

**UNIVERSIDAD AUTÓNOMA DE MADRID**

**ESCUELA POLITÉCNICA SUPERIOR**



**TRABAJO FIN DE MÁSTER**

# **Control digital de convertidores conmutados mediante LabVIEW**

**Máster Universitario en Ingeniería de  
Telecomunicación**

**María Murillo Moya**  
**Tutor: Alberto Sánchez González**  
**Ponente: Ángel de Castro Martín**

**Septiembre 2017**



# **CONTROL DIGITAL DE CONVERTIDORES CONMUTADOS MEDIANTE LABVIEW**

**Autora: María Murillo Moya**

**Tutor: Alberto Sánchez González**

**Ponente: Ángel de Castro Martín**

Trabajo realizado en el grupo

# **HCTLab**

Hardware and Control Technology Laboratory

Escuela Politécnica Superior

Universidad Autónoma de Madrid

**Septiembre 2017**





## ***Agradecimientos***

Llegando al final de este verano, se cierra por partida doble otra gran etapa de mi vida en la que solo tengo palabras de agradecimiento a todas las personas que, de una forma u otra, han permitido que hoy escriba estas líneas.

En primer lugar, repito una vez más a mi tutor, Alberto Sánchez, gracias de nuevo por toda la ayuda, dedicación y recursos e ideas sin las cuales no habría sido posible terminar a tiempo este Trabajo Fin de Máster, así como a mi ponente, Ángel de Castro. También agradecer a los demás integrantes del HCTLab por haber formado parte del grupo y a los profesores de la Escuela Politécnica Superior que a lo largo de esta etapa no solo me han formado a nivel educativo, sino también personal.

A todos mis compañeros de universidad, que han hecho que estos años siempre los recuerde con una sonrisa y llenos de buenos momentos dentro y fuera de la EPS. A Miriam, compañera y amiga desde el primer día de universidad, además del *Chanchu Team*, Adri, Mayte, Ryan, Sandra, Laura, Alberto y Edu.

No me olvido de mis inicios en la vida laboral. Gracias al grupo de becarios CESA 2015-2017 por hacer que fuese un año tan increíble. A toda la Unidad Electrónica, encantada de haber sido parte de ella y aprender tanto. A Juan Antonio por ser mi tutor, y a *Andrew*. A mi brazo izquierdo, CCC, gracias por todos los momentos (tan... dulces) compartidos, así como mis dos chicas de Administración, Anahí y mi tocaya. Además de Miguel Ángel, por ser el primero en recibirme cada mañana con una sonrisa.

Cómo no voy a nombrar a Alberto, Álvaro, Cristian, Nico y Paz. No podía imaginar unos meses más exóticos sin vuestra compañía recorriendo las islas afortunadas desde el mar hasta las alturas. Así como a todo el *Instrumentation Team* del IAC por hacer más amenas las horas en la sala AIV y a los *afterwork* más divertidos con los talleres de Nau.

A mi gente de Tarancón, por los buenos momentos, risas, días y noches de fiestas. También especial recuerdo al *secondo piano* de Milán por aquel maravilloso año en *Casa dello Studente* y el *Politecnico*.

Por último, agradecer a mis padres su apoyo y amor incondicional. Han sido vuestros ánimos los que han hecho que siempre siguiese hacia adelante con todo, especialmente estos años de Máster. Os quiero y os necesito.

También a mi *sorellina* Laura y mi hermano Alexis, sois esenciales para mí y estoy orgullosa de teneros como ejemplos a seguir y junto a mí. A mis abuelos y tío Eduardo, que siempre estaréis presentes en cada éxito de mi vida. Y a la nueva familia numerosa, con Darío y los mellizos, Héctor y Rubén. En breve regresaré, para no perderme vuestros primeros pasos mientras doy mis últimas pisadas en esta etapa y seguir creciendo juntos.

María Murillo Moya

Septiembre 2017

# RESUMEN

En la electrónica de potencia, los convertidores de potencia transforman la energía eléctrica que reciben a la entrada en otra con diferentes características a su salida. En particular, los convertidores conmutados presentan numerosas ventajas frente a los lineales en términos de eficiencia, rendimiento y funcionalidad. Para el control digital de un convertidor conmutado se utilizan elementos semiconductores que actúan como interruptores para permitir el paso de la corriente.

En la transformación de energía de corriente alterna a corriente continua (CA a CC) es necesario realizar una corrección del factor de potencia (f.d.p.) para evitar introducir armónicos a la red eléctrica. Una de las técnicas para realizar la corrección del f.d.p. es la implementación de un doble lazo con un lazo de tensión y un lazo de corriente con sus respectivos reguladores.

El principal objetivo de este Trabajo Fin de Máster es realizar un control digital sobre un convertidor conmutado con el software de LabVIEW, un entorno de desarrollo integrado que permite una mejor integración de los dispositivos hardware y numerosas funcionalidades mediante una programación gráfica.

A lo largo de este documento, han sido comparadas las características de dos posibles dispositivos para realizar la corrección del f.d.p para desarrollar un control digital sobre un convertidor elevador de tensión (tipo *boost*). Se hallaron características críticas para la selección del dispositivo y finalmente se eligió el dispositivo reprogramable NI myRIO, que incluye una FPGA (*Field Programmable Gate Array*), cuya programación se realiza a través de una interfaz de usuario desarrollada con LabVIEW que permite el control digital del convertidor *boost* con doble lazo para implementar la corrección del f.d.p.

El trabajo concluye con una serie de resultados experimentales de corrección del f.d.p. sobre el convertidor *boost* junto con el dispositivo NI myRIO y la interfaz desarrollada en LabVIEW. Se han probado ambos lazos de control, comprobando el funcionamiento de los mismos tanto en régimen permanente como ante transitorios.

Por último, se ha realizado un análisis de la adecuación de dicho software y NI myRIO en aplicaciones de control digital, así como la influencia de las características del convertidor utilizado en los resultados de las pruebas experimentales.





# ABSTRACT

In power electronics, the power converters transform the electric energy that receive at their entrance into another with different characteristics at their output. In particular, switching converters show many advantages with respect to linear converters in terms of efficiency, performance and functionality. For the digital control of a switching converter, semiconductor elements are used acting as switches to allow the current flow.

In the case of transforming alternating current into direct current (AC to DC) it is necessary to perform a Power Factor Correction (PFC) to avoid introduce harmonics into the electrical grid. One of the techniques to perform the PFC is the implementation of a double loop with a voltage loop and a current loop with their respective regulators.

The main goal of this Project is to perform a digital control over a switching converter with the software of LabVIEW, an integrated development environment that allows a better integration of power devices and numerous functionalities through a graphical programming.

Throughout this document, the characteristics of two possible devices to perform the PFC have been compared to develop a digital control over a boost converter. Critical features were found and finally NI myRIO reprogrammable device was chosen, that includes an FPGA (*Field Programmable Gate Array*), which is programmed through a user interface developed with LabVIEW that allows to perform the digital control over the boost converter with double loop to implement the PFC.

This Project concludes with a series of experimental results of PFC over the boost converter together with the NI myRIO device and the interface developed in LabVIEW. Both control loops have been tested, verifying the operation in steady state and transients.

Finally, an analysis was made of the suitability of the LabVIEW software and NI myRIO in digital control applications, as well as the influence of the characteristics of the boost converter used in the results of the experimental tests.



## PALABRAS CLAVE

Control digital, convertidor conmutado, LabVIEW, *Field Programmable Gate Array*, myRIO, corrección del factor de potencia.

## KEYWORDS

Digital control, switching converter, LabVIEW, *Field Programmable Gate Array*, myRIO, *Power Factor Correction*.



# ÍNDICE GENERAL

<b>AGRADECIMIENTOS</b> .....	<b>V</b>
<b>RESUMEN</b> .....	<b>VII</b>
<b>ABSTRACT</b> .....	<b>IX</b>
<b>PALABRAS CLAVE</b> .....	<b>XI</b>
<b>KEYWORDS</b> .....	<b>XI</b>
<b>ÍNDICE GENERAL</b> .....	<b>XIII</b>
<b>ÍNDICE DE FIGURAS</b> .....	<b>XV</b>
<b>ÍNDICE DE TABLAS</b> .....	<b>XIX</b>
<b>GLOSARIO</b> .....	<b>XXI</b>
<b>1. INTRODUCCIÓN Y MOTIVACIÓN</b> .....	<b>23</b>
1.1. Objetivos.....	25
1.2. Estructuración de la memoria .....	26
<b>2. ESTADO DEL ARTE EN CONVERTIDORES CONMUTADOS</b> .....	<b>27</b>
2.1 Topologías de convertidores conmutados más comunes .....	28
2.2 Modos de operación según el tipo de conducción .....	29
2.3. Factor de potencia .....	29
2.3.1. Corrección del Factor de Potencia.....	31
<b>3. LABVIEW</b> .....	<b>33</b>
3.1 NI myDAQ.....	35
3.2 NI myRIO .....	37
<b>4. REQUISITOS Y SELECCIÓN DEL DISPOSITIVO</b> .....	<b>40</b>
4.1. Frecuencia de conmutación.....	40
4.2. Nivel de voltaje para las entradas analógicas .....	43
4.3. Tasa de muestreo de los ADCs.....	44
4.4. Resolución de los ADCs .....	45
4.5. Adquisición de muestras durante el ciclo de conmutación .....	45
<b>5. IMPLEMENTACIÓN Y PRUEBAS</b> .....	<b>46</b>
5.1 Implementación.....	46
5.1.1. Comunicación entre PC y myRIO .....	46

5.1.2. Programación de myRIO .....	48
5.1.3. Sensado, control y actuación en paralelo .....	51
5.1.4. Almacenamiento y disponibilidad de datos .....	53
5.1.5. Calibraciones.....	57
5.1.6. Reguladores .....	63
<b>5.2. Descripción de la ejecución .....</b>	<b>71</b>
5.2.1. FPGA VI .....	71
Generación del PWM.....	71
Control de Lazo de tensión .....	73
Control de Lazo de corriente.....	76
5.2.2. Host VI .....	78
<b>6. RESULTADOS.....</b>	<b>81</b>
6.1. Integración de los dispositivos .....	81
6.2. Resultados experimentales .....	83
<b>7. CONCLUSIONES .....</b>	<b>90</b>
7.1 Líneas futuras .....	92
<b>BIBLIOGRAFÍA.....</b>	<b>93</b>
<b>ANEXO I .....</b>	<b>95</b>
I.1 Características de convertidor boost .....	95
<b>ANEXO II.....</b>	<b>96</b>
II.1 Calibración para Voltaje de entrada $V_{in}$ .....	96
II.2 Calibración para Corriente de entrada $I_{in}$ .....	99
<b>ANEXO III .....</b>	<b>101</b>
III.1 Implementación de planta del convertidor boost .....	101
III.2 Implementación de regulador para lazo de tensión .....	103
III.3 Implementación de regulador para lazo de corriente.....	104

# ÍNDICE DE FIGURAS

FIGURA 1: EJEMPLO DE CONVERTIDOR CONMUTADO CON RECTIFICACIÓN DE CORRIENTE A LA ENTRADA ..	27
FIGURA 2: CONVERTIDOR BUCK CON INTERRUPTOR $T_{ON}$ Y $T_{OFF}$ RESPECTIVAMENTE .....	28
FIGURA 3: CONVERTIDOR BOOST CON INTERRUPTOR $T_{ON}$ Y $T_{OFF}$ RESPECTIVAMENTE.....	28
FIGURA 4: RELACIÓN ENTRE LOS DIFERENTES TIPOS DE POTENCIA.....	30
FIGURA 5: CONFIGURACIÓN PARA PFC Y CONTROL DE DOBLE LAZO PARA CONVERTIDOR BOOST .....	32
FIGURA 6: EJEMPLO DE PANEL FRONTAL DE LABVIEW .....	34
FIGURA 7: EJEMPLO DE DIAGRAMA DE BLOQUES.....	34
FIGURA 8: PANEL FRONTAL DE ELVISMX .....	41
FIGURA 9: GENERADOR DE FUNCIONES EN ELVISMX .....	41
FIGURA 10: FUNCIÓN PWM DE LABVIEW FPGA DE NI MYRIO .....	42
FIGURA 11: GENERACIÓN INVÁLIDA DE PWM EN MYRIO .....	43
FIGURA 12: CIRCUITO RC.....	46
FIGURA 13: INTERACCIÓN ENTRE HOST VI Y FPGA VI.....	47
FIGURA 14: ESQUEMA DE GENERACIÓN DE ARCHIVO BITFILE (.BIT) .....	49
FIGURA 15: DIÁLOGO DE OPCIONES PARA COMPILACIÓN .....	50
FIGURA 16: INFORMACIÓN DE PROCESO DE GENERACIÓN DE FICHEROS INTERMEDIOS.....	50
FIGURA 17: INFORMACIÓN DE PROCESO DE ESTADO DE COMPILACIÓN .....	51
FIGURA 18: EJEMPLO DE SINCRONIZACIÓN DE BUCLES MEDIANTE OCURRENCIAS .....	52
FIGURA 19: EJEMPLO DE USO DE VARIABLES LOCALES .....	53
FIGURA 20: EJEMPLO DE FEEDBACK NODE Y SHIFT REGISTER.....	54
FIGURA 21: CONFIGURACIÓN DE MEMORIA BRAM (OPCIONES GENERALES).....	55
FIGURA 22: CONFIGURACIÓN DE MEMORIA BRAM (TIPO DE DATOS) .....	56
FIGURA 23: EJEMPLO DE ESCRITURA DE DATOS EN MEMORIA BRAM EN MYRIO .....	57
FIGURA 24: EJEMPLO DE LECTURA DE DATOS EN MEMORIA BRAM EN MYRIO .....	57
FIGURA 25: RECTA DE CALIBRACIÓN PARA $V_{OUT}$ EN MYRIO .....	59
FIGURA 26: RECTA DE CALIBRACIÓN PARA $V_{OUT}$ EN OSCILOSCOPIO.....	60
FIGURA 27: CALIBRACIÓN PARA $V_{OUT}$ EN FPGA VI (PANEL FRONTAL).....	61
FIGURA 28: CALIBRACIÓN PARA $V_{OUT}$ EN FPGA VI (DIAGRAMA DE BLOQUES) .....	61

FIGURA 29: EJEMPLO DE CONFIGURACIÓN EN FUNCIONES HIGH THROUGHPUT.....	63
FIGURA 30: VI DE FDT DIRECTA DEL MÓDULO FGPA EN LABVIEW.....	64
FIGURA 31: CONTROL EN FPGA VI DE MODELO FDT PARA DISCRETE TRANSFER FUNCTION DIRECT VI ..	65
FIGURA 32: PANEL DE CONFIGURACIÓN PARA TIPO DE DATO FXP.....	65
FIGURA 33: CONFIGURACIÓN DE LOS REGULADORES EN EL CONTROL DEL PANEL FRONTAL FPGA VI. ....	66
FIGURA 34: ERROR DE COMPILACIÓN CON REGULADORES IMPLEMENTADOS CON DISCRETE TRANSFER FUNCTION DIRECT VI.....	67
FIGURA 35: ESTIMACIÓN DE USO DE RECURSOS EN UN FPGA VI VACÍO.....	68
FIGURA 36: USO DE RECURSOS DE LA FPGA TRAS ADOPTAR TÉCNICAS DE OPTIMIZACIÓN.....	69
FIGURA 37: IMPLEMENTACIÓN DE ECUACIÓN EN DIFERENCIAS EN LABVIEW.....	70
FIGURA 38: USO DE RECURSOS DE LA FPGA FINAL.....	70
FIGURA 39: ACTIVACIÓN DE PULSO PWM Y DUTY CYCLE.....	72
FIGURA 40: CONTADOR PARA GENERACIÓN DEL PWM.....	72
FIGURA 41: GENERACIÓN DE HIGH Y LOW PERIOD TICKS PARA EL PWM.....	73
FIGURA 42: CONTROL DE LAZO DE TENSIÓN: LECTURA VREF.....	74
FIGURA 43: FUNCIÓN SELECT.....	75
FIGURA 44: CONTROL DE LAZO DE TENSIÓN: REGULADOR Y CONTROL DE LA ACTUACIÓN.....	76
FIGURA 45: CONTROL DE LAZO DE CORRIENTE: LECTURA VIN.....	77
FIGURA 46: CONTROL DE LAZO DE CORRIENTE: LECTURA IIN Y CALIBRACIÓN.....	77
FIGURA 47: CONTROL DE LAZO DE CORRIENTE: OBTENCIÓN DE CORRIENTE PARA REFERENCIA CON IIN....	78
FIGURA 48: CONTROL DE LAZO DE CORRIENTE: REGULADOR Y CONTROL DE LA ACTUACIÓN.....	78
FIGURA 49: EJEMPLO DE DIAGRAMA DE BLOQUES DE HOST VI.....	79
FIGURA 50: INTERFAZ FINAL DE USUARIO EN HOST VI.....	80
FIGURA 51: CONEXIÓN DE PINES EN MXP A MYRIO CON CONVERTIDOR BOOST.....	82
FIGURA 52: ESQUEMA DE INTERCONEXIÓN FINAL.....	82
FIGURA 53: MONTAJE FINAL PARA LAS PRUEBAS REALIZADAS EN EL LABORATORIO.....	82
FIGURA 54: ESCALÓN DE CARGA NEGATIVO DE $2k\Omega$ A $1k\Omega$ CON $V_{IN} = 20\text{ V}$ Y $V_{OUT} = 30\text{ V}$ .....	83
FIGURA 55: ESCALÓN DE CARGA POSITIVO DE $533,33\Omega$ A $688\Omega$ CON $V_{IN} = 20\text{ V}$ Y $V_{OUT} = 30\text{ V}$ .....	84
FIGURA 56: TRANSITORIO DE SUBIDA DE VOLTAJE DE $V_{REF} = 25\text{ V}$ A $V_{REF} = 33\text{ V}$ CON $V_{IN} = 20\text{ V}$ .....	85
FIGURA 57: POTENCIA DE ENTRADA Y POTENCIA Y TENSIÓN DE SALIDA CARACTERÍSTICOS DE UNA CORRECCIÓN DE F.D.P.....	85



FIGURA 58: TRANSITORIO DE BAJADA DE VOLTAJE DE $V_{REF} = 33\text{ V}$ A $V_{REF} = 25\text{ V}$ CON $V_{IN} = 20\text{ V}$ .....	86
FIGURA 59: RÉGIMEN PERMANENTE CON $V_{IN} = 26\text{ V}$ Y $V_{REF} = 38\text{ V}$ .....	87
FIGURA 60: RÉGIMEN PERMANENTE CON $V_{IN} = 30\text{ V}$ Y $V_{REF} = 45\text{ V}$ .....	87
FIGURA 61: INTERFAZ DE USUARIO PARA EL CONTROL DIGITAL DEL CONVERTIDOR CONMUTADO EN LABVIEW .....	88
FIGURA 62: RECTAS DE CALIBRACIÓN PARA $V_{IN}$ EN OSCILOSCOPIO Y MYRIO .....	97
FIGURA 63: CALIBRACIÓN PARA $V_{IN}$ EN FPGA VI (PANEL FRONTAL) .....	98
FIGURA 64: CALIBRACIÓN PARA $V_{IN}$ EN FPGA VI (DIAGRAMA DE BLOQUES) .....	98
FIGURA 65: RECTA DE CALIBRACIÓN PARA $I_{IN}$ EN MYRIO .....	99
FIGURA 66: CALIBRACIÓN PARA $I_{IN}$ EN FPGA VI (PANEL FRONTAL) .....	100
FIGURA 67: CALIBRACIÓN PARA $V_{IN}$ EN FPGA VI (DIAGRAMA DE BLOQUES) .....	100
FIGURA 68: RESPUESTA AL IMPULSO DEL REGULADOR DEL LAZO DE TENSIÓN .....	103
FIGURA 69: RESPUESTA AL IMPULSO DEL REGULADOR DEL LAZO DE CORRIENTE .....	104



# ÍNDICE DE TABLAS

TABLA 1: RANGOS DE MEDIDA DISPOSITIVO MYDAQ .....	37
TABLA 2: DATOS CALIBRACIÓN VOLTAJE DE SALIDA VOUT EN MYRIO .....	58
TABLA 3: DATOS CALIBRACIÓN VOLTAJE DE SALIDA VOUT EN OSCILOSCOPIO LEÍDOS TRAS DIVISORES RESISTIVOS .....	59
TABLA 4: DATOS PARA AJUSTE DE OFFSET DE VOUT .....	60
TABLA 5: CONFIGURACIÓN DE TIPO DE DATO FXP DE LOS REGULADORES.....	66
TABLA 6: PARÁMETROS DE DISEÑO DEL CONVERTIDOR BOOST ESCOGIDO .....	95
TABLA 7: DATOS CALIBRACIÓN VOLTAJE DE ENTRADA VIN EN OSCILOSCOPIO Y MYRIO.....	96
TABLA 8: DATOS PARA AJUSTE DE OFFSET DE VIN .....	97
TABLA 9: DATOS CALIBRACIÓN CORRIENTE DE ENTRADA IIN Y VIN EN RSHUNT EN MYRIO .....	99
TABLA 10: ECUACIÓN, PARÁMETROS Y VALORES PARA MODELOS DE PLANTAS DEL CONVERTIDOR BOOST .....	101



# GLOSARIO

<b>ADC</b>	Analog to Digital Converter
<b>BRAM</b>	Block Random Access Memory
<b>CA</b>	Corriente Alterna
<b>CC</b>	Corriente Continua
<b>CCM</b>	Continuous Conduction Mode
<b>DAC</b>	Digital to Analog Converter
<b>DC</b>	Direct Current
<b>DCM</b>	Discontinuous Conduction Mode
<b>DMA</b>	Direct Memory Access
<b>DRAM</b>	Dynamic Random Access Memory
<b>FDT</b>	Función de transferencia
<b>FIFO</b>	First In First Out
<b>FP</b>	Factor de Potencia
<b>FPGA</b>	Field Programmable Gate Array
<b>FXP</b>	Fixed-Point
<b>HDL</b>	Hardware Language Description
<b>HIL</b>	Hardware In the Loop
<b>IEC</b>	International Electrotechnical Commission
<b>I/O</b>	Input/Output
<b>LDR</b>	Lugar de las raíces
<b>LUT</b>	Look-Up Table
<b>LVTTTL</b>	Low Voltage Transistor-Transistor Logic
<b>NI</b>	National Instruments
<b>PCB</b>	Printed Circuit Board
<b>PFC</b>	Power Factor Correction
<b>PWM</b>	Pulse Width Modulation
<b>RIO</b>	Reconfigurable Input Output
<b>RT</b>	Real Time
<b>SISO</b>	Single-Input Single-Output
<b>TFM</b>	Trabajo Fin de Máster

<b>UART</b>	Universal Asynchronous Receiver/Transmitter
<b>VHDL</b>	VHSIC Hardware Description Language
<b>VI</b>	Virtual Instrument

# 1. INTRODUCCIÓN Y MOTIVACIÓN

El control digital en la electrónica de potencia ha experimentado una gran evolución en las últimas décadas en la aplicación de convertidores conmutados. Las principales ventajas que presentan los convertidores conmutados frente a los lineales son su mayor eficiencia y elevado rendimiento, menor coste y peso y mejores prestaciones en cuanto a funcionalidades [1]. Existen numerosas ventajas que motivan la realización de un control digital frente a uno analógico:

- Su implementación se puede realizar en dispositivos reprogramables como son las FPGAs, que permiten una gran flexibilidad en el diseño de un sistema de control.
- La facilidad para la realización de cambios del control digital mediante software. La modificación de un control analógico supone un rediseño a nivel de PCB (*Printed Circuit Board*).
- Menor uso de componentes analógicos y, por tanto, menor peso del diseño final.
- Disminución de los tiempos de diseño debido a la facilidad de modificaciones que implica el control digital.
- Mejor integración en otros sistemas debido a la capacidad de reconfiguración.
- Menores costes de mantenimiento por la mayor vida útil e inmunidad a condiciones ambientales de los componentes digitales.

Sin embargo, también se pueden encontrar algunos inconvenientes de la realización de un control digital frente al analógico tales como una mayor complejidad, pérdida de resolución de medida en las señales puesto que depende de las prestaciones del dispositivo digital o retardos por las etapas de conversión analógico-digital.

Los convertidores conmutados permiten transformar energía eléctrica con unas características de entrada a otras diferentes en su salida, como cambios de CA a CC y viceversa, elevación (*boost*) o reducción (*buck*) del voltaje suministrado, inversión de la tensión, etc.

Para la implementación de un control digital de un convertidor conmutado se utilizan dispositivos semiconductores que actúan a modo de interruptores para permitir el paso de la corriente. El control de los elementos de conmutación puede realizarse mediante una técnica llamada PWM (*Pulse Wide Modulation*). Una señal PWM permite controlar el porcentaje de tiempo que está o no conduciendo el elemento conmutador (comúnmente llamado ciclo de

trabajo o *duty cycle* en inglés) mientras que la frecuencia de conmutación del convertidor permanece constante.

El cálculo del ciclo de trabajo del PWM se suele basar en las medidas obtenidas de la tensión de entrada o referencia, tensión de salida y corriente de entrada del convertidor. Para ello, se realizan lazos cerrados de tensión [2] o dobles lazos de tensión y corriente [3], con la ventaja de éstos últimos de permitir la corrección del factor de potencia [4].

El control del PWM se obtiene mediante controladores que engloban las variables y parámetros que se deben tener en cuenta para ejercer la correcta actuación sobre los ciclos de conmutación. Los controladores utilizados más comúnmente son los PI y PID [5] [6].

La frecuencia de conmutación del convertidor depende del tipo de aplicación. Los convertidores de alta potencia (kW) tienen frecuencias de conmutación lentas (decenas de kHz), mientras que los convertidores de baja potencia pueden tener frecuencias de conmutación de hasta decenas de MHz. Por ello, la frecuencia de conmutación limita el dispositivo de control y su tipo de control. Esto conlleva a que el dispositivo elegido en este TFM para el control digital del convertidor conmutado deba ser adecuado a los requerimientos más críticos de éste [7], siendo capaz de generar un PWM de la frecuencia adecuada y con un control preciso de medidas de y cálculos.

LabVIEW es un entorno de desarrollo integrado y multiplataforma para diseño de sistemas mediante un lenguaje de programación visual gráfico perteneciente a la empresa *National Instruments* (NI) [8]. Permite realizar adquisición y análisis de datos, procesamiento de señales, creación de sistemas de control, monitoreo y automatización de sistemas de pruebas y validación. Proporciona librerías específicas para el manejo de diferentes interfaces de comunicaciones (puertos serie, GPIB, PXI, USB, TCP/IP, Bluetooth, etc.), así como capacidad de interactuar con otros lenguajes y aplicaciones tales como librerías de funciones (DLL, Dynamic-Link Library), lenguaje de programación C, .NET, Python o Matlab/Simulink.

La programación gráfica que dispone LabVIEW permite abstraerse de la integración de los dispositivos hardware, siendo myDAQ un dispositivo de adquisición de datos y myRIO un SoC (*System on Chip*), que no necesitan de otros entornos de desarrollo para funcionar, lo que permite reducir la complejidad y con ello, los tiempos de desarrollo [9][10]y [11].

La interfaz de usuario desarrollada con LabVIEW en el presente TFM tendrá la versatilidad de poder ser ejecutada en cualquier Sistema Operativo, sin necesidad de ningún otro software



instalado y permitirá la parametrización de diversos reguladores de los convertidores conmutados.

## 1.1. Objetivos

Los principales objetivos de este TFM son:

- Desarrollo de una interfaz de usuario en el software de LabVIEW que permita realizar el control de convertidores conmutados a través de dicho software.
- Gestión de la interacción entre la interfaz de usuario y ejecución del control en LabVIEW.
- Evaluación de las características de los dispositivos disponibles (NI myDAQ y NI myRIO) y adecuación para el control digital en un convertidor conmutado.
- Inclusión de reguladores en la interfaz mediante la programación gráfica de la forma más óptima desde el punto de vista de recursos disponibles con el dispositivo seleccionado para el control.
- Parametrización de la interfaz para la realización de pruebas con un convertidor conmutado mediante la interfaz desarrollada.
- Realización de un control para la conversión CA/CC utilizando un convertidor conmutado de tipo elevador.
- Análisis de la viabilidad y adecuación de LabVIEW para el control de convertidores con frecuencia de conmutación media (centenas de kHz).

## 1.2. Estructuración de la memoria

El primer capítulo ha expuesto la introducción del TFM, motivaciones de su realización y consecuencias perseguidas.

En el segundo capítulo se explica el contexto de los convertidores conmutados, topologías disponibles, modos de funcionamiento y la importancia de corrección del factor de potencia en el caso de la corriente alterna.

En el tercer capítulo se describe el software de LabVIEW y su método para el desarrollo de interfaces mediante programación gráfica. También se explican las características de ambos dispositivos disponibles para el desarrollo de este TFM.

En el cuarto capítulo se analizan los requisitos más críticos que debe disponer cada dispositivo para su uso en el control digital de un convertidor conmutado. Estos requisitos son contrastados y finalmente se selecciona uno de los dos dispositivos para la implementación de la interfaz de usuario.

En el quinto capítulo se muestran los detalles del diseño e implementación de la interfaz, como son la comunicación, programación y ejecución con el dispositivo escogido, calibraciones e integración de los reguladores.

En el sexto capítulo se expone la integración final de todos los dispositivos con los cuales se realiza el control digital y los resultados obtenidos con la interfaz desarrollada.

En el séptimo capítulo se exponen las conclusiones obtenidas y posibles líneas de desarrollo futuro.

## 2. ESTADO DEL ARTE EN CONVERTIDORES CONMUTADOS

Un convertidor conmutado es un dispositivo electrónico de potencia que transforma la energía eléctrica de entrada, mediante conmutación de transistores a modo de interruptores, hacia otros valores adecuados a su aplicación final. Las posibles conversiones que se permiten son las de CA-CA, CC-CC, CA-CC y viceversa.

A diferencia de los reguladores lineales, los convertidores conmutados no utilizan disipación de energía en forma de calor para conseguir el voltaje deseado, sino que mediante la modificación del ancho de pulso que se aplica a los transistores, se permite adaptar la conducción de energía, por lo que son notablemente más eficientes llegando a alcanzar eficiencias de conversión superiores al 95%.

Previamente a la entrada a un convertidor conmutado, se suele añadir una etapa de potencia para rectificar la onda de CA. Esto se realiza mediante un puente de diodos para obtener una onda rectificada, sin semiciclos negativos y posteriormente una etapa de filtrado para aplanar la señal de salida rectificada y que sea lo más continua posible.

En la Figura 1 se muestra un circuito típico con dichas etapas. La entrada es de corriente alterna, por lo que en el semiciclo positivo, la tensión en los diodos D1 y D3 será mayor en el ánodo que en el cátodo y estarán polarizados en directa, permitiendo el paso de la corriente, mientras que en D2 y D4 estarán en inversa. En el semiciclo negativo de la señal alterna, los diodos que conducen son D2 y D4, de forma que se invierte la polaridad de la señal alterna y la corriente que llega a la carga sigue siendo un potencial positivo.

