



Universitat de Lleida

TRABAJO FINAL DE GRADO



ESCOLA
POLITÈCNICA SUPERIOR
UNIVERSITAT DE LLEIDA
INSPIRING THE FUTURE

Estudiante: Guillermo Doste Torres

Titulación: Grado en ingeniería electrónica industrial y automática

Título de Trabajo Final de Grado:

Posicionamiento de una esfera sobre un carril pivotante mediante técnicas de control discreto: Plataforma de uso docente.

Director/a: Tomàs Pallejà Cabré

Presentación

Mes: Septiembre

Año: 2021

Abstract

Presentamos un proyecto que consiste en un tipo de sistema denominado "Beam and Ball", en el que se utilizan técnicas de control discreto para posicionar un cuerpo esférico sobre un carril pivotante. Nuestro objetivo consiste en desarrollar una maqueta sencilla que pueda ser operada por estudiantes, que permita la aplicación de los conocimientos adquiridos durante la docencia sobre los sistemas de control discretos básicos. El prototipo consta de una serie de piezas diseñadas para impresión 3D, que sostienen un servomotor y un carril de aluminio que pivota sobre un eje. El control se realiza mediante una placa de evaluación "Arduino Due", una interfaz gráfica desarrollada en lenguaje C# y unos sensores de proximidad. A nivel visual se ha implementado una tira de LED que muestra la información relacionada con el posicionamiento de la esfera. Debido al ruido de las señales analógicas y holguras mecánicas, en los ensayos se han experimentado dificultades que afectan a la respuesta del sistema y a la adquisición de datos para las simulaciones, a pesar de ello, realizando unos pequeños ajustes el sistema de control proporciona una respuesta aceptable y permite al usuario posicionar la esfera correctamente.

We present a project that consists of a type of system called "Beam and Ball", in which discrete control techniques are used to position a spherical body on a pivoting rail. Our objective is to develop a simple model that can be operated by students, which allows the application of the knowledge acquired during teaching on the basic discrete control systems. The prototype consists of a series of parts designed for 3D printing, which hold a servomotor and an aluminum rail that pivots on an axis. The control is carried out by means of an "Arduino Due" evaluation board, a graphical interface developed in C # language and some proximity sensors. At a visual level, an LED strip has been implemented that shows the information related to the positioning of the sphere. Due to the noise of the analog signals and mechanical clearances, in the tests difficulties have been experienced that affect the response of the system and the acquisition of data for the simulations, despite this, making a few small adjustments the control system provides a response acceptable and allows the user to position the sphere correctly.

Índice

Introducción	5
Objetivos	5
Situación actual	7
Antecedentes teóricos	9
Elementos mecánicos	10
Diseño conceptual	10
Inclinación del perfil	11
Diseño del mecanismo	16
Elementos estructurales	21
Elementos electrónicos	25
Diseño conceptual	25
Arduino DUE	26
Servomotor	27
Sensor de proximidad	29
Tira de LED	31
Fungible	33
Sistema de control	35
Controlador PID	35
Anti derivative kick	38
Derivative filter	39
Anti wind-up	41
Diseño del modelo matemático	43
Resultados simulados	48
Entrada escalón	50
Entrada senoidal	52
Entrada rampa	54
Programación	56
Interfaz gráfica	56
Programación de la tira de LED	58
Calibración de los sensores de proximidad	60
Control del ángulo de rotación del servomotor	63
Controlador	65
Discusión de resultados	67
Conclusiones	69

Trabajo futuro	70
Mejoras mecánicas	70
Mejoras electrónicas	70
Mejoras modelo matemático	70
Referencias bibliográficas	71

Introducción

Objetivos

El objetivo principal del proyecto consiste en la realización de una maqueta para uso docente, en la que futuros estudiantes del grado en ingeniería electrónica y automática industrial podrán poner en práctica los conocimientos adquiridos sobre sistemas de control discreto. Mediante el uso de un PID y una señal de referencia los estudiantes podrán variar el ángulo de rotación de un carril pivotante para posicionar una esfera.

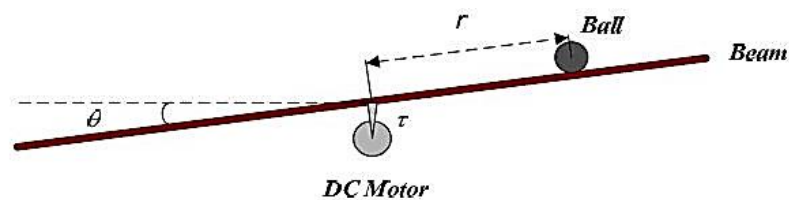


Ilustración 1: Boceto del sistema; <https://www.chegg.com/homework-help/questions-and-answers/consider-ball-beam-system-depicted--system-beam-made-rotate-applying-torque-center-ball-free-q18102424>

El sistema debe ser sencillo para facilitar el aprendizaje, se ha partido de diseños prácticos y económicos, así como la integración de elementos comerciales de fácil acceso o existentes en anteriores proyectos de uso docente. Además, se desea que se pueda realizar la implementación de una interfaz gráfica desarrollada en lenguaje C# por los propios estudiantes de manera que el resultado de este proyecto se adapte a la guía docente actual.

El desarrollo de este proyecto se materializa en el diseño e integración de una serie de elementos:

Elementos estructurales

- Una base que contiene alojamientos en la parte frontal para insertar los componentes electrónicos y de alimentación, unas asas en los laterales y orificios donde se sujetan los demás elementos estructurales.
- Un soporte en el que se instala un servomotor ubicado en el centro de la base y de baja altura para facilitar las conexiones eléctricas.
- Una columna donde se instalan los elementos de sujeción del carril pivotante.
- Un balancín que se acopla al eje del servomotor para transmitir la rotación. El balancín se ha diseñado con la posibilidad de añadir un contrapeso para equilibrar la fuerza de la gravedad.
- Un carril de aluminio en forma de U, en la que se aloja el cuerpo esférico.
- Una barra con articulaciones de cojinetes esféricos en los extremos para conectar el carril de aluminio con el balancín.

- Un eje en el que se sujeta el carril de aluminio.
- Unos rodamientos alojados en la columna y sujetan el eje.
- Unos soportes para los sensores de proximidad ubicados en los extremos del carril de aluminio.
- Un cuerpo esférico con buena reflexión de la luz infrarroja, de poca masa y con un buen coeficiente de fricción estática con el perfil de aluminio.

Sistema de control

- Una placa de evaluación “Arduino Due”.
- Un módulo de alimentación de tipo “Breadboard”, para facilitar las conexiones.
- Un servomotor, para generar el movimiento de rotación del carril pivotante.
- Una tira de LED de control independiente para indicar el estado de la señal de referencia y del posicionamiento de la esfera.
- Una interfaz gráfica desarrollada en lenguaje C# para incrementar las posibilidades de aprendizaje. La interfaz permite recibir información y configurar los parámetros del sistema en tiempo real.

Situación actual

En la actualidad existen múltiples diseños desarrollados por alumnos y profesores de varias universidades (Jover y Casanova, 2019; Luis Ángel Blas Sánchez et al., 2015; Alirio, Romero y Silva, 2010; Araujo, Vivas y Alvarado, 2018), incluso algunos diseños se comercializan bajo el nombre de una firma (Tecquipment, CE106 y RT121; Quanser, SRV02; ERM, Acrome). Los materiales empleados en la construcción, el tipo de sistema de control, la disposición del mecanismo y el tamaño del concepto ha sido objeto de estudio durante años llegando a obtener sistemas funcionales muy acertados.

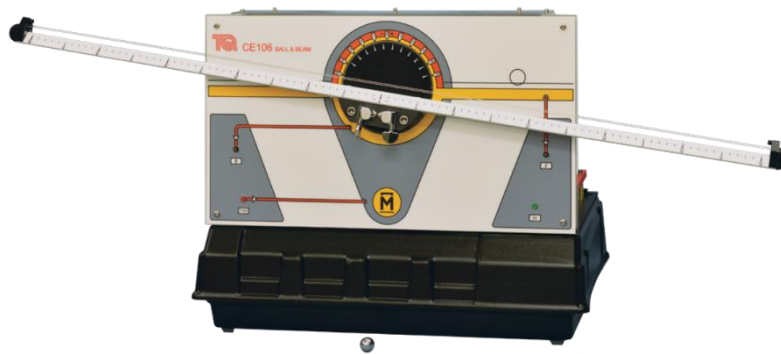


Ilustración 2: Tecquipment, CE106



Ilustración 3: Quanser, SRV02



Ilustración 4: ERM, Acrome

Tras una visión general se observa que el prototipo más utilizado consiste en un sistema mecánico sencillo fabricado en impresión 3D (Espeso, 2015), el diseño de una geometría específica para el control de un carril pivotante con muy poco recorrido y ángulo de rotación, la integración de una placa de evaluación con microcontrolador, un módulo de alimentación, sensores de proximidad, un motor de corriente continua o servo y en algunos casos un “display” informativo o interfaz gráfica.

El cometido del sistema consiste en la estabilización del movimiento de un cuerpo esférico hasta detenerlo en una posición determinada, mediante el uso de un algoritmo que procesa o genera variables en un microcontrolador a partir de los datos de un sensor/es.

Se utilizan varios tipos de sistemas de control, algunos de ellos muy avanzados y de gran precisión, aunque en términos generales el sistema de control más utilizado es el sistema de control PID.

Antecedentes teóricos

Las investigaciones previas muestran que han sido necesarios estudios relacionados con:

- El comportamiento físico de los elementos fijos, algunos sistemas existentes constan de un estudio de la resistencia de los elementos estructurales debido a la baja resistencia de los elementos de construcción, sobre todo en elementos críticos donde se sustentan los elementos móviles. Estos estudios concluyen que debido al escaso peso de los elementos móviles en la mayoría de los diseños con una ligera sobredimensión de las piezas de impresión 3D es suficiente para garantizar la durabilidad.
- El comportamiento físico de los elementos móviles, la mayoría de los sistemas existentes cuentan con un estudio de la resistencia de los materiales, y un estudio de la cinemática y dinámica para determinar los movimientos e inercias de los componentes para obtener una modelización matemática del sistema y conocer las limitaciones del sistema. A nivel teórico se obtienen unos resultados que sirven para aproximar el comportamiento real del sistema, que facilitan los ajustes posteriores necesarios para alcanzar los resultados deseados.
- El estudio de sistemas de control, el sistema de control más utilizado es el sistema de control PID aunque se utilizan distintas formas de implementarlo utilizando versiones muy sencillas hasta las más avanzadas dependiendo del tipo de sistema. Los resultados apreciados son proporcionales a la complejidad del sistema utilizado y a la precisión del algoritmo empleado, en el caso del sistema de control PID los resultados son funcionales admitiendo bajos niveles de error debido a imprecisiones del diseño y perturbaciones externas.
- Estudio de los sensores y/o actuadores, se han implementado varios tipos de sensores y actuadores en los sistemas conocidos, siendo el más común, el uso de sensores de distancia por infrarrojos y servomotores como actuador debido a su sencillez en el control de la posición.

