

Trabajo Fin de Grado

Grado en Ingeniería de Tecnologías Industriales

Puesta en marcha de una plataforma de simulación  
en tiempo real para el estudio de la operación de  
convertidores de potencia mediante técnicas  
Hardware-in-the-loop

Autor: Emilia Pérez Martín

Tutor: Sergio Vázquez Pérez

Eduardo Zafra Ratia

**Dpto. Ingeniería Electrónica**  
**Escuela Técnica Superior de Ingeniería**  
**Universidad de Sevilla**

Sevilla, 2021





Trabajo Fin de Grado  
Grado en Ingeniería de Tecnologías Industriales

**Puesta en marcha de una plataforma de simulación  
en tiempo real para el estudio de la operación de  
convertidores de potencia mediante técnicas  
Hardware-in-the-loop**

Autor:

Emilia Pérez Martín

Tutor:

Sergio Vázquez Pérez

Profesor titular

Eduardo Zafra Ratia

PDI Universidad de Sevilla

Dpto. de Ingeniería Electrónica  
Escuela Técnica Superior de Ingeniería  
Universidad de Sevilla

Sevilla, 2021



Proyecto Fin de Carrera: Puesta en marcha de una plataforma de simulación en tiempo real para el estudio de la operación de convertidores de potencia mediante técnicas Hardware-in-the-loop

Autor: Emilia Pérez Martín

Tutor: Sergio Vázquez Péro  
Eduardo Zafra Ratiaz

El tribunal nombrado para juzgar el Proyecto arriba indicado, compuesto por los siguientes miembros:

Presidente:

Vocales:

Secretario:

Acuerdan otorgarle la calificación de:

Sevilla, 2021

El Secretario del Tribunal

*A mis padres y a mi hermano*

*A Fernando*



# Agradecimientos

---

El trabajo de Fin de Grado supone el final de una de las etapas más importantes de mi vida como estudiante. Es lo que pone punto final a cuatro años de trabajo y esfuerzo, pero no podría haberlo conseguido sin todo el apoyo y el cariño que me dan los que me rodean.

En primer lugar, me gustaría agradecer a mi familia, mis padres y mi hermano. Ellos son mi apoyo incondicional, y me dan la fuerza y la confianza que necesito para superar los obstáculos del camino. Gracias por creer en mí.

Por otra parte, darle las gracias a mi novio, Fernando, que también ha sido un pilar fundamental en estos cuatro años de carrera. Como también es ingeniero, aparte de compartir las mismas experiencias, me ha ayudado mucho, dedicándome su tiempo y su cariño siempre que lo he necesitado.

También quería darle las gracias a mis amigos, que han compartido conmigo el peso de los estudios en primera línea de batalla, haciéndolo más ligero. Su amistad ha sido una pieza clave para haber disfrutado de la carrera tanto como lo he hecho yo.

Por último, me gustaría agradecer a todos los que han hecho que el desarrollo de este trabajo de investigación haya sido una experiencia tan bonita. En primer lugar, a mi tutor, Sergio Vázquez, por ofrecerme el proyecto, y por su cuidadoso y entregado tutelaje. Su atención, paciencia y dedicación han hecho posible que haya desarrollado un proyecto de investigación como este, que inicialmente se me antojaba muy complejo.

También quería agradecerle a Abraham Márquez, por su interés en el proyecto y por explicarme y ayudarme con mucha paciencia. Por último, me gustaría agradecerles también a Eduardo Zafra y a Pablo Gómez, porque me han dedicado su tiempo siempre que me he quedado atascada, explicándome cosas que no entendía o ayudándome a resolver los problemas que iban surgiendo. A todos vosotros, muchas gracias, porque habéis hecho que la experiencia de desarrollar un proyecto como este haya sido inmejorable, y he aprendido muchísimo de todos.

*Emilia Pérez Martín*

*Sevilla, 2021*



# Resumen

---

Este proyecto tiene como objetivo diseñar, construir y poner en marcha un sistema que permita probar los controladores de una planta PV en un simulador en tiempo real mediante técnicas hardware in the loop (HIL). Esto elimina la necesidad de disponer de la planta PV real, lo que reduce el coste de diseño del controlador y el “time to market”, tiempo desde que se diseña hasta que se comprueba su buen funcionamiento y sale al mercado. Este sistema está formado por una plataforma de simulación en tiempo real en la que se modela el sistema PV, y de una plataforma hardware de control externa de tipo System on a Chip (SoC), donde se programarán los algoritmos de control necesarios. La conexión de estas dos plataformas mediante un hardware adicional y la puesta en marcha del sistema completo permitirá la evaluación de los controladores de la planta PV mediante las técnicas HIL.



# Abstract

---

The aim of this project is to design, build and execute a system that allows testing the controllers of a PV plant in a real time simulator using hardware in the loop techniques (HIL). This eliminates the need to dispose of the real PV plant, which reduces the controller design cost and the “time to market”, defined as the time frame from the design of the controller until its proper operation is verified and it goes on the market. This system consists of a real-time simulation platform in which the PV system is modeled, and an external control hardware platform System on a Chip (SoC), where the control algorithms will be programmed. Connecting these two platforms using additional hardware and executing the complete system allows the evaluation of the PV plant controllers using HIL techniques.

<b>Agradecimientos</b>	<b>ix</b>
<b>Resumen</b>	<b>xi</b>
<b>Abstract</b>	<b>xiii</b>
<b>Índice</b>	<b>xiv</b>
<b>Índice de Tablas</b>	<b>xvi</b>
<b>Índice de Figuras</b>	<b>xviii</b>
<b>Notación</b>	<b>xxi</b>
<b>1 Introducción</b>	<b>1</b>
1.1 <i>Motivación y objetivos del proyecto</i>	1
1.2 <i>Sistema completo</i>	1
<b>2 Marco teórico</b>	<b>5</b>
2.1 <i>Ventajas de las técnicas de simulación HIL</i>	5
2.1.1 <i>Introducción</i>	5
2.1.2 <i>Esquema de la conexión completa</i>	6
2.2 <i>Ventajas de las técnicas de simulación HIL</i>	8
2.2.1 <i>Seguridad</i>	8
2.2.1 <i>Calidad y rango de las pruebas</i>	8
2.2.2 <i>Coste temporal</i>	8
2.2.3 <i>Coste económico</i>	9
2.2.4 <i>Factor humano</i>	9
2.3 <i>Aplicación de las técnicas Hardware in the Loop en otros campos</i>	10
2.3.1 <i>Industria automovilística</i>	10
2.3.2 <i>Ingeniería aeroespacial</i>	10
2.3.3 <i>Ingeniería marítima</i>	10
<b>3 OPAL 4510</b>	<b>11</b>
3.1 <i>Principio de funcionamiento</i>	11
3.2 <i>Características técnicas</i>	14
3.3 <i>Otras aplicaciones de la plataforma OPAL 4510</i>	15
3.3.1 <i>Rapid Control Prototyping (RCP)</i>	15
3.3.2 <i>Power hardware in the Loop (PHIL)</i>	15
3.4 <i>Implementación del modelo a simular con técnicas HIL</i>	16
3.4.1 <i>Modelo CPU</i>	16
3.4.2 <i>Modelo FPGA</i>	22
<b>4 Limitaciones de la opal 4510</b>	<b>25</b>
4.1 <i>Comparación entre el sistema real y el sistema implementado</i>	25
4.1.1 <i>Error de driver (error hardware)</i>	26
4.1.2 <i>Desdoble de señales de control</i>	27
4.1.3 <i>Función del enable</i>	29
4.2 <i>Circuito de precarga</i>	29
<b>5 Placas de adaptación</b>	<b>31</b>

5.1	<i>Canales analógicos y digitales de la OPAL</i>	31
5.2	<i>Entradas y salidas del hardware de control</i>	33
5.2.1	Entradas analógicas: conectores RJ12	34
5.2.2	Entradas y salidas digitales: fibra óptica	34
5.2.3	Comunicación con el ordenador host	35
5.3	<i>Placas analógicas</i>	36
5.3.1	Salidas analógicas de la OPAL	38
5.3.2	Entradas analógicas a la OPAL	39
5.4	<i>Placas digitales</i>	40
5.4.1	Entradas digitales a la OPAL	41
5.4.2	Salidas digitales de la OPAL	41
<b>6</b>	<b>Conexión del sistema y puesta en marcha</b>	<b>43</b>
6.1	<i>Problemas de ruido</i>	43
6.2	<i>Calibración de sensores: canales ADC y DC</i>	46
6.2.1	Calibración del sensor de tensión DC	47
6.2.2	Calibración del sensor de tensión AC	48
6.2.3	Calibración del sensor de corriente AC	52
<b>7</b>	<b>Modelos implementados</b>	<b>55</b>
7.1	<i>Inversor en modulación abierta</i>	55
7.2	<i>Inversor conectado a carga RL con control de corrientes</i>	56
7.2.1	Salidas digitales de la OPAL	57
7.2.2	Funcionamiento de la placa de adaptación	59
7.2.3	Funcionamiento de la etapa de adaptación del DSP	59
7.2.4	Lectura del canal analógico y representación por la interfaz de Matlab	59
7.2.5	Conclusiones	59
7.3	<i>Inversor conectado a una fuente DC y a la red con control de corrientes</i>	59
7.3.1	Representación de magnitudes eléctricas	60
7.3.2	Comprobación del funcionamiento del controlador	65
7.4	<i>Inversor conectado a red con circuito de precarga y control de la tensión del DC-link</i>	67
7.4.1	Comportamiento inusual del modelo. Mal funcionamiento en la FPGA	68
7.4.2	Representación de magnitudes eléctricas	71
7.4.3	Comprobación del funcionamiento del controlador	79
<b>8</b>	<b>Algoritmos MPPT</b>	<b>81</b>
8.1	<i>Sistema a implementar</i>	81
8.2	<i>Modelo de los paneles fotovoltaicos</i>	82
8.2.1	Caracterización de los paneles solares	82
8.2.2	Modelo de un panel fotovoltaico	83
8.2.3	Dimensionado del campo fotovoltaico	85
<b>9</b>	<b>Conclusiones y trabajos futuros</b>	<b>91</b>
9.1	<i>Medidas y calibración de sensores</i>	91
9.2	<i>Medidas y calibración de sensores</i>	92
9.2.1	Uso de bloques propios de la librería RT-LAB	92
9.2.2	Diseño de un hardware adicional	93
9.3	<i>Conclusiones</i>	94
	<b>Referencias</b>	<b>95</b>

# ÍNDICE DE TABLAS

---

Tabla 5-1. Canales analógicos y digitales	32
Tabla 5-2. Características Entradas Analógicas [6]	36
Tabla 5-3. Características Salidas Analógicas [6]	37
Tabla 5-4. Características Entradas Digitales [6]	41
Tabla 5-5. Características Salidas Digitales [6]	41
Tabla 6-1. Calibración del sensor de tensión DC	47
Tabla 6-2. Calibración de los sensores de tensión AC	51
Tabla 6-3. Calibración de los sensores de corriente AC	53
Tabla 7-1. Asignación canal de entrada-magnitud leída del DSP	57
Tabla 0-1. Asignación canal de entrada DSP-canal de salida analógico	58
Tabla 8-1. Parámetros que configuran las curvas características de un panel	84
Tabla 8-2. Configuración de las células solares de un panel	85
Tabla 8-3. Caracterización de un solo panel	86
Tabla 8-4. Caracterización del panel solar completo	87



# ÍNDICE DE FIGURAS

---

Figura 1-1. Montaje del sistema completo	2
Figura 2-1. Conexión del controlador al sistema real o al simulador en tiempo real [1]	5
Figura 2-2. Sistema HIL genérico	6
Figura 2-3. Sistema HIL en este proyecto	6
Figura 2-4. Esquema de conexión completo	7
Figura 2-5. Coste de los errores frente al tiempo que tardan en ser detectados [1]	8
Figura 2-6. Training simulator para una grúa de brazo articulado [1]	9
Figura 3-1. Simulador en tiempo real, OPAL 4510	11
Figura 3-2. Conexión de un simulador en tiempo real en una aplicación HIL [7]	12
Figura 3-3. Componentes de un simulador en tiempo real	13
Figura 3-4. Arquitectura de la plataforma de simulación en tiempo real OPAL 4510 [7]	14
Figura 3-5. Características técnicas de la OPAL [7]	14
Figura 3-6. Rapid Control Prototyping (RCP) [7]	15
Figura 3-7. Power Hardware-in-the-Loop (PHIL) [7]	16
Figura 0-8. Subsistemas del modelo CPU	17
Figura 0-9. Subsistema computacional o máster	17
Figura 0-10. Subsistema GUI típico	18
Figura 0-11. Bloque OpComm y representación de las medidas del circuito	19
Figura 0-12. Representación de las medidas en tiempo real (aproximación)	19
Figura 0-13. Usos típicos de las salidas del bloque GUI	20
Figura 0-14. Analog Output Mapping and Rescaling block Control Panel	21
Figura 0-15. Modelo FPGA	22
Figura 0-16. Configuración de los inputs	23
Figura 0-17. Configuración de los gates	24
Figura 0-1. Sistema de potencia real	25
Figura 0-2. Sistema HIL	26
Figura 4-3. Desdoble de las señales de control	27
Figura 4-4. Posible solución para poder manipular las entradas digitales	27
Figura 4-5. Configuración de los gates	28
Figura 4-6. Función de la señal “Enable”	29
Figura 0-7. Condensador (dc-link) conectado a red a través de un inversor	30
Figura 5-2. Distribución de canales entrada/salida de la OPAL [8]	31
Figura 5-2. Distribución detallada de canales analógicos y digitales de entrada y salida [8]	32
Figura 5-3. Conexión de las placas de adaptación a la OPAL	33
Figura 5-4. Entradas y salidas del hardware de control	34
Figura 5-5. Interfaz Matlab del DSP	35
Figura 5-6. Conexión de las placas de adaptación a la OPAL	36
Figura 5-7. Conexión de las placas de adaptación a la OPAL	37
Figura 5-8. Etapas por las que pasa la señal analógica	38

Figura 5-9. Placas digitales	40
Figura 0-3. Salida del ADC previa	44
Figura 0-4. Filtro LC elegido para la implementación (para cada entrada de alimentación)	44
Figura 6-3. Filtro LC implementado	455
Figura 6-4. Salida del ADC tras filtrar la alimentación de la placa de adaptación y el DSP	45
Figura 6-5. Ruido antes del filtro LC (señal amarilla) y ruido después del filtro LC (señal azul)	46
Figura 6-6. Recta de regresión del canal de tensión DC	48
Figura 0-7. Código implementado para aplicar la ganancia y el offset a la salida del ADC	48
Figura 0-8. Recta de mejor ajuste	49
Figura 0-9. Polinomio de mejor ajuste de grado 3	50
Figura 0-10. Resultados de las medidas manteniendo Vdc constante. Recta de regresión.	50
Figura 0-11. Recta de regresión de uno de los canales de corriente AC	52
Figura 0-12. Código implementado para aplicar la ganancia y el offset a la salida del ADC	53
Figura 7-5. Inversor implementado	55
Figura 7-2. Tensión a la salida del inversor modulada, medida con un osciloscopio	56
Figura 7-3. Inversor conectado a carga RL con control de corrientes	57
Figura 7-4. Conexión a través de cables RJ12 de la salida de las placas	58
Figura 7-5. Modelo eléctrico implementado	60
Figura 7-6. Tensiones de la red en la consola (subsistema GUI)	60
Figura 7-7. Tensiones de la red en el osciloscopio (salida OPAL)	61
Figura 7-8. Tensiones de la red en la interfaz Matlab del DSP	61
Figura 7-9. Tensión del dc-link en la consola (subsistema GUI)	62
Figura 7-10. Tensión del dc-link en el osciloscopio (salida OPAL)	62
Figura 7-11. Tensión del dc-link en la interfaz Matlab del DSP	63
Figura 7-12. Tensión del dc-link en la consola (subsistema GUI)	63
Figura 7-13. Tensión del dc-link en el osciloscopio (salida OPAL)	64
Figura 7-14. Tensión del dc-link en la interfaz Matlab del DSP	64
Figura 7-15. El lazo de control de corrientes	65
Figura 7-16. Interfaz Matlab del DSP antes de iniciar el control	66
Figura 7-17. Interfaz Matlab del DSP al controlar con $p=5000W$	66
Figura 7-18. Tensión del dc-link en la interfaz Matlab del DSP	67
Figura 7-19. Dc-link conectado a red	68
Figura 7-20. Comportamiento correcto de la corriente que sale del condensador	69
Figura 7-21. Comportamiento correcto de las corrientes inyectadas desde la red	70
Figura 7-22. Tensiones de la red en la consola (subsistema GUI)	71
Figura 7-23. Tensiones de la red en el osciloscopio (salida OPAL)	72
Figura 7-24. Tensiones de la red en la interfaz Matlab del DSP	72
Figura 7-25. Tensión del dc-link en la consola (subsistema GUI)	73
Figura 7-26. Tensión del dc-link en el osciloscopio (salida OPAL)	73
Figura 7-27. Tensión del dc-link en la interfaz Matlab del DSP	74
Figura 7-28. Corrientes en la consola (subsistema GUI)	75
Figura 7-29. Corrientes en el osciloscopio (salida OPAL)	75

Figura 7-30. Corrientes en la interfaz Matlab del DSP	76
Figura 7-31. Potencia en el DC-link	77
Figura 7-32. Potencia en la parte AC	78
Figura 7-33. Potencia en el DC-link	79
Figura 7-34. Salto en la potencia AC	80
Figura 7-35. Salto en las corrientes	80
Figura 8-6. Sistema eléctrico a implementar en la FPGA	81
Figura 8-2. Sistema eléctrico a implementar en la FPGA	82
Figura 8-3. Ejemplo de curvas características de un PV array	83
Figura 8-4. Circuito equivalente a un panel fotovoltaico	84
Figura 8-5. Circuito simulink del panel fotovoltaico	84
Figura 8-6. S-Function que contiene el modelado matemático del panel	85
Figura 8-7. Curvas características de un solo panel PV	86
Figura 8-8. Ecuaciones del campo solar	87
Figura 8-9. Modelo simulink del campo solar	88
Figura 8-10. Curvas características del campo solar	88
Figura 9-7. Sistema eléctrico a implementar en la FPGA	91
Figura 9-2. Tiempos muertos de las señales de disparo	92
Figura 9-3. Convertidores de potencia propios del software RT-LAB	93
Figura 9-4. Funciones lógicas que se deben implementar en la placa driver	93

HIL	Hardware In the Loop
FPGA	Field Programmable Gate Array
CPU	Central Processing Unit
I/O	Inputs and Outputs
DSP	Digital Signal Processing



# 1 INTRODUCCIÓN

---

La investigación sobre sistemas de conversión de potencia de alta eficiencia para integración en la red eléctrica de fuentes de energía renovables y sistemas de almacenamiento de energía es una de las líneas de investigación del departamento de Ingeniería Electrónica de la Universidad de Sevilla. El proyecto propuesto se enmarca en esta línea.

## 1.1 Motivación y objetivos del proyecto

Este trabajo de fin de grado tiene como objetivo diseñar, construir y poner en marcha un sistema que permita probar los controladores de una planta PV en un simulador en tiempo real mediante técnicas hardware in the loop (HIL).

Un sistema de simulación HIL es un tipo de simulación en tiempo real. En las técnicas HIL, el sistema eléctrico a controlar es sustituido por una plataforma de simulación en tiempo real. Las señales generadas por esta plataforma se adaptan de manera que al controlador le llegan los mismos estímulos que si estuviera conectado a un sistema de potencia real. De esta manera, se puede probar el diseño del controlador en tiempo real. Esto supone un gran avance, ya que elimina la necesidad de disponer de la planta PV real, lo que reduce el coste de diseño del controlador y el "time to market", tiempo desde que se diseña hasta que se comprueba su buen funcionamiento y sale al mercado.

En concreto, el montaje realizado en este proyecto está formado por una plataforma de simulación en tiempo real en la que se modela el sistema PV, y de una plataforma hardware de control externa DSP (Digital Signal Processing), donde se programarán los algoritmos de control necesarios. La conexión de estas dos plataformas mediante un hardware adicional y la puesta en marcha del sistema completo permitirá la evaluación de los controladores de la planta PV mediante las técnicas HIL.

En líneas generales, los objetivos de este proyecto son:

- Poner en marcha una plataforma de simulación en tiempo real para realizar pruebas mediante técnicas Hardware in the Loop (HIL). En concreto, la plataforma OPAL-RT OP4510
- Implementar en dicha plataforma modelos de plantas de sistemas fotovoltaicos (PV) controlados mediante convertidores DC/DC y DC/AC
- Operar estos convertidores DC/DC mediante técnicas HIL, usando hardware de control basado en integrados de tipo System on a Chip (SoC)
- Evaluar diversos algoritmos de seguimiento del punto de máxima potencia (MPPT) a través de la plataforma de simulación en tiempo real

## 1.2 Sistema completo

Durante el desarrollo del proyecto, se utilizarán varios elementos hardware para diseñar y construir un sistema completo y funcional que permita el estudio de algoritmos MPPT en sistemas PV mediante plataformas de simulación en tiempo real.

La figura 1-1 es una foto del sistema completo, tomada en el laboratorio.

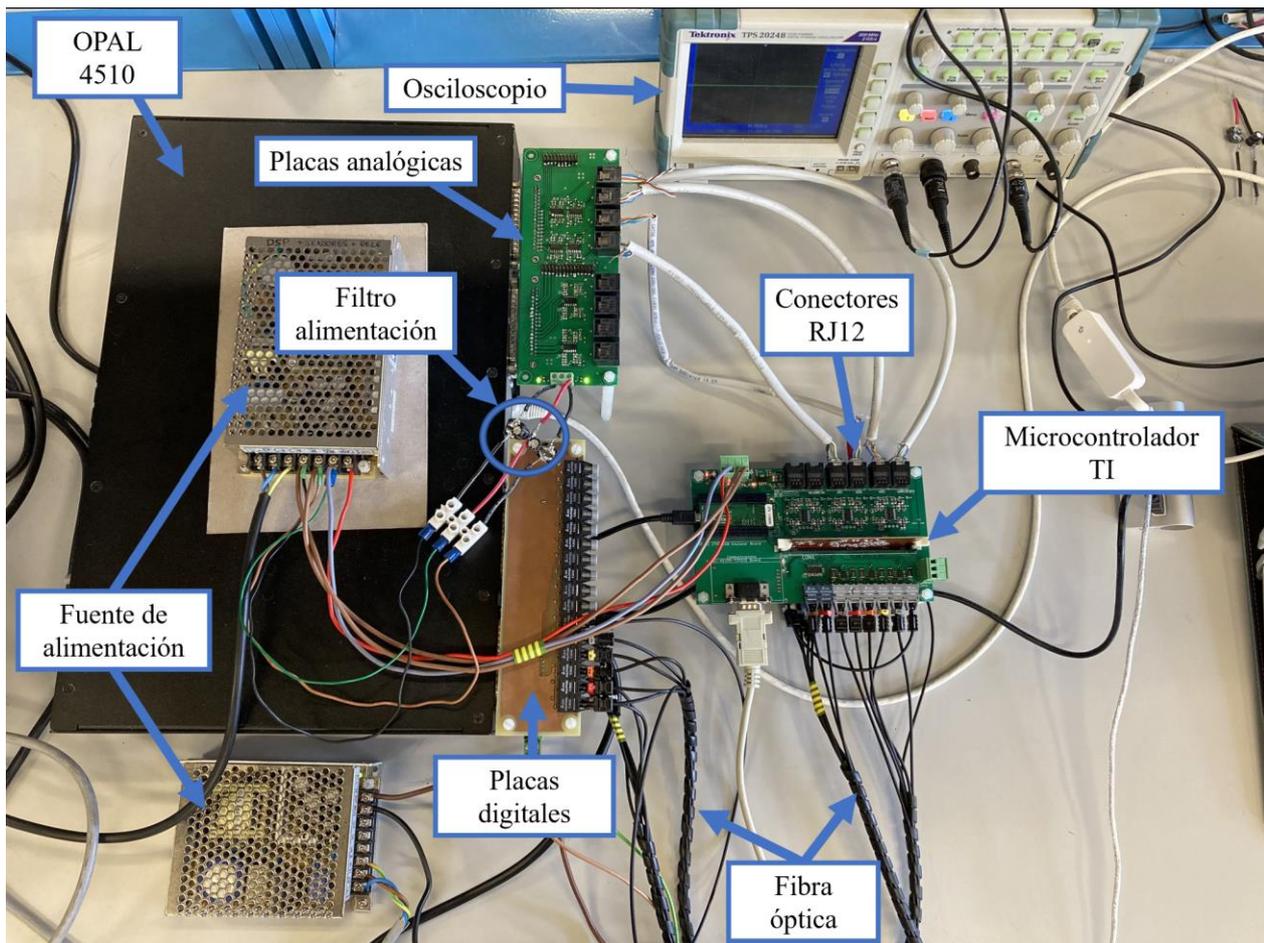


Figura 1-1. Montaje del sistema completo

En primer lugar, se dispone de la plataforma de simulación en tiempo real OPAL-RT OP4510, adquirida por el departamento, la cual se configurará y programará. Además, se desarrollarán e implementarán en dicha plataforma los modelos de plantas fotovoltaicas y convertidores de potencia, a los que posteriormente se aplicarán las distintas estrategias de control en tiempo real. Este simulador es el que permite prescindir de la planta PV real.

Por otra parte, se dispone del hardware integrado correspondiente al controlador, de tipo DSP. Para la conexión de este hardware de control externo a la plataforma de simulación en tiempo real se hará uso de varias placas de conexión, que adaptan las salidas y entradas del controlador y del simulador entre sí, haciéndolas compatibles. Dichas placas fueron diseñadas y desarrolladas por el departamento en años anteriores, por lo que en este proyecto se procederá a su conexión y puesta en marcha.

El hardware de control externo se programará implementando los distintos controladores diseñados para la planta fotovoltaica. Se hará uso de algoritmos MPPT, que hagan un seguimiento del punto de máxima potencia.

Por último, una vez que se compruebe el correcto funcionamiento del conjunto, se evaluarán las estrategias de control implementadas en el hardware de control externo mediante técnicas HIL en la plataforma de simulación en tiempo real.

Por tanto, la secuencia de tareas a realizar es la siguiente:

- 1) Configuración y puesta en marcha de la plataforma de simulación en tiempo real
- 2) Desarrollo de los modelos de plantas PV y convertidores de potencia para la plataforma
- 3) Conexión del hardware de control externo a la plataforma de simulación en tiempo real
- 4) Programación de los algoritmos de control y MPPT en el hardware de control externo

- 5) Evaluación de los diferentes algoritmos mediante técnicas HIL en la plataforma de simulación en tiempo real



## 2 MARCO TEÓRICO

En esta sección se procede a explicar el marco teórico en el que se desarrolla el proyecto: las técnicas Hardware in the Loop. Se explicará el principio de funcionamiento de esta técnica de simulación en tiempo real, y el papel que tiene en la ingeniería actual.

### 2.1 Ventajas de las técnicas de simulación HIL

#### 2.1.1 Introducción

Las técnicas “Hardware in the loop” constituyen un método para probar el funcionamiento de un controlador en fase de desarrollo sobre un sistema en tiempo real prescindiendo del propio sistema. Esto es así porque el sistema sobre el que se ejecutaría el control es sustituido por una plataforma de simulación en tiempo real, como se representa en la figura 2-1. Esta plataforma genera unas entradas y salidas analógicas y digitales que emulan las entradas y salidas del sistema real, de manera que el controlador se ejecuta como si actuara sobre el sistema real a todos los efectos.

Esto supone un gran avance, ya que permite testear el algoritmo de control diseñado sin tener en cuenta las restricciones de seguridad, coste o disponibilidad que tendría probar el control sobre el producto construido. De esta manera, probando exhaustivamente el controlador en un entorno virtual antes de proceder a realizar pruebas con el dispositivo físico, se puede asegurar la fiabilidad del sistema de control sin aumentar el “time-to-market” o los costes asociados a las pruebas sobre el sistema real.

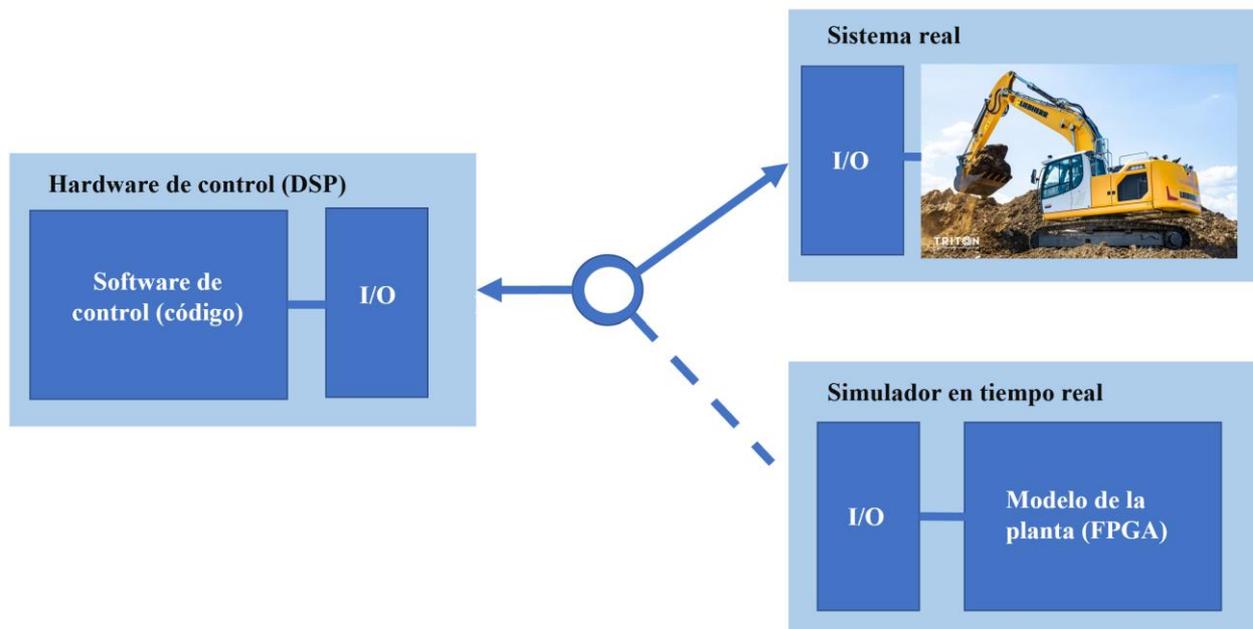


Figura 2-1. Conexión del controlador al sistema real o al simulador en tiempo real [1]

## 2.1.2 Esquema de la conexión completa

Un sistema HIL está constituido por 3 componentes fundamentales: una unidad de control electrónico, un simulador en tiempo real, y una interfaz hacia el usuario.



Figura 2-2. Sistema HIL genérico

El simulador en tiempo real es el elemento fundamental de un sistema HIL. En este proyecto, el simulador el tiempo real es la plataforma OPAL 4510, en cuya FPGA se implementa el modelo eléctrico que se va a ejecutar en tiempo real y que se pretende controlar. Esta plataforma recibe las señales digitales procedentes del control y genera señales analógicas que emulan las distintas medidas que se tomarían sobre el sistema eléctrico real.

El otro elemento fundamental de un sistema HIL es el controlador. En este proyecto es una unidad del tipo DSP. Se trata del microcontrolador 28335 de Texas Instruments, conectado a una tarjeta diseñada por el departamento. Aunque más adelante se detallará el funcionamiento de esta placa, en líneas generales, recibe las señales analógicas generadas por la plataforma de simulación en tiempo real y genera las señales digitales que controlan el modelo, según un algoritmo de control en bucle cerrado.

Por último, la interfaz con el usuario permite ver en tiempo real las medidas que se toman sobre el modelo eléctrico, y cambiar parámetros del sistema modelado también en tiempo real. En este proyecto en concreto se trabaja con dos interfaces distintas. En primer lugar, la interfaz propia de un simulador en tiempo real, la consola, que en este caso utiliza Simulink para representar las magnitudes que se generan y que permite cambiar parámetros del modelo eléctrico. Por otro lado, una interfaz de Matlab que se conecta con el DSP a través de puerto serie, que representa las medidas que lee el DSP y permite cambiar los parámetros de control. Por tanto, al tener dos interfaces distintas y representarse las dos en el mismo ordenador host, el esquema de conexión aplicado al montaje de este proyecto concreto será:



Figura 2-3. Sistema HIL en este proyecto

Por último, en los sistemas HIL se suele utilizar un hardware adicional que compatibiliza las entradas y salidas de la plataforma de simulación y del controlador entre sí. En este proyecto se han utilizado unas placas diseñadas por un alumno previamente a este proyecto [6]. Se trabaja con unas placas analógicas, que adaptan las señales generadas por la OPAL a un rango adecuado para el controlador, y de igual manera, existen unas

placas digitales que adaptan las señales de control generadas por el controlador a un rango adecuado para la plataforma. El funcionamiento de estas placas se detallará más adelante.

Por tanto, el esquema de conexión completo del sistema de simulación HIL implementado en este proyecto se puede resumir en los siguientes puntos (ver figura 2-4):

- Plataforma de simulación en tiempo real (OPAL 4510): Contiene el modelo del sistema eléctrico a simular.
- Placas de adaptación analógicas: Adaptan las salidas analógicas de la OPAL (medidas eléctricas) al rango de entrada analógica del controlador.
- Hardware de control (28335 TI): Contiene el algoritmo de control.
- Placas de adaptación digitales: Adaptan las salidas digitales del controlador a los requisitos de entrada digital de la OPAL.

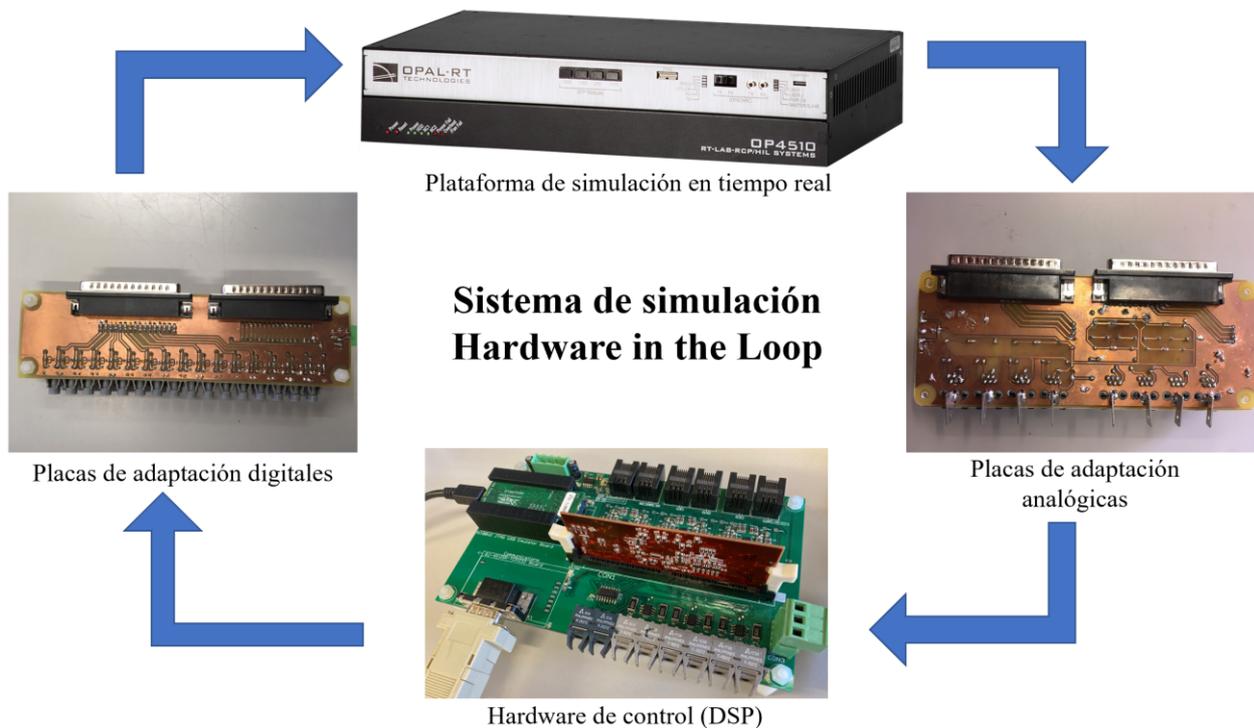


Figura 2-4. Esquema de conexión completo

## 2.2 Ventajas de las técnicas de simulación HIL

Los beneficios o ventajas que presentan las técnicas HIL son muchas. En este apartado se listan las más llamativas [1].

### 2.2.1 Seguridad

En las técnicas HIL se prueban los distintos algoritmos de control sobre un sistema simulado, no un sistema real. Por eso se pueden ejecutar sin problema pruebas de control que, en una situación real, podrían dañar el sistema o incluso podrían comprometer la seguridad de personas.

Por eso, los sistemas HIL son extremadamente útiles para simular sistemas como convertidores de potencia o sistemas de grúas pesadas, en los que garantizar la seguridad del personal es un requisito indispensable. Probar distintos algoritmos de control sobre este tipo de sistema es potencialmente peligroso y requiere medidas de seguridad costosas.

### 2.2.1 Calidad y rango de las pruebas

Por otra parte, las técnicas HIL permiten ampliar el rango en el que se prueba el sistema. Sin aplicar técnicas HIL, el sistema de control embebido se probaría sobre un sistema real, pero el sistema real impone limitaciones que reducen el rango en el que se puede hacer pruebas, ya sea por motivos de seguridad o por simple limitación física.

Por ejemplo, al probar un controlador sobre un motor real, no se podría testear el comportamiento real del motor ante un fallo crítico o para un valor de parámetros fuera de rango, porque ejecutar esas situaciones potencialmente peligrosas puede comprometer la integridad tanto del sistema físico como del técnico realizando las pruebas.

Con técnicas HIL, sin embargo, se pueden simular todas estas situaciones sin riesgo a que haya fallo de seguridad, ya que, aunque el controlador recibe las magnitudes correspondientes a un comportamiento anormal del sistema y actúa en consecuencia, el simulador en tiempo real no es un sistema peligroso con un comportamiento impredecible.

### 2.2.2 Coste temporal

El coste de los errores aumenta con la fase de desarrollo en la que se encuentre el proyecto, es decir, con el tiempo que tardan en descubrirse, como se representa en la figura 2-5. Por eso, localizar errores durante la puesta en marcha supone uno de los atrasos temporales y sobrecostos más importantes a los que se enfrentan los proyectos de ingeniería.

El procedimiento habitual es desarrollar máquinas y sistemas de control en paralelo. Esto hace que muchos errores en el sistema de control sean localizados solamente en la puesta en marcha del sistema. Las simulaciones HIL son una herramienta muy potente en ese aspecto, porque ayudan a identificar errores en las primeras etapas del sistema de control, reduciendo significativamente el tiempo de desarrollo del producto o "time to market" (tiempo desde que comienza el proceso de diseño hasta que el producto sale al mercado).

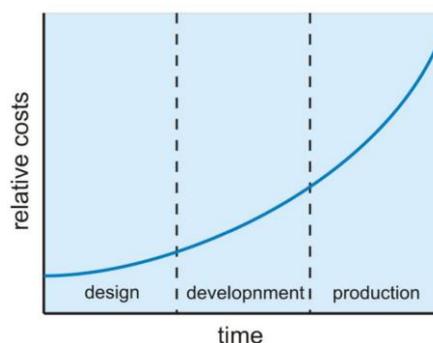


Figura 2-5. Coste de los errores frente al tiempo que tardan en ser detectados [1]

### 2.2.3 Coste económico

De manera evidente, las simulaciones HIL reducen los costes económicos del proyecto, ya que eliminan la necesidad de hacer pruebas sobre sistemas reales. El sistema real puede que no sea fácilmente adquirible o accesible, y además puede sufrir daños durante las pruebas.

Otra ventaja en cuanto a coste económico es el número de horas trabajadas en función de la fase o etapa en la que se encuentre el proyecto, como se comentó anteriormente. Las horas trabajadas en la fase de puesta en marcha son, por lo general, más caras que las horas trabajadas en las fases de desarrollo, especialmente si se trata de una puesta en marcha en una localización remota. Las técnicas HIL ayudan a pulir el control en las primeras etapas de desarrollo, reduciendo el número de horas de la fase de implementación.

### 2.2.4 Factor humano

Cuando el sistema HIL va acompañado de un soporte de visualización 3D que permita representar la planta que se está simulando, el sistema se conoce como “training simulator”, que es lo que se representa en la figura 2-6. Estos simuladores permiten al operador formarse en el manejo de la planta tanto en situaciones de funcionamiento normal como situaciones de riesgo o malfuncionamiento, desde un entorno seguro.



Figura 2-6. Training simulator para una grúa de brazo articulado [1]

## 2.3 Aplicación de las técnicas Hardware in the Loop en otros campos

En este proyecto, se utilizan las técnicas HIL para simular distintos sistemas de potencia y su conexión a la red eléctrica. Sin embargo, las altas prestaciones y beneficios que presentan hacen que estas técnicas de simulación estén teniendo mucho éxito en distintas áreas de la ingeniería y estén cada vez más extendidas. En concreto, el tipo de proyecto en el que las técnicas HIL tienen más éxito es aquel en el que se necesita una simulación en tiempo real verídica para cumplir con requisitos de prestaciones, costes y seguridad en el desarrollo de un producto.

### 2.3.1 Industria automovilística

En la actualidad de la industria automovilística, se tiende a controlar los vehículos con ordenadores cada vez más avanzados, llamados ECU (Electronic Control Unit). Probar estos controladores sobre vehículos reales incrementa los costes de diseño, los riesgos de seguridad, y a veces estos ensayos pueden ser destructivos para la mercancía. Para acelerar el proceso de diseño, se prueba el control utilizando softwares que simulan distintos componentes físicos como motores, ruedas...etc. Estos modelos se conocen como MIL (model in the loop) o modelos software.

Sin embargo, dado los altos requerimientos de seguridad para este tipo de producto, una simulación software no es suficiente. Los sistemas HIL representan una alternativa cada vez más extendida para probar el control en tiempo real. Un ECU se comunica con el vehículo a través de entradas y salidas analógicas y digitales. El papel del simulador en tiempo real es generar estas entradas y salidas, y reaccionar consecuentemente a las señales de control del ECU. De esta forma, se puede probar el funcionamiento del ECU de manera física y en tiempo real.

Otra ventaja de esta técnica de simulación es que permite que los diseñadores ECU trabajen en paralelo a equipo mecánico que fabrica el vehículo real. [2], [3]

### 2.3.2 Ingeniería aeroespacial

En los sistemas aeroespaciales existen maniobras de control que se suelen automatizar, como la operación Rendezvous and Docking (RvD). Este tipo de control crítico debe ser analizado cuidadosamente, simulado y verificado en detalle antes de ejecutarse en una misión espacial real. Las técnicas HIL proporcionan una gran herramienta para aproximar el comportamiento real del sistema en el espacio y probar físicamente el controlador.

En concreto, el artículo referenciado [4] trata sobre un satélite autónomo cuya misión es reparar otros satélites que ya están en órbita, por lo que la maniobra RvD debe ser automatizada.

### 2.3.3 Ingeniería marítima

En sistemas offshore, el sistema de control y la estructura mecánica se diseñan a la vez, lo que es propicio a que, una vez integrados ambos sistemas, aparezcan muchos fallos que pueden conllevar daños personales y materiales.

De nuevo, las técnicas de simulación HIL se sitúan como pioneras en esta área de la ingeniería. Una prueba del éxito de esta metodología es la empresa Det Norske Veritas (DNV), que actualmente utiliza técnicas HIL para probar todo tipo de sistemas de control offshore: DP (Dynamic Positioning), Power Management System, Steering, Propulsion and Thruster... etc. [5]

# 3 OPAL 4510

Uno de los elementos fundamentales de un sistema de simulación Hardware In the Loop es el simulador en tiempo real. En este proyecto se va a trabajar con la plataforma OPAL 4510. Esta sección está dedicada a explicar el concepto de simulador en tiempo real, qué elementos lo forman internamente y qué lo distingue de otros simuladores que no sean en tiempo real.

Tras explicar el principio de funcionamiento, se listarán brevemente las características técnicas de la plataforma concreta que se ha utilizado y se expondrán otras aplicaciones que se pueden desarrollar sobre la plataforma. Por último, se explicará detalladamente el uso que se le da a la plataforma en este proyecto: es decir, se explicará cómo implementar un modelo eléctrico en la FPGA de la OPAL 4510.

La figura 3-1 es una fotografía de la plataforma de simulación en tiempo real, la OPAL 4510.



Figura 3-1. Simulador en tiempo real, OPAL 4510

## 3.1 Principio de funcionamiento

Cuando se habla de “simulación” en el ámbito académico, lo habitual es que haga referencia a una simulación software, que es un tipo de simulación que permite un primer análisis del comportamiento del sistema utilizando exclusivamente los recursos existentes en un ordenador. Una simulación software se caracteriza porque el modelo se ejecuta haciendo uso de los recursos computacionales de los núcleos de la CPU, con las restricciones temporales que esto conlleva.

Estos tipos de simuladores calculan, al intervalo de tiempo que les permiten los recursos de la CPU, la respuesta del sistema, y la representan. Por ejemplo, es posible que para representar el comportamiento de un sistema durante un intervalo de 3 segundos, el simulador necesite 5 minutos para calcular la simulación.

Otra característica de este tipo de simulación que hace que no sea en tiempo real es que, al inicio de cada periodo, lee los datos necesarios, calcula la acción de control, y la aplica en ese mismo periodo. Esto es un comportamiento ideal que no ocurre en la realidad, porque no da tiempo a hacer toda la secuencia de operaciones. Lo que ocurre en un control real es que, al inicio de cada periodo, se leen los datos y se aplica la acción de control que se calculó con los datos tomados en el periodo anterior, mientras se calcula la acción de control en función de los datos que se acaban de tomar, y que se aplicará en el siguiente periodo.

Por todos estos motivos, las simulaciones conocidas como simulaciones software no se pueden clasificar como simulaciones en tiempo real. Aun así, estas simulaciones son muy útiles para obtener una primera aproximación del comportamiento del sistema y comprender qué está ocurriendo a grandes rasgos.