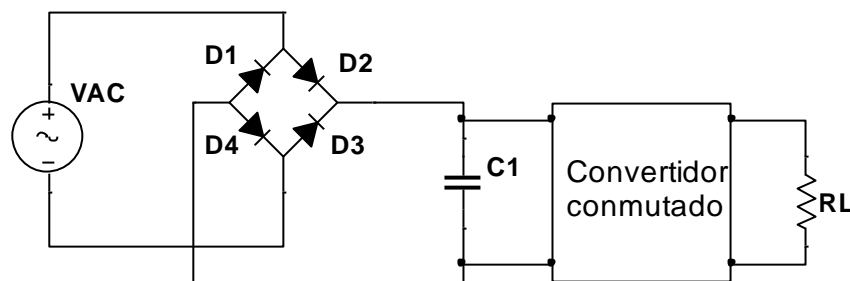


Figura 1: Ejemplo de convertidor conmutado con rectificación de corriente a la entrada

## 2.1 Topologías de convertidores conmutados más comunes

Algunas de las topologías básicas que permiten en esta configuración aumentar o reducir el voltaje se clasifican como:

- Convertidor *buck*: permite obtener un voltaje de salida menor que el de entrada. Cuando el interruptor está cerrado, la corriente circula a través de la bobina, cargando al condensador y alcanzando la resistencia de carga. El diodo permanece polarizado en inversa, puesto que el voltaje es mayor en su cátodo que en el ánodo, como se muestra en la Figura 2 (izquierda).

Si el interruptor está abierto, la corriente en la bobina disminuye, invierte su polaridad y hace que el diodo conduzca al haber un voltaje más positivo en su ánodo. El condensador que previamente fue cargado en el tiempo de ON ( $t_{ON}$ ), se encarga de mantener corriente fluyendo a través de la resistencia de carga (ver Figura 2 derecha).

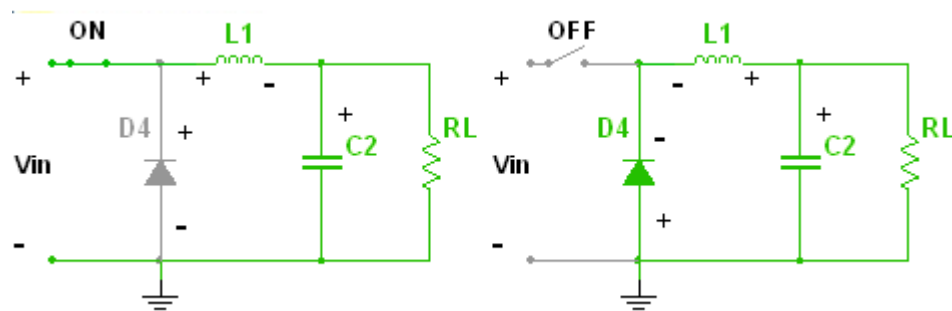


Figura 2: Convertidor *buck* con interruptor  $t_{ON}$  y  $t_{OFF}$  respectivamente

- Convertidor *boost*: obtiene un voltaje de salida mayor que el de entrada. Durante el tiempo que permanece cerrado el interruptor, la bobina almacena energía y la carga permanece alimentada por el condensador, que estará cargado tras un ciclo de conmutación (Figura 3 izquierda). Una vez que se abre el circuito, el diodo entra en conducción, y se transfiere la energía a la carga (ver Figura 3 derecha).

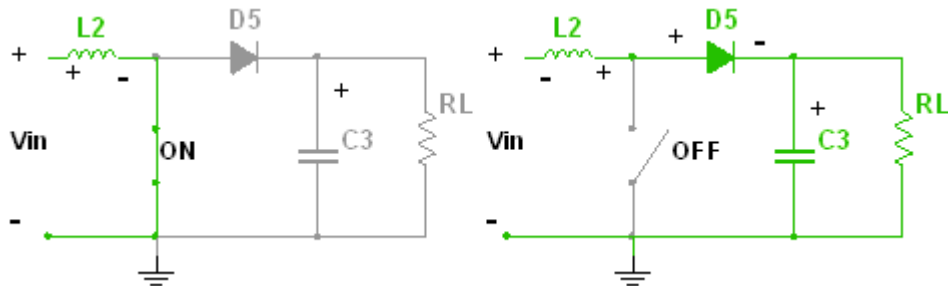


Figura 3: Convertidor *boost* con interruptor  $t_{ON}$  y  $t_{OFF}$  respectivamente

- Convertidor *buck-boost*: combina las topologías previas permitiendo obtener a la salida un voltaje de salida mayor o menor que el de entrada.

## 2.2 Modos de operación según el tipo de conducción

Clasificando los convertidores según su tipo de conducción, existen dos modos de operación independientemente de su topología:

- Modo de conducción continua (CCM, *Continuous Conduction Mode*): la corriente de la bobina nunca llega a ser cero durante cada ciclo de conmutación. La tensión de salida solo depende de la tensión de entrada y del ciclo de trabajo.
- Modo de conducción discontinua (DCM, *Discontinuous Conduction Mode*): la corriente de la bobina se anula durante algún momento del ciclo de conmutación. Ocurre porque la energía que requiere la carga es suficientemente grande o pequeña (según el convertidor *buck* o *boost*, respectivamente) de forma que puede ser transferida en un tiempo menor que el del ciclo de conmutación. En este caso, la tensión de salida depende de la corriente de salida, valor de la bobina y periodo de conmutación además del ciclo de trabajo y tensión de entrada.

## 2.3. Factor de potencia

En un dispositivo eléctrico se encuentran 3 elementos principales:

- Resistencias: si el circuito es puramente resistivo, no existe desfase entre la tensión y corriente.
- Inductancias: si el circuito es puramente inductivo, donde la carga se comporta como una bobina, la corriente está atrasada  $90^\circ$  respecto a la tensión.
- Capacitancias: si el circuito es puramente capacitivo, la corriente está adelantada  $90^\circ$  respecto a la tensión.

El factor de potencia (PFC, *Power Factor Correction*) establece la relación entre la potencia activa (P) y la potencia aparente (S). Cuando en un circuito existen elementos inductivos o capacitivos, también aparece la potencia reactiva (Q). En la Figura 4 se muestra el diagrama que relaciona las 3 potencias.

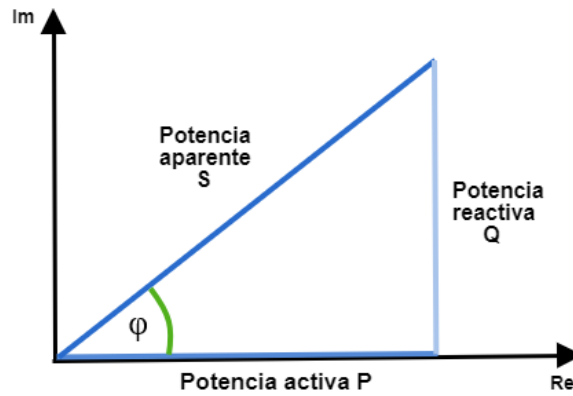


Figura 4: Relación entre los diferentes tipos de potencia

La potencia activa es aquella que es aprovechada por los dispositivos conectados a la red eléctrica o a una fuente de alimentación. La potencia aparente es la potencia consumida de la red eléctrica y la potencia reactiva es la que necesitan los elementos inductivos o capacitivos dentro del circuito que precisan de energía para generar un campo magnético para su funcionamiento, pero dicha potencia no se transforma en trabajo efectivo, sino que fluctúa por la red entre el generador y los componentes conectados a ella.

El factor de potencia (f.d.p.) es una medida de la eficiencia de consumo eléctrico. El ángulo que se forma entre la potencia activa y la aparente, se denomina comúnmente  $\cos \varphi$  e influye directamente en el factor de potencia. El  $\cos \varphi$  depende del desfase que haya entre las ondas de tensión y corriente por el comportamiento no lineal de la carga al tener elementos inductivos o capacitivos. Si las ondas no tienen desfase el ángulo es  $0^\circ$ , por lo que  $\cos 0^\circ = 1$ . A mayor ángulo de desfase, el  $\cos \varphi$  disminuye de la unidad, lo que indica que la potencia no se está aprovechando eficientemente.

En las fuentes de alimentación, se generan además del desfase entre las ondas, distorsiones en la señal sinusoidal. Concretamente, el condensador de la etapa de rectificación no llega a cargarse y descargarse completamente, lo que provoca distorsión o fluctuaciones en la intensidad del circuito de forma que no se consigue una forma sinusoidal perfecta. Esas distorsiones en la corriente son los armónicos, que se generan a frecuencias múltiplo de la frecuencia fundamental y que se mantienen circulando por la red eléctrica generando interferencias o sobrecalentamientos a otros dispositivos que estén conectados a dicha red.

### 2.3.1. Corrección del Factor de Potencia

Actualmente existen normativas internacionales, como la Norma EN61000-3-2 publicada por el IEC (*International Electrotechnical Commission*) que limitan el contenido armónico de la red eléctrica causado por los dispositivos conectados a la red.

El desfase introducido por el comportamiento de las cargas de forma no lineal y las distorsiones introducidas en la corriente influyen directamente en la eficiencia de consumo eléctrico y, por lo tanto, en el parámetro de f.d.p., de ahí la necesidad de corrección de dicho factor.

Además, si se controla el f.d.p., la potencia aparente no útil se disminuye, y la mayor eficiencia conseguida permite reducir las pérdidas de energía por lo que se puede aumentar el voltaje a distribuir en la red eléctrica y disminuir la posibilidad de dañar los dispositivos conectados a ella, puesto que el contenido armónico se encuentra limitado.

La corrección del f.d.p. puede estar basada en:

- Soluciones pasivas: es la solución más simple, mediante el uso de elementos pasivos que permiten eliminar el desfase entre corriente e intensidad, pero no resuelven completamente el problema de la distorsión de la corriente.
- Soluciones activas: como los convertidores conmutados, cuyo objetivo es conseguir eliminar el desfase que existe entre tensión y corriente además de la distorsión. Permiten obtener un f.d.p. cercano a la unidad, sin embargo, la complejidad del control de los convertidores conmutados es mayor.

Una de las soluciones activas para corrección de f.d.p. es el de la implementación de un sistema de control de doble lazo: un lazo de tensión y otro de corriente. En este TFM se ha implementado un control digital de corrección del f.d.p. mediante doble lazo (ver Figura 5) para un convertidor conmutado tipo *boost* en el que se realiza un control digital. El convertidor *boost* se ha reutilizado del diseño que se realizó en [12], cuyas características de diseño se describen en el Anexo I.1 Características de convertidor boost

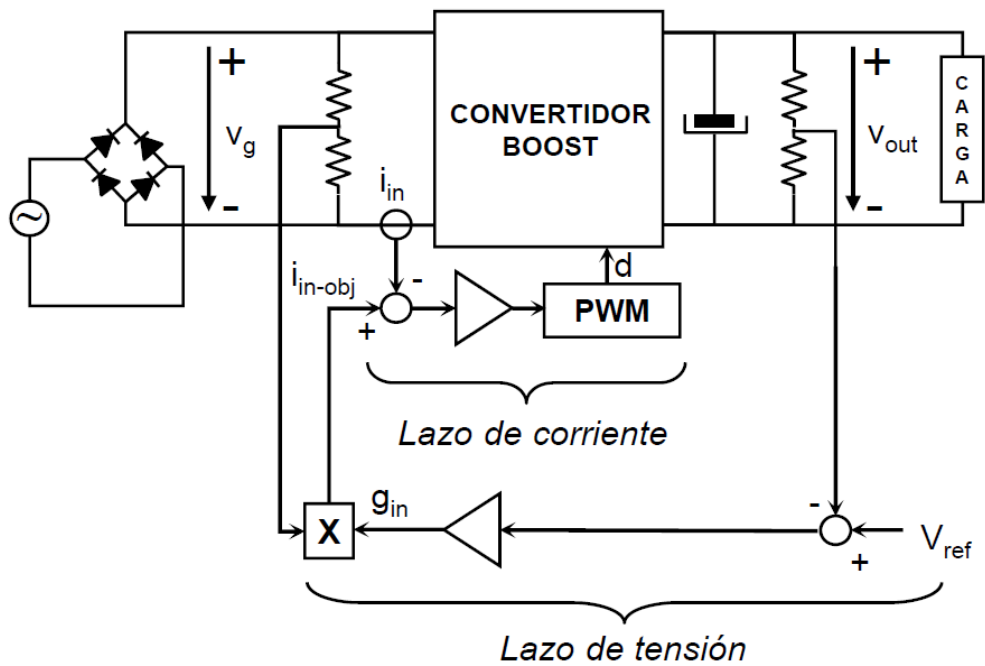


Figura 5: Configuración para PFC y control de doble lazo para convertidor *boost*

En el diagrama de la Figura 5, obtenida de [13], se muestra cómo se realiza el control mediante la implementación de dos lazos, cada uno con su propio regulador. El lazo externo de tensión controla la tensión media de salida para mantenerla fija al voltaje de referencia deseado y el lazo interno se encarga de eliminar la distorsión y que la corriente sea también de forma sinusoidal, al igual que la tensión de entrada. En el lazo de tensión, obtiene a la salida del regulador un valor de conductancia que se multiplica por el voltaje de entrada al convertidor, de forma que se genera la corriente de referencia para ser comparada con la corriente de entrada. Esta comparación genera el error que debe ser corregido mediante la actuación del regulador del lazo de corriente, cuya salida da la referencia para el control del pulso PWM y la generación del ciclo de trabajo apropiado.



### 3. LabVIEW

LabVIEW es un entorno de desarrollo integrado y multiplataforma para diseño de diversos sistemas mediante un lenguaje de programación visual gráfico que pertenece a la empresa *National Instruments (NI)*. El software se compone de diferentes módulos de instalación según los dispositivos utilizados con el que crear diagramas de bloques específicos para diversas funcionalidades como la adquisición y análisis de datos, procesamiento de señales, creación de sistemas de control, monitoreo y automatización de pruebas y simulaciones. Todas estas aplicaciones hacen que LabVIEW sea ampliamente utilizado en entornos educativos como industriales y de I+D.

Las principales ventajas que presenta la programación gráfica es la reducción de complejidad y tiempos de desarrollo, permitiendo al usuario enfocarse en la implementación de una forma mucho más directa. NI dispone de una amplia variedad de dispositivos hardware para la adquisición de datos, donde el usuario simplemente debe conectar su sistema a controlar y desarrollar su interfaz.

LabVIEW dispone de los medios para la comunicación entre software y hardware, siendo el propio software el que maneja la asignación de hilos y procesos del ordenador, permitiendo realizar ejecuciones en paralelo.

La ejecución es determinada por la estructura de los diagramas de bloque gráficos en los cuales el desarrollador conecta las diferentes funciones o nodos a través de cables. Esos cables son los que propagan variables y cada nodo puede ejecutarse tan pronto como todas sus entradas estén disponibles.

El elemento básico en LabVIEW se llama *Virtual Instrument (VI)*, formado por estos dos componentes:

- Panel frontal: es la interfaz de usuario con la que se interactúa cuando el programa se está ejecutando. Se compone básicamente de dos tipos de elementos:
  - Controles: son las entradas al VI.
  - Indicadores: son las salidas del VI.

En la Figura 6 se muestra un ejemplo de un panel frontal con controles que pueden ser modificados e indicadores que muestran resultados.

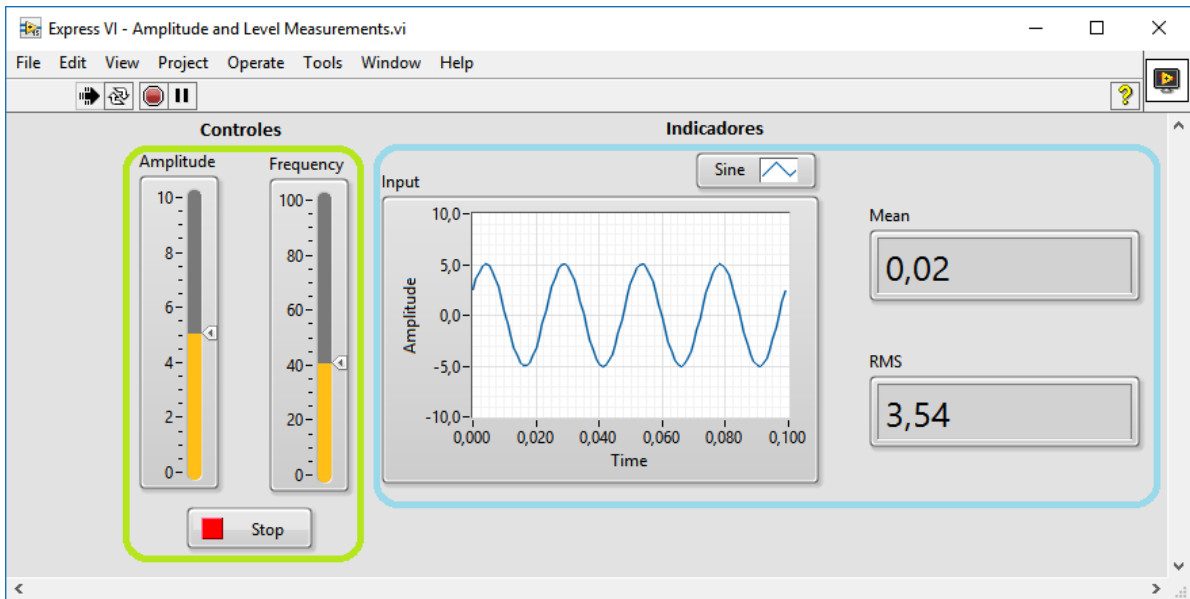


Figura 6: Ejemplo de panel frontal de LabVIEW

- Diagrama de bloques: es el programa donde el desarrollador define la funcionalidad mediante las interconexiones de bloques (ver Figura 7):
  - o Implementa las estructuras y funciones que se ejecutarán.
  - o Los datos se propagan a través de los cables, desde los controles pasan a través de funciones y terminan en los indicadores, que indican el resultado de la ejecución.
  - o Los bloques se ejecutan a través del flujo de datos.

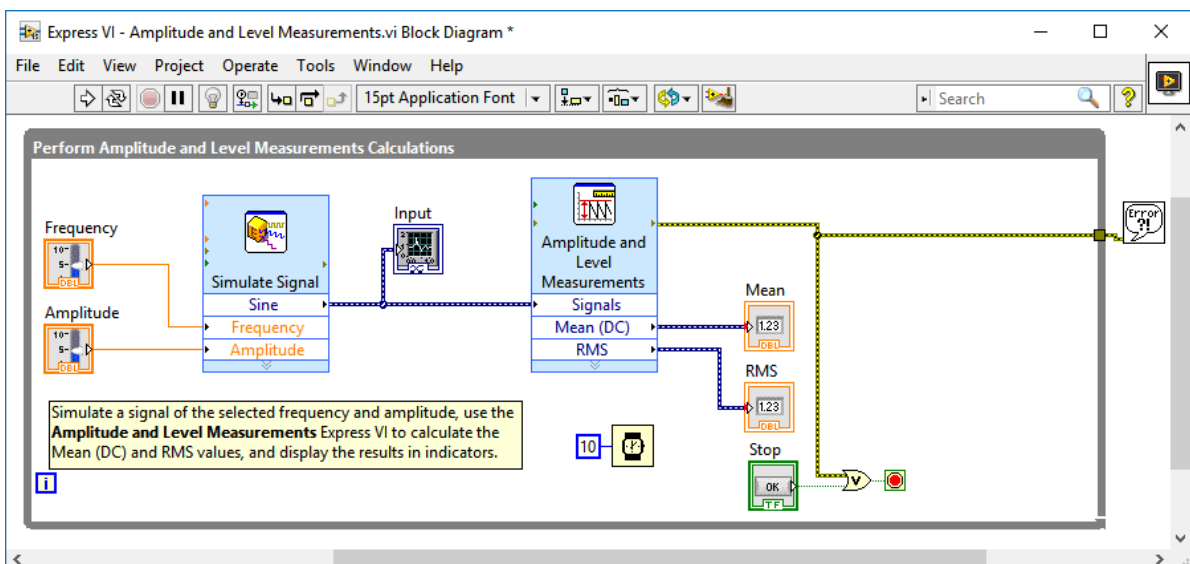


Figura 7: Ejemplo de diagrama de bloques

En LabVIEW es de vital importancia la temporización y sincronización de eventos en el tiempo para la ejecución de los VIs. Las principales estructuras que se encuentran en los VIs son secuencias, *case* y bucles *while* y *for*.

Para mantener el registro del tiempo dentro del VI, LabVIEW utiliza un componente software llamado motor de nanosegundos que se ejecuta como un proceso de fondo y se comunica con el sistema operativo para el manejo del tiempo. Al utilizar los contadores del sistema operativo en el que se ejecuta el VI, la resolución de ellos recae sobre dicho sistema operativo. Las resoluciones típicas de Windows son de 1 ms y para Linux y Mac de 10 ms [14].

Al depender del software, la ejecución de los bucles puede variar entre cada iteración dependiendo de los demás procesos que se estén ejecutando en el sistema operativo en cada momento y, por tanto, del uso de la CPU. La mejora de precisión y fiabilidad en la ejecución y adquisición de datos se puede conseguir mediante la ejecución basada en hardware que puede configurarse tanto en dispositivos DAQ (*Data Acquisition*) y RIO (*Reconfigurable Input Output*) de NI.

Utilizando LabVIEW para adquisición de datos se permite configurar tasas de muestreo adquiriendo cierto número de muestras en un determinado tiempo, por lo que las desigualdades entre iteraciones de bucles y dependencia del software desaparecen. En el caso de LabVIEW para FPGA, la ejecución se realiza con el reloj interno de la FPGA, por lo que se alcanzan las velocidades de procesamiento más elevadas.

A continuación, se describen las características que poseen los posibles dispositivos para el control del convertidor conmutado que se utilizará en este TFM.

### 3.1 NI myDAQ

Es un dispositivo de adquisición de datos para aplicaciones en entornos de laboratorios educativos. Entre sus características destaca su portabilidad, pequeñas dimensiones y bajo coste. Permite la medición, análisis y visualización de señales en tiempo real programando el dispositivo mediante el software específico de LabVIEW. La conexión al ordenador se realiza mediante USB. Las principales características que posee son:

- Señales analógicas

Posee 2 entradas y 2 salidas, siendo configurables como entradas o salidas de propósito general o de audio. En cada caso, se pueden medir o generar señales de hasta  $\pm 10$  V

con una tasa de muestreo máxima de 200.000 muestras/s. Las especificaciones para cada tipo de señal son:

- Las 2 entradas analógicas tienen 16 bits de resolución y pueden obtenerse en rangos de  $\pm 10\text{ V}$  o  $\pm 2\text{V}$ . Estas entradas son multiplexadas, por lo que el mismo ADC (*Analog to Digital Converter*) es el que se encarga de muestrear ambos canales.
  - 2 salidas analógicas de 16 bits de resolución y en rangos de  $\pm 10\text{ V}$  o  $\pm 2\text{V}$ . En este caso, cada canal tiene un DAC (*Digital to Analog Converter*) dedicado, por lo que se actualizan simultáneamente. Cada salida es capaz de suministrar 2 mA.
- Señales digitales

Posee 8 líneas para señales digitales, siendo cada una de ellas una Interfaz de Funciones Programables (PFI, *Programmable Function Interface*) que pueden configurarse como entradas o salidas digitales de propósito general controladas mediante software o como contadores digitales. Las entradas y salidas digitales son líneas *Low Voltage TTL* (LVTTL) y admiten hasta entradas de 5 V. Las salidas digitales conducen hasta 4 mA por línea y no son compatibles con niveles lógicos de 5V CMOS.

Las líneas DIO <0-4> pueden ser configuradas como contadores, temporizadores, medidores de ancho de pulso y para aplicaciones de encoders en cuadratura. El reloj interno que poseen es de 100 MHz y el máximo contador y tasa de actualización de la generación de pulsos es de 1.000.000 muestras/s y 32 bits de resolución. Según las diferentes configuraciones que se realicen en el software, estas líneas permitirán:

- Generación de PWM.
  - Medidas de periodo de señales digitales de entrada.
  - Medida de ancho de pulso Contar flancos de entradas digitales.
  - y frecuencia de entradas digitales.
  - Medida de separación entre flancos de dos señales digitales.
  - Medida de posición con una señal proveniente de un encoder lineal o angular.
- Multímetro digital

Permite realizar mediciones de corriente y voltaje tanto en DC como AC, medidas de resistencia, test de diodos y de continuidad. La resolución que tiene es de 3,5 dígitos y los rangos de medida son los siguientes mostrados en la Tabla 1:

Voltaje		Corriente		Resistencia	Diodo
AC	DC	AC	DC		
200 mV <sub>rms</sub>	200 mV	20 mA <sub>rms</sub>	20 mA	200Ω	2V
2 V <sub>rms</sub>	2 V	200 mA <sub>rms</sub>	200 mA	2kΩ	
20 V <sub>rms</sub>	20 V	1 A <sub>rms</sub>	1 A	20kΩ	
	60 V			200kΩ	
				2MΩ	
				20MΩ	

Tabla 1: Rangos de medida dispositivo myDAQ

- Salidas de potencia

NI myDAQ posee salidas de +15 V, - 15 V y 5 V que pueden servir para alimentar otros sistemas. En las dos primeras, son capaces de suministrar hasta un máximo de 32 mA, y en la salida de 5 V, hasta de 100 mA.

Con todos los consumos de voltaje y corriente, hay que tener en cuenta que el dispositivo permite un máximo de 500 mW disponibles para todas las salidas analógicas, digitales y de potencia y un mínimo de 100 mW.

### 3.2 NI myRIO

Es otro dispositivo portable de NI con entradas y salidas reconfigurables para diseño de sistemas de control, robótica, mecatrónica o sistemas embebidos. Entre sus características más destacables se encuentran la integración de una FPGA Xilinx Zynq y un procesador ARM Cortex-A9 con los que poder programar las entradas, salidas y realizar procedimientos en tiempo real. En la actualidad se denominan SoC (*System on Chip*). La conexión con el ordenador se puede realizar mediante USB o Wireless y requiere de alimentación externa para su funcionamiento. Posee tres puertos de expansión:

- MXP A/B (*Expansion Port*): ambas expansiones con las mismas señales, con entradas y salidas analógicas y digitales, salidas de potencia de 5 y 3,3 V con sus referencias de masa y las líneas de recepción y transmisión de una UART.
- MSP C (*Mini System Port*): idéntico al conector de NI myDAQ. Posee entradas y salidas analógicas y digitales, salidas de potencia de 5 V y  $\pm 15$  V, sus referencias de masa y conexiones de entrada y salida de audio. A diferencia de los otros dos puertos de expansión, posee unas entradas analógicas en modo diferencial.

A continuación, se muestran las características para cada tipo de señal y según su puerto de expansión:

- Señales analógicas
  - Las entradas analógicas son multiplexadas a un único ADC que muestrea todos los canales. En los conectores MXP A/B, cada uno posee 4 entradas para medir en rangos de 0 a 5 V. En MSP C, las dos entradas analógicas son diferenciales para medir hasta  $\pm 10$  V. Las dos entradas de audio permiten hasta  $\pm 2.5$  V. La tasa de muestreo máxima es de 500.000 muestras/s con resolución de 12 bits.
  - Las salidas analógicas poseen un DAC dedicado por cada canal, al igual que en NI myDAQ. Los DACs son controlados por dos buses de comunicación serial desde la FPGA (SPI, *Serial Peripheral Interface*). Los conectores MXP A/B comparten un mismo bus y el conector MSP C junto con las salidas de audio, comparten el otro. En el MXP, cada conector posee dos salidas configurables entre 0 y 5 V con 3 mA de corriente de salida y en el puerto MSP hay otras dos salidas capaces de generar entre  $\pm 10$  V y 2 mA. La tasa de muestreo en los tres puertos es de 345.000 muestras/s en cada canal.

- Señales digitales

Posee 16 líneas DIO de 3.3 V de propósito general en cada puerto MXP A/B y 8 en el puerto MSP, con sus respectivas referencias a masa. Son compatibles con niveles lógicos LVTTTL Algunas de estas líneas pueden programarse como:

- SPI con frecuencia de 4 MHz
- PWM con frecuencia de hasta 100 kHz
- Entrada de encoder en cuadratura de hasta 100 kHz
- I<sup>2</sup>C con hasta 400 kHz
- Una UART en cada puerto de MXP A/B con baud rate máximo de 230.400 bps

- **Acelerómetro**  
Muestra continuamente los 3 ejes principales de coordenadas, tiene 12 bits de resolución y una tasa de muestreo de 800 muestras/s.
- **SoC**  
El SoC completo es un Zynq-7010 compuesto del procesador Dual-core ARM Cortex A-9 con velocidad de 667 MHz, 512 MB de memoria no volátil y 256 MB de memoria DDR3 y una FPGA con lógica programable derivada de la serie Artix-7 de Xilinx. El reloj interno es de 40 MHz, con ciclo de reloj cada 25 ns.
- **Salidas de potencia**  
NI myRIO posee las mismas salidas de potencia que NI myDAQ, además de una de 3.3 V. Los suministros de corriente son los mismos que en el otro dispositivo, y en la nueva salida de 3.3 V es capaz de suministrar hasta 150 mA.
- **Otras conexiones**  
También dispone de conexión WiFi y de un puerto USB compatible con cámaras web (UVC, *USB Video Device Class* o *USB3 Vision Standard*) y memorias USB, muy útiles para guardar ficheros de datos extraídos de la ejecución de los VIs.

El SoC que posee NI myRIO permite crear aplicaciones en tiempo real y a nivel de FPGA. Al estar una parte de la FPGA configurada desde la fábrica, permite iniciar la ejecución de funciones sin necesidad de programar desde el nivel inferior la FPGA.

## 4. REQUISITOS Y SELECCIÓN DEL DISPOSITIVO

El control digital de este TFM se realizará sobre un convertidor conmutado de tipo *boost* diseñado con especificaciones para CA. Las características específicas de requisitos para el control de convertidores conmutados son determinantes para la elección del dispositivo que deba realizar el control digital. A continuación, se analizan y contrastan las diversas características para la elección final del dispositivo que será utilizado.

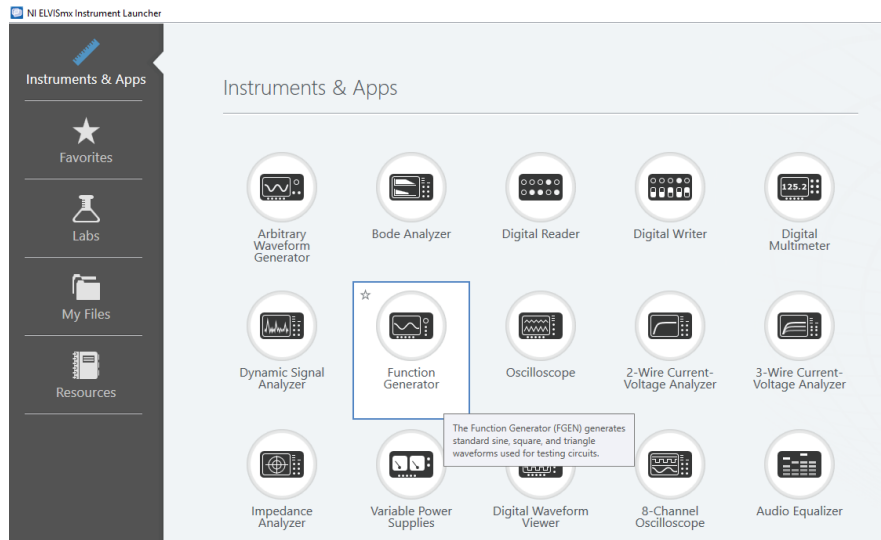
### 4.1. Frecuencia de conmutación

La frecuencia de conmutación del elemento que actúa como interruptor del convertidor *boost* viene impuesta por el propio diseño e influye en los valores de bobina que se escogieron para su realización.