Elementos mecánicos

Diseño conceptual

Considerando las investigaciones previas y con el fin de alcanzar los objetivos propuestos en este proyecto, se ha realizado un estudio de los elementos mecánicos.

El carril pivotante podría estar formado por dos barras cilíndricas de aluminio unidas a través de tres pletinas del mismo material. El problema surge en el ensamblaje del conjunto, pues aparecen deformaciones en el carril si se sueldan las piezas. Las deformaciones afectarían al movimiento de la esfera, por ello se opta por un perfil cuadrado extruido en aluminio, dándole forma de "U" eliminando una cara para que la bola ruede sobre el carril sin salirse. Además, este tipo de perfil facilita la instalación de la tira de leds y demás elementos de sujeción y soportes. Al tratarse de un elemento comercial conviene consultar la normalización de dimensiones que se pueden encontrar en el mercado considerando los diámetros posibles de la esfera que se aloja en el carril.

La esfera más adecuada para el carril requiere una masa muy baja, un diámetro que permita una correcta detección y que sea de color claro (preferiblemente blanca). Inicialmente se han realizado pruebas con esferas huecas (pelotas de ping-pong) pero se ha comprobado la masa no es uniforme en la superficie, esto causa un desplazamiento del centro de masas respecto al origen de la geometría, en consecuencia, el movimiento de esfera hueca no es el adecuado. Tras una búsqueda de esferas macizas en los comercios de proximidad, la mejor solución consiste en esferas de EPS (poliestireno expandido) de 60 milímetros de diámetro.

La anchura del carril pivotante depende del diámetro de la esfera, consultando las medidas normalizadas de este elemento comercial, se ha elegido la anchura de 30 milímetros.

Atendiendo a la ficha técnica del sensor de proximidad más adecuado para esta aplicación (Sharp GP2Y0141SK0F), se determina que el carril debería ser de 500 milímetros de longitud (con el uso de 2 sensores de proximidad).

Una vez que se conocen las dimensiones del carril pivotante y de la esfera, se puede iniciar el diseño de los elementos mecánicos. Para obtener el dimensionado de las partes estructurales, primero hay que determinar el ángulo máximo de rotación del carril pivotante.

Inclinación del perfil

Se pretende averiguar el ángulo máximo de rotación del carril pivotante sin que la esfera se deslice por él, la elección de este ángulo, como se demostrará a continuación, depende de la distancia entre las caras interiores del carril pivotante, cuyas paredes tienen un grosor de 2 milímetros. Se trata por tanto del estudio de la fricción o fuerza de rozamiento existente entre ambos elementos.

La fricción se puede definir como la fuerza de contacto que existe entre dos cuerpos y que dificulta o imposibilita el deslizamiento o movimiento entre ambos. Esta fuerza es debida a los enlaces de las moléculas de los cuerpos en aquellos lugares donde las superficies están en íntimo contacto. Cuando la ausencia de movimiento es total, se habla de fricción estática, la cual varía desde cero hasta un valor máximo.

La fricción estática actúa en sentido contrario al componente de una fuerza aplicada a lo largo de la superficie de deslizamiento. Cuando esta fuerza aplicada es tal que produce el movimiento relativo entre los cuerpos, se habla de fricción dinámica, la cual actúa en sentido contrario al de la velocidad de un cuerpo moviéndose sobre el otro, en su punto de contacto. Por tanto, la magnitud de la fricción dinámica es menor que el valor máximo de la fricción estática.

Ambas magnitudes de fricción son independientes del área de contacto aparente entre las dos superficies, y son proporcionales a la magnitud N de la fuerza normal que ejerce una de las superficies sobre la otra. La constante de proporcionalidad que relaciona la fuerza de fricción con la fuerza normal es el coeficiente de fricción μ . Esta constante adimensional, que depende del tipo de superficies que intervienen, se determina experimentalmente. Estas relaciones se expresan en ecuaciones del siguiente modo:

$$f_e \leq \mu_e \cdot N \quad f_{e,\max} = \mu_e \cdot N \quad f_c = \mu_c \cdot N$$

$$f_c < f_e \quad \Leftrightarrow \quad \mu_c < \mu_e$$

El estudio de la fricción se complica cuando se pretende que la bola ruede sin deslizar sobre el plano inclinado que forma el carril al ser girado un cierto ángulo. De este estudio se obtiene el máximo ángulo que se puede girar el carril pivotante, para que la esfera ruede sin deslizarse.

$$d = 0.026 \text{ [m]}$$

$$R = 0.030 \text{ [m]}$$

$$\theta = \sin^{-1} \left(\frac{d}{2R} \right) \text{ [deg]}$$

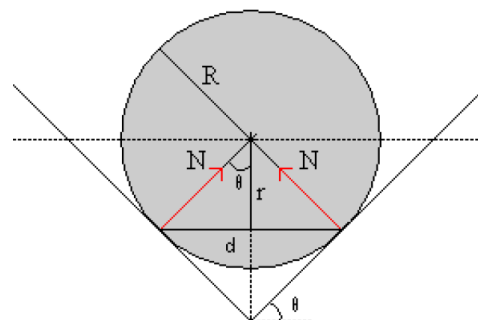


Ilustración 5: Relaciones geométricas

Partimos de la situación en la que la esfera se va a encontrar apoyada en dos puntos del carril pivotante. Supondremos la existencia de rozamientos iguales en cada uno de esos puntos de apoyo. Según las relaciones geométricas dadas en la primera figura, r y d se pueden expresar en función del radio de la esfera R del siguiente modo:

$$r = R \cdot \cos(\theta) \Leftrightarrow \frac{R}{r} = \frac{1}{\cos(\theta)}$$

De igual forma, se definen las magnitudes F_{RT} y N_T como las resultantes de las fuerzas de rozamiento y de las reacciones de la superficie de la viga sobre la esfera respectivamente:

$$F_{RT} = 2 \cdot F_T \quad [N]$$

$$N_T = 2 \cdot N \cdot \cos(\theta) \quad [N]$$

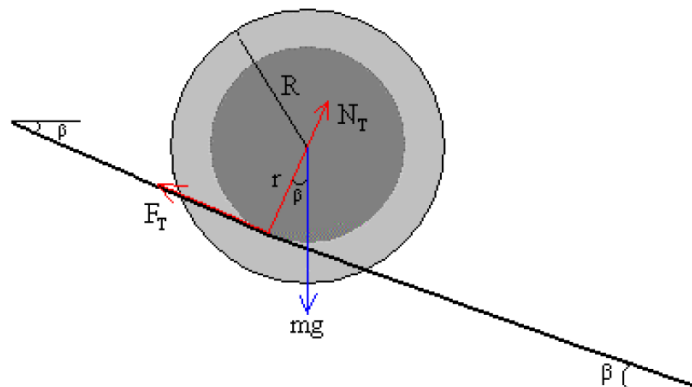


Ilustración 6: Dinámica esfera con plano inclinado

El primer paso consiste en averiguar las ecuaciones de movimiento de la esfera. Cuando ésta rueda hacia abajo por el plano inclinado, el centro de masas adquiere dos tipos de movimiento (siendo I_c el momento de inercia de una esfera maciza):

- Traslación: $m \cdot g \cdot \text{sen}(\beta) - F_{RT} = m \cdot a$
- Rotación: $F_{RT} \cdot r = I_c \cdot \alpha$

Ambos movimientos se relacionan a través de la condición de rodadura:

$$a = r \cdot \alpha \quad [m/s^2]$$

Además, si se hace equilibrio de fuerzas en la dirección perpendicular al plano inclinado se obtiene el valor de la reacción total en la superficie:

$$N_T = m \cdot g \cdot \cos(\beta) \quad [N]$$

A partir de las ecuaciones anteriores obtenemos la siguiente igualdad de la que podemos obtener la aceleración a , y la fuerza de rozamiento F_T y de la reacción N en un apoyo.

$$m \cdot g \cdot \text{sen}(\beta) - \frac{I_c \cdot \alpha}{r} = m \cdot a \Leftrightarrow m \cdot g \cdot \text{sen}(\beta) - \frac{I_c \cdot a}{r^2} = m \cdot a$$

Para una esfera maciza, el momento de inercia con respecto al centro de masas es:

$$I_C = \frac{2}{5} \cdot m \cdot R^2 \quad [Kg \cdot m]$$

Sustituyendo esta expresión en la ecuación anterior, se obtiene finalmente el valor de a:

$$a = g \cdot \text{sen}(\beta) \cdot \frac{1}{1 + \frac{2}{5} \cdot \left(\frac{R}{r}\right)^2} \quad [m/s^2]$$

El cálculo de F_T se obtiene de forma similar, pero en este caso sustituyendo la magnitud global F_{RT} por su expresión equivalente en función de la componente:

$$m \cdot g \cdot \text{sen}(\beta) - 2F_T = m \cdot a$$

Haciendo uso de las ecuaciones anteriores, se llega a una expresión en la que sólo hay que despejar F_T :

$$m \cdot g \cdot \text{sen}(\beta) - 2 \cdot F_T = m \cdot r \cdot a \Leftrightarrow m \cdot g \cdot \text{sen}(\beta) - 2 \cdot F_T = m \cdot r \cdot \frac{2F_T \cdot r}{I_C}$$

$$F_T = m \cdot g \cdot \text{sen}(\beta) \cdot \frac{\left(\frac{R}{r}\right)^2}{5 + 2 \cdot \left(\frac{R}{r}\right)^2} \quad [N]$$

La última incógnita que queda por resolverse es N, para la cual se parte de la ecuación originada del equilibrio de fuerzas en la dirección perpendicular a la superficie:

$$2 \cdot N \cdot \cos(\theta) = m \cdot g \cdot \cos(\beta) \Leftrightarrow N = m \cdot g \cdot \cos(\beta) \cdot \frac{1}{2 \cdot \cos(\theta)}$$

$$N = m \cdot g \cdot \cos(\beta) \cdot \frac{R}{2 \cdot r}$$

Como se observa, todas las magnitudes se han expresado en función de la relación de radios R/r , además de depender del ángulo de inclinación β .

