



Universidad de Valladolid



**ESCUELA DE INGENIERÍAS
INDUSTRIALES**

UNIVERSIDAD DE VALLADOLID

ESCUELA DE INGENIERÍAS INDUSTRIALES

Grado en Ingeniería Electrónica Industrial y Automática

CONVERTIDOR CC/CA DE TRES NIVELES

Autor:

García Fernández, Daniel

Tutor:

**Martínez Rodrigo, Fernando
Departamento de Tecnología
Electrónica**

Valladolid, Julio de 2015.



RESUMEN Y PALABRAS CLAVES

En el presente proyecto se ha diseñado e implementado la etapa de potencia y de control en lazo abierto de un inversor de tres niveles. Se han desarrollado dos ideas principales. En primer lugar se han analizado los drivers empleados para configurarlos de forma correcta. Posteriormente se ha utilizado un entorno de programación para generar las 12 señales PWM que controlarán nuestro inversor. Una vez finalizadas estas dos etapas se han implementado en el sistema real y se ha verificado su correcto funcionamiento, pudiéndose observar diferentes señales eléctricas de entrada y salida. Complementariamente se han fabricado unas protecciones mecánicas para poder asegurar el equipo y mantener los diferentes elementos fijos a una estructura.

Palabras claves:

- CC/CA
- Multinivel
- Inversor
- DSP
- Space Vector Modulation (SVM)





ÍNDICE

RESUMEN Y PALABRAS CLAVES	i
ÍNDICE	iii
ÍNDICE DE FIGURAS	vii
ÍNDICE DE TABLAS	xiii
ÍNDICE DE ECUACIONES.....	xv
1. INTRODUCCIÓN	1
1.1. OBJETIVOS	1
1.2. MOTIVACIÓN	1
1.3. ESTRUCTURA DEL PROYECTO.....	2
2. ESTUDIO DE LA TOPOLOGÍA MULTINIVEL EMPLEADA.....	5
2.1. POSIBLES ESTADOS DEL INVERSOR.....	7
2.2. ÁREAS DE OPERACIÓN DEL INVERSOR.....	9
2.3. PÉRDIDAS DE POTENCIA.....	12
3. HARDWARE UTILIZADO.....	15
3.1. DSP TMS320F28069	15
3.1.1. Docking Station.....	18
3.2. TARJETA DE ADAPTACIÓN DE NIVELES	20
3.3. DRIVER DE CONTROL.....	23
3.3.1. CONFIGURACIÓN DEL DRIVER DE CONTROL	24



3.4. DISEÑO DE CABLES DE INTERCONEXIÓN DE LA TARJETA DE ADAPTACIÓN DE NIVELES CON DSP DE CONTROL Y DRIVER DE CONTROL	34
4. ENTORNO DE PROGRAMACIÓN	39
4.1. CONFIGURACIÓN CODE COMPOSER STUDIO Y SIMULINK	39
4.2. CÓDIGO DESARROLLO EN CODE COMPOSER STUDIO.....	53
4.2.1. SIMULACIÓN DEL CÓDIGO	61
4.3. ENTORNO DE PROGRAMACIÓN MATLAB Y SIMULINK.....	64
4.3.1. FUNCIONAMIENTO MODULACIÓN VECTORIAL DE INVERSORES MULTINIVEL	64
4.3.2. CÓDIGO DESARROLLADO	67
4.3.3. SIMULACIÓN DEL CÓDIGO	70
4.3.4. PARÁMETROS CONFIGURABLES DEL CÓDIGO	73
5. MONTAJE DEL EQUIPO	77
6. PRUEBAS REALIZADAS	83
6.1. PRUEBAS DE VALIDACIÓN	83
6.2. PRUEBAS SPACE VECTOR MODULATION (SVM)	90
7. CONCLUSIONES	97
8. BIBLIOGRAFÍA.....	99
9. ANEXOS	101
9.1. ANEXO 1: PROGRAMA main.c CODE COMPOSER STUDIO.....	101
9.2. ANEXO 2: CÓDIGO FUNCIÓN PWM_POLOS_SVM SIMULINK	109
9.3. ANEXO 3: DATASHEET SKYPER 23 PRO R	117



9.4. ANEXO 4: DATASHEET BOARD 1 SKYPER 32 PRO R	118
9.5. ANEXO 5: DATASHEET SF75MLI066T	119
9.6. ANEXO 6: ESQUEMA ELÉCTRICO PLACA POTENCIA	121
9.7. ANEXO 7: ESQUEMA UBICACIÓN DE COMPONENTES PLACA DE POTENCIA	122
9.8. ANEXO 8: DATASHEET RADIADOR EMPLEADO	123
9.9. ANEXO 9: HOJA EXCEL PARA CALCULO DE R_{CE} Y C_{CE}	124





ÍNDICE DE FIGURAS

FIGURA 2.1 – CONVERTIDOR TRIFÁSICO DE TRES NIVELES DIODE-CLAMPED	5
FIGURA 2.2 – TENSIÓN DE SALIDA FASE-NEUTRO EN FUNCIÓN DE LOS POLOS CONECTADOS.....	6
FIGURA 2.3 – ÁREAS DE OPERACIÓN PARA CARGA INDUCTIVA	9
FIGURA 2.4 – ÁREA DE OPERACIÓN 1.....	9
FIGURA 2.5 – CONMUTACIÓN DE LOS POLOS EN EL ÁREA 1 DE OPERACIÓN	10
FIGURA 2.6 – ÁREA DE OPERACIÓN 2.....	10
FIGURA 2.7 – CONMUTACIÓN DE LOS POLOS EN EL ÁREA 2 DE OPERACIÓN	10
FIGURA 2.8 – ÁREA DE OPERACIÓN 3.....	10
FIGURA 2.9 – CONMUTACIÓN DE LOS POLOS EN EL ÁREA 3 DE OPERACIÓN	11
FIGURA 2.10 – ÁREA DE OPERACIÓN 4.....	11
FIGURA 2.11 – CONMUTACIÓN DE LOS POLOS EN EL ÁREA 3 DE OPERACIÓN	11
FIGURA 2.12 – RESUMEN ÁREAS DE OPERACIÓN	12
FIGURA 2.13 – PERDIDAS DE POTENCIA.....	13
FIGURA 2.14 – PERDIDAS DE POTENCIA EN UN EQUIPO COMPLETO	13
FIGURA 3.1 – DIAGRAMA DE BLOQUES FUNCIONAL DEL DSP TMS320F28069	18
FIGURA 3.2 – DOCKING STATION CON LA TARJETA DE CONTROL INSTALADA	19
FIGURA 3.3 – TARJETA DE ADAPTACIÓN DE NIVELES	20
FIGURA 3.4 – ESQUEMA ELÉCTRICO DE LA TARJETA DE ADAPTACIÓN DE NIVELES	21
FIGURA 3.5 – SKYPER 32 PRO R	24
FIGURA 3.6 – BOARD 1 SKYPER 32 PRO R.....	24
FIGURA 3.7 – FORMA DE ONDA DEL VALOR V_{CE}	26
FIGURA 3.8 – HIGH VOLTAGE DIODE	27
FIGURA 3.9 – CALCULO DE LA RESISTENCIA Y DEL CONDENSADOR EN FUNCIÓN DEL VOLTAJE Y DEL TIEMPO.....	28
FIGURA 3.10 – DESACTIVACIÓN DE LA PROTECCIÓN DINÁMICA CONTRA CORTOCIRCUITO PARA T22-T11	29
FIGURA 3.11 – CONEXIONES PARA DESACTIVAR LA PROTECCIÓN DINÁMICA CONTRA CORTOCIRCUITO	29
FIGURA 3.12 – CIRCUITO ELÉCTRICO PARA LAS RESISTENCIAS DE PUERTA.....	30
FIGURA 3.13 – PLACA DRIVER DE POTENCIA.....	33
FIGURA 3.14 – CONSUMO PLACA ADAPTACIÓN DE NIVELES CON DRIVER DE POTENCIA	33
FIGURA 3.15 – INTERCONEXIONADO DE PLACAS DE HARDWARE	34
FIGURA 3.16 – PEGATINA CABLE 1	35
FIGURA 3.17 – EXTREMO 1 DEL CABLE 1	35
FIGURA 3.18 – EXTREMO 2 DEL CABLE 1	35
FIGURA 3.19 – PEGATINA CABLE 3	35
FIGURA 3.20 – EXTREMO 2 DEL CABLE 3	36
FIGURA 3.21 – PEGATINA CABLE 5	36



FIGURA 3.22 – EXTREMO 1 DEL CABLE 5	36
FIGURA 3.23 – EXTREMO 2 DEL CABLE 5	36
FIGURA 3.24 – INTERCONEXIONADO EN LA ETAPA DE PRUEBAS	37
FIGURA 4.1 – CONFIGURACIÓN DSP EN CODE COMPOSER STUDIO.....	40
FIGURA 4.2 – TEST DE CONEXIÓN.....	40
FIGURA 4.3 – CONEXIÓN CON TARJETA DSP (1)	40
FIGURA 4.4 – CONEXIÓN CON TARJETA DSP (2)	41
FIGURA 4.5 – INSTALACIÓN C2000 EN MATLAB (1).....	41
FIGURA 4.6 – INSTALACIÓN C2000 EN MATLAB (2).....	42
FIGURA 4.7 – INSTALACIÓN C2000 EN MATLAB (3).....	42
FIGURA 4.8 – INSTALACIÓN C2000 EN MATLAB (4).....	43
FIGURA 4.9 – INSTALACIÓN C2000 EN MATLAB (5).....	43
FIGURA 4.10 – CONFIGURACIÓN XMAKEFILESETUP (1).....	44
FIGURA 4.11 – CONFIGURACIÓN XMAKEFILESETUP, MAKE UTILITY	45
FIGURA 4.12 – CONFIGURACIÓN XMAKEFILESETUP, COMPILER	45
FIGURA 4.13 – CONFIGURACIÓN XMAKEFILESETUP, LINKER	45
FIGURA 4.14 – CONFIGURACIÓN XMAKEFILESETUP, ARCHIVER.....	45
FIGURA 4.15 – CONFIGURACIÓN XMAKEFILESETUP, PRE-BUILD	46
FIGURA 4.16 – CONFIGURACIÓN XMAKEFILESETUP, POST-BUILD	46
FIGURA 4.17 – CONFIGURACIÓN XMAKEFILESETUP, EXECUTE	46
FIGURA 4.18 – COMPROBACIÓN CONFIGURACIÓN (1)	47
FIGURA 4.19 – COMPROBACIÓN CONFIGURACIÓN (2)	47
FIGURA 4.20 – CONFIGURACIÓN VACÍA	48
FIGURA 4.21 – AÑADIR VARIABLE DE SISTEMA	48
FIGURA 4.22 – CONFIGURACIÓN SIMULINK (1).....	49
FIGURA 4.23 – CONFIGURACIÓN SIMULINK (2).....	49
FIGURA 4.24 – CONFIGURACIÓN SIMULINK (4).....	50
FIGURA 4.25 – CONFIGURACIÓN SIMULINK Y EJECUCIÓN DEL PROGRAMA.....	50
FIGURA 4.26 – EJECUCIÓN CORRECTA DE LA COMPILACIÓN DE PROGRAMA EN SIMULINK.....	51
FIGURA 4.27 – ERROR EN LA EJECUCIÓN DE LA COMPILACIÓN DEL PROGRAMA DE SIMULINK	51
FIGURA 4.28 – PESTAÑA DISASSEMBLY CON INFORMACIÓN	52
FIGURA 4.29 – SELECCIÓN DEL FICHERO .OUT (1).....	53
FIGURA 4.30 – SELECCIÓN DEL FICHERO .OUT (2).....	53
FIGURA 4.31 – SEÑALES TRIANGULARES MODULADORAS Y SEÑALES EPWMX	56
FIGURA 4.32 – ESTRUCTURA DEL DSP DE CONTROL PARA GENERAR LAS SEÑALES EPWMX.....	57
FIGURA 4.33 – ESTRUCTURA DE UN MÓDULO EPWMX.....	58
FIGURA 4.34 – EJEMPLO DE SEÑAL TRIANGULAR	59



FIGURA 4.35 – PROPAGACIÓN DE LA SEÑAL DE SINCRONISMO	59
FIGURA 4.36 – SELECCIÓN DE LA SEÑAL DE SINCRONISMO DE SALIDA	59
FIGURA 4.37 – EJEMPLO DE SEÑAL DE SINCRONISMO	60
FIGURA 4.38 – MÓDULO DEAD-BAND.....	60
FIGURA 4.39 – EFECTO DE LOS TIEMPOS MUERTOS.....	61
FIGURA 4.40 – ESQUEMA DE SIMULACIÓN PSIM	62
FIGURA 4.41 – TENSIONES V_{AN} , V_{BN} Y V_{CN} EN SIMULACIÓN	62
FIGURA 4.42 – TENSIONES V_{BA} Y V_{CA} EN SIMULACIÓN.....	63
FIGURA 4.43 – CAÍDAS DE TENSIÓN EN LOS POLOS DE POTENCIA DE LA RAMA A	63
FIGURA 4.44 – CONMUTACIONES EN EL INVERSOR DE TRES NIVELES.....	64
FIGURA 4.45 – MAPA DE VECTORES EN UN INVERSOR DE TRES NIVELES.....	65
FIGURA 4.46 – DIAGRAMA VECTORIAL PARA UN INVERSOR DE TRES NIVELES.....	65
FIGURA 4.47 – SECTORES DEL DIAGRAMA VECTORIAL PARA UN INVERSOR DE TRES NIVELES.....	66
FIGURA 4.48 – DIAGRAMA DE BLOQUES DE SIMULINK PARA IMPLEMENTAR SVM	67
FIGURA 4.49 – CONFIGURACIÓN GENERAL DE LOS BLOQUES EPWMX.....	68
FIGURA 4.50 – CONFIGURACIÓN EPWMA DE LOS BLOQUES EPWMX	69
FIGURA 4.51 – CONFIGURACIÓN EPWMB DE LOS BLOQUES EPWMX	69
FIGURA 4.52 – CONFIGURACIÓN DEAD-BAND DE LOS BLOQUES EPWMX.....	70
FIGURA 4.53 – COMPORTAMIENTO DEL CÓDIGO.....	70
FIGURA 4.54 – ESQUEMA DE SIMULACIÓN EN SIMULINK	71
FIGURA 4.55 – VOLTAJES DE FASE EN SIMULACIÓN SVM CON $M=0.9$	71
FIGURA 4.56 – VOLTAJES DE LÍNEA EN SIMULACIÓN SVM CON $M=0.9$	72
FIGURA 4.57 – VOLTAJES DE FASE EN SIMULACIÓN SVM CON $M=0.3$	72
FIGURA 4.58 – VOLTAJES DE LÍNEA EN SIMULACIÓN SVM CON $M=0.3$	73
FIGURA 4.59 – PLANO ALPHA-BETA PARA $M<0.5$	73
FIGURA 4.60 – PLANO ALPHA-BETA PARA $M>0.5$	73
FIGURA 4.61 – FRECUENCIA DE TRABAJO DE LA CPU	74
FIGURA 5.1 – PARED LATERAL	77
FIGURA 5.2 – UNIONES PARA DISIPAR CALOR.....	78
FIGURA 5.3 – ESTRUCTURAS DE SUJECCIÓN ENTRE CARAS LATERALES Y SUPERIOR.	78
FIGURA 5.4 – INTERIOR DE LA ESTRUCTURA (1).....	79
FIGURA 5.5 – INTERIOR DE LA ESTRUCTURA (2).....	79
FIGURA 5.6 – VISTA GENERAL DE LA ESTRUCTURA MONTADA.....	80
FIGURA 5.7 – VISTA LATERAL DEL EQUIPO.....	81
FIGURA 6.1 – S1_A Y S22_A	83
FIGURA 6.2 – S1_A Y S11_A	83
FIGURA 6.3 – S1_A Y S2_A	83



FIGURA 6.4 – S1_A Y S1_B	84
FIGURA 6.5 – S1_A Y S22_B	84
FIGURA 6.6 – S1_A Y S11_B	84
FIGURA 6.7 – S1_A Y S2_B	84
FIGURA 6.8 – S1_A Y S1_C	84
FIGURA 6.9 – S1_A Y S22_C	84
FIGURA 6.10 – S1_A Y S22_C	85
FIGURA 6.11 – S1_A Y S2_C	85
FIGURA 6.12 – TENSIONES V_{EB} DE S1_A Y S1_B.....	85
FIGURA 6.13 – V_{AN} Y V_{BN}	86
FIGURA 6.14 – V_{AN} Y V_{CN}	86
FIGURA 6.15 – V_{AN} , V_{CN} Y V_{BC} MATEMÁTICO	86
FIGURA 6.16 – V_{BC} REAL	87
FIGURA 6.17 – V_{BA} Y V_{CA} A 10KHZ.....	87
FIGURA 6.18 – V_{BA} Y V_{CA} A 20KHZ.....	88
FIGURA 6.19 – V_{BA} Y V_{CA} A 1KHZ.....	88
FIGURA 6.20 – V_{BA} Y V_{CA} A 500HZ.....	88
FIGURA 6.21 – V_{CE} EN S1_A	89
FIGURA 6.22 – V_{CE} EN S22_A	89
FIGURA 6.23 – V_{CE} EN S11_A	89
FIGURA 6.24 – V_{CE} EN S2_A	89
FIGURA 6.25 – S1_A Y S22_A EN SVM.....	90
FIGURA 6.26 – S1_A Y S11_A EN SVM.....	90
FIGURA 6.27 – S1_A Y S2_A EN SVM.....	90
FIGURA 6.28 – S1_A Y 1B EN SVM.....	90
FIGURA 6.29 – S1_A Y S22_B EN SVM.....	90
FIGURA 6.30 – S1_A Y S11_B EN SVM.....	91
FIGURA 6.31 – S1_A Y S2_B EN SVM.....	91
FIGURA 6.32 – S1_A Y S1_C EN SVM.....	91
FIGURA 6.33 – S1_A Y S22_C EN SVM.....	91
FIGURA 6.34 – S1_A Y S11_C EN SVM.....	91
FIGURA 6.35 – S1_A Y S2_C EN SVM.....	91
FIGURA 6.36 – TENSIÓN V_{EB} DE S1_A CON SVM	92
FIGURA 6.37 – TENSIÓN V_{CE} EN S1_A CON SVM	92
FIGURA 6.38 – TENSIÓN V_{CE} EN S22_A CON SVM	92
FIGURA 6.39 – TENSIÓN V_{CE} EN S1A CON SVM	92
FIGURA 6.40 – TENSIÓN V_{CE} EN S2_A CON SVM	92



FIGURA 6.41 – V_{AN} Y V_{BN} CON SVM.....	93
FIGURA 6.42 – V_{AN} Y V_{CN} CON SVM.....	93
FIGURA 6.43 – V_{AN} , V_{BN} Y V_{AB} MATEMÁTICA CON SVM	93
FIGURA 6.44 – V_{AN} , V_{BN} Y V_{AC} MATEMÁTICA CON SVM	93
FIGURA 6.45 – V_{AB} EN SVM CON FREQ = 100Hz	93
FIGURA 6.46 – V_{AB} EN SVM CON FREQ = 50 Hz	93
FIGURA 6.47 – V_{AB} EN SVM CON FREQ = 5Hz	94
FIGURA 6.48 – V_{AB} EN SVM CON FREQ = 0.5Hz.....	94
FIGURA 6.49 – PERIODO COMPLETO DE V_{AB} CON $M=0.9$ EN SVM	94
FIGURA 6.50 – NIVEL DE TENSIÓN MÁS ALTO DE V_{AB} CON $M=0.9$ EN SVM.....	94
FIGURA 6.51 – PERIODO COMPLETO DE V_{AB} CON $M=0.6$ EN SVM	94
FIGURA 6.52 – NIVEL DE TENSIÓN MÁS ALTO DE V_{AB} CON $M=0.9$ EN SVM.....	94
FIGURA 6.53 – PERIODO COMPLETO DE V_{AB} CON $M=0.4$ EN SVM	95
FIGURA 6.54 – NIVEL DE TENSIÓN MÁS ALTO DE V_{AB} CON $M=0.4$ EN SVM.....	95
FIGURA 6.55 – PERIODO COMPLETO DE V_{AB} CON $M=0.1$ EN SVM	95
FIGURA 6.56 – NIVEL DE TENSIÓN MÁS ALTO DE V_{AB} CON $M=0.1$ EN SVM.....	95
FIGURA 6.57 – FRECUENCIA MODULADORA BAJA (1).....	96
FIGURA 6.58 – FRECUENCIA MODULADORA BAJA (2).....	96
FIGURA 6.59 – FRECUENCIA MODULADORA MEDIA (1)	96
FIGURA 6.60 – FRECUENCIA MODULADORA MEDIA (2)	96
FIGURA 6.61 – FRECUENCIA MODULADORA ALTA (1).....	96
FIGURA 6.62 – FRECUENCIA MODULADORA ALTA (2).....	96





ÍNDICE DE TABLAS

TABLA 2.1 – TENSION DE SALIDA FASE-NEUTRO EN FUNCION DE LOS POLOS CONECTADOS	6
TABLA 2.2 – POSIBLES ESTADOS DE LOS POLOS DE POTENCIA	8
TABLA 3.1 – CORRESPONDENCIA DE LOS PINES LA TARJETA DE CONTROL CON LOS PINES DEL DOCKING STATION	19
TABLA 3.2 – AJUSTE DEL DEAD TIME	25
TABLA 3.3 – AJUSTES POR DEFECTO DEL DEAD TIME	25
TABLA 3.4 – AJUSTE DE LA PROTECCION DINAMICA CONTRA CORTOCIRCUITO	29
TABLA 3.5 – AJUSTE DE LA RESISTENCIA SERIE DE COLECTOR	30
TABLA 3.6 – VALOR DEL CONDENSADOR C_{IES}	31
TABLA 3.7 – VALOR DE LOS TIEMPOS DE ENCENDIDO Y APAGADO	31
TABLA 3.8 – AJUSTE DE LA RESISTENCIA DE PUERTA	32
TABLA 3.9 – AJUSTE DE LA RESISTENCIA PARA EL APAGADO SUAVE	32
TABLA 3.10 – RESISTENCIA PARA EL SENSOR DE TEMPERATURA	33
TABLA 4.1 – EJEMPLO DE CALCULO DE LOS TIEMPOS DE APLICACION CUANDO EL VECTOR DE REFERENCIA ESTÁ EN EL SECTOR 1	66





ÍNDICE DE ECUACIONES

ECUACIÓN 1 – ECUACIÓN PARA CALCULAR LA RESISTENCIA A INSTALAR PARA LA PROTECCIÓN CONTRA CORTOCIRCUITO	27
ECUACIÓN 2 – R_{CE} MÍNIMA	27
ECUACIÓN 3 – R_{CE} MÁXIMA	27
ECUACIÓN 4 – VALOR DEL CONDENSADOR C_{CE}	28
ECUACIÓN 5 – VALOR DE LAS RESISTENCIAS DE PUERTA EN FUNCIÓN DE LOS TIEMPOS DE ENCENDIDO Y APAGADO ...	31
ECUACIÓN 6 – VALOR TEÓRICO DE LA RESISTENCIA DE PUERTA	31
ECUACIÓN 7 – NUMERO DE VECTORES SINTETIZABLES	65
ECUACIÓN 8 – VALOR DE LA FRECUENCIA MODULADORA	74
ECUACIÓN 9 – VALOR DE LA FRECUENCIA PORTADORA.....	75
ECUACIÓN 10 – VALOR DE TBPRD	75



1. INTRODUCCIÓN

1.1. OBJETIVOS

El objetivo principal de este proyecto es el montaje y puesta en marcha de un inversor multinivel de tres niveles con fijación por diodos usando como controlador un procesador de señales digitales (DSP).

Sin embargo, este proyecto tiene unos objetivos específicos, que son:

- Realizar un breve estudio de la topología empleada para poder aplicar una estrategia de control en lazo abierto.
- Analizar los drivers de control usados para establecer las configuraciones adecuadas.
- Realización de pequeñas simulaciones del equipo para poder verificar los resultados experimentales obtenidos.
- Implementar una estrategia de conmutación con la que se puedan observar los diferentes niveles de forma inmediata.
- Validar el modelo para diferentes parámetros de configuración del DSP, para distintas cargas en la salida y para diferentes valores del voltaje de entrada.
- Implementar un modelo que sea capaz de controlar tanto la amplitud como la frecuencia del voltaje de salida.

1.2. MOTIVACIÓN

Los inversores o convertidores de corriente continua en corriente alterna (CC-CA) están teniendo cada día mayor aplicación en diferentes campos tales como la interconexión a red de sistemas de energías renovables, la calidad de energía, los métodos de compensación de potencia activa y reactiva, los filtros activos en los sistemas interconectados, y el accionamiento y variación de velocidad de máquinas eléctricas rotativas.



Dichos convertidores son el fundamento para la aplicación de las tecnologías mencionadas, pero ha existido la limitación de la potencia que puedan manejar debido a las especificaciones propias de cada dispositivo semiconductor que las componen, en cuanto a los niveles de tensión y potencia.

La perspectiva que se tiene es aplicar los inversores en sistemas eléctricos a tensiones cada vez más altas, lo que implica plantear una modificación en la topología convencional de los inversores, que les permitan a los dispositivos de estado sólido que los conforman soportar tensiones menores con respecto a las de un inversor de dos niveles y dos dispositivos por rama. Es ahí donde los inversores multinivel desempeñan un papel importante por su capacidad para manejar mayor potencia, reducir el contenido armónico de la tensión de salida y tener menores exigencias eléctricas en los dispositivos de conmutación.

Entre los diferentes convertidores multinivel, la topología *Diode-Clamped* es, sin duda, la más utilizada y estudiada. Para el caso particular de tres niveles, la topología *Diode-Clamped* se denomina *Neutral-Point-Clamped* (NPC), siendo esta la topología multinivel pionera y la más atractiva puesto que, en general, el compromiso óptimo entre ventajas e inconvenientes se consigue para tres niveles. Por esta razón, el presente proyecto se ha centrado en esta topología.

1.3. ESTRUCTURA DEL PROYECTO

En este apartado se detalla la estructura del presente proyecto, indicando el contenido de cada capítulo y las aportaciones más importantes realizadas.

Capítulo 2: Introducción a la topología multinivel

En este capítulo se hace una pequeña introducción a la topología empleada, especificando sus características principales, posibles estados de operación y pérdidas de potencia que tienen lugar en el inversor.



Capítulo 3: Hardware utilizado

En el capítulo 3 se presentan las diferentes placas hardware de control y acondicionamiento empleadas, y la configuración de cada uno de estos elementos para poder hacer posible un control óptimo.

Capítulo 4: Entorno de programación

En este capítulo se comentan los diferentes códigos de programación creados, simulaciones realizadas para comprobar el funcionamiento y la forma en la que hemos tenido que configurar los programas para poder comunicarnos con los dispositivos hardware comentados en el capítulo anterior.

Capítulo 5: Montaje del equipo

En el capítulo 5 se muestra cómo se ha ensamblado todo el equipo, habiendo creado una estructura para que la placa de potencia y los drives se encuentren montada sobre ella y aislados entre sí.

Capítulo 6: Pruebas realizadas

En este capítulo se muestran los resultados obtenidos en las diferentes pruebas y se procederá a validar los modelos realizados.





2. ESTUDIO DE LA TOPOLOGÍA MULTINIVEL EMPLEADA

La Figura 2.1 muestra el esquema eléctrico del convertidor multinivel empleado. Esta topología se denomina Diode-Clamped de tres niveles o Neutral-Point-Clamped-Converter (NPC).

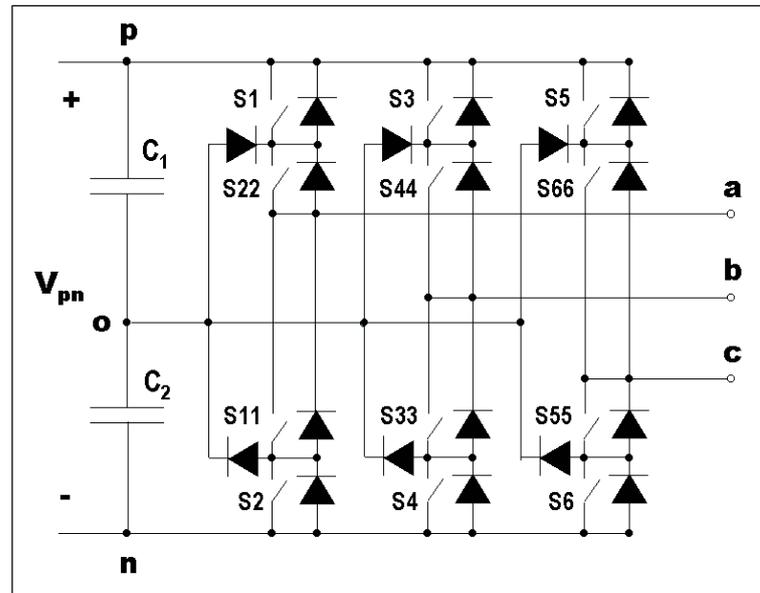


Figura 2.1 - Convertidor trifásico de tres niveles Diode-Clamped

En este circuito, la tensión del bus de continua V_{pn} se divide mediante dos capacidades, donde el punto medio 'o' suele definirse también como el punto neutro del bus de continua. La tensión de salida para cada fase (V_{ao} , V_{bo} , V_{co}) puede tomar tres niveles diferentes: $V_{pn}/2$, 0 y $-V_{pn}/2$. La tabla 2.1 muestra, a modo de ejemplo, el accionamiento adecuado de los interruptores para obtener los tres niveles en la tensión de salida V_{ao} , fácilmente extensible a las tensiones V_{bo} y V_{co} , resultado ilustrado en la Figura 2.2. En cada rama se encuentran dos pares complementarios, definiendo como tal aquel par de interruptores que, estando uno cerrado, obliga al otro a estar abierto. Para este convertidor son pares complementarios S1-S11 y S2-S22.

Polos Conectados	Tensión V_{ao}
S1 y S22	$V_{pn}/2$
S22 y S11	0
S11 y S2	$-V_{pn}/2$

Tabla 2.1 – Tensión de salida fase-neutro en función de los polos conectados

En la siguiente figura podemos ver cómo será la forma de onda de la tensión fase-neutro para cualquiera de las tres ramas, así como los interruptores que debemos activar en cada instante para lograr la tensión de salida deseada.

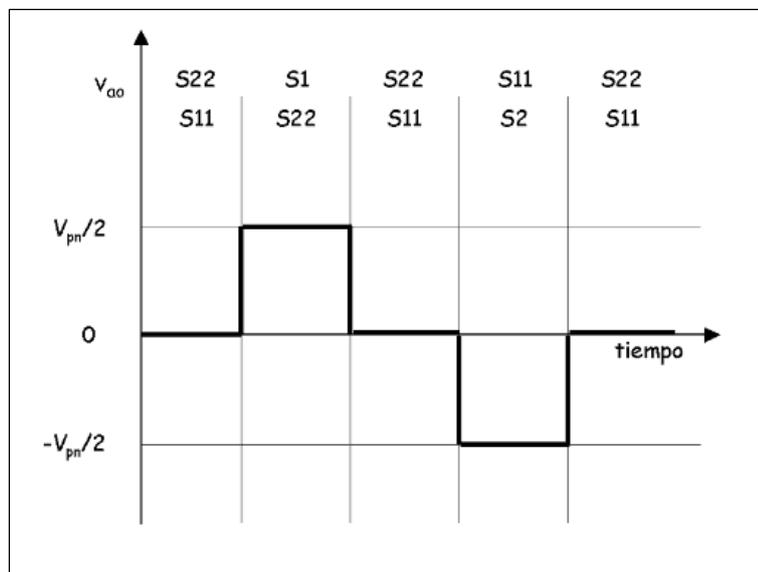


Figura 2.2 – Tensión de salida fase-neutro en función de los polos conectados

Los diodos conectados al punto medio del bus de continua 'o' son el elemento clave que diferencia este circuito de un inversor convencional, siendo su función la de fijar las tensiones de bloqueo de los interruptores a una fracción de la tensión del bus de continua, es decir, a la tensión de una capacidad, $V_{pn}/2$ en este caso. Los diodos de fijación también deben bloquear la misma tensión en este convertidor. Por otro lado, la tensión de salida analizada V_{ao} es alterna. No obstante, si se considera como tensión de salida la existente entre fase y negativo 'N', por ejemplo V_{an} , se obtienen tres niveles de tensión positivos (V_{pn} , $V_{pn}/2$, 0), y el convertidor trabaja como convertidor CC/CC. En una aplicación de continua, la corriente de salida es unidireccional, y no es posible mantener equilibrado el punto medio del bus de continua si se emplea



una sola rama, de aquí que esta topología no sea viable para la conversión CC/CC, a menos que se empleen dos ramas.

Las ventajas de esta topología son:

- La tensión de bloqueo de los interruptores es la tensión de una capacidad de entrada, $V_{pn}/2$.
- El número de capacidades requeridas es pequeño en comparación con otras topologías multinivel. Este punto es especialmente interesante dado que son los componentes reactivos los que suponen un mayor coste en el convertidor.
- Se pueden conectar directamente a un bus de continua, sin necesidad de crear otros buses adicionales.
- No requiere transformadores.
- Cambio de un estado a otro accionando un solo interruptor

Sim embargo esta topología también presenta algún inconveniente:

- Se requiere que los diodos de fijación sean de recuperación rápida y capaces de conducir la corriente nominal del convertidor, además de estar sometidos a una recuperación inversa exigente. Todos estos requisitos deben considerarse seriamente en el diseño del convertidor. Así mismo, los interruptores internos de cada rama pueden llegar a soportar tensiones instantáneas elevadas en el instante de la conmutación, provocadas por las inductancias parásitas.
- Es necesario que las tensiones de las capacidades se mantengan equilibradas en cualquier punto de trabajo, complicando el sistema de control del convertidor.

2.1. POSIBLES ESTADOS DEL INVERSOR

Existen diversas configuraciones posibles en función de que polos estén activados o desactivados en cada momento. Alguno de estos estados están permitidos, otros son potencialmente destructivos y otros no son



recomendados pero son posibles. Todos estos estados quedan reflejados en la siguiente tabla.

T1	0	0	0	1	0	0	1	0	1	1	0	1	1	1	0	1
T22	0	1	0	1	1	0	0	0	0	0	1	1	1	0	1	1
T11	0	0	1	0	1	1	0	0	0	1	0	1	0	1	1	1
T2	0	0	0	0	0	1	0	1	1	0	1	0	1	1	1	1
estado	Permitido						No recomendado					Destructivo				

Tabla 2.2 – Posibles estados de los polos de potencia

En cuanto a los estados permitidos tenemos los siguientes:

- Todos los polos apagados, el convertidor estará apagado.
- Cuando T22 o T11 se encienden por separado.
- Cuando dos polos adyacentes (T1 con T22, T22 con T11 y T11 con T2) están encendidos.

En cuanto a estados no recomendados, que podrán producir efectos negativos si el patrón aplicado a otra rama no es adecuado:

- Cuando únicamente T1 y T2 se encienden de forma junta o separada.
- Cuando dos polos no adyacentes (T1 con T11 o T2 con T22) están encendidos.

Por último, los estados destructivos:

- Cuando tres polos adyacentes están encendidos de forma simultánea (T1, T22 y T11 producen un cortocircuito en la rama superior de los condensadores, y T22, T22 y T2 producen un cortocircuito en la rama inferior de los condensadores).
- Cuando tres polos no adyacentes están encendidos de forma simultánea (T1, T11 y T2 produce que T22 soporte toda la tensión continua de entrada, y T1, T22 y T2 produce que T11 soporte toda la tensión continua de entrada)
- Los cuatro transistores estén activos al mismo tiempo, ya que se cortocircuitaría la tensión continua positiva de entrada con la negativa



y con el valor 0V de continua produciendo un cortocircuito en la fuente de alimentación con la que alimentamos la etapa de potencia

2.2. ÁREAS DE OPERACIÓN DEL INVERSOR

La Figura 2.3 muestra la forma de onda de la tensión y corriente para una carga inductiva-resistiva.

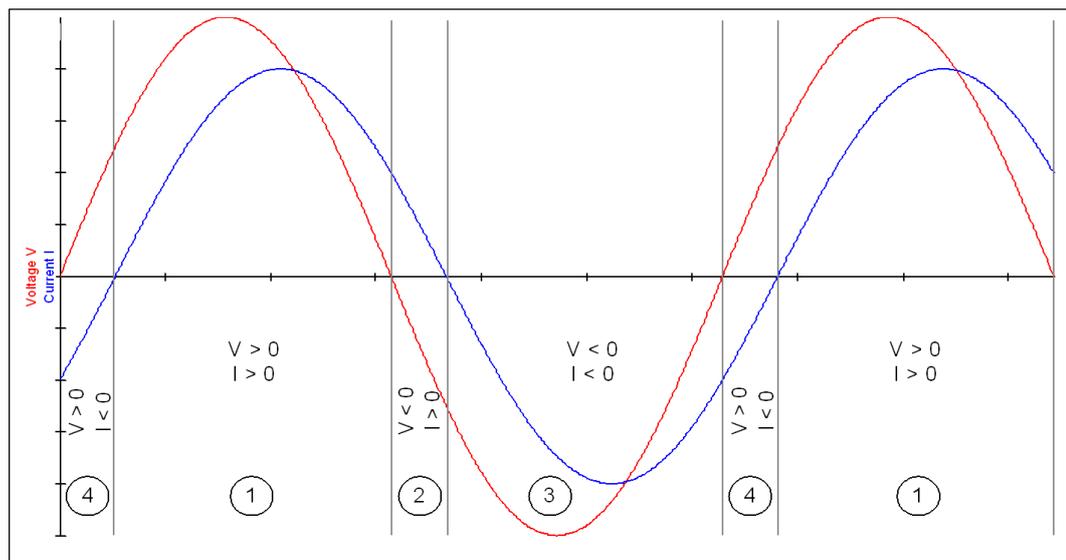


Figura 2.3 – Áreas de operación para carga inductiva

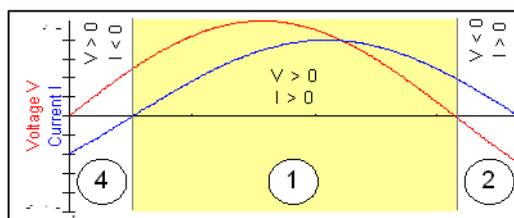


Figura 2.4 – Área de operación 1

En esta área de operación tanto la tensión de salida como la corriente son positivas. Para conseguir una corriente positiva con tensión positiva es necesario activar los dos polos superiores y para obtener corriente positiva con tensión cero es necesario activar el segundo polo. En la siguiente figura podemos ver como circulará la corriente.

superiores y para obtener corriente positiva con tensión cero es necesario activar el segundo polo. En la siguiente figura podemos ver como circulará la corriente.

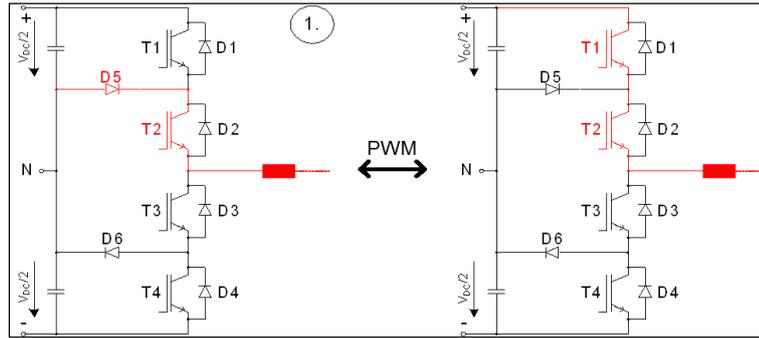


Figura 2.5 – Conmutación de los polos en el área 1 de operación

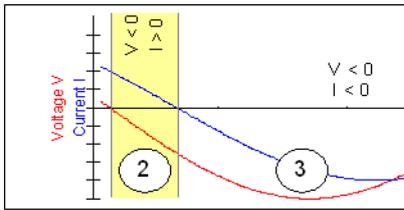


Figura 2.6 – Área de operación 2

En esta segunda área de operación la tensión de salida es negativa y la corriente positiva. Con tensión negativa (polos 3 y 4 activados) la corriente no circulará por los transistores sino por los diodos de estos polos para que sea positiva. Con tensión nula, el polo 2 debe estar activado para poder tener corriente positiva a través del diodo D5.

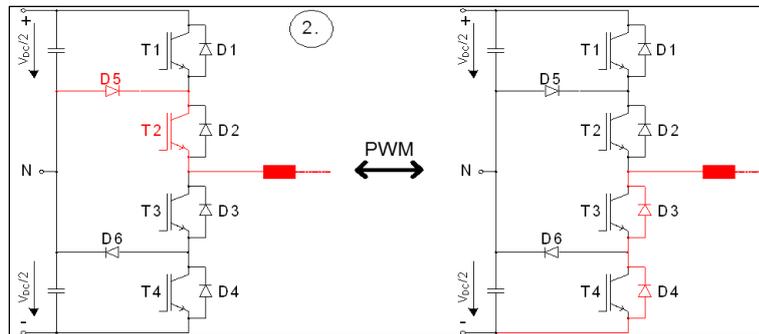


Figura 2.7 – Conmutación de los polos en el área 2 de operación

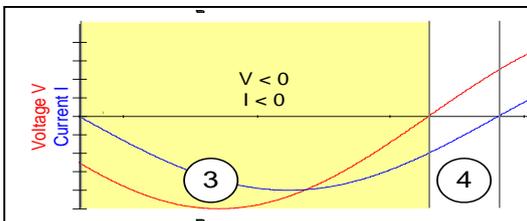


Figura 2.8 – Área de operación 3

En el área 3 la tensión de salida es negativa al igual que la corriente. Con tensión de salida negativa, los polos 3 y 4 están activados, y al ser la corriente negativa, ésta circulará por los transistores de estos polos. Cuando la tensión de salida sea nula, la corriente circulará por el diodo 6 estando el polo 3 encendido.

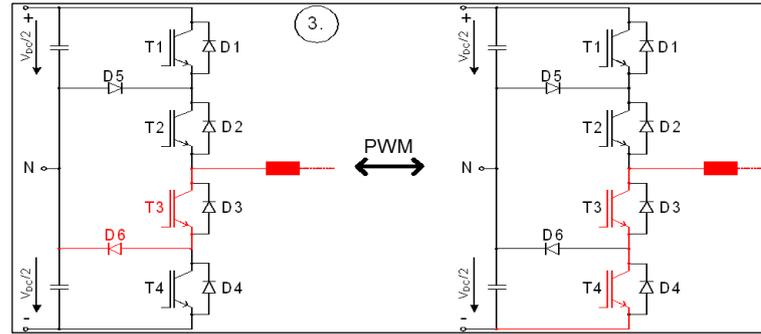


Figura 2.9 - Conmutación de los polos en el área 3 de operación

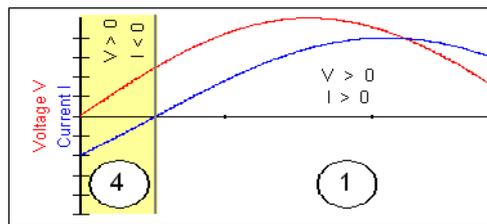


Figura 2.10 - Área de operación 4

En esta última área de operación la tensión de salida es positiva y la corriente negativa. Como bien sabemos para obtener una tensión positiva debemos activar los polos 1 y 2, pero la corriente no circulará por los transistores sino por los diodos de estos polos. Cuando la tensión de salida sea nula, la corriente circulará por el polo 3 y el diodo 6.

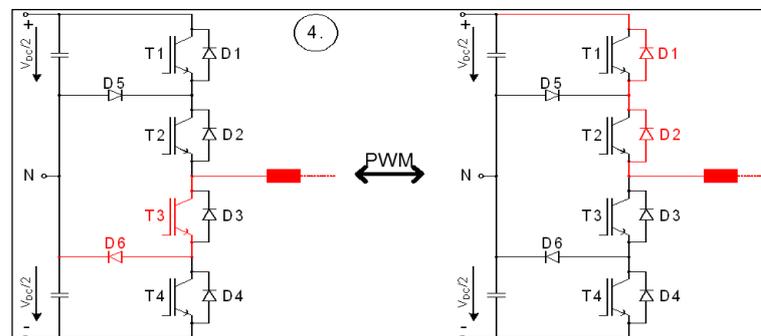


Figura 2.11 - Conmutación de los polos en el área 3 de operación

La Figura 2.12 muestra un resumen de todos estos estados para cargas únicamente resistivas, estando diferenciados por los casos en los que la corriente de salida es positiva o negativa, en función de los estados de los cuatro polos de potencia.

