

Trabajo Fin de Grado

Grado en Ingeniería de la Energía

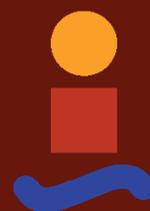
Herramientas de programación automática para sistemas de control de convertidores de potencia basados en DSP

Autor: Marta Muñoz Barrios

Tutor: Sergio Vázquez Pérez

Departamento de Ingeniería Electrónica
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla

Sevilla, 2019



Proyecto Fin de Carrera
Grado en Ingeniería de la Energía

**Herramientas de programación
automática para sistemas de control de
convertidores de potencia basados en DSP**

Autor:

Marta Muñoz Barrios

Tutor:

Sergio Vázquez Pérez

Profesor titular

Departamento de Ingeniería Electrónica
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla

Sevilla, 2019

Trabajo de Fin de Grado: Herramientas de programación automática para sistemas de control
de convertidores de potencia basados en DSP

Autor: Marta Muñoz Barrios

Tutor: Sergio Vázquez Pérez

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

Presidente:

Vocales:

Secretario:

Acuerdan otorgarle la calificación de:

Sevilla, 2019

El/la Secretario/a del Tribunal

A mi familia

A mis maestros

Agradecimientos

A mi tutor, Sergio, por todos sus aportes y enseñanzas.

Agradecer también a mis compañeros de trabajo por haberme apoyado durante la realización de este trabajo, y haberme facilitado la dedicación al mismo.

Marta Muñoz Barrios

Sevilla, 2019

Resumen

El presente Trabajo de Fin de Grado trata de la programación de dispositivos de Electrónica de Potencia mediante lenguaje de alto nivel. En concreto, describe la implementación del control de un Convertidor de Potencia de una planta fotovoltaica en un Procesador Digital de Señales (DSP) mediante lenguaje de bloques Simulink®.

Con este fin, se simulan un convertidor DC-DC elevador, así como una máquina de estados en esta plataforma, ejecutándose en pseudo-tiempo real en el Microcontrolador TMS230F28335.

Las pruebas realizadas sobre el programa se realizan tanto en SIL (Software in the Loop), PIL (Processor in the Loop) como External Mode. Finalmente, se analizan comparativamente los resultados de las distintas pruebas.

Abstract

The Project is an approach to high level language programming of Power Electronics devices. Specifically, it describes the implementation of a photovoltaic power converter controller in a Digital Signal Processor (DSP) using Simulink® blocks.

A DC-DC boost converter and a Mealy Machine are simulated in Simulink® environment. All these simulations are run in pseudo real time in the TMS230F28335 Microcontroller.

The program is tested either in SIL (Software in the Loop), PIL (Processor in the Loop) and External Mode. To conclude, results from the different tests are compared.

Índice

Agradecimientos	8
Resumen	9
Abstract	10
Índice	11
ÍNDICE DE TABLAS	13
ÍNDICE DE ECUACIONES	14
ÍNDICE DE FIGURAS	15
Capítulo 1	12
1.1 <i>Microcontroladores en plantas fotovoltaicas</i>	15
1.1.1 Funcionamiento del microcontrolador	16
1.2 <i>Hardware utilizado</i>	19
1.1.2 Microcontrolador f28335	19
1.3 <i>Software utilizado</i>	22
Capítulo 2	23
2.1 <i>Modelo del Convertidor Elevador</i>	24
2.1.1 Especificaciones de Diseño	27
2.1.2 Implementación en Simulink®	28
2.1.3 Mecanismo de precarga	33
2.2 <i>Algoritmo MTTP y Modelo de Control</i>	38
2.2.1 Pulse Width Modulation PWM	43
2.2.2 Transferencia de datos entre subsistemas	44
2.3 <i>Detalle global del modelo</i>	45
2.3.1 Controlador y mecanismo PWM	46
2.3.2 Máquina de estados	50
2.3.3 Convertidor Elevador	55
Capítulo 3	57
2.4 <i>Modos de ejecución</i>	57
2.4.1 Software in the Loop (SIL)	57
2.4.2 Processor in the Loop (PIL)	59
2.4.3 External Mode	65
2.5 <i>Implementación en equipo real</i>	65
Conclusiones	74

Anexo I

75

Bibliografia

82

ÍNDICE DE TABLAS

Tabla 1. Resumen características generación TMS320C28x [11].	20
Tabla 2. Ecuaciones del Convertidor Elevador [14].	25
Tabla 3. Parámetros de diseño del equipo real.	28
Tabla 4. Tipos de máquinas de estado de Matlab® [15].	36
Tabla 5. Diseño de los parámetros de control.	41
Tabla 6. Parámetros del bloque “Step”	46
Tabla 7. Tiempo de muestreo a la salida de los bloques “Rate Transition”	47
Tabla 8. Parámetros de “Discrete PI 1”.	49
Tabla 9. Parámetros de “Discrete PI 2”.	49
Tabla 10. Parámetros del bloque “Repeating Sequence”.	49
Tabla 11. Parámetros de los dispositivos tipo “IGBT/Diode”.	55
Tabla 12. Parámetros de los dispositivos tipo “Series RLC Branch”.	56

ÍNDICE DE ECUACIONES

Ecuación 1. Cálculo de la corriente por la bobina.	28
Ecuación 2. Cálculo del Duty Cycle.	42

ÍNDICE DE FIGURAS

Ilustración 1. Evolución de la demanda eléctrica anual. REE	12
Ilustración 2. Evolución de la producción energética bruta mundial. (IEA.org)	13
Ilustración 3. Evolución de las emisiones de CO ₂ y cobertura de la demanda de energía según REE en 2017.	13
Ilustración 4. Esquema de comunicación entre los módulos del microcontrolador [7].	17
Ilustración 5. Mapa de Pines del Datasheet del f2833x [11].	21
Ilustración 6. Diagrama de Bloques de un Convertidor DC-DC	23
Ilustración 7. Circuito del Convertidor Boost [14].	24
Ilustración 8. Cronogramas MCC [14].	26
Ilustración 9. Circuito del convertidor en MCC [14].	26
Ilustración 10. Cronogramas MCD [14].	27
Ilustración 11. Circuito del convertidor en MCD [14].	27
Ilustración 12. Modelo del Convertidor en Simulink®.	29
Ilustración 13. Tensión e intensidad a la entrada del Convertidor.	30
Ilustración 14. Señal ampliada de la corriente de la bobina I_L .	31
Ilustración 15. Tiempo de carga.	31
Ilustración 16. Tensión e intensidad en la carga.	32
Ilustración 17. Potencias a la entrada y la salida del convertidor.	33
Ilustración 18. Topología del convertidor Boost con mecanismo de precarga.	34
Ilustración 19. Diagrama de estados de la máquina.	35
Ilustración 20. Estado de los interruptores K_1, K_2 y K_3 para $R_{precarga} = 22,5 \Omega$.	37
Ilustración 21. Modelo de convertidor en Simulink® con precarga.	38
Ilustración 22. Relación Potencia-Tensión en función de la irradiancia [16].	38
Ilustración 23. Tensión de referencia del controlador.	39
Ilustración 24. Diagrama de flujo control MPPT [16].	40
Ilustración 25. Esquema de control.	41

Ilustración 26. Cálculo del Duty Cycle del controlador.	42
Ilustración 27. Zoom de la señal de Duty Cycle.	43
Ilustración 28. Pulsos de modulación del convertidor y Duty Cycle (rojo).	44
Ilustración 29. Bloque Rate Transition	44
Ilustración 30. Información de tiempo de muestreo.	45
Ilustración 31. Conexión entre el subsistema del controlador y del PWM.	46
Ilustración 32. Señales de entrada y salida del controlador.	47
Ilustración 33. Cálculo de z .	48
Ilustración 34. Cálculo de z^* .	48
Ilustración 35. Modelo Simulink® del controlador.	49
Ilustración 36. Programación mediante bloques del PWM.	50
Ilustración 37. Topología del interruptor manual para la activación de los pulsadores P, B y RESET.	50
Ilustración 38. Entradas y salidas de la máquina de estados.	51
Ilustración 39. Localización del bloque “Estado” en la barra de herramientas de Stateflow®.	52
Ilustración 40. Configuración de uno de los estados de la máquina.	53
Ilustración 41. Máquina de estado con Stateflow®.	54
Ilustración 42. Fuente generadora de tensión continua.	55
Ilustración 43. Tensión a la salida V_0 .	58
Ilustración 44. Intensidad a la salida I_0	58
Ilustración 45. Señal DS (duty cycle) salida del controlador.	59
Ilustración 46. Máscara de Bloque PIL.	60
Ilustración 47. Ejemplo de uso del software para verificación y validación de código mediante PIL [16].	61
Ilustración 48. Señal diferencial entre salida del control simulado y bloque PIL.	62
Ilustración 49. Tensión en la carga V_0 .	63
Ilustración 50. Corriente en la carga I_0 .	63
Ilustración 51. Señal DS (duty cycle) salida del controlador.	64

Ilustración 52. Diferencia entre el bloque de control SIL y PIL.	64
Ilustración 53. Tarea de Idle.	66
Ilustración 54. Número de PIE [15].	67
Ilustración 55. Configuración del bloque de Interrupción.	68
Ilustración 56. Concepto de CMPA del PWM.	69
Ilustración 57. Configuración general del PWM.	69
Ilustración 58. Configuración de la señal de salida del PWM.	70
Ilustración 59. Configuración del control del ADC.	70
Ilustración 60. Configuración de la entrada del ADC.	71
Ilustración 61. Configuración de la Tarea de Idle.	71
Ilustración 62. Cronograma representativo de los Idle Time, Interrupciones, Tiempo de Muestreo [16].	72
Ilustración 63. Sincronización ADC-PWM con bloque ADC Interrupt [21].	73

Capítulo 1

Introducción

La actual crisis de recursos, así como la creciente concienciación social y gubernamental entorno al cambio climático y la gran generación de residuos, ponen la investigación en energías renovables y estrategias de ahorro y eficiencia en el centro del debate científico y tecnológico.

La producción de energía va destinada de forma mayoritaria a la generación de energía eléctrica, suponiendo aproximadamente un 30% del consumo. Según el informe anual del Sistema Eléctrico Español elaborado por Red Eléctrica, en la actualidad se observa una tendencia al alza, siendo la demanda de energía eléctrica un 0,4% mayor que la de 2017. Esto continúa con la tendencia alcista de los cuatro años anteriores (1,2% en 2017) [1].

Evolución de la demanda eléctrica peninsular en los últimos 10 años [TWh]

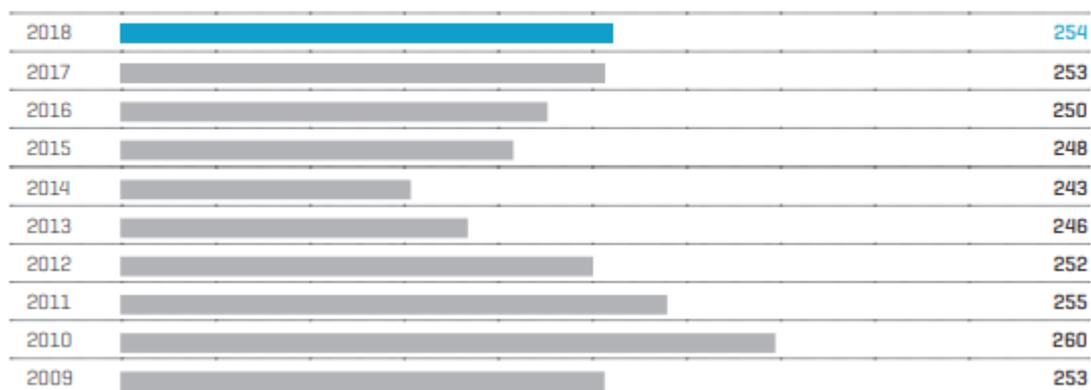


Ilustración 1. Evolución de la demanda eléctrica anual. REE

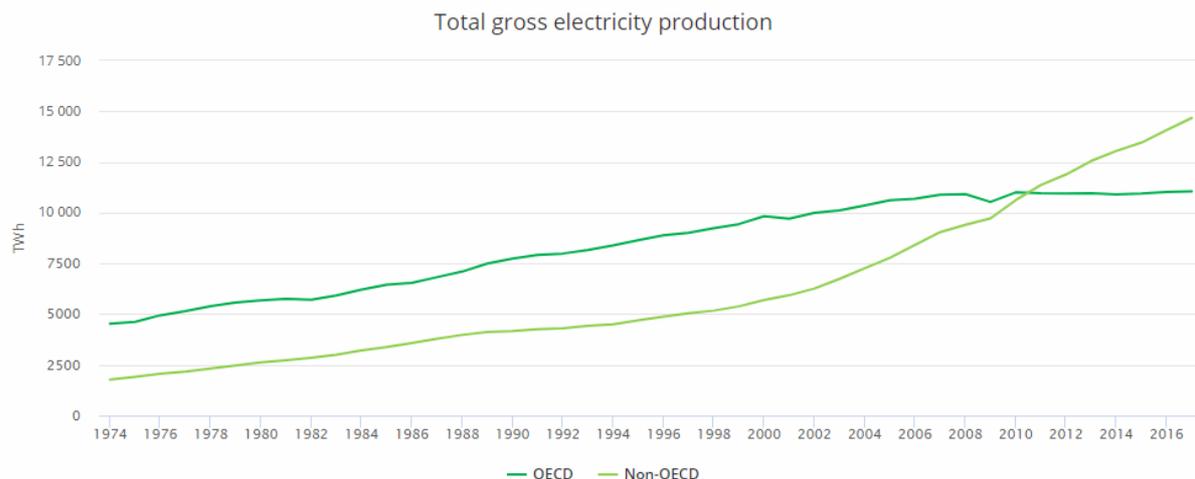


Ilustración 2. Evolución de la producción energética bruta mundial. (IEA.org)

En relación con el impacto medioambiental de esta producción, por parte de las administraciones, se establece la prioridad de reducir las emisiones de CO₂, siendo estas últimas objeto de monitorización exhaustiva por parte de entidades públicas y privadas. Como se muestra en la Ilustración 3, en el último año se ha conseguido reducir el volumen de emisiones, debido a una mayor cobertura de la demanda por parte de las energías renovables.

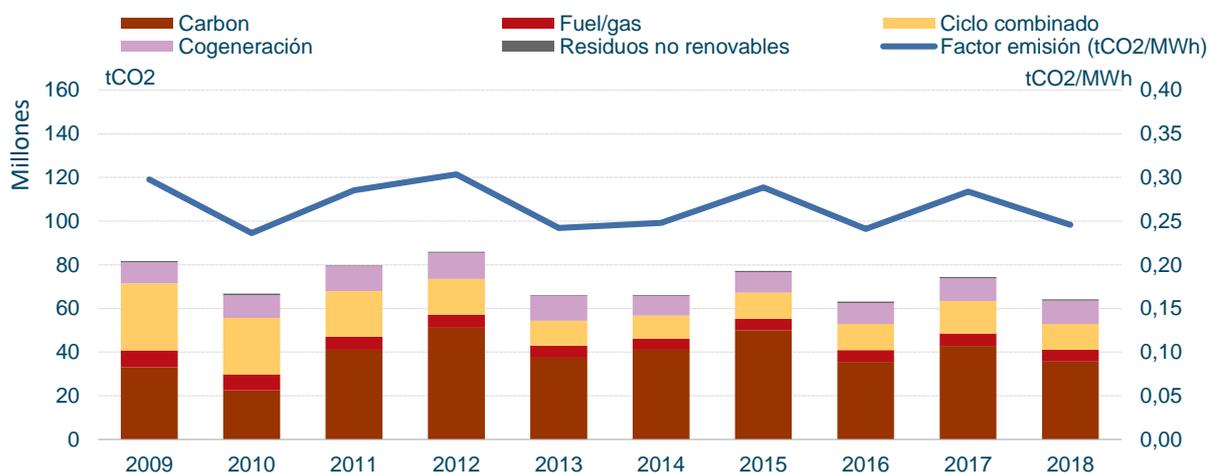


Ilustración 3. Evolución de las emisiones de CO₂ y cobertura de la demanda de energía según REE en 2017.

Teniendo en cuenta el contexto descrito, los principales retos en materia de energía pasan por el desarrollo y maduración de las tecnologías de generación renovables, así como por la continua mejora de la eficiencia en la generación y transporte [2, 1].

Esta apuesta por mejorar la eficiencia de los procesos de generación de energía renovable y hacerla competitiva en el mercado, sitúa a la electrónica de potencia como una de las principales herramientas tecnológicas para la reducción de pérdidas energéticas en la mayoría de procesos de generación, transporte y almacenamiento [3].

En concreto, en cuanto a la aplicación principal de la electrónica de potencia en la generación de energía renovable, esta cumple la misión de transformar la energía eléctrica suministrada en el origen en corriente eléctrica adecuada para las redes de distribución, haciendo el transporte lo más eficiente posible. En concreto, en la producción fotovoltaica, la energía procedente de los paneles debe pasar por una serie de dispositivos que ajusten su frecuencia y tensión a los de la red de distribución [3].

El dispositivo que se va a analizar se trata en concreto de un convertidor elevador, presente en una planta de energía fotovoltaica, que pretende abordar la optimización del transporte de energía eléctrica entre el origen y la carga mediante un mecanismo de control que modelará las conmutaciones de los interruptores.

Mediante el uso de un dispositivo microcontrolador para ejecutar el control descrito de las conmutaciones, se pretende abordar esta tarea de la forma más óptima posible.

Los microcontroladores son un tipo de sistema integrado programable para realizar una serie de tareas, con el objetivo de controlar dispositivos electrónicos. Su uso está ampliamente extendido en cuanto a aplicaciones en la industria, y ha dejado obsoletos otros dispositivos de control.

La principal ventaja de estos dispositivos reside en la optimización de recursos, ya que se adapta a las necesidades de almacenamiento y tamaño de la aplicación que ocupa. Otras de sus características ventajosas son:

- Son programables, lo cual aporta flexibilidad en el desarrollo de proyectos complejos y prototipos.
- Fiabilidad, al tratarse de un único circuito integrado, reduce el riesgo de fallo de alguno de los elementos con respecto a dispositivos compuestos por relés.
- Menor consumo energético que los de dispositivos convencionales, compuestos de numerosos dispositivos electrónicos conectados entre sí.

Las desventajas del uso de esta tecnología en la industria son:

- La complejidad de programación de estos dispositivos, así como el gran número de lenguajes que la posibilitan, por lo que se necesita personal altamente cualificado [4].

1.1 Microcontroladores en plantas fotovoltaicas

La evolución de la tecnología de generación renovable en los últimos años la ha llevado a posicionarse como una alternativa rentable a la tradicional. En concreto, la energía fotovoltaica comienza a competir en el mercado liberalizado de la energía, reportando generosos beneficios financieros en recientes proyectos.

La energía solar fotovoltaica se trata de una fuente de energía renovable, que produce energía a partir de la radiación solar incidente sobre sus células. Estas células de material semiconductor, absorben la radiación solar, liberando una corriente de electrones de sus átomos. Esta fuente de energía además de renovable, limpia e inagotable, se trata de una inversión cada vez más lucrativa.

A medida que la tecnología fotovoltaica ha madurado, los costes de producción han experimentado una considerable reducción. Un ejemplo de ello, es la sustancial caída de costes de compraventa de proyectos fotovoltaicos de hasta un 70% desde el año 2009, según fuentes del diario *The Economist*. En concreto, en 2016 llegaban a alcanzar los 30 \$/MWh [5].

Sin embargo, a medida que esta tecnología se vuelve competitiva, más inversores se suman a la tendencia, reduciendo esto los beneficios de los competidores. En este panorama, el mercado requiere la maduración de tecnologías que permitan ahorrar los costes de generación y distribución, optimizando el aprovechamiento de los recursos, así como minimizando las pérdidas en el transporte, como forma de aumentar los márgenes, así como de independizarse de las impredecibles regulaciones administrativas.

El desarrollo de los sistemas electrónicos de potencia se sitúa como una de las principales vías de optimización de los procesos de producción y transporte. Sin embargo, la alta necesidad de especialización a la hora de trabajar con dispositivos electrónicos es una de las limitaciones que existen en la actualidad.

Para la optimización de la producción de las plantas fotovoltaicas, es decir, máximo aprovechamiento del recurso solar, se emplean técnicas de seguimiento del punto de máxima potencia (Maximum Power Point Tracking or MPPT techniques). Este punto máximo puede ser alcanzado mediante diferentes algoritmos tales como perturbación observación o conductancia incremental, entre los más populares [6]. Este consiste, a grandes rasgos, en la obtención de un patrón de conmutación de los dispositivos semiconductores para la obtención de la máxima potencia de salida.

1.1.1 Funcionamiento del microcontrolador

La historia de los microcontroladores comienza en el año 1971, con el primer microprocesador en salir al mercado, el Intel 4004. Éste fue inicialmente utilizado en calculadoras, las Busicom 141-PF.

Sin embargo, no fue hasta 1974 cuando se comercializó el primer dispositivo considerado como microcontrolador, el TMS 1000, entendiéndose este como un dispositivo integrado consistente en el procesador, módulo RAM, ROM y entradas y salidas digitales [7] [4].

El uso de los microcontroladores se ha extendido rápidamente en múltiples aplicaciones:

- Domésticas. Presentes en electrodomésticos (cafeteras, secadoras...)
- Teléfonos móviles, smartphones.
- Automóviles.
- Industria aeroespacial.
- Climatización.
- Industria energética.

Los microcontroladores son, por definición, dispositivos digitales, es decir, únicamente pueden interpretar información discreta.

Puesto que la información que proviene de los procesos reales es continua, para el establecimiento de una comunicación entre un dispositivo digital y la realidad, ésta última debe ser discretizada. Con este fin, los microcontroladores poseen un módulo de conversión analógico digital (ADC) [8].

El esquema de la Ilustración 4, modela el funcionamiento del microcontrolador en base a la comunicación entre sus distintos módulos, así como dispositivos externos.

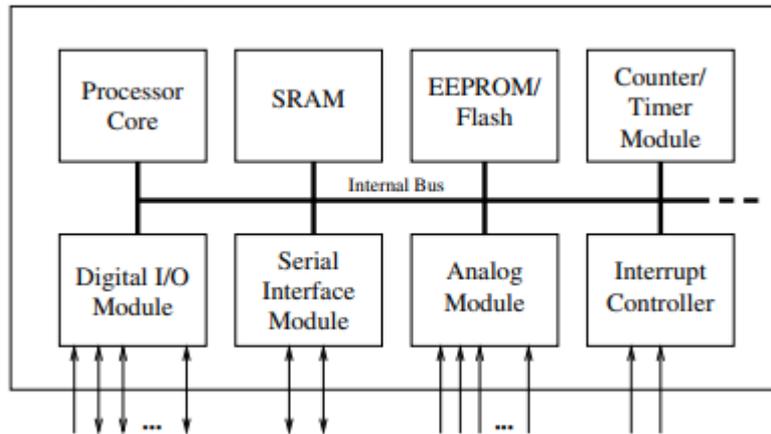


Ilustración 4. Esquema de comunicación entre los módulos del microcontrolador [7].

La estructura genérica de módulos que integran estos dispositivos, son los siguientes:

- Procesador (Core Processor Unit CPU). Esta unidad ejecuta las instrucciones presentes en el programa.
- Memoria. Existiendo varios tipos de memoria según sus funciones y durabilidad.
- Controlador de interrupciones. Gestiona las interrupciones del programa principal, bien procedentes de eventos externos o internos.
- Contador/Timer. Además, pueden incluir salidas PWM.
- E/S Digitales. Puertos paralelos de entrada y salida. El número de pines variará en función de la familia y tipo del controlador.
- Interfaces. Para comunicar con el PC para el desarrollo, así como con dispositivos periféricos.
- Watchdog. Garantiza la seguridad del proceso, al detectar fallos catastróficos en el funcionamiento del programa y, en consecuencia, realizar el reseteo del dispositivo.

En cuanto a la elección del modelo de microcontrolador a utilizar depende de múltiples factores relativas, principalmente a la aplicación del mismo:

- Herramientas de desarrollo del programa. La disposición de herramientas específicas para el modelo de microcontrolador a utilizar agiliza el proceso de desarrollo y puede suponer un ahorro de recursos.
- Conocimientos y especialización del personal técnico. Asimismo, la disposición de personal conocedor de la tecnología a utilizar, así como la existencia de una base sólida

de conocimientos sobre esta incurrirá en un ahorro de tiempo y costes de implementación considerable.

- Desarrollo de la familia de dispositivos y solidez del fabricante. Esto puede garantizar la provisión a futuro de dispositivos compatibles, así como una mejor integración con otros dispositivos.
- Cantidad de memoria y periféricos. Siendo este requisito un limitante en cuanto a la selección, ya que se debe garantizar la correcta consecución del objetivo definido.

Finalmente, se deberá realizar un análisis de la rentabilidad teniendo en cuenta todas las variables que afecten al proceso, tanto de tiempo como de consumo de recursos humanos y materiales para una correcta evaluación del modelo de dispositivo.

Algunos de los fabricantes de microcontroladores más comercializados son [9]:

- Texas instruments:
 - MSP430 (16 bit).
 - F2xxx (32 bit).
- Freescale
 - HCS08 (8 bit).
 - Coldfire (32 bit).
- Renesas
 - 740, 78K0S (8 bit).
 - R32C, RX (32 bit).