Ahora se pretende averiguar el ángulo máximo que se puede inclinar el carril pivotante para que la esfera ruede sin deslizar en su recorrido. Cuando ocurre esto, no se produce movimiento relativo en el punto de contacto, y por tanto se da fricción estática, tomando un valor entre 0 y su máximo, es decir:

$$F_R \leq \mu_e \cdot N$$

Sustituyendo valores en esta condición, se calcula la desigualdad que obtiene el ángulo máximo de inclinación:

$$m \cdot g \cdot \operatorname{sen}(\beta) \cdot \frac{\left(\frac{R}{r}\right)^2}{5 + 2 \cdot \left(\frac{R}{r}\right)^2} \leq \mu_e \cdot m \cdot g \cdot \operatorname{cos}(\beta) \cdot \frac{R}{2 \cdot r}$$

$$\operatorname{tg}(\beta) \leq \frac{\mu_e \cdot \left(5 + 2 \cdot \left(\frac{R}{r}\right)^2\right)}{2 \cdot \left(\frac{R}{r}\right)}$$

De nuevo se obtiene una expresión que depende de la relación entre radios (es función del ángulo de inclinación β), y en este caso también del coeficiente de rozamiento estático, y por tanto, del tipo de material que elijamos tanto para el carril pivotante como para la esfera. Para obtener el valor del ángulo β es necesario conocer el coeficiente de rozamiento estático y debe realizarse experimentalmente en ausencia de datos proporcionados por el fabricante de la esfera.

Primero se calcula el coeficiente de rozamiento estático teórico asumiendo un valor de β igual a 30 grados, utilizando las ecuaciones anteriores se obtiene que:

$$\mu_e \geq 0.1724$$

Se procede a calcular el tiempo teórico que transcurre desde que se libera la esfera hasta el final del recorrido en un carril de aluminio 1 metro de distancia. Utilizando las ecuaciones del movimiento rectilíneo uniformemente acelerado con condiciones iniciales nulas se obtiene:

$$l = 1 \text{ [m]}$$

$$t_T = \sqrt{\frac{2 \cdot l}{a}} = 0.7835 \text{ [s]}$$

Ahora se realiza un experimento generando las mismas condiciones que en los cálculos teóricos. Se realizan varias medidas utilizando un cronómetro, y se obtiene el siguiente registro de datos:

<i>Medida</i>	<i>Tiempo transcurrido (s)</i>
1	0.97
2	0.91
3	0.87
4	0.93
5	0.89
6	0.98
7	0.92
8	0.93

Tomando un valor medio de todas las lecturas se obtiene el tiempo real en segundos:

$$t_{T_r} = 0.925 \text{ [s]}$$

Cabe señalar que el método utilizado para realizar las medidas no es muy preciso, pero viendo que el valor medio obtenido es superior al teórico, se puede considerar válido para obtener un coeficiente de rozamiento estático aproximado al real. Además, hay que añadir a la imprecisión los defectos en la superficie de la esfera y las perturbaciones de la atmósfera a la que es sometida debido a su escasa masa.

Utilizando la ecuación siguiente ecuación se puede conocer la aceleración real de la esfera:

$$a_{T_r} = \frac{2 \cdot l}{t_{T_r}^2}$$

Conocida la masa en kilogramos se obtiene la fuerza de rozamiento:

$$m = 0.002 \text{ [Kg]}$$

$$F_{T_r} = \frac{m}{2} \cdot (g \cdot \sin(\beta) - a_{T_r})$$

Utilizando la ecuación anterior de la siguiente forma, estimamos el valor del coeficiente de rozamiento estático real:

$$\mu_{e_r} = \frac{F_{T_r}}{N} = 0.255$$

En nuestro caso utilizando el coeficiente de rozamiento estático real podremos llegar a rotar el carril pivotante β grados sin deslizamiento de la esfera:

$$\beta \cong 40 \text{ [deg]}$$

El valor del coeficiente de rozamiento real estimado es superior al valor del coeficiente teórico, a pesar de la incertidumbre en la precisión de las mediciones, podemos validar el experimento concluyendo que la esfera no se desliza para ángulos de rotación inferiores a 40 grados del carril pivotante.

Nuestro sistema apenas requiere grados de rotación para el control de la esfera, el valor de 40 grados puede resultar excesivo, por lo que el estudio del mecanismo se inicia con un límite en 30 grados de rotación del carril pivotante ya que es más que suficiente para el control de la esfera.

Diseño del mecanismo

Considerando un ángulo máximo de aproximadamente 30 grados de rotación para el carril pivotante, es necesario desarrollar un boceto geométrico para dimensionar los demás elementos estructurales. Tras una visión general de los ejemplos de mecanismos previamente diseñados y considerando los objetivos de nuestro sistema se ha obtenido el siguiente tipo de mecanismo:

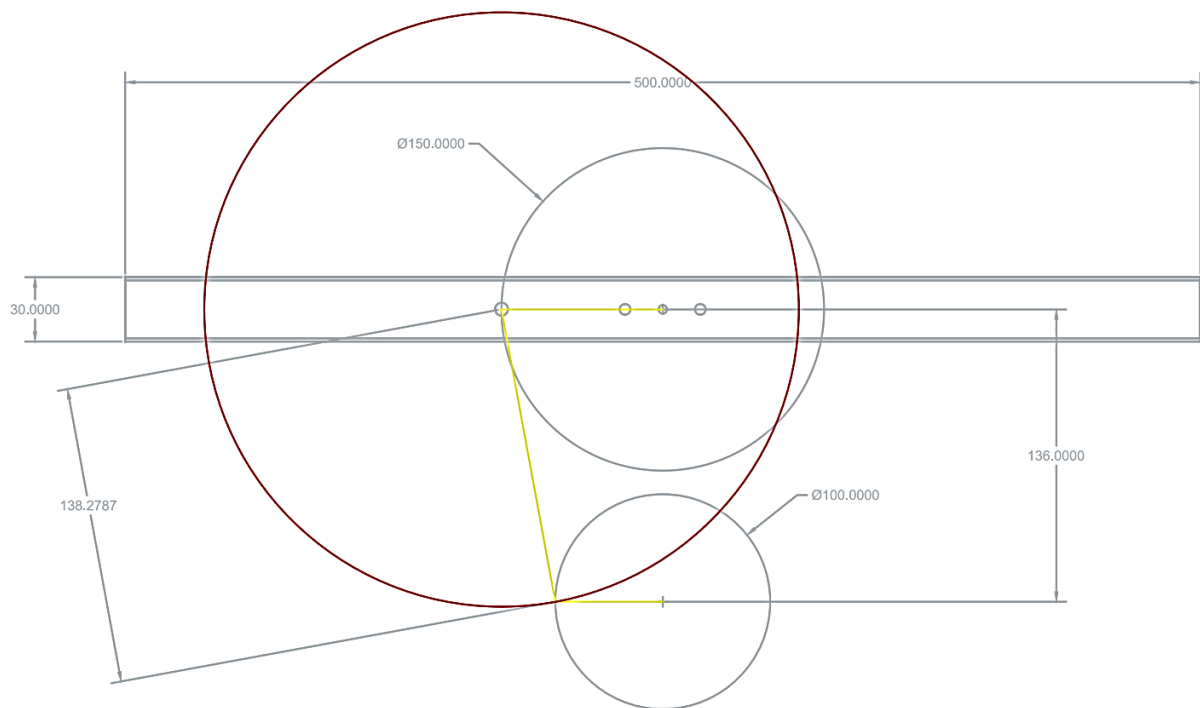


Ilustración 7: Cinemática del mecanismo (1)

Debido a la geometría de este mecanismo, cuando el ángulo de rotación del balancín del servomotor es de α grados, el ángulo de rotación del carril pivotante no se corresponde con los mismos, ya que la disposición de la barra de conexión (líneas amarillas del boceto) funciona como un reductor. Por ello a continuación se analizan las posibles soluciones cinemáticas del sistema, siendo los mismos el problema cinemático directo e inverso.

Inicialmente, en el problema cinemático directo se define la inclinación del carril pivotante con respecto a un sistema de coordenadas de referencia, partiendo del ángulo del servomotor. Por otro lado, el problema cinemático inverso consiste en, como bien su nombre indica, justamente lo contrario, a partir de la inclinación del carril pivotante, cuantificar el ángulo que deben alcanzar los actuadores. En este caso se opta por el problema cinemático inverso, ya que será bastante provechoso conocer qué ángulo debe lograr el servomotor para que el carril pivotante se mantenga en la inclinación deseada.

De tal modo, se prosiguen los trabajos desarrollando dicho problema utilizando matrices de transformación homogénea (MTH) o por el método geométrico, indicado en el siguiente esquema, tratándose del método elegido dada su mayor facilidad de implementación.

El método geométrico elegido trata de averiguar las relaciones existentes que permiten hallar la situación espacial del sistema a partir de sus coordenadas articulares.

$$\begin{aligned}
 D1 &= 0 \text{ [m]} \\
 D2 &= 0.136 \text{ [m]} \\
 D3 &= 0.075 \text{ [m]} \\
 R1 &= 0.050 \text{ [m]} \\
 R2 &= 0.1383 \text{ [m]} \\
 \beta_{min} &= -30 \text{ [deg]} \\
 \beta_{max} &= 30 \text{ [deg]}
 \end{aligned}$$

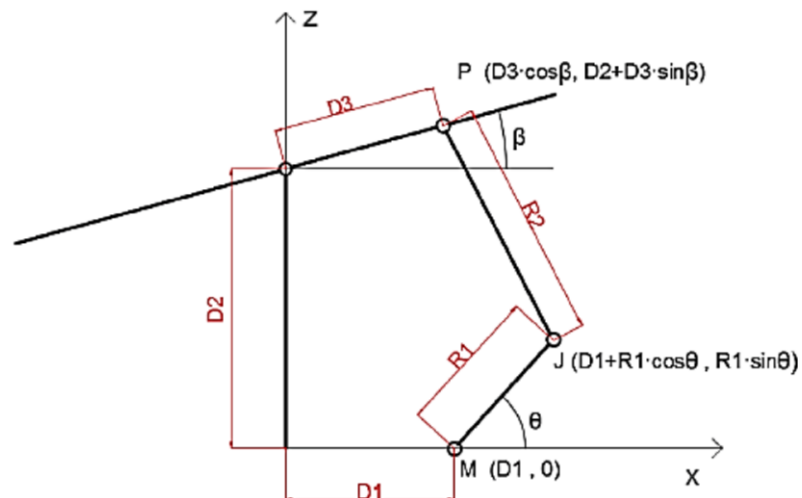


Ilustración 8: Relaciones trigonométricas

El ángulo β es conocido y aporta la solución de las coordenadas del punto de unión P de la barra de conexión con el carril pivotante. Por otro lado, es relevante destacar que las coordenadas buscadas son las de la unión J, para poder determinar el ángulo de rotación θ del servomotor.