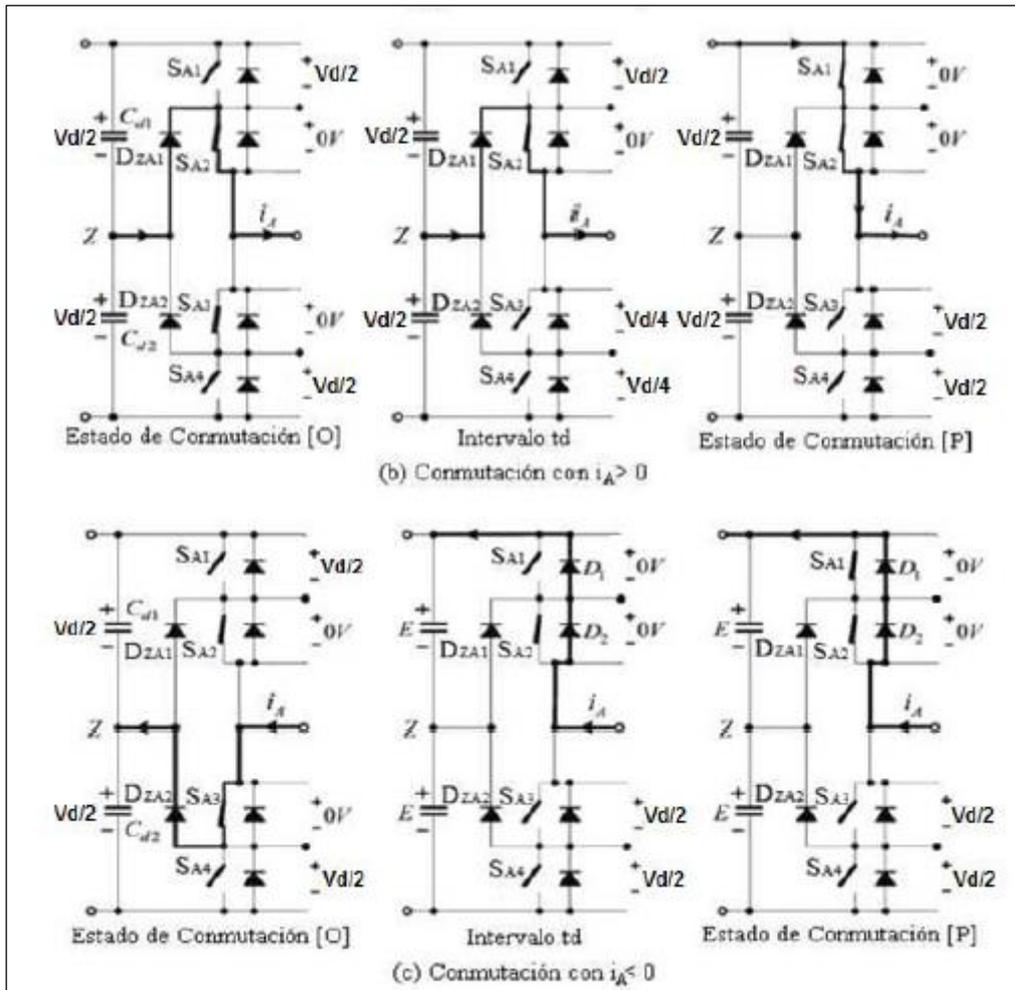


Figura 2.12 - Resumen áreas de operación

2.3. PÉRDIDAS DE POTENCIA

Las pérdidas dependerán de si estamos usando el dispositivo como rectificador o como inversor. Se estará usando como rectificador cuando la potencia consumida sea negativa, es decir, cuando estemos aportando una potencia a la red, o como inversor cuando estemos necesitando que la red nos proporcione una potencia para poder actuar.

Para un caso determinado las pérdidas de potencia se pueden observar en la siguiente figura.

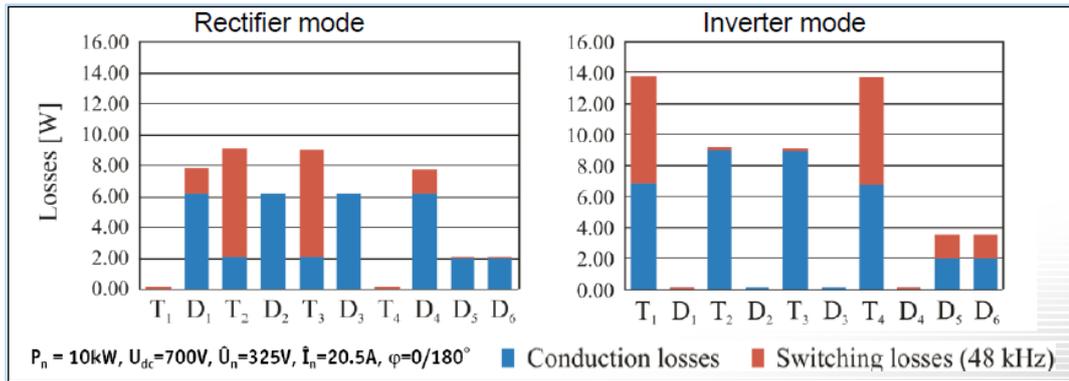


Figura 2.13 – Pérdidas de potencia

Como podemos ver, cuando estamos trabajando como inversor, las pérdidas se producen sobre todo en los transistores, teniendo especial importancia las pérdidas por conmutación en los transistores superior e inferior. En cambio, al trabajar como rectificador, las pérdidas están mucho más repartidas entre los diferentes elementos, destacando que los transistores superior e inferior que antes sufrían la mayor parte de pérdidas, ahora casi no tienen pérdidas, y la mayor parte de las pérdidas por conmutación se deben a los dos transistores intermedios.

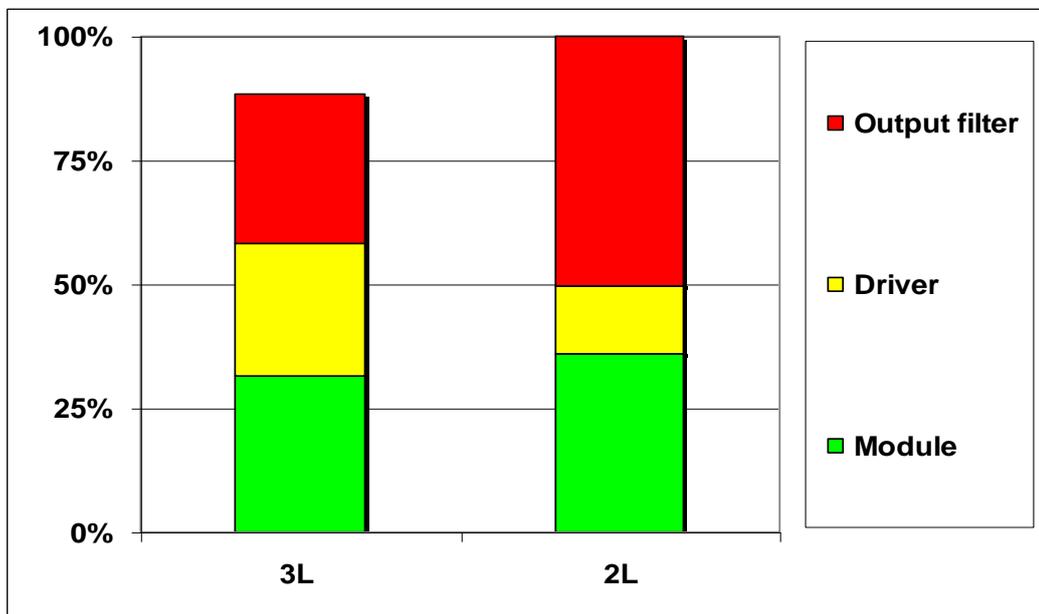


Figura 2.14 – Pérdidas de potencia en un equipo completo

En esta figura podemos ver como el inversor multinivel sufre menos pérdidas en comparación con un inversor tradicional, y vemos como las pérdidas de los



diferentes elementos están más repartidas. Se han disminuido las pérdidas en el filtro con el que conseguimos una tensión sinodal y en los transistores y diodos, pero se ha aumentado las pérdidas en el driver de control, ya que en este caso necesitamos el doble de drivers que en un inversor tradicional.



3. HARDWARE UTILIZADO

En este proyecto se han empleado tres elementos de hardware, que son: DSP de Texas Instruments cuyo controlador es el TMS320F28069, junto con la estación Docking Station que permite obtener las salidas de la tarjeta en forma de pines, una tarjeta de adaptación de niveles, que transforma el nivel de 3.3V que nos proporciona el DSP en 15V que necesita el driver de control para funcionar correctamente.

3.1. DSP TMS320F28069

La razón para no utilizar un procesador común y utilizar este procesador digital de señales se debe a la eficiencia de cálculo y la resolución de manejo de datos que disponen los DSP frente a los demás tipos de procesadores, por lo que son más rápidos y exactos.

El DSP es el elemento inteligente que gobierna todo el equipo de hardware diseñado y por lo tanto se encargará del control del sistema. Realizará el control según los algoritmos que se le graben, los cuales se explicarán en capítulos posteriores.

Las principales características que posee este procesador son las siguientes:

- CPU de 32 bits de alta eficiencia (TMS320C28x).
 - Frecuencia de funcionamiento de hasta 90 MHz.
 - 16 x 16 y 32 x 32 operaciones de multiplicación y acumulación.
 - 16 x 16 Dual MAC.
 - Arquitectura de bus Harvard.
 - Operaciones atómicas.
 - Rápida respuesta y procesamiento de las interrupciones.
 - Código-eficiente (en C / C++).
- Unidad de punto flotante (FPU).
 - Operaciones nativas de precisión simple de punto flotante.
- Programable Control Law Accelerator (CLA).
 - 32-bit de punto flotante.



- Ejecuta el código independientemente de la CPU principal.
- Viterbi, Matemáticas Complejas, Unidad CRC (VCU).
- Memoria integrada.
 - Hasta 256 KB de flash.
 - Hasta 100 KB de RAM.
 - 2 KB de One-Time Programmable ROM (OTP).
- 6-Canales acceso directo a memoria (DMA).
- Dispositivo pequeño y de poco consumo.
 - 3.3V de tensión de alimentación.
 - Baja potencia en los modos de funcionamiento.
- Endianness: Little Endian.
- Soporte JTAG Boundary Scan.
 - Acceso IEEE Standard 1.149,1-1990 prueba estándar Puerto y Arquitectura Boundary Scan.
- Señales de reloj.
 - Dos osciladores internos.
 - Reloj externo de entrada.
 - Módulo Watchdog Timer.
- Periféricos de interrupción de expansión (PIE) que soporta todas las Interrupciones de los periféricos.
- Tres temporizadores CPU de 32 bits.
- Periféricos de control avanzados.
- Hasta 8 módulos ePWM.
 - 16 PWM totales (8 HRPWM).
 - Temporizador de 16 bits independiente en cada módulo.
- Tres entradas de captura mejorada (ECAP).
- Hasta 4 módulos de captura de alta resolución (HRCAP).
- 12-Bit Analog-to-Digital Converter (ADC).
 - Hasta 3,46 MSPS.
 - Hasta 16 Canales.
- Sensor de temperatura integrado en el Chip.
- Clave de seguridad de 128 bits y bloqueo.



- Protege bloques de memoria de forma segura.
- Evita lecturas del código programado.
- **Periféricos Serial Port.**
 - Dos Serial Communications Interface (SCI).
 - Dos módulos Serial Peripheral Interface (SPI).
 - Un Bus I²C.
 - Un Enhanced Controller Area Network (ECAN).
 - Universal Serial Bus (USB) 2.0.
 - Modo de Dispositivo Full-Speed.
 - Full-Speed o Modo Host a baja velocidad.
- Hasta 54, programables de forma independiente, de uso general entrada / salida (GPIO) pines con filtrado de entrada.
- **Características avanzadas de Emulación.**
 - Depuración Real-Time a través de Hardware.

En la siguiente figura podemos observar el diagrama de bloques funcional del dispositivo, en el que se pueden observar todos los elementos arriba comentados, aunque para este proyecto los bloques más importantes son los bloques PWMx, los cuales se analizarán a la hora de desarrollar el código de programación en apartados posteriores.

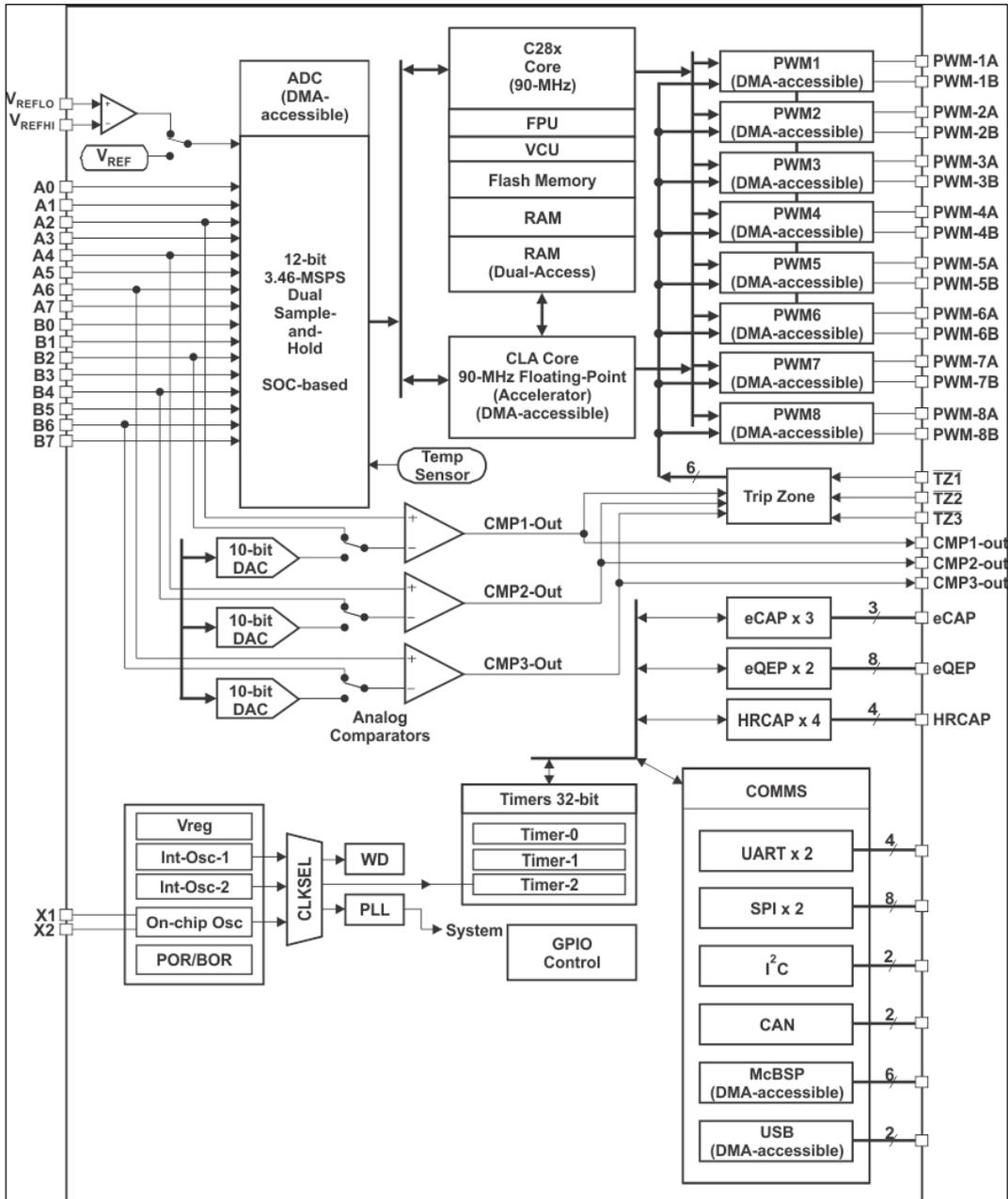


Figura 3.1 - Diagrama de bloques funcional del DSP TMS320F28069

3.1.1. Docking Station

Para poder usar las salidas de la tarjeta DSP de control se usará este modelo en cual se conecta por USB a un ordenador, y a través de este puerto se alimentará el dispositivo y se transferirá el programa de control. En la siguiente figura podemos observar el aspecto de esta tarjeta con la tarjeta DSP de control instalada.

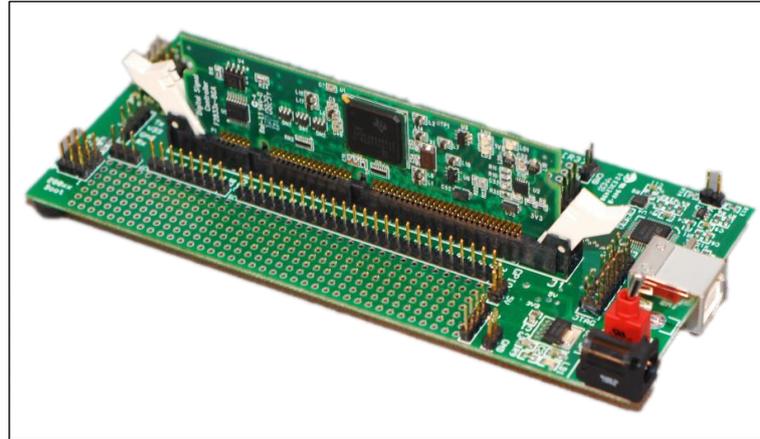


Figura 3.2 - Docking Station con la tarjeta de control instalada

En la Figura 3.2 podemos observar los pines accesibles, los que están alineados de forma horizontal son los pines GPIO y los que están en vertical son las señales de 3.3V y GND. Las señales de salida que se puede ver en cada pin están reflejas en la siguiente tabla.

F28035 controlCARD board pin-out description

V33D-ISO	1	51	V33D-ISO
ISO-RX-RS232	2	52	ISO-TX-RS232
spare	3	53	spare
spare	4	54	spare
spare	5	55	spare
GND_ISO	6	56	GND_ISO
ADCIN-B0	7	57	ADCIN-A0
GND	8	58	GND
ADCIN-B1	9	59	ADCIN-A1
GND	10	60	GND
ADCIN-B2/COMP1B/AIO10	11	61	ADCIN-A2/COMP1A/AIO2
GND	12	62	GND
ADCIN-B3	13	63	ADCIN-A3
GND	14	64	GND
ADCIN-B4/COMP2B/AIO12	15	65	ADCIN-A4/COMP2A/AIO4
VREFLO-ext	16	66	VREFHI-ext
ADCIN-B5	17	67	ADCIN-A5
spare	18	68	spare
ADCIN-B6/COMP3B/AIO14	19	69	ADCIN-A6/COMP3A/AIO6
spare	20	70	spare
ADCIN-B7	21	71	ADCIN-A7
spare	22	72	spare
GPIO-00 / EPWM-1A	23	73	GPIO-01 / EPWM-1B / EMU0 / COMP1OUT
GPIO-02 / EPWM-2A	24	74	GPIO-03 / EPWM-2B / SPISOMIA / COMP2OUT
GPIO-04 / EPWM-3A	25	75	GPIO-05 / EPWM-3B / SPISOMIA / ECAP1
GPIO-06 / EPWM-4A / EPWMSYNCl	26	76	GPIO-07 / EPWM-4B / SCIRX-A
GND	27	77	ESD1
GPIO-08 / EPWM-5A / -- / ADCSOCA	28	78	GPIO-09 / EPWM-5B / LINTX-A
GPIO-10 / EPWM-6A / -- / ADCSOCA	29	79	GPIO-11 / EPWM-6B / LINTX-A
GPIO-40 / EPWM-7A	30	80	GPIO-41 / EPWM-7B
GPIO-42 / -- / -- / COMP1OUT	31	81	GPIO-43 / -- / -- / COMP2OUT
GPIO-44	32	82	ESD2
GPIO-12 / TZ1n / SCITX-A / SPISIMO-B	33	83	GPIO-13 / TZ2n / -- / SPISOMI-B
GPIO-15 / TZ1n / LINTX-A / SPISTE-B	34	84	GPIO-14 / TZ3n / LINTX-A / SPICLK-B
GPIO-24 / ECAP1 / -- / SPISIMO-B	35	85	GPIO-25 / -- / -- / SPISOMI-B
GPIO-26 / -- / -- / SPICLK-B	36	86	GPIO-27 / -- / -- / SPISTE-B
GND	37	87	ESD3
GPIO-16 / SPISIMO-A / -- / TZ2n	38	88	GPIO-17 / SPISOMI-A / -- / TZ3n
GPIO-18 / SPICLK-A / LINTX-A / XCLKOUT	39	89	GPIO-19 / SPISTE-A / LINTX-A / ECAP1
GPIO-20 / EQEP1A / -- / COMP1OUT	40	90	GPIO-21 / EQEP1B / -- / COMP2OUT
GPIO-22 / EQEP1S / -- / LINTX-A	41	91	GPIO-23 / EQEP1I / -- / LINTX-A
GPIO-39	42	92	ESD4
GPIO-28 / SCIRX-A / I2CSDA-A / TZ2n	43	93	GPIO-29 / SCITX-A / I2CSCL-A / TZ3n
GPIO-30 / CANRX-A / -- / --	44	94	GPIO-31 / CANTX-A / -- / --
GPIO-32 / I2CSDA-A / EPWMSYNCl / ADCSOCA	45	95	GPIO-33 / I2CSCL-A / EPWMSYNCO / ADCSOCA
GPIO-34 / COMP2OUT / EMU1 / COMP3OUT	46	96	ESD5
GND	47	97	TDI / GPIO-35
TCK / GPIO-38	48	98	TDO / GPIO-37
TMS / GPIO-36	49	99	TRSTn
	50	100	

Tabla 3.1 - Correspondencia de los pines la tarjeta de control con los pines del Docking Station

3.2. TARJETA DE ADAPTACIÓN DE NIVELES

Para transformar los pulsos digitales que salen de los pines GPIO del DSP en señales de un nivel de tensión suficiente para poder realizar los disparos de activación de los polos de potencia del convertidor se tiene la tarjeta de adaptación de niveles, cuya fotografía real aparece representada en la Figura 3.3. Esta tarjeta de adaptación de niveles recibirá la secuencia de disparos de control generada por el DSP de niveles 0/3,3V, elevará dichos disparos al nivel 0/15V capaz de disparar los polos de potencia a través del driver de control, y mandará estas señales elevadas a dicho driver. Esta tarjeta se alimentará a una tensión continua de 5V, y puede requerir hasta 4A (2A cada tarjeta, 4A las dos tarjetas) para un correcto funcionamiento.

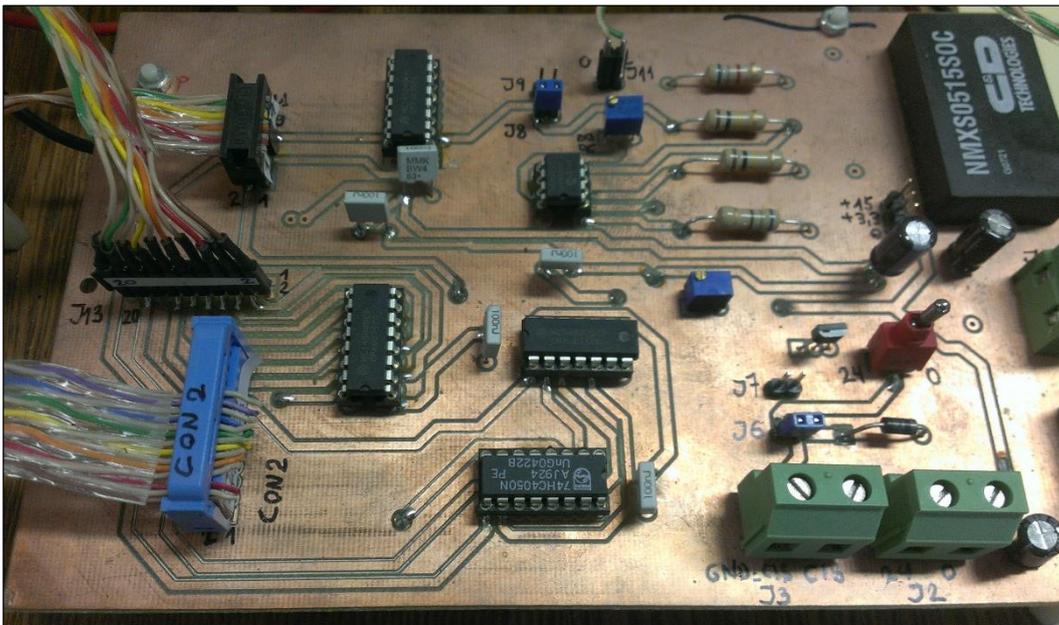


Figura 3.3 – Tarjeta de adaptación de niveles

Además de esta función primordial, la tarjeta de adaptación de niveles permite llevar a cabo otras funciones de control y manejo del convertidor. Para indicar cuáles son estas funciones y qué elementos las realizan, se muestra en la Figura 3.4 el esquema eléctrico de esta tarjeta seccionado por las diferentes zonas de funcionamiento que lo componen. Aunque estas funcionalidades extras no serán necesarias en este proyecto.

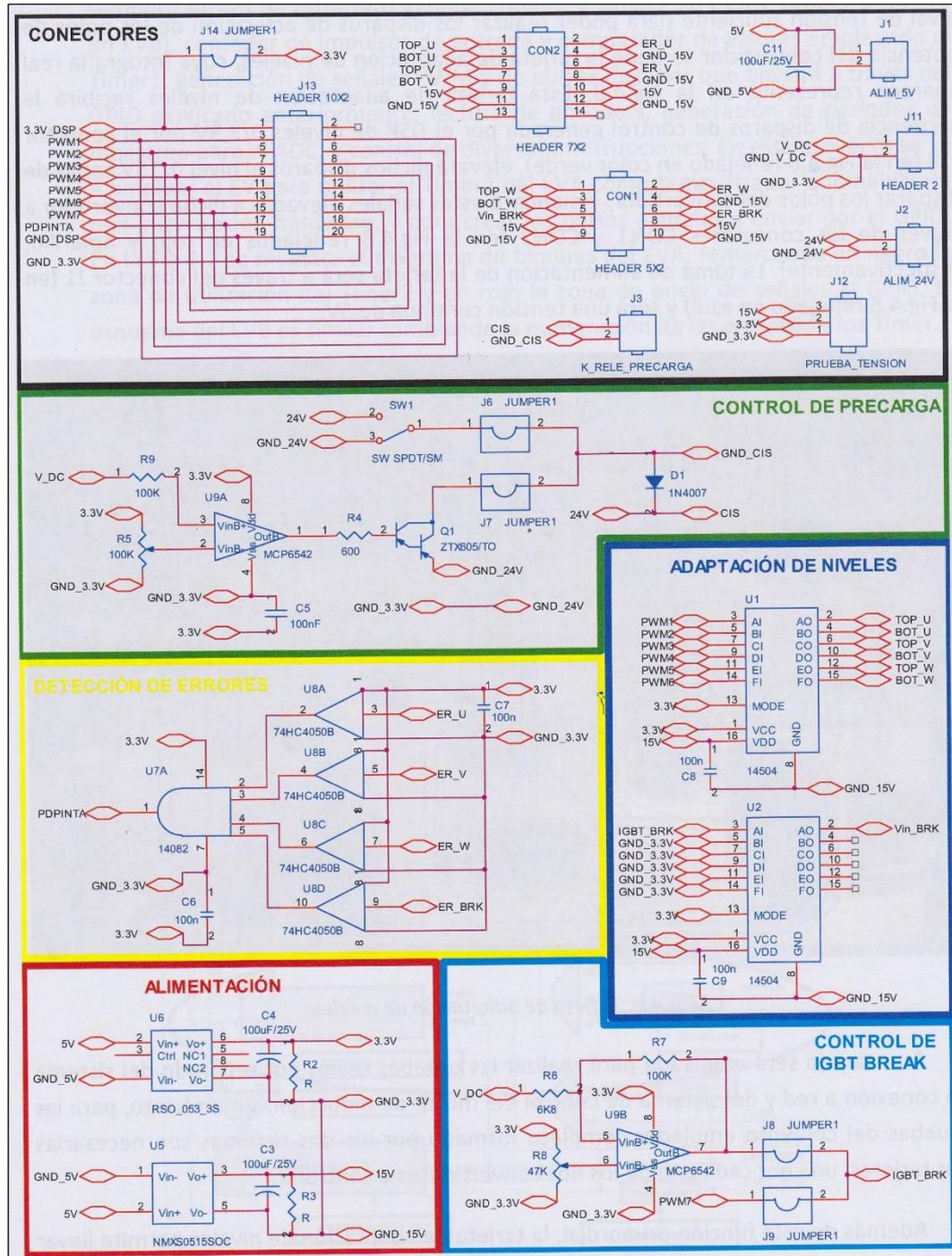


Figura 3.4 - Esquema eléctrico de la tarjeta de adaptación de niveles

Entre los diversos jumpers y conectores tenemos los siguientes, cada uno con una funcionalidad diferente:

- **J1:** Alimentación de 5V. Es la alimentación general de la tarjeta.
- **J2:** Alimentación de 24V. No es necesaria para este proyecto.
- **J3:** Conexión Relé de Precarga. No se usará en este proyecto.



- **J6 y J7: Configuración IGBT Precarga. No se usará relé de precarga.**
- **J8 y J9: Configuración IGBT Frenado (PWM7). No se usará.**
- **J11: V_{dc} . Medida de esta tensión sin importancia en este proyecto.**
- **J12: Prueba de diferentes tensiones (15V, 3.3V y GND).**
- **J13: Conexión PWMx (entrada a 3.3V)**
 - **PIN 1: 3.3V para la alimentación (solo si DSP necesita alimentación externa).**
 - **PIN 3 y 4: PWM1**
 - **PIN 5 y 6: PWM2**
 - **PIN 7 y 8: PWM3**
 - **PIN 9 y 10: PWM4**
 - **PIN 11 y 12: PWM5**
 - **PIN 13 y 14: PWM6**
 - **PIN 15 y 16: Señal PWM para IGBT frenado**
 - **PIN 17 y 18: Señal de error global**
 - **PIN 19 y 20: GND**
- **J14: Sirve para alimentar 3.3V el DSP. Solo si necesitamos alimentar externamente el DSP poner este jumper usar el cable del PIN1 de J13.**
- **CON1: Salidas PWM5 y PWM6 (salida a 15V)**
 - **PIN 1: PWM5**
 - **PIN 3: PWM6**
 - **PIN 7 y 8: Alimentación 15V**
 - **PIN 4, 9 y 10: GND**
 - **PIN 5, 2, y 6: Señales de error y Frenado. No se usan.**
- **CON2: Salidas PWM1, PWM2, PWM3 y PWM4 (salida a 15V)**
 - **PIN 1: PWM1**
 - **PIN 3: PWM2**
 - **PIN 5: PWM3**
 - **PIN 7: PWM4**
 - **PIN 9 y 10: Alimentación 15V**
 - **PIN 4, 8, 11 y 12: GND**
 - **PIN 2 y 6: Señales de error. No se usan**
 - **PIN 13 y 14: Sin conexión**



Para conseguir todas estas funcionalidades la placa cuenta con una serie de circuitos impresos.

- U1 y U2: Adaptación de niveles desde DSP hasta placa del inversor.
- U5 y U6: Generar tensiones de 3.3V y 15V a partir de 5V.
- U7: Adaptar y combinar señales de error.
- U9: IGBT de frenado y relé de precarga.

Por último la placa cuenta con unos potenciómetros e interruptores configurables.

- R5: Configuración relé de precarga.
- R8: Configuración IGBT de frenado.
- SW1: Selección de señales para configurar relé de precarga.

3.3. DRIVER DE CONTROL

El driver de control del inversor está formado por dos partes, en primer lugar el núcleo del driver que realiza el control, y por otro lugar la placa de adaptación sobre la que se configurarán las opciones deseadas.

El driver de control seleccionado es de la marca Texas Instruments modelo SKYPER 32 PRO R, que cuenta con estas características principales:

- Posibilidad de controlador dos transistores con un driver
- Protección ante cortocircuitos
- Emite una señal digital con valor positivo cuando se produce una parada en su funcionamiento.
- Frecuencia máxima de operación: 50 KHz.

El driver usado es el reflejado en la siguiente figura.



Figura 3.5 - SKYPER 32 PRO R

La tarjeta de adaptación del driver es de la misma marca que el driver, Texas Instruments, y modelo Board 1 SKYPER 32 PRO R, que cuenta con las siguientes características:

- Tensión de alimentación: 15V.
- Tensión de salida para transistor encendido: 15V.
- Tensión de salida para transistor apagado: -7V.

En la siguiente figura podemos ver esta placa, junto con el driver instalado, y con un supuesto módulo de transistores conectado en sus extremos.



Figura 3.6 - Board 1 SKYPER 32 PRO R

3.3.1. CONFIGURACIÓN DEL DRIVER DE CONTROL

Los requerimientos del driver se configuran sobre la placa de adaptación, sobre la cual se montarán o quitarán unos determinados elementos para conseguir el objetivo deseado.



AJUSTE DEL DEAD TIME

Este parámetro hace referencia a un tiempo de espera entre que se da la orden de apagado de un transistor hasta que el otro transistor puede ser encendido. En nuestro caso, al ser una topología multinivel debemos deshabilitar esta función ya que queremos que en ciertos momentos ambos transistores estén activos y no se producirá un cortocircuito.

Si observamos la hoja de características del driver obtenemos que este valor se puede ajustar siguiendo esta tabla:

Adjustment of Dead time / Neutralizing Locking Functions				
Interlock time [μs]	PRIM_CFG_TDT1_IN	PRIM_CFG_TDT2_IN	PRIM_CFG_TDT3_IN	PRIM_CFG_SELECT_IN
1	GND	GND	open	open
1,3	GND	GND	GND	open
2	GND	open	open	open
2,3	GND	open	GND	open
3	open	GND	open	open
3,3	open	GND	GND	open
4 *	open	open	open	open
4,3	open	open	GND	open
no interlock	open	open	open	GND

* Factory setting

Tabla 3.2 – Ajuste del Dead Time

Y en la placa sobre la que irá montado el driver es en donde tenemos que configurar estos parámetros en función de otra tabla proporcionada:

DT adjustment		
Designation	Shape	Setting
R43 (connected to GND)	0603 (SMD)	PRIM_CFG_TDT2_IN Factory setting: 0Ω
R44 (connected to GND)	0603 (SMD)	PRIM_CFG_SELECT_IN Factory setting: not equipped
R45 (connected to GND)	0603 (SMD)	PRIM_CFG_TDT3_IN Factory setting: 0Ω
R46 (connected to GND)	0603 (SMD)	PRIM_CFG_TDT1_IN Factory setting: not equipped

Factory setting: 3,3μs

Tabla 3.3 – Ajustes por defecto del Dead Time

En el puerto PRIM_CFG_TDT1_IN, que se corresponde con el hueco denominado R46 en la placa debemos dejarlo al aire, y como esta resistencia está al aire por defecto no tenemos que modificar este hueco.

En el puerto PRIM_CFG_TDT2_IN, que se corresponde con el hueco denominado R43 en la placa debemos dejarlo al aire, y como en este hueco hay montada una resistencia de 0Ω por defecto, debemos desoldar esta resistencia y dejar el hueco al aire.

En el puerto PRIM_CFG_TDT2_IN, que se corresponde con el hueco denominado R43 en la placa debemos dejarlo al aire, y como en este hueco hay montada una resistencia de 0Ω por defecto, debemos desoldar esta resistencia y dejar el hueco al aire.

En el puerto PRIM_CFG_SELECT_IN, que se corresponde con el hueco denominado R44 en la placa debemos cortocircuitarlo, y como esta resistencia está al aire por defecto tenemos que soldar un puente en este hueco.

AJUSTE DE LA PROTECCIÓN DINÁMICA CONTRA CORTOCIRCUITO (DYNAMIC SHORT CIRCUIT PROTECTION)

Es protección se implantará con la conexión en paralelo entre un condensador y una resistencia. Estos dos componentes intentarán generar una constante de tiempo que sea similar al voltaje V_{CE} real.

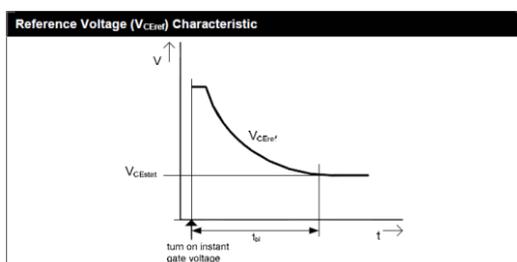


Figura 3.7 – Forma de onda del valor V_{CE}

En la figura de la izquierda vemos el comportamiento que seguirá la tensión creada. Vemos como a partir de un voltaje alto, una vez recibida la instrucción de activar el transistor, se llega a un voltaje menor en un

intervalo de tiempo determinado. El valor final alcanzado, llamado V_{CEstat} es el valor V_{CE} máximo de cada transistor cuando están activados. El tiempo que tarda en producirse esta transición se llamará t_{b1} , y se elegirá de manera que la curva se adapte de la mejor forma posible a la curva real.



$$R_{CE} [K\Omega] = -15.5k\Omega * \ln \left(1 - \frac{V_{CEstat} + R_{VCE} * \frac{V}{k\Omega}}{8} \right)$$

Ecuación 1 – Ecuación para calcular la resistencia a instalar para la protección contra cortocircuito

Para determinar R_{VCE} observamos la hoja de características y vemos que el valor de esta resistencia es de 0Ω ya que esta resistencia solo se debe montar cuando se están usando unos polos de potencia de 1700V. Esta información esta sacada de la siguiente figura:

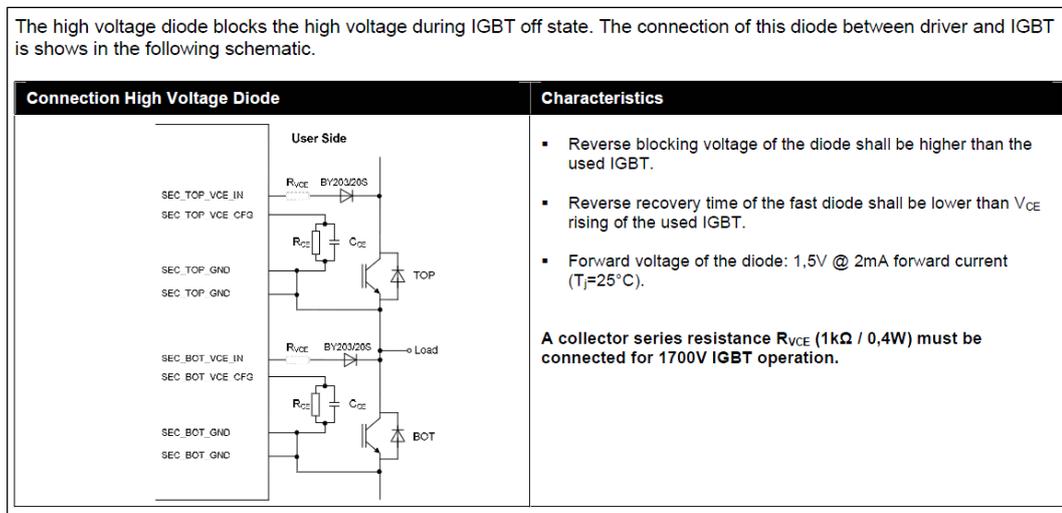


Figura 3.8 – High Voltaje Diode

Y procederemos a calcular el valor de la resistencia que usaremos, considerando que el voltaje estará entre 4.90V y 5.1V tal y como nos recomienda el fabricante:

$$R_{CE} [K\Omega] = -15.5k\Omega * \ln \left(1 - \frac{4.90 + 0 * \frac{V}{k\Omega}}{8} \right) = 14.69 k\Omega$$

Ecuación 2 – R_{CE} mínima

$$R_{CE} [K\Omega] = -15.5k\Omega * \ln \left(1 - \frac{5.1 + 0 * \frac{V}{k\Omega}}{8} \right) = 15.73 k\Omega$$

Ecuación 3 – R_{CE} máxima

La resistencia elegida será de 15 k Ω , 1%.

En cuanto al cálculo del condensador primero debemos determinar qué tiempo necesitamos, para ello nos fijamos en una hoja de cálculo en Excel recibida por correo del fabricante:

Topic: Calculation R_{CE} , C_{CE} , V_{CEstat} & t_{bl}

Driver: SKYPER 32 R, SKYPER 32PRO R

$R_{VCE} =$ $k\Omega$

Calculation for SKYPER 3

$V_{CEstat} =$ $V \Rightarrow R_{CE} =$ $k\Omega$

$t_{bl} =$ $\mu s \Rightarrow C_{CE} =$ pF

$R_{CE} =$ $k\Omega \Rightarrow V_{CEstat} =$ V

$C_{CE} =$ $pF \Rightarrow t_{bl} =$ μs

Calculation for SKYPER 32PRO R

$V_{CEstat} =$ $V \Rightarrow R_{CE} =$ $k\Omega$

$t_{bl} =$ $\mu s \Rightarrow C_{CE} =$ pF

$R_{CE} =$ $k\Omega \Rightarrow V_{CEstat} =$ V

$C_{CE} =$ $pF \Rightarrow t_{bl} =$ μs

Hello Daniel,

Please find attached an excel file that can help you to calculate the Rce and Cce depending on Vce_stat and tbl.

Vce_stat should be higher than Vce_sat, we would suggest around 5V.
tbl should be smaller than tpsc, we would suggest 3-4us.

You can find more information about this topic on the pages 348 and following of the SEMIKRON Application Book:
<http://shop.semikron.com/en/Products-and-Shop/Product-Groups/Others/APPLICATION-MANUAL-2011.html>
You can download it for free by clicking "PDF Version as download".

Best Regards,
Hugo Paz

Figura 3.9 – Cálculo de la resistencia y del condensador en función del voltaje y del tiempo

Para que no tengamos problemas debemos considerar que nuestra gráfica vaya adelantada sobre la gráfica real, para ello elegiremos un condensador teniendo en cuenta que el tiempo debe de estar en torno a 4 microsegundos.

$$C_{CE} [pF] = \frac{t_{bl} [\mu s] - 2.1 \mu s - 0.11 \frac{\mu s}{\Omega} * R_{CE}}{0.00323 \frac{\mu s}{pF}} = \frac{4 \mu s - 2.1 \mu s - 0.11 \frac{\mu s}{\Omega} * 15.2}{0.00323 \frac{\mu s}{pF}}$$

$$= 70.49 pF$$

Ecuación 4 – Valor del condensador C_{CE}

El condensador elegido será de 68 pF.

Por último tenemos que considerar que vamos a realizar un convertidor multinivel, y si analizamos la hoja de características del driver multinivel podemos ver como en cada rama de cuatro transistores que hay, solo debemos programar esta protección para los transistores superior e inferior del puente y no para los transistores T22 y T11, esto es debido a que la tensión colector-emisor en estos transistores no sigue la forma de onda esperada, ya



que cuando los polos superior e inferior están apagados, la tensión en el punto medio de estos dos transistores no será fija.

Setting Dynamic Short Circuit Protection

The short circuit protection of T2/T3 is deactivated to ensure a safe switch off procedure in multilevel topology.

Figura 3.10 – Desactivación de la protección dinámica contra cortocircuito para T22-T11

Para desactivar esta protección en los transistores indicados debemos establecer las conexiones que nos marca el datasheet:

Application hints

If the DSCP function is not used, for example during the experimental phase, SEC_TOP_VCE_IN must be connected with SEC_TOP_GND for disabling SCP @ TOP side and SEC_BOT_VCE_IN must be connected with SEC_BOT_GND for disabling SCP @ BOT side.

Figura 3.11 – Conexiones para desactivar la protección dinámica contra cortocircuito

Los componentes a montar los debemos conectar en las siguientes referencias:

R _{CE} & C _{CE}			
Designation	Shape	Setting	
R162	1206 (SMD)	R _{CE} Factory setting: not equipped	TOP
C150	1206 (SMD)	C _{CE} Factory setting: not equipped	TOP
R262	1206 (SMD)	R _{CE} Factory setting: not equipped	BOT
C260	1206 (SMD)	C _{CE} Factory setting: not equipped	BOT

Tabla 3.4 – Ajuste de la protección dinámica contra cortocircuito

Por ejemplo, para la primera rama, el primer transistor si tendrá activada esta protección, el segundo y el tercero no y el último sí. Para ello, como cada driver nos controla dos transistores los debemos configurar de la siguiente manera:

Como en todos los drivers el transistor TOP será S1 o S2 y el transistor BOT será S11 o S22, habrá que desactivar esta funcionalidad en el lado del transistor BOT y montar la resistencia y el condensador en el lado del transistor TOP.

RESISTENCIA SERIE DE COLECTOR

Esta resistencia, denominada R_{VCE} es la resistencia comentada anteriormente, y que solo debemos montar cuando nuestros transistores sean de 1700V. Como los nuestros son de 1200V habrá que cortocircuitar esta posición como nos indican en la siguiente tabla.

R_{VCE}		
Designation		Setting
R150	MiniMELF (SMD)	R_{VCE}^* Factory setting: not equipped TOP
R250	MiniMELF (SMD)	R_{VCE}^* Factory setting: not equipped BOT

* 1200V IGBT operation: 0 Ω
1700V IGBT operation: 1k Ω / 0,4W

Tabla 3.5 – Ajuste de la resistencia serie de colector

Como vemos en la tabla debemos cortocircuitar los huecos R150 y R250 de todas las placas de adaptación.

RESISTENCIA DE PUERTA

Esta resistencia es un parámetro crítico en el circuito, especialmente para las conmutaciones, un valor pequeño hace unas respuestas más rápidas, pero con un aumento de dv/dt y di/dt , con una mayor corriente absorbida por el driver. El tiempo de subida y bajada se determina ajustando las resistencias en el circuito RC de la figura:

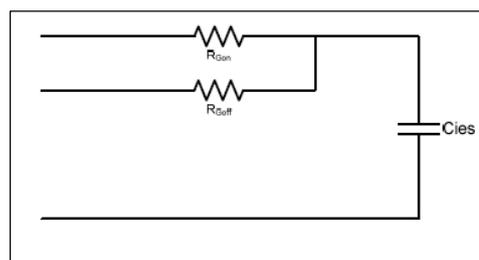


Figura 3.12 – Circuito eléctrico para las resistencias de puerta

El valor del condensador en nuestro caso es de 4,7nF como podemos ver en la hoja de características de nuestros transistores:



C_{ies}	$V_{CE} = 25, V_{GE} = 0 V$	$f = 1 MHz$	4,7	nF
C_{oes}			0,3	nF
C_{res}			0,145	nF

Tabla 3.6 – Valor del condensador C_{ies}

Y el valor de los tiempos de encendido y apagado son 34ns y 65ns respectivamente como vemos también en la hoja de características de los transistores.