1.2 Hardware utilizado

Para el desarrollo del presente Trabajo de Fin de Grado, se ha seleccionado el Kit de Experimentación C2000 de la familia TMS320F28x, con microcontrolador tipo F28x, concretamente el modelo F28335.

Esta selección responde a la sencillez y rapidez de integración de estos dispositivos, así como a la disponibilidad de herramientas de software específicas para el lenguaje que se va a utilizar para su programación.

Asimismo, se trata de un dispositivo económico y robusto, que permite su operación en múltiples entornos de experimentación.

La estación de conexión compatible con el microcontrolador dota de acceso a las señales de GPIO y ADC. Además, contiene una serie numerada de pines para la comunicación bidireccional entre el DSP (Digital Signal Processor) y los distintos periféricos.

1.1.2 Microcontrolador f28335

Los microcontroladores C2000 de Texas Instruments son dispositivos de 32 bits diseñados para aplicaciones de control en tiempo real. constan de un núcleo optimizado que ofrece al sistema eficiencia, flexibilidad y fiabilidad. Las herramientas de desarrollo y software de control cumplen el objetivo de maximizar la usabilidad y minimizar el tiempo de desarrollo [10]. Las principales aplicaciones de control en tiempo real son:

- Control de motores digitales.
- Sistemas de energía renovable.
- Conversión de potencia digital.
- Sistemas de iluminación adaptables.
- Aplicaciones automovilísticas.

El microcontrolador es de la generación TMS320C28x, la cual incluye la familia de microcontroladores Delfino, que son dispositivos altamente integrados, soluciones de alto rendimiento para aplicaciones de control exigentes de la industria. A continuación, en la Tabla 1, se aporta un resumen de las características de los dispositivos pertenecientes a dicha generación.

FEATURE		F28335 (150 MHz)	F28334 (150 MHz)	F28332 (100 MHz)
Instruction cycle		6.67 ns	6.67 ns	10 ns
Floating-point Unit		Yes	Yes	Yes
3.3-V on-chip flash (16-bit word)		256K	128K	64K
Single-access RAM (SARAM) (16-bit word)		34K	34K	26K
Code security for on-chip flash/SARAM/OTP blocks		Yes	Yes	Yes
Boot ROM (8K X16)		Yes	Yes	Yes
One-time programmable (OTP) ROM (16-bit word)		1K	1K	1K
6-channel Direct Memory Access (DMA)		Yes	Yes	Yes
PWM outputs		ePWM1/2/3/4/5/6	ePWM1/2/3/4/5/6	ePWM1/2/3/4/5/6
HRPWM channels		ePWM1A/2A/3A/4A/5A/6A	ePWM1A/2A/3A/4A/5A/6A	ePWM1A/2A/3A/4A
32-bit Capture inputs or auxiliary PWM outputs		6	6	4
32-bit QEP channels (four inputs/channel)		2	2	2
Watchdog timer		Yes	Yes	Yes
12-Bit ADC	No. of channels	16	16	16
	MSPS	12.5	12.5	12.5
	Conversion time	80 ns	80 ns	80 ns
32-Bit CPU timers		3	3	3
Multichannel Buffered Serial Port (McBSP)/SPI		2	2	1
Serial Peripheral Interface (SPI)		1	1	1
Serial Communications Interface (SCI)		3	3	2
Enhanced Controller Area Network (eCAN)		2	2	2
Inter-Integrated Circuit (I2C)		1	1	1
Digital I/O pins (shared)		88	88	88
External interrupts		8	8	8
Packaging	100-Pin PGF	Yes	Yes	Yes
	100-Ball ZHH	Yes	Yes	Yes
Temperature options	A: -40°C to 85°C	(PGF, ZHH)	(PGF, ZHH)	(PGF, ZHH)
Product status		TMX	TMX	TMX

Tabla 1. Resumen características generación TMS320C28x [11].

La asignación de los 176 pines se muestra en la Ilustración 5:

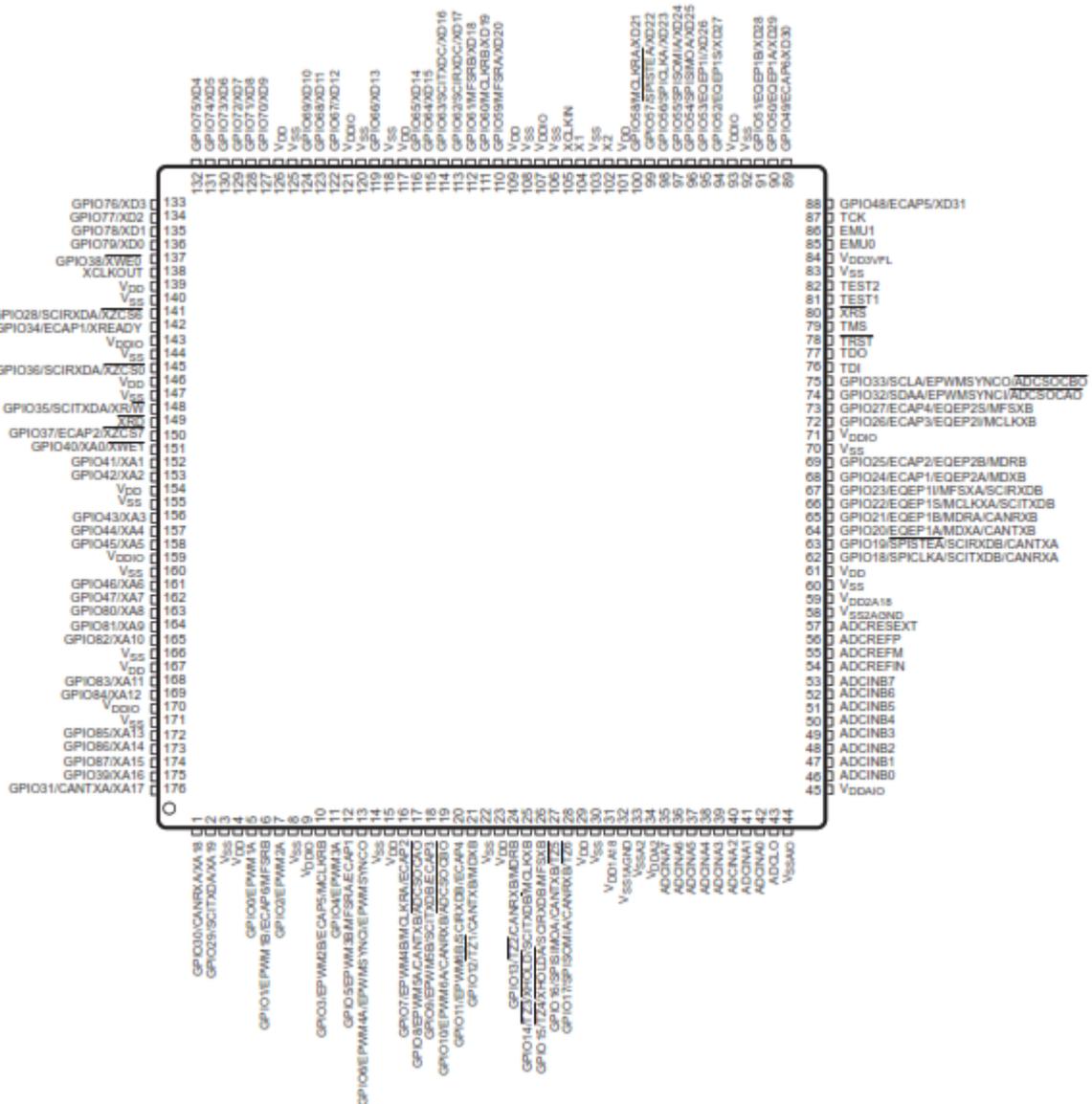


Ilustración 5. Mapa de Pines del Datasheet del f2833x [11].

1.3 Software utilizado

Para la programación y compilación de los diferentes módulos del proyecto, se han utilizado el siguiente software:

- Matlab® 2017a. Se recomienda utilizar una versión por encima de la R2015b, de forma que no haya problemas de compatibilidad con otros programas. Dentro de este entorno, se utilizará:
 - Simulink. Utilizará las siguientes funcionalidades añadidas [11]:
 - 1) MATLAB Coder™. Herramienta necesaria para la compatibilidad de lenguaje de alto nivel MATLAB® y lenguaje de bajo nivel C y C++ que deberá recibir el microcontrolador.
 - 2) Simulink Coder™. Herramienta necesaria para la compatibilidad de lenguaje de alto nivel Simulink® con el que trabajará el usuario y lenguaje de bajo nivel C y C++ que deberá recibir el microcontrolador. A partir del modelo creado, genera una serie de archivos interpretables por el dispositivo.
 - 3) Embedded Coder® Support Package for Texas Instruments™ C2000™ Processors. Se trata de un paquete de herramientas específico para los dispositivos de TI, permite la ejecución de los programas en tiempo real.
 - 4) Stateflow®. Esta librería aporta un lenguaje de bloques en el entorno Simulink® para la construcción de máquinas de estados y diagramas de flujo. Es compatible con las herramientas previamente enumeradas en los puntos 1), 2) y 3).
 - 5) Simscape™. Se utiliza esta librería para disponer de los elementos necesarios para simular circuitos y componentes eléctricos.
- CCS v.6. Versión del compilador ti-cgt-c2000_15.12.7.LTS. En el Anexo se describirá la configuración específica para el dispositivo TMS320F28335.
- ControlSUITE TI.

Capítulo 2

Modelo del Sistema

En términos muy generales, la Electrónica de Potencia se encarga de la transformación y control de las características del flujo energético [12].

En concreto, los convertidores de potencia transforman, por medio de dispositivos compuestos por semiconductores, este flujo energético entre dos formas de presentación del mismo, en función de las cuales se establecen cuatro tipos principales: [13].

- DC-DC.
- DC-AC (inversores).
- AC-DC (rectificadores).
- AC-AC.

Los convertidores DC-DC cumplen la misión de adaptar la entrada de tensión continua a unos parámetros específicos de tensión del mismo tipo a la salida con la que se desea alimentar una determinada carga.

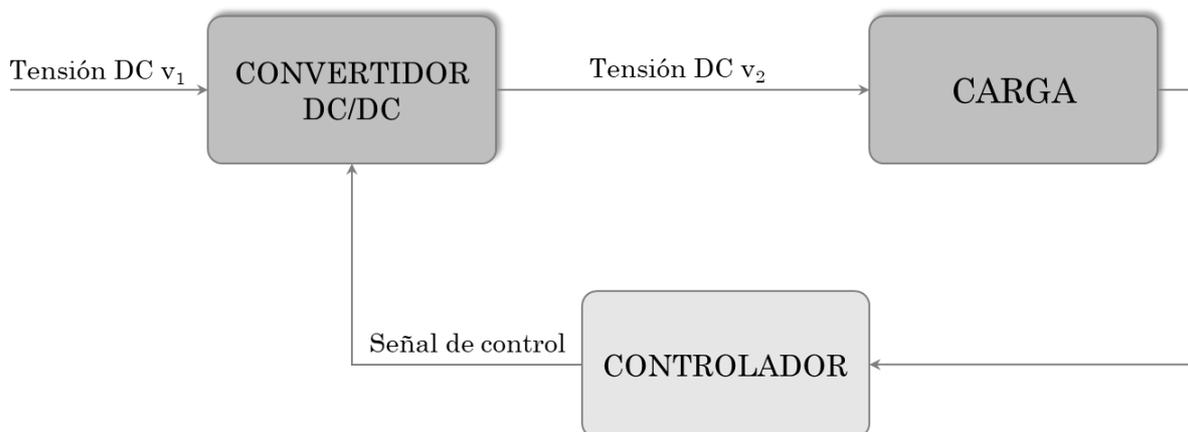


Ilustración 6. Diagrama de Bloques de un Convertidor DC-DC

Los convertidores DC-DC asimismo poseen múltiples configuraciones:

- Convertidor elevador (Boost).
- Convertidor reductor (Buck).

- Convertidor reductor-elevador (Buck-Boost).
- Convertidor Flyback.

2.1 Modelo del Convertidor Elevador

El Convertidor Elevador o tipo Boost es un convertidor de potencia DC-DC que eleva la tensión media a la salida con respecto a la entrada mediante el uso en dos modos de conducción (MCC y MCD) de un elemento inductivo (bobina en la etapa de entrada) y un elemento capacitivo (condensador en la etapa de salida) [12].

Esquema básico de un convertidor tipo Boost:

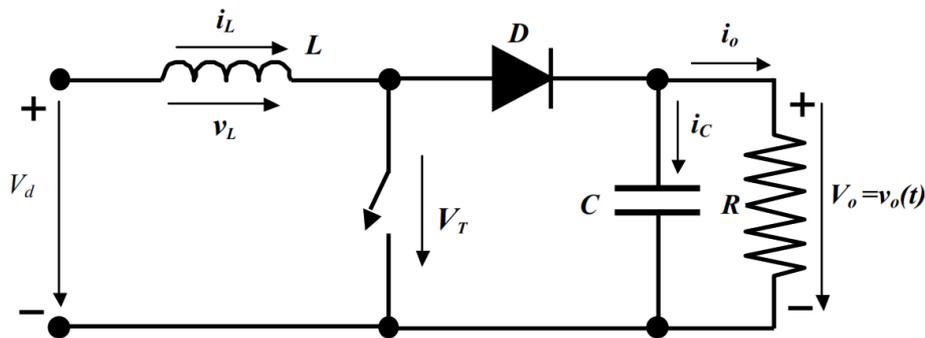


Ilustración 7. Circuito del Convertidor Boost [14].

Su funcionamiento se resume de la siguiente forma [12]:

- MCC: El interruptor se encuentra conduciendo, por tanto, el elemento inductor L almacena energía, mientras el diodo está polarizado inversamente, aislando la carga R , que será alimentada por el elemento capacitivo C , que se descarga.
- MCD: El interruptor se abre, de forma que la corriente circula a través del diodo, alimentando la carga R y el elemento capacitivo C .

Estas dos situaciones se denominan respectivamente Modo de Conducción Continua (MCC) y Modo de Conducción discontinua, y se producen durante los periodos t_{on} y t_{off} , respectivamente. Sus esquemas de funcionamiento se muestran a continuación en la Tabla 2.

Los parámetros de diseño del convertidor según el modo de conducción serán los mostrados en la Tabla 2.

Parámetro	MCC	MCD
$\frac{V_o}{V_d}$	$\frac{1}{1-D}$	$\frac{D + \Delta_1}{\Delta_1}$
D	$\frac{V_o - V_d}{V_o}$	$\sqrt{\frac{4 V_o}{27 V_d} \left(\frac{V_o}{V_d} - 1\right) \frac{I_o}{I_{oBmax}}}$
I_L	$\frac{I_o}{1-D}$	$\frac{T_s V_d}{2L} D(D + \Delta_1)$
I_{L1}	$I_L + \frac{\Delta I_L}{2}$	$\frac{V_d}{L} D T_s = \frac{(V_o - V_d)}{L} \Delta_1 T_s$
Δ_1	----	$\frac{V_d}{V_o - V_d} D$
$\Delta I_L = I_{L1} - I_{L0}$	$\frac{V_d}{L} D T_s = \frac{V_o T_s}{L} D(1-D)$	I_{L1}
$\frac{\Delta V_o}{V_o}$	$\frac{D T_s}{RC} = \frac{I_o D T_s}{V_o C}$	$\frac{T_s (I_{L1} - I_o)^2 D V_d}{2 C I_{L1} (V_o - V_d) V_o}$
	$\frac{V_o}{V_d} = \frac{I_d}{I_o}$	$\frac{V_o}{V_d} = \frac{I_d}{I_o}$
	$I_o = I_L(1-D)$	$I_o = \frac{\Delta_1}{D + \Delta_1} I_L$
I_{LB}	$\frac{T_s V_o}{2L} D(1-D)$	
I_{oB}	$\frac{T_s V_o}{2L} D(1-D)^2 = \frac{27}{4} D(1-D)^2 I_{oBmax}$	
I_{oBmax}	$\frac{2 T_s V_o}{27 L}$	

Tabla 2. Ecuaciones del Convertidor Elevador [14].

A continuación, los cronogramas de las Ilustraciones 9 y 10 muestran la evolución de los parámetros de funcionamiento de los convertidores Boost según su modo de conducción.

a) Modo de Conducción Continua

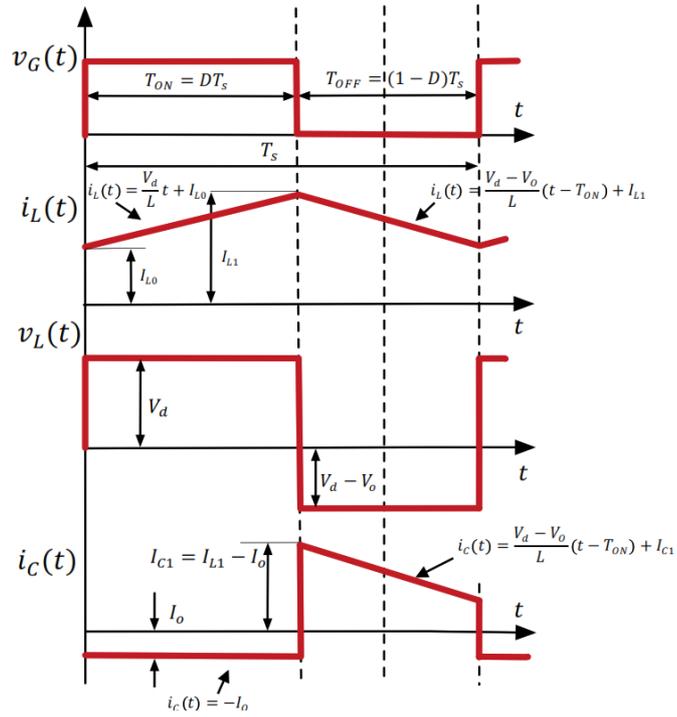


Ilustración 8. Cronogramas MCC [14].

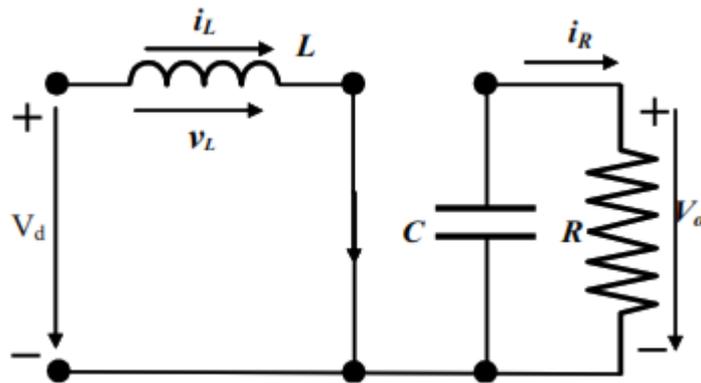


Ilustración 9. Circuito del convertidor en MCC [14].

b) Modo de Conducción Discontinua

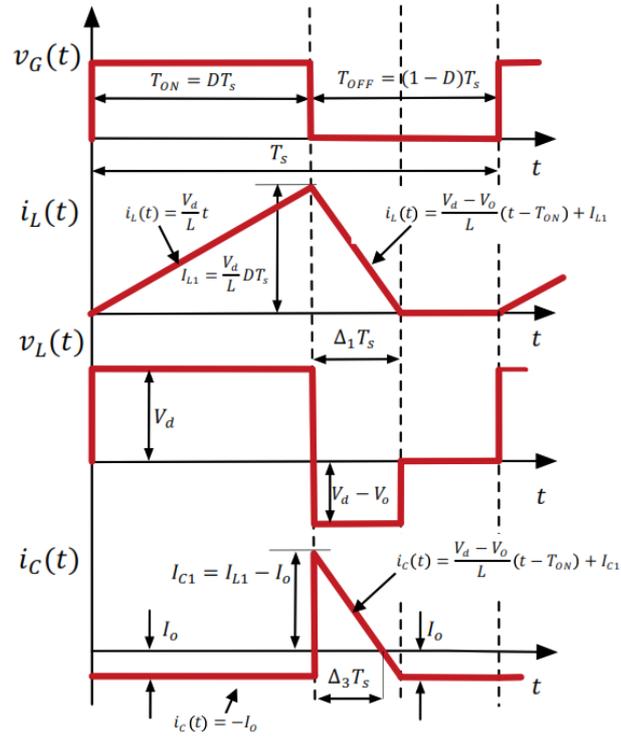


Ilustración 10. Cronogramas MCD [14].

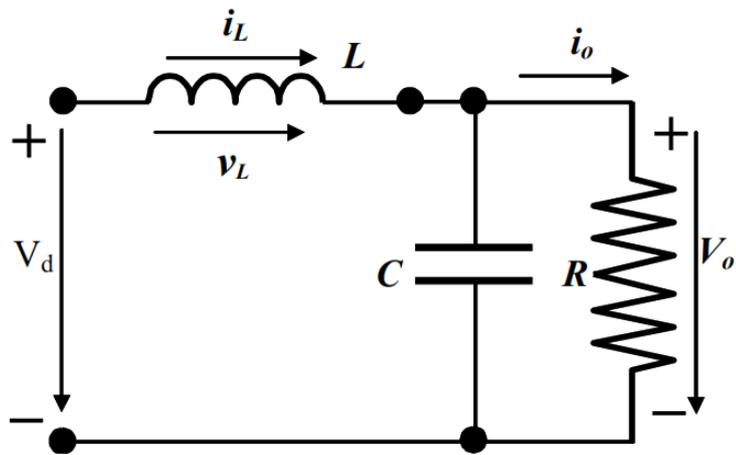


Ilustración 11. Circuito del convertidor en MCD [14].

2.1.1 Especificaciones de Diseño

Para simular modelo de convertidor se tienen en cuenta los parámetros reales de diseño especificados en la Tabla 3:

Parámetro	Valor
Tensión de entrada	120 V
Corriente máxima a la salida	20 A
Inductancia	3 mH
Resistencia	22.5 Ω
Frecuencia de conmutación	4 kHz
Tensión máxima a la salida	400 V

Tabla 3. Parámetros de diseño del equipo real.

Si se diseña para $V_{in} = 100$, se deberá limitar la corriente por la bobina a 12 A, ya que la potencia nominal del equipo es de $P_n = 1.2$ kW.

$$I_L = \frac{P_n}{V_{in}} = 12 \text{ A.}$$

Ecuación 1. Cálculo de la corriente por la bobina.

Si se conecta la fuente de tensión a la entrada del circuito directamente, al proporcionar la tensión a la entrada se produciría un pico de corriente (cortocircuito) en la resistencia de salida o carga.

Para evitar el cortocircuito, se implementa una etapa de precarga, consistente en una resistencia de precarga controlada por dos contactores. De esta forma, el pico de corriente es limitado por la resistencia R, que introducirá una pérdida de corriente determinada a la entrada del circuito.

2.1.2 Implementación en Simulink®

Una vez conocidos los componentes y la operativa del convertidor Boost, se pasa a construir el modelo en Simulink®, para su posterior simulación. La topología del convertidor será la representada en la Ilustración 12.:

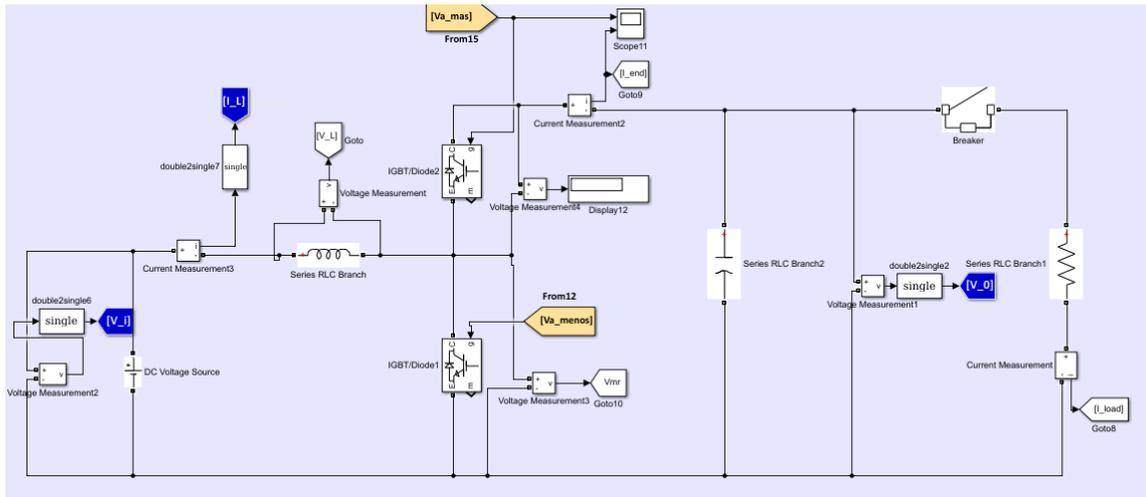


Ilustración 12. Modelo del Convertidor en Simulink®.

Para el análisis del flujo energético a través del convertidor, se presenta la evolución de las variables de tensión e intensidad a la entrada y a la salida.

La entrada se ha simulado mediante un generador de continua, siendo la tensión a la entrada de 100 V.

