



INSTITUTO TECNOLÓGICO DE SONORA

**IMPLEMENTACIÓN DE LA TÉCNICA MODULACIÓN
DE VECTORES ESPACIALES APLICADA A UN
INVERSOR TRIFÁSICO DE TRES NIVELES CON
DIODO ANCLADO AL PUNTO NEUTRO**

TESIS

QUE PARA OBTENER EL TÍTULO DE

INGENIERO EN ELECTRÓNICA

PRESENTA:

OMAR LORETO BACA VALLE

CD. OBREGÓN, SONORA

JUNIO DE 2009

*A Dios
a mis padres Loreto y Manuelita
a mis hermanas Miroslava y Erika
a mi familia y amigos
a mi novia Sara Yael.*

AGRADECIMIENTOS

A Dios, por haberme dado vida y salud, por llenarme de bendiciones, por haberme dado fuerzas, entendimiento y sabiduría para sacar adelante mis estudios y permitirme culminar mi carrera universitaria.

A mis padres Loreto Baca Portillo y Manuelita Valle Carlón, por haber hecho de mí un hombre de bien, por la educación que me han dado, por el apoyo y el amor incondicional que día a día me dan, siempre les estaré agradecido.

A mis hermanas Miroslava Baca Valle y Erika Baca Valle, por su amor y amistad sincera, por sus valiosos consejos y por su apoyo incondicional, gracias por ser parte de mi vida.

A mi familia, a mis tíos y tías, a mis abuelas y a todos los que siempre me apoyaron a lo largo de mi carrera.

A mi novia Sara Yael Angulo Guzmán, que has estado conmigo en las buenas y en las malas, por tus buenos consejos, por tu amor incondicional, por los grandes y maravillosos momentos que hemos pasado juntos, gracias por ser parte de mi vida.

A mi compañero de tesis Emmanuel Amarillas García, a mis amigos Aurelio Rodríguez Echevarria, Jesús Ivan Navarro Figueroa y Cinthya María Verdugo Barceló, que siempre me apoyaron y me dieron buenos consejos.

A mis mejores amigos, Jesús Manuel Ayala Armenta, Jaime Omar Espinoza Rodriguez, Fernando López de Haro, Randy Borbón Sabory, Edgar Allan Gastélum Valdez, Gabriela Gámez Camberos, Delia Karina Chávez Vega y a todos los que siempre han estado conmigo en todo momento, les estoy eternamente agradecido.

Un especial agradecimiento al Dr. José Antonio Beristáin Jiménez, por todo su apoyo y por sus consejos durante el desarrollo del proyecto, por guiarnos por un buen camino y por ser un excelente profesor y un muy buen amigo.

Al Dr. Héctor Hernández y al Ing. Alejandro Aganza porque sus comentarios ayudaron a ser de ésta una mejor tesis.

También agradezco al Programa de Mejoramiento del Profesorado (PROMEP) que mediante el proyecto “Optimización de sistemas de energía renovable utilizando convertidores de potencia” apoyó con recursos para la realización de este trabajo.

Omar Loreto Baca Valle

ÍNDICE

	Página
LISTA DE FIGURAS	x
LISTA DE TABLAS	xv
LISTA DE ACRÓNIMOS Y ABREVIATURAS	xi
RESUMEN	xvii
CAPÍTULO I. Introducción	
1.1 Antecedentes teóricos.....	3
1.2 Planteamiento del problema.....	5
1.3 Objetivos.....	6
1.4 Justificación.....	6
1.5 Limitaciones.....	7
1.6 Delimitaciones.....	7
1.7 Alcances.....	8
Referencias.....	9
CAPÍTULO II. Marco teórico	
2.1 Inversores.....	11
2.2 Clasificación de los inversores.....	11
2.3 Clasificación de los inversores por su número de fases.....	11
2.3.1 Inversores monofásicos.....	11
2.3.2 Inversores trifásicos.....	12
2.4 Clasificación de los inversores por su número de niveles.....	13
2.4.1 Inversores de 2 niveles.....	13
2.4.1.1 Ventajas.....	14
2.4.1.2 Desventajas.....	14
2.4.1 Inversores multinivel.....	14

2.4.1.1 Ventajas.....	16
2.4.1.2 Desventajas.....	17
2.5 Topologías multinivel.....	17
2.5.1 Inversor multinivel con puentes H en cascada.....	17
2.5.1.1 Propiedades del inversor con puentes H en cascada.....	19
2.5.1.2 Ventajas.....	20
2.5.1.3 Desventajas.....	20
2.5.2 Inversor multinivel con capacitores flotantes.....	20
2.5.2.1 Propiedades de los inversores con capacitores flotantes.....	21
2.5.2.2 Ventajas.....	22
2.5.2.3 Desventajas.....	23
2.5.3 Inversor multinivel con diodo fijador (NPC).....	23
2.5.3.1 Principio de operación.....	24
2.5.3.2 Propiedades del inversor NPC.....	27
2.5.3.3 Ventajas.....	28
2.5.3.4 Desventajas.....	28
2.6 Características de inversores multinivel.....	28
2.7 Comparación de convertidores multinivel.....	29
2.8 Técnicas de modulación.....	30
2.9 Técnicas de modulación analógicas.....	31
2.10 Técnicas de modulación digitales.....	32
2.10.1 Modulación por vectores espaciales (SVM).....	32
2.10.2 SVM de 2 Niveles.....	33
2.10.3 SVM de 3 Niveles.....	33
2.10.3.1 Descripción General.....	34
2.10.3.2 Simetría.....	38
2.10.3.3 Estrategia para sintetizar el vector de referencia.....	38

2.10.3.4	Modulación SVM-NTV (<i>Nearest Three Vectors</i>).....	39
2.10.3.5	Cálculo de los ciclos de trabajo para los vectores de estado de tensión.....	39
2.10.3.6	Elección de los estados de conmutación.....	43
2.10.3.7	Una conexión a tensión cero.....	44
2.10.3.8	Dos conexiones a tensión cero.....	45
2.11	Fuentes de alimentación conmutada.....	45
2.11.1	Convertidor <i>Push – Pull</i>	46
2.12	Controlador digital de señales.....	47
2.12.1	La familia dsPIC30F.....	47
2.12.2	dsPIC30F4011.....	48
2.12.2.1	Oscilador.....	51
2.12.2.2	Módulo temporizador.....	53
2.12.2.3	Temporizador clase A.....	53
2.12.2.4	Temporizador clase B.....	54
2.12.2.5	Temporizador clase C.....	54
2.12.2.6	Registros de control del temporizador.....	55
2.12.2.7	Módulo ADC.....	56
2.12.2.8	Registros de control del ADC.....	57
2.13	MPLAB IDE.....	59
2.14	Lenguaje C.....	60
2.15	MPLAB C30.....	60
	Referencias.....	61

CAPÍTULO III. Desarrollo

3.1	Requerimientos.....	64
3.2	Descripción del sistema.....	64
3.3	Implementación.....	65
3.4	Inversor trifásico NPC de tres niveles.....	65
3.4.1	Elección de la topología.....	65

3.4.2 Diseño de la topología	66
3.4.3 Implementación del inversor.....	68
3.5 Fuente de alimentación <i>Push - Pull</i> para los dispositivos del inversor.....	69
3.6 Implementación digital.....	73
3.6.1 Elección del dispositivo a utilizar	74
3.6.2 Elección del lenguaje de programación.....	74
3.6.3 Elección del software de desarrollo y compilador del lenguaje.....	74
3.6.4 Desarrollo del programa.....	75
3.6.4.1 Declaración de las funciones globales.....	77
3.6.4.2 Declaración de las constantes y variables.....	78
3.6.4.3 Programa principal.....	80
3.6.4.4 Configuración de puertos, ADC, temporizadores y variables.....	80
3.6.4.5 Configuración de puertos.....	81
3.6.4.6 Configuración del ADC.....	81
3.6.4.7 Configuración de los temporizadores.....	82
3.6.4.8 Configuración de las variables.....	83
3.6.4.9 SVM.....	83
3.6.4.10 Magnitud.....	84
3.6.4.11 ADCRead.....	84
3.6.4.12 Índice de modulación.....	85
3.6.4.13 Posición.....	86
3.6.4.14 Calcula sector.....	90
3.6.4.15 Calcula triángulo.....	91
3.6.4.16 Duty cycle.....	92
3.6.4.17 Periodos.....	94
3.6.4.18 Resolución en grados y periodo de conmutación.....	94
3.6.4.19 Calcula los tres vectores más cercanos.....	96

3.6.4.20 Rutina de servicio a la interrupción.....	97
3.6.4.21 Rutina de servicio a la interrupción ISR TIMER1.....	97
3.6.4.22 Rutina de servicio a la interrupción ISR TIMER2.....	98
3.6.4.23 Rutina de servicio a la interrupción ISR TIMER3.....	99
3.6.4.24 Rutina de servicio a la interrupción ISR TIMER4.....	99
3.6.5 Implementación del circuito del DSC.....	100
3.7 Implementación de la técnica SVM aplicada a un inversor trifásico de tres niveles con diodo anclado al punto neutro.....	102
Referencias.....	103
CAPÍTULO IV. Pruebas y resultados	
4.1 Disparo de compuerta a fuente.....	105
4.2 Forma de onda de salida del inversor con carga resistiva.....	106
4.3 Forma de onda de salida del inversor con carga inductiva.....	112
4.4 Forma de onda de corriente de salida del inversor con carga resistiva.....	116
4.5 Corriente en el neutro.....	118
4.6 Espectro de frecuencia.....	120
4.7 Desbalance en los capacitores.....	123
Referencias.....	127
Conclusiones y recomendaciones.....	128
BIBLIOGRAFÍA.....	131

APÉNDICE

Código fuente del programa para la implementación de la técnica SVM.....	133
--	-----

ANEXOS

Anexo 1: Hoja de datos del controlador digital de señales dsPIC30F4011.....	138
Anexo 2: Hoja de datos del transistor MOSFET STW25NM60N.....	144

LISTA DE FIGURAS

		Página
Figura 2.1	Topología del inversor monofásico medio puente.....	12
Figura 2.2	Topología del inversor trifásico.....	12
Figura 2.3	Rama de un inversor de (a) 2 niveles, (b) 3 niveles, (c) m niveles.....	15
Figura 2.4	Topología del inversor multinivel con puentes H en cascada.....	19
Figura 2.5	Topología de un inversor puente completo de 5 niveles con capacitores flotantes.....	21
Figura 2.6	Topología de un inversor monofasico NPC de 3 niveles.....	24
Figura 2.7	Forma de onda de salida del inversor NPC de tres niveles	26
Figura 2.8	Ejemplo de la técnica PWM analógica.....	32
Figura 2.9	Construcción de los vectores de estado de tensión asociados al estado de conmutación poo del inversor de tres niveles.....	34
Figura 2.10	Diagrama normalizado SVM asociado a los 27 estados de conmutación para el inversor de tres niveles.....	36
Figura 2.11	Notación relativa a los triángulos definidos por los vértices de los vectores de estado: triángulos grandes (azul) y triángulos pequeños (verde).....	38
Figura 2.12	Vector de referencia en el triángulo $j = 1$	40
Figura 2.13	Vector de referencia en el triángulo $j = 2$	41
Figura 2.14	Vector de referencia en el triángulo $j = 3$	42
Figura 2.15	Vector de referencia en el triángulo $j = 4$	43
Figura 2.16	Diagrama del Convertidor <i>Push – Pull</i>	47
Figura 2.17	Diagrama a bloques de la arquitectura tipo Harvard para dsPIC30F.....	48
Figura 2.18	Diagrama de pines del dsPIC30F4011.....	49

	Página
Figura 2.19	Diagrama a bloques del dsPIC30F4011..... 50
Figura 2.20	Diagrama a bloques del oscilador del dsPIC30F4011..... 52
Figura 2.21	Diagrama a bloques del temporizador clase A..... 53
Figura 2.22	Diagrama a bloques del temporizador clase B..... 54
Figura 2.23	Diagrama a bloques del temporizador clase C..... 55
Figura 2.24	Diagrama a bloques del ADC del dsPIC30F4011..... 57
Figura 3.1	Diagrama a bloques del sistema..... 65
Figura 3.2	Inversor trifásico NPC de tres niveles..... 67
Figura 3.3	Inversor implementado..... 69
Figura 3.4	Referencias de los transistores en el inversor implementado..... 70
Figura 3.5	Ubicación de los Drivers en el inversor implementado..... 71
Figura 3.6	Tres ramas altas del inversor (azul) y tres ramas bajas (rojo)..... 72
Figura 3.7	Diagrama de la fuente de alimentación <i>Push Pull</i> propuesta..... 72
Figura 3.8	Fuente de alimentación <i>Push – Pull</i> implementada..... 73
Figura 3.9	Diagrama de flujo del programa principal..... 75
Figura 3.10	Diagrama de flujo de la rutina de interrupción del TIMER1. 76
Figura 3.11	Diagrama de flujo de la rutina de interrupción del TIMER2. 76
Figura 3.12	Diagrama de flujo de la rutina de interrupción del TIMER3. 76
Figura 3.13	Diagrama de flujo de la rutina de interrupción del TIMER4 76
Figura 3.14	Código de la declaración de funciones prototipo..... 77
Figura 3.15	Código de la declaración de constantes y variables..... 78
Figura 3.16	Código del programa principal..... 80
Figura 3.17	Código de la función prototipo <i>configuración_inicial</i> 81
Figura 3.18	Código de la configuración de los puertos..... 81
Figura 3.19	Código de la configuración del ADC..... 81
Figura 3.20	Código de la configuración de los temporizadores..... 82
Figura 3.21	Código de la configuración de las variables..... 83

Figura 3.22	Código de la rutina SVM.....	84
Figura 3.23	Código de la rutina <i>magnitud</i>	84
Figura 3.24	Código de la rutina <i>ADCRead</i>	85
Figura 3.25	Código de la rutina <i>índice de modulación</i>	85
Figura 3.26	Código de la rutina <i>posición</i>	86
Figura 3.27	Diagrama SVM para ubicar el vector de referencia.....	87
Figura 3.28	Ecuaciones de las seis rectas mostradas en la figura 3.27.....	88
Figura 3.29	Notación relativa a los triángulos definidos por los vértices de los vectores de estado: triángulos grandes (azul) y triángulos pequeños (verde).....	89
Figura 3.30	Asociación de la variable de 6 elementos <i>pasa[6]</i> con las diferentes rectas.....	89
Figura 3.31	Código de la rutina <i>calcula_sector</i>	90
Figura 3.32	Código de la rutina <i>calcula_triángulo</i>	92
Figura 3.33	Código de la rutina <i>duty_cicle</i>	93
Figura 3.34	Código de la subrutina <i>periodos</i>	94
Figura 3.35	Código de la rutina <i>calcula_vectores</i>	96
Figura 3.36	Código de la rutina de servicio a la interrupción del temporizador 1.....	97
Figura 3.37	Código de la rutina de servicio a la interrupción del temporizador 2.....	98
Figura 3.38	Código de la rutina de servicio a la interrupción del temporizador 3.....	99
Figura 3.39	Código de la rutina de servicio a la interrupción del temporizador 4.....	100
Figura 3.40	Diagrama esquemático del DSC.....	101
Figura 3.41	Control digital del inversor.....	101
Figura 3.42	Placa de circuito impreso final.....	102
Figura 4.1	Señal de disparo de compuerta a fuente del transistor S_{11}	105

Figura 4.2	Ubicación de las terminales del disparo de compuerta a fuente en el osciloscopio.....	106
Figura 4.3	Voltaje de fase V_{AN} del inversor trifásico con carga resistiva para un índice de modulación de 0.2 y una frecuencia de 60 Hz.....	107
Figura 4.4	Voltaje de fase V_{AN} del inversor trifásico con carga resistiva para un índice de modulación de 0.8 y una frecuencia de 60 Hz.....	107
Figura 4.5	Voltaje de fase V_{AN} , V_{BN} , y V_{CN} con carga resistiva, índice de modulación de 0.2 y frecuencia de 60 Hz.....	108
Figura 4.6	Voltaje de fase V_{AN} , V_{BN} , y V_{CN} con carga resistiva, índice de modulación de 0.8 y frecuencia de 60 Hz.....	109
Figura 4.7	Voltaje de línea V_{AB} con carga resistiva, índice de modulación de 0.2 y frecuencia de 60 Hz.....	110
Figura 4.8	Voltaje de línea V_{AB} con carga resistiva, índice de modulación de 0.8 y frecuencia 60 Hz.....	110
Figura 4.9	Vector de referencia con un índice de modulación de 0.8...	111
Figura 4.10	Voltaje de fase V_{AN} del inversor trifásico con carga inductiva para un índice de modulación de 0.2 y una frecuencia de 60 Hz.....	112
Figura 4.11	Voltaje de fase V_{AN} del inversor trifásico con carga inductiva para un índice de modulación de 0.8 y una frecuencia de 60 Hz.....	113
Figura 4.12	Voltaje de Fase V_{AN} , V_{BN} , y V_{CN} con carga inductiva, índice de modulación de 0.2 y frecuencia 60 Hz.....	114
Figura 4.13	Voltaje de Fase V_{AN} , V_{BN} , y V_{CN} con carga inductiva, índice de modulación de 0.8 y frecuencia 60 Hz.....	114
Figura 4.14	Voltaje de línea V_{AB} con carga inductiva, índice de modulación de 0.2 y frecuencia 60 Hz.....	115
Figura 4.15	Voltaje de línea V_{AB} con carga inductiva, índice de modulación de 0.8 y frecuencia 60 Hz.....	116

Figura 4.16	Forma de onda de corriente de fase “a” con carga resistiva, y un índice de modulación de 0.2.....	117
Figura 4.17	Forma de onda de corriente de fase “a” con carga resistiva, y un índice de modulación de 0.8.....	118
Figura 4.18	Forma de onda de corriente en el neutro con carga resistiva, y un índice de modulación de 0.2.....	119
Figura 4.19	Forma de onda de corriente en el neutro con carga resistiva, y un índice de modulación de 0.8.....	119
Figura 4.20	Forma de onda de corriente de fase “a” con carga inductiva, e índice de modulación de 0.8.....	120
Figura 4.21	Espectro de frecuencia del voltaje de fase V_{AN} con índice de modulación de 0.2 y carga resistiva $R = 50 \Omega$	121
Figura 4.22	Espectro de frecuencia del voltaje de fase V_{AN} con índice de modulación de 0.8 y carga resistiva $R = 50 \Omega$	121
Figura 4.23	Espectro de frecuencia del voltaje de línea V_{AB} con índice de modulación de 0.2 y carga resistiva $R = 50 \Omega$	122
Figura 4.24	Espectro de frecuencia del voltaje de línea V_{AB} con índice de modulación de 0.8 y carga resistiva $R = 50 \Omega$	123
Figura 4.25	Desbalance de voltaje del capacitor C_1 con índice de modulación de 0.2.....	124
Figura 4.26	Desbalance de voltaje del capacitor C_1 con índice de modulación de 0.8.....	125
Figura 4.27	Desbalance de voltaje del capacitor C_2 con índice de modulación de 0.2.....	125
Figura 4.28	Desbalance de voltaje del capacitor C_2 con índice de modulación de 0.8.....	126

LISTA DE TABLAS

		Página
Tabla 2.1	Comparación de necesidades de componentes por rama de los tres inversores.....	30
Tabla 2.2	Estados de conmutación y voltajes de línea asociados al inversor de tres niveles	35
Tabla 2.3	Estados de conmutación escogidos para cada vector de tensión con una conexión a tensión cero.....	44
Tabla 2.4	Estados de conmutación escogidos para cada vector de tensión con dos conexiones a tensión cero.....	45
Tabla 2.5	Registros de control del ADC del dsPIC30F4011.....	58

LISTA DE ACRÓNIMOS Y ABREVIATURAS

ADC	Convertidor analógico - digital (<i>Analog to Digital Converter</i>)
DSC	Controlador digital de señal (<i>Digital Signal Controller</i>)
DSP	Procesador digital de señal (<i>Digital Signal Processor</i>)
HS	Alta velocidad (<i>High Speed</i>)
IGBT	Tiristor controlado de compuerta integrada (<i>Insulated-Gate Bipolar Transistors</i>)
MOSFET	Transistor de Efecto de Campo de Semiconductor Metal-Óxido (<i>Metal-Oxid Semiconductor Field Effect Transistor</i>)
MCU	Microcontrolador (<i>Microcontroller</i>)
PLL	Ciclo de amarre de fase (<i>Phase Locked Loop</i>)
PWM	Modulación por ancho de pulso (<i>Pulse Width Modulation</i>)
SPWM	Modulación por ancho de pulso senoidal (<i>Sinusoidal Pulse Width Modulation</i>)
SVM	Modulación por vectores espaciales (<i>Space Vector Modulation</i>)
NPC	Anclado al Punto Neutro (<i>Neutral Point Clamped</i>)
DCI	Inversor de Diodo Anclado (<i>Diode Clamped Inverter</i>)
FCMLI	Inversor Multinivel de Capacitores Flotantes (<i>Flying-Capacitors MultiLevel Inverter</i>)
DCMLI	Inversor Multinivel Con Diodo Anclado (<i>Diode-Clamped MultiLevel Inverter</i>)
NTV	Tres Vectores Más Cercanos (<i>Nearest Three Vectors</i>)

Resumen

En éste documento se presenta el análisis y diseño para la implementación de la técnica de Modulación de Vectores Espaciales (SVM) aplicada a un Inversor Trifásico de Tres Niveles con Diodo Anclado al Punto Neutro. Esta técnica se implementa en un Controlador Digital de Señales (DSC) que genera la señalización para el control de dicho inversor. Se presenta también el procedimiento y el código fuente del programa en Lenguaje C para la implementación de la técnica SVM, así como la placa de circuito impreso en la que fueron realizadas las pruebas con carga resistiva y carga inductiva, haciendo uso de un motor de inducción trifásico para variar la ganancia del convertidor, al incrementar o disminuir el índice de modulación que hace variar el ancho del pulso a la salida del inversor.

El trabajo se presenta en 4 capítulos:

En el capítulo I se presenta la problemática general del proyecto, el objetivo que se persigue al finalizar la tesis, así como la justificación del mismo, limitaciones, delimitaciones y alcances que se lograrán en el mismo.

En el capítulo II se presenta el marco teórico del proyecto, la investigación bibliográfica referente los inversores de potencia, la clasificación que se tiene dependiendo del número de fases y el número de niveles, así como las diferentes topologías y principios de operación de los Inversores Multinivel. También se presentan diferentes técnicas de modulación, como los son las analógicas y las digitales, y se hace énfasis a la técnica SVM implementada en esta tesis.

El capítulo III presenta el desarrollo del proyecto, la forma en la que se implementó el inversor, y cómo se implementaron las alimentaciones de los dispositivos del mismo. Se hace énfasis en la implementación de la técnica SVM y se analiza a detalle todo el código fuente del programa para los cálculos de las rutinas. Al finalizar se presenta la placa de circuito impreso implementada.

En el capítulo IV se presentan las pruebas y resultados obtenidos, para diferentes tipos de carga, como lo es resistiva e inductiva haciendo uso de un motor de inducción trifásico.

Para finalizar se presentan las conclusiones a las que se llegaron al finalizar la tesis, y las recomendaciones para posibles mejoras en futuros trabajos relacionados.

CAPÍTULO I

Introducción

Los convertidores de CD a CA se llaman *inversores*. La función de un inversor es cambiar un voltaje de entrada de CD a un voltaje simétrico de salida de CA, con la magnitud y frecuencia deseadas. El voltaje de salida podría ser fijo o variable, a una frecuencia fija o variable. Un voltaje variable de salida se puede obtener haciendo variar el voltaje de entrada de CD, manteniendo constante la ganancia del inversor. Por otro lado, si el voltaje de CD de entrada es fijo y no es controlable, se puede obtener un voltaje variable de salida haciendo variar la ganancia del inversor, lo que se consigue normalmente con control por modulación por ancho de pulso (PWM, de *Pulse Width Modulation*) dentro del inversor. La *ganancia del inversor* se puede definir como la relación entre el voltaje de salida de CA y el voltaje de entrada de CD [1].

Las formas de onda del voltaje de salida de los inversores ideales deberían ser sinusoidales. Sin embargo, las de los inversores prácticos no son sinusoidales y contienen ciertas armónicas. En aplicaciones de potencia baja e intermedia se pueden aceptar voltajes de onda cuadrada o de onda casi cuadrada, y para aplicaciones con alta potencia se requieren formas de onda sinusoidal, con poca distorsión. Con la disponibilidad de los dispositivos semiconductores de potencia de alta velocidad, se pueden minimizar los contenidos armónicos del voltaje de salida, o al menos reducirlos en forma importante, mediante técnicas de conmutación.

Los inversores se usan mucho en aplicaciones industriales, como por ejemplo impulsores (“variadores”, “reguladores” o “controles”) de motor de CA y velocidad variable, o en el calentamiento por inducción, fuentes de alimentación de reserva y fuentes de alimentación ininterrumpibles. La entrada puede ser una batería, una celda de combustible, celda solar u otra fuente de CD. Las salidas más comunes monofásicas son:

- 1) 120 V a 60 Hz.
- 2) 220 V a 50 Hz.
- 3) 115 V a 400 Hz.

Para sistemas trifásicos de gran potencia, las salidas normales son:

- 1) 220 a 380 V a 50 Hz.
- 2) 120 a 208 V a 60 Hz.
- 3) 115 a 200 V a 400 Hz.

Los inversores se pueden clasificar en el sentido amplio en dos tipos:

- 1) Inversores Monofásicos.
- 2) Inversores Trifásicos.

Cada uno puede usar dispositivos controlados de encendido y apagado (como transistores bipolares de unión [BJT], transistores de efecto de campo, de metal

óxido semiconductor [MOSFET], transistores bipolares de compuerta aislada [IGBT], tiristores controlados por compuerta [GTO]). Estos inversores usan en general señales de control por modulación PWM. Un inversor se denomina *inversor alimentado por voltaje* (VFI, de *Voltage Fed Inverter*) si el voltaje de entrada permanece constante; *inversor alimentado por corriente* (CFI, de *Current Fed Inverter*) si la corriente de entrada se mantiene constante, y *convertidor enlazado con CD variable* si el voltaje de entrada es controlable. Si se hace pasar el voltaje o la corriente de salida del inversor por cero, creando un circuito resonante LC, a esta clase de inversor se le llama *inversor de pulso resonante* y tiene muchas aplicaciones en la electrónica de potencia.

1.1 Antecedentes teóricos

Desde la introducción de los convertidores de tres niveles de tensión de topología NPC (*Neutral Point Clamped*) presentada por primera vez en 1981 por Nabae [2], se han mostrado las ventajas significativas que éstos y, en general, presentan frente a los convertidores convencionales de dos niveles en aplicaciones de media y gran potencia (a partir de 50 kW) y media y alta tensión (2-13 kV). Entre las aplicaciones más interesantes se incluyen las fuentes de energía renovables, máquinas eléctricas, distribución de potencia, calidad de potencia y las aplicaciones de acondicionamiento de potencia. Además, dado que los precios de los semiconductores de potencia y de los microprocesadores continúan bajando es de esperar que el uso de las topologías multinivel también se extiendan a aplicaciones de baja potencia (menos de 10 kW).

Hasta hace relativamente pocos años la implementación física de convertidores multinivel se encontraba limitada debido a la complejidad del control de conmutación de los dispositivos de potencia del convertidor. En los últimos años se han utilizado inversores fuente de tensión multinivel en muchas aplicaciones de media y alta potencia, siendo la topología DCI (*Diode Clamped Inverter*), derivada de la NPC, la más extendida. Sin embargo, la elección de la mejor topología, estrategia de control y dispositivos semiconductores de potencia utilizados en el convertidor dependerá de la aplicación concreta y de los criterios de diseño [3].

Los inversores multinivel han atraído mucho interés en el campo de la electrónica de potencia. Presentan un conjunto nuevo de propiedades que son muy adecuadas para usarse en compensación de potencia reactiva. Puede ser más fácil producir un inversor de alta potencia y alto voltaje con la estructura multinivel, por la forma en la que se controlan en la estructura los esfuerzos dieléctricos del dispositivo. Al aumentar la cantidad de niveles de voltaje en el inversor sin necesidad de mayores especificaciones nominales de los dispositivos individuales, se puede aumentar la potencia nominal. La estructura única de fuente de voltaje de los inversores multinivel les permite alcanzar altos voltajes con pocas armónicas, sin el uso de transformadores o de dispositivos de conmutación sincronizada conectados en serie. A medida que aumenta la cantidad de niveles de voltaje, se reduce en forma importante el contenido de armónicas en la forma de onda de voltaje de salida [1].

Dado el creciente interés en mejorar la calidad de la onda que los sistemas convertidores de potencia inyectan en la red a partir de fuentes de energía renovable, comienza a adquirir importancia el desarrollo de los convertidores multinivel, especialmente el de tres niveles, en este tipo de aplicaciones, pues son los sistemas de potencia idóneos en aplicaciones de tecnologías que tienden a aumentar la potencia de generación y a mejorar el aprovechamiento de la energía disponible. Los convertidores multinivel constituyen una interesante alternativa a los convertidores de dos niveles convencionales para conectar en serie o en paralelo a la red de energía de corriente alterna con fuentes de energía renovable, como pueden ser: energía eólica, energía solar fotovoltaica, celdas de combustible o dispositivos de almacenamiento de energía como capacitores o baterías. Existen además otras aplicaciones adicionales de convertidores multinivel para máquinas eléctricas y filtros activos de potencia. De hecho, la evolución de estas tecnologías está basada en el incremento gradual de potencia instalada [3].

Una importante área de aplicación de los convertidores multinivel es la conversión de Energía Fotovoltaica. El Instituto Tecnológico de Sonora (ITSON) unidad Obregón en el Departamento de Ingeniería Eléctrica y Electrónica, ha desarrollado un sistema de generación de energía eléctrica mediante sistemas fotovoltaicos [4].

1.2 Planteamiento del problema

En la carrera de Ingeniería en Electrónica del Instituto Tecnológico de Sonora, los inversores multinivel son un área de estudio nueva, ya que los trabajos anteriores de inversores hacen referencia a los convertidores de 2 niveles, utilizando técnicas de modulación analógicas y, recientemente, existe un trabajo que hace referencia a la técnica SVM para un inversor trifásico de 2 niveles. Pero no existe en la actualidad, un documento de consulta que haga referencia a la técnica de modulación digital SVM para inversores trifásicos de 3 niveles.

Las formas de onda del voltaje de salida de los inversores ideales deberían ser sinusoidales. Sin embargo, las de los inversores prácticos no son sinusoidales y contienen ciertas armónicas. En aplicaciones de potencia baja e intermedia se pueden aceptar voltajes de onda cuadrada o de onda casi cuadrada, y para aplicaciones con alta potencia se requieren formas de onda sinusoidal, con poca distorsión [1].

La necesidad de altas intensidades de corriente utilizando bajos niveles de tensión para generación de potencias del orden de varios kW obliga a modificar el diseño de los generadores de corriente eléctrica y los convertidores de potencia. Además, la utilización de convertidores multinivel permite la conexión directa a la red de media tensión sin la necesidad de utilizar filtros a la salida, que normalmente son de alto costo [3].

Actualmente casi todos los convertidores estáticos utilizados en la industria están basados en inversores de dos niveles con técnicas de modulación de ancho de pulso (PWM). Este tipo de modulación no genera una onda de voltaje perfecta y por lo tanto se tienen problemas relacionados principalmente con las armónicas y la alta frecuencia de operación de los transistores que componen los puentes de estos convertidores.

En el control de motores, daños y fallas en las máquinas han sido evidenciados en la industria debido a estas altas frecuencias de operación. Entre los principales problemas están las fallas en los rodamientos del motor y pérdidas del aislamiento en las bobinas de las máquinas, causadas por corrientes circulantes, desgaste dieléctrico, sobretensión y descargas por efecto corona. Las corrientes circulantes son generadas por capacidades parásitas que se generan en las distintas capas de las bobinas del motor. Los bruscos cambios de voltaje (dV/dt) inducen corrientes y descargas corona en los enrollados [5].

1.3 Objetivos

1. Implementar la técnica SVM en un inversor trifásico de 3 niveles.
 - 1.1. Diseñar e implementar un inversor trifásico de 3 niveles con topología NPC.
 - 1.2. Desarrollar un algoritmo funcional para la técnica SVM.
 - 1.3. Implementar la técnica SVM en un controlador digital de señales (DSC).
 - 1.4. Aplicar la técnica SVM al inversor multinivel implementado.
2. Realizar pruebas de rendimiento a la modulación del inversor.

1.4 Justificación

En la carrera de Ingeniería en Electrónica del Instituto Tecnológico de Sonora, los alumnos podrán tener acceso a un documento, en donde se pueda consultar cómo funciona la técnica SVM, y como aplicarla a un inversor trifásico de 3 niveles.

Los convertidores multinivel presentan muchas ventajas respecto a los convertidores convencionales de dos niveles de tensión que justifican el reciente interés por su aplicación en sistemas de gran potencia. Los convertidores multinivel no sólo presentan la capacidad de incrementar la magnitud de la tensión de salida aumentando el rendimiento del convertidor sino que además, reducen el contenido armónico de la tensión y de la intensidad de corriente de salida y permiten disminuir la frecuencia de conmutación y la tensión soportada por cada dispositivo

semiconductor de potencia. Además, permiten mitigar problemas de interferencias electromagnéticas (EMI) disminuyendo la tensión de conmutación dV/dt . De esta forma los convertidores multinivel limitan de forma natural los problemas de grandes transitorios de tensión debidos a reflexiones en los cables y que pueden causar daños en los arrollamientos de un motor y otro tipo de problemas.

Un convertidor multinivel permite sintetizar una forma de onda sinusoidal de tensión a partir de varios niveles de tensión, obtenidos típicamente a partir de fuentes de tensión capacitivas. Los convertidores multinivel permiten incrementar la tensión de alterna sin utilizar transformador. Además, la reducción de armónicos de baja frecuencia debido a los diferentes niveles de tensión del convertidor hace posible disminuir el tamaño de las inductancias de los filtros de salida.

1.5 Limitaciones

Las limitaciones presentadas en este proyecto es que no se cuenta con un punto de referencia en inversores trifásicos de tres niveles en la institución, ya que los inversores estudiados e implementados hacen referencia a los inversores trifásicos de dos niveles de tensión.

No se cuentan con DSC suficientemente rápidos para realizar operaciones matemáticas con puntos flotantes en un tiempo sumamente reducido para poder incrementar la frecuencia de conmutación del inversor.

1.6 Delimitaciones

Se diseñará un inversor trifásico de tres niveles con diodo anclado al punto neutro (NPC) el cual se implementará en placa de circuito impreso, en el cual se incluirán todos los componentes necesarios para sintetizar una señal sinusoidal a partir de 3 niveles de tensión.

Se programará la modulación por vectores espaciales (SVM) en un controlador digital de señales dsPIC30F4011 de Microchip® que se encargará de variar el ancho de pulso para obtener un voltaje variable a la salida del inversor.

Se conectará la salida del inversor trifásico a una carga resistiva para poder realizar pruebas y mediciones, posteriormente se conectará a un motor de inducción trifásico para realizar las mismas pruebas y hacer una comparación entre los dos tipos de carga.

1.7 Alcances

Obtener a la salida del convertidor un voltaje trifásico el cual pueda ser variado con un potenciómetro conectado a un canal analógico del dsPIC30F4011 de Microchip.

Realización de pruebas y obtención de resultados a partir del inversor trifásico conectado a carga inductiva o resistiva.

Referencias

[1] MUHAMMAD H. Rashid, "Electrónica de Potencia, Circuitos, Dispositivos y Aplicaciones." Prentice Hall, Tercera edición, 2004.

[2] A. NABAE, I. Takahashi, and H. Akagy, "A Neutral-Point Clamped PWM Inverter." IEEE Transactions on Industrial Applications, 1981.

[3] MARTÍN Prats M. Ángeles, "Nuevas Técnicas de Modulación Vectorial para Convertidores Electrónicos de Potencia Multinivel" Tesis de Doctor por la Universidad de Sevilla, Sevilla, Junio de 2003.

[4] PEÑUELAS Machado José Ángel, "Algoritmo de Seguimiento del Punto de Máxima Potencia para Sistemas Fotovoltaicos en Cd. Obregón, Sonora", Tesis de Ingeniero en Electrónica, Instituto Tecnológico de Sonora, Cd. Obregón, Sonora, Octubre de 2008.

[5] ELGUETA Díaz Cristian Marcelo, "Aplicación de un Inversor Multinivel como Variador de Frecuencia en un Motor de Inducción Trifásico", Tesis de Maestro en Ciencias de la Ingeniería, Pontificia Universidad Católica de Chile, Santiago de Chile, Julio de 2005.

CAPÍTULO II

Marco Teórico

En este capítulo se aborda la teoría necesaria para la comprensión de la tesis en general, se mencionan los inversores existentes y su clasificación, así como sus ventajas y desventajas, y se explica a detalle el convertidor multinivel implementado. También se hace mención a las diferentes técnicas de modulación analógicas y digitales existentes para el control de los inversores de dos niveles y multinivel, así como también se explica a detalle la técnica de modulación por vectores espaciales (SVM) para su correcta implementación en un controlador digital de señales (DSC). Así también se aborda el tema del DSC utilizado y el lenguaje y compilador elegidos para trabajar con el mismo y se incluye la bibliografía consultada. El objetivo de este capítulo es obtener los conocimientos necesarios para la comprensión del diseño e implementación de un inversor trifásico de tres niveles con topología NPC.

2.1 Inversores

Como se mencionó en el Capítulo I, un inversor es un circuito utilizado para convertir corriente continua en corriente alterna. La función de un inversor es cambiar un voltaje de entrada de corriente directa a un voltaje simétrico de salida de corriente alterna, con la magnitud y frecuencia deseadas. Los inversores son utilizados en una gran variedad de aplicaciones, desde pequeñas fuentes de alimentación para computadoras, hasta aplicaciones industriales para manejar alta potencia. Los inversores también son utilizados para convertir la corriente continua generada por los paneles solares fotovoltaicos, acumuladores, baterías, etc; en corriente alterna y de esta manera poder ser inyectados en la red eléctrica o usados en instalaciones eléctricas aisladas.

2.2 Clasificación de los inversores

Los inversores, en forma general, pueden clasificarse en 2 grandes grupos, siendo estos grupos divididos por el número de fases de voltaje de salida, y por la cantidad de niveles para sintetizar la señal de CA.

2.3 Clasificación de los inversores por su número de fases

En esta clasificación de inversores hay 2 grupos muy importantes, los *inversores monofásicos* y los *inversores trifásicos*.

2.3.1 Inversores monofásicos

Los inversores se pueden clasificar dependiendo del número de fases de voltaje de corriente alterna que generan a la salida. Cuando se genera una sola fase de voltaje a la salida se conoce como un inversor monofásico. En la figura 2.1 se puede observar la topología general de un inversor monofásico medio puente.

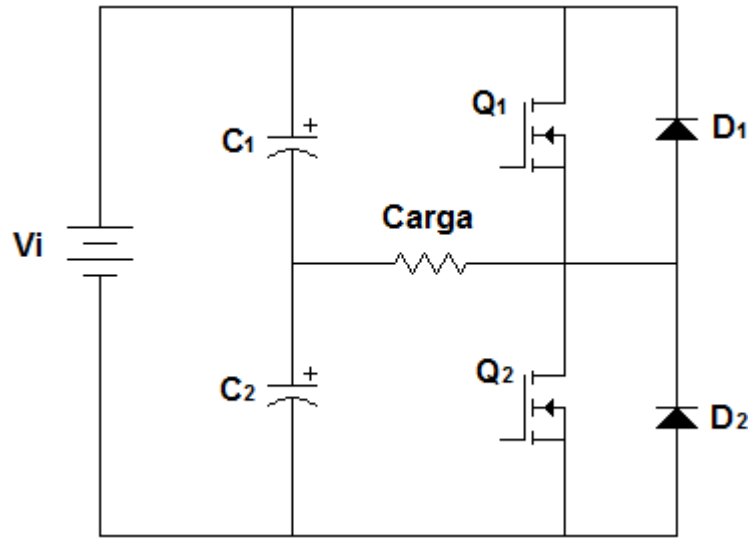


Figura 2.1. Topología del inversor monofásico medio puente.

2.3.2 Inversores trifásicos

Cuando se generan 3 fases de voltaje a la salida, se le conoce como inversor trifásico. En el caso normal, los inversores trifásicos se usan en aplicaciones de grandes potencias. Se pueden conectar 3 inversores monofásicos medio puente en paralelo, como se ve en la figura 2.2, para formar la configuración de un inversor trifásico. Las señales de control de los inversores monofásicos se deben adelantar o atrasar 120 grados entre sí, para obtener voltajes (fundamentales) trifásicos balanceados [1].

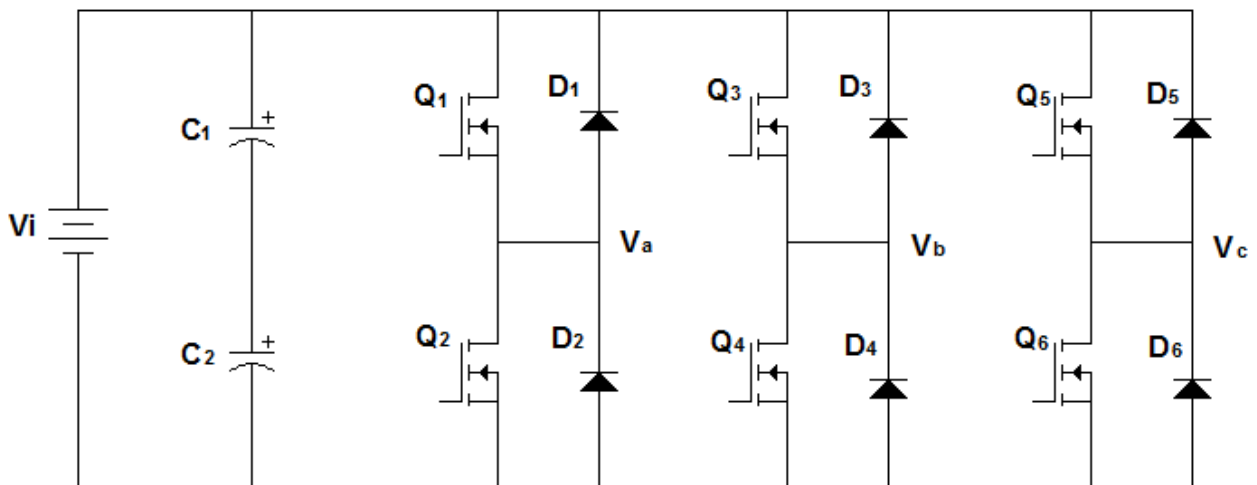


Figura 2.2. Topología del inversor trifásico.

En el inversor trifásico, hay seis dispositivos de conmutación designados (Q_1 a Q_6) y seis diodos de marcha libre (D_1 a D_6) dispuestos en antiparalelos con los interruptores. Estos diodos aseguran por un lado la continuidad de la corriente en la carga inductiva y por otro lado la reversibilidad de la potencia al poder inyectar corriente desde la carga a la batería de CD. Cada rama del inversor está formada por dos interruptores en paralelo con sus diodos de marcha libre, estando la salida a cada fase de la carga situada en el punto medio de la rama.

Las señales de control de los dos interruptores de una misma rama deben ser complementarias a fin de no cortocircuitar la fuente de CD de alimentación. Además debe considerarse que los interruptores precisan de un tiempo mínimo, tanto en la apertura para anular la corriente, como en su cierre para su establecimiento. Ésta corriente de descarga circulará por los diodos dispuestos en paralelo con cada interruptor. Una vez ésta corriente sea cero, se permitirá el cierre del interruptor complementario. Éste tiempo de espera se denomina generalmente *tiempo muerto* y debe ser respetado y tenido en cuenta durante el diseño.

La inclusión de estos tiempos muertos provocan una deformación de la tensión entre fases de salida, que será proporcional al valor del tiempo muerto y a la tensión de la fuente de CD, e inversamente proporcional al periodo de modulación. Otro efecto desfavorable de los tiempos muertos es el aumento de la amplitud de los armónicos de la tensión de salida.

2.4 Clasificación de los inversores por su número de niveles

En esta clasificación, existen 2 grandes grupos, los que se conocen como *inversores de 2 niveles* y los *inversores multinivel*.

2.4.1 Inversores de 2 niveles

Los inversores de fuente de voltaje que producen un voltaje o una corriente de salida cuyos niveles son $\pm V_{CD}$ se conocen como inversores en 2 niveles. Para obtener una forma de onda de voltaje o corriente con alta calidad, con un contenido mínimo de

rizo, requieren conmutación en alta frecuencia, junto con diversas estrategias PWM. Sin embargo, en aplicaciones de alta potencia y alto voltaje, esos inversores en dos niveles tienen algunas limitaciones para operar a alta frecuencia, principalmente a causa de las pérdidas por conmutación y limitaciones de las especificaciones nominales de los dispositivos. Además, los dispositivos semiconductores de conmutación se deben usar de tal manera que se eviten problemas asociados con sus combinaciones en serie-paralelo, necesarias para obtener la capacidad de manejo de altos voltajes y corrientes [1]. La figura 2.1 es un ejemplo de un inversor de dos niveles monofásico medio puente.

2.4.1.1 Ventajas

- Su implementación es de bajo costo, gracias al reducido número de componentes.
- El control es más sencillo, gracias al reducido número de dispositivos de conmutación.
- Las técnicas de modulación son más sencillas.

2.4.1.2 Desventajas

- Tienen alta distorsión armónica debido al reducido número de niveles disponibles a la salida.
- Se incrementan las pérdidas por conmutación debido a la alta frecuencia a la que operan.
- Los filtros de salida se vuelven complicados y de alto costo por las grandes inductancias que se manejan.

2.4.2 Inversores multinivel

Los inversores multinivel son topologías que se basan en un arreglo de semiconductores y fuentes CD, para formar el voltaje alterno. Las conmutaciones de los semiconductores permiten el escalonamiento de las distintas fuentes de voltaje continuo, generando una onda de voltaje de varios niveles. Si bien son más

complejos que los tradicionales inversores de dos niveles, algunas topologías permiten que los semiconductores trabajen con voltajes más reducidos y a una menor frecuencia de conmutación [5]. En la figura 2.3 se muestra la forma general de un inversor de 2 niveles (a), un inversor de 3 niveles (c) y un inversor de m niveles (c).