El objetivo principal es hallar las coordenadas de la anteriormente citada unión J, como la intersección de la trayectoria realizada por el balancín del servomotor R1 y por la barra de conexión R2, considerándose estas dos circunferencias de radio R1 con centro en M, y otra circunferencia de radio R2 y centro P respectivamente.

Considerando que los puntos M y P representan los centros de las trayectorias circulares anteriormente expuestas, es posible plantear el siguiente sistema de ecuaciones:

$$\begin{aligned}
 (x - x_M)^2 + (z - 0)^2 &= R1^2 \\
 (x - x_P)^2 + (z - z_P)^2 &= R2^2
 \end{aligned}$$

Desarrollando y expandiendo las ecuaciones anteriores, se obtiene:

$$\begin{aligned}
 x^2 - 2 \cdot x \cdot x_M + x_M^2 + z^2 &= R1^2 \\
 x^2 - 2 \cdot x \cdot x_P + x_P^2 + z^2 - 2 \cdot z \cdot z_P + z_P^2 &= R2^2
 \end{aligned}$$

A continuación, si se realiza el resto de las ecuaciones anteriores:

$$z = A + B \cdot x$$

$$A = \frac{R1^2 - R2^2 - x_M^2 + x_P^2 + z_P^2}{2 \cdot z_P}$$

$$B = \frac{x_M - x_P}{z_P}$$

Sustituyendo estos valores en las ecuaciones previas:

$$x^2 - 2 \cdot x \cdot x_M + x_M^2 + A^2 - 2 \cdot A \cdot B + B^2 \cdot x^2 = R1^2$$

$$x^2 \cdot (1 + B^2) + x \cdot (2 \cdot A \cdot B - 2 \cdot x_M) + (A^2 + x_M^2 - R1^2) = 0$$

Finalmente, resolviendo la ecuación de segundo grado se halla la posición del enlace J, y en consecuencia, se cuantifica el valor del ángulo θ . Por otro lado, se sustituyen las coordenadas de los puntos M y P.

$$A = \frac{R1^2 - R2^2 - D1^2 + (D3 \cdot \cos(\beta))^2 + (D2 + D3 \cdot \sin(\beta))^2}{2 \cdot (D2 + D3 \cdot \sin(\beta))}$$

$$B = \frac{D1 - D3 \cdot \cos(\beta)}{D2 + D3 \cdot \sin(\beta)}$$

$$x^2 \cdot (1 + B^2) + x \cdot (2 \cdot A \cdot B - 2 \cdot D1) + (A^2 + D1^2 - R1^2) = 0$$

Se representan gráficamente las dos soluciones que se obtienen de la ecuación de segundo grado, y se observa que en cada solución hay un rango de valores que se corresponden con grados deseamos que alcance el balancín del servomotor.

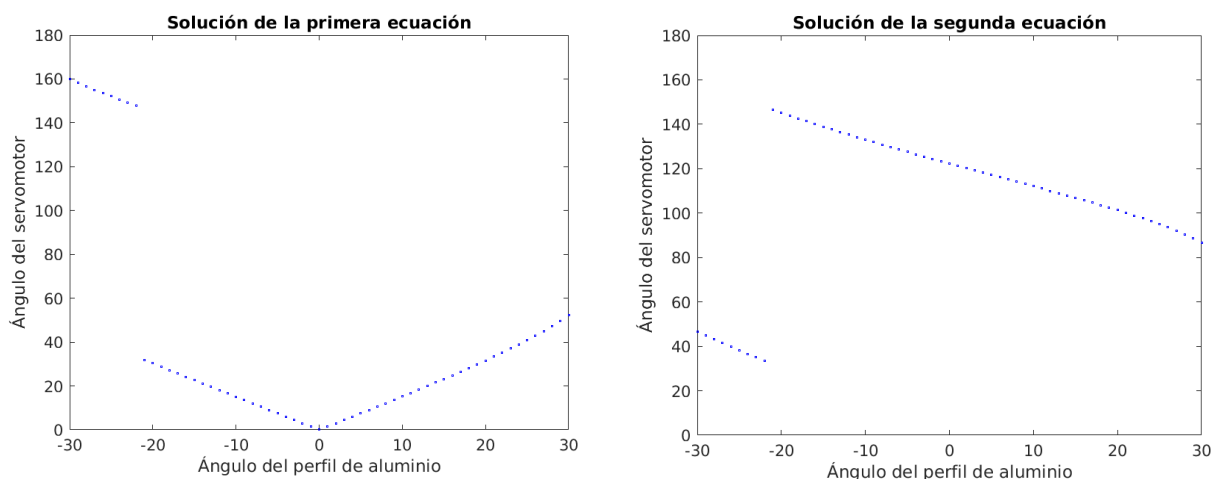


Ilustración 9: Izquierda: 1ª solución ecuación 2º grado; Derecha: 2ª solución ecuación 2º grado

En la representación de la primera solución de la ecuación aceptamos el intervalo de $[-21, 30]$ grados, y en la representación de la segunda solución de la ecuación aceptamos el intervalo $[-30, -20]$ grados.

Representando en una gráfica los intervalos anteriores, se obtiene el siguiente resultado:

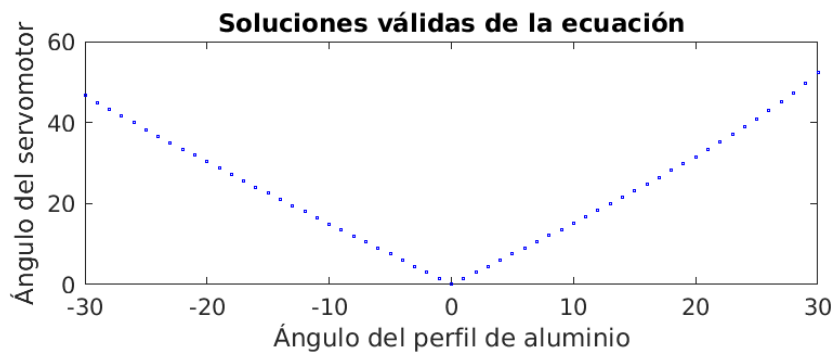


Ilustración 10: Relación del ángulo del servomotor y perfil de aluminio

Se resuelve que para un ángulo máximo de rotación del perfil de aluminio de -30 grados la rotación del balancín del servomotor es de 46.66 grados y para un ángulo máximo de rotación del perfil de aluminio de 30 grados la rotación del balancín del servomotor es de 52.24 grados. Se realizan dos bocetos para verificar los resultados:

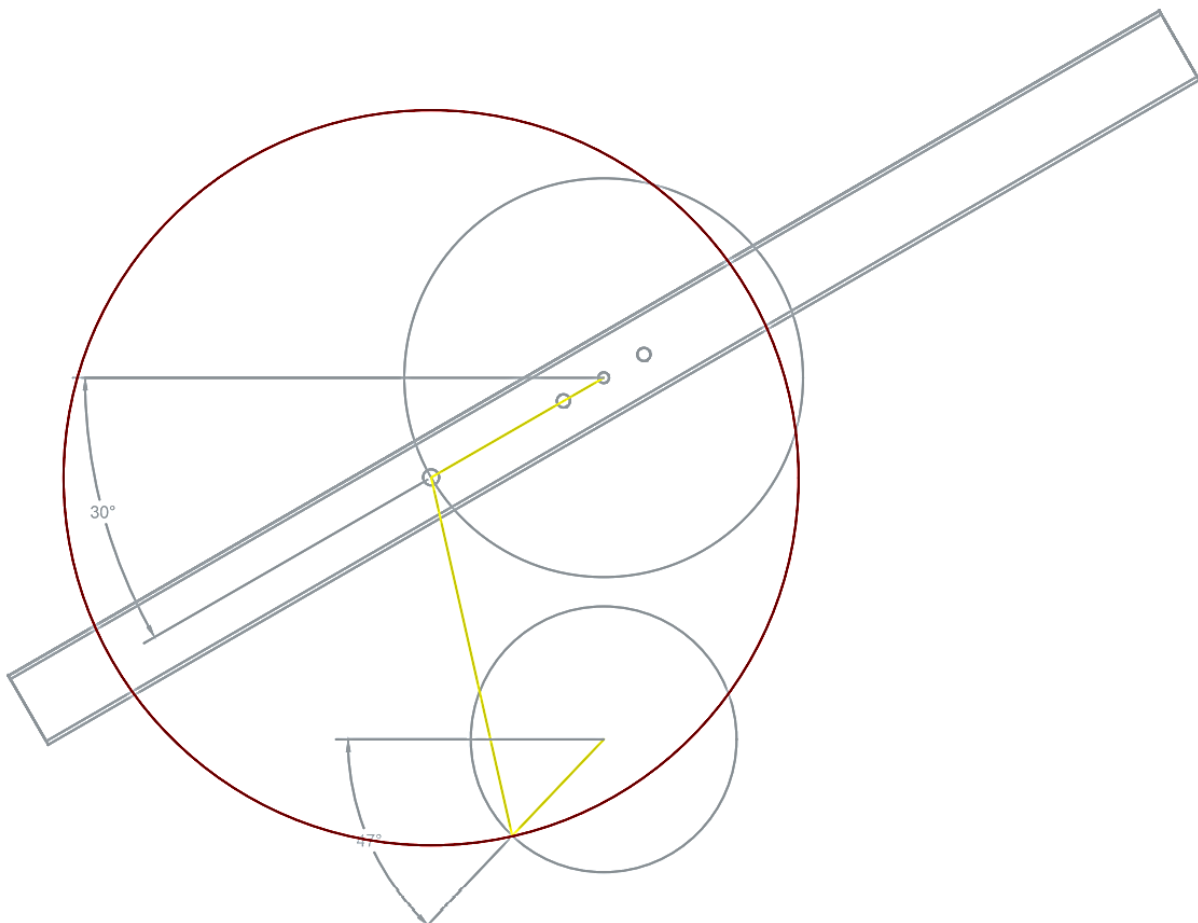


Ilustración 11: Cinemática del mecanismo (2)