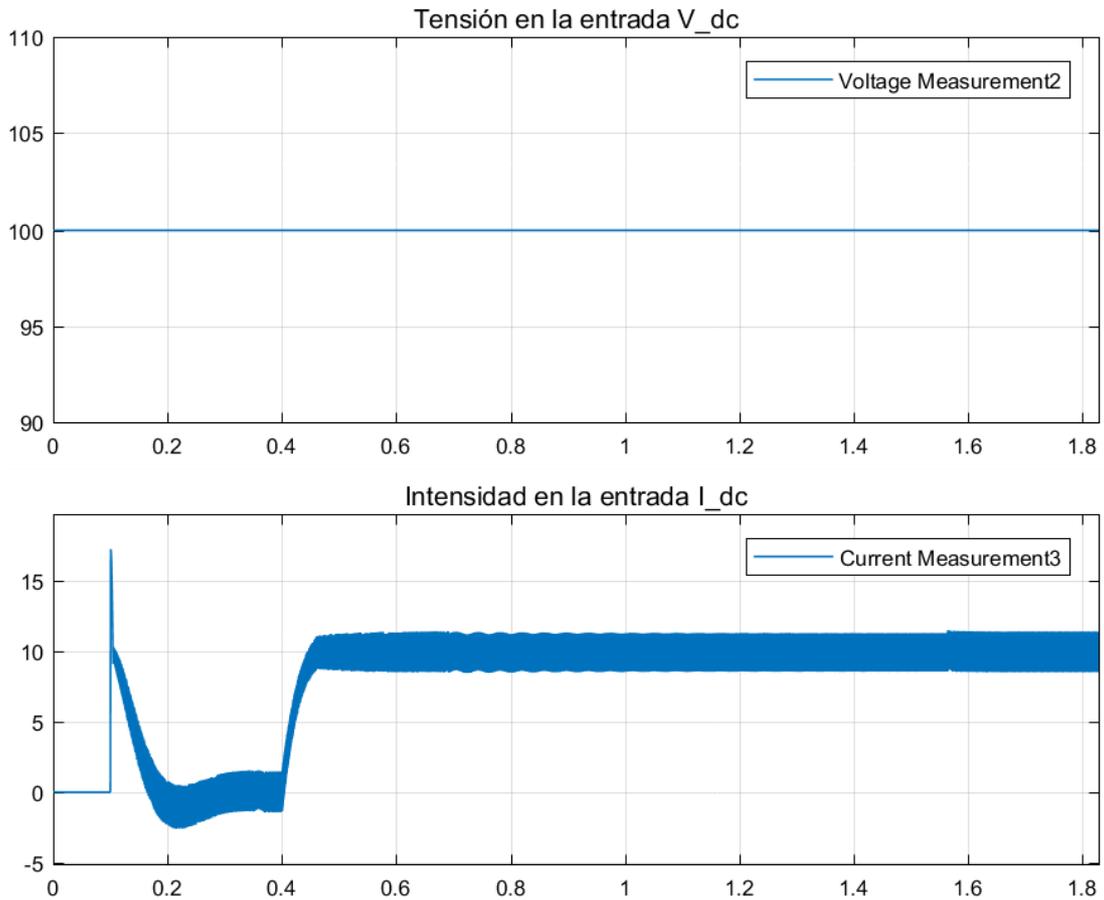


Ilustración 13. Tensión e intensidad a la entrada del Convertidor.

La Ilustración 13 representa la corriente de la bobina ampliada. Puede apreciarse que el convertidor se encuentra en modo de conducción continua (MCC). Siendo el valor de la corriente de entrada para T_{OFF} de $I_0 = 8.51 A$.

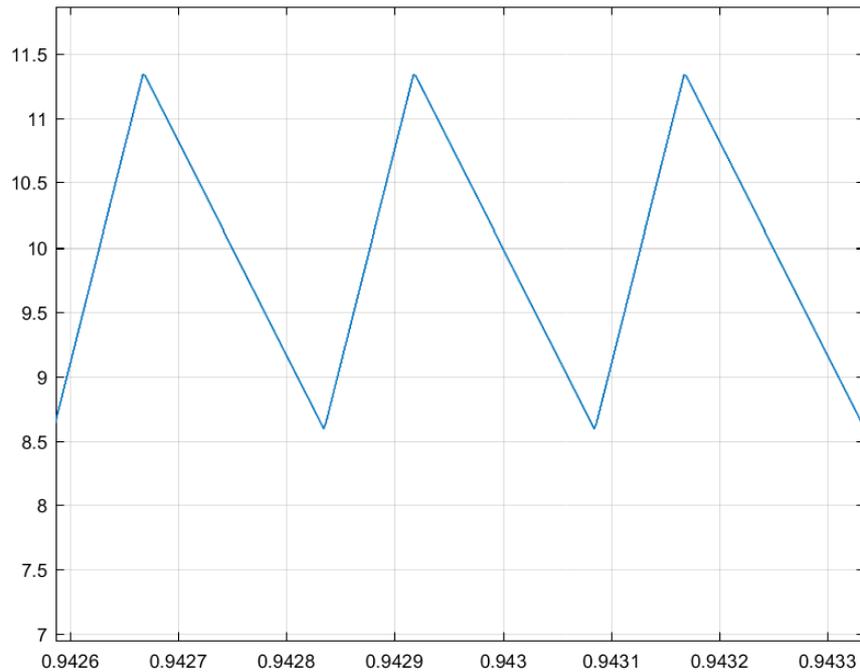


Ilustración 14. Señal ampliada de la corriente de la bobina I_L .

El valor de la corriente de la bobina aumenta hasta un valor máximo, $t_{enable} = 0,1s$, momento en el que se habilita el control, y éste comienza a enviar a los IGBTs una señal de Duty Cycle.

Una vez ha descendido la corriente por efecto del control, el interruptor automático, que se había configurado para cerrarse en $t_{load} = 0.4 s$, conecta la carga a la salida del convertidor. A partir de este momento, la corriente se regula y se estabiliza.

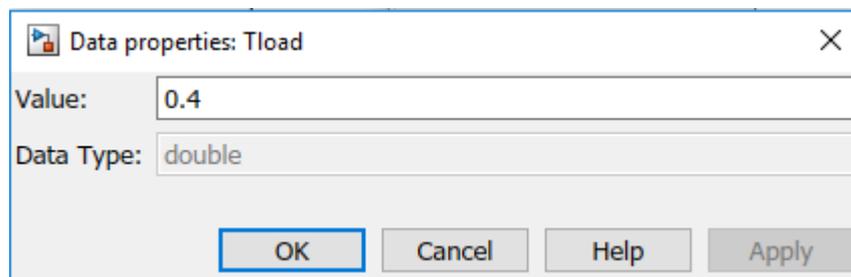


Ilustración 15. Tiempo de carga.

A continuación, la Ilustración 15 muestra las representaciones de la tensión e intensidad a la salida del convertidor. Se puede apreciar, como se ha explicado previamente, el momento en el que se comienza a controlar el convertidor, así como el instante en el que comienza a conducir la

carga, ya que comienza a circular corriente a través de esta.

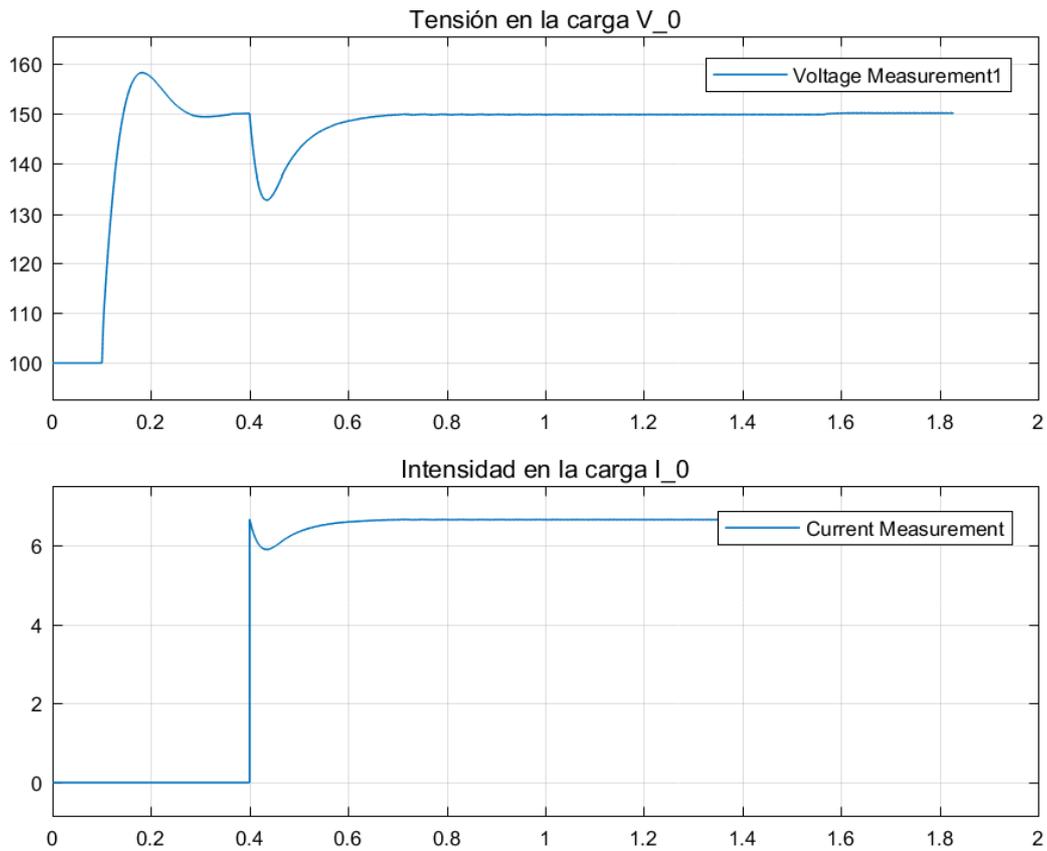


Ilustración 16. Tensión e intensidad en la carga.

Finalmente, en la Ilustración 16 se puede observar la potencia a la entrada del convertidor frente a la salida. Cuando se estabiliza la dinámica del convertidor, $P_{IN} = P_{OUT} = 1000W$.

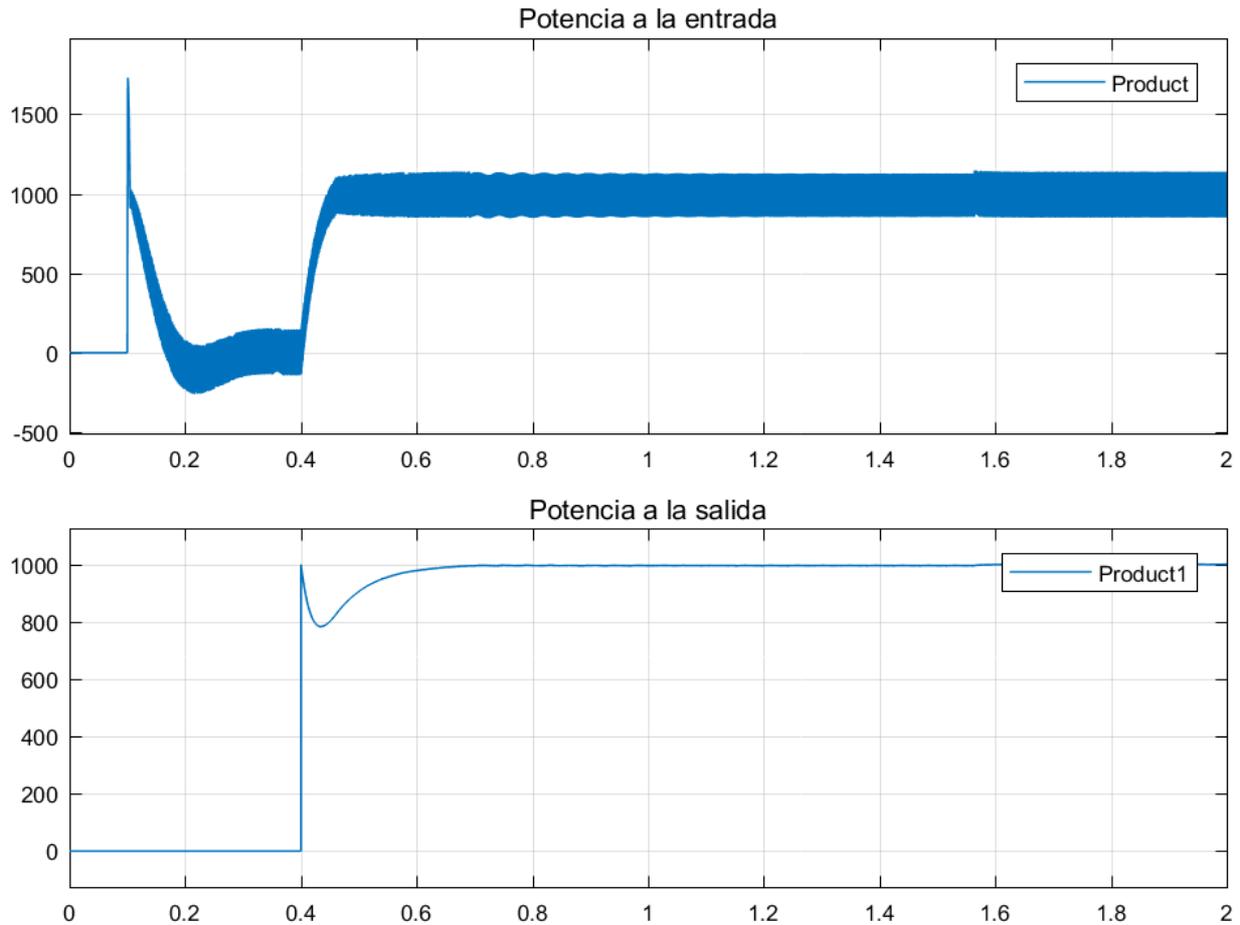


Ilustración 17. Potencias a la entrada y la salida del convertidor.

2.1.3 Mecanismo de precarga

Como queda reflejado en los gráficos anteriores, al conectar la fuente de tensión a la entrada del circuito directamente, en el momento de cierre del interruptor, se produce un pico de corriente en la resistencia de salida o carga.

Para evitar el cortocircuito, se implementa una etapa de precarga, consistente en una resistencia de precarga controlada por dos contactores. De esta forma, el pico de corriente es limitado por la resistencia R , que introducirá una pérdida de corriente determinada a la entrada del circuito.

Al instalar mayor resistencia de precarga, el pico de corriente producido será menor, pero alcanzará la tensión deseada más lentamente, por lo que se debe alcanzar un compromiso entre estas variables.

El diseño de la $R_{precarga} = 100 \Omega$ en caso ideal, sin embargo, se utiliza $R_{precarga} = 22,5 \Omega$ para agilizar las simulaciones.

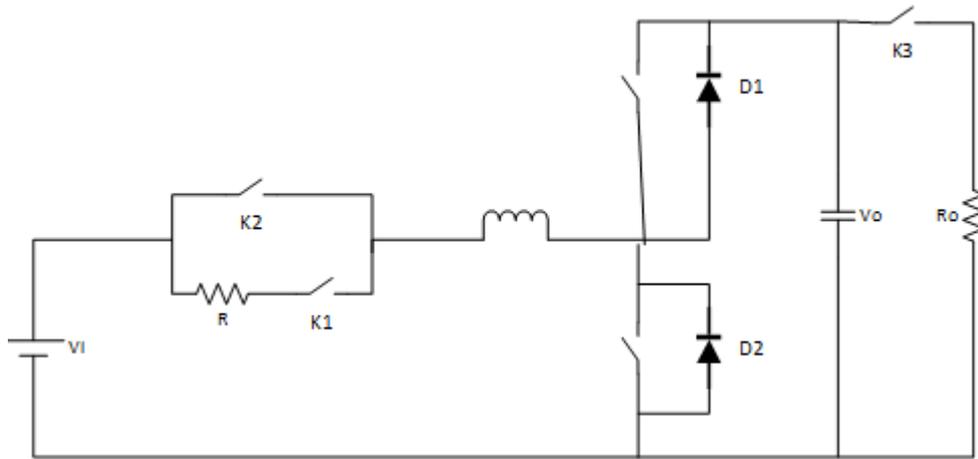


Ilustración 18. Topología del convertidor Boost con mecanismo de precarga.

Para la precarga, se dispone de tres interruptores K_1 , K_2 y K_3 . Estos controlarán la conexión y desconexión de los dispositivos al circuito: resistencia de precarga y resistencia de carga.

Inicialmente, se conecta el circuito a la fuente de tensión V_i encontrándose K_1 cerrado y K_2 y K_3 abiertos. De esta forma, la corriente de entrada sufre una pérdida mediante la resistencia de precarga.

Se realiza un bypass a dicha resistencia de precarga, abriendo el contactor K_1 y cerrando K_2 , una vez el condensador ha alcanzado una tensión mínima, momento a partir del cual comienza la carga del mismo.

Una vez se tiene la tensión V_0 controlada, es decir, ha superado un cierto límite, se cierra el contactor K_3 , conectando la resistencia R_0 al circuito.

Se representa esta operativa mediante el diagrama de flujo de la Ilustración 19.

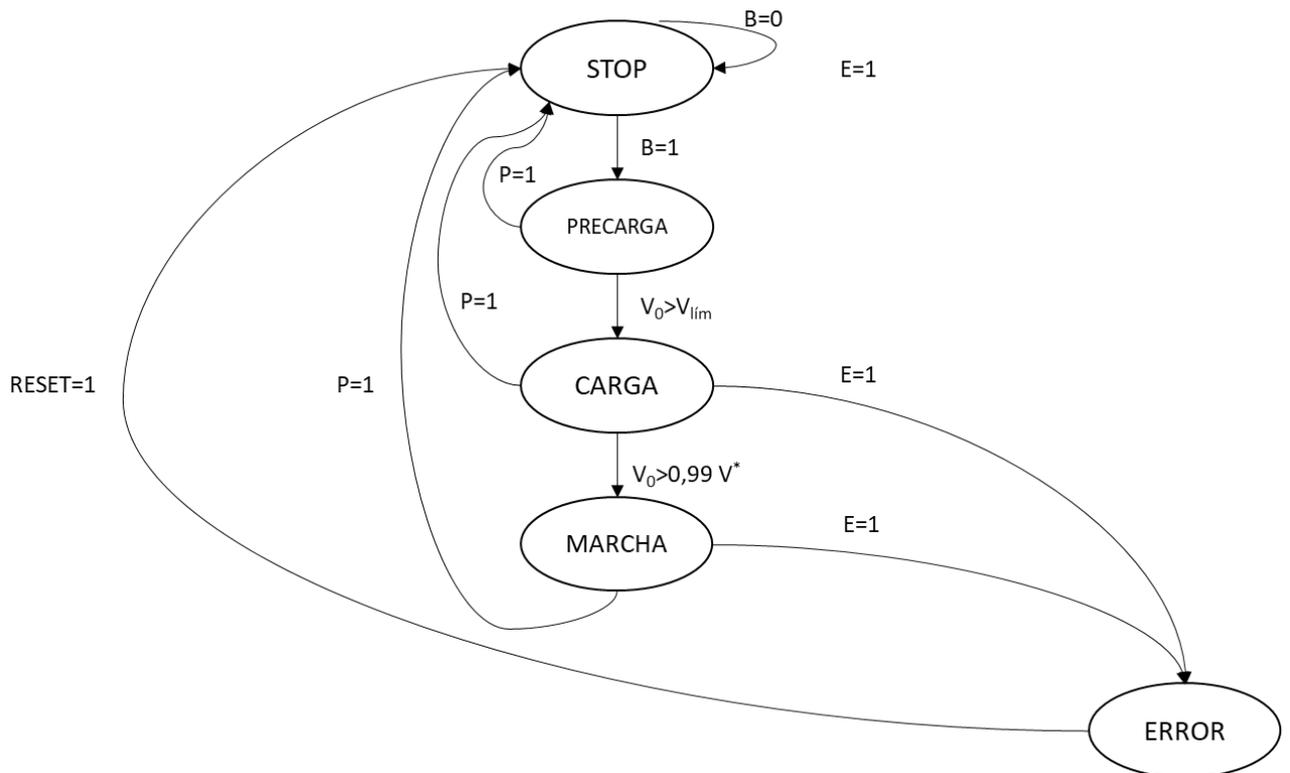


Ilustración 19. Diagrama de estados de la máquina.

Los estados a través de los que pasa la máquina son:

- **STOP.** Estado de Parada. Es el estado inicial en el que se encuentra la máquina hasta que se pulsa el botón de encendido “B”. Los interruptores K_1 y K_2 se encuentran abiertos.
- **PRECARGA.** Se comienza a transferir energía al condensador, introduciendo una pérdida a la entrada para evitar picos de corriente. Se pasa a este estado una vez se acciona el botón “B”. K_1 se cierra hasta que la tensión en el condensador alcanza un valor mínimo. En este caso, siendo $V_0 = 150 V$, se establece la tensión límite igual a la de la entrada $V_{lim} = 100 V$.
- **CARGA.** La energía se transfiere de forma directa al condensador cuando éste ya ha alcanzado una energía mínima. Se abre K_1 y se cierra K_2 . K_3 se mantiene abierto, hasta alcanzar una tensión cercana a la tensión deseada a la salida $(z^*) V_{lim} \cong 0.95 * V_0^{ref} = 95 V$
- **MARCHA.** Se inicia la conversión de potencia. K_2 se mantiene cerrado y se cierra K_3 , conectando la carga a la salida.
- **ERROR.** En cualquier momento que se detecte algún error, debe pasar a este estado. Constará de un botón de RESET que pasa del estado de ERROR a STOP, independientemente del estado anterior. Posibles detonadores de este estado serán:

- Problema de sobre corriente. Supera la corriente que soportan los dispositivos:
 - Bobina: 20 A.
 - Mosfet IRFP4227: 65 A.
- Sobretensión del condensador. Cuando supera la tensión máxima que este puede soportar.
 - Condensador 3D22C5: 200V
- Además, se accionará en aquellos casos en los que la tensión a la entrada no supere $\frac{V_0}{3}$, ya que es la tensión mínima para la que podrá conmutar. De
- En resumen, teniendo en cuenta un margen de seguridad, el estado de error se accionará para:
 - $E = 1$.
 - $I_L > 17 A$.
 - $V_0 > 180 V$.
 - $V_{IN} < 50 V$.

Para evitar incoherencias en la máquina de estados, se introducirá un escalón (step) en el botón “P” que permite parar la máquina, de forma que, al arrancar el modelo, éste siempre se encuentre deshabilitado, hasta que se haya iniciado la precarga.

En resumen, la **máquina de estados** constará de

- 4 entradas: E (señal de emergencia), B (señal de inicio), P (señal de parada), V_0 (tensión a la salida del convertidor).
- 3 salidas: Señales K_1, K_2 y K_3 . Las cuales conmutarán entre 0 y 1 en función del estado en el que se encuentre.
- 5 estados: Parada, precarga, carga, marcha y emergencia.

Debido a esta definición y a las necesidades de nuestra máquina de estados, se puede definir como una máquina de Mealy, puesto que la salida es función del estado y de una entrada.

Type of Machine	Semantics	Applications
Mealy	Output is a function of inputs <i>and</i> state: $y = g(X, u)$	Clocked synchronous machines where state transitions occur on clock edges
Moore	Output is a function <i>only</i> of state: $y = g(X)$	Clocked synchronous machines where outputs are modified at clock edges

Tabla 4. Tipos de máquinas de estado de Matlab® [15].

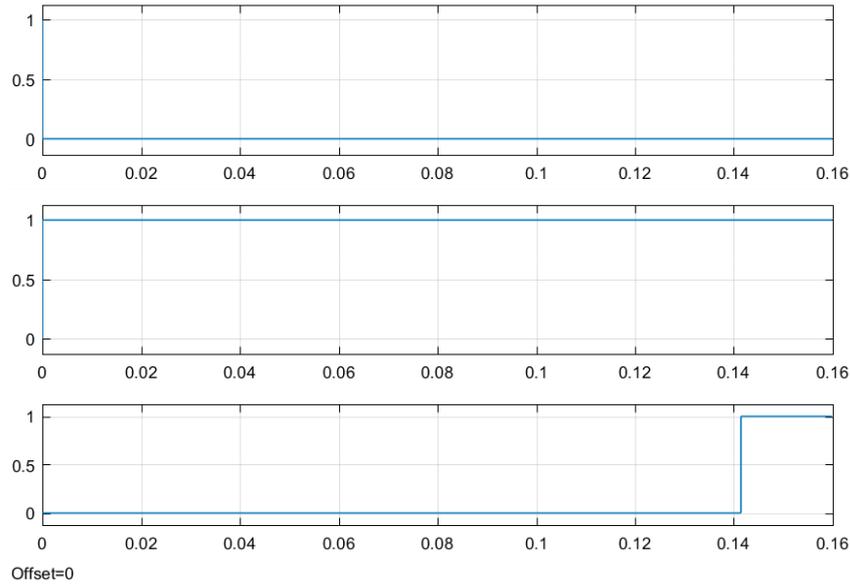


Ilustración 20. Estado de los interruptores K_1, K_2 y K_3 para $R_{precarga} = 22,5 \Omega$.

La Ilustración 20 representa la variación de las señales de salida de la máquina de estados K_1, K_2 y K_3 desde el estado de STOP hasta RUN, mostrando cada gráfico una de estas señales en orden descendente.

El gráfico superior, representa cómo la señal de salida K_1 está en el instante inicial a nivel alto, y pasa a nivel bajo inmediatamente, ya que la máquina se inicia con el botón B activado a nivel alto. Se mantiene durante toda la simulación en dicho valor alcanzado (bajo).