Sin embargo, en el ámbito comercial, a la hora de desarrollar nuevos productos y sus respectivos controladores, se necesita comprobar el buen funcionamiento del controlador de una manera más exigente y realista. Por ejemplo, siguiendo la tendencia en la industria automovilística a automatizar la conducción, es necesario que el controlador del vehículo se haya probado con algún tipo de simulador que emule el comportamiento del sistema real más fielmente que la simulación software.

A medio camino entre la simulación software y testear el control sobre el propio sistema se encuentra la simulación en tiempo real, en concreto, las técnicas HIL sobre las que se desarrollan este trabajo. En la figura 3-2 se representa un esquema de un sistema HIL.

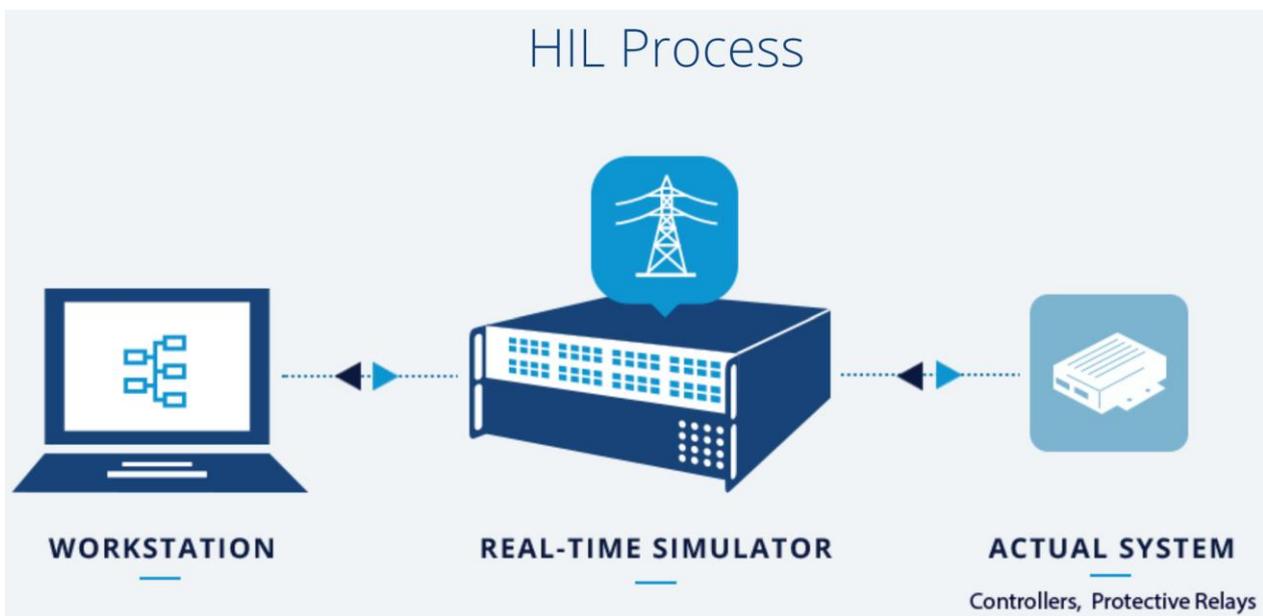


Figura 3-2. Conexión de un simulador en tiempo real en una aplicación HIL [7]

Lo que permite que una simulación se ejecute en tiempo real y que, por tanto, se ajuste mucho más a la realidad, es el soporte sobre el que se implementa el modelo a simular. En los simuladores en tiempo real, el modelo se implementa en la FPGA y no en la CPU.

Ejecutar un modelo codificado en una FPGA en vez de utilizar los recursos de la CPU supone un salto cualitativo importante. El time step o tiempo de ejecución se reduce drásticamente, pasando de microsegundos a nanosegundos, debido a que el circuito está implementado de manera hardware, y no tiene que esperar a entrar en tiempo de ejecución de la CPU.

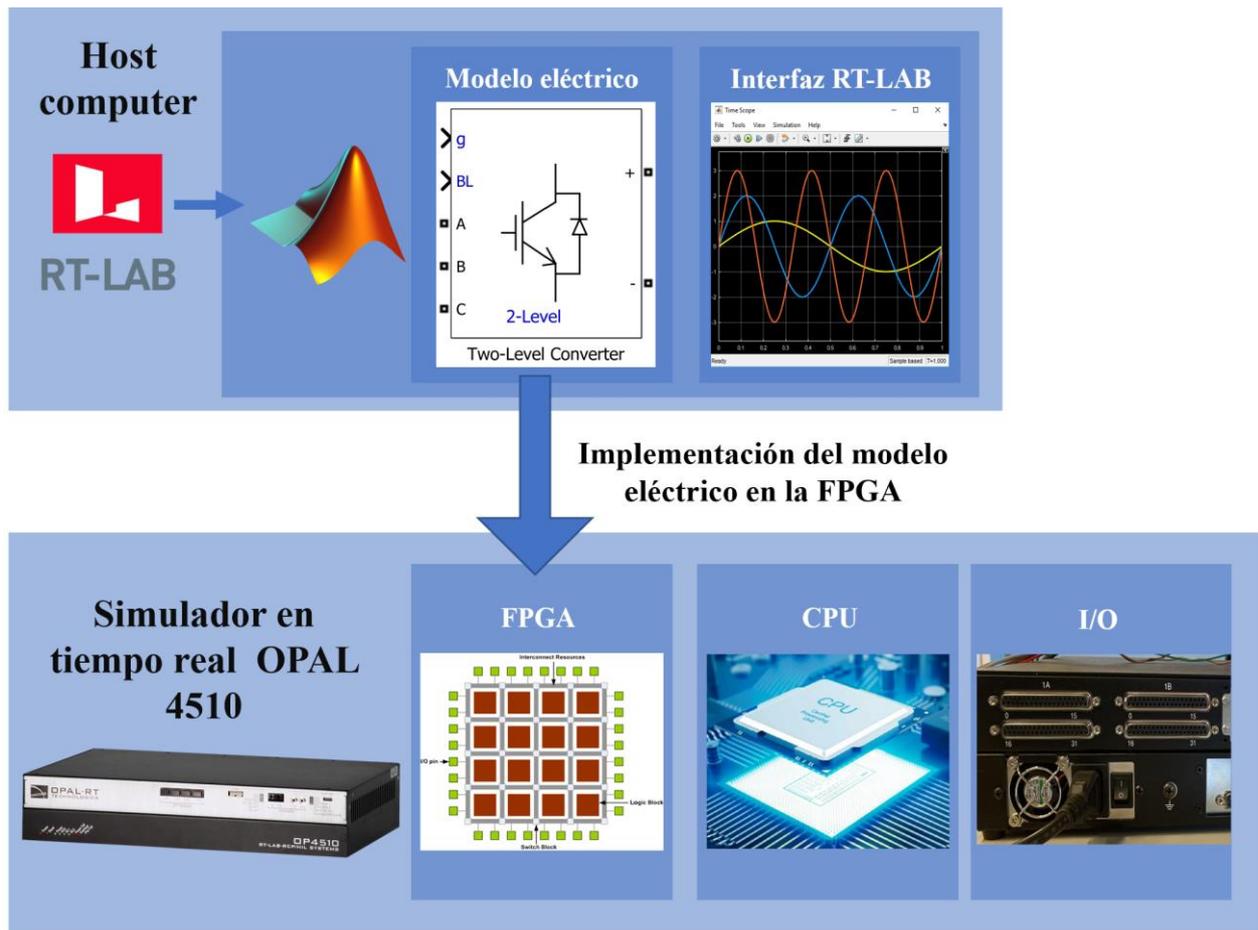


Figura 3-3. Componentes de un simulador en tiempo real

La dinámica de un sistema eléctrico como el que se desea implementar está gobernada por una serie de ecuaciones de segundo grado o de orden superior, que serían difíciles de programar en una FPGA usando lenguaje de descripción hardware (HDL, Hardware Description Language). De ahí la ventaja que presenta utilizar una plataforma como OPAL. Esta plataforma permite al usuario diseñar el modelo eléctrico a través de una interfaz más sencilla, como Simulink, y después implementa este modelo en la FPGA automáticamente, de manera que la simulación se ejecuta en la FPGA, con todas las bondades que eso conlleva: time step de simulación de nano segundos, no ocupa recursos de la CPU... etc.

Por tanto, se puede concluir que un simulador en tiempo real se compone principalmente de tres elementos:

1. FPGA
2. CPU
3. Canales de entrada y salida analógicos y digitales

En la figura 3-3 se representa la relación que existe entre el ordenador host donde se diseña el modelo y la plataforma de simulación en tiempo real. También se representan los componentes internos del simulador que se acaban de listar.

### 3.2 Características técnicas

En este apartado se especifican las características técnicas de los elementos que se han mencionado en el apartado 3.1.

En la figura 3-4 se representa la arquitectura interna de la OPAL. Se pueden destacar el multicore CPU y la KINTEX 7 FPGA. También se representan todos los canales de entrada y salida analógicos y digitales que tiene la OPAL. Estos canales se explicarán con más detalle en el apartado 5.1.

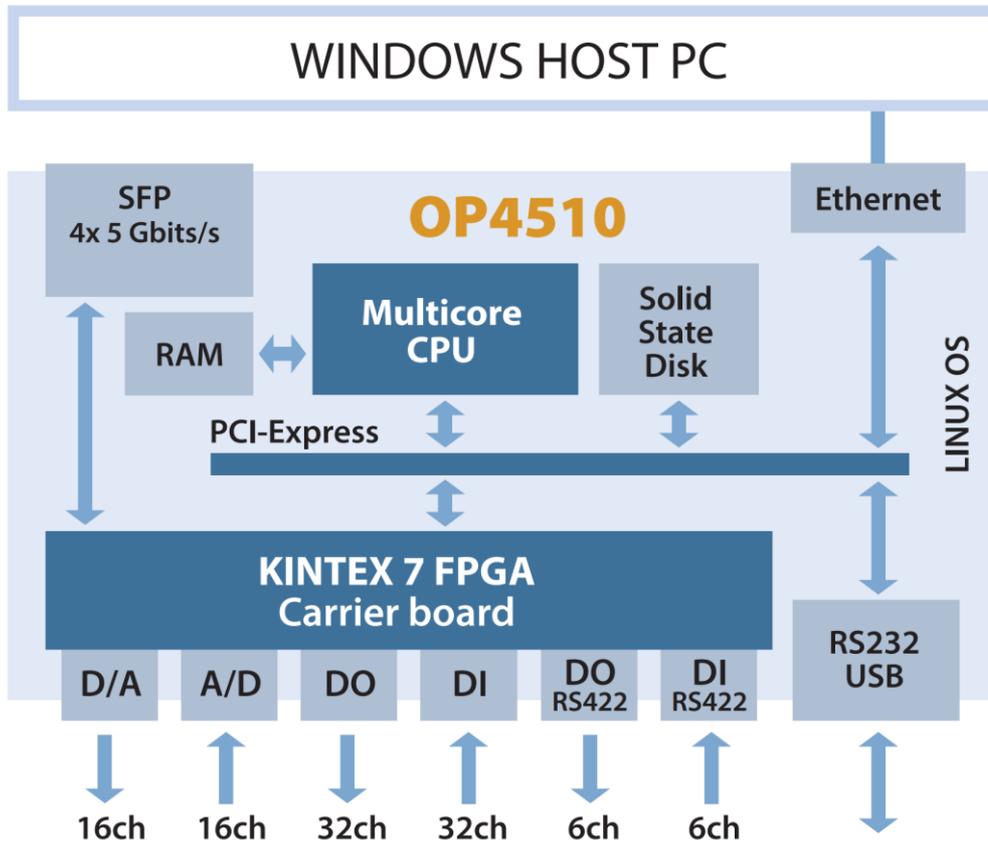


Figura 3-4. Arquitectura de la plataforma de simulación en tiempo real OPAL 4510 [7]

También se adjunta la figura 3-5, en la que se recoge el nombre comercial de cada uno de los elementos que componen la OPAL.

<b>Processors</b>	CPU	Intel Xeon E3 4-core 3.5 GHz
<b>FPGA</b>	Kintex™-7 K325T standard (K410T optional)	
<b>Performance</b>	CPU	Min. time step model execution of 7 microseconds
	FPGA	Timer resolution of 10ns and minimum time steps of 250 ns
<b>I/O cassettes (max. 4 per system)</b>	Analog	16 channels (max. of 64 per system)
	Digital	32 channels (max. of 128 per system)
<b>Optional I/O boards</b>	RS422	12 channels for I/O, encoder, decoder or other applications
	Optical fiber	6 optical links, 50 Mbps
<b>High-speed optical interface</b>	4 SFP sockets, up to 5Gbps	
<b>Connectivity</b>	Standard	Ethernet
		RS-232
		USB
		JTAG
		VGA
		Keyboard and mouse
<b>Dimensions</b>	W x D x H	2U, 17" x 10.8" x 3.5"

Figura 3-5. Características técnicas de la OPAL [7]

### 3.3 Otras aplicaciones de la plataforma OPAL 4510

En este proyecto, la plataforma OPAL 4510 se utiliza para implementar un sistema de simulación HIL. Sin embargo, no es el único tipo de sistema de simulación que soporta esta plataforma, y se le pueden dar otros usos, para desarrollar otras aplicaciones.

#### 3.3.1 Rapid Control Prototyping (RCP)

El prototipado de control rápido es otro tipo de aplicación que se puede implementar utilizando la OPAL 4510. En este caso, el montaje es el contrario a un sistema HIL: el controlador se implementa en la CPU de la OPAL, y esta se conecta al sistema de potencia real que se pretende controlar. Este esquema se representa en la figura 3-6.

Diseñar el controlador haciendo uso del software de la OPAL permite realizar un diseño del control más rápido, y una detección temprana de errores, reduciendo el coste de atrasos temporales. Este tipo de diseño es especialmente útil para tareas de investigación.

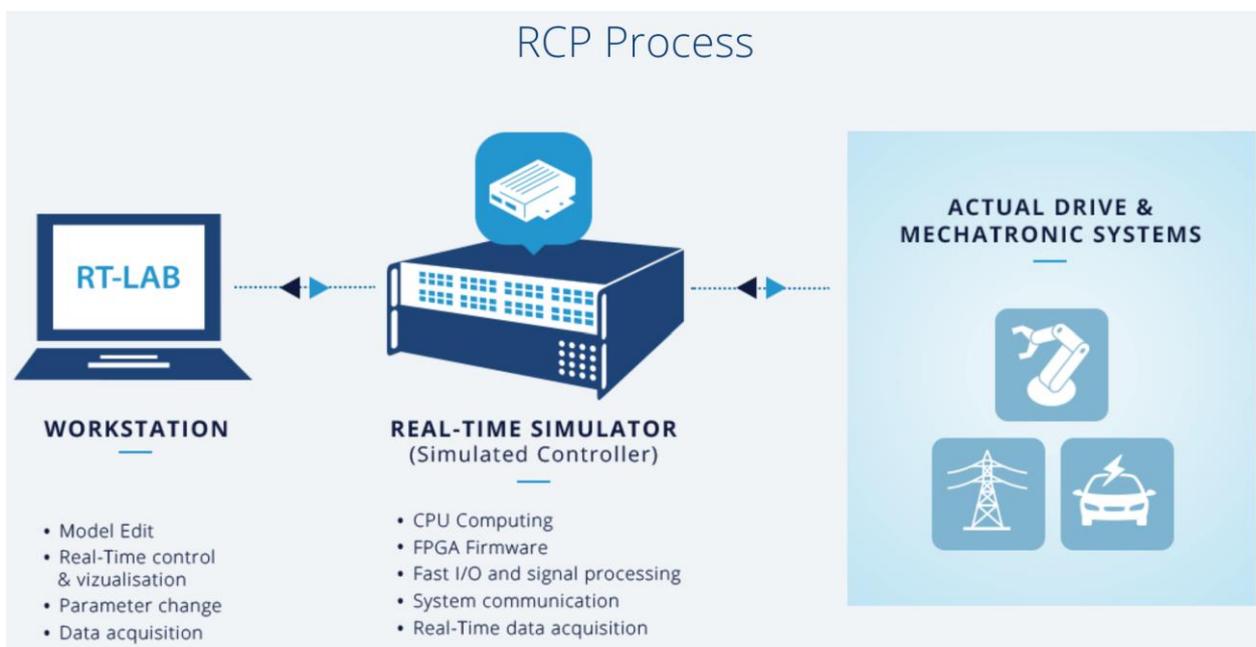


Figura 3-6. Rapid Control Prototyping (RCP) [7]

#### 3.3.2 Power hardware in the Loop (PHIL)

La simulación Power Hardware-in-the-Loop (PHIL) es una extensión natural de las técnicas HIL. En este tipo de simulación, la plataforma de simulación en tiempo real no intercambia exclusivamente señales de baja tensión y baja corriente, sino que es capaz de proporcionar la potencia necesaria para probar el Device Under Test (DUT).

Para ello, se colocan amplificadores de potencia entre el simulador en tiempo real de baja potencia y el DUT de alta potencia, y el sistema se realimenta con las señales necesarias para cerrar el lazo de control. Los amplificadores de potencia se seleccionan en función de la aplicación y las necesidades del DUT. Este esquema de conexión se sintetiza en la figura 3-7.

Por tanto, las simulaciones PHIL mantienen todas las ventajas que presentaban las simulaciones HIL en bucle cerrado, pero también permiten la simulación de flujos de alta potencia entre distintos DUTs.

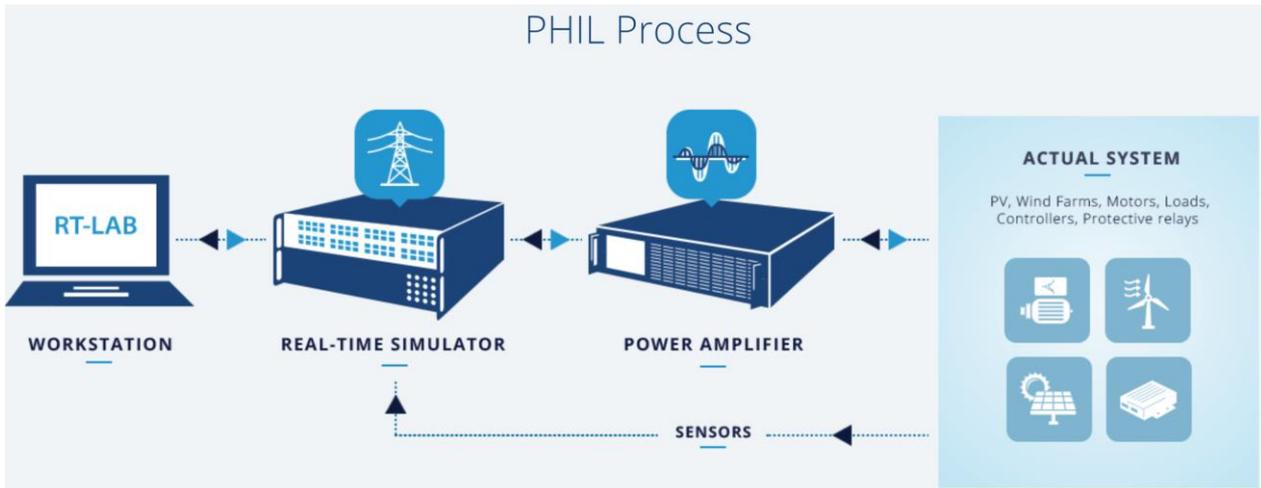


Figura 3-7. Power Hardware-in-the-Loop (PHIL) [7]

### 3.4 Implementación del modelo a simular con técnicas HIL

Este apartado se centra en explicar cómo se implementa un modelo eléctrico en la FPGA para ejecutar una simulación HIL. Se necesitan dos archivos distintos:

- Modelo Simulink a implementar en la FPGA
- Modelo CPU

Para explicar esta configuración se va a tomar como ejemplo un modelo muy simple, un inversor modulado en abierto.

#### 3.4.1 Modelo CPU

El modelo CPU, como su nombre indica, se ejecuta en la CPU de la OPAL, no en la FPGA, y actúa como intermediario hacia el modelo implementado en la FPGA. Es decir, sirve como conexión entre el modelo eléctrico que se ejecuta en la FPGA y las entradas y salidas que tiene: analógicas y digitales. También permite modificar ciertos parámetros del modelo en tiempo real, desde la interfaz con el usuario.

El modelo CPU está basado en una estructura Simulink jerárquica, que utiliza subsistemas. El top level del modelo CPU sólo debe mostrar dichos subsistemas. El objetivo de esta jerarquía es doble:

- Distinguir entre sistemas computacionales y la interfaz gráfica del usuario GUI (Graphical User Interface)
- Asignar los bloques computacionales a distintos núcleos del simulador CPU.

Tiene dos bloques bien diferenciados: el máster, que es el subsistema computacional, y la consola, que es la interfaz con el usuario.

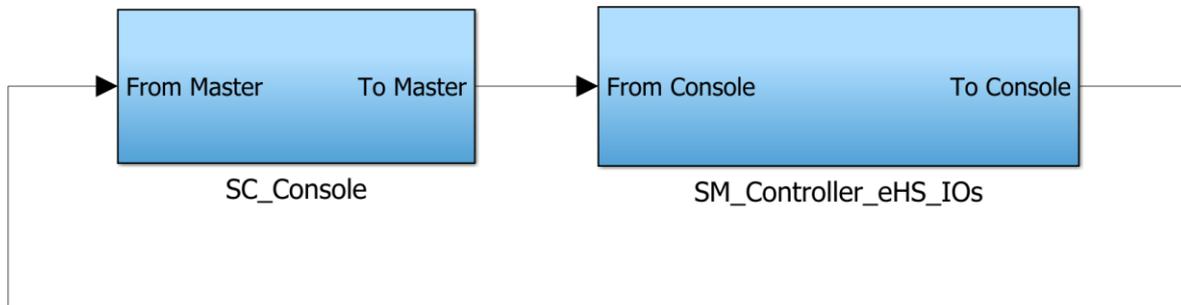


Figura 3-8. Subsistemas del modelo CPU

Los nombres de ambos bloques deben seguir una estructura muy concreta para que el compilador de la OPAL los reconozca: SM\_nombre (bloque máster) y SC\_nombre (bloque consola), como se muestra en la figura 3-8.

El subsistema computacional (master) se ejecutará en tiempo real en un núcleo de la CPU del simulador en tiempo real (OPAL), mientras que la consola o subsistema GUI se ejecutará como un archivo MATLAB/Simulink en el ordenador desde el que se está ejecutando el proyecto de RT-LAB. La comunicación entre el sistema computacional (máster) y el sistema GUI (interfaz) es una comunicación asíncrona a través TCP/IP desde el PC a la OP4510.

Al dividir el sistema en distintos subsistemas, se debe configurar la comunicación entre los distintos subsistemas, para que funcionen correctamente durante la simulación. Para ello, se inserta un bloque extra, el OpComm, en cada uno de los subsistemas. Este bloque es el responsable de la comunicación entre dos subsistemas computacionales entre sí o entre un subsistema computacional y el subsistema GUI. El bloque OpComm debe insertarse una vez que el subsistema ha sido creado y nombrado correctamente. Más adelante se hará referencia a este bloque, cuando se explique la información que pasa entre el bloque máster y el bloque GUI de la CPU.

### 3.4.1.1 Bloque máster

El bloque máster o subsistema computacional contiene elementos que interactúan directamente con el modelo eléctrico que se ejecuta en la FPGA. En concreto, el bloque más importante de este subsistema es el bloque eHSx64 solver, en el que se detallan todas las entradas y salidas, analógicas y digitales, que recibe el modelo ejecutándose en la FPGA.

El papel del bloque computacional puede cambiar en función del tipo de sistema de simulación que se esté implementando. Por ejemplo, en una aplicación RCP, el bloque máster es el subsistema computacional que contiene el controlador diseñado.

El objetivo de este proyecto es implementar un sistema de simulación HIL, lo que significa que el bloque computacional hace de mero intermediario entre el modelo que se ejecuta en la FPGA y las entradas y salidas que recibe. Los distintos bloques que hay en el subsistema computacional se representan en la figura 3-9, y, a continuación, se procede a su explicación detallada.

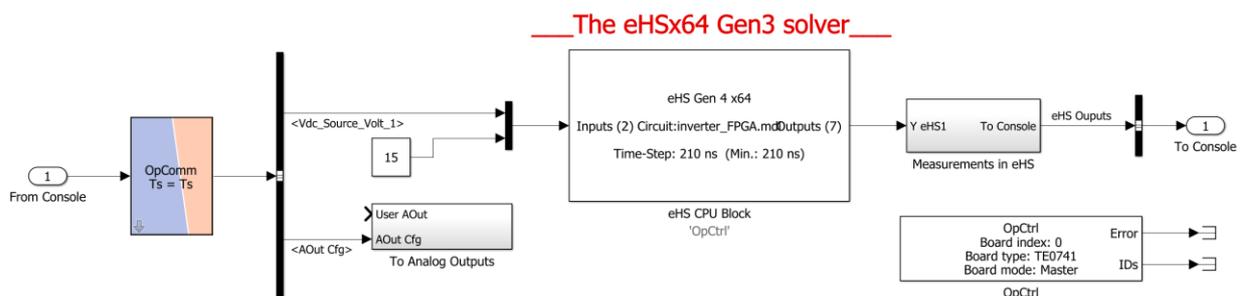


Figura 3-9. Subsistema computacional o máster

### Bloque eHS solver

El bloque eHS solver tiene un papel fundamental a la hora de implementar un modelo eléctrico en la FPGA de la OPAL 4510. En él se especifican todas las entradas y salidas del sistema que se ejecuta en la FPGA en tiempo real. Posteriormente, cuando se explique el modelo FPGA del sistema, se detallarán cada una de las entradas y salidas del modelo y qué necesita recibir desde la CPU para ejecutarse correctamente. Sin embargo, en líneas generales, este bloque sirve para configurar tres tipos de elementos:

1. **Inputs o entradas al modelo.** Se refiere a las fuentes eléctricas que haya en el modelo eléctrico. El circuito implementado en la FPGA necesita que le proporcionen desde el modelo CPU el valor de todas las fuentes eléctricas que haya en el modelo eléctrico.
2. **Gates o señales de control (señales de disparo).** Son las señales de disparo de los elementos switch, a través de las cuales se controla el funcionamiento del convertidor de potencia. En este proyecto las genera el DSP (hardware de control externo) y entran a la FPGA a través de las entradas digitales de la OPAL.
3. **Outputs o salidas del modelo.** Son las medidas de los sensores implementados en el modelo FPGA. Generan unas señales en tiempo real (tensiones e intensidades) que permiten estudiar el comportamiento del circuito eléctrico en tiempo real.

#### 3.4.1.2 Bloque consola

El bloque consola o subsistema GUI se implementa en el ordenador desde el que se ha configurado la CPU y FPGA de la OPAL a través del software RT-LAB. Se ejecuta como un archivo MATLAB/Simulink de manera asíncrona, utilizando el protocolo TCP/IP para comunicarse con la OPAL. Este bloque no está asociado a ningún núcleo CPU de la OPAL, y sólo contiene bloques de interfaz con el usuario. Los bloques que hay típicamente en un subsistema GUI se representan en la figura 3-10.

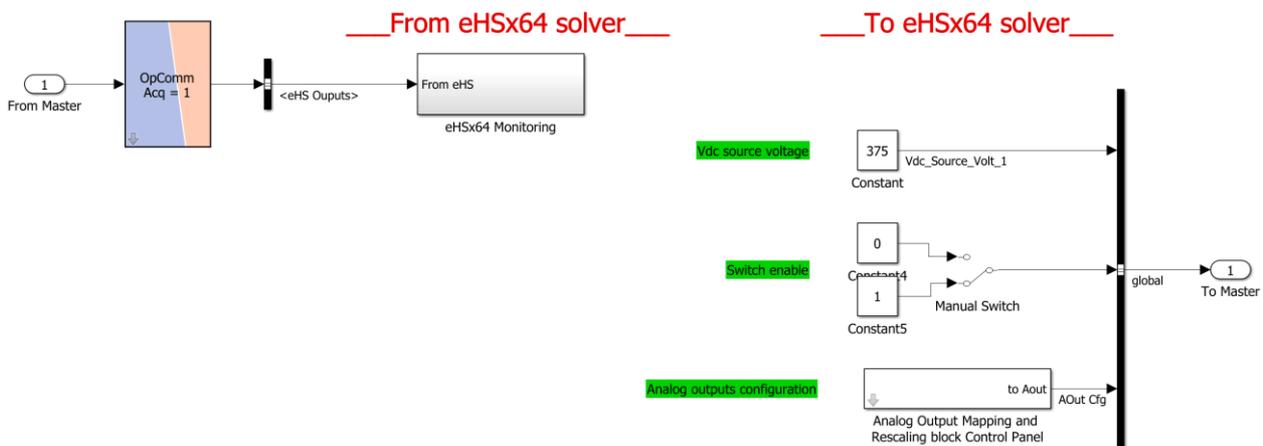


Figura 3-10. Subsistema GUI típico

#### Información recibida desde el bloque computacional

El bloque OpComm se encarga de recibir la información que genera el modelo eléctrico de la FPGA, y el subsistema eHSx64 Monitoring de representarla en scopes en tiempo real. Estos dos bloques se representan en la figura 3-11, y la figura 3-12 es el interior del subsistema eHSx64 Monitoring, donde se pueden ver los scopes de representación en tiempo real. Como se ha explicado anteriormente, los datos generados en la FPGA pasan al bloque computacional del modelo CPU a través del bloque eHSx64 solver, para finalmente ser enviados al bloque GUI, también en el modelo CPU, a través del bloque OpComm.

En el bloque GUI, estos datos se representan en scopes, teóricamente en tiempo real de ejecución. Sin embargo, las distintas etapas de procesamiento y el hecho de que es un modelo Simulink ejecutándose en la CPU del ordenador host (lo que limita el time step mínimo de ejecución) hacen que esta representación no sea a tiempo real. Para estudiar las magnitudes que se generan en la FPGA en tiempo de ejecución se deben medir las señales que genera la OPAL en los puertos de salida analógicos con un osciloscopio. Esto proporciona una representación más precisa de la evolución de las señales en tiempo real. Aun así, los scopes de la interfaz GUI proporcionan una buena primera aproximación de lo que está ocurriendo en el circuito.

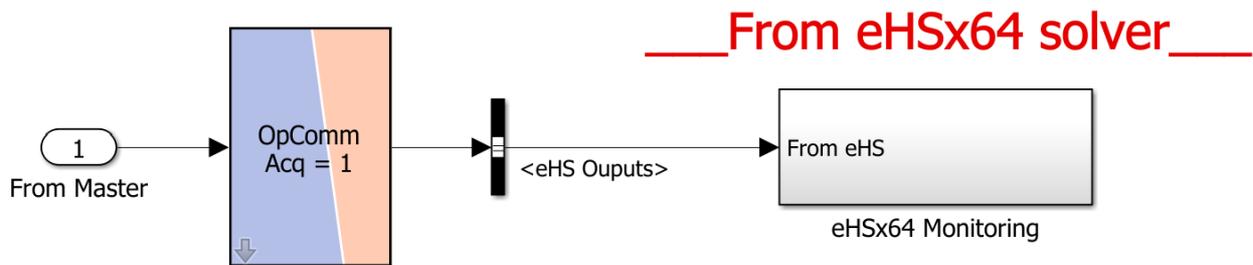


Figura 3-11. Bloque OpComm y representación de las medidas del circuito

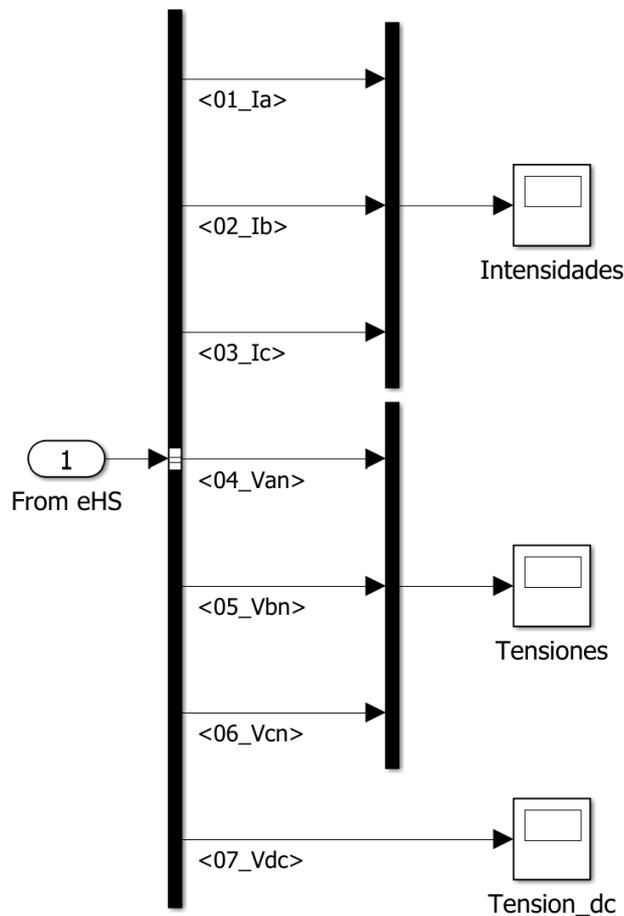


Figura 3-12. Representación de las medidas en tiempo real (aproximación)

### Información enviada hacia el bloque computacional

El puerto “To Master” desempeña el papel opuesto al bloque anterior: permite al usuario modificar en tiempo real ciertos parámetros del modelo eléctrico. De esta forma, se envía información desde la interfaz al modelo eléctrico ejecutándose en la FPGA.

El camino que sigue la información enviada es el mismo que el de la información recibida, pero a la inversa. Es decir, a nivel de CPU, primero se envían los datos desde el bloque GUI al bloque computacional, que los recibe a través del bloque OpComm. A continuación, el bloque computacional los envía al modelo implementado en la FPGA a través del eHSx64 solver. De esta manera, el usuario puede modificar distintos parámetros en tiempo real, y estudiar cómo se comporta el sistema ante estos cambios.

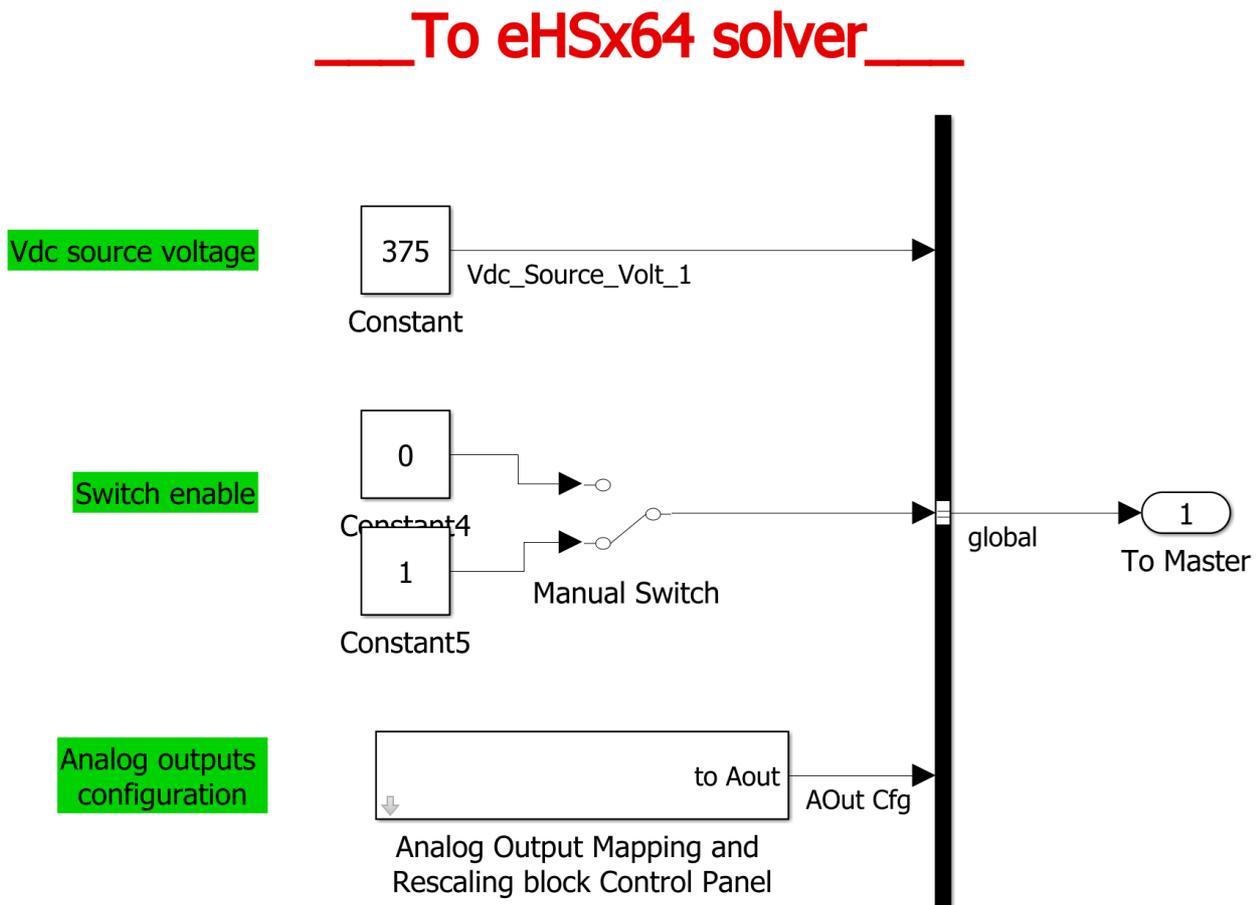


Figura 3-13. Usos típicos de las salidas del bloque GUI

En la figura 3-13 se representan algunas funciones típicas:

1. **Control del valor de una fuente eléctrica:** como se explicará a continuación, el modelo FPGA necesita que le pasen los valores de todas las fuentes eléctricas que haya en el circuito. Conectando este “input” del bloque eHSx64 a una variable de la CPU que el usuario pueda modificar desde la interfaz, se puede controlar el valor de las fuentes eléctricas en tiempo real.
2. **Control de un switch:** Otro uso es asociar a un elemento switch del modelo FPGA una señal CPU, controlada por un switch manual desde la interfaz. De esta manera, se podrían abrir y cerrar interruptores en el modelo de la FPGA en tiempo real, y estudiar así el comportamiento del sistema ante distintas configuraciones (por ejemplo, los estados sucesivos en la precarga de un condensador que se conecta a la red a través de un inversor).

Por último, el tercer bloque tiene una funcionalidad distinta. Este bloque no tiene como objetivo enviar

información al modelo de la FPGA, sino configurar los canales analógicos a través de los cuales salen las medidas que se están tomando en dicho modelo.

Hay 32 canales de salida analógicos en la OPAL 4510. Como se explicará a continuación, cada medida que se toma en el modelo FPGA tiene asociado un nombre, que sigue la nomenclatura YXX. Este bloque permite asociar una medida a cada canal, como puede observarse en la figura 3-14 en la que, al canal cero (pestaña “ch0”) se le ha asociado la medida correspondiente al sensor Y01, en el primer campo del bloque de configuración, “signal”.

También permite configurar la ganancia y el offset que se le aplica a la señal de salida analógica, y el rango de tensión máxima, a partir del cual debe saturar. Estos parámetros son muy útiles a la hora de configurar la comunicación OPAL-DSP. Aunque en apartados posteriores se explicará con más detalle, en líneas generales, estos parámetros sirven para adaptar el valor de un sensor de tensión AC [-400, 400] V al rango de tensión de los puertos de salida analógicos de la OPAL, [-16, 16] V. De esta manera, a la medida que toma el sensor en la FPGA se le aplica una ganancia que transforma el rango de medida del sensor real en el rango de tensión que proporciona la OPAL, obteniendo un equivalente de la medida real a la salida analógica de la OPAL.

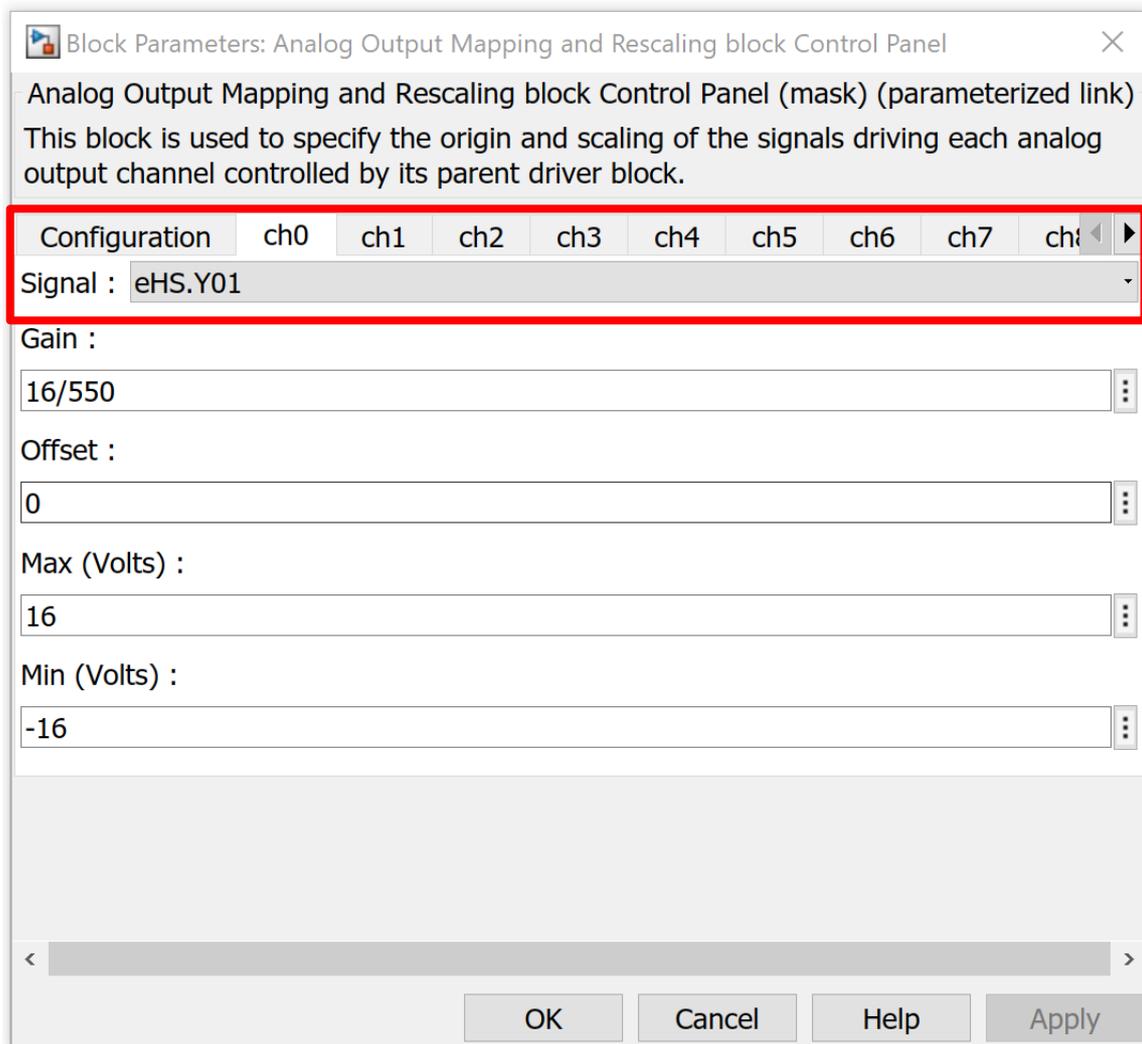


Figura 3-14. Analog Output Mapping and Rescaling block Control Panel

Para funcionar correctamente, este bloque situado en el subsistema GUI del modelo CPU necesita un compañero, el bloque “To Analog Outputs”, situado en el subsistema computacional del modelo CPU (puede verse en la figura 3-10). Este segundo bloque hace efectivos en el modelo computacional los cambios que se hacen desde la interfaz. La acción conjunta de ambos bloques permite modificar en tiempo real la asignación medida-canal analógico de salida.

### 3.4.2 Modelo FPGA

El otro archivo hay que generar para poder ejecutar una simulación HIL es el modelo FPGA. Este archivo es un fichero .mdl, que contiene el circuito eléctrico que se va a implementar en la FPGA de la plataforma de simulación del tiempo real (OPAL 4510).

Como se comentó en el apartado 3.1, el circuito eléctrico se diseña en Simulink, y el software RT-LAB se encarga de implementarlo en la FPGA de la OPAL. El circuito eléctrico se implementa en la FPGA de la OPAL sin que el usuario tenga que programarla haciendo uso de lenguajes HDL (Hardware Desing Languages). Al implementar el circuito eléctrico en la FPGA, el time step mínimo de ejecución disminuye drásticamente, hasta los nanosegundos, de manera que es capaz de emular el comportamiento de un sistema eléctrico real.

Para comprobar que el modelo es correcto, es una buena práctica ejecutarlo offline una vez diseñado. Si el modelo necesita de señales externas o bucles de control para comprobar su funcionamiento, se pueden generar dentro del propio fichero. Esto no representa ningún problema porque una vez implementado en la FPGA, se ignoran todos los bloques que no sean de naturaleza eléctrica.

Para que el compilador de RT-LAB funcione correctamente y reconozca los elementos del circuito eléctrico, este modelo sigue una nomenclatura concreta. Se pueden distinguir tres elementos de distinta naturaleza: fuentes, señales de control y medidas.