El dispositivo encargado de regular el paso de la corriente a través del *boost* es un MOSFET tipo N que deberá conmutar a 100 kHz, lo que implica que la frecuencia del pulso PWM debe ser generada a dicha frecuencia. Además, el reloj interno del controlador PWM debe tener una frecuencia mayor (al menos 100 veces mayor) para poder tener suficiente resolución a la hora de generar el ciclo de trabajo deseado.

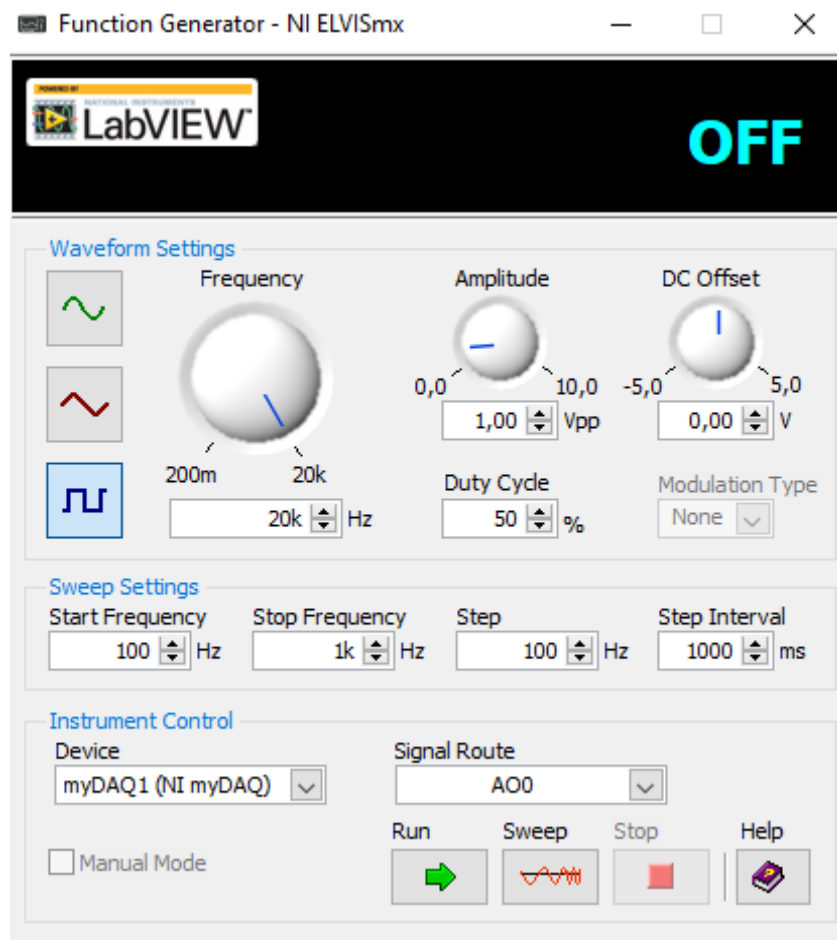
El dispositivo NI myDAQ dispone con su instalación un paquete de instrumentos integrados asociado a sus diversas funcionalidades para acceder rápidamente a todas ellas, llamado NI ELVISmx (*Educational Laboratory Virtual Instrumentation Suite Instrument Launcher*). En la Figura 8 se muestra el panel con todas las opciones disponibles.





**Figura 8: Panel frontal de ELVISmx**

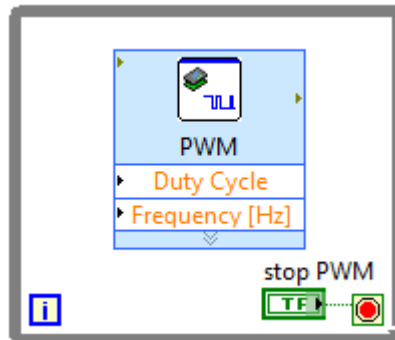
Se comprobó en la opción de generador de funciones, que es posible generar una salida analógica a través de las líneas AO, pero limitada a una frecuencia máxima de 20 kHz, como muestra la Figura 9.



**Figura 9: Generador de funciones en ELVISmx**

En la programación del VI, debido a la ejecución de los bucles que se explicó en el apartado previo (3), la generación de muestras digitales está limitada por la resolución de ejecución del VI, que para el SO de Windows es de 1ms y es dependiente de los recursos y procesos que se estén ejecutando en ese momento en el PC host.

En el dispositivo de myRIO, existe una función configurable para PWM (ver Figura 10) donde se permite elegir en qué pin de salida de la placa se generará y a qué frecuencia.



**Figura 10: Función PWM de LabVIEW FPGA de NI myRIO**

Se ha comprobado que tampoco permite configurar la frecuencia de conmutación del *boost* a 100 kHz, sino que a 100 kHz indica que es una frecuencia inválida y lo limita a 40 kHz como máximo (ver Figura 11). Sin embargo, la ventaja de myRIO es que al contener una FPGA, LabVIEW permite un modo de programación a nivel interno de la FPGA a través del diagrama de bloques, cuyo compilador se encarga de generar los ficheros traducidos a un lenguaje HDL (*Hardware Description Language*). La FPGA de NI myRIO tiene un reloj interno de 40 MHz, lo que permitirá generar una funcionalidad de PWM mediante un contador de los pulsos de reloj (denominados *ticks* en LabVIEW) para alcanzar la frecuencia de 100 kHz. Esta funcionalidad se analiza en el apartado 5.2.1.

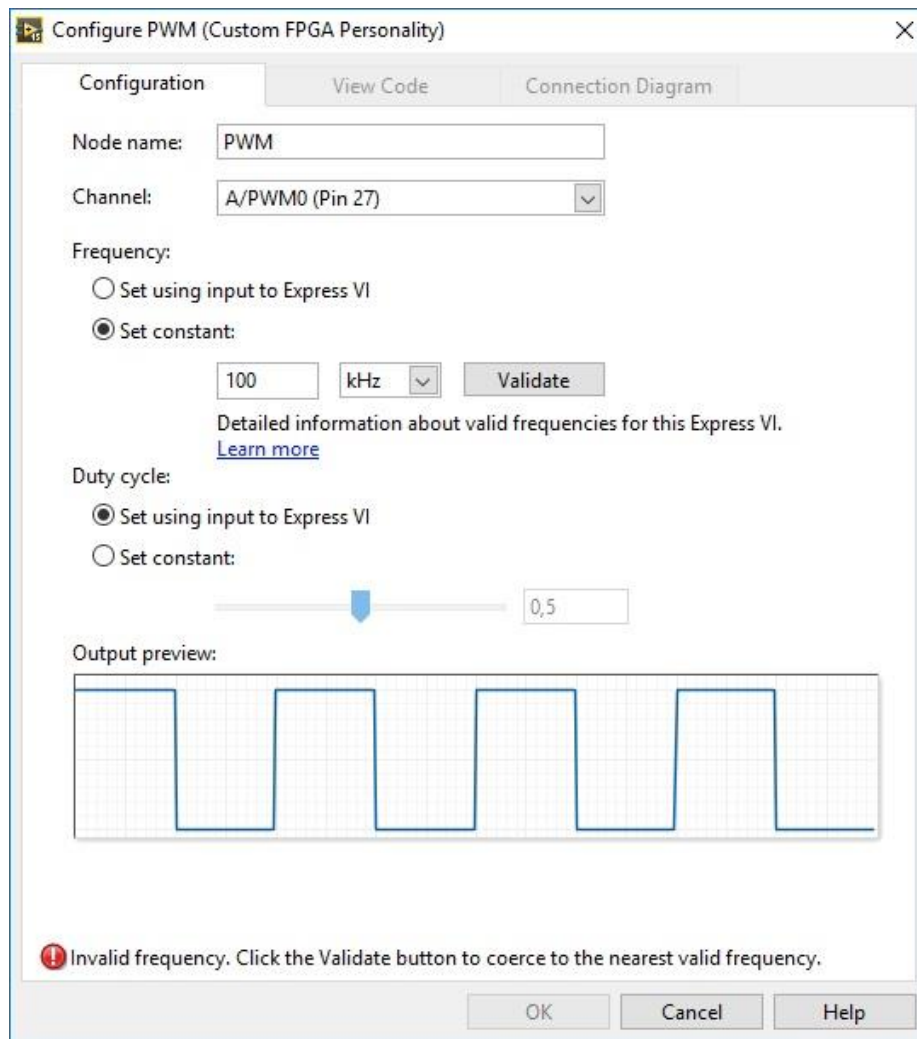


Figura 11: Generación inválida de PWM en myRIO

Dada la criticidad que conlleva el no poder generar un PWM con diferente *duty cycle* que se ejecute de manera continua y uniforme, se descarta el uso de NI myDAQ para la corrección de factor de potencia.

## 4.2. Nivel de voltaje para las entradas analógicas

Este parámetro viene impuesto por el diseño del convertidor *boost* utilizado, que tiene etapas de aislamiento de tensión y corriente mediante optoacopladores. A la salida de los optoacopladores se encuentran unos divisores resistivos que limitan la tensión a 1 V, que es la máxima tensión que soporta el ADC de la placa externa, una *SpCard* [15] que originariamente realizaba la etapa de control y generación del PWM.

Para realizar un control en tensión con alguno de los dos dispositivos disponibles, basta con tener 2 entradas analógicas disponibles cuyos ADCs soporten al menos hasta 1 V de entrada.

Si se quiere realizar la etapa de control con corrección del factor de potencia, se debe disponer de 3 entradas analógicas: dos para medir las tensiones de entrada y salida y la otra para medir la corriente de entrada.

El dispositivo NI myDAQ solo posee dos entradas analógicas diferenciales de hasta  $\pm 10$  V, por lo que para un control en tensión sí es válido, pero no para la aplicación de corrección del f.d.p.

NI myRIO, además de disponer de esas mismas dos entradas analógicas diferenciales, tiene en total 8 entradas analógicas para medir tensiones entre 0 y 5 V, por lo que este dispositivo es más adecuado tanto para el tipo de control que se quiera realizar como para minimizar las pérdidas de resolución por la adaptación de niveles de voltaje.

### 4.3. Tasa de muestreo de los ADCs

Para realizar un control digital para corrección del factor de potencia basado en doble lazo, se deben muestrear una vez por cada ciclo de conmutación del convertidor *boost* las variables de voltaje de entrada y corriente de entrada mientras que la tensión de salida puede ser muestreada a menor frecuencia, a 100 Hz.

La criticidad de estos parámetros en los dispositivos de NI se comprueba en el número de muestras por segundo que son capaces de adquirir en sus entradas analógicas.

Para NI myDAQ, sus ADCs tienen una tasa de muestreo máxima de 200.000 muestras/s y NI myRIO de 500.000 muestras/s. En ambos dispositivos, las entradas analógicas son multiplexadas a un único ADC, por lo que el número de muestras disponibles se reduce proporcionalmente al número de canales muestreados.

En el caso de myDAQ, se tienen  $\frac{200000 \text{ muestras/s}}{3 \text{ canales}} \sim 66665$  muestras por canal en cada segundo de forma que no se alcanza a tener una muestra por cada ciclo de conmutación.

En myRIO, se tienen  $\frac{500000 \text{ muestras/s}}{3 \text{ canales}} \approx 166665$  muestras por canal y por segundo, por lo que hay al menos una muestra de cada dato por ciclo de conmutación.

Este requisito también es otra razón para el descarte de uso del dispositivo de NI myDAQ.

## 4.4. Resolución de los ADCs

Esta característica es importante para saber cuál va a ser el mínimo valor que podremos leer de corriente y tensiones de entrada y salida al convertidor *boost*.

En este caso, el dispositivo myDAQ dispone de mejor resolución que myRIO, con unos ADCs de 16 bits de resolución frente a los 12 bits de myRIO. Esto significa que en myDAQ se pueden representar una mayor cantidad de valores discretos para representar el rango de valores analógicos admitidos por el ADC, y por tanto tendría mayor precisión que myRIO. A continuación, se muestran los valores mínimos de voltaje detectables por cada dispositivo.

Resolución de myDAQ para entradas analógicas de  $\pm 2 \text{ V} = \frac{4}{2^{16}} = 6,1035 \times 10^{-5} \text{ V}$

Resolución de myRIO para entradas analógicas de 0 a 5 V =  $\frac{5}{2^{12}} = 1,2207 \times 10^{-3} \text{ V}$

Sin embargo, aunque para esta característica sería más óptima la utilización de NI myDAQ, no se permite por el incumplimiento de los requisitos previos analizados.

## 4.5. Adquisición de muestras durante el ciclo de conmutación

Además de poder elegir la frecuencia de muestreo adecuada, se debe tener en cuenta el punto exacto de la adquisición de las muestras. En cada ciclo de conmutación, la apertura y cierre del interruptor genera ruido eléctrico de conmutación. Cuanto mayor sea la energía manejada por el convertidor, el ruido es mayor. Este problema es habitual en el control de convertidores conmutados. Para evitar adquirir muestras en puntos donde hay ruido, el mejor punto para la adquisición de los datos es en el punto medio del  $t_{\text{ON}}$  o de  $t_{\text{OFF}}$  de cada ciclo de conmutación.

Como se explicó en apartados previos, no se puede conseguir la frecuencia de conmutación ni tasa de adquisición requeridas por las limitaciones hardware del dispositivo myDAQ, por lo que no habría método de adquirir muestras exactamente en el punto medio de algún intervalo.

En cambio, desde la programación interna de la FPGA de myRIO a través de LabVIEW se permite contabilizar el número de *ticks* y muestrear a la frecuencia deseada. El método seguido para obtener las muestras en el punto medio de los intervalos se mostrará en el siguiente apartado, en 5.2.

## 5. IMPLEMENTACIÓN Y PRUEBAS

Tras el análisis de las características y limitaciones de cada uno de los dispositivos y los requisitos del convertidor *boost*, se escoge a NI myRIO para la implementación y pruebas que se realizarán.

### 5.1 Implementación

Inicialmente se han realizado pruebas de control de voltaje sobre un pequeño circuito electrónico para comprender cómo se debe desarrollar la interfaz, cuál es el funcionamiento de NI myRIO y las funciones que se deben ejecutar en el VI. El circuito utilizado es un RC como el mostrado en la Figura 12 sobre el que se controlaba la tensión de salida mediante la generación del pulso PWM.

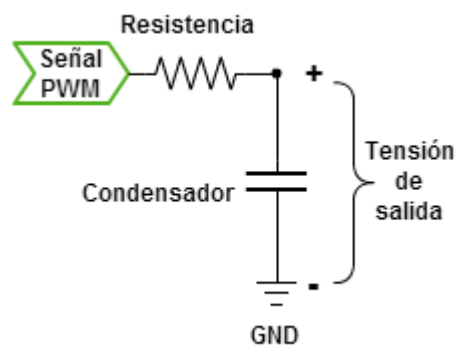


Figura 12: Circuito RC

A continuación, se muestran las características, configuraciones y pruebas que se han ido descubriendo a lo largo del desarrollo.

#### 5.1.1. Comunicación entre PC y myRIO

Para la programación de myRIO se necesita tener instalado el módulo de LabVIEW FPGA que contiene las funciones y operaciones necesarias para la programación y comunicación con la FPGA del dispositivo myRIO y el PC.

Los VIs creados para ejecutarse y descargarse en un módulo de NI con FPGA se nombran FPGA VI mientras que los módulos que permiten la interacción del módulo FPGA VI con el usuario se denominan Host VI.

Cuando el *bitfile* resultado de un FPGA VI se descarga al dispositivo, se programa la FPGA para que ejecute la funcionalidad que se ha diseñado desde LabVIEW.

Al desarrollar en este TFM un sistema de control, es necesario transferir datos tales como el voltaje de referencia o la activación o desactivación del pulso PWM. El módulo FPGA permite la transferencia de datos de 3 formas diferentes:

- Comunicación programada del panel frontal: recomendada cuando se necesita transferir poca cantidad de datos frecuentemente y tan rápido como sea posible, como en el caso de transferencia de datos para control o simulaciones. Para poder acceder y modificar los controles que del FPGA VI se utilizan funciones de control de lectura o escritura desde el Host VI. La interacción con la FPGA se realiza a través del panel frontal del Host VI que se ejecuta en el PC, permitiendo además que se realicen otros procesamientos en el Host VI. En la Figura 13 se muestra un ejemplo de cómo es la interacción de la comunicación con una entrada y salida digital.

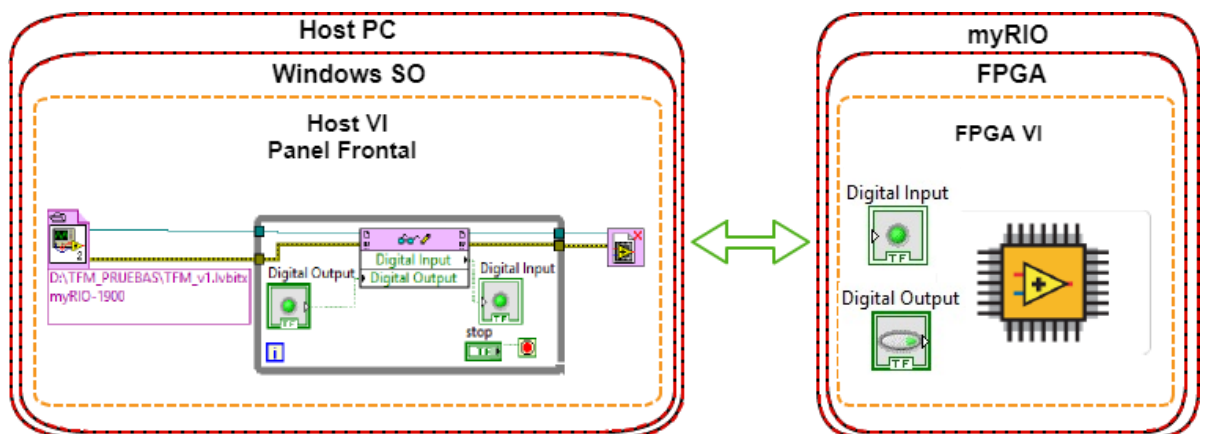


Figura 13: Interacción entre Host VI y FPGA VI

- Acceso directo a memoria (DMA, *Direct Memory Access*): si se necesitan transferir grandes cantidades de datos este método permite definir DMA FIFOs para acumular o transferir los datos de forma eficiente, como en los casos de transferencia de datos de *log* o generación de formas de onda. La principal desventaja es la sobrecarga asociada a cada transferencia de datos, al contrario del primer método.
- Variables de I/O definidas por el usuario: son un tipo de variables compartidas que pueden utilizar los dispositivos con soporte de NI Scan Engine (Motor de escaneo) el cual permite un acceso eficiente a canales de I/O, almacenamiento de datos mediante el mapeo de una memoria global y actualizando todos los valores a la misma tasa. Este

método permite reducir la cantidad de código necesario para acceder y transferir conjuntos de datos entre los canales de I/O y el Host VI, procesando datos en la FPGA antes de enviarlos al Host VI y posteriormente enviarlos a la FPGA.

Puesto que la aplicación de este TFM no requiere de un envío de numerosos conjuntos de datos, sino de valores puntuales y transmitidos lo más rápido posible, se opta por la opción de comunicación programada del panel frontal.

### 5.1.2. Programación de myRIO

El módulo de LabVIEW FPGA es el encargado de traducir los diseños desarrollados en el VI FPGA al hardware de la FPGA mediante un proceso de compilación para obtener un fichero para programación de FPGA (*bitfile*). Para realizar la compilación debe instalarse en LabVIEW la versión adecuada de herramientas de compilación de Xilinx, que para el caso de la FPGA de myRIO es la versión 2014.4 de Vivado. Existe un panel de opciones en el que se puede modificar la estrategia de diseño para la compilación (optimizar área, rendimiento, reducir tiempo de compilación, etc.). Los valores utilizados para las pruebas son los que están por defecto.

El proceso que se sigue para la creación del *bitfile* es el mostrado en la Figura 14:





Figura 14: Esquema de generación de archivo *bitfile* (.bit)

En el inicio de la compilación, permite elegir el servidor de compilación, como se muestra en la Figura 15. El compilador se encarga de crear un sistema de conexiones lógicas entre los elementos de procesamiento de la FPGA, realizando el mapeo, colocación y enrutamiento más óptimo dentro de los límites lógicos del chip. Dado que los tiempos de compilación son dependientes de la complejidad de la aplicación y prestaciones del Host PC y pueden ser elevados, LabVIEW permite, además de la compilación en PC local, la transferencia de dicho

trabajo a otras máquinas distintas que puedan tener prestaciones superiores. Para este TFM se ha escogido la compilación en el propio PC en el que se ha realizado el desarrollo.

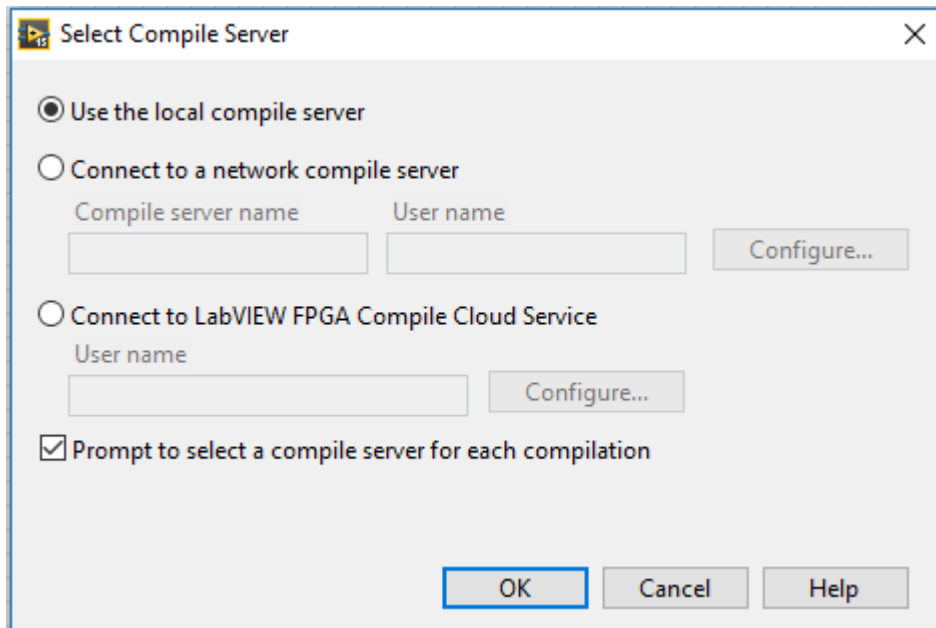


Figura 15: Diálogo de opciones para compilación

En el proceso de generación de ficheros intermedios (ver Figura 16), LabVIEW convierte el diagrama de bloques del FGPA VI a código que el compilador de Xilinx pueda interpretar (código HDL). En este proceso pueden surgir errores debido a la programación del VI que se haya implementado, combinación errónea de VIs, etc.

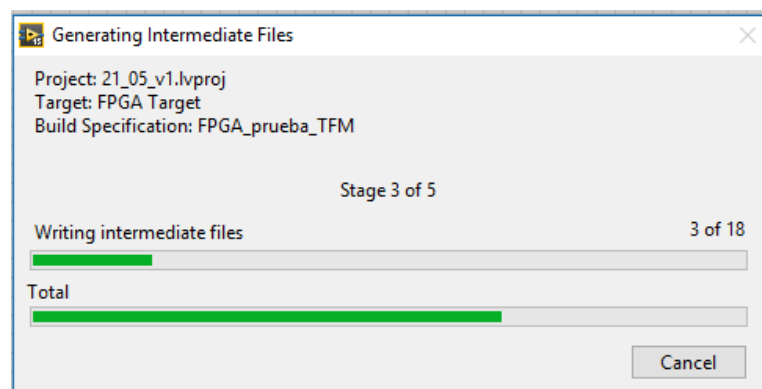


Figura 16: Información de proceso de generación de ficheros intermedios

Si durante este proceso no surge ningún error, se muestra al usuario una ventana de Estado de Compilación que informa del progreso hacia la generación del *bitfile*. En la Figura 17 se muestra el estado, tiempo transcurrido, *reports* que van apareciendo a lo largo de la compilación y posibles alertas. Si la compilación es exitosa, significa que se ha generado el

*bitfile* y está preparado para ser descargado en la FPGA y probar la ejecución de la aplicación desarrollada.

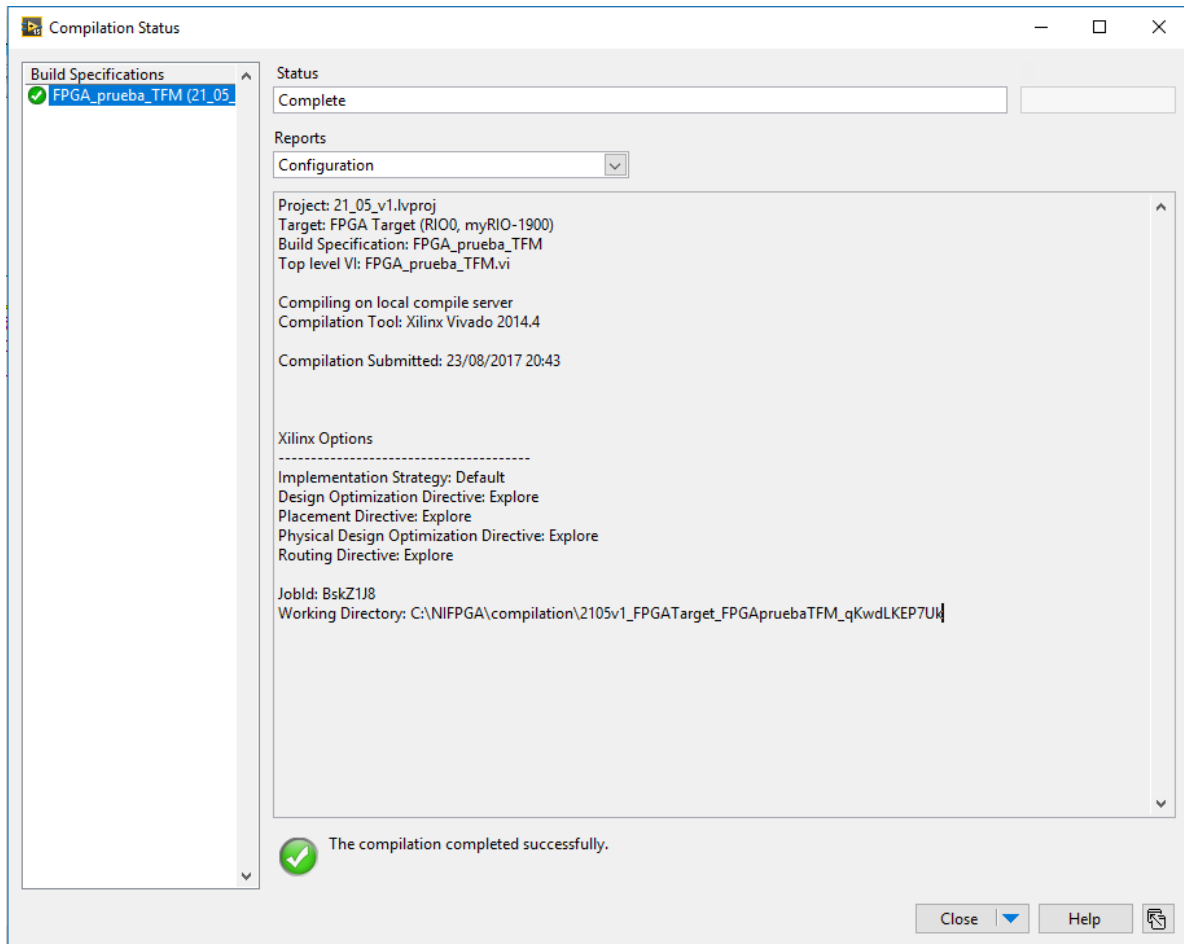


Figura 17: Información de proceso de estado de compilación

### 5.1.3. Sensado, control y actuación en paralelo

Para el sistema de control con la programación en LabVIEW se deben ejecutar en paralelo la generación del PWM, adquisición de datos y procesamiento para los reguladores que controlan el sistema. Una de las principales ventajas de la ejecución hardware es el mejor rendimiento y determinismo frente a la ejecución software basada en procesador.

En la ejecución software basada en procesador, LabVIEW se encarga de la gestión de los múltiples bucles, prioridades y tiempos que determinan cuándo se ejecuta cada función. Esto lleva a una ejecución en serie donde todas las operaciones son manejadas secuencialmente por el procesador, creando interacciones de tiempos entre diferentes partes de la aplicación y generando *jitter* en la ejecución del programa. El *jitter* en la ejecución software es una medida

de la desviación en la ejecución de tareas entre el tiempo que se espera que tarde en ejecutarse una tarea y el tiempo que realmente ha consumido.

Cada aplicación o función que se define en un VI diseñado para ejecución en FPGA se realiza en una estructura de bucle. El diagrama diseñado es mapeado a puertas lógicas y *slices* de la FPGA de forma que cada bucle es implementado en diferentes secciones de la FPGA, permitiendo a todos los procesos ejecutarse simultáneamente, en paralelo. La temporización de cada proceso es independiente del resto, por lo que no existe *jitter*.

Con el uso de programación paralela, la aplicación del VI puede ser particionada en varios bucles, permitiendo el paso de datos entre los bucles y sincronización entre ellos.

El método seguido para la sincronización de bucles en este TFM ha sido la de las ocurrencias, con las que LabVIEW permite bloquear la ejecución de un bloque (con un *Wait for Occurrence*) hasta que otro bucle ha completado su operación y llama a la función *Set Occurrence*. Un ejemplo del funcionamiento se muestra en este diagrama de la Figura 18.

