

UNIVERSIDAD POLITÉCNICA DE MADRID

**ESCUELA TÉCNICA SUPERIOR
DE INGENIEROS DE TELECOMUNICACIÓN**



**MÁSTER UNIVERSITARIO EN INGENIERÍA DE
SISTEMAS ELECTRÓNICOS**

TRABAJO FIN DE MÁSTER

**MODELADO Y DISEÑO DE UN SISTEMA
CONTROL DE UNA ESTRUCTURA DE BALANCEO
DE DOS GRADOS DE LIBERTAD**

MARTIN NENKOV GOROSTANOV

2021

UNIVERSIDAD POLITÉCNICA DE MADRID

**ESCUELA TÉCNICA SUPERIOR
DE INGENIEROS DE TELECOMUNICACIÓN**



**MÁSTER UNIVERSITARIO EN INGENIERÍA DE
SISTEMAS ELECTRÓNICOS**

TRABAJO FIN DE MÁSTER

**MODELADO Y DISEÑO DE UN SISTEMA
CONTROL DE UNA ESTRUCTURA DE BALANCEO
DE DOS GRADOS DE LIBERTAD**

Autor

MARTIN NENKOV GOROSTANOV

Tutor

ÁLVARO GUTIÉRREZ MARTÍN

2021

Resumen

El control automatizado ha desempeñado un papel muy importante en el avance de la ingeniería en las últimas décadas. Es una parte fundamental de los sistemas robóticos, sistemas de vehículos autónomos y, en general, en cualquier proceso industrial en el que se requiera de un ajuste preciso de posición, temperatura, humedad, etc.

La ingeniería de control realiza un estudio de los sistemas físicos con el objetivo de desarrollar controladores capaces de dirigir estos sistemas hacia un estado objetivo. En la industria, la implementación de estos controladores se realiza por parte de los ingenieros electrónicos, mediante el uso de microcontroladores, que son los encargados de procesar la señal de entrada, calcular la acción de control y dar la orden a los actuadores.

Nuestra labor, como ingenieros electrónicos, está fuertemente ligada a este trabajo de implementación. Por lo tanto, es necesario conocer las distintas técnicas de control que existen, con el objetivo de realizar una implementación mas fluida e incluso para poder optimizar estos controladores o las arquitecturas en las que serán implementados.

En este Trabajo Fin de Máster se propone el control de una estructura de dos grados de libertad con el objetivo de analizar su respuesta real frente a la modelada y la viabilidad de la implementación del sistema. Para cumplir con el objetivo del proyecto, se ha decidido dividir el trabajo en varias etapas. Inicialmente, se propone realizar un análisis del modelado físico del sistema con el objetivo de conocer su respuesta temporal ante determinadas perturbaciones. A continuación, se realizará el diseño y simulación de los controladores con los que se propone realizar el control en posición de una esfera colocada en la estructura. Además, en este Trabajo Fin de Máster se propone el diseño de dos controladores, uno corresponde con la teoría de control clásica, que busca realizar un control temporal de los sistemas y un segundo controlador, basado en la teórica de control óptimo, en el que el objetivo es desarrollar un controlador que satisfaga una determinada función de coste. Por último, se realizará la implementación, ajuste y la validación del control en un microcontrolador.

Abstract

Automated control has played a very important role in the advancement of engineering in recent decades. It is a fundamental part of robotics, autonomous vehicles and, in general, in any industrial process where precise control of position, temperature, humidity, etc. is required.

Control engineering studies physical systems with the aim of developing controllers capable of directing these systems towards a target state. In industry, the implementation of these controllers is carried out by electronic engineers, through the use of microcontrollers, which are responsible for processing the input signal, calculating the control signal and giving the order to the actuators.

Our work as electronic engineers is strongly related to this work. Therefore, it is necessary to know the different control techniques that exist, with the aim of making a smoother implementation and to be able to optimise these controllers or the architectures in which they will be implemented.

This Msc Thesis proposes the control of a structure with two degrees of freedom in order to analyse its real response compared to that modelled and the feasibility of the implemented solution. In order to fulfil the objective of the project, it is divided into several stages. Initially, it is proposed to carry out an analysis of the physical modelling of the system in order to determine its temporal response to certain disturbances. Next, the design and simulation of the controllers has to be done with which the control of the position of the sphere will be made. In addition, this Master's Thesis proposes the design of two controllers, one corresponding to the classical control theory, which seeks to perform a temporal control of the systems and a second controller, based on the theory of optimal control, in which the objective is to develop a controller that satisfies a certain cost function. Finally, the control will be implemented in a microcontroller and its adjustment and validation will be carried out.

Índice general

Resumen	V
Abstract	VII
Índice General	IX
Índice de Figuras	XIII
Índice de Tablas	XV
Lista de Acrónimos	XVII
1. Introducción	1
1.1. Introducción a la ingeniería de control	1
1.2. Motivación y objetivos del proyecto	2
1.3. Estructura del documento	2
2. Modelado de un sistema <i>Ball and Plate</i>	5
2.1. Introducción	6
2.2. Modelado del sistema	7
2.2.1. Representación en transformada de Laplace	9
2.2.2. Representación en modelo de variables de estado	11
2.3. Análisis de la respuesta de un sistema	13
2.4. Lugar de las raíces	18
2.5. Diseño de controladores	20
2.5.1. Controlador Proporcional-Derivativo-Integral	21
2.5.1.1. Sintonización del controlador mediante el LDR	22
2.5.2. Regulador cuadrático-lineal (LQR)	23
2.5.2.1. Ajuste empírico del controlador LQR	26
3. Descripción del <i>hardware</i> del sistema	29
3.1. Estructura mecánica	29
3.2. Sensores	30
3.2.1. Pantalla resistiva	31
3.2.2. Cámara	32
3.3. Actuadores	33
3.4. Microcontrolador	34
3.5. Desarrollo de la tarjeta electrónica	35

3.5.1. Realización de esquemáticos	35
3.5.2. Posicionamiento de los componentes	36
3.5.3. Rutado del circuito impreso	37
4. Diseño e implementación del sistema	39
4.1. Descripción del <i>software</i> empleado	39
4.1.1. Sistema operativo en tiempo real: FreeRTOS	39
4.1.2. Entorno de desarrollo: STM32CubeIDE	40
4.2. Modelado del sistema	40
4.2.1. Análisis de requisitos	41
4.2.2. Máquinas de estado finitas	41
4.3. Desarrollo del <i>firmware</i>	42
4.3.1. Implementación del <i>firmware</i> de la cámara	42
4.3.2. Implementación del <i>firmware</i> del microcontrolador	42
4.3.2.1. Comunicación con los sensores	43
4.3.2.2. Comunicación con los servo-motores	44
4.3.2.3. Estructura del <i>software</i> del sistema	45
4.4. Desarrollo de controladores	46
4.4.1. Controlador PID	46
4.4.2. Regulador cuadrático-lineal	49
4.5. Pruebas y ajuste del sistema	51
4.5.1. Calibración de la estructura	52
4.5.2. Corrección de la posición de la esfera	52
4.6. Planificación en tiempo real	52
4.7. Optimización del consumo	56
5. Resultados y validación del sistema completo	59
5.1. Controlador PID	59
5.1.1. Sistema con pantalla resistiva	59
5.1.2. Sistema con cámara	62
5.2. Controlador LQR	63
5.2.1. Sistema con pantalla resistiva	63
6. Conclusiones	67
6.1. Líneas futuras	68
Bibliografía	72
A. Esquemáticos y circuito impreso de la tarjeta de expansión	73
B. Diseño de las máquinas de estados finitas	79
B.1. Lectura de los datos de los sensores	79
B.2. Cálculo de la acción de control y actuación sobre servo-motores	80
B.3. Envío de datos del sistema al exterior	81
B.4. Recepción de datos del exterior	82
C. Análisis del periodo de diferenciación	83

D. Aspectos éticos, económicos, sociales y ambientales	87
E. Presupuesto económico	89

Índice de figuras

2.1. Sistema de control en lazo cerrado.	6
2.2. Representación de la estructura. Researchgate.	7
2.3. Sistema realimentado en lazo cerrado. Dademuch.	10
2.4. Representación del espacio de estados. Springer.	12
2.5. Respuesta temporal de un sistema de segundo orden a una entrada escalón. ELAI-UPM.	14
2.6. Lugar de las raíces del sistema sin controlador.	20
2.7. Sistema realimentado con controlador PID. Researchgate.	21
2.8. Sistema realimentado con controlador LQR. Springer.	24
3.1. Fotografía de la estructura mecánica del sistema.	29
3.2. Relación entre ángulo del servo-motor y plataforma.	30
3.3. Esquema del circuito resistivo de la pantalla táctil. Microchip.	31
3.4. Fotografía de la cámara OpenMV Cam H7. OpenMV.	32
3.5. Fotografía de un servo-motor. Bricogeek.	34
3.6. Fotografía de la placa de desarrollo. STMicroelectronics.	35
4.1. Diagrama de tiempos de la señal PWM para el control de los servo- motores. PanamaHitek.	44
4.2. Estructura del código implementado.	45
4.3. Respuesta del sistema ante distintos periodos de muestro.	47
4.4. Respuesta del sistema aplicando el controlador PD diseñado.	48
4.5. Lugar de las raíces del sistema aplicando el controlador PD diseñado. .	49
4.6. Respuesta del sistema aplicando el controlador PD diseñado.	50
4.7. Ángulo de la plataforma ante un escalón de 20 centímetros.	51
5.1. Respuesta del sistema ante el PD teórico.	60
5.2. Acción de control del sistema ante el PD teórico.	60
5.3. Respuesta del sistema ante el PD con ajuste de ganancia.	61
5.4. Respuesta del sistema ante el PD con ajuste de cero.	61
5.5. Respuesta del sistema ante el PD experimental.	62
5.6. Respuesta del sistema formado por la cámara ante el PD teórico. . . .	63
5.7. Respuesta del sistema formado por la cámara ante el PD experimental. .	64
5.8. Respuesta del sistema con pantalla resistiva ante el LQR.	64
5.9. Respuesta del sistema con cámara ante el LQR.	65
A.1. Esquemático de la tarjeta de expansión.	74

A.2. Modelo 3D de la tarjeta de expansión.	75
A.3. Captura de la capa superior de la PCB.	76
A.4. Captura de la capa inferior de la PCB.	77
B.1. Diagrama de entradas y salidas de la máquina de toma de datos de los sensores.	79
B.2. Diagrama de estados de la máquina de toma de datos de los sensores.	80
B.3. Diagrama de entradas y salidas de la máquina de control.	80
B.4. Diagrama de estados de la máquina de control.	81
B.5. Diagrama de entradas y salidas de la máquina de envío de datos. . . .	81
B.6. Diagrama de estados de la máquina de envío de datos.	81
B.7. Diagrama de entradas y salidas de la máquina de recepción de comandos.	82
B.8. Diagrama de estados de la máquina de envío de recepción de comandos.	82
C.1. Señal de control del sistema ante diferentes periodos de muestreo. . . .	84
C.2. Respuesta del sistema ante el PD utilizado para el análisis del efecto del periodo de diferenciación.	85
C.3. Respuesta del sistema ante diferentes periodos de muestreo.	86

Índice de tablas

4.1. Tiempos asociados a la ejecución de las tareas.	54
4.2. Asignación de prioridades a las Tareas.	55
4.3. Tiempos de respuesta de las Tareas.	55
E.1. Costes de personal.	89
E.2. Costes de recursos materiales.	90
E.3. Costes totales.	90

Lista de Acrónimos

UPM : Universidad Politécnica de Madrid

ETSIT : Escuela Técnica Superior de Ingenieros de Telecomunicación

TFM : Trabajo Fin de Máster

LTI : Linear Time-Invariant

SISO : Single-Input Single-Output

MIMO : Multiple-Input Multiple-Output

PID : Proportional–Integral–Derivative

LQR : Linear–Quadratic Regulator

LDR : Lugar De las Raíces

I2C : Inter-Integrated Circuit

SPI : Serial Peripheral Interface

UART : Universal Asynchronous Receiver-Transmitter

USB : Universal Serial Bus

RAM : Random-Access Memory

PWM : Pulse-Width Modulation

RTOS : Real-Time Pperating System

FSM : Finite-State Machine

PCB : Printed Circuit Board

FIFO : First In First Out

HAL : Hardware Abstraction Layer

DMA : Direct Memory Access

GPIO : General Purpose Input/Output

SRAM : Static Random-Access Memory

Capítulo 1

Introducción

1.1. Introducción a la ingeniería de control

El control automatizado ha desempeñado un papel muy importante en el avance de la ingeniería en las últimas décadas. Es una parte fundamental de los sistemas robóticos[1], sistemas de vehículos autónomos y, en general, en cualquier proceso industrial en el que se requiera de un ajuste preciso de posición, temperatura, humedad, etc.

Al comienzo del siglo pasado, se empiezan a desarrollar controladores para el guiado automático de embarcaciones[2][3], demostrando que es posible controlar de forma estable un sistema a partir de las ecuaciones diferenciales que describen su dinámica. A raíz de estos logros, surgieron teorías que analizaban la estabilidad del sistema en lazo cerrado con su respuesta en lazo abierto. Junto con esto, a mediados de siglo, se desarrollan métodos para observar y analizar la respuesta del sistema en frecuencia, dando la posibilidad de diseñar controladores lineales que satisfagan un amplio conjunto de especificaciones. Esta etapa inicial de la ingeniería de control se conoce como teoría de control clásica, en la que se hace uso de las técnicas comentadas anteriormente, junto con controladores PID realizar el control de sistemas lineales invariantes en el tiempo (LTI) con una entrada y una salida (SISO).

Con el avance de la industria, surgen máquinas más complejas, con respuesta variante en el tiempo y múltiples entradas (MIMO), que requieren de controladores más sofisticados. Debido a esto, surgen teorías de control modernas en las que se realiza un análisis en tiempo de los sistemas y se describe su comportamiento mediante variables de estado. Junto con el desarrollo de la industria, la ingeniería de control aumenta su complejidad desarrollando nuevas teorías, buscando controladores óptimos[4], intentando optimizar la respuesta según un criterio, robustos[5], con el objetivo de asegurar la respuesta del sistema real en un determinado punto de trabajo, o adaptativos[6], capaces de adaptar el controlador en función del estado del sistema.

1.2. Motivación y objetivos del proyecto

Es fundamental que los ingenieros que realicen tareas de control estén familiarizados con las teorías de control que existen y los distintos tipos de controladores que se pueden implementar. Por este motivo, los alumnos de la asignatura de ingeniería de control deben disponer de sistemas reales en los que poder realizar la implementación y validación de estos controladores. La idea de diseñar e implementar un sistema en el que se puedan realizar estas labores educativas resulta interesante desde el punto de vista pedagógico.

El sistema que se encuentra disponible en el Laboratorio de Control y Robótica de la ETSIT no dispone de información sobre los controladores que implementa o la forma en la que se realiza su ajuste. Por este motivo, para que este sistema permita su uso en el ámbito educativo es necesario realizar una documentación extensa de su diseño, implementación y validación.

En este Trabajo Fin de Máster se propone el control de una estructura de dos grados de libertad con el objetivo de analizar su respuesta real frente a la modelada y la viabilidad de la implementación del sistema. Para cumplir con los objetivos principales del proyecto, es necesario definir algunos objetivos secundarios, es necesario realizar el modelado físico del sistema, diseño y simulación de controladores, implementación del control y por último, el ajuste del sistema y la validación del mismo.

1.3. Estructura del documento

Este documento se divide en seis capítulos en los que se detallan todos los pasos que se han seguido, junto con los componentes necesarios para la realización de este Trabajo Fin de Máster.

En el Capítulo 1 se realiza una breve introducción al contexto del TFM y se exponen la motivación y los objetivos de este.

En el Capítulo 2 se detallan los conceptos teóricos en los que se fundamenta este TFM. En estos capítulos se realiza una introducción a las dos teorías de control aplicadas y se explican los conceptos básicos que son necesarios para comprender la implementación del sistema.

En el Capítulo 3 se exponen todos los componentes por los que se compone el proyecto, junto con las decisiones tomadas para su selección. Además, en este capítulo se detallan todas las decisiones relacionadas con el *hardware* diseñado.

En el Capítulo 4 se expone la metodología que se ha seguido en el proyecto, las principales líneas de trabajo y se desarrolla detalladamente cada una de las partes del sistema. En este capítulo se encuentra el diseño, modelado e implementación del

sistema de control.

Por último, en los Capítulos 5 y 6 se exponen los resultados del proyecto junto con las conclusiones finales, además de incluir un apartado dedicado a posibles futuros trabajos.

Capítulo 2

Modelado de un sistema *Ball and Plate*

Es necesario remontarse a la antigua Grecia para conocer los inicios de la ingeniería de control. Los antiguos griegos, junto con los bizantinos y los romanos, emplearon sistemas de control para mantener los niveles de sus depósitos de agua, aceite y vino, respectivamente, estables. Para ello emplearon un sistema que, aunque antiguo, se sigue usando en la actualidad. Este sistema consistía en una esfera hueca que flotaba en el líquido y, que a su vez, controlaba el flujo de una válvula. De esta forma, cuando el líquido llegaba al nivel deseado, la válvula era cerrada de forma automática. Este sistema es idéntico al que se utiliza para el control de las cisternas de los inodoros en las casas. Desde entonces, la ingeniería de control ha seguido desarrollando nuevos controladores, sobre todo gracias a la revolución industrial, capaces de controlar todo tipo de sistemas, desde el control de velocidad de máquinas de vapor hasta el aterrizaje autónomo de naves espaciales, pasando por sistemas de pilotaje automático, robots industriales, etc. En la actualidad existen métodos muy complejos de control capaces de predecir los cambios ambientales por medio de algoritmos de inteligencia artificial, de controlar sistemas no-lineales o, incluso, el desarrollo de los sistemas distribuidos permite un control descentralizado capaz de comunicar diferentes nodos con un objetivo único.

Como se ha comentado en el capítulo 1, existen diferentes estrategias para el control de un sistema. Este TFM se centra en la teoría de control clásica, basada en el análisis de la dinámica de los sistemas, y de control óptimo, centrada en el diseño de controladores que minimicen una determinada función de coste. Para ello se realizará el análisis del sistema siguiendo cada una de estas teorías y la implementación de un controlador de cada tipo.

2.1. Introducción

El sistema que se pretende controlar es comúnmente conocido en la literatura anglosajona como *Ball and Plate*, en castellano, Bola y Plato. En este sistema, el objetivo es realizar el control en posición de una esfera maciza, normalmente de metal o de plástico, que se encuentra situada sobre un plato, o plataforma. Otra configuración similar es la que sólo dispone de un grado de libertad, llamada *Ball and Beam*, en castellano Bola y Viga. Ambas estructuras son comunes en el ámbito educativo, ya que permiten la implementación de una amplia variedad de controladores en una maqueta con un tamaño reducido. En la literatura se pueden encontrar diferentes métodos de adquisición de la posición de la esfera, como por ejemplo, el uso de pantallas resistivas, de cámaras con algoritmos de detección de objetos o, incluso, de sensores ultrasónicos. Por otro lado, para el control de la estructura se han desarrollado diferentes configuraciones mecánicas, como por ejemplo configuraciones en serie, en paralelo, etc.

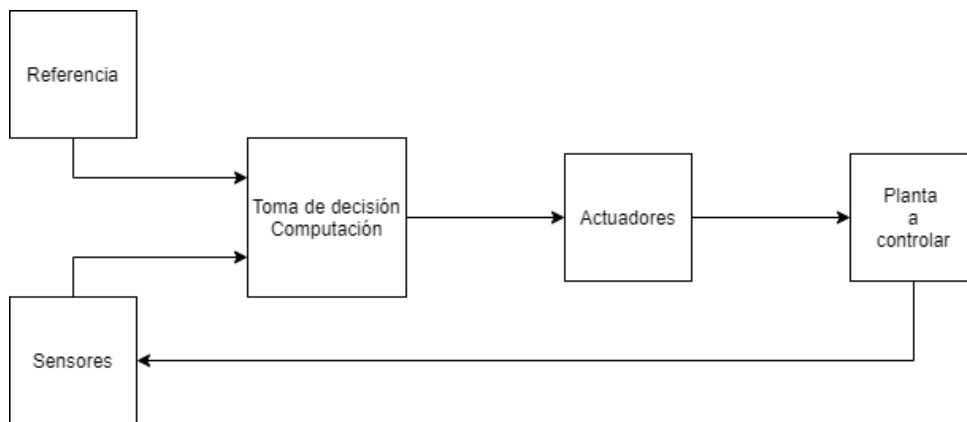


Figura 2.1: Sistema de control en lazo cerrado.

Generalmente los sistemas de control se componen de varios procesos. Inicialmente, es necesario la utilización de sensores que permiten conocer el estado actual del sistema. Posteriormente, se realiza un análisis de los datos tomados y, junto con la información que se dispone del sistema y del estado al que se desea llevar el sistema, se procesa la acción de control necesaria. Finalmente, ésta es comunicada al módulo encargado de controlar los actuadores, que perturban al sistema, modificando así su estado y comenzando de nuevo el ciclo. A este tipo de control se le conoce como control con realimentación en lazo cerrado ya que, como se observa en la Figura 2.1, se cierra el lazo de control gracias a los sensores, que realimentan el estado del sistema al controlador. Este tipo de configuración permite disminuir el efecto del ruido en el sistema y controlar sistemas originalmente inestables.

2.2. Modelado del sistema

El objetivo del modelado es describir el sistema mediante ecuaciones diferenciales que permitan conocer y controlar su comportamiento.

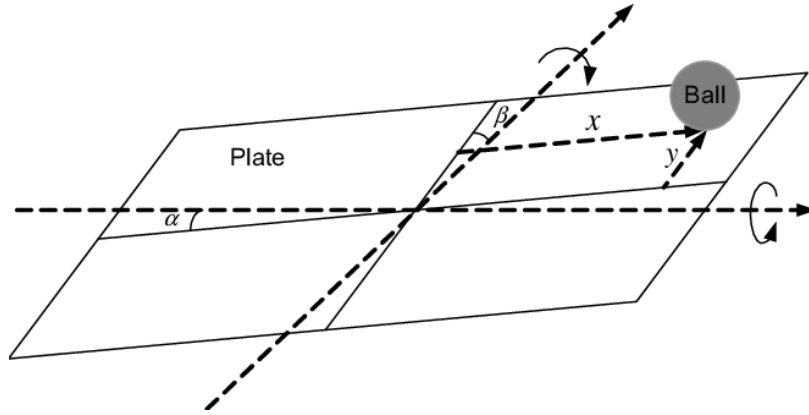


Figura 2.2: Representación de la estructura. Researchgate.

Antes de desarrollar las ecuaciones se necesario definir la estructura sobre la que se trabajará. Se observa en la Figura 2.2 un esquema de la plataforma con la que se desarrollará este TFM. En esta imagen se pueden observar algunas de las variables necesarias para realizar el control de la esfera. En este caso, las variables conocidas son el ángulo de la plataforma respecto al plano horizontal, definido como α , y la posición de la esfera en cada uno de los ejes, x e y .

Para realizar el análisis teórico de la estructura se aplica la primera ley de Newton, esto es, *un cuerpo no puede cambiar por sí solo su estado inicial, ya sea en reposo o en movimiento rectilíneo uniforme, a menos que se aplique una fuerza o una serie de fuerzas*. De forma simplificada, esta ley dice que un cuerpo se mantiene en reposo si la suma de las fuerzas que se le aplican es nula.

Para realizar un correcto análisis de la dinámica del sistema se ha realizado la suposición de que la esfera permanece en constante contacto con la plataforma sin que se produzca deslizamiento. De esta forma, toda la energía potencial que posee la esfera se transforma en energía cinética de traslación y rotación en torno al centro de masas. Es decir, las fuerzas que se ejercen sobre la esfera son aplicadas para modificar la velocidad de la esfera, sin pérdidas. Definida la estructura, los parámetros necesarios para su análisis y las leyes físicas con las que son consideradas en este caso, podemos obtener las ecuaciones que definen su mecánica:

$$\sum F = F_{tras} + F_{in} + F_g = 0 \quad (2.1)$$

Donde F_{tras} corresponde a la fuerza de traslación de la esfera, F_{in} , a la fuerza

equivalente al momento de inercia que presenta la esfera en movimiento y F_g , la fuerza de la gravedad que es ejercida sobre la esfera.

La fuerza de traslación en un movimiento curvilíneo presenta dos componentes, una tangencial, definida por la aceleración lineal de la esfera, \ddot{x} , y la masa de la esfera, m , y otra normal, definida por la masa de la esfera, la distancia de la esfera del centro de giro, x , y la velocidad angular de la estructura, $\dot{\alpha}$, mediante la siguiente ecuación:

$$F_{tras} = F_{tras}^t + F_{tras}^n = m \cdot \ddot{x} - m \cdot x \cdot \dot{\alpha}^2 \quad (2.2)$$

La fuerza de inercia queda definida mediante la siguiente ecuación por el momento de inercia de la esfera, J_b , el radio de la esfera, R , y su aceleración lineal, \ddot{x} :

$$F_{in} = \frac{J_b}{R^2} \cdot \ddot{x} \quad (2.3)$$

La fuerza de la gravedad, definida mediante la siguiente ecuación por la aceleración de la gravedad, g , la masa de la esfera, m , y el ángulo que forma el eje X de la estructura con la horizontal, α :

$$F_g = m \cdot g \cdot \sin(\alpha) \quad (2.4)$$

Desarrollando cada fuerza en su expresión física para uno de los ejes de la estructura y agrupando los términos, obtenemos:

$$\left(m + \frac{J_b}{R^2}\right)\ddot{x} - mx\dot{\alpha}^2 + mg\sin(\alpha) = 0 \quad (2.5)$$

En esta ecuación, se relaciona el ángulo de la plataforma, α , con la distancia de la esfera al centro, x , mediante su masa, m , su radio, R , su momento de inercia, J_b y la aceleración de la fuerza de la gravedad, g . Mediante la Ecuación 2.5 se puede definir completamente el comportamiento de la esfera ante las variaciones del ángulo de la plataforma. Desafortunadamente, esta ecuación no es lineal, por lo que para simplificar el modelado y su posterior control se necesario linealizar esta ecuación en un punto de trabajo. En este caso, el momento sobre el que se realiza este análisis corresponde al instante en el que la esfera se encuentra en una posición de reposo, es decir, cuando el ángulo de la plataforma es cero respecto de la horizontal.