$t_{d(on)}$	$R_{Gon} = 4 \Omega$ $di/dt = 4100 A/\mu s$	$V_{CC} = 300V$ $I_C = 75A$	97	ns
t_r			34	ns
E_{on}	$R_{Goff} = 4 \Omega$ $di/dt = 4100 A/\mu s$	$T_J = 150 \text{ }^\circ C$ $V_{GE} = -7/+15V$	1,7	mJ
$t_{d(off)}$			339	ns
t_f			65	ns
E_{off}			2,8	mJ

Tabla 3.7 – Valor de los tiempos de encendido y apagado

Considerando que un circuito RC determina una constante de tiempo τ dada por $R \cdot C$. El valor del condensador lo encontramos en el datasheet y aproximando los tiempos de subida y bajada como 5τ podemos determinar las resistencias. Obtenemos que:

$$\tau_r = \frac{34}{5} = 6.8, \tau_f = \frac{65}{5} = 13, R_{GON} = \frac{C_{ies}}{\tau_r} = \frac{4.7}{6.8} = 0.7\Omega, R_{GOFF} = \frac{C_{ies}}{\tau_t} = 0.36\Omega.$$

Ecuación 5 – Valor de las resistencias de puerta en función de los tiempos de encendido y apagado

Sin embargo tenemos que tener en cuenta que las resistencias no solo están determinadas por las características del transistor. Hay que considerar el valor de la corriente de pico proporcionada por el driver como restricción.

El driver tiene un voltaje de salida de 15V cuando enciende el transistor y de 7V cuando le apaga. Hay una diferencia de 22V. A pesar de que la corriente máxima que puede proporcionar es de 15A, esta etapa diseñada no necesitará tanta corriente. Se considerará que el valor máximo será de un tercio de 15A, es decir de 5A.

$$R_{Gmin} = \frac{V_{ON} - V_{OFF}}{I_{pkmax}} = \frac{22}{5} = 4.4\Omega$$

Ecuación 6 – Valor teórico de la resistencia de puerta



Designation	Shape	Setting	
R151, R152, R153 (parallel connected)	MiniMELF (SMD)	R_{Gon} Factory setting: not equipped	TOP
R154, R155, R156 (parallel connected)	MiniMELF (SMD)	R_{Goff} Factory setting: not equipped	TOP
R251, R252, R253 (parallel connected)	MiniMELF (SMD)	R_{Gon} Factory setting: not equipped	BOT
R254, R255, R256 (parallel connected)	MiniMELF (SMD)	R_{Goff} Factory setting: not equipped	BOT

Tabla 3.8 – Ajuste de la resistencia de puerta

A pesar de que existen tres huecos para soldar tres resistencias de montaje superficial en paralelo, también existe un conector para poder montar una resistencia de encapsulado normal, de valor equivalente al de las tres resistencias en paralelo. Por este motivo hemos elegido en montar resistencias de $4,3\Omega$ de pequeña tolerancia (1%) en las posiciones de la resistencia de encendido y apagado.

APAGADO SUAVE

Esta funcionalidad hace referencia a que cuando se produce un cortocircuito el driver se encarga de añadir esta resistencia para disminuir la corriente de pico que atraviesa el transistor. Esta función no se implementará y se deberá de conectar un puente en los huecos R160 o R161 y R260 o R261 como podemos observar en la siguiente tabla. Solo será necesario conectar un puente por transistor ya que ambas resistencias están en paralelo

Designation	Shape	Setting	
R160, R161 (parallel connected)	MiniMELF (SMD)	R_{Goff_sc} Factory setting: not equipped	TOP
R260, R261 (parallel connected)	MiniMELF (SMD)	R_{Goff_sc} Factory setting: not equipped	BOT

Tabla 3.9 – Ajuste de la resistencia para el apagado suave

SENSOR DE TEMPERATURA

Como nuestro circuito no tendrá un sensor de temperatura debemos conectar las tres resistencias como se muestra.



Dimensioning OTP

If no temperature sensor is connected:

- R172: 0Ω (factory setting: not equipped)
- R175: not equip (factory setting: equipped)
- R177: not equip (factory setting: not equipped)

Tabla 3.10 – Resistencia para el sensor de temperatura

En el hueco R172 debemos conectar un puente ya que no hay una resistencia instalada por defecto, en el hueco R175 tenemos que quitar la resistencia que hay instalada ya que tenemos que dejar esta posición al aire, y el hueco R177 debemos dejarlo al aire, que es como viene por defecto.

En la Figura 3.13 podemos ver cómo queda el driver una vez se han desoldado, soldado y puentado todos los elementos necesarios.

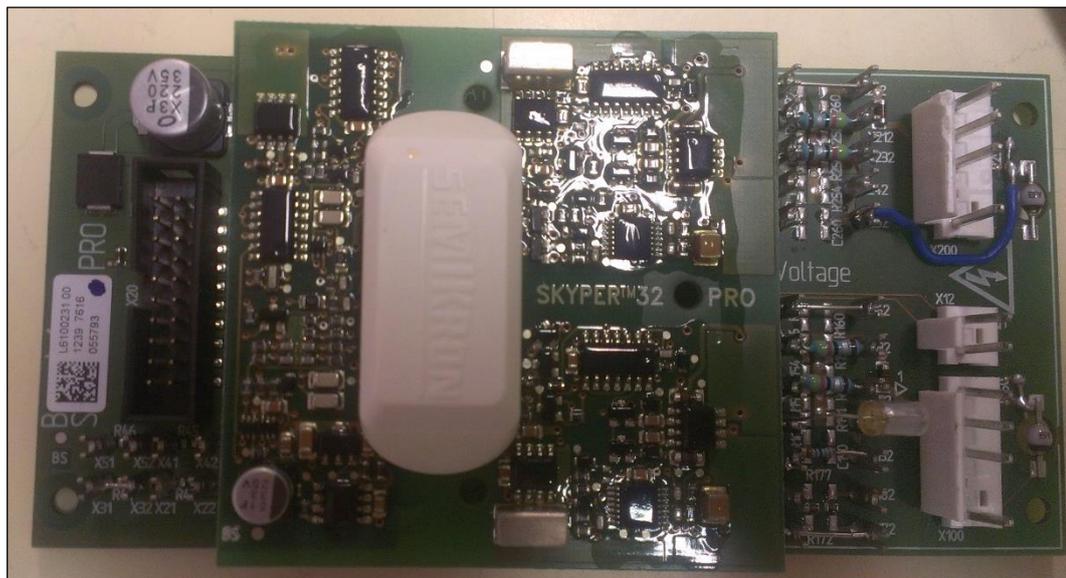


Figura 3.13 – Placa driver de potencia



Figura 3.14 – Consumo placa adaptación de niveles con driver de potencia

En la Figura 3.14 podemos ver el consumo de las dos placas de adaptación de niveles junto a los seis drivers de control. La alimentación será de 5V, y consumirá una corriente de 3.52A,

es decir será necesario proporcionar una potencia de 17.65W, aunque este consumo podrá aumentar hasta 20W cuando se aumenta la frecuencia.

3.4. DISEÑO DE CABLES DE INTERCONEXIÓN DE LA TARJETA DE ADAPTACIÓN DE NIVELES CON DSP DE CONTROL Y DRIVER DE CONTROL

Los cables de interconexión se han diseñado para que las señales vayan de la forma más directa posible desde un extremo a otro. Para ello se han configurado las salidas del DSP de tal modo que en un flanco tenemos las señales de los polos S2 y S11 de cada rama, estas seis señales se llevan a la placa de adaptación de niveles número 1, se obtienen las salidas para tres drivers. Del otro flanco se obtienen las señales de los polos S1 y S22 las cuales se llevan a la tarjeta de adaptación de niveles número 2. En total tenemos 6 cables, los cuales están explicados en páginas posteriores.

En la Figura 3.5 podemos ver un esquema de estas interconexiones, así como las señales que irán en cada cable.

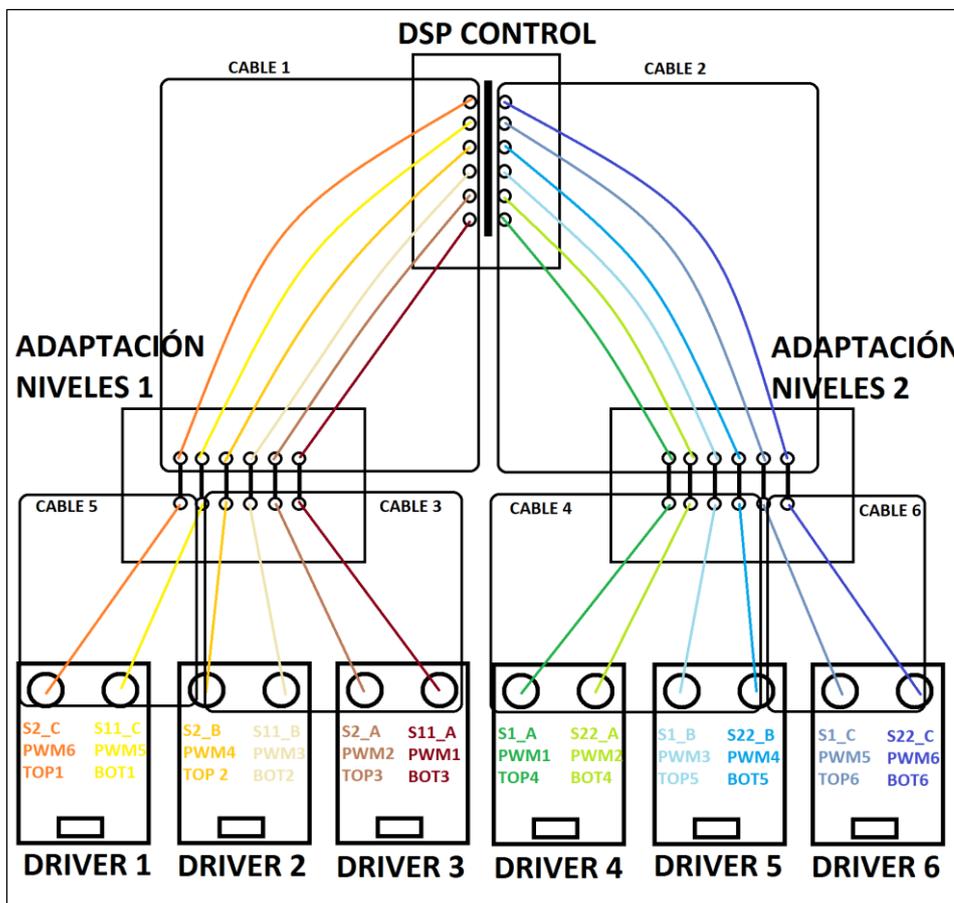


Figura 3.15 – Interconexión de placas de hardware



El cable número 1, lleva las ondas PWM de 6 polos de potencia y una señal de GND, desde el DSP de control hasta la placa de adaptación de niveles, es decir, las señales PWM tendrán valor 0V o 3,3V. El cable número 2 lleva las señales de los 6 polos restantes al mismo voltaje. En la Figura 3.16 podemos ver las señales que lleva el cable 1, a qué pines están conectadas cada una de esas señales, y a qué señal PWMx de la placa de adaptación de niveles hace referencia.

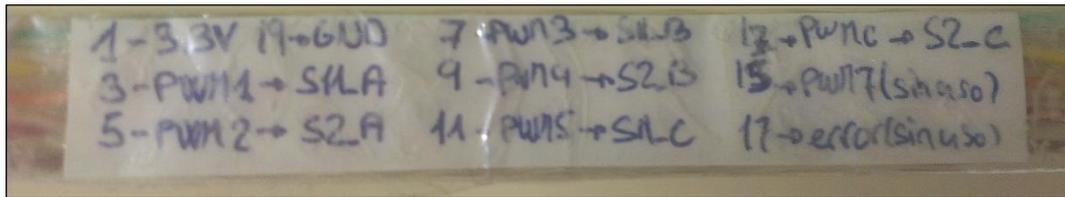


Figura 3.16 - Pegatina Cable 1



Figura 3.17 - Extremo 1 del cable 1

Los dos extremos de este cable los podemos ver en las dos siguientes figuras, en la Figura 3.17 vemos el extremo que está conectado a la placa de adaptación de niveles, y en la Figura 3.18 vemos el extremo conectado al DSP, en este extremo el cable suelto será el que lleve la señal GND.

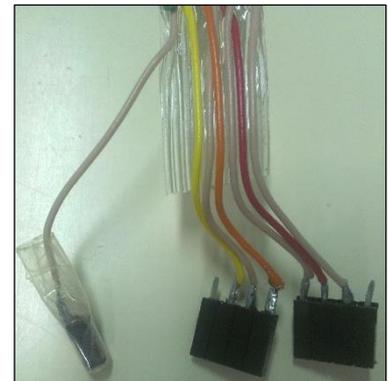


Figura 3.18 - Extremo 2 del cable 1

Los cables 3 y 4 estarán diseñados de la misma forma, aunque cada cable llevará señales diferentes. Por ejemplo, la pegatina que muestra las señales del cable 3 la podemos ver en la Figura 3.19.

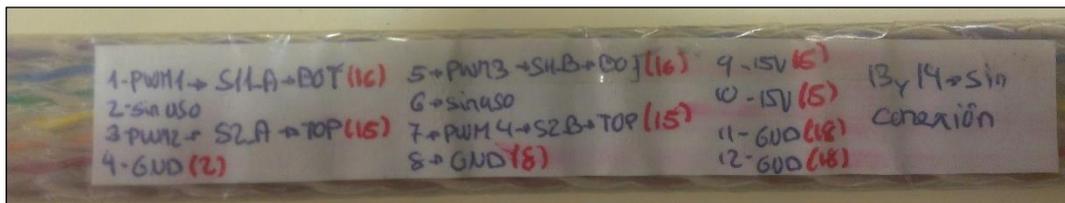


Figura 3.19 - Pegatina Cable 3

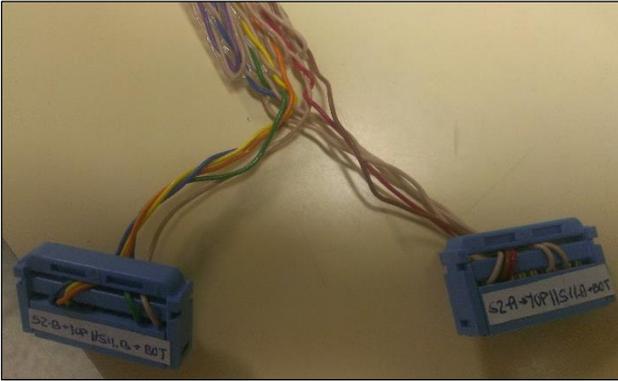


Figura 3.20 - Extremo 2 del cable 3

En la Figura 3.20 podemos ver los dos conectores que salen de este cable para conectarlos en los drivers de control de la potencia. Como podemos ver, al igual que en los casos anteriores, en cada conector aparecen las señales que lleva cada cable.

Por último mostraremos la pegatina del cable 5, que a su vez este cable estará diseñado de la misma forma que el cable 6, ya que éstos únicamente llevarán las señales de control de dos polos, que se controlan a través de un único driver de control. Podemos ver la pegatina del cable 5 en la Figura 3.21.

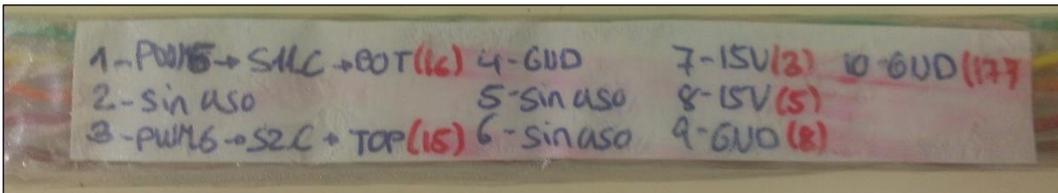


Figura 3.21 - Pegatina Cable 5

Como podemos ver este cable es de 5x2, a diferencia del cable 3 que es de 7x2, ya que uno lleva dos señales PWM y otro 4 señales PWM.



Figura 3.23 - Extremo 2 del cable 5

Los dos extremos de este cable se ven en las Figuras 3.22 y 3.23.



Figura 3.22 - Extremo 1 del cable 5

En cuanto a los conectores se ha seguido un orden para la conexión de los cables, los extremos azules son los conectados al driver de control y los extremos grises y negros son los conectados a la placa de adaptación de



niveles. Los conectores negros llevarán las señales de salida y los grises las señales de entrada.

En la Figura 3.24 podemos ver cómo quedan las interconexiones de los cables durante la etapa de ensayos.

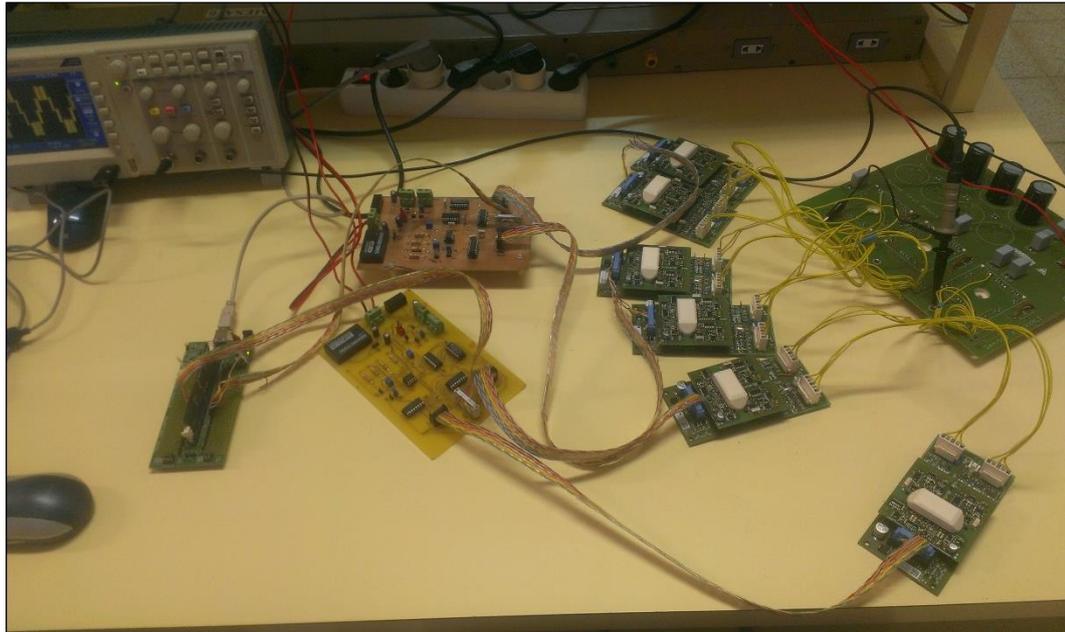


Figura 3.24 - Interconexión en la etapa de pruebas





4. ENTORNO DE PROGRAMACIÓN

Se realizarán dos programas de control capaces de conmutar los polos de potencia para ver los resultados deseados. En primer lugar se hará una programación en un entorno Code Composer Studio V5.5 y posteriormente otro código desarrollado en lenguaje de Simulink.

4.1. CONFIGURACIÓN CODE COMPOSER STUDIO Y SIMULINK

Para realizar esta configuración se necesitará tener instalados en el ordenador en el que se desee trabajar tres programas:

- Code Composer Studio V5.x
- Code Composer Studio V4.x
- Software Control Suite de Texas Instruments
- Matlab R2013b o superior (en este caso Matlab R2014a)

He de comentar que la versión de Code Composer Studio V4.x no es necesaria aunque sí recomendable ya que no se ha probado la configuración sin tener instalada esa versión.

En primer lugar abrimos el programa Code Composer Studio V5.x y configuramos una nueva tarjeta. Para ello en el menú denominado “Target Configuration” debemos seleccionar añadir tarjeta. Se nos abrirá una ventana donde podemos configurar la tarjeta nueva. Tenemos que seleccionar el tipo de conexión que usaremos con nuestra tarjeta, que en nuestro caso es “Texas Instruments XDS100v1 USB Emulator” y como tarjeta seleccionaremos TMS320F28069. Pulsaremos sobre el botón Save. Probaremos que la conexión se ha realizado correctamente pulsado sobre el botón test conexión. Si todos estos pasos se han realizado de forma correcta el test debería de devolver el mensaje de test realizado satisfactoriamente. En la Figura 4.1 podemos ver nuestra pantalla de configuración, y en la Figura 4.2 el resultado satisfactorio del test de conexión.

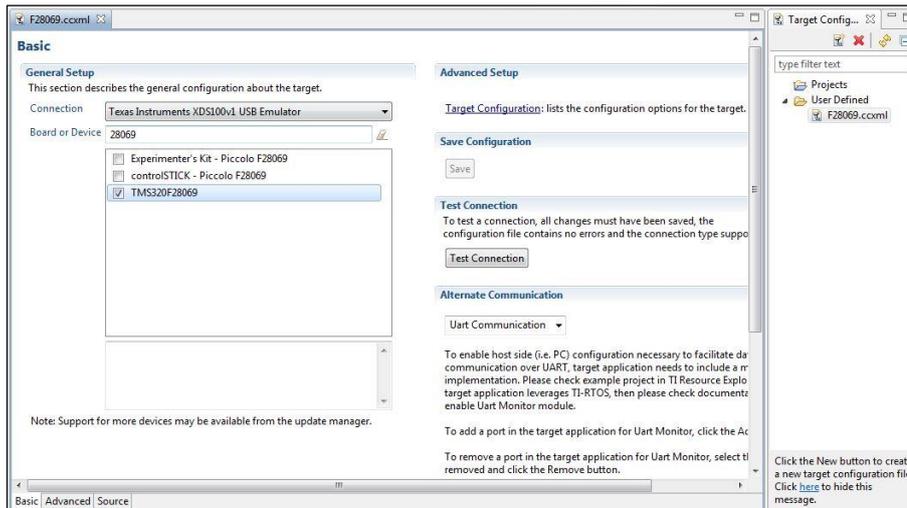


Figura 4.1 – Configuración DSP en Code Composer Studio

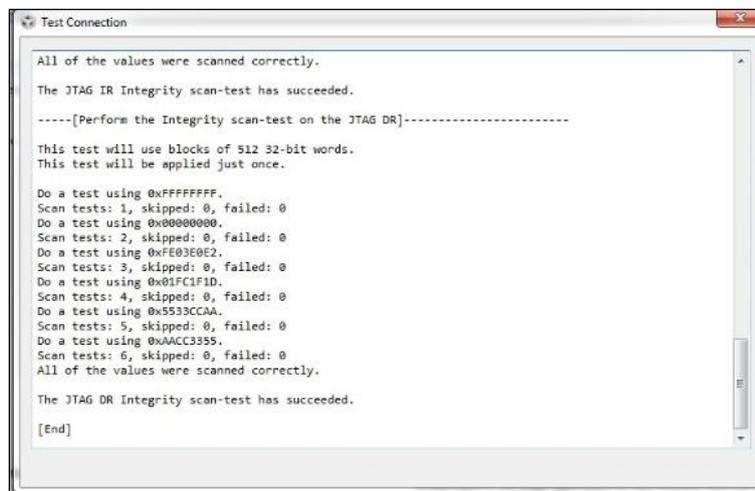


Figura 4.2 – Test de Conexión

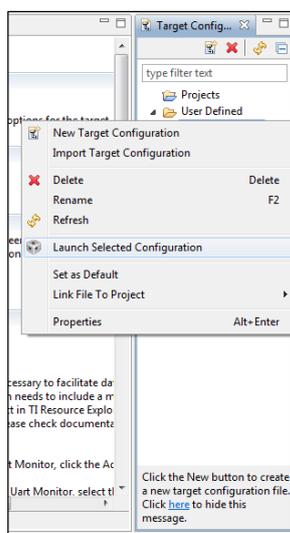


Figura 4.3 – Conexión con tarjeta DSP (1)

En algunos casos será necesario conectarse a la tarjeta antes de ejecutar el código desde Simulink, para ello pulsaremos botón derecho sobre la tarjeta añadida y seleccionaremos “Launch Selected Configuration”. Se nos abrirá la ventana de desarrollo y podremos conectarnos con la tarjeta. Podemos ver este procedimiento en la Figura 4.3. Para ello pulsaremos botón derecho sobre nuestro emulador y pulsaremos en la opción de conectarse con la tarjeta.

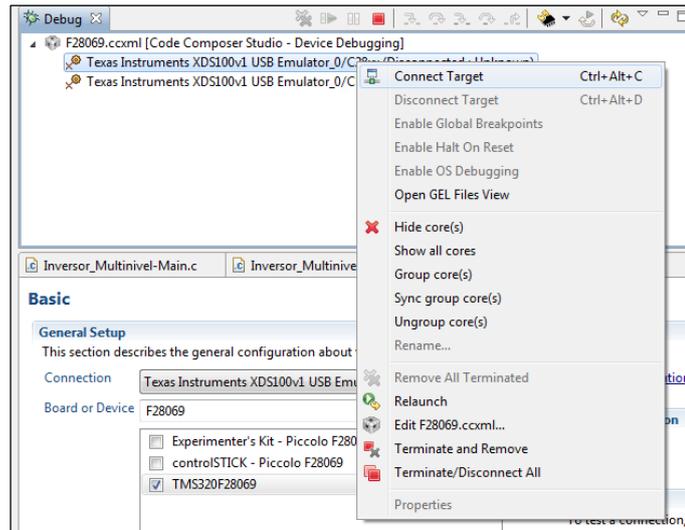


Figura 4.4 – Conexión con tarjeta DSP (2)

Una vez realizados todos estos pasos ya tendremos el programa Code Composer configurado y pasaremos a configurar el programa Matlab, en nuestro caso la versión usada será la 2014a.

En primer lugar debemos instalar el paquete de librerías y demás elementos de la familia C2000 de Texas Instruments, para ello escribiremos el comando `supportPackageInstaller`, y seleccionaremos la opción de instalar desde internet. Este proceso se muestra en la Figura 4.5.

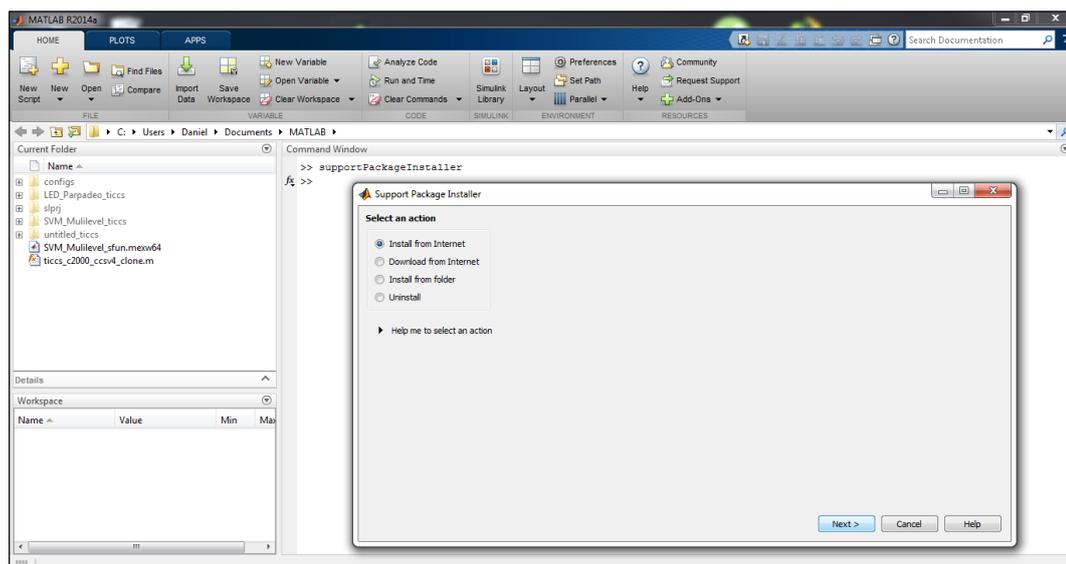


Figura 4.5 – Instalación C2000 en Matlab (1)

Una vez hayamos seleccionado esta opción nos aparecerá una tabla con los diferentes paquetes que podemos instalar en nuestra versión de Matlab,

nosotros queremos instalar el soporte de Texas Instruments para las tarjetas C2000, para ello buscaremos la opción “Texas Instruments C2000” y continuaremos el proceso de instalación, en el que necesitamos una cuenta validada de Matlab para poder iniciar sesión y descargar el paquete. También tenemos que aceptar las condiciones de uso que nos muestra Matlab.

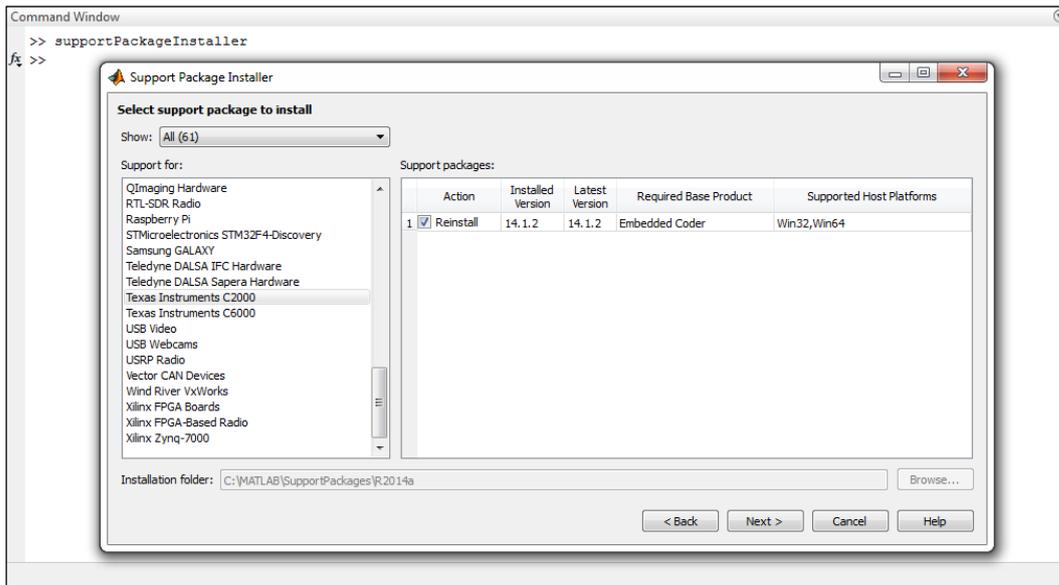


Figura 4.6 – Instalación C2000 en Matlab (2)

Durante el proceso de instalación aparecerán varias ventanas en las que tenemos que configurar las opciones seleccionadas. En la Figura 4.7 podemos ver cómo hemos configurado la versión de Code Composer Studio usada, en nuestro caso la versión usada es la 5.5.

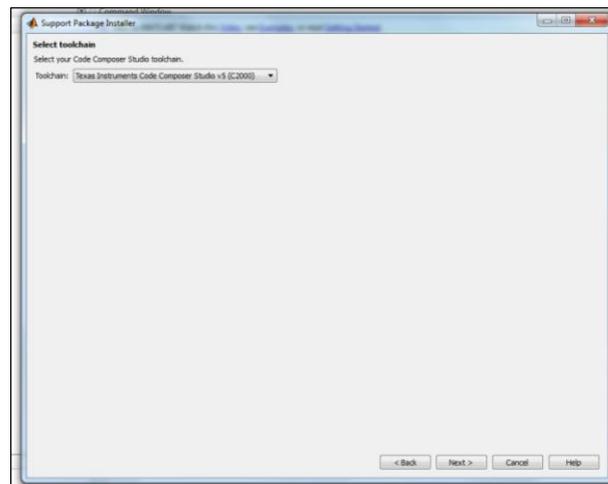


Figura 4.7 – Instalación C2000 en Matlab (3)



También nos aparecerá una ventana con los requerimientos que necesitamos para poder hacer funcionar este paquete, en nuestro caso ya los deberíamos tener instalados.

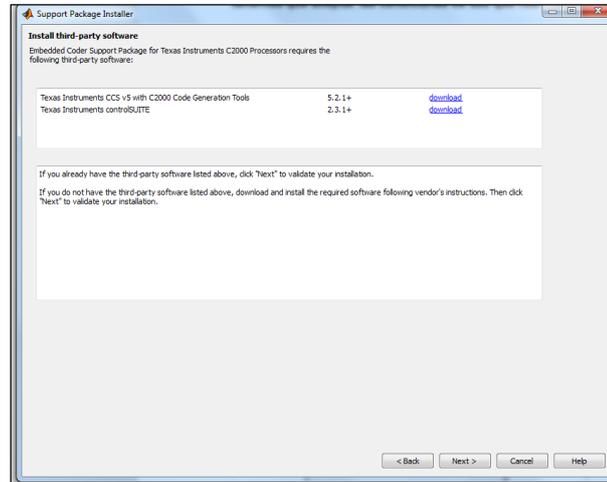


Figura 4.8 – Instalación C2000 en Matlab (4)

Por último nos pedirá la ubicación de la instalación del software “C2000 Code Generation Tools”. Nuestra ruta de instalación es la mostrada en la Figura 4.9.

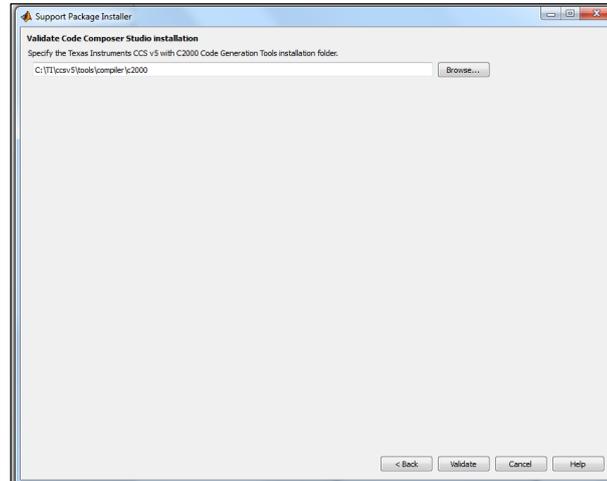


Figura 4.9 – Instalación C2000 en Matlab (5)

Una vez hecho este paso ya tendremos el paquete instalado y procederemos a configurar la forma en la que Simulink compila el programa. Para ello escribiremos el comando de Matlab `xmakefilesetup`, con el que se nos abrirá una ventana en la que se nos muestra la configuración actual.

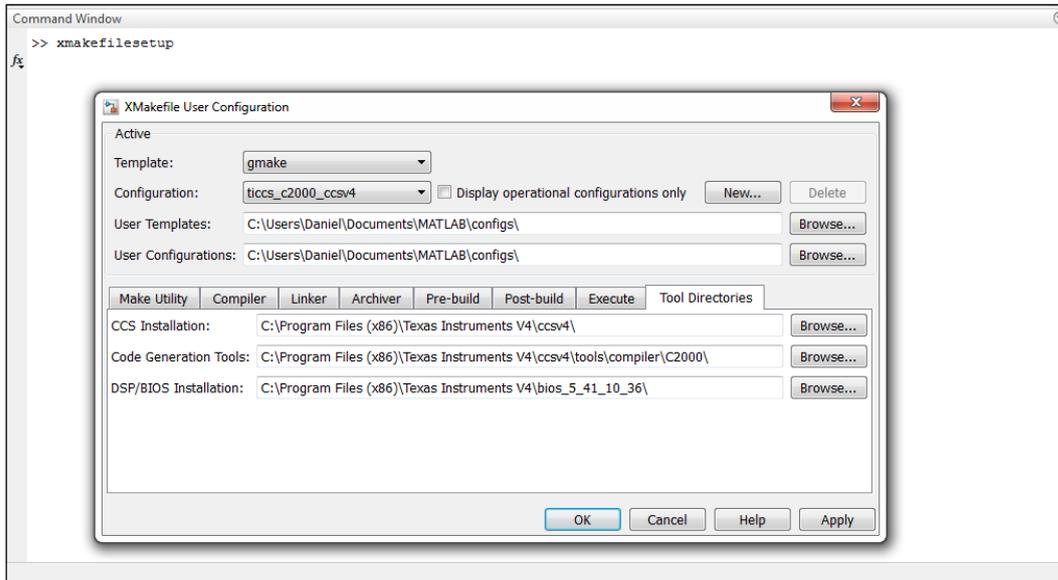


Figura 4.10 – Configuración xmakefilesetup (1)

En la ventana abierta debemos desmarcar la opción “Display operational configurations only” y seleccionar la configuración “ticcs_c2000_ccsv4”, nos moveremos a la pestaña Tool Directories y comprobaremos que los tres campos están rellenos de forma correcta.

- El campo CCS Installation hace referencia al directorio de instalación de Code Composer Studio V4.x.
- El Campo Code Genration Tools hace referencia al directorio donde tenemos instalado el compilador C2000.
- El campo DSP/BIOS Installation hace referencia al directorio donde está instaladoda la bios del programa Code Composer Studio.

Una vez rellenos estos campos debemos crear una nueva configuración, para ello en los campos User Temmplates y User Configurations seleccionaremos una carpeta localizable donde guardar la nueva configuración. Posteriormente pulsaremos sobre el botón New y elegiremos de nombre “ticcs_c2000_ccsv4_clone”, debemos asegurarnos que las pestañas Make Utility, Compiler, Linker, Archiver, Pre-Build y Post-Build están configuradas como podemos ver en las siguientes 6 figuras. La pestaña Execute la configuraremos a continuación.

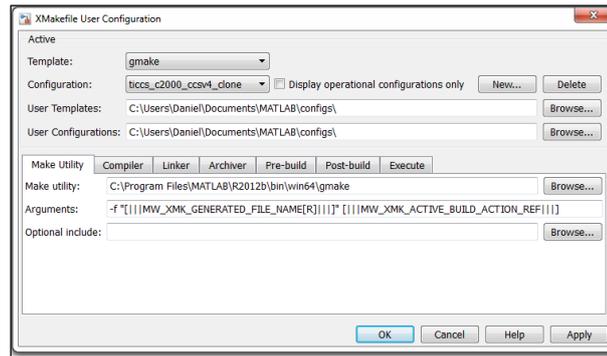


Figura 4.11 – Configuración xmakefilesetup, Make Utility

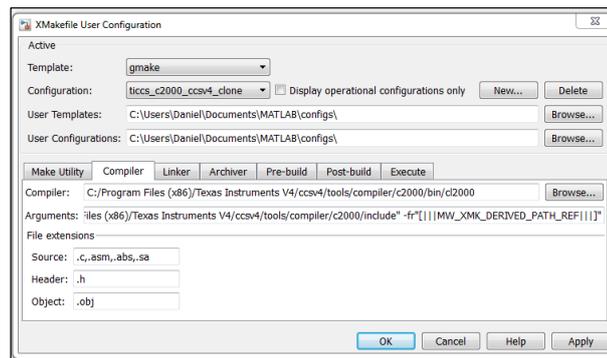


Figura 4.12 – Configuración xmakefilesetup, Compiler

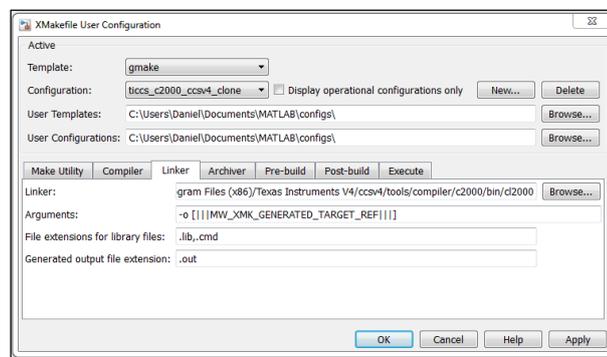


Figura 4.13 – Configuración xmakefilesetup, Linker

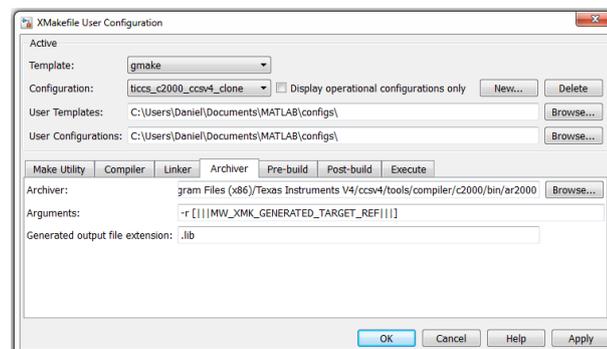


Figura 4.14 – Configuración xmakefilesetup, Archiver

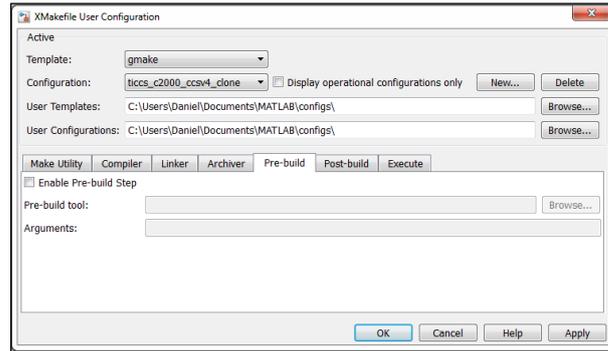


Figura 4.15 – Configuración xmakefilesetup, Pre-build

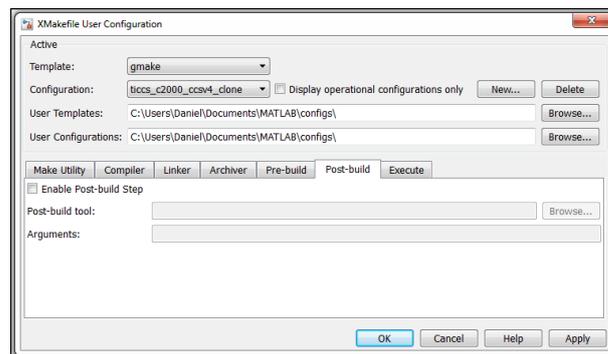


Figura 4.16 – Configuración xmakefilesetup, Post-build

La pestaña Execute, la configuraremos de la siguiente manera:

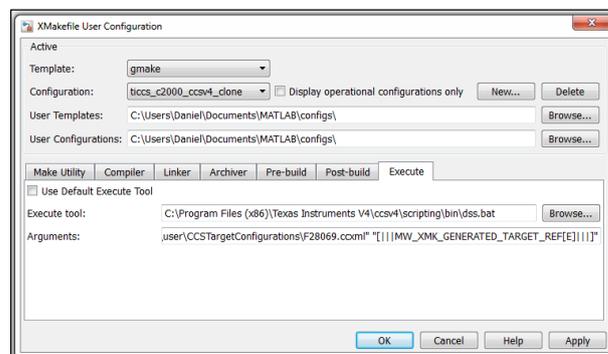


Figura 4.17 – Configuración xmakefilesetup, Execute

En “Execute Tool” debemos seleccionar el archivo mostrado. En “Arguments” debemos escribir lo siguiente:

```
"C:\Program Files\MATLAB\R2014b\toolbox\idelink\extensions\ticcs\ccsdemos\runProgram.js"  
"C:\Users\Daniel\user\CCSTargetConfigurations\F28069.ccxml"  
"[|||MW_XMK_GENERATED_TARGET_REF[E]|||]"
```



Estos tres argumentos están separados por un espacio y cada uno de ellos debe de ir entre comillas. El primer argumento hace referencia a un archivo de la toolbox de Matlab, el segundo argumento hace referencia al fichero de configuración de la tarjeta creado con anterioridad como se explicaba en la Figura 4.1, en estos dos argumentos se deberá seleccionar el directorio adecuado, y el tercer argumento hace referencia a la manera de crear el fichero makefile, este comando deberá copiarse tal cual está escrito.

Una vez configurada la manera de compilar el programa se comprobarán las referencias faltantes, para ello se ejecutará el comando `checkEnvSetup('ccsv4','f28069','check')`, y se obtendrán unos resultados parecidos a los mostrados en las dos figuras siguientes.

```

Command Window
>> checkEnvSetup('ccsv4','f28069','check')

1. CCSv4 (Code Composer Studio)
Your version      : 4.2.4
Required version: 4.0.0 to 4.2.0
Required for     : Code Generation
TI_DIR="C:\Program Files (x86)\Texas Instruments V4\ccsv4"

2. CGT (Texas Instruments C2000 Code Generation Tools)
Your version      : 6.0.1
Required version: 5.2.1 to 5.2.3
Required for     : Code generation
C2000_CGT_INSTALLDIR="C:\Program Files (x86)\Texas Instruments V4\ccsv4\tools\compiler\c2000"

3. DSP/BIOS (Real Time Operating System)
Your version      : 5.41.10.36
Required version: 5.33.05 to 5.41.09.34
Required for     : Code generation
CCSV4_DSPBIOS_INSTALLDIR="C:\Program Files (x86)\Texas Instruments V4\bios_5_41_10_36"

4. XDC Tools (eXpress DSP Components)
Your version      : 3.22.01.21
Required version: 3.16.02.32 or later
Required for     : Code generation

5. 2806x C/C++ Header Files
Your version      : 1.20
Required version: 1.20
  
```

Figura 4.18 – Comprobación Configuración (1)

```

Command Window
2. Uti (Texas Instruments C2000 Code generation tools)
Your version      : 6.0.1
Required version: 5.2.1 to 5.2.3
Required for     : Code generation
C2000_CGT_INSTALLDIR="C:\Program Files (x86)\Texas Instruments V4\ccsv4\tools\compiler\c2000"

3. DSP/BIOS (Real Time Operating System)
Your version      : 5.41.10.36
Required version: 5.33.05 to 5.41.09.34
Required for     : Code generation
CCSV4_DSPBIOS_INSTALLDIR="C:\Program Files (x86)\Texas Instruments V4\bios_5_41_10_36"

4. XDC Tools (eXpress DSP Components)
Your version      : 3.22.01.21
Required version: 3.16.02.32 or later
Required for     : Code generation

5. 2806x C/C++ Header Files
Your version      : 1.20
Required version: 1.20
Required for     : Code generation
DSP2806x_INSTALLDIR="C:\ti\controlSUITE\device_support\2806x\v120"

6. Flash Tools (TMS320F2806x Piccolo(TM) Flash API)
Your version      : 1.00
Required version: 1.00
Required for     : Flash Programming
FLASH_2806X_API_INSTALLDIR="C:\ti\controlSUITE\libs\utilities\flash_api\2806x\v100"
  
```

Figura 4.19 – Comprobación Configuración (2)

En el caso de que algunas referencias estuvieran vacías, como vemos en la Figura 4.20, se deberán crear unas variables de sistema.

```
2. CGT (Texas Instruments C2000 Code Generation Tools)
Your version      :
Required version: 5.2.1 to 5.2.3
Required for     : Code generation
C2000_CGT_INSTALLDIR=""
```

Figura 4.20 – Configuración Vacía

En este caso vemos como C2000_CGT_INSTALLDIR no hace referencia a ninguna ubicación. Por lo que debemos crear una variable de sistema, para ello tenemos que dirigirnos a propiedades del sistema, seleccionar del menú izquierdo la opción Configuración avanzada del sistema, en la ventana abierta seleccionar la opción Variables de entorno, y en la nueva ventana abierta seleccionar el botón Nueva de la parte inferior que hace referencia a Variables del sistema. Para crear la nueva variable en el campo Nombre de la Variable pegaremos el nombre de Matlab que está sin referencia, en este caso C2000_CGT_INSTALLDIR, y en el campo Valor de la variable pegaremos la ubicación del directorio seleccionado.

