

Fernando Rodríguez Saavedra

**Prototipo de un controlador digital para
convertidores continua-continua de elevadas
prestaciones**

**Treball Fi de Màster
dirigit pel Dr. Ramon Leyva**

Màster en Enginyeria Industrial



UNIVERSITAT ROVIRA I VIRGILI

Tarragona

2017

Índice

Índice	i
Índice de figuras	iii
Índice de tablas	vi
Acrónimos	vii
Resumen	viii
Resum	viii
Abstract	viii
1 Introducción.....	1
1.1 Motivación y punto de partida	1
1.2 Objetivos.....	1
1.3 Organización de la memoria	2
2 Conceptos previos.....	3
2.1 Módulos reguladores de voltaje (VRM)	3
2.2 Convertidor <i>buck</i>	3
2.3 Métodos de control.....	7
2.3.1 Mantenedor de orden cero (ZOH).....	7
2.3.2 Mantenedor de orden uno (FOH).....	8
2.3.3 Aproximación <i>Tustin</i>	8
2.4 Lugar geométrico de las raíces	9
2.5 FPGA.....	11
3 Acondicionamiento de señal.....	14
3.1 Buffer de amplificadores operacionales.....	14
3.1.1 Análisis teórico	15
3.1.2 Simulación PSIM	19
3.1.3 Implementación de la PCB.....	21
3.1.4 Integración Buffer-ADC	27
3.2 Divisor de tensión y seguidor.....	30
3.2.1 Análisis teórico	30
3.2.2 Simulaciones PSIM.....	31
3.2.3 Implementación de la PCB.....	33
4 Caracterización <i>buck</i>	38
4.1 Programación <i>buck</i> y comprobación a partir del osciloscopio	38
5 Toma de datos con Chipscope	44
5.1 Resultados obtenidos Buffer	44
5.2 Resultados obtenidos sistema.....	48

6	Control.....	51
6.1	Modelo convertidor buck en lazo abierto en Matlab	51
6.2	Diseño del controlador	56
7	Conclusiones y trabajos futuros.....	63
8	Referencias	64

Índice de figuras

Figura 1. Convertidor <i>buck</i> síncrono sobre el que se realizará el control que ocupa este trabajo.....	1
Figura 2. Ejemplo de módulo de regulación de voltaje.	3
Figura 3. Esquema típico de un convertidor <i>buck</i>	4
Figura 4. Esquema de un convertidor <i>buck</i> cuando el MOSFET conduce.	4
Figura 5. Esquema de un convertidor <i>buck</i> cuando el MOSFET no conduce.....	4
Figura 6. Ejemplo de tensiones de entrada y salida de un convertidor <i>buck</i> con un ciclo de trabajo de 0,24.	6
Figura 7. Ejemplo de tensión en la bobina de un convertidor <i>buck</i> con un ciclo de trabajo de 0,24.....	6
Figura 8. Ejemplo de corriente en la bobina de un convertidor <i>buck</i> con un ciclo de trabajo de 0,24.....	7
Figura 9. Diagrama de bloques de la discretización de un sistema al aplicar un zoh.	7
Figura 10. Ejemplo de sistema realimentado.	9
Figura 11. Lugar geométrico de las raíces en el plano s.....	10
Figura 12. Lugar de raíces de un sistema en el plano z.	10
Figura 13. FPGA Sparta-3E.	11
Figura 14. Esquema del modelo VHDL.	12
Figura 15. Ejemplo de un fichero VHDL, el cual emula el comportamiento de un <i>flip-flop</i> tipo D.....	13
Figura 16. Estructura de referencia común de los ADCs.....	14
Figura 17. Buffer seleccionado para su implementación.....	15
Figura 18. Circuitería correspondiente a la salida negativa del buffer.....	15
Figura 19. Circuitería correspondiente a la salida positiva del buffer.	16
Figura 20. Filtro pasa-bajas pasivo.	18
Figura 21. Diagrama bode del filtro pasa-bajas.....	19
Figura 22. Simulación PSIM del buffer (1).	20
Figura 23. Simulación PSIM del buffer (2).	20
Figura 24. Simulación PSIM del buffer a alta frecuencia.....	21
Figura 25. Distribución de las entradas y de la salida COM del ADC en el kit de evaluación.....	22
Figura 26. Esquema del ADA4891-2.	23
Figura 27. Esquema del diseño del buffer.	23
Figura 28. <i>Layout</i> de los elementos del buffer en la PCB (1).	24
Figura 29. <i>Layout</i> de los elementos del buffer en la PCB (2).	25
Figura 30. PCB del buffer implementado.	26
Figura 31. PCB del buffer implementado con los componentes soldados.	26
Figura 32. Resultados obtenidos en la salida del buffer al introducir una señal de entrada de 500 mV de amplitud, 100 KHz y 100 mV de offset.	27
Figura 33. Esquema de la distribución de las entradas INP y INN y de la salida COM del ADC en el kit de evaluación.	28
Figura 34. Lente de aumento Mantis utilizada para la soldadura sobre el kit de evaluación del ADC.	28
Figura 35. Soldadura sobre R22 para atacar a la señal COM del ADC.....	29
Figura 36. Soldaduras sobre R13 y R14 para atacar a las señales de INN e INP.	29
Figura 37. Conjunto formado por el buffer implementado y el kit de evaluación del ADC.....	30
Figura 38. Rango de valores de entrada del ADC.	30
Figura 39. Salida del convertidor <i>buck</i> con un divisor de tensión a continuación.	31
Figura 40. Esquema del divisor de tensión y el buffer.....	31

Figura 41. Resultados obtenidos al colocar el divisor de tensión a la entrada del buffer.	32
Figura 42. Esquema del divisor de tensión, seguidor y buffer.	32
Figura 43. Resultados obtenidos al incorporar el seguidor de tensión entre el divisor y el buffer.	33
Figura 44. Esquema de la PCB del divisor de tensión y del seguidor.	34
Figura 45. <i>Layout</i> de la PCB del divisor de tensión y del seguidor.	35
Figura 46. PCB del divisor de tensión y del seguidor.	36
Figura 47. PCB del divisor de tensión y del seguidor con los componentes soldados.	37
Figura 48. Entidades en VHDL utilizadas para el control del <i>buck</i> .	38
Figura 49. Resultados obtenidos de las señales <i>control_buck</i> (azul), <i>low_buck</i> (rosa) y <i>high_buck</i> (amarillo).	39
Figura 50. Resultados obtenidos de las señales <i>low_buck</i> (azul) y <i>high_buck</i> (amarillo).	39
Figura 51. Resultados obtenidos de las señales <i>control_buck</i> (azul), <i>control_load</i> (rosa) y <i>low_load</i> (amarillo).	40
Figura 52. Código VHDL para la implantación de un ciclo de trabajo del 50%.	40
Figura 53. Resultados obtenidos del convertidor <i>buck</i> con un ciclo de trabajo del 50% de las señales de <i>control_buck</i> (azul), tensión de salida (amarillo) y corriente de la bobina (verde).	41
Figura 54. Código VHDL para la implementación de un ciclo de trabajo del 25%.	41
Figura 55. Resultado obtenido de la señal del <i>control_buck</i> al aplicar un ciclo de trabajo del 25%.	42
Figura 56. Resultados obtenidos de las señales de <i>low_buck</i> (azul) y <i>high_buck</i> (rosa) al aplicar un ciclo de trabajo del 25%.	42
Figura 57. Resultados obtenidos del convertidor <i>buck</i> con un ciclo de trabajo del 25% de las señales de tensión de salida (amarillo) y corriente de la bobina (verde).	43
Figura 58. Entidad VHDL para la lectura de la salida del ADC.	44
Figura 59. Configuración de la conexión del buffer, el ADC y la FPGA.	44
Figura 60. Resultados obtenidos al introducir una señal de entrada de 50 mV de amplitud, 100 KHz de frecuencia y 0 V de offset.	45
Figura 61. Resultados obtenidos al introducir una señal de entrada de 250 mV de amplitud, 100 KHz de frecuencia y 0 V de offset.	45
Figura 62. Resultados obtenidos al introducir una señal de entrada de 250 mV de amplitud, 1 MHz de frecuencia y 0 V de offset.	46
Figura 63. Resultados obtenidos al introducir una señal de entrada de 250 mV de amplitud, 100 KHz de frecuencia y 100 mV de offset.	46
Figura 64. Resultados obtenidos al introducir una señal de entrada de 250 mV de amplitud, 100 KHz de frecuencia y 250 mV de offset.	47
Figura 65. Configuración de la FPGA, el convertidor <i>buck</i> , el divisor de tensión y seguidor, el <i>buffer</i> y el ADC.	48
Figura 66. Resultados obtenidos con Chipscope de la tensión de salida del buck con un ciclo de trabajo del 50%.	49
Figura 67. Resultados obtenidos con Chipscope de la tensión de salida del buck con un ciclo de trabajo del 25%.	50
Figura 68. Diagrama de bloques del comportamiento de un convertidor <i>buck</i> .	52
Figura 69. Código Matlab para la inicialización de parámetros.	53
Figura 70. Representación del valor del ciclo de trabajo al trabajar al 24%.	53
Figura 71. Resultados obtenidos de la tensión de entrada.	54
Figura 72. Resultados obtenidos de la caída de tensión en la bobina.	54
Figura 73. Resultados obtenidos de la corriente en la bobina.	55
Figura 74. Resultados obtenidos de la tensión de salida.	55

Figura 75. Resultados obtenidos de la tensión de salida en régimen transitorio.	56
Figura 76. Diagrama típico de un control digital aplicando un zoh.	57
Figura 77. Lugar geométrico de las raíces de la planta de nuestro sistema al ser discretizada.....	58
Figura 78. Polos y ceros añadidos para el diseño del controlador.	58
Figura 79. Lugar geométrico de las raíces del sistema una vez incluido el controlador.	59
Figura 80. Respuesta del sistema a una perturbación en forma de escalón unitario. ..	59
Figura 81. Respuesta de la salida del controlador a una perturbación en forma de escalón unitario.	60
Figura 82. Diagrama de bloques del sistema planta + controlador.....	61

Índice de tablas

Tabla 1. Componentes seleccionados para implementar el buffer.....	22
Tabla 2. Componentes seleccionados para la implementación de la PCB que incluye el divisor de tensión y el seguidor.....	33

Acrónimos

ADC Analog to Digital Converter

COM/CM Common Mode

DC-DC Direct Current – Direct Current

FPGA Field Programmable Gate Array

FOH First-Order Hold

IN Entrada

MOSFET Metal Oxide Semiconductor Field Effect Transistor

OUT Salida

PCB Printed Circuit Board

PID Proportional-Integral-Derivative

SMA Sub Miniature version A

VHDL Very High Speed Integrated Circuit

VRM Voltage Regulator Module

ZOH Zero-Order Hold

Resumen

Este trabajo de fin de máster ha sido concebido con la idea de elaborar un análisis de la viabilidad técnica y la posterior implementación de un prototipo de controlador digital para su utilización en fuentes conmutadas de elevadas prestaciones. Con este fin, se ha realizado la elección de los diferentes componentes y subsistemas que integrarían el acondicionamiento de señal pertinente, así como también la confección de placas de circuito impreso y el ensamblado de los componentes adecuados sobre estas. Para continuar, se ha llevado a cabo la programación VHDL necesaria para la adquisición de datos y para el control de las fuentes conmutadas a través de algoritmos numéricos. De manera adicional, se ha verificado el comportamiento de los subsistemas particulares y del sistema en conjunto a través de diferentes simulaciones. Por último, se ha realizado la corroboración de dichas simulaciones a través de medidas de osciloscopio sobre el prototipo realizado.

Resum

Aquest treball de fi de màster ha estat concebut amb la idea d'elaborar un anàlisi de la viabilitat tècnica i la posterior implementació d'un prototip de controlador digital per a la seva utilització en fonts commutades d'elevades prestacions. Amb aquesta finalitat, s'ha realitzat l'elecció dels diferents component i subsistemes que integrarien l'acondicionament dels senyals, així com també el disseny i la confecció de plaques de circuit imprès i l'assemblatge dels components requerits sobre aquestes. Per a continuar, s'ha dut a terme la programació VHDL necessària per a l'adquisició de dades a utilitzar en el control de les fonts commutades a través d'algorismes numèrics. A més a més, s'ha verificat el comportament dels subsistemes particulars i del sistema en el seu conjunt a través de diferents simulacions. Per últim, s'han corroborat aquestes simulacions mitjançant mesures d'oscil·loscopi sobre el prototip realitzat.

Abstract

This masters' thesis is aiming at performing a technical viability analysis and the subsequent implementation of a digital controller prototype for high performance switching converters. With this propose, components and subsystems election, PCB confection and appropriate component assembly have been carried out. Then, we have performed the VHDL programming necessary for data acquisition and for controlling of switching converters through digital algorithm. Furthermore, we have completed the verification of the expected behaviour of each subsystem and the overall system using accurate simulations. Finally, we corroborate that simulations results exhibits an excellent similarity with oscilloscope waveforms.

1 Introducción

Hoy en día, debido al rápido progreso de la tecnología integrada, el nivel de la tensión de alimentación de microprocesadores y otros circuitos integrados está decreciendo cada vez más rápidamente, con el propósito de aumentar su eficiencia energética.

Debido a esta tendencia, la realización de convertidores de diferentes niveles de tensión de altas prestaciones se ha convertido en un elemento clave en la electrónica actual. Al trabajar con tensiones tan pequeñas, las variaciones de carga que pueden producirse en un microprocesador pueden afectar considerablemente a las tensiones de salida de los convertidores utilizados.

Para evitar estos inconvenientes, es imprescindible realizar un acondicionamiento de señal y un control adecuado de dichos convertidores.

1.1 Motivación y punto de partida

Este trabajo nace con el objetivo de realizar el diseño, la implementación física y la programación necesaria de los diferentes elementos correspondientes al control de un convertidor DC-DC *buck* síncrono. Para poder explicar las tareas llevadas a cabo, en primer lugar debemos definir el punto de partida desde el cual se ha comenzado a trabajar.

Este proyecto puede entenderse como la continuación del trabajo realizado en [1], en el que se realizó la implementación de un convertidor *buck* síncrono de elevadas prestaciones. En la figura 1 se muestra el diseño final del mismo.

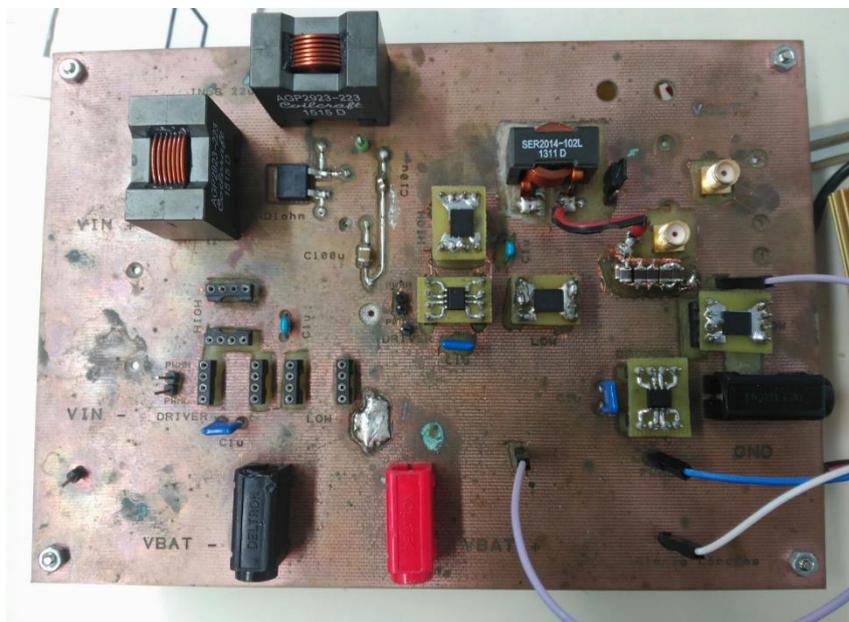


Figura 1. Convertidor *buck* síncrono sobre el que se realizará el control que ocupa este trabajo

1.2 Objetivos

Una vez definido el alcance del trabajo, se muestran los pasos a seguir para la realización del control en lazo cerrado del convertidor. Es decir, para hacer más tratable esta tarea,

hemos dividido el trabajo en una serie de objetivos que nos ayudarán a cumplir la finalidad última de este proyecto. Los objetivos particulares son los siguientes:

- Familiarizarse con los programas necesarios para la correcta realización de las diferentes tareas correspondientes: PSIM, *Design Spark*, Matlab, ISE-Xilinx y Chipscope.
- Estudiar las hojas de datos y la utilización de los componentes y sistemas que formarán parte de nuestro proyecto: MAX12555, kit de evaluación de MAX12555, Spartan3E, ADA4891-2.
- Diseño y fabricación de placas de circuito impreso y soldadura de componentes correspondientes al acondicionamiento de señal necesario para el control del convertidor *buck* síncrono.
- Programación VHDL de la adquisición de datos y los algoritmos numéricos de control y señales del actuador para el control digital de convertidores continua-continua utilizando el software ISE-Xilinx y la herramienta Chipscope.
- Simulación en Matlab del convertidor *buck* síncrono y posterior diseño del controlador que nos regulará el comportamiento del convertidor, logrando que este trabaje dentro de los márgenes preestablecidos.

1.3 Organización de la memoria

La memoria de este proyecto ha quedado dividida en una serie de capítulos.

El primer capítulo se trata de la introducción en la que nos encontramos ahora en la cual se explican el punto de partida, los objetivos y la composición del trabajo en sí.

En el segundo de los capítulos se ha realizado una introducción teórica donde se explican los conceptos preliminares que hay que conocer para poder comprender el tema del trabajo. En este se expone información sobre VRMs, convertidores *buck*, tipos de controladores digitales, el lugar geométrico de las raíces y dispositivos FPGAs.

En el tercer capítulo se expone como se ha desarrollado todo el acondicionamiento de señal pertinente al control del convertidor.

En cuanto al cuarto capítulo, se ha explicado cómo se realiza el control del convertidor utilizando la FPGA.

En el quinto capítulo, se procede a explicar cómo se ha realizado la toma de datos a través de la herramienta Chipscope y también se muestran resultados obtenidos.

Por lo respecta al sexto capítulo, se ha dedicado a la exposición del diseño del controlador.

Para continuar, en el séptimo capítulo se pueden apreciar las conclusiones finales del trabajo junto a los posibles trabajos futuros a realizar

Por último, en el octavo capítulo se exponen todas las fuentes utilizadas para documentarse debidamente y las referencias utilizadas a la hora de realizar el trabajo.

2 Conceptos previos

Para poder explicar correctamente el desarrollo de este proyecto, es conveniente definir previamente una serie de conceptos clave.

2.1 Módulos reguladores de voltaje (VRM)

Como se ha comentado anteriormente y tal como exponen Panov y Jovanovic en [2], para minimizar el consumo de potencia y aumentar la velocidad, la siguiente generación de microprocesadores deberá operar a voltajes significativamente bajos y a corrientes más altas que la generación actual. Al mismo tiempo, estos microprocesadores requerirán una elevada precisión en la regulación de dicho voltaje que no puede ser alcanzada por un sistema de alimentación centralizado.

Esta regulación precisa puede lograrse con sistemas de alimentación distribuidos en los que un módulo de regulación de voltaje (VRM), que está localizado en la misma placa base junto a la carga, suministra una alimentación de alta calidad al microprocesador. Estos módulos suelen tratarse de un convertidor conmutado DC-DC de tipo *buck* el cual trabaja típicamente con una tensión de entrada de 5 o 12 V, la cual se encarga de convertir a tensiones más bajas para satisfacer las necesidades del microprocesador que alimenta. De esta manera, gracias a la implementación mediante VRMs es posible la implantación de microprocesadores con tensiones de alimentación diferentes en una misma placa base.

Generalmente, los VRM requieren disponer de una alta densidad de potencia y operar con elevada eficiencia. Para satisfacer estos requerimientos y disponer de una rápida respuesta transitoria, la conversión de voltaje se debe realizar a muy alta frecuencia de conmutación lo que representa un serio desafío para el diseño. En la figura 2 puede verse un ejemplo comercial de un VRM.

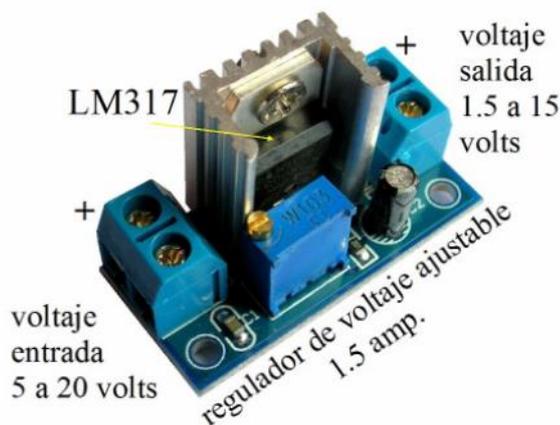


Figura 2. Ejemplo de módulo de regulación de voltaje.

2.2 Convertidor *buck*

En el mundo de la electrónica, existen varios tipos de convertidores continua-continua y el convertidor *buck* es uno de ellos. Como podemos ver en [3], este convertidor, que suele ser el utilizado para la implementación de VRMs, se basa en un inductor (bobina) que se controla por medio de dos semiconductores los que alternan la conexión del inductor bien a la fuente de alimentación o bien a tierra, consiguiendo así reducir el valor de tensión continua de la salida en función al ciclo de trabajo (tiempo de conducción del semiconductor conectado a la alimentación).

En la figura 3 se muestra el esquema de un convertidor tipo *buck*.

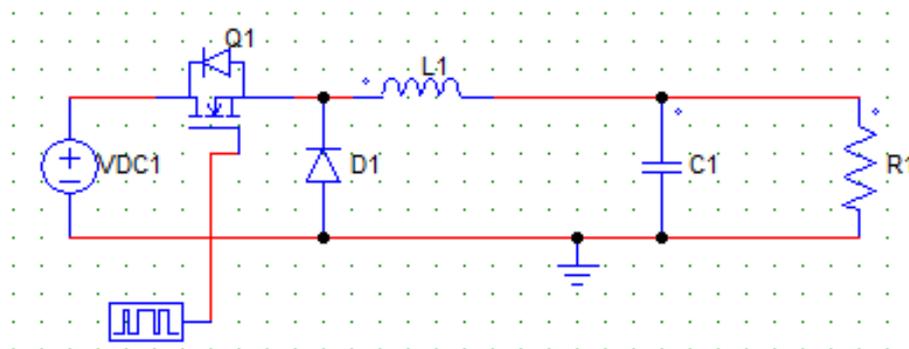


Figura 3. Esquema típico de un convertidor *buck*.

Para poder explicar el funcionamiento de este sistema, se ha analizado en sus dos estados: cuando el interruptor formado por el MOSFET Q1 está cerrado y cuando está abierto:

- Interruptor cerrado: En este momento, la bobina está conectada directamente a la fuente de tensión de entrada y la intensidad que pasa por este aumenta linealmente, siendo la tensión del inductor:

$$V_L = V_i - V_o \quad (1)$$

En la figura 4 puede verse como sería el esquema del circuito en este momento.

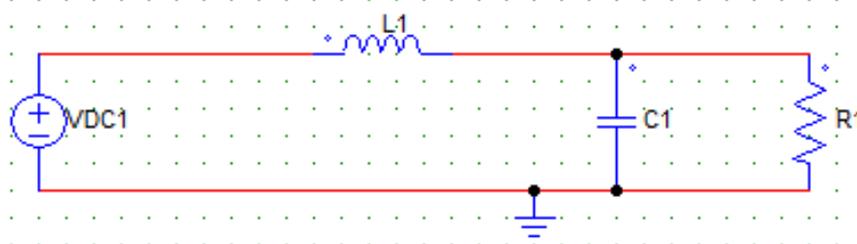


Figura 4. Esquema de un convertidor *buck* cuando el MOSFET conduce.

- Interruptor abierto: En este caso, tenemos la bobina conectada entre la tensión de salida y tierra, por lo tanto la corriente que pasa a través de este descende linealmente, siendo la expresión de la tensión la siguiente:

$$V_L = -V_o \quad (2)$$

Podemos ver el esquema correspondiente al circuito con el interruptor abierto en la figura 5.

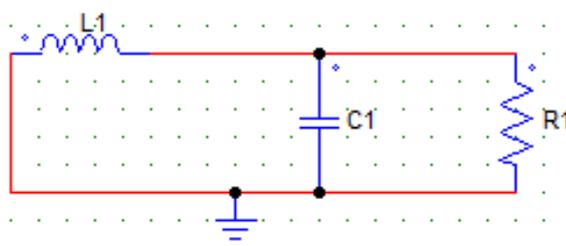


Figura 5. Esquema de un convertidor *buck* cuando el MOSFET no conduce.

Sabemos que la expresión que rige el comportamiento de la tensión en función de la intensidad que pasa por una bobina es la siguiente:

$$V_L = L \cdot \frac{di_L}{dt} \quad (3)$$

Si llamamos al tiempo en el que conduce el transistor Q_1 t_{on} y al tiempo en que no t_{off} podemos escribir el incremento de la intensidad de la corriente en cada situación de la siguiente manera:

$$\Delta I_{L_{on}} = \int_0^{t_{on}} \frac{v_L}{L} \cdot dt = \frac{(V_i - V_o) \cdot t_{on}}{L} \quad (4)$$

$$\Delta I_{L_{off}} = \int_0^{t_{off}} \frac{v_L}{L} \cdot dt = \frac{-V_o \cdot t_{off}}{L} \quad (5)$$

Si asumimos que la energía almacenada en cada ciclo de conmutación es igual, la suma de los incrementos de corriente en cada estado tiene que ser cero, por lo que tenemos:

$$\frac{(V_i - V_o) \cdot t_{on}}{L} - \frac{V_o \cdot t_{off}}{L} = 0 \quad (6)$$

Para seguir analizando el convertidor, supondremos que cada ciclo de conmutación dura un tiempo T y definiremos el tiempo de conducción como:

$$t_{on} = D \cdot T \quad (7)$$

Siendo D el ciclo de trabajo del sistema (tanto por uno del tiempo de cada ciclo en que el transistor conduce). De esta manera el tiempo de apagado será el correspondiente a la siguiente expresión:

$$t_{off} = T - D \cdot T \quad (8)$$

Por lo tanto, podemos reescribir la expresión 6 como:

$$(V_i - V_o) \cdot D \cdot T - V_o \cdot (T - D \cdot T) = 0 \quad (9)$$

Finalmente, obtenemos que la ecuación que relaciona las tensiones de salida y entrada corresponde a:

$$V_o = D \cdot V_i \quad (10)$$

Así, se puede observar como a través del ciclo de trabajo puede convertirse el valor de entrada de tensión al que se desee en la salida (siempre que este sea más pequeño que la tensión de alimentación, obviamente). En la figura 6 se muestran cómo serían las señales de entrada y salida de un convertidor *buck* fijando un ciclo de trabajo de 0,24 y una tensión V_{DC} de 5 V en régimen permanente.

