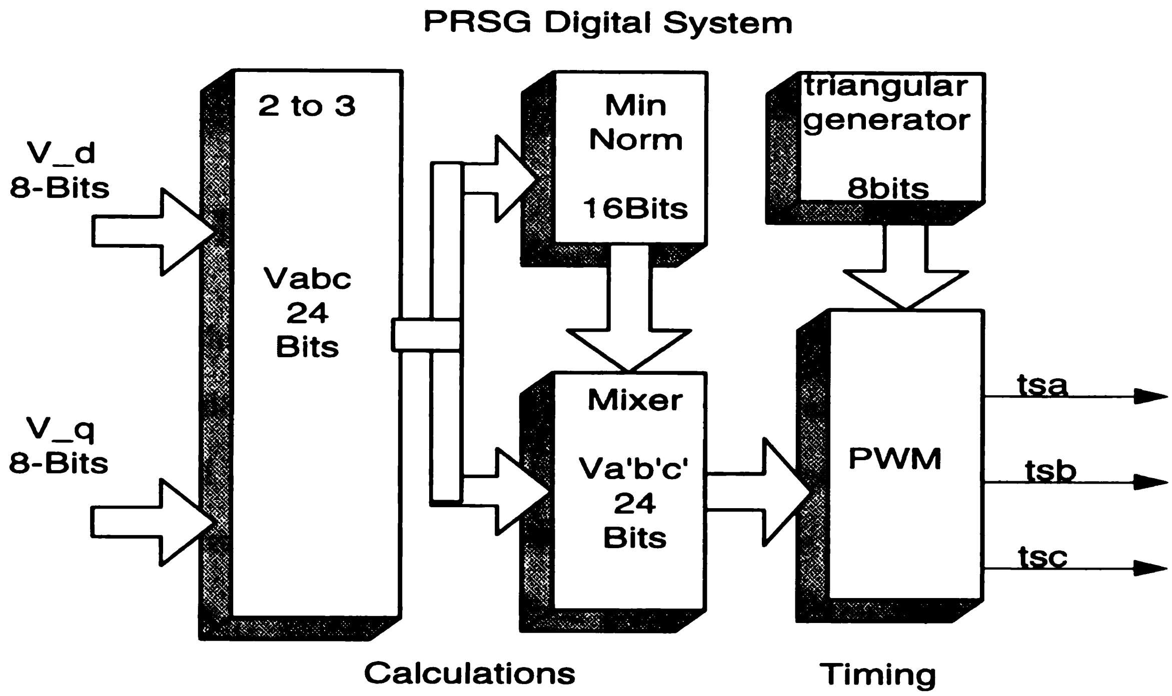


Figure 7.3. Subsystem block diagram.

mix the sinusoidal reference signal with its respective square signal using the instantaneous voltage and the constants previously calculated. The timing block consists of two parts: the triangular generator which includes counters, and the comparator block, which consists on three eight-bit comparators. Figure 7.3 shows these five main parts of the PRSG digital system.

The complete system was first simulated using a single component in VHDL with two 8-bit input ports and 3 single-bit output ports. The five blocks were coded as VHDL procedures, which were stored in a user package. The procedures that involve the 2-to-3 calculation used directly multiplication and division functions, which are allowed in VHDL. This first architecture, although difficult to synthesize, shows that the PRSG method is easier to implement than previous methods, since VHDL does not allow the use of more complex mathematical functions like direct and inverse trigonometric functions. This first code is also useful to verify if the selected frequencies and word length provide the required resolution to produce a good result. Although is not practical to show the complete VHDL code (including the five procedures), the entity and architecture are described in Figure 7.4.

7.2 Test Procedure

In order to test this behavioral architecture of the PRSG-PWM digital system, a test bench VHDL code was used. This code reads data for v_d and v_q from an ASCII file. This file, created using a Matlab code, contains the reference voltage for 10 cycles. Each cycle consists of 83 points, with values in the range of (-127,+127). In order to simulate data that actually come from a PC or DSP, this data is saved in 2's complement form. Therefore the ASCII file contains values from 0 to 255. This data is read with a sampling frequency of 10kHz.

The test-bench file converts these data into bit_vectors that are mapped to the input ports of the *pwmchip* module. The output bit ports of the *pwmchip* module (swa, swb, swc) are also mapped to a three-bit signal, which is then converted to a single integer (0-7), using the data conversion function from the user package AD_pack. This was necessary, because of the large amount of data that is generated during the simulation. The *pwmchip* model runs one process each $500\mu s$. This time corresponds to the clock of the triangular wave generator. During this process all five procedures described above are executed. The five procedures are coded and compiled in the file AD_pack.vhd. The procedure to test the behavioral architecture of the PRSG-PWM digital system is shown in Figure 7.5.

7.2.1 PRSG Using Modified Reference Signal

The equations for the 2-to-3 transformation are the same as used in the simulation and microprocessor implementation. However, the equations used to calculate the

```
ENTITY pwmchip IS
PORT (Vd,Vq : in std_logic_vector(7 DOWNTO
0);
    swa,swb,swc : out std_logic);
END pwmchip;

ARCHITECTURE chipv1 OF pwmchip IS
SIGNAL swai,swbi,swci: std_logic;
BEGIN
PROCESS
VARIABLE triang: INTEGER:=100;
VARIABLE inc: INTEGER:=-1;
VARIABLE Var,Vbr,Vcr: INTEGER;
VARIABLE Va,Vb,Vc: INTEGER;
VARIABLE Mi,Norm: INTEGER;
BEGIN

-- 2 to 3 transformation
conv2to3(Vd,Vq,Var,Vbr,Vcr);

-- modulation index calculation
minorm(Var,Vbr,Vcr,Mi,Norm);

-- mix square with sinusoidal
mixer(Var,Vbr,Vcr,Mi,Norm,Offset,Va,Vb,Vc);

-- Triangular generator
tri_wave(triang,inc,triang,inc);

-- PWM compare triang with vref
PWM(Va,Vb,Vc,triang,swai,swbi,swci);

swa <= swai;
swb <= swbi;
swc <= swci;

WAIT FOR 500 NS;

END PROCESS;
END chipv1;
```

Figure 7.4. Partial VHDL code for PRSG digital system.