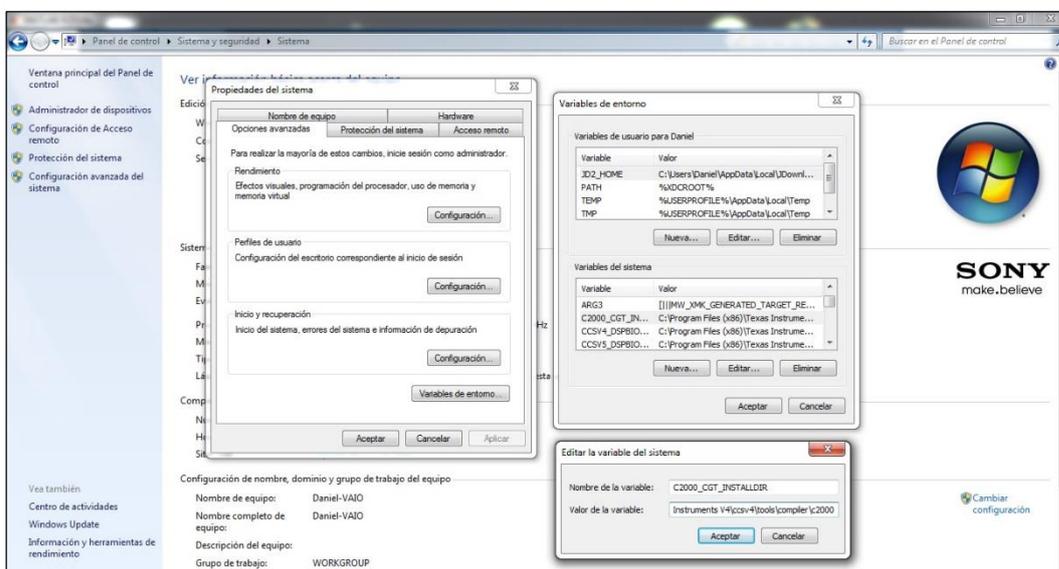


Figura 4.21 – Añadir Variable de Sistema

Los directorios que hay que seleccionar se pueden ver en las Figuras 4.18 y 4.19. Este paso hay que repetirlo para todas las variables que no tengan asociada una referencia al ejecutar la comprobación.



Una vez hayamos configurado todas estas variables volveremos a ejecutar el comando `checkEnvSetup('ccsv4','f28069','check')`, para asegurarnos de que todo se ha configurado de forma correcta.

Por último solo nos queda configurar el entorno de programación de Simulink. Para ello pulsando sobre el botón de configuración de Simulink debemos seleccionar las siguientes opciones:

El modo de resolución debe de ser de paso fijo y discreto, configurado como se muestra en la figura siguiente.

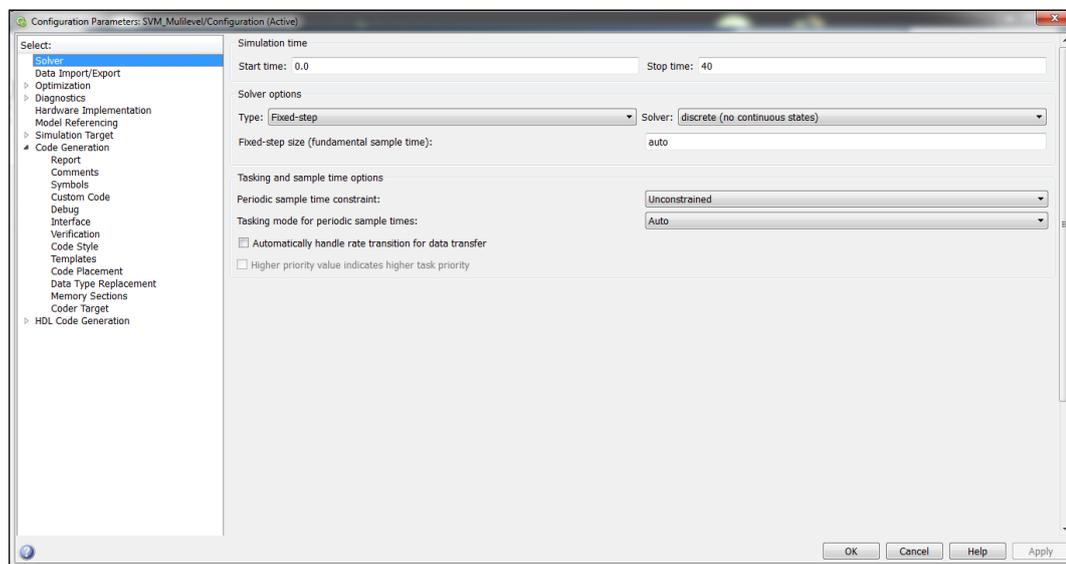


Figura 4.22 – Configuración Simulink (1)

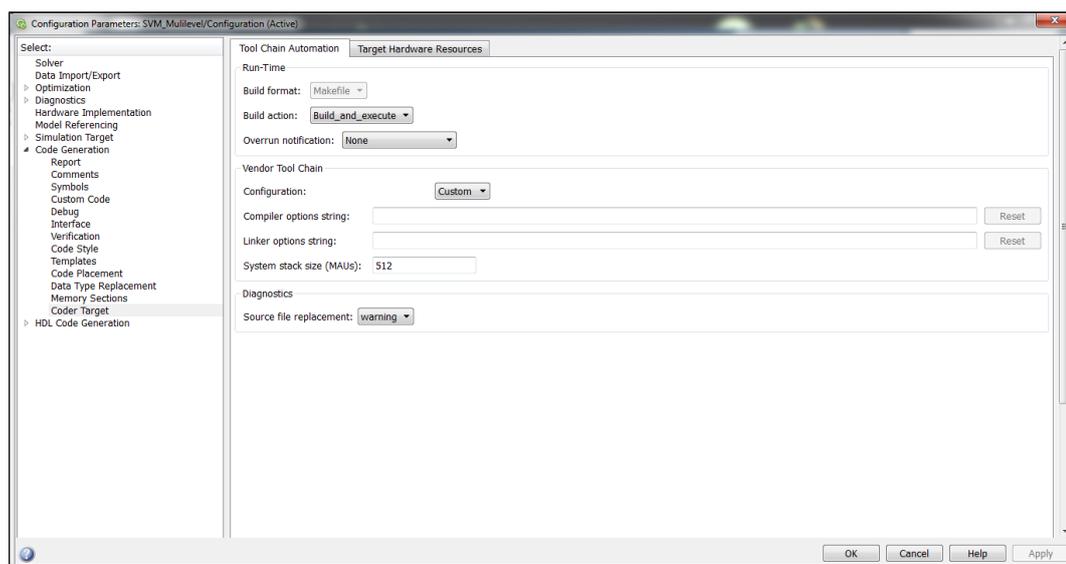


Figura 4.23 – Configuración Simulink (2)

Posteriormente seleccionaremos el menú Code Generation y dentro de este menú el submenú Coder Targer, como podemos ver en la Figura 4.23. Hay que configurarlo tal y como muestra la Figura.

Finalmente seleccionaremos la pestaña “Targer Hardware Resources”, y configuraremos la versión de Code Composer Usada, en este caso seleccionaremos la versión 4, ya que toda la configuración la hemos realizado sobre esa versión; la tarjeta usada, en nuestro caso C2000_Custom; y el procesador DSP usado, en nuestro caso F28069 como habíamos comentado.

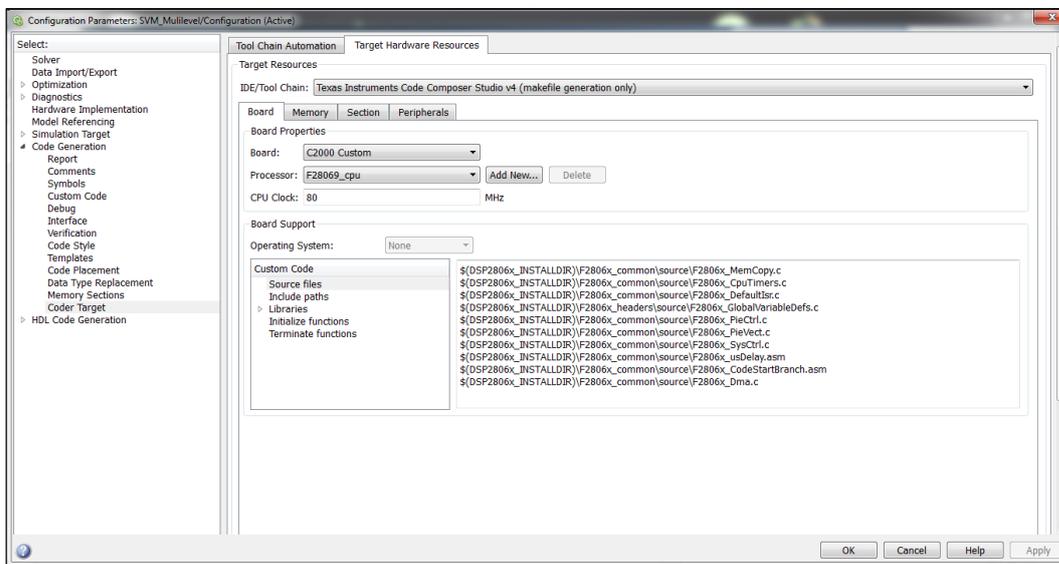


Figura 4.24 – Configuración Simulink (4)

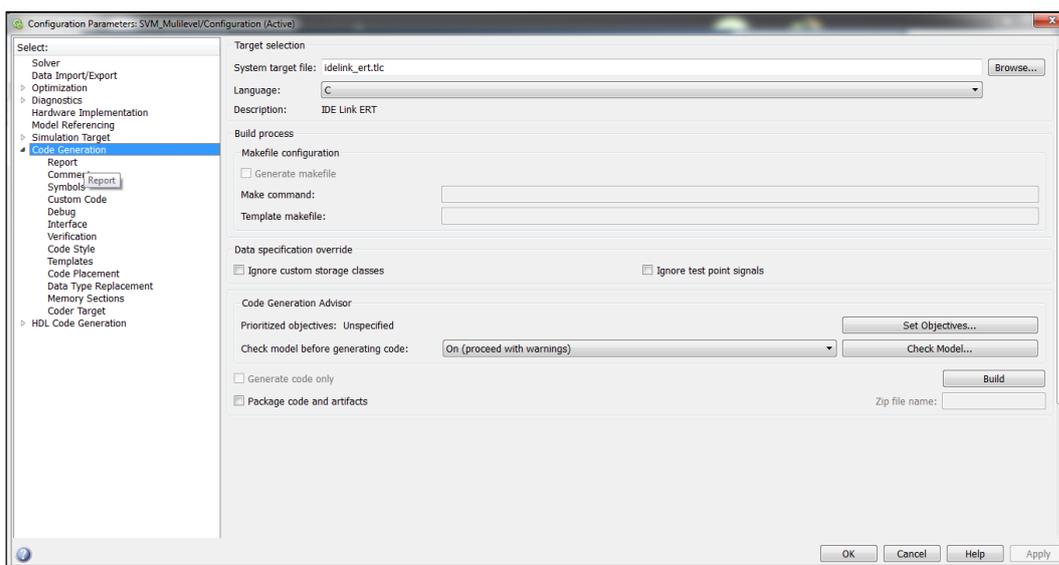


Figura 4.25 – Configuración Simulink y Ejecución del Programa



Por último y para compilar y ejecutar el programa en nuestra tarjeta debemos pulsar sobre el botón Build del menú Code Generation. Después de pulsar este botón lo más seguro es que nos compile correctamente, se conecte a la tarjeta y sea capaz de ejecutar el programa en el DSP de control, si esto es así obtendremos una ventana como la mostrada en la Figura 4.26.

```
C:\Windows\system32\cmd.exe
U\SUM_Mullevel.out ./derived/MW_c28xx_board.obj ./derived/MW_c28xx_csl.obj ./de
rievd/MW_c28xx_pun.obj ./derived/SUM_Mullevel1.obj ./derived/SUM_Mullevel1_data
.obj ./derived/SUM_Mullevel1_data.obj ./derived/rt_nonfinite.obj ./derived/F2806x_CodeStartBranch.obj ./derived/F28
06x_CpuTimers.obj ./derived/F2806x_Interrupter.obj ./derived/F2806x_Dma.obj ./der
ived/F2806x_HemCopy.obj ./derived/F2806x_FlashCtrl.obj ./derived/F2806x_Flashed.ob
j ./derived/F2806x_SysCtrl.obj ./derived/F2806x_usDelay.obj ./derived/F2806x_Glo
balUserInterface.obj ./SUM_Mullevel1.cmd /.../.../MATLAB\SupportPackages/R2
014a/tic2800/rtlib/Iqmath_fpu32.lib
CLinking>
"Evaluating build target..."
"Evaluating execute target..."
"C:\Program Files (x86)\Texas Instruments\U4\ccscd\scripting\bin\des.bat" "C:\Pr
ogram Files\MATLAB\R2012b\toolbox\ide\link\extensions\ticcs\ccsdemos\runProgram.js
" "C:\Users\Daniel\user\CCSTargetConfigurations\F28069.ccxml" ".\Custom\MW\SUM_M
ullevel1.out"
**** User inputs:
Target Configuration: C:\Users\Daniel\user\CCSTargetConfigurations\F28069.cc
xml
Program Name: ./Custom\MW\SUM_Mullevel1.out
Board Name: unspecified (connect to first board)
CPU Name: unspecified (connect to first CPU)
**** Starting debug session...
**** Debug Session Name: Texas Instruments XDS1000 USB Emulator_0/C28xx
**** Board Name: Texas Instruments XDS1000 USB Emulator_0
**** CPU Name: C28xx
**** Connecting to target...
**** Loading the program to the target...
**** Program is running.
**** Disconnecting from target...
**** Terminating debug session...
**** LOAD & RUN DONE.
C:\Users\Daniel\Documents\MATLAB\SUM_Mullevel1_ticcs_
```

Figura 4.26 - Ejecución correcta de la compilación de programa en Simulink

Sin embargo también he tenido casos en los que se ha producido un error, ya que el programa ha compilado pero no ha sido capaz de conectar con la tarjeta. Este error se da si se ha estado trabajado antes con Code Composer Studio. La ventana que veremos en este caso está mostrada en la Figura 4.27.

```
C:\Windows\system32\cmd.exe
**** Connecting to target...
SEVERE: C28xx: Error connecting to the target: (Error -154 @ 0x0) This utility fa
iled to use the adapter for a custom emulator to perform a write to a controlle
r register. The adapter returned an error. (Release 5.0.429.0)
SEVERE: emulation failure occurred
SEVERE: Error connecting to the target: emulation failure occurred
SEVERE: Cannot perform operation, target is not connected.
SEVERE: Error resetting target: Cannot perform operation, target is not connecte
d.
org.mozilla.javascript.WrappedException: Wrapped com.ti.ccestudio.scripting.envir
onment.ScriptingException: Error resetting target: Cannot perform operation, targ
et is not connected. (C:\Program Files\MATLAB\R2012b\toolbox\ide\link\extensions\
ticcs\ccsdemos\runProgram.js@93)
    at org.mozilla.javascript.Context.throwSriptRuntimeEx<<Context.java:17
95>
    at org.mozilla.javascript.MemberBox.invoke<<MemberBox.java:157>
    at org.mozilla.javascript.NativeJavaMethod.call<<NativeJavaMethod.java:20
11>
    at org.mozilla.javascript.optimizer.OptRuntime.callProp0<<OptRuntime.java
1117>
    at org.mozilla.javascript.gen.c1_c0(C:\Program Files\MATLAB\R2012b\tool
box\ide\link\extensions\ticcs\ccsdemos\runProgram.js:93)
    at org.mozilla.javascript.gen.c1_call(C:\Program Files\MATLAB\R2012b\too
lbox\ide\link\extensions\ticcs\ccsdemos\runProgram.js)
    at org.mozilla.javascript.ContextFactory.doTopCall<<ContextFactory.java:3
37>
    at org.mozilla.javascript.ScriptRuntime.doTopCall<<ScriptRuntime.java:275
5>
    at org.mozilla.javascript.gen.c1_call(C:\Program Files\MATLAB\R2012b\too
lbox\ide\link\extensions\ticcs\ccsdemos\runProgram.js)
    at org.mozilla.javascript.gen.c1_exec(C:\Program Files\MATLAB\R2012b\too
lbox\ide\link\extensions\ticcs\ccsdemos\runProgram.js)
    at org.mozilla.javascript.tools.shell.Main.evaluateScript<<Main.java:500>
    at org.mozilla.javascript.tools.shell.Main.processFileSecure<<Main.java:4
22>
    at org.mozilla.javascript.tools.shell.Main.processFile<<Main.java:388>
    at org.mozilla.javascript.tools.shell.Main.processSource<<Main.java:379>
    at org.mozilla.javascript.tools.shell.Main.processFiles<<Main.java:176>
    at org.mozilla.javascript.tools.shell.Main$1Proxy.run<<Main.java:97>
    at org.mozilla.javascript.Context.call<<Context.java:540>
    at org.mozilla.javascript.ContextFactory.call<<ContextFactory.java:447>
    at org.mozilla.javascript.tools.shell.Main.exec<<Main.java:159>
    at org.mozilla.javascript.tools.shell.Main.main<<Main.java:132>
Caused by: com.ti.ccestudio.scripting.environment.ScriptingException: Error reset
ting target: Cannot perform operation, target is not connected.
    at com.ti.ccestudio.scripting.target.reset<<Target.java:618>
    at sun.reflect.NativeMethodAccessorImpl.invoke0<<Native Method>
    at sun.reflect.NativeMethodAccessorImpl.invoke<<Unknown Source>
    at sun.reflect.DelegatingMethodAccessorImpl.invoke<<Unknown Source>
    at java.lang.reflect.Method.invoke<<Unknown Source>
    at org.mozilla.javascript.MemberBox.invoke<<MemberBox.java:142>
    .. 18 more
C:\Users\Daniel\Documents\MATLAB\SUM_Mullevel1_ticcs_
```

Figura 4.27 - Error en la ejecución de la compilación del programa de Simulink

En este caso se nos habrá generado un fichero .out dentro de una carpeta llamada con el nombre del fichero de Simulink en el directorio de trabajo de Matlab. Para poder ejecutar este programa debemos conectarnos a la tarjeta a través de Code Composer Studio como se muestra en las Figuras 4.3 y 4.4. Para asegurarnos de que estamos conectados correctamente podemos ver la pestaña Disassembly en el entorno de programación Code Composer Studio.

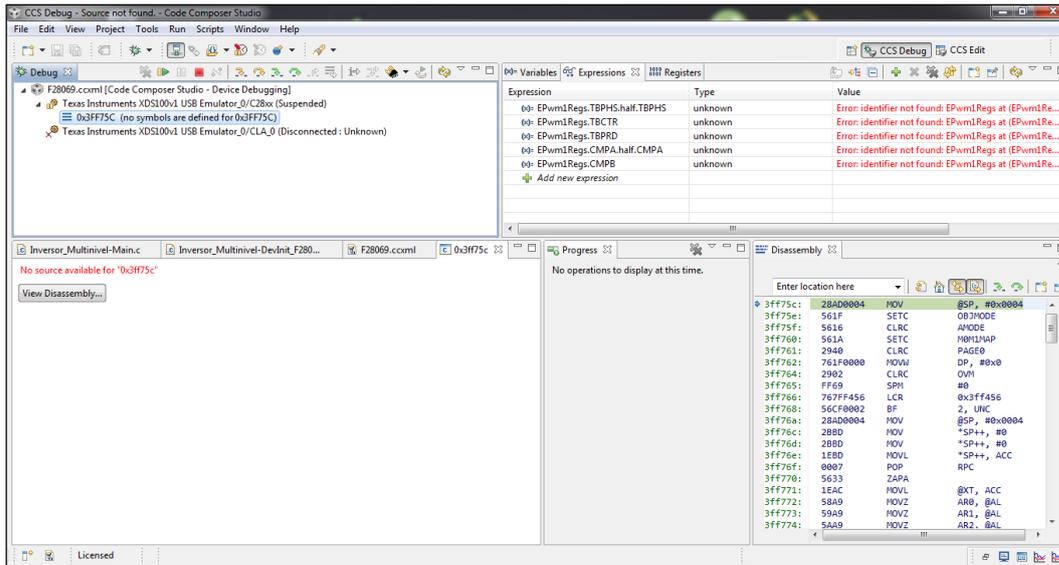


Figura 4.28 – Pestaña Disassembly con información

Una vez nos hemos asegurado de que estamos correctamente conectados con la tarjeta nos dirigiremos al menú desplegable run, seleccionaremos load, y posteriormente load program.

Se nos abrirá una nueva ventana donde tendremos que seleccionar el fichero .out mencionado anteriormente. Buscaremos este fichero con el botón Browse y aceptaremos. Una vez hecho este paso ya tendremos cargado el programa y solo tendremos que dar al botón Play para comenzar a ejecutarlo. En las dos siguientes figuras podemos ver el proceso de selección del fichero .out.

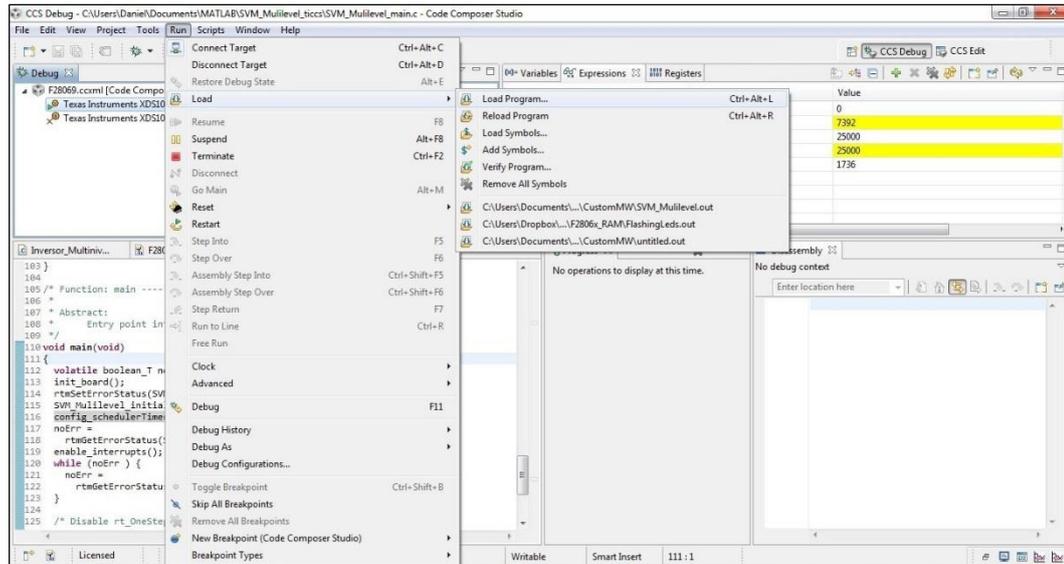


Figura 4.29 – Selección del fichero .out (1)

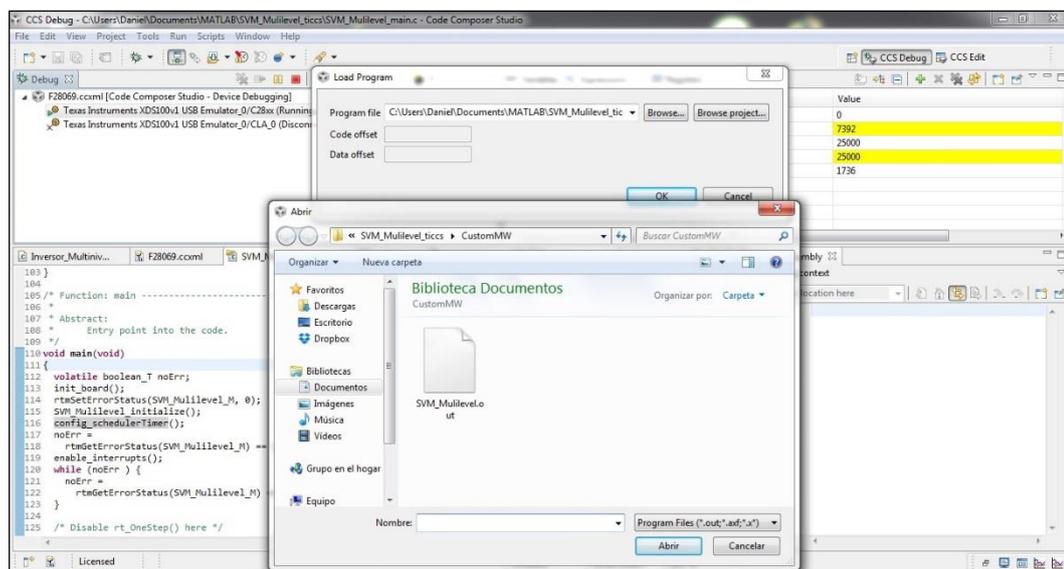


Figura 4.30 – Selección del fichero .out (2)

4.2. CÓDIGO DESARROLLO EN CODE COMPOSER STUDIO

En este software se ha desarrollado un código sencillo que se basa en una conmutación simple de los polos de potencia para obtener 5 niveles en la salida sin posibilidad de control de amplitud pero sí de control de la frecuencia.



Para ello se hacen conmutar los polos de la misma forma que se especifica en la Figura 2.2.

Como bien sabemos se deben de controlar 12 polos, por lo que será necesario activar 6 señales ePWM de Code Composer Studio, ya que cada señal ePWMx es capaz de controlar dos polos.

Para cada señal ePWM habrá que configurarla introduciendo unos parámetros. Por ejemplo, la señal ePWM2 se han configurado de la siguiente manera:

```
EPwm2Regs.TBPRD = period/2;
EPwm2Regs.TBPHS.all = 0;
EPwm2Regs.TBPHS.half.TBPHS = period/2;
EPwm2Regs.TBCTR = 0;
EPwm2Regs.TBCTL.bit.PRDL = TB_IMMEDIATE;
EPwm2Regs.TBCTL.bit.CTRMODE = TB_COUNT_UPDOWN;
EPwm2Regs.TBCTL.bit.PHSDIR = TB_DOWN;
EPwm2Regs.TBCTL.bit.PHSEN = TB_ENABLE;
EPwm2Regs.TBCTL.bit.SYNCOSEL = TB_SYNC_IN;

EPwm2Regs.CMPCTL.bit.SHDWAMODE = CC_SHADOW;
EPwm2Regs.CMPCTL.bit.SHDWBMODE = CC_SHADOW;
EPwm2Regs.CMPCTL.bit.LOADAMODE = CC_CTR_ZERO;
EPwm2Regs.CMPCTL.bit.LOADBMODE = CC_CTR_ZERO;

EPwm2Regs.CMPA.half.CMPA = period/4;
EPwm2Regs.CMPB = period/4;

EPwm2Regs.AQCTLA.bit.ZRO = AQ_CLEAR;
EPwm2Regs.AQCTLA.bit.CAU = AQ_SET;

EPwm2Regs.AQCTLB.bit.ZRO = AQ_SET;
EPwm2Regs.AQCTLB.bit.CBU = AQ_CLEAR;

EPwm2Regs.DBCTL.bit.IN_MODE = DBA_ALL;
EPwm2Regs.DBCTL.bit.OUT_MODE = DB_FULL_ENABLE;
EPwm2Regs.DBCTL.bit.POLSEL = DB_ACTV_HIC;
EPwm2Regs.DBRED = RED_delay;
EPwm2Regs.DBFED = FED_delay;
```

Cada uno de estos parámetros hace referencia a una configuración diferente:

- El periodo con `EPwm2Regs.TBPRD`, como veremos más adelante, al ser una señal triangular tenemos que seleccionar como valor la mitad del periodo deseado, ya que la señal triangular será ascendente/descendente.



- El retraso de sincronismo `EPwm2Regs.TBPHS.half.TBPHS`, en este caso se retrasará 180° ($\text{period}/2$) con respecto a la señal `ePWM1`.
- Tipo de señal triangular moduladora deseada `EPwm2Regs.TBCTL.bit.CTRMODE`, se quiere una señal triangular ascendente/descendente.
- Señal triangular ascendente o descendente cuando se sincroniza `EPwm2Regs.TBCTL.bit.PHSDIR`, se quiere que cuando se sincronice la señal triangular empiece en el flanco descendente.
- Habilitación de sincronización habilitada `EPwm2Regs.TBCTL.bit.PHSEN`, para esta señal se ha habilitado el sincronismo.
- Selección de la señal de sincronismo de salida `EPwm2Regs.TBCTL.bit.SYNCOSSEL`, en esto caso queremos que la señal de salida sea la misma que la de entrada, es decir estamos propagando la señal de sincronismo de `ePWM1`.
- Valor de comparación cuando se activarán o desactivarán la señales `ePWMA` y `ePWMb` `EPwm2Regs.CMPA.half.CMPA` y `EPwm2Regs.CMPB`
- Qué hacer cuando la señal de comparación de `ePWMA` coincide en flanco ascendente `EPwm2Regs.AQCTLA.bit.ZRO`, es decir, cuando se alcance el valor de `CMPA` en flanco ascendente se desactivará `ePWM2a`.
- Qué hacer cuando la señal de comparación de `ePWMA` coincide en flanco descendente `EPwm2Regs.AQCTLA.bit.CAU`, es decir, cuando se alcance el valor de `CMPA` en flanco descendente se activará `ePWM2a`.
- Qué hacer cuando la señal de comparación de `ePWMb` coincide en flanco ascendente `EPwm2Regs.AQCTLA.bit.ZRO`, es decir, cuando se alcance el valor de `CMPB` en flanco descendente se desactivará `ePWM2b`.
- Qué hacer cuando la señal de comparación de `ePWMb` coincide en flanco descendente `EPwm2Regs.AQCTLB.bit.CBU`, es decir, cuando se alcance el valor de `CMPB` en flanco descendente se activará `ePWM2b`.
- Activación del dead time `EPwm2Regs.DBCTL.bit.IN_MODE`, `EPwm2Regs.DBCTL.bit.OUT_MODE` y `EPwm2Regs.DBCTL.bit.POLSEL`.

- Tiempo deseado de dead time en flanco ascendente $EPwm2Regs.DBRED$.
- Tiempo deseado de dead time en flanco descendente $EPwm2Regs.DBFED$.

En la siguiente figura se pueden ver con detalles las señales triangulares asociadas a cada par de polos y la activación y desactivación de estas señales se producirán cuando las gráficas verde y azul corten a la roja. En este punto de corte la gráfica roja tendrá el valor CMPA o CMPB deseado. Además se puede ver el valor que tiene la señal roja de las diferentes ramas cuando se sincronice, y si empieza en flanco ascendente o descendente.

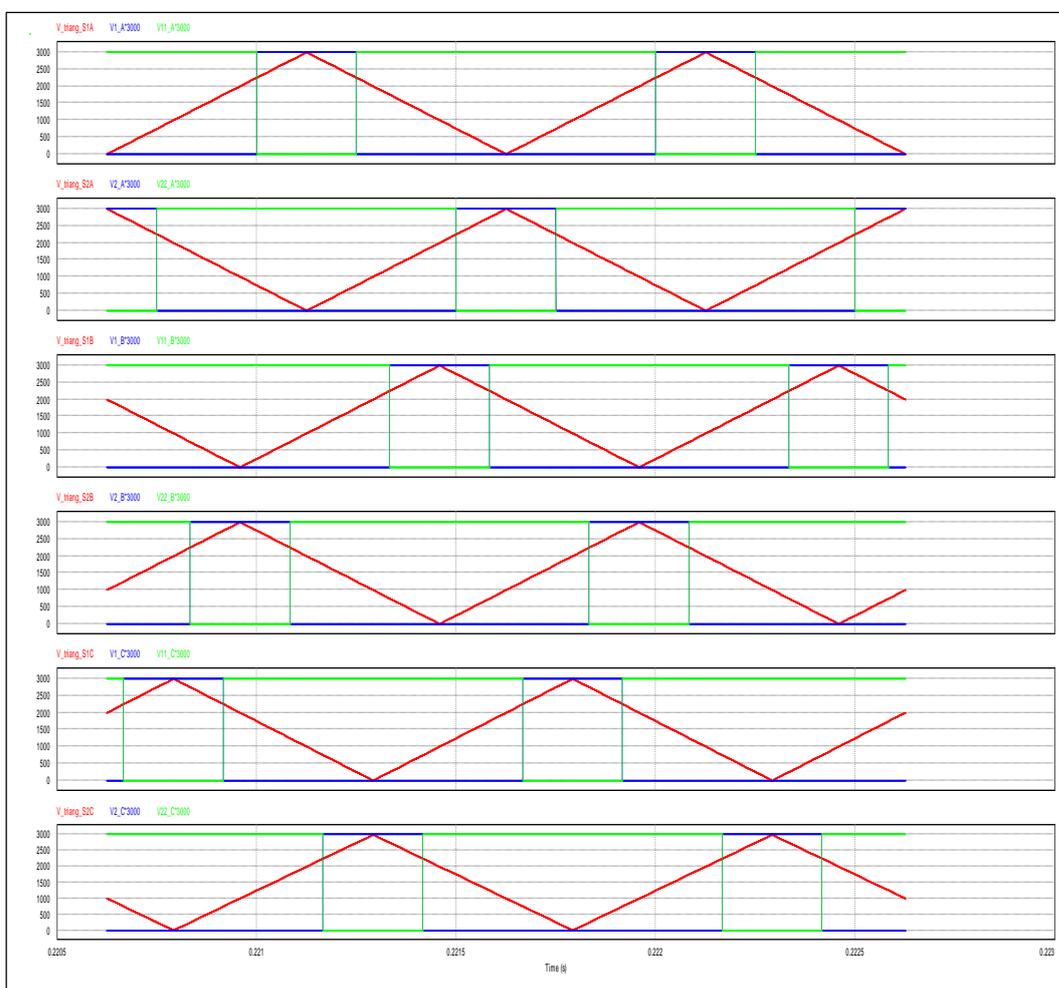


Figura 4.31 – Señales triangulares moduladoras y señales ePWMx

El código completo desarrollado en Code Composer Studio lo podemos encontrar en el Anexo 1.

El dispositivo DSP elegido cuenta con la siguiente estructura interna para generar las señales ePWMx.

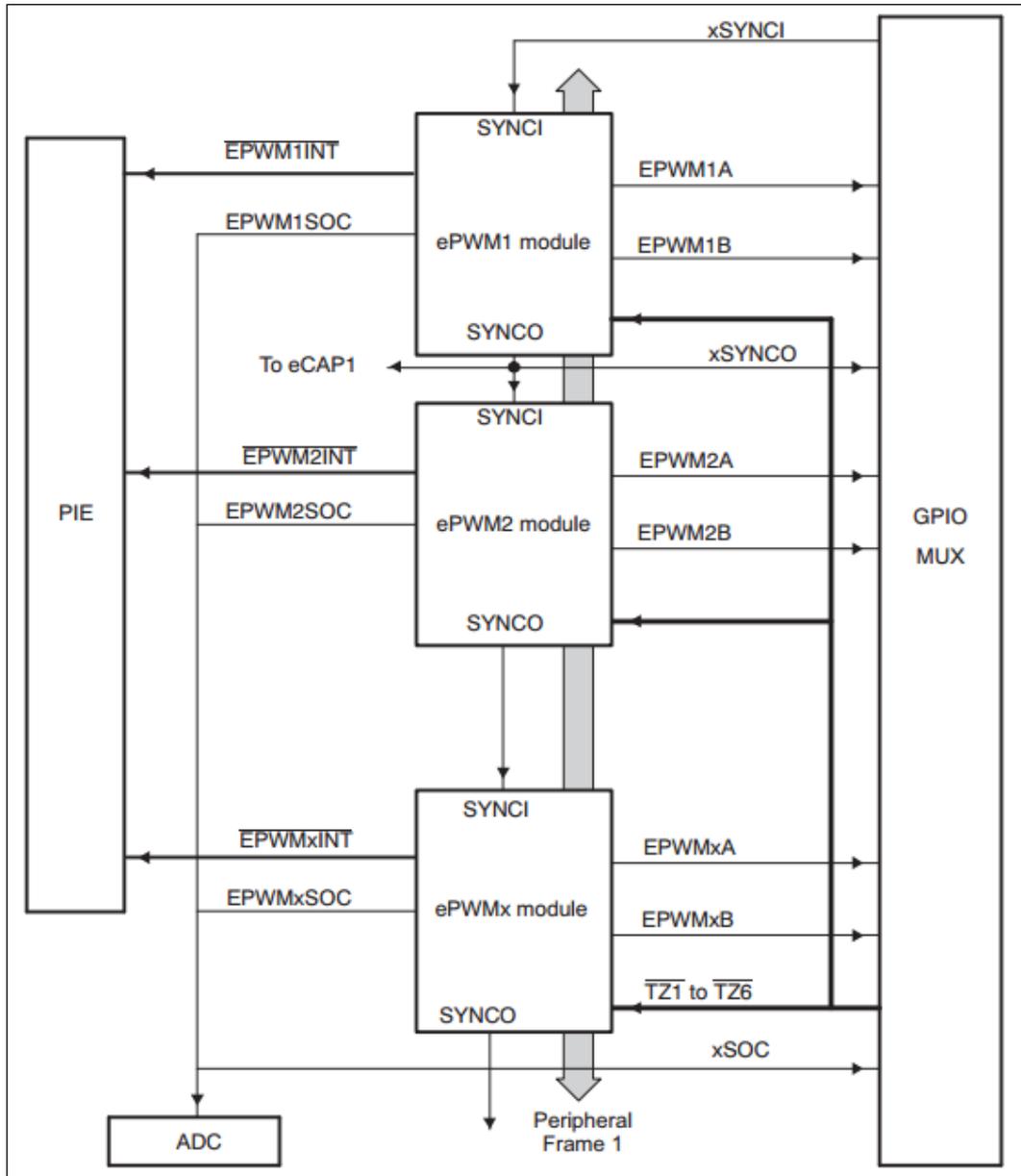


Figura 4.32 - Estructura del DSP de control para generar las señales ePWMx

Y la estructura interna de cada módulo ePWMx lo podemos ver en la siguiente figura.

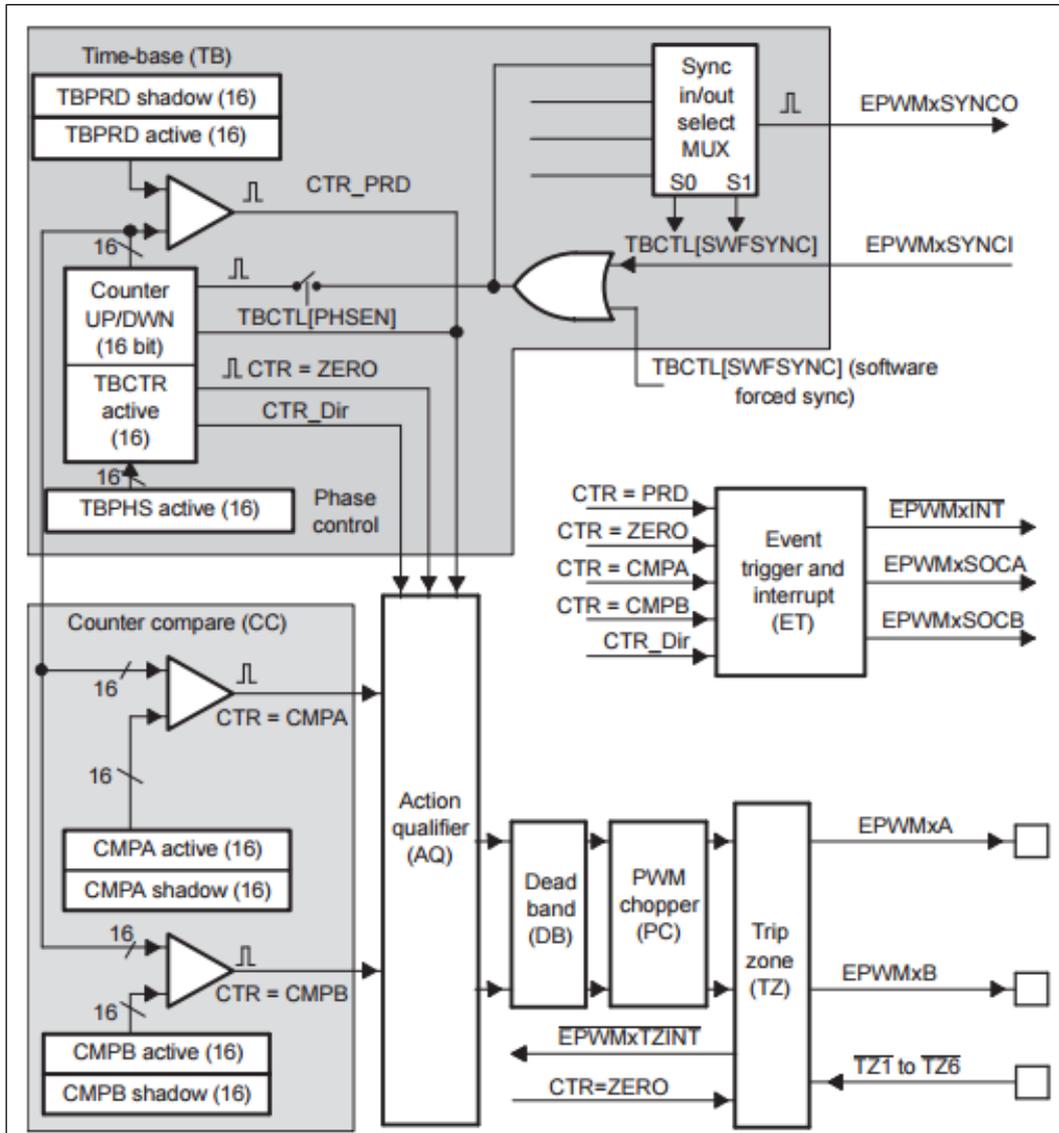


Figura 4.33 – Estructura de un módulo ePWMx.

Y se explicará el funcionamiento de varios de los bloques que aparecen en la Figura 4.33. En primer lugar analizaremos el bloque que se encarga de generar la señal triangular, denominado Counter UP/DOWN.

Este bloque tiene como entrada la señal TBPHS, el valor del periodo deseado y la señal del reloj interno del DSP. Para generar la señal triangular se va aumentando en una unidad el valor de esta señal cada vez que se recibe un flanco de subida de la señal de reloj, hasta alcanzar el valor del periodo deseado, si por ejemplo el valor del periodo deseado es de 4, y la señal triangular queremos que sea ascendente/descendente obtendremos una señal como la mostrada en la Figura 4.34.

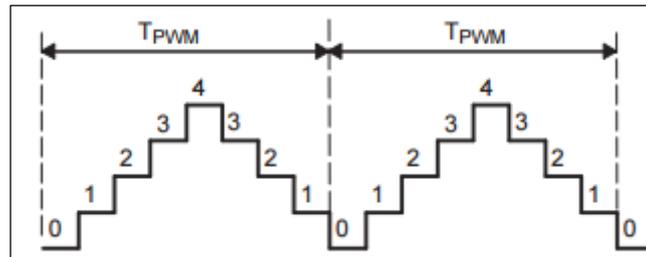


Figura 4.34 - Ejemplo de señal triangular

Como podemos ver la señal va aumentando en una unidad hasta alcanzar el valor 4, momento en el que empieza a descender hasta volver a alcanzar en valor 0, y así repetidamente.

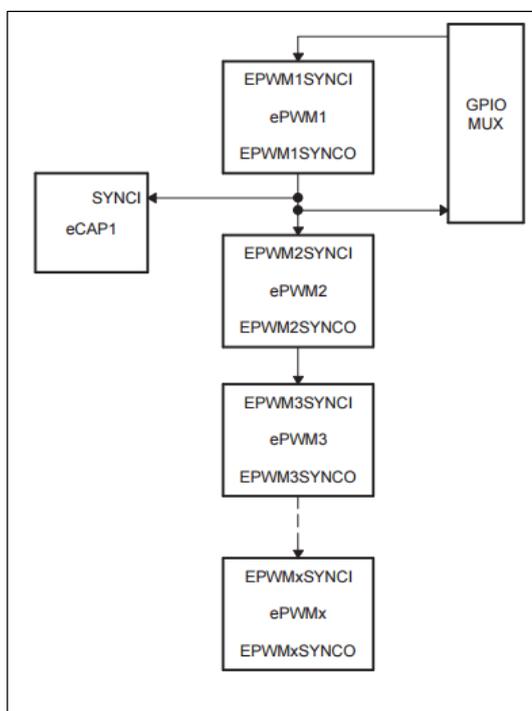


Figura 4.35 - Propagación de la señal de sincronismo

Si analizamos el bloque que se encarga de generar una señal de sincronismo, denominado TBPHS active, podemos mostrar cómo se propaga la señal de sincronismo, mostrada en la Figura 4.35. Como podemos ver la señal de sincronismo puede ir propagándose por los diferentes bloques, sin embargo dentro de cada bloque podremos seleccionar si queremos como señal de sincronismo de salida la misma señal que la de entrada, si queremos una señal de salida activa cuando la señal triangular pase por 0 ó por un

valor deseado. En la Figura 4.36 podemos ver este aspecto.