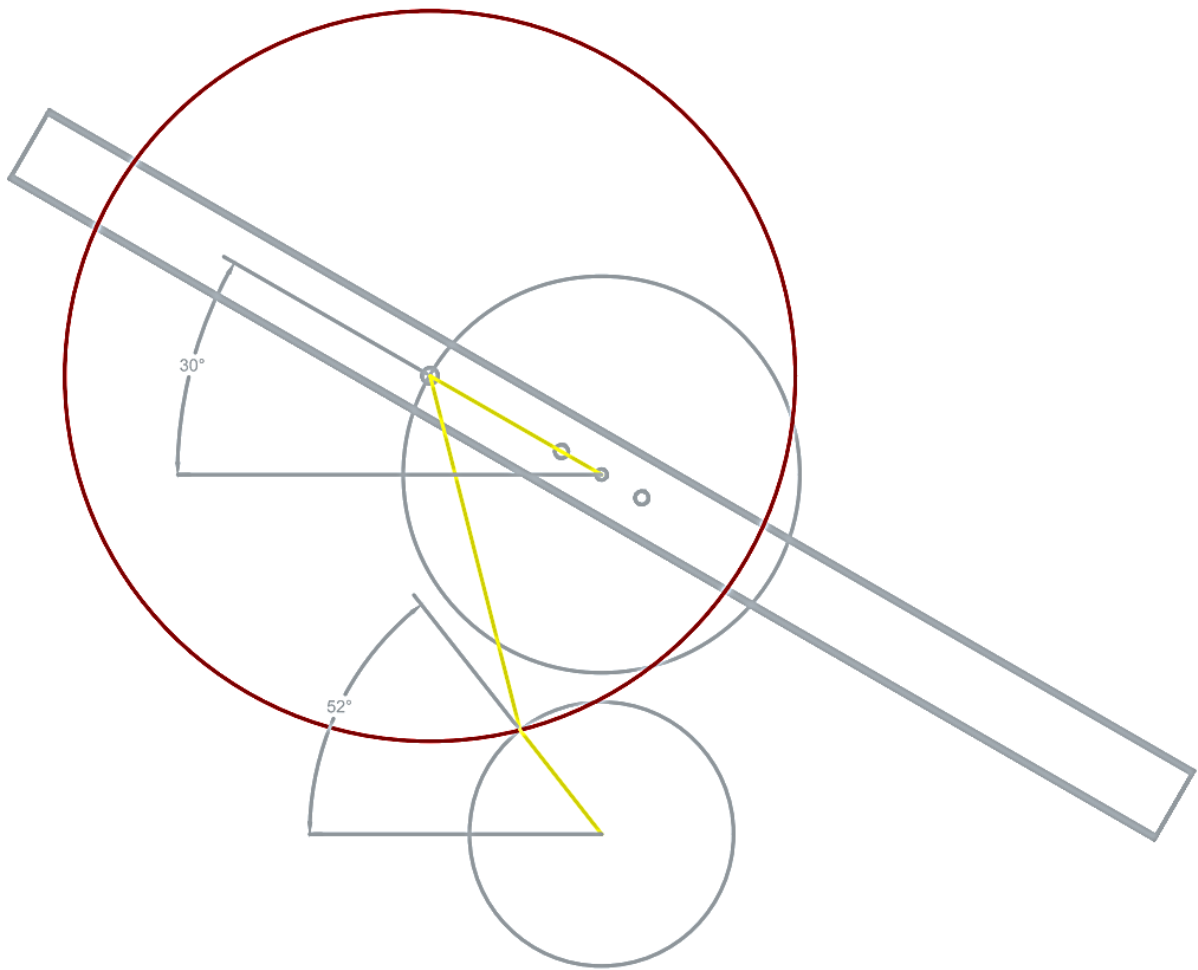


Ilustración 12: Cinemática del mecanismo (3)

Elementos estructurales

Para reducir el tiempo y coste de fabricación, minimizar las limitaciones de diseño y facilitar la integración de elementos comerciales, se ha elegido la opción de fabricación 3D para todos los elementos estructurales.

El carril pivotante debe sujetarse desde la parte posterior, minimizando los elementos de conexión con las partes estructurales fijas. Esto implica el diseño de un sistema de sujeción formado por 3 piezas, un vástago con un registro que va atornillado al carril, un collarín separador y un casquillo que hace de tope atornillado al vástago. Este conjunto permite la rotación del carril gracias a 2 rodamientos. De este modo el carril queda inmóvil en cualquier dirección que no sea la de rotación.

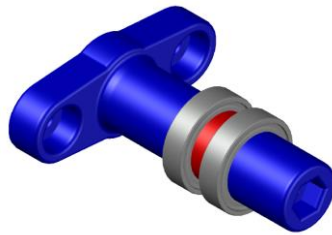


Ilustración 13: Conjunto eje/collarín/casquillo/rodamientos

La columna principal está formada por 2 piezas. Una pieza que consiste en 3 apoyos que se atornillan a la base y se unen en la parte superior, donde hay unas cavidades cilíndricas para insertar el eje y los rodamientos. La otra pieza que es una tapa con las mismas cavidades cilíndricas, la tapa encierra el conjunto del eje mediante unos tornillos. La altura de la columna proporciona una altura de 190 milímetros al eje del carril pivotante.

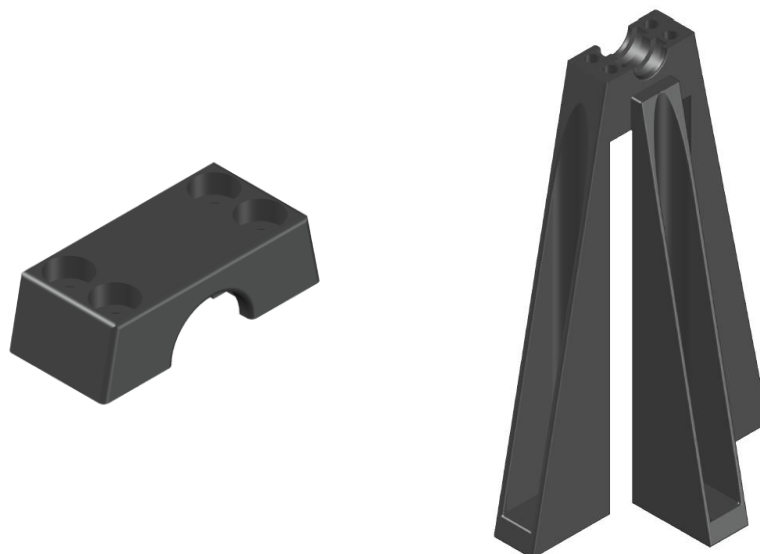


Ilustración 14: Izquierda: Tapa columna; Derecha: Columna; Derecha

El servomotor necesita un soporte de sujeción con la base, se ha diseñado manteniendo estilo de diseño de la columna principal y proporciona una altura de 54 milímetros en el eje de rotación.

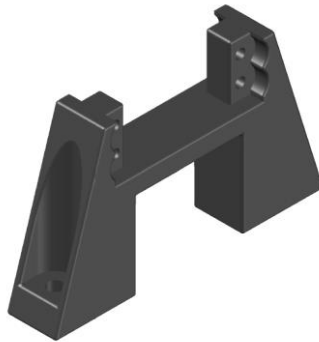


Ilustración 15: Soporte del servomotor

El servomotor necesita un balancín para aumentar el brazo de palanca tal y como se ha mostrado anteriormente en el boceto del mecanismo. El balancín contiene un alojamiento para poder contrapesar el mecanismo y una serie de orificios para adaptarse a múltiples tipos de acople del servomotor.



Ilustración 16: Balancín servomotor

Es necesaria la construcción de una barra de conexión para transmitir el movimiento de rotación del servomotor al carril pivotante. El material empleado para la construcción es comercial y consta de dos articulaciones de plástico "IGUS" de métrica 6 con cojinetes esféricos y una varilla roscada de métrica 6 para ajustar la longitud a la mostrada en el boceto del mecanismo.



Ilustración 17: Barra de conexión

Los elementos estructurales diseñados requieren una base para ser instalados, teniendo en cuenta la disposición de los orificios de los tornillos, las premisas de accesibilidad, usabilidad y ergonomía consideradas en los objetivos y las limitaciones de tamaño de impresión de la impresora 3D utilizada, la mejor opción resulta ser la siguiente:

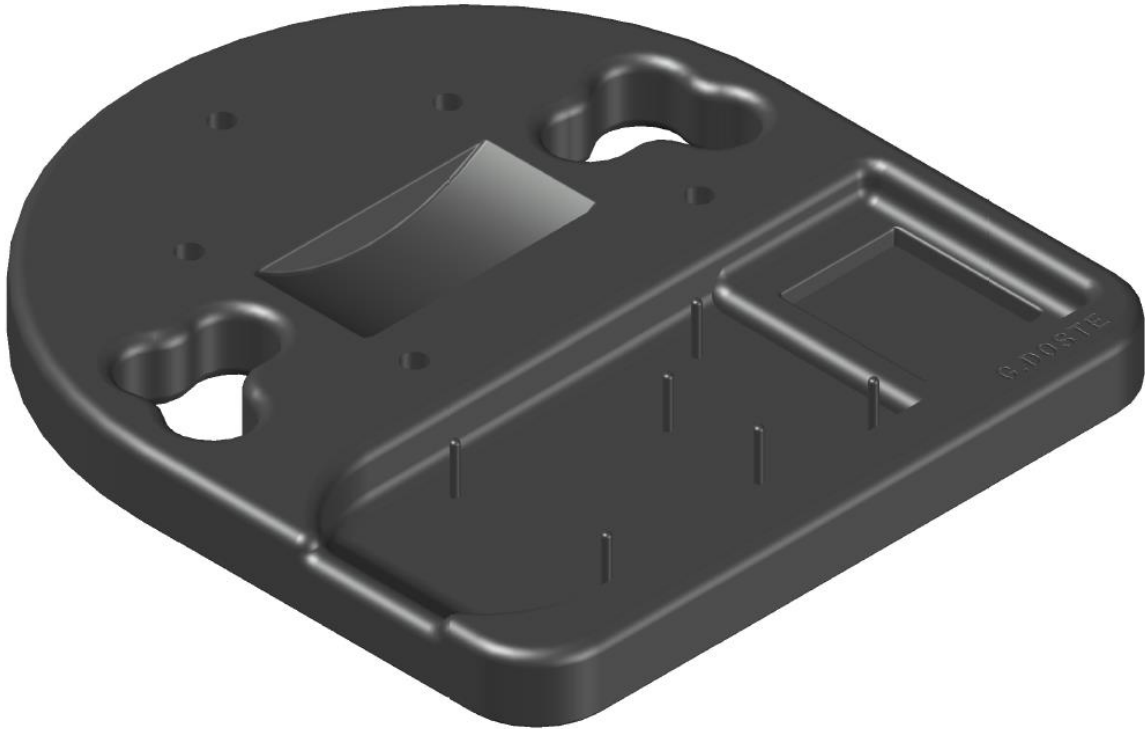


Ilustración 18: Base

Se finaliza esta parte con el diseño de los soportes que se utilizan para unir los sensores de proximidad con el carril pivotante. El sistema consta de dos soportes para la sujeción de los sensores, y permitan adaptar su altura para mejorar la detección del cuerpo esférico. Ambos soportes disponen de una base con una ranura profunda que permite encajar a presión el soporte en el interior del carril, adicionalmente se pueden atornillar para mejorar la sujeción.