El gráfico intermedio, representa la señal K_2 , tal y como se ha explicado, comienza a nivel bajo durante los estados de STOP y PRECARGA, y pasa a nivel alto en el instante que la máquina pasa a CARGA, manteniéndose con ese valor durante el resto de la simulación.

En el gráfico inferior, se muestra como la salida K_3 se mantiene a nivel bajo desde el estado STOP hasta el estado de CARGA, pasando por PRECARGA, y pasa a nivel alto una vez la máquina alcanza el estado RUN.

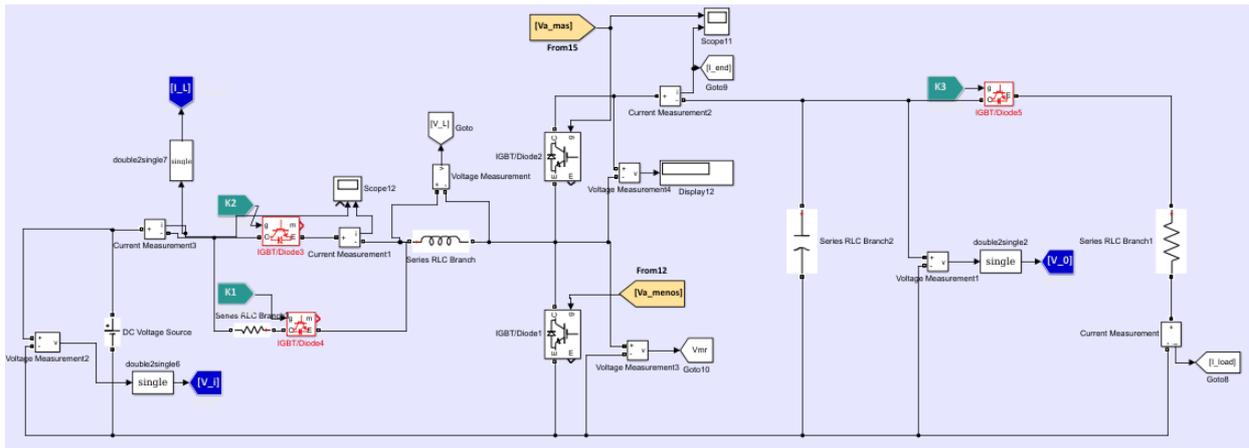


Ilustración 21. Modelo de convertidor en Simulink® con precarga.

2.2 Algoritmo MTTP y Modelo de Control

Los ingenieros que desarrollan inversores solares implementan algoritmos MPPT para maximizar la energía generada por los sistemas fotovoltaicos (FV). Estos algoritmos controlan la tensión para garantizar que el sistema funcione en el “punto de máxima potencia” (o tensión máxima) de la curva de potencia, tal y como se muestra a continuación [16].

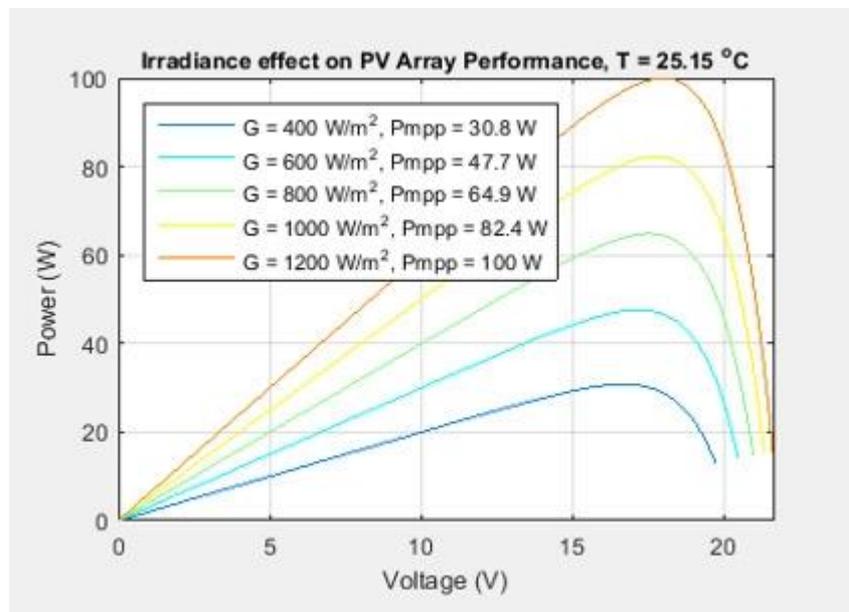


Ilustración 22. Relación Potencia-Tensión en función de la irradiancia [16].

Para regular la tensión de entrada del convertidor al valor de máxima extracción, se implementará el algoritmo de MPPT por conductancia incremental. Este algoritmo calcula el punto de máxima potencia mediante la comparación de la conductancia incremental ($\frac{\delta P}{\delta V}$) con la del sistema. Según ésta sea mayor o menor, se varía el sentido del incremento de tensión, hasta

alcanzar el punto óptimo. De manera que:

- $\frac{\delta P}{\delta V} = 0 \rightarrow \Delta v_{ref} = 0$
- $\frac{\delta P}{\delta V} > 0 \rightarrow \Delta v_{ref} < 0$
- $\frac{\delta P}{\delta V} < 0 \rightarrow \Delta v_{ref} > 0$

Observando las oscilaciones de la tensión de referencia del modelo simulado, se aprecian los incrementales de tensión aplicados para encontrar el punto óptimo.

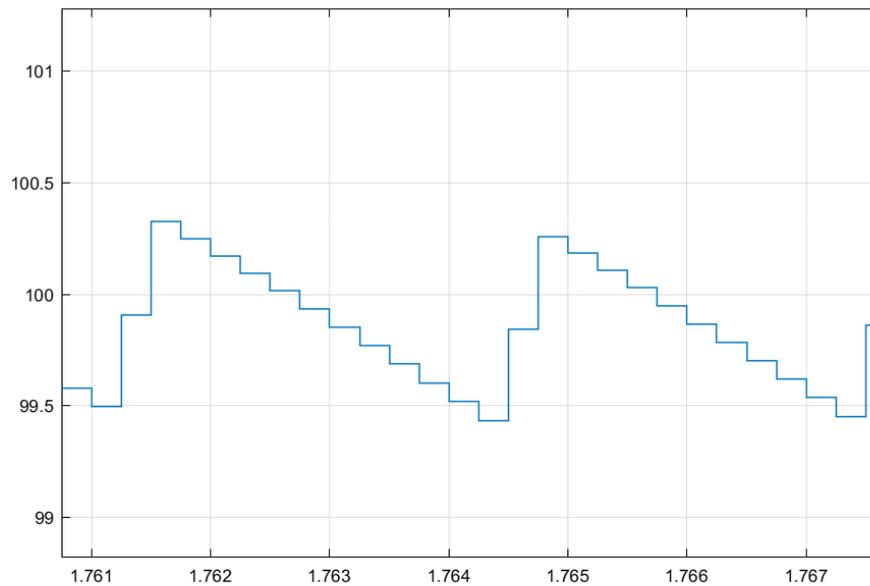


Ilustración 23. Tensión de referencia del controlador.

Las ventajas que ofrece este algoritmo son [17]:

- Ofrece un buen rendimiento bajo condiciones atmosféricas cambiantes.
- Consigue una oscilación menor alrededor del MPP.

Inconveniente:

- Requiere un circuito de control complejo que puede conllevar un elevado coste del sistema.

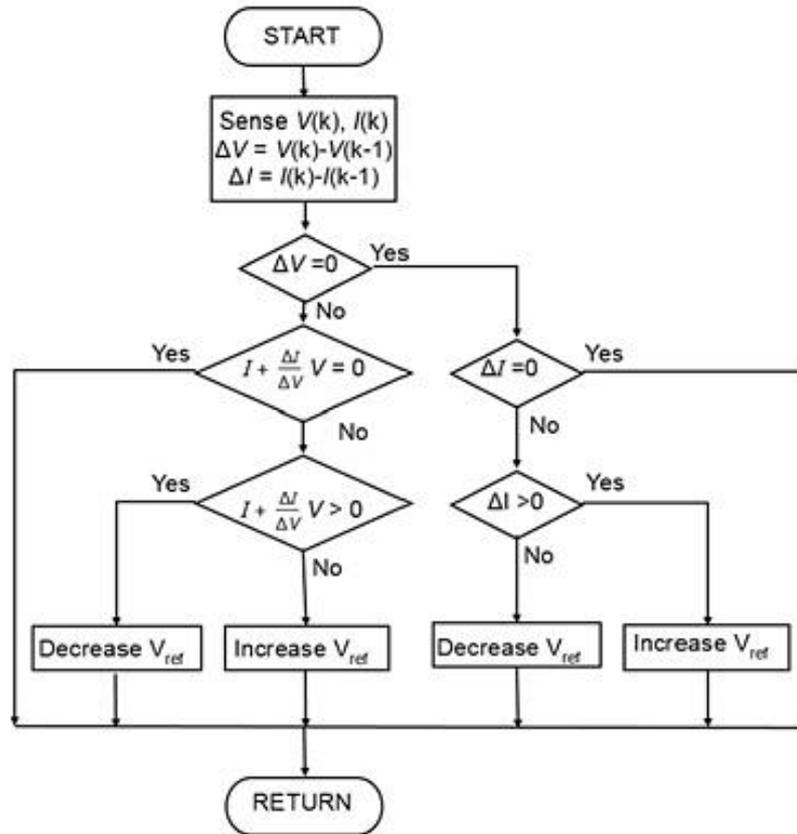


Ilustración 24. Diagrama de flujo control MPPT [16].

La frecuencia de conmutación de los interruptores IGBT se fijará en 4 kHz, el cual es un valor típicamente usado en dichas aplicaciones.

La implementación del lazo de control para el convertidor elevador mediante PI, compara la tensión a la salida con la referencia (Tensión de salida deseada, $v_0 = 150$ V). Mediante el controlador PI se calcula una potencia a la entrada deseada, de forma que se obtiene una corriente de la bobina deseada $\frac{p_{in}^*}{v_{in}} = i_L^*$. Al comparar con la referencia de la corriente de la bobina, se obtiene la tensión de referencia v_m .

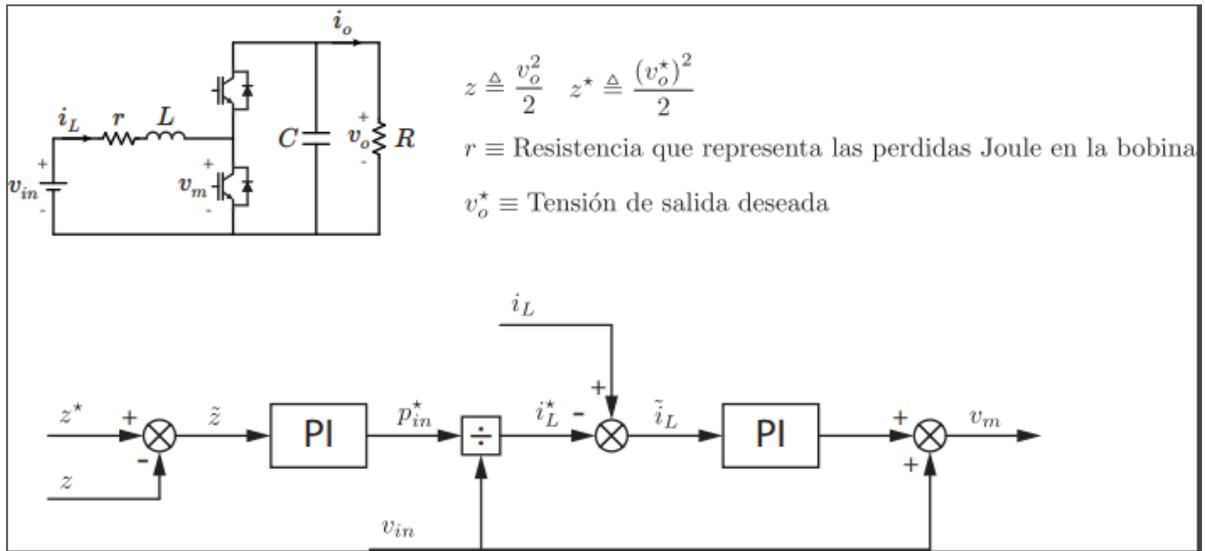


Ilustración 25. Esquema de control.

Se ajustan los controladores PI con los parámetros de control Proporcional e Integral de la Tabla 5.

COMPONENTE	PI_1	PI_2
Proporcional (P)	0.2	5
Integral (I)	5	20

Tabla 5. Diseño de los parámetros de control.

Para obtener el Duty Cycle necesario para parametrizar la conmutación, se divide la tensión de referencia obtenida entre la tensión a la salida del convertidor $\frac{v_m}{v_o} = D$.

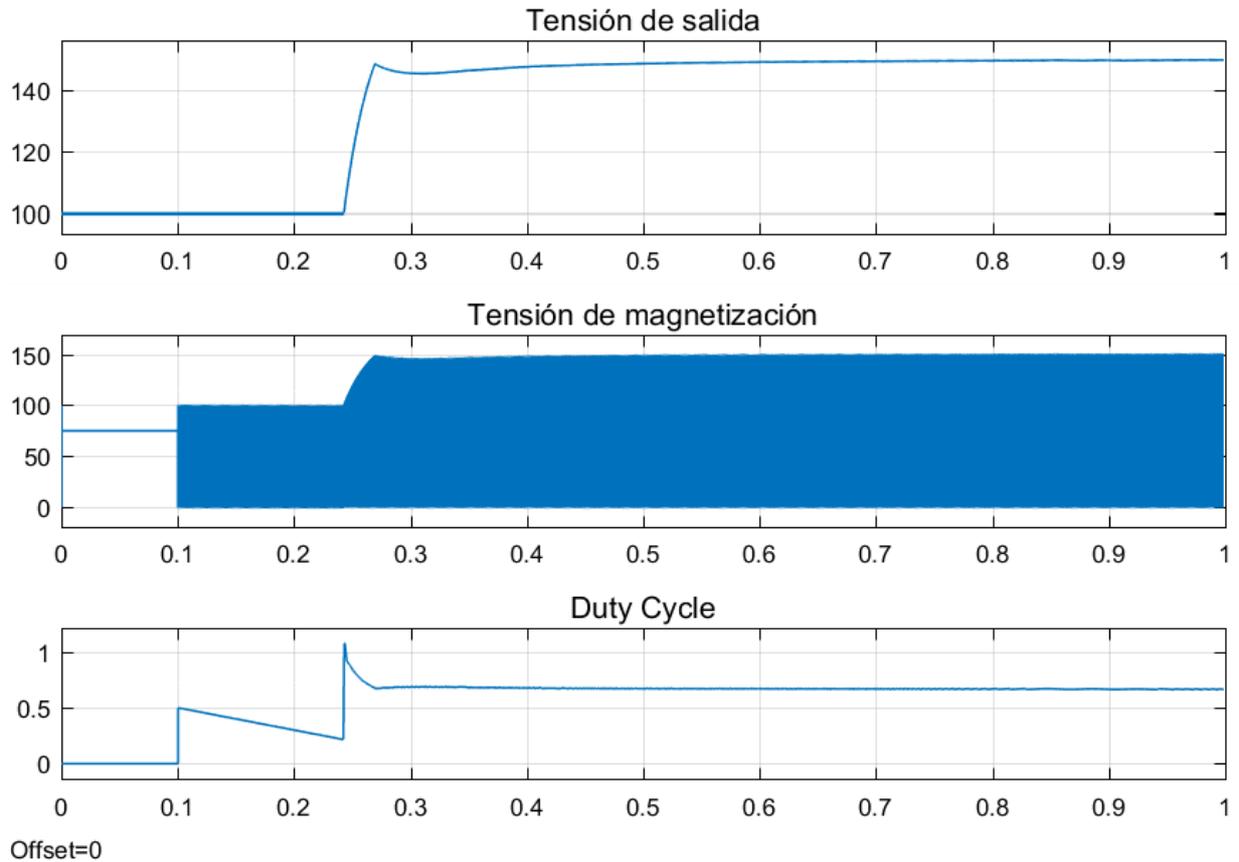


Ilustración 26. Cálculo del Duty Cycle del controlador.

El Duty Cycle resulta ser de $D \cong 0,67$. Este resultado empírico es coherente con la tensión de salida deseada, ya que cumple con la condición de:

$$D = \frac{V_i}{V_0} = \frac{100}{150} = \frac{2}{3} \cong 0.67$$

Ecuación 2. Cálculo del Duty Cycle.

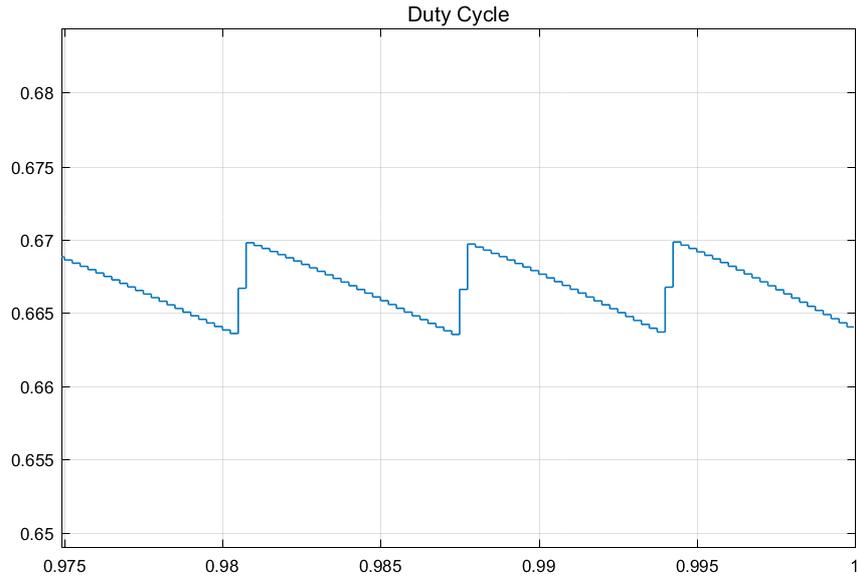


Ilustración 27. Zoom de la señal de Duty Cycle.

2.2.1 Pulse Width Modulation PWM

La modulación por ancho de pulsos es la estrategia de disparo de los IGBTs del convertidor que se va a utilizar para la conversión de potencia. La principal ventaja de esta técnica es la obtención de una señal de salida con armónicos lejanos a la frecuencia fundamental (corrientes parásitas).

Este método consiste en emplear un periodo de conmutación $T_s = t_{on} + t_{off}$, de forma que variando ton es posible variar la tensión media de salida. Esta relación entre periodos de conducción y conmutación es conocida como Duty Cycle. [18]

La señal de disparo del semiconductor es generada por la comparación de una señal de control con una triangular.

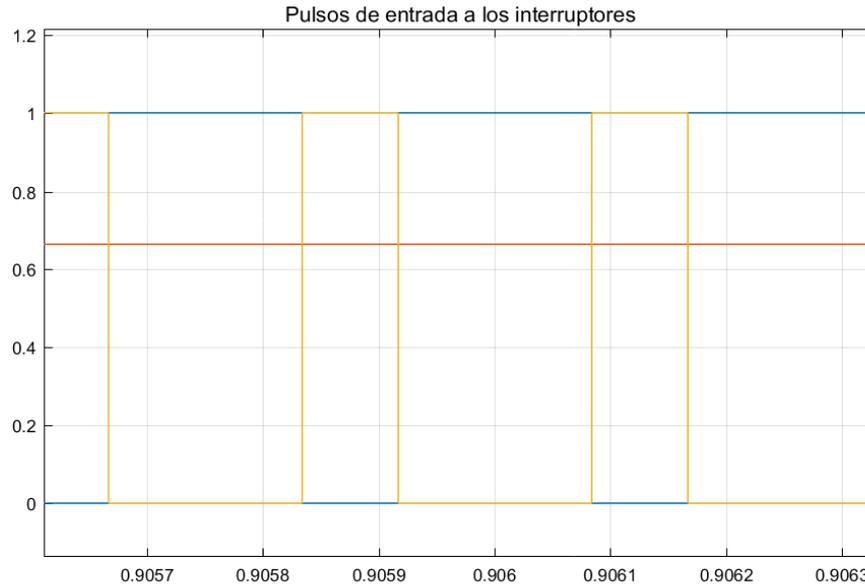


Ilustración 28. Pulsos de modulación del convertidor y Duty Cycle (rojo).

2.2.2 Transferencia de datos entre subsistemas

Siendo la frecuencia del PWM de tiempo de integración de Matlab® de $f_s = \frac{1}{10^{-6}s}$ para tener una buena resolución, y la de muestreo del controlador de $f_s = 4 \text{ kHz}$, debe disponerse un mecanismo de gestión de la transferencia de datos entre bloques que operan a distinta frecuencia.

Este mecanismo se integrará mediante el bloque Rate Transition [15], el cual transfiere datos desde la salida de un bloque a una frecuencia a la entrada de otro que opera a una frecuencia distinta. Para una mayor calidad del dato, se selecciona la opción de “Ensure data integrity during data transfer” en detrimento de la velocidad y de la disponibilidad de memoria. Toma una medida y la deja en espera durante 250us.

Asimismo, para vincular la temporalidad de la transferencia de datos a la frecuencia específica de cada subsistema, se selecciona “Ensure deterministic data transfer”. En caso contrario, dicha velocidad dependería de la disponibilidad del dato, así como del instante relativo en el que cada bloque comienza a ejecutarse.

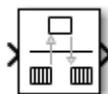


Ilustración 29. Bloque Rate Transition

Para realizar un análisis de la frecuencia a la que opera cada bloque y cada subsistema del

modelo, se puede usar la opción de Ver Información de Tiempo de Muestreo (Display> Sample Time > All). Esta opción muestra los tiempos de muestreo presentes en el modelo, clasificados por tipo, mediante colores y anotaciones en el mismo.

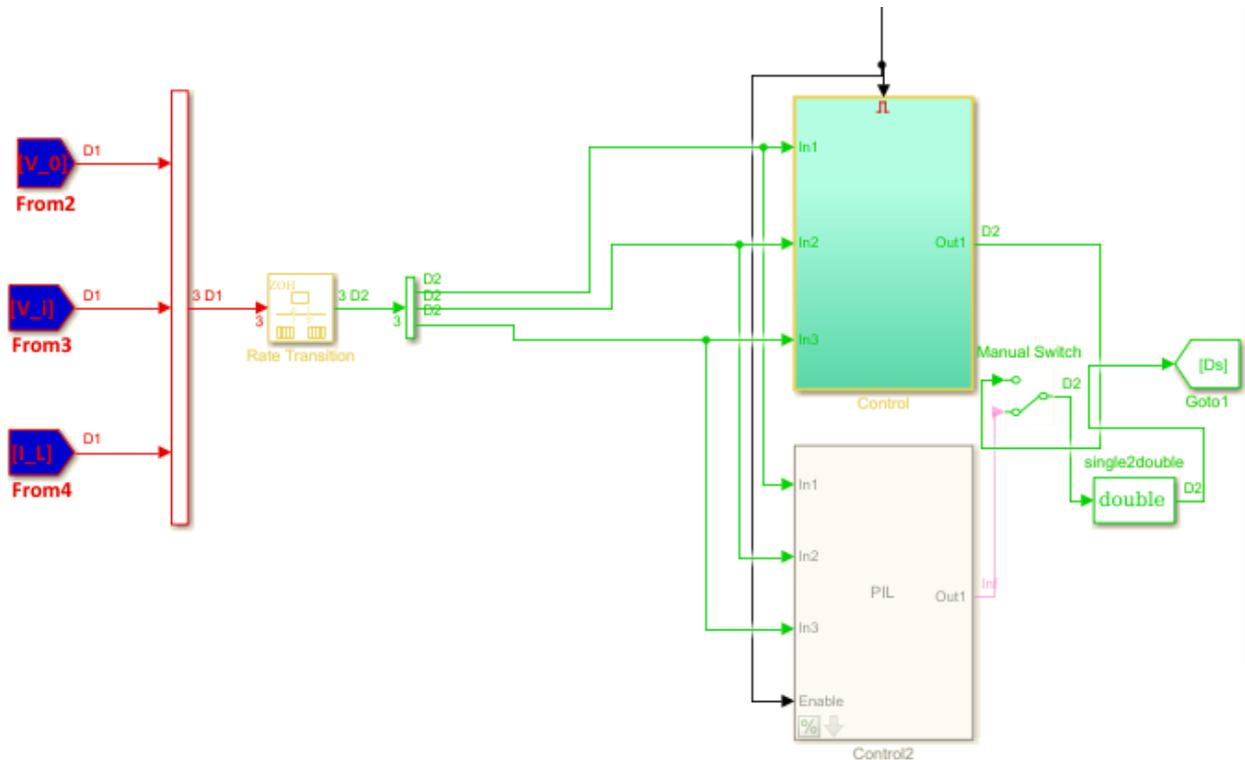


Ilustración 30. Información de tiempo de muestreo.

2.3 Detalle global del modelo

El modelo construido en Simulink® consta de tres secciones principales, coincidentes con las secciones anteriores del presente Capítulo:

- Controlador y mecanismo PWM. Mediante el cual se mandan las señales de conmutación a los Mosfet.
- Máquina de estados. Para controlar el mecanismo de activación del convertidor.
- Convertidor Elevador. Es el dispositivo que se quiere controlar.