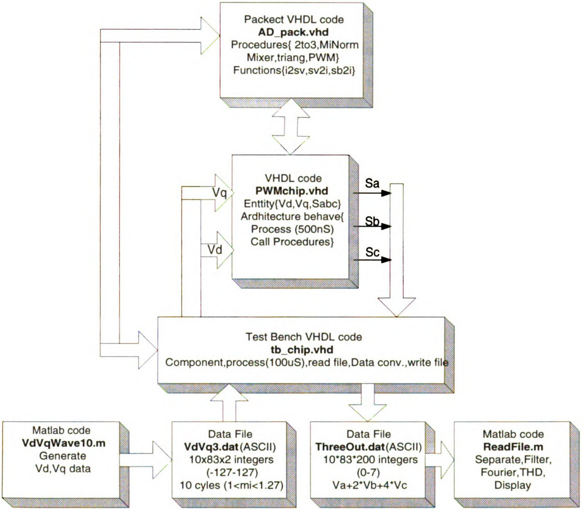


Figure 7.5. Test procedure for PRSG-PWM digital system.

overmodulation index, the normalization factor, and the mixer are simplified. Since a multiplication involves a lot of resources when it is implemented in hardware, all the multiplications are moved from the mixer to the modulation index procedure. In the mixer, each multiplication has to be performed three times (one for each phase), while in the *minorm* procedure is calculated only once. The normalization factor, that was used to convert the instantaneous reference values (v_a, v_b, v_c) into unitary values (v_{au}, v_{bu}, v_{cu}), is now transformed to one constant that includes mixer factors. The way this constant is transformed is detailed in the following equations. These equations belong to the mixer that produces the reference signal v_{ra} from the signal v_a that comes from the 2-to-3 transformation.

$$v_{ra} = \frac{m_i - 1}{.27}(sign(v_{au}) - v_{au}) + v_{au} \tag{7.1}$$

$$v_{ra} = \frac{m_i - 1}{.27}(sign(v_{au}) + \frac{1.27 - m_i}{.27}v_{au} \tag{7.2}$$

Since $v_{au} = v_a/m_i$ and $sign(v_{au}) = sign(v_a)$

$$v_{ra} = \frac{m_i - 1}{.27}sign(v_a) - \frac{1.27 - m_i}{.27m_i}v_a. \tag{7.3}$$

For the necessary scaling for fixed point multiplication

$$v_{ra} = \frac{(m_i - 100) * 100}{27}sign(v_a) + \frac{32 * (127 - m_i) * 100}{m_i * 27}\frac{v_a}{32}. \tag{7.4}$$

The first 100 constant in equation (7.4) corresponds to $m_i = 1$, while the second eliminates a multiplication in the mixer. In the second term, the 100 constant is used to scale before the division by m_i. The factor 32 is a power of two that does the scaling before the division by 27.

Defining m_{in} and $Norm$ as

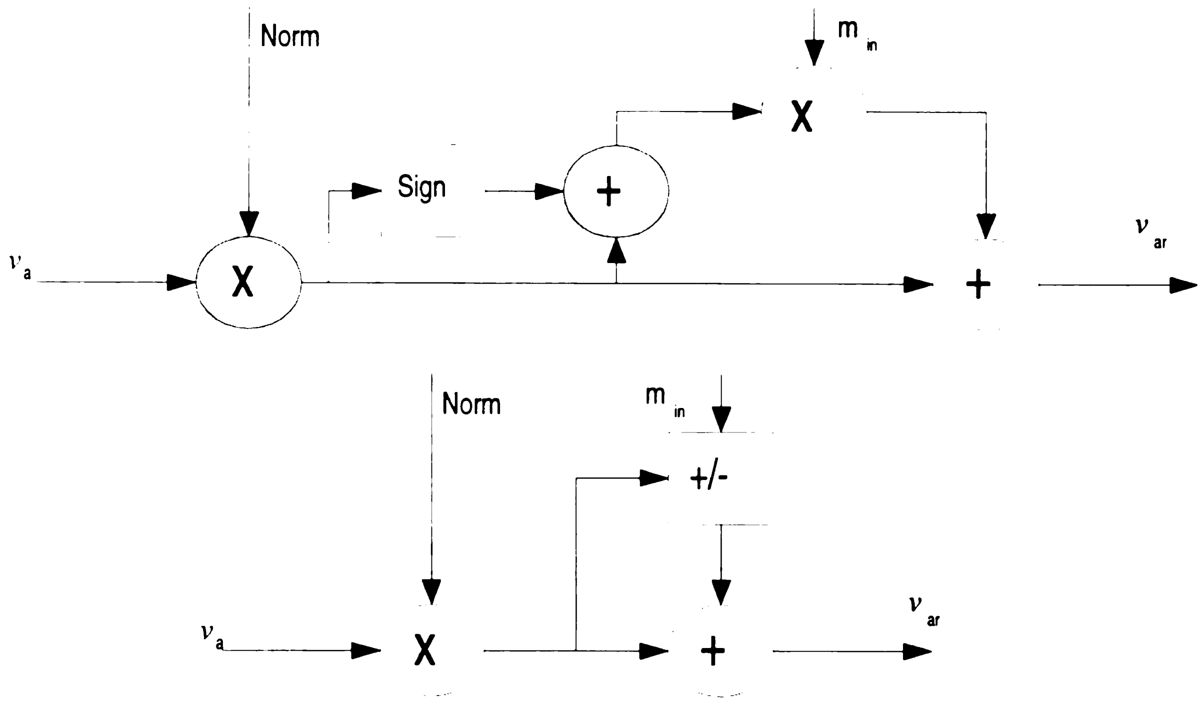


Figure 7.6. Mixer simplification.

$$m_{in} = \frac{(m_i - 100) * 100}{27} \quad \text{and} \quad Norm = \frac{32 * (127 - m_i) * 100}{m_i * 27}, \tag{7.5}$$

the equation for the user is reduced to

$$v_{ra} = m_{in} sign(v_a) + \frac{Norm\, v_a}{32}. \tag{7.6}$$

The range for M_{in} is (0-100), while the range for $Norm$ is (0-32). These factors eliminate one division, one multiplication and one subtraction from the mixer. Since these operations are performed three times in the mixer, 9 operations and 6 scalings are saved. Figure 7.6 shows this simplification.