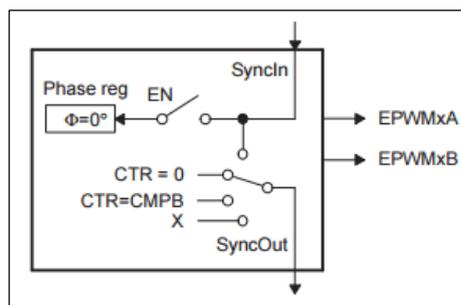


Figura 4.36 - Selección de la señal de sincronismo de salida

Un ejemplo de cómo funciona este sistema se ve en la siguiente figura.

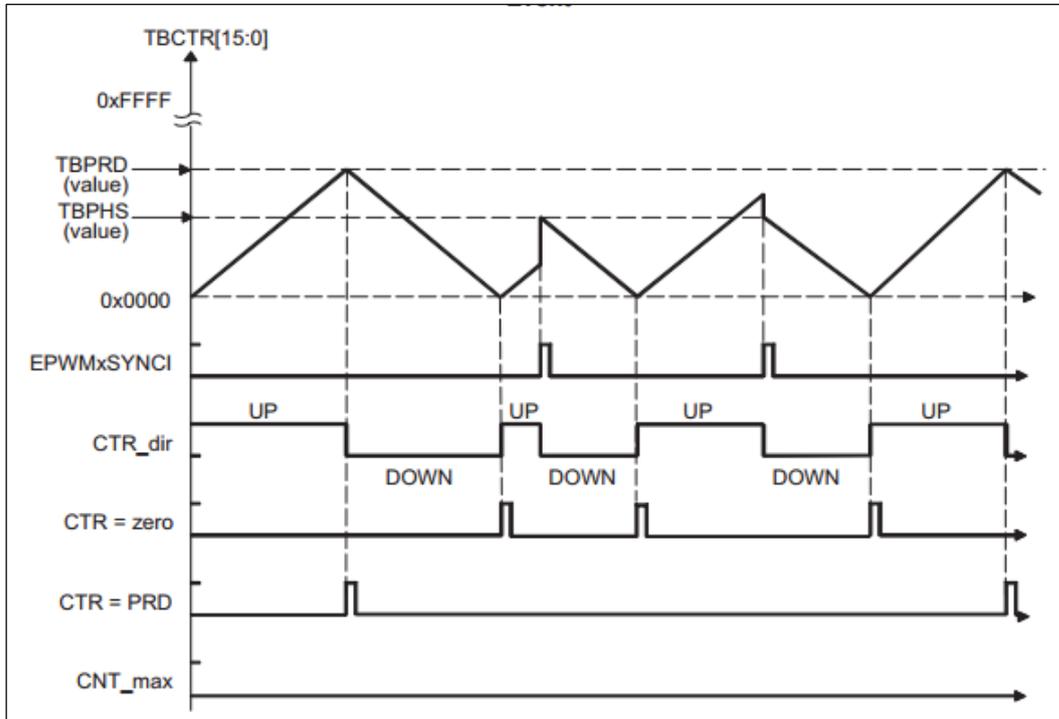


Figura 4.37 – Ejemplo de señal de sincronismo

Se puede ver como cuando se activa la señal de sincronismo la señal triangular pasa a tener el valor y sentido deseado.

Por último se analizará el bloque Dead-Band que se encarga de generar unos tiempos muertos en las señales para evitar cortocircuitos. El esquema de este módulo lo podemos ver en la siguiente figura.

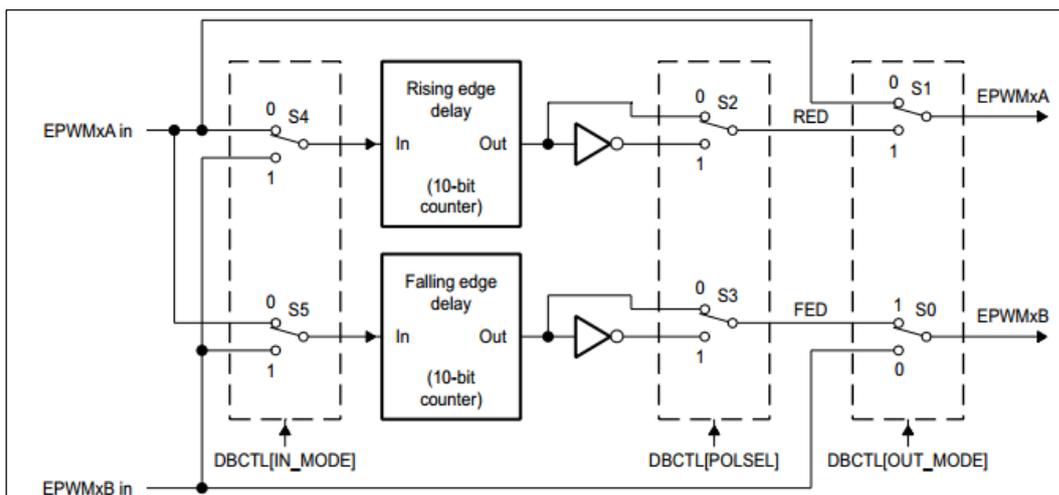


Figura 4.38 – Módulo Dead-Band

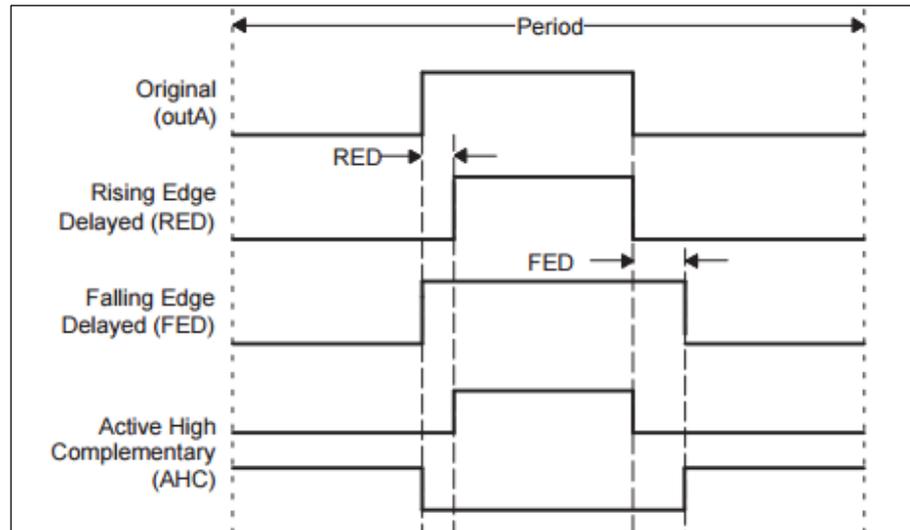


Figura 4.39 - Efecto de los tiempos muertos

Como podemos ver el valor de RED hace que la señal ePWMxa (penúltima señal de la Figura 4.39) tenga el flanco de subida un poco más tarde de lo esperado, y el valor FED hace que la señal ePWMxb tenga el flanco de subida un poco más tarde de lo esperado. Esto se consigue seleccionando como opción “Active High Complementary” que sirve para establecer estos tiempos muertos en señales asimétricas, como en nuestro caso.

4.2.1. SIMULACIÓN DEL CÓDIGO

Para simular este código se ha usado el programa de simulación PSIM con el que podemos simular la etapa de control y la etapa de potencia de forma conjunta.

Para generar las señales PWM de control se han empleado unas fuentes de alimentación denominadas “square”, configuradas para que generen las ondas adecuadas.

En la etapa de potencia se han añadido los cuatro polos por rama, con las características reales de resistencias internas y voltajes de caída internos.

En la siguiente figura se puede ver cuál es el esquema de PSIM para la simulación.

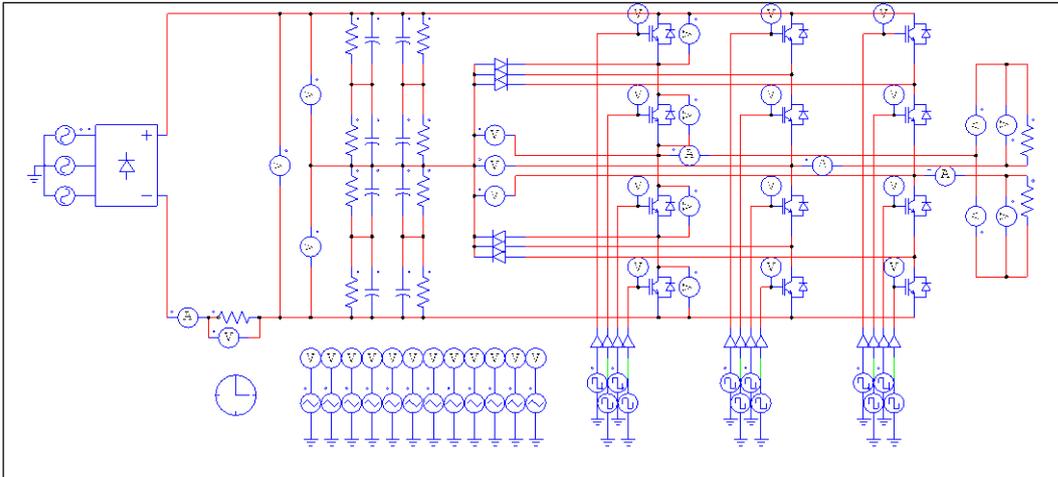


Figura 4.40 – Esquema de Simulación PSIM

Como tensiones de salida fase-neutro se han obtenido los siguientes resultados.

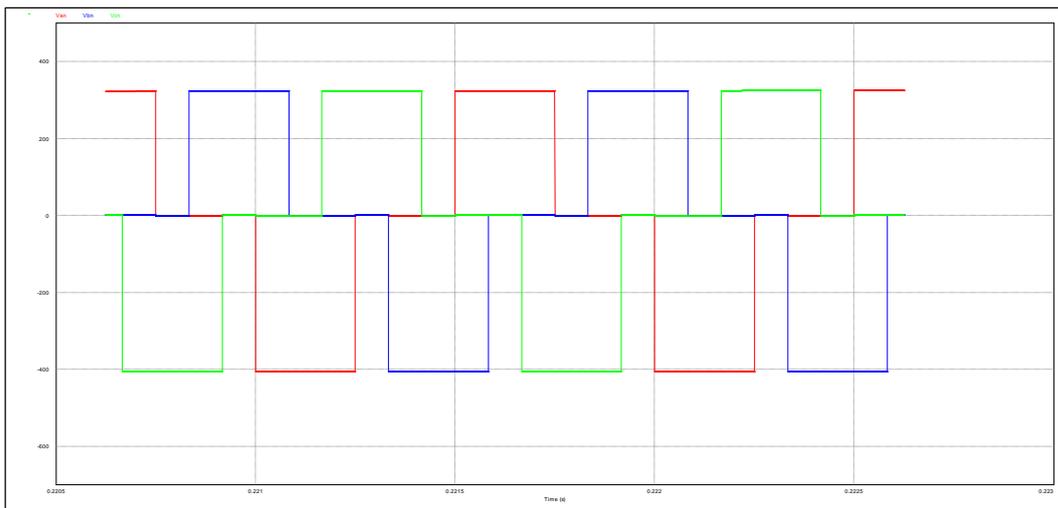


Figura 4.41 – Tensiones V_{an} , V_{bn} y V_{cn} en simulación

Como podemos ver las tensiones tienen tres niveles, y están desfasadas 120° grados.

Si simulamos las tensiones fase-fase que hay obtenemos los resultados mostrados en la siguiente figura.

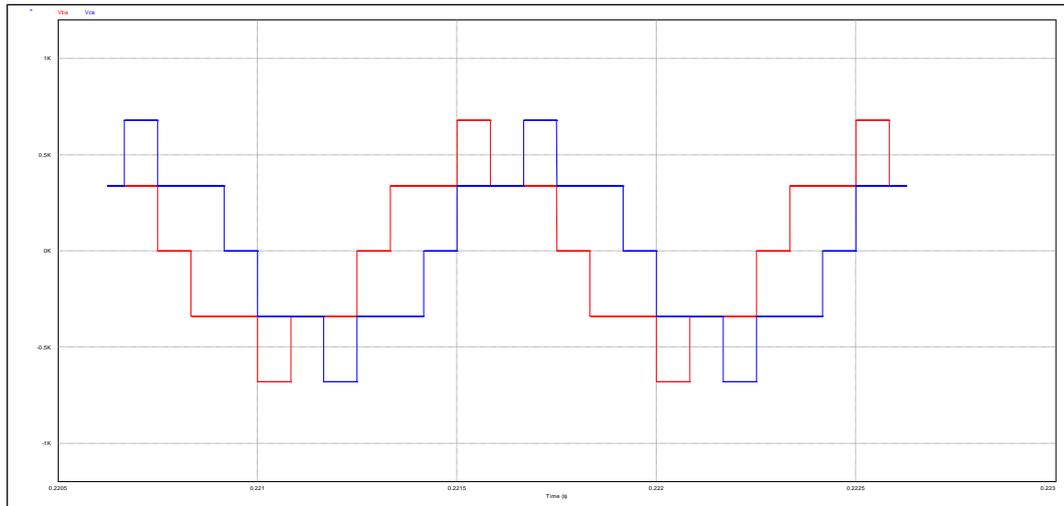


Figura 4.42 – Tensiones V_{be} y V_{ce} en simulación

Por último mostraremos cuáles son las caídas de tensiones en los diferentes polos de la rama A.

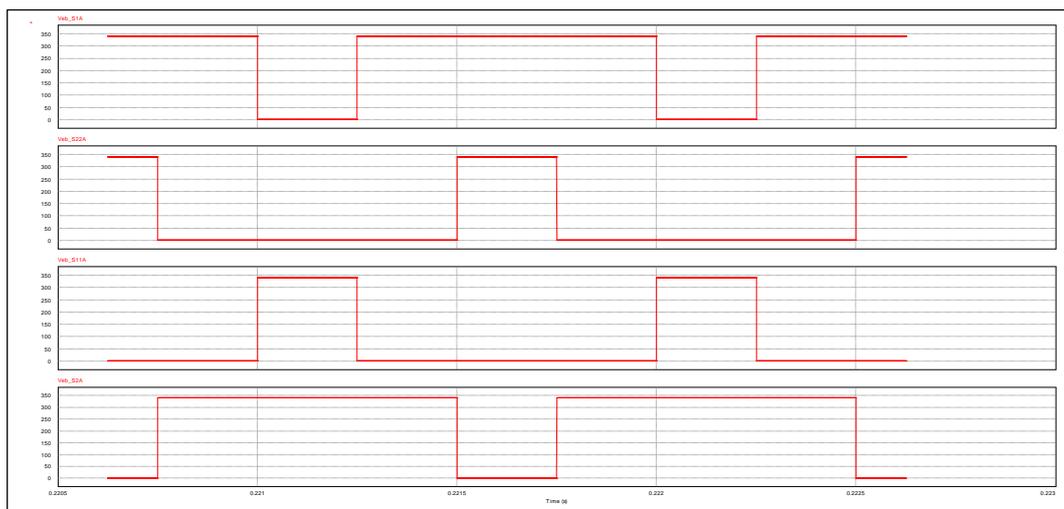


Figura 4.43 – Caídas de tensión en los polos de potencia de la rama A

Si comparamos estos resultados con los resultados obtenidos en las pruebas reales podemos ver como en las caídas de tensiones de los polos se obtienen resultados diferentes, esto es debido a que en la unión entre el emisor de S1 (o S11) con el colector de S22 (o S2) existe una tensión difícil de predecir.

4.3. ENTORNO DE PROGRAMACIÓN MATLAB Y SIMULINK

Se ha desarrollado mediante un esquema de Simulink y funciones de MATLAB un código que implementa una modulación vectorial en el inversor multinivel. Para ello hemos tenido que configurar MATLAB como se muestra en el apartado 4.1.

4.3.1. FUNCIONAMIENTO MODULACIÓN VECTORIAL DE INVERSORES MULTINIVEL

Considerando el inversor de tres niveles mostrado en la Figura 4.44, se observa que cada conmutador de selección posee tres posiciones posibles. El mapa de vectores sintetizables se muestra en la Figura 4.45. En el extremo de cada vector están indicadas las posiciones de los conmutadores de columna que los sintetizan. Se observa en la misma figura, que los vectores interiores pueden ser obtenidos con más de una combinación de estados de los interruptores. Precisamente todos los vectores que conforman el hexágono interno poseen dos combinaciones posibles, mientras que el vector nulo posee tres. Esta característica de redundancia en la síntesis de tensiones puede resumirse en 19 vectores diferentes con $3^3=27$ combinaciones de interruptores.

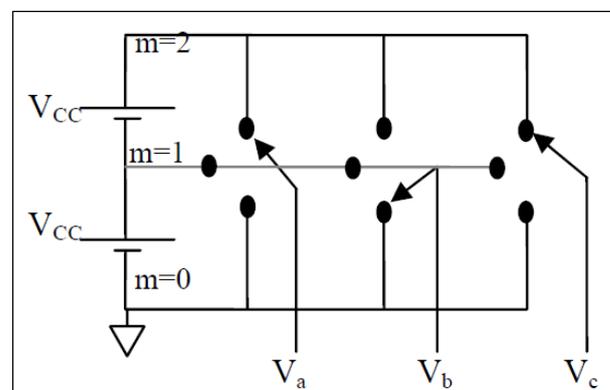


Figura 4.44 - Conmutaciones en el inversor de tres niveles



En general, para un convertidor de N niveles, existen N^3 maneras diferentes de combinar las tensiones de columna, mientras que el número de vectores sintetizables está dado por la siguiente expresión.

$$L = 1 + 6 \sum_{i=1}^{N-1} i$$

Ecuación 7 - Numero de vectores sintetizables

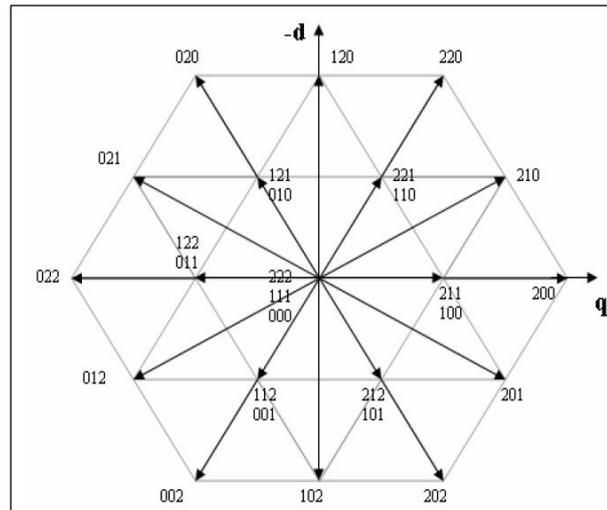


Figura 4.45 - Mapa de vectores en un inversor de tres niveles

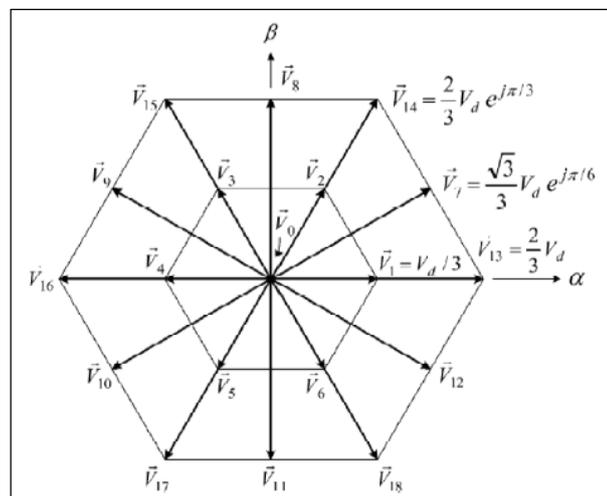


Figura 4.46 - Diagrama vectorial para un inversor de tres niveles

Para facilitar el cálculo de los tiempos de aplicación, el diagrama de vectores espaciales de la Figura 4.46, puede ser dividido en seis sectores triangulares (I a VI), y a su vez cada uno de ellos puede ser subdividido en cuatro regiones (1 a 4) tal y como se muestra en la Figura 4.47. Además, en la Figura 4.46

también se pueden observar los estados de conmutación asociados a cada vector espacial. Observe que los vectores activos y los vectores cero no se mueven, por ello también son conocidos como vectores estacionarios. Por el contrario, el vector de referencia (\vec{V}_{ref}) que se desea generar y que también es mostrado en la Figura 4.47, gira a una velocidad angular dada por la frecuencia fundamental del voltaje de salida del inversor.

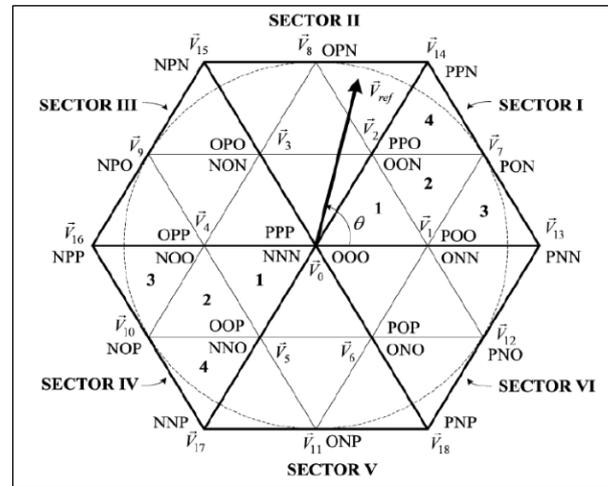


Figura 4.47 – Sectores del diagrama vectorial para un inversor de tres niveles

El tiempo de aplicación de los vectores estacionarios representa la relación de tiempo en la que un determinado interruptor está cerrado o abierto durante un periodo de conmutación T_s . El cálculo del tiempo de aplicación de los vectores espaciales se basa en que el producto del voltaje de referencia \vec{V}_{ref} y el periodo de conmutación T_s , debe ser igual a la suma del voltaje.

Región	T_a	T_b	T_c
1	$\vec{V}_1 \quad T_s [2m_a \sin(\frac{\pi}{3} - \theta)]$	$\vec{V}_0 \quad T_s [1 - 2m_a \sin(\frac{\pi}{3} + \theta)]$	$\vec{V}_2 \quad T_s [2m_a \sin \theta]$
2	$\vec{V}_1 \quad T_s [1 - 2m_a \sin \theta]$	$\vec{V}_7 \quad T_s [2m_a \sin(\frac{\pi}{3} + \theta) - 1]$	$\vec{V}_2 \quad T_s [1 - 2m_a \sin(\frac{\pi}{3} - \theta)]$
3	$\vec{V}_1 \quad T_s [2 - 2m_a \sin(\frac{\pi}{3} + \theta)]$	$\vec{V}_7 \quad T_s [2m_a \sin \theta]$	$\vec{V}_{13} \quad T_s [2m_a \sin(\frac{\pi}{3} - \theta) - 1]$
4	$\vec{V}_{14} \quad T_s [2m_a \sin(\theta) - 1]$	$\vec{V}_7 \quad T_s [2m_a \sin(\frac{\pi}{3} - \theta)]$	$\vec{V}_{21} \quad T_s [2 - 2m_a \sin(\frac{\pi}{3} + \theta)]$

Tabla 4.1 – Ejemplo de cálculo de los tiempos de aplicación cuando el vector de referencia está en el sector 1



En este caso, para elegir la secuencia de conmutación se han tenido en cuenta tres aspectos principales:

- La transición entre cada uno de los estados de conmutación debe involucrar únicamente dos interruptores en la misma rama, uno que está siendo abierto y el otro que se está cerrando.
- Es necesario minimizar o si es posible prescindir de conmutaciones cuando el \vec{V}_{ref} transita entre los diferentes sectores (o regiones) del espacio vectorial.
- Minimizar el efecto de la variación del voltaje en el punto neutro, que es propiciada por la aplicación de los diferentes estados de conmutación.

4.3.2. CÓDIGO DESARROLLADO

El esquema de Simulink implementado para conseguir una conmutación SVM es el siguiente.

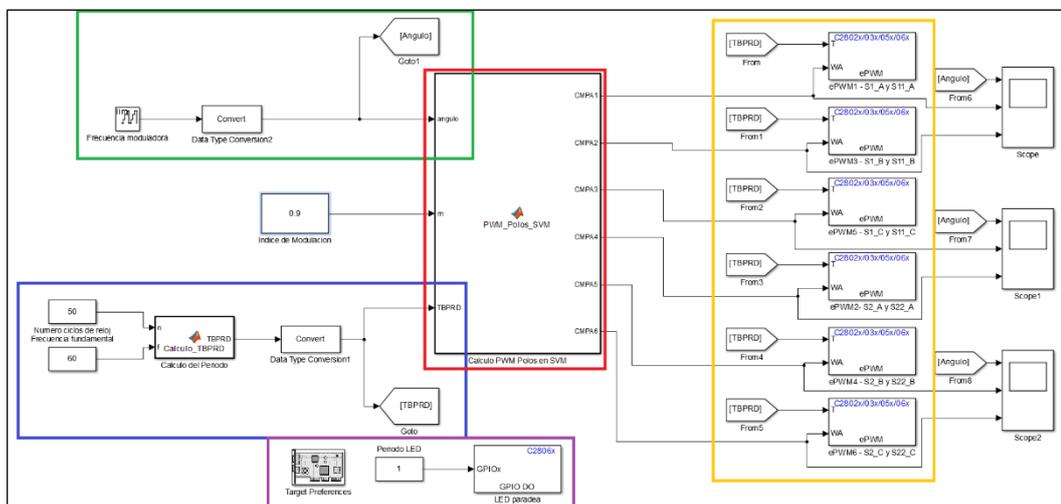


Figura 4.48 – Diagrama de bloques de Simulink para implementar SVM

El presente esquema está dividido en varias partes, las cuales realizan una función específica cada una.

- Bloque morado: En este bloque se han incluido el elemento “Target Preferences”, en el que se indican las características de la tarjeta en la

que se implementará este esquema, y un bloque que realizará la función de que el LED que tiene la placa DSP luzca con una determinada frecuencia para saber que el código está ejecutándose.

- **Bloque Azul:** En este bloque se puede especificar la frecuencia fundamental a través de una constante, y se convierte esta frecuencia portadora en el valor TBPRD, que tal como hemos visto es el valor del periodo de las señales ePWM.
- **Bloque verde:** En este bloque se crea una señal que simula la velocidad angular de rotación del vector de referencia. Este vector irá desde la posición 0 hasta la 359. El tiempo especificado en este bloque determina la frecuencia de modulación.
- **En el bloque rojo** se calcula, a partir del valor de vector de referencia, del índice de modulación, y del valor TBPRD el valor CMPA que necesitará cada rama para conmutar de forma correcta.
- **En el bloque amarillo** se implementarán las señales ePWM a partir de los valores TBPRD y CMPA generados anteriormente.

Estos bloques están configurados de la siguiente manera:

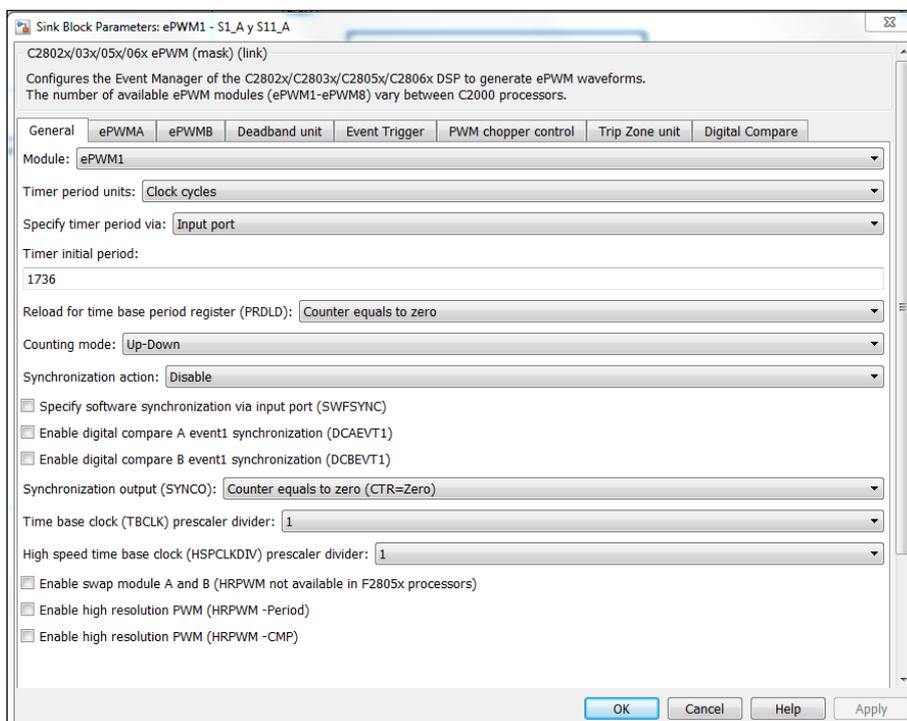


Figura 4.49 – Configuración general de los bloques ePWMx

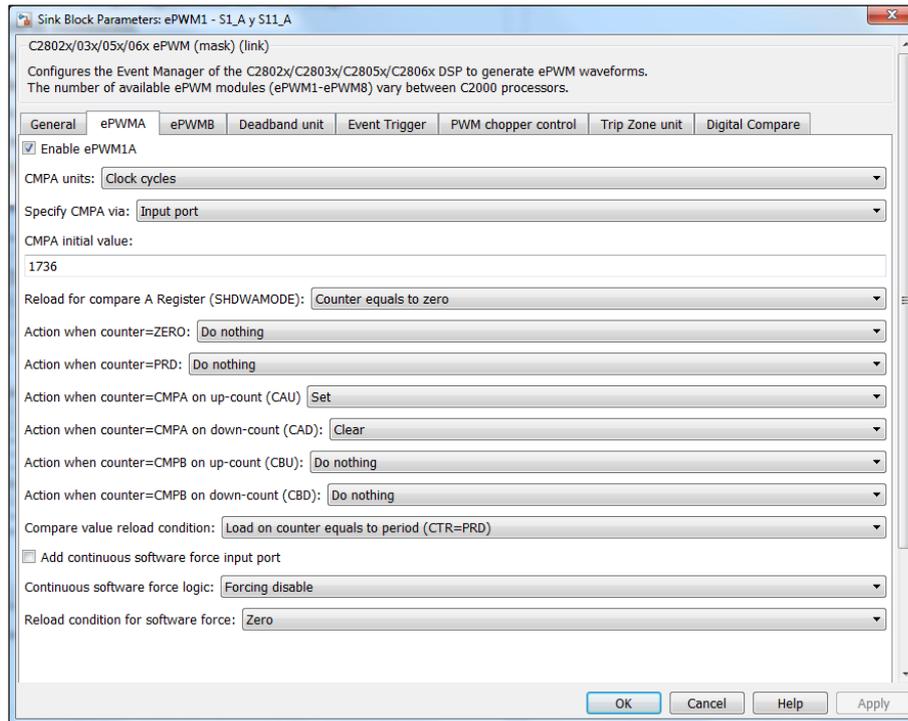


Figura 4.50 – Configuración ePWMA de los bloques ePWMx

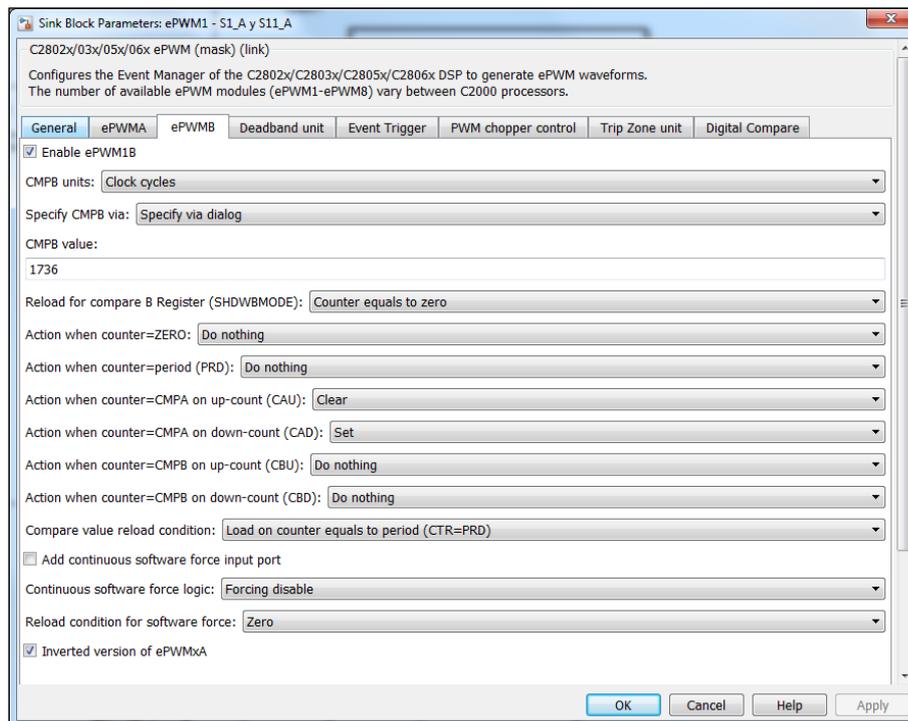


Figura 4.51 – Configuración ePWMB de los bloques ePWMx

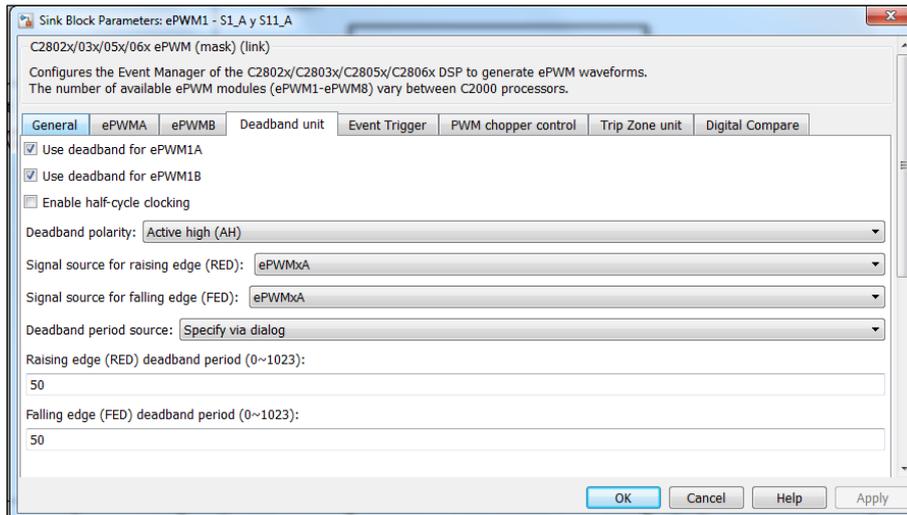


Figura 4.52 – Configuración Dead-Band de los bloques ePWMx

El funcionamiento del sistema puede verse en la siguiente figura.

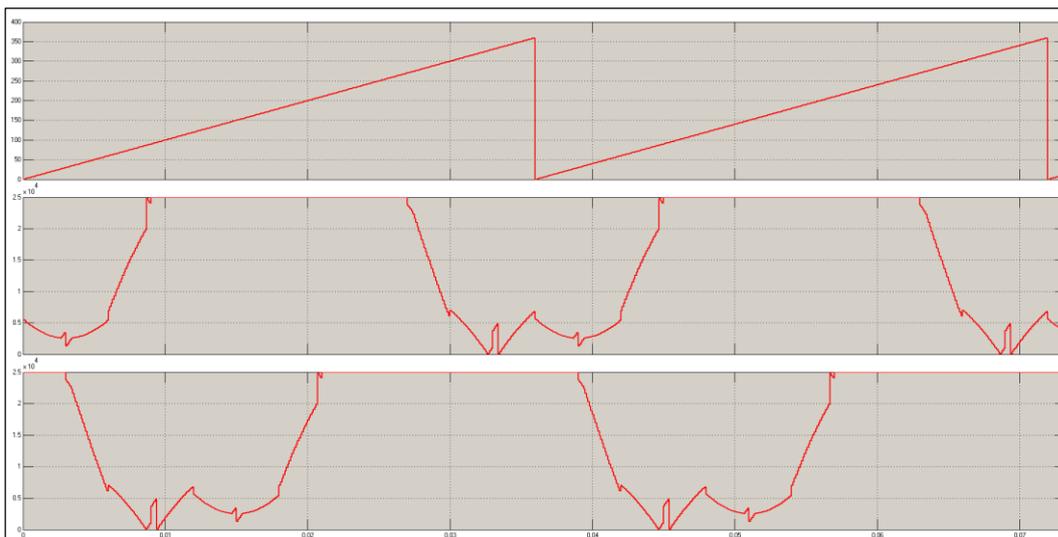


Figura 4.53 – Comportamiento del código

Como podemos ver se generan unas señales CMPA1 y CMPA2 en función del valor del ángulo, y se actualizan cada vez que el ángulo cambia de valor.

4.3.3. SIMULACIÓN DEL CÓDIGO

Se ha creado otro esquema en Simulink capaz de simular el comportamiento de un inversor trifásico de tres niveles controlado mediante modulación vectorial.

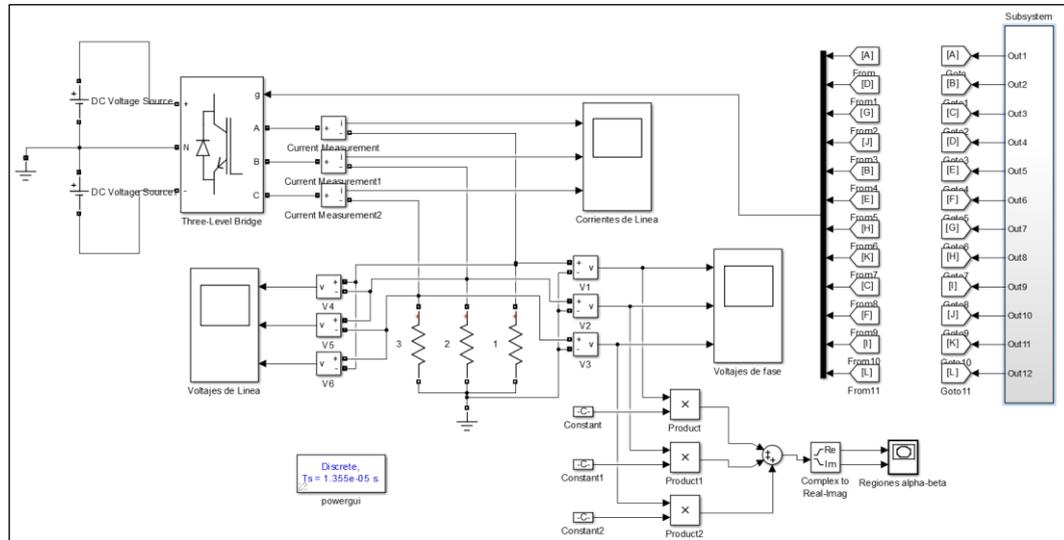


Figura 4.54 – Esquema de Simulación en Simulink

Como se puede ver se ha simulado el sistema con una carga resistiva, y se ha añadido la posibilidad de visualizar el plano alpha-beta.

Se ha simulado el sistema para una frecuencia de 50Hz y varios índices de modulación. En primer lugar para un índice de modulación de 0.9 obteniéndose los siguientes resultados.

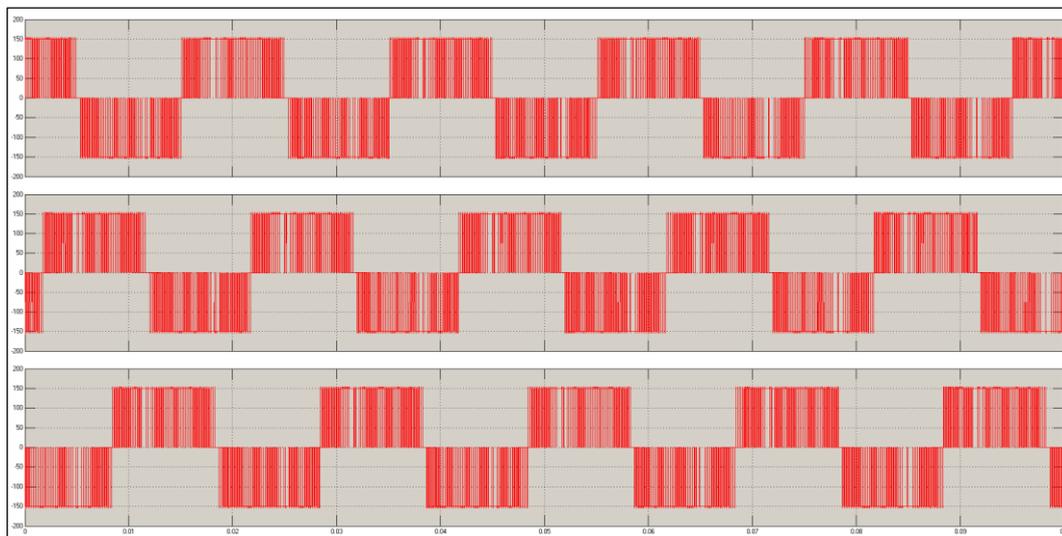


Figura 4.55 – Voltajes de fase en simulación SVM con $m=0.9$

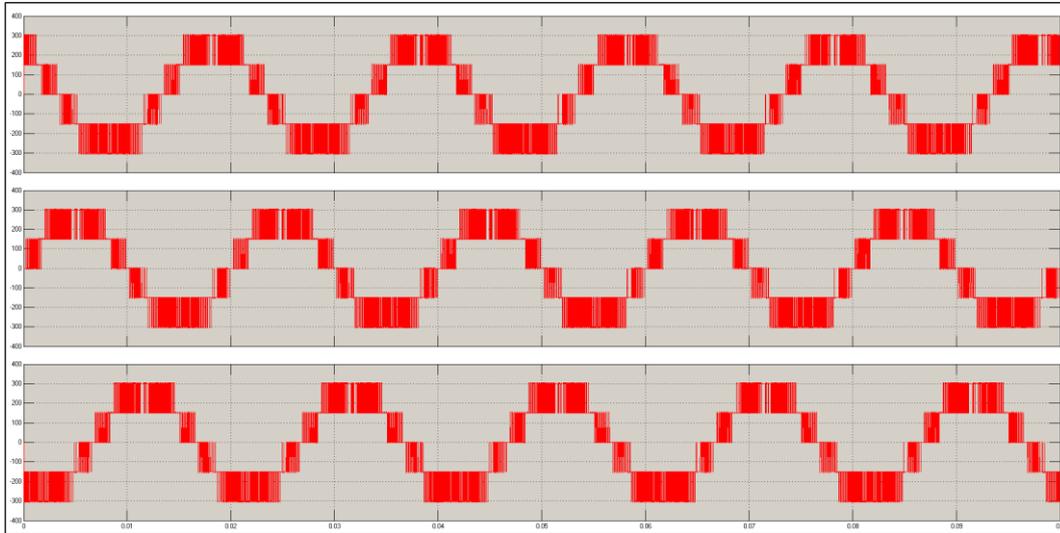


Figura 4.56 - Voltajes de línea en simulación SVM con $m=0.9$

Posteriormente se ha modificado el índice de modulación a un valor de 0,3.

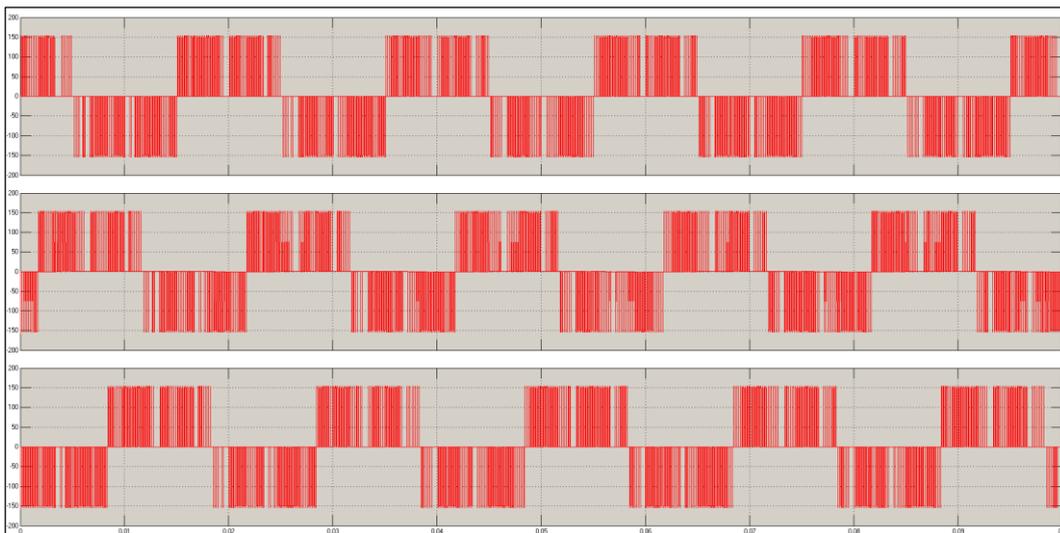


Figura 4.57 - Voltajes de fase en simulación SVM con $m=0.3$

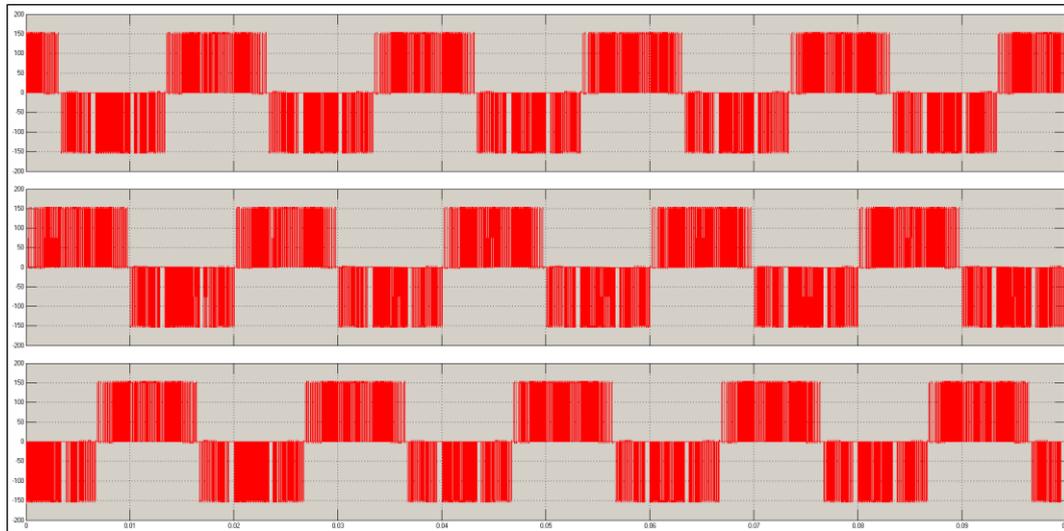


Figura 4.58 – Voltajes de línea en simulación SVM con $m=0.3$

Como podemos ver para índices de modulación menores que 0,5 no se producen 5 niveles de la tensión de salida, esto es debido a que para alcanzar amplitudes menores de la mitad, es mejor hacerlo con 3 niveles que con 5. Este aspecto se ve claramente si analizamos el plano alpha-beta en el que se mueve el inversor en cada caso. Para amplitudes mayores que 0,5 podemos ver que se mueve tanto por el hexágono interior como por el exterior y para amplitudes menores solo se mueve por el hexágono interior.