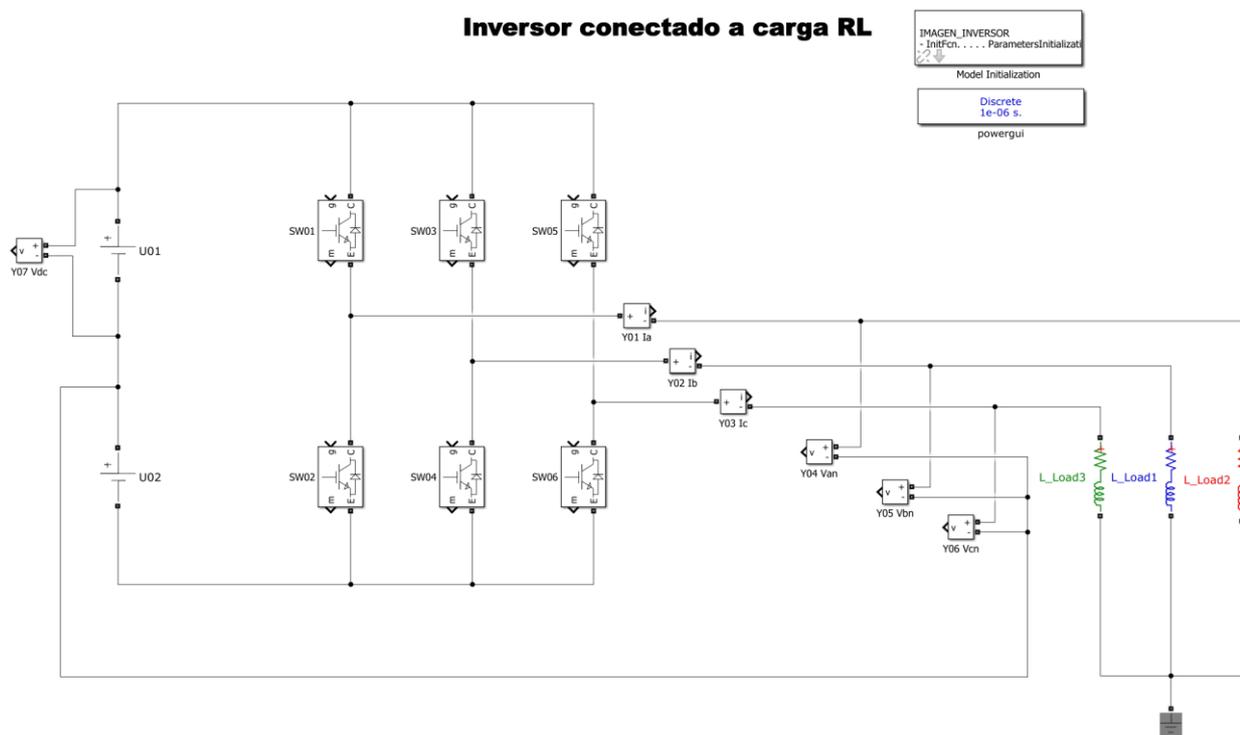


Figura 3-15. Modelo FPGA

#### 3.4.2.1 Fuentes

Son las fuentes eléctricas del modelo. Se nombran como “UXX” (U01, U02, U03...) y es importante que estén numeradas de forma consecutiva.

Como se indicó en el apartado 3.4.1.2, al explicar la información enviada desde el subsistema GUI al subsistema computacional, es necesario que el valor de estas fuentes venga dado a través del modelo CPU, ya sea por una señal generada dentro del bloque computacional o por una señal manipulable dese del bloque GUI.

Esto se configura a través del bloque eHSx64 solver, en la pestaña “Inputs”. Abriendo el desplegable de configuración, aparece un listado de las fuentes eléctricas presentes en el modelo de la FPGA, a las que previamente se les ha asignado un nombre de acuerdo a la nomenclatura “UXX”.

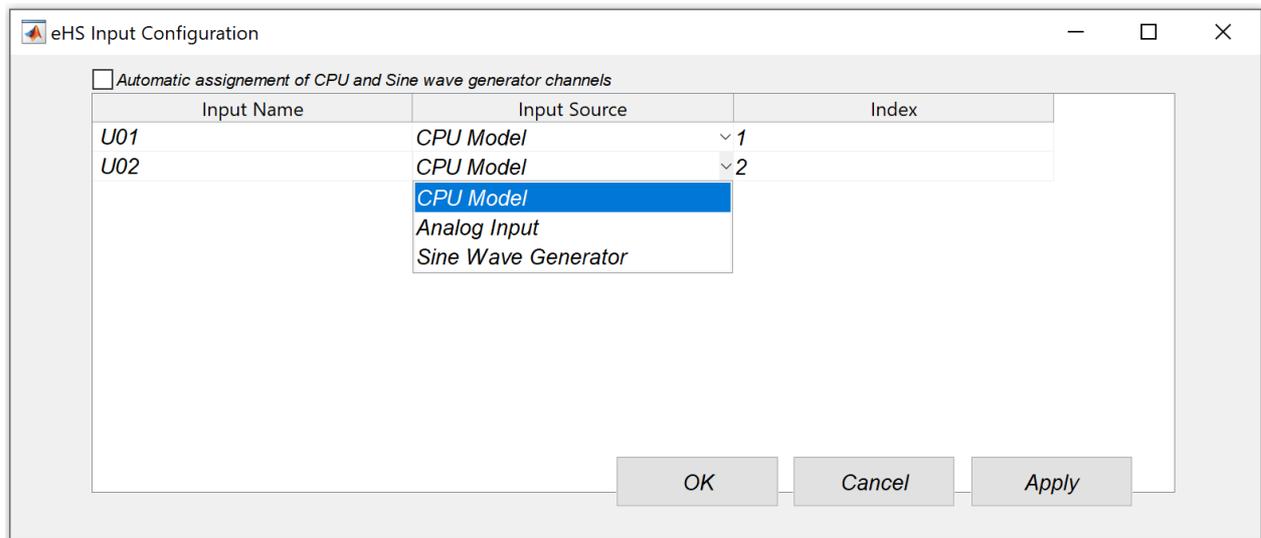


Figura 3-16. Configuración de los inputs

Este listado permite definir, para cada fuente del circuito eléctrico, el origen de la señal que se le asocia y, dentro de las señales de ese tipo que llegan al modelo CPU, el índice o el canal que ocupa. En este proyecto, sólo hemos utilizado inputs que vienen del modelo CPU. Es decir, señales que, o bien se definen directamente en el bloque computacional del modelo CPU, o bien se definen desde la interfaz con el usuario, y llegan al bloque computacional a través del bloque OpComm, como se explicó anteriormente.

### 3.4.2.2 Medidas

Las medidas se refieren a la serie de datos que se genera al incluir sensores en el modelo, tanto de tensión como de corriente. Estos sensores también deben seguir una nomenclatura concreta, con el formato “YXX” (Y01, Y02, Y03...), y también deben estar numerados de manera consecutiva.

Estas señales salen del modelo de la FPGA por dos caminos distintos:

1. **Hacia el modelo CPU a través del bloque eHSx64.** En este caso, las medidas que se toman en tiempo real en la FPGA llegan al bloque computacional de la CPU a través del bloque eHSx64. Para poder representar estos datos en tiempo real, se envían a la interfaz de usuario a través del bloque OpComm, y se representan en scopes, como se explicó anteriormente. La representación que se obtiene no es exacta, y se pueden observar algunos saltos en los que la gráfica intenta reajustarse al tiempo de ejecución, pero es una buena primera representación.
2. **Hacia el exterior de la plataforma de simulación en tiempo real, a través de los puertos de salida analógicos.** Estas son las señales analógicas que genera la OPAL en tiempo real, y que, como se explicará más tarde, entran a los canales ADC del DSP. Son las medidas que está esperando el DSP para ejecutar el algoritmo de control y obtener las señales de disparo. Si se miden con un osciloscopio representan las señales del sistema eléctrico en tiempo real. La asignación medida – canal de salida se hace utilizando el bloque “Analog Output Mapping and Rescaling block Control Panel”, como se comentó en el apartado 3.4.1.2, al explicar la información que se envía desde el bloque GUI al bloque computacional.

### 3.4.2.3 Señales de disparo

Las señales de disparo son las señales de habilitación que esperan los elementos switch del modelo eléctrico, como los IGBTs. Al igual que fuentes y sensores, los elementos switch también siguen una nomenclatura concreta, “SWXX” (SW01, SW02, SW03...) y es importante que se numeren en orden consecutivo.

Cada elemento switch del modelo eléctrico tiene asociado una señal de disparo. Estas señales de disparo son el resultado del un algoritmo de control, que hace que el convertidor funcione de la forma deseada.

El control puede estar implementado en distintos dispositivos, lo que hace que estas señales puedan tener distintas fuentes. Si el control está implementado a nivel CPU (bloque computacional), las señales de disparo serán señales CPU. Por otra parte, si las señales de disparo las genera un controlador externo (DSP), que es el caso de este montaje HIL, las señales llegan al modelo que de la FPGA directamente por los puertos de entradas digitales (a través del hardware de adaptación que explicaremos en los siguientes apartados).

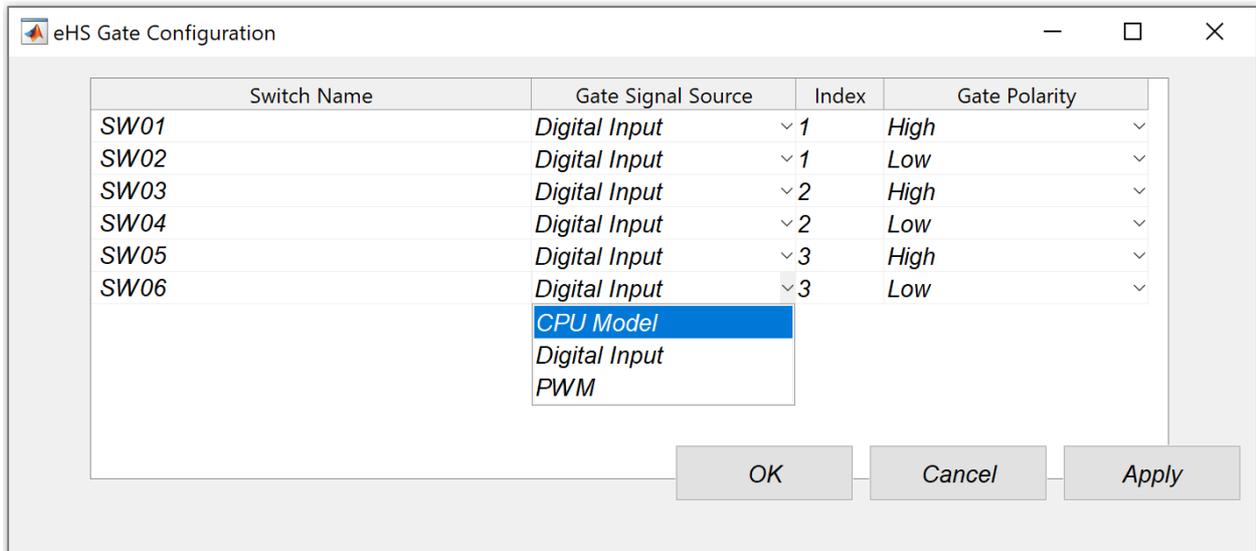


Figura 3-17. Configuración de los gates

El bloque eHSx64 asigna una señal de control a cada bloque switch del circuito. Para cada elemento permite seleccionar la fuente de la señal de control y el índice que ocupa en el vector de señales de entrada (ya sean canales digitales o señales del bloque CPU). También permite seleccionar la polaridad de la señal de control. Esto sirve, por ejemplo, para duplicar las señales de control provenientes del DSP, ya que normalmente, el DSP sólo genera las señales de los IGBTs superiores del convertidor. Cambiando la polaridad de la señal se puede asignar la misma entrada digital a los 2 IGBTs de una misma rama, que es la configuración que se puede observar en la figura 3-17.

# 4 LIMITACIONES DE LA OPAL 4510

El objetivo del sistema de simulación HIL es sustituir el convertidor de potencia real por la plataforma de simulación OPAL 4510, de manera que la plataforma de control reciba los mismos estímulos que si estuviera conectado a un sistema real.

Sin embargo, el software RT-LAB es un software pensado para ciertas aplicaciones, por lo que carece, quizás, de cierta flexibilidad para implementar un sistema con total libertad.

Por esto, en esta sección se va a analizar el montaje y control de un sistema de potencia real, para compararlo con el modelo más aproximado que se ha podido simular, según las reglas de diseño de RT-LAB. Se ponen así de manifiesto algunas limitaciones que se han tenido que sortear.

## 4.1 Comparación entre el sistema real y el sistema implementado

En un sistema de potencia real el hardware de control y el convertidor de potencia se conectan a través de una placa de conexión llamada driver. Esta placa no existe en el sistema de simulación HIL, pero cumple varias funciones esenciales, por lo que es necesario simular su comportamiento en el sistema HIL. Entre sus funciones está implementar una serie de funciones lógicas que adaptan las señales de control que recibe del DSP para pasarle al convertidor el número total de señales las que necesita. Es precisamente este paso de lógica combinatorial el que no se puede implementar en la OPAL y el que hace que no se puedan simular determinados comportamientos. Es decir, el modelo FPGA no permite implementar las funciones lógicas propias del driver, por lo que algunas de las tareas que lleva a cabo sí se han podido emular, pero otras no.

En concreto, la mayor limitación del software RT-LAB a la hora de implementar un circuito eléctrico en la FPGA es que **no permite manipular libremente las señales digitales de entrada**, sino que obliga al diseñador a asociar una señal digital de entrada a cada elemento switch del circuito. Es decir, no se puede asociar una señal digital de entrada a una variable y manipularla libremente dentro del modelo (por ejemplo, para hacer una estructura lógica). Lo único se puede hacer con una señal digital es asociarla a un elemento, a través de la lista “Gates Configuration” del bloque eHS que se explicó en el apartado 3.4.2.3. Esto conlleva ciertas dificultades, que se pondrán de manifiesto en los siguientes apartados.

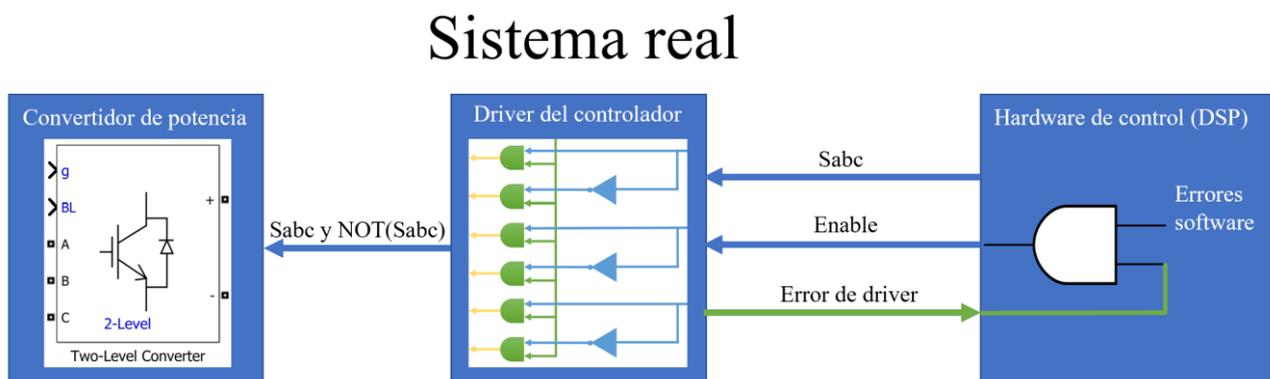


Figura 4-1. Sistema de potencia real

# Sistema HIL

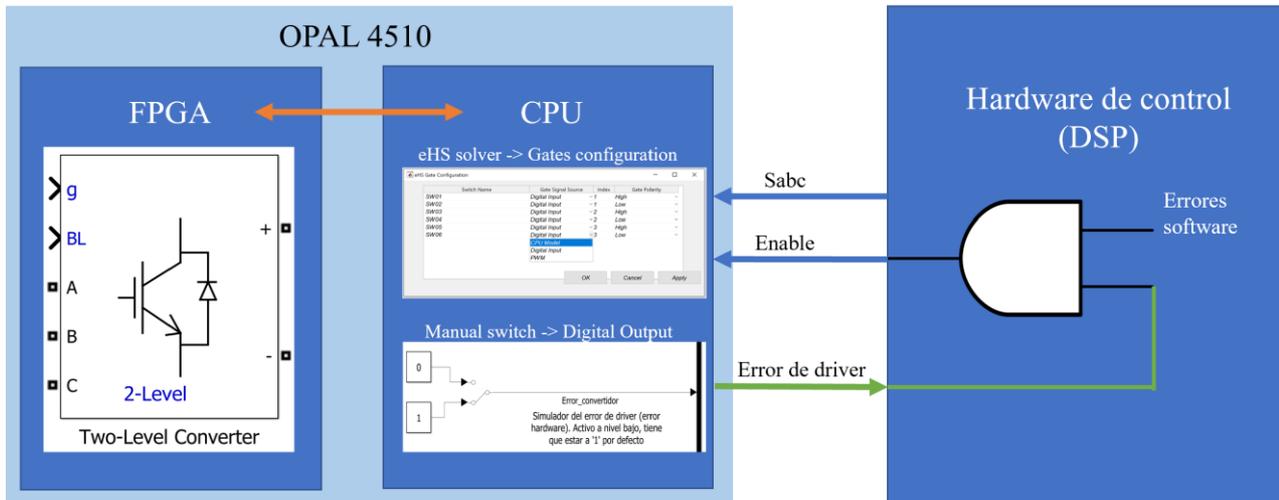


Figura 4-2. Sistema HIL

A continuación se van a ir listando las distintas funciones de la placa driver, para poder explicar por qué no se pueden implementar de manera exacta en la FPGA, y qué soluciones se han buscado. Las figuras 4-2 y 4-3 representan esta adaptación de funcionalidades de la placa driver al sistema HIL, y se irán referenciando en los siguientes apartados.

## 4.1.1 Error de driver (error hardware)

Una de las funciones más importantes de la placa driver es detectar cualquier error hardware que se dé en el circuito del convertidor de potencia. Es decir, los elementos del convertidor de potencia tienen ciertos mecanismos para detectar si se está dando algún fallo a nivel hardware (por ejemplo, un cortocircuito). En ese caso, la placa driver envía una señal de error al DSP para que el control entre en el estado de error y se abran todos los IGBTs. Esta señal se representa en verde en las figuras 4-1 y 4-2. En el sistema real la genera la placa driver, y en la simulación HIL se genera en la CPU y se envía a través de las salidas digitales de la OPAL.

Para el algoritmo de control utilizado en este proyecto, esta señal es activa a nivel bajo, por lo que debería estar normalmente a '1' y ponerse a '0' cuando haya un error. Esta señal de error se conoce como el error de la placa driver o error hardware.

En el algoritmo de control que se ejecuta en el DSP se implementa una función AND de todos los errores (tanto software como hardware) y el resultado de esta función lógica implementada con código es la señal "Enable". De esta forma, si la placa driver detectara un error hardware, la señal de error procedente del driver se pondría a '0' y la señal enable también se pondría a '0', abriendo todos los IGBTs.

En el sistema de simulación no hay hardware, ya que el convertidor de potencia está simulado en la FPGA, por lo que no se genera ningún error hardware. Sin embargo, el DSP está configurado para recibir esta señal, que además es activa a nivel bajo, por lo que si no recibe ninguna señal por esa fibra receptora, interpreta un cero lógico, y el control no saldría del estado de error.

Por eso, se usa una salida digital desde la CPU que se pueda cambiar con un manual switch desde la consola entre 0 y 1. Esta señal va a simular el error hardware que genera la placa driver y llega al controlador. Normalmente vale '1', pero se puede cambiar a '0' desde la consola, para así ver cómo respondería el controlador ante un error hardware del sistema de potencia.

### 4.1.2 Desdoble de señales de control

El hardware de control o DSP genera tan sólo 3 señales de control, que se corresponden con las señales de disparo de los IGBTs superiores del convertidor,  $S_{abc}$  (figuras 4-2 y 4-3). Sin embargo, el convertidor de potencia necesita 6 señales de control, porque en cada rama hay 2 IGBTs,  $S_{abc}$  y  $\text{NOT}(S_{abc})$  (figuras 4-2 y 4-3).

Esta es otra de las funciones que cumple la placa driver, desdoblar las señales de control para que haya 1 señal para cada IGBT. El circuito es muy simple:

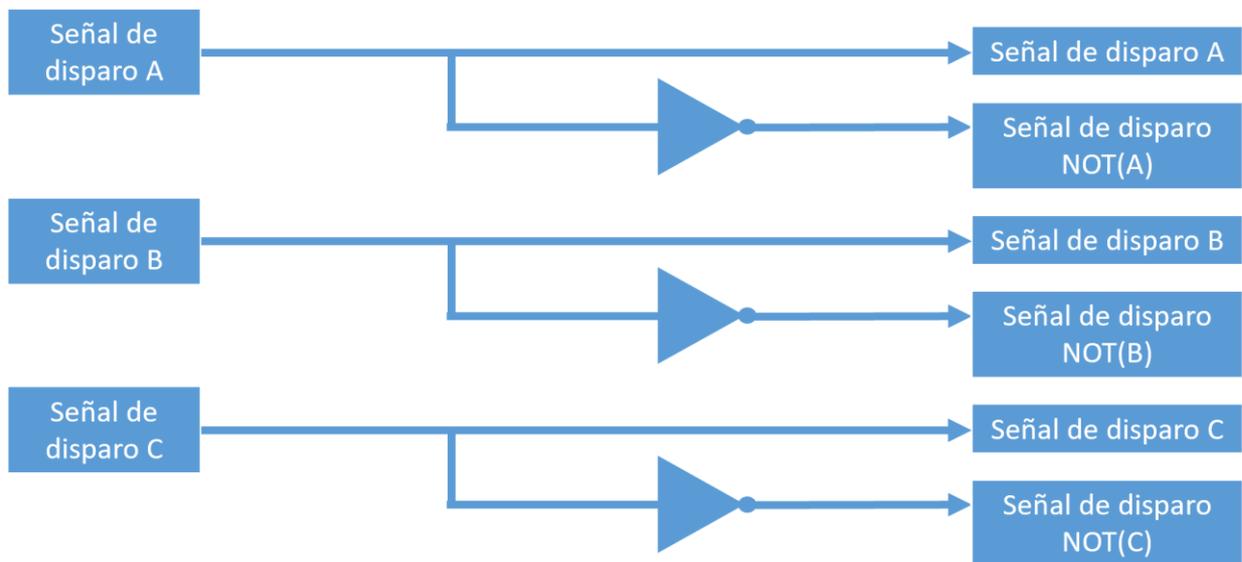


Figura 4-3. Desdoble de las señales de control

Esta función lógica se puede implementar fácilmente en la placa driver. Sin embargo, no se puede implementar en el modelo FPGA de la OPAL, porque no hay ningún bloque que permita recibir la señal digital y manipularla libremente para construir una función lógica como esta, para luego asociarla a un IGBT.

En primer lugar se pensó en asociar cada una de estas tres señales a un IGBT siguiendo esta estructura:

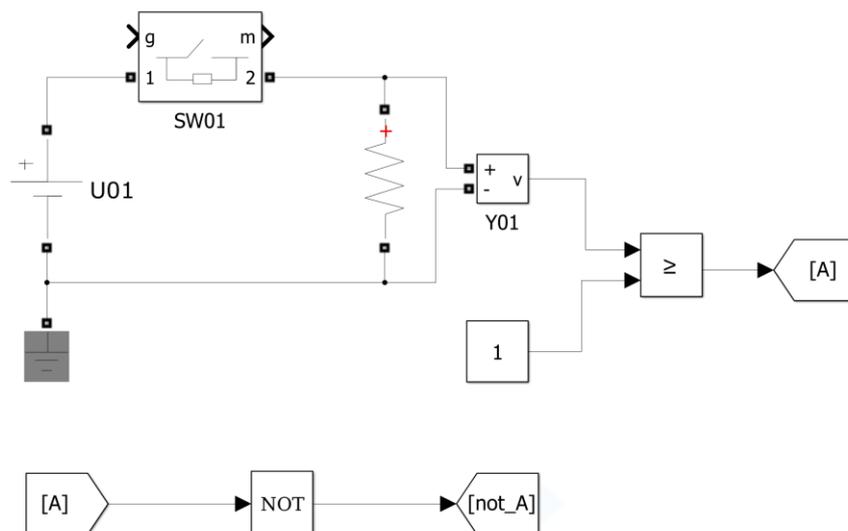


Figura 4-4. Posible solución para poder manipular las entradas digitales

Asociando la señal digital de entrada al SW01 en el bloque eHS (CPU), tendríamos control sobre la señal A: cuando la señal vale ‘1’ el IGBT está cerrado y la tensión que mide Y01 es mayor que 1 (U01= 5V), por lo que la señal A se pondría a ‘1’. En caso contrario, A valdría ‘0’.

Se puede llegar a pensar que a través de esta estructura se pueden manipular las señales digitales, y así poder implementar la función lógica que se necesita. Es cierto, se puede implementar la lógica, y se obtendrían las señales correctamente, pero no se pueden asociar a los IGBTs. A pesar de que el puerto de control del IGBT no desaparece, al implementarlo en la FPGA está inutilizado: el IGBT sólo responderá a la señal que se le asocie en la lista “Gates Configuration” del bloque eHS. Por tanto, esta estructura no serviría.

Otra posible solución, que se valoró en su momento, sería en vez de leer las señales de control directamente en la FPGA, leerlas en la CPU, manipularlas, implementar la función lógica, y luego asociarlas a los distintos IGBTs en el bloque eHS, pero en vez de como “Digital In” como señales “CPU”. Esto sí funcionaría, y los IGBTs estarían controlador por las señales correctas.

Sin embargo, esta opción se descartó porque la principal característica que tiene este tipo de simulación en tiempo real es que el modelo se implementa y se ejecuta en la FPGA, y el time step de ejecución llega a los nanosegundos. Es decir, el muestreo de las señales de entrada (señales digitales de control) se ejecuta cada muy pocos nanosegundos. Al pasar la lectura de las entradas digitales a nivel CPU, se estaría perdiendo el principal avance que se pretende conseguir con la OPAL, ya que el tiempo de muestreo en la CPU es mucho mayor (del orden de los microsegundos). Por eso, esta opción también se descartó.

Sin embargo, realmente para desdoblarse las señales no existe ningún problema, porque el listado de “Gates Configuration” del bloque eHS nos permite asociar a dos IGBTs la misma señal, y además cambiarles la polaridad, de manera que una sea activa a nivel alto y la otra sea activa a nivel bajo.

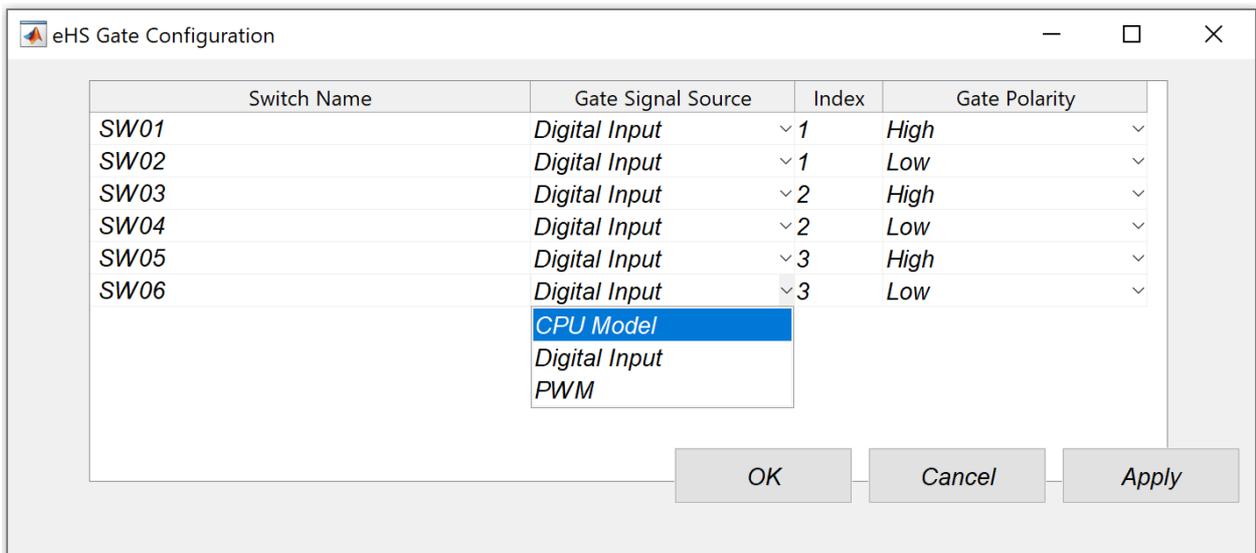


Figura 4-5. Configuración de los gates

Es una manera muy sencilla de generar las 6 señales. Sin embargo, tiene un defecto: aunque tres de los IGBTs estén a ‘0’, los otros tres siempre estarán a ‘1’, de manera que no hay ningún caso en el que se puedan abrir todos a la vez. Esto supone una gran desventaja, porque hay situaciones en las que se requieren que todos los IGBTs estén abiertos (por ejemplo durante la fase de precarga, o un error del convertidor), pero no es posible simular este estado del convertidor por la propia configuración de señales digitales que nos ofrece el software RT-LAB.

### 4.1.3 Función del enable

Otra de las funciones de la placa driver es implementar el efecto de la señal “enable”, que también se representa en las figuras 4-2 y 4-3. Esto es un circuito lógico que hace que cuando enable valga ‘0’, todas las señales de control de todos los IGBTs se pongan a ‘0’, abriendo todos los elementos de conmutación.

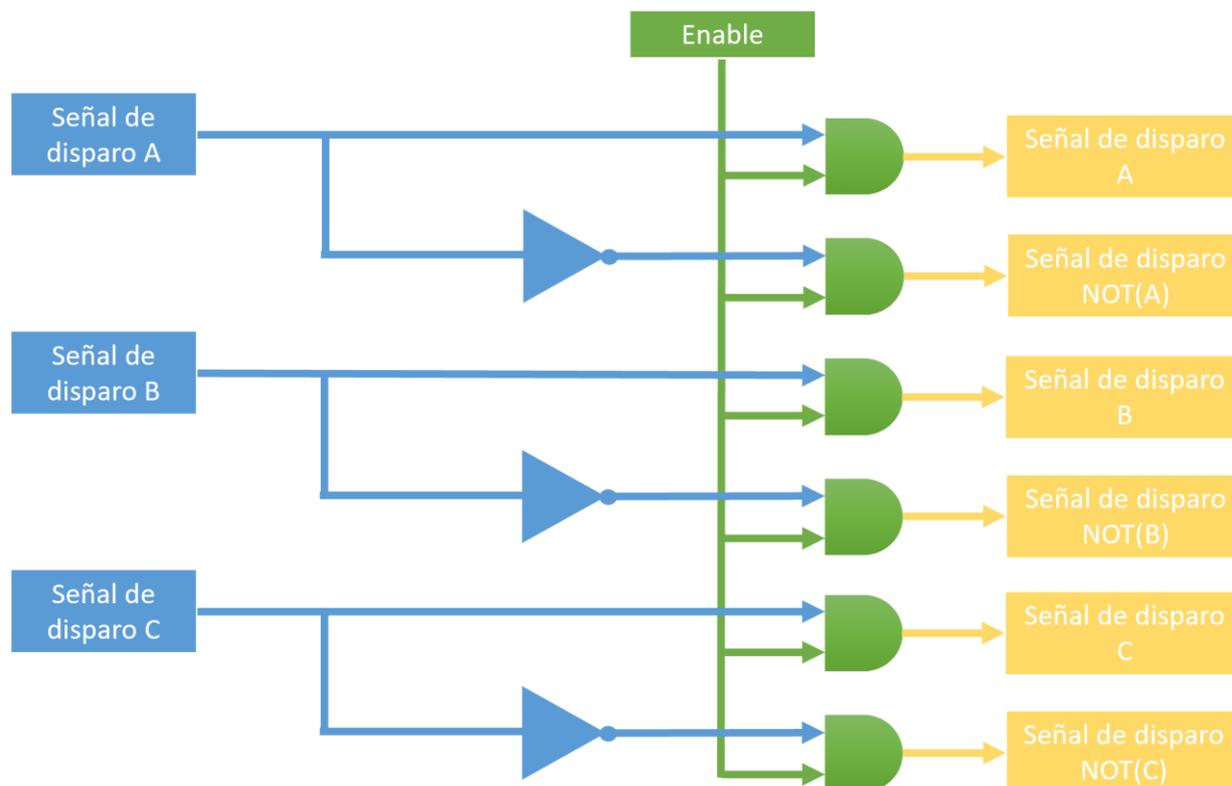


Figura 4-6. Función de la señal “Enable”

El enable entra en acción en situaciones en las que se requiere que todos los IGBTs estén abiertos. Por ejemplo, cuando hay un error en el convertidor. Sin embargo, por todas las razones que se han expuesto en el apartado anterior, no es posible implementar esta funcionalidad en el modelo de la FPGA.

Por tanto, el controlador genera la señal enable, pero no tiene efecto alguno en el modelo porque no se recibe, ya que no se puede implementar la función lógica que haría que esta señal tuviera sentido. Esta funcionalidad del driver no se ha podido implementar en el modelo FPGA, por lo que no es un modelo de convertidor fiel a la realidad. En otras palabras, : siempre que 3 estén cerrados, los otros 3 estarán abiertos, debido al tratamiento de señales digitales de entrada que permite el software, por lo que nunca estarán todos abiertos.

## 4.2 Circuito de precarga

Estas restricciones supusieron un problema a la hora de simular un comportamiento concreto, el circuito de precarga.

Uno de los modelos implementados, que se detallará en el apartado 7, consiste en un condensador conectado a red a través de un inversor. El objetivo de este sistema es controlar la tensión del DC-link.

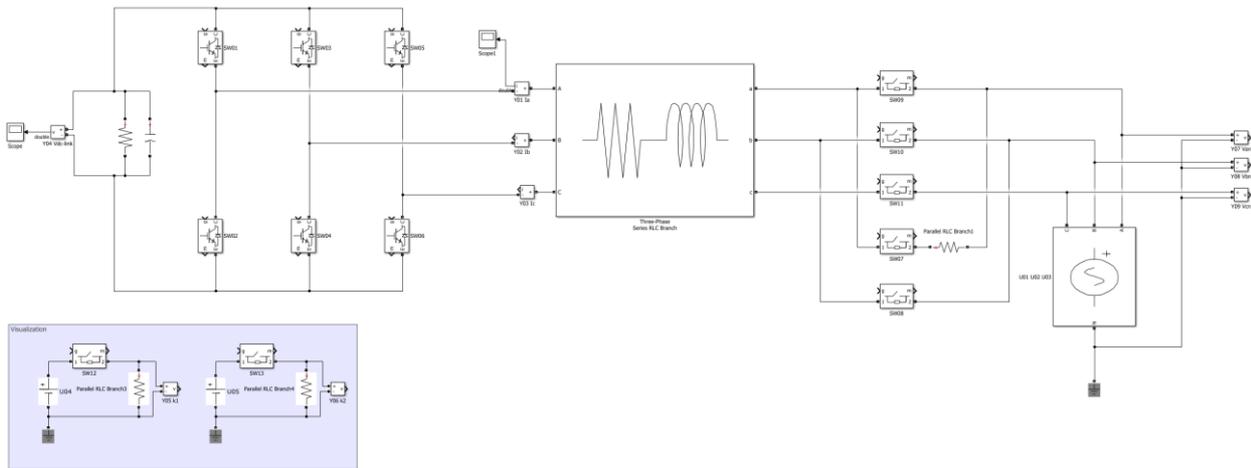


Figura 4-7. Condensador (dc-link) conectado a red a través de un inversor

Sin embargo, antes de comenzar el control, debe realizarse una etapa de precarga del condensador, para que el incremento tensión objetivo-tensión en el condensador no sea tan grande y no provoque un pico de tensión y de intensidad que podría dañar el circuito.

Esto se implementa a través de una serie de switches como se aprecia en la figura 4-7. Pero, se cierran los dos switches inferiores de la parte izquierda del circuito, de manera que el condensador queda conectado a la red a través de una resistencia limitadora de la corriente. El inversor, con todos los IGBTs abiertos ( $Enable=0$ ) actúa a todos los efectos como un rectificador de diodos, por lo que el condensador comienza a cargarse viendo una tensión rectificadora. Una vez que está cargado a un cierto valor, se abren los dos interruptores inferiores, se cierran los tres superiores y comienza el control.

Para que este circuito funcione es vital que durante la precarga todos los IGBTs estén abiertos, quedando un puente de diodos que rectifica la tensión. Sin embargo, no es posible conseguir esta configuración de IGBTs, ya que se definen con polaridades opuestas desde el bloque eHS, por lo que siempre que tres estén abiertos, los otros tres estarán cerrados.

Se proponen dos soluciones. La primera es fijar una tensión inicial del condensador, de manera que cuando comience a ejecutarse el modelo ya esté cargado, y el salto de tensiones no sea tan grande.

Otra posible solución es la que se aplica en el modelo PV. En este modelo, el condensador se conecta a paneles PV, de manera que cuando comienza a ejecutarse el modelo, todos los switches están abiertos, y el PV array carga el condensador a la tensión de circuito abierto propia del panel. Cuando se ha cargado el condensador, comienza el control. Esto se explicará con más detalle en el último apartado.

# 5 PLACAS DE ADAPTACIÓN

Este capítulo se va a dedicar a explicar las placas de adaptación conectadas a la salida de la OPAL. La plataforma de simulación en tiempo real tiene canales de entrada y salida digitales y analógicos. El objetivo último de una simulación HIL es proporcionar al hardware de control los estímulos digitales y analógicos que generaría el sistema real, para que el controlador actúe como si estuviera conectado al sistema real.

Sin embargo, las entradas y salidas de la OPAL tienen un rango concreto, que no tiene por qué coincidir con el rango en el que se mueven las magnitudes en el sistema real. Por eso se necesita un hardware de adaptación que transforme la señal que de la OPAL en una señal que el controlador sea capaz de entender.

Estas placas fueron diseñadas como proyecto de fin de grado por el alumno Jerónimo Chacopino García. Este proyecto, por tanto, tiene como objetivo la puesta en marcha del sistema de simulación completo, integrando estas placas en el montaje.

La primera tarea de este trabajo fue, por tanto, comprobar el buen funcionamiento todos los canales de todas las placas de adaptación, digitales y analógicas.

## 5.1 Canales analógicos y digitales de la OPAL

En primer lugar, es necesario conocer la disposición hardware de los puertos entrada/salida de la OPAL. En concreto, los conectores que se utilizan en este proyecto son, en la figura 5-1, los marcados con la letra “A”, que son los conectores DB37 para entradas y salidas digitales y analógicas.

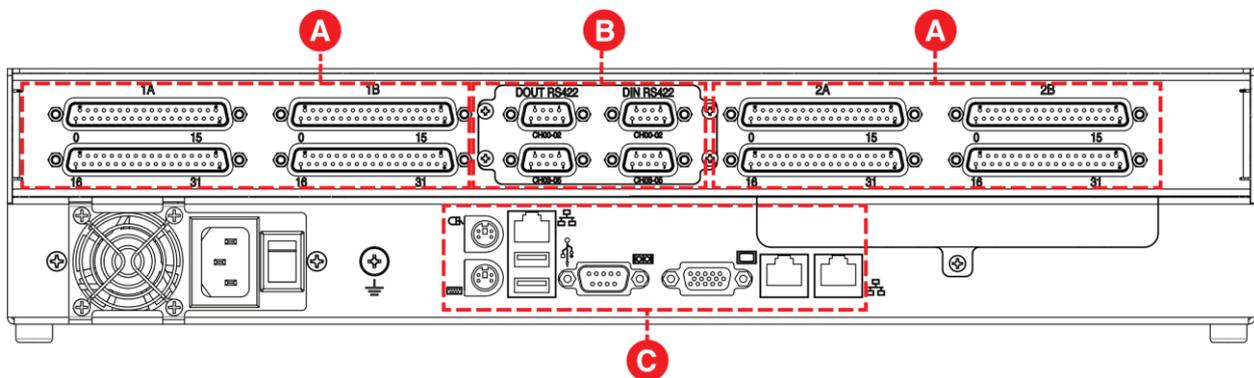


Figura 5-1. Distribución de canales entrada/salida de la OPAL [8]

Los conectores de la derecha son los dedicados a entradas y salidas analógicas, mientras que los de la izquierda, a entradas y salidas digitales, como se muestra en la figura 5-2.

### IO GROUP 1A (CH 00-31): DIGITAL INPUTS



**IO GROUP 1B (CH 00-31): DIGITAL OUTPUTS****IO GROUP 2A (CH 00-15): ANALOG INPUTS****IO GROUP 2B (CH 00-15): ANALOG OUTPUTS**

Figura 5-2. Distribución detallada de canales analógicos y digitales de entrada y salida [8]

El número de canales digitales y analógicos se resume en la siguiente tabla:

Tabla 5-1. Canales analógicos y digitales

Número de canales\Tipo de canal	Entrada	Salida
Canales digitales	32	32
Canales analógicos	16	16

El montaje en el laboratorio queda como se muestra en la figura 5-3.

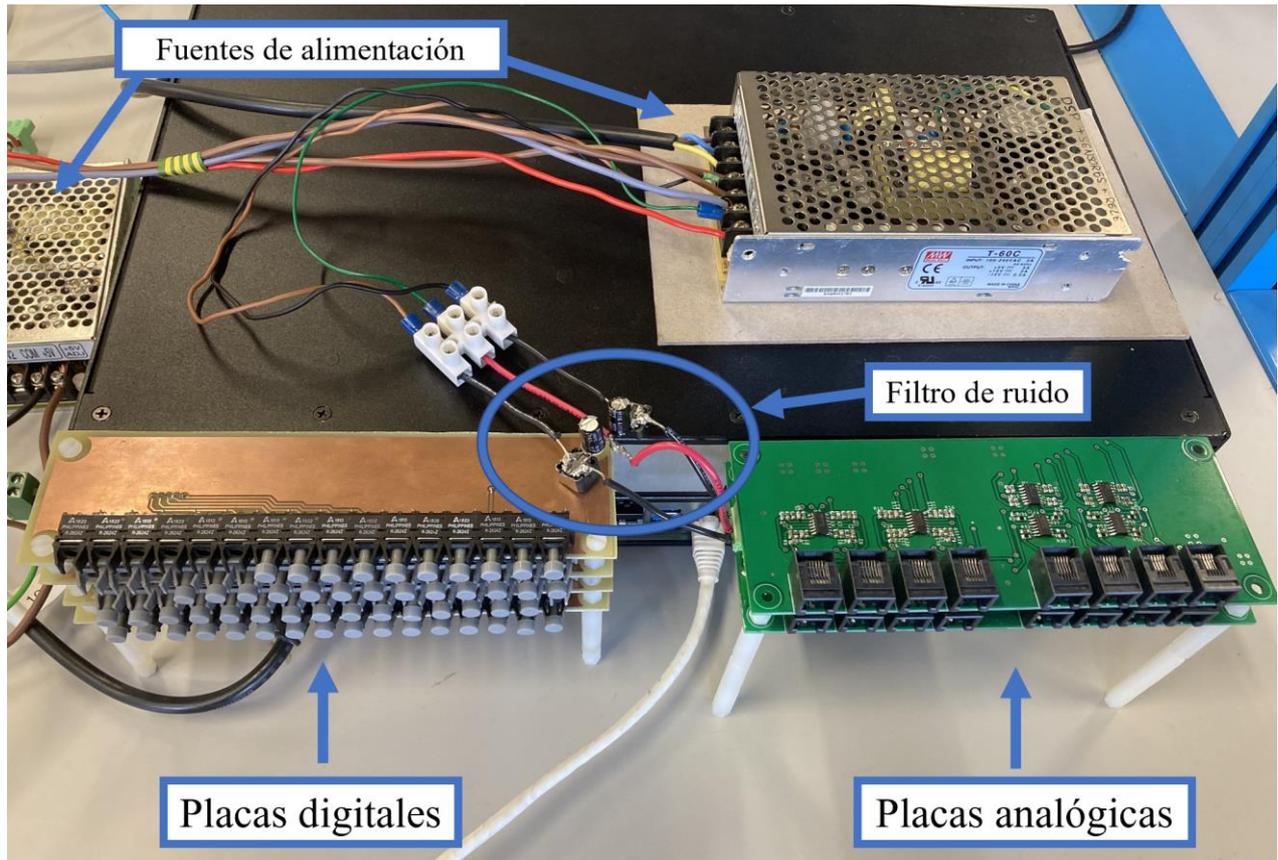


Figura 5-3. Conexión de las placas de adaptación a la OPAL

Cabe destacar que las placas están diseñadas para utilizar todos y cada uno de los canales que tiene la OPAL, por lo que existía un problema de espacio, que se solucionó diseñando las placas en dos niveles. Todo este proceso de diseño está explicado por su autor en el trabajo de fin de carrera. [6]

## 5.2 Entradas y salidas del hardware de control

Para comprender los circuitos de adaptación de las placas, es necesario conocer las características de las entradas y salidas analógicas y digitales a las que se conectarán. Esto depende de la aplicación para la que se esté utilizando la plataforma y el sistema o controlador al que se conecte.

En este proyecto, la plataforma con la que interactúa el sistema es una tarjeta diseñada por el departamento, que actúa como soporte para el microcontrolador 28335 de Texas Instruments. Se van a listar todas las entradas y salidas de la placa, detallando la funcionalidad y características de cada una.