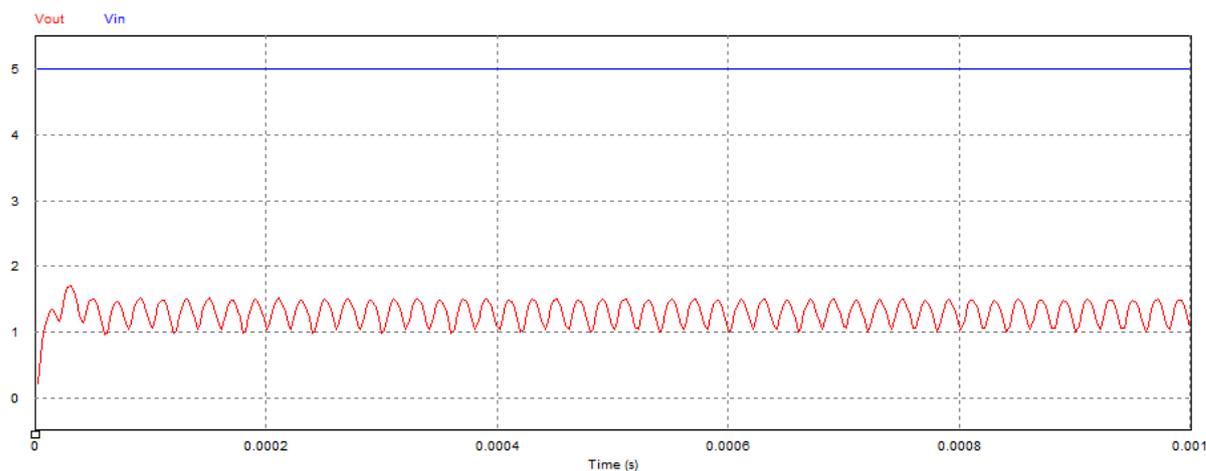


Figura 6. Ejemplo de tensiones de entrada y salida de un convertidor *buck* con un ciclo de trabajo de 0,24.

Se puede ver cómo se produce un rizado sobre la tensión de salida, este rizado depende de los valores de la inductancia y de la capacidad. En este caso, los resultados son los correspondientes a unos valores de 1 μH y 100 μF respectivamente. Seguidamente en las figuras 7 y 8 pueden verse la tensión y la corriente en la bobina respectivamente.

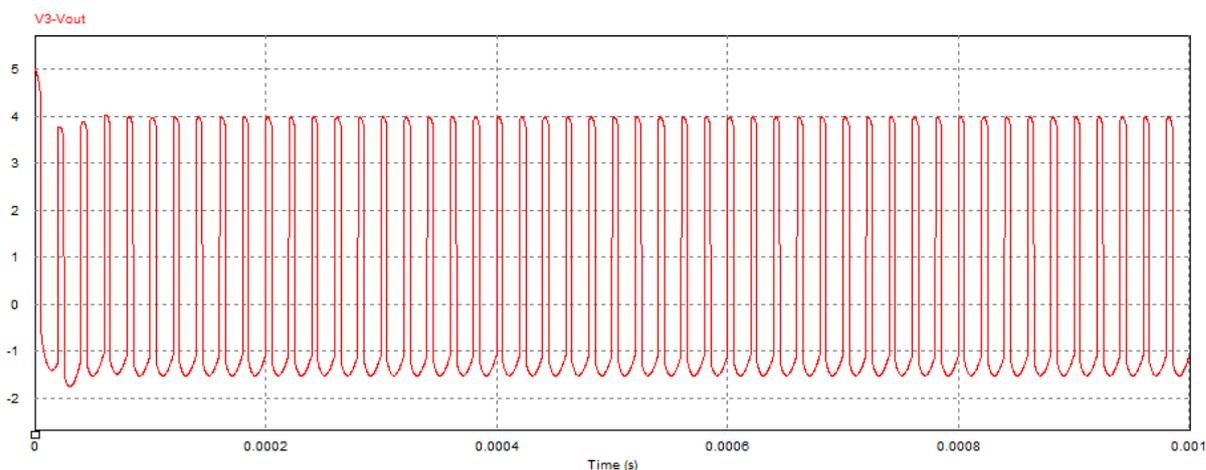


Figura 7. Ejemplo de tensión en la bobina de un convertidor *buck* con un ciclo de trabajo de 0,24.

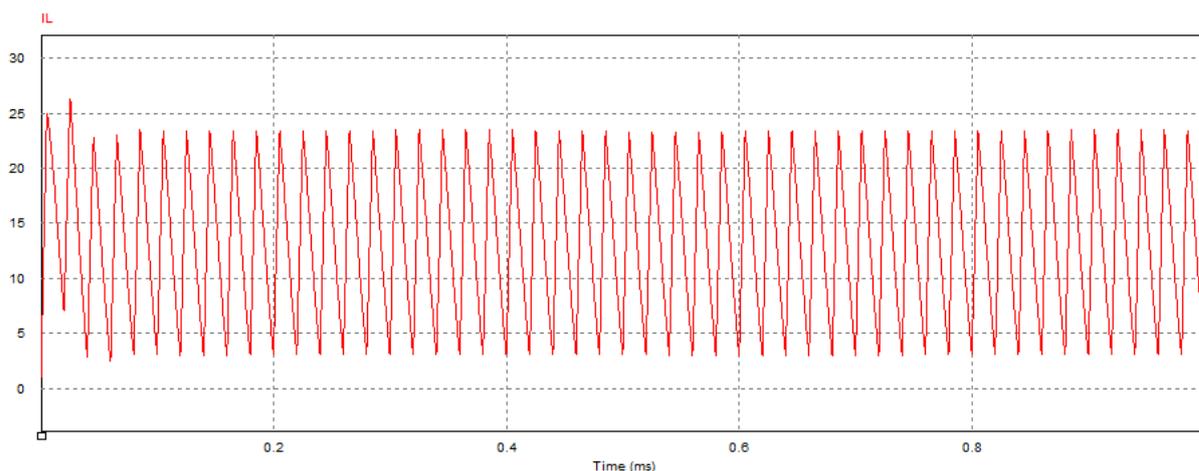


Figura 8. Ejemplo de corriente en la bobina de un convertidor *buck* con un ciclo de trabajo de 0,24.

2.3 Métodos de control

Para realizar el diseño e implantación de un controlador digital es indispensable conocer cómo realizar la conversión del dominio continuo al discreto y viceversa. Como se expone en [4] y [5], existen varios métodos y cada uno de ellos cuenta con una serie de características, que lo hacen más convenientes para unas ocasiones u otras. A continuación se exponen los más comunes.

2.3.1 Mantenedor de orden cero (ZOH)

Uno de los métodos más comúnmente utilizado es el mantenedor de orden cero (*zero-order hold*). Este método describe el efecto de convertir una señal de tiempo discreto a una de tiempo continuo manteniendo el valor de cada muestra durante un período. En la figura 9 puede verse como sería el diagrama de bloques de la discretización de la función de transferencia $H(z)$ utilizando este método.

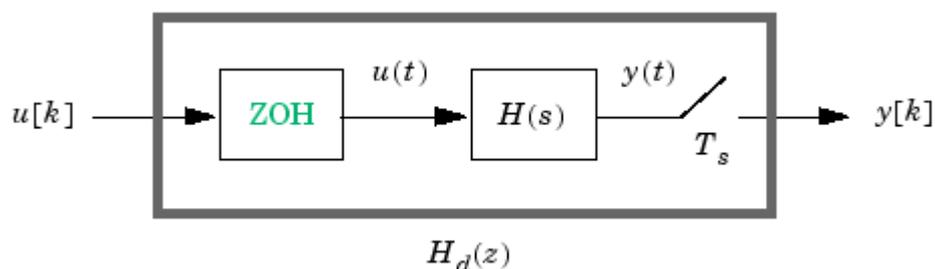


Figura 9. Diagrama de bloques de la discretización de un sistema al aplicar un zoh.

El mantenedor de orden cero genera una señal de tiempo continuo $u(t)$ manteniendo el valor de $u[k]$ constante durante un período de muestreo, tal y como se muestra en la siguiente expresión:

$$u(t) = u(k), \quad kT_s \leq t \leq (k+1)T_s \quad (11)$$

La señal $u(t)$ es la entrada al sistema continuo $H(s)$. La salida $y[k]$ es el resultado del muestreo de $y(t)$ cada T_s segundos. A pesar de esto, la conversión discreta a continuo mediante el mantenedor de orden cero tiene una serie de limitaciones:

- Este método no puede convertir sistemas con polos en $z=0$.

- Para sistemas de tiempo discreto con polos reales negativos, la conversión utilizando el ZOH produce un sistema continuo con un orden mayor. Esto sucede debido a que un polo real negativo en el dominio z se traduce en un valor puro imaginario en el dominio s .

2.3.2 Mantenedor de orden uno (FOH)

El método utilizando el mantenedor de orden uno (*first-order hold*) proporciona una equivalencia exacta entre sistemas de tiempo continuo y discreto en el dominio temporal para entrada lineales a trozos. A diferencia del mantenedor de orden cero, este método utiliza una interpolación lineal para convertir las muestras $u[k]$ en $u(t)$, tal como puede verse en la siguiente expresión:

$$u(t) = u[k] + \frac{t - kT_s}{T_s} \cdot (u[k+1] - u[k]), \quad kT_s \leq t \leq (k+1)T_s \quad (12)$$

Este método es más preciso que el ZOH, pero también es mucho más lento, ya que para realizar la conversión de una muestra $u[k]$ necesita conocer el valor de la siguiente muestra $u[k+1]$.

2.3.3 Aproximación *Tustin*

La aproximación *Tustin* o bilineal proporciona la mejor equivalencia entre sistemas de tiempo continuo y tiempo discreto en el dominio de frecuencia. Este método relaciona las funciones de transferencia del dominio s con las del dominio z utilizando la siguiente equivalencia:

$$z = e^{sT_s} \approx \frac{\left(1 + \frac{sT_s}{2}\right)}{\left(1 - \frac{sT_s}{2}\right)} \quad (13)$$

En conversiones de continuo a discreto, la discretización $H_d(z)$ de la función de transferencia continua $H(s)$ sería la que vemos en la siguiente expresión:

$$H_d(z) = H_d(s'), \quad s' = \frac{2(z-1)}{T_s(z+1)} \quad (14)$$

De manera similar, en conversiones de discreto a continuo, la expresión de $H(s)$ sería la siguiente:

$$H(s) = H_d(z'), \quad z' = \frac{\left(1 + \frac{sT_s}{2}\right)}{1 - \frac{sT_s}{2}} \quad (15)$$

Utilizando este método los estados no se conservan. En el caso que ocupa este proyecto, ya se necesita que la conversión se haga lo más rápido posible y no precisamos tampoco una precisión muy elevada, se utilizará el método del mantenedor de orden cero.

2.4 Lugar geométrico de las raíces

En el momento de diseñar el controlador digital, se realizará a través de la posición de los polos y ceros en el lugar geométrico de las raíces, así que, es indispensable exponer bien este concepto, para ello se ha utilizado en lo expuesto en [6].

Supongamos que tenemos un sistema realimentado como el que aparece en la figura 10.

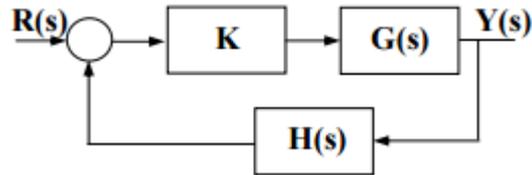


Figura 10. Ejemplo de sistema realimentado.

La función de transferencia del sistema sería la descrita a continuación:

$$\frac{Y(s)}{R(s)} = \frac{K \cdot G(s)}{1 + K \cdot G(s) \cdot H(s)} \quad (16)$$

La ecuación característica de este sistema corresponde al denominador de la función de transferencia. Si se iguala ésta a cero, se obtiene la siguiente expresión, que determinará cuando el sistema es estable:

$$1 + K \cdot G(s) \cdot H(s) = 0 \quad (17)$$

Así, para que el sistema sea estable se deben cumplir dos condiciones como son la condición de módulo y la condición de argumento que pueden verse respectivamente a continuación:

$$|K \cdot G(s) \cdot H(s)| = 1 \quad (18)$$

$$\angle K \cdot G(s) \cdot H(s) = 180^\circ \quad (19)$$

De esta forma puede entenderse que los sistemas serán estables si los polos de este se encuentran en el semiplano positivo del plano s . Dada la situación de trabajar con sistemas digitales, que es el caso que ocupa este proyecto, los polos del sistemas deberán localizarse dentro del círculo unidad.

Además de determinar la estabilidad del sistema, esta herramienta también puede utilizarse para determinar el valor del tiempo de establecimiento y del amortiguamiento del sistema. En la figura 11 pueden verse las zonas que aseguran un tiempo de establecimiento y un porcentaje de amortiguamiento concretos del lugar de raíces de un sistema en el plano s .

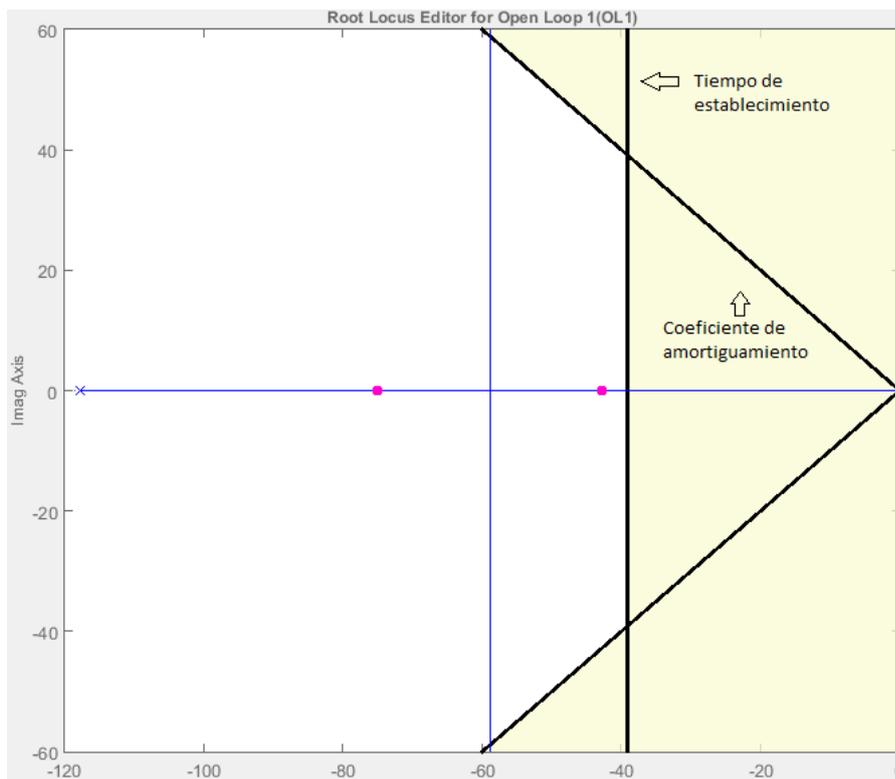


Figura 11. Lugar geométrico de las raíces en el plano s.

En la figura 12 puede verse como se delimitan las zonas correspondientes a un tiempo de establecimiento y un porcentaje de amortiguamiento concretos, en este caso, en el plano z.

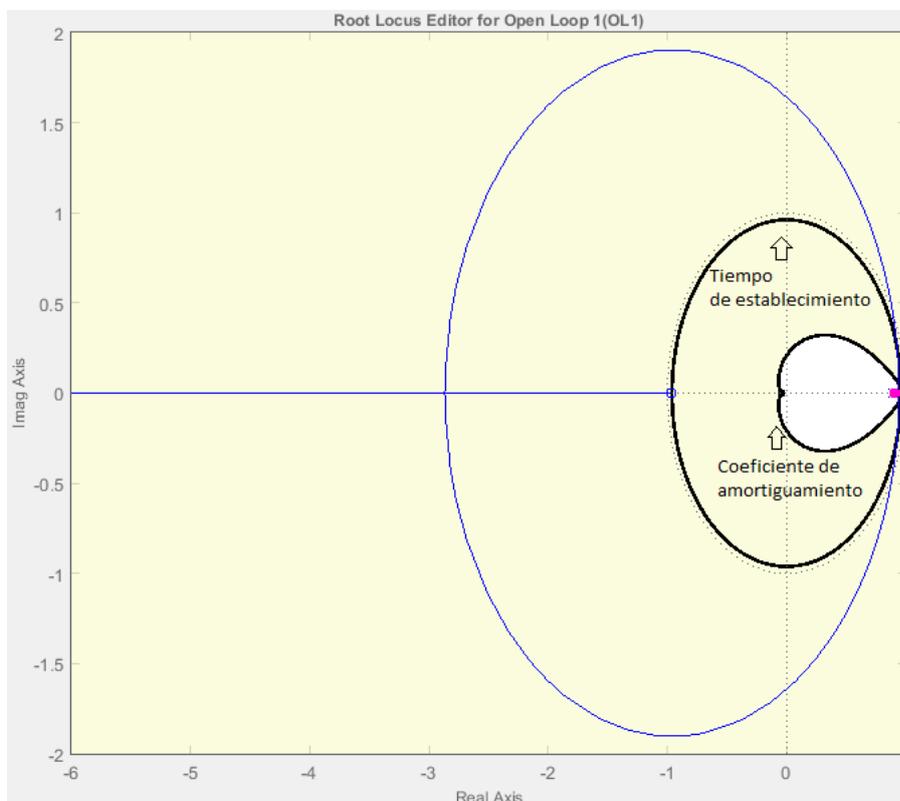


Figura 12. Lugar de raíces de un sistema en el plano z.

2.5 FPGA

Para realizar la programación del algoritmo de control del convertidor *buck* y para obtener los datos de la tensión de la salida de este digitalmente, se ha utilizado una FPGA. Puede definirse FPGA (*Field-Programmable Gate Array*) como un circuito integrado diseñado para ser configurado por el propio cliente después de ser fabricado.

Estos sistemas cuentan con una jerarquía de interconexiones programables que permite a los bloques lógicos ser interconectados según las necesidades del programador. Esto hace que la FPGA pueda desempeñar cualquier tipo de función lógica necesaria. La programación de estos dispositivos suele realizarse mediante lenguajes de descripción de hardware.

En este trabajo en particular se ha utilizado la FPGA SPARTAN-3E de Xilinx. Este dispositivo es el que puede verse en la figura 13, extraída del manual del fabricante [7].

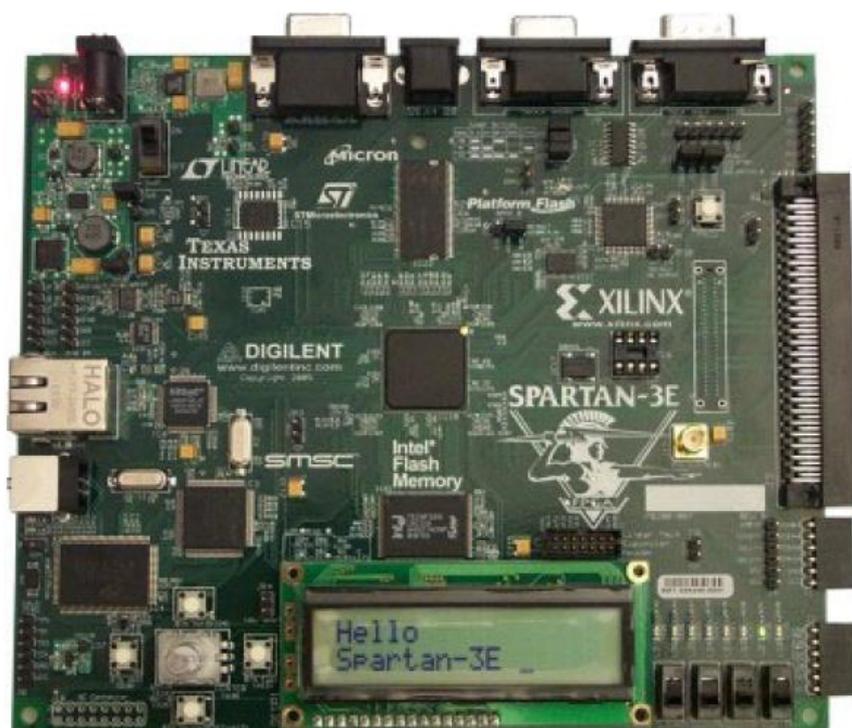


Figura 13. FPGA Sparta-3E.

Las características principales con las que cuenta esta placa son las siguientes:

- Hasta 232 pines de E/S.
- 4 interruptores tipo slide.
- 4 interruptores tipo pulsador.
- Reloj oscilador de 50 MHz.
- 2 entradas ADC con ganancia programable.
- 8 LEDs.
- Pantalla LCD de 2 líneas x 16 caracteres.
- 4 salidas DAC.
- 4 Mbits de memoria PROM *Platform Flash* de Xilinx.
- 64 MBytes (512 Mbits) de memoria DDR SDRAM.

- 16 MBytes (128 Mbits) de memoria *Parallel NOR Flash* (Intel StrataFlash).
- 16 Mbits de SPI serial Flash (STMicro).

El lenguaje de programación que utiliza la FPGA y se ha utilizado en este trabajo es VHDL. A continuación se exponen algunas nociones básicas de este lenguaje.

En primer lugar hay que tener claro que este lenguaje estructura cualquier sistema digital como un modelo. Para definir sus entradas y salidas utiliza el concepto de entidad, mientras que para describir su comportamiento interno usa el de arquitectura. En la figura 14 puede verse un esquema de este modelo VHDL.

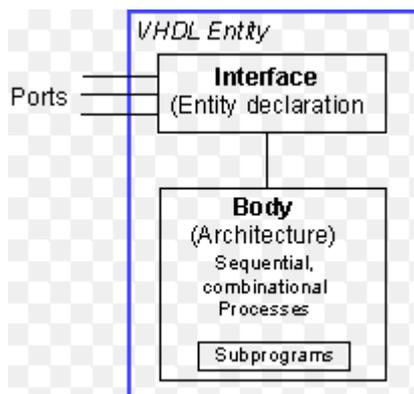


Figura 14. Esquema del modelo VHDL.

Así, un fichero VHDL debe seguir siempre una misma estructura que es la siguiente:

- Bibliotecas: En primer lugar se debe indicar cuales son las bibliotecas que se requieren utilizar dependiendo de las operaciones que se deseen realizar en nuestro sistema.
- Entidad: Aquí se indican cuáles son las entradas y salidas del sistema digital en cuestión y el tipo de estas (booleanas, enteros, vectores, etc.).
- Arquitectura: En este apartado se definirán todas las variables a implementar en el sistema y las interacciones de estas con las entradas y salidas, correspondientes al comportamiento deseado.

En la figura 15 puede verse un ejemplo de un fichero VHDL.

```
-----  
-- Model of a simple D Flip-Flop --  
-----  
  
-- library declaration  
library IEEE;  
use IEEE.std_logic_1164.all;  
  
-- entity  
entity d_ff is  
  port (  
    D, CLK : in std_logic;  
    Q       : out std_logic);  
end d_ff;  
  
-- architecture  
architecture my_d_ff of d_ff is  
begin  
  
  -- process  
  dff: process (CLK)  
  begin  
    if rising_edge(CLK) then  
      Q <= D;  
    end if;  
  end process dff;  
  
end my_d_ff;
```

Figura 15. Ejemplo de un fichero VHDL, el cual emula el comportamiento de un *flip-flop* tipo D.

3 Acondicionamiento de señal

El primer problema que se debe abordar se produce debido a que el convertidor analógico-digital integrado en la FPGA que se va a utilizar es demasiado lento para las elevadas frecuencias a las que se desea trabajar. Por eso, se ha decidido utilizar el convertidor de Maxim MAX12555 que trabaja a una frecuencia de muestreo de 95 GHz. Así, el primer paso de este trabajo será realizar el diseño y la implementación de un buffer de amplificadores operacionales que nos permita acoplar el convertidor *buck* con el conversor analógico digital.

3.1 Buffer de amplificadores operacionales

En la mayoría de los casos los conversores analógico-digital (ADCs) requieren de un buffer de amplificadores. En el caso que ocupa este proyecto concretamente, el kit de evaluación en el que viene integrado el convertidor que se quiere utilizar, presenta un buffer con entrada diferencial y que pasa por dos transformadores que eliminan la tensión continua. Como la información que nos proporciona la tensión continua es imprescindible para realizar el control del *buck* y la salida de este está referida a tierra, se debe implantar un buffer que acople la continua y que pase de una entrada referida a tierra a una diferencial, para así poder atacar directamente al convertidor.

Para esto, el modelo de convertidor analógico-digital que se utilizará es el que obedece a la figura 16. Donde IN+ i IN- son las entradas del convertidor positiva y negativa respectivamente, y CM es el valor de tensión correspondiente a la mitad del fondo de escala.

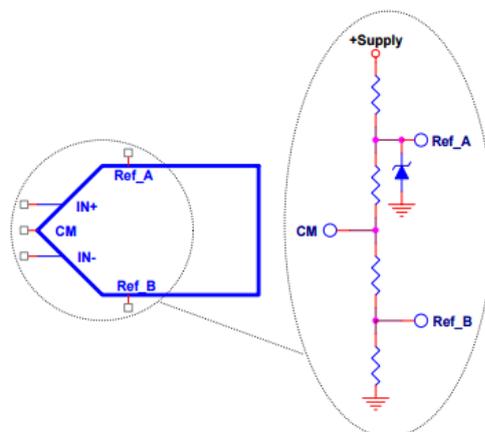


Figura 16. Estructura de referencia común de los ADCs.

Así, se ha decidido implementar el esquema que puede verse en la figura 17, inspirado en lo documentado en [8].

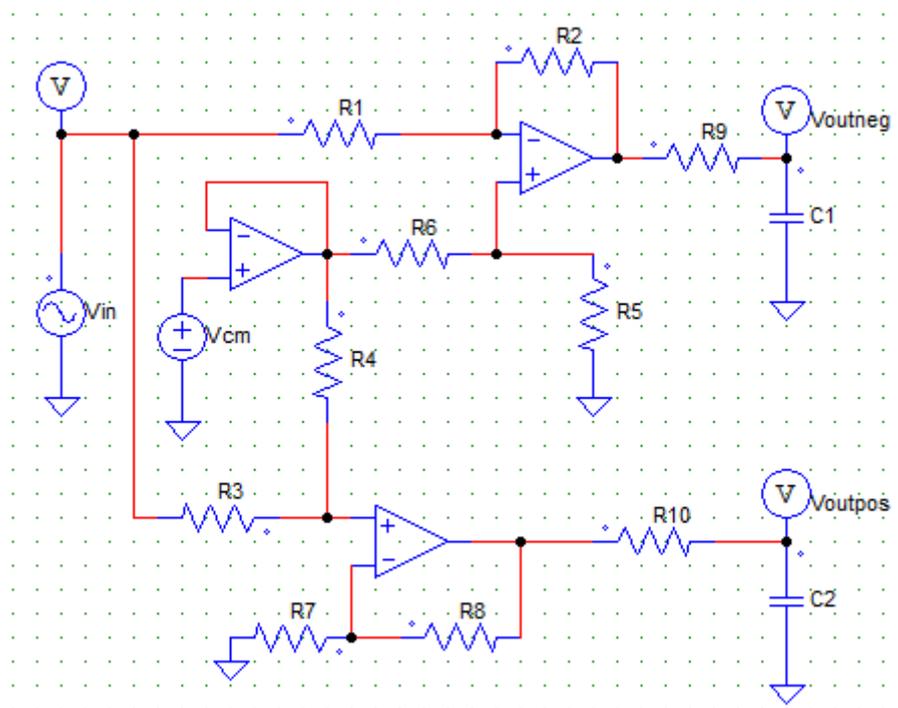


Figura 17. Buffer seleccionado para su implementación.