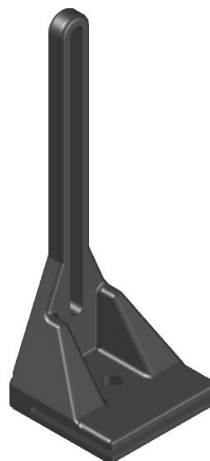


Ilustración 19: Soporte sensor

En la siguiente ilustración se observa el resultado del diseño mecánico del prototipo, una vez que se han montado todos los componentes e instalado los sistemas de control y alimentación:

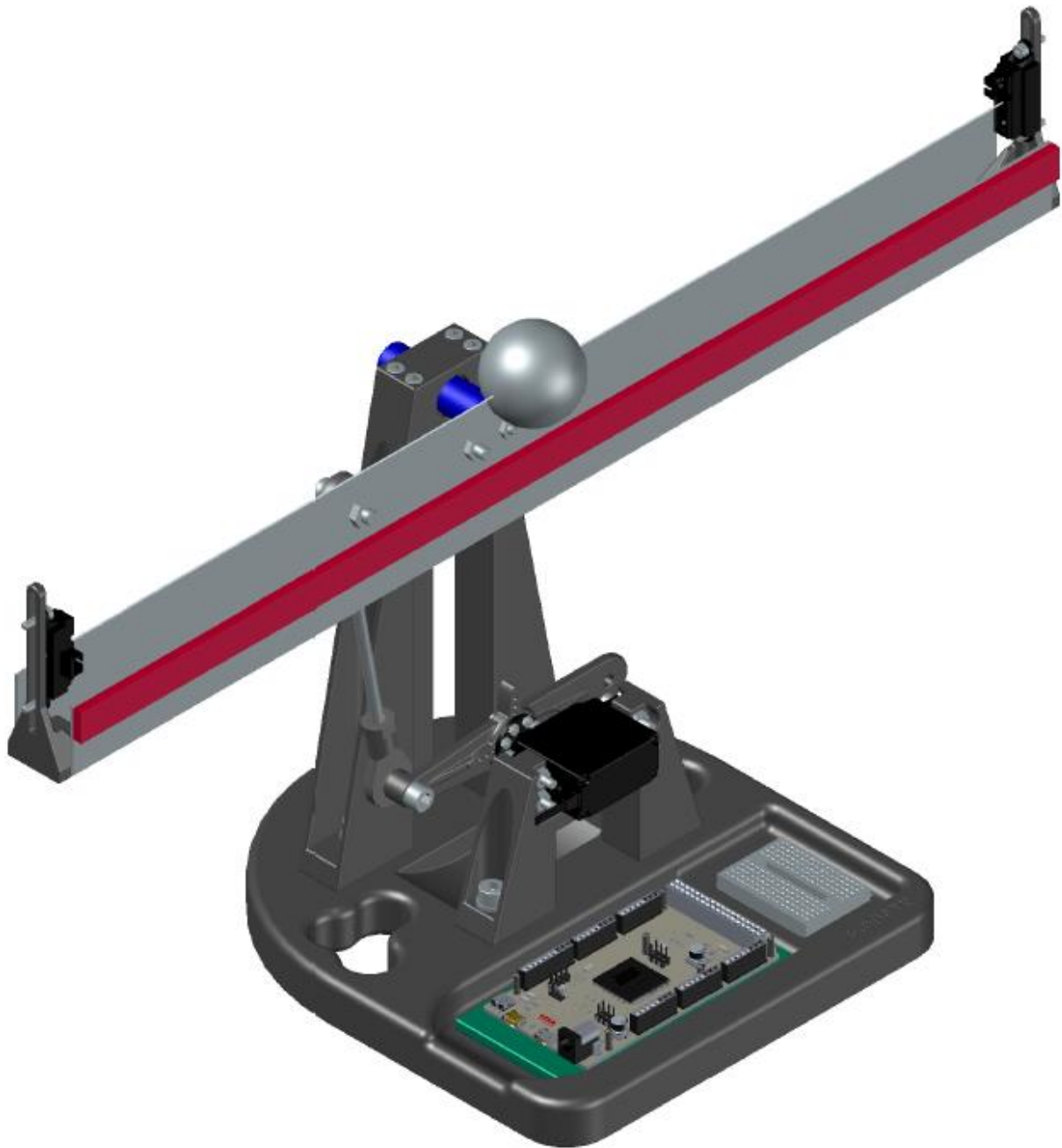


Ilustración 20: Prototipo

Elementos electrónicos

Diseño conceptual

El diseño de este sistema está destinado al uso docente, en consecuencia, los elementos electrónicos que se integran en el diseño mecánico deben ser de fácil acceso y bajo costo.

Nuestro sistema control consta de una placa de evaluación cuyo firmware realiza tareas como el control de la posición del servomotor, la iluminación de la tira de LED, el análisis de la señal analógica de los sensores de proximidad y la comunicación con la interfaz gráfica del PC.

Partiendo de materiales utilizados regularmente por estudiantes y añadiendo componentes sencillos (“mini breadboard”, “jumper wires”, condensadores, una fuente...) disponibles en el mercado se ha elaborado un conjunto electrónico que se corresponde con el siguiente diagrama de funcionamiento:

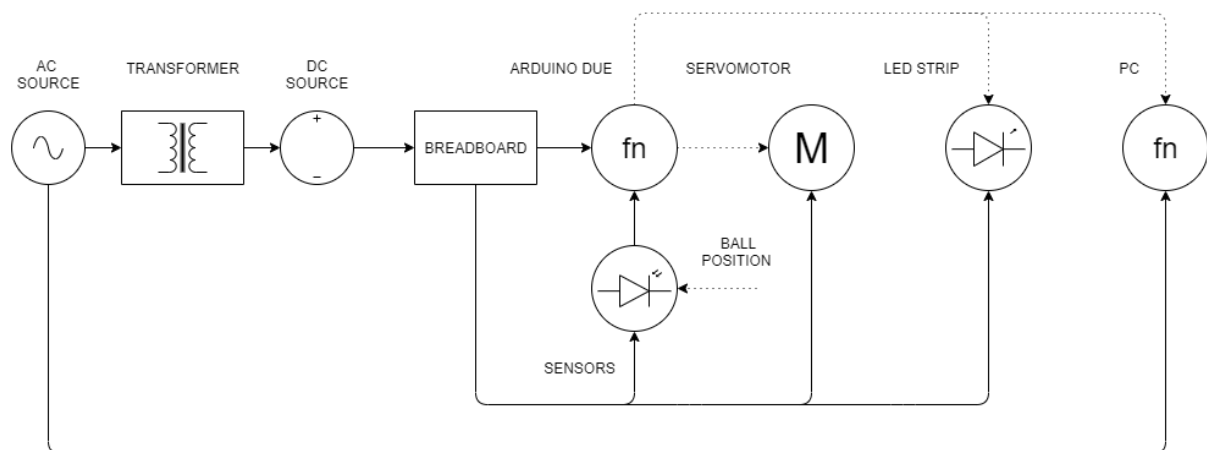


Ilustración 21: Diagrama de funcionamiento de la electrónica

El panel de conexiones “mini breadboard” distribuye la alimentación y masa a los componentes electrónicos de corriente continua. Este panel combinado con el uso de “jumper wires” resulta muy útil porque las conexiones se realizan de una forma más intuitiva y segura, además, permite insertar condensadores tradicionales para filtrar las señales analógicas.

Arduino DUE

La comunicación entre el prototipo y la interfaz gráfica se tiene que realizar mediante una herramienta que nos permita controlar los parámetros y observar los resultados a tiempo real. Arduino cumple estos requisitos, es una plataforma abierta que facilita la programación de un microcontrolador y una de las más utilizadas en el área de desarrollo de prototipos para la robótica.

Esta plataforma tiene como objetivo acercar y facilitar el uso de la electrónica y programación de sistemas embebidos en proyectos multidisciplinarios y se ha expandido rápidamente gracias a su plataforma hardware con diseño abierto, su entorno desarrollado de código abierto, su lenguaje fácil de aprender basado en librerías, un entorno de desarrollo integrado (IDE) también con código abierto y que está disponible y se puede trabajar desde Windows, Mac y Linux. Los proyectos hechos con Arduino pueden ejecutarse sin necesidad de estar conectados a un ordenador, ya que una vez cargado el código con sus respectivas librerías en la placa ya solo será necesario conectar el dispositivo a la placa.



Ilustración 22: Placa de evaluación "Arduino Due"

Características:

- Microcontrolador: AT91SAM3X8E
- Voltaje de operación: 3.3V
- Voltaje de entrada recomendado: 7-12 V
- Voltaje límite de entrada: 6-16 V
- I/O digitales: 54 (12 salidas PWM)
- Entradas analógicas: 12
- Salidas analógicas: 2
- Corriente continua I/O digitales: 130 mA
- Corriente continua 3.3V Pin: 800 mA
- Corriente continua 5V Pin: 800 mA
- Memoria flash: 512 kB
- SRAM: 96 kB
- Velocidad del reloj: 84 MHz
- Peso: 36g

Servomotor

El servomotor estándar Parallax es ideal para la creación de prototipos de brazos de robot y aplicaciones animatrónicas. Puede mantener cualquier posición en un rango de 0 a 180 grados y se comunica fácilmente con cualquier microcontrolador capaz de producir una salida PWM de 50 Hz.

