

Escuela de Ingeniería Industrial

TRABAJO FIN DE GRADO

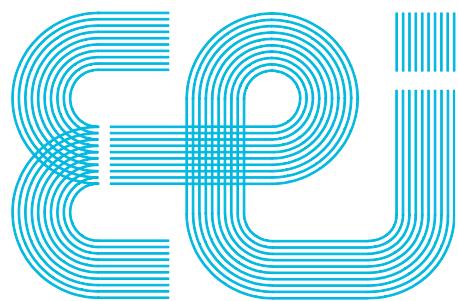
Implementación del filtrado de retraso temporal en tiempo real para un microcontrolador de 32 bits

Grado en Ingeniería en Electrónica Industrial y Automática

ALUMNO: Julio Benito Represa Montenegro

DIRECTORES: Gerardo Peláez Lourido

Universidade de Vigo



Escuela de Ingeniería Industrial

TRABAJO FIN DE GRADO

Implementación del filtrado de retraso temporal en tiempo real para un microcontrolador de 32 bits

Grado en Ingeniería en Electrónica Industrial y Automática

Documento

MEMORIA

Universidade de Vigo

CONTENIDO

Contenido	3
Índice de figuras.....	6
Índice de Tablas	9
1 Introducción	10
2 Objeto.....	11
2.1 Problema a solucionar	11
2.2 Objetivos.....	11
2.2.1 Objetivo general	11
2.2.2 Objetivos específicos.....	11
3 Historia.....	12
3.1 Input shaping	12
4 Contextualización	13
4.1 Descripción del sistema	13
4.1.1 Valores de las variables.....	13
5 Modelo dinámico	14
5.1 Fuerzas y modelado	14
5.1.1 Consideraciones iniciales	14
5.1.2 Variables.....	14
5.1.3 Método Euler-Lagrange	15
5.1.4 Método Newton-Euler.....	19
5.2 Frecuencia natural de vibración.....	20
5.3 Modelado dinámico en Simulink.....	21
6 Shapers	22
6.1 Que es el input shaping.....	22
6.2 Tipos de shapers	22
6.2.1 Introducción	22
6.2.2 Input Shaper ZV	22
6.2.3 Input Shaper ZVD	24
6.2.4 Input Shaper EI.....	24
6.2.5 Comparación de la robustez de los shapers.....	24
6.2.6 Comparación de los shapers en Matlab	25
6.3 Cálculo del shaper para este caso	27
6.4 Vectores de impulso	27
6.4.1 Definición de los vectores de impulso	28
6.4.2 Propiedades de los vectores de impulso	29

6.4.3 Vectores de impulso aplicados a este sistema.....	29
7 Método de diseño.....	31
7.1 Caja negra	31
7.2 Caja transparente	32
8 Valoración de alternativas para el diseño	33
8.1 Elementos necesarios.....	33
8.1.1 Unidad de control	33
8.1.2 Sensor de consigna de movimiento	34
8.1.3 Controlador del actuador	35
8.1.4 Actuador de posición.....	35
9 Elementos finales del diseño.....	37
9.1 Unidad de control	37
9.2 Actuador de movimiento	38
9.3 Controlador del motor	39
9.4 Interacción con el operario	40
10 Desarrollo.....	41
10.1 Esquema eléctrico.....	41
10.2 Recursos del microcontrolador.....	42
10.2.1 Motor	42
10.2.2 Potenciómetro.....	43
10.2.3 Pulsador	43
10.2.4 Interrupciones.....	44
10.2.5 Control de la dirección del motor.....	44
10.2.6 LED	45
10.2.7 Resumen de los recursos necesarios.....	45
10.2.8 Asignación de pines	45
10.3 Programación.....	46
10.3.1 Distribución del programa.....	46
10.3.2 Interrupciones	46
10.3.3 Interrupción del pulsador	47
10.3.4 Bucle principal	47
10.3.5 Cálculo de consignas	48
10.3.6 Interrupción de convolución.....	49
11 Simulación en Matlab/Simulink	50
11.1 Proceso completo.....	50
11.2 Virtualización del sistema	51
11.2.1 Partes del montaje	51
11.2.2 Ensamblaje	53
11.3 Simscape	54
11.3.1 Bloques.....	55

11.4 Comunicaciones.....	57
11.4.1 Transmisión de datos.....	57
11.4.2 Adquisición de datos	58
11.4.3 Procesado de datos	59
11.5 Video de la simulación	60
12 Resultados	61
12.1 Resultados sin filtro	61
12.1.1 Simulación.....	61
12.1.2 Gráficas	61
12.2 Resultados con filtro.....	63
12.2.1 Simulación.....	63
12.2.2 Gráficas	64
12.3 Retraso temporal.....	65
12.4 Resumen de los resultados.....	66
13 Conclusiones y futuros pasos.....	67
14 Bibliografía	68
Anexo I: Código.....	70
Anexo II: Hojas de características.....	77
Presupuesto.....	93

ÍNDICE DE FIGURAS

Figura 1: Brazo robótico de alta precisión de la marca KUKA. [25]	10
Figura 2: Grúa pórtico empleada por la empresa ferroviaria Adif y construida por la empresa JASO Industrial Cranes para mejorar el tráfico ferroviario en un centro logístico en Irún, punto transfronterizo a Francia. [18]	11
Figura 3: Esquema del sistema a estudiar, carro con masa oscilante. Elaboración propia.	13
Figura 4: Esquema del sistema a estudiar con las distintas variables y fuerzas dibujadas. Elaboración propia.....	14
Figura 5: Esquema del sistema a estudiar acotado y con las variables dibujadas. Elaboración propia.....	15
Figura 6: Esquema del sistema a estudiar con las variables necesarias para aplicar el método Newton-Euler. Elaboración propia.....	19
Figura 7: Diagrama de bloques para la simulación del modelo matemático en Simulink.....	21
Figura 8: Diagrama de bloques para la simulación del modelo matemático en Simulink.....	21
Figura 9: Oscilación de la carga (α) en el modelado dinámico del sistema en Simulink.	21
Figura 10: Diagrama de bloques habitual donde se implementa un prefiltro input shaper. Elaboración propia a partir del documento [10].....	22
Figura 11: Filtrado de una señal con un filtro ZV. Elaboración propia a partir del documento [10].	23
Figura 12: Comparación de la robustez del filtro ZV y filtro ZVD. Elaboración propia en Matlab.....	25
Figura 13: Robustez de los filtros ZV y ZVD cuando la frecuencia calculada se aleja un 7% de la frecuencia real. Elaboración propia.	26
Figura 14: Vector de impulso positivo en un sistema de segundo orden.....	28
Figura 15: Vector de impulso negativo en un sistema de segundo orden. Elaboración propia a partir del documento [10].	28
Figura 16: Espiral de amortiguamiento. Elaboración propia a partir del documento [10].	29
Figura 17: I_1	30
Figura 18: I_2	30
Figura 19: I_3	30
Figura 20: Vectores de impulso del filtro ZVD. Elaboración propia.	30
Figura 21: Modelo caja negra	31
Figura 22: Modelo caja transparente.....	32
Figura 23: PSoC 4 con el controlador a su derecha. [6]	33
Figura 24: Mando de control Curve AJQ de la marca Autec Radio Remote Control S.L. [23]	34
Figura 25: Potenciómetro lineal de la marca RS Components. [26].....	34
Figura 26: Circuito integrado L293D. [7].....	35
Figura 27: Circuito integrado L298N.	35
Figura 28: Motor paso a paso del Kit de evaluación Cypress Semiconductor PSoC 4 Pioneer Kit [27].	36
Figura 29: PSoC 5LP Arm Cortex-M3 [19].	37
Figura 30: Diagrama de los componentes del microcontrolador PSoC 5LP [20].....	38
Figura 31: Motor con reductora de escobillas de la marca RS PRO.	38

Figura 32: Controlador L298n [21].....	39
Figura 33: Esquema eléctrico del controlador L298n [22].	39
Figura 34: Joystick Keyes sJoys. [24].....	40
Figura 35: Esquema eléctrico. Elaboración propia.	41
Figura 36: Conexionado de los distintos elementos electrónicos. Elaboración propia. El programa para el diseño del conexionado es Fritzing.	42
Figura 37: Visualización del periférico PWM en el esquemático de PSoC Creator.	
Elaboración propia en el programa PSoC Creator.....	43
Figura 38: Visualización del módulo ADC en el esquemático de PSoC Creator. Elaboración propia en el programa PSoC Creator.	43
Figura 39: Esquema digital del comportamiento del módulo “debouncer”. Elaboración propia en el programa PSoC Creator.	44
Figura 40: Visualización del módulo “debouncer” en el esquemático de PSoC Creator.	
Elaboración propia en el programa PSoC Creator.....	44
Figura 41: Visualización de la interrupción de convolución en el esquemático de PSoC Creator. Elaboración propia en el programa PSoC Creator.....	44
Figura 42: Visualización de las salidas digitales de dirección de giro en el esquemático de PSoC Creator. Elaboración propia en el programa PSoC Creator.	45
Figura 43: Visualización la salida digital para controlar el LED en el esquemático de PSoC Creator. Elaboración propia en el programa PSoC Creator.....	45
Figura 44: Máquina de estados del programa del microcontrolador PSoC 5LP. Elaboración propia.....	46
Figura 45: Diagrama de flujo de la rutina de atención a interrupciones. Elaboración propia.	46
Figura 46: Diagrama de flujo de la función de interrupción del pulsador. Elaboración propia.	47
Figura 47: Máquina de estados para definir si aplicar el filtro o no. Elaboración propia.....	47
Figura 48: Diagrama que representa la posición del joystick junto a la consigna asociada para esa posición. Elaboración propia.....	48
Figura 49: Función para calcular de consigna en función del valor analógico. Elaboración propia.....	48
Figura 50: Diagrama de flujo de la función que se realiza en la interrupción de convolución. Elaboración propia.....	49
Figura 51: Esquema de los elementos necesarios para realizar la simulación.....	50
Figura 52: Maqueta de una “overhead crane” del laboratorio del profesor Gerardo Peláez, Universidad de Vigo.....	51
Figura 53: Despiece del diseño CAD. Elaboración propia en el programa de diseño SolidWorks.	52
Figura 54: Diseño CAD en SolidWorks de una “overhead crane”. Elaboración propia en el programa de diseño SolidWorks.	53
Figura 55: Vista en detalle de la juta rotacional que une el carro y el cable del sistema. Elaboración propia en el programa de diseño SolidWorks.	53
Figura 56: Comparación del diseño CAD (derecha) con la maqueta real (izquierda).	54
Figura 57: Diseño CAD simplificado para su simulación. Elaboración propia.....	54
Figura 58: Diagrama de bloques necesarios para la simulación en el entorno Simulink. Elaboración propia.....	55

Figura 59: Diagrama de bloques de los objetos fijos en el entorno Simulink. Elaboración propia.....	55
Figura 60: Diagrama de bloques de las juntas y objetos móviles en el entorno Simulink. Elaboración propia.....	56
Figura 61: Diagrama de bloques de las entradas y salidas de las juntas móviles en el entorno Simulink. Elaboración propia.....	56
Figura 62: Visualización del módulo “UART” en el esquemático de PSoC Creator. Elaboración propia en el porgrama PSoC Creator.....	57
Figura 63: Configuración del módulo “UART” en el esquemático de PSoC Creator.....	57
Figura 64: Comandos para la transmisión. Elaboración propia.....	58
Figura 65: Configuración y recepción de los datos en el programa PuTTY.	58
Figura 66: Configuración del puerto de entrada de datos. Elaboración propia.	58
Figura 67: Configuración de la actuación en la junta traslacional barra-carro.	59
Figura 68: Sistema de referencia representado en el diseño CAD. Elaboración propia en el programa de diseño SolidWorks.	59
Figura 69: Obtener el valor de la posición de consigna. Elaboración propia.	60
Figura 70: Video de las simulaciones realizadas. Elaboración propia.....	60
Figura 71: Trayectoria que sigue el punto inferior de la esfera sin aplicar el filtro ZVD. Elaboración propia en el programa de simulación Simscape Multibody.	61
Figura 72: Grafica de la posición del carro en la simulación de Simscape. Sin filtro ZVD. Elaboración propia.....	61
Figura 73: Gráfica de la señal de entrada que recibe Matlab del microcontrolador. Sin filtro ZVD. Elaboración propia.	62
Figura 74: Gráfica de la oscilación de la carga en la simulación de Simscape. Sin filtro ZVD. Elaboración propia.....	62
Figura 75: Comprobación experimental de la frecuencia natural del sistema a partir de la gráfica de la oscilación en la simulación de Simscape. Elaboración propia.	63
Figura 76: Trayectoria que sigue el punto inferior de la esfera aplicando el filtro ZVD. Elaboración propia en el programa de simulación Simscape Multibody.	63
Figura 77: Gráfica de la señal de entrada que recibe Matlab del microcontrolador. Con filtro ZVD. Elaboración propia.	64
Figura 78: Grafica de la posición del carro en la simulación de Simscape. Con filtro ZVD. Elaboración propia.....	64
Figura 79: Gráfica de la oscilación de la carga en la simulación de Simscape. Con filtro ZVD. Elaboración propia.	65
Figura 80: Comparación del tiempo requerido para el sistema con filtro y sin filtro. Elaboración propia.....	66

ÍNDICE DE TABLAS

Tabla 1: Valores de las variables del sistema a estudiar.....	13
Tabla 2: Cálculo de los vectores de impulso del filtro que se aplicará.....	30
Tabla 3: Tipos de señales que entran y salen del modelo caja negra.....	31
Tabla 4: Comparación de velocidades microcontroladores	33
Tabla 5: Características del motor.	39
Tabla 6: Asignación de pines a las entradas y salidas utilizadas.	45
Tabla 7: Despiece del diseño CAD.	52
Tabla 8: Resultados de la simulación.....	66

1 INTRODUCCIÓN

El control del movimiento ha sido una de las áreas de mayor interés y desarrollo para eliminar la vibración residual de sistemas que no interesa que oscilen. Desde la existencia de sistemas oscilatorios se han intentado crear sistemas de control para su amortiguamiento que han ido evolucionando hasta la actualidad.

Las nuevas tecnologías han derivado en que se requiera de sistemas mecatrónicos de posicionamiento de muy alta precisión (Figura 1), lo que ha provocado un avance en los sistemas de control para reducir las vibraciones residuales inducidas por el movimiento. Para intentar reducir las indeseadas oscilaciones se han desarrollado diversos sistemas de control:

- Feedback: Son sistemas con realimentación, siempre están limitados porque sus consignas vienen dadas por datos tomados en el pasado.
- Feedforward: Su objetivo es predecir el movimiento indeseado para así poder anularlo, estos sistemas de control no tienen las limitaciones de los anteriores, aunque son menos robustos ante desviaciones.



Figura 1: Brazo robótico de alta precisión de la marca KUKA. [25]

En este trabajo se pretenden estudiar los filtros de retraso temporal o “input shapers”, que se enmarcan dentro de los sistemas de control “feedforward”. Estos filtros tratan de predecir y eliminar las oscilaciones no deseadas de un sistema. Entre las muchas técnicas de filtración de señales, los filtros de retraso temporal disfrutan de distintas ventajas por su sencillez, por su optimización y por su capacidad para ser utilizados con comandos de actuadores arbitrarios en tiempo real.

2 OBJETO

2.1 Problema a solucionar

Este trabajo tiene como finalidad dar solución a un problema, dado el extenso uso de grúas en la industria, un aspecto muy importante es mitigar las oscilaciones de la carga para el movimiento seguro de mercancías. Además, si se reducen las oscilaciones indeseadas, aumenta la rapidez del proceso y permite una carga y descarga más eficiente. Esto reduciría la probabilidad de fallos mecánicos y aumenta la seguridad de trabajo de los operarios por ejemplo sería una ventaja para los estibadores en el movimiento de contenedores de mercancías (Figura 2).



Figura 2: Grúa pórtico empleada por la empresa ferroviaria Adif y construida por la empresa JASO Industrial Cranes para mejorar el tráfico ferroviario en un centro logístico en Irún, punto transfronterizo a Francia. [18]

2.2 Objetivos

2.2.1 *Objetivo general*

El objeto del presente proyecto es implementar un filtro de retraso temporal a un microcontrolador de 32 bits en tiempo real con la finalidad de obtener un sistema robusto que permita a un sistema formado por un carro y una masa en suspensión conectada al carro mediante un cable, moverse con las mínimas oscilaciones posibles.

2.2.2 *Objetivos específicos*

- Estudio de la dinámica del sistema.
- Estudio de los filtros de retraso temporal, tipos y funcionamiento de los mismos.
- Estudio del microcontrolador y los recursos necesarios para implementar el filtro de retraso temporal, valorando alternativas.
- Implementación del filtro en el microcontrolador mediante el programa PSoC Creator.
- Estudio de los elementos necesarios para implementarlo en una maqueta de puente grúa (overhead crane).
- Realizar un diseño CAD del sistema a estudiar.
- Comunicación serie.
- Simulación en Simscape.
- Valorar los resultados obtenidos.

3 HISTORIA

3.1 Input shaping

Como se ha comentado en la introducción, desde la existencia de sistemas oscilatorios se han intentado crear sistemas de control para su amortiguamiento. Los estudios más pioneros se remontan hasta 1931 donde en una patente publicada por Norbert Wiener y Yuk-Wing Lee [1] se describía un nuevo método para corregir la amplitud y la fase de la señal combinándola con otra señal. Aunque este método no corresponde con el filtro de retraso temporal que se conoce hoy en día.

Más tarde en 1952 D. J. Gimpel y J. F. Calvert publican en la revista “Transactions of the American Institute of Electrical Engineers” [2] un estudio donde se trataban de desarrollar nuevas técnicas para el control de los componentes de la señal de entrada. Este estudio se denomina “Signal component Control”.

En 1957, Otto J. M. Smith publica su nuevo método para reducir la respuesta oscilatoria de un sistema subamortiguado, llamado Posicast [3], este método consiste en excitar un sistema con oscilaciones transitorias, en espacios muy cortos de tiempo, y con magnitudes y fases muy ajustadas para que la resultante de la suma de las fases de oscilación transitoria sea igual a cero. El principio del diseño es ajustar un sistema para la máxima frecuencia de resonancia posible, conservando su estabilidad, independientemente del factor de amortiguamiento, y una vez conseguido esto se aplica el control de Posicast para que elimine completamente la componente oscilatoria de la salida.

Precisamente este es el mayor problema del método de Smith, pues es muy sensible al ajuste de la frecuencia de oscilación natural del sistema; si se desviaba de su frecuencia de vibración un poco, el método era inservible.

Finalmente, en 1990, Neil C. Singer, Warren P. Seering y Kenneth A. Pasch patentan el estudio “Shaping command inputs to minimize unwanted dynamics” [4], al que se denominó finalmente input shaping que consiste en determinar una secuencia de impulsos que elimina la dinámica no deseada del sistema oscilatorio. Estos impulsos se combinan con una secuencia de entrada de comando arbitraria para llevar al sistema dinámico a una salida sin oscilaciones indeseadas. Este sistema resulta ser bastante robusto ante los parámetros del sistema, como su frecuencia natural de vibración. Se puede aplicar tanto a sistemas en bucle abierto como en bucle cerrado.

Tras esto, surgieron nuevos estudios como el de John L. Junkins y Youdan Kim donde aplicaron el input shaping a estructuras flexibles en su libro “Introduction to Dynamics and Control of flexible structures” [5] aplicado sobre todo a naves aeroespaciales y aeronáuticas.

4 CONTEXTUALIZACIÓN

4.1 Descripción del sistema

Este sistema consta de un carro que se desplaza entre unos railes con una fuerza de rozamiento despreciable, unido al carro se dispone de un cable de longitud dada y en su extremo una carga de masa indeterminada. No se necesita conocer la masa de la carga como se comprobará más adelante.

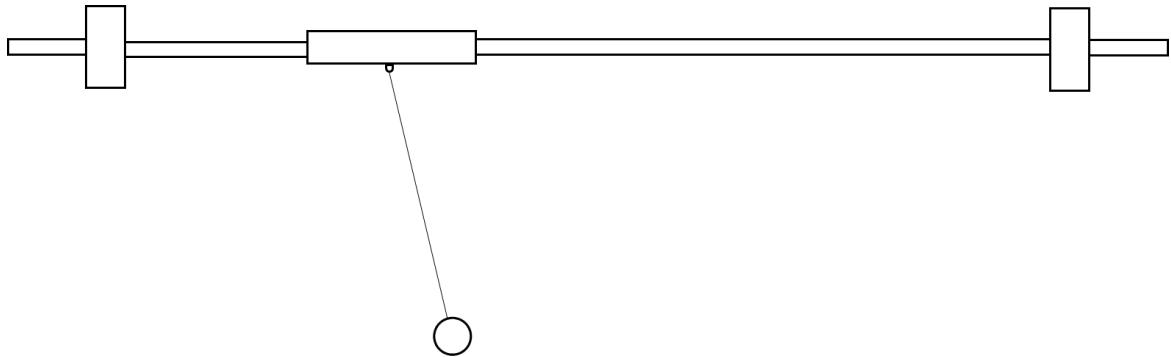


Figura 3: Esquema del sistema a estudiar, carro con masa oscilante. Elaboración propia.

4.1.1 Valores de las variables

	Unidades	Valor
Longitud del cable.	m	0,24
Longitud de la distancia que puede recorrer el carro.	m	0,92
Radio de la esfera	m	0,017
Masa del conjunto carro, cable, carga.	kg	< 1
Gravedad	$\frac{m}{s^2}$	9,81
Coeficientes de rozamiento (μ)	-	0
Coeficiente de amortiguamiento (ζ)	-	0

Tabla 1: Valores de las variables del sistema a estudiar.

5 MODELO DINÁMICO

5.1 Fuerzas y modelado

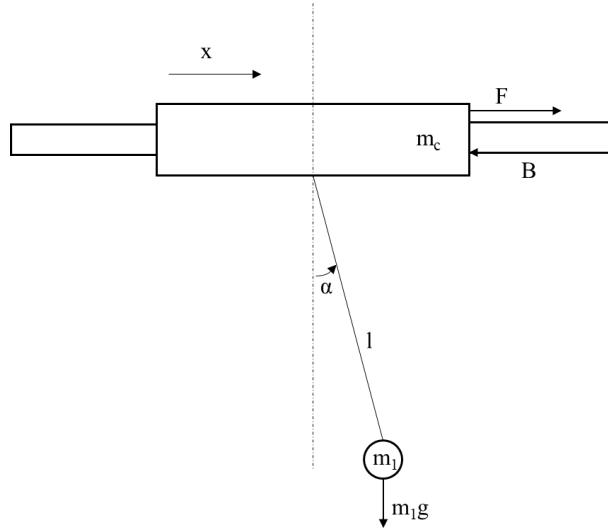


Figura 4: Esquema del sistema a estudiar con las distintas variables y fuerzas dibujadas.
Elaboración propia.

5.1.1 Consideraciones iniciales

Antes de la resolución del problema se definen unas premisas para poder obtener el modelo cinemático de este sistema:

- Se considera la carga como una masa puntual.
- Se supone que el cable es un elemento rígido y de masa despreciable.
- La carga se mueve estrictamente en el plano vertical.
- Se supone nulo el momento de inercia de la carga.

En un problema normal no se debería de despreciar el momento de inercia de la carga ya que puede llevar a error, pues supone un gran cambio en el comportamiento de la carga; pero debido a que el objetivo es que la carga no oscile y su vibración sea la mínima posible, se puede despreciar.

Este sistema se considerará un sistema de dos grados de libertad, el primero es la posición del carro (x) que es controlada por el motor y el segundo grado de libertad es el ángulo (α) que coincide con el ángulo que tiene el cable con respecto al eje vertical.

5.1.2 Variables

Las variables necesarias para la resolución de este problema son las siguientes:

- | | |
|---------------------------------|----------------|
| – Masa carga | m_l |
| – Masa del carro | m_c |
| – Posición de la grúa | x |
| – Velocidad de la grúa | \dot{x} |
| – Ángulo de la carga | α |
| – Velocidad angular de la carga | $\dot{\alpha}$ |
| – Fuerza motor | F |

– Longitud del cable	1
– Gravedad	g
– Rozamiento	B
– Frecuencia de oscilación	f

La resolución de este problema se llevará a cabo por dos métodos diferentes, el método Euler-Lagrange y el método Newton-Euler.

5.1.3 Método Euler-Lagrange

La resolución de este problema mediante el método Euler-Lagrange.

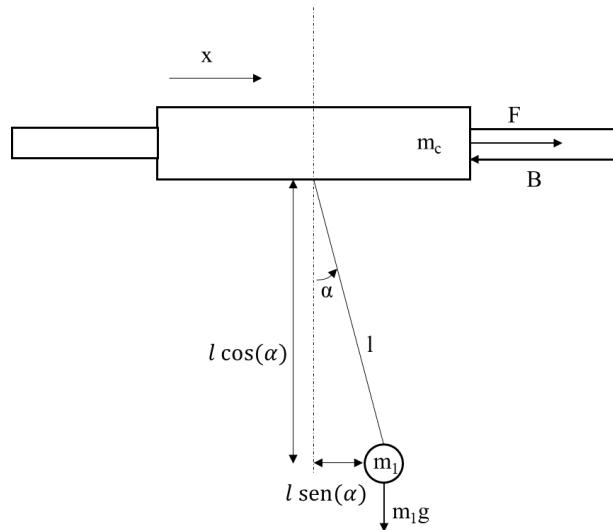


Figura 5: Esquema del sistema a estudiar acotado y con las variables dibujadas. Elaboración propia.

El lagrangiano (L) es igual a la diferencia entre la energía cinética (E_c) y la energía potencial (E_p) en un punto determinado. (1)

$$L = \sum E_c - \sum E_p \quad (1)$$

La energía cinética de este sistema se define como la suma (4) de la energía cinética del carro (2) y la energía cinética de la carga (3):

$$E_{carro} = \frac{1}{2} m_c \dot{x}^2 \quad (2)$$

$$E_{carga} = \frac{1}{2} m_1 v^2 \quad (3)$$

$$E = \frac{1}{2} m_1 v^2 + \frac{1}{2} m_c \dot{x}^2 \quad (4)$$

Para poder obtener la energía cinética de la carga se necesita antes calcular su velocidad para ello, se necesita la longitud del cable y la posición angular de la carga con respecto a la vertical. Las siguientes ecuaciones muestran su posición en el eje x (5) y eje y (6).

$$P_x = x + l \sin(\alpha) \quad (5)$$

$$P_y = l \cos(\alpha) \quad (6)$$

Para obtener las velocidades se deriva las posiciones en el eje x (7) y en el eje y (8):

$$V_x = \frac{d}{dt} (x + l \sin(\alpha)) = \dot{x} + l \cos \alpha \dot{\alpha} \quad (7)$$

$$V_y = \frac{d}{dt} (l \cos(\alpha)) = -l \sin \alpha \dot{\alpha} \quad (8)$$

Con estas velocidades ya se puede extraer el vector de velocidad de la carga (V_c) (9):

$$V_c = \begin{pmatrix} \dot{x} + l \cos \alpha \dot{\alpha} \\ -l \sin \alpha \dot{\alpha} \end{pmatrix} \quad (9)$$

Se calcula la norma del vector de velocidades (V_c) (10) y operando se llega a un resultado (11):

$$V_c^2 = (\dot{x} + l \cos \alpha \dot{\alpha})^2 + (-l \sin \alpha \dot{\alpha})^2 \quad (10)$$

$$V_c^2 = \dot{x}^2 + 2l\dot{x} \cos \alpha \dot{\alpha} + l^2 \dot{\alpha}^2 \quad (11)$$

Sustituyendo en la ecuación de la energía cinética (12) y operando se llega al resultado de la energía cinética de la carga (13):

$$E_{carga} = \frac{1}{2} m_1 (\dot{x}^2 + 2l\dot{x} \cos \alpha \dot{\alpha} + l^2 \dot{\alpha}^2) \quad (12)$$

$$E = \frac{1}{2} m_1 (\dot{x}^2 + 2l\dot{x} \cos \alpha \dot{\alpha} + l^2 \dot{\alpha}^2) + \frac{1}{2} m_c \dot{x}^2 \quad (13)$$

Para poder calcular el lagrangiano de este sistema se necesita obtener también la energía potencial que será el valor resultante de la suma de la energía potencial del carro (E_{pc}) y la energía potencial de la carga (E_{p1}), dado que el carro se mueve solo en un plano ortogonal al de la fuerza de la gravedad y no se ve afectado por ella se sabe que su energía potencial es igual a cero ($E_{pc}=0$), por otro lado la energía potencial de la carga coincidiría con $E_{p1} = mg l \cos(\alpha)$ siendo g la gravedad.

Como se ha visto antes (1), el lagrangiano coincide con la diferencia del sumatorio de las energías cinéticas y el sumatorio de las energías potenciales, si se sustituye y opera el resultado obtenido es el siguiente (14):

$$L = \left[\frac{1}{2} m_1 (\dot{x}^2 + 2l\dot{x} \cos \alpha \dot{\alpha} + l^2 \dot{\alpha}^2) + \frac{1}{2} m_c \dot{x}^2 \right] + m_1 g l (1 - \cos(\alpha)) \quad (14)$$

Una vez obtenido el lagrangiano se tiene que aplicar las ecuaciones de Euler-Lagrange (15) y (16):

$$\frac{d}{dt} \left[\frac{\partial L}{\partial \dot{x}} \right] - \frac{\partial L}{\partial x} = F \quad (15)$$

$$\frac{d}{dt} \left[\frac{\partial L}{\partial \dot{\alpha}} \right] - \frac{\partial L}{\partial \alpha} = 0 \quad (16)$$

Donde F es la fuerza que aplica el motor para mover el carro (15). Ahora se resuelve cada uno de los términos para después sustituirlos en la ecuación. Se empieza por la derivada parcial de L con respecto a \dot{x} (17). Operando se obtiene el siguiente resultado (18).

$$\frac{\partial L}{\partial \dot{x}} = \frac{1}{2} m_c \frac{\partial}{\partial \dot{x}} (\dot{x}^2) + m_1 l (\cos \alpha \dot{\alpha}) \frac{\partial}{\partial \dot{x}} \dot{x} + \frac{1}{2} m_1 \frac{\partial}{\partial \dot{x}} \dot{x}^2 \quad (17)$$

$$= m_c \dot{x} + m_1 \dot{x} + m l \cos \alpha \dot{\alpha} \quad (18)$$

Una vez se llega a la expresión, se deriva con respecto al tiempo (19) para poder sustituirla en la ecuación.

$$\frac{d}{dt} \left[\frac{\partial L}{\partial \dot{x}} \right] = (m_c + m_1) \frac{d}{dt} (\dot{x}) + m l \frac{d}{dt} (\cos \alpha \dot{\alpha}) \quad (19)$$

$$= (m_c + m_1) \ddot{x} + m l (\cos \alpha \ddot{\alpha} - \operatorname{sen} \alpha \dot{\alpha}^2) \quad (20)$$

Ahora se deriva L con respecto a $\dot{\alpha}$ (21). El resultado será (22):

$$\frac{\partial L}{\partial \dot{\alpha}} = \frac{1}{2} m_1 \left[2l\dot{x} \cos \alpha \frac{d\dot{\alpha}}{d\dot{\alpha}} + l^2 \frac{d\dot{\alpha}^2}{d\dot{\alpha}} \right] \quad (21)$$

$$= m_1 l \dot{x} \cos \alpha + m_1 l^2 \dot{\alpha} \quad (22)$$

Se deriva la expresión que se obtiene, con respecto al tiempo (23). Y Operando se llega al siguiente resultado (24).

$$\frac{d}{dt} \left[\frac{\partial L}{\partial \dot{\theta}} \right] = m_1 l \frac{d(\dot{x} \cos \alpha)}{dt} + m_1 l^2 \frac{d\dot{\alpha}}{dt} \quad (23)$$

$$= m_1 l (\ddot{x} \cos \alpha + \dot{x} \sin \alpha \dot{\alpha}) + m_1 l^2 \ddot{\alpha} \quad (24)$$

Finalmente se necesita la derivada parcial de L con respecto a α (25).

$$\frac{\partial L}{\partial \alpha} = \frac{m_1 l \dot{x} \dot{\alpha} d(\cos \alpha)}{d\dot{\alpha}} - m_1 g l \frac{d(\cos \alpha)}{d\dot{\alpha}} \quad (25)$$

$$= -m_1 l \dot{x} \dot{\alpha} \sin \alpha + m_1 g l \sin \alpha \quad (26)$$

Una vez se obtienen todos los términos necesarios se sustituyen en las ecuaciones de Euler-Lagrange (15) y (16). Sustituyendo (27) y (28).

$$F = (m_g + m_1) \ddot{x} + m_1 l (\cos \alpha \ddot{\alpha} - \sin \alpha \dot{\alpha}^2) \quad (27)$$

$$0 = m_1 l (\ddot{x} \cos \alpha - \dot{x} \sin \alpha \dot{\alpha}) + m_1 l^2 \ddot{\alpha} + m_1 l \sin \alpha \dot{x} \dot{\alpha} + g m_1 l \sin \alpha \quad (28)$$

Si se tuviera en cuenta el posible rozamiento del carro con la vía B se incluiría de la siguiente manera (29):

$$F - B \dot{x} = (m_g + m_1) \ddot{x} + m_1 l (\cos \alpha \ddot{\alpha} - \sin \alpha \dot{\alpha}^2) \quad (29)$$

Una vez se sustituyen todos los términos despejando las variables que nos interesan que son la aceleración del carro y la aceleración angular de la carga, se obtienen las ecuaciones que determinan el comportamiento del sistema (30) y (31).

$$\ddot{x} = \frac{F - B \dot{x} - m_1 l (\cos \alpha \ddot{\alpha} - \sin \alpha \dot{\alpha}^2)}{m_g + m_1} \quad (30)$$

$$\ddot{\alpha} = - \frac{\ddot{x} \cos \alpha - \dot{x} \sin \alpha \dot{\alpha} + \dot{x} \dot{\alpha} \sin \alpha + g \sin \alpha}{l} \quad (31)$$

Dado que se intenta reducir el ángulo de oscilación lo máximo posible, siempre que α sea lo suficientemente pequeño el valor de $\sin(\alpha)$ será muy próximo a α expresado en radianes y el valor del $\cos(\alpha)$ será muy próximo a 1, entonces se puede simplificar el modelado de la siguiente

manera, $\cos(\alpha) \cong 1$ $\sin(\alpha) \cong \alpha$ si $\alpha < 30^\circ$, pudiendo asegurar que el error será menor a un 5%, por tanto el modelado simplificado es igual a (32) y (33):

$$\ddot{x} = \frac{F - B\dot{x} - m_1 l(\ddot{\alpha} + \alpha \dot{\alpha}^2)}{m_g + m_1} \quad (32)$$

$$\ddot{\alpha} = -\frac{\ddot{x} + g\alpha}{l} \quad (33)$$

5.1.4 Método Newton-Euler

La resolución de este problema mediante el método Newton-Euler.