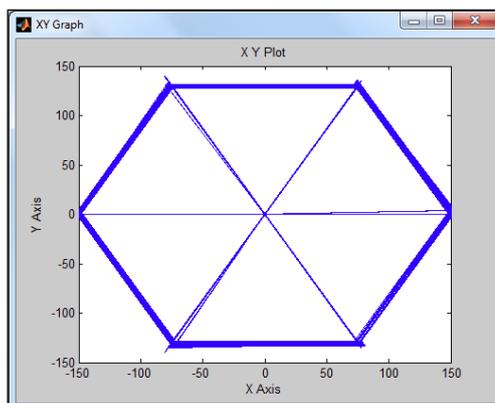


Figura 4.59 – Plano alpha-beta para $m < 0.5$

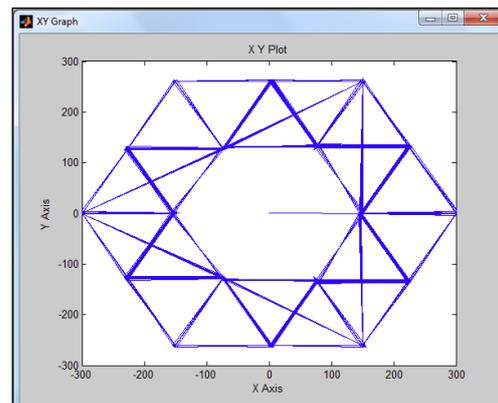


Figura 4.60 – Plano alpha-beta para $m > 0.5$

4.3.4. PARÁMETROS CONFIGURABLES DEL CÓDIGO

El primer parámetro configurable en este código de Simulink es la frecuencia de la señal moduladora. Para poder variar esta frecuencia habrá que

modificar el valor del campo “Sample time” en el bloque llamado “Frecuencia moduladora”. Como este valor será el que tarde en aumentar su valor en una unidad y el periodo completo será lo que tarde en aumentar 360 unidades, el valor del periodo seguirá la siguiente ecuación.

$$Frecuencia\ Moduladora = \frac{1}{360 * 'Sample\ Time'}$$

Ecuación 8 – Valor de la frecuencia moduladora

En segundo lugar también se puede variar el índice de modulación. Este campo tomará un valor comprendido entre 0 y 1. El índice de modulación controlará la amplitud de la señal de salida, teniendo esta amplitud máxima cuando el valor del índice de modulación sea 1.

Por último se podrá ajustar el valor de la frecuencia de la señal fundamental. Para poder determinar el valor de esta frecuencia necesitaremos saber cuál será el valor al que está configurado la señal de reloj de la CPU. Este valor lo podremos encontrar en el apartado de configuraciones de la tarjeta, y en nuestro caso será de 80 MHz, pero podrá ser modificado por el usuario.

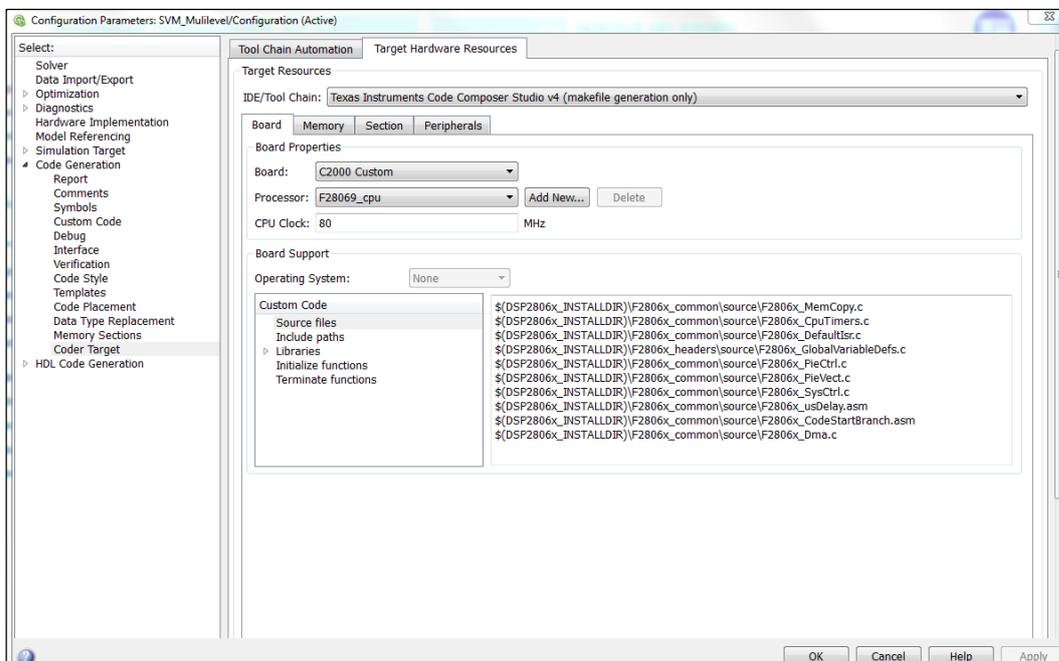


Figura 4.61 – Frecuencia de trabajo de la CPU

Una vez que conocemos la frecuencia de trabajo de la CPU se podrá calcular cual será el valor de la frecuencia fundamental en función de dos valores



modificables en el esquema, que son los bloques denominados “número de ciclos de reloj” y “frecuencia fundamental”. El valor de esta frecuencia seguirá la siguiente ecuación.

$$F_{portadora} = 'numero\ de\ ciclos\ de\ reloj' * 'frecuencia\ fundamental'$$

Ecuación 9 – Valor de la frecuencia portadora

Y a partir del valor de esta frecuencia se determinará el valor de TBPRD, que seguirá esta otra ecuación.

$$TBPRD = \frac{Frecuencia\ de\ la\ CPU}{2 * F_{portadora}}$$

Ecuación 10 – Valor de TBPRD

Por lo que si se cambia el valor al que la señal de reloj de la CPU está configurado, habrá que cambiar el valor en la función “cálculo del periodo” para que este cálculo sea el correcto.





5. MONTAJE DEL EQUIPO

Se ha montado una estructura de metacrilato para proteger a los usuarios que usen el equipo. Para ello se han cortado a medida tres placas de metacrilato, dos para las paredes laterales y otra para la parte superior. Las paredes laterales se han fijado al radiador a través de 4 tornillos en cada pared, será necesario además emplear unas arandelas para lograr una separación correcta.

En una de estas tapas se ha abierto una pequeña ventana para hacer que los cables que van desde el driver hasta la placa de potencia puedan conectarse de forma correcta. En la siguiente figura podemos ver esta ventana abierta y los cuatro tornillos que sujetan estas paredes con el radiador.



Figura 5.1 – Pared lateral

Los conjuntos de transistores y el rectificador de entrada están atornillados al radiador para poder disipar el calor de forma correcta. Podemos ver en la Figura 5.2 como han quedado unidos estos dispositivos con el radiador.



Figura 5.2 – Uniones para disipar calor

Para finalizar la estructura se ha colocado una tapa superior en la que van atornillados los seis drivers de control. Esta placa superior se ha unido a las placas laterales a través de seis 'L' metálicas. Estas 'L' han servido para que el peso de la placa superior se reparta por las paredes laterales. En la Figura 5.3 se observa cómo quedan estos elementos de sujeción una vez atornillados tanto en las caras laterales como en el plano superior.

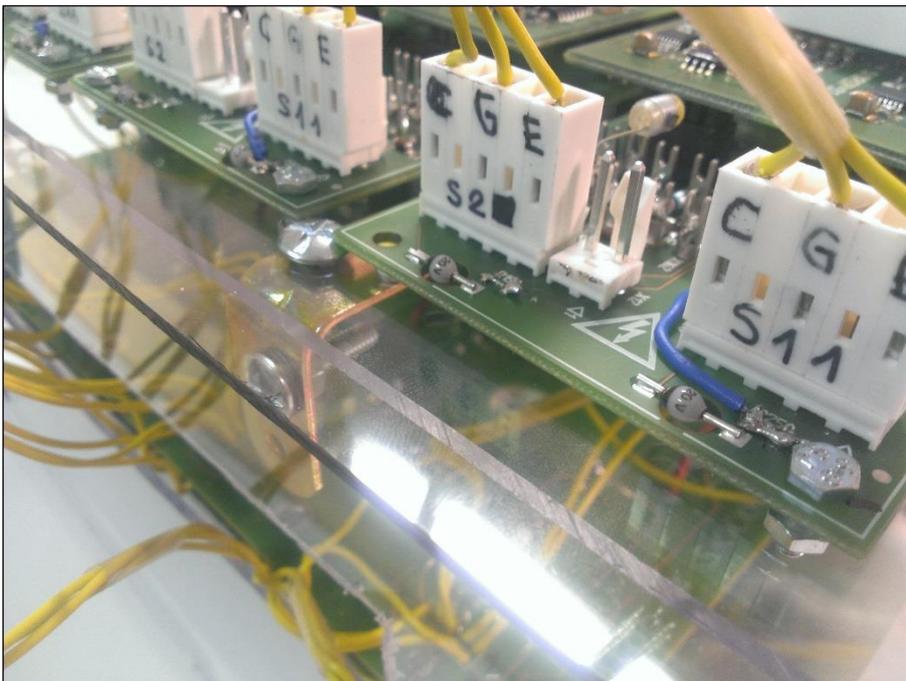


Figura 5.3 – Estructuras de sujeción entre caras laterales y superior.

En las dos siguientes figuras podemos ver el interior de la estructura para observar las 'L' y los tornillos que sujetan los drivers.

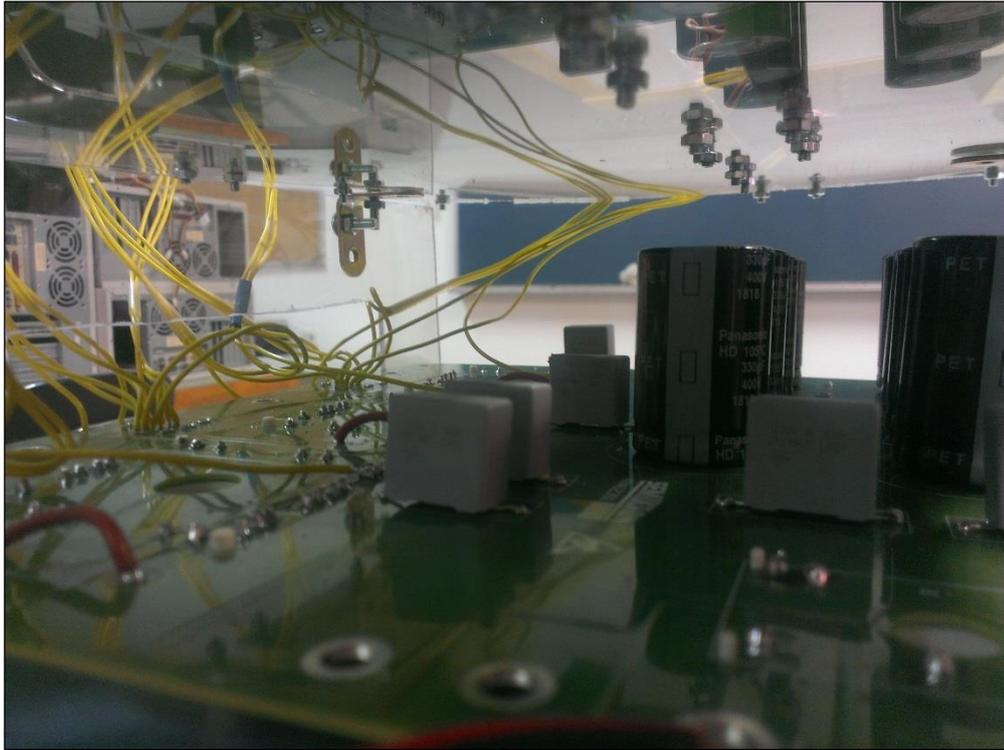


Figura 5.4 - Interior de la estructura (1)



Figura 5.5 - Interior de la estructura (2)

La estructura final del equipo con todos los elementos sujetos y los cables conectados se puede ver en la siguiente figura.

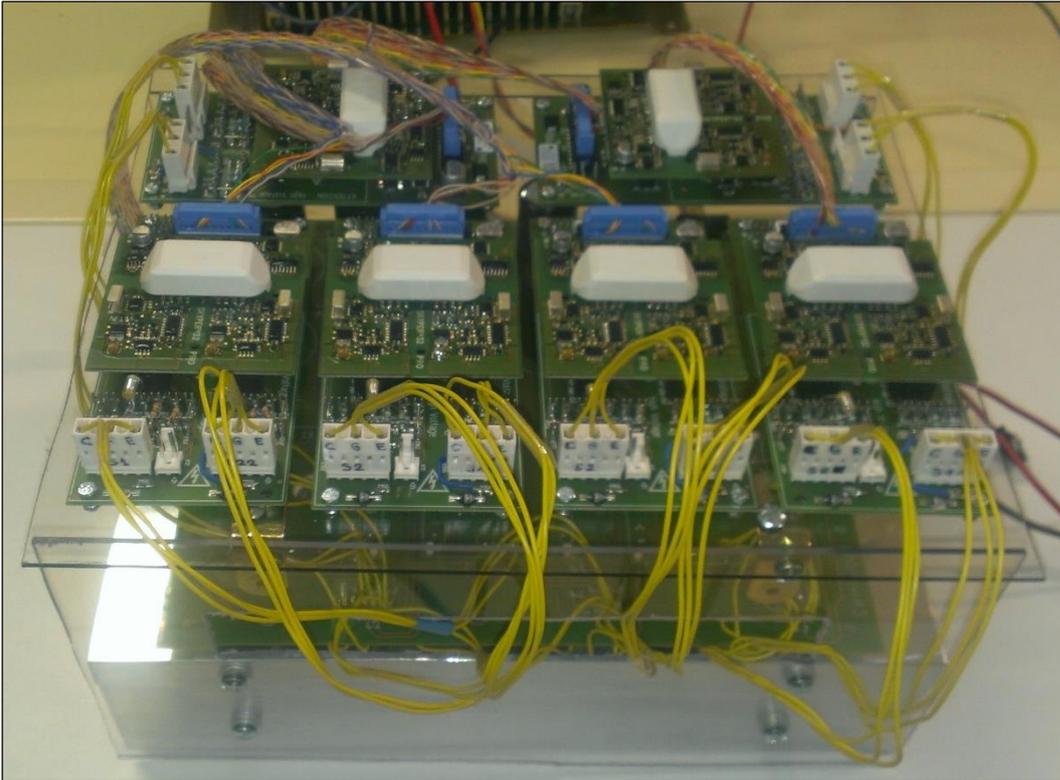


Figura 5.6 – Vista general de la estructura montada

Se observa en esta figura como el orden de los componentes es bueno ya que permite una correcta distribución de las señales por los diferentes cables, además el interior de la estructura ha quedado protegida.

Para acabar cabe comentar que en los otros dos laterales no se han colocado paredes de protección ya que es necesario que por estos lados circule un caudal de aire para permitir la refrigeración del equipo. También se han soldado de forma provisional tres cables para poder conectar a la fuente de alimentación trifásica de entrada. Esto lo podemos ver en la siguiente figura.

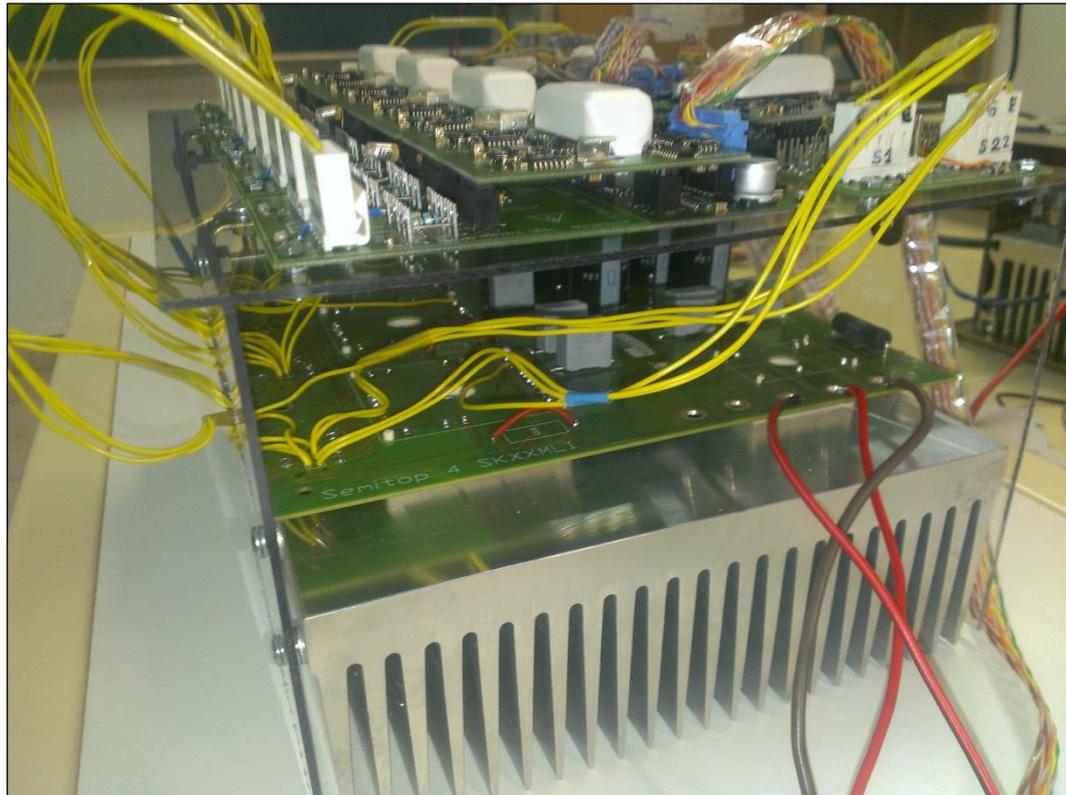


Figura 5.7 – Vista lateral del equipo





6. PRUEBAS REALIZADAS

En primer lugar haremos las pruebas de validación para comprobar que el inversor funciona correctamente, para ello ejecutaremos los programas creados en los entornos de programación de Code Composer Studio y de MATLAB/Simulink.

6.1. PRUEBAS DE VALIDACIÓN

Antes de conectar la placa de potencia con el DSP de control comprobaremos que las salidas PWM del DSP son correctas y están desfasadas el ángulo deseado. Para ello tomaremos como referencia la onda PWM que gobernará el transistor S1_A y la muestrearemos junto a las demás ondas PWM.

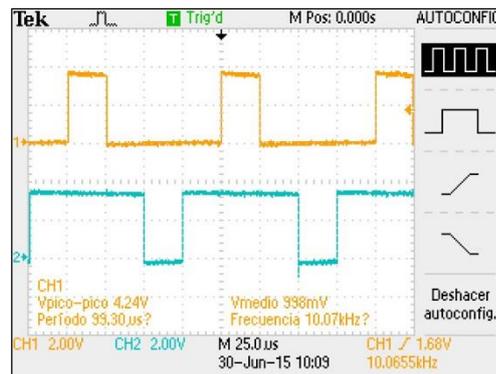


Figura 6.1 – S1_A y S22_A

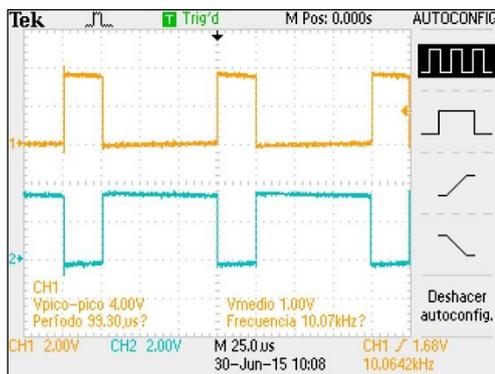


Figura 6.2 – S1_A y S11_A

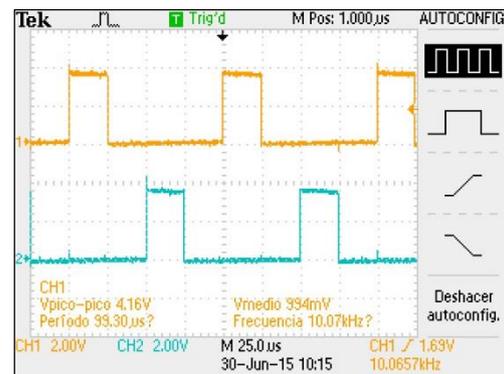


Figura 6.3 – S1_A y S2_A

Como se puede observar en esta Serán opuestas S1 con S11 y S2 con figura estas dos ondas son opuestas. S22 en las tres ramas.

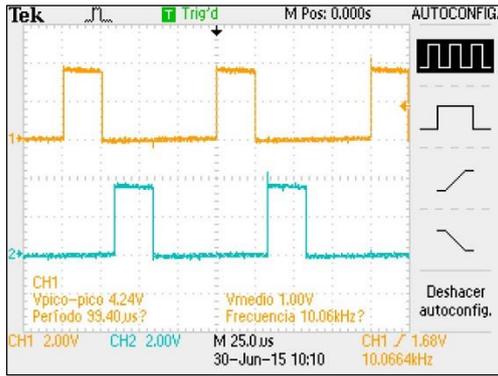


Figura 6.4 - S1_A y S1_B

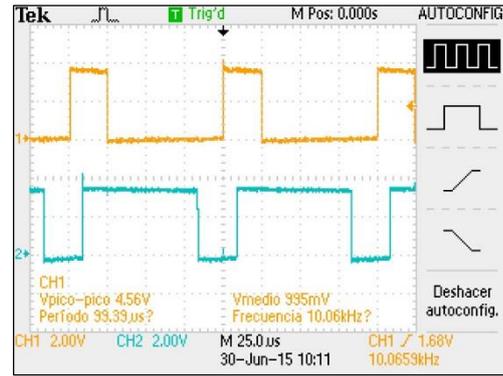


Figura 6.5 - S1_A y S22_B

En esta figura se puede ver como el desfase entre la rama A y la B es de 120° .

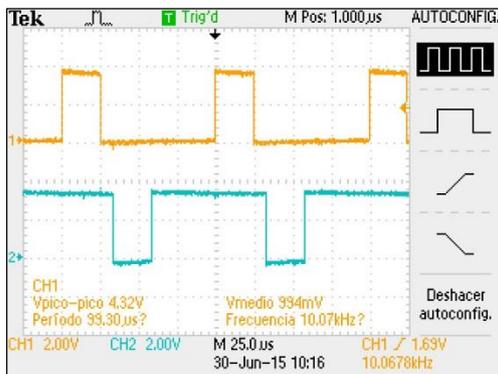


Figura 6.6 - S1_A y S11_B

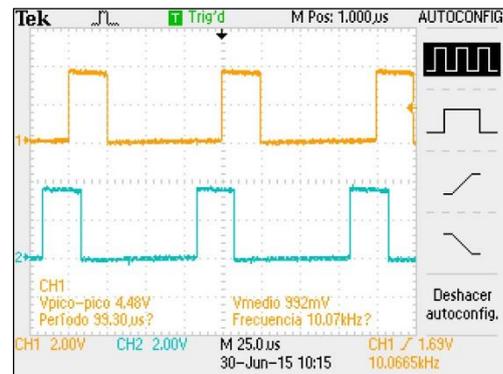


Figura 6.7 - S1_A y S2_B

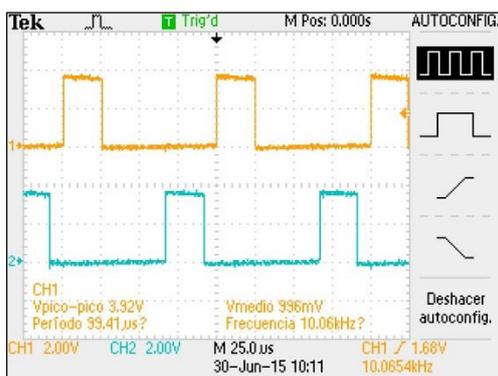


Figura 6.8 - S1_A y S1_C

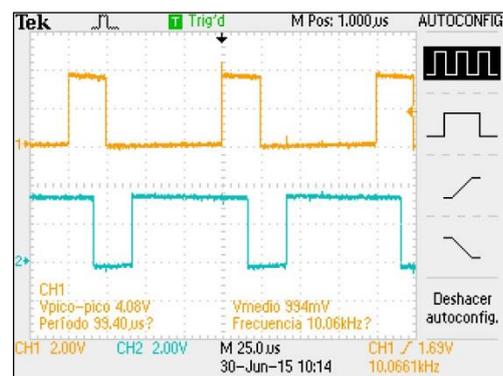


Figura 6.9 - S1_A y S22_C

Aquí podemos ver el desfase entre la rama A y C, que es de 240° .

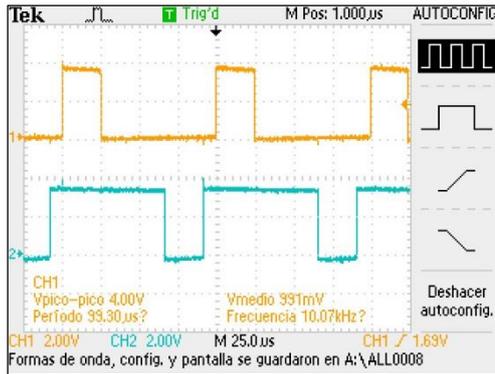


Figura 6.10 – S1_A y S22_C

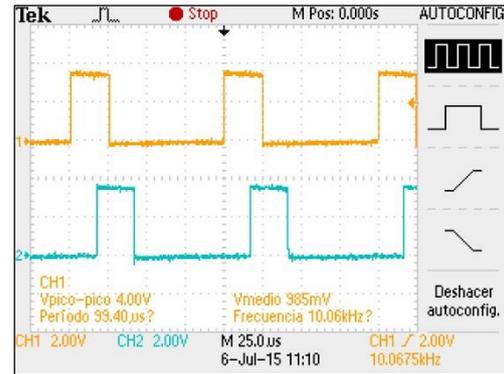
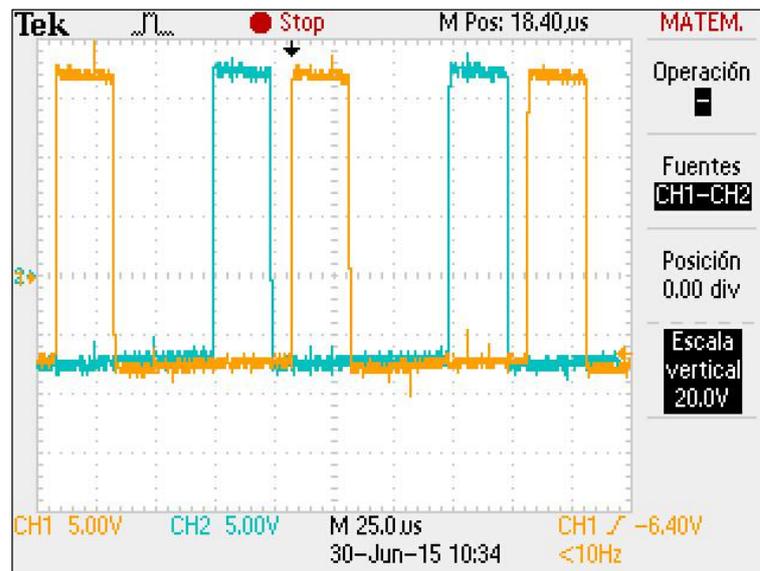


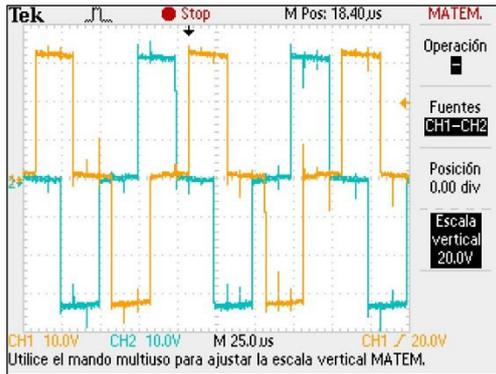
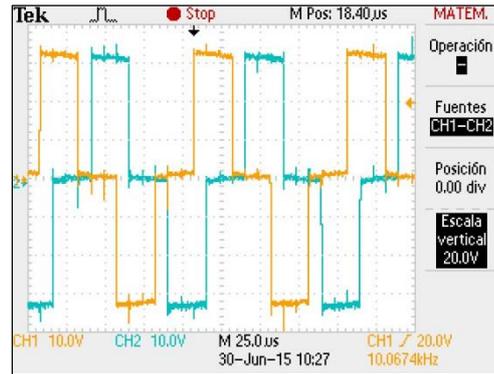
Figura 6.11 – S1_A y S2_C

Por otro lado podemos visualizar cuales serán las tensiones que el driver de control emite para controlar el transistor. Si mostramos las tensiones base emisor para los transistores S1_A y S1_B obtenemos estos resultados.

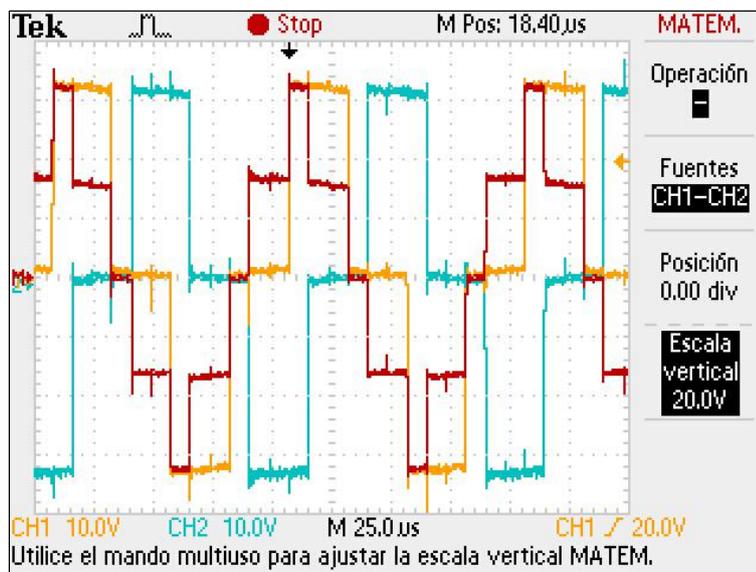
Figura 6.12 – Tensiones V_{EB} de S1_A y S1_B

Se puede observar cómo para activar el polo se genera una señal positiva de 17V y para apagar el polo una señal negativa de -7V tal y como nos dice la hoja de características del driver usado.

Siguiendo con las pruebas podemos observar la tensión de salida fase-neutro para las tres ramas.

Figura 6.13 - V_{AN} y V_{BN} Figura 6.14 - V_{AN} y V_{CN}

Como podemos observar en las dos figuras anteriores los desfases son de 120° entre dos ramas tal y como habíamos comentado. Si simulamos la operación matemática de restar la tensión fase neutro de dos líneas obtenemos la gráfica roja de la siguiente figura.

Figura 6.15 - V_{AN} , V_{CN} y V_{BC} matemático

Y el resultado real obtenido es el siguiente:

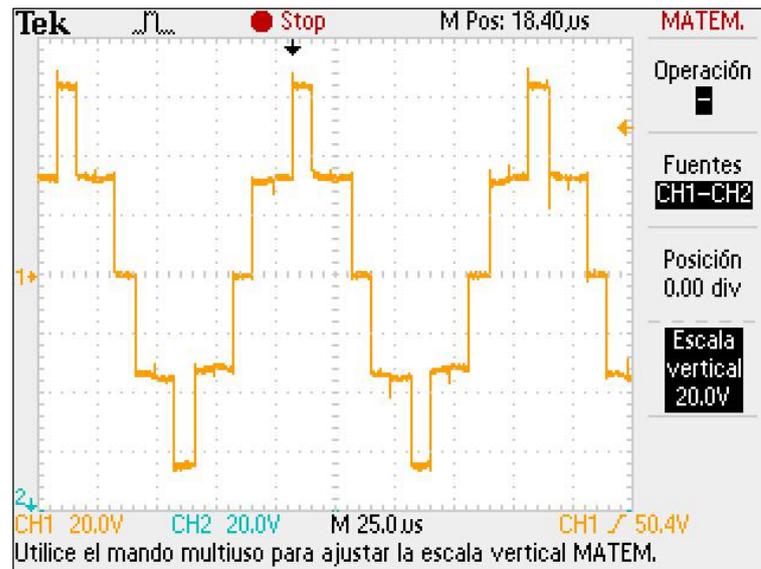


Figura 6.16 - V_{BC} real

El resultado teórico y el real son idénticos. Y si mostramos las tensiones fase-fase tomando como neutro la fase A, y como tensión positiva las otras dos fases obtenemos la siguiente gráfica.

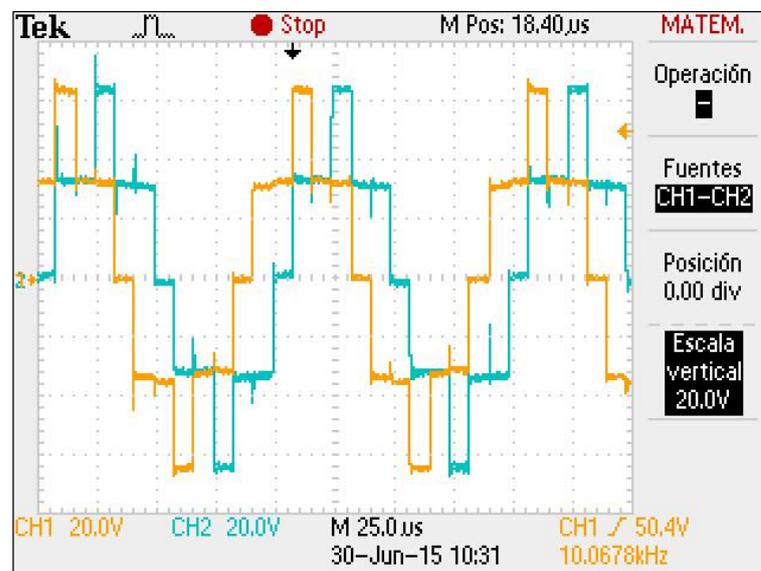
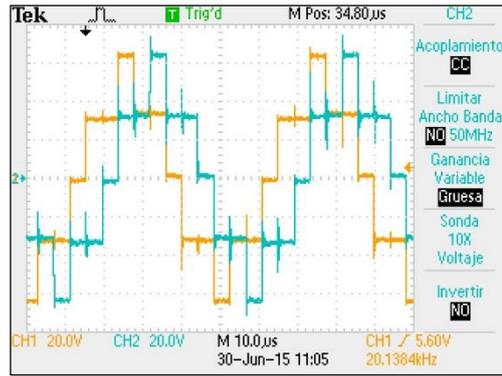
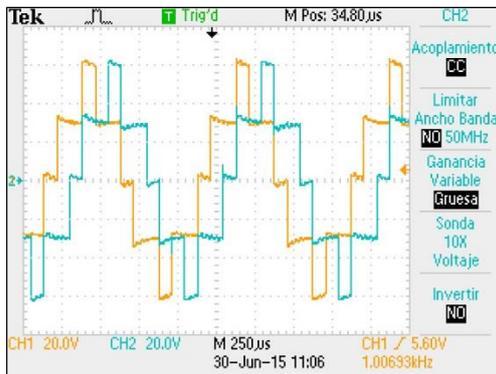
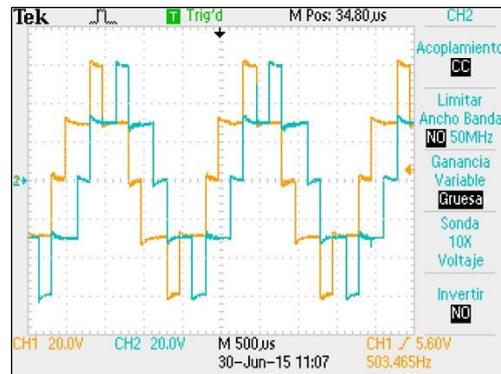
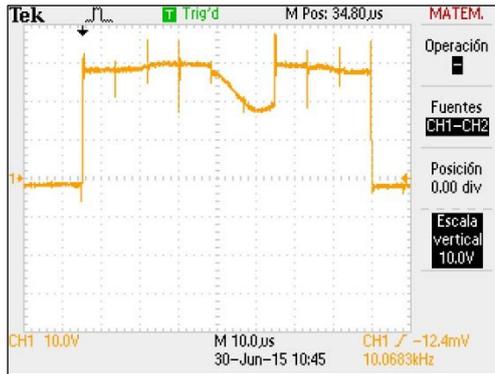


Figura 6.17 - V_{BA} y V_{CA} a 10KHz

Por último comprobaremos el correcto funcionamiento del sistema a diferentes tensiones.

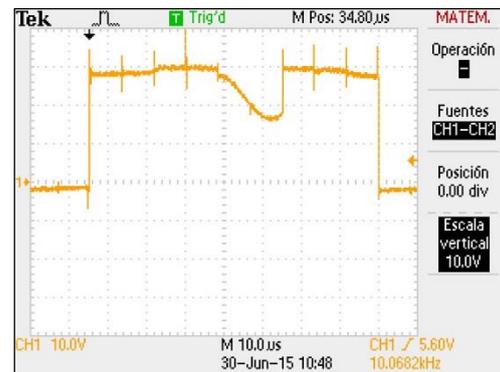
Figura 6.18 - V_{BA} y V_{CA} a 20KHzFigura 6.19 - V_{BA} y V_{CA} a 1KHzFigura 6.20 - V_{BA} y V_{CA} a 500Hz

Las caídas de tensión V_{CE} en los cuatro transistores de la rama A quedan reflejadas en las siguientes cuatro figuras:

Figura 6.21 - V_{CE} en S1_AFigura 6.22 - V_{CE} en S22_A

El eje horizontal refleja un periodo completo. Por ello que la caída de tensión será casi nula cuando el transistor S1 está en ON, $\frac{1}{4}$ del periodo, y durante los otros $\frac{3}{4}$ la caída de tensión será la de la tensión de alimentación.

En este caso, el transistor S22 está en ON durante $\frac{3}{4}$ del periodo, por ello que en este tiempo la caída será casi nula, y cuando está en OFF la caída de tensión es la de alimentación al igual que en el caso del polo S1.

Figura 6.23 - V_{CE} en S11_AFigura 6.24 - V_{CE} en S2_A

En los Polos S1 y S2 existe una pequeña región de incertidumbre, que es cuando S1 y S22 están en OFF para el polo S1 y cuando S2 y S11 están en OFF para el polo S2. En ese instante que ambos polos están en OFF la tensión que existe entre el colector del inferior y emisor del superior no será fija y por ello que existe una región de incertidumbre.

6.2. PRUEBAS SPACE VECTOR MODULATION (SVM)

Antes de conectar la placa de potencia con el DSP de control comprobaremos que las salidas PWM del DSP son correctas. Para ello tomaremos como referencia la onda PWM que gobernará el transistor S1 de la rama A y la muestrearemos junto a las demás ondas PWM.

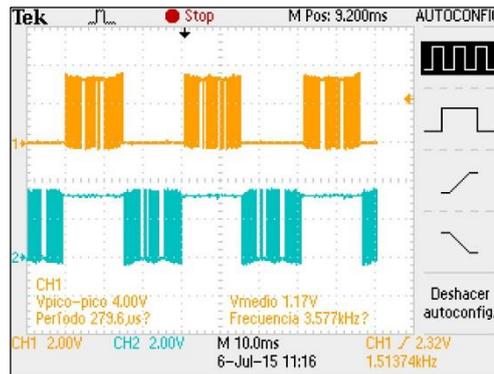


Figura 6.25 – S1_A y S22_A en SVM

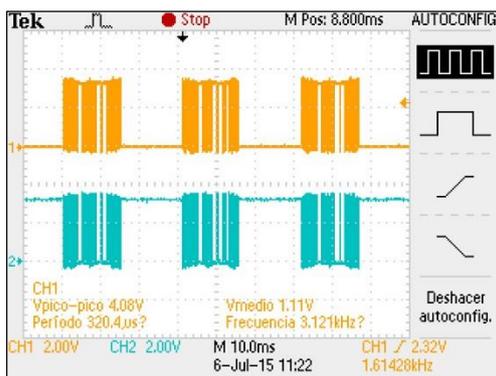


Figura 6.26 – S1_A y S11_A en SVM

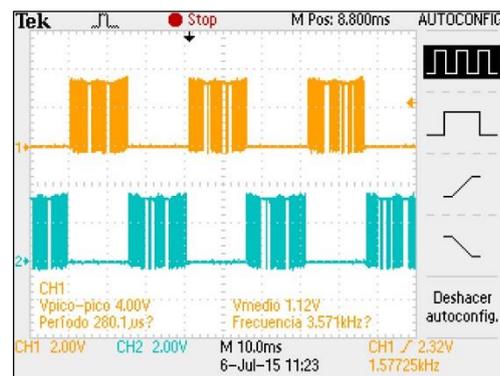


Figura 6.27 – S1_A y S2_A en SVM

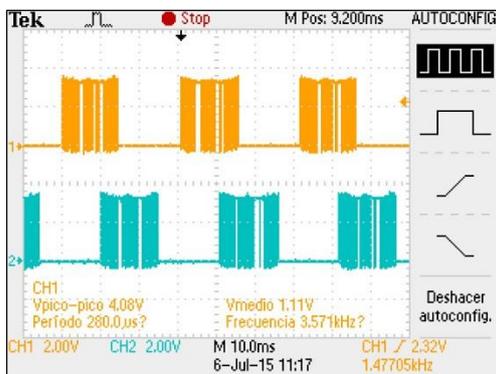


Figura 6.28 – S1_A y 1B en SVM

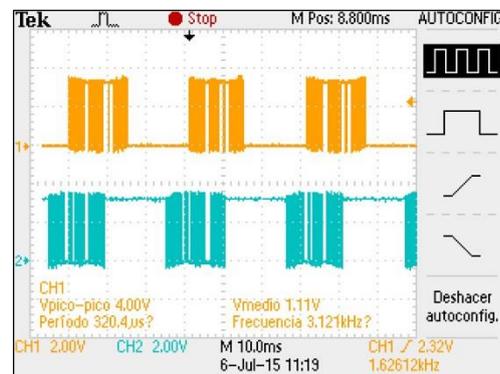


Figura 6.29 – S1_A y S22_B en SVM

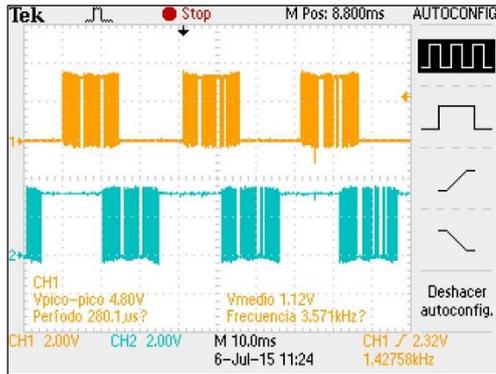


Figura 6.30 – S1_A y S11_B en SVM

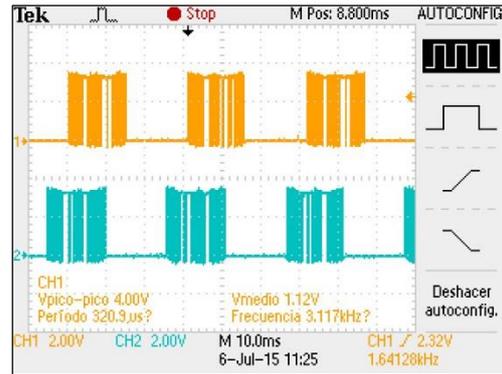


Figura 6.31 – S1_A y S2_B en SVM

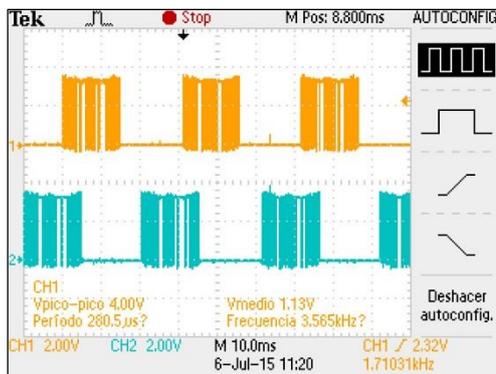


Figura 6.32 – S1_A y S1_C en SVM

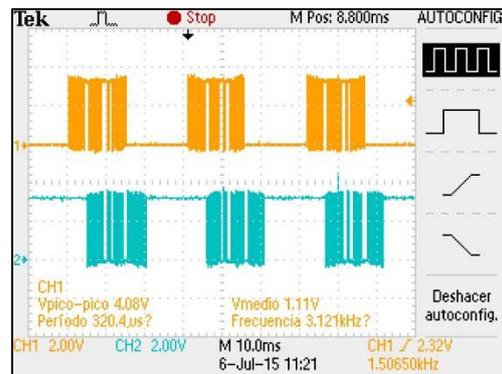


Figura 6.33 – S1_A y S22_C en SVM

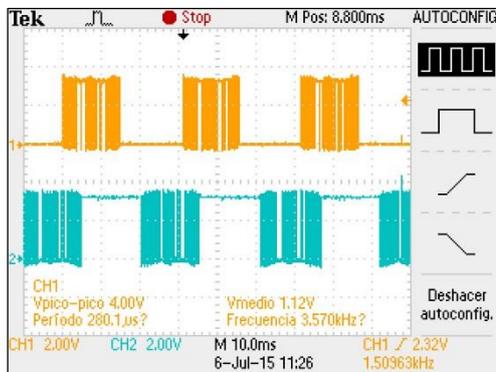


Figura 6.34 – S1_A y S11_C en SVM

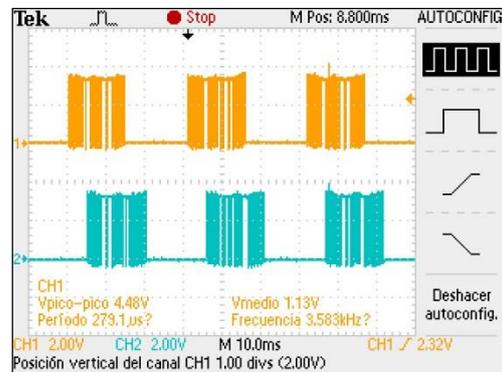


Figura 6.35 – S1_A y S2_C en SVM

En este caso también podemos visualizar cómo se comporta la salida del driver de control que gobierna el transistor S1_A.