Con este punto de trabajo, en el que $\sin(\alpha) \approx \alpha$, y $\dot{\alpha} = 0$, se obtiene:

$$\left(m + \frac{J_b}{R^2}\right)\ddot{x} + mg\alpha = 0 \quad (2.6)$$

Modificando el sistema de referencia, de tal forma que los ángulos positivos de la plataforma impliquen aceleraciones de la esfera hacia el exterior, se obtiene:

$$\left(m + \frac{J_b}{R^2}\right)\ddot{x} = mg\alpha \quad (2.7)$$

Llegado a este punto se puede optar por dos caminos diferentes para la transformación de este modelo con el objetivo de prepararlo para el diseño de controladores. Por un lado, se realiza una transformación al dominio de Laplace con el objetivo de realizar un control clásico, mediante el análisis de la respuesta temporal del sistema. También es posible realizar un modelado en espacio de estados, para realizar un control enfocado a la optimización de algún factor del sistema. En las secciones posteriores se realizará el análisis en ambos métodos.

2.2.1. Representación en transformada de Laplace

Esta transformación forma parte de lo que se conoce como la teoría de control clásica, en la época en la que se pretendía analizar la estabilidad de los sistemas. Algunas técnicas mediante las que se conseguía este objetivo son el análisis del lazo abierto del sistema, el lugar de las raíces, el teorema de Nyquist de la estabilidad, etc.

A partir de la Ecuación 2.7 que describe el sistema mediante ecuaciones diferenciales, se realiza la transformada de Laplace con el objetivo de obtener una función con la que se pueda definir el comportamiento del sistema en el tiempo, e incluso en la frecuencia. Aplicando esta transformada se obtiene:

$$\left(m + \frac{J_b}{R^2}\right)X(s)s^2 = mgA(s) \quad (2.8)$$

Reorganizando los términos con el objetivo de obtener una función de transferencia entre entrada, ángulo de la plataforma, y salida, distancia del centro a la esfera:

$$\frac{X(s)}{A(s)} = \frac{mg}{\left(m + \frac{J_b}{R^2}\right)s^2} \quad (2.9)$$

Sabiendo que el momento de inercia de una esfera maciza se encuentra relacionada con su radio y su masa de la siguiente forma:

$$J_b = \frac{2}{5}mR^2 \quad (2.10)$$

Además, se sustituye la aceleración de la gravedad por su valor, con lo que

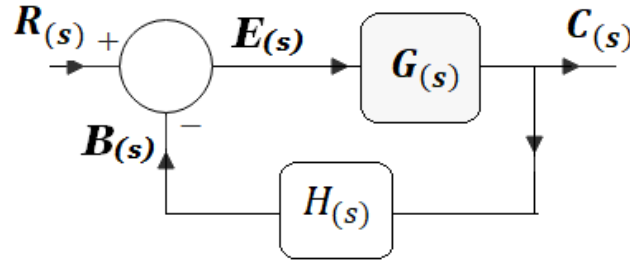


Figura 2.3: Sistema realimentado en lazo cerrado. Dademuch.

finalmente se obtiene:

$$\frac{X(s)}{A(s)} = \frac{5g}{7} \frac{1}{s^2} \approx \frac{7}{s^2} \quad (2.11)$$

Cabe resaltar el hecho de que la función de transferencia del sistema que se pretende controlar es totalmente independiente de la esfera que es colocada sobre la plataforma. Por otro lado, se observa que la función de transferencia presenta un polo doble en el origen, lo que hace al sistema críticamente estable. Dicho de otra forma, el sistema integra la señal de entrada, lo que hace imposible conseguir un control de la esfera con un control puramente proporcional.

Como se ha comentando anteriormente, la teoría clásica de control se basa en el análisis dinámico de los sistemas, por ello, una vez obtenida la función de transferencia del mismo, es necesario exponer algunas de las técnicas que se utilizan para realizar este análisis.

Antes de comenzar con este análisis es necesario definir una serie de conceptos. Como se ha visto al comienzo del capítulo, el control se realiza mediante la realimentación de la salida que, comparada con una referencia, es analizada por el controlador para obtener la acción de control. Como se puede observar en la Figura 2.3, la función de transferencia en el lazo directo es denominada $G(s)$ y la función de transferencia del lazo de realimentación, $H(s)$.

Siguiendo el esquema propuesto, se obtiene la función de transferencia que relaciona la salida del sistema, $C(s)$, con la referencia, $R(s)$:

$$\frac{C(s)}{R(s)} = \frac{E(s) \cdot G(s)}{E(s) + B(s)} = \frac{E(s) \cdot G(s)}{E(s) + C(s) \cdot H(s)} = \frac{E(s) \cdot G(s)}{E(s) + E(s) \cdot G(s) \cdot H(s)} \quad (2.12)$$

Y desarrollando la Ecuación 2.12 se obtiene la función final del sistema realimentado:

$$\frac{C(s)}{R(s)} = \frac{G(s)}{1 + G(s) \cdot H(s)} \quad (2.13)$$

Por otro lado, es importante comentar que las raíces del numerador de la función de transferencia son los ceros y las raíces del denominador, los polos. Dadas las propiedades de la transformada de Laplace, los polos del sistema definen la naturaleza de la respuesta del régimen transitorio. Además, polos en el semieje positivo real suponen que el sistema presenta una respuesta inestable, es decir, ante cualquier señal de entrada, la salida del sistema crece indefinidamente. Teniendo esto en mente, es necesario realizar un análisis sobre la función $1+G(s)\cdot H(s)$, ya que es el denominador de la función de transferencia y define el lugar de los polos del sistema.

2.2.2. Representación en modelo de variables de estado

La representación de los sistemas en el dominio de Laplace forma parte de la teoría de control clásica, en la que se pretende realizar el control de sistemas LTI y SISO. Con el avance de la industria, y sobre todo de los sectores industrial, automoción, defensa y espacio, surgen máquinas con dinámicas muy complejas y que requieren del control de muchas variables para operar. Con la necesidad de controlar y analizar el comportamiento de estos sistemas surgen nuevas técnicas de representación de sistemas.

Una de las técnicas comúnmente utilizadas es el espacio de estados. Por medio de esta representación se definen variables de estado que presentan una determinada relación entre ellas y que, mediante su análisis temporal, permiten describir el comportamiento de sistemas complejos. Además, esta técnica permite el control y análisis de múltiples entradas y salidas en un mismo sistema.

Como el nombre de este modelo indica, es necesario definir una serie de estados. Para ello se puede observar la Ecuación 2.7 que describe el sistema. Para evitar confusiones más adelante, para este modelado se define a r como la distancia de la esfera al centro de la estructura. Desarrollando esta ecuación, se obtiene:

$$\left(m + \frac{J_b}{R^2}\right)\ddot{r}(t) + 0\dot{r}(t) + 0r(t) = mg\alpha \quad (2.14)$$

Simplificando el momento de inercia a partir de la Ecuación 2.10, se obtiene:

$$\frac{7}{5}\ddot{r}(t) + 0\dot{r}(t) + 0r(t) = g\alpha \quad (2.15)$$

A partir de la ecuación 2.15 se definen fácilmente los estados que son necesarios para este modelo, además de la relación que existe entre estos y la salida, r .

$$x_1 = r \quad x_2 = \dot{r} \quad (2.16)$$

Por lo tanto, se define la variación temporal de los estados de la siguiente forma:

$$\dot{x}_1 = \dot{r} \quad \dot{x}_2 = \frac{5g}{7}\alpha \quad (2.17)$$

Además, se puede representar la salida en función de los estados, de tal forma que:

$$y = x_1 \quad (2.18)$$

Una vez definidos los estados, la relación entre ellos y su relación con la salida, se realiza una definición formal del sistema en el modelo de espacio de estados siguiendo el esquema de la Figura 2.4.

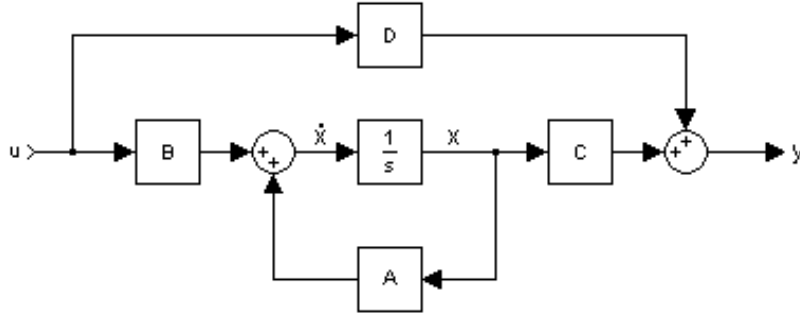


Figura 2.4: Representación del espacio de estados. Springer.

Por lo que las ecuaciones formales del modelo de espacio de estados son:

$$\dot{X} = AX + BU \quad Y = CX + DU \quad (2.19)$$

Como se puede observar en este modelo existe la posibilidad de realizar el control de un sistema con varias entradas y salidas, ya que todas las variables de este modelo son matriciales. Finalmente, se representa el sistema en el modelo de espacio de estados de la siguiente manera:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \dot{X} = A \cdot X + B \cdot U = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{5g}{7} \end{bmatrix} \cdot [\alpha] \quad (2.20)$$

$$[y] = Y = C \cdot X + D \cdot U = \begin{bmatrix} 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \end{bmatrix} \cdot [\alpha] \quad (2.21)$$

Sustituyendo los estados del modelo por las variables del sistema que son

conocidas, se obtiene:

$$\begin{bmatrix} \dot{r} \\ \ddot{r} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} r \\ \dot{r} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{5g}{7} \end{bmatrix} \cdot [\alpha] \quad (2.22)$$

$$\begin{bmatrix} r \end{bmatrix} = \begin{bmatrix} 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} r \\ \dot{r} \end{bmatrix} + \begin{bmatrix} 0 \end{bmatrix} \cdot [\alpha] \quad (2.23)$$

Cabe destacar que, por el modo en el que se ha definido el sistema en el espacio de estados, los estados son fácilmente observables. Esto es, los estados que seleccionados corresponden con la posición y la velocidad de la esfera y, por tanto, estas variables son conocidas.

2.3. Análisis de la respuesta de un sistema

Con carácter general los sistemas se analizan con tres tipos de respuesta temporal en función de la entrada del sistema. Son la respuesta al impulso, cuando la señal de entrada presenta la forma de la función delta de Dirac; la respuesta al escalón, cuando la señal de entrada presenta la forma de la función escalón de Heaviside, es decir, presenta un valor nulo en cualquier tiempo anterior a un instante t_0 y un valor unitario para cualquier tiempo superior o igual a t_0 ; y la respuesta a la rampa, de forma similar al escalón pero después del instante t_0 la señal de entrada es creciente con pendiente constante. Estas son las señales de entrada más comunes debido a que con la primera, el impulso, se obtiene la respuesta del sistema ante una entrada puntual y muy energética y, la respuesta al escalón y a la rampa intentan representar el comportamiento del sistema ante una referencia en posición y velocidad respectivamente. Este TFM se centra en la respuesta al escalón, debido a que el objetivo es realizar un control de la esfera en posición. Por otro lado, para simplificar el desarrollo se realizará el análisis de un sistema de segundo orden, es decir, que posee dos polos. En general, la mayoría de sistema físicos permiten ser descritos mediante una función de segundo orden o inferior.

Representando un sistema de segundo orden con polos complejos conjugados con la transformada de Laplace, se obtiene la primera ecuación canónica:

$$G(s) = \frac{k\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} = \frac{k\omega_n^2}{(s + \sigma)^2 + \omega_d^2} \quad (2.24)$$

En esta representación, los polos del sistema quedan definidos por ω_n , la frecuencia natural del sistema y ζ , el coeficiente de amortiguamiento. Además, se puede observar que desarrollando esta ecuación se representan los polos por su componente real, σ , e imaginaria, ω_d , en el dominio de Laplace.

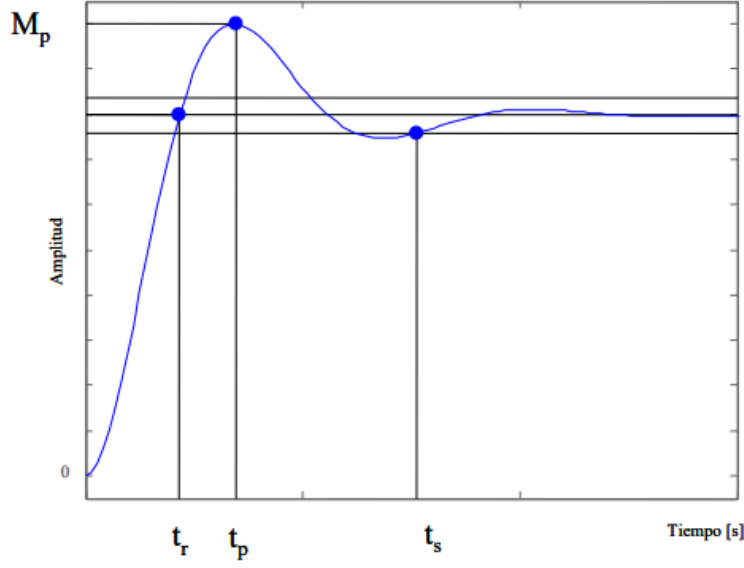


Figura 2.5: Respuesta temporal de un sistema de segundo orden a una entrada escalón. ELAI-UPM.

La Figura 2.5 representa la respuesta temporal de un sistema de segundo orden a una entrada escalón. Como se observa, ésta queda definida por varios parámetros. En primer lugar, el tiempo de subida, t_r , correspondiente al tiempo en el que la salida del sistema alcanza por primera vez el valor final, M_p y t_p , representando la sobreoscilación en tanto por ciento y el tiempo de pico en el que se alcanza respectivamente y el tiempo de establecimiento, t_s , como el tiempo en el que la salida de sistema alcanza su valor final y se mantiene hasta el infinito.

Además, estos parámetros están directamente relacionados con los polos del sistema. Por lo tanto, si se consigue ajustar la posición de estos polos en el dominio de Laplace, se puede ajustar la respuesta del sistema según unas especificaciones dadas. Para obtener la relación que existe entre estos parámetros y la posición de los polos, es necesario realizar la transformada inversa de Laplace de la función de transferencia ante una entrada de tipo escalón.

La función en el dominio de Laplace que debe ser transformada se puede observar en la Ecuación 2.25.

$$Y(s) = G(s) \cdot F_{escalón}(s) = \frac{k\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \cdot \frac{1}{s} = \frac{k\omega_n^2}{(s + \sigma)^2 + \omega_d^2} \cdot \frac{1}{s} \quad (2.25)$$

De esta forma, para realizar la transformada inversa de Laplace es necesario representar esta función en la suma de productos parciales de la siguiente forma:

$$Y(s) = \frac{A}{s} + \frac{Bs + C}{(s + \sigma)^2 + \omega_d^2} \quad (2.26)$$

Se obtiene el valor de A del siguiente modo:

$$A = Y(s) \cdot s \Big|_{s=0} = \frac{k \cdot \omega_n^2}{\omega_n^2} = k \quad (2.27)$$

Por otro lado, para obtener los valores de B y C , se sustituye el valor de A y se igualan las Ecuaciones 2.25 y 2.26:

$$Y(s) = \frac{k\omega_n^2}{(s + \sigma)^2 + \omega_d^2} \cdot \frac{1}{s} = \frac{k}{s} + \frac{Bs + C}{(s + \sigma)^2 + \omega_d^2} \quad (2.28)$$

De esta forma, igualando cada termino, se obtiene:

$$0 = k + B \quad (2.29)$$

en s^2 ,

$$0 = 2k\zeta\omega_n + C \quad (2.30)$$

en s y

$$k \cdot \omega_n^2 = k \cdot \omega_n^2 \quad (2.31)$$

en s^0 . De las Ecuaciones 2.29 y 2.30 se obtiene los valores de B y C , respectivamente. Finalmente, en la Ecuación 2.32 se muestra la función descompuesta en una suma de productos parciales.

$$Y(s) = \frac{k}{s} - \frac{ks + 2k\zeta\omega_n}{(s + \sigma)^2 + \omega_d^2} \quad (2.32)$$

Sabiendo que:

$$\sigma = \omega_n \cdot \zeta \quad (2.33)$$

y

$$\omega_d = \omega_n \cdot \sqrt{1 - \zeta^2} \quad (2.34)$$

Se obtiene:

$$Y(s) = \frac{k}{s} - k \frac{s + 2\sigma}{(s + \sigma)^2 + \omega_d^2} \quad (2.35)$$

Antes de realizar la transformación inversa de Laplace, es necesario modificar el segundo término de la ecuación. De esta forma, se obtiene:

$$Y(s) = \frac{k}{s} - k \frac{(s + \sigma) + \frac{\zeta}{\sqrt{1-\zeta^2}} \cdot \omega_d}{(s + \sigma)^2 + \omega_d^2} = k \left[\frac{1}{s} - \frac{(s + \sigma) + \frac{\zeta}{\sqrt{1-\zeta^2}} \cdot \omega_d}{(s + \sigma)^2 + \omega_d^2} \right] \quad (2.36)$$

Con todo esto, la función en el dominio del tiempo equivalente es:

$$y(t) = k \left[1 - e^{-\sigma t} \cdot \left(\cos(\omega_d t) + \frac{\zeta}{\sqrt{1-\zeta^2}} \cdot \text{sen}(\omega_d t) \right) \right] \quad (2.37)$$

Por otro lado, sabiendo que:

$$\zeta = \cos(\phi) \quad (2.38)$$

y

$$\sqrt{1 - \zeta^2} = \text{sen}(\phi) \quad (2.39)$$

Se obtiene la siguiente ecuación:

$$y(t) = k \left[1 - e^{-\sigma t} \cdot \left(\cos(\omega_d t) + \frac{\cos(\phi)}{\text{sen}(\phi)} \cdot \text{sen}(\omega_d t) \right) \right] \quad (2.40)$$

Sacando factor común $\text{sen}(\phi)$:

$$y(t) = k \left[1 - \frac{e^{-\sigma t}}{\text{sen}(\phi)} \cdot \left(\cos(\omega_d t) \cdot \text{sen}(\phi) + \cos(\phi) \cdot \text{sen}(\omega_d t) \right) \right] \quad (2.41)$$

y sabiendo que

$$\text{sen}(\omega_d t + \phi) = \cos(\omega_d t) \cdot \text{sen}(\phi) \pm \cos(\phi) \cdot \text{sen}(\omega_d t) \quad (2.42)$$

se obtiene la función en el dominio del tiempo que permite analizar la respuesta de un sistema de segundo orden:

$$y(t) = k \left(1 - \frac{e^{-\sigma t}}{\sqrt{1 - \zeta^2}} \text{sen}(\omega_d t + \phi) \right) \quad (2.43)$$

Para obtener el tiempo de establecimiento, es necesario comparar esta función con el valor en el régimen permanente. Suponiendo que se el objetivo es que la señal de salida se mantenga en un intervalo de $\pm 5\%$ del valor final, se obtiene:

$$\frac{e^{-\sigma t}}{\sqrt{1 - \zeta^2}} \approx 0,05 = e^{-\pi} \quad (2.44)$$

Y, teniendo en cuenta que el factor de amortiguamiento, ζ , al cuadrado presenta valores muy inferiores a 1, se obtiene el valor de t_s en función de la componente real del polo:

$$\zeta^2 \ll 1 \rightarrow t_s \approx \frac{\pi}{\sigma} \quad (2.45)$$

La sobreoscilación corresponde al valor máximo alcanzado en la respuesta temporal del sistema. Generalmente, se suele expresar como la relación entre el valor máximo frente al valor en el régimen permanente como un porcentaje. Por otro lado, el tiempo de pico es el instante en el que se produce este máximo. Para obtener el tiempo de pico, se debe obtener el máximo de la Ecuación 2.43 igualando su derivada a 0. Realizando el desarrollo de la derivada, reordenando, y resolviendo esta ecuación, se obtiene la relación del tiempo de pico con la parte imaginaria del polo:

$$\frac{dy(t)}{dt} = 0 \rightarrow \omega_d t_p = \pi \rightarrow t_p = \frac{\pi}{\omega_d} \quad (2.46)$$

Sustituyendo este tiempo en la Ecuación 2.43 se obtiene la sobreoscilación:

$$M_p = \frac{y_{max} - y_{inf}}{y_{inf}} = \frac{y(t_p) - k}{k} = e^{\frac{-\sigma \pi}{\omega_d}} = e^{tg(\phi)} \quad (2.47)$$

Cabe destacar que sólo habrá sobreoscilación si los polos del sistema son complejos conjugados sub-amortiguados. El límite en el que se produce la sobreoscilación se obtiene al comparar el tiempo en el que se produce el máximo de la respuesta (t_p) con el momento en el que nuestra salida alcanza su valor en el régimen permanente (t_s). Dicho de otra forma, si el tiempo de establecimiento es igual o inferior al tiempo de

pico, no existe sobreoscilación. De esta forma, comparando la Ecuación 2.45 con la Ecuación 2.46, se obtiene:

$$t_s < t_p \rightarrow \frac{\pi}{\sigma} < \frac{\pi}{\omega_d} \rightarrow \omega_d < \sigma \quad (2.48)$$

Se observa que el límite de la sobreoscilación se produce cuando la parte imaginaria del polo es igual a la real. De esta forma, se obtiene el valor del factor de amortiguamiento en el que se produce este fenómeno:

$$\zeta = \cos(\phi) = \frac{\sigma}{\sqrt{\sigma^2 + \omega_d^2}} = \frac{\sigma}{\sqrt{\sigma^2 + \sigma^2}} = \frac{1}{\sqrt{2}} = \frac{\sqrt{2}}{2} \quad (2.49)$$

Como se ha visto previamente, la sobreoscilación se produce cuando la parte imaginaria del polo presenta un valor superior a la real, es decir, sólo cuando $0 < \zeta < \sqrt{2}/2$. De forma similar, si no hay sobreoscilación, no hay tiempo de pico.

El tiempo de subida corresponde al intervalo que tarda en pasar desde el 10 % hasta el 90 % del valor en régimen permanente del sistema. En este caso, para simplificar los cálculos, se obtiene el tiempo que se tarda en alcanzar el 100 % del régimen permanente desde la posición inicial:

$$\frac{e^{-\sigma t}}{\sqrt{1 - \zeta^2}} \text{sen}(\omega_d t + \phi) = 0 \rightarrow \text{sen}(\omega_d t + \phi) = 0 \quad (2.50)$$

Por lo tanto, la relación entre el tiempo de subida y la posición del polo es:

$$\omega_d t_r + \phi = \pi \rightarrow t_r = \frac{\pi - \phi}{\omega_d} \quad (2.51)$$

2.4. Lugar de las raíces

El objetivo fundamental de la teoría de control clásica es analizar la estabilidad de los sistemas, por este motivo se han desarrollado métodos analíticos que permiten realizar este análisis. Una técnica muy utilizada es el lugar de las raíces (LDR). Este método permite obtener la respuesta del sistema en lazo cerrado mediante el análisis del mismo sistema en lazo abierto. Esta técnica presenta una gran ventaja frente a las demás, a parte de ser una técnica gráfica y por lo tanto tener una complejidad matemática reducida, permite observar el comportamiento del sistema cuando se produce una variación en un parámetro del mismo. Generalmente, este parámetro corresponde con una ganancia ajustable de la señal de error.

Después de haber realizado una introducción sobre la respuesta temporal y haber obtenido el modelo del sistema, se realiza la descripción de la técnica del LDR, que permitirá colocar los polos del sistema en lazo cerrado en la posición deseada. Para realizar el LDR es necesario volver a la función de transferencia del sistema realimentado, que se puede observar en la Figura 2.3 y la Ecuación 2.13. A partir de esta ecuación, se puede observar que la respuesta temporal de este sistema queda definida por la ecuación $1 + G(s)H(s) = 0$. De esta forma, se obtiene la respuesta temporal del sistema en lazo cerrado a partir de su función en lazo abierto, $G(s)H(s)$.

Ambas funciones son complejas, por lo que se puede descomponer esta ecuación en dos, la primera correspondiente al argumento

$$|G(s) \cdot H(s)| = 1 \quad (2.52)$$

y la segunda, al módulo

$$\angle G(s) \cdot H(s) = \pi(2n + 1) \quad n = 0, \pm 1, \pm 2, \pm 3, \dots \quad (2.53)$$

Comprobando si estas dos ecuaciones se cumplen para cada uno de los puntos del plano se obtiene el lugar de las raíces del polinomio del denominador y, por lo tanto, el lugar de los polos del sistema y, posteriormente, su respuesta temporal. Realizar el cálculo para todos los puntos del plano pueden suponer un tiempo muy elevado, afortunadamente existen ciertas reglas que simplifican esta tarea significativamente:

1. Regla 1. Cada rama del LDR comienza en un polo del sistema en lazo abierto, con una ganancia nula, y termina en un cero cuando la ganancia es infinita. Por lo tanto, el número de ramas que existen en el LDR es igual al número de polos del lazo abierto, pero, además, si existen menos ceros que polos el un número de ramas es igual a la diferencia entre ceros y polo terminarán en el infinito.
2. Regla 2. Un punto del eje real pertenecerá al LDR si el número de polos y ceros a su izquierda, en el dominio de Laplace, es un número impar.
3. Regla 3. El LDR es simétrico.
4. Regla 4. La ramas del LDR que terminan en el infinito presentan un ángulo respecto del eje real

$$\theta = \frac{(2n + 1) \cdot \pi}{n - m} \quad n = 0, \pm 1, \pm 2, \pm 3, \dots \quad (2.54)$$

Con n el número de polos en el lazo abierto y m el número de ceros. Además, estas asíntotas cortan al eje real en el semieje negativo, en el punto:

$$\sigma = \frac{\sum \text{parte real polos lazo abierto} - \sum \text{parte real ceros lazo abierto}}{n - m} \quad (2.55)$$

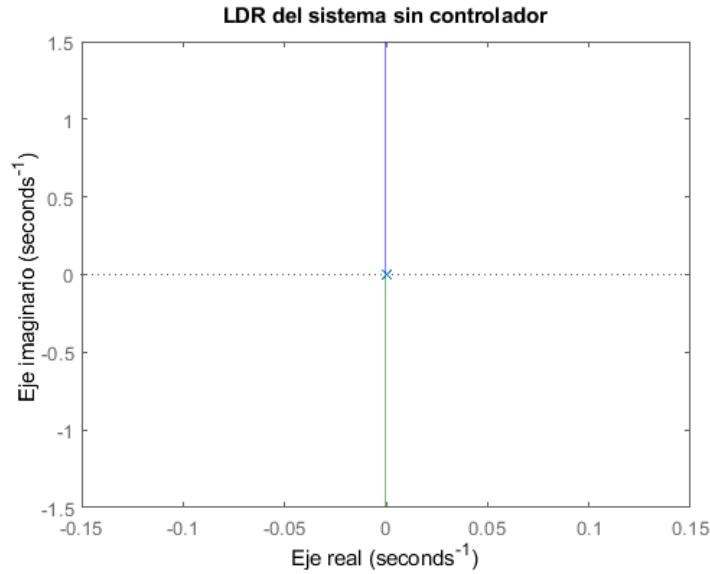


Figura 2.6: Lugar de las raíces del sistema sin controlador.