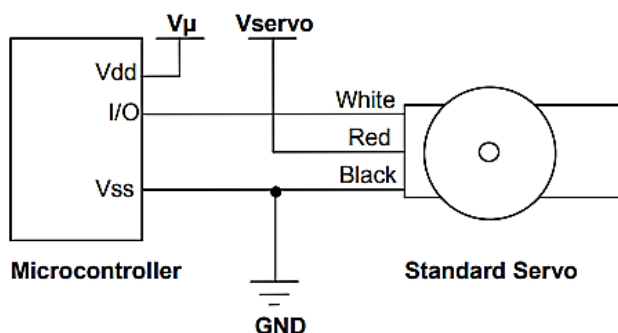


Ilustración 23: Servomotor Parallax

Características:

- Engranajes de alta precisión fabricados de resina de poliacetal para un funcionamiento uniforme sin holgura
- Par: 0.268 Nm a +6 V
- Montaje de servo R/C (hobby) estándar y conector de 3 contactos
- Fuente de alimentación: +4 V a +6 V DC, 15 a 200 mA
- Rango de temperaturas de funcionamiento: -10 a +50 °C
- Dimensiones: 55,8 x 19 x 40,6 mm excluida bocina del servo
- Peso: 44 g

Conexión:

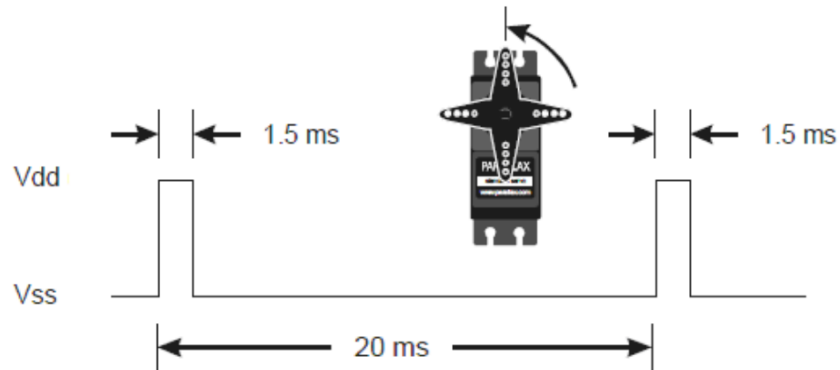


Pin	Name	Description	Minimum	Typical	Maximum	Units
1 (White)	Signal	Input; TTL or CMOS	3.3	5.0	$V_{servo} + 0.2$	V
2 (Red)	V_{servo}	Power Supply	4.0	5.0	6.0	V
3 (Black)	V_{ss}	Ground		0		V

Ilustración 24: Conexión del servomotor

Protocolo de comunicación:

El servo estándar Parallax se controla mediante modulación de ancho de pulso, donde la posición del eje del servo depende de la duración del pulso. Para mantener su posición, el servo necesita recibir un pulso cada 20 ms. A continuación, se muestra un diagrama de tiempo de muestra para la posición central del Parallax Servo estándar.



BASIC Stamp Module	0.75 ms	1.5 ms (center)	2.25 ms
--------------------	---------	-----------------	---------

Ilustración 25: Control de la posición del servomotor

Sensor de proximidad

GP2Y0A41SK0F es una unidad de sensor de medición de distancia con un rango de medición de 4 a 30 cm, el sensor de medición incluye un PSD (detector sensible a la posición) y un IR-LED (diodo emisor de infrarrojos) y un circuito de procesamiento de señales.



Ilustración 26: Sensor de proximidad Sharp

Este tipo de sensores están compuestos por un LED infrarrojo junto al dispositivo detector de posición (PSD) y un procesador integrado encargado de realizar el cálculo de la distancia. El sensor barre el entorno y escanea en tiempo real los objetos situados enfrente suyo y proporciona una salida en forma de un rango de tensión analógica. El emisor LED produce un haz de luz infrarroja con una determinada longitud de onda. Esta luz es pulsada para aminorar la influencia de las características ambientales y cromáticas del objeto en cuestión. El detector de posición PSD (Position Sensitive Detection) es en realidad un sensor de pequeñas dimensiones CCD (Charge Coupled Device) lineal que recibe la luz reflejada en cualquier obstáculo en dirección de la trayectoria del rayo. El detector emplea triangulación para calcular la longitud del sensor al objeto situado en frente de él.

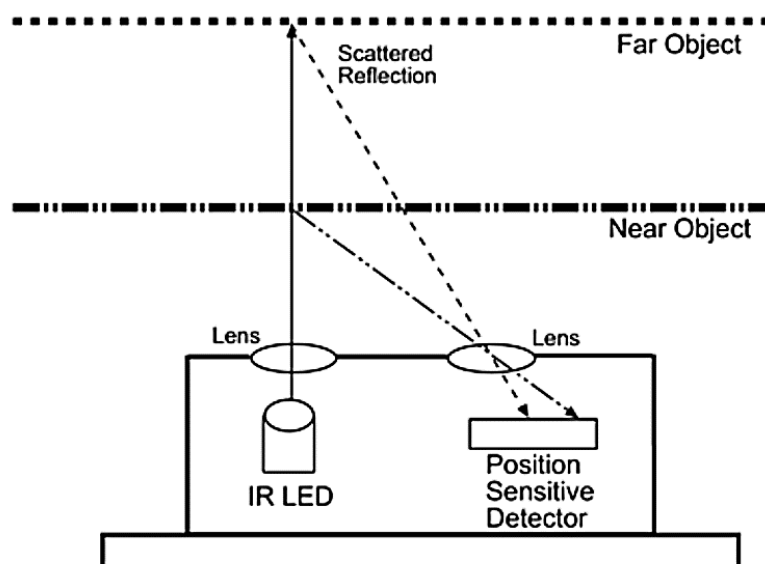


Ilustración 27: Funcionamiento del sensor

Características:

- Tipo de salida analógica
- Rango de medida de distancia de 4cm a 30cm
- La tensión del terminal de salida es de 0,4V
- La tensión del terminal de entrada es de 4V a 6V

Esquemático:

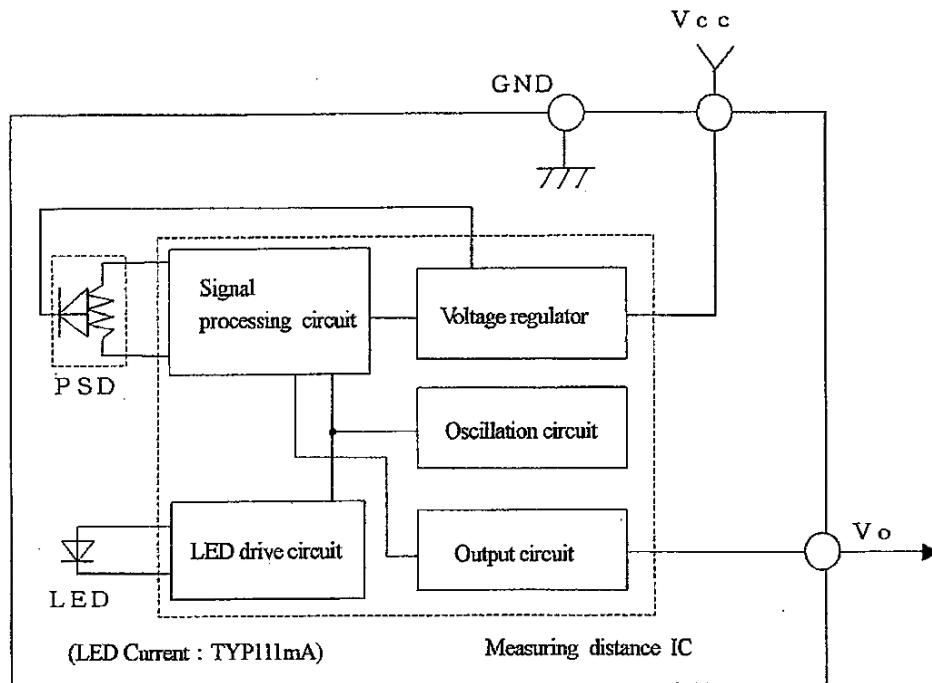


Ilustración 28: Esquema electrónico del sensor

Tiempo de respuesta:

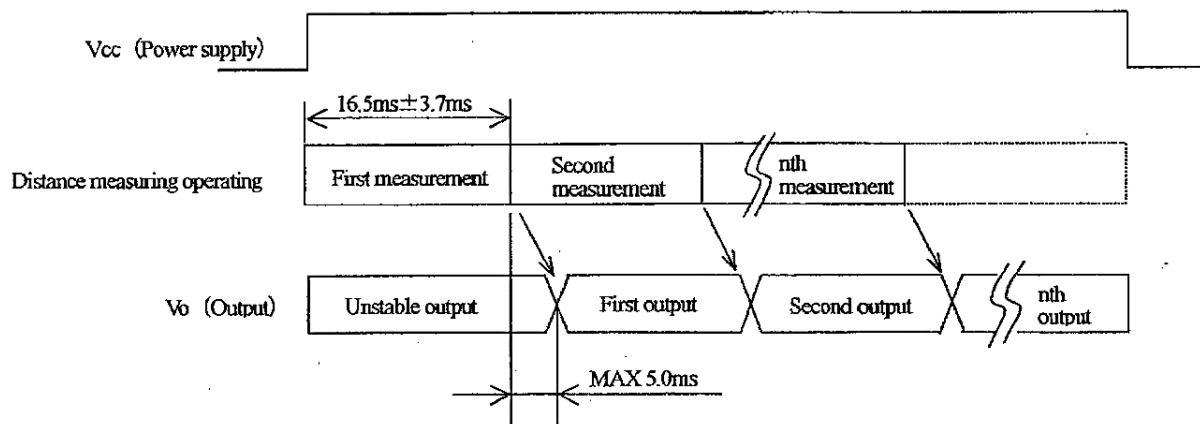


Ilustración 29: Tiempo de respuesta del sensor

Tira de LED

La tira de LED SMD 5050 de "DotStar" consta de un controlador integrado en el interior de cada LED. Las tiras están fabricadas en material de PCB flexible y se suministran con un revestimiento resistente a la intemperie. Se pueden cortar a cualquier tamaño respetando las líneas de corte visibles o conectarse entre sí para crear tiras más largas.



Ilustración 30: Tira de LED "Adafruit Dotstar 5050"

Características:

Para controlar los LED, se utiliza un SPI de 2 cables genérico. El color o brillo de cada LED se establece usando color de 24 bits (8 bits cada uno para rojo, verde y azul). Cada LED actúa como un registrador de desplazamiento, leyendo los datos de color entrantes en los contactos de entrada y, a continuación, desplazando el dato del color anterior en el conector de salida. Por tanto, enviando una cadena de datos larga se puede controlar muchos LED.

La señal PWM está integrada en cada chip del LED y tienen tasas de actualización altas. En aplicaciones de persistencia de visión (POV), tiene menos parpadeo, sobre todo en bajos niveles de brillo. Los "DotStars" tienen una tasa de PWM de 20 kHz, por lo que incluso agitando rápidamente la tira, no se puede ver el pixelado y la combinación de color es uniforme.

Cada tira se suministra con un conector SM JST de 4 pines en un extremo. Estos LED tienen 2 cables (entrada y salida) para enviar datos: una clavija de reloj y una clavija de datos. El reloj y los datos están separados, por lo que se puede utilizar con cualquier velocidad del procesador para controlar estas tiras. La compatibilidad con SPI de hardware es útil pero no es necesaria.