En este epígrafe se detalla cada una de estas partes de código, viendo el funcionamiento y los parámetros de los principales bloques de la simulación.

2.3.1 Controlador y mecanismo PWM

El siguiente subsistema está formado por dos secciones principalmente, el controlador y el mecanismo PWM, los cuales se comunican entre sí de forma que:

- La salida “Ds” (el duty cycle) del controlador, es entrada del subsistema generador de la señal PWM.
- La entrada “Step” del controlador, que lo habilita (dispara el bloque “enable”), también afectará a la salida de las señales que modularán las conmutaciones de los Mosfet: “Va_mas” y “Va_menos”, que estarán a 0 hasta que dicho “Step” envíe valores de señal de nivel alto.

Parámetro	Valor
Periodo	0,1 s
Valor inicial	0
Valor final	1

Tabla 6. Parámetros del bloque “Step”

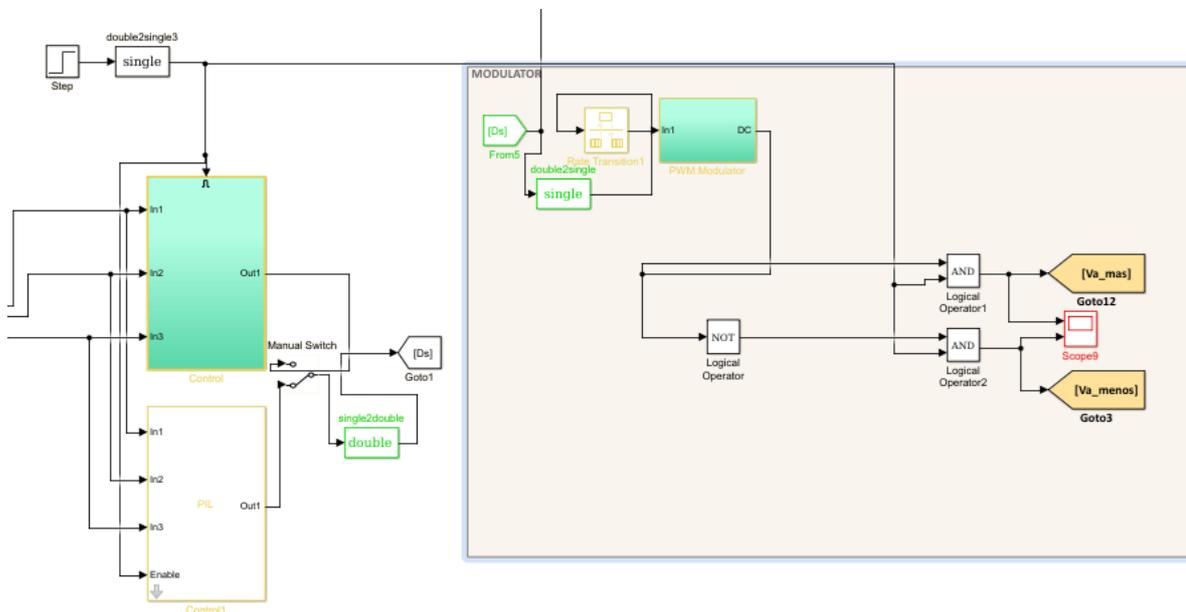


Ilustración 31. Conexión entre el subsistema del controlador y del PWM.

El controlador recibe las siguientes señales de entrada procedentes del convertidor:

- V_0 . Correspondiente a la tensión de salida v_0 .
- V_i . Tensión de entrada del convertidor v_i .

- I_L . Corriente que pasa por la bobina I_L .

Como se ha explicado en el epígrafe 1.1.7 Transferencia de datos entre subsistemas, se hace uso de los bloques “Rate Transition” para la transferencia de las señales de entrada del controlador y el PWM.

Rate Transition	Tiempo de muestro
Controlador	250 μ s
PWM	1 μ s

Tabla 7. Tiempo de muestreo a la salida de los bloques “Rate Transition”

Además, para la conmutación entre los diferentes modos de simulación (SIL, PIL, External Mode), se sitúa un bloque “Manual Switch”, que hace de interruptor manual para seleccionar entre el bloque en lenguaje Simulink® y el bloque PIL.

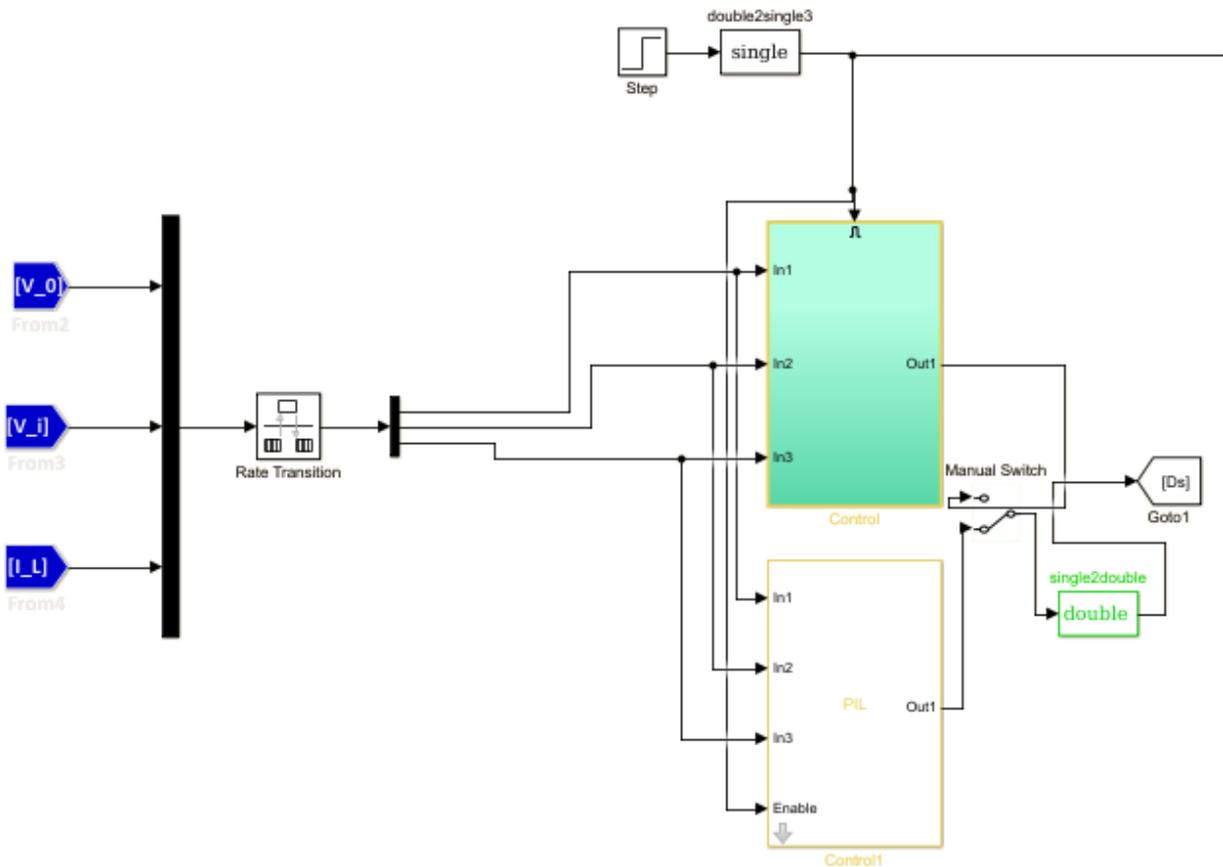


Ilustración 32. Señales de entrada y salida del controlador.

A continuación, se pasa a describir el detalle de cada subsistema de forma independiente:

- **Bloque Controlador.**

Por un lado, se calculan las dos señales de entrada del control de bucle cerrado, z y z^* , haciendo uso de la señal de entrada V_0 y el parámetro de diseño de tensión deseada a la salida 150 v.

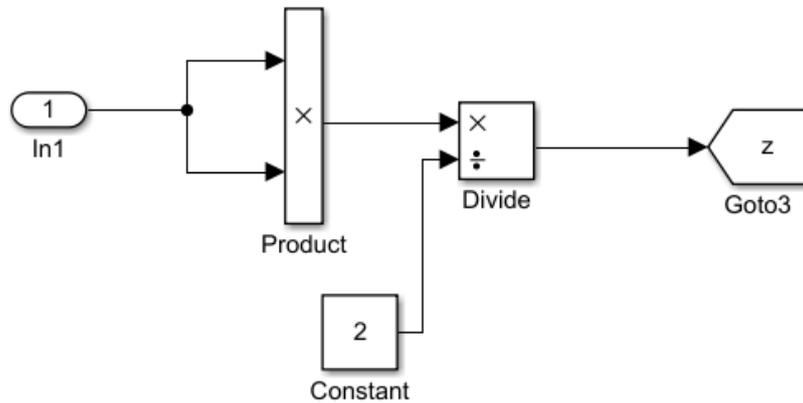


Ilustración 33. Cálculo de z .

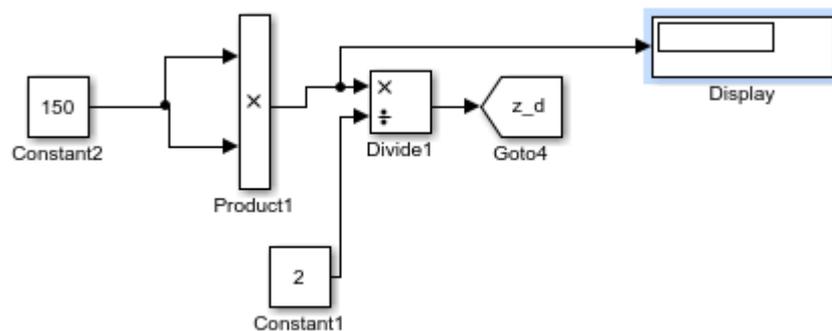


Ilustración 34. Cálculo de z^* .

Posteriormente, se colocan dos controladores PI discretizados en serie, los cuales se denominan “Discrete PI 1” y “Discrete PI 2”, ambos con un tiempo de muestreo heredado del subsistema en que se encuentran, es decir, de $250\mu\text{s}$.

Las señales de z y z^* se comparan con el resto de entradas del controlador: V_i , I_L y V_0 , que han sido transformadas para ser coherentes con el nuevo tiempo de muestreo.

Parámetro	Valor
Tiempo de muestreo	250 μ s
P	0,2
I	5

Tabla 8. Parámetros de “Discrete PI 1”.

Parámetro	Valor
Tiempo de muestreo	250 μ s
P	5
I	20

Tabla 9. Parámetros de “Discrete PI 2”.

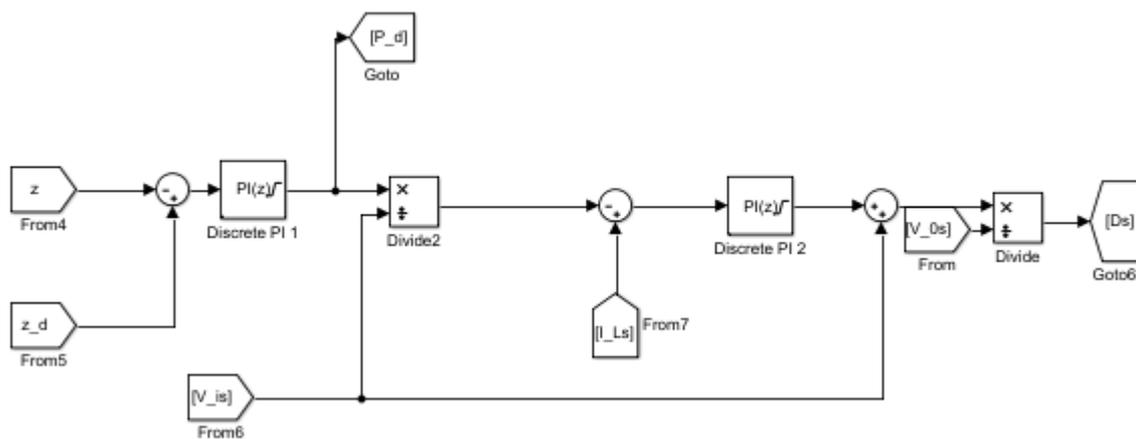


Ilustración 35. Modelo Simulink® del controlador.

- **Bloque PWM.**

La triangular del PWM se representa mediante un bloque “Repeating Sequence” con las características indicadas en la Tabla 10.

Parámetro	Valor
Valores temporales	[0 125 250] μ s
Valores de salida	[0 1 0]

Tabla 10. Parámetros del bloque “Repeating Sequence”.

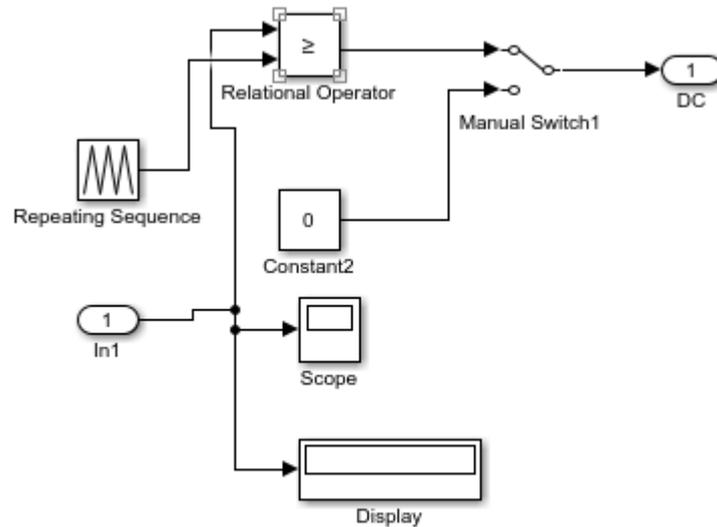


Ilustración 36. Programación mediante bloques del PWM.

2.3.2 Máquina de estados

En la Ilustración 41, se muestra el modelo realizado con Stateflow® de la máquina de Mealy. Para poder hacer uso de estas herramientas, en primer lugar, se debe introducir un bloque tipo Chart. Este bloque, constituirá un subsistema dentro del cual se construirá la máquina.

En la Ilustración 38, se representa el bloque Chart con sus respectivas entradas y salidas. En el modelo, estas serán:

- B. Se modela como un interruptor manual que envía señales de 0-1.
- P. Interruptor manual que envía señales de 0-1.
- V₀. Entrada de la medida del voltaje a la salida del convertidor. Se compara con una serie de valores límite para el paso entre estados de precarga y carga.
- V_i. Entrada de la medida del voltaje del generador de tensión continua.
- RESET. Entrada manual de una señal 0-1 a introducir mediante un interruptor manual.

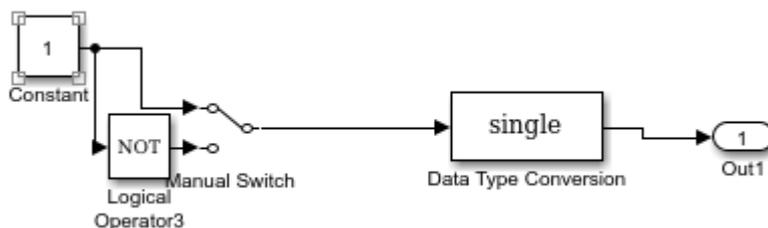


Ilustración 37. Topología del interruptor manual para la activación de los pulsadores P, B y

RESET.

- I_l. Entrada de la medida de la corriente que circula por la bobina.
- K1, K2 y K3 son las señales de salida que envían medidas de 0-1 a los interruptores que configuran el sistema de precarga.

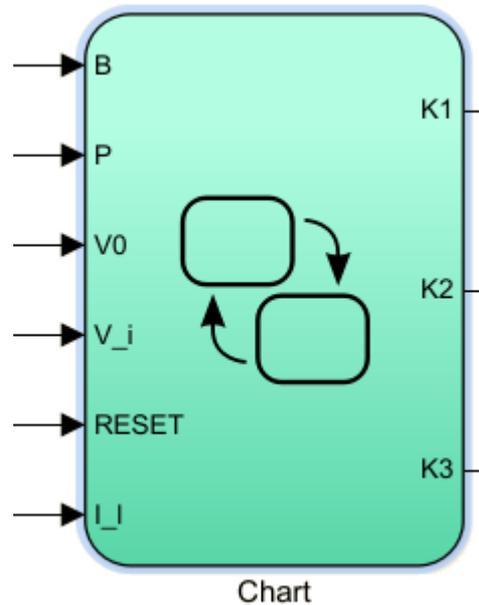


Ilustración 38. Entradas y salidas de la máquina de estados.

Este bloque Chart, está constituido por 5 bloques de estado, representativos de los estados por los que pasa el mecanismo de precarga, así como sus correspondientes transiciones.

Cada estado, se representa mediante un bloque de "Estado", los cuales se seleccionan en la barra de herramientas del lateral izquierdo de la ventana de Simulink®, tal y como se muestra en la Ilustración 41.

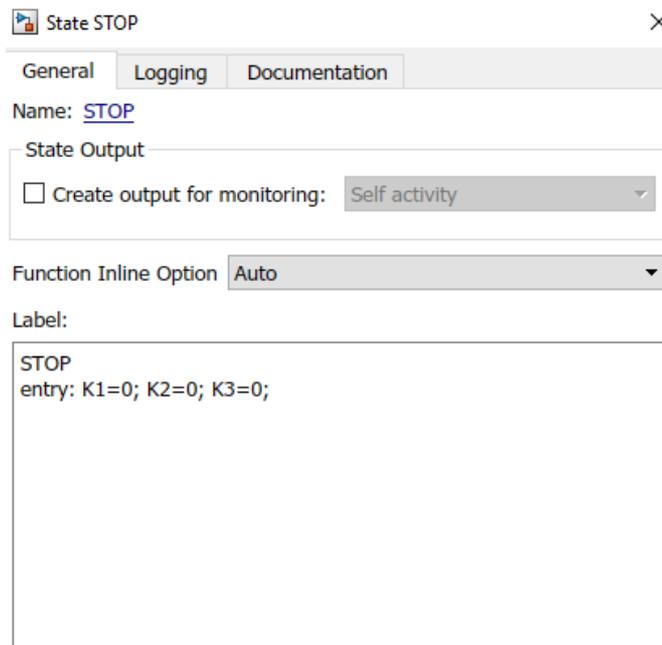


Ilustración 40. Configuración de uno de los estados de la máquina.

Las transiciones entre los estados se introducen mediante los elementos “Default Transition” de la barra lateral mostrada en la Ilustración 39. Si no se introduce una condición de transición, tal y como se muestra en la transición 1, de la Ilustración 41, esta transición se efectuará por defecto. Si se debe cumplir una condición para pasar del estado presente al de destino, esta debe definirse mediante los símbolos $>$, $<$, $=$. Las condiciones deben ir entre corchetes ([]).

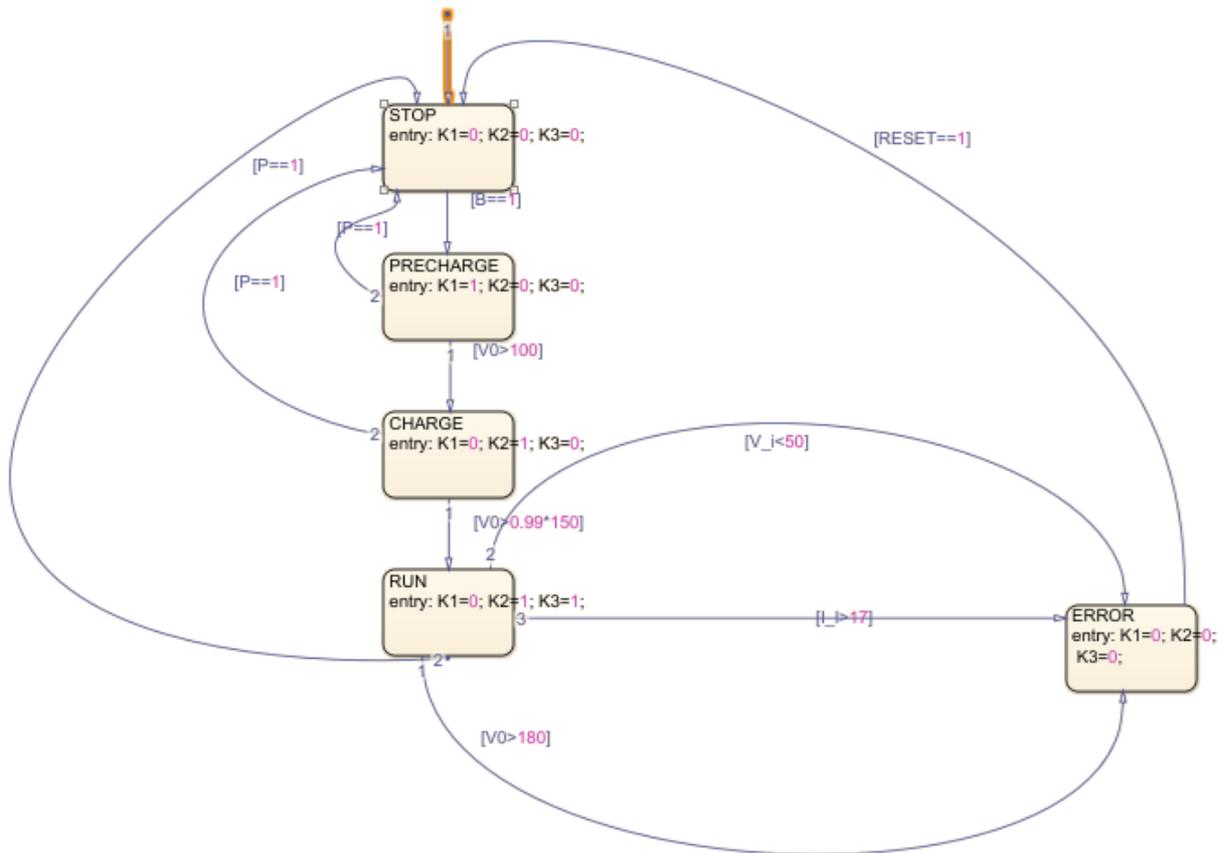


Ilustración 41. Máquina de estado con Stateflow®.

La máquina de estados programada, sigue la operativa indicada anteriormente en este apartado: Inicialmente, se encuentra en el estado STOP, que es alcanzado por defecto, y cuyas salidas K_1, K_2 y K_3 se encuentran a nivel bajo ($=0$). Para abandonar dicho estado, se deberá pulsar el botón manual B, de forma que $B=1$, entrando al estado de PRECARGA, en el cual sólo variará la señal de salida K_1 , que pasará a nivel alto ($=1$).

A partir de este estado y en los sucesivos, si el botón P envía una señal de entrada de nivel alto, se volverá al estado STOP.

Una vez en precarga, si no se envía señal $P=1$, si la tensión a la salida v_0 supera los 100 V, la máquina pasará al estado de CARGA, en el cual, las salidas K_1 y K_3 enviarán señales de nivel bajo, mientras K_2 pasará a nivel alto. Nuevamente, si no se envía la señal de paro ($P=1$), y la tensión a la salida de la máquina supera la tensión límite $v_{lim} = 0.99 * 100 V$, pasará al estado de RUN, donde la señal de salida de K_1 será de nivel bajo, siendo K_2 y K_3 de nivel alto.

Desde el estado RUN, tal y como se ha descrito, se podrá pasar a STOP en caso de recibir una señal de entrada de P a nivel alto, o bien, si se recibe una de las siguientes condiciones: $v_i < 50 v$, $I_l > 17 A$, $v_0 > 180 v$ pasará a estado EMERGENCIA.

El estado de EMERGENCIA enviará las mismas señales de salida que el de STOP (K_1 , K_2 y K_3 a nivel bajo), pero sólo podrá ser abandonado mediante una entrada de la señal RESET a nivel alto (=1).

2.3.3 Convertidor Elevador

El convertidor elevador, recibe las siguientes señales de entrada de los subsistemas descritos en los epígrafes anteriores:

- V_{a_mas} . Señal procedente del módulo PWM que entra en el bloque “IGBT/Diode1”
- V_{a_menos} . Señal procedente del módulo PWM que entra en el bloque “IGBT/Diode2”
- K_1 , K_2 y K_3 . Señales procedentes de la máquina de estados que entran a los bloques “IGBT/Diode3”, “IGBT/Diode4” y “IGBT/Diode5”, respectivamente.

Los bloques tipo “IGBT/Diode” modelan los interruptores y conmutadores del convertidor.

Dispositivo	R_{on}	R_s	C_s
IGBT/Diode1	1 m Ω	1 $\mu\Omega$	inf
IGBT/Diode2	1 m Ω	1 $\mu\Omega$	inf
IGBT/Diode3	1 m Ω	1 $\mu\Omega$	inf
IGBT/Diode4	1 m Ω	1 $\mu\Omega$	inf
IGBT/Diode5	1 m Ω	1 $\mu\Omega$	inf

Tabla 11. Parámetros de los dispositivos tipo “IGBT/Diode”.

El convertidor es alimentado a la entrada por un bloque tipo “DC Voltage Source”, generador de tensión continua de amplitud 100 V.