3.1.1 Análisis teórico

A continuación se analizará cuidadosamente el buffer propuesto y se comprobará que se comporta de la manera deseada. Para esto, puede suponerse el valor de tensión dado por CM como una fuente ideal de continua y se dividirá el circuito en dos partes: la que proporciona la entrada positiva (IN+) y la que determina la negativa (IN-). A demás, de momento se dejará de lado las resistencias R9, R10 y los condensadores C1 y C2, que forman un filtro pasa-bajas el cual se explicará en el apartado posterior.

De esta forma, analizando únicamente la circuitería correspondiente a la entrada negativa del convertidor, se obtiene el sistema que puede verse en la figura 18.

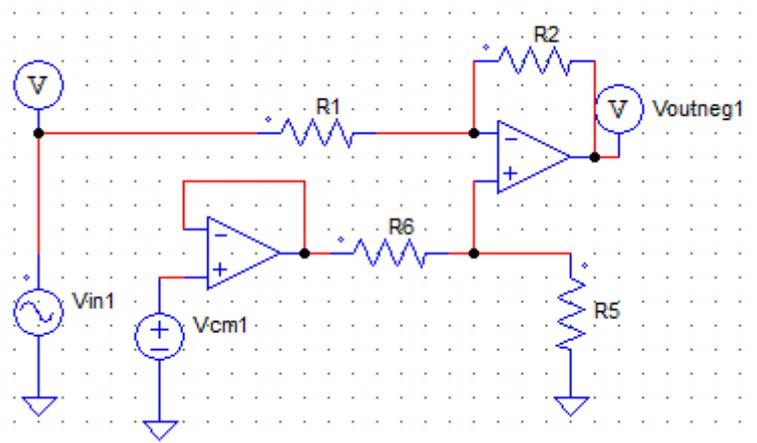


Figura 18. Circuitería correspondiente a la salida negativa del buffer.

Aplicando el principio de superposición, se obtienen los siguientes resultados:

- Si $V_{cm} = 0$:

$$V_{o_{IN-}(1)} = -\frac{R_2}{R_1} \cdot V_{in} \quad (20)$$

- Si $V_{in} = 0$:

$$V_{o_{IN-}(2)} = \frac{R_5}{R_5 + R_6} \cdot \frac{R_1 + R_2}{R_1} \cdot V_{cm} \quad (21)$$

$$\begin{aligned} V_{o_{IN-}} &= V_{o_{IN-}(1)} + V_{o_{IN-}(2)} \\ &= -\frac{R_2}{R_1} \cdot V_{in} + \frac{R_5}{R_5 + R_6} \cdot \frac{R_1 + R_2}{R_1} \cdot V_{cm} \end{aligned} \quad (22)$$

Si $R_1 = 1 \text{ K}\Omega$; $R_2 = 500 \text{ }\Omega$; $R_5 = 1 \text{ k}\Omega$; $R_6 = 500 \text{ }\Omega$:

$$V_{o_{IN-}} = -\frac{1}{2} \cdot V_{in} + \frac{2}{3} \cdot \frac{3}{2} \cdot V_{cm} \quad (23)$$

$$V_{o_{IN-}} = -\frac{1}{2} \cdot V_{in} + V_{cm} \quad (24)$$

En cuanto a la parte correspondiente a la entrada positiva, es la que puede verse en la figura 19.

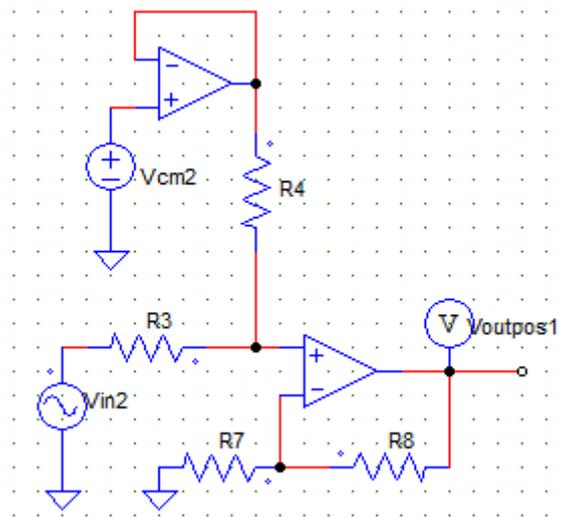


Figura 19. Circuitería correspondiente a la salida positiva del buffer.

De igual forma que en el caso anterior, se aplica el principio de superposición, obteniendo:

- Si $V_{cm} = 0$:

$$V_{O_{IN+}}(1) = \frac{R_4}{R_3 + R_4} \cdot \frac{R_7 + R_8}{R_7} \cdot V_{in} \quad (25)$$

- Si $V_{in} = 0$:

$$V_{O_{IN+}}(2) = \frac{R_3}{R_3 + R_4} \cdot \frac{R_7 + R_8}{R_7} \cdot V_{cm} \quad (26)$$

$$\begin{aligned} V_{O_{IN+}} &= V_{O_{IN+}}(1) + V_{O_{IN+}}(2) \\ &= \frac{R_4}{R_3 + R_4} \cdot \frac{R_7 + R_8}{R_7} \cdot V_{in} + \frac{R_3}{R_3 + R_4} \cdot \frac{R_7 + R_8}{R_7} \cdot V_{cm} \end{aligned} \quad (27)$$

Si $R_3 = 1 \text{ k}\Omega$; $R_4 = 500 \text{ }\Omega$; $R_7 = 1 \text{ k}\Omega$; $R_8 = 500 \text{ }\Omega$:

$$V_{O_{IN+}} = \frac{1}{3} \cdot \frac{3}{2} \cdot V_{in} + \frac{2}{3} \cdot \frac{3}{2} \cdot V_{cm} \quad (28)$$

$$V_{O_{IN+}} = \frac{1}{2} \cdot V_{in} + V_{cm} \quad (29)$$

Así, una vez se ha obtenido el valor de cada una de las entradas, puede calcularse el valor de la entrada diferencial:

$$V_{O_{dif}} = V_{O_{IN+}} - V_{O_{IN-}} = \frac{1}{2} \cdot V_{in} + V_{cm} + \frac{1}{2} \cdot V_{in} - V_{cm} \quad (30)$$

$$V_{O_{dif}} = V_{in} \quad (31)$$

Por lo tanto puede verse como se ha conseguido mantener el valor de la entrada, pasando de unipolar a diferencial y también manteniendo la continua, ya que no se tiene ninguna etapa de transformadores. Una vez visto esto, se procederá a analizar los filtros pasa-bajas compuestos por las resistencias R9 y R10, y los condensadores C1 y C2.

3.1.1.1 Filtro antialiasing

Una vez diseñado el buffer, debe tenerse en cuenta cual es la frecuencia de muestro del conversor para así evitar que se produzca *aliasing*. Según el teorema de Nyquist, que puede verse en la ecuación 32, la frecuencia de muestreo debe ser mayor que el doble de la frecuencia de la muestra.

$$f_s > 2f_m \quad (32)$$

Siendo f_s la frecuencia de muestreo y f_m la frecuencia de la muestra. Así, debe asegurarse que el doble de la frecuencia de las muestras esté siempre por debajo de la frecuencia de muestreo. Con este propósito, instalaremos un filtro pasa-bajas en las dos salidas del buffer formados por las resistencias y condensadores R9, C1 y R10 y C2 respectivamente.

En el caso que ocupa el proyecto, el ADC muestrea a una frecuencia de 95GHz, por lo que siguiendo el teorema de Nyquist la frecuencia de la muestra debería ser inferior a:

$$95 > 2fm \quad (33)$$

$$fm < 47.5 \text{ GHz} \quad (34)$$

Teniendo esto en cuenta, se han determinado los valores de las resistencias y condensadores que formarán el filtro tal como puede verse seguidamente:

$$R_9 = R_{10} = 24.9\Omega \quad (35)$$

$$C_1 = C_2 = 15pF \quad (36)$$

A continuación se analiza el comportamiento de este para asegurar que trabajará de la forma deseada, en la figura 20 puede verse el esquema de un filtro pasa-bajas como los que implementaremos en sendas salidas del buffer.

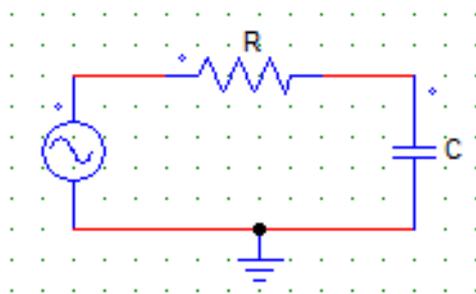


Figura 20. Filtro pasa-bajas pasivo.

Para comprender el funcionamiento de este sistema, obtendremos su función de transferencia, que obedecerá la siguiente expresión:

$$\frac{V_o(s)}{V_i(s)} = \frac{\frac{1}{Cs}}{\frac{1}{Cs} + R} = \frac{1}{1 + RCs} = \frac{\frac{1}{RC}}{\frac{1}{RC} + s} \quad (37)$$

Una vez sabemos cuál es la función de transferencia a la que obedece nuestro sistema, podemos elaborar un diagrama bode a través de Matlab, obteniendo los resultados que pueden verse en la figura 21.

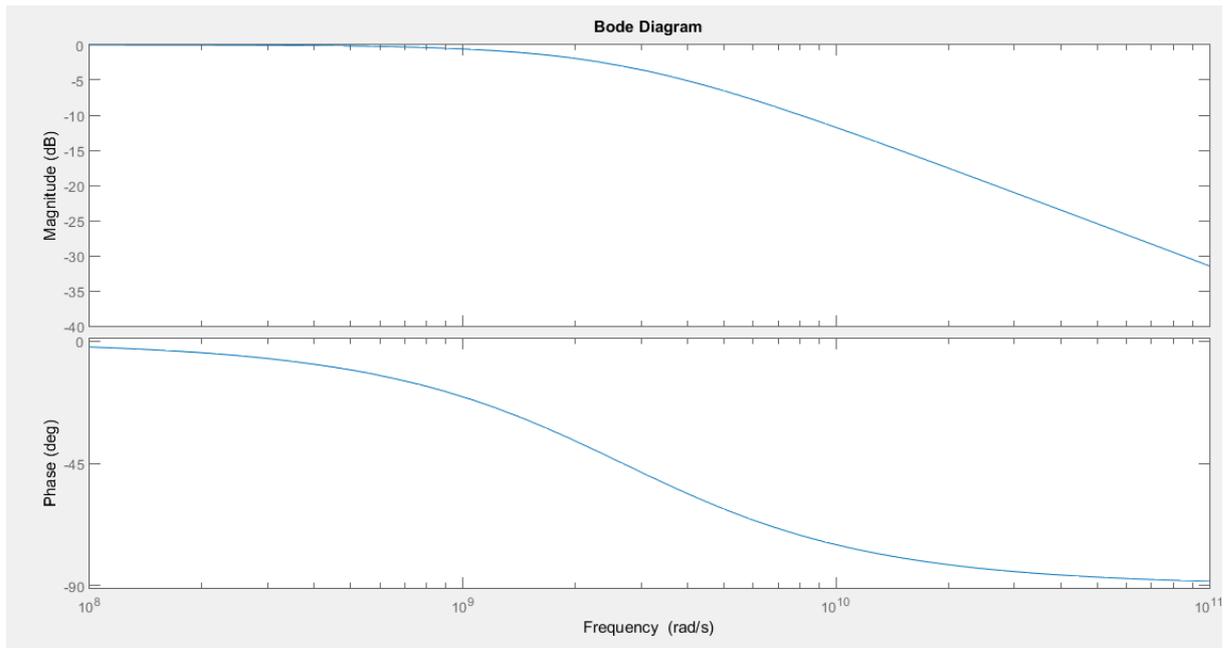


Figura 21. Diagrama bode del filtro pasa-bajas.

Así, puede verse como el sistema tiene un polo en $-1/RC$, por lo que la frecuencia de corte será la siguiente:

$$\omega_o = \frac{1}{RC} = 2,68 \cdot 10^9 \text{ rad/s} \quad (38)$$

$$f_m = f_o = \frac{\omega_o}{2\pi} = 426 \text{ MHz} \quad (39)$$

$$f_m = 426 \text{ MHz} < 47.5 \text{ GHz} = \frac{f_s}{2} \quad (40)$$

De esta manera, puede observarse como la frecuencia de la muestra siempre será muy inferior a la mitad de la frecuencia de muestreo, con lo cual se evitará que se produzca *aliasing* en cualquier caso.

3.1.2 Simulación PSIM

Una vez se ha comprobado analíticamente que el buffer se comporta de la manera deseada, puede realizarse la simulación del sistema PSIM para verificar los resultados obtenidos.

Así, utilizando el esquema de PSIM que se ha visto en la figura 17 se han graficado las salidas del sistema, introduciendo en la entrada una onda sinusoidal de 1,3V de offset (ya que esta será la tensión de referencia con la que se desea que trabaje el convertidor *buck*) y una frecuencia de 50 Hz.

De esta manera, se obtienen los resultados que pueden verse en las figuras 22 y 23.

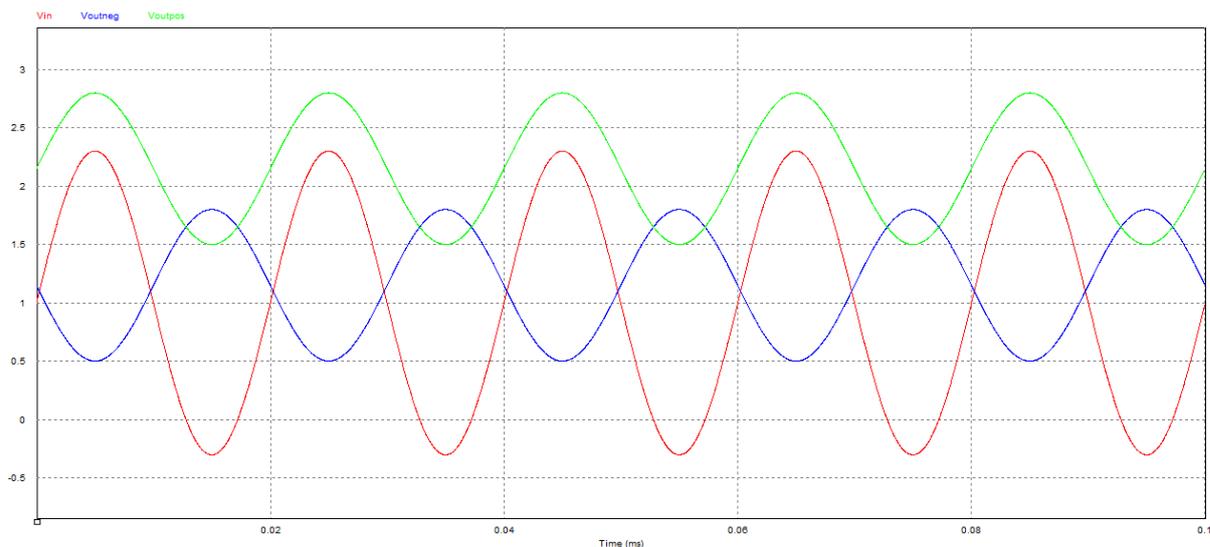


Figura 22. Simulación PSIM del buffer (1).

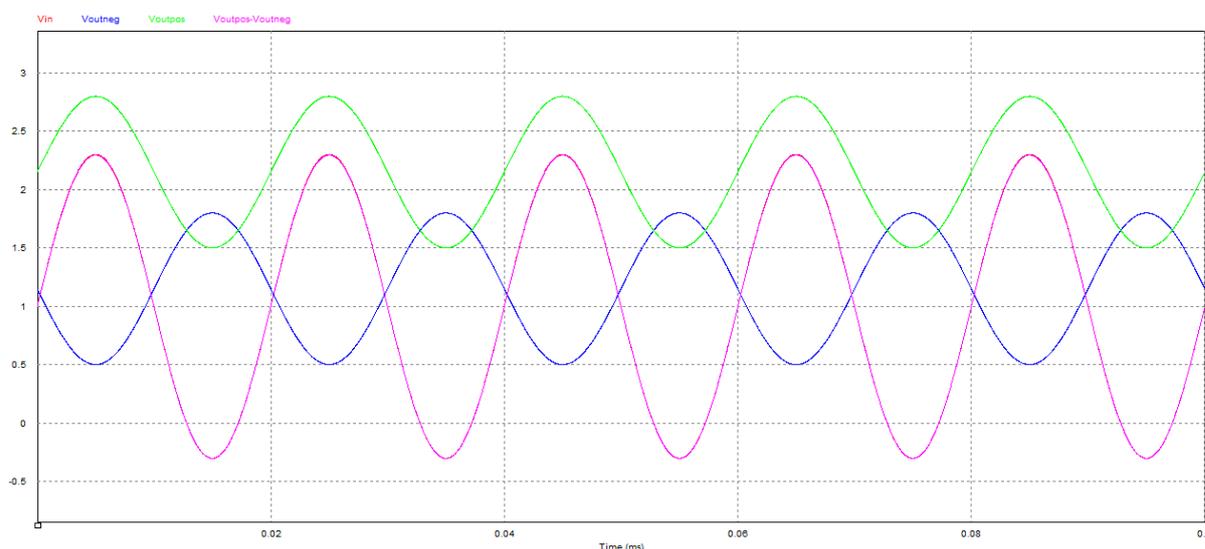


Figura 23. Simulación PSIM del buffer (2).

En las figuras 22 y 23 puede verse como la señal diferencial de las salidas, coincide con la entrada, manteniendo la continua, por lo tanto puede determinarse que el buffer opera de la forma requerida.

Por último, para comprobar el funcionamiento del filtro pasa-bajas, se ha realizado una simulación aumentando la frecuencia de la entrada a 50 GHz. El resultado de esta puede verse en la figura 24.

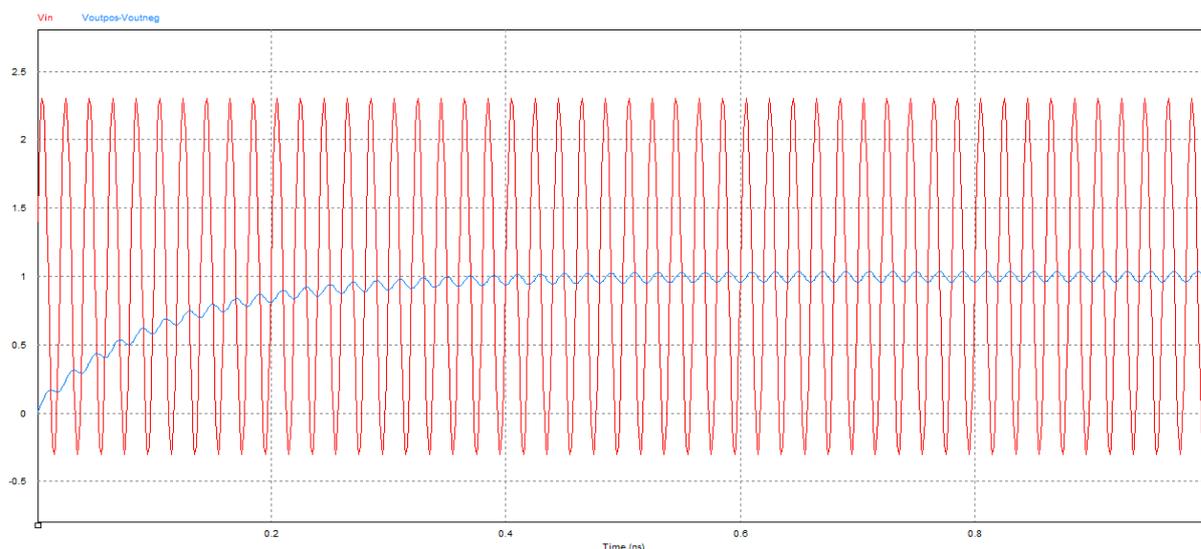


Figura 24. Simulación PSIM del buffer a alta frecuencia.

Puede verse como el filtro elimina la contribución de las altas frecuencias, quedándose prácticamente solo con la tensión en continua. De esta manera puede concluirse que el buffer se comporta de la manera deseada.

3.1.3 Implementación de la PCB

3.1.3.1 Diseño de la PCB

Una vez se ha simulado y comprobado que el buffer se comporta de la forma requerida puede pasarse a diseñar la PCB en la que irá integrado. Para esto se ha utilizado el software *Design Spark*.

En primer lugar, se han elegido los componentes que formarán parte de nuestro buffer. Como hemos visto en los apartados anteriores, el buffer estará integrado por cuatro resistencias de $500\ \Omega$, cuatro más de $1\ \text{k}\Omega$, dos de $24,9\ \Omega$ y dos condensadores de $15\ \text{pF}$. Como puede verse en la figura 25, el kit de evaluación del ADC que se utilizará [9] cuenta con unos condensadores de $15\ \text{pF}$ que cuelgan de las entradas diferenciales que serán atacadas (que ya han sido colocados pensando en formar un filtro *antialiasing*), por lo tanto no se incluirán estos condensadores en la PCB.

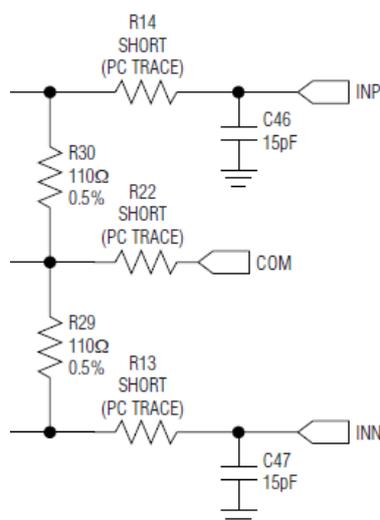


Figura 25. Distribución de las entradas y de la salida COM del ADC en el kit de evaluación.

Por otra parte, sí debe incluirse un conector SMA hembra, mediante el cual se introducirá la señal de entrada del sistema, además de 2 *pin-headers* para la conexión de tierra y alimentación. En la tabla 1 pueden verse enumerados los diferentes componentes que se han decidido utilizar para el buffer.

Tabla 1. Componentes seleccionados para implementar el buffer.

Componente	Código RS	Fabricante	Referencia fabricante	Valor	Dimensiones (mm)	Tolerancia	Precio (€)	Unidades	Precio total (€)
Resistencia 499Ω	679-2171	Vishay	CRCW1206499RFKEA	499 Ω	3.2 x 1.6 x 0.55	±1	0,031	4	0,124
Resistencia 24,9Ω	679-1945	Vishay	CRCW120624R9FKEA	24,9Ω	3.2 x 1.6 x 0.55	±1	0,032	2	0,064
Conector SMA	616-3422	TE Conect.	5-1814832-1	-	-	-	2,02	1	2,02
Resistencia 1kΩ	679-1844	Vishay	CRCW12061K00FKEA	1 kΩ	3.2 x 1.6 x 0.55	±1	0,035	4	0,14
ADA4891-2	-	A. Devices	-	-	SOIC8	-	-	2	-
Pin-headers	-	-	-	-	-	-	-	2	-

Se han elegido resistencia de montaje superficial, para poder soldarlas manualmente en la PCB. De igual forma, se ha considerado que tengan unas dimensiones adecuadas para poder soldarlas sin mucha dificultad, pero también teniendo en cuenta que no deben ocupar mucho espacio en la placa.

En los componentes que se han enumerado en la tabla 1 vemos que se ha incorporado el ADA4891-2 de *Analog Devices*. Este dispositivo cuenta con dos amplificadores operacionales, tal y como puede verse en la figura 26, extraída del *data sheet* de este dispositivo, [10]. Como se requieren tres, se instalarán dos ADA4891-2 y uno de los operacionales quedará sin ser utilizado.

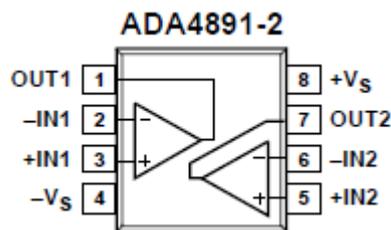


Figura 26. Esquema del ADA4891-2.

Una vez se conocen los componentes que se van a integrar en la PCB, puede pasarse a diseñarla. En primer lugar, gracias al código RS y la referencia del fabricante, se ha exportado el símbolo de cada uno de los elementos de la librería de *Design Spark* y construido el *schematic* del circuito. En la figura 27 puede verse el diseño de este.

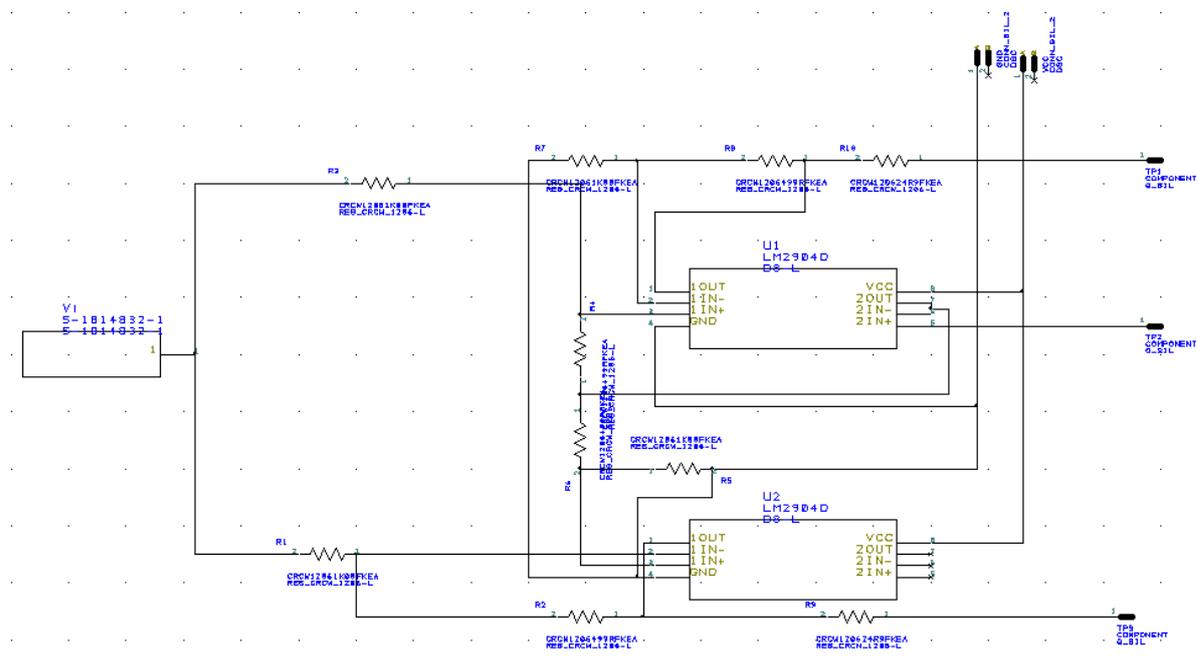


Figura 27. Esquema del diseño del buffer.

