

UNIVERSIDAD POLITÉCNICA DE MADRID

**ESCUELA TÉCNICA SUPERIOR
DE INGENIEROS DE TELECOMUNICACIÓN**



**GRADO UNIVERSITARIO EN INGENIERÍA EN
TECNOLOGÍAS Y SERVICIOS DE
TELECOMUNICACIÓN**

TRABAJO FIN DE GRADO

**MODELADO E IMPLEMENTACIÓN DE UN SISTEMA
DE CONTROL DE UNA ESTRUCTURA DE
BALANCEO CON DOS GRADOS DE LIBERTAD**

JORGE QUIJORNA SANTOS

2022

UNIVERSIDAD POLITÉCNICA DE MADRID

**ESCUELA TÉCNICA SUPERIOR
DE INGENIEROS DE TELECOMUNICACIÓN**



**GRADO UNIVERSITARIO EN INGENIERÍA EN
TECNOLOGÍAS Y SERVICIOS DE
TELECOMUNICACIÓN**

TRABAJO FIN DE GRADO

**MODELADO E IMPLEMENTACIÓN DE UN SISTEMA
DE CONTROL DE UNA ESTRUCTURA DE
BALANCEO CON DOS GRADOS DE LIBERTAD**

Autor

JORGE QUIJORNA SANTOS

Tutor

ÁLVARO GUTIÉRREZ MARTÍN

2022

Resumen

Actualmente, es innegable que la tecnología está muy presente en nuestro día a día y los sistemas de control están desempeñando un papel fundamental en el desarrollo de esta tecnología. Por ello, como ingenieros, debemos ser capaces de diseñar estos sistemas, prever cómo van a funcionar antes de implementarlos y entender por qué funcionan de la manera que lo hacen una vez han sido implementados.

Por esta razón, el presente TFG se centra en el diseño, modelado e implementación de un sistema de balanceo con dos grados de libertad. En concreto, se va a trabajar en un sistema Ball And Plate. El objetivo de este sistema es controlar la posición de una esfera sobre una superficie para que nunca se salga de la misma y alcance el centro de ésta lo más rápido posible. Para ello, en primer lugar se realizará un análisis acerca de los componentes que vamos a necesitar y las prestaciones que nos van a dar. Posteriormente, se realizará un modelado con el objetivo de obtener la función de transferencia del sistema. En tercer lugar, se diseñará e implementará un controlador tanto en un sistema real como en un sistema simulado que permitirá actuar sobre la posición de la esfera. Y, finalmente se hará un estudio sobre los resultados obtenidos.

Palabras clave: grado de libertad, sistema de control, controlador P, controlador PD, referencia, tiempo de establecimiento

Abstract

Nowadays, it is well-known that technology is very present in our day to day and control systems are being very important in the development of this technology. In fact, as engineers, we must be able to design these systems, to foresee how they are going to work after and before their implementation.

For this reason, the present Thesis is focus on the design, modelling and implementation of a two degrees of freedom system. In particular, we are going to work in a system called Ball And Plate. The aim of this system is to control the position of a ball which is over a plate avoiding that the ball leaves the plate and getting that it arrives to the plate's center as soon as possible. To achieve this purpose, firstly, an analysis is going to be done about the required components and their capabilities. Afterwards, we are going to work in the modelling of the system with the aim of obtaining the system's transfer function. In third place, we will design and implement a controller for a real and for a simulate system which is going to allow us to control the position of the ball. And, lastly, we are going to do an analysis about the obtained results.

Keywords: degree of freedom, control system, P controller, PD controller, reference, settling time

Agradecimientos

Me gustaría utilizar estas líneas para agradecer, en primer lugar, a mi familia por todo el apoyo tanto moral como económico que me han dado a lo largo de la carrera. Sin ellos no habría sido posible.

En segundo lugar me gustaría tener unas palabras de agradecimiento para Blanca, ya que sus clases en la asignatura “Sistemas Electrónicos de Control” me han ayudado a entender partes importantes de este TFG. También quería agradecerle a Miguel su ayuda. Siempre está en el laboratorio dispuesto a ayudar cuando alguien le necesita y su ayuda ha sido de vital importancia para la realización de este TFG.

En tercer lugar, a mi tutor Álvaro, por haber confiado en mí desde el primer momento a pesar de que mis conocimientos de partida era nulos. También estoy muy agradecido por la paciencia que ha tenido conmigo y lo abierto que ha estado siempre a tener una tutoría cuando la he necesitado.

Por último, me gustaría tener unas palabras de agradecimiento también con el resto de profesores que he tenido a lo largo de la carrera. Los ha habido más y menos exigentes, pero todos han contribuido a que mi formación a lo largo de estos años haya sido la que me esperaba cuando accedí a esta escuela y a que a día de hoy sea la persona que soy.

Índice general

Resumen	v
Abstract	vii
Agradecimientos	ix
Índice General	xi
Índice de Figuras	xiii
Índice de Tablas	xv
Lista de Acrónimos	xvii
1. Introducción	1
1.1. Introducción a los sistemas de control	1
1.2. Diferencia entre un sistema en lazo abierto y un sistema en lazo cerrado	2
1.3. Resumen de objetivos	3
1.4. Organización y planificación previstas	3
1.5. Estructura del documento	4
2. Descripción del hardware del sistema	5
2.1. Cámara	6
2.2. Microcontrolador	7
2.3. Actuadores	8
2.4. Comunicación entre la cámara y el microcontrolador	10
2.4.1. Bus SPI	10
2.4.1.1. Polaridad y Fase del reloj utilizando el protocolo SPI	12
2.4.2. Puerto Serie UART	13
3. Modelado	15
3.1. Introducción	15
3.2. Modelado de un sistema Ball And Plate	16
3.3. Transformada de Laplace	18
3.4. Modelado en Simulink	19

4. Diseño de controladores	23
4.1. Controlador proporcional (P)	23
4.2. Análisis del problema de seguimiento al escalón, la rampa y la parábola utilizando un controlador proporcional (P)	24
4.3. Controlador proporcional derivativo (PD)	25
4.4. Análisis del problema de seguimiento al escalón, la rampa y la parábola utilizando un controlador proporcional derivativo (PD)	25
4.5. Sintonización de parámetros	26
5. Implementación	29
5.1. Código desarrollado en <i>OpenMV IDE</i>	29
5.2. Código desarrollado en <i>STM32CubeIDE</i>	30
6. Resultados del experimento	31
6.1. Controlador P	31
6.1.1. Comparativa entre el comportamiento del sistema real y el sistema ideal	31
6.2. Controlador PD	35
6.2.1. Comparativa entre el comportamiento del sistema real y el sistema ideal	35
7. Conclusiones y líneas futuras	41
7.1. Conclusiones	41
7.2. Líneas futuras	42
Bibliography	43
A. Esquema de montaje del sistema	45
B. Aspectos éticos, económicos, sociales y ambientales	47
C. Presupuesto económico	49

Índice de figuras

1.1. Sistema en lazo abierto	2
1.2. Sistema en lazo cerrado	2
2.1. Diagrama que muestra los bloques que componen el sistema	5
2.2. Sensor <i>OpenMV Cam H7 Plus</i>	6
2.3. Imágenes del sistema real y el sistema visto desde la cámara.	7
2.4. Microcontrolador <i>STM32F446RE</i>	8
2.5. <i>Servomotor S3001</i>	9
2.6. Relación entre el ciclo de trabajo de la señal PWM y la rotación del servomotor.	10
2.7. Comunicación SPI	11
2.8. Ejemplo de transferencia de datos full-duplex utilizando el protocolo SPI [1].	11
2.9. Ejemplo de transferencia de datos half-duplex utilizando el protocolo SPI [1].	12
2.10. Comunicación UART.	13
3.1. Sistema en lazo cerrado.	15
3.2. Representación de la estructura Ball And Plate [2].	16
3.3. Diagrama de bloques del sistema simulado en Simulink.	19
3.4. Simulación.	20
3.5. Representación de la posición de la esfera sobre los ejes X e Y en la simulación	20
3.6. Representación de la posición de la esfera sobre los ejes X e Y en el sistema real.	21
4.1. Sistema con controlador P.	23
4.2. Sistema con controlador PD.	25
4.3. Ejemplo de respuesta al escalón unidad de un sistema de segundo orden [3].	26
6.1. Diagrama de bloques del sistema simulado en Simulink con controlador P	31
6.2. Representación de la posición de la esfera sobre los ejes X e Y en la simulación utilizando controlador P con una $K_p = 0,1$	32
6.3. Respuesta del sistema real utilizando un controlador P con $K_p = 0,1$	32
6.4. Representación de la posición de la esfera sobre los ejes X e Y en la simulación utilizando controlador P con una $K_p = 0,25$	33

6.5. Respuesta del sistema real utilizando un controlador P con $K_p = 0,25$.	33
6.6. Representación de la posición de la esfera sobre los ejes X e Y en la simulación utilizando controlador P con una $K_p = 2$	34
6.7. Respuesta del sistema real utilizando un controlador P con $K_p = 2$. . .	34
6.8. Diagrama de bloques del sistema simulado en Simulink con controlador PD	35
6.9. Representación de la posición de la esfera sobre los ejes X e Y en la simulación utilizando controlador PD con una $K_p = 0,386$ y $K_D = 0,2154$	36
6.10. Respuesta del sistema real utilizando un controlador PD con $K_p = 0,386$ y $K_D = 0,2154$	36
6.11. Representación de la posición de la esfera sobre los ejes X e Y en la simulación utilizando controlador PD con $K_{px} = 2$, $K_D = 0,2154$ y $K_{py} = 0,386$	37
6.12. Respuesta del sistema real utilizando un controlador PD con $K_{px} = 2$, $K_D = 0,2154$ y $K_{py} = 0,386$	37
6.13. Representación de la posición de la esfera sobre los ejes X e Y en la simulación utilizando controlador PD con $K_p = 0,386$, $K_{DX} = 0,8$ y $K_{DY} = 0,2154$	38
6.14. Respuesta del sistema real utilizando un controlador PD con $K_p = 0,386$, $K_{DX} = 0,8$ y $K_{DY} = 0,2154$	38
6.15. Representación de la posición de la esfera sobre los ejes X e Y en la simulación utilizando controlador PD con $K_{px} = 0,05$, $K_D = 0,2154$ y $K_{py} = 0,386$	39
6.16. Respuesta del sistema real utilizando un controlador PD con $K_{px} = 0,05$, $K_D = 0,2154$ y $K_{py} = 0,386$	40
A.1. Conexiones entre los distintos componentes de nuestro sistema Ball And Plate.	45

Índice de tablas

2.1. Tabla resumen sobre los posibles modos del CLK en función de la polaridad y la fase.	12
C.1. Costes de personal.	49
C.2. Costes de recursos materiales.	49
C.3. Costes totales.	50

Lista de Acrónimos

ETSIT: Escuela Técnica Superior de Ingenieros de Telecomunicación.

UPM: Universidad Politécnica de Madrid.

TFG: Trabajo Fin de Grado.

GPS: Global Positioning System.

2D: 2 Dimensiones.

ARM: Advanced RISC Machine.

RISC: Reduced Instruction Set Computing.

SRAM: Static Random Access Memory.

E/S: Entrada/Salida.

SPI: Serial Peripheral Interface.

I2C: Inter-Integrated Circuit.

CAN: Controller Area Network.

UART: Universal Asynchronous Receiver-Transmitter.

RGB: Red Green Blue.

DSP: Digital Signal Processing.

MPU: Memory Protection Unit.

DMA: Direct Memory Access.

FIFO: First In First Out.

USART: Universal Synchronous/Asynchronous Receiver-Transmitter.

IDE: Integrated Development Environment.

Capítulo 1

Introducción

La robótica está desempeñando un papel fundamental en el desarrollo de la tecnología y en hacer más sencillo nuestro día a día. Tanto es así que cada vez es más frecuente salir a la calle y encontrarnos con uno de estos sistemas, o incluso tenerlos en nuestra propia casa. Por ejemplo, el robot *Starship* se encarga desde 2016 de llevar pedidos de manera autónoma mediante el uso de cámaras, radar y GPS, que le permiten hacer entregas en distintos puntos de manera eficiente [4]. Otro ejemplo que todos conocemos es el famoso aspirador *Roomba*, que funciona de manera autónoma gracias al uso de sensores que le ayudan a detectar obstáculos y, de esta forma, poder evitarlos. Por último, el robot *Peanut* ha sido de vital importancia durante la pandemia, pues se encargaba de llevar comida a personas en cuarentena, evitando así la exposición de otras personas al virus [5].

Como se puede observar, cada vez aparecen más sistemas de este tipo que llegan para quedarse y, como ingenieros, es interesante entender cómo funcionan y ser capaces de diseñarlos y controlarlos. Los sistemas de control son una pieza fundamental en la robótica, y es el campo en el que se centra este Trabajo Fin de Grado.

1.1. Introducción a los sistemas de control

En el presente Trabajo de Fin de Grado se va a diseñar, modelar e implementar un sistema de control de una estructura de balanceo con dos grados de libertad. El concepto de grado de libertad se refiere al número máximo de variables que deben ser consideradas para tener un control total sobre un sistema. En concreto, se va a trabajar en un sistema comúnmente conocido como “*Ball And Plate*” que, de manera autónoma, es capaz de controlar la posición de una esfera sobre una superficie actuada con 2 grados de libertad, la posición de la esfera en el eje X y la posición de la esfera en el eje Y. El funcionamiento de dicho sistema sería el siguiente: si se coloca una esfera sobre una superficie, el sistema debería ser capaz de mantener la esfera siempre en el centro de la superficie de forma autónoma, es decir, si la esfera se aproxima a un borde del plato, éste se debería balancear de tal forma que la esfera nunca se salga y siempre esté lo más cercana posible al centro. El comportamiento ideal sería que la esfera estuviera siempre en el centro de la superficie o, en el caso de

haber sufrido un desplazamiento, que tarde el menor tiempo posible en alcanzar esta posición. En posteriores capítulos se entrará más en detalle para entender qué es necesario considerar para conseguir este funcionamiento.

1.2. Diferencia entre un sistema en lazo abierto y un sistema en lazo cerrado

En este punto del proyecto conviene definir qué es un sistema en lazo abierto y un sistema en lazo cerrado y tener claras las diferencias entre ambos.

Un sistema en lazo abierto se caracteriza por que la salida $y(t)$ no tiene efecto sobre la señal de control $u(t)$, como se puede observar en la Figura 1.1. Se pueden diferenciar 2 bloques, $G_c(s)$ y $G(s)$, y 3 señales, $r(t)$, $u(t)$ e $y(t)$. $G_c(s)$ corresponde a la función de transferencia del controlador del sistema y variará en función del controlador que decidamos utilizar. $G(s)$ es la función de transferencia de nuestro sistema. $r(t)$ es la señal de referencia que introducimos en la entrada. $u(t)$ es la señal de salida del controlador e $y(t)$ es la señal de salida del sistema.