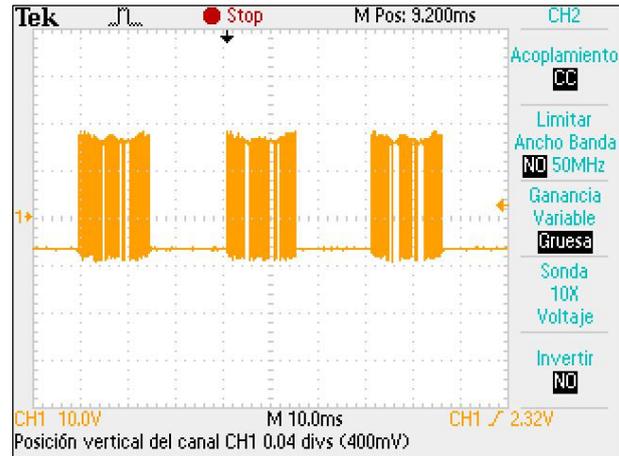


Figura 6.36 – Tensión V_{EB} de S1_A con SVM

En este caso se mostrarán cuáles serán las caídas de tensión V_{CE} para cada transistor cuando aplicamos esta modulación.

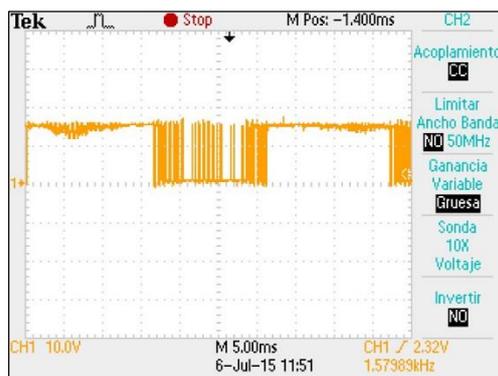


Figura 6.37 – Tensión V_{CE} en S1_A con SVM

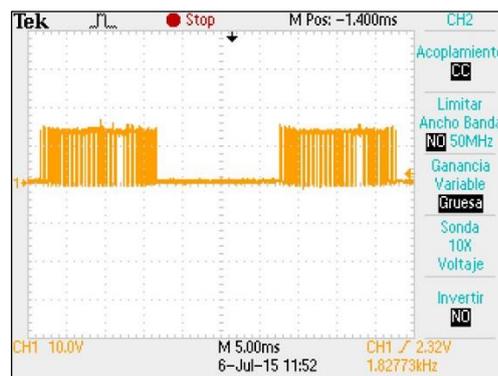


Figura 6.38 – Tensión V_{CE} en S22_A con SVM

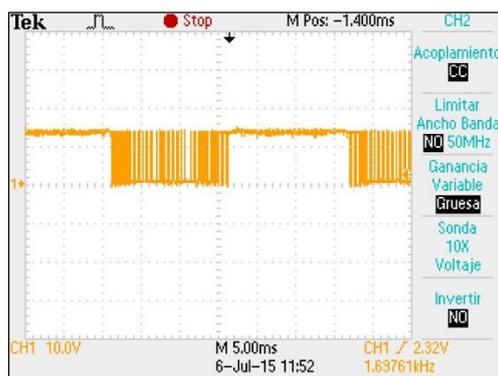


Figura 6.39 – Tensión V_{CE} en S1A con SVM

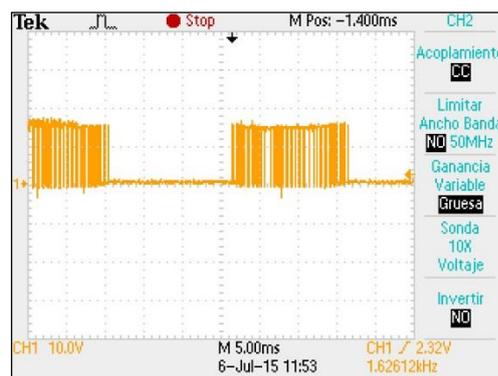


Figura 6.40 – Tensión V_{CE} en S2_A con SVM

En primer lugar analizaremos cuales son las tensiones fase-neutro para la frecuencia con lo que se han mostrado las figuras anteriores y posteriormente se mostrarán las tensiones fase-fase para diferentes frecuencias y diferentes índices de modulación.

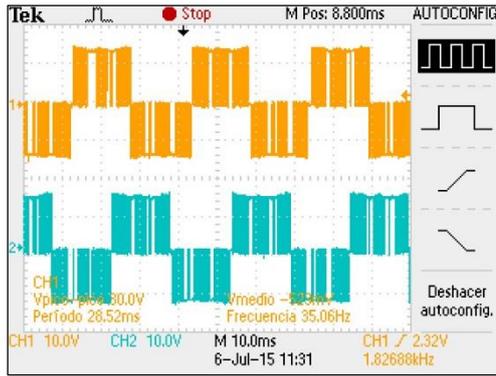


Figura 6.41 – V_{AN} y V_{BN} con SVM

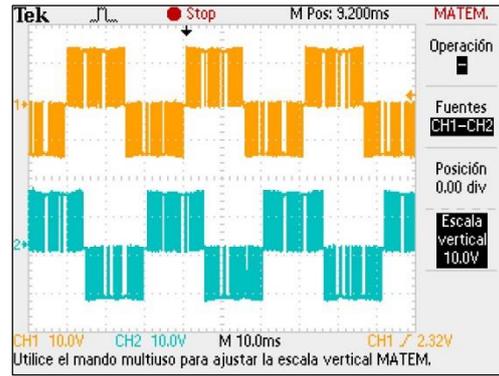


Figura 6.42 – V_{AN} y V_{CN} con SVM

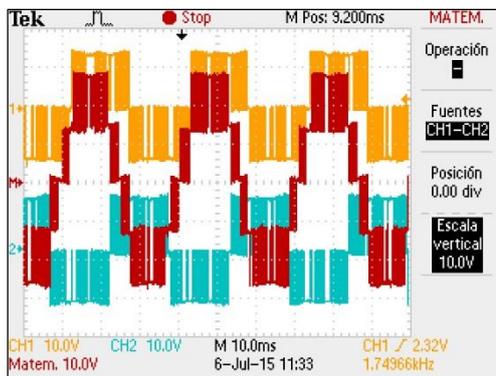


Figura 6.43 – V_{AN} , V_{BN} y V_{AB} matemática con SVM

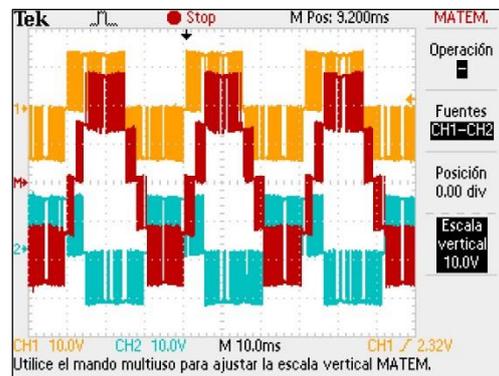


Figura 6.44 – V_{AN} , V_{BN} y V_{AC} matemática con SVM

En primer lugar probaremos el correcto funcionamiento para cuatro frecuencias de la señal portadora. Se han seleccionado frecuencias bajas para que se observen claramente las conmutaciones producida por la señal moduladora. Para variar la frecuencia potadora debemos variar el tiempo configurable en el bloque que genera la señal rampa que se reinicia al llegar a 359.

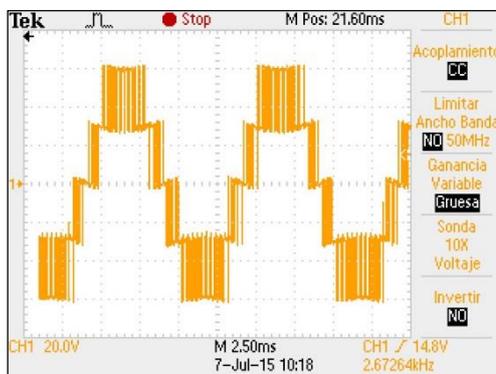


Figura 6.45 – V_{AB} en SVM con freq = 100Hz

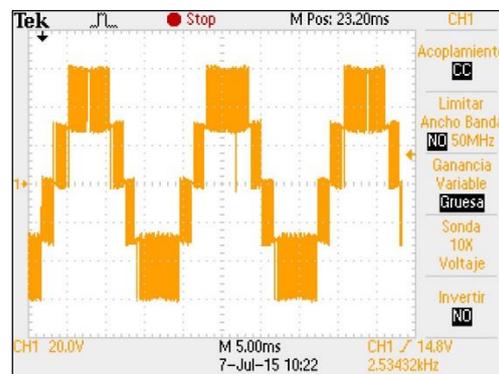


Figura 6.46 – V_{AB} en SVM con freq = 50 Hz

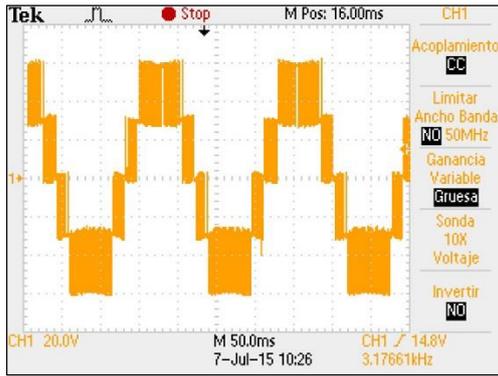


Figura 6.47 - V_{AB} en SVM con $freq = 5Hz$

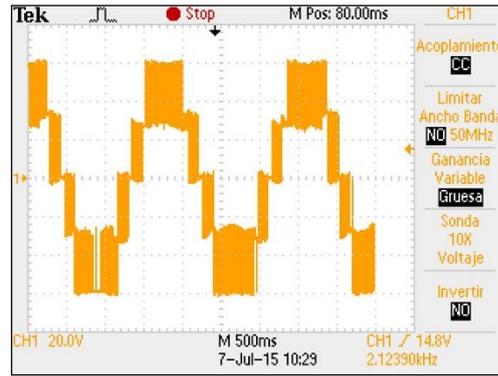


Figura 6.48 - V_{AB} en SVM con $freq = 0.5Hz$

Por último se mostrarán los efectos de variar el índice de modulación.

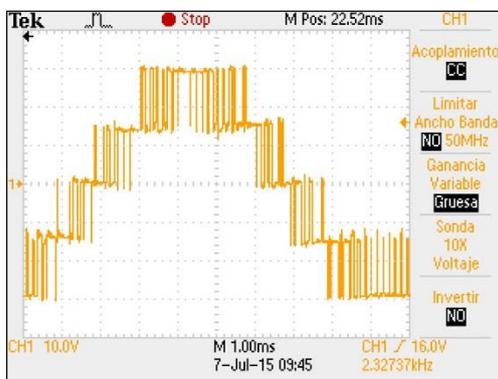


Figura 6.49 - Periodo completo de V_{AB} con $m=0.9$ en SVM

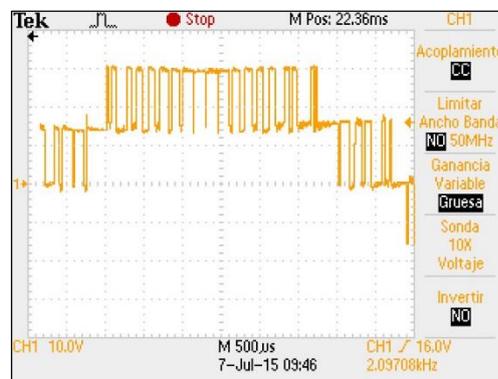


Figura 6.50 - Nivel de tensión más alto de V_{AB} con $m=0.9$ en SVM

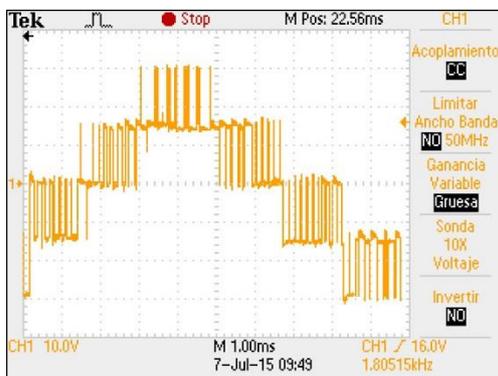


Figura 6.51 - Periodo completo de V_{AB} con $m=0.6$ en SVM

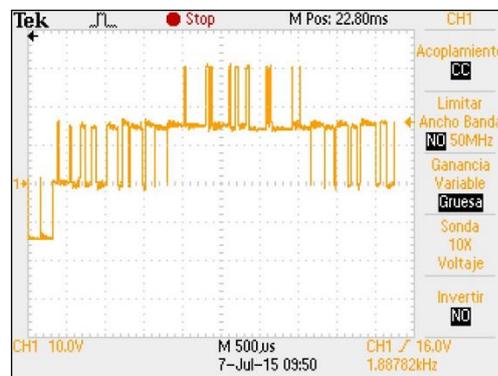


Figura 6.52 - Nivel de tensión más alto de V_{AB} con $m=0.9$ en SVM

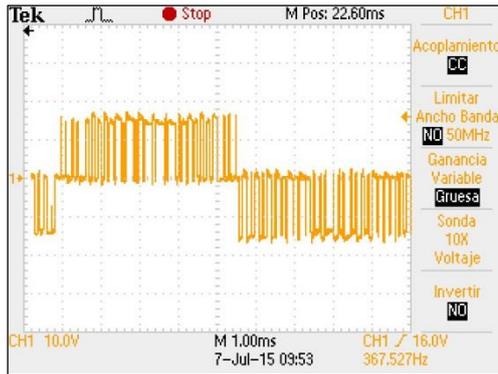


Figura 6.53 – Periodo completo de V_{AB} con $m=0.4$ en SVM

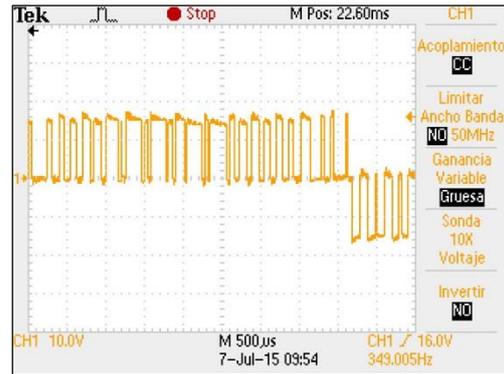


Figura 6.54 – Nivel de tensión más alto de V_{AB} con $m=0.4$ en SVM

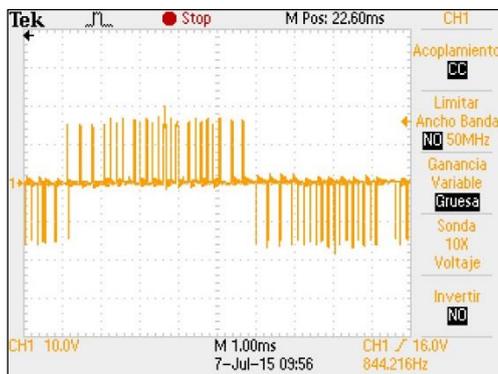


Figura 6.55 – Periodo completo de V_{AB} con $m=0.1$ en SVM



Figura 6.56 – Nivel de tensión más alto de V_{AB} con $m=0.1$ en SVM

Como se puede ver en las 8 figuras anteriores y tal y como habíamos comentado, un índice de modulación menor que 0,5 hace que la tensión de salida tenga solo 3 niveles (tensiones de fase) y no 5. Por otro lado si comparamos las Figuras 6.50 y 6.52 vemos que en la Figura 6.50 el nivel superior dura más tiempo y los polos están más tiempo a la tensión superior que a la inferior, a diferencia de la Figura 6.52, que el nivel superior permanece durante menos tiempo y los polos permanecen más tiempo en la tensión inferior. Lo mismo ocurre si comparamos las Figuras 6.54 y 6.56.

Además se puede observar claramente en la Figura 6.50 que esta modulación vectorial hace que los tiempos de aplicación varíen en cada instante para reducir el contenido armónico de la señal, en la figura mencionada, en la zona superior se ve como en la parte media predomina el estado superior y en los extremos predomina el estado inferior.

También podemos comprobar cómo el funcionamiento de la variación de frecuencia fundamental es la correcta. Para ello se ha configurado una frecuencia portadora fija y se ha ido variando la frecuencia moduladora en tres valores, se ha ampliado la zona superior de la gráfica para ver correctamente el efecto de esta variación.

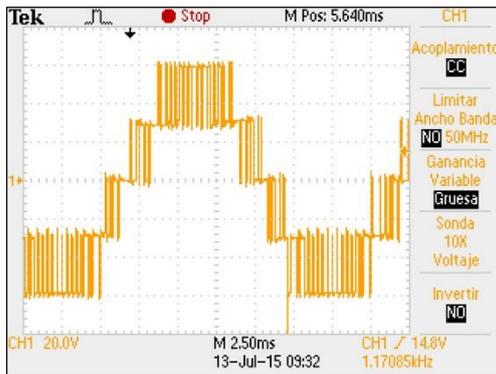


Figura 6.57 - Frecuencia moduladora baja (1)

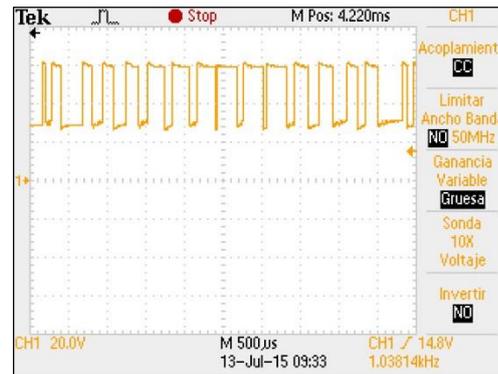


Figura 6.58 - Frecuencia moduladora baja (2)

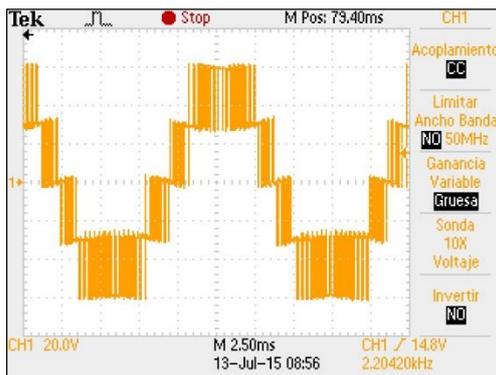


Figura 6.59 - Frecuencia moduladora media (1)

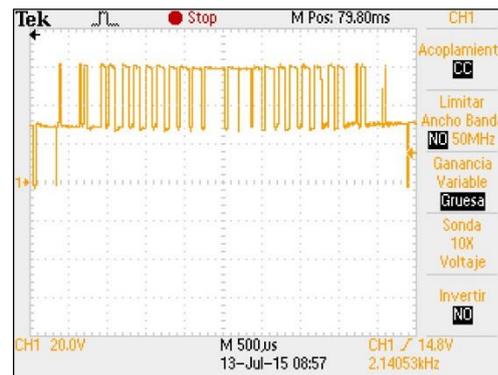


Figura 6.60 - Frecuencia moduladora media (2)

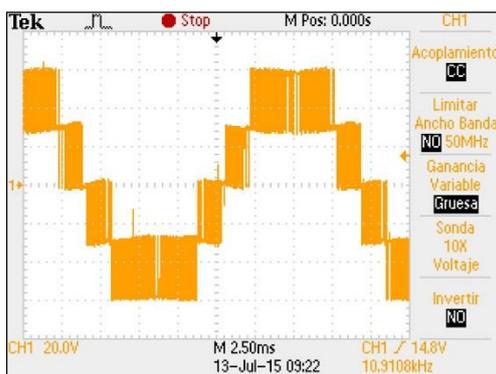


Figura 6.61 - Frecuencia moduladora alta (1)

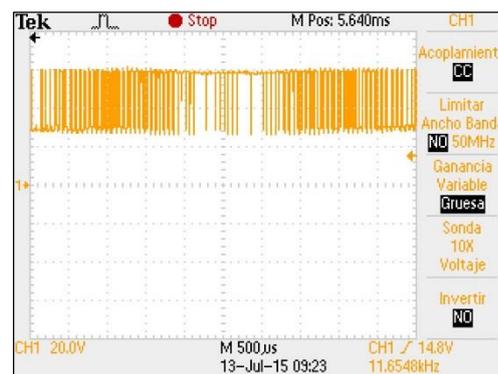


Figura 6.62 - Frecuencia moduladora alta (2)

Podemos concluir que el funcionamiento del código es el esperado ya que todas las variaciones, tanto de frecuencias, como de amplitudes, responden correctamente.



7. CONCLUSIONES

A continuación, se presentan las conclusiones más importantes que se extraen del presente proyecto:

- El análisis del estado actual de la técnica permite afirmar que las técnicas de conversión multinivel se han consolidado como una opción competitiva para la conversión de energía en el rango de media-alta.
- La estrategia de conmutación propuesta para el convertidor NPC se puede entender como la extensión de la modulación SVM de dos a tres niveles. La estrategia es simple, y otorga gran capacidad de control al regulador. Esta modulación dispone de la capacidad de regular la tensión a secuencia cero mediante las relaciones de conducción a secuencia cero.
- El entorno de simulación y programación (MatLab-Simulink) destaca por su facilidad de uso y su versatilidad. Puesto que el DSP empleado para la implementación del control admite su programación desde MatLab-Simulink, los ficheros fuente de simulación pueden utilizarse, con pequeños cambios, como ficheros fuente de programación del control. Esta posibilidad facilita mucho el trabajo del diseñador y reduce considerablemente los tiempos de desarrollo de un nuevo control.
- Debido al análisis de resultados de la simulación y a la programación de algoritmo de modulación en un procesador de señales digitales (DSP), se logró realizar una verificación real en un modelo de laboratorio de un inversor trifásico multinivel de tres niveles con fijación por diodos (NPC). Las formas de onda y los valores prácticos obtenidos demuestran que el modelo funciona correctamente.
- La estructura montada es adecuada para cumplir unas mínimas condiciones de seguridad cuando se trabaje a altas tensiones debido a la peligrosidad del uso de condensadores electrolíticos, en los cuales una sobretensión puede producir una explosión del condensador.
- Una conclusión muy importante es la peligrosidad de usar el equipo sin leer las correspondientes instrucciones que aparecen escritas en la



estructura. Esto es debido a que los polos de potencia están encendidos por defecto cuando se aplica tensión al driver, y si se aplica voltaje a la etapa de potencia antes que se haya lanzado el programa estaremos cortocircuitando la fuente de alimentación.



8. BIBLIOGRAFÍA

- [1] TEXAS INSTRUMENTS. "TMS320x2806x Piccolo". Technical Reference Manual. Año 2014
- [2] TEXAS INSTRUMENTS. "TMS320x280x, 2801x, 2804x Enhanced Pulse Width Modulator (ePWM) Module". Reference Guide. Año 2009.
- [3] Darshan.S.Patel. "Simulation of Multilevel Inverter Using PSIM". Año 2005.
- [4] Luís Miguel Faria Miranda. "Veículo eléctrico com interface para a rede eléctrica. Grid interface for an electric vehicle". Año 2011.
- [5] Julio César Arias Flórez. "Implementación de un modulador por vector espacial (SVM - 2D) para un inversor multinivel de tres niveles con fijación por diodos (NPC)". Año 2015.
- [6] Salvador Alepuz Menéndez. "Aportación al Control del Convertidor CC/CA de Tres Niveles". Año 2004.
- [7] TEXAS INSTRUMENTS. "SEMITOP® Technical Information". Año 2008.
- [8] TEXAS INSTRUMENTS. "3L NPC & TNPC Topology". Año 2009.
- [9] TEXAS INSTRUMENTS. "Application Manual Power Semiconductors". Año 2011.
- [10] TEXAS INSTRUMENTS. "Datasheet Board 1 SKYPER 32PRO R". Año 2010.
- [11] TEXAS INSTRUMENTS. "Datasheet MLI SKYPER® 32 PRO R". Año 2012.



- [12] TEXAS INSTRUMENTS. “Datasheet SK75MLI066T”. Año 2009.
- [13] TEXAS INSTRUMENTS. “Datasheet SKYPER® 32 PRO R”. Año 2007.
- [14] Revista Iberoamericana de Automatica e Informatica Industrial. “Modulación Vectorial de Inversores Multinivel con Diodos de Enclavamiento con Balance del Bus de Continua”. Año 2009.
- [15] Julio Cesar Arias Florez. “Implementación de un Modulador por Vector Espacial (SVM-2D) para un Inversor Multinivel de Tres Niveles con Fijación por Diodos (NPC)”. Año 2015.



9. ANEXOS

9.1. ANEXO 1: PROGRAMA main.c CODE COMPOSER STUDIO

```
//-----
//
// FILE:          Inversor_Multinivel-Main.C
//
// Descripcion: Control en lazo Abierto de los 12 transistores de un
// inversor multi-nivel
//
// Version:       1.0
//
// Tarjeta:     TMS320F28069
//
// Tipo:        Docking-Sth USB_EMU
//
//-----
//
// ALGUNAS NOTAS SOBRE ESTE PROYECTO
//
// Todo este proyecyto esta formado por diversos ficheros, los mas
// importantes son:
//      "Inversor_Multinivel-Main.C" - este archivo
//      - Inicializacion de las configuraciones, configuraciones
// externas
//      - Gestion de la aplicacion
//      - Lazos de control y planificacion de tareas
//      "Inversor_Multinivel-DevInit_F28xxx.C
//      - Inicializacion del despositivo, ej. Reloj, PLL, WD, mapa
// GPIO
//      - Habilitacion de diversos relojes
//
//-----
//
// Este programa generará 12 señales PWM configuradas para controlar en
// lazo abierto
// un inversor multi-nivel trifasico. Solo se podra elegir la frecuencia
// de trabajo
//-----
//
// #include "PeripheralHeaderIncludes.h"
// #include "F2806x_EPwm_defines.h"          // Definiciones para
// inicializaciones
//
// %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
// %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
// %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```



```

// FUNCIONES
//%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

// ----- GENERALES -----
-----
void DeviceInit(void);
void SCIA_Init();
void SerialHostComms();
void InitFlash();
void MemCopy();

// ----- USUARIO -----
-----

//%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
// DECLARACION DE VARIABLES
//%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

// ----- GENERALES -----
-----
int16 SerialCommsTimer;
int16 CommsOKflg;
// Usadas para ejecuciones en segundo plano con memoria flash, y ISR con
memoria RAM
extern Uint16 *RamfuncsLoadStart, *RamfuncsLoadEnd, *RamfuncsRunStart;

// ----- USUARIO -----
-----
// Dado el tiempo t en ns, el dead time sigue la ecuacion: RED = FED =
0.06*t
// Esta ecuacion es valida cuando el PLL esta configurado a 0xC (60MHz)
Uint16 RED_delay=6; // 100ns
Uint16 FED_delay=6; // 100ns

//%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
// VARIABLES PARA SOPORTE GUI
//%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

// ----- FRAMEWORK -----
-----

// Establece una cantidad limite para los controles externos de GUI
// Incrementar si es necesario
int16 *varSetTxtList[8]; //8 textbox controlled
variables
int16 *varSetBtnList[8]; //8 button controlled
variables

```



```

int16 *varSetSliderList[8];           //8 slider controlled
variables
int16 *varGetList[8];                 //8 variables
sendable to GUI
int16 *arrayGetList[8];               //8 arrays sendable to GUI

//%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
// CODIGO PRINCIPAL
//%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

void main(void)
{
//=====
//   Inicialización del dispositivo
//=====

    DeviceInit();           // Inicialización y mapa de entradas/salidas
    SCIA_Init();            // Inicializa periférico Serial Comms A

// Solo se ejecutará al seleccionar ejecución con memoria FLASH
#ifdef FLASH
// Copy time critical code and Flash setup code to RAM
// The RamfuncsLoadStart, RamfuncsLoadEnd, and RamfuncsRunStart
// symbols are created by the linker. Refer to the linker files.
    MemCopy(&RamfuncsLoadStart, &RamfuncsLoadEnd, &RamfuncsRunStart);

// Call Flash Initialization to setup flash waitstates
// This function must reside in RAM
    InitFlash(); // Call the flash wrapper init function
#endif //(FLASH)

// ----- USUARIO -----
// -----
// Se podran modificar estas variables en cada ejecución en función de
// las necesidades

// El periodo se selecciona en función de la frecuencia deseada
// siguiendo esta ecuación
// period=(60*10^6)/2*frecuencia, estando la frecuencia deseada en Hz
// Ecuación valida cuando el PLL esta configurado a 60MHz 0xC
#define period 3000 // 1kHz
//Frecuencia max - 50 KHz (limitado por driver), Frecuencia min - 230 Hz
// (limitado por DSP)

//=====
//   Inicialización de las señales PWM
//=====

```



```
//=====
//   EPwm1.A es transistor S1 de la rama A (PIN 00)
//   EPwm1.B es transistor S11 de la rama A (PIN 01)
//=====

//Registros TIME-BASE
EPwm1Regs.TBPRD = period/2;
//Periodo de la señal
EPwm1Regs.TBPHS.all = 0;
//Inicializacion del registro Time-Base Phase
EPwm1Regs.TBPHS.half.TBPHS = 0;
//Inicializacion del registro Time-Base Phase, angulo de desfase
EPwm1Regs.TBCTR = 0;
//Inicializacion del registro Time-Base Counter
EPwm1Regs.TBCTL.bit.PRDL = TB_IMMEDIATE; //Inicio Inmediato,
lecturas directas al registro activo
EPwm1Regs.TBCTL.bit.CTRMODE = TB_COUNT_UPDOWN; //Modo ascendente-
descendente, triangular
EPwm1Regs.TBCTL.bit.PHSEN = TB_DISABLE;
//Deshabilitada sincronización
EPwm1Regs.TBCTL.bit.SYNCOSEL = TB_CTR_ZERO; //Envia señal
de sincronismo cuando pase por CTR=0

//Registro Counter-Compare
EPwm1Regs.CMPCTL.bit.SHDWAMODE = CC_SHADOW; //Modo
Shadow, operando como un doble buffer
EPwm1Regs.CMPCTL.bit.SHDWBMODE = CC_SHADOW; //Modo
Shadow, operando como un doble buffer
EPwm1Regs.CMPCTL.bit.LOADAMODE = CC_CTR_ZERO; //Carga en CTR=Zero
EPwm1Regs.CMPCTL.bit.LOADBMODE = CC_CTR_ZERO; //Carga en CTR=Zero

//Establecemos ciclo de servicio
EPwm1Regs.CMPA.half.CMPA = period/4; //25%
EPwm1Regs.CMPB = period/4;

//Acciones PWM
EPwm1Regs.AQCTLA.bit.ZRO = AQ_SET; //PWM=1
EPwm1Regs.AQCTLA.bit.CAU = AQ_CLEAR; //PWM=0

EPwm1Regs.AQCTLB.bit.ZRO = AQ_CLEAR; //PWM=0
EPwm1Regs.AQCTLB.bit.CBU = AQ_SET; //PWM=1

//Configuracion de tiempos muertos
EPwm1Regs.DBCTL.bit.IN_MODE = DBA_ALL; //Tiempo
muerto de subida y bajada para señales A y B
EPwm1Regs.DBCTL.bit.OUT_MODE = DB_FULL_ENABLE; //Tiempo
muerto de subida y bajada para señales A y B
EPwm1Regs.DBCTL.bit.POLSEL = DB_ACTV_HIC; //Activamos
High Complementary (AHC)
EPwm1Regs.DBRED = RED_delay;
//Tiempo muerto en flanco de subida
EPwm1Regs.DBFED = FED_delay;
//Tiempo muerto en flanco de bajada
```



```

//=====
//   EPwm2.A es transistor S22 de la rama A (PIN 02)
//   EPwm2.B es transistor S2 de la rama A (PIN 03)
//=====

//Registros TIME-BASE
EPwm2Regs.TBPRD = period/2;
EPwm2Regs.TBPHS.all = 0;
EPwm2Regs.TBPHS.half.TBPHS = period/2;           //Angulo de
desfase 180º con respecto PWM1
EPwm2Regs.TBCTR = 0;
EPwm2Regs.TBCTL.bit.PRDLN = TB_IMMEDIATE;
EPwm2Regs.TBCTL.bit.CTRMODE = TB_COUNT_UPDOWN; //Modo ascendente-
descendente, triangular
EPwm2Regs.TBCTL.bit.PHSDIR = TB_DOWN;           //Cuando
sincroniza, empieza la señal triangular descendente
EPwm2Regs.TBCTL.bit.PHSEN = TB_ENABLE;         //Habilitamos
sincronizacion
EPwm2Regs.TBCTL.bit.SYNCSEL = TB_SYNC_IN;      //Señal de
salida de sincronismo es la misma que la de entrada

//Se realimenta la señal CTR=0 de la rama PWM1 como señal
de sincronismo

//Registro Counter-Compare
EPwm2Regs.CMPCTL.bit.SHDWAMODE = CC_SHADOW;
EPwm2Regs.CMPCTL.bit.SHDWBMODE = CC_SHADOW;
EPwm2Regs.CMPCTL.bit.LOADAMODE = CC_CTR_ZERO;
EPwm2Regs.CMPCTL.bit.LOADBMODE = CC_CTR_ZERO;

//Establecemos ciclo de servicio
EPwm2Regs.CMPA.half.CMPA = period/4;
EPwm2Regs.CMPB = period/4;

//Acciones PWM
EPwm2Regs.AQCTLA.bit.ZRO = AQ_CLEAR;
EPwm2Regs.AQCTLA.bit.CAU = AQ_SET;

EPwm2Regs.AQCTLB.bit.ZRO = AQ_SET;
EPwm2Regs.AQCTLB.bit.CBU = AQ_CLEAR;

//Configuracion de tiempos muertos
EPwm2Regs.DBCTL.bit.IN_MODE = DBA_ALL;
EPwm2Regs.DBCTL.bit.OUT_MODE = DB_FULL_ENABLE;
EPwm2Regs.DBCTL.bit.POLSEL = DB_ACTV_HIC;
EPwm2Regs.DBRED = RED_delay;
EPwm2Regs.DBFED = FED_delay;

//=====
//   EPwm3.A es transistor S1 de la rama B (PIN 04)
//   EPwm3.B es transistor S11 de la rama B (PIN 05)
//=====

//Registros TIME-BASE
EPwm3Regs.TBPRD = period/2;

```



```
EPwm3Regs.TBPHS.all = 0;
EPwm3Regs.TBPHS.half.TBPHS = period/3;           //Angulo de
desfase 120º con respecto PWM1
EPwm3Regs.TBCTR = 0;
EPwm3Regs.TBCTL.bit.PRDL = TB_IMMEDIATE;
EPwm3Regs.TBCTL.bit.CTRMODE = TB_COUNT_UPDOWN;
EPwm3Regs.TBCTL.bit.PHSDIR = TB_DOWN;           //Cuando
sincroniza, empieza la señal triangular descendente
EPwm3Regs.TBCTL.bit.PHSEN = TB_ENABLE;
EPwm3Regs.TBCTL.bit.SYNCOSEL = TB_CTR_ZERO;     //Envia señal
de sincronismo cuando pase por CTR=0

//Las posteriores señales sincronizan siguiendo como
referencia a PWM3

//Registro Counter-Compare
EPwm3Regs.CMPCTL.bit.SHDWAMODE = CC_SHADOW;
EPwm3Regs.CMPCTL.bit.SHDWBMODE = CC_SHADOW;
EPwm3Regs.CMPCTL.bit.LOADAMODE = CC_CTR_ZERO;
EPwm3Regs.CMPCTL.bit.LOADBMODE = CC_CTR_ZERO;

//Establecemos ciclo de servicio
EPwm3Regs.CMPA.half.CMPA = period/4;
EPwm3Regs.CMPB = period/4;

//Acciones PWM
EPwm3Regs.AQCTLA.bit.ZRO = AQ_SET;
EPwm3Regs.AQCTLA.bit.CAU = AQ_CLEAR;

EPwm3Regs.AQCTLB.bit.ZRO = AQ_CLEAR;
EPwm3Regs.AQCTLB.bit.CBU = AQ_SET;

//Configuración de tiempos muertos
EPwm3Regs.DBCTL.bit.IN_MODE = DBA_ALL;
EPwm3Regs.DBCTL.bit.OUT_MODE = DB_FULL_ENABLE;
EPwm3Regs.DBCTL.bit.POLSEL = DB_ACTV_HIC;
EPwm3Regs.DBRED = RED_delay;
EPwm3Regs.DBFED = FED_delay;

//=====
// EPwm4.A es transistor S22 de la rama B (PIN 06)
// EPwm4.B es transistor S2 de la rama B (PIN 07)
//=====

//Registros TIME-BASE
EPwm4Regs.TBPRD = period/2;
EPwm4Regs.TBPHS.half.TBPHS = period/2;         //Angulo de
desfase 180º con respecto PWM3
EPwm4Regs.TBCTR = 0;
EPwm4Regs.TBCTL.bit.PRDL = TB_SHADOW;
EPwm4Regs.TBCTL.bit.CTRMODE = TB_COUNT_UPDOWN;
EPwm4Regs.TBCTL.bit.PHSDIR = TB_DOWN;
EPwm4Regs.TBCTL.bit.PHSEN = TB_ENABLE;
EPwm4Regs.TBCTL.bit.SYNCOSEL = TB_SYNC_IN;     //Señal de
salida de sincronismo es la misma que la de entrada
```



//Se realimenta la señal CTR=0 de la rama PWM3 como señal de sincronismo

//Registro Counter-Compare

```
EPwm4Regs.CMPCTL.bit.SHDWAMODE = CC_SHADOW;
EPwm4Regs.CMPCTL.bit.SHDWBMODE = CC_SHADOW;
EPwm4Regs.CMPCTL.bit.LOADAMODE = CC_CTR_ZERO;
EPwm4Regs.CMPCTL.bit.LOADBMODE = CC_CTR_ZERO;
```

//Establecemos ciclo de servicio

```
EPwm4Regs.CMPA.half.CMPA = period/4;
EPwm4Regs.CMPB = period/4;
```

//Acciones PWM

```
EPwm4Regs.AQCTLA.bit.ZRO = AQ_CLEAR;
EPwm4Regs.AQCTLA.bit.CAU = AQ_SET;
```

```
EPwm4Regs.AQCTLB.bit.ZRO = AQ_SET;
EPwm4Regs.AQCTLB.bit.CBU = AQ_CLEAR;
```

//Configuracion de tiempos muertos

```
EPwm4Regs.DBCTL.bit.IN_MODE = DBA_ALL;
EPwm4Regs.DBCTL.bit.OUT_MODE = DB_FULL_ENABLE;
EPwm4Regs.DBCTL.bit.POLSEL = DB_ACTV_HIC;
EPwm4Regs.DBRED = RED_delay;
EPwm4Regs.DBFED = FED_delay;
```

//=====

// EPwm5A es transistor S1 de la rama C (PIN 8)

// EPwm5B es transistor S11 de la rama C (PIN 9)

//=====

//Registros TIME-BASE

```
EPwm5Regs.TBPRD = period/2;
```

```
EPwm5Regs.TBPHS.half.TBPHS = period/3;
```

//Angulo de

desfase 120° con respecto PWM3

```
EPwm5Regs.TBCTR = 0;
```

```
EPwm5Regs.TBCTL.bit.PRDLT = TB_SHADOW;
```

```
EPwm5Regs.TBCTL.bit.CTRMODE = TB_COUNT_UPDOWN;
```

```
EPwm5Regs.TBCTL.bit.PHSDIR = TB_DOWN;
```

```
EPwm5Regs.TBCTL.bit.PHSEN = TB_ENABLE;
```

```
EPwm5Regs.TBCTL.bit.SYNCSEL = TB_CTR_ZERO;
```

//Envia señal

de sincronismo cuando pase por CTR=0

//Las posteriores señales sincronizan siguiendo como referencia a PWM5

//Registro Counter-Compare

```
EPwm5Regs.CMPCTL.bit.SHDWAMODE = CC_SHADOW;
```

```
EPwm5Regs.CMPCTL.bit.SHDWBMODE = CC_SHADOW;
```

```
EPwm5Regs.CMPCTL.bit.LOADAMODE = CC_CTR_ZERO;
```

```
EPwm5Regs.CMPCTL.bit.LOADBMODE = CC_CTR_ZERO;
```

//Establecemos ciclo de servicio



```
EPwm5Regs.CMPA.half.CMPA = period/4;
EPwm5Regs.CMPB = period/4;

//Acciones PWM
EPwm5Regs.AQCTLA.bit.ZRO = AQ_SET;
EPwm5Regs.AQCTLA.bit.CAU = AQ_CLEAR;

EPwm5Regs.AQCTLB.bit.ZRO = AQ_CLEAR;
EPwm5Regs.AQCTLB.bit.CBU = AQ_SET;

//Configuracion de tiempos muertos
EPwm5Regs.DBCTL.bit.IN_MODE = DBA_ALL;
EPwm5Regs.DBCTL.bit.OUT_MODE = DB_FULL_ENABLE;
EPwm5Regs.DBCTL.bit.POLSEL = DB_ACTV_HIC;
EPwm5Regs.DBRED = RED_delay;
EPwm5Regs.DBFED = FED_delay;

//=====
// EPwm6.A es transistor S22 de la rama C (PIN 10)
// EPwm6.B es transistor S2 de la rama C (PIN 11)
//=====

//Registros TIME-BASE
EPwm6Regs.TBPRD = period/2;
EPwm6Regs.TBPHS.half.TBPHS = period/2; //Angulo de
desfase 120º con respecto PWM3
EPwm6Regs.TBCTR = 0;
EPwm6Regs.TBCTL.bit.PRDL D = TB_SHADOW;
EPwm6Regs.TBCTL.bit.CTRMODE = TB_COUNT_UPDOWN;
EPwm6Regs.TBCTL.bit.PHSDIR = TB_DOWN;
EPwm6Regs.TBCTL.bit.PHSEN = TB_ENABLE;
EPwm6Regs.TBCTL.bit.SYNCOSEL = TB_SYNC_IN; //Señal de
salida de sincronismo es la misma que la de entrada

//Se realimenta la señal CTR=0 de la rama PWM3 como señal
de sincronismo

//Usariamos esta señal de sincronismo para una posible
futura señal PWM7

//Registro Counter-Compare
EPwm6Regs.CMPCTL.bit.SHDWAMODE = CC_SHADOW;
EPwm6Regs.CMPCTL.bit.SHDWBMODE = CC_SHADOW;
EPwm6Regs.CMPCTL.bit.LOADAMODE = CC_CTR_ZERO;
EPwm6Regs.CMPCTL.bit.LOADBMODE = CC_CTR_ZERO;

//Establecemos ciclo de servicio
EPwm6Regs.CMPA.half.CMPA = period/4;
EPwm6Regs.CMPB = period/4;

//Acciones PWM
EPwm6Regs.AQCTLA.bit.ZRO = AQ_CLEAR;
EPwm6Regs.AQCTLA.bit.CAU = AQ_SET;

EPwm6Regs.AQCTLB.bit.ZRO = AQ_SET;
```



```

EPwm6Regs.AQCTLB.bit.CBU = AQ_CLEAR;

//Configuracion de tiempos muertos
EPwm6Regs.DBCTL.bit.IN_MODE = DBA_ALL;
EPwm6Regs.DBCTL.bit.OUT_MODE = DB_FULL_ENABLE;
EPwm6Regs.DBCTL.bit.POLSEL = DB_ACTV_HIC;
EPwm6Regs.DBRED = RED_delay;
EPwm6Regs.DBFED = FED_delay;

//=====
//      LAZO INFINITO DE EJECUCION
//=====
for(;;)
{
    //En este programa no es necesaria ninguna accion
}

//FIN DEL CODIGO
}

```

9.2. ANEXO 2: CÓDIGO FUNCIÓN PWM_POLOS_SVM SIMULINK

```

function [CMPA1, CMPA2, CMPA3, CMPA4, CMPA5, CMPA6] =
PWM_Polos_SVM(angulo, m, TBPRD)
%% Cálculo de número de muestras
% n=360/angulo;