Dimensiones:

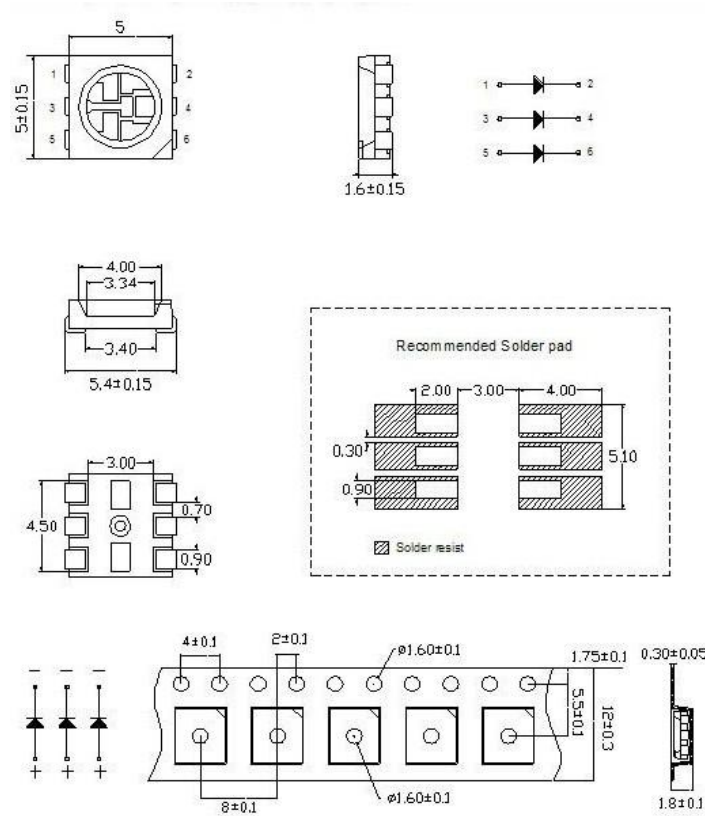


Ilustración 31: Dimensiones del chip LED y tira

Esquemático:

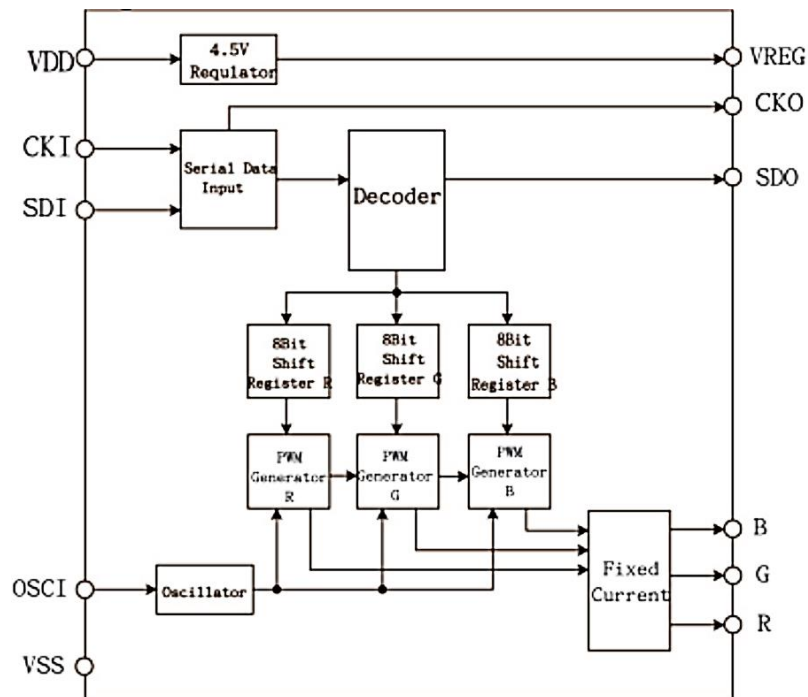


Ilustración 32: Esquema electrónico del sensor

Fungible

- “Mini breadboard” es una placa pequeña de prototipado (4,6 x 3,6 cm), con 170 puntos de conexión. Viene con un adhesivo y es compatible con el “Arduino Proto Shield”.



Ilustración 33: “Mini breadboard”

- “Jumper wires” es una banda de cables utilizada normalmente para realizar puentes en prototipos, cada cable tiene un conector en cada punta, que se usa normalmente para interconectar entre sí los componentes en una placa de evaluación.



Ilustración 34: “Jumper wires”

- Capacitador electrolítico de aluminio de 1 microfaradio a 16 voltios. Rango de temperatura de +85 a -40° Celsius de tolerancia de $\pm 20\%$ voltios. Utilizado para el filtrado de señales analógicas (imagen solo con fines ilustrativos).



Ilustración 35: Condensador electrolítico

- Conectores JST: Conector JST GH 1.25 de 3 pines Macho/Hembra



Ilustración 36: Conector JST con cables pre-crimpados

- Cables pre-crimpados



Ilustración 37: Cable pre-crimpado

- Ficha de empalmes



Ilustración 38: Ficha de empalmes

Sistema de control

Controlador PID

Actualmente se conocen muchas estrategias de sistemas de control, en nuestro caso se ha establecido como objetivo utilizar el controlador más común, el PID, para establecer los conceptos básicos.

Un controlador PID es un mecanismo de control de retroalimentación ampliamente utilizado en sistemas de control industrial. Éste calcula la desviación o el error entre un valor de referencia y un valor de salida del sistema. El algoritmo de control consta de tres parámetros diferentes: el proporcional, el integrador y el derivativo. El valor proporcional depende del error actual, la integral depende de los errores pasados y la derivada es una predicción de errores futuros. Un problema común en este tipo de controlador es la alta sensibilidad al ruido en la parte derivada, el ruido puede afectar el sistema, haciendo que esas pequeñas variaciones hagan que el cambio en la salida del sistema sea muy grande.

La función de transferencia típica del controlador PID en el plano "s" es la siguiente:

$$\frac{U_o(s)}{E(s)} = PID(s) = K_p + \frac{K_i}{s} + K_d \cdot s$$

El objetivo principal de este proyecto consiste en obtener un sistema de control discreto PID, para obtener la función de transferencia discreta es necesario transformar la acción integradora y derivativa del plano "s" al plano "z", para ello se realizan las siguientes transformaciones:

- Transformación de la acción integradora utilizando el método de integración trapezoidal:

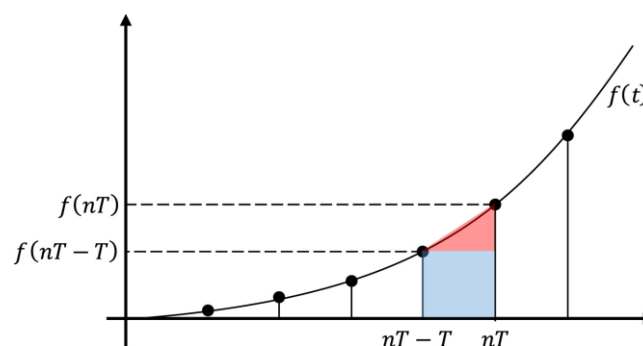


Ilustración 37: Integración trapezoidal

$$A(nT) = T \cdot f(nT - T) + \frac{T \cdot (f(nT) - f(nT - T))}{2}$$

$$y(nT) = y(nT - T) + A(nT)$$

$$Y(z) = Y(z) \cdot z^{-1} + \frac{T}{2} \cdot (F(z) \cdot z^{-1} + F(z))$$

$$\frac{Y(z)}{F(z)} = \frac{T \cdot (1 + z^{-1})}{2 \cdot (1 - z^{-1})} = \frac{T \cdot (z + 1)}{2 \cdot (z - 1)} \cong \frac{1}{s}$$

- Transformación de la acción derivativa utilizando el método “Backwards Euler”:

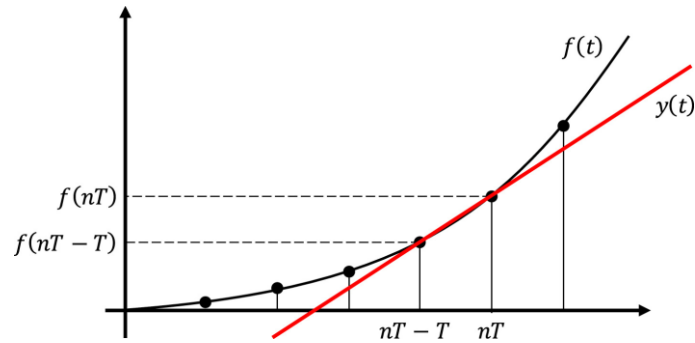


Ilustración 38: Derivación “Backwards Euler”

$$a(nT) = \frac{f(nT) - f(nT - T)}{T}$$

$$A(z) \cdot T = F(z) - F(z) \cdot z^{-1}$$

$$\frac{A(z)}{F(z)} = \frac{(1 - z^{-1})}{T} = \frac{(z - 1)}{z \cdot T} \cong s$$

Realizando estas transformaciones sobre la función de transferencia PID(s) se obtiene la siguiente función de transferencia discreta:

$$PID(z) = \frac{a \cdot z^2 + b \cdot z + c}{z \cdot (z - 1)}$$

$$a = K_p + \frac{K_i \cdot T_s}{2} + \frac{K_d}{T_s}$$

$$b = -K_p + \frac{K_i \cdot T_s}{2} - \frac{2 \cdot K_d}{T_s}$$

$$c = \frac{K_d}{T_s}$$

Recordando que la entrada de la función de transferencia del controlador es el error del sistema y la salida es el valor que recibe el actuador, se transforma la expresión en “z” negativas:

$$\frac{U_o(z)}{E(z)} = \frac{a + b \cdot z^{-1} + c \cdot z^{-2}}{1 - z^{-1}}$$

Realizando el producto cruzado de la expresión anterior:

$$U_o(z) = a \cdot E(z) + b \cdot E(z) \cdot z^{-1} + c \cdot E(z) \cdot z^{-2} + U_o(z) \cdot z^{-1}$$

Finalmente aplicando la transformada inversa “z” se obtiene la siguiente ecuación en diferencias:

$$u_o(k) = a \cdot e(k) + b \cdot e(k - 1) + c \cdot e(k - 2) + u_o(k - 1)$$

Anti derivative kick

La acción de la derivada tiene dos inconvenientes principales, amplifica el ruido de alta frecuencia en la señal de error y, un cambio brusco en la señal de error genera por un instante una salida del controlador muy elevada conocida como "derivative kick". Para resolver la patada de la derivada, se supone que la señal de entrada es constante (su derivada es cero), luego, la derivada solo se aplica a la señal de salida tal y como se muestra en el siguiente diagrama de bloques de un controