



Universidad de Valladolid



**ESCUELA DE INGENIERÍAS
INDUSTRIALES**

UNIVERSIDAD DE VALLADOLID

ESCUELA DE INGENIERIAS INDUSTRIALES

Grado en Ingeniería Eléctrica

**“Simulación en tiempo real de un enlace
HVDC con convertidores MMC”**

Autor:

Gregori Cano, Aitor

Tutor(es):

**De Pablo Gómez, Santiago
Departamento de Tecnología
Electrónica**

Valladolid, febrero 2022.

Resumen

El trabajo consiste en la realización de una simulación en tiempo real de una línea HVDC (High Voltage Direct Current, Alta Tensión en Corriente Continua), con convertidores AC/DC y DC/AC del tipo MMC (Modular Multilevel Converter, Convertidor Modular Multinivel), con PWM.

La instalación está conformada por una red trifásica en corriente alterna a 50 Hz, conectado a un primer convertidor que transforma esa corriente alterna en corriente continua, para ser transportada por la línea hasta otro convertidor del mismo tipo pero que, ahora funciona en sentido inverso, transformando la corriente continua en alterna. Finalmente, este último convertidor alimenta a otra red trifásica en corriente alterna a 50 Hz.

El objetivo del trabajo es evaluar la simulación, contrastarla y concluir si es o no un modelo de simulación fiable.

Palabras clave

HVDC, MMC, PWM, Simulador en tiempo real, Sistemas eléctricos.

Abstract

The project is about making a real-time simulation of a HVDC line (High Voltage Direct Current), with AC/DC and DC/AC MMC converters (Modular Multilevel Converter) and using PWM.

The installation is about a three-phase network in alternating current at 50 Hz, connected to a first converter that will transform that alternating current into direct current, to be transported by the line to another converter of the same type, but in this case, it will work in the opposite direction, transforming direct current into alternating current. Finally, this last converter will supply another three-phase network in alternating current at 50 Hz.

The aim of the project will be to evaluate the simulation, contrast it and conclude whether or not it is a reliable simulation model.

Keywords

HVDC, MMC, PWM, Real time simulator, Electric Systems.

ÍNDICE

ÍNDICE DE FIGURAS	9
ÍNDICE DE ECUACIONES	15
1.- Capítulo 1: Introducción y objetivos.....	17
1.1.- Presentación de objetivos	18
1.1.1.- Objetivo principal	18
1.1.2.- Objetivos parciales	18
1.2.- Resumen del documento	18
2.- Capítulo 2: Estado del arte	21
2.1.- La Guerra de las Corrientes	21
2.2.- Introducción histórica del HVDC	21
2.3.- HVDC en la actualidad.....	22
2.4.- HVDC en España	22
2.5.- Clasificación de los HVDC	22
2.5.1.- HVDC según el tipo de conexión.....	23
2.5.1.1.- Monopolar	23
2.5.1.2.- Bipolar	23
2.5.1.3.- Homopolar	24
2.5.2.- HVDC según su configuración	25
2.5.2.1.- Punto a punto.....	25
2.5.2.2.- Back to back.....	25
2.5.2.3.- Multiterminal	26
2.5.2.4.- Unitaria	26
2.6.- HVDC y redes AC	27
2.6.1.- Control de frecuencia, tensión, factor de potencia, potencia activa y reactiva	27
2.6.2.- Interconexión entre sistemas eléctricos.....	31
2.6.3.- Transporte de energía mediante cable aislado, submarino y subterráneo.....	31
2.6.4.- Otras aplicaciones de los HVDC	32
2.7.- HVDC vs AC en cuanto a costes.....	33
2.8.- Convertidores de los HVDC	34
2.8.1.- Convertidor LCC.....	34
2.8.2.- Convertidor VSC.....	37

2.8.2.1.-	Convertidor MMC	40
2.8.2.1.1.-	Tipos de celda: puente y semi puente	41
2.9.-	Modulación por Ancho de Pulso o PWM	43
2.9.1.-	Phase Disposition	44
2.9.2.-	Phase Opposition Disposition.....	44
2.9.3.-	Alternative Phase Opposition and Disposition	45
2.9.4.-	Phase Shifted.....	45
2.9.5.-	Carrier Overlapping.....	46
2.9.6.-	Variable Frequency 2.....	46
2.10.-	Apartado sobre Matlab y Simulink:.....	47
2.11.-	Apartado sobre simulación en tiempo real:	48
2.11.1.-	Fundamentos.....	48
2.11.2.-	Beneficios	49
2.11.3.-	Presentación del entorno de simulación	51
2.11.3.1.-	Interfaz de usuario.....	53
2.11.3.2.-	Presentación de bloques más relevantes.....	54
2.11.3.2.1.-	Explicación bloque del MMC.....	54
2.11.3.2.2.-	Explicación bloque RCPcore.....	55
2.11.3.3.-	Funciones embebidas ya incorporadas en el software.....	59
2.11.3.3.1.-	readVector (), writeVector ()	59
2.11.3.3.2.-	uvw2dq () y dq2uvw ()	59
2.11.3.3.3.-	pwmConfig ()	62
2.11.4.-	Limitaciones.....	62
3.-	Capítulo 3: Desarrollo	65
3.1.-	Presentación de la línea que se pretende simular.....	65
3.1.1.-	Valores de interés de los convertidores.....	66
3.2.-	Convertidor en Matlab/Simulink	68
3.2.1.-	Esquema del convertidor en Simulink	68
3.2.2.-	Controlador	75
3.2.2.1.-	Setup.....	76
3.2.2.2.-	loop	78
3.2.2.2.1.-	Generación de señales de sincronización.....	78
3.2.2.2.2.-	Consolidación del estado de los polos	80
3.2.2.2.3.-	Cálculo del número de celdas activas en cada rama....	81

3.2.2.2.4.-	Generación de señales moduladoras.....	82
3.2.2.2.5.-	Cálculo del residuo.....	82
3.2.2.2.6.-	Selección de las celdas que deben ser insertadas o retiradas.....	83
3.2.2.2.7.-	Generación de las salidas	92
3.2.3.-	Generador de PWM en Simulink	92
3.2.4.-	Resultados del convertidor en Simulink	96
3.3.-	Simulación en tiempo real del HVDC.....	100
3.3.1.-	Planta	100
3.3.2.-	Controlador	101
3.3.2.1.-	Global.....	101
3.3.2.1.1.-	Funciones posVmin posVmax	107
3.3.2.2.-	Setup.....	111
3.3.2.3.-	Loop	115
3.3.2.3.1.-	PWMConfig	115
3.3.2.3.2.-	Variables locales	116
3.3.2.3.3.-	Lectura de amperímetros y voltímetros.....	116
3.3.2.3.4.-	Señal de sincronización.....	118
3.3.2.3.5.-	Sincronización con la red	119
3.3.2.3.6.-	Generación de las moduladoras.....	126
3.3.2.3.7.-	Cálculo del residuo.....	127
3.3.2.3.8.-	Cálculo estado de los polos de potencia.....	127
3.3.2.3.9.-	Vectores de recuperación de valores	128
3.3.2.3.10.-	Elegir celda que cambia	129
3.3.3.-	Explicación funcionamiento Master y Slave	131
3.3.3.1.-	Funcionamiento modo Master.....	132
3.3.3.2.-	Funcionamiento en modo Slave	133
3.3.4.-	Convertidor realista	134
3.3.4.1.-	Convertidor realista con carga aislada.....	135
3.3.4.2.-	Convertidor realista conectado a red	136
3.3.5.-	Convertidor promediado	138
3.3.5.1.-	Explicación del convertidor promediado	140
3.3.5.2.-	Convertidor promediado con carga aislada.....	143
3.3.5.3.-	Convertidor promediado conectado a red	145

3.3.6.-	Convertidor realista vs Convertidor promediado.....	147
3.3.7.-	Interconexión de los dos convertidores	147
3.3.8.-	Selección de las constantes de los reguladores PI.....	149
3.3.9.-	Evaluación de recursos disponibles tras la simulación	149
3.3.10.-	Resultados de la Simulación en tiempo real.....	151
3.3.10.1.-	Resultados en cuanto a corrientes en alterna	151
3.3.10.2.-	Resultados en cuanto a tensiones en alterna	153
3.3.10.3.-	Comparación entre tensiones y corrientes de ambos convertidores.....	155
3.3.10.4.-	Tensión en la línea HVDC	156
3.3.10.5.-	Cálculo de corrientes circulantes	157
3.3.10.5.1.-	Corrientes circulantes del convertidor realista.....	158
3.3.10.5.2.-	Corrientes circulantes del convertidor promediado.	160
4.-	Capítulo 4: Repercusiones del TFG.....	163
5.-	Capítulo 5: Conclusiones	165
5.1.-	Implementación de objetivos propuestos.....	165
5.1.1.-	Implementación de los objetivos parciales	165
5.1.2.-	Implementación del objetivo principal	166
5.2.-	Líneas de futuros trabajos	166
6.-	Capítulo 6: Bibliografía	169

ÍNDICE DE FIGURAS

Figura 1 – Esquema de un HVDC con conexión monopolar (9).....	23
Figura 2 - Esquema de un HVDC con conexión bipolar (9)	24
Figura 3 - Esquema de un HVDC con conexión homopolar (9).....	25
Figura 4 - Esquema de un HVDC con configuración punto a punto (elaboración propia).....	25
Figura 5 – Esquema de un HVDC con configuración Back to Back (elaboración propia).....	26
Figura 6 - Esquema de un HVDC con configuración multiterminal en serie (elaboración propia).....	26
Figura 7 - Esquema de un HVDC con configuración multiterminal en paralelo (elaboración propia).....	26
Figura 8 - Límite de estabilidad estática (10)	28
Figura 9 - Curvas P-V para diferentes f.d.p. Y zonas de de la curva P-V (10) ..	28
Figura 10 - Curvas Q-V para distintos f.d.p. y zonas de la curva (10).....	29
Figura 11 - Costes HVDC vs costes HVAC (12).....	33
Figura 12 - Imagen de tiristor real y símbolo del tiristor (13)	35
Figura 13 - Esquema de un rectificador de doce pulsos.....	36
Figura 14 - Corriente positiva (izq.) y negativa (dcha.) por un rectificador tipo puente (12).....	36
Figura 15 - Imagen de un IGBT simple, un módulo IGBT y el símbolo que representa al IGBT (16) (17)	38
Figura 16 - Esquema de un convertidor de dos niveles (2)	39
Figura 17 - Esquema de un convertidor multinivel restringido por diodo (2) ..	39
Figura 18 - Esquema de un convertidor multinivel con condensador flotante (2)	40
Figura 19 - Esquema de un convertidor modular multinivel (18).....	40
Figura 20 - Celdas encendidas en un convertidor MMC de cuatro celdas por rama y 800 V por celda (elaboración propia).....	41
Figura 21 - Celda tipo semi puente y celda tipo puente completo (18).....	42
Figura 22 - Circulación de corriente por una celda insertada (elaboración propia).....	43
Figura 23 - Circulación de corriente por una celda en bypass (elaboración propia).....	43
Figura 24 - Phase Disposition PWM.....	44
Figura 25 - Phase Opposition Disposition PWM.....	45
Figura 26 - Alternante Phase Opposition Disposition PWM	45
Figura 27 - Phase Shifted PWM	46
Figura 28 - Carrier Overlapping PWM	46
Figura 29 - Variable Frequency 2	47
Figura 30 – Esquema sobre la composición de un HIL (24)	50
Figura 31 - Esquema sobre la composición de un RCP (24)	51
Figura 32 - Interfaz de usuario del software del simulador en tiempo real.....	54

Figura 33 - Bloque de convertidor MMC del software del simulador en tiempo real	54
Figura 34 - Menú del bloque del convertidor MMC	55
Figura 35 - Bloque RCPcore	56
Figura 36 - Pestaña "Config" del menú del bloque RCPcore.....	56
Figura 37 - Pestaña "Global" del menú del bloque RCPcore.....	57
Figura 38 - Pestaña "setup()" del menú del bloque RCPcore	57
Figura 39 - Pestaña "loop()" del menú del bloque RCPcore.....	58
Figura 40 - Pestaña "Help" del menú del bloque RCPcore.....	58
Figura 41 - Línea objetivo de la simulación (elaboración propia).....	65
Figura 42 - Esquema del convertidor de la simulación (elaboración propia) ..	67
Figura 43 - Planta del convertidor MMC en Simulink.....	70
Figura 44 - Bloque subsistema del controlador en Simulink.....	71
Figura 45 - Bloque función embebida "controller"	72
Figura 46 - Interior de una celda del convertidor de Simulink.....	73
Figura 47 - Bloque subsistema que contiene las celdas de una rama del convertidor MMC en Simulink	73
Figura 48 - Bloque función embebida "pwm_gen"	74
Figura 49 - Scopes de corrientes circulantes y tensiones de las ramas del convertidor MMC en Simulink	74
Figura 50 - Bloque subsistema "Medida Corrientes"	75
Figura 51 - Código de la función "controller" del convertidor de Simulink. Setup	77
Figura 52 - Actualización de tiempo en el convertidor de Matlab.....	78
Figura 53 - Generación de señales de sincronización del convertidor de Simulink.....	79
Figura 54 - Asignación de señal de sincronización.....	80
Figura 55 - Consolidación del estado de los polos del convertidor de Simulink (Parte 1).....	81
Figura 56 - Consolidación del estado de los polos del convertidor de Simulink (Parte 2).....	81
Figura 57 - Cálculo del número de celdas activas en el convertidor de Simulink	82
Figura 58 - Código de Matlab. Generación de señales moduladoras	82
Figura 59 - Cálculo del residuo del convertidor de Simulink	83
Figura 60 - Conmutación de las celdas con criterio de conmutación ineficiente	84
Figura 61 - Tensiones de los condensadores de las celdas usando el criterio de conmutación ineficiente	84
Figura 62 - Corriente de salida del convertidor con el criterio de conmutación ineficiente	85
Figura 63 - Tensión de salida del convertidor con el criterio de conmutación ineficiente.....	85
Figura 64 - Conmutación de las celdas con el criterio de conmutación eficiente	86

Figura 65 - Tensiones de los condensadores de las celdas usando el criterio de conmutación eficiente	87
Figura 66 - Corriente de salida del convertidor con el criterio de conmutación eficiente	87
Figura 67 - Tensión de salida del convertidor con el criterio de conmutación eficiente	87
Figura 68 - Código de la función pos = maxV (insertadas, V, S)	89
Figura 69 - Código de la función pos = minV (insertadas, V, S)	90
Figura 70 - Selección de celdas en el convertidor de Simulink (Parte 1)	92
Figura 71 - Selección de celdas en el convertidor de Simulink (Parte 2)	92
Figura 72 - Definición e inicialización de variables de la función "pwm_gen".	93
Figura 73 - Loop de la función "pwm_gen"	93
Figura 74 - Flanco ascendente asociado a un ON seguido de un flanco descendente asociado a un OFF (elaboración propia)	94
Figura 75 - Flanco descendente asociado a un OFF seguido de un flanco ascendente asociado a un ON (elaboración propia)	95
Figura 76 - Comparación entre moduladora y portadora	96
Figura 77 - Tensiones de salida del convertidor MMC en Simulink	96
Figura 78 - Corrientes de salida del convertidor de Simulink	97
Figura 79 - Corriente de la rama superior (amarillo), de la rama inferior (azul) y la suma de ambas (rojo) del convertidor de Simulink	97
Figura 80 - Corrientes circulantes del convertidor MMC en Simulink	98
Figura 81 - Corrientes circulantes del convertidor MMC en Simulink (zoom) .	98
Figura 82 - Corrientes circulantes del convertidor en Simulink (zoom x 2)	99
Figura 83 - Planta de la instalación HVDC en la simulación en tiempo real.	100
Figura 84 - Código del Global (1/9)	102
Figura 85 - Código del Global (2/9)	103
Figura 86 - Código del Global (3/9)	104
Figura 87 - Código del Global (4/9)	104
Figura 88 - Código del Global (5/9)	105
Figura 89 - Código del Global (6/9)	105
Figura 90 - Código del Global (7/9)	106
Figura 91 - Código del Global (8/9)	107
Figura 92 - Código del Global (9/9)	107
Figura 93 - Función posVmin	110
Figura 94 - Función posVmax	111
Figura 95 - Código del setup (1/5)	111
Figura 96 - Código del setup (2/5)	112
Figura 97 - Código del setup (3/5)	113
Figura 98 - Código del setup (4/5)	114
Figura 99 - Código del setup (5/5)	115
Figura 100 - Código del loop. Configuración de PWM	116
Figura 101 - Código del loop. Definición de variables locales	116
Figura 102 - Código del loop. Lectura de valores convertidor promediado ..	117
Figura 103 - Código del loop. Lectura de valores convertidor realista	118

Figura 104 - Código del loop. Lectura de tensiones de los condensadores.	118
Figura 105 - Código del loop. Señal de sincronización	118
Figura 106 - Diagrama de bloques del dq PLL (elaboración propia).....	119
Figura 107 - Código del loop. Aplicar dqPLL a las tensiones de la red	120
Figura 108 - Código del loop. Obtención del ángulo theta.....	121
Figura 109 - Desfase entre la tensión de la red (en amarillo) y la tensión a la salida del convertidor tras aplicar el dqPLL (en azul).....	121
Figura 110 - Código del loop. Cálculo de las corrientes de salida del convertidor	122
Figura 111 - Código del loop. Paso de corrientes lo_a, lo_b e lo_c a lo_d e lo_q	122
Figura 112 - Derivada temporal de los vectores d y q (elaboración propia)	124
Figura 113 - Diagrama de bloques sincronización con la red (elaboración propia).....	126
Figura 114 - Código del loop. Cálculo de referencias mediante la regulación de las corrientes.....	126
Figura 115 - Código del loop. Generación de señales moduladoras	127
Figura 116 - Código de loop. Cálculo del residuo	127
Figura 117 - Código del loop. Cálculo del estado actual.....	127
Figura 118 - Código del loop. Recuperación de valores de integración.....	128
Figura 119 - Código del loop. Volcado de valores al vector	128
Figura 120 - Código del loop. Recuperación de valores en el mmc02	129
Figura 121 - Código del loop. Elección de la celda que cambia. Parte 1.....	131
Figura 122 - Código del loop. Elección de la celda que cambia. Parte 2.....	131
Figura 123 - Código funcionamiento en modo Master.....	132
Figura 124 - Diagrama de bloques funcionamiento en modo Slave (elaboración propia).....	133
Figura 125 - Código funcionamiento en modo Slave	133
Figura 126 - Corriente de referencia lo_d_ref de cada convertidor. En rojo la del promediado en modo Slave, en azul la del realista en modo Master.....	134
Figura 127 - Planta del convertidor realista conectado a carga resistiva en el simulador en tiempo real.....	135
Figura 128 - Tensiones de salida del convertidor realista conectado a carga resistiva.....	135
Figura 129 - Corrientes de salida del convertidor realista conectado a carga resistiva.....	136
Figura 130 - Corrientes de salida de la rama superior (en amarillo), la rama inferior (en azul) y la suma de ambas (en rojo) del convertidor realista conectado a carga resistiva	136
Figura 131 - Planta del convertidor realista conectado a red en el simulador en tiempo real.....	137
Figura 132 - Tensiones de salida del convertidor realista conectado a Red	137
Figura 133 - Corrientes de salida del convertidor realista conectado a Red	138
Figura 134 - Corrientes de salida de la rama superior (amarillo), la rama inferior (azul) y la suma de ambas (rojo) del convertidor realista conectado a Red.	138

Figura 135 - Bloque fuente de tensión trifásica programable.....	139
Figura 136 - Elección del tipo de convertidor dentro del bloque del convertidor MMC.....	139
Figura 137 - Cálculo tensiones promedio en cada rama	142
Figura 138 - Paso de tensiones promedio a variables escalares por celda.	142
Figura 139 - Cálculo de la tensión promedio en cada rama.....	142
Figura 140 - Envío de tensiones promedio a fuentes de tensión promediadas	143
Figura 141 - Planta del convertidor promediado conectado a carga resistiva en el simulador en tiempo real.....	143
Figura 142 - Tensiones de salida del convertidor promediado conectado a carga resistiva.....	144
Figura 143 - Corrientes de salida del convertidor promediado conectado a carga resistiva.....	144
Figura 144 - Corrientes de salida de la rama superior (amarillo), la rama inferior (azul) y la suma de ambas (roja) del convertidor promediado conectado a carga resistiva.....	145
Figura 145 - Planta del convertidor promediado conectado a Red en el simulador en tiempo real.....	145
Figura 146 - Tensiones de salida del convertidor promediado conectado a Red	146
Figura 147 - Corrientes de salida del convertidor promediado conectado a Red	146
Figura 148 - Corrientes de salida de la rama superior, la rama inferior y la suma de ambas del convertidor promediado conectado a Red	147
Figura 149 - Mejor opción para interconectar los convertidores	148
Figura 150 - Planta de la línea HVDC en el simulador en tiempo real.....	148
Figura 151 - Valores de las constantes de los reguladores PI (elaboración propia).....	149
Figura 152 - Información de la simulación	150
Figura 153 - Tiempo promedio de ejecución del loop.....	150
Figura 154 - Tiempo de ejecución del loop en tiempo real.....	151
Figura 155 - Corrientes AC del convertidor realista.....	152
Figura 156 - Corrientes AC del convertidor promediado.....	152
Figura 157 - Tasa de distorsión armónica de las corrientes de salida del convertidor promediado	153
Figura 158 - Tasa de distorsión armónica de las corrientes de salida del convertidor realista	153
Figura 159 - Tensiones AC del convertidor realista.....	154
Figura 160 - Tensiones AC del convertidor promediado	154
Figura 161 - Comparación entre corrientes y tensiones de los dos convertidores	155
Figura 162 - Detalle de las corrientes. Comparación entre los dos convertidores (corriente del realista en rojo, corriente del promediado en amarillo).....	156
Figura 163 - Tensión en DC en terminales del convertidor realista	157

Figura 164 - Tensión en DC en terminales del convertidor promediado	157
Figura 165 - Corrientes en el interior del convertidor	158
Figura 166 - Cálculo de las corrientes circulantes	158
Figura 167 - Corrientes circulantes del convertidor realista. En amarillo la de la fase A, azul la de la fase B, rojo la de la fase C y en verde la suma de las tres	159
Figura 168 - Corrientes circulantes del convertidor realista sin componente de continua	159
Figura 169 - Valor medio de la suma de las corrientes circulantes de cada fase del convertidor realista	160
Figura 170 - Corrientes circulantes del convertidor promediado En amarillo la de la fase A, azul la de la fase B, rojo la de la fase C y en verde la suma de las tres	160
Figura 171 - Corrientes circulantes del convertidor promediado sin componente de continua.....	161
Figura 172 - Valor medio de la suma de las corrientes circulantes de cada fase del convertidor promediado	161

ÍNDICE DE ECUACIONES

Ecuación 1 - Transformación de Clarke.....	60
Ecuación 2 - Resultado Transformación de Park para el eje alfa.....	60
Ecuación 3 - Resultado Transformación de Park para el eje beta	60
Ecuación 4 - Resultado Transformación de Park para el eje gamma	60
Ecuación 5 - Matriz de rotación	61
Ecuación 6 - Transformación inversa de Clarke	61
Ecuación 7 - Cálculo potencia del convertidor de Simulink	98
Ecuación 8 - Potencia que entra en cada fase desde DC en el convertidor de Simulink.....	99
Ecuación 9 - Método de integración en lenguaje C. Razonamiento	120
Ecuación 10 - Método de integración en lenguaje C. Aplicado en el controlador	120
Ecuación 11 - Estructura de cálculo de productos y sumas del procesador.	120
Ecuación 12 - Cálculo de la integral de V_q sin multiplicar por el periodo. ...	120
Ecuación 13 - Cálculo de ω	121
Ecuación 14 - Cálculo corriente de salida a partir de las corrientes de rama	122
Ecuación 15 - Relación entre la potencia activa con I_d e I_q	123
Ecuación 16 - Relación entre la potencia activa y la componente d de la corriente.....	123
Ecuación 17 - Relación entre potencia reactiva con I_d e I_q	123
Ecuación 18 - Relación entre la potencia reactiva y la componente q de la corriente.....	123
Ecuación 19 - Cálculo de la inductancia equivalente a la salida del convertidor	123
Ecuación 20 - Tensión del convertidor corregida	124
Ecuación 21 - Tensión del convertidor corregida en ejes d y q	124
Ecuación 22 - Tensión del convertidor en ejes d y q con vectores unitarios d y q	124
Ecuación 23 - Derivada temporal del vector unitario d	124
Ecuación 24 - Derivada temporal del vector unitario q	125
Ecuación 25 - Componente correspondiente al eje d	125
Ecuación 26 - Componente correspondiente al eje q	125
Ecuación 27 - Variación temporal de I_{o_d} es un valor cercano a cero	125
Ecuación 28 -Variación temporal de I_{o_q} es un valor cercano a cero	125
Ecuación 29 - Cambio de referencia de potencia activa a referencia de corriente	132
Ecuación 30 - Cambio de referencia de potencia reactiva a referencia de corriente.....	132

Ecuación 31 - Relación entre la potencia de referencia y los límites de la saturación.....	134
Ecuación 32 - Cálculo de tensión del convertidor promediado	140
Ecuación 33 - Tensión en un condensador	140
Ecuación 34 - Desglose tensión en un condensador	140
Ecuación 35 - Fórmula para la evolución de la tensión en un condensador programada en C.....	141
Ecuación 36 - Cálculo de la tensión media en una rama en el convertidor promediado	141
Ecuación 37 - Cálculo de la potencia absorbida de la red por el convertidor realista	154
Ecuación 38 - Cálculo de la potencia entregada a la red por el convertidor promediado	154
Ecuación 39 - Cálculo de potencia que sale de cada fase del convertidor realista hacia la línea	159
Ecuación 40 - Cálculo de potencia que entra de cada fase del convertidor promediado hacia la línea	160

1.- CAPÍTULO 1: INTRODUCCIÓN Y OBJETIVOS

En las últimas décadas, el número de proyectos de líneas de transporte en corriente continua en alta tensión se ha incrementado de manera notable. Este tipo de líneas, conocidas como HVDC (*High Voltage Direct Current*, Alta tensión en corriente continua), son útiles en determinadas ocasiones en las que no es posible realizar una línea en corriente alterna, o no es rentable, como, por ejemplo, la imposibilidad de interconectar dos sistemas eléctricos de potencia que funcionen a distinta frecuencia o conectar a la red continental un archipiélago alejado geográficamente mediante un cable submarino de larga longitud.

Un proyecto de construcción de una línea eléctrica convencional ya supone una alta inversión inicial. Este tipo de líneas, hoy en día, conlleva una inversión inicial superior. Además, los convertidores AC/DC y DC/AC, están compuestos de semiconductores como, por ejemplo, transistores IGBT. También disponen de condensadores y bobinas. Estos elementos tienen unos costes considerables y son relativamente sensibles a sobre cargas y sobretensiones, por lo que en determinadas situaciones podrían dañarse. Además, los costes de operación dependen del número de conmutaciones en los semiconductores, del número de semiconductores y de la eficiencia de las conmutaciones. Asimismo, estos dispositivos suelen introducir armónicos, que deben ser reducidos al máximo. Si se tiene que invertir una gran cantidad de dinero, será mejor asegurarse de que todo va a funcionar bien, conocer dónde están los límites de la instalación, detectar posibles fallos a tiempo, conocer la forma más eficiente de operar con las menores pérdidas posibles, descubrir los armónicos que producen los convertidores para tratar de eliminarlos. Para ello, se podría crear un prototipo de convertidores y línea en el que tomar medidas y observar el funcionamiento real, pero cada vez que hubiera algún error habría que sustituir los componentes dañados, lo que supone un coste añadido al de la propia instalación de prueba. Además, en este caso existiría riesgo para las personas que operen en la línea de pruebas.

Pero existe una manera de probar todo lo necesario mediante la creación de un prototipo virtual, de bajo coste, sin la necesidad de construir una línea de prueba, sin poner en riesgo a ninguna persona y sin la posibilidad de estropear ningún elemento de ninguna instalación. Esto es posible mediante realización de una simulación en tiempo real. En esta simulación pueden realizarse infinitud de pruebas a partir de las cuales se puede mejorar el diseño inicial, hacerlo más seguro, más eficiente y con mayor fiabilidad.

Por estos motivos en este Trabajo de Fin de Grado se desarrolla una simulación en tiempo real de una línea HVDC que interconecta dos sistemas eléctricos independientes.

1.1.- Presentación de objetivos

1.1.1.- *Objetivo principal*

El objetivo principal es conseguir simular la transferencia de potencia a través de la línea HVDC desde una red trifásica hasta otra en tiempo real.

1.1.2.- *Objetivos parciales*

- Realizar un convertidor funcional en Simulink.
- Transferir las ideas del convertidor de Simulink al convertidor en tiempo real, construyendo en el simulador en tiempo real un convertidor conectado a una carga resistiva.
- Sincronizar el convertidor con una red trifásica.
- Conectar el convertidor a la red trifásica.
- Construir convertidor promediado conectado a una carga resistiva.
- Conectar convertidor promediado a la red trifásica.
- Interconectar ambos convertidores.

1.2.- Resumen del documento

El primer capítulo corresponde a la introducción y la descripción de los objetivos del trabajo, donde se expone la motivación para la realización de este proyecto y se mencionan los diferentes objetivos. También se encuentra ubicado en este capítulo este resumen del contenido de la memoria.

En el segundo capítulo se desarrolla el estado del arte. En este apartado se expone todo lo necesario para comprender este trabajo. Esta sección comienza con una introducción histórica sobre los dos tipos de corrientes que existen, la corriente continua y la corriente alterna. A continuación, se presenta la tecnología HVDC, junto con varios proyectos que se han llevado a cabo en los últimos años. Tras esto, se exponen los diferentes tipos de HVDC que se conocen y las diferencias entre las líneas de transporte convencionales y los HVDC, así como las diferencias de costes. Seguido, se muestran los tipos de convertidores que se utilizan en las líneas HVDC para transformar la corriente eléctrica. Después se introduce la tecnología de los convertidores MMC y la técnica de PWM. A continuación, hay un apartado destinado a presentar una de las herramientas de simulación que se ha utilizado que es Matlab/Simulink. Por último, hay un apartado destinado a explicar ciertos fundamentos de la simulación en tiempo real, algunos beneficios y, también, se presenta el entorno en el que se ha realizado la simulación en tiempo real.

En el capítulo 3, es donde se desarrolla el trabajo. Inicialmente se presenta la línea que va a ser simulada. En segundo lugar, se habla del convertidor creado

a través de Matlab/Simulink. Después, presenta la planta del HVDC en el simulador en tiempo real y se exponen las partes más relevantes del código del controlador. A continuación, se muestra el convertidor realista y algunas gráficas para observar el comportamiento del convertidor. Después se hace lo propio con el convertidor promediado, para comparar su funcionamiento. Finalmente, se muestran los resultados de la simulación de la línea HVDC.

El capítulo 4 va destinado a exponer las repercusiones sociales, económicas, seguridad y salud del TFG.

En cuanto al capítulo 5, corresponde a la conclusión. En esta sección se contrasta el nivel de culminación de los objetivos parciales y del objetivo principal del TFG. También se presentan algunas de las posibles líneas futuras a raíz de este trabajo.

Finalmente, el capítulo 6, donde se encuentra la bibliografía de consulta que se ha utilizado para realizar este trabajo.

2.- CAPÍTULO 2: ESTADO DEL ARTE

2.1.- La Guerra de las Corrientes

Entre 1888 y 1889 comenzó lo que hoy día se conoce como la Guerra de las Corrientes. Por un lado, se encontraba el inventor y empresario Thomas Alva Edison, defensor del uso de la corriente continua, que se había hecho con la patente de la primera dinamo industrial movida por una máquina de vapor. En 1882, Edison instaló la primera central eléctrica y red de distribución en corriente continua en Pearl Street, Nueva York. Por el otro lado, George Westinghouse, empresario defensor de la utilización de corriente alterna, compró las patentes del transformador en 1885 y del motor eléctrico de inducción, inventado por Nikola Tesla en 1888. En 1886, Westinghouse realiza la primera instalación de alumbrado en corriente alterna, en Great Barrington, Massachusetts. La guerra de las corrientes fue la carrera por ver que tecnología se imponía.

La corriente continua presentó el problema de las pérdidas por efecto Joule, que se deben a la pérdida de energía en forma de calor al circular la corriente por el cable. Para tratar de minimizar estas pérdidas, se colocaban los generadores en los centros de los lugares de consumo, por lo que fueron denominadas “centrales”.

En cambio, en corriente alterna, los generadores eran más sencillos y fiables que los de corriente continua. El transformador permitía elevar la tensión para reducir la corriente que circulaba por los cables, reduciendo así las pérdidas. Gracias a esto, eran capaces de transportar la energía a grandes distancias, y, por tanto, se podía aprovechar recursos alejados de las ciudades, como los saltos de agua.

Por estos motivos, la corriente alterna se llevó la victoria. En 1891 se creó el primer sistema trifásico y en 1895 se construyó la central de las Cataratas del Niagra, que confirmaba a la corriente alterna como la forma óptima de generar, transportar y distribuir la energía eléctrica. (1)

2.2.- Introducción histórica del HVDC

En los últimos años, con el desarrollo de la electrónica de potencia, la corriente continua ha ganado importancia en el transporte de electricidad, siendo hoy día, una opción a considerar cuando la línea cumple unas condiciones específicas: Líneas de gran longitud, subterráneas o submarinas (aéreas en menor medida), gran cantidad de potencia a transferir.

La tecnología que se emplea para el transporte en corriente continua es el HVDC (High Voltage Direct Current) alta tensión en corriente continua. Esta tecnología comenzó a desarrollarse en la década de 1930, cuando fueron

inventados los rectificadores de arco de mercurio. En 1941 se firmó el primer contrato comercial de HVDC en Berlín. Una línea subterránea de 115 km debía alimentar 60 MW a la ciudad. En 1945 estaba listo para entrar en operación, pero durante el fin de la Segunda Guerra Mundial fue desmantelada y nunca llegó a utilizarse. Por tanto, la primera vez que se utilizó este método fue en 1954 entre Suecia y la isla de Gotland (90 km), que transportaba una potencia de 20 MW a 100 kV. (2)

2.3.- HVDC en la actualidad

En la actualidad estos números no son tan modestos, ya que el HVDC más largo y potente del mundo, ubicado en China (Xiangjiaba – Shanghai), es de 2071 km de longitud y 6400 MW, en Australia, el sistema subterráneo Murraylink de 180 km, 220 MW a 150 kV o la conexión submarina entre Noruega y Países Bajos (NorNed) de 580 km, 700 MW a 450 kV. (3)

En la Unión Europea, entre los años 2014 y 2018, se llevó a cabo el proyecto *Best Path*, cuyo objetivo era el estudio de diversas soluciones con el fin de transmitir “cantidades masivas de energía de fuentes renovables como la solar y eólica marina, satisfaciendo la demanda de manera competitiva y reduciendo los niveles de CO2 producidos” (4). Entre las soluciones se estudió:

- Interacciones entre parques eólicos marinos y redes HVDC.
- Interoperabilidad entre estaciones convertidoras HVDC.
- Mejora de enlaces HVDC multiterminales.
- Cable superconductor de corriente continua.

2.4.- HVDC en España

En España, en 2012, por el llamado proyecto Rómulo de Red Eléctrica de España, se creó en enlace entre la península y Mallorca, una conexión submarina de 244 km de longitud, 400 MW a 250 kV, llegando hasta 1485 m de profundidad, fue la primera línea HVDC submarina en España. También en 2015, el nuevo enlace entre España y Francia, subterráneo a través de los pirineos, de 64.5 km, 1400 MW a 320 kV, por el proyecto INELFE. Está en proyecto otro enlace HVDC con Francia a través del Golfo de Vizcaya entre Bilbao y Burdeos, de unos 400 km, con una capacidad de transporte de 2000 MW. (5) (6) (7)

2.5.- Clasificación de los HVDC

Los HVDC se componen de al menos dos convertidores y la línea que une las dos estaciones convertidoras. Los convertidores tienen la capacidad de transformar la corriente alterna en corriente continua y viceversa. Hay un convertidor en cada extremo de la línea. Existen diferentes formas de conexión

entre estaciones convertidoras, como también existen diferentes configuraciones de HVDC.

2.5.1.- HVDC según el tipo de conexión

Según el tipo de conexión, un HVDC puede ser monopolar, bipolar u homopolar:

2.5.1.1.- Monopolar

Consta de un solo conductor para transmitir la energía eléctrica entre estaciones convertidoras, de polaridad negativa para reducir el efecto corona y las interferencias. El retorno se produce por tierra o mar generalmente, pero en ocasiones es necesario que haya retorno por cable, también llamado retorno metálico (ver Figura 1).

Ventajas de la conexión monopolar:

- Este es el tipo de conexión más económico.
- Necesita menor mantenimiento que otros tipos de conexión.
- Solamente utiliza un conductor, por lo que su diseño es el más simple.

Como desventajas de la conexión monopolar:

- Menor fiabilidad, ya que, en caso de suceder una falta, toda la línea queda desconectada.
- Solo se utilizan para el caso de transporte para bajas potencias.
- En enlaces submarinos, el retorno por mar puede afectar a las brújulas magnéticas de los barcos.

(8)

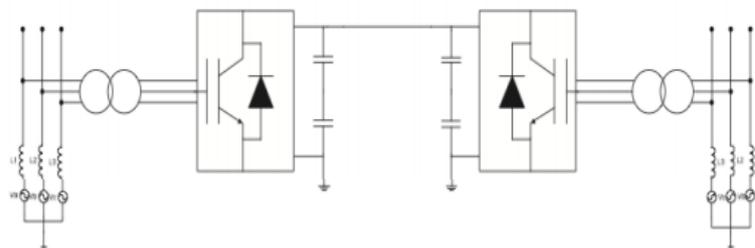


Figura 1 - Esquema de un HVDC con conexión monopolar (9)

2.5.1.2.- Bipolar

Este tipo de conexión se compone de dos sistemas monopolares, de los cuales, uno hace de polo positivo y el otro de polo negativo, de tal forma que hay dos conductores, uno positivo y otro negativo y cada polo tiene dos convertidores (ver Figura 2). Entonces, en cada terminal también tiene dos convertidores de igual tensión (uno del polo positivo y otro del negativo) conectados entre sí.

Entre la unión de los convertidores de cada terminal se sitúan los puntos neutros conectados a tierra en ambos extremos. También cabe la posibilidad de unir los neutros a través de un conductor, con un único extremo conectado a tierra (retorno metálico). En condiciones normales la corriente es igual en los dos polos y no hay circulación por tierra. Este tipo de conexión también puede funcionar como monopolar en caso de fallo en uno de los dos polos, por lo que aumenta la fiabilidad de la instalación. También tiene una mayor capacidad de transporte de potencia que la monopolar.

Ventajas del tipo de conexión bipolar:

- Una falta en una de las líneas no afecta a la operación de la otra línea.
- Durante la falta en una de las líneas, este tipo de conexión puede seguir funcionando como si fuera una conexión monopolar.
- La dirección del flujo de la potencia puede ser cambiada cambiando la polaridad de los polos.

Desventajas:

- En comparación con la conexión homopolar, las interferencias y el efecto corona son mayores.
- Económicamente es más costoso.
- La instalación es más compleja.

(8)

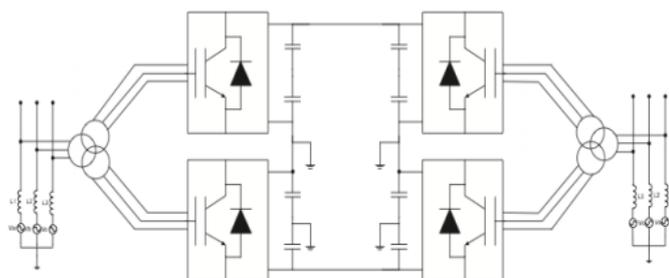


Figura 2 - Esquema de un HVDC con conexión bipolar (9)

2.5.1.3.- Homopolar

Al igual que en la conexión bipolar, la conexión homopolar está formada por dos sistemas monopolares, pero en este caso, ambos tienen la misma polaridad, normalmente negativa, y el retorno por tierra (ver Figura 3). Este tipo se utiliza en menor medida. Cuando sucede una falta, al igual que en la conexión bipolar, también es capaz de funcionar como monopolar.

Ventajas:

- El efecto corona y las interferencias son menores debido al uso de la polaridad negativa. Esto se traduce en menos pérdidas en el transporte.
- Se podría operar en la línea bajo condiciones de falta, debido a la independencia entre los dos polos.

Desventajas:

- Al haber circulación por tierra, se debe tener en cuenta a la hora de diseñar la línea para que no afecte negativamente.

(8)

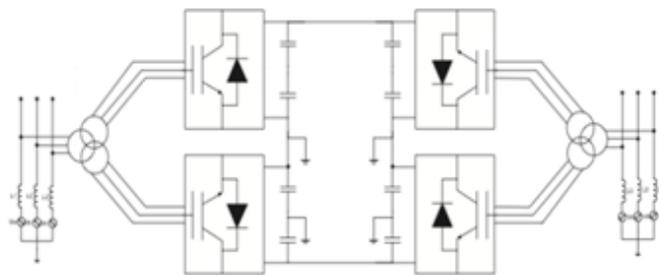


Figura 3 - Esquema de un HVDC con conexión homopolar (9)

2.5.2.- HVDC según su configuración

Según la configuración, existen diferentes tipos de HVDC:

2.5.2.1.- Punto a punto

Consiste en dos estaciones convertidoras conectadas entre sí mediante una línea HVDC (ver Figura 4). Es la topología más utilizada para interconectar puntos a grandes distancias, especialmente en líneas submarinas, para conectar parques eólicos offshore o sistemas aislados.

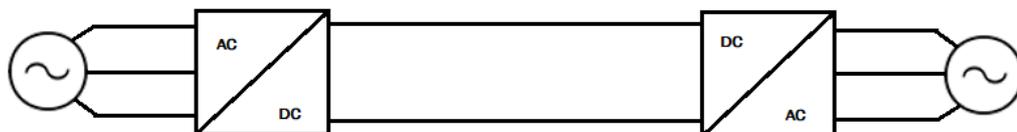


Figura 4 - Esquema de un HVDC con configuración punto a punto (elaboración propia)

2.5.2.2.- Back to back

Se trata de la interconexión de dos convertidores en la misma estación convertidora (ver Figura 5), en la que se realiza la rectificación de la onda para que, posteriormente se invierta de nuevo. En esta configuración no es necesaria una línea para transmitir la potencia. Su principal utilidad consiste en conectar dos sistemas a distinta frecuencia.

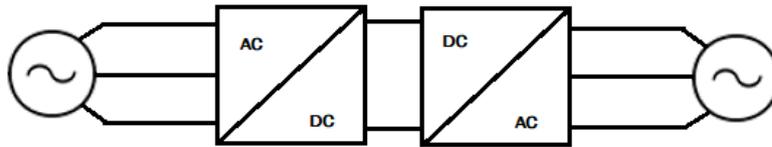


Figura 5 – Esquema de un HVDC con configuración Back to Back (elaboración propia)

2.5.2.3.- Multiterminal

La configuración multiterminal permite la conexión de tres o más estaciones convertidoras, pudiendo, de esta manera crear una red de transmisión en corriente continua. Esto permite aumentar la fiabilidad de las conexiones por HVDC, ya que en caso de fallo habría caminos alternativos entre los puntos de conexión. Existen tres tipos de configuración multiterminal:

- Paralela: todas las subestaciones se encuentran a la misma tensión (Figura 7).
- serie: cada una a tensión diferente (Figura 6).
- mixta: combinación de serie y paralelo.

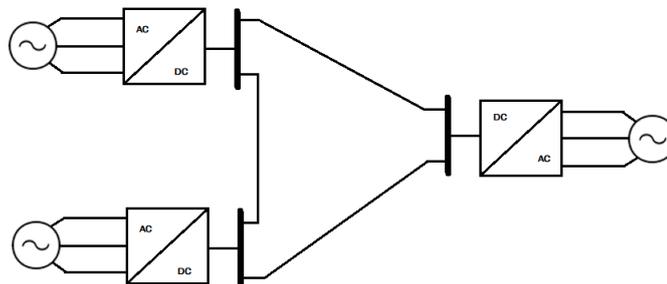


Figura 6 - Esquema de un HVDC con configuración multiterminal en serie (elaboración propia)

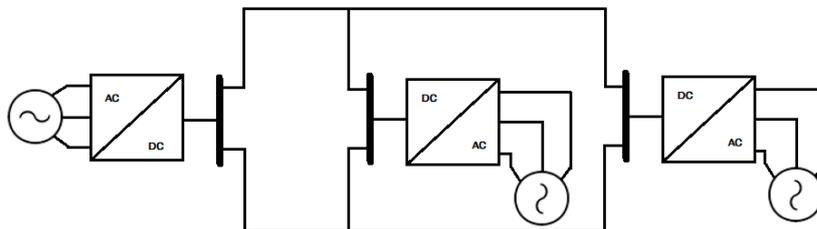


Figura 7 - Esquema de un HVDC con configuración multiterminal en paralelo (elaboración propia)

2.5.2.4.- Unitaria

En este caso, el convertidor que hace de rectificador se conecta directamente al generador, entonces, a efectos prácticos es como si la energía fuera generada en corriente continua. Gracias a esto, se podría generar electricidad

a una frecuencia independiente de la frecuencia del sistema. Es como un back to back: en un lado la frecuencia del sistema a frecuencia fija, al otro lado la frecuencia variable del generador.

2.6.- HVDC y redes AC

Las líneas de transporte de energía eléctrica han sido históricamente de corriente alterna, generalmente mediante líneas aéreas y conductores desnudos. Esto se debe a que al generar la energía en corriente alterna, se puede elevar la tensión fácilmente y sin apenas pérdidas en transformadores, disminuyendo así las pérdidas durante el transporte. Hoy en día, los costes de instalación y los costes de operación de un HVDC, por lo general no pueden competir con los de las líneas de corriente alterna. Sin embargo, estas redes de corriente alterna presentan algunas limitaciones, mientras que las líneas HVDC poseen ciertas cualidades que pueden ayudar a minimizar algunas de estas limitaciones con el uso combinado de las líneas de corriente alterna y el HVDC.

2.6.1.- *Control de frecuencia, tensión, factor de potencia, potencia activa y reactiva*

En el transporte en corriente alterna, se deben tener en cuenta una serie de factores como la frecuencia, la tensión en los nodos o la caída de tensión que hay en las líneas. Todos estos factores tienden a cambiar continuamente por las constantes alteraciones en la energía que se demanda o las perturbaciones y transitorios que puedan aparecer.

Es conocida la existencia de una fuerte dependencia entre la circulación de potencia reactiva en una línea y la caída de tensión en la misma y, también, por otro lado, que hay una fuerte relación entre la circulación de potencia activa en una línea y el ángulo de fase. Por este motivo, al variar el carácter de la carga, ya sea hacia más inductivo o hacia más capacitivo, estas variaciones afectan a la caída de tensión. A su vez, manteniendo la potencia reactiva constante, si aumenta la demanda de potencia activa, también varía el ángulo de fase de la tensión ($\delta = \varphi_2 - \varphi_1$; siendo φ_1 el ángulo de la tensión de origen y φ_2 el de la tensión de destino en una línea).

Las líneas tienen unos límites, regulados por ley, con respecto a la caída de tensión con el objetivo de evitar defectos de aislamiento y mantener los valores de tensión adecuados para los consumidores.

También existen límites físicos a la capacidad de transporte de las líneas para garantizar la estabilidad. Por un lado, existe el límite de estabilidad estática y por otro lado el límite de estabilidad de tensión:

El límite de estabilidad estática de una línea es el límite de potencia activa transmitida a la carga cuando $\delta = \pi/2$ (Figura 8). La capacidad estática puede

mejorarse elevando la tensión de servicio y/o disminuyendo el valor de la reactancia de la línea, gracias a la compensación de reactiva en serie y en paralelo.

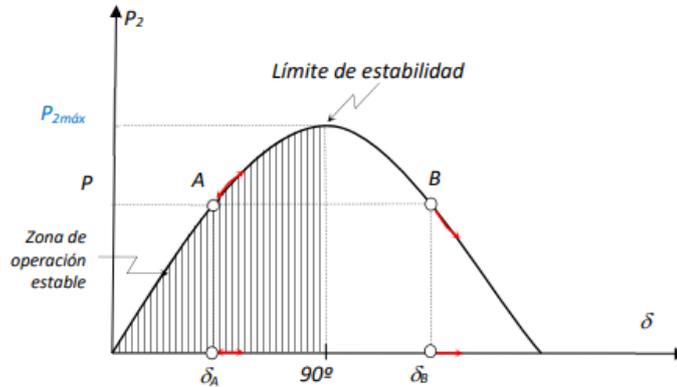


Figura 8 - Límite de estabilidad estática (10)

En cuanto a los límites de estabilidad de tensión, se conoce que la tensión y la potencia activa se encuentran ampliamente relacionadas. En la Figura 9, se puede ver distintas curvas P-V, en valores por unidad, en función del factor de potencia. También se puede ver en la izquierda como existe un punto a partir del cual existe inestabilidad.

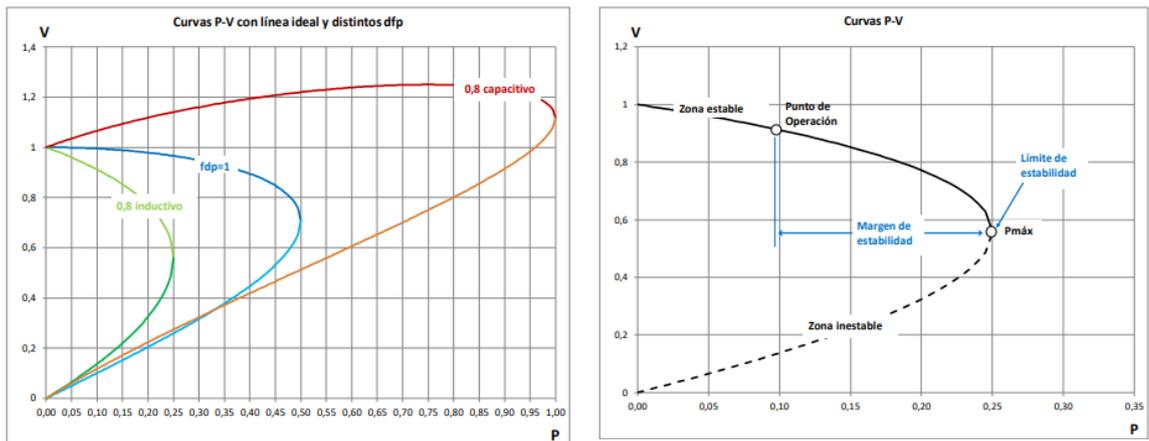


Figura 9 - Curvas P-V para diferentes f.d.p. Y zonas de la curva P-V (10)

Además, también existe una estrecha relación entre la variación de la tensión y la potencia reactiva. En la Figura 10 se puede ver la evolución de las curvas

Q-V según el factor de potencia. A su vez, se muestra que a partir de un punto límite de estabilidad, el sistema se hace inestable.

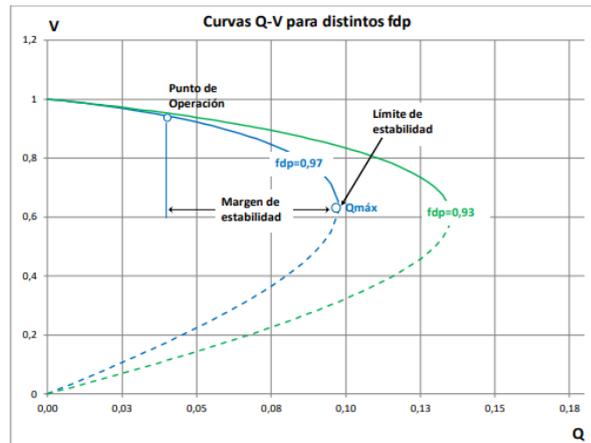


Figura 10 - Curvas Q-V para distintos f.d.p. y zonas de la curva (10)

Debido a estas dependencias entre Q y ΔV y P y δ , en las líneas AC es necesario un control constante de estos valores para garantizar que no se rebasen en ningún momento los respectivos límites, además de unas condiciones estándar de frecuencia y tensión, tratando de hacerlo de la manera más económica posible. Con este objetivo existe una gran infraestructura de control de la frecuencia y control de la tensión con tres niveles diferentes de control:

- Control primario (a nivel local)
- Control secundario (a nivel regional)
- Control terciario (a nivel global de la red).

En cuanto a la tensión en los nodos, debe quedarse en el valor de referencia, ya que estas tensiones nodales, dadas desde un punto de vista global de la red, permiten que la potencia fluya por las líneas de la forma más adecuada, llevando la potencia desde los puntos donde se genera hasta los puntos donde se consume, con las menores pérdidas.

Un requisito fundamental para mantener la estabilidad es que todos los generadores deben funcionar en sincronismo. La variabilidad de la demanda afecta a la frecuencia de la red de la siguiente manera: Tenemos una red en un estado estacionario a una frecuencia exactamente igual a la de referencia. Desde este punto, al aumentar la demanda en esta red, los generadores que la alimentan verían cómo aumenta la carga y tenderían a decelerar. Para evitar de deceleren haciendo caer la frecuencia de la red, se debe aumentar la velocidad de las turbinas que mueven a los generadores, haciendo que la frecuencia permanezca en su valor de referencia. Si, por el contrario, desde el punto estacionario inicial, la demanda de la red cae, los generadores

acelerarían, para evitarlo, se debería tratar de reducir la velocidad de giro de los generadores para que la frecuencia se mantenga invariable.

Todas estas acciones mencionadas a lo largo de este apartado tienen el fin de poder transmitir la energía desde los generadores hasta los consumidores a través de una red estable cuya onda cumple con la normativa, garantizando que los aparatos que se conecten a esta red funcionen correctamente y sin dañarse.

Estas causas suponen unos costes económicos importantes asociados a los elementos de compensación de reactiva o la infraestructura para el control del sistema mencionada anteriormente (control primario, secundario, terciario, reservas estratégicas y servicios de interrumpibilidad).

Una línea HVDC no presenta esta serie de problemas con respecto a la frecuencia, la tensión o la caída de tensión, ya que:

- El convertidor transforma la corriente alterna, independientemente de su frecuencia y tensión y la convierte a corriente continua de unas características determinadas donde ya no influyen ni la frecuencia ni la fase de la tensión o el factor de potencia.
- En el transporte hay cierta caída de tensión debido a las pérdidas por efecto Joule, pero no se transporta, en ningún caso, potencia reactiva, por lo que ya no es un factor que varía con la carga.
- La potencia que es capaz de transmitir un HVDC es mayor que la de una línea en alterna de características similares, ya que en corriente continua no se transporta la potencia reactiva ni hay un ángulo límite para la estabilidad.
- En cuanto a la tensión en el lado de continua, es un valor de diseño que depende de la longitud de la línea y de la potencia que se desea transmitir. Se debe tener en cuenta que a mayor tensión menores serán las pérdidas en el conductor, pero mayores podrían ser las pérdidas en la conversión y más caros serán los convertidores.
- Al final de la línea, la corriente se vuelve a transformar de continua a corriente alterna, pudiendo ser esta de las condiciones de frecuencia, tensión y el factor de potencia que se requieran en dicho extremo de la línea.
- Los convertidores del HVDC son capaces de absorber de la red la potencia que se desee con total precisión y transportarla al otro extremo.

Por lo tanto, incorporando alguna línea HVCD dentro de una red de corriente alterna, la red podría beneficiarse de este control preciso del flujo de potencia

y del control dinámico de la frecuencia y de la tensión propios de los HVDC. Como consecuencia, aumentaría la estabilidad de la red, facilitando en cierto modo, la operación del sistema.

2.6.2.- *Interconexión entre sistemas eléctricos*

Mediante el transporte en corriente alterna, no se puede interconectar sistemas eléctricos de potencia que funcionen a distinta frecuencia nominal. Además, en caso de conectar dos sistemas eléctricos que tengan la misma frecuencia nominal, las posibles fallas se podrían transmitir de un sistema a otro, ocasionando problemas de estabilidad.

El HVDC permite interconectar dos sistemas eléctricos de potencia que funcionen a distinta frecuencia nominal, pero, además, también impide que se transmitan fallas de un sistema al otro, consecuencia de ese control preciso y dinámico de la potencia que se transmite.

Por lo tanto, con un HVDC, se podrían compartir fuentes de energía entre diferentes sistemas eléctricos, independientemente de su frecuencia nominal, esto permite aprovechar al máximo las energías renovables, ya que tienen una alta variabilidad. Se podría enviar la energía de las zonas donde se está produciendo un excedente hacia las zonas donde en ese instante no lo haya. Un ejemplo de compartir una fuente de energía entre diferentes sistemas podría ser un parque eólico offshore que alimente a varios países cercanos. De esta forma se puede reducir el uso de combustibles fósiles para producir electricidad. Esto supone la creación de redes HVDC, mediante configuraciones multiterminales o la unión de varias líneas punto a punto. Incluso, es posible unir con una red HVDC varios parques eólicos offshore con sus respectivas conexiones a tierra firme, de tal forma que en caso de fallar una de las conexiones, los parques eólicos pueden mantener el servicio, aumentando la fiabilidad.

2.6.3.- *Transporte de energía mediante cable aislado, submarino y subterráneo*

Por otra parte, el transporte en corriente alterna mediante cable aislado está limitado por la longitud de la línea, ya que en este tipo de cables los efectos capacitivos son mayores y aumentan con la distancia. Además, en cables submarinos y subterráneos es muy complicado poner elementos de compensación de reactiva, por lo que a partir de cierta distancia comienza a ser muy difícil transmitir potencia mediante corriente alterna. Este hecho implica también ciertas consecuencias económicas, ya que no poder alimentar sistemas aislados, implica tener que generar la energía en la propia isla con combustibles fósiles, más caros y contaminantes y con un bajo rendimiento, ya que, aunque puedan disponer de energías renovables, estas cuentan con una

gran variabilidad y no siempre se puede contar con ellas. Además, el menor número de generadores implica menor estabilidad y fiabilidad en la red aislada.

En cuanto al transporte mediante cables aislados, el HVDC marca la diferencia. Sobre todo, en líneas submarinas y subterráneas donde el HVDC no tiene límite técnico en cuanto a la longitud de la línea ya que en corriente continua no hay problemas de capacitancias ni es necesario colocar elementos de compensación de reactiva. Esto permite la conexión entre los sistemas eléctricos aislados y tierra firme, aumentando la estabilidad de estos pequeños sistemas, pudiendo aprovechar energía generada en otros puntos, evitando en cierta medida que tengan que producir su electricidad con combustibles fósiles.

2.6.4.- Otras aplicaciones de los HVDC

Gracias a los HVDC existe la posibilidad de generar a una frecuencia independiente a la de la red, es decir, permite la independencia entre el control de la frecuencia y la generación de energía eléctrica, ya que, como se verá más adelante, con la configuración unitaria las fluctuaciones de la frecuencia en la generación no afectan a la red, ni las fluctuaciones de la frecuencia a causa de variaciones en la demanda afectan a la generación, ya que ambas frecuencias serían independientes. Esto podría aumentar la estabilidad de la red.

En la actualidad, la principal aplicación de los HVDC es el transporte de grandes cantidades de potencia a través de grandes distancias, sobre todo en líneas subterráneas y submarinas, donde los costes del HVDC son menores que los de una línea convencional.

Aunque los HVDC presentan ciertas ventajas y cualidades adicionales a las de las líneas convencionales, hoy en día, también presentan una serie de inconvenientes: (11)

- En corriente continua no es posible utilizar transformadores para cambiar los niveles de tensión, como si es posible con los transformadores con tomas de AC.
- Los costes de un HVDC son muy elevados con respecto a los de las líneas de corriente alterna, por lo que su uso queda reducido a ciertas situaciones.
- Los convertidores generan armónicos. Esto implica tener que utilizar filtros y transformadores con topologías diferentes con el fin de eliminar o reducir los efectos de dichos armónicos.
- Los convertidores de un HVDC tienen menor capacidad de sobrecarga. Los semiconductores marcan el límite soportable.

- Los convertidores de los HVDC tienen mayores pérdidas que las de las subestaciones en AC debido a las conmutaciones en los semiconductores y las propias pérdidas de los semiconductores.

2.7.- HVDC vs AC en cuanto a costes

Para que una línea HVDC sea viable, los costes de los convertidores, los costes de las pérdidas en dichos convertidores y los costes de las pérdidas en el transporte deben ser menores que los costes de la línea AC y los costes de las pérdidas de transporte en AC, los elementos de compensación, subestaciones, etc.

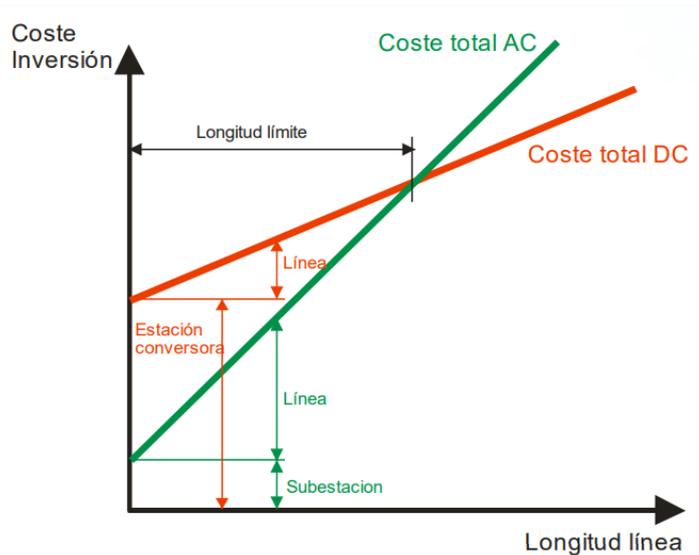


Figura 11 - Costes HVDC vs costes HVAC (12)

Como se puede ver en la gráfica de la Figura 11, los costes de las estaciones convertidoras de un HVDC superan con creces a los costes de una subestación de una línea en corriente alterna. A su vez, se observa que el coste por unidad de longitud de la línea HVDC es menor que el de la línea en corriente alterna, ya que el HVDC requiere de una instalación más simple, con menor número de cables, no se necesitan elementos de compensación de reactiva. Los costes de la línea aumentan en mayor medida en las líneas de corriente alterna, debido al mayor número de cables, estructura más compleja y la necesidad de más elementos de compensación de reactiva según aumenta la longitud, mientras que, en un HVDC, a medida que aumenta la longitud solamente aumenta el coste que supone la mayor longitud del cable. Debido a esto, existe una longitud a partir de la cual los costes del HVDC son menores que los de la línea en corriente alterna. Esta longitud límite es diferente dependiendo de múltiples factores, por lo que el HVDC empieza a ser una opción a considerar cuando la línea cumple con una serie de características. En líneas subterráneas y submarinas, con cables aislados, la longitud límite sería unos 100 km, debido

a los efectos capacitivos en los cables aislados en corriente alterna y la dificultad de compensar reactiva, mientras que, en líneas aéreas con conductores desnudos, los efectos capacitivos en los cables en corriente alterna son menores y existe la posibilidad de compensar la potencia reactiva, por lo que en líneas aéreas, la longitud límite asciende a más de 1000 km. (12)

2.8.- Convertidores de los HVDC

En cuanto a las tecnologías de conversión de corriente alterna a continua y de continua a alterna, se ha comentado al inicio del capítulo que al principio se utilizaban rectificadores de arco de mercurio. Actualmente, las tecnologías más utilizadas son las llamadas LCC (Line Commutated Converter) Convertidor por línea conmutada (conmutado por red) y VSC (Voltage Sourced Converter) Convertidor por fuente de tensión (auto conmutada). Ambos tipos de convertidores utilizan electrónica de potencia.

2.8.1.- Convertidor LCC

El LCC (*Line Commutated Current Source Converter*) Convertidor conmutado por línea, es el tipo de convertidor más longevo de entre las dos tecnologías actuales, ya que se utilizó de forma exclusiva entre los años 1972 y 2000, por ello es llamado el convertidor clásico. Su funcionamiento está basado en tiristores. (12)

Un tiristor es un semiconductor similar al diodo, que permite el paso de la corriente solamente en un sentido, cuando circula del ánodo hacia el cátodo, pero, en el caso del tiristor no permite el paso de la corriente hasta que no se le aplica una corriente, en la puerta. Dicha corriente recibe el nombre de corriente de disparo. Entonces, es posible elegir el momento en el que entra en funcionamiento controlando la corriente de disparo. No se puede decidir cuándo se apaga el tiristor, este deja de conducir la corriente automáticamente cuando la corriente pasa por cero y se invierte la polaridad.

Como se puede ver en la Figura 12, el tiristor se comporta como un interruptor abierto cuando la corriente circula desde el cátodo hacia el ánodo o cuando no se ha aplicado la señal de disparo, y, se comporta como un interruptor cerrado tras aplicar la corriente de disparo.

En la Figura 12 se puede ver la forma que tiene un tiristor real.

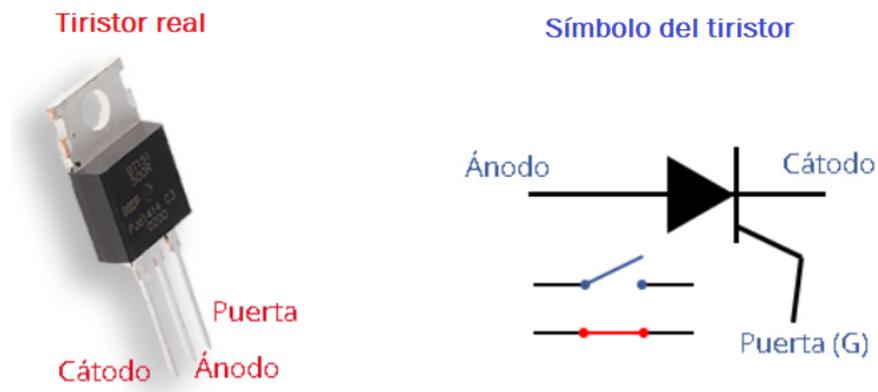


Figura 12 - Imagen de tiristor real y símbolo del tiristor (13)

Este tipo de convertidor se basa en los rectificadores trifásicos en puente, como el rectificador de seis pulsos o el rectificador de 12 pulsos (Figura 13). Este último es el que más se utiliza en los convertidores LCC debido a que genera menor rizado y menor cantidad de armónicos, por lo que necesita menos filtrado, además, con este rectificador es posible transmitir más potencia que con uno de seis pulsos. (2) (12)

El rectificador de 12 pulsos recibe este nombre debido a que en un periodo de la tensión de alimentación en corriente alterna genera 12 cúpulas en corriente continua.

Se compone de dos rectificadores trifásicos de 6 pulsos, uno de ellos conectado a la red trifásica mediante un transformador con conexión estrella-estrella y el otro conectado a través de una conexión estrella-triángulo. Al utilizar estas conexiones, los armónicos 5° , 7° , 17° y 19° se anulan debido a que en cada rectificador se generan estos mismos armónicos, pero de secuencia opuesta: (14)

- Armónicos estrella-estrella: $5i$, $7d$, $11i$, $13d$, $17i$, $19d$.
- Armónicos estrella-triángulo: $5d$, $7i$, $11i$, $13d$, $17d$, $19i$.

El rectificador que está en estrella-triángulo está retrasado 30° con respecto al otro rectificador, a causa de la propia conexión estrella triángulo. Por lo tanto, en el rectificador estrella-estrella llegan tres tensiones desfasadas 120° entre sí, y, el rectificador estrella-triángulo se alimenta de tres tensiones desfasadas 120° entre sí, pero retrasadas 30° de las primeras. De esta manera las seis tensiones hacen de referencia para las conmutaciones de los tiristores. Controlando el ángulo de disparo se puede modificar la tensión de salida en corriente continua.

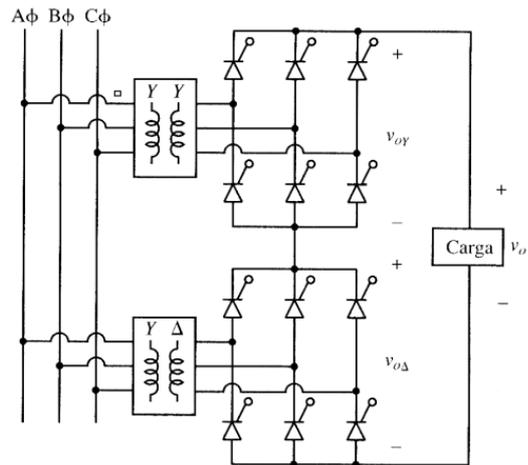


Figura 13 - Esquema de un rectificador de doce pulsos

Por lo tanto, debido al uso del rectificador de 12 pulsos y al uso de tiristores, los convertidores del tipo LCC presentan una serie de características: (12)

- Cuando la señal de disparo no permite el paso de la corriente durante todo el periodo, la onda generada no es senoidal perfecta, por lo que introduce armónicos. Al introducir armónicos, es necesaria la implementación de filtros y topologías de conexión (transformadores de diseño especial) para tratar de minimizarlos.
- El convertidor consume potencia reactiva de la red de alterna, aportada por los filtros de armónicos y condensadores en paralelo.
- Debido al uso de tiristores, la corriente circula siempre en el mismo sentido, es decir, cuando la corriente en alterna es positiva (Figura 14 izquierda) circula en sentido favorable para los tiristores, y, cuando la corriente alterna es negativa (Figura 14 derecha), la polaridad en el lado de continua cambia de tal forma que la corriente continua circula en sentido a favor de los tiristores.

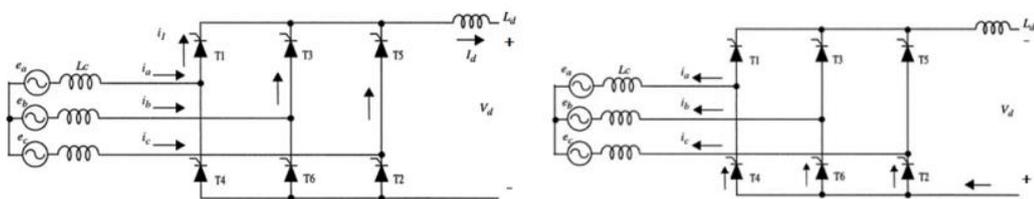


Figura 14 - Corriente positiva (izq.) y negativa (dcha.) por un rectificador tipo puente (12)

- Los convertidores LCC son capaces de limitar cortocircuitos en el lado de continua por su propio diseño, ya que los tiristores pueden actuar como interruptores simplemente al apagar la señal de disparo la corriente se extingue cuando pasa por cero.

- Los tiristores toman como referencia la tensión en corriente alterna de la red para realizar las conmutaciones, es decir, la conmutación se produce cuando la corriente alterna pasa por cero y los tiristores que estaban conduciendo pasan a actuar como un interruptor abierto.

Esta dependencia entre la onda de la tensión y las conmutaciones de los tiristores hace que este convertidor sea vulnerable ante perturbaciones en la red de corriente alterna, ya que es posible que distorsiones en la red puedan provocar un fallo en las conmutaciones del convertidor, pudiendo incluso apagar temporalmente todo el sistema HVDC. (2)

- Este tipo de convertidores junto con sus equipos auxiliares para el filtrado de armónicos y compensadores de reactiva, requieren una gran cantidad de espacio, por lo que, aunque sean en general más baratos que los VSC por su mayor simplicidad, no son una buena opción para aplicaciones off shore como, por ejemplo, comunicar un parque eólico marino con tierra firme, pues haría falta una plataforma de mayor tamaño para poder utilizar un convertidor LCC. (2)

2.8.2.- **Convertidor VSC**

La tecnología del LCC, convive con la VSC, (*Voltage Sourced Converter*) convertidor por fuente de tensión. Suele estar formado por dispositivos de conmutación forzada como los transistores IGBT (Insulated Gate Bipolar Transistor, Transistor Bipolar de Puerta Aislada) controlado por tensión, o, en el caso de los más modernos, con GTO (Gate Turn Off Thyristor), que es un tipo de tiristor en el que es posible controlar cuando se enciende y cuando se apaga. El método más extendido para el control de las conmutaciones en los convertidores VSC es el PWM (*Pulse Width Modulation*) Modulación por ancho de pulso, que se verá más adelante. (2) (12)

El funcionamiento del convertidor es similar tanto con IGBTs como con GTOs, ya que con ambos se puede controlar el paso de la corriente mediante señales de control. Gracias a esta capacidad de control:

- La onda que inyectan es senoidal pura, por lo que generan menos armónicos en la red de corriente alterna. Por este motivo, los dispositivos para filtrar los armónicos son de menor tamaño y, por tanto, más económicos.
- Este tipo de convertidores tienen mayor flexibilidad para el control de potencia tanto activa como reactiva y para el control de la frecuencia. Esto permite al convertidor operar en los cuatro cuadrantes del plano P-Q. Es decir, es posible elegir el factor de potencia que se desee con total

libertad, permite absorber o inyectar la potencia activa y/o reactiva que sea necesaria.

Por otro lado, al tener una mayor frecuencia de conmutación, estos dispositivos tienen más pérdidas que los LCC.

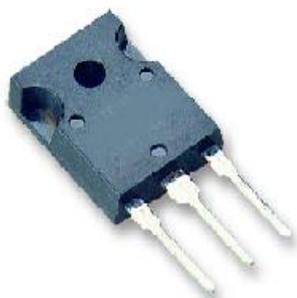
Tampoco pueden limitar cortocircuitos en el lado de continua, siendo necesarios interruptores para cortar la corriente en caso de producirse una falta.

El convertidor que se ha utilizado en la simulación es del tipo VSC y funciona con transistores IGBT.

Un IGBT (Figura 15) (Insulated Gate Bipolar Transistor, Transistor Bipolar de Puerta Aislada), es un tipo de transistor que permite el control por tensión con tiempos de conmutación muy pequeños, del orden de 1 microsegundo. A penas produce caída de tensión cuando está conduciendo y soporta potencias considerables, ya que existen módulos de hasta 6.5kV y pueden llegar a soportar frecuencias de conmutación de entre 2 kHz y 50 kHz. (15)

Transistores IGBT reales

Símbolo del IGBT



IGBT simple



Módulo IGBT

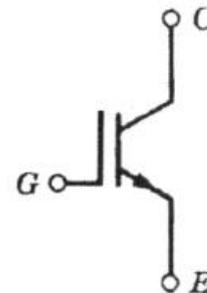


Figura 15 - Imagen de un IGBT simple, un módulo IGBT y el símbolo que representa al IGBT (16) (17)

Dentro de los convertidores VSC existen diferentes tipos de convertidores, a continuación, se muestran algunos ejemplos: (2)

- Convertidor de dos niveles (2-Level Converter) (Figura 16): Es capaz de generar dos niveles de tensión diferentes $+V_{dc}$ y $-V_{dc}$. Es posible utilizar un PWM para que la salida sea senoidal.

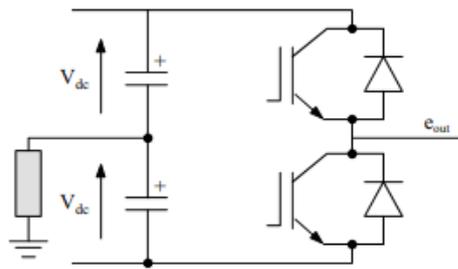


Figura 16 - Esquema de un convertidor de dos niveles (2)

- Convertidor Multinivel restringido por diodo (*Multilevel Diode-Clamped Converter*) (Figura 17): Este tipo de convertidores puede aportar hasta tres niveles de tensión $+V_{dc}$, 0 y $-V_{dc}$. También es posible utilizar PWM para conseguir una onda senoidal.

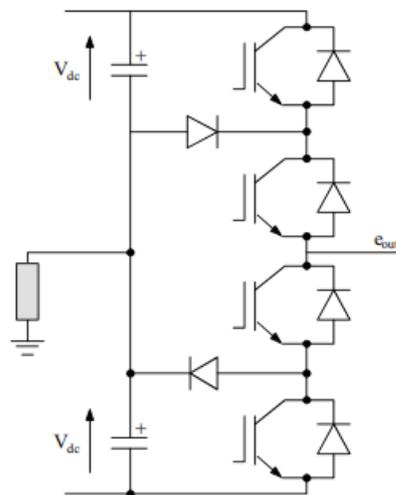


Figura 17 - Esquema de un convertidor multinivel restringido por diodo (2)

- Convertidor Multinivel con Condensador Flotante (*Multilevel Floating Capacitor Converter*) (Figura 18): Este convertidor tiene un comportamiento similar al del anterior, pudiendo generar tres niveles de tensión diferentes ($+V_{dc}$, 0 y $-V_{dc}$) y siendo posible utilizar PWM para obtener una onda senoidal.

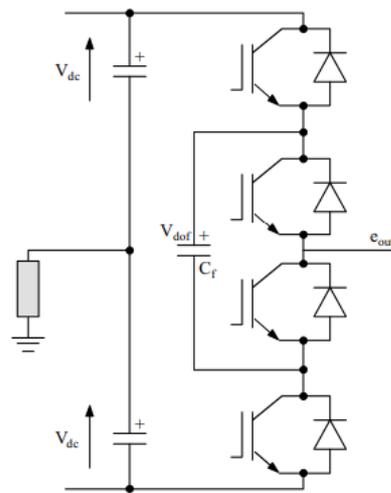


Figura 18 - Esquema de un convertidor multinivel con condensador flotante (2)

- Convertidor Modular Multinivel (*Modular Multilevel Converter*) (Figura 19): Está formado por varios submódulos o celdas. Este convertidor se ve más en detalle en el siguiente apartado.

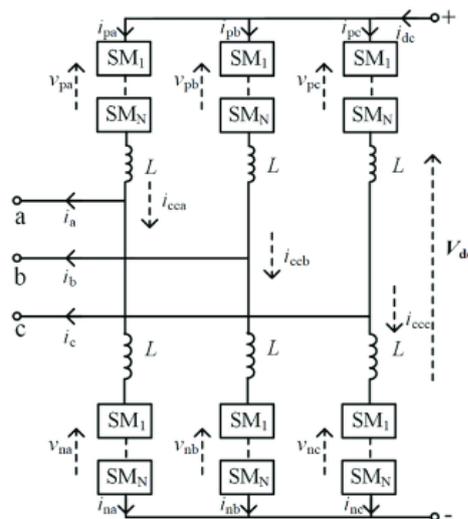


Figura 19 - Esquema de un convertidor modular multinivel (18)

2.8.2.1.- Convertidor MMC

Tal como se cava de ver en el apartado anterior, dentro de los convertidores VSC se encuentra la tecnología MMC (Modular Multilevel Converter, Convertidor Modular Multinivel). Un convertidor MMC está formado por múltiples submódulos que, en función del estado (encendido o apagado) de la señal de encendido de los IGBT (polos de potencia) que lo forman, cada celda puede aportar, de forma independiente, la tensión máxima cuando está encendido o 0 V cuando se encuentre apagado. Por tanto, a mayor número de submódulos por fase, menor será el rizado, la proporción de armónicos y menor será el ruido de alta frecuencia que introduce el convertidor a la red. También se podrá tener

una mayor tensión en corriente continua, ya que, habrá un mayor número de celdas aportando tensión.

En la Figura 19, se puede ver la estructura de un convertidor MMC con n celdas en cada fase, por un lado, se encuentra el polo P, de positivo, en la parte superior y, por otro lado, el polo N, de negativo, en la parte inferior, representando la parte en corriente continua. Entre estos polos, están las celdas correspondientes a las tres fases propias de un sistema trifásico. La conmutación correcta de los polos de potencia de las celdas, proporcionan la salida la corriente alterna.

En un convertidor MMC como el que se muestra en la Figura 19, las celdas de la rama superior de una fase generan una señal que tiene una componente de corriente continua positiva y otra componente de corriente alterna. Igualmente ocurre con las celdas de la rama inferior, pero siendo la componente en corriente continua de igual magnitud, pero de signo negativo. De esta manera, al sumar ambas corrientes la componente en corriente continua se anula y la salida es puramente en corriente alterna.

En todo momento el número de celdas activas entre la rama superior y la rama inferior de una misma fase debe ser igual al número de celdas por rama, tal como se puede ver en la Figura 20 con un supuesto convertidor de cuatro celdas de 800 voltios por rama.

Tensión	Celdas UP ON	Celdas LW ON
3200	4	0
1600	3	1
0	2	2
-1600	1	3
-3200	0	4

Figura 20 - Celdas encendidas en un convertidor MMC de cuatro celdas por rama y 800 V por celda (elaboración propia)

Las conmutaciones en las celdas no ocurren de forma simultánea en la rama superior y en la inferior, ni en las tres fases, por lo que en esos instantes la suma de las celdas activas en una fase puede no ser igual al número de celdas por rama. Esto puede generar problemas como cortocircuitos y sobrecargas. Por este motivo, en la salida de las ramas superior e inferior de cada fase y a la salida del convertidor en el lado de alterna, se colocan unas inductancias para disipar estos problemas transitorios.

2.8.2.1.1.- Tipos de celda: puente y semi puente

Un submódulo de un MMC está compuesto por al menos dos transistores del tipo IGBT (también podría funcionar con GTOs). Los submódulos pueden ser de puente completo o de semi puente:

Una celda de puente completo tiene tres ramas en paralelo, dos ramas con dos IGBT cada una y una rama con el condensador. Cada transistor tiene su propio diodo en contracorriente. Se puede ver en la Figura 21 un esquema eléctrico sobre cómo es una celda de puente completo.

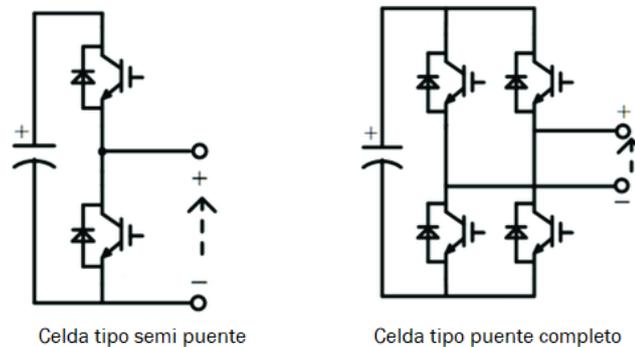


Figura 21 - Celda tipo semi puente y celda tipo puente completo (18)

Una celda del tipo semi puente posee un condensador en paralelo con una rama con dos IGBT, cada transistor con su respectivo diodo en contracorriente. En la imagen de la izquierda de la Figura 21 se puede ver un ejemplo gráfico de una celda del tipo semi puente.

A continuación, se puede ver gráficamente cómo circula la corriente por una celda semi puente en función de si la celda se encuentra insertada (Figura 22) o en bypass (Figura 23) y si la corriente es positiva o negativa.

Cuando la celda está encendida se dice que está insertada, hay dos escenarios:

- La corriente es positiva, la corriente circula por el diodo en contracorriente del polo superior, pasa por el condensador y sale por el otro terminal de la celda.
- Cuando la corriente es negativa la corriente entra a la celda por el terminal negativo, pasa por el condensador, después por el transistor IGBT superior y sale por el terminal positivo de la celda.

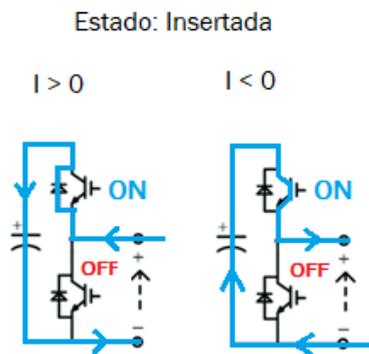


Figura 22 - Circulación de corriente por una celda insertada (elaboración propia)

Cuando la celda está apagada se dice que la celda está en bypass y la corriente tiene los siguientes comportamientos según su signo:

- Cuando la corriente es positiva, la corriente entra por el terminal positivo, circula por el IGBT y sale por el terminal negativo.
- En el caso contrario la corriente circula por el diodo, entra por el terminal negativo y sale por el positivo.

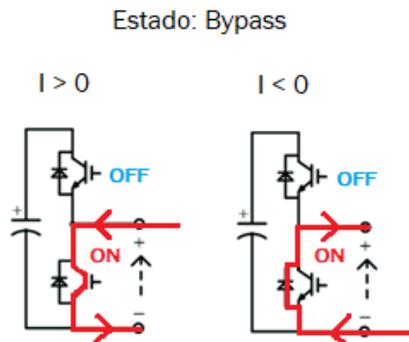


Figura 23 - Circulación de corriente por una celda en bypass (elaboración propia)

2.9.- Modulación por Ancho de Pulso o PWM

El PWM (*Pulse Width Modulation*) Modulación por Ancho de Pulso, es una estrategia de activación de los polos de potencia para llevar a cabo la transformación de corriente continua a corriente alterna. Es la forma más común de modulación. La modulación es una herramienta que sirve para decidir cómo conmutar los semiconductores controlados.

La señal de activación de los polos se genera a partir de la comparación entre dos señales: una señal triangular de alta frecuencia llamada portadora y una señal sinusoidal de la frecuencia de la tensión alterna que se quiere generar o transformar en corriente continua, que recibe el nombre de moduladora. La frecuencia de la portadora será la frecuencia de conmutación de los polos

mientras que la frecuencia de la moduladora será la frecuencia del fundamental de la tensión de salida.

La señal portadora se compara con la moduladora y, en función de si la moduladora se sitúa por encima o por debajo de la portadora, se dictará el estado de los polos de potencia, es decir, se ordenará a las celdas el estado en el que deben encontrarse, encendido o apagado. Existen diferentes tipos de portadoras para convertidores multinivel con PWM. A continuación, se muestran algunos tipos de señal portadora:

2.9.1.- *Phase Disposition*

PD PWM (*Phase Disposition*): en este caso las portadoras tienen forma de diente de sierra y se mantiene la misma fase en cada nivel de tensión, como se muestra en la Figura 24.

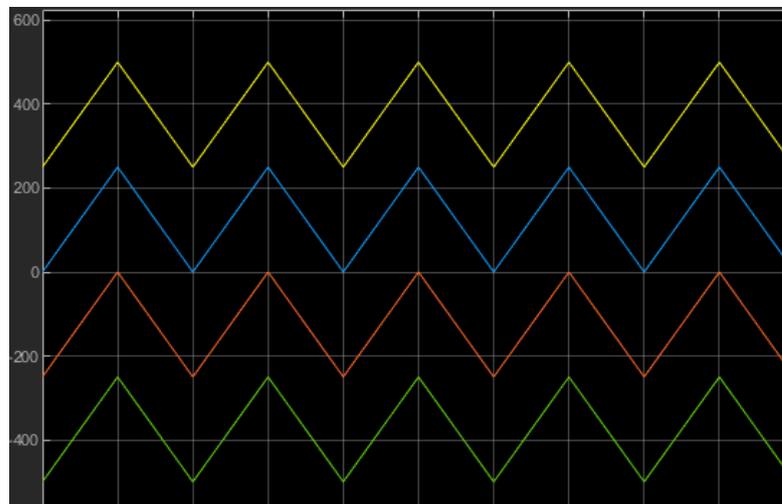


Figura 24 - *Phase Disposition PWM*

2.9.2.- *Phase Opposition Disposition*

POD PWM (*Phase Opposition Disposition*): tiene la misma forma que el PD, pero, en este caso, las portadoras para valores negativos están en contrafase, como se puede ver en la Figura 25.

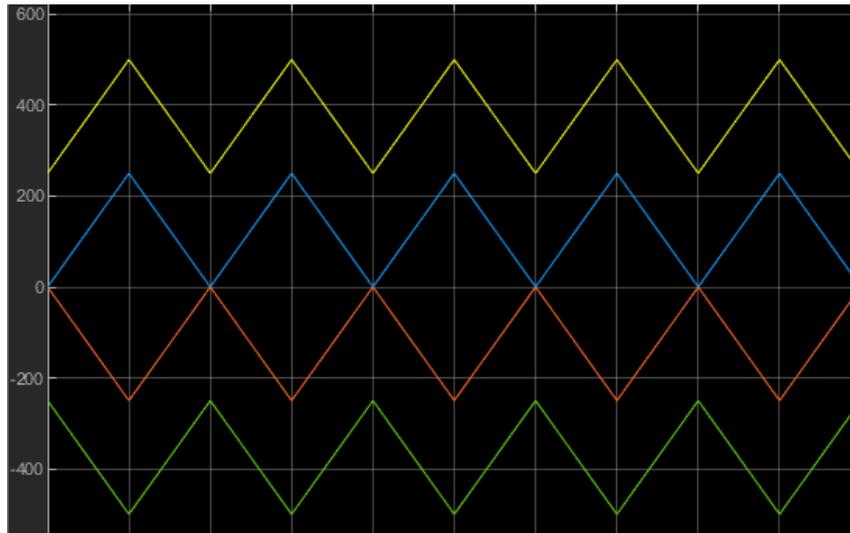


Figura 25 - Phase Opposition Disposition PWM

2.9.3.- *Alternative Phase Opposition and Disposition*

APOD PWM (*Alternative Phase Opposition and Disposition*): en este tipo de portadoras se alterna una portadora en fase y otra en oposición de fase, tal como se ve en la Figura 26.

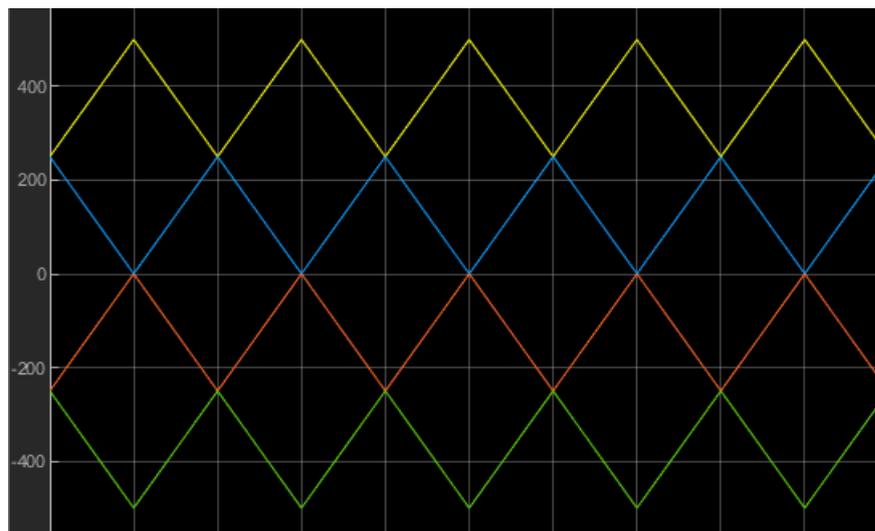


Figura 26 - Alternante Phase Opposition Disposition PWM

2.9.4.- *Phase Shifted*

PS PWM (*Phase Shifted*): tal como se puede ver en la Figura 27, las portadoras tienen la misma forma que en el PD, pero se encuentran todas al mismo nivel de tensión, pero a distinta fase.

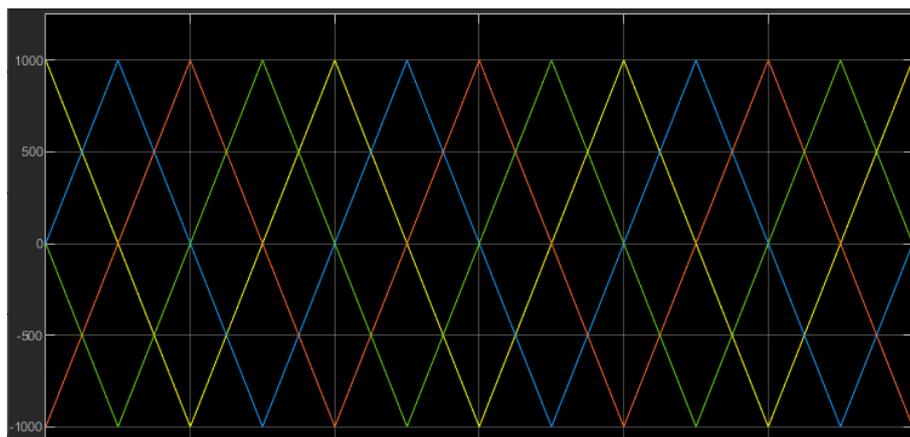


Figura 27 - Phase Shifted PWM

2.9.5.- Carrier Overlapping

CO PWM (*Carrier Overlapping*): en este tipo de portadoras la forma es similar a la del PD, pero, como se observa en la Figura 28, sin llegar a cruzarse, las portadoras del nivel inferior entran en el nivel superior.

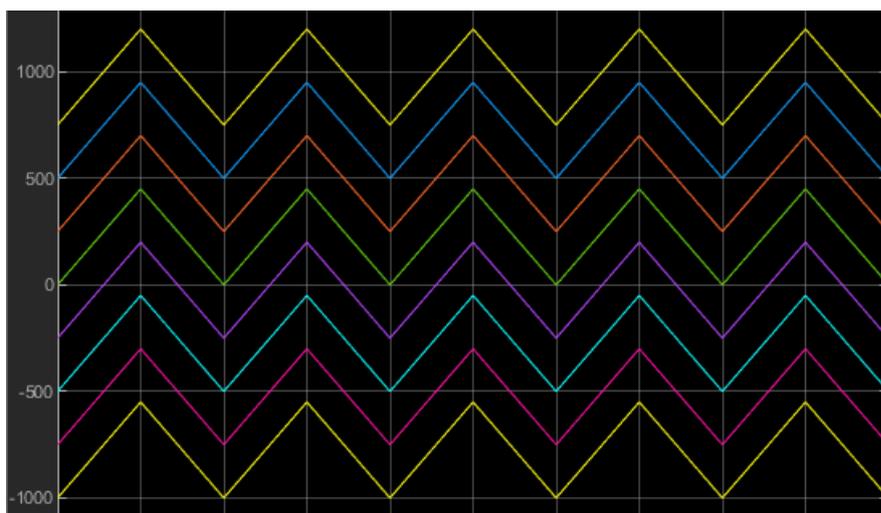


Figura 28 - Carrier Overlapping PWM

2.9.6.- Variable Frequency 2

VF2 PWM (*Variable Frequency 2*): en este caso hay portadoras en ciertos niveles de tensión que tienen una frecuencia diferente a la de otros niveles (Figura 29).

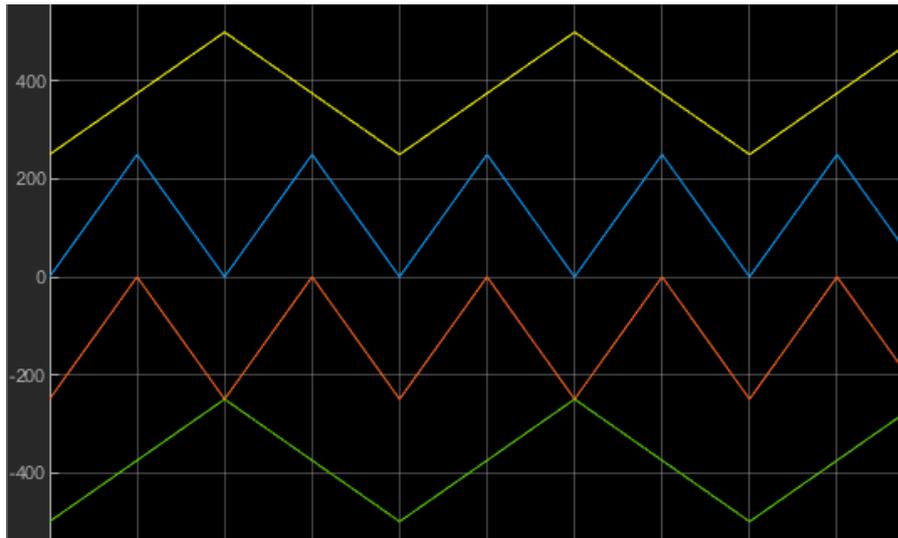


Figura 29 - Variable Frequency 2

2.10.- Apartado sobre Matlab y Simulink:

Matlab es una herramienta de programación y cálculo numérico cuyo lenguaje está basado en matrices. Permite la visualización de datos mediante gráficas. Esta plataforma ofrece multitud de herramientas y aplicaciones diversas que permiten trabajar en diversos ámbitos de la ingeniería. (19) (20)

Una aplicación dentro de Matlab es Simulink, que permite realizar modelos de simulación mediante bloques. En sus librerías existen multitud de bloques para distintas aplicaciones, lo que permite poder diseñar todo el modelo de forma rápida sin la necesidad de tener que escribir el código mediante programación. También existe un bloque que permite implementar funciones embebidas programadas en el lenguaje de Matlab por el usuario. (21)

En este trabajo, para la realización del modelo de convertidor en Simulink, se ha utilizado, dentro de la librería *Simscape*, la carpeta *Specialized Power Systems*, para construir las celdas y montar la planta del convertidor. También se utilizaron varios bloques de funciones embebidas para diseñar el controlador y los generadores de PWM del convertidor. El modelo en Simulink se explica en el capítulo siguiente.

Estos programas se han utilizado en el proceso de creación del convertidor desde cero, ya que las características de esta plataforma permiten este desarrollo inicial:

- El software de MATLAB y Simulink es comercial, está completo y optimizado.
- En Matlab se dispone de un número ilimitado de variables. Se pueden crear tantas variables como se necesite.

- No importa el tiempo de ejecución porque no hay tiempo real. No existe un tiempo límite en el cual se haya tenido que ejecutar todo el programa en cada periodo.

Esto permite desarrollar el código inicialmente sin restricciones de tiempo ni de número de variables para solamente centrarse en conseguir un convertidor que tenga un comportamiento deseado.

2.11.- Apartado sobre simulación en tiempo real:

En este apartado se comentan algunos fundamentos de las simulaciones en tiempo real, después se mencionan algunos de los beneficios que aportan a la industria y, por último, se presenta el entorno de simulación que se ha utilizado para realizar este trabajo.

2.11.1.- Fundamentos

En el ámbito de la ingeniería, la simulación es una técnica mediante la cual, utilizando modelos físicos y/o matemáticos, es posible recrear la actividad de un sistema antes de ser utilizado en el mundo real.

Un sistema en tiempo real implica que no solo importa el resultado de los cálculos de computación, sino que además existen restricciones temporales. Es decir, el resultado de los cálculos debe estar disponible en el momento en el que se necesitan. Un sistema en tiempo real utiliza el mismo tiempo que el del ambiente que controla, es decir, días, horas, minutos, segundos, milisegundos, microsegundos, sea lo que sea, si se pretende controlar en tiempo real algo que varía cada unidad temporal, el sistema en tiempo real deberá evolucionar bajo la misma unidad temporal. La reacción ante perturbaciones externas debe darse dentro de los sucesivos periodos de ejecución del sistema. (22)

No se debe confundir el tiempo real con una ejecución muy rápida de un sistema ni con un sistema que se encuentre en funcionamiento todo el tiempo. Existen sistemas en tiempo real que requieren de un tiempo de ejecución muy rápido y también deben funcionar de forma continua en el tiempo, pero esto depende solamente de la aplicación.

Por lo tanto, no todas las simulaciones funcionan en tiempo real.

Una simulación en tiempo real sería aquella que permite que un modelo de simulación pueda ser ejecutado al mismo ritmo que lo haría el sistema real. Es decir, el modelo simulado debe tener un comportamiento cualitativamente similar al del sistema real al que replica ante las mismas perturbaciones. (23)

“Cualitativamente, el sistema simulado en tiempo real es capaz de generar y recibir las mismas señales de monitoreo y control que el sistema físico real”. (24)

Existen diversas técnicas para realizar simulaciones en tiempo real. En este caso se mencionan las técnicas de Hardware In the Loop (HIL) y Rapid Control Prototyping (RCP), que suelen utilizarse de forma conjunta y son las que se han utilizado para la realización de la simulación correspondiente a este trabajo.

Rapid Control Prototyping (RCP, Prototipaje de Control Rápido) es una metodología de diseño de software de control que permite desarrollar aplicaciones, herramientas o productos de manera acelerada. Los prototipos desarrollados con RCP son más flexibles, rápidos de implementar y es más fácil detectar errores y defectos.

Hardware in the loop (HIL) es una técnica con la cual se pueden validar modelos de sistemas de control simulados en tiempo real. Las señales de control del modelo se conectan a un sistema de pruebas que simula la aplicación real para la que ha sido diseñado el modelo. De esta forma, el sistema de control puede ser probado en multitud de situaciones posibles en la realidad, encontrando los posibles errores y defectos que pueda haber en el modelo antes de ser utilizado en campo. (25) (26)

Algunas ventajas que aporta este método son las siguientes: (24)

- El HIL, permite realizar pruebas a los controladores (RCP) sin necesidad de contar con una planta física.
- Reduce el tiempo de desarrollo y los costes, ya que no es necesaria una planta de prueba ni se deterioran elementos al probarlos al límite.
- Es posible replicar con exactitud diversos escenarios de prueba. Al contrario que con un prototipo real.