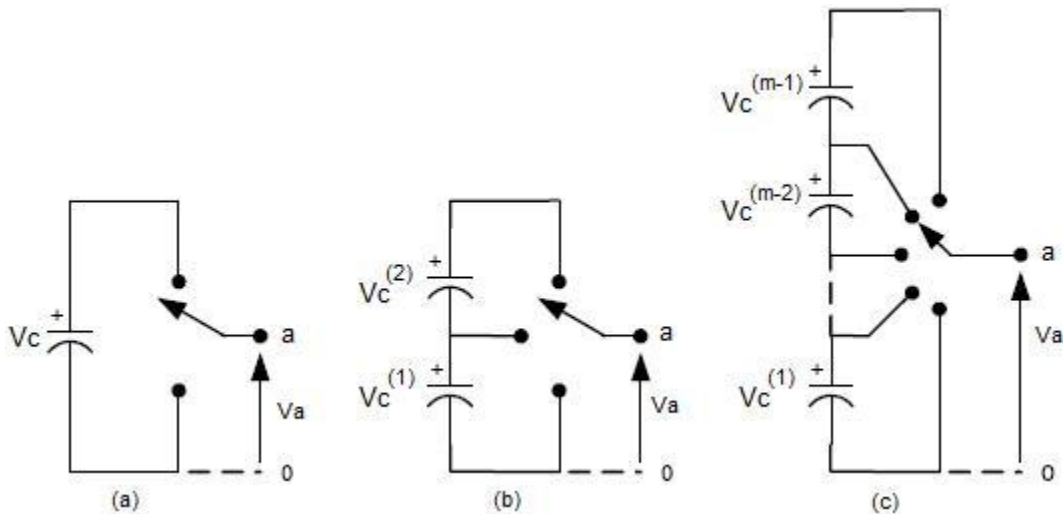


Figura 2.3. Rama de un inversor de (a) 2 niveles, (b) 3 niveles, (c) m niveles.

Los convertidores multinivel en los últimos años se han consolidado como una opción más para convertir energía CD – CA en el rango de potencia media y alta desde el punto de vista económico y técnico, podemos encontrar un incremento en las aplicaciones donde la conversión de la energía se realiza mediante convertidores multinivel.

De acuerdo con la historia, el concepto de conversión multinivel existe antes de los trabajos presentados por Nabae, Takahashi y Akagi en 1980[2], pero a partir de esta fecha se considera el punto de partida de los convertidores multinivel actuales. Es por esto que la comunidad científica y la industria han enfocado sus investigaciones y desarrollo en estos convertidores. La técnica de conversión multinivel se considera un área joven en el campo de la conversión de energía, presentando unas expectativas que hacen pensar en un gran potencial para una aplicación más amplia en el futuro.

La forma más general de entender los convertidores multinivel es considerarlo como un seccionador de voltaje. El voltaje de alterna de salida se obtiene a partir de diferentes niveles de voltaje de continua de entrada, de valor más pequeño accionando adecuadamente los interruptores del convertidor, siendo ésta la diferencia básica respecto a un inversor de 2 niveles [6].

2.4.2.1 Ventajas

- La disposición del voltaje (V_{CD}) de entrada en múltiples niveles permite aumentar varias veces el voltaje de trabajo del inversor empleando los mismos interruptores que un convertidor de 2 niveles. Cada interruptor debe bloquear únicamente el voltaje correspondiente a un único nivel de continua, evitando de esta manera el inconveniente del equilibrado estático y dinámico de la interconexión serie de los dispositivos electrónicos.
 - La potencia de los inversores se incrementa al emplear voltajes mayores, sin necesidad de incrementar la corriente, evitando así mayores pérdidas durante la conducción y por consecuencia mejorar el rendimiento del convertidor.
 - El voltaje de salida obtenido en un inversor multinivel presenta un contenido armónico menor que el obtenido en un inversor de 2 niveles de potencia equivalente. Teóricamente puede obtenerse una distorsión armónica nula si se dispone de un número infinito de niveles del voltaje de entrada trayendo como consecuencia la reducción del filtro de salida.
 - La corriente de entrada absorbida presenta una distorsión armónica baja.
 - Pueden operar con una baja frecuencia de conmutación, debajo de los 1000 Hz [7].
 - La respuesta dinámica del convertidor es más rápida, por tener más niveles de voltaje de salida y emplear filtros de menor tamaño.
 - Para aplicaciones industriales la demanda de potencia se ha incrementado en los últimos años, llegando a niveles hasta de 10 MW [8].
-

2.4.2.2 Desventajas

- Cuando se emplea un número grande de interruptores, la complejidad del control se incrementa.
- Es necesario tener diferentes niveles de voltaje de entrada, que pueden obtenerse con capacitores o mediante fuentes de continua. Cuando se emplean capacitores, los voltajes de estos deben permanecer constantes o equilibrados en cualquier condición de trabajo, con objeto de que cada capacitor actúe como una fuente de CD.
- Alto costo.

2.5 Topologías multinivel

El objetivo general del convertidor multinivel es sintetizar un voltaje casi sinusoidal a partir de varios niveles de voltaje de CD, que en forma típica se obtienen con fuentes de voltaje con capacitores. A medida que aumenta la cantidad de niveles, la forma de onda de salida sintetizada tiene más escalones, que producen una onda escalonada que tiende a la forma de onda que se desea. También, al añadir más escalones a la forma de onda, disminuye la distorsión armónica de la onda de salida, y tiende a cero a medida que aumenta la cantidad de niveles. A medida que aumenta la cantidad de niveles, también aumenta el voltaje que puede suministrarse sumando múltiples niveles de voltaje.

Dentro de las topologías multinivel más conocidas se encuentran:

- Inversor Multinivel con Puentes H en Cascada.
- Inversor Multinivel con Capacitores Flotantes.
- Inversor Multinivel con Diodo Anclado al Punto Neutro (NPC).

2.5.1 Inversor multinivel con puentes H en cascada

Este inversor, como su denominación lo indica, consta de una serie de etapas o puentes “H” conectados en serie, con una fuente CD independiente para cada etapa.

Los valores de cada una de estas fuentes podrían ser iguales, pero si se utilizan valores escalonados en potencia de tres, se maximiza la cantidad de niveles de salida del inversor y se minimizan las fuentes CD necesarias. No obstante, esta solución maximizada implica que para generar ciertos niveles de tensión, en la salida alterna del inversor, se hace necesario que las fuentes CD de algunos puentes auxiliares estén absorbiendo potencia. Esta absorción de potencia requiere que las fuentes CD de los puentes auxiliares sean bidireccionales, ya que bajo esta topología de fuentes escalonadas, no existen combinaciones redundantes de semiconductores para generar el mismo nivel de tensión en la salida del inversor sin que los puentes auxiliares absorban potencia en algunos momentos. A este tipo de inversores con fuentes escalonadas se les llamará inversores multinivel asimétricos [5].

En la figura 2.4 se puede observar la topología de este inversor. Se llama Principal al puente que trabajaba con el voltaje más alto, mientras que al resto de los puentes “H” se les llama Auxiliares. El Principal, además, es el que trabaja con la menor frecuencia de conmutación, mientras que el Auxiliar superior de la cadena presenta las características inversas, es decir, la mayor frecuencia de conmutación, pero el menor voltaje.

En éste inversor, cada fuente de CD está conectada a un inversor puente H. Cada inversor puede generar tres niveles de voltaje de salida ($+V_{CD}$, cero y $-V_{CD}$). Cada nivel del inversor está formado por cuatro interruptores (Q_1 , Q_2 , Q_3 y Q_4). Con la corriente de fase i_a , y retardando el voltaje de fase 90° , la carga promedio de cada interruptor será igual a cero en cada ciclo de línea. Por lo tanto, el voltaje de cada uno de los capacitores estará balanceado. Por otra parte, el número de niveles de voltaje a la salida del inversor está definido por $m = 2s + 1$, donde m es el número de niveles de voltaje a la salida, y s es el número de fuentes de CD. Por ejemplo, un inversor de éste tipo, de 9 niveles, tiene cuatro fuentes de CD, y cuatro inversores de puente H. Para la conversión de potencia, éste tipo de inversores necesita fuentes de CD separadas.

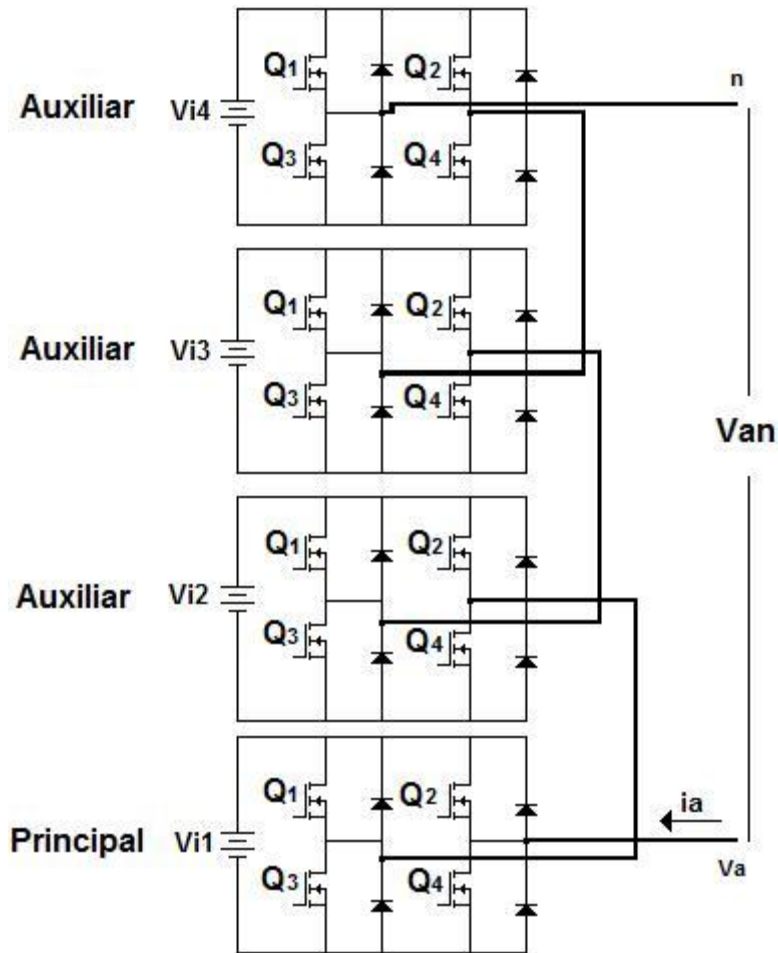


Figura 2.4. Topología del inversor multinivel con puentes H en cascada.

2.5.1.1 Propiedades del inversor con puentes H en cascada

Las propiedades principales del inversor en cascada son las siguientes:

- Para conversiones de potencia real, de CA a CD y después de CD a CA, los inversores en cascada necesitan fuentes separadas de CD. La estructura de las fuentes separadas de CD es adecuada para diversas fuentes de energía renovable, como celdas de combustible, energías renovables y biomasa.
- No es posible conectar las fuentes de CD a una continuación de otra entre dos convertidores, porque puede introducirse un cortocircuito cuando dos convertidores seguidos no conmutan en forma síncrona.

2.5.1.2 Ventajas

Las ventajas principales del inversor en cascada se pueden resumir como sigue:

- En comparación con los inversores de diodo fijador y de capacitores flotantes, requiere la mínima cantidad de componentes para obtener la misma cantidad de niveles de voltaje.
- Son posibles la distribución y el encapsulado optimizados del circuito, porque cada nivel tiene la misma estructura y no hay diodos fijadores adicionales, ni capacitores de balanceo de voltaje.
- Se pueden usar técnicas de conmutación suave para reducir las pérdidas por conmutación y los esfuerzos en los dispositivos.

2.5.1.3 Desventajas

La principal desventaja del inversor con puentes H en cascada es que:

- Necesita fuentes de CD separadas para conversiones de potencia real, y con ello limita sus aplicaciones.

2.5.2 Inversor multinivel con capacitores flotantes

Se le llama inversor multinivel de capacitores flotantes o FCMLI (de *Flying-Capacitors MultiLevel Inverter*). La figura 2.5 muestra la estructura de un convertidor multinivel de puente completo con capacitores flotantes basado en un convertidor de 5 niveles. Asumiendo que cada capacitor tiene el mismo rango de voltaje, la conexión en serie de los capacitores en la figura 2.5 es para indicar el nivel de voltaje entre los puntos anclados. El balance de los tres capacitores de la rama de fase *a*, C_{a1} , C_{a2} y C_{a3} es independiente de la rama de fase *b*. También, las dos ramas de fase *a* y *b* comparten los mismos capacitores ($C_1 - C_4$) del bus de CD de entrada.

El nivel de voltaje definido en el convertidor multinivel con capacitores flotantes es similar al del convertidor de diodo fijador. El voltaje de fase de un convertidor de *m*

niveles está formado por m niveles incluyendo el nivel de referencia o cero volts, y el voltaje de línea por $2m-1$ niveles.

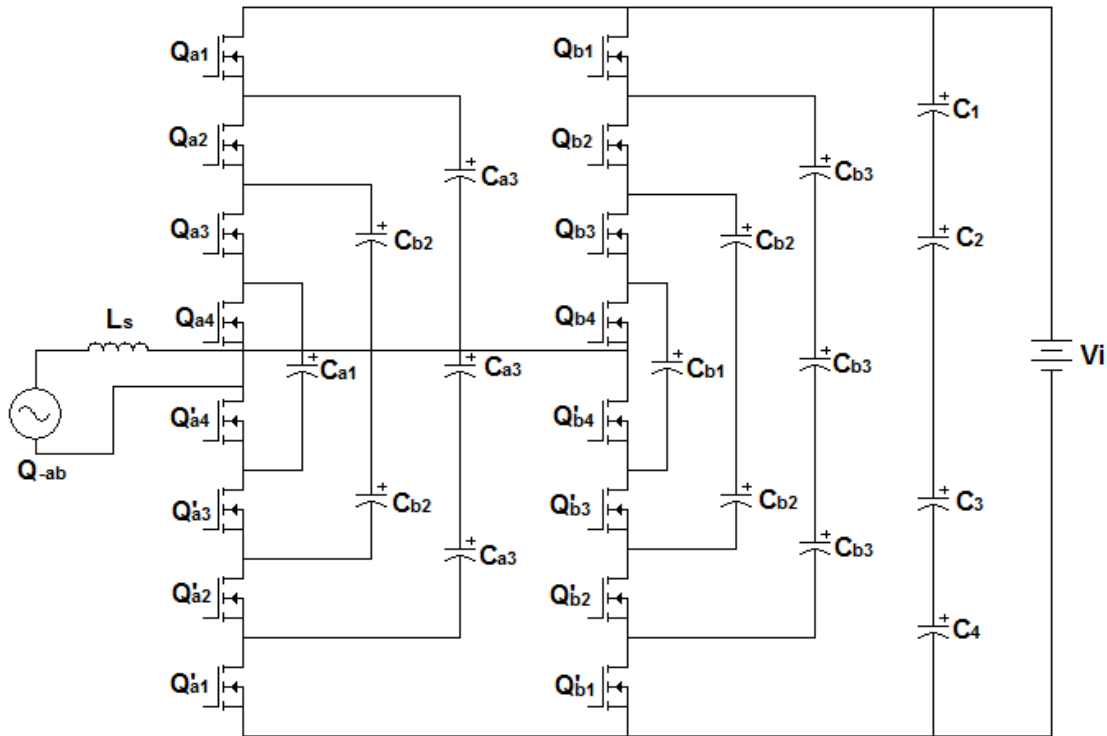


Figura 2.5. Topología de un inversor puente completo de 5 niveles con capacitores flotantes.

2.5.2.1 Propiedades de los inversores con capacitores flotantes

Las propiedades principales son las siguientes:

1. **Gran cantidad de capacitores.** El inversor requiere una gran cantidad de capacitores de almacenamiento. Suponiendo que el voltaje nominal de cada capacitor sea el mismo que la de un dispositivo de conmutación, un convertidor de m niveles requiere de un total de $(m-1) \times (m-2)/2$ capacitores auxiliares por rama de fase, y además $(m-1)$ capacitores de canal principal. Al contrario de un inversor con diodo fijador sólo requiere de $(m-1)$ capacitores con el mismo voltaje nominal. Así, para $m=5$, $N_C = 4 \times 3/2 + 4 = 10$, en comparación con $N_C = 4$ para el tipo con diodo fijador.
2. **Balanceo de voltajes de capacitor.** A diferencia del inversor con diodo fijador, el FCMLI tiene redundancia en sus niveles internos de voltaje. Un nivel

de voltaje es redundante si puede sintetizarse con dos o más combinaciones válidas de interruptor. La disponibilidad de redundancias de voltaje permite controlar los voltajes individuales de capacitor. Para producir el mismo voltaje de salida, el inversor puede utilizar distintas combinaciones de capacitores, permitiendo la carga o descarga preferencial de los capacitores individuales. Esta flexibilidad facilita la manipulación de los voltajes de capacitor, y el mantenerlos en sus valores correctos. Es posible emplear dos o más combinaciones de interruptor para niveles intermedios de voltajes (es decir, $3V_{CD}/4$, $V_{CD}/2$ y $V_{CD}/4$) en uno o varios ciclos de salida, para balancear la carga y la descarga de los capacitores. Así, con una selección adecuada de combinaciones de interruptor, se puede usar el convertidor multinivel con capacitores flotantes para conversiones de potencia real. Sin embargo, cuando se trata de conversiones de potencia real, la selección de una combinación de interruptores se hace muy complicada, y la frecuencia de conmutación debe ser mayor que la frecuencia fundamental.

2.5.2.2 Ventajas

Las ventajas principales del inversor con capacitores flotantes se pueden resumir de la siguiente manera:

- Grandes cantidades de capacitores de almacenamiento pueden proporcionar operación durante cortes de energía.
 - Estos inversores proporcionan redundancia de combinación de interruptor, para balancear distintos niveles de voltaje.
 - Como el inversor con diodo fijador con más niveles de voltaje, el contenido de armónicas es suficientemente bajo como para no necesitar filtros de salida.
 - Se puede controlar el flujo de potencia tanto real como reactiva.
-

2.5.2.3 Desventajas

Las desventajas principales del inversor con capacitores flotantes se pueden resumir de la siguiente manera:

- Se requiere una cantidad excesiva de capacitores de almacenamiento cuando la cantidad de niveles es grande. Los inversores que cuentan con gran cantidad de niveles de tensión son más difíciles de encapsular por los voluminosos capacitores de potencia, que también son más costosos.
- El control del inversor puede ser muy complicado, y la frecuencia de conmutación y las pérdidas por conmutación son altas para la transmisión de potencia real.

2.5.3 Inversor multinivel con diodo fijador (NPC)

El inversor multinivel de diodo anclado, fue presentado por Nabae en el año de 1980, se conoce también con el nombre de NPC (*Neutral Point Clamped*), o convertidor fijado al punto neutro, es una de las topologías multinivel más ampliamente estudiadas y aplicadas dentro del conjunto de inversores, y es considerada como el origen de la conversión multinivel reciente [2].

Un inversor multinivel de m niveles con diodo fijador (DCMLI, de *Diode-Clamped MultiLevel Inverter*) consiste, en forma típica, en $(m - 1)$ capacitores en el canal de CD y produce m niveles en el voltaje de fase.

En la figura 2.6 se presenta el inversor NPC de tres niveles de voltaje, en este caso particular, el voltaje continuo de entrada V_i , se divide mediante dos capacitores. El voltaje de salida para cada fase puede tomar tres niveles diferentes, $V_i/2$, 0 y $-V_i/2$. Los diodos conectados al punto neutro del bus de continua fijan el voltaje de bloqueo de los interruptores a una fracción del voltaje de CD de entrada, es decir, a $V_i/2$ en este caso.

Esta topología de inversor de tres niveles NPC tiene los siguientes componentes:

- Cuatro interruptores ($Q_1 - Q_4$).
- Cuadro diodos de circulación libre.
- Dos diodos fijadores de voltaje.
- Dos capacitores divisores de voltaje.
- Una fuente de CD.

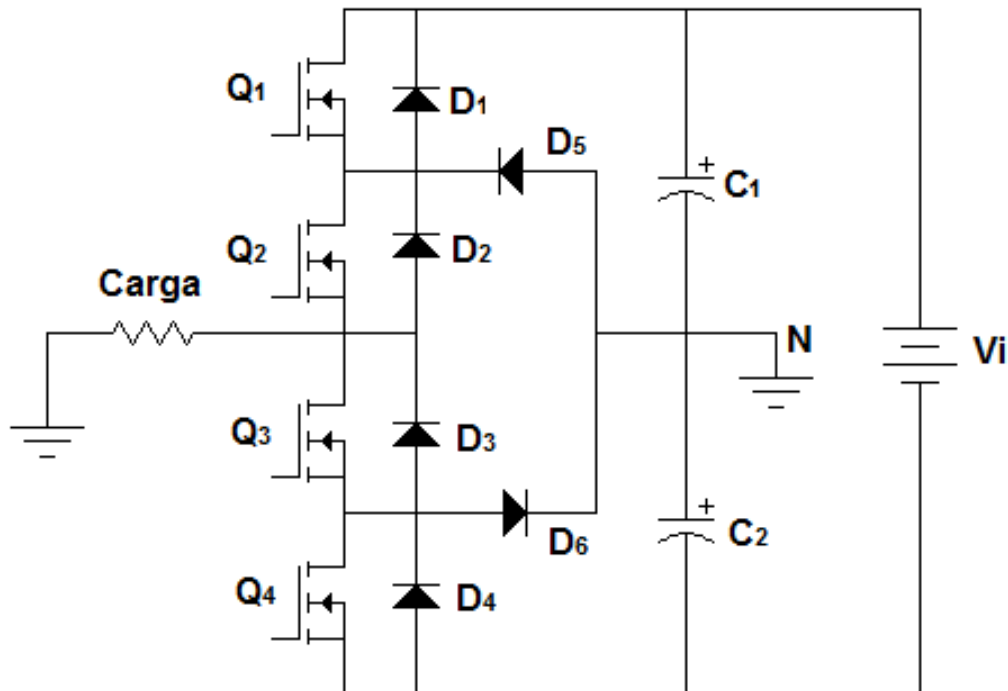


Figura 2.6. Topología de un inversor monofásico NPC de 3 niveles.

2.5.3.1 Principio de operación

El análisis se realiza en estado estacionario, se consideran los elementos que componen al inversor como dispositivos ideales.

- 1) Primera etapa de operación (t_0): En este instante conducen los diodos D_1 y D_2 , la corriente está en atraso respecto a la tensión. La tensión de salida es " $V_i/2$ " (figura 2.6 (a)).
- 2) Segunda etapa de operación (t_1): En t_1 la corriente es cero, se invierte la polaridad de los diodos D_1 y D_2 y comienzan a conducir los interruptores Q_1 y Q_2 . La tensión de salida es " $V_i/2$ " (figura 2.6 (b)).

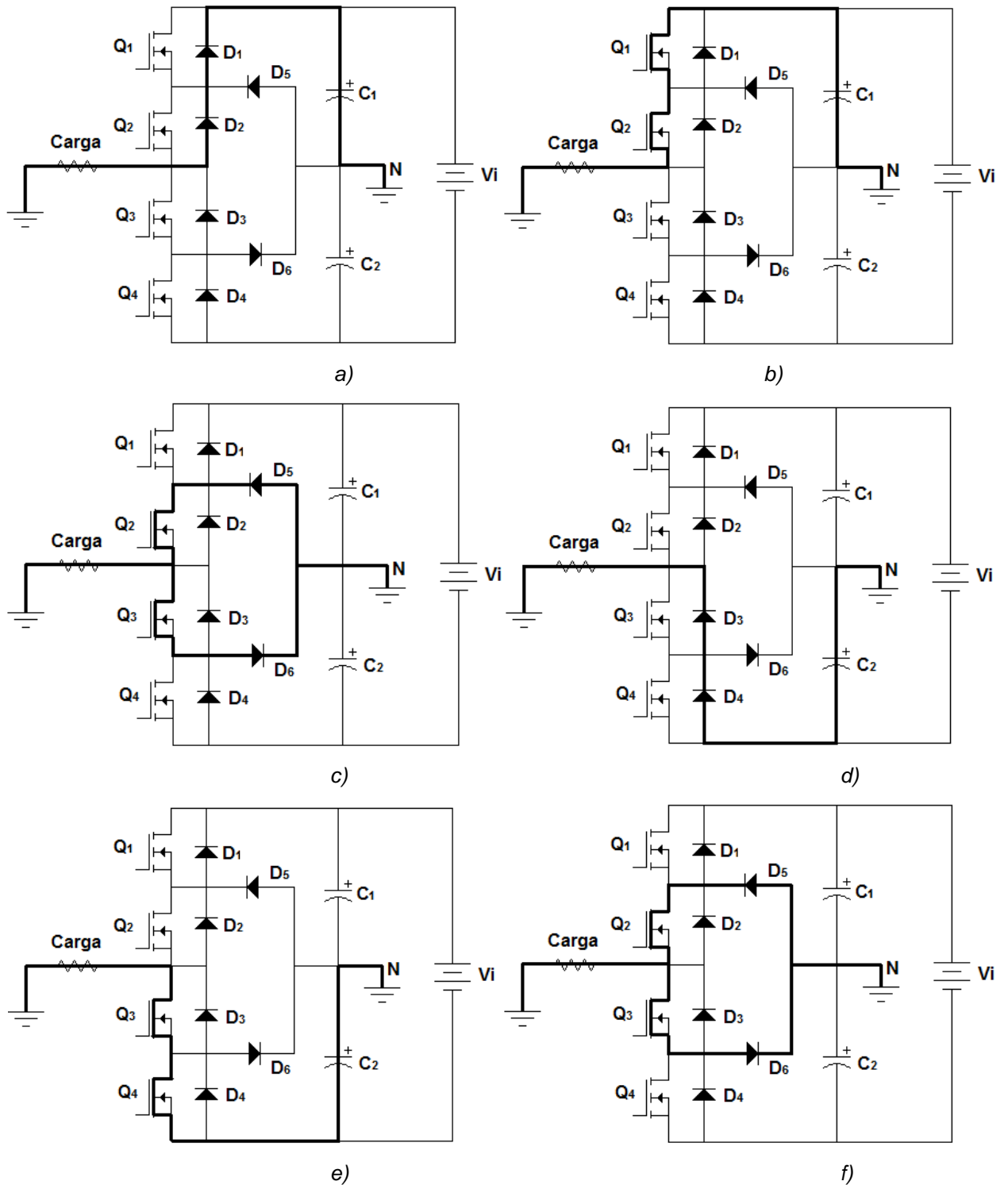


Figura 2.6 Flujo de corriente del inversor.

- 3) Tercera etapa de operación (t_2): en t_2 deja de conducir el interruptor Q_1 , sigue conduciendo Q_2 y comienza a conducir Q_3 . La corriente fluye por D_5 si esta va del punto neutro a la carga, y fluye por D_6 si esta va de la carga al punto neutro. La tensión de salida es cero (figura 2.6 (c)).
- 4) Cuarta etapa de operación (t_3): en t_3 conducen los diodos fijadores de tensión D_3 y D_4 debido a que la corriente no se ha invertido. La tensión de salida es “ $-V_i/2$ ” (figura 2.6 (d)).
- 5) Quinta etapa de operación (t_4): en t_4 deja de conducir el interruptor Q_2 , se invierte la polaridad de los diodos D_3 y D_4 , sigue conduciendo Q_3 y empieza a conducir Q_4 . La tensión de salida es “ $-V_i/2$ ” (figura 2.6 (e)).
- 6) Sexta etapa de operación (t_5): en t_5 deja de conducir Q_4 , sigue conduciendo Q_3 y comienza a conducir Q_2 . La corriente fluye por D_5 si esta va del punto neutro a la carga, y fluye por D_6 si esta va de la carga al punto neutro. La tensión de salida es cero (figura 2.6 (f)).

En la figura 2.7 se muestra la forma de onda de voltaje de salida para el inversor NPC de tres niveles, además del periodo de conducción de los interruptores.

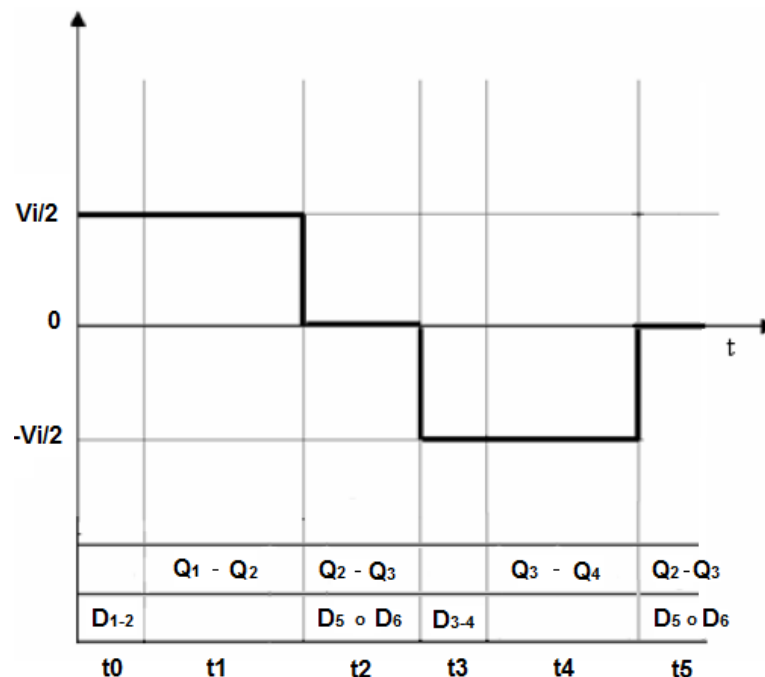


Figura 2.7. Forma de onda de salida del inversor NPC de tres niveles.

2.5.3.2 Propiedades del inversor NPC

Las propiedades principales son las siguientes:

1. **Especificación de alto voltaje para diodos de bloqueo.** Aunque cada dispositivo de conmutación solo debe bloquear un nivel de voltaje $V_{CD}/(m - 1)$, los diodos fijadores deben tener distintas especificaciones de bloqueo de voltaje en sentido inverso. En una rama de m niveles puede haber dos diodos, viendo cada uno un voltaje de bloqueo de:

$$V_D = \frac{m-1-k}{m-1} V_{CD} \quad (2.1)$$

donde:

m es la cantidad de niveles.

k va de 1 a $(m - 2)$.

V_{CD} es el voltaje total de enlace CD.

Si el voltaje nominal de bloqueo de cada diodo es igual al del dispositivo conmutador, la cantidad de diodos necesarios para cada fase es $N_D = (m - 1) \times (m - 2)$. Esta cantidad representa un aumento cuadrático en m . Así, para $m = 5$, $N_D = (5 - 1) \times (5 - 2) = 12$. Cuando m es suficientemente grande, la cantidad de diodos hace impráctica la implementación del sistema, lo cual limita, de hecho, la cantidad de niveles.

2. **Desbalance de voltaje de capacitores.** Como los niveles de voltaje en las terminales de capacitor son diferentes, las corrientes que suministran los capacitores también son diferentes. Cuando se opera con un factor de potencia unitario, el tiempo de descarga para la operación de inversor (o el tiempo de carga para la operación de rectificación), es distinto para cada capacitor. Ese perfil de carga de capacitor se repite cada medio ciclo, y el resultado son voltajes desbalanceados de capacitor, entre los distintos niveles. Este problema de desbalance de voltaje en un convertidor multinivel se puede

resolver con métodos como reemplazar capacitores por una fuente controlada de voltaje de CD constante, reguladores de voltaje PWM o con baterías.

2.5.3.3 Ventajas

Las ventajas principales del inversor NPC se pueden resumir de la siguiente manera:

- Cuando la cantidad de niveles es suficientemente alta, el contenido de armónicas es lo suficientemente bajo para evitar el uso de filtros.
- La eficiencia del inversor es alta, porque todos los dispositivos son conmutados a la frecuencia fundamental.
- El método de control es sencillo.

2.5.3.4 Desventajas

Las desventajas principales del inversor NPC se pueden resumir como sigue:

- Se requieren demasiados diodos fijadores cuando la cantidad de niveles es elevada.
- Es difícil controlar el flujo de la potencia real del convertidor individual, en sistemas con varios convertidores.

2.6 Características de inversores multinivel

Un inversor multinivel puede eliminar la necesidad del transformador elevador, y reducir las armónicas producidas. Aunque la estructura de inversor multinivel se introdujo al principio como medio de reducir el contenido de armónicas en la forma de onda de salida, se encontró que el voltaje de CD del canal se puede aumentar más allá del valor nominal de un dispositivo individual de potencia, usando una red de fijación de voltaje formada por diodos. Una estructura multinivel con más de tres niveles puede reducir en forma apreciable el contenido de armónicas. La propiedad intrigante que tienen las estructuras de inversor multinivel es su posibilidad de aumentar la especificación nominal de kilovolt-amperes (kVA), y también de mejorar

mucho el rendimiento sin tener que recurrir a las técnicas de PWM. Las características clave de una estructura multinivel son las siguientes:

- El voltaje y la potencia de salida aumentan con la cantidad de niveles. La adición de un nivel de voltaje implica la adición de un dispositivo de conmutación principal a cada fase.
- El contenido de armónicas disminuye a medida que la cantidad de niveles aumenta, y se reducen las necesidades de filtrado.
- Con niveles adicionales de voltaje, la forma de onda de voltaje tiene más ángulos de conmutación libre que se pueden seleccionar para eliminar armónicas.
- En ausencia de alguna técnica PWM, se pueden evitar pérdidas por conmutación. El aumento del voltaje y la potencia de salida no requieren un aumento en las especificaciones del dispositivo individual.
- Se incorpora a la estructura, ya sea por diodos fijadores o por capacitores, el voltaje estático y dinámico compartido entre los dispositivos de conmutación.
- Los dispositivos de conmutación no tienen problemas de voltaje compartido alguno. Por esta razón, los inversores multinivel se pueden aplicar con facilidad en altos voltajes, como impulsores de grandes motores y en fuentes de servicio eléctrico.
- El voltaje fundamental de salida del inversor se establece con el voltaje de CD de canal, V_{CD} , que se puede controlar con un enlace variable de CD.

2.7 Comparación de convertidores multinivel

Los convertidores multinivel pueden reemplazar a los sistemas existentes que usan convertidores multipulso tradicionales sin necesidad de transformadores. Para un sistema trifásico, la relación entre la cantidad m de niveles y la cantidad p de pulsos se puede formular como $p = 6 \times (m - 1)$. Los tres convertidores mencionados anteriormente tienen el potencial para aplicaciones en sistemas de alto voltaje y alta potencia, como un SVG (generador de VAR estáticos), sin problema de desbalance

de voltaje, porque el SVG no toma potencia real. El convertidor NPC es más adecuado para el sistema de interconexión espalda con espalda funcionando como un controlador unificado de flujo de potencia. Los otros dos convertidores también pueden ser adecuados para la interconexión espalda con espalda, pero necesitan más conmutación por ciclo y técnicas más avanzadas de control para balancear el voltaje. Los inversores multinivel pueden encontrar aplicaciones potenciales en impulsores de velocidad ajustable, en los que el uso de ellos no sólo puede resolver los problemas de armónicas y de inducción electromagnética, sino también pueden evitar fallas de motor inducidas por la tasa dV/dt de conmutación en alta frecuencia.

La tabla 2.1 compara los requisitos de componentes por rama de fase entre los tres convertidores multinivel. Se supone que todos los dispositivos tienen la misma especificación de voltaje, pero no necesariamente la misma especificación de corriente. El inversor en cascada usa un puente completo en cada nivel, en comparación con la versión en medio puente para los otros dos tipos. El inversor en cascada requiere la cantidad mínima de componentes, y tiene el potencial para aplicaciones de interconexión con el servicio eléctrico, por sus posibilidades de aplicar técnicas de modulación y conmutación suave.

Tabla 2.1. Comparación de necesidades de componentes por rama de los tres inversores.

Tipo de Convertidor	Con diodo fijador	Con capacitores flotantes	Inversor Puentes H en cascada
Dispositivos de interrupción principal	$(m - 1) \times 2$	$(m - 1) \times 2$	$(m - 1) \times 2$
Diodos principales	$(m - 1) \times 2$	$(m - 1) \times 2$	$(m - 1) \times 2$
Diodos de fijación	$(m - 1) \times (m - 2)$	0	0
Capacitores de canal de CD	$(m - 1)$	$(m - 1)$	$(m - 1)/2$
Capacitores de balance	0	$(m - 1) \times (m - 2)/2$	0

2.8 Técnicas de modulación

En muchas aplicaciones industriales, para controlar el voltaje de salida de los inversores, se necesita con frecuencia 1) hacer frente a las variaciones de voltaje de

entrada de CD, 2) regular el voltaje de los inversores y 3) satisfacer los requisitos de control de voltaje y frecuencia constantes. Hay varias técnicas para variar la ganancia del inversor. El método más eficiente de controlar la ganancia (y el voltaje de salida) es incorporar control por modulación por ancho de pulso (PWM) en los inversores.

La modulación por ancho de pulso PWM es una técnica en la que se modifica el ciclo de trabajo de una señal periódica. El ciclo de trabajo de una señal periódica es el ancho relativo de su parte positiva en relación al período. Matemáticamente:

$$D = \frac{t}{T} \quad (2.2)$$

donde:

D es el ciclo de trabajo.

t es el tiempo en que la función es positiva (ancho de pulso).

T es el periodo de la función.

En general, podemos clasificar las técnicas de modulación en dos grandes grupos: las técnicas de modulación analógicas y las digitales.

2.9 Técnicas de modulación analógicas

La construcción típica de un circuito PWM se lleva a cabo mediante un comparador con dos entradas y una salida. Una de las entradas se conecta a un oscilador de onda triangular, mientras que la otra queda disponible para la señal moduladora (generalmente una señal sinusoidal). En la salida, la frecuencia es generalmente igual a la de la señal triangular y el ciclo de trabajo está en función de la portadora. El proceso de demodulación es aplicar un filtro pasa bajas a la señal modulada.

En la figura 2.8 se muestra un ejemplo de cómo una señal portadora (señal triangular) es comparada con una señal moduladora (señal sinusoidal) y a la salida se obtiene una variación en el ancho de pulso.

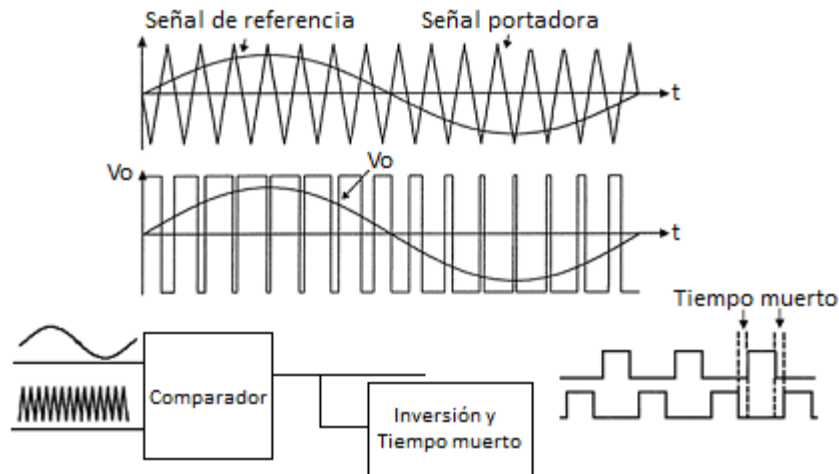


Figura 2.8. Ejemplo de la técnica PWM analógica.

Las técnicas analógicas PWM que se usan con frecuencia son:

1. Modulación por ancho de un solo pulso.
2. Modulación por ancho de pulsos múltiples.
3. Modulación por ancho de pulso sinusoidal (SPWM).
4. Modulación por ancho de pulso sinusoidal modificado.
5. Control por desplazamiento de fase.

2.10 Técnicas de modulación digitales

Las técnicas de modulación digital son muy diferentes a las analógicas, en donde en estas últimas se hacía la comparación entre una señal portadora y una moduladora analógicas. En las técnicas de modulación digital se utilizan sistemas digitales como son los microcontroladores (MCU), controladores digitales de señales (DSC) o procesadores digitales de señales (DSP), realizando cálculos internos haciendo uso de la arquitectura de los sistemas digitales para resolver operaciones matemáticas en tiempos reducidos.

2.10.1 Modulación por vectores espaciales (SVM)

La modulación por vectores espaciales (SVM, de *Space Vector Modulation*) es muy distinta de los métodos por PWM. Con las PWM, se puede imaginar al inversor como

tres etapas separadas de activación simétrica o en contrafase, que modelan cada onda de fase en forma independiente. Sin embargo, la SVM maneja al inversor como una sola unidad; en forma específica, el inversor puede activarse a 27 estados únicos (en el caso de tres niveles de tensión a la salida). La modulación se obtiene por el estado de conmutación del inversor. Las estrategias de control se implementan con sistemas digitales. La SVM es una técnica de modulación digital, en la que el objetivo es generar voltajes PWM en la línea de carga que en promedio sean iguales a determinados voltajes de línea (o de referencia) de carga. Esto se hace en cada periodo de muestreo seleccionando en forma adecuada los estados de los interruptores del inversor, y calculando el tiempo adecuado para cada estado.

Para entender la técnica SVM aplicada a un inversor de 3 niveles, se hará cita primeramente a la técnica SVM de 2 niveles estudiada en el Instituto Tecnológico de Sonora por alumnos de la carrera Ing. En Electrónica [10].

2.10.2 SVM de 2 niveles

La técnica de modulación por ancho de pulso de vectores espaciales trabaja al inversor como una unidad y se basa en el hecho de que un solo vector es capaz de representar las tres fases de un sistema trifásico, este vector se crea a partir de los tiempos de trabajo de los estados de conmutación del inversor.

Para saber más a detalle cómo implementar la técnica de Modulación de Vectores Espaciales SVM para un inversor trifásico de 2 niveles, puede consultarse [10].

2.10.3 SVM de 3 niveles

En este apartado se explicará la técnica de modulación SVM de tres niveles, que presenta una descripción formal de los estados de conmutación del convertidor y muestra sus posibles combinaciones.

2.10.3.1 Descripción general

En el caso del convertidor de tres niveles, disponemos de tres posibles conexiones al bus de continua (p , o y n) para cada fase (a , b , c). Esto supone un total de $3^3 = 27$ estados de conmutación diferentes, cada uno con sus voltajes de línea asociados. Estos estados de conmutación se presentan en la tabla 2.2 (se considera que el voltaje de salida es $V_{on} = V_{pn}/2$).

Se puede construir el diagrama SVM colocando dos de los voltajes en los ejes de voltaje de línea del diagrama, trazando una línea perpendicular en cada eje, y donde se crucen las perpendiculares será el punto de voltaje de línea real asociado al estado de conmutación. En la figura 2.9 se muestra el ejemplo del estado de conmutación **poo**.

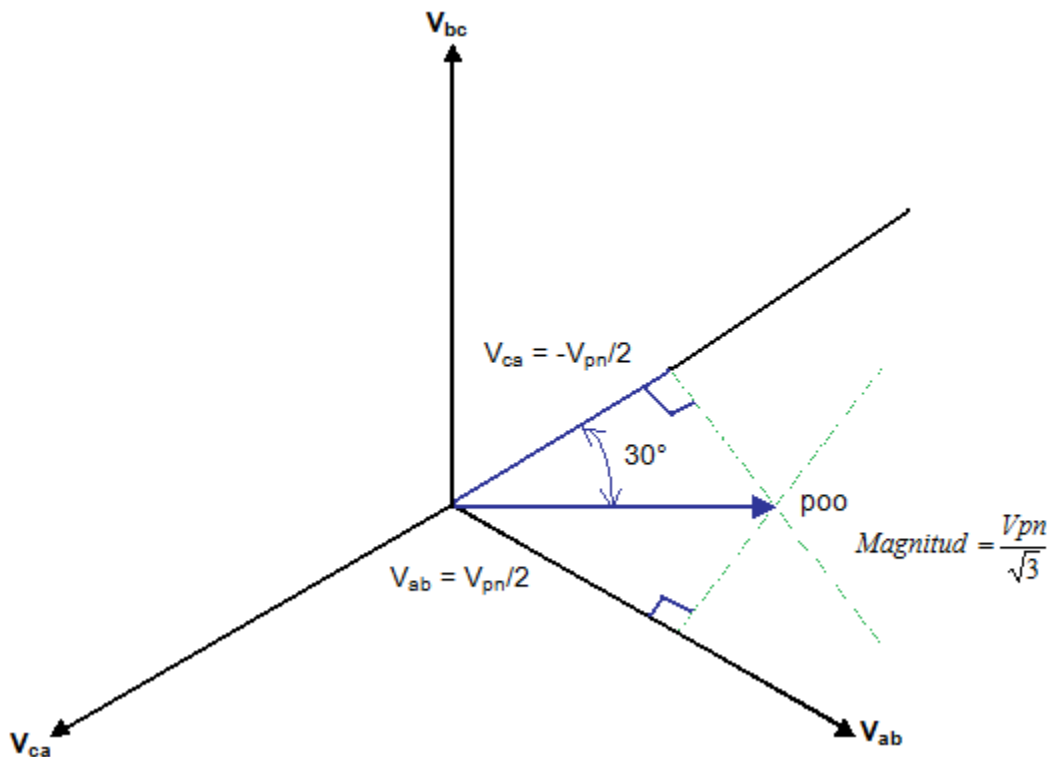


Figura 2.9. Construcción de los vectores de estado de tensión asociados al estado de conmutación **p00** del inversor de tres niveles.

Tabla 2.2. Estados de conmutación y voltajes de línea asociados al inversor de tres niveles.

Nº Estado	Conexiones			V_{ab}	V_{bc}	V_{ca}
	Fase a	Fase b	Fase c			
1	n	n	n	0	0	0
2	n	n	o	0	$-V_{pn}/2$	$V_{pn}/2$
3	n	n	p	0	$-V_{pn}$	V_{pn}
4	n	o	n	$-V_{pn}/2$	$V_{pn}/2$	0
5	n	o	o	$-V_{pn}/2$	0	$V_{pn}/2$
6	n	o	p	$-V_{pn}/2$	$-V_{pn}/2$	V_{pn}
7	n	p	n	$-V_{pn}$	V_{pn}	0
8	n	p	o	$-V_{pn}$	$V_{pn}/2$	$V_{pn}/2$
9	n	p	p	$-V_{pn}$	0	V_{pn}
10	o	n	n	$V_{pn}/2$	0	$-V_{pn}/2$
11	o	n	o	$V_{pn}/2$	$-V_{pn}/2$	0
12	o	n	p	$V_{pn}/2$	$-V_{pn}$	$V_{pn}/2$
13	o	o	n	0	$V_{pn}/2$	$-V_{pn}/2$
14	o	o	o	0	0	0
15	o	o	p	0	$-V_{pn}/2$	$V_{pn}/2$
16	o	p	n	$-V_{pn}/2$	V_{pn}	$-V_{pn}/2$
17	o	p	o	$-V_{pn}/2$	$V_{pn}/2$	0
18	o	p	p	$-V_{pn}/2$	0	$V_{pn}/2$
19	p	n	n	V_{pn}	0	$-V_{pn}$
20	p	n	o	V_{pn}	$-V_{pn}/2$	$-V_{pn}/2$
21	p	n	p	V_{pn}	$-V_{pn}$	0
22	p	o	n	$V_{pn}/2$	$V_{pn}/2$	$-V_{pn}$
23	p	o	o	$V_{pn}/2$	0	$-V_{pn}/2$
24	p	o	p	$V_{pn}/2$	$-V_{pn}/2$	0
25	p	p	n	0	V_{pn}	$-V_{pn}$
26	p	p	o	0	$V_{pn}/2$	$-V_{pn}/2$
27	p	p	p	0	0	0

De la misma manera, se muestra el diagrama SVM normalizado en la figura 2.10, de los vectores de tensión asociados a todos los estados de conmutación del convertidor de tres niveles. Para obtener los voltajes de línea asociados a cada

vector sólo hace falta proyectar ortogonalmente sobre los ejes correspondientes y multiplicar después por V_{PN} . Análogamente, para obtener los voltajes de fase sólo hace falta proyectar sobre los ejes correspondientes y multiplicar por $\frac{V_{pn}}{\sqrt{3}}$.