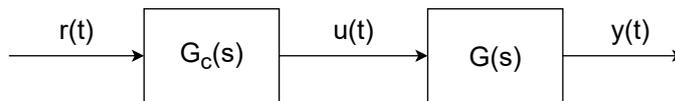


Figura 1.1: Sistema en lazo abierto

Un sistema en lazo cerrado es aquel en el que la salida $y(t)$ tiene efecto sobre la señal de control $u(t)$, como se puede observar en la Figura 1.2. Se pueden distinguir los mismos bloques que en el sistema en lazo abierto pero en este caso se observa una señal más, $e(t)$.

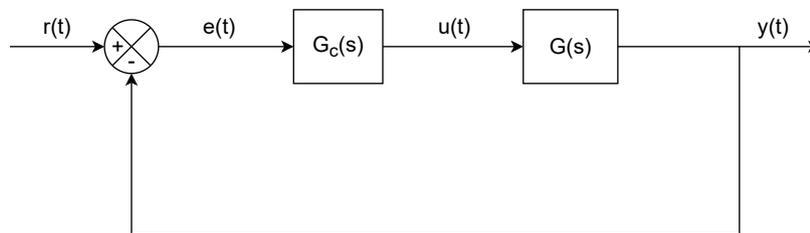


Figura 1.2: Sistema en lazo cerrado

En estos sistemas, se quiere que la salida $y(t)$ sea lo más parecida posible a la referencia que se introduce en la entrada $r(t)$. La señal $e(t)$ es la diferencia entre ambas señales, y es la señal que entrará en el controlador. Al tener un sistema realimentado,

a medida que pase el tiempo en los experimentos, la salida $y(t)$ se parecerá cada vez más a la señal de entrada $r(t)$ y, con esto, el error será cada vez menor por lo que el control que se aplique en el bloque del controlador para conseguir la igualdad entre ambas señales también será cada vez más pequeño.

En este TFG se va a trabajar con un sistema en lazo cerrado. Por ello, resulta necesario profundizar un poco más y saber cuál es la función de transferencia del conjunto, que se puede obtener trabajando en el dominio de Laplace de la siguiente manera:

$$Y(s) = E(s) \cdot G_c(s) \cdot G(s) \quad (1.1)$$

$$E(s) = R(s) - Y(s) \quad (1.2)$$

Sustituyendo el valor de $E(s)$ de la Ecuación 1.2 en la Ecuación 1.1 y sabiendo que $H(s) = \frac{Y(s)}{R(s)}$, se obtiene que la función de transferencia del sistema en lazo cerrado es:

$$H_{lc}(s) = \frac{R(s)}{Y(s)} = \frac{G_c(s)G(s)}{1 + G_c(s)G(s)} \quad (1.3)$$

1.3. Resumen de objetivos

El propósito de este TFG consiste en modelar, controlar e implementar un sistema de control de balanceo de 2 grados de libertad. A partir de la información que se recibe a través de una cámara, la posición de una esfera sobre un plato, se controlará la posición de unos actuadores para que una esfera nunca se salga de la superficie.

Para lograr este objetivo, el sistema estará compuesto por: una cámara, que monitorizará la posición de la esfera en todo momento; un microcontrolador STM32 que recibirá información de la cámara y controlará el comportamiento de unos actuadores; y, dos servomotores, que servirán para mover la plataforma en la dirección de los ejes X e Y en función de la posición en la que se encuentre la esfera.

1.4. Organización y planificación previstas

A continuación, se muestra la planificación seguida:

- Familiarización con el entorno de desarrollo de STM32, así como con el microcontrolador y los servomotores.
- Modelado del sistema
- Monitorización de la posición de la esfera sobre el plato mediante una cámara y transmisión de la información obtenida al microcontrolador.

- Diseño e implementación de un controlador para el sistema.
- Pruebas y ajustes del sistema.
- Diseño de placa de circuito impreso para la integración de todos los componentes.

1.5. Estructura del documento

Este documento se divide en 7 Capítulos donde se detallan los pasos que se han seguido para la realización del TFG así como otros aspectos relevantes.

En el Capítulo 1 se ha hecho una breve introducción a los sistemas de control y se han expuesto la motivación y los objetivos de este TFG.

En el Capítulo 2 se ha descrito con detalle el hardware necesario para el correcto funcionamiento del sistema, así como la comunicación entre los distintos componentes.

En el Capítulo 3 se expone el procedimiento que se ha seguido para la obtención de la función de transferencia del sistema. En este Capítulo también se ha hecho un breve análisis y una primera simulación sin control.

En el Capítulo 4 se describe la metodología empleada para el diseño de los controladores que formarán parte de nuestro sistema.

En el Capítulo 5 se expone cómo se han implementado los controladores diseñados y los protocolos de comunicación entre los componentes.

En el Capítulo 6 se hace un estudio de los resultados obtenidos para el sistema real y de su similitud con los resultados obtenidos para el sistema simulado.

Finalmente, en el Capítulo 7 se plantean las conclusiones extraídas tras la realización del TFG, así como líneas futuras.

Capítulo 2

Descripción del hardware del sistema

Aunque posteriormente se entrará en más detalles, conviene tener en cuenta cómo va a ser el funcionamiento del sistema Ball And Plate. De esta manera, se podrá estudiar el hardware que formará parte del mismo, así como el papel que tendrá cada componente dentro del conjunto. En primer lugar, se dispondrá de una cámara que, recibirá información de lo que ocurre sobre la superficie y se la pasará a un microcontrolador en forma de coordenadas x,y . El microcontrolador, en función de la información recibida, definirá el comportamiento que deben seguir los actuadores. Por último, los actuadores cambiarán la inclinación de la plataforma siguiendo las instrucciones del microcontrolador. Como se va a trabajar con un sistema en lazo cerrado, se repetirá este proceso de manera periódica mientras que el sistema esté en funcionamiento. A continuación, en la Figura 2.1, se muestra un diagrama que ilustra los bloques que componen el sistema y la conexión entre ellos, que puede verse más en detalle en el Anexo A.

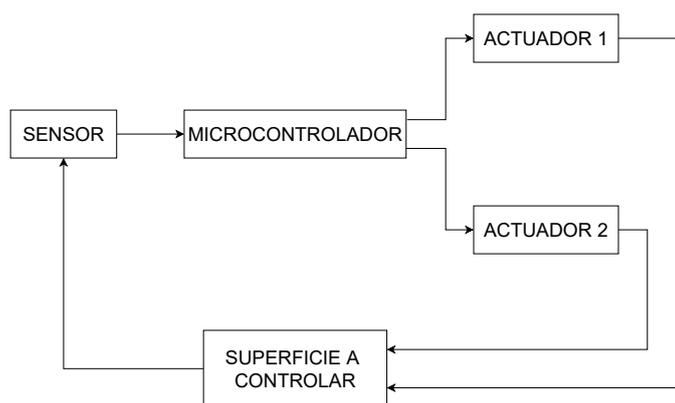


Figura 2.1: Diagrama que muestra los bloques que componen el sistema

2.1. Cámara

Como sensor se utilizará la cámara de bajo consumo *OpenMV Cam H7* [6] que se muestra en la Figura 2.2. Dicha cámara se basa en la Transformada de Hough [7] para la detección de objetos. La Transformada de Hough es capaz de detectar todas las figuras que se puedan expresar mediante ecuaciones matemáticas. Para lograr este objetivo, su misión principal será la detección de bordes de la figura de la cuál queremos obtener información. Estos bordes pueden tener ciertas imperfecciones que harían que la detección no fuera la correcta. La Transformada de Hough se encarga de solucionar estos problemas realizando agrupaciones de puntos que podrían encontrarse en el borde de la figura a detectar. En el sistema *Ball And Plate*, se quiere detectar un círculo ya que, aunque realmente se trate de una esfera, la cámara verá la imagen en 2D. Posteriormente se entrará en detalles sobre la implementación de un programa que aborde esta tarea.



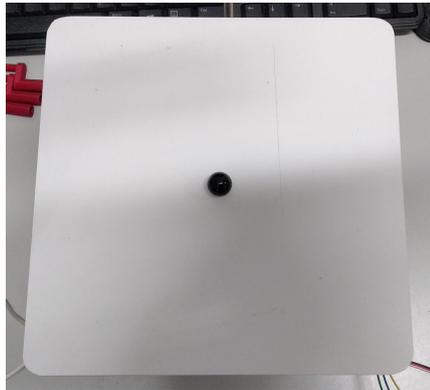
Figura 2.2: Sensor *OpenMV Cam H7 Plus*.

Entre las características de esta cámara se pueden destacar las siguientes:

- Un procesador STM32H743VI ARM Cortex M7, que funciona a 480 MHz con 1 MB de SRAM y 2 MB de flash.
- Todos los pines de E/S emiten 3,3 V y son tolerantes a 5 V.
- Una interfaz USB de alta velocidad (12 Mbs) para conectar la cámara al ordenador.
- Buses SPI, I2C, CAN y UART que permiten la transferencia de datos con otros sensores y microcontroladores a una velocidad de hasta 80 Mb/s, 1Mb/s, 1Mb/s y 7,5Mb/s, respectivamente.

Por otro lado, la forma más apropiada de utilizar este sensor es trabajando con una esfera y una superficie de colores opuestos para que la cámara pueda detectarlos

fácilmente. Por ello, la esfera del sistema será de color negro y el plato de color blanco. Además, con estos colores, la cámara podrá trabajar con escala de grises en lugar de con RGB, lo que hace que el sistema funcione más rápido. En la Figura 2.3 se puede ver una imagen de como se ve desde arriba el contraste entre los colores de la esfera y la plataforma en una foto hecha con el móvil y como ve el sistema la cámara.



(a) Imagen del sistema real.



(b) Imagen del sistema visto desde la cámara.

Figura 2.3: Imágenes del sistema real y el sistema visto desde la cámara.

Una particularidad de esta cámara es que se programa en scripts de Python de alto nivel (cortesía del sistema operativo MicroPython). Se hará un programa para que se detecte negro sobre blanco y, con ello, la posición de la esfera sobre el plato, que se mandará al microcontrolador.

2.2. Microcontrolador

Como se ha mencionado anteriormente, el microcontrolador será el encargado de gestionar la comunicación con el sensor y los actuadores, recibir los datos recogidos por el sensor y en base a esos datos, calcular la acción de control que se enviará a los actuadores. Utilizaremos el STM32F446RE [8], que se puede ver en la Figura 2.4.

Entre las características de este microcontrolador destacan las siguientes:

- Núcleo ARM Cortex M4 de 32 Bits capaz de operar a una frecuencia de hasta 180 MHz.
- Dispone de un conjunto completo de instrucciones DSP y una unidad de protección de memoria (MPU) que mejora la seguridad de la aplicación.
- 16 controladores de flujo de acceso directo a memoria (DMA) con FIFO y soporte de ráfagas
- 17 timers de los cuales 12 son de 16 bits, 2 son de 32 bits y todos ellos pueden funcionar a una frecuencia de hasta 180 MHz.

- 20 interfaces de comunicación entre las que se destacan 4 USARTs, 2 UARTs y 4 SPIs, que pueden transmitir y recibir datos a una velocidad máxima de 11,25 Mbit/s, 11,25 Mbit/s y 45 Mbit/s, respectivamente.

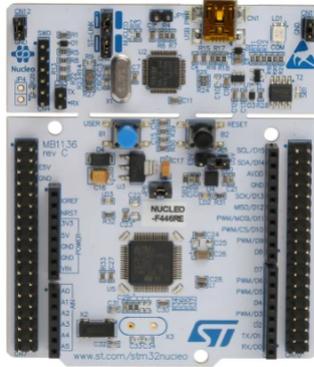


Figura 2.4: Microcontrolador *STM32F446RE*.

Se ha decidido utilizar este microcontrolador debido a que permite trabajar con un reloj de 180 MHz, lo cuál para el sistema en el que se está trabajando es más que suficiente para tener un funcionamiento correcto. Además se necesitará al menos 1 de las 20 interfaces de comunicación de las que dispone para establecer la comunicación con la cámara y, como mínimo de dos de los 17 timers que tiene para generar las señales PWM. Estos PWM se utilizarán para controlar los actuadores, como se explicará más adelante.

Por último, cabe destacar que el entorno de desarrollo de la familia STM32 permite gestionar el consumo y sumando esto a todo lo mencionado anteriormente y a su bajo coste, hace que sea un microcontrolador idóneo para este tipo de proyectos.

2.3. Actuadores

Como actuadores se utilizarán 2 servomotores Futaba S3001 [9] como el que se muestra en la Figura 2.5, que permiten rotaciones de entre 0° y 180° .

Como podemos observar en la Figura 2.5, el servomotor dispone de 3 cables. Los cables negro y rojo se utilizan para conectar la masa y la tensión de alimentación ($V_{min} = 4,8V$ y $V_{max} = 6V$), respectivamente. Es importante tener en cuenta que su velocidad mínima será de $0,28seg/60^\circ$ cuando se alimenta con 4,8V y su velocidad máxima será de $0,22seg/60^\circ$ cuando se alimenta con 6V. Por el cable blanco llegará la señal de control PWM procedente del microcontrolador.

En este momento conviene hacer un inciso para explicar qué es una señal PWM y cómo afecta al sistema. Las siglas PWM hacen referencia a la Modulación Por Ancho

de Pulso, que es el proceso mediante el cuál, la información que se transmite en una comunicación o la cantidad de energía que se envía a un dispositivo es controlada por el ciclo de trabajo de la señal. El ciclo de trabajo de una señal PWM es el intervalo de tiempo sobre el total en el que la señal se encuentra a nivel alto.



Figura 2.5: *Servomotor S3001.*

En la Figura 2.6 se puede ver una representación del funcionamiento del servomotor. En primer lugar, hay que tener en cuenta que para que el servomotor funcione, es necesario generar una señal PWM con un periodo de 20 ms. En la imagen se observa que si en estos 20 ms, se tiene un pulso a nivel alto con una duración de 1ms, lo que equivale a tener un ciclo de trabajo del 5%, el servomotor se sitúa en la posición 0°; si el pulso a nivel alto tiene una duración de 1,5 ms, lo que equivale a tener un ciclo de trabajo del 7,5%, el servomotor se situará en la posición 90°; y, si el pulso a nivel alto dura 2ms, lo que equivale a un ciclo de trabajo del 10%, el servomotor se situará en la posición 180°.