Finalmente, después de seguir todas las reglas y haber obtenido el lugar de las raíces de forma gráfica, se puede obtener el valor de k aplicando el criterio del módulo a un punto cualquiera:

$$1 + k|G(s) \cdot H(s)| \quad \Rightarrow \quad k = \frac{1}{|G(s) \cdot H(s)|} = \frac{\prod |s + p_i|}{\prod |s + z_i|} \quad (2.56)$$

Una vez definida la forma en la que se realiza el LDR, se realiza el LDR del sistema a controlar, sin añadir ningún tipo de controlador externo. Para ello, siguiendo las reglas definidas anteriormente se observa el LDR obtenido en la Figura 2.6. En esta figura, se puede apreciar que el sistema es críticamente estable, es decir, que ante una entrada escalón oscilará de forma indefinida.

Teniendo en cuenta todo lo que se ha comentado sobre el LDR y su importancia para el control de los sistemas, en este sistema es necesario simplemente añadir un cero en lazo abierto de forma que se obtenga un sistema estable en lazo cerrado. Por lo tanto, el controlador más sencillo que se puede implementar deberá, al menos, tener un cero que permita realizar este ajuste.

2.5. Diseño de controladores

En este TFM se aplican dos teorías de control, por lo que va a ser necesario el uso de dos o más tipos de controladores. En esta sección se definirán los controladores que serán implementados en el sistema junto con las técnicas que son utilizadas para su obtención.

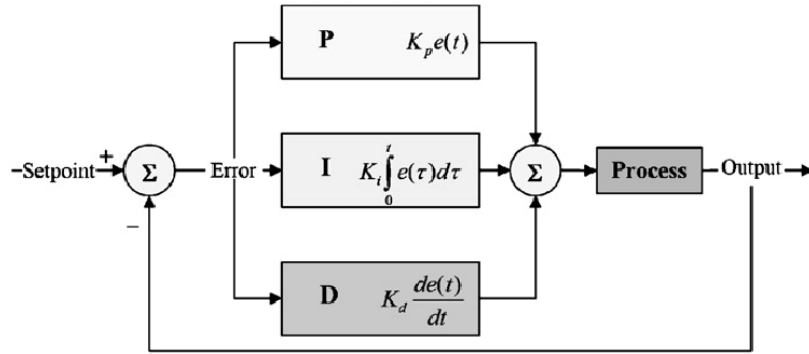


Figura 2.7: Sistema realimentado con controlador PID. Researchgate.

2.5.1. Controlador Proporcional-Derivativo-Integral

Como se comentó en el Capítulo 1 este tipo de controlador es descubierto en los inicios de la ingeniería de control, cuando el objetivo fundamental era el estudio y control de sistemas inestables LTI y SISO. Este controlador es el más utilizado en la industria, ya que existen métodos de sintonización automática sin necesidad de realizar un análisis analítico del sistema.

Un sistema controlado por PID basa su funcionamiento en la realimentación del error del sistema, esto es, la diferencia entre la salida actual del sistema y la consigna a la que se quiere llevar dicho sistema. Lo que realiza el controlador PID es proporcionar cierta ganancia a esta señal de error con el objetivo de modificar el comportamiento del sistema. Este controlador está formado por tres componentes: la componente proporcional, que es, como su nombre indica, una señal proporcional al error, la componente integral, que es proporcional a la integral del error, y la componente derivativa, que es proporcional a la derivada del error. Con la suma de estas tres componentes se obtiene la acción de control con la que se alimenta al sistema.

$$U(s) = [K_p + K_d s + K_i \frac{1}{s}] \cdot E(s) \quad (2.57)$$

En la Ecuación 2.58 se observa la forma que tiene este controlador en el dominio de Laplace. Desarrollando esta ecuación, se puede obtener la función con la que se realiza la sintonización analítica de este controlador por medio del LDR. En esta ecuación se observa que el controlador añade al sistema dos ceros ajustables y un polo en el origen.

$$G_c(s) = \frac{U(s)}{E(s)} = K_p + K_d s + K_i \frac{1}{s} = \frac{K_d s^2 + K_p s + K_i}{s} \quad (2.58)$$

Este tipo de controlador proporciona dos ceros ajustables mediante las ganancias del controlador y un polo en el origen. Para realizar un ajuste exacto del controlador, habiendo definido previamente una serie de especificaciones que se desean cumplir,

es necesario realizar un análisis extenso del LDR con el objetivo de hallar la posición en la que se deberían colocar los polos del sistema realimentado. Una vez realizada esta tarea, simplemente habría que definir la ecuación de segundo grado que satisface esas condiciones. A continuación, el paso para obtener las ganancias de los distintos componentes del controlador es inmediato. Normalmente este análisis se realiza con un programa de computación por ordenador como MATLAB, pero es importante que los ingenieros que realizan este tipo trabajos sean conscientes de como se realiza de forma analítica.

2.5.1.1. Sintonización del controlador mediante el LDR

Realizar un análisis del LDR y sintonizar las variables del controlador no es una tarea simple, sobre todo cuando se trabaja con un controlador PID completo. Por este motivo, para facilitar esta tarea y debido a que el sistema no requiere un controlador complejo para ser estable, se realizará la sintonización de un controlador PD, que aporta un cero ajustable. Este controlador tiene la siguiente función de transferencia:

$$G_c(s) = K_p + K_d s = K_p(1 + \frac{K_d}{K_p} s) = K_d(s + \frac{K_p}{K_d}) \quad (2.59)$$

Obteniendo así un cero real, en $\frac{K_p}{K_d}$, ajustable y una ganancia, K_d , también ajustable. En este caso, presenta dos parámetros que deben ser ajustado para conseguir las especificaciones deseadas.

Partiendo de dos de los parámetros más importantes para describir la respuesta temporal de un sistema, esto es, el tiempo de establecimiento y la sobreoscilación, se obtiene la posición del polo dominante del sistema realimentado.

$$t_s = \frac{\pi}{\sigma} \rightarrow \sigma = \frac{\pi}{t_s} \quad (2.60)$$

Teniendo la posición en el eje real del polo, con la sobreoscilación se obtiene la posición del polo en el eje imaginario:

$$M_p = e^{\frac{-\sigma\pi}{\omega_d}} \rightarrow \omega_d = \frac{\sigma\pi}{\ln(\frac{1}{M_p})} = \frac{\pi^2}{\ln(\frac{1}{M_p}) \cdot t_s} \quad (2.61)$$

Aplicando el criterio del ángulo, se obtiene la posición, a , en la que se debe colocar el cero para que el polo dominante se encuentre en el LDR.

$$\sum \angle \text{ceros} - \sum \angle \text{polos} = \tan^{-1}\left(\frac{\omega_d}{\sigma - a}\right) - 2\tan^{-1}\left(\frac{\omega_d}{\sigma}\right) = \pi(2n+1) \quad n = 0, \pm 1, \pm 2, \dots \quad (2.62)$$

Resolviendo, para despejar a , se obtiene:

$$a = [\tan(\pi(2n + 1) + 2\tan^{-1}(\frac{\omega_d}{\sigma})) \cdot \omega_d] + \sigma \quad n = 0, \pm 1, \pm 2, \pm 3, \dots \quad (2.63)$$

Una vez hallada la posición del polo, es necesario aplicar el criterio del módulo para obtener el valor de la ganancia necesaria:

$$|K| = \frac{\prod(s + p_i)}{\prod(s + z_j)} = \frac{\sqrt{\omega_d^2 + \sigma^2} \cdot \sqrt{\omega_d^2 + \sigma^2}}{\sqrt{\omega_d^2 + (\sigma - a)^2}} \quad (2.64)$$

Finalmente, habiendo obtenido la posición del polo, a , y la ganancia que debe tener el sistema en el lazo abierto, K , en función de la posición del polo que previamente se ha obtenido y a partir de las especificaciones deseadas, se calculan los dos parámetros del controlador:

$$K_d = \frac{K}{7} \quad (2.65)$$

$$-\frac{K_p}{K_d} = a \rightarrow K_p = -K_d \cdot a = -\frac{K \cdot a}{7} \quad (2.66)$$

El factor $\frac{1}{7}$ es a causa de la ganancia que presenta la función de transferencia del sistema. Hay que tener en cuenta que no se está calculando la ganancia del controlador si no la ganancia del lazo abierto, que incluye las funciones de transferencia del controlador y del sistema. Por otro lado, hay que tener en cuenta que para sistemas con respuesta estable los polos se encuentran en el semieje negativo. Por este motivo, para obtener un valor de ganancia positiva se ha añadido el signo negativo en la Ecuación 2.66.

2.5.2. Regulador cuadrático-lineal (LQR)

A los ingenieros de control, después de controlar de forma exitosa sistemas que por su naturaleza eran inestables, les surge el deseo de optimizar estos controladores [7]. Inicialmente, la labor de investigación se centró en conseguir controladores que fuesen óptimos desde algún punto de vista, por ejemplo, el de minimizar las variaciones en la acción de control, de esta forma se puede reducir el consumo de los dispositivos o incluso abaratando su coste, ya que requerirían menor potencia para operar, por lo tanto se podrían fabricar con componentes de menor precio.

A esta rama de la ingeniería de control se le conoce como teoría de control óptimo, y el controlador más utilizado es el regulador lineal cuadrático, o *linear-quadratic*

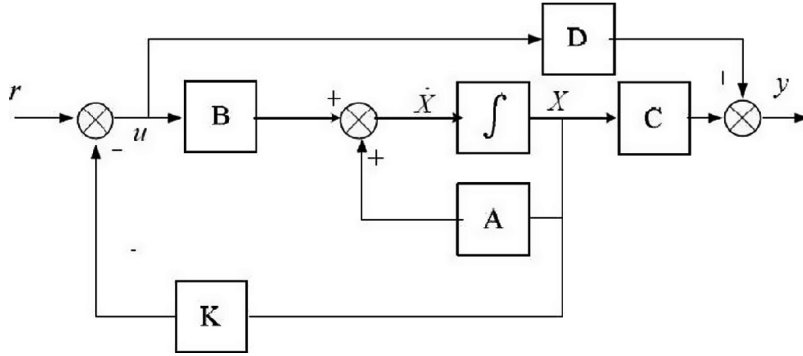


Figura 2.8: Sistema realimentado con controlador LQR. Springer.

regulator (LQR) en inglés. De forma similar al controlador PID, este controlador se basa en la realimentación, pero en este caso no del error, si no de las variables de estado.

Como se observa en la Figura 2.8 la realimentación se produce restando a la señal de consigna, t , el valor de las variables de estado, X , con una cierta ganancia, K . El ajuste de esta matriz de realimentación es la que permite realizar un control óptimo sobre la señal de salida, y , o la de control, u .

Como se ha comentado, esta rama de la ingeniería de control se basa en el diseño de controladores óptimos, en los que no sólo se realiza un control de la salida si no también de la entrada. Por este motivo, se define la función de coste que se desea optimizar:

$$J = \int_0^{\infty} x^T(t) \cdot Q \cdot x(t) + u(t)^T \cdot R \cdot u(t) dt \quad (2.67)$$

En esta ecuación se observa que tanto la entrada, $u(t)$, como los estados, $x(t)$, están presentes. Las matrices R y Q corresponden con el coste de la acción de control y los estados respectivamente, de esta forma, modificando sus valores se pueden obtener diferentes comportamientos del sistema.

Para optimizar esta acción de control se realiza la realimentación de los estados. De esta forma, se obtiene la acción de control, en la que se omite la referencia(r) del sistema. Esto es debido a que en régimen permanente éste presenta un error nulo, independiente de la señal de entrada. De no ser así, la función de coste tendería a infinito, lo cuál no sería una solución óptima.

$$u(t) = -Kx(t) \quad (2.68)$$

Por lo que se obtiene un nuevo modelo del sistema, definido por:

$$\dot{X} = AX - B \cdot KX = (A - B \cdot K)X \quad (2.69)$$

Además, sustituyendo la Ecuación 2.68 en la Ecuación 2.67, se obtiene la función de coste que se pretende optimizar:

$$J = \int_0^\infty x^T(t)Qx(t) + x^T(t)K^T RKx(t) dt = \int_0^\infty x^T \cdot (Q + K^T RK) \cdot x dt \quad (2.70)$$

Siendo P una matriz simétrica real o hermética definida positiva que cumple la siguiente igualdad para todos los valores de x , es decir, para cualquier estado del sistema:

$$J = x^T Px \quad (2.71)$$

Desarrollando esta integral, se obtiene la siguiente ecuación:

$$x^T(Q + K^T RK)x = -\frac{d}{dt}(x^T Px) \quad (2.72)$$

Desarrollando esta ecuación, se obtiene:

$$x^T(Q + K^T RK)x = -\dot{x}^T Px - x^T P\dot{x} = -x^T[(A - BK)P + P(A - BK)]x \quad (2.73)$$

Por tanto, se obtiene la ecuación que minimiza la función de coste:

$$(A - BK)^T P + P(A - BK) = -(Q + K^T RK) \quad (2.74)$$

De esta forma, resolviendo la Ecuación 2.74 se obtiene un controlador que satisfaga el problema del control óptimo. Para obtener esta solución se define una matriz R simétrica real o hermética definida positiva, tal que:

$$R = T^T T \quad (2.75)$$

Tomando esto en cuenta, se reescribe la Ecuación 2.74 de la siguiente forma:

$$(A^T - K^T B^T)P + P(A - BK) + Q + K^T T^T T K = 0 \quad (2.76)$$

Desarrollando y reorganizando esta ecuación, se obtiene:

$$A^T P + P A + [T K - (T^T)^{-1} B^T P^T]^T [T K - (T^T)^{-1} B^T P] - P B R^{-1} B^T P + Q = 0 \quad (2.77)$$

Si uno de los objetivos es la minimización de J respecto de K , es necesario realizar la minimización de

$$x^T [T K - (T^T)^{-1} B^T P^T]^T [T K - (T^T)^{-1} B^T P] x \quad (2.78)$$

con respecto a K . Se observa que en esta ecuación se produce un mínimo en cero, cuando

$$K = T^{-1} (T^T)^{-1} B^T P = R^{-1} B^T P \quad (2.79)$$

En consecuencia, una vez obtenida la solución deseada, se sustituye en la Ecuación 2.68 para obtener la acción de control que requiere el sistema:

$$u(t) = -K x(t) = -R^{-1} B^T P x(t) \quad (2.80)$$

Además, la matriz P debe satisfacer la Ecuación 2.76, con lo que sustituyendo en esta la Ecuación 2.79, se obtiene:

$$A^T P + P A - P B R^{-1} B^T P + Q = 0 \quad (2.81)$$

Esta ecuación recibe el nombre de ecuación reducida de Ricatti. Con esta ecuación, los pasos que hay que seguir para diseñar un controlador LQR son sencillos. Inicialmente es necesario resolver la ecuación de Ricatti para obtener el valor de P que satisfaga las especificaciones que se han definido a través de las matrices Q y R . Posteriormente, sustituyendo esta matriz en la Ecuación 2.80 se obtiene la matriz K con la que se realimenta el sistema para obtener la respuesta deseada.

2.5.2.1. Ajuste empírico del controlador LQR

En el apartado anterior se ha mencionado que las matrices Q y R son fundamentales para la selección de la matriz de realimentación de estados, K , que son necesarios para realizar el control del sistema. Sin embargo, no se ha definido cuales son las implicaciones de modificar estas matrices. Aunque resulta intuitivo observando la Ecuación 2.67, la función que desempeñan estas matrices es fundamental para diseñar el controlador LQR.

Estas matrices están directamente relacionadas con el peso que se le proporciona a cada uno de los estados y a la acción de control en la función de coste. Es decir, mediante un ajuste de estas matrices se modifica la acción del sistema. Por ejemplo, para reducir de forma considerable la acción de control, reduciendo así el coste energético o, incluso, dando la posibilidad de hacer uso de actuadores menos potentes.

Con este controlador se consigue adaptar la respuesta del sistema a las características físicas de éste. Por ejemplo, permite diseñar directamente un controlador que trabaje en la zona de lineal de los sistemas y, por lo tanto, que la respuesta no se vea afectada por la saturación de los actuadores o de los sensores, entre otros efectos. Esto es una gran ventaja, ya que permite modificar el controlador en función del estado de los dispositivos que conforman el sistema.

Una gran desventaja de este tipo de controladores es que no existe un método analítico con el que seleccionar las matrices Q y R . Existen técnicas comúnmente utilizadas para el ajuste de estas matrices, como por ejemplo, definir R como una matriz identidad y $Q = C^T C$. Pero el método más adecuado es el uso de herramientas de simulación como MATLAB, que permiten observar el comportamiento del sistema realimentado y el ajuste de estos valores de forma que la respuesta de este sea la deseada.

Capítulo 3

Descripción del *hardware* del sistema

Este capítulo detalla los componentes por los que está formado el sistema, desde los sensores encargados de realizar la obtención de la posición de la esfera hasta el sistema encargado de realizar el movimiento de la estructura, pasando por el microcontrolador, encargado de realizar todas las operaciones asociadas al control del sistema.

3.1. Estructura mecánica

Como ya se ha comentado anteriormente, en este TFM se realiza el control de una estructura de dos grados de libertad.



Figura 3.1: Fotografía de la estructura mecánica del sistema.

Se observa en la Figura 3.1 que esta estructura está formada por una bandeja, en la que se posicionará la esfera, y cuyo ángulo respecto del plano horizontal es controlado por dos motores eléctricos, uno para cada eje. Como se ha visto en el Capítulo 2, el control se realiza sobre el ángulo de la plataforma, pero el ángulo sobre el que realmente se puede actuar, es el ángulo de los motores. Por este motivo, es necesario realizar un ajuste previo al control. Para analizar la relación entre estos dos ángulos se ha realizado un modelo mecánico utilizando el módulo Simscape de Simulink.

Esta herramienta permite definir la estructura del sistema en un entorno de simulación, y realizar un análisis detallado de la relación de ambos ángulos. En la Figura 3.2 se muestra la relación que existe entre estos dos ángulos, además de una estimación polinómica de esta.

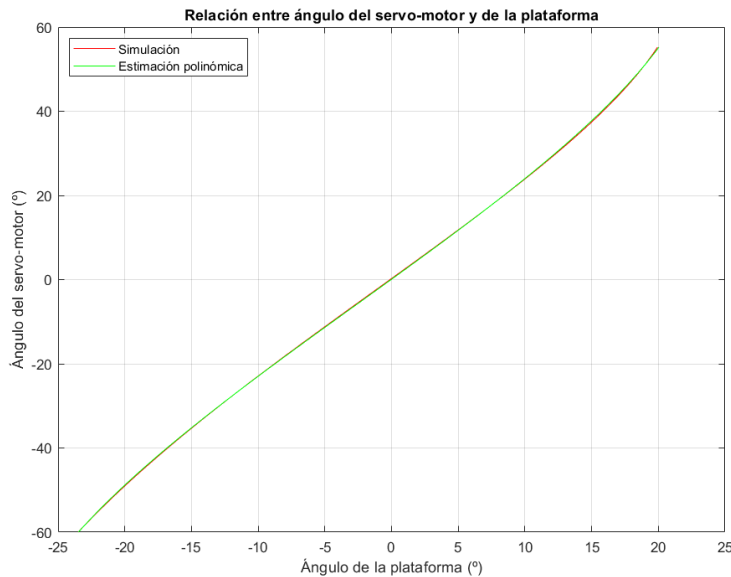


Figura 3.2: Relación entre ángulo del servo-motor y plataforma.

A partir de esta simulación, se ha realizado el análisis y control directamente sobre la plataforma, realizando un abstracción del mecanismo por el cuál se obtiene esta inclinación. De esta forma, permite realizar un modelado más sencillo, y por lo tanto, reducir la complejidad asociada al diseño del controlador.

3.2. Sensores

Una vez visto el sistema mecánico, es necesario realizar la selección de los componentes electrónicos del sistema. En este TFM es necesario realizar el control de la posición de una esfera. Por lo tanto, existe la necesidad de componentes capaces de proporcionar estos datos. Se ha decidido que para cumplir esta función se hará uso de dos sistemas, uno basado en una pantalla resistiva y otro en el uso de una cámara.

3.2.1. Pantalla resistiva

En este caso, este sistema de medida consiste un cristal en el que se encuentran integrados cuatro hilos conductores en los lados. Como se puede observar en la Figura 3.3, mediante la alimentación de los hilos superior e inferior se puede realizar la medida de la tensión en uno de los conductores laterales. De estado forma, obtiene la posición, en el eje vertical, del objeto que se encuentre en contacto con la pantalla. De forma análoga, se obtiene la posición en el eje horizontal, alimentando los hilos laterales y midiendo la tensión en uno de los dos hilos restantes.

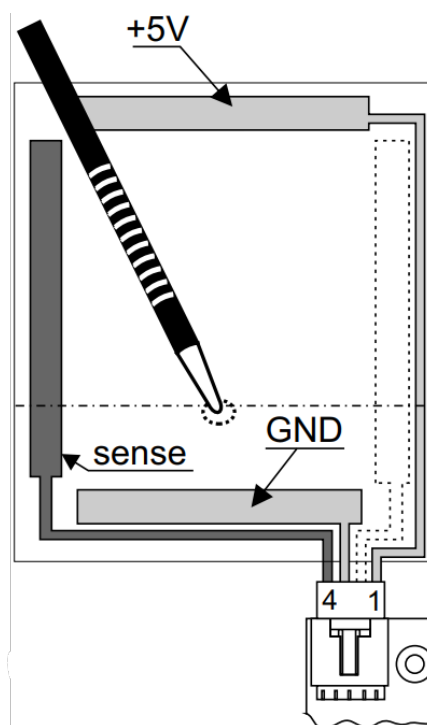


Figura 3.3: Esquema del circuito resistivo de la pantalla táctil. Microchip.

Este sistema presenta varias desventajas, la primera, es que es necesario disponer de un controlador que gestione la alimentación y medida en cada uno de los ejes, la segunda tiene relación con el periodo mínimo de muestro que se puede alcanzar, dependiente de las capacidades parásitas del sistema y, por último, sólo debe haber un objeto en contacto con el cristal, si no es así, la medida que se obtendría sería errónea.

En este caso, el único problema que provoca un efecto negativo es el primero, debido a que es necesario un sistema encargado de tomar las medidas del sensor. Para solucionar esto, existen dos alternativas, la primera es adquirir directamente un controlador para pantallas resistivas y la segunda consiste en realizar el circuito necesario para la alimentación del sensor y realizar el control directamente en el microcontrolador. La única ventaja que presenta la implementación propia del

controlador, es que permitiría obtener un período de muestreo más reducido que con la primera, a costa de eliminar la comunicación entre los dos sistemas y aumentar la carga computacional del microcontrolador. Debido a la complejidad asociada a esta solución se ha decidido adquirir directamente un controlador comercial para pantallas resistivas.

El sistema encargado de tomar las mediadas de la pantalla y transmitir las al microcontrolador será el controlador de pantallas resistivas de la serie AR1000 de Microchip. Este controlador permite el control de pantallas de hasta 8 hilos conductores con un tiempo de muestro típico de 7ms. Además, dispone de comunicación por I2C, SPI y/o UART.

3.2.2. Cámara

Se ha decidido implementar un segundo método para la detección de la posición de la esfera. Este método consiste en el uso de una cámara que se encuentra posicionada sobre la estructura. Para realizar la detección de la esfera, es necesario aplicar algoritmos de visión artificial. En este caso, al ser el objeto detectado una esfera, que en las imágenes obtenidas se muestra como un objeto en 2D, es necesario el uso de métodos para la detección de círculos.



Figura 3.4: Fotografía de la cámara OpenMV Cam H7. OpenMV.

Este algoritmo se basa en la transformada de Hough para la detección de geometrías en imágenes. La identificación de los bordes, o de geometrías, en una imagen se puede describir como la variación de la intensidad en una orientación particular. Cuanto mayor sea esta variación, mayor será la posibilidad de que se

corresponda con un borde de un objeto. Por lo tanto, matemáticamente, se puede definir esta variación como el gradiente de la intensidad en una dirección particular. Teniendo esto en consideración, y simplificando, se define que la transformada de Hough se basa en el análisis del gradiente de la intensidad en una dirección en particular. Por lo tanto, para facilitar la labor del algoritmo, es necesario que exista gran contraste entre los objetos a detectar y el fondo de la imagen, en este caso, entre el color de la esfera y de la plataforma.

De forma similar al caso anterior, existe dos posibles alternativas para la implementación de este sistema de medida, una en el que nuestro microcontrolador es el encargado de realizar la adquisición y procesamiento de las imágenes y, otra en la que un controlador auxiliar es el encargado exclusivamente de esta tarea. A diferencia del caso anterior, no cabe duda que de el método por el que se ha optado es el segundo. Esto se debe a que para realizar la implementación de la primera solución es necesario realizar un trabajo extenso que se sale de los objetivos y alcance de este TFM, que incluye el uso de controladores mucho más potentes y de algoritmos de visión artificial complejos. Con todo esto, se ha decidido que el controlador y cámara a utilizar se encuentran integrados en el sistema comercial OpenMV Cam H7.

Este sistema dispone de un microcontrolador ARM Cortex-M7 que funciona a una frecuencia de 408MHz con 1MB de RAM y 2MB de memoria *flash*. Este microcontrolador dispone de un bus USB, de hasta 12Mb/s, I2C, de hasta 1Mb/s, UART, de hasta 7.5Mb/s, y SPI, de hasta 80Mb/s. Este controlador es el encargado de obtener y procesar las imágenes obtenidas de un sensor óptico OV7725 capaz de proporcionar imágenes en escala de grises de hasta 640x480 píxeles con una velocidad de 75 fotogramas por segundo. Este sistema permite la ejecución de programas en el lenguaje de programación de alto nivel *MicroPython*, lo que facilita la implementación de algoritmos de detección de objetos.

3.3. Actuadores

Para realizar el control de un sistema es fundamental disponer de instrumentos de medida que permitan conocer el estado actual del sistema, pero es igual de importante el uso de dispositivos que sean capaces de alterar su estado de la forma en la que se desee en cada momento. Por este motivo, la selección de actuadores no es trivial y es necesario dedicar tanto o más tiempo que a la selección de sensores.

Por las características de este TFM, es necesario el uso de motores eléctricos, preferiblemente capaces de realizar un ajuste preciso del ángulo. Por este motivo, y como se ha podido apreciar anteriormente, para el control de la estructura se hará uso de motores de corriente continua que permiten un control directo sobre su ángulo de actuación. A este tipo de actuadores comúnmente se les conoce como servo-motores.

Este tipo de actuadores integran un controlador que permite realizar un control



Figura 3.5: Fotografía de un servo-motor. Bricogeeek.

preciso de su posición sin necesidad de lógica de control externa. Una característica, fundamental para este proyecto, es que presentan un par motor elevado, que, junto con un control preciso de su ángulo, los hacen especialmente útiles en tareas de control.

3.4. Microcontrolador

El microprocesador será el encargado de gestionar la comunicación con los sensores y actuadores, del procesamiento de las señales que le llegan de los sensores y del cálculo de la acción de control necesaria en cada momento. Teniendo en cuenta estas tareas, es necesario que el procesador disponga al menos de un periférico de comunicación (UART, I2C o SPI) junto dos temporizadores para la gestión de las señales de tipo PWM necesarias para el control de los actuadores. A parte de estos requisitos, es deseable que el microprocesador seleccionado disponga de unas prestaciones suficientes, de cómputo y de memoria, para poder ejecutar un sistema operativo en tiempo real, algo que facilita la integración de todos los componentes y permitirá la implementación de los controladores.

Por lo motivado comentado anteriormente se ha decidido que el microprocesador que se utilizará en el desarrollo del proyecto es el ARM Cortex-M4, ya que proporciona una solución de bajo coste cumpliendo los requisitos del proyecto. Para ello, la implementación del sistema se realizará haciendo uso de la plataforma de desarrollo STM32L476RG que integra el microcontrolador seleccionado, y que se encuentra disponible para su uso en el Laboratorio de Robótica y Control.

Este microcontrolador funciona a una frecuencia máxima de 80MHz e incluye una unidad de coma flotante. Dispone de 128KB de RAM y 1MB de memoria *flash*, además, incluye periféricos de comunicación I2C, SPI, UART, junto con una interfaz de programación y depuración a través de USB.

Además de lo mencionado anteriormente, este microprocesador incluye un regula-

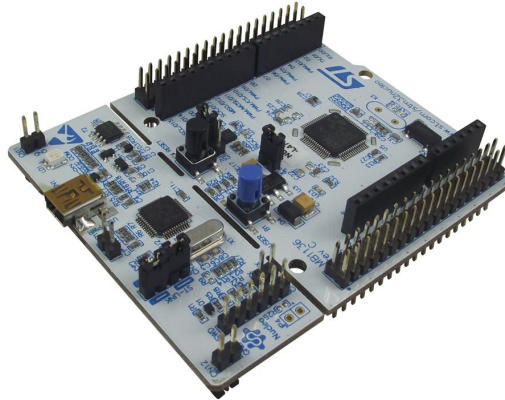


Figura 3.6: Fotografía de la placa de desarrollo. STMicroelectronics.

dor de voltaje que, junto con la posibilidad de regular la frecuencia de funcionamiento, permite el uso de siete modos de potencia. Esta funcionalidad lo hace idóneo para la implementación en tareas en las que el consumo es un factor clave en el diseño del sistema.

3.5. Desarrollo de la tarjeta electrónica

En las secciones anteriores se han descrito los componentes que se van a utilizar en la implementación del sistema. Para la integración de todos estos elementos en un sistema único se ha desarrollado una tarjeta de expansión, que además, debe ser compatible con la tarjeta de desarrollo utilizada. En esta tarjeta de expansión se encontrarán tanto los componentes electrónicos como los conectores a los dispositivos externos, como los servo-motores o la cámara.

Para el diseño de la tarjeta de expansión se ha utilizado el entorno *Altium Designer*. Este programa permite el diseño de componentes, la captura de esquemáticos y diseño de circuitos impresos. Incorpora, en una sola aplicación, un sistema que permite la gestión del entorno de desarrollo y la comunicación del usuario con los servidores que contienen la información sobre los componentes electrónicos. Se ha decidido utilizar este programa debido a que es uno de los programas más utilizados en la industria electrónica por su amplia funcionalidad y comodidad, además, se ha utilizado en una de las asignaturas del Máster con lo que se conoce el funcionamiento del programa.

3.5.1. Realización de esquemáticos

Antes de comenzar con el diseño del circuito impreso es necesario definir los elementos del sistema y la relación que existen entre ellos. Esto se realiza mediante el desarrollo de esquemáticos. Generalmente, se agrupan los componentes del sistema en bloques por su funcionalidad y se realiza un esquemático global en el que se definen

las relaciones que existen entre los bloques. Este tipo de esquemáticos se les conoce como diseño jerárquico. En este caso, esto los bloques en los que se divide el sistema son tres: sensores, actuadores y controlador.

Dado que este sistema es sencillo, la utilización de esquemáticos jerárquicos no es necesaria. Por ello, se ha decidido que todo el sistema se incluya en un sólo esquemático, con el objetivo de que se pueda observar de forma más clara y sencilla la relación entre los distintos componentes. El esquemático obtenido se muestra en el Anexo A.

A continuación, se detallan las decisiones de diseño y los motivos por los que se han tomado:

- Tamaño de los componentes pasivos: se ha decidido que el tamaño de los componentes pasivos que se van a utilizar en esta tarjeta sea de tipo 0805, equivalente a 1.3 mm x 2.1 mm. Dado que el circuito impreso que se realizará dispone de un tamaño suficiente, no es necesario el uso de componente de tamaños más reducidos. Además, este encapsulado permite la soldadura de los componentes a mano.
- Bloque de control de la pantalla resistiva: para el diseño de este bloque se ha seguido el esquema que proporciona el fabricante[8]. Este diagrama presenta todos los elementos que son necesarios para el correcto funcionamiento del componente.
- Selección de la estructura: con el objetivo de realizar una tarjeta de expansión que sea capaz de funcionar tanto con el sistema formado por la cámara como por el que dispone de la pantalla resistiva, se ha decidido incluir un conector que permite seleccionar con qué sensor trabajar en cada momento.
- Configuración del modo de comunicación: dado que el controlador de la pantalla resistiva permite la comunicación por dos protocolos, I2C y SPI, se ha decidido añadir pines de configuración que permiten seleccionar el tipo de comunicación empleado.
- Control de servo-motores: para el control de los actuadores es necesario añadir un componente adicional que además de reducir el consumo, aporta seguridad al sistema. Con el objetivo de poder desactivar los servo-motores cuando no sea necesaria su actuación, se han añadido dos *buffers* que interrumpen la señal de control de los servo-motores.

3.5.2. Posicionamiento de los componentes

Una vez se han realizado los esquemáticos, es momento de posicionar los componentes en la placa de circuito impreso (*Printed Circuit Board*, PCB). Para ello hay que tener en cuenta varias consideraciones:

- Longitud de las pistas: con el objetivo de reducir la impedancia de las pistas, es necesario realizar un posicionamiento de tal forma que los componentes con gran relación entre sí presenten una distancia reducida.
- Conectores externos e interfaz de usuario: hay que tener en cuenta como se produce la conexión de los conectores al realizar el posicionamiento. En este caso, esta conexión se realiza con conectores verticales, es decir, la parte superior de estos debe encontrarse en todo momento libre. Además, hay que tener en cuenta que la tarjeta de expansión diseñada debe ser compatible con el uso de tarjetas adicionales. Por lo tanto, es necesario realizar un posicionamiento de los conectores de tal forma que permita su conexión cuando se encuentran múltiples tarjetas de expansión conectadas a la placa de desarrollo. De forma similar, todos los elementos que permitan la comunicación del usuario con nuestro sistema deberán estar en un lugar de la PCB accesible.
- Condensadores en la alimentación: la función de los condensadores en la alimentación de los integrados es la de eliminar el ruido de la fuente de alimentación y de proporcionar alimentación cuando se requieren pequeños picos de corriente. Por estos motivos, es importante que los condensadores se encuentren lo más cerca posible de la alimentación.
- Componentes digitales y analógicos: con el objetivo de reducir el ruido en las pistas analógicas, es recomendable separar los componentes que trabajen con señales analógicas de los digitales. En este caso, estas señales analógicas provienen únicamente de la pantalla resistiva, por lo que se debe posicionar los componentes de tal forma que las pistas digitales se encuentren lo más lejos posible de estas.

3.5.3. Rutado del circuito impreso

Una vez realizado el posicionamiento de los componentes siguiendo las consideraciones comentadas anteriormente, es momento de realizar el rutado de las pistas de la PCB. De forma similar al posicionamiento, en esta tarea también hay que tener en cuenta varias cosas:

- Planos de masa: el uso de planos de masa permite reducir el ruido electromagnético además de un camino continuo y de baja impedancia de retorno de la corriente.
- Distribución de las pistas de potencia y separación de pistas digitales y analógicas: por los motivos comentados anteriormente, es recomendable realizar una distribución de las pistas de potencia de forma que se realice una separación entre la etapa de potencia de componentes analógicos y digitales. De forma similar, es muy recomendable realizar una separación entre las pistas de señal analógica y digital.

- *Via stitching*: consiste en realizar agujeros en toda la superficie de la placa manteniendo una cierta distancia constante entre ellos. Esta técnica permite reducir el efecto de señales de alta frecuencia en los circuitos, pero, además, crea puntos de interconexión entre los planos de masa, en el caso de que la PCB disponga de varios planos distribuidos.
- Señales digitales de alta frecuencia: este tipo de señales, por su gran frecuencia de conmutación, generan mucho ruido electromagnético. Para reducir este efecto, que puede ser muy perjudicial para algunos componentes, es recomendable reducir lo máximo posible la longitud de estas pistas. Además, si es posible, existe la posibilidad de aislar estas pistas entre un plano de masa.

Después de haber realizado el posicionamiento de los componentes en la PCB y el rutado de las pistas, es necesario comprobar que el diseño cumple con todos los requisitos para la fabricación. Las especificaciones de la PCB quedan definidas por las resoluciones y tolerancias de las que disponen las máquinas de los fabricantes. Estos requisitos generalmente se encuentran en la página web de los fabricantes, por ejemplo, las tolerancias que permite la empresa JLCPB se encuentran en la sección de Capacidades de su página web[9]. En el diseño realizado, dado que se trata de un sistema sencillo, estas restricciones no suponen un problema importante, pero, en cualquier caso, hay que realizar un diseño que las satisfaga, ya que si no se arriesga a que la PCB no se pueda fabricar. Por tanto, para el diseño de esta PCB se han seguido las especificaciones del fabricante *JLCPB* mencionadas anteriormente.

Con todos estos comentarios, recomendaciones y restricciones, se ha realizado el diseño del circuito impreso. Se puede apreciar la PCB obtenida en el Anexo A.

Capítulo 4

Diseño e implementación del sistema

En este Capítulo se describirá el software que se ha empleado en el TFM junto con las alternativas y decisiones tomadas para la implementación del sistema.

4.1. Descripción del *software* empleado

4.1.1. Sistema operativo en tiempo real: FreeRTOS

Un sistema operativo es un programa que permite al usuario gestionar operaciones básicas de un ordenador, además de proporcionar servicios a otros programas o aplicaciones que se encuentren ejecutando en la máquina. Estos servicios permiten la codificación de aplicaciones de forma más rápida, simple y fácil de mantener.

La mayoría de sistemas operativos permite una ejecución simultánea de tareas, a esto se le denomina multi-tarea. En realidad, un procesador sólo permite la ejecución de un hilo, pero esta idea de paralelismo es proporcionada por el planificador, que es el encargado de decir que tarea es ejecutada en cada momento. El tipo de sistema operativo queda definido por la forma en la que tiene el planificador de dividir el tiempo del microprocesador entre las tareas.

Los sistemas operativos en tiempo real (RTOS, *Real Time Operative System*) son los que presentan una traza de ecuación predecible, esto es, en este tipo de sistemas se puede asegurar que el programa responderá a un determinado evento en un tiempo determinado. Este tipo de sistemas operativos son comunes en los sistemas embebidos, ya que permiten realizar tareas que presenten restricciones temporales, pudiendo asegurar con certeza que se cumplen estas.

Generalmente, en estos sistemas operativos se consigue esta predictibilidad debido

a que el planificador permite que el programador asigne prioridades a las tareas, con lo que la ejecución se realizará en orden de mayor a menor prioridad.

Uno de los sistemas operativos en tiempo real más usado es FreeRTOS. Este sistema operativo está especialmente diseñado para ser ejecutado en cualquier tipo de microprocesador, ya que proporciona un código fuente muy ligero, pero a su vez cumple con los requisitos necesarios de los sistemas operativos en tiempo real más sofisticados.

Algunas de las características relevantes para la realización de este TFM son: disponer de un planificador en tiempo real, siguiendo el modelo de planificación *Round-Robin* con prioridades, necesario para realizar un análisis de planificabilidad del sistema; permitir la comunicación entre Tareas, esto resulta muy útil debido a que permite dividir en tareas cada una de las funciones del sistema.

Realizar la implementación sobre un sistema operativo, independientemente de que sea en tiempo real o no, permite abstraer el código realizado de la arquitectura empleada, además de facilitar la escalabilidad y la portabilidad.

4.1.2. Entorno de desarrollo: STM32CubeIDE

Para la realización del TFM se va a utilizar una tarjeta de desarrollo de la empresa *STMicroelectronics* que incorpora un procesador ARM Cortex M4. Este microcontrolador fue seleccionado por sus características de velocidad, periféricos y memoria de los existentes en el Laboratorio de Robótica y Control de la ETSIT. Este fabricante proporciona un entorno de desarrollo con el que facilita la labor de los programadores en la configuración inicial del *hardware*.

El entorno que proporciona *STMicroelectronics* es una herramienta de desarrollo que integra todos los productos de la compañía y que permite la implementación de diferentes sistemas operativos. Esta herramienta permite configurar los periféricos, generar y compilar el código, junto con el depurado. En este TFM, estas herramientas permiten desarrollar un código portable a toda la familia de procesadores STM32, además de realizar un depurado avanzado con el objetivo de obtener información relevante sobre el estado del sistema y los requisitos de memoria.

4.2. Modelado del sistema

Antes de comenzar con la implementación es necesario realizar un diseño del sistema electrónico. En este primer paso es necesario definir todos los requisitos que debe cumplir el sistema. Posteriormente se realizará el diseño teniendo en cuenta estos requisitos y, finalmente, se analizará la solución propuesta para observar si los resultados obtenidos cumplen los requisitos definidos.

4.2.1. Análisis de requisitos

Inicialmente, es necesario definir las características del sistema. En este caso, gran parte de los requisitos ya están definidos en los Capítulos 2 y 3, por lo que se ha realizado una recopilación de ellos y se exponen a continuación:

1. Control de la esfera sobre la plataforma
 - a) Obtención de la posición de la esfera
 - b) Cálculo de la acción de control necesaria
 - c) Comunicación de la acción de control al actuador
2. Comunicación del estado del sistema al exterior

Estos requisitos no son características técnicas si no que corresponden con la funcionalidad de debe tener el mismo. Es fundamental realizar este análisis previo sin especificar tecnologías o características técnicas.

4.2.2. Máquinas de estado finitas

Después de haber definido los requisitos del sistema, se realiza el diseño, sin tener en cuenta la arquitectura del mismo, el lenguaje de programación u otros aspectos relacionados con la implementación. Esta metodología de trabajo permite realizar un análisis previo del comportamiento del sistema con el objetivo de facilitar la corrección de los errores que puedan surgir más adelante.

Una forma de realizar este modelado es mediante la división de las máquinas de estados en diferentes tareas del sistema operativo. El modelo de máquinas de estados es una forma de modelar el comportamiento de un sistema de forma discreta y determinista, es decir, que un sistema sólo puede responder de una única forma ante cada combinación posible de entradas. De esta forma, se modela un sistema que es totalmente predecible. Por otro lado, este tipo de modelados permite realizar una implementación semi-automática, además de una verificación formal del sistema para poder asegurar con certeza que se cumplen las especificaciones definidas.

La descripción en este tipo de modelos se realiza mediante la definición de una lista de estados, y de las posibles transiciones que se producen entre ellos. Con el objetivo de facilitar la implementación, se va a realizar una implementación mediante máquinas de Mealy, es decir, las acciones que realiza la máquina de estados se producen en las transiciones entre estados. Además, estas máquinas pueden ser activadas por eventos, es decir, se evalúan las posibles transiciones cuando se produce una variación en algunas de las entradas del sistema, o bien, por tiempo, es decir, existe un reloj virtual que indica el instante en el que se evalúan las transiciones.

Teniendo en cuenta las especificaciones que debe cumplir el sistema diseñado, reflejadas en la Sección 4.2.1, se ha decidido que será necesario el diseño de cuatro máquinas de estados: una para la toma de datos de los sensores, una para el cálculo de la acción de control y la actuación sobre los servo-motores y, adicionalmente, dos máquinas de estados para la comunicación con el exterior. Si se desea profundizar en el diseño de las máquinas de estados implementadas y las decisiones tomadas a la hora de realizarlo, se aconseja revisar el Anexo B en el que se trata en mayor profundidad.

4.3. Desarrollo del *firmware*

4.3.1. Implementación del *firmware* de la cámara

Como se ha comentado en el Capítulo 3, la programación del sistema de posicionamiento de la esfera se realiza mediante el lenguaje *MicroPython*. El fabricante de la cámara, *OpenMV*, proporciona un entorno de desarrollo sencillo, que es el que se ha empleado para el desarrollo del *firmware*.

En este caso, se ha decidido que la comunicación que se realice con el microcontrolador sea a través del protocolo de comunicación UART. Este protocolo serie y asíncrono permite la comunicación entre dos dispositivos de forma sencilla, sin necesidad de transmitir la señal del reloj. Por otro lado, dado que el sistema de medida sólo tiene una función, se ha creído oportuno que esta comunicación sean unidireccional, es decir, que el sistema de medida transmita la información al microcontrolador sin necesidad de que éste dé ninguna orden.

Este entorno de desarrollo, junto con las herramientas disponibles en *MicroPython*, ha facilitado la implementación de los algoritmos de detección de círculos y los protocolos de comunicación necesarios para transmitir la posición de la esfera al microcontrolador.

4.3.2. Implementación del *firmware* del microcontrolador

Como se ha comentado en las secciones anteriores, para la implementación de las máquinas de estados se ha hecho uso de las *Task*(Tarea) que permite implementar FreeRTOS. Estas tareas son ejecutadas de forma pseudo-simultánea en el microcontrolador. Es el planificador el encargado de gestionar el orden de ejecución de las mismas. Por otro lado, estas tareas trabajan de forma conjunta para conseguir un único objetivo, el control de la esfera sobre la plataforma, con lo que va a ser necesario el uso de recursos compartidos.

Cuando dos tareas intentan acceder a una variable que se encuentra en memoria se pueden producir lo que se conoce como condiciones de carrera, por ejemplo, cuando dos o más tareas acceden a un registro de memoria sin ningún control. A

las zonas de código en las que se dan condiciones de carrera se les conoce como secciones críticas. Como es lógico, la existencia de secciones críticas en el código puede dar lugar a problemas indeseados que pueden ocasionar el fallo del sistema. Para conseguir neutralizar este problema, FreeRTOS permite el uso de Colas (*Queue*) y mutex (*Mutual Exclusion*). Las colas son espacios de memoria de tipo FIFO, *First In, First Out*, es decir, que los elementos que hayan entrado primero son los que primero salen. De esta forma, una tarea puede colocar en la cola mensajes, que son recibidos por otras tareas manteniendo el orden de entrada. Por otro lado, los mutex equivalen a poseer la llave que abre un determinado espacio de memoria, si una tarea solicita un mutex se le entrega la llave, si esta está libre, y, después de realizar todas las modificaciones sobre esa variable, la tarea libera el mutex, dejándolo disponible para las demás. Si una tarea solicita un mutex y este no se encuentra disponible, generalmente queda bloqueada hasta que lo obtiene. Se suelen usar colas cuando el tamaño de la variable es relativamente bajo y mutex cuando se quieren compartir recursos *hardware* o variables de tamaños elevados. Estos son los métodos que están disponibles en FreeRTOS para la implementación de una comunicación segura entre tareas.

4.3.2.1. Comunicación con los sensores

Para realizar la comunicación con los sensores se van a emplear dos protocolos de comunicación. Para la pantalla resistiva se va a utilizar el protocolo I2C y para la comunicación con la cámara, UART. Como se ha visto en el Capítulo 3, el microcontrolador seleccionado dispone de periféricos encargados de gestionar esta comunicación. Por medio de la capa de abstracción *hardware* (HAL) que proporciona *STMicroelectronics* en los archivos *stm32l4xx_hal_i2c.c* y *stm32l4xx_hal_uart.c* se realiza la implementación de esta comunicación de forma sencilla.

Con el objetivo de reducir el tiempo que el microprocesador se encuentra ejecutando instrucciones, se va a intentar aprovechar al máximo los periféricos que dispone el microcontrolador. Especialmente cuando se realiza la comunicación, existen periféricos que permiten optimizar estos procesos. Son varias las formas de utilizar los periféricos de conectividad. En el método más sencillo, denominado *polling*, el microcontrolador da la instrucción al periférico y espera hasta que este la haya completado. Por otro lado, se puede realizar la misma función por medio de interrupciones o por acceso directo a memoria (DMA), esto es, se da la instrucción al periférico y el procesador continúa ejecutando sin esperar a que el periférico termine. En el caso de las interrupciones, el procesador no espera a que termine el periférico, pero interviene en la configuración de este. En el acceso directo a memoria, es el propio controlador DMA el que se encarga de configurar al periférico sin necesidad de intervención del microprocesador. Por este motivo, en la implementación de este sistema, siempre que sea posible se ha evitado el método de *polling* y se utilizará DMA.

Por último, es necesario comentar el protocolo de comunicación que se ha definido

para distinguir si la esfera se encuentra en la plataforma o no. En el sistema formado por la cámara, si no se ha detectado una esfera en la imagen, los valores que se transmiten al microcontrolador superan los límites de la imagen. Esto indica al microcontrolador que no se ha detectado una esfera en la plataforma, por lo que no es necesario realizar un control sobre la estructura. Por otro lado, en el sistema de la pantalla resistiva, el controlador seleccionado presenta una función de *Pen-Up and Pen-Down*, esto es, puede identificar si hay un objeto tocando la pantalla o no. De esta forma, en la trama que es enviada se encuentra esta información, que es recogida por el microcontrolador para saber si hay o no esfera sobre la plataforma.

4.3.2.2. Comunicación con los servo-motores

En el Capítulo 3 se ha mencionado el método que se va a utilizar para transmitir el ángulo objetivo a los servo-motores, esto es, mediante el uso de señales PWM. Este tipo de señal corresponde a una modulación de ancho de pulso, es decir, se modula el ancho del pulso de una señal cuadrada con el objetivo de modificar el ángulo del servo-motor.

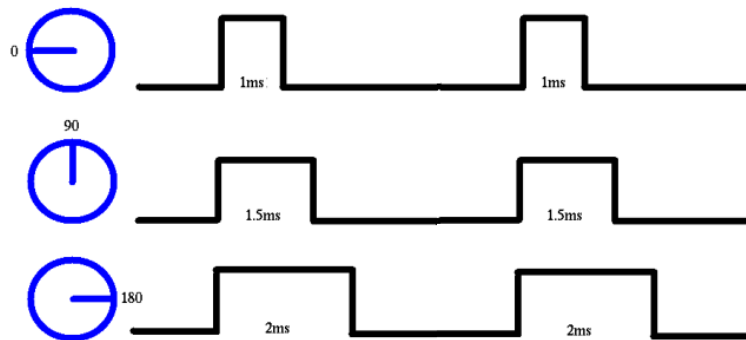


Figura 4.1: Diagrama de tiempos de la señal PWM para el control de los servo-motores. PanamaHitek.

Los servo-motores integran un controlador que permite un control preciso en posición. La comunicación del ángulo objetivo se consigue mediante la generación de una señal PWM con periodo de 20 milisegundos. Como se puede observar en la Figura 4.1, para que el servo se posicione en un ángulo de 0° se debe mantener la señal a nivel alto durante 1 milisegundo y el resto del periodo a nivel bajo, lo que equivale a un 5 % del periodo a nivel alto. Por otro lado, para obtener un ángulo en el servo-motor de 180° , se debe mantener a nivel alto la señal durante 2 milisegundos, o lo que es lo mismo, un 10 % del periodo. Por lo tanto, el tiempo que se debe mantener

la señal a nivel alto para un determinado ángulo se muestra en la Ecuación 4.1.

$$t_{alto}(ms) = 1 + \frac{Angulo(^{\circ})}{180} = 1 + \frac{Angulo(rad)}{2\pi} \quad (4.1)$$

Para la generación de esta señal el microcontrolador dispone de temporizadores, que pueden ser configurados en el modo *Output Compare* con un periodo de 20 milisegundos. En este modo, el contador del temporizador es comparado con un valor umbral y, si el contador es mayor que el umbral, la salida se encuentra a nivel alto, en cambio, si es inferior, la salida se encuentra a nivel bajo. Variando este umbral, se modifica el ancho del pulso y, por lo tanto, se puede modificar el ángulo objetivo del servo-motor.

4.3.2.3. Estructura del *software* del sistema

En el Figura 4.2 se muestra la estructura del código implementando en el microcontrolador. Se puede observar que el código de usuario, definido por el paquete *Application*, hace uso de las librerías HAL que proporciona el fabricante *STMicroelectronics* para el control de los periféricos del microprocesador, en este caso, estos son UART, I2C y temporizadores. También, este paquete utiliza la interfaz que proporciona FreeRTOS y que permite la gestión de las tareas y su comunicación. Además, dispone de las funciones encargadas de gestionar el controlador, PID o LQR, junto con el procesamiento de los datos de los sensores. Todas estas funciones se encuentran agrupadas en un mismo paquete denominado *Drivers*.

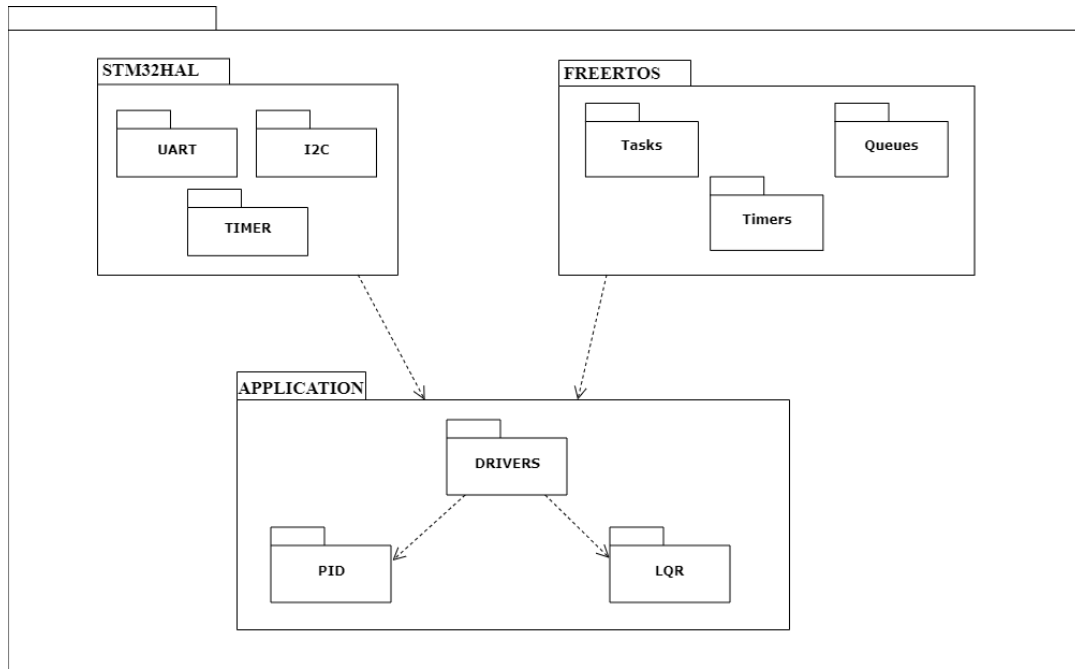


Figura 4.2: Estructura del código implementado.

4.4. Desarrollo de controladores

Como se ha comentado en el Capítulo 2, en este TFM se va a realizar la implementación de dos tipos de controladores: PID y LQR. Al igual que en desarrollo del *firmware*, en este apartado también es necesario definir los requisitos de nuestro sistema. En este caso, se ha de definir la respuesta temporal que se desea obtener.

En capítulos anteriores se ha visto que la respuesta temporal de un sistema viene definida por varios parámetros. Entre todos ellos, los que son comúnmente usados son el tiempo de establecimiento y la sobreoscilación. Dado que en este caso estas restricciones no quedan definidas de antemano, queda a cargo del usuario definir estos dos parámetros.

Se ha decidido que el tiempo de establecimiento deberá ser aproximadamente unos 3 segundos, con una sobreoscilación de menos de un 20 %. Se han escogido estos valores, debido a que se considera que tiempos de establecimiento menores requieren un control más agresivo y preciso del ángulo de inclinación de la plataforma del que se dispone. Por otro lado, una sobreoscilación del 20 % es un valor comúnmente utilizado para sistemas en los que este efecto no suponga un gran problema. Valores inferiores requerirían acciones de control más agresivas, o lo que es lo mismo, actuadores más potentes y valores superiores pueden causar efectos indeseados en el sistema.

Cuando se realiza la implementación de estos controladores en un dispositivos digital se produce una discretización. Esto se debe a que las señales de un microcontrolador no son continuas, por lo que el desarrollo teórico que se ha realizado con anterioridad no es completamente correcto. Dependiendo del periodo de muestreo de las señales, este efecto de discretización puede afectar significativamente la respuesta del sistema.

En la Figura 4.3 se observa el efecto de modificar el periodo de muestro de la señal en un sistema con un tiempo de establecimiento de 4 segundos y una sobreoscilación de 20 %. Se puede observar que si el sistema presenta un tiempo de muestreo suficientemente bajo comparado con el tiempo de establecimiento, la respuesta no se ve afectada por la discretización. Por otro lado, un periodo de muestro del mismo orden al tiempo de establecimiento puede causar inestabilidad en el sistema. Comparando las respuestas ante diferentes periodos de muestro, se observa que con un periodo de muestro dos ordenes de magnitud inferior al tiempo de establecimiento la respuesta que se obtiene no difiere significativamente del sistema continuo, por lo que siempre que se pueda se utilizará un tiempo de muestro de estas magnitudes.

4.4.1. Controlador PID

Una vez definidos los requisitos temporales de la respuesta del sistema, es momento de realizar el diseño de los controladores acorde a estos. En el Capítulo 2 se ha visto

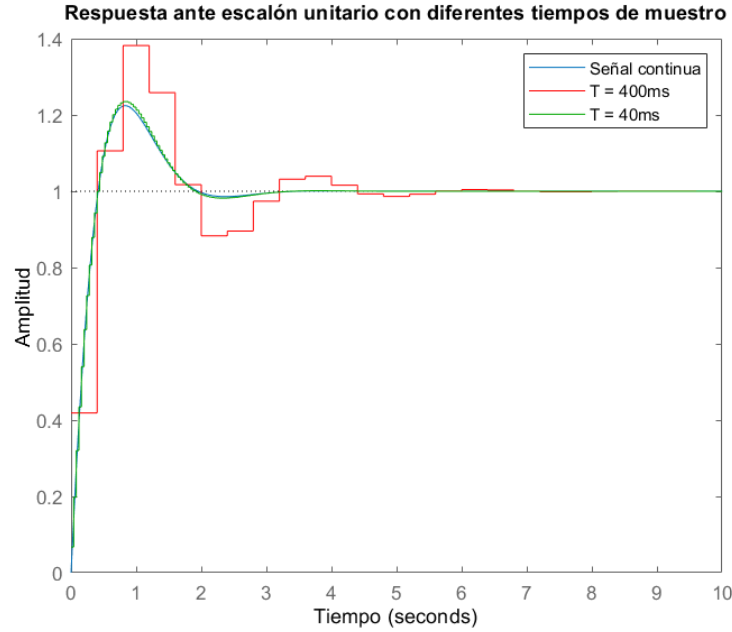


Figura 4.3: Respuesta del sistema ante distintos periodos de muestreo.

que el sistema mecánico no requiere un controlador PID completo. Por este motivo, para realizar este control se implementará un controlador PD, es decir, formado por una parte proporcional y derivativa. A partir de las Ecuaciones 2.63 y 2.64 se obtiene la ganancia y la posición del cero que debe ser añadidos al sistema en lazo abierto. A partir de estos valores, con las Ecuaciones 2.65 y 2.66, se obtienen las ganancias necesarias para implementar el controlador PD.

A partir de los requisitos planteados y de las Ecuaciones 2.60 y 2.61 se obtiene la posición en la que se debe colocar el polo:

$$\sigma = \frac{\pi}{t_s} = \frac{\pi}{3} = 1,0472 \quad (4.2)$$

$$\omega_d = \frac{\pi^2}{\ln\left(\frac{1}{M_p}\right) \cdot t_s} = 2,0441 \quad (4.3)$$

Con estos valores, aplicando la Ecuación 2.63, se obtiene la posición del cero que debe ser introducido en lazo abierto para que el polo se encuentre en el LDR:

$$a = \tan\left(\pi + 2 \cdot \left(\tan^{-1}\left(\frac{\omega_d}{\sigma}\right)\right)\right) \cdot \omega_d + \sigma = -1,7925 \quad (4.4)$$

Por otro lado, aplicando la Ecuación 2.64, se obtiene la ganancia del sistema en

lazo abierto:

$$K = \frac{\sqrt{\omega_d^2 + \sigma^2} \cdot \sqrt{\omega_d^2 + \sigma^2}}{\sqrt{\omega_d^2 + (\sigma - a)^2}} = 1,5076 \quad (4.5)$$

Por lo tanto, aplicando las Ecuaciones 2.65 y 2.66, se obtienen las ganancias de las componentes proporcional y derivativa del controlador:

$$K_D = \frac{K}{7} = 0,2154 \quad (4.6)$$

$$K_P = -K_D \cdot a = -\frac{K}{7} \cdot a = 0,3860 \quad (4.7)$$

Una vez diseñado el controlador, es necesario realizar una simulación en MATLAB para observar si la respuesta del sistema cumple los requisitos propuestos.

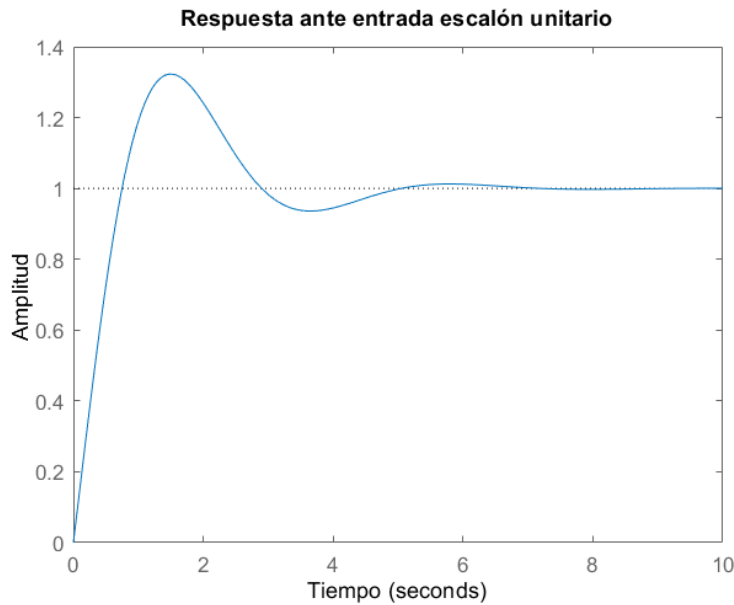


Figura 4.4: Respuesta del sistema aplicando el controlador PD diseñado.

Además, se puede observar la posición del polo y el LDR en la Figura 4.5.

Se observa claramente en la Figura 4.4 que la respuesta del sistema no satisface los requisitos que se han propuesto. Esto se debe a que las suposiciones que se han realizado para el cálculo de la posición del polo dominante no son correctas en este caso. Dado que en este sistema el factor de amortiguamiento es comparable a 1, la Ecuación 2.60 no es del todo correcta, lo que provoca que el comportamiento

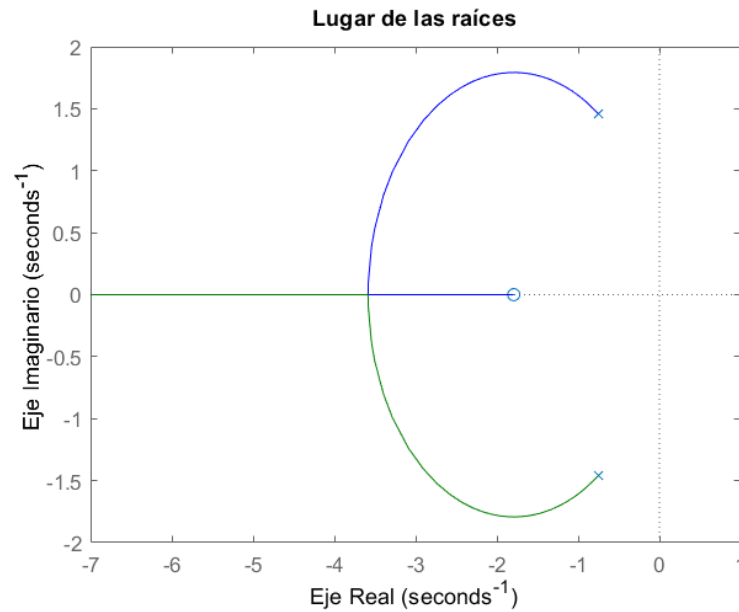


Figura 4.5: Lugar de las raíces del sistema aplicando el controlador PD diseñado.

del sistema no sea el deseado. Aunque este diseño no satisfaga los requisitos, esto no significa que sea inútil. Es un punto de partida a la hora de realizar el ajuste experimental del controlador, ya que proporciona una respuesta similar a la que se desea obtener.

Finalmente, para la implementación de este controlador es necesario calcular el error, y su derivada, de la posición de la esfera respecto de una determinada referencia y multiplicar estos valores por las ganancias obtenidas anteriormente.

4.4.2. Regulador cuadrático-lineal

Como se ha visto en el Capítulo 2, el diseño de un controlador LQR se realiza experimentalmente con el uso de herramientas de simulación como MATLAB.

En este tipo de controladores, no sólo se realiza un ajuste en la señal de salida, si no que también se puede realizar un control en la señal de entrada. Es decir, dado que los servo-motores no presentan un rango de actuación infinito, su ángulo de movimiento se encuentra reducido en un determinado rango. Se puede diseñar el controlador de forma que se minimice el efecto de estas imperfecciones. Para ello, a la hora de simular el compartimiento del sistema, se puede realizar un control sobre el ángulo de inclinación de la plataforma e intentar mantenerlo en un régimen que se ajuste a las limitaciones del sistema.

Inicialmente, se ha realizado una simulación siguiendo las recomendaciones que se comentaron en la Sección 2.5.2.1, en la que la matriz R se define como la unidad y Q

como $C^T C$. Se observan los resultados obtenidos de esta simulación en la Figura 4.6 que muestra la respuesta del sistema ante una entrada escalón de 1 centímetro.

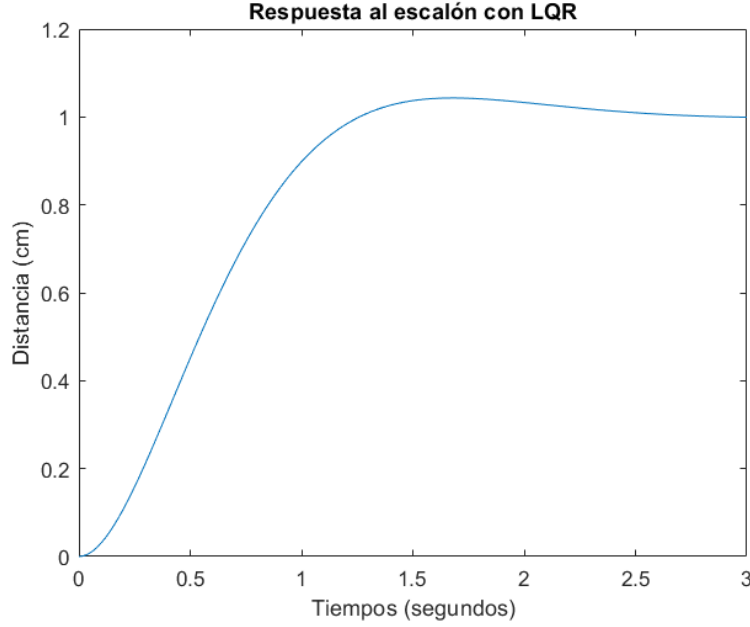


Figura 4.6: Respuesta del sistema aplicando el controlador PD diseñado.

Como se puede observar en la Figura 4.6, la respuesta cumple satisfactoriamente los requisitos que se han propuesto, siendo el tiempo de establecimiento inferior a los 3 segundos y la sobreoscilación de un 4 % aproximadamente. Por otro lado, para estimar el ángulo máximo de inclinación de la plataforma se realiza la simulación de la respuesta del sistema ante una entrada escalón de 20 centímetros. Se ha escogido este valor de entrada, debido a que el valor máximo de la señal de error se produce en el momento en el que se desea llevar la esfera a la zona opuesta de la plataforma. Dado que la plataforma presenta una longitud máxima de 20 centímetros, en estas condiciones son las que se observará un error de posición máximo, lo que equivale a una señal de control lo más agresiva posible. Se observa en la Figura 4.7 que el valor máximo de inclinación corresponde a unos 16° , un valor que es alcanzable y que se ajusta correctamente a las limitaciones de esta estructura.

Se ha conseguido obtener esta respuesta con los siguientes pesos para la función de coste:

$$Q = C^T \cdot C = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \quad (4.8)$$

$$R = 1 \quad (4.9)$$

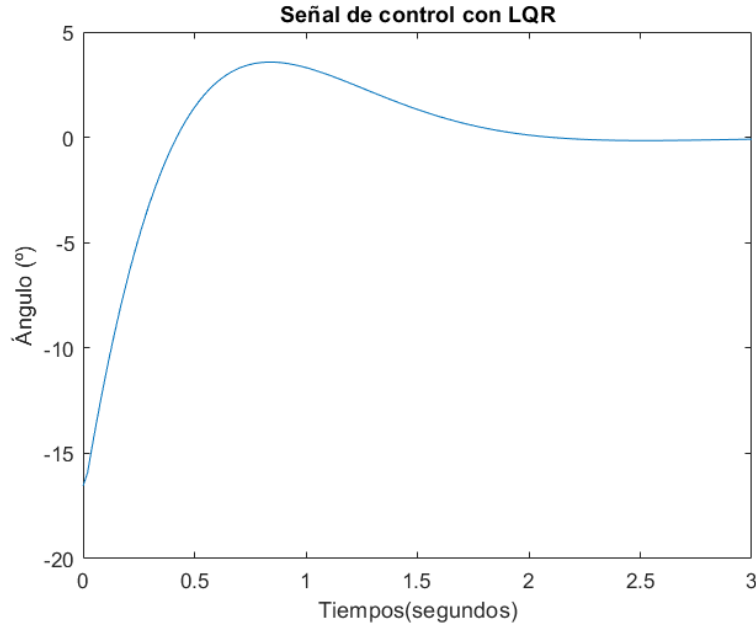


Figura 4.7: Ángulo de la plataforma ante un escalón de 20 centímetros.

Con esto, se obtiene la matriz de realimentación de estados:

$$K = \begin{bmatrix} 1 & 0,5345 \end{bmatrix} \quad (4.10)$$

La implementación es similar al controlador anterior, pero en este caso, la realimentación se realiza a través de las variables de estado. Para obtener la acción de control necesaria en cada instante es necesario conocer el estado del sistema. Al haber modelado el sistema en el espacio de estados teniendo en cuenta esto mismo, se obtiene un estado que corresponde a la posición de la esfera y su velocidad en cada uno de los ejes. Gracias a esto, se puede obtener el estado del sistema de una forma sencilla. Al conocer la posición de la esfera, se puede obtener su velocidad, y con esto, se obtiene la acción de control necesaria al multiplicar la matriz de estados del sistema por la matriz de realimentación, K .

4.5. Pruebas y ajuste del sistema

Antes de mostrar los resultados, es necesario realizar unos ajustes sobre la plataforma. El objetivo de estos pasos previos es el de corregir las posibles imperfecciones que se existen al realizar la implementación del modelo simulado en un sistema real.

4.5.1. Calibración de la estructura

Es fundamental realizar una calibración de la inclinación de la plataforma antes de comenzar con las pruebas. Con el objetivo de intentar conseguir una posición horizontal de la plataforma, se puede compensar esta inclinación añadiendo un cierto valor a los ángulos en los servo-motores.

Para conocer el valor exacto que es necesario corregir el ángulo de los servo-motores se puede hacer uso de un acelerómetro o un giroscopio. El acelerómetro es un sensor que mide la aceleración que sufre un cuerpo. Con lo que si se coloca un sensor de este tipo sobre la plataforma, y se miden las componentes horizontal y vertical se puede obtener el ángulo de inclinación. Utilizando la relación entre el ángulo de la plataforma y los servo-motores, se obtiene en ángulo con el que se corrige este efecto. Por otro lado, si se dispone de un giroscopio, al ser este un sensor que proporciona directamente la inclinación de un objetivo, podemos realizar el mismo procedimiento para realizar la corrección.

4.5.2. Corrección de la posición de la esfera

A diferencia del efecto comentado en la Sección 4.5.1, que simplemente induce un error en régimen permanente, el efecto que se describe a continuación puede producir que el sistema sea inestable, aún sabiendo que éste es perfectamente controlable con los controladores propuestos.

Dado que la cámara se encuentra en una posición y orientación fijas, cuando la plataforma presenta un ángulo de inclinación elevado, la posición de la esfera observada presenta un error relevante respecto a la real. Por lo tanto, se puede obtener la distancia real de la esfera al centro a partir de la distancia medida y la inclinación de la plataforma, α :

$$d_{real} = \frac{d_{medida}}{\cos(\alpha)} \quad (4.11)$$

4.6. Planificación en tiempo real

Como se ha visto al comienzo de este Capítulo, se realizará la implementación del *firmware* del sistema sobre un sistema operativo en tiempo real. Una pregunta que puede surgir al mencionar este tipo de sistemas operativos y los motivos por los que se usan es por qué es necesario en este TFM. Es fundamental garantizar que el periodo en el que se calcula la acción de control sea conocido e invariable. Esto es debido a que para realizar un cálculo correcto de la acción de control se requiere conocer el periodo de ejecución de esta tarea.

Para garantizar estos requisitos temporales es necesario realizar la implementación en un sistema operativo en tiempo real, pero, además, requiere de un análisis de planificabilidad. Por medio de este análisis se puede garantizar con una certeza máxima que el sistema implementado cumplirá con las especificaciones. Para realizar este análisis es necesario definir primero la política del planificador que se implementa en el sistema operativo. En el caso de FreeRTOS, se implementa la política *Round-Robin* con prioridades, esto es, la ejecución de las tareas se realiza siguiendo un orden pre-establecido por el programador, definido por prioridades. Además, un planificador con política *Round-Robin* implica que se reparte el tiempo de ejecución entre todas las tareas. En FreeRTOS, sólo en el caso de que dos tareas presentan la misma prioridad, el planificador reparte el tiempo de ejecución a cada una de ellas.

Una vez definida la política del planificador, es necesario realizar una estimación del tiempo de ejecución máximo de las tareas. Para ello se ha decidido realizar esta estimación en un entorno real, es decir, mientras el sistema se encuentran en funcionamiento. De esta forma, el resultado que se obtiene es lo más realista posible. Por otro lado, es necesario definir los plazos y los períodos de ejecución de cada una de las tareas. Siendo el plazo el tiempo máximo que debe pasar entre la activación de una tarea y su finalización y el periodo, el tiempo entre dos activaciones consecutivas de una tarea. En este caso, estos tiempos quedan definidos por las especificaciones del sistema de la siguiente forma:

- Toma de datos: el tiempo de entre dos ejecuciones consecutivas de esta tarea queda definido por el periodo de muestreo mínimo que permiten los sensores. En el caso de la cámara, este tiempo se encuentra entre los 5 y los 10 milisegundos, en el caso de la pantalla resistiva, es de unos 10 milisegundos. Por lo tanto, con el objetivo de diseñar un sistema que permite el uso de ambos sensores, esta tarea deberá tener un periodo de ejecución mínimo de 5 milisegundos.
- Actualización de la señal de control: la selección del periodo de esta tarea es fundamental, ya que esta directamente relacionado con la respuesta del sistema. Si se elige un periodo demasiado largo, la respuesta del sistema variará de forma considerable de la simulación ya que los efectos de discretización presentarán un gran peso. Por otro lado, un periodo demasiado corto puede suponer una carga excesiva para el microcontrolador, sin verse mejorada la respuesta del sistema de forma considerable. Dado que no se dispone de una restricción directa en esta tarea, se ha decidido que un periodo adecuado para esta tarea podría ser similar al periodo de la lectura de los sensores.
- Comunicación con el exterior: la tarea de recepción de comandos, como la que se encarga de enviar el estado del sistema al exterior, pueden tener periodos similares. Dado que en este caso no se dispone de ningún requisito que pueda restringir en la selección de un periodo de ejecución, se ha decidido que el periodo de esta tarea sea aproximadamente 10 veces el de la tarea de actualización de la señal de control.

Por otro lado, es necesario realizar una estimación de los tiempos de ejecución de cada una de las tareas. Existen varios métodos para conseguir este objetivo. En este caso, se ha decidido que el método que se va a emplear consiste en activar un GPIO del microcontrolador cuando la ejecución de la tarea de comienzo, y desactivarla cuando este termine su ejecución. De esta forma, con la ayuda de un osciloscopio se puede observar la señal, y medir los tiempo de ejecución de cada tarea. Además, como todas las tareas comparten recursos entre sí, es necesario medir el tiempo de ocupación de los recursos por cada una. Una vez obtenidos todos los tiempos, se muestra una tabla en la que se sintetiza esta información:

Tarea	$C(\mu s)$	$T(\mu s)$	$D(\mu s)$	$R1(\mu s)$	$R2(\mu s)$	$R3(\mu s)$
Lectura de sensores	4.2	5000	5000	0.2	-	-
Cálculo control	500	5000	5000	10.4	20.8	20.8
Envío de datos	10	5000	5000	-	0.4	-
Recepción de datos	4.6	50000	50000	-	-	0.4

Tabla 4.1: Tiempos asociados a la ejecución de las tareas.

En esta tabla, se observan los tiempo de ejecución(C), los periodos de la tareas(T), los plazos de respuesta(D), que en este caso son iguales a los periodos, y los tiempos de ocupación en cada uno de los recursos compartidos. Los tiempos de acceso a los recursos $R1$, $R2$ y $R3$ corresponden al procesado de los datos de la detección de la esfera, la posición de la esfera y la referencia de posición, respectivamente. Cada uno de las tareas requiere leer o escribir en estas variables, lo que las hace susceptibles a condiciones de carrera. Por estos motivos, es necesario el uso de herramientas como los Mutex que permiten realizar un control de acceso a estas posiciones de memoria. Dado que en un determinado instante sólo una de las tareas puede estar leyendo o escribiendo en estas variables, hay que realizar una planificación teniendo esto en cuenta.

En la Tabla 4.1 se puede observar que se ha decidido que el periodo de activación de la tarea de control sea de 5 milisegundos. Hay que tener en cuenta que un control tan rápido no va ser posible en este sistema debido a que se encuentra condicionado por la velocidad de los actuadores, pero se ha decidido seleccionar este periodo debido a que si se mejora estos dispositivos, se puede obtener un periodo de este orden. Además, permite realizar una estimación del mínimo periodo de muestro que permite este microcontrolador con los sistemas de medida disponibles.

Para realizar la planificación es necesario calcular el tiempo máximo que tarda una tarea en finalizar su trabajo. Este tiempo es la suma de su tiempo de ejecución, del tiempo de ejecución de las tareas con prioridad superior a la suya y el tiempo en el que esta tarea se encuentra bloqueada por otras de menor prioridad.

Inicialmente, se debe elegir la prioridad de cada una de las tareas. Es una práctica muy común asignar la prioridad máxima a la tarea que presenta un plazo de respuesta

mínimo. De este modo, se puede observar en la Tabla 4.2 las prioridades asignadas a cada tarea.

Tarea	T/D(μs)	P
Lectura de sensores	5000	3
Cálculo control	5000	1
Envío de datos	5000	2
Recepción de datos	50000	4

Tabla 4.2: Asignación de prioridades a las Tareas.

Finalmente, se obtiene el tiempo de respuesta máximo(R) para una tarea como se muestra en la Ecuación 4.12.

$$R_i = C_i + \sum_{j \neq p(i)} \left\lceil \frac{R_j}{T_j} \right\rceil + B_i \quad (4.12)$$

Siendo B_i el tiempo máximo que es bloqueada una tarea. Este bloqueo se produce cuando una tarea intenta acceder a un recurso compartido que se encuentra ocupado por otra tarea de igual o menor prioridad. Si se realiza este cálculo para cada una de las tareas, se obtiene finalmente la tabla de tiempos completa (ver Tabla 4.3).

Tarea	C(μs)	T(μs)	D(μs)	R(μs)	P	R1(μs)	R2(μs)	R3(μs)
Lectura de sensores	4.2	5000	5000	514.2	3	0.2	-	-
Cálculo control	500	5000	5000	501	1	10.4	20.8	20.8
Envío de datos	10	5000	5000	510	2	-	0.4	-
Recepción de datos	4.6	50000	50000	518.8	4	-	-	0.4

Tabla 4.3: Tiempos de respuesta de las Tareas.

El tiempo de respuesta(R) corresponde con el tiempo máximo que puede tardar una tarea desde su activación hasta que finalice su ejecución. Para que el sistema sea planificable, es decir, que cumpla con las restricciones temporales, el tiempo de respuesta de cada tarea debe ser igual o inferior a su plazo. Se observa en la Tabla 4.3 que esta condición se cumple para todas las tareas, con lo que se puede afirmar que el sistema es planificable.

A parte del método que se ha comentado anteriormente, existe una forma más sencilla de realizar el cálculo de la planificabilidad del sistema. Este método se basa en calcular el factor de ocupación del procesador y compararlo con un valor teórico límite. Si el valor real obtenido es inferior al límite se puede concluir con certeza que el sistema es planificable. Por otro lado, si el valor obtenido es superior al límite es necesario realizar el método comentado con anterioridad. Este factor se calcula como

el sumatorio del factor de ocupación para todas las tareas, como se muestra en la Ecuación 4.13.

$$U = \sum_{i=1}^N \frac{C_i}{T_i} \quad (4.13)$$

En este sistema, se obtiene:

$$U = \frac{4,2}{5000} + \frac{500}{5000} + \frac{10}{5000} + \frac{4,6}{50000} = 0,1029 \quad (4.14)$$

La utilización máxima con la que se asegura planificabilidad para N tareas ejecutándose en el sistema es:

$$U_0(N) = N \cdot (2^{\frac{1}{N}} - 1) \quad (4.15)$$

Al resolver la Ecuación 4.15 para un sistema de cuatro tareas se obtiene un factor de utilización de 0,7568. Comparando este valor con el factor real de utilización en el sistema y, al ser este valor inferior al máximo posible, se puede asegurar con certeza de que este sistema es planificable.

4.7. Optimización del consumo

En una gran parte de los sistemas embebidos, el tiempo en el que estos se encuentran en estado de espera es mucho mayor del que se encuentran ocupados. Si el consumo del procesador fuese idéntico en ambos modos, implicaría unas pérdidas energéticas importantes, con lo que surge la necesidad de implementar distintos modos de funcionamiento que permitan el ahorro energético en momentos de reposo.

Muchas de las técnicas que existen hoy en día para la reducción del consumo realizan un uso dinámico de estos modos, es decir, se producen transiciones entre modos de consumo durante la ejecución de los programas. Llegando a darse casos en los que una parte es ejecutada con un modo de funcionamiento distinto en función de su estado de ejecución.

El sistema que se desea implementar tiene como objetivo realizar una tarea que, desde el punto de vista computacional, es muy sencilla, por lo que requerirá un tiempo de ejecución bajo. Por otro lado, se puede suponer que el periodo de ejecución de las tareas de control es varios ordenes de magnitud superior a su tiempo de ejecución, con lo que el microprocesador se mantendrá en un estado de espera gran parte su ciclo de ejecución. Por estos motivos, es necesario definir una estrategia de ahorro de energía.

La estrategia que se va a implementar consiste en realizar todas las tareas a la frecuencia máxima que permite el microcontrolador, con el objetivo de mantenerlo en un modo de bajo consumo el máximo tiempo posible. Se ha decidido que esta será la técnica empleada en este caso, ya que los tiempos de ejecución que presentan las tareas son muy bajos comparados con los tiempos de espera. Si se aprovechan estos periodos de espera para colocar al microcontrolador en un modo de ahorro de energía, se puede obtener una reducción significativa del consumo.

En el Capítulo 3 se ha visto que el microcontrolador seleccionado para el proyecto será el STM32L476RG. Este microcontrolador dispone de siete modos de bajo consumo. Para saber cuál de ellos es el adecuado en este caso, primero es necesario realizar un análisis de estos.

- *Sleep*: modo en el que la CPU es lo único parado. Todos los periféricos continúan funcionando de forma normal. La CPU puede ser despertada por medio de interrupciones o eventos. Este modo presenta un tiempo de despertado máximo de 6 ciclos de reloj.
- *Low-power run*: en este modo se reduce el voltaje del sistema, limitando la frecuencia de la CPU a 2MHz.
- *Low-power sleep*: este modo se puede alcanzar únicamente accediendo desde el modo *Low-power run*. Tiene las mismas restricciones que el anterior, pero el reloj de la CPU está parado. Cuando se despierta al sistema mediante interrupción o evento, se activa el *Low-power run*. Este modo presenta un tiempo de despertado máximo de 9 ciclos de reloj.
- *Stop 0*, *Stop 1* y *Stop 2*: estos modos permiten el mínimo consumo posible, manteniendo el contenido de la memoria SRAM y los registros. Todos los relojes que se encuentran alimentados por el regulador de voltaje interno son parados. Los relojes LSE y LSI (*Low-speed external* y *Low-speed internal*) son los únicos que se mantiene activos. El modo *Stop 0* permite que el regulador de voltaje se mantenga encendido, lo que reduce los tiempo de despertado aumentando significativamente el consumo. El modo *Stop 1* permite tener un gran número de periféricos activos y fuentes de despertado, y el modo *Stop 2*, es el que presenta un consumo mínimo pero a costa de aumentar el tiempo de despertado del sistema. Los modos *Stop 0*, *Stop 1* y *Stop 2* presentan tiempos de despertado máximos de 11, 13 y 17 μ s, respectivamente.
- *Standby*: es el modo que permite el menor consumo posible con BOR(*Brown-out Reset*)¹. En este modo todos los sistemas alimentados por el regulador de voltaje son apagados. Además, los contenidos de las memorias SRAM y de los registros son borrados, excepto los que hayan sido almacenados en los registros de respaldo. El dispositivo puede salir de este modo por medio de interrupciones

¹El *Brown-out Reset* es un restablecimiento del sistema que se produce cuando se detecta un voltaje de alimentación inusualmente bajo. Es una medida de seguridad que permite garantizar un arranque seguro cuando se restablezca la energía.

externas o generadas por errores internos. Este modo presenta un tiempo de despertado máximo de $40\ \mu\text{s}$.

- *Shutdown*: este modo permite alcanzar el mínimo consumo posible. Este modo es similar al anterior pero no es posible realizar un seguimiento del voltaje de alimentación, además, no es posible acceder a los registros de respaldo, con lo que los contenidos de la memoria SRAM y de los registros son borrados. Este modo presenta un tiempo de despertado máximo de $300\ \mu\text{s}$.

Para la implementación de estos modos de consumo, FreeRTOS permite ejecutar una secuencia de instrucciones cuando el microcontrolador se encuentra en modo de reposo, es decir, cuando no hay ninguna tarea a la espera de ser ejecutada. De esta forma, cada vez que todas las tareas se hayan ejecutado se deberá acceder al modo de bajo consumo y mantener al microcontrolador en este estado hasta la siguiente activación de alguna tarea.

Habiendo realizado un análisis de los modos de consumo que proporciona este microcontrolador, es necesario definir los requisitos del modo de consumo óptimo en este sistema. Como se ha comentado anteriormente, es necesario mantener la señal de control en los servo-motores en todo momento. Dado que se utiliza una señal de tipo PWM para su control, y ésta es generada por los temporizadores del microcontrolador, el modo de funcionamiento seleccionado debe mantener estos temporizadores encendidos constantemente. Con esta restricción, se descartan los modos *Shutdown*, *Standby*, *Stop 0*, *Stop 1* y *Stop 2*, ya que en ellos los periféricos funcionan a una frecuencia de reloj muy reducida o se encuentran apagados. Además, es importante que los modos de funcionamiento que se escojan permitan la vuelta con tiempos razonablemente bajos, para que esto no afecte a la planificabilidad del sistema, y por medio de eventos generados por el sistema operativo. Con estas restricciones, el único modo de bajo consumo que puede ser implementado es el *Sleep*. Los modos *Low-power* reducen significativamente la frecuencia de la CPU, lo que aumenta significativamente los tiempos de despertado. Por este motivo, no son adecuados en este caso, en el que mantener las restricciones temporales propuestas es un objetivo prioritario.

Al realizar la implementación de esta estrategia de ahorro de energía, se ha conseguido reducir el consumo del sistema en un 30 % aproximadamente. Manteniendo el microcontrolador en el modo de funcionamiento estándar constantemente, se requiere de una corriente de 21 mA para su funcionamiento y, aplicando esta estrategia de ahorro de energía, se consigue reducir su consumo hasta los 14 mA aproximadamente.

Capítulo 5

Resultados y validación del sistema completo

5.1. Controlador PID

En esta sección se muestra el análisis de la respuesta del sistema ante un controlador PID. Por otro lado, se propone realizar los ajustes necesarios para conseguir los requisitos planteados en capítulo anterior. Dado que para realizar el ajuste teórico de este controlador se ha utilizado un método analítico, se puede realizar un ajuste experimental siguiendo los mismos procedimientos.

5.1.1. Sistema con pantalla resistiva

Inicialmente, se realiza el control con el PD que se ha calculado en el Capítulo 2. Se puede observar en la Figura 5.1 que el comportamiento real del sistema difiere en gran medida del simulado, tanto en el tiempo de establecimiento como en la sobreoscilación. Al analizar esta respuesta se observa que se produce un error en régimen permanente de unos 2cm. Esto se debe a una no linealidad del sistema, esta es, un fenómeno denominado como zona muerta. Este efecto se produce cuando los actuadores, o sistemas, presentan un rango de entradas en las que no se obtiene ninguna respuesta (ver Figura 5.2). En este sistema, con inclinaciones reducidas de la plataforma no se produce un movimiento en la esfera. Una forma de minimizar el efecto de este fenómeno es el uso de controladores más específicos.

Dado que se ha realizado el LDR de este sistema, se puede obtener una respuesta más rápida y más acorde a los requisitos de dos formas: moviendo el cero que proporciona el PD o aumentando la ganancia del controlador. En las Figuras 5.3 y 5.4 se muestran las respuestas aplicando estos dos métodos. En el primer caso, se observa una respuesta mucho más oscilatoria y con un tiempo de establecimiento mucho mayor. En el segundo caso, al mover el cero, pero no aumentar la ganancia

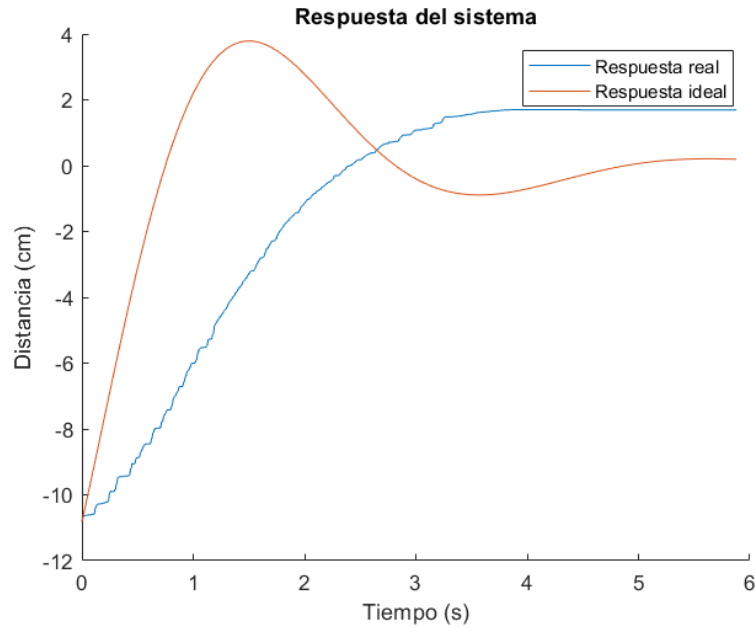


Figura 5.1: Respuesta del sistema ante el PD teórico.

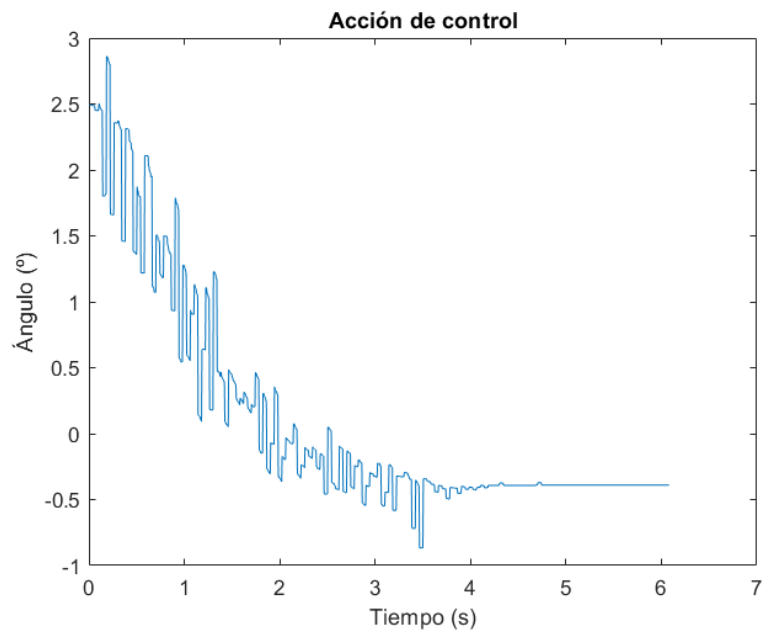


Figura 5.2: Acción de control del sistema ante el PD teórico.

del sistema de forma considerable, la respuesta presenta una oscilación mucho menor, entorno al 15 %. Además, se consigue reducir el tiempo de establecimiento hasta los 4 segundos, junto con el error en régimen permanente.

Una vez comprobada la respuesta que presenta el sistema aplicando el controlador teórico es necesario realizar un ajuste experimental. El objetivo es modificar los parámetros del controlador para conseguir los requisitos propuestos. Para ello,

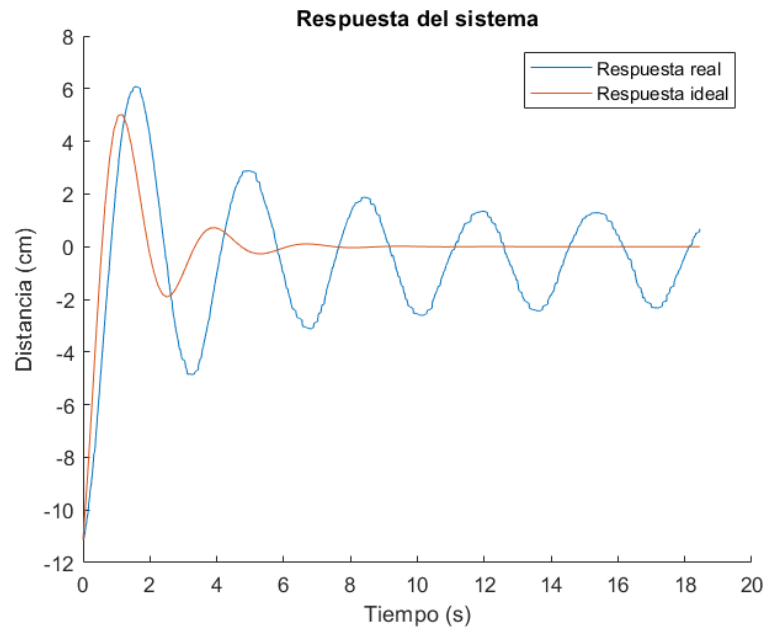


Figura 5.3: Respuesta del sistema ante el PD con ajuste de ganancia.

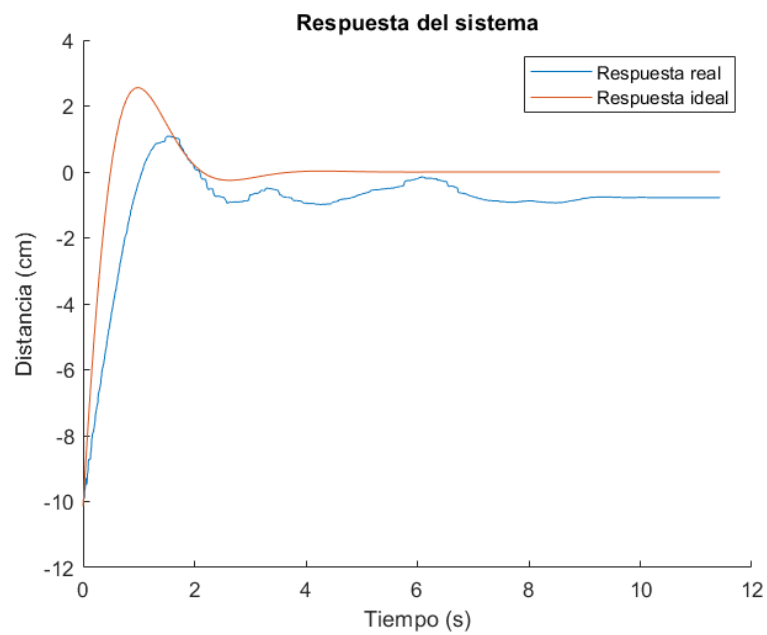


Figura 5.4: Respuesta del sistema ante el PD con ajuste de cero.

siguiendo el diagrama LDR que se ha obtenido con anterioridad (ver Figura 4.5), se ha conseguido ajustar el controlador de forma satisfactoria. Se observa en la Figura 5.5 la respuesta del sistema con el controlador PD obtenido de forma experimental.

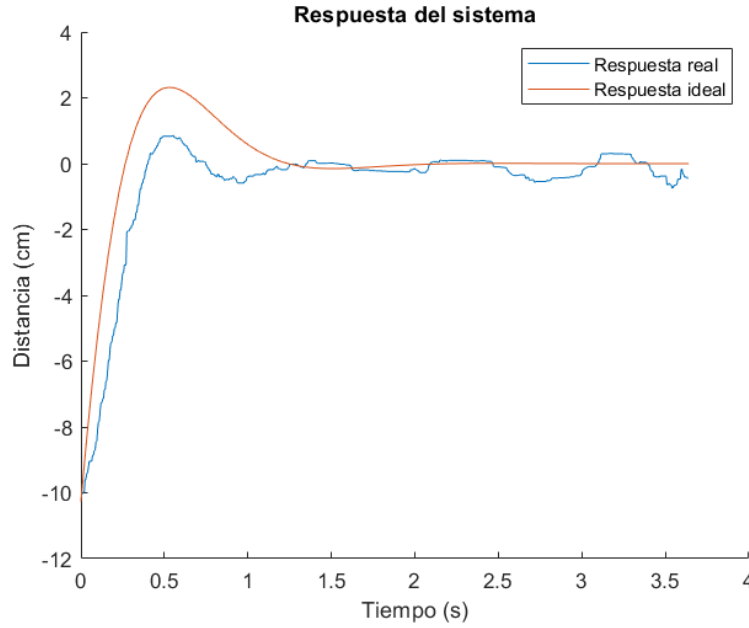


Figura 5.5: Respuesta del sistema ante el PD experimental.

Para esta respuesta, el PD ha sido sintonizado las siguientes constantes:

$$K_P = 2,6 \quad K_D = 0,8 \quad (5.1)$$

Se puede comprobar que la respuesta no presenta una diferencia elevada respecto de la simulada como en el primer caso. En este caso, se observa una sobreoscilación de un 9% y un tiempo de establecimiento aproximadamente de 1.5 segundos. Esto se debe a que el fenómeno de la zona muerta ejerce un efecto importante sobre el sistema. Por otro lado, algo que también se puede apreciar es la oscilación que se produce en el régimen permanente. Esto se debe a que los sensores y actuadores de los que se dispone no presentan una resolución suficiente, lo que impide un control preciso, que es necesario para detener la esfera en el centro de la plataforma. La única forma de eliminar este efecto es adquirir dispositivos más precisos, lo que aumentaría significativamente el coste del sistema.

5.1.2. Sistema con cámara

En el sistema formado por la cámara, el procedimiento que se ha seguido es idéntico al anterior. Inicialmente, se ha observado la respuesta del sistema ante el PD teórico (ver Figura 5.6). Se puede apreciar que la respuesta no presenta una forma similar a la simulada. Esto se debe esencialmente al efecto que produce la zona muerta. Además, en este caso, la respuesta es mucho más rápida. Por otro lado, esta respuesta presenta un tiempo de establecimiento de 2 segundos y en la que no se observa una sobreoscilación. De forma similar al caso anterior, se observa que sigue existiendo un

error en el régimen permanente de aproximadamente 1 centímetro debido al efecto de la zona muerta. Aunque en este caso se satisfacen los requisitos en el régimen transitorio, la respuesta presenta un error que debe ser corregido.

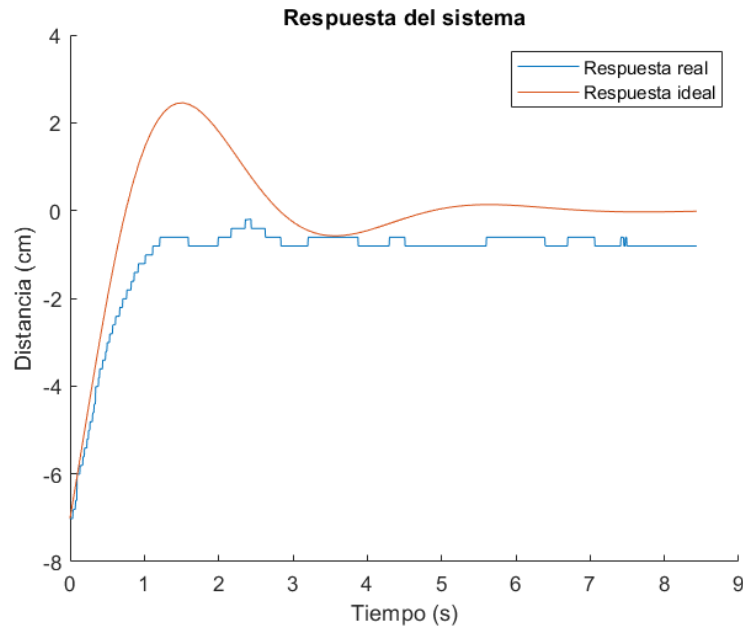


Figura 5.6: Respuesta del sistema formado por la cámara ante el PD teórico.

Dado que ya se ha conseguido obtener una respuesta casi satisfactoria, resulta interesante aplicar a este sistema el PD obtenido de forma experimental en el sistema con la pantalla, con el objetivo de comparar ambas respuestas y reducir el error en régimen permanente. Se puede observar la respuesta del PD experimental en la Figura 5.7. Esta respuesta presenta una gran similitud con la simulada, aunque se sigue observando la oscilación en el régimen permanente, se consigue eliminar el error, ya que oscila sobre el centro de la plataforma. Por otro lado, con esta respuesta se observa una sobreoscilación de un 25 % y un tiempo de establecimiento de unos 2 segundos.

5.2. Controlador LQR

El objetivo de esta sección es el de comparar la respuesta que se obtiene a partir de controlador LQR en un entorno real frente a la simulación.

5.2.1. Sistema con pantalla resistiva

Se muestra en la Figura 5.8 el resultado de aplicar el LQR al sistema con la pantalla resistiva. En esta figura se observa un ligero error en régimen permanente

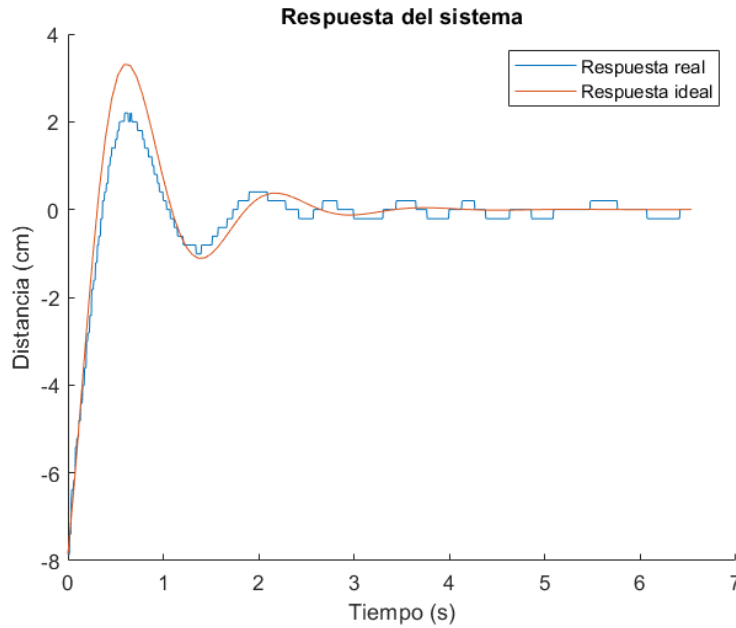


Figura 5.7: Respuesta del sistema formado por la cámara ante el PD experimental.

de aproximadamente 0.5 centímetros. Por otro lado, no se observa sobreoscilación y el tiempo de establecimiento es de unos 2 segundos. De forma similar a los casos anteriores, la oscilación en régimen permanente se encuentra presente en la respuesta.

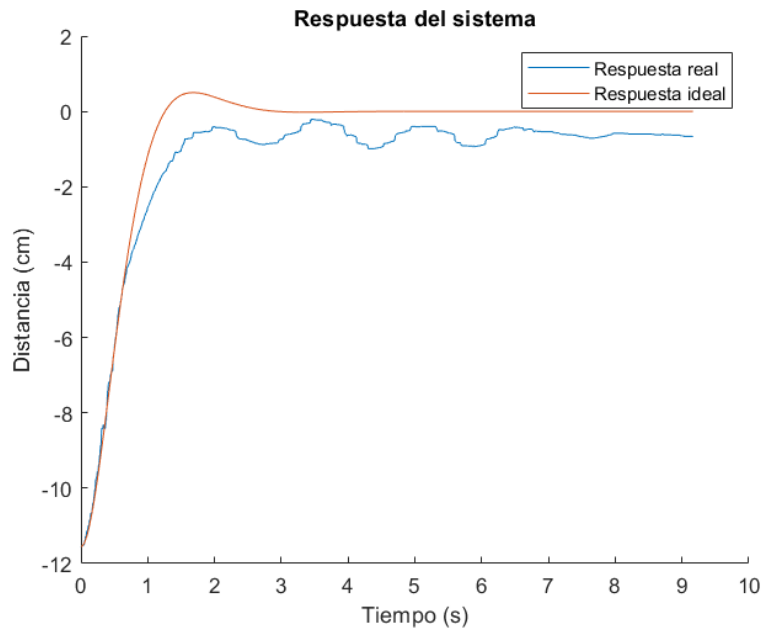


Figura 5.8: Respuesta del sistema con pantalla resistiva ante el LQR.

Se puede observar en la Figura 5.9 la respuesta del sistema con la cámara aplicando el mismo LQR. En este caso se observa que el tiempo de establecimiento se encuentra

entorno a los 2 segundos junto con una sobreoscilación de un 5 %. A diferencia del controlador PD, este es el sistema en el que menor similitud presenta entre su respuesta simulada con la real.

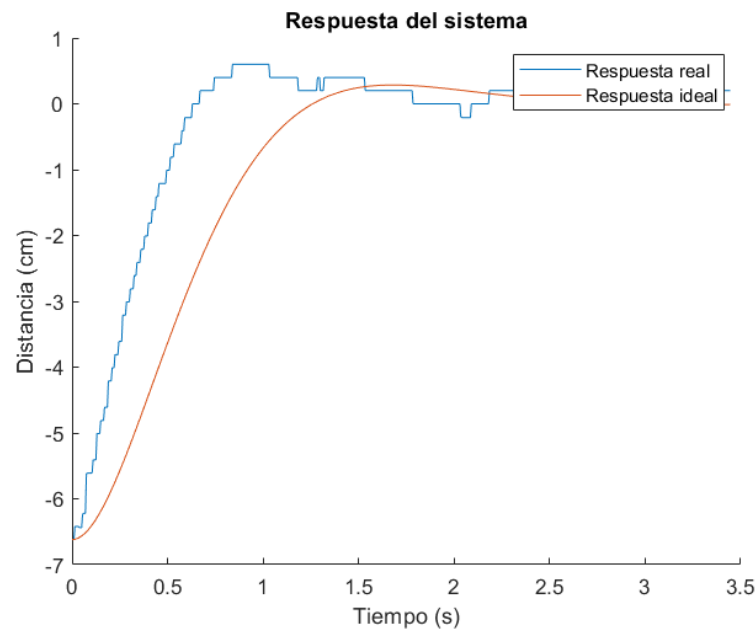


Figura 5.9: Respuesta del sistema con cámara ante el LQR.

En general, se puede concluir que el sistema formado por la cámara presenta una mayor similitud con el modelo ideal. Uno de los motivos por el cuál puede suceder esto es el tamaño y peso de la esfera. En el sistema de la pantalla resistiva, la esfera presenta un peso mucho mayor y un tamaño significativamente más grande, lo que probablemente afecte a su respuesta. Por otro lado, ante un mismo controlador, se observa que la respuesta del sistema de la cámara presenta un tiempo de establecimiento mucho menor, algo que también se puede ver afectado por el tamaño de la esfera. Aunque ambos sistemas presentan la misma función de transferencia, se observa que en la realidad su comportamiento no es idéntico.

Capítulo 6

Conclusiones

En este Trabajo Fin de Máster se ha diseñado e implementado un sistema control de una estructura de balanceo de dos grados de libertad. Inicialmente, se han realizado un modelado del sistema a partir del cuál se han diseñado dos tipos de controladores: PID y LQR. Finalmente, se ha realizado la implementación de los controladores en un sistema real y se han analizado las diferencias respecto del modelo simulado. Todo el proceso de implementación se ha descrito de forma detallada con el objetivo de que este sistema pueda ser utilizado por los alumnos del Laboratorio de Robótica y Control de la ETSIT. El sistema final que se ha obtenido cumple satisfactoriamente con los objetivos propuestos.

Aunque el resultado final es satisfactorio, es necesario comentar algunos de los problemas encontrados durante su desarrollo. Inicialmente, se ha realizado un modelado matemático del sistema, lo que ha permitido diseñar los controladores por los métodos comentados en el Capítulo 2. En el diseño basado en el LDR, se han realizado aproximaciones que posteriormente se ha observado que no eran del todo correctas. Hay que tener en cuenta esto al realizar la implementación de los controladores en el sistema real, ya que puede llevar asociado una respuesta indeseada.

Por otro lado, a la hora de realizar la implementación de los controladores en el sistema real se han observado varios problemas que no han podido ser solucionados. En el caso de los actuadores, estos no presentaban la suficiente resolución, por lo que en todas las respuestas observadas se aprecia una pequeña oscilación en el régimen estacionario. Por otro lado, en el caso del sistema formado por la cámara, se han observado dos problemas que han podido interferir en la respuesta del sistema. Primero, la lente de la cámara no permite observar toda la plataforma, lo que provoca que el movimiento de la esfera quede restringido a un área reducida de la estructura. Segundo, las especificaciones del microcontrolador asociado a la cámara no permiten obtener un tiempo de muestreo y resolución tan buenos como en el caso de la pantalla resistiva. Por otro lado, es interesante comentar el tiempo mínimo de muestreo que permite el controlador, que puede ser relevante de cara a futuras implementaciones del sistema, con sensores o actuadores más rápidos. Como se ha observado en el

Capítulo 4 el microcontrolador permite reducir de forma considerable el tiempo de muestreo, lo que en teoría permitiría obtener tiempos de establecimiento del sistema mucho menores. Por ello, si se consiguiese obtener sensores y actuadores más rápidos, se podría obtener un tiempo de establecimiento inferior a 1 segundo.

Después de haber realizado la implementación de ambos controladores, se ha comparado la respuesta en ambos sistemas. Se ha observado que aún cuando estos presentan la misma estructura mecánica, la respuesta no es idéntica. Los motivos de esta diferencia pueden ser varios. Inicialmente, se ha supuesto que el motivo es la diferencia entre los dos sistemas de medidas, aunque este motivo se ha descartado debido a que ambos tiempos de muestreo que presentan son muy inferiores al tiempo de establecimiento del sistema, por lo que no debería afectar significativamente a su respuesta. Otro motivo puede ser el tamaño y el peso de la esfera. Aunque se ha observado en el modelado que las características de la esfera no influye en la respuesta ideal del sistema, el desarrollo físico realizado puede no adaptarse a ambos casos. Si se requiere un control más preciso del sistema, es necesario realizar un análisis físico más complejo.

Como conclusión final, este sistema permite su uso con fines educativos, siempre teniendo en cuenta que presenta ciertas limitaciones y que es un sistema complicado desde el punto de vista de la implementación, ya que presenta muchos componentes no ideales que afectan a su comportamiento. Por otro lado, hay que tener en cuenta que este sistema dispone de actuadores potentes y, dependiendo del tipo de controlador que se aplique, puede llegar a ser peligroso. Sin embargo, si se realiza un uso adecuado del mismo, se recomienda encarecidamente su implementación en un entorno educativo ya que permite a los alumnos realizar el control de un sistema real. De esta forma, pueden comprobar las diferencias entre un sistema simulado y uno real.

6.1. Líneas futuras

Dado que han surgido varios problemas que no han podido ser solucionados en este TFM, de cara a futuros desarrollos de este sistema, surge la necesidad de definir algunos puntos de mejora:

- **Identificación del sistema.** El método que se ha utilizado para realizar un análisis del comportamiento del sistema es mediante un modelado físico de este. Existe otro método basado en el análisis de la respuesta del sistema ante una entrada determinada. Este método consiste en analizar la respuesta y, mediante métodos numéricos, obtener una función de transferencia del sistema. Este tipo de métodos resultan muy útiles en casos en los que no se conoce el modelo o no se puede realizar un análisis físico del sistema. Resultaría interesante poder comprobar las diferencias en el modelo obtenido a partir de estos dos métodos.
- **Control de los motores.** Los actuadores que se han utilizado son servo-

motores. Como se ha comentado anteriormente, estos motores incorporan un controlador que permite realizar un control en posición. Sin embargo, este método reduce significativamente su tiempo de establecimiento y actuación. Por este motivo, con el objetivo de realizar un control más directo, se propone el uso de motores sin control, junto con algún sensor que permite su realimentación, como un potenciómetro o un *encoder*.

- **Mejora del sistema de la cámara.** Como se ha comentado anteriormente, es necesario realizar una modificación en el sensor de la cámara, con el objetivo de mejorar el ángulo de visión de esta, el periodo de muestreo y su resolución. Un posible método es el que se ha comentado en el Capítulo 3, en el que se propone la toma, procesado y análisis en el microprocesador donde se realiza el control. Esta solución puede suponer una reducción significativa del tiempo de muestreo y un aumento de la resolución.
- **Diseño de controladores más complejos.** En este TFM se han desarrollado controladores de dos teorías de control. Con el avance de la ingeniería han surgido nuevas y más complejas formas de realizar el control de los sistemas. Por este motivo, se propone el diseño e implementación de controladores más modernos con el objetivo de comparar su implementación y respuesta en un sistema real.
- **Filtrado de la señal de los sensores.** Aunque en el Anexo C se expone el método que se ha seguido para reducir el efecto del ruido en este sistema, surge la necesidad de implementar un método que permita eliminar completamente el ruido que proporcionan los elementos de medida. Por este motivo, un posible punto de mejora es el diseño de un filtro digital que permita reducir la dependencia de la respuesta del sistema al ruido.
- **Implementación de algoritmo de comunicación.** En este sistema se encuentra implementado un algoritmo de comunicación sencillo con el exterior, en el que se comunica la posición de la esfera y la acción de control y se recibe la referencia a la que se quiere llevar el sistema. Dado que este sistema se ha planteado para su uso en labores docentes, surge la necesidad de definir e implementar un algoritmo de comunicación que permita la selección del controlador, el ajuste de los parámetros del controlador o incluso modificar el tipo de sensor o actuador en cada momento.

Bibliografía

- [1] Nurbaiti Wahid and Mohd Fua'ad Rahmat. Pitch control system using lqr and fuzzy logic controller. In *2010 IEEE Symposium on Industrial Electronics and Applications (ISIEA)*, pages 389–394, 2010.
- [2] N. Minorsky. Directional stability of automatically steered bodies. *Journal of the American Society for Naval Engineers*, 34(2):280–309, 1922.
- [3] Thor I. Fossen. A survey on nonlinear ship control: from theory to practice. *IFAC Proceedings Volumes*, 33(21):1–16, 2000. 5th IFAC Conference on Manoeuvring and Control of Marine Craft (MCMC 2000), Aalborg, Denmark, 23-25 August 2000.
- [4] *Optimal Control Theory: Applications to Management Science and Economics*, pages 1–22. Springer US, Boston, MA, 2000.
- [5] Kemin Zhou and John Comstock Doyle. *Essentials of robust control*, volume 104. Prentice hall Upper Saddle River, NJ, 1998.
- [6] K.J. Astrom. Adaptive control around 1960. *IEEE Control Systems Magazine*, 16(3):44–49, 1996.
- [7] S. Bennett. *A History of Control Engineering 1800-1930*. Control Engineering Series 8. The Institution of Engineering and Technology, London, United Kingdom, 2008.
- [8] Microchip. AR1000 Series Resistive Touch Screen Controller. <https://ww1.microchip.com/downloads/en/DeviceDoc/40001393C.pdf>, 2009.
- [9] JLCPCB. PCB Capabilities. <https://jlcpcb.com/capabilities/Capabilities>.
- [10] Katsuhiko Ogata. *Modern Control Engineering*. Prentice Hall PTR, USA, 4th edition, 2001.
- [11] Heeseung Bang and Young Lee. Embedded Model Predictive Control for Enhancing Tracking Performance of a Ball and Plate System. *IEEE Access*, 7:39652–39659, March 2019.

- [12] John Hauser, Shankar Sastry, and Petar Kokotović. Nonlinear control via approximate input-output linearization: The ball and beam example. *Automatic Control, IEEE Transactions on*, 37:392–398, April 1992.
- [13] László Keviczky, Ruth Bars, Jenő Hetthéssy, and Csilla Bányász. *Control Engineering*. Advanced Textbooks in Control and Signal Processing. Springer Singapore, 2019.
- [14] FreeRTOS - Market leading RTOS (Real Time Operating System) for embedded systems with Internet of Things extensions. <https://www.freertos.org/>.
- [15] Kiam Heong Ang, G. Chong, and Yun Li. PID control system analysis, design, and technology. *IEEE Transactions on Control Systems Technology*, 13(4):559–576, July 2005. Conference Name: IEEE Transactions on Control Systems Technology.
- [16] B. Meenakshipriya and K. Kalpana. Modelling and Control of Ball and Beam System using Coefficient Diagram Method (CDM) based PID controller. *IFAC Proceedings Volumes*, 47(1):620–626, 2014.
- [17] Mohammad Keshmiri, Ali Fellah Jahromi, Abolfazl Mohebbi, Mohammad Amoozgar, and Wen-Fang Xie. Modeling and control of ball and beam system using model based and non-model based control approaches. *International Journal on Smart Sensing and Intelligent Systems*, 5:14, March 2012.
- [18] A. Zeeshan, N. Nauman, and M. Jawad Khan. Design, control and implementation of a ball on plate balancing system. In *Proceedings of 2012 9th International Bhurban Conference on Applied Sciences & Technology (IBCAST)*, pages 22–26, Islamabad, Pakistan, January 2012. IEEE.

Anexo A

Esquemáticos y circuito impreso de la tarjeta de expansión

En este Anexo se muestran con mayor detalle los esquemáticos y el circuito impreso de la tarjeta de expansión.

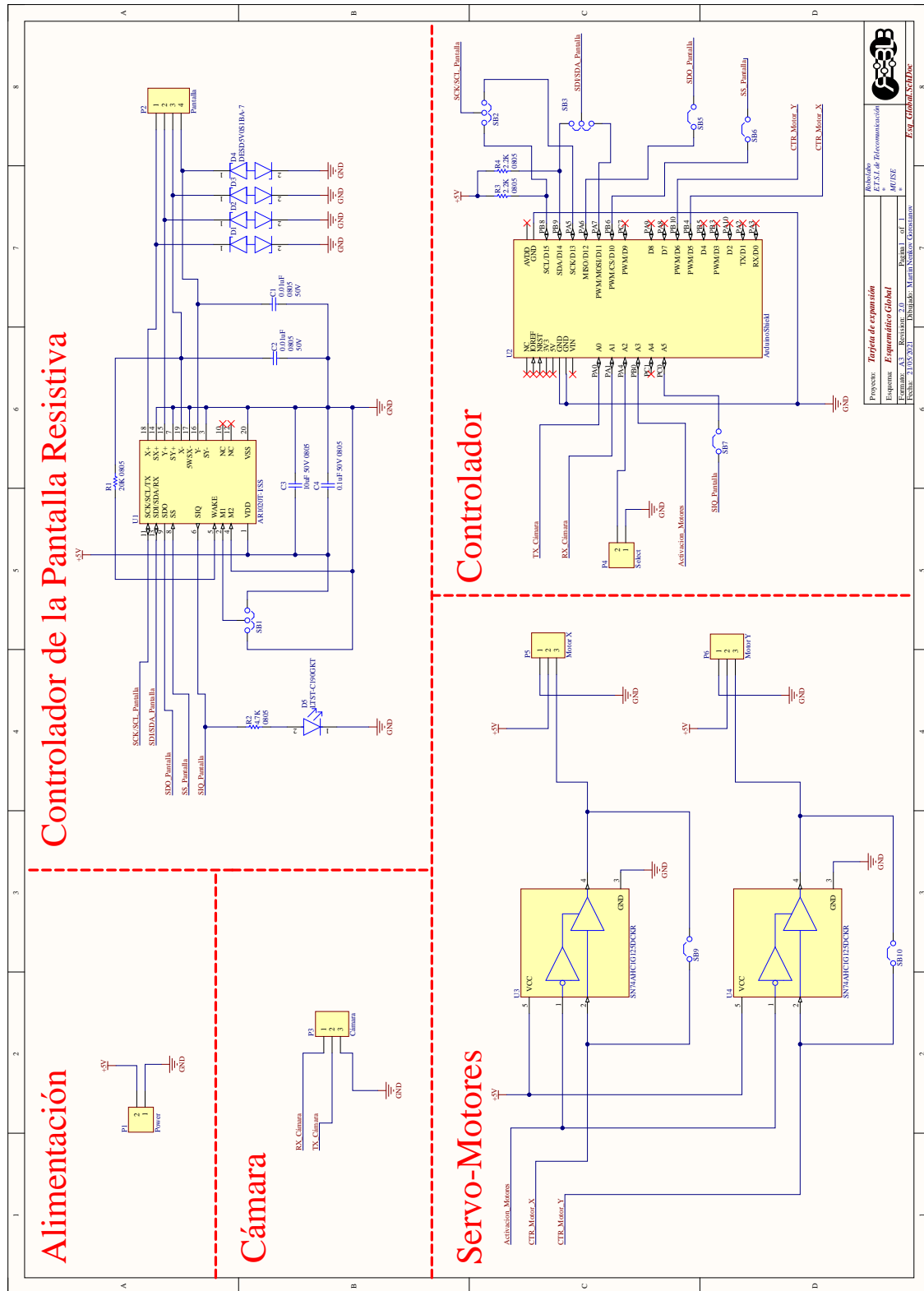


Figura A.1: Esquemático de la tarjeta de expansión.

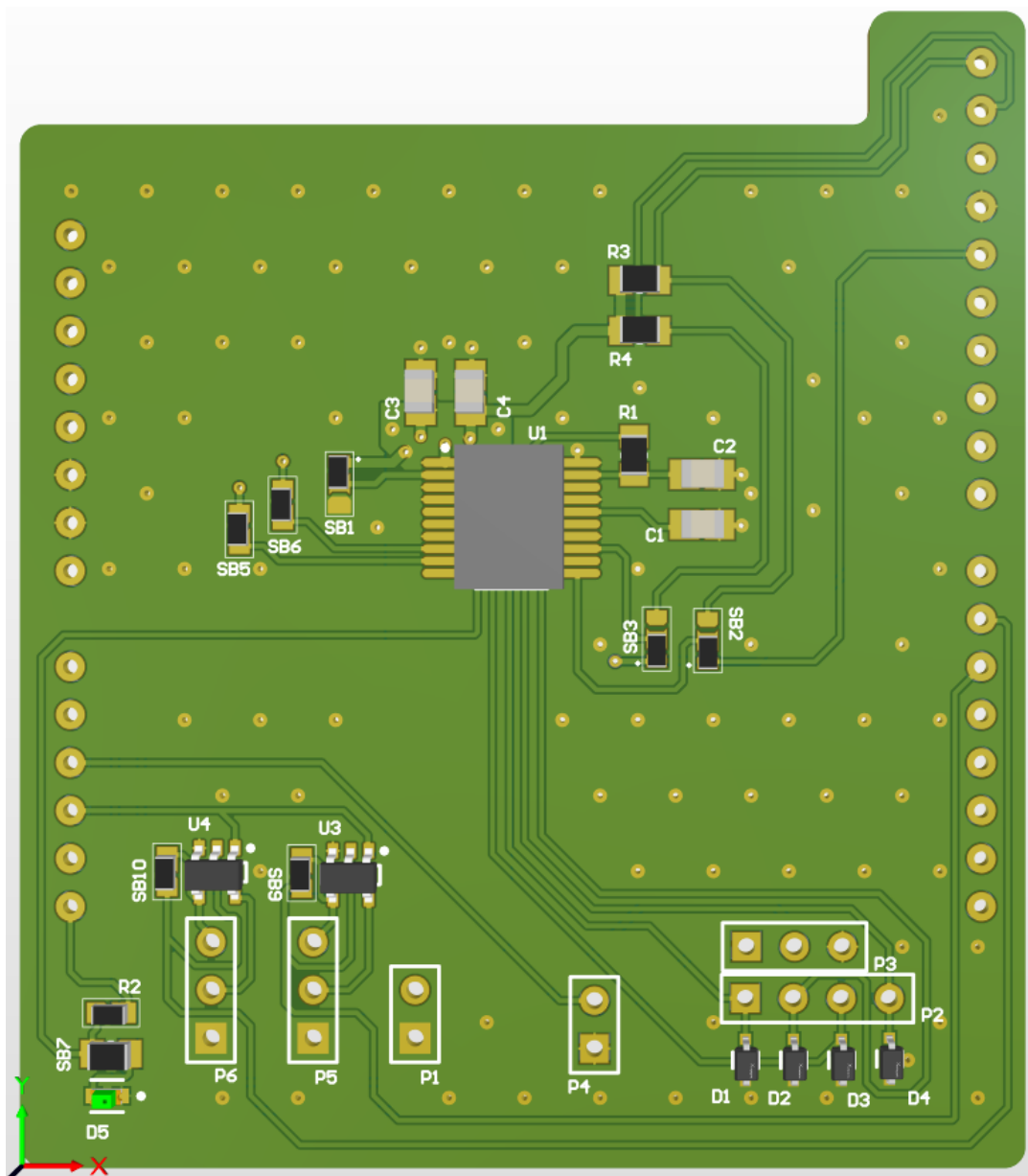


Figura A.2: Modelo 3D de la tarjeta de expansión.

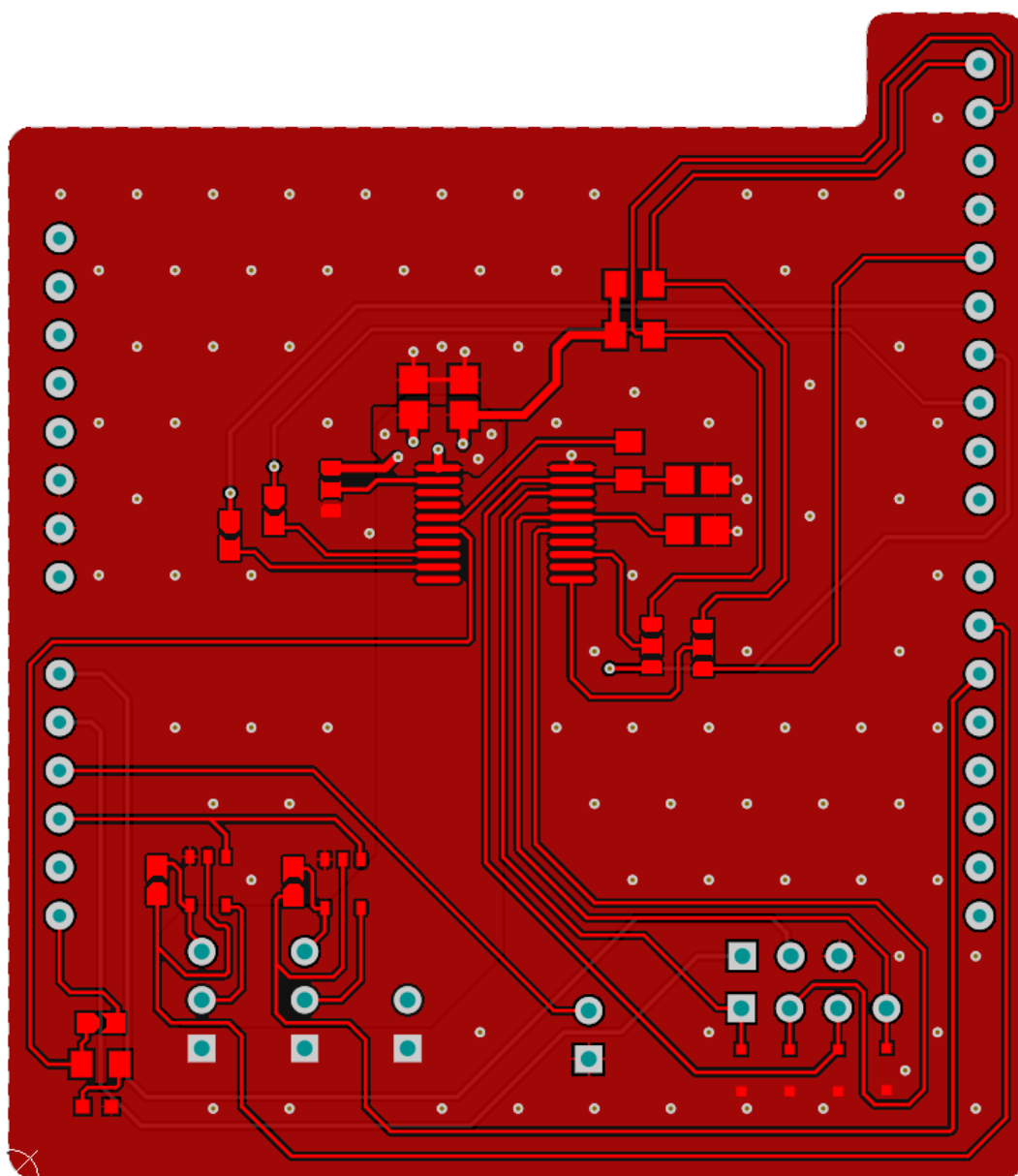


Figura A.3: Captura de la capa superior de la PCB.

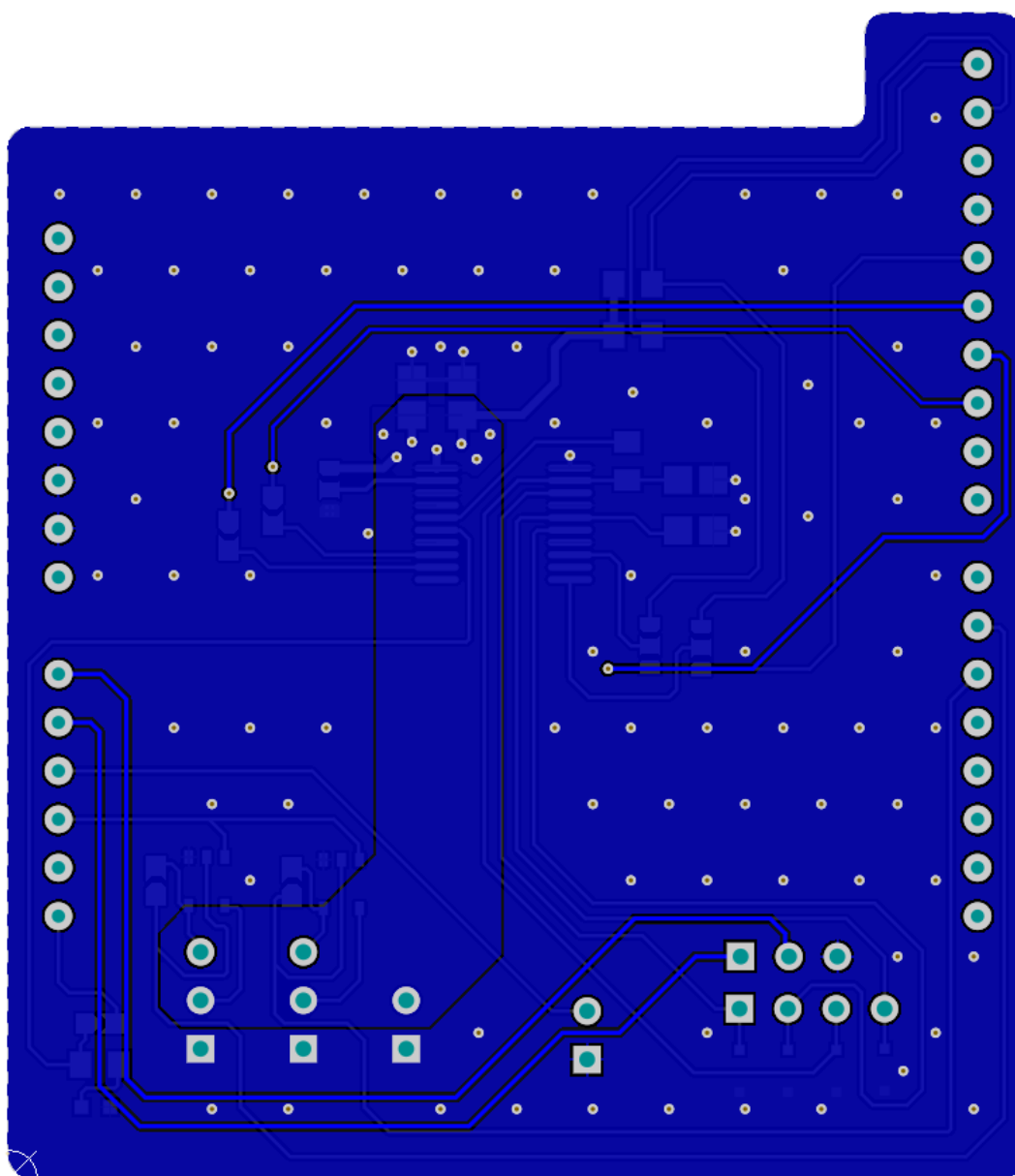


Figura A.4: Captura de la capa inferior de la PCB.

Anexo B

Diseño de las máquinas de estados finitas

B.1. Lectura de los datos de los sensores

Esta máquina será la encargada de realizar la lectura de los sensores cuando estos se encuentren listos. Después de recibirlos, realizará el procesamiento de los mismos previo a actualizar los datos del sistema.

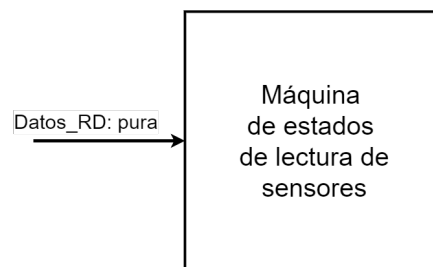


Figura B.1: Diagrama de entradas y salidas de la máquina de toma de datos de los sensores.

En la Figura B.1 se puede observar que esta máquina de estados presenta una sola entrada, *Datos_RD*. La activación de esta entrada significa que los sensores están listos para enviar los datos. Por otro lado, no es necesario que disponga de ninguna salida, ya que los datos son accesibles en todo el sistema, es decir, por todas las máquinas de estados. Con todo esto, se muestra a continuación el diagrama de estados:

Se observa que en el Diagrama de estados se han simplificado todas las tareas que se realizan al recibir los datos por medio de dos funciones, *Lectura_Datos* y *Actualización_Datos*. Estas dos funciones son las encargadas de comunicarse con los sensores y de procesar los datos obtenidos.

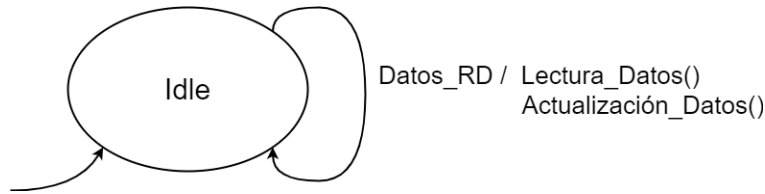


Figura B.2: Diagrama de estados de la máquina de toma de datos de los sensores.

B.2. Cálculo de la acción de control y actuación sobre servo-motores

Las funciones que deberá realizar esta máquina de estados son tres: recibir los datos de los sensores, realizar el cálculo de la acción de control necesaria en cada momento y transmitir este valor a los actuadores.

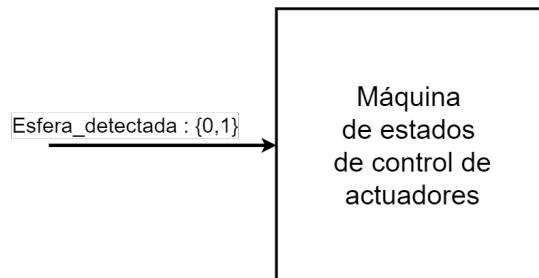


Figura B.3: Diagrama de entradas y salidas de la máquina de control.

Como se observa en la Figura B.3, en esta máquina de estados es necesario disponer de una entrada, *esfera_detectada*, que nos indique si se la esfera se encuentra sobre la plataforma y por tanto es necesario realizar un control sobre esta. Por otro lado, de forma similar a la máquina de estado anterior, no dispone de ninguna salida y todas las operaciones necesarias se realizan mediante las funciones *Lectura_Datos*, encargada de actualizar el valor de los datos locales con los actualizados por la máquina de estados anterior, *Cálculo_Acción_Control*, que aplica los controladores diseñados, y las funciones *Activación_Actuadores* y *Desactivación_Actuadores*, encargados de apagar los servo-motores cuando no sean necesarios. Con todo esto, se muestra el diagrama de estados en la Figura B.4.

Como se puede observar en este diagrama, esta máquina dispone de dos estados, en uno la esfera no se ha detectado, por lo que los actuadores se encuentran en reposo y no se realiza la computación de la acción de control y en el segundo, la esfera se encuentra sobre la plataforma, con lo que el sistema con total normalidad.

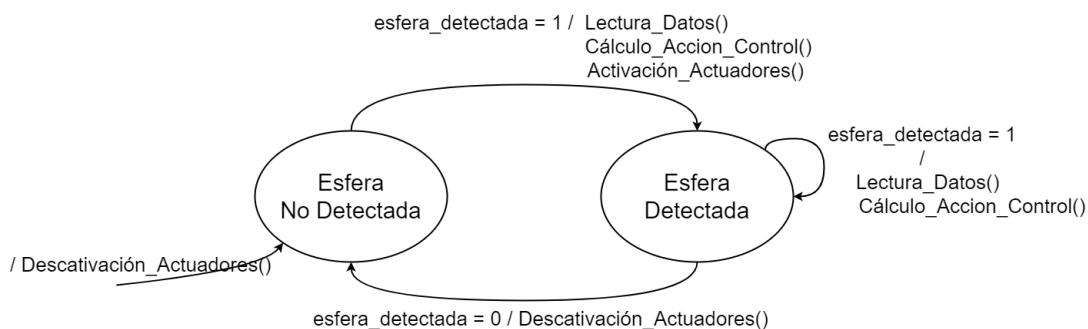


Figura B.4: Diagrama de estados de la máquina de control.

B.3. Envío de datos del sistema al exterior

Esta será la máquina de estados encargada de transmitir el estado del sistema al exterior. Para ello, se ha decidido que dispondrá de un *buffer*, en el que se almacenarán los datos. Cuando este *buffer* se encuentre lleno, se procederá a enviar un bloque de datos. A continuación, se muestra la interfaz de la máquina de estados (ver Figura B.5) y su diagrama (ver Figura B.6).

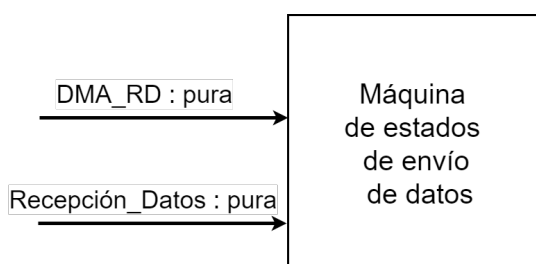


Figura B.5: Diagrama de entradas y salidas de la máquina de envío de datos.

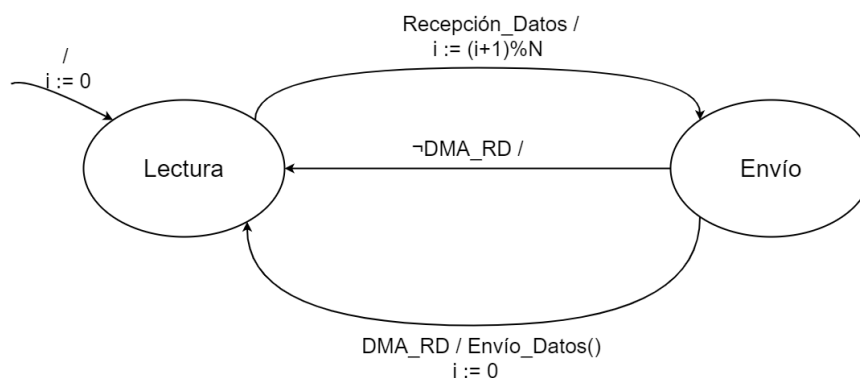


Figura B.6: Diagrama de estados de la máquina de envío de datos.

En esta máquina de estados se realiza el conteo de los datos que se encuentran almacenados mediante la variable, *i*. Cada vez que se activa el *flag* de que se ha

recibido un dato, aumentará en una unidad el contador de datos y realizará el envío si el controlador DMA está disponible. Si el controlador está disponible, el contador de datos almacenados se anulará, comenzando de nuevo a llenarse el *buffer*. Como se puede observar en la Figura B.6, se ha supuesto que el controlador DMA se encontrará disponible en algún momento durante el llenado del *buffer*, ya que si esto no sucede los datos que se encontraban almacenados hasta el momento se perderán.

B.4. Recepción de datos del exterior

De forma similar a la máquina encargada de leer los sensores, esta máquina será la encargada de recibir los comandos del exterior y de realizar las acciones sobre el sistema pertinentes. En este caso, presenta una entrada, la cual indica que ha llegado una nueva instrucción, *Comando_Nuevo*, y una función que será la encargada de realizar las tareas que correspondan a cada orden, *Decodificar_Comando*. De esta forma, se muestran la interfaz de entradas (ver Figura B.7) y salidas y el diagrama de estados de esta máquina (ver Figura B.8).

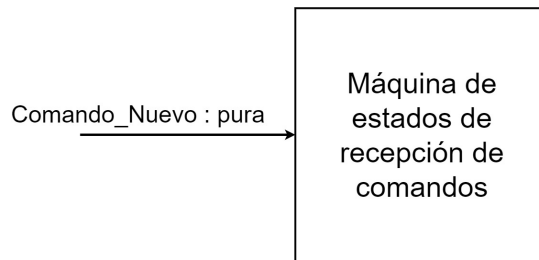


Figura B.7: Diagrama de entradas y salidas de la máquina de recepción de comandos.

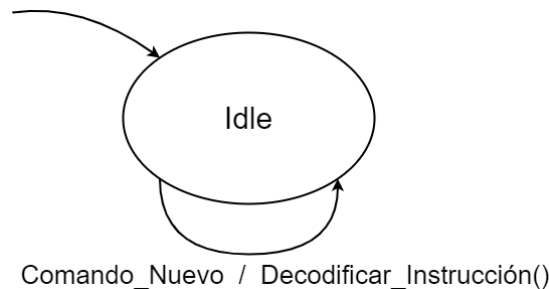


Figura B.8: Diagrama de estados de la máquina de envío de recepción de comandos.

Anexo C

Análisis del periodo de diferenciación

El ruido produce un efecto negativo en los sistemas de control, en algunos casos modificando su respuesta de forma significativa. Por este motivo, es necesario aplicar ciertas técnicas que reduzca este efecto. En los sistemas discretos, este efecto está particularmente presente cuando se realiza la diferenciación. Al realizar una discretización de una señal continua, hay instantes en los que se produce una variación abrupta de la señal. En estos instantes, si se realiza la derivada de esta señal, se observa que se producen variaciones anormales. Este fenómeno produce un compartimiento incorrecto del sistema, pudiendo causar un fallo en el control.

Para minimizar este efecto, se ha propuesto que el cálculo de las componentes proporcional e integral se realice con un periodo diferente de la componente derivativa. Se observa en la Ecuación C.1 la forma en la que se realiza la computación de la derivada discreta de una señal continua, $x(t)$.

$$\delta(k) = \frac{x(k) - x(k - n)}{T_d} \quad (\text{C.1})$$

En esta ecuación se obtiene la derivada(δ) en un determinado instante k , a partir del valor muestreado en ese instante y un instante n -veces anterior. Por lo tanto, se define el periodo de diferenciación(T_d) de la siguiente forma:

$$T_d = n \cdot T_m \quad (\text{C.2})$$

Siendo T_m el periodo con el que se realiza el muestro de la señal continua.

En la Figura C.1 se muestra el efecto que produce en la señal de control la variación del periodo de diferenciación. Se observa que con un periodo de diferenciación superior a los 25ms, se obtiene una señal de control que no presenta variaciones abruptas. Por

otro lado, se aprecia que con un mayor periodo, la señal que se obtiene presenta una mayor similitud con una señal continua.

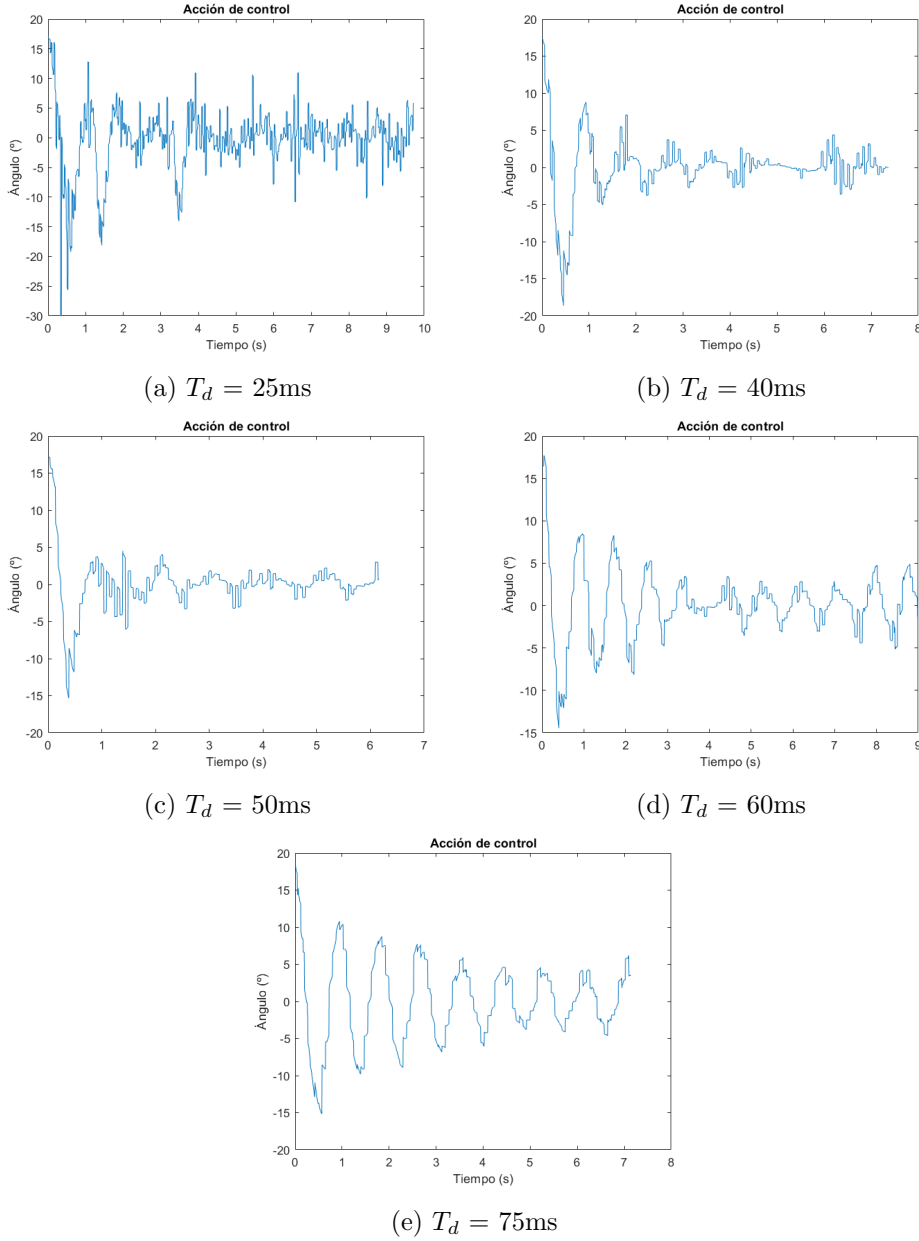


Figura C.1: Señal de control del sistema ante diferentes periodos de muestreo.

En el análisis del efecto del periodo de diferenciación en la señal de control no se ha observado ningún indicio con el que se pueda concluir que un periodo presenta unos resultados superiores a otro. Por este motivo, es necesario realizar un análisis de este efecto en la respuesta del sistema. Es necesario comentar que para este análisis se ha analizado la respuesta del sistema ante un controlador PD, cuya respuesta ideal ante una entrada escalón unitario se puede observar en la Figura C.2.

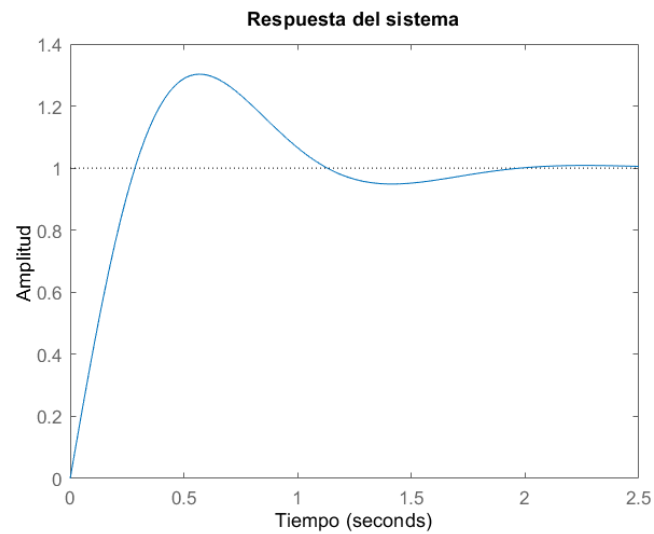


Figura C.2: Respuesta del sistema ante el PD utilizado para el análisis del efecto del periodo de diferenciación.

Se observa en la Figura C.3 la respuesta del sistema con diferentes tiempos de diferenciación. A partir de la repuestas observadas se puede concluir que con un periodo de 40ms se reduce significativamente la diferencia entre la simulación (ver Figura C.2) y la realidad. En la respuesta real, se sigue observando una oscilación en régimen permanente, pero la respuesta en el régimen transitorio es similar a la ideal, presentando una sobreoscilación de un 20 % y un tiempo de establecimiento de aproximadamente 2 segundos.

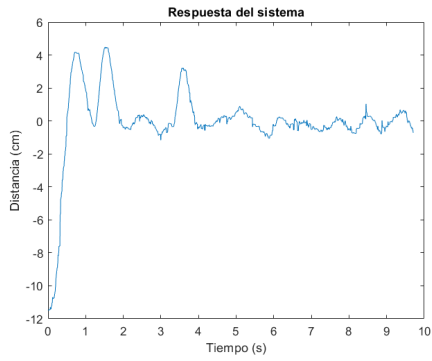
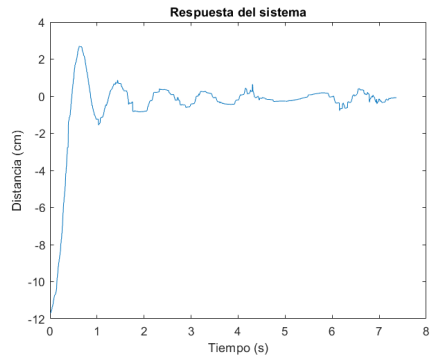
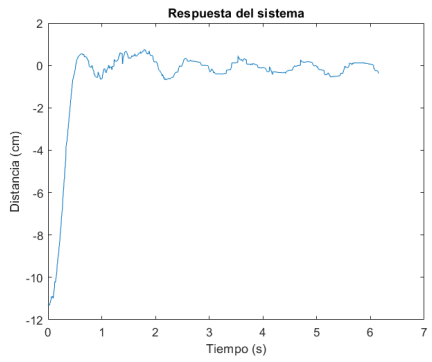
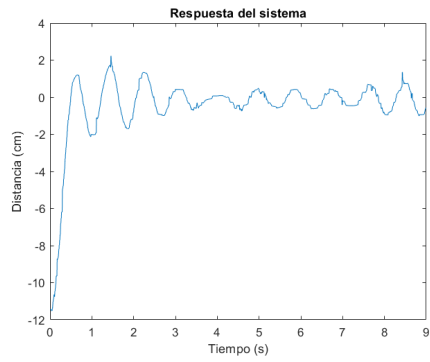
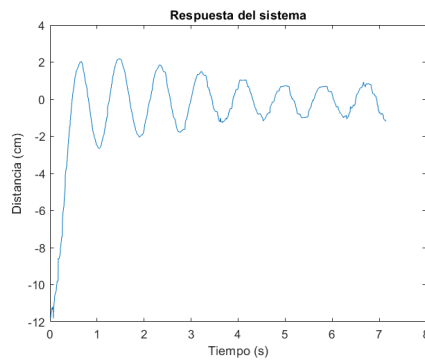
(a) $T_d = 25\text{ms}$ (b) $T_d = 40\text{ms}$ (c) $T_d = 50\text{ms}$ (d) $T_d = 60\text{ms}$ (e) $T_d = 75\text{ms}$

Figura C.3: Respuesta del sistema ante diferentes periodos de muestreo.

Anexo D

Aspectos éticos, económicos, sociales y ambientales

El objetivo principal de este TFM es aplicar técnicas de control a un sistema de dos grados de libertad y diseñar los componentes electrónicos necesarios para su uso. Se ha planteado este objetivo ya que se pretende replicar el trabajo realizado en laboratorios para educar al alumnado en la ingeniería de control.

Este TFM se ha planteado para que presente el mayor impacto social posible sobre los alumnos y profesores de las asignaturas de ingeniería de control de la Escuela, ya que son ellos para los que inicialmente presentan mayor interés por los temas que se han tratado. En este documento se ha realizado una explicación detallada de los pasos que se han seguido para su realización, por lo que la replicación de los resultados obtenidos es sencilla. Las tareas realizadas en este proyecto se pueden incluir en el ámbito de una asignatura con el objetivo de que el alumno se familiarice con los procesos necesarios para el diseño e implementación de un controlador sobre un sistema real.

El impacto económico que tiene este TFM es que presenta una solución de bajo coste, como se puede observar en el Anexo E, para la implementación de una maqueta sobre la que poder realizar pruebas de distintos controladores, además de poder comparar los resultados obtenidos en una simulación frente a los reales.

En cuanto al impacto medioambiental de la solución propuesta, se puede comentar que se han aplicado técnicas de ahorro de energía que permiten desactivar complementemente el sistema cuando este no se encuentra en activo. Esto permite reducir la energía que consume este significativamente. Por otro lado, la mayoría de componentes que se han utilizado para la realización de este TFM han sido adquiridos en Asia, por lo que se desconoce el impacto medioambiental de estos. Hay que añadir también el tiempo de vida de los estos dispositivos es alto, que junto con una fiabilidad alta permite una utilización por largos periodos de tiempo, lo reduce su impacto en el medioambiente.

Anexo E

Presupuesto económico

Se divide el coste del proyecto en dos grupos según su origen: el coste del personal que ha intervenido en el desarrollo del TFM y el coste del material empleado.

- **Personal:** los costes equivalentes al trabajo realizado por el tutor del TFM, junto con las horas de dedicación del alumno (ver Tabla E.1).

	Coste horario (€)	Horas	Total (€)
Director del trabajo	60	30	1800
Estudiante de ingeniería	30	450	13500
TOTAL			15300

Tabla E.1: Costes de personal.

- **Costes de recursos materiales:** los costes relacionados con el material empleado en el TFM (ver Tabla E.2). Estos corresponden con los componentes empleados en el desarrollo de la tarjeta de expansión, junto con el coste de la estructura, los sensores y actuadores empleados en el desarrollo del TFM.

Finalmente, se obtiene el coste total de la ejecución del proyecto está formado por el coste del personal junto con el coste de los recursos materiales (ver Tabla E.3).

	Tiempo de vida (años)	Uds.	Coste (€)	Amortización (€/mes)	Uso (meses)	Total (€)
Tarjeta de expansión	5	5	0.6	0.13	6	0.8
Componentes electrónicos	5	1	20	0.13	6	2
Estructura mecánica	5	2	50	1.67	6	10
Pantalla resistiva	5	1	20	0.13	6	2
Cámara	5	1	30	0.5	6	3
Actuadores	5	4	5	0.13	6	2
TOTAL						19.8

Tabla E.2: Costes de recursos materiales.

	Coste
Costes de personal	15300€
Costes de material	19.8€
Subtotal	15319.8€
IVA	3217.16€
Total	18537.96€

Tabla E.3: Costes totales.