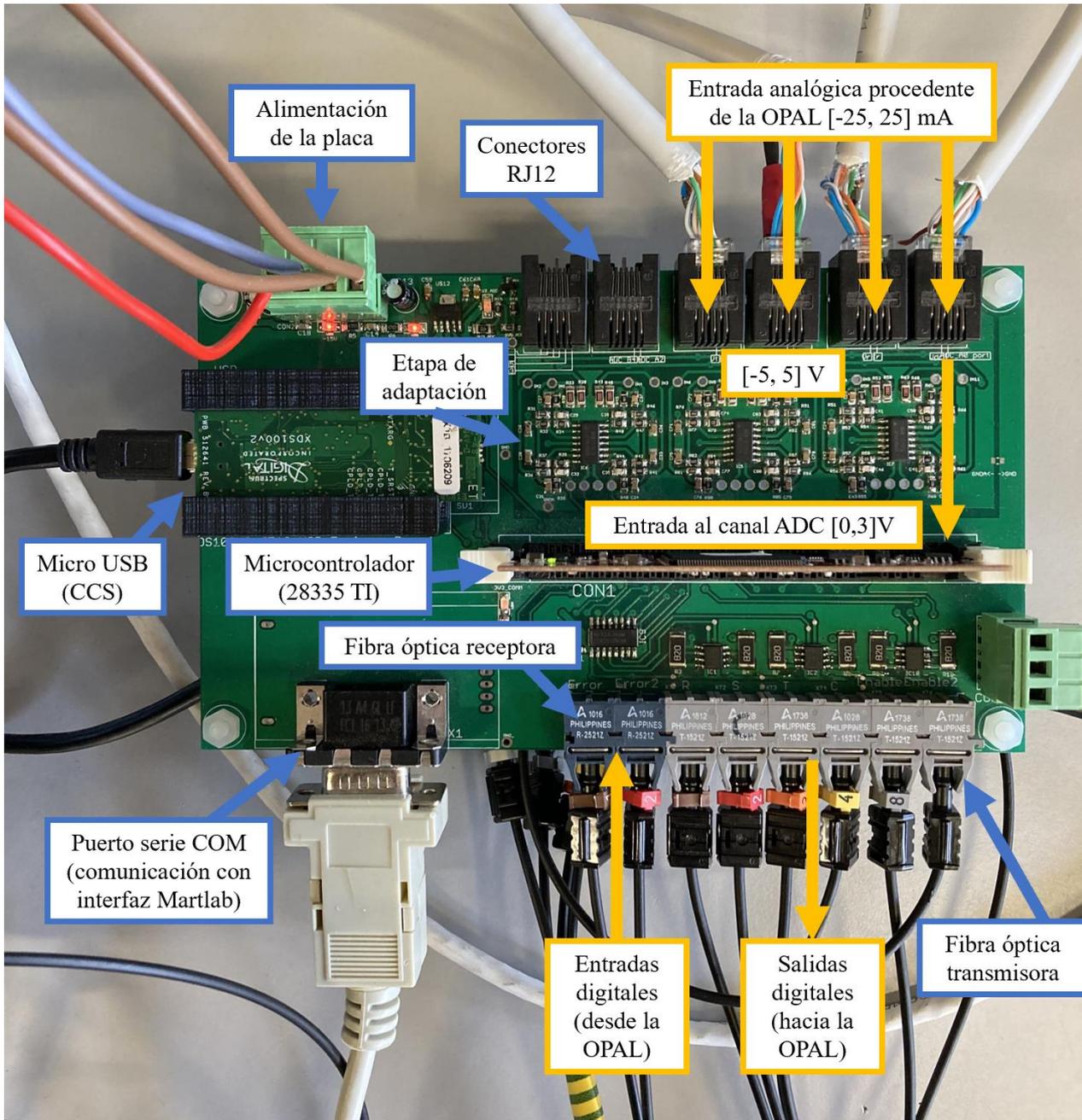


Figura 5-4. Entradas y salidas del hardware de control

### 5.2.1 Entradas analógicas: conectores RJ12

Las medidas analógicas del sistema a controlar entran a la placa del controlador a través de conectores RJ45. En un sistema real, estos cables procederían de un sensor de tensión o de corriente. En el sistema HIL, estas entradas vienen de la OPAL.

A la entrada de la tarjeta se esperan corrientes de  $[-25,25]$ mA, que una etapa de adaptación de entrada convierte en  $[-5,5]$ V, y finalmente en  $[0,3]$ V, que entran al canal de conversión ADC.

### 5.2.2 Entradas y salidas digitales: fibra óptica

El controlador genera unas señales de PWM que deben llegar al sistema para regular los disparos de los

IGBTs, y así controlar el comportamiento del sistema. Estas señales las genera en tensión, pero las convierte a señal de fibra óptica, y las envía a través de transmisores. En el sistema real, esta fibra llegaría a la placa driver del convertidor de potencia. En el sistema HIL, llega a la placa de adaptación digital, y llega directamente al modelo implementado en la FPGA como entrada digital.

### 5.2.3 Comunicación con el ordenador host

La tarjeta que soporta el microcontrolador se conecta al ordenador host a través de dos puertos: el puerto serie COM, que establece la comunicación del DSP con la interfaz de Matlab, y el puerto micro USB, a través del cual se programa el micro a través del software Code Composer Studio.



Figura 5-5. Interfaz Matlab del DSP

### 5.3 Placas analógicas

Las placas analógicas se encargan de adaptar tanto las salidas analógicas de la OPAL al exterior como las entradas analógicas desde el exterior hacia el interior del simulador.

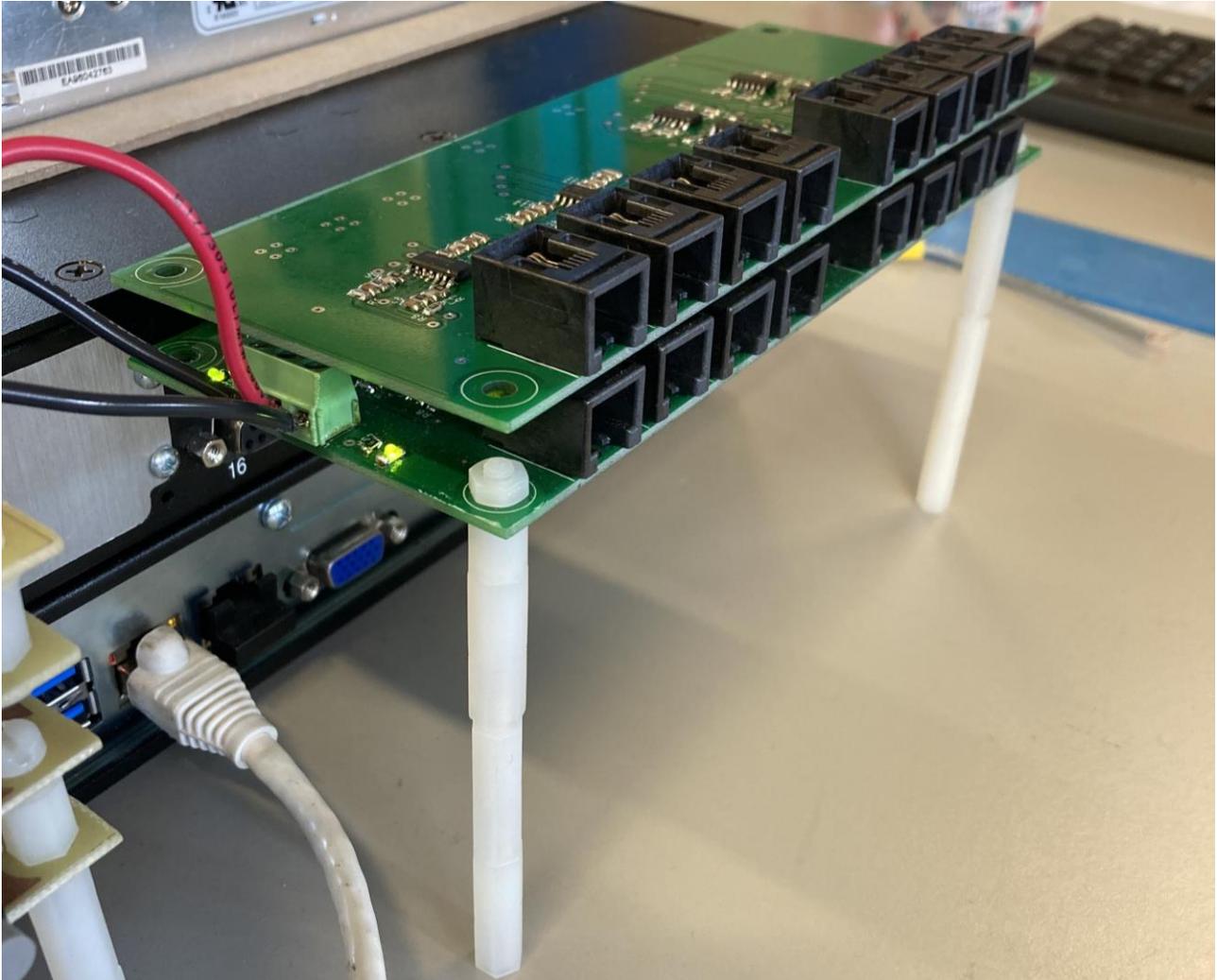


Figura 5-6. Conexión de las placas de adaptación a la OPAL

Las características de los canales analógicos de entrada y salida de la OPAL se resumen en las siguientes tablas:

Tabla 5–2. Características Entradas Analógicas [6]

Número de canales	16
Resolución	16 bits
Rango de tensión	+/-20V

Tabla 5-3. Características Salidas Analógicas [6]

Número de canales	16
Resolución	16 bits
Rango de tensión	+16V
Corriente max	15 mA

Por tanto, se tienen dos partes bien diferenciadas en las placas de adaptación analógicas, que responden a circuitos de adaptación distintos.

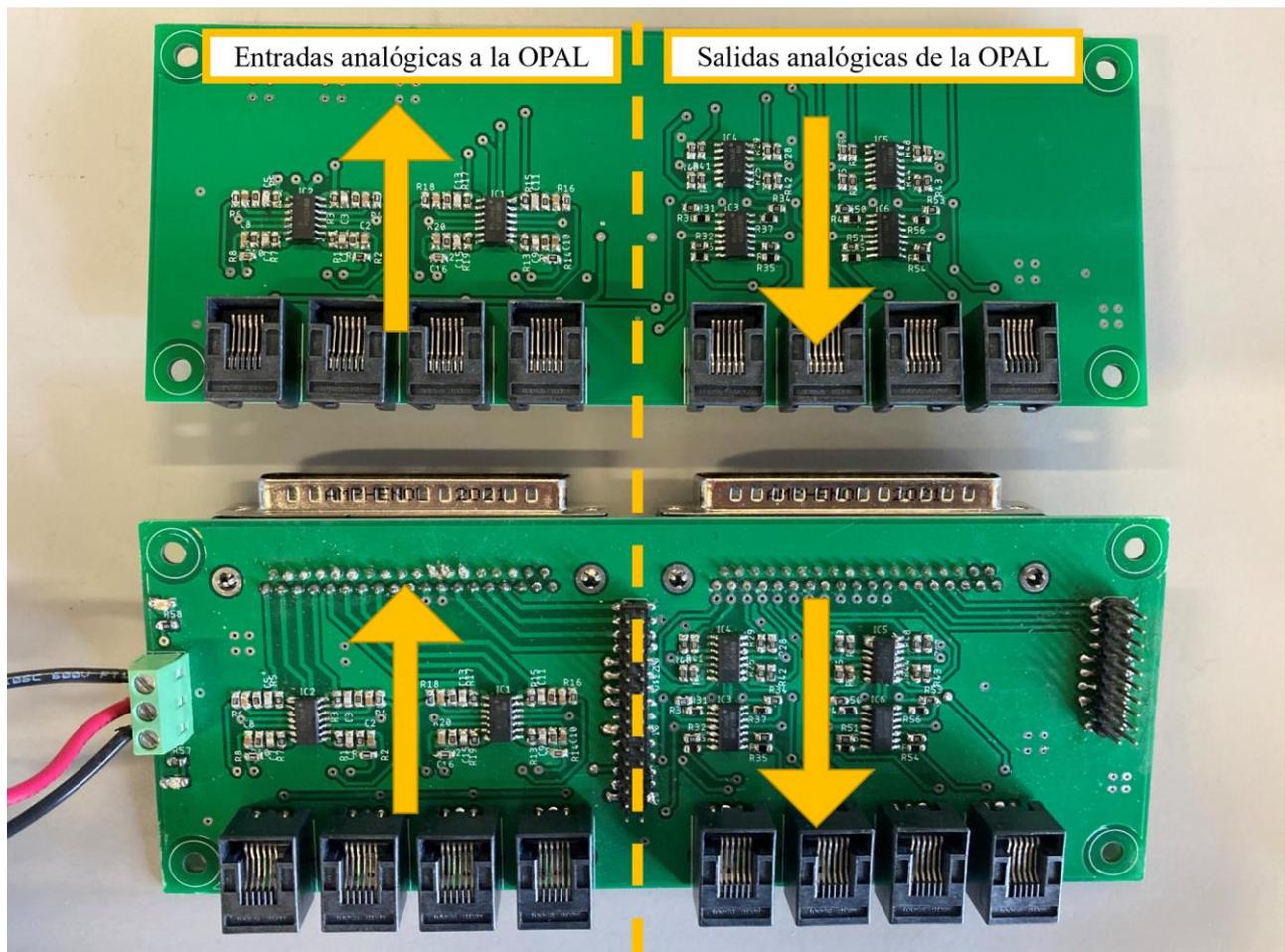


Figura 5-7. Conexión de las placas de adaptación a la OPAL

### 5.3.1 Salidas analógicas de la OPAL

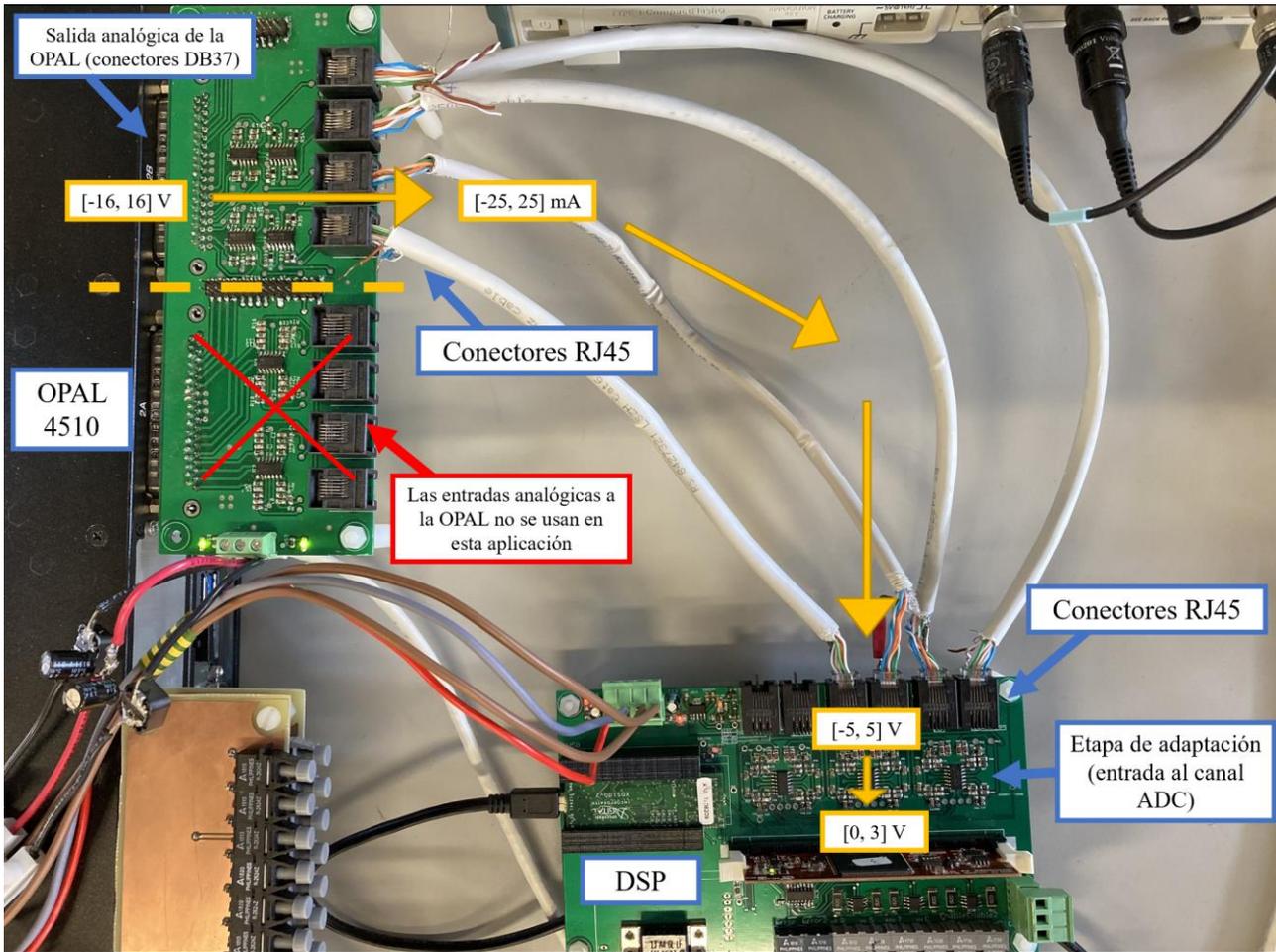


Figura 5-8. Etapas por las que pasa la señal analógica

Las salidas analógicas de la OPAL se utilizan cuando se trabaja con un sistema HIL, en el que el modelo eléctrico se simula en la plataforma de tiempo real y el control se implementa en un hardware de control externo. En este tipo de montaje, el objetivo de la plataforma es generar las mismas señales que el sistema real. Sin embargo, como el rango de salida de la OPAL es limitado, se usa la placa de adaptación para imitar las señales que se generarían en la realidad.

En un sistema real, las medidas se toman usando sensores de tensión y de corriente que, tras una etapa de adaptación, envían a la tarjeta del DSP una señal de  $\pm 25\text{mA}$ . Por tanto, el objetivo de la placa de adaptación es transformar la señal en tensión de  $\pm 16\text{V}$  en una corriente de  $\pm 25\text{mA}$ . Esto se hace a través de un circuito compuesto por un divisor resistivo a la entrada, un seguidor de tensión, y finalmente otro amplificador operacional realimentado, que transforma la tensión en corriente.

La corriente de  $\pm 25\text{mA}$  que llega a la tarjeta del DSP es transformada en  $\pm 5\text{V}$  al caer en una resistencia de  $200\ \Omega$  a la entrada de la placa. Tras una etapa de adaptación, se obtiene un rango de tensión de  $0\text{--}3\text{V}$ , que es la entrada a los canales ADC del convertidor. Este convertidor tiene una resolución de 12 bits, por lo que devuelve un número entre 0 y 4095. Idealmente, la tensión media,  $1.5\text{V}$ , se corresponde con el número central del intervalo, 2048.

En teoría, todas estas etapas de adaptación son lineales, por lo que la relación entre la tensión a la salida analógica de la OPAL y la tensión a la entrada del canal ADC también debería ser lineal. Sin

embargo, a la hora de calibrar los sensores (sección 6), se pone de manifiesto que esto no es así, y que existen no linealidades e imperfecciones que hacen que la señal se distorsione ligeramente, haciendo que sea mucho más difícil aproximar el comportamiento de estas etapas de adaptación por una recta de regresión lineal. El estudio y optimización de cada etapa de adaptación se deja como objetivo a mejorar a la finalización de este trabajo de fin de grado. Este tema se abordará de nuevo en la sección 6.

### 5.3.2 Entradas analógicas a la OPAL

Las entradas analógicas de la OPAL no se usan en sistemas HIL, porque el sistema se simula dentro de la plataforma, por lo que las señales analógicas se generan dentro y salen hacia fuera, hacia el controlador.

Sin embargo, las entradas analógicas sirven para implementar otro tipo de configuraciones, por ejemplo el Rapid Control Prototyping (RCP) que se explicó en la introducción de esta memoria. En este tipo de montaje, la plataforma de simulación en tiempo real se utiliza para implementar el control, y el sistema real se conecta a la salida de la OPAL. Por tanto, para implementar un bucle de control cerrado, el controlador necesita como entradas las magnitudes eléctricas del sistema, que entran a través de las entradas analógicas de la OPAL.

Por tanto, el papel de la placa de adaptación analógica para las entradas a la OPAL es el opuesto al que tenía para las salidas de la OPAL. En este caso, debe transformar los  $\pm 25\text{mA}$  que devuelven los sensores reales en el rango de  $\pm 20\text{V}$  a la entrada de la OPAL.

A pesar de no haberse utilizado en este proyecto, estos canales de entradas digitales también se testaron, y se comprobó su buen funcionamiento, de manera que están listos para ser utilizados en otro tipo de proyectos en el futuro.

## 5.4 Placas digitales

La funcionalidad de las placas digitales es mucho más reducida: transforman tensión en señales de fibra óptica y viceversa, ya que todas las entradas y salidas del hardware de control se hacen a través de fibra óptica.

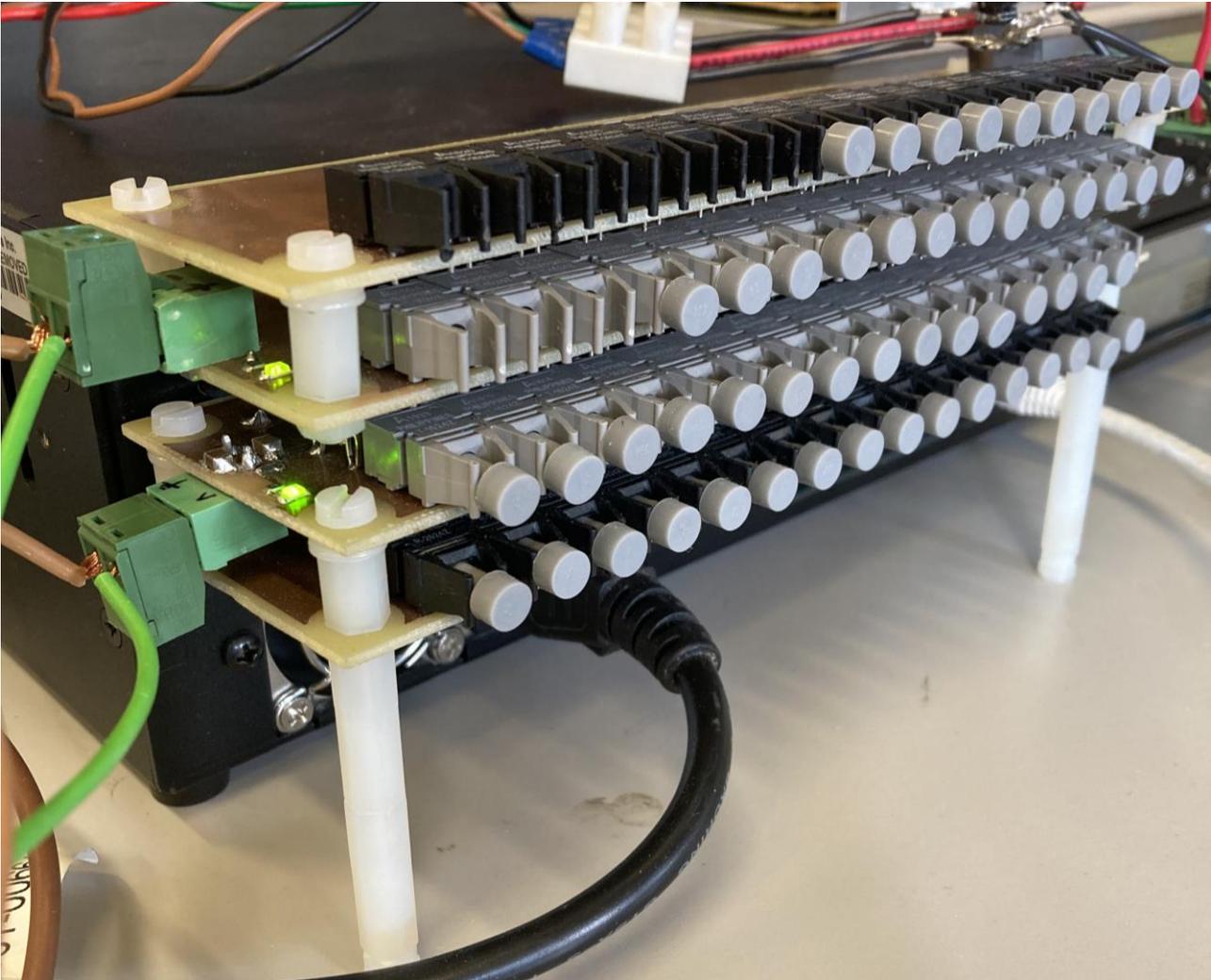


Figura 5-9. Placas digitales

En este proyecto se utilizan ambos tipos de canales, entradas y salidas digitales, las primeras para el control del convertidor de potencia implementado dentro del simulador, y las segundas como señales de error desde el convertidor hacia el controlador.

Para el correcto diseño de estas placas, se tuvieron en cuenta las características de los canales digitales:

Tabla 5–4. Características Entradas Digitales [6]

Número de canales	32
Corriente de entrada	3.6 mA
Rango de tensión	4 a 50 Vdc
Tensión máxima inversa	30V
Aislamiento	Óptico

Tabla 5–5. Características Salidas Digitales [6]

Número de canales	32
Protección Salida	Fusible 50mA
Rango Tensión Salida	5 a 30 Vdc max
Tensión Max. Inversa	30V
Aislamiento	Galvánico

#### 5.4.1 Entradas digitales a la OPAL

Las entradas digitales a la OPAL son las señales de control que genera el DSP. Es decir, las de disparo de los IGBTs que forman el convertidor de potencia. Cambiando el duty cycle de estas señales se puede variar la tensión que genera el convertidor.

En un sistema real, el DSP genera estas señales de control usando módulos PWM en el código (Code Composer), y las envía a través de transmisores de fibra óptica al driver del convertidor de potencia real. Este driver es una placa que transforma las señales de fibra óptica de los receptores en señales de tensión que hacen que los IGBTs conmuten.

La placa de adaptación de entradas digitales tiene el mismo papel que el driver del convertidor de potencia real: transforma las señales de fibra óptica de los receptores en una señal de tensión que la OPAL pueda interpretar, para que pueda asociarla a uno de los elementos switch del modelo implementado en la FPGA.

#### 5.4.2 Salidas digitales de la OPAL

Por último, en sistemas HIL, las salidas digitales de la OPAL se emplean típicamente para enviar señales de error desde el convertidor implementado en la FPGA hacia el controlador.

Esto se explica con más detalle en el apartado 4.1.1.



# 6 CONEXIÓN DEL SISTEMA Y PUESTA EN MARCHA

A la hora de poner en marcha el sistema completo, es de vital importancia comprobar que la comunicación OPAL-DSP funciona correctamente. En la sección 7 de esta memoria se hace un recorrido temporal por todos y cada uno de los modelos con los que se ha trabajado en este proyecto. De esta manera, se pueden ver las distintas etapas por las que ha pasado el proyecto, ya que cada modelo aportó nueva información sobre el funcionamiento de la plataforma.

Sin embargo, este apartado está dedicado a explicar con detalle dos etapas de la puesta en marcha especialmente relevantes. En primer lugar, cuando se conectó por primera vez la OPAL y el DSP a través del hardware de adaptación, se advirtió que existía un problema de ruido, ya que el DSP leía magnitudes muy inexactas, con un gran rango de variación, lo que podría afectar al control en etapas posteriores.

Por otro lado, una de las tareas fundamentales de la configuración y puesta en marcha del sistema es la calibración de los sensores, de manera que el controlador interprete de manera correcta las señales que le llegan al canal ADC.

Por la especial relevancia que tienen estas dos tareas a la hora de configurar y poner en marcha el sistema completo, se les ha dedicado este apartado.

## 6.1 Problemas de ruido

La plataforma de simulación en tiempo real implementa en la FPGA el circuito eléctrico a simular. A través de los sensores que se colocan en el modelo, se extraen ciertas señales, que son las magnitudes eléctricas del circuito.

La OPAL saca estas medidas por los canales analógicos de salida, que tienen un rango de  $\pm 16V$ . A continuación, estas señales son convertidas por una placa de adaptación a una corriente entre de  $\pm 25mA$ , que es el rango de salida que tendría el sensor real. Estas corrientes, a su vez, entran en la tarjeta del DSP, donde atraviesan otras etapas de adaptación, que las transforman primero a un rango de  $[-5,5] V$ , y finalmente a un rango de  $[-3,3] V$  a la entrada del canal ADC.

Para probar la adquisición de datos por parte del DSP, se generó una tensión constante en la FPGA de  $265V$ , y se sacó por el canal analógico de salida 6, que, como se explicará en el apartado 7.2.1, se conecta al IN11 de la tarjeta, canal que lee magnitudes DC. Al representar la salida del canal de conversión ADC del DSP, se observaron oscilaciones bruscas y de gran amplitud (ver figura 6-1). Estas oscilaciones eran inadmisibles y necesitaban ser atenuadas, ya que una mala medida de las magnitudes eléctricas puede hacer que el algoritmo de control en etapas posteriores no funcione.

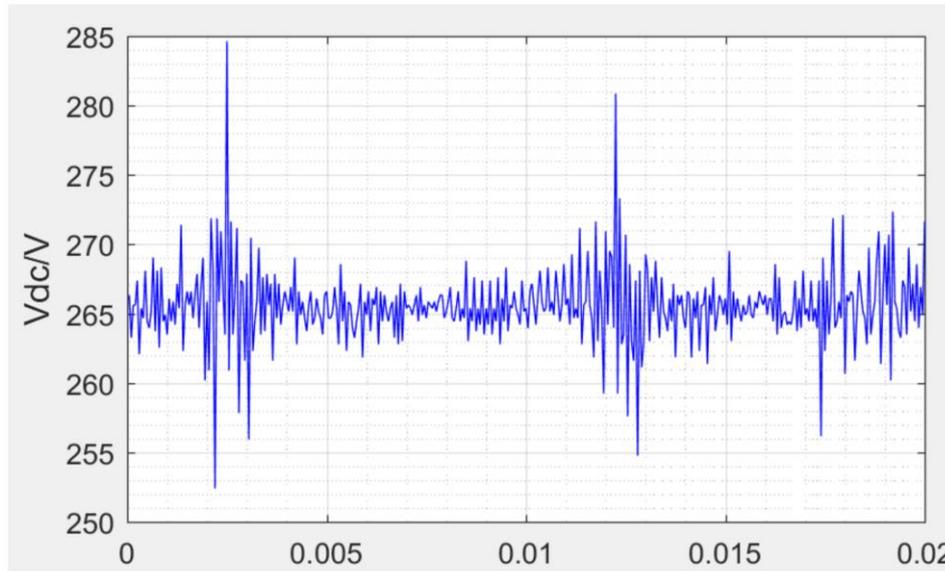


Figura 6-1. Salida del ADC previa.

Tras caracterizar el ruido, se llegó a la conclusión de que se trataba de ruido provocado por la fuente conmutada de alimentación de las placas de adaptación, que son las placas que transforman el rango [-16, 16]V de salida de la OPAL a la corriente [-25,25] mA que llega a la tarjeta del DSP.

Por ello, se optó por poner un filtro LC (figura 6-2) entre las entradas de tensión positiva y negativa de la placa de adaptación y la fuente de alimentación DC conectada a la red.

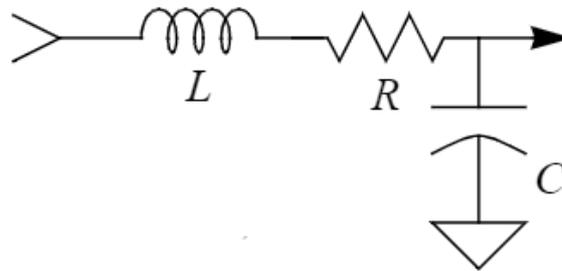


Figura 6-2. Filtro LC elegido para la implementación (para cada entrada de alimentación)

Se construyeron dos filtros distintos, con la misma estructura, pero diferentes valores de condensador, para probar cual respondía mejor. El primero  $C=100\mu\text{F}$ , y el segundo  $C=220\mu\text{F}$ . La bobina es de  $33\mu\text{H}$  en ambos casos. Finalmente se optó por utilizar el filtro de  $220\mu\text{F}$ , porque presentaba mejores características. El filtro construido se muestra en la figura 6-3.



Figura 6-3. Filtro LC implementado

Tras conectar el filtro entre la fuente de alimentación conectada a red y la alimentación de la placa de adaptación, se pudo observar una mejora drástica en el ruido de la señal. Esto se puede comprobar comparando la representación de la salida del canal ADC del DSP antes (figura 6-1) y después (figura 6-4) del filtro. En la figura 6-4, la tensión medida es 217V en vez de 265V, pero simplemente porque se cambió la tensión desde la FPGA, para probar otros puntos de funcionamiento.

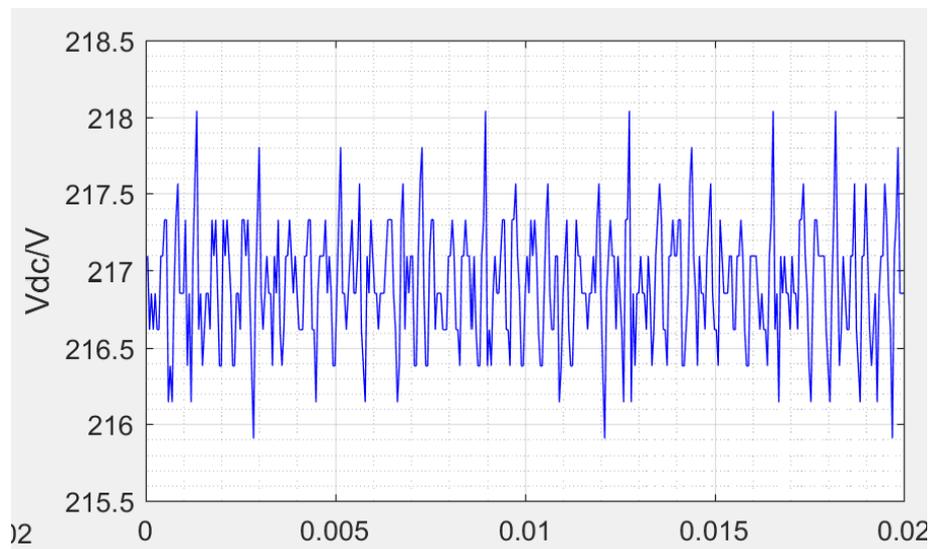


Figura 6-4. Salida del ADC tras filtrar la alimentación de la placa de adaptación y el DSP

Este efecto también se puede observar en el osciloscopio, midiendo el ruido de la alimentación que llega a la placa de adaptación antes y después del filtro (figura 6-5). Al alimentar la placa de adaptación con la alimentación filtrada, el ruido de la señal se reduce drásticamente, permitiendo un mejor control en las etapas .

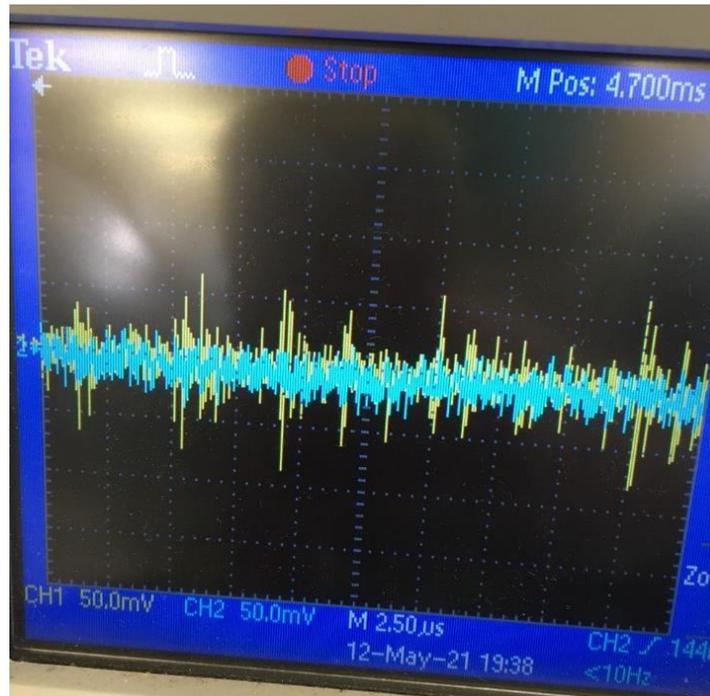


Figura 6-5. Ruido antes del filtro LC (señal amarilla) y ruido después del filtro LC (señal azul)

## 6.2 Calibración de sensores: canales ADC y DC

Como se ha explicado en la introducción de este apartado, una de las tareas más importantes a la hora de poner en marcha el sistema es calibrar los sensores para que el algoritmo de control reciba un valor coherente de la magnitud que espera recibir.

El DSP tiene 12 canales de entrada al convertidor ADC, pero en este proyecto sólo se utilizan 7: tres corrientes AC, tres tensiones AC y una tensión DC.

Para emular la toma de medidas en un convertidor de potencia real, se determina que el valor de fondo de escala de los sensores en la OPAL sea parecido al de los sensores reales que hay en el laboratorio. Esto es, para el sensor de tensión DC  $V_{max}=900V$ , para los sensores de tensión AC  $V_{max}=400V$ , y para los sensores de corriente AC  $I_{max}=35A$ .

Por tanto, la primera adaptación a la que hay que someter a las medidas que toman los sensores en la FPGA, es multiplicarlas por una ganancia que haga que todo el rango de medida del sensor se corresponda con una salida en tensión de  $+16V$ . Esto se hace internamente en la OPAL, y se configura en el bloque “Analog Output Mapping and Rescaling block Control Panel”, como se explicó en el apartado 3.4.1.2. Por ejemplo, si un sensor de corriente está midiendo  $20A$  en un punto del circuito, al multiplicarlo por  $16/35$ , se obtiene *el nivel de tensión equivalente en un rango de  $+16V$  al valor de esa corriente en un rango de  $+35A$* . Por tanto, la señal DC se multiplica por  $16/900$ , la de tensión AC por  $16/400$  y la de corriente AC por  $16/35$ .

A continuación, dicha señal atraviesa una serie de etapas de adaptación: primero, la placa de adaptación la convierte en una corriente en el rango de  $+25mA$ . Esta corriente llega a la tarjeta del DSP, en la que una resistencia de  $200\Omega$  la convierte en una tensión en el rango de  $+5V$ . Por último, una etapa de adaptación en la propia tarjeta convierte esta tensión a un rango de  $0-3V$ , que es la entrada al canal ADC.

A partir de la señal  $0-3V$  que llega a la entrada del canal ADC, el convertidor asigna un número entre  $1$  y  $2^{12}$  (convertidor ADC de 12 bits) al registro del ADC correspondiente. A este número se le aplica una ganancia y un offset, de manera que el resultado de esta operación debe ser igual (o aproximado) al valor numérico que está leyendo el sensor en el modelo ejecutándose en la FPGA (en el ejemplo,  $20A$ ).

La ganancia y el offset se calculan a través de una recta de regresión. Para calibrar los sensores, se implementa

un modelo muy sencillo en la OPAL: una fuente de tensión cuyo valor se puede modificar manualmente en tiempo real de ejecución a través de la interfaz del usuario, y un sensor que devuelve la medida de dicha tensión. Se elabora una tabla en la que se recogen dos datos: el valor de la fuente de tensión y el valor del registro ADC obtenido. Finalmente, calculamos la recta de regresión, que es la recta que mejor ajusta todos los puntos que se obtienen, y esa es la ganancia y offset que debemos aplicar al valor de salida del registro ADC.

### 6.2.1 Calibración del sensor de tensión DC

El sensor de tensión DC presenta muy buen comportamiento, por lo que fue muy fácil de calibrar, y se obtuvieron buenos resultados casi de manera inmediata. El rango de medidas de este sensor es [0, 900]V, por lo que sólo usa [0,16]V de tensión de salida de la OPAL.

Este canal ADC tiene una configuración distinta respecto de los canales AC, ya que está preparado para medir magnitudes DC y no AC. Esto hace que este canal esté mejor calibrado y funcione mejor que los canales AC, como se explicará en el siguiente apartado, 6.2.2.

Los datos recopilados para el sensor DC se representan en la tabla 6-1, y la representación gráfica de la recta de regresión es la figura 6-6.

Tabla 6–1. Calibración del sensor de tensión DC

Tensión en la OPAL	Salida del ADC (canal DC)
0	2
50	222
100	452
150	683
200	913
250	1143
300	1373
350	1603
400	1834
450	2064
500	2295
550	2525
600	2755
650	2986
700	3215
750	3445
800	3675
850	3905
900	4095

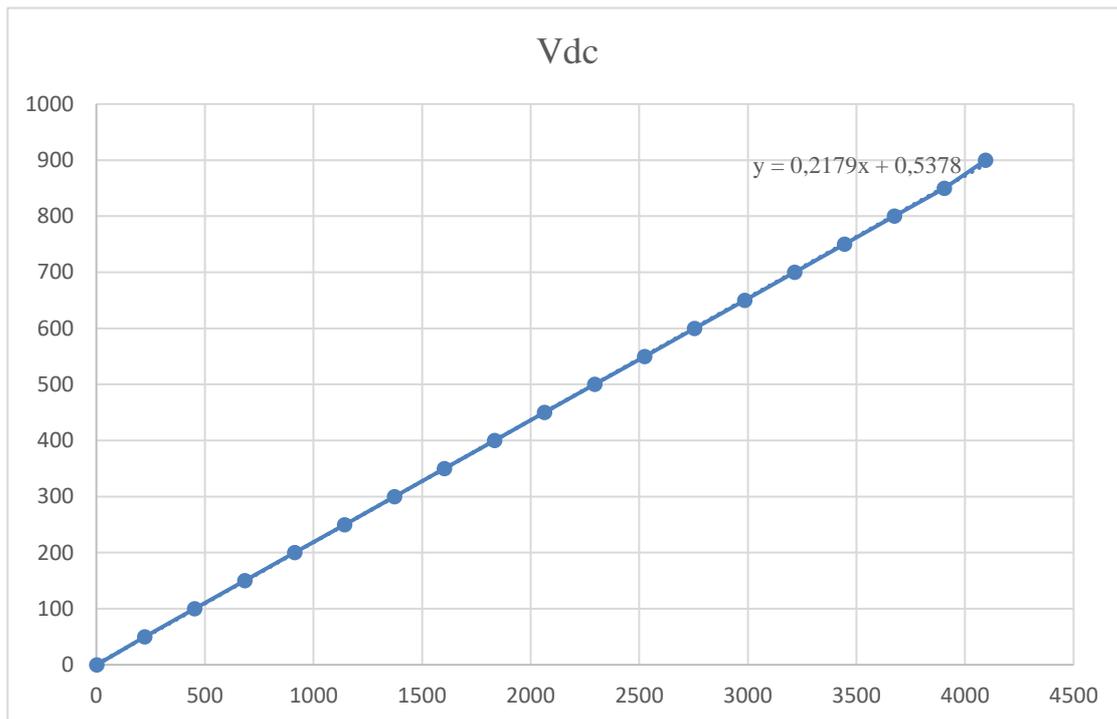


Figura 6-6. Recta de regresión del canal de tensión DC

Por último, se adjunta la implementación de esta transformación lineal en el código de code composer (figura 6-7), donde `dc[0]` es la variable en la que se guarda la salida del convertidor ADC, y `inv2L.mag.dc` es la variable a la que se le ha aplicado la ganancia y el offset, la que utiliza el algoritmo de control y la que se representa en la interfaz de Matlab. Las variables `inv2L.cal.dc[0].gain` y `inv2L.cal.dc[0].offset` son las variables que el programa utiliza para guardar los datos de calibración que se especifican desde la interfaz. En esta aplicación, fijamos esos valores a los que se obtienen de la recta de regresión y no se vuelven a modificar desde la interfaz.

```
//Funcion InitVar, que le da los valores iniciales a estas magnitudes
inv2L.cal.dc[0].gain = 0.2179;
inv2L.cal.dc[0].offset = 0.5378;

//Función: UpdateMeasures, función a la que se llama en cada rutina de interrupción
dc[0] = AdcRegs.ADCRESULT3 >> 4; //Get voltage values from ADC register
inv2L.mag.dc = (float)dc[0] * inv2L.cal.dc[0].gain + inv2L.cal.dc[0].offset;
```

Figura 6-7. Código implementado para aplicar la ganancia y el offset a la salida del ADC

## 6.2.2 Calibración del sensor de tensión AC

Los sensores de tensión AC son los que peor comportamiento presentan, y para conseguir una calibración aceptable, se ha tenido que recurrir a un polinomio de mejor ajuste de grado superior a 1, y a posteriores calibraciones manuales sobre la ganancia y el offset. Esto se debe a que en alguna de las múltiples etapas de adaptación por las que pasa la señal, existen defectos que introducen no linealidades en las medidas. Esto hace que la recta de regresión o mejor ajuste no aproxime todos los puntos de manera óptima.

Este efecto de las no linealidades se produce en todos los canales AC (corrientes y tensiones), pero es más evidente en los tres canales de tensión porque son los que mayor intervalo recorren. Los canales de tensión AC cubren un rango entre -500V y 500V, es decir, un rango total de 1000V. Los sensores de corriente en cambio, sólo miden desde -35 A a +35 A, es decir, un rango total de 70 A. Por tanto, la misma desviación en la tensión de entrada de ambos canales puede suponer una variación de cientos de voltios en el canal de tensión y solo unos pocos amperios en el canal de corriente. Por ejemplo, un incremento de 1V en la salida de los canales de corriente de la OPAL se traduce en un incremento de 2.18 A. Sin embargo, el mismo incremento de 1 V a la salida de los canales de tensión se corresponde con un incremento de 34.37 V. Esto hace que cualquier no linealidad o defecto en la señal se amplifique mucho más en los canales de tensión que en los de corriente, obteniendo un resultado peor.

Para paliar este efecto, se tomaron varias medidas. En primer lugar, se tomaron muchos más puntos que en la calibración del sensor DC o las corrientes AC. Además, pese a que teóricamente el sensor de tensión AC puede medir hasta 500V, sólo se toman datos en el rango [-400, 400]V, que es el que realmente se va a utilizar, aumentando la densidad de puntos entre [200, 400] y [-200, -400]. Esto es así porque en el modelo eléctrico que se implementa en este proyecto, nunca se obtienen tensiones AC mayores a 400V, y como se manejan tensiones senoidales, es importante que el rango intermedio se ajuste muy bien, aunque signifique perder exactitud en los valores extremos del intervalo (por encima de 400 V, que no se utiliza).

Al ajustar los puntos con una recta de regresión, la forma no lineal de la sucesión de puntos se hacía muy evidente, y se obtenían resultados muy malos en la lectura, como se observa en la figura 6-8.

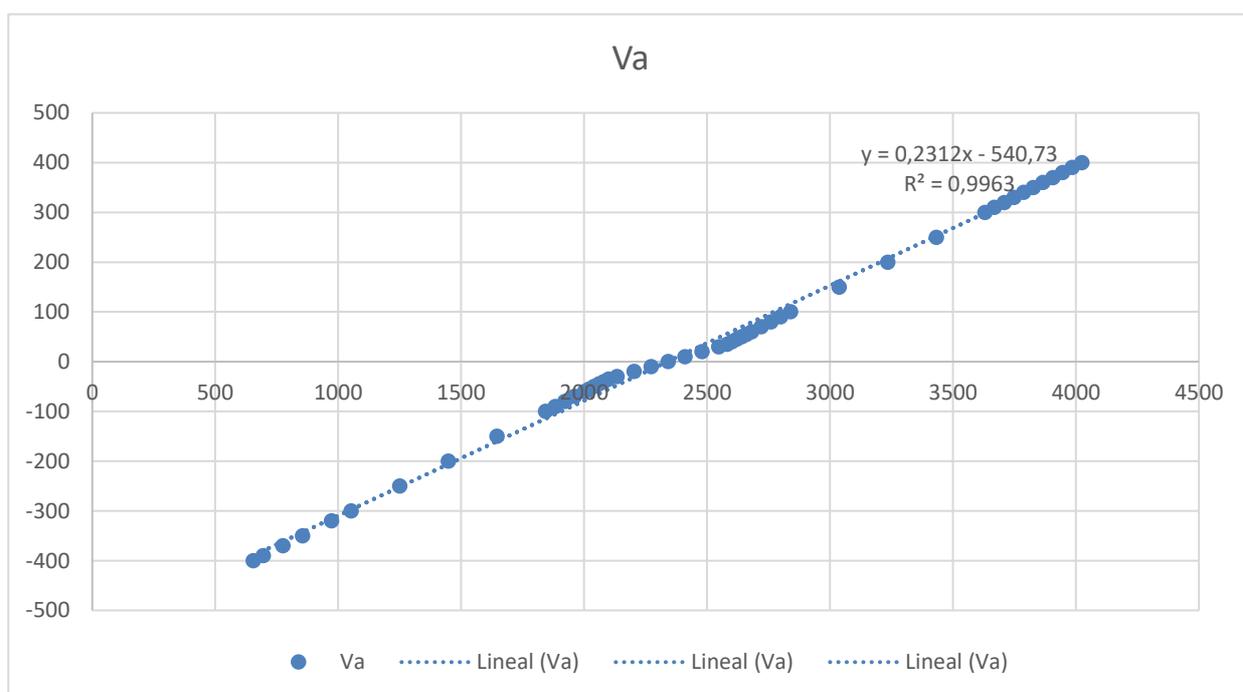


Figura 6-8. Recta de mejor ajuste.

Por eso, y aunque no es habitual, se decidió aproximar los puntos no por una recta, si no por un polinomio de mejor ajuste. Primero se probó con un polinomio de segundo orden, pero finalmente se decidió implementar un polinomio de tercer grado porque proporcionaba resultados mucho mejores.

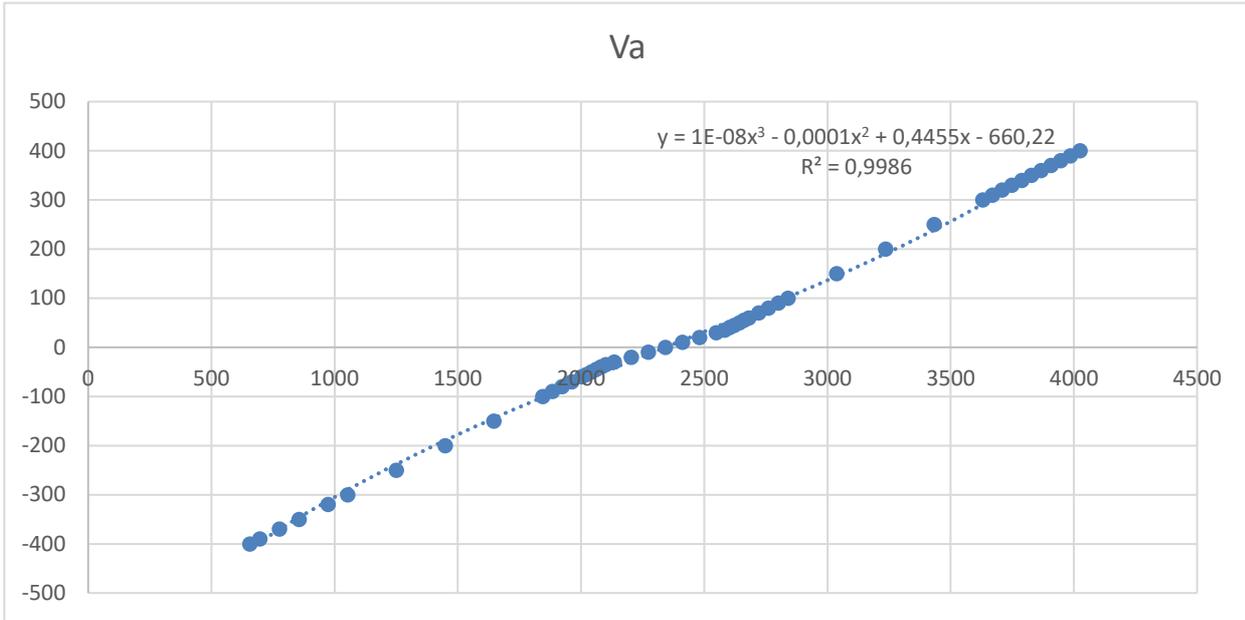


Figura 6-9. Polinomio de mejor ajuste de grado 3

Para una mayor precisión, se calculó el polinomio de mejor ajuste de grado 3 utilizando Matlab, mediante el comando “n=polyfit(x,y,3)”, lo que proporcionó unos coeficientes mucho más exactos que Excel. Este ajuste se representa en la figura 6-9.

Sin embargo, estas calibraciones todavía no proporcionaban resultados suficientemente exactos para permitir al control funcionar correctamente. Además, se comprobó empíricamente que, si al tomar las medidas en los canales AC, el canal DC se mantenía a una tensión constante, gran parte del efecto no lineal desaparecía.

Con esta premisa, se obtuvo la figura 6-10, basada en los datos de la tabla 6-2. Aunque en la figura 6-10 se representa la recta de mejor ajuste, en el código del controlador se ha implementado un polinomio de mejor ajuste de grado 5, para conseguir mejorar el ajuste en la zona del centro de la gráfica, que como se observa en la figura 6-10, sigue presentando cierto comportamiento no lineal.

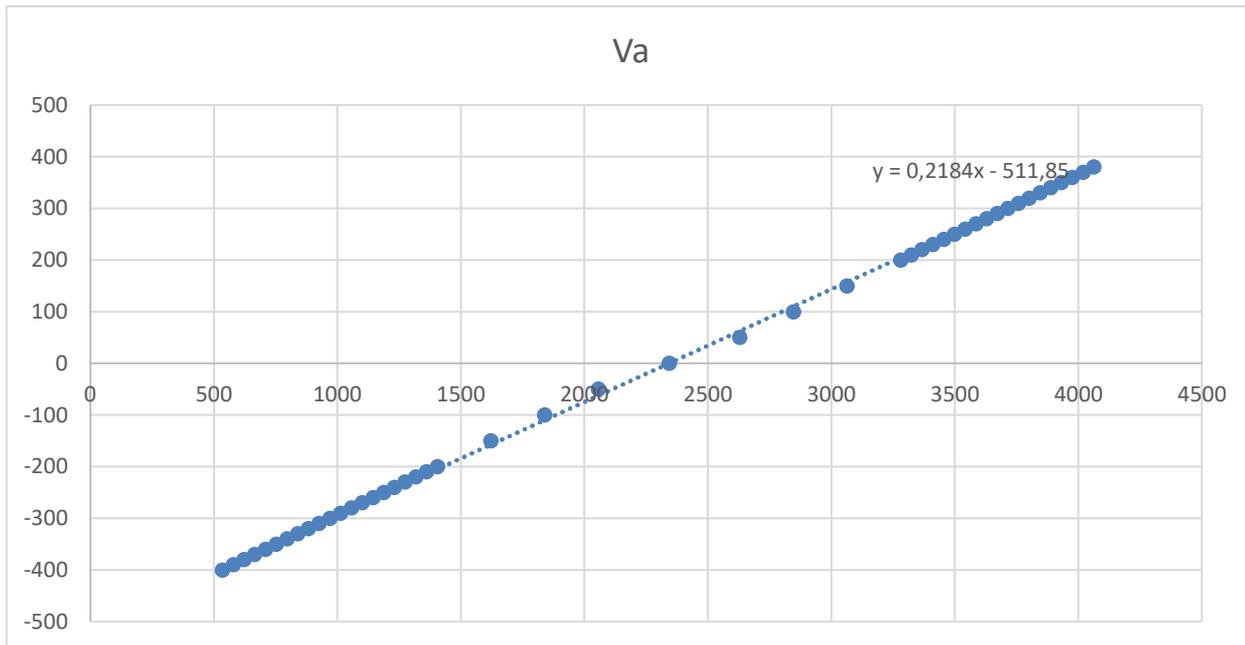


Figura 6-10. Resultados de las medidas manteniendo Vdc constante. Recta de regresión.

Tabla 6–2. Calibración de los sensores de tensión AC

Tensión en la OPAL	Salida canal ADC Vt	Salida canal ADC Vs	Salida canal ADC Vr
-400	529	535	535
-390	573	578	579
-380	616	621	622
-370	659	665	665
-360	703	708	709
-350	746	751	752
-340	789	795	796
-330	833	838	839
-320	877	881	883
-310	920	925	926
-300	965	969	970
-290	1007	1011	1013
-280	1050	1055	1057
-270	1093	1099	1100
-260	1137	1142	1144
-250	1181	1185	1187
-240	1224	1229	1231
-230	1267	1272	1274
-220	1311	1316	1318
-210	1354	1359	1361
-200	1398	1402	1405
-150	1615	1619	1622
-100	1832	1836	1839
-50	2049	2051	2057
0	2335	2339	2343
50	2620	2623	2629
100	2837	2841	2846
150	3054	3057	3063
200	3271	3275	3281
210	3314	3317	3324
220	3357	3361	3367
230	3402	3405	3411
240	3445	3448	3455
250	3489	3491	3498
260	3532	3534	3541
270	3576	4577	3585
280	3619	3621	3628
290	3662	3664	3671
300	3706	3707	3714
310	3750	3751	3758

320	3793	3794	3801
330	3837	3837	3845
340	3880	3881	3888
350	3924	3924	3931
360	3969	3968	3975
370	4011	4011	4019
380	4054	4054	4062

Incluso tras calibrar el sensor de tensión AC teniendo en cuenta todos estos factores, fue necesario añadir modificaciones manuales sobre la ganancia y el offset de la primera calibración, hasta que se obtuvo un comportamiento aceptable.

Sin embargo, a pesar de que se pueda trabajar con esta calibración y obtener resultados aceptables, estos efectos no lineales deben ser corregidos y eliminados, para que el sistema funcione mejor. Es una de las líneas de trabajo futuro que se explica con más detalle en la última sección de esta memoria, la sección 9.

### 6.2.3 Calibración del sensor de corriente AC

En los canales de corriente AC se aprecian los mismos defectos no lineales que en los canales de tensión AC, aunque su efecto es mucho menor debido al reducido rango de medida.

En este caso, y a pesar de la no linealidad, no es necesario recurrir a un polinomio de mejor ajuste, sino que al calcular la recta de mejor ajuste y debido al efecto de la reducción del rango de medida, los resultados de la conversión son aceptables.

De esta forma, se obtiene la recta de regresión representada en la figura 6-11, obtenida a partir de los datos recogidos en la tabla 6-3.

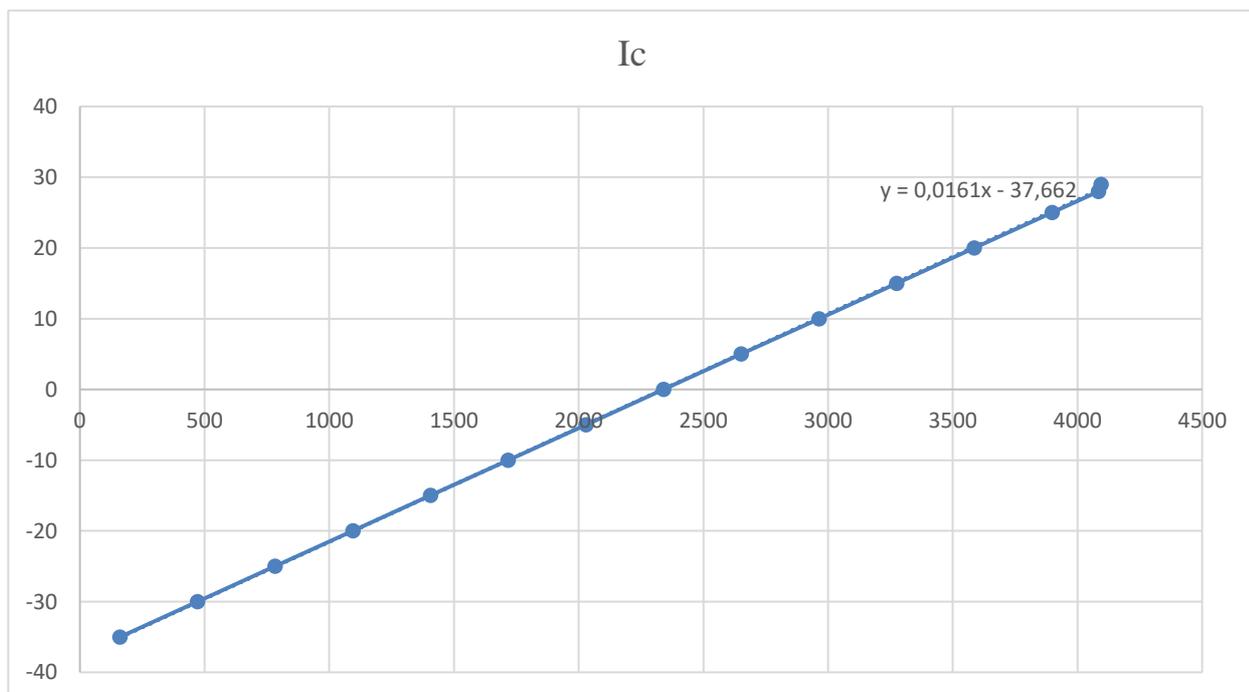


Figura 6-11. Recta de regresión de uno de los canales de corriente AC

Tabla 6–3. Calibración de los sensores de corriente AC

Tensión en la FPGA	Salida del ADC del canal It	Salida del ADC del canal Is	Salida del ADC del canal Ir
-35	2	2	2
-30	222	222	222
-25	452	452	452
-20	683	683	683
-15	913	913	913
-10	1143	1143	1143
-5	1373	1373	1373
0	1603	1603	1603
5	1834	1834	1834
10	2064	2064	2064
15	2295	2295	2295
20	2525	2525	2525
25	2755	2755	2755
28	2986	2986	2986
29	3215	3215	3215

Al igual que en la calibración del sensor DC, se adjunta la implementación de esta transformación lineal en el código de code composer(figura 6-12), donde **i[0]** es la variable en la que se guarda la salida del convertidor ADC, y **inv2L.mag.i[0]** es la variable a la que se le ha aplicado la ganancia y el offset. Las variables **inv2L.cal.i[0].gain** y **inv2L.cal.i[0].offset** son las variables que el programa utiliza para guardar los datos de calibración que se especifican desde la interfaz. En esta aplicación, fijamos esos valores a los que se obtienen de la recta de regresión y no se vuelven a modificar desde la interfaz.

```
//Funcion InitVar, que le da los valores iniciales a los parámetros de calibración
inv2L.cal.i[0].gain = 0.0162;
inv2L.cal.i[0].offset = -37.704;
inv2L.cal.i[1].gain = 0.0162;
inv2L.cal.i[1].offset = -37.697;
inv2L.cal.i[2].gain = 0.0161;
inv2L.cal.i[2].offset = -37.662;

//Función: UpdateMeasures, función a la que se llama en cada rutina de interrupción
i[0] = AdcRegs.ADCRESULT0 >> 4; //Get current values from ADC register
i[1] = AdcRegs.ADCRESULT1 >> 4;
i[2] = AdcRegs.ADCRESULT2 >> 4;
inv2L.mag.i[0] = (i[0] * inv2L.cal.i[0].gain + inv2L.cal.i[0].offset);
inv2L.mag.i[1] = (i[1] * inv2L.cal.i[1].gain + inv2L.cal.i[1].offset);
inv2L.mag.i[2] = (i[2] * inv2L.cal.i[2].gain + inv2L.cal.i[2].offset);
```

Figura 6-12. Código implementado para aplicar la ganancia y el offset a la salida del ADC



# 7 MODELOS IMPLEMENTADOS

El objetivo final de este proyecto es, una vez puesto en marcha el sistema de simulación HIL, probar algoritmos de control MPPT sobre placas solares.

Para llegar al sistema final, se han ido construyendo distintos modelos, aumentando el grado de complejidad en el control, descubriendo así las bondades pero también las limitaciones del software RT-Lab y de la plataforma OPAL, como comentamos anteriormente. El conocimiento adquirido es acumulativo, y la solución a problemas que se presentaban en modelos más simples se aplicaron luego a los modelos más complejos.

En orden cronológico, estos han sido los distintos modelos que se han probado:

- Inversor en modulación abierta
- Inversor conectado a carga RL con control de corrientes
- Inversor conectado a red con circuito de pre carga y control de la tensión del condensador Dc-link
- Inversor conectado a red y a placas solares con algoritmo de control MPPT

Cabe destacar que todos los algoritmos de control, excepto el algoritmo MPPT, al que se le dedicará el siguiente apartado entero, fueron proporcionados por el tutor de este proyecto de investigación, a fin de agilizar su implementación en el sistema HIL, que es el verdadero objetivo de este proyecto.

## 7.1 Inversor en modulación abierta

El primer modelo que se probó se representa en la figura 7-11. El modelo consiste en un inversor en abierto conectado a una fuente DC. El control implementado en code composer es una modulación en bucle abierto. No hay realimentación de ningún tipo. En esta primera etapa, el objetivo era comprobar que la comunicación desde el DSP hacia el modelo implementado en la FPGA era correcta.

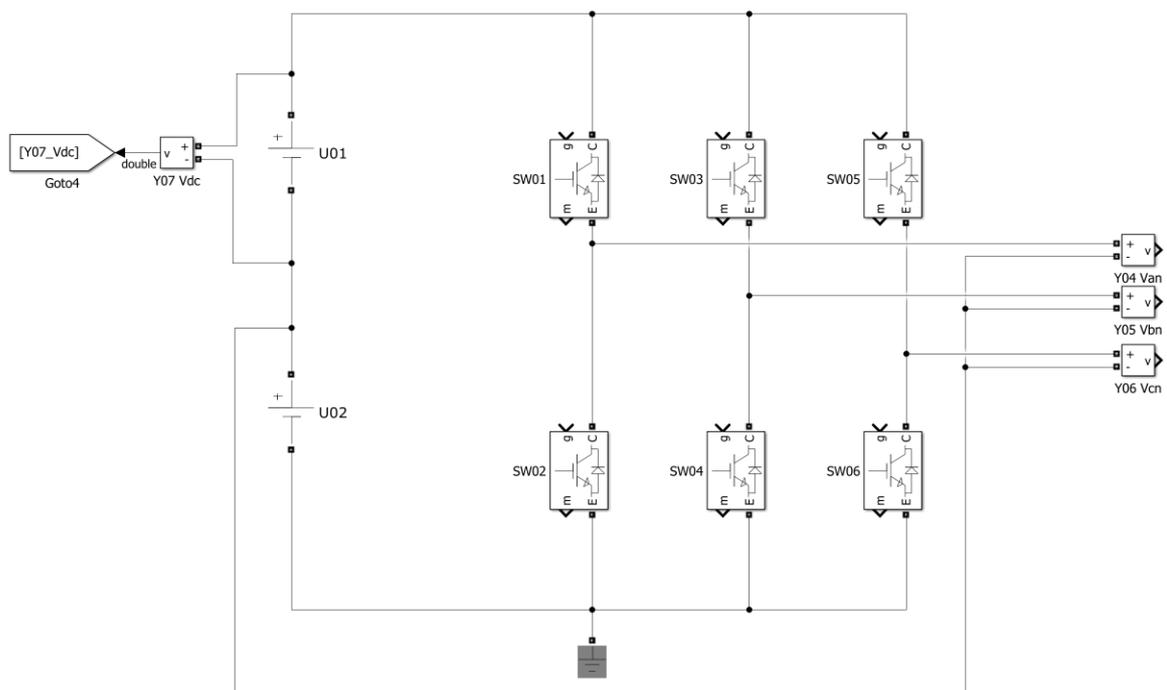


Figura 7-1. Inversor implementado

Este modelo también sirvió para hacer distintas pruebas sobre el tiempo de muestreo del modelo implementado en la FPGA de la OPAL. Es decir, aunque se conoce el tiempo de muestreo del modelo teóricamente, se quiso comprobar el duty cycle mínimo que el modelo de la OPAL es capaz de adquirir. Para ello, a una frecuencia de disparo de 10 kHz, se fue modificando manualmente el duty cycle de la modulación, y, midiendo la salida analógica de tensión modulada con el osciloscopio (figura 7-2), se comprobó experimentalmente cuál era el pulso mínimo capaz de leer la OPAL. Los resultados obtenidos fueron que reconocía pulsos con un duty cycle desde 0.03 hasta 0.98.

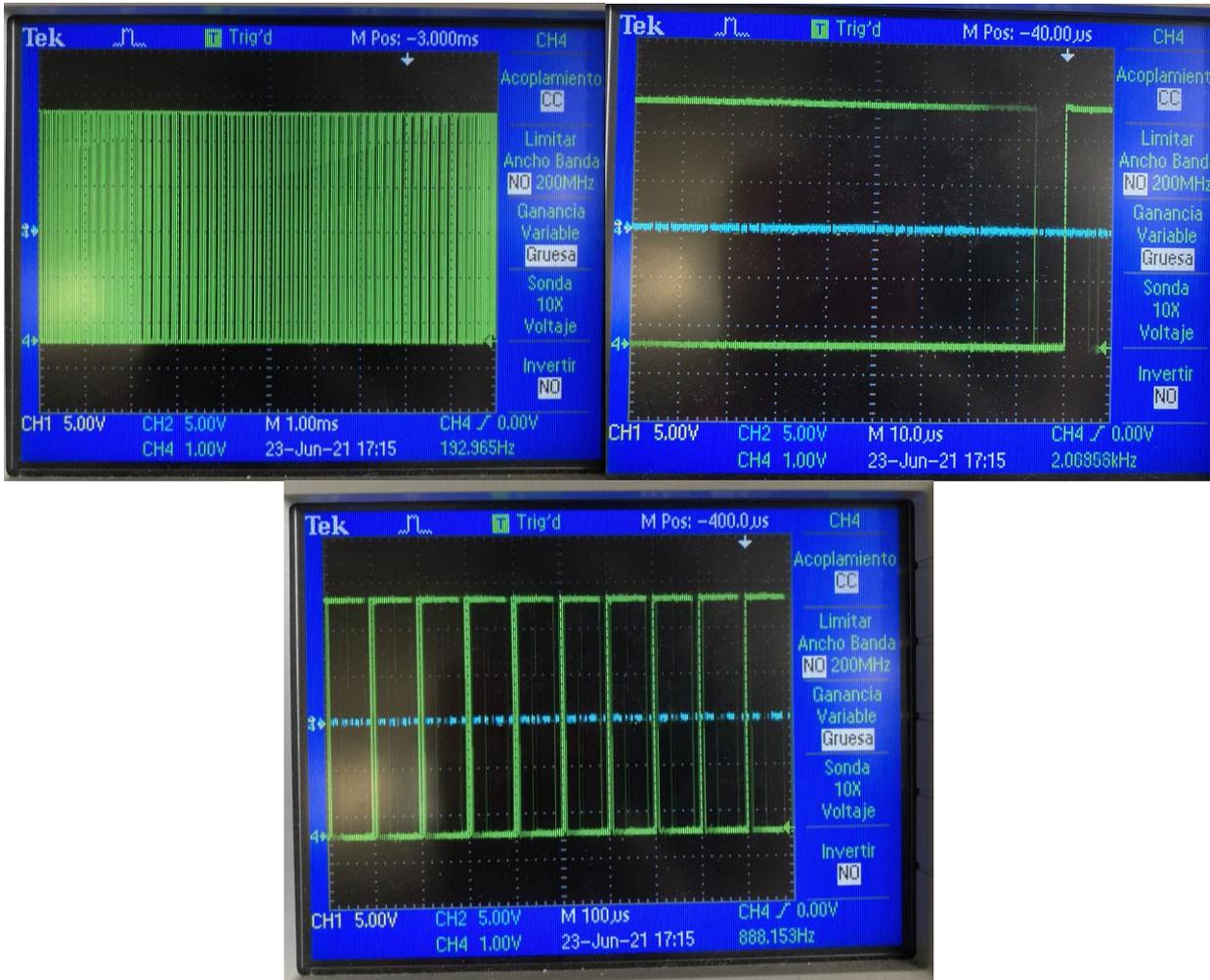


Figura 7-2. Tensión a la salida del inversor modulada, medida con un osciloscopio

## 7.2 Inversor conectado a carga RL con control de corrientes

El objetivo de este segundo modelo era comprobar que la comunicación desde la OPAL hacia el DSP también funciona bien. Esto es algo más laborioso que la comprobación de los pulsos digitales, y comprende varias pruebas.

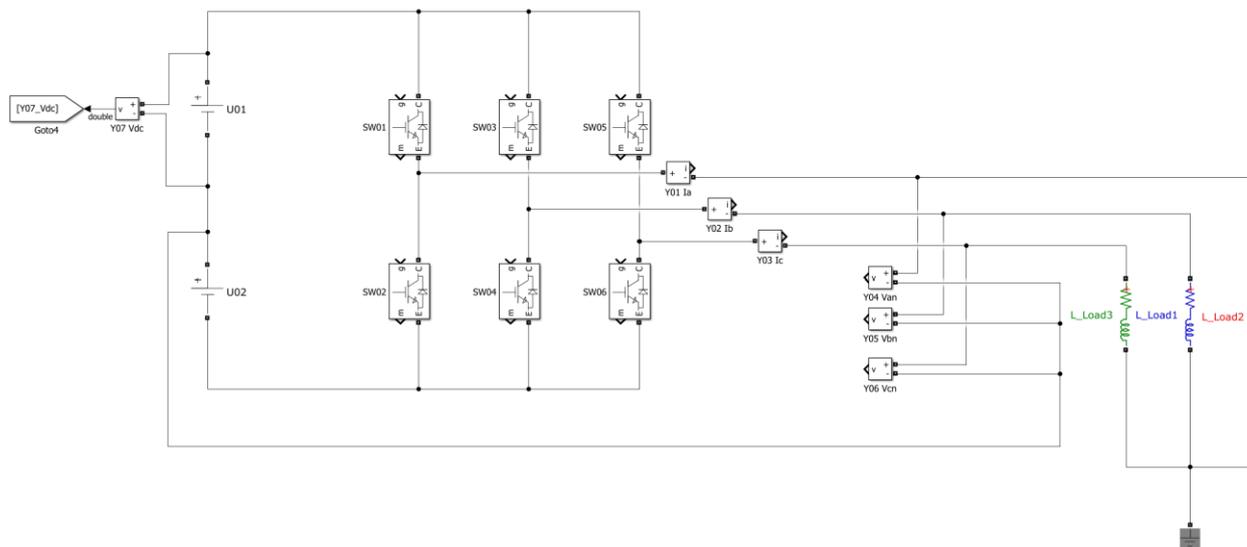


Figura 7-3. Inversor conectado a carga RL con control de corrientes

### 7.2.1 Salidas digitales de la OPAL

Primero, comprobar que la plataforma OPAL saca la medida que se quiere obtener a través del canal analógico especificado, multiplicada por la ganancia especificada. Esto se controla a través del bloque “Analog Output Mapping and Rescaling block Control Panel” como se comentó en la sección OPAL 4510.

En concreto, a la hora de conectar las salidas de la OPAL con las entradas ADC del DSP, es necesario que cada medida entre por el canal preparado para leer esa magnitud concreta, según esté codificado en código del controlador. En este caso, los canales ADC se corresponden con las siguientes magnitudes:

Tabla 7–1. Asignación canal de entrada-magnitud leída del DSP

Entradas al DSP	Magnitud que lee
IN 5	Vt
IN 6	It
IN 7	Vs
IN 8	Is
IN 9	Vr
IN 10	Ir
IN 11	Vdc

La conexión a través de cables con conectores RJ12 se realiza como se muestra en la figura 7-4.

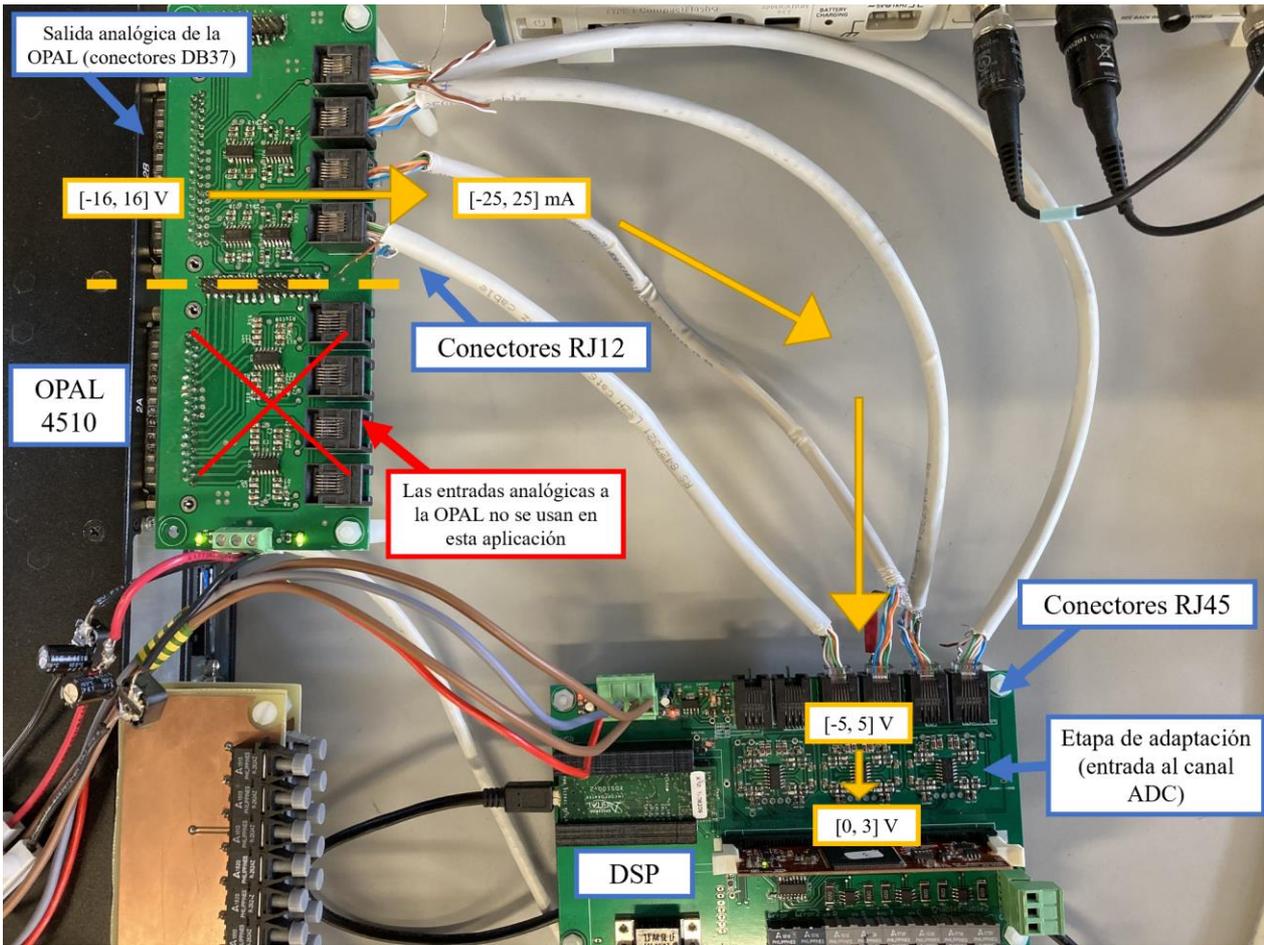


Figura 7-4. Conexión a través de cables RJ12 de la salida de las placas

Es necesario que la medida que sale por el canal analógico de la OPAL se corresponda con la magnitud que está preparado para leer ese canal ADC. Por tanto, se puede especificar otra tabla, donde la última columna se configura con el bloque “Analog Output Mapping and Rescaling block Control Panel”. Es decir, se asocia a cada canal de salida analógico la medida que ese canal del DSP está preparado para medir.

Tabla 7-2. Asignación canal de entrada DSP-canal de salida analógico

Entradas al DSP	Magnitud que lee	Canal por el que sale la medida
IN 5	Vt	Ch0
IN 6	It	Ch1
IN 7	Vs	Ch2
IN 8	Is	Ch3
IN 9	Vr	Ch4
IN 10	Ir	Ch5
IN 11	Vdc	Ch6

A partir de esta correspondencia, da igual el orden de los sensores dentro del modelo (Y01, Y02... etc) siempre que en el bloque “Analog Output Mapping” se asocien correctamente: por ejemplo, el sensor YXX, que mide la tensión Vdc del modelo, debe asociarse al canal de salida 6.

## 7.2.2 Funcionamiento de la placa de adaptación

Una vez que se ha comprobado que cada medida se lee por el canal correcto, comprobamos que la placa de adaptación cumple su cometido correctamente y transforma una tensión en el rango  $\pm 16\text{V}$  a una corriente de  $\pm 25\text{ mA}$ . Es en esta etapa donde se detectó por primera vez un problema de ruido en la señal. Tras varias pruebas, se llegó a la conclusión de que el ruido no se debía al propio diseño del circuito, sino a la alimentación que recibía (aunque se estudiaron algunas técnicas de diseño para mejorar el ruido en circuitos de placas impresas, y con la intención de implementarse en un nuevo diseño más adelante). Como solución, se construyó un filtro LC que se colocó entre la salida de la fuente de alimentación DC (conectada a la red) y la alimentación de la placa de adaptación. La estructura del filtro, pruebas y resultados se explican con detalle en el apartado 6.1.

Una vez corregido el ruido, se pudo comprobar que la salida en corriente de esta placa ( $\pm 25\text{mA}$ ), se convertía satisfactoriamente en un rango de tensión de  $\pm 5\text{V}$  al caer en una resistencia de  $200\ \Omega$  en la entrada de la tarjeta del DSP.

## 7.2.3 Funcionamiento de la etapa de adaptación del DSP

La tarjeta del DSP, a su vez, convierte el rango  $\pm 5\text{V}$  que tiene en su resistencia de entrada de  $200\ \Omega$  a un rango de  $\pm 3\text{V}$ , que es, finalmente, la entrada al canal ADC del micro.

## 7.2.4 Lectura del canal analógico y representación por la interfaz de Matlab

Tras comprobar que la tensión que sale de la OPAL llega correctamente a la entrada del canal ADC, el último paso es calibrar los sensores. Es decir, obtener la ganancia y el offset que hay que aplicar al número de salida del convertidor ADC para que el algoritmo de control reciba el valor de la magnitud que se está leyendo en el modelo eléctrico implementado en la OPAL. Este proceso de calibración se explica en el apartado 6.2.

## 7.2.5 Conclusiones

Una vez que comprobados todos los pasos anteriores (junto con el primer modelo), se confirma que funciona la comunicación FPGA-DSP: el DSP lee correctamente el valor del sensor emulado en la OPAL y los IGBTs de la OPAL leen correctamente las señales de disparo generadas por el DSP. Por tanto, se procedió a implementar el primer algoritmo de control en bucle cerrado, un control de corrientes sobre un inversor con una carga.

## 7.3 Inversor conectado a una fuente DC y a la red con control de corrientes

En los siguientes modelos se prueban los lazos internos al control completo que se implementará para los paneles PV.

En primer lugar se prueba el lazo interno de corrientes: a partir de una potencia de referencia, el controlador genera los disparos que imponen las corrientes necesarias para alcanzar esa referencia de potencia. La tensión del dc-link está fijada por una fuente dc, de manera que no hay control sobre la tensión del condensador, solamente sobre las corrientes que salen y entran del convertidor. El modelo que se implementa en la FPGA es el de la figura 7-5.

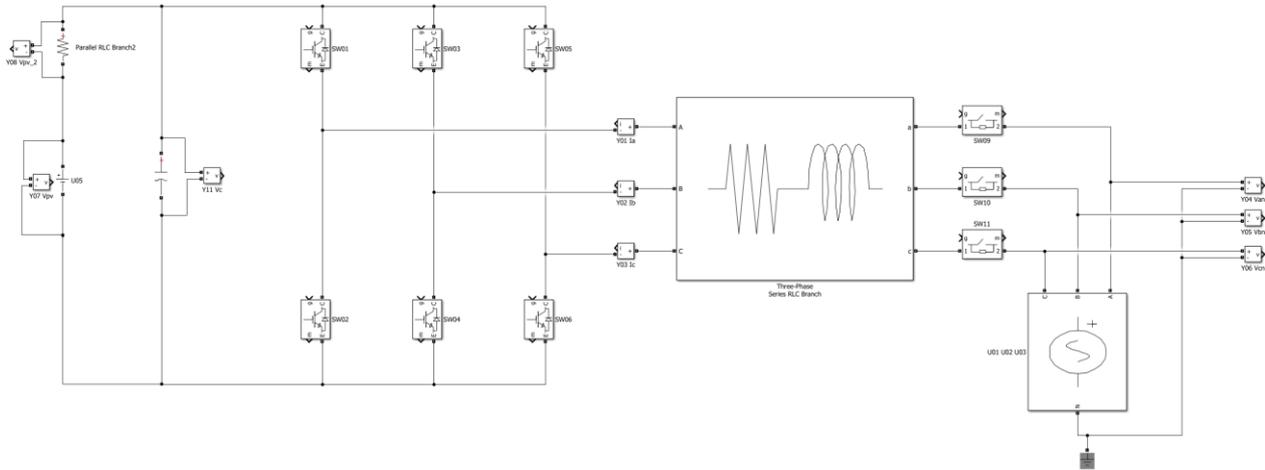


Figura 7-5. Modelo eléctrico implementado

Con este modelo se intentó implementar la etapa de precarga del condensador, pero por los motivos expuestos anteriormente no fue posible. Por eso, se optó por darle un valor inicial de tensión al condensador.

Para ver que el modelo está funcionando correctamente, se comprueba la medida de ciertas magnitudes eléctricas en tres puntos distintos: interfaz de la OPAL, interfaz del DSP y osciloscopio.

### 7.3.1 Representación de magnitudes eléctricas

#### 7.3.1.1 Tensiones de red

Se generan en la CPU, que se las pasa a la FPGA a través del bloque eHS. Si se representan en el scope de la interfaz gráfica se obtiene la figura 7-6.

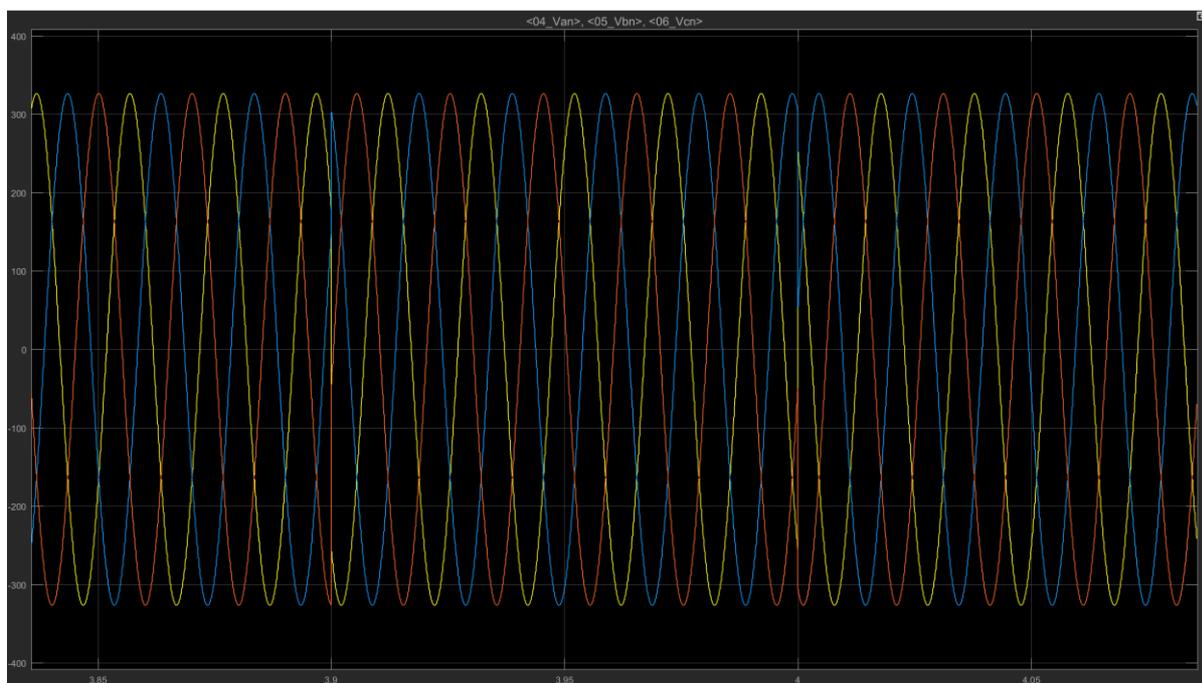


Figura 7-6. Tensiones de la red en la consola (subsistema GUI)

También se pueden medir a la salida de la OPAL con el osciloscopio (figura 7-7). La ganancia que se les aplica a las tensiones AC es de 16/500. Como el pico de tensión está en 325V aproximadamente, a la salida de la OPAL se deberían obtener aproximadamente 10V, como se puede comprobar en la 7-7.

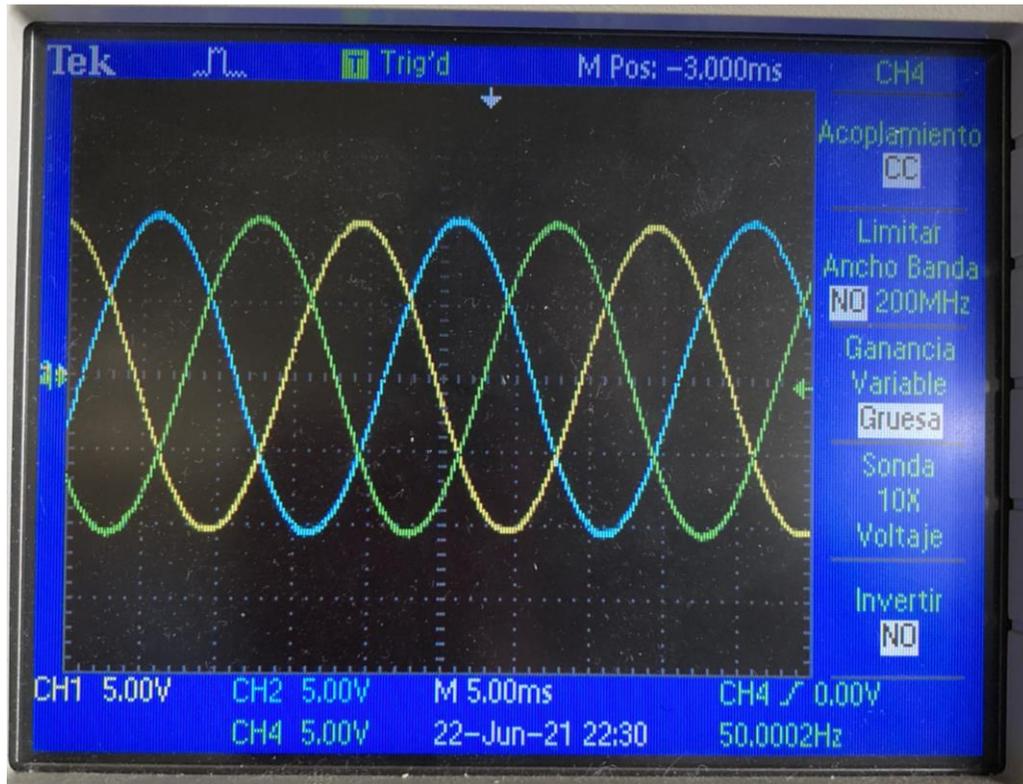


Figura 7-7. Tensiones de la red en el osciloscopio (salida OPAL)

Por último, para comprobar que la medida está llegando bien al DSP, y que la calibración de los canales es correcta, se representan las tensiones de red en la interfaz gráfica del DSP (figura 7-8).

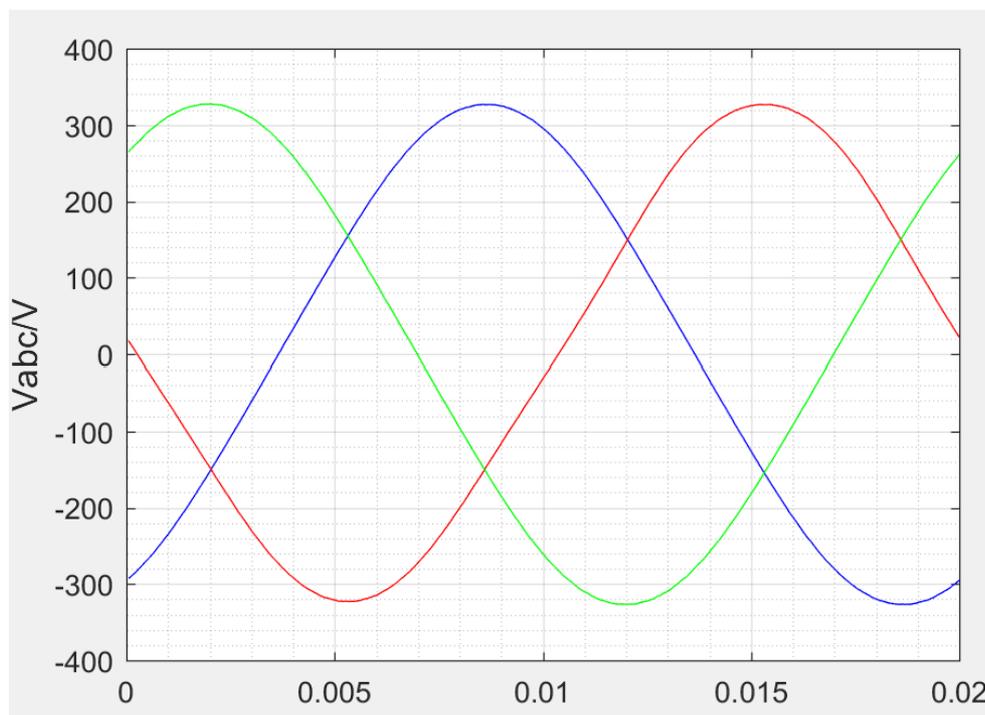


Figura 7-8. Tensiones de la red en la interfaz Matlab del DSP

De esta manera, se comprueba que las tensiones de red que se generan en la FPGA llegan correctamente al DSP, por lo que el algoritmo de control puede aplicar el PLL correctamente.