The results for the simulation of this behavioral architecture for sinusoidal case is shown in Figure 7.7. The modulation index has been changed from 1 to 1.27 in steps of 0.03. The fundamental and harmonic distortion appear linear, but because of limitations (resolution) in the modulation index calculation some distortion is observed.

7.2.2 PRSG Using Modified Reference Signal

For the space vector modulation case the *minorm* procedure was modified. The new equations for m_i and $Norm$ are:

$$m_{in} = \frac{(m_i - 115) * 100}{12} \quad \text{and} \quad Norm = \frac{32 * (127 - m_i) * 115}{m_i * 12} \tag{7.7}$$

The simulation results for this case are shown in Figure 7.8. The harmonics distortion performance is better than the SVM case, as was expected. The output voltage was not distorted until a modulation index of 1.15 is reached. The fundamental appears linear over the overmodulation range.

7.2.3 PRSG Using Two-step Transformation

For two-step transformation, procedures *minorm* and *mixer* were modified. Two values of m_{in} (m_{in1} and m_{in2}) and two of $Norm$ ($Norm1$ and $Norm2$) are needed for the two stages. In the first modulation sub-range ($1.15 < m_i < 1.21$) the second mixer does not affect the reference signal and the equations for those four variables are

$$m_{in1} = \frac{(m_i - 115) * 128}{6} \quad \text{and} \quad Norm1 = \frac{32 * (121 - m_i) * 115}{m_i * 6} \tag{7.8}$$

$$m_{in2} = 0; \quad and \quad Norm2 = 32; \tag{7.9}$$

In the second modulation index sub-range ($1.21 < m_i < 1.27$) the first mixer

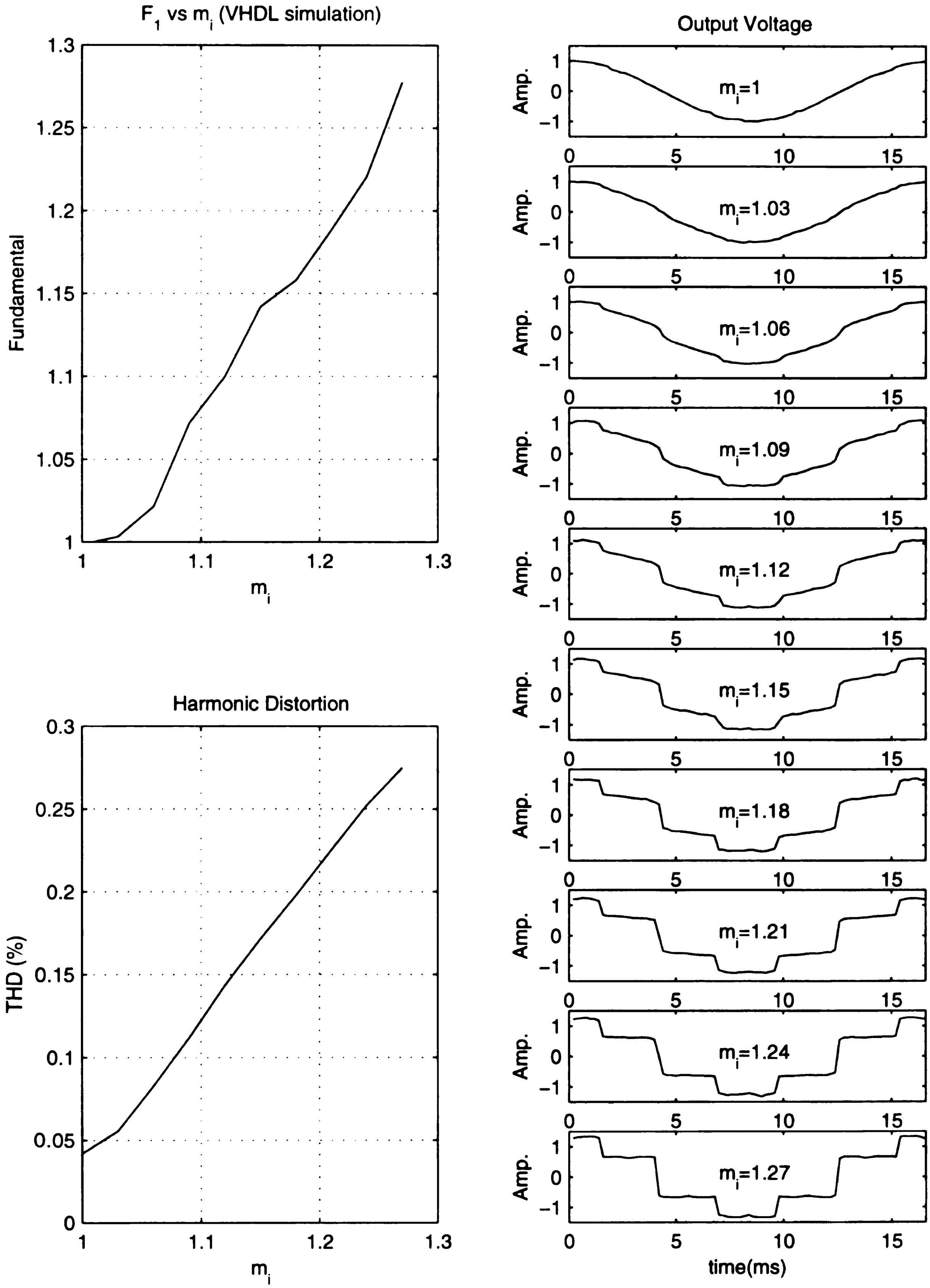
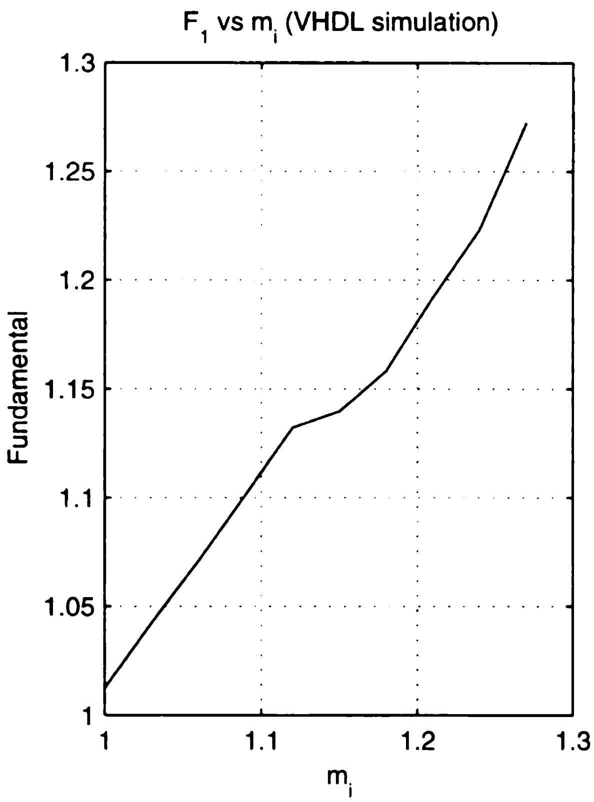
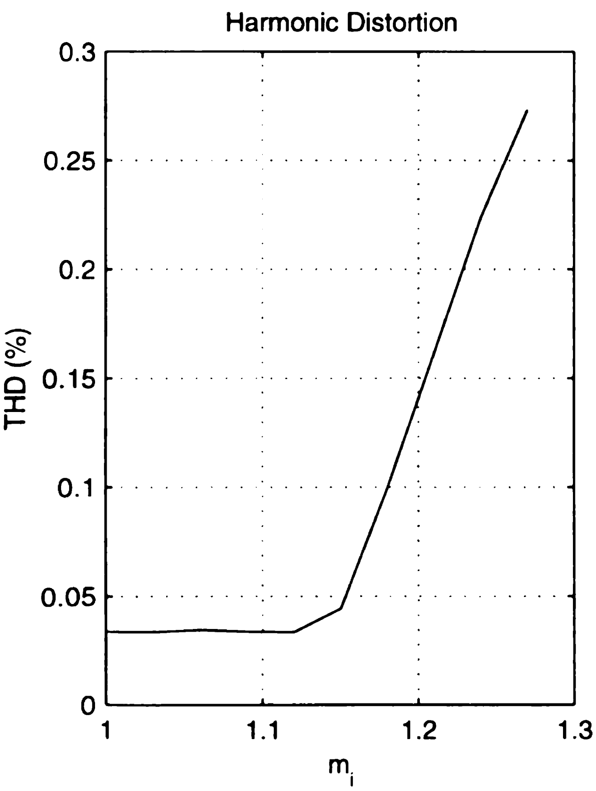