Como puede verse en la figura 27, uno de los operacionales del segundo ADA4891-2 no se utiliza, por lo tanto se han dejado sus pines de entradas y salida desconectados.

A continuación, puede pasarse a diseñar el *layout* de la PCB. Para realizar este diseño, se ha tenido en cuenta que, al trabajar a altas frecuencias, la longitud del cable mediante el cual se conecten las salidas del buffer con las entradas del ADC debe ser mínima. Para esto se ha decidido implementar una placa de las mismas dimensiones que la que contiene el kit de evaluación del ADC, dejando una serie de huecos a través de los cuales puedan pasar los diferentes elementos del kit de evaluación que sobresalen de la placa (conectores SMA, *jumpers*, *pin-headers*, etc.). De esta manera, se podrá acoplar la PCB encima del kit de evaluación, quedando así las entradas y salidas muy próximas entre sí y minimizando la longitud del cable que las unirá. En la figura 28 puede verse este *layout* de la PCB.

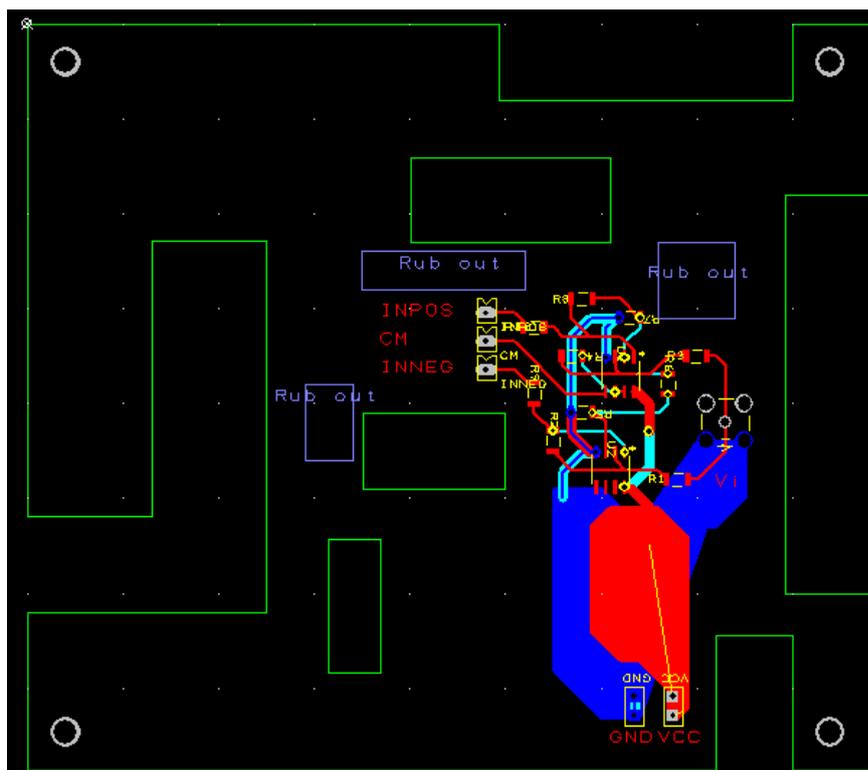


Figura 28. *Layout* de los elementos del buffer en la PCB (1).

Además de los huecos dejados, se ha indicado también varias zonas donde deberá realizarse un *rub out* del cobre, para asegurar que, en el momento de montar la placa sobre el kit de evaluación, no se produce contacto con ninguna conexión del kit. Los pines INPOS, CM y INNEG, se han colocado justo encima de donde se realizará su conexión con el ADC, con la finalidad de minimizar la longitud del cable, como se ha comentado anteriormente.

Al margen de estas consideraciones, también se ha tratado de minimizar la longitud de las pistas para minimizar las pérdidas a alta frecuencias, así como también evitar la formación de ángulos de 90°, que producirían interferencias.

Como puede observarse, se ha dedicado una gran superficie a GND y VCC, para que el valor de estos sean lo más estable posible. Las pistas correspondientes a estos valores también se han realizado de un grueso superior al de la señal. En la figura 29 puede verse de manera más detalla el diseño de las pistas correspondientes a la señal.

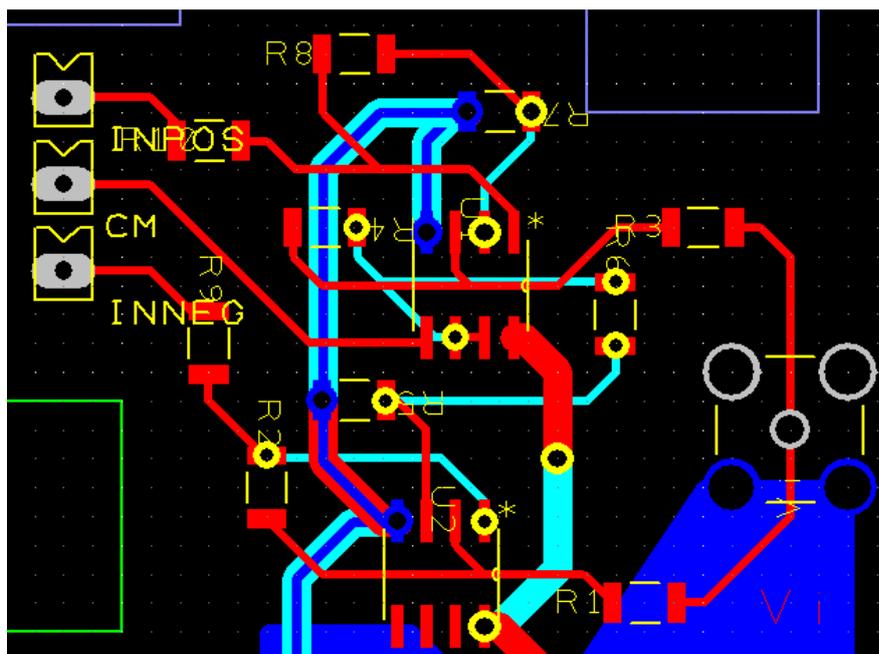


Figura 29. *Layout* de los elementos del buffer en la PCB (2).

El diseño de las pista por las que se transmitirá la señal se ha realizado tratando de conservar una cierta simetría entre las pista correspondientes a las salidas positiva y negativa así como también procurando que la longitud de cada uno de los circuitos sea similar, con la finalidad de equilibrar las pérdidas, no idealidades y desfases de las dos señales.

Con el objetivo de reducir la longitud de las pistas, se ha ido variando la utilización de la capa superior y de la capa inferior.

3.1.3.2 Implementación física de la PCB y resultados obtenidos

Una vez se ha realizado el diseño de la PCB tal y como se ha explicado en el apartado anterior, se ha mandado a fabricar al técnico de laboratorio. La fabricación de la placa se ha realizado extrayendo el cobre sobrante de una placa de este. Finalmente, se ha obtenido la PCB que puede verse en la figura 30.

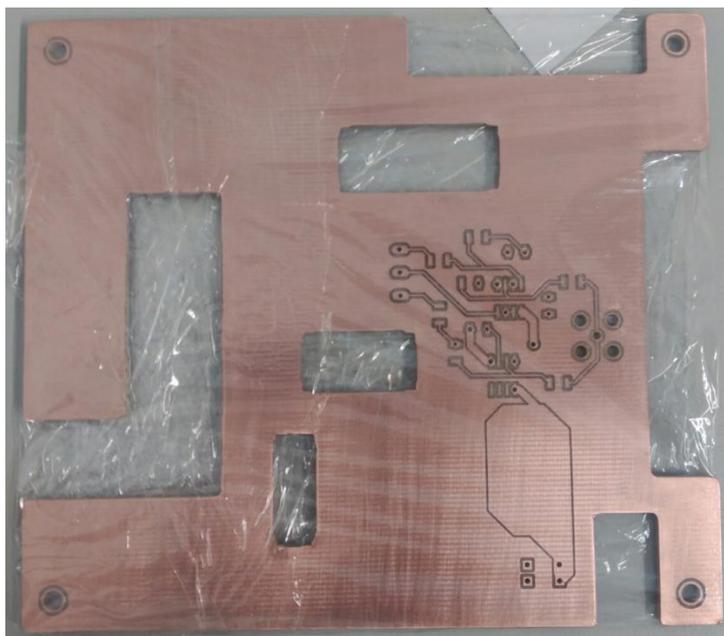


Figura 30. PCB del buffer implementado.

Para continuar, deben soldarse cada uno de los diferentes componentes que integran el buffer en sus respectivos *pads*. Estas soldaduras se han realizado principalmente con estaño y un soldador convencional, salvo en el caso de los ADA4891-2, para los que se ha utilizado una pasta para soldar de SMA junto a una pistola de calor, ya que las patas de cada uno de los pines de este elemento eran mucho más pequeña que las resistencias y con la pistola de calor y la pasta obtenemos una precisión mayor. Así, en la figura 31 puede verse la PCB con todos sus componentes soldados.

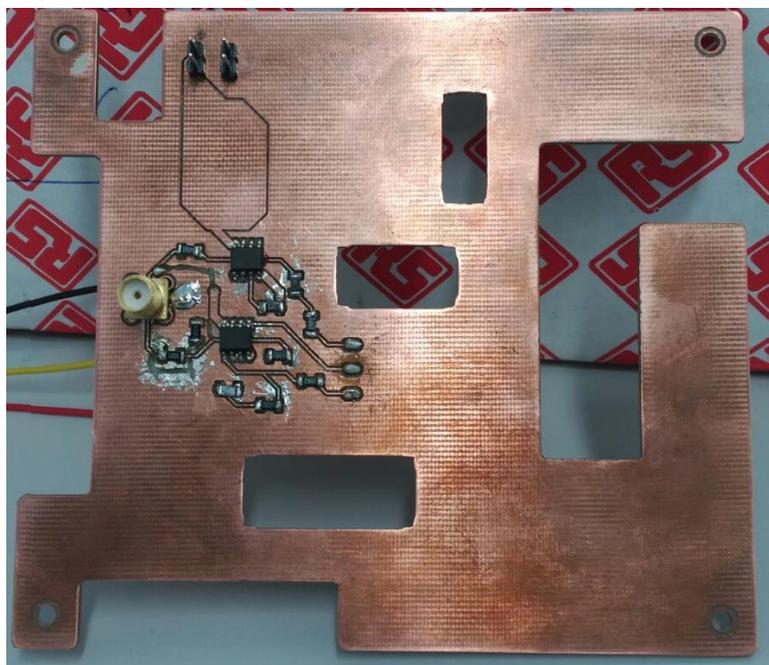


Figura 31. PCB del buffer implementado con los componentes soldados.

Para continuar, se ha pasado a comprobar que la placa funciona de la manera deseada. Para ello se ha introducido una señal sinusoidal de 500 mV de amplitud, 100 KHz de frecuencia y 100 mV de offset. Para simular el valor de tensión proporcionado por CM, se

ha utilizado una fuente de tensión continua a 1,65 V. En la figura 32 pueden verse los resultados obtenidos.

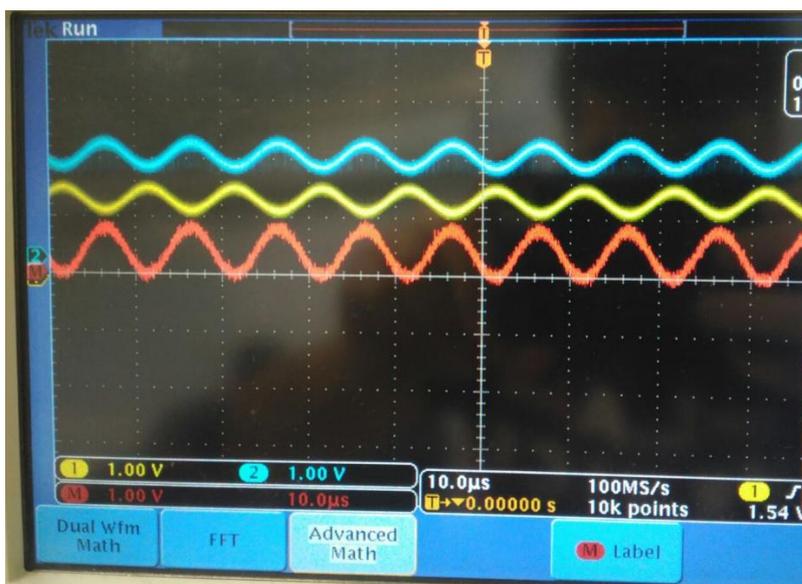


Figura 32. Resultados obtenidos en la salida del buffer al introducir una señal de entrada de 500 mV de amplitud, 100 KHz y 100 mV de offset.

En la figura anterior podemos ver las señales correspondientes a las salidas positiva (azul) y negativa (amarillo) y la resta de estas (rojo). Puede verse como el valor de la resta de las salidas es igual que la entrada del buffer, salvo por la aparición de un offset debido a la resistencia de salida del generador, como se verá más adelante, este problema será solucionado incluyendo un seguidor de tensión. Por lo tanto, sin tener cuenta este efecto, puede determinarse que el buffer funciona de la manera deseada.

Al ir aumentando en frecuencia, se ha podido observar que el buffer funciona hasta una frecuencia de 10 MHz, en la que ya empieza a filtrar toda la señal. Hay que comentar que este suceso puede deberse también a las inductancias y capacidades parásitas de los equipos de medida (tanto sondas como osciloscopios) ya que probablemente no estén optimizados para medir señales a tan altas frecuencias.

3.1.4 Integración Buffer-ADC

Una vez se ha obtenido la PCB correspondiente al buffer de amplificadores operacionales y se ha comprobado que este funciona de la manera deseada, puede pasarse a acoplarlo junto al kit de evaluación del conversor analógico-digital. Para ello, en primer lugar se soldarán los cables en cuestión a los *pads* correspondientes del kit de evaluación, después se colocará el buffer encima y se soldarán los cables a este, de esta forma reduciremos la longitud de los cables.

Para comenzar se debe tener claro dónde deben soldarse los cables para atacar al ADC. En la figura 33 puede verse un esquema del kit de evaluación del ADC, obtenido de [9]. En este puede observarse como a través de las resistencias R22, R13 y R14 se puede atacar a COM, INN e INP respectivamente. Estas resistencias no están colocadas en el kit, en lugar de ellas, se ha introducido un corto circuito. Así, haciendo un pequeño corte en esta pista, se podrá desacoplar el resto del acondicionamiento de señal del ADC y conectarse a él directamente.

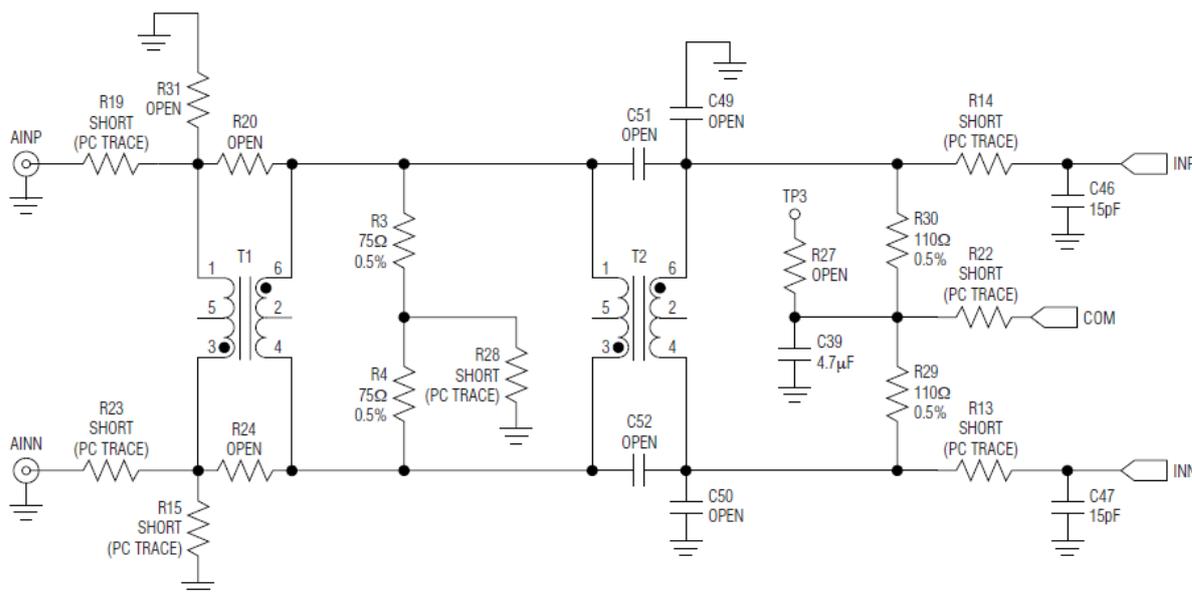


Figura 33. Esquema de la distribución de las entradas INP y INN y de la salida COM del ADC en el kit de evaluación.

3.1.4.1 Implementación física del conjunto

Para realizar esta operación, se ha utilizado una lente de aumento para poder trabajar en superficies tan pequeñas, tal y como puede verse en la figura 34.



Figura 34. Lente de aumento Mantis utilizada para la soldadura sobre el kit de evaluación del ADC.

Utilizando la lente de aumento, la pasta para soldar SMA y una pistola de calor, se ha procedido a soldar tres cables en COM, INP e INN. En el caso de COM, la pista que debía atacarse se encontraba en la cara de debajo de la placa, por lo que se ha tenido que pasar el cable por un hueco una vez ha sido soldado, en la figura 35 podemos ver esta soldadura.

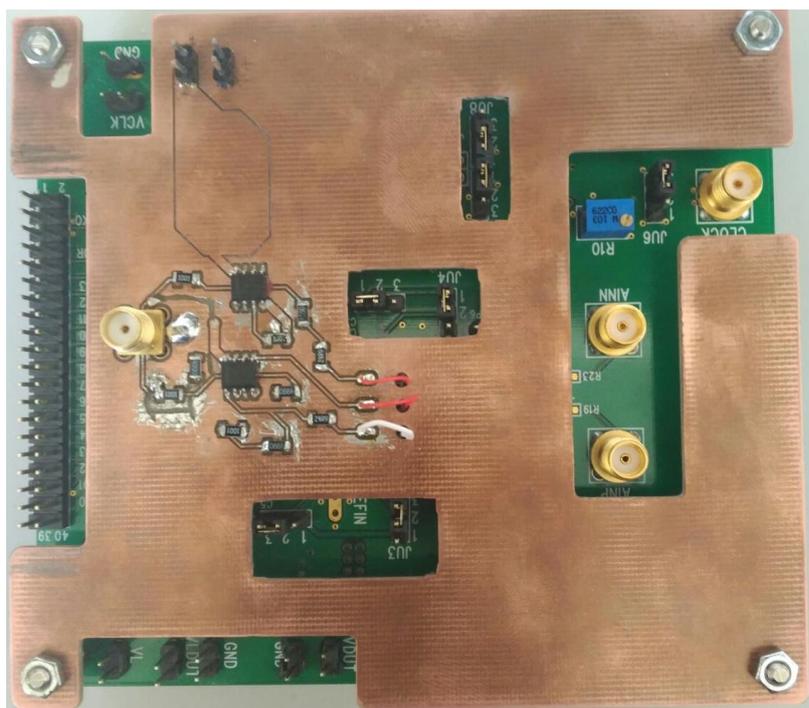


Figura 37. Conjunto formado por el buffer implementado y el kit de evaluación del ADC.

3.2 Divisor de tensión y seguidor

Para continuar con la implementación del acondicionamiento de señal necesario, es necesario tener en cuenta que el convertidor *buck* sobre el que se va a trabajar está preparado para ser alimentado a 5 V. Por lo tanto, el rango de tensiones con el que se va a trabajar es de 0 – 5 V, dependiendo del valor del ciclo de trabajo en cada caso. Como puede verse en la figura 38 (extraída del *data sheet* [11]), el conversor analógico digital del que se dispone opera dentro de un rango de $\pm 1,024$ V. Por este motivo es necesario implementar un divisor de tensión que nos reduzca la tensión una quinta parte, pasando así de un rango de 0 – 5 V a 0 – 1 V.

ANALOG INPUT (INP, INN)				
Differential Input Voltage Range	V _{DIFF}	Differential or single-ended inputs	± 1.024	V

Figura 38. Rango de valores de entrada del ADC.

Al introducir este divisor de tensión, la resistencia que verá la entrada del buffer aumentará lo que hará que no opera de la forma deseada. Este es el efecto que producía el offset que se ha visto al realizar las medidas con el osciloscopio en el apartado anterior. Por lo tanto, para evitar la aparición de este offset, será necesaria la incorporación de un seguidor de tensión para aislar el buffer.

3.2.1 Análisis teórico

Para continuar se analizará cuidadosamente tanto el divisor de tensión como el seguidor a implementar. En primer lugar, debe tenerse en cuenta que el convertidor *buck* está diseñado de tal manera que puede tener un resistencia de carga de $0,5 \Omega$ o 50Ω . Por este motivo, debe implementarse el divisor de tensión con dos resistencias mucho mayores que esta. Tal como puede verse en la figura 39, se ha decidido utilizar resistencias de $20 \text{ k}\Omega$ y $5 \text{ k}\Omega$.

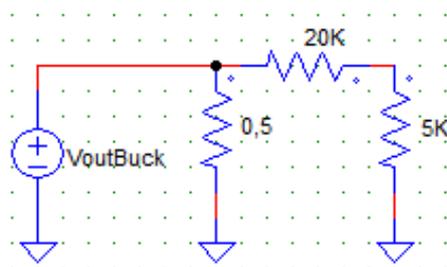


Figura 39. Salida del convertidor *buck* con un divisor de tensión a continuación.

Al utilizar resistencias cuyos valores son tan alto en comparación a la resistencia de carga, el efecto de esta puede ser menospreciado. De esta manera se obtiene la tensión de salida que puede verse a continuación:

$$V_{out} = V_{outBuck} \cdot \frac{5 \text{ K}\Omega}{25 \text{ K}\Omega} = \frac{V_{outBuck}}{5} \quad (41)$$

Así, puede observarse como se consigue reducir la tensión de salida una quinta parte. Como se ha comentado anteriormente, este divisor aumenta la resistencia de salida del convertidor *buck*, y por lo tanto la que ve el buffer. Así sería necesaria la implantación de un seguidor de tensión a continuación del divisor.

3.2.2 Simulaciones PSIM

Para justificar la implantación del seguidor de tensión, en primer lugar se ha realizado una simulación del divisor de tensión seguido del buffer para ver cómo afectaría la introducción del primero a los resultados. En la figura 40 podemos ver un esquema del conjunto del sistema.

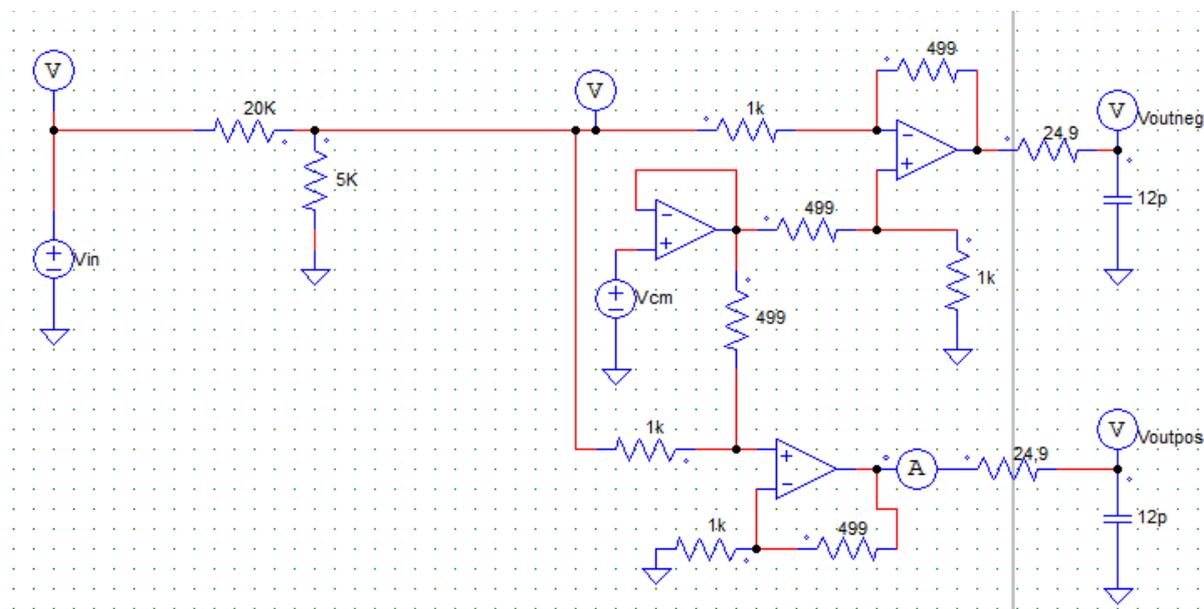


Figura 40. Esquema del divisor de tensión y el buffer.

Al introducir una tensión de 2 V en la entrada, se obtienen los resultados que pueden verse en la figura 41

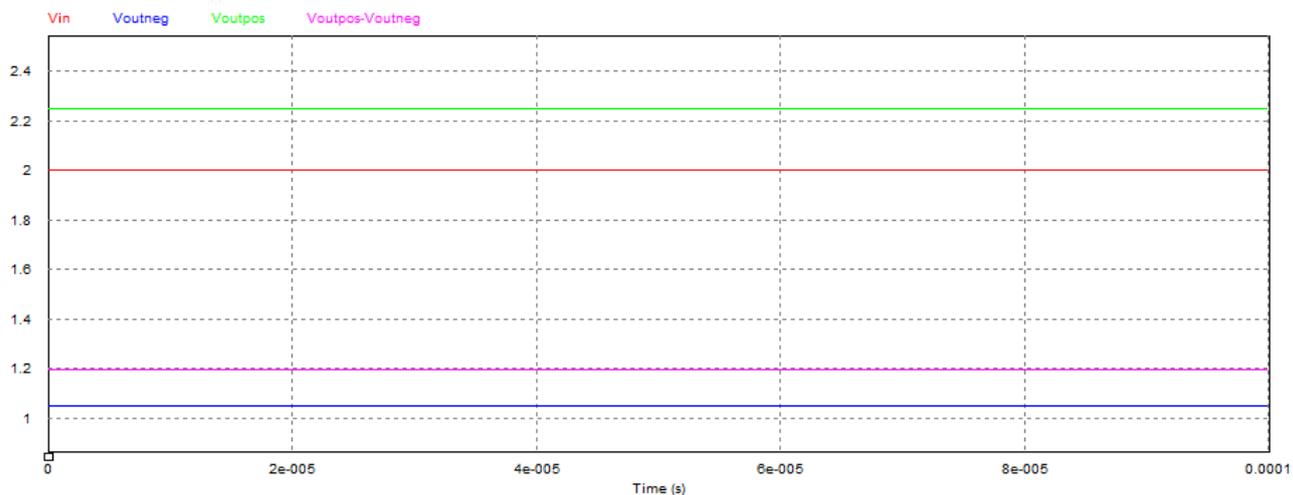


Figura 41. Resultados obtenidos al colocar el divisor de tensión a la entrada del buffer.

Puede verse como la tensión diferencial de salida no es igual que la de entrada. Esto se debe a la introducción del divisor de tensión. A continuación se comprobará si los resultados vuelven a ser los deseados al introducir un seguidor de tensión entre los dos subsistemas. En la figura 42 puede verse un esquema general del conjunto.

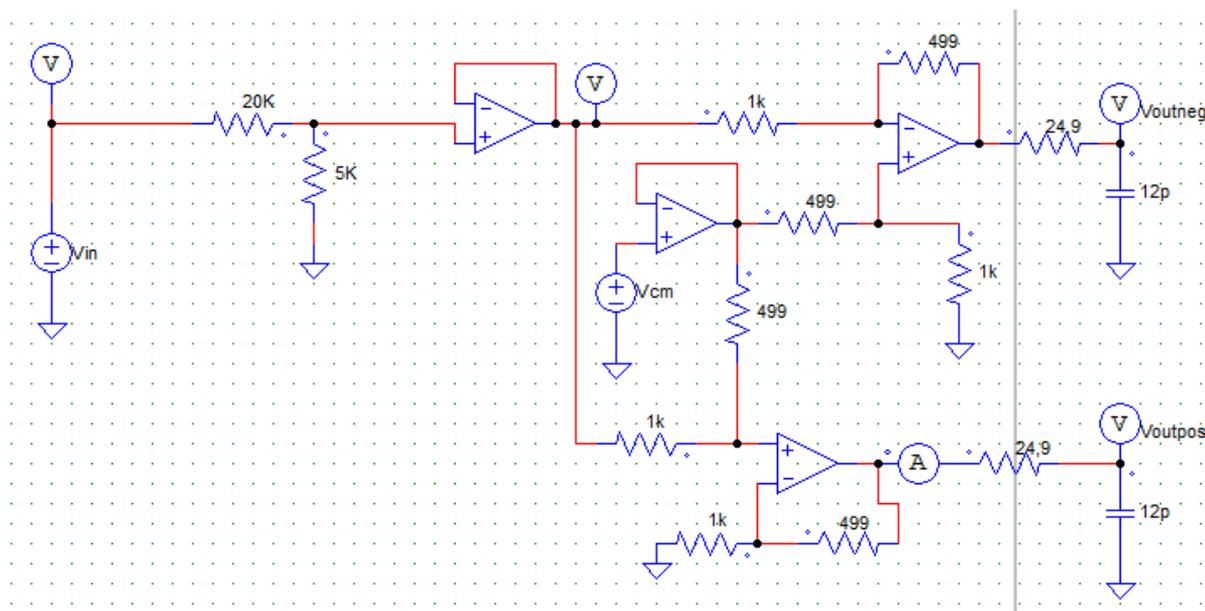


Figura 42. Esquema del divisor de tensión, seguidor y buffer.

Al simular este sistema, se obtienen los resultados que pueden verse en la figura 43.

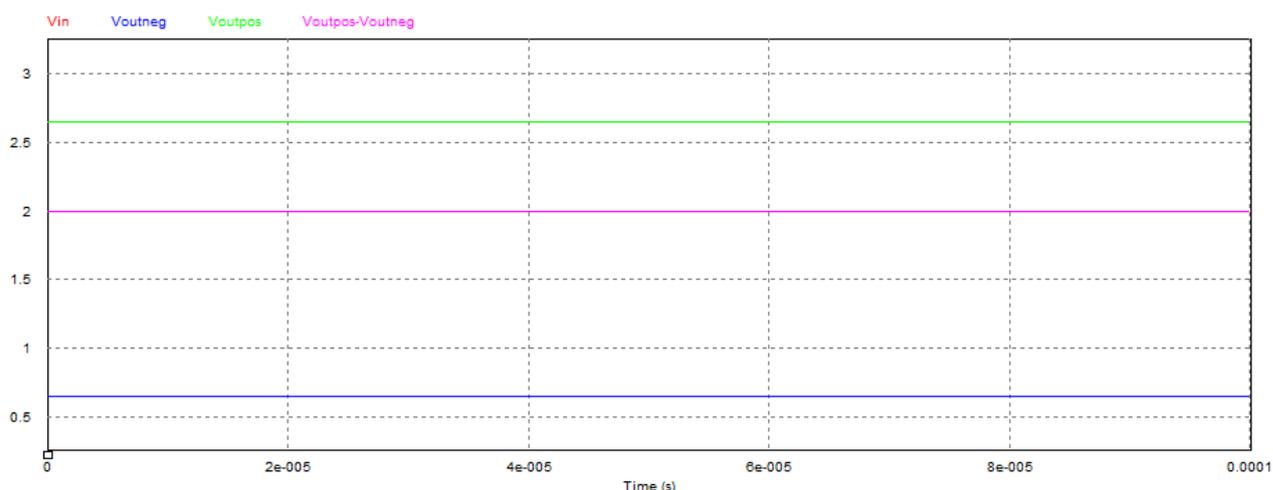


Figura 43. Resultados obtenidos al incorporar el seguidor de tensión entre el divisor y el buffer.

Puede verse como la tensión diferencial de salida, vuelve a ser la misma que la introducida en la entrada, por lo tanto se puede determinar que el sistema opera de la forma deseada.

3.2.3 Implementación de la PCB

De esta manera se ha decidido implementar una PCB para colocarla entre la salida del convertidor *buck* y la entrada del buffer. En esta se incluirán el divisor de tensión y el seguidor, asegurando así el correcto funcionamiento de todo el sistema.

3.2.3.1 Diseño de la PCB

De igual forma que en el diseño anterior, se ha utilizado el software de *Design Spark*. En primer lugar, se enumeran todos los componentes que integrarán la PCB. En la tabla 2 pueden verse estos.

Tabla 2. Componentes seleccionados para la implementación de la PCB que incluye el divisor de tensión y el seguidor.

Componente	Código RS	Fabricante	Referencia fabricante	Valor	Dimensiones (mm)	Tolerancia	Precio (€)	Unidades	Precio total (€)
Resistencia 25K Ω	-	-	-	25 k Ω	8 mm	1%	-	1	-
Resistencia 5K Ω	-	-	-	5 k Ω	8 mm	1%	-	1	-
ADA 4981-2	-	-	-	-	SOIC8	-	-	1	-
Conectores SMA	616-3422	TE Conect.	5-1814832-1	-	-	-	2,02	2	4,04
Pin headers	-	-	-	-	-	-	-	2	-

Para que sea más sencillo cambiar las resistencias en caso de que se desee cambiar el valor del divisor de tensión para trabajar con otros conversores, se ha decidido instalar resistencia de orificios pasantes. También se han incluido dos conectores SMA para conectar las entradas y salidas del sistema, y dos *pin-headers* para la conexión de la tierra y la alimentación.

Para continuar, se exportan estos componentes de la librería de RS de *Design Spark* y se dibujan las conexiones entre cada uno de los elementos. En la figura 44 puede verse un esquema de la circuitería de la placa.

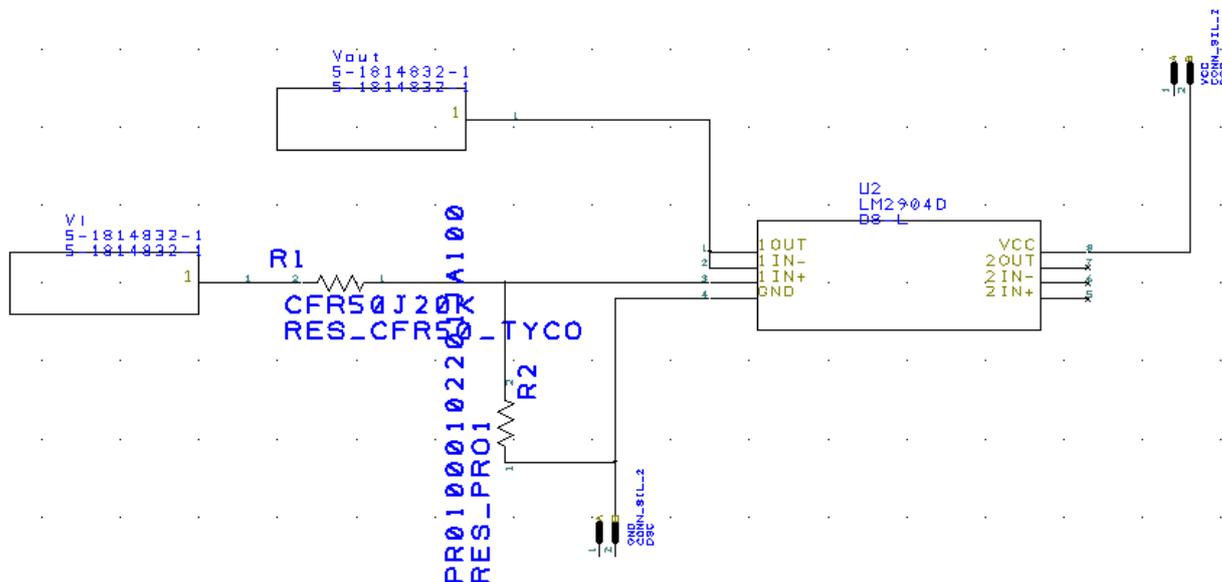


Figura 44. Esquema de la PCB del divisor de tensión y del seguidor.

Como pasaba en el caso anterior, solo se ha utilizado uno de los operacionales con los que cuenta el ADA4891-2, por lo tanto se ha dejado el segundo operacional sin conectar.

Una vez se han determinado las conexiones entre los componentes, puede diseñarse el *layout* de la placa. Este se ha realizado tratando de minimizar el espacio y en una sola cara, ya que el número de componentes instalados es pequeño. También se ha tratado de minimizar el recorrido de las pistas por las que circulará la señal y de aumentar la superficie de las pista de alimentación y tierra. En la figura 45 puede verse el resultado obtenido al aplicar todos los criterios anteriores.

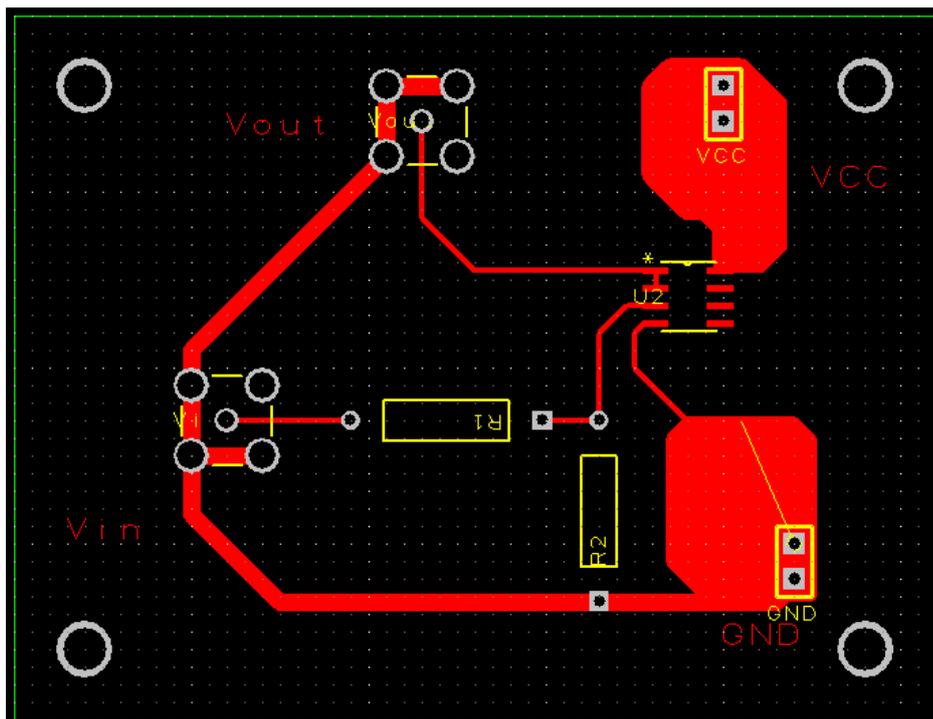


Figura 45. *Layout* de la PCB del divisor de tensión y del seguidor.

Una vez diseñado el *layout* de la placa, puede enviarse a fabricar.

3.2.3.2 Implementación física

Para continuar, se ha procedido a soldar los componentes sobre los *pads* correspondientes de la PCB. En la figura 46 podemos ver la placa antes de comenzar el proceso de soldadura.

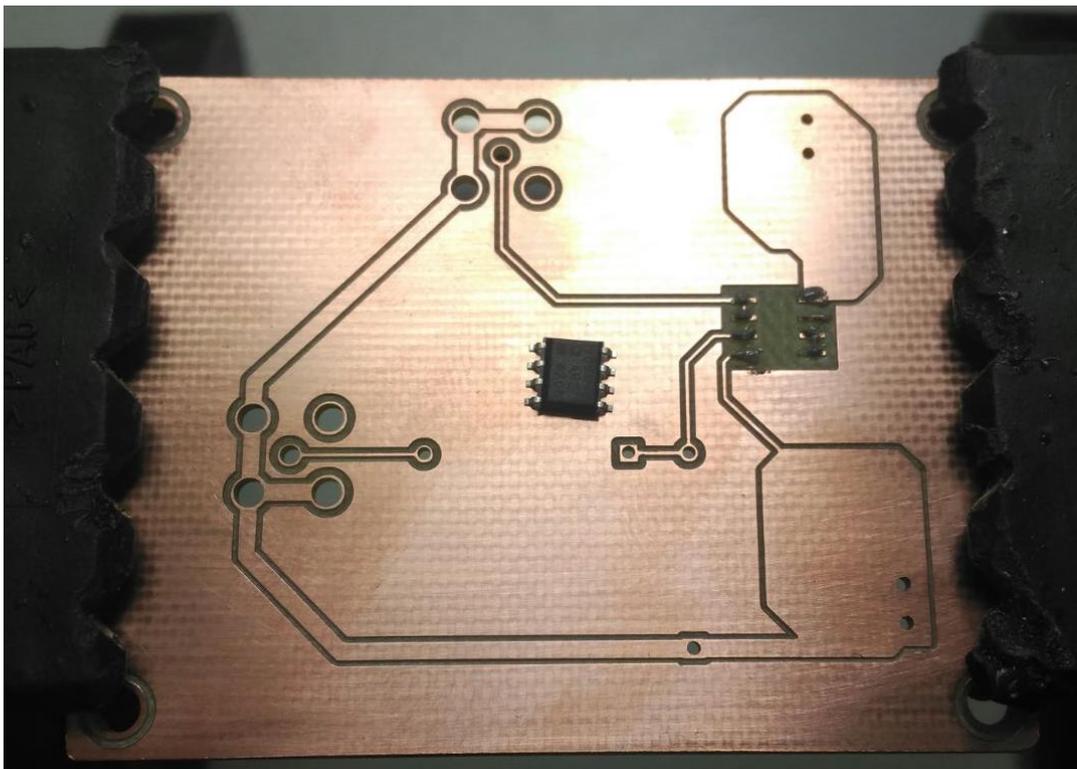


Figura 46. PCB del divisor de tensión y del seguidor.

El ADA 4891-2 se ha soldado utilizando la pasta SMA y una pistola de aire, ya que los *pads* de este son muy pequeños y soldarlos con estaño podría ser más problemático. En cambio, para el resto de componentes se ha utilizado un soldador y estaño, ya que las dimensiones de estos no presentan serias dificultades. En la figura 47 puede verse la placa con todos los componentes soldados.

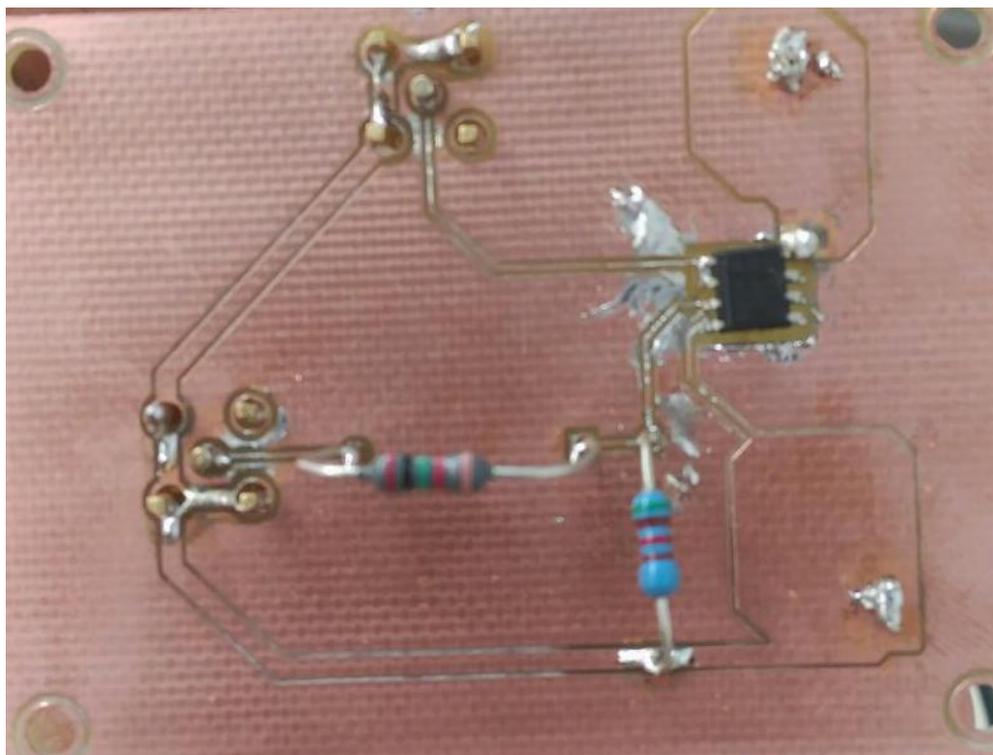


Figura 47. PCB del divisor de tensión y del seguidor con los componentes soldados.

Como puede verse en la figura 47 los componentes quedarían en la cara de debajo de placa, realizándose así el cableado de los conectores SMA y los *pin-headers* por la parte superior. Esto se ha realizado de esta manera para facilitar la soldadura de los conectores SMA.

4 Caracterización *buck*

Una vez diseñado el acondicionamiento de señal que permitirá conectar el convertidor *buck* con la FPGA se tratará de visualizar las señales de salidas de este. En primer lugar se procurará visualizarlas a través del osciloscopio y seguidamente utilizando el software Chipscope.

4.1 Programación *buck* y comprobación a partir del osciloscopio

En primer lugar, se debe programar debidamente las señales que controlarán el comportamiento del *buck*. Para esto se han utilizado dos bloques en VHDL diseñados en [1]. El primero es el encargado de controlar los transistores que regulan el ciclo de trabajo del sistema y ha sido programado de tal manera que deja un tiempo muerto entre la señal de activación de cada transistor, para asegurar que nunca conduzcan al mismo tiempo. La señal *control_buck* determina el ciclo de trabajo del sistema, mientras que las señales *high_buck* y *low_buck*, controlan los transistores de lado alto y bajo respectivamente.

El segundo, ha sido implementado mediante otro par de transistores y permite variar la carga de 0,5 a 50 Ω para comprobar cómo se comporta el sistema ante cambios bruscos de carga. Las señales *control_load*, *low_load* y *high_load* funcionan de manera análoga al bloque anterior. En la figura 48 pueden verse ambos bloques.

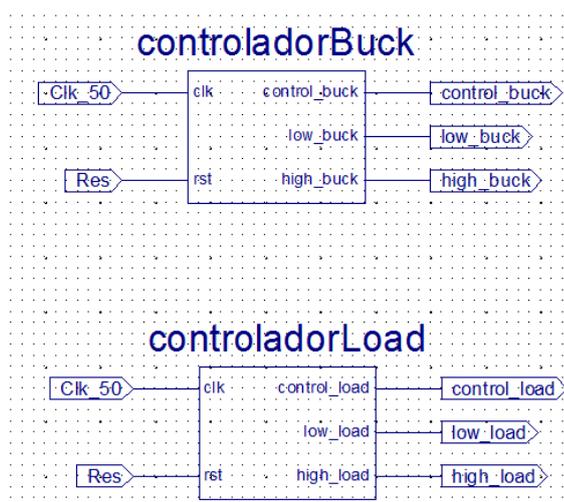


Figura 48. Entidades en VHDL utilizadas para el control del *buck*.

Las salidas y entradas de cada uno de los bloques han sido asociadas a diferentes pines de la FPGA.

En la figura 49 pueden observarse las señales de *control_buck* (azul), *low_buck* (rosa) y *high_buck* (amarillo), y más detalladamente en la figura 50 puede verse el tiempo muerto de unos 80 ns entre *low_buck* (azul) y *high_buck* (amarillo).

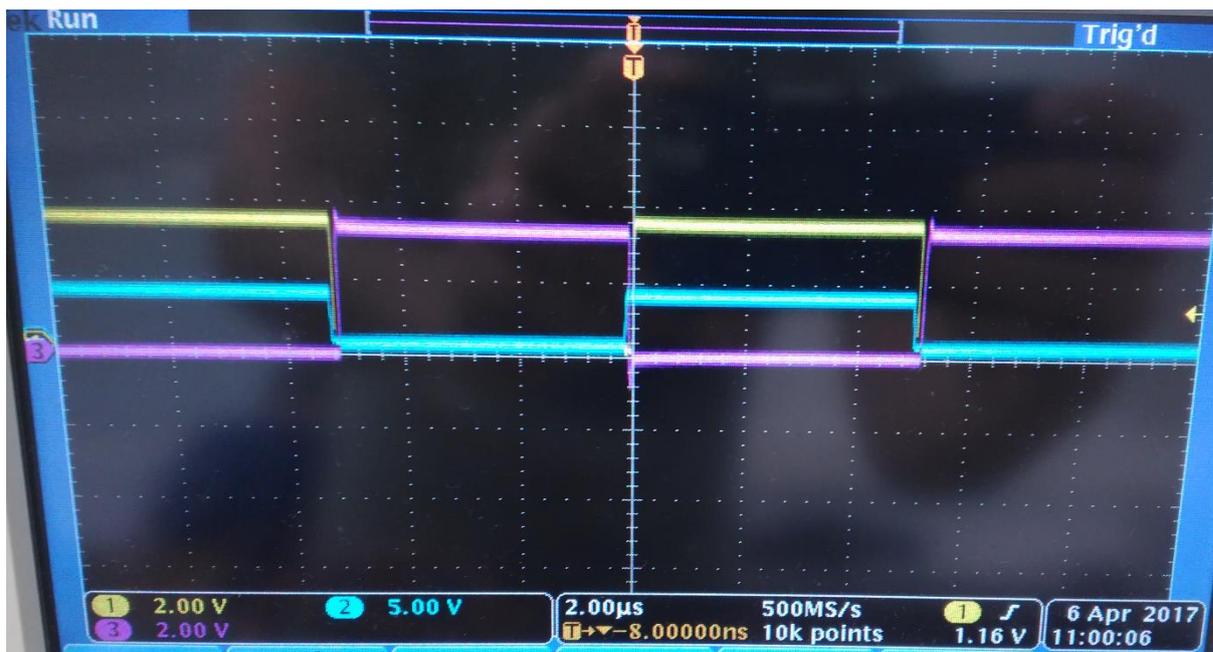


Figura 49. Resultados obtenidos de las señales *control_buck* (azul), *low_buck* (rosa) y *high_buck* (amarillo).



Figura 50. Resultados obtenidos de las señales *low_buck* (azul) y *high_buck* (amarillo).

A continuación se comprobará cómo funciona el controlador de carga. La señal que se encargará de controlar esta variación de carga (*control_load*) tendrá un periodo mucho mayor que la *control_buck* (de 10,48 ms) para que de esta manera puedan verse los cambios producidos por esta claramente. Esto puede apreciarse claramente en la figura 51, en la que se ven las señales de *control_buck* (azul), *control_load* (rosa) y *low_load* (amarillo).

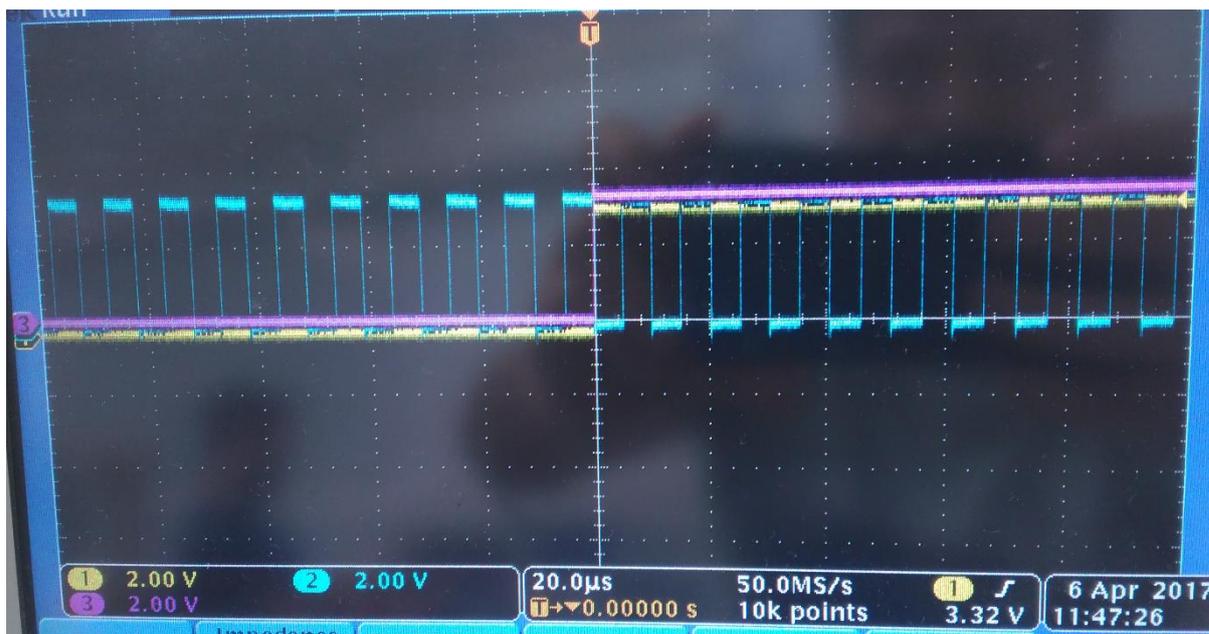


Figura 51. Resultados obtenidos de las señales *control_buck* (azul), *control_load* (rosa) y *low_load* (amarillo).

Sin embargo, para que sea más sencilla la toma de datos se ha decidido conectar solo el pin correspondiente a la señal *low_load* y dejar esta señal fija a 1. De esta manera tendremos siempre la carga de $0,5 \Omega$.

En las simulaciones anteriores puede verse que el ciclo de trabajo es del 50%. Para entender como se ha realizado la programación de este puede verse el código VHDL que corresponde a la figura 52.

```

generar_u: process (clk, rst)
begin
    u_buck <= counter(8);
    --u_buck <= counter(8) and counter(7);
    if rst = '1' then
        counter <= (others => '0');
    elsif (clk'EVENT and clk = '1') then --50Mhz un periodo de reloj
        counter <= counter+1;
    end if;
end process;
    
```

Figura 52. Código VHDL para la implantación de un ciclo de trabajo del 50%.

En la figura 52 puede verse como *u_buck* (el ciclo de trabajo) toma el valor del bit más significativo de un contador de 8 bits. De esta manera el *u_buck* estará a cero mientras el valor del contador sea de 0 a 127 y a uno cuando este valga entre 128 y 255. Como el reloj utilizado es de 50 MHz, el tiempo en el que el contador va aumentando de valor es de 20 ns. Por lo tanto, *u_buck* estará a 0 durante $2,56 \mu\text{s}$ y después el mismo tiempo a 1, así que puede entenderse que el ciclo de trabajo es del 50%.

Finalmente, se controlará el *buck* a través de las señales descritas anteriormente y visualizar en el osciloscopio los valores de tensión de salida y de corriente en el inductor. Para esto se han cargado las entidades en VHDL arriba descritas a la FPGA y conectados los pines correspondientes a las salidas a las entradas de control del convertidor *buck*. En la figura 53 pueden verse los resultados de dichas simulaciones.



Figura 53. Resultados obtenidos del convertidor *buck* con un ciclo de trabajo del 50% de las señales de *control_buck* (azul), tensión de salida (amarillo) y corriente de la bobina (verde).

Puede verse como los resultados obtenidos, tanto de la tensión de salida como de la corriente de la bobina, coinciden con las simulaciones realizadas y, por lo tanto, se ajustan a lo esperado.

Para continuar con el análisis del convertidor *buck*, puede verse cómo se comporta este al variar el ciclo de trabajo al 25%. Para cambiar el valor de este, no se tiene más que hacer que el *u_buck* tome el valor del producto de los dos bits más significativos, de esta manera estará a 0 de 0 a 191 y a 1 de 192 a 255. En la figura 54 puede verse el código cambiado en VHDL.

```

generar_u: process (clk, rst)
begin
    --u_buck <= counter(8);
    u_buck <= counter(8) and counter(7);
    if rst = '1' then
        counter <= (others => '0');
    elsif (clk'EVENT and clk = '1') then --50Mhz un periodo de reloj
        counter <= counter+1;
    end if;
end process;
    
```

Figura 54. Código VHDL para la implementación de un ciclo de trabajo del 25%.

En la figura 55 y 56 puede verse el ciclo de trabajo del sistema y las señales de *low_buck* y *high_buck* respectivamente. Se observa como este ha sido reducido al 25% y que mantenemos un tiempo de retardo de 80 ns.

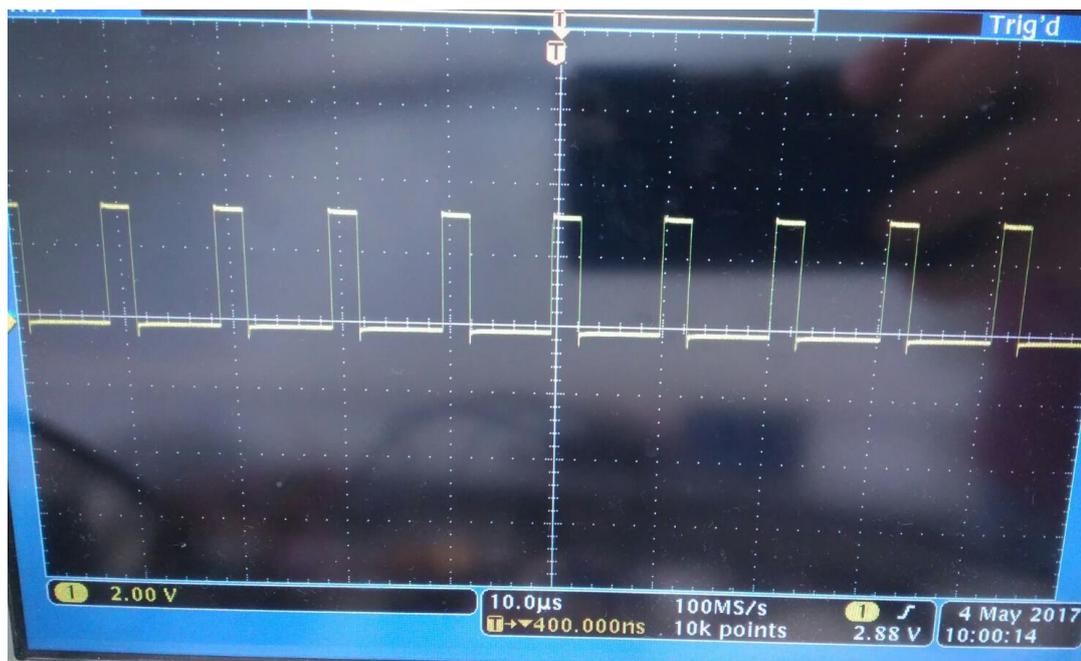


Figura 55. Resultado obtenido de la señal de *control_buck* al aplicar un ciclo de trabajo del 25%.

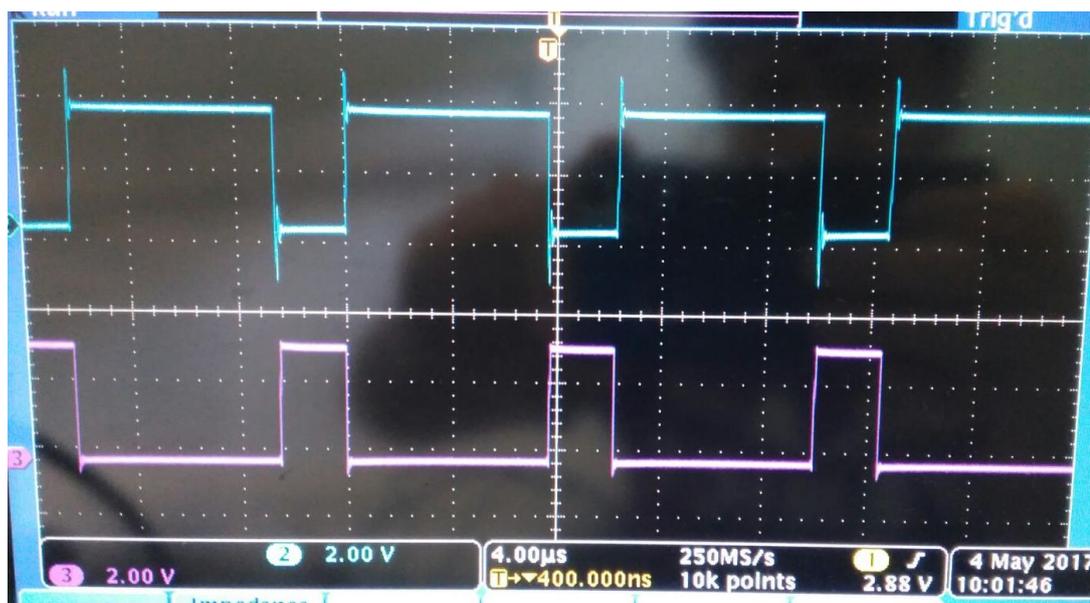


Figura 56. Resultados obtenidos de las señales de *low_buck* (azul) y *high_buck* (rosa) al aplicar un ciclo de trabajo del 25%.

En la figura 57 puede verse que a pesar de disminuir el ciclo de trabajo, tanto la tensión de salida como la corriente de la bobina siguen comportándose de la manera esperada.



Figura 57. Resultados obtenidos del convertidor *buck* con un ciclo de trabajo del 25% de las señales de tensión de salida (amarillo) y corriente de la bobina (verde).

5 Toma de datos con ChipScope

Para continuar con el análisis del sistema y comprobar que el acondicionamiento de señal diseñado funciona correctamente, se tratará de visualizar los resultados con la herramienta ChipScope, que permite monitorizar el valor de diferentes bits de la FPGA. Para esto es necesario implementar una conexión entre las salidas del conversor analógico digital y alguna de las entradas de la FPGA. Este trabajo ha sido realizado previamente a la elaboración de este proyecto, así que simplemente se utilizará la entidad VHDL diseñada para la toma de datos. En la figura 58 puede verse el bloque utilizado.

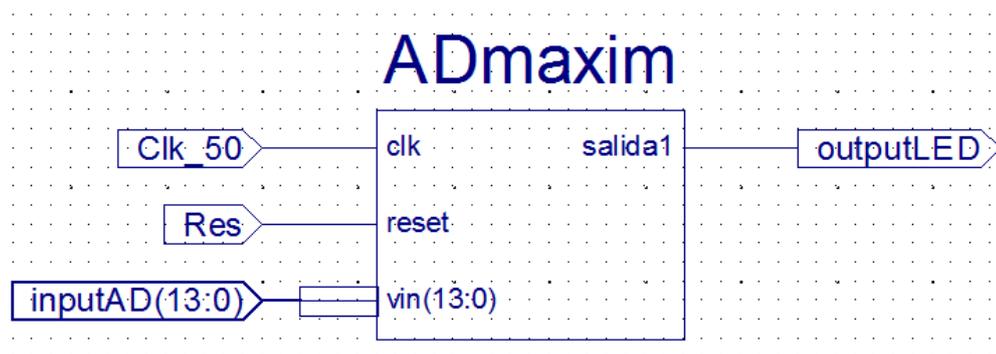


Figura 58. Entidad VHDL para la lectura de la salida del ADC.

5.1 Resultados obtenidos Buffer

Una vez explicado cómo se realiza la obtención de datos a partir de ChipScope, en primer lugar, se comprobará el funcionamiento del Buffer diseñado por separado. Con esta finalidad, se han introducido en la entrada del buffer una señal sinusoidal, se han conectado las alimentaciones y tierras, el reloj del kit de evaluación con la señal de reloj de la FPGA y los pines de salida del ADC con la FPGA para poderlos monitorizar a través de ChipScope. En la figura 59 puede verse esta configuración.

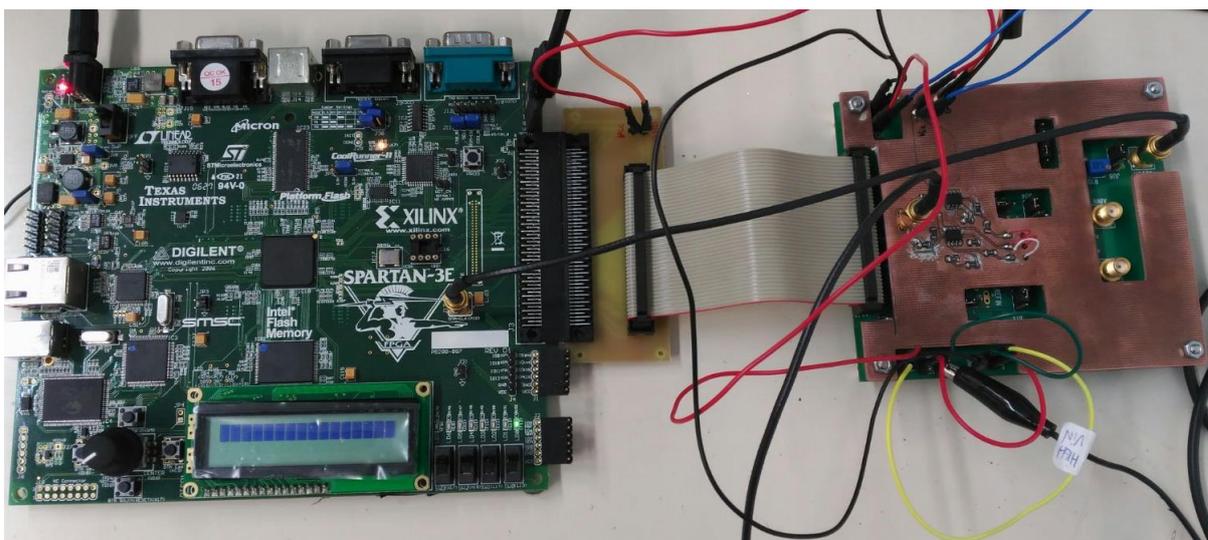


Figura 59. Configuración de la conexión del buffer, el ADC y la FPGA.

De esta manera, se obtienen los resultados que pueden verse en las figuras 60, 61, 62, 63 y 64, para diferentes valores de la señal de entrada.

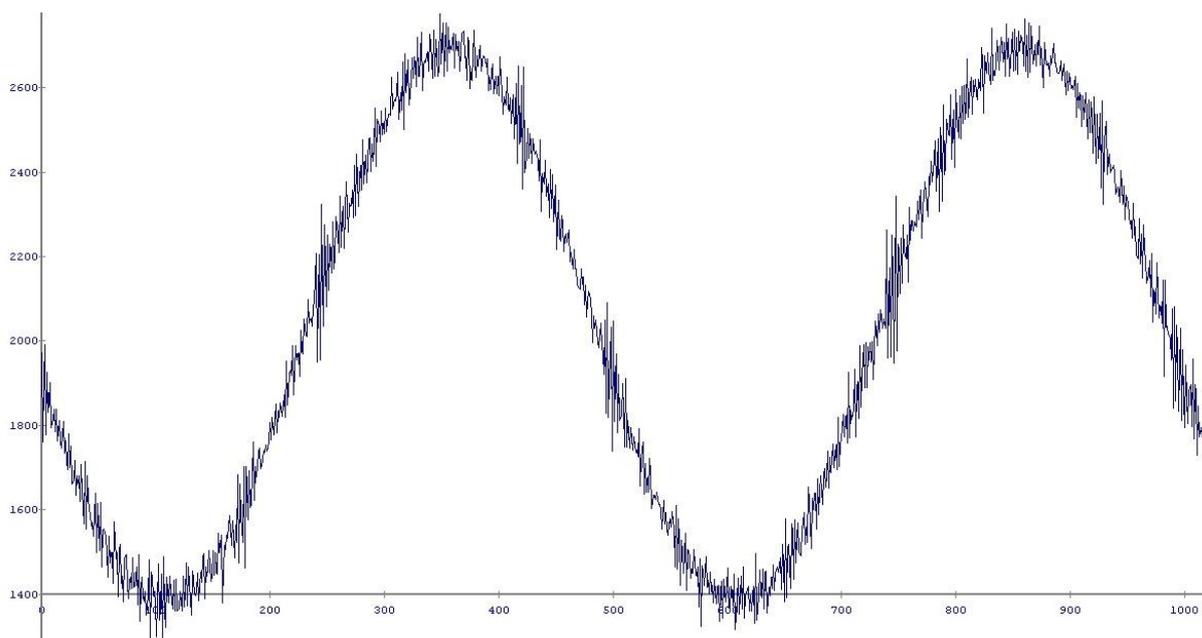


Figura 60. Resultados obtenidos al introducir una señal de entrada de 50 mV de amplitud, 100 KHz de frecuencia y 0 V de offset.

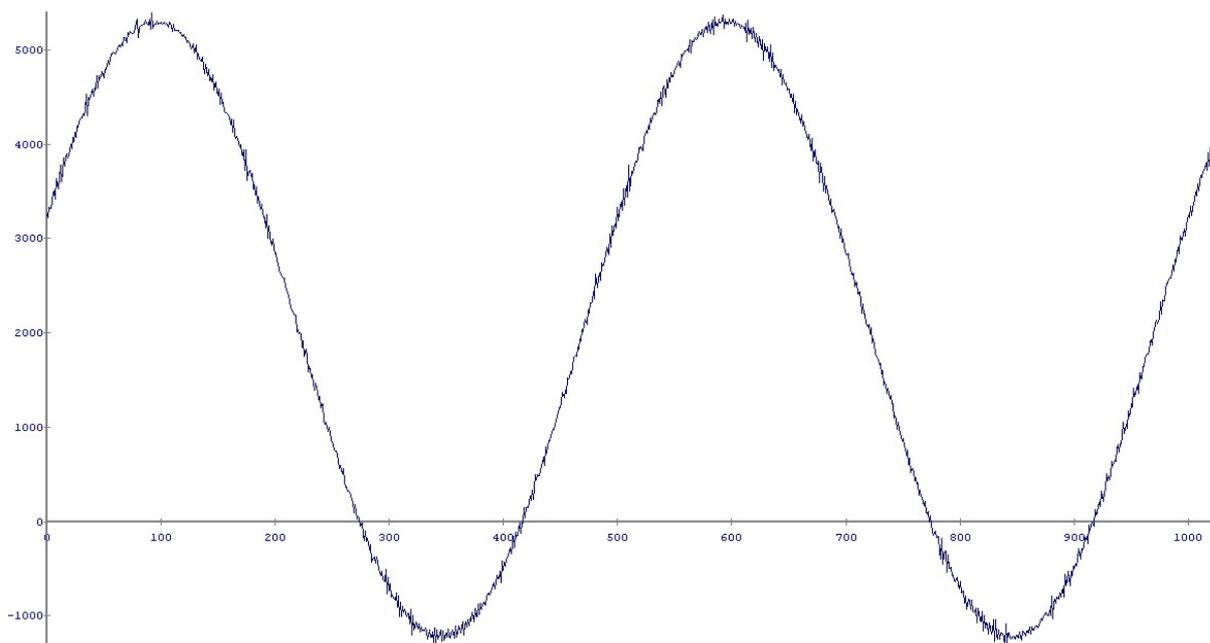


Figura 61. Resultados obtenidos al introducir una señal de entrada de 250 mV de amplitud, 100 KHz de frecuencia y 0 V de offset.

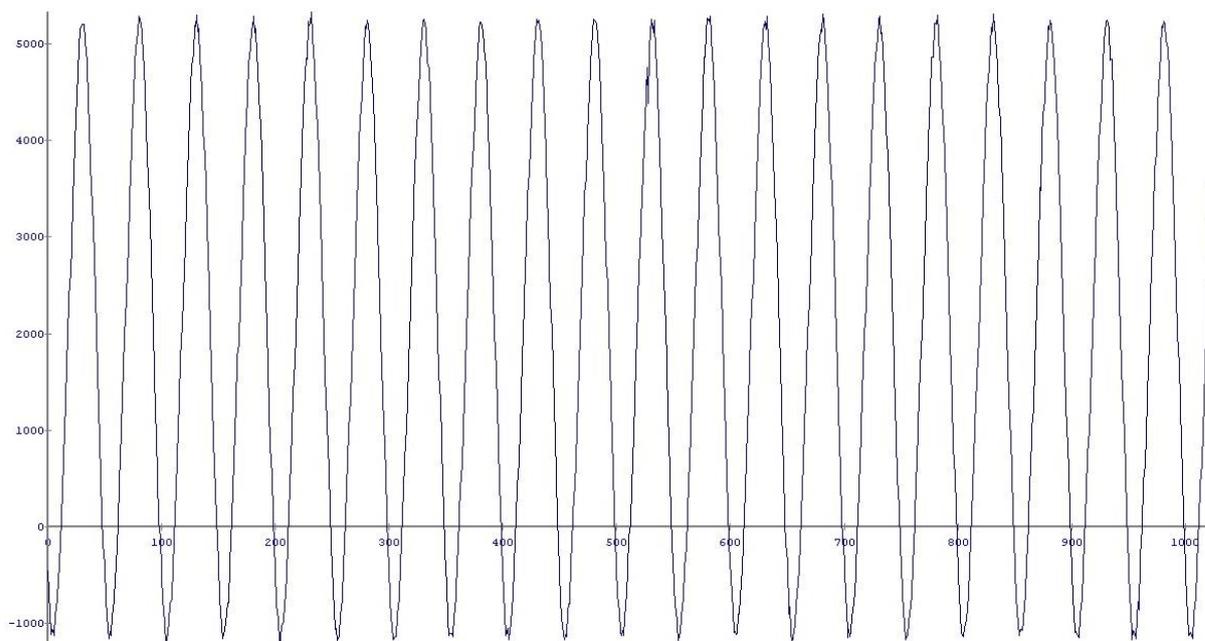


Figura 62. Resultados obtenidos al introducir una señal de entrada de 250 mV de amplitud, 1 MHz de frecuencia y 0 V de offset.

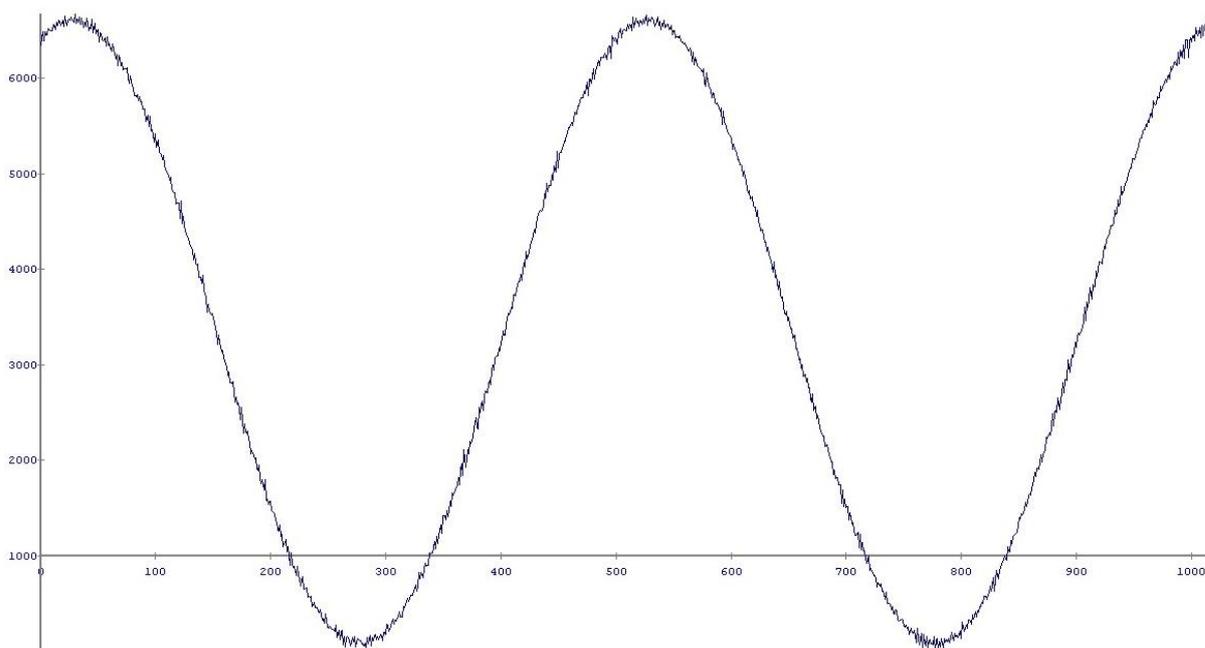


Figura 63. Resultados obtenidos al introducir una señal de entrada de 250 mV de amplitud, 100 KHz de frecuencia y 100 mV de offset.

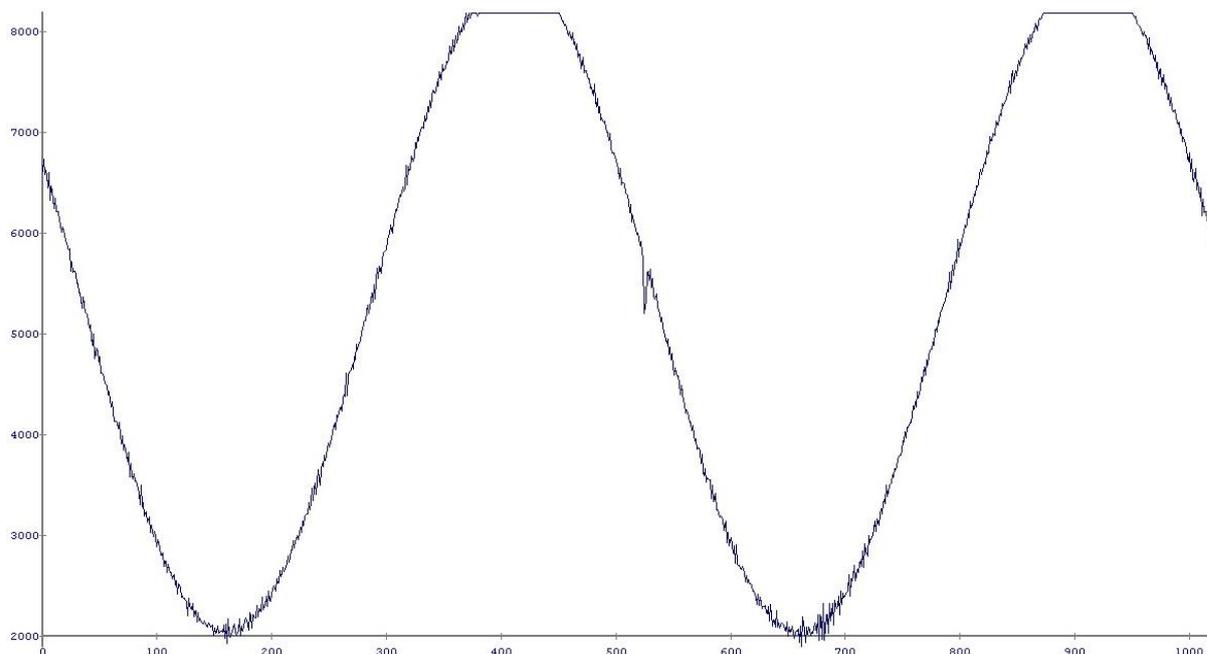


Figura 64. Resultados obtenidos al introducir una señal de entrada de 250 mV de amplitud, 100 KHz de frecuencia y 250 mV de offset.

En las figuras anteriores puede verse como la onda sinusoidal que se le proporciona a la entrada es captada por el software de monitorización con mucha fidelidad, aunque apareciendo ciertos defectos que se comentarán al final de este apartado. Al aumentar la frecuencia a 1 MHz, se observa cómo el sistema sigue funcionando perfectamente. Para entender los valores que va tomando la onda en estas representaciones, se debe analizar la resolución de nuestro convertidor ADC. En [11] (*data sheet* del Maxim12555), puede verse que la resolución de este es de 14 bits. Al estar diseñado para representar valores negativos, el rango de valores que puede representar es el siguiente:

$$(-2^{n-1} - 1 \dots 2^{n-1}) \quad (42)$$

Siendo n el número de bits. Así, en el caso del ADC que ocupa este proyecto se obtiene:

$$(-2^{14-1} - 1 \dots 2^{14-1}) \quad (43)$$

$$(-8.191 \dots 8.192) \quad (44)$$

De esta manera puede verse como el valor máximo que puede llegar a tomar la onda es de 8.192, tal como se ve en la figura 64, donde la onda es recortada en este valor, al no poder representar un número mayor.

Una vez se ha determinado el rango de valores que puede representar el ADC, el siguiente paso sería saber qué valor de tensión corresponde a cada uno de ellos. En [11] se puede ver que el fondo de escala del ADC es de 2,048 V, siendo el rango de $\pm 1,024$ V. Así, si se divide el valor de fondo de escala entre el valor del rango de bits, se obtiene el valor de tensión que corresponde al aumento de 1 bit, y se puede traducir cualquier valor de bits en tensión:

$$\frac{2,048 \text{ V}}{2^{14}} = 0,000125 \text{ V} = 125 \mu\text{V} \quad (45)$$

Así, puede observarse que la resolución en valores de tensión del ADC es de 125 μV .

Como se ha comentado anteriormente, en los resultados obtenidos puede verse que se tienen ciertas desviaciones respecto al valor proporcionado por la fuente: se obtiene el doble de la señal generada por la fuente y aparece un offset por defecto. Estos efectos son producidos debido a la impedancia que ve el Buffer, que en este caso es la impedancia de salida del generador de funciones (50 Ω). Estos efectos quedarán eliminados en el momento de incorporar el seguidor de tensión que integra la otra placa.

5.2 Resultados obtenidos sistema

Para continuar el análisis, se tratará de comprobar el funcionamiento de todo el sistema en conjunto. Para ello, se alimentará el convertidor *buck* y lo se controlará a partir de la FPGA. A la salida de este se colocará la placa integrada por los divisores de tensión y el seguidor. A continuación se conecta el Buffer y el conversor analógico digital y por último éste irá conectado a la FPGA a partir de la cual se realizará la monitorización de los resultados de salida a partir de Chipscope. Esta configuración es la que puede verse en la figura 65.

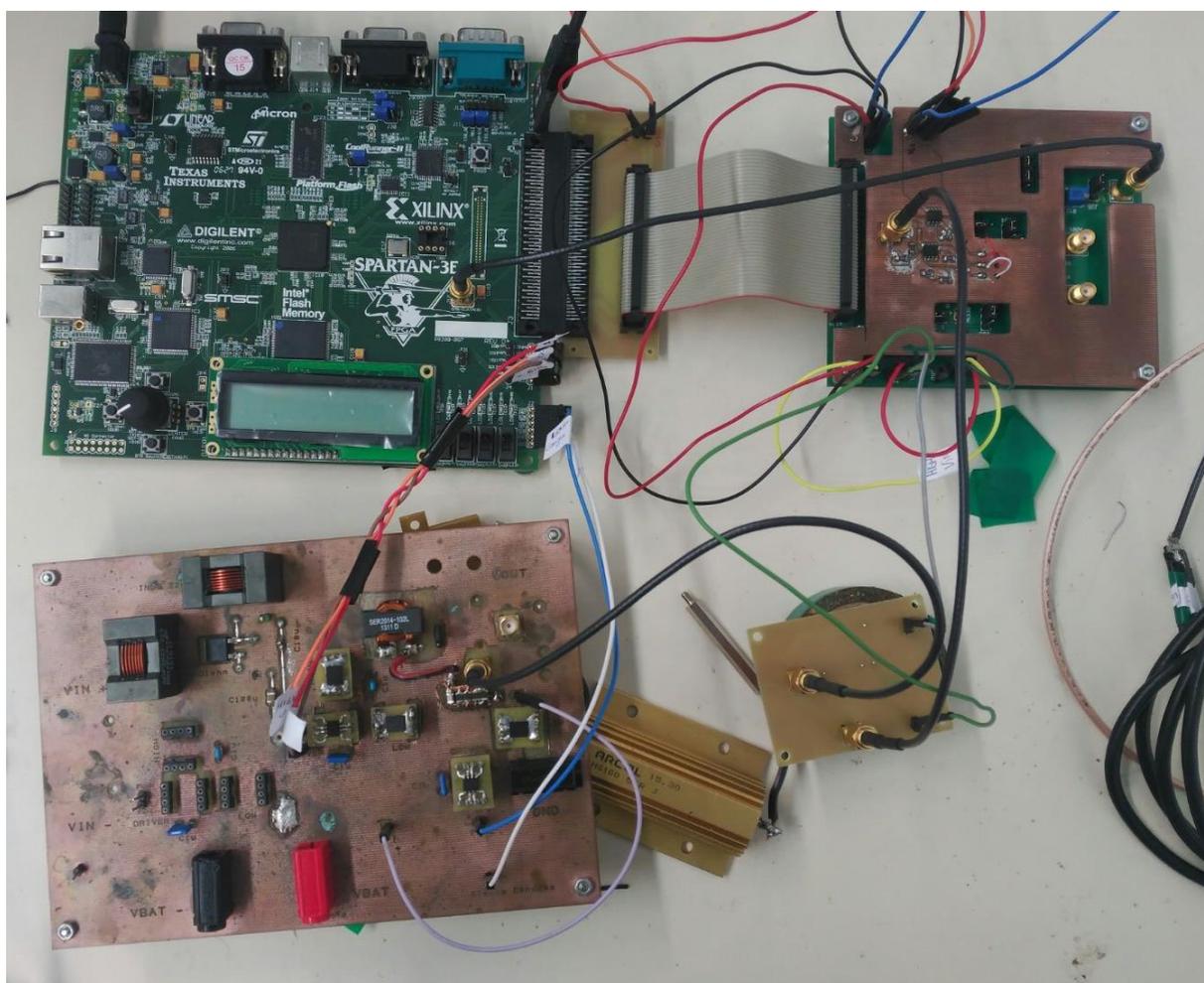


Figura 65. Configuración de la FPGA, el convertidor *buck*, el divisor de tensión y seguidor, el *buffer* y el ADC.

En la figura 66 pueden verse los resultados obtenidos con un ciclo de trabajo del 50%.

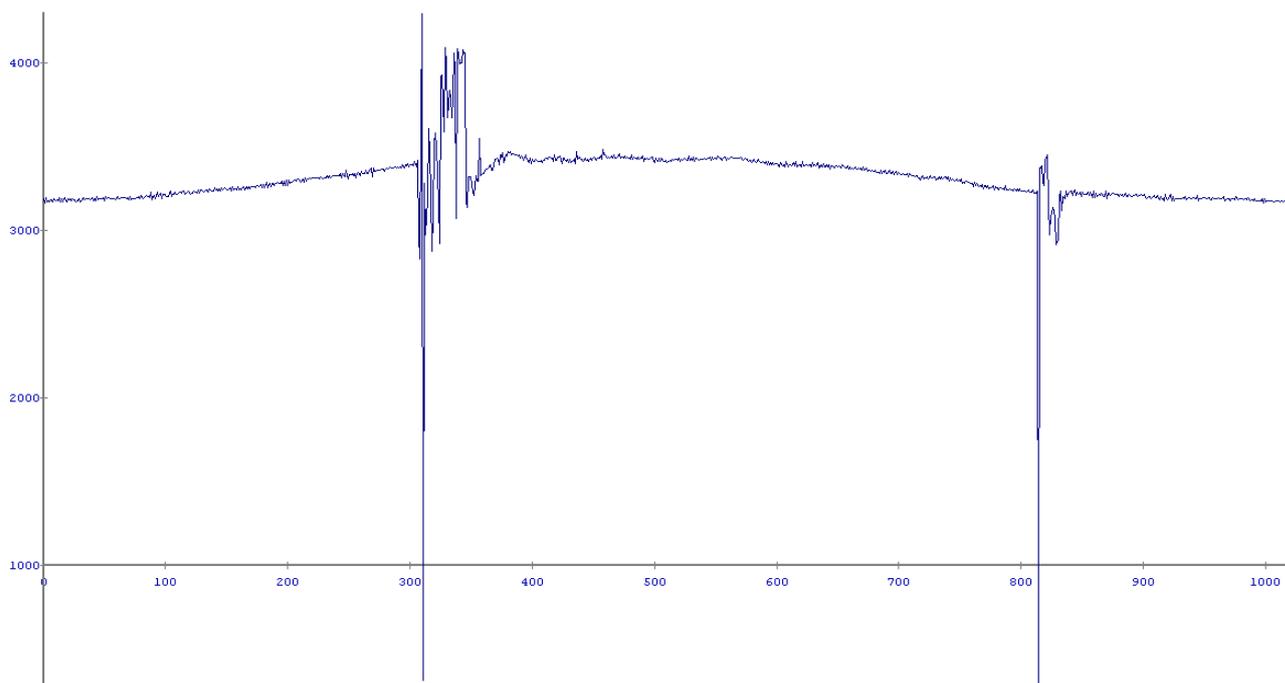


Figura 66. Resultados obtenidos con Chipscope de la tensión de salida del buck con un ciclo de trabajo del 50%.

Puede observarse como los tiempos de conducción de los transistores de lado alto y bajo duran el mismo tiempo, ya que se está trabajando con un ciclo de trabajo del 50%. Así, el valor proporcionado por el *buck* debería ser de 2,5 V (ya que está alimentado a 5 V). Si se tiene en cuenta que el divisor de tensión añadido a su salida reduce la tensión en un quinto, la tensión que debería leer el conversor analógico digital sería de 500 mV.

Como se ha comentado anteriormente, la resolución del ADC es de 125 μV y en la representación puede observarse como el valor oscila alrededor 3500, así que se tendría un voltaje de 438 mV.

Esta pequeña discrepancia entre el valor ideal y el real es debido a las pérdidas del convertidor, ya que en las medidas tomadas con el osciloscopio, ya podían apreciarse estas desviaciones. De esta manera puede determinarse que las medidas tomadas en Chipscope son correctas.

En la figura 67 puede verse los datos obtenidos al introducir un ciclo de trabajo del 25%.

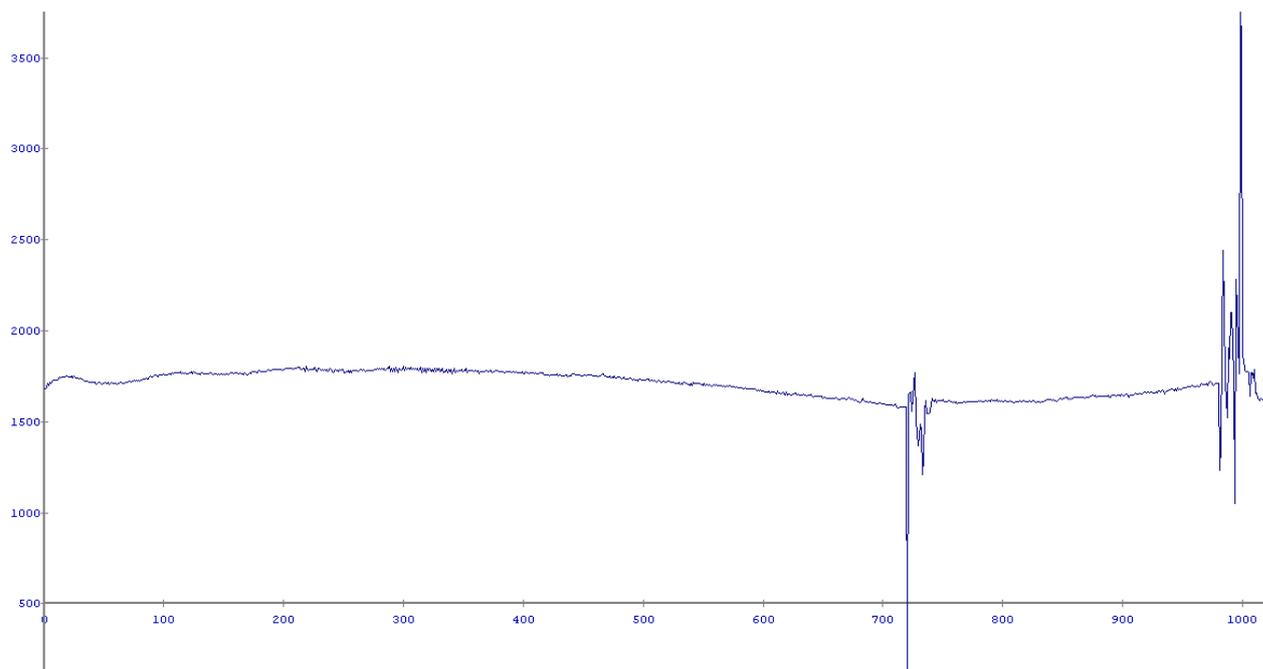


Figura 67. Resultados obtenidos con Chipscope de la tensión de salida del buck con un ciclo de trabajo del 25%.

En esta representación puede verse como el tiempo de conducción del transistor de lado alto se ha reducido, mientras que el del transistor de lado bajo ha aumentado (debido al ciclo de trabajo del 25%). De manera análoga al caso anterior, la tensión proporcionada por el convertidor debería ser de 1,25 V que al pasar por el divisor de tensión quedaría reducida a 250 mV.

Si se multiplica la resolución del ADC por el valor medio obtenido en la representación ($125 \mu\text{V} \cdot 1750$) se obtiene una tensión de 220 mV. De igual manera que en la representación anterior, puede verse como existe una pequeña discrepancia entre los valores teóricos y real que se debe a las pérdidas producidas en el convertidor.

6 Control

6.1 Modelo convertidor buck en lazo abierto en Matlab

Para diseñar el controlador PID que se encargará de mantener la tensión de salida en el valor deseado se utilizará el software de Matlab. En primer lugar se deberá crear un subsistema que se comporte como el convertidor *buck* en lazo abierto, esto se conseguirá utilizando la herramienta *Simulink* que permite introducir el diagrama de bloques que forman las ecuaciones de estado del sistema. Las ecuaciones de estado que rigen el comportamiento del convertidor son las siguientes:

$$V_L = D \cdot V_i - V_o \quad (46)$$

$$i_L = \frac{1}{L} \cdot \int V_L \cdot dt \quad (47)$$

$$V_c = \frac{1}{C} \cdot \int i_c \cdot dt \quad (48)$$

$$i_o = i_L - i_c \quad (49)$$

$$i_o = \frac{V_o}{R} \quad (50)$$

Por lo tanto, teniendo en cuenta las ecuaciones de estado anteriores, puede definirse el diagrama de bloques que puede verse en la figura 68.

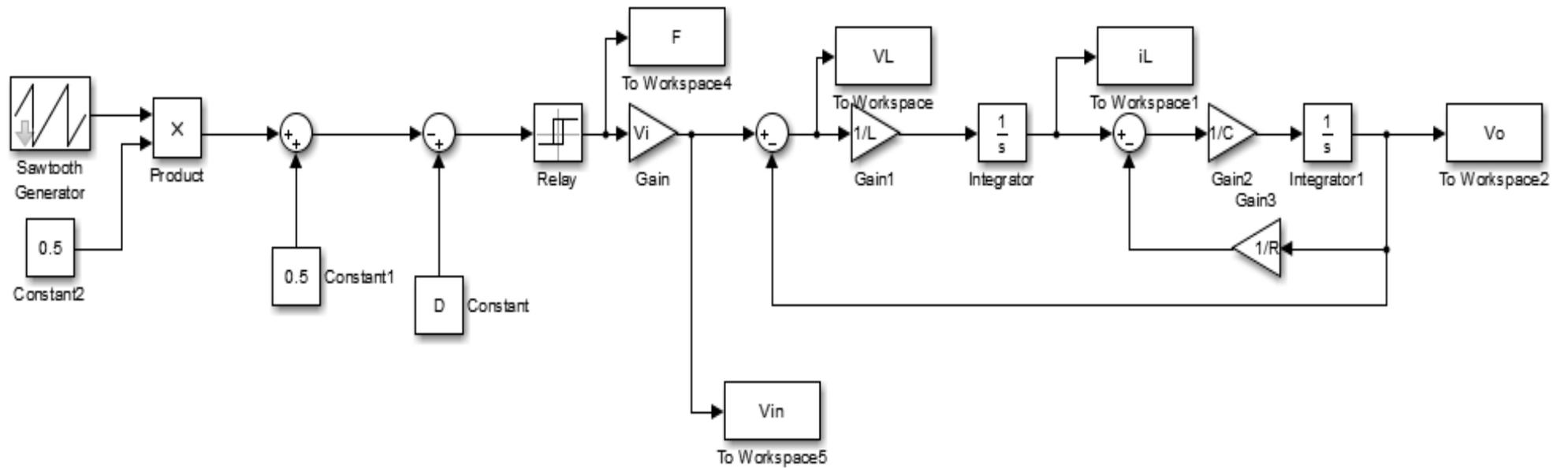


Figura 68. Diagrama de bloques del comportamiento de un convertidor *buck*

En esta figura puede verse como en primer lugar se ha implementado un modulador *pwm* utilizando un generador de señal de diente de sierra y un relé. Esto se ha utilizado para simular el valor de la señal de control del transistor, que tendrá un valor de 1 durante DT y de 0 durante T-DT.

Una vez se ha obtenido el diagrama de bloques implementado, puede verse como la entrada del sistema es el ciclo de trabajo (D) y la salida la tensión V_{out} . El resto de parámetros tomarán el valor del convertidor implementado en [1], ya que es el que se querrá controlar físicamente. En la figura 69 puede verse el fichero de inicialización de estos parámetros utilizado.

```

Vi=5; %Voltaje de entrada (V)
L=1.263e-6; % Inductancia (H)
C=270e-6%; Capacidad (F)
R=0.5; % Resistencia de carga (ohmios)
D=0.26; % Duty Cicle
    
```

Figura 69. Código Matlab para la inicialización de parámetros.

Se ha fijado un ciclo de trabajo de 0,26 debido a que se desea tener 1,3 V en la tensión de salida. De esta manera, en las figura 70, 71, 72, 73 y 74 pueden verse los resultados obtenidos del valor de las diferentes variable del sistema en régimen estacionario al simular en lazo abierto.

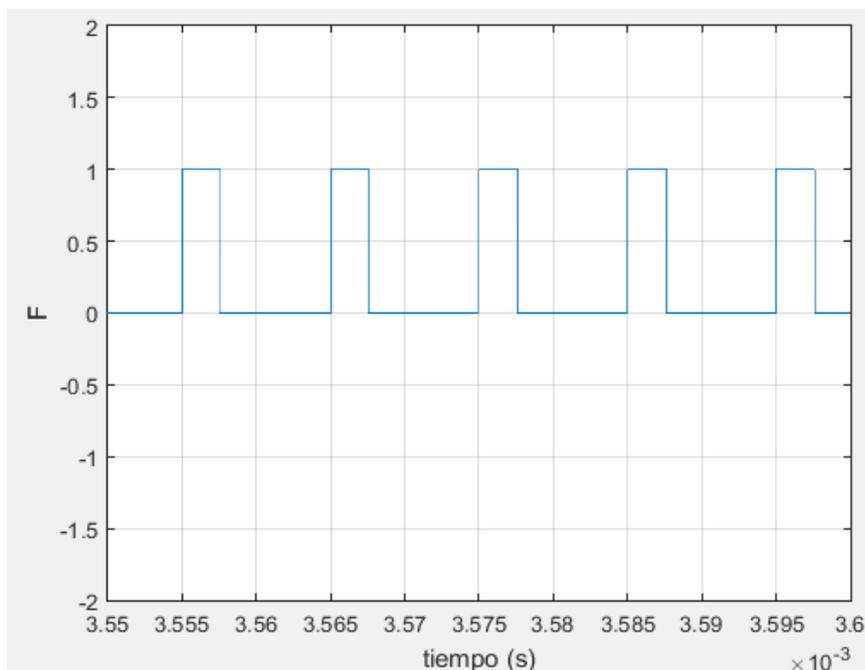


Figura 70. Representación del valor del ciclo de trabajo al trabajar al 24%.

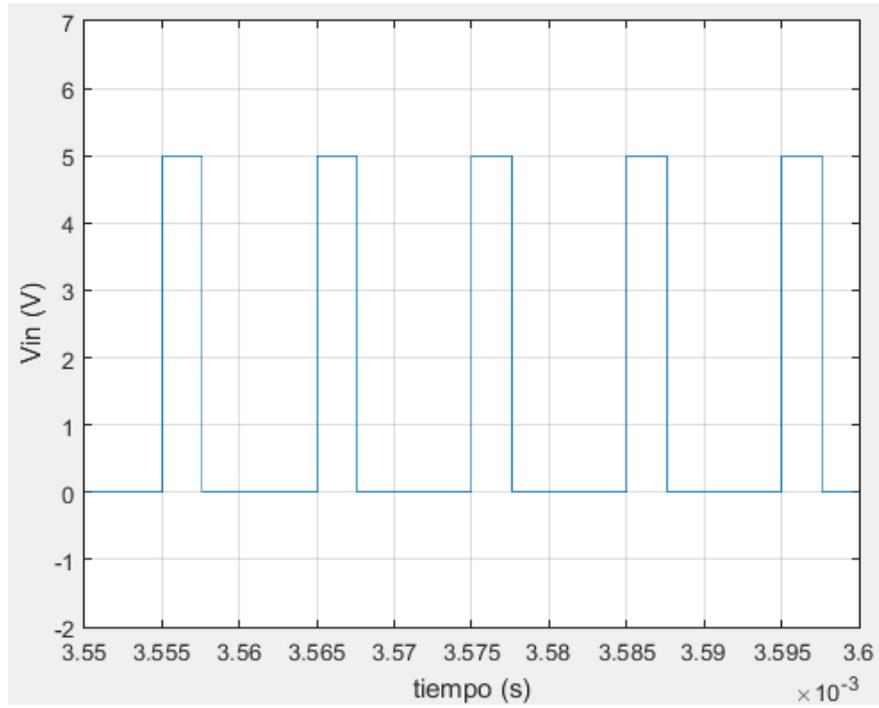


Figura 71. Resultados obtenidos de la tensión de entrada.

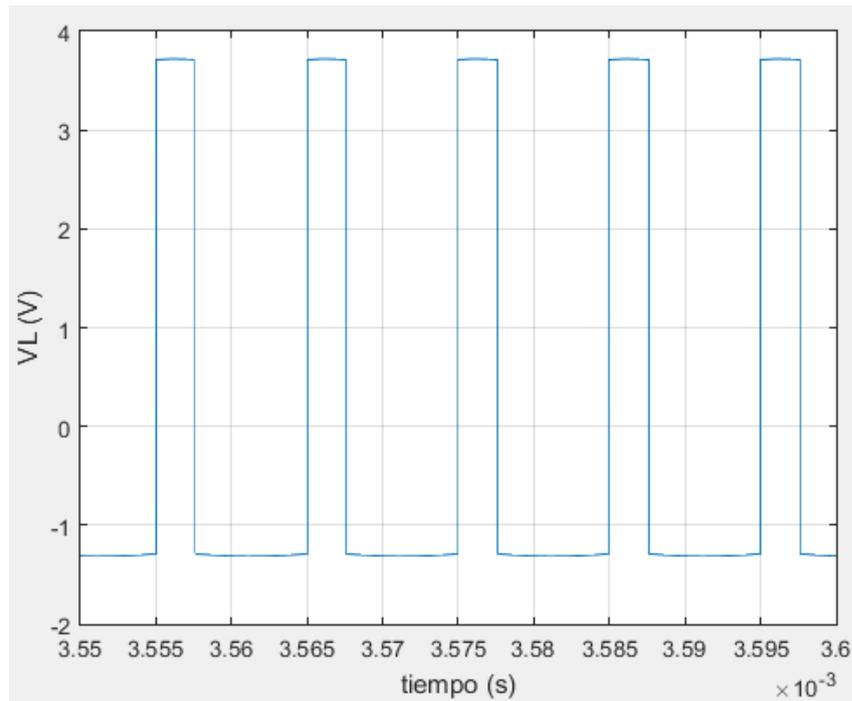


Figura 72. Resultados obtenidos de la caída de tensión en la bobina.

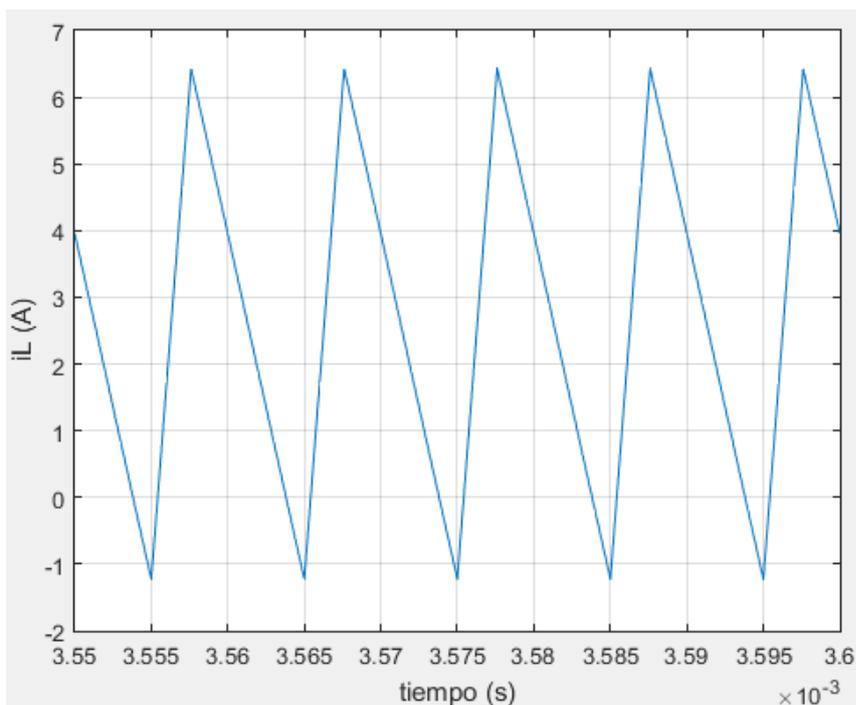


Figura 73. Resultados obtenidos de la corriente en la bobina.

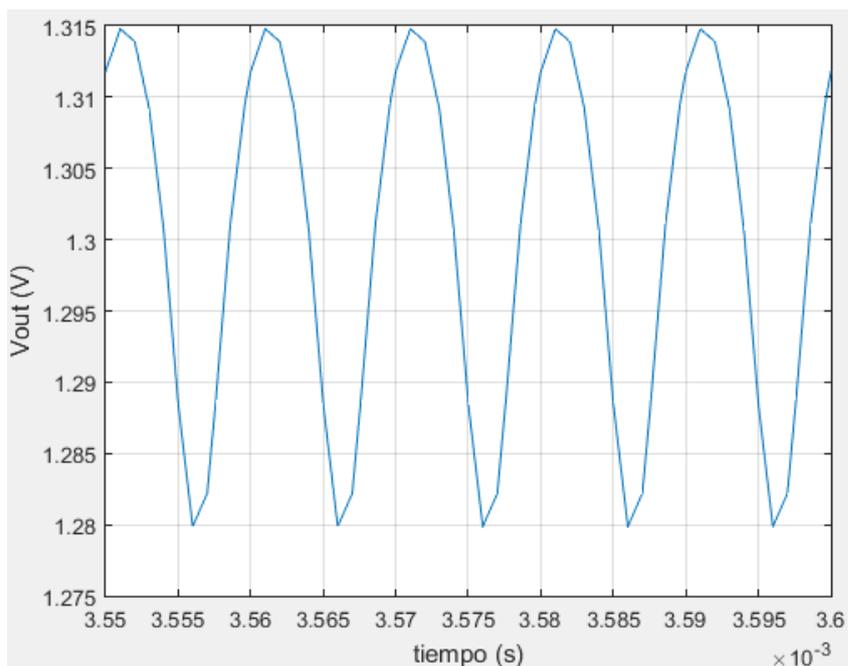


Figura 74. Resultados obtenidos de la tensión de salida.

Puede verse como cada uno de los parámetros va tomando valores acorde con sus ecuaciones de estado, y finalmente se obtiene una tensión de salida que oscila levemente sobre 1,3 V.

Sin embargo, también debe analizarse el régimen transitorio del sistema. En la figura 75 puede verse que al iniciar el sistema se obtiene un sub amortiguamiento que llega hasta una tensión de casi 2,5 V y que tarda un tiempo en estabilizarse. Además, en la implementación física del sistema debe tenerse en cuenta que el convertidor *buck* sufrirá

algunas pérdidas. De esta manera, para conseguir un correcto funcionamiento del sistema debe implementarse un controlador.

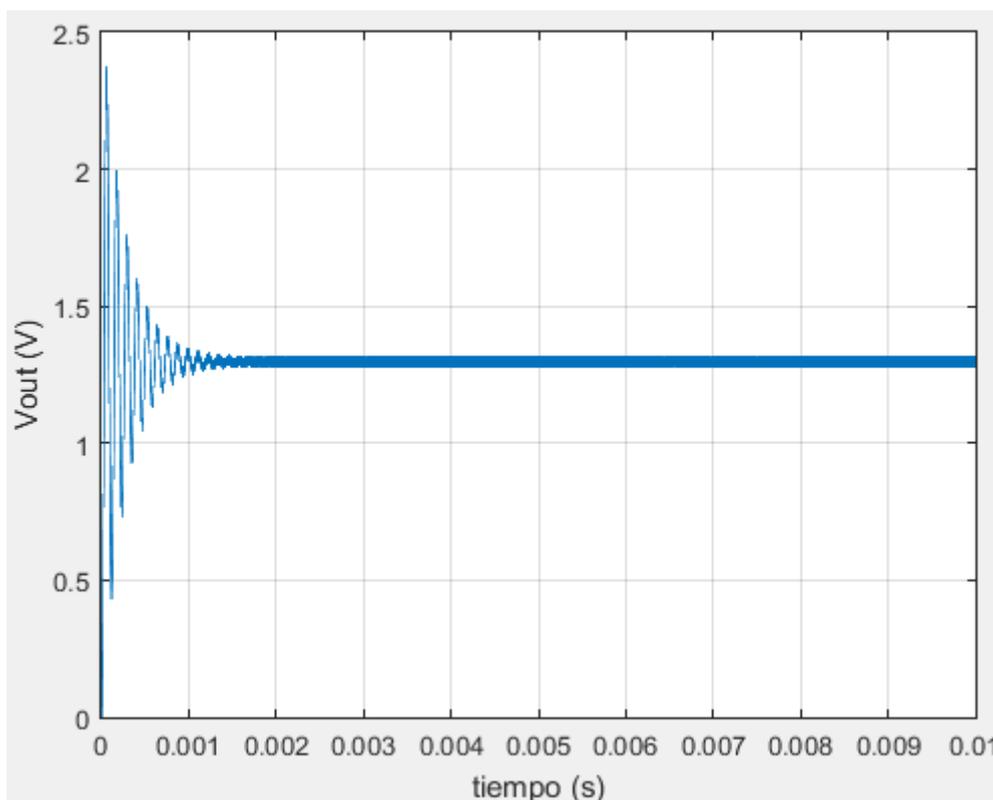


Figura 75. Resultados obtenidos de la tensión de salida en régimen transitorio.

6.2 Diseño del controlador

En primer lugar se analizará la función de transferencia del *buck* que es la que se desea controlar. A partir del diagrama de bloques visto en la figura 72 puede obtenerse la función de transferencia que corresponde a la expresión que se ve a continuación:

$$G(s) = \frac{\frac{V_i}{L \cdot C}}{s^2 + \frac{1}{R \cdot C} s + \frac{1}{L \cdot C}} \quad (51)$$

Sin embargo, debe tenerse en cuenta que el control del sistema se va a realizar digitalmente. Es decir, se muestrea la tensión de salida del *buck*, se introduce este valor digital al controlador y la salida de este vuelve a pasarse al dominio analógico a través de un mantenedor de orden cero (zoh). Al conjunto formado por el mantenedor de orden cero, la función de transferencia del sistema y el muestreador puede llamarse *Gzas*. El conjunto de este sistema puede verse en la figura 76.

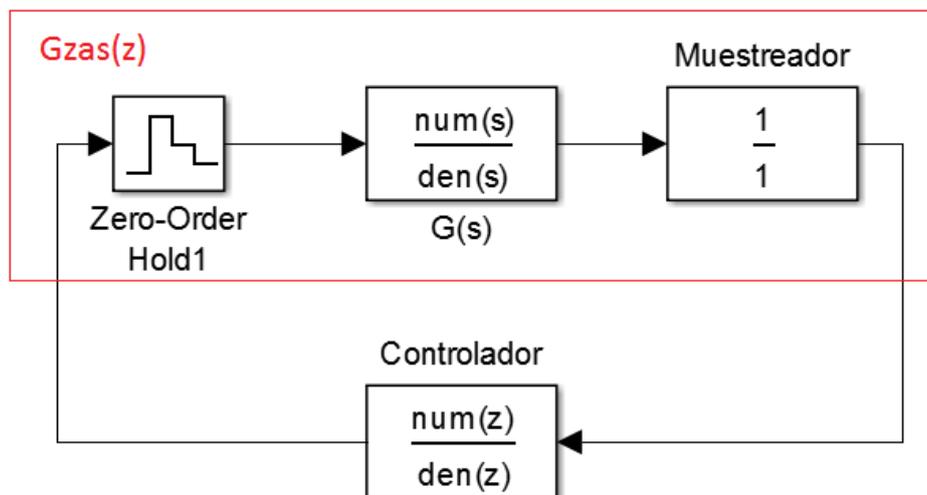


Figura 76. Diagrama típico de un control digital aplicando un zoh.

Así, para diseñar el controlador es necesario calcular la función de transferencia correspondiente al G_{zas} del sistema. La expresión correspondiente al G_{zas} es el que puede verse a continuación:

$$G_{zas}(z) = (1 - z^{-1}) \cdot Z \left\{ L^{-1} \left\{ \frac{G(s)}{s} \right\}^* \right\} \quad (52)$$

Para realizar este cálculo, se utilizará la función $g2d$ en Matlab, fijando un período de muestre de $20 \mu s$. De esta manera se obtienen los resultados que pueden verse a continuación:

$$G_{zas}(z) = \frac{2,533z + 2,406}{z^2 - 0,8745z + 0,8623} \quad (53)$$

Una vez obtenida la función correspondiente al G_{zas} del sistema, puede pasarse a diseñar el controlador. Para ello se ha utilizado la herramienta *sisotool* de Matlab, con la que se podrá visualizar el lugar geométrico de las raíces y la salida del sistema para diferentes valores del controlador.

En la figura 77 puede verse el lugar geométrico de las raíces del g_{zas} sin controlar. En esta figura puede observar como el sistema presenta dos polos, procedentes de la función de transferencia del convertidor, y un cero que lo añade el mantener de orden cero.

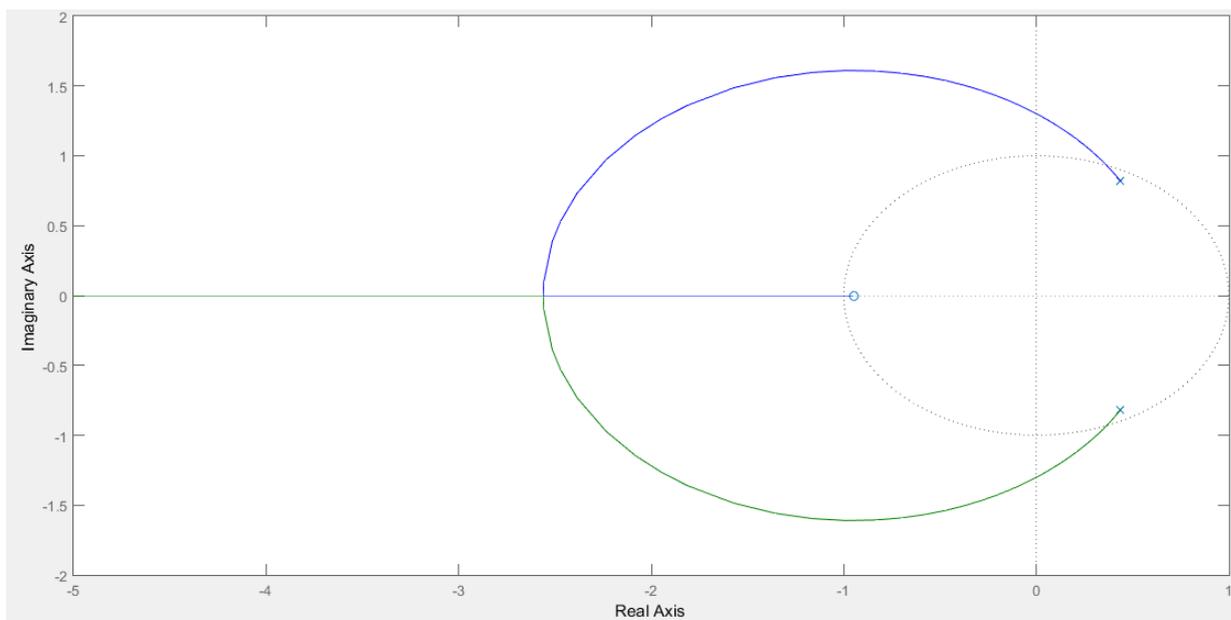


Figura 77. Lugar geométrico de las raíces de la planta de nuestro sistema al ser discretizada.

Para diseñar el controlador debe añadirse un serie de ceros y polos teniendo en cuenta diversos criterios:

- La función de transferencia del controlador deberá ser propia (deberá tener el mismo número de polos que de ceros) para poder asegurar que el valor de control va de 0 a 1, ya que este corresponderá al ciclo de trabajo.
- La oscilación del sistema no deberá ser muy elevada ya que si no podría tenerse una tensión de salida muy superior a la esperada.
- El tiempo de establecimiento deberá reducirse para obtener una respuesta lo más rápida posible.

Teniendo los criterios anteriores en cuenta, se han añadido dos polos reales y una pareja de ceros complejos, tal y como puede verse en la figura 78.

Type	Location	Damping	Frequency
Real Pole	0.993	1	354
Complex ...	0.49 +/- 0....	-0.0157	5.34e+04
Real Pole	0.0613	1	1.4e+05

Figura 78. Polos y ceros añadidos para el diseño del controlador.

De esta forma, el lugar geométrico de las raíces resultante es el que puede verse en la figura 79.

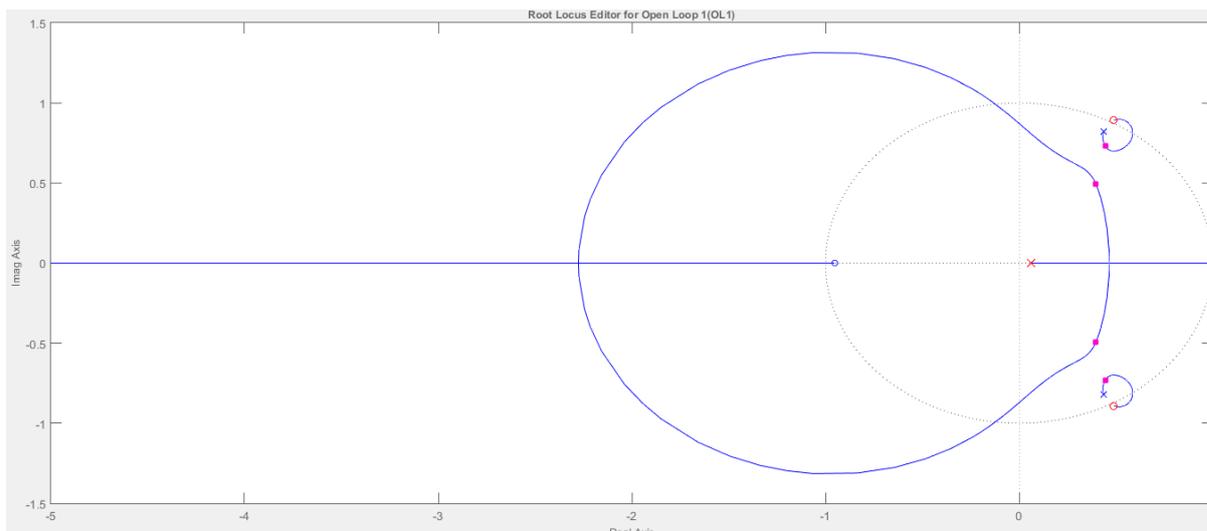


Figura 79. Lugar geométrico de las raíces del sistema una vez incluido el controlador.

Como también puede verse en la figura 79, se ha dado una cierta ganancia al controlador para desplazar los polos del sistema y hacer que este sea más rápido y amortiguado. En la figura 80 puede verse la respuesta del sistema a una perturbación escalón.

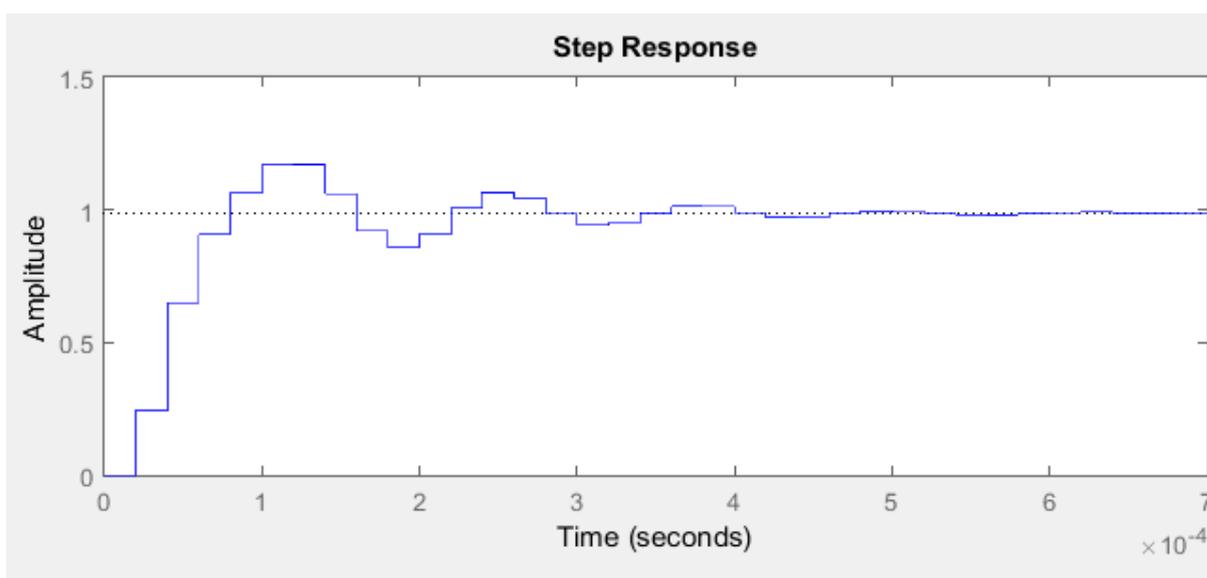


Figura 80. Respuesta del sistema a una perturbación en forma de escalón unitario.

Puede verse como el sistema se estabiliza en unos 400 μ s y no se tiene una oscilación demasiado elevada. En la figura 81 puede verse la señal de salida del controlador obtenida también a una perturbación de escalón unitario.

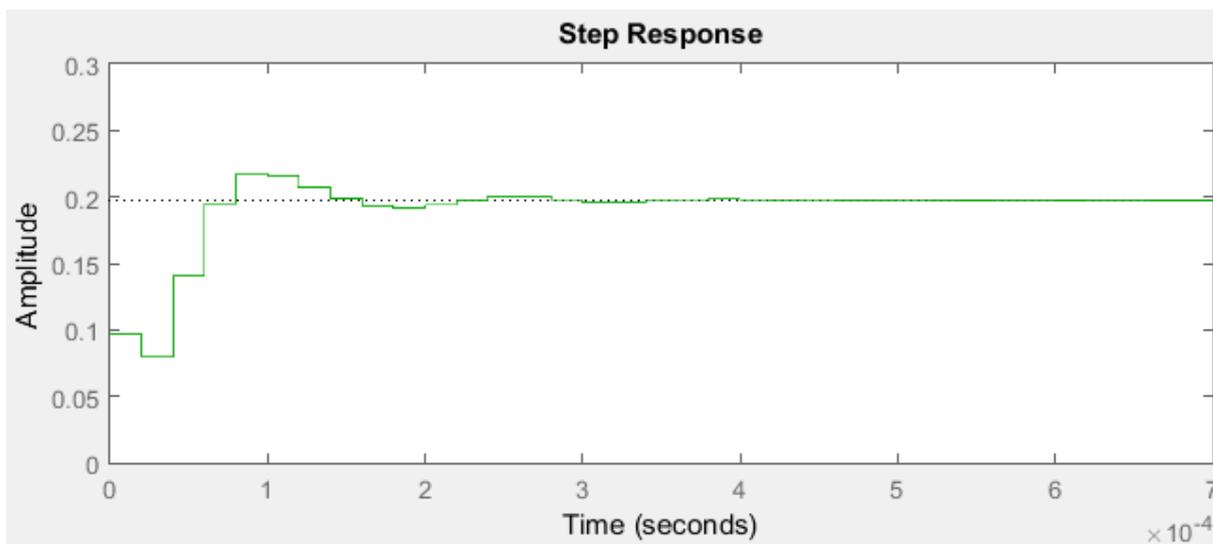


Figura 81. Respuesta de la salida del controlador a una perturbación en forma de escalón unitario.

Se observa como el valor de la salida de control se sitúa en todo momento entre 0 y 1. Por lo tanto puede determinarse que el controlador opera de la manera deseada y hace que el sistema trabaje de forma óptima.

Para continuar, se ha exportado el valor de este controlador diseñado. La función de transferencia correspondiente a este puede verse a continuación:

$$\frac{0,096669z^2 - 0,094658248z + 0,099955746}{z^2 - 1,05434z + 0,06091062} \quad (54)$$

Finalmente, en la figura 82 puede verse el diagrama de bloques final resultante del sistema.

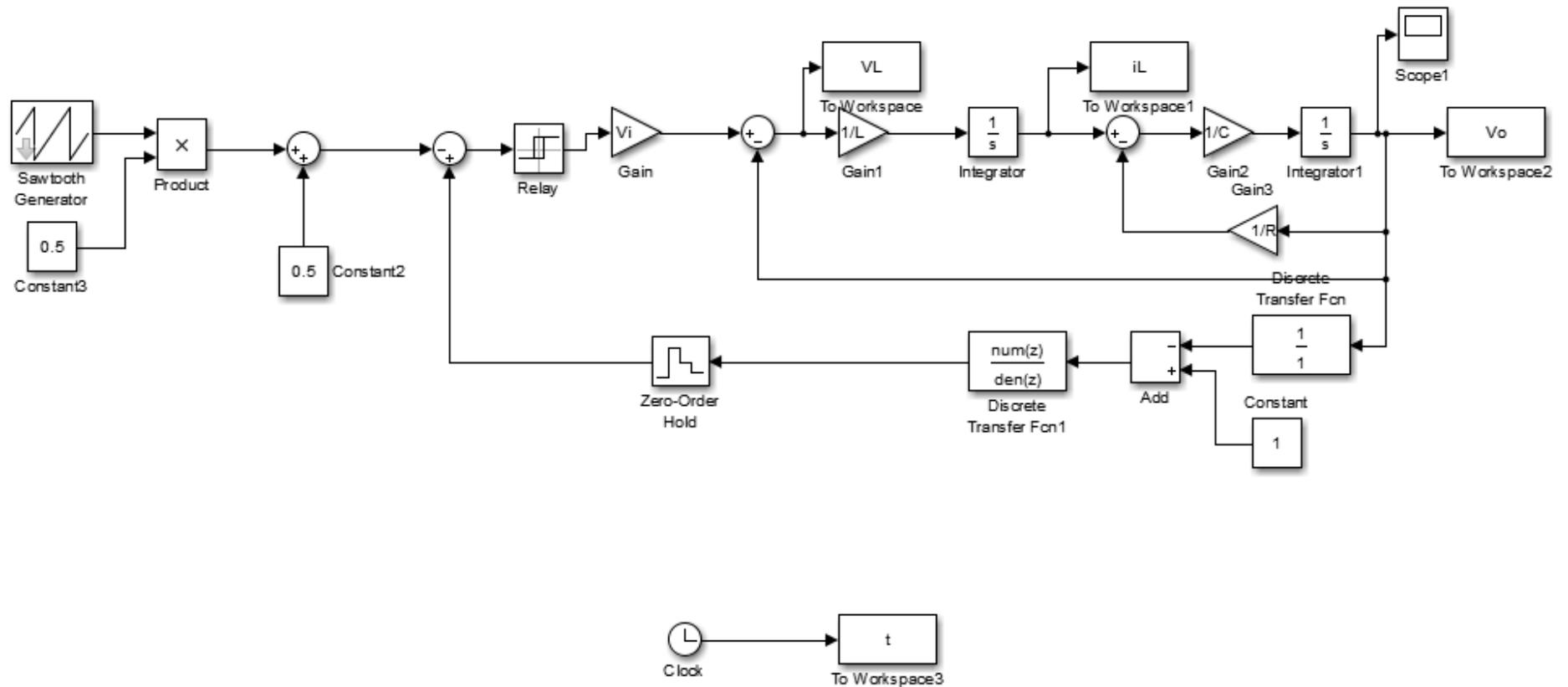


Figura 82. Diagrama de bloques del sistema planta + controlador.

Para poder aplicar el controlador diseñado al convertidor *buck* debe pasarse la función de transferencia de este en el dominio z en tiempo discreto. La función de transferencia obedece la estructura siguiente:

$$C(z) = \frac{b_2 z^{-2} + b_1 z^{-1} + b_0}{a_2 z^{-2} + a_1 z^{-1} + a_0} \quad (55)$$

Puede entenderse esta función de transferencia como la relación que existe entre la entrada y salida del controlador:

$$\frac{Y(z)}{X(z)} = \frac{b_2 z^{-2} + b_1 z^{-1} + b_0}{a_2 z^{-2} + a_1 z^{-1} + a_0} \quad (56)$$

$$Y(z)[a_2 z^{-2} + a_1 z^{-1} + a_0] = X(z)[b_2 z^{-2} + b_1 z^{-1} + b_0] \quad (57)$$

Ahora, al pasar a tiempo discreto se obtiene:

$$a_2 \cdot y(n-2) + a_1 \cdot y(n-1) + a_0 \cdot y(n) = b_2 \cdot x(n-2) + b_1 \cdot x(n-1) + b_0 \cdot x(n) \quad (58)$$

$$y(n) = \frac{b_2}{a_0} \cdot x(n-2) + \frac{b_1}{a_0} \cdot x(n-1) + \frac{b_0}{a_0} \cdot x(n) - \frac{a_1}{a_0} \cdot y(n-1) - \frac{a_2}{a_0} \cdot y(n-2) \quad (59)$$

Teniendo en cuenta la función de transferencia del controlador, el valor de las constantes del sistema sería:

$$\begin{aligned} b_2 &= 0,099955746 \\ b_1 &= -0,094658248 \\ b_0 &= 0,096669 \\ a_2 &= 0,06091062 \\ a_1 &= 1,05434 \\ a_0 &= 1 \end{aligned} \quad (60)$$

De esta manera, se ha obtenido el valor de las constantes que deberá tener el controlador digital al ser programado y volcado sobre la FPGA.

7 Conclusiones y trabajos futuros

Una vez finalizado este trabajo de fin de máster pueden llegarse a una serie de conclusiones.

En primer lugar, puede determinarse que hemos conseguido familiarizarnos y desenvolvemos con soltura en la utilización de los programas que han sido necesarios para la realización del trabajo, como son: PSIM, *Design Spark*, Matlab o ISE-Xilinx.

También puede observarse como se ha conseguido diseñar e implementar correctamente las dos PCBs necesarias para el acondicionamiento de la señal de salida del convertidor *buck*, obteniendo los resultados deseados.

Se ha realizado una programación para el control en lazo abierto en lenguaje VHDL de las señales que rigen el ciclo de trabajo y la variación de carga del *buck*.

Se ha conseguido visualizar las señales de salida del *buck* a través de la herramienta ChipScope, gracias al acondicionamiento de señal implementado y a la programación de toma de datos realizada.

Para finalizar, puede observarse cómo se ha realizado el diseño de un controlador digital eficiente a través del software Matlab gracias al cual la tensión de salida del *buck* se comporta de la manera deseada.

En cuanto a trabajos futuros, el siguiente paso a abordar sería la implantación del controlador diseñado en lenguaje VHDL y el volcado de éste sobre la FPGA donde se comprobaría el funcionamiento de todo el sistema en lazo cerrado. Si se obtuvieran los resultados deseados, se podría continuar con la aplicación de este algoritmo de control sobre sistemas VRMs

8 Referencias

- [1] A. Cánovas, R. Leyva. "Kit evaluador de leyes de control digital para VRMs". Universitat Rovira i Virgili. 2016.
- [2] Y. Panov, M. Jovanovic. "Design considerations for 12-V/1.5-V, 50-A Voltage Regulator Modules". IEEE Transactions on Power Electronics, VOL. 16, NO. 6, 2001
- [3] S. Roberts. "DC-DC Book of knowledge". Recom. 2014.
- [4] Franklin, G.F., Powell, D.J., and Workman, M.L., "Digital Control of Dynamic Systems" (3rd Edition), Prentice Hall, 1997.
- [5] Åström, K.J. and B. Wittenmark, "Computer-Controlled Systems: Theory and Design", Prentice-Hall, 1990, pp. 48-52.
- [6] [1] Clark, R. N. "Control System Dynamics", Cambridge University Press, 1996.
- [7] "Spartan-3E FPGA Starter Kit Board User Guide". Xilinx. 2011.
- [8] B. Carter. "Buffer Op Amp to ADC Circuit Collection". Texas Instruments. 2011.
- [9] "MAX12553/MAX12554/MAX12555 Evaluation Kits". Maxim. 2006.
- [10] "Data Sheet ADA4891-1/ADA4891-2/ADA4891-3/ADA4891-4". Analog Devices. 2015.
- [11] "MAX12555" Maxim. 2005.