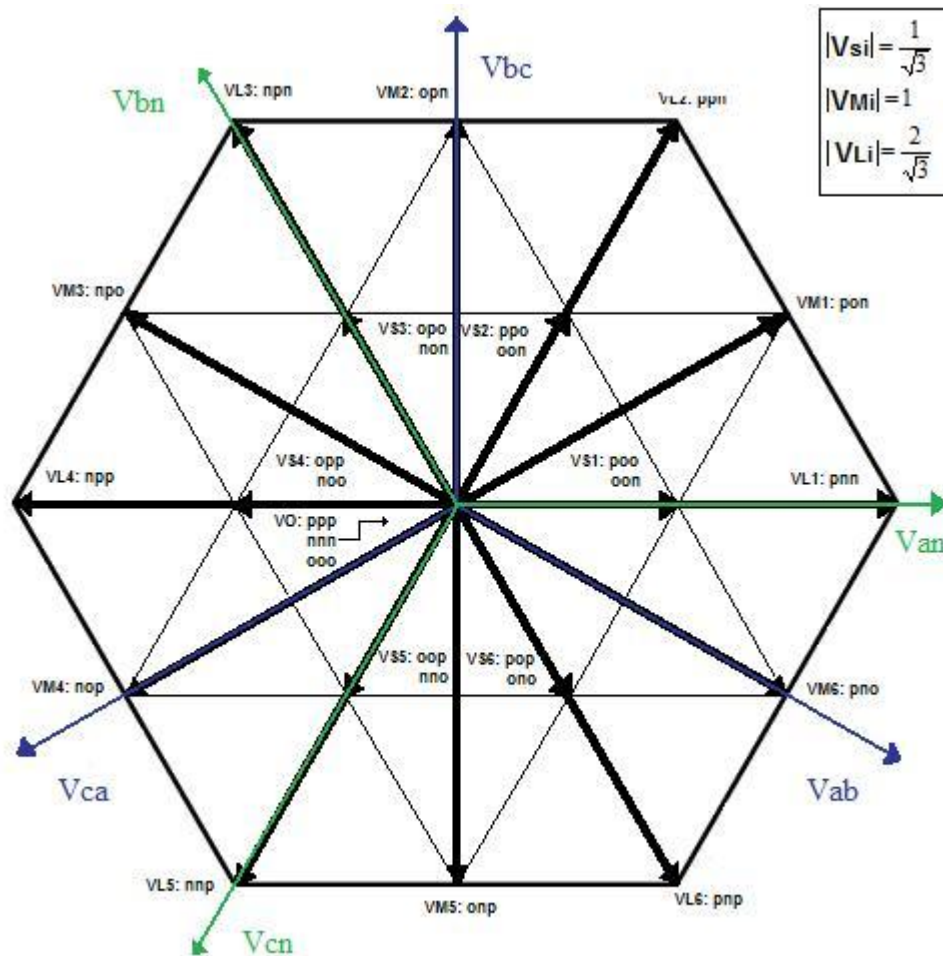


Figura 2.10. Diagrama normalizado SVM asociado a los 27 estados de conmutación para el inversor de tres niveles.

En este caso, aparecen cuatro tipos de vectores en función de su magnitud:

- El vector cero (V_0) que es el estado de tensión asociado a los tres estados de conmutación **ppp**, **nnn** y **ooo**. Su magnitud es cero.

- b) Los vectores pequeños identificados con las letras \mathbf{V}_{SI} , correspondientes cada uno de ellos a dos estados de conmutación. Su magnitud normalizada es $\frac{1}{\sqrt{3}}$. Estos vectores de estados de tensión se forman cuando el estado de conmutación tiene dos conexiones de las fases a los puntos del bus de continua y uno de los tres al menos es una conexión a cero.
- c) Los vectores medianos identificados por las letras \mathbf{V}_{MI} , y correspondientes a un único estado de conmutación. Su magnitud normalizada es 1. Estos estados de tensión se logran cuando el estado de conmutación tiene una conexión a **p** una a **n** y una a **o**.
- d) Los vectores grandes identificados por las letras \mathbf{V}_{LI} , y correspondientes a un único estado de conmutación. Su magnitud normalizada es $\frac{2}{\sqrt{3}}$. Estos estados de tensión se logran cuando únicamente tenemos conexión a **p** y a **n**. estos estados de conmutación son los mismos de los que disponemos en el caso del convertidor de dos niveles.

Es importante destacar que en este caso, no sólo los vectores cero pueden lograrse con diferentes estados de conmutación, sino que también los vectores pequeños es posible lograrlos con diferentes configuraciones de conexión de las tres fases a los puntos de tensión del bus de continua.

La magnitud del vector de referencia puede tomar valores normalizados entre 0 y 1. El valor normalizado de 1 corresponde con el máximo voltaje de línea sin distorsión que puede alcanzarse con este convertidor.

Los diferentes vectores de tensión definen áreas (triángulos) que en el caso de los triángulos grandes se representan con la letra **i** (dado que estos triángulos corresponden con los triángulos encontrados en el diagrama SVM del convertidor de dos niveles) y en el caso de los triángulos pequeños se representan con la letra **j**, tal como se puede observar en la figura 2.11.

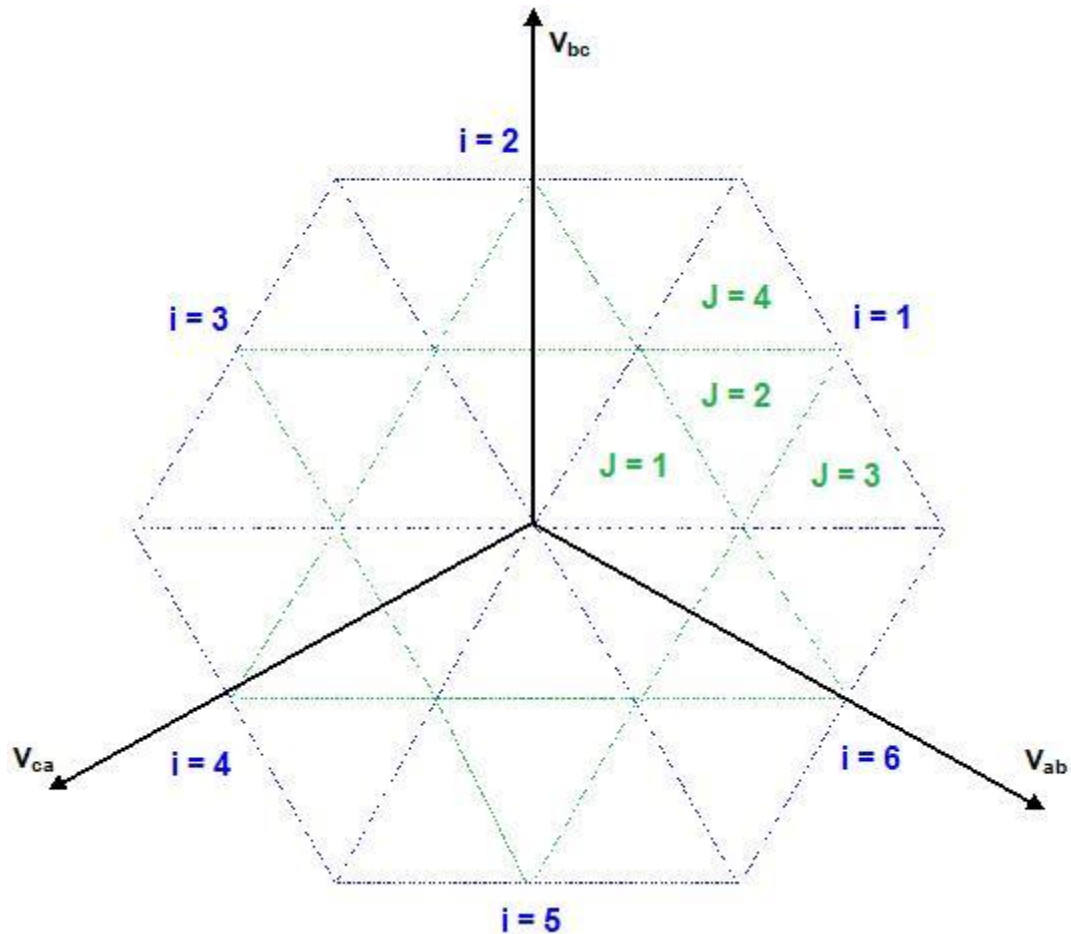


Figura 2.11. Notación relativa a los triángulos definidos por los vértices de los vectores de estado: triángulos grandes (azul) y triángulos pequeños (verde).

2.10.3.2 Simetría

El diagrama SVM del convertidor de tres niveles también presenta la simetría señalada en los triángulos i como es en el caso del convertidor de dos niveles. Esto implica que para el estudio de la estrategia de modulación se puede restringir a uno de los triángulos grandes i . Ahora, se centrará el estudio en el triángulo $i = 1$ para describir las diferentes modulaciones propuestas en base al diagrama general SVM.

2.10.3.3 Estrategia para sintetizar el vector de referencia

En este caso se dispone de más vectores de estado y más estados de conmutación relativos a los mismos para sintetizar el vector de referencia, comparado con el caso

del convertidor de dos niveles. En cada periodo de conmutación (T_s) hace falta escoger cómo aproximar el vector de referencia que representa el sistema de tensiones trifásicas balanceadas que se quiere lograr a la salida del convertidor. En primer lugar se necesita elegir los vectores de estado que se usarán para sintetizar el vector de referencia en cada periodo de conmutación. En segundo lugar hace falta escoger los estados de conmutación que se utilizarán para materializar los estados de tensión escogidos anteriormente. En el siguiente apartado se muestra la estrategia utilizada para elegir el vector de estado ideal para sintetizar el vector de referencia.

2.10.3.4 Modulación SVM-NTV (*Nearest Three Vectors*)

Esta es la técnica más comúnmente empleada a la hora de sintetizar el vector de referencia. Recibe su nombre NTV (*Nearest Three Vectors*) del hecho que para cada periodo de conmutación, se escogen los tres vectores de tensión más cercanos que definen los vértices del triángulo pequeño, j , donde se encuentra el vector de referencia. Estos son los tres vectores de tensión más próximos al vector de referencia y, por lo tanto, serán los que intuitivamente mejor se aproximen en cada periodo de conmutación.

Con estos tres vectores de estado de tensión haremos una combinación lineal para aproximar el vector de referencia, de modo que se cumpla que el voltaje promedio en un periodo de conmutación sea la secuencia de los vectores de estado de tensión la misma que para la del vector de referencia.

2.10.3.5 Cálculo de los ciclos de trabajo para los vectores de estado de tensión

El vector de referencia, en un periodo de conmutación cualquiera, puede estar situado en cualquiera de los cuatro triángulos pequeños j . Consideraremos cada uno de estos cuatro casos y se presentarán las expresiones que definen los ciclos de trabajo de los vectores de estado relevantes en cada caso. Estas ecuaciones se deducen a partir de las siguientes ecuaciones, ya vistas en el caso del convertidor de dos niveles:

$$V_{ref}(t) = m \cdot e^{j\theta} = d_1 \cdot V_{estado1} + d_2 \cdot V_{estado2} + d_3 \cdot V_{estado3} \quad (2.3)$$

$$d_1 = \frac{T_1}{T_s}; d_2 = \frac{T_2}{T_s}; d_3 = \frac{T_3}{T_s}$$

Donde $V_{estado1}$, $V_{estado2}$ y $V_{estado3}$ son los tres vectores de estado con los que se aproxima el vector de referencia y d_1 , d_2 y d_3 son sus respectivos ciclos de trabajo.

Triángulo $j = 1$

En la figura 2.12 se ilustra el caso en el que el vector de referencia se ubica en el triángulo $j = 1$.

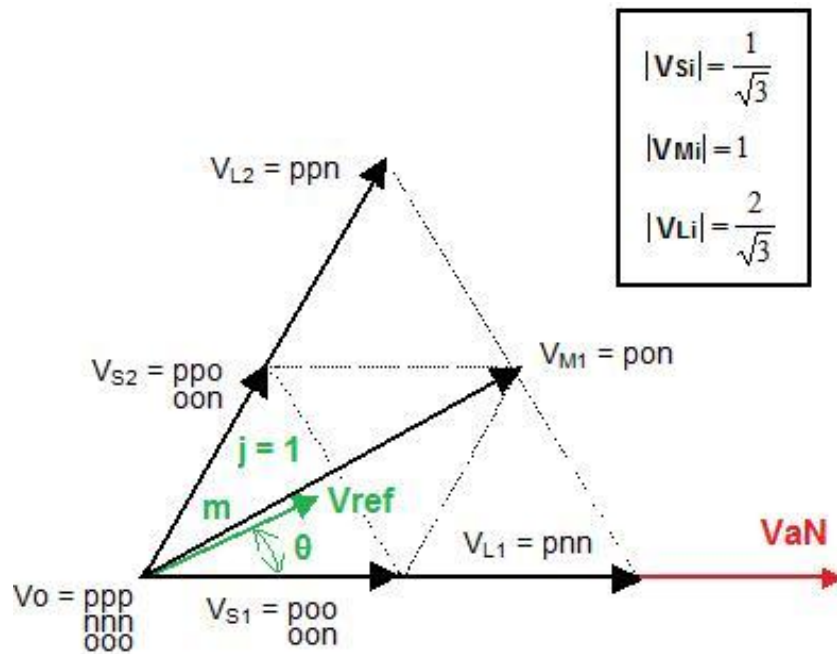


Figura 2.12. Vector de referencia en el triángulo $j = 1$.

En este caso, se eligen los vectores de estado de tensión V_{S1} , V_{S2} y V_0 . Resolviendo las ecuaciones de la expresión general obtenemos:

$$d_{S1} = m \cdot (\sqrt{3} \cdot \cos(\theta) - \sin(\theta))$$

$$d_{S2} = 2 \cdot m \cdot \sin(\theta) \quad (2.4)$$

$$d_0 = 1 - d_{S1} - d_{S2} = 1 - m \cdot (\sqrt{3} \cdot \cos(\theta) + \sin(\theta))$$

Triángulo j = 2

A continuación se muestra el caso donde el vector de referencia está situado en el triángulo j = 2.

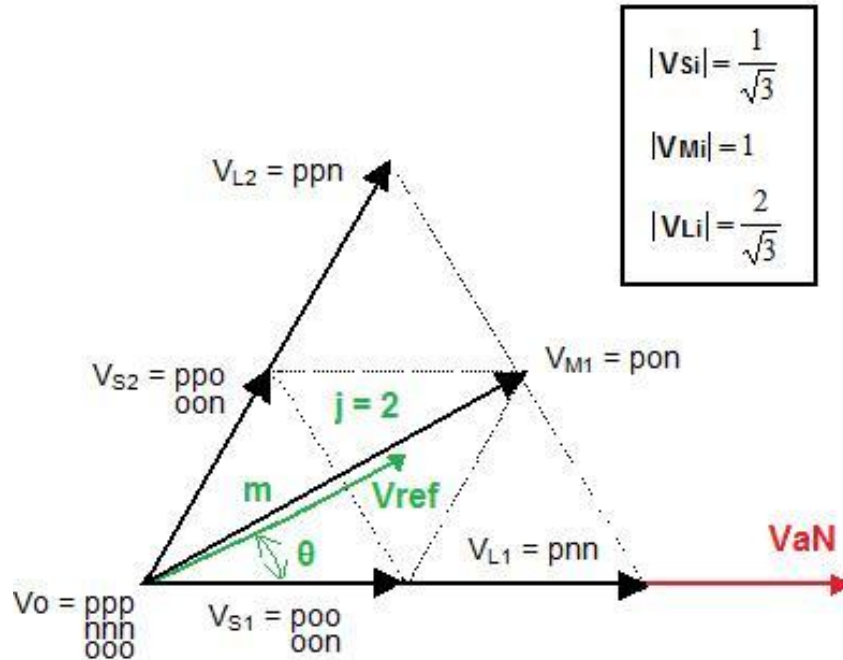


Figura 2.13. Vector de referencia en el triángulo j = 2.

En este caso, se eligen los vectores de estado de tensión V_{S1} , V_{S2} , V_{M1} . Resolviendo las ecuaciones de la expresión general se obtiene:

$$\begin{aligned}
 d_{S1} &= 1 - 2 \cdot m \cdot \sin(\theta) \\
 d_{S2} &= 1 + m \cdot (\sin(\theta) - \sqrt{3} \cdot \cos(\theta)) \\
 d_{M1} &= 1 - d_{S1} - d_{S2} = -1 + m \cdot (\sin(\theta) - \sqrt{3} \cdot \cos(\theta))
 \end{aligned}
 \tag{2.5}$$

Triángulo j = 3

A continuación se muestra el caso donde el vector de referencia está situado en el triángulo j = 3.

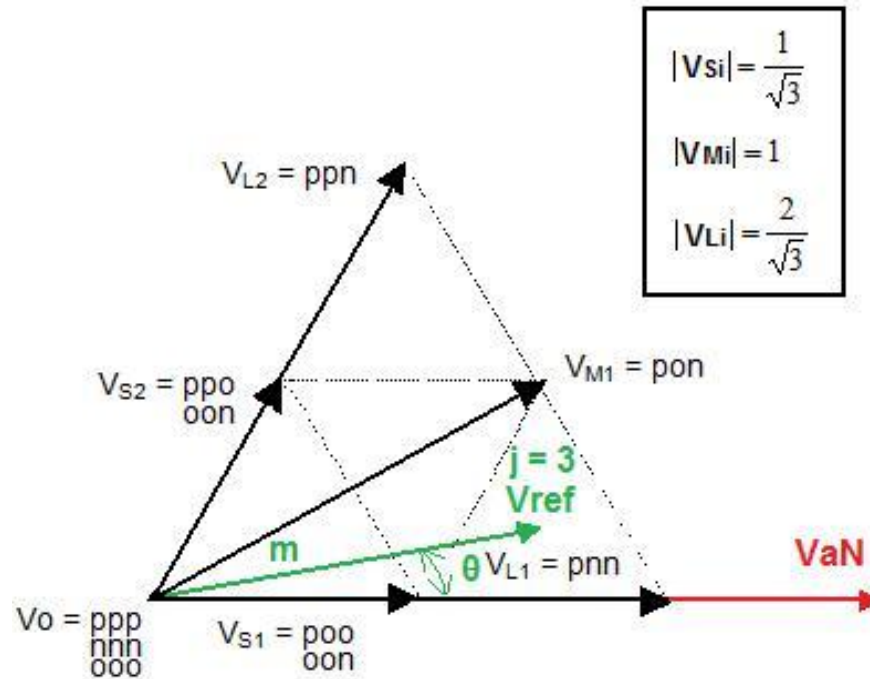


Figura 2.14. Vector de referencia en el triángulo $j = 3$.

En este caso, se eligen los vectores de estado de tensión V_{S1} , V_{M1} , V_{L1} . Resolviendo las ecuaciones de la expresión general se obtiene:

$$\begin{aligned}
 d_{L1} &= -1 + m \cdot (\sqrt{3} \cdot \cos(\theta) - \sin(\theta)) \\
 d_{M1} &= 2 \cdot m \cdot \sin(\theta) \\
 d_{S1} &= 1 - d_{L1} - d_{M1} = 2 - m \cdot (\sqrt{3} \cdot \cos(\theta) + \sin(\theta))
 \end{aligned}
 \tag{2.6}$$

Triángulo $j = 4$

A continuación se muestra el caso donde el vector de referencia está situado en el triángulo $j = 4$.

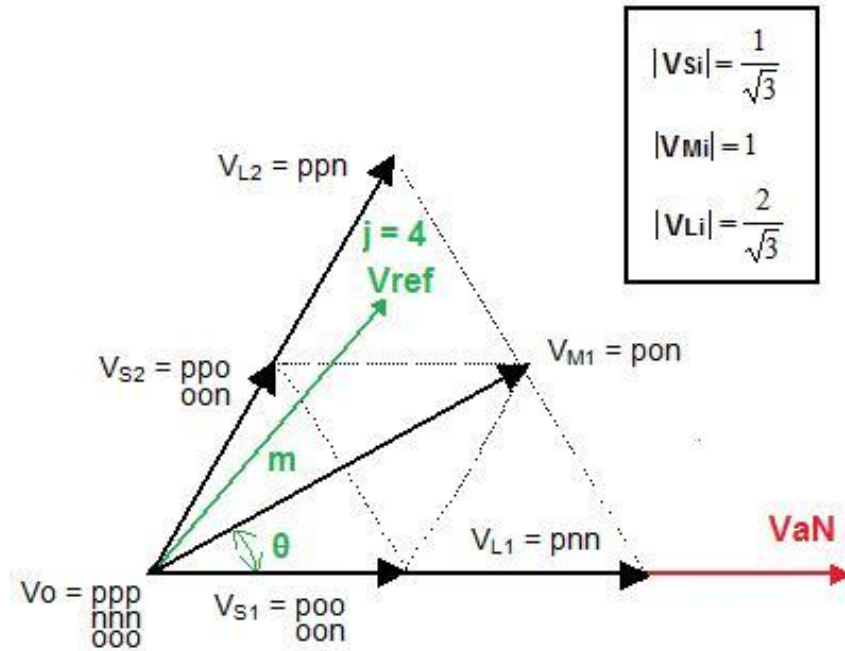


Figura 2.15. Vector de referencia en el triángulo $j = 4$.

En este caso, se eligen los vectores de estado de tensión V_{S2} , V_{M1} , V_{L2} . Las expresiones para los ciclos de trabajo son las mismas que en el caso $j = 3$ pero sustituyendo θ por $\frac{\pi}{3} - \theta$.

$$\begin{aligned}
 d_{L2} &= -1 + m \cdot (\sqrt{3} \cdot \cos(\frac{\pi}{3} - \theta) - \sin(\frac{\pi}{3} - \theta)) \\
 d_{M1} &= 2 \cdot m \cdot \sin(\frac{\pi}{3} - \theta) \\
 d_{S2} &= 1 - d_{L2} - d_{M1}
 \end{aligned} \tag{2.7}$$

2.10.3.6 Elección de los estados de conmutación

Una vez escogidos los vectores de estados de tensión que se emplearán para sintetizar el vector de referencia, se escogerán los estados de conmutación asociados que se emplearán para lograr éstos. En los casos en donde sólo hay un estado de conmutación para el vector de tensión (V_{Mi} y V_{Li}) no hay duda posible, pero en los casos que se dispone de diferentes alternativas (V_0 y V_{Si}), hace falta escoger cuales estados de conmutación se emplearán.

A continuación se presentan diferentes soluciones propuestas referentes a la elección de los estados de conmutación para la estrategia SVM-NTV.

2.10.3.7 Una conexión a tensión cero

Para lograr los diferentes vectores de estados de tensión V_{S1} , se tienen dos posibles estados de conmutación. Una de ellas tiene dos fases conectadas a su punto \bullet y el otro tiene sólo una fase conectada a este punto. En este caso se eligen para lograr los vectores de estado de tensión V_{S1} el estado de conmutación con una única conexión. El vector V_0 se podrá implementar con el estado **ppp** o **nnn**, indistintamente.

La tabla 2.3 muestra los estados de conmutación elegidos para cada uno de los vectores de tensión.

Tabla 2.3. Estados de conmutación escogidos para una conexión a tensión cero.

Vector de estado de tensión	Estado de conmutación
V_0	ppp ó nnn
V_{S1}	onn
V_{S2}	ppo
V_{S3}	non
V_{S4}	opp
V_{S5}	nno
V_{S6}	pop

Ciclo de trabajo de los estados de conmutación

El ciclo de trabajo de los estados de conmutación escogidos será igual a los ciclos de trabajo de sus respectivos vectores de tensión. En el caso del vector V_0 , dependiendo de si usamos la conexión **ppp** o **nnn** tendremos $d_{PPP} = d_0$ y $d_{NNN} = 0$ o $d_{NNN} = d_0$ y $d_{PPP} = 0$. El resto de los ciclos de trabajo de los estados de conmutación no escogidos es cero.

2.10.3.8 Dos conexiones a tensión cero

En este caso, se eligen para los vectores V_{S1} aquellos estados de conmutación con dos fases conectadas al punto o . Para el vector V_0 escogemos el estado de conmutación ooo .

Tabla 2.4. Estados de conmutación escogidos para dos conexiones a tensión cero.

Vector de estado de tensión	Estado de conmutación
V_0	ooo
V_{S1}	po0
V_{S2}	oon
V_{S3}	opo
V_{S4}	noo
V_{S5}	oop
V_{S6}	ono

Ciclo de trabajo de los estados de conmutación

El ciclo de trabajo de los estados de conmutación escogidos será igual a los ciclos de trabajo de sus respectivos estados de tensión. El resto de los ciclos de trabajo de los estados de conmutación no escogidos son cero.

Hasta ahora se ha explicado lo referente a las topologías de inversores y las técnicas de modulación, en los siguientes apartados los temas de interés son las fuentes de alimentación para dichos circuitos.

2.11 Fuente de alimentación conmutada

Las fuentes en modo conmutado tienen alta eficiencia y pueden alimentar una gran corriente de carga a bajo voltaje. Hay cuatro configuraciones comunes para la operación en modo conmutado o PWM de la etapa inversora:

- Flyback o en retorno.

- Directa.
- *Push – Pull* o en contrafase.
- Medio puente.
- Puente completo.

Para el caso de este estudio, se hará énfasis en la fuente con el Convertidor *Push – Pull*.

2.11.1 Convertidor *Push – Pull*

La topología *Push – Pull* o contrafase, consta de 2 transistores conmutando en estado complementario, en donde su dren es conectado a un transformador, el cual puede realizar la función de reductor, elevador, o mantener una relación de vueltas constante para que el voltaje sea el mismo en el secundario. A la salida del secundario las señales se hacen pasar por un puente rectificador que posteriormente es filtrado para poder obtener una señal limpia a la salida.

Una de las características importantes de esta topología es el aislamiento que produce el transformador, razón por la cual es utilizada en la alimentación de las ramas del inversor trifásico de tres niveles, ya que necesita haber aislamiento en éste para no producir cortocircuitos en ninguna de sus ramas.

La figura 2.16 muestra el diagrama del Convertidor *Push – Pull*.

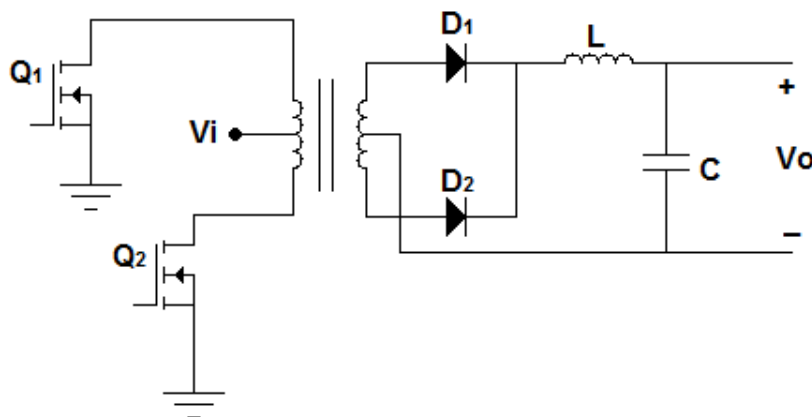


Figura 2.16. Diagrama del Convertidor *Push – Pull*.

2.12 Controlador digital de señales

Para la técnica de modulación SVM estudiada en este capítulo, es necesario contar con sistemas digitales de alta velocidad para su procesamiento. Debido a la alta velocidad que manejan los controladores digitales de señales, es posible realizar operaciones matemáticas en un lapso reducido de tiempo. Su programación es muy sencilla, además de que no necesita cambios en el hardware cuando se tiene que hacer una modificación, y sólo se tiene que reescribir el código.

Un controlador digital de señales (DSC, por sus siglas en inglés *Digital Signal Controller*) es un dispositivo creado por la empresa Microchip Technology Inc., al cual le nombró con el nombre de dsPIC. Este es un microcontrolador de 16 bits que integra de manera compacta todas las capacidades de control un microcontrolador con las capacidades de computación y rendimiento de un procesador de señales (DSP, por sus siglas en inglés *Digital Signal Processor*).

La gama de los dispositivos dsPIC se divide en dos clases o familias:

- Familia dsPIC30F
- Familia dsPIC33F

Para la implementación de la técnica SVM se utilizó un dispositivo de la familia dsPIC30F, por lo cual se hará referencia en este apartado solamente a esta familia.

2.12.1 La familia dsPIC30F

Los dispositivos pertenecientes a esta familia se pueden dividir en tres categorías dependiendo de la aplicación a la que está orientada su arquitectura, estas categorías son:

- Dispositivos de propósito general.
 - Dispositivos para control de motores y manejo de energía.
 - Dispositivos para control de sensores.
-

El controlador digital utilizado en la realización del trabajo pertenece a la categoría de control de motores y manejo de energía, ya que el trabajo que realizará el microcontrolador es el control de un motor trifásico.

2.12.2 dsPIC30F4011

El controlador digital de señales, modelo dsPIC30F4011, al igual que toda la familia dsPIC30F cuenta con una arquitectura tipo Harvard modificada, en donde las memorias de programa y de datos se encuentran separadas físicamente, esto se puede observar en la figura 2.17.

La arquitectura Harvard modificada permite dos diferentes tamaños de palabra, uno de 16 bits para datos y otro de 24 bits para instrucciones. Esto permite eficiencia al conjunto de instrucciones y un procesamiento más rápido, ya que puede acceder a la memoria de programa para leer la siguiente instrucción y simultáneamente estar ejecutando una instrucción que accede a la memoria de datos.

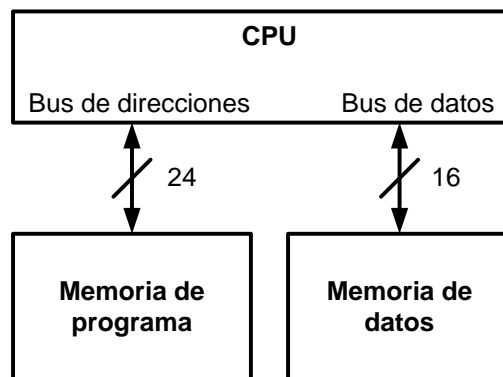


Figura 2.17. Diagrama a bloques de la arquitectura tipo Harvard para dsPIC30F.

El diagrama de pines del dsPIC30F4011 que se utilizó en la realización de este proyecto se presenta en la figura 2.18.

Entre las principales características y recursos de hardware del dsPIC30F4011, se encuentran las siguientes:

- Velocidad de procesamiento 30 MIPS (*Millions of instructions Per Second*).
-

- 48 KB de memoria de programa.
- 2048 Bytes de memoria de datos.
- 5 contadores de 16 bits con la posibilidad de concatenarse para trabajar con 32 bits.
- Módulo PWM con 6 canales de salida.
- Módulo ADC de 10 bits de resolución con la capacidad de operar a 1 MSPS (*Millions of Samples Per Second*).
- Módulo de comparadores analógicos.
- Módulo de comunicación serial para el manejo de periféricos externos.

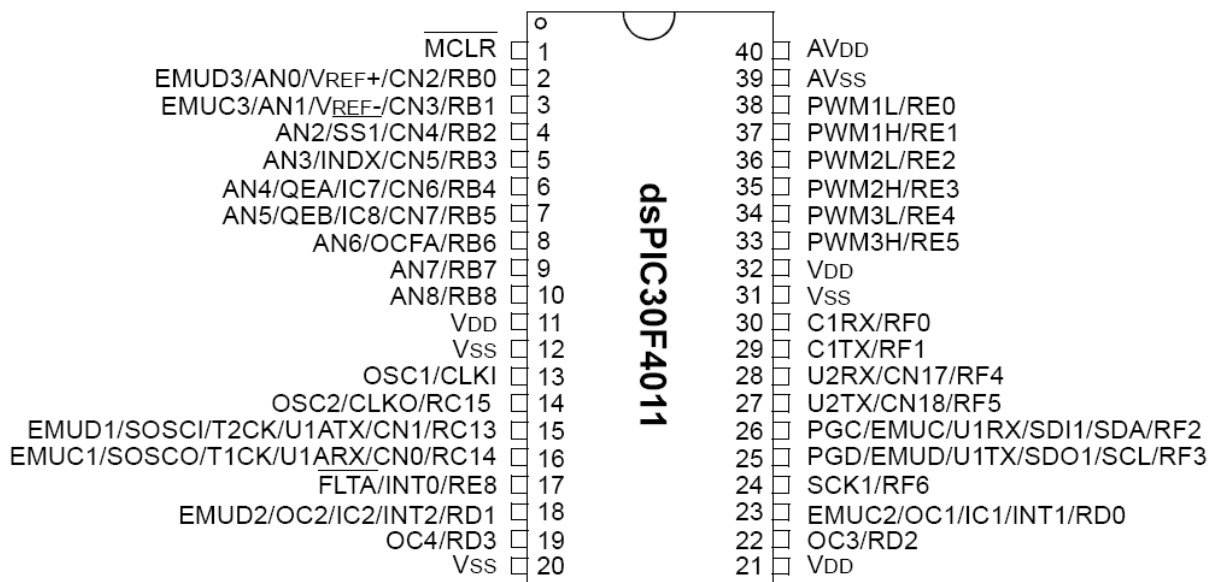


Figura 2.18. Diagrama de pines del dsPIC30F4011.

El diagrama a bloques del controlador digital dsPIC30F4011 donde se muestran todos los recursos de hardware con los que cuenta se muestra en la figura 2.19.

De los recursos de hardware mostrados en el diagrama a bloques de la figura 2.19, los más importantes para la realización del presente trabajo son el rápido oscilador interno, ya que permite trabajar a una frecuencia de oscilación máxima de 120 MHz, el módulo temporizador, que permite aplicar cada vector con su respectivo ciclo de

trabajo, el módulo ADC de 10 bits y 1 MSPS. Enseguida se analizan a mayor detalle estos recursos y se explica su funcionamiento.

FIGURE 1-1: dsPIC30F4011 BLOCK DIAGRAM

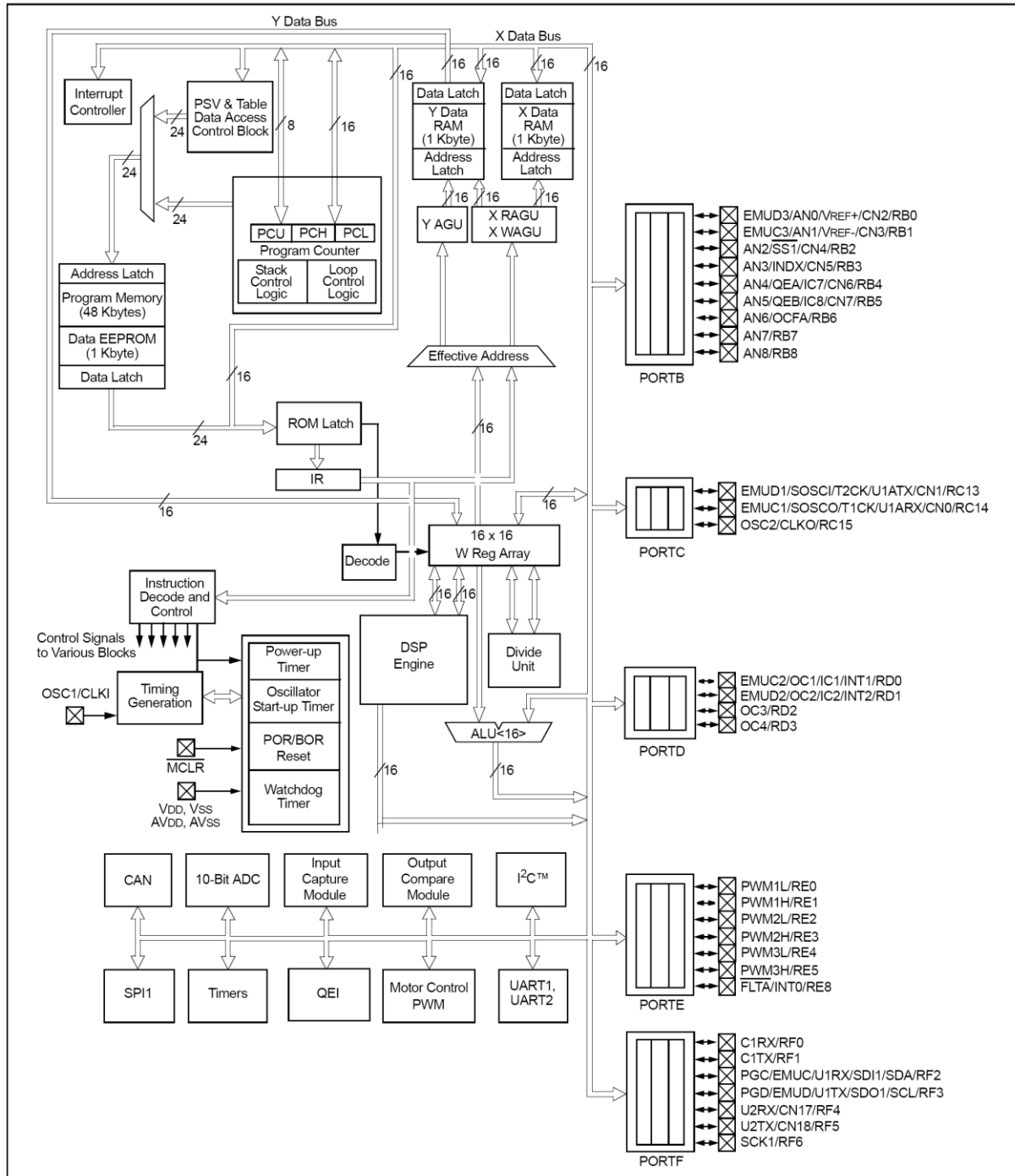


Figura 2.19. Diagrama a bloques del dsPIC30F4011.

2.12.2.1 Oscilador

El controlador digital dsPIC30F4011 cuenta con tres fuentes para el oscilador:

- 1) Oscilador primario con dos modos:
 - HS, para frecuencias de 10 a 25 MHz.
 - EC, para oscilador externo.
- 2) FRC, oscilador rápido interno RC (7.37 MHz).
- 3) LPRC, oscilador interno RC de baja velocidad (512 kHz).

En la figura 2.20 se muestra el diagrama a bloques del oscilador del controlador digital dsPIC30F4011, en el se pueden ver las tres diferentes fuentes mencionadas, se muestra además un bloque llamado PLL, este es un bloque multiplicador de la frecuencia de oscilación, al estar activado este bloque la señal del oscilador primario o del oscilador interno se multiplica por 4, 8 o 16 para obtener las frecuencias necesarias para el módulo Temporizador y el ADC, además de la señal de reloj para el funcionamiento del controlador digital.

En este controlador digital, un ciclo máquina equivale a 4 ciclos de oscilación, esto es:

$$F_{CY} = \frac{F_{OSC}}{4} \quad (2.8)$$

Lo que quiere decir, que si trabajamos con una frecuencia de oscilación de 120 MHz, la frecuencia real con la que el controlador digital ejecuta cada instrucción es 30 MHz (30 MIPS), de ahí que pueda realizar 30 millones de instrucciones por segundo. El periodo de oscilación es igual al inverso de la frecuencia de instrucción, esto es:

$$T_{CY} = \frac{1}{F_{CY}} \quad (2.9)$$

Lo que quiere decir que cada instrucción que se ejecute en un ciclo máquina, si se trabaja a 120 MHz, tardara 33.33 ns en ejecutarse.

Figure 7-1: Oscillator System Block Diagram

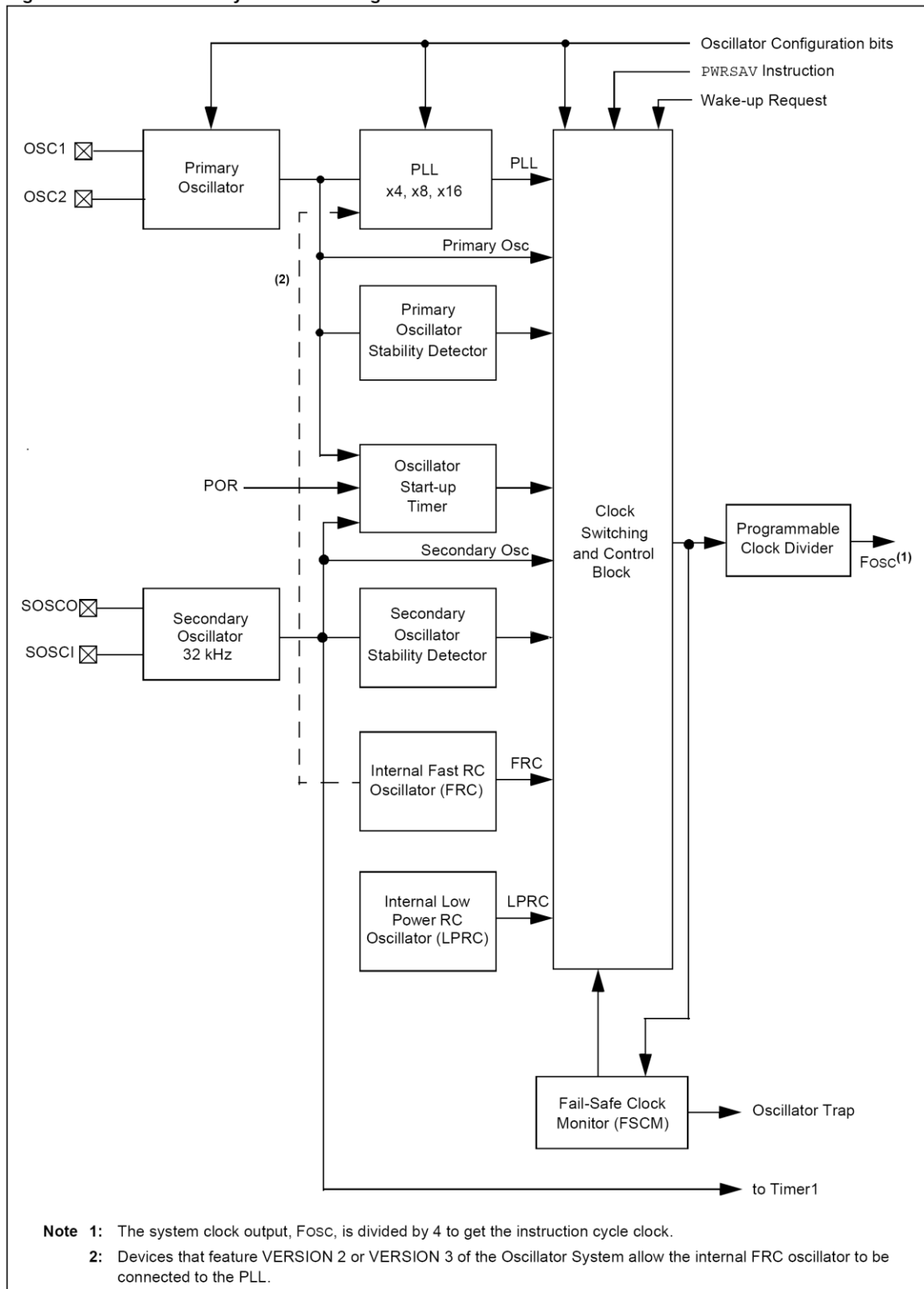


Figura 2.20. Diagrama a bloques del oscilador del dsPIC30F4011.

2.12.2.2 Módulo temporizador

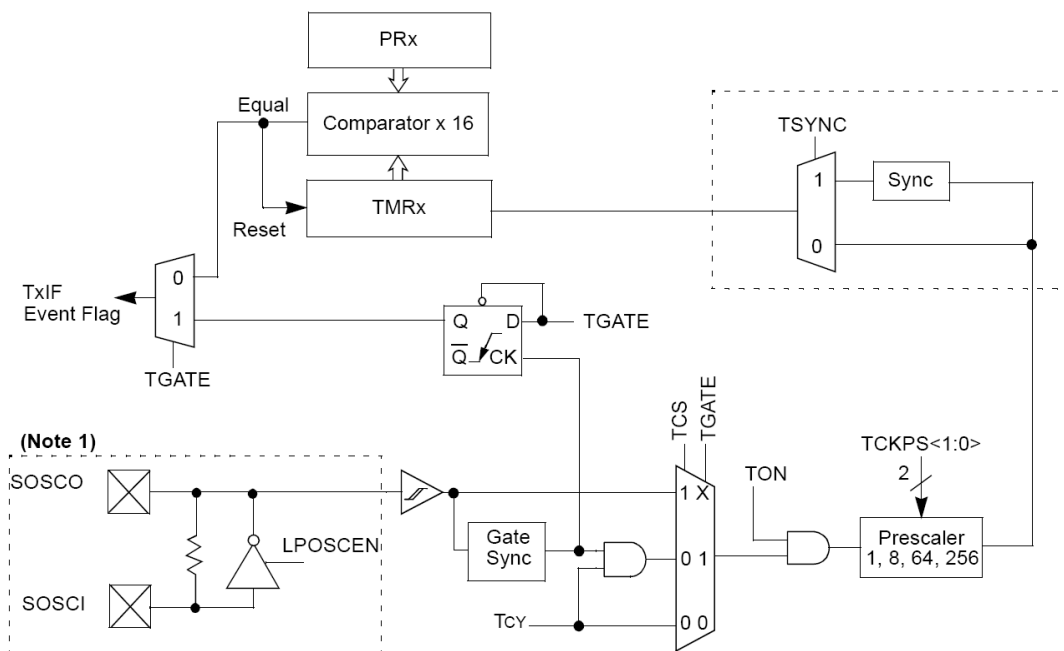
El módulo de temporización del dsPIC30F4011 se compone de 5 Temporizadores de 16 bits cada uno, los cuales pueden ser clasificados en 3 categorías:

- 1) Temporizador clase A.
- 2) Temporizador clase B.
- 3) Temporizador clase C.

2.12.2.3 Temporizador clase A

El temporizador clase A es un temporizador de propósito general, y en la mayoría de los dsPIC30F el temporizador clase A es el TIMER1. Entre sus características podemos encontrar las siguientes:

- Puede operar desde el oscilador de baja potencia (32 kHz).
- Puede operar en modo asíncrono desde una fuente de reloj externa.



Note 1: Refer to Section 7. "Oscillator" for information on enabling the LP Oscillator.

Figura 2.21. Diagrama a bloques del temporizador clase A.

En la figura 2.21 se muestra el diagrama a bloques de un temporizador clase A.

2.12.2.4 Temporizador clase B

En la mayoría de los dsPIC30F, los temporizadores clase B son los TIMER2 y TIMER4. Entre sus características puede encontrarse:

- Puede concatenarse con un temporizador clase C para formar un temporizador de 32 bits.

En la figura 2.22 se muestra el diagrama a bloques de un temporizador clase B.