Figure 7.7. Simulation results for PRSG-unmodified.





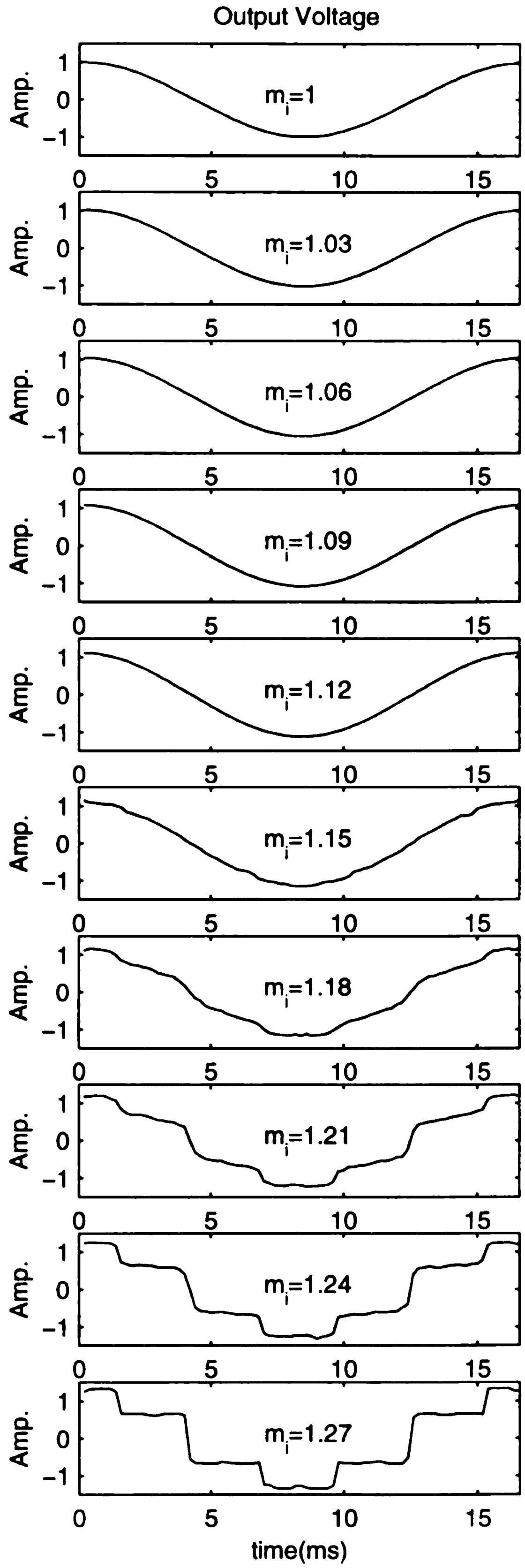


Figure 7.8. Simulation results PRSG-modified.

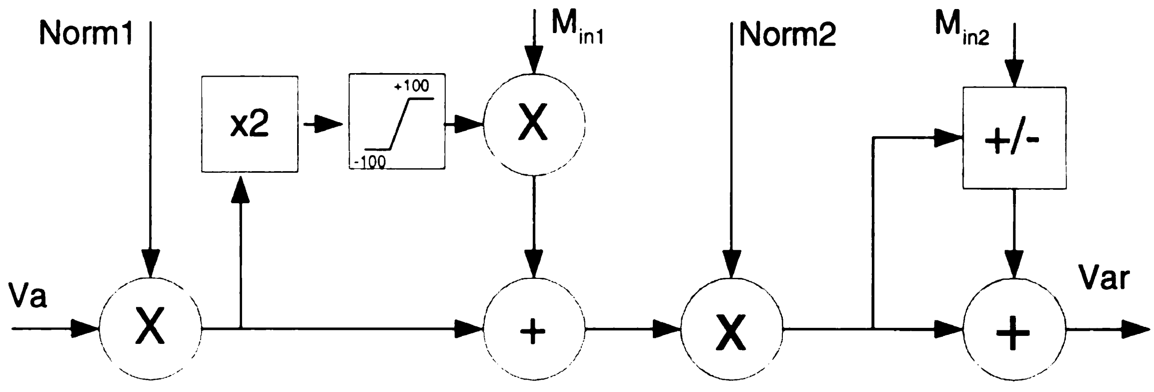


Figure 7.9. Mixer simplification for PRSG-two-step transformation.