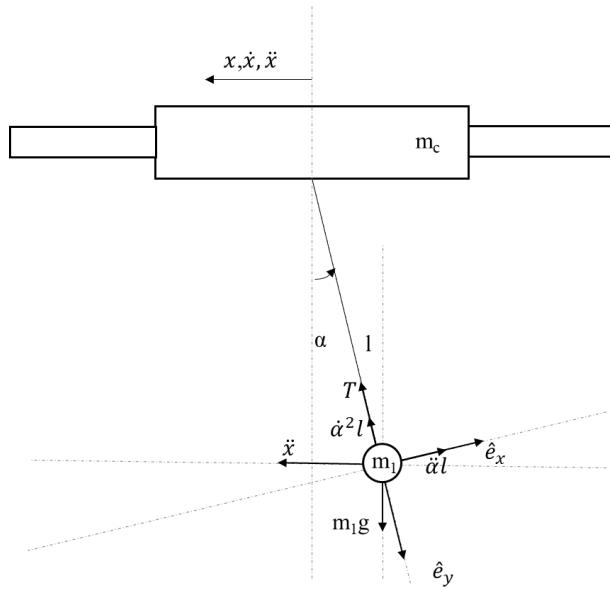


Figura 6: Esquema del sistema a estudiar con las variables necesarias para aplicar el método Newton-Euler. Elaboración propia.

Primero se igualan las fuerzas en el eje indicado en la figura anterior (34) y operando (35) se llega al siguiente resultado (36):

$$\hat{e}_t) \quad -m_1 g \sin \alpha = m_1 (\ddot{\alpha} l - \dot{x} \cos \alpha) \quad (34)$$

$$ml\ddot{\alpha} + mg \sin \alpha = m \cos \alpha U \quad (35)$$

$$\ddot{\alpha} + mg \sin \alpha = m \cos \alpha U \quad (36)$$

Como se aplicó anteriormente en el método Euler-Lagrange se pueden hacer las siguientes suposiciones sabiendo que no se va a cometer un error mayor al 5%, si $\alpha < 30^\circ$ (37):

$$\cos(\alpha) \cong 1, \quad \sin(\alpha) \cong \alpha \rightarrow \varepsilon < 5\% \quad (37)$$

Entonces sustituyendo en la ecuación anterior (36) y sabiendo que (38) se obtiene la ecuación (40):

$$\omega_n^2 = \frac{g}{l} \quad (38)$$

$$\ddot{\alpha} + \frac{g}{l} \alpha = \frac{\omega_n^2}{g} U \quad ODE \text{ (Ordinary Differential Equation)} \quad (39)$$

$$\ddot{\alpha} + \omega_n^2 \alpha = \frac{\omega_n^2}{g} U \quad (40)$$

Para obtener la función de transferencia, se opera de la siguiente manera (41), obteniendo (42):

$$(s^2 + \omega_n^2) \alpha(s) = \frac{\omega_n^2}{g} U(s) \quad (41)$$

$$\frac{\alpha(s)}{U(s)} = \frac{\frac{\omega_n^2}{g}}{(s^2 + \omega_n^2)} \quad (42)$$

5.2 Frecuencia natural de vibración

Para calcular correctamente un filtro de retraso temporal (input shaper) es esencial conocer la frecuencia natural de oscilación del sistema a estudiar. El período de oscilación del sistema en cuestión se determina de la siguiente manera (43):

$$T = 2\pi \sqrt{\frac{l}{g}} \quad (43)$$

Se conocen todos los valores para sustituir, l coincide con la suma de la longitud del cable y el radio de la esfera:

$$l = r_{esfera} + l_{cable} = 0,017 + 0,24 = 0,257 \text{ m} \quad (44)$$

Sustituyendo por los valores del sistema en (43):

$$T = 2\pi \sqrt{\frac{0,257}{9,81}} = 1,017 \text{ s} \quad (45)$$

La frecuencia coincide con la inversa del período (46):

$$f = \frac{1}{T} = \frac{1}{1,017} = 0,983 \text{ Hz} \quad (46)$$

Se calcula la frecuencia angular que se necesitará utilizar más adelante:

$$\omega = 2 \pi f = 6,176 \text{ Hz} \quad (47)$$

5.3 Modelado dinámico en Simulink

Para posteriormente poder verificar el correcto funcionamiento de la simulación que se realizará en Simulink con Simscape, y verificar que el modelo realizado en el apartado 5.1.4 es correcto, se realizará una simulación del modelo dinámico calculado.

Para ello solo se tiene que escribir la función de transferencia en un bloque que representará la planta, su entrada será un step y en la salida se grafica el ángulo en un “scope”.

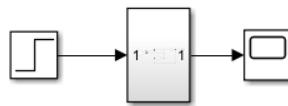


Figura 7: Diagrama de bloques para la simulación del modelo matemático en Simulink.

En el interior del bloque de la planta se escribe la función del modelado (40). El resultado es el siguiente:

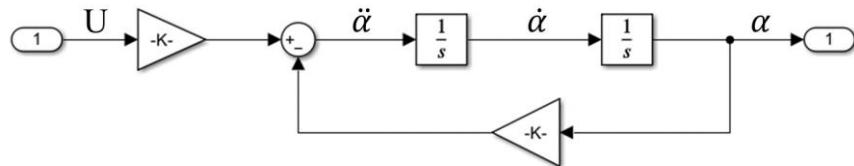


Figura 8: Diagrama de bloques para la simulación del modelo matemático en Simulink.

Por la salida se puede visualizar el ángulo de oscilación de la carga y verificar que los datos coinciden con los anteriormente calculados, en la siguiente figura se puede comprobar que el período es igual a 1,017s, coincidiendo con el valor de la ecuación (45).

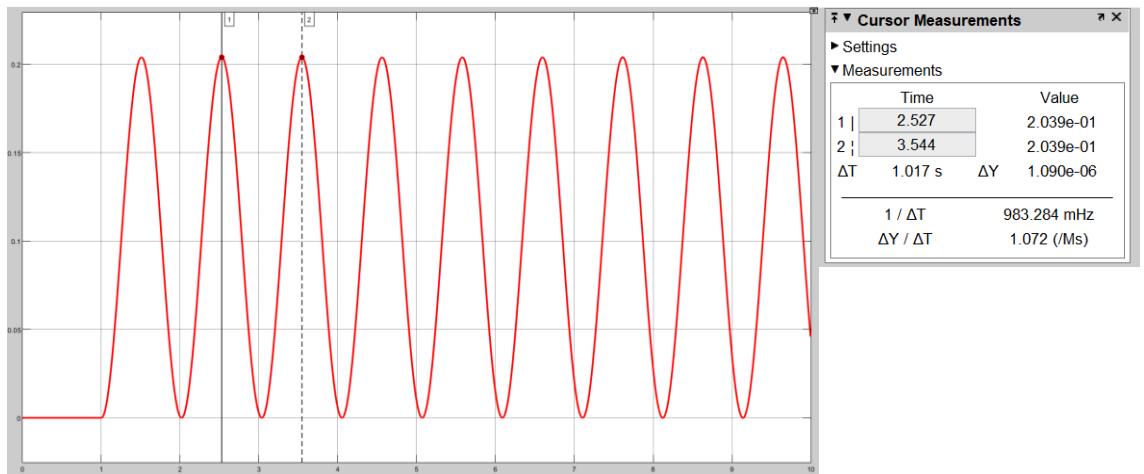


Figura 9: Oscilación de la carga (α) en el modelado dinámico del sistema en Simulink.

6 SHAPERS

6.1 Que es el input shaping

El input shaping o modelado de entrada es una técnica de control en bucle abierto que permite reducir las vibraciones de sistemas automatizados o controlados por ordenador, es implementado mediante la convolución de una secuencia de impulsos con un comando de entrada deseado. Los tiempos en los que se aplican los impulsos y sus magnitudes vienen dadas por la resolución de una serie de ecuaciones de restricción. El objetivo de este trabajo no es el cálculo de los filtros de retraso temporal, solo se hará una introducción para contextualizar.

Dado que el input shaper se localiza fuera del bucle (Figura 10) de realimentación, es un método que no afecta al sistema de control, lo que lo hace fácilmente implementable.

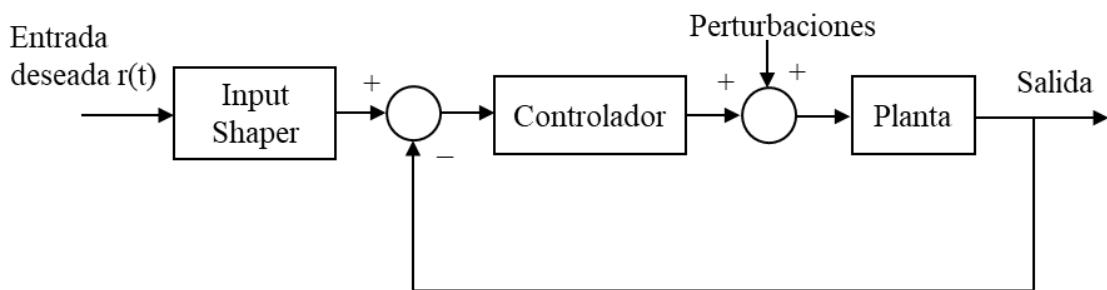


Figura 10: Diagrama de bloques habitual donde se implementa un prefiltro input shaper.
Elaboración propia a partir del documento [10].

6.2 Tipos de shapers

6.2.1 Introducción

Como se ha visto en el apartado de historia, el primer input shaper en surgir fue el de vibración cero (ZV) se propuso por primera vez en 1950 y se desarrolló más tarde, después de este, para proporcionar robustez frente a la variación de los parámetros del sistema, se introdujeron el shaper vibración cero derivado (ZVD) y el shaper de insensibilidad especificada (SI).

Con todo este desarrollo también se llegó a otros enfoques como por ejemplo los vectores de impulso que se verán más adelante u otros nuevos como impulsos negativos para reducir el tiempo transitorio. También se desarrollaron métodos que involucran filtros discretos. La técnica de ubicación cero se extendió al dominio discreto al construir las secuencias de impulsos en el plano Z.

6.2.2 Input Shaper ZV

Una señal de entrada se convoluciona con dos impulsos, debidamente ubicados en el tiempo y con la amplitud adecuada. Es decir, la señal final se consigue convolucionando la señal de entrada con el input shaper (IS) (48) donde $\delta(t)$ es la función delta de Dirac, I_1 e I_2 son las amplitudes y ΔT es el tiempo entre el primer impulso y el segundo. El primero en desarrollar este filtro ZV fue la técnica de Posicast en 1957 por Otto J. M. Smith.

$$ZV(t) = I_1\delta(t) + I_2\delta(t - \Delta T) \quad (48)$$

En la siguiente figura (Figura 11) se puede observar a la izquierda la señal de entrada sin aplicar el filtro de retraso temporal, una vez que se convoluciona con el filtro la señal adquiere un retraso ΔT que coincide con el valor de tiempo en el que se aplica el último impulso.

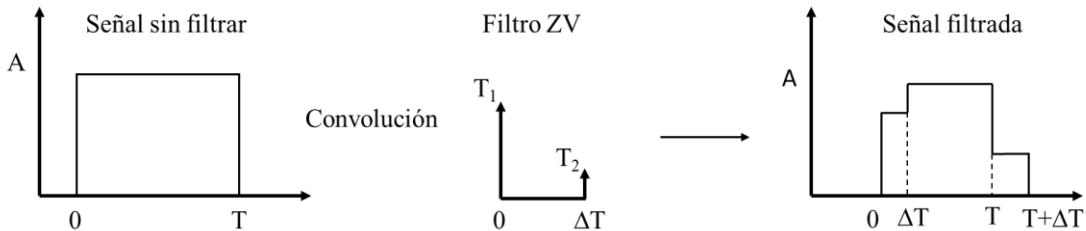


Figura 11: Filtrado de una señal con un filtro ZV. Elaboración propia a partir del documento [10].

Los parámetros del shaper son dependientes de la frecuencia natural del sistema ω y el coeficiente de amortiguamiento ζ . El instante en el que se aplica el impulso en el filtro se calcula de la siguiente manera:

$$\Delta T = \frac{\pi}{\omega \sqrt{1 - \zeta^2}} \quad (49)$$

Las magnitudes de los impulsos vienen dadas por las siguientes ecuaciones (50) y (52):

$$I_1 = \frac{\bar{I}}{1 + \bar{I}} \quad (50)$$

Donde \bar{I} es igual a:

$$\bar{I} = e^{-\frac{\zeta\pi}{\sqrt{1-\zeta^2}}} \quad (51)$$

$$I_2 = 1 - I_1 \quad (52)$$

O siendo equivalente, a:

$$ZV = \begin{bmatrix} A_i \\ t_j \end{bmatrix} = \begin{bmatrix} \frac{1}{1+K} & \frac{K}{1+K} \\ 0 & \frac{1}{T} \end{bmatrix} \quad (53)$$

Donde T es igual a (54) y K es igual a (55):

$$T = \frac{\pi}{\omega \sqrt{1 - \zeta^2}} \quad (54)$$

$$K = e^{-\frac{\zeta\pi}{\sqrt{1-\zeta^2}}} \quad (55)$$

Este shaper tiene el mínimo tiempo posible para ejecutarse solo con impulsos positivos. Este tiempo es importante porque en la convolución del input shaper con la señal de entrada se produce un retraso y aumenta la duración del proceso. Si el shaper ZV está bien diseñado minimizará la máxima energía residual de la vibración.

6.2.3 Input Shaper ZVD

Posteriormente al shaper ZV se trató de diseñar un shaper que funcionase bien en sistemas reales, aumentando la robustez. Siendo efectivo igualmente ante errores en el modelado del sistema. La primera forma de shaper robusto fue desarrollada estableciendo la derivada con respecto a la frecuencia de la vibración residual igual a cero (Singer, 1989). El filtro resultante fue el “zero vibration and derivative” (ZVD). El precio para este aumento de robustez en el filtro es el aumento de tiempo que le lleva realizar las operaciones al shaper, por tanto, aumenta el retraso del sistema. Se puede expresar como:

$$ZVD = \begin{bmatrix} A_i \\ t_j \end{bmatrix} = \begin{bmatrix} 1 & \frac{2K}{1+2K+K^2} & \frac{K^2}{1+2K+K^2} \\ 0 & T & 2T \end{bmatrix} \quad (56)$$

Donde T y K se definen de la misma manera que en el Shaper ZV (54) y (55).

6.2.4 Input Shaper EI

Otro tipo de input shaper es el Extra-Insensitive (EI), el tiempo de ejecución de las operaciones de este shaper es muy similar al del shaper ZVD, pero su insensibilidad a cambios de parámetros es mucho mayor que el del ZVD. La sensibilidad del shaper EI depende de la magnitud de oscilación permitida en el sistema, si se aumenta la vibración residual permitida la insensibilidad del filtro aumenta. Se describe como:

$$EI = \begin{bmatrix} A_i \\ t_j \end{bmatrix} = \begin{bmatrix} 1+V & \frac{1-V}{2} & \frac{1+V}{4} \\ 0 & T & 2T \end{bmatrix} \quad (57)$$

Donde T es igual a (54) y V representa el nivel de insensibilidad a las oscilaciones del sistema.

6.2.5 Comparación de la robustez de los shapers

Para poder comparar lo robustos que son cada uno de los shapers frente a la variación de la frecuencia real con respecto a la frecuencia natural del sistema (es decir la precisión del modelo dinámico) se pueden comparar gráficamente.

La energía residual viene dada por esta ecuación, dependientes de la frecuencia y el coeficiente de amortiguación del sistema:

$$V(\omega_i, \zeta_i) = e^{-\zeta_i \omega_i t_n} \sqrt{C^2 + D^2} \quad (58)$$

Donde C y D son (59) y (60) respectivamente:

$$C = \sum_{j=1}^n A_j e^{\zeta_i \omega_i t_j} \cos \left(\omega_i \sqrt{1 - \zeta_i^2} t_j \right) \quad (59)$$

$$D = \sum_{j=1}^n A_j e^{\zeta_i \omega_i t_j} \sin\left(\omega_i \sqrt{1 - \zeta_i^2} t_j\right) \quad (60)$$

Donde A_j es la magnitud del impulso del input shaper y t_j el instante en el que se aplica ese impulso. Por tanto, para cada input shaper se grafica la ecuación de la energía residual con respecto a la frecuencia y se obtiene una gráfica donde, el punto más próximo a cero de la energía residual sea donde la frecuencia coincida con la natural del sistema. El input shaper será más robusto cuanto más cercana a cero sea la energía residual alejándose de la frecuencia natural del sistema.

6.2.6 Comparación de los shapers en Matlab

Para poder visualizar gráficamente la robustez de los distintos tipos de filtro de retraso temporal, se graficarán las funciones anteriormente descritas (vibración residual) (58) en función de la frecuencia del sistema. La vibración residual debe ser mínima cuando el filtro que se usa para resolver las ecuaciones corresponde con el filtro calculado para esa frecuencia, en este caso se usará el filtro que posteriormente se utilizará para este caso. Por lo que la vibración residual mínima coincidirá en la frecuencia natural del sistema anteriormente calculada, $\omega=6.178 \frac{rad}{s}$.

Cuando se grafica en MATLAB la función que describe la energía residual, se obtiene la siguiente gráfica:

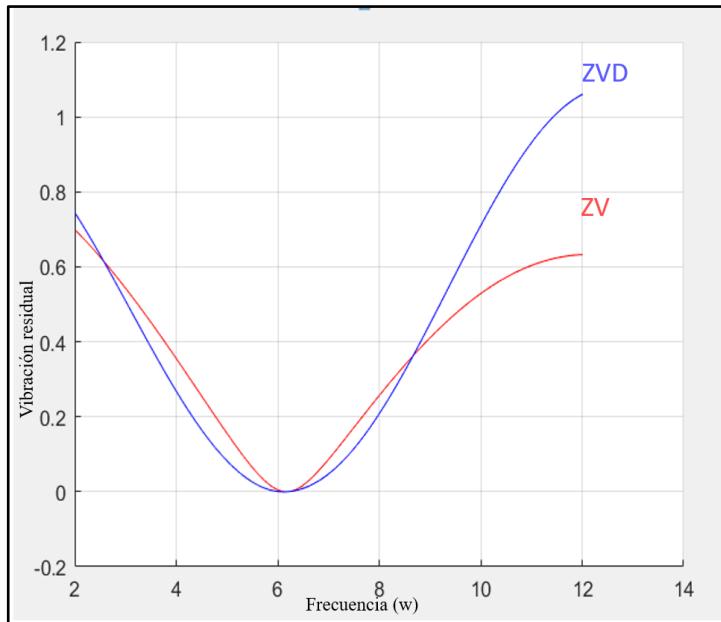


Figura 12: Comparación de la robustez del filtro ZV y filtro ZVD. Elaboración propia en Matlab.

En el eje de ordenadas se representan las vibraciones residuales del sistema, la vibración residual es la amplitud de la vibración cuando se usa el shaper entre la amplitud cuando no se usa el shaper. En el eje de abscisas se representa la frecuencia angular que es igual a $\omega=2\pi f$.

Se puede saber por esta figura que cuando el error del modelado del sistema aumenta, la vibración residual también aumenta rápidamente cuando se usa un filtro ZV, por el contrario, cuando se usa un filtro ZVD la vibración se mantiene en un nivel inferior para un error del modelado mayor.

Para demostrar, que si la frecuencia calculada se aleja de la frecuencia real del sistema, el filtro ZVD tiene un mejor comportamiento, se realiza la siguiente prueba. Por ejemplo, se obtiene un 7% de error en el cálculo de la frecuencia, la energía residual con el filtro ZV aumentaría un 2,93% por el contrario con el filtro ZVD solo aumentaría un 1,4%.

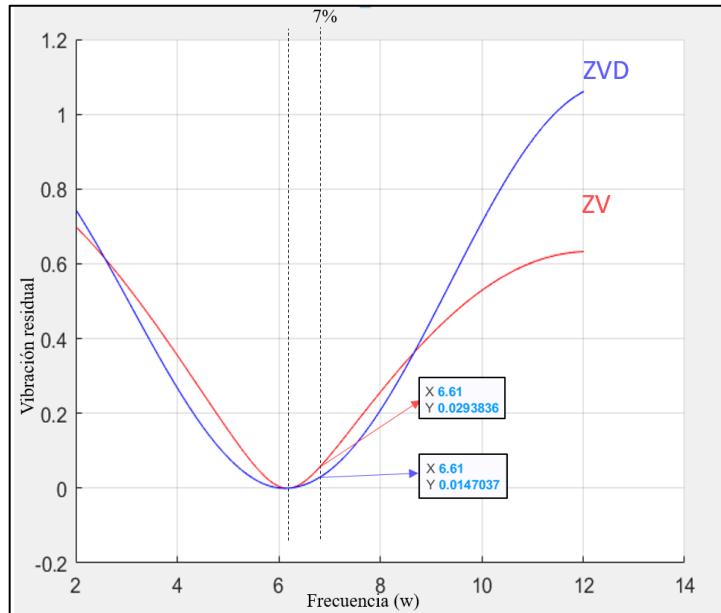


Figura 13: Robustez de los filtros ZV y ZVD cuando la frecuencia calculada se aleja un 7% de la frecuencia real. Elaboración propia.

Se puede observar que la vibración residual es mínima cuando alcanzamos el punto de la frecuencia angular natural del sistema que coincide tanto en la gráfica como la calculada en la ecuación (47), con 6,61.

6.3 Cálculo del shaper para este caso.

El diseño de los filtros de retraso temporal se realiza resolviendo un problema de restricciones, el fin de este trabajo no es el diseño de un filtro de retraso temporal sino su implementación por lo que solo se calcularán sus valores.

Para el caso estudiado se calcula un filtro de retraso temporal ZVD, su alta robustez permite que el cálculo de la frecuencia de oscilación tenga más margen de error que si se aplicase un filtro ZV.

Como se ha explicado antes, para calcular el filtro ZVD se tiene que calcular K (61) y T (62):

$$K = e^{-\frac{\zeta\pi}{\sqrt{1-\zeta^2}}} \quad (61)$$

La única variable en esta ecuación, es el coeficiente de amortiguamiento que como se ha mencionado en el apartado 4.1.1 será cero. Por lo que:

$$K = 1 \quad (62)$$

Para calcular T, se sustituye el coeficiente de amortiguamiento y la frecuencia angular por el valor calculado anteriormente en (47):

$$T = \frac{\pi}{\omega\sqrt{1-\zeta^2}} \quad (63)$$

Se obtiene:

$$T = \frac{\pi}{\omega\sqrt{1-\zeta^2}} = \frac{\pi}{6.176} = 0,508 \text{ s} \quad (64)$$

Sustituyendo en esta ecuación:

$$ZVD = \begin{bmatrix} A_i \\ t_j \end{bmatrix} = \begin{bmatrix} \frac{1}{1+2K+K^2} & \frac{2K}{1+2K+K^2} & \frac{K^2}{1+2K+K^2} \\ 0 & T & \frac{2T}{1+2K+K^2} \end{bmatrix} \quad (65)$$

El filtro d retraso temporal queda de la siguiente forma:

$$ZVD = \begin{bmatrix} A_i \\ t_j \end{bmatrix} = \begin{bmatrix} 0,25 & 0,5 & 0,25 \\ 0 & 0,508 & 1,017 \end{bmatrix} \quad (66)$$

6.4 Vectores de impulso

Los vectores de impulso son una herramienta matemática muy útil para el diseño y el análisis de los input shapers y se pueden utilizar tanto para sistemas subamortiguados como para sobreamortiguados. Este concepto fue introducido por el profesor William E. Singhouse para facilitar el desarrollo de nuevos filtros de retraso temporal. De esta manera se puede entender el funcionamiento de los input shapers desde una perspectiva gráfica permitiéndonos desde crear nuevos shapers o incluso mejorar los ya diseñados.

6.4.1 Definición de los vectores de impulso

Un vector de impulso I_i se define inicialmente usando la magnitud y el ángulo del vector en un sistema polar de coordenadas.

Para un sistema de segundo orden subamortiguado de frecuencia ω_n y factor de amortiguamiento ζ , la magnitud I_i y el ángulo θ_i del vector de impulso I_i se define en un sistema de coordenadas polares de dos dimensiones como:

$$I_i = A_i e^{\zeta \omega_n t_i} \quad (67)$$

$$\theta_i = \omega_d t_i \quad (68)$$

Donde A_i corresponde a la magnitud del impulso, t_i a la localización temporal del impulso y ω_d a la frecuencia amortiguada del sistema, es decir:

$$\omega_d = \omega_n \sqrt{1 - \zeta^2} \quad (69)$$

Para un impulso positivo ($A_i > 0$), el punto inicial del vector está localizado en el origen del sistema de coordenadas polar, si el impulso es negativo ($A_i < 0$), el punto de terminación del vector se localiza en el origen de coordenadas polares.

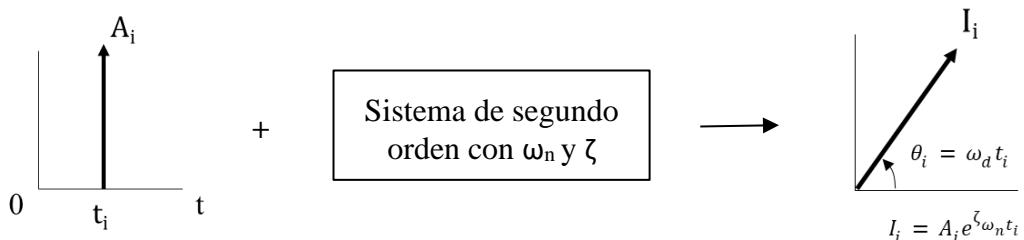


Figura 14: Vector de impulso positivo en un sistema de segundo orden.

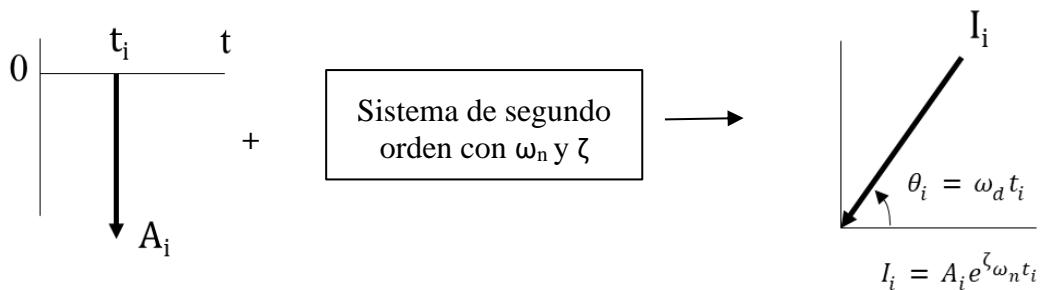


Figura 15: Vector de impulso negativo en un sistema de segundo orden. Elaboración propia a partir del documento [10].

del vector está localizado en el origen de coordenadas. En la parte inferior el vector de impulso es negativo ($A_i < 0$), porque su punto terminal está localizado en el origen de coordenadas.

La diferencia esencial entre un impulso y un vector de impulso, es principalmente que el impulso ($A_i \delta(t-t_i)$) define estrictamente una cantidad matemática mientras que el vector de impulso I_i también incluye parámetros físicos como ω_d y ζ de un sistema de segundo orden.

Como se puede observar en la ecuación de I_i (67), la amplitud en $t = t_i$ decremente bajo un factor de $e^{\zeta\omega_n t_i}$ en comparación con la amplitud en $t = 0$ este efecto también se puede visualizar gráficamente como la espiral de amortiguamiento. Como se puede ver en la Figura 16.

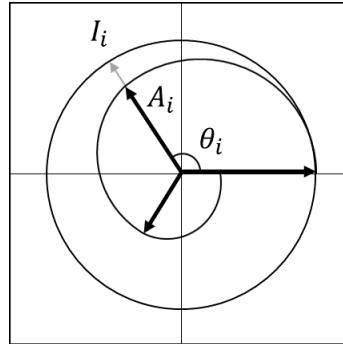


Figura 16: Espiral de amortiguamiento. Elaboración propia a partir del documento [10].

En esta figura la circunferencia externa representa los vectores de impulso con magnitudes constantes (sin amortiguamiento) y la espiral representa la disminución en la magnitud de los impulsos A_i que decrecen debido al amortiguamiento.

$$A_i = \frac{I_i}{e^{\zeta\omega_n t_i}} \quad (70)$$

6.4.2 Propiedades de los vectores de impulso

Las ventajas de utilizar estos vectores de impulso para la creación de nuevos input shapers se debe fundamentalmente a dos propiedades.

- Propiedad 1: En un sistema de segundo orden independientemente de que sea subamortiguado o sobreamortiguado si le se aplica dos vectores de impulso su respuesta será la misma que si le aplicase la resultante de esos dos vectores de impulso, una vez transcurrido el tiempo de los dos impulsos.
- Propiedad 2: Si la resultante de dos vectores de impulso es cero, entonces después del tiempo de que estos vectores de entrada actúen en el sistema de segundo orden la respuesta será igual a cero. Esta propiedad es la más relevante dado que nuestro objetivo es obtener una respuesta que sea igual a cero.

Para comprobar estas propiedades y la utilidad de los vectores de impulso se aplicarán al input shaper que se ha calculado en el apartado 6.3.