% %Cálculo de TBPRD
% TBPRD=75e+6/(n*f);

% Cálculo del Sector
theta = angulo*pi/180;
N = floor(theta/(pi/3))+1;
alpha = rem(theta,pi/3);

% Calculo de la Región.
a = ceil(m*2*cos(alpha+pi/6));
b = ceil(m*2*cos(pi/6-alpha));
h = floor(m*2*sin(alpha))+1;
Reg = -a+(b^2-b+1)+h;
SubReg = floor(alpha/(pi/6));
%% Cálculo de los ePWMS
% Tiempos programados en la Tesis.
% SECTOR 1
if (N == 1) && (Reg==1) && (SubReg==0)
    ta = 2*1*m*sin((pi/3)-alpha);
    tb = (1-2*1*m*sin((pi/3)+alpha));
    tc = 2*1*m*sin(alpha);
    CMPA1 = (ta/2+tc+tb)*TBPRD;

```



```
    CMPA2 = TBPRD;
    CMPA3 = TBPRD;
    CMPA4 = 0;
    CMPA5 = (ta/2)*TBPRD;
    CMPA6 = (ta/2+tc)*TBPRD;
elseif (N == 1) && (Reg==1) && (SubReg==1)
    ta = 2*1*m*sin((pi/3)-alpha);
    tb = (1-2*1*m*sin((pi/3)+alpha));
    tc = 2*1*m*sin(alpha);
    CMPA1 = (tc/2+tb)*TBPRD;
    CMPA2 = (tc/2+tb+ta)*TBPRD;
    CMPA3 = TBPRD;
    CMPA4 = 0;
    CMPA5 = 0;
    CMPA6 = (tc/2)*TBPRD;
elseif (N == 1) && (Reg==2)
    ta = 2*(1-1*m*sin((pi/3)+alpha));
    tb = 2*1*m*sin(alpha);
    tc = (2*1*m*sin((pi/3)-alpha)-1);
    CMPA1 = (ta/2)*TBPRD;
    CMPA2 = TBPRD;
    CMPA3 = TBPRD;
    CMPA4 = 0;
    CMPA5 = (ta/2+tc)*TBPRD;
    CMPA6 = (ta/2+tc+tb)*TBPRD;
elseif (N == 1) && (Reg==3) && (SubReg==0)
    ta = (1-2*1*m*sin(alpha));
    tb = (2*1*m*sin((pi/3)+alpha)-1);
    tc = (1-2*1*m*sin((pi/3)-alpha));
    CMPA1 = (ta/2+tc)*TBPRD;
    CMPA2 = TBPRD;
    CMPA3 = TBPRD;
    CMPA4 = 0;
    CMPA5 = (ta/2)*TBPRD;
    CMPA6 = (ta/2+tc+tb)*TBPRD;
elseif (N == 1) && (Reg==3) && (SubReg==1)
    ta = (1-2*1*m*sin(alpha));
    tb = (2*1*m*sin((pi/3)+alpha)-1);
    tc = (1-2*1*m*sin((pi/3)-alpha));
    CMPA1 = (tc/2)*TBPRD;
    CMPA2 = (tc/2+tb+ta)*TBPRD;
    CMPA3 = TBPRD;
    CMPA4 = 0;
    CMPA5 = 0;
    CMPA6 = (tc/2+tb)*TBPRD;
elseif (N == 1) && (Reg==4)
    ta = (2*1*m*sin(alpha)-1);
    tb = 2*1*m*sin((pi/3)-alpha);
    tc = 2*(1-1*m*sin((pi/3)+alpha));
    CMPA1 = (tc/2)*TBPRD;
    CMPA2 = (tc/2+tb)*TBPRD;
    CMPA3 = TBPRD;
    CMPA4 = 0;
    CMPA5 = 0;
    CMPA6 = (tc/2+tb+ta)*TBPRD;

%% SECTOR 2
elseif (N == 2) && (Reg==1) && (SubReg==0)
    ta = 2*1*m*sin((pi/3)-alpha);
```



```

    tb = (1-2*1*m*sin((pi/3)+alpha));
    tc = 2*1*m*sin(alpha);
    CMPA1 = (tc/2+tb+ta)*TBPRD;
    CMPA2 = (tc/2+tb)*TBPRD;
    CMPA3 = TBPRD;
    CMPA4 = 0;
    CMPA5 = 0;
    CMPA6 = (tc/2)*TBPRD;
elseif (N == 2) && (Reg==1) && (SubReg==1)
    ta = 2*1*m*sin((pi/3)-alpha);
    tb = (1-2*1*m*sin((pi/3)+alpha));
    tc = 2*1*m*sin(alpha);
    CMPA1 = TBPRD;
    CMPA2 = (ta/2+tc+tb)*TBPRD;
    CMPA3 = TBPRD;
    CMPA4 = (ta/2)*TBPRD;
    CMPA5 = 0;
    CMPA6 = (ta/2+tc)*TBPRD;
elseif (N == 2) && (Reg==2)
    ta = 2*(1-1*m*sin((pi/3)+alpha));
    tb = 2*1*m*sin(alpha);
    tc = (2*1*m*sin((pi/3)-alpha)-1);
    CMPA1 = (tc/2+tb)*TBPRD;
    CMPA2 = (tc/2)*TBPRD;
    CMPA3 = TBPRD;
    CMPA4 = 0;
    CMPA5 = 0;
    CMPA6 = (tc/2+tb+ta)*TBPRD;
elseif (N == 2) && (Reg==3) && (SubReg==0)
    ta = (1-2*1*m*sin(alpha));
    tb = (2*1*m*sin((pi/3)+alpha)-1);
    tc = (1-2*1*m*sin((pi/3)-alpha));
    CMPA1 = (tc/2+tb+ta)*TBPRD;
    CMPA2 = (tc/2)*TBPRD;
    CMPA3 = TBPRD;
    CMPA4 = 0;
    CMPA5 = 0;
    CMPA6 = (tc/2+tb)*TBPRD;
elseif (N == 2) && (Reg==3) && (SubReg==1)
    ta = (1-2*1*m*sin(alpha));
    tb = (2*1*m*sin((pi/3)+alpha)-1);
    tc = (1-2*1*m*sin((pi/3)-alpha));
    CMPA1 = TBPRD;
    CMPA2 = (ta/2+tc)*TBPRD;
    CMPA3 = TBPRD;
    CMPA4 = (ta/2)*TBPRD;
    CMPA5 = 0;
    CMPA6 = (ta/2+tc+tb)*TBPRD;
elseif (N == 2) && (Reg==4)
    ta = (2*1*m*sin(alpha)-1);
    tb = 2*1*m*sin((pi/3)-alpha);
    tc = 2*(1-1*m*sin((pi/3)+alpha));
    CMPA1 = TBPRD;
    CMPA2 = (ta/2)*TBPRD;
    CMPA3 = TBPRD;
    CMPA4 = (ta/2+tc)*TBPRD;
    CMPA5 = 0;
    CMPA6 = (ta/2+tc+tb)*TBPRD;

```



```
%% SECTOR 3
elseif (N == 3) && (Reg==1) && (SubReg==0)
    ta = 2*1*m*sin((pi/3)-alpha);
    tb = (1-2*1*m*sin((pi/3)+alpha));
    tc = 2*1*m*sin(alpha);
    CMPA1 = TBPRD;
    CMPA2 = (ta/2+tc+tb)*TBPRD;
    CMPA3 = TBPRD;
    CMPA4 = (ta/2+tc)*TBPRD;
    CMPA5 = 0;
    CMPA6 = (ta/2)*TBPRD;
elseif (N == 3) && (Reg==1) && (SubReg==1)
    ta = 2*1*m*sin((pi/3)-alpha);
    tb = (1-2*1*m*sin((pi/3)+alpha));
    tc = 2*1*m*sin(alpha);
    CMPA1 = TBPRD;
    CMPA2 = (tc/2+tb)*TBPRD;
    CMPA3 = (tc/2+tb+ta)*TBPRD;
    CMPA4 = (tc/2)*TBPRD;
    CMPA5 = 0;
    CMPA6 = 0;
elseif (N == 3) && (Reg==2)
    ta = 2*(1-1*m*sin((pi/3)+alpha));
    tb = 2*1*m*sin(alpha);
    tc = (2*1*m*sin((pi/3)-alpha)-1);
    CMPA1 = TBPRD;
    CMPA2 = (ta/2)*TBPRD;
    CMPA3 = TBPRD;
    CMPA4 = (ta/2+tc+tb)*TBPRD;
    CMPA5 = 0;
    CMPA6 = (ta/2+tc)*TBPRD;
elseif (N == 3) && (Reg==3) && (SubReg==0)
    ta = (1-2*1*m*sin(alpha));
    tb = (2*1*m*sin((pi/3)+alpha)-1);
    tc = (1-2*1*m*sin((pi/3)-alpha));
    CMPA1 = TBPRD;
    CMPA2 = (ta/2+tc)*TBPRD;
    CMPA3 = TBPRD;
    CMPA4 = (ta/2+tc+tb)*TBPRD;
    CMPA5 = 0;
    CMPA6 = (ta/2)*TBPRD;
elseif (N == 3) && (Reg==3) && (SubReg==1)
    ta = (1-2*1*m*sin(alpha));
    tb = (2*1*m*sin((pi/3)+alpha)-1);
    tc = (1-2*1*m*sin((pi/3)-alpha));
    CMPA1 = TBPRD;
    CMPA2 = (tc/2)*TBPRD;
    CMPA3 = (tc/2+tb+ta)*TBPRD;
    CMPA4 = (tc/2+tb)*TBPRD;
    CMPA5 = 0;
    CMPA6 = 0;
elseif (N == 3) && (Reg==4)
    ta = (2*1*m*sin(alpha)-1);
    tb = 2*1*m*sin((pi/3)-alpha);
    tc = 2*(1-1*m*sin((pi/3)+alpha));
    CMPA1 = TBPRD;
    CMPA2 = (tc/2)*TBPRD;
    CMPA3 = (tc/2+tb)*TBPRD;
    CMPA4 = (tc/2+tb+ta)*TBPRD;
```



```

CMPA5 = 0;
CMPA6 = 0;

%% SECTOR 4
elseif (N == 4) && (Reg==1) && (SubReg==0)
    ta = 2*1*m*sin((pi/3)-alpha);
    tb = (1-2*1*m*sin((pi/3)+alpha));
    tc = 2*1*m*sin(alpha);
    CMPA1 = TBPRD;
    CMPA2 = (tc/2+tb+ta)*TBPRD;
    CMPA3 = (tc/2+tb)*TBPRD;
    CMPA4 = (tc/2)*TBPRD;
    CMPA5 = 0;
    CMPA6 = 0;
elseif (N == 4) && (Reg==1) && (SubReg==1)
    ta = 2*1*m*sin((pi/3)-alpha);
    tb = (1-2*1*m*sin((pi/3)+alpha));
    tc = 2*1*m*sin(alpha);
    CMPA1 = TBPRD;
    CMPA2 = TBPRD;
    CMPA3 = (ta/2+tc+tb)*TBPRD;
    CMPA4 = (ta/2+tc)*TBPRD;
    CMPA5 = (ta/2)*TBPRD;
    CMPA6 = 0;
elseif (N == 4) && (Reg==2)
    ta = 2*(1-1*m*sin((pi/3)+alpha));
    tb = 2*1*m*sin(alpha);
    tc = (2*1*m*sin((pi/3)-alpha)-1);
    CMPA1 = TBPRD;
    CMPA2 = (tc/2+tb)*TBPRD;
    CMPA3 = (tc/2)*TBPRD;
    CMPA4 = (tc/2+tb+ta)*TBPRD;
    CMPA5 = 0;
    CMPA6 = 0;
elseif (N == 4) && (Reg==3) && (SubReg==0)
    ta = (1-2*1*m*sin(alpha));
    tb = (2*1*m*sin((pi/3)+alpha)-1);
    tc = (1-2*1*m*sin((pi/3)-alpha));
    CMPA1 = TBPRD;
    CMPA2 = (tc/2+tb+ta)*TBPRD;
    CMPA3 = (tc/2)*TBPRD;
    CMPA4 = (tc/2+tb)*TBPRD;
    CMPA5 = 0;
    CMPA6 = 0;
elseif (N == 4) && (Reg==3) && (SubReg==1)
    ta = (1-2*1*m*sin(alpha));
    tb = (2*1*m*sin((pi/3)+alpha)-1);
    tc = (1-2*1*m*sin((pi/3)-alpha));
    CMPA1 = TBPRD;
    CMPA2 = TBPRD;
    CMPA3 = (ta/2+tc)*TBPRD;
    CMPA4 = (ta/2+tc+tb)*TBPRD;
    CMPA5 = (ta/2)*TBPRD;
    CMPA6 = 0;
elseif (N == 4) && (Reg==4)
    ta = (2*1*m*sin(alpha)-1);
    tb = 2*1*m*sin((pi/3)-alpha);
    tc = 2*(1-1*m*sin((pi/3)+alpha));
    CMPA1 = TBPRD;

```



```
    CMPA2 = TBPRD;
    CMPA3 = (ta/2)*TBPRD;
    CMPA4 = (ta/2+tc+tb)*TBPRD;
    CMPA5 = (ta/2+tc)*TBPRD;
    CMPA6 = 0;

%% SECTOR 5
elseif (N == 5) && (Reg==1) && (SubReg==0)
    ta = 2*1*m*sin((pi/3)-alpha);
    tb = (1-2*1*m*sin((pi/3)+alpha));
    tc = 2*1*m*sin(alpha);
    CMPA1 = TBPRD;
    CMPA2 = TBPRD;
    CMPA3 = (ta/2+tc+tb)*TBPRD;
    CMPA4 = (ta/2)*TBPRD;
    CMPA5 = (ta/2+tc)*TBPRD;
    CMPA6 = 0;
elseif (N == 5) && (Reg==1) && (SubReg==1)
    ta = 2*1*m*sin((pi/3)-alpha);
    tb = (1-2*1*m*sin((pi/3)+alpha));
    tc = 2*1*m*sin(alpha);
    CMPA1 = (tc/2+tb+ta)*TBPRD;
    CMPA2 = TBPRD;
    CMPA3 = (tc/2+tb)*TBPRD;
    CMPA4 = 0;
    CMPA5 = (tc/2)*TBPRD;
    CMPA6 = 0;
elseif (N == 5) && (Reg==2)
    ta = 2*(1-1*m*sin((pi/3)+alpha));
    tb = 2*1*m*sin(alpha);
    tc = (2*1*m*sin((pi/3)-alpha)-1);
    CMPA1 = TBPRD;
    CMPA2 = TBPRD;
    CMPA3 = (ta/2)*TBPRD;
    CMPA4 = (ta/2+tc)*TBPRD;
    CMPA5 = (ta/2+tc+tb)*TBPRD;
    CMPA6 = 0;
elseif (N == 5) && (Reg==3) && (SubReg==0)
    ta = (1-2*1*m*sin(alpha));
    tb = (2*1*m*sin((pi/3)+alpha)-1);
    tc = (1-2*1*m*sin((pi/3)-alpha));
    CMPA1 = TBPRD;
    CMPA2 = TBPRD;
    CMPA3 = (ta/2+tc)*TBPRD;
    CMPA4 = (ta/2)*TBPRD;
    CMPA5 = (ta/2+tc+tb)*TBPRD;
    CMPA6 = 0;
elseif (N == 5) && (Reg==3) && (SubReg==1)
    ta = (1-2*1*m*sin(alpha));
    tb = (2*1*m*sin((pi/3)+alpha)-1);
    tc = (1-2*1*m*sin((pi/3)-alpha));
    CMPA1 = (tc/2+tb+ta)*TBPRD;
    CMPA2 = TBPRD;
    CMPA3 = (tc/2)*TBPRD;
    CMPA4 = 0;
    CMPA5 = (tc/2+tb)*TBPRD;
    CMPA6 = 0;
elseif (N == 5) && (Reg==4)
    ta = (2*1*m*sin(alpha)-1);
```



```

tb = 2*1*m*sin((pi/3)-alpha);
tc = 2*(1-1*m*sin((pi/3)+alpha));
CMPA1 = (tc/2+tb)*TBPRD;
CMPA2 = TBPRD;
CMPA3 = (tc/2)*TBPRD;
CMPA4 = 0;
CMPA5 = (tc/2+tb+ta)*TBPRD;
CMPA6 = 0;

%% SECTOR 6
elseif (N == 6) && (Reg==1) && (SubReg==0)
    ta = 2*1*m*sin((pi/3)-alpha);
    tb = (1-2*1*m*sin((pi/3)+alpha));
    tc = 2*1*m*sin(alpha);
    CMPA1 = (tc/2+tb)*TBPRD;
    CMPA2 = TBPRD;
    CMPA3 = (tc/2+tb+ta)*TBPRD;
    CMPA4 = 0;
    CMPA5 = (tc/2)*TBPRD;
    CMPA6 = 0;
elseif (N == 6) && (Reg==1) && (SubReg==1)
    ta = 2*1*m*sin((pi/3)-alpha);
    tb = (1-2*1*m*sin((pi/3)+alpha));
    tc = 2*1*m*sin(alpha);
    CMPA1 = (ta/2+tc+tb)*TBPRD;
    CMPA2 = TBPRD;
    CMPA3 = TBPRD;
    CMPA4 = 0;
    CMPA5 = (ta/2+tc)*TBPRD;
    CMPA6 = (ta/2)*TBPRD;
elseif (N == 6) && (Reg==2)
    ta = 2*(1-1*m*sin((pi/3)+alpha));
    tb = 2*1*m*sin(alpha);
    tc = (2*1*m*sin((pi/3)-alpha)-1);
    CMPA1 = (tc/2)*TBPRD;
    CMPA2 = TBPRD;
    CMPA3 = (tc/2+tb)*TBPRD;
    CMPA4 = 0;
    CMPA5 = (tc/2+tb+ta)*TBPRD;
    CMPA6 = 0;
elseif (N == 6) && (Reg==3) && (SubReg==0)
    ta = (1-2*1*m*sin(alpha));
    tb = (2*1*m*sin((pi/3)+alpha)-1);
    tc = (1-2*1*m*sin((pi/3)-alpha));
    CMPA1 = (tc/2)*TBPRD;
    CMPA2 = TBPRD;
    CMPA3 = (tc/2+tb+ta)*TBPRD;
    CMPA4 = 0;
    CMPA5 = (tc/2+tb)*TBPRD;
    CMPA6 = 0;
elseif (N == 6) && (Reg==3) && (SubReg==1)
    ta = (1-2*1*m*sin(alpha));
    tb = (2*1*m*sin((pi/3)+alpha)-1);
    tc = (1-2*1*m*sin((pi/3)-alpha));
    CMPA1 = (ta/2+tc)*TBPRD;
    CMPA2 = TBPRD;
    CMPA3 = TBPRD;
    CMPA4 = 0;
    CMPA5 = (ta/2+tc+tb)*TBPRD;

```



```
    CMPA6 = (ta/2)*TBPRD;
elseif (N == 6) && (Reg==4)
    ta = (2*1*m*sin(alpha)-1);
    tb = 2*1*m*sin((pi/3)-alpha);
    tc = 2*(1-1*m*sin((pi/3)+alpha));
    CMPA1 = (ta/2)*TBPRD;
    CMPA2 = TBPRD;
    CMPA3 = TBPRD;
    CMPA4 = 0;
    CMPA5 = (ta/2+tc+tb)*TBPRD;
    CMPA6 = (ta/2+tc)*TBPRD;

%%NINGUNO DE LOS CASOS ANTERIORES
else
    CMPA1 = TBPRD;
    CMPA2 = TBPRD;
    CMPA3 = TBPRD;
    CMPA4 = 0;
    CMPA5 = 0;
    CMPA6 = 0;
end
```



9.3. ANEXO 3: DATASHEET SKYPER 23 PRO R

SKYPER 32PRO R ...



SKYPER™

IGBT Driver Core

SKYPER 32PRO R

Preliminary Data

Features

- Two output channels
- Integrated potential free power supply
- Under voltage protection
- Drive interlock top / bottom
- Dynamic short circuit protection
- Halt logic signal
- Failure management
- Soft turn-off
- External error input (sec. side)
- IEC 60068-1 (climate) 40/085/56, no condensation and no dripping water permitted, non-corrosive, climate class 3K3 acc. EN60721
- Coated with varnish

Typical Applications

- Driver for IGBT modules in bridge circuits in choppers, inverter drives, UPS and welding inverters
- DC bus voltage up to 1200V

- 1) with external high voltage diode
- 2) Please Note: the isolation test is not performed as a series test at SEMIKRON and must be performed by the user
- 3) according to VDE 0110-20

Isolation coordination in compliance with EN50178 PD2

Operating temperature is real ambient temperature around the driver core

Degree of protection: IP00

Absolute Maximum Ratings

Symbol	Conditions	Values	Units
V_S	Supply voltage primary	16	V
V_{IH}	Input signal voltage (High)	$V_S + 0,3$	V
V_{IL}	Input signal voltage (Low)	GND - 0,3	V
$I_{outPEAK}$	Output peak current	15	A
$I_{outAVmax}$	Output average current	50	mA
f_{max}	max. switching frequency	50	kHz
V_{CE}	Collector emitter voltage sense across the IGBT ¹⁾	1700	V
dv/dt	Rate of rise and fall of voltage secondary to primary side	50	kV/ μ s
V_{isolIO}	Isolation test voltage input - output (AC, rms, 2s) ²⁾	4000	V
V_{isolPD}	Partial discharge extinction voltage, rms, $Q_{PD} \leq 10pC$ ³⁾	1500	V
V_{isol12}	Isolation test voltage output 1 - output 2 (AC, rms, 2s) ²⁾	1500	V
R_{Gonmin}	Minimum rating for R_{Gon}	1,5	Ω
$R_{Goffmin}$	Minimum rating for R_{Goff}	1,5	Ω
$Q_{out/pulse}$	Max. rating for output charge per pulse	6,3	μ C
T_{op}	Operating temperature	- 40 ... + 85	$^{\circ}$ C
T_{slg}	Storage temperature	- 40 ... + 85	$^{\circ}$ C

Characteristics

$T_a = 25^{\circ}$ C, unless otherwise specified

Symbol	Conditions	min.	typ.	max.	Units
V_S	Supply voltage primary side	14,4	15	15,6	V
I_{SO}	Supply current primary side (no load)	80			mA
	Supply current primary side (max.)			500	mA
V_I	Input signal voltage on/off		15 / 0		V
V_{IT+}	Input threshold voltage (High)			12,3	V
V_{IT-}	Input threshold voltage (Low)	4,6			V
R_{in}	Input resistance (switching signals, HALT signal)		100		k Ω
$V_{G(on)}$	Turn on gate voltage output		+ 15		V
$V_{G(off)}$	Turn off gate voltage output		- 7		V
f_{ASIC}	Asic system switching frequency		8		MHz
$t_{d(on)IO}$	Input-output turn-on propagation time		1,2		μ s
$t_{d(off)IO}$	Input-output turn-off propagation time		1,2		μ s
$t_{d(terr)}$	Error input-output propagation time	3,1		5,8	μ s
$t_{d(terr)ext}$	External error (secondary side) input-output propagation time		6,1		μ s
$t_{PERRRESET}$	Error reset time		9		μ s
t_{TD}	Top-Bot Interlock Dead Time	no interlock		4,3	μ s
C_{ps}	Coupling capacitance primary secondary		12		pF
w	weight		34		g
MTBF	Mean Time Between Failure @ $T_a = 40^{\circ}$ C, max. load		1,3		10^6 h

This technical information specifies semiconductor devices but promises no characteristics. No warranty or guarantee expressed or implied is made regarding delivery, performance or suitability.

9.4. ANEXO 4: DATASHEET BOARD 1 SKYPER 32 PRO R

Board 1 SKYPER 32PRO R



SKYPER®

Adaptor board

Board 1 SKYPER 32PRO R

Preliminary Data

Features

- Two output channels
- Failure management

Typical Applications*

- Adaptor board for SKYPER 32 IGBT drivers in bridge circuits for industrial applications
- DC bus up to 1200V

Footnotes

All characteristics listed in the data sheet are guilty for the use with SKYPER 32. Please consider the derating of the ambient temperature. Please refer to the datasheet of SKYPER 32 for further information.



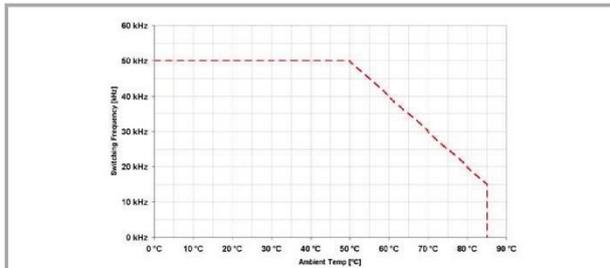
Adaptor board

Absolute Maximum Ratings

Symbol	Conditions	Values	Unit
V_s	Supply voltage primary	16	V
$I_{outPEAK}$	Output peak current	15	A
$I_{outAVmax}$	Output average current	50	mA
f_{max}	max. switching frequency	50	kHz
V_{CE}	Collector emitter voltage sense across the IGBT	1700	V
V_{isolIO}	Isolation test voltage input - output (AC, rms, 2s)	4000	V
V_{isolPD}	Partial discharge extinction voltage, rms, $Q_{PD} \leq 10pC$	1500	V
V_{isol12}	Isolation test voltage output 1 - output 2 (AC, rms, 2s)	1500	V
$R_{Gon\ min}$		1.5	Ω
$R_{Goff\ min}$	Minimum rating for external R_{Goff}	1.5	Ω
T_{op}	Operating temperature	-25 ... 85	$^{\circ}C$
T_{slg}	Storage temperature	-25 ... 85	$^{\circ}C$

Characteristics

Symbol	Conditions	min.	typ.	max.	Unit
V_s	Supply voltage primary side	14.4	15	15.6	V
V_i	input signal voltage on / off		15 / 0		V
V_{IT+}	Input threshold voltage HIGH			12.3	V
V_{IT-}	Input threshold voltage (LOW)	4.6			V
$V_{G(on)}$	Turn on gate voltage output		15		V
$V_{G(off)}$	Turn off gate voltage output		-7		V
$t_{d(on)IO}$	Input-output turn-on propagation time		1.2		μs
$t_{d(off)IO}$	Input-output turn-off propagation time		1.2		μs



Derating

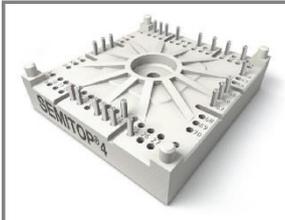
This is an electrostatic discharge sensitive device (ESDS), international standard IEC 60747-1, Chapter IX

* The specifications of our components may not be considered as an assurance of component characteristics. Components have to be tested for the respective application. Adjustments may be necessary. The use of SEMIKRON products in life support appliances and systems is subject to prior specification and written approval by SEMIKRON. We therefore strongly recommend prior consultation of our personal.



9.5. ANEXO 5: DATASHEET SF75MLI066T

SK75MLI066T



SEMISTOP® 4

IGBT Module

SK75MLI066T

Preliminary Data

Features

- Compact design
- One screw mounting
- Heat transfer and isolation through direct copper bonded aluminium oxide ceramic (DCB)
- Trench IGBT technology
- CAL technology FWD
- Integrated NTC temperature sensor

Typical Applications*

- Multi level inverter

Remarks

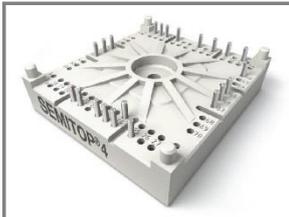
- $V_{isol} = 3000V$ AC, 1s, 50Hz
- Dynamic measure: DUT= IGBT (Gate pin 55) and Neutral Clamp Diode (Kathode pin 56) as free-wheeling diode



MLI-T

Absolute Maximum Ratings		$T_g = 25\text{ }^\circ\text{C}$, unless otherwise specified	
Symbol	Conditions	Values	Units
IGBT			
V_{CES}	$T_j = 25\text{ }^\circ\text{C}$	600	V
I_C	$T_j = 175\text{ }^\circ\text{C}$	$T_g = 25\text{ }^\circ\text{C}$	83 A
		$T_g = 70\text{ }^\circ\text{C}$	67 A
I_{CRM}	$I_{CRM} = 2 \times I_{Cnom}$	150	A
V_{GES}		± 20	V
t_{psc}	$V_{CC} = 360\text{ V}; V_{GE} \leq 20\text{ V}; T_j = 125\text{ }^\circ\text{C}; V_{CES} < 600\text{ V}$	6	μs
Inverse Diode			
I_F	$T_j = 175\text{ }^\circ\text{C}$	$T_g = 25\text{ }^\circ\text{C}$	92 A
		$T_g = 70\text{ }^\circ\text{C}$	73 A
I_{FRM}	$I_{FRM} = 2 \times I_{Fnom}$	150	A
Freewheeling Diode			
I_F	$T_j = 175\text{ }^\circ\text{C}$	$T_g = 25\text{ }^\circ\text{C}$	92 A
		$T_g = 70\text{ }^\circ\text{C}$	73 A
I_{FRM}	$I_{FRM} = 2 \times I_{Fnom}$	150	A
Module			
$I_{H(RMS)}$			A
T_{vj}		-40 ... +175	$^\circ\text{C}$
T_{slg}		-40 ... +125	$^\circ\text{C}$
V_{isol}	AC, 1 min.	2500	V

Characteristics		$T_g = 25\text{ }^\circ\text{C}$, unless otherwise specified			
Symbol	Conditions	min.	typ.	max.	Units
IGBT					
$V_{GE(th)}$	$V_{GE} = V_{CE}, I_C = 1,2\text{ mA}$	5	5,8	6,5	V
I_{CES}	$V_{GE} = 0\text{ V}, V_{CE} = V_{CES}$	$T_j = 25\text{ }^\circ\text{C}$		0,0038	mA
		$T_j = 125\text{ }^\circ\text{C}$			mA
I_{GES}	$V_{CE} = 0\text{ V}, V_{GE} = 20\text{ V}$			600	nA
V_{CE0}		$T_j = 25\text{ }^\circ\text{C}$	0,8	1,1	V
		$T_j = 150\text{ }^\circ\text{C}$	0,7	1	V
r_{CE}	$V_{GE} = 15\text{ V}$	$T_j = 25\text{ }^\circ\text{C}$	8	10	$\text{m}\Omega$
		$T_j = 150\text{ }^\circ\text{C}$	13	14	$\text{m}\Omega$
$V_{CE(sat)}$	$I_{Cnom} = 75\text{ A}, V_{GE} = 15\text{ V}$	$T_j = 25\text{ }^\circ\text{C}_{chiplev.}$	1,45	1,85	V
		$T_j = 150\text{ }^\circ\text{C}_{chiplev.}$	1,65	2,05	V
C_{ies}	$V_{CE} = 25, V_{GE} = 0\text{ V}$		4,7		nF
C_{oes}			0,3		nF
C_{res}			0,145		nF
Q_G	$V_{GE} = -7V...+15V$		650		nC
R_{Gint}	$T_j = 150\text{ }^\circ\text{C}$		4		Ω
$t_{d(on)}$	$R_{Gon} = 4\text{ }\Omega$	$V_{CC} = 300V$		97	ns
			$di/dt = 4100\text{ A}/\mu\text{s}$	34	ns
E_{on}	$R_{Goff} = 4\text{ }\Omega$	$I_C = 75A$		1,7	mJ
$t_{d(off)}$	$R_{Goff} = 4\text{ }\Omega$	$T_j = 150\text{ }^\circ\text{C}$		339	ns
			$di/dt = 4100\text{ A}/\mu\text{s}$	65	ns
E_{off}		$V_{GE} = -7/+15V$		2,8	mJ
$R_{th(j-s)}$	per IGBT		0,75		K/W

SK75MLI066T

SEMITOP® 4
IGBT Module
SK75MLI066T

Preliminary Data

Features

- Compact design
- One screw mounting
- Heat transfer and isolation through direct copper bonded aluminium oxide ceramic (DCB)
- Trench IGBT technology
- CAL technology FWD
- Integrated NTC temperature sensor

Typical Applications*

- Multi level inverter

Remarks

- $V_{isol} = 3000V$ AC, 1s, 50Hz
- Dynamic measure: DUT= IGBT (Gate pin 55) and Neutral Clamp Diode (Kathode pin 56) as free-wheeling diode


MLI-T

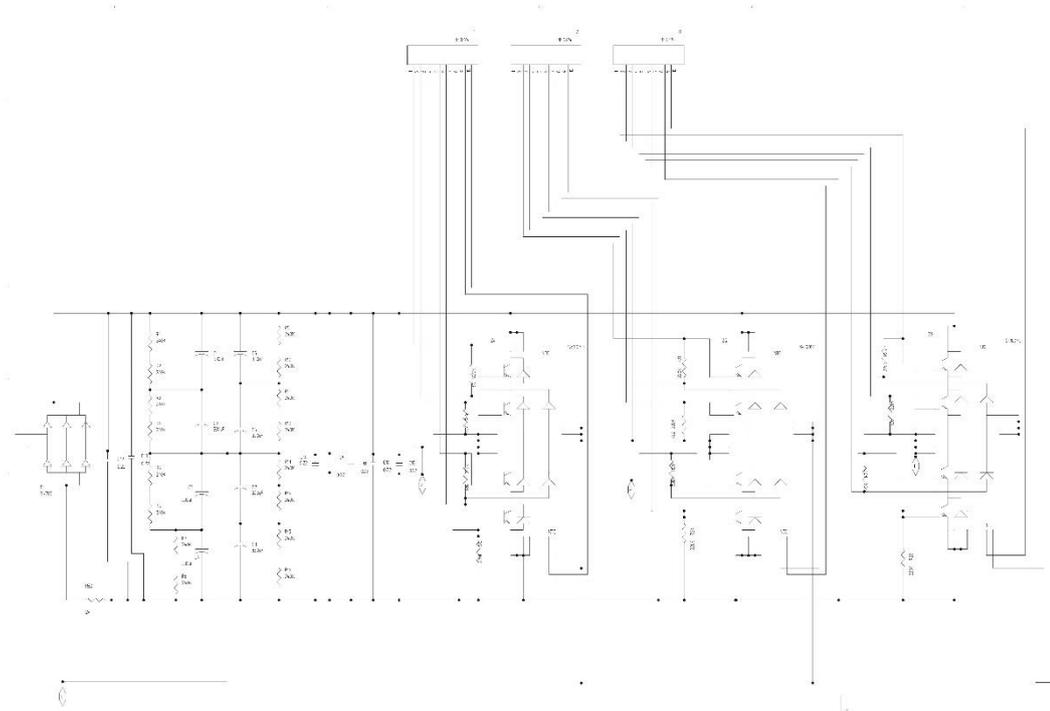
Characteristics					
Symbol	Conditions	min.	typ.	max.	Units
Inverse Diode					
$V_F = V_{EC}$	$I_{Fnom} = 75 A; V_{GE} = 0 V$				
					$T_j = 25 ^\circ C_{chiplev.}$
			1,5	1,7	V
					$T_j = 150 ^\circ C_{chiplev.}$
			1,5	1,7	V
V_{F0}					$T_j = 25 ^\circ C$
			1	1,1	V
					$T_j = 150 ^\circ C$
			0,9	1	V
r_F					$T_j = 25 ^\circ C$
			6,7	8	mΩ
					$T_j = 150 ^\circ C$
			8	9,3	mΩ
I_{RRM}	$I_F = 75 A$				$T_j = 150 ^\circ C$
Q_{rr}	$di/dt = 4100 A/\mu s$				A
E_{rr}	$V_R = 300V$				μC
			4		mJ
$R_{th(j-s)D}$	per diode				1,1
					K/W
Freewheeling Diode (Neutral Clamp Diode)					
$V_F = V_{EC}$	$I_{Fnom} = 75 A; V_{GE} = 0 V$				
					$T_j = 25 ^\circ C_{chiplev.}$
					$T_j = 150 ^\circ C_{chiplev.}$
			1,5	1,7	V
					$T_j = 150 ^\circ C_{chiplev.}$
			1,5	1,7	V
V_{F0}					$T_j = 25 ^\circ C$
			1	1,1	V
					$T_j = 150 ^\circ C$
			0,9	1	V
r_F					$T_j = 25 ^\circ C$
			6,7	8	V
					$T_j = 150 ^\circ C$
			8	9,3	V
I_{RRM}	$I_F = 75 A$				$T_j = 150 ^\circ C$
Q_{rr}	$di/dt = 3100 A/\mu s$				A
E_{rr}	$V_R = 300V$				μC
			4		mJ
$R_{th(j-s)FD}$	per diode				1,1
					K/W
M_s	to heat sink		2,5	2,75	Nm
w				60	g
Temperature sensor					
R_{100}	$T_s = 100 ^\circ C (R_{25} = 5k\Omega)$			493±5%	Ω

This is an electrostatic discharge sensitive device (ESDS), international standard IEC 60747-1, Chapter IX.

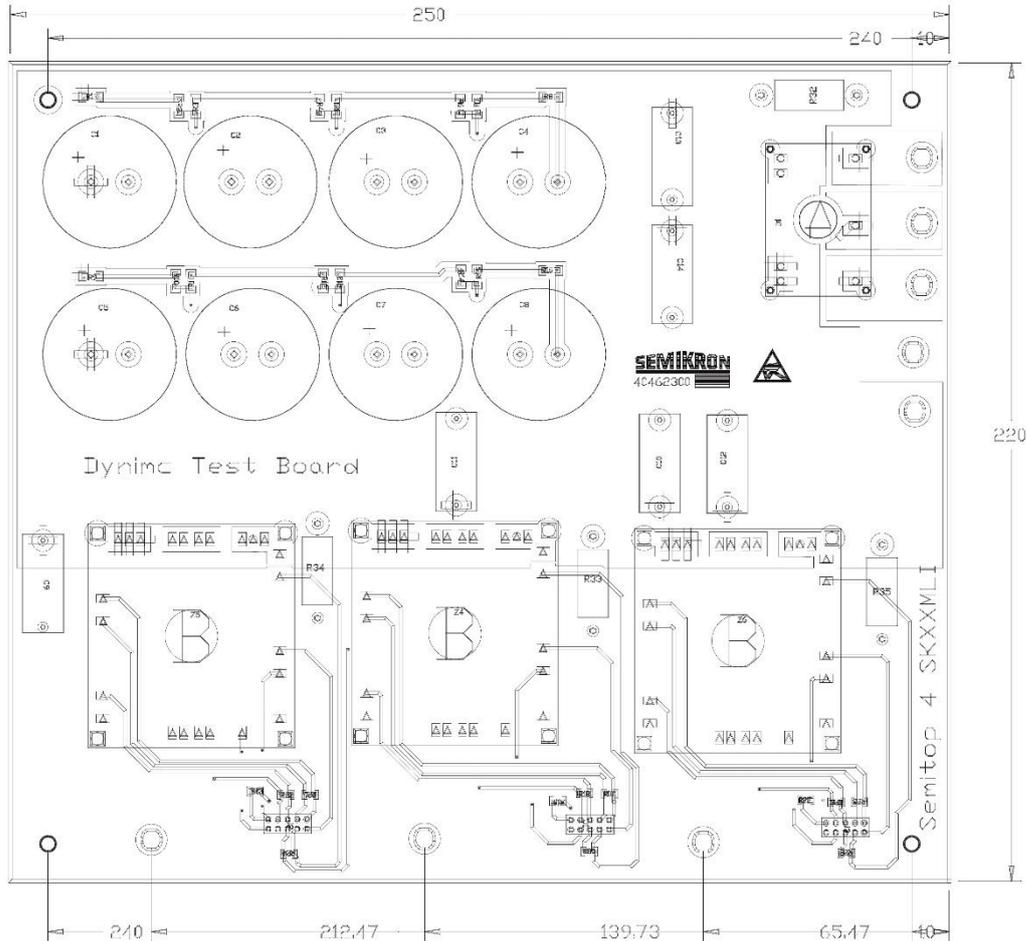
* The specifications of our components may not be considered as an assurance of component characteristics. Components have to be tested for the respective application. Adjustments may be necessary. The use of SEMIKRON products in life support appliances and systems is subject to prior specification and written approval by SEMIKRON. We therefore strongly recommend prior consultation of our personal.



9.6. ANEXO 6: ESQUEMA ELÉCTRICO PLACA POTENCIA



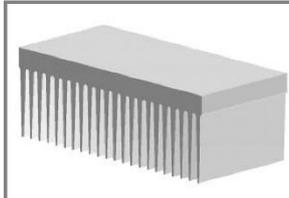
9.7. ANEXO 7: ESQUEMA UBICACIÓN DE COMPONENTES PLACA DE POTENCIA





9.8. ANEXO 8: DATASHEET RADIADOR EMPLEADO

P 16



Heatsink

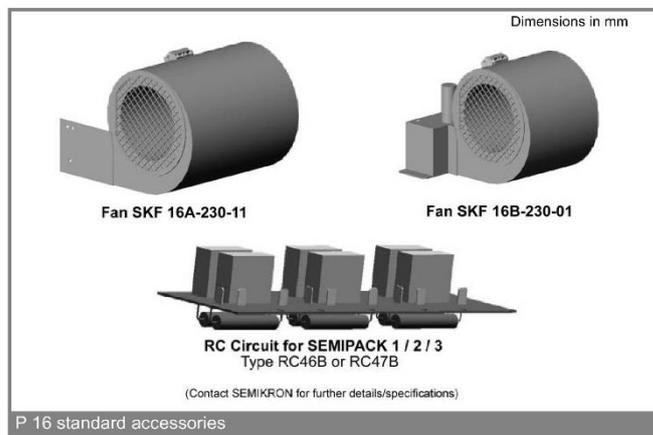
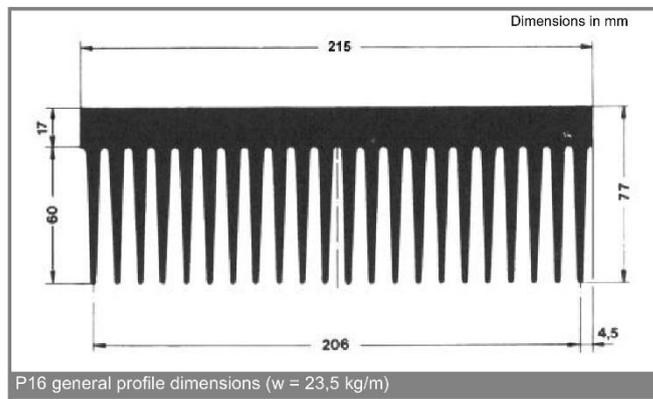
Standard lengths	n	b / d Ø mm	R_{thha} K/W	R_{thha} with fan SKF 16B-230-01 K/W	w kg
P 16/170	3	20		0,05	4
P 16/200	3	20		0,046	4,7
	6	20		0,039	
	3	34		0,038	
	2	50		0,04	
	3	50		0,033	
P 16/300	6	34		0,036	7
	6	50		0,024	

For all isolated power modules

P 16

Features

- Intended for all isolated power modules : SEMIPACK, SEMITRANS, SEMIPONT, SKiM, SEMiX, SKiiP
- Excellent efficiency/volume ratio
- Best suited fans : SKF 16-A and SKF 16-B
- Available in various lengths





9.9. ANEXO 9: HOJA EXCEL PARA CALCULO DE R_{CE} Y

C_{CE}

Topic: Calculation R_{CE} , C_{CE} , V_{CEstat} & t_{bl}

Driver: SKYPER 32 R, SKYPER 32PRO R

$R_{VCE} = 0.00 \text{ k}\Omega$

Calculation for SKYPER

$V_{CEstat} = 5.00 \text{ V} \Rightarrow R_{CE} = 15.08 \text{ k}\Omega$

$t_{bl} = 4.50 \text{ }\mu\text{s} \Rightarrow C_{CE} = 105.493 \text{ pF}$

$R_{CE} = 15.00 \text{ k}\Omega \Rightarrow V_{CEstat} = 4.98 \text{ V}$

$C_{CE} = 100.00 \text{ pF} \Rightarrow t_{bl} = 4.47 \text{ }\mu\text{s}$

Calculation for SKYPER 32PRO R

$V_{CEstat} = 5.10 \text{ V} \Rightarrow R_{CE} = 15.73 \text{ k}\Omega$

$t_{bl} = 4.00 \text{ }\mu\text{s} \Rightarrow C_{CE} = 52.60 \text{ pF}$

$R_{CE} = 15.00 \text{ k}\Omega \Rightarrow V_{CEstat} = 4.96 \text{ V}$

$C_{CE} = 68.00 \text{ pF} \Rightarrow t_{bl} = 3.97 \text{ }\mu\text{s}$

Hello Daniel,

Please find attached an excel file that can help you to calculate the Rce and Cce depending on Vce_stat and tbl.

Vce_stat should be higher than Vce_sat, we would suggest around 5V.
tbl should be smaller than tpsc, we would suggest 3-4us.

You can find more information about this topic on the pages 348 and following of the SEMIKRON Application Book:
<http://shop.semikron.com/en/Products-and-Shop/Product-Groups/Other/APPLICATION-MANUAL-2011.html>
You can download it for free by clicking "PDF Version as download".

Best Regards,
Hugo Paz

SEMIKRON has made every effort to make this document / tool complete and accurate to assist you in using SEMIKRON device. Information furnished in this document / tool is believed to be accurate and reliable. However, no representation or warranty is given and no liability is assumed with respect to the accuracy or use of such information. Since the devices are subject to change and improvements all data are subject to change or correction without prior notice. The data presented here is for general information only, providing an overview of the SEMIKRON products capabilities. SEMIKRON neither knows your application nor can check your input data. User environmental conditions may be different from the assumed one. SEMIKRON will not accept any responsibility or assume any liability for the correctness or validity of the results presented by this tool. The user himself is responsible for the consequences when using the results presented by this tool. Specifications mentioned in this document / tool are subject to change without notice. This document / tool supersedes and replaces all documents / tools previously supplied and may be supersede by updates.