should produce a trapezoidal unitary signal. Therefore, the equations for these four variables are changed to

$$m_{in1} = 100; \tag{7.10}$$

$$Norm1 = 0; \tag{7.11}$$

$$m_{in2} = \frac{(m_i - 121) * 100}{6} \tag{7.12}$$

$$Norm2 = \frac{32 * (127 - m_i)}{6}. \tag{7.13}$$

The simplification in the one-step case is used in the second sub-range of the two-step case mixer. For the creation of the trapezoidal waveform, more information is needed about the phase than for the creation of the square wave, and therefore one multiplication has to be left in the mixer for the first subrange. Figure 7.9 shows the simplification of the mixer for the two-step case. The simulation results for this case are shown in Figure 7.10.

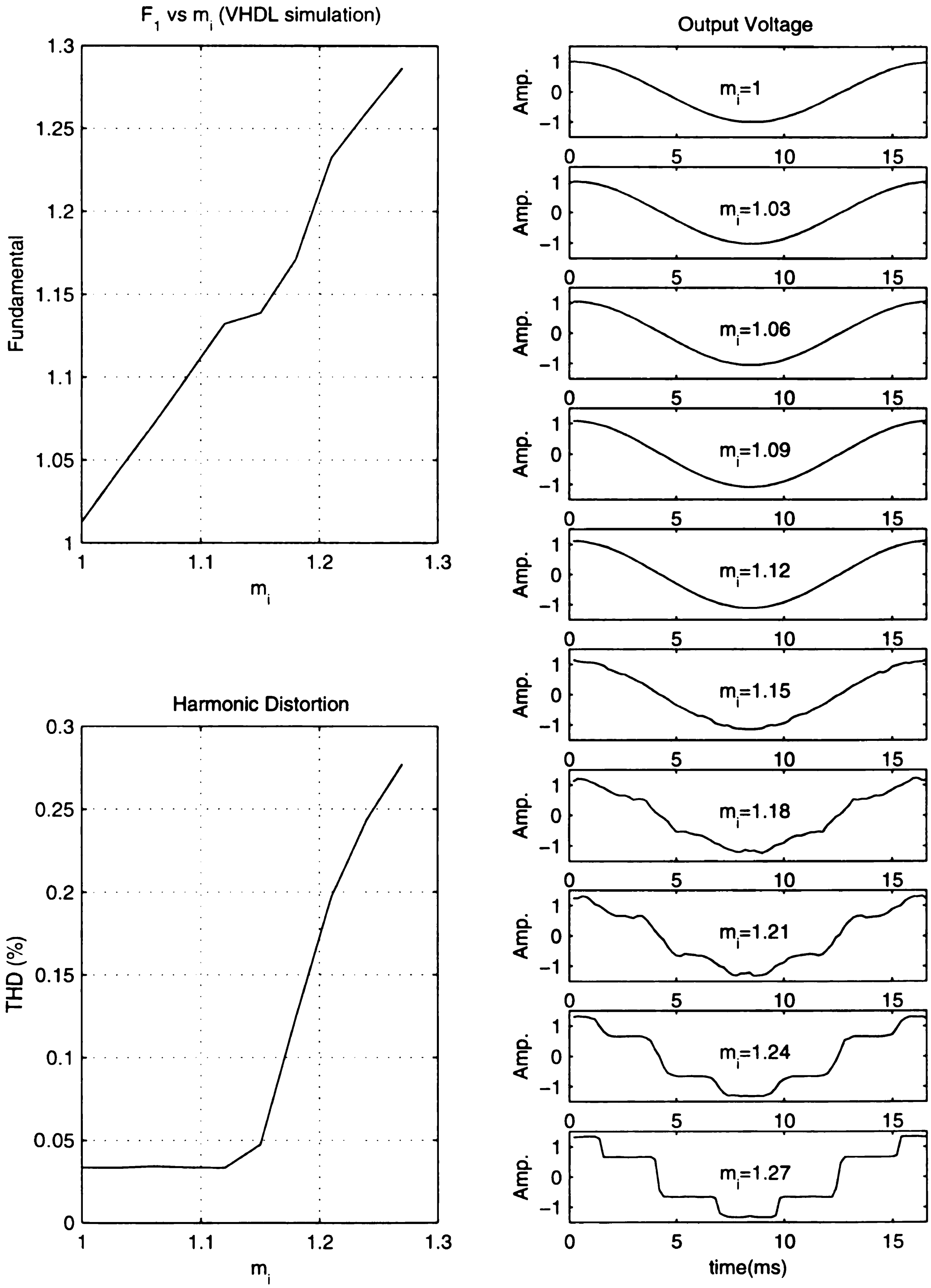


Figure 7.10. Simulation results for PRSG-two-step.

7.3 Analysis of Results

The test results for the VHDL architecture presented in this chapter shows that the proposed design is suitable for implementing the PRSG-PWM in a separate digital block, in terms of resolution and output quality. The compiler used for the simulation was BlueHDL from Blue Pacific Computing, Inc. The timing block was described in a structural architecture using standard components. The triangular generator was implemented using two bidirectional 4-bit counters and logic gates to limit the count from 27 to 227. The PWM blocks were described using standard 4-bit comparator (2 for each phase) and the output was latched using flip-flops. The calculation blocks remain in an behavioral architecture. However, signed multiplication and division, were converted to a code that is possible to be synthesized for available package as Xilinx or Altera. The implementation of this design in VHDL makes it more portable to be mapped using different technologies and different software packages, than if it was implemented using schematic entry.

CHAPTER 8

Conclusions

In order to obtain a better transient response and maximum torque from an induction motor-drive system, the maximum fundamental output voltage should be obtained from any given DC source. This maximum voltage is obtained when an overmodulation method is used to extend the normal PWM method until it the reaches six-step mode. In this thesis, it is shown that it is possible to make this extension using only linear equations. The advantage of simplicity over the previous methods allows this overmodulation method to be easily integrated in the overall PWM scheme. The proposed method, called Proportional Reference Signal Generator, is based on a linear interpolation between a sinusoidal reference signal and a square one. The key of the method is a sinusoidal-square mixer, which obtains the square and sinusoidal unitary waveforms from the instantaneous values of the v_d and v_q reference signals.