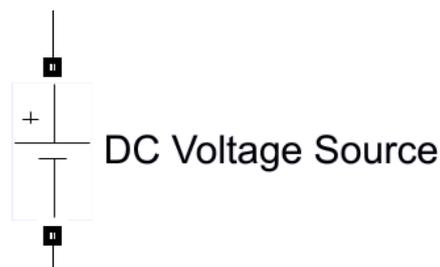


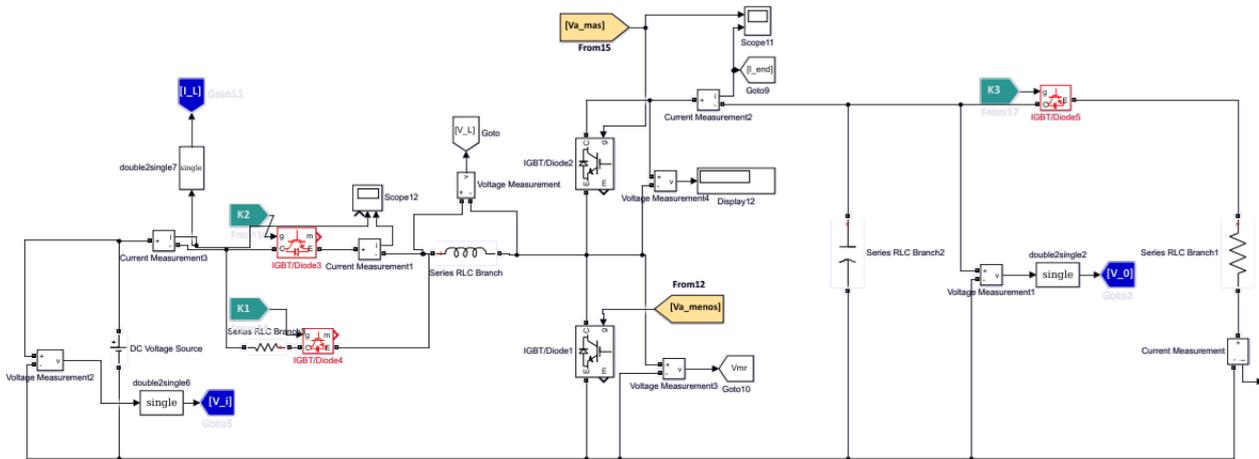
Ilustración 42. Fuente generadora de tensión continua.

La inductancia, la carga a la salida (resistiva), la capacitancia y la resistencia de precarga, se

modelan mediante los bloques “Series RLC Branch”, “Series RLC Branch1”, “Series RLC Branch2” y “Series RLC Branch3”, respectivamente.

Dispositivo	R	L	C	Voltaje inicial
Series RLC Branch (tipo L)	-	3 mH	-	-
Series RLC Branch1 (tipo R)	22,5 Ω	-	-	-
Series RLC Branch2 (tipo C)	-	-	5,6 mF	100 V
Series RLC Branch3 (tipo R)	22,5 Ω	-	-	-

Tabla 12. Parámetros de los dispositivos tipo “Series RLC Branch”.



Capítulo 3

Ejecución

2.4 Modos de ejecución

Para implementar los modelos simulados en Simulink® en el microcontrolador F28335 Delfino, se han utilizado los modos descritos a continuación:

- Software in the Loop (SIL)
- Processor in the Loop (PIL)
- External Mode

En primer lugar, el modo de ejecución SIL (Software in the Loop), se trata de pruebas de un programa en software que simula el funcionamiento del hardware deseado.

Tanto PIL como external mode implican al dispositivo electrónico y cuentan con una configuración que permite la correcta ejecución del programa lanzado, así como comunicación PC-dispositivo. Mientras PIL comunica con el dispositivo de destino en diferido, el External Mode permite la comunicación en tiempo real con éste.

Así, la prueba PIL cumpliría una etapa previa de las pruebas de funcionamiento del programa, estableciendo un modo más ligero y óptimo de funcionamiento, que permitiría reducir los tiempos de espera en las primeras fases.

El External Mode es útil para una fase posterior de ajuste de parámetros del modelo.

2.4.1 Software in the Loop (SIL)

A continuación, se presentan en las Ilustraciones 43, 44 y 45, los resultados de la simulación en Simulink® del modelo previamente a realizar las pruebas en el hardware. Estos resultados se contrastarán con los obtenidos posteriormente en PIL.

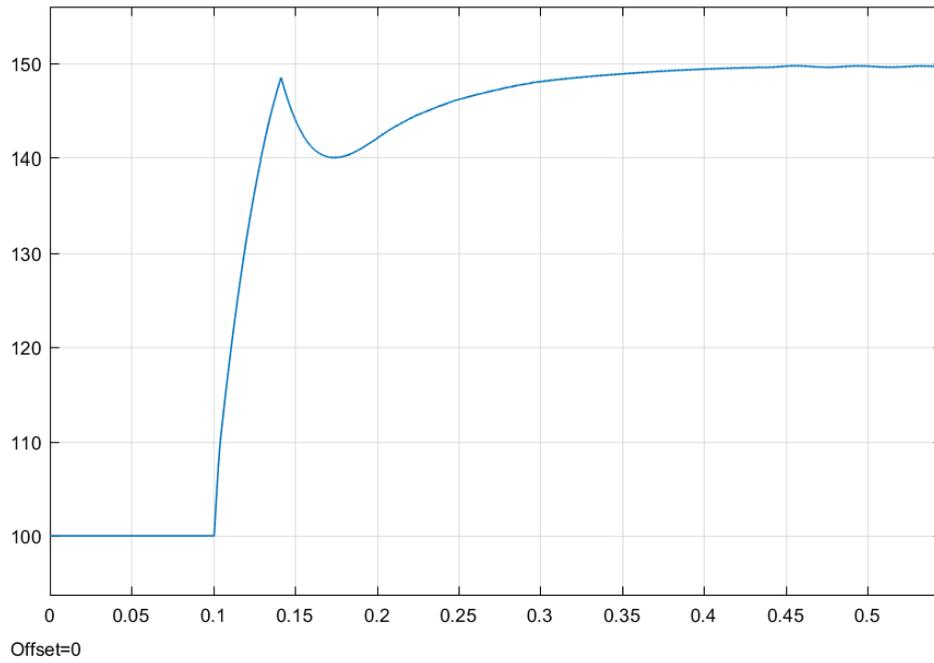


Ilustración 43. Tensión a la salida V_0 .

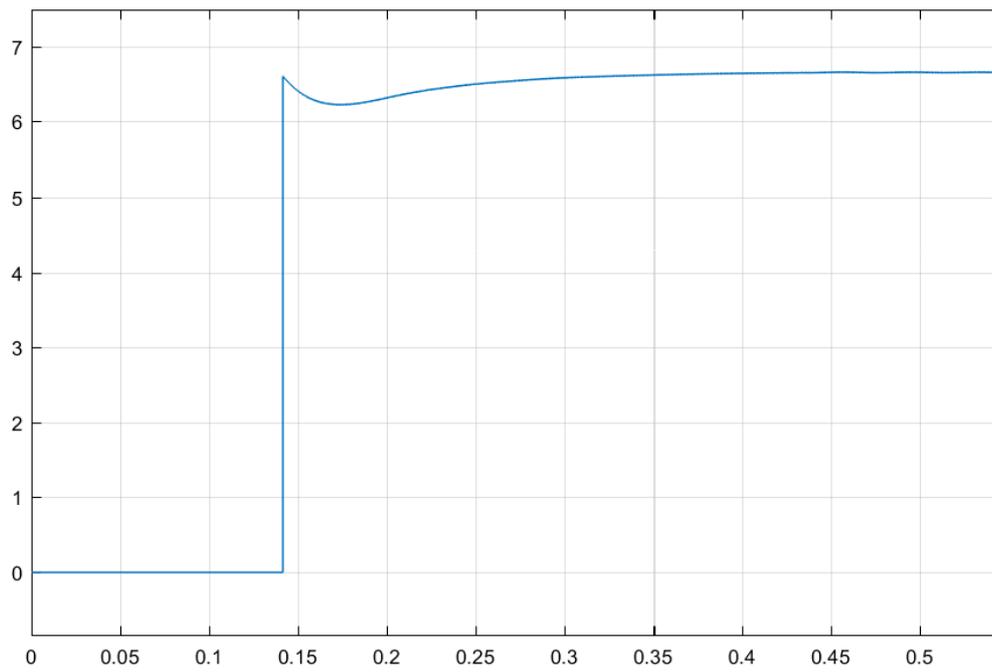


Ilustración 44. Intensidad a la salida I_0

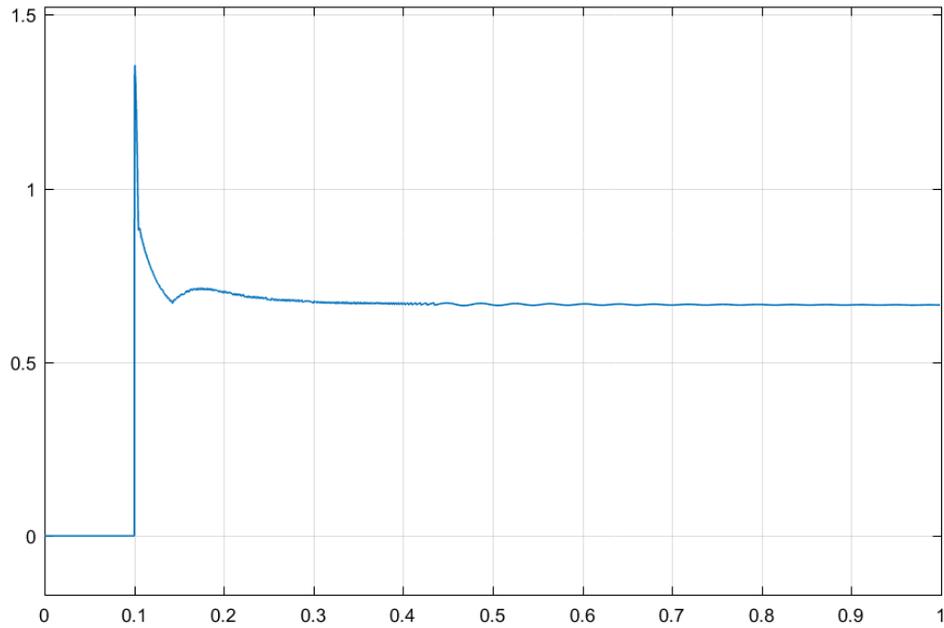


Ilustración 45. Señal D_s (duty cycle) salida del controlador.

2.4.2 Processor in the Loop (PIL)

Una simulación de Procesamiento en Bucle (Processor in the Loop *PIL*), genera código fuente, para después descargarlo y ejecutarlo en el hardware de destino. Una simulación *PIL* puede generar el código fuente de una parte o todo el modelo.

Una simulación *PIL* permite:

- Comprobar si el modelo y el código generado son numéricamente equivalentes.
- Descargar, iniciar y detener la aplicación en el equipo de destino.
- Comunicación en diferido entre Simulink® y el dispositivo.

Las pruebas preliminares consisten en la simulación mediante *PIL* de un programa básico en el dispositivo de destino, el cual debe encontrarse conectado al PC. Los pasos del Anexo I de configuración de Software deben haberse seguido para un correcto funcionamiento [19].

Una vez cumplidos dichos pasos, se comprobará el correcto funcionamiento con un modelo de ejemplo, escribiendo “*c2000_pil_block*” en la ventana de comandos de Matlab®.

Esta prueba preliminar permitirá identificar una configuración inicial de los parámetros de Simulink® para ejecución de un programa genérico en el dispositivo de destino.

Una vez abierto el modelo, se deberán establecer los parámetros de configuración requeridos para su correcta ejecución:

- Solver Options
 - Type: Fixed-step; Solver: discrete (o auto).
 - Se fijará el tamaño de Fixed Step a un valor apropiado, por ejemplo, $1e^{-4}$.
- Hardware Implementation:
 - Hardware board: TI Delfino F2833x.
 - De esta forma, se tomarán las herramientas de generación de código específicas para nuestro modelo de microcontrolador.
 - Build Option: Build.
 - Al desplegar el trozo de modelo deseado generará el código correspondiente, sin ejecutar directamente en el microcontrolador.
 - Target Hardware Resources > Groups > Clocking
 - Se establece el Oscillator Clock frequency en 30 MHz.
- Code Generation:
 - Create Block: PIL
 - Al desplegar el trozo de modelo deseado, se creará un bloque en una ventana emergente representativo del código generado.

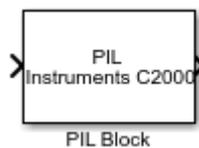


Ilustración 46. Máscara de Bloque PIL.

- Toolchain Settings: Texas Instruments Code Composer Studio v6 (C2000).
 - Seleccionar la versión utilizada de Code Composer permitirá una correcta comunicación con el dispositivo. En caso contrario, generará el siguiente mensaje:

The model is using Texas Instruments Code Composer Studio vX (C2000) toolchain, but your support package is not set up to use that toolchain. To configure this toolchain, type 'targetupdater' in MATLAB®.
 - Conecta el microcontrolador al ordenador vía USB. Para la ejecución del modelo en PIL se realizan los siguientes ajustes:

- Establecer conexión serie vía puerto COM. Para más detalle de cómo configurar esta conexión visitar esta página. Para el caso de conectarse al COM4, introducir en Matlab®:

```
setpref('MathWorks_Embedded_IDE_Link_PIL_Preferences','COMPort','COM4');
```

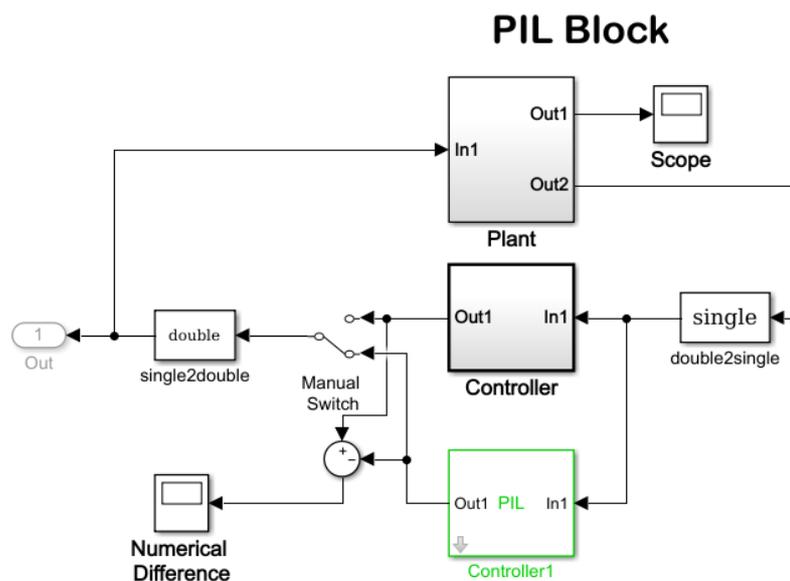
- Fijar velocidad de transmisión acorde con el dispositivo (ver Administrador de Dispositivos) introduciendo:

```
setpref('MathWorks_Embedded_IDE_Link_PIL_Preferences','BaudRate',115200);
```

- Habilitar PIL sobre serie mediante el siguiente comando:

```
setpref('MathWorks_Embedded_IDE_Link_PIL_Preferences','enableserial', true);
```

Tras haber configurado correctamente la baud rate, COM port, y comprobar que el interruptor conecta el bloque PIL a la salida, se puede ejecutar con éxito la simulación.



Copyright 2015 The MathWorks, Inc.

Ilustración 47. Ejemplo de uso del software para verificación y validación de código mediante PIL [16].

Con este ejemplo, se puede comprobar cómo la diferencia entre el controlador y el bloque de código es nula.

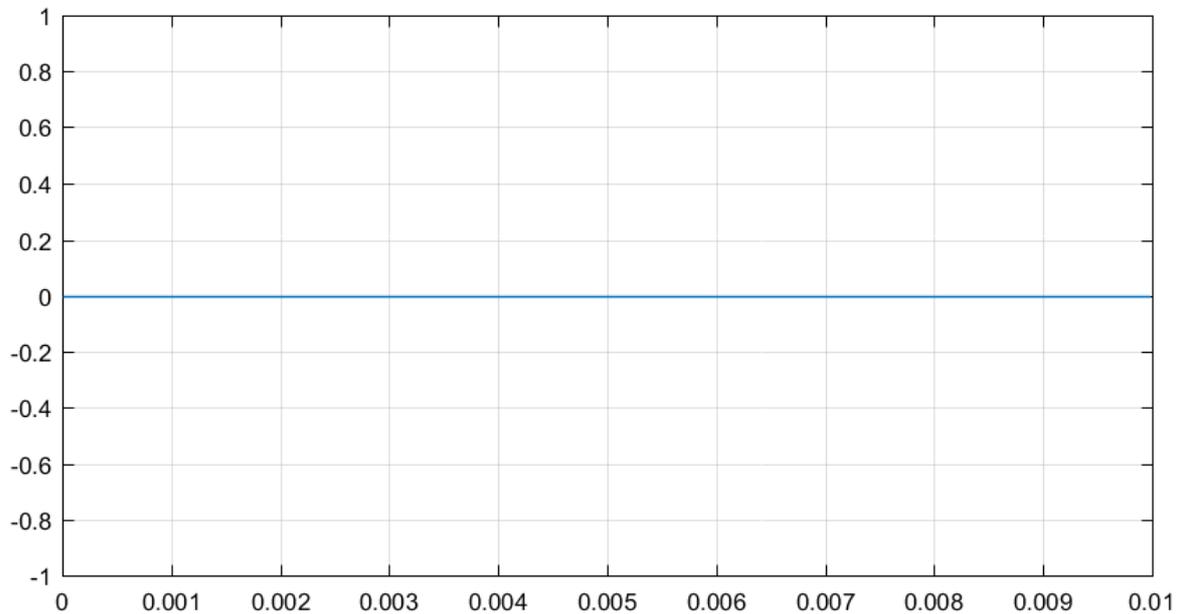


Ilustración 48. Señal diferencial entre salida del control simulado y bloque PIL.

Una vez se dispone de la configuración básica de Simulink®, se pasa a la creación del bloque PIL para el controlador. Para ello, se hace necesario establecer un retraso en la ejecución de dicho programa con respecto del resto del modelo, evitando así desajustes en el control. Esto se realizará mediante un bloque “enable”, que habilitará el control una vez pasados 0.1 s.

En primera instancia, se obtiene el siguiente error:

Data type "double" is not supported for SIL or PIL execution with the current target configuration. This is because the host size (8 bytes) and target size (4 bytes) do not match. To avoid this error, do not use this data type at the component interface.

Por tanto, se deberán cambiar de formato todos los datos de tipo double del bloque de control antes de generar el bloque PIL. Además, se suprimen los parámetros de este tipo (fs).

A continuación, se presentan en las Ilustraciones 49, 50 y 51, los resultados obtenidos de las pruebas realizadas con estas configuraciones en PIL.

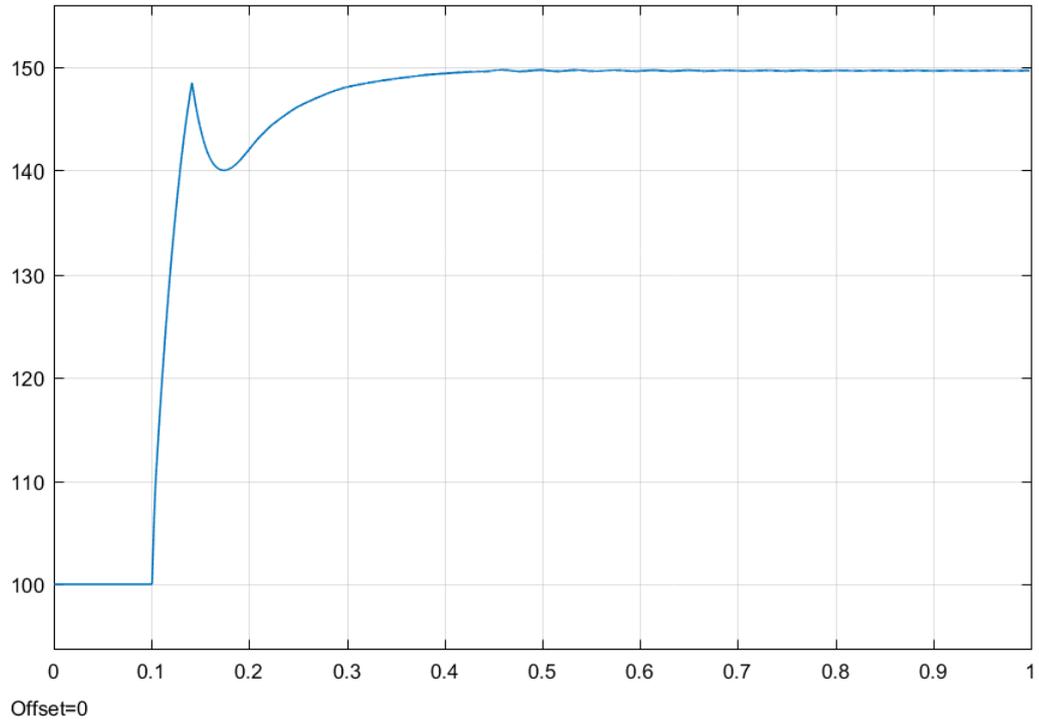


Ilustración 49. Tensión en la carga V_0 .

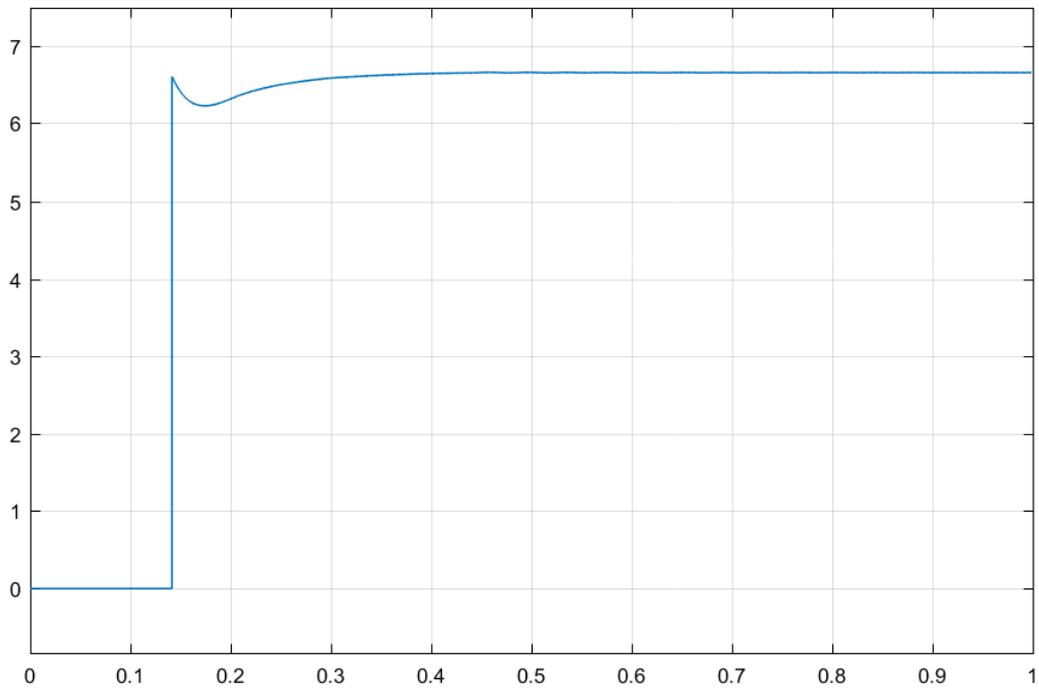


Ilustración 50. Corriente en la carga I_0 .

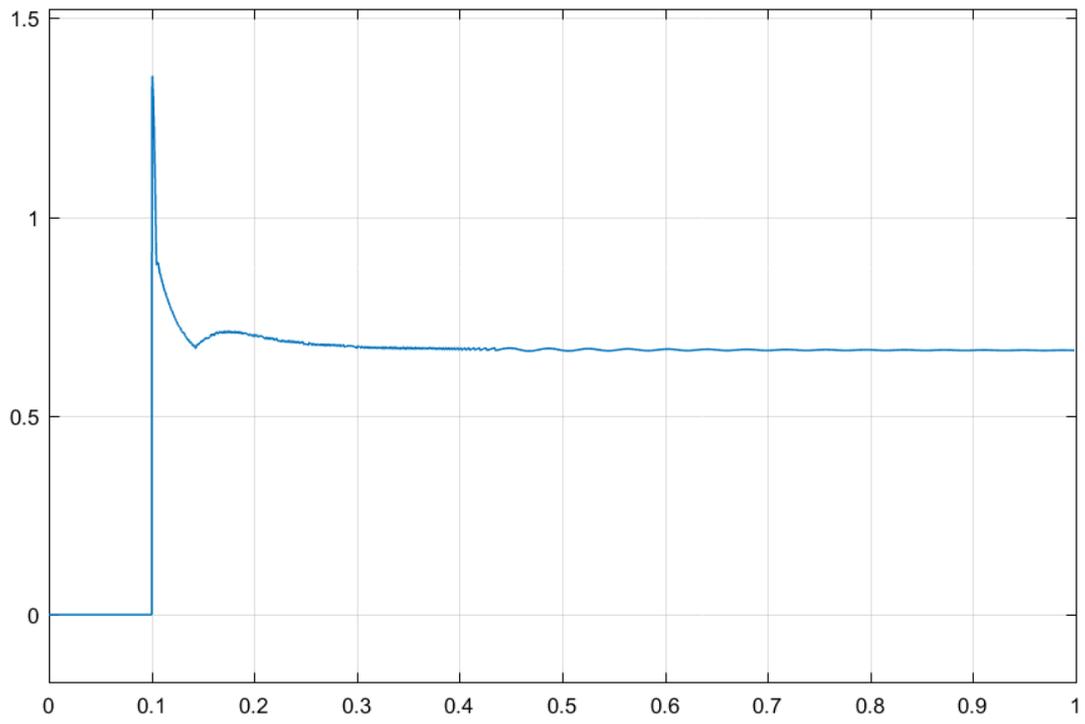


Ilustración 51. Señal D_S (duty cycle) salida del controlador.