### 7.3.1.2 Tensión del dc-link

Se puede hacer la misma comprobación para la tensión del dc-link. En este modelo, la tensión del condensador no es objeto de control ni es variable, sino que la fija la fuente de tensión DC. Por eso, debería ser un número próximo a 750, que es la tensión de la fuente.

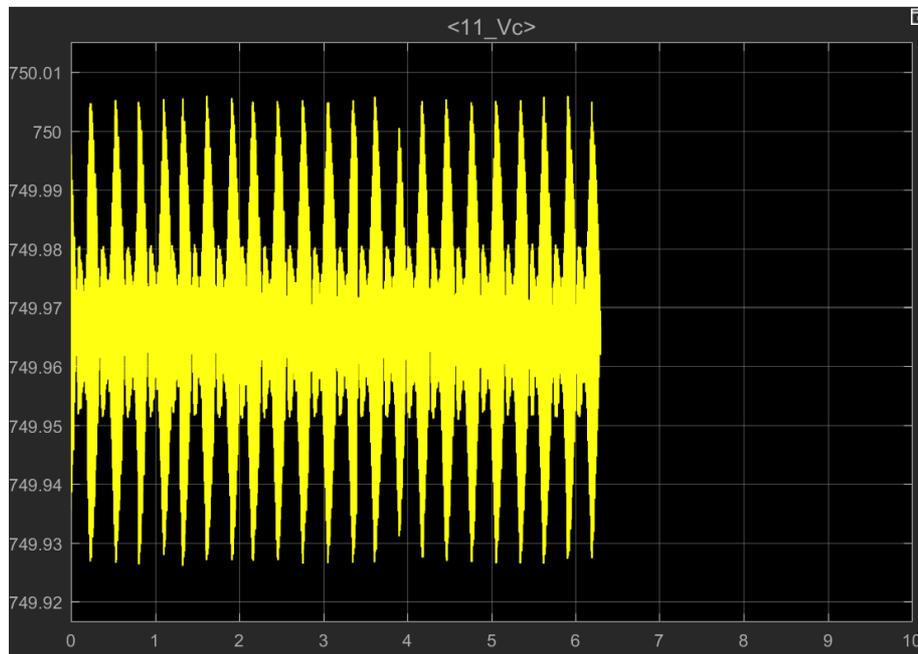


Figura 7-9. Tensión del dc-link en la consola (subsistema GUI)

A la tensión DC se le aplica una ganancia de 16/900. Por tanto, los 750V se transforman en 13 V a la salida de la OPAL, como se puede observar en la figura 7-10.

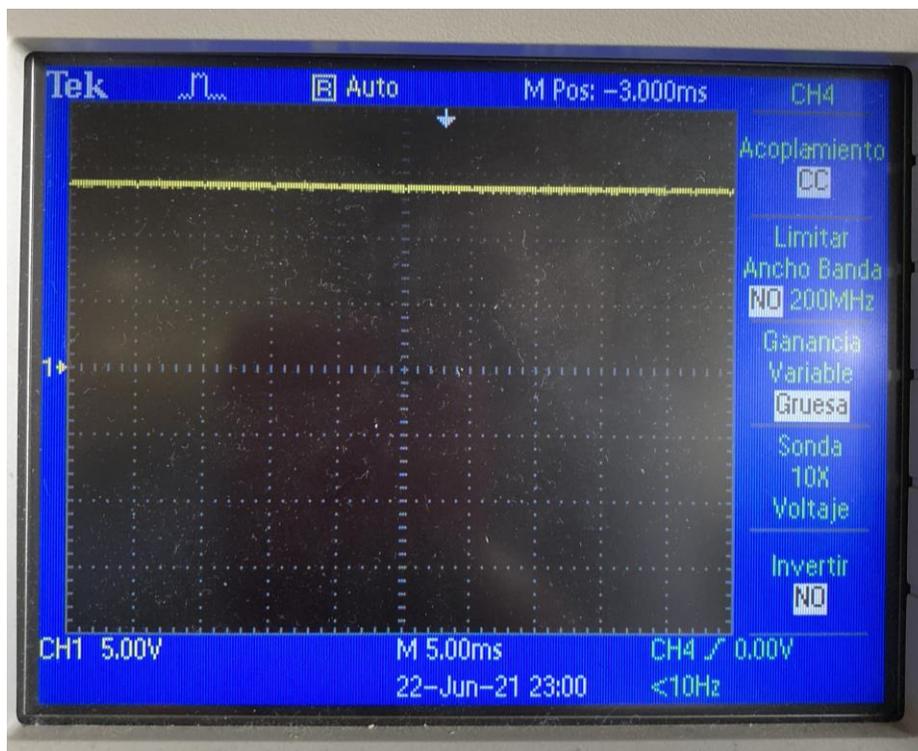


Figura 7-10. Tensión del dc-link en el osciloscopio (salida OPAL)

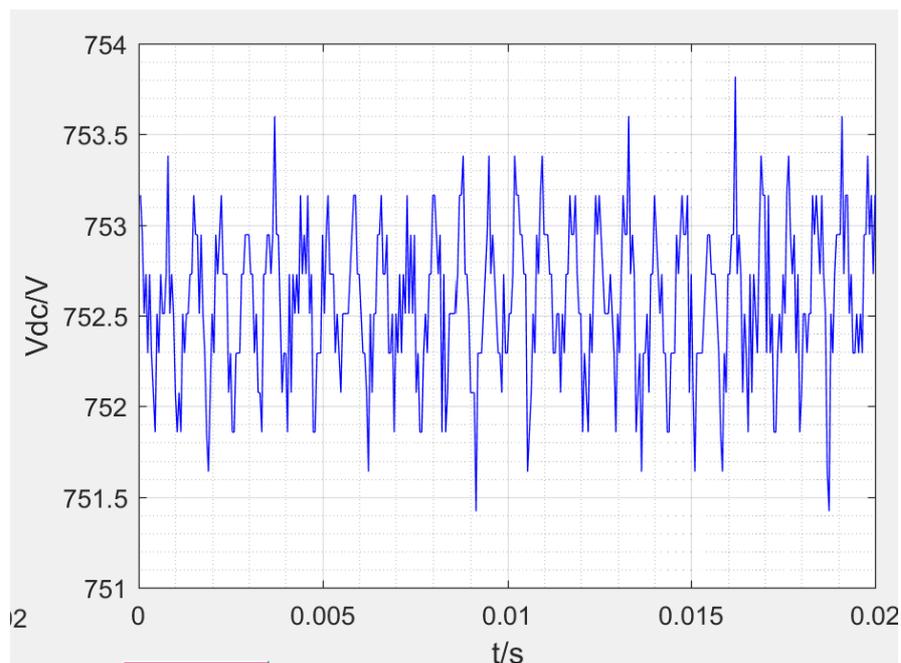


Figura 7-11. Tensión del dc-link en la interfaz Matlab del DSP

Se comprueba que la medida se repite en las tres representaciones gráficas, figuras 7-9, 7-10 y 7-11, lo que significa que la tensión del condensador de 750V en la FPGA se lee correctamente en el DSP.

### 7.3.1.3 Corrientes

Las corrientes son el objeto de control de este lazo interno, y varían para inyectar o extraer potencia del dc-link según la referencia. La fuente DC absorbe o cede potencia, en una cantidad definida por estas corrientes, siempre a una tensión fija de 750V. Estos cambios en las corrientes también se pueden observar en la ejecución en tiempo real del sistema.

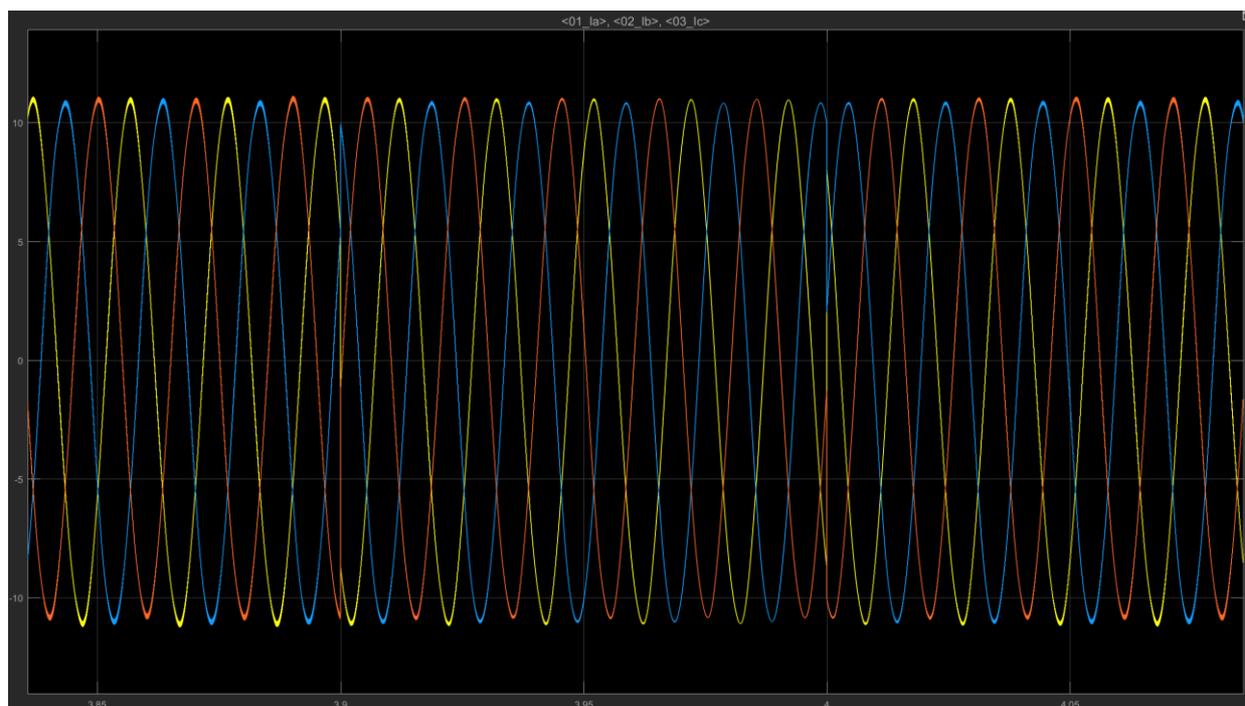


Figura 7-12. Tensión del dc-link en la consola (subsistema GUI)

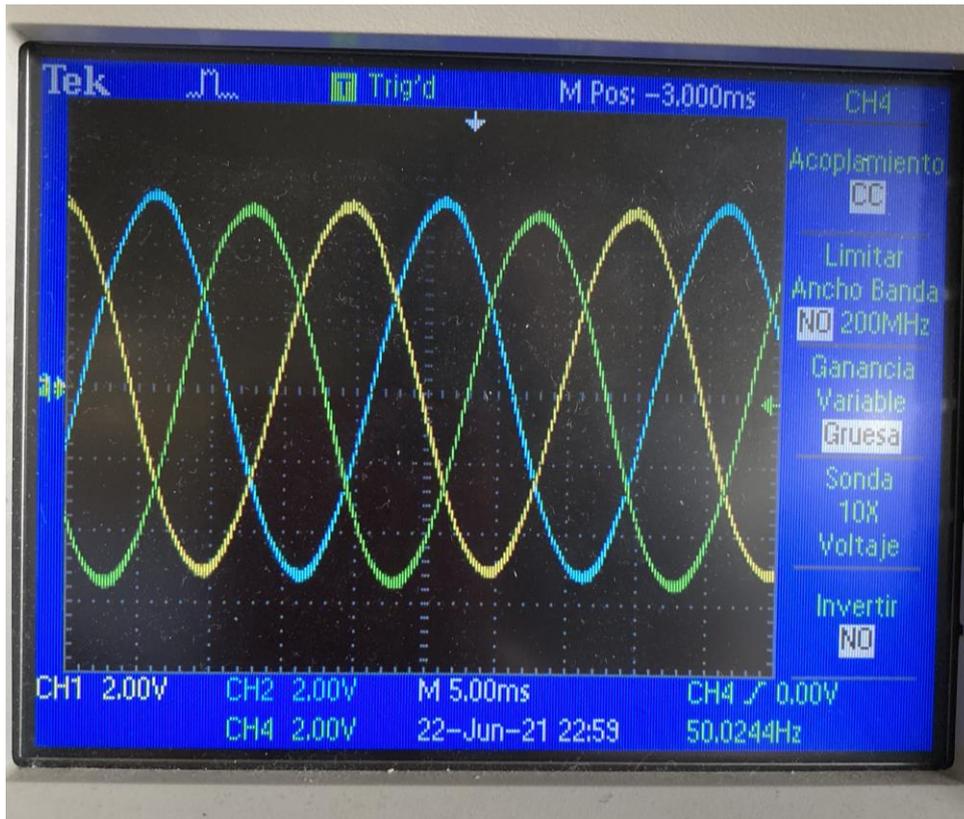


Figura 7-13. Tensión del dc-link en el osciloscopio (salida OPAL)

Para una referencia de potencia activa de 5000W, se obtienen unas corrientes de referencia de 12.8A aproximadamente, y las corrientes reales del sistema siguen a esta referencia, como se puede observar en las figuras 7-12 y 7-14. A la salida de la OPAL se obtienen unas corrientes de 5.8A, ya que están afectadas por la ganancia 16/35 (figura 7-13).

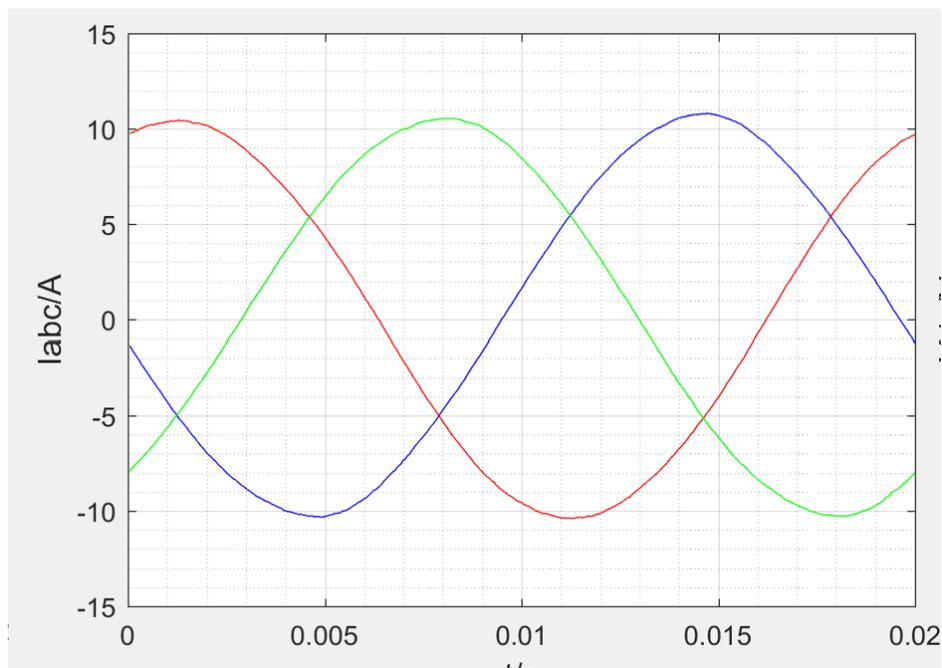


Figura 7-14. Tensión del dc-link en la interfaz Matlab del DSP

A pesar de que las corrientes deberían estar en torno a los 12.8 A, se puede apreciar una pequeña inexactitud en la medida, ya que no alcanzan los 12A (figura 7-14). Esto se debe a la calibración del sensor, que debido a las etapas intermedias no lineales que existen en los canales AC, no es todo lo buena que debería ser, y esto puede afectar al control. Sin embargo, se acepta este margen de error como válido, y se comprueba que el control funciona correctamente. Como se indicó en la sección 6.2.2 se trata de un defecto que queda pendiente de corregir en trabajos futuros sobre este sistema de simulación HIL.

### 7.3.2 Comprobación del funcionamiento del controlador

Para demostrar que el control funciona correctamente, se adjuntan una serie de figuras, en las que se representa el estado del sistema desde la interfaz del DSP. También se adjuntan capturas de las variables que muestra Code Composer en tiempo real.

El esquema de control es el siguiente: a partir de una potencia de referencia, el controlador genera los disparos que imponen las corrientes necesarias para alcanzar esa referencia de potencia. La tensión del dc-link está fijada por una fuente dc, de manera que no hay control sobre la tensión del condensador, solamente sobre las corrientes que salen y entran del convertidor. El lazo de control de las corrientes implementado en este modelo sigue el esquema de la figura 7-15.

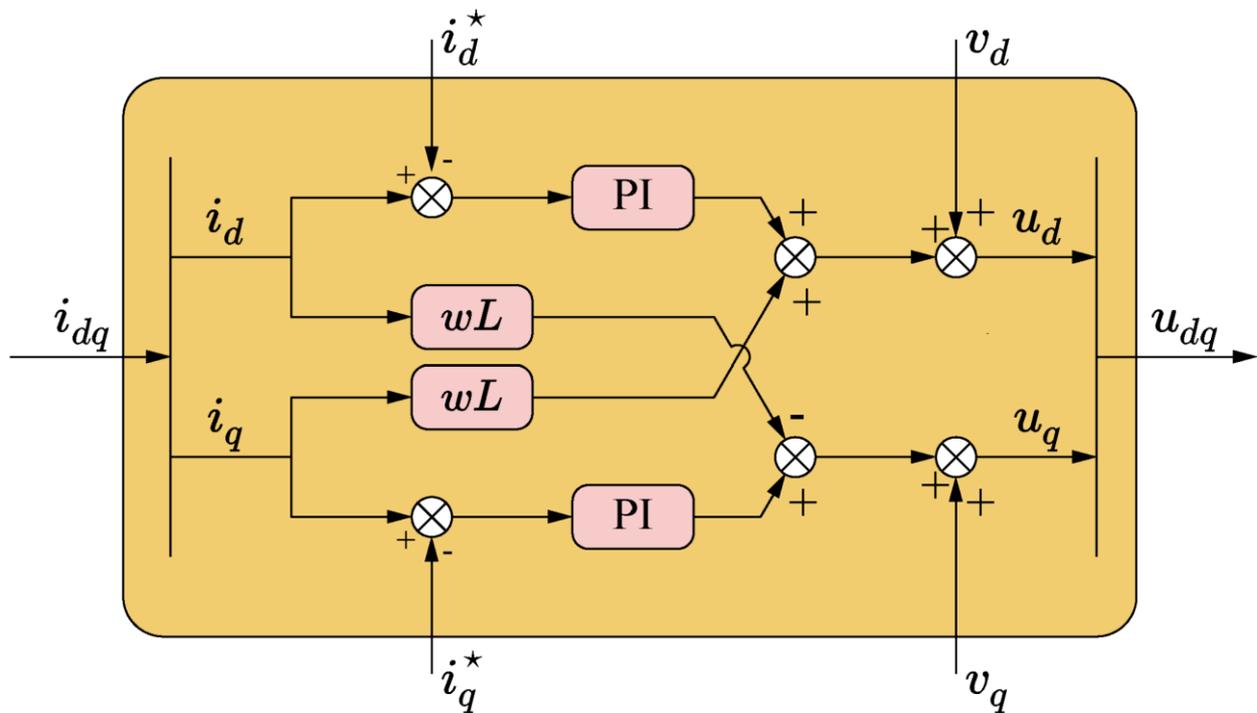


Figura 7-15. El lazo de control de corrientes

#### 7.3.2.1 Antes del control

Al ejecutar el modelo pero antes de iniciar el control, el DSP lee correctamente las tensiones de red (izquierda) y la tensión del condensador (derecha). Las corrientes están a 0 porque todavía no se ha iniciado el control y el convertidor no ha comenzado a conmutar.

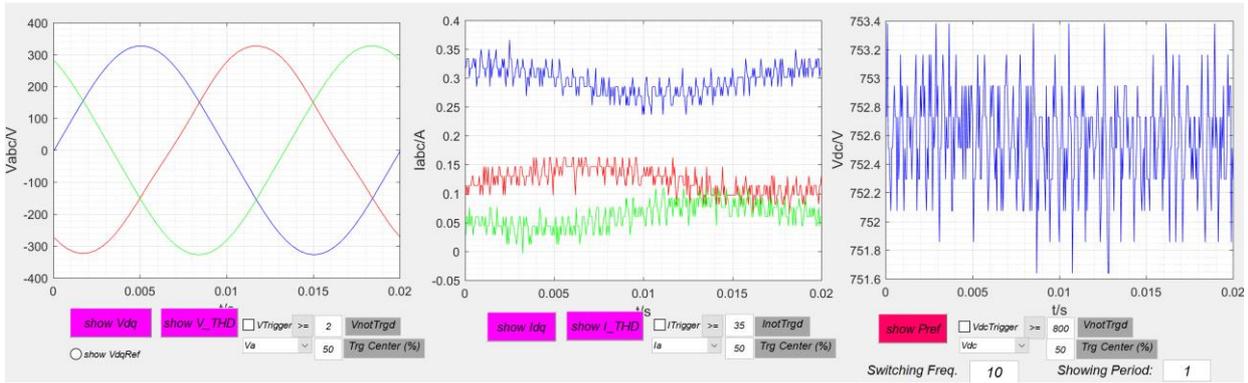


Figura 7-16. Interfaz Matlab del DSP antes de iniciar el control

### 7.3.2.2 Potencia de referencia 5 kW

Una vez que arranca el control, para una potencia de referencia de 5kW, se puede comprobar que las corrientes se estabilizan entorno a un valor concreto, que es el valor que hace que en el DC-link haya una potencia de 5k.

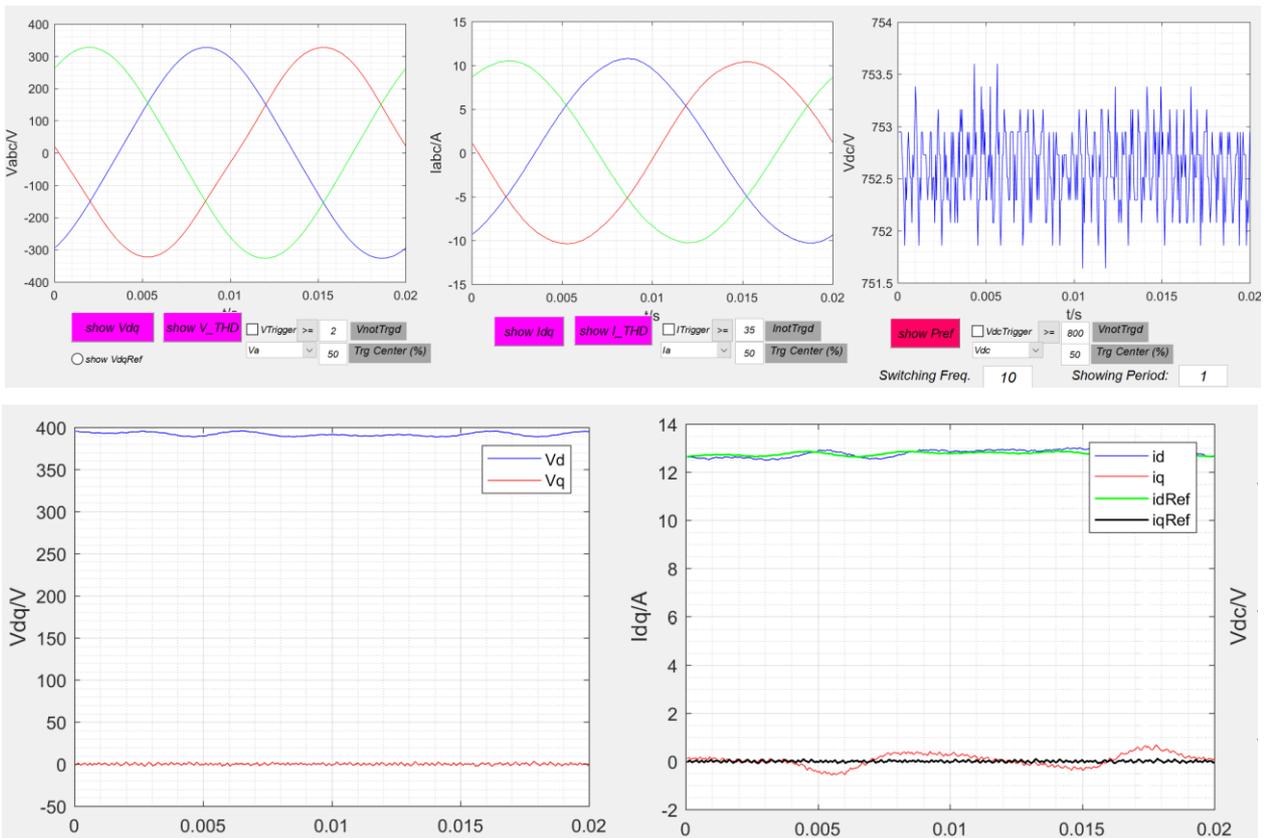


Figura 7-17. Interfaz Matlab del DSP al controlar con  $p=5000W$

En la figura 7-17 se pueden ver las corrientes dq que calcula el controlador, y cómo las corrientes reales medidas sobre el sistema siguen dichas referencias.

Las referencias dq calculadas deberían ser una magnitud constante. Sin embargo, se pueden observar ciertas oscilaciones que no deberían aparecer. Esto se debe a la calibración de los sensores AC, que, a pesar de representar una senoide en la interfaz del DSP, existen ciertas inexactitudes en los valores intermedios, que hacen que la conversión abc-dq no sea tan exacta como debería ser. Este defecto en las calibraciones se abordará con más detalle en la sección 9.

### 7.3.2.3 Potencia de referencia 10 kW

Cuando la referencia de potencia pasa de ser 5k a 10k, las corrientes de referencia cambian, y las del sistema siguen a la referencia, como se puede observar en la imagen. Las corrientes han pasado de 12A a 25A aproximadamente.

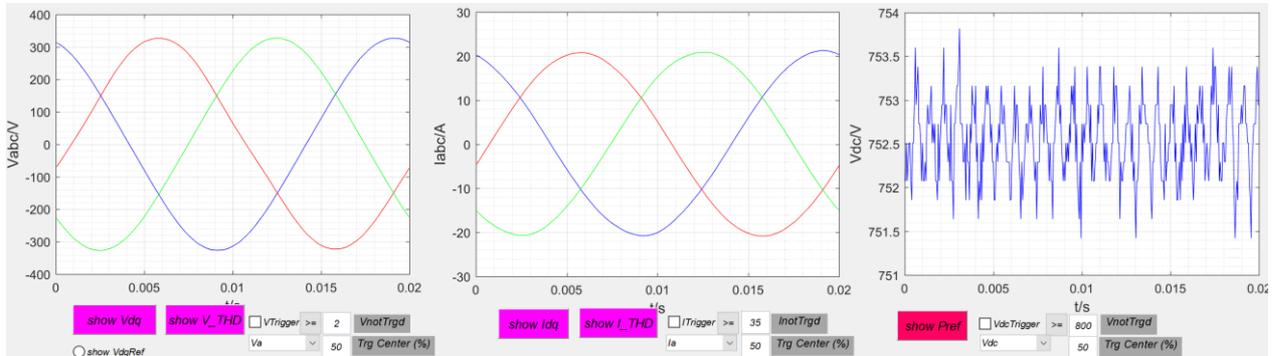


Figura 7-18. Tensión del dc-link en la interfaz Matlab del DSP

En las variables mostradas por code composer, se puede comprobar que la nueva potencia del sistema es 10k. Al cambiar la referencia de potencia, el algoritmo recalcula los valores de referencia de Idq, y las corrientes reales del sistema siguen a estos nuevos valores de referencia.

## 7.4 Inversor conectado a red con circuito de pregarra y control de la tensión del DC-link

El último modelo que se probó antes de comenzar a modelar los paneles PV fue un dc-link (condensador) conectado a la red a través de un inversor. Para este modelo se implementan dos lazos de control, un lazo interno de corrientes (que es el que se probó en el modelo anterior) y un lazo externo sobre la tensión del condensador. A partir de una tensión de referencia, se calcula la consigna de p y q que se le pasa al lazo interno, para que las corrientes inyecten o extraigan la potencia necesaria para regular la tensión del condensador al valor de referencia. Es decir, la diferencia con el modelo anterior es que no se imponen las potencias de referencia directamente, sino que estas referencias son el resultado de un bucle de control más externo que las calcula en función de la tensión actual del condensador y de la tensión que se quiere obtener en el condensador.

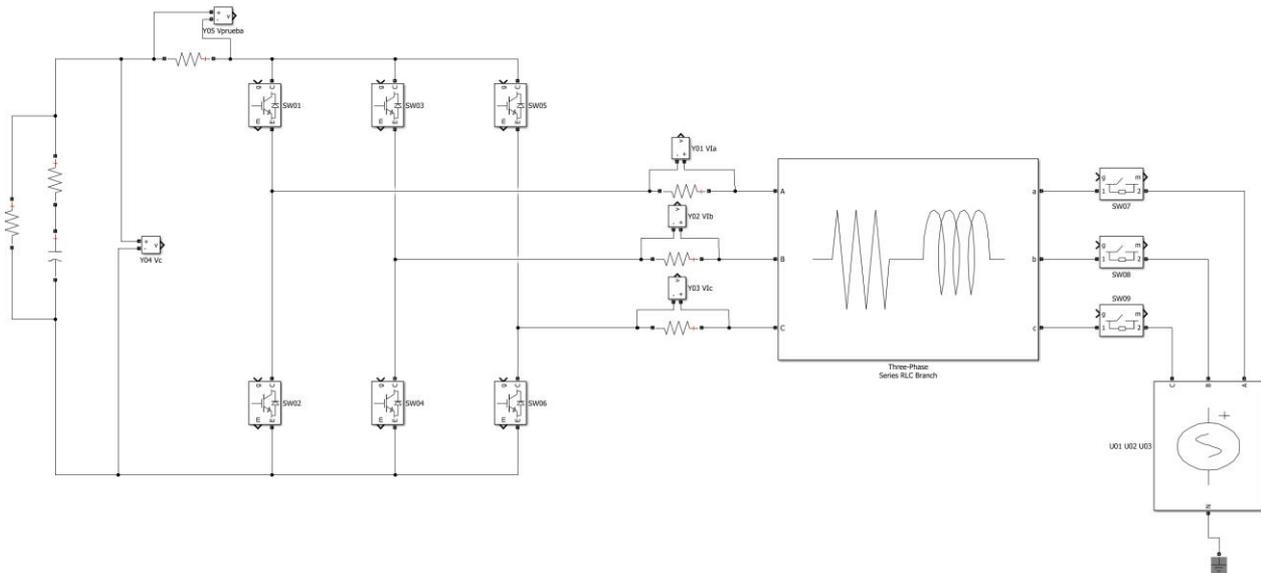


Figura 7-19. Dc-link conectado a red

En esta simulación se detectó un comportamiento extraño del modelo, que no se ha podido resolver, y que se explica a continuación.

#### 7.4.1 Comportamiento inusual del modelo. Mal funcionamiento en la FPGA

En este punto de desarrollo, se detectó por primera vez un comportamiento inusual del modelo, que hace que el campo solar con control MPPT, objetivo último de este proyecto, no se haya podido implementar satisfactoriamente, como se detallará más adelante. Este fallo en el modelo se ha atribuido a un mal funcionamiento de la FPGA, que no modela bien el sistema eléctrico. Otra posible explicación es que, como no se conoce el orden exacto en el que la FPGA ejecuta el modelo, puede que sea necesario modelar los tiempos muertos propios de los IGBTs, que no han sido considerados en ningún modelo, y que esta diferencia sea la que está causando el mal funcionamiento. Estas hipótesis quedan pendientes de verificar y de probar más en profundidad, en estudios que se realizarán a la finalización de este trabajo de fin de grado.

El comportamiento que cabría esperar de un sistema como el representado en la figura 7-19 es que, una vez que la tensión del condensador cambia y se estabiliza entorno a su nuevo valor, tanto las corrientes que se inyectan desde la red, como la que sale del condensador, deberían hacerse 0, ya que el condensador se mantiene a una tensión estable.

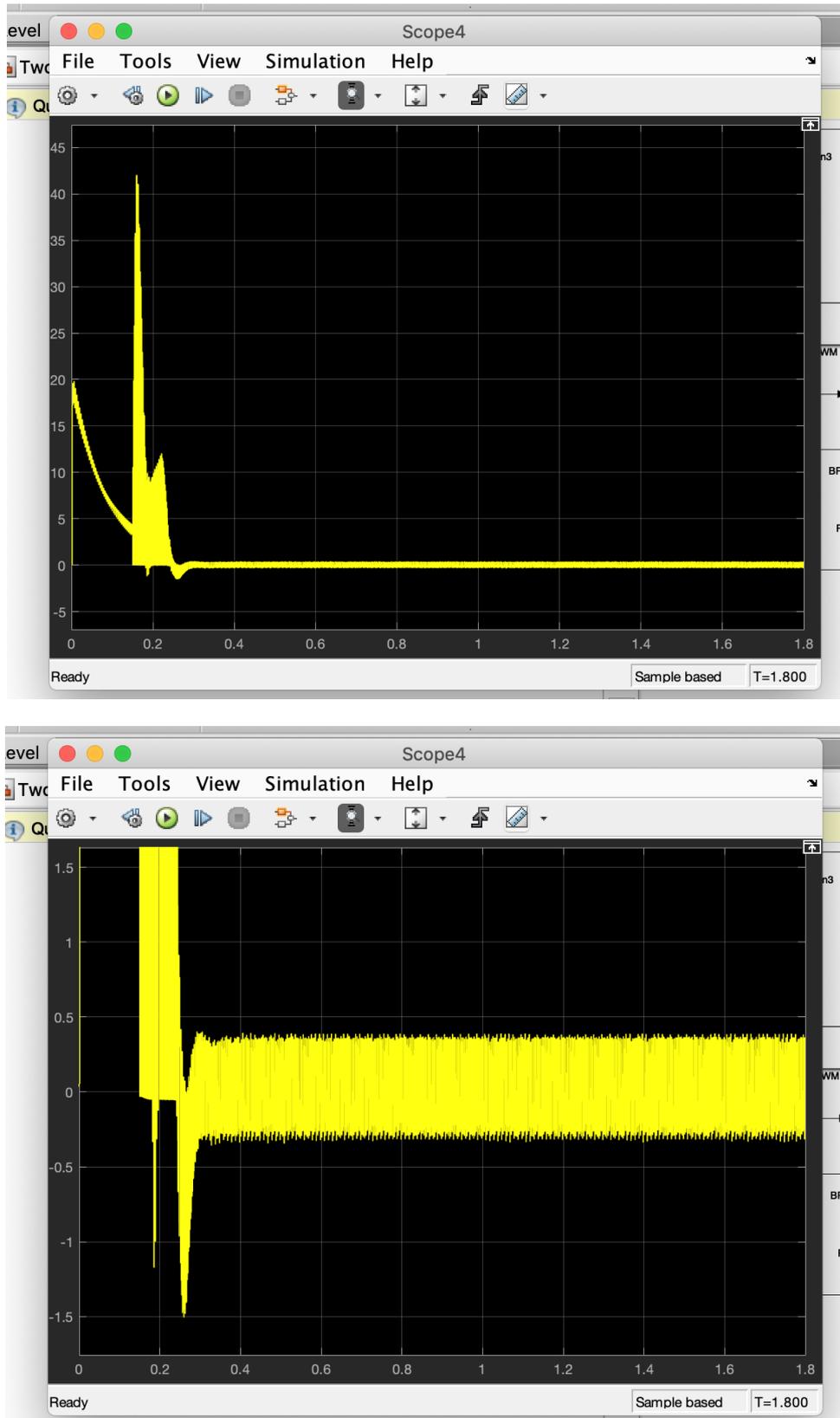


Figura 7-20. Comportamiento correcto de la corriente que sale del condensador

En estas gráficas se representa la corriente que sale del condensador. Se puede observar que en un transitorio inicial crece para cargar el condensador y aumentar su tensión hasta la tensión de referencia (750V), pero que, una vez que se estabiliza la tensión del condensador entorno al valor de referencia, la corriente se hace cero. r

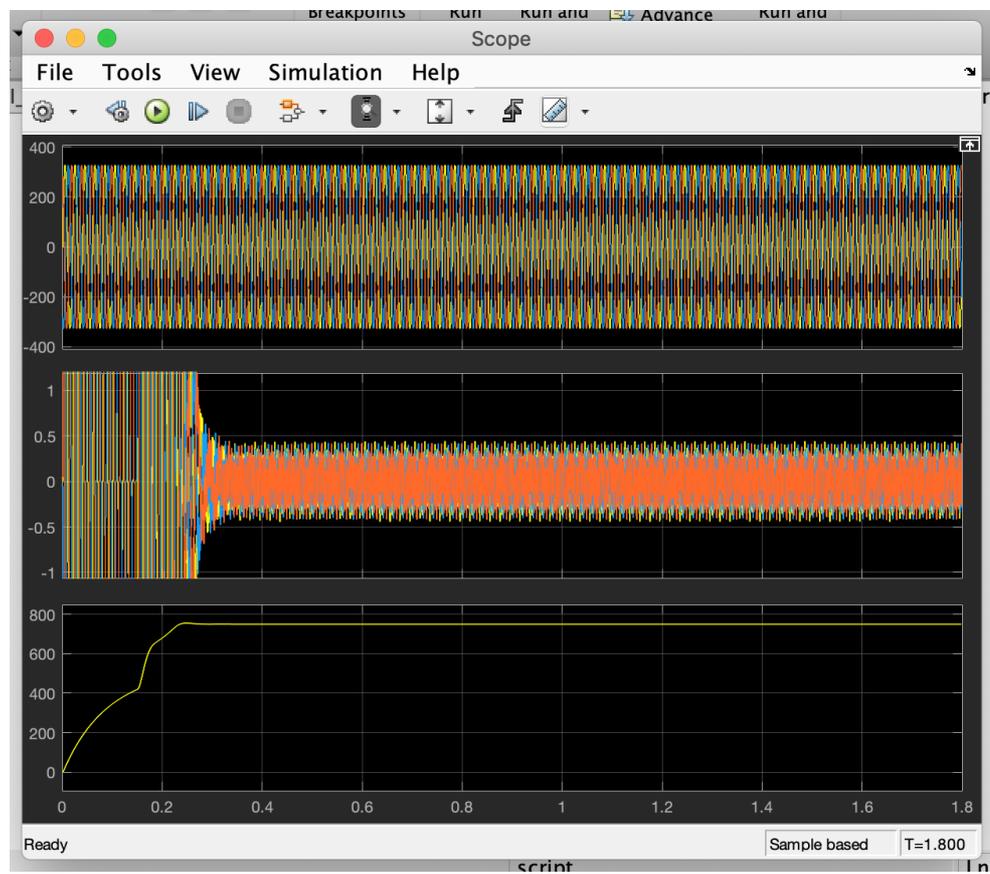
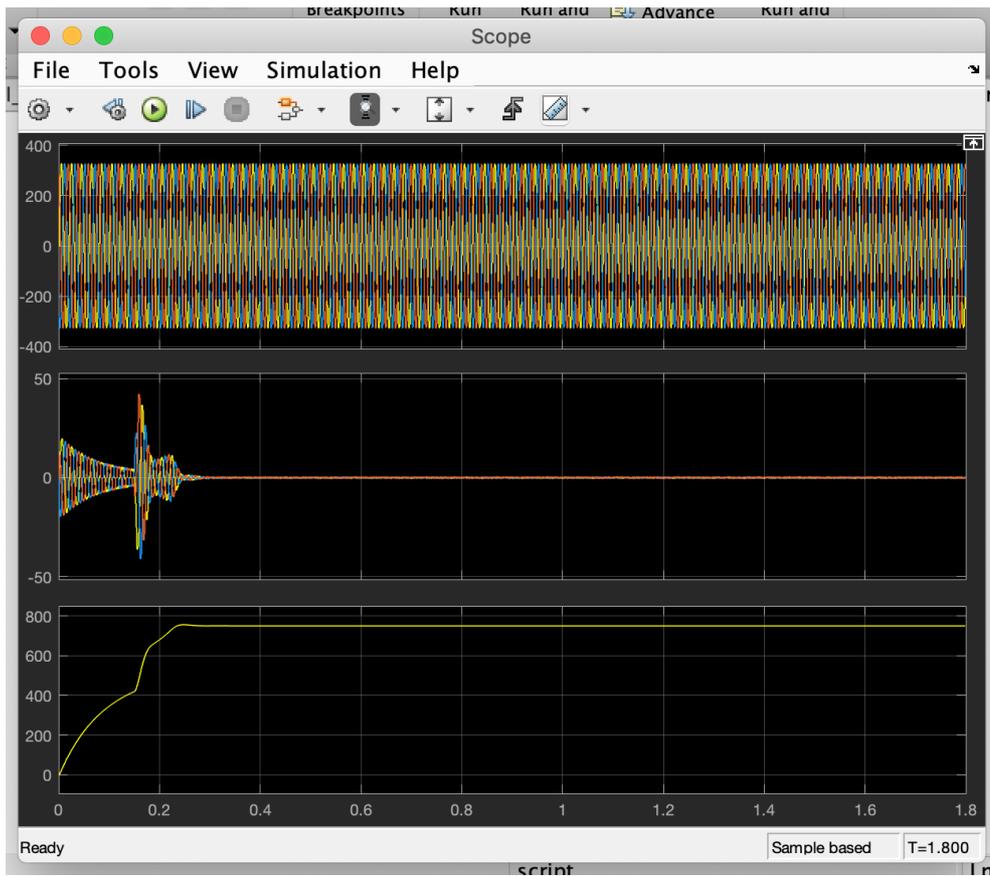


Figura 7-21. Comportamiento correcto de las corrientes inyectadas desde la red

Por otro lado, este es el comportamiento que cabría esperar de las corrientes que se inyectan desde la red a través del inversor. Al igual que la corriente que sale del condensador, crecen en un transitorio inicial, para cargar el condensador, pero después se hacen cero. Las gráficas de las figuras 7-20 y 7-21 han sido obtenidas en simulaciones offline de Simulink.

Sin embargo, este no es el comportamiento que se observa en el sistema implementado en la FPGA. A pesar de que la potencia medida en el DC-link es cero, las corrientes inyectadas desde la red no son cero, y la potencia medida en el lado AC son 7kW aproximadamente. Es decir, el sistema se comporta como si hubiera alguna carga conectada al otro lado del inversor que consumiera 7kW de manera continuada e ininterrumpida. Como se puede comprobar en el modelo, no hay ningún elemento conectado que consuma esa gran cantidad de potencia por lo que, aparentemente, esa potencia no va a ningún sitio. Además, si se aumenta la tensión de referencia del condensador, también aumenta esa potencia consumida.

A pesar de este mal funcionamiento, el controlador funciona correctamente. Es decir, inyecta la potencia necesaria para que el condensador se regule a la tensión de referencia, inyectando también ese plus de potencia consumida que necesita el modelo para que la tensión del condensador no se desplome.

Este efecto hace que no se pueda simular correctamente un campo solar controlado con un algoritmo MPPT, porque la fuente FER solar no está preparada para absorber potencia como lo está una fuente dc.

A continuación, se adjunta la representación de las magnitudes más relevantes del modelo, para comprobar el funcionamiento explicado.

## 7.4.2 Representación de magnitudes eléctricas

### 7.4.2.1 Tensiones de red

Al igual que en el modelo anterior, se generan en la CPU, que se las pasa a la FPGA a través del bloque eHS. Si se representan en el scope de la interfaz gráfica se la figura 7-22.