The modulation index, which is necessary for the interpolation, is also calculated from the instantaneous reference values of the voltages of the three phases, using a linear equation proposed in this work. The total harmonic distortion of the reference signal created by the PRSG, varies linearly with respect to the modulation index command. Since the optimal THD performance is not the one that behaves linearly, but rather the one which shows the lowest THD values for each modulation index, a two-step transformation is also proposed. This consists of converting the sinusoidal waveform into a trapezoidal one, and then converting the trapezoidal waveform into a square one as the value of the modulation index increases.

Mathematical analysis showed that the modified voltage reference signal, behaves linearly with respect to the modulation index. This analysis also showed that the mixer used in the PRSG can transition from any waveform into any other, while the fundamental varies linearly from the fundamental of the first signal to the fundamental of the other. That means that PRSG works perfectly either for sinusoidal PWM, space vector PWM or any other transformation as used in the two-step case (sinusoidal-trapezoidal and trapezoidal-square). The linearity of the THD with respect to the overmodulation index command was also shown in this analysis. Finally, an optimization for the constants, used in the modulation index, proposed equations was performed. Simulations using the Simplorer and Matlab packages verified the linear behavior of the proposed method.

The new method was described in block diagrams, state diagrams and equations. The data was analyzed using the Matlab package. For both cases, unmodified and modified reference signal, the fundamental of the output varies linearly with the reference, as was predicted in the mathematical analysis. The two-step transformation case was simulated and compared with the classic overmodulation methods Mode I and Mode II. Although linearity and the simplicity of implementation were a clear advantage in the proposed method, it was important to compare its THD to that of the overmodulation method consisting of Mode I and Mode II, since this last method has shown the better result so far. The test results show that the PRSG (two-step transformation case) has a lower THD than Mode I and Mode II in the second subrange of $1.21 < m_i < 1.27$.

The proposed overmodulation method was implemented in a microcontroller (Intel 80C196) and was used to control a three-phase voltage inverter. The results of this experiment verify the linearity and simplicity of the PRSG method. In the experimental setup the difference between the calculated modulation index and the actual modulation index was larger than predicted by simulation. Also, the maximum fundamental obtained with the calculated modulation index was 1.256, which is very close to the theoretical maximum of 1.27. This difference is caused by two different factors: the limited numerical resolution of the calculations inside the microprocessor which amplified the ripple of the modulation index, and inability to produce an average error of zero at the extremes of the voltage range.

The simplicity of the new method can be exploited by implementing it in a stand alone device like an FPGA. A digital system was designed, using the VHDL language. Some equations were modified in order to minimize the number of numerical operations and enhance the performance of the system. The test results show that the new overmodulation method is suitable to be implemented in a stand-alone system.

The proposed PRSG method is suitable to be used in vector control, since the reference signals are calculated from the instantaneous reference values of v_d and v_q. Additionallly this new method uses variables calculated for other methods in the linear range, being suitable to be easily integrated with them. The results obtained with the PRSG methods suggests that this research extends into the overmodulation range the simplicity obtained recently in the linear range, with the unification of the space vector theory and classical triangular comparison methods.

8.1 Future Work

A number of tasks can be carried out to further analyze, enhance and apply the proposed overmodulation technique. These include:

1. Convert the VHDL design into a real stand-alone device. This device could be integrated with the inverter, simplifying the control setup.

2. Investigate the ripple torque caused by the overmodulation method as the mod-

ulation index approach 1.27. All overmodulation methods are based on a complete cycle. This means that the fundamental is calculated over an entire waveform cycle. In vector control, if the angle applied differs from the desired ones, reverse torque could be produced in the motor. A new overmodulation method that alleviates this effect can be derived from this investigation.

3. Investigate analytically the optimal path for THD of a transformation from sinusoidal to square wave. Then, the actual method including PRSG can be compared with it.

4. Find an algorithm to avoid the tedious maximum and minimum calculations in the space vector PWM. Although the unified approach is simpler than the original approach for space vector modulation, the implementation in software of the maximum and minimum of the three phases values takes approximately 12 instructions in C Language and 17 in the 80C196 assembler language.

5. Investigate the possibility of applying the overmodulation method directly from references v_d and v_q. If the overmodulation is based on the reference signals, v_d and v_q instead of v_a, v_b and v_c, 50% of the calculations are saved.