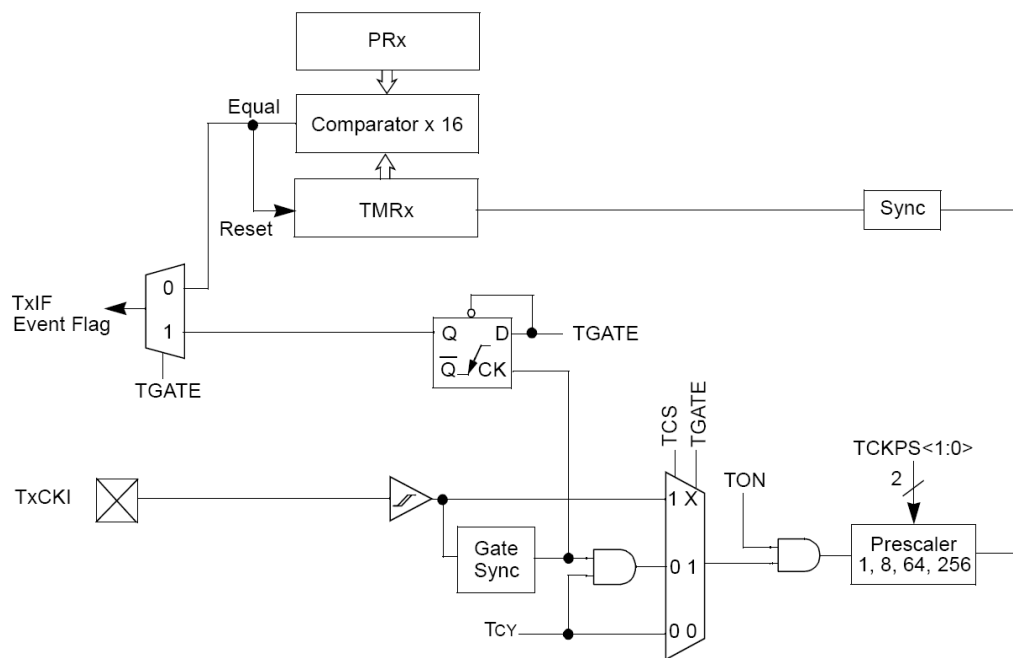


Figura 2.22. Diagrama a bloques del temporizador clase B.

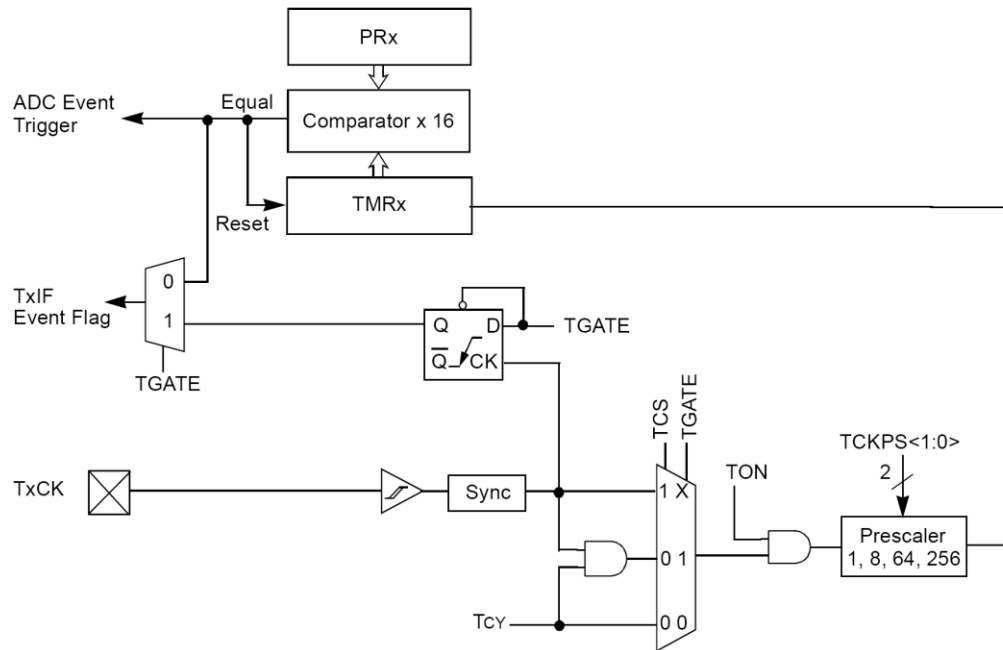
2.12.2.5 Temporizador clase C

En la mayoría de los dsPIC30F, los temporizadores clase C son los TIMER3 y TIMER5. Entre sus características puede encontrarse:

- Puede concatenarse con un temporizador clase B para formar un temporizador de 32 bits.

- En algunos dispositivos, al menos un temporizador clase C tiene la capacidad de disparar el convertidor analógico digital (ADC).

En la figura 2.23 se muestra el diagrama a bloques de un Temporizador clase C.



Note: In certain variants of the dsPIC30F family, the TxCK pin may not be available. Refer to the device data sheet for the I/O pin details. In such cases, the timer must use the system clock ($F_{osc}/4$) as its input clock, unless it is configured for 32-bit operation.

Figura 2.23. Diagrama a bloques del Temporizador clase C.

2.12.2.6 Registros de control del temporizador

El módulo temporizador tiene 3 registros de control, los cuales son el TxCON, TMRx y PRx.

TxCON

El registro de control TxCON controla todo lo referente a la configuración del temporizador, desde que el temporizador se enciende, su prescalador, su sincronización, de dónde proviene la fuente de reloj, y si se quiere concatenar con

otro tipo de temporizador. La “x” indica que puede ser cualquiera de los cinco temporizadores, por ejemplo T1CON, T2CON, etc.

TMRx

El registro de control TMRx es un contador, que se va incrementando cada vez que el temporizador detecta que ha pasado un ciclo de instrucción (dependiendo de la configuración del prescalador). Es un registro de 16 bits, por lo que este puede contar desde 0x0000 hasta 0xFFFF, cuando el registro TMRx es mayor a 0xFFFF se enciende una bandera lo que indica que puede haber una interrupción por desborde del temporizador.

PRx

El registro de control PRx es un registro de 16 bits de lectura y escritura, el cual sirve para poder poner una marca que indica hasta donde se desea que el registro TMRx cuente. Por ejemplo: si se requiere que el registro TMRx no cuente hasta 0xFFFF sino hasta 0x1000, entonces en el registro PRx se carga el valor 0x1000, por lo tanto, cuando el TMRx supere ese valor, se encenderá una bandera que indica que puede haber una interrupción.

2.12.2.7 Módulo ADC

El controlador digital dsPIC30F4011 contiene un convertidor analógico a digital (ADC, por sus siglas en inglés *Analog to Digital Converter*) con las características que se enlistan a continuación.

- 10 bits de resolución.
 - Conversión por aproximaciones sucesivas.
 - 1 MSPS (*Millions of Samples Per Second*).
 - 9 entradas analógicas.
 - 4 circuitos de muestreo y retención.
 - Capacidad de muestrear 2 entradas analógicas al mismo tiempo.
-

En la figura 2.24 se puede observar que el Convertidor Analógico Digital cuenta con 4 circuitos de muestreo y retención, y la capacidad de muestrear 2 entradas analógicas al mismo tiempo, así como los 9 canales analógicos (AN0 – AN8).

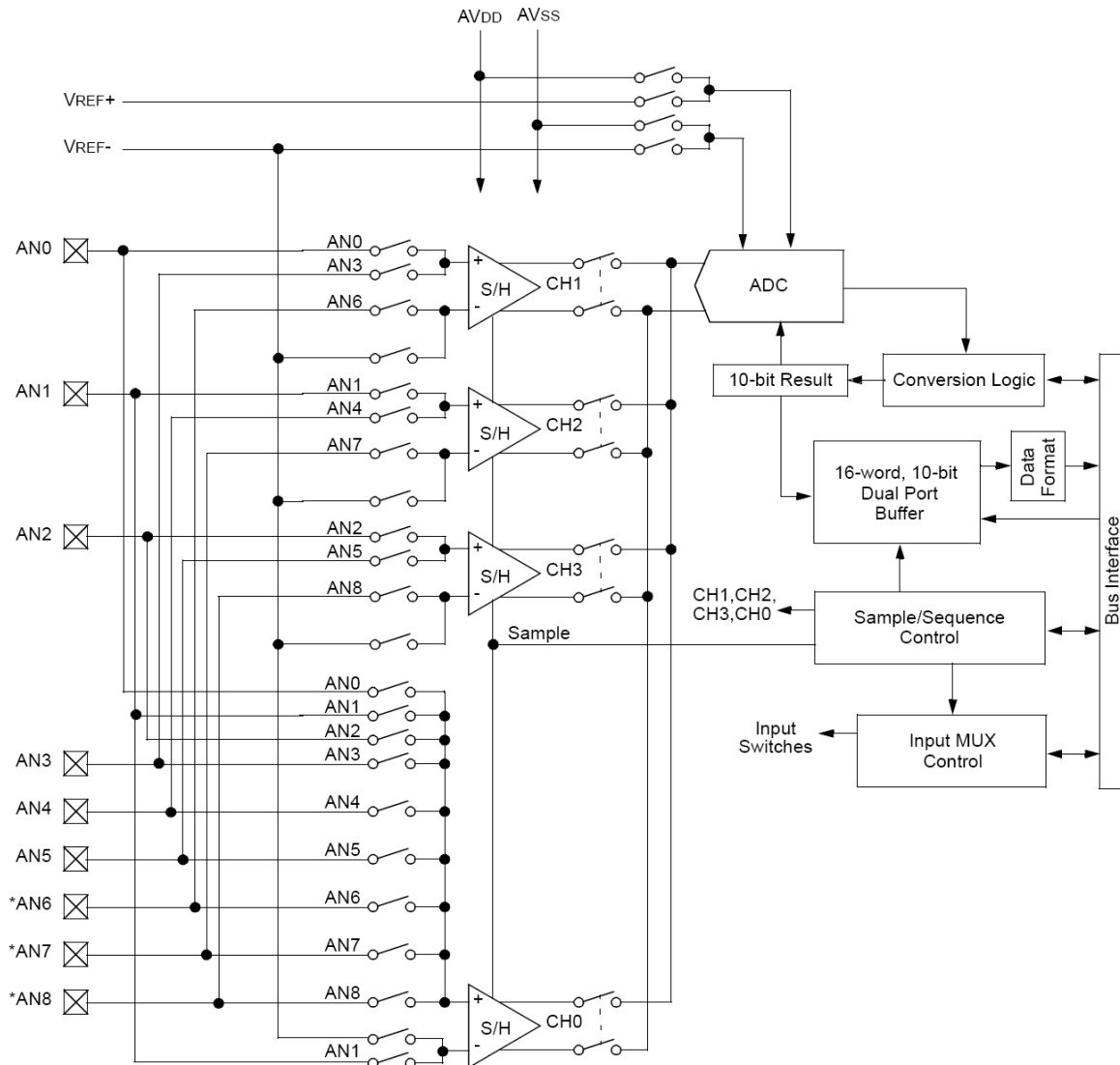


Figura 2.24. Diagrama a bloques del ADC del dsPIC30F4011.

2.12.2.8 Registros de control del ADC

Los registros de control del Convertidor Analógico Digital del dsPIC30F4011 se muestran en la tabla 2.5.

Tabla 2.5. Registros de control del ADC del dsPIC30F4011.

Registro	Descripción
ADCON1	Registro de control del ADC
ADCON2	Registro de control del ADC
ADCON3	Registro de control del ADC
ADCHS	Registro de selección del canal de entrada
ADPCFG	Registro de configuración de puertos
ADCSSL	Registro de selección de entradas a escanear

A continuación se muestra una descripción de los registros de control del Convertidor Analógico - Digital.

ADCON1

El registro ADCON1 es un registro de 16 bits en el cual se configura cuándo se enciende el módulo ADC, el formato de salida que tendrá la conversión, el disparo de conversión que se tendrá, como se hará el muestro y retención, y también se indica cuando una conversión ha terminado.

ADCON2

El registro ADCON2 es un registro de 16 bits en el cual se pueden elegir los voltajes de referencia que se tendrán, cuáles son los canales que se utilizarán para convertir, como se llenarán los registros binarios de resultado de conversión y periodo de muestreo y retención.

ADCON3

El registro ADCON3 es un registro de 16 bits en el cual se elige la velocidad de conversión que se requiere para el ADC, además de la fuente de reloj con la que trabajará el convertidor.

ADCHS

En este registro se puede elegir el canal físico por el cual se hará la conversión analógico - digital.

ADPCFG

En este registro se puede configurar los canales de entrada como analógicos o digitales.

ADCSSL

En este registro se pueden configurar los canales que van a escanearse para la conversión analógico - digital.

Cabe destacar que para el desarrollo de este trabajo, para la conversión analógico - digital, sólo se necesita un canal de conversión para una señal digital que representa el índice de modulación del sistema.

2.13 MPLAB IDE

El software MPLAB es un entorno de desarrollo IDE (por sus siglas en inglés *Integrated Development Enviroment*), es un programa de distribución gratuita por parte de Microchip para el desarrollo con PIC's y dsPIC's. Las posibilidades de este entorno de desarrollo son muy amplias y van desde la creación de programas fuente hasta la simulación paso a paso del programa creado y su efecto en los registros. En este entorno de desarrollo es posible programar los PIC's y dsPIC's tanto en lenguaje ensamblador como en lenguaje C, si se cuenta con el compilador adecuado para el tipo de dispositivo, en este proyecto se utiliza el lenguaje C para el desarrollo del programa de la Modulación por Vectores Espaciales (SVM). Para consultar la forma de utilizar y crear proyectos en el software MPLAB se pueden consultar [10] y [11].

2.14 Lenguaje C

C es un lenguaje de programación estructurado de propósito general, es considerado un lenguaje de nivel intermedio ya que incluye características que permiten su uso a un nivel más bajo.

El lenguaje C se desarrolló inicialmente en los laboratorios Bell entre 1969 y 1973 por Dennis Ritchie. Es el resultado de dos lenguajes anteriores, el BCPL y el B, que se desarrollaron también en Bell Laboratories. En 1978 se publicó una descripción definitiva del lenguaje por Brian Kernighan y Dennis Ritchie.

La mayoría de las implementaciones comerciales de C, difieren en algo de la definición original de Kernighan y Ritchie. Esto ha creado algunas pequeñas incompatibilidades entre las diferentes implementaciones del lenguaje, disminuyendo la portabilidad [10].

En este trabajo se utiliza el compilador MPLAB C30, por lo tanto sólo se desarrolla lo relacionado con el lenguaje C para este compilador. Para conocer más acerca del lenguaje C en general se puede consultar [12].

2.15 MPLAB C30

MPLAB C30 es un compilador de lenguaje C optimizado para el uso de controladores digitales de señales, especialmente dsPIC30F y MCU de 16 bits PIC24. Este software es una aplicación de ambiente Windows que provee una plataforma de desarrollo de lenguaje C.

El MPLAB C30 realiza la compilación de archivos de lenguaje C produciendo archivos en lenguaje ensamblador, con éste se puede crear el archivo ejecutable que se puede cargar al DSC.

Para más información sobre el funcionamiento y las características del compilador MPLAB C30 se pueden consultar [10] y [13].

Referencias

- [1] MUHAMMAD H. Rashid, "Electrónica de Potencia, Circuitos, Dispositivos y Aplicaciones." Prentice Hall, Tercera edición, 2004.
- [2] A. NABAE, I. Takahashi, and H. Akagy, "A Neutral-Point Clamped PWM Inverter." IEEE Transactions on Industrial Applications, 1981.
- [3] MARTÍN Prats M. Ángeles, "Nuevas Técnicas de Modulación Vectorial para Convertidores Electrónicos de Potencia Multinivel" Tesis de Doctor por la Universidad de Sevilla, Sevilla, Junio de 2003.
- [4] PEÑUELAS Machado José Ángel, "Algoritmo de Seguimiento del Punto de Máxima Potencia para Sistemas Fotovoltaicos en Cd. Obregón, Sonora", Tesis de Ingeniero en Electrónica, Instituto Tecnológico de Sonora, Cd. Obregón, Sonora, Octubre de 2008.
- [5] ELGUETA Díaz Cristian Marcelo, "Aplicación de un Inversor Multinivel como Variador de Frecuencia en un Motor de Inducción Trifásico", Tesis de Maestro en Ciencias de la Ingeniería, Pontificia Universidad Católica de Chile, Santiago de Chile, Julio de 2005.
- [6] MARTINEZ Hernández Domitilo, "Control Digital Para Convertidor Multinivel Alimentado con Energía Solar", Tesis de Maestro en Ciencias con especialidad en Ingeniería Electrónica, Universidad de las Américas Puebla, Cholula, Puebla, Mayo de 2006.
- [7] T. A. Meynard, H. Foch, "Multi-Level Conversion: High Voltage Choppers and Voltage Source Inverters", IEEE Power Electronics Specialists Conference Vol. 1, 1992.
-

[8] RODRIGUEZ J., Jih-Sheng Lai, Fang Zheng Peng, "Multilevel Inverters: a survey of topologies, controls and applications", Industrial Electronics, IEEE Transactions on Volume 49, Issue 4, Agosto de 2002.

[9] CAMPOS Valenzuela Rodrigo Andrés, "Evaluación Técnica y Económica de Tres tipos de Inversores de Tecnología Multinivel para Aplicaciones en Media Tensión", Tesis de Ingeniero Civil, Pontificia Universidad Católica de Valparaíso-Chile, Chile, Noviembre de 2007.

[10] IRIBE Quintero Víctor, "Técnica de Modulación por Vectores Espaciales Implementada en un dsPIC30F3010", Tesis de Ingeniero en Electrónica, Instituto Tecnológico de Sonora, Cd. Obregón, Sonora, Diciembre de 2007.

[11] MICROCHIP Technology Inc., "MPLAB IDE User`s Guide", 2006, (Página Web), <http://ww1.microchip.com/downloads/en/DeviceDoc/51519a.pdf>, consultada en Noviembre 2008.

[12] DOMITSU Kono, Manuel, "Programación en el lenguaje C", (Página Web), http://www.itson.mx/die/mdomitsu/bibliotecaDigital/Programacion_Lenguaje_C/, consultada en Noviembre 2008.

[13] MICROCHIP Technology Inc., "MPLAB C30 C Compiler User`s Guide", 2007, (Página Web), http://ww1.microchip.com/downloads/en/DeviceDoc/C30_Users_Guide_51284e.pdf, consultada en agosto 2008.

[14] D. Grahame Holmes, Thomas A. Lipo, "Pulse Width Modulation For Power Converters, Principles And Practice", IEEE Press Series on Power Engineering, 2003.

CAPÍTULO III

Desarrollo

En el presente capítulo se describe el proceso a seguir para la implementación digital de la técnica de modulación de vectores espaciales (SVM) utilizando un controlador digital de señal (DSC), en el cual se plantean de forma general los requerimientos del sistema, y de forma detallada la descripción con la cual debe cumplir el mismo. Además se explica la implementación digital describiendo los pasos a seguir para la elección del DSC, el lenguaje de programación, el software y el compilador, así como también se describe y explica el desarrollo del código del programa, finalizando con el esquemático del circuito a implementar.

3.1 Requerimientos

Se requiere implementar la técnica de Modulación de Vectores Espaciales (SVM) en un sistema digital, para aplicarla a un Inversor Trifásico de Tres Niveles con Diodo Anclado al Punto Neutro. La función es poder tener incidencia sobre la ganancia del inversor, para poder alimentarlo con un voltaje de entrada de CD fijo, y a su salida poder tener un voltaje en CA que pueda ser variado simplemente con variar el ancho del pulso de la señal de control PWM.

3.2 Descripción del sistema

El sistema cumple con las siguientes características:

- Cuenta con una señal de referencia, la cual determina el índice de modulación de la señal de control.
- La señal de referencia se procesa en un controlador digital de señales (DSC) el cual se encargará de hacer los cálculos necesarios para obtener los ciclos útiles de los 3 vectores más cercanos en el diagrama SVM (figura 2.10).
- Aplica la técnica de los 3 vectores más cercanos para sintetizar la tensión de salida del inversor.
- La señal de referencia se varía con un potenciómetro conectado al canal AN8 del DSC, para variar el índice de modulación y así, variar la ganancia del inversor.

La figura 3.1 muestra el diagrama a bloques del sistema. En éste se muestra el bloque del voltaje de entrada en CD, el bloque inversor, el bloque de modulación SVM y el bloque de salida.

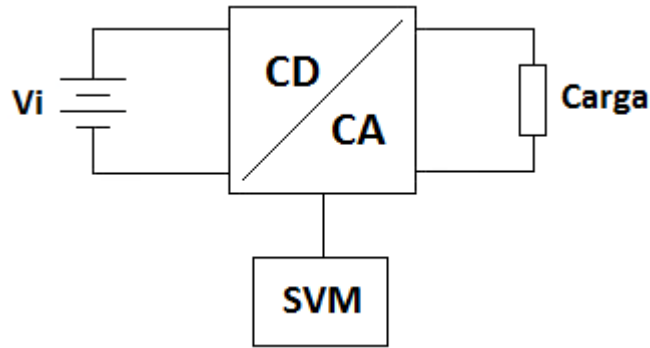


Figura 3.1. Diagrama a bloques del sistema.

3.3 Implementación

El sistema mostrado en la figura 3.1 puede dividirse en 3 partes fundamentales:

- Inversor Trifásico NPC de Tres Niveles.
- Fuente de alimentación *Push - Pull* para los dispositivos del inversor.
- Implementación digital de la técnica de Modulación de Vectores Espaciales.

Para el desarrollo de este trabajo, se dividió la implementación en los 3 puntos mencionados anteriormente para facilitar su comprensión.

3.4 Inversor trifásico NPC de tres niveles

Para el diseño del inversor se utilizó el siguiente método:

- Elección de la topología.
- Optimización del hardware a utilizar.
- Implementación del inversor.

3.4.1 Elección de la topología

En el apartado 2.5 se mostraron las topologías multinivel existentes, y se eligió la topología NPC (2.5.3), debido a que la técnica SVM trabaja con inversores trifásicos con 3 medios puentes en paralelo. La topología de capacitores flotantes (2.5.2)

también se implementa con 3 medios puentes conectados en paralelo, pero en este caso, se tiene el gran inconveniente con el número de capacitores existentes en el diseño, lo que dificulta el desarrollo ya que existen desbalances en los capacitores lo cual hace más complicado el inversor y propenso a sufrir desbalances en la tensión de salida.

Por otro lado, en la topología NPC, en vez de capacitores existen diodos fijadores de tensión, lo cual facilita el diseño y lo hace más económico, ya que es más sencillo trabajar con diodos fijadores que con capacitores.

Los capacitores de entrada pueden ser propensos a desbalances. Estos capacitores simulan 2 fuentes de tensión para obtener los niveles de voltaje $V_i/2$ y $-V_i/2$, para esto se necesita distribuir bien la entrega de energía de los capacitores de entrada al inversor, para que se carguen y descarguen con la misma frecuencia.

3.4.2 Diseño de la topología

La topología del inversor implementado se muestra en la figura 3.2.

La topología consta de:

- 12 Transistores ($S_{11} - S_{34}$).
- 6 Diodos fijadores de tensión.
- 2 Capacitores divisores de tensión.

Cada rama del inversor está compuesta por 4 transistores que se identifican como S_{x1} , S_{x2} y S_{x3} y S_{x4} , y 2 diodos fijadores de tensión identificados como D_{x5} y D_{x6} (donde x puede ser una de las tres ramas).

Para la estrategia de conmutación de la rama k (donde k puede tomar los valores {1, 2, 3}) los transistores S_{k1} y S_{k3} siempre tienen que ser complementarios. La misma restricción debe darse en los transistores S_{k2} y S_{k4} .

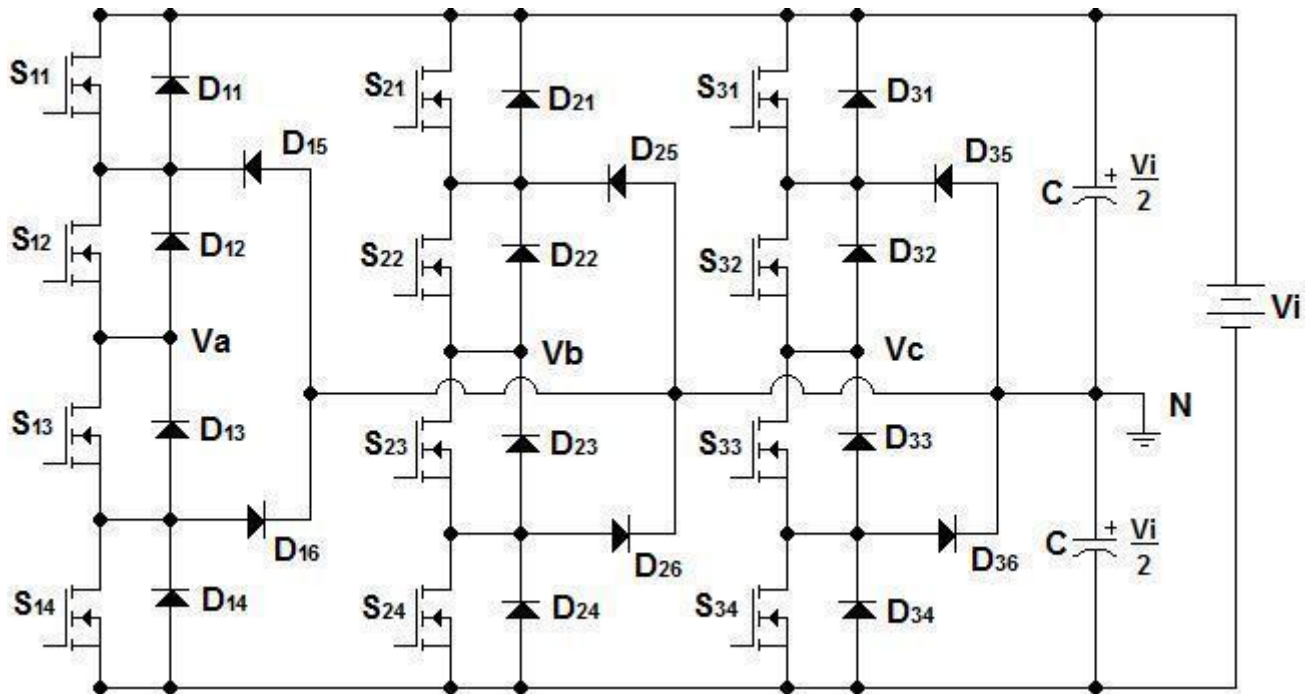


Figura 3.2. Inversor trifásico NPC de tres niveles.

Para obtener los tres niveles de tensión a la salida ($V_i/2$, 0 y $-V_i/2$) la estrategia de conmutación a seguir es la siguiente:

- Cuando los transistores S_{K1} y S_{K2} se encuentran encendidos, el voltaje de entrada es $V_i/2$.
- Cuando los transistores S_{K2} y S_{K3} se encuentran encendidos, el voltaje de entrada es cero.
- Cuando los transistores S_{K3} y S_{K4} se encuentran encendidos, el voltaje de entrada es $-V_i/2$.

Los diodos fijadores de tensión, conducen una corriente de marcha libre cuando los transistores S_{K2} y S_{K3} están encendidos, por lo tanto la corriente fluye por los mismos directamente al punto neutro N.

Los niveles de tensión de entrada $V_i/2$ y $-V_i/2$ se logran por los capacitores divisores de tensión C_1 y C_2 .

Los 3 voltajes de salida, desfasados 120 grados entre sí, V_{AN} , V_{BN} y V_{CN} , se obtienen en la fuente de S_{K2} y el dren de S_{K3} .

3.4.3 Implementación del inversor

Los componentes de potencia que se eligieron son los siguientes:

Transistores: en el caso de los transistores se eligieron los transistores MOSFET, por su alta velocidad de conmutación, se eligió el transistor STW25NM60N del fabricante ST Microelectronics, este dispositivo cuenta con una resistencia de encendido de 0.130Ω , un voltaje de dren a fuente, V_{DS} , de 600 V, y una corriente de conducción de 21 A. Lo que lo hace muy resistente soportando potencias de hasta 12.6 kW.

Diodos: en el caso de los diodos, se eligió el diodo STTH3012W del fabricante ST Microelectronics, este dispositivo cuenta con un voltaje de bloqueo V_{AK} de 1200 V, una capacidad de corriente de 30 A, y un tiempo de recuperación de 57 ns.

Capacitores divisores de voltaje: para los capacitores, se eligieron 2 capacitores del fabricante Rubycon, con una capacidad de 820 μF a 250 V. Capacidad suficiente para que el desbalance existente en los capacitores no se refleje en la salida del inversor.

En la figura 3.3 se muestra el inversor implementado.

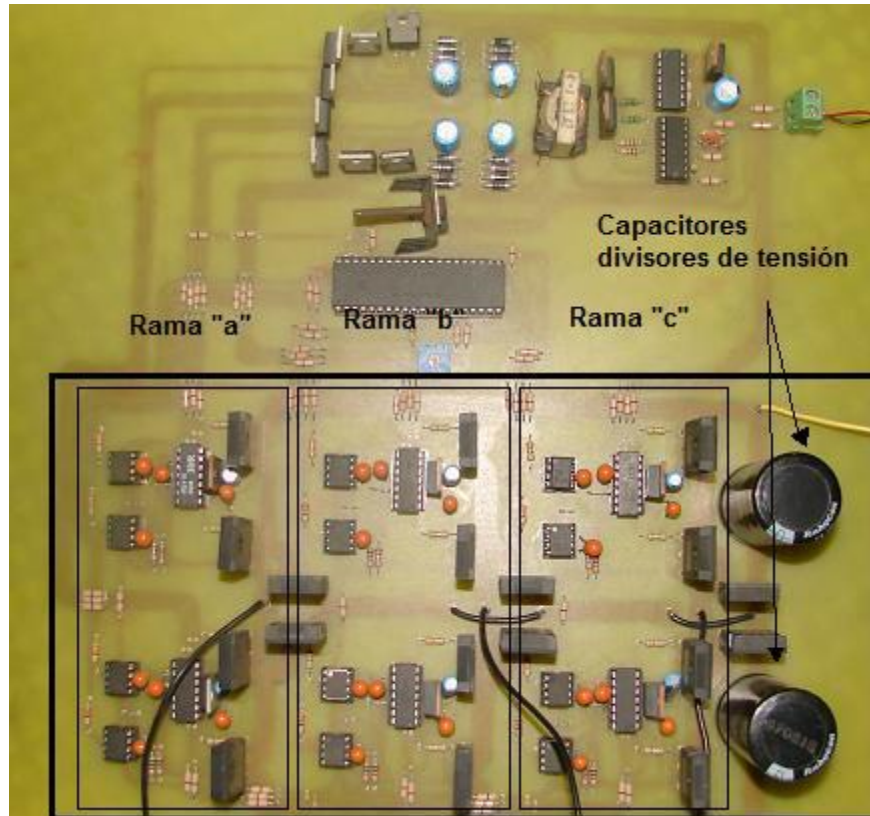


Figura 3.3. Inversor implementado.

3.5 Fuente de alimentación *Push - Pull* para los dispositivos del inversor

Para alimentar a los dispositivos que hacen posible la operación de cada rama del inversor (controlador digital de señales, optoacopladores, drivers) es necesario utilizar una fuente de voltaje que no tengan una referencia a tierra física, sino que cada dispositivo esté referenciado a un único nivel de voltaje. La figura 3.4 muestra la forma en la que los transistores están conectados en el inversor implementado.

Las flechas indican cada una de las referencias distintas que existen en los transistores (fuente del transistor), esto es, no se puede referenciar al mismo nivel cero la fuente del transistor S_{X1} con la referencia cero del transistor S_{X2} , ya que quedarían los transistores en cortocircuito, para esto se utiliza el driver IR2110 (Driver X_1) de International Rectifier, el cual se encarga de suministrar la potencia para excitar la compuerta de los transistores, en este caso dos transistores, haciendo

un disparo referenciado a un nivel de voltaje cero (S_{X2}) y el otro nivel de voltaje también es cero pero no la misma referencia que S_{X2} (transistor S_{X1}) esto es a lo que se llama tierra flotada.

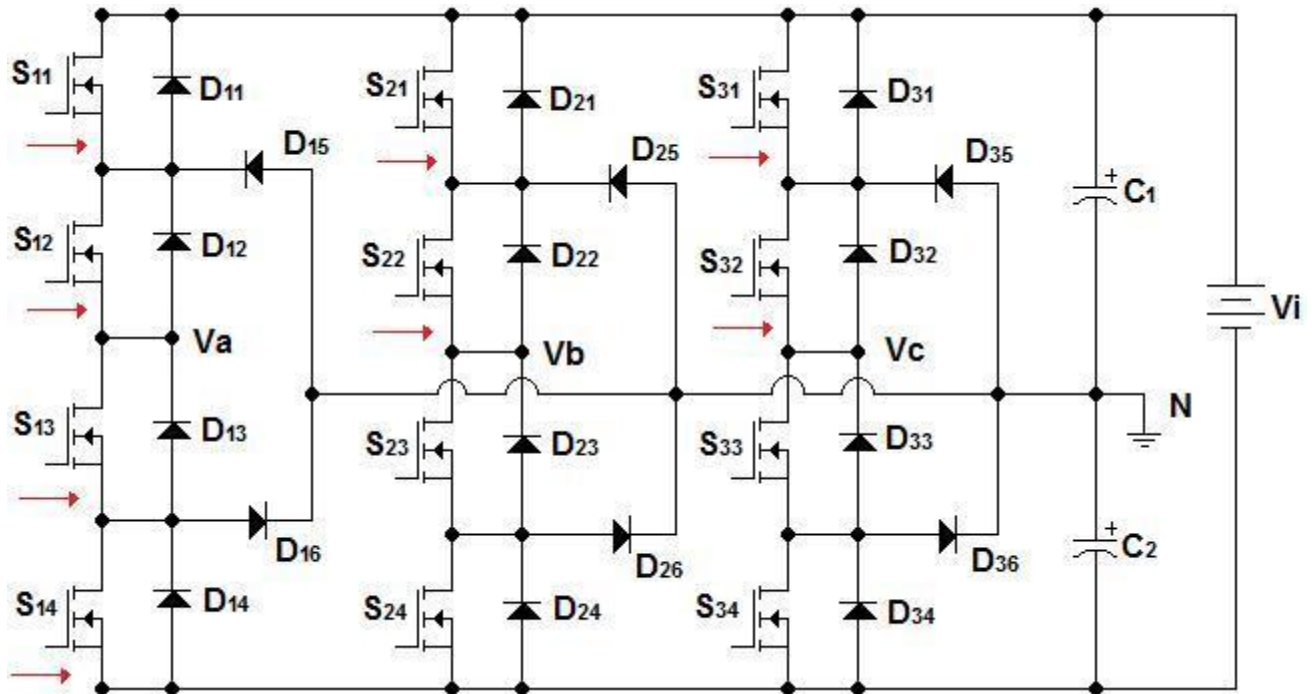


Figura 3.4. Referencias de los transistores en el inversor implementado.

Dado que en serie al transistor S_{X2} se tiene al transistor S_{X3} , se necesita otro driver que excite a los transistores S_{X3} y S_{X4} (Driver X_2), que lo hará de la misma manera que en S_{X1} y S_{X2} , pero la fuente de alimentación del Driver 1 tiene que ser diferente a la del Driver 2, ya que si es la misma se estarían poniendo en corto a esas 2 medias partes de la rama [Transistor S_{X1} y S_{X2} (parte alta) con los transistores S_{X3} y S_{X4} (parte baja)]. En el caso de S_{12} , S_{22} y S_{32} no se puede tener la misma referencia, ya que las tres fuentes de S_{X2} son las tres fases (a, b, c) del inversor trifásico respectivamente. Por lo que para cada parte alta de la rama (Driver X_1) se necesita una fuente de alimentación individual, esto es, tres fuentes de alimentación para cada una de las partes altas del inversor, y una fuente de alimentación para toda la rama baja del inversor, ya que como se puede ver, los transistores S_{X4} están en las tres ramas referenciados al mismo punto, por lo que, en ese caso, si se puede usar una

fuente de alimentación para las tres ramas bajas. En la figura 3.5 se muestran los Drivers S_{XX} del inversor.

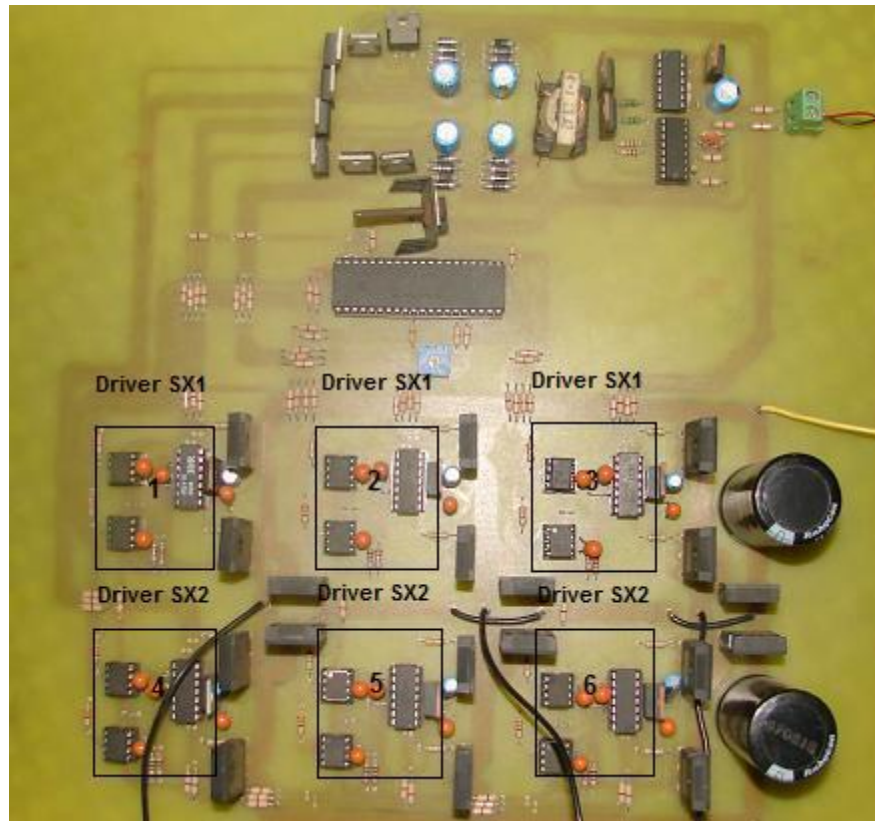


Figura 3.5. Ubicación de los Drivers en el inversor implementado.

Los drivers 1, 2 y 3 representan a las 3 ramas altas del inversor, y los Drivers 4, 5 y 6 representan a la rama baja del inversor, estas últimas están referenciadas al mismo nivel de voltaje. En la figura 3.6 se muestra las ramas bajas y altas del inversor.

Los optoacopladores se utilizan para aislar galvánicamente la parte digital de la parte de potencia, como es el controlador digital de señales, ya que si existe un cortocircuito en los puentes del inversor, el daño sólo llegará hasta los optoacopladores y no más atrás.

El diagrama de la fuente de alimentación *Push - Pull* propuesta se muestra en la figura 3.7.

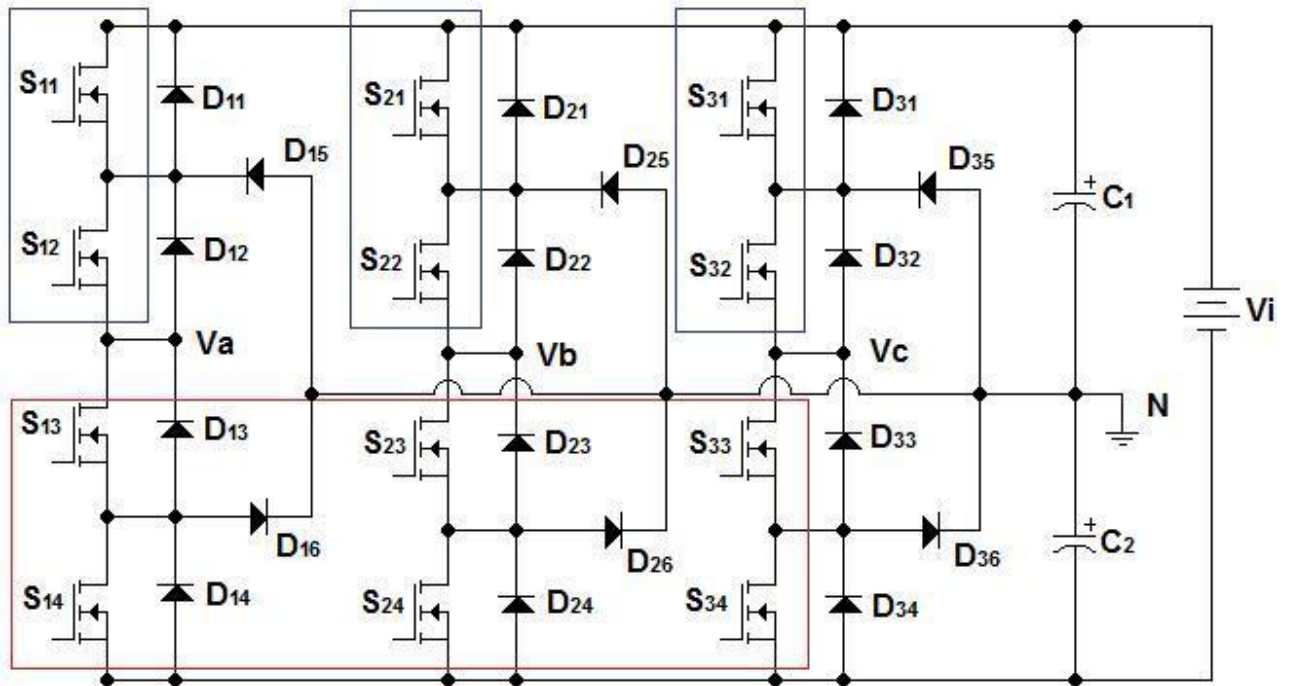


Figura 3.6. Tres ramas altas del inversor (rectángulos pequeños) y tres ramas bajas (rectángulo grande).

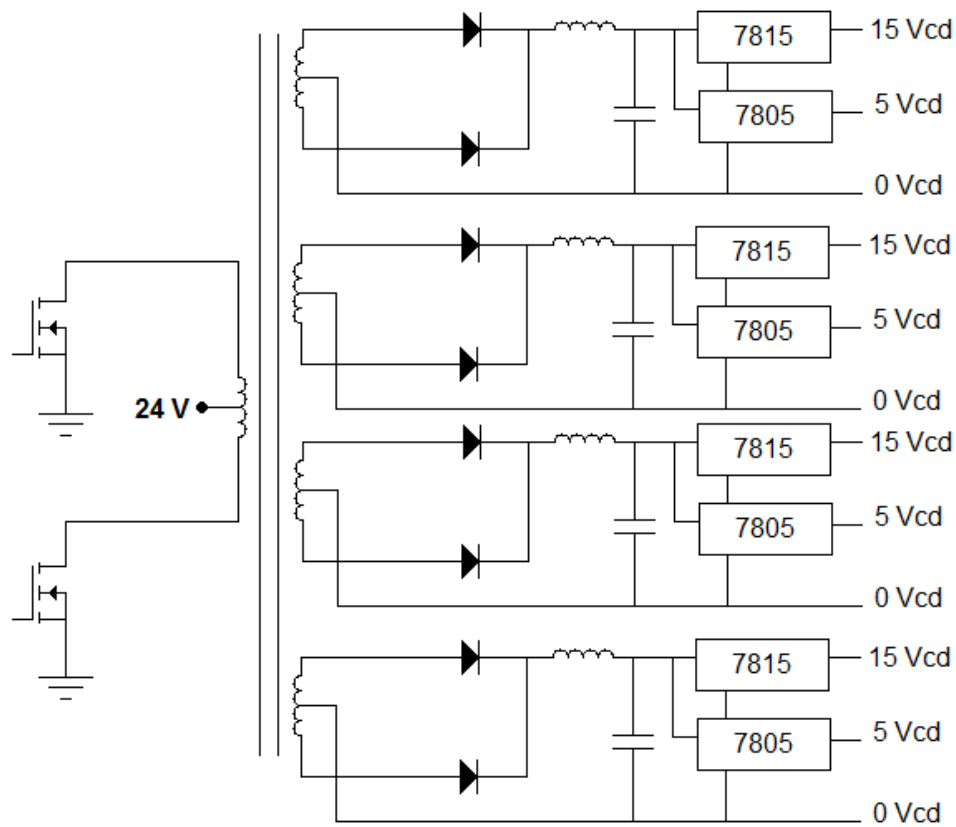


Figura 3.7. Diagrama de la fuente de alimentación Push - Pull propuesta.

Los niveles de voltaje a la salida se logran con los reguladores de voltaje 7805 y 7815, los cuales regulan el voltaje a 5 y 15 V respectivamente. Para lograr esto el voltaje de entrada de cada uno de ellos debe ser al menos 3 V más alto que el voltaje que se quiere lograr (al menos 8 V para regular a 5 V y 18 V para regular a 15 V).

En la figura 3.8 se muestra la fuente de alimentación *Push - Pull* implementada.

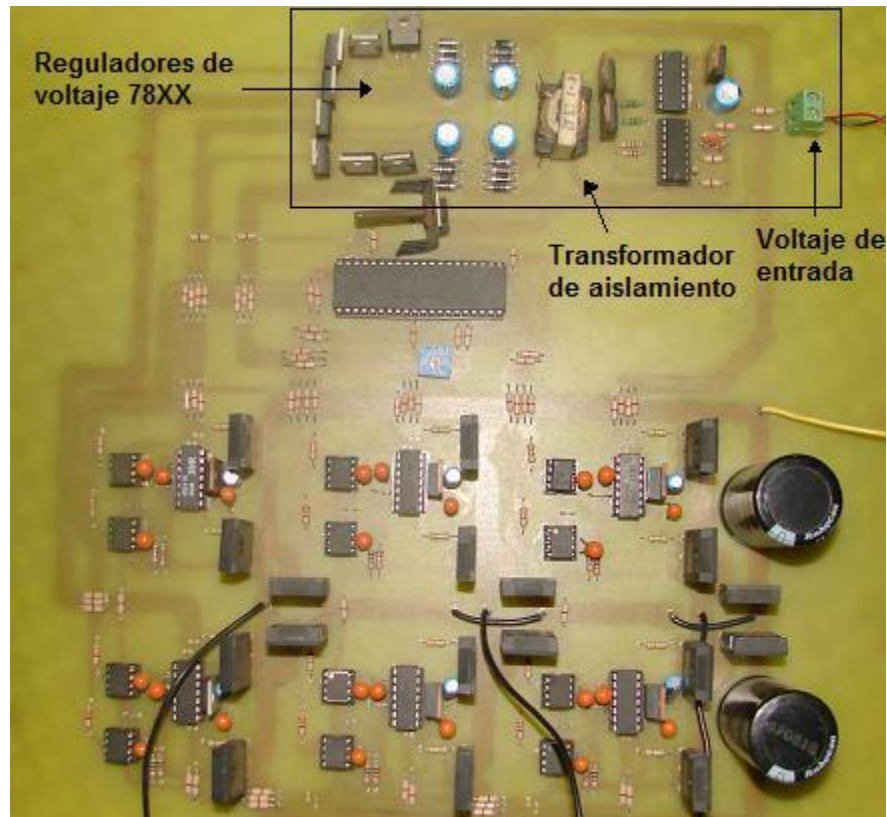


Figura 3.8. Fuente de alimentación Push – Pull implementada.