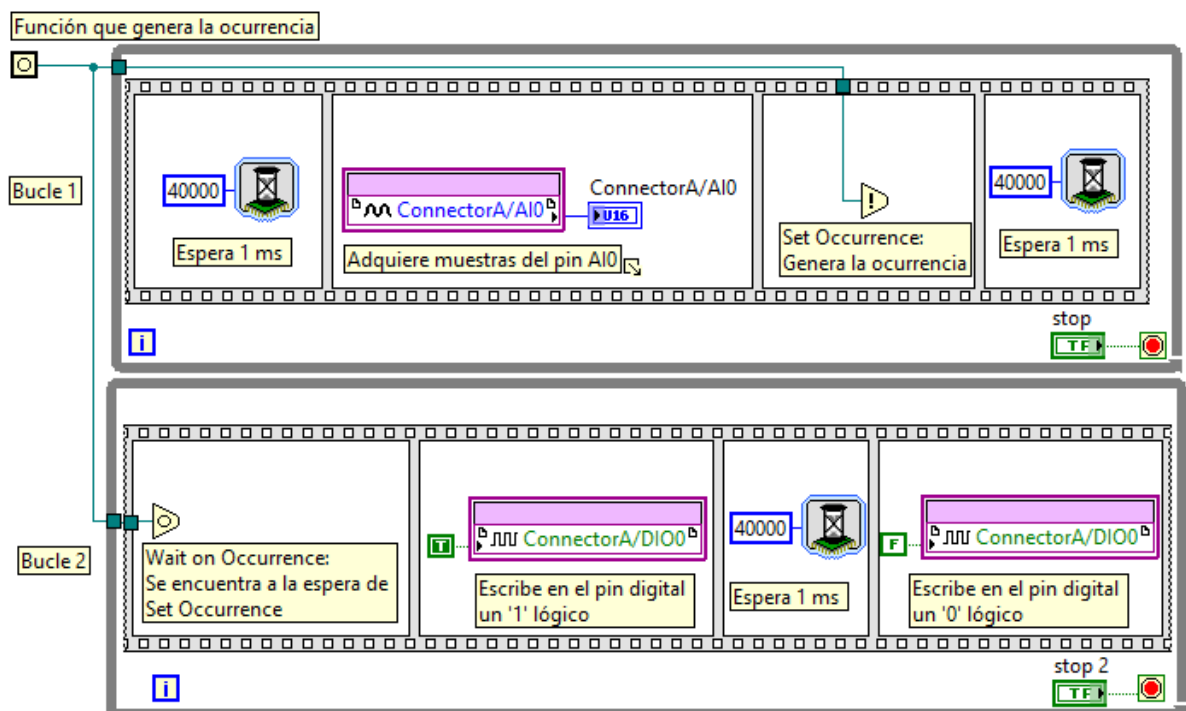


Figura 18: Ejemplo de sincronización de bucles mediante ocurrencias

El reloj de la FPGA es de 40 MHz, por lo que un ciclo de reloj tiene una duración de  $\frac{1}{40 \times 10^6} = 25 \text{ ns}$ . La función de *Wait*, está configurada para *ticks* de reloj, por lo que 40000 *ticks* se corresponden con 1 ms. El primer bucle tarda en ejecutarse por completo 2 ms, por los tiempos de espera que se programan en la secuencia del bucle. Tras el primer milisegundo adquiere

muestras analógicas del pin AIO y genera la ocurrencia que activa el segundo bucle, que escribirá un pulso digital en una salida digital de myRIO durante 1 ms y después lo desactiva. El primer bucle volverá a ejecutarse tras terminar la secuencia con la segunda espera de 1 ms y cuando vuelva a la función *Set Occurrence*, activará de nuevo el segundo bucle y así sucesivamente.

La principal ventaja frente a otros métodos de sincronización, como es el uso de variables globales, es que al permanecer el segundo bucle desactivado hasta que se genera la ocurrencia, se consumen menos recursos y se evita tener que realizar sondeos recursivos de variables hasta que éstas cambien su estado.

#### 5.1.4. Almacenamiento y disponibilidad de datos

En LabVIEW FPGA existen diferentes métodos de almacenar datos durante la ejecución de la aplicación. A continuación, se describen los utilizados durante el desarrollo de este TFM:

- Variables locales: las variables locales almacenan el último valor que se escribe en ellas y solo son accesibles a través de un único VI, suficiente para la aplicación de este TFM, puesto que solo se ejecutará un FPGA VI. Pueden ser configuradas como variables de lectura o escritura y son muy útiles para referenciar datos entre diferentes bucles *while*. En la Figura 19 se muestra un ejemplo de variables de lectura o escritura según sea su referencia un indicador (*Numeric*) o un control (*Boolean*), respectivamente.

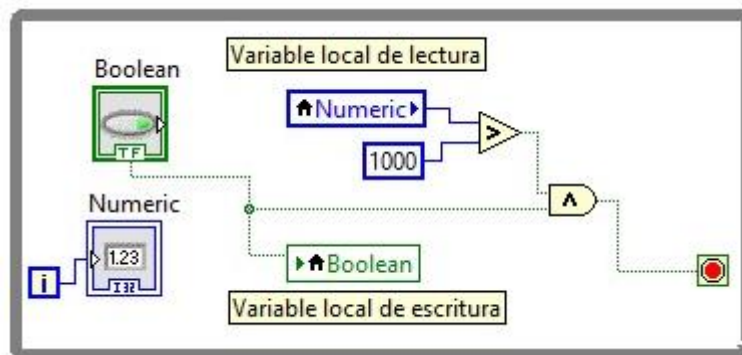


Figura 19: Ejemplo de uso de variables locales

- Feedback nodes y Shift registers: en LabVIEW los nodos de realimentación o registros de almacenamiento almacenan datos de la ejecución de un VI o un bucle hasta la siguiente ejecución o iteración. Por defecto, solo almacenan datos de la

ejecución o iteración previa, pudiendo ser configurados para almacenar más de un único dato incrementando el parámetro configurable de retardo (solo disponible en los *Feedback nodes*). En la siguiente Figura 20 se muestra un ejemplo donde ambos valores al finalizar el bucle tras las 5 iteraciones obtendrán el mismo valor.

Para el desarrollo del FPGA VI se han utilizado múltiples *Shift Registers* para obtener los valores de *Ticks* de ejecución en los bucles y conocer a qué frecuencia se están muestreando los datos y generando el PWM.

También se han utilizado múltiples *Feedback Nodes* para la creación de la ecuación en diferencias de los reguladores implementados para los lazos de tensión y corriente, ya que éstas funciones actúan como bloques de retardo para el procesamiento de señales.

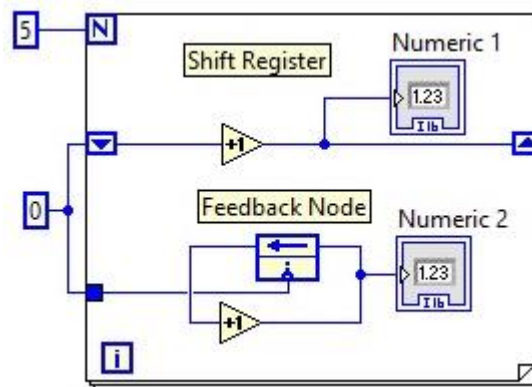


Figura 20: Ejemplo de *Feedback Node* y *Shift Register*

- Elementos de memoria: LabVIEW permite almacenar datos y obtener su valor mediante acceso aleatorio, al contrario de lo que ocurre con los elementos almacenados en estructuras de datos FIFO (*First In, First Out*) o registros. Para el FPGA VI desarrollado y las necesidades del sistema se compararon los 3 tipos de implementación de elementos de memoria disponibles (LUTs, BRAM o DRAM). Se concluyó que el mejor método era la implementación en BRAM, puesto que no consume tantos recursos de lógica, cuya conservación de elementos LUTs ha sido un parámetro crítico a lo largo de todo el desarrollo de la aplicación como se explicará en el apartado 5.1.6.

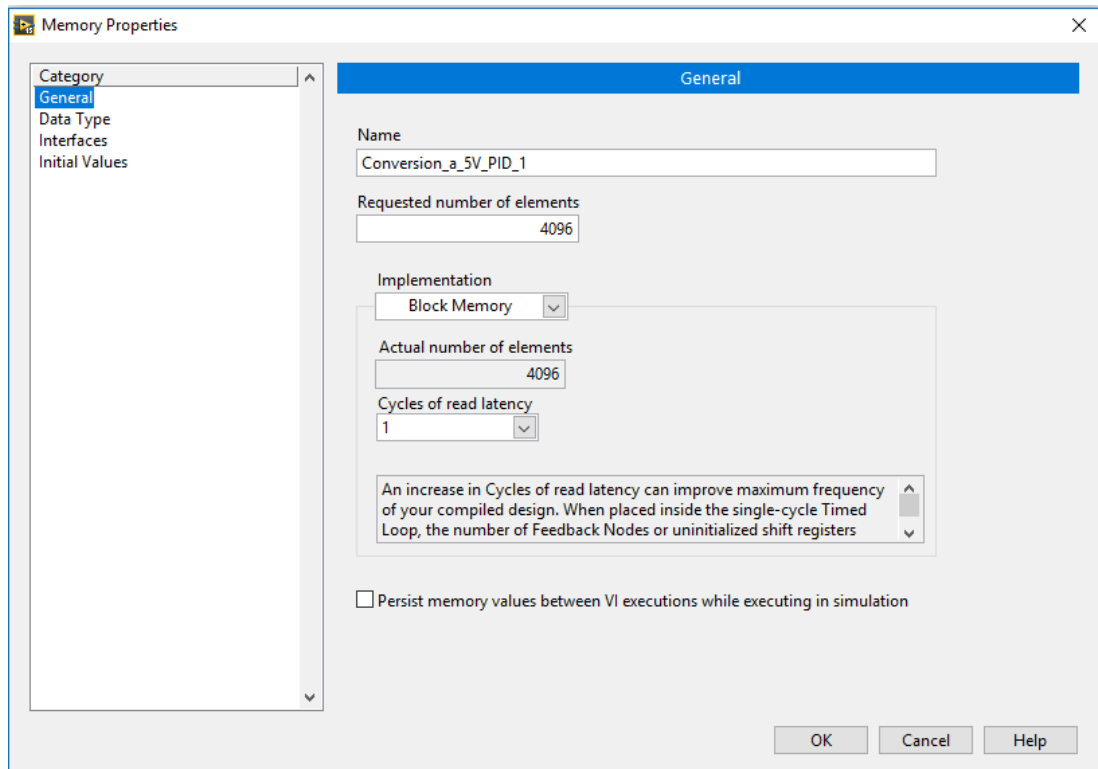
La lectura de los valores analógicos obtenidos en los pines de los puertos de expansión requiere una conversión, que se explica en las especificaciones de myRIO de esta forma:

$$V = \text{Valor de dato sin procesar} \times \frac{\text{Rango nominal}}{2^{\text{bits\_ADC}}}$$

El valor de dato sin procesar en el caso de la aplicación concreta del FPGA VI es el valor que devuelve myRIO de los pines de entradas analógicas en los conectores de expansión MXP, un número entero entre 0 y 4095 (valor máximo representable con salida de 12 bits en el ADC, empezando desde 0). Puesto que el máximo voltaje de entrada en dichos pines es de 5V, el mínimo valor de voltaje que se obtiene es de:

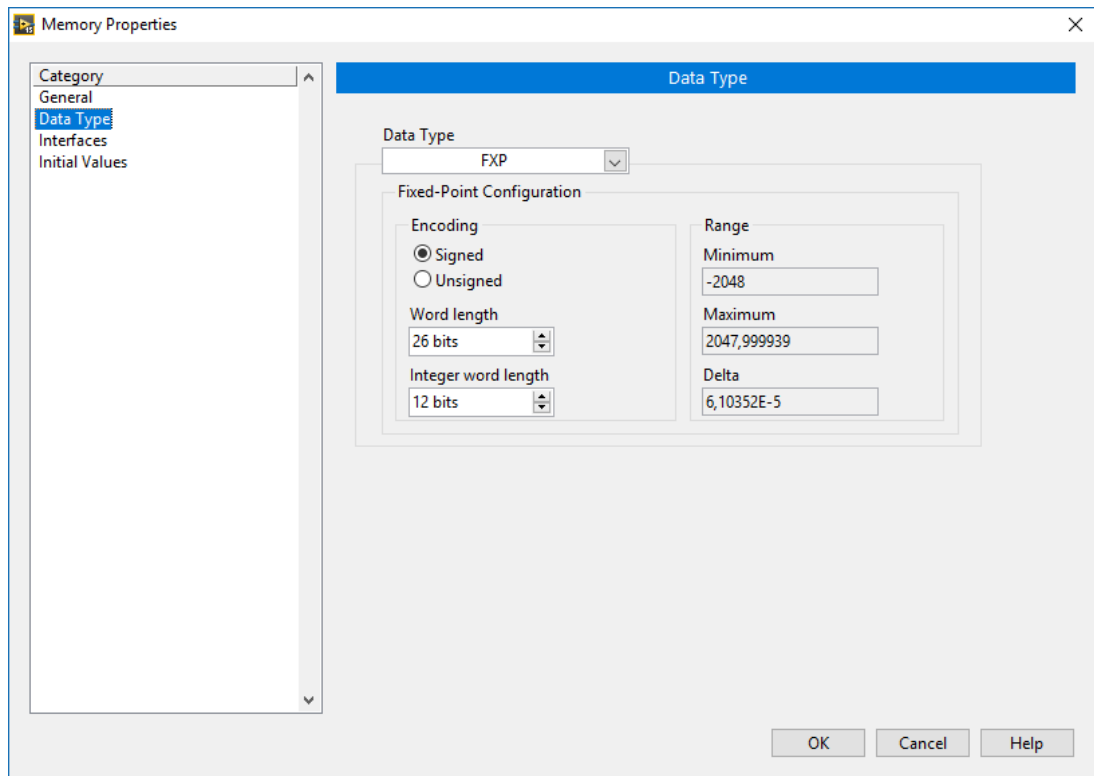
$$\frac{\text{Rango nominal}}{2^{\text{bits\_ADC}}} = \frac{5 \text{ V}}{2^{12}} = 1,221 \text{ mV}$$

Se han creado 3 BRAM de 4096 posiciones. Cada una de las memorias contienen los valores a los que se corresponde cada posición de memoria con el valor convertido a voltaje, de forma que el valor que devuelve myRIO se corresponde con el índice de la dirección de memoria que contendrá el valor real de voltaje tras aplicar las calibraciones. En las siguientes figuras, Figura 21 y Figura 22, se muestra la configuración y valores que se han almacenado:



**Figura 21: Configuración de memoria BRAM (Opciones generales)**

Puesto que las conversiones de los voltajes y corriente contendrán decimales, se ha configurado que los datos almacenados serán de aritmética en coma fija (FXP, *Fixed-Point*), y con los bits mínimos necesarios para almacenar su valor.



**Figura 22: Configuración de memoria BRAM (Tipo de datos)**

Para la aplicación del FPGA VI, las memorias BRAM se inicializan en modo escritura al principio de la ejecución con los valores de calibración que se necesita aplicar para obtener los voltajes y corrientes reales medidas y que se explican en el apartado 5.1.5. En la Figura 23 se muestra cómo se realiza la escritura de los datos. Los bucles *while* tienen un indicador de iteraciones que se incrementa en una unidad cada vez que se ejecuta el bucle. Ese indicador será el índice de la posición de memoria a escribir el dato de voltaje o corriente real obtenido tras la calibración. Puesto que el ADC es de 12 bits, la medida de un valor en los pines analógicos de myRIO se puede representar con hasta 4095 valores, comenzando desde el 0, de ahí el tamaño configurado para la memoria. Para averiguar qué valor se tiene de voltaje real leído en el pin de 0 a 5 V, el contador de iteraciones se multiplica por la constante de incremento de valor leído del ADC, que es de 1,221 mV y para saber el voltaje real correspondiente en el convertidor *boost*, dado que este realiza una adaptación mediante divisores resistivos para obtener valores entre 0 y 1 V, se realizan las operaciones de calibración (mostradas en 5.1.5 y Anexos II.1 Calibración para Voltaje de entrada  $V_{in}$  y II.2 Calibración para Corriente de entrada  $I_{in}$ ) al valor de voltaje leído en el pin del ADC.



La Figura 23 mostrada a continuación es un ejemplo de cómo se ha realizado la inicialización de la memoria para los valores del voltaje de salida. Las otras dos memorias se han inicializado de forma idéntica, simplemente cambiando los valores de las variables de los coeficientes y la variable de ajuste con el osciloscopio, por lo que no se ha incluido figura de su bucle.

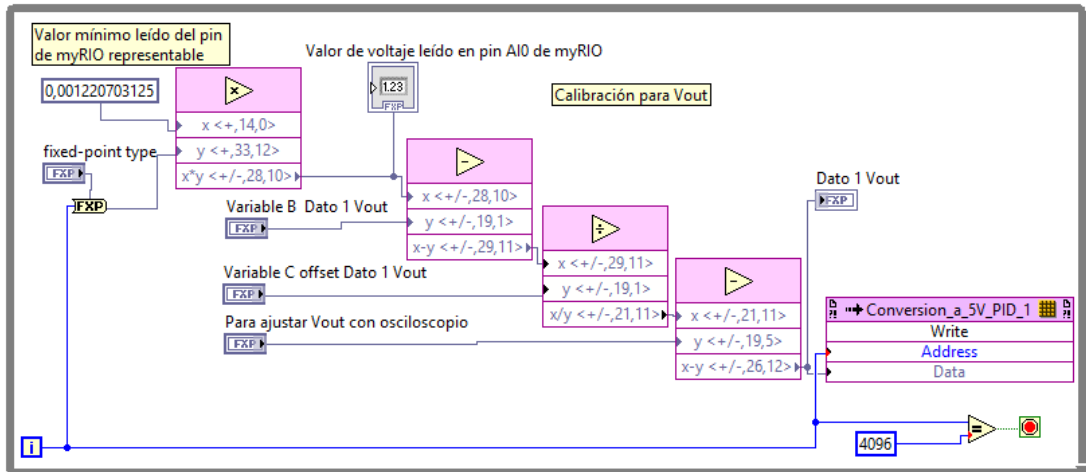


Figura 23: Ejemplo de escritura de datos en memoria BRAM en myRIO

En el modo de lectura, se requiere un ciclo de reloj leer la dirección especificada de la memoria. Por ello, se añade un *Feedback Node* en la salida y se lee el dato en el siguiente ciclo de reloj. Este ciclo de reloj extra se tendrá en cuenta en la frecuencia de ejecución de los bucles, permitiendo que la frecuencia sea un parámetro configurable desde un control de la función *Configure Wait* en el panel frontal del FPGA VI. En la Figura 24 se muestra un ejemplo de lectura de la memoria para obtención de un valor de voltaje leído en el pin AI0 de un conector de expansión de myRIO.

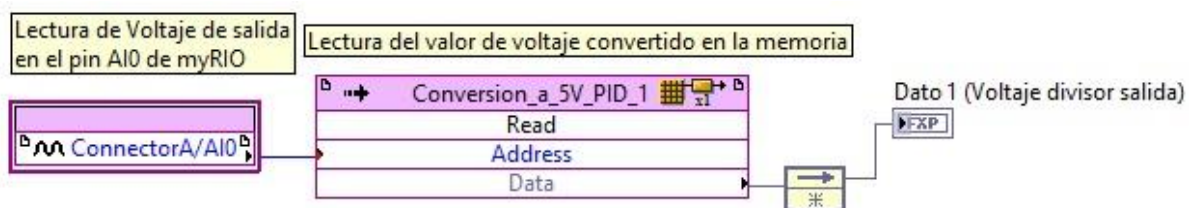


Figura 24: Ejemplo de lectura de datos en memoria BRAM en myRIO

### 5.1.5. Calibraciones

Ha sido necesario realizar calibraciones entre los datos de voltaje y corrientes obtenidas en el convertidor *boost* tras los divisores de tensión y los valores obtenidos en la placa de myRIO

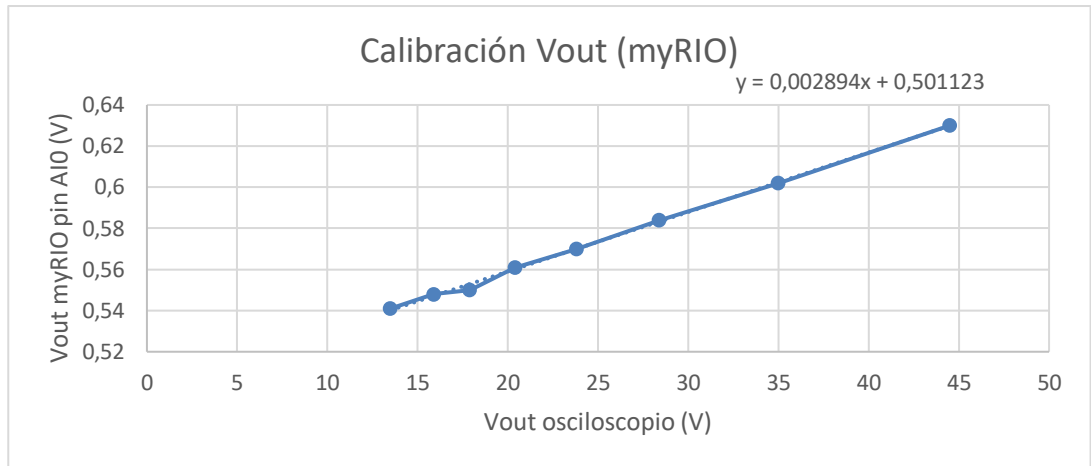
para una correcta obtención de los valores a aplicar en los reguladores. Los datos de estas calibraciones son almacenados en las memorias BRAM, de forma que los valores que devuelven los pines analógicos se corresponden con los índices de la posición de memoria que contiene el valor convertido a voltaje o corriente real.

- Calibración para Voltaje de salida  $V_{out}$ : para un voltaje de 15 V de entrada, se ha suministrado con un generador de señales externo un pulso digital con diferentes valores de *duty cycle* para comprobar los valores leídos por myRIO tras los divisores resistivos y el voltaje de salida obtenido leído con el osciloscopio. Los valores mostrados en la Tabla 2 muestran los valores leídos en los pines de myRIO, el voltaje teórico que se debería obtener y el voltaje real medido con el osciloscopio, observándose que a mayor *duty cycle* se obtienen mayores pérdidas en el convertidor *boost*.

Duty cycle (D)	Vout teórica (V) $V_{out} = \frac{V_{in}}{1 - D}$	Vout Osciloscopio (V)	Vout myRIO pin AI0 (V)	Pérdidas Vout teórica – Vout osciloscopio
0%	15	13,5	0,541	1,5
10%	16,66	15,9	0,548	0,76
20%	18,75	17,9	0,550	0,85
30%	21,42	20,4	0,561	1,02
40%	25	23,8	0,570	1,2
50%	30	28,4	0,584	1,6
60%	37,5	35	0,602	2,5
70%	50	44,5	0,630	5,5

Tabla 2: Datos calibración Voltaje de salida  $V_{out}$  en myRIO

En la Figura 25 se muestra la ecuación de la recta que se debe aplicar para obtener el voltaje real de salida a partir del valor obtenido en myRIO (el que se lee en los divisores de tensión que adaptan el voltaje). Se tomará como ecuación  $y = 0,00289x + 0,50112$ . En este caso, al ser la medida tomada de  $V_{out}$  en el pin AI0 y aplicar la calibración, se obtendrá voltaje real a la salida para restarlo al de la consigna de referencia  $V_{ref}$  seleccionada por el usuario y obtener el valor de error que entrará al primer regulador.

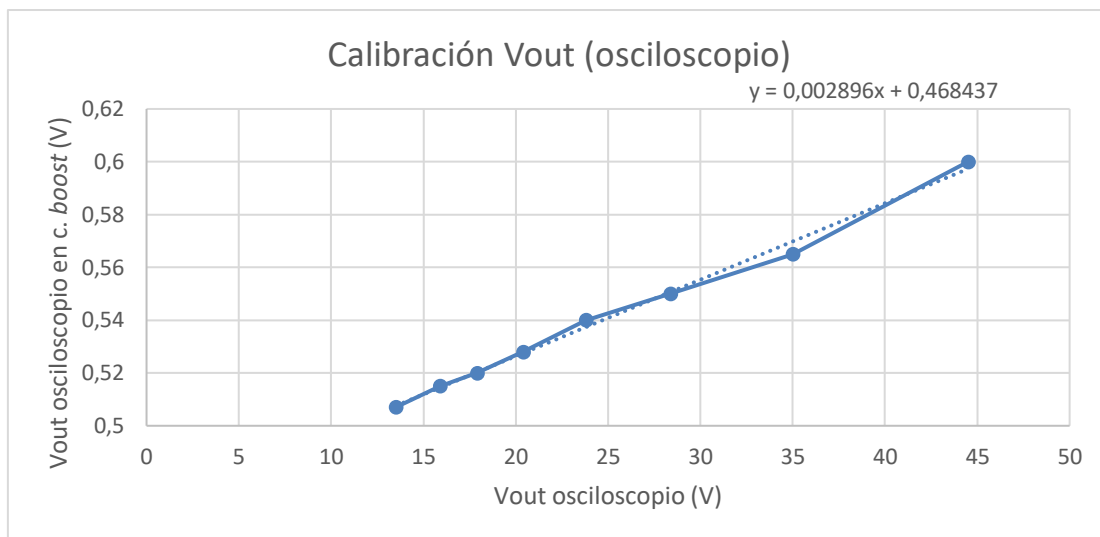


**Figura 25: Recta de calibración para Vout en myRIO**

También se comprobó con el osciloscopio los valores leídos a la salida de los divisores de tensión para comprobar su ajuste con los obtenidos en myRIO y se obtuvieron los siguientes valores mostrados en la Tabla 3 y su recta de calibración (ver Figura 26):

<b>Duty cycle (D)</b>	<b>Vout teórica (V)</b> $V_{out} = \frac{V_{in}}{1 - D}$	<b>Vout myRIO pin A10 (V)</b>	<b>Vout osciloscopio (V)</b>	<b>Pérdidas Vout myRIO – Vout osciloscopio (V)</b>
<b>0%</b>	15	0,541	0,507	0,034
<b>10%</b>	16,66	0,548	0,515	0,03
<b>20%</b>	18,75	0,550	0,520	0,03
<b>30%</b>	21,42	0,561	0,528	0,033
<b>40%</b>	25	0,570	0,540	0,03
<b>50%</b>	30	0,584	0,550	0,034
<b>60%</b>	37,5	0,602	0,565	0,037
<b>70%</b>	50	0,630	0,600	0,03

**Tabla 3: Datos calibración Voltaje de salida Vout en osciloscopio leídos tras divisores resistivos**



**Figura 26: Recta de calibración para Vout en osciloscopio**

Debido a las desviaciones entre los valores observados en el osciloscopio y los leídos con myRIO, fue necesario ajustar un valor de *offset* para que el error de  $V_{ref}$  y  $V_{out}$  fuese lo más preciso posible. Con cada recta de calibración, se obtuvieron los valores (ver Tabla 4) de voltaje real a partir del leído en los divisores de tensión y se calculó la diferencia entre los obtenidos de myRIO y el osciloscopio, permitiendo desde el FPGA VI de las pruebas la modificación del *offset* mediante un control para un mejor ajuste a los valores reales.

Duty cycle (D)	Vout teórica (V) $V_{out} = \frac{V_{in}}{1 - D}$	Valores con recta de calibración		Pérdidas Vout myRIO – Vout osciloscopio
		Vout myRIO (V)	Vout osciloscopio (V)	
0%	15	13,779	13,316	0,463
10%	16,66	16,198	16,078	0,120
20%	18,75	16,889	17,805	-0,916
30%	21,42	20,690	20,567	0,123
40%	25	23,800	24,711	-0,911
50%	30	28,638	28,164	0,474
60%	37,5	34,857	33,344	1,514
70%	50	44,532	45,429	-0,897

**Tabla 4: Datos para ajuste de offset de Vout**

A continuación, se explica y muestra la implementación en LabVIEW. Todos los controles aquí configurados son los dispuestos en el FPGA VI, el cual se ha parametrizado para poder realizar todas las pruebas y obtener los mejores ajustes a los resultados reales. Una vez concluidas las pruebas, los valores quedan fijados para el usuario, de forma que solo se utilice la interfaz del Host VI.

Para obtener el voltaje real leído, se debe aplicar la calibración previamente obtenida. La ecuación para  $V_{out}$  se interpreta de la siguiente forma:  $y = Bx + C$ , donde en este caso,  $B = 0,00289$  y  $C = 0,50112$  y el valor de *offset* tras varias pruebas y las observaciones de la Tabla 4, se fija a  $-0,9$ . En la Figura 27 aparece el panel frontal del FPGA VI que permite su configuración y en la Figura 28 se muestra el diagrama de bloques correspondiente. La calibración es aplicada y guardada para cada uno de los posibles valores leídos en myRIO como se explicó en el apartado 5.1.4.

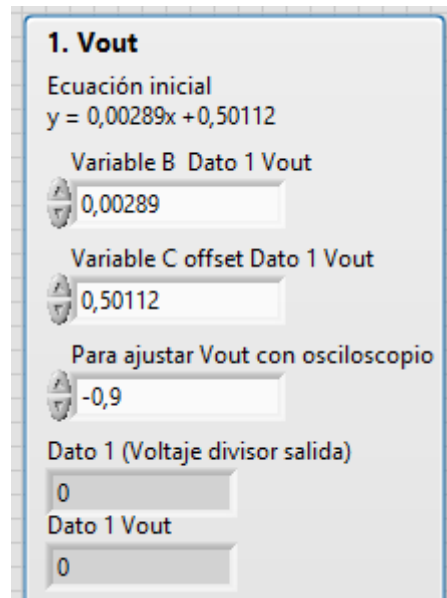


Figura 27: Calibración para Vout en FPGA VI (panel frontal)

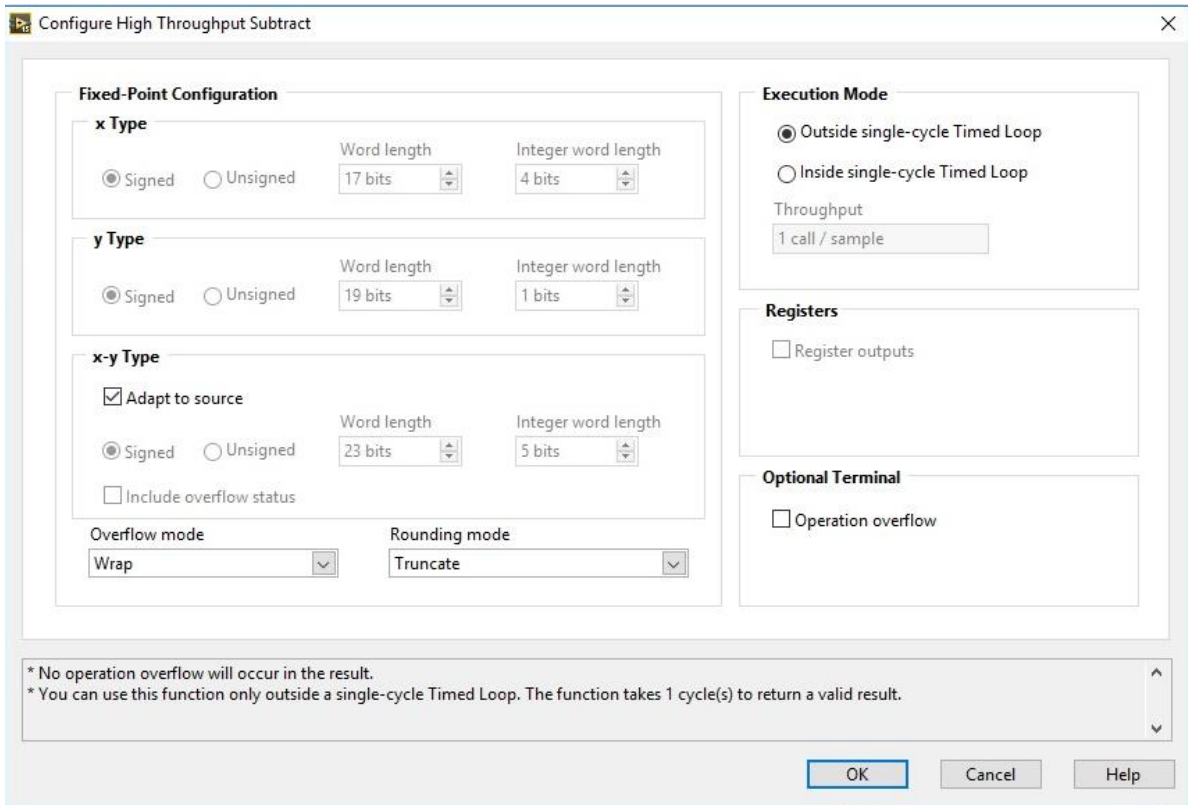


Figura 28: Calibración para Vout en FPGA VI (diagrama de bloques)

Las funciones con las que se realizan los cálculos son funciones *High Throughput*, especialmente diseñadas para reducir el uso de elementos lógicos en la FPGA y disminuir los recursos utilizados durante el enrutamiento para el cálculo con números en coma fija a través de su configuración. En cada control y operación se ha seleccionado el número de bits necesarios para la parte entera y digital de forma lo más precisa posible para optimizar la lógica disponible en la FPGA.