6.4.3 Vectores de impulso aplicados a este sistema

Primero se calculan los vectores de impulso uno a uno y después se comprueban sus propiedades:

$$ZVD = \begin{bmatrix} A_i \\ t_j \end{bmatrix} = \begin{bmatrix} 0,25 & 0,5 & 0,25 \\ 0 & 0,508 & 1,017 \end{bmatrix} \quad (71)$$

El cálculo de I_i y θ_i se realiza con las ecuaciones anteriormente estudiadas (67) y (68):

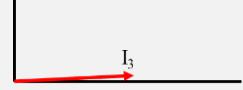
Vector de impulso	I_i	$\theta_i(rad)$	Resultado
$I_1 = \begin{bmatrix} 0,25 \\ 0 \end{bmatrix}$	0,25	0	
$I_2 = \begin{bmatrix} 0,5 \\ 0,508 \end{bmatrix}$	0,5	3,137	
$I_3 = \begin{bmatrix} 0,25 \\ 1,017 \end{bmatrix}$	0,25	6,281	

Tabla 2: Cálculo de los vectores de impulso del filtro que se aplicará.

Representándolos juntos:

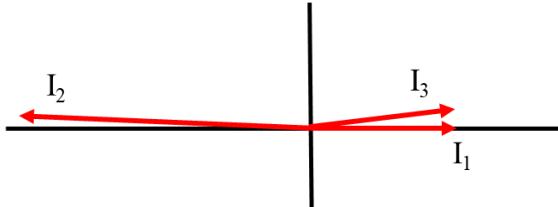


Figura 20: Vectores de impulso del filtro ZVD. Elaboración propia.

Se realiza la suma de estos vectores de impulso sumándolos como si fueran vectores normales:

$$R_x = I_1 \cos \theta_1 + I_2 \cos \theta_2 + I_3 \cos \theta_3 \quad (72)$$

$$R_y = I_1 \sin \theta_1 + I_2 \sin \theta_2 + I_3 \sin \theta_3 \quad (73)$$

Sustituyendo por los valores de la tabla:

$$R_x = 0,25 \cos(0) + 0,5 \cos(3,137) + 0,25 \cos(6,281) = 0 \quad (74)$$

$$R_y = 0,25 \sin(0) + 0,5 \sin(3,137) + 0,25 \sin(6,281) = 0 \quad (75)$$

Con estos resultados validamos el filtro calculado y la segunda propiedad de los vectores de impulso pues como la resultante de estos vectores es cero, transcurrido el tiempo necesario para que actúen los tres la respuesta del sistema será muy próxima a cero.

7 MÉTODO DE DISEÑO

Se utiliza una técnica de metodología de diseño propuesta por Christopher Jones denominado Caja negra, Caja transparente. Esto consiste en realizar una abstracción del sistema con el fin de conocer tanto las interacciones que tiene con el exterior (Figura 21) como el funcionamiento interno del sistema (Figura 22).

7.1 Caja negra

En este tipo de método, el diseño está compuesto de entradas y salidas sin conocer lo que se realiza con ellas. Nuestro diseño se define como:

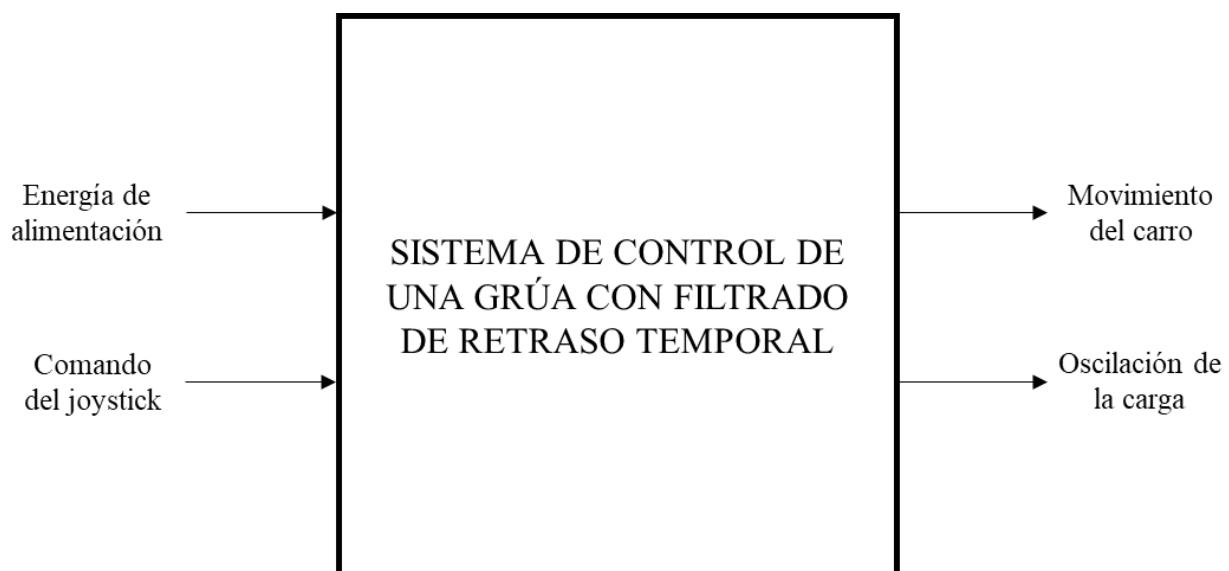


Figura 21: Modelo caja negra

Señal	Tipo	Dimensiones	Descripción
Energía alimentación	Entrada	Tensión	Se necesita energía para suprir las necesidades del sistema electrónico.
Comando del joystick	Entrada	Tensión	Esta señal sale del joystick y es la encargada de decirle al sistema la dirección del movimiento que se quiere que haga la grúa.
Movimiento del carro	Salida	Longitud	El carro actúa moviéndose en la dirección deseada.

Tabla 3: Tipos de señales que entran y salen del modelo caja negra.

7.2 Caja transparente

Una vez se conocen las interacciones que el sistema tiene con el exterior, toca comprender el modo de funcionamiento y como interactúa con esas señales.

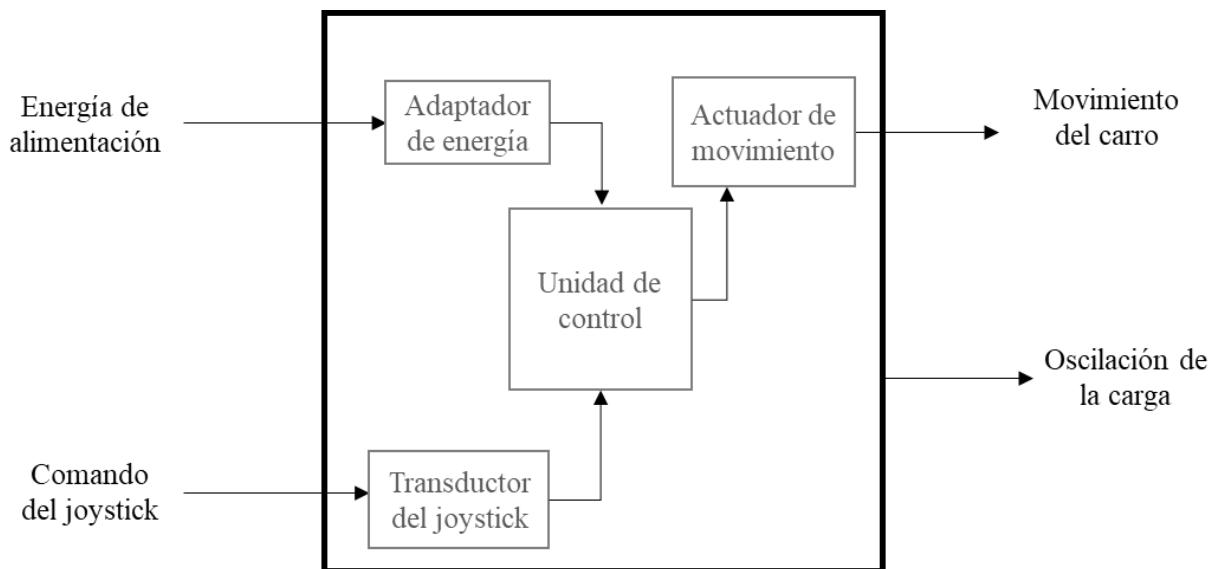


Figura 22: Modelo caja transparente

Este modelo de caja transparente está compuesto de sensores, actuadores y la unidad de control.

- Adaptador de energía: Este elemento se encarga de adaptar la energía que se obtiene del exterior para que pueda ser utilizada por el resto de los componentes.
- Unidad de control: Es la encargada de procesar las entradas y salidas, será configurable y será donde se implementarán los algoritmos necesarios para este proyecto.
- Actuador de movimiento: Es el encargado de proporcionar movimiento al carro, es la única salida de nuestro sistema.
- Transductor del joystick: Se encargad en convertir la señal física con la que se actúa en el joystick en una señal eléctrica que pueda procesar la unidad de control.

8 VALORACIÓN DE ALTERNATIVAS PARA EL DISEÑO

8.1 Elementos necesarios

Se describirán los elementos necesarios y los requisitos que se deben cumplir para este proyecto.

8.1.1 Unidad de control

Se necesita un módulo que gestione todas las entradas y salidas donde se implementará el filtro de retraso temporal. Se han valorado dos opciones muy similares, ambas son microcontroladores que es la opción más interesante para realizar sistemas embebidos, los dos microcontroladores son de la marca Cypress y cuentan con altas prestaciones que más adelante se describirán. Estas dos posibilidades son, la PSoC 4 y la PSoC 5LP. La característica que ha motivado la decisión ha sido la velocidad de procesamiento, dado que esta aplicación se realizará en tiempo real la velocidad es clave para el correcto funcionamiento.

Microcontrolador	Velocidad
PSoC 4	48 MHz
PSoC 5LP	74 MHz

Tabla 4: Comparación de velocidades microcontroladores

Una ventaja que presenta la PSoC 4 es que cuenta con un controlador que se integra fácilmente conectándolo ambas placas una encima de la otra, como se observa en la siguiente.

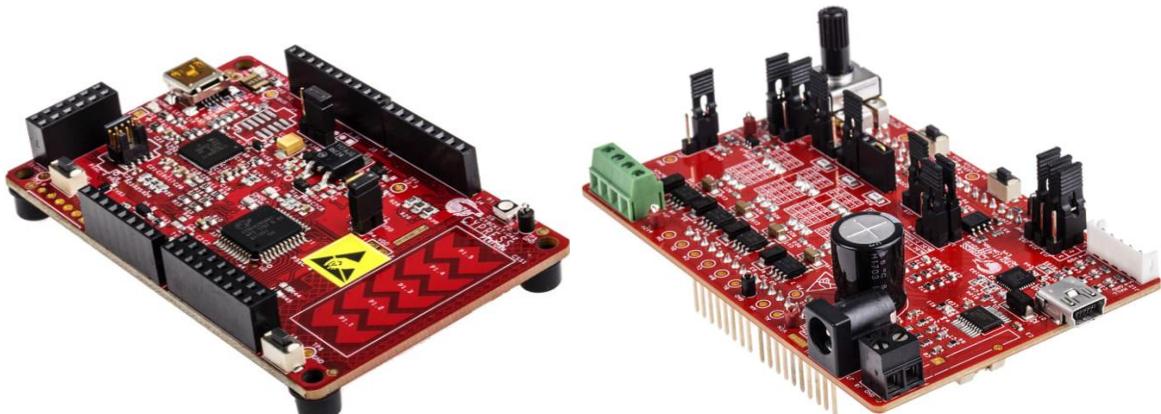


Figura 23: PSoC 4 con el controlador a su derecha. [6]

Dada esta ventaja se valoró realizar la implementación del filtro en la PSoC 5LP para después transmitir los datos a la PSoC 4 cuya única funcionalidad sería controlar el motor, finalmente se descartó por la inconveniencia de necesitar dos microcontroladores lo cual complica la implementación.

8.1.2 Sensor de consigna de movimiento

La consigna de movimiento viene dada por un operario que controlará un sensor de movimiento para decidir la dirección y la velocidad de la carga. Para el control de un puente grúa lo más habitual es utilizar mandos a distancia que se comunican por radio con la unidad de control de la grúa, están provistos de diversos actuadores para su control, como potenciómetros, selectores rotatorios y pulsadores. Un ejemplo de ese mando de control podría ser el modelo Curve AJQ de la marca Autec Radio Remote Control S.L. que cuenta incluso con un display TFT LCD.



Figura 24: Mando de control Curve AJQ de la marca Autec Radio Remote Control S.L. [23]

Como se realiza una simulación se valoraron dos opciones más sencillas:

- Joystick: El joystick cuenta con dos potenciómetros uno en el eje x y otro en el eje y, como en este sistema solo se necesita movimiento en un solo eje, se utiliza uno de los potenciómetros, otra ventaja con la que cuenta el joystick es el botón que tiene integrado, este permite activar y desactivar el filtrado de manera rápida.
- Potenciómetro lineal: Este potenciómetro de $10\text{ K}\Omega$ tiene una ventaja importante con respecto al joystick, su precisión es mucho mayor dado que cuenta con 10mm de carril para ajustar la velocidad a la deseada. Por otro lado, el potenciómetro no tiene retorno automático por lo que si el operario lo suelta la grúa no dejaría de avanzar lo cual supone un aumento en el riesgo. Otra desventaja es que para definir el sentido de movimiento se tendría que añadir un botón pues el potenciómetro solo definiría la velocidad.



Figura 25: Potenciómetro lineal de la marca RS Components. [26]

Dadas las desventajas del potenciómetro lineal finalmente se ha escogido el joystick.

8.1.3 Controlador del actuador

Como se ha mencionado antes se ha valorado la posibilidad de implementar el controlador que viene con el kit CY8CKIT-037 de la PSoC 4 [6]. Este controlador permite tener la flexibilidad de adaptarlo a diferentes motores, tanto motores de corriente continua con escobillas como paso a paso. Se ha descartado esta opción por las razones anteriormente mencionadas ya que esta placa controladora no se puede conectar a la PSoC 5LP. Por tanto, se optará por controladores más sencillos:

- L293D: Es un circuito integrado que se compone de dos configuraciones de diodos en puente en H, lo que permite manejar hasta dos motores, esta opción fue descartada por no cumplir con los requisitos de intensidad máxima que permite el circuito [7]. Máximo de 0,6 A de salida.

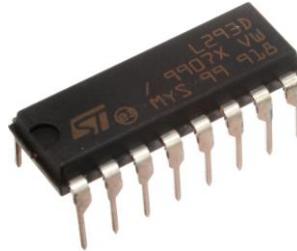


Figura 26: Circuito integrado L293D. [7]

- L298N: Esta placa que lleva integrada el chip L298 el cual permite el control de hasta dos motores de corriente continua al mismo tiempo. Tiene las características solicitadas para este uso, cumpliendo con los requisitos de intensidad máxima. Más adelante se describirán sus características.



Figura 27: Circuito integrado L298N.

8.1.4 Actuador de posición

El actuador de posición es el que recibe la consigna de movimiento y el que actúa sobre la carga, convirtiendo la señal eléctrica en rotación de su eje, el cual, mecánicamente se convertirá en movimiento lineal del carro. Se han valorado dos posibilidades:

- Motor de corriente continua con escobillas: Son los más convencionales e ideales para aplicaciones de baja potencia, como es este caso. Son económicos y sencillos de manejar. Dado que están diseñados para altas velocidades, la mejor opción es escoger un motor que lleve acoplado una reductora. Más adelante se describirán sus características.

- Motor paso a paso: Son considerablemente más caros y el control de este tipo de motores es más complejo, pero tienen la ventaja que sus movimientos son mucho más precisos. Lo cual no es un requerimiento de este proyecto, pues se busca un filtro robusto que funcione ante cualquier desventaja. Dado que este proyecto busca la simplicidad y sencillez de implementación de los filtros de retraso temporal se ha descartado esta opción. El kit de la PSoC 4 anteriormente mencionado incluye el motor paso a paso que se muestra en la siguiente figura:

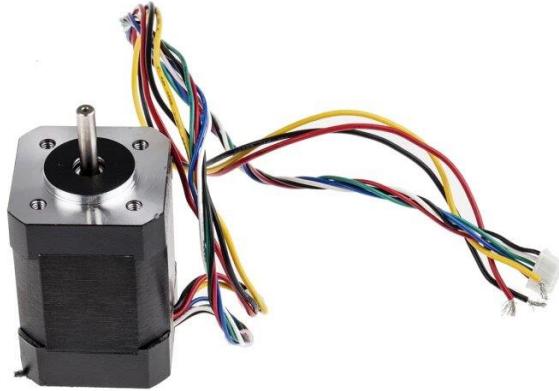


Figura 28: Motor paso a paso del Kit de evaluación Cypress Semiconductor PSoC 4 Pioneer Kit [27].

9 ELEMENTOS FINALES DEL DISEÑO

9.1 Unidad de control

Como unidad de control se utiliza una PSoC 5LP Cortex-M3 (Programable System on Chip). Pertenece a una familia de microcontroladores desarrollados por Cypress Semiconductor en 2002. Está formado por un chip configurable, tiene una matriz también configurable de funciones analógicas solapada con otras funciones digitales, por lo cual este microcontrolador es muy versátil. El microcontrolador PSoC 5LP proporciona funciones tanto de unidad de control como de memoria y periféricos analógicos y digitales en un solo chip.

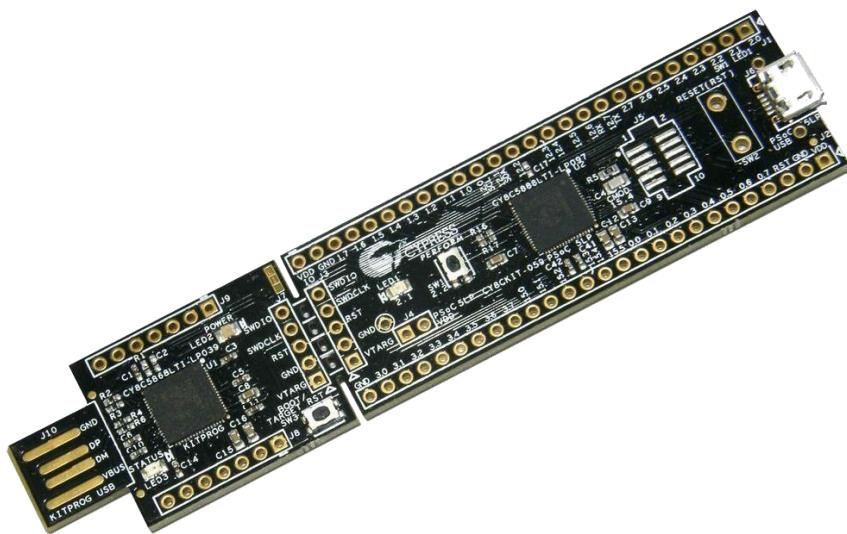


Figura 29: PSoC 5LP Arm Cortex-M3 [19].

Tiene las siguientes características que lo hacen idóneo para este propósito:

- Núcleo de CPU ARM Cortex-M3 de 32 bits.
 - Tiene alta velocidad de procesamiento (reloj interno de hasta 74 MHz).
 - Periféricos digitales:
 - o Cuatro temporizadores de 16 bits.
 - o Contador.
 - o PWM.
 - Periféricos analógicos:
 - o Convertidor analógico digital (CAD) delta-sigma configurable de 8 a 12 bits.
 - o Dos convertidores analógico digitales (CAD) SAR de 12 bits.
 - o Cuatro convertidores digitales analógicos (DAC).
 - o Cuatro comparadores.
 - o Cuatro amplificadores operacionales.
 - Hasta 62 GPIO (entradas y salidas de uso general).

Se representa en el siguiente diagrama:

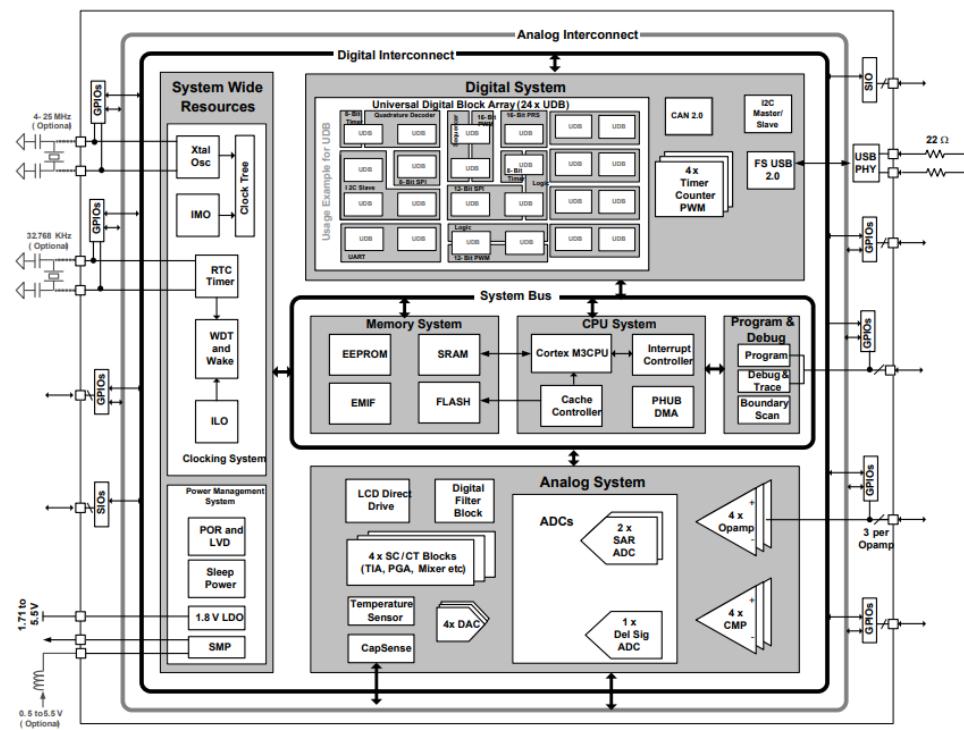


Figura 30: Diagrama de los componentes del microcontrolador PSoC 5LP [20].

Para configurar el microcontrolador, Cypress dispone de un entorno gráfico de programación llamado PSoC Creator que es un IDE gráfico de tipo Pick&Place, el compilador es de lenguaje C.

9.2 Actuador de movimiento

Una grúa de pórtico suele implementar motores trifásicos de muy altos voltajes para el movimiento de las cargas. En este caso solo se requiere de una demostración por lo que se escoge un motor de corriente continua con escobillas para probar el funcionamiento del filtro de retraso temporal, es una solución sencilla y económica.

Dado que el objetivo es mover una carga a baja velocidad se opta por un motor con una reductora acoplada. El motor escogido es de la marca RS PRO, número 2588293 [8], se trata de un motor de 24V con escobillas acoplado a una reductora planetaria, el par máximo es más que suficiente para mover la carga.

En el Anexo II se puede consultar la hoja de características abreviada de este motor.



Figura 31: Motor con reductora de escobillas de la marca RS PRO.

Las características del motor son las siguientes:

Modelo	Voltaje (V)	Par (Nm)	Reducción	Velocidad (rpm)	Peso (Kg)
2588293	24	0.1	20:1	190	0.668

Tabla 5: Características del motor.

9.3 Controlador del motor

Para poder manejar el actuador de movimiento se utilizará una placa para el control del motor que lleva implementado el chip L298 de doble puente en H (Figura 31).

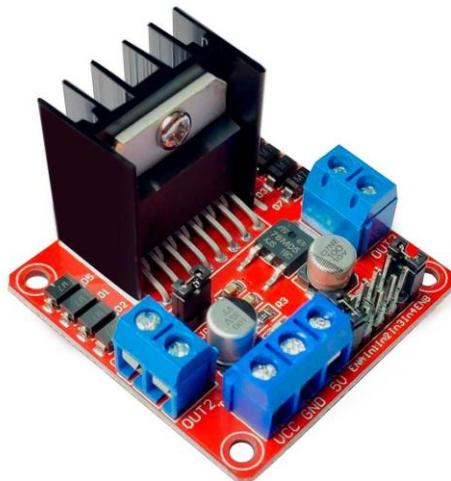


Figura 32: Controlador L298n [21].

La placa permite controlar hasta dos motores, consta de dos puentes en H uno para cada salida, de forma que se puede invertir el sentido de giro de los motores. También incorpora electrónica para agrupar en 3 pines la conexión al puente en H lo que evita la posibilidad de generar un cortocircuito. La entrada de tensión tiene un rango entre 3V y 35 V, cuando la tensión de entrada no está comprendida entre 5V y 15V habrá, además, que alimentar el controlador con 5 V. Para consultar más detalles en el Anexo II se incluye la hoja de características del chip L298.

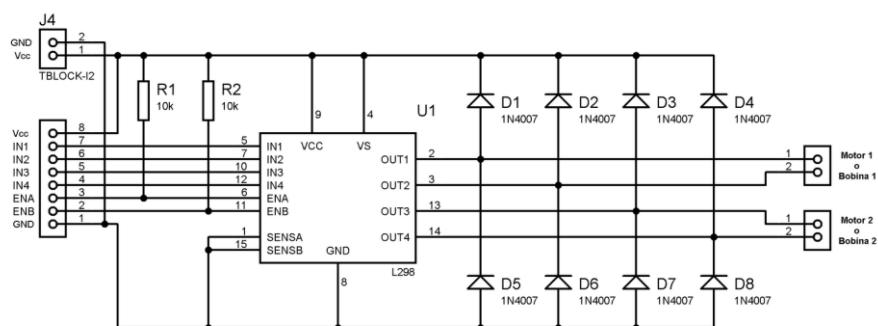


Figura 33: Esquema eléctrico del controlador L298n [22].

9.4 Interacción con el operario

Como se explica en el apartado 8.1.2 se cuenta con un Joystick sencillo (Figura 33), dispone de dos potenciómetros, uno en el eje x y otro en el eje y, dado que en este proyecto solo se necesita desplazamiento en un eje solo se hace uso de uno de los potenciómetros. Que es el responsable de generar la señal analógica de control de velocidad del motor.

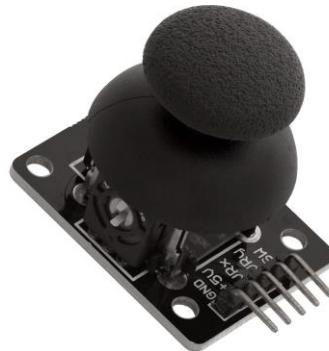


Figura 34: Joystick Keyes sJoys. [24]

10 DESARROLLO

10.1 Esquema eléctrico

Una vez se conocen los componentes que se van a utilizar, se realizará en conexiónado entre ellos para tener una clara visión de su funcionamiento.

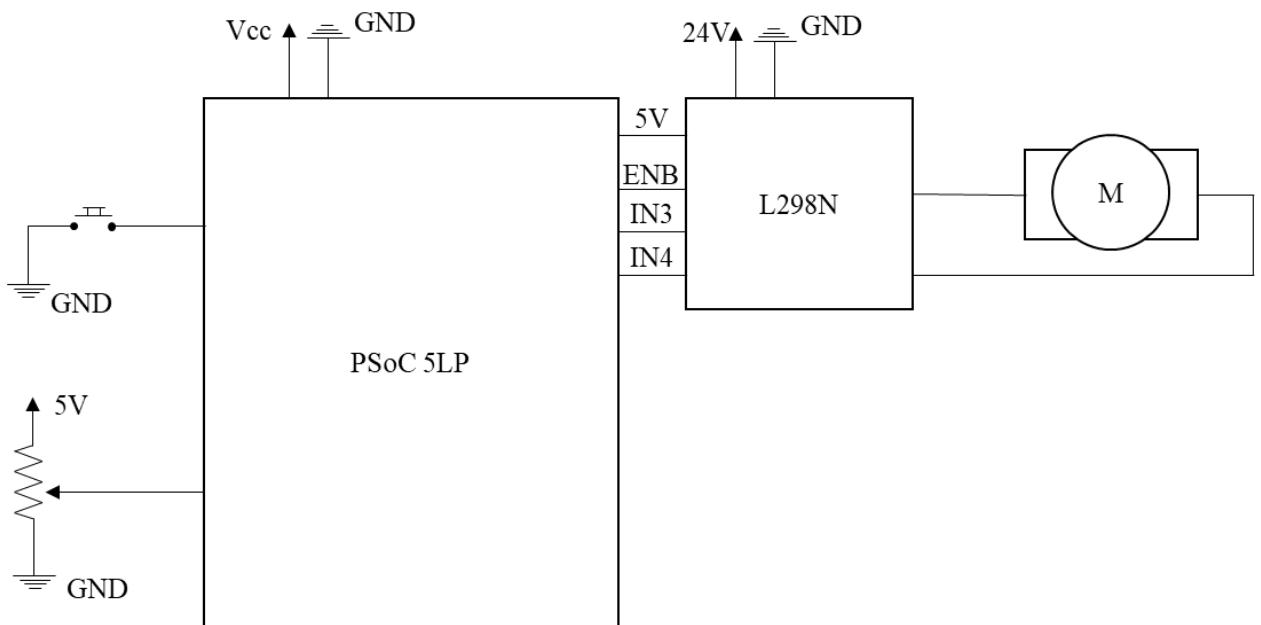


Figura 35: Esquema eléctrico. Elaboración propia.

- La alimentación proviene de una fuente de tensión de 24 V la cual alimenta el controlador del motor.
- El controlador también necesita una alimentación a 5V cuando el voltaje del motor supera los 15 V por ello está conectado a V_{dd} del microcontrolador.
- Solo se utilizará una de las dos posibles conexiones para motores del controlador. Se dispone de 3 pines diferentes para su control.
 - ENB: La función de este pin es habilitar su conexión con la tensión deseada (la que proviene de la fuente). Se hace uso de este pin para transmitir una señal PWM, la variación del ciclo de trabajo de esta señal permitirá tener control sobre la velocidad del motor.
 - IN3 Y IN4: Estos dos pines sirven para controlar el sentido de giro del motor. Por ello se conecta IN3 y IN4 dos salidas digitales para que el motor gire en el sentido deseado.
- Para el control de la velocidad del motor se dispone de un joystick desde el cual saldrá la señal analógica que se procesa con el microcontrolador PSoC 5LP.
- Finalmente se dispone de un pulsador, la activación de este pulsador provocará la activación o desactivación del filtro de retraso temporal para que se pueda comparar ambos casos. Al estar conexiónado a GND deberá configurarse el microcontrolador para utilizar el pull-up interno.

El conexionado entre los diferentes elementos sería el siguiente:

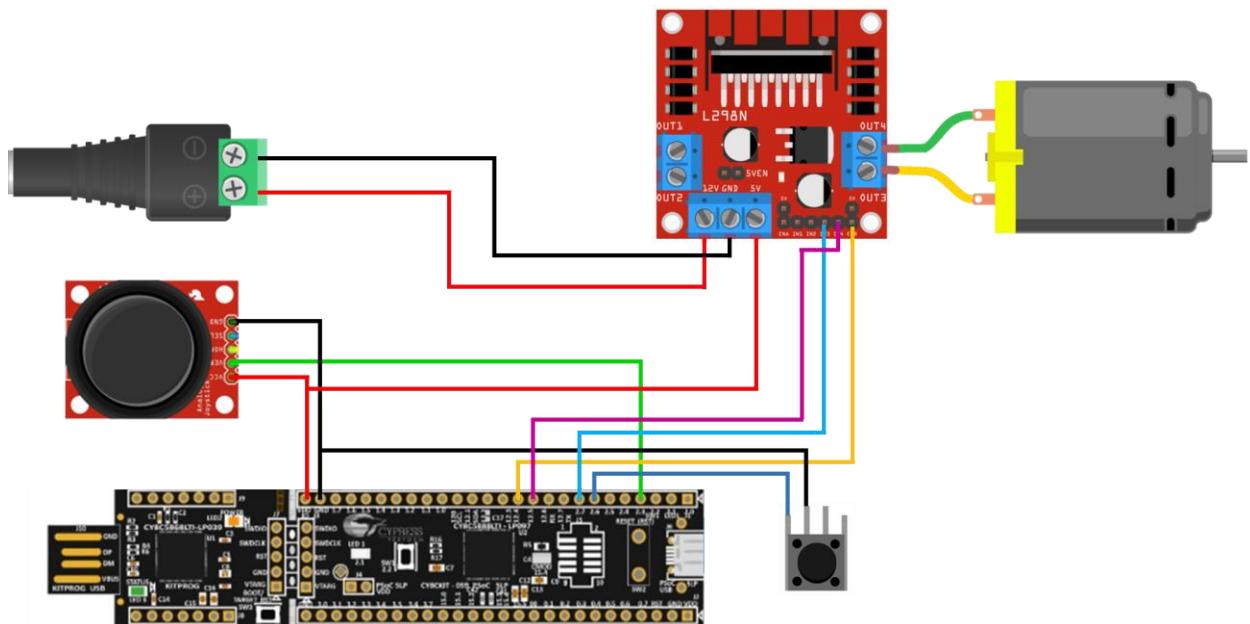


Figura 36: Conexionado de los distintos elementos electrónicos. Elaboración propia. El programa para el diseño del conexionado es Fritzing.

10.2 Recursos del microcontrolador

Para poder conectar los elementos al microcontrolador se utilizarán distintos recursos con los que cuenta el microcontrolador.

10.2.1 Motor

Para el control del motor se necesita una única salida digital, conectada a la señal de habilitación del controlador del motor (ENB). Se utilizará una señal PWM, para generarla el microcontrolador cuenta con 24 bloques digitales universales (UDB) programables para crear entre distintos tipos de módulos digitales, PWMs de hasta 16 bits.

Para este caso se necesita una única señal PWM, dado que la señal analógica que se obtiene del potenciómetro tiene una resolución de 8 bits, el módulo PWM contará con la misma resolución, 8 bits.

Para determinar la frecuencia de la señal PWM se han probado distintos valores hasta obtener uno con una respuesta óptima llegando a una frecuencia aproximada de 10 KHz.

Dado que los microcontroladores de la familia Cypress cuentan con el programa PSoC Creator es sencillo configurar el módulo PWM. Se inserta en el esquemático que contiene los distintos módulos que se utilizarán para el programa y se conectan entre ellos. Una vez insertado solo se tiene que conectar la salida del PWM al pin de salida digital, a la entrada “clock” del PWM se le conecta un clock, “Clock_1” de la frecuencia escogida, 10 KHz. Entre las propiedades del módulo se encuentra la resolución que se configurara como 8 bits.

La variación del ciclo de trabajo se realizará por programa, para que el ciclo de trabajo cambie correctamente se debe resetear el módulo antes de variarlo, para ello en la entrada “reset” del módulo PWM se le conectará una entrada digital “Reset_Pin” conectada a una señal que se podrá controlar desde el código y así resetear el módulo PWM.

El resultado esquemático es el siguiente:

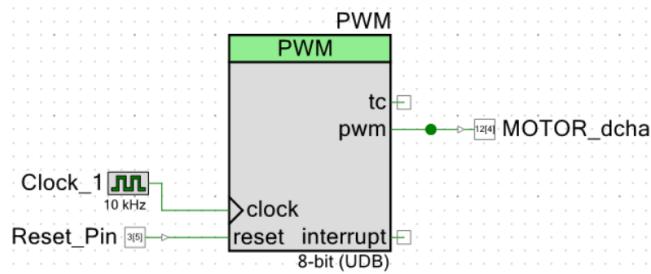


Figura 37: Visualización del periférico PWM en el esquemático de PSoC Creator.

Elaboración propia en el programa PSoC Creator.

10.2.2 Potenciómetro

Para la lectura del potenciómetro del joystick se habilita una entrada analógica, se utiliza uno de los dos posibles módulos Convertidor Analógico Digital de Aproximaciones Sucesivas ADC SAR. Este módulo es de hasta 12 bits de resolución y de una velocidad de hasta 1 msps (un millón de muestras por segundo).

Para la aplicación que corresponde la resolución será fijada a la mínima posible, 8 bits, dado que el potenciómetro tiene un rango de movimiento de 20mm cada valor cambiará cada 0,0784mm que es resultado de dividir los 20mm entre los 255 posibles estados. Esta resolución es más que suficiente para que el operario tenga la máxima precisión posible.

La frecuencia de reloj que se le transmite a este módulo coincidirá con la del reloj interno del microcontrolador. Dado el objetivo es tener un constante control sobre la medida del potenciómetro el modo de muestreo (Sample Mode) del módulo ADC se configurará como “Free Running” esto significa que una vez se inicie el convertidor en el código, este estará convirtiendo todo el tiempo a la velocidad de muestreo hasta que se pare con otra instrucción en el código. El voltaje de referencia será V_{ref} interno.

El resultado esquemático es el siguiente:

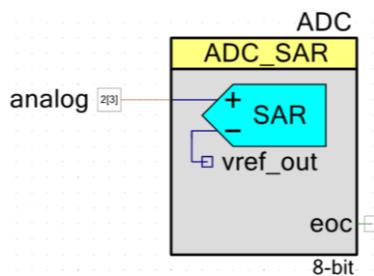


Figura 38: Visualización del módulo ADC en el esquemático de PSoC Creator. Elaboración propia en el programa PSoC Creator.

10.2.3 Pulsador

Este pulsador tiene como objetivo activar o desactivar la aplicación del filtrado de retraso temporal para así poder comparar las dos respuestas del sistema de manera sencilla. Los recursos necesarios para implementarlo son, una entrada digital, y un bloque para eliminar los posibles rebotes “debouncer”. Esta entrada será atendida por interrupción.

El bloque para la eliminación de rebotes tiene como entrada el pin digital conectado al pulsador y en la salida su interrupción correspondiente. Es necesario conectarle un clock, que será

configurado a 100 Hz siendo la frecuencia óptima para para la lectura del pulsador en este caso. Será configurado para la lectura de un flanco positivo.

El bloque “debouncer” para la detección de un flanco positivo está compuesto de tres biestables y una puerta “or” con una entrada negada, dispuestos de la siguiente manera:

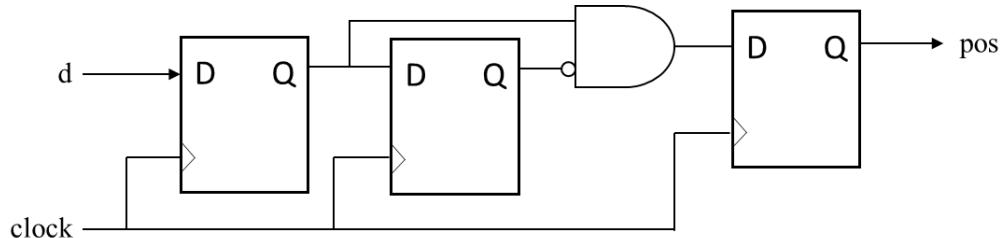


Figura 39: Esquema digital del comportamiento del módulo “debouncer”. Elaboración propia en el programa PSoC Creator.

El resultado esquemático es el siguiente:

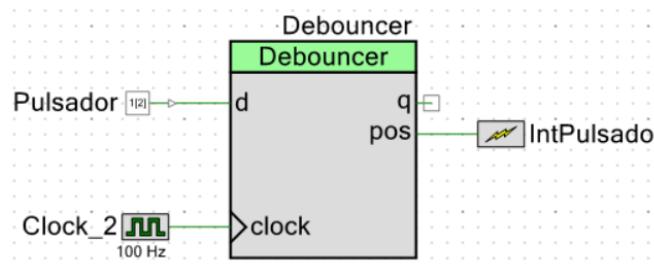


Figura 40: Visualización del módulo “debouncer” en el esquemático de PSoC Creator. Elaboración propia en el programa PSoC Creator.

10.2.4 Interrupciones

No se han definido niveles de prioridad en las interrupciones por lo que solo hay que habilitar las interrupciones globales. Se utilizarán dos interrupciones de las 32 posibles una ya ha sido explicada, para detectar el flanco positivo del pulsador y otra que se habilitará cuando se realice el filtrado de retraso temporal. Esta interrupción está conectada a un clock de 200Hz por lo que se activará cada 5 ms.

El resultado esquemático es el siguiente:



Figura 41: Visualización de la interrupción de convolución en el esquemático de PSoC Creator. Elaboración propia en el programa PSoC Creator.

10.2.5 Control de la dirección del motor

Para controlar la dirección de giro del motor el controlador cuenta con dos entradas IN3 e IN4, para su control se conectan a dos salidas digitales del microcontrolador. Para que el motor gire en sentido horario se debe activar la salida digital conectada a IN3 y desactivar la salida digital IN4 y viceversa para que gire en sentido antihorario.

El resultado esquemático es el siguiente:

2[7] dir_derecha

12[5] dir_izquierda

Figura 42: Visualización de las salidas digitales de dirección de giro en el esquemático de PSoC Creator. Elaboración propia en el programa PSoC Creator.

10.2.6 LED

Para comprobar físicamente si está activado o no el filtro de retraso temporal se dispone de un LED que está integrado en el microcontrolador. Se habilita una salida digital para el manejo del LED.

El resultado esquemático es el siguiente:

2[1] LED

Figura 43: Visualización la salida digital para controlar el LED en el esquemático de PSoC Creator. Elaboración propia en el programa PSoC Creator.

10.2.7 Resumen de los recursos necesarios

Para concluir se enumeran los recursos utilizados:

- 2 interrupciones
- 1 entrada analógica
- 1 entrada digital
- 4 salidas digitales
- 1 ADC (SAR)
- 3 Clocks
- 1 módulo PWM
- 1 LED integrado en el microcontrolador

10.2.8 Asignación de pines

En la siguiente tabla se ven los nombres de las entradas y salidas utilizadas y los puertos y pines asociados a las mismas.

Nombre	Puerto	Pin
Entrada analógica	P2[3]	65
IN4	P12[5]	4
IN3	P2[7]	2
LED	P2[1]	63
Botón	P2[6]	1

Tabla 6: Asignación de pines a las entradas y salidas utilizadas.

10.3 Programación

Los microcontroladores de la familia Cypress cuentan con el IDE (Entorno de diseño integrado) PSoC Creator para su programación en lenguaje C.

10.3.1 Distribución del programa

El programa se divide en el “main.c” donde se ejecuta el programa principal donde, en un bucle infinito se obtiene la consigna del potenciómetro y se actúa sobre el motor sin filtrado. Si se detecta la interrupción del pulsador se habilita la interrupción de convolución, en el archivo “convolucion.c” donde se realiza el mismo proceso, pero con el filtrado de retraso temporal.

Al añadir elementos al esquemático del programa, se generan nuevos archivos donde se almacenan las funciones de los mismos. Por ello el programa contará con más ficheros de los antes mencionados, pero no se modificarán para este caso.

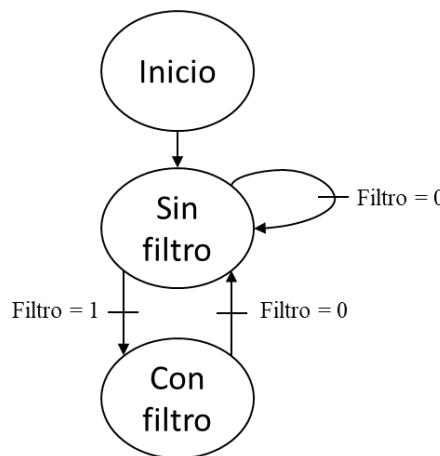


Figura 44: Máquina de estados del programa del microcontrolador PSoC 5LP. Elaboración propia.

10.3.2 Interrupciones

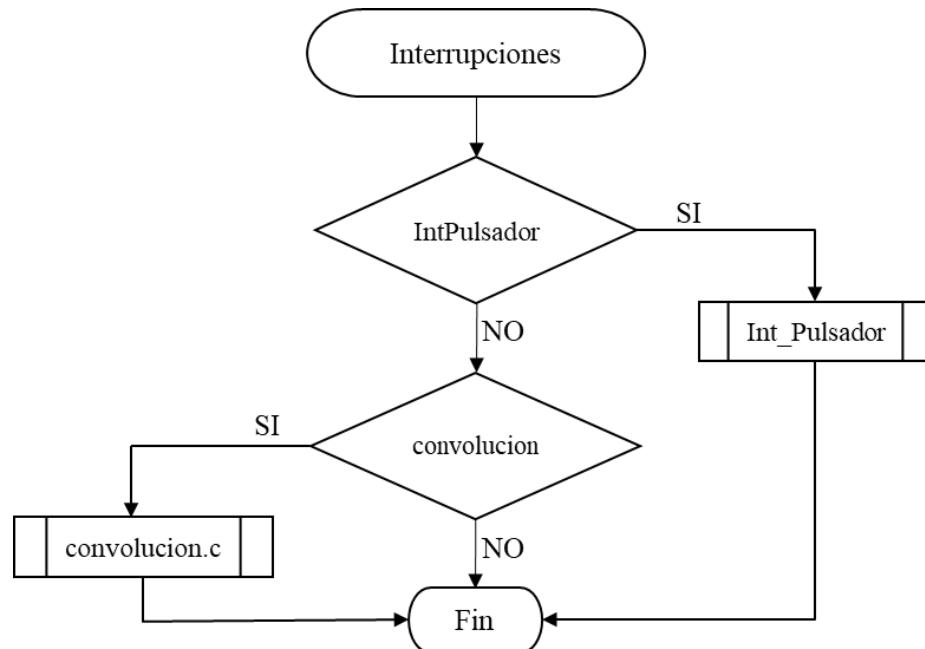


Figura 45: Diagrama de flujo de la rutina de atención a interrupciones. Elaboración propia.

10.3.3 Interrupción del pulsador

Se utilizar la variable “Filtro” para saber cuándo habilitar o deshabilitar la interrupción de convolución, por lo que el valor de la variable cambia cada vez que salta la interrupción del flanco positivo del pulsador. Tal y como se muestra en el diagrama de flujo.

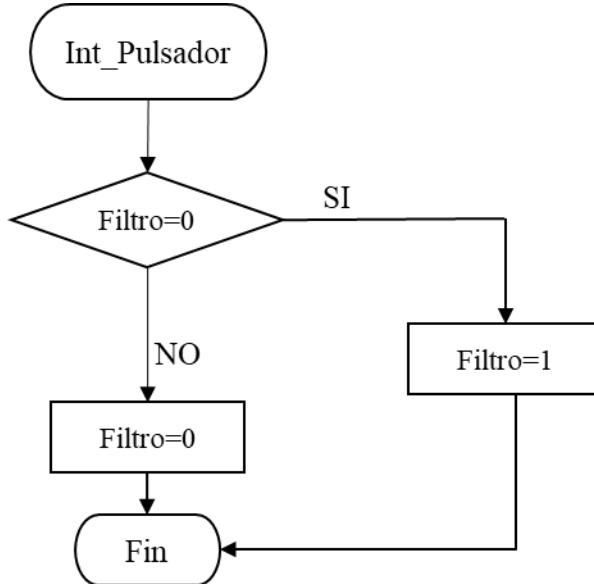


Figura 46: Diagrama de flujo de la función de interrupción del pulsador. Elaboración propia.

10.3.4 Bucle principal

En el bucle principal se obtendrá el valor que devuelve el ADC una vez almacenado en una variable se valora si se tiene que habilitar la interrupción de convolución. Esto se hará mediante una máquina de estados como se muestra en el siguiente diagrama. Con las variables “Filtro” que varía según la interrupción del pulsador y “Filtroant” que corresponde al estado de la variable “Filtro” en el ciclo anterior, se puede determinar el estado en el que se está. En cada ciclo del bucle principal también se resetea el valor del módulo PWM para poder cargar un valor nuevo.

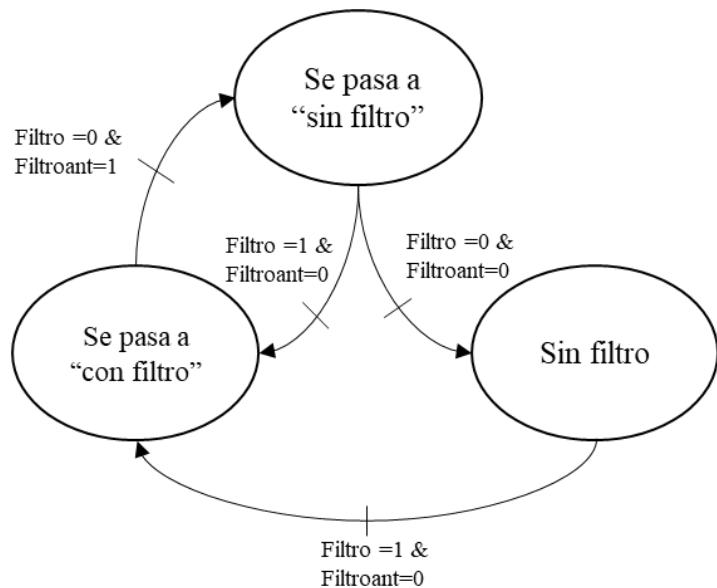


Figura 47: Máquina de estados para definir si aplicar el filtro o no. Elaboración propia.

- Se pasa a “con filtro”: Si Filtro=1 y Filtroant=0 se sabe que hay que activar el filtrado de retraso temporal, por lo tanto, se activará el LED, se reiniciarán las variables necesarias para realizar el algoritmo de convolución y se habilitará la interrupción de convolución.
- Se pasa a “sin filtro”: Si Filtro=0 y Filtroant=1 se sabe que hay que desactivar el filtrado de retraso temporal, por lo tanto, se deshabilitará la interrupción de convolución, se transmitirá el valor de consigna del motor al módulo PWM y se desactivará el LED.
- Sin filtro: Si Filtro=0 y Filtroant=0, se escribe en el módulo PWM el valor de la consigna de la velocidad a la que se quiere que se mueva el motor.

10.3.5 Cálculo de consignas

El cálculo de la consigna que se tiene que escribir en el módulo PWM, se realiza en una función “obtener_consigna”. El main.c del código se puede ver en el Anexo II. Esta función tiene como entrada el valor obtenido del ADC, y devuelve la consigna de velocidad para el motor.

El joystick en su posición de reposo devuelve aproximadamente la mitad del valor máximo es decir 129, debido a que el joystick no es muy preciso se dejará un margen para detectar si está en la posición de reposo.

El objetivo del cálculo de la consigna en función de la posición del joystick es la siguiente:

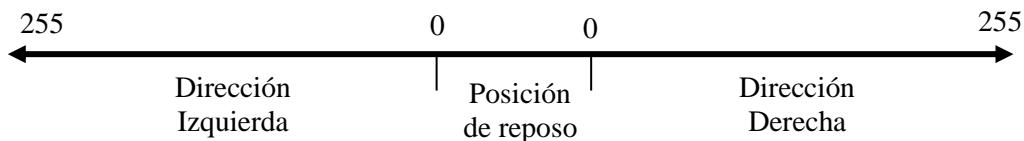


Figura 48: Diagrama que representa la posición del joystick junto a la consigna asociada para esa posición. Elaboración propia.

El algoritmo necesario para obtener esta consigna es el siguiente:

```
//Función obtener consigna
int obtener_consigna(int valorh)
{
    if ((valorh>132) &&(valorh<256))
    {
        valorfinal=(valorh-128)*2;
    }
    if ((valorh<128) &&(valorh>=0))
    {
        valorfinal=(127-valorh)*2;
    }
    if ((valorh>120) &&(valorh<135))
    {
        valorfinal=0;
    }
    return valorfinal;
}
```

Figura 49: Función para calcular de consigna en función del valor analógico. Elaboración propia.

10.3.6 Interrupción de convolución

En la interrupción de convolución se produce el filtrado de retraso temporal de la señal de consigna. Como el nombre del propio filtro indica se tiene un retraso temporal en la señal de salida hacia el motor, por eso se debe saber el estado en el que se está y almacenarlo, en el arranque (el operario acaba de mover el joystick), el movimiento intermedio (el operario mantiene el joystick fuera de la posición de reposo) o la parada (el operario deja en la posición de reposo el joystick).

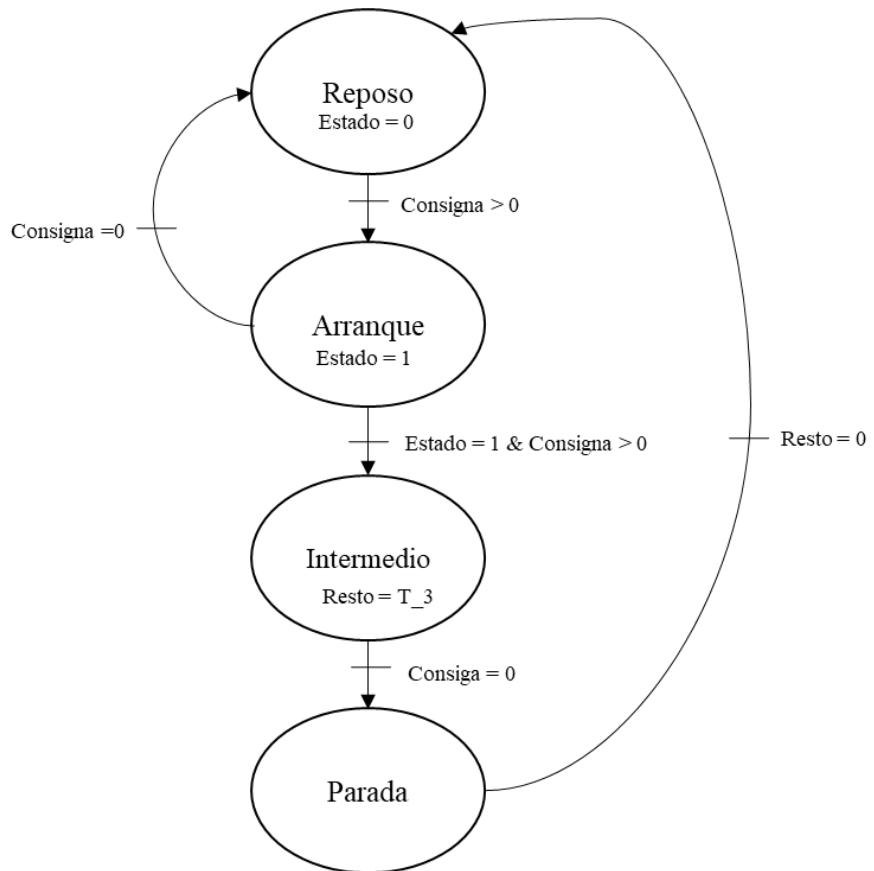


Figura 50: Diagrama de flujo de la función que se realiza en la interrupción de convolución.
Elaboración propia.

En los estados de arranque, intermedio y parada, para calcular la consigna de velocidad filtrada que se transmite al motor, se llama a una función que realiza su cálculo, este código está desarrollado por el profesor Gerardo Peláez Lourido [9].

Debido al filtrado de retraso temporal una vez se libere el joystick en la posición de reposo, el sistema continuará moviéndose un tiempo T_3 que corresponde al máximo valor de tiempo del filtro de retraso temporal, eso se comprobará experimentalmente en la simulación.

11 SIMULACIÓN EN MATLAB/SIMULINK

11.1 Proceso completo

Para explicar el proceso de simulación primero se realizará un resumen para tener una visión global del proceso y después se irá profundizando en cada paso.

Se puede prescindir de algunos elementos como el controlador y el motor, dado que solo interesa la señal filtrada que devuelve el microcontrolador.

La simulación no se realizará en tiempo real debido a la complejidad y velocidad de computación que esto requeriría, es decir primero se adquieren los datos de la consigna filtrada del microcontrolador y después, con esos datos, se realiza la simulación.

Los pasos para realizar la simulación son los siguientes:

- Virtualización del sistema: Se realiza en el programa SolidWorks.
- Importar el CAD al entorno Simulink.
- Implementar una transmisión serie en el microcontrolador, para enviar los datos de la señal filtrada a Matlab.
- Procesado de datos en Matlab.
- Simulación.

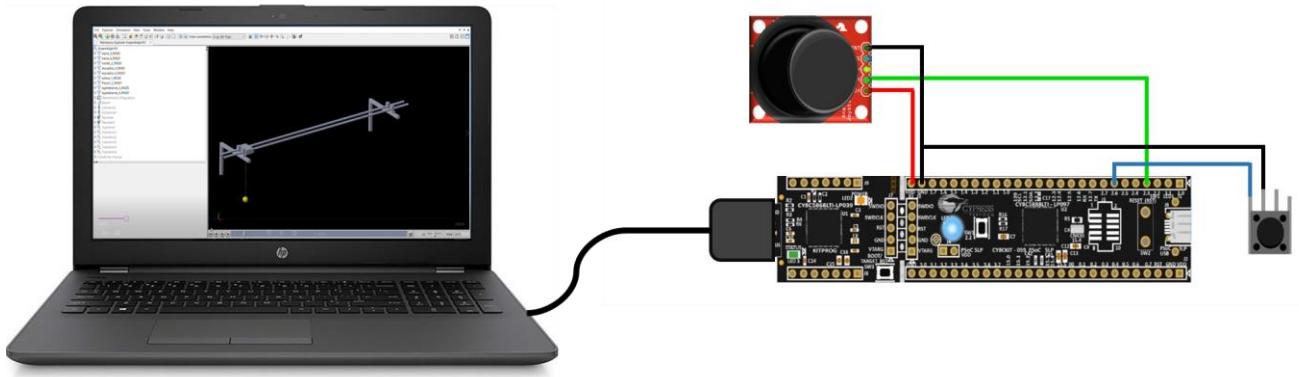


Figura 51: Esquema de los elementos necesarios para realizar la simulación.

11.2 Virtualización del sistema

Para la virtualización del sistema se realizará un diseño CAD en SolidWorks. El sistema a virtualizar es el siguiente:

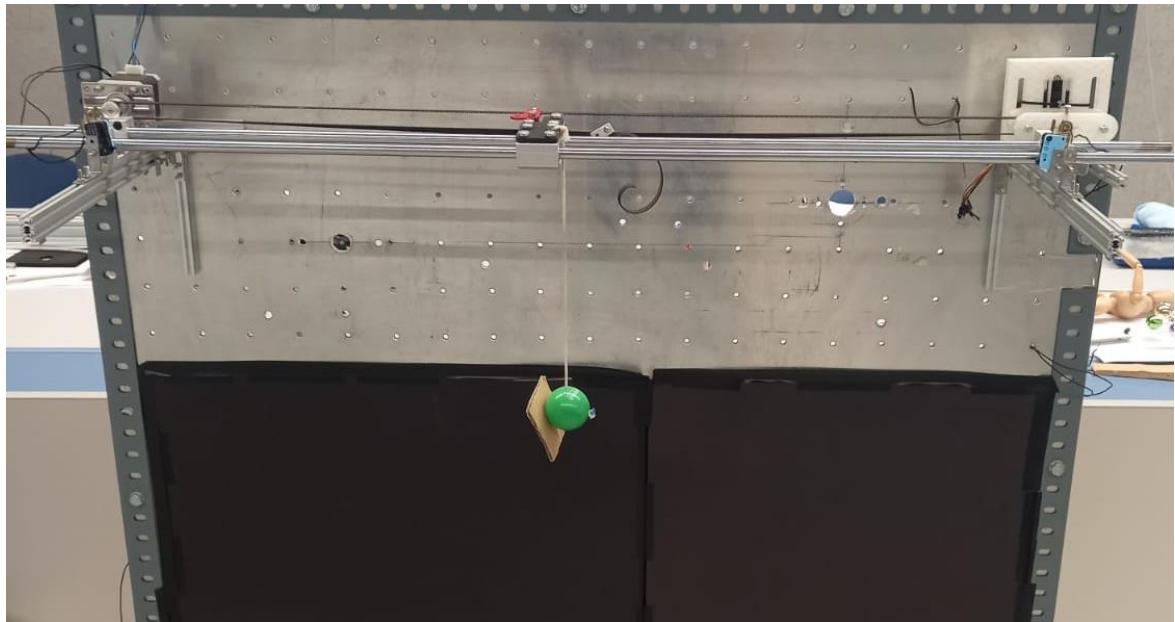


Figura 52: Maqueta de una “overhead crane” del laboratorio del profesor Gerardo Peláez, Universidad de Vigo.

11.2.1 Partes del montaje

El diseño CAD consta de las siguientes piezas:

Número pieza	Nombre	Cantidad	Utilidad
1	Soporte barras	2	Sujetan las barras a la escuadra.
2	Carro	1	Parte móvil que se mueve a través de las barras.
3	Fijador carro-correa	2	Se sujetan a la correa mediante presión y al carro con un tornillo.
4	Rodamiento	1	Para que la correa gire libremente en el extremo en el que no hay motor.
5	Escuadra	2	Fijan todo el conjunto a una pared.
6	Cordel	1	Es una barra con la posibilidad de balancearse que se fija al carro y a la carga.

7	Motor DC	1	Se encarga del movimiento de la correa.
8	Soporte motor	1	Sujeta el motor a la escuadra para fijarlo.
9	Soporte rodamiento	1	Sujeta el rodamiento a la escuadra.
10	Correa	1	Transmite el movimiento del motor al carro.
11	Esfera	1	Es la carga de la grúa, está unida al cable.
12	Barra	2	Son las guías del carro, están sujetas por los elementos número 1.

Tabla 7: Despiece del diseño CAD.

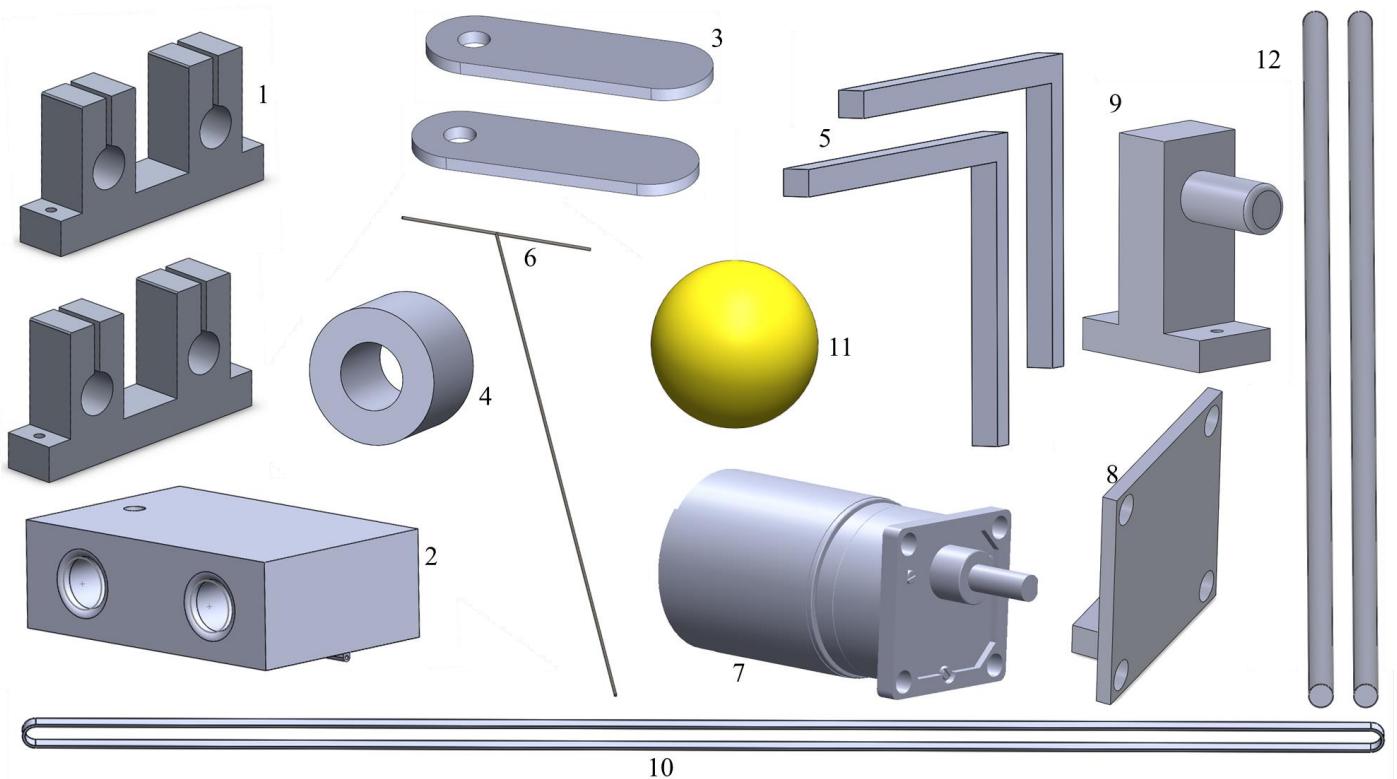


Figura 53: Despiece del diseño CAD. Elaboración propia en el programa de diseño SolidWorks.

11.2.2 *Ensamblaje*

Una vez unidas todas las piezas con sus respectivas relaciones de posición el montaje queda de la siguiente manera:

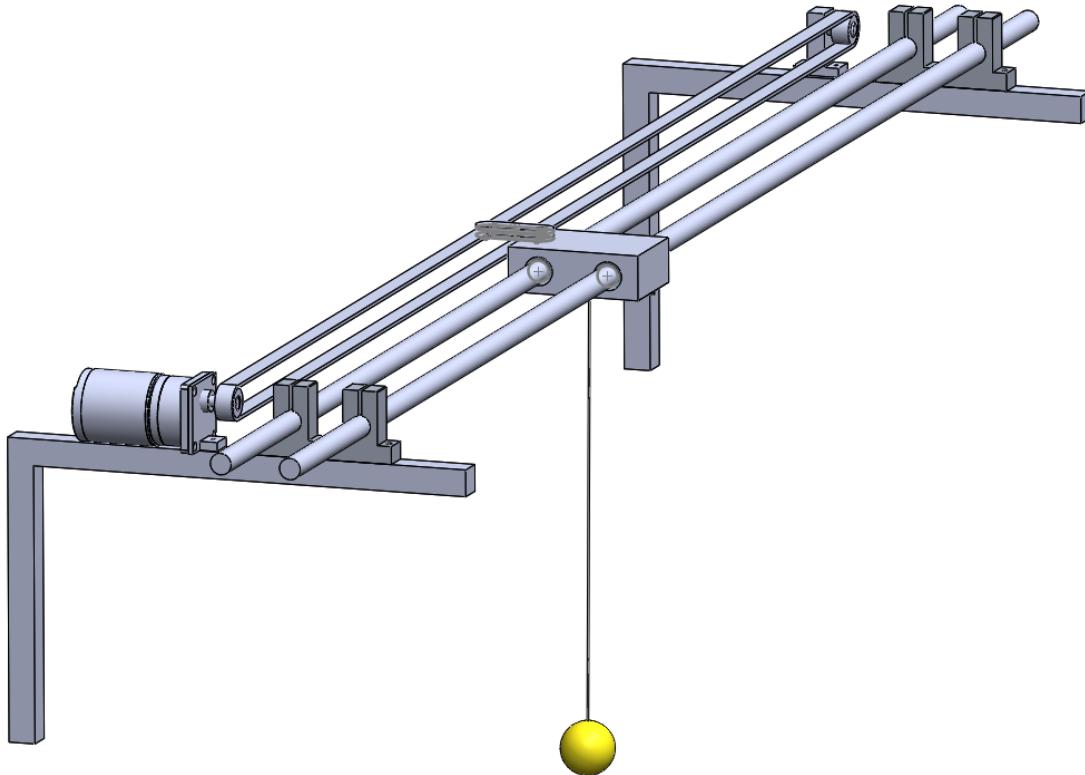


Figura 54: Diseño CAD en SolidWorks de una “overhead crane”. Elaboración propia en el programa de diseño SolidWorks.

El carro tiene libertad para moverse entre las dos escuadras a través de las barras. El motor al girar genera el movimiento de la correa, a su vez, el carro está unido a la correa a través de dos piezas (pieza nº 3) que aprietan la correa y transmiten el movimiento al carro.

Dada la dificultad para simular el comportamiento de un cordel se ha optado por sustituirlo por una barra y fijarlo al carro a través de una junta rotacional como se puede observar en la siguiente figura:

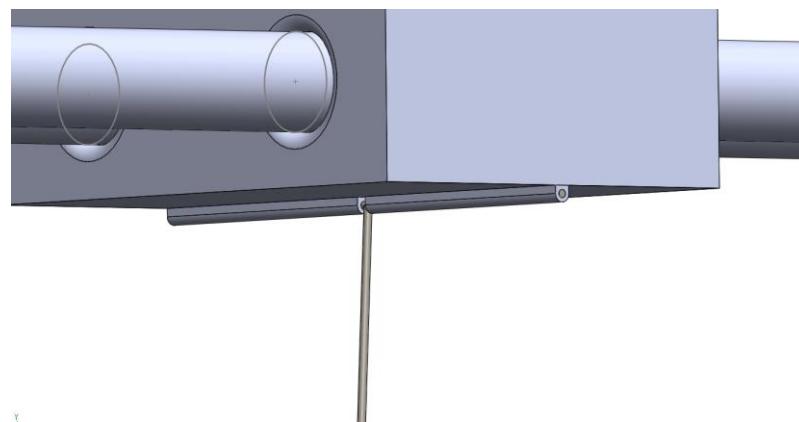


Figura 55: Vista en detalle de la juta rotacional que une el carro y el cable del sistema. Elaboración propia en el programa de diseño SolidWorks.

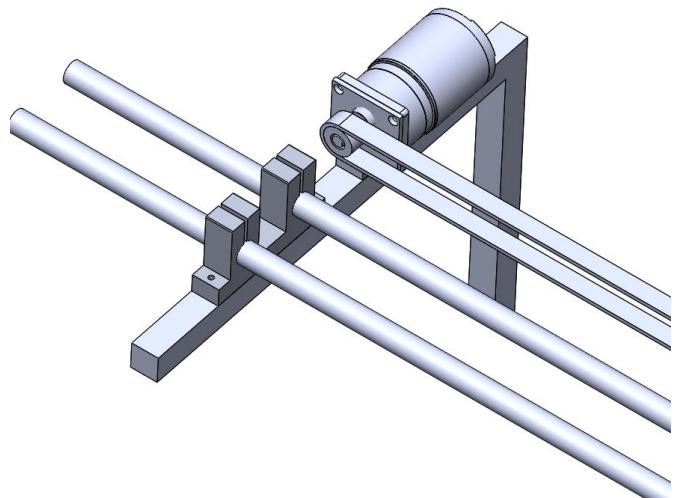
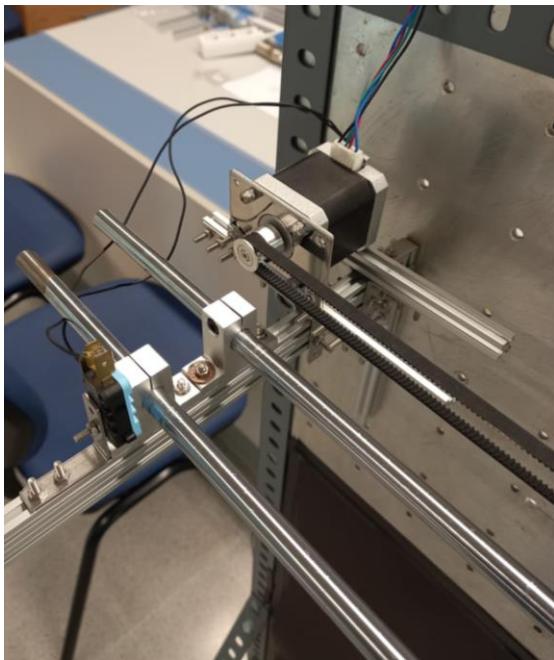


Figura 56: Comparación del diseño CAD (derecha) con la maqueta real (izquierda).

Para poder simularlo se exporta como un archivo “.xml” y para que en la simulación su libertad de movimiento sea la que corresponde se deben definir correctamente las relaciones de posición en SolidWorks.

11.3 Simscape

Simscape permite crear modelos de sistemas físicos dentro del entorno de Simulink, como el caso que estudiado. Una vez exportado se abre en Simulink como un archivo de Simscape, este sustituirá las relaciones de posición por bloques que simularán el movimiento físico del sistema. Para reducir la carga de procesamiento de simulink se eliminan algunos elementos innecesarios para la simulación, la correa, el motor, la pieza que sujeta el rodamiento y el rodamiento. Se exporta el siguiente sistema:

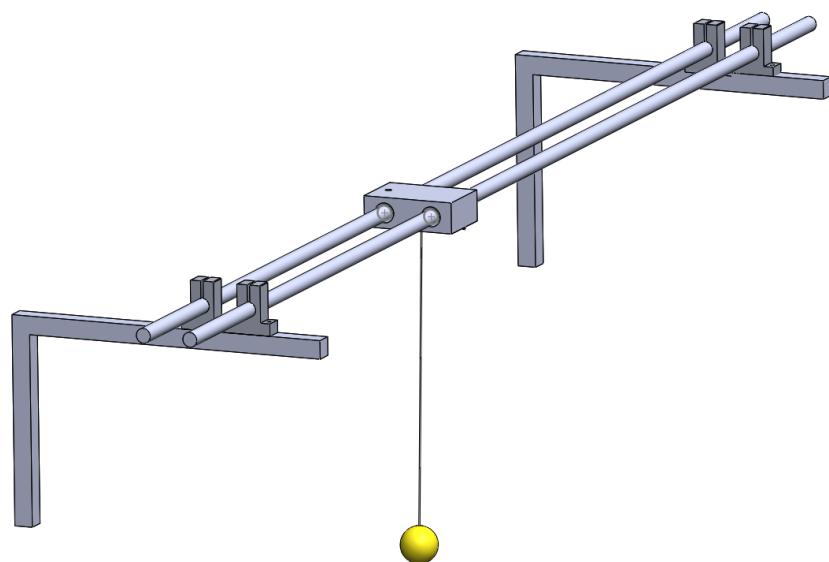


Figura 57: Diseño CAD simplificado para su simulación. Elaboración propia.

11.3.1 Bloques

Para la realización de la simulación, Simscape cuenta con algunos bloques que siempre están presentes, un bloque para definir el sistema de referencia, otro para definir parámetros que afectan a todo el mecanismo, como por ejemplo la gravedad y un último bloque que configura la resolución de la simulación.

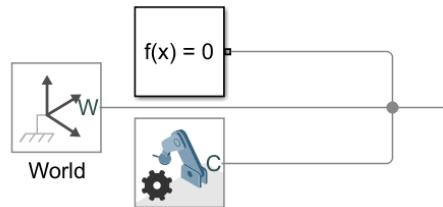


Figura 58: Diagrama de bloques necesarios para la simulación en el entorno Simulink.
Elaboración propia.

Después los bloques necesarios son los que definen la posición y características de los objetos fijos que en este caso son las escuadras las barras y los bloques que sujetan las barras. En la figura se puede observar que para cada objeto hay dos bloques, el primero define la posición del objeto con respecto al sistema de referencia y el segundo bloque define las características del objeto.

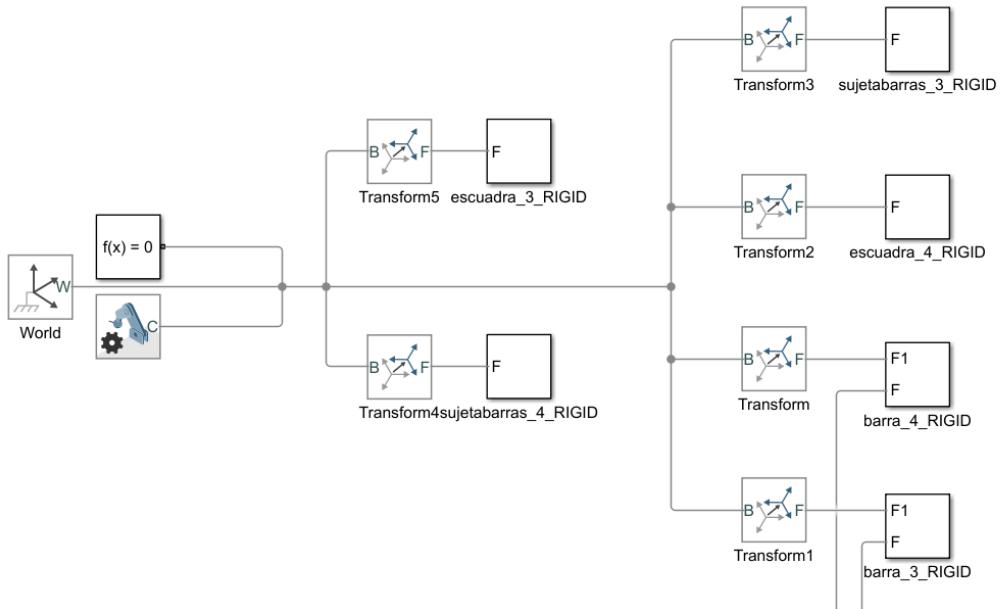


Figura 59: Diagrama de bloques de los objetos fijos en el entorno Simulink. Elaboración propia.

Finalmente, solo quedan los bloques de los objetos móviles y de las juntas que definen el tipo de movimiento. El carro se une a las barras mediante dos juntas cilíndricas con un movimiento que está limitado, solo puede ser translacional no rotacional, y el carro también se une al cable mediante una junta de revolución que permite al cable oscilar. El cable se une sin ningún movimiento posible a la esfera.

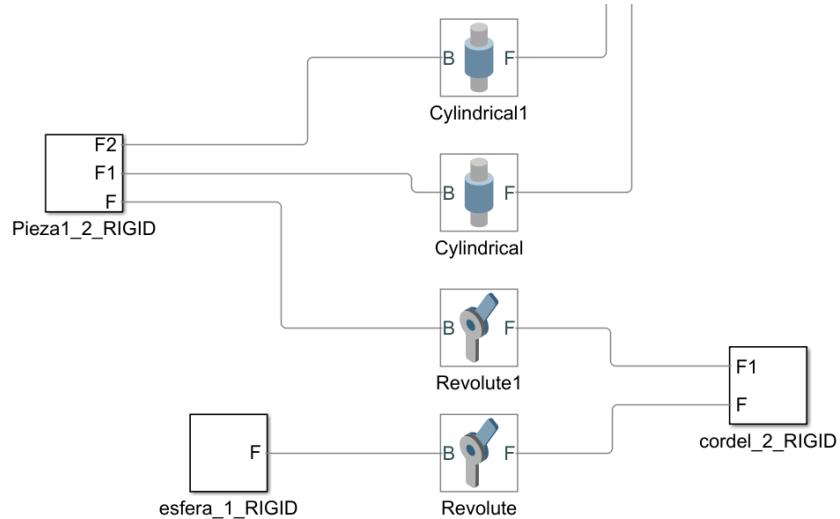


Figura 60: Diagrama de bloques de las juntas y objetos móviles en el entorno Simulink.
Elaboración propia.

Para trasmisitir el movimiento se configura la junta cilíndrica que une una de las barras y el carro para poder admitir movimiento. La matriz que define el movimiento se adquiere del “workspace” de Matlab, posteriormente se profundizará más en esto. Dado que las señales físicas necesarias y las señales simulink son diferentes se necesitan unos elementos para convertir estas señales. En la salida de la matriz se le añade este elemento cuya única configuración necesaria es definir las unidades en la que está la señal. Además, se configura la junta traslacional de la barra-carro para que tenga como salida su posición y se pueda graficar, al igual que con la junta rotacional del cable-carro para poder visualizar el ángulo de oscilación de la carga. Para poder visualizar estas señales en una gráfica “scope”, se añaden dos convertidores de señal física a señal de simulink. Queda como en la siguiente figura:

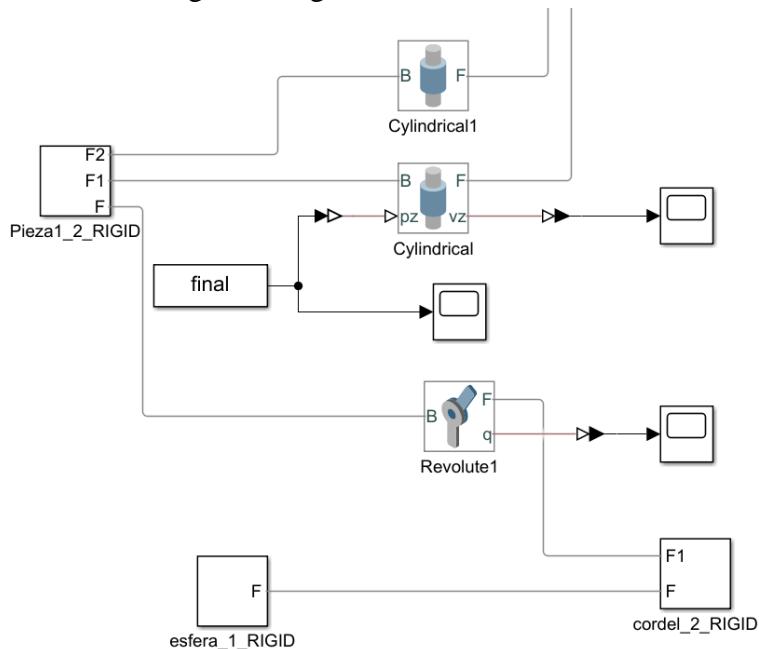


Figura 61: Diagrama de bloques de las entradas y salidas de las juntas móviles en el entorno Simulink. Elaboración propia.

11.4 Comunicaciones

El objetivo es trasmitir la señal PWM que recibe el controlador del motor a Matlab para que simulink pueda hacer uso de ella y realizar la simulación.

11.4.1 Transmisión de datos

La transmisión de datos se realiza desde el microcontrolador a el ordenador. Dado que PSoC Creator cuenta con un bloque de comunicación UART (Universal Asynchronous Transmitter Receiver) se hará uso de él. UART es un protocolo de comunicaciones en serie asíncrono.

Dado que UART es un protocolo de transmisión asíncrono se tiene que escoger una velocidad de transmisión para que mantenga la misma temporización de los bits en este caso se han escogido 115200 baudios, por ser la máxima velocidad que habitualmente se utiliza para este protocolo de comunicación.

El bloque utilizado en el esquemático de PSoC Creator es el siguiente:

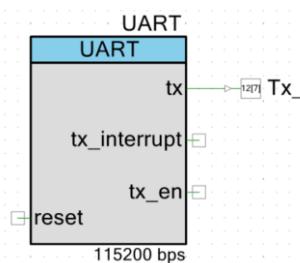


Figura 62: Visualización del módulo “UART” en el esquemático de PSoC Creator.

Elaboración propia en el porgrama PSoC Creator.

La comunicación será solamente transmisión, es decir, simplex. Finalmente, se tiene que saber el número de bits que ocupa el dato que se envía, dado que se utiliza una señal PWM de 8 bits (256 valores posibles), solo se necesitan 8 bits. La configuración del bloque es de la siguiente manera:

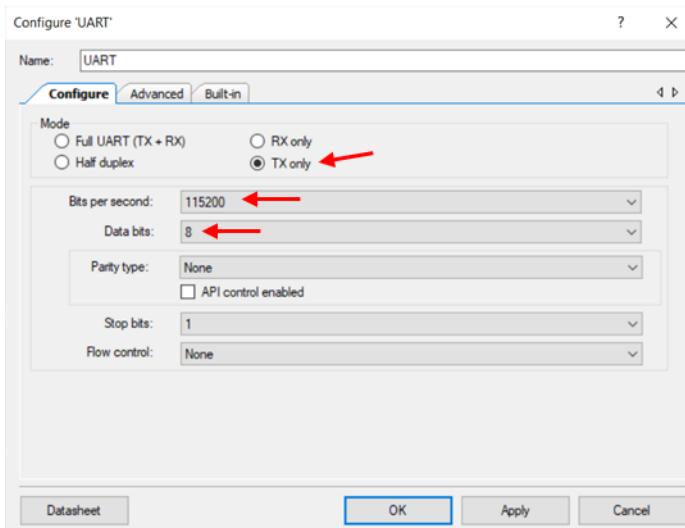


Figura 63: Configuración del módulo “UART” en el esquemático de PSoC Creator.

El código en el programa es muy sencillo solo se necesita añadir dos líneas, en la misma parte del código donde se envía el valor al periférico PWM:

En la primera línea, devuelve un “string” con el dato a enviar y la segunda lo transmite.

```
//transmisión
sprintf(str,"%d \n \r",valort);
UART_PutString(str);
```

Figura 64: Comandos para la transmisión. Elaboración propia.

11.4.2 Adquisición de datos

Para comprobar la correcta transmisión de los datos se utiliza el software PuTTY que tiene licencia libre y permite comprobar la recepción de los datos. Solo se tiene que especificar la velocidad de transmisión, el tipo de conexión (serie) y a que puerto está conectado. Al abrirlo aparecen los datos que se están recibiendo en tiempo real.

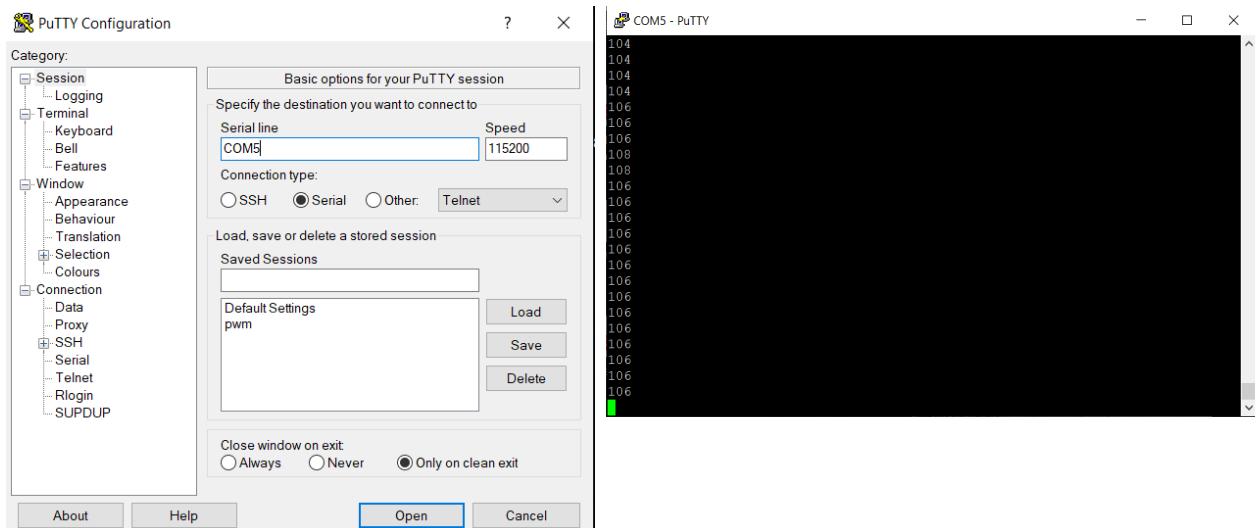


Figura 65: Configuración y recepción de los datos en el programa PuTTY.

Una vez se comprueba que la transmisión funciona correctamente se procederá a recibir esos datos en Matlab para que simulink pueda hacer uso de ellos.

```
puerto=serial("COM5"); % Se configura el puerto
puerto.BaudRate=115200; % Se configura la velocidad de transmisión
fopen(puerto); %Se abre el puerto
```

Figura 66: Configuración del puerto de entrada de datos. Elaboración propia.

Una vez configurado solo se tiene que realizar un bucle donde se lee el valor recibido con el comando `valor=fscanf(puerto,'%d')` y guardar los datos en un vector hasta que se llene por completo.

11.4.3 Procesado de datos

Para la simulación no hará falta simular el motor de continua. Simscape cuenta con la posibilidad de añadir actuación en las juntas traslacionales de las barras que sujetan el carro. Hay dos tipos de actuación posibles:

- Actuación por fuerza
- Actuación por movimiento

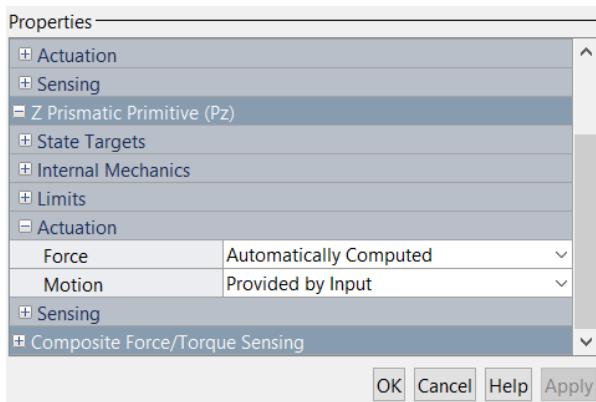


Figura 67: Configuración de la actuación en la junta traslacional barra-carro.

La actuación por fuerza simula peor el comportamiento que tendría un motor DC ya que se ve alterado por la inercia de la carga llegando a hacer retroceder al carro. Por el contrario, en la actuación por movimiento no se permite ninguna perturbación del exterior que afecte a la posición del carro por lo que se ha escogido finalmente la actuación por movimiento.

La consigna recibida es la señal PWM, es decir, un valor entre 0 y 255 que indica la velocidad del carro, para transformar esta velocidad en movimiento se realizan una serie de operaciones a los valores del vector que se ha recibido.

Se debe definir el sistema de referencia que se utiliza para saber la dirección del movimiento, en este caso será en el eje Z negativo.

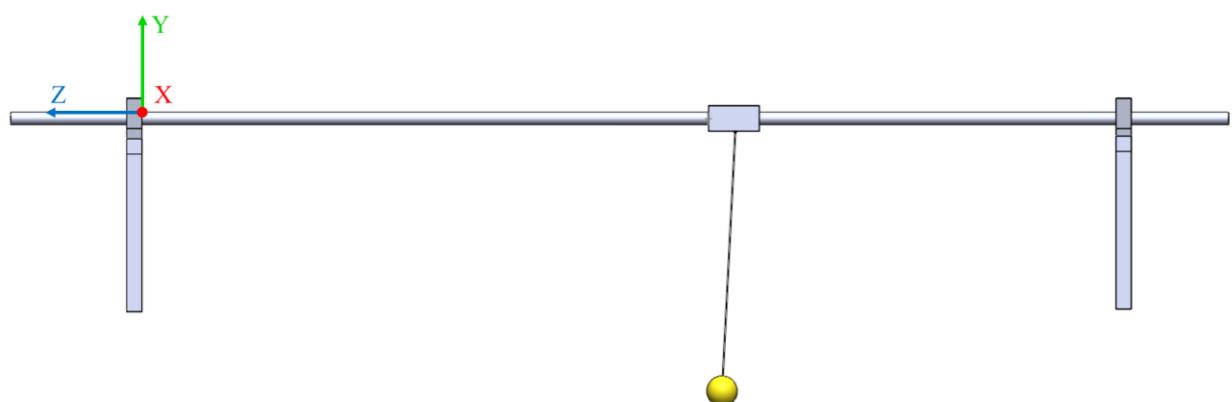


Figura 68: Sistema de referencia representado en el diseño CAD. Elaboración propia en el programa de diseño SolidWorks.

La señal de posición que se obtiene tiene forma de rampa descendente dado que avanza en el sentido negativo del eje Z, la pendiente mínima será de -1 (cuando la entrada tenga un valor de 255). Por esto lo primero es modificar el vector recibido para que los valores estén entre 0 y 1 en vez de entre 0 y 255, para ello se realiza un simple cálculo.

Una vez realizado esto se convierte la pendiente de la rampa a la posición donde se debe posicionar el carro.

El valor de la posición coincide con el valor de la posición anterior menos el valor de la pendiente de la rampa.

$$\text{posicion}(m) = \text{posicion}(m-1) - \text{PWM}(j);$$

Figura 69: Obtener el valor de la posición de consigna. Elaboración propia.

Además, para que Simulink pueda hacer uso de este vector se convierte a una matriz donde una columna coincide con el valor del tiempo, y la otra el valor de la posición. Esto se realiza creando una nueva matriz que se rellena con un valor de tiempo que aumenta 0.01 para cada valor y con el valor de la posición ya calculada. Una vez realizado esto Simulink podrá hacer uso de esta matriz para usarla de entrada en el movimiento del carro.

11.5 Video de la simulación

Para poder ver la simulación realizada y comparar la simulación con filtro y sin filtro se ha realizado un video. La imagen tiene un hipervínculo para acceder a su visualización.



Figura 70: Video de las simulaciones realizadas. Elaboración propia.

12 RESULTADOS

12.1 Resultados sin filtro

Primero se presentan los resultados sin filtro para poder posteriormente compararlos con la simulación con el filtro de retraso temporal aplicado.

12.1.1 Simulación

En la simulación generada por Simulink aun sin ver los resultados de la oscilación de la carga se puede apreciar un considerable balanceo de la carga. En la siguiente figura se puede observar el camino que sigue la carga para apreciar su oscilación. Es mucho más apreciable en el video anteriormente adjuntado (11.5).

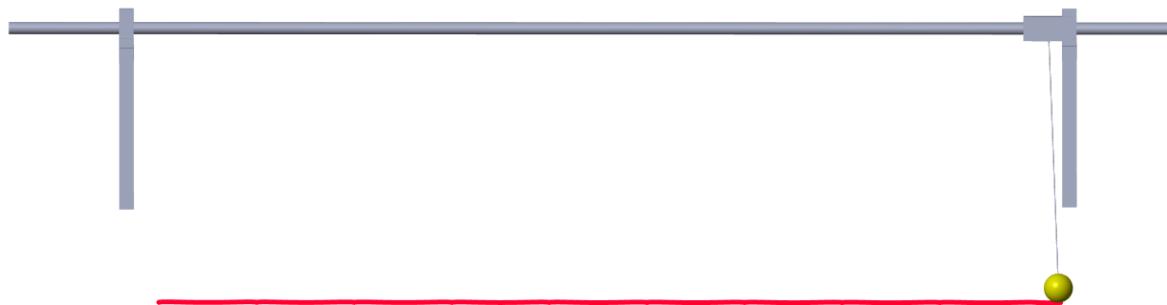


Figura 71: Trayectoria que sigue el punto inferior de la esfera sin aplicar el filtro ZVD.
Elaboración propia en el programa de simulación Simscape Multibody.

12.1.2 Gráficas

La primera gráfica que se muestra es la de la entrada que se recibe desde el microcontrolador.

Una vez procesados los datos se grafican los datos de la posición que seguirá el carro respecto al sistema de referencia anteriormente definido. Como se ha explicado antes la rampa es de pendiente negativa por avanzar en el sentido negativo del eje Z. En el eje de abscisas se representa el tiempo en segundos y en el eje de ordenadas el desplazamiento en milímetros.

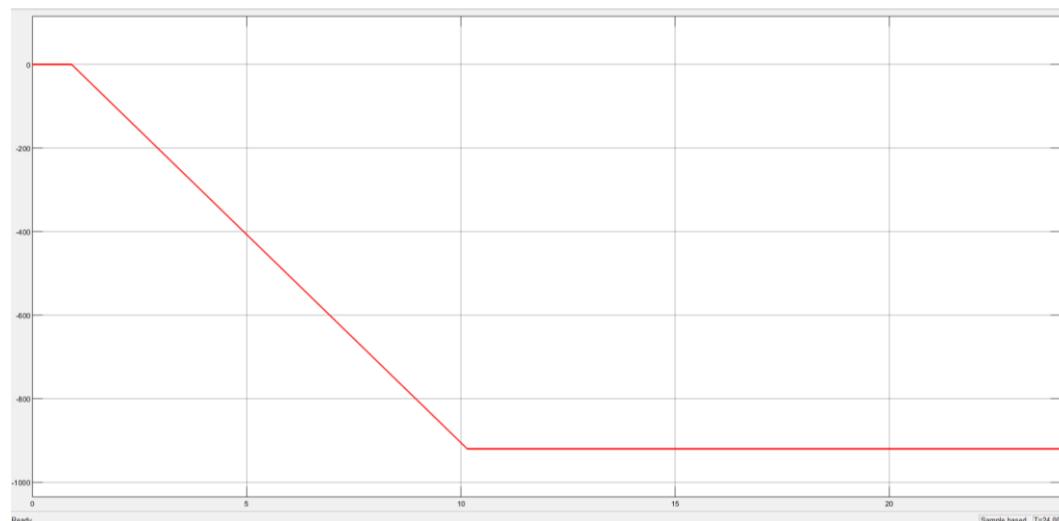


Figura 72: Grafica de la posición del carro en la simulación de Simscape. Sin filtro ZVD.
Elaboración propia.

La siguiente gráfica representa la señal de entrada a la simulación. Como se ha explicado en el apartado 11.4.1 en el eje de ordenadas se representa el ciclo de trabajo de la señal PWM, es decir un valor entre 0 y 255, en el eje de abscisas se representa el tiempo de la simulación en milisegundos.

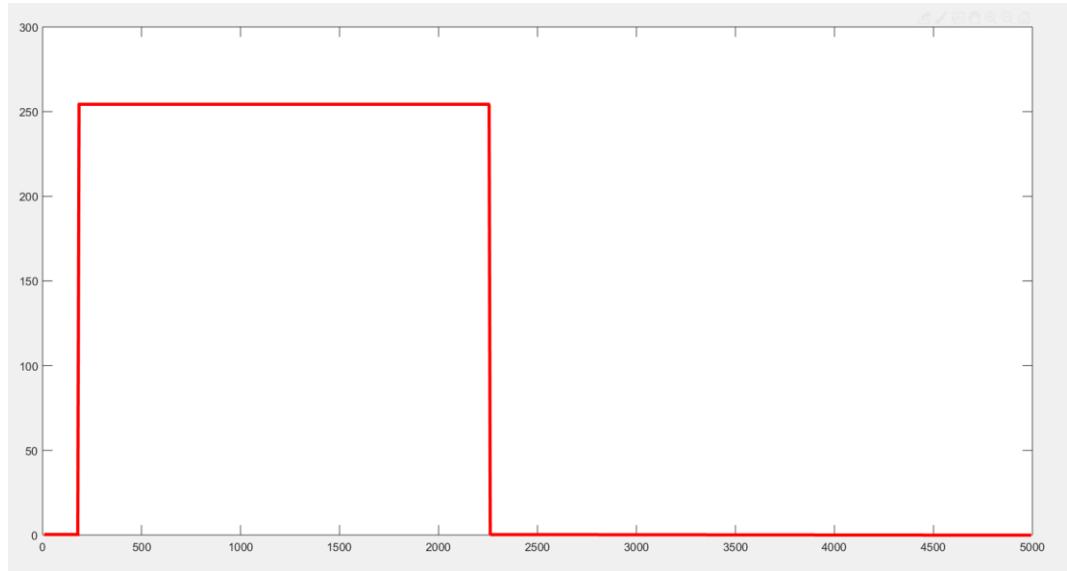


Figura 73: Gráfica de la señal de entrada que recibe Matlab del microcontrolador. Sin filtro ZVD. Elaboración propia.

La gráfica más ilustrativa es la que muestra el ángulo de separación de la carga con respecto a la vertical, muestra una oscilación constante, hasta que la carga llega al final donde la parada provoca un cambio en la amplitud de la oscilación. En el eje de abscisas se representa el tiempo en segundos, el movimiento comienza en el instante $T_1=1s$ y finaliza en $T_2=10,3s$. En el eje de ordenadas se representa el ángulo de oscilación de la carga en radianes, la amplitud coincide con $\alpha = 0,125 \text{ rad}$ en grados son $\alpha = 7,162^\circ$, este dato servirá más tarde para compararlo con la simulación con el filtro de retraso temporal.

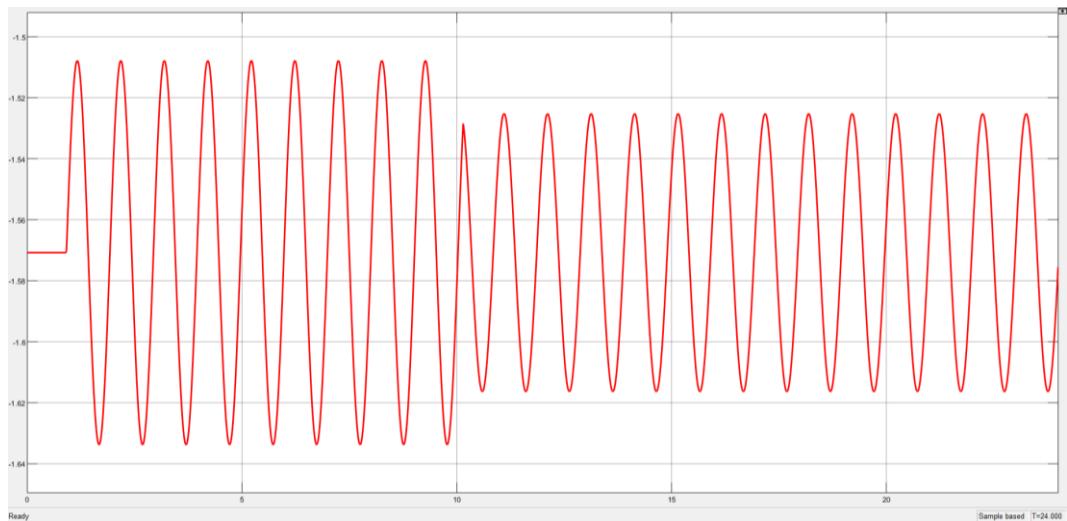


Figura 74: Gráfica de la oscilación de la carga en la simulación de Simscape. Sin filtro ZVD. Elaboración propia.

En esta gráfica también se puede comprobar si el cálculo previamente realizado de la frecuencia natural de oscilación está correcto. Como se puede comprobar el cálculo del período teóricamente en la ecuación (45) y en la simulación coinciden.

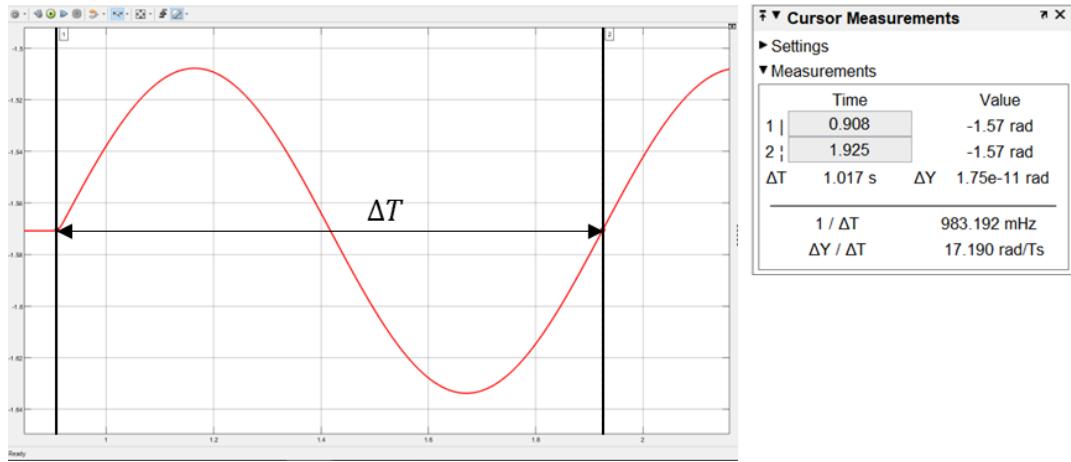


Figura 75: Comprobación experimental de la frecuencia natural del sistema a partir de la gráfica de la oscilación en la simulación de Simscape. Elaboración propia.

12.2 Resultados con filtro

12.2.1 Simulación

En esta simulación con el filtro de retraso temporal se puede apreciar una considerable disminución de la oscilación cuando se aplica el filtro completo (pasado $Ts3$). La trayectoria dibujada que sigue la carga es prácticamente plana como se ve en la imagen. Es mucho más apreciable en el video anteriormente adjuntado (11.5).

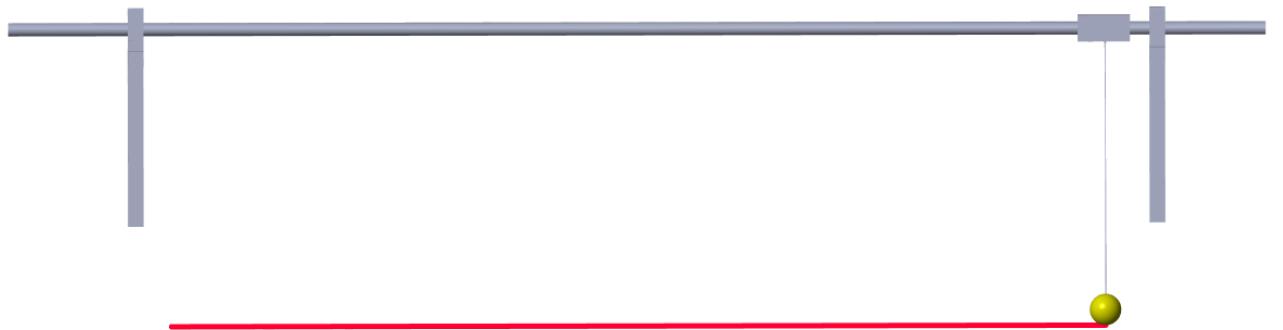


Figura 76: Trayectoria que sigue el punto inferior de la esfera aplicando el filtro ZVD. Elaboración propia en el programa de simulación Simscape Multibody.

12.2.2 Gráficas

Para confirmar que el filtro funciona se comparan las gráficas que se muestran en los “scope” anteriormente citados, como antes la primera que se muestra es la entrada, dado que el filtro se aplica en el microcontrolador la entrada ya se ve modificada con respecto a la entrada sin filtro. Como antes en el eje de abscisas se representa el tiempo en milisegundos, y en el eje de ordenadas el ciclo de trabajo de la señal PWM (valor entre 0 y 255).

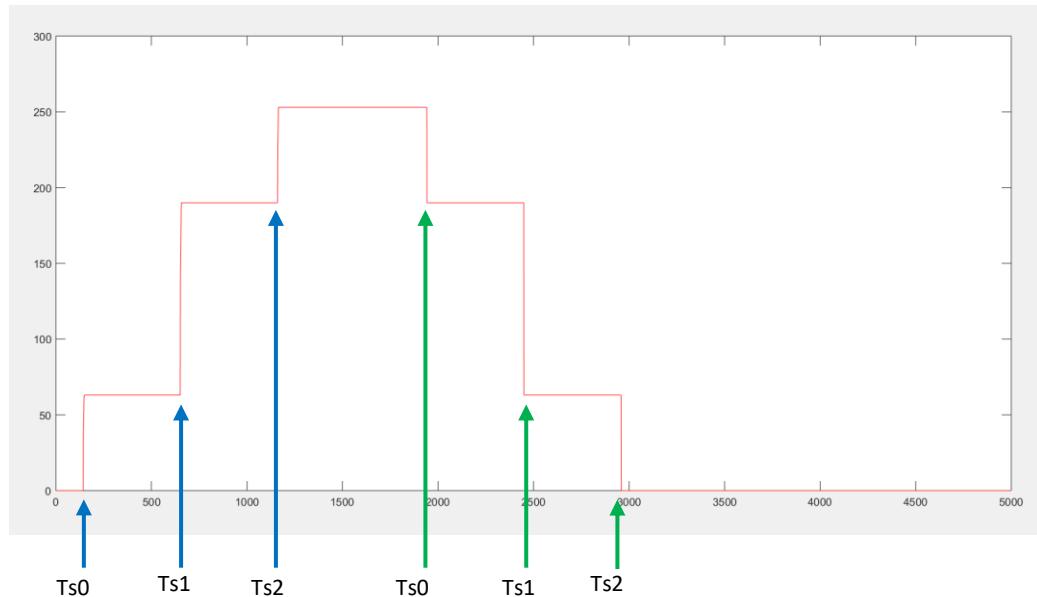


Figura 77: Gráfica de la señal de entrada que recibe Matlab del microcontrolador. Con filtro ZVD. Elaboración propia.

Se puede apreciar los instantes en los que se aplica el filtro que se ha calculado e implementado en el microcontrolador. Las flechas azules para el arranque y las flechas verdes para la parada.

La posición del carro se representa en la siguiente gráfica, se observa como la curva se suaviza en el arranque y en la parada siendo más progresivas. Los instantes en los que cambia la pendiente de la curva coinciden como antes con los tiempos en los que se aplica el filtro ZVD.

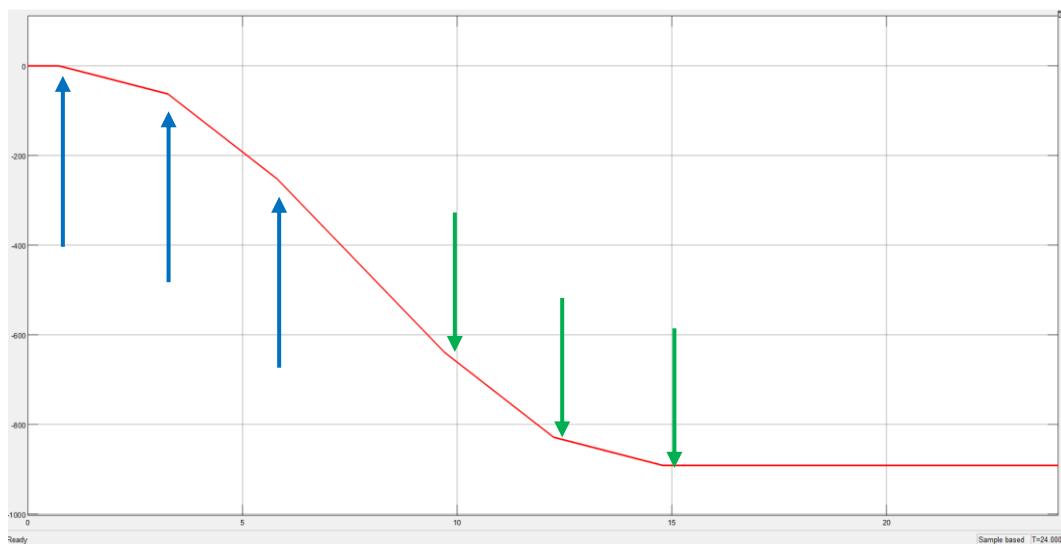


Figura 78: Grafica de la posición del carro en la simulación de Simscape. Con filtro ZVD. Elaboración propia.

De nuevo la gráfica más representativa y la que más interesa es la del ángulo de la carga con respecto a la vertical. Como antes en el ángulo en el eje de ordenadas se mide en radianes, en este caso al comenzar de una manera más progresiva la amplitud de la oscilación ya se reduce en gran medida.

La amplitud después del primer impulso:

$$\alpha = 0,031 \text{ rad}$$

$$\alpha = 1,78^\circ$$

Esto supone que la amplitud de la oscilación máxima que alcanza la carga se disminuye a un 24,85% de la oscilación sin aplicar el filtro de retraso temporal.

Si se espera a que se aplique el filtro por completo, es decir que el carro adquiera la velocidad de consigna, después de $Ts2$ se observa una reducción aun mayor de la amplitud de oscilación.

La amplitud después del tercer impulso:

$$\alpha = 0,0016 \text{ rad}$$

$$\alpha = 0,0916^\circ$$

Una vez aplicado el filtro, la amplitud de oscilación de la carga se reduce al **1,2%** de la amplitud de fechan la oscilación sin aplicar el filtro de retraso temporal. Este 1,2% de oscilación se produce por la falta de precisión, para corregirlo se podría aumentar el número de estados en el PWM o incluso aumentar la precisión de los instantes en los que se aplica el filtro.

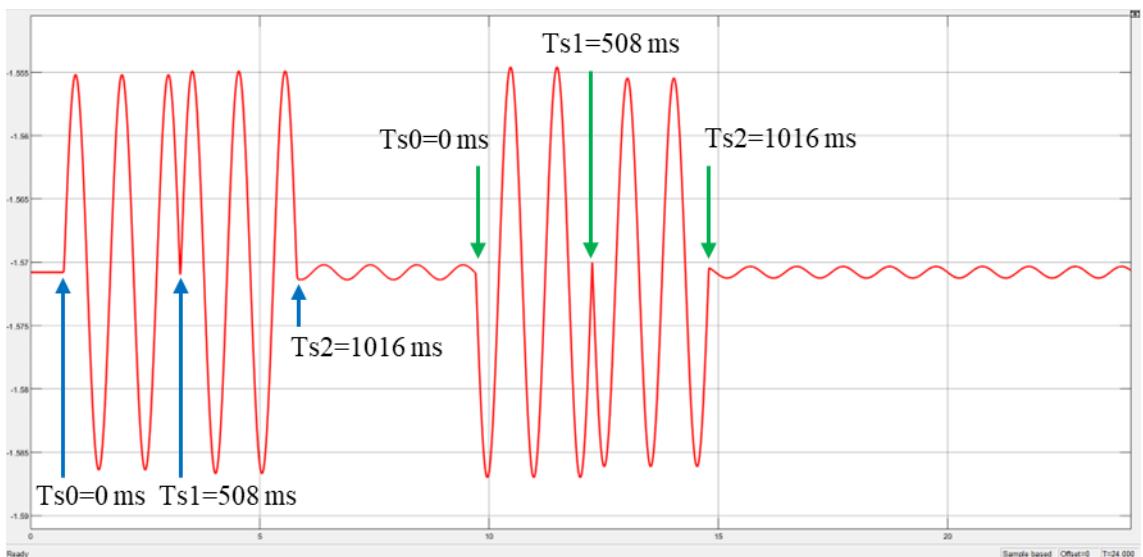


Figura 79: Gráfica de la oscilación de la carga en la simulación de Simscape. Con filtro ZVD. Elaboración propia.

12.3 Retraso temporal

Finalmente se va a valorar el inconveniente que presenta el input shaping, como se ha explicado anteriormente el precio a cambio de la reducción de la oscilación indeseada, es un retraso temporal entre la velocidad de consigna y la velocidad real, esto supone que el tiempo necesario para recorrer una distancia sea mayor.

Se comparan los tiempos que requiere el sistema sin filtro y el sistema filtrado para recorrer la distancia máxima en nuestra simulación (920mm).

Calculando con las diferentes velocidades en los instantes donde se aplica el filtro, se obtiene lo que tarda en recorrer los 920mm y así se pueden comparar los tiempos.

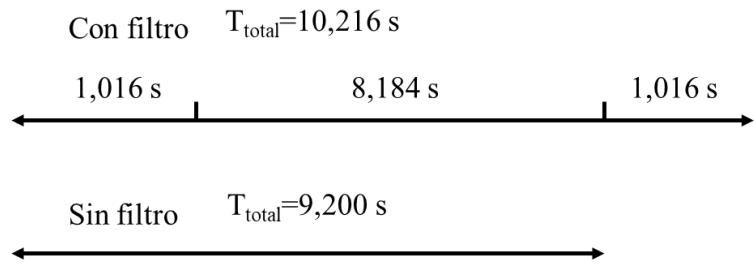


Figura 80: Comparación del tiempo requerido para el sistema con filtro y sin filtro.
Elaboración propia.

El tiempo que se aumenta para recorrer los 920mm, entre el filtro de arranque y el de parada son 1,016 s.

12.4 Resumen de los resultados

	Amplitud máxima	Amplitud durante el movimiento	Tiempo requerido
Sin filtro	7,162°	7,162°	9,200 s
Con filtro de retraso temporal	1,78°	0,0916°	10,216s

Tabla 8: Resultados de la simulación.

13 CONCLUSIONES Y FUTUROS PASOS

Se han cumplido los objetivos del trabajo, dado que se ha implementado con éxito un filtro de retraso temporal en un microcontrolador de 32 bits, validando los resultados con una simulación. La finalidad del filtro de retraso temporal es reducir la oscilación de la carga y se ha comprobado que funciona correctamente con un error prácticamente nulo.

Se han estudiado los diferentes componentes, valorando distintas alternativas para que el diseño sea fácilmente implementable en los procesos industriales que lo requieran, de una manera económica.

La metodología aplicada ha permitido simplificar el proceso para obtener el mejor resultado. Se han usado los recursos que permite el microcontrolador, procurando que su programación sea eficiente. Esta implementación permite adaptarlo a cualquier grúa dado que solo hay que cambiar los valores del filtro para la frecuencia de oscilación requerida.

La simulación ha permitido comprobar los resultados variando el diseño según se necesitaba, es un método idóneo para poder realizar las pruebas que se requieran hasta conseguir los resultados solicitados para posteriormente implementarlo en la vida real.

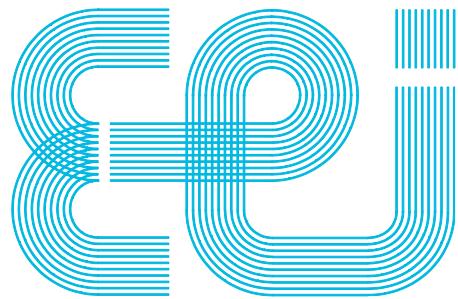
Para continuar con este proyecto los siguientes pasos consistirían en ampliar los distintos modos de implementación, y aplicarlo en otros diseños mecánicos que puedan mejorar su respuesta con un filtro de retraso temporal sin necesidad de ser grúas. Se podrían reducir las vibraciones de vehículos u otros objetos móviles. Otro avance importante es valorar los distintos tipos de filtros posibles dependiendo de la robustez necesaria para el caso en cuestión.

Una vez estudiados los sistemas de implementación se podría diseñar una PCB compacta a la que se conecte la consigna de entrada y la actuación sobre el motor, con ello se conseguiría implementar un filtro de retraso temporal de manera sencilla y económica dado que no se requiere ningún cambio físico del sistema mecánico.

14 BIBLIOGRAFÍA

- [1] N. W. Y.-W. Lee, "New electrical corrective network". 1931.
- [2] D. J. G. J. F. Calvert, "Signal component control," *Transactions of the American Institute of Electrical Engineers*, vol. 71, pp. 339-343, 1952.
- [3] O. J. M. SMITH, *Posicast Control of Damped Oscillatory Systems*, Urbana, 1957.
- [4] W. P. S. K. A. P. Neil C. Singer, "Shaping command inputs to minimize unwanted dynamics". Estados Unidos Patent 4916635, 10 Abril 1990.
- [5] J. L. J. Y. Kim, *Introduction to Dynamics and Control of Flexible Structures*, Washington, DC: AIAA, 1993.
- [6] Cypress, «Cypress,» [En línea]. Available: <https://www.infineon.com/cms/en/product/evaluation-boards/cy8ckit-037/>.
- [7] T. I. Incorporated, «Texas Instruments,» [En línea]. Available: <https://www.ti.com/product/L293D>.
- [8] R. PRO, «RS PRO,» [En línea]. Available: <https://es.rs-online.com/web/p/motores-dc/2588293>.
- [9] G. P. Lourido, «Código de implementación de un filtro de retraso temporal en un microcontrolador.,» 2022.
- [10] C.-G. Kang, "Impulse vectors for input-shaping control: A mathematical tool to design and analyze input shapers," *IEEE Control Systems Magazine*, vol. 39, no. 10.1109, pp. 40-55, 2019.
- [11] I. Torres-Rodríguez, P. Huerta-González y J. Rodríguez-Rivas, «Control de movimiento de una grúa viajera utilizando el control vectorial método indirecto de un motor de inducción,» *Científica*, vol. 14, nº 1, pp. 41-51, 2010.
- [12] M. T. a. R. D. Mustefa Jibril, "Design and Control of a Gantry Crane System with Limited Payload Angle using Robust and State Feedback Controllers," *Journal of Engineering and Applied Sciences*, vol. 15, no. 24, pp. 3770-3777, 2020.
- [13] S. C. P. C. Joaquim M. Veciana, "Minimizing Residual Vibrations for Non-Zero Initial States: Application to an Emergency Stop of a Crane," Department of Mechanical Engineering, Universitat Politècnica de Catalunya, Barcelona.
- [14] P. Šarafin, «Input Shaping,» *Information Sciences and Technologies*, vol. 10, nº 1, pp. 1-13, 2018.
- [15] T. J. Rodgers, «Infineon Technologies: Semiconductor & System Solutions,» Infineon, 2002. [En línea]. Available: <https://www.infineon.com/>. [Último acceso: 23 04 2022].
- [16] I. Anaheim Automation, «Anaheim Automation - Your source for Stepper Motor,» Anaheim Automation, 2021. [En línea]. Available: <https://www.anheimautomation.com/>. [Último acceso: 23 04 2022].
- [17] S. Fodor, "Towards Semi-Automation of Forestry Automated Trajectory Planning and Active Vibration," *Robotics and Control Group. Dissertations*, no. 9, 2017.
- [18] J. Group, «Jaso Industrial Cranes,» [En línea]. Available: <https://jasoindustrial.com/es-es/gruas-industriales/porticos/>.

- [19] I. T. AG, «Infineon,» [En línea]. Available: <https://www.infineon.com/cms/en/product/microcontroller/32-bit-psoc-arm-cortex-microcontroller/32-bit-psoc-5-lp-arm-cortex-m3/>.
- [20] Cypress, «Datasheet PSoC 5LP,» [En línea]. Available: https://www.mouser.es/pdfdocs/PSoC_5LP_CY8C56LP_Family_Datasheet_Programmable_System-on-chip_PSoC_Datasheet-2.pdf.
- [21] GeekFactory, «GeekFactory,» [En línea]. Available: <https://www.geekfactory.mx/tienda/modulos/motores/modulo-l298n-puente-h-driver-motores/>.
- [22] GeekFactory, «Electgpl,» [En línea]. Available: <http://electgpl.blogspot.com/2012/03/>.
- [23] Autec, «Autec,» [En línea]. Available: https://www.autecsafety.com/es/product/2-serie_air/82-curve_ajq.html.
- [24] «ipowerelectronics,» [En línea]. Available: <https://ipowerelectronics.com/movimiento/423-ky-023-modulo-de-joystick-de-ejes-x-y-.html>.
- [25] KUKA, «KUKA,» [En línea]. Available: <https://www.kuka.com/es-es/productos-servicios/sistemas-de-robot/robot-industrial/robot-de-paletizado>.
- [26] R. Components, «RS Components,» [En línea]. Available: <https://es.rs-online.com/web/>.
- [27] Cypress, «Cypress,» [En línea]. Available: <https://es.rs-online.com/web/p/kits-desarrollo-de-alimentacion-motores-y-robots/1244182>.



Escuela de Ingeniería Industrial

TRABAJO FIN DE GRADO

Implementación del filtrado de retraso temporal en tiempo real para un microcontrolador de 32 bits

Grado en Ingeniería en Electrónica Industrial y Automática

Documento

Anexo I: Código

UniversidadeVigo

ÍNDICE

Índice.....	2
1 Código Matlab para comparar la robustez de los filtros	3
2 Código Matlab procesado de datos	4
3 Código PSoC Creator.....	5
3.1 main.c	5

1 CÓDIGO MATLAB PARA COMPARAR LA ROBUSTEZ DE LOS FILTROS

```
A3=0; %Impulso numero 3
t3=0; %Instante de impulso numero 3
j=0; % Variable para ir rellenando los vectores

Datos=zeros(1000,1); % Vector de los datos ZV
tiempo=zeros(1000,1); % Vector de frecuencias para ZV
Datos2=zeros(1000,1); % Vector de los datos para ZVD
tiempo2=zeros(1000,1); % Vector de frecuencias para ZVD

I=1; % Valor calculado de I
A1=I/(1+I); % Cálculo del impulso 1
A2=1-A1; % Cálculo del impulso 2
t1=0; % Instante del impulso 1
w=2; % Variable de la frecuencia que iremos aumentando para llenar el vector de datos
t2=0.508; % Instante del impulso 2
while w<12
    C=(A1*exp(0.1*w*t1)*cos(w*sqrt(1-0.01)*t1))+(A2*exp(0.1*w*t2)*cos(w*sqrt(1-
0.01)*t2))+(A3*exp(0.1*w*t3)*cos(w*sqrt(1-0.01)*t3)); % Para el cálculo de la energía residual
    D=(A1*exp(0.1*w*t1)*sin(w*sqrt(1-0.01)*t1))+(A2*exp(0.1*w*t2)*sin(w*sqrt(1-
0.01)*t2))+(A3*exp(0.1*w*t3)*sin(w*sqrt(1-0.01)*t3)); % Para el cálculo de la energía residual
    V=(exp(-0.1*w*t2)*sqrt(C^2+D^2)); % Para el cálculo de la energía residual
    w=w+0.01;
    j=j+1;
    Datos(j)=V; % rellenamos el vector de datos ZV
    tiempo(j)=w; % rellenamos el vector de frecuencias
end
hold on
plot(tiempo,Datos,'r') % graficamos

% Ahora calculamos lo mismo para el filtro de retraso temporal ZVD
A1=0.25; %Impulso numero 1
A2=0.5; %Impulso numero 2
A3=0.25; %Impulso numero 3
t1=0; %Instante de impulso numero 1
t2=0.508; %Instante de impulso numero 2
t3=1.017; %Instante de impulso numero 3
w=2; % Variable de la frecuencia que iremos aumentando para llenar el vector de datos
j=0; % Variable para ir rellenando los vectores

while w<12
    C=(A1*exp(0.1*w*t1)*cos(w*sqrt(1-0.01)*t1))+(A2*exp(0.1*w*t2)*cos(w*sqrt(1-
0.01)*t2))+(A3*exp(0.1*w*t3)*cos(w*sqrt(1-0.01)*t3)); % Para el cálculo de la energía residual
    D=(A1*exp(0.1*w*t1)*sin(w*sqrt(1-0.01)*t1))+(A2*exp(0.1*w*t2)*sin(w*sqrt(1-
0.01)*t2))+(A3*exp(0.1*w*t3)*sin(w*sqrt(1-0.01)*t3)); % Para el cálculo de la energía residual
    V=(exp(-0.1*w*t2)*sqrt(C^2+D^2)); % Para el cálculo de la energía residual
    w=w+0.01; % Aumentamos el valor de la frecuencia
    j=j+1;
    Datos2(j)=V; % rellenamos el vector de datos ZV
    tiempo2(j)=w; % rellenamos el vector de frecuencias
end
plot(tiempo2,Datos2,'b') % garficamos
hold off
```

2 CÓDIGO MATLAB PROCESADO DE DATOS

```
i=0;
m=2;
j=1;
tiempo=0;
t=1;
k=0;
delete(instrfind({'port'},{'COM5'})); % Borramos cualquier configuración del puerto anterior
puerto=serial("COM5"); % Configuramos el puerto por el que se reciben los datos
% Se definen las matrices que se utilizarán
PWM=zeros(1,5000);
entrada=zeros(1,5000);
posicion=zeros(1,25000);
final=zeros(25000,2);
puerto.BaudRate=115200; % Se define la velocidad de transmisión
fopen(puerto); % Se abre el puerto a partir de aquí se están recibiendo datos
% se almacenan los valores en un vector
while i<5000
    valor=fscanf(puerto,'%d');
    slope=valor*0.00039215686; % se hace una conversión para que estén entre 0 y 1
    i=i+1;
    entrada(i)=valor;
    PWM(i)=slope;
end
i=0;
%Almacenamos en un nuevo vector la posición
while j<5000
    i=i+1;
    if i==5
        i=0;
        j=j+1;
    end
    m=m+1;
    posicion(m)=posicion(m-1)-PWM(j); % Cálculo de la posición
    if posicion(m)<(-920.105) % Restricciones de la posición del carro
        posicion(m)=-920.105;
    end
end
%Rellenamos una matriz con tiempo y posición
while k<25000
    k=k+1;
    if t==2
        final(k,t)=posicion(k);
        t=1;
    end
    if t==1
        final(k,t)=tiempo;
        t=2;
    end
    tiempo=tiempo+0.001;
end
% Esta matriz la puede importar simulink
```

3 CÓDIGO PSOC CREATOR

Este código solo incluye el main.c, la interrupción de convolución donde se aplica el filtro ha sido desarrollada por Gerardo Peláez Lourido.