Como se ha mencionado anteriormente, el servomotor solo puede recorrer el rango de 0° a 180°, es por esto que el mayor ciclo de trabajo que podremos tener en la señal PWM es el 10%.

Se ha analizado qué intervalo de tiempo a nivel alto se necesita para conseguir que el servomotor se encuentre en las posiciones 0°, 90° o 180°, pero si se quiere que el servomotor rote a cualquier otra posición dentro del rango, el tiempo en el que la señal debería estar a nivel alto se puede calcular como:

$$t_{alto}(ms) = 1 + \frac{Angulo(^{\circ})}{180} = 1 + \frac{Angulo(rad)}{2\pi} \quad (2.1)$$

Y el ciclo de trabajo se obtendría de la siguiente manera:

$$Duty(\%) = \frac{t_{alto}(ms)}{T_{PWM}(ms)} = \frac{t_{alto}(ms)}{20ms} \quad (2.2)$$

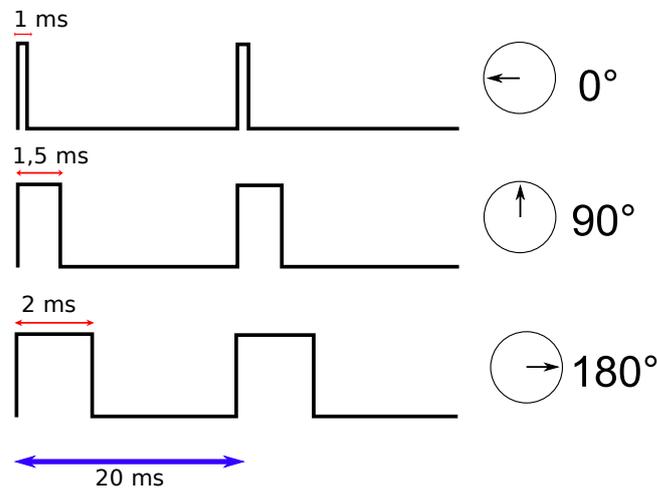


Figura 2.6: Relación entre el ciclo de trabajo de la señal PWM y la rotación del servomotor.

2.4. Comunicación entre la cámara y el microcontrolador

Para comunicar la cámara y el microcontrolador se consideraron 2 opciones: utilizar el bus SPI o utilizar el puerto serie UART. A continuación se detalla en qué consiste cada protocolo de comunicación y cuál será la mejor opción para el proyecto.

2.4.1. Bus SPI

En la Figura 2.7 se muestra el esquema típico de una comunicación SPI [10] compuesta por un maestro y un único esclavo. Podemos observar 4 señales: SCLK, MOSI, MISO y \overline{SS} . SCLK corresponde a la señal de reloj (por lo general del orden de MHz) que se utiliza para sincronizar la transferencia de datos y es generada por el maestro, que es el que empieza siempre la comunicación; la señal MOSI corresponde a los datos que salen del maestro y entran en el esclavo; la señal MISO es justo lo contrario a la señal MOSI, en este caso son los datos que proceden del esclavo y recibe el maestro; y, por último, la señal \overline{SS} se utiliza en primer lugar para definir el comienzo de la transferencia de datos poniendo esta señal a nivel bajo y, en segundo lugar, cuando hay más de un esclavo se utiliza para seleccionar el esclavo con el que queremos establecer la comunicación.

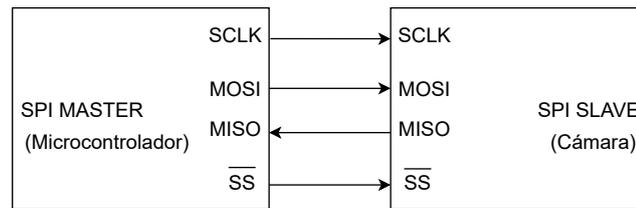


Figura 2.7: Comunicación SPI

En la Figura 2.8 se muestra un ejemplo de como se produce una transferencia de datos utilizando el protocolo SPI. En ella se observa como se activa la señal de reloj SCK y, a partir de ese momento, es la señal SS la que marca el inicio de la transferencia cuando se pone a nivel bajo. Una vez esta señal se ha puesto a nivel bajo comienza la transmisión de datos y cada bit presente en las señales MOSI y MISO se manda a su correspondiente destinatario con cada flanco de subida del reloj.

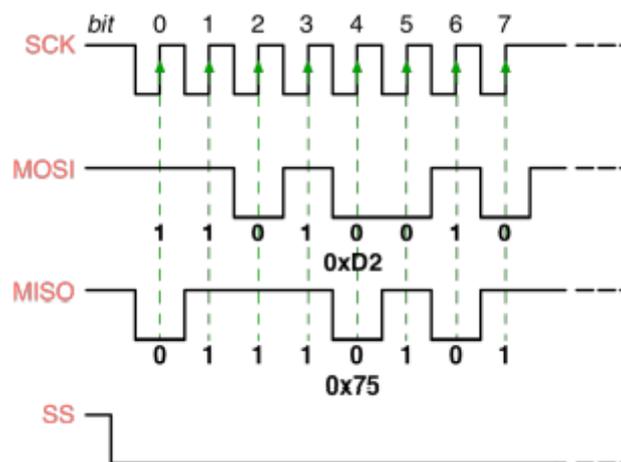


Figura 2.8: Ejemplo de transferencia de datos full-duplex utilizando el protocolo SPI [1].

En la Figura 2.8 se ha visto un ejemplo realizando una transmisión full-duplex, esto quiere decir que tanto los datos que se envían desde el maestro al esclavo como los datos que se mandan desde el esclavo al maestro, se transmiten al mismo tiempo. Otra posibilidad sería utilizar half-duplex, como se muestra en la Figura 2.9 donde, en este caso, en primer lugar se transmiten los datos en un sentido y, cuando ha finalizado la transmisión, se transmiten los datos en el otro sentido. En este caso, en primer lugar se mandan datos desde el maestro al esclavo y, posteriormente, se mandan los datos del esclavo al maestro.

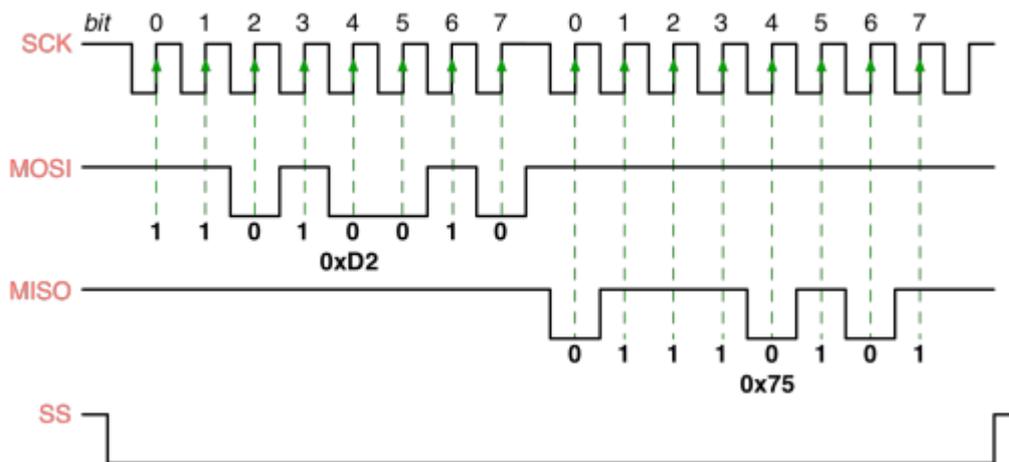


Figura 2.9: Ejemplo de transferencia de datos half-duplex utilizando el protocolo SPI [1].

2.4.1.1. Polaridad y Fase del reloj utilizando el protocolo SPI

Es importante tener en cuenta que cuando utilizamos el protocolo SPI es necesario configurar la polaridad y la fase del CLK ya que son las que van a determinar como se realiza la transmisión y recepción de datos en función de si hay un flanco de subida o un flanco de bajada del CLK. En la siguiente tabla se muestran los posibles modos de CLK que hay en función de los valores de polaridad y fase que elijamos.

Modo	Polaridad	Fase
0	0	0
1	0	1
2	1	0
3	1	1

Tabla 2.1: Tabla resumen sobre los posibles modos del CLK en función de la polaridad y la fase.

A continuación se explica el significado que tienen los modos del CLK, así como los valores de la polaridad y la fase:

- Si la polaridad del CLK es 0, tenemos que el 0 es el valor base del CLK, que se activará a nivel alto. Considerando la polaridad a 0:
 - Si la fase es 0, los datos serán capturados en el flanco de subida y transmitidos en el flanco de bajada.
 - Si la fase es 1, los datos serán capturados en el flanco de bajada y transmitidos en el flanco de subida.

- Si la polaridad del CLK es 1, tenemos que el 1 es el valor base del CLK, que se activará a nivel bajo. Considerando la polaridad a 1:
 - Si la fase es 0, los datos serán capturados en el flanco de bajada y transmitidos en el flanco de subida.
 - Si la fase es 1, los datos serán capturados en el flanco de subida y transmitidos en el flanco de bajada.

2.4.2. Puerto Serie UART

En la Figura 2.10 se observa un diagrama de bloques que corresponde a utilizar el puerto serie para comunicar el microcontrolador con la cámara. En ella se puede ver que, para establecer la comunicación bastaría con 2 cables, uno que conectara el pin configurado como transmisor del microcontrolador con el pin configurado como receptor en la cámara y otro cable que conectara el pin configurado como transmisor de la cámara con el pin configurado como receptor en el microcontrolador. Además, en los correspondientes IDEs de la cámara y el microcontrolador, se establecerá una velocidad de transmisión, que debe ser la misma en ambos dispositivos.

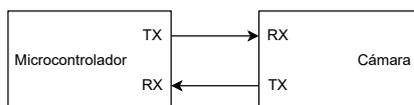


Figura 2.10: Comunicación UART.

Para el desarrollo del TFG, se ha optado por establecer la comunicación entre el microcontrolador y la cámara mediante el puerto serie. Esta decisión ha sido tomada debido a que el protocolo SPI en el modelo de cámara utilizado se encontraba en una versión beta al inicio del TFG. Actualmente, los ingenieros de OpenMV han programado una serie de librerías que habría que importar tanto en el IDE de la cámara como en el IDE del microcontrolador, añadiendo una dificultad innecesaria al TFG.

Capítulo 3

Modelado

3.1. Introducción

Como se ha mencionado en la Introducción de este documento, se va a trabajar con un sistema en lazo cerrado cuyo diagrama de bloques se puede observar nuevamente en la Figura 3.1 donde se encuentran presentes las siguientes señales:

- $r(t)$ es la señal de referencia deseada (la posición a la que debe llegar la esfera).
- $e(t)$ es la señal de error, diferencia entre la referencia $r(t)$ y la señal de salida $y(t)$.
- $u(t)$ es la señal de control (ángulo que se va a aplicar a los servomotores para que balanceen la estructura y se esté más cerca de conseguir que la esfera se encuentre en la posición deseada).
- $y(t)$ es la señal de salida (posición en la que se encuentra la esfera sobre la superficie).

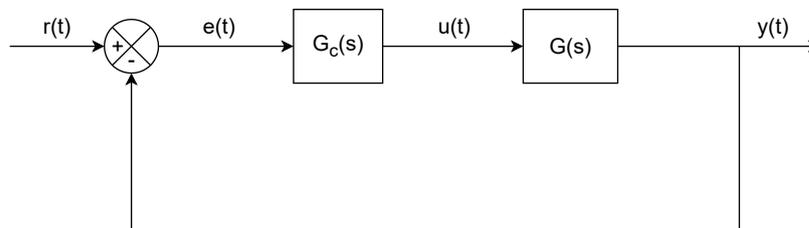


Figura 3.1: Sistema en lazo cerrado.

En esta sección se va a obtener la función de transferencia del sistema sin control, que se representa en la Figura 3.1 como $G(s)$.

3.2. Modelado de un sistema Ball And Plate

Para modelar un sistema Ball And Plate y obtener su función de transferencia $G(s)$, se deben analizar en primer lugar las fuerzas a las que está sometido el sistema [11]. En la Figura 3.2 se puede ver cómo va a ser la estructura con la que se va a trabajar. Esto ayudará a entender qué fuerzas hay que considerar en el estudio.

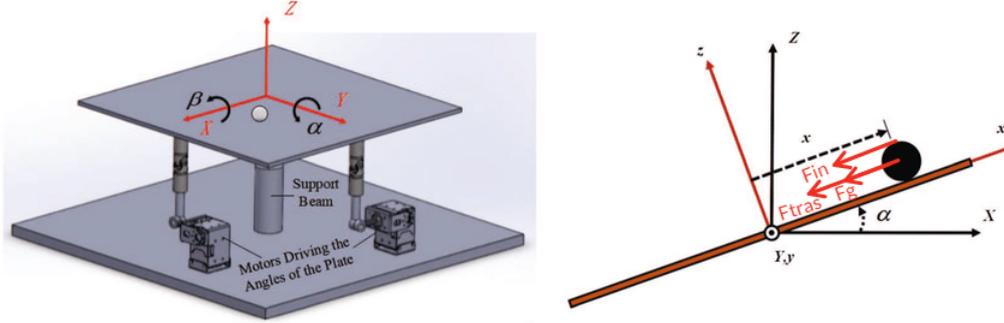


Figura 3.2: Representación de la estructura Ball And Plate [2].

En la imagen de la izquierda se muestra el sistema completo, donde se puede observar que la superficie se inclina sobre el eje X y el eje Y. En la imagen de la derecha se puede ver lo que sucede cuando la superficie se inclina sobre uno de los ejes mencionados, en este caso sobre el eje Y. En esta situación, aparece una fuerza debida a la gravedad, F_g , una fuerza debida a la traslación de la esfera, F_{tras} y una fuerza inercial, F_{in} . Aparecería además una cuarta fuerza que sería de rozamiento y que no se ha representado en el diagrama puesto que se supondrá que es nula para simplificar el modelo.

F_g corresponde, a la fuerza de atracción que hay entre 2 cuerpos con masa, en este caso entre la esfera y el centro de la Tierra. Matemáticamente se puede describir como:

$$F_g = m \cdot g \cdot \text{sen}(\alpha) \quad (3.1)$$

siendo m la masa de la esfera, g la aceleración de la gravedad y α el ángulo de inclinación entre la superficie y el plano horizontal.