2.11.2.- Beneficios

La tecnología de la simulación en tiempo real aporta multitud de ventajas frente a la utilización de modelos reales o simulaciones sin tiempo real. En función del sistema que se quiera simular habrá más o menos ventajas, pero de forma general se pueden enumerar las siguientes:

- Permite realizar pruebas en una planta que no existe o que no está disponible.
- No es necesario realizar un prototipo real para realizar pruebas, por lo que se reducen los costes en el desarrollo al utilizar menos recursos.
- El número de pruebas que se pueden realizar es ilimitado, pudiendo incluso realizar pruebas destructivas o alcanzando los límites recomendados sin dañar ningún elemento. Esto supone un ahorro en costes.

- Reduce el riesgo para las personas en las pruebas destructivas o en el límite. En un prototipo real, este tipo de pruebas conllevan costes debido a la sustitución de aparatos dañados. Además, en un prototipo real, el personal se encuentra sujeto a ciertos riesgos.
- Las distintas pruebas se pueden realizar en un menor periodo de tiempo, permitiendo validar el modelo (HIL) y/o el control (RCP) de la planta rápidamente. Por tanto, también supone un ahorro de tiempo.
- El tiempo de desarrollo se reduce. Es posible el trabajo en paralelo, es decir, mientras la planta no está disponible, es posible desarrollar los algoritmos del controlador mediante RCP o trabajar en el desarrollo del HIL.
- Los errores de diseño pueden ser detectados rápidamente, antes de construir un prototipo.
- El equipo de simulación se amortiza rápidamente, ya que es posible adaptarlo a toda clase de proyectos.
- Permite la simulación bajo unas condiciones inalterables en el tiempo, lo que permite repetir pruebas en las mismas condiciones. En los sistemas físicos no es posible controlar factores como la temperatura o la humedad.

(24)

Muchas de estas ventajas son consecuencia del uso de los métodos HIL y RCP.



Figura 30 – Esquema sobre la composición de un HIL (24)

En la práctica, el uso de HIL consta de tres partes (Figura 30): La estación de trabajo, la planta que se pretende simular y el controlador. (24)

- La llamada estación de trabajo no es más que un ordenador con el que, a través de un software, se puede editar el modelo, visualizar los datos en tiempo real, cambiar parámetros, etcétera.
- La planta que se pretende simular es diseñada desde la estación de trabajo y controlada por el controlador real de la planta. La tarea de controlador puede ser ejecutada por una FPGA o un ASIC.

- El controlador puede estar formado por un RCP.



Figura 31 - Esquema sobre la composición de un RCP (24)

En el caso del RCP, también se constituye de tres partes (Figura 31): La estación de trabajo, el controlador que se pretende diseñar y la planta real o el HIL. (24)

- De la misma manera, la estación de trabajo es un ordenador con un software que permite el desarrollo del controlador.
- En este caso se pretende diseñar el controlador para la planta que se pretende simular. El control y los cálculos necesarios para ello son realizados por una CPU o una FPGA.
- Por último, la planta, que puede ser la planta real o puede utilizarse un HIL.

2.11.3.- *Presentación del entorno de simulación*

El software que se utilizará para la simulación en tiempo real está en desarrollo. Ha sido creado por el Departamento de Tecnología Electrónica de la Universidad de Valladolid.

Con el desarrollo de este trabajo se ha colaborado en la mejora de este software dando feedback de los errores que se han ido encontrando durante su uso. A su vez, el convertidor promediado que se ha desarrollado en el trabajo que se explica más adelante en el apartado 3.3.5.1.- *Explicación del convertidor promediado*, ha sido implementado como nueva opción dentro del bloque del convertidor MMC, lo que permite poder simular convertidores MMC con un mayor número de celdas gracias a que este convertidor utiliza menos recursos que el convertidor realista.

El software es ejecutado por una FPGA (Field Programmable Gate Arrays, matriz de puertas programables), que son dispositivos formados por semiconductores basados en una matriz de bloques lógicos configurables (CLBs) conectados mediante interconexiones programables. Estos dispositivos pueden ser programados una y otra vez para las aplicaciones que se deseen. (27)

Esta FPGA se encarga por un lado del controlador, el RCP, y, por otro lado, de la planta, que corresponde al HIL.

La tarjeta tiene:

- 36 digital outputs.
- 36 digital inputs.
- 24 analog outputs.
- 12 analog inputs.
- Soporta un máximo de 8192 instrucciones.
- Esta tarjeta cuenta con dos cores: el core#1 y el core#2.
- Tiene un máximo de 256 registros de 80 bytes repartidos entre distintos tipos de registro:
 - 96 registros tipo R, correspondientes a las variables escalares globales, privadas de cada core,
 - 32 registros tipo S, que serían variables escalares locales, entre las que se incluyen los argumentos de las funciones.
 - 64 registros tipo K, que marcan el número máximo de constantes que se pueden soportar.
 - 64 registros tipo P, que son multiplicadores, es decir, variables que almacenan el resultado de una multiplicación. Una vez se realiza la multiplicación, se necesita esperar al menos 11 ciclos de reloj para poder utilizar el resultado de la operación.
- Memoria externa de doble puerto con 2048 variables vectoriales, compartida entre los dos cores. Cuando los dos cores definen el mismismo vector, del mismo nombre y tamaño, esa variable vectorial es compartida, y, por tanto, ambos cores pueden leer y escribir sobre dicha variable. Aunque es recomendable que solamente escriba un solo core.
- El core#1 y el core#2 tienen un solo multiplicador cada uno. La memoria compartida tiene 32 multiplicadores.

Estos recursos se han aprovechado en el software de la siguiente manera:

El core#1 se encarga de los cálculos correspondientes a la planta. Se ejecuta cada 4 microsegundos como máximo. Hay un límite con respecto al número de elementos (resistencias, bobinas, condensadores) que puede haber en la planta, que no puede ser mayor de 32.

Los 24 *Analog Output* se utilizan para leer valores de amperímetros y voltímetros de la planta desde el controlador.

El core#2 maneja el controlador, es decir, el bloque RCPcore, que se explica en el siguiente apartado.

El periodo de ejecución del core#2 se indica en el bloque RCPcore en la pestaña configuración. En el caso de la simulación de este trabajo son 200 microsegundos. El bloque RCPcore se muestra también en el siguiente apartado.

Las 36 salidas digitales se utilizan para configurar los generadores de señales de PWM a través de la función pwmConfig, que es una función embebida que ya viene implementada dentro del software, se explicará en el próximo apartado.

Dentro del software del simulador en tiempo real, el compilador presenta una limitación con respecto a trabajar con variables vectoriales, ya que, resulta problemático pasar valores de un vector a otro.

2.11.3.1.- Interfaz de usuario

El software cuenta con un interfaz de usuario sencillo. Al no ser un software conocido, a continuación, se va a hacer una breve presentación del entorno en el que se ha desarrollado la simulación:

- En primer lugar, cuenta con una barra de herramientas básicas (recuadro rojo en la Figura 32), que, como se puede ver de izquierda a derecha, cuenta con: “archivo nuevo”, “abrir archivo”, “guardar”, “deshacer”, “rehacer”, “cortar”, “copiar”, “pegar”, “girar sentido horario”, “espejo” y “girar sentido antihorario”. También cuenta con otras funciones para la conectividad entre la FPGA y el PC (por cable o remota), los botones para compilar, ejecutar/pausar simulación, detener simulación, mostrar scope y exportar datos a Excel.
- Por otro lado, en la barra vertical situada a la izquierda, recuadrada en color azul en la Figura 32, están las librerías. Se puede cambiar de librería pulsando en el menú *libraries*, recuadrado en color amarillo en la Figura 32, seleccionando la librería buscada.
- Por último, en el centro, recuadrado en verde en la Figura 32, se encuentra la zona de trabajo, donde se colocan los distintos elementos de las librerías, para construir la planta de la instalación.

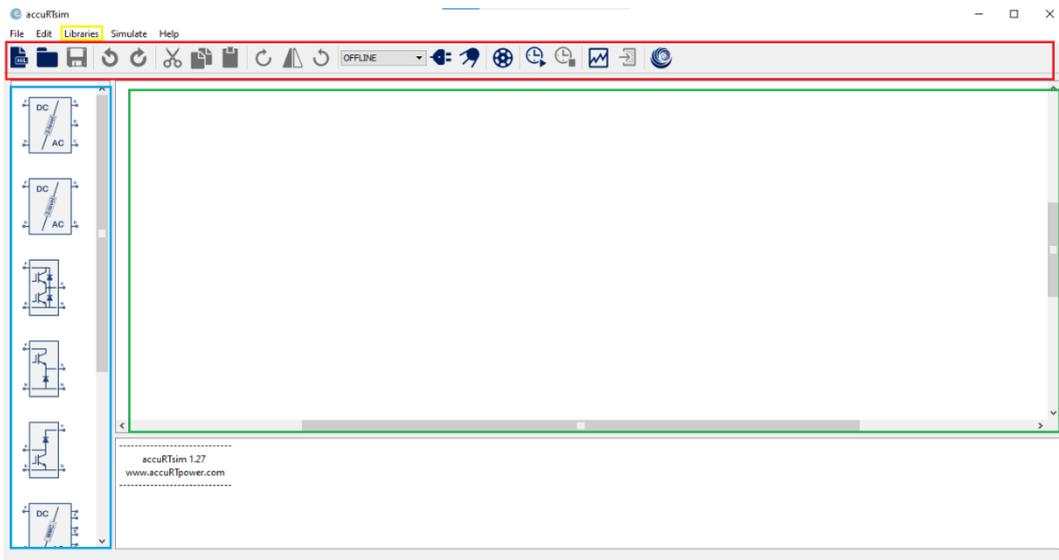


Figura 32 - Interfaz de usuario del software del simulador en tiempo real

2.11.3.2.- Presentación de bloques más relevantes

A continuación, se van a presentar dos de los bloques más importantes para la realización de la simulación que se va a realizar en este trabajo. Estos bloques son el bloque del convertidor MMC y el bloque del RCPcore.

2.11.3.2.1.- Explicación bloque del MMC

El bloque que representa el convertidor MMC realista tiene la apariencia que se muestra en la Figura 33. Como puede observarse, este bloque tiene un terminal P y otro terminal N que corresponden a los polos positivo y negativo del lado de corriente continua del convertidor, y, en el lado de corriente alterna, los seis terminales, dos por cada fase.

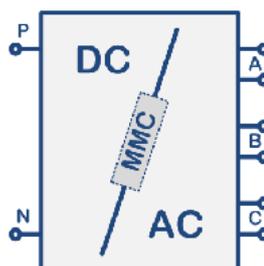


Figura 33 - Bloque de convertidor MMC del software del simulador en tiempo real

Dentro del bloque (Figura 34) se puede configurar:

- *Module tag*: para poner un nombre con el que referenciar las variables compartidas usadas por el controlador para recibir las medidas de tensión de los condensadores del convertidor MMC.

- *Number of serial cells per Branch*: donde se puede definir el número de celdas por rama del convertidor.
- Se puede elegir entre celdas con baterías o celdas con condensadores. En caso de ser condensadores se define la capacidad.
- *Cell voltage*: En el caso de haber elegido baterías, en este punto se configura la tensión nominal de cada celda. En caso de seleccionar condensadores, esta tensión sería la tensión inicial de cada celda.
- También existe la posibilidad de configurar los condensadores como precargados y la opción de añadir fuentes de corriente a las celdas.
- Finalmente, se encuentran la configuración de los inputs, es decir, se asocian los digital inputs a las celdas de cada rama.

Este bloque genera seis vectores de cuatro componentes donde se almacenan las tensiones en los condensadores de cada celda. El nombre de estos vectores debe ser: *<Module tag>_ph(A, B o C)_V(up o lw)*. De la misma manera, también podría obtenerse otros seis vectores con las corrientes. El vector para las corrientes tendría la forma: *Module tag_ph(A, B o C)_I(up o lw)*.

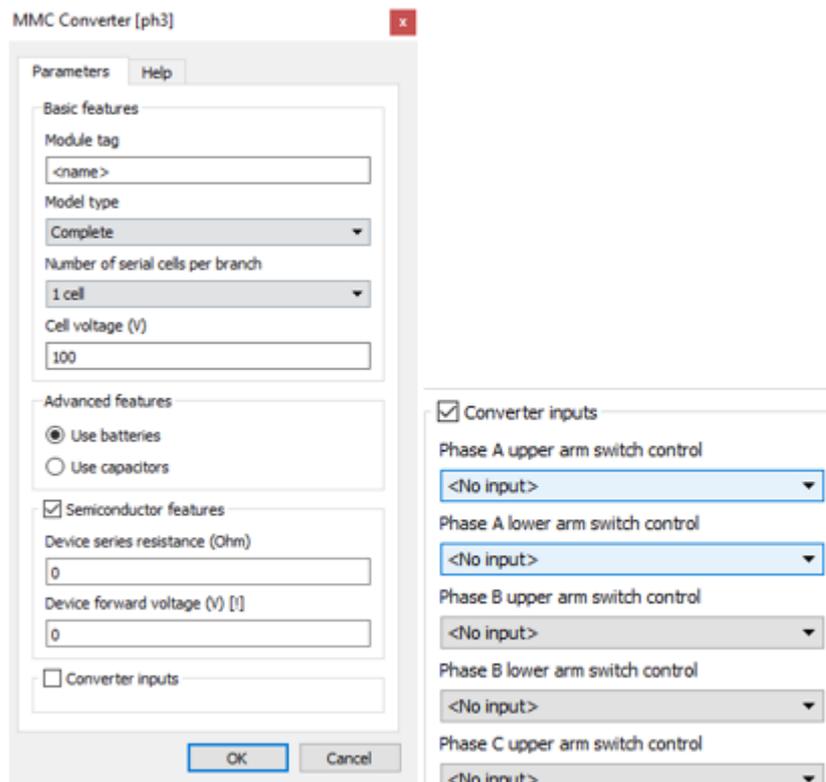


Figura 34 - Menú del bloque del convertidor MMC

2.11.3.2.2.- Explicación bloque RCPcore

El bloque del RCPcore (Figura 35) es el que permite el control de la planta. Dentro de este bloque se escribe el código que controla los parámetros de la instalación.

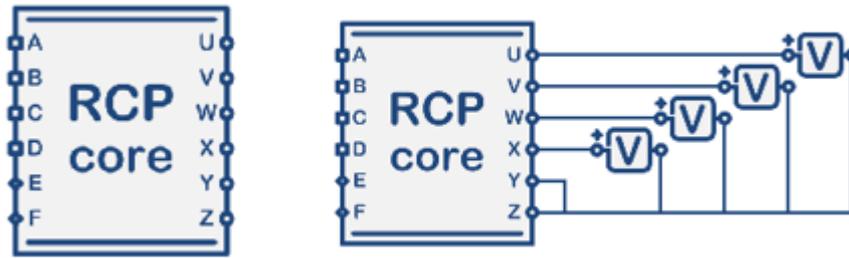


Figura 35 - Bloque RCPcore

Es posible recoger datos de la planta leyendo los voltímetros y amperímetros. Para ello, se les asigna un *Analog Output* o salida analógica, de entre los 24 que se acaba de comentar que hay. Esta salida analógica se puede leer en el controlador con el comando `readAO()`, poniendo entre los paréntesis el número del *Analog Output* asignado al aparato de medida deseado. El valor de lectura se puede asignar a una variable con la que realizar los cálculos requeridos.

Mediante el código se pueden establecer reguladores para que la instalación tenga el comportamiento deseado.

El menú del RCPcore consta de cinco pestañas:

- La primera pestaña “*Config*” (Figura 36) es para la configuración del bloque. En esta parte se puede definir el periodo del *loop* en microsegundos. Se pueden definir los puertos del RCPcore. También se pueden definir las ramas entre las cuales, como se puede observar en la Figura 35, aparecen los medidores de tensión a través de los cuales se puede ver la evolución de la variable que se desee dentro del RCPcore, por medio de los *ports* y *branches*.

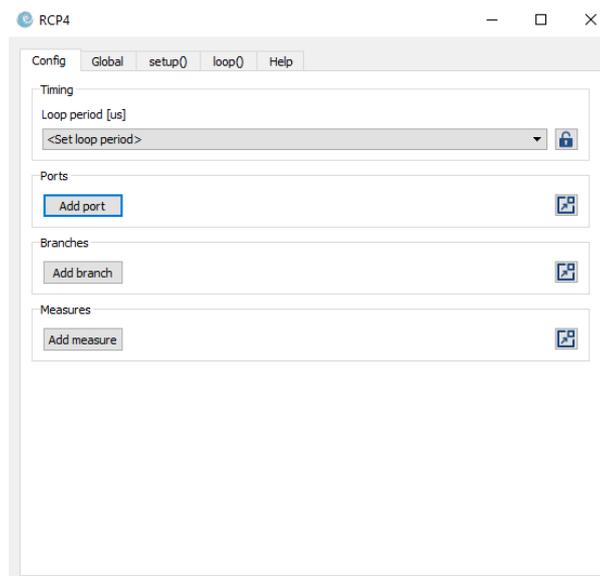


Figura 36 - Pestaña “*Config*” del menú del bloque RCPcore

- Después cuenta con una pestaña “Global” (Figura 37), donde se declaran *#define*, variables globales, tanto escalares como vectoriales, y funciones.

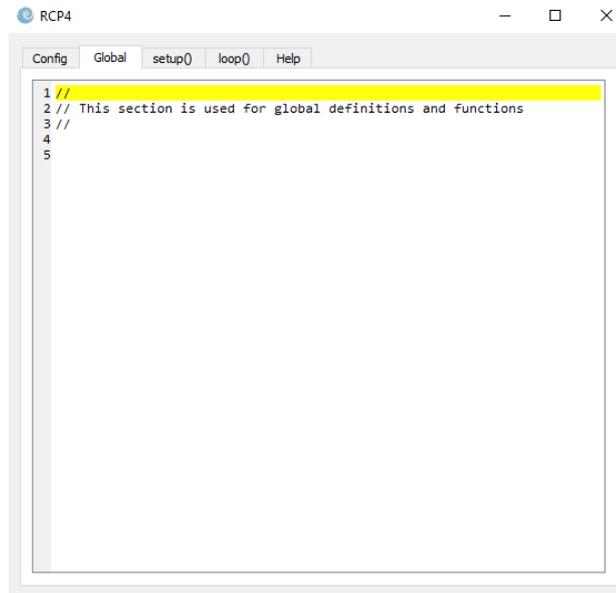


Figura 37 - Pestaña "Global" del menú del bloque RCPcore

- A continuación, se encuentra la pestaña “setup ()” (Figura 38), en la que se inicializan las variables que lo requieran. Esta pestaña solamente se ejecuta una vez al comienzo de la simulación.

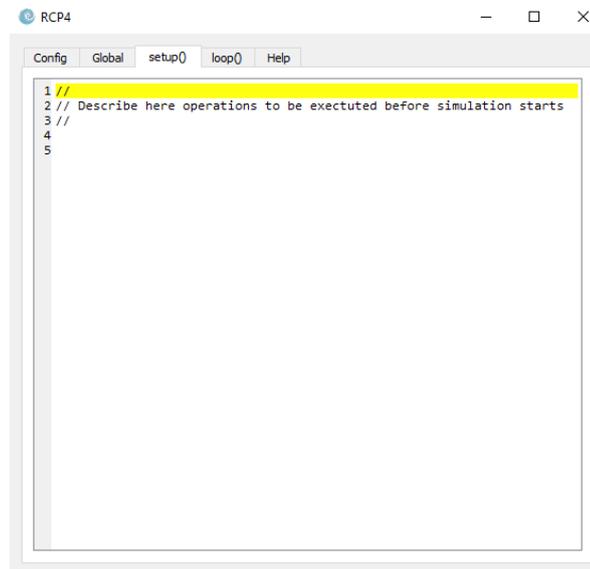


Figura 38 - Pestaña "setup()" del menú del bloque RCPcore

- La siguiente pestaña es la de “loop ()” (Figura 39), dentro de la cual se encuentra todo el bucle que se repite en cada periodo. El tiempo que dura cada periodo se ha definido en la pestaña “Config” en

microsegundos. A su vez se pueden declarar hasta 32 variables locales dentro de “loop ()”, contando los argumentos de las funciones como variables locales. En esta pestaña se encuentra la parte principal del código.

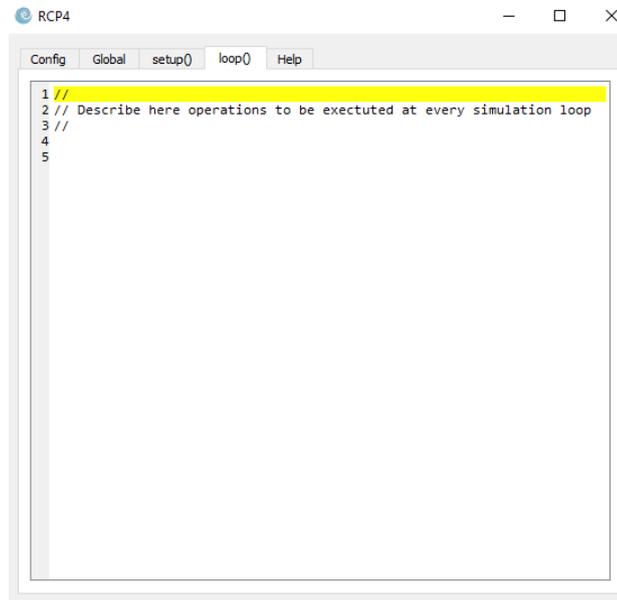


Figura 39 - Pestaña "loop()" del menú del bloque RCPcore

- Finalmente, la pestaña “Help” (Figura 40) muestra una breve explicación sobre el funcionamiento del RCPcore y las funciones embebidas que contiene.

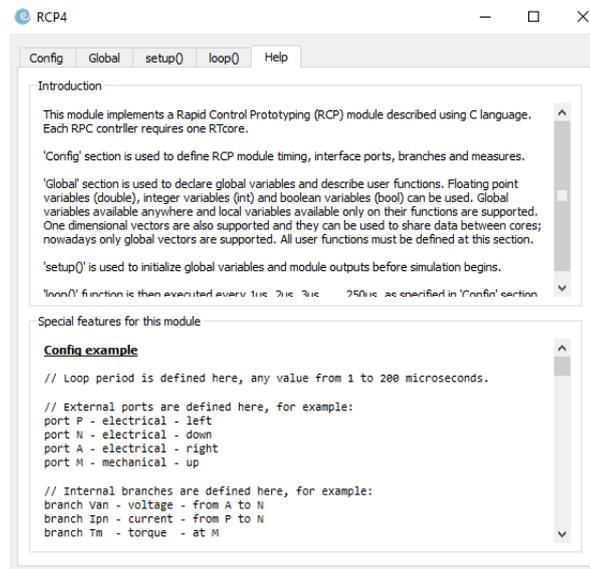


Figura 40 - Pestaña "Help" del menú del bloque RCPcore

2.11.3.3.- Funciones embebidas ya incorporadas en el software

Como se ha mencionado, en la pestaña “Help” aparecen una serie de funciones embebidas que ya vienen incorporadas en el software. Estas funciones han sido programadas en lenguaje ensamblador, con el objetivo de optimizar al máximo el tiempo de ejecución. A continuación, se va a explicar el funcionamiento de alguna de las funciones embebidas más relevantes que han sido utilizadas para la realización de la simulación en tiempo real.

2.11.3.3.1.- *readVector ()*, *writeVector ()*

Las funciones embebidas *readVector ()* y *writeVector ()*, sirven para leer y escribir vectores, respectivamente, de una forma más optimizada. El método utilizado recibe el nombre de lectura/escritura de vectores en modo *burst*.

readVector (nombre_vector, índice_inicial, valor 1, ... valor n)

Esta función está diseñada para leer los valores de un vector y enviarlos a unas variables escalares. No serviría para pasar información de un vector a otro. Tiene como argumentos el nombre del vector que se quiere leer, el índice del vector desde el que se quiere empezar a leer, y las variables donde se quiere almacenar los valores del vector. Esta función lee el vector y va escribiendo los valores en las variables que se le han indicado “*valor 1, ... valor n*”. Para ello toma el índice inicial y en las próximas iteraciones usa el índice más uno (++i).

El objetivo de esta función es reducir el tiempo de lectura de un vector.

Con *readVector ()*, se podrían leer seguidos varios vectores que han sido definidos uno detrás de otro, ya que, los vectores que se han inicializado seguidos se habrán guardado en espacios seguidos en la memoria, pero no hay garantías de que esto vaya a ser así el 100% de las veces, por lo que para reducir el riesgo de que no lea donde debe, se ha utilizado la función una vez para cada vector, sacrificando algo de tiempo, pero asegurando que cogerá el valor deseado.

writeVector (nombre_vector, índice_inicial, valor 1, ... valor n)

Esta función lee los valores de las variables escalares y las escribe en el vector que se le indique en “*nombre_vector*”. El *índice_inicial* es el primer índice del vector donde comienza a escribir los valores. Funciona de la misma manera que *readVector ()*, pero de forma opuesta.

2.11.3.3.2.- *uvw2dq ()* y *dq2uvw ()*

Estas funciones sirven para pasar de un sistema trifásico a un sistema de ejes móviles y viceversa. A continuación, se detalla su funcionamiento.

uvw2dq (X_u, X_v, X_w, sen, cos, X_d, X_q)

Esta función embebida sirve para cambiar de un sistema definido por tres ejes separados 120° entre sí a un sistema definido por dos ejes móviles separados

90°. De esta manera, con tan solo dos magnitudes, el módulo y el ángulo, se tiene totalmente definido el sistema. Los ejes móviles son solidarios a la rotación del módulo de la magnitud que se está observando.

El principio de funcionamiento de esta función embebida es la aplicación de la matriz de Clarke al sistema trifásico, obteniendo un sistema fijo definido por dos componentes (ejes alfa y beta), para, a continuación, utilizar la matriz de rotación para aplicar el giro, resultando los ejes móviles d y q.

Tal como se acaba de comentar, en primer lugar, se aplica la Transformación de Clarke:

Esta transformación es aplicable a sistemas trifásicos preferiblemente sin componente homopolar (equilibrados), es decir, que la suma de las componentes de las tres fases es cero. Para obtener la transformación, se multiplica las tres componentes del sistema trifásico por la matriz de Clarke (Ecuación 1). En la Ecuación 2, Ecuación 3 y Ecuación 4 se muestran los resultados del sistema que aparece en la Ecuación 1.

Ecuación 1 - Transformación de Clarke

$$\begin{bmatrix} x_\alpha \\ x_\beta \\ x_\gamma \end{bmatrix} = \begin{bmatrix} 2/3 & -1/3 & -1/3 \\ 0 & \sqrt{3}/3 & -\sqrt{3}/3 \\ 1/3 & 1/3 & 1/3 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

Ecuación 2 - Resultado Transformación de Park para el eje alfa

$$x_\alpha = \frac{2}{3}x_1 - \frac{1}{3}x_2 - \frac{1}{3}x_3 - \frac{1}{3}x_1 + \frac{1}{3}x_1 = x_1 - \frac{1}{3}(x_1 + x_2 + x_3) = x_1$$

Ecuación 3 - Resultado Transformación de Park para el eje beta

$$x_\beta = \frac{\sqrt{3}}{3}x_2 - \frac{\sqrt{3}}{3}x_3$$

Ecuación 4 - Resultado Transformación de Park para el eje gamma

$$x_\gamma = \frac{1}{3}(x_1 + x_2 + x_3) = 0$$

Cuando el sistema es equilibrado, como se puede ver en la ecuación 2, la componente alfa es igual a la primera fase. De la misma manera, cuando el sistema es equilibrado, la ecuación 4 resulta ser cero.

Se obtienen dos ejes fijos con unas componentes alfa y beta separadas 90°:

$$x_\alpha = x_1$$

$$x_\beta \approx x_\alpha - 90^\circ$$

$$x_\gamma = 0$$

De esta forma con el módulo y el ángulo se tendría completamente definido la tensión o la corriente a la que se le haya aplicado esta transformación.

Una vez obtenidos estos ejes alfa y beta, se aplica la matriz de rotación (Ecuación 5) para añadir el giro:

Ecuación 5 - Matriz de rotación

$$\begin{bmatrix} x_d \\ x_q \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} x_\alpha \\ x_\beta \end{bmatrix}$$

De esta manera, los ejes serían solidarios al giro del componente que representan, siendo la componente en X_d equivalente al módulo y la componente en X_q un valor muy cercano a cero.

Como se puede ver, la función $uvw2dq$ () tiene como entradas las tres magnitudes del sistema de tres ejes y el seno y el coseno del ángulo con el que se define el sistema con los ejes d y q. Como salidas aporta las componentes d y q.

Es posible aplicar la Transformada de Clarke y la matriz de rotación en un solo paso mediante el uso de la Transformada de Park.

$dq2uvw$ (X_d , X_q , sen , cos , X_u , X_v , X_w)

La utilidad de esta función embebida es realizar el cambio contrario a la función $uvw2dq$ (). Partiendo de las componentes d y q de los ejes móviles separados 90° , se consigue llegar a un sistema definido por tres ejes separados 120° entre sí.

Para aplicar este cambio en sentido contrario, se utilizan las matrices inversas de las que se han utilizado para realizar el cambio en el sentido opuesto:

Para pasar de ejes alfa y beta a ejes abc, se utiliza la matriz inversa de Clarke (Ecuación 6):

Ecuación 6 - Transformación inversa de Clarke

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 \\ -1/2 & \sqrt{3}/2 & 1 \\ -1/2 & -\sqrt{3}/2 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_\alpha \\ x_\beta \\ x_\gamma \end{bmatrix}$$

Lo mismo con la matriz de rotación que se utilizó para pasar a ejes móviles, se utiliza la matriz inversa para volver a ejes fijos.

Esta función tiene como entradas las componentes d y q y el seno y el coseno del ángulo con el que se define el sistema con los ejes d y q. Tiene como salidas las tres componentes del sistema trifásico.

En este software también vienen incorporadas las funciones intermedias:

- uvw2ab () para pasar de ejes a, b, c a alfa y beta.
- ab2dq () para cambiar de ejes alfa y beta a ejes móviles d y q.
- dq2ab () para realizar el cambio opuesto de d y q a ejes alfa y beta.
- ab2uvw () para volver de ejes alfa y beta a tres ejes a, b, c.

2.11.3.3.3.- *pwmConfig ()*

Esta función se encarga de configurar los generadores de PWM del simulador en tiempo real utilizando las salidas digitales o *Digital Output*.

La función *pwmConfig ()* tiene cinco argumentos:

- Canal *pwm*, que es el *Digital Output* donde se va a enviar la configuración del PWM. Es un número que debe estar entre el 1 y el 36, ya que es el número de *Digital Output* que hay.
- El periodo en microsegundos del PWM, es decir, el tiempo que tarda en describir la rampa de subida (o de bajada) y la rampa de bajada (o de subida). Este tiempo debe estar comprendido entre 1 y 1000 microsegundos.
- 0 o 1, para indicar la forma de la portadora ($\sphericalangle\sphericalangle$ o $\sphericalangle\backslash$).
- Por último, el ciclo de servicio, que es un valor perteneciente al intervalo [0.0 - 1.0], es el porcentaje del periodo en el que el polo de potencia correspondiente va a estar encendido.

Las configuraciones de PWM se envían a los generadores de PWM mediante los *Digital Output* correspondientes. A través de los *Digital Inputs*, el bloque del convertidor MMC puede recibir las órdenes correspondientes para cambiar el estado de los polos.

2.11.4.- *Limitaciones*

Como se ha podido ver en el apartado 2.11.3.-*Presentación del entorno de simulación*, los recursos para la realización de la simulación en tiempo real dependen de factores como:

- Las características de la FPGA, ya que, la tarjeta utilizada en este caso limita los recursos que se pueden utilizar a los mencionados en el apartado "*Presentación del entorno de simulación*", como por ejemplo el número máximo de variables que se pueden utilizar o el número de constantes. A su vez, la FPGA utilizada tiene una velocidad de procesamiento de datos determinada. Existen FPGA con mayor cantidad de recursos y mayor velocidad de procesamiento, a costa de un mayor precio.

- El periodo de ejecución del *loop*, debido a que, cuanto menor sea este periodo, habrá menos tiempo disponible para ejecutar el programa del *loop*.
- El tiempo de ejecución de la planta, que no debe superar los 4 microsegundos, ya que para tiempos mayores los errores de la planta empiezan a ser importantes. Es otra de las limitaciones que hay a la hora de utilizar el software de simulación en tiempo real.

Además, por otro lado, al no poder realizar todas las medidas de forma simultánea, se deben realizar lo más próximas entre sí, ya que, en tiempo real, en un mismo periodo, podría llegar a haber un desfase considerable entre unas medidas y otras, distorsionando los cálculos.

También se debe tener en cuenta el fenómeno del “*jitter*” a la hora de generar las portadoras del PWM. Este efecto se produce al no poderse generar todas de forma simultánea, se produce un desfase entre las primeras portadoras que se generan y las últimas.

La FPGA del simulador en tiempo real es muy rápida haciendo operaciones de suma y multiplicaciones. Es mucho más lenta para dividir, hacer raíces, senos, cosenos. Pero, sobre todo, es más lenta para tomar decisiones, como, por ejemplo, con las condiciones *if* o los bucles *for* o bucles *while*.

En Matlab no se ha tenido en cuenta del todo esto porque realmente no importaba el tiempo ni hay límite de variables. Con el simulador en tiempo real, hay que tener presente siempre la manera de gestionar el tiempo de ejecución, el número de variables, el número de instrucciones, el número de constantes, el número de elementos (resistencias, condensadores y bobinas) que hay en la planta, el número de amperímetros y voltímetros de los que puede leer el controlador (número de *Analog Output*).

3.- CAPÍTULO 3: DESARROLLO

3.1.- Presentación de la línea que se pretende simular

En este apartado se va a presentar la línea objetivo que se pretende simular en tiempo real.

Tal como se observa en el esquema de la Figura 41, por un lado, hay un primer convertidor conectado a una red trifásica de corriente alterna a 50 Hz y 850 V de tensión simple (en valor eficaz). A continuación, está la línea para transferir la potencia en corriente continua de una estación transformadora a otra. Por último, se encuentra el otro convertidor, conectado a otra red trifásica de corriente alterna a 50 Hz y 850V de tensión simple. En definitiva, la línea objetivo es un HVDC punto a punto que conecta dos redes de igual frecuencia o dos puntos de una misma red distantes entre sí.

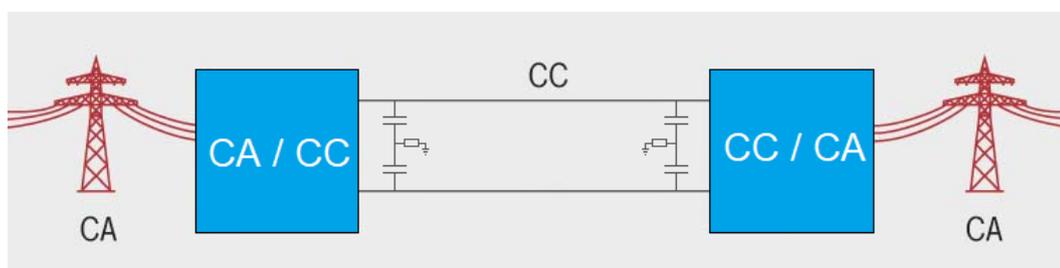


Figura 41 - Línea objetivo de la simulación (elaboración propia)

Los convertidores cuentan con cuatro celdas por cada rama, que hacen un total de 24 celdas en cada convertidor. Las celdas son de tipo semi puente, con un condensador y dos transistores IGBT. Los condensadores de las celdas son de 30 mF. La tensión en el lado de corriente continua es de 3200 V, por tanto, en cada celda hay unos 800 V. Al haber cuatro celdas en cada rama, en todo momento debe haber 4 celdas encendidas por fase.

Este convertidor de tan solo 24 celdas y una tensión en continua de 3200 voltios no se acerca lo más mínimo a las magnitudes de un convertidor MMC real para el uso en una línea HVDC. Como se ha visto en el apartado [2.3.- HVDC en la actualidad](#), las tensiones de un HVDC son del orden de cientos de kilovoltios. Por lo tanto, si los IGBT pueden llegar a soportar como máximo unas tensiones de 6.5 kV en el mejor de los casos, para llegar a un nivel de tensión de un HVDC real harían falta decenas de celdas. También habría que tener en cuenta que los IGBT de 6.5 kV serían más caros que otros de menor tensión nominal, por lo que probablemente saldría más económica una opción con mayor número de celdas, pero de menor tensión cada una.

El motivo por el que el convertidor de la simulación solamente tiene cuatro celdas por rama es debido a que los recursos disponibles en el simulador son limitados. Por un lado, un mayor número de celdas conlleva un mayor número

de cálculos que implican mayor tiempo de ejecución, este tiempo no puede superar los 4 microsegundos, que es el periodo de ejecución de la planta. Por otro lado, el número de *Digital Output* es de 36, las salidas digitales se utilizan para configurar los generadores de PWM, por lo que un convertidor de 24 celdas necesitará 24 *Digital Output* para funcionar. Por lo tanto, se ha decidido diseñar el convertidor de 24 celdas para poder simular un comportamiento realista con los recursos disponibles.

El número de celdas y la tensión nominal de las celdas determinan las tensiones en corriente continua y en corriente alterna del convertidor. La tensión nominal de las celdas viene determinada por la tensión nominal de los IGBT, teniendo en cuenta su punto de rotura. Al quedar la tensión en 3200 voltios en continua y en 1200 en alterna y transferir una cantidad de potencia significativa, las corrientes serán más elevadas de lo deseado.

Al hacer el convertidor de 24 celdas, no sería posible realizar la simulación porque son necesarios dos convertidores, por lo tanto, surgen dos problemas:

- El tiempo de ejecución de la planta de 4 microsegundos se vería superado.
- No habría suficientes *Digital Output* para generar todas las señales PWM necesarias.

Entonces, para poder simular la línea HVDC con los dos convertidores se ha optado por utilizar un convertidor promediado que utiliza menos recursos. Este convertidor obtiene un promedio de las tensiones de las celdas insertadas. El convertidor promediado se explica más a fondo, en el apartado [3.3.5.1.- Explicación del Convertidor Promediado](#).

3.1.1.- Valores de interés de los convertidores

Inicialmente se han utilizado unos valores de bobinas (ver Figura 42) y condensadores elegidos de manera arbitraria, tratando de garantizar la estabilidad de la simulación. Con estos valores se ha conseguido que el convertidor funcione correctamente:

- Las bobinas de las ramas (L_{arm}) de 1.36 mH y la bobina L_0 de 2.72 mH.
- Los condensadores de las celdas de 30 mF y los condensadores de la línea en el lado de corriente continua de 1 mF.
- En las pruebas en las que se ha conectado los convertidores a una carga resistiva, cada fase tenía una resistencia de 13 Ω . El resto de las resistencias son de 1 M Ω y su función es para evitar que la corriente divida en esas bifurcaciones, amortiguar transitorios y/o poder colocar las bobinas a la salida en el simulador.

- Por otro lado, las celdas del convertidor y las bobinas tienen una resistencia interna de 0.1Ω .

Después de conseguir simular la línea HVDC correctamente con estos valores, se han reducido algunos de ellos con el fin de obtener una respuesta de los convertidores más rápida y amortiguada. Por otro lado, cuanto menor tamaño tengan estos elementos, menor será su coste. Aun así continua siendo un modelo conservador donde prima más la estabilidad que el realismo. Por lo tanto, los valores definitivos son los siguientes:

- Las bobinas de las ramas pasan a ser de $500 \mu\text{H}$, mientras que las bobinas L_0 pasa a valer $1000 \mu\text{H}$.
- Los condensadores de las celdas se mantienen en 30 mF . Los condensadores de la línea en corriente continua también se quedan en 1 mF .
- En cuanto a las resistencias internas en el convertidor quedan eliminadas para tener un mejor comportamiento y en cuanto a las de las bobinas, se reducen hasta los 0.05Ω en las de las ramas y hasta los 0.01Ω en las L_0 .

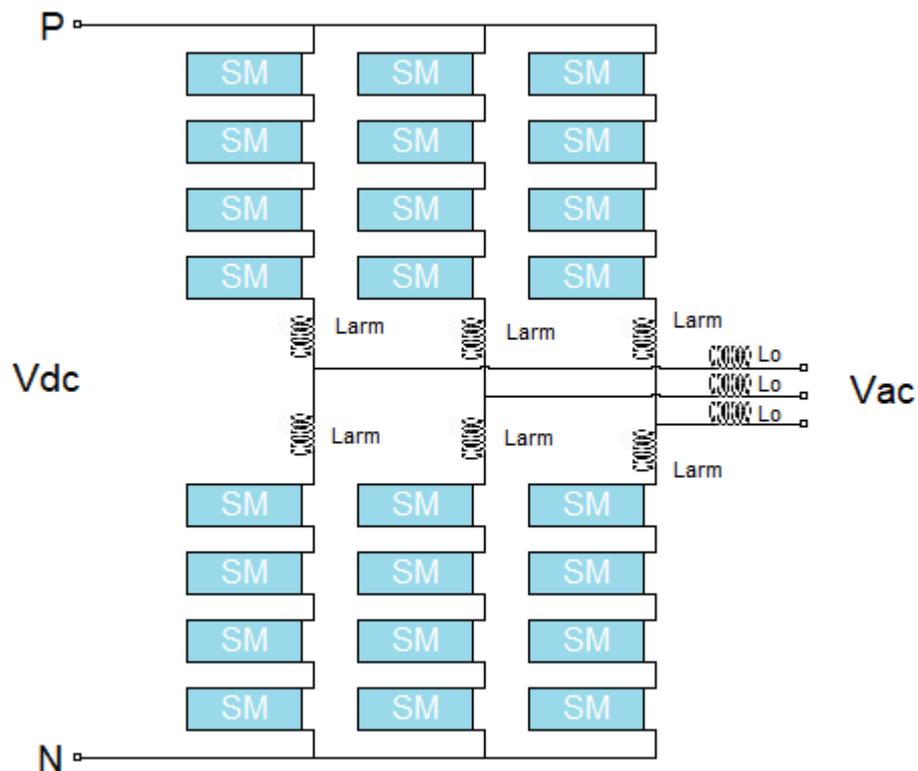


Figura 42 - Esquema del convertidor de la simulación (elaboración propia)

3.2.- Convertidor en Matlab/Simulink

Inicialmente, partiendo desde cero, se ha utilizado la herramienta Simulink para realizar un modelo del convertidor MMC a partir del cual, van a estar basados los convertidores de la línea HVDC que va a ser simulada.

Como ya se ha comentado, programar el modelo en Simulink tiene la ventaja de no tener límite en el número de variables ni en el tiempo de ejecución del código, lo que permite crear un convertidor funcional sin sufrir las limitaciones que si tiene la simulación en tiempo real.

Al no simular en tiempo real, se puede observar el comportamiento del convertidor en los primeros milisegundos de simulación. El objetivo de este modelo es alcanzar unas señales estables y que converjan, ya que, en el simulador en tiempo real, cualquier inestabilidad es mucho más sensible y no es posible observar ciertos comportamientos de las ondas en los primeros milisegundos de simulación, y, por tanto, es más complicado observar el origen de ciertos errores en el modelo.

El desarrollo del convertidor en Simulink debe estar sujeto a las restricciones técnicas que existen en el simulador en tiempo real, por lo tanto, el convertidor también debe tener 24 celdas en total, cuatro celdas por rama. No tendría sentido diseñar un convertidor que no fuera posible simular después en tiempo real con los recursos disponibles.

3.2.1.- Esquema del convertidor en Simulink

En este apartado se presenta la planta del modelo del convertidor creado a partir de Simulink. Al ser un esquema demasiado grande se ha fragmentado en varias figuras para poder observarlo en detalle. En primer lugar, en la Figura 43, se puede ver la estructura del convertidor propiamente. En la Figura 44 se observa el subsistema donde se encuentra la función embebida *controller*. A continuación, se encuentra la Figura 45 en la que se puede ver el interior del subsistema de la Figura 44, con la función *controller*. Después se muestra en la Figura 47 uno de los subsistemas en los que se encuentran las cuatro celdas de una rama y el subsistema donde se encuentra la función *PWM_Generator*. En la siguiente imagen, en la Figura 48 se ve el interior de este subsistema. Por último, en la Figura 49 y Figura 50, se puede observar el diagrama para observar las corrientes circulantes.

Como ya se ha mencionado, la Figura 43 corresponde a la planta del convertidor. La planta consta de dos baterías de 1600 V cada una en el lado de corriente continua, entre las dos baterías hay una conexión a tierra en serie con una resistencia de un mega ohmio.

Hay seis bloques de subsistema en los que, en cada uno de los cuales, se encuentran las cuatro celdas correspondientes a cada rama y el subsistema

dentro del cual está la función *PWM_Generator*. A la salida de estos bloques hay una bobina, que recibe el nombre de L_{arm} . En cada fase, la suma entre la corriente de la rama superior y la corriente de la rama inferior circula por la bobina de salida, llamada L_o . Siguiendo el camino que describiría la corriente en su camino hacia la carga resistiva, como se puede ver en el esquema, se encontraría con una bobina con resistencia en serie y unos condensadores en paralelo, cuyo objetivo es amortiguar transitorios y filtrar el rizado de la corriente, respectivamente.

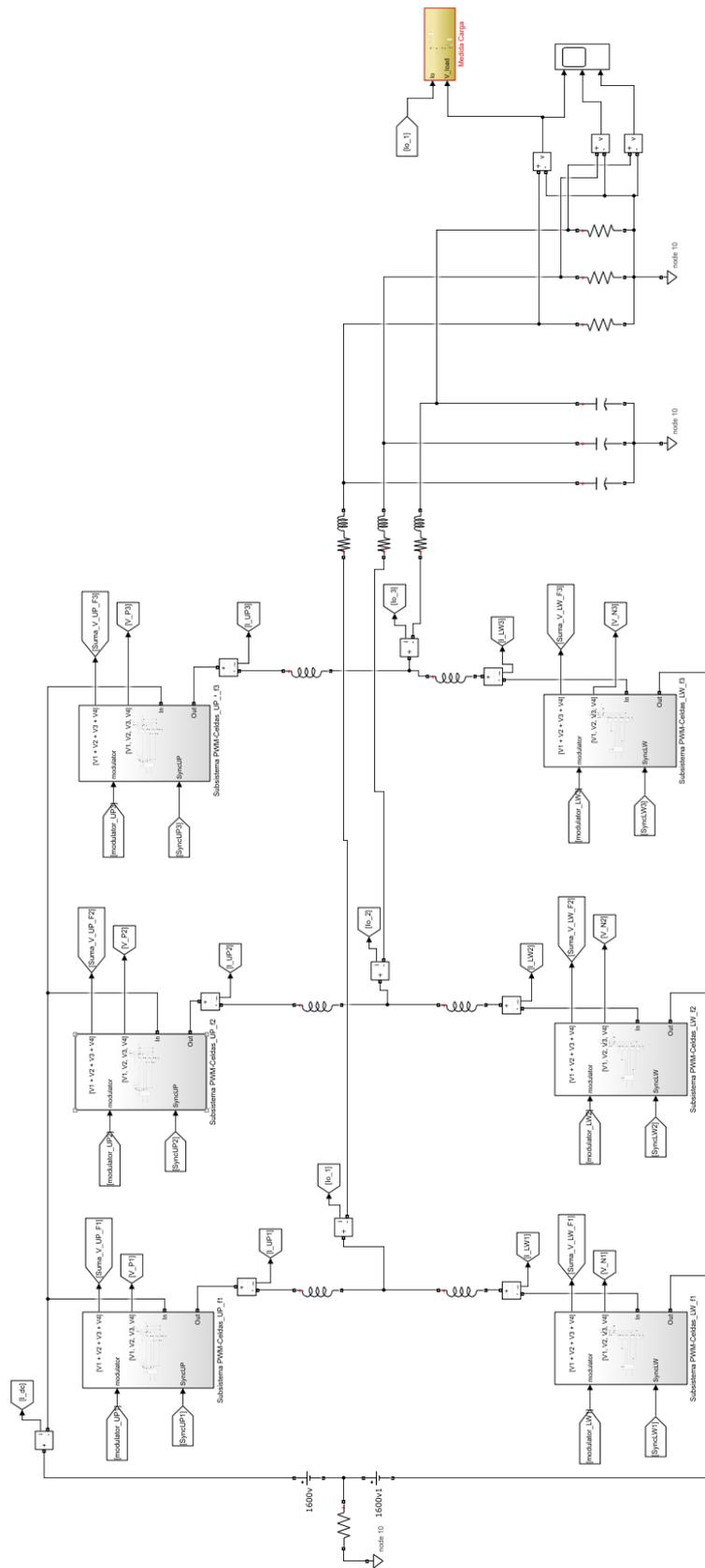


Figura 43 - Planta del convertidor MMC en Simulink

En la Figura 44, hay un bloque del tipo subsistema dentro del cual se encuentra la función embebida *controller*. También se puede observar que hay dos bloques *mux* en los que las corrientes de cada fase de la rama superior e inferior, respectivamente, se agrupan en dos vectores, uno para las corrientes de las ramas superiores y otro para las ramas inferiores.

El bloque del subsistema tiene como entradas las tres tensiones de las tres fases de las ramas superiores, las de las ramas inferiores y los dos vectores que contienen las corrientes de cada rama.

Como salidas tiene las seis señales moduladoras, una para cada rama y dos vectores que contienen las señales de sincronización, uno para las ramas superiores y otro para las inferiores. Los vectores se descomponen en tres componentes escalares con el bloque *demux*. El bloque nombrado como “250 us”, genera un pulso cada 250 microsegundos, que es el periodo con el que se ejecuta el código del controlador.

El bloque en el que aparece escrito “Discrete”, es un bloque *powergui*, el cual permite ejecutar la planta de forma discretizada en intervalos de tiempo fijos, en este caso de 5e-07 segundos.

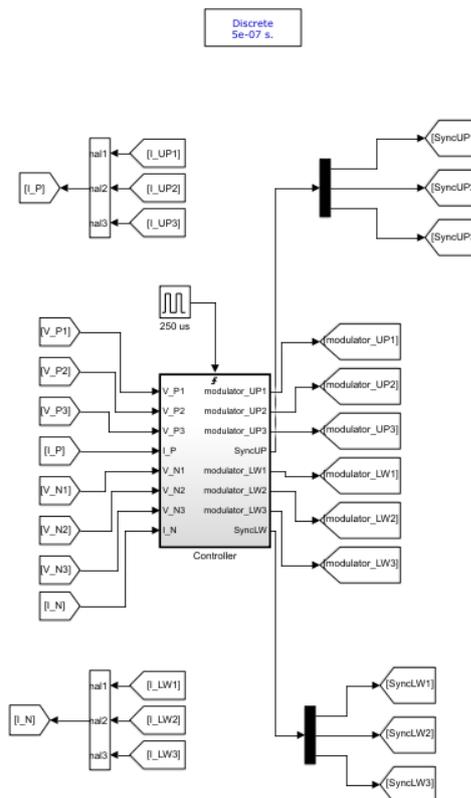


Figura 44 - Bloque subsistema del controlador en Simulink

Dentro del subsistema, se puede ver la función embebida *controller*. La función contiene las mismas entradas y salidas que se han mencionado para el subsistema de la Figura 45, por lo que no se van a volver a nombrar. Cabe destacar la presencia del bloque *Trigger*, cuya función sirve para que el programa del controlador se ejecute en el periodo establecido, cada vez que recibe el pulso del bloque “250 us” ejecuta el programa.