3.1 main.c

```
#include "project.h"
#include <stdio.h>

int control=1;
int Filtro=0;
int paro=0;
int val;
int var=0;
uint32 indice_resto=0;
uint32 resto=0;
const float TAP_1 = 0.25;
const float TAP_2 = 0.50;
const float TAP_3 = 0.25;
#define buffer_size 15000
uint8 x[buffer_size];
uint8 state=0;
uint32 contajes=0;
uint8 y_n=0;
int valorh=0;
int valort=0;
int valorfinal=0;
int Filtroant=0;
char str[12];
int obtener_consigna(int valorh)
{
    if ((valorh>132) &&(valorh<256))
    {
        valorfinal=(valorh-128)*2;
    }
    if ((valorh<128) &&(valorh>=0))
    {
        valorfinal=(127-valorh)*2;
    }
    if ((valorh>120) &&(valorh<135))
    {
        valorfinal=0;
    }
    return valorfinal;
}

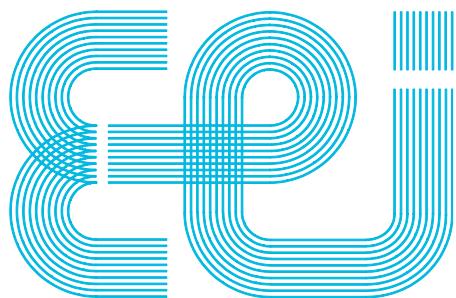
void clear_buffer(uint32 limit) {
    uint32 i;

    for (i=0; i<limit; i++) {
        x[i]= 0;
    }
}

CY_ISR(Int_Izquierda)
```

```
{  
    if (Filtro==0)  
    {  
        Filtro=1;  
    }  
    else  
    {  
        Filtro=0;  
    }  
    Pin_Izquierda_ClearInterrupt();  
    Int_Izquierda_Disable();  
}  
  
int main(void)  
{  
    PWM_Start();  
    ADC_Start();  
    ADC_StartConvert();  
    Clock_1_Start();  
    Clock_2_Start();  
    Clock_3_Start();  
    Clock_4_Start();  
    Int_Izquierda_StartEx(Int_Izquierda);  
    UART_Start();  
    Int_Izquierda_ClearPending();  
    CyGlobalIntEnable;  
  
    for(;;)  
    {  
        val = ADC_GetResult16();  
        valort = obtener_consigna(val);  
  
        if (Filtro==0 && Filtroant==1) //Pasamos a sin filtro  
        {  
            convolucion_Stop();  
            Filtroant=Filtro;  
            contajes=0;  
            state=0;  
            LED_Write(0);  
            PWM_WriteCompare(valort);  
            sprintf(str, "%d \n \r", valort);  
            UART_PutString(str);  
        }  
        if (Filtro==1 && Filtroant==0) //Pasamos a con filtro  
        {  
            LED_Write(1);  
            clear_buffer(buffer_size);  
            state=0;  
            contajes=0;  
            convolucion_Start();  
            Filtroant=Filtro;  
        }  
    }  
}
```

```
if (Filtro==0 && Filtroant==0) //Sin filtro
{
    LED_Write(0);
    if ((val>135) &&(val<256))
    {
        dir_derecha_Write(1);
        dir_izquierda_Write(0);
        PWM_WriteCompare(valort);
        sprintf(str, "%d \n \r",valort);
        UART_PutString(str);
    }
    if ((val<125) &&(val>=0))
    {
        dir_derecha_Write(0);
        dir_izquierda_Write(1);
        PWM_WriteCompare(valort);
        sprintf(str, "%d \n \r",valort);
        UART_PutString(str);
    }
    if ((val>120) &&(val<135))
    {
        PWM_WriteCompare(0);
        sprintf(str, "%d \n \r",0);
        UART_PutString(str);
    }
}
CyDelay(5);
Int_Izquierda_Enable();
Reset_PWM_Write(1);
Reset_PWM_Write(0);
}
```



Escuela de Ingeniería Industrial

TRABAJO FIN DE GRADO

Implementación del filtrado de retraso temporal en tiempo real para un microcontrolador de 32 bits

Grado en Ingeniería en Electrónica Industrial y Automática

Documento

Anexo II: Hojas de Características

Universidade de Vigo

ÍNDICE

Índice.....	2
1 Motor CC con reductora RS PRO.....	3
2 Controlador L298.....	4



Datasheet

RS PRO Geared DC Geared Motor

Stock No: 2588293, 2588388



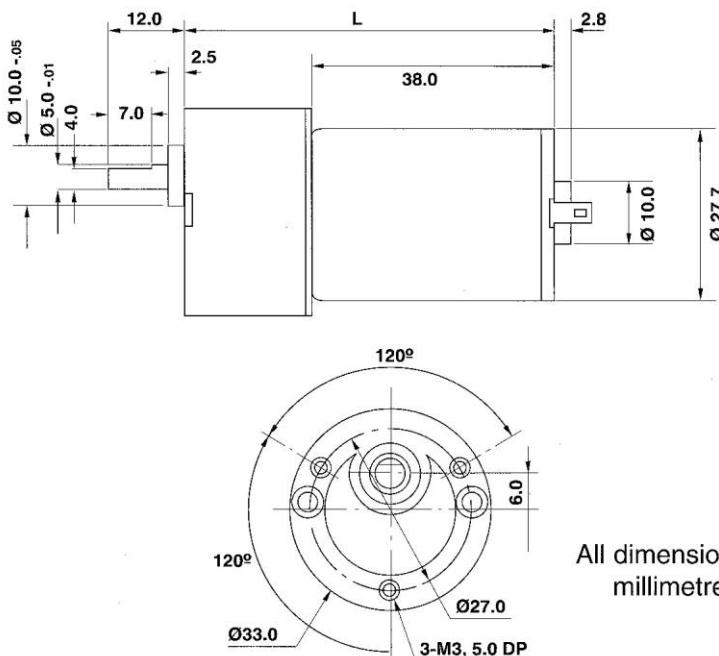
ENGLISH

The Igarashi IG33 range is designed for low cost, high volume applications such as vending machines, magnetic card readers, printer mechanisms, actuators and all types of coin control systems.

The IG33 is fitted with an all metal gearhead with sintered iron gears running on hardened steel shafts. The output shaft runs in sintered bronze bearings.

The permanent magnet motors are wound for different speeds for each voltage to give a wide choice of final speeds when matched to the seven gearheads.

Weight 115 - 140 gms according to ratio
Max. radial load: 50N (8mm from mounting face)
Max. axial load: 33N (at shaft centre)
Max. radial play: 0.08mm
Max. axial play: 0.50mm



Type	Voltage(V)	Ratio	*Output Rotation	Length (L) (mm)	Maximum Torque (mNm)	Nominal No load speed rev/min	Nominal No load current mA
25882	24	20:1	CW	57.8	100	190	30
25883	24					5	30

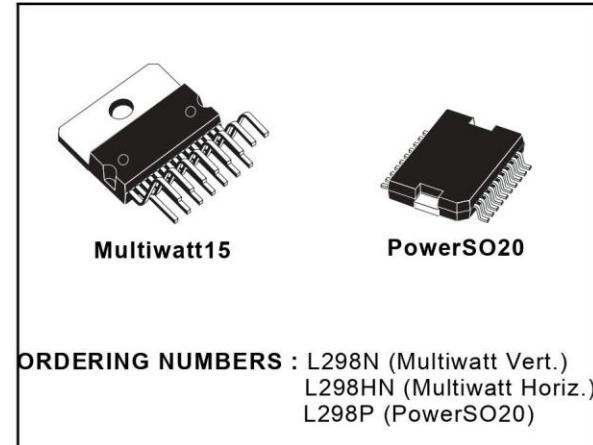
* The motors are bidirectional; the rotation listed is achieved with positive supply connected to +ve terminal on motor.

DUAL FULL-BRIDGE DRIVER

- OPERATING SUPPLY VOLTAGE UP TO 46 V
 - TOTAL DC CURRENT UP TO 4 A
 - LOW SATURATION VOLTAGE
 - OVERTEMP PROTECTION
 - LOGICAL "0" INPUT VOLTAGE UP TO 1.5 V
(HIGH NOISE IMMUNITY)

DESCRIPTION

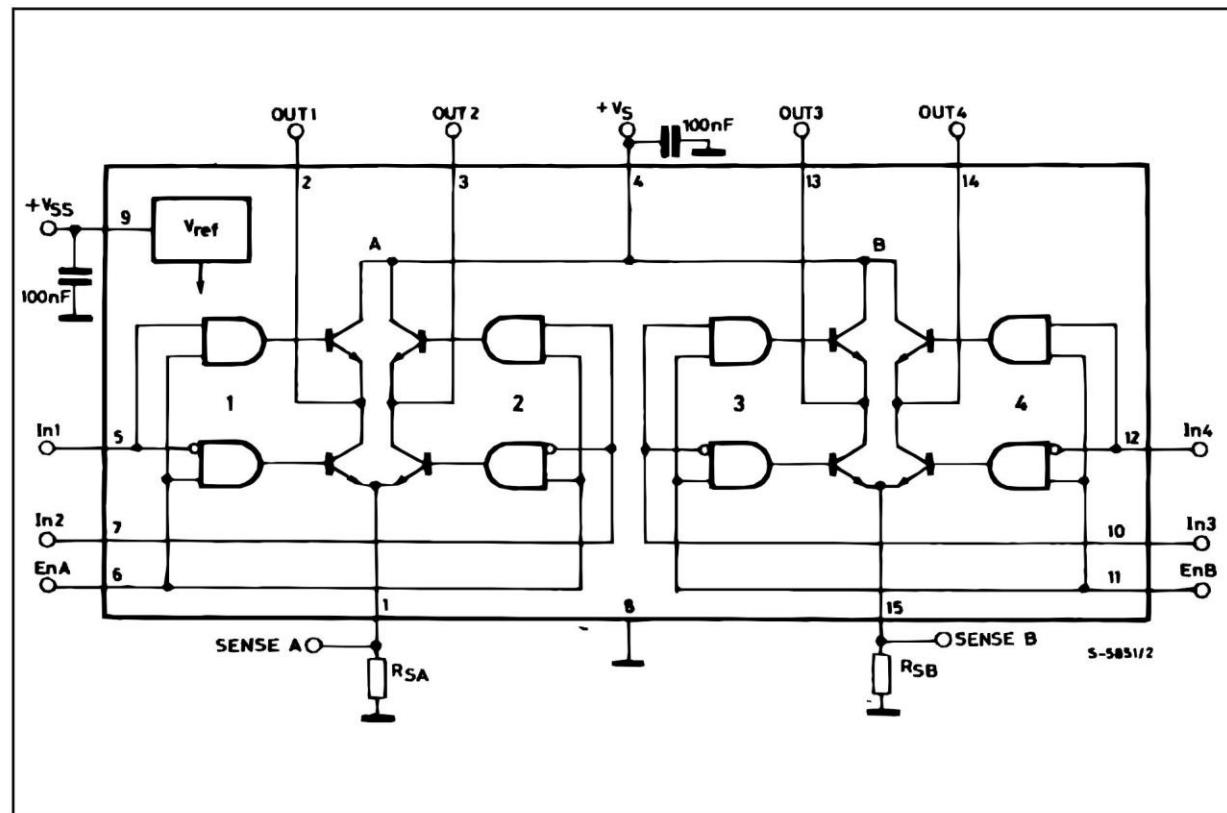
The L298 is an integrated monolithic circuit in a 15-lead Multiwatt and PowerSO20 packages. It is a high voltage, high current dual full-bridge driver designed to accept standard TTL logic levels and drive inductive loads such as relays, solenoids, DC and stepping motors. Two enable inputs are provided to enable or disable the device independently of the input signals. The emitters of the lower transistors of each bridge are connected together and the corresponding external terminal can be used for the con-



ORDERING NUMBERS : L298N (Multiwatt Vert.)
L298HN (Multiwatt Horiz.)
L298P (PowerSO20)

nection of an external sensing resistor. An additional supply input is provided so that the logic works at a lower voltage.

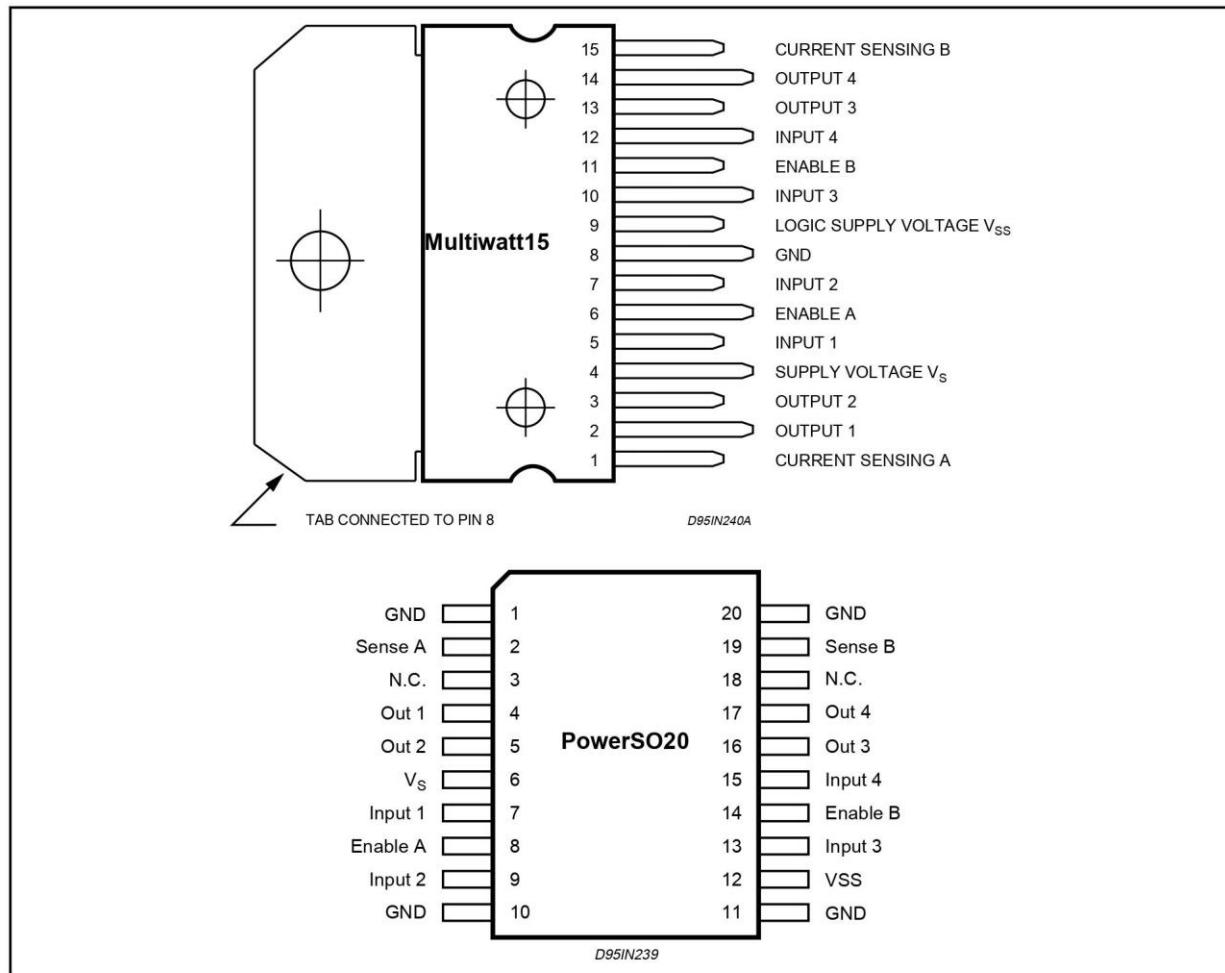
BLOCK DIAGRAM



ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Value	Unit
V_S	Power Supply	50	V
V_{SS}	Logic Supply Voltage	7	V
V_I, V_{en}	Input and Enable Voltage	-0.3 to 7	V
I_O	Peak Output Current (each Channel)		
	– Non Repetitive ($t = 100\mu s$)	3	A
	– Repetitive (80% on -20% off; $t_{on} = 10ms$)	2.5	A
	– DC Operation	2	A
V_{sens}	Sensing Voltage	-1 to 2.3	V
P_{tot}	Total Power Dissipation ($T_{case} = 75^\circ C$)	25	W
T_{op}	Junction Operating Temperature	-25 to 130	°C
T_{stg}, T_j	Storage and Junction Temperature	-40 to 150	°C

PIN CONNECTIONS (top view)



THERMAL DATA

Symbol	Parameter	PowerSO20	Multiwatt15	Unit
$R_{th\ j-case}$	Thermal Resistance Junction-case	Max.	–	3 $^\circ C/W$
$R_{th\ j-amb}$	Thermal Resistance Junction-ambient	Max.	13 (*)	35 $^\circ C/W$

(*) Mounted on aluminum substrate

PIN FUNCTIONS (refer to the block diagram)

MW.15	PowerSO	Name	Function
1;15	2;19	Sense A; Sense B	Between this pin and ground is connected the sense resistor to control the current of the load.
2;3	4;5	Out 1; Out 2	Outputs of the Bridge A; the current that flows through the load connected between these two pins is monitored at pin 1.
4	6	V _S	Supply Voltage for the Power Output Stages. A non-inductive 100nF capacitor must be connected between this pin and ground.
5;7	7;9	Input 1; Input 2	TTL Compatible Inputs of the Bridge A.
6;11	8;14	Enable A; Enable B	TTL Compatible Enable Input: the L state disables the bridge A (enable A) and/or the bridge B (enable B).
8	1,10,11,20	GND	Ground.
9	12	V _{SS}	Supply Voltage for the Logic Blocks. A100nF capacitor must be connected between this pin and ground.
10; 12	13;15	Input 3; Input 4	TTL Compatible Inputs of the Bridge B.
13; 14	16;17	Out 3; Out 4	Outputs of the Bridge B. The current that flows through the load connected between these two pins is monitored at pin 15.
–	3;18	N.C.	Not Connected

ELECTRICAL CHARACTERISTICS (V_S = 42V; V_{SS} = 5V, T_j = 25°C; unless otherwise specified)

Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit
V _S	Supply Voltage (pin 4)	Operative Condition	V _{IH} +2.5		46	V
V _{SS}	Logic Supply Voltage (pin 9)		4.5	5	7	V
I _S	Quiescent Supply Current (pin 4)	V _{en} = H; I _L = 0 V _i = L V _i = H		13 50	22 70	mA mA
		V _{en} = L V _i = X			4	mA
I _{SS}	Quiescent Current from V _{SS} (pin 9)	V _{en} = H; I _L = 0 V _i = L V _i = H		24 7	36 12	mA mA
		V _{en} = L V _i = X			6	mA
V _{iL}	Input Low Voltage (pins 5, 7, 10, 12)		-0.3		1.5	V
V _{iH}	Input High Voltage (pins 5, 7, 10, 12)		2.3		V _{SS}	V
I _{IL}	Low Voltage Input Current (pins 5, 7, 10, 12)	V _i = L			-10	mA
I _{iH}	High Voltage Input Current (pins 5, 7, 10, 12)	V _i = H ≤ V _{SS} -0.6V		30	100	mA
V _{en} = L	Enable Low Voltage (pins 6, 11)		-0.3		1.5	V
V _{en} = H	Enable High Voltage (pins 6, 11)		2.3		V _{SS}	V
I _{en} = L	Low Voltage Enable Current (pins 6, 11)	V _{en} = L			-10	mA
I _{en} = H	High Voltage Enable Current (pins 6, 11)	V _{en} = H ≤ V _{SS} -0.6V		30	100	mA
V _{CEsat (H)}	Source Saturation Voltage	I _L = 1A I _L = 2A	0.95 2	1.35	1.7 2.7	V V
V _{CEsat (L)}	Sink Saturation Voltage	I _L = 1A (5) I _L = 2A (5)	0.85	1.2 1.7	1.6 2.3	V V
V _{CEsat}	Total Drop	I _L = 1A (5) I _L = 2A (5)	1.80		3.2 4.9	V V
V _{sens}	Sensing Voltage (pins 1, 15)		-1 (1)		2	V

ELECTRICAL CHARACTERISTICS (continued)

Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit
T ₁ (V _i)	Source Current Turn-off Delay	0.5 V _i to 0.9 I _L (2); (4)		1.5		μs
T ₂ (V _i)	Source Current Fall Time	0.9 I _L to 0.1 I _L (2); (4)		0.2		μs
T ₃ (V _i)	Source Current Turn-on Delay	0.5 V _i to 0.1 I _L (2); (4)		2		μs
T ₄ (V _i)	Source Current Rise Time	0.1 I _L to 0.9 I _L (2); (4)		0.7		μs
T ₅ (V _i)	Sink Current Turn-off Delay	0.5 V _i to 0.9 I _L (3); (4)		0.7		μs
T ₆ (V _i)	Sink Current Fall Time	0.9 I _L to 0.1 I _L (3); (4)		0.25		μs
T ₇ (V _i)	Sink Current Turn-on Delay	0.5 V _i to 0.9 I _L (3); (4)		1.6		μs
T ₈ (V _i)	Sink Current Rise Time	0.1 I _L to 0.9 I _L (3); (4)		0.2		μs
f _c (V _i)	Commutation Frequency	I _L = 2A		25	40	KHz
T ₁ (V _{en})	Source Current Turn-off Delay	0.5 V _{en} to 0.9 I _L (2); (4)		3		μs
T ₂ (V _{en})	Source Current Fall Time	0.9 I _L to 0.1 I _L (2); (4)		1		μs
T ₃ (V _{en})	Source Current Turn-on Delay	0.5 V _{en} to 0.1 I _L (2); (4)		0.3		μs
T ₄ (V _{en})	Source Current Rise Time	0.1 I _L to 0.9 I _L (2); (4)		0.4		μs
T ₅ (V _{en})	Sink Current Turn-off Delay	0.5 V _{en} to 0.9 I _L (3); (4)		2.2		μs
T ₆ (V _{en})	Sink Current Fall Time	0.9 I _L to 0.1 I _L (3); (4)		0.35		μs
T ₇ (V _{en})	Sink Current Turn-on Delay	0.5 V _{en} to 0.9 I _L (3); (4)		0.25		μs
T ₈ (V _{en})	Sink Current Rise Time	0.1 I _L to 0.9 I _L (3); (4)		0.1		μs

1) Sensing voltage can be -1 V for $t \leq 50$ μ sec; in steady state V_{sens} min ≥ -0.5 V.

2) See fig. 2.

3) See fig. 4.

4) The load must be a pure resistor.

Figure 1 : Typical Saturation Voltage vs. Output Current.

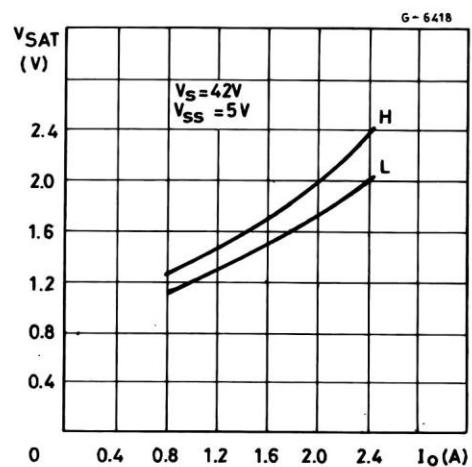
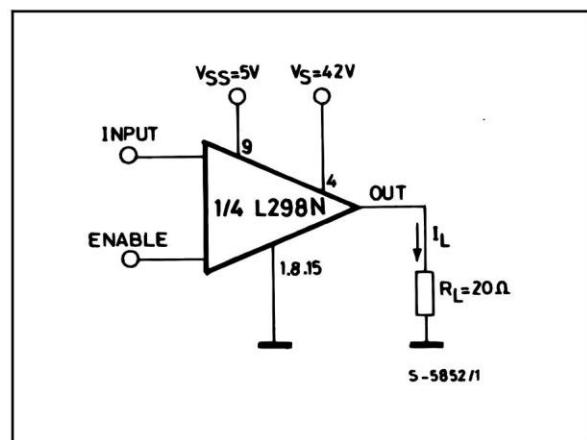


Figure 2 : Switching Times Test Circuits.



Note : For INPUT Switching, set EN = H
For ENABLE Switching, set IN = H

Figure 3 : Source Current Delay Times vs. Input or Enable Switching.

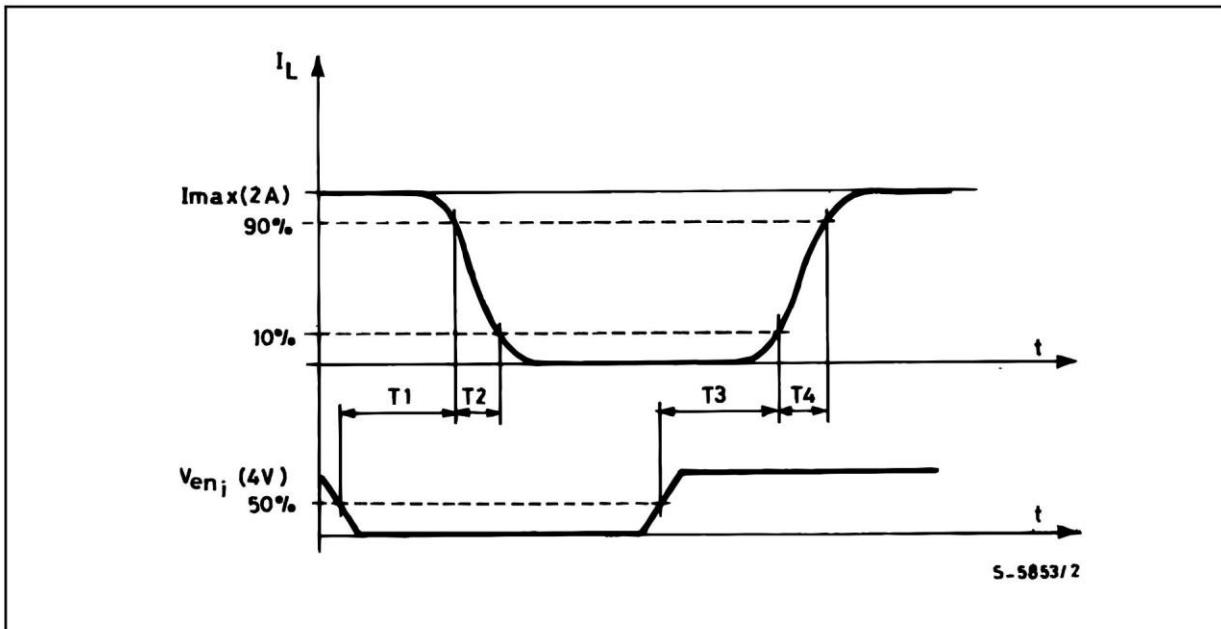
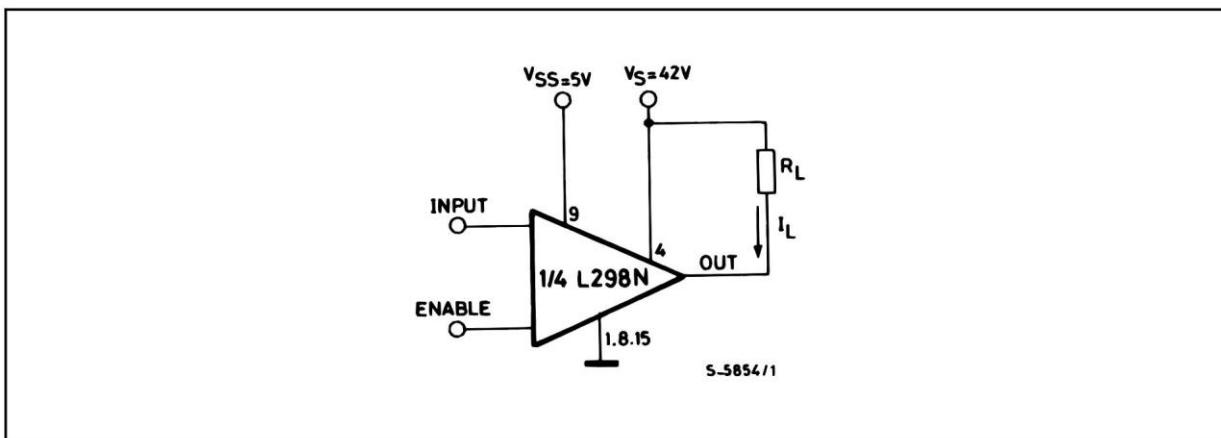


Figure 4 : Switching Times Test Circuits.



Note : For INPUT Switching, set EN = H
 For ENABLE Switching, set IN = L

Figure 5 : Sink Current Delay Times vs. Input 0 V Enable Switching.

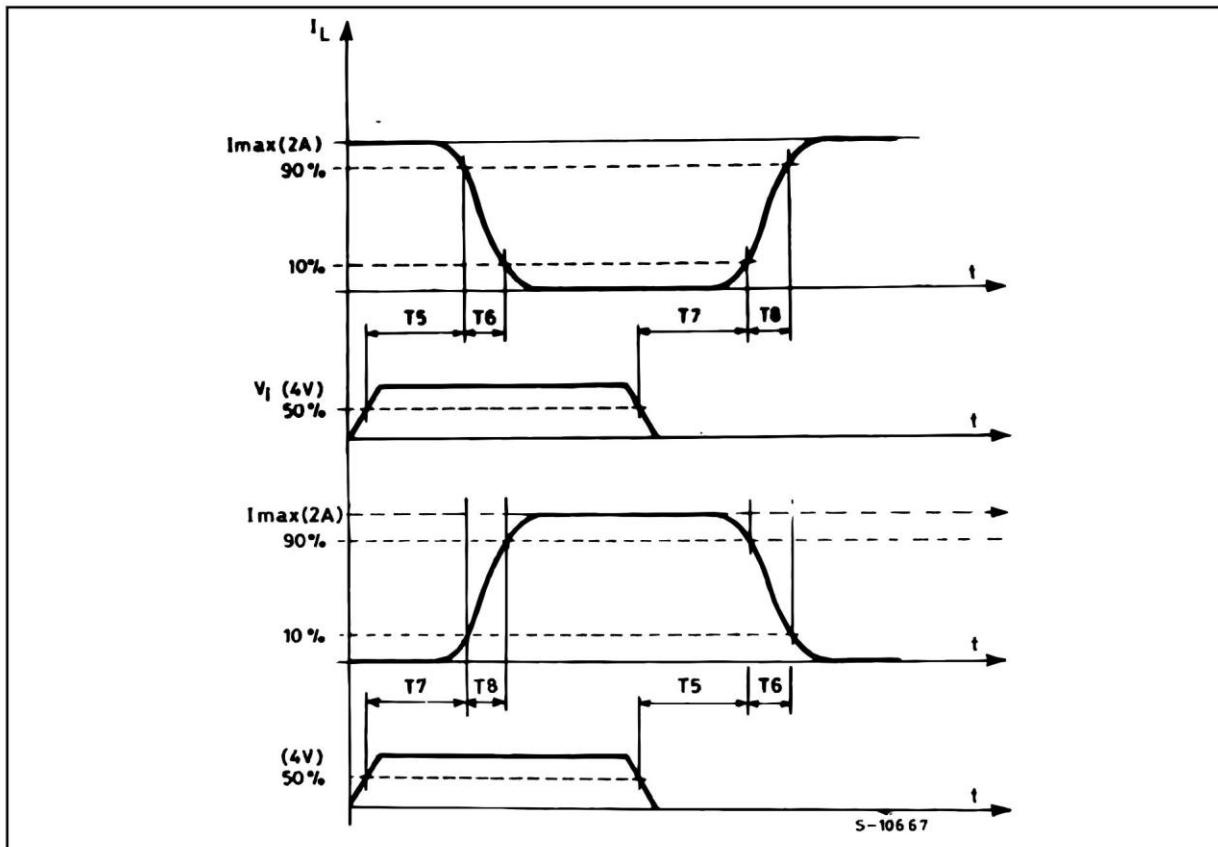


Figure 6 : Bidirectional DC Motor Control.