3.6 Implementación digital

En la implementación digital para la técnica de Modulación de Vectores Espaciales se utilizó la siguiente metodología:

- Elección del dispositivo a utilizar.
- Elección del lenguaje de programación a utilizar.
- Elección del software de desarrollo y compilador del lenguaje.

- Desarrollo del programa.
- Diagrama esquemático de la implementación digital.

A continuación se explica a detalle cada uno de los puntos anteriores.

3.6.1 Elección del dispositivo a utilizar

El dispositivo utilizado en el desarrollo de este proyecto es el controlador digital de señales (DSC) dsPIC30F4011 de Microchip®, el cual se muestra en la figura 2.18.

Este dispositivo cuenta con las características básicas para el desarrollo del proyecto ya que cuenta con una velocidad de procesamiento de 120 MHz y 30 MIPS y puede efectuar operaciones en 1 ciclo máquina (33.33 ns). Además cuenta con un convertidor analógico digital (ADC) el cual puede tomar 1 millón de muestras de una señal por cada segundo, y 5 temporizadores, los cuales pueden funcionar en modo de contador o temporizador. Esta característica es importante ya que lo hace ideal en el manejo del control de motores y convertidores de potencia.

3.6.2 Elección del lenguaje de programación

El lenguaje de programación utilizado para la técnica SVM es el lenguaje C. Con este lenguaje es posible realizar operaciones matemáticas de una forma sencilla, así como poder tener una visión más amplia de lo que es el lenguaje estructurado y poder tener un mejor control y comprensión del algoritmo implementado.

Además se puede optimizar el trabajo utilizando las librerías de dicho lenguaje para hacer uso de los registros internos del DSC.

3.6.3 Elección del software de desarrollo y compilador del lenguaje

El software de desarrollo elegido es el MPLAB (*Integrated Development Environment*) (IDE) v8.0 de *Microchip Technology Inc.* que se distribuye de manera gratuita. Este ambiente de desarrollo está optimizado para el trabajo con todos los dispositivos del fabricante *Microchip Technology Inc.*

El compilador es el *MPLAB C30 Compiler v3.02*, este es un compilador de lenguaje C para dispositivos de las familias PIC24, dsPIC30F y dsPIC33F, este compilador también es distribuido por *Microchip Technology Inc.*

3.6.4 Desarrollo del programa

Para explicar el desarrollo del programa, se enlistan las rutinas que ejecuta el mismo y posteriormente se explican en el orden en que son ejecutadas, los puntos a seguir son:

- Declaración de funciones globales.
- Declaración de constantes y variables.
- Programa principal.
 - Configuración de puertos, ADC, Temporizadores y variables.
 - Rutina SVM.
- Esperar interrupción y cargar los ciclos útiles.

En la figura 3.9 se presenta el diagrama de flujo del programa principal SVM, y en las figuras 3.10 3.11 3.12 y 3.13 se presentan los diagramas de flujo de las rutinas de interrupción de cada Temporizador.

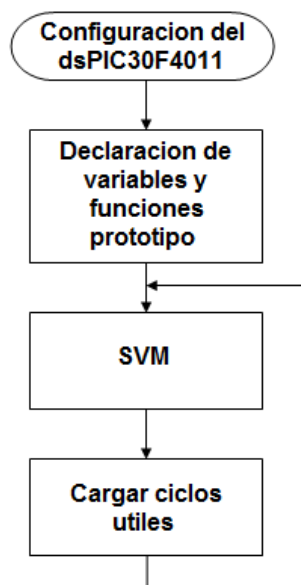


Figura 3.9. Diagrama de flujo del programa principal.

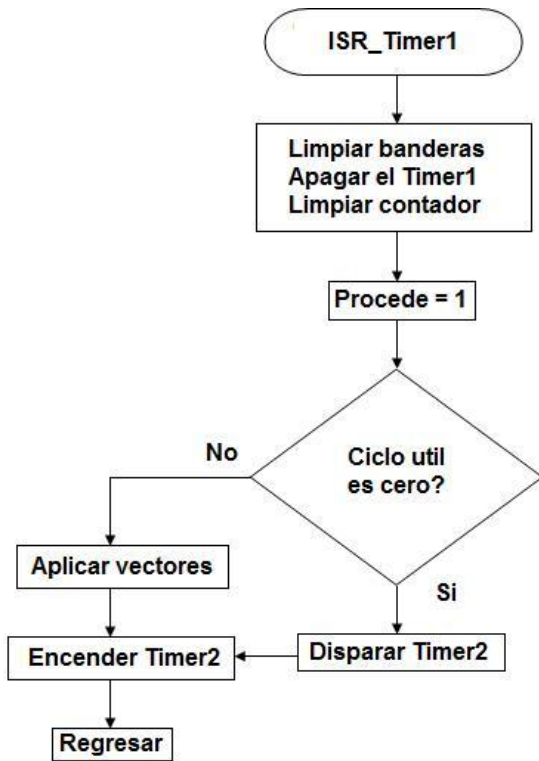


Figura 3.10. Diagrama de flujo de la rutina de interrupción del TIMER1.

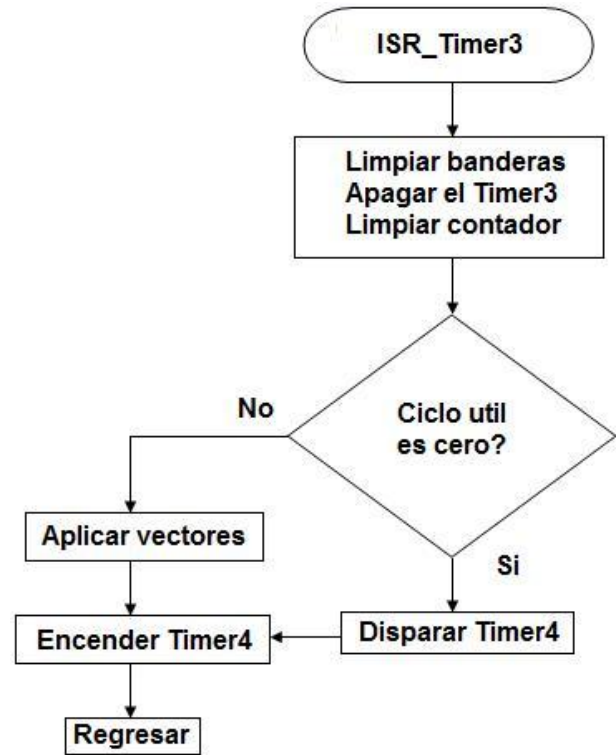


Figura 3.12. Diagrama de flujo de la rutina de interrupción del TIMER3.

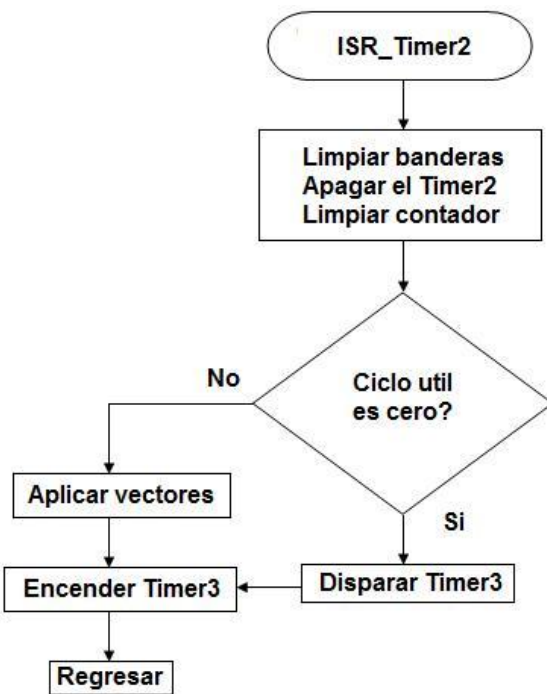


Figura 3.11. Diagrama de flujo de la rutina de interrupción del TIMER2.

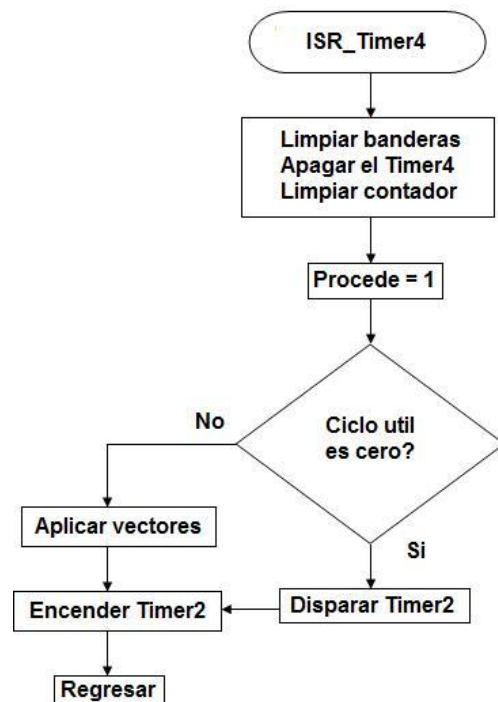


Figura 3.13. Diagrama de flujo de la rutina de interrupción del TIMER4

3.6.4.1 Declaración de funciones globales

Las funciones prototipo son funciones que se declaran antes del programa principal, e indican el nombre de las subrutinas que se utilizarán. En el programa desarrollado se cuenta con 15 funciones prototipo que serán explicadas detalladamente, las funciones prototipo son:

```
void
configuracion_inicial(void);
void puertos(void);
void ADC(void);
void Timers(void);
void variables_bits(void);
void SVM(void);
void magnitud(void);
unsigned int ADCRead(void);
void indice_modulacion(void);
void posicion(void);
void calcula_sector(void);
void calcula_triangulo(void);
void duty_cycle(void);
void periodos(void);
void calcula_vectores(void);
```

Figura 3.14. Código de la declaración de funciones globales.

La acción que realiza cada función se describe a continuación:

- Función *configuración_inicial*, inicializa todos los módulos constantes y variables.
- Función *puertos*, configura los puertos E y F como salida.
- Función *ADC*, configura el módulo ADC.
- Función *Timers*, Configura los temporizadores.
- Función *variables_bits*, inicializa los valores de las constantes y variables.
- Función *SVM*, realiza todos los cálculos de la técnica SVM.
- Función *magnitud*, calcula la magnitud del vector de referencia.
- Función *ADCRead*, lee el valor de la señal de referencia y lo convierte a digital.

- Función *indice_modulacion*, calcula el índice de modulación del vector de referencia.
- Función *posicion*, calcula la posición en X y Y en la que se encuentra el vector de referencia.
- Función *calcula_sector*, calcula el sector en el que se encuentra el vector de referencia.
- Función *calcula_triangulo*, calcula el triángulo en el que se encuentra el vector de referencia.
- Función *duty_cicle*, calcula los ciclos útiles de cada vector.
- Función *periodos*, carga los ciclos útiles a los temporizadores.
- Función *calcula_vectores*, calcula los vectores que se aplicarán dependiendo del sector en que se encuentre el vector de referencia.

Se llaman funciones globales por que las funciones pueden llamarse las veces que sean necesarias, sólo con escribir su nombre.

3.6.4.2 Declaración de constantes y variables

En la figura 3.15 se muestran las constantes y variables utilizadas en el programa.

```

unsigned short wt, pasa[6], triangulo, triangulox, sector, procede, normalizar;
unsigned short periodo[3], vector[3];
float ValorADC, xpos, ypos, m, d1, d2, d3, res0, res1, res2;
const float sin[90] = {0, 0,0.0698,0.1392,0.2079,0.2756,0.342 ... 0};
const float cos[90] = {1,0.9976,0.9903,0.9781,0.9613,0.9397 ... 1};
const unsigned int vectores[27] = {0x0CCC,0x0CC6,0x0CC3 ... 0x0333};

```

Figura 3.15. Código de la declaración de constantes y variables.

La acción que realiza cada variable se explica a continuación:

- *wt*, indica el ángulo en grados en el que está posicionado el vector de referencia.
- *pasa*, es un vector de seis elementos que indica si el vector a cruzado una de las líneas que definen los límites entre triángulos pequeños *j* y los triángulos grandes *i* en el diagrama SVM.

- *triangulo*, indica en cuál de los 24 triángulos *j* se encuentra el vector de referencia.
 - *triangulox*, variable auxiliar de *triangulo*.
 - *sector*, indica en cuál de los seis sectores se encuentra el vector de referencia.
 - *procede*, indica si se puede proceder o no a hacer el cálculo total de la rutina SVM.
 - *normalizar*, variable auxiliar que normaliza el valor del sector para hacer los cálculos de los ciclos útiles sólo para el sector uno, independientemente de que sector sea.
 - *periodo*, variable arreglo de tres elementos que almacena el valor hasta el cual el temporizador va a contar para los ciclos de trabajo *d1*, *d2* y *d3*.
 - *vector*, arreglo variable de tres elementos que representan los tres vectores que se aplicarán dependiendo de la posición del vector de referencia.
 - *ValorADC*, almacena el resultado de la conversión del ADC para el vector de referencia.
 - *xpos*, indica en qué posición en el eje de las X se encuentra el vector de referencia.
 - *ypos*, indica en qué posición en el eje de las Y se encuentra el vector de referencia.
 - *m*, almacena el índice de modulación de la señal de referencia.
 - *d1*, *d2*, *d3*, almacenan el resultado de los tres ciclos útiles para los 3 vectores de referencia al final de los cálculos de la rutina SVM.
 - *res0*, *res1*, *res2*, almacenan el resultado del cálculo de la posición del vector de referencia a partir de la ecuación de la recta $y = mx + c$.
 - *sin*, arreglo constante de 90 elementos que contiene los valores de la función seno de 0 a 360 grados con intervalos de 4 grados.
 - *cos*, arreglo constante de 90 elementos que contiene los valores de la función coseno de 0 a 360 grados con intervalos de 4 grados.
 - *vectores*, arreglo constante de 27 elementos que contiene el valor de los 27 vectores que se muestran en el diagrama SVM.
-

Una vez explicadas las funciones globales, las variables y constantes, se explicará el programa principal.

3.6.4.3 Programa principal

En todo programa escrito en C tiene que existir una función principal *main*, en la figura 3.16 se muestra el programa principal desarrollado.

```
int main(void)
{
    __builtin_write_OSCCONL(4);
    OSCTUNbits.TUN = 7;
    configuracion_inicial();
    while(1)
    {
        if(procede == 1)
        {
            SVM();
        }
    }
    return 0;
}
```

Figura 3.16. Código del programa principal.

La función principal del programa, según el código de la figura 3.16, en primer lugar configura el oscilador interno del DSC para trabajar al máximo (120 MHz a 30 MIPS) modificando el valor del registro OSCTUN. Después se realiza la configuración inicial, en donde se configura el módulo ADC, los temporizadores y la inicialización de las variables, posteriormente se entra en un ciclo infinito, el cual condiciona si es momento de entrar a hacer los cálculos de la rutina SVM, y ahí se mantiene infinitamente esperando interrupciones y volviendo a ejecutar la rutina SVM, a continuación se explicará a detalle todas las subrutinas internas del programa principal.

3.6.4.4 Configuración de puertos, ADC, temporizadores y variables

En la quinta línea del programa principal de la figura 3.16, se encuentra la función prototipo *configuración_inicial()*; la cual, configura los puertos, el ADC, Temporizadores y las variables, en la figura 3.17. Se muestra el código de esta función global.

```
void configuracion_inicial(void)
{
    puertos();
    ADC();
    Timers();
    variables_bits();
    return;
}
```

Figura 3.17. Código de la función prototipo configuración_inicial.

A continuación se explica cada una de las subrutinas que conlleva esta función.

3.6.4.5 Configuración de puertos

Para la configuración de puertos, sólo se tienen 1 entrada y 2 salidas, la entrada analógica por el canal AN8 y los puertos E y F para la salida de los 12 disparos de los transistores. Como el canal analógico siempre es entrada, sólo se configura como salida los puertos E y F por medio de los registros TRISE y TRISF, el código para la configuración de los puertos se presenta en la figura 3.18.

```
void puertos(void)
{
    TRISE = 0;
    TRISF = 0;
    return;
}
```

Figura 3.18. Código de la configuración de los puertos.

3.6.4.6 Configuración del ADC

El código para la configuración del ADC se muestra en la figura 3.19.

```
void ADC(void)
{
    ADCON1 = 0x00E4;
    ADCON2 = 0x0104;
    ADCON3 = 0x0287;
    ADCHS = 0x0008;
    ADPCFG = 0xFEFF;
    ADCSSL = 0x0000;
    return;
}
```

Figura 3.19. Código de la configuración del ADC.

En el código de la figura 3.19, el ADC se configura para trabajar en modo de conversión automática por el canal AN8, a una velocidad de muestro de 600 kps, mostrando el resultado en un formato entero de 10 bits. El resultado de la conversión se toma en el modo de sondeo, es decir, cuando se requiere tener el resultado de la conversión solo se llama a la función que lo devuelve, y nunca existe una interrupción por fin de conversión del ADC. Se toman como voltajes de referencia las alimentaciones del DSC, las cuales son $V_{REF+} = 5\text{ V}$ y $V_{REF-} = 0\text{ V}$.

3.6.4.7 Configuración de los temporizadores

En la figura 3.20 se muestra el código de la configuración de los Temporizadores.

```
void Timers(void){
    T1CON = 0x0004;
    T2CON = 0x0004;
    T3CON = 0x0004;
    T4CON = 0x0004;
    PR1 = 5556;
    IEC1 = 0x0020;
    IEC0 = 0x00C8;
    return;
}
```

Figura 3.20. Código de la configuración de los Temporizadores.

En el código de la figura 3.20 se observa la configuración de 4 temporizadores, los cuales se configuran para trabajar en modo síncrono con la frecuencia del reloj interno (120 MHz) y trabajando sin prescalador, esto es, cada ciclo máquina equivale a un incremento en el contador TMRx. Adicionalmente se precargó el registro de periodo PR1 con el número 5556, es decir, que cuando el registro TMR1 cuente hasta el valor de PR1, existirá una interrupción, la cual llevará a la rutina de servicio para la aplicación de los vectores correspondientes. El valor 5556 trabajando a una frecuencia de 120 MHz (30 MIPS) quiere decir que si cada conteo se ejecuta en 33.33 ns, entonces si se cuenta hasta 5556 tardará 185.18 μs , o bien 5.4 kHz. Esto quiere decir que sólo se dispone de 185.18 μs para ejecutar todos los cálculos de la rutina SVM, lo cual a la salida será reflejado en una frecuencia de conmutación para el PWM de 5.4 kHz.

3.6.4.8 Configuración de las variables

El código de la configuración de las variables y banderas se muestra en la figura 3.21.

```
void variables_bits(void)
{
    IFS0 = 0;
    wt = 0;
    ADCON1bits.ADON = 1;
    T1CONbits.TON = 1;
    procede = 1;
    return;
}
```

Figura 3.21. Código de la configuración de las variables.

Para la configuración de las variables y bits, inicialmente se limpian las banderas de interrupción de los Temporizadores para evitar disparos falsos, se inicializa el ángulo con el que se trabajará en cero grados, se enciende el convertidor analógico - digital, se enciende el Temporizador 1, y la variable procede se pone en 1 para entrar directamente a ejecutar los cálculos de la rutina SVM.

Una vez explicado el código de las variables y configurado los módulos internos del DSC, se puede explicar el punto principal del programa: la rutina SVM.

3.6.4.9 SVM

En este apartado se explicará a detalle la rutina más importante del programa, y la que es de sumo interés en el estudio del proyecto, la rutina SVM, la cual mediante una serie de algoritmos y cálculos podrá finalmente calcular los ciclos útiles para aplicar los vectores de la rutina SVM. En la figura 3.22 se muestra el código de la rutina SVM.

El código de la rutina SVM que se muestra en la figura 3.22 se explica a modo de bloques o subrutinas, existen 7 subrutinas que serán explicadas una a una en los siguientes apartados.

```
void SVM(void)
{
    magnitud();
    indice_modulacion();
    posicion();
    calcula_sector();
    calcula_triangulo();
    duty_cicle();
    calcula_vectores();
    return;
}
```

Figura 3.22. Código de la rutina SVM.

3.6.4.10 Magnitud

En la figura 3.23 se muestra el código de la función *magnitud*.

Dentro de la función *magnitud*, también se puede encontrar otra subrutina interna llamada *ADCRead*, la cual se muestra en la figura 3.24. Y que también será explicada a detalle en el apartado 3.6.4.11.

```
void magnitud(void)
{
    ValorADC = ADCRead();
    if(ValorADC > 52)
    {
        ValorADC = 52;
    }
    if(ValorADC < 12)
    {
        ValorADC = 12;
    }
    return;
}
```

Figura 3.23. Código de la rutina *magnitud*.

Antes de seguir explicando la rutina *magnitud* se explicará la rutina *ADCRead*.

3.6.4.11 ADCRead

En la figura 3.24 se muestra el código de la rutina *ADCRead*.

Cuando se llama a la rutina *ADCRead*, se espera a que la bandera de interrupción de fin de conversión del ADC se ponga en 1, cuando se haya puesto en 1 quiere decir

que la conversión a finalizado, se borra la bandera de interrupción y se toma el valor de la conversión del registro ADCBUF0 y se recorre 4 bits a la derecha para evitar interferencias de ruido generados por la fuente de alimentación (en vez de trabajar con 10 bits de resolución se trabajará con los 6 más significativos).

```
unsigned int ADCRead(void)
{
    while(IFS0bits.ADIF == 0)
    {
        asm("nop");
    }
    IFS0bits.ADIF = 0;
    return ADCBUF0>>4;
}
```

Figura 3.24. Código de la rutina ADCRead.

Volviendo al apartado de magnitud, una vez obtenido el valor del ADC en 6 bits, se condicionará su valor máximo y mínimo, esto es, para tener sólo un índice de modulación máximo del 80% y mínimo del 20% (para evitar sobremodulación o apagar totalmente los transistores), se condiciona si el valor máximo del ADC es mayor a 52 (80%) que tome el valor de 52, y si el valor es menor a 12 (20%) que tome el valor de 12; así se garantiza que se trabaje con un índice de modulación mínimo del 20% y máximo del 80%.

3.6.4.12 Índice de modulación

El código de la rutina *índice de modulación* se muestra en la figura 3.25.

```
void indice_modulacion(void)
{
    m = (ValorADC / 64);
    return;
}
```

Figura 3.25. Código de la rutina *índice de modulación*.

Esta rutina es muy simple, sólo toma el valor que se tiene almacenado del ADC, y lo divide entre el valor total que se puede tener, ya que el índice de modulación es el cociente del valor que se tiene entre el valor total, por lo tanto sólo se pueden obtener valores porcentuales (de 0 a 1). En este caso el valor total que se puede tener con 6

bits es 64, por lo tanto el valor que se tenga en el ADC se divide entre 64 y así se obtiene el índice de modulación m .

En la ecuación 3.1 se puede observar la fórmula para calcular el índice de modulación:

$$m = \frac{v}{V} \quad (3.1)$$

donde m es el índice de modulación (porcentual).

v es el valor de lectura del ADC que se tiene.

V es el valor máximo que se puede tener con 6 bits (64 decimal).

Este valor se va a calcular cada periodo de conmutación, es decir, cada vez que se entre a la rutina SVM.

3.6.4.13 Posición

El código de la rutina *posición* se muestra en la figura 3.26.

```
void posicion(void)
{
    xpos = (ValorADC) * (cos[wt>>2]);
    ypos = (ValorADC) * (sin[wt>>2]);

    res0 = ypos + 1.732*xpos;
    if(res0 > 64){pasa[0] = 1;}
    else{pasa[0] = 0;}

    res1 = ypos - 1.732*xpos;
    if(res1 > 64){pasa[1] = 1;}
    else{pasa[1] = 0;}

    res2 = ypos;
    if(res2 < -32){pasa[2] = 1;}
    else{pasa[2] = 0;}

    if(res2 > 32){pasa[3] = 1;}
    else{pasa[3] = 0;}

    if(res0 < -64){pasa[4] = 1;}
    else{pasa[4] = 0;}

    if(res1 < -64){pasa[5] = 1;}
    else{pasa[5] = 0;}

    return;
}
```

Figura 3.26. Código de la rutina *posición*.

Para la comprensión de este código, se recurrirá a una explicación gráfica, ya que se utilizó la ecuación de la recta para determinar exactamente si cruzó o no una línea del diagrama SVM el vector de referencia. En la figura 3.27 se muestra el diagrama SVM donde se muestran los 2 triángulos que forman los vértices de los vectores medianos y los vectores grandes.

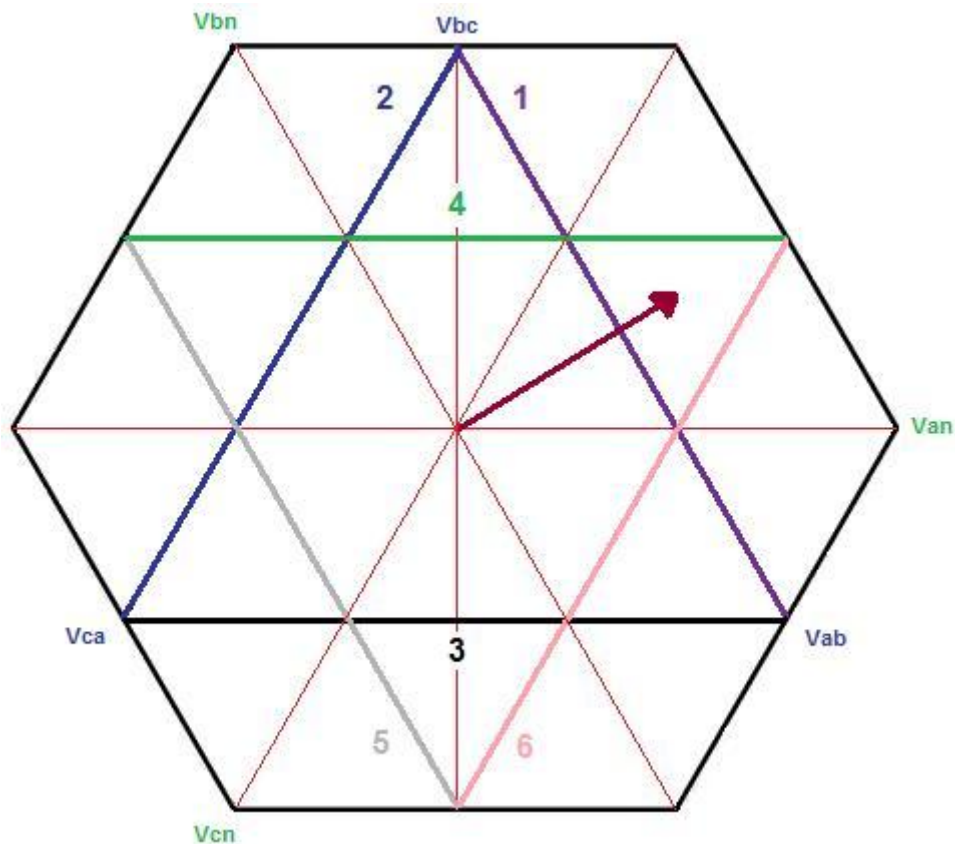


Figura 3.27. Diagrama SVM para ubicar el vector de referencia.

En el diagrama de la figura 3.27, se puede observar que se dibujan 2 triángulos equiláteros, uno en posición derecha (línea 1, 2 y 3) y otro triángulo en posición invertida (línea 4, 5 y 6). Cada línea tiene una pendiente, y esta se puede calcular, formando en total seis ecuaciones de la recta:

La ecuación de la pendiente cumple con la fórmula:

$$m = \frac{y_2 - y_1}{x_2 - x_1} \quad (3.2)$$

Por lo tanto, calculando la pendiente de la recta número 2, la ecuación 3.2 quedaría:

$$m = \frac{1-0}{\frac{1}{\sqrt{3}}-0}; \frac{1}{\frac{1}{\sqrt{3}}}; 1.732 \quad (3.3)$$

El valor $\frac{1}{\sqrt{3}}$ es el valor normalizado del vector pequeño V_{SI} , y 1 es el valor del vector mediano V_{MI} , con estos datos se pueden obtener los valores y_2 , y_1 , x_2 y x_1 para obtener el resultado de la ecuación 3.3.

La ecuación de la pendiente sería positiva, ya que la pendiente está hacia el lado derecho, cuando la pendiente está hacia el lado izquierdo el resultado es negativo, y cuando sólo tiene componentes en el eje de las Y no existe pendiente, con este razonamiento podemos deducir las otras 5 ecuaciones, y colocando los valores en la ecuación de la recta:

$$c = y - mx \quad (3.4)$$

Además, como puede verse, todas las rectas que tienen pendiente, ésta pendiente es del mismo valor, salvo que sea positiva o negativa, en la figura 3.28 se muestran las seis ecuaciones de las seis rectas.

1:	$64 = y + 1.73x$
2:	$64 = y - 1.73x$
3:	$y = -32$
4:	$y = 32$
5:	$-64 = y + 1.73x$
6:	$-64 = y - 1.73x$

Figura 3.28. Ecuaciones de las seis rectas mostradas en la figura 3.27.

Por lo tanto, para saber exactamente dentro de qué triángulo pequeño j se encuentra el vector de referencia, sólo hace falta hacer comparaciones y comprobar si el vector a cruzado una línea, y si es así comprobar si a cruzado otra línea, y así haciendo comparaciones para asegurar la ubicación del vector. Por ejemplo, para el caso de la

figura 3.27, el vector de referencia se encuentra en el triángulo pequeño $j = 2$, (figura 3.29) para que el código de programación pueda entenderlo, se hace la comparación y se observa que el vector a cruzado la línea 1, pero aun no ha cruzado la línea 4 y 6, esto garantiza que el vector se encuentre en el triángulo $j = 2$.

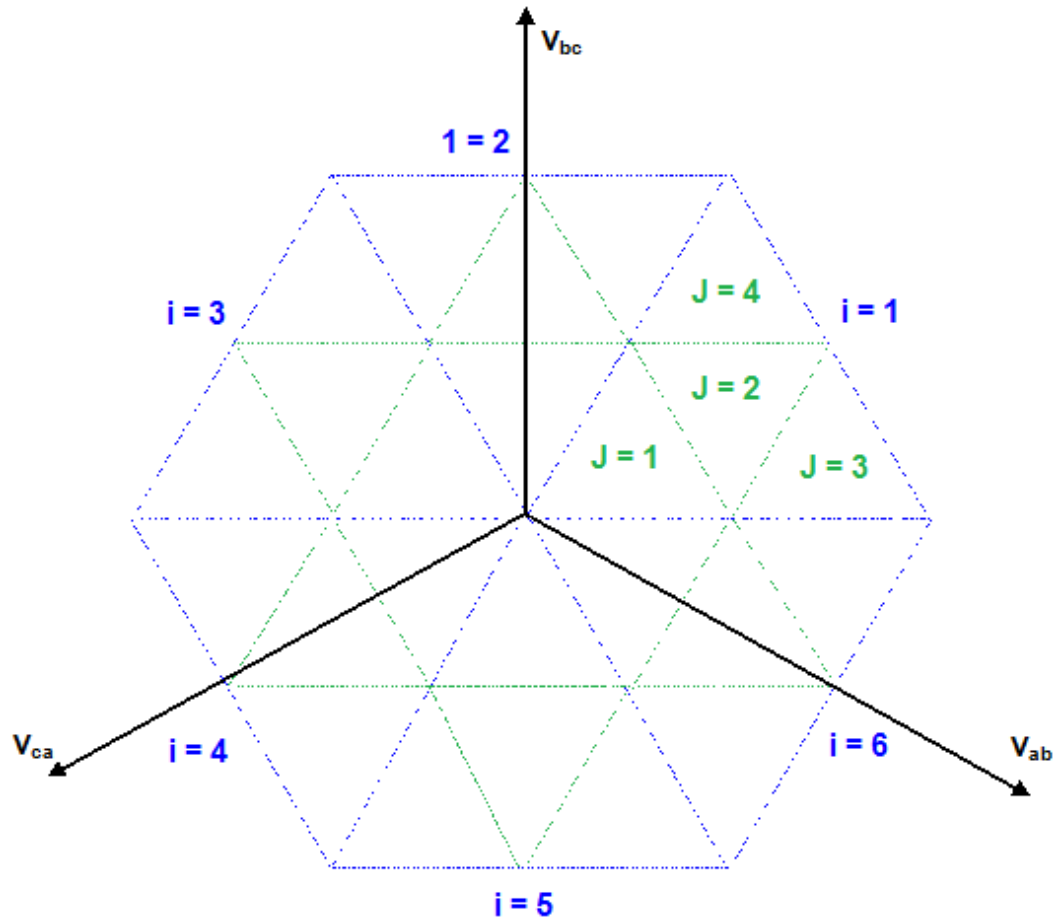


Figura 3.29. Notación relativa a los triángulos definidos por los vértices de los vectores de estado: triángulos grandes (azul) y triángulos pequeños (verde).

Para el caso del arreglo de variables “*pasa*”, la asociación de cada variable con cada recta se presenta en la figura 3.30.

<code>pasa[0]</code>	=	Recta 1
<code>pasa[1]</code>	=	Recta 2
<code>pasa[2]</code>	=	Recta 3
<code>pasa[3]</code>	=	Recta 4
<code>pasa[4]</code>	=	Recta 5
<code>pasa[5]</code>	=	Recta 6

Figura 3.30. Asociación de la variable de 6 elementos “*pasa*” con las diferentes rectas.

Por lo tanto, cada vez que en el programa se detecte que el vector de referencia a cruzado una línea (a superado su valor constante “C” de la ecuación 3.4), el vector “pasa” se pondrá en 1, y cuando no la haya cruzado aún se pondrá en 0, esto con el fin de hacer más sencillas las comparaciones a la hora que se tenga que ubicar exactamente el vector de referencia.

La asignación de la ubicación del vector en los triángulos grandes *i* se plantea en la figura 3.31.

3.6.4.14 Calcula sector

En la figura 3.31 se muestra el código de la rutina *calcula_sector*.

```
void calcula_sector(void)
{
    if(wt >= 0 && wt <= 60)
    {
        sector = 0;
    }
    else if(wt > 60 && wt <= 120)
    {
        sector = 1;
    }
    else if(wt > 120 && wt <= 180)
    {
        sector = 2;
    }
    else if(wt > 180 && wt <= 240)
    {
        sector = 3;
    }
    else if(wt > 240 && wt <= 300)
    {
        sector = 4;
    }
    else if(wt > 300 && wt < 360)
    {
        sector = 5;
    }
    return;
}
```

Figura 3.31. Código de la rutina *calcula_sector*.

Calcular el sector *i* en el que se encuentra el vector de referencia es mucho más sencillo que calcular la posición en triángulos pequeños *j* en el que se encuentra el mismo, como en total se tienen seis sectores en 360 grados (60 grados por sector),

solo hace falta comparar si el vector de referencia a cruzado de 60, 120, 180, 240, 300 y 360 grados, al primer sector no se le denomina sector uno sino sector cero, para posteriormente poder trabajar con la variable *normalizar*.

3.6.4.15 Calcula triángulo

El código de la rutina *calcula_triángulo* se muestra en la figura 3.32.

El código de la rutina *calcula_triángulo* es extenso, pero en realidad es muy sencillo, ya que sólo se hacen comparaciones para saber si el vector a cruzado o no una de las líneas descritas en la figura 3.27, entonces se asigna la posición a dos variables, una llamada *triángulo*, la cual almacena la posición del triángulo en dos números, por ejemplo, si el vector se encuentra en el sector cero y en el triángulo pequeño $j = 2$, entonces la variable *triángulo* se almacenaría con el valor triángulo = 12, esto es, el primer número indica el sector (en realidad es el sector número 1, pero como se mencionó al final del apartado 3.6.4.14, se almacena en la variable *sector* como sector 0 para posteriormente trabajar con la variable *normalizar*, la variable *sector* va de 0 a 5 y los triángulos grandes van de 1 a 6), si el vector de referencia se encuentra en el sector 1 y triángulo 4 la variable quedará *triángulo* = 24.

La variable *triangulox*, se utiliza sólo para comprobar en qué número de triángulo pequeño se encuentra independientemente del sector en el que esté, por ejemplo si el vector está en el sector 0, 1, 2, 3, 4 ó 5, y el vector se encuentra en el triángulo pequeño $j = 1$, en cualquier sector, la variable *triangulox* se almacenará con un único valor *triangulox* = 1, esto para poder trabajar con la variable *normalizar* al igual que la variable *sector*.

```

void calcula_trianguo(void)
{
if(pasa[0] == 0 && pasa[1] == 0 && pasa[2] == 0 && pasa[3] == 0 && pasa[4] == 0 &&   pasa[5] == 0)
{
if(sector == 0)   {trianguo = 11; trianguox = 1;}
else if(sector == 1)   {trianguo = 21; trianguox = 1;}
else if(sector == 2)   {trianguo = 31; trianguox = 1;}
else if(sector == 3)   {trianguo = 41; trianguox = 1;}
else if(sector == 4)   {trianguo = 51; trianguox = 1;}
else if(sector == 5)   {trianguo = 61; trianguox = 1;}
}
else if(pasa[0] == 1 && pasa[5] == 0 && pasa[3] == 0) {trianguo = 12; trianguox = 2;}
else if(pasa[0] == 1 && pasa[5] == 1 && sector == 0) {trianguo = 13; trianguox = 3;}
else if(pasa[0] == 1 && pasa[3] == 1 && sector == 0) {trianguo = 14; trianguox = 4;}
else if(pasa[3] == 1 && pasa[0] == 0 && pasa[1] == 0) {trianguo = 22; trianguox = 2;}
else if(pasa[0] == 1 && pasa[3] == 1 && sector == 1) {trianguo = 23; trianguox = 3;}
else if(pasa[3] == 1 && pasa[1] == 1 && sector == 1) {trianguo = 24; trianguox = 4;}
else if(pasa[3] == 0 && pasa[1] == 1 && pasa[4] == 0) {trianguo = 32; trianguox = 2;}
else if(pasa[1] == 1 && pasa[3] == 1 && sector == 2) {trianguo = 33; trianguox = 3;}
else if(pasa[1] == 1 && pasa[4] == 1 && sector == 2) {trianguo = 34; trianguox = 4;}
else if(pasa[4] == 1 && pasa[1] == 0 && pasa[2] == 0) {trianguo = 42; trianguox = 2;}
else if(pasa[4] == 1 && pasa[1] == 1 && sector == 3) {trianguo = 43; trianguox = 3;}
else if(pasa[2] == 1 && pasa[4] == 1 && sector == 3) {trianguo = 44; trianguox = 4;}
else if(pasa[2] == 1 && pasa[4] == 0 && pasa[5] == 0) {trianguo = 52; trianguox = 2;}
else if(pasa[2] == 1 && pasa[4] == 1 && sector == 4) {trianguo = 53; trianguox = 3;}
else if(pasa[2] == 1 && pasa[5] == 1 && sector == 4) {trianguo = 54; trianguox = 4;}
else if(pasa[5] == 1 && pasa[2] == 0 && pasa[0] == 0) {trianguo = 62; trianguox = 2;}
else if(pasa[2] == 1 && pasa[5] == 1 && sector == 5) {trianguo = 63; trianguox = 3;}
else if(pasa[5] == 1 && pasa[0] == 1 && sector == 5) {trianguo = 64; trianguox = 4;}
return;
}

```

Figura 3.32. Código de la rutina *calcula_trianguo*.

3.6.4.16 Duty_cycle

El código para la rutina *duty_cicle* se muestra en la figura 3.33.

La rutina *duty_cicle* probablemente es la más importante en el cálculo de SVM, ya que en ésta se calculan los ciclos de trabajo de cada vector más cercano al vector de referencia.

Como se dijo anteriormente, se utiliza la variable *normalizar* para facilitar los cálculos; como hay una simetría en todos los sectores, las ecuaciones y las figuras que se presentan en el apartado 2.10.3.5 hacen referencia al primer sector del diagrama SVM, pero con estas mismas ecuaciones y figuras se pueden calcular los ciclos de trabajo para cualquier sector del diagrama SVM. Para el caso de la tercer línea, en

donde se escribe la instrucción “*normalizar = (sector*60);*” se escribe para que posteriormente, sólo se tenga que escribir en la función trigonométrica seno o coseno la forma normalizada $\cos(wt - \text{normalizar})$, esto es, si el vector cuenta con un ángulo de 190 grados se sabe de antemano que se ubica en el triángulo grande $i = 4$, y en el sector 3, entonces la variable *normalizar* quedaría $\text{normalizar} = (\text{sector} * 60) = 180$, entonces la función trigonométrica del coseno ya no sería $\cos(190)$ sino $\cos(10)$, lo que es un valor normalizado llevado al plano del primer sector para que las ecuaciones no se modifiquen y siempre se calcule como si el vector de referencia se ubicará en el primer sector del diagrama SVM.

```

void duty_cicle(void)
{
    normalizar = (sector*60);
    if(triangulox == 1)
    {
        d1 = m * (1.732 * cos[(wt - (normalizar))>>2] - sin[(wt - (normalizar))>>2]);
        d2 = 2 * m * sin[(wt - (normalizar))>>2];
        d3 = (1 - d1 - d2);
    }
    else if(triangulox == 2)
    {
        d1 = 1 - (2 * m * sin[(wt - (normalizar))>>2]);
        d2 = 1 + (m * (sin[(wt - (normalizar))>>2] - (1.732 * cos[(wt - (normalizar))>>2])));
        d3 = (1 - d1 - d2);
    }
    else if(triangulox == 3)
    {
        d1 = (m * (1.732 * cos[(wt - (normalizar))>>2] - sin[(wt - (normalizar))>>2])) - 1;
        d2 = 2 * m * sin[(wt - (normalizar))>>2];
        d3 = (1 - d1 - d2);
    }
    else if(triangulox == 4)
    {
        d1 = (m * (1.732 * cos[(60 - (wt - (normalizar))>>2] - sin[(60 - (wt - (normalizar))>>2])) - 1;
        d2 = 2 * m * sin[(60 - (wt - (normalizar))>>2];
        d3 = (1 - d1 - d2);
    }
    wt = wt + 4;
    if(wt == 360)
    {
        wt = 0;
    }
    periodos();
    procede = 0;
    return;
}

```

Figura 3.33. Código de la rutina *duty_cicle*.

Los ciclos de trabajo sólo tienen 4 casos diferentes, y estos dependen de dónde se ubique el vector de referencia dentro de los 4 triángulos pequeños j (*triangulox*). Una vez hechos estos cálculos de los ciclos de trabajo, sólo resta incrementar el ángulo del vector de referencia para que siga girando (ya que teóricamente es un vector giratorio), lo cual se hace en intervalos de 4 grados.

Una vez incrementado el ángulo de giro del vector de referencia, sólo hace falta traducir los ciclos de trabajo que se tienen en valores porcentuales a periodos en los temporizadores, para que éstos puedan hacer su conteo y temporización exactos que duran los disparos hacia los transistores. En la figura 3.32 se muestra el código de la subrutina *periodos* en donde se cargan los ciclos de trabajo a los registros de periodos de los temporizadores.

3.6.4.17 Periodos

El código mostrado en la figura 3.34.

```
void periodos(void)
{
    periodo[0] = (d1 * 5556);
    periodo[1] = (d2 * 5556);
    periodo[2] = (d3 * 5556);
    return;
}
```

Figura 3.34. Código de la subrutina *periodos*.

Se muestran las tres líneas en las que los ciclos de trabajos se traducen en periodos que posteriormente serán copiados al registro PRx del Temporizador, los ciclos de trabajo se multiplican por el número 5556 que será explicado en el apartado 3.6.4.17.1.

3.6.14.18 Resolución en grados y periodo de conmutación

Se tiene un periodo de línea:

$$f_0 = 60\text{Hz}$$

$$T_0 = 16.66\text{ms}$$

y se quiere trabajar a determinada frecuencia de conmutación, se tienen 360 grados en la circunferencia y hay que reducirlos, ya que si se trabaja con incrementos de grado en grado se tendrán que hacer cálculos cada grado, es decir, 360 rutinas SVM en un periodo de línea (360 periodos de conmutación en un periodo de línea), lo que para el DSC utilizado no es posible, debido a que la frecuencia de procesamiento a la que trabaja no basta para realizar toda esa serie de cálculos en un periodo de línea (16.66 ms), por lo que se decide reducir el número de veces que se ejecutará la rutina SVM, en vez de ser 360 veces se realizará 90 veces (la cuarta parte de 360 son 90 periodos de conmutación por periodo de línea). En ese caso, en un periodo de línea se tendrán 90 rutinas SVM, lo que ya es considerable para el DSC. Entonces, si:

$$\frac{T_0}{T_S} = n \quad (3.5)$$

En donde T_0 es el periodo de línea.

T_S es el periodo de conmutación.

n es el número de periodos de conmutación en un periodo de línea.

Si se despeja T_S , se tiene

$$\frac{T_0}{n} = T_S; \quad \rightarrow 185.18\mu\text{s} \quad (3.6)$$

Esto se traduce a 5400 Hz, es decir, se trabajará a una frecuencia de conmutación $T_S = 5.4$ kHz, y se tendrán que realizar todos los cálculos de la rutina SVM en 185.18 μs .

Como el DSC realiza una instrucción en 33.33 ns, esto quiere decir que cada conteo del Temporizador lo realiza en 33.33 ns, entonces, para saber qué número se tiene que cargar en el registro de periodo PRx, se sigue:

$$(185.18\mu s)(30 \times 10^6 \text{ Hz}) = 5555.5 \quad (3.7)$$

Esto es, el inverso de 33.33 ns es 30 MHz (30×10^6 Hz), por lo tanto, se le cargará el valor 5556 al registro PRx, que es el tiempo que dura un periodo de conmutación.

Como punto final de la figura 3.33 de la rutina *duty_cicle*, la variable *procede* se pone en cero, esto con el fin de no volver a hacer los cálculos de la rutina SVM hasta que los tres vectores se hayan aplicado, lo que quiere decir que primero se hacen los cálculos de SVM y posteriormente se aplican los vectores, y mientras los vectores se están aplicando, la rutina SVM se está volviendo a ejecutar para cuando sea necesario aplicar los vectores siguientes estos ya estén calculados y listos para aplicarse de nuevo y así sucesivamente los 90 periodos de conmutación en los 360 grados del diagrama SVM.

3.6.4.19 Calcula los tres vectores más cercanos

La última parte de la rutina SVM se muestra en la figura 3.35.

```
void calcula_vectores(void)
{
    if(triangulo == 11)
    {
        vector[0] = vectores[4];
        vector[1] = vectores[14];
        vector[2] = vectores[13];
    }
    else if(triangulo == 12)
    {
        vector[0] = vectores[4];
        vector[1] = vectores[14];
        vector[2] = vectores[3];
    }
    .
    .
    .
    else if(triangulo == 64)
    {
        vector[0] = vectores[8];
        vector[1] = vectores[7];
        vector[2] = vectores[4];
    }
    return;
}
```

Figura 3.35. Código de la rutina *calcula_vectores*.

En esta parte del código, cuando ya se tiene la posición en la que está el vector de referencia se calculan los tres vectores más cercanos y se aplican el periodo que se cálculo en la rutina *duty_cicle*. Se almacenan en una variable llamada *vector* de tres elementos, la que posteriormente será utilizada en las rutinas de interrupción de los temporizadores.

3.6.4.20 Rutina de servicio a la interrupción

Como parte final del código del programa, se tienen cuatro rutinas de servicio a la interrupción para cada uno de los temporizadores, las cuales se explican a continuación.

3.6.4.21 Rutina de servicio a la interrupción ISR TIMER1

El código de la primera rutina de servicio se muestra en la figura 3.36.

```
void __attribute__((interrupt, auto_psv))
_T1Interrupt(void)
{
    IFS0bits.T1IF = 0;
    T1CONbits.TON = 0;
    TMR1 = 0;

    PR2 = periodo[0];
    PR3 = periodo[1];
    PR4 = periodo[2];

    procede = 1;

    if(PR2 != 0)
    {
        LATE = vector[0]>>6;
        LATF = vector[0];
    }
    else
    {
        IFS0bits.T2IF = 1;
    }
    T2CONbits.TON = 1;
    return;
}
```

Figura 3.36. Código de la rutina de servicio a la interrupción del temporizador 1.

Esta rutina de servicio es la que se ejecuta sólo en una ocasión, una vez que se inicializa el programa, el Temporizador empieza a contar y se desborda produciendo la interrupción cuando la rutina SVM ha finalizado. Como primer paso al entrar a la rutina de servicio se tienen que borrar las banderas de interrupción, se tiene que deshabilitar la interrupción por Temporizador 1 y reiniciar el contador, se cargan los periodos calculados en los registros de periodo PRx del registro TIMERx y se pone en 1 la variable *procede*, para cuando salga de la rutina de interrupción vuelva a calcular el SVM para el siguiente periodo de conmutación. Se comprueba que el periodo del primer vector a aplicar no sea cero, si no es cero se aplican los vectores correspondientes, y si es cero, entonces se pasa directamente a la siguiente rutina de servicio, encendiendo el Temporizador 2.

3.6.4.22 Rutina de servicio a la interrupción ISR TIMER2

El código de la rutina de servicio del Temporizador 2 se muestra en la figura 3.37.

```
void __attribute__((interrupt, auto_psv))
_T2Interrupt(void)
{
    IFS0bits.T2IF = 0;
    T2CONbits.TON = 0;
    TMR2 = 0;

    if(PR3 != 0)
    {
        LATE = vector[1]>>6;
        LATF = vector[1];
    }
    else
    {
        IFS0bits.T3IF = 1;
    }
    T3CONbits.TON = 1;
    return;
}
```

Figura 3.37. Código de la rutina de servicio a la interrupción del temporizador 2.

El mismo procedimiento se sigue para la rutina de servicio del Temporizador 2 que para la rutina de servicio del Temporizador 1, sólo que en este caso los registros PRx ya fueron cargados en el Temporizador 1 por lo que no se tienen que cargar de

nuevo aquí, al salir, se enciende el Temporizador 3. Para que éste empiece a contar y mientras cuenta, los cálculos de la rutina SVM se siguen efectuando.

3.6.4.23 Rutina de servicio a la interrupción ISR TIMER3

El código de la rutina de servicio del TIMER3 se muestra en la figura 3.38.

```

void __attribute__((interrupt, auto_psv))
_T3Interrupt(void)
{
    IFS0bits.T3IF = 0;
    T3CONbits.TON = 0;
    TMR3 = 0;

    if(PR4 != 0)
    {
        LATE = vector[2]>>6;
        LATF = vector[2];
    }
    else
    {
        IFS1bits.T4IF = 1;
    }
    T4CONbits.TON = 1;
    return;
}

```

Figura 3.38. Código de la rutina de servicio a la interrupción del temporizador 3.

La tercera rutina de interrupción es exactamente igual a la segunda rutina de interrupción, sólo que aquí se borran banderas del Temporizador 3, al salir de esta rutina se enciende el Temporizador 4.

3.6.4.24 Rutina de servicio a la interrupción ISR TIMER4

El código de la rutina de servicio del Temporizador 4 se muestra en la figura 3.39.

La última rutina de servicio es la rutina del Temporizador 4, ésta funciona de la misma manera que funciona la rutina de servicio del Temporizador 1, en este punto la rutina SVM ya debe haber terminado de hacer todos los cálculos, se cargan los periodos nuevos y empiezan a aplicarse, se levanta la bandera *procede* y se vuelve a entrar a la rutina SVM, a la salida de esta rutina se enciende el Temporizador 2 y se convierte en un ciclo de cálculo y aplicación de vectores.

```
void __attribute__((interrupt, auto_psv))
_T4Interrupt(void)
{
    IFS1bits.T4IF = 0;
    T4CONbits.TON = 0;
    TMR4 = 0;

    PR2 = periodo[0];
    PR3 = periodo[1];
    PR4 = periodo[2];

    procede = 1;

    if(PR2 != 0)
    {
        LATE = vector[0]>>6;
        LATF = vector[0];
    }
    else
    {
        IFS0bits.T2IF = 1;
    }
    T2CONbits.TON = 1;
    return;
}
```

Figura 3.39. Código de la rutina de servicio a la interrupción del temporizador 4.

3.6.5 Implementación del circuito del DSC

Una vez realizado el programa completo, sólo hace falta probarlo cargándolo al DSC dsPIC30F4011, la implementación física se muestra en la figura 3.40.

Se puede observar que el circuito cuenta con todas sus alimentaciones y referencias, los puertos E y F se utilizan para las señales de disparo de los transistores, el canal AN8 se utiliza para la entrada analógica que representa el vector de referencia, y cuenta con un circuito de *reset* por si el programa tiene alguna falla que pueda ser reiniciado.

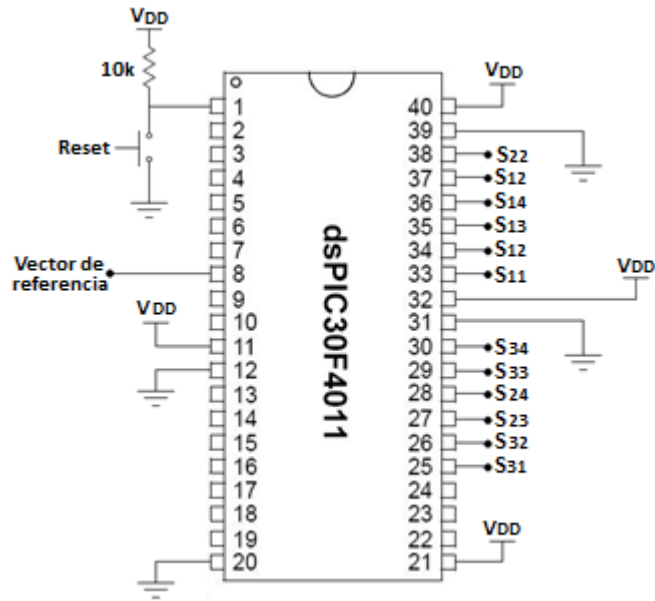


Figura 3.40. Diagrama esquemático del DSC.

En la figura 3.41 se muestra la parte digital en la placa de circuito impreso.

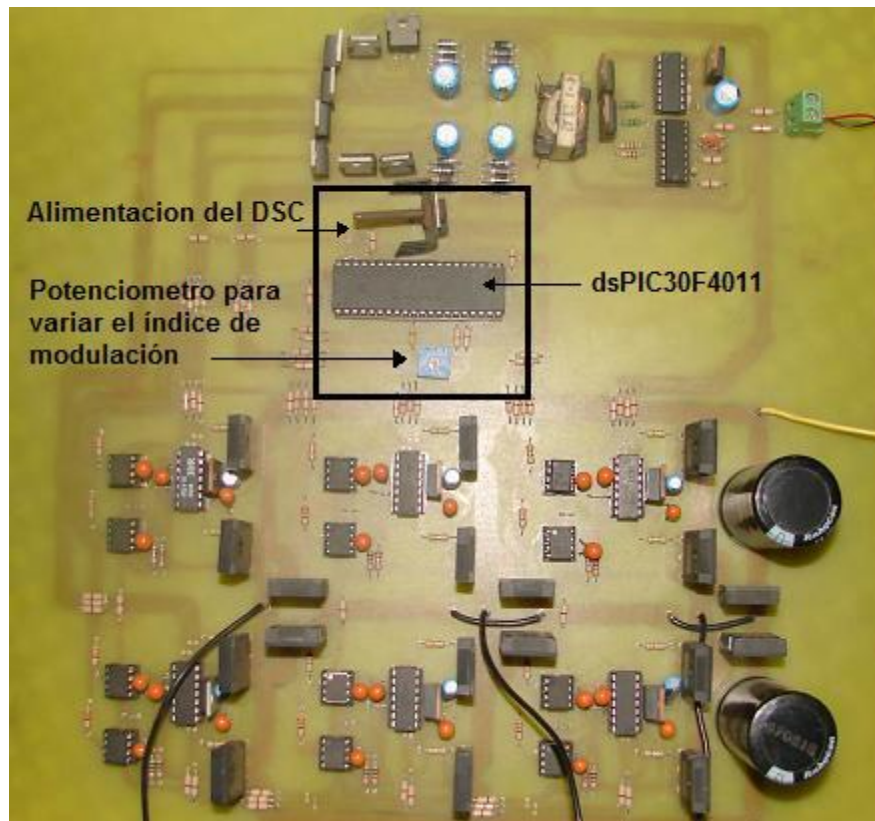


Figura 3.41. Control digital del inversor.

3.7 Implementación de la técnica SVM aplicada a un inversor trifásico de tres niveles con diodo anclado al punto neutro

En la figura 3.42 se muestra la placa de circuito impreso para todo el sistema. Se muestran además los bloques que componen el sistema.

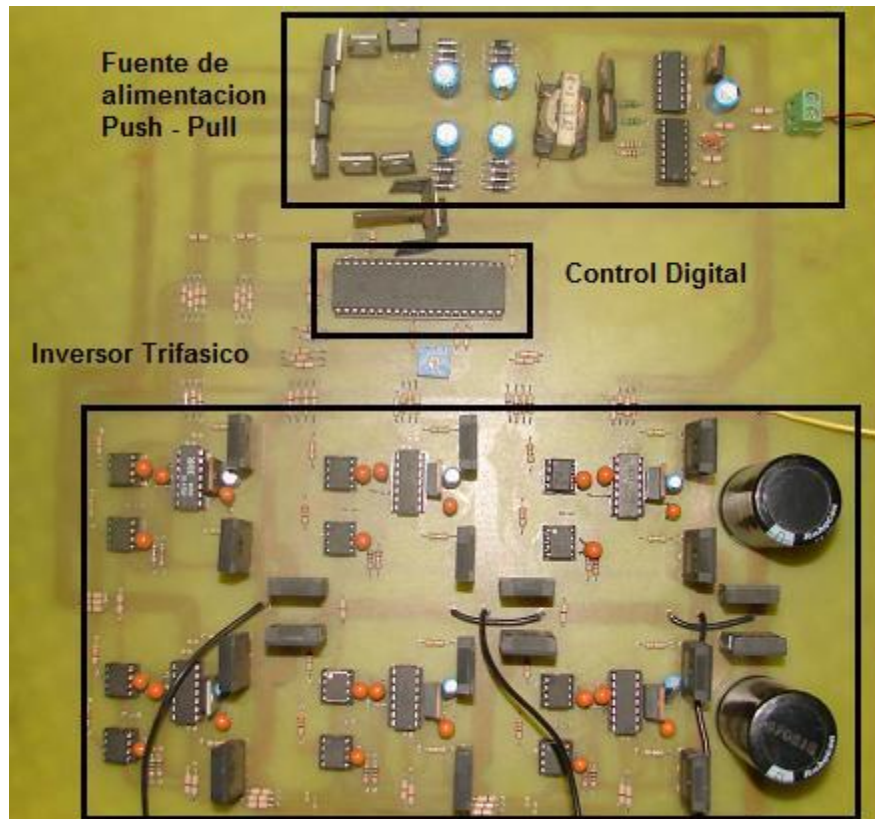


Figura 3.42. Placa de circuito impreso final.

Hasta aquí se muestra a detalle cómo se implementó el sistema para la Modulación de Vectores Espaciales aplicada a un Inversor Trifásico de Tres Niveles con Diodo Anclado al Punto Neutro, en el siguiente capítulo se mostrarán las pruebas y resultados obtenidos en la placa de circuito impreso.

Referencias

- [1] MUHAMMAD H. Rashid, "Electrónica de Potencia, Circuitos, Dispositivos y Aplicaciones." Prentice Hall, Tercera edición, 2004.
- [2] A. NABAE, I. Takahashi, and H. Akagy, "A Neutral-Point Clamped PWM Inverter." IEEE Transactions on Industrial Applications, 1981.
- [3] MARTÍN Prats M. Ángeles, "Nuevas Técnicas de Modulación Vectorial para Convertidores Electrónicos de Potencia Multinivel" Tesis de Doctor por la Universidad de Sevilla, Sevilla, Junio de 2003.
- [4] PEÑUELAS Machado José Ángel, "Algoritmo de Seguimiento del Punto de Máxima Potencia para Sistemas Fotovoltaicos en Cd. Obregón, Sonora", Tesis de Ingeniero en Electrónica, Instituto Tecnológico de Sonora, Cd. Obregón, Sonora, Octubre de 2008.
- [5] J. Fernando Silva, N. Rodrigues, J. Costa, "Space Vector Alpha-Beta Sliding mode Current Controllers for Three-Phase Multilevel Inverters" IEEE Transactions on Industrial Electronics, Lisboa, Portugal, 2000.
- [6] D. Grahame Holmes, Thomas A. Lipo, "Pulse Width Modulation For Power Converters, Principles And Practice", IEEE Press Series on Power Engineering, 2003.
- [7] J. Fernando Silva, N. Rodrigues, J. Costa, "Space Vector Alpha-Beta Sliding Mode Current Controllers for Three-Phase Multilevel Inverters" IEEE Transactions, 2000.
-

CAPÍTULO IV

Pruebas y resultados

En este capítulo se muestran las pruebas realizadas a la implementación del Inversor trifásico de 3 Niveles con Diodo Anclado al punto Neutro. Se verifica la implementación de la técnica SVM, así como el funcionamiento del inversor con carga resistiva y un motor trifásico. Se muestran las gráficas de tensión a la salida del inversor, con diferentes índices de modulación, así como la corriente en cada fase de salida y en el punto neutro. Se muestran además los disparos de compuerta a fuente de los transistores, para verificar su forma de onda con la que son excitados por los drivers.

4.1 Disparo de compuerta a fuente

Para que los transistores puedan conducir, las compuertas deben ser excitadas con las señales PWM de la técnica SVM. Los drivers IR2110 son los encargados de excitar las compuertas de los transistores. Se utiliza el osciloscopio Tektronix modelo DPO7104 del laboratorio de Eléctrica y Electrónica del ITSON, la señal PWM de compuerta a fuente del transistor S_{11} se puede ver en la figura 4.1. La escala de voltaje del osciloscopio es de 5 V/Div.

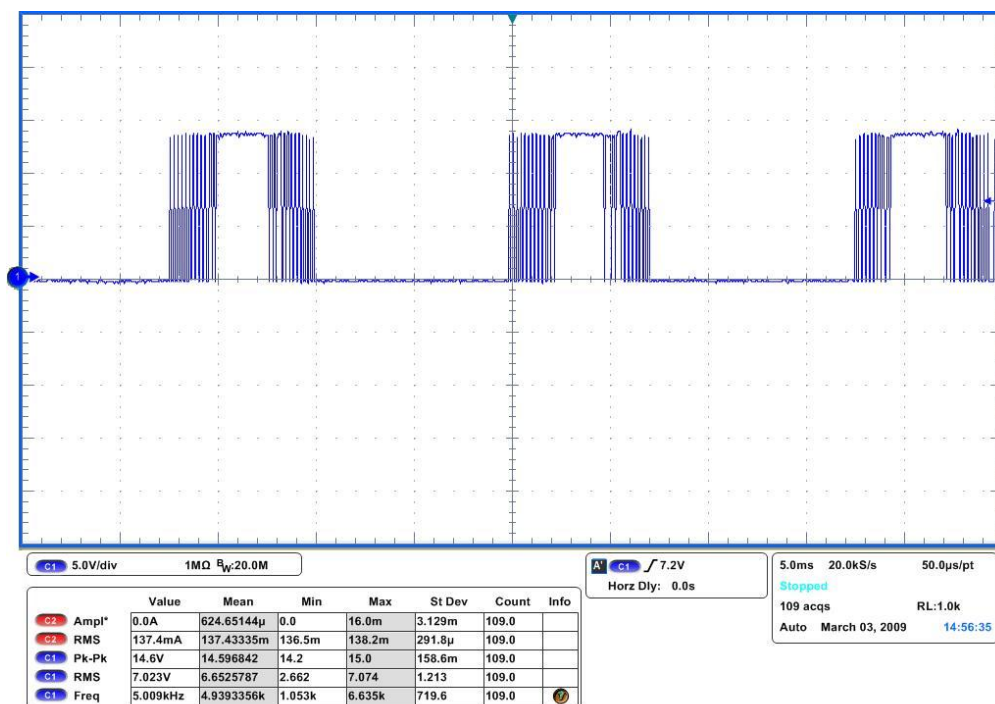


Figura 4.1. Señal de disparo compuerta a fuente del transistor S_{11} .

La figura 4.1 muestra el disparo de compuerta a fuente V_{DS} , el cual tiene un valor pico-pico de 14.5 V_{CD} y un índice de modulación de 0.8 (80%) del valor total, se puede ver claramente cómo los pulsos siguen una variación modulada por anchura de pulso, empiezan siendo muy delgados, y a la mitad del ciclo se hacen más anchos, y al final del ciclo terminan siendo delgados como los del inicio del ciclo.

De esta misma forma, el resto de los transistores son excitados para que puedan conducir el voltaje V_{DS} e ir formando los 3 niveles de tensión de fase del inversor. En

la figura 4.2 se muestra como colocar la punta del osciloscopio para ver el disparo de compuerta a fuente del transistor S_{11} , colocando la punta del osciloscopio positiva en la pata 1 del transistor, que corresponde a la compuerta, y la referencia en la pata 3, que corresponde a la fuente.



Figura 4.2. Ubicación de las terminales del disparo de compuerta a fuente en el osciloscopio.

4.2 Forma de onda de salida del inversor con carga resistiva

El voltaje de salida de fase V_{AN} del inversor, tiene una forma de onda de pulsos característica, la cual se muestra en la figura 4.3, con un índice de modulación de 0.2, a una frecuencia fundamental de 60 Hz, en una carga trifásica resistiva de $R = 50 \Omega$. La escala de voltaje del osciloscopio es de 5 V/Div.

Se muestra el voltaje de fase V_{AN} , y se establece un índice de modulación mínimo del 20%, para no tener que apagar todos los transistores al hacer el índice de modulación 0, ya que esto significa que ningún transistor está encendido, y se puede observar que los pulsos son muy delgados, es decir, al aplicar un filtro a la salida el

voltaje será muy pequeño, apenas el 20% del valor total que puede lograr con el V_{CD} de entrada.

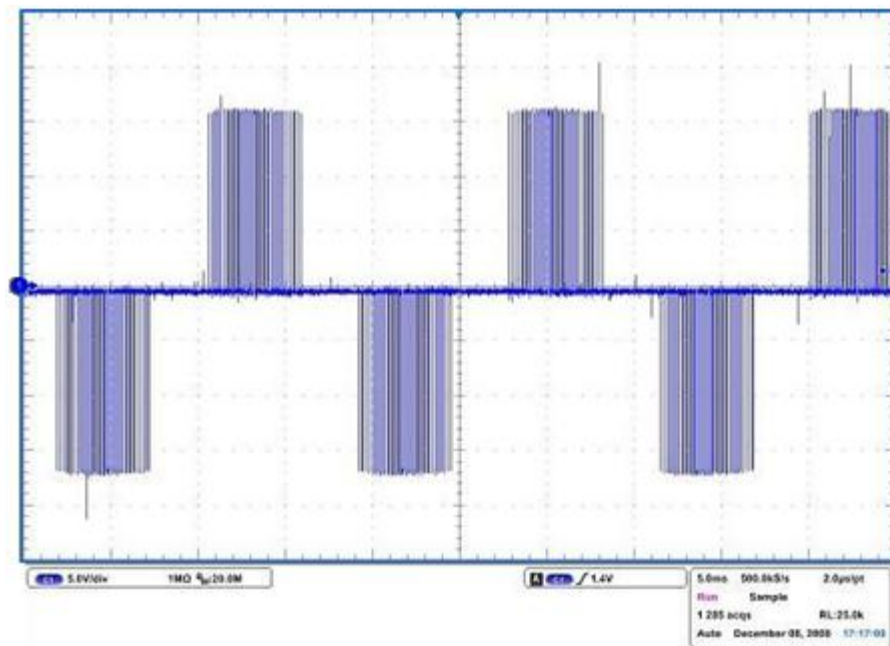


Figura 4.3. Voltaje de fase V_{AN} del inversor trifásico con carga resistiva para un índice de modulación de 0.2 y una frecuencia de 60 Hz.

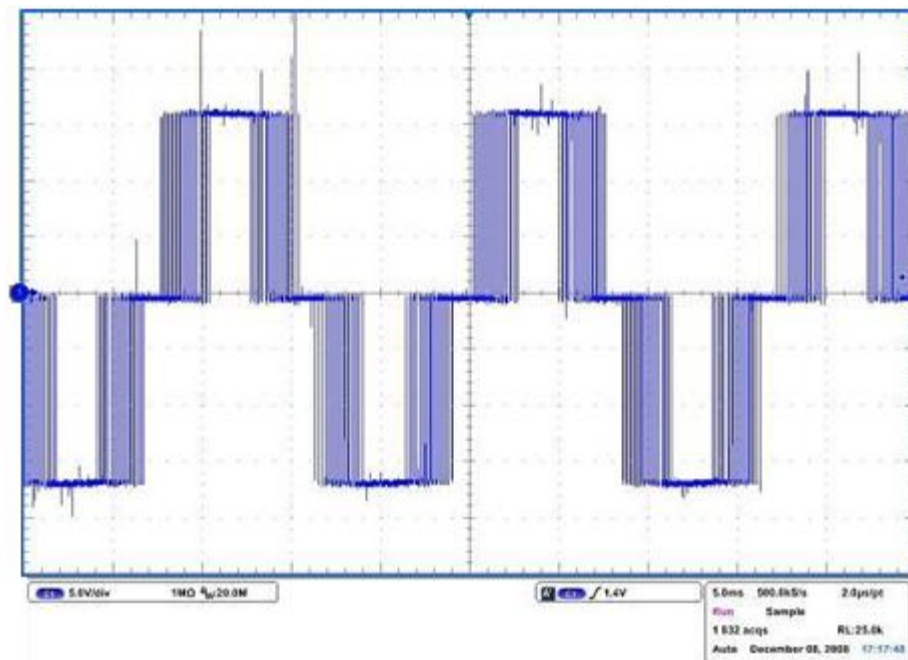


Figura 4.4. Voltaje de fase V_{AN} del inversor trifásico con carga resistiva para un índice de modulación de 0.8 y una frecuencia de 60 Hz.

En la figura 4.4 se muestra el mismo voltaje de fase V_{AN} con un índice de modulación de 0.8. La escala de voltaje del osciloscopio es de 5 V/Div.

En este caso, el índice de modulación máximo que se puede lograr es de 0.8, es decir, el 80% del total que se puede lograr con el V_{CD} de entrada, esto es, con el motivo de que el vector giratorio de la técnica SVM recorra todos los sectores y pueda tocar todos los triángulos, y con esto poder lograr un voltaje de salida modulado por anchura de pulso.

Para observar en forma simultánea la forma de onda de las 3 señales alternas trifásicas a la salida del inversor, se utilizan 3 canales en el osciloscopio. En la figura 4.5 se pueden observar los 3 voltajes de fase del inversor con una separación de 120 grados entre sí, y un índice de modulación de 0.2. La escala del osciloscopio es de 5 V/Div.

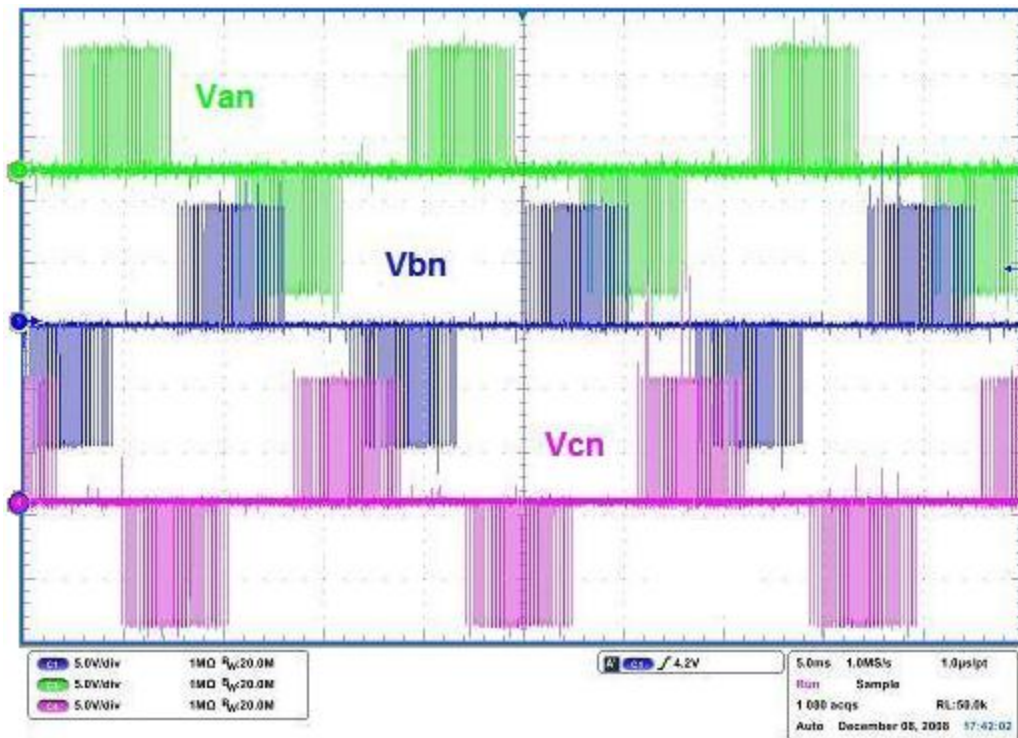


Figura 4.5. Voltaje de fase V_{AN} , V_{BN} , y V_{CN} con carga resistiva, índice de modulación de 0.2 y frecuencia de 60 Hz.

Las 3 señales están desfasadas entre sí 120 grados, y tienen un índice de modulación del 20%, es por eso que los pulsos son delgados, y al aplicar un filtro a su salida el voltaje sería el mínimo a obtener en el inversor.

En la figura 4.6 se muestran los 3 voltajes de fase, esta vez con un índice de modulación de 0.8. La escala de voltaje del osciloscopio es de 5 V/Div.

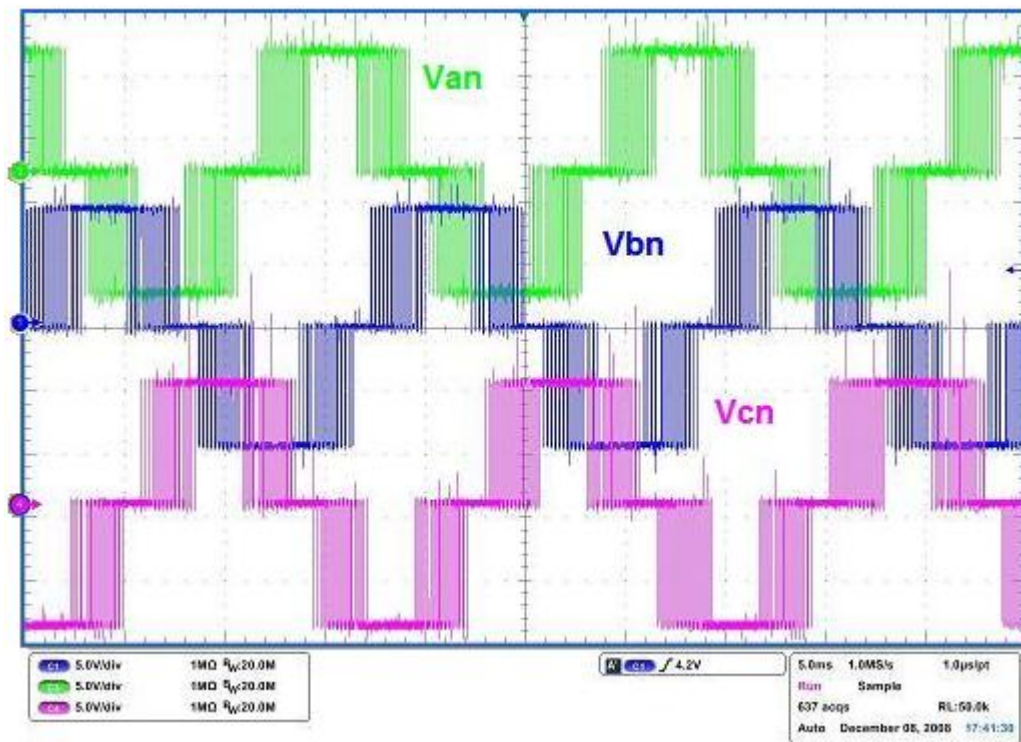


Figura 4.6. Voltaje de fase V_{AN} , V_{BN} , y V_{CN} con carga resistiva, índice de modulación de 0.8 y frecuencia de 60 Hz.

En la figura 4.6 se puede observar que el índice de modulación es casi el máximo, ya que se dibuja un pulso muy ancho justo en el medio, esto es equivalente a decir que es el voltaje máximo que se puede tener para el inversor implementado.

Una vez vistos los voltajes de fase a neutro del inversor, también es posible ver los voltajes de línea, en la figura 4.7 se muestra el voltaje de línea V_{AB} , con un índice de modulación de 0.2. La resolución del osciloscopio es de 5 V/Div.

En este caso, se puede observar que la forma de onda que se genera a la salida en la figura 4.7 está incompleta, esto es, el voltaje es el mínimo y no puede verse un voltaje pico a pico mayor.

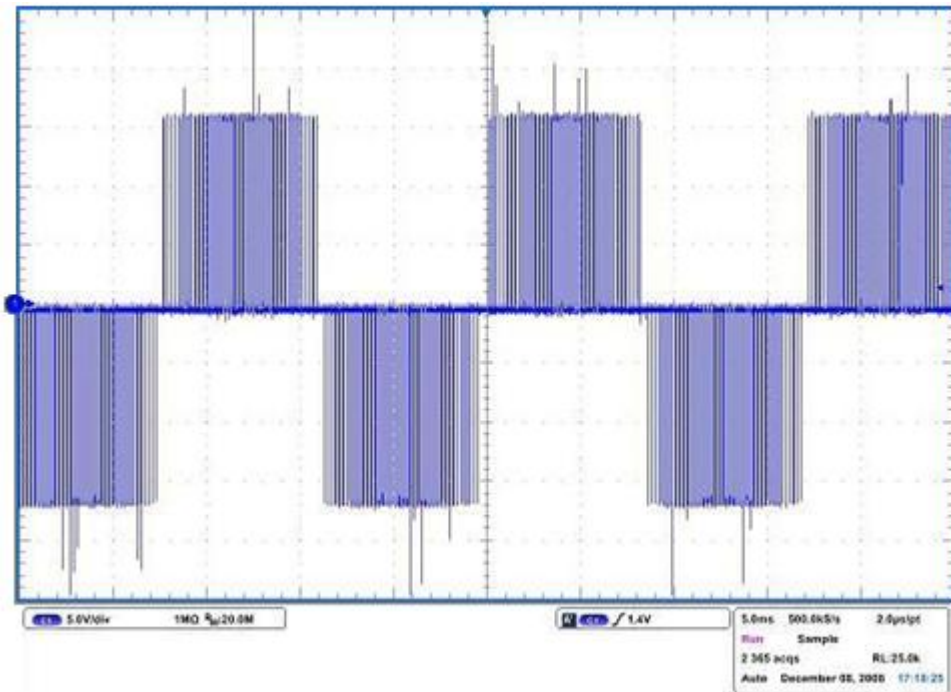


Figura 4.7. Voltaje de línea V_{AB} con carga resistiva, índice de modulación de 0.2 y frecuencia 60 Hz.

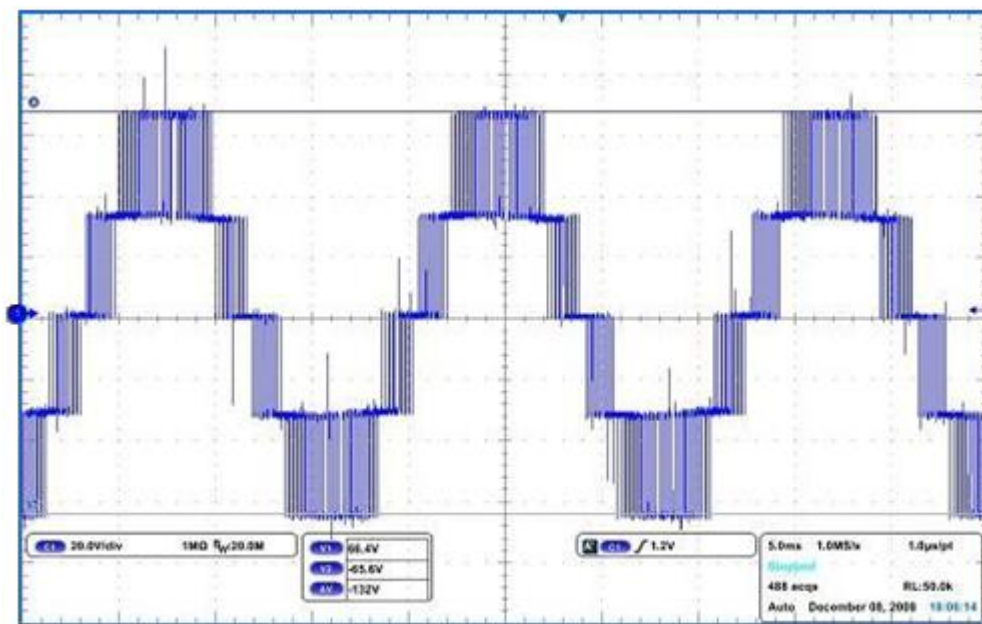


Figura 4.8. Voltaje de línea V_{AB} con carga resistiva, índice de modulación de 0.8 y frecuencia 60 Hz.

En la figura 4.8 se muestra la misma tensión de salida, sólo que esta vez el índice de modulación es de 0.8, y se puede apreciar de mejor manera los escalones que se forman en el voltaje de línea V_{AB} . La escala de voltaje del osciloscopio es de 20 V/Div.

En la figura 4.8, se puede observar de mejor manera cómo se forman los escalones de tensión del voltaje de línea V_{AB} , ya que el índice de modulación es de 80% y se puede genera una forma de onda máxima debido a que el voltaje es el mayor.

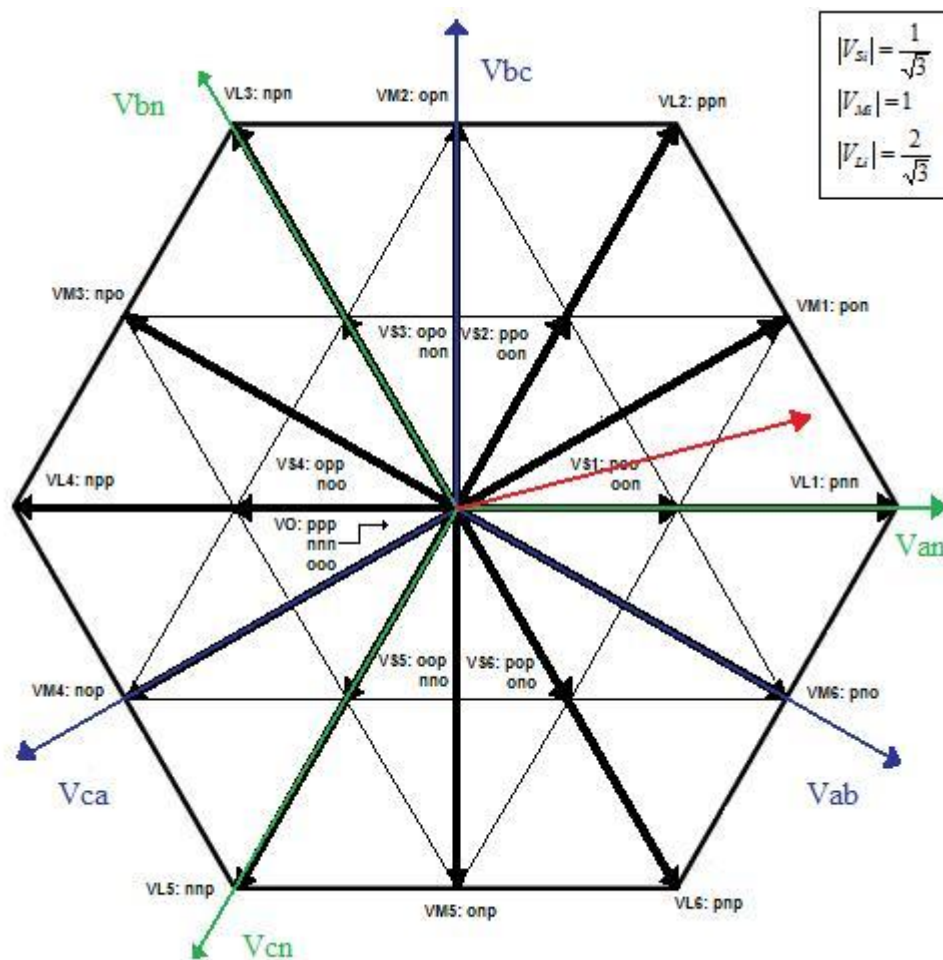


Figura 4.9. Vector de referencia con un índice de modulación de 0.8.

Para observar cómo se va dibujando la forma de onda de la figura 4.8 a partir del diagrama SVM de la figura 2.10, se presenta la figura 4.9, en donde se muestra el

vector de referencia con una amplitud de 0.8, lo que significa un índice de modulación del 80%.

4.3 Forma de onda de salida del inversor con carga inductiva

Para realizar las pruebas al inversor con una carga inductiva, se utilizó un motor de inducción trifásico de 1/3 de caballo de potencia, del laboratorio de Eléctrica del ITSON.

En la figura 4.10 se puede observar la forma de onda de salida del inversor trifásico, conectado a un motor de inducción con un índice de modulación de 0.2. La escala de voltaje del osciloscopio es de 20 V/Div.

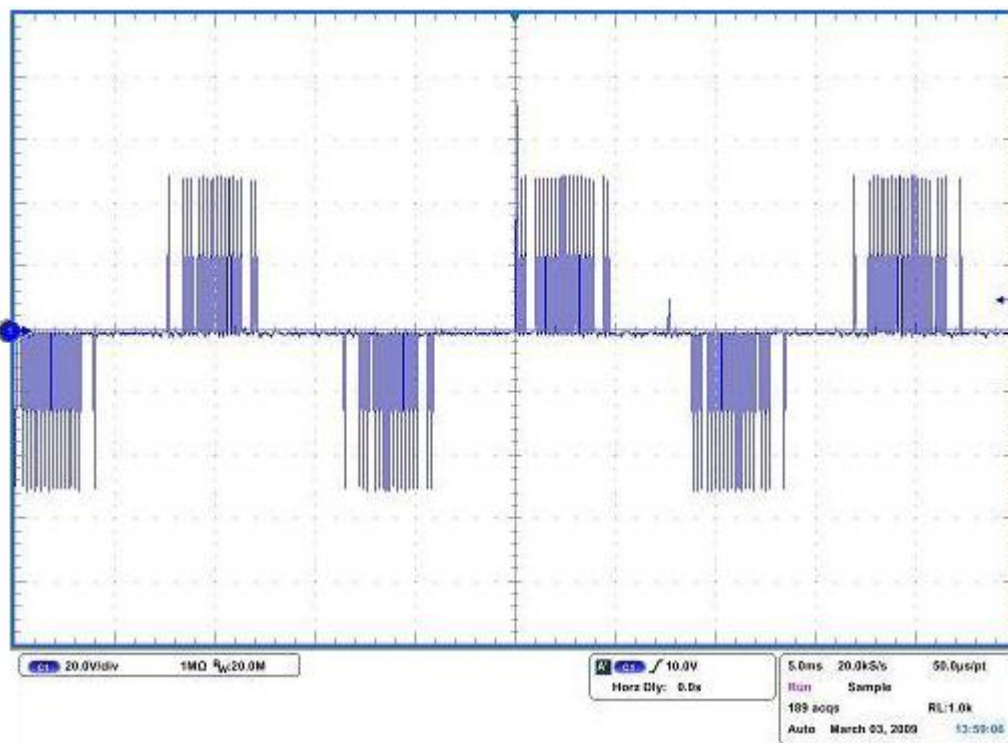


Figura 4.10. Voltaje de fase V_{AN} del inversor trifásico con carga inductiva para un índice de modulación de 0.2 y una frecuencia de 60 Hz.

Se puede observar, que a comparación de la figura 4.3, existe una muy ligera deformación de la señal, esto debido al cambio de carga de resistiva a inductiva, sin

embargo se sigue presentando una buena señal que al ser filtrada se traduce en una señal senoidal.

Para la misma señal, con un índice de modulación de 0.8, se muestra el resultado en la figura 4.11. La resolución del osciloscopio es de 20 V/Div.

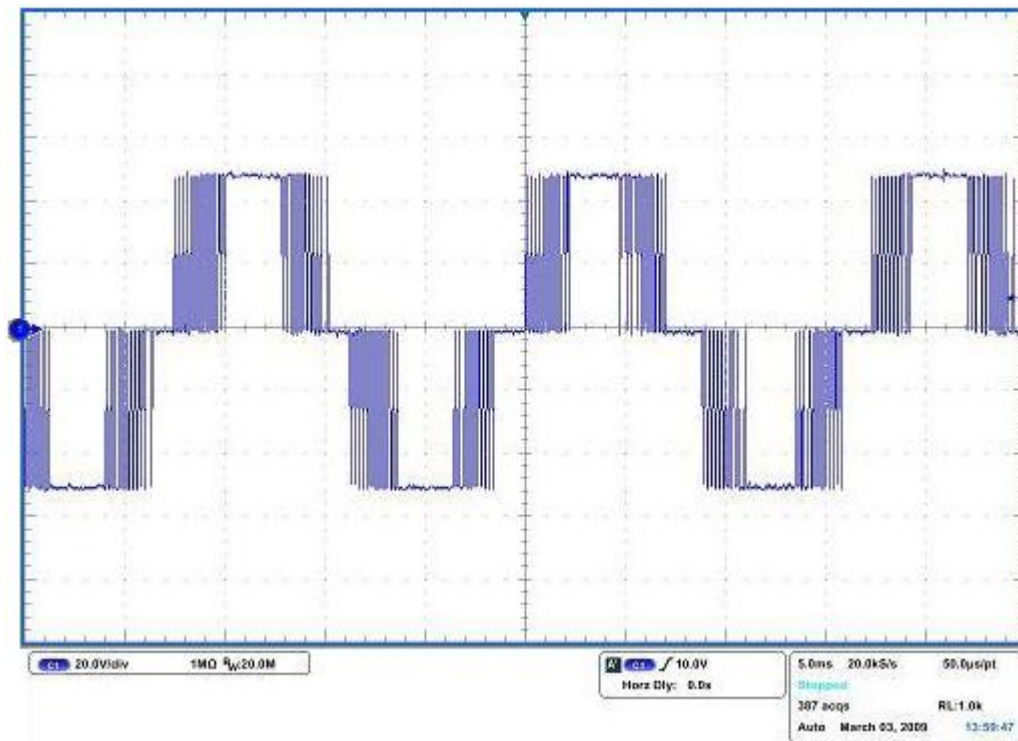


Figura 4.11. Voltaje de fase V_{AN} del inversor trifásico con carga inductiva para un índice de modulación de 0.8 y una frecuencia de 60 Hz.

En la figura 4.11, se observa una ligera deformación en la señal, debido al cambio de carga de resistiva a inductiva en el voltaje de fase V_{AN} .

En la figura 4.12 se observan las 3 señales de salida del inversor con una separación de 120 grados entre ellas, esta vez en una carga inductiva, la forma de onda es muy similar a la presentada en la figura 4.5, en este caso cada una de las fases representa a las fases del motor de inducción, en la figura 4.12 el índice de modulación es de 0.2. La escala de voltaje del osciloscopio es de 10 V/Div.

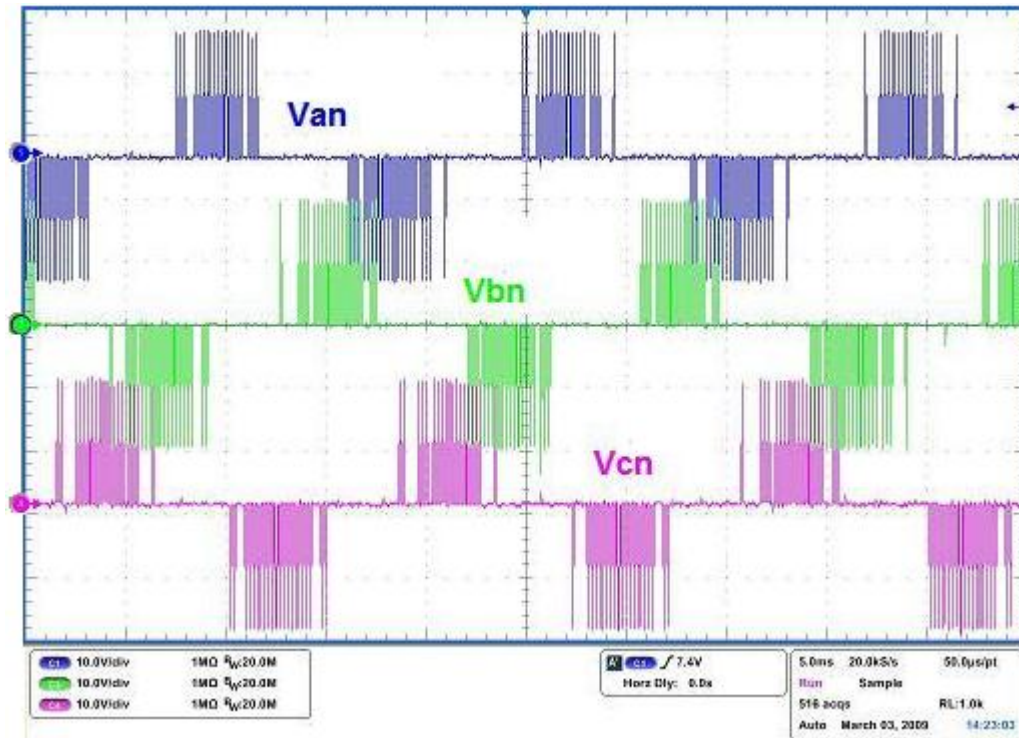


Figura 4.12. Voltaje de Fase V_{AN} , V_{BN} , y V_{CN} con carga inductiva, índice de modulación de 0.2 y frecuencia 60 Hz.

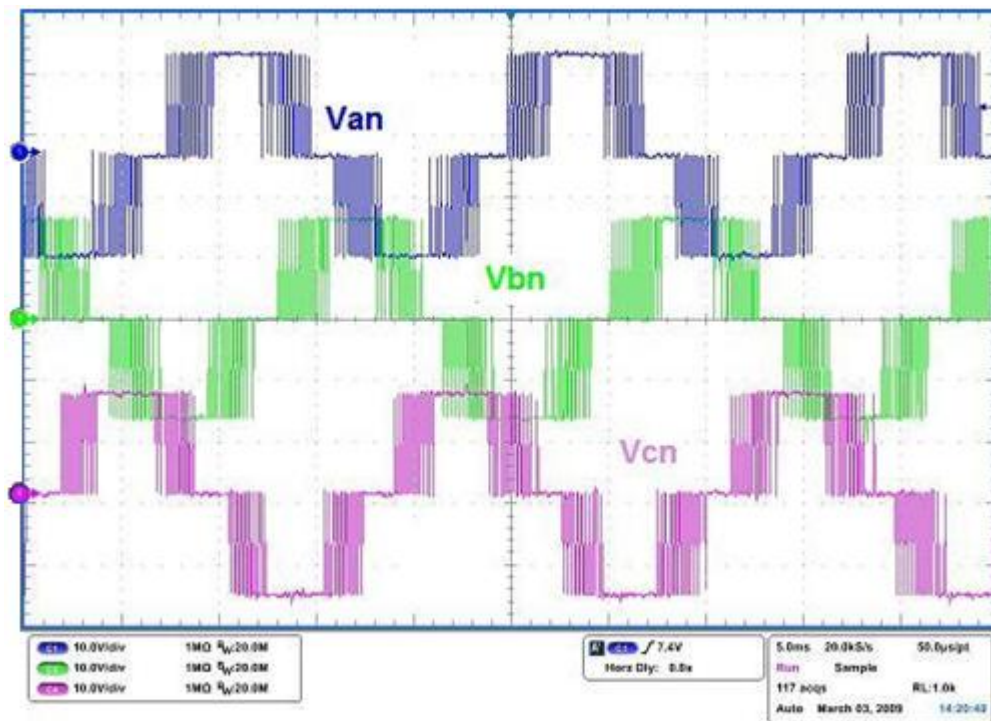


Figura 4.13. Voltaje de Fase V_{AN} , V_{BN} , y V_{CN} con carga inductiva, índice de modulación de 0.8 y frecuencia 60 Hz.

En la figura 4.13 se muestra la misma señal que la figura 4.12, con un índice de modulación de 0.8. La escala de voltaje del osciloscopio de 10 V/Div.

En la figura 4.14 se muestra el voltaje de línea V_{AB} , con un índice de modulación de 0.2 y una escala de voltaje del osciloscopio de 40 V/Div.

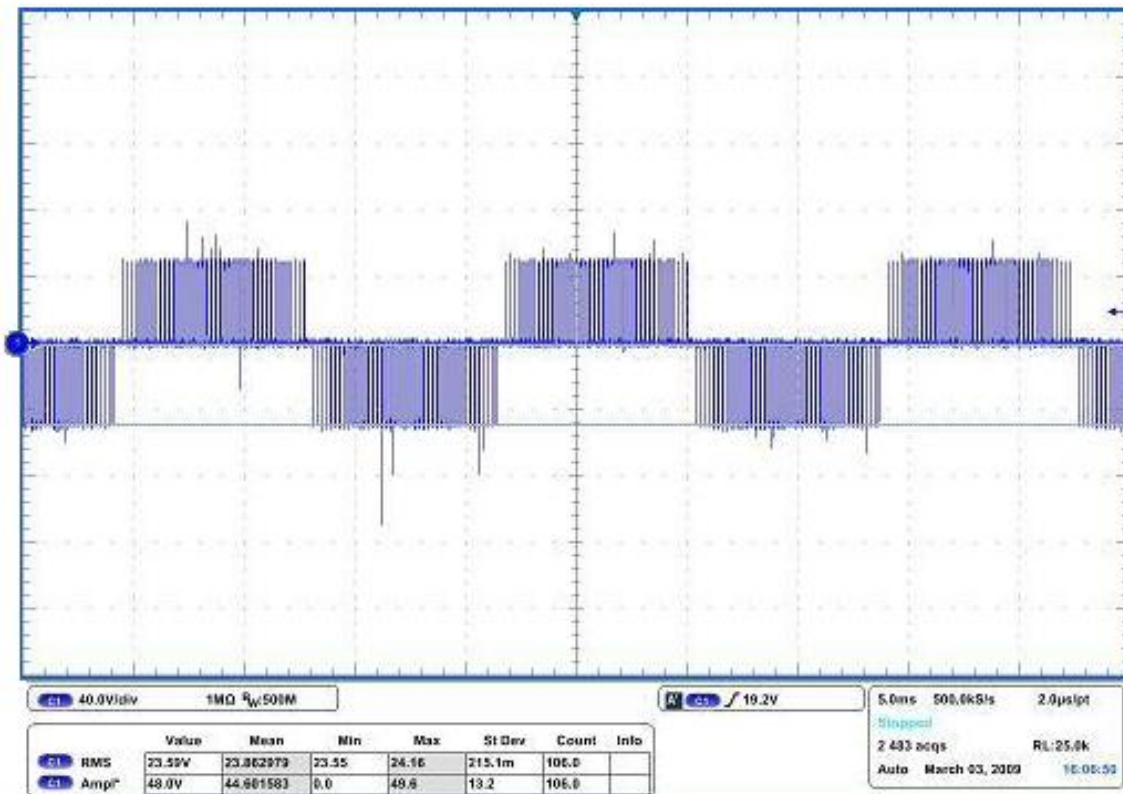


Figura 4.14. Voltaje de línea V_{AB} con carga inductiva, índice de modulación de 0.2 y frecuencia 60 Hz.

Para el voltaje de línea V_{AB} con un índice de modulación de 0.8, se presenta la figura 4.15, en la cual se observa que hay una amplitud pico-pico de 200V, y la forma de onda es casi exacta a la que se presentó en la figura 4.8 con carga resistiva. La escala de voltaje del osciloscopio es de 40 V/Div.

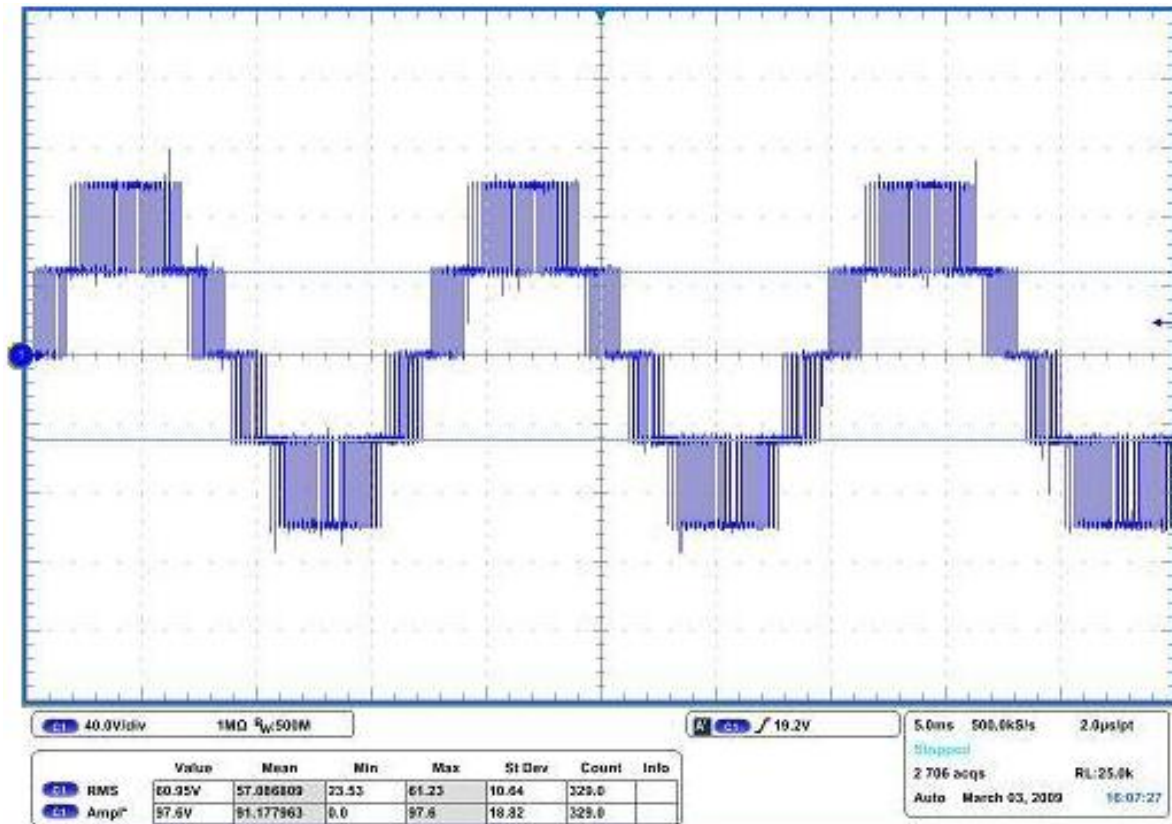


Figura 4.15. Voltaje de línea V_{AB} con carga inductiva, índice de modulación de 0.8 y frecuencia 60 Hz.

4.4 Forma de onda de corriente de salida del inversor con carga resistiva

Utilizando la punta de corriente con la que se cuenta en el laboratorio de Electrónica del ITSON, es posible medir la corriente de cada una de las fases de salida del inversor.

En la figura 4.16 se puede observar la corriente de fase “a”.

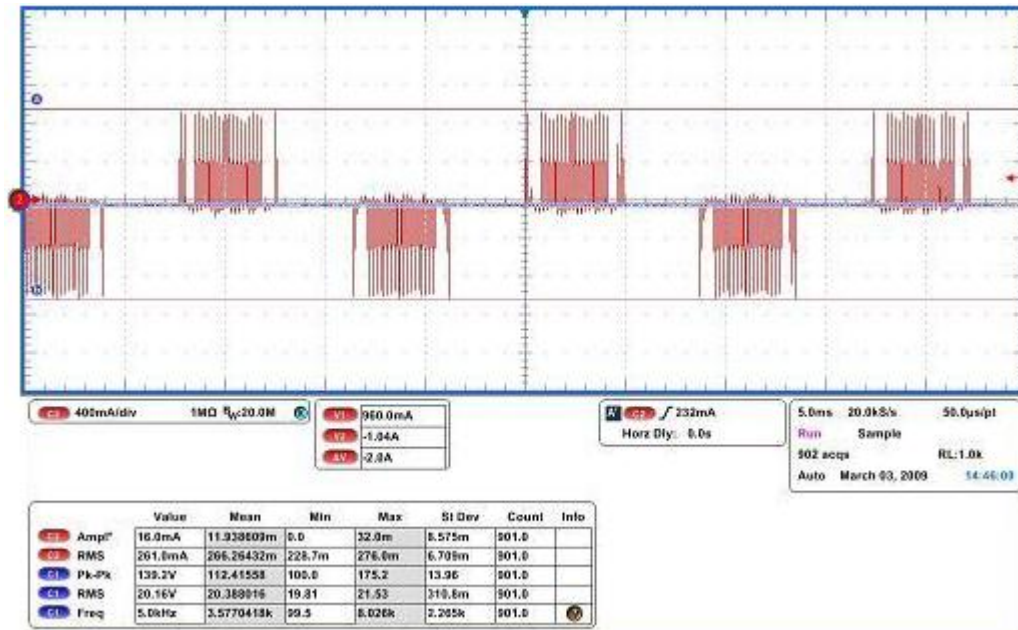


Figura 4.16. Forma de onda de corriente de fase “a” con carga resistiva, y un índice de modulación de 0.2.

Como se puede observar en la figura 4.16, en la fase “a” del inversor existe una amplitud de 2 A pico-pico, y la forma de onda corresponde al índice de modulación más bajo, es decir, al 20% del valor total, el cual es alimentado con 100 V_{CD}, esto se traduce en una corriente efectiva de 261 mA RMS.

De la misma manera, se puede observar en la figura 4.17, donde el índice de modulación se elevó a 0.8, y la forma de onda de corriente se parece mucho a la forma de onda de voltaje cuando el índice de modulación es el máximo, que la amplitud de corriente es de 2 A pico-pico, y se traduce en una corriente efectiva de 625.8 mA RMS.

Las gráficas de corriente de la fase “b” y fase “c” son iguales a la de fase “a”, debido a que no se pueden medir las 3 corrientes de fase al mismo tiempo porque sólo se cuenta con una punta de corriente, sólo se presentarán las gráficas de corriente de la fase “a” mostradas en las figuras 4.16 y 4.17.

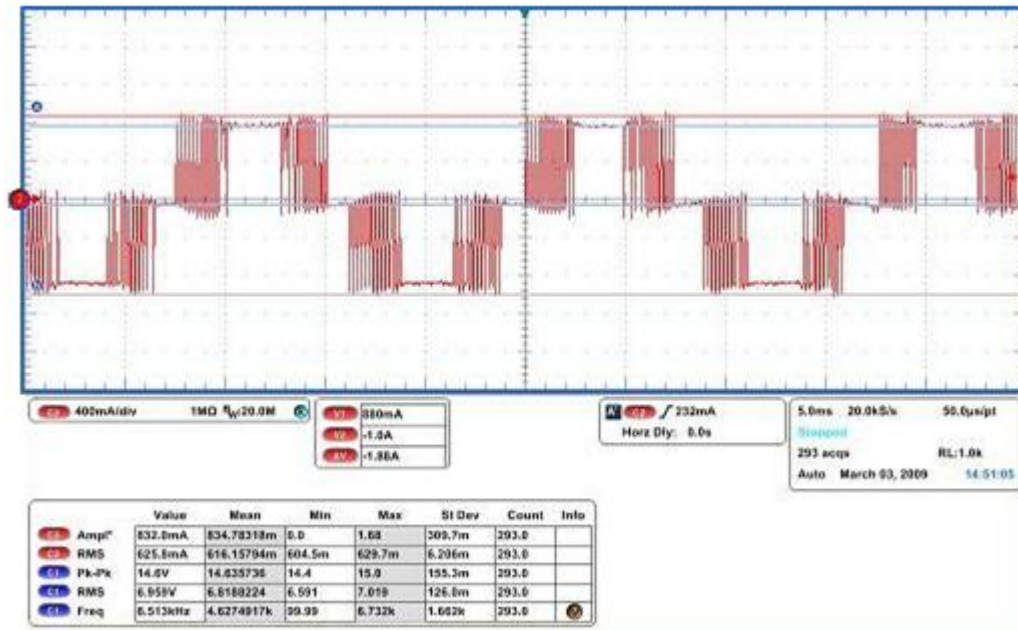


Figura 4.17. Forma de onda de corriente de fase "a" con carga resistiva, y un índice de modulación de 0.8.

4.5 Corriente en el neutro

Idealmente, en un sistema de carga trifásica balanceada, no debería existir ninguna corriente de fuga por el neutro, debido a que las cargas son balanceadas y las corrientes en cada fase idealmente deberían ser iguales, pero en la realidad no lo son, y existe una corriente de fuga que circula hacia el punto neutro de la carga, corriente que fluye por los Diodos Anclados, de ahí el nombre de la topología NPC (*Neutral Point Clamped*). Esta corriente de fuga fue medida en el osciloscopio, y la forma de onda de la misma es presentada en la figura 4.18, para un índice de modulación de 0.2, en la figura 4.19 se presenta para un índice de modulación de 0.8.

Cuando se aplica el 20% del voltaje total, existe una corriente de fuga por el neutro de 424 mA RMS, y cuando es del 80% es de 576 mA RMS.

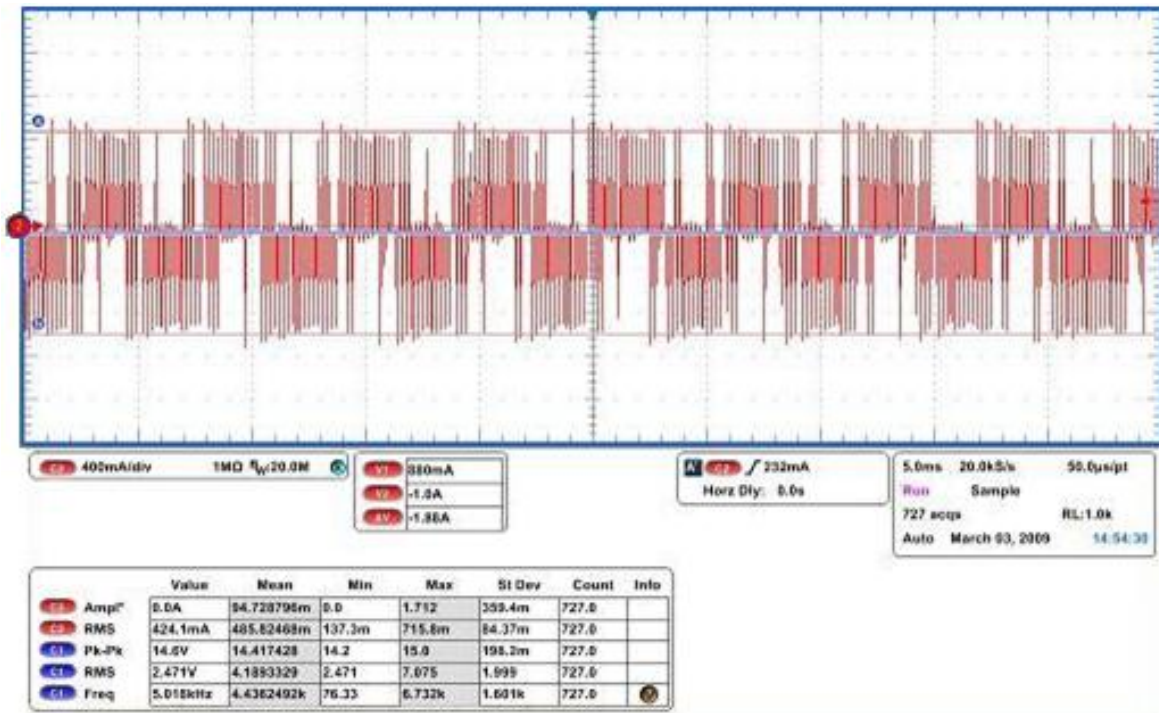


Figura 4.18. Forma de onda de corriente en el neutro con carga resistiva, y un índice de modulación de 0.2.

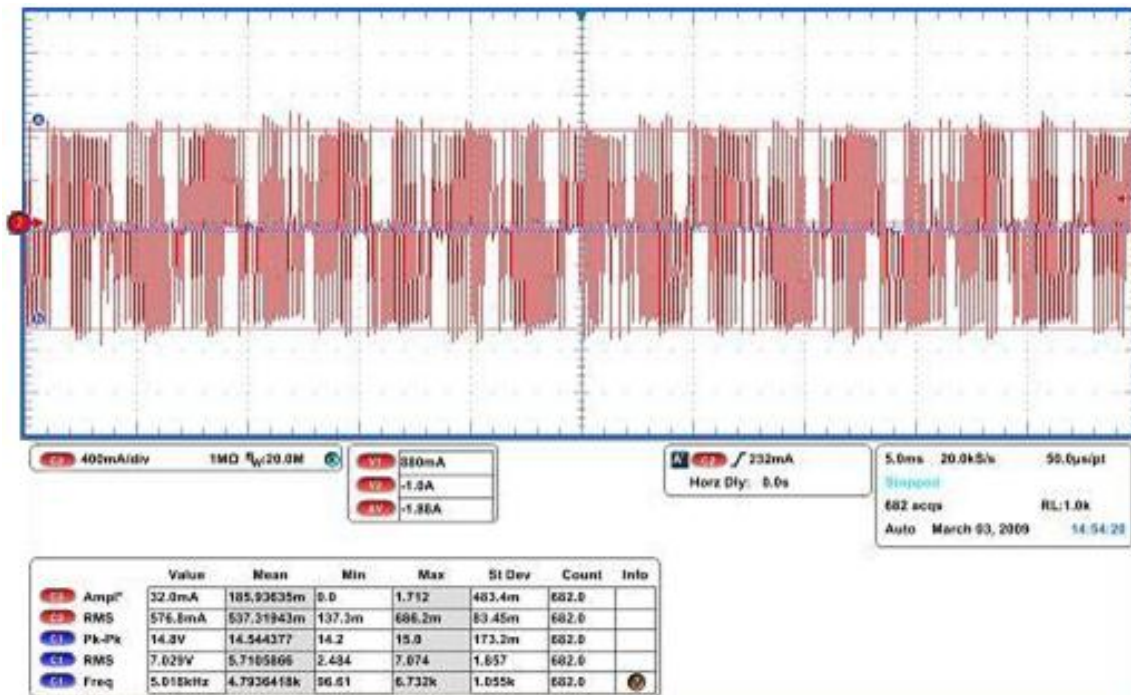


Figura 4.19. Forma de onda de corriente en el neutro con carga resistiva, y un índice de modulación de 0.8.

Por último, se tiene la gráfica de corriente con carga inductiva, la cual no se asemeja mucho una señal senoidal perfecta, debido a que la señal que se inyecta al motor no está filtrada. En la figura 4.20 se muestra la gráfica de corriente de fase “a” con carga inductiva.

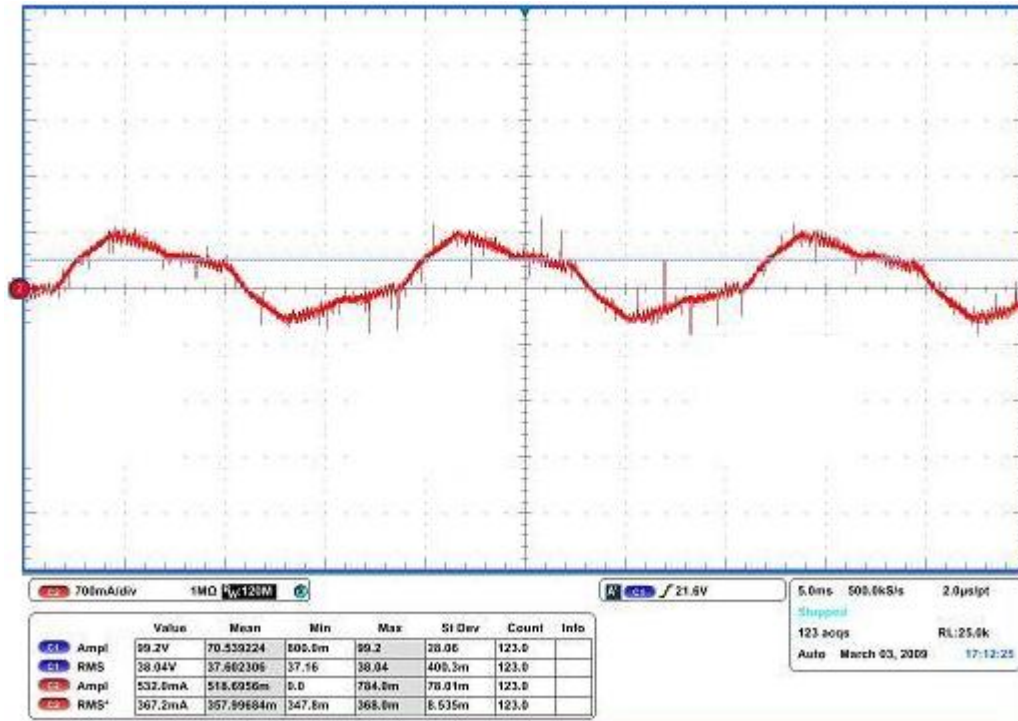


Figura 4.20. Forma de onda de corriente de fase “a” con carga inductiva, e índice de modulación de 0.8.

En ésta gráfica, la resolución del osciloscopio es de 700 mA/Div. Y se cuenta con una corriente RMS de 367.2 mA. Con un índice de modulación de 0.8.

4.6 Espectro de frecuencia

En la figura 4.21, se muestra el espectro de frecuencia, para el voltaje de fase V_{AN} , con un índice de modulación de 0.2.

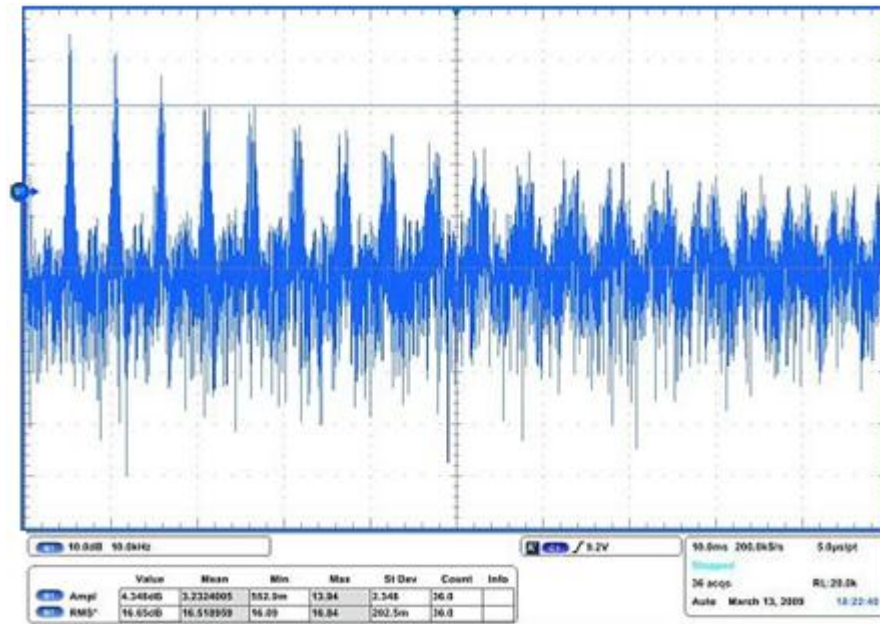


Figura 4.21. Espectro de frecuencia del voltaje de fase V_{AN} con índice de modulación de 0.2 y carga resistiva $R = 50 \Omega$.

En la figura 4.21 se observa que existe una amplitud de 4.348 dB y 16.65 dB RMS, a una frecuencia de 10 kHz.

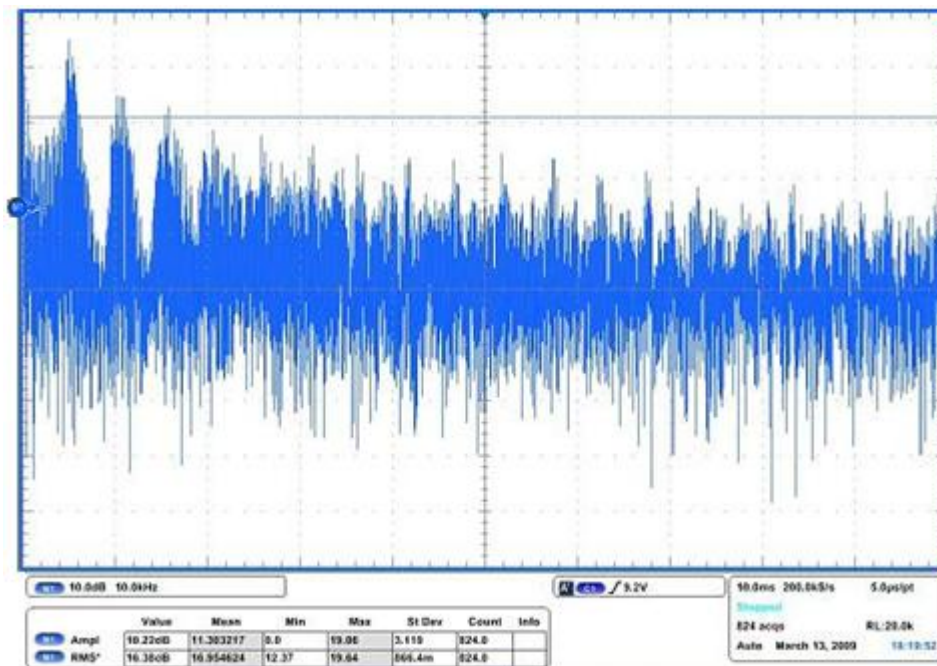


Figura 4.22. Espectro de frecuencia del voltaje de fase V_{AN} con índice de modulación de 0.8 y carga resistiva $R = 50 \Omega$.

En la figura 4.22 se muestra el mismo espectro de frecuencia, con un índice de modulación de 0.8.

En este caso, la amplitud aumentó a 10.22 dB y la amplitud RMS permaneció igual que el caso de la figura 4.21 a una frecuencia de 10 kHz.

Para el caso del voltaje de línea V_{AB} , también se muestran los espectros de frecuencia, en la figura 4.23 se muestra el espectro de frecuencia con un índice de modulación de 0.2 para el voltaje de línea V_{AB} .

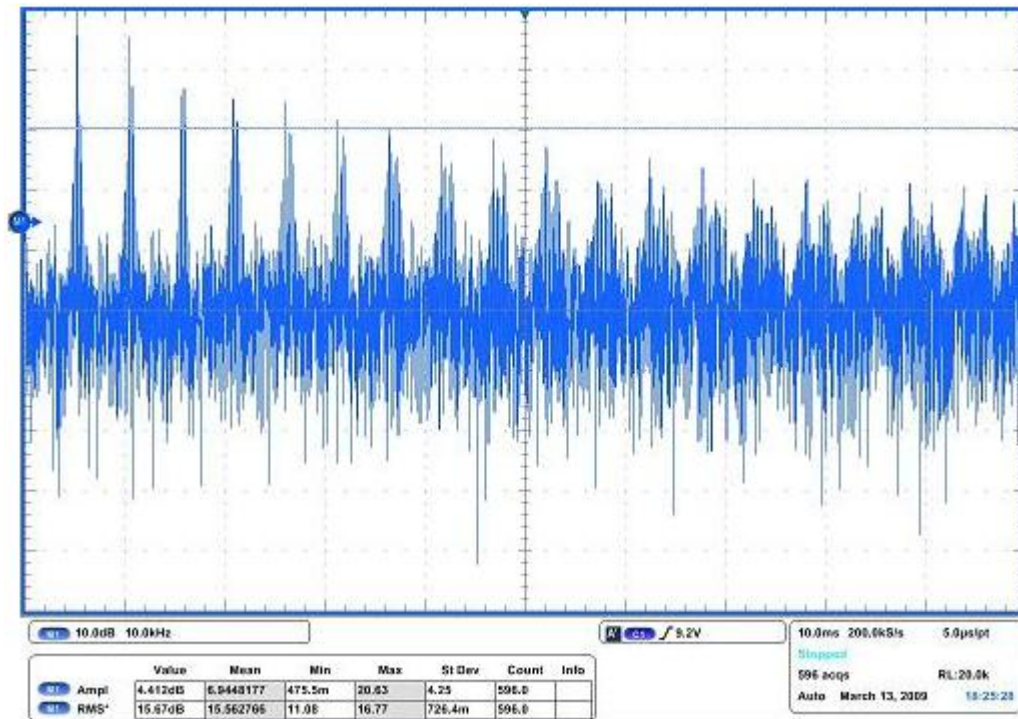


Figura 4.23. Espectro de frecuencia del voltaje de línea V_{AB} con índice de modulación de 0.2 y carga resistiva $R = 50 \Omega$.

En la figura 4.23 se observa que la señal tiene una amplitud de 4.412 dB, y un valor RMS de 15.67 dB a una frecuencia de 10 kHz. Se puede observar por que los armónicos se presentan cada frecuencia de conmutación, es decir, cada 5.6 kHz los armónicos se hacen presentes, debido a que es la frecuencia de conmutación a la que trabaja el inversor, por lo tanto, se puede implementar un filtro pasabajos para eliminar armónicos arriba de 5.6 kHz.

Por último se muestra el espectro de frecuencia del voltaje de línea V_{AB} con un índice de modulación de 0.8.

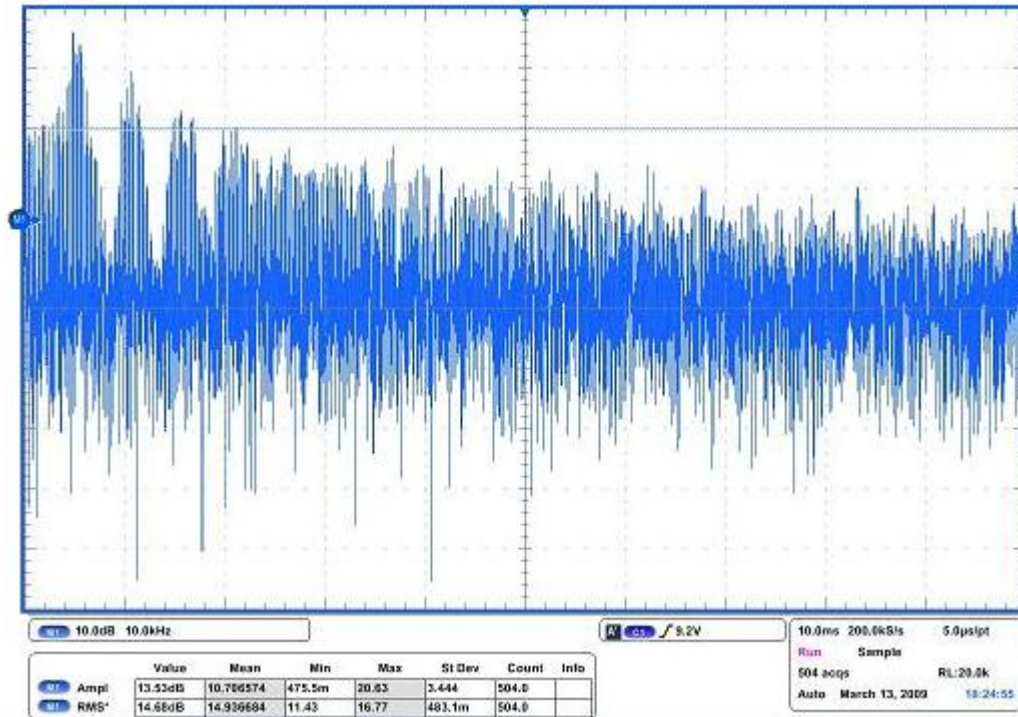


Figura 4.24. Espectro de frecuencia del voltaje de línea V_{AB} con índice de modulación de 0.8 y carga resistiva $R = 50 \Omega$.

En la figura 4.24, la amplitud subió a 13.53 dB y la amplitud RMS permaneció constante. Debido a que la energía efectiva al variar el índice de modulación es la misma, los armónicos se siguen presentando en múltiplos de la frecuencia de conmutación,

4.7 Desbalance en los capacitores

En el caso de los capacitores divisores de tensión de entrada, se había mencionado en el Capítulo II, que existe cierto desbalance de estos debido a los diferentes niveles de tensión y a la conmutación de los transistores. La carga y descarga de los capacitores es rápida, pero ese pequeño desbalance afecta la señal de salida del inversor. En la figura 4.25 se muestra el desbalance del capacitor C_1 con un índice de modulación de 0.2.

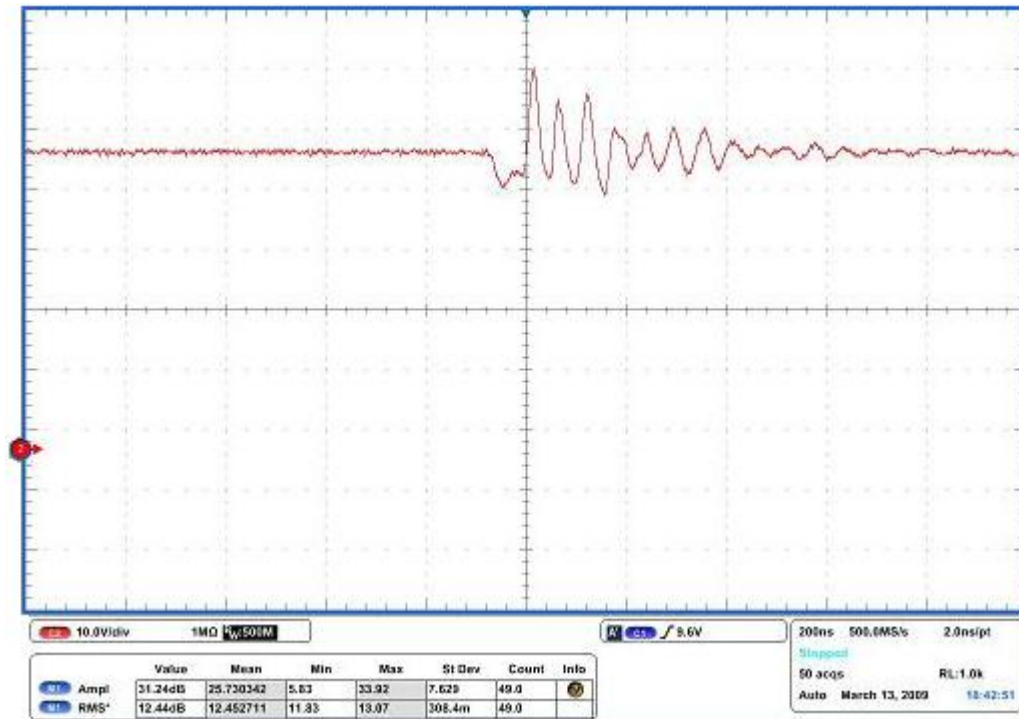


Figura 4.25. Desbalance de voltaje del capacitor C_1 con índice de modulación de 0.2.

En la figura 4.25 se observa que el tiempo de estabilización del capacitor C_1 es de aproximadamente 200 ns, lo cual se considera relativamente rápido, ya que es suficientemente estable para suministrar el voltaje necesario al inversor a la frecuencia de conmutación a la que éste opera. En el caso del inversor de 3 niveles es un tiempo reducido, debido a que sólo cuenta con 2 capacitores de entrada, en un inversor de más niveles el desbalance puede ser mayor. Se observa también un sobre impulso de voltaje de hasta $40 V_{CD}$.

En la figura 4.26 se muestra el mismo capacitor con un índice de modulación de 0.8.

El resultado en la figura 4.26 es un tiempo mayor de estabilización del capacitor, siendo este de 1 μ s, 200 ns más lento que con índice de modulación de 0.2.

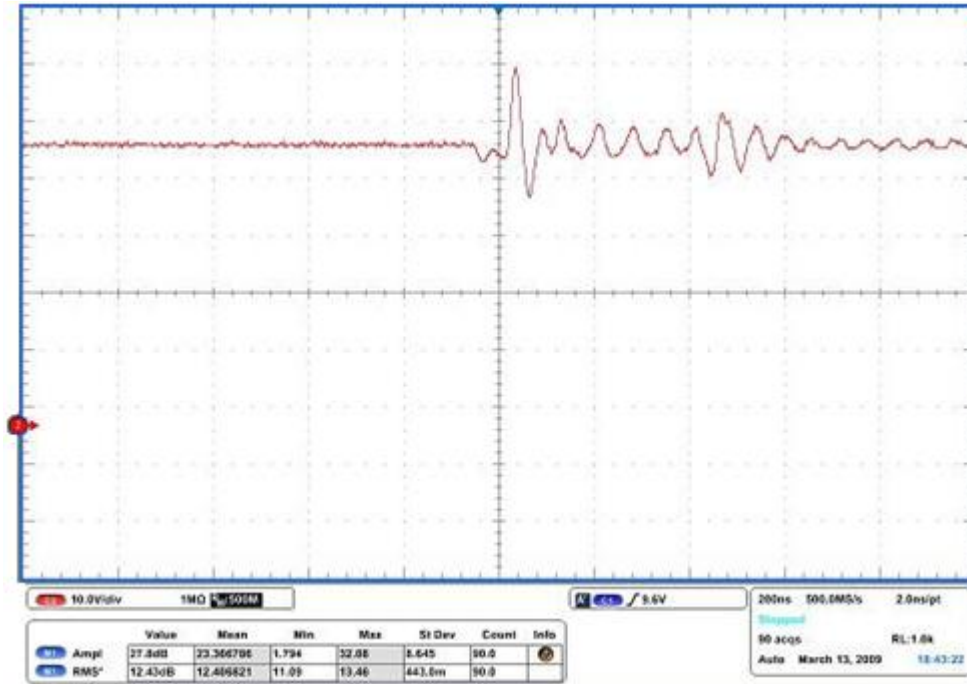


Figura 4.26. Desbalance de voltaje del capacitor C_1 con índice de modulación de 0.8.

Para el caso del capacitor C_2 , con un índice de modulación de 0.2, se muestra la figura 4.27.

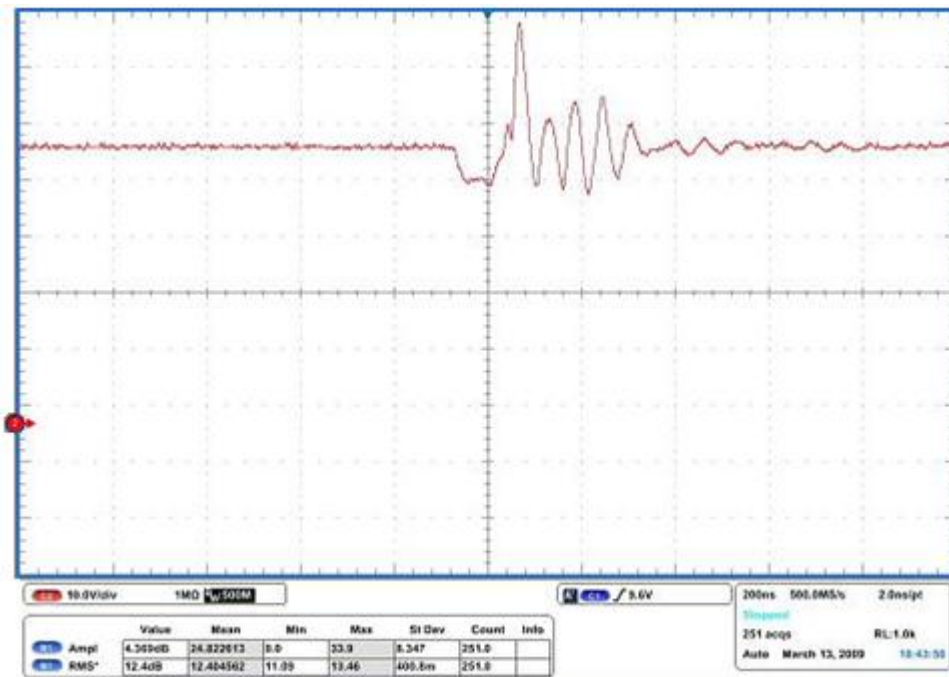


Figura 4.27. Desbalance de voltaje del capacitor C_2 con índice de modulación de 0.2.

En el caso del capacitor C2, el tiempo de estabilización fue menor, siendo este de 600 ns, con un sobre impulso de voltaje de $28 V_{CD}$.

Por último se puede observar en la figura 4.28 el desbalance de voltaje del capacitor C₂ con un índice de modulación de 0.8.

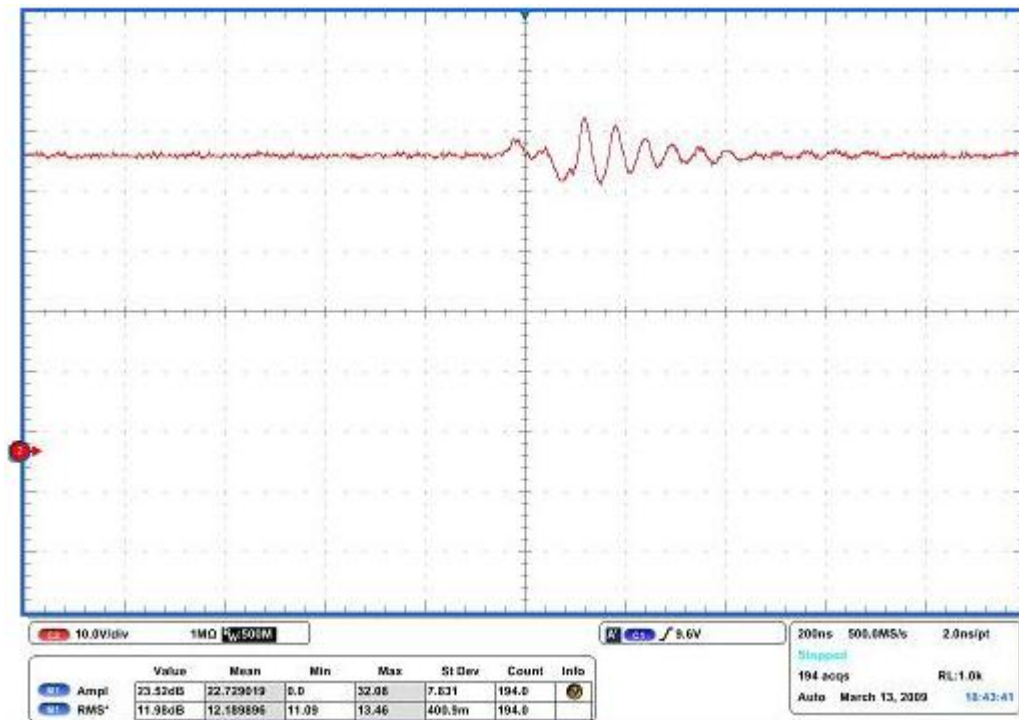


Figura 4.28. Desbalance de voltaje del capacitor C₂ con índice de modulación de 0.8.

En la figura 4.28 se observa un sobre impulso más pequeño, de sólo $33 V_{CD}$, y un tiempo de estabilización de 600 ns.

Hasta aquí se han abordado los 4 capítulos principales de este trabajo, y como punto final se muestran las conclusiones a las que se llegaron, y las recomendaciones para posibles mejoras del mismo.

Referencias

[1] PEÑUELAS Machado José Ángel, "Algoritmo de Seguimiento del Punto de Máxima Potencia para Sistemas Fotovoltaicos en Cd. Obregón, Sonora", Tesis de Ingeniero en Electrónica, Instituto Tecnológico de Sonora, Cd. Obregón, Sonora, Octubre de 2008.

Conclusiones y recomendaciones

En lo que respecta al desarrollo del proyecto, se logró cumplir con los objetivos planteados, de implementar la técnica SVM en un Controlador Digital de Señales, para ser aplicado a un Inversor Trifásico de 3 Niveles con Diodo Anclado al Punto Neutro.

Se logró implementar la placa de circuito impreso, con todos los componentes necesarios para las alimentaciones y la señalización y control del inversor, implementando 4 fuentes aisladas de alimentación Push – Pull para las alimentaciones de los puentes del inversor, así como la implementación de la técnica SVM en el dsPIC30F4011, presentándose también el diagrama de flujo del código, para que éste pueda ser implementado en cualquier tipo de MCU, DSC, o DSP, según sea el caso y las necesidades del diseñador y la aplicación.