La Figura 29 muestra como ejemplo las opciones de configuración de una operación de sustracción, así como la información de cuántos ciclos tardará en obtener el valor del resultado utilizando la función *High Throughput*. Se ha configurado:

- Modo de *overflow* o desbordamiento: sucede cuando el resultado de la operación es mayor o menor que el valor máximo o mínimo representables, respectivamente. Se ha seleccionado el modo *Wrap*, donde se descartan los bits más significativos del resultado hasta que su valor esté dentro del rango definido. El otro modo permitido es el de saturación (*Saturate*), pero este modo requiere más recursos de la FPGA para detectar, reportar y saturar los números cuando se da un caso de *overflow*.
- Modo de redondeo: sucede cuando la precisión esperada de un valor es mayor que la precisión del tipo de dato que representa, de forma que se fuerza a un valor que conlleva una pérdida de precisión. El modo de truncamiento (*Truncate*) es el recomendado, dado que elimina los bits menos significativos y no requiere recursos de adicionales en la FPGA para realizarlo con respecto a los de redondeo hacia arriba y hacia abajo (*Round-Half-Up* y *Round-Half-Down*).
- Modo de ejecución: se ejecutará fuera de un SCTL (*Single-Cycle Timed Loop*), que es un bucle especial que solo tarda 1 ciclo de reloj en ejecutarse.



**Figura 29: Ejemplo de configuración en funciones *High Throughput***

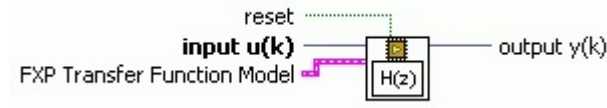
Los pasos para obtener las calibraciones de  $V_{in}$  e  $I_{in}$  son iguales a los aquí descritos en  $V_{out}$ , y se pueden consultar en los Anexos II.1 Calibración para Voltaje de entrada  $V_{in}$  y II.2 Calibración para Corriente de entrada  $I_{in}$

### 5.1.6. Reguladores

Para la corrección del f.d.p., son necesarios dos reguladores, uno para el lazo de tensión (lazo lento) y otro para el lazo de corriente (lazo rápido). Con las características del convertidor *boost*, se generó en Matlab el modelo de la planta y varios reguladores han sido analizados con la herramienta de diseño SISO (*Single-Input Single-Output*), también en Matlab. El análisis y los datos se muestran en el Anexo III.1 Implementación de planta del convertidor *boost*.

La implementación de los reguladores en LabVIEW ha sido un punto crítico para los recursos disponibles en la FPGA. Se abordaron diferentes estrategias que se mostrarán a lo largo de este apartado hasta que se encontró la solución más óptima.

Inicialmente se utilizó un sub-VI predefinido del módulo FPGA de LabVIEW *Discrete Transfer Function Direct VI*, que implementa en coma fija una FDT (Función de transferencia) de un modelo de sistema en la FPGA.



**Figura 30: VI de FDT Directa del módulo FGPA en LabVIEW**

La Figura 30 muestra las 3 entradas y salida disponibles:

- *Reset*: devuelve al sistema a las condiciones iniciales.
- *Input  $u(k)$* : es la entrada al modelo. En un sistema con realimentación (*feedback*), se corresponde con la diferencia entre la referencia de entrada y la medida que se realiza en la planta.
- *FXP Transfer Function Model*: especifica el modelo que implementa dicho VI. En caso de que el modelo del sistema se encuentre en representación en coma flotante, se debe utilizar el VI de *Discrete FP Transfer Function to FXP* para obtenerlo en coma fija. Cada elemento de *FXP Transfer Function Model* se representa mediante tipo de dato con signo con 32 bits de los cuales 12 son enteros (<+/-, 32, 12>).
- *Output  $y(k)$* : devuelve la salida del modelo.

Los reguladores obtenidos son representados mediante FDTs, que son modelos matemáticos que caracterizan la relación de entrada y salida de un sistema a través de ecuaciones lineales. La FDT de los reguladores en LabVIEW está representada en el dominio discreto con la transformada Z, con la entrada definida por  $X(z)$  y la salida definida por  $Y(z)$ , de la siguiente forma:

$$H(z) = \frac{Y(z)}{X(z)} = \frac{B_0 + B_1 z^{-1}}{A_0 + A_1 z^{-1}} = \frac{B_0(z + \frac{B_1}{B_0})}{A_0 z + A_1}$$

En LabVIEW, los coeficientes  $A_0$ ,  $B_0$  y  $A_1$  y  $B_1$  son los que se deberán introducir en el control de *FXP Transfer Function Model* que se muestra al usuario en el FPGA VI de la forma que muestra la Figura 31:



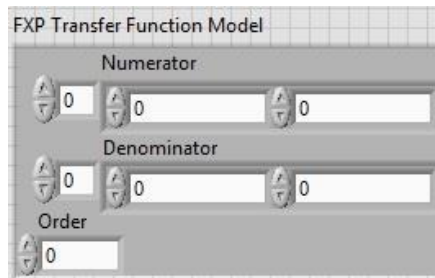


Figura 31: Control en FPGA VI de modelo FDT para *Discrete Transfer Function Direct VI*

En el numerador se deben indicar los coeficientes  $B_0$  y  $B_1$  y en el denominador,  $A_0$  y  $A_1$  y el orden de la FDT, que para ambos reguladores diseñados serán de orden 1. Los coeficientes deben ser introducidos en coma fija, y por defecto el valor configurado del control es de 32 bits, con 16 bits de número entero ( $\langle 32, 16 \rangle$ ). Para optimizar los recursos y utilizar únicamente los bits necesarios, se configura cada uno de los coeficientes según el valor que se ha definido en los reguladores, dando cierto margen en la resolución de los decimales para que se puedan probar otros coeficientes de diferentes reguladores. En la Figura 32 se muestra el panel de configuración para el tipo de dato FXP y la Tabla 5 recoge los valores de *Word length* e *Integer Word length* configurados.

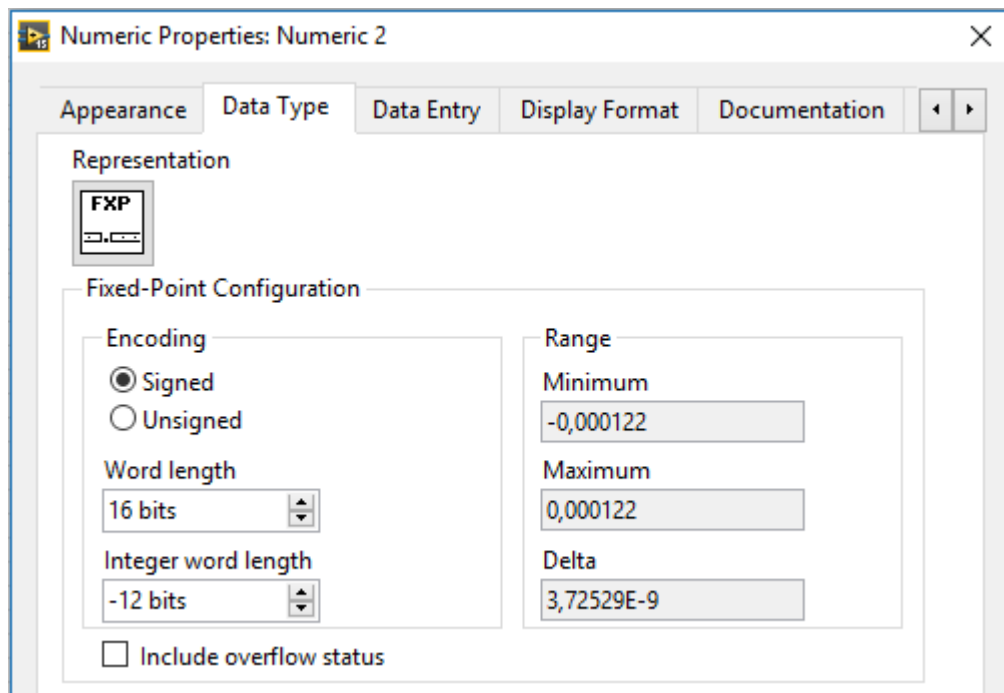


Figura 32: Panel de configuración para tipo de dato FXP

Regulador	Coefficientes	Configuración FXP (bits)
Lazo de Tensión $R1(z)$	$B_0 = 7,206 \times 10^{-6}$	$\langle 16, -12 \rangle$

$R_1(z) = \frac{B_0z + B_1}{A_0z + A_1}$	$A_0 = 1$	<2,2>
	$B_1 = -4,213 \times 10^{-6}$	<16, -12>
	$A_1 = -1$	<2,2>
<b>Lazo de Corriente <math>R_2(z)</math></b>  $R_2(z) = \frac{B_0z + B_1}{A_0z + A_1}$	$B_0 = 0,28$	<16, 2>
	$A_0 = 1$	<2,2>
	$B_1 = -0,2625$	<16, 2>
	$A_1 = -1$	<2,2>

Tabla 5: Configuración de tipo de dato FXP de los reguladores

En la Figura 33 se muestra cómo quedan configurados en el panel frontal del FPGA VI.

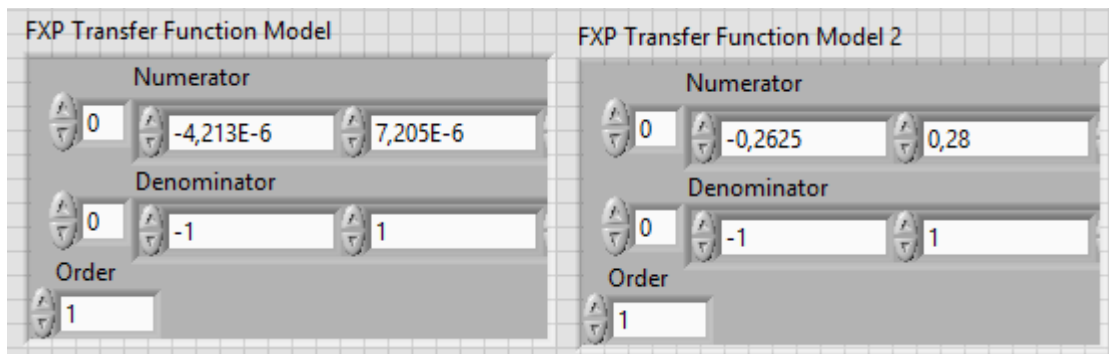
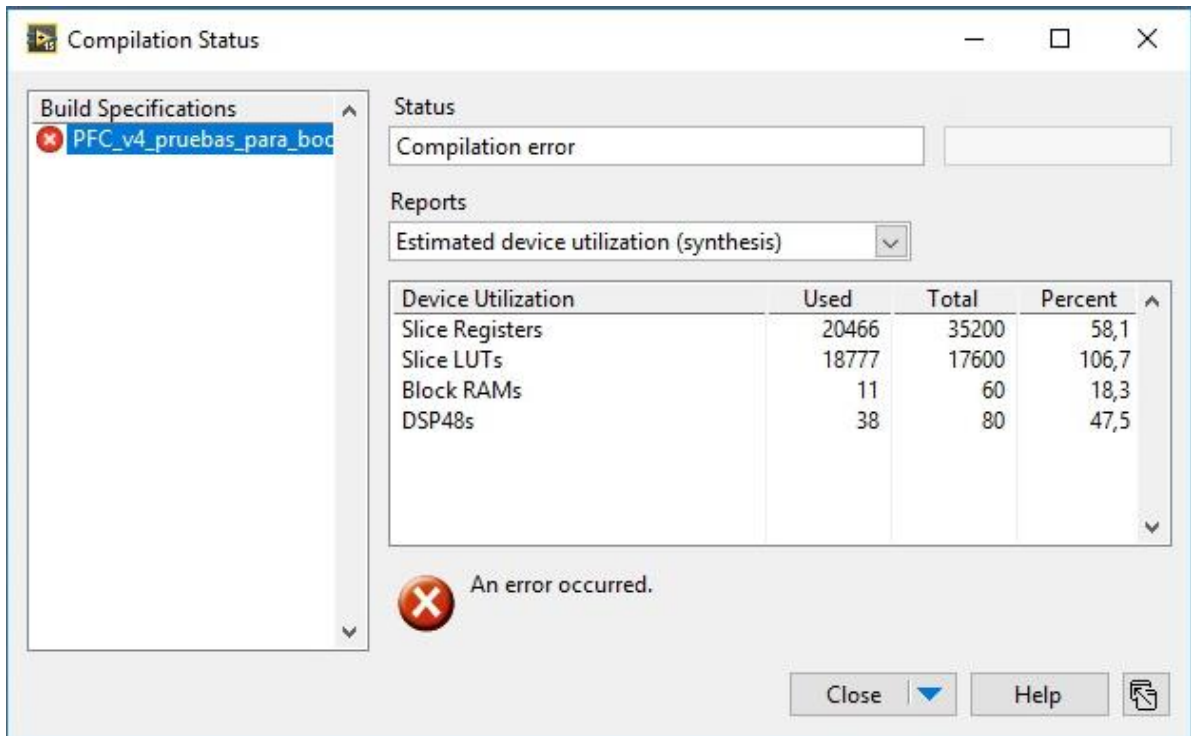


Figura 33: Configuración de los reguladores en el control del panel frontal FPGA VI.

Con esta primera configuración, se intentó realizar la programación de la FPGA, pero durante la etapa de síntesis, se obtuvieron errores por no tener el dispositivo suficientes *slices* para implementar el diseño, como muestra la Figura 34:



**Figura 34: Error de compilación con reguladores implementados con *Discrete Transfer Function Direct VI***

Se hicieron pruebas con un VI sin ninguna funcionalidad programada en él y en los resultados de compilación se obtuvo que únicamente con esa funcionalidad, se estaba utilizando ya casi la mitad de LUTs disponibles en la FPGA (ver Figura 35). Esto se debe a que de la FPGA de fábrica viene pre-configurada para utilizar fácilmente otros recursos tanto desde el módulo NI Real-Time como del de FPGA, como se muestra en las especificaciones y eso hace que, en cada síntesis, también configure diversos recursos por si se da el caso de que el usuario quiera ejecutarlos en el Host VI en tiempo real.

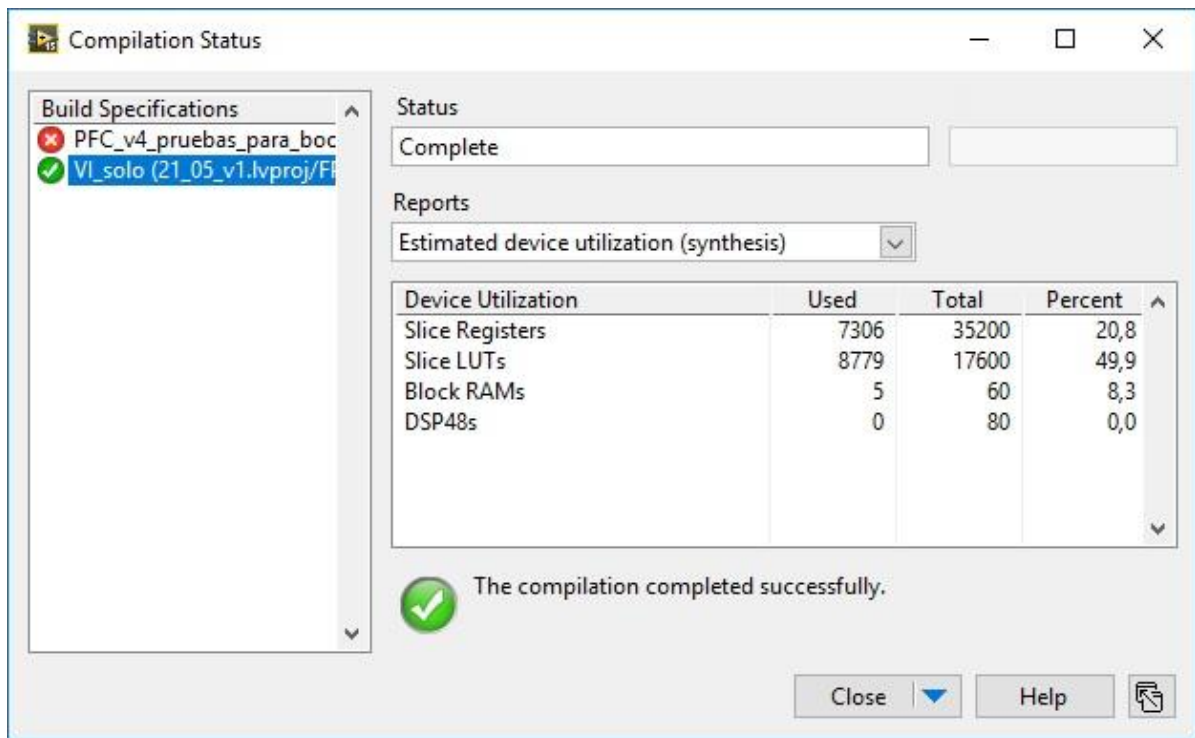


Figura 35: Estimación de uso de recursos en un FPGA VI vacío

Tras esta limitación, se optó por aplicar técnicas de optimización descritas en [16], como la de minimizar el uso de elementos del panel frontal del FPGA VI, puesto que cada elemento que se encuentra en dicho panel crea un registro para permitir al Host comunicarse e interactuar con la FPGA, por lo que se consumen más recursos del chip. Las mejoras que se realizaron en este punto, más algunas previamente adoptadas fueron:

- Se redujo el uso de controles disponibles en el panel frontal de la aplicación.
- Los coeficientes de los reguladores de *Discrete Transfer Function Direct VI* se pusieron en el diagrama de bloques como constantes.
- Se simplificaron datos para que hubiese el mínimo número de operaciones matemáticas (divisiones y multiplicaciones mayoritariamente).
- Se hizo uso de las memorias BRAM para evitar el uso de LUTs.
- Se reemplazaron algunas operaciones aritméticas por cálculos binarios.
- Se adaptaron todos los controles y variables al mínimo número de bits necesarios para su representación y resolución en los cálculos.

Con estas modificaciones se permitió generar el archivo de programación sin errores por exceso de uso de recursos, pero el uso de LUTs utilizadas era del 84,2% y el total de *slices* utilizados tras realizar el proceso de *placement* alcanzaba el 99,8% (ver Figura 36).

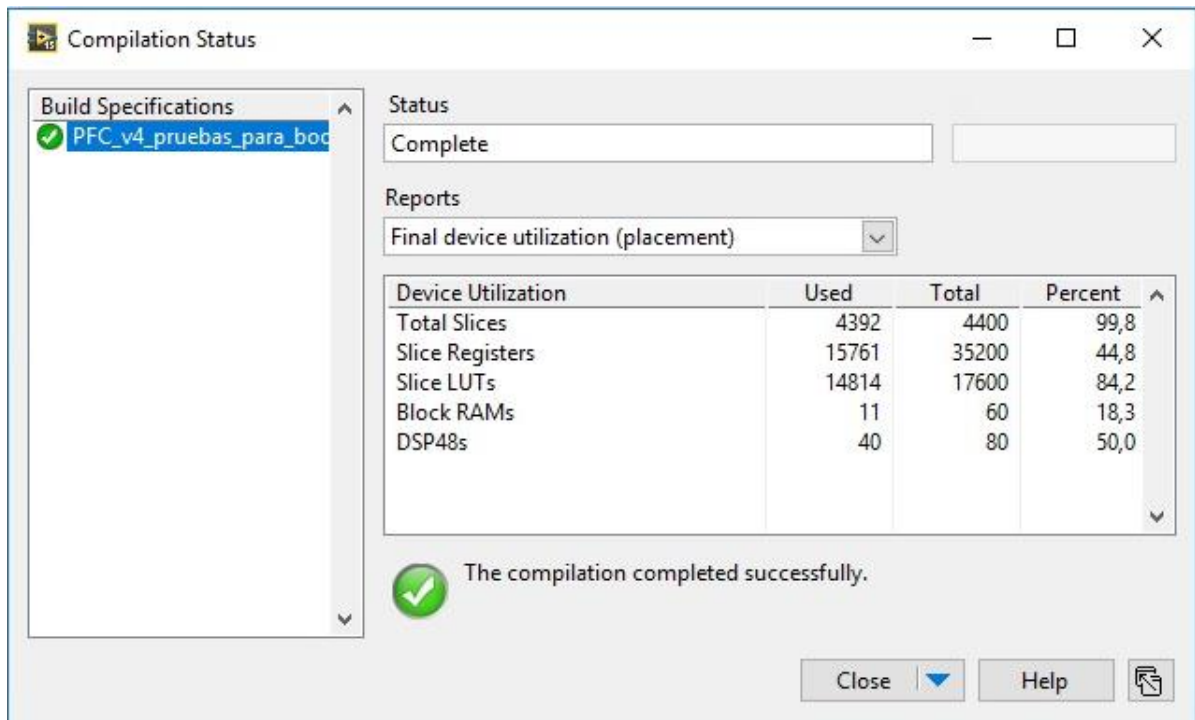


Figura 36: Uso de recursos de la FPGA tras adoptar técnicas de optimización

Por último, la última optimización que se ha realizado es evitar el uso de sub-VIs, como es el caso del uso de *Discrete Transfer Function Direct VI* para la obtención de la actuación de cada lazo del convertidor *boost*.

Las FDT obtenidas para los reguladores, también pueden ser definidas mediante ecuaciones en diferencias como el siguiente ejemplo, donde  $y(k)$  representa la salida del sistema en el instante de tiempo  $k$ , y  $x(k)$  la entrada también en dicho instante:

$$y(k) = y(k - 1) + B_0x(k) + B_1x(k - 1)$$

En la siguiente Figura 37 se muestra un ejemplo de cómo se ha realizado la implementación de la ecuación en diferencias para los reguladores. La entrada para el primer regulador (lazo de tensión) será la diferencia entre el voltaje de referencia  $V_{ref}$  y el voltaje de salida Dato 1  $V_{out}$  (lectura del pin AI0 de la FPGA con calibraciones aplicadas). Se realizan las multiplicaciones de los coeficientes del regulador y se suman todos los términos. Para los términos retardados, se utilizan *Feedback nodes* para obtener las muestras previas necesarias para los cálculos.

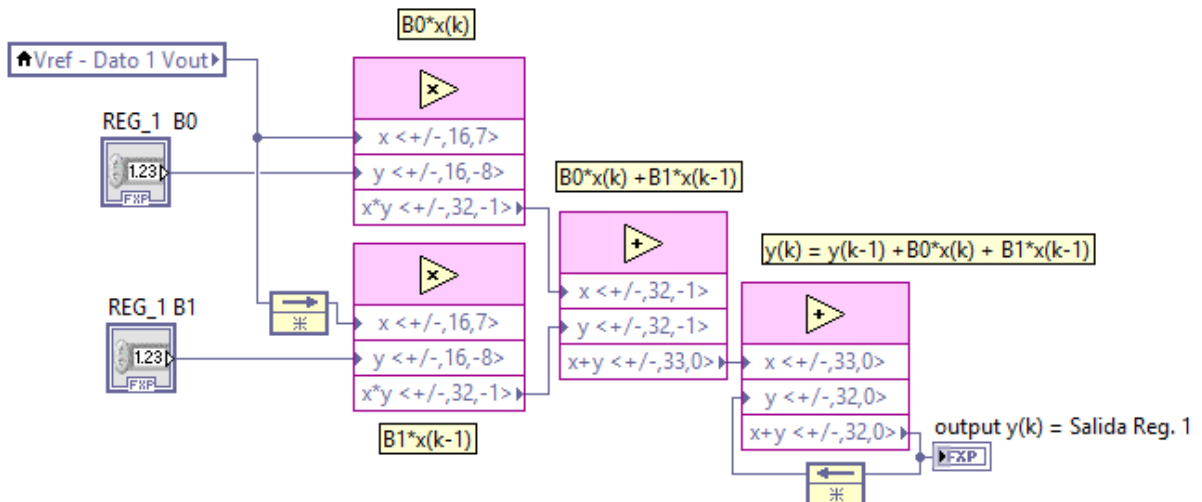


Figura 37: Implementación de ecuación en diferencias en LabVIEW

Una vez implementados los reguladores con las ecuaciones en diferencias, se procedió la compilación del diseño, donde esta vez se obtuvo una notable mejoría respecto a los otros casos en el uso de recursos de la FPGA, con una ocupación del 68,8% de LUTs y el 93,4% de *slices* (ver Figura 38). Estos datos son los de la ocupación con todo el diseño finalizado y con el que se han realizado las pruebas finales.

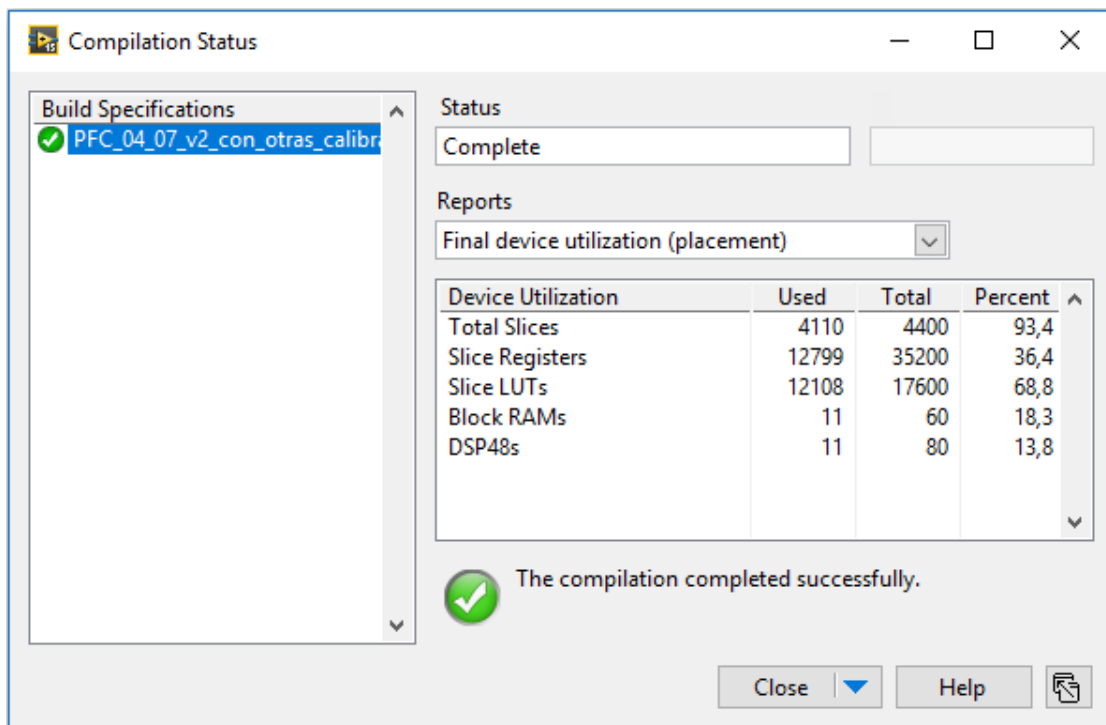


Figura 38: Uso de recursos de la FPGA final

## 5.2. Descripción de la ejecución

En este apartado se muestran los diferentes módulos y bucles que se han desarrollado para la ejecución final del control digital. Se puede dividir en dos apartados, el de implementación en el FPGA VI y la interfaz dispuesta para el control por parte del usuario en el Host VI.

### 5.2.1. FPGA VI

El diagrama de bloques que contiene este VI es el que será programado en la FPGA, ejecutándose de forma continua. Su análisis se ha dividido en 3 bloques principales que están relacionados mediante las ocurrencias entre cada uno de ellos. Estos bloques son:

- Generación del PWM.
- Control del lazo de tensión.
- Control del lazo de corriente.

### Generación del PWM

La generación del PWM para el control del convertidor *boost* viene dada por la salida del regulador del lazo de corriente. Para su implementación en LabVIEW, se han creado diferentes bucles *while*.

El primer bucle está implementado mediante una secuencia interna, donde la primera parte mantiene el pulso digital a nivel lógico de '0' y la segunda parte lo activa siempre y cuando se cumplan las siguientes condiciones:

1. Se ha activado en el Host VI la generación del PWM mediante el control *Output Enable*.
2. El voltaje de referencia que se ha escogido desde el Host VI es superior a 0 V.

Para observar que la señal de control se está generando adecuadamente a la frecuencia de 100 kHz se ha creado a través de un *Shift Register* el conteo de número de *ticks* total que dura un ciclo de PWM, que se corresponde con el valor de 400 *ticks*. Esto implica que la suma de *Low Period (Ticks)* y *High Period (Ticks)* no podrá ser superior a 400 *ticks*. Cada *tick* con el reloj de la FPGA que contiene myRIO (40 MHz), equivale a 25 ns. La comprobación de que dicho bucle se ejecuta cada 400 *ticks* ha sido comprobado mediante un indicador en el panel frontal, así como con mediciones en el osciloscopio de la salida digital DIO8 que genera el PWM.

El control del ciclo de trabajo viene dado por las funciones de *Configure Wait*, con los valores de las variables *Low Period (Ticks)* y *High Period (Ticks)*, cuyos valores se obtienen de los dos siguientes bucles. La ocurrencia generada en la segunda parte de la secuencia será la que active el bucle del lazo de corriente, que se analizará en el siguiente apartado.