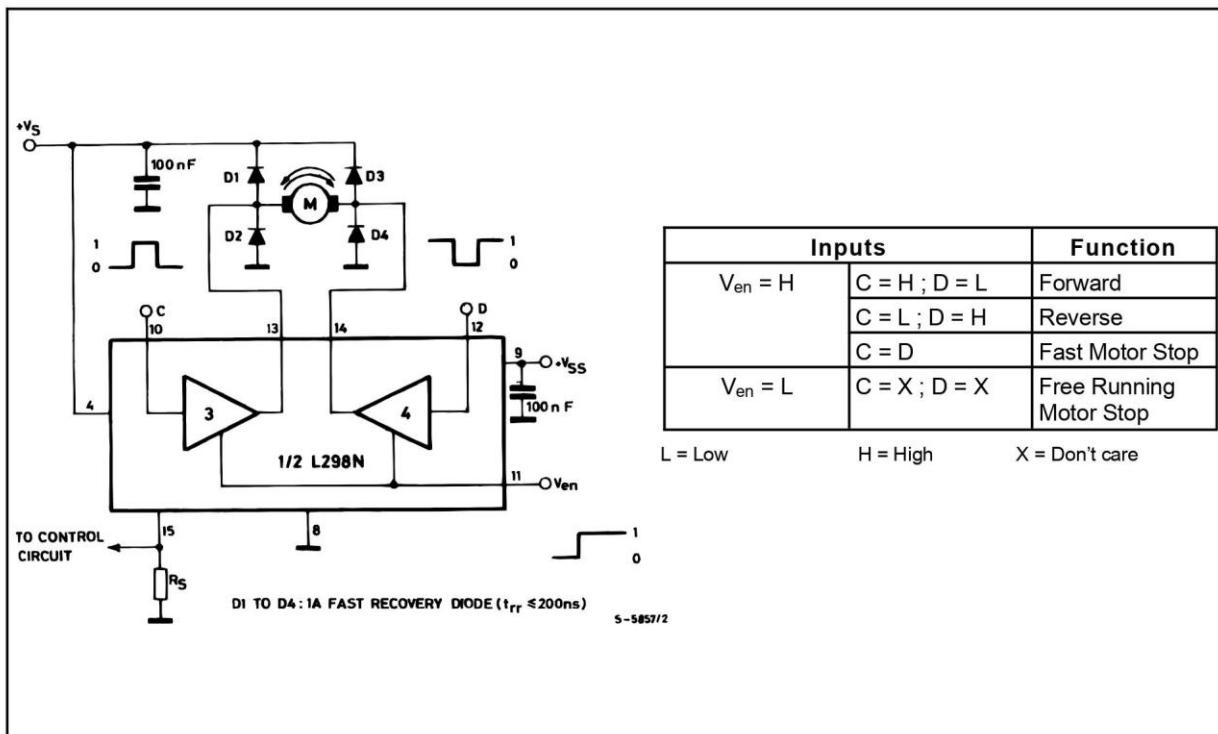
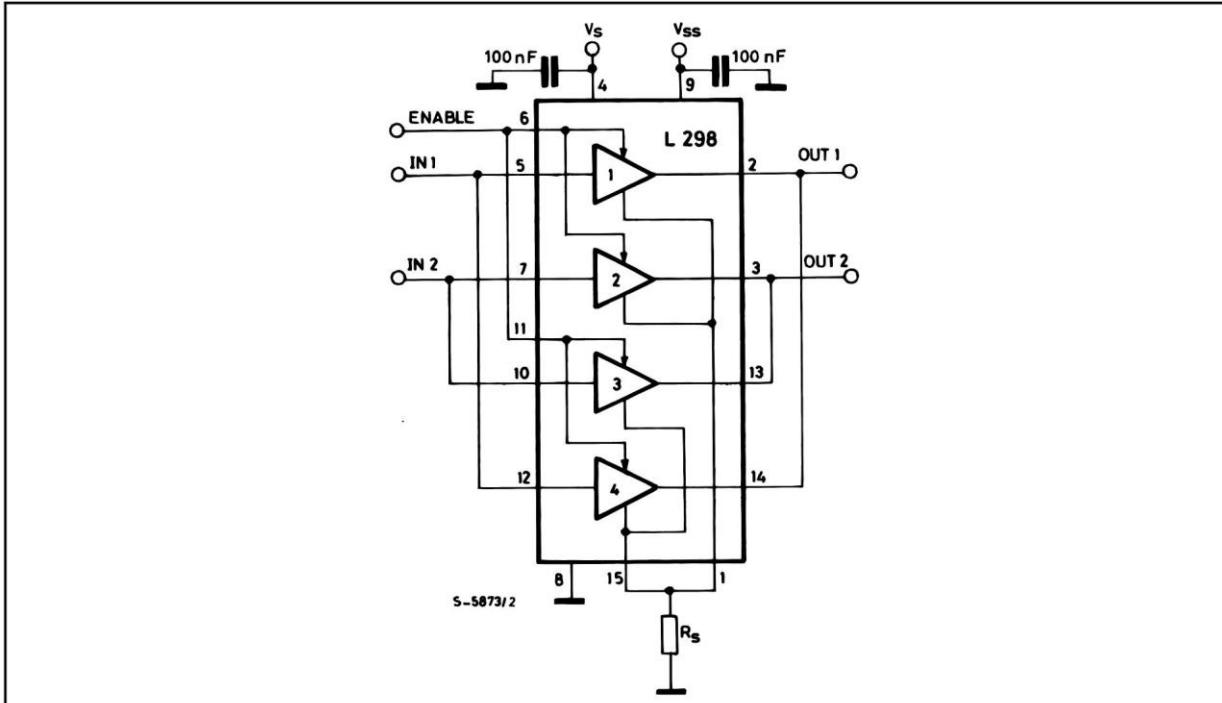


Figure 7 : For higher currents, outputs can be paralleled. Take care to parallel channel 1 with channel 4 and channel 2 with channel 3.



APPLICATION INFORMATION (Refer to the block diagram)

1.1. POWER OUTPUT STAGE

The L298 integrates two power output stages (A ; B). The power output stage is a bridge configuration and its outputs can drive an inductive load in common or differenzial mode, depending on the state of the inputs. The current that flows through the load comes out from the bridge at the sense output : an external resistor (R_{SA} ; R_{SB}) allows to detect the intensity of this current.

1.2. INPUT STAGE

Each bridge is driven by means of four gates the input of which are IN_1 ; IN_2 ; EN_A and IN_3 ; IN_4 ; EN_B . The IN inputs set the bridge state when The EN input is high ; a low state of the EN input inhibits the bridge. All the inputs are TTL compatible.

2. SUGGESTIONS

A non inductive capacitor, usually of 100 nF, must be foreseen between both V_s and V_{ss} , to ground, as near as possible to GND pin. When the large capacitor of the power supply is too far from the IC, a second smaller one must be foreseen near the L298.

The sense resistor, not of a wire wound type, must be grounded near the negative pole of V_s that must be near the GND pin of the I.C.

Each input must be connected to the source of the driving signals by means of a very short path.

Turn-On and Turn-Off : Before to Turn-ON the Supply Voltage and before to Turn it OFF, the Enable input must be driven to the Low state.

3. APPLICATIONS

Fig 6 shows a bidirectional DC motor control Schematic Diagram for which only one bridge is needed. The external bridge of diodes D1 to D4 is made by four fast recovery elements ($trr \leq 200$ nsec) that must be chosen of a VF as low as possible at the worst case of the load current.

The sense output voltage can be used to control the current amplitude by chopping the inputs, or to provide overcurrent protection by switching low the enable input.

The brake function (Fast motor stop) requires that the Absolute Maximum Rating of 2 Amps must never be overcome.

When the repetitive peak current needed from the load is higher than 2 Amps, a paralleled configuration can be chosen (See Fig.7).

An external bridge of diodes are required when inductive loads are driven and when the inputs of the IC are chopped ; Shottky diodes would be preferred.

This solution can drive until 3 Amps in DC operation and until 3.5 Amps of a repetitive peak current.

On Fig 8 it is shown the driving of a two phase bipolar stepper motor ; the needed signals to drive the inputs of the L298 are generated, in this example, from the IC L297.

Fig 9 shows an example of P.C.B. designed for the application of Fig 8.

Figure 8 : Two Phase Bipolar Stepper Motor Circuit.

This circuit drives bipolar stepper motors with winding currents up to 2 A. The diodes are fast 2 A types.

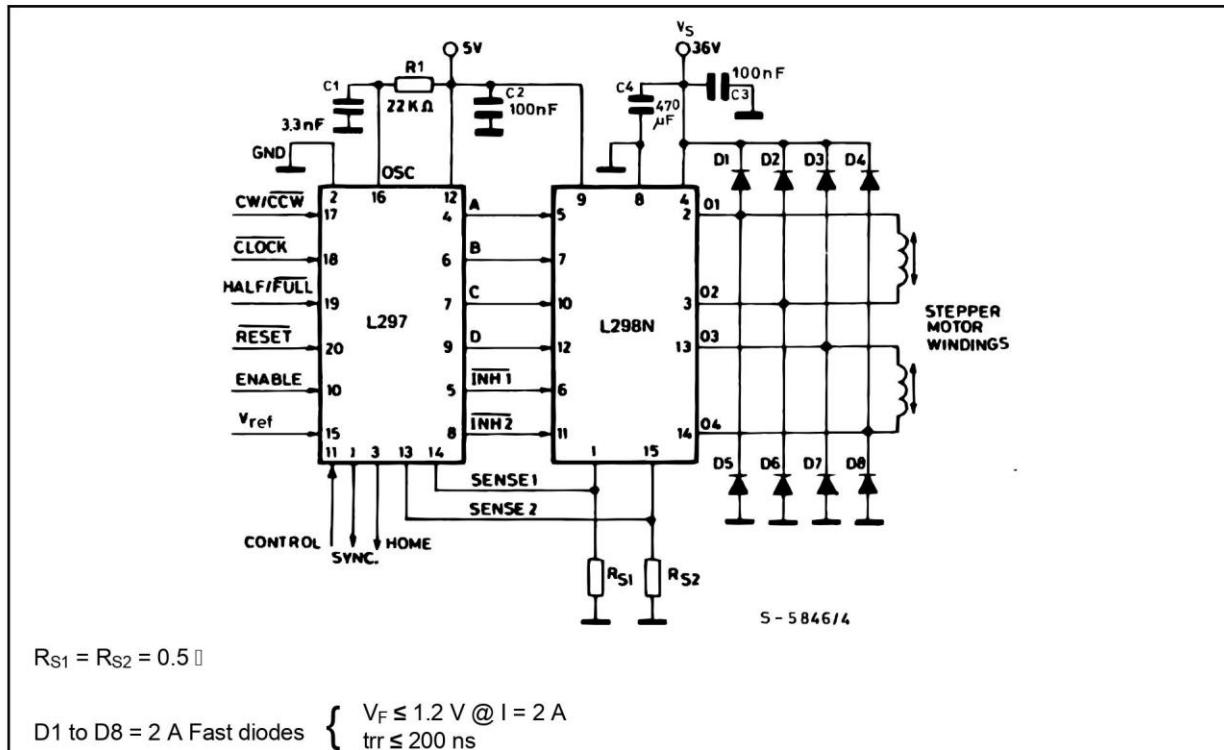


Figure 9 : Suggested Printed Circuit Board Layout for the Circuit of fig. 8 (1:1 scale).

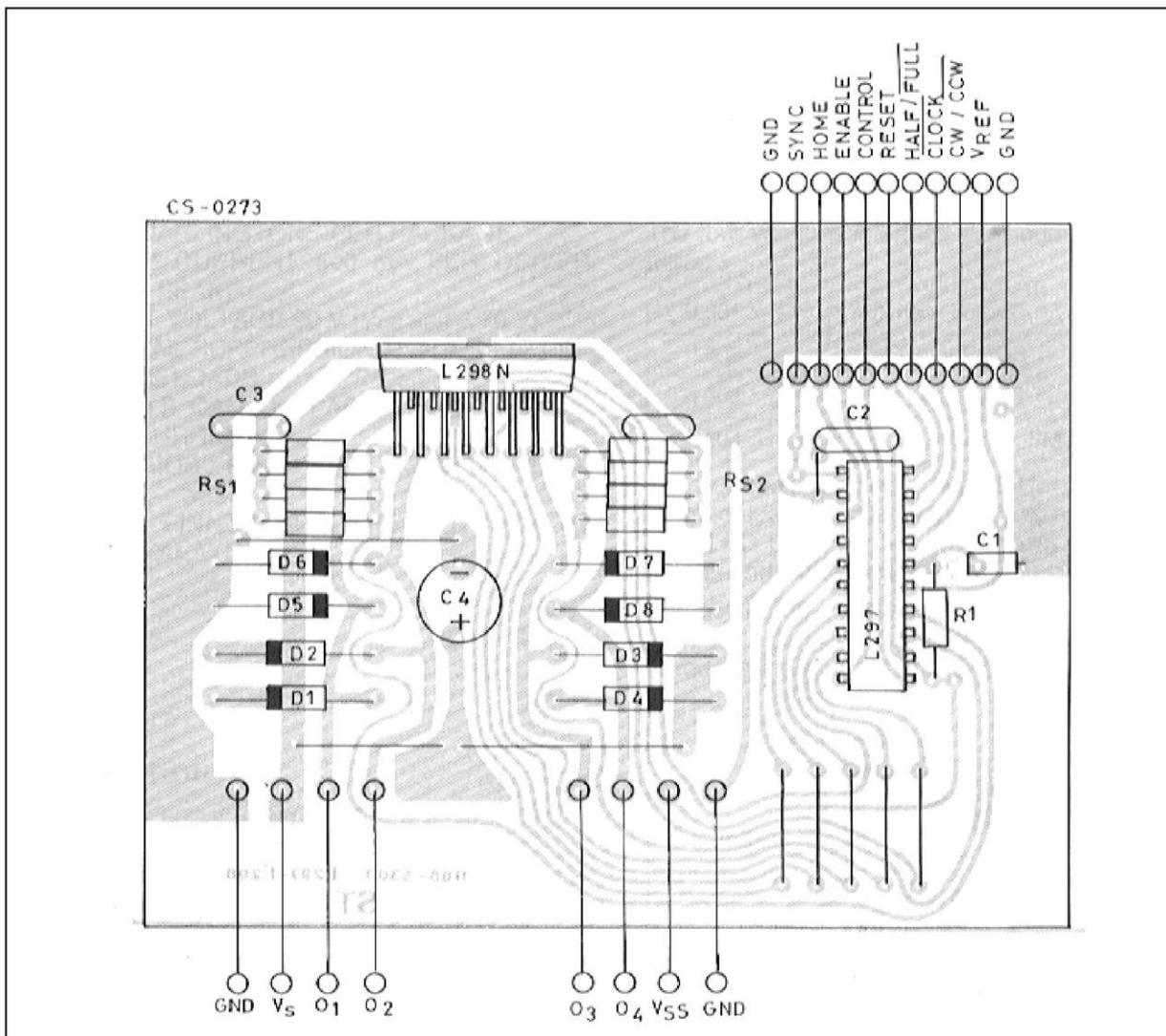
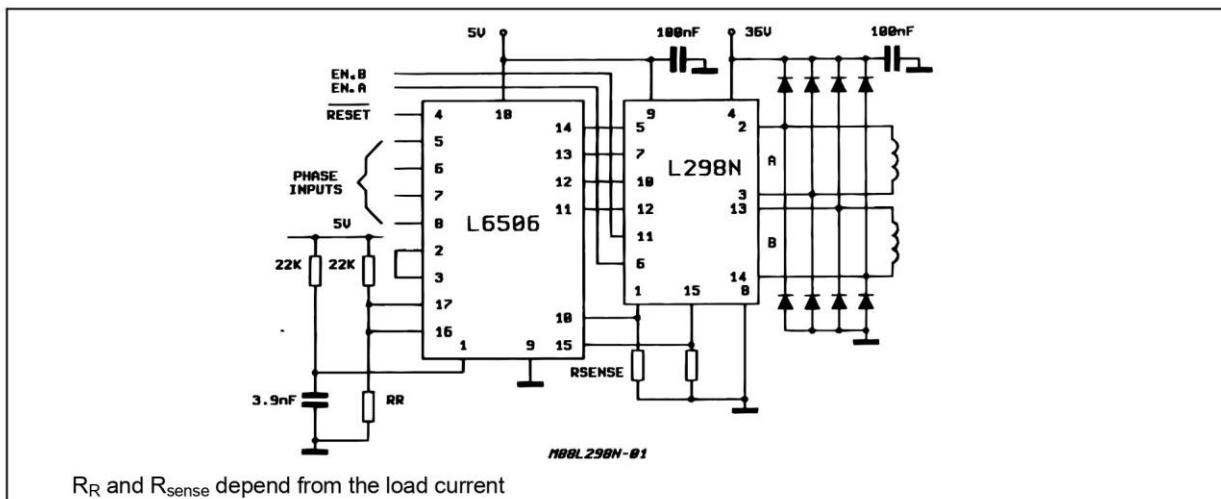
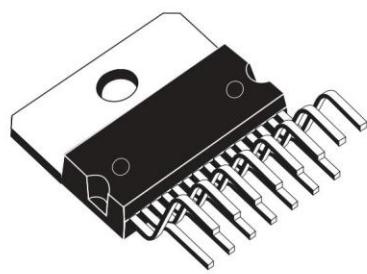


Figure 10 : Two Phase Bipolar Stepper Motor Control Circuit by Using the Current Controller L6506.

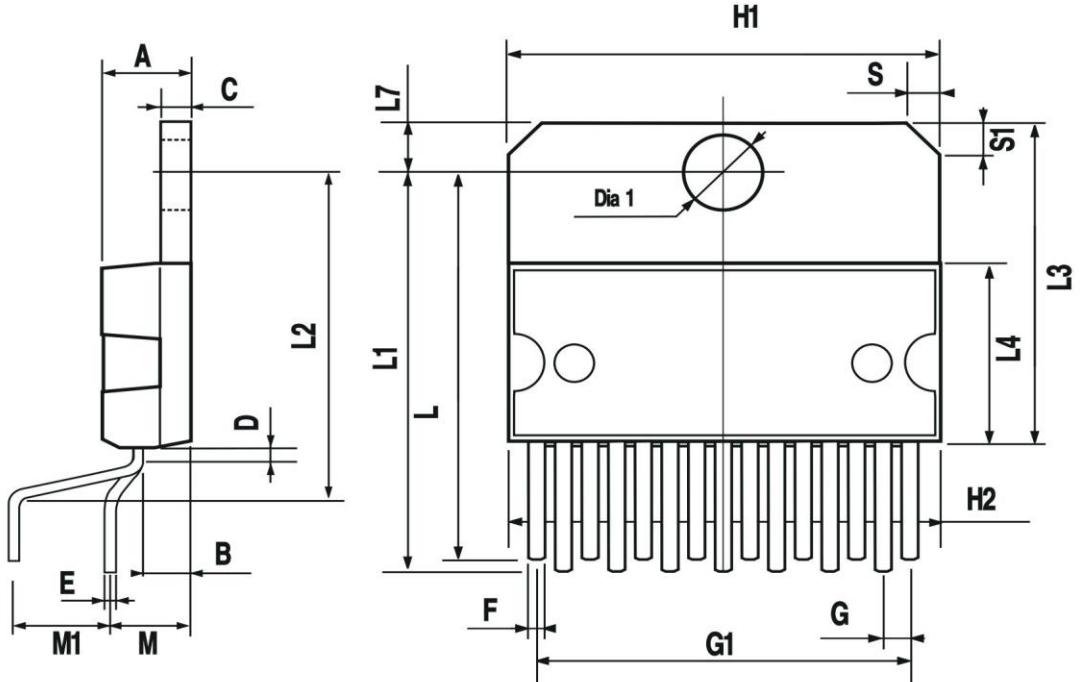


DIM.	mm			inch		
	MIN.	TYP.	MAX.	MIN.	TYP.	MAX.
A			5			0.197
B			2.65			0.104
C			1.6			0.063
D		1			0.039	
E	0.49		0.55	0.019		0.022
F	0.66		0.75	0.026		0.030
G	1.02	1.27	1.52	0.040	0.050	0.060
G1	17.53	17.78	18.03	0.690	0.700	0.710
H1	19.6			0.772		
H2			20.2			0.795
L	21.9	22.2	22.5	0.862	0.874	0.886
L1	21.7	22.1	22.5	0.854	0.870	0.886
L2	17.65		18.1	0.695		0.713
L3	17.25	17.5	17.75	0.679	0.689	0.699
L4	10.3	10.7	10.9	0.406	0.421	0.429
L7	2.65		2.9	0.104		0.114
M	4.25	4.55	4.85	0.167	0.179	0.191
M1	4.63	5.08	5.53	0.182	0.200	0.218
S	1.9		2.6	0.075		0.102
S1	1.9		2.6	0.075		0.102
Dia1	3.65		3.85	0.144		0.152

OUTLINE AND MECHANICAL DATA

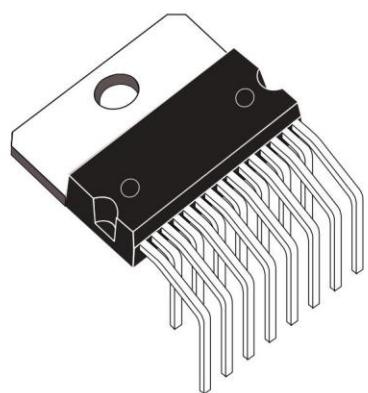


Multiwatt15 V

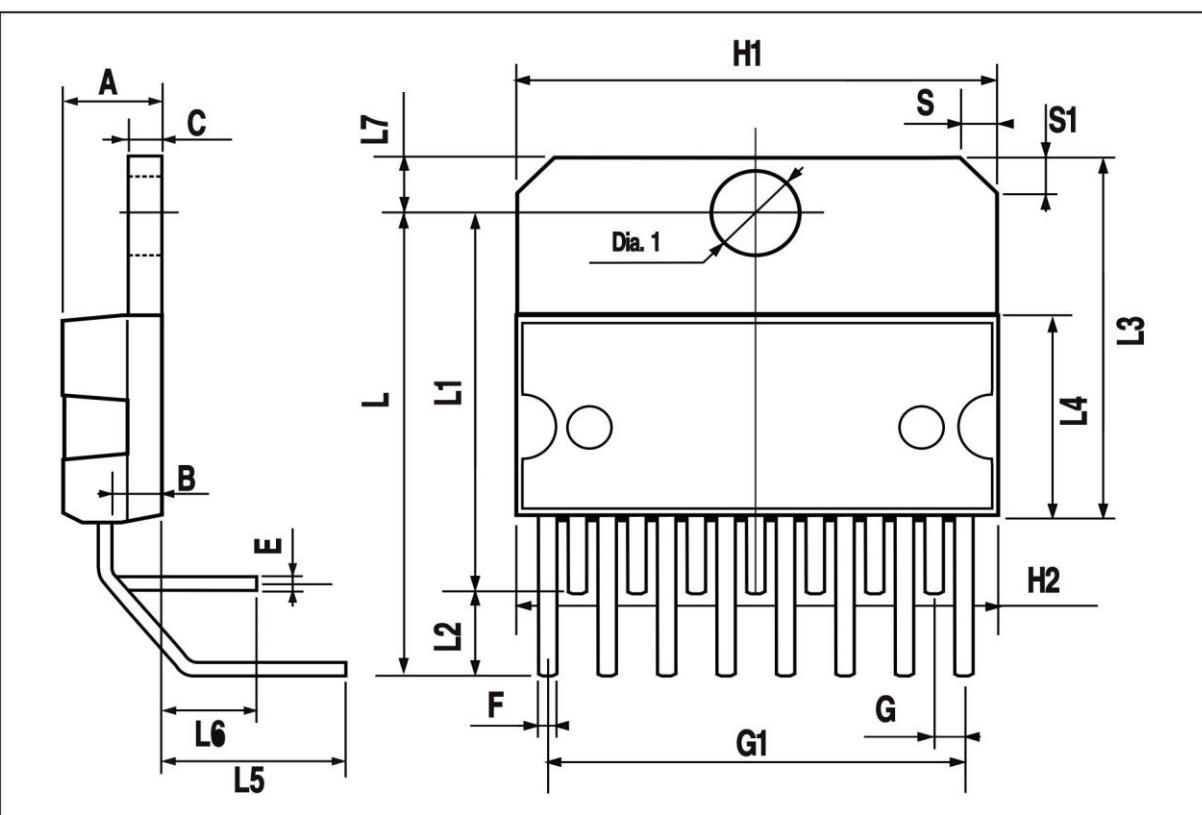


DIM.	mm			inch		
	MIN.	Typ.	MAX.	MIN.	Typ.	MAX.
A			5			0.197
B			2.65			0.104
C			1.6			0.063
E	0.49		0.55	0.019		0.022
F	0.66		0.75	0.026		0.030
G	1.14	1.27	1.4	0.045	0.050	0.055
G1	17.57	17.78	17.91	0.692	0.700	0.705
H1	19.6			0.772		
H2			20.2			0.795
L		20.57			0.810	
L1		18.03			0.710	
L2		2.54			0.100	
L3	17.25	17.5	17.75	0.679	0.689	0.699
L4	10.3	10.7	10.9	0.406	0.421	0.429
L5		5.28			0.208	
L6		2.38			0.094	
L7	2.65		2.9	0.104		0.114
S	1.9		2.6	0.075		0.102
S1	1.9		2.6	0.075		0.102
Dia1	3.65		3.85	0.144		0.152

OUTLINE AND MECHANICAL DATA



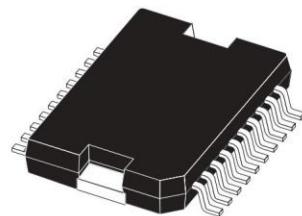
Multiwatt15 H



DIM.	mm			inch		
	MIN.	TYP.	MAX.	MIN.	TYP.	MAX.
A			3.6			0.142
a1	0.1		0.3	0.004		0.012
a2			3.3			0.130
a3	0		0.1	0.000		0.004
b	0.4		0.53	0.016		0.021
c	0.23		0.32	0.009		0.013
D (1)	15.8		16	0.622		0.630
D1	9.4		9.8	0.370		0.386
E	13.9		14.5	0.547		0.570
e		1.27			0.050	
e3		11.43			0.450	
E1 (1)	10.9		11.1	0.429		0.437
E2			2.9			0.114
E3	5.8		6.2	0.228		0.244
G	0		0.1	0.000		0.004
H	15.5		15.9	0.610		0.626
h			1.1			0.043
L	0.8		1.1	0.031		0.043
N	10° (max.)					
S	8° (max.)					
T		10			0.394	

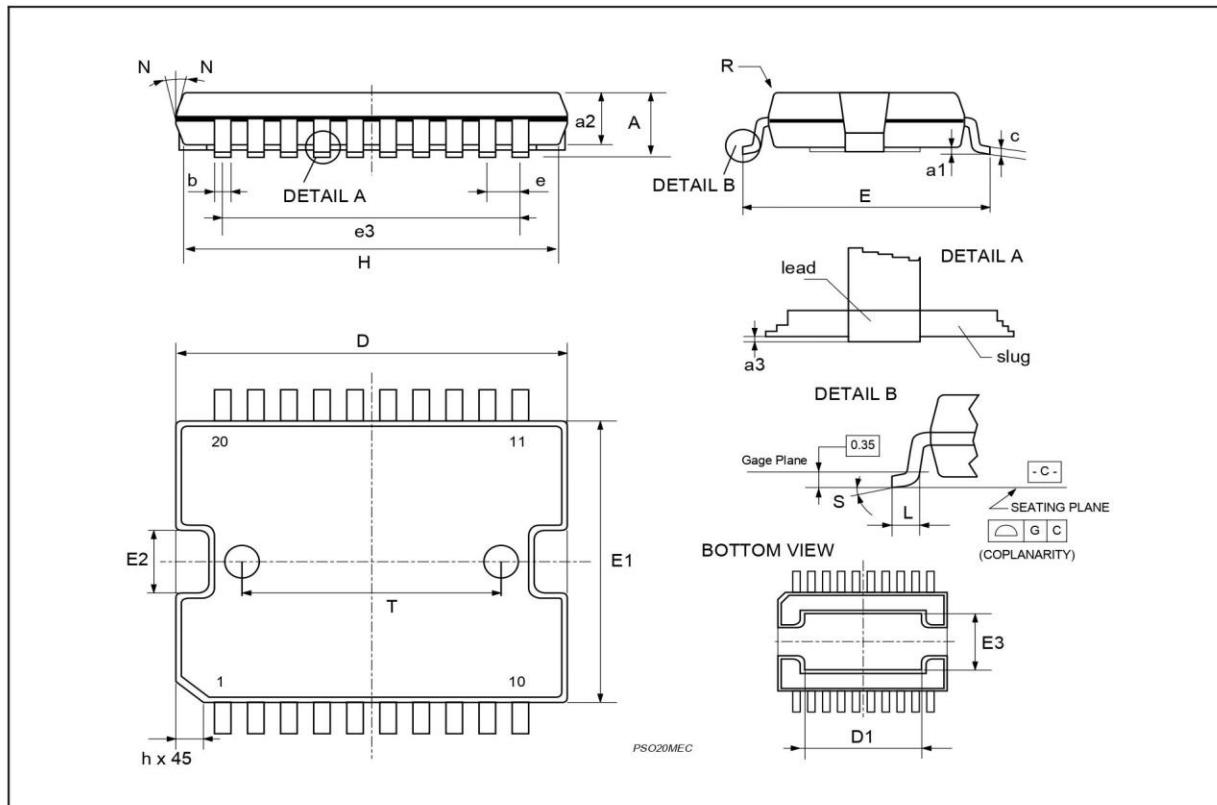
(1) "D and F" do not include mold flash or protrusions.
- Mold flash or protrusions shall not exceed 0.15 mm (0.006").
- Critical dimensions: "E", "G" and "a3"

OUTLINE AND MECHANICAL DATA



JEDEC MO-166

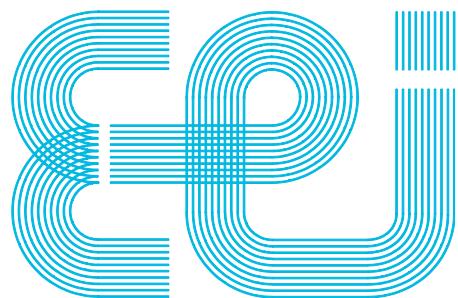
PowerSO20



Information furnished is believed to be accurate and reliable. However, STMicroelectronics assumes no responsibility for the consequences of use of such information nor for any infringement of patents or other rights of third parties which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of STMicroelectronics. Specification mentioned in this publication are subject to change without notice. This publication supersedes and replaces all information previously supplied. STMicroelectronics products are not authorized for use as critical components in life support devices or systems without express written approval of STMicroelectronics.

The ST logo is a registered trademark of STMicroelectronics
© 2000 STMicroelectronics – Printed in Italy – All Rights Reserved
STMicroelectronics GROUP OF COMPANIES

Australia - Brazil - China - Finland - France - Germany - Hong Kong - India - Italy - Japan - Malaysia - Malta - Morocco -
Singapore - Spain - Sweden - Switzerland - United Kingdom - U.S.A.
<http://www.st.com>



Escuela de Ingeniería Industrial

TRABAJO FIN DE GRADO

Implementación del filtrado de retraso temporal en tiempo real para un microcontrolador de 32 bits

Grado en Ingeniería en Electrónica Industrial y Automática

Documento

PRESUPUESTO

UniversidadeVigo

ÍNDICE

Índice.....	2
1 Presupuesto TFG.....	3
1.1 Mano de obra.....	3
1.2 Hardware	4
1.3 Software	5
1.4 Valoración económica total	6

1 PRESUPUESTO TFG

1.1 Mano de obra

	Cantidad (horas)	Precio (€)	Total (€)
Trabajo de un ingeniero técnico industrial	300	19,39	5.817,00
Total (€)			5.817,00

El coste de la mano de obra de este proyecto asciende a *cinco mil ochocientos diecisiete euros*.

1.2 Hardware

En el coste del hardware se incluyen los elementos que componen el proyecto, los cuales se han explicado en el apartado 8.

Elemento	Cantidad	Precio (€)	Total (€)
PSoC 5LP	1	209,29	209,29
Joystick	1	5,48	5,48
Motor DC	1	56,17	56,17
Controlador L298N	1	5,90	5,90
Pulsador	1	0,47	0,47
Fuente de alimentación	1	15,95	15,95
Total (€)			293,26

El coste de los elementos hardware de este proyecto asciende a *doscientos noventa y tres euros con veintiséis céntimos*.

1.3 Software

Para la realización de este proyecto se han necesitado diversas licencias de programas, las licencias son de duración determinada, mayor de la necesaria por lo que podrían amortizarse utilizándolas para otros proyectos.

Licencia	Duración	Precio (€)	Total (€)
Matlab	1 año	800,00	800,00
PSoC Creator	Permanente	Gratis	0,00
SolidWorks	3 meses	947,00	947,00
Putty	Permanente	Gratis	0,00
Algoritmo del filtrado de retraso temporal	-	A determinar por el autor*	-
		Total (€)	1.747,00

(*) El autor del código de implementación del filtro de retraso temporal es el profesor Gerardo Peláez Lourido, profesor de la Universidad de Vigo.

El coste del software de este proyecto asciende a *mil setecientos cuarenta y siete euros*.

1.4 Valoración económica total

Sumando los costes de los anteriores apartados podemos obtener la valoración económica total o presupuesto del proyecto.

Concepto	Coste (€)
Mano de obra	5.817,00
Hardware	293,26
Software	1.747,00
Total (€)	7.857,26

La valoración económica total de este proyecto asciende a *siete mil ochocientos cincuenta y siete euros con veintiséis céntimos*.

Vigo, 4 de Julio de 2022
Fdo. Julio Benito Represa Montenegro