F_{tras} hace referencia a la fuerza que hace que la esfera se mueva sobre la superficie. Se trata de un movimiento curvilíneo que presenta una componente tangencial y una componente normal. Se expresaría de la siguiente manera:

$$F_{tras} = F_{tras}^t + F_{tras}^n = m \cdot \ddot{x} - m \cdot x \cdot \dot{\alpha}^2 \quad (3.2)$$

donde m es la masa de la esfera, $\ddot{x} = \frac{d^2x}{dt^2}$ es su aceleración lineal, x es la distancia a la que se encuentra la esfera del centro de giro y $\dot{\alpha} = \frac{d\alpha}{dt}$ es la velocidad angular de la

superficie.

Por último, F_{in} es la fuerza que hace que todo objeto en reposo siga en reposo y que todo objeto en movimiento siga en movimiento si no se aplica una fuerza sobre el mismo. En el caso de la esfera, esta fuerza se podría describir de la siguiente manera:

$$F_{in} = \frac{J_b}{R^2} \cdot \ddot{x} \quad (3.3)$$

donde J_b es el momento de inercia de la esfera, R es el radio de la esfera y \ddot{x} su aceleración lineal.

Si se aplica la Primera Ley de Newton, se afirma que en un sistema en equilibrio, la suma de las fuerzas es igual a cero, es decir:

$$\sum F = F_g + F_{tras} + F_{in} = 0 \quad (3.4)$$

Sustituyendo cada fuerza por las ecuaciones obtenidas anteriormente se tiene que:

$$m \cdot g \cdot \text{sen}(\alpha) + m \cdot \ddot{x} - m \cdot x \cdot \dot{\alpha}^2 + \frac{J_b}{R^2} \cdot \ddot{x} = 0 \quad (3.5)$$

Agrupando términos:

$$\left(m + \frac{J_b}{R^2}\right) \ddot{x} - m \cdot x \cdot \dot{\alpha}^2 + m \cdot g \cdot \text{sen}(\alpha) = 0 \quad (3.6)$$

La Ecuación 3.6 define el comportamiento de la esfera ante cualquier posible variación en la estructura. Conviene linealizar esta ecuación para poder realizar el modelado de una manera más sencilla. Para ello, se supondrá que la superficie se encuentra en una situación cercana al equilibrio, con poca inclinación. En ese caso, el valor de α , el ángulo que forma la superficie con el plano horizontal, es muy pequeño. De esta manera, si $\alpha = 0$, se puede aproximar el $\text{sen}(\alpha)$ a α y $\dot{\alpha} = 0$. Teniendo en cuenta todo esto la ecuación quedaría de la siguiente manera:

$$\left(m + \frac{J_b}{R^2}\right) \ddot{x} + m \cdot g \cdot \alpha = 0 \quad (3.7)$$

Por último, para no tener problemas de signos, se hará un cambio en el sistema de referencia. Se considerará que ángulos positivos de la plataforma impliquen aceleraciones de la esfera hacia el exterior. Considerando esta modificación, la ecuación finalmente quedaría:

$$\left(m + \frac{J_b}{R^2}\right) \ddot{x} = m \cdot g \cdot \alpha \quad (3.8)$$

3.3. Transformada de Laplace

Para obtener la función de transferencia de un sistema de una manera sencilla se suele hacer uso del dominio de Laplace. En este caso, si se realiza la transformada de Laplace a la ecuación que se ha obtenido en la sección anterior se llega a:

$$\left(m + \frac{J_b}{R^2}\right) \cdot X(s) \cdot s^2 = m \cdot g \cdot A(s) \quad (3.9)$$

A continuación, se busca definir la función de transferencia del sistema. Para ello hay que tener en cuenta que:

- En el caso general, la función de transferencia $H(s) = \frac{Y(s)}{X(s)}$, siendo $X(s)$ la entrada del sistema e $Y(s)$ la salida del sistema.
- Para identificar cuál es la entrada y cuál es la salida de nuestro sistema debemos fijarnos en que en el denominador de la función de transferencia nos quede siempre el polinomio de grado más alto.

Teniendo en cuenta estas consideraciones, la función de transferencia del sistema quedaría como:

$$G(s) = \frac{X(s)}{A(s)} = \frac{m \cdot g}{\left(m + \frac{J_b}{R^2}\right) \cdot s^2} \quad (3.10)$$

Teniendo en cuenta que el momento de inercia de la esfera responde a la ecuación $J_b = \frac{2}{5} \cdot m \cdot R^2$ y sustituyendo tanto el momento de inercia de la esfera como el valor de la aceleración de la gravedad ($g \approx 10$) en la Ecuación 3.10 se llega a que, finalmente, la función de transferencia del sistema Ball And Plate es:

$$G(s) = \frac{7}{s^2} \quad (3.11)$$

3.4. Modelado en Simulink

Una vez se ha obtenido la función de transferencia, se ha realizado una simulación en Simulink para ver cómo se comporta el sistema. Para ello, se han creado los bloques que componen actualmente el sistema, como se muestra en la Figura 3.3. Algo que podría llamar la atención es que la rama superior e inferior son idénticas. Esto se debe a que se van a analizar por separado el comportamiento sobre el eje X y el comportamiento sobre el eje y.

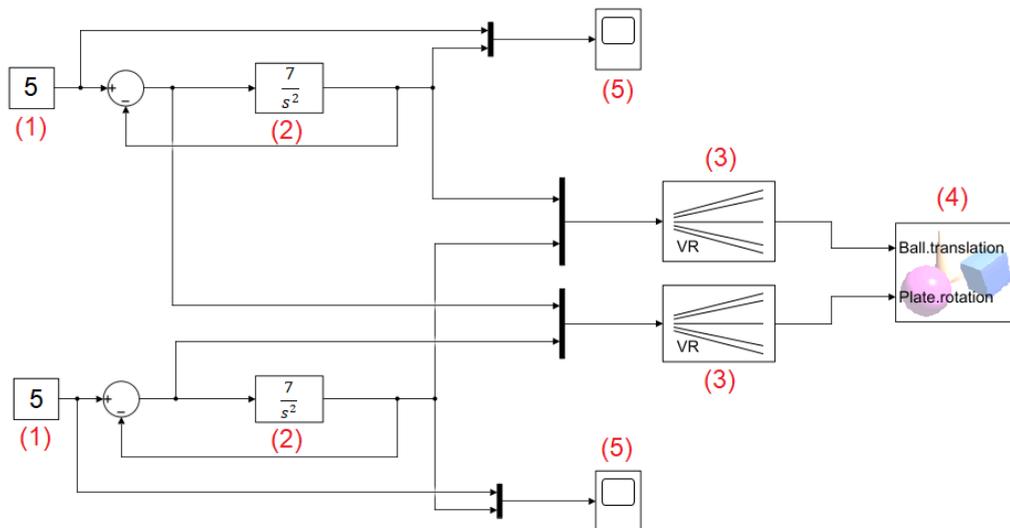


Figura 3.3: Diagrama de bloques del sistema simulado en Simulink.

Analizando más en detalle la Figura 3.3 se distinguen 5 bloques. El bloque (1) corresponde a la señal de referencia del sistema. En este caso, se ha hecho una simulación donde el centro de la superficie se encuentra en las coordenadas (5,5), por ello, este es el valor constante que se introduce a la entrada de ambas ramas. El bloque (2) es la función de transferencia del sistema, que se ha calculado en la sección anterior. El bloque (3) sirve para que tanto la salida del lazo realimentado como el error, se conviertan a vector. El bloque (4) sirve para visualizar la simulación que se ha realizado, como se muestra en la Figura 3.4. Por último, el bloque (5) se utiliza para obtener las gráficas de referencia y de posición de la esfera y, de esta manera, poder ver cómo se ajusta la posición de la esfera a la referencia.

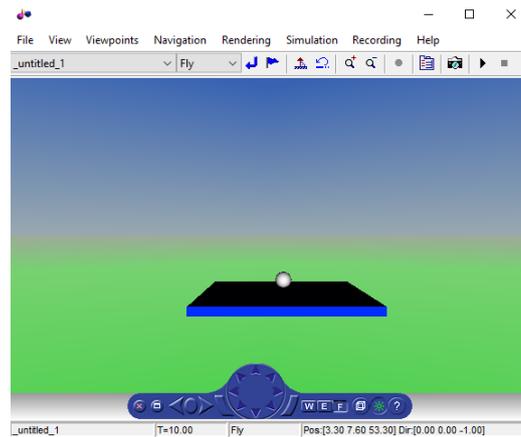
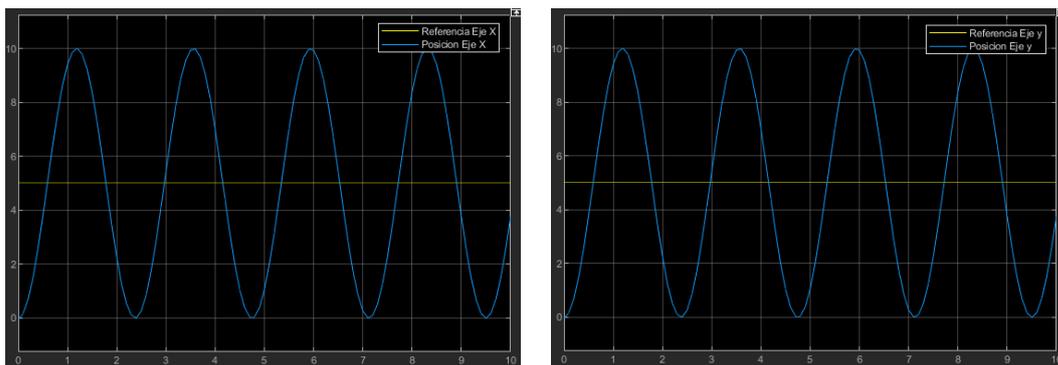


Figura 3.4: Simulación.

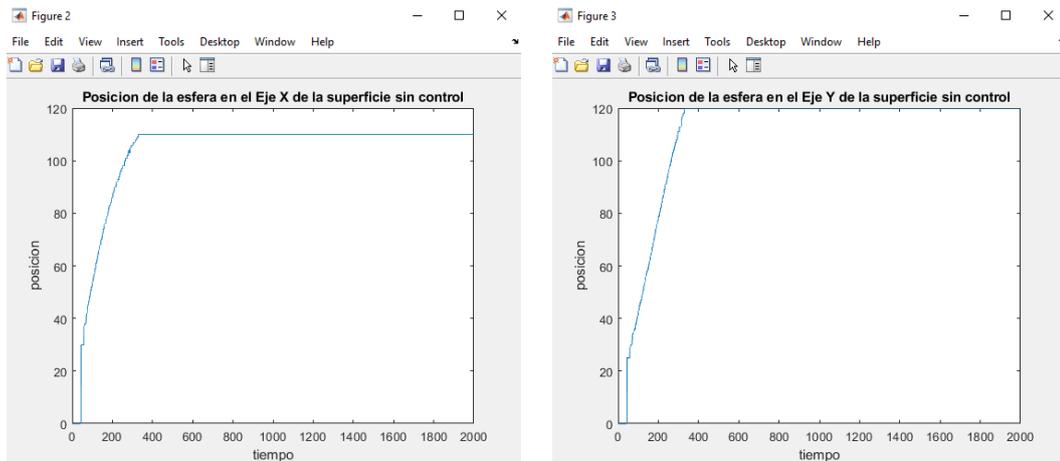
Si estuviera hecho el control, a la salida de cada rama se debería obtener una señal que se aproxime a la señal de referencia. Sin embargo, lo que se obtiene en un experimento realizado durante 10 segundos es una señal periódica que está lejos de converger al valor de referencia, como se puede observar en la Figura 3.5.



(a) Posición de la esfera en el Eje X de la simulación (b) Posición de la esfera en el Eje Y de la simulación

Figura 3.5: Representación de la posición de la esfera sobre los ejes X e Y en la simulación

En el sistema real lo que sucede es que al soltar la esfera sobre la superficie, seguirá en movimiento hasta que se salga de la misma. En la Figura 3.6 se observa el comportamiento de la esfera sobre los ejes X e Y de la superficie en un caso de los infinitos casos que podrían presentarse. En este experimento, se suelta la esfera desde la esquina con coordenadas $(0,0)$ y se observa que, a partir de cierto instante de tiempo el valor de la posición es constante. Esto se debe a que se ha programado que cuando la esfera se salga de la superficie, se imprima como valor de posición el último valor que se ha medido. Así que, en realidad, este instante de tiempo corresponde al momento en el que la esfera se ha salido de la superficie.



(a) Posición de la esfera en el Eje X.

(b) Posición de la esfera en el Eje Y.

Figura 3.6: Representación de la posición de la esfera sobre los ejes X e Y en el sistema real.

En la Sección 4 el objetivo será diseñar un controlador capaz de conseguir que la esfera se mantenga en el centro de la superficie y en el caso de sufrir algún desplazamiento, que sea capaz de corregir su posición y termine de nuevo en el centro.

Capítulo 4

Diseño de controladores

En esta sección se trabajará en el bloque de control del sistema Ball And Plate, representado en la Figura 1.2 como $G_c(s)$. El objetivo consiste en analizar varios controladores [12] y, que de esta manera, se puedan sintonizar los parámetros necesarios para diseñar un controlador para el sistema. Dicho controlador, a su entrada recibirá la señal error, la diferencia entre la señal de salida (posición de la esfera) y la referencia del sistema (posición a la que se quiere llegar) y trabajará sobre ella para que a medida que avance el tiempo del experimento la salida del sistema sea cada vez más próxima a su entrada. En concreto, se trabajará con 2 controladores: un controlador Proporcional (P) y un controlador Proporcional Derivativo (PD).

4.1. Controlador proporcional (P)

Como se ha mencionado en el Capítulo 3, se va a trabajar independientemente con el eje X y con el eje Y, es decir, se van a diseñar dos controladores, uno por cada eje. Teniendo esto en cuenta, el esquema del sistema utilizando un controlador proporcional para ambos ejes sería:

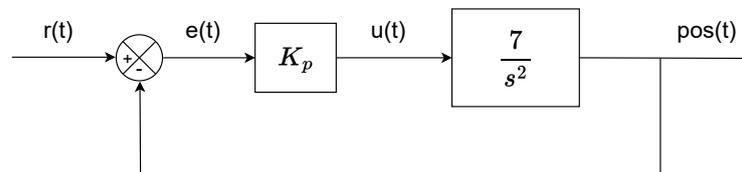


Figura 4.1: Sistema con controlador P.