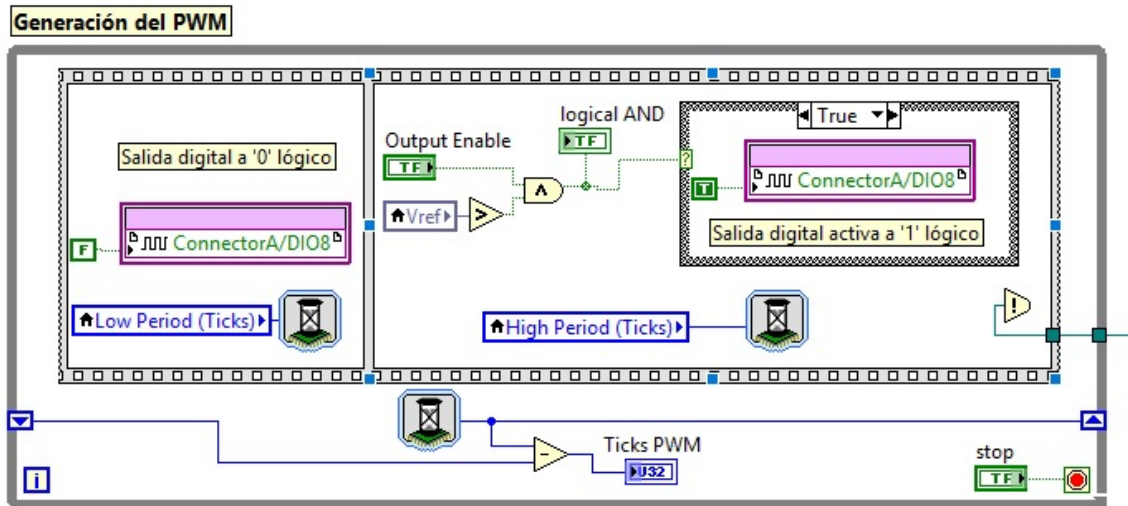


Figura 39: Activación de pulso PWM y *duty cycle*

El segundo bucle es el contador para el PWM. Se ha realizado un contador de 12 bits, con 4096 valores posibles. El bucle se ejecuta con cada ciclo de reloj e indefinidamente, ya que (no se ha puesto el control de *Stop*). Por cada ciclo de reloj, dicho contador se incrementa en una unidad siempre y cuando no se haya alcanzado el límite del contador y la variable *Reset contador* no tenga valor de *True*. En tal caso, el contador se vuelve a iniciar en 0.

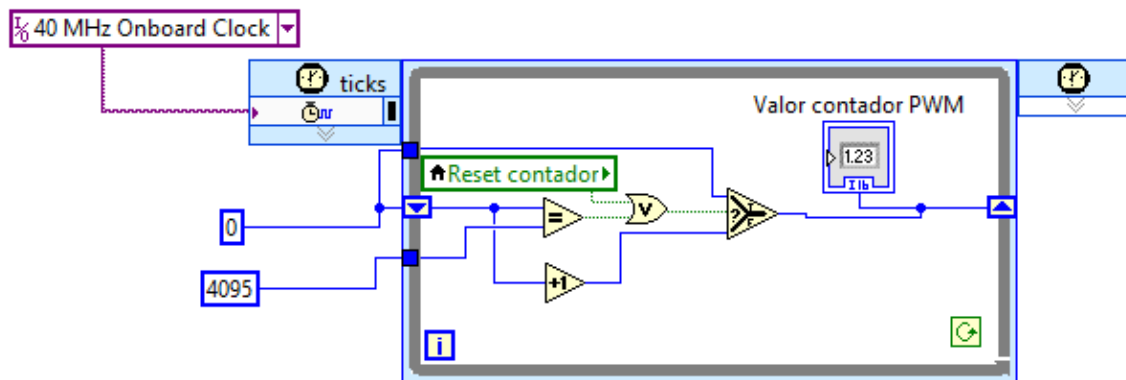


Figura 40: Contador para generación del PWM

La variable *Reset contador* se evalúa en el último y tercer bloque para la generación del PWM. En dicho bucle (ver Figura 41), con la salida del regulador del lazo de corriente se obtiene al dividir por la resolución del contador ( $\frac{1}{2^{12}-1} = 2,4414 \times 10^{-4}$ ) un valor para comparar. Si el



valor que tiene el contador del PWM (Figura 41) es mayor o igual que el obtenido a partir del dato de salida del regulador y la conversión, se resetea el contador. A partir de esa misma salida, se obtienen los *ticks* para el periodo de PWM activo y desactivado.

Partiendo de la base de que 400 *ticks* se corresponden con los 100 kHz y a su vez con el valor máximo de contador del regulador, de la división de  $\frac{4095}{400} \sim 10,25$  se obtiene el número de *ticks* (valor que se redondea al entero más cercano, ya que los *ticks* o ciclos de reloj solo pueden ser evaluados como enteros en la FPGA). Tras la conversión, se utiliza la función *In Range and Coerce*, donde en el terminal inferior se limita al valor de 0 *ticks* y en el superior, se establece el límite de *ticks* para el *duty cycle*. Como se comentó en apartados previos, el *duty cycle* no podrá llegar a tener un valor unitario (ciclo de trabajo del 100%), ya que necesita periodos de descarga del convertidor *boost*. La limitación de *duty cycle* para las pruebas se establece a un 97,5%. Si los *ticks* obtenidos se encuentran dentro del rango del límite para *duty cycle* y el valor 0, se tomará dicho valor como el valor calculado de *High Period (Ticks)* que se corresponde con la salida de los cálculos previos, si no, se fuerza al máximo valor permitido (al 97,5%, que se corresponde con un máximo de 390 *ticks* permitidos a PWM activo, salida digital DIO8 activada a '1', de forma que 10 *ticks* sean a '0').

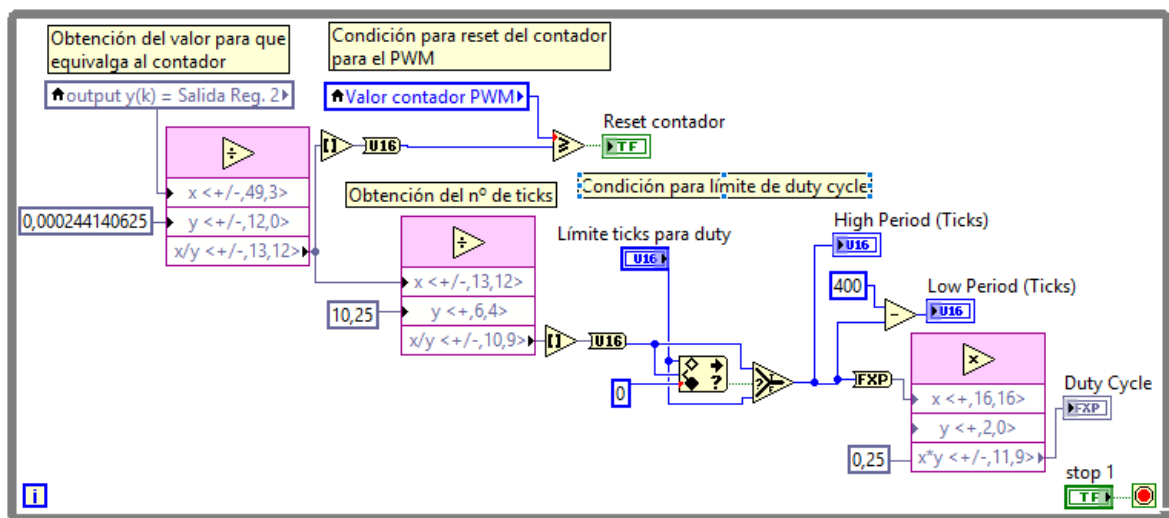


Figura 41: Generación de *High* y *Low Period Ticks* para el PWM

## Control de Lazo de tensión

El lazo externo de tensión es el que regula la tensión media de salida del convertidor *boost*. Se encarga de mantener la tensión de salida al valor de voltaje deseado, requiriendo que la corriente de entrada aporte la potencia necesaria para alcanzar  $V_{ref}$ . Este lazo genera un

comando de potencia que, multiplicado por el voltaje a la entrada del convertidor  $V_{in}$ , genera la entrada al regulador del lazo de corriente.

El control del lazo de tensión debe ser mucho más lento que el de corriente para no introducir distorsión en este último por lo que el bucle de lectura del dato  $V_{out}$  se estará ejecutando continuamente, sin recibir ocurrencias de ninguno otro para su ejecución. En la Figura 42 se muestra cómo se implementa mediante una secuencia dentro de un bucle *while* la lectura del dato real desde el valor almacenado en la memoria cada 100 Hz (10 ms, debido a la rectificación de la señal de entrada).

Para realizar la lectura cada 10 ms ha sido necesario establecer como control la función *Configure Wait*, puesto que como se comentó previamente, la lectura de la memoria BRAM tarda un ciclo de reloj en obtener el dato además de que se debe generar la espera de 10 ms con dicho control. El valor del control *Configure Wait* se ha establecido en *ticks* de forma que se contrarresten esos ciclos de latencia dentro del bucle.

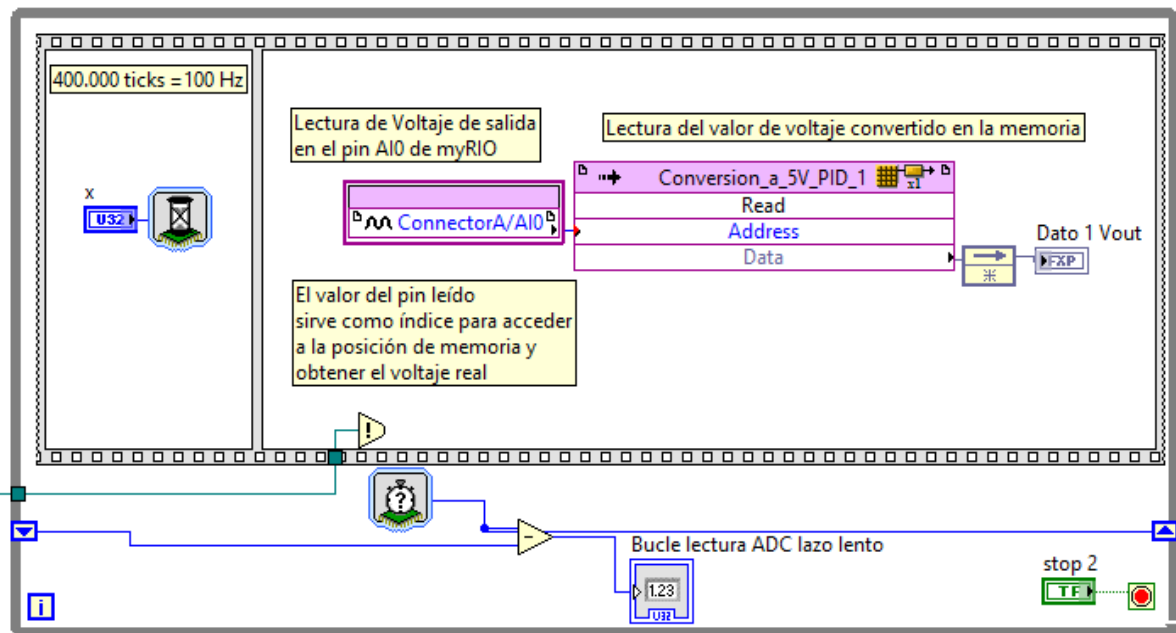
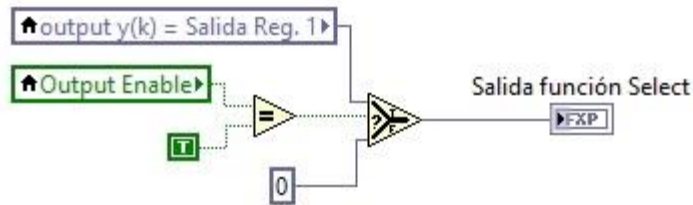


Figura 42: Control de Lazo de tensión: Lectura  $V_{ref}$

El siguiente bucle del control de lazo de tensión (Figura 44) se activa con la ocurrencia del anterior bucle, cuando el dato de  $V_{out}$  ha sido leído. Este bucle se encarga de:

- Control de valores de entrada y salida al regulador: se tiene en cuenta que si no se ha activado la generación del PWM mediante el control situado en el host VI, no deben realizarse cálculos con las referencias de entrada al regulador y la salida debe ser nula. Este control se realiza mediante funciones *Select*. La función *Select* (ver Figura 43)

devuelve uno de los dos valores asignados en los extremos según sea el valor de la condición booleana (designada con el signo de interrogación). Si la condición es verdadera, devolverá el valor de la parte superior, en este ejemplo, la salida del regulador (*True*) y en caso de que la condición sea falsa, devolverá el valor de la parte inferior (*False*), el valor 0.



**Figura 43: Función Select**

- Obtener la referencia de entrada al regulador mediante la diferencia entre el voltaje al que se quiere controlar el lazo  $V_{ref}$  que escoge el usuario desde el panel frontal y la lectura del voltaje medido a la salida del convertidor *boost*.
- Realizar los cálculos del regulador que se ha implementado mediante la ecuación en diferencias explicada en 5.1.6.
- Control de la saturación del regulador: inicialmente, el error obtenido será positivo ( $V_{ref} > \text{Dato 1 } V_{out}$ ), que con la acción integral del regulador PI generará una acción de control creciente que alcanza su valor máximo cuando el error se convierte en nulo. La salida del regulador permanecerá saturada en dicho punto debido a que la acción integral todavía es elevada y hasta que el error no sea negativo o nulo suficiente tiempo, la acción de control no saldrá de la saturación y empezará a descender. Para evitar que la acción del regulador se incremente o disminuya de forma infinita, se ponen límites para la saturación de la acción del regulador, de forma que su salida siempre se encuentre entre 0 y 1. Cuando la acción sea superior a 1, se fuerza a que no supere a dicho valor y en caso de que la acción sea negativa, que no disminuya del valor 0. Las operaciones lógicas para estas condiciones se realizan mediante comparadores y funciones de selección, como la mostrada en la Figura 43.

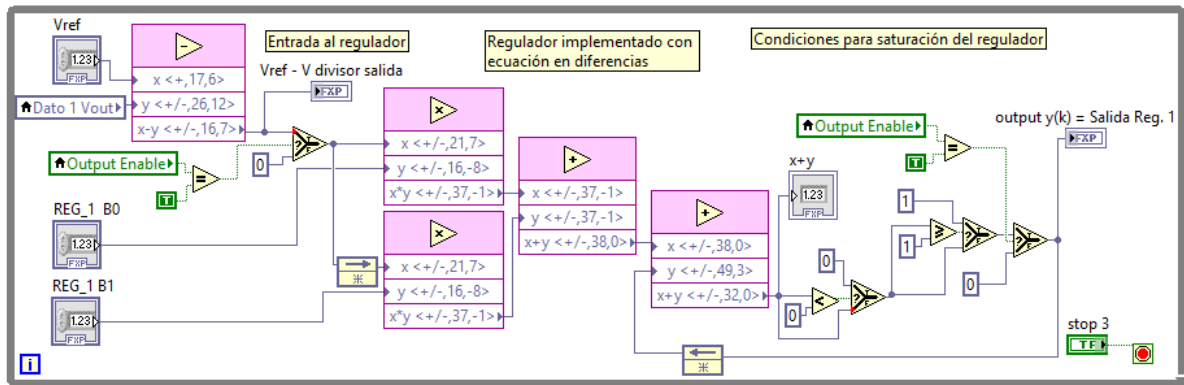


Figura 44: Control de Lazo de tensión: Regulador y control de la actuación

## Control de Lazo de corriente

El lazo interno de corriente es el encargado de que la corriente de entrada al convertidor *boost* esté en fase con la tensión de entrada, de manera que también siga una forma sinusoidal proporcional a dicha tensión  $V_{in}$  y se minimice el contenido armónico de la corriente.

Para realizar el control, el regulador recibe la referencia de la diferencia de corriente medida a la entrada del convertidor y la actuación del lazo externo multiplicado por el voltaje de entrada  $V_{in}$ . El control del lazo de corriente debe implementarse a la frecuencia de conmutación de 100 kHz para poder generar la señal PWM a dicha frecuencia, por tanto, el muestreo de los datos también debe realizarse a dicha frecuencia.

El bucle de lectura del dato  $V_{in}$  es similar al de la lectura de  $V_{out}$  a excepción de la frecuencia de ejecución, que debe realizarse a 100 kHz y para obtener el dato en el punto más estable, se decide muestrear en el punto medio del periodo  $T_{on}$  del pulso PWM. Para ello, en la primera parte de la secuencia de la Figura 45 se realiza un desplazamiento lógico a la derecha (equivalente a una división por la mitad del valor) con el número de *ticks* del periodo  $T_{on}$  correspondiente con la variable *High Period (Ticks)*.

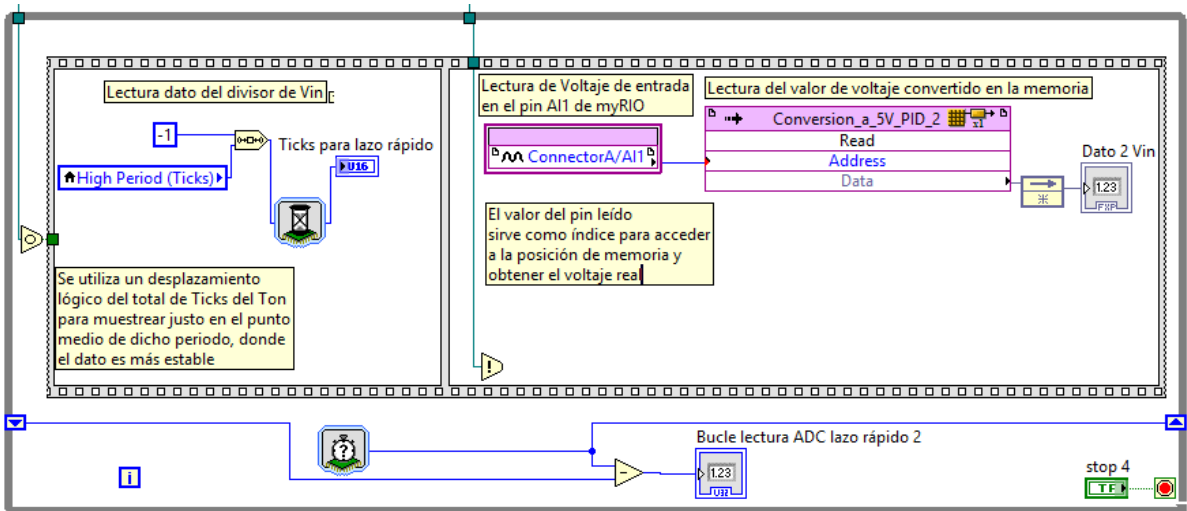


Figura 45: Control de Lazo de corriente: Lectura *Vin*

La segunda parte de la secuencia es similar a la del lazo de tensión, donde la ocurrencia generada se corresponde a la activación de dos bucles:

- El bucle de lectura de la corriente de entrada *Iin* y su calibración (Figura 46), puesto que la lectura se corresponde al valor de la *Vshunt* y debe obtenerse su valor en corriente. Se ha tenido en cuenta que este bucle no debe llevar tiempos de espera mediante la función *Wait*, dado que ya se esperó el tiempo necesario en el bucle que generó su ocurrencia y ambos se ejecutarán de forma paralela.

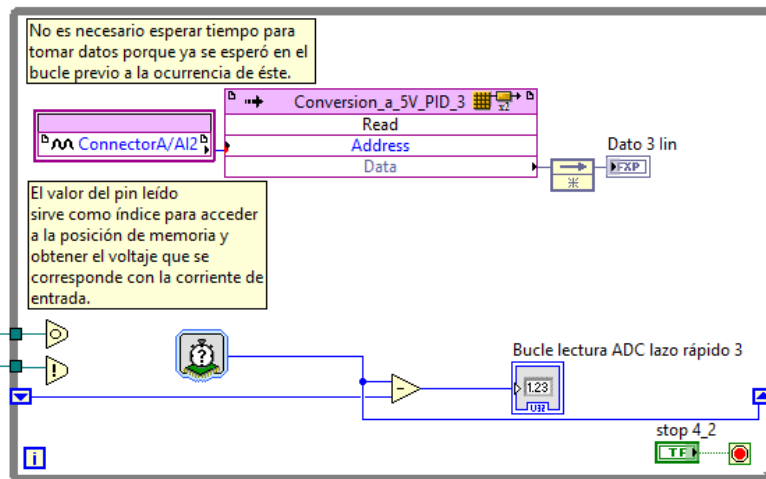


Figura 46: Control de Lazo de corriente: Lectura *Iin* y calibración

- El bucle de activación del cálculo para la obtención de la intensidad que debe ser comparada con la de entrada, *Iin* (ver Figura 47). Este bucle multiplica la actuación del regulador de tensión por el voltaje de entrada al convertidor *boost*, *Vin*. A su vez, este bucle activa la ocurrencia del bucle del regulador del lazo de corriente (Figura

48), puesto que ya se han realizado todos los cálculos necesarios para los datos de entrada de dicho regulador.

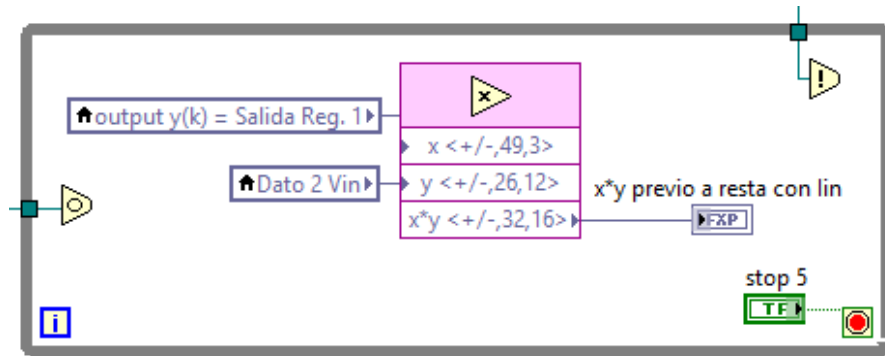


Figura 47: Control de Lazo de corriente: Obtención de corriente para referencia con  $I_{in}$

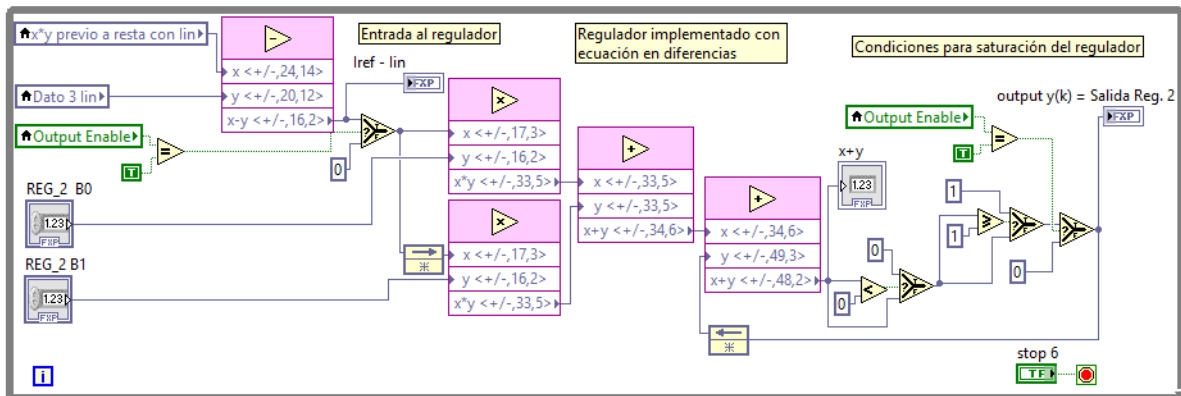


Figura 48: Control de Lazo de corriente: Regulador y control de la actuación

### 5.2.2. Host VI

El Host VI contiene la interfaz que el usuario maneja. Permite la interacción con algunos de los controles de la FPGA y muestra gracias al módulo de RT (*Real Time*) de LabVIEW los resultados extraídos del diagrama de bloques del FPGA VI.

Los controles que son configurables por el usuario son:

- $V_{ref}$ : el Voltaje de referencia deseado para el convertidor *boost* a través de *Set Vref*.
- *Output Enable*: que activa la generación del PWM y los cálculos en los reguladores y, por tanto, todo el control del convertidor.
- Coeficientes de los reguladores: se permite modificarlos para posibles cambios y pruebas en los reguladores implementados. En su configuración, se estableció suficiente resolución y rango de valores del tipo de dato FXP para que no hubiese problemas en caso de que se quieran modificar.

En la siguiente Figura 49 se muestra un pequeño ejemplo de las variables que son configuradas en modo escritura o lectura. Para ello, una vez creado el FPGA VI y realizada su compilación y generación exitosa del *bitfile*, en el Host VI se especifica el dispositivo y la referencia a dicho *bitfile* mediante la función *Open FPGA VI Reference* para que la FPGA sea programada.

En el bucle *while* continuamente se está ejecutando la escritura de las constantes para los coeficientes de los reguladores, que en caso de que sean modificados, los cambios se aplicarán en tiempo real a la FPGA y los reguladores harán los cálculos con esos nuevos valores. La lectura de diferentes variables, entre ellas las obtenidas en los pines analógicos de entrada de myRIO también se está ejecutando continuamente y para una mejor visualización, se genera una gráfica con un conjunto representativo de las muestras extraídas para observar sus valores. Una vez que se decide parar la ejecución, con la función *Close FPGA VI Reference* se cierra la referencia a la FPGA y por defecto, se resetea.

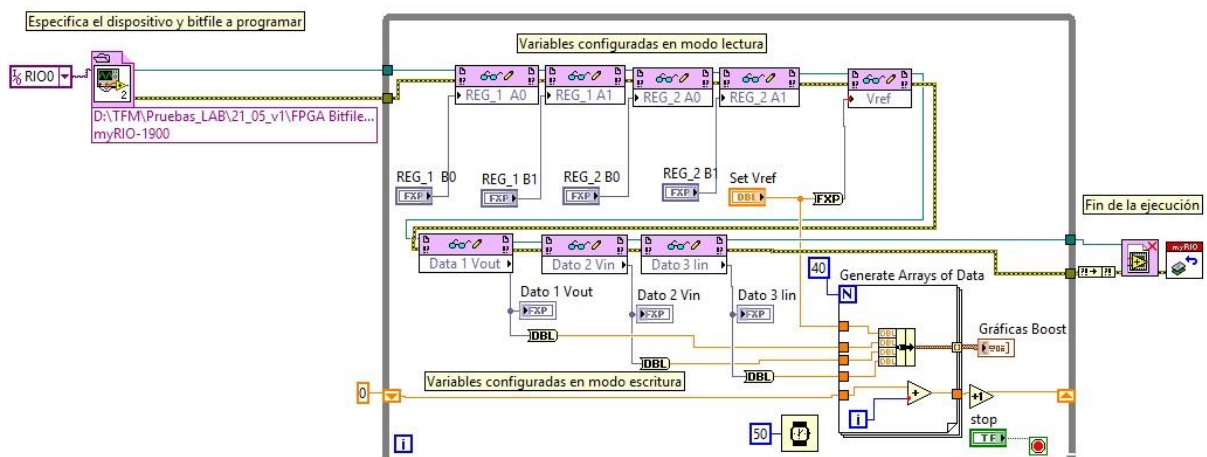
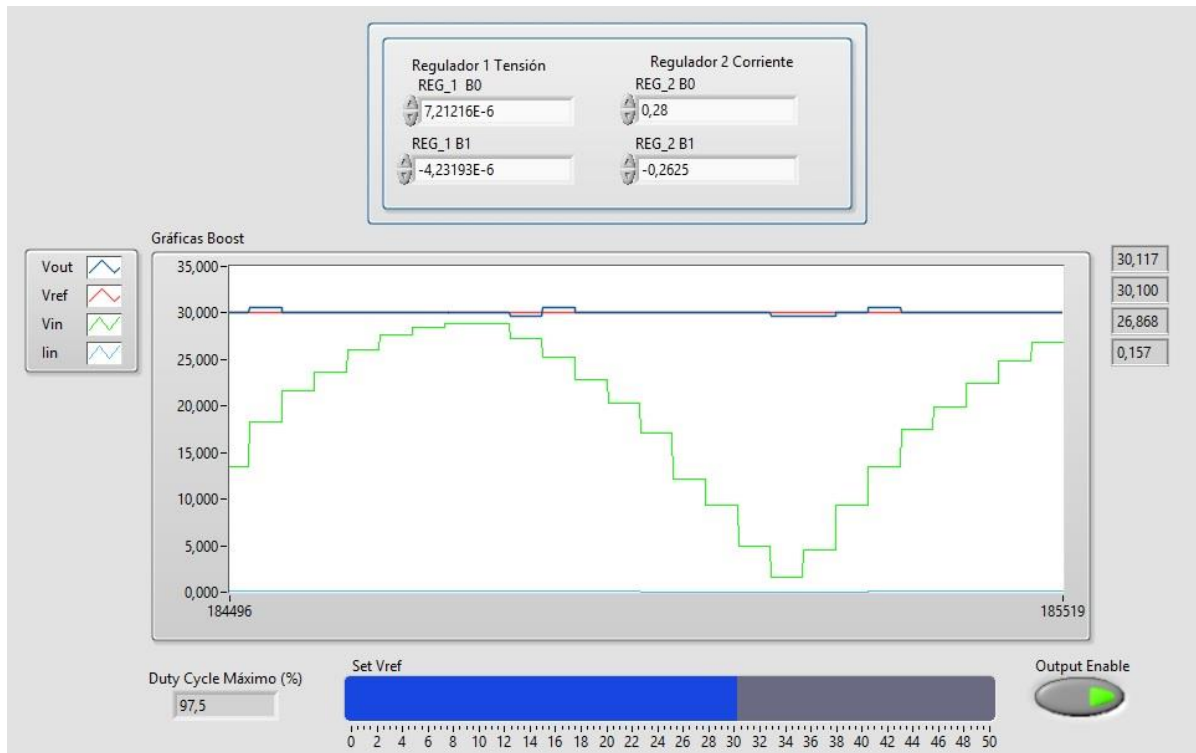


Figura 49: Ejemplo de diagrama de bloques de Host VI

En la siguiente Figura 50 se muestra el ejemplo final de Host VI durante la ejecución y control del factor de potencia. Inicialmente, se deben configurar los valores de los reguladores y elegir el voltaje de referencia para el convertidor *boost* y para iniciar la regulación. Una vez establecidos dichos valores, se debe activar el botón de *Output Enable*. En la gráfica de la Figura 50 se muestra el conjunto de datos extraídos de las lecturas de los pines, donde se observa el voltaje de entrada *Vin* rectificado y cómo el voltaje de salida *Vout* permanece estable alcanzando al voltaje configurado por el usuario con referencia *Vref*. La corriente de entrada también se muestra, pero debido a su pequeño valor comparado con la escala de

valores de la gráfica, apenas es visible. Sin embargo, se han configurado todas estas variables como indicadores, de forma que se puede observar su valor en el margen derecho de la gráfica.



**Figura 50: Interfaz final de usuario en Host VI**



## 6. RESULTADOS

A lo largo de todo el desarrollo de este TFM y una vez seleccionado el dispositivo de NI myRIO como el óptimo para la implementación del control digital, se han realizado numerosas pruebas para comprender su funcionamiento de ejecución, comunicación e interacción con el usuario como se ha descrito a lo largo del apartado 5. Posteriormente, se han realizado pruebas con un pequeño circuito a modo de prueba para un control de tensión en DC, y finalmente con el convertidor *boost*, primero realizando un control en tensión y por último la corrección del factor de potencia.

### 6.1. Integración de los dispositivos

Para la realización de las pruebas finales, han sido necesarios los siguientes dispositivos:

- Convertidor *boost* [12].
- Dispositivo NI myRIO-1900.
- Osciloscopio MSOX3014A.
- Sondas de tensión y de corriente (N2790A).
- Fuente de alimentación de corriente continua a 15 V y 5 V para los componentes del convertidor (Aim-TTi EX354TV).
- Fuente de alimentación de corriente alterna (Adaptative Power Systems FC200).
- Tablero de resistencia de carga variable.
- Multímetro.