APPENDICES

APPENDIX A

VHDL code for AD-pack

```
------------------------------------------------------------------
-- File:        AD_pack.vhd                                     --
-- Description: A generic package used by the design "PWMchip"  --
-- Author:      Andres Diaz                                     --
-- DOC:         July 24, 2000                                   --
--      (c) Copyright 1999-00 Custom Solution Group, Inc.       --
--              ALL RIGHTS RESERVED                             --
------------------------------------------------------------------

library ieee;
use ieee.std_logic_1164.all;

package AD_pack is
  function sv2i (sv: std_logic_vector) return integer;
  function sb2i (sv: std_logic_vector) return integer;
  function i2sv (integ: integer) return std_logic_vector;
  procedure conv2to3 (Vd,Vq: in std_logic_vector( 7 downto 0);
                  vra,vrb,vrc: out integer);
  procedure tri_wave(triang,inc: in integer; tri,incre: out integer);
  procedure PWM(Var,Vbr,Vcr,triang: in integer;
              signal swa,swb,swc: out std_logic);
  procedure minorm(Var,Vbr,Vcr: in integer; Mi,Norm,Offset:
                 out integer);
  procedure mixer(Var,Vbr,Vcr,Mi,Norm,Offset:in integer; Vao,Vbo,Vco:
                out integer);
  procedure mixer3(Var,Vbr,Vcr,Mi1,Norm1,Mi2,Norm2,Offset:in integer;
                 Vao,Vbo,Vco: out integer);
  procedure minorm2(Var,Vbr,Vcr: in integer; Mi,Norm,Offset:
                  out integer);
  procedure minorm3 (Var,Vbr,Vcr: in integer;
                 Mi1,Norm1,Mi2,Norm2,Offset: out integer);
end AD_pack;

package body AD_pack is
```

```
function sv2i (sv: std_logic_vector) return integer is
  variable result, num: integer;
begin
  result := 0; num := 1;
  for i in 0 to sv'length-1 loop
    if (sv(i) /= '0') and (sv(i) /= '1') then
  return -1;
    elsif (sv(i) = '1') then
  result := result + num;
    end if;
    num := num * 2;
  end loop;
  return result;
end sv2i;

 function sb2i (sv: std_logic_vector) return integer is
  variable result, num: integer;
begin
  result := 0; num := 1;
  for i in 0 to sv'length-1 loop
    if (sv(i) /= '0') and (sv(i) /= '1') then
  return -1;
    elsif (sv(i) = '1') then
  result := result + num;
    end if;
    num := num * 2;
  end loop;
  if result > 127 then
    result:=result-256;
  end if;
  return result;
end sv2i;

 function i2sv (integ: integer) return std_logic_vector is
  variable result: std_logic_vector(7 downto 0);
  variable num: integer;
begin
  num := integ;
  for i in 0 to 7 loop
    if num mod 2 = 0 then
      result(i) := '0';
    else result(i) := '1';
    end if;
    num := num / 2;
```

```
    end loop;
    return result;
  end i2sv;

-- Convert from vd,vq to va,vb,vc
procedure conv2to3 (Vd,Vq: in std_logic_vector(7 downto 0);
                    vra,vrb,vrc: out integer) is
variable Vdi,Vqi: integer;
begin
Vdi := sB2i(Vd);
Vqi := sB2i(Vq);
vra :=  Vdi;
vrb := -(Vdi/2)+((86*Vqi)/100);
vrc := -(Vdi/2)-((86*Vqi)/100);
end conv2to3;

-- Triangular generator
procedure tri_wave(triang, inc: in integer; tri,incre: out integer) is
variable t1,i1 : integer;
BEGIN
IF triang >= 227 THEN
    i1 := -1;
ELSIF triang <= 27 THEN
    i1 := 1;
else
    i1 := inc;
END IF;
t1 := triang + i1;
tri := t1;
incre := i1;
END tri_wave;

-- compare vref with triangular wave
procedure PWM(Var,Vbr,Vcr,triang: in integer;
              signal swa,swb,swc: out std_logic) is
begin
IF Var > triang THEN
    swa <= '1';
ELSE
    swa <= '0';
END IF;
IF Vbr > triang THEN
    swb <= '1';
ELSE
    swb <= '0';
```

```
END IF;
IF Vcr > triang THEN
    swc <= '1';
ELSE
    swc <= '0';
END IF;
end PWM;

procedure minorm(Var,Vbr,Vcr: in integer; Mi,Norm,Offset:
                 out integer) is
variable v1,v2,v3,Mit: integer;
variable flag: bit:=1;
begin
if Var < Vbr then
  v1:=Var;
  v2:=Vbr;
else
  v2:= Var;
  v1:= Vbr;
end if;
if Vcr < v1 then
  v3:= v2;
  v2:= v1;
  v1:= Vcr;
elsif Vcr < v2 then
  v3:= v2;
  v2:= Vcr;
else
  v3:=Vcr;
end if;
if v2>0 then
  v1 := v3;
end if;

Mit:=abs(83*v2+115*v1)/100;
if Mit < 100 then
  Mi := 0;
  Norm := 100;
else
  Norm:=(((32*(127-Mit))/27)*100)/Mit;
  Mi := ((Mit-100)*100)/27;
end if;
Offset:=0;
end minorm;
```

```
procedure mixer(Var,Vbr,Vcr,Mi,Norm,Offset:in integer; Vao,Vbo,
                Vco:out integer) is
variable Va,Vb,Vc,Mit: integer;
 begin
 Va:=Var+Offset;
 Vb:=Vbr+Offset;
 Vc:=Vcr+Offset;

 if Mi /= 0 then
   Mit:=Mi;
   if Va > 0 then
    Va:=(Norm*Va)/32+Mit;
   elsif Va < 0 then
    Va:=(Norm*Va)/32-Mit;
   else
    Va:=0;
   end if;

   if Vb > 0 then
    Vb:=(Norm*Vb)/32+Mit;
   elsif Vb < 0 then
    Vb:=(Norm*Vb)/32-Mit;
   else
   Vb:=0;
   end if;

   if Vc > 0 then
    Vc:=(Norm*Vc)/32+Mit;
   elsif Vc < 0 then
    Vc:=(Norm*Vc)/32-Mit;
   else
    Vc:=0;
   end if;
 end if;

   Vao:=Va+127;
   Vbo:=Vb+127;
   Vco:=Vc+127;
end mixer;

procedure minorm2(Var,Vbr,Vcr: in integer; Mi,Norm,Offset:
                  out integer) is
variable v1,v2,v3,Mit: integer;
variable flag: bit:=1;
begin
```

```
if Var < Vbr then
  v1:=Var;
  v2:=Vbr;
else
  v2:= Var;
  v1:= Vbr;
end if;
if Vcr < v1 then
  v3:= v2;
  v2:= v1;
  v1:= Vcr;
elsif Vcr < v2 then
  v3:= v2;
  v2:= Vcr;
else
  v3:=Vcr;
end if;

if v2>0 then
  v1 := v3;
end if;

Mit:=abs(83*v2+115*v1)/100;

if Mit < 116 then
  Mi := 0;
  Norm := 100;
else
  Norm:=(((32*(127-Mit))/12)*115)/Mit;
  Mi := ((Mit-115)*100)/12;
end if;

Offset:=v2/2;

end minorm2;

procedure minorm3 (Var,Vbr,Vcr: in integer;
                   Mi1,Norm1,Mi2,Norm2,Offset: out integer) is
variable v1,v2,v3,Mit: integer;
variable flag: bit:=1;
begin
if Var < Vbr then
  v1:=Var;
  v2:=Vbr;
else
```

```
  v2:= Var;
  v1:= Vbr;
end if;
if Vcr < v1 then
  v3:= v2;
  v2:= v1;
  v1:= Vcr;
elsif Vcr < v2 then
  v3:= v2;
  v2:= Vcr;
else
  v3:=Vcr;
end if;

if v2>0 then
  v1 := v3;
end if;

Mit:=abs(83*v2+115*v1)/100;

if Mit < 116 then
  Mi1 := 0;
  Mi2 := 0;
  Norm1 := 32;
  Norm2 := 32;
 elsif Mit < 122 then
  Norm1:=(((32*(121-Mit))/6)*115)/Mit;
  Mi1 := ((Mit-115)*128)/6;
  Mi2 :=0;
  Norm2 := 32;
 else
  Norm1:=0;
  Mi1 := 128;
  Mi2 :=((Mit-121)*100)/6;
  Norm2 := (32*(127-Mit))/6;
 end if;


Offset:=v2/2;

end minorm3;

procedure mixer3(Var,Vbr,Vcr,Mi1,Norm1,Mi2,Norm2,Offset:in integer;
                Vao,Vbo,Vco:out integer) is
```

```
variable Va,Vb,Vc: integer;
variable tra,trb,trc: integer;
 begin
 Va:=Var+Offset;
 Vb:=Vbr+Offset;
 Vc:=Vcr+Offset;

 if Mi1 /= 0 then
    tra:=2*Va;                          -- Trapezoidal Generation
    if tra > 100 then
      tra:=100;
    elsif tra < -100 then
      tra:=-100;
    end if;
    Va:=(Norm1*Va)/32+(Mi1*tra)/128;    -- First transformation
    if Va > 0 then
     Va:=(Norm2*Va)/32+Mi2;             -- second transformation
    elsif Va < 0 then
     Va:=(Norm2*Va)/32-Mi2;
    end if;

    trb:=2*Vb;                          -- Trapezoidal Generation
    if trb > 100 then
      trb:=100;
    elsif trb < -100 then
      trb:=-100;
    end if;
    Vb:=(Norm1*Vb)/32+(Mi1*trb)/128;   -- First transformation
    if Vb > 0 then
     Vb:=(Norm2*Vb)/32+Mi2;            -- second transformation
    elsif Vb < 0 then
     Vb:=(Norm2*Vb)/32-Mi2;
    end if;

    trc:=2*Vc;                          -- Trapezoidal Generation
    if trc > 100 then
      trc:=100;
    elsif trc < -100 then
      trc:=-100;
    end if;
    Vc:=(Norm1*Vc)/32+(Mi1*trc)/128;   -- First transformation
    if Vc > 0 then
     Vc:=(Norm2*Vc)/32+Mi2;            -- second transformation
    elsif Vc < 0 then
     Vc:=(Norm2*Vc)/32-Mi2;
```

```
    end if;

  end if;

   Vao:=Va+127;
   Vbo:=Vb+127;
   Vco:=Vc+127;

end mixer3;
```