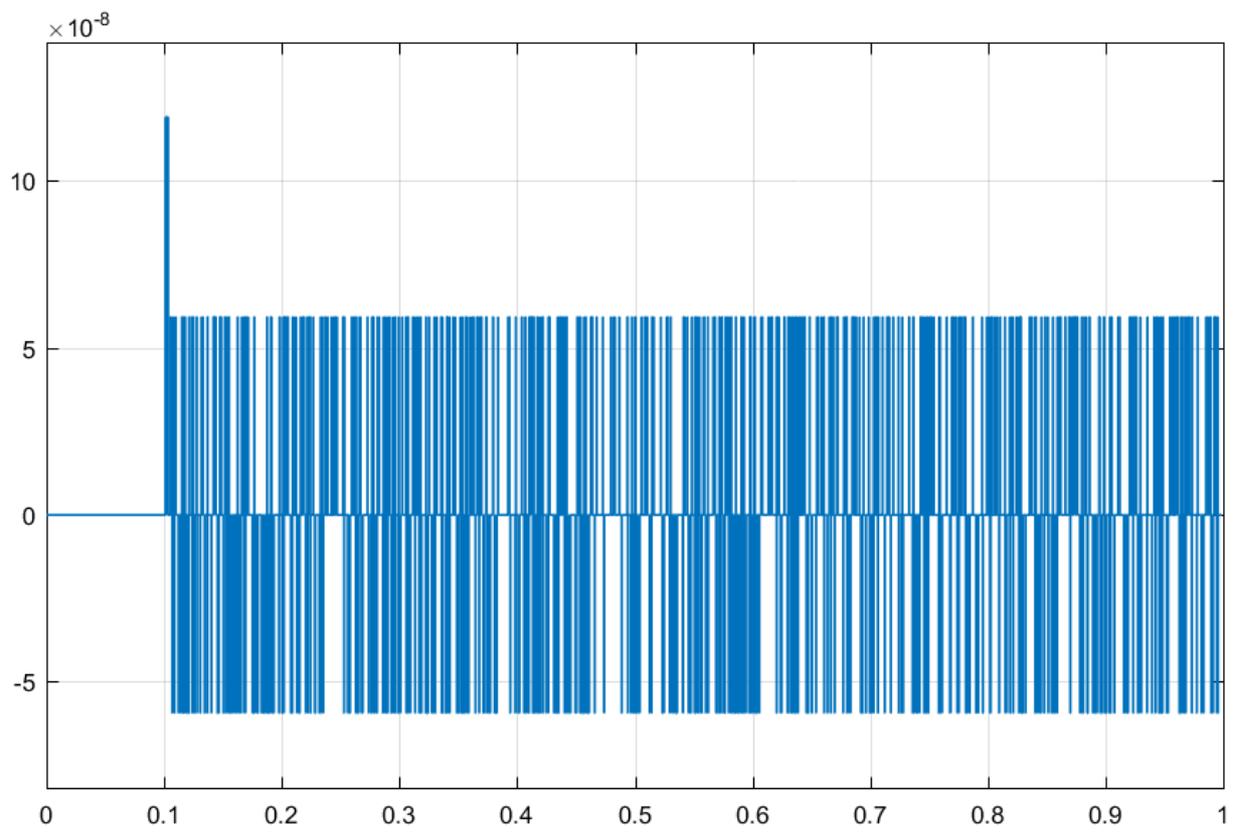


Ilustración 52. Diferencia entre el bloque de control SIL y PIL.

Se observa en la Ilustración 52 una diferencia periódica de $6 * 10^{-8}$ en el Duty Cycle a la salida del control ejecutado por ambos métodos debido al retraso en la ejecución en hardware con respecto al de software.

2.4.3 External Mode

El *External Mode* se trata de un modo de ejecución de pruebas en el dispositivo DSP que permite la comunicación en tiempo real entre este y el ordenador. Este modo de ejecución, cumple las siguientes funciones a la hora de realizar pruebas de programa [11]:

- Agilidad a la hora de realizar modificaciones del programa, ya que estos se pueden efectuar en tiempo real desde los bloques Simulink®.
- Visualización en tiempo real de la evolución de la señal y su respuesta ante los ajustes efectuados.

Este modo de ejecución tiene unos grandes requerimientos de memoria RAM del dispositivo, por lo que para poder realizar las pruebas habría que bajar considerablemente la resolución del muestreo de las señales. De esta forma, no permite el correcto control y lectura de las señales.

2.5 Implementación en equipo real

Para la implementación de la ejecución del programa con el equipo real, se hace necesario el establecer un protocolo de ejecución y comunicación con dicho sistema. Para esto, en primer lugar, es necesario habilitar las interrupciones en cada ciclo de reloj, de forma que se pueda realizar la conversión analógico digital de las señales del equipo para su procesamiento por parte del microcontrolador.

Para poder ejecutar en el mismo microcontrolador el control y la máquina de estados, se deberá configurar la prioridad entre estas. Se hace diferenciación entre tareas periódicas, que serán aquellas con alta prioridad y tareas sin prioridad. Estas últimas se ejecutarán en el tiempo de idle, que se definirá a continuación. Con este fin, se programan las interrupciones.

El funcionamiento de este procesado de información es el siguiente: El ADC habilita la interrupción a partir de las señales del PWM. Cuando la señal del PWM pasa por cero, este lanza el ADC, el cual habilita la interrupción una vez ha terminado la conversión, es decir, el fin de conversión del ADC lanza la interrupción. Esta interrupción se habilita internamente en el microcontrolador.

La máquina de estados se ejecuta en el idle time. El Idle Time es el tiempo restante entre interrupciones, es decir, el que excede a la suma entre el tiempo de medida y cálculo del ADC. El ADC tarda en medir las señales de entrada unos $\approx 5\mu s$ aproximadamente, mientras que puede tardar de 10 a $100\mu s$. Se debe configurar una frecuencia de muestreo tal que esta conversión se pueda realizar en un periodo. Si se establece el tiempo de muestreo en $t_s = 250\mu s$, el tiempo restante es el Idle Time (*tiempo libre*). A continuación, en la Ilustración 53 se representa un ejemplo de este concepto:

$t_c + t_m < t_s$. Siendo:

- Tiempo de medida $t_m \cong 5\mu s$
- Tiempo de cálculo $t_c \cong 100\mu s$
- Tiempo de muestreo o sampleo $t_s \cong 250\mu s$
- $t_s - (t_c + t_m) = t_{idle}$.

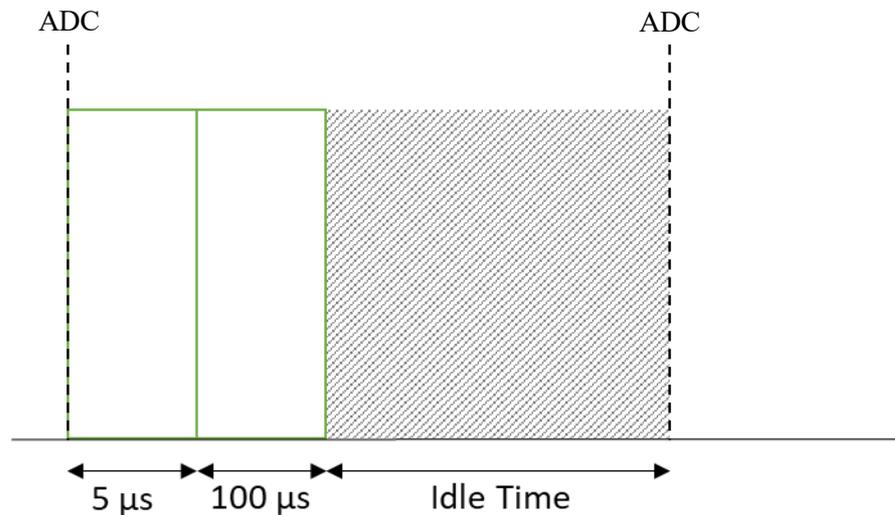


Ilustración 53. Tarea de Idle.

Para el diseño de este proceso, se utilizan los siguientes bloques de Simulink®:

- C28x Hardware Interrupt. Indica qué función lanzará la interrupción y el modo. Hay que configurar los siguientes parámetros:

- PIE/CPU interrupt numbers. Se configura el vector correspondiente a los diferentes periféricos del dispositivo. En este caso [1, 1], de forma que se realizará a través de la entrada del ADC, *ADCINT1*.

PIE and CPU Interrupt Numbers for F280x, F2802x, F2803x, F2805x, F2806x, F2833x, F28M35x, and F28M36x Processors

PIE ⇒ CPU #	1	2	3	4	5	6	7	8
1	SEQ1INT (ADC) / ADCINT1	SEQ2INT (ADC) / ADCINT2	Reserved	XINT1	XINT2	ADCINT / ADCINT9	TINT0 (TIMER 0)	WAKEINT (LPM/WD)
2	EPWM1_TZINT	EPWM2_TZINT	EPWM3_TZINT	EPWM4_TZINT	EPWM5_TZINT	EPWM6_TZINT	EPWM7_TZINT	EPWM8_TZINT
3	EPWM1_INT	EPWM2_INT	EPWM3_INT	EPWM4_INT	EPWM5_INT	EPWM6_INT	EPWM7_INT	EPWM8_INT
4	ECAP1_INT	ECAP2_INT	ECAP3_INT	ECAP4_INT	ECAP5_INT	ECAP6_INT	EPWM10_TZINT / HRCAP1_INT	EPWM9_TZINT / HRCAP2_INT
5	EQEP1_INT	EQEP2_INT	EQEP3_INT	HRCAP3_INT	HRCAP4_INT	Reserved	EPWM10_INT	EPWM9_INT
6	SPIRXINTA (SPI- A)	SPITXINTA (SPI-A)	SPIRXINTB (SPIB_RX) / MRINTB (McBSP- B)	SPITXINTB (SPIB_TX) / MXINTB (McBSP- B)	SPIRXINTC (SPI- C) / MRINTA (McBSP-A_RX)	SPITXINTC (SPI- C) / MXINTA (McBSP-A_TX)	SPIRXINTD (SPI- D) / EPWM12_TZINT	SPITXINTD (SPI-D) / EPWM11_TZINT
7	DINTCH1 (DMA1)	DINTCH2 (DMA2)	DINTCH3 (DMA3)	DINTCH4 (DMA4)	DINTCH5 (DMA5)	DINTCH6 (DMA6)	EPWM12_INT	EPWM11_INT
8	I2CINT1A	I2CINT2A	Reserved	Reserved	SCIRXINTC (SCI- C)	SCITXINTC (SCI- C)	Reserved	Reserved
9	SCIRXINTA (SCIA_RX)	SCITXINTA (SCIA_TX)	SCIRXINTB (SCIB_RX) / LINA_INT0	SCITXINTB (SCIB_TX) / LINA_INT1	ECAN0INTA (CAN_A_1)	ECAN1INTA (CAN_A_2)	ECAN0INTB (CAN_B_1)	ECAN1INTB (CAN_B_2)
10	EPWM9_TZINT / ADCINT1	EPWM10_TZINT / ADCINT2	EPWM11_TZINT / ADCINT3	EPWM12_TZINT / ADCINT4	EPWM13_TZINT / ADCINT5	EPWM14_TZINT / ADCINT6	EPWM15_TZINT / ADCINT7	EPWM16_TZINT / ADCINT8
11	CLA1_INT1 / EPWM9_INT7 / MTOCIPCINT1	CLA1_INT2 / EPWM10_INT / MTOCIPCINT2	CLA1_INT3 / EPWM11_INT / MTOCIPCINT3	CLA1_INT4 / EPWM12_INT / MTOCIPCINT4	CLA1_INT5 / EPWM13_INT	CLA1_INT6 / EPWM14_INT	CLA1_INT7 / EPWM15_INT	CLA1_INT8 / EPWM16_INT

Ilustración 54. Número de PIE [15].

- Simulink task priorities. Establece la prioridad de las interrupciones a procesar asíncronamente. Los valores tienen relación inversa con la prioridad del evento, es decir, para dar una mayor prioridad a uno de ellos, se debe establecer un valor menor que aquellos eventos de menor prioridad.
- Preemption flags. Determina si una tarea se puede anticipar a otra de mayor prioridad.

C28x Interrupt Block (mask) (link)

Create Interrupt Service Routine which will execute the downstream subsystem.

Note: The default model base sample rate priority is set to 40 with a lower priority value indicating a higher priority task. These parameters can be changed in the 'Solver' pane of the 'Configuration Parameters'. The Simulink task priority of the selected interrupt is relative to the model base rate priority settings.

Parameters

CPU interrupt numbers:
[1]

PIE interrupt numbers:
[1]

Simulink task priorities:
[30]

Preemption flags: preemptible-1, non-preemptible-0
[0]

Enable simulation input

Ilustración 55. Configuración del bloque de Interrupción.

- PWM. El PWM es la señal que modula por ancho de pulsos el sistema. Además, cada vez que pasa por 0, lanza el ADC. El PWM tendrá una frecuencia de 4 kHz. Se debe configurar el timer period, según Datasheet, este será $CLOCKSYS/2 = 64000$.

El parámetro CMPA, establece el valor con el que se compara la señal triangular para definir el ancho de los pulsos del PWM.

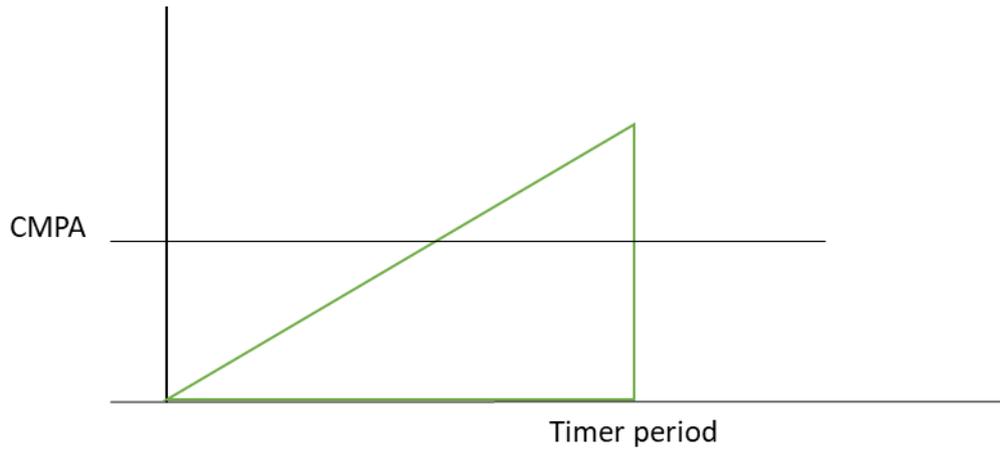


Ilustración 56. Concepto de CMA del PWM.

C280x/C2833x ePWM (mask) (link)
 Configures the Event Manager of the C280x/C2833x DSP to generate ePWM waveforms.

General ePWMA ePWMB Deadband unit Event Trigger PWM chopper control Trip Zone unit

Allow use of 16 HRPWMs (for C28044) instead of 6 PWMs

Module: ePWM1

Timer period units: Clock cycles

Specify timer period via: Specify via dialog

Timer period: 64000

Reload for time base period register (PRDL): Counter equals to zero

Counting mode: Up

Synchronization action: Disable

Specify software synchronization via input port (SWFSYNC)

Synchronization output (SYNCO): Disable

Time base clock (TBCLK) prescaler divider: 1

High speed clock (HSPCLKDIV) prescaler divider: 1

Ilustración 57. Configuración general del PWM.

CMPA units:

Specify CMPA via:

CMPA value:

Reload for compare A Register (SHDWAMODE):

Action when counter=ZERO:

Action when counter=period (PRD):

Action when counter=CMPA on up-count (CAU):

Action when counter=CMPA on down-count (CAD):

Action when counter=CMPB on up-count (CBU):

Action when counter=CMPB on down-count (CBD):

Compare value reload condition:

Add continuous software force input port

Continuous software force logic:

Reload condition for software force:

Ilustración 58. Configuración de la señal de salida del PWM.

- ADC. Se parametriza para la realización de la función de conversión analógico digital de las señales analógicas, es decir, procedentes del equipo que entran al dispositivo [11].

El módulo ADC del dispositivo al que se conectarán estas señales será el módulo ADCINT0.

Module:

Conversion mode:

Start of conversion:

Sample time:

Data type:

Post interrupt at the end of conversion

Ilustración 59. Configuración del control del ADC.

Number of conversions: 1

Conversion no. 1 ADCINA0

Use multiple output ports

Ilustración 60. Configuración de la entrada del ADC.

- Idle Task. Define la función que se ejecutará en el tiempo de Idle. Se configura mediante:
 - o Task number, número de funciones que va a meter en el tiempo de idle y prioridad. La posición del vector indica el número de función, y el valor, su orden.
 - o Preemption flag: define si una tarea se puede anticipar a una interrupción.

Idle Task (mask) (link)

Create a free-running task which will execute the downstream subsystem.

Parameters

Task numbers:

[1]

Preemption flags: preemptible-1, non-preemptible-0

[1]

Enable simulation input

Ilustración 61. Configuración de la Tarea de Idle.

En el tiempo de idle debe ir la máquina de estados (tarea no prioritaria). En el código fuente del programa, la tarea de idle irá en el main.

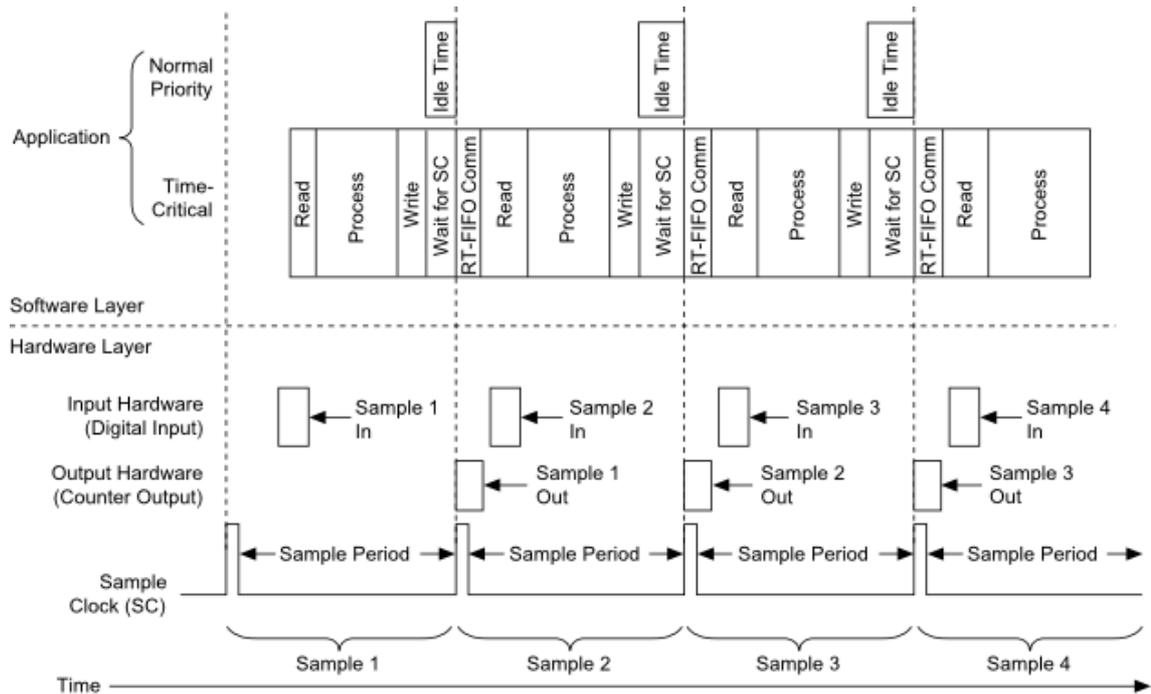
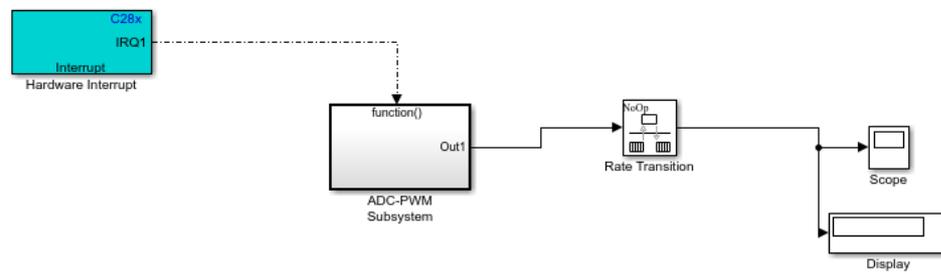


Ilustración 62. Cronograma representativo de los Idle Time, Interrupciones, Tiempo de Muestreo [16].

Para realizar las pruebas de la configuración base de las interrupciones, se realizarán los ajustes necesarios sobre el ejemplo de Simulink®: c280x_2833x_adcpwmasynctest_ert.slx. El ejemplo muestra cómo se habilita la interrupción a partir del final de conversión del ADC, que es lanzado por el pulso del PWM. Para esto, se hace uso de un bloque tipo “Hardware Interrupt”, así como de un bloque tipo “ADC” y otro “PWM” con las configuraciones representadas en las Ilustraciones 57, 58 y 59 [11].

ADC-PWM Synchronization via ADC Interrupt



Copyright 2007-2019 The MathWorks, Inc.

Ilustración 63. Sincronización ADC-PWM con bloque ADC Interrupt [21].

Con las configuraciones de los bloques mostradas previamente, se ejecuta el programa en modo PIL. El ADC habilita las interrupciones a partir de las señales del PWM, pero el PWM no genera la señal de salida por el pin del microcontrolador, por lo que no se podrá implementar el control del convertidor en el equipo real mediante esta técnica, con el dispositivo utilizado.

Conclusiones

La programación de microcontroladores en lenguaje de alto nivel mediante Simulink®, dota a la industria de las renovables de una herramienta útil y accesible para el diseño y ajuste de los mecanismos de control y conversión de la potencia.

Mediante la realización de una configuración base del software, se facilita la creación de código mediante las herramientas integradas de Simulink® para cada modelo de microcontrolador.

A la creciente demanda de perfiles especializados en energías renovables, se suma la necesidad de optimizar dichos procesos de producción de potencia. La gran accesibilidad del lenguaje de bloques para estos perfiles técnicos, permite profundizar en el estudio de diferentes topologías de dispositivos de potencia.

Sin embargo, la implementación del control en el convertidor real requiere de un mayor estudio de la tecnología utilizada, en cuanto a los métodos de comunicación con el equipo real a través del microcontrolador.

Anexo I

Software Requerido

- Code Composer Studio (CCS).
- Se recomienda CCSv6.
- Matlab®.
- Se recomienda R2015b o superior. (Se utiliza el R2017a para esta demostración)
- Simulink.
- Simulink Coder.
- Embedded Coder.
- MATLAB® Coder.
- Texas Instruments controlSUITE.
- FTDI Drivers acordes a la versión de Windows 10¹.

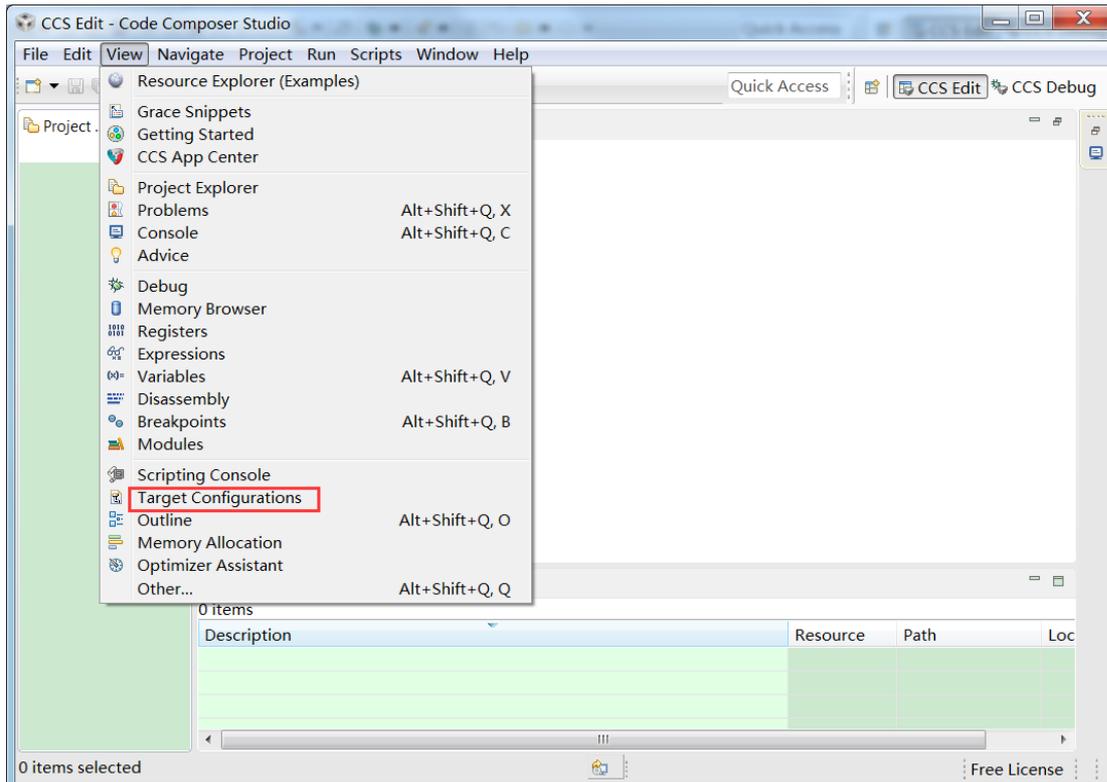
Hardware

- TMS320F28335 Experimenter Kit.

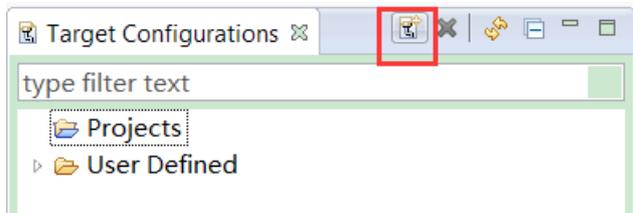
Conexión de CCS al dispositivo

- Abrir CCS > View > Target Configurations.

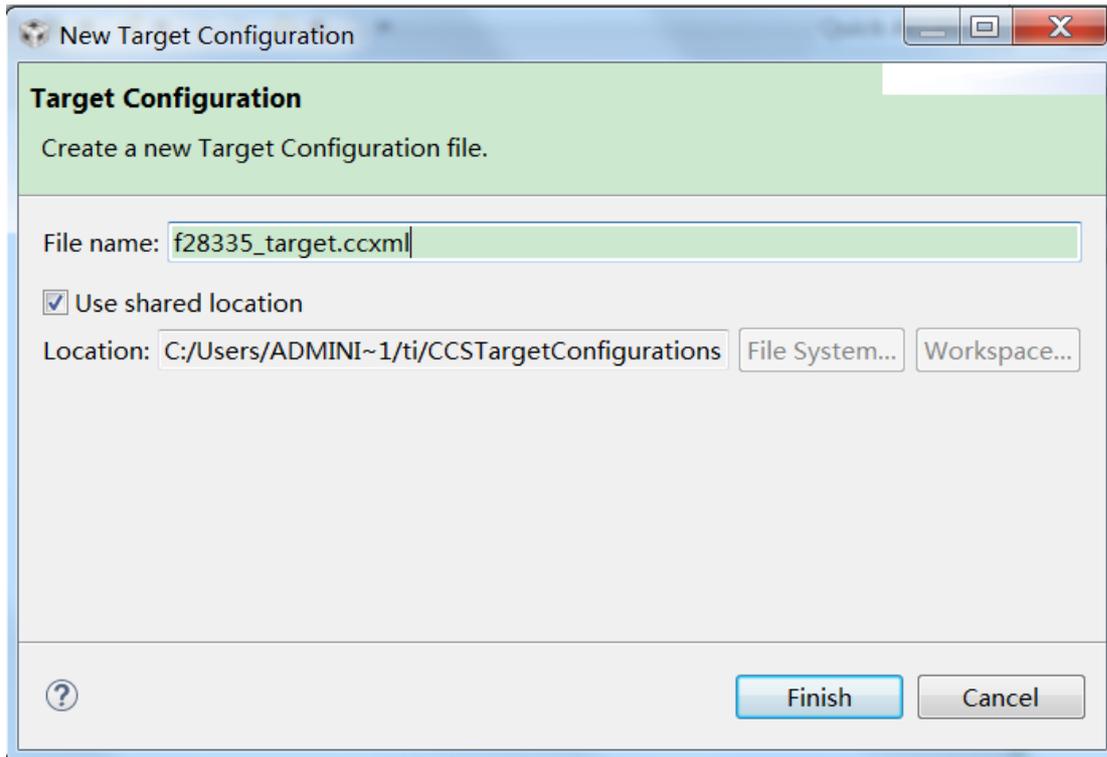
¹ Cualquier actualización del sistema (Windows 10) puede afectar a los drivers del controlador.



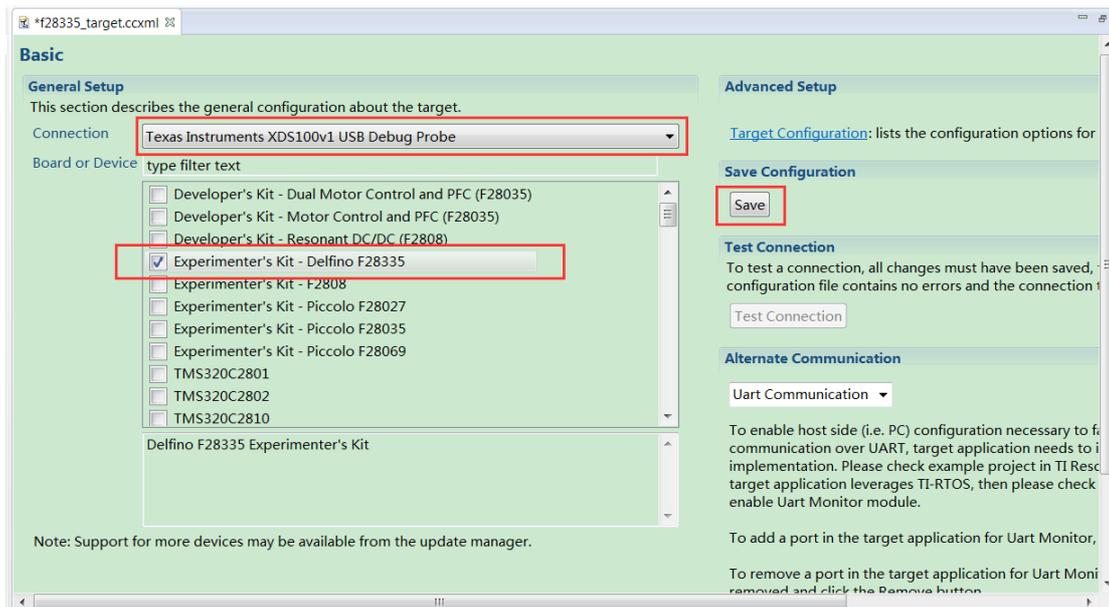
- Pulsar "New Target Configuration File"



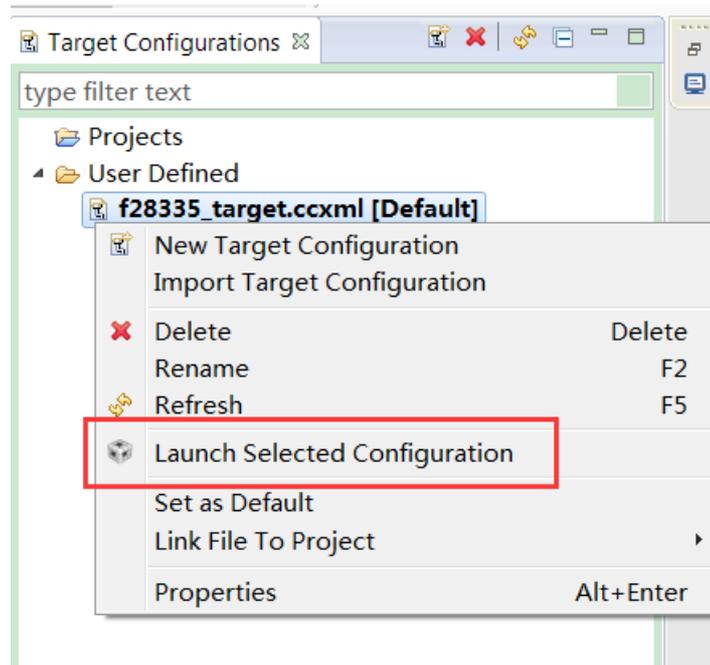
- Crear nuevo fichero de extensión ".ccxml", marcar "Use shared location", pulsar finalizar.



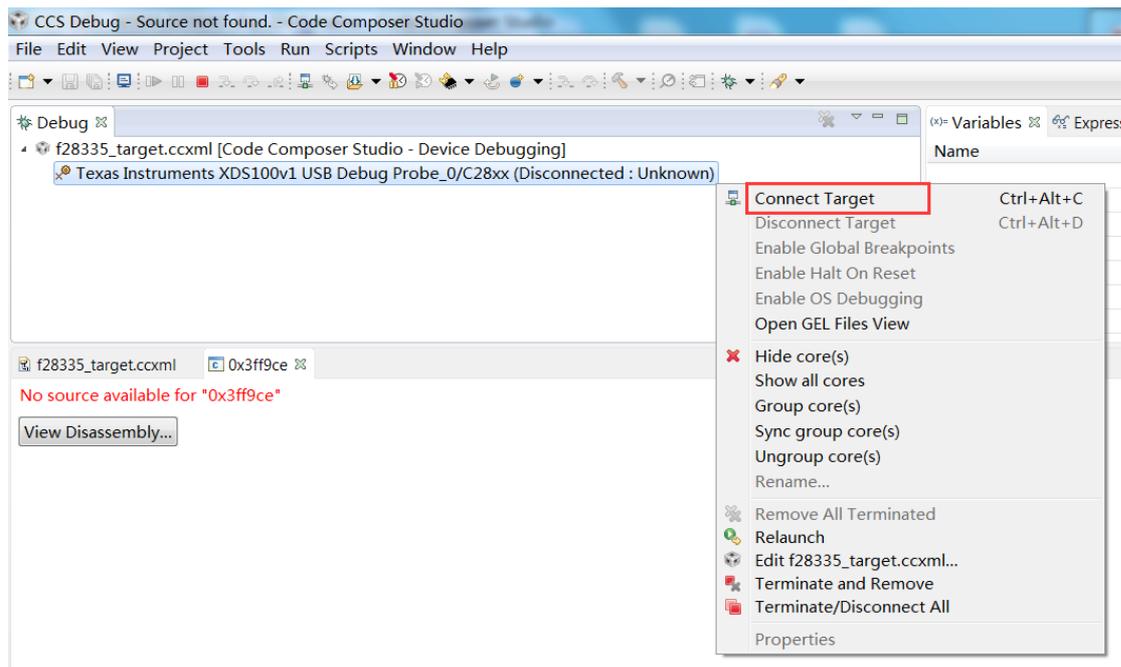
- Seleccionar el tipo de conexión y base, guardar.



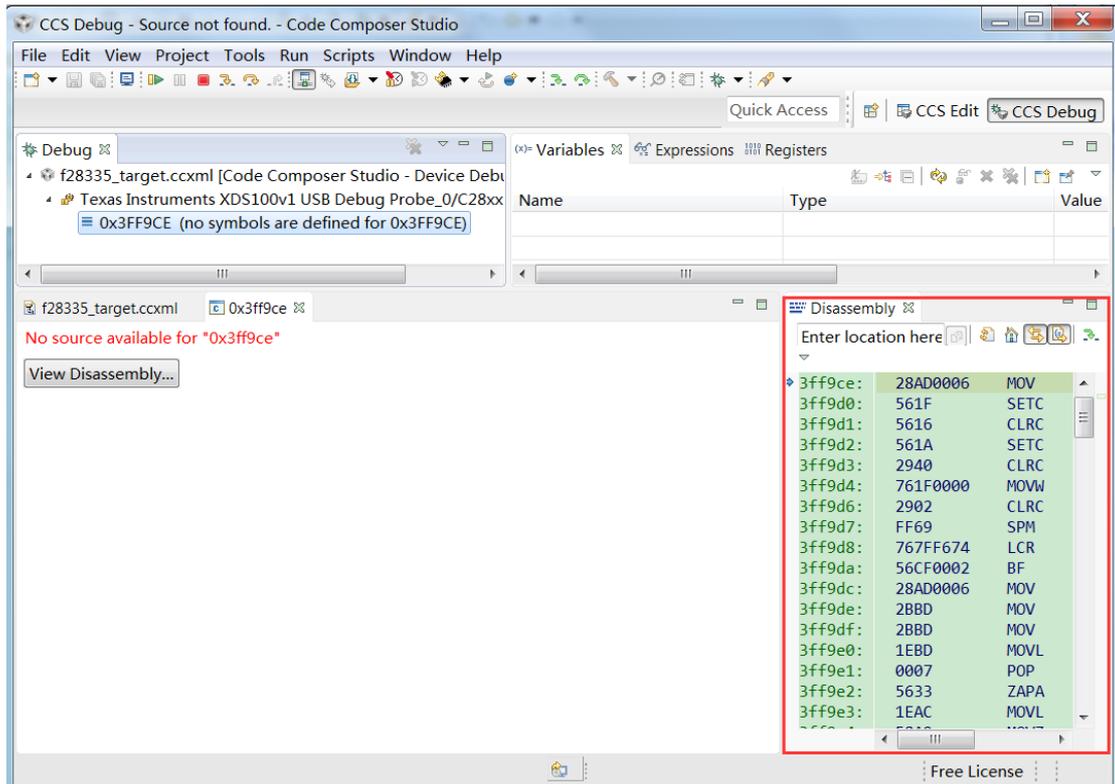
- Click derecho en el fichero “.ccxml” que se acaba de crear, y a continuación sobre “Launch Selected”. (El dispositivo de destino debe estar conectado al PC mediante JTAG y encendido).



- Seleccionar la opción “Connect target”.



- Se abrirá el cuadro de desensamblado, indicado que se ha realizado satisfactoriamente la conexión entre el CCS y el dispositivo de destino.

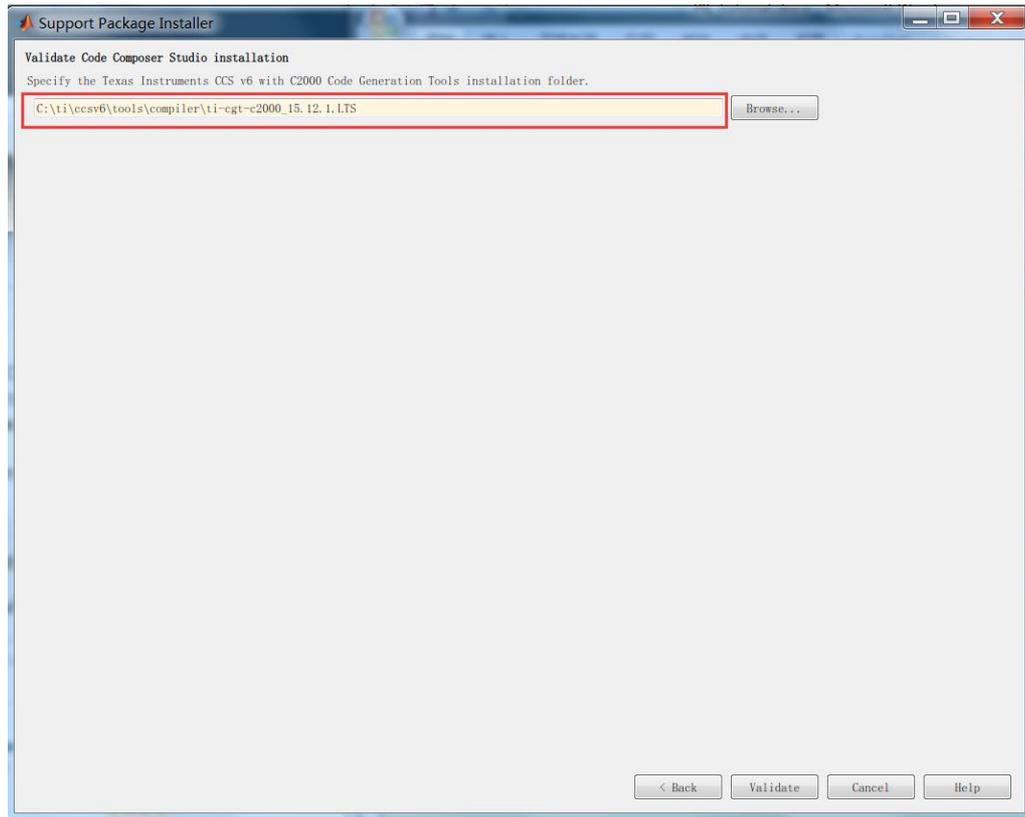


Instalación de "Embedded Coder Support Package for Texas Instruments C2000 Processors".

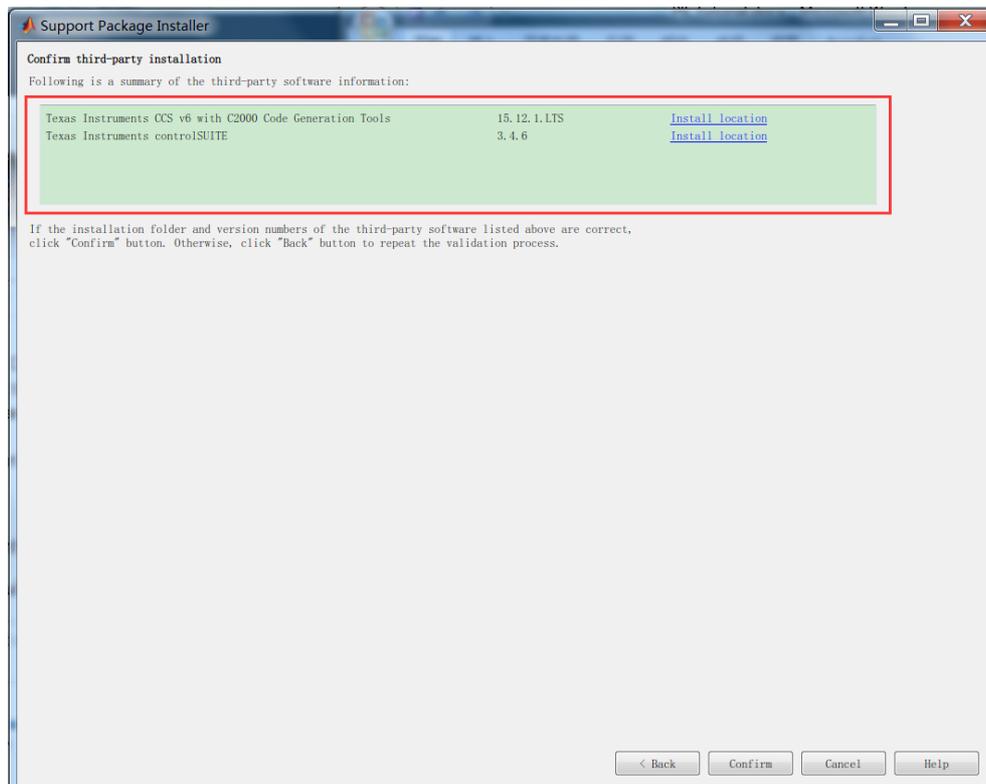
- Abrir Matlab® > HOME > Add-Ons > Get Hardware Support Packages.
- Buscar "Embedded Coder Support Package for Texas Instruments C2000 Processors" y seleccionarlo.
- Pulsar sobre "Install".



- Tras instalarlo, pulsar sobre "Setup Now".
- Pulsar "Next".
- Seleccionar la versión correcta de CCS (CCS v6), después pulsar "Next".
- Confirmar que se ha instalado el software de terceros necesario y pulsar sobre "Next".
- Configurar la ruta del compilador c2000_6.2.9 de CCS y pulsar sobre "Validate".



- Configurar la ruta de controlSUITE y pulsar "Validate".
- Comprobar las versiones detectadas de las herramientas de TI y controlSUITE y pulsar "Confirm".



- Pulsar "Finish". El pack está instalado.

Configuración del sistema

- Comprobar que el número de COM port es el mismo al establecido en Matlab®.
- Comprobar que la Baud Rate coincide con la establecida en Matlab
- Crear los modelos de Simulink® en carpetas del sistema que no contengan ningún espacio en su ruta. Preferentemente, cercanos a la raíz.
- Crear una carpeta para cada programa creado, ya que, al construir el código, se generan múltiples archivos que podrían ser sobrescritos al generar el de otro programa.

Bibliografía

- [1] Red Eléctrica de España S.A., «El Sistema Eléctrico Español,» 2017.
- [2] E. B. P. Robert Piqué López, «Electricidad, electrónica de potencia y sostenibilidad,» 2010.
- [3] Carlos Rivas Pereda, «Electrónica de potencia, tecnología estratégica para la sostenibilidad,» Redacción Energía de Hoy, 2018.
- [4] Howard H. Gerrish, William E. Dugger Jr. y Richard M. Roberts, *Electricity & Electronics*, 10th Edition, Goodheart-Willcox, 2009.
- [5] D. Frankel, A. Perrine y D. Pinner, «How Solar Energy can (finally) create value,» *McKinsey & Company*, Octubre 2016.
- [6] A. Panchbhai, A. Chauhan y D. H. Shah, Microcontroller based solar power system with grid synchronization., 2016.
- [7] B. W. Gunther Gridling, «Introduction to Microcontrollers,» Vienna University of Technology. Institute of Computer Engineering. Embedded Computing Systems Group., Viena, 2007.
- [8] T. Lambert, *Introduction to Microcontrollers and Embedded Systems.*, 2017.
- [9] E. T. S. d. I. D. d. I. Electrónica., *Apuntes de Sistemas Electrónicos Digitales. Tema 2.*, Sevilla, 2016.
- [10] Digy-Key Electronics, «digykey.com,» [En línea]. Available: <https://www.digikey.com/en/product-highlight/t/texas-instruments/c2000>. [Último acceso: 28 Agosto 2019].
- [11] Texas Instruments, TMS320F28335, TMS320F28334, TMS320F28332 Digital Signal Controllers (DSCs) Data Manual, 2007.
- [12] Mathworks, «Mathworks,» [En línea]. Available: <https://es.mathworks.com/>. [Último acceso: 30 Agosto 2019].

- [13] D. d. I. Electrónica., Apuntes de Electrónica de Potencia., Universidad de Sevilla, 2017.
- [14] M. M. Canteli, «Regulación, Control y Protección de Máquinas Eléctricas. Apuntes de la Asignatura. Convertidores electrónicos de potencia,» [En línea]. Available: <https://ocw.unican.es/pluginfile.php/1986/course/section/2310/convertidores.pdf>.
- [15] D. d. I. Electrónica, *Formulario Convertidores. Apuntes Asignatura Sistemas Electrónicos de Conversión de Potencia Grado en Ingeniería de la Energía.*, Sevilla: Universidad de Sevilla., 2017.
- [16] Mathworks, «<https://es.mathworks.com/>,» [En línea]. Available: <https://es.mathworks.com/>. [Último acceso: 01 Septiembre 2019].
- [17] Mathworks, «<https://es.mathworks.com/solutions/power-electronics-control/>,» [En línea]. Available: <https://es.mathworks.com/solutions/power-electronics-control/mppt-algorithm.html>. [Último acceso: 01 Septiembre 2018].
- [18] V. P. Cordero y J. M. Ruiz González, «Simulación mediante Psim de algoritmos de seguimiento del punto de máxima potencia en sistemas fotovoltaicos,» 4 Julio 2014.
- [19] Departamento de Ingeniería Electrónica., Apuntes de Electrónica de Potencia., Sevilla: Universidad de Sevilla., 1999.
- [20] Mathworks, «<https://es.mathworks.com/help/>,» [En línea]. Available: <https://es.mathworks.com/help/supportpkg/beaglebone/examples/code-verification-and-validation-with-pil-and-external-mode.html>. [Último acceso: 01 Septiembre 2019].
- [21] Mathworks, «Mathworks,» [En línea]. Available: <https://es.mathworks.com/help/supportpkg/xilinxzynq7000ec/ug/external-mode.html>. [Último acceso: 01 Septiembre 2019].
- [22] Autor, «Este es el ejemplo de una cita,» *Tesis Doctoral*, vol. 2, n° 13, 2012.
- [23] O. Autor, «Otra cita distinta,» *revista*, p. 12, 2001.
- [24] L. A. D'Alessio¹, L. G. González^{1,2*}, R. O. Cáceres, Diseño y Construcción de un Convertidor Boost de Pequeña Potencia, con Propósitos, 2014.

- [25] ikerlan, «ikerlan.es,» 02 12 2016. [En línea]. Available: <https://www.ikerlan.es/noticias/noticia/sistemas-de-almacenamiento-y-electronica-de-potencia-claves-en-los-sectores-de-la-energia-y-el-transporte>. [Último acceso: 28 08 2018].
- [26] «FactorCO2.com,» FactorCO2, 14 Junio 2017. [En línea]. Available: <https://www.factorco2.com/es/las-emisiones-de-co2-derivadas-de-la-generacion-electrica-alcanzan-el-nivel-mas-bajo-en-espana-en-los-ultimos-diez-anos/noticia/1528>. [Último acceso: 29 Agosto 2019].
- [27] F. F. Valderrama, H. Moreno C y H. M. Vega, «Análisis, simulación y control de un convertidor de potencia tipo boost,» 2011.
- [28] J. J. Marulanda Durango, J. J. Ordóñez y L. F. Mosquera Machado, «Diseño y construcción de un convertidor tipo Boost con PWM ajustable .,» *Scientia et Technica Año XXII*, vol. 22, n° 1, p. 6, 23 Marzo 2017.