El convertidor *boost* utilizado para las pruebas dispone de *jumpers* para seleccionar qué señales se quieren medir para el control, si los voltajes y corriente previas a la etapa de aislamiento o las señales posteriores al aislamiento y tras los divisores resistivos que limitan su salida a 1 V. Para todas las pruebas realizadas se ha escogido la segunda configuración, para evitar voltajes superiores a los que admiten los pines del dispositivo NI myRIO (5 V) en detrimento de la resolución obtenida para las mediciones de las entradas analógicas.

En la figura Figura 51 se muestra un detalle de a qué pines se ha conectado una placa externa de myRIO al puerto de expansión MXP A para un mejor acceso a los pines con el convertidor *boost*.

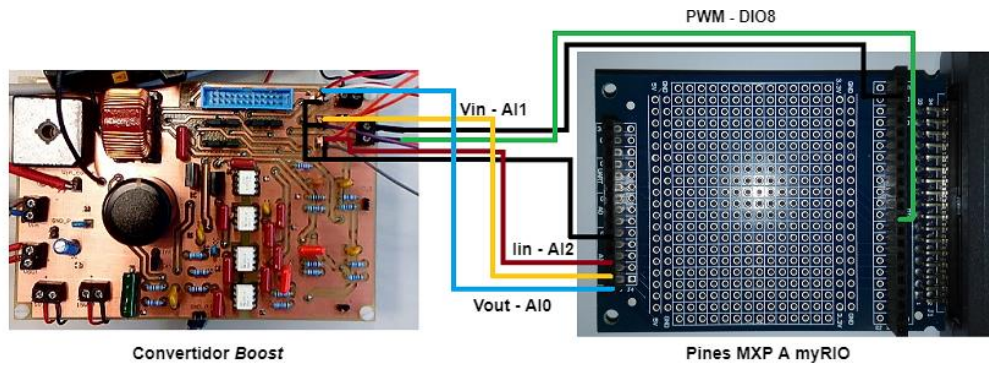


Figura 51: Conexión de pines en MXP A myRIO con convertidor *boost*

En la Figura 52 se muestra el diagrama de interconexión completo y, por último, el montaje del sistema completo con el que se han hecho las pruebas en laboratorio en la Figura 53.

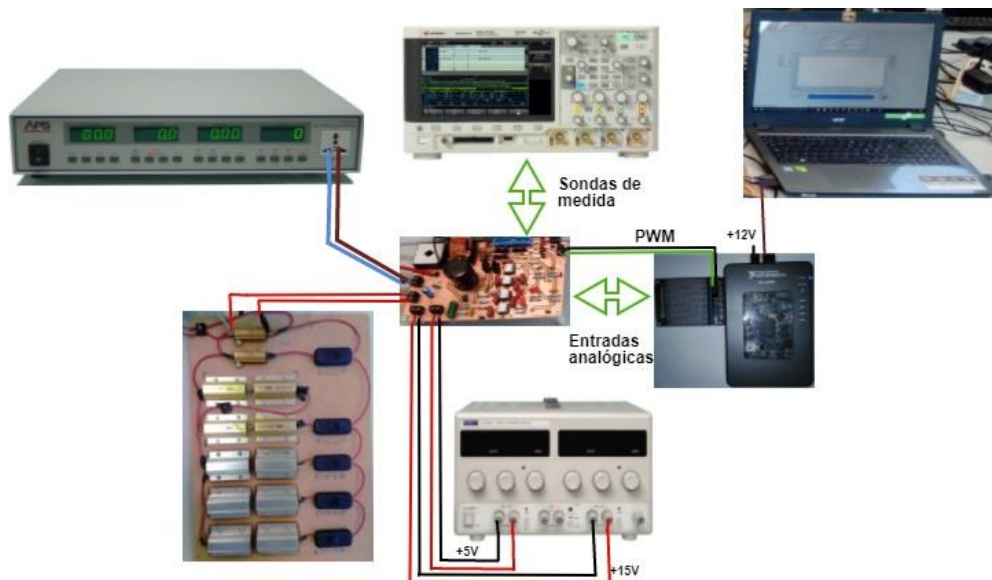


Figura 52: Esquema de interconexión final

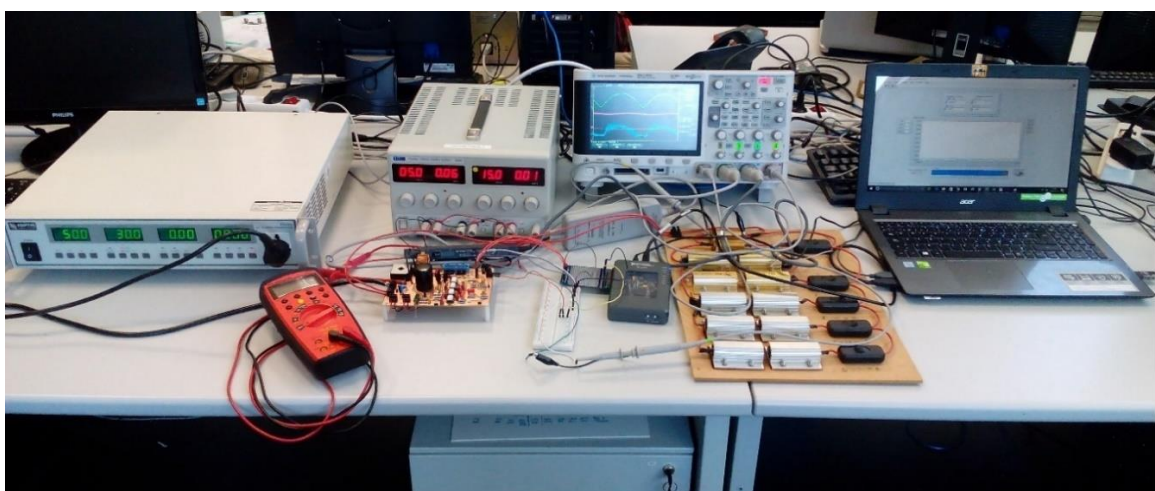


Figura 53: Montaje final para las pruebas realizadas en el laboratorio

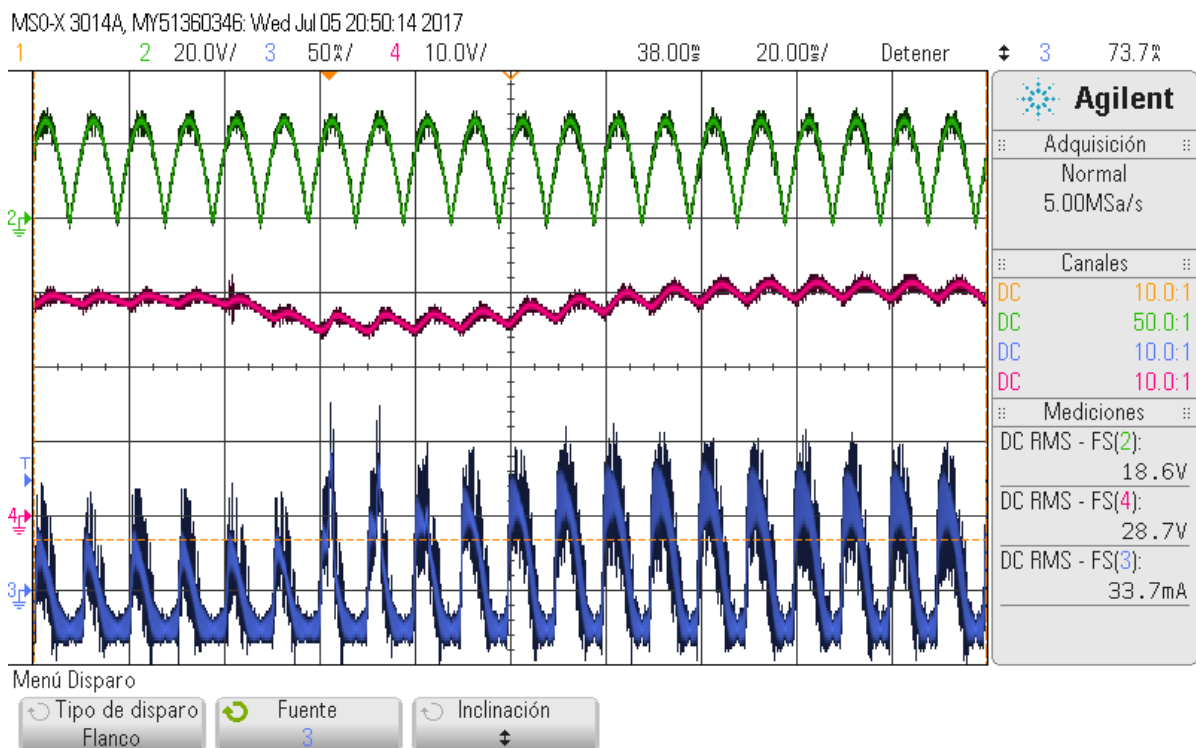
## 6.2. Resultados experimentales

Tras realizar diversas pruebas intermedias de control de tensión en lazo abierto y cerrado en DC, visualización de generación de PWM, sincronización entre bucles de la interfaz y datos leídos del convertidor *boost*, se procedió a la prueba de la interfaz final desarrollada en LabVIEW y programada en el dispositivo NI myRIO para la corrección del f.d.p.

Inicialmente, se configura la fuente de corriente alterna a una frecuencia de 50 Hz y tensiones bajas para comprobar el funcionamiento. Se realizaron diversas pruebas para observar el comportamiento de los transitorios y diversos escalones de carga.

### Transitorios

Con un voltaje de entrada de  $V_{in} = 20\text{ V}$  y el Host VI diseñado en LabVIEW activado, se realizan capturas para observar la respuesta a escalones de carga. En el primer caso (ver Figura 54), se realiza una transición de  $2\text{k}\Omega$  a  $1\text{k}\Omega$ .



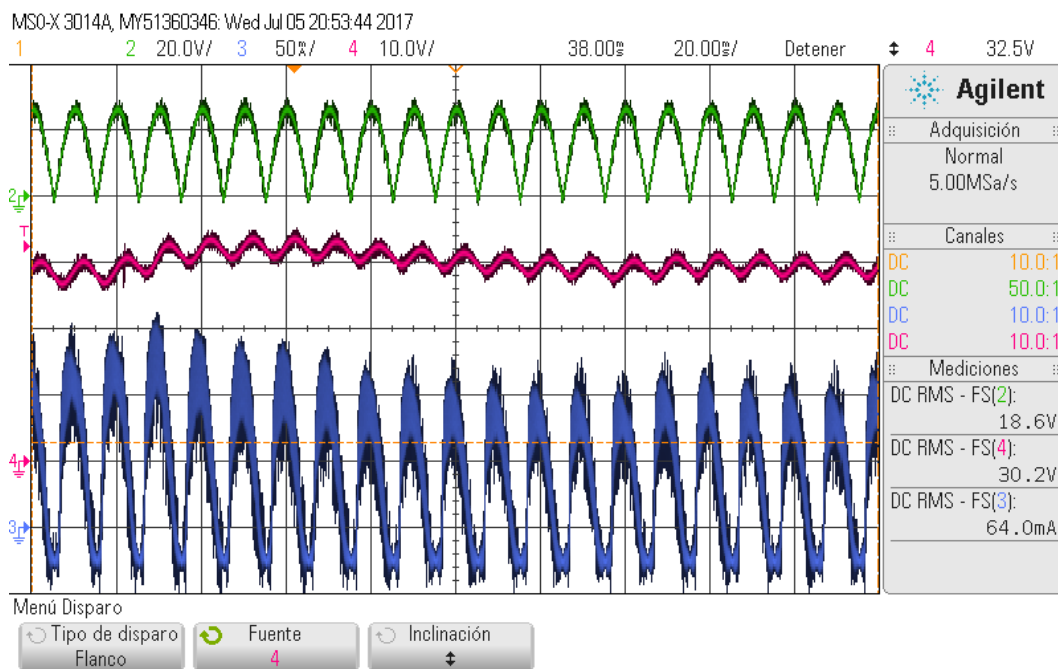
**Figura 54: Escalón de carga negativo de  $2\text{k}\Omega$  a  $1\text{k}\Omega$  con  $V_{in} = 20\text{ V}$  y  $V_{out} = 30\text{V}$**

Se observa que con mayor carga la corriente solicitada a la entrada es menor y que a menor carga, la corriente solicitada es mayor. Esto se debe a la acción reguladora del lazo de corriente, donde al variar la carga, el lazo de tensión debe mantener el balance de potencias,

generando un comando de potencia que solicitará una referencia de corriente mayor en el caso de que la carga haya disminuido.

En la siguiente Figura 55, se ha variado la carga realizando un escalón de carga positivo de  $533,33\Omega$  a  $688\Omega$ , donde se puede apreciar que el incremento de carga supone una bajada en la corriente solicitada al lazo interno.

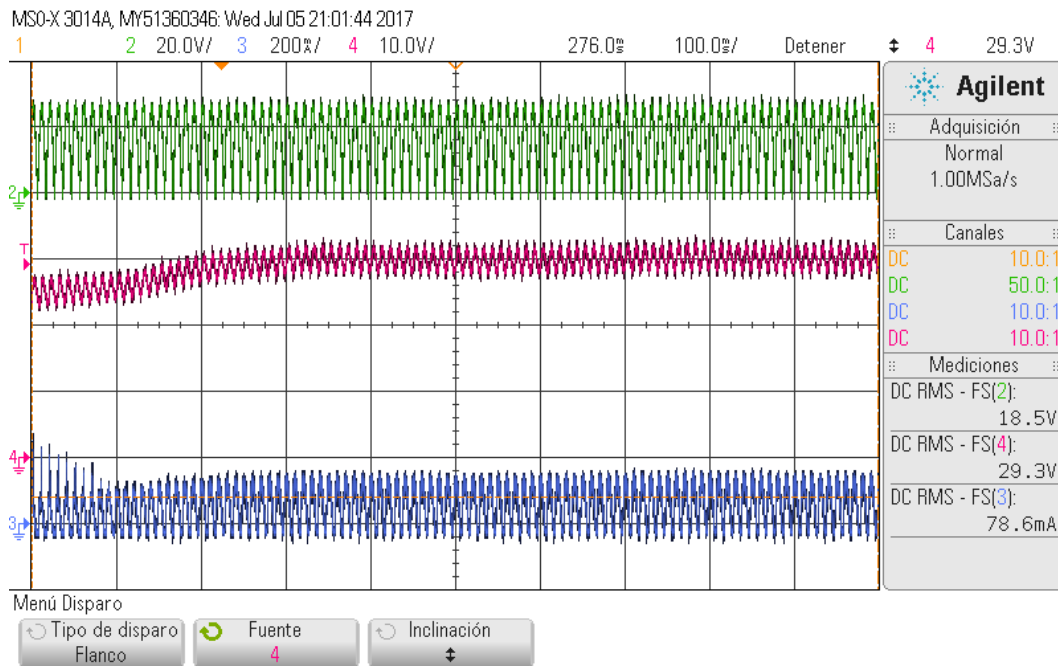
Además, se comprueba en términos generales que, al disminuir la carga (de las pruebas de  $2k\Omega$  a  $1k\Omega$  a las capturas de  $533,33\Omega$  a  $688\Omega$ ), la corriente aumenta, donde en el primer caso los valores oscilan entre 30 y 40 mA de valores de intensidad de pico mientras que los valores del segundo caso, con menor carga, oscilan entre 100 y 75 mA.



**Figura 55: Escalón de carga positivo de  $533,33\Omega$  a  $688\Omega$  con  $V_{in} = 20\text{ V}$  y  $V_{out} = 30\text{ V}$**

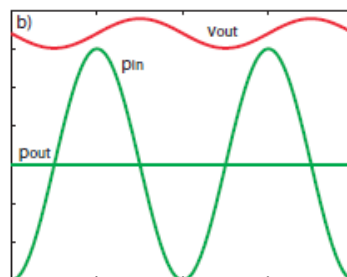
A continuación, se muestran algunas capturas de transitorios con diferentes voltajes de referencia para la salida del convertidor *boost*. La carga resistiva aquí se fija a  $1k\Omega$ .

La Figura 56 muestra una subida de tensión de 25 V a 33 V con un voltaje de entrada de 20V.



**Figura 56: Transitorio de subida de voltaje de  $V_{ref} = 25\text{ V}$  a  $V_{ref} = 33\text{ V}$  con  $V_{in} = 20\text{ V}$**

Se observa cómo la tensión de salida se regula al nuevo valor de referencia alcanzando un tiempo de establecimiento cercano a los 350 ms. Este valor es similar al tiempo de establecimiento que se simuló de la planta del lazo de tensión junto con su regulador en el Anexo III.2 Implementación de regulador para lazo de tensión. Se puede comprobar que, ante un cambio del voltaje de referencia, es el lazo externo de tensión el que se encarga de corregirlo, de ahí que el tiempo de respuesta sea más lento. Que la respuesta sea más lenta es una característica inherente a la corrección del f.d.p, puesto que, si se realizase un lazo de tensión rápido, se corregiría el rizado característico del voltaje de salida (debido a la variación de la potencia de entrada, que varía al doble de frecuencia de la red eléctrica como muestra la Figura 57 obtenida de [13]) y estaría continuamente variando la referencia para el lazo de corriente, introduciendo armónicos en la red.



**Figura 57: Potencia de entrada y potencia y tensión de salida característicos de una corrección de f.d.p**

La siguiente Figura 58 muestra un transitorio de bajada de tensión de 33 V a 25 V. Se aprecia cierto ruido en la tensión de salida con el cambio del voltaje de referencia.

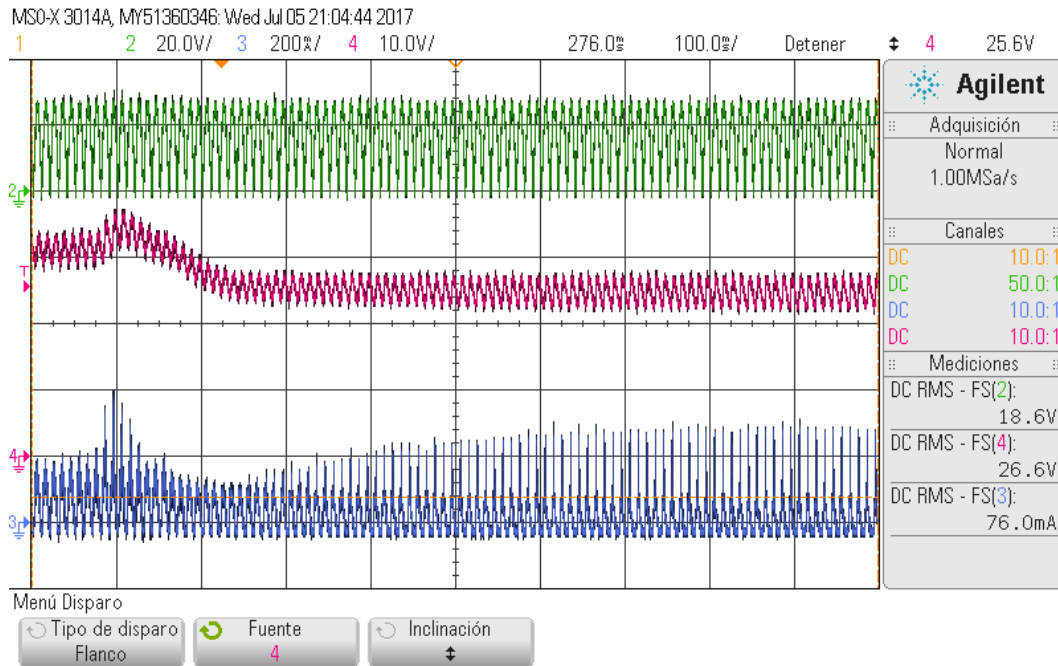


Figura 58: Transitorio de bajada de voltaje de  $V_{ref} = 33\text{ V}$  a  $V_{ref} = 25\text{ V}$  con  $V_{in} = 20\text{ V}$

### Régimen permanente

También se tomaron capturas en régimen permanente a algunos de los valores máximos que se permitió alcanzar la corrección del f.d.p. con una carga de  $688\Omega$ . La primera Figura 60 muestra un régimen permanente con una  $V_{in} = 26\text{ V}$  y un voltaje de referencia de  $V_{ref} = 38\text{ V}$ . Se observa cómo la tensión y corriente son proporcionales y se encuentran en fase debido a la acción del lazo rápido de corriente. El lazo de tensión se encarga de mantener el balance entre las potencias de entrada y salida al convertidor y así poder regular la tensión de entrada al valor de referencia establecido. Se aprecia también el rizado característico del voltaje de salida por la corrección del f.d.p. como el mostrado en la Figura 57.



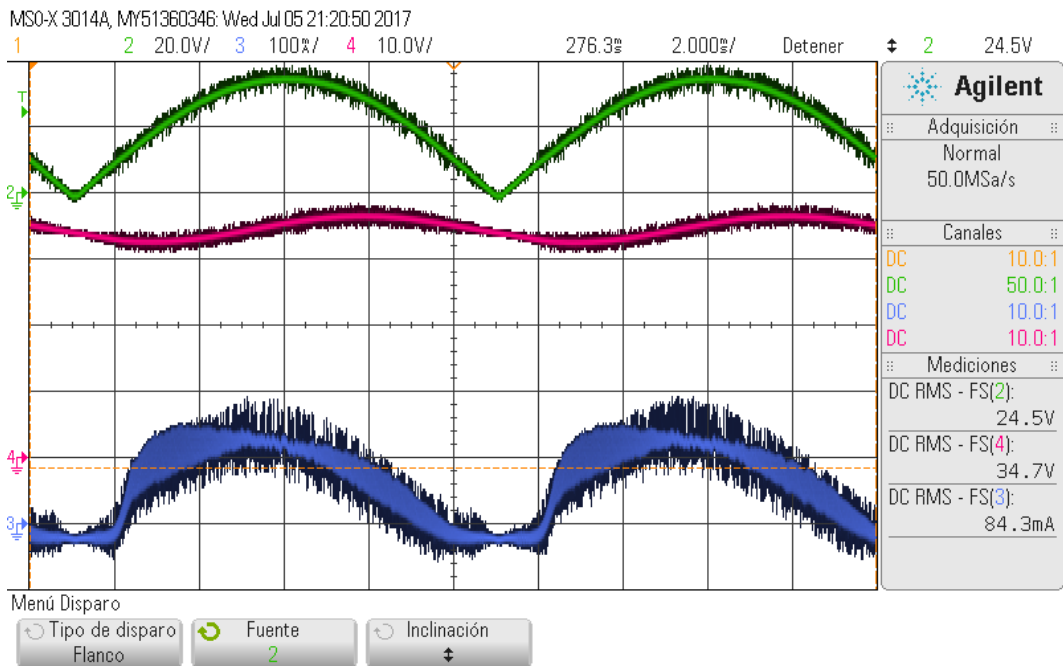


Figura 59: Régimen permanente con  $V_{in} = 26\text{ V}$  y  $V_{ref} = 38\text{ V}$

En la Figura 60 se muestra una captura con un  $V_{ref}$  de  $45\text{ V}$  con el que se consiguió un FP de  $0,98$  (dato tomado de la fuente de alimentación).

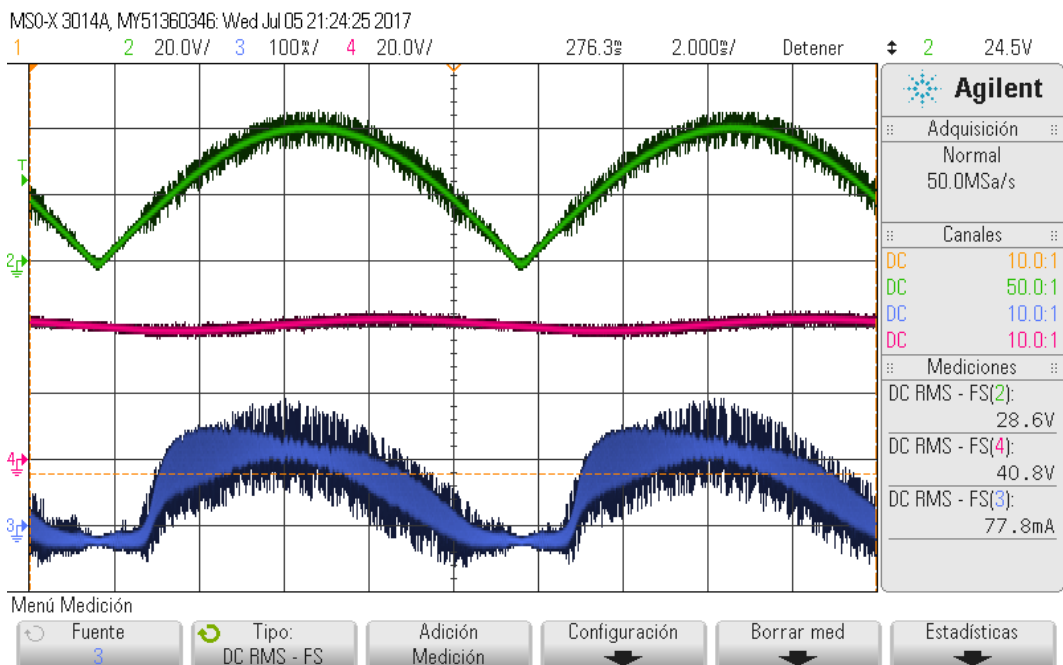
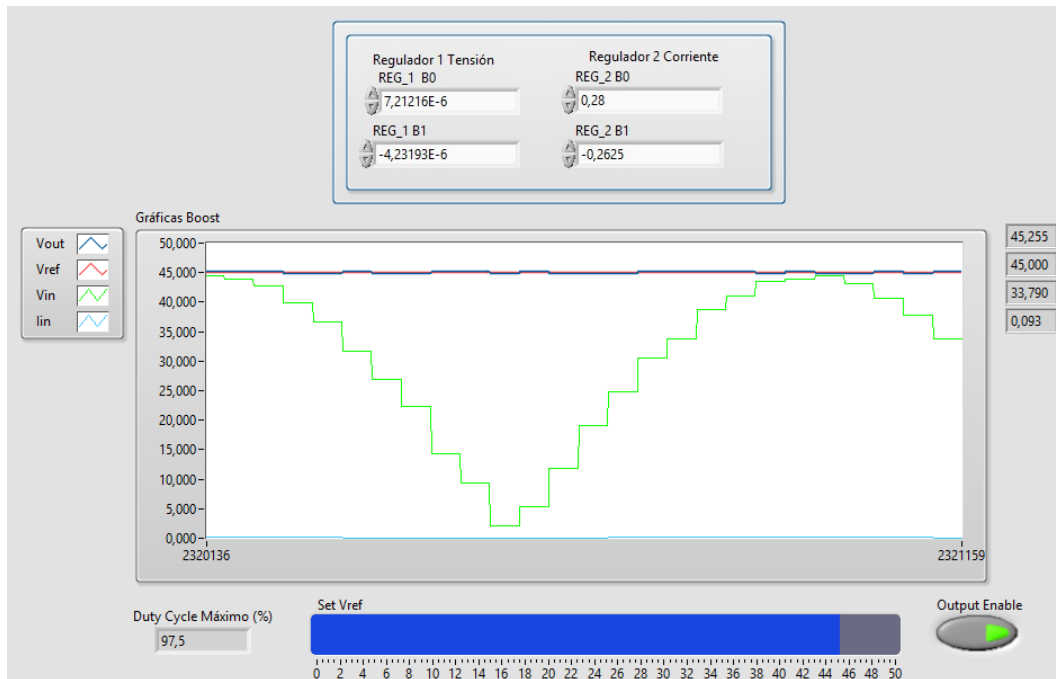


Figura 60: Régimen permanente con  $V_{in} = 30\text{ V}$  y  $V_{ref} = 45\text{ V}$

Se observa en ambas capturas cierta distorsión en la corriente de entrada que se comprobó que generaba fallos en el control del f.d.p., volviéndose el sistema inestable cuando se

incrementaba más el valor de referencia para la tensión de salida. Por último, se muestra en la Figura 61 la interfaz de LabVIEW para el control digital del convertidor, con los valores programados de los reguladores, el voltaje de referencia seleccionado a 45 V y las gráficas de salida con los datos de voltajes de entrada, salida, referencia y corriente de entrada.



**Figura 61: Interfaz de usuario para el control digital del convertidor conmutado en LabVIEW**

Se hicieron diversos análisis y comprobaciones de las causas de los fallos en la corrección del f.d.p y se concluyeron varias opciones que afectaban al control digital:

- En el convertidor *boost* utilizado se comprobó que a valores elevados de *duty cycle* y de tensiones de entrada generaban mayores pérdidas y, por tanto, menor rendimiento. En la Tabla 2 se puede observar que a mayores valores de *duty cycle*, más pérdidas se obtienen. Esto afecta a que es necesario un valor mayor de *duty cycle* para conseguir la misma tensión de salida, lo que conlleva a mayores pérdidas, incremento de temperatura en los componentes y más ruido. Además, las pérdidas suponen que el comportamiento de las tensiones de entrada y salida se degraden y no sean de forma lineal a valores altos de *duty cycle* y voltaje de entrada, lo que lleva a desajustes en las calibraciones para las medidas y puede ser fuente de errores para el control digital.
- Fueron necesarios cambios en la etapa de potencia del convertidor *boost*. El driver del optoacoplador presentaba fallos de conmutación, para lo cual fue necesario añadir una resistencia entre el driver y la entrada del MOSFET que actúa de interruptor. Se comprobó que, a mayores corrientes, el fallo de conmutación ocurría con valores de



tensión más bajas. La elección de la resistencia se realizó de manera experimental mediante un compromiso entre el valor de la temperatura en la resistencia y la tensión que podía alcanzar. Se comprobó que a valores elevados de un *duty cycle* del 80%, dicha resistencia se empezaba a desoldar por elevada temperatura. El hecho de que la elección de la resistencia fuese realizada de manera experimental para CC y no se comprobase su comportamiento para CA, también podían suponer fallos en la conmutación para CA, haciendo que el sistema se volviese inestable. Además, el driver del optoacoplador también presentaba *ringing* con valores elevados de su alimentación.

- Hay grandes pérdidas en resolución debido a los divisores resistivos del convertidor *boost* y el rango de valores leídos por el ADC en NI myRIO. Los divisores resistivos fueron diseñados para tensiones de 0 a 1 V, mientras que el rango de medida que permite NI myRIO es de 0 a 5 V. Además, el rango de valores medidos con myRIO se vuelven a dividir por 5, puesto que el ADC que muestrea los datos funciona entre 0 y 1 V. Esto supone una importante pérdida de la resolución de los valores obtenidos, puesto que de los 4095 valores disponibles para la representación de los valores entre 0 y 5 V, solo se están aprovechando 819 valores correspondientes a el rango de valores de 0 a 1 V. Con todo ello, las diferencias de medición entre los valores tomados de NI myRIO pasan de ser de 1,221 mV a 6,105 mV mientras que en todo el rango de medida del convertidor *boost* se pasa de tener una resolución de  $\frac{600}{4095} = 0.14652 V$  a  $\frac{600}{819} = 0,7326 V$ . Esta pérdida de resolución puede afectar en gran medida a las calibraciones realizadas en el control digital de LabVIEW, puesto que se están perdiendo precisión en la lectura de los datos y puede dar lugar a valores erróneos en los reguladores para el control digital.
- Otro factor importante es el ruido en general de todo el convertidor *boost*. Al estar continuamente oscilando la tensión y corriente de entrada, se genera mucho ruido en la conversión de CA/CC. El convertidor fue diseñado para corrección de f.d.p. pero no llegaron a hacerse pruebas con él en CA por lo que, si ya presentaba inestabilidades con CC, en CA se verán más incrementadas todavía.

## 7. CONCLUSIONES

La continua evolución en los sistemas de control para la electrónica de potencia motiva que cada vez se investigue más el control digital. En concreto, el control digital de convertidores conmutados ha tomado gran importancia y es motivo de investigación por las numerosas ventajas que supone frente a un control analógico.

Para la realización del control digital se pueden utilizar dispositivos que sean reprogramables por el usuario para así aumentar las opciones de modificación, implementación de algoritmos y mejor integración con otros sistemas. La programación hardware de los dispositivos que realicen el control digital se suele realizar mediante código HDL descrito por el usuario, sin embargo, hay opciones software que permiten una programación más intuitiva mediante la programación gráfica del dispositivo. Un ejemplo de ello es LabVIEW, que a través de diagramas de bloques permite implementar las mismas funcionalidades que las descritas por un código de alto nivel.

En el presente TFM, se han investigado las prestaciones que ofrecían dos dispositivos orientados a entornos educativos de la empresa NI para realizar un control digital de convertidores conmutados mediante el software de LabVIEW. Tras el análisis de las características que se necesitaban para un convertidor conmutado concreto, se descartó uno de los dispositivos de adquisición de datos por no disponer de requisitos críticos que necesitaba el control digital y se escogió el dispositivo con lógica reconfigurable NI myRIO que integra un SoC con una FPGA.

El control digital a realizar sobre el convertidor conmutado escogido se implementó con un doble lazo en tensión y corriente para realizar corrección del f.d.p., lo que supone un muestreo de 3 señales analógicas y con una frecuencia de conmutación medias-elevadas (100 kHz). Estas características supusieron diversas limitaciones para el diseño de la interfaz con myRIO mediante LabVIEW tales como:

- La función estándar de PWM para myRIO que ofrece una fácil integración en el diseño de la interfaz no pudo ser utilizada por su limitación en frecuencia (40 kHz). Esto obligó a implementar la funcionalidad del PWM a nivel de programación en la FPGA con diversos bucles, lo que requirió mayores esfuerzos que si el diseño se realizase en código HDL.

- Las funciones que dispone LabVIEW para implementar los reguladores discretos tampoco pudieron ser utilizados por el elevado uso de recursos de la FPGA, llegando a impedir su programación por sobreuso de recursos. De nuevo, fue necesario implementar ambos reguladores a nivel de FPGA y adoptar técnicas de optimización para lograr la compilación en la FPGA.
- La sincronización entre las adquisiciones de valores analógicos, cálculos en los reguladores y generación del PWM no fue de forma trivial. Hubo que hacer diversas pruebas y comprobar con el osciloscopio el correcto funcionamiento de la generación del PWM, su *duty cycle* y ocurrencia de los muestreos.

Todo esto ha supuesto que los tiempos de desarrollo no hayan obtenido una mejoría respecto a la de una implementación mediante código HDL. Aun así, se destaca que la interfaz desarrollada permite un control digital en el que el usuario puede realizar modificaciones en tiempo real, en concreto, permitiendo activar la generación del PWM y variar los valores de voltaje de referencia y reguladores y observar su respuesta.

Por otra parte, el dispositivo myRIO utilizado presentaba algunas limitaciones ya no solo por los elevados recursos que utiliza la FPGA para un sistema de control, sino también por los bits que dispone el ADC para la obtención de los valores analógicos. Esto, junto con las adaptaciones de voltaje mediante los divisores resistivos del convertidor conmutado suponían una importante pérdida de resolución que podía generar inestabilidades para la corrección de f.d.p. Además, el convertidor conmutado, aunque fue diseñado para corrección de f.d.p, nunca fue probado con tal fin, sino que se realizaron pruebas en CC en las que se observaron degradaciones a altos valores de *duty cycle*, calentamiento de componentes y ruido, que en CA se ha visto incrementado.

En los resultados experimentales se ha comprobado que a bajas tensiones sí funciona la corrección del f.d.p, pero se observaron inestabilidades que imposibilitaron las pruebas a tensiones mayores.

Por lo tanto, no se recomienda el uso de LabVIEW junto con el dispositivo NI myRIO para la realización de un control digital de un convertidor conmutado tipo *boost* para corrección del f.d.p. con frecuencias de conmutación medias-elevadas. Las limitaciones del dispositivo expuestas junto con el convertidor seleccionado hacen que la programación gráfica no presente ventajas frente a una programación de FPGA en código HDL.

## 7.1 Líneas futuras

Como líneas de desarrollo futuro y con las conclusiones extraídas, se pueden obtener ventajas del uso de LabVIEW para controles digitales de convertidores conmutados de menor frecuencia o con dispositivos con mejores prestaciones y más óptimos que NI myRIO. Por ejemplo, en la serie de NI CompactRIO Single-Board existen dispositivos como el sbRIO-9627 con un SoC FPGA de Xilinx Zynq-7020 con mayor número de LUTs y ADCs con 16 bits de resolución [19]. En estos casos sí se observarían ventajas de la programación gráfica y tiempos de desarrollo en LabVIEW frente a la programación hardware mediante código descrito en HDL.

Asimismo, sería conveniente depurar el sistema desarrollado en el convertidor utilizado para PFC de tensiones mayores, como 110 V o 220 V.

También se podría explorar el uso de LabVIEW junto con myRIO para la implementación de técnicas de ciclos de trabajo pre-calculados para corrección de f.d.p [13][18] dado que esta técnica necesita un menor coste computacional o técnicas HIL (*Hardware In the Loop*) y contrastar los resultados obtenidos con en el dispositivo.

# BIBLIOGRAFÍA

- [1] Alejandro García Talón, “Control Digital de Fuentes de Alimentación”. Universidad Autónoma de Madrid, 2009.
- [2] David Escalona Moreno, “Control de convertidor conmutado mediante FPGA”. Trabajo Fin de Grado, Universidad Autónoma de Madrid, 2016.
- [3] Patricia Cuesta Marcos, “Control digital de convertidor conmutado mediante FPGA mixta analógica/digital”. Proyecto Fin de Carrera, Universidad Autónoma de Madrid, 2015.
- [4] Santiago Montes Fernández, “Convertidor conmutado para corrección del factor de potencia mediante FPGA”, Proyecto Fin de Carrera, Universidad Autónoma de Madrid, 2015.
- [5] M. Demirtaş and G. Gezer, "Analyzing of PI controlled DC/DC converter output voltage using Labview," IEEE 18th Signal Processing and Communications Applications Conference, Diyarbakir, 2010, pp. 704-707.
- [6] Udaya Kumar, B. and Ramesh Patnaik, M., “Time Response Analysis of a DC Motor Speed Control with PI and Fuzzy Logic Using LABView Compact RIO”, International Research Journal of Engineering and Technology (IRJET), Volume 2, Issue 8, 2015.
- [7] Vijayalakshmi, S. and Sree Renga Raja, T., “Time Domain Based Digital Controller for Buck-Boost Converter,” *Automatika – Journal for Control, Measurement, Electronics, Computing and Communications, J Electr Eng Technol* Vol. 9, No. 5: 1551-1561, 2014.
- [8] National Instruments. <http://www.ni.com/es-es.html>.
- [9] Shenbagalakshmi, R. and Sree Renga Raja, T., “Discrete prediction controller for DC-DC converter,” *Acta Scientiarum. Technology, Maringá*, v. 36, n. 1, p. 41-48, Jan.-Mar., 2014.
- [10] Bingöl, O., Mahmut Pişirir, O., Efe, R., Duru, I., Özkaya, B., “DC-DC converter experimental tool,” 4th International Symposium on Innovative Technologies in Engineering and Science, 2016.
- [11] A. Izadian, G. Edelman and S. Johnson, "Gate driver of DC-DC boost converters using national instruments LabVIEW and NImyDAQ," IEEE International Conference on Electro/Information Technology, Milwaukee, WI, 2014.

- [12] Ester Vicario Arroyo. "Diseño y construcción de convertidor conmutado de potencia, su control y comparativa con emulación HIL". Trabajo Fin de Grado, Universidad Autónoma de Madrid, 2016.
- [13] Alberto Sánchez González, "Aportaciones mediante implementación basada en sistemas embebidos al control digital de convertidores conmutados". Tesis doctoral, Universidad Autónoma de Madrid, 2013.
- [14] <http://digital.ni.com/public.nsf/allkb/8F35B8099427B48686257A8B003A72D8>
- [15] «SpControl Technologies,». <http://www.spcontroltechnologies.com/es/>.
- [16] <http://digital.ni.com/public.nsf/allkb/311C18E2D635FA338625714700664816>
- [17] Katsuhiko Ogata. Ingeniería de control moderna. PEARSON, 2010.
- [18] A. Sanchez, A. de Castro, V. M. Lopez, F. J. Azcondo & J. Garrido, "Single ADC Digital PFC Controller using Pre-calculated Duty Cycles", in IEEE Transactions on Power Electronics, vol. 29, no. 2, pp. 996-1005, feb 2014. Factor de Impacto: 6,008 (3/249)
- [19] NI sbRIO-9627 Specifications. <http://www.ni.com/pdf/manuals/375278a.pdf>
- [20] R. W. Erickson and D. Maksimovic, Fundamentals of Power Electronics. Kluwer Academic Publishers, 2001.

# ANEXO I

## I.1 Características de convertidor boost

El convertidor conmutado que se ha utilizado para las pruebas de este TFM se ha reutilizado de [12]. Las principales características de diseño con las que se implementó se recogen en la siguiente Tabla 6:

<b>Parámetro</b>	<b>Valor</b>
Frecuencia de conmutación	100 kHz
Frecuencia rectificadora	100 Hz
V <sub>in</sub>	230 Veficaz (red eléctrica)
V <sub>pico</sub>	325,27 V
V <sub>out</sub>	400 V en DC
P <sub>out</sub>	300 W
I <sub>in</sub>	1,304 A

**Tabla 6: Parámetros de diseño del convertidor *boost* escogido**

# ANEXO II

## II.1 Calibración para Voltaje de entrada Vin

Para obtener las calibraciones en este caso, ha sido necesario variar el voltaje de entrada alimentando con una fuente externa y tomar los valores de los distintos casos en los divisores resistivos de la entrada, tanto con el osciloscopio como con myRIO. El valor de tensión aplicada a la entrada es el valor previo a la tensión rectificadora, que tendrá siempre un valor inferior. En la Tabla 7 se muestran los valores obtenidos y en la Figura 62 las ecuaciones para la calibración:

<b>Voltaje Entrada</b>	<b>Vin Osciloscopio (V)</b>	<b>Vin myRIO (V)</b>	<b>Vin myRIO – Vin Osciloscopio (V)</b>
5	0,462	0,513	0,051
7	0,470	0,520	0,050
9	0,477	0,524	0,047
11	0,483	0,531	0,048
13	0,490	0,537	0,047
15	0,498	0,543	0,045
17	0,508	0,549	0,041
19	0,514	0,554	0,040
21	0,521	0,561	0,040
23	0,529	0,568	0,039
25	0,535	0,574	0,039

Tabla 7: Datos calibración Voltaje de entrada Vin en osciloscopio y myRIO

La ecuación de la recta a aplicar para myRIO es  $y = 0,003x + 0,4976$ . Se vuelve a observar que existe un pequeño *offset* entre los valores del osciloscopio y los leídos por myRIO. En la se muestran los valores obtenidos de las calibraciones y el *offset* que se pondrá como un valor programable mediante un control en el Host VI.



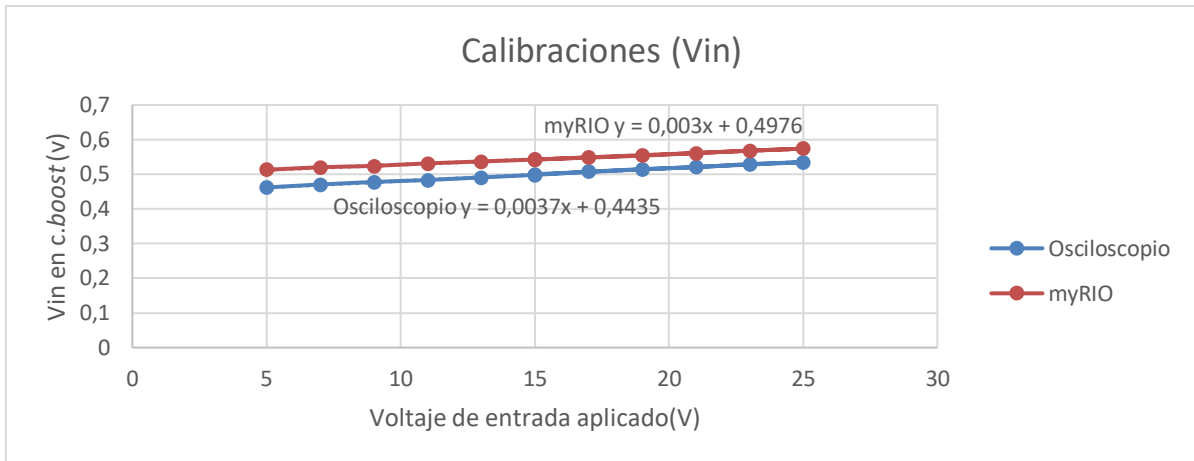


Figura 62: Rectas de calibración para Vin en osciloscopio y myRIO

Voltaje Entrada	Valores con recta de calibración		Pérdidas Vout myRIO – Vout osciloscopio
	Vin myRIO (V)	Vin osciloscopio (V)	
5	5,133	5,000	0,133
7	7,467	7,162	0,305
9	8,800	9,054	-0,254
11	11,133	10,676	0,458
13	13,133	12,568	0,566
15	15,133	14,730	0,404
17	17,133	17,432	-0,299
19	18,800	19,054	-0,254
21	21,133	20,946	0,187
23	23,467	23,108	0,359
25	25,467	24,730	0,737

Tabla 8: Datos para ajuste de offset de Vin

En este caso, en la ecuación de Vin los valores de los parámetros son  $B = 0,003$  y  $C = 0,4976$  y el valor de *offset* tras varias pruebas y las observaciones de la Tabla 8, se fija a 0,25. En la Figura 63 aparece el panel frontal del FPGA VI que permite su configuración y en la Figura 64 se muestra el diagrama de bloques correspondiente.

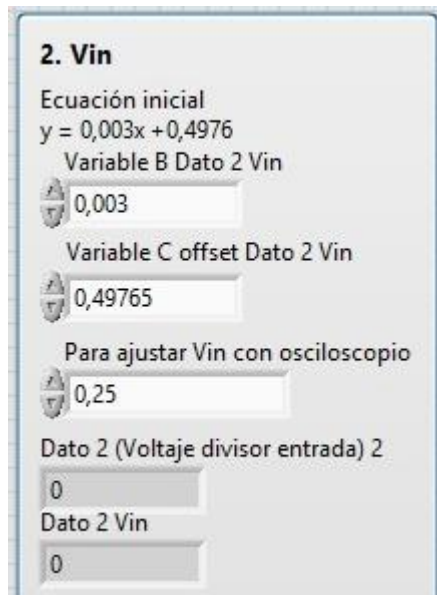


Figura 63: Calibración para Vin en FPGA VI (panel frontal)

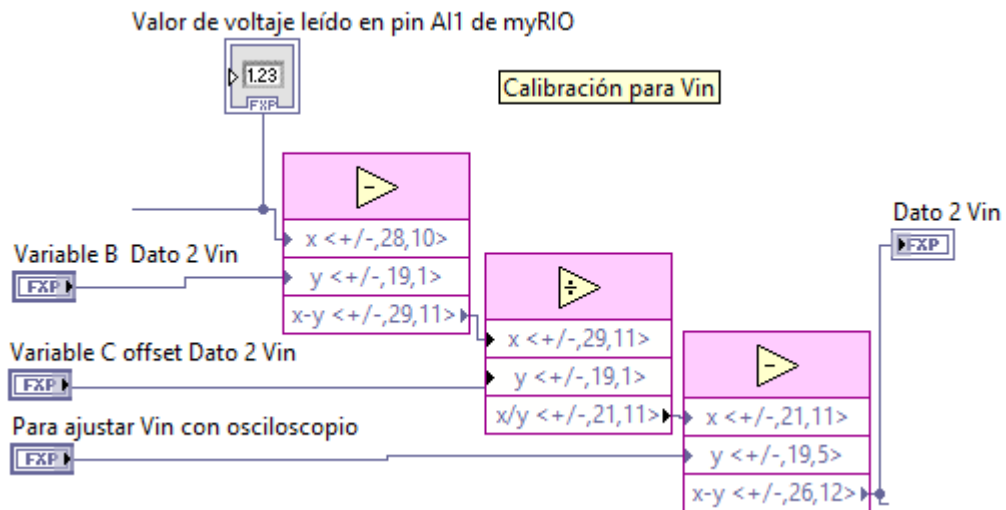


Figura 64: Calibración para Vin en FPGA VI (diagrama de bloques)

## II.2 Calibración para Corriente de entrada Iin

Se utilizó una pinza amperimétrica para medir la corriente de entrada y los valores de voltaje obtenidos por myRIO tras los divisores resistivos en la resistencia de shunt, variando el valor del voltaje de entrada. También se midió en el osciloscopio el voltaje en la resistencia de shunt para observar el error entre las medidas obtenidas con myRIO y poder controlar el valor de *offset* como en los casos anteriores. Se obtuvieron los valores mostrados en la Tabla 9:

<b>Voltaje Entrada</b>	<b>Iin pinza amperimétrica (A)</b>	<b>Vin en Rshunt myRIO (V)</b>	<b>Vin osciloscopio (V)</b>	<b>Vin Rshunt myRIO-Vin osciloscopio (V)</b>
5	0,038	0,09	0,083	0,007
7	0,043	0,13	0,125	0,005
9	0,048	0,17	0,163	0,007
11	0,055	0,19	0,187	0,003
13	0,062	0,24	0,235	0,005
15	0,068	0,28	0,278	0,002
17	0,077	0,31	0,306	0,004
19	0,082	0,34	0,336	0,004
21	0,089	0,38	0,377	0,003
23	0,093	0,41	0,408	0,002
25	0,1	0,46	0,457	0,003

Tabla 9: Datos calibración Corriente de entrada Iin y Vin en Rshunt en myRIO

La siguiente Figura 65 muestra la ecuación de la recta obtenida ( $y = 5,6384x - 0,1143$ ) para los valores necesarios a programar en el FPGA VI, además del valor de *offset*:

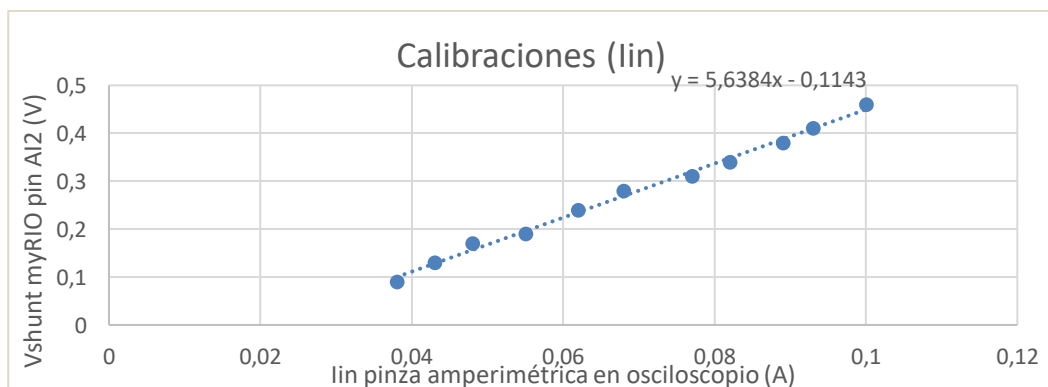


Figura 65: Recta de calibración para Iin en myRIO

En la ecuación de  $I_{in}$  los valores de los parámetros son  $B = 5,6384$  y  $C = -0,1143$  y el valor de *offset* tras varias pruebas y las observaciones de la , se fija a  $0,002$ . En la Figura 66 aparece el panel frontal del FPGA VI que permite su configuración y en la Figura 67 se muestra el diagrama de bloques correspondiente.

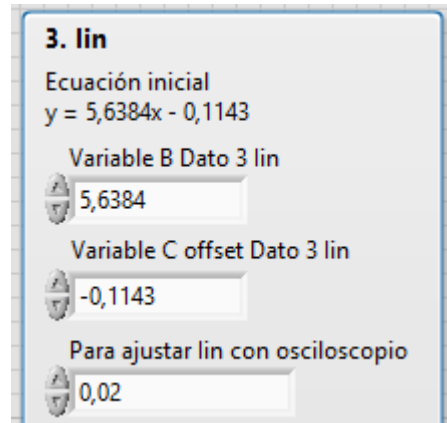


Figura 66: Calibración para  $I_{in}$  en FPGA VI (panel frontal)

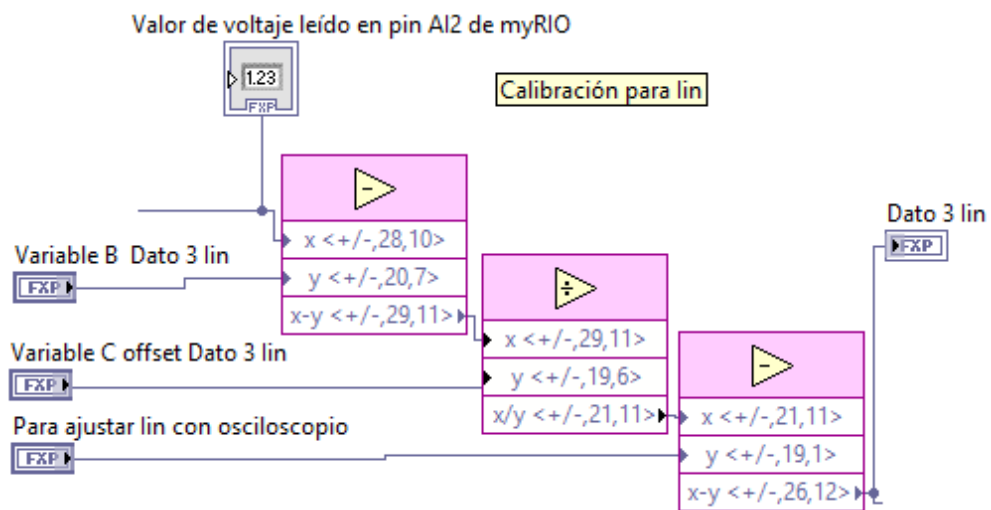


Figura 67: Calibración para  $V_{in}$  en FPGA VI (diagrama de bloques)

# ANEXO III

## III.1 Implementación de planta del convertidor *boost*

Antes de la obtención de los reguladores, se debe obtener la expresión para la planta del convertidor *boost* para CA. Se establece la configuración deseada para los valores del convertidor tales como tensión de entrada ( $v_g$ ), tensión de salida ( $V_{out}$ ) y las de los componentes que lo componen. De la consulta de [20] se extraen las siguientes ecuaciones para el modelado de las ecuaciones de las plantas del lazo de tensión y de corriente a regular. Dichos valores se muestran en la Tabla 10:

Ecuaciones de plantas del convertidor <i>boost</i>	Valores
<p><b>Planta del lazo de tensión</b></p> $G_{vg}(s) = \frac{v_g^2 R}{\frac{2V_{out}}{RCs} + 1}$	$v_g = 230 V$ $V_{out} = \frac{V_{in}}{1-D} = 400 V$ $L = 1,1 mH$ $C = 68 \mu F$
<p><b>Planta del lazo de corriente</b></p> $G_{ID}(s) = \frac{V_{out}}{LS}$	$R = 533,33 \Omega$

Tabla 10: Ecuación, parámetros y valores para modelos de plantas del convertidor *boost*

El script generado en Matlab es el siguiente:

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Modelado plantas convertidor boost %
%      María Murillo Moya      %
%      TFM 2017      %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% Valores de variables %%
% Voltaje de entrada al boost (V)
Vg = 230;
% Voltaje de salida del boost = 37.5 (V)
Vout = 400;
% Valor de la resistencia de carga (Ohmios)
R = 533.33;
% Valor de la bobina (H)
L = 1.1*10^-3;
% Valor del condensador (F)
C = 68*10^-6;

```

```

% Frecuencia del lazo de tensión
f_tension = 100; % 100 Hz
% Frecuencia del lazo de corriente
f_corriente = 100*10^3; % 100 kHz
% Periodo del lazo de tensión
T_tension = 1/f_tension;
% Periodo del lazo de corriente
T_corriente = 1/f_corriente;

%% Función de transferencia de la planta del
%% lazo de tensión
G_vg_num = [0 (vg^2*R)/(2*Vout)];
G_vg_den = [(R*C)/2 1];
G_vg_s = tf(G_vg_num, G_vg_den);
G_vg_z = c2d(G_vg_s, T_tension);

%% Función de transferencia de la planta del
%% lazo de corriente
G_id_num = [0 Vout];
G_id_den = [L 0];
G_id_s = tf(G_id_num, G_id_den);
G_id_z = c2d(G_id_s, T_corriente);

```

## III.2 Implementación de regulador para lazo de tensión

Una vez obtenida la planta del convertidor, con la herramienta SISO, ejecutando el comando *sisotool* de la planta de tensión obtenida, se modela un regulador de tipo PI para el lazo de tensión. Los coeficientes obtenidos son los siguientes:

$$R_1(z) = \frac{B_0 + B_1 z^{-1}}{1 - z^{-1}} = G \times \frac{(z - b)}{(z - 1)}$$

$$B_0 = G = 7,206 \times 10^{-6}$$

$$B_1 = -4,213 \times 10^{-6}$$

$$b = 0,5846$$

Para obtener la FDT en Matlab y su respuesta al escalón en lazo cerrado, se escriben las siguientes líneas de código:

```
%% Función de transferencia del Regulador 1 (lazo tensión)
R1_num = [(7.206*10^-6) - (4.213*10^-6)];
R1_den = [1 -1];
disp('FDT del regulador del lazo de tensión R1(z)')
R1_z = tf(R1_num, R1_den, T_tension)
%% Obtención de respuesta a escalón en lazo cerrado
sisotool(G_vg_z, R1_z)
)
```

Se obtiene la siguiente respuesta, con un tiempo de establecimiento de 353ms.

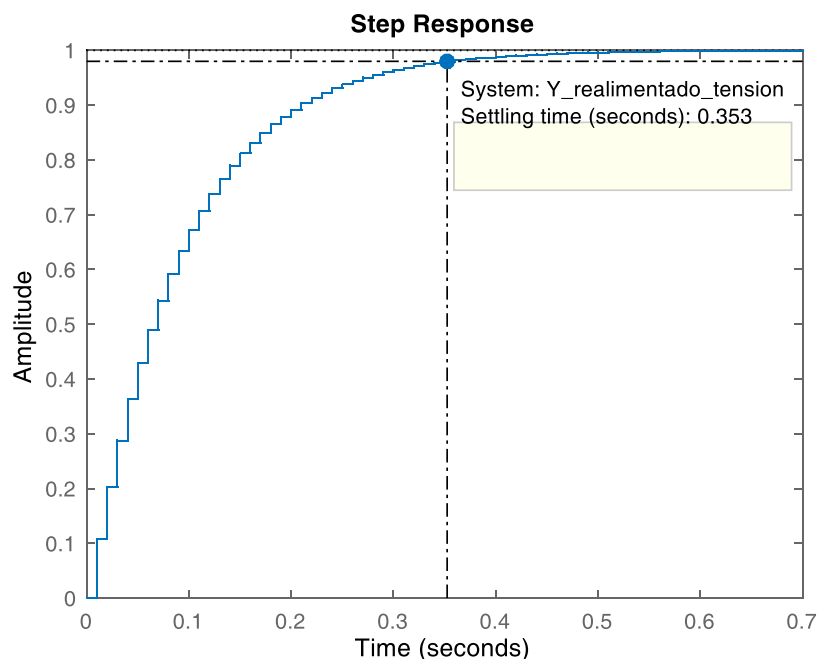


Figura 68: Respuesta al impulso del regulador del lazo de tensión

### III.3 Implementación de regulador para lazo de corriente

Al igual que para el regulador de tensión, se modela un regulador de tipo PI para el lazo de corriente. Los coeficientes obtenidos son los siguientes:

$$R_2(z) = \frac{B_0 + B_1 z^{-1}}{1 - z^{-1}} = G \times \frac{(z - b)}{(z - 1)}$$

$$B_0 = G = 0,28$$

$$B_1 = -0,2625$$

$$b = 0,9375$$

Para obtener la FDT en Matlab se escriben las siguientes líneas de código:

```
%% Función de transferencia del Regulador 2 (lazo corriente)
R2_num = [(0.28) -(0.2625)];
R2_den = [1 -1];
disp('FDT del regulador del lazo de corriente R2(z)')
R2_z = tf(R2_num, R2_den, T_corriente)
%% Obtención de respuesta a escalón en lazo cerrado
sisotool(G_id_z, R2_z)
```

Se observa que el tiempo de establecimiento es mucho menor que en el lazo de tensión, siendo de 0,192 ms.

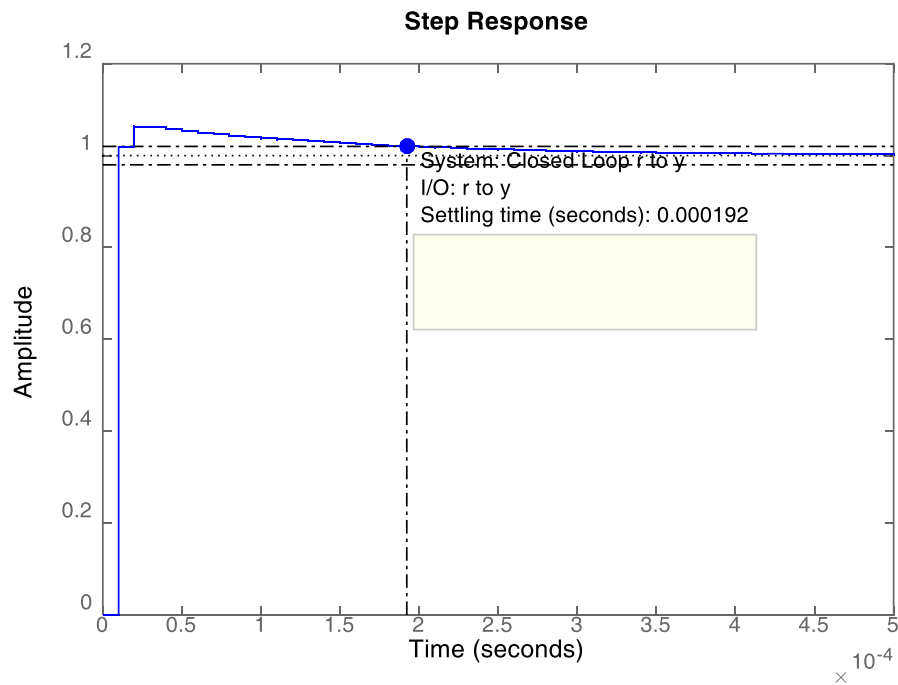


Figura 69: Respuesta al impulso del regulador del lazo de corriente