end AD_pack;

APPENDIX B

VHDL code for test bench

```
-- test ADpackage
-- Andres Diaz Jul,9,2000

library ieee;
use ieee.std_logic_1164.all;
use std.textio.all;
use ieee.std_logic_arith.all;
use work.AD_pack.all;


ENTITY testchip IS
END testchip;

ARCHITECTURE testchip_0 OF testchip IS

COMPONENT pwmchip IS
PORT (Vd,Vq: in std_logic_vector(7 downto 0);
      swa,swb,swc: out std_logic);
END COMPONENT;

SIGNAL  vdg,vqg: std_logic_vector(7 DOWNTO 0);
SIGNAL  ta,tb,tc: std_logic;
SIGNAL  FLAG: BIT :=0;
BEGIN
    uut: pwmchip
    port map (Vd => vdg, Vq => vqg, swa => ta, swb => tb, swc => tc);

   PROCESS
    VARIABLE v1,v2:std_logic_vector(7 DOWNTO 0);
    VARIABLE vdi,vqi:INTEGER;
    VARIABLE BUFF:line;
    FILE dataint: text OPEN READ_MODE IS "vdvq3.dat";
        BEGIN
```

```
        READLINE(dataint,buff);
        READ(buff,vdi);
            v1:=i2sv(vdi);    -- convert integer to binary vector
            vdg <= v1;
        READ(buff,vqi);
            v2:=i2sv(vqi);    -- convert integer to binary vector
            vqg <= v2;


   WAIT FOR 100 US;

END PROCESS;
END testchip_0;
```

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] A. R. Balhshai, G. Joos, and H. Jin. Incorporating the overmodulation range in space vector pattern generators using a classification algorithm. *IEEE Transactions on Power Electronics*, 13:1007–1013, 1995.

[2] V. Blasco. Analysis of a hybrid pwm based on modified space-vector and triangle-comparison methods. *IEEE Transactions on Industrial Applications*, 33(3):756–764, May/June 1997.

[3] D. W. Cheng, J. S. King, and S. K. Sul. Unified voltage modulation technique for real-time three-phase power conversion. *IEEE Transactions on Industrial Applications*, 34(2):374–380, March/April 1997.

[4] A. Diaz and E. Strangas. New wide range overmodulation method. *IEEE Applied Power Electronics Conference*, pages 1276–1280, Febuary 2000.

[5] J. Holtz, Wolfang Lotzkat, and A. M. Khambadkone. On continuous control of pwm inverters in the overmodulation range including the six-step mode. *IEEE Transactions on Power Electronics*, 8(4), October 1993.

[6] J. A. Houldsworth and D. A. Grant. The use of harmonic distortion to increase the output voltage of a three-phase pwm inverter. *IEEE Transactions on Industrial Applications*, vol. IA-20(2):1124–1127, Sept./Oct 1984.

[7] V. Kaura. A new method to linearize any triangle comparison based pwm by reshaping the modulation command. *IEEE Industrial Application Meeting*, 33(5):1254–1256, May 1996.

[8] V. Kaura and Blasko V. A new method to extend linearity of a sinusoidal pwm in the overmodulation region. *IEEE Transactions on Industrial Applications*, 32(5), Sep./Oct. 1996.

[9] R. J. Kerman, T.M Rowan, D.Leggate, and B. J. Seibel. Control of pwm voltage inverters in the pulse dropping region. *IEEE Transactions on Industrial Electronics.*, 43(1):132–141, 1995.

[10] D. C. Lee and G. M. Lee. A novel overmodulation technique for space vector pwm inverters. *IEEE Transactions on Power Electronics.*, 13:1014–1019, Nov 1998.