Conclusiones y recomendaciones

Se pudieron hacer las mediciones pertinentes de los voltajes de salida del inversor, de fase y de línea con carga resistiva y carga inductiva, haciendo uso de resistencias de 50 ohms para la carga resistiva y un motor de inducción trifásico para carga inductiva, proporcionados por el laboratorio de Electrónica del Instituto Tecnológico de Sonora. Se obtuvieron las formas de onda esperadas, para diferentes índices de modulación, y se observó como varía el ancho del pulso variando la ganancia del inversor aumentando o disminuyendo el voltaje de entrada en el canal analógico del DSC. Además que se pudo medir la forma de onda de corriente, utilizando la punta de corriente existente en el laboratorio del ITSON.

Con base en las gráficas de corriente con carga inductiva, se ve como existe una curva de esfuerzo de corriente, la cual es máxima cuando se empieza a hacer girar el motor, y empieza a decrecer conforme la velocidad del motor aumenta, debido a que disminuye el esfuerzo que este necesita para hacerlo girar.

Como recomendación, una de las principales sería aumentar la frecuencia de conmutación de la técnica SVM, con lo cual se llevarían los armónicos a una frecuencia más alta, y podrían ser filtrados haciendo uso de inductores más pequeños que harán más eficiente y con menos pérdidas una forma de onda senoidal a la salida. Para lograr esto, se pueden seguir diferentes pasos: se puede hacer uno de un DSC o un DSP que sea lo suficientemente rápido para realizar operaciones matemáticas en un tiempo más reducido, o que cuente con una arquitectura para manejar operaciones en punto decimal en un ciclo máquina, ya que los cálculos que se hacen para la técnica SVM hacen uso de punto decimal, lo que obliga al DSC a hacer las operaciones por software, lo cual consume muchos ciclos máquina que tardan tiempo suficiente, por lo que no permite el incremento de la frecuencia de conmutación. Si no se cuenta con DSC que manejen operación en punto decimal por hardware, se puede recurrir a DSC o MCU de 32 bits, para poder aumentar el valor de las operaciones para trabajar con puros números enteros, y que todas las operaciones se realicen por hardware sin usar el punto decimal.

Conclusiones y recomendaciones

Existe otra opción, que es la de agregar una memoria RAM externa al DSC, o trabajar con DSC o DSP que cuenten con memoria RAM interna lo suficientemente grande para almacenar muchos valores de vectores, que puedan ser calculados antes de que el inversor sea encendido, y al momento que se necesite aplicar dichos vectores, ya estén calculados y almacenados en tablas, que al variar el índice de modulación se sigan aplicando los anteriores hasta que los nuevos sean calculados y anexados a otra tabla, lo cual consume mucha memoria RAM, y por lo general los DSC cuentan con memorias internas de 2 MB, por lo cual es necesario conectarlas a una RAM externa.

BIBLIOGRAFÍA

- 1 MUHAMMAD H. Rashid, "Electrónica de Potencia, Circuitos, Dispositivos y Aplicaciones." Prentice Hall, Tercera edición, 2004.
- 2 A. NABAE, I. Takahashi, and H. Akagy, "A Neutral-Point Clamped PWM Inverter." IEEE Transactions on Industrial Applications, 1981.
- 3 MARTÍN Prats M. Ángeles, "Nuevas Técnicas de Modulación Vectorial para Convertidores Electrónicos de Potencia Multinivel" Tesis de Doctor por la Universidad de Sevilla, Sevilla, Junio de 2003.
- 4 PEÑUELAS Machado José Ángel, "Algoritmo de Seguimiento del Punto de Máxima Potencia para Sistemas Fotovoltaicos en Cd. Obregón, Sonora", Tesis de Ingeniero en Electrónica, Instituto Tecnológico de Sonora, Cd. Obregón, Sonora, Octubre de 2008.
- 5 ELGUETA Díaz Cristian Marcelo, "Aplicación de un Inversor Multinivel como Variador de Frecuencia en un Motor de Inducción Trifásico", Tesis de Maestro en Ciencias de la Ingeniería, Pontificia Universidad Católica de Chile, Santiago de Chile, Julio de 2005.
- 6 MARTINEZ Hernández Domitilo, "Control Digital Para Convertidor Multinivel Alimentado con Energía Solar", Tesis de Maestro en Ciencias con especialidad en Ingeniería Electrónica, Universidad de las Américas Puebla, Cholula, Puebla, Mayo de 2006.
- 7 T. A. Meynard, H. Foch, "Multi-Level Conversion: High Voltage Choppers and Voltage Source Inverters", IEEE Power Electronics Specialists Conference Vol. 1, 1992.

Bibliografía

8 RODRIGUEZ J., Jih-Sheng Lai, Fang Zheng Peng, "Multilevel Inverters: a survey of topologies, controls and applications", Industrial Electronics, IEEE Transactions on Volume 49, Issue 4, Agosto de 2002.

9 CAMPOS Valenzuela Rodrigo Andrés, "Evaluación Técnica y Económica de Tres tipos de Inversores de Tecnología Multinivel para Aplicaciones en Media Tensión", Tesis de Ingeniero Civil, Pontificia Universidad Católica de Valparaíso-Chile, Chile, Noviembre de 2007.

10 IRIBE Quintero Víctor, "Técnica de Modulación por Vectores Espaciales Implementada en un dsPIC30F3010", Tesis de Ingeniero en Electrónica, Instituto Tecnológico de Sonora, Cd. Obregón, Sonora, Diciembre de 2007.

11 MICROCHIP Technology Inc., "MPLAB IDE User`s Guide", 2006, (Página Web), <http://ww1.microchip.com/downloads/en/DeviceDoc/51519a.pdf>, consultada en Noviembre 2008.

12 DOMITSU Kono, Manuel, "Programación en el lenguaje C", (Página Web), http://www.itson.mx/die/mdomitsu/bibliotecaDigital/Programacion_Lenguaje_CI/, consultada en Noviembre 2008.

13 MICROCHIP Technology Inc., "MPLAB C30 C Compiler User`s Guide", 2007, (Página Web), http://ww1.microchip.com/downloads/en/DeviceDoc/C30_Users_Guide_51284e.pdf, consultada en agosto 2008.

14 D. Grahame Holmes, Thomas A. Lipo, "Pulse Width Modulation For Power Converters, Principles And Practice", IEEE Press Series on Power Engineering, 2003.

15 J. Fernando Silva, N. Rodrigues, J. Costa, "Space Vector Alpha-Beta Sliding mode Current Controllers for Three-Phase Multilevel Inverters" IEEE Transactions on Industrial Electronics, Lisboa, Portugal, 2000.

APÉNDICE

**Código fuente del programa para la
implementación de la técnica SVM**

APÉNDICE

```

#include <p30f4011.h>

void configuracion_inicial(void);
void puertos(void);
void ADC(void);
void Timers(void);
void variables_bits(void);
void SVM(void);
void magnitud(void);
unsigned int ADCRead(void);
void indice_modulacion(void);
void posicion(void);
void calcula_sector(void);
void calcula_triangulo(void);
void duty_cicle(void);
void periodos(void);
void calcula_vectores(void);

unsigned short wt, pasa[6], triangulo, triangulox, sector, procede, normalizar;
unsigned short periodo[3], vector[3];
float ValorADC, xpos, ypos, m, d1, d2, d3, res0, res1, res2;

const float sin[90] =
{0.0,0.0698,0.1392,0.2079,0.2756,0.342,0.4067,0.4695,0.5299,0.5878,0.6428,0.
6947,0.7431,0.788,0.829,0.866,0.8988,0.9272,0.9511,0.9703,0.9848,0.9945,0.
9994,
0.9994,0.9945,0.9848,0.9703,0.9511,0.9272,0.8988,0.866,0.829,
0.788,0.7431,0.6947,0.6428,0.5878,0.5299,0.4695,0.4067,0.342,0.2756,0.207
9,0.1392,0.0698,
0,-0.0698,-0.1392,-0.2079,-0.2756,-0.342,-0.4067,-0.4695,-
0.5299,-0.5878,-0.6428,-0.6947,-0.7431,-0.788,-0.829,-0.866,-0.8988,-0.9272,-
0.9511,-0.9703,-0.9848,-0.9945,-0.9994,
-0.9994,-0.9945,-0.9848,-0.9703,-0.9511,-0.9272,-0.8988,-
0.866,-0.829,-0.788,-0.7431,-0.6947,-0.6428,-0.5878,-0.5299,-0.4695,-0.4067,-
0.342,-0.2756,-0.2079,-0.1392,-0.0698,0);

const float cos[90] =
{1,0.9976,0.9903,0.9781,0.9613,0.9397,0.9135,0.8829,0.848,0.809,0.766,0.71
93,0.6691,0.6157,0.5592,0.5,0.4384,0.3746,0.309,0.2419,0.1736,0.1045,0.034
9,
-0.0349,-0.1045,-0.1736,-0.2419,-0.309,-0.3746,-0.4384,-0.5,-
0.5592,-0.6157,-0.6691,-0.7193,-0.766,-0.809,-0.848,-0.8829,-0.9135,-0.9397,-
0.9613,-0.9781,-0.9903,-0.9976,
-1,-0.9976,-0.9903,-0.9781,-0.9613,-0.9397,-0.9135,-0.8829,-
0.848,-0.809,-0.766,-0.7193,-0.6691,-0.6157,-0.5592,-0.5,-0.4384,-0.3746,-
0.309,-0.2419,-0.1736,-0.1045,-0.0349,
0.0349,0.1045,0.1736,0.2419,0.309,0.3746,0.4384,0.5,0.5592,0.
6157,0.6691,0.7193,0.766,0.809,0.848,0.8829,0.9135,0.9397,0.9613,0.9781,0.
9903,0.9976,1);

const unsigned int vectores[27] =
{0x0CCC,0x0CC6,0x0CC3,0x0C63,0x0C66,0x0C6C,0x0C3C,0x0C36,0x0C33,
0x0633,0x0636,0x063C,0x066C,0x0666,
0x0663,0x06C3,0x06C6,0x06CC,0x03CC,0x03C6,0x03C3,0x036
3,0x0366,0x036C,0x033C,0x0336,0x0333};

int main(void)
{
    configuracion_inicial();
    while(1)
    {
        if(procede == 1)
        {
            SVM();
        }
    }
    return 0;
}

void configuracion_inicial(void)
{
    puertos();
    ADC();
    Timers();
    variables_bits();
    return;
}

void puertos(void)
{
    Calcula();
    ADC_Lectura();
}

void ADC(void)
{
    TRISE = 0;
    TRISF = 0;
    return;
}

void ADCRead(void)
{
    ADCON1 = 0x00E4;
    ADCON2 = 0x0104;
    ADCON3 = 0x0287;
    ADCHS = 0x0008;
    ADPCFG = 0xFEFF;
    ADCSSL = 0x0000;
    return;
}

void Timers(void)
{
    T1CON = 0x0004;
    T2CON = 0x0004;
    T3CON = 0x0004;
    T4CON = 0x0004;
    PR1 = 5556;
    IEC1 = 0x0020;
    IEC0 = 0x00C8;
    return;
}

void variables_bits(void)
{
    IFS0 = 0;

    wt = 0;
    ADCON1bits.ADON = 1;
    T1CONbits.TON = 1;
    procede = 1;
    return;
}

void magnitud(void)
{
    ValorADC = ADCRead();
    if(ValorADC > 52)
    {
        ValorADC = 52;
    }
    if(ValorADC < 12)
    {
        ValorADC = 12;
    }
    return;
}

unsigned int ADCRead(void)
{
    while(IFS0bits.ADIF == 0)
    {
        asm("nop");
    }
    IFS0bits.ADIF = 0;
    return ADCBUF0>>4;
}

void indice_modulacion(void)
{
    m = (ValorADC / 64);
    return;
}

void SVM(void)
{
    magnitud();
    indice_modulacion();
    posicion();
    calcula_sector();
    calcula_triangulo();
    duty_cicle();
    calcula_vectores();
    return;
}

void posicion(void)
{
    xpos = (ValorADC) * (cos[wt>>2]);
    ypos = (ValorADC) * (sin[wt>>2]);

    res0 = ypos + 1.732*xpos;
    if(res0 > 64)
    {
        pasa[0] = 1;
    }
    else
}

```

APÉNDICE

```

    {
        pasa[0] = 0;
    }
    res1 = ypos - 1.732*xpos;
    if(res1 > 64)
    {
        pasa[1] = 1;
    }
    else
    {
        pasa[1] = 0;
    }
    res2 = ypos;
    if(res2 < -32)
    {
        pasa[2] = 1;
    }
    else
    {
        pasa[2] = 0;
    }
    if(res2 > 32)
    {
        pasa[3] = 1;
    }
    else
    {
        pasa[3] = 0;
    }
    if(res0 < -64)
    {
        pasa[4] = 1;
    }
    else
    {
        pasa[4] = 0;
    }
    if(res1 < -64)
    {
        pasa[5] = 1;
    }
    else
    {
        pasa[5] = 0;
    }
    return;
}

void calcula_sector(void)
{
    if(wt >= 0 && wt <= 60)
    {
        sector = 0;
    }
    else if(wt > 60 && wt <= 120)
    {
        sector = 1;
    }
    else if(wt > 120 && wt <= 180)
    {
        sector = 2;
    }
    else if(wt > 180 && wt <= 240)
    {
        sector = 3;
    }
    else if(wt > 240 && wt <= 300)
    {
        sector = 4;
    }
    else if(wt > 300 && wt < 360)
    {
        sector = 5;
    }
    return;
}

void calcula_triangulo(void)
{
    if(pasa[0] == 0 && pasa[1] == 0 && pasa[2] == 0 && pasa[3] == 0
    && pasa[4] == 0 && pasa[5] == 0)
    {
        if(sector == 0)
        {
            triangulo = 11;
            triangulox = 1;
        }
        else if(sector == 1)
        {
            triangulo = 21;
            triangulox = 1;
        }
        else if(sector == 2)
        {
            triangulo = 31;
            triangulox = 1;
        }
        else if(sector == 3)
        {
            triangulo = 41;
            triangulox = 1;
        }
        else if(sector == 4)
        {
            triangulo = 51;
            triangulox = 1;
        }
        else if(sector == 5)
        {
            triangulo = 61;
            triangulox = 1;
        }
    }
    else if(pasa[0] == 1 && pasa[5] == 0 && pasa[3] == 0)
    {
        triangulo = 12;
        triangulox = 2;
    }
    else if(pasa[0] == 1 && pasa[5] == 1 && sector == 0)
    {
        triangulo = 13;
        triangulox = 3;
    }
    else if(pasa[0] == 1 && pasa[3] == 1 && sector == 0)
    {
        triangulo = 14;
        triangulox = 4;
    }
    else if(pasa[3] == 1 && pasa[0] == 0 && pasa[1] == 0)
    {
        triangulo = 22;
        triangulox = 2;
    }
    else if(pasa[0] == 1 && pasa[3] == 1 && sector == 1)
    {
        triangulo = 23;
        triangulox = 3;
    }
    else if(pasa[3] == 1 && pasa[1] == 1 && sector == 1)
    {
        triangulo = 24;
        triangulox = 4;
    }
    else if(pasa[3] == 0 && pasa[1] == 1 && pasa[4] == 0)
    {
        triangulo = 32;
        triangulox = 2;
    }
    else if(pasa[1] == 1 && pasa[3] == 1 && sector == 2)
    {
        triangulo = 33;
        triangulox = 3;
    }
    else if(pasa[1] == 1 && pasa[4] == 1 && sector == 2)
    {
        triangulo = 34;
        triangulox = 4;
    }
    else if(pasa[4] == 1 && pasa[1] == 0 && pasa[2] == 0)
    {
        triangulo = 42;
        triangulox = 2;
    }
    else if(pasa[4] == 1 && pasa[1] == 1 && sector == 3)
    {
        triangulo = 43;
        triangulox = 3;
    }
    else if(pasa[2] == 1 && pasa[4] == 1 && sector == 3)
    {
        triangulo = 44;
        triangulox = 4;
    }
    else if(pasa[2] == 1 && pasa[4] == 0 && pasa[5] == 0)
    {
        triangulo = 52;
        triangulox = 2;
    }
    else if(pasa[2] == 1 && pasa[4] == 1 && sector == 4)
    {
        triangulo = 53;
    }
}

```

APÉNDICE

```

        triangulox = 3;
    }
    else if(pasa[2] == 1 && pasa[5] == 1 && sector == 4)
    {
        triangulo = 54;
        triangulox = 4;
    }
    else if(pasa[5] == 1 && pasa[2] == 0 && pasa[0] == 0)
    {
        triangulo = 62;
        triangulox = 2;
    }
    else if(pasa[2] == 1 && pasa[5] == 1 && sector == 5)
    {
        triangulo = 63;
        triangulox = 3;
    }
    else if(pasa[5] == 1 && pasa[0] == 1 && sector == 5)
    {
        triangulo = 64;
        triangulox = 4;
    }
    }
    return;
}

void duty_cicle(void)
{
    normalizar = (sector*60);
    if(triangulox == 1)
    {
        d1 = m * (1.732 * cos[(wt - (normalizar))>>2] -
sin[(wt - (normalizar))>>2]);
        d2 = 2 * m * sin[(wt - (normalizar))>>2];
        d3 = (1 - d1 - d2);
    }
    else if(triangulox == 2)
    {
        d1 = 1 - (2 * m * sin[(wt - (normalizar))>>2]);
        d2 = 1 + (m * (sin[(wt - (normalizar))>>2] - (1.732 *
cos[(wt - (normalizar))>>2])));
        d3 = (1 - d1 - d2);
    }
    else if(triangulox == 3)
    {
        d1 = (m * (1.732 * cos[(wt - (normalizar))>>2] -
sin[(wt - (normalizar))>>2])) - 1;
        d2 = 2 * m * sin[(wt - (normalizar))>>2];
        d3 = (1 - d1 - d2);
    }
    else if(triangulox == 4)
    {
        d1 = (m * (1.732 * cos[(60 - (wt - (normalizar))>>2] -
sin[(60 - (wt - (normalizar))>>2])) - 1;
        d2 = 2 * m * sin[(60 - (wt - (normalizar))>>2];
        d3 = (1 - d1 - d2);
    }
    wt = wt + 4;
    if(wt == 360)
    {
        wt = 0;
    }
    periodos();
    procede = 0;
    return;
}

void periodos(void)
{
    periodo[0] = (d1 * 5556);
    periodo[1] = (d2 * 5556);
    periodo[2] = (d3 * 5556);
    return;
}

void calcula_vectores(void)
{
    if(triangulo == 11)
    {
        vector[0] = vectores[4];
        vector[1] = vectores[14];
        vector[2] = vectores[13];
    }
    else if(triangulo == 12)
    {
        vector[0] = vectores[4];
        vector[1] = vectores[14];
        vector[2] = vectores[3];
    }
    else if(triangulo == 13)
    {
        vector[0] = vectores[8];
        vector[1] = vectores[3];
        vector[2] = vectores[4];
    }
    else if(triangulo == 14)
    {
        vector[0] = vectores[2];
        vector[1] = vectores[3];
        vector[2] = vectores[14];
    }
    else if(triangulo == 21)
    {
        vector[0] = vectores[14];
        vector[1] = vectores[16];
        vector[2] = vectores[13];
    }
    else if(triangulo == 22)
    {
        vector[0] = vectores[14];
        vector[1] = vectores[16];
        vector[2] = vectores[15];
    }
    else if(triangulo == 23)
    {
        vector[0] = vectores[2];
        vector[1] = vectores[15];
        vector[2] = vectores[14];
    }
    else if(triangulo == 24)
    {
        vector[0] = vectores[20];
        vector[1] = vectores[15];
        vector[2] = vectores[16];
    }
    else if(triangulo == 31)
    {
        vector[0] = vectores[16];
        vector[1] = vectores[22];
        vector[2] = vectores[13];
    }
    else if(triangulo == 32)
    {
        vector[0] = vectores[16];
        vector[1] = vectores[22];
        vector[2] = vectores[19];
    }
    else if(triangulo == 33)
    {
        vector[0] = vectores[20];
        vector[1] = vectores[19];
        vector[2] = vectores[16];
    }
    else if(triangulo == 34)
    {
        vector[0] = vectores[18];
        vector[1] = vectores[19];
        vector[2] = vectores[22];
    }
    else if(triangulo == 41)
    {
        vector[0] = vectores[22];
        vector[1] = vectores[12];
        vector[2] = vectores[13];
    }
    else if(triangulo == 42)
    {
        vector[0] = vectores[22];
        vector[1] = vectores[12];
        vector[2] = vectores[23];
    }
    else if(triangulo == 43)
    {
        vector[0] = vectores[18];
        vector[1] = vectores[23];
        vector[2] = vectores[22];
    }
    else if(triangulo == 44)
    {
        vector[0] = vectores[24];
        vector[1] = vectores[23];
        vector[2] = vectores[12];
    }
    else if(triangulo == 51)
    {
        vector[0] = vectores[12];
        vector[1] = vectores[10];
        vector[2] = vectores[13];
    }
    else if(triangulo == 52)
    {
        vector[0] = vectores[12];
        vector[1] = vectores[10];
        vector[2] = vectores[11];
    }
    else if(triangulo == 53)

```

APÉNDICE

```

    {
        vector[0] = vectores[24];
        vector[1] = vectores[11];
        vector[2] = vectores[12];
    }
    else if(triangulo == 54)
    {
        vector[0] = vectores[6];
        vector[1] = vectores[11];
        vector[2] = vectores[10];
    }
    else if(triangulo == 61)
    {
        vector[0] = vectores[10];
        vector[1] = vectores[4];
        vector[2] = vectores[13];
    }
    else if(triangulo == 62)
    {
        vector[0] = vectores[10];
        vector[1] = vectores[4];
        vector[2] = vectores[7];
    }
    else if(triangulo == 63)
    {
        vector[0] = vectores[6];
        vector[1] = vectores[7];
        vector[2] = vectores[10];
    }
    else if(triangulo == 64)
    {
        vector[0] = vectores[8];
        vector[1] = vectores[7];
        vector[2] = vectores[4];
    }
    }
    return;
}

void __attribute__((interrupt, auto_psv)) _T1Interrupt(void)
{
    IFS0bits.T1IF = 0;
    T1CONbits.TON = 0;
    TMR1 = 0;

    PR2 = periodo[0];
    PR3 = periodo[1];
    PR4 = periodo[2];

    procede = 1;

    if(PR2 != 0)
    {
        LATE = vector[0]>>6;
        LATF = vector[0];
    }
    else
    {
        IFS0bits.T2IF = 1;
    }
    T2CONbits.TON = 1;
    return;
}

void __attribute__((interrupt, auto_psv)) _T2Interrupt(void)
{
    IFS0bits.T2IF = 0;
    T2CONbits.TON = 0;
    TMR2 = 0;

    if(PR3 != 0)
    {
        LATE = vector[1]>>6;
        LATF = vector[1];
    }
    else
    {
        IFS0bits.T3IF = 1;
    }
    T3CONbits.TON = 1;
    return;
}

void __attribute__((interrupt, auto_psv)) _T3Interrupt(void)
{
    IFS0bits.T3IF = 0;
    T3CONbits.TON = 0;
    TMR3 = 0;

    if(PR4 != 0)
    {
        LATE = vector[2]>>6;
        LATF = vector[2];
    }
}

    else
    {
        IFS1bits.T4IF = 1;
    }
    T4CONbits.TON = 1;
    return;
}

void __attribute__((interrupt, auto_psv)) _T4Interrupt(void)
{
    IFS1bits.T4IF = 0;
    T4CONbits.TON = 0;
    TMR4 = 0;

    PR2 = periodo[0];
    PR3 = periodo[1];
    PR4 = periodo[2];

    procede = 1;

    if(PR2 != 0)
    {
        LATE = vector[0]>>6;
        LATF = vector[0];
    }
    else
    {
        IFS0bits.T2IF = 1;
    }
    T2CONbits.TON = 1;
    return;
}

```

ANEXO 1

Hoja de datos del controlador digital de señales dsPIC30F4011



dsPIC30F4011/4012
Data Sheet

High-Performance, 16-Bit
Digital Signal Controllers



dsPIC30F4011/4012

dsPIC30F4011/4012 Enhanced Flash 16-Bit Digital Signal Controller

Note: This data sheet summarizes features of this group of dsPIC30F devices and is not intended to be a complete reference source. For more information on the CPU, peripherals, register descriptions and general device functionality, refer to the "dsPIC30F Family Reference Manual" (DS70046). For more information on the device instruction set and programming, refer to the "dsPIC30F/33F Programmer's Reference Manual" (DS70157).

High-Performance, Modified RISC CPU:

- Modified Harvard architecture
- C compiler optimized instruction set architecture with flexible addressing modes
- 83 base instructions
- 24-bit wide instructions, 16-bit wide data path
- 48 Kbytes on-chip Flash program space (16K instruction words)
- 2 Kbytes of on-chip data RAM
- 1 Kbyte of nonvolatile data EEPROM
- Up to 30 MIPS operation:
 - DC to 40 MHz external clock input
 - 4 MHz-10 MHz oscillator input with PLL active (4x, 8x, 16x)
- 30 interrupt sources:
 - 3 external interrupt sources
 - 8 user-selectable priority levels for each interrupt source
 - 4 processor trap sources
- 16 x 16-bit working register array

DSP Engine Features:

- Dual data fetch
- Accumulator write-back for DSP operations
- Modulo and Bit-Reversed Addressing modes
- Two, 40-bit wide accumulators with optional saturation logic
- 17-bit x 17-bit single-cycle hardware fractional/integer multiplier
- All DSP instructions are single cycle
- ± 16 -bit, single-cycle shift

Peripheral Features:

- High-current sink/source I/O pins: 25 mA/25 mA
- Timer module with programmable prescaler:
 - Five 16-bit timers/counters; optionally pair 16-bit timers into 32-bit timer modules
- 16-bit Capture input functions
- 16-bit Compare/PWM output functions
- 3-wire SPI modules (supports 4 Frame modes)
- I²C™ module supports Multi-Master/Slave mode and 7-bit/10-bit addressing
- 2 UART modules with FIFO Buffers
- 1 CAN module, 2.0B compliant

Motor Control PWM Module Features:

- 6 PWM output channels:
 - Complementary or Independent Output modes
 - Edge and Center-Aligned modes
- 3 duty cycle generators
- Dedicated time base
- Programmable output polarity
- Dead-time control for Complementary mode
- Manual output control
- Trigger for A/D conversions

Quadrature Encoder Interface Module Features:

- Phase A, Phase B and Index Pulse input
- 16-bit up/down position counter
- Count direction status
- Position Measurement (x2 and x4) mode
- Programmable digital noise filters on inputs
- Alternate 16-Bit Timer/Counter mode
- Interrupt on position counter rollover/underflow

dsPIC30F4011/4012

Analog Features:

- 10-Bit Analog-to-Digital Converter (A/D) with 4 S/H inputs:
 - 1 Msp/s conversion rate
 - 9 input channels
 - Conversion available during Sleep and Idle
- Programmable Brown-out Reset

Special Digital Signal Controller Features:

- Enhanced Flash program memory:
 - 10,000 erase/write cycle (min.) for industrial temperature range, 100K (typical)
- Data EEPROM memory:
 - 100,000 erase/write cycle (min.) for industrial temperature range, 1M (typical)
- Self-reprogrammable under software control
- Power-on Reset (POR), Power-up Timer (PWRT) and Oscillator Start-up Timer (OST)

Special Digital Signal Controller Features (Cont.):

- Flexible Watchdog Timer (WDT) with on-chip, low-power RC oscillator for reliable operation
- Fail-Safe Clock Monitor operation detects clock failure and switches to on-chip, low-power RC oscillator
- Programmable code protection
- In-Circuit Serial Programming™ (ICSP™)
- Selectable Power Management modes:
 - Sleep, Idle and Alternate Clock modes

CMOS Technology:

- Low-power, high-speed Flash technology
- Wide operating voltage range (2.5V to 5.5V)
- Industrial and Extended temperature ranges
- Low-power consumption

dsPIC30F Motor Control and Power Conversion Family*

Device	Pins	Program Mem. Bytes/Instructions	SRAM Bytes	EEPROM Bytes	Timer 16-bit	Input Cap	Output Comp/Std PWM	Motor Control PWM	10-Bit A/D 1 Msp/s	Quad Enc	UART	SPI	I ² C™	CAN
dsPIC30F2010	28	12K/4K	512	1024	3	4	2	6 ch	6 ch	Yes	1	1	1	-
dsPIC30F3010	28	24K/8K	1024	1024	5	4	2	6 ch	6 ch	Yes	1	1	1	-
dsPIC30F4012	28	48K/16K	2048	1024	5	4	2	6 ch	6 ch	Yes	1	1	1	1
dsPIC30F3011	40/44	24K/8K	1024	1024	5	4	4	6 ch	9 ch	Yes	2	1	1	-
dsPIC30F4011	40/44	48K/16K	2048	1024	5	4	4	6 ch	9 ch	Yes	2	1	1	1
dsPIC30F5015	64	66K/22K	2048	1024	5	4	4	8 ch	16 ch	Yes	1	2	1	1
dsPIC30F6010	80	144K/48K	8192	4096	5	8	8	8 ch	16 ch	Yes	2	2	1	2

* This table provides a summary of the dsPIC30F6010 peripheral features. Other available devices in the dsPIC30F Motor Control and Power Conversion Family are shown for feature comparison.

dsPIC30F4011/4012

Pin Diagrams

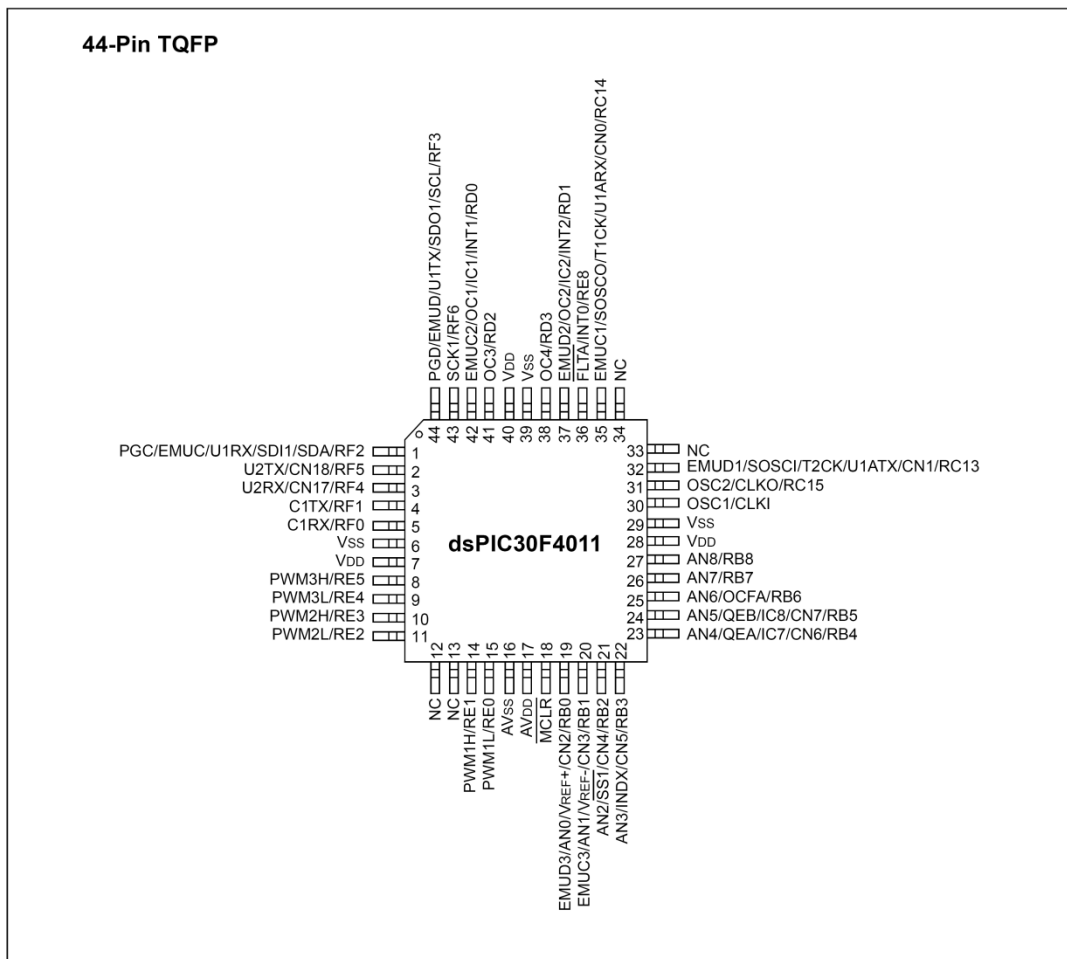
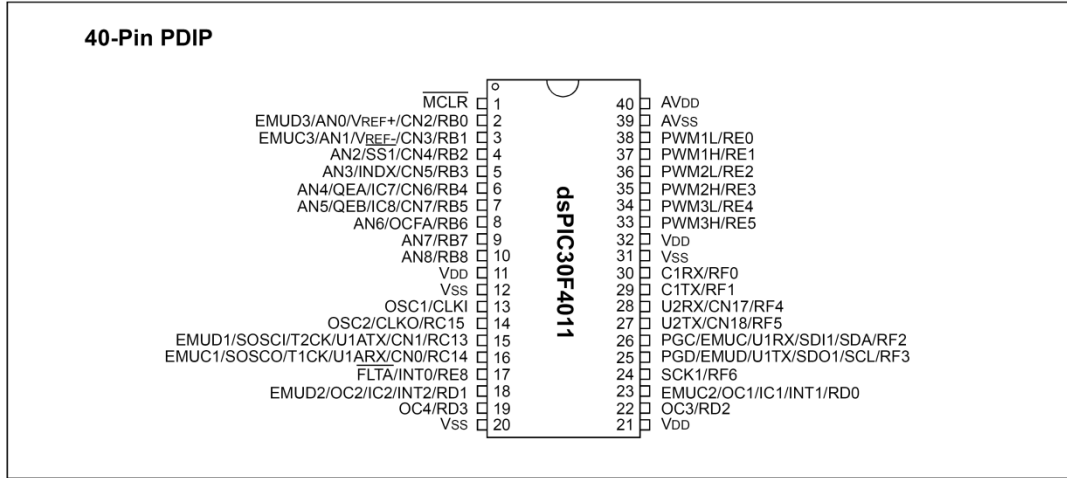
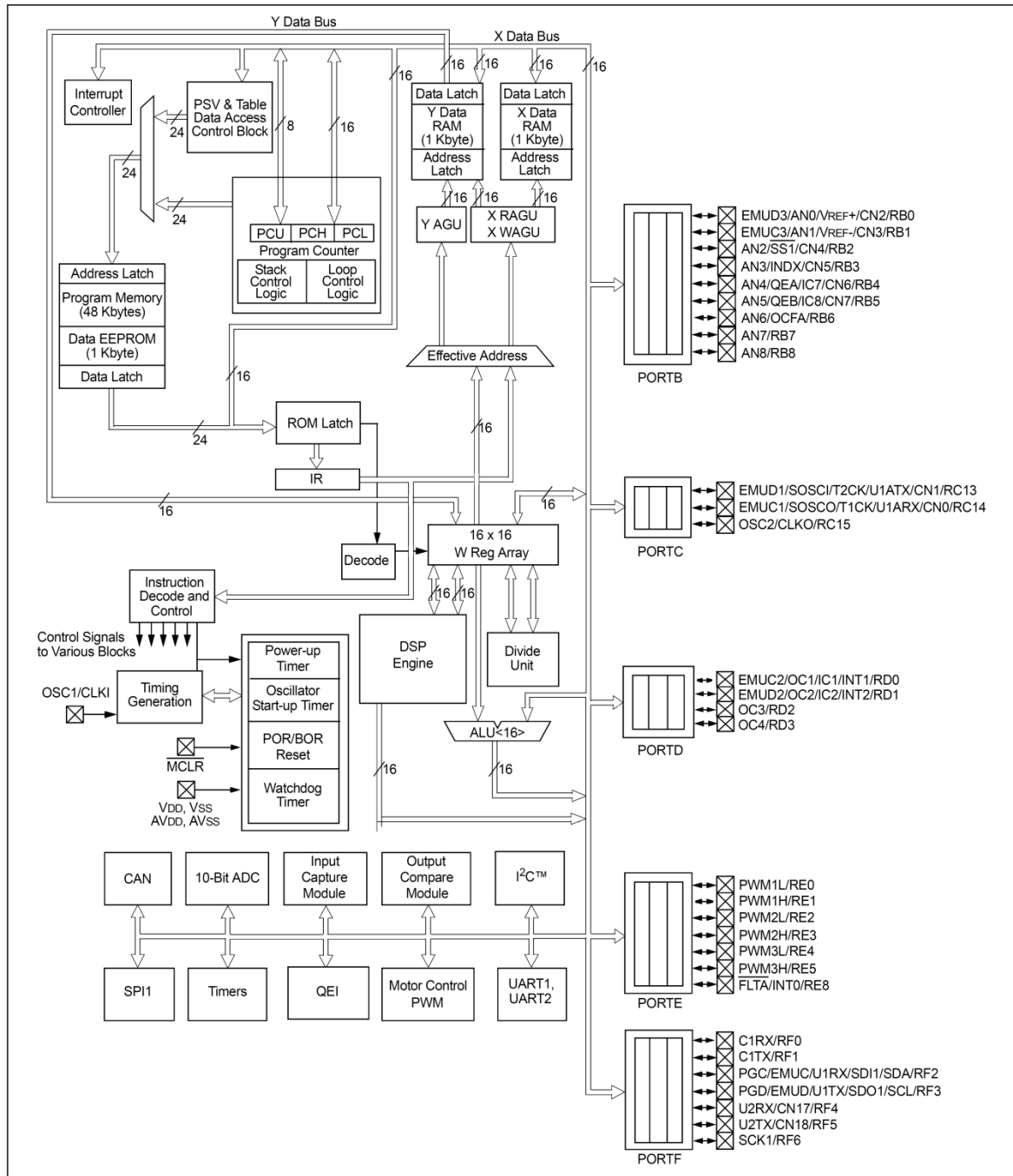


FIGURE 1-1: dsPIC30F4011 BLOCK DIAGRAM



ANEXO 2

Hoja de datos del Transistor MOSFET STW25NM60N



STB25NM60Nx - STF25NM60N STP25NM60N - STW25NM60N

N-channel 600 V, 0.130 Ω , 21 A, MDmesh™ II Power MOSFET
TO-220, TO-220FP, I²PAK, D²PAK, TO-247

Features

Type	V _{DSS} (@T _{jmax})	R _{DS(on)} max	I _D
STB25NM60N	650 V	< 0.160 Ω	21 A
STB25NM60N-1	650 V	< 0.160 Ω	21 A
STF25NM60N	650 V	< 0.160 Ω	21 A ⁽¹⁾
STP25NM60N	650 V	< 0.160 Ω	21 A
STW25NM60N	650 V	< 0.160 Ω	21 A

1. Limited only by maximum temperature allowed

- 100% avalanche tested
- Low input capacitance and gate charge
- Low gate input resistance

Application

- Switching applications

Description

This series of devices is realized with the second generation of MDmesh™ technology. This revolutionary MOSFET associates a new vertical structure to the company's strip layout to yield one of the world's lowest on-resistance and gate charge. It is therefore suitable for the most demanding high efficiency converters

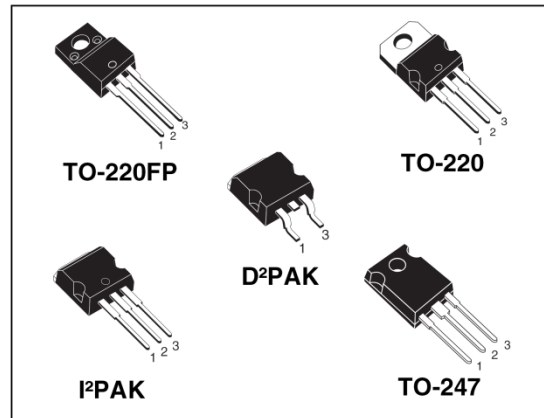


Figure 1. Internal schematic diagram

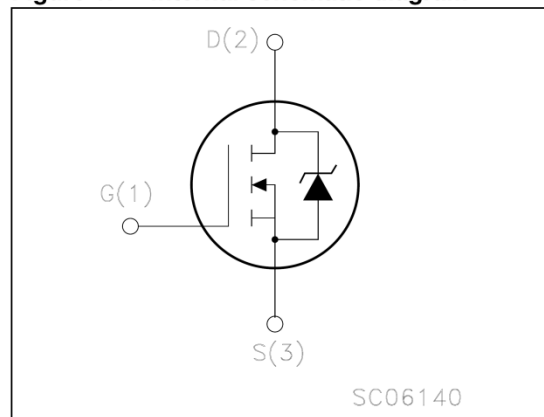


Table 1. Device summary

Order codes	Marking	Package	Packaging
STB25NM60N	B25NM60N	D ² PAK	Tape and reel
STB25NM60N-1	B25NM60N	I ² PAK	Tube
STF25NM60N	F25NM60N	TO-220FP	Tube
STP25NM60N	P25NM60N	TO-220	Tube
STW25NM60N	W25NM60N	TO-247	Tube

1 Electrical ratings

Table 2. Absolute maximum ratings

Symbol	Parameter	Value		Unit
		TO-220 - I ² PAK D ² PAK - TO-247	TO-220FP	
V _{DS}	Drain-source voltage (V _{GS} = 0)	600		V
V _{GS}	Gate- source voltage	±25		V
I _D	Drain current (continuous) at T _C = 25 °C	21	21 ⁽¹⁾	A
I _D	Drain current (continuous) at T _C = 100 °C	13	13 ⁽¹⁾	A
I _{DM} ⁽²⁾	Drain current (pulsed)	84	84 ⁽¹⁾	A
P _{TOT}	Total dissipation at T _C = 25 °C	160	40	W
V _{ISO}	Insulation withstand voltage (RMS) from all three leads to external heat sink (t=1 s; T _C =25 °C)	--	2500	V
dv/dt ⁽³⁾	Peak diode recovery voltage slope	15		V/ns
T _{stg}	Storage temperature	-55 to 150		°C
T _j	Max. operating junction temperature	150		°C

- Limited only by maximum temperature allowed
- Pulse width limited by safe operating area
- I_{SD} ≤ 21 A, di/dt ≤ 400 A/μs, V_{DD} = 80% V_{(BR)DSS}

Table 3. Thermal data

Symbol	Parameter	Value					Unit
		TO-220	I ² PAK	D ² PAK	TO-247	TO-220FP	
R _{thj-case}	Thermal resistance junction-case max	0.78				3.1	°C/W
R _{thj-pcb}	Thermal resistance junction-pcb max	--	--	30	--	--	
R _{thj-amb}	Thermal resistance junction-ambient max	62.5	--	50	62.5	°C/W	
T _l	Maximum lead temperature for soldering purpose	300					°C

Table 4. Avalanche characteristics

Symbol	Parameter	Value	Unit
I _{AR}	Avalanche current, repetitive or not-repetitive (pulse width limited by T _j Max)	10	A
E _{AS}	Single pulse avalanche energy (starting T _J = 25 °C, I _D =I _{AR} , V _{DD} = 50 V)	850	mJ

2 Electrical characteristics

($T_{CASE} = 25\text{ °C}$ unless otherwise specified)

Table 5. On/off states

Symbol	Parameter	Test conditions	Min.	Typ.	Max.	Unit
$V_{(BR)DSS}$	Drain-source breakdown voltage	$I_D = 1\text{ mA}$, $V_{GS} = 0$	600			V
$dv/dt^{(1)}$	Drain source voltage slope	$V_{DD} = 480\text{ V}$, $I_D = 21\text{ A}$, $V_{GS} = 10\text{ V}$	48			V/ns
I_{DSS}	Zero gate voltage drain current ($V_{GS} = 0$)	$V_{DS} = \text{Max rating}$ $V_{DS} = \text{Max rating}$, @ 125 °C			1 100	μA μA
I_{GSS}	Gate-body leakage current ($V_{DS} = 0$)	$V_{GS} = \pm 20\text{ V}$			100	nA
$V_{GS(th)}$	Gate threshold voltage	$V_{DS} = V_{GS}$, $I_D = 250\text{ }\mu\text{A}$	2	3	4	V
$R_{DS(on)}$	Static drain-source on resistance	$V_{GS} = 10\text{ V}$, $I_D = 10.5\text{ A}$		0.130	0.160	Ω

1. Characteristic value at turn off on inductive load

Table 6. Dynamic

Symbol	Parameter	Test conditions	Min.	Typ.	Max.	Unit
$g_{fs}^{(1)}$	Forward transconductance	$V_{DS} = 15\text{ V}$, $I_D = 11\text{ A}$		17		S
C_{iss} C_{oss} C_{rss}	Input capacitance Output capacitance Reverse transfer capacitance	$V_{DS} = 50\text{ V}$, $f = 1\text{ MHz}$, $V_{GS} = 0$		2400 200 25		pF pF pF
$C_{oss\text{ eq.}}^{(2)}$	Equivalent output capacitance	$V_{GS} = 0$, $V_{DS} = 0\text{ to }480\text{ V}$		310		pF
Q_g Q_{gs} Q_{gd}	Total gate charge Gate-source charge Gate-drain charge	$V_{DD} = 480\text{ V}$, $I_D = 21\text{ A}$, $V_{GS} = 10\text{ V}$, (see Figure 19)		84 14 44		nC nC nC
R_g	Gate input resistance	$f = 1\text{ MHz}$ Gate DC Bias = 0 Test signal level = 20 mV open drain		1.6		Ω

1. Pulsed: Pulse duration = 300 μs , duty cycle 1.5%

2. $C_{oss\text{ eq.}}$ is defined as a constant equivalent capacitance giving the same charging time as C_{oss} when V_{DS} increases from 0 to 80% V_{DS}

Table 7. Switching times

Symbol	Parameter	Test conditions	Min.	Typ.	Max.	Unit
$t_{d(on)}$	Turn-on delay time	$V_{DD} = 300\text{ V}$, $I_D = 10\text{ A}$ $R_G = 4.7\ \Omega$ $V_{GS} = 10\text{ V}$ (see Figure 18)		24.5		ns
t_r	Rise time			18		ns
$t_{d(off)}$	Turn-off delay time				94	ns
t_f	Fall time				24	ns

Table 8. Source drain diode

Symbol	Parameter	Test conditions	Min	Typ.	Max	Unit
I_{SD}	Source-drain current				21	A
$I_{SDM}^{(1)}$	Source-drain current (pulsed)				84	A
$V_{SD}^{(2)}$	Forward on voltage	$I_{SD} = 21\text{ A}$, $V_{GS} = 0$			1.3	V
t_{rr}	Reverse recovery time	$I_{SD} = 21\text{ A}$, $di/dt = 100\text{ A}/\mu\text{s}$ $V_{DD} = 100\text{ V}$ (see Figure 23)		427		ns
Q_{rr}	Reverse recovery charge				7.2	μC
I_{RRM}	Reverse recovery current				33.6	A
t_{rr}	Reverse recovery time	$I_{SD} = 21\text{ A}$, $di/dt = 100\text{ A}/\mu\text{s}$ $V_{DD} = 100\text{ V}$, $T_j = 150\text{ }^\circ\text{C}$ (see Figure 23)		526		ns
Q_{rr}	Reverse recovery charge				9.1	μC
I_{RRM}	Reverse recovery current				34.5	A

1. Pulse width limited by safe operating area

2. Pulsed: Pulse duration = 300 μs , duty cycle 1.5%