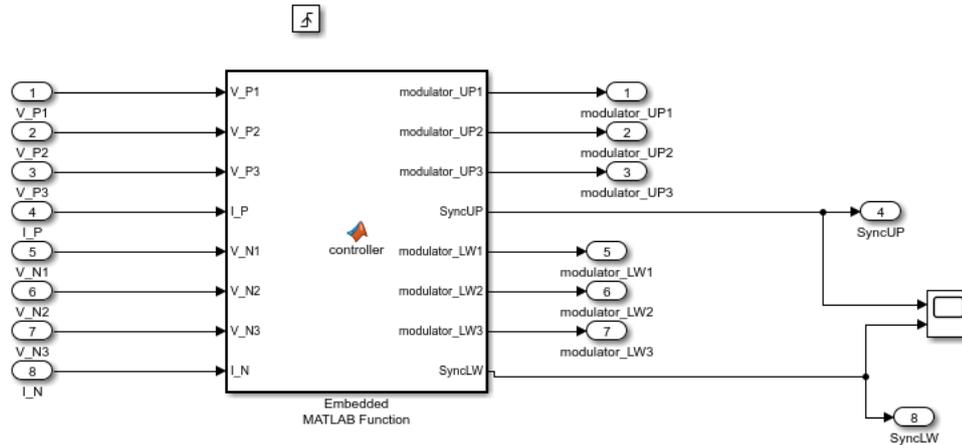


Figura 45 - Bloque función embebida "controller"

Entrando en uno de los seis subsistemas de las ramas el convertidor (Figura 47), se pueden observar las cuatro celdas. Estas celdas, cuyo interior se puede ver en la Figura 46, son otro bloque subsistema que contiene el condensador y los dos IGBT. La corriente circula entrando por la entrada A y sale por la salida B. Por la salida *Out* de estos subsistemas se obtiene la tensión de los condensadores.

Por otro lado, está el bloque *PWM_generator*. Este bloque recibe la moduladora y la señal de sincronización correspondiente. Hay una moduladora y una señal de sincronización para cada rama del convertidor. También, recibe una señal de pulso en cada microsegundo. Como salida obtiene el estado en el que deben encontrarse los polos de potencia y se lo comunica directamente a las celdas.

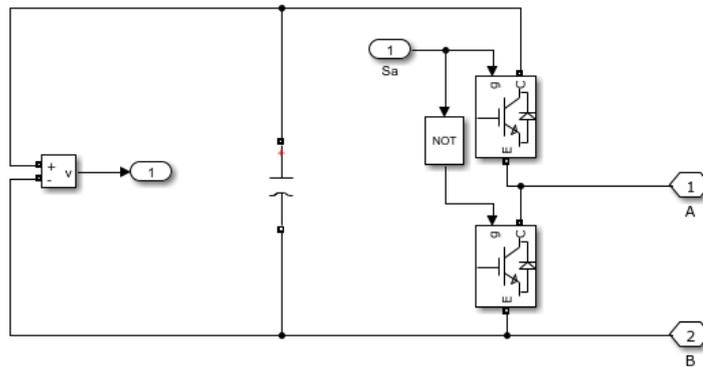


Figura 46 - Interior de una celda del convertidor de Simulink

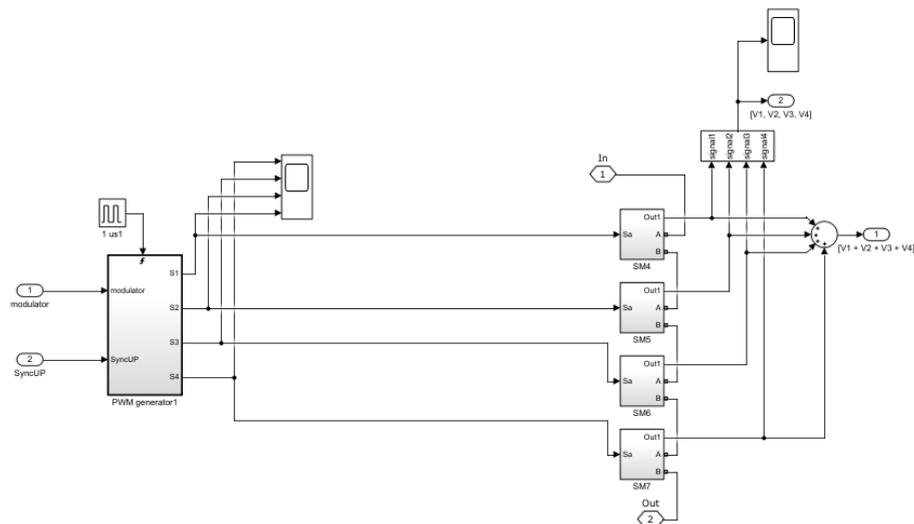


Figura 47 - Bloque subsistema que contiene las celdas de una rama del convertidor MMC en Simulink

En el interior de cada subsistema de la función de PWM (Figura 48), se encuentran las mismas entradas y salidas que se han nombrado anteriormente. Este subsistema también contiene el bloque *Trigger*, para ordenar la ejecución del código de la función cada vez que hay un pulso, en este caso, en cada microsegundo.

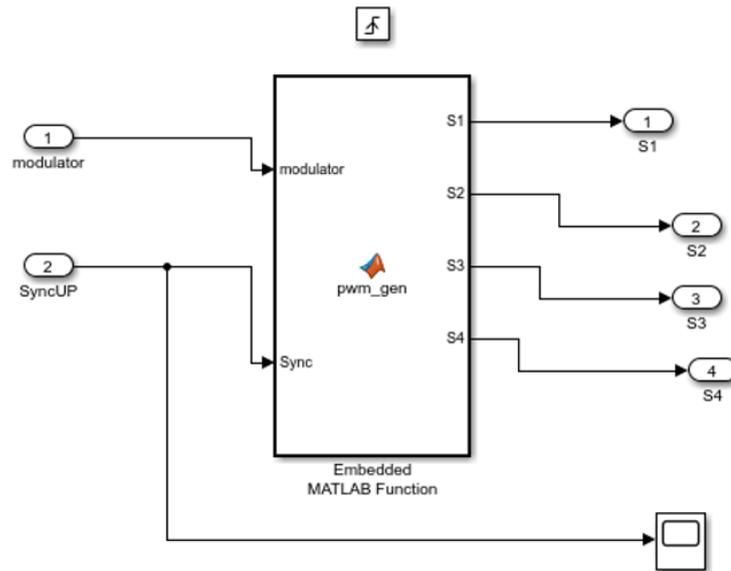


Figura 48 - Bloque función embebida "pwm_gen"

En la Figura 49 se observa cómo se obtienen las gráficas de las corrientes y tensiones. Para las corrientes, se ha creado un subsistema para cada fase. A este subsistema llegan como entradas la corriente de la rama superior, la corriente de la rama inferior, la corriente de salida de la fase correspondiente y la corriente en el lado de continua. Con las corrientes de las ramas superior e inferior se calcula la corriente circulante de cada fase. Como se puede ver en la Figura 50, la corriente circulante es el resultado de la suma entre las corrientes de la rama superior e inferior dividido entre dos.

En cuanto a las tensiones, en cada fase, la diferencia entre la suma de las tensiones de las celdas de la rama superior y la suma de las tensiones de las celdas de la rama inferior resulta la tensión de salida del convertidor.

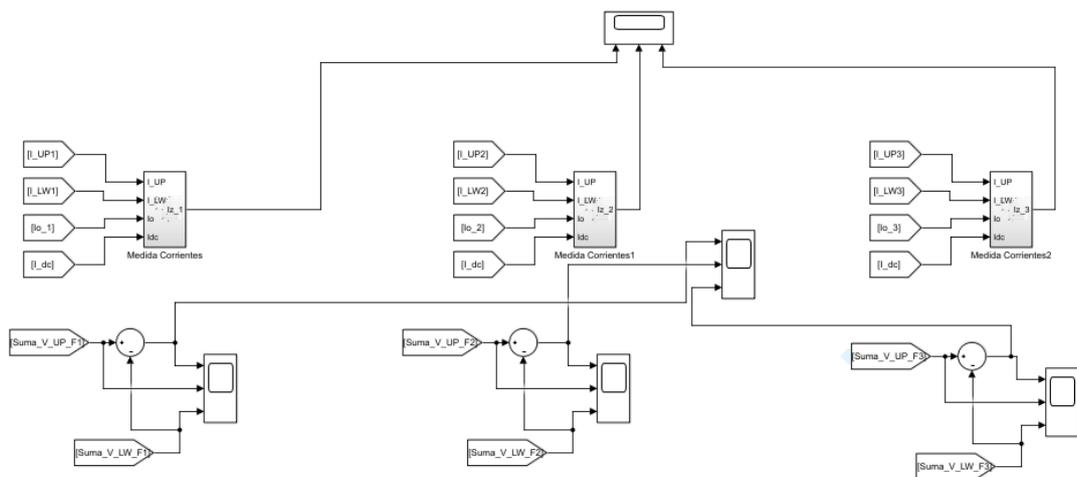


Figura 49 - Scopes de corrientes circulantes y tensiones de las ramas del convertidor MMC en Simulink

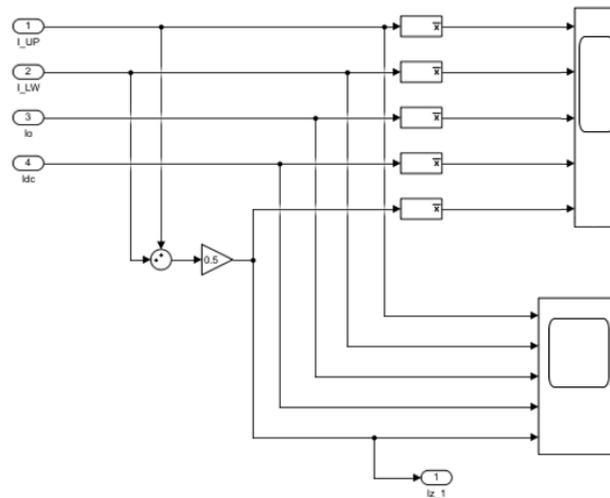


Figura 50 - Bloque subsistema "Medida Corrientes"

3.2.2.- Controlador

En el apartado anterior, se ha presentado la planta del convertidor. En este apartado se procede a explicar de un modo más concreto el bloque de la función embebida *controller*, mostrando las diferentes partes del código y su explicación. Algunas partes del código se repiten para las tres fases o para la rama superior e inferior. En estos casos solo se explica para uno de los casos (rama superior, fase A). El código completo se puede ver en el anexo.

La función embebida “*controller*” ejerce de controlador del convertidor, recibe información de la planta y envía las órdenes oportunas. Como se ha visto en el apartado anterior, la función se ejecuta cada 250 μ s, que es la frecuencia con la que el bloque de tipo pulso acciona el bloque *Trigger* que ordena la ejecución del controlador.

Este periodo de 250 microsegundos es un valor arbitrario, pero se ha tenido en cuenta que:

- Este periodo debe ser lo suficientemente grande como para que, en el simulador en tiempo real, dé tiempo a ejecutarse el programa del controlador en cada periodo.
- La frecuencia de las portadoras marca la frecuencia de conmutación de los polos de potencia, cuantas más conmutaciones haya habrá mayores pérdidas.

La función *controller* debe estar sincronizada con las funciones *PWM_generator* de todas las ramas. Para ello, una vez fijado el periodo de ejecución del controlador en 250 microsegundos, las funciones de PWM generan rampas ascendentes o descendentes durante 250 μ s, creciendo o decreciendo a un ritmo de 1 por cada microsegundo. Por lo tanto, cada nivel de

tensión de las portadoras es de 250. Para poder comparar el valor normalizado de la moduladora (son valores entre 0.0 y 1.0) con el de las portadoras, se multiplica estos valores normalizados por 250.

Las entradas y salidas de la función del controlador se han podido ver en el apartado anterior cuando se ha mostrado el bloque correspondiente al controlador (Figura 44). A continuación, se recuerdan cuáles eran las entradas y salidas de forma resumida.

Tiene como entradas:

- Las tensiones de las tres fases del polo positivo.
- La corriente de dicho polo.
- Las tensiones de las tres fases del polo negativo.
- La corriente del polo negativo.

Por otro lado, la función proporciona:

- Las señales moduladoras
- Las señales de sincronización.

3.2.2.1.- *Setup*

El programa comienza con el *setup*, es decir, con la configuración, definiendo e inicializando unas variables del tipo "*persistent*" o persistentes que aparecen en color azul. Este tipo de variables conserva su valor de una iteración a otra, por ello, se inicializan solamente en la primera iteración, con el condicional "*isempty*", como se muestra en la Figura 51.

```

% setup()

persistent Ts myTime w pSyncUP pSyncLW ;
persistent m_UP1 m_UP2 m_UP3 m_LW1 m_LW2 m_LW3 swUP swLW;
persistent Nsm Vsm invVsm Vac_peak Vdc_half;
if ( isempty(Ts) )
    Ts      = 250e-6;          % El controlador se ejecuta cada 250 us
    myTime = 0;
    w       = 2 * pi * 50;    % 50 Hz

    Nsm = 4;
    Vsm  = 800;
    Vac_peak = 1200;
    invVsm = 1 / Vsm;
    Vdc_half = Nsm * Vsm / 2;

    pSyncUP = [0 0 0];
    pSyncLW = [0 0 0];
    m_UP1 = [1 1 0 0];
    m_UP2 = [1 1 0 0];
    m_UP3 = [1 1 0 0];
    m_LW1 = [1 1 0 0];
    m_LW2 = [1 1 0 0];
    m_LW3 = [1 1 0 0];
    swUP = [0 0 0]; % 1 es ON, 0 es OFF
    swLW = [0 0 0]; % 1 es ON, 0 es OFF
end

```

Figura 51 - Código de la función "controller" del convertidor de Simulink. Setup

Gracias a la condición "if" que contiene el comando "isempty", si la variable cumple la condición de "vacía", entonces se inicializa con el valor que se le indica en el código dentro de este condicional. Basta con comprobar si una de las variables está vacía, tal como se ve en la Figura 51, solo se comprueba la variable T_s , y en caso afirmativo se inicializan todas las variables, ya que en la primera iteración todas las variables estarán vacías.

En las próximas iteraciones, ya no se entrará en esta línea al contener las variables el valor de la iteración anterior, es decir, ya no cumplen la condición de "vacías".

A continuación, se describen las variables *persistent* que se inicializan en el *setup*:

T_s es el periodo de ejecución del programa. Cada 250 μ s el programa se empieza a ejecutar y cuando termina permanece a la espera de que termine el periodo. En Matlab/Simulink, al no estar en tiempo real esta variable de periodo solo influye a nivel de cálculo como referencia de tiempo. En el simulador en tiempo real además será un límite que el tiempo de ejecución del programa no podrá sobrepasar.

myTime es una variable para acumular el tiempo transcurrido. Cada vez que se ejecuta el programa habrán pasado 250 μ s.

La variable w es la frecuencia angular, depende de la frecuencia de la corriente alterna que es de 50 Hz.

Nsm es el número de submódulos o celdas por rama que tiene el convertidor, que en este caso son 4.

Vsm es la tensión que aporta cada submódulo cuando está encendido. En este convertidor esta tensión es de 800 V.

Vac_peak es la variable para la tensión pico en corriente alterna, cuyo valor es de 1200 V.

InvVsm tiene el valor de la inversa de *Vsm*. Esta variable se ha definido porque las operaciones de división son más costosas de realizar que las sumas o multiplicaciones en cuanto a tiempo.

Vdc_half está definida como el producto de *Nsm* por *Vsm* dividido entre 2. Es la tensión media en corriente continua.

pSyncUP y *pSyncLW* son variables vectoriales de tres componentes, *pSyncUP* es la señal de sincronización correspondiente a las tres fases de la rama superior. *pSyncLW* es la variable vectorial encargada de la sincronización de las tres fases de la rama inferior. Se inicializan como [0, 0, 0] como punto de partida.

Los vectores *m_UP1*, *m_UP2*, *m_UP3*, *m_LW1*, *m_LW2* y *m_LW3*, de cuatro componentes cada uno, almacenan el estado de los polos de potencia de cada rama en cada iteración. Se han inicializado en [1, 1, 0, 0] de manera arbitraria, ya que daría igual qué celdas estén activas en el inicio siempre y cuando la suma de las celdas activas en cada fase sumen cuatro.

Los vectores *swUP* y *swLW* se inicializan como [0, 0, 0]. Almacenan si en la iteración anterior hubo un *off* (*off* = 0) o si hubo un *on* (*on* = 1). La variable *swUP* es para las tres fases de la rama inferior y *swLW* para las de la rama superior.

3.2.2.2.- *loop*

Tras inicializar las variables, comienza el bucle o *loop*:

El *loop* comienza con la actualización de tiempo, que, como el controlador se ejecuta cada 250 μ s, el tiempo actualizado será el tiempo de la iteración anterior más 250 microsegundos. Tal como se puede ver en la Figura 52.

```
% Actualización del tiempo  
myTime = myTime + Ts;
```

Figura 52 - Actualización de tiempo en el convertidor de Matlab

3.2.2.2.1.- *Generación de señales de sincronización*

A continuación, se generan las señales de sincronización. Estas señales se envían a los generadores de PWM y según lleven el valor 1 o 0, las portadoras describirán un flanco ascendente o descendente respectivamente.

En Simulink se ha programado una señal de sincronización para cada rama, es decir, seis señales que se encuentran en dos vectores *persistent* de tres elementos cada uno, *pSyncUP* la rama superior o positiva y *pSyncLW* para la rama inferior o negativa. Al tener cada rama su propia señal de sincronización se puede ser más preciso con las necesidades de cada una de ellas, es decir, si necesitan un flanco (o varios) ascendente o descendente, al tener cada rama una señal diferente, se puede ajustar mejor que si hubiera una sola señal de sincronización para todas las ramas.

Las variables “*persistent*” *swUP* y *swLW*, indican qué ocurrió en la iteración anterior, es decir, si fue necesario un ON o si se requirió un OFF. Son vectores de tres componentes, una componente para cada rama.

```

for i = 1:3
    if (swUP(i) == 0)
        if (pSyncUP(i) > 1)
            pSyncUP(i) = 0;
        else
            pSyncUP(i) = pSyncUP(i) - 1;
        end
    else
        if (pSyncUP(i) < -1)
            pSyncUP(i) = 0;
        else
            pSyncUP(i) = pSyncUP(i) + 1;
        end
    end
end

if (swLW(i) == 0)
    if (pSyncLW(i) > 1)
        pSyncLW(i) = 0;
    else
        pSyncLW(i) = pSyncLW(i) - 1;
    end
else
    if (pSyncLW(i) < -1)
        pSyncLW(i) = 0;
    else
        pSyncLW(i) = pSyncLW(i) + 1;
    end
end
end % for señal sincronización

```

Figura 53 - Generación de señales de sincronización del convertidor de Simulink

El programa, que se puede ver en la Figura 53, funciona de la siguiente manera: si se necesitó un OFF la señal de sincronización se disminuye en uno (se resta 1). En el caso contrario, si se necesitó un ON se aumenta en uno la respectiva señal de sincronización. Con este método, la señal de sincronización podría irse a valores muy grandes o muy pequeños a causa de que pueda haber más OFF que ON o viceversa. Como medida de prevención, se añadió la condición de que en caso de ser mayor que 1 su valor pase a ser 0 y en el caso de que llegue a ser menor que -1 que pase a valer 0. El bucle *for* sirve para que se repita el programa para las tres fases.

Finalmente, como se muestra en la Figura 54, se pasa los valores calculados de los vectores “*persistent*” a los vectores que son argumentos de salida de la función embebida “*controller*”

```
SyncUP = pSyncUP;  
SyncLW = pSyncLW;
```

Figura 54 - Asignación de señal de sincronización

3.2.2.2.2.- Consolidación del estado de los polos

El controlador elige en cada periodo que celda o celdas deben encenderse o apagarse. La celda elegida es la encargada de hacer la modulación, se enciende y apaga durante un tanto por ciento del periodo para que el valor promedio sea el buscado. Cuando se acaba el periodo esta celda debe permanecer en el estado que se había elegido en el periodo anterior (encendida o apagada). Es decir, se consolida el estado de los polos de potencia.

Para la consolidación del estado de los polos, con las variables “*persistent*” *swUP* y *swLW* se conoce si en la iteración anterior hubo un ON o un OFF.

- En el caso de que hubo un OFF, busca en el vector que guarda el estado de los polos la componente que es menor que 1 y la pone a 0.
- En caso de que hubiera habido un ON, busca la componente del vector que sea mayor que 0 y la pone igual a 1.

Esto lo hace con las tres ramas superiores y las tres inferiores. Al utilizar vectores para almacenar los datos, se ha utilizado bucles *for* para realizar estas operaciones repetitivas para las tres fases. En Simulink no hay tiempo real, por lo que el tiempo que tarda en ejecutarse el programa no importa tanto como sí que lo hará luego con el simulador en tiempo real. Por eso mismo se ha utilizado aquí los bucles *for*, pero no se ha hecho así en el simulador en tiempo real.

En la Figura 55 y la Figura 56 se muestra el código que se ha implementado en la función embebida “*controller*” para realizar la consolidación del estado de los polos.

```

% 1. Consolidación del estado de los polos
% Rama superior
for j = 1:3 % Para la fase 1, 2 y 3
    if swUP(j) == 0 % se hizo un off en el periodo anterior
        % el que no sea 1.0 que sea 0.0
        if j == 1
            for i=1:4
                if (m_UP1(i) < 1.0)
                    m_UP1(i) = 0.0;
                end
            end
        elseif j == 2
            for i=1:4
                if (m_UP2(i) < 1.0)
                    m_UP2(i) = 0.0;
                end
            end
        else % j = 3
            for i=1:4
                if (m_UP3(i) < 1.0)
                    m_UP3(i) = 0.0;
                end
            end
        end
    end
end

```

Figura 55 - Consolidación del estado de los polos del convertidor de Simulink (Parte 1)

```

else % el que no sea 0.0 que sea 1.0 (hubo un ON)
    if j == 1
        for i=1:4
            if (m_UP1(i) > 0.0)
                m_UP1(i) = 1.0;
            end
        end
    elseif j == 2
        for i=1:4
            if (m_UP2(i) > 0.0)
                m_UP2(i) = 1.0;
            end
        end
    else % j = 3
        for i=1:4
            if (m_UP3(i) > 0.0)
                m_UP3(i) = 1.0;
            end
        end
    end
end
end
end
end

```

Figura 56 - Consolidación del estado de los polos del convertidor de Simulink (Parte 2)

3.2.2.2.3.- Cálculo del número de celdas activas en cada rama

Para calcular el número de celdas activas en cada rama, se suman las cuatro componentes del vector que contiene el estado de los polos consolidado. Los vectores que almacenan el estado de los polos en cada rama son m_{UP1} , m_{UP2} , m_{UP3} , m_{LW1} , m_{LW2} y m_{LW3} . Es decir, solo tiene componentes que valen 1 o que valen 0, por tanto, con la suma de las componentes de estos

vectores se obtendrá el número de celdas que se encuentran encendidas en esta iteración. En la Figura 57 se puede ver cómo se ha implementado esto en el código de la función “*controller*” del convertidor de Simulink.

```

% Rama superior
N_activasUP1 = m_UP1(1) + m_UP1(2) + m_UP1(3) + m_UP1(4);
N_activasUP2 = m_UP2(1) + m_UP2(2) + m_UP2(3) + m_UP2(4);
N_activasUP3 = m_UP3(1) + m_UP3(2) + m_UP3(3) + m_UP3(4);

% Rama inferior
N_activasLW1 = m_LW1(1) + m_LW1(2) + m_LW1(3) + m_LW1(4);
N_activasLW2 = m_LW2(1) + m_LW2(2) + m_LW2(3) + m_LW2(4);
N_activasLW3 = m_LW3(1) + m_LW3(2) + m_LW3(3) + m_LW3(4);

```

Figura 57 - Cálculo del número de celdas activas en el convertidor de Simulink

3.2.2.2.4.- Generación de señales moduladoras

En este punto se generan las señales moduladoras. Esta señal tiene la forma de la onda que se quiere transformar. Para que al comparar la señal moduladora con las portadoras el resultado de un número normalizado dentro del rango 0.0 a 4.0, las moduladoras se dividen entre la tensión de una celda.

```

% 3. Generación de las señales moduladoras (valor de 0.0 a 4.0)
% Rama superior
m_P1 = invVsm * (Vdc_half - Vac_peak * sin(w * myTime)); % FASE 1
m_P2 = invVsm * (Vdc_half - Vac_peak * sin(w * myTime - 2/3*pi)); % FASE 2
m_P3 = invVsm * (Vdc_half - Vac_peak * sin(w * myTime - 4/3*pi)); % FASE 3

% Rama inferior
m_N1 = invVsm * (Vdc_half + Vac_peak * sin(w * myTime)); % FASE 1
m_N2 = invVsm * (Vdc_half + Vac_peak * sin(w * myTime - 2/3*pi)); % FASE 2
m_N3 = invVsm * (Vdc_half + Vac_peak * sin(w * myTime - 4/3*pi)); % FASE 3

```

Figura 58 - Código de Matlab. Generación de señales moduladoras

El cálculo para obtener dichas señales es como se observa en la Figura 58, la mitad de la tensión en corriente continua menos (más en la rama inferior) la tensión en alterna, todo esto dividido entre la tensión de un submódulo.

Al sumar las señales superior e inferior de una fase, se anula la componente continua y quedaría la señal en corriente alterna sin componente de continua alguna.

3.2.2.2.5.- Cálculo del residuo

La celda que elige el controlador para que cambie de estado, solamente se encuentra un tanto por ciento del periodo encendida. Este porcentaje de actividad lo determina el residuo.

Se llama residuo a la parte decimal del valor de las moduladoras que se han calculado. Es decir, si el valor normalizado de la moduladora es 2.6, tendrá que haber dos celdas encendidas durante todo el periodo y una tercera celda, la

elegida para encenderse o apagarse, que estará encendida solamente un 60% del tiempo.

Entonces se puede dividir el valor de la moduladora en celdas encendidas (o en 1) y celdas apagadas (o en 0) y la celda elegida por el controlador que solo está encendida un % del periodo. El valor del residuo es el tanto por uno del tiempo del periodo que la celda debe estar encendida para que en promedio se dé el valor de tensión deseado.

Con la función *floor* de Matlab (también existe en lenguaje C) se obtiene la parte entera del valor de la moduladora. Al calcular la diferencia entre el valor de la moduladora y la parte entera de la moduladora, se halla el residuo.

Este razonamiento ha sido incorporado al código de la función “*controller*” de la forma en que se puede ver en la Figura 59.

```
% Rama superior
residuo_P1 = m_P1 - floor(m_P1);
residuo_P2 = m_P2 - floor(m_P2);
residuo_P3 = m_P3 - floor(m_P3);

% Rama inferior
residuo_N1 = m_N1 - floor(m_N1);
residuo_N2 = m_N2 - floor(m_N2);
residuo_N3 = m_N3 - floor(m_N3);
```

Figura 59 - Cálculo del residuo del convertidor de Simulink

3.2.2.2.6.- Selección de las celdas que deben ser insertadas o retiradas

La tensión que aporta cada celda es aportada por el condensador que contiene cada una de ellas. El número de celdas encendidas depende de la comparación entre la señal moduladora con las portadoras, de tal forma que hay cuatro niveles de tensión, uno por cada celda.

Las conmutaciones entre las celdas deben ser óptimas, ya que, si el condensador de una celda se descarga demasiado, no podrá mantener la tensión y aportará menos de lo esperado, por lo que podría distorsionar la onda.

En la Figura 60 se observa un criterio de conmutación poco eficiente. Con este criterio, cada celda tiene asignado un nivel de tensión, de tal forma que, en el primer nivel de tensión, siempre se enciende la primera celda, para el segundo nivel estaría siempre encendida la primera y la segunda celda haría la modulación, en el tercer nivel de tensión siempre se encenderían la primera, la segunda y la tercera haría la modulación y, por último, en el cuatro nivel de tensión, se encontrarían las tres primeras celdas encendidas y con la cuarta se haría la modulación.

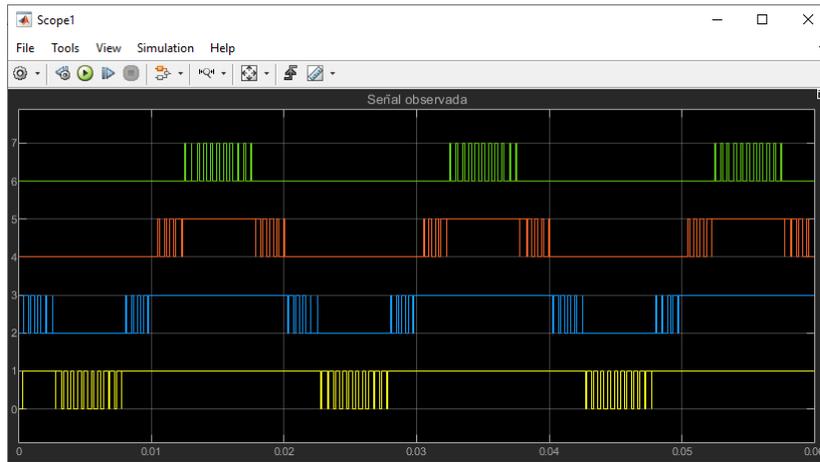


Figura 60 - Conmutación de las celdas con criterio de conmutación ineficiente

Con este criterio de conmutación, las celdas de los niveles inferiores se utilizarían la mayor parte del tiempo y se descargarían más que las de los niveles superiores. La cuarta celda apenas entraría en funcionamiento. En la Figura 61 se puede ver las tensiones de los condensadores. Se puede ver que mientras las tensiones de algunas celdas apenas varían, las de otras celdas caen y oscilan mucho.

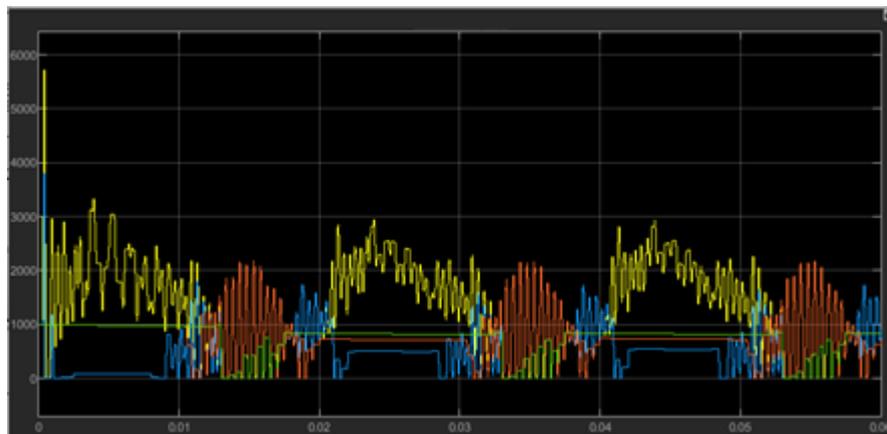


Figura 61 - Tensiones de los condensadores de las celdas usando el criterio de conmutación ineficiente

Este criterio de conmutación tiene consecuencias sobre la forma de onda de la corriente y la tensión, tal como se observa en la Figura 62 y en la Figura 63, las ondas quedan bastante distorsionadas.

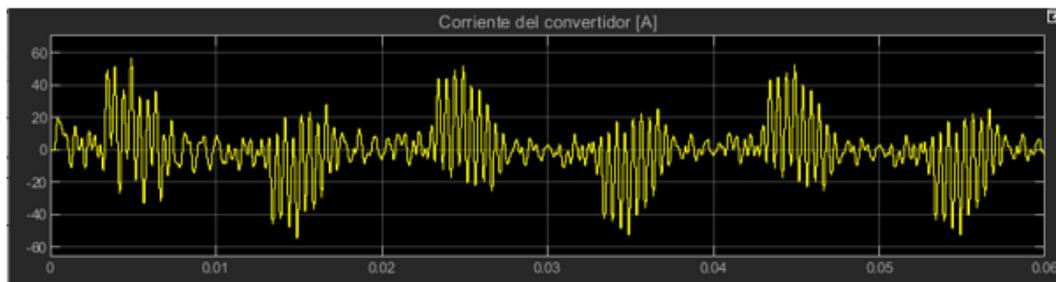


Figura 62 - Corriente de salida del convertidor con el criterio de conmutación ineficiente

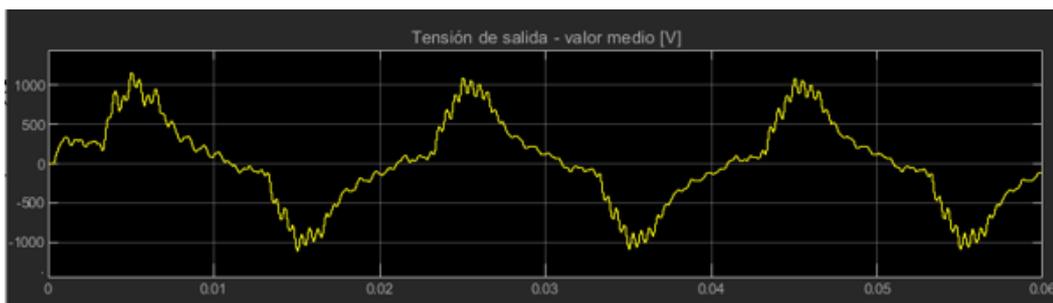


Figura 63 - Tensión de salida del convertidor con el criterio de conmutación ineficiente

Para que esto no ocurra se deben repartir las conmutaciones entre todas las celdas de forma equitativa.

Para poder repartir de esta manera equitativa el uso de todas las celdas, se debe tener en cuenta el sentido en el que está circulando la corriente:

- Si la corriente circula en sentido positivo:
 - Si la celda está encendida, la corriente circulará por el condensador y lo cargará.
 - Si la celda está apagada, no circulará corriente por el condensador, entonces permanecerá inalterado.
- Si la corriente circula en sentido negativo:
 - Si la celda está encendida, la corriente circulará por el condensador, pero en este caso se descargará.
 - Si la celda está apagada, la corriente no circulará por el condensador, entonces no lo descargará.

El controlador deberá encender una celda si el número de celdas activas es menor que el valor normalizado de la moduladora y tendrá que apagar una celda cuando el número de celdas activas sea mayor que el valor normalizado de la moduladora.

Por lo tanto, el criterio para seleccionar las celdas que cambian de estado depende de si hay que encender una celda o de si hay que apagarla y del sentido de circulación de la corriente.

- Cuando haya que apagar una celda:
 - Si la corriente es positiva, se deberá seleccionar de entre las celdas insertadas la celda que tenga la tensión más elevada, para que no siga aumentando.
 - Si la corriente es negativa, se debe elegir de entre las celdas encendidas la de menor tensión, para que no continúe descargándose.
- Cuando haya que encender una celda:
 - Si la corriente es positiva, se debe escoger de las celdas apagadas la que tenga menor tensión, para que al circular la corriente positiva se cargue.
 - Si la corriente es negativa, la celda que se debe seleccionar será, de las no insertadas la que tenga la máxima tensión, para que se descargue al paso de la corriente.

Con este criterio de conmutación, se reparte equitativamente el trabajo entre las cuatro celdas de cada rama. Cada celda ya no está vinculada a un nivel de tensión, sino que se escoge la celda más conveniente para encenderla o para apagarla. En la Figura 64 se ve como las celdas conmutan de forma más equitativa.

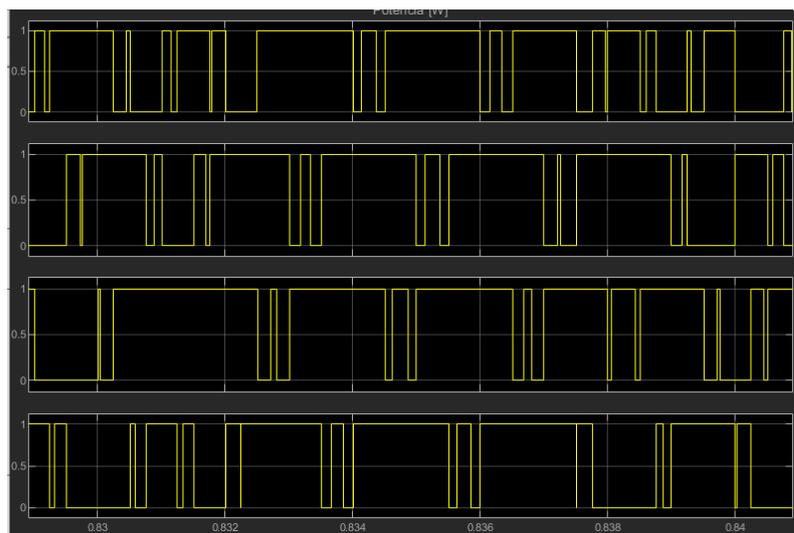


Figura 64 - Conmutación de las celdas con el criterio de conmutación eficiente

Por otro lado, en la Figura 65 se puede observar cómo, con este criterio, todas las tensiones en los condensadores se comportan de forma similar. Las

tensiones de todos los condensadores de una rama se mueven dentro de un rango menor que ± 5 .

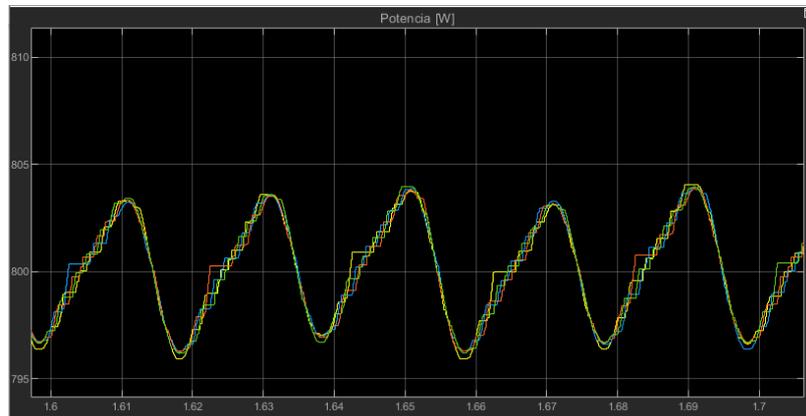


Figura 65 - Tensiones de los condensadores de las celdas usando el criterio de conmutación eficiente

Al cambiar el criterio de selección de las celdas a este más eficiente, las ondas de corriente y tensión de salida ya no se ven tan distorsionadas (Figura 66 y Figura 67). La corriente tiene algo de rizado, pero conserva la forma senoidal.

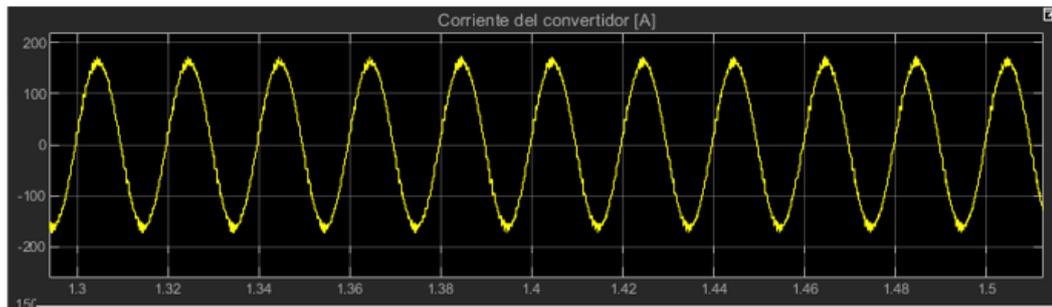


Figura 66 - Corriente de salida del convertidor con el criterio de conmutación eficiente

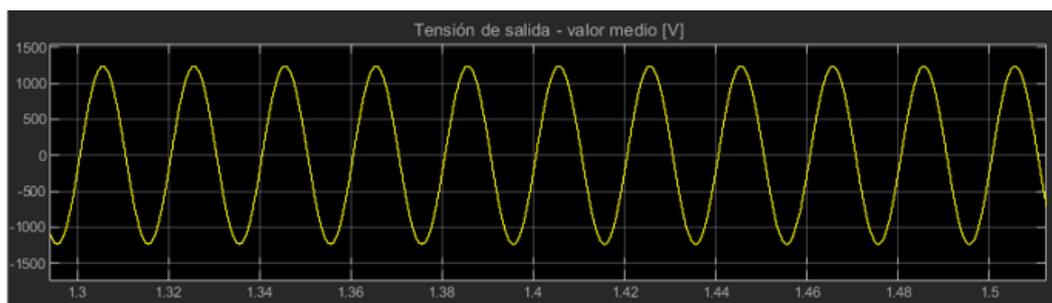


Figura 67 - Tensión de salida del convertidor con el criterio de conmutación eficiente

Para imponer este criterio de selección en el controlador, se han creado las funciones:

```
pos = maxV(insertadas, V, S) y pos = minV(insertadas, V, S)
```

Ambas funciones tienen como argumentos de entrada:

- La variable “*insertadas*” que vale 0 cuando se requiere buscar entre las NO insertadas y vale 1 cuando hay que buscar entre las insertadas.
- El argumento *V*, es un vector que almacena las tensiones de las cuatro celdas
- El argumento *S* es otro vector que almacena el estado de los polos de potencia, para saber cuáles están encendidos (1 == ON) y cuales no (0 == OFF).

Como argumento de salida se encuentra la variable “*pos*”, que da la posición de la celda que hay que encender o apagar, es decir, retorna un número del 1 al 4.

Las funciones *maxV* y *minV* funcionan de manera similar pero opuesta:

La función *maxV* selecciona la celda de mayor tensión de entre las celdas apagadas o encendidas según se indique en la variable “*insertadas*”.

La función *minV* selecciona la celda que tenga la menor tensión de entre las celdas apagadas o encendidas según el valor de la variable “*insertadas*”.

Dentro de la función *maxV*, se inicializa la variable “*pos*” a 0 y la variable “*maxValue*” a un número considerado menos infinito, en este caso -10000 es suficiente.

La función toma el valor de la primera posición de *S*, para saber si esa celda está insertada o no. En caso de coincidir el estado de la celda con el buscado (insertada o no insertada), compara el valor de la tensión de esa celda con el “*maxValue*”, que es seguro inferior a ese valor de tensión, entonces ese valor pasa a ser la máxima y se pone temporalmente como la celda seleccionada la 1, *pos* = 1. Se realiza la misma comparación con la posición 2, 3 y 4. De esta manera, elegirá la celda que tenga la mayor tensión de entre las que estén insertadas o que no lo estén según se lo indiquemos en el argumento “*insertadas*”. El código de la función *maxV* se puede ver en la Figura 68.

```

function pos = maxV(insertadas, V, S)

pos=0;
maxValue = -10000;
if S(1) == insertadas
    if V(1) > maxValue
        maxValue = V(1);
        pos = 1;
    end
end
if S(2) == insertadas
    if V(2) > maxValue
        maxValue = V(2);
        pos = 2;
    end
end
if S(3) == insertadas
    if V(3) > maxValue
        maxValue = V(3);
        pos = 3;
    end
end
if S(4) == insertadas
    if V(4) > maxValue
        % maxValue = V4;
        pos = 4;
    end
end
end

```

Figura 68 - Código de la función $pos = \max V(insertadas, V, S)$

La función *minV* funciona de la misma manera, pero en este caso la variable *minValue* inicializada a 10000, un número muy elevado. Ahora, si el valor de la tensión de la celda es menor que ese *minValue*, pasará a ser esa tensión la *minValue* y *pos* cogerá la posición de esa celda hasta que revise de la 1 a la 4 las que estén insertadas o no, el estado que se le haya indicado buscar. El código de la función *minV* se muestra en la Figura 69.

```

function pos = minV(insertadas, V, S)

pos=0;
minValue = 10000;
if S(1) == insertadas
    if V(1) < minValue
        minValue = V(1);
        pos = 1;
    end
end
if S(2) == insertadas
    if V(2) < minValue
        minValue = V(2);
        pos = 2;
    end
end
if S(3) == insertadas
    if V(3) < minValue
        minValue = V(3);
        pos = 3;
    end
end
if S(4) == insertadas
    if V(4) < minValue
        % minValue = V4;
        pos = 4;
    end
end
end % minV

```

Figura 69 - Código de la función $pos = \min V(insertadas, V, S)$

Una vez aclarado el criterio de selección de las celdas y presentadas las funciones que lo aplican, se procede a explicar cómo se ha implementado este criterio en el controlador:

Lo primero de todo, se compara el valor de la moduladora (m_{P1}) con el número de celdas activas ($N_{activasUP1}$), si el valor normalizado de la moduladora es menor que el número de celdas activas será necesario que haya un OFF, es decir, que se apague una celda, mientras que en el caso contrario necesitaremos un ON, es decir, que se encienda una celda.

Si debe haber un OFF (ver Figura 70):

La componente correspondiente del vector sw_{UP} o sw_{LW} se pone a 0, para indicar en la próxima iteración que se ha apagado una celda en esta iteración.

Para usar las funciones $maxV$ y $minV$, se tiene en cuenta que hay que buscar entre las celdas insertadas para apagar la más conveniente, de acuerdo con el criterio que se ha explicado anteriormente en este apartado.

- Cuando la corriente sea positiva, se busca la celda que tenga la mayor tensión, por lo que corresponde la función $maxV$.
- Cuando la corriente sea negativa, se busca la celda con menor tensión entre las insertadas, para que no se descargue más, por lo que la función $minV$ es la requerida.

La celda elegida por la función que haya entrado en la iteración actual, se le pone el valor del residuo, ya que el residuo representa el tanto por uno del periodo que debe estar la celda encendida para que, en promedio, el valor de tensión sea el deseado.

Sobre la variable *pos* se ha puesto una condición. Debe ser mayor que 0 para que se realice la asignación del residuo a la celda seleccionada. Simplemente para asegurar que en todo momento la celda seleccionada tiene un valor positivo (1, 2, 3, 4).

Podría ocurrir que se requiera el apagado de más de una celda en la misma iteración. Por este motivo, todo lo anterior se encuentra en un bucle *while*, para que se ejecute el bucle mientras el valor de la moduladora sea menor que el número de celdas activas.

En las últimas líneas del bucle *while* se actualiza el número de celdas activas, ya que tras cada repetición del bucle quedará una celda menos por apagar.

Después de la primera entrada en el bucle, se fija el residuo en un valor igual a 0, para que la primera celda elegida para apagarse tenga el valor del residuo, pero el resto se apaguen por completo, es decir, se les asigna directamente el valor de 0.

Si debe haber un ON (ver Figura 71):

En este caso la componente correspondiente del vector *swUP* o *swLW* pasa a valer 1, para que en la próxima iteración se recuerde que en la anterior iteración hubo un ON.

El valor normalizado de la moduladora es mayor que el número de celdas activas, entonces, es necesario encender al menos una celda. Por lo tanto, en esta situación hay que buscar con las funciones *maxV* y *minV* entre las celdas no insertadas.

- Cuando la corriente sea positiva, se escogerá la celda que tenga la menor tensión de entre las no insertadas, para que se cargue. Por tanto, se utilizará la función *minV*.
- Cuando la corriente sea negativa, se elegirá la celda que tenga la mayor tensión de las no insertadas, utilizando la función *maxV* para que, de esta manera, se descargue.

Igualmente, este programa se encuentra en un bucle *while* para cuando sea necesario encender más de una celda en una misma iteración.

Ahora cuando se actualiza el número de celdas activas se suma 1, ya que cada vez habría una celda más encendida, y, el valor del residuo pasa a ser 1 tras la primera iteración del bucle *while*.

```

% FASE 1 Rama superior
if (m_P1 < N_activasUP1)           % debe haber OFF (o doble OFF)

    while (m_P1 < N_activasUP1) % hace OFF una o varias veces

        swUP(1) = 0;

        if (I_P(1) > 0)
            pos_P1 = maxV(1, V_P1, m_UP1);
        else % I_P < 0
            pos_P1 = minV(1, V_P1, m_UP1);
        end

        if (pos_P1 > 0)
            m_UP1(pos_P1) = residuo_P1;
        end
        N_activasUP1 = N_activasUP1 - 1;
        residuo_P1 = 0;
    end
end

```

Figura 70 - Selección de celdas en el convertidor de Simulink (Parte 1)

```

else                               % debe haber ON (o doble ON)

    while (m_P1 > N_activasUP1) % hace ON una o varias veces

        swUP(1) = 1;

        if (I_P(1) > 0)
            pos_P1 = minV(0, V_P1, m_UP1);
        else % I_P < 0
            pos_P1 = maxV(0, V_P1, m_UP1);
        end

        if (pos_P1 > 0)
            m_UP1(pos_P1) = residuo_P1;
        end
        N_activasUP1 = N_activasUP1 + 1;
        residuo_P1 = 1;
    end
end

```

Figura 71 - Selección de celdas en el convertidor de Simulink (Parte 2)

3.2.2.2.7.- Generación de las salidas

Finalmente, el controlador obtiene el nuevo estado de los polos de potencia y extrae las seis señales moduladoras, una para cada rama. Estas señales se envían a los generadores de PWM para comparar la moduladora con las portadoras.

3.2.3.- Generador de PWM en Simulink

Para generar las señales de PWM en Simulink, se ha utilizado otro bloque de función embebida. Cada fase de las ramas superiores y cada fase de las ramas inferiores tiene su bloque de generación de PWM, en total son 6 generadores de PWM. En todos los casos la función embebida es la misma.

Como ya se ha mencionado, la función embebida de los generadores de PWM se ejecutan cada microsegundo.

Tiene como argumentos de entrada el vector “*modulator*” que contiene el estado de cada celda, y la variable *sync*, que contiene la señal de sincronización correspondiente.

La función comienza definiendo las variables *persistent carrier*, *oldsync* y *step*, para generar las portadoras, como se muestra en la Figura 72.

Estas variables se inicializan tal como se observa en la Figura 72. Para ello se ha utilizado el comando *isempty*.

```
persistent carrier step oldSync;
if ( isempty(carrier) )
    carrier = 249.5;
    step = -1;
    oldSync = 0;
end
```

Figura 72 - Definición e inicialización de variables de la función "pwm_gen"

Tras haber inicializado las variables en la primera iteración, comienza el *loop*, es decir, lo que se va a ejecutar en cada periodo.

Esta función tiene que generar la forma de las portadoras. Con este fin, el programa comienza con la primera línea que aparece en la Figura 73.

```
% Generación de la portadora
carrier = carrier + step;
if (Sync < oldSync)      % 250us subiendo
    carrier = 0.5;
    step = +1;
end
if (Sync > oldSync)      % 250us bajando
    carrier = 249.5;
    step = -1;
end

oldSync = Sync;
```

Figura 73 - Loop de la función "pwm_gen"

Entonces en cada microsegundo, que es el periodo de ejecución del generador de PWM, el *carrier* (la portadora) disminuye en 1, porque el *step* ha sido definido como -1. Por lo tanto, se genera un flanco descendente (\searrow) desde 249.5, que es el valor inicial de *carrier*, hasta 0.5 después de 250 μ s, que es el periodo de ejecución del controlador.

Al terminar el periodo de 250 μ s el controlador aportará una nueva señal de sincronización que, en función de si se requiere un ON o un OFF, se buscará que haya un flanco ascendente (/) o un flanco descendente (\), respectivamente.

Si la nueva señal de sincronización es menor que la anterior, durante el próximo periodo de 250 microsegundos se describirá un flanco ascendente. Para ello, se fija *carrier* = 0.5 y *step* = 1. Tal como se puede ver en la Figura 73.

Mientras que, si la señal de sincronización actual es mayor que la anterior, se dibujará un flanco descendente. En consecuencia, se pone *carrier* = 249.5 y *step* = -1.

Tras cada iteración se fija *oldsync* = *sync*, de tal manera que no se inicialice el *carrier* hasta que termine periodo de 250 μ s del controlador, cuando envía las nuevas señales de sincronización a los generadores de PWM.

Según el valor de la señal de sincronización que llega del controlador se indica al generador de PWM si se necesita un OFF o un ON. Si en la iteración anterior hubo un OFF, en la actual se ordena un flanco ascendente para que se dé un ON, como se muestra de forma esquemática en la Figura 75. Si en la iteración anterior hubo un ON, en la iteración actual se ordena un flanco descendente para que se dé un OFF, como se puede ver en la Figura 74 de forma esquemática.

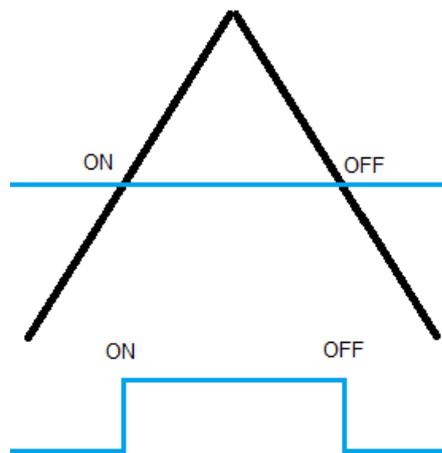


Figura 74 - Flanco ascendente asociado a un ON seguido de un flanco descendente asociado a un OFF (elaboración propia)

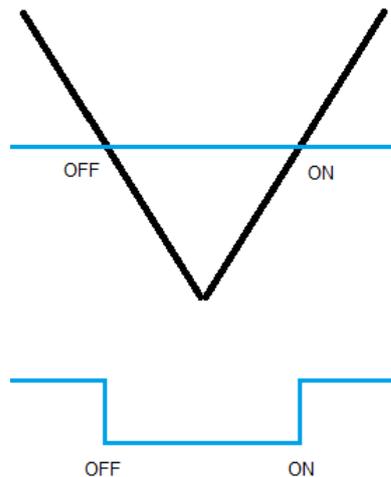


Figura 75 - Flanco descendente asociado a un OFF seguido de un flanco ascendente asociado a un ON (elaboración propia)

En el controlador se pondrá la señal de sincronización que disminuya en razón de -1 , cuando en la iteración anterior se necesitó apagar una celda y de esta forma tener un flanco ascendente para que en el periodo actual haya un ON. En el código del generador de PWM, la variable *sync* pasaría a ser menor que *oldsync*, generando el mencionado flanco ascendente.

En el controlador se asignará la señal de sincronización que aumente en $+1$ cuando en la iteración anterior se requirió encender una celda, para tener un flanco descendente con el objetivo de que en este periodo haya un OFF. En el código del generador de PWM, la variable *sync* sería mayor que *oldsync*, formando dicho flanco descendente.

Después de haber generado la señal portadora, esta se compara con las moduladoras, como se muestra en la Figura 76. Para que el valor normalizado de las moduladoras sea comparable con la magnitud de la portadora, se multiplica a las moduladoras por 250. Cuando la moduladora se encuentra por debajo de la portadora la celda estará apagada, el polo de potencia vale 0, cuando la moduladora esté por encima de la portadora la celda correspondiente estará encendida, el polo de potencia vale 1.

```

if (modulator(1) * 250 < carrier)
    S1 = 0;
else
    S1 = 1;
end

if (modulator(2) * 250 < carrier)
    S2 = 0;
else
    S2 = 1;
end

if (modulator(3) * 250 < carrier)
    S3 = 0;
else
    S3 = 1;
end

if (modulator(4) * 250 < carrier)
    S4 = 0;
else
    S4 = 1;
end

```

Figura 76 - Comparación entre moduladora y portadora

3.2.4.- Resultados del convertidor en Simulink

Una vez expuesta la planta del convertidor, el código del controlador y el código de los generadores de PWM, en este apartado se va a evaluar la validez de este convertidor. Para ello se verá las tensiones y corrientes que aporta el convertidor a la salida en el lado de corriente alterna, las corrientes circulantes y los armónicos.

En la Figura 77 se pueden ver las tensiones de salida del convertidor. Como se observa, estas tensiones representan un sistema trifásico equilibrado y con la frecuencia esperada de 50 Hz. Como se puede apreciar, la tensión pico es de unos 1200 voltios, lo que hace unos 850 voltios eficaces.

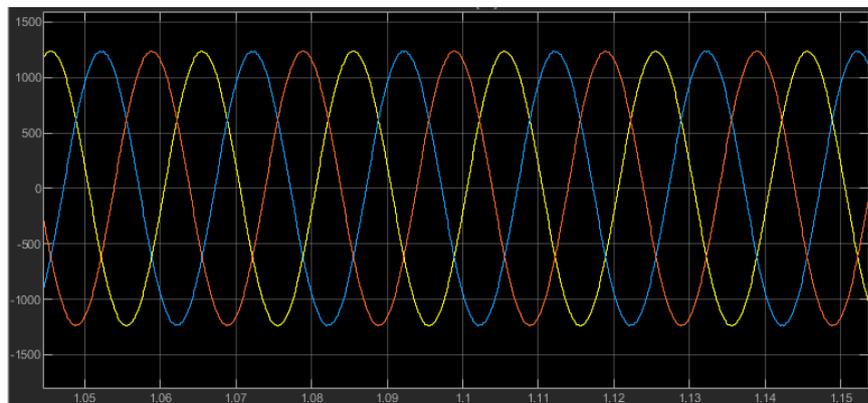


Figura 77 - Tensiones de salida del convertidor MMC en Simulink

En cuanto a las corrientes a la salida del convertidor, como se puede ver en la Figura 78, se observa cierto rizado, aunque, al no ser el objetivo del trabajo el

estudio del rizado y los armónicos para su eliminación, se considera aceptable, ya que la onda conserva su forma senoidal, la frecuencia de 50 Hz y la amplitud, de unos 160 amperios pico, que equivalen a 114 amperios eficaces. En cualquier caso, las corrientes también forman un sistema trifásico equilibrado.

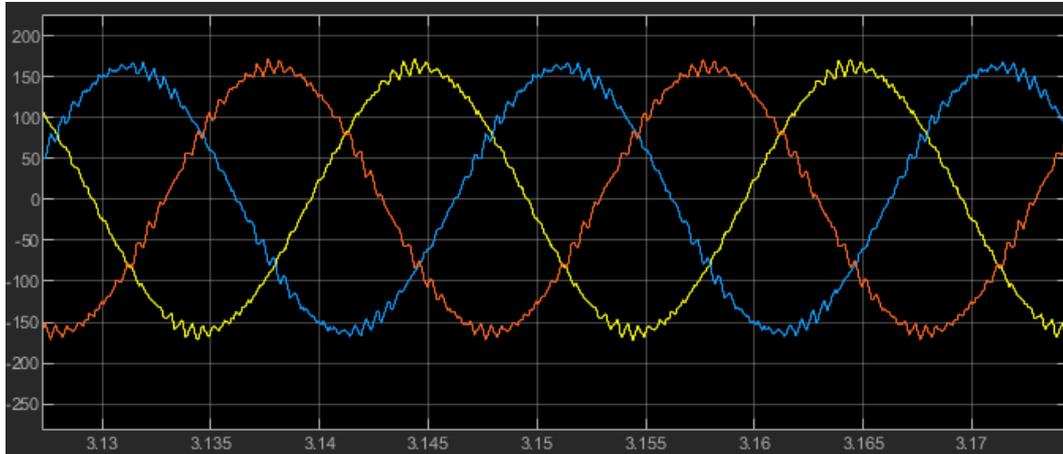


Figura 78 - Corrientes de salida del convertidor de Simulink

En la Figura 79 se puede ver la corriente que circula por la rama superior, la corriente que circula por la rama inferior y la suma de ambas, que corresponde a la corriente de salida de una fase del convertidor. Se puede ver que cada una de las corrientes de las ramas tienen componente de corriente continua. Componente positiva la correspondiente a la rama superior y negativa la de la rama inferior. Al sumar las dos corrientes la componente de corriente continua se anula.

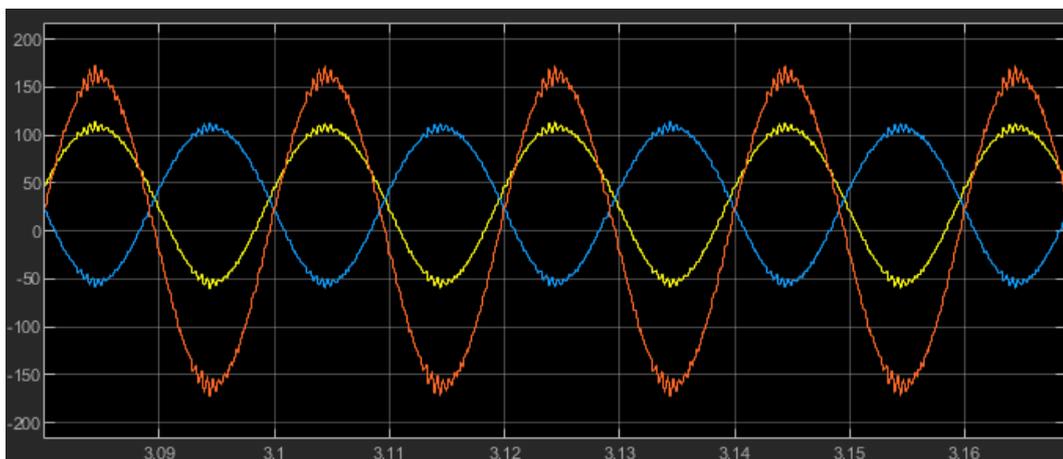


Figura 79 - Corriente de la rama superior (amarillo), de la rama inferior (azul) y la suma de ambas (rojo) del convertidor de Simulink

A partir de los datos observables en las gráficas de tensiones y corrientes de la Figura 77 y la Figura 78, se puede calcular la potencia del convertidor en esta simulación. Como se puede observar en la Ecuación 7, la potencia del convertidor de Simulink es de 288 kW.

Ecuación 7 - Cálculo potencia del convertidor de Simulink

$$P_{convertidor} = \frac{3}{2} V_{pico} \cdot I_{pico} = \frac{3}{2} 1200 V \cdot 160 A = 288 kW$$

En la Figura 80 se puede ver la evolución de las corrientes circulantes del convertidor de Simulink en los primeros 1,8 segundos de simulación. Como se puede observar, las oscilaciones tienden a converger en un punto en torno a los 30 A, lo que quiere decir que el convertidor tiene un comportamiento estable en el tiempo, por lo tanto, se puede concluir que las premisas utilizadas para el desarrollo del convertidor de Simulink se pueden aplicar en el simulador en tiempo real.

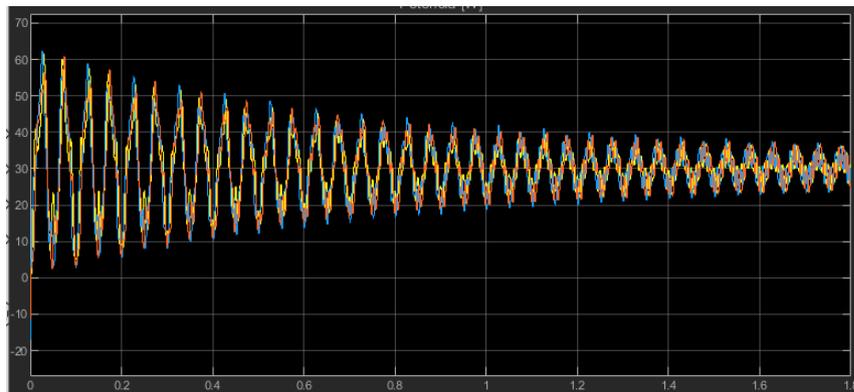


Figura 80 - Corrientes circulantes del convertidor MMC en Simulink

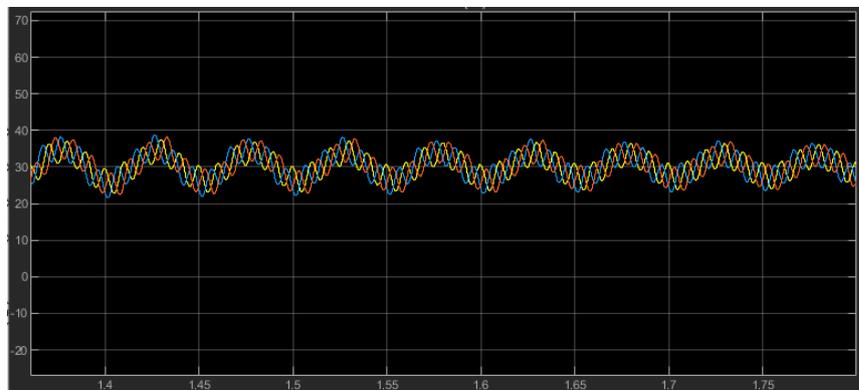


Figura 81 - Corrientes circulantes del convertidor MMC en Simulink (zoom)

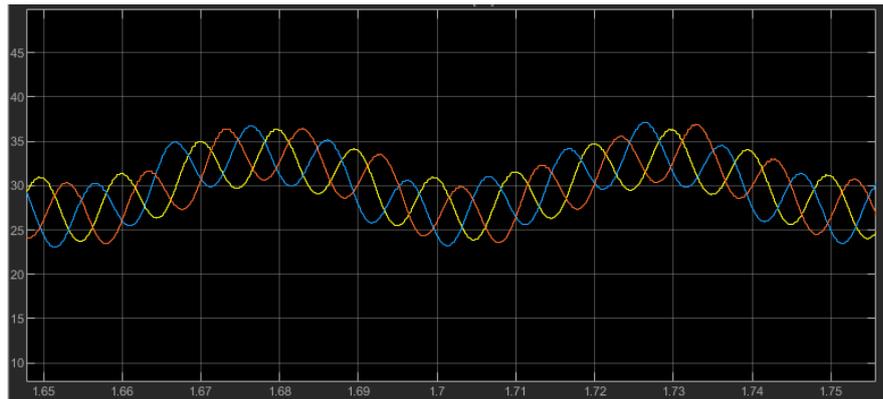


Figura 82 - Corrientes circulantes del convertidor en Simulink (zoom x 2)

En la Figura 81 se puede ver la evolución de las corrientes circulantes en el último medio segundo de simulación, donde las señales ya se encuentran bastante estabilizadas. Ampliando hasta poder ver un periodo completo, (Figura 82), se puede observar que estas corrientes circulantes tienen una frecuencia de 100 Hz, ya que en 20 milisegundos se observan dos máximos. Esta frecuencia se debe al uso de la técnica de PWM con portadoras *Phase Disposition*.

Con las corrientes circulantes es posible calcular la potencia que entra en cada fase desde las baterías, conociendo la tensión en el lado de corriente continua y la componente de corriente continua de las corrientes circulantes. En este caso la tensión en corriente continua es de 3200 V y la componente de corriente continua de las corrientes circulantes tiene un valor en torno a 30 amperios. Por lo tanto, como se muestra en la Ecuación 8, en cada fase entran 96 kW, por lo que las tres fases suman un total de 288 kW, que es exactamente la misma potencia que se ha calculado anteriormente en la Ecuación 7, por lo tanto, los cálculos son consistentes.

Ecuación 8 - Potencia que entra en cada fase desde DC en el convertidor de Simulink

$$P = V_{DC} \cdot I_{Z_{DC}} = 3200 \text{ V} \cdot 30 \text{ A} = 96 \text{ kW}$$

Como ya se ha comentado, el tratamiento de los armónicos y el rizado no es un objetivo del trabajo y, además, se requerirían más recursos de los disponibles para simular en tiempo real en el intento de mitigar dichos armónicos.

Una vez presentado, explicado y validado el funcionamiento del convertidor de Simulink, se deben implementar en el simulador en tiempo real los principios que se han utilizado en este convertidor. Para ello se deben tener en cuenta las restricciones propias de una simulación en tiempo real y las limitaciones existentes a causa de los recursos disponibles. Es por ello que al trasladar estas ideas al simulador en tiempo real, se deben tener en cuenta factores como el tiempo de ejecución o el número de variables utilizadas.

3.3.- Simulación en tiempo real del HVDC

3.3.1.- Planta

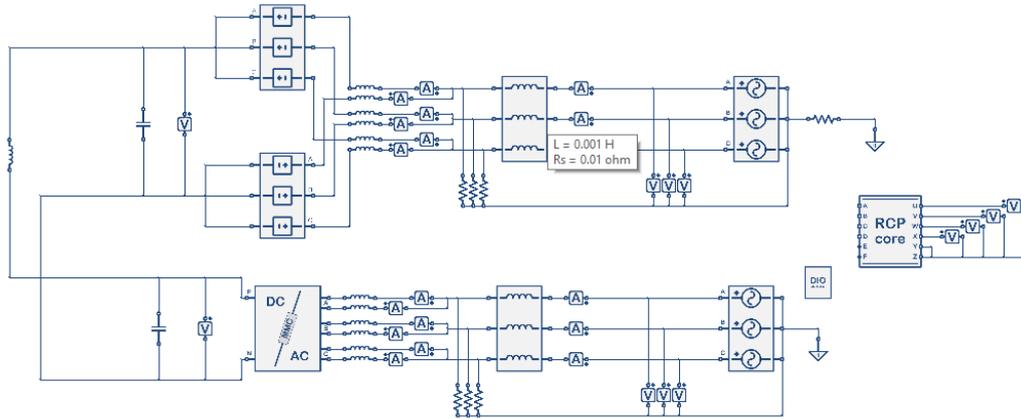


Figura 83 - Planta de la instalación HVDC en la simulación en tiempo real

La planta de la línea HVDC en el simulador en tiempo real se muestra en la Figura 83. Lo más destacable de la planta es el hecho de que un convertidor utiliza el bloque del convertidor MMC que se ha explicado en el punto 2.11.3.2.1.- Explicación bloque del MMC que viene incluido en el software, y, el otro convertidor está formado por fuentes de tensión trifásicas dependientes. Esto es debido a las limitaciones de recursos disponibles que se han mencionado en el apartado 3.1.- Presentación de la línea que se pretende simular, por las que no ha sido posible implementar en una sola FPGA un modelo que funcione con dos convertidores completos. Se recuerda que cada convertidor completo de cuatro celdas por rama, 24 celdas en total. Esto supone que:

- No hay suficientes *Digital Outputs* (36 en total), para poder configurar las 48 señales de PWM que serían necesarias.
- Un solo convertidor ocupa más de dos microsegundos de los 4 microsegundos como máximo que puede tardar el *core#1* en ejecutarse por completo, por lo tanto, si los dos convertidores fueran realistas se sobrepasaría este tiempo límite.

Por lo tanto, para poder simular la línea HVDC con dos convertidores utilizando los recursos disponibles, se ha recurrido como solución a la elaboración de un convertidor promediado que, utilizando menos recursos, tiene un comportamiento muy similar al del convertidor completo. A partir de ahora, se llamará “convertidor realista” al convertidor completo que utiliza el bloque MMC y, se llamará “convertidor promediado” a este nuevo convertidor con el que se consiguen ahorrar recursos.

Por lo tanto, dentro del controlador, el bloque RCPcore, las variables relacionadas exclusivamente con el convertidor realista se nombran con la etiqueta “*mmc01*”, mientras que las variables correspondientes al convertidor promediado se referencian con “*mmc02*”.

El convertidor promediado se explica en detalle más adelante en el apartado [3.3.5.1.- Explicación del convertidor promediado.](#)

Viendo la planta de la Figura 83, también destacan las tres resistencias en paralelo entre las bobinas de las ramas superior e inferior del convertidor y las bobinas de salida del convertidor en el lado de alterna. En la realidad no existirían, pero el simulador presenta problemas al colocar las bobinas de esta manera, porque es como colocar fuentes de corriente en serie.

3.3.2.- Controlador

El controlador es el bloque RCPcore, que permite programar las conmutaciones en las celdas de los convertidores, leer datos de la planta, poner reguladores, programar fuentes de tensión o de corriente programables. Este bloque es dirigido por el *core#2*. Cuenta con varias pestañas, una de configuración, la pestaña “*Global*”, la pestaña “*setup*” y la pestaña “*loop*”, tal como se ha mencionado más atrás en el apartado [2.11.3.2.2.- Explicación bloque RCPcore.](#)

Como no es posible realizar los cálculos de ambos convertidores simultáneamente, se ha decidido que, en primer lugar, se realizan los cálculos del convertidor realista, el *mmc01* y, después, los del promediado, el *mmc02*. El código para el funcionamiento de ambos convertidores es prácticamente el mismo, salvo por algunas diferencias, por lo que se explicarán las partes generales del código que son comunes para los dos convertidores utilizando solamente el código del convertidor *mmc01*, y, en los puntos en los que haya diferencias se explicarán de forma detallada en el apartado del convertidor promediado. Además, hay partes del código que se repiten para las tres fases o para la rama superior e inferior. En estos casos se detalla el código para uno de los casos. El código completo queda recogido en el anexo.

3.3.2.1.- Global

La pestaña *Global* comienza con los *#define*, que es un comando que sustituye un texto por otro.

En primer lugar, como se puede ver en la Figura 84, se encuentra el *#define TPWM 400* que sirve para configurar el periodo en microsegundos de las portadoras del PWM. Corresponde a la suma de dos periodos de ejecución del controlador, es decir, un periodo del *loop* como rampa ascendente más otro periodo como rampa descendente (\wedge) o, al revés, primero como rampa descendente y luego como ascendente (\vee), igualmente suman dos periodos

de ejecución del controlador. Este número se requerirá como argumento de la función `pwmConfig ()`.

En segundo lugar, `#pragma GETTIME OFF`, sirve para que el simulador en tiempo real no trate de estimar la duración de la función `loop ()`, ya que al haber condicionales `if` y bucles `while`, es complicado estimar la duración.

```
// Global

#define TPWM 400

#pragma GETTIME OFF

#define MMC_PROMEDIADO
#define MMC_REALISTA

#define _MMC01_SLAVE
#define MMC01_MASTER
```

Figura 84 - Código del Global (1/9)

A continuación, como se puede ver en la Figura 84, los siguientes `#define` son `MMC_PROMEDIADO` y `MMC_REALISTA`, para dividir las partes del programa que pertenecen al convertidor promediado y las partes propias del convertidor realista, gracias al comando `#ifdef`, que solo ejecuta su contenido cuando se cumple la condición `#ifdef`. De esta manera si no está definido uno de los convertidores no se ejecutan sus líneas del programa, permitiendo el estudio individual de ambos convertidores.

De la misma manera se define el modo de funcionamiento, *Master* o *Slave*, mediante `#define MMC01_Slave` o `MMC01_MASTER` (ver Figura 84), cuando se encuentre definido que el convertidor realista es *Master*, el promediado será *Slave* y, por el contrario, si el realista está en modo *Slave*, el convertidor promediado funcionará en modo *Master*. El funcionamiento de los modos *Master* y *Slave* de los convertidores se detallan más adelante en el apartado [3.3.3.- Explicación funcionamiento Master y Slave](#).

En la Figura 85 se puede ver que hay un `#define` para cada celda, es decir, a cada celda se le da un nombre según si es UP o LW, está en la fase A, B o C y número de celda 1, 2, 3 o 4. Por ejemplo, `UP_A_1`. A estos nombres se les asigna en orden un número desde el 1 hasta el 24 mediante `#define`. De esta forma se podrá identificar fácilmente dentro de un vector de 24 componentes, la posición que corresponde el valor de cualquier celda.

```

#define UP_A_1      1
#define UP_A_2      2
#define UP_A_3      3
#define UP_A_4      4

#define LW_A_1      5
#define LW_A_2      6
#define LW_A_3      7
#define LW_A_4      8

#define UP_B_1      9
#define UP_B_2     10
#define UP_B_3     11
#define UP_B_4     12

#define LW_B_1     13
#define LW_B_2     14
#define LW_B_3     15
#define LW_B_4     16

#define UP_C_1     17
#define UP_C_2     18
#define UP_C_3     19
#define UP_C_4     20

#define LW_C_1     21
#define LW_C_2     22
#define LW_C_3     23
#define LW_C_4     24

```

Figura 85 - Código del Global (2/9)

A continuación, están los `#define` para utilizar las variables temporales, que se pueden ver en la Figura 86. Hay definidas seis variables temporales: `temp_1`, `temp_2`, `temp_3`, `temp_4`, `temp_5` y `temp_6`. Mediante `#define` se les da a estas variables un nombre que describe mejor el uso que se les va a dar. La utilización de variables temporales viene forzada por el límite de 96 variables globales y 32 Variables locales y argumentos de funciones. Entonces, con el objetivo de utilizar menos variables, se han definido 6 variables temporales que se reutilizan varias veces para sustituir variables que no necesitan conservar su valor de una iteración a otra.

```

#define DUTYCYCLE_1      temp_1
#define DUTYCYCLE_2      temp_2
#define DUTYCYCLE_3      temp_3
#define DUTYCYCLE_4      temp_4

#define V_A_REF           temp_3
#define V_B_REF           temp_4
#define V_C_REF           temp_5

#define ESTADO_ACTUAL_A_UP temp_1
#define ESTADO_ACTUAL_B_UP temp_2
#define ESTADO_ACTUAL_C_UP temp_3
#define ESTADO_ACTUAL_A_LW temp_4
#define ESTADO_ACTUAL_B_LW temp_5
#define ESTADO_ACTUAL_C_LW temp_6

// Para calculo de tensión rama convertidor promediado
#define V_A_UP_PROMED     temp_1
#define V_B_UP_PROMED     temp_2
#define V_C_UP_PROMED     temp_3
#define V_A_LW_PROMED     temp_4
#define V_B_LW_PROMED     temp_5
#define V_C_LW_PROMED     temp_6

// Variables corrientes circulantes
#define IZ_A              temp_1
#define IZ_B              temp_2
#define IZ_C              temp_3
#define IZ                 temp_4

```

Figura 86 - Código del Global (3/9)

Después, se encuentran los *#define* cuyo texto es sustituido por una lectura de un *Analog Output*. Los *Analog Output* se ponen para trasladar la medida de un amperímetro o de un voltímetro, desde la planta hasta el controlador, para poder programar o regular según el valor que proporciona. En la Figura 87 se pueden ver estos *#define* para el convertidor realista. En el caso del convertidor promediado sería igual, pero con los *Analog Output* del 11 al 20.

```

#ifdef MMC_REALISTA
#define I_UP_A_MMC01     readAO(1)
#define I_LW_A_MMC01     readAO(2)
#define I_UP_B_MMC01     readAO(3)
#define I_LW_B_MMC01     readAO(4)
#define I_UP_C_MMC01     readAO(5)
#define I_LW_C_MMC01     readAO(6)

#define V_A_GRID_MMC01   readAO(7)
#define V_B_GRID_MMC01   readAO(8)
#define V_C_GRID_MMC01   readAO(9)

#define VDC_MMC01       readAO(10)
#endif

```

Figura 87 - Código del Global (4/9)

Tras estos, están los *#define* que sustituyen el texto por el valor de una constante. Es decir, todas las constantes que se necesitan están definidas en

este punto. Los valores de las constantes se pueden ver en la Figura 88 y en la Figura 89.

```

#define V_DC_HALF          1600

#define INSERTED           1
#define BYPASS             0

#define INV_PERIOD         5000 // 1/200us //4000 // 1/250 us
#define FREC_REF           314.1592654 // 2 x pi x 50 = 314.15926

#define WXL_EQ_MMC01      0.39269908 // W = 2*pi*50; Leq = Lo +
0.5 * Larm
#define WXL_EQ_MMC02      0.39269908 // W = 2*pi*50; Leq = Lo +
0.5 * Larm

#define K_TS_CSM           0.00166666666666 //Ts/(n°celdas*Csm)
n°celdas=4

```

Figura 88 - Código del Global (5/9)

```

#ifdef MMC_REALISTA
#define KP_SYNC_MMC01      1
#define TS_KI_SYNC_MMC01  0.02

#define KP_IODQ_MMC01      0.2
#define TSXKI_IODQ_MMC01  0.002 // Ts * Ki_iod //400 // 200us

#define KPV_DC_MMC01       1
#define TSKIV_DC_MMC01    0.0012
#endif

#ifdef MMC_PROMEDIADO
#define KP_SYNC_MMC02      1
#define TS_KI_SYNC_MMC02  0.02

#define KP_IODQ_MMC02      0.2
#define TSXKI_IODQ_MMC02  0.002 // Ts * Ki_iod //400 // 200us

#define KPV_DC_MMC02       0.6
#define TSKIV_DC_MMC02    0.0025
#endif

#define PI_LIMIT_A         666 // 1000kW / 1200V / 1.5 +20%

#define PERIODO             200e-6

#define P_REF               -1000000

```

Figura 89 - Código del Global (6/9)

El bloque del convertidor MMC, como se ha mencionado en el apartado 2.11.3.2.1.- Explicación bloque del MMC, genera vectores con el nombre de la etiqueta que se le ha puesto, en este caso mmc01. Para que el controlador reciba estos valores, se definen esos vectores en esta parte de global que aparece en la Figura 90. Seis vectores de cuatro componentes para almacenar las 24 tensiones de las celdas del convertidor.

A continuación, como se puede ver en la Figura 90, se definen las variables temporales, que son variables globales que, como se ha explicado antes, se reutilizan varias veces en el código, de tal forma que se ahorran variables.

Después, se definen el resto de las variables globales, que se muestran en la Figura 90 y en la Figura 91, las variables vectoriales, que se ven en la Figura 92 y, finalmente, las funciones *posVmin* y *posVmax*, que son las que buscan qué celda debe encenderse o apagarse en cada iteración. Estas funciones se ven en profundidad en el apartado siguiente.

```
// Para la lectura de variables
//

double mmc01_phA_Vup[4];
double mmc01_phA_Vlw[4];

double mmc01_phB_Vup[4];
double mmc01_phB_Vlw[4];

double mmc01_phC_Vup[4];
double mmc01_phC_Vlw[4];

// Variables temporales
double temp_1, temp_2, temp_3, temp_4, temp_5, temp_6;

//
// Global variables
//

int    syncP, syncN;

double Va_up_ref, Vb_up_ref, Vc_up_ref;
double Va_lw_ref, Vb_lw_ref, Vc_lw_ref;

double Vdc_ref;
double integralVdc;
```

Figura 90 - Código del Global (7/9)

```

double Vcond_UP_A_1, Vcond_UP_A_2, Vcond_UP_A_3, Vcond_UP_A_4,
       Vcond_LW_A_1, Vcond_LW_A_2, Vcond_LW_A_3, Vcond_LW_A_4;

double Vcond_UP_B_1, Vcond_UP_B_2, Vcond_UP_B_3, Vcond_UP_B_4,
       Vcond_LW_B_1, Vcond_LW_B_2, Vcond_LW_B_3, Vcond_LW_B_4;

double Vcond_UP_C_1, Vcond_UP_C_2, Vcond_UP_C_3, Vcond_UP_C_4,
       Vcond_LW_C_1, Vcond_LW_C_2, Vcond_LW_C_3, Vcond_LW_C_4;

double porcentaje_A_UP, porcentaje_B_UP, porcentaje_C_UP,
       porcentaje_A_LW, porcentaje_B_LW, porcentaje_C_LW;

double V_A_UP_ave, V_B_UP_ave, V_C_UP_ave, V_A_LW_ave,
       V_B_LW_ave, V_C_LW_ave;

double IupA, IupB, IupC;
double IlwA, IlwB, IlwC;

//
// Variables de sincronización con la RED
//

double V_d_ref, V_q_ref, integral_Vq;
double theta;
double seno_mmc01, coseno_mmc01;
double seno_mmc02, coseno_mmc02;
double w;

double Io_d, Io_q;
double Io_d_ref, Io_q_ref;
double integral_Iod, integral_Ioq;
double Vo_d_ref_conv, Vo_q_ref_conv;

```

Figura 91 - Código del Global (8/9)

```

//
// Variables vectoriales
//

double pS_mmc01[25];
double dutyCycle_mmc01[25];
double valores_mmc01[12];

double pS_mmc02[25];
double dutyCycle_mmc02[25];
double valores_mmc02[12];

double fuente_externa_UP[3];
double fuente_externa_LW[3];

```

Figura 92 - Código del Global (9/9)

3.3.2.1.1.- Funciones *posVmin* *posVmax*

Las funciones *posVmin* y *posVmax* son el equivalente a las funciones *maxV* y *minV* en el controlador en Matlab. El criterio para conmutar las celdas es el mismo, por lo que el comportamiento es similar. Pero existen algunas diferencias en cuanto al código.

La gran diferencia entre estas funciones es que mientras en Matlab las funciones elegían la celda entre la 1, la 2, la 3 y la 4, en este caso el número de celda seleccionada es la posición correspondiente a la primera celda de la rama, más 0 en caso de seleccionar la primera, más 1 si es la segunda, más 2 si es la tercera y más 3 si es la cuarta. Se recuerda que a cada celda se le ha puesto un nombre que, mediante un *#define* ha pasado a ser un número que marca la posición dentro de un vector de 25 componentes ignorando la primera [0].

A continuación, se procede a explicar el funcionamiento de las funciones *posVmin* y *posVmax*. Al igual que ocurría en el convertidor de Matlab, estas funciones funcionan de forma similar pero opuesta. La función *posVmin* se puede ver en la Figura 93 y la función *posVmax* en la Figura 94.

Como argumentos, estas funciones tienen la variable “ESTADO_BUSCADO”, para indicar si la función debe buscar entre las insertadas o las no insertadas (*INSERTED* o *BYPASS* en *#define*), las tensiones de los condensadores en las cuatro celdas de la rama correspondiente y el estado de los polos de potencia. Como se puede ver, estas funciones tienen los mismos argumentos que las del convertidor de Matlab. En el caso del simulador en tiempo real, estos nueve argumentos ocupan espacio como variables globales, por lo que se debe tener en cuenta la reducción del número de este tipo de variables.

Al inicio de la función se define un valor mínimo de valor “infinito” para la función *Vmin* y un valor máximo de valor “menos infinito” para la *Vmax*. Este valor se ha fijado en 1600, ya que es un número que, frente al número de celdas, se puede considerar suficientemente grande, y, es una constante que ya se utiliza para la mitad de la componente de tensión en continua, por lo que se puede ahorrar una constante.

También se declara la variable “*pos*” y se inicializa como -1, tanto en la función *posVmin* como en la función *posVmax*.

De la misma manera que en las funciones homólogas del convertidor de Matlab, se comienza a buscar la celda que debe ser seleccionada, partiendo de la celda número 1 de la rama hasta la celda número 4. Para que una celda sea una posible candidata para ser seleccionada, debe cumplir dos condiciones:

- Que el estado de la celda sea igual al estado buscado (encendido o apagado).
- Que la tensión en el condensador sea mayor que el valor máximo “*Vmax*” en la función *posVmax* o que la tensión en el condensador sea menor que el valor mínimo “*Vmin*” en la función *posVmin*.

Cuando se cumplen las dos condiciones para que una celda sea candidata para ser seleccionada, la tensión de dicha celda pasa a ser el valor máximo en la función *posVmax* o el valor mínimo en el caso de la función *posVmin*. Esto es igual que en las funciones del convertidor de Matlab.

La principal diferencia con respecto a las funciones del convertidor de Matlab es la siguiente:

- En la celda número 1, la posición que se le asigna a la variable “*pos*” es 0, en lugar de 1.
- En la celda número 2, la posición que se le asigna a la variable “*pos*” es 1, en lugar de 2.
- En la celda número 3, la posición que se le asigna a la variable “*pos*” es 2, en lugar de 3.
- En la celda número 4, la posición que se le asigna a la variable “*pos*” es 3, en lugar de 4.

Esto es así porque, la celda finalmente elegida sale a partir de la posición del vector de la primera celda de cada rama más la posición que aporta la función que corresponda, *posVmax* o *posVmin*. Por lo tanto, si la posición 13 corresponde a la primera celda de la fase B de la rama inferior (*LW_B_1*):

- Si la celda seleccionada es la primera, al devolver la función la variable *pos* = 0, esta será la celda seleccionada.
- Si se selecciona la segunda celda, la función retorna *pos* = 1, por lo que la posición final será la $13 + 1 = 14$, es decir, la segunda celda de la fase B de la rama inferior (*LW_B_2*).
- Si se elige la tercera celda, la función devuelve *pos* = 2, entonces la posición seleccionada es la 15 que corresponde a la tercera celda de la fase B de la rama inferior (*LW_B_3*).
- Y, por último, si la celda seleccionada es la cuarta, la función aporta *pos* = 3, resultando ser seleccionada la posición 16, que corresponde a la cuarta celda de la fase B de la rama inferior (*LW_B_4*).

```

// Funciones para calcular posiciones de Vmin, Vmax
//

int posVmin (int estado_buscado, double Vcond_1, double Vcond_2,
double Vcond_3, double Vcond_4, double pS_1, double pS_2, double
pS_3, double pS_4)
{
    double Vmin;
    Vmin = 1600;

    int pos;
    pos = -1;

    if ((pS_1 == estado_buscado) && (Vcond_1 < Vmin)) {
        pos = 0;
        Vmin = Vcond_1;
    }

    if ((pS_2 == estado_buscado) && (Vcond_2 < Vmin)) {
        pos = 1;
        Vmin = Vcond_2;
    }

    if ((pS_3 == estado_buscado) && (Vcond_3 < Vmin)) {
        pos = 2;
        Vmin = Vcond_3;
    }

    if ((pS_4 == estado_buscado) && (Vcond_4 < Vmin)) {
        pos = 3;
        //Vmin = Vcond_4;
    }

    return pos;
}

```

Figura 93 - Función posVmin

```

int posVmax (int estado_buscado, double Vcond_1, double Vcond_2,
double Vcond_3, double Vcond_4, double pS_1, double pS_2, double
pS_3, double pS_4)
{
    double Vmax;
    Vmax = -1600;

    int pos;
    pos = -1;

    if ((pS_1 == estado_buscado) && (Vcond_1 > Vmax)) {
        pos = 0;
        Vmax = Vcond_1;
    }

    if ((pS_2 == estado_buscado) && (Vcond_2 > Vmax)) {
        pos = 1;
        Vmax = Vcond_2;
    }

    if ((pS_3 == estado_buscado) && (Vcond_3 > Vmax)) {
        pos = 2;
        Vmax = Vcond_3;
    }

    if ((pS_4 == estado_buscado) && (Vcond_4 > Vmax)) {
        pos = 3;
        //Vmax = Vcond_4;
    }

    return pos;
}

```

Figura 94 - Función posVmax

3.3.2.2.- Setup

En esta pestaña se inicializan las variables que así lo requieren. Es el punto de partida en cada simulación. Solo se ejecuta en la primera iteración.

En primer lugar, tal como se muestra en la Figura 95, se inicializan a cero las *Vuz*, *Vvz*, *Vwz*, *Vxz*, que son las ramas de medida del RCPcore, que se pueden observar tanto en la Figura 35 como en la Figura 83. En esos voltímetros del RCPcore se puede visualizar la variable que se necesite dándole el valor de esa variable a uno de estos puertos *Vuz*, *Vvz*, *Vwz*, *Vxz*, y luego colocando los visores en el voltímetro correspondiente.

```

// Setup

Vuz = 0;
Vvz = 0;
Vwz = 0;
Vxz = 0;

syncP = 1;
syncN = 0;

```

Figura 95 - Código del setup (1/5)

Después, se inicializa las señales de sincronización *SyncP* y *SyncN*, teniendo en cuenta que siempre tienen que ser opuestas.

A continuación, se inicializan los vectores del ciclo de servicio para el convertidor realista, referenciado con el nombre “*mmc01*” y para el convertidor promediado, referenciado con “*mmc02*”. Estos son dos vectores de 25 componentes en los que la posición [0] queda vacía, por lo que en la práctica es como si fueran vectores de 24 componentes. De esta manera las posiciones de los vectores cuadran con los números asignados mediante los *#define* a las celdas en la pestaña *Global*. En la Figura 96 se puede ver cómo se inicializan los vectores del ciclo de servicio del convertidor realista. Los vectores correspondientes al ciclo de servicio del convertidor promediado están definidos de la misma manera.

```
// setup() ciclo de servicio mmc01
dutyCycle_mmc01[UP_A_1] = 0;
dutyCycle_mmc01[UP_A_2] = 1;
dutyCycle_mmc01[UP_A_3] = 0;
dutyCycle_mmc01[UP_A_4] = 1;

dutyCycle_mmc01[LW_A_1] = 0;
dutyCycle_mmc01[LW_A_2] = 1;
dutyCycle_mmc01[LW_A_3] = 0;
dutyCycle_mmc01[LW_A_4] = 1;

dutyCycle_mmc01[UP_B_1] = 0;
dutyCycle_mmc01[UP_B_2] = 1;
dutyCycle_mmc01[UP_B_3] = 0;
dutyCycle_mmc01[UP_B_4] = 1;

dutyCycle_mmc01[LW_B_1] = 0;
dutyCycle_mmc01[LW_B_2] = 1;
dutyCycle_mmc01[LW_B_3] = 0;
dutyCycle_mmc01[LW_B_4] = 1;

dutyCycle_mmc01[UP_C_1] = 0;
dutyCycle_mmc01[UP_C_2] = 1;
dutyCycle_mmc01[UP_C_3] = 0;
dutyCycle_mmc01[UP_C_4] = 1;

dutyCycle_mmc01[LW_C_1] = 0;
dutyCycle_mmc01[LW_C_2] = 1;
dutyCycle_mmc01[LW_C_3] = 0;
dutyCycle_mmc01[LW_C_4] = 1;
```

Figura 96 - Código del setup (2/5)

También se inicializan los vectores que conservan el estado de los polos de potencia *pS_mmc01* y *pS_mmc02* (ver Figura 97), que también son de 25 componentes, pero la inicial, la [0], queda vacía, debido al mismo motivo que se ha explicado con los vectores del ciclo de servicio.

```

// setup() estado polos mmc01
pS_mmc01[UP_A_1] = 0;
pS_mmc01[UP_A_2] = 1;
pS_mmc01[UP_A_3] = 0;
pS_mmc01[UP_A_4] = 1;

pS_mmc01[LW_A_1] = 0;
pS_mmc01[LW_A_2] = 1;
pS_mmc01[LW_A_3] = 0;
pS_mmc01[LW_A_4] = 1;

pS_mmc01[UP_B_1] = 0;
pS_mmc01[UP_B_2] = 1;
pS_mmc01[UP_B_3] = 0;
pS_mmc01[UP_B_4] = 1;

pS_mmc01[LW_B_1] = 0;
pS_mmc01[LW_B_2] = 1;
pS_mmc01[LW_B_3] = 0;
pS_mmc01[LW_B_4] = 1;

pS_mmc01[UP_C_1] = 0;
pS_mmc01[UP_C_2] = 1;
pS_mmc01[UP_C_3] = 0;
pS_mmc01[UP_C_4] = 1;

pS_mmc01[LW_C_1] = 0;
pS_mmc01[LW_C_2] = 1;
pS_mmc01[LW_C_3] = 0;
pS_mmc01[LW_C_4] = 1;

```

Figura 97 - Código del setup (3/5)

En la Figura 98 se pueden ver las tensiones promedio en los condensadores del convertidor promediado, que se inicializan en 800 V porque es la tensión que deben tener en promedio los condensadores de las celdas. También se inicializan a cero todas las variables que se integran y contadores que deben mantener su valor para las próximas iteraciones.

```

V_A_UP_ave = 800;
V_B_UP_ave = 800;
V_C_UP_ave = 800;
V_A_LW_ave = 800;
V_B_LW_ave = 800;
V_C_LW_ave = 800;

ESTADO_PREVIO = 0;

theta = 0;
seno_mmc01 = sin(theta);
coseno_mmc01 = cos(theta);

seno_mmc02 = sin(theta);
coseno_mmc02 = cos(theta);

integral_Vq = 0;
w = 0;

Vdc_ref = 3200;

Io_d_ref = 0;
Io_q_ref = 0;

integralVdc = 0;

integral_Iod = 0;
integral_Ioq = 0;

```

Figura 98 - Código del setup (4/5)

Los vectores *valores_mmc01* y *valores_mmc02*, que se pueden ver inicializados en la Figura 99, son necesarios para que sea posible la simulación, ya que, un solo convertidor necesita prácticamente las 96 variables escalares globales para funcionar. Por lo tanto, como solución, se utilizan estos vectores para almacenar los datos que se necesitan en las próximas iteraciones y así poder usar las mismas variables globales en los dos convertidores. También se utilizan para tomar las medidas de ambos convertidores en un espacio de tiempo corto y que no difieran demasiado unas de otras. El modo en el que se utilizan estos vectores se muestra más en detalle más adelante en el punto [3.3.2.3.9.- Vectores de recuperación de valores](#). Lo que importa en el apartado actual es conocer que solamente se inicializan a cero las componentes desde la 7 a la 11, que son las componentes encargadas de guardar el valor de las variables que se integran y contadores que deben mantener su valor para las próximas iteraciones.

Los vectores de las fuentes de tensión programables, que componen el convertidor promediado, se inicializan en 1600 V cada componente, que es la mitad de la tensión de continua, y, por tanto, la tensión a la que va a estar cada rama del convertidor.

```

valores_mmc01[7] = 0; // integralVdc
valores_mmc01[8] = 0; // theta
valores_mmc01[9] = 0; // integralVq
valores_mmc01[10] = 0; // integral_Iod
valores_mmc01[11] = 0; // integral_Ioq

valores_mmc02[7] = 0; // integralVdc
valores_mmc02[8] = 0; // theta
valores_mmc02[9] = 0; // integralVq
valores_mmc02[10] = 0; // integral_Iod
valores_mmc02[11] = 0; // integral_Ioq

fuente_externa_UP[0] = 1600;
fuente_externa_UP[1] = 1600;
fuente_externa_UP[2] = 1600;

fuente_externa_LW[0] = 1600;
fuente_externa_LW[1] = 1600;
fuente_externa_LW[2] = 1600;

```

Figura 99 - Código del setup (5/5)

3.3.2.3.- Loop

Tras la pestaña de *setup*, se encuentra la pestaña *loop*, que se va a ejecutar cada 200 microsegundos, como indica el *loop period*. En esta parte del RCPcore es donde se programa toda la actividad de ambos convertidores.

3.3.2.3.1.- PWMConfig

Dentro de la pestaña *loop*, la primera parte del bucle consiste en generar las salidas de PWM. Estos cálculos solo son necesarios para el convertidor realista. No se han colocado al inicio del programa por casualidad, sino que se busca que la generación de las portadoras se produzca en el mismo instante en cada periodo. La generación de PWM se hace a partir de la función embebida *pwmConfig* () cuyo funcionamiento se ha explicado en el punto 2.11.3.3.- Funciones embebidas ya incorporadas en el código.

A la función *pwmConfig* se le debe aportar el ciclo de servicio, pero no se coge el valor directamente del vector *dutyCycle_mmc01*, sino que se vuelcan a las variables temporales *DUTYCYCLE_1*, *DUTYCYCLE_2*, *DUTYCYCLE_3*, *DUTYCYCLE_4* mediante la función *readVector* y son estas variables escalares las que se utilizan para la generación de PWM. El motivo para no usar las variables vectoriales directamente es la necesidad de ahorro de tiempo, ya que es más rápido leer el valor de una variable escalar que el de una variable vectorial.

Como se puede ver en la Figura 100, el volcado de valores del vector a las variables temporales se realiza de cuatro en cuatro, es decir, se leen las cuatro primeras correspondientes a la fase A superior y se generan las portadoras, seguido se leen las cuatro siguientes correspondientes a la fase A inferior y se generan las portadoras y así sucesivamente hasta haber generado todas las

portadoras. El motivo para hacerlo de esta manera es minimizar el “jitter”. Se llama “jitter” a la diferencia temporal que se da entre la generación de las portadoras entre unas ramas y otras. Para que el funcionamiento sea correcto el “jitter” debe ser el mínimo posible, ya que no se puede eliminar al no ser posible realizar todas las operaciones de forma simultánea.

```
// Generar salidas de PWM sincronizadas
#ifdef MMC_REALISTA

readVector(dutyCycle_mmc01, 1, DUTYCYCLE_1, DUTYCYCLE_2, DUTYCYCLE_3, DUTYCYCLE_4);

pwmConfig(UP_A_1, TPWM, syncP, DUTYCYCLE_1);
pwmConfig(UP_A_2, TPWM, syncP, DUTYCYCLE_2);
pwmConfig(UP_A_3, TPWM, syncP, DUTYCYCLE_3);
pwmConfig(UP_A_4, TPWM, syncP, DUTYCYCLE_4);

readVector(dutyCycle_mmc01, 5, DUTYCYCLE_1, DUTYCYCLE_2, DUTYCYCLE_3, DUTYCYCLE_4);

pwmConfig(LW_A_1, TPWM, syncN, DUTYCYCLE_1);
pwmConfig(LW_A_2, TPWM, syncN, DUTYCYCLE_2);
pwmConfig(LW_A_3, TPWM, syncN, DUTYCYCLE_3);
pwmConfig(LW_A_4, TPWM, syncN, DUTYCYCLE_4);
```

Figura 100 - Código del loop. Configuración de PWM

3.3.2.3.2.- Variables locales

Tras la generación de PWM, se definen las variables locales. El uso de variables locales solamente se debe a la limitación de 96 variables globales. Estas variables se definen en cada periodo y consumiendo algo de tiempo, pero la limitación del número de variables es más importante que la del tiempo en esta simulación.

Las variables locales que se han definido se pueden ver en la Figura 101. Estas variables representan:

- Las corrientes en el lado de alterna.
- Seis variables para almacenar las lecturas de tensión de la red. Tres para cada convertidor.
- La variable V_{dc} para almacenar la tensión en el lado de continua.

```
// Definición de variables locales

double Io_a, Io_b, Io_c;

double Va_grid_mmc01, Vb_grid_mmc01, Vc_grid_mmc01;
double Va_grid_mmc02, Vb_grid_mmc02, Vc_grid_mmc02;

double V_dc;
```

Figura 101 - Código del loop. Definición de variables locales

3.3.2.3.3.- Lectura de amperímetros y voltímetros

Después de definir las variables locales, en esta parte del programa, se realizan todas las medidas de corriente y de tensión para todo el periodo, con el objetivo

de que todas las medidas con las que se realizan cálculos sean lo más cercanas entre sí, ya que no se pueden leer todas paralelamente.

En primer lugar, se leen las medidas del convertidor promediado y estas medidas se vierten al vector *valores_mmc02*, tal como se observa en la Figura 102. A continuación, con las mismas variables se leen las medidas relacionadas con el convertidor realista, como se muestra en la Figura 103. El motivo de hacerlo de esta manera es porque primero se hacen los cálculos del convertidor realista y después los del promediado. Entonces, para reutilizar el mayor número de variables posible en los dos convertidores, se utilizan unos vectores (*valores_mmc01* y *valores_mmc02*) para almacenar los datos, guardarlos de una iteración a otra y recuperarlos cuando se necesiten.

También se vierten los valores de tensión en los condensadores del convertidor realista a variables escalares (ver Figura 104), para ello también se utiliza la función *readVector*. Estos valores de tensión los aportan los seis vectores que genera el bloque del convertidor realista, que tienen la etiqueta *mmc01*.

El convertidor promediado calcula el promedio de las tensiones en los condensadores, por lo que no es necesario hacer ninguna medición adicional para ello.

```
#ifdef MMC_PROMEDIADO
// PRIMERO: leemos las corrientes del 2º mmc
// Lectura de amperímetros del mmc02
IupA = I_UP_A_MMC02;
IlwA = I_LW_A_MMC02;
IupB = I_UP_B_MMC02;
IlwB = I_LW_B_MMC02;
IupC = I_UP_C_MMC02;
IlwC = I_LW_C_MMC02;

// Lectura de la tensión en continua
V_dc = VDC_MMC02;

// Lectura tensiones red
Va_grid_mmc02 = V_A_GRID_MMC02;
Vb_grid_mmc02 = V_B_GRID_MMC02;
Vc_grid_mmc02 = V_C_GRID_MMC02;

// meter valores en vector.

writeVector(valores_mmc02, 0, IupA, IlwA, IupB, IlwB, IupC, IlwC, V_dc);

#endif
```

Figura 102 - Código del loop. Lectura de valores convertidor promediado

```

#ifdef MMC_REALISTA
// DESPUES: leer corrientes del primer mmc
IupA = I_UP_A_MMC01;
IlwA = I_LW_A_MMC01;
IupB = I_UP_B_MMC01;
IlwB = I_LW_B_MMC01;
IupC = I_UP_C_MMC01;
IlwC = I_LW_C_MMC01;

// Lectura de la tensión en continua
V_dc = VDC_MMC01;

// Lectura tensiones red
Va_grid_mmc01 = V_A_GRID_MMC01;
Vb_grid_mmc01 = V_B_GRID_MMC01;
Vc_grid_mmc01 = V_C_GRID_MMC01;

#endif

```

Figura 103 - Código del loop. Lectura de valores convertidor realista

```

// Lectura tensiones de los condensadores
#ifdef MMC_REALISTA

readVector(mmc01_phA_Vup, 0, Vcond_UP_A_1, Vcond_UP_A_2, Vcond_UP_A_3, Vcond_UP_A_4);
readVector(mmc01_phA_Vlw, 0, Vcond_LW_A_1, Vcond_LW_A_2, Vcond_LW_A_3, Vcond_LW_A_4);

readVector(mmc01_phB_Vup, 0, Vcond_UP_B_1, Vcond_UP_B_2, Vcond_UP_B_3, Vcond_UP_B_4);
readVector(mmc01_phB_Vlw, 0, Vcond_LW_B_1, Vcond_LW_B_2, Vcond_LW_B_3, Vcond_LW_B_4);

readVector(mmc01_phC_Vup, 0, Vcond_UP_C_1, Vcond_UP_C_2, Vcond_UP_C_3, Vcond_UP_C_4);
readVector(mmc01_phC_Vlw, 0, Vcond_LW_C_1, Vcond_LW_C_2, Vcond_LW_C_3, Vcond_LW_C_4);

#endif // MMC_REALISTA

```

Figura 104 - Código del loop. Lectura de tensiones de los condensadores

3.3.2.3.4.- Señal de sincronización

Es uno de los argumentos que necesita la función *pwmConfig* para saber si debe haber un flanco ascendente (/) o un flanco descendente (\) en las portadoras. A diferencia del convertidor en Matlab, en este caso la señal de sincronización es la misma para todas las ramas, ya que calcular una señal de sincronización independiente para cada rama consumiría tiempo y demasiados recursos y apenas tendría efectos sobre la simulación.

Las variables para la sincronización se habían definido como *SyncP = 1* y *SyncN = 0*. Para programar un *Phase Disposition*, las señales de sincronización deberían ir alternándose entre 0 y 1. Por ello se ha programado de la manera en que se muestra en la Figura 105.

```

// Generación señal de sincronización
syncP = syncN;
syncN = (syncN == 0);

```

Figura 105 - Código del loop. Señal de sincronización

De esta forma, $SyncP$ cogerá el valor anterior de $SyncN$, y, $SyncN$ valdrá 1 cuando la igualdad $SyncN == 0$ se cumpla, porque será true, y valdrá 0 cuando no se cumpla la igualdad.

Este método es muy robusto, ya que es menos propenso a fallos y en caso de haber alguno se puede solucionar rápidamente en las siguientes iteraciones.

3.3.2.3.5.- Sincronización con la red

Para conectar el convertidor a la red sin problemas, es necesario que el convertidor se encuentre sincronizado con la red. Con este fin de realizar la sincronización de forma correcta se ha utilizado el método del dqPLL. Este sistema consiste en pasar de un sistema trifásico con ejes fijos a un sistema de ejes móviles d y q. De esta forma se consigue el valor del ángulo de fase de la tensión de la red con la máxima precisión, siendo posible imponerle al convertidor esos valores como referencia.

Para implementar el dqPLL en el controlador, se ha utilizado la función $uvw2dq$ que se ha explicado más atrás en el apartado 2.11.3.3.- Funciones embebidas ya incorporadas en el software. Tal como se muestra en el diagrama de bloques de la Figura 106, en el que se puede ver el funcionamiento del método del dqPLL, con la transformación de la tensión de la red de un sistema de ejes fijos a un sistema de dos ejes móviles, se obtiene una componente V_d equivalente al módulo de la tensión y otra componente V_q , con la que se puede obtener el ángulo de fase con la ayuda de un regulador proporcional-integral (PI). El objetivo es que, con este ángulo Θ obtenido a partir de aplicar el dqPLL a la tensión de la red, se consigan las referencias de tensión del convertidor sincronizadas con la red.

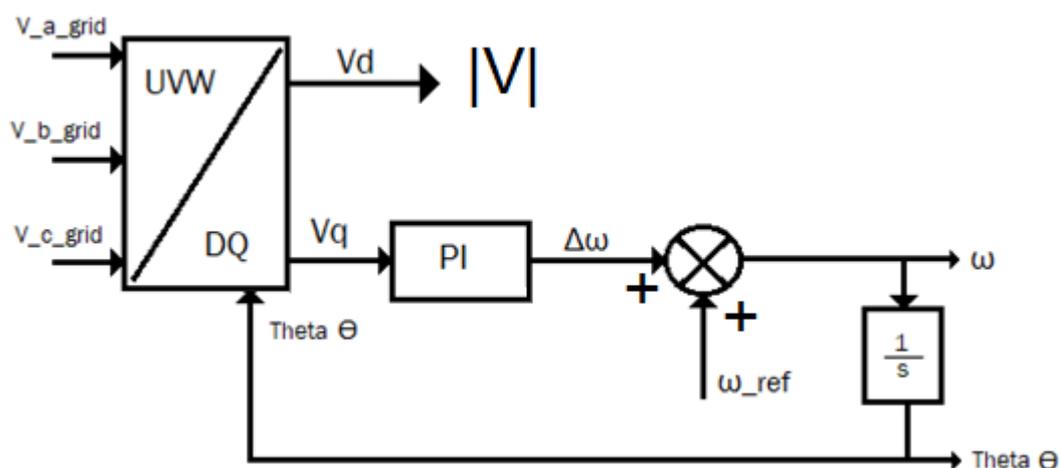


Figura 106 - Diagrama de bloques del dq PLL (elaboración propia)

Por lo tanto, con este objetivo, se miden las tensiones simples de la red, se envían al RCPcore mediante los *Analog Output* y se pasan a ejes móviles d y q a través de la función embebida $uvw2dq$, como se muestra en la Figura 107. Y,

como ya se ha mencionado, se obtiene el ángulo de fase de la red, y, regulando la componente V_q mediante un regulador proporcional integral se consigue la ω existente. Al integrar esta ω , se obtiene el ángulo Θ , con el que se cierra el lazo.

```
// Sincronizacion con la RED, DQ_PLL
uvw2dq (Va_grid_mmc01, Vb_grid_mmc01, Vc_grid_mmc01,
        seno_mmc01, coseno_mmc01, V_d_ref, V_q_ref);
```

Figura 107 - Código del loop. Aplicar dqPLL a las tensiones de la red

Para aplicar el regulador proporcional integral es necesario integrar, así como para obtener Θ a partir de ω . Pero no es posible programar estas integrales en el controlador con lenguaje C mediante funciones de transferencia como en Matlab y Simulink, por lo que el para integrar en lenguaje C se utiliza el razonamiento que se muestra en la Ecuación 9.

Ecuación 9 – Método de integración en lenguaje C. Razonamiento

$$\int V_q dV_q = V_q + \Delta V_q \cdot T_s$$

Bajo este razonamiento, para integrar la variable V_q , se ha creado una variable “*integralVq*”, inicializada a 0 en el *setup*, cuya función es almacenar en cada periodo el valor de la integral y conservarlo para la siguiente iteración. Entonces la integración de la variable V_q del regulador PI, quedaría de la forma que se observa en la Ecuación 10.

Ecuación 10 – Método de integración en lenguaje C. Aplicado en el controlador

$$integralVq = integralVq + V_q_ref \cdot T_s$$

Después de calcular la integral se multiplicaría la variable V_q por la constante proporcional y la integral de V_q por la constante integral, con el objetivo de implementar la regulación del PI.

El procesador es más rápido ejecutando sumas que ejecutando multiplicaciones. La forma más eficiente de ejecutar operaciones con multiplicación es la siguiente:

Ecuación 11 – Estructura de cálculo de productos y sumas del procesador.

$$a = b \cdot c + d \cdot e + \dots + y \cdot z + \dots$$

Por este motivo, se ha decidido suprimir de la Ecuación 10 el periodo T_s , quedando esta línea como se muestra en la Ecuación 12.

Ecuación 12 – Cálculo de la integral de V_q sin multiplicar por el periodo.

$$integralVq = integralVq + V_q_ref$$

Entonces el periodo T_s se incluye dentro de la constante integral TS_KI_SYNC , como se puede ver en la Ecuación 13. De esta manera se ahorra el tiempo de hacer esta multiplicación.

Ecuación 13 - Cálculo de ω

$$\omega = k_p \cdot V_{q_{ref}} + TS_KI_SYNC \cdot integralVq + FREC_REF$$

La integral de ω es theta (Θ), por lo tanto, utilizando el mismo razonamiento de la Ecuación 9, se obtiene el ángulo theta. En la Figura 108 se muestra la forma en la que se ha implementado esto en el código del controlador.

Al integrar de esta forma, el ángulo theta se iría a infinito y este procesador solo opera con valores entre cero y 2π . Por este motivo se utiliza la función *wrapToPI* que se observa en la última línea de la Figura 108, que devuelve el valor de la variable theta al intervalo entre 0 y 2π .

```
integral_Vq = integral_Vq + V_q_ref;
w = KP_SYNC_MMC01 * V_q_ref + TS_KI_SYNC_MMC01 * integral_Vq +
FREC_REF;
theta = theta + loopPeriod() * w;
theta = wrapToPI(theta);
```

Figura 108 - Código del loop. Obtención del ángulo theta

Para comprobar si la sincronización del dqPLL ha funcionado, se compara la tensión de la red con la tensión del convertidor aislado de la red, pero con las referencias extraídas del dqPLL. Al comparar estas tensiones, que se muestran en la Figura 109, se observa que ambas son similares, pero existe un pequeño desfase entre ellas. Podría valer para establecer la conexión a la red, pero podría generar transitorios bruscos a la hora de conectar.

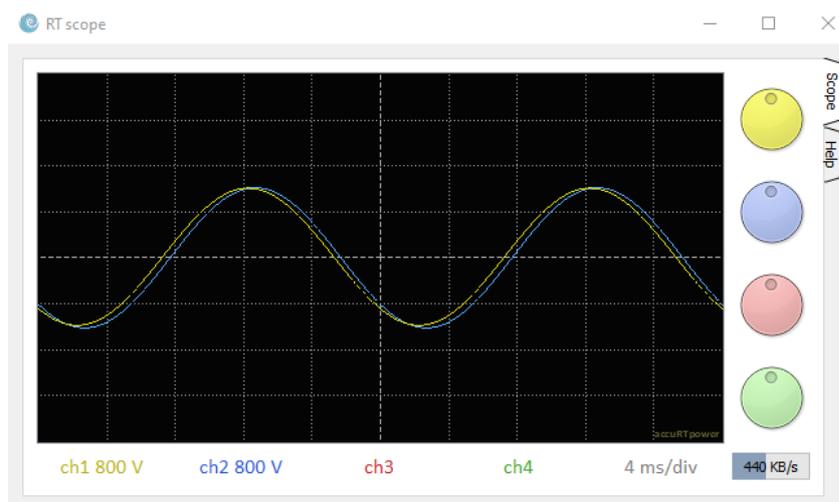


Figura 109 - Desfase entre la tensión de la red (en amarillo) y la tensión a la salida del convertidor tras aplicar el dqPLL (en azul)

Por lo tanto, para la completa sincronización hay que tener en cuenta que entre el convertidor y la red se encuentran las bobinas de las ramas del convertidor y las bobinas de salida del convertidor, que por su carácter inductivo modifican la fase de la tensión del convertidor.

Como solución para eliminar este desfase, se regulan las corrientes de salida del convertidor, de tal manera que, a la tensión V_d del dqPLL se le añade una componente que está relacionada con la inductancia de las bobinas del convertidor, otra componente relacionada con la componente resistiva de dichas bobinas. Al controlar las corrientes de salida, también será posible controlar la potencia que se intercambia con la red. A continuación, se explican estos cambios de forma detallada.

Las corrientes de salida del convertidor Io_a , Io_b , Io_c , se calculan a partir de las medidas de los Analog Output de los amperímetros que hay en cada rama del convertidor. La corriente de salida es igual a la diferencia entre la corriente de la rama superior y la de la rama inferior, como se muestra en la Ecuación 14. En la Figura 110 se puede ver cómo se han implementado estos cálculos en el controlador.

Ecuación 14 - Cálculo corriente de salida a partir de las corrientes de rama

$$Io = I_{armUP} - I_{armLW}$$

Estas corrientes de salida se podrían medir directamente con amperímetros y llevar estas medidas al RCPcore sin necesidad de realizar cálculos, pero al poder utilizar solamente un máximo de 24 *Analog Output*, se ha decidido obtener estas medidas mediante el cálculo directo.

```
Io_a = IupA - IlwA;
Io_b = IupB - IlwB;
Io_c = IupC - IlwC;
```

Figura 110 - Código del loop. Cálculo de las corrientes de salida del convertidor

Las corrientes de salida del convertidor se pasan, mediante la función *uvw2dq*, a un sistema definido por dos ejes móviles. De esta manera, la corriente se descompone en dos componentes, la componente Io_d , estrechamente relacionada con la potencia activa (ver Ecuación 16) y la componente Io_q , estrechamente relacionada con la potencia reactiva (ver Ecuación 18). Este cambio permite tener un mayor control sobre la potencia que se intercambia con la red.

```
uvw2dq (Io_a, Io_b, Io_c, seno_mmc01, coseno_mmc01, Io_d, Io_q);
```

Figura 111 - Código del loop. Paso de corrientes Io_a , Io_b e Io_c a Io_d e Io_q

Para deducir la relación existente entre la componente i_{o_d} y la potencia activa, se parte de la Ecuación 15, que se muestra a continuación.

Ecuación 15 - Relación entre la potencia activa con I_d e I_q

$$P^* = \frac{3}{2} \bar{V}_{fn} \cdot \bar{I}_f = \frac{3}{2} (V_\alpha \cdot I_\alpha + V_\beta \cdot I_\beta) = \frac{3}{2} (V_d \cdot I_d + V_q \cdot I_q)$$

Siendo V_d la componente d resultante de aplicar el dqPLL a la tensión de salida del convertidor y, la I_d , como ya se ha comentado, es la componente de la corriente responsable de la potencia activa. Por otro lado, la tensión V_q es prácticamente 0, a causa de la sincronización del dqPLL.

De esta manera se consigue una expresión de la Ecuación 16, que relaciona la potencia de referencia, la que se quiere transmitir en uno u otro sentido, y la corriente I_d que debe haber en el lado de alterna del convertidor.

Ecuación 16 - Relación entre la potencia activa y la componente d de la corriente

$$P^* = \frac{3}{2} (V_{d_PLL} \cdot I_d^*)$$

Del mismo modo, la expresión que relaciona la componente i_{o_q} con la potencia reactiva se muestra en la Ecuación 17.

Ecuación 17 - Relación entre potencia reactiva con I_d e I_q

$$Q^* = \frac{3}{2} \bar{V}_{fn} \times \bar{I}_f = \frac{3}{2} (V_\alpha \cdot I_\beta - V_\beta \cdot I_\alpha) = -\frac{3}{2} (V_d \cdot I_q - V_q \cdot I_d)$$

Igual que en el caso de la potencia activa, la V_q es prácticamente 0, por lo que esa expresión queda tal como se ve en la Ecuación 18.

Ecuación 18 - Relación entre la potencia reactiva y la componente q de la corriente

$$Q^* = -\frac{3}{2} (V_{d_PLL} \cdot I_q^*)$$

Como ya se ha mencionado anteriormente, la tensión en la salida del convertidor depende de la tensión de la red y de la tensión en las bobinas que hay a la salida del convertidor.

Para tener en cuenta la tensión en las bobinas para la correcta sincronización con la red, se calcula la inductancia equivalente en cada fase mediante la expresión que se muestra en la Ecuación 19.

Ecuación 19 - Cálculo de la inductancia equivalente a la salida del convertidor

$$L_{eq} = \frac{L_{arm} \cdot L_{arm}}{L_{arm} + L_{arm}} + L_0 = \frac{L_{arm} \cdot L_{arm}}{2 \cdot L_{arm}} + L_0 = \frac{L_{arm}}{2} + L_0$$

Por lo tanto, la expresión de la tensión en la salida del convertidor será la que se observa en la Ecuación 20.

Ecuación 20 - Tensión del convertidor corregida

$$\bar{V}_{conv} = L_{eq} \frac{d\bar{I}_o}{dt} + \bar{V}_{grid}$$

Pasando la tensión del convertidor a ejes móviles d y q, la expresión de la Ecuación 20 pasaría a tener la forma de la Ecuación 21.

Ecuación 21 - Tensión del convertidor corregida en ejes d y q

$$(V_d, V_q) = L_{eq} \frac{d(I_{od}, I_{oq})}{dt} + (V_{grid_d}, V_{grid_q})$$

Siendo \bar{d} y \bar{q} vectores de módulo unitario y con la dirección del eje d y del eje q respectivamente. Entonces la expresión de la Ecuación 21 podría ponerse como aparece en la Ecuación 22.

Ecuación 22 - Tensión del convertidor en ejes d y q con vectores unitarios \bar{d} y \bar{q}

$$\begin{aligned} (V_d, V_q) &= L_{eq} \frac{d(I_{od} \cdot \bar{d})}{dt} + L_{eq} \frac{d(I_{oq} \cdot \bar{q})}{dt} + V_{grid_d} \cdot \bar{d} + V_{grid_q} \cdot \bar{q} = \\ &= L_{eq} \frac{d(I_{od})}{dt} \cdot \bar{d} + L_{eq} + L_{eq} I_{od} \frac{d(\bar{d})}{dt} + L_{eq} \frac{d(I_{oq})}{dt} \cdot \bar{q} + L_{eq} I_{oq} \frac{d(\bar{q})}{dt} + V_{grid_d} \cdot \bar{d} + V_{grid_q} \cdot \bar{q} \end{aligned}$$

El cálculo de la derivada temporal de los vectores unitarios \bar{d} y \bar{q} no es trivial, por lo que se deduce con la ayuda la representación en la Figura 112.

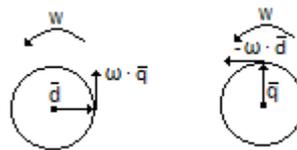


Figura 112 - Derivada temporal de los vectores \bar{d} y \bar{q} (elaboración propia)

El sistema de ejes móviles gira a una velocidad ω . Los vectores unitarios \bar{d} y \bar{q} son perpendiculares entre sí, por lo que, siendo la derivada en un punto tangente a la curva en dicho punto, se tiene que la derivada del vector \bar{d} es la que se muestra en la Ecuación 23.

Ecuación 23 - Derivada temporal del vector unitario \bar{d}

$$\frac{d(\bar{d})}{dt} = \omega \cdot \bar{q}$$

Con el mismo razonamiento se obtiene la derivada del vector \bar{q} . En este caso la derivada sería $\omega \cdot \bar{d}$, que apunta en dirección contraria al vector \bar{d} , por lo que, como se puede ver en la Ecuación 24, tiene signo negativo.

Ecuación 24 – Derivada temporal del vector unitario \bar{q}

$$\frac{d(\bar{q})}{dt} = -\omega \cdot \bar{d}$$

Ahora es posible descomponer los términos de la Ecuación 22 en los ejes d (en la Ecuación 25) y q (en la Ecuación 26):

Ecuación 25 – Componente correspondiente al eje \bar{d}

$$\left(V_{grid_d} - L_{eq} \cdot I_{oq} \cdot \omega + R_L \cdot I_{od} + L_{eq} \frac{d(I_{od})}{dt} \right) \cdot \bar{d}$$

Ecuación 26 – Componente correspondiente al eje \bar{q}

$$\left(V_{grid_q} + L_{eq} \cdot I_{od} \cdot \omega + R_L \cdot I_{oq} + L_{eq} \frac{d(I_{oq})}{dt} \right) \cdot \bar{q}$$

El término R_L representa la resistencia correspondiente a las bobinas. El valor de esta resistencia no es conocido y puede variar en función de diversos factores. Por lo tanto, el tercer sumando de la Ecuación 25 ($R_L \cdot I_{od}$) y el de la Ecuación 26 ($R_L \cdot I_{oq}$), que dependen de R_L , no es posible llegar a conocer su valor. Entonces, como solución, puede utilizarse un regulador proporcional integral para obtener dicho valor actualizado según las condiciones en cada instante.

Por otro lado, en cuanto a la $d(I_{od})$ y $d(I_{oq})$, al considerar la potencia intercambiada constante en el tiempo y ser la tensión estable en el tiempo, entonces las corrientes I_{od} e I_{oq} , responsables de la componente activa y reactiva de la corriente, respectivamente, tampoco variarán en el tiempo. Por lo tanto, su derivada temporal es nula, como se muestra en la Ecuación 27 y en la Ecuación 28.

Ecuación 27 - Variación temporal de I_{od} es un valor cercano a cero

$$L_{eq} \frac{d(I_{od})}{dt} = 0$$

Ecuación 28 - Variación temporal de I_{oq} es un valor cercano a cero

$$L_{eq} \frac{d(I_{oq})}{dt} = 0$$

Aplicando estas deducciones, mediante una referencia de potencia activa y de potencia reactiva y es posible controlar la potencia que se está proporcionando a la red o la se está absorbiendo. Tal como muestra el diagrama de la Figura 113, de las potencias de referencia se puede sacar las tensiones de referencia del convertidor. A su vez, el convertidor está completamente sincronizado con la red.

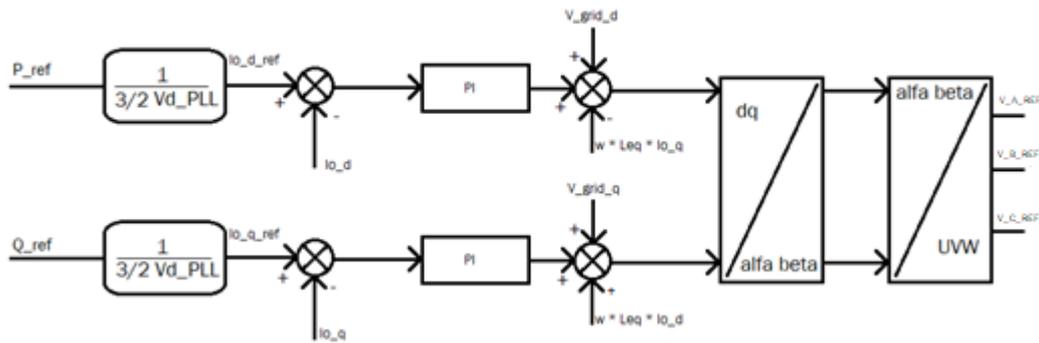


Figura 113 - Diagrama de bloques sincronización con la red (elaboración propia)

Para la simulación que se pretende desarrollar en este trabajo, la referencia de potencia activa es de un megavatio, mientras que la referencia de potencia reactiva es de cero voltio-amperios reactivos.

A continuación, en la Figura 114 se expone cómo se ha implementado el diagrama de bloques de la Figura 113 en el código del controlador.

```
// errorIoD = Io_d_ref - Io_d;
integral_Iod = integral_Iod + Io_d_ref - Io_d;
Vo_d_ref_conv = KP_IODQ_MMC01 * Io_d_ref - KP_IODQ_MMC01 * Io_d + TSXKI_IODQ_MMC01 *
               integral_Iod - WXL_EQ_MMC01 * Io_q + V_d_ref;

// errorIoQ = Io_q_ref - Io_q;
integral_Ioq = integral_Ioq + Io_q_ref - Io_q;
Vo_q_ref_conv = KP_IODQ_MMC01 * Io_q_ref - KP_IODQ_MMC01 * Io_q + TSXKI_IODQ_MMC01 *
               integral_Ioq + WXL_EQ_MMC01 * Io_d + V_q_ref;

dq2uvw(Vo_d_ref_conv, Vo_q_ref_conv, seno_mmc01, coseno_mmc01, V_A_REF, V_B_REF, V_C_REF);
```

Figura 114 - Código del loop. Cálculo de referencias mediante la regulación de las corrientes

Como muestra la Figura 114, el cálculo del error se implementa directamente junto al cálculo de la *integral_Iod*. Al igual que se ha hecho en la Ecuación 12, no se multiplica por el periodo en el cálculo de la integral del regulador, sino que se aplica junto a la constante integral en la siguiente línea dentro del *#define TSKXKI_IODQ*. A continuación, se obtiene la referencia de tensión *Vo_d_ref_conv*, multiplicando por las constantes del regulador proporcional integral, restando la componente relacionada con las inductancias de las bobinas del convertidor y sumando la tensión calculada con el dqPLL sincronizada con la red. De esta forma, el desfase que había en la Figura 109 se elimina.

De la misma manera se obtiene *Vo_q_ref_conv*, teniendo en cuenta que el término correspondiente a la componente relacionada con las inductancias en este caso resta y está multiplicado por *Io_d*.

3.3.2.3.6.- Generación de las moduladoras

Una vez se han calculado las referencias de tensión sincronizadas con la red, se puede generar las señales moduladoras. Para obtener las referencias de las tensiones de las tres fases, se utiliza la función embebida *dq2uvw* (), pasando

de las componentes en ejes móviles dq a ejes fijos uvw, como se muestra en la última línea de la Figura 114.

En la Figura 115 se muestra cómo se calculan las moduladoras con la tensión media en corriente continua, las tensiones de referencias obtenidas de la sincronización. Se multiplica por 0.00125, que es la inversa de 800V, la tensión de una celda, para obtener el valor normalizado de la moduladora.

```
// Obtención tensiones de referencia (sincronizadas con la red)
superior e inferior para las celdas del MMC con 800 V
Va_up_ref = V_DC_HALF * 0.00125 - V_A_REF * 0.00125;
Vb_up_ref = V_DC_HALF * 0.00125 - V_B_REF * 0.00125;
Vc_up_ref = V_DC_HALF * 0.00125 - V_C_REF * 0.00125;
Va_lw_ref = V_DC_HALF * 0.00125 + V_A_REF * 0.00125;
Vb_lw_ref = V_DC_HALF * 0.00125 + V_B_REF * 0.00125;
Vc_lw_ref = V_DC_HALF * 0.00125 + V_C_REF * 0.00125;
```

Figura 115 - Código del loop. Generación de señales moduladoras

3.3.2.3.7.- Cálculo del residuo

De la misma forma que en Matlab, con la función *floor* se calcula la parte entera del valor normalizado de la moduladora. El residuo se calcula como la diferencia entre el valor normalizado de la moduladora y su parte entera. Así lo muestra la Figura 116.

```
// Obtención de los ciclos de trabajo de cada rama
porcentaje_A_UP = Va_up_ref - floor (Va_up_ref);
porcentaje_B_UP = Vb_up_ref - floor (Vb_up_ref);
porcentaje_C_UP = Vc_up_ref - floor (Vc_up_ref);

porcentaje_A_LW = Va_lw_ref - floor(Va_lw_ref);
porcentaje_B_LW = Vb_lw_ref - floor(Vb_lw_ref);
porcentaje_C_LW = Vc_lw_ref - floor(Vc_lw_ref);
```

Figura 116 - Código de loop. Cálculo del residuo

3.3.2.3.8.- Cálculo estado de los polos de potencia

En la Figura 117 se puede ver cómo se calcula el estado de los polos de potencia. Como se puede ver, se suman las componentes correspondientes del vector *pS_mmc01* para el realista, o, *pS_mmc02* para el promediado, que son los vectores que almacenan el estado de los polos.

```
// Estado actual
ESTADO_ACTUAL_A_UP = pS_mmc01[UP_A_1] + pS_mmc01[UP_A_2] + pS_mmc01[UP_A_3] + pS_mmc01[UP_A_4];
ESTADO_ACTUAL_B_UP = pS_mmc01[UP_B_1] + pS_mmc01[UP_B_2] + pS_mmc01[UP_B_3] + pS_mmc01[UP_B_4];
ESTADO_ACTUAL_C_UP = pS_mmc01[UP_C_1] + pS_mmc01[UP_C_2] + pS_mmc01[UP_C_3] + pS_mmc01[UP_C_4];

ESTADO_ACTUAL_A_LW = pS_mmc01[LW_A_1] + pS_mmc01[LW_A_2] + pS_mmc01[LW_A_3] + pS_mmc01[LW_A_4];
ESTADO_ACTUAL_B_LW = pS_mmc01[LW_B_1] + pS_mmc01[LW_B_2] + pS_mmc01[LW_B_3] + pS_mmc01[LW_B_4];
ESTADO_ACTUAL_C_LW = pS_mmc01[LW_C_1] + pS_mmc01[LW_C_2] + pS_mmc01[LW_C_3] + pS_mmc01[LW_C_4];
```

Figura 117 - Código del loop. Cálculo del estado actual

3.3.2.3.9.- Vectores de recuperación de valores

Para poder utilizar los dos convertidores al mismo tiempo sin tener que duplicar los recursos, ya se ha decidido que uno de los convertidores como mínimo tiene que ser promediado para salvar las limitaciones ya comentadas con anterioridad. Pero aún existe una limitación con respecto al número de variables globales. No hay variables suficientes como para duplicarlas para que cada convertidor tenga sus propias variables globales. Incluso duplicar solamente las variables que necesiten conservar el valor de una iteración a otra no sería posible.

Como solución, al tener 2000 variables vectoriales en la memoria compartida, se ha optado por generar dos vectores, uno para el *mmc01* (el realista) y otro para el *mmc02* (el promediado), para que, con estos vectores, se almacenen las variables que cada convertidor necesita guardar tras cada iteración, para recuperar esos datos antes de volver a necesitarlos y así poder utilizar ambos convertidores sin tener que aumentar el número de variables globales escalares.

También se utilizan estos vectores para realizar todas las medidas de corriente y tensión con la menor diferencia tiempo posible. Como ya se ha comentado, primero se leen los valores relacionados con el convertidor promediado, a continuación, se vuelcan estos valores en el vector *valores_mmc02*, para después leer los valores correspondientes al convertidor realista, que, al realizar en primer lugar los cálculos correspondientes a este convertidor, no es necesario almacenar estas medidas de tensión y corriente.

El vector *valores_mmc02* utiliza las primeras 6 posiciones para almacenar las medidas de tensión y corriente y las posiciones de la 7 a la 11 para almacenar los valores de contadores e integrales. El vector *valores_mmc01* solamente tiene que almacenar los valores de contadores e integrales, pero por simetría y al tener disponibles variables vectoriales de sobra, se ha decidido ocupar solamente las componentes de la 7 a la 11, dejando libre las otras posiciones.

```
// RECUPERAR VALORES DE INTEGRACIÓN
readVector(valores_mmc01, 7, integralVdc, theta, integral_Vq, integral_Iod, integral_Ioq);
```

Figura 118 - Código del loop. Recuperación de valores de integración

En la Figura 118, se puede ver cómo mediante la función *readVector* se toman los datos de la iteración anterior y se pasan a las variables escalares correspondientes. Tanto en el convertidor realista como en el convertidor promediado, antes de empezar a realizar cálculos, se recuperan estos valores.

```
// VOLCADO DE VALORES AL VECTOR PARA CONSERVARLOS PARA LA PRÓXIMA ITERACIÓN
writeVector(valores_mmc01, 7, integralVdc, theta, integral_Vq, integral_Iod, integral_Ioq);
```

Figura 119 - Código del loop. Volcado de valores al vector

Cuando ya se han realizado todos los cálculos, se procede al volcado de estos datos al vector para que se conserven mientras se utilizan las variables escalares en el otro convertidor. Como se muestra en la Figura 119, en este caso se utiliza la función `writeVector` para enviar los datos de las variables escalares al vector.

```
// RECUPERAR VALORES MEDIDOS AL INICIO Y RECORDAR VALOR DE VARIABLES INTEGRADAS  
  
readVector(valores_mmc02, 0, IupA, IlwA, IupB, IlwB, IupC, IlwC, V_dc,  
integralVdc, theta, integral_Vq, integral_Iod, integral_Ioq);
```

Figura 120 - Código del loop. Recuperación de valores en el mmc02

En el caso del convertidor promediado, al contener también los datos de corrientes y tensiones medidas al inicio del periodo, al recuperar los datos, se recuperan al mismo tiempo que las otras variables de integración, como se observa en la Figura 120.

3.3.2.3.10.- Elegir celda que cambia

En este apartado se explica cómo se selecciona la celda que debe encenderse o apagarse en función de lo que requiera el convertidor correspondiente. El criterio de conmutación es el mismo que se ha explicado en el convertidor de Matlab, pero al ser diferentes las funciones *posVmin* y *posVmax* del simulador en tiempo real con respecto a las funciones del convertidor de Matlab, el código es diferente, más adaptado a las necesidades de la simulación en tiempo real.

En primer lugar, una variable llamada *ESTADO_PREVIO* toma el valor del estado actual de la rama (ver Figura 121).

Después, se actualiza el ciclo de servicio. El polo de potencia que tenía el residuo pasa a valer 1 si se tenía que encender o 0 si se tenía que apagar (ver Figura 121).

Bajo el mismo razonamiento que en el convertidor en Simulink:

Si el valor normalizado de la moduladora es mayor que el número de celdas activas, se necesita encender al menos una celda.

Cuando la corriente es positiva, es decir, la celda que se encienda se cargará, entonces, la celda que debe encenderse es la de menor tensión de entre las que están apagadas.

En el caso contrario, si la corriente es negativa, se debe elegir la celda de mayor tensión entre las que están apagadas para que se descargue.

Por otro lado, si el valor normalizado de la moduladora es menor que el número de celdas activas es necesario apagar al menos celda.

Entonces, cuando la corriente es positiva, se deberá apagar la celda con la tensión más elevada de las que se encuentran encendidas, para que no se cargue más.

Y, por lo tanto, cuando la corriente es negativa, se buscará entonces apagar la celda con la menor tensión de entre las insertadas, para que no se descargue más.

Una vez elegida la celda, se calcula la posición a partir del número correspondiente a primera celda de la rama, sumándole el valor que resulta de la función *posVmin* o *posVmax*. A la posición seleccionada del vector del ciclo de servicio se le asigna el valor del residuo, es decir, el porcentaje del periodo que estará la celda encendida. Al vector que conserva el estado de los polos de potencia se le asigna directamente 1 o 0, según tenga que evolucionar la celda a encendido o apagado.

Si ha sido necesario encender alguna celda, a la variable *porcentaje_(fase)_(polo)*, se le asigna un 1 después de la primera iteración del bucle *while* para que, en el caso de necesitar encender más de una celda, las siguientes iteraciones del bucle enciendan una celda entera. La variable *ESTADO_ACTUAL* pasa a ser *ESTADO_ACTUAL +1*.

En el caso de que haya que apagar celdas, a la variable *porcentaje_(fase)_(polo)*, se le asigna un 0 después de la primera iteración del bucle *while* para que, en el caso de necesitar apagar más de una celda, las siguientes iteraciones del bucle apaguen una celda entera. La variable *ESTADO_ACTUAL* pasa a ser *ESTADO_ACTUAL -1*.

Entre el bucle *while* para encender celdas (el de la Figura 121) y el de apagar celdas (el de la Figura 122) se actualiza el estado previo: *ESTADO_ACTUAL = ESTADO_PREVIO*. Para que, al haber encendido celdas en el primer bucle, ya no se pueda volver a entrar en el segundo bucle.

El código implementado en el controlador para la elección de la celda que cambia referido a la fase A rama UP (superior), se puede ver en la Figura 121 y en la Figura 122.

```

// Elección del que cambia y asignación de porcentaje
// ***** A_UP *****

ESTADO_PREVIO = ESTADO_ACTUAL_A_UP;

dutyCycle_mmc01[UP_A_1] = pS_mmc01[UP_A_1];
dutyCycle_mmc01[UP_A_2] = pS_mmc01[UP_A_2];
dutyCycle_mmc01[UP_A_3] = pS_mmc01[UP_A_3];
dutyCycle_mmc01[UP_A_4] = pS_mmc01[UP_A_4];

while (Va_up_ref > ESTADO_ACTUAL_A_UP) {
    if (IupA > 0) {
        pos = posVmin (BYPASS, Vcond_UP_A_1, Vcond_UP_A_2, Vcond_UP_A_3, Vcond_UP_A_4,
            pS_mmc01[UP_A_1], pS_mmc01[UP_A_2], pS_mmc01[UP_A_3], pS_mmc01[UP_A_4]);
    } else {
        pos = posVmax (BYPASS, Vcond_UP_A_1, Vcond_UP_A_2, Vcond_UP_A_3, Vcond_UP_A_4,
            pS_mmc01[UP_A_1], pS_mmc01[UP_A_2], pS_mmc01[UP_A_3], pS_mmc01[UP_A_4]);
    }
    if (pos >= 0){
        pos_vector          = UP_A_1 + pos;
        dutyCycle_mmc01[pos_vector] = porcentaje_A_UP;
        pS_mmc01[pos_vector]   = 1;
        porcentaje_A_UP       = 1;
    }
    ESTADO_ACTUAL_A_UP     = ESTADO_ACTUAL_A_UP + 1;
}

ESTADO_ACTUAL_A_UP = ESTADO_PREVIO;

```

Figura 121 - Código del loop. Elección de la celda que cambia. Parte 1

```

while (Va_up_ref < ESTADO_ACTUAL_A_UP) {
    if (IupA > 0) {
        pos = posVmax (INSERTED, Vcond_UP_A_1, Vcond_UP_A_2, Vcond_UP_A_3, Vcond_UP_A_4,
            pS_mmc01[UP_A_1], pS_mmc01[UP_A_2], pS_mmc01[UP_A_3], pS_mmc01[UP_A_4]);
    } else {
        pos = posVmin (INSERTED, Vcond_UP_A_1, Vcond_UP_A_2, Vcond_UP_A_3, Vcond_UP_A_4,
            pS_mmc01[UP_A_1], pS_mmc01[UP_A_2], pS_mmc01[UP_A_3], pS_mmc01[UP_A_4]);
    }
    if (pos >= 0){
        pos_vector          = UP_A_1 + pos;
        dutyCycle_mmc01[pos_vector] = porcentaje_A_UP;
        pS_mmc01[pos_vector]   = 0;
        porcentaje_A_UP       = 0;
    }
    ESTADO_ACTUAL_A_UP     = ESTADO_ACTUAL_A_UP - 1;
}

```

Figura 122 - Código del loop. Elección de la celda que cambia. Parte 2

3.3.3.- Explicación funcionamiento Master y Slave

Ambos convertidores funcionan de la misma manera, pudiendo enviar potencia en cualquiera de los dos sentidos. Para ello, hay dos modos de convertidor, el modo *Master* y el modo *Slave*. El modo en el que se encuentra cada convertidor se ordena en la pestaña “*Global*” del controlador mediante un *#define*. Si el mmc01, está definido en modo *Slave*, implica que el mmc02 estará en modo *Master*. Por el contrario, si el mmc01 se encuentra definido en modo *Master*, el convertidor promediado funcionará en modo *Slave*.

El funcionamiento en modo *Master* y modo *Slave* se explica más en detalle a continuación.

3.3.3.1.- Funcionamiento modo Master

El convertidor que absorbe la potencia de la red es el que funciona en modo *Master*. Este convertidor impone una referencia de potencia activa que debe absorber de la red, esta referencia de potencia se traduce en una referencia fija de la corriente que debe demandar de la red. La potencia que ha tomado de la red pasa a la línea y llega al otro convertidor.

También podría imponerse una referencia de potencia reactiva para que el convertidor absorba o inyecte potencia reactiva a la red, pero en esta simulación se ha impuesto una referencia de reactiva igual a cero.

Como ya se ha mencionado, la referencia de potencia que el convertidor en modo Master debe absorber se pasa a una referencia de corriente mediante la expresión de la Ecuación 29 para pasar la referencia de potencia activa a corriente $I_{o_d_ref}$. En el caso de la potencia reactiva, mediante la Ecuación 30 se pasaría de la referencia de potencia reactiva a una referencia de corriente $I_{o_q_ref}$.

Ecuación 29 - Cambio de referencia de potencia activa a referencia de corriente

$$I_{o_d_ref} = P_{ref} \frac{1}{\frac{3}{2} \cdot V_{d_PLL}}$$

Ecuación 30 - Cambio de referencia de potencia reactiva a referencia de corriente

$$I_{o_q_ref} = -Q_{ref} \frac{1}{\frac{3}{2} \cdot V_{d_PLL}}$$

La Ecuación 29 se ha implementado en el código del controlador de la manera en que se muestra en la Figura 123.

```
#ifndef MMC01_MASTER
Io_d_ref = P_REF * 0.6666666667 / V_d_ref;
#endif
```

Figura 123 - Código funcionamiento en modo Master

Como se puede ver, en el código se ha puesto directamente el valor de $\frac{1}{3/2}$, que es $\frac{2}{3}$. Esto es para evitar que se invierta tiempo en una operación sencilla pero que, al ser un cociente, la FPGA destina más tiempo que a otro tipo de operaciones. Por otro lado, se evita que el compilador tome los números 2 y 3 como enteros y entonces el resultado sea un número entero, 0. Esta solución no sería válida.

3.3.3.2.- Funcionamiento en modo Slave

El convertidor que recibe la potencia, que trabaja en modo *Slave*, debe mantener la tensión en el lado de continua en su valor de referencia. Al llegar la potencia por la línea la tensión tiende a aumentar, por lo que reacciona enviándola hacia la red a la que está conectado, manteniendo la tensión en continua en el valor nominal.

El código del modo de funcionamiento *Slave*, se muestra en la Figura 125. Este código está basado en el diagrama de bloques de la Figura 124.

Con un voltímetro se observa la medida de tensión de continua que hay en cada momento, esta medida se envía al controlador a través de un *Analog Output*. Se pone como referencia de tensión en corriente continua los 3200 V que corresponden. Hallando la diferencia entre el valor actual de tensión y el de consigna se calcula el error entre ambas. Mediante un regulador PI (Proporcional Integral) se obtiene la corriente que debe intercambiar con la red para mantener la tensión en su valor de consigna.

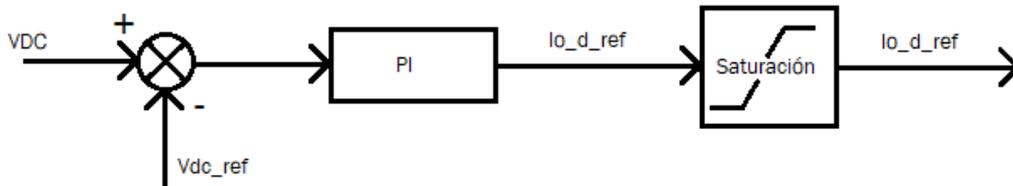


Figura 124 - Diagrama de bloques funcionamiento en modo Slave (elaboración propia)

```
// Calculo referencia para mantener tensión de los condensadores
#ifdef MMC01_SLAVE
// error = VDC - Vdc_ref;
integralVdc = integralVdc + V_dc - Vdc_ref;
Io_d_ref = KPV_DC_MMC01 * V_dc - KPV_DC_MMC01 * Vdc_ref + TSKIV_DC_MMC01 * integralVdc;
Io_d_ref = constrain(Io_d_ref, -PI_LIMIT_A, PI_LIMIT_A);
#endif
```

Figura 125 - Código funcionamiento en modo Slave

Para que el valor de la referencia de la corriente no se eleve hasta valores demasiado altos, se ha puesto unos límites superior e inferior mediante la función *constrain*, y, de esta manera los supuestos equipos del convertidor no recibirían sobretensiones descontroladas durante los transitorios.

Para elegir los límites superior e inferior de la función *constrain* se ha definido *PI_LIMIT_A*. Como se puede ver en la Ecuación 31, se ha establecido como la potencia de referencia dividido entre la tensión pico de la red multiplicada por un margen a mayores del 50% y, el resultado se incrementa en un 20%. Esta relación con estos márgenes funciona de la forma deseada, impidiendo que la corriente se vaya a valores demasiado elevados en el inicio de la simulación y permitiendo el funcionamiento normal de los convertidores.

$$PI_LIMIT_A = \frac{P_{REF}}{V_{P_GRID} \cdot 1.5} + 20\%$$

En la gráfica de la Figura 126 se muestran la corriente de referencia (Io_d_ref) tanto del convertidor realista en modo *Master*, como del convertidor promediado funcionando en modo *Slave*. Tal como se puede ver, la línea azul no presenta ninguna variación, ya que el convertidor que funciona en modo *Master* se impone la referencia. En la línea roja se pueden observar pequeñas variaciones, ya que, al estar en modo *Slave*, la referencia de la corriente varía para mantener la tensión en el lado de continua en su valor nominal. La corriente del convertidor realista es negativa porque la está absorbiendo de la red de origen, mientras que la del convertidor promediado es positiva porque la inyecta en la red de destino.

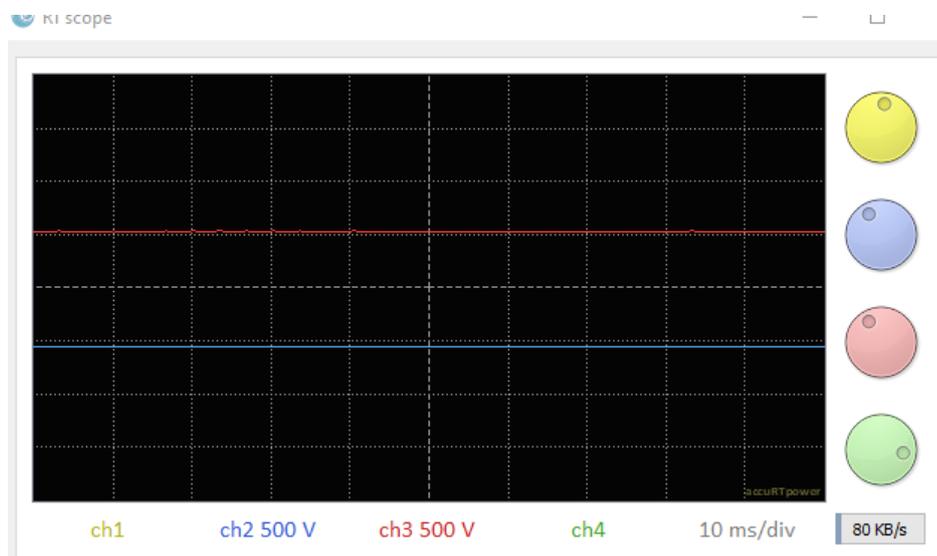


Figura 126 - Corriente de referencia Io_d_ref de cada convertidor. En rojo la del promediado en modo *Slave*, en azul la del realista en modo *Master*

3.3.4.- Convertidor realista

En este punto se va a presentar comportamiento del convertidor realista de la simulación en tiempo real funcionando de forma aislada. En primer lugar, se observará el comportamiento del convertidor conectado a una carga resistiva y, después, se verá el convertidor conectado a una red trifásica en corriente alterna.

En esta planta, al haber solamente un convertidor, hay elementos de sobra para poner un filtro de condensadores en la salida del convertidor, como el que se puede ver en la planta que muestra la Figura 127. Este filtro suaviza el rizado de la corriente. Como en la planta de la simulación del HVDC no es posible tener este filtro de condensadores, las medidas de corriente que se van a observar en este apartado son tomadas antes del filtro.

Como convertidor se utiliza el bloque de convertidor MMC que se ha presentado más atrás en el apartado 2.11.3.2.1.- Explicación bloque del MMC. Al haber un solo convertidor, en el lado de corriente continua hay dos baterías de 1600 V representando la tensión que habría en la supuesta línea HVDC. Entre las dos baterías suman un total de 3200 V, como se muestra en la Figura 127, hay una referencia en mitad conectada a través de una resistencia de un mega ohmio.

3.3.4.1.- Convertidor realista con carga aislada

En este apartado se va a mostrar el comportamiento del convertidor realista cuando se encuentra conectado a una carga resistiva.

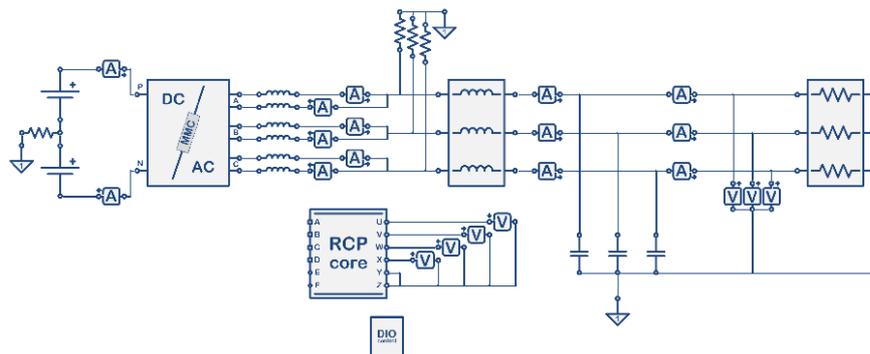


Figura 127 - Planta del convertidor realista conectado a carga resistiva en el simulador en tiempo real

En la gráfica que aparece en la Figura 128, se muestran las tensiones de salida del convertidor. En esta gráfica, cada división en el eje y son 600 V y cada división en el eje x son 4 milisegundos. Por lo tanto, las tensiones a la salida son de unos 1200 voltios pico y las tres fases están separadas por 120° entre sí. Pasando los voltios pico a valor eficaz quedarían unos 850 V_{rms}.

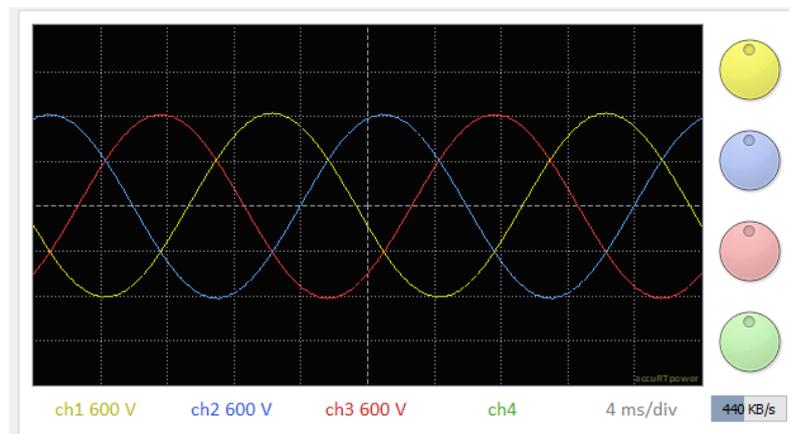


Figura 128 - Tensiones de salida del convertidor realista conectado a carga resistiva

Las corrientes de salida del convertidor antes del filtro de condensadores se muestran en la Figura 129. Estas corrientes tienen un rizado considerable, pero, como ya se ha explicado, debido a la limitación de elementos a 32 como

máximo, no es posible colocar el filtro de condensadores cuando se interconectan los dos convertidores en la misma planta. Por otro lado, el tratamiento de la corriente para eliminar el rizado y los armónicos, no es un objetivo de este trabajo, ya que conlleva mayor cantidad de recursos. Esto podría ser una posible línea para trabajos futuros.

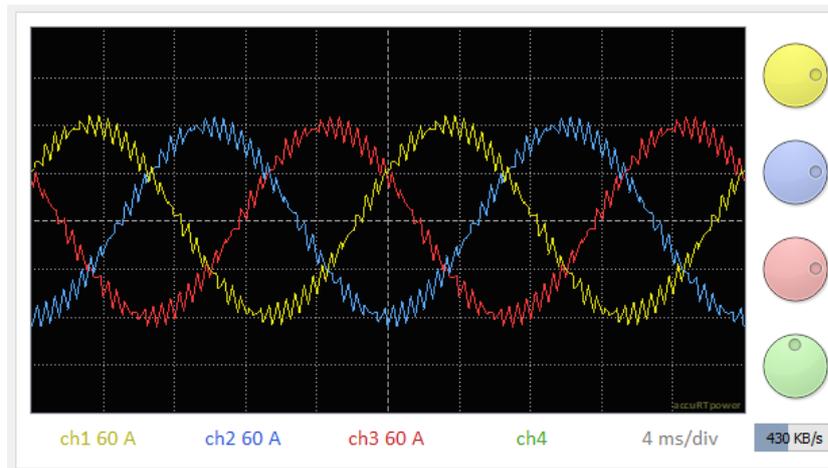


Figura 129 - Corrientes de salida del convertidor realista conectado a carga resistiva

En la Figura 130 se puede ver la corriente de la rama superior I_{up} , la corriente de la rama inferior I_{lw} y la suma de ambas corrientes, resultando la corriente de salida como una de las de la Figura 129. Las corrientes de cada rama tienen componente de corriente continua que al sumarse desaparece.

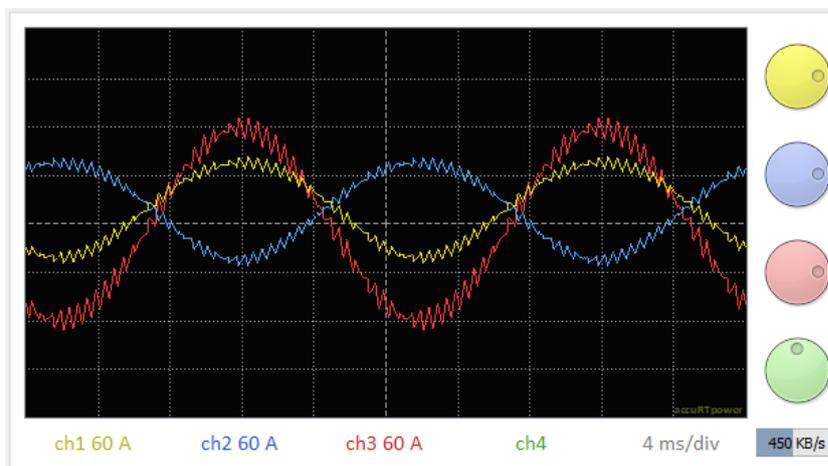


Figura 130 - Corrientes de salida de la rama superior (en amarillo), la rama inferior (en azul) y la suma de ambas (en rojo) del convertidor realista conectado a carga resistiva

3.3.4.2.- Convertidor realista conectado a red

En este apartado se expone el comportamiento del convertidor realista cuando está conectado a una red de 850 voltios eficaces y 50 Hz. En este caso, el convertidor también está conectado a baterías en el lado de continua al no haber una línea que le proporcione la tensión.

La planta del convertidor realista conectado a red es la que se ve en la Figura 131.

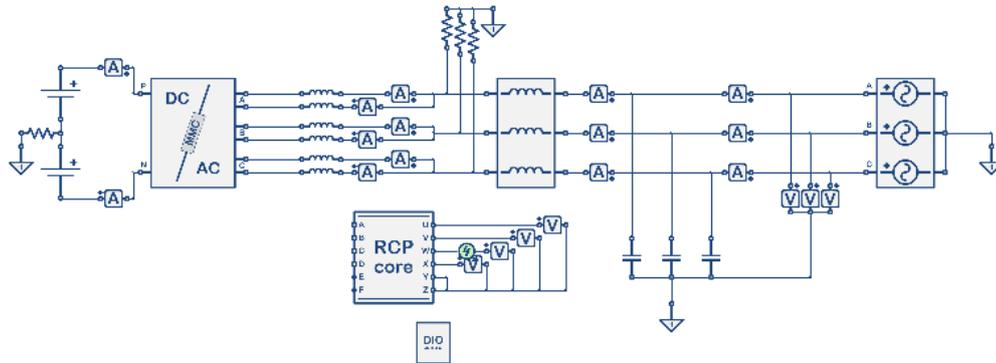


Figura 131 - Planta del convertidor realista conectado a red en el simulador en tiempo real

En la siguiente imagen (Figura 132) se observan las tensiones de salida del convertidor. Al igual que en el caso del convertidor conectado a carga resistiva, cada división en el eje y son 600 V y cada división en el eje x son 4 milisegundos, para poder comparar los resultados. Al encontrarse conectado a la red, las tensiones son idénticas a las de la red. Por lo tanto, las tensiones a la salida del conertidor también son de unos 1200 voltios pico y las tres fases equidistan 120° entre sí. De la misma manera, pasando los voltios pico a valor eficaz quedarían los $850 V_{rms}$.

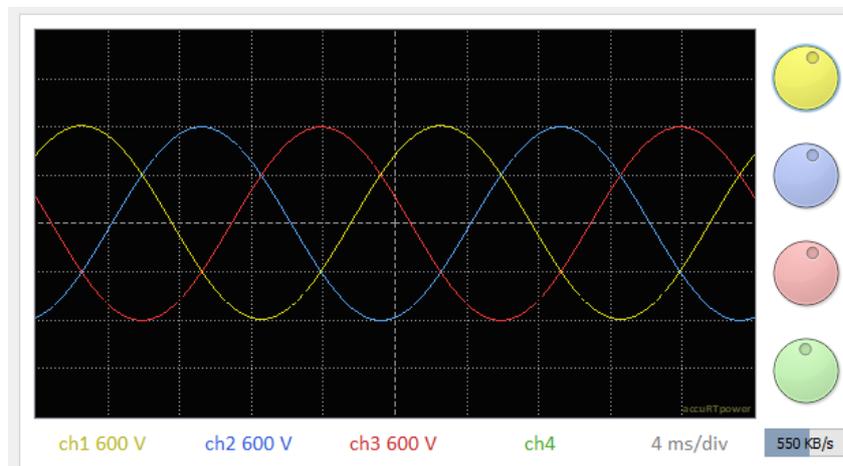


Figura 132 - Tensiones de salida del convertidor realista conectado a Red

A continuación, en la Figura 133 se muestran las corrientes a la salida del convertidor. Como se puede ver, la corriente del convertidor conectado a red presenta un rizado similar al que había en el caso del convertidor conectado a carga resistiva. Se puede ver cómo la amplitud de la corriente sí que cambia. Esto es debido solamente a que no coincide la potencia intercambiada con la

red en estos dos casos. Pero lo destacable en este apartado es que las corrientes forman un sistema trifásico equilibrado.

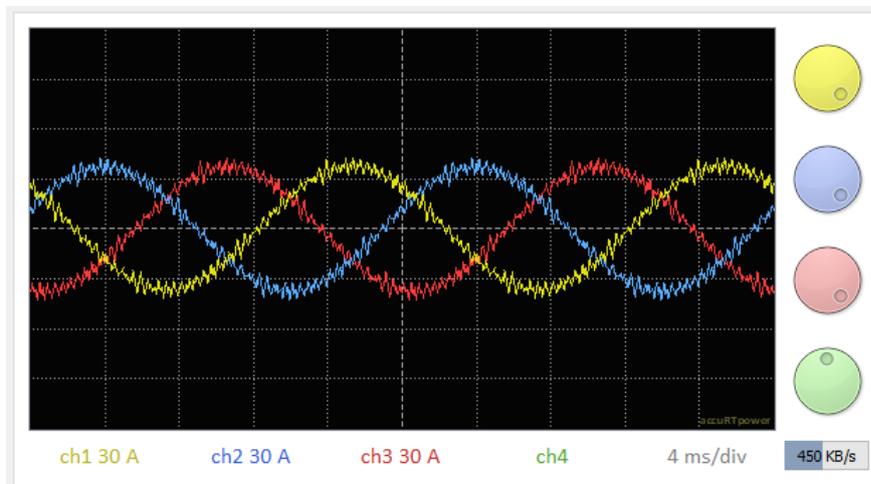


Figura 133 - Corrientes de salida del convertidor realista conectado a Red

Finalmente, en la Figura 134, se pueden observar las corrientes de salida de cada rama de una misma fase junto a la suma de ambas corrientes, que resulta ser la corriente de salida del convertidor. El comportamiento es el esperado.

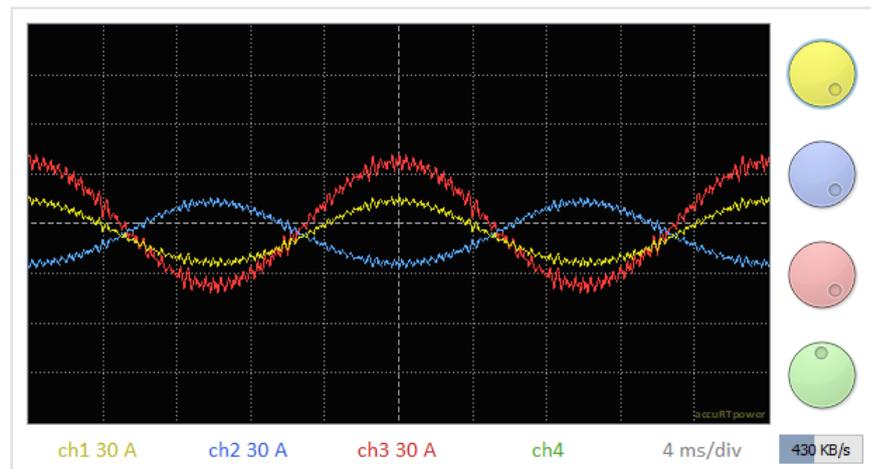


Figura 134 - Corrientes de salida de la rama superior (amarillo), la rama inferior (azul) y la suma de ambas (rojo) del convertidor realista conectado a Red

3.3.5.- Convertidor promediado

Como ya se ha explicado con anterioridad, la implementación del convertidor promediado se debe a la necesidad de economizar los recursos invertidos en la realización de esta simulación.

Dadas las características de la tarjeta FPGA para la simulación en tiempo real, no es posible implementar dos convertidores realistas de 24 celdas. Ya se han explicado los motivos por lo que esto sucede en el apartado 3.1.- Presentación de la línea que se pretende simular.

Por tanto, una solución con la que se consigue implementar la línea HVDC con los dos convertidores MMC, evitando tener que utilizar soluciones económicamente más caras, es usar un convertidor realista y que el otro convertidor sea promediado a partir de los datos de la simulación.

El objetivo del convertidor promediado es conseguir un comportamiento lo más parecido posible al convertidor realista, utilizando una menor cantidad de recursos de simulación.

El convertidor promediado se ha creado a partir de un par de bloques de fuentes de tensión trifásicos programables desde el controlador, como el que se puede ver en la Figura 135. Un bloque para las tensiones de la rama superior y otro para las de la rama inferior. Este tipo de bloques funcionan definiendo una variable vectorial de tres componentes y llamando al bloque de la misma manera que al vector. El bloque aportará la tensión que se haya programado en el controlador a través del vector.

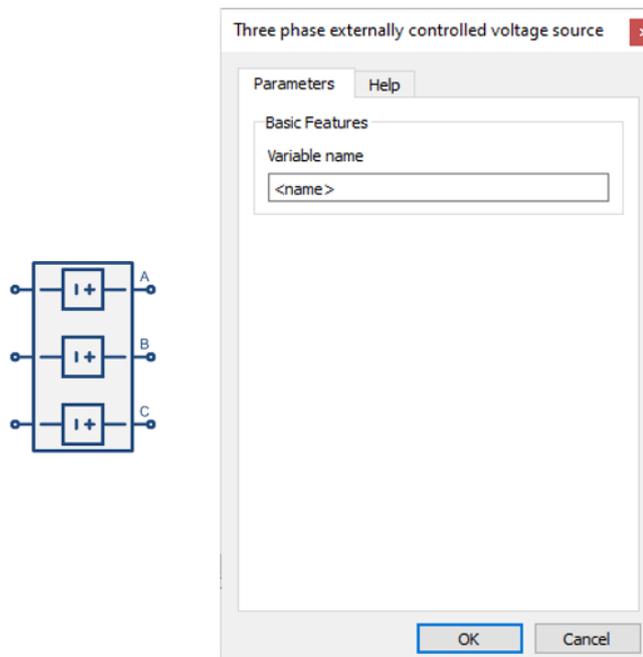


Figura 135 - Bloque fuente de tensión trifásica programable

A partir del desarrollo de este Trabajo de Fin de Grado, se ha implementado, en el bloque MMC del simulador en tiempo real, la opción de utilizar el convertidor realista o promediado, tal como muestra la Figura 136, *Complete* o *Averaged*.

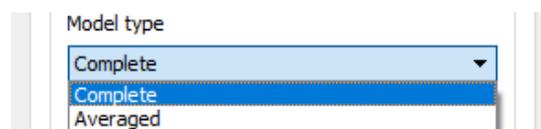


Figura 136 - Elección del tipo de convertidor dentro del bloque del convertidor MMC

3.3.5.1.- Explicación del convertidor promediado

El convertidor promediado parte del siguiente razonamiento:

En una rama del convertidor hay cuatro celdas, cada una de ellas con un condensador. El convertidor promediado toma los valores de tensión para cada rama a partir del ciclo de servicio multiplicado por la tensión promedio en los condensadores de cada rama. De esta forma, se obtiene la tensión promediada de la fase A superior, fase A inferior, fase B superior, fase B inferior, fase C superior y fase C inferior. Estas tensiones promedio se envían a los bloques de tensión trifásicos programables.

Por ejemplo, si el ciclo de servicio es 2.6, quiere decir que hay dos celdas conectadas el 100% del periodo y una tercera celda conectada solo durante el 60% del tiempo. Si la tensión media en los cuatro condensadores esa rama es 800 V, la tensión promedio de esa rama se calculará de la forma que aparece en la Ecuación 32.

Ecuación 32 - Cálculo de tensión del convertidor promediado

$$V_{A_UP} = [1 + 1 + 0 + 0.6] \cdot 800 \text{ V} = \text{CicloDeServicio} \cdot V_{\text{average_Cap_A_UP}}$$

Por otro lado, la tensión promedio en los condensadores de cada rama, no es constante, sino que oscila en el tiempo en torno a un valor de 800 V. Esta oscilación se debe a que en cada periodo se van encendiendo y apagando las celdas en función de la tensión que se requiera en cada momento. Esto hace que los condensadores se vayan descargando (o cargando) cuando los condensadores estén encendidos (cuando la corriente sea en sentido favorable se cargarán los condensadores que estén encendidos) y esto hará que el promedio de la tensión en las celdas caiga (o aumente). Para replicar este comportamiento en el convertidor promediado, se ha utilizado el siguiente razonamiento:

La fórmula que describe la tensión en un condensador es la presente en la Ecuación 33.

Ecuación 33 - Tensión en un condensador

$$V_c = \frac{1}{C} \int i \, dt$$

Una forma de ver esta expresión de la Ecuación 33 es como que la tensión del condensador es igual al valor que tenía en el periodo anterior más el valor actual de la corriente, multiplicado por el periodo y por la inversa de la capacidad del condensador. Como se ve en la Ecuación 34.

Ecuación 34 - Desglose tensión en un condensador

$$V_c = V_c(t - T_s) + \frac{T_s}{C} \cdot i$$

De esta forma se puede realizar la integración en lenguaje C de manera sencilla y rápida, ya que la evolución de la tensión en un condensador sería la que aparece en la Ecuación 35.

Ecuación 35 - Fórmula para la evolución de la tensión en un condensador programada en C

$$V_C = V_C + \frac{T_S}{C} \cdot i$$

De esta manera, V_C , en cada iteración, obtiene el valor anterior que tiene almacenado en el momento de realizar esta operación y le suma el valor actual de la corriente, multiplicado por el periodo y por la inversa de C, obteniendo como resultado el nuevo valor de V_C y logrando la integración en cada periodo. De esta manera, cuando la corriente sea positiva aumentará la tensión promedio, mientras que, si la corriente es negativa, la tensión promedio disminuirá.

Pero el objetivo es calcular el promedio de la tensión de los cuatro condensadores de cada rama para conseguir replicar el comportamiento del convertidor realista.

Para calcular esta tensión promedio en los cuatro condensadores, se tiene en cuenta el ciclo de servicio, es decir, el número de celdas activas en cada periodo. Además, hay que tener en cuenta que, al ser cuatro celdas, la capacidad, C, está multiplicada por cuatro (hay 4 condensadores en una rama). Por lo tanto, quedaría la expresión de la Ecuación 36.

Ecuación 36 - Cálculo de la tensión media en una rama en el convertidor promediado

$$V_{AVE_CAP} = V_{AVE_CAP} + \frac{[CicloDeServicio] \cdot T_S}{4 \cdot C} \cdot i_{arm}$$

De esta forma se calcula la tensión promedio de los condensadores de las 4 celdas de cada rama.

La implementación del convertidor promediado en el código es prácticamente igual que la del convertidor realista salvo por algunas diferencias. Por ejemplo, en el caso del convertidor promediado no es necesario utilizar los generadores de PWM.

Como ya se ha comentado en el apartado del RCPcore, a la hora de realizar las medidas de tensiones y corrientes, se toman primero las correspondientes al convertidor promediado y después se guardan en un vector porque primero se realizan los cálculos del convertidor realista.

Una vez se recuperan los valores de corrientes y tensiones del vector *valores_mmc02*, en primer lugar, se calculan las tensiones promedio de cada rama, aplicando los razonamientos que se han comentado con la Ecuación 36.

En la Figura 137 se puede ver cómo se ha implementado la Ecuación 36 en el código del controlador.

```
V_A_UP_ave = V_A_UP_ave + (dutyCycle_mmc02[UP_A_1] + dutyCycle_mmc02[UP_A_2] +
    dutyCycle_mmc02[UP_A_3] + dutyCycle_mmc02[UP_A_4]) * K_TS_CSM * IupA;
V_B_UP_ave = V_B_UP_ave + (dutyCycle_mmc02[UP_B_1] + dutyCycle_mmc02[UP_B_2] +
    dutyCycle_mmc02[UP_B_3] + dutyCycle_mmc02[UP_B_4]) * K_TS_CSM * IupB;
V_C_UP_ave = V_C_UP_ave + (dutyCycle_mmc02[UP_C_1] + dutyCycle_mmc02[UP_C_2] +
    dutyCycle_mmc02[UP_C_3] + dutyCycle_mmc02[UP_C_4]) * K_TS_CSM * IupC;

V_A_LW_ave = V_A_LW_ave + (dutyCycle_mmc02[LW_A_1] + dutyCycle_mmc02[LW_A_2] +
    dutyCycle_mmc02[LW_A_3] + dutyCycle_mmc02[LW_A_4]) * K_TS_CSM * IlwA;
V_B_LW_ave = V_B_LW_ave + (dutyCycle_mmc02[LW_B_1] + dutyCycle_mmc02[LW_B_2] +
    dutyCycle_mmc02[LW_B_3] + dutyCycle_mmc02[LW_B_4]) * K_TS_CSM * IlwB;
V_C_LW_ave = V_C_LW_ave + (dutyCycle_mmc02[LW_C_1] + dutyCycle_mmc02[LW_C_2] +
    dutyCycle_mmc02[LW_C_3] + dutyCycle_mmc02[LW_C_4]) * K_TS_CSM * IlwC;
```

Figura 137 - Cálculo tensiones promedio en cada rama

Estas tensiones promedio de cada rama se llevan a las variables escalares reservadas para almacenar la tensión del condensador de cada celda. En el caso del convertidor promediado, como se ve en la Figura 138, para una misma rama los cuatro condensadores tienen la misma tensión, la tensión promedio recién calculada.

```
Vcond_UP_A_1 = V_A_UP_ave;
Vcond_UP_A_2 = V_A_UP_ave;
Vcond_UP_A_3 = V_A_UP_ave;
Vcond_UP_A_4 = V_A_UP_ave;

Vcond_LW_A_1 = V_A_LW_ave;
Vcond_LW_A_2 = V_A_LW_ave;
Vcond_LW_A_3 = V_A_LW_ave;
Vcond_LW_A_4 = V_A_LW_ave;
```

Figura 138 - Paso de tensiones promedio a variables escalares por celda

Tras realizar todos los cálculos correspondientes, una vez se han seleccionado las celdas que se encienden o se apagan en cada rama, se calcula la tensión promedio resultante en cada una de las ramas, de la manera en que se ha indicado en la Ecuación 32. En la Figura 139 se muestra como se ha implementado estos cálculos.

```
V_A_UP_PROMED = (dutyCycle_mmc02[UP_A_1] + dutyCycle_mmc02[UP_A_2] +
    dutyCycle_mmc02[UP_A_3] + dutyCycle_mmc02[UP_A_4]) * V_A_UP_ave;
V_B_UP_PROMED = (dutyCycle_mmc02[UP_B_1] + dutyCycle_mmc02[UP_B_2] +
    dutyCycle_mmc02[UP_B_3] + dutyCycle_mmc02[UP_B_4]) * V_B_UP_ave;
V_C_UP_PROMED = (dutyCycle_mmc02[UP_C_1] + dutyCycle_mmc02[UP_C_2] +
    dutyCycle_mmc02[UP_C_3] + dutyCycle_mmc02[UP_C_4]) * V_C_UP_ave;

V_A_LW_PROMED = (dutyCycle_mmc02[LW_A_1] + dutyCycle_mmc02[LW_A_2] +
    dutyCycle_mmc02[LW_A_3] + dutyCycle_mmc02[LW_A_4]) * V_A_LW_ave;
V_B_LW_PROMED = (dutyCycle_mmc02[LW_B_1] + dutyCycle_mmc02[LW_B_2] +
    dutyCycle_mmc02[LW_B_3] + dutyCycle_mmc02[LW_B_4]) * V_B_LW_ave;
V_C_LW_PROMED = (dutyCycle_mmc02[LW_C_1] + dutyCycle_mmc02[LW_C_2] +
    dutyCycle_mmc02[LW_C_3] + dutyCycle_mmc02[LW_C_4]) * V_C_LW_ave;
```

Figura 139 - Cálculo de la tensión promedio en cada rama

Finalmente, los valores de tensión calculados en la Figura 139 se envían a los vectores de las fuentes de tensión programables. Como se ve en la Figura 140,

se utiliza la función *writeVector*. De esta manera, las fuentes de tensión trifásica aportan las tensiones promediadas de cada rama, actuando como un convertidor MMC.

```
// ENVIAR TENSIONES PROMEDIO A LAS FUENTES DE TENSION EXTERNAS
writeVector(fuente_externa_UP, 0, V_A_UP_PROMED, V_B_UP_PROMED, V_C_UP_PROMED);
writeVector(fuente_externa_LW, 0, V_A_LW_PROMED, V_B_LW_PROMED, V_C_LW_PROMED);
```

Figura 140 - Envío de tensiones promedio a fuentes de tensión promediadas

Es importante comentar el hecho de que el convertidor realista, al ser ejecutado por el core#1, los cálculos que realiza el bloque MMC son menores debido a que el periodo de ejecución es de 4 microsegundos. En el caso del convertidor promediado, se ejecuta mediante el controlador, por lo que tiene un periodo de ejecución de 200 microsegundos, por lo tanto, al integrar de la manera que se muestra en la Ecuación 34, se produce un mayor error a causa de tener periodos de ejecución más largo.

3.3.5.2.- Convertidor promediado con carga aislada

En este apartado se estudia el comportamiento del convertidor promediado conectado a una carga resistiva. Al igual que en el caso del convertidor realista, en el lado de corriente continua hay baterías, ya que no se encuentra conectada la línea en el lado de continua.

La planta del convertidor promediado conectado a una carga resistiva es la que se ve en la Figura 141.

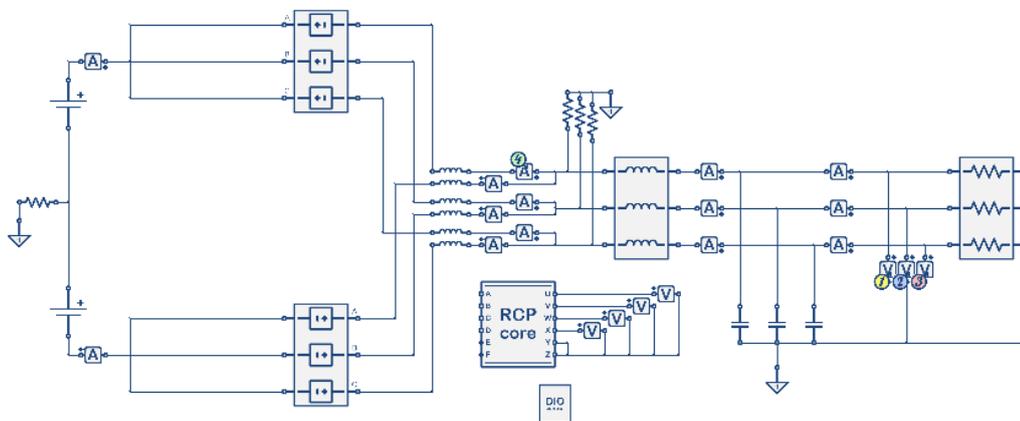


Figura 141 - Planta del convertidor promediado conectado a carga resistiva en el simulador en tiempo real

En la Figura 142 se observan las tensiones de salida del convertidor. Como se puede observar, esta gráfica se encuentra a la misma escala de 600 voltios por división en el eje y, y, por otro lado, 4 milisegundos por división en el eje x, con el objetivo de poder comparar estas tensiones con las del convertidor realista y así demostrar la validez del convertidor promediado. Observando las tensiones de salida en la gráfica, se puede ver como en este caso se superan

ligeramente los 1200 voltios de amplitud. Esta pequeña diferencia se debe a que el convertidor promediado no presenta resistencias internas. Por lo tanto, se puede decir que, en cuanto a tensiones, el comportamiento del convertidor promediado es equivalente al del convertidor realista cuando se eliminan las resistencias internas.

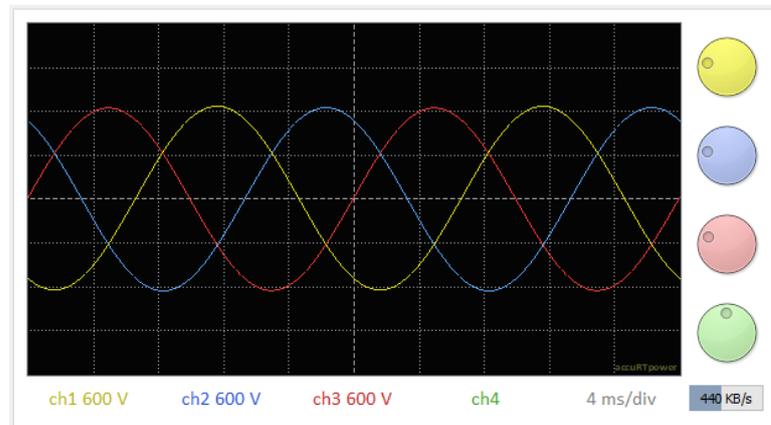


Figura 142 - Tensiones de salida del convertidor promediado conectado a carga resistiva

A continuación, en la Figura 143 se muestran las corrientes a la salida del convertidor. Lo primero que llama la atención de las corrientes del convertidor promediado es que no presentan ningún rizado. Esto se debe a que todas las celdas toman siempre el valor promedio de tensión correspondiente a su rama, a diferencia del convertidor realista, en el que cada condensador tiene su tensión independiente de las demás. A pesar de esta diferencia, las corrientes que aporta el convertidor promediado son admisibles, ya que el convertidor realista eliminando el rizado con un filtro de condensadores presentaría una salida similar a la de la Figura 143.

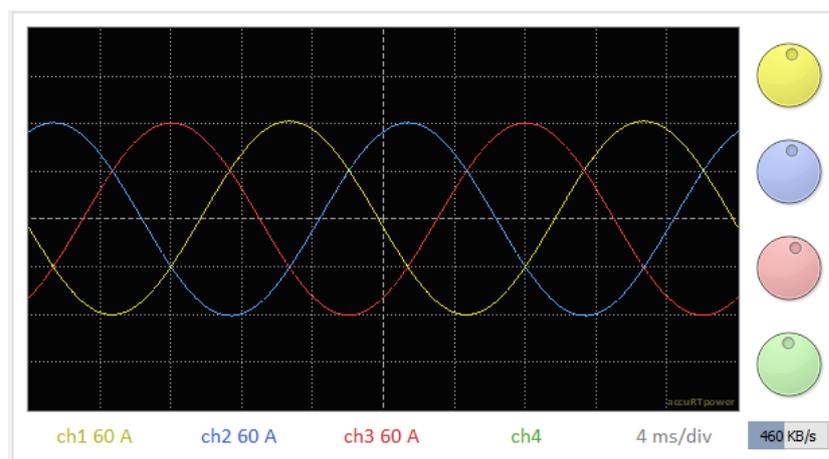


Figura 143 - Corrientes de salida del convertidor promediado conectado a carga resistiva

Finalmente, en la Figura 144, se pueden observar las corrientes de salida de cada rama de una misma fase junto a la suma de ambas corrientes, que resulta

ser la corriente de salida del convertidor. Se puede ver que el resultado es el esperado, similar a la misma gráfica del convertidor realista, pero sin el rizado.

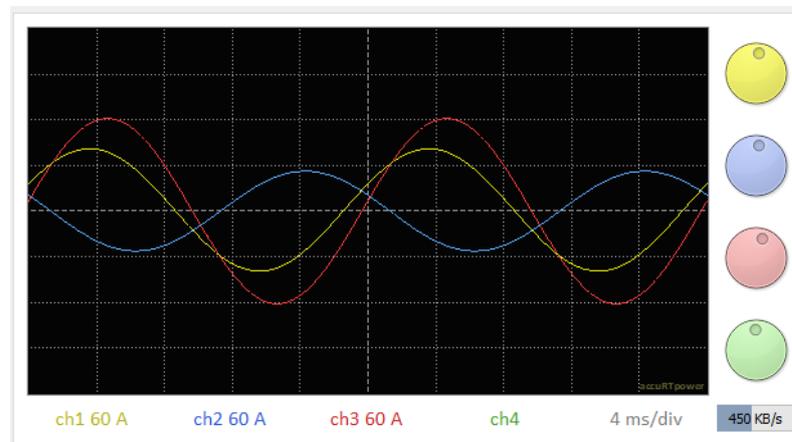


Figura 144 - Corrientes de salida de la rama superior (amarillo), la rama inferior (azul) y la suma de ambas (roja) del convertidor promediado conectado a carga resistiva

3.3.5.3.- Convertidor promediado conectado a red

En este apartado se estudia la planta del convertidor promediado conectado a red, que es la que se ve en la Figura 145. Al igual que en los casos anteriores, el convertidor se encuentra conectado a baterías en ausencia de la línea.

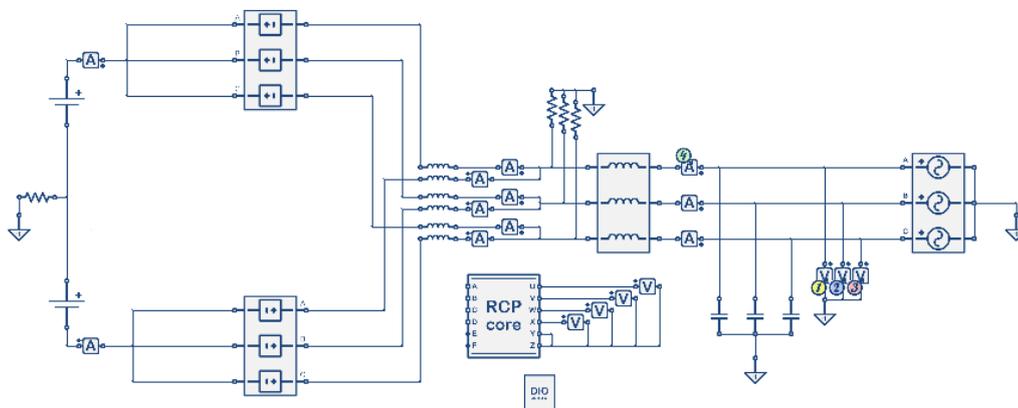


Figura 145 - Planta del convertidor promediado conectado a Red en el simulador en tiempo real

En la imagen de la Figura 146 se observan las tensiones de salida del convertidor cuando está conectado a la red. Al igual que en los casos anteriores, la escala en el eje y es de 600 V por división y, en el eje x, de 4 milisegundos por división, para poder comparar estas tensiones con las de los otros casos. Se puede ver que las tensiones son las esperadas, iguales a las de la red a la que está conectado el convertidor.

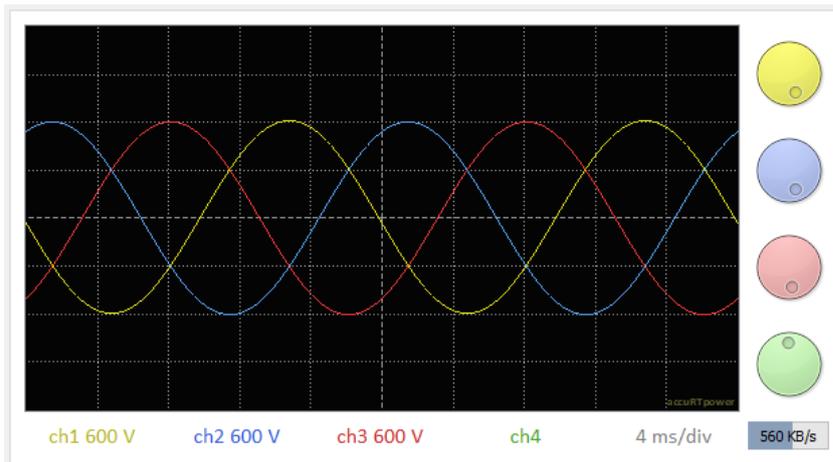


Figura 146 - Tensiones de salida del convertidor promediado conectado a Red

A continuación, en la Figura 147 se muestran las corrientes a la salida del convertidor. Se puede observar como las corrientes no presentan rizado, por los mismos motivos que se han comentado en el caso anterior. La diferencia en la amplitud de las corrientes se debe a que las potencia que circulan en los dos casos es diferente.

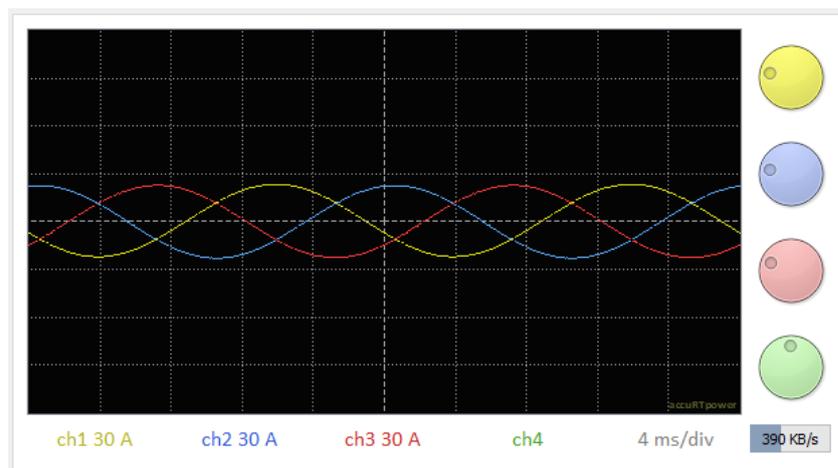


Figura 147 - Corrientes de salida del convertidor promediado conectado a Red

Finalmente, en la Figura 148, se pueden observar las corrientes de salida de cada rama de una misma fase junto a la suma de ambas corrientes, que resulta ser la corriente de salida del convertidor. Se puede ver que el resultado es similar a la misma gráfica del convertidor realista, pero sin el rizado.

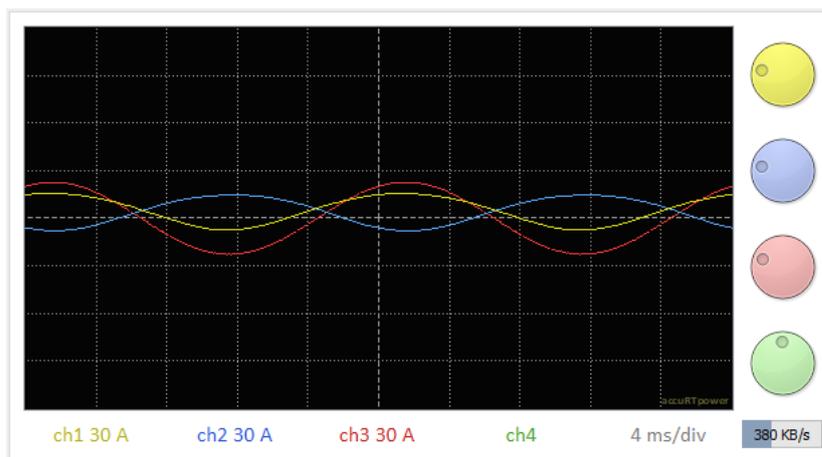


Figura 148 - Corrientes de salida de la rama superior, la rama inferior y la suma de ambas del convertidor promediado conectado a Red

3.3.6.- *Convertidor realista vs Convertidor promediado*

En los apartados anteriores se han visto las salidas que generan tanto el convertidor realista como el convertidor promediado cuando se encuentran conectados a una carga resistiva y cuando se encuentran conectados a red.

Se puede ver que tanto las tensiones como las corrientes de salida son similares. La diferencia más notoria es que las corrientes en el convertidor promediado no presentan rizado. Esto se debe a que, como ya se ha comentado, el convertidor promediado en todo momento toma valores promedio entre las cuatro celdas de cada rama.

Por lo tanto, el comportamiento de ambos convertidores se puede considerar válido para la simulación de la línea HVDC.

3.3.7.- *Interconexión de los dos convertidores*

Una vez validado el funcionamiento de los dos convertidores, solamente queda interconectarlos mediante la línea en corriente continua. Para ello existen diferentes alternativas, aunque finalmente se ha elegido una opción teniendo en cuenta la cantidad de recursos disponibles.

La opción más completa sería utilizar dos condensadores en serie a la salida del convertidor por el lado de continua, con un neutro entre los dos condensadores con una resistencia muy grande de por medio, como la planta que aparece en la Figura 149. Para ello serían necesario cuatro condensadores, una bobina y dos resistencias. En total son siete elementos de los 32 que hay como límite.

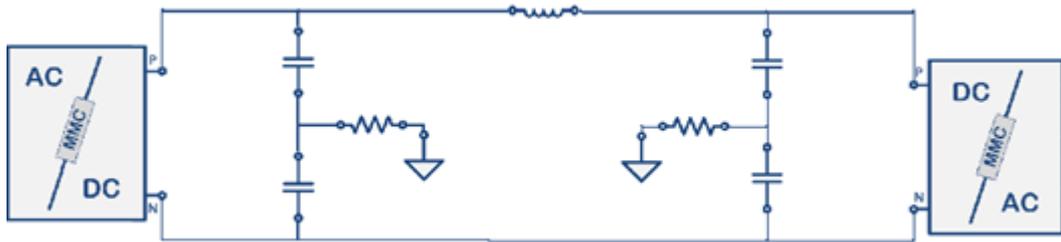


Figura 149 - Mejor opción para interconectar los convertidores

Como solución final se ha elegido poner un solo condensador en la salida de cada convertidor, pero equivalente a los dos que habría en la opción que se ha explicado antes. De esta forma se ahorran dos elementos. Tampoco se ponen los neutros en el punto medio, por lo que tampoco son necesarias las resistencias entre el neutro y el circuito, ahorrando otros dos elementos. De esta forma solo se necesitan tres elementos para interconectar los convertidores. En total se han utilizado 28 de los 32 elementos disponibles, pero, aunque parezca un margen holgado, el neutro de las dos redes necesita una ecuación para que se cumpla la condición de que la suma de las tres corrientes es igual a cero. Estas ecuaciones cuentan también como un elemento. Por lo tanto, realmente es como si estuvieran ocupados 30 de los 32 elementos disponibles.

En la Figura 150 se ve la planta definitiva, con la interconexión tal como se acaba de indicar y cada convertidor conectado a su propia red en corriente alterna.

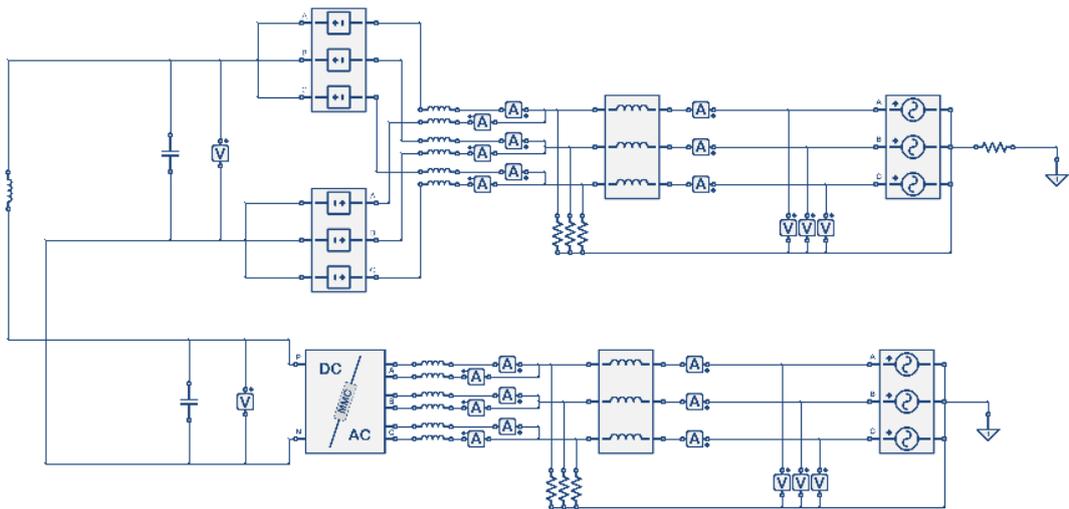


Figura 150 - Planta de la línea HVDC en el simulador en tiempo real

3.3.8.- Selección de las constantes de los reguladores PI

En este apartado se procede a justificar los valores de las constantes de los reguladores proporcional-integral que se han aplicado en el código del controlador. En total hay tres reguladores PI:

- En primer lugar, hay un PI en la aplicación del método del dqPLL, para regular el valor de theta. Los valores de las constantes de este PI se pueden ver en la Figura 151 con la etiqueta *_SYNC*.
- Después se ha visto el PI con el que obtener las referencias de tensión del convertidor, para ajustar la sincronización con la red. Las constantes correspondientes a este PI son las que tienen la etiqueta *_IODQ* en la Figura 151.
- Por último, el funcionamiento en modo *Slave* tiene asociado otro regulador PI para regular el valor de la tensión en corriente continua. Las constantes de este PI se nombran en la Figura 151 con la etiqueta *V_DC*.

	KP_SYNC	TS_KI_SYNC	KP_IODQ	TSXKI_IODQ	KPV_DC	TSKIV_DC
MMC01	1	0,02	0,2	0,002	1	0,0012
MMC02	1	0,02	0,2	0,002	0,6	0,0025

Figura 151 - Valores de las constantes de los reguladores PI (elaboración propia)

Las constantes del regulador PI que se ha nombrado en primer lugar se han elegido de forma arbitraria, la constante proporcional vale 1 y la integral 100, pero se encuentra multiplicada por el periodo de 200 microsegundos. Al funcionar correctamente con estos valores, se ha decidido darlos por válidos.

Para ajustar los dos reguladores PI restantes, se han realizado múltiples simulaciones en las que se ha programado un cambio de consigna en cada periodo, de tal forma que se puede observar la respuesta ante un cambio repentino en la referencia. Los valores elegidos son los que generan una reacción relativamente rápida, sin sobre pico y lo suficientemente amortiguada. Los valores finalmente seleccionados se pueden ver en la Figura 151.

En el caso del regulador que se utiliza para obtener las referencias de tensión del convertidor, la consigna alterada para obtener las constantes es la *lo_d_ref*.

Por otro lado, las constantes del regulador del modo *Slave* se han obtenido provocando un cambio en la tensión en el lado de corriente continua, utilizando una fuente de corriente programable desde el RCPcore en paralelo con los condensadores del convertidor.

3.3.9.- Evaluación de recursos disponibles tras la simulación

Antes de observar los resultados de la simulación en tiempo real, en este apartado se va a evaluar la cantidad de recursos que se han requerido

finalmente, de entre los disponibles que se han mencionado en el apartado 2.11.4.- Limitaciones.

En cuanto a la ejecución de la planta, de la que se encarga el core#1, como muestra la Figura 152, son necesarias 344 instrucciones, se necesitan más de 2.92 microsegundos de los 4 microsegundos que dura el periodo de ejecución de la planta, y, quedan libres:

- 53 variables globales de las 96 en total.
- 43 constantes de las 64 posibles.
- 16 multiplicadores de los 64 que hay en total.

Core #1 needs 344 code words, it has 53R/43K/16P free registers and runs every 4 us (requires more than 2.92 us).
Core #2 needs 6367 code words, it has 7R/3K/48P free registers and must run every 200 us.

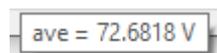
Figura 152 - Información de la simulación

Con respecto a la ejecución del controlador, por el core#2, tal como se puede ver en la Figura 152, son necesarias 6367 instrucciones, se emplean 72,7 microsegundos (ver Figura 153) de los 200 microsegundos que dura el periodo del controlador y, han quedado sin utilizar:

- 7 variables globales de 96.
- 13 variables locales de 32, teniendo en cuenta que se han definido 10 variables locales y se utilizan dos funciones que tienen 9 argumentos cada una. Estos argumentos de función ocupan las variables locales en el momento en el que se utilizan y después las liberan, por lo tanto, se encuentran ocupadas 19 variables locales.
- 3 constantes de 64.
- 48 multiplicadores de 64.

Los recursos comunes para los dos cores que quedan disponibles son los siguientes:

- 4 de los 24 *Analog Output*.
- 12 *Digital Output* de 36.
- 3 multiplicadores vectoriales (resistencias, bobinas, condensadores) de 32 en total.



ave = 72.6818 V

Figura 153 - Tiempo promedio de ejecución del loop

En la gráfica de la Figura 154 se puede ver la salida de la función *loopMicroseconds*, que cuenta el tiempo que tarda en ejecutarse el código completo del *loop* en cada periodo. De esta manera se puede ver que el tiempo de ejecución varía de un periodo a otro. Esta variación se debe a que en cada

periodo se toman diferentes decisiones en los condicionales *if* y en los bucles *while*, siendo unos caminos más largos que otros.

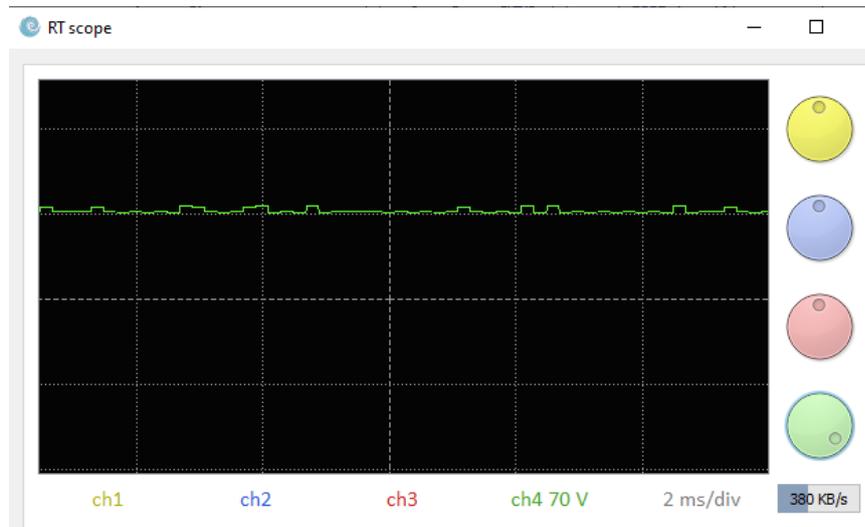


Figura 154 - Tiempo de ejecución del loop en tiempo real

Como se puede ver, hay algunos recursos que se encuentran lejos del límite porque no se han necesitado tantos, como por ejemplo las variables globales en la planta, mientras que hay otro tipo de recursos que son los que marcan las capacidades de la simulación, como por ejemplo el número de constantes en el controlador, con los que se ha tenido que lidiar para conseguir realizar los objetivos.

3.3.10.- *Resultados de la Simulación en tiempo real*

Llegado hasta este punto, ya se ha mostrado la planta objetivo de la simulación en tiempo real, explicado cómo funciona el código del controlador de la planta y se ha validado el funcionamiento tanto del convertidor realista como del convertidor promediado. Por lo tanto, en este apartado se van a exponer los resultados que se pueden extraer de la simulación en tiempo real.

Para la presentación de los resultados de la simulación, se ha dispuesto el convertidor realista en el modo *Master* y el convertidor promediado en modo *Slave*. El convertidor realista tiene una potencia activa de referencia tal que absorbe un megavatio de la red a la que se encuentra directamente conectado. Esta potencia es transmitida a la otra red por medio de la línea HVDC a 3200 V en corriente continua.

Bajo estas condiciones de operación, se extraen los resultados que vienen a continuación.

3.3.10.1.- *Resultados en cuanto a corrientes en alterna*

En las siguientes figuras (Figura 155 y Figura 156) se puede ver la forma de las corrientes en alterna de ambos convertidores. Como se puede observar, la

amplitud de las corrientes cuando se transmite un megavatio es en torno a 600 amperios pico, o lo que es lo mismo 425 amperios eficaces.

Más allá de destacar la diferencia sobre el rizado que tienen las corrientes en el convertidor realista, se puede observar como la amplitud de las corrientes del convertidor realista es ligeramente mayor que las del convertidor promediado. El motivo de esta diferencia se debe a que el convertidor realista se ejecuta cada 4 microsegundos por el *core#1*, reduciendo el error cometido en comparación con el convertidor promediado, que, al ejecutarse cada 200 microsegundo, el error es mayor. Como se puede ver, la diferencia es pequeña, por lo que se puede considerar aceptable para cumplir el objetivo del trabajo.

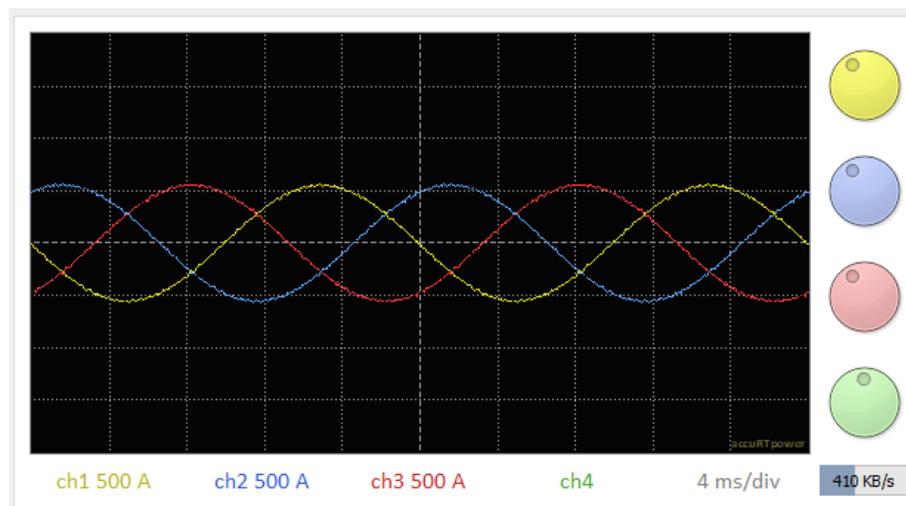


Figura 155 - Corrientes AC del convertidor realista

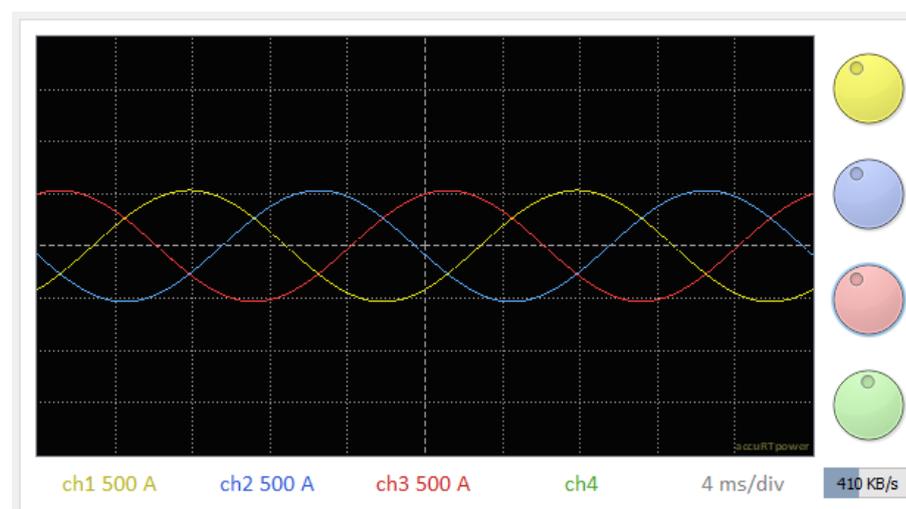


Figura 156 - Corrientes AC del convertidor promediado

Este comportamiento se puede considerar aceptable, ya que la diferencia entre ambas corrientes es pequeña y permite realizar la simulación de forma estable.

A continuación, se muestra la tasa de distorsión armónica (THD) de las corrientes de salida del convertidor promediado, en la Figura 157, y, en la Figura 158, se muestra la tasa de distorsión armónica de las corrientes de salida del convertidor realista. Como se puede ver, la tasa de distorsión armónica es menor en el convertidor promediado que en el realista. Esto se debe a que el convertidor promediado no presenta rizado al utilizar en todo momento un valor promedio de las tensiones de los condensadores en cada rama.

Promediado	I_{o_A}	I_{o_B}	I_{o_C}
THD (%)	0,1	0,1	0,09

Figura 157 - Tasa de distorsión armónica de las corrientes de salida del convertidor promediado

Realista	I_{o_A}	I_{o_B}	I_{o_C}
THD (%)	1,59	1,6	1,59

Figura 158 - Tasa de distorsión armónica de las corrientes de salida del convertidor realista

Estos valores de tasa de distorsión armónica se pueden considerar admisibles ya que permanecen por debajo del 5% y la simulación puede llevarse a cabo con unas señales de tensión y corriente estables.

3.3.10.2.- Resultados en cuanto a tensiones en alterna

A continuación, en la Figura 159 y Figura 160, se muestran las tensiones en alterna del convertidor realista y del convertidor promediado, respectivamente. Se observa que en ambos casos las tensiones son idénticas. Esto es así porque para esta simulación sendas redes se han configurado con las mismas tensiones. Además, los convertidores han sido diseñados para aportar en el lado de corriente alterna una tensión pico de 1200 V, o lo que es lo mismo en tensión en valor eficaz, unos 850 V.

El diseño de los convertidores, de tan solo 4 celdas por rama y 800 voltios en cada celda, condiciona el valor de tensión en corriente alterna, ya que para aumentar la tensión en alterna se requerirían de un número mayor de celdas y/o una tensión mayor en cada celda.

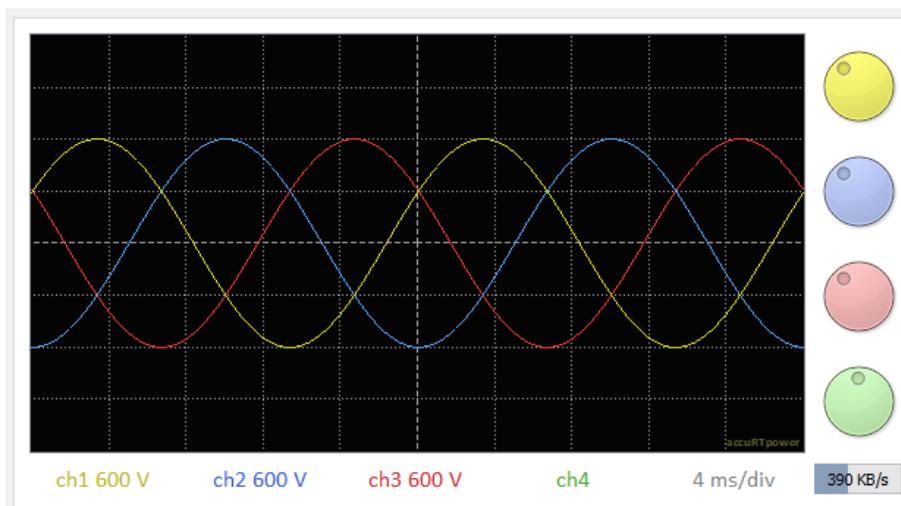


Figura 159 - Tensiones AC del convertidor realista

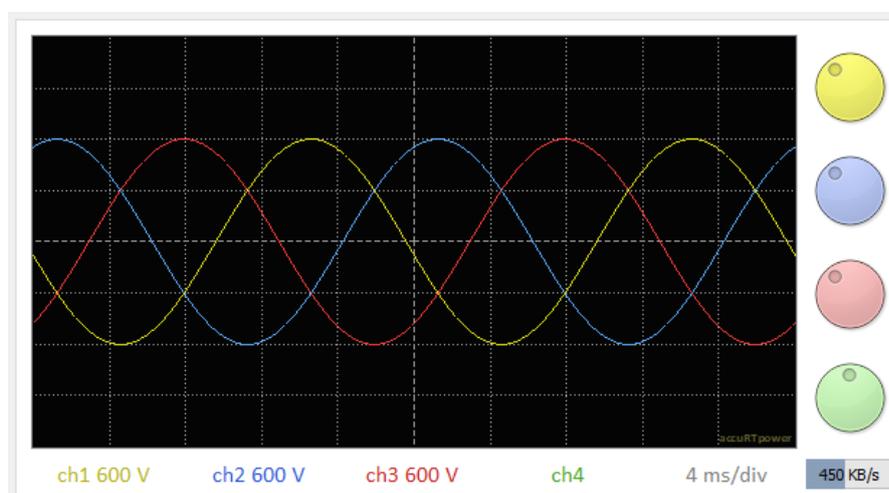


Figura 160 - Tensiones AC del convertidor promediado

Con estos resultados de tensiones y corrientes se puede confirmar el correcto funcionamiento de la simulación calculando la potencia transmitida por la línea. El valor pico de las corrientes se estima a través de las gráficas de la Figura 155 y la Figura 156, por lo que no es un valor del todo exacto que añade cierto error al cálculo de la potencia.

Ecuación 37 - Cálculo de la potencia absorbida de la red por el convertidor realista

$$P = \frac{3}{2} V_p \cdot I_p = \frac{3}{2} 3200 \cdot 600 = 1080 \text{ kW}$$

Ecuación 38 - Cálculo de la potencia entregada a la red por el convertidor promediado

$$P = \frac{3}{2} V_p \cdot I_p = \frac{3}{2} 3200 \cdot 550 = 990 \text{ kW}$$

En la Ecuación 37 se pueden ver los cálculos para obtener la potencia absorbida de la red de origen por el convertidor realista en valor absoluto,

mientras que en la Ecuación 38, se muestran los cálculos para obtener la potencia inyectada a la red de destino por el convertidor promediado. Como se puede observar, en ambos casos se obtiene en torno a un megavatio, por lo que se puede decir que la simulación cumple con lo establecido.

3.3.10.3.- Comparación entre tensiones y corrientes de ambos convertidores

En la Figura 161 se observa en azul y verde las tensiones del convertidor promediado y del convertidor realista, respectivamente. De la misma manera se encuentran en amarillo y rojo las corrientes del convertidor promediado y realista, respectivamente. El desfase entre las tensiones de ambos convertidores se debe únicamente a la configuración de la red a la que se encuentra conectado el convertidor promediado, a la que se le ha aplicado un desfase para poder observar ambas señales, ya que son prácticamente idénticas.

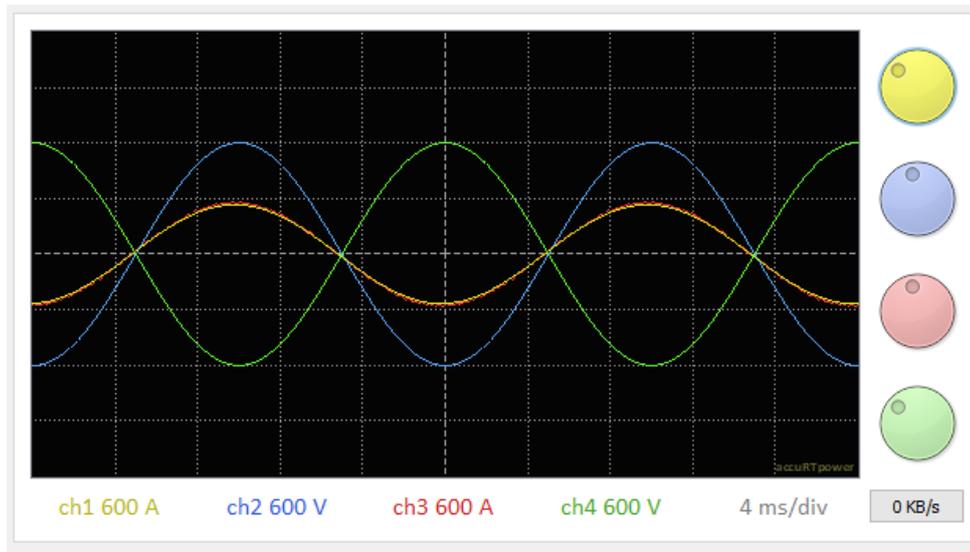


Figura 161 - Comparación entre corrientes y tensiones de los dos convertidores

Por otro lado, al observar las corrientes es posible notar la pequeña diferencia entre ambas, que se observa mejor en la Figura 162. Esta diferencia, como ya se ha explicado anteriormente, se debe a la diferencia con respecto al error cometido en los cálculos de cada convertidor. El convertidor realista se ejecuta cada 4 microsegundos, minimizando mucho el error, mientras que el convertidor promediado al ejecutarse cada 200 microsegundos comete un error mayor.

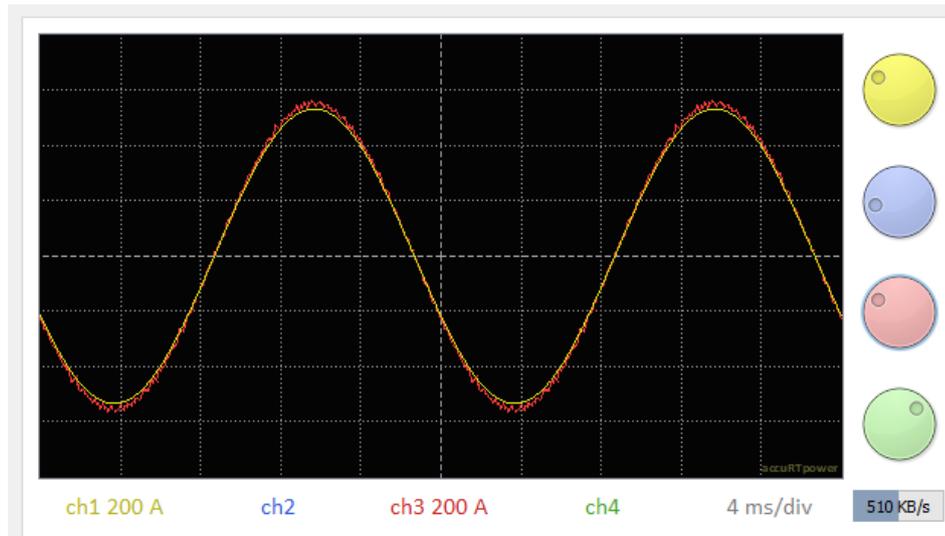


Figura 162 - Detalle de las corrientes. Comparación entre los dos convertidores (corriente del realista en rojo, corriente del promediado en amarillo)

3.3.10.4.- Tensión en la línea HVDC

La tensión nominal en la línea es de 3200 voltios en corriente continua, tal como se puede ver en la Figura 163 y Figura 164. Como se observa, en el convertidor realista la tensión presenta pequeñas oscilaciones debido a que en el modo *Master* no se controla esta tensión. Por el contrario, en el convertidor promediado la tensión no presenta estas oscilaciones ya que al funcionar en modo *Slave*, esta tensión está controlada para permanecer en el valor nominal. Estos 3200 voltios en corriente continua se deben a que, al tener celdas de 800 V, y, que en todo momento debe haber cuatro celdas encendidas en cada fase, es decir, entre la rama superior e inferior de una misma fase, en total debe haber cuatro celdas insertadas. La tensión que aportan cuatro celdas de 800 voltios cada una resultan ser 3200 voltios. El condensador (o condensadores) a la salida del convertidor en el lado de corriente continua, se encarga de mantener la tensión lo más estable posible.

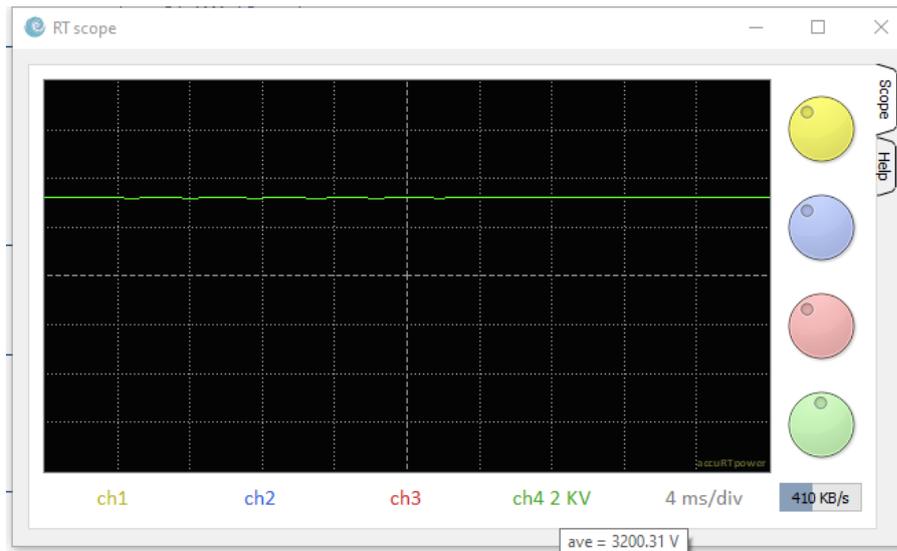


Figura 163 - Tensión en DC en terminales del convertidor realista

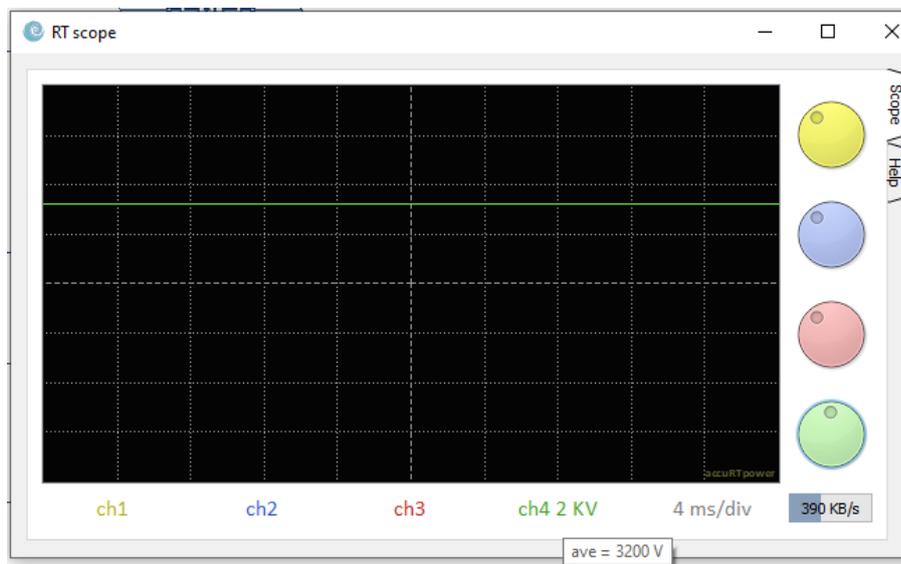


Figura 164 - Tensión en DC en terminales del convertidor promediado

3.3.10.5.- Cálculo de corrientes circulantes

La corriente que circula por la rama superior de cualquiera de las fases se compone por la mitad de la corriente de salida más una corriente que recibe el nombre de corriente circulante (ver Figura 165). Por otro lado, la corriente que circula por la rama inferior es igual a la corriente circulante menos la mitad restante de la corriente de salida. Esta corriente circulante determina la potencia que entra o sale de los respectivos convertidores.

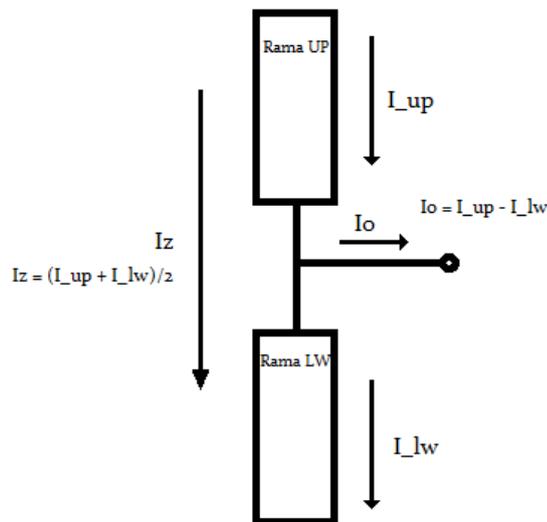


Figura 165 - Corrientes en el interior del convertidor

Para realizar el cálculo de las corrientes circulantes en ambos convertidores, se ha utilizado el código que aparece en la Figura 166. Como se puede ver, la corriente circulante en una fase es igual a la suma de la corriente de la rama superior y la corriente de la rama inferior dividido entre dos.

```
// Calculo de corrientes circulantes:
IZ_A = 0.5 * IupA + 0.5 * IlwA;
IZ_B = 0.5 * IupB + 0.5 * IlwB;
IZ_C = 0.5 * IupC + 0.5 * IlwC;

IZ = IZ_A + IZ_B + IZ_C;
```

Figura 166 - Cálculo de las corrientes circulantes

3.3.10.5.1.- Corrientes circulantes del convertidor realista

Las corrientes circulantes presentes en el convertidor realista durante la simulación en tiempo real de la línea HVDC se pueden observar en la Figura 167. Al funcionar el convertidor realista en modo *Master*, está absorbiendo potencia de la red, por lo que las corrientes circulantes tienen componente de continua negativa, es decir, por debajo de cero. Observando la gráfica se podría decir que cuando el convertidor realista está absorbiendo un megavatio de la red, las corrientes circulantes en dicho convertidor poseen una componente de corriente continua de unos 100 amperios. En la Figura 168 se pueden ver las corrientes circulantes tras anular la componente de continua. De esta manera se puede observar de forma más clara que estas corrientes son de 100 Hz, que son debido al uso de la técnica de PWM con portadoras *Phase Disposition*.

Al calcular la potencia que sale del convertidor hacia la línea HVDC, como se ve en la Ecuación 39, con la tensión en corriente continua y la componente de continua de las corrientes circulantes, se obtiene que salen unos -320 kW en

cada fase, por lo tanto, la potencia total que se transmite es en torno a un megavatio, es decir, lo esperado.

Ecuación 39 - Cálculo de potencia que sale de cada fase del convertidor realista hacia la línea

$$P_{Sale} = V_{DC} \cdot I_{Z_{DC}} = 3200 \text{ V} \cdot (-100) = -320 \text{ kW}$$

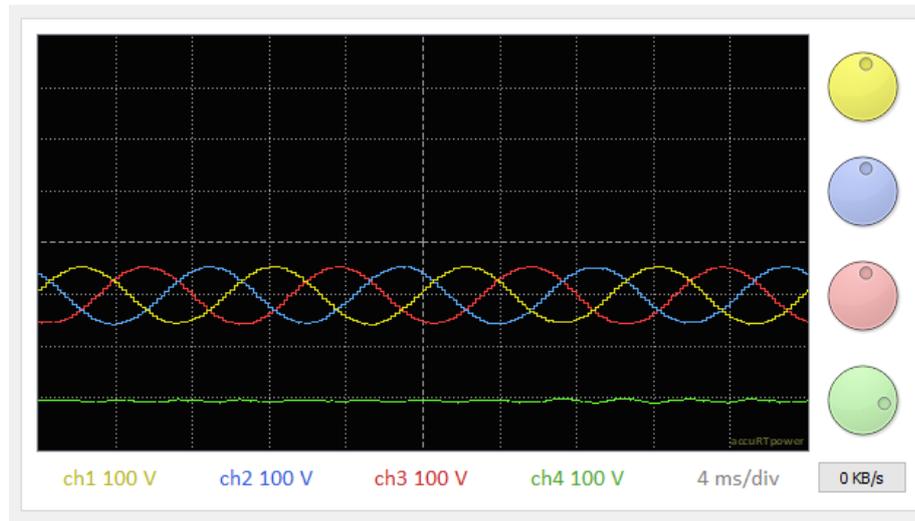


Figura 167 - Corrientes circulantes del convertidor realista. En amarillo la de la fase A, azul la de la fase B, rojo la de la fase C y en verde la suma de las tres

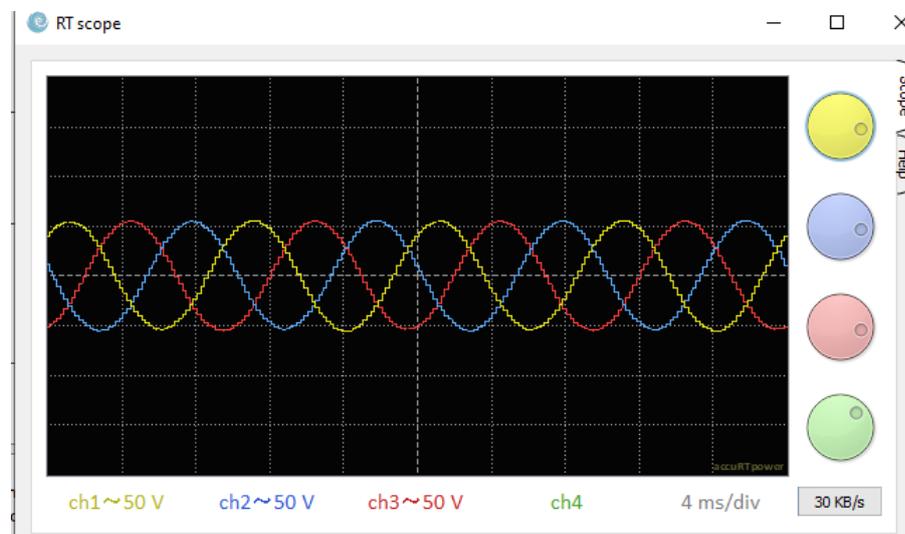


Figura 168 - Corrientes circulantes del convertidor realista sin componente de continua

En la figura que se muestra a continuación (Figura 169), se puede ver el valor medio de la línea verde en la gráfica de la Figura 167. Este valor alcanza los -305,575 amperios en el convertidor realista. Es igual a la corriente que circula

por la línea. Al multiplicar este valor por la tensión en corriente continua, se obtiene la potencia total transmitida por la línea, que resulta ser un megavatio.

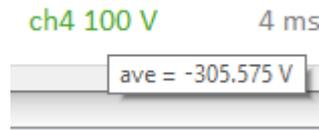


Figura 169 - Valor medio de la suma de las corrientes circulantes de cada fase del convertidor realista

3.3.10.5.2.- Corrientes circulantes del convertidor promediado

En el caso de las corrientes circulantes del convertidor promediado, que se pueden ver en la Figura 170, se puede observar que, al estar inyectando potencia a la red de destino, las corrientes poseen componente de continua positiva, es decir, por encima de cero. Mirando la gráfica se podría señalar que esta componente de continua es de unos 100 amperios cuando el convertidor inyecta un megavatio a la red. Al igual que se ha hecho con el convertidor realista, se elimina la componente de continua para observar la frecuencia de las corrientes (Figura 171), que resulta ser también de 100 Hz, que se debe al uso de PWM con *Phase Disposition*.

La potencia que recibe el convertidor promediado se puede calcular a través de la componente de corriente continua de las corrientes circulantes y la tensión en corriente continua. Como se muestra en la Ecuación 40, la potencia que entra en cada fase es de 320 kW, por lo tanto, la potencia total entrante de las tres fases ronda el megavatio, que es la potencia esperada.

Ecuación 40 - Cálculo de potencia que entra de cada fase del convertidor promediado hacia la línea

$$P_{\text{Entra}} = V_{DC} \cdot I_{Z_{DC}} = 3200 \text{ V} \cdot 100 = 320 \text{ kW}$$

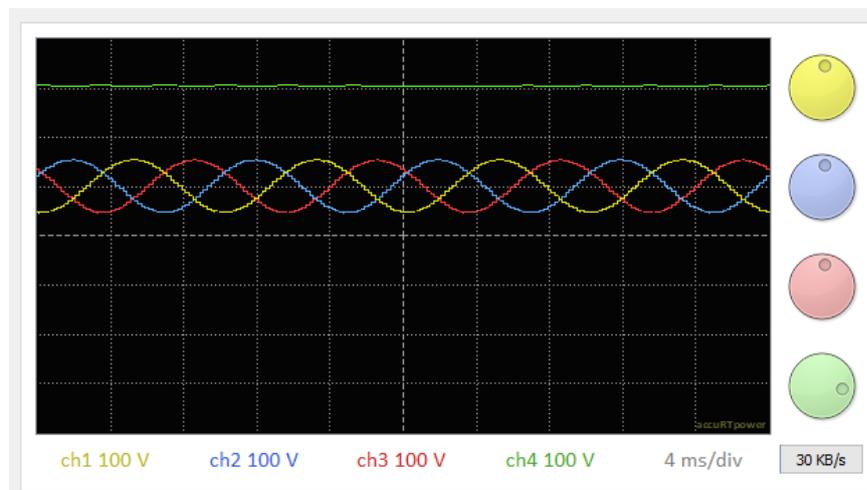


Figura 170 - Corrientes circulantes del convertidor promediado En amarillo la de la fase A, azul la de la fase B, rojo la de la fase C y en verde la suma de las tres

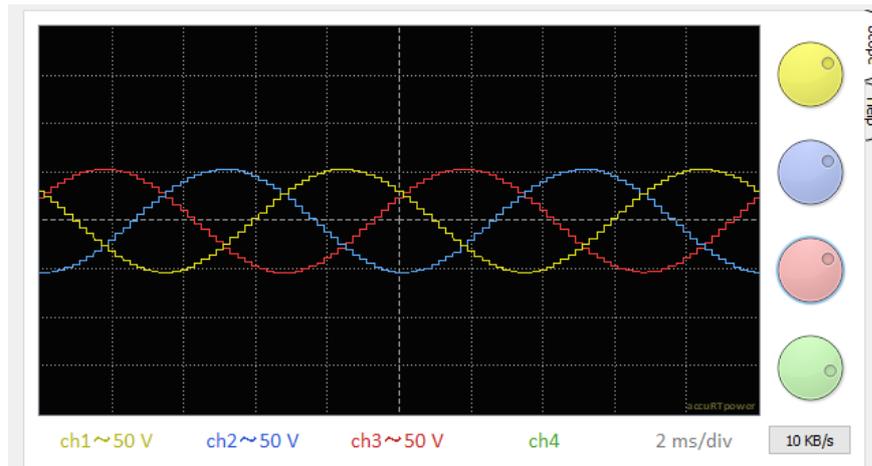


Figura 171 - Corrientes circulantes del convertidor promediado sin componente de continua

En la Figura 172 se muestra el valor medio de la línea verde de la gráfica de la Figura 170. Como se puede ver, dicho valor medio es de 305,541 A, que es exactamente el valor opuesto al que ha salido en el convertidor realista. Es igual a la corriente que circula por la línea. Al multiplicar este valor por la tensión en corriente continua se obtiene la potencia total que llega al convertidor promediado, resultando ser un megavatio.

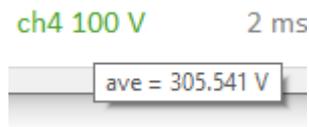


Figura 172 - Valor medio de la suma de las corrientes circulantes de cada fase del convertidor promediado

4.- CAPÍTULO 4: REPERCUSIONES DEL TFG

La electricidad es un bien de consumo básico e imprescindible en nuestras vidas. A su vez, el cambio climático nos afecta de una forma directa y una forma de combatirlo es dejar de consumir combustibles fósiles a la hora de generar nuestra energía. Pero las energías renovables son muy variables y no podemos controlar cuanto ni cuando las podemos generar, además, los sistemas de almacenamiento de energía todavía no son una opción demasiado asequible, por lo menos en la medida que requeriría el sistema para dejar de depender de los combustibles fósiles. Por lo tanto, una buena alternativa podría ser la interconexión de los sistemas eléctricos de los distintos países para que cuando en unos países se disponga de excedentes de energía, se pueda transportar dicho excedente hacia otros países que en ese momento tengan escasez. La tecnología del HVDC es fundamental para la interconexión entre países para aprovechar las energías renovables, conexión con parques eólicos offshore, ya que permite conectar sistemas que funcionen a distinta frecuencia y no transmite las fallas de un sistema al resto, como se ha comentado a lo largo de este trabajo. Además, las energías renovables son más baratas de producir, por lo que la consolidación de este tipo de generación implica una reducción del precio de la electricidad. El HVDC también es fundamental para llevar energía a islas, aumentando la estabilidad de la red de estos pequeños sistemas, evitando o reduciendo en gran medida cortes de suministro debido a inestabilidad de la red, por lo que los consumidores de las islas recibirían una mayor calidad de servicio. Como se puede ver, las repercusiones sociales de este trabajo van ligadas a las ambientales.

Además de las repercusiones ambientales de este trabajo que se acaban de comentar, podemos sumarle el hecho de que el tema del mismo es una simulación en tiempo real, que permite estudiar el comportamiento de la línea y de los convertidores sin la necesidad de montar una línea de prueba, lo que supone un ahorro de recursos materiales, económicos y de energía. Por otro lado, con respecto al HVDC, por lo general se utilizan cables submarinos o subterráneos y se busca que tengan el mínimo impacto. En líneas aéreas, el HVDC tiene menor impacto visual que una línea de corriente alterna, ya que posee menor número de cables (dos como mucho) y los apoyos son de menor tamaño. También presenta menor efecto corona, menores interferencias con ondas de radio y menor ruido en condiciones de lluvia.

Una simulación en tiempo real tiene multitud de repercusiones económicas. Ya se ha comentado antes que una simulación en tiempo real permite estudiar el comportamiento de la línea y los convertidores sin la necesidad de montar una línea de prueba. Esto supone el ahorro de los materiales de la instalación de prueba, la energía consumida para construirla y el coste de la obra. Además, la simulación permite probar la línea sin riesgo a estropear los elementos de los

convertidores, también se pueden hacer todo tipo de pruebas, lo que se traduce en un mejor estudio del comportamiento de la línea, por lo que aumenta la fiabilidad de la línea ya que se han podido estudiar todos los escenarios antes de construir la línea, permitiendo así anticiparse a posibles problemas.

La simulación en tiempo real de una línea HVDC también tiene repercusiones sobre la seguridad y salud sobre las personas. Durante un cortocircuito se pueden dar situaciones de riesgo para las personas, tanto un operario que trabaje en una línea de prueba como cualquier persona que pueda estar cerca de una línea de transporte una vez instalada. Con la simulación, por una parte, no pondría en riesgo a nadie durante todas las pruebas que se realicen sobre el modelo de la línea. Y, por otro lado, pueden realizarse multitud de pruebas en un corto periodo de tiempo y de forma económica, lo que permite que la línea esté preparada, en la medida de lo posible, para cualquier tipo de evento, tratando de garantizar la seguridad de las personas.

5.- CAPÍTULO 5: CONCLUSIONES

5.1.- Implementación de objetivos propuestos

Alcanzando este capítulo, llega la hora de ver el grado en el que se han conseguido implementar los objetivos de este Trabajo de Fin de Grado. En primer lugar, se evaluarán los objetivos parciales para finalmente acabar con el objetivo principal.

5.1.1.- *Implementación de los objetivos parciales*

Los objetivos parciales son los que han marcado los pasos a seguir en el intento de cumplir con el objetivo principal.

El primer objetivo parcial era realizar un convertidor funcional en Simulink. Gracias a esta herramienta de simulación se ha podido desarrollar paso a paso el convertidor, consiguiendo una respuesta estable como se ha visto en el apartado de resultados de la simulación del convertidor de Simulink.

Después de desarrollar el convertidor en Simulink, el siguiente objetivo consistió en transferir estas ideas al convertidor en tiempo real, construyendo en el simulador un convertidor conectado a una carga resistiva. Debido a las limitaciones del simulador en tiempo real, entre otras diferencias como la diferencia de lenguaje de programación, el código del controlador del simulador en tiempo real se ha tenido que adaptar. Pero como se puede ver en el apartado del “*Convertidor realista con carga aislada*”, los resultados son admisibles en cuanto a funcionamiento estable.

Una vez incorporado el convertidor al simulador en tiempo real, se debía sincronizar el convertidor la red trifásica antes de poder conectar. Para ello se ha incorporado la sincronización mediante la aplicación del método del dqPLL, y, a través de la regulación de la corriente de salida mediante un PI, teniendo en cuenta el efecto de las bobinas del convertidor.

Tras la sincronización, se pudo conectar el convertidor a la red trifásica. El comportamiento del convertidor conectado a la red, como se ha visto en el apartado “*Convertidor realista conectado a red*”, es adecuado.

Ante las limitaciones de recursos que se han comentado, que no permiten la simulación de dos convertidores completos, se optó por construir un convertidor promediado, probándolo en primer lugar conectado a una carga resistiva. Tal como muestran los resultados, presentes en el apartado “*Convertidor promediado conectado a carga aislada*”, el comportamiento de este convertidor es similar al del convertidor realista.

Después, utilizando el mismo razonamiento que con el convertidor realista, conectar convertidor promediado a la red trifásica. Observando que el

convertidor se comporta de manera correcta en el apartado “*Convertidor promediado conectado a red*”, siendo por lo tanto aceptable para simular la línea HVDC que se pretende.

Finalmente, interconectar ambos convertidores. No sin antes superar una vez más las limitaciones del simulador en tiempo real, ya que no había suficientes elementos disponibles para hacer la conexión ideal.

5.1.2.- Implementación del objetivo principal

Tras la culminación de los objetivos parciales, los dos convertidores se encuentran conectados a sus respectivas redes y conectados entre sí mediante la línea HVDC. El objetivo principal era:

“Conseguir simular la transferencia de potencia a través de la línea HVDC desde una red trifásica hasta otra”

Como se muestra en el apartado 3.3.10.- **Resultados de la simulación en tiempo real**, se ha conseguido simular en tiempo real la transferencia de un megavatio desde la red a la que está conectado el convertidor realista e inyectar esta potencia a la red a la que se encuentra conectado el convertidor promediado. Por lo tanto, se puede concluir que el objetivo principal ha sido cumplido con éxito.

5.2.- Líneas de futuros trabajos

La simulación en tiempo real que se ha llevado a cabo en este trabajo forma parte de un primer paso de cara a simular una línea HVDC con convertidores HVDC. Todavía serían necesarias ciertas mejoras para poder simular una línea HVDC más cercana a la realidad. Para ello se proponen varias líneas de trabajos futuros que se puedan desarrollar a partir de este.

En primer lugar, se podría presentar un estudio de diversas simulaciones, a partir del modelo desarrollado en este trabajo, en las que se transfieran diferentes potencias, con diferente factor de potencia o simulaciones cambiando el sentido del flujo de la potencia, pudiendo ver así la operabilidad completa de la línea.

Como ya se ha comentado durante este trabajo, el convertidor de esta simulación solamente tiene 4 celdas por rama, lo cual no está cerca de un convertidor MMC con aplicación para una línea HVDC. Por lo tanto, otra posible línea de trabajo podría ser crear otra simulación con convertidores ambos convertidores promediados y que tengan más celdas por rama.

Otra posible línea de trabajo podría ser la simulación de la línea HVDC con convertidores utilizando una técnica distinta al PWM, como puede ser el método NLC (*nearest level control*), que tiene un mejor comportamiento en cuanto a armónicos.

También se ha comentado varias veces durante este trabajo que el tratamiento de los armónicos y el rizado no son objetivos de este trabajo. Por lo tanto, se puede proponer como línea de trabajo tratar de reducir o mitigar los armónicos y el rizado de la corriente.

Cuando la potencia fluye a través de una línea de gran longitud, por ejemplo, una línea HVDC punto a punto, el tiempo que puede tardar en fluir la potencia de una estación convertidora hasta la otra es del orden de milisegundos. El simulador en tiempo real es capaz de realizar los cálculos que simulan la transferencia de potencia en un tiempo del orden de microsegundos, es decir, significativamente más rápido que en la realidad. Por lo que, finalmente, se propone colocar un retraso en la transferencia de potencia por la línea HVDC para que tenga un comportamiento más real, es decir, desarrollar un modelo que tenga en cuenta este retraso para conseguir un funcionamiento más real.

6.- CAPÍTULO 6: BIBLIOGRAFÍA

En este apartado se muestra la bibliografía que ha sido consultada para la realización de la memoria de este Trabajo de Fin de Grado. Como se puede ver, han sido consultados libros, páginas web, blogs y artículos de revistas. Las referencias bibliográficas se muestran en el orden en el que han sido utilizadas en el texto. Para ello, se ha utilizado la norma ISO 690 – Referencia numérica.

1. **Planas, Oriol.** Energía Solar Blog. [En línea] 18 de abril de 2020. [Último acceso: 25 de febrero de 2021.] <https://solar-energia.net/blog/guerra-de-las-corrientes-en-eeuu>.
2. **Chan-Ki, Kim, y otros.** *HVDC TRANSMISSION: Power Conversion Applications*. s.l. : John Wiley & Sons, 2009.
3. **ABB.** Corriente Directa de Alto Voltaje (HVDC) | ABB . [En línea] 27 de mayo de 2013. [Último acceso: 5 de junio de 2021.] <https://new.abb.com/south-america/sobre-nosotros/tecnologia/tecnologia-hvdc>.
4. **Grupo Red Eléctrica.** Proyecto Best Paths | Red Eléctrica de España. [En línea] 26 de enero de 2015. [Último acceso: 10 de abril de 2021.] <https://www.ree.es/es/sostenibilidad/proyectos-destacados/proyectos-de-idi-de-contribucion-al-negocio/proyecto-best-paths>.
5. **Grupo Red Eléctrica.** Interconexión Península - Baleares. [En línea] 16 de julio de 2016. [Último acceso: 10 de abril de 2021.] <https://www.ree.es/es/actividades/proyectos-singulares/interconexion-peninsula-baleares>.
6. **Inelfe.** Interconexión eléctrica por el Golfo de Bizkaia. [En línea] 9 de febrero de 2021. [Último acceso: 10 de abril de 2021.] <https://www.inelfe.eu/es/proyectos/golfo-de-bizkaia>.
7. **Grupo Red eléctrica.** Interconexión subterránea España-Francia. [En línea] 20 de febrero de 2015. [Último acceso: 10 de abril de 2021.] <https://www.ree.es/es/actividades/proyectos-singulares/nueva-interconexion-electrica-con-francia>.
8. **Electrical Deck.** Types of HVDC Systems or Links - Monopolar, Bipolar & Homopolar Links. [En línea] 21 de agosto de 2021. [Último acceso: 10 de septiembre de 2021.] <https://www.electricaldeck.com/2021/08/types-of-hvdc-systems-or-links.html>.
9. **TECNOLOGÍA MODULAR MULTILEVEL CONVERTER (MMC) EN SISTEMAS HVdc.** Hoyos, Sebastián, y otros. 1, Medellín: s.n., 2013, Revista Investigaciones Aplicadas, Vol. 7, págs. 37-47.

10. *Apuntes asignatura "Sistemas de Energía Eléctrica". Departamento de Ingeniería Eléctrica de la Escuela de Ingenierías Industriales de la Universidad de Valladolid.* Valladolid : s.n., 2019.
11. **Electrical Deck.** What is HVDC Transmission? - Types, Advantages, Disadvantages & Applications. [En línea] 16 de agosto de 2021. [Último acceso: 3 de septiembre de 2021.] <https://www.electricaldeck.com/2021/08/what-is-hvdc-transmission.html>.
12. **Universidad Carlos III de Madrid.** Operación de sistemas HVDC. Particularidades. [En línea] 1 de abril de 2007. [Último acceso: 3 de marzo de 2021.] <https://electronica.uc3m.es/geste/HVDC1-UC3M.pdf>.
13. **Ingeniería Mecafenix.** ¿Que es un tiristor y como funciona? - Ingeniería Mecafenix. [En línea] 22 de octubre de 2018. [Último acceso: 30 de noviembre de 2021.] <https://www.ingmecafenix.com/electronica/que-es-un-tiristor-y-como-funciona/>.
14. **Universidad Técnica Federico Santa María.** Armonicos rectificador de 12 pulsos. [En línea] 15 de enero de 2018. [Último acceso: 20 de diciembre de 2021.] <http://www2.elo.utfsm.cl/~ipd411/archivos/apuntes/papers-Capitulo3.pdf>.
15. **Erick, Roch.** transistores.info | Transistores IGBT de Potencia. [En línea] 3 de enero de 2021. [Último acceso: 3 de diciembre de 2021.] <https://transistores.info/transistores-igbt-de-potencia/#:~:text=Estos%20dispositivos%20combinan%20simult%C3%A1neamente%20potencia,precisi%C3%B3n%20posible%20en%20los%20equipos..>
16. **Farnell.** AIKW50N65DH5XKSA1 Infineon, IGBT, 80 A, 1.66 V - Farnell. [En línea] [Último acceso: 30 de noviembre de 2021.] <https://es.farnell.com/infineon/aikw50n65dh5xksa1/igbt-aec-q101-650v-80a-270w-to/dp/2986337>.
17. **Mouser Electronics.** CM600DY-24T Mitsubishi Electric | Mouser España. [En línea] [Último acceso: 30 de noviembre de 2021.] <https://www.mouser.es/ProductDetail/Mitsubishi-Electric/CM600DY-24T?qs=GedFDFLaBXHf3XJ35Q9Umg%3D%3D>.
18. *The Fault Detection, Localization, and Tolerant Operation of Modular Multilevel Converters withan Insulated Gate Bipolar Transistor (IGBT) OpenCircuit Fault.* Li, Wei , y otros. 837, 4 de abril de 2018, Energies, Vol. 11.
19. **MathWorks.** MATLAB El lenguaje del cálculo técnico - MathWorks. [En línea] 2 de diciembre de 2016. [Último acceso: 20 de noviembre de 2021.] <https://es.mathworks.com/products/matlab.html>.
20. **MathWorks.** Descripción del producto MATLAB - MathWorks. [En línea] 30 de diciembre de 2015. [Último acceso: 20 de noviembre de 2021.]

https://es.mathworks.com/help/matlab/learn_matlab/product-description.html.

21. **MathWorks**. Simulación y diseño basado en modelos con Simulink. [En línea] 3 de noviembre de 2021. [Último acceso: 20 de noviembre de 2021.] <https://es.mathworks.com/products/simulink.html>.

22. **Wikipedia**. Tiempo real - Wikipedia, la enciclopedia libre. [En línea] 24 de febrero de 2020. [Último acceso: 18 de diciembre de 2021.] https://es.wikipedia.org/wiki/Tiempo_real.

23. **ConceptoDefinición**. ¿Qué es simulación? >> Su Definición y Significado 2021. [En línea] 30 de agosto de 2021. [Último acceso: 15 de diciembre de 2021.] <https://conceptodefinicion.de/simulacion/>.

24. **ATAYDE, Sergio** . Simulación en tiempo real para facilitar la | lceep.cl opal-rt. [En línea] 31 de octubre de 2020. [Último acceso: 24 de noviembre de 2021.] http://lceep.cl/wp-content/uploads/2020/10/Sergio_Atayde.pdf.

25. **National Instruments**. ¿Qué es el hardware-in-the-loop? [En línea] 14 de mayo de 2020. [Último acceso: 15 de diciembre de 2021.] <https://www.ni.com/es-es/innovations/white-papers/17/what-is-hardware-in-the-loop-.html>.

26. **MathWorks**. Simulación y pruebas en tiempo real: MATLAB y Simulink. [En línea] 23 de agosto de 2020. [Último acceso: 20 de noviembre de 2021.] <https://es.mathworks.com/solutions/real-time-simulation-and-testing.html>.

27. **xilinx**. What is an FPGA? Field Programmable Gate Array - Xilinx. [En línea] 14 de julio de 2017. [Último acceso: 1 de noviembre de 2021.] <https://www.xilinx.com/products/silicon-devices/fpga/what-is-an-fpga.html>.