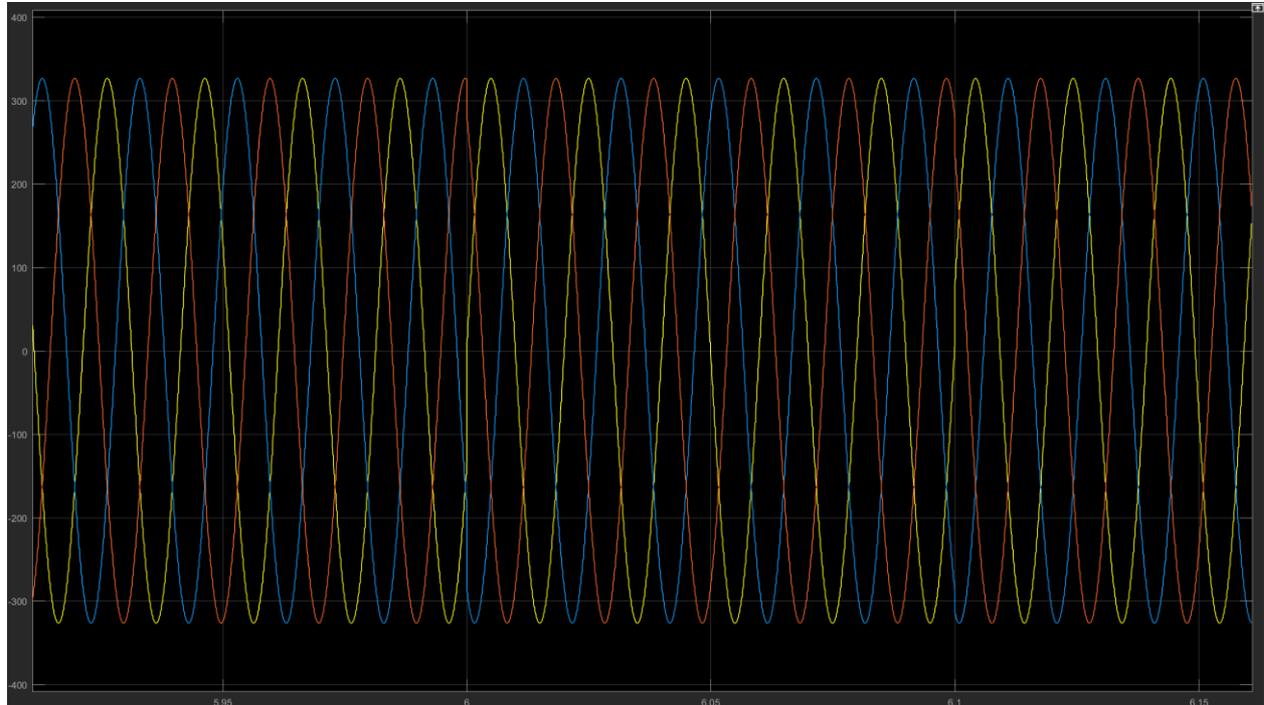


Figura 7-22. Tensiones de la red en la consola (subsistema GUI)

También se pueden medir a la salida de la OPAL, con la ganancia correspondiente aplicada 16/500 (figura 7-23)

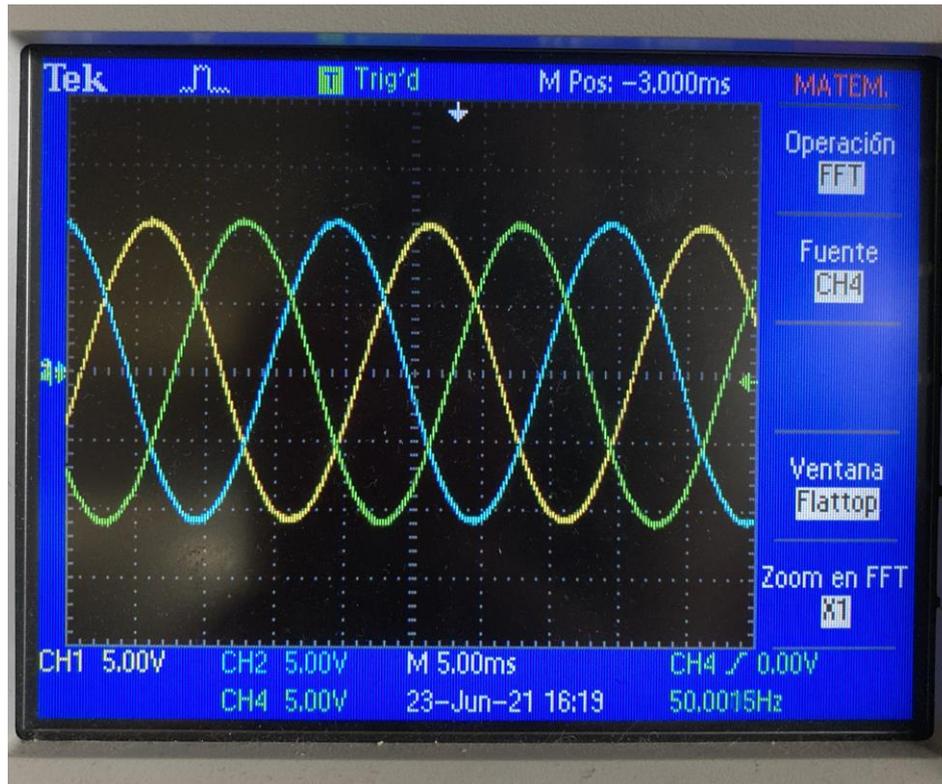


Figura 7-23. Tensiones de la red en el osciloscopio (salida OPAL)

Por último se representan las tensiones de red en la interfaz gráfica del DSP (figura 7-24)

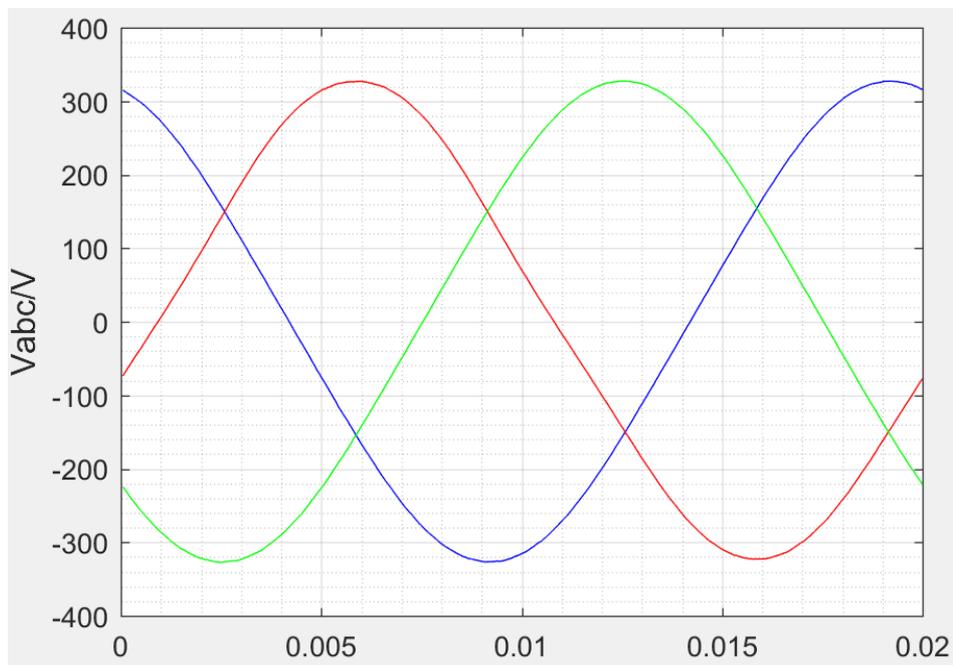


Figura 7-24. Tensiones de la red en la interfaz Matlab del DSP

Se comprueba que las tensiones de red que se generan en la FPGA llegan correctamente al DSP, por lo que el algoritmo de control puede aplicar el PLL correctamente.

### 7.4.2.2 Tensión del dc-link

La tensión del condensador es el objeto de control de este algoritmo, y se regula entorno al valor de referencia establecido. Este valor de referencia se puede modificar desde la interfaz de Matlab, y así comprobar el funcionamiento del lazo de control en tiempo real.

Por ejemplo, si se establece que la tensión de referencia sea de 700V, se puede comprobar que la tensión del condensador alcanza ese valor y se mantiene en ese punto.

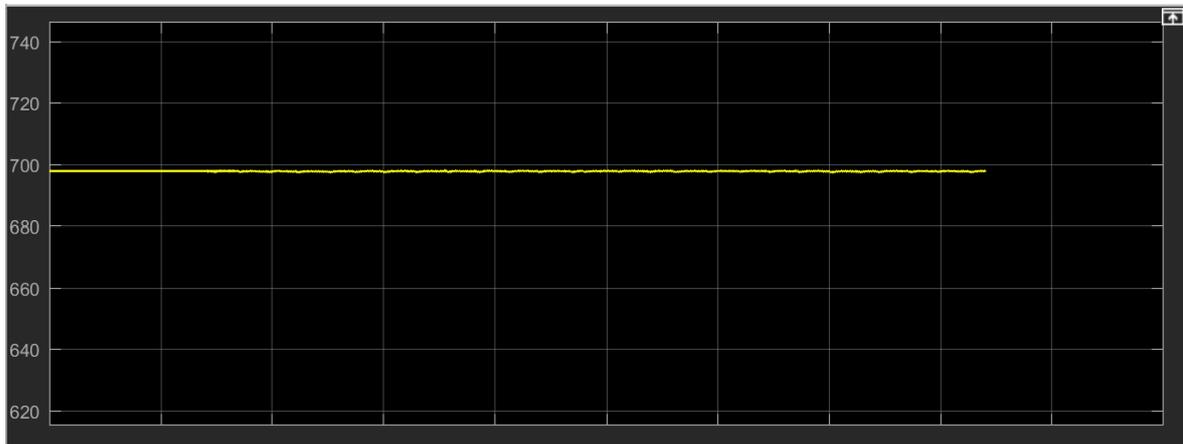


Figura 7-25. Tensión del dc-link en la consola (subsistema GUI)

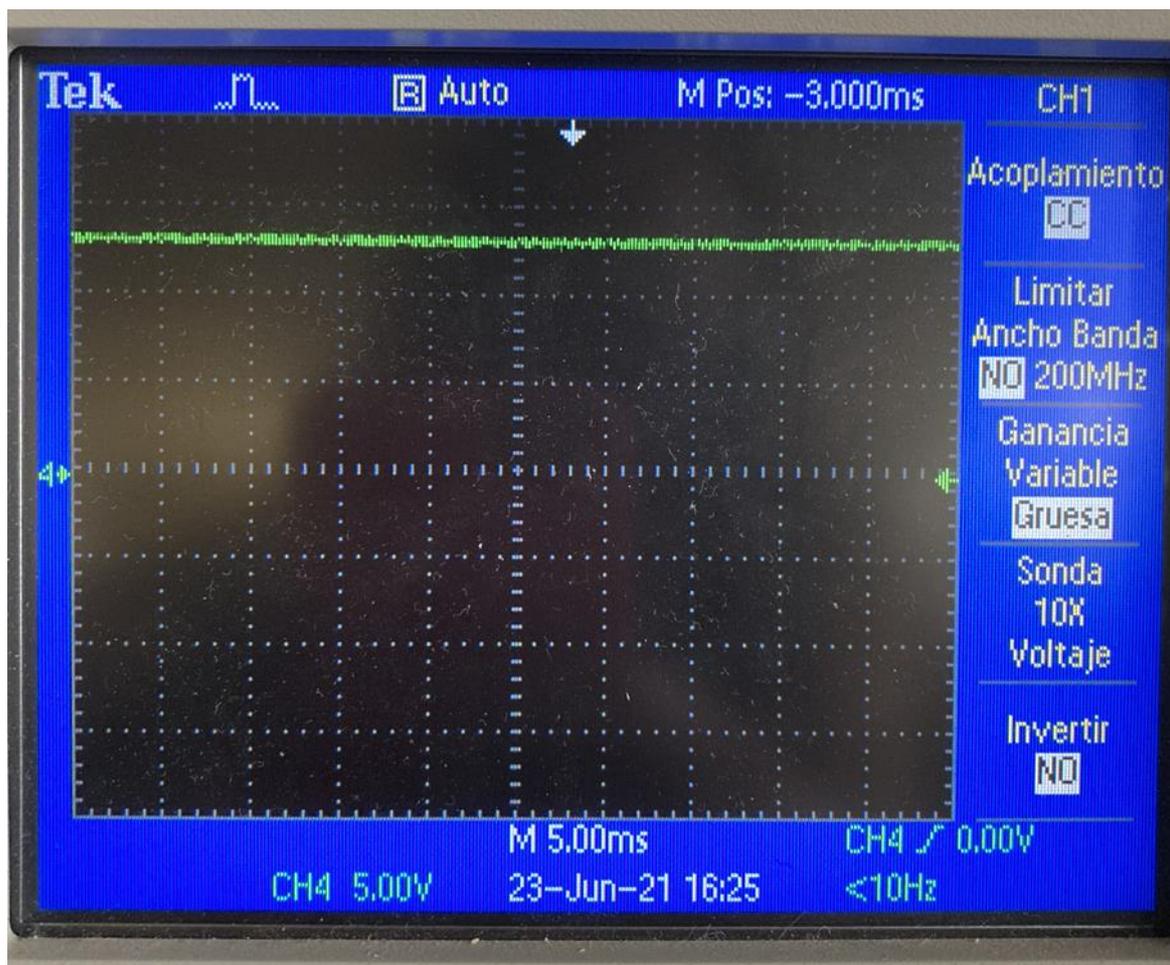


Figura 7-26. Tensión del dc-link en el osciloscopio (salida OPAL)

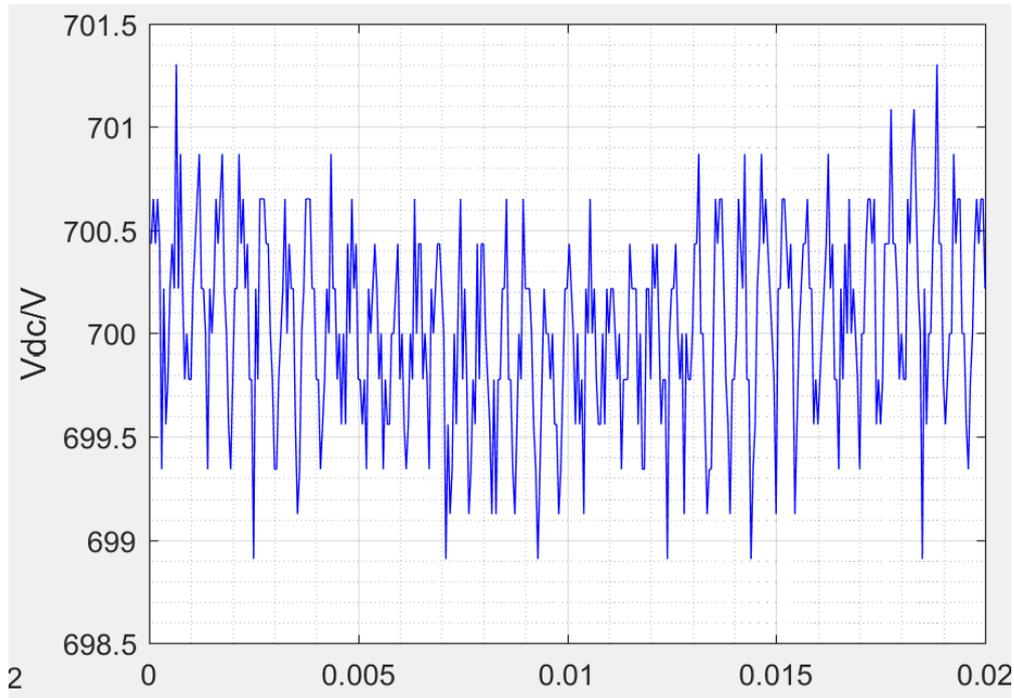


Figura 7-27. Tensión del dc-link en la interfaz Matlab del DSP

En las figuras 7-25, 7-26 y 7-27 se comprueba que la tensión del dc-link se genera y se mide correctamente.

#### 7.4.2.3 Corrientes

En este punto se pone de manifiesto el mal funcionamiento del modelo. Una vez que la tensión del condensador se ha regulado entorno a un valor de tensión concreto, las corrientes que entran/salen desde la red deberían hacerse cero. Sin embargo, estas corrientes adoptan un valor constante, como si en el sistema hubiera alguna carga que consumiera potencia en régimen permanente.

Además, tras varias pruebas, se llegó a la conclusión de que los sensores de corriente a veces no funcionaban correctamente en la FPGA, por lo que, pensando que quizás fuera un error de medida, se sustituyeron los amperímetros por una resistencia con un voltímetro, de manera que se calcula la corriente como el cociente de esa tensión entre el valor de la resistencia, que es muy pequeña, 0.01 Ohm, para evitar que genere pérdidas.

Aun así, las corrientes seguían siendo distintas de cero y con un valor estable, por lo que se llegó a la conclusión de que no se trataba de un error de medida ni de un transitorio, sino que el error está en el modelo del sistema eléctrico, en su implementación en la FPGA.

Por tanto, estas corrientes se corresponden con las que necesitaría una carga que consuma esa cantidad de potencia en régimen permanente, pero esta carga no está en el modelo.

En las figuras 7-28, 7-29 y 7-30 se comprueba que las corrientes que hay en el sistema y que lee el DSP no son cero, como deberían ser.

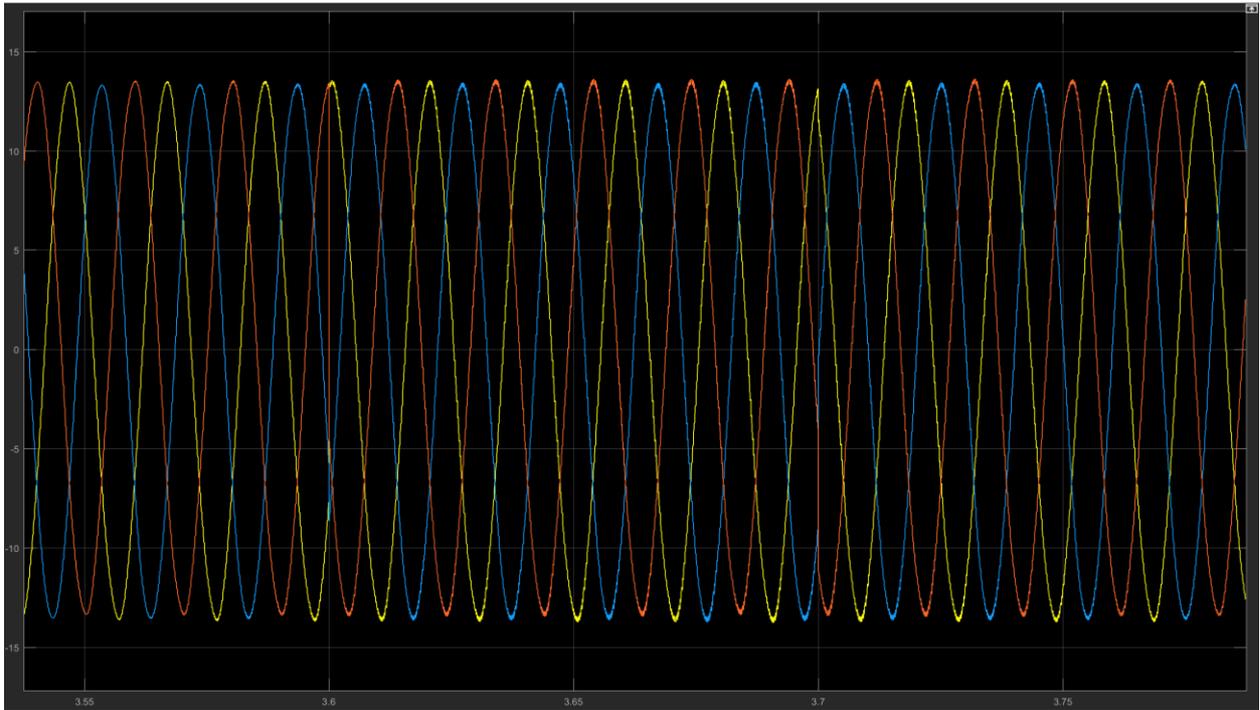


Figura 7-28. Corrientes en la consola (subsistema GUI)

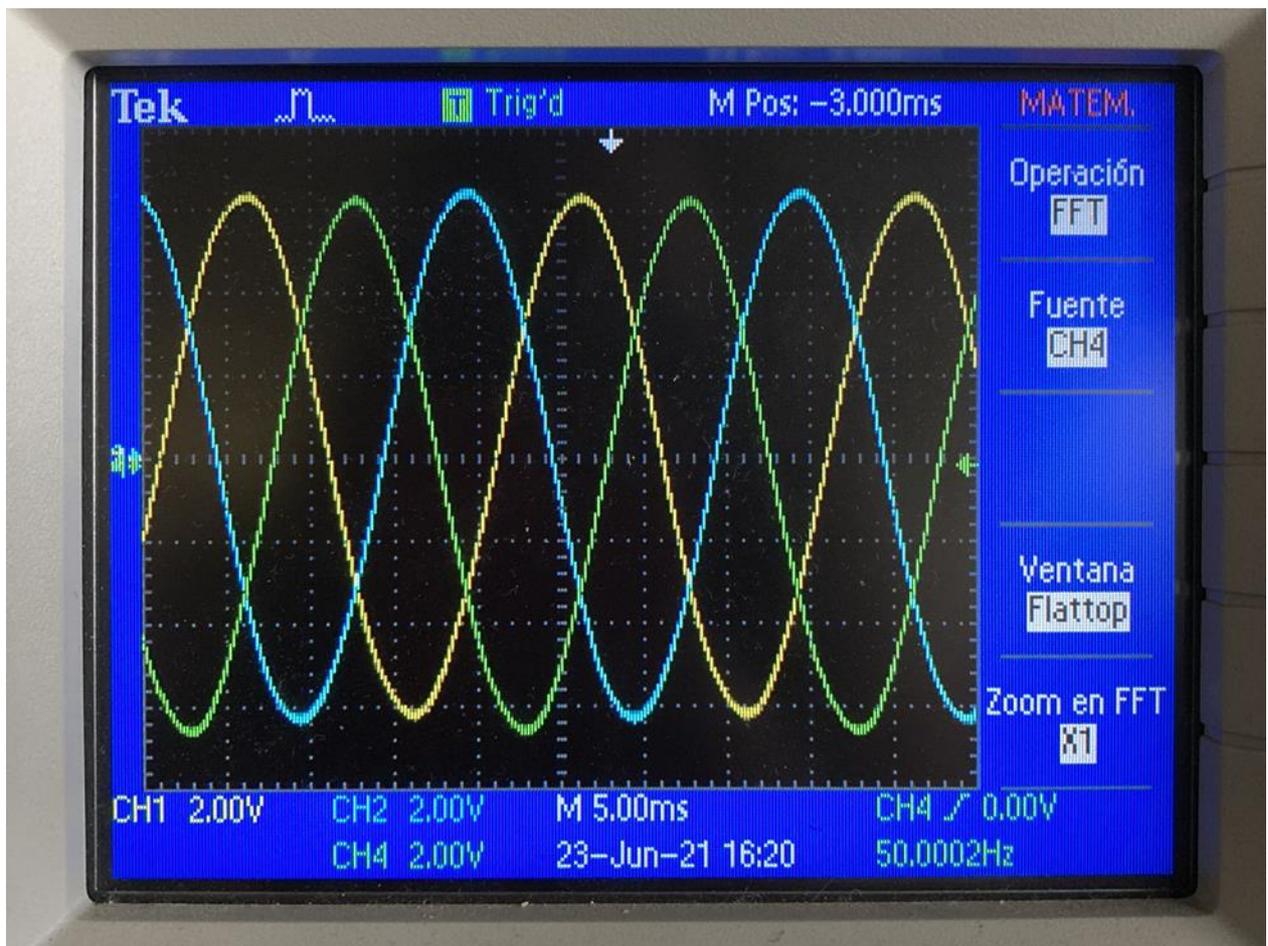


Figura 7-29. Corrientes en el osciloscopio (salida OPAL)

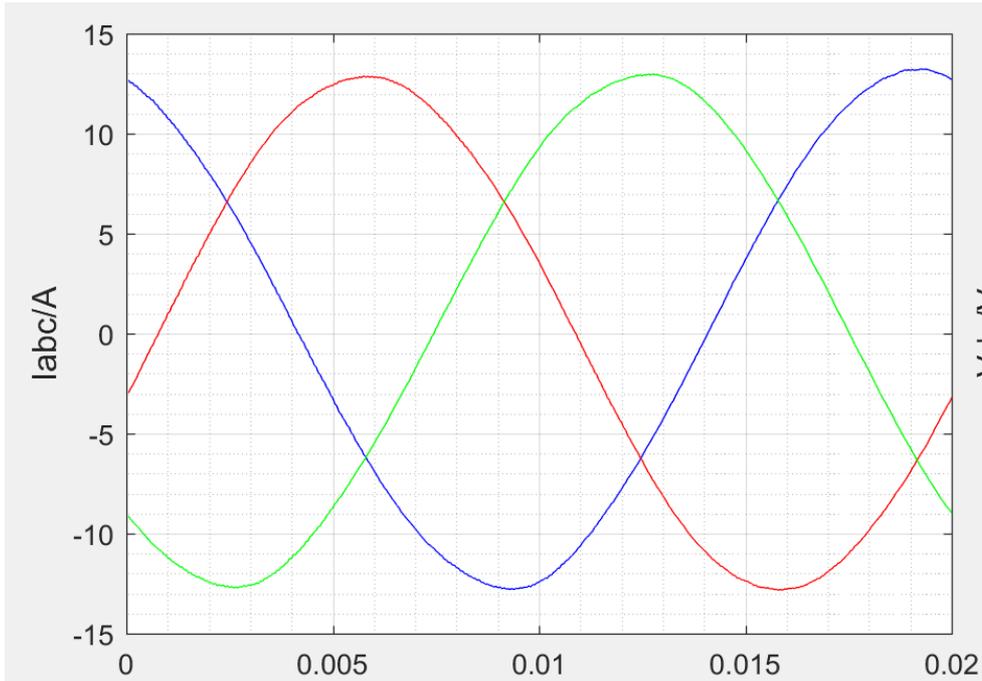


Figura 7-30. Corrientes en la interfaz Matlab del DSP

#### 7.4.2.4 Análisis de potencias

Como última evidencia de este comportamiento extraño, se adjuntan las potencias del sistema, calculadas tanto en el lado DC (figura 7-31) como en el lado AC (figura 7-32).

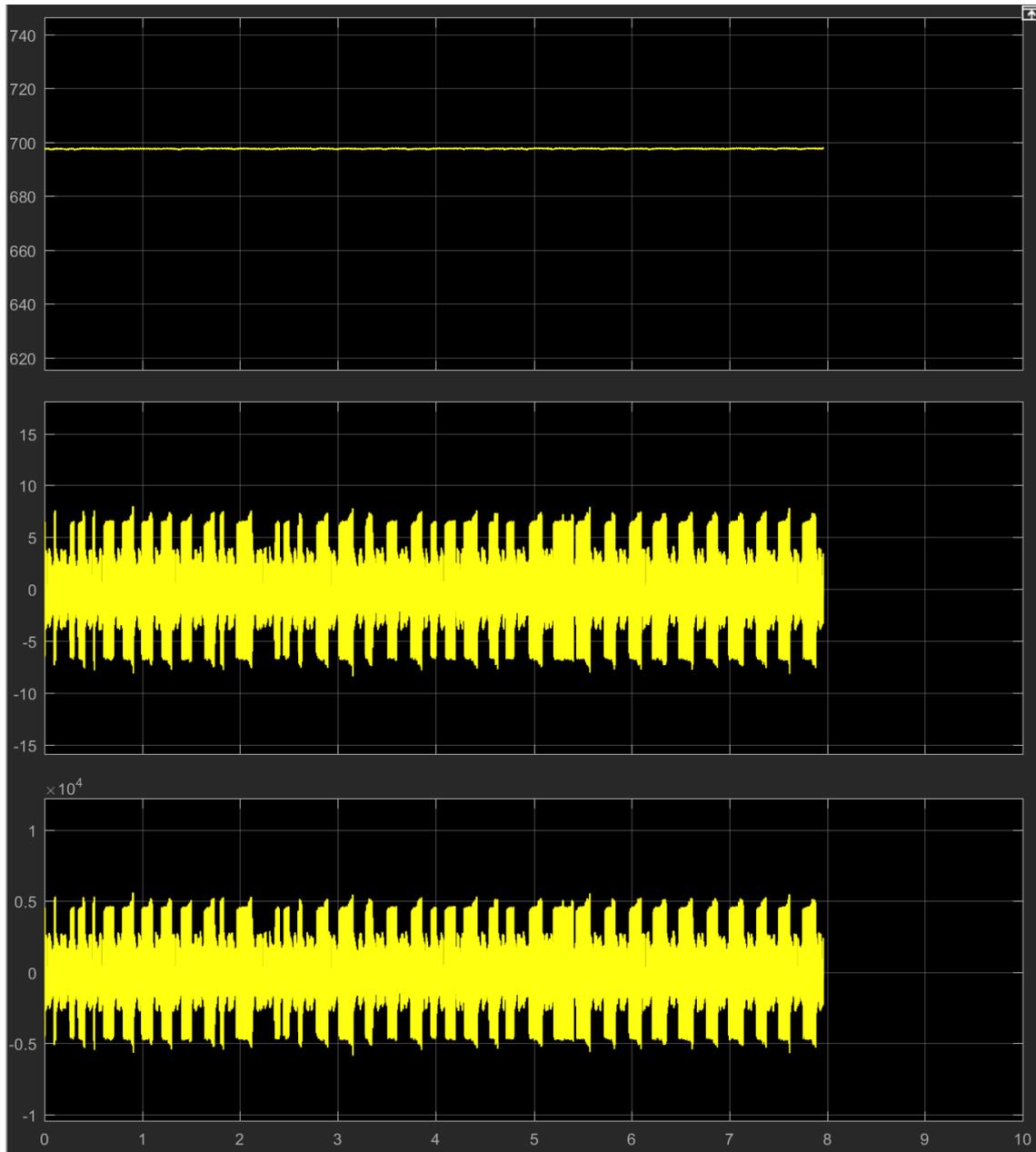


Figura 7-31. Potencia en el DC-link

La primera gráfica es la tensión del condensador, la segunda, la corriente a la salida del condensador, y por último, la tercera gráfica es la potencia en el DC-link, calculada como el producto de las dos primeras magnitudes. Como se puede observar, este resultado podría llegar a ser coherente con lo que ocurre en el condensador en la realidad, ya que la potencia está centrada en cero.

Sin embargo, el rizado que se obtiene es de casi 5kW, y tampoco debería ser así, lo que también indica que el modelo no se está ejecutando correctamente.

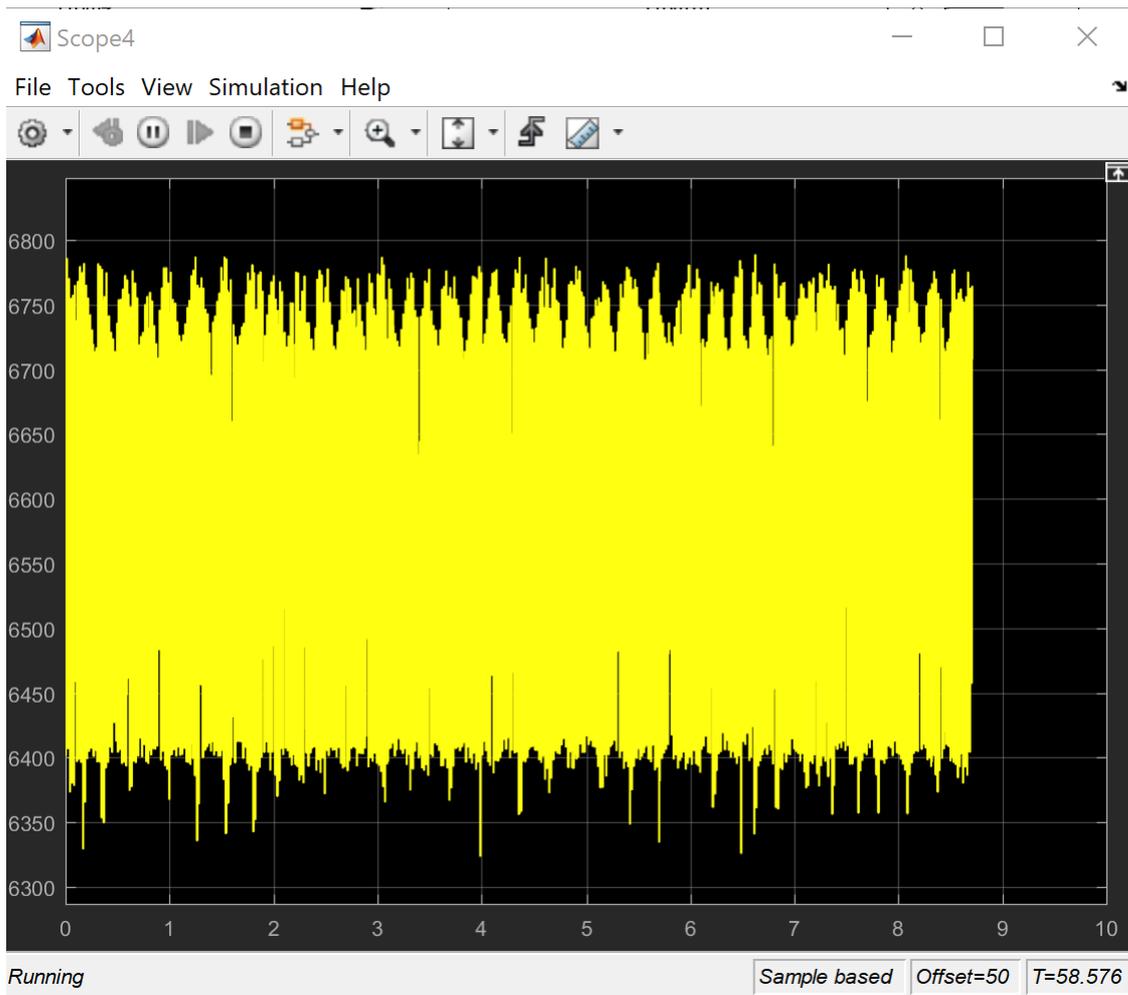


Figura 7-32. Potencia en la parte AC

Por otra parte, si se calcula la potencia en el lado AC, aunque debería ser 0, se puede comprobar que no lo es. Como se ha explicado anteriormente, el modelo se comporta como si hubiera un consumo de potencia que no existe.

### 7.4.3 Comprobación del funcionamiento del controlador

A pesar del mal funcionamiento del modelo, el control sí funciona correctamente, como se ha explicado anteriormente. Es decir, el algoritmo de control es capaz de regular la tensión del condensador al valor de referencia especificado, aunque para ello deba inyectar en el sistema 6.6 kW que no debería inyectar. En la figura 7-33 se muestra cómo la tensión del condensador sigue a la referencia cuando esta cambia de 700V a 750V.

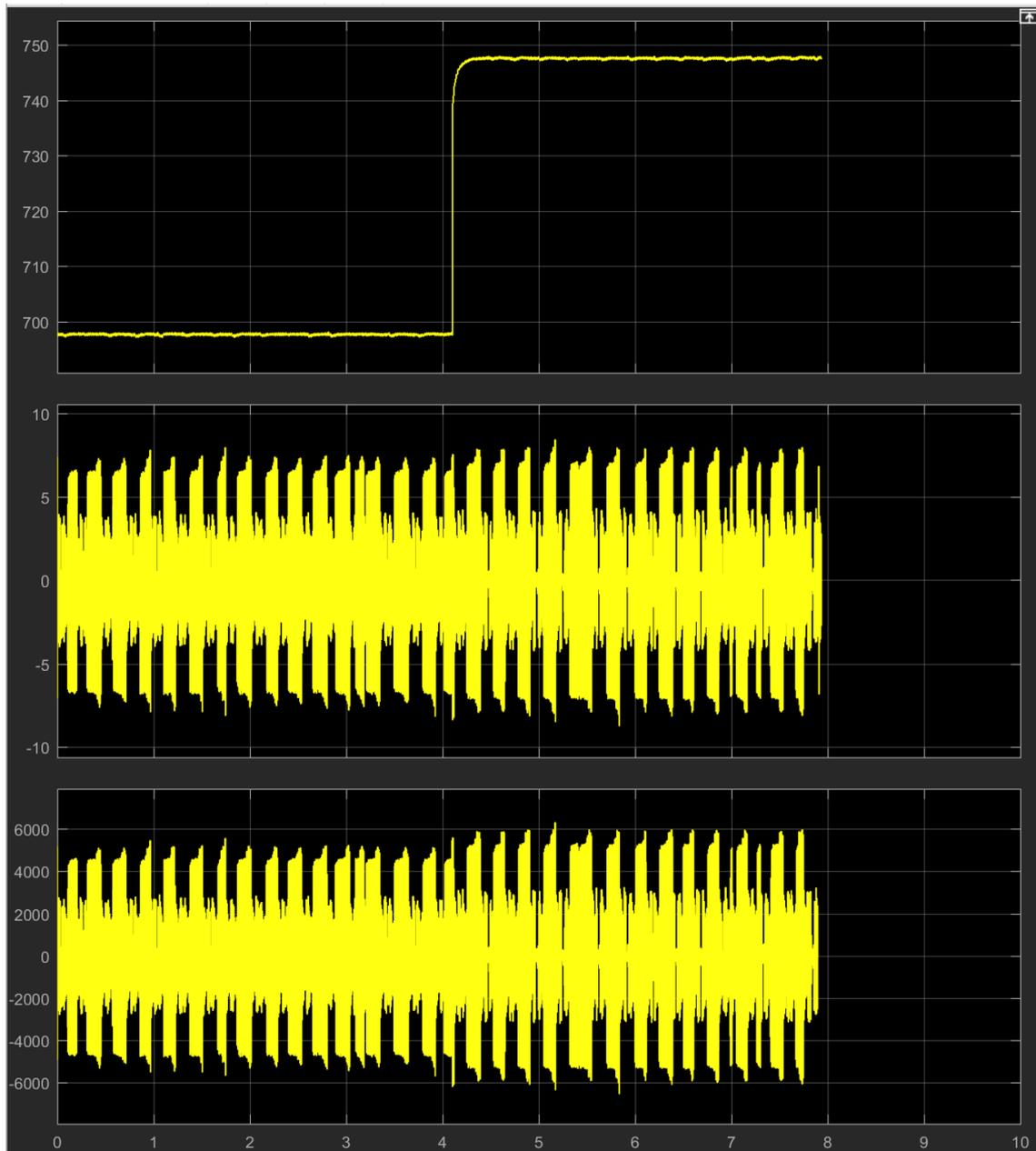


Figura 7-33. Potencia en el DC-link

Por último, este cambio de tensión también conlleva un salto en las corrientes que se inyectan desde la red (figura 7-35) y en la potencia que se consume (figura 7-34). El comportamiento que cabría esperar sería un transitorio de corriente y potencia, hasta que la tensión del condensador alcanzara su nuevo valor, y que luego, tanto corrientes como potencia, se hicieran cero. Sin embargo, se puede observar en las gráficas que este no es el comportamiento que se obtiene del modelo implementado en la FPGA.

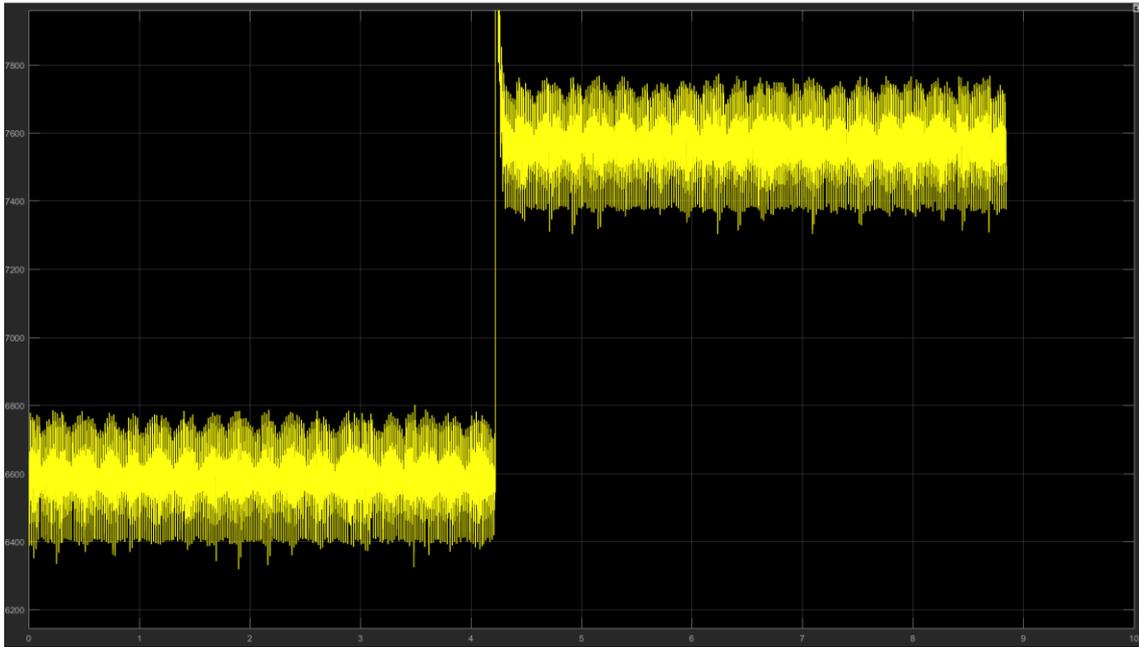


Figura 7-34. Salto en la potencia AC

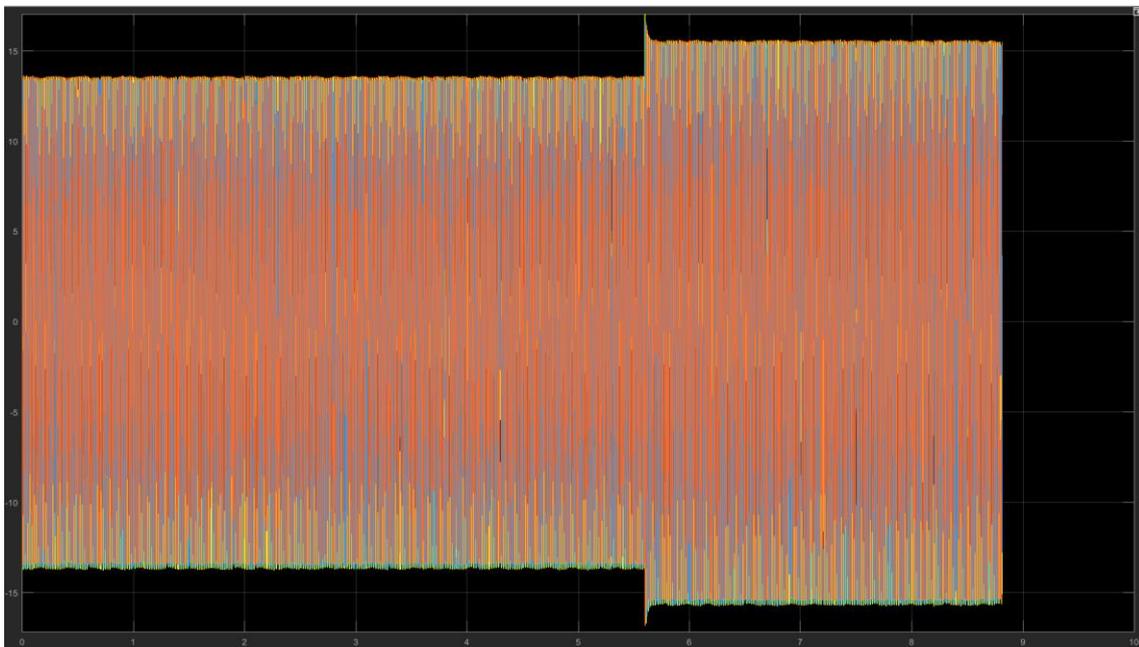


Figura 7-35. Salto en las corrientes

## 8 ALGORITMOS MPPT

Este capítulo se centra en explicar lo que sería la implementación del sistema PV en la FPGA. Sin embargo, este modelo no se ha podido llegar a ejecutar satisfactoriamente debido al mal funcionamiento del sistema, pero se deja propuesto para probar tras la entrega de esta memoria.

### 8.1 Sistema a implementar

El sistema total que se pretende controlar está compuesto por una fuente de energía renovable conectada a la red eléctrica a través de un convertidor de potencia.

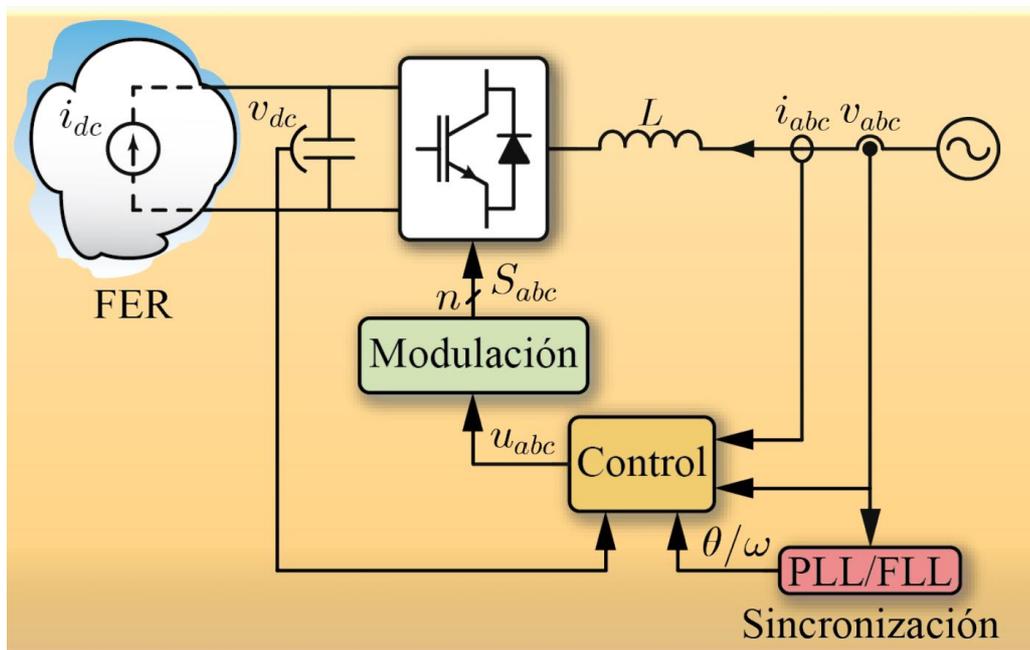


Figura 8-1. Sistema eléctrico a implementar en la FPGA

De este sistema se pueden extraer dos ecuaciones fundamentales que determinan su comportamiento. La primera ecuación regula la dinámica de la corriente de entrada al convertidor (proveniente de la red), y la segunda ecuación es un balance de potencias sobre el condensador.

La primera ecuación se puede expresar en función de los ejes abc o de los ejes dq. En esta ecuación se observa que la corriente que entra al convertidor desde la red es una magnitud controlable, ya que se puede variar la tensión  $u_{dq}$  mediante la modulación del convertidor. Por eso, esta ecuación se utiliza para implementar el bucle de control sobre las corrientes.

$$v_{abc} = Ri_{abc} + L \frac{di_{abc}}{dt} + u_{abc}$$

$$v_{dq} = Ri_{dq} + L \frac{di_{dq}}{dt} + J\omega Li_{dq} + u_{dq}$$

Por otro lado, la segunda ecuación es un balance de potencias sobre el condensador: la potencia que se almacena en el condensador es igual a la que sale del convertidor proveniente de la red, menos la que absorbe la carga, que en este caso sería la FER.

$$C \frac{d}{dt} \left( \frac{V_{dc}^2}{2} \right) = u_{dq}^T i_{dq} - V_{dq} i_{Load}$$

Esta ecuación sirve para regular la tensión a la que se encuentra el condensador. En el algoritmo de control se utiliza para calcular las referencias de potencia activa y reactiva que deben entrar al convertidor provenientes de la red en cada instante, de manera que la tensión del condensador (o Vdc-link) se regule al valor de referencia.

Un tercer lazo de control, externo a los otros dos, sería el algoritmo MPPT (seguimiento del punto de máxima potencia), que en función de las características del panel y de la irradiancia, proporciona la tensión de referencia del DC-link para obtener la máxima potencia posible de la FER en cada instante.

Por tanto, los bloques funcionales que se necesitan para poder regular el sistema completo son cinco:

- Algoritmo MPPT: proporciona la tensión de referencia del dc-link.
- Balance de potencia: utiliza la segunda ecuación, y regula la tensión del condensador. Proporciona los valores instantáneos de potencia activa y reactiva de referencia.
- Teoría de potencia instantánea: mediante un simple cálculo algebraico, calcula la corriente de referencia instantánea a partir de las referencias de potencia instantáneas. Por tanto, este bloque proporciona la corriente de referencia instantánea que entra al convertidor desde la red.
- Control de corrientes: utiliza la primera ecuación y, a partir de la intensidad de referencia y de la intensidad real, mediante un PID, proporciona la tensión de referencia en bornes del convertidor.
- Modulador: a partir de la referencia de tensión que hay que generar en bornes del convertidor, el modulador se encarga de generar las señales de disparo de los interruptores para obtener la tensión deseada.