Siguiendo el desarrollo explicado en la Sección 1.2, la función de transferencia en lazo cerrado tanto para el sistema de control para el eje X como el sistema de control para el eje Y quedaría:

$$H_{lc}(s) = \frac{7K_p}{s^2 + 7K_p} \quad (4.1)$$

4.2. Análisis del problema de seguimiento al escalón, la rampa y la parábola utilizando un controlador proporcional (P)

Un requisito indispensable en el diseño de controladores es analizar el error en régimen permanente (e_{ss}) del sistema cuando se introducen a la entrada como señales de referencia el escalón, la rampa y la parábola. Para ello, se aplica el teorema del valor final, que se describe a continuación:

$$e_{ss} = \lim_{s \rightarrow 0} s \cdot H_e(s) \cdot R(s) \quad (4.2)$$

donde $H_e(s)$ es la función de transferencia del error y $R(s)$ corresponde a la señal de referencia y va tomando distintos valores en función de la señal de referencia que se introduce a la entrada:

- Escalón: $R(s) = \frac{1}{s}$
- Rampa: $R(s) = \frac{1}{s^2}$
- Parábola: $R(s) = \frac{1}{s^3}$

La función de transferencia del error, $H_e(s)$ para el sistema utilizando un controlador proporcional se obtendría como:

$$H_{e,P}(s) = 1 - H_{lc}(s) = \frac{s^2}{s^2 + 7K_p} \quad (4.3)$$

Se observa que, cuanto mayor sea K_p , menor será el error.

El error en régimen permanente cuando se introduce una señal escalón sería:

$$e_{ss0,P}(\infty) = \lim_{s \rightarrow 0} s \cdot \frac{s^2}{s^2 + 7K_p} \cdot \frac{1}{s} = 0 \quad (4.4)$$

Cuando se introduce una rampa:

$$e_{ss1,P}(\infty) = \lim_{s \rightarrow 0} s \cdot \frac{s^2}{s^2 + 7K_p} \cdot \frac{1}{s^2} = 0 \quad (4.5)$$

Y, cuando se introduce una parábola:

$$e_{ss2,P}(\infty) = \lim_{s \rightarrow 0} s \cdot \frac{s^2}{s^2 + 7K_p} \cdot \frac{1}{s^3} = \frac{1}{7K_p} \quad (4.6)$$

4.3. Controlador proporcional derivativo (PD)

El controlador proporcional derivativo (PD) se diferencia del controlador proporcional (P) en que se introduce al control una constante derivativa $K_D = K_p\tau_D$ como se muestra en la Figura 4.2.

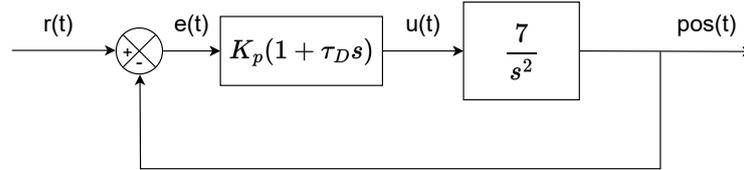


Figura 4.2: Sistema con controlador PD.

La función de transferencia en lazo cerrado para ambos ejes en este caso sería:

$$H_{lc}(s) = \frac{7K_p(1 + \tau_D s)}{s^2 + 7K_p(1 + \tau_D s)} \quad (4.7)$$

4.4. Análisis del problema de seguimiento al escalón, la rampa y la parábola utilizando un controlador proporcional derivativo (PD)

Utilizando este controlador, la función de transferencia del error sería:

$$H_{e,PD}(s) = 1 - H_{lc}(s) = \frac{s^2}{s^2 + 7K_p(1 + \tau_D s)} \quad (4.8)$$

Cuando se introduce un escalón, el error en régimen permanente sería:

$$e_{ss0,PD}(\infty) = \lim_{s \rightarrow 0} s \cdot \frac{s^2}{s^2 + 7K_p(1 + \tau_D s)} \cdot \frac{1}{s} = 0 \quad (4.9)$$

Cuando se introduce una rampa:

$$e_{ss1,P}(\infty) = \lim_{s \rightarrow 0} s \cdot \frac{s^2}{s^2 + 7K_p(1 + \tau_D s)} \cdot \frac{1}{s^2} = 0 \quad (4.10)$$

Y, cuando se introduce una parábola:

$$e_{ss2,P}(\infty) = \lim_{s \rightarrow 0} s \cdot \frac{s^2}{s^2 + 7K_p(1 + \tau_D s)} \cdot \frac{1}{s^3} = \frac{1}{7K_p} \quad (4.11)$$

4.5. Sintonización de parámetros

Teniendo en cuenta que, con el análisis del problema de seguimiento obtenemos los mismos resultados para ambos controladores y que, utilizando un controlador proporcional derivativo (PD) tenemos un grado de libertad extra, se van a sintonizar los parámetros para dicho controlador.

Antes de empezar a sintonizar los parámetros conviene tener en cuenta cómo va a ser la respuesta de nuestro sistema y los parámetros presentes en ella. En la Figura 4.3 encontramos los siguientes parámetros:

- M_p es la sobreelongación máxima. Esto es la diferencia máxima que hay entre la respuesta del sistema y la señal de entrada.
- t_r corresponde al tiempo de subida. Se refiere al tiempo que se tarda en alcanzar la señal de referencia.
- t_p es el tiempo de pico. El tiempo que se tarda en llegar al valor máximo de la respuesta.
- t_s es el tiempo de establecimiento. Esto es el instante de tiempo en el que se considera que la señal empieza a ser estable.
- ν es la tolerancia, que define la diferencia máxima permitida entre la respuesta del sistema y la referencia para que se considere que el sistema es estable.

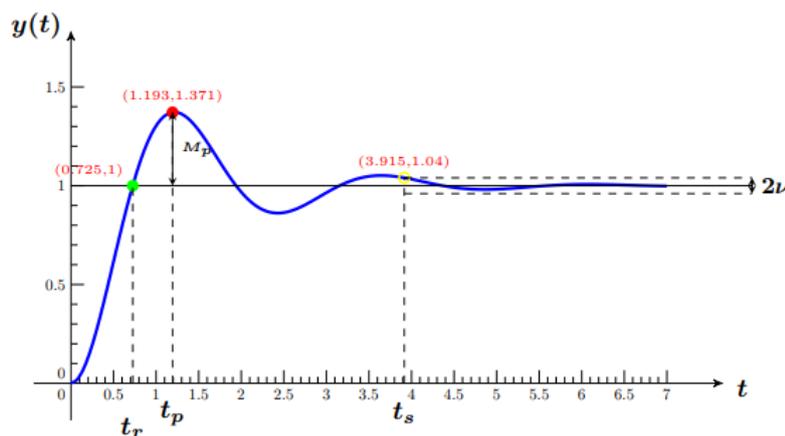


Figura 4.3: Ejemplo de respuesta al escalón unidad de un sistema de segundo orden [3].

Para la sintonización de los parámetros del controlador, se va a fijar una sobreelongación máxima y un tiempo de establecimiento, ya que únicamente fijando estos parámetros se puede llegar al comportamiento esperado. Se consideran valores

razonables una $M_p = 0,2$ y un $t_s = 3s$.

Otro factor a tener en cuenta es que si se analiza la función de transferencia del controlador se obtiene que:

$$G_c = K_p + K_D s = K_p \left(1 + \frac{K_D}{K_p} s\right) = K_D \left(s + \frac{K_p}{K_D}\right) \quad (4.12)$$

De esta manera, se presenta un cero real ajustable de valor $\frac{K_p}{K_D}$. Para obtener la posición real del cero ajustable se hará uso del Lugar de Raíces. El Lugar de Raíces [13] representa en el plano complejo s el lugar geométrico de los polos de lazo cerrado del sistema de control realimentado, en función de los diferentes valores de la ganancia $K \in [0, \infty)$ de un controlador situado en el lazo directo, como se muestra en la Figura 4.2.

Aplicando el criterio de argumentos [14], se puede obtener la posición a , en la que se debe colocar el cero para que el polo dominante se encuentre en el Lugar de Raíces:

$$\sum \arg\{ceros\} - \sum \arg\{polos\} = \tan^{-1}\left(\frac{\omega_d}{\sigma - a}\right) - 2\tan^{-1}\left(\frac{\omega_d}{\sigma}\right) = (2q + 1)\pi \quad (4.13)$$

donde σ es la posición en el eje real del polo complejo, ω_d la posición en el eje imaginario del polo complejo y con $q = 0, \pm 1, \pm 2, \dots$

Si se despeja a en la ecuación anterior se obtiene que:

$$a = \tan\left(\pi + 2 \cdot \left(\tan^{-1}\left(\frac{\omega_d}{\sigma}\right)\right)\right) \cdot \omega_d + \sigma \quad (4.14)$$

Por otro lado σ y ω_d podrían obtenerse como:

$$\sigma = \frac{\pi}{t_s} = \frac{\pi}{3} = 1,047 \quad (4.15)$$

$$\omega_d = \frac{\pi^2}{\ln\left(\frac{1}{M_p}\right) \cdot t_s} = \frac{\pi^2}{\ln\left(\frac{1}{0,2}\right) \cdot 3} = 2,044 \quad (4.16)$$

Introduciendo los valores de σ y ω_d en la Ecuación 4.14 se llega a que $a = -1,792$.

Los parámetros de diseño K_p y K_D se pueden obtener aplicando las siguientes fórmulas:

$$K_p = \frac{-K \cdot a}{7} \quad (4.17)$$

$$K_D = \frac{K}{7} \quad (4.18)$$

donde K es la ganancia del sistema en lazo abierto y a es la posición del cero ajustable.

La ganancia del sistema en lazo abierto se puede obtener como:

$$K = \frac{\sqrt{\omega_d^2 + \sigma^2} \cdot \sqrt{\omega_d^2 + \sigma^2}}{\sqrt{\omega_d^2 + (\sigma - a)^2}} = 1,507 \quad (4.19)$$

Finalmente, se obtienen los siguientes parámetros de diseño:

$$\begin{aligned} K_p &= 0,386 \\ K_D &= 0,2154 \end{aligned}$$

En el Capítulo 6 se estudiarán los resultados obtenidos para estos parámetros tanto en el sistema ideal como en el sistema real.

Capítulo 5

Implementación

Para la implementación del sistema se han utilizado dos entornos de desarrollo: *OpenMV IDE* para la cámara y *STM32CubeIDE* para el microcontrolador. Como se ha comentado anteriormente en este documento, la cámara se ha programado en un script de Python de alto nivel mientras que el microcontrolador se ha programado en C. En este capítulo se analizará el código desarrollado en ambos IDEs.

5.1. Código desarrollado en *OpenMV IDE*

En esta parte de la implementación se tenían que realizar dos tareas: detectar la esfera sobre la superficie y mandar la posición de la esfera por la UART al microcontrolador.

Para detectar la posición de la esfera sobre el plato se ha trabajado con escala de grises ya que de esta manera el sistema funciona más rápido. Para ello, se ha utilizado la función `sensor.set_pixformat()` donde si se introduce como parámetro `sensor.GRAYSCALE` la cámara funciona con la escala deseada. Para la detección de la esfera se ha hecho uso de la función `img.find_circles`, que permite detectar círculos sobre una superficie ya que la cámara ve la imagen en 2D y la esfera la ve como un círculo. Esta función permite obtener las coordenadas de la posición en la que se encuentra la esfera y definir un umbral con el tamaño máximo de la esfera que detecta la cámara. Para esto, se han ido haciendo ajustes de este umbral de tal manera que las esquinas de la superficie no fueran detectadas como círculos. Una vez realizado este ajuste y habiendo comprobado que la esfera se detectaba correctamente se ha pasado a enviar la posición de la esfera al microcontrolador.

Para mandar la posición de la esfera, en primer lugar se ha creado un array donde se van guardando las coordenadas x e y de la esfera. Posteriormente, se ha definido que la tasa de envío de la UART sea de 115200 baudios que es un valor estándar muy utilizado en este tipo de comunicaciones. Finalmente, haciendo uso de la función `uart.write()` se han mandado los datos de la posición por el puerto serie.

5.2. Código desarrollado en *STM32CubeIDE*

El objetivo de esta parte de la implementación consiste en recibir los datos enviados por la cámara por el puerto serie, realizar un control sobre estos datos y generar una señal PWM que permita mover los servomotores en función de los datos recibidos.

Para la recepción de los datos procedentes de la cámara se ha creado una función *recogeDatos*. En esta función se ha programado que los datos que llegan por la UART1, que previamente ha sido configurada con la misma tasa de envío y recepción de datos que la cámara, se guarden en un array data.

A continuación, se han creado las funciones correspondientes a los controladores, que se probarán por separado. Las funciones *void setPositionPX(int referenciaPX)* y *void setPositionPY(int referenciaPY)* corresponden al controlador P para cada eje. En las funciones *void setPositionPDX(int referenciaPDX)* y *void setPositionPDY(int referenciaPDY)* se ha programado el controlador PD para cada eje. En estas funciones se han definido ángulos máximos y mínimos de control con el objetivo de no dañar la estructura con la que estamos trabajando. Además, en estas funciones se ha hecho uso de las funciones *void setDegreeCCR1_X(float degX)* y *void setDegreeCCR2_Y(float degY)*. Estas funciones se encargan de que, una vez calculado el ángulo de control que se va a aplicar a cada eje, se convierta este ángulo en su correspondiente señal PWM aplicando la Ecuación 2.1 y 2.2. Para esto, se ha utilizado un timer que, a través de sus canales 1 y 2, va introduciendo la señal PWM en los servomotores.

Una vez programadas todas las funciones, es importante tener en cuenta que tanto la recepción de datos, como el control y la generación de señales PWM deben realizarse de manera periódica. Para conseguir esto, se ha configurado un timer de tal manera que cada 5ms se ejecuten estas tareas.

Por último se ha configurado un segundo puerto serie, UART2, para mandar datos desde el microcontrolador al ordenador, con el objetivo de poder graficar los datos del experimento y entender cómo se está comportando nuestro sistema. Este envío de datos se ha hecho utilizando un tercer timer que, cada 5ms actualiza el valor de un contador y, que cuando la cuenta llega a 2000, manda estos datos al ordenador. De esta manera, se tiene que la duración del experimento es de 10 segundos.

Capítulo 6

Resultados del experimento

En este Capítulo, se pasarán a comentar los resultados obtenidos en el experimento. Para ello, se hará una comparativa entre los resultados obtenidos para el sistema real y los resultados obtenidos en la simulación para el sistema ideal utilizando los controladores diseñados. Es importante tener en cuenta que, para ambos casos, la duración del experimento ha sido de 10 segundos.

6.1. Controlador P

En esta sección se muestra el análisis de los datos obtenidos utilizando un controlador proporcional. Se han ido probando distintos valores para K_p y se ha estudiado el comportamiento del sistema para dichos valores.

6.1.1. Comparativa entre el comportamiento del sistema real y el sistema ideal

Para estudiar el comportamiento del sistema ideal, en el esquema de la Figura 3.3 se ha añadido un bloque para el controlador P como se muestra en la Figura 6.1.

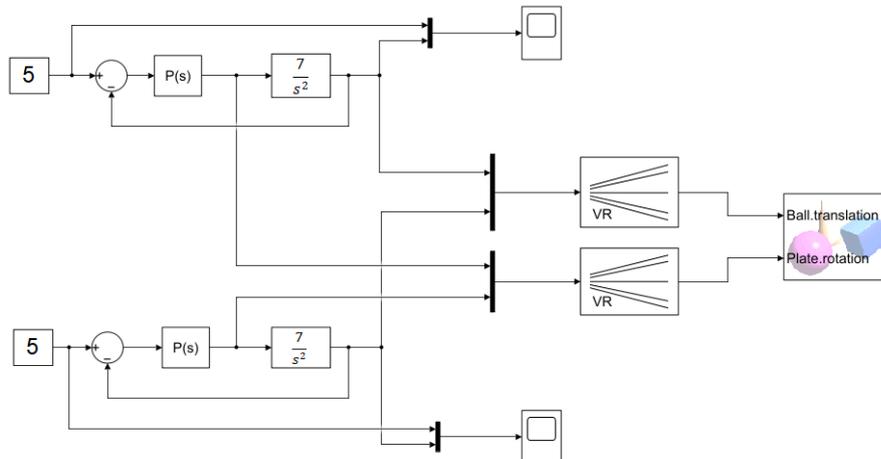
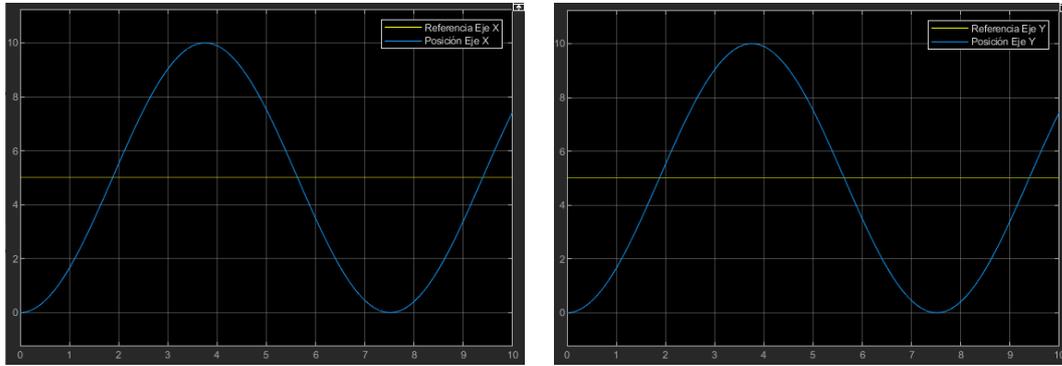


Figura 6.1: Diagrama de bloques del sistema simulado en Simulink con controlador P

En primer lugar, se empezó a trabajar con un valor pequeño de la constante de proporcionalidad, $K_p = 0,1$. Se eligió este valor ya que la estrategia fue introducir control gradualmente para estudiar cómo afecta al sistema el incremento de esta constante. Los resultados se muestran en la Figura 6.2 y en la Figura 6.3.



(a) Posición de la esfera en el Eje X de la simulación utilizando controlador P (b) Posición de la esfera en el Eje Y de la simulación utilizando controlador P

Figura 6.2: Representación de la posición de la esfera sobre los ejes X e Y en la simulación utilizando controlador P con una $K_p = 0,1$

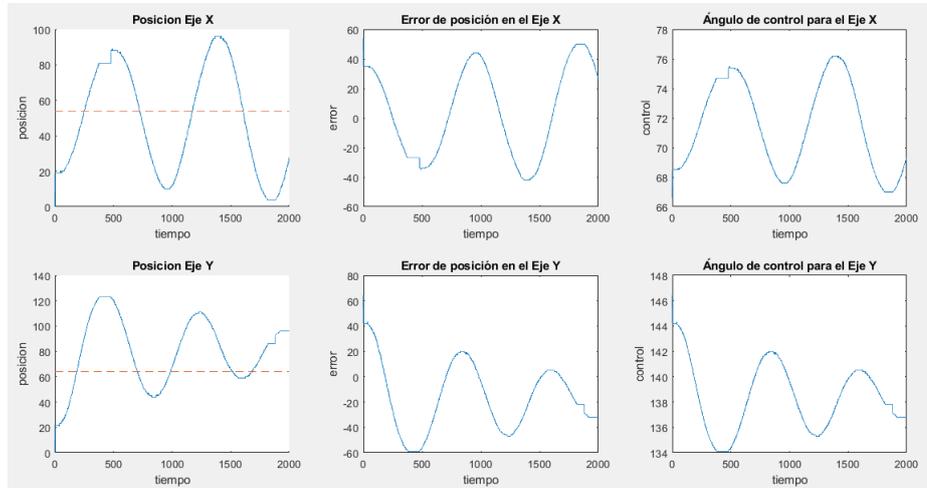


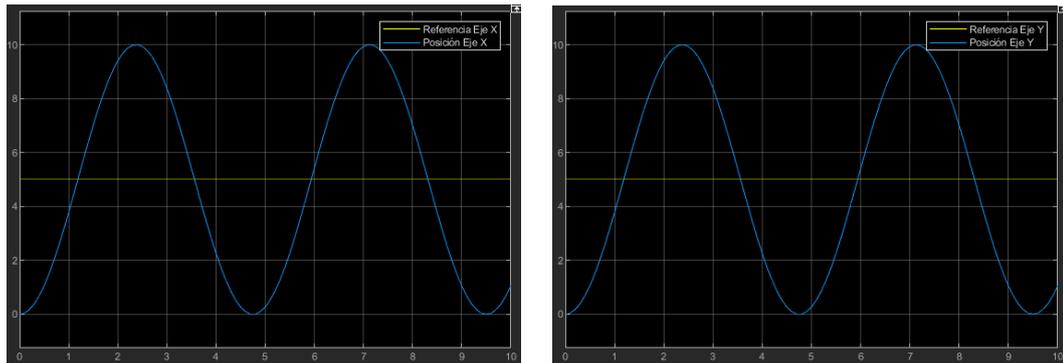
Figura 6.3: Respuesta del sistema real utilizando un controlador P con $K_p = 0,1$.

Como se puede observar en los resultados de la simulación, se obtiene una respuesta del sistema lenta, con un tiempo de subida próximo a 2 segundos. Esto hace que, durante los 10 segundos que dura el experimento, la respuesta del sistema oscile y no llegue a ajustarse al valor de la referencia.

Si se observa la respuesta del sistema real, se ve que tiene un comportamiento similar al caso simulado. La posición de la esfera también oscila y no llega a ser

estable. Esto se traduce en que el error nunca llegue a ser nulo y el ángulo de control tenga variaciones muy grandes durante el tiempo que dura el experimento.

A continuación, se ha incrementado ligeramente el valor de K_p , $K_p = 0,25$, con el objetivo de que la superficie del sistema real no se incline demasiado, evitando así que se acelere la esfera. Los resultados obtenidos son los siguientes:



(a) Posición de la esfera en el Eje X de la simulación utilizando controlador P (b) Posición de la esfera en el Eje Y de la simulación utilizando controlador P

Figura 6.4: Representación de la posición de la esfera sobre los ejes X e Y en la simulación utilizando controlador P con una $K_p = 0,25$

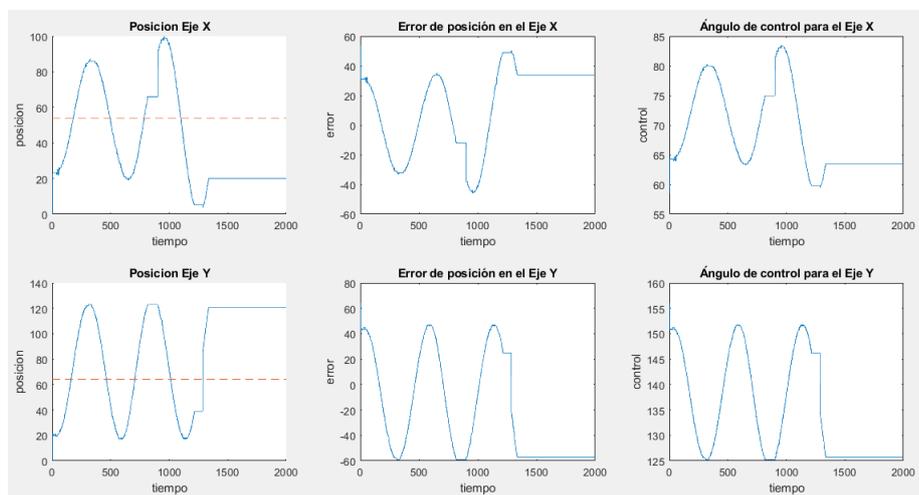


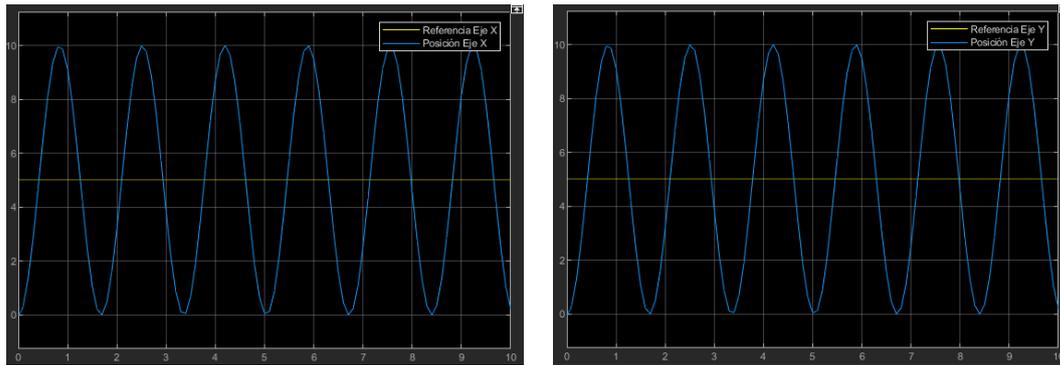
Figura 6.5: Respuesta del sistema real utilizando un controlador P con $K_p = 0,25$.

En este caso, en la simulación se observa una respuesta más rápida que para $K_p = 1$ con un tiempo de subida algo superior a 1s. Por otro lado, el control sigue sin ser suficiente ya que la respuesta tampoco se ajusta al valor de referencia.

En el sistema real tenemos que la esfera se sale del plato antes de que finalice

el experimento, como se observa en el punto en el que la gráficas toman un valor constante.

Por último, como los resultados obtenidos hacen pensar que se necesita aplicar más control, para este experimento se ha decidido utilizar $K_p = 2$. Se ha considerado que si este incremento no es suficiente para controlar la estructura, en este sistema una constante de proporcionalidad alta no supone un control mayor sobre la posición de la esfera. Los resultados obtenidos se muestran a continuación:



(a) Posición de la esfera en el Eje X de la simulación utilizando controlador P

(b) Posición de la esfera en el Eje Y de la simulación utilizando controlador P

Figura 6.6: Representación de la posición de la esfera sobre los ejes X e Y en la simulación utilizando controlador P con una $K_p = 2$

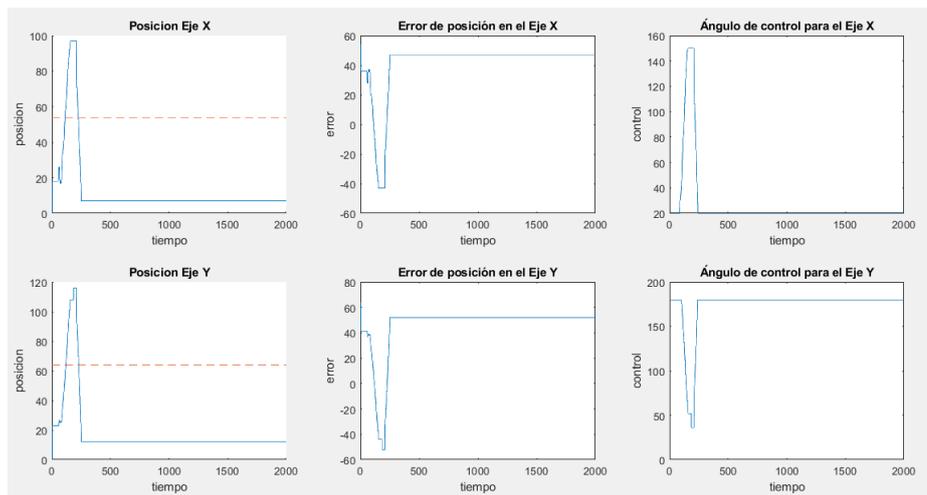


Figura 6.7: Respuesta del sistema real utilizando un controlador P con $K_p = 2$.

En este caso, se observa que, en la simulación, el tiempo de respuesta es muy pequeño, aparentemente la mitad que en el experimento anterior. Esto hace que en la Figura 6.11 aparezcan más oscilaciones. Sin embargo, la respuesta tampoco se ajusta

al valor de la referencia.

En los resultados del sistema real, se obtiene que la esfera se sale rápidamente del plato. Esto se debe a que, a medida que se aumenta el valor de K_p , el ángulo de control será cada vez mayor, lo que provoca que la superficie se incline más y la esfera se acelere. Puesto que el controlador K_p no es lo suficientemente rápido para actuar frente a esta aceleración, es muy difícil conseguir que la esfera no se salga del plato.

6.2. Controlador PD

En esta sección se muestra el análisis de los datos obtenidos utilizando un controlador proporcional derivativo. Como punto de partida, se han implementado los parámetros calculados en la Sección 4.5. Posteriormente, se han variado los parámetros K_p y K_D para entender cómo afecta cada parámetro a la respuesta del sistema.

6.2.1. Comparativa entre el comportamiento del sistema real y el sistema ideal

Para estudiar el comportamiento del sistema ideal, en esta ocasión, se ha añadido un bloque para el controlador PD en el esquema de la simulación realizada en Simulink como se muestra en la Figura 6.8.

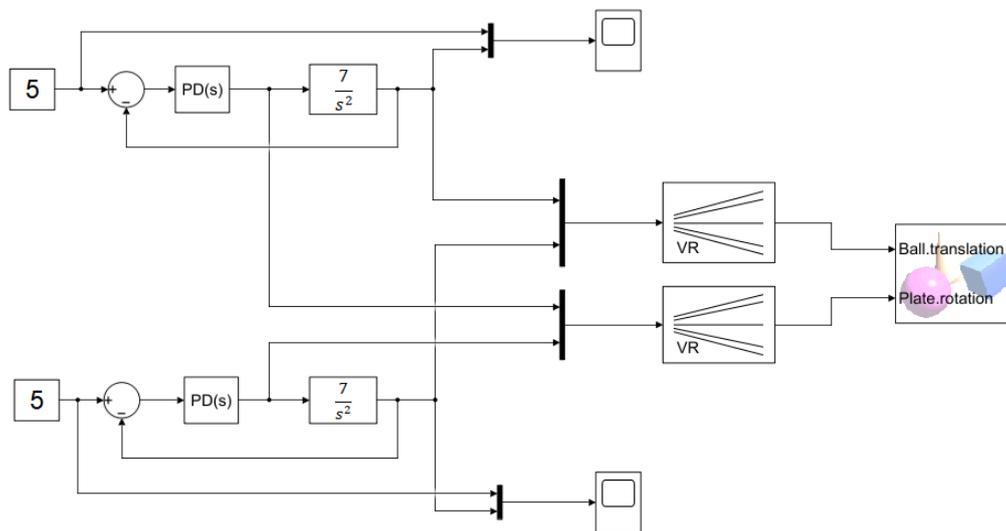
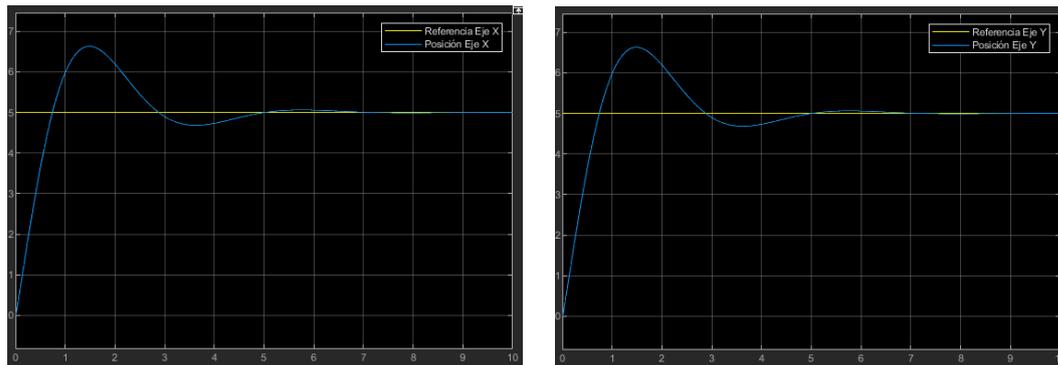


Figura 6.8: Diagrama de bloques del sistema simulado en Simulink con controlador PD

En primer lugar, se han implementado las constantes $K_p = 0,386$ y $K_D = 0,2154$, valores sintonizados en la Sección 4.5, y se han obtenido los resultados que se muestran en la Figura 6.9 y la Figura 6.10.



(a) Posición de la esfera en el Eje X de la simulación utilizando controlador PD (b) Posición de la esfera en el Eje Y de la simulación utilizando controlador PD

Figura 6.9: Representación de la posición de la esfera sobre los ejes X e Y en la simulación utilizando controlador PD con una $K_p = 0,386$ y $K_D = 0,2154$.

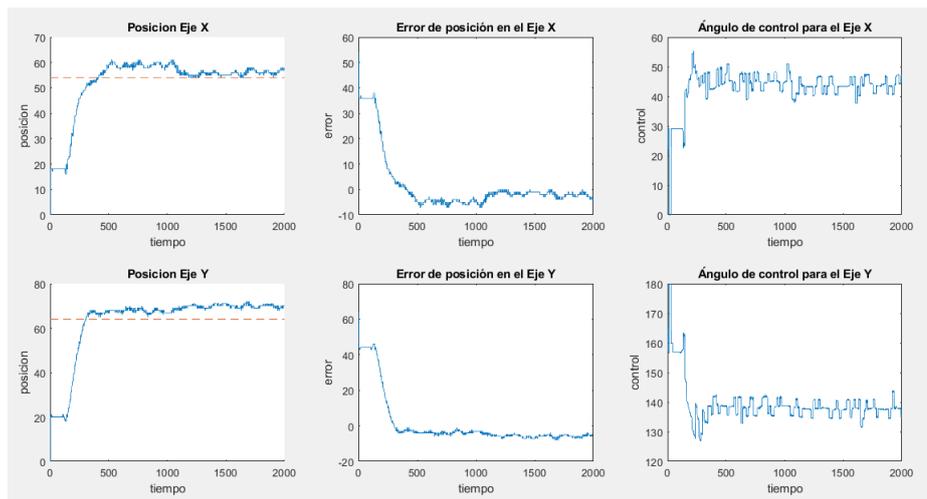
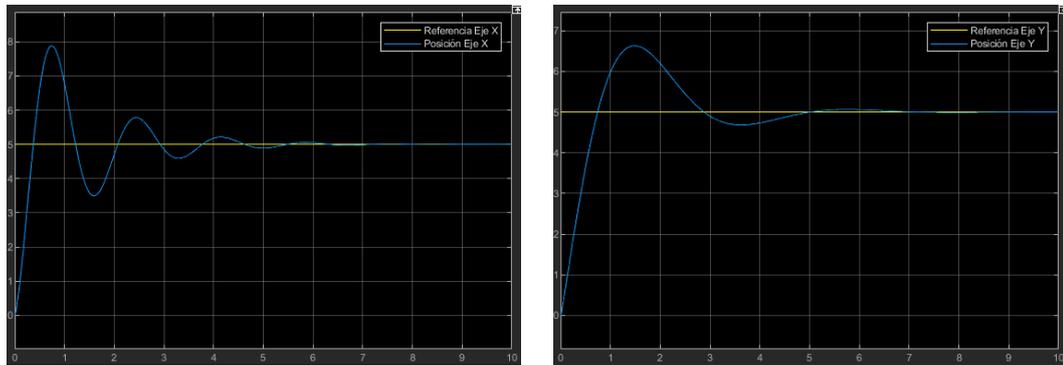


Figura 6.10: Respuesta del sistema real utilizando un controlador PD con $K_p = 0,386$ y $K_D = 0,2154$.

Como se puede observar en las imágenes, tanto el sistema real como el sistema ideal tienen un mejor comportamiento que con el controlador P. En la simulación, tenemos un tiempo de subida inferior a 1s y la respuesta del sistema se ajusta perfectamente al valor de la referencia.

En el sistema real, se ve que tanto la posición en el eje X como la posición en el eje Y también tienen el comportamiento esperado y se ajustan a los valores de referencia. Además, se observa que la sobreelongación máxima es inferior que para el sistema simulado. Por otro lado, en las gráficas del error puede verse que el error a partir de cierto punto es prácticamente nulo mientras que los ángulos de control que se aplican a los servomotores, son cada vez más constantes cuando el sistema es estable.

A continuación, se le ha dado a K_p del eje X un valor de 2 para analizar las diferencias entre la respuesta del sistema utilizando este controlador y el controlador P. El resto de parámetros se han mantenido exactamente igual que en el caso anterior. Los resultados se muestran a continuación:



(a) Posición de la esfera en el Eje X de la simulación utilizando controlador PD (b) Posición de la esfera en el Eje Y de la simulación utilizando controlador PD

Figura 6.11: Representación de la posición de la esfera sobre los ejes X e Y en la simulación utilizando controlador PD con $K_{px} = 2$, $K_D = 0,2154$ y $K_{py} = 0,386$.

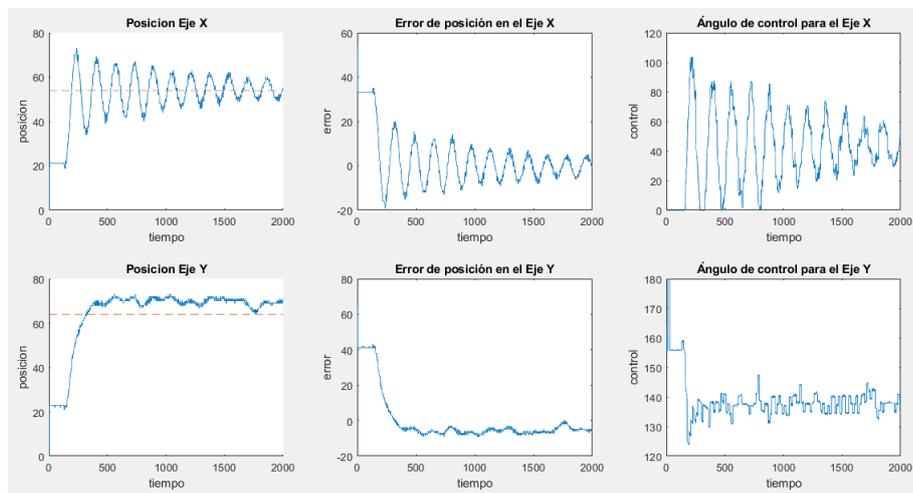


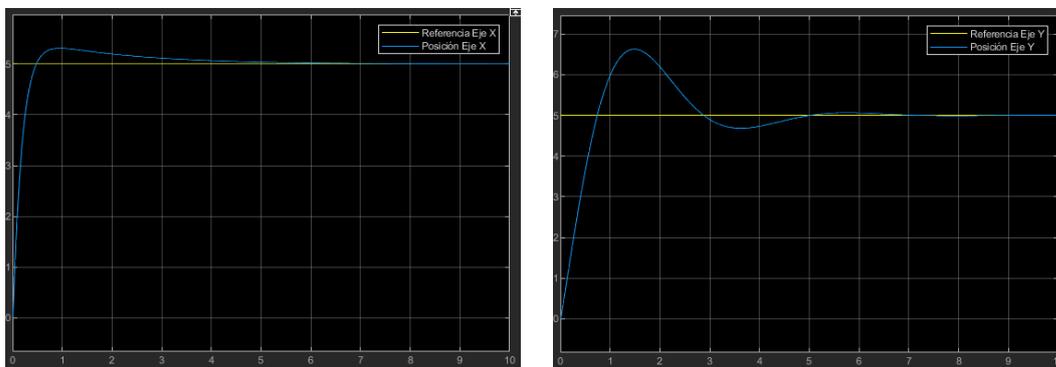
Figura 6.12: Respuesta del sistema real utilizando un controlador PD con $K_{px} = 2$, $K_D = 0,2154$ y $K_{py} = 0,386$.

En este caso, en la simulación se tiene que, al haber aumentado K_p para el eje X, se producen más oscilaciones antes de llegar a la estabilidad en la respuesta para el eje X. Por otro lado, el eje Y no se ve afectado por el comportamiento del eje X y su respuesta es idéntica al caso anterior.

En el sistema real se obtiene que la esfera tampoco se sale del plato, pero han

aumentado significativamente las oscilaciones en el Eje X. El error en este eje va siendo cada vez menor al igual que la variación de los ángulos de control, pero durante los 10 segundos que dura el experimento, el sistema no llega a ser estable. Se observa que, mientras el Eje X está corrigiendo su posición, la posición en el eje Y sí ha llegado a ser estable y su comportamiento es similar al del caso anterior. Es decir, en el sistema real, el eje Y tampoco se ve afectado por lo que suceda en el eje X.

Si ahora se modifica el valor de la constante K_{Dx} haciendo $K_{Dx} = 0,8$, que es aproximadamente el doble que K_p y se dejan el resto de parámetros con su valor calculado, se puede analizar qué sucede si K_D es mayor que K_p . Los resultados obtenidos son los siguientes:



(a) Posición de la esfera en el Eje X de la (b) Posición de la esfera en el Eje Y de la simulación utilizando controlador PD simulación utilizando controlador PD

Figura 6.13: Representación de la posición de la esfera sobre los ejes X e Y en la simulación utilizando controlador PD con $K_p = 0,386$, $K_{DX} = 0,8$ y $K_{DY} = 0,2154$.

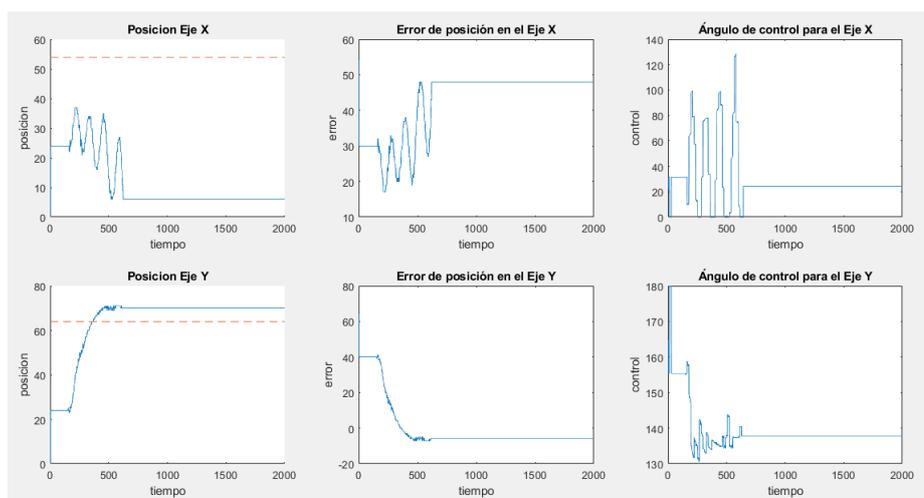
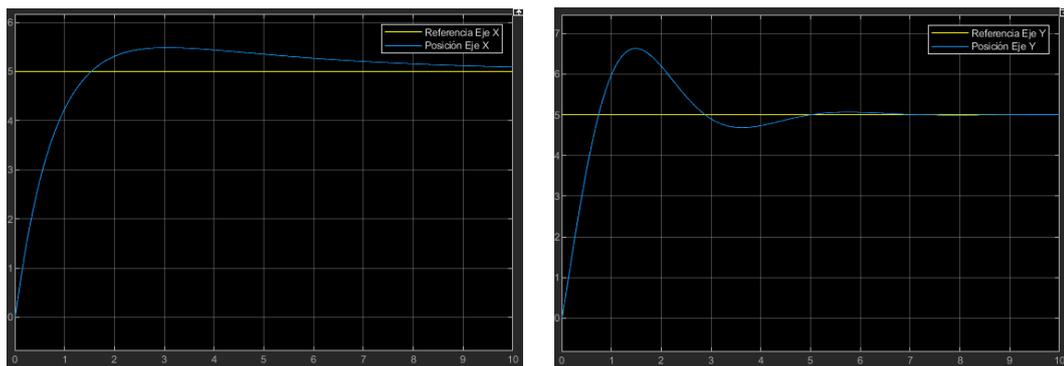


Figura 6.14: Respuesta del sistema real utilizando un controlador PD con $K_p = 0,386$, $K_{DX} = 0,8$ y $K_{DY} = 0,2154$.