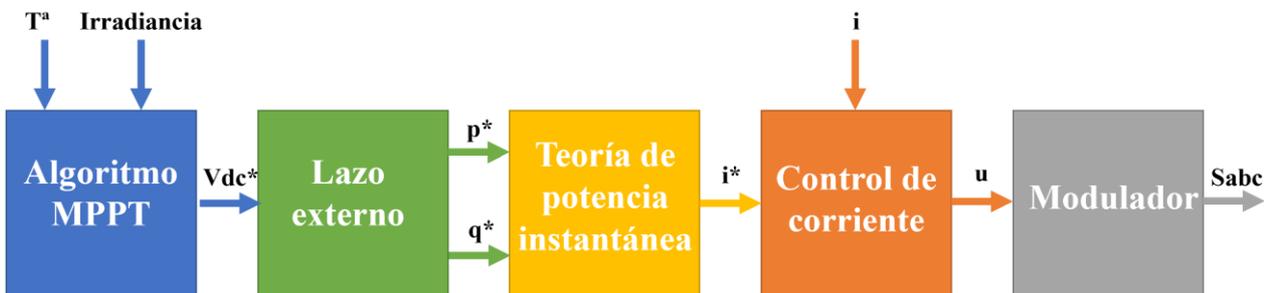


Figura 8-2. Sistema eléctrico a implementar en la FPGA

Este sería el control completo del sistema. Se va a dividir en varias fases, para poder comprobar el funcionamiento de cada etapa.

## 8.2 Modelo de los paneles fotovoltaicos

### 8.2.1 Caracterización de los paneles solares

Para simular el sistema se necesita implementar en la FPGA el modelo de un PV array. Es decir, un conjunto de paneles solares conectados entre sí, de manera que proporcionen la tensión e intensidad que se necesita para nuestra aplicación.

Un campo solar caracteriza con dos curvas: la curva intensidad-tensión y la curva potencia-tensión. Estas curvas características cambian con la temperatura y la irradiancia, de manera que para cada par de condiciones de contorno (irradiancia, temperatura), existe un punto de funcionamiento en el que se extrae la máxima potencia posible.

La función de los algoritmos MPPT es, por tanto, calcular la tensión del punto de máxima potencia en función de la temperatura y la irradiancia, y pasarla como tensión de referencia al algoritmo de control, para que la tensión del condensador se regule a ese valor de referencia y el sistema trabaje en el punto de máxima potencia.

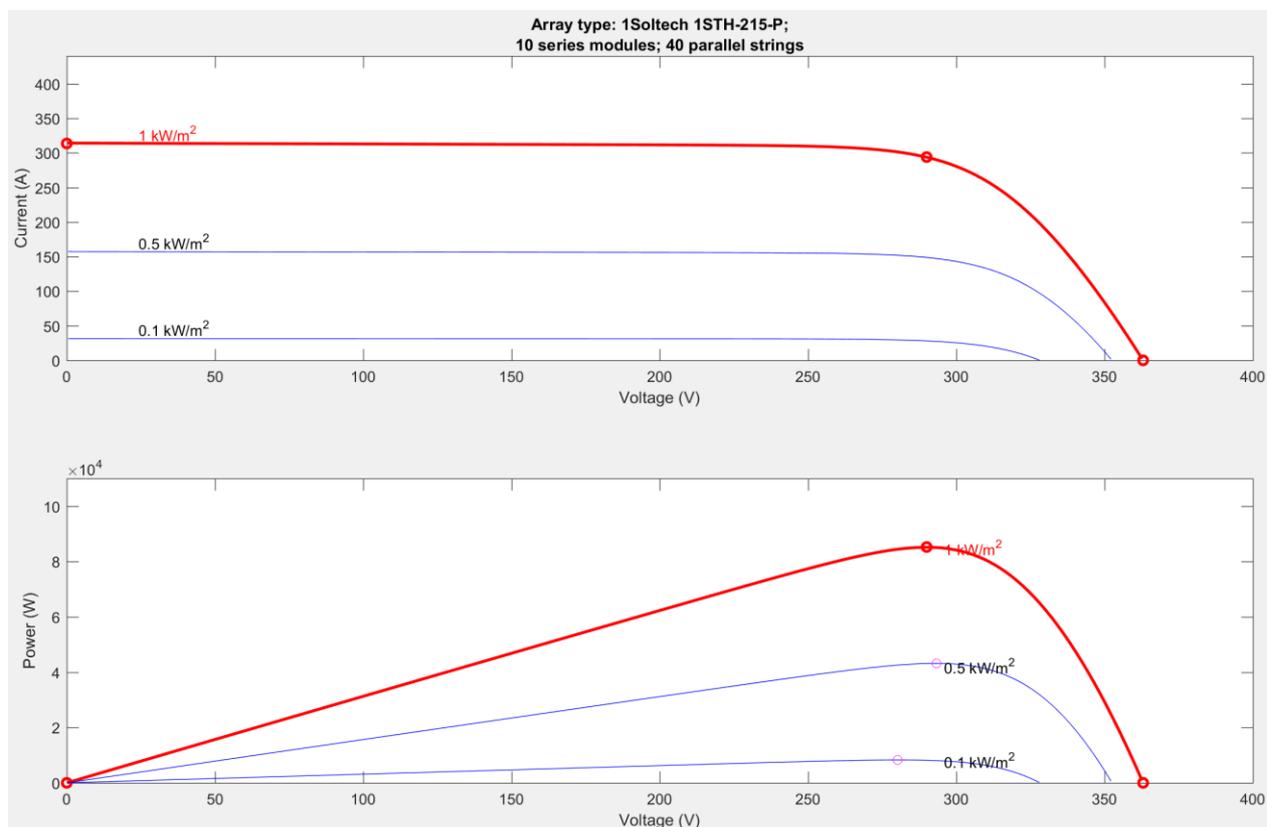


Figura 8-3. Ejemplo de curvas características de un PV array

Para esta aplicación, se supone que tanto la temperatura como la irradiancia son constantes y uniformes. Es decir, todos los paneles están a la misma temperatura y la irradiancia afecta por igual todos los paneles, no hay sombreado parcial. Una posible variación de este modelo es precisamente simular el efecto del sombreado parcial, dividiendo el PV array en zonas a las que les llegan distintos valores de temperatura e irradiancia.

## 8.2.2 Modelo de un panel fotovoltaico

Existe un bloque de Simulink que permite seleccionar el tipo de panel (modelo comercial) y la configuración de conexión (nº de paneles en serie y en paralelo), de manera que se obtenga la curva característica que interesa. Sin embargo, el soporte técnico de la OPAL confirmó que el software RT-LAB no soporta este bloque en la FPGA, por lo que no se puede utilizar.

Sin embargo, un panel PV no es más que una fuente de corriente regulada, y esto sí que se puede implementar en la FPGA. El modelo interno de un panel PV es el que se muestra en la figura 8-4.

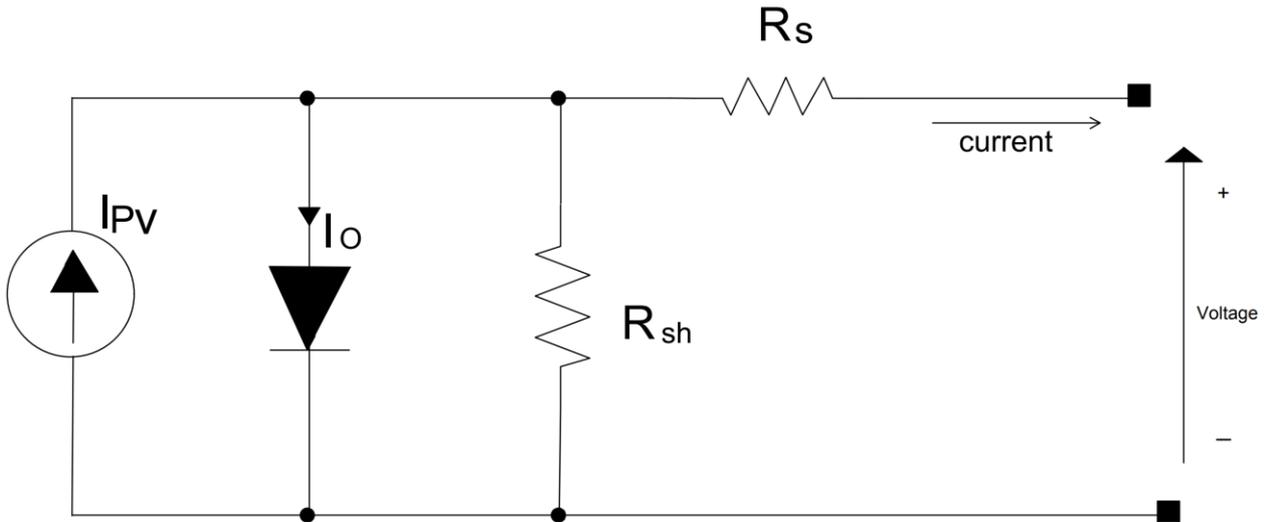


Figura 8-4. Circuito equivalente a un panel fotovoltaico

Para calcular el valor de la fuente de corriente regulable, se utiliza una S-function que calcula  $I_{pv}$  a partir de la temperatura, la irradiancia y la tensión y corriente a la salida del panel. Por tanto, el esquema completo de la implementación del panel es el de la figura 8-5.

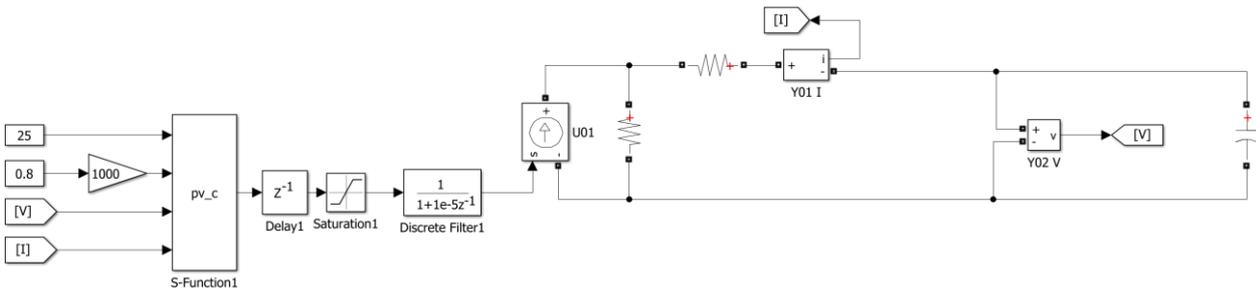


Figura 8-5. Circuito simulink del panel fotovoltaico

La S-function recoge el modelo matemático del panel, y contiene las ecuaciones que rigen su comportamiento. La parte más relevante del código es la declaración de las variables y las ecuaciones, como se puede ver en la figura 8-6.

En concreto, hay una serie de parámetros (tabla 8-1) que afectan a la configuración del panel. Modificando estos valores se puede conseguir la curva característica del panel que se desee, de manera que se puede customizar el panel que se desea obtener.

Tabla 8-1. Parámetros que configuran las curvas características de un panel

Parámetro	Significado
$N_s$	Número de células solares en serie
$N_p$	Número de células solares en paralelo
$V_{oc\_n}$	Tensión nominal a circuito abierto (V)
$I_{sc\_n}$	Corriente nominal de cortocircuito (A)

```

//Variables y significado
Ns = 1*54;      // [] - number of solar cells in a string in solar array
Np = 1;        // [] - number of solar cell strings in parallel solar array
Tn = 25+273.15; // [C] - nominal temperature of solar cells
Gn = 1000;     // [W/m^2] - nominal irradiation of solar cells
V_OC_n = 132.9; // [V]-nominal open circuit voltage of array (single panel)
I_SC_n = 8.21; // [A] - nominal short circuit current of array
Rs = 0.221;   // [ohm] - equivalent series resistance of array
Rp = 415.405; // [ohm] - equivalent parallel resistance of array
a = 1.3;      // [] - diode ideality constant
KV = -0.1230; // [V/K] - thermal voltage coefficient of array
KI = 0.0032;  // [A/K] - thermal current coefficient of array
C_in = 22e-6; // [F] - input capacitance of the DC/DC converter
k = 1.3806503e-23; // [J/K] - Boltzmann constant
q = 1.60217646e-19; // [C] - electron charge

//Ecuaciones de comportamiento
T_Kelvin = T + 273.15;
delta_T = T_Kelvin - Tn;
V_t = Ns*k*T_Kelvin/q;
I_PV_n = I_SC_n;
I_pv_cell = (I_PV_n + KI*delta_T) * (G/Gn);
I_pv_array = I_pv_cell * Np;
I_0_cell = (I_SC_n + KI*delta_T) / (exp((V_OC_n + KV*delta_T) / (a*V_t)) - 1);
I_0_array = I_0_cell * Np;
I_m = I_pv_array - I_0_array * (exp((V + Rs*I) / (a*V_t)) - 1);
if (I_m < 0){
    I_m = 0; // helps solver avoid setting I_m to -inf during minor step
}
    
```

Figura 8-6. S-Function que contiene el modelado matemático del panel

### 8.2.3 Dimensionado del campo fotovoltaico

Para poder definir tanto la curva característica de un panel como el número de paneles y su conexión, se deben conocer los valores de tensión, potencia y corriente que se desean obtener del campo solar para la aplicación que se va a simular. Los valores que se quieren conseguir son (para una irradiancia de 1000 W/m<sup>2</sup> y una temperatura de 25°C):

- Potencia de pico superior a los 20kW e inferior a los 25kW
- Tensión de máxima potencia superior a los 700V e inferior a los 750V
- Tensión de circuito abierto superior a los 750 e inferior a los 850V

#### 8.2.3.1 Caracterización de 1 único panel

Para estudiar cuántos paneles se deben colocar en serie y en paralelo, primero se debe estudiar el comportamiento de un solo panel. Para obtener la curva característica deseada del panel los parámetros de configuración de las células solares del panel deben tener los valores recogidos en la tabla 8-2.

Tabla 8-2. Configuración de las células solares de un panel

Parámetro	Valor
Ns	170
Np	1
V_oc_n	400 V
I_sc_n	11 A

Para representar las curvas características, se conecta una fuente de tensión regulable a la salida del panel, que recorre una rampa de valores. También se coloca un switch que inicialmente está abierto, de manera que el panel carga el condensador a la tensión de circuito abierto. Las curvas que se obtienen se representan en la figura 8-7, y las características más relevantes se recogen en la tabla 8-3.

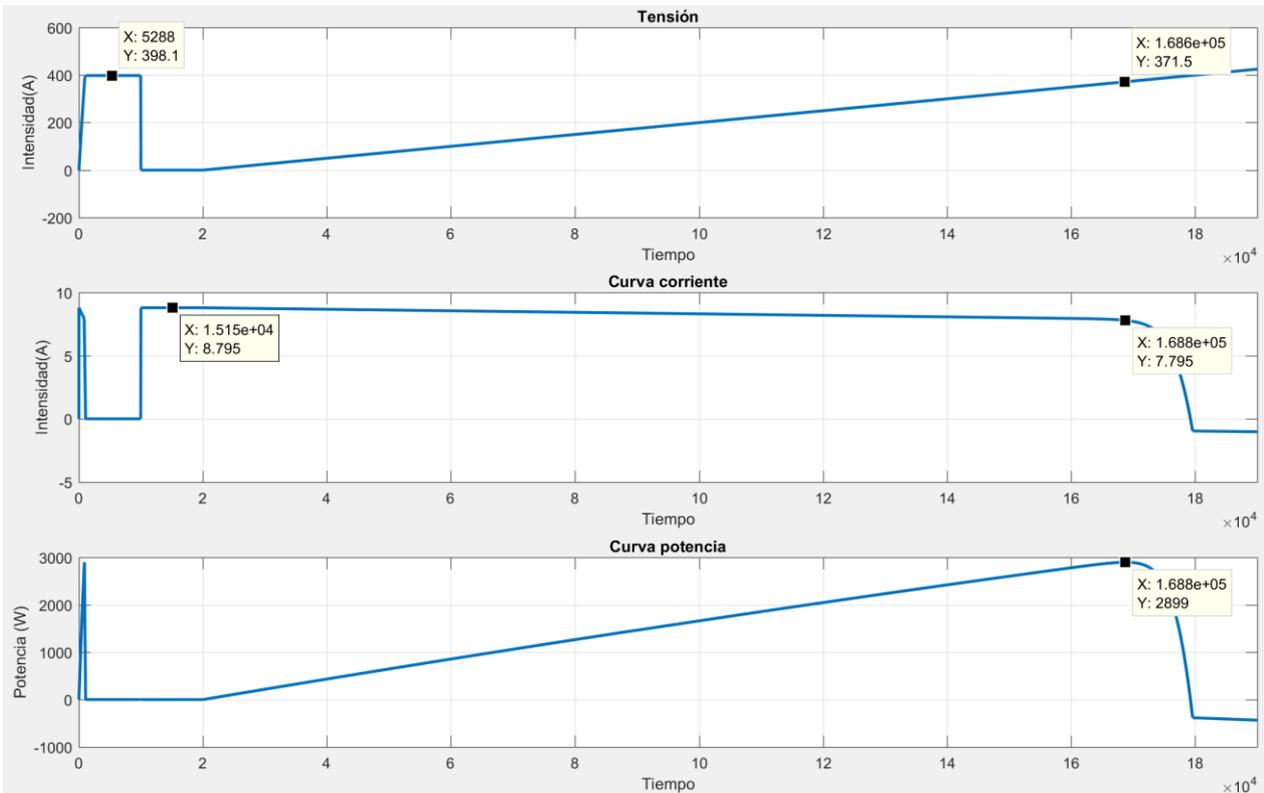


Figura 8-7. Curvas características de un solo panel PV

Tabla 8-3. Caracterización de un solo panel

Magnitud	Valor
Corriente de cortocircuito	8.795 A
Tensión del punto de máxima potencia	371.5 V
Máxima potencia	2899 W
Tensión a circuito abierto	398.1 V
Intensidad del punto de máxima potencia	7.795 A

### 8.2.3.2 Caracterización del campo solar completo

A continuación se estudia el número de paneles en serie y paralelo (figura 8-8) que se deben colocar para obtener la curva característica del campo solar que cumpla con los requisitos especificados al inicio del apartado.

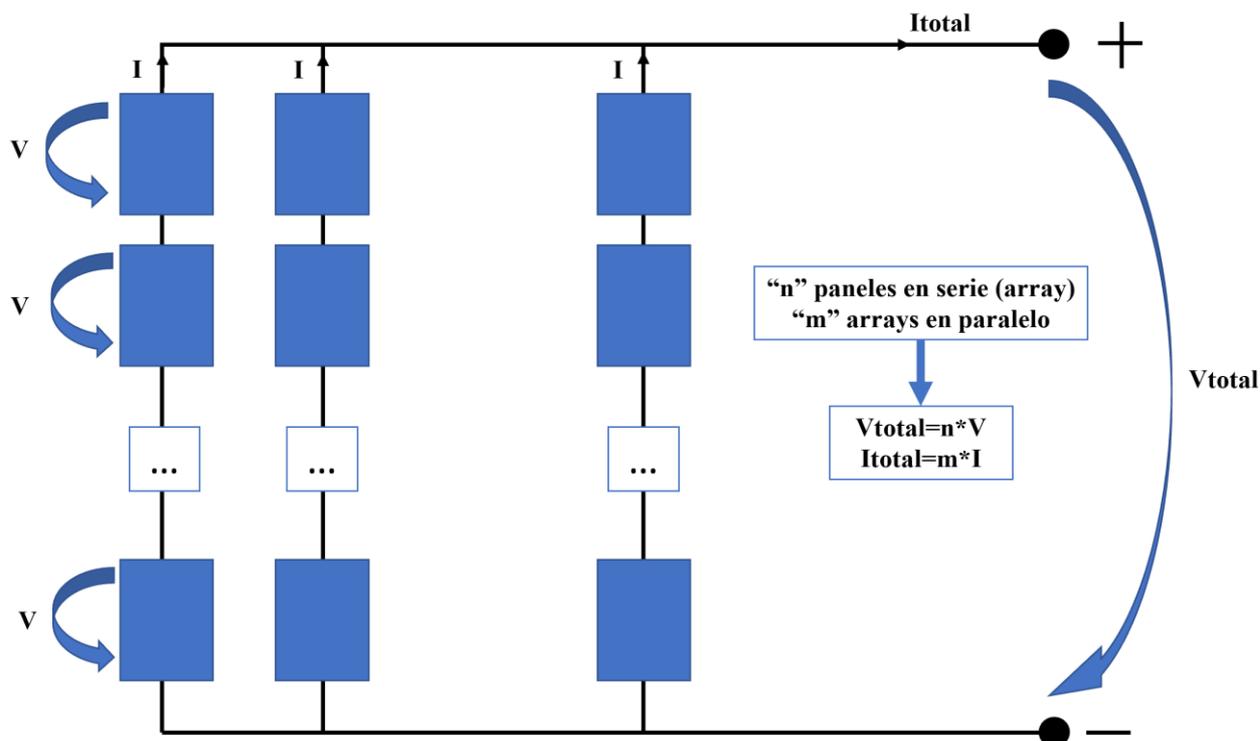


Figura 8-8. Ecuaciones del campo solar

Aplicando las ecuaciones de la figura 8-8-, para una configuración de 2 paneles por array y 3 arrays en paralelo, se consiguen las características recogidas en la tabla 8-4.

$$V_{mp,total} = 2 * V_{mp} = 2 * 371.5 = 743 \text{ V} \in [700, 750] \text{ V}$$

$$V_{oc,total} = 2 * V_{oc} = 2 * 398.1 = 796.2 \text{ V} \in [750, 850] \text{ V}$$

$$I_{mp,total} = 3 * I_{mp} = 3 * 7.795 = 23.385 \text{ A}$$

$$I_{sc,total} = 3 * I_{sc} = 3 * 8.795 = 26.385 \text{ A}$$

$$P_{pico} = V_{mp,total} * I_{mp,total} = 743 * 23.385 = 17.375 \text{ kW} \sim 20 \text{ kW}$$

Tabla 8-4. Caracterización del panel solar completo

Magnitud	Valor
Corriente de cortocircuito	26.385 A
Tensión del punto de máxima potencia	743 V
Máxima potencia	17.375 kW
Tensión a circuito abierto	796.2 V
Intensidad del punto de máxima potencia	23.385 A

Se comprueba que esta configuración cumple todos los requisitos. Para representar las curvas características, se repite la estructura del apartado anterior, un switch ideal y una fuente de tensión regulable que recorre una rampa de tensiones.

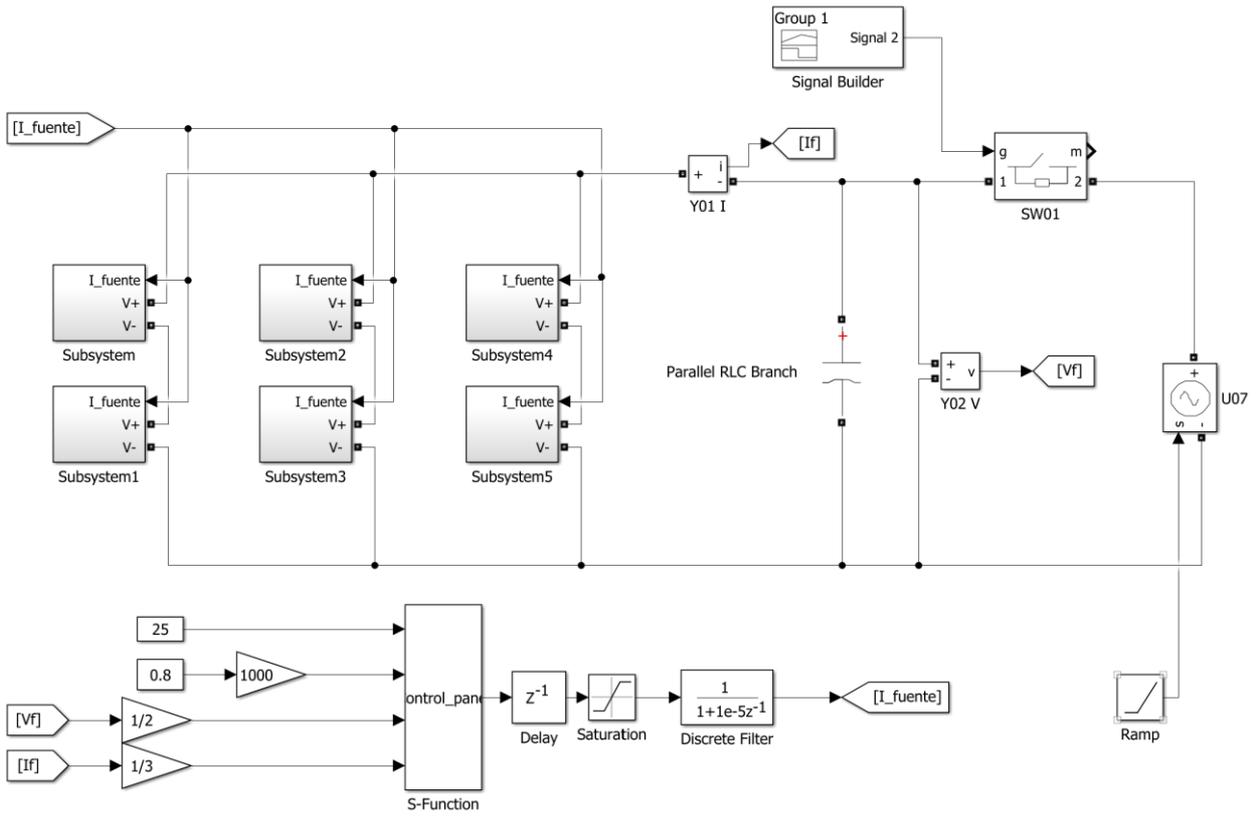


Figura 8-9. Modelo simulink del campo solar

Las curvas características que se obtienen que representan en la figura

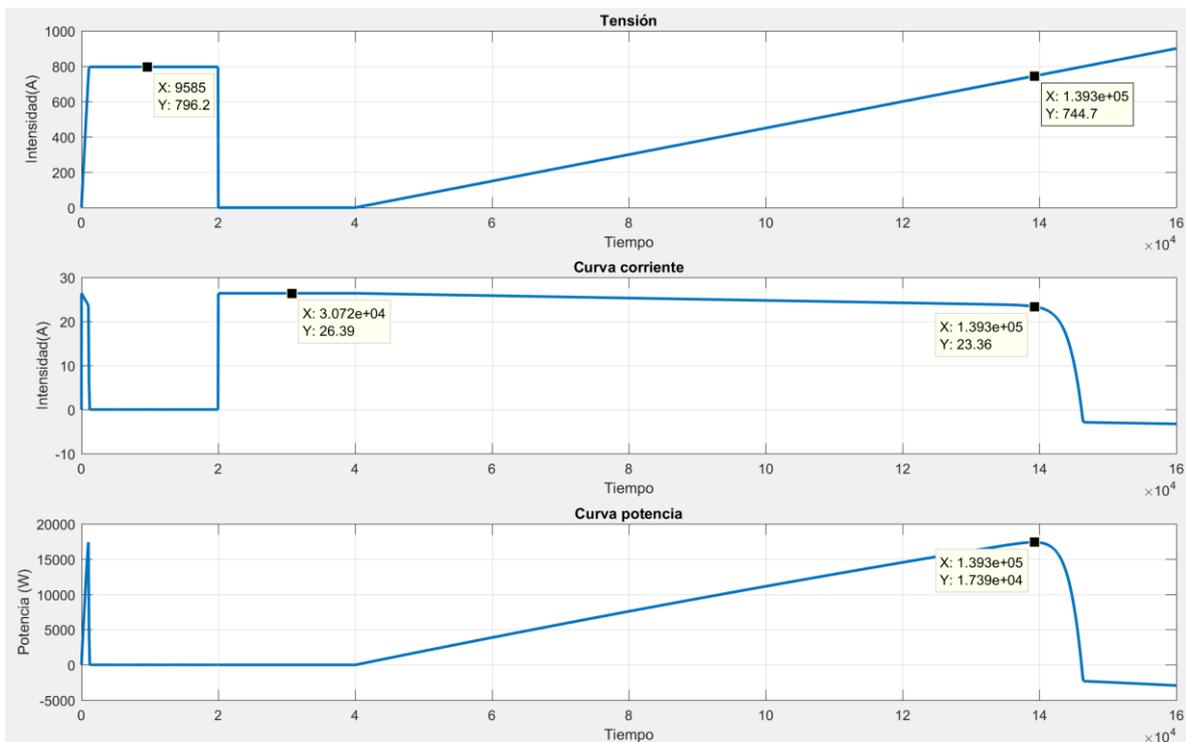


Figura 8-10. Curvas características del campo solar

Este modelo de campo solar se puede implementar en la FPGA de la OPAL, y se comprueba que funciona

correctamente. Por tanto, se ha conseguido modelar e implementar un campo fotovoltaico en la FPGA de la plataforma de simulación en tiempo real OPAL 4510. Este campo solar servirá para modelar una fuente de energía renovable (FER) y probar el control de convertidores de potencia que conecten dicha fuente a la red en un sistema de simulación en tiempo real.



## 9 CONCLUSIONES Y TRABAJOS FUTUROS

En este último capítulo se repasan brevemente los problemas de implementación que han ido apareciendo a lo largo del desarrollo del trabajo. También se proponen posibles soluciones o líneas de investigación a seguir trabajando a la finalización de este proyecto.

### 9.1 Medidas y calibración de sensores

Como se comentó en las secciones 5.3.1 y 6.2, la toma de medidas de la placa del DSP no funciona correctamente, sino que existen efectos no lineales que hacen que la medida se distorsione, y el sensor no se pueda calibrar utilizando una recta de regresión, que es el procedimiento habitual.

La solución que se le ha dado a esta cuestión ha sido ajustar los puntos no con una recta de regresión, sino con un polinomio de mejor ajuste, de grado superior a uno. Esto, junto con varias calibraciones sucesivas manuales, permiten obtener medidas lo suficientemente exactas como para permitir que el algoritmo de control se ejecute.

Sin embargo, se trata de una solución temporal, que ha permitido continuar con el desarrollo del proyecto, pero que debe mejorarse. Por eso, una de las líneas de trabajo futuro sobre este proyecto es precisamente estudiar cada una de las etapas de adaptación por las que pasa la señal, localizar el origen (u orígenes) de estos efectos no lineales y solucionarlos.

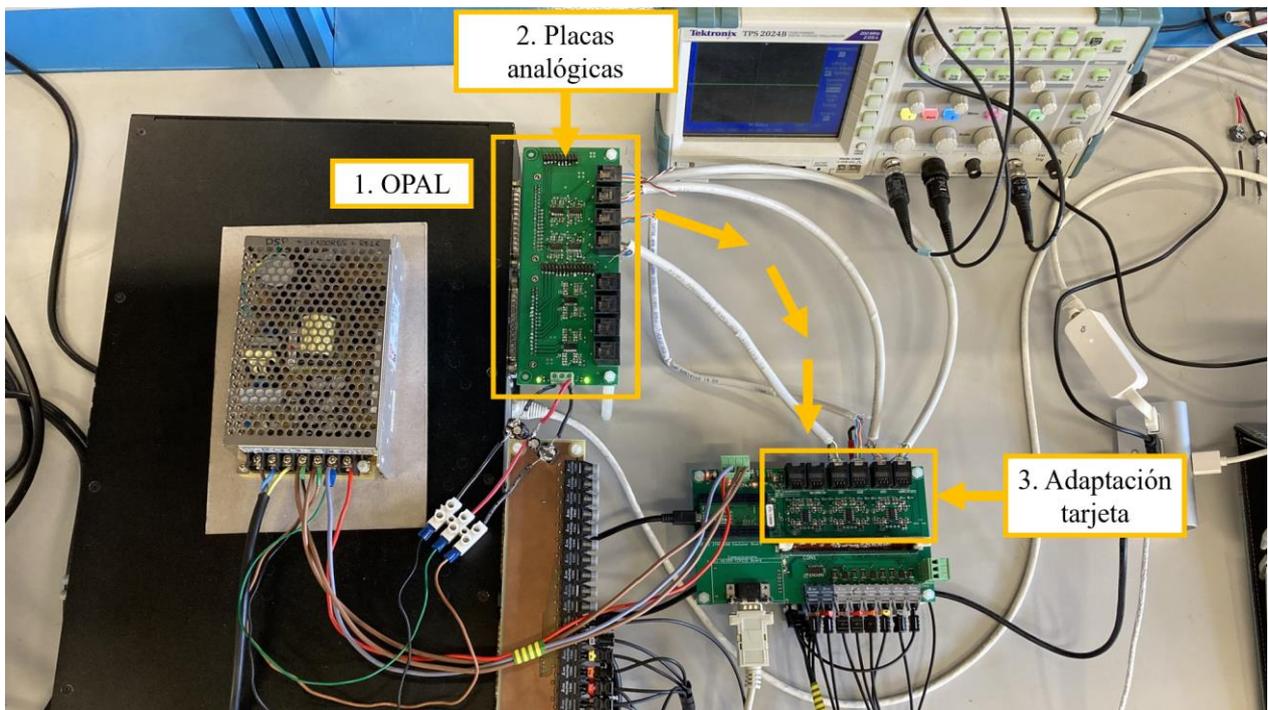


Figura 9-1. Sistema eléctrico a implementar en la FPGA

La señal pasa por 3 etapas de adaptación:

1. **Ganancia OPAL:** se refiere a la ganancia que aplica la OPAL a las señales de salida analógicas para que se puedan representar en un rango de  $\pm 16V$ . Es una adaptación interna de la OPAL, y se ha comprobado empíricamente que la ganancia se aplica de manera exacta. Por tanto, esta ganancia no parece que sea el origen de los problemas.
2. **Placas analógicas:** esta placa es propia del departamento, por lo que podría estar sujeta a distintos

errores de diseño, como por ejemplo el control del ruido que llega a los operacionales a través de la alimentación. Por eso, esta placa se analizará con detalle a la finalización del proyecto, para detectar cualquier no linealidad que pudiera introducir en el camino de la señal.

3. **Etapa de entrada de adaptación de la tarjeta:** al igual que la etapa anterior, también se trata de un circuito de diseño propio, pero esta misma tarjeta está en uso en convertidores de potencia reales del laboratorio, cuya adquisición de medidas funciona correctamente. Por tanto, aunque también se revisará y estudiará con cuidado, se tiene la certeza de que el mismo diseño está funcionando correctamente en otro sistema.

Por tanto, se deben estudiar todas estas etapas de adaptación y solucionar el problema, ya que una toma de medidas incorrecta impide que cualquier algoritmo de control en bucle cerrado funcione correctamente.

## 9.2 Medidas y calibración de sensores

Otro mal funcionamiento de este sistema HIL se mencionó en la sección 7.4, en la que se explica que un modelo eléctrico implementado en la FPGA no tiene el comportamiento esperado.

Como no se conoce exactamente cómo se ejecuta el modelo en la FPGA, no se sabe con exactitud cuál puede ser la causa de este comportamiento erróneo. Sin embargo, una modificación que sufren las señales de control en la realidad pero que no se ha modelado en estos sistemas son los tiempos muertos.

Los tiempos muertos son un retraso que se añade al encendido de los IGBTs para asegurar que los dos transistores de una misma rama no estén encendidos a la vez, y no cortocircuiten la fuente de tensión (figura 9-2). Añadir los tiempos muertos es otra de las funciones de la placa driver (figura 9-2), que completa lo que se explicó en la sección 5. Sin embargo, es un efecto que no se ha tenido en cuenta y que no se ha modelado.

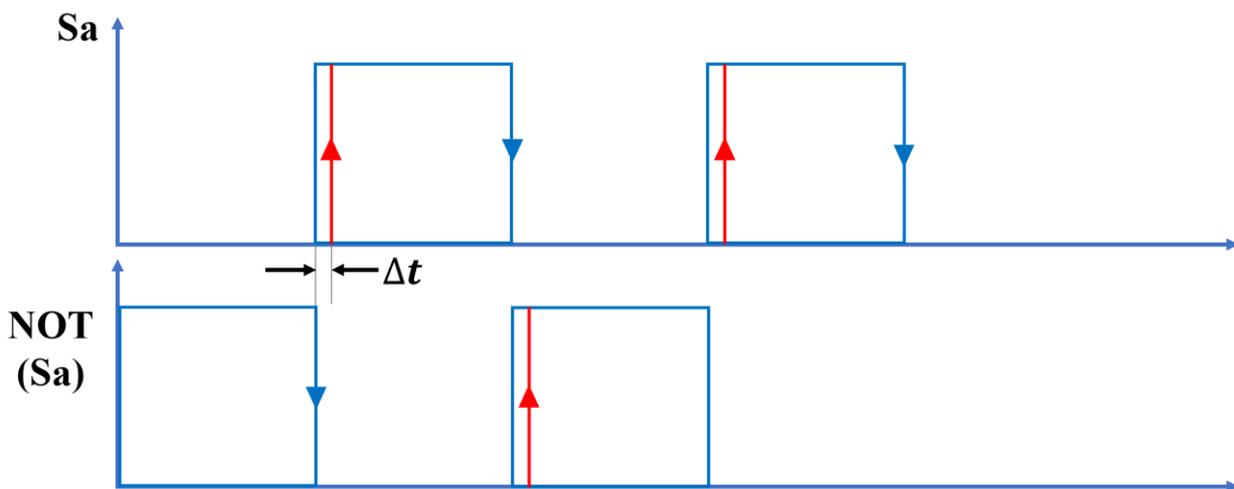


Figura 9-2. Tiempos muertos de las señales de disparo

Quizás el motivo de que el modelo eléctrico se ejecute mal en la FPGA es que no se han tenido en cuenta estos tiempos muertos, pero la FPGA necesita que se modelen por la manera que tiene de ejecutar el sistema eléctrico. Eso es lo que se pretende averiguar en las pruebas que se harán a la finalización de este proyecto.

Para introducir el efecto de los tiempos muertos, se proponen dos soluciones, que se probarán a la finalización de este trabajo de fin de grado.

### 9.2.1 Uso de bloques propios de la librería RT-LAB

En primer lugar, una posible solución es, en lugar de utilizar los bloques IGBTs propios de Simulink, Matlab (librería Simscape, Power Systems, Specialized technology), utilizar los bloques propios de RT-LAB para modelar los convertidores de potencia.

En un primer lugar se utilizaron los bloques de Simscape por simplicidad, y porque se había trabajado previamente con ellos. Además, permiten construir el convertidor de potencia de manera personalizable, ya que los bloques de RT-LAB proporcionan convertidores de potencia ya definidos.

Teóricamente, no debería haber problema en utilizar los bloques que se han implementado, porque la RT-LAB los reconoce y compila sin problema, pero ante el mal funcionamiento del modelo, la alternativa es probar los bloques propios.

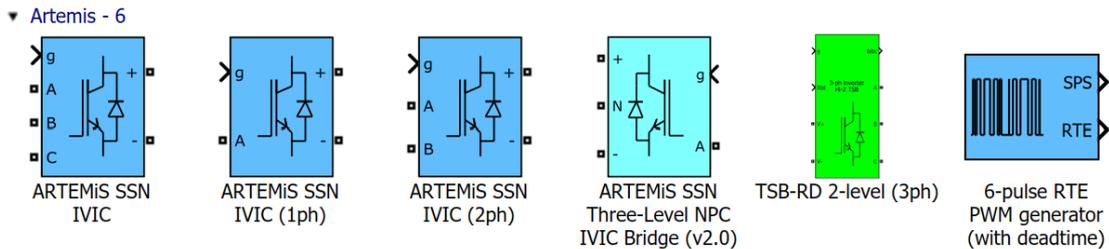


Figura 9-3. Convertidores de potencia propios del software RT-LAB

### 9.2.2 Diseño de un hardware adicional

Otra posible solución es diseñar y construir un hardware de adaptación intermedio entre el DSP y las entradas digitales de la OPAL, que realizara las funciones propias del driver. El esquema lógico que debería implementar esta nueva placa sería el que se muestra en la figura 9-4.

Esto no sólo resolvería la introducción de los tiempos muertos en el modelo, sino que también solucionaría otras limitaciones que se comentaron en el apartado 5. Por ejemplo, este tratamiento de la señal de enable y de las señales de disparo haría posible tener un estado en el que todos los IGBTs estuvieran abiertos.

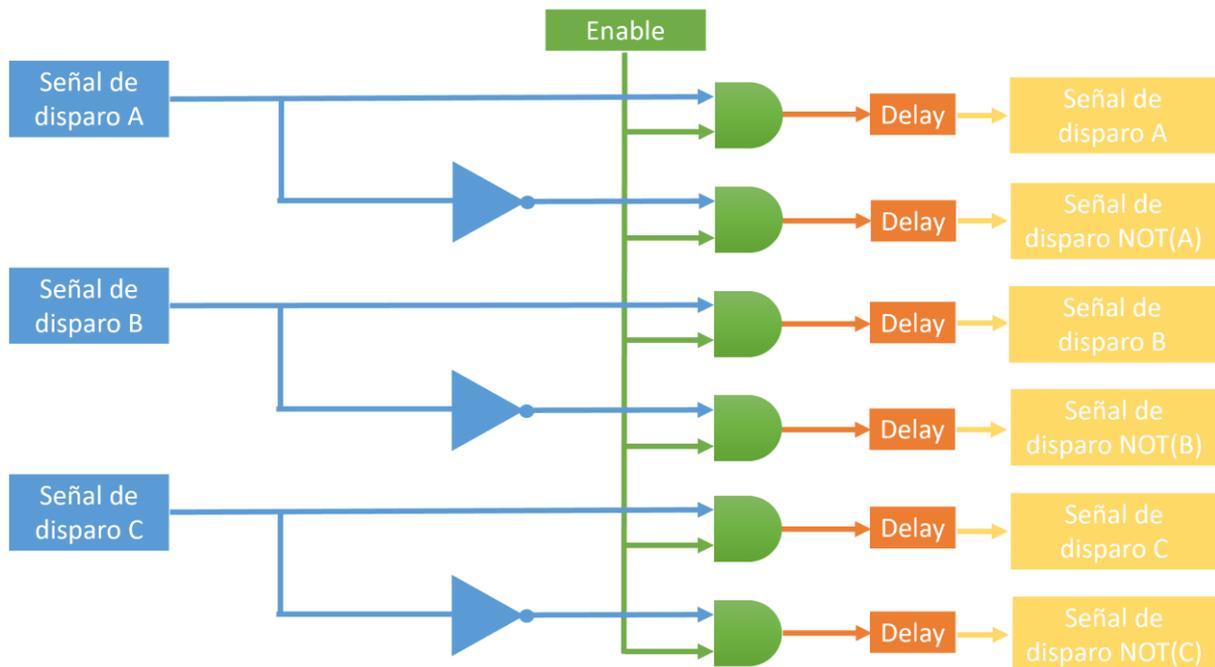


Figura 9-4. Funciones lógicas que se deben implementar en la placa driver

Sin embargo, esto no es una solución óptima, ya que supondría tener que diseñar una placa de adaptación ligeramente diferente para cada convertidor de potencia que se quisiera simular, adaptándose a su topología y entradas y salidas. Además, es una solución más cara, ya que construir este hardware de adaptación lleva un coste asociado.

### 9.3 Conclusiones

Este trabajo de fin de grado tenía como objetivo la puesta en marcha de un sistema de simulación HIL para probar algoritmos de control MPPT sobre campos solares conectados a la red a través de un convertidor de potencia.

Por una parte, se ha conseguido conectar todas las plataformas y elementos que conforman el sistema de simulación HIL, integrando en el montaje las placas de adaptación desarrolladas por el departamento. También se ha establecido la comunicación entre todos los elementos de manera exitosa, y se han implementado determinados modelos eléctricos en la FPGA de la OPAL, que, junto con la implementación del algoritmo de control en el hardware de control externo, han permitido ejecutar sus simulaciones en tiempo real.

Sin embargo, queda pendiente continuar el trabajo en el futuro, para mejorar algunas características y solucionar determinados problemas. Por una parte, es necesario mejorar la comunicación a través de los canales analógicos desde la OPAL hacia el DSP, estudiando las etapas de adaptación intermedias y asegurando que todas las transformaciones sean lineales.

Por otra parte, también queda pendiente estudiar y solucionar el mal comportamiento del modelo dc-link conectado a red, encontrar la causa y probar las distintas soluciones propuestas en el apartado 9.2. Una vez que este modelo funcione correctamente, y haciendo uso de los paneles solares ya modelados e implementados satisfactoriamente en la FPGA, queda pendiente montar el modelo completo y probar los algoritmos MPPT, que era el objetivo final de este trabajo de fin de grado.

En conclusión, se ha desarrollado gran parte del trabajo en cuanto a la integración de los distintos elementos, la comunicación entre ellos y la puesta en marcha del sistema completo, pero queda pendiente la implementación de modelos y algoritmos de control más complejos, que permitan sacar el máximo partido al sistema de simulación en tiempo real ya operativo.

# REFERENCIAS

---

- [1] C. Kleijn , «Introduction to Hardware-in-the-Loop Simulation», Controllab
- [2] Jason Benfer, «What is Hardware-in-the-Loop (HIL) Testing?», *Genuen, ideas taking flight*, 2018
- [3] S.Raman, N. Sivashankar, W. Milam, W. Stuart, and S. Nabi, «Design and Implementation of HIL Simulators for Powertrain Control System Software Development», *Proceedings of the American Control Conference*, 1999
- [4] T. Boge, T.Wimmer, O.Ma, M.Zebenay, «EPOS - A Robotics-Based Hardware-in-the-Loop Simulator for Simulating Satellite RvD Operations», *i-SAIRAS 2010*
- [5] Tor A. Johansen, Thor I. Fossen, Bjørnar Vik, «Hardware-in-the-loop Testing of DP systems», *Dynamic positioning conference*, 2005
- [6] Jerónimo Chacopino García, «Algoritmos Flexibles de Seguimiento de Potencia para Sistemas Fotovoltaicos», Trabajo de Fin de Grado, 2020
- [7] Pagina web OPAL
- [8] System Description, OPAL-RT <https://www.opal-rt.com/>