En este caso, en la simulación se tiene que la posición de la esfera para el eje X consigue ajustarse a la referencia. Además, la sobreelongación máxima es mucho menor que en el resto de experimentos y el tiempo de establecimiento también parece menor que para el Eje Y, que sigue comportándose igual que en los casos anteriores.

Por el contrario, a pesar de estar utilizando un controlador con el que según la simulación la esfera se debería mantener sobre la superficie, se tiene que la esfera se sale de la misma en el sistema real. Como se puede observar en los resultados obtenidos, en el eje X, la posición de la esfera cada vez tiene más error y el ángulo que es necesario aplicar para controlar la estructura a medida que avanza el experimento sufre más variaciones. Por el contrario, la esfera es capaz de corregir su posición sobre el eje Y antes de salirse por el eje X, teniendo un comportamiento similar al de los casos anteriores.

Por último, se quiere analizar qué pasaría si le damos a K_p un valor muy pequeño, un orden de magnitud por debajo de las constantes calculadas. Para ello, se utiliza $K_{px} = 0,05$. Los resultados obtenidos son los siguientes:



(a) Posición de la esfera en el Eje X de la simulación utilizando controlador PD (b) Posición de la esfera en el Eje Y de la simulación utilizando controlador PD

Figura 6.15: Representación de la posición de la esfera sobre los ejes X e Y en la simulación utilizando controlador PD con $K_{px} = 0,05$, $K_D = 0,2154$ y $K_{py} = 0,386$.

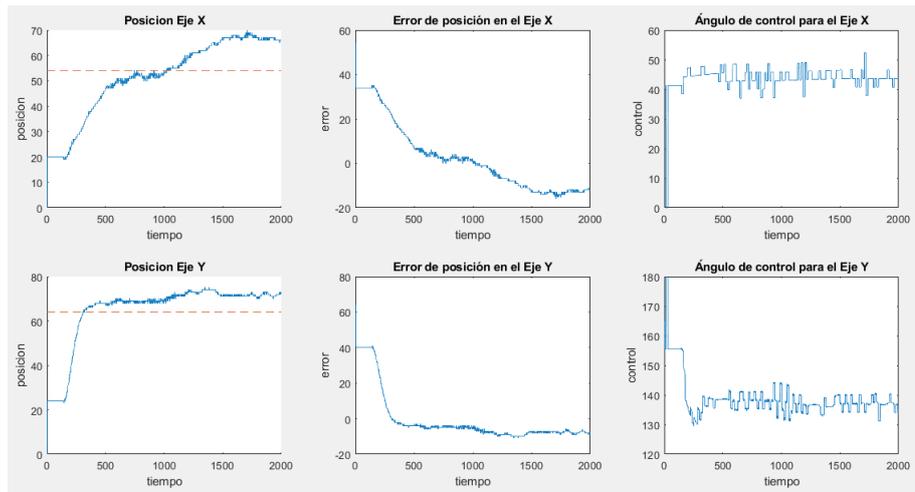


Figura 6.16: Respuesta del sistema real utilizando un controlador PD con $K_{px} = 0,05$, $K_D = 0,2154$ y $K_{py} = 0,386$.

En la simulación, en el eje X se ve un comportamiento muy similar al del caso anterior, pero en este caso el tiempo de establecimiento es mayor y la respuesta es más lenta. El eje Y tampoco se ve afectado.

Para el caso real, se obtiene que la gráfica de la posición en el eje X crece de manera muy lenta hasta que supera el valor de referencia, pero no llega a salirse del plato durante el tiempo que dura el experimento. Se observa también que el error se va reduciendo lentamente y que las variaciones en el ángulo de control son muy pequeñas debido a lo pequeña que es la constante proporcional. La posición en el eje Y, sigue sin verse afectada por el comportamiento en el eje X.

Si se realizara un experimento introduciendo una K_D muy pequeña, se obtendrían resultados similares al comportamiento cuando se utiliza un controlador P. Esto se debe a que el controlador P es un caso particular del controlador PD ya que, un controlador P, es un controlador PD con $K_D = 0$.

Capítulo 7

Conclusiones y líneas futuras

7.1. Conclusiones

En este TFG se pretendía controlar un sistema Ball And Plate en el que los objetivos consistían en actuar sobre la posición de una esfera sobre un plato y que ésta no se saliera de la superficie. Estos objetivos han sido solventados satisfactoriamente en el sistema implementado.

Para llegar al sistema final, en primer lugar, se investigó acerca de los componentes que se iban a utilizar. Posteriormente, se analizaron las fuerzas a las que iba a estar sometido el sistema con el objetivo de obtener la función de transferencia. El primer problema surgió al intentar hacer una simulación del sistema en Simulink. Tras modelar el sistema real con el programa *Autodesk Inventor*, resultó imposible importarlo a Simulink debido a un tema de compatibilidades, que no se consiguió solucionar. Finalmente se optó por representar la estructura en 3D utilizando la herramienta *VR Sink* de Simulink.

Otro problema se presentó fue a la falta de documentación que había de la cámara que se decidió utilizar. Además, el protocolo SPI se encontraba en una versión beta al inicio del TFG, lo que provocó que finalmente se optara por el puerto serie como protocolo de comunicación entre la cámara y el microcontrolador.

Por otro lado, al implementar los controladores, se observó que los servomotores que se tenían inicialmente no respondían lo suficientemente rápido. Para solucionar este problema, se optó por sustituirlos por otro modelo y alimentarlos con la tensión recomendada por el fabricante para llegar a su velocidad máxima.

Finalmente, como conclusión, se puede determinar que este sistema o una adaptación del mismo es viable para su uso académico. Además, en la siguiente sección, se exponen una serie de propuestas que podrían ser implementadas con el objetivo de hacer un análisis más profundo y crear un sistema más completo.

7.2. Líneas futuras

Como líneas futuras de trabajo partiendo del sistema Ball And Plate en el que se ha trabajado se plantean las siguientes:

En primer lugar, hay que tener en cuenta que no solamente se ha llegado a implementar un sistema Ball And Plate en el que se consigue que la esfera no se salga de la superficie y llegue al centro de la misma lo más rápido posible. Se ha implementado un sistema de control capaz de controlar la posición de una esfera sobre una superficie y esto puede tener el comportamiento descrito anteriormente u otro completamente distinto. Por ejemplo, hacer que la esfera siga una trayectoria cuadrada sin salirse de la superficie en lugar de que la esfera vaya únicamente al centro. Para lograr este objetivo bastaría con ir cambiando las posiciones de referencia que debe alcanzar la esfera. En esta línea, se podría trabajar en implementar que la esfera tenga una serie de comportamientos sobre la superficie y que el usuario, mediante un pulsador o una App sea capaz de seleccionar como se comporta la esfera en cada momento.

Por otro lado, la tarea de la obtención de la posición de la esfera en un sistema Ball And Plate suele implementarse de dos maneras: utilizando una cámara o utilizando una pantalla resistiva. En este TFG se ha utilizado una cámara pero como línea futura podría acoplarse también la parte de la pantalla resistiva con el objetivo de tener un sistema en el que se puedan obtener los datos de manera distinta. Además, esto permitiría hacer un estudio de qué sensor sería mejor para este sistema.

Por último, destacar que en este TFG se ha trabajado únicamente con 2 controladores pero existen un gran número de ellos. Una posible mejora de este proyecto podría consistir en diseñar e implementar otros controladores con el objetivo de analizar su comportamiento y estudiar si, con alguno de ellos, se podría mejorar la funcionalidad de nuestro sistema.

References

- [1] Carmine Noviello. Mastering stm32. 2018.
- [2] Estructura Ball and Plate. https://www.researchgate.net/figure/Structure-and-schematic-of-ball-and-plate-system_fig1_352172266.
- [3] Félix Monasterio-Huelin y Álvaro Gutiérrez Martín. Introducción a los sistemas electrónicos de control. <http://robolabo.etsit.upm.es/asignaturas/seco/apuntes/introSECO.pdf>, 2021.
- [4] Juan Castromil. Starship, el robot de reparto urbano sobre ruedas. <https://clipset.com/starship-el-robot-de-reparto-urbano-sobre-ruedas/>, 2016.
- [5] Dario Enfermero. El robot peanut lleva comida a la gente en cuarentena en china. <https://diarioenfermero.es/el-robot-peanut-lleva-comida-a-la-gente-en-cuarentena-en-china/>, 2020.
- [6] OpenMV Cam H7. <https://openmv.io/products/openmv-cam-h7>.
- [7] Transformada de Hough. <https://porprofesionalmic.files.wordpress.com/2015/09/investigacion-documental-transformada-hough.pdf>.
- [8] Microcontrolador STM32F446RE. https://www.st.com/en/microcontrollers-microprocessors/stm32f446re.html#st_all-features_sec-nav-tab.
- [9] Servomotor Futaba S3001. <https://servodatabase.com/servo/futaba/s3001>.
- [10] Digi-Key. Por qué y cómo usar la interfaz periférica serial para simplificar las conexiones entre dispositivos. 2019.
- [11] Martin Nenkov. Modelado y diseño de un sistema de control de una estructura de balanceo con dos grados de libertad. 2021.
- [12] Félix Monasterio-Huelin y Álvaro Gutiérrez Martín. Diseño. <http://robolabo.etsit.upm.es/asignaturas/seco/apuntes/design.pdf>, 2020.
- [13] Félix Monasterio-Huelin y Álvaro Gutiérrez Martín. Análisis. <http://robolabo.etsit.upm.es/asignaturas/seco/apuntes/analisis.pdf>, 2022.
- [14] Universidad de Alcalá. Análisis en frecuencia: lugar de raíces. [https://www.cartagena99.com/recursos/alumnos/apuntes/Tema9\(1\).pdf](https://www.cartagena99.com/recursos/alumnos/apuntes/Tema9(1).pdf), 2022.

Apéndice A

Esquema de montaje del sistema

A continuación se muestran las conexiones entre los distintos componentes que forman parte del sistema Ball And Plate.

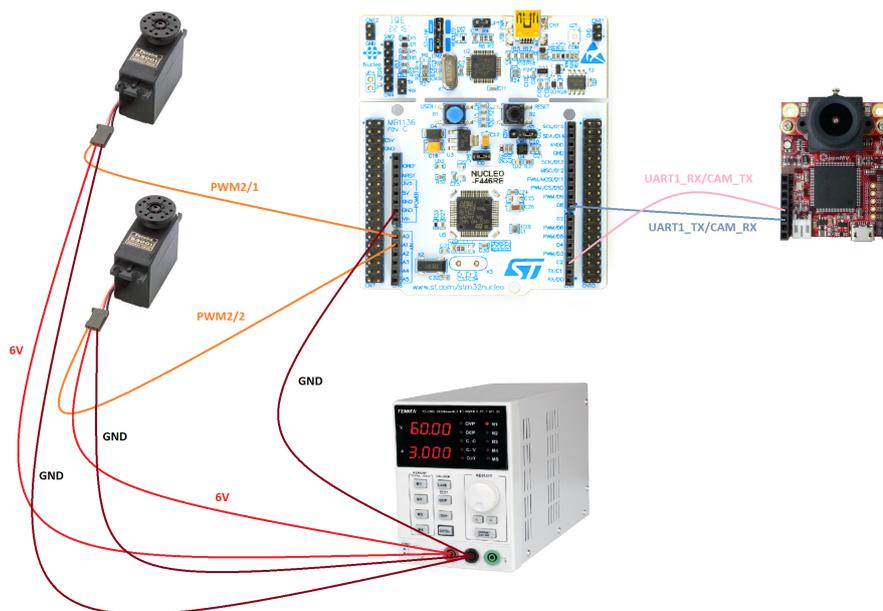


Figura A.1: Conexiones entre los distintos componentes de nuestro sistema Ball And Plate.

Apéndice B

Aspectos éticos, económicos, sociales y ambientales

El objetivo de este TFG ha consistido en diseñar, modelar e implementar un sistema de control de una estructura de balanceo con dos grados de libertad, con el objetivo de tener un sistema funcional que pueda ser replicado fácilmente.

Este TFG tiene un impacto social bastante alto ya que va a ayudar a los alumnos interesados a que sean capaces de diseñar un sistema de control e implementarlo. Una de las metas de este proyecto es que los alumnos de la asignatura “Sistemas Electrónicos de Control” trabajen sobre este sistema o una adaptación del mismo desde el próximo curso.

En cuanto al impacto económico, una de las grandes ventajas de este proyecto es que el precio de sus componentes no es muy elevado como se puede ver en el Anexo C. De esta forma se favorece a que pueda haber más puestos de laboratorio en los que se pueda trabajar sobre este sistema, o incluso que los alumnos se animen a comprar sus propios componentes para profundizar más en sus casas.

Por último, el impacto medioambiental de la producción de los componentes se desconoce ya que se fabrican en China. En cuanto a su uso, mientras que está en funcionamiento el sistema solo necesita consumo eléctrico. Además, se dispone de un interruptor en el generador que permite alimentar con electricidad al sistema solamente cuando se vaya a utilizar, ahorrando de esta manera energía. A esto se suma que la vida útil de los componentes, si se utilizan con responsabilidad, es bastante alta.

Apéndice C

Presupuesto económico

Este TFG se ha realizado en el laboratorio de Robótica y Control de la Escuela Técnica Superior de Ingenieros de Telecomunicación de la UPM en un periodo de 9 meses. Para la elaboración del presupuesto necesario se han analizado los costes de personal y los costes de recursos materiales, como se muestra a continuación:

- **Personal:**

	Coste horario (€)	Horas	Total (€)
Director del trabajo	60	30	1800
Estudiante de ingeniería	30	600	18000
TOTAL			19800

Tabla C.1: Costes de personal.

- **Costes de recursos materiales:**

	Tiempo de vida (años)	Uds.	Coste (€)	Amortización (€/mes)	Uso (meses)	Total (€)
STM32F446	5	1	17	0.28	9	2.55
Cámara	5	1	63	1.05	9	9.45
Servomotor	5	2	20	0.67	9	6
Generador	10	1	125	1.04	9	9.375
Estructura	5	1	50	0,83	9	7.5
TOTAL						34.875

Tabla C.2: Costes de recursos materiales.

	Coste
Costes de personal	19800€
Costes de material	34,875€
Subtotal	19834,86€
IVA	4165,32€
Total	24000,18€

Tabla C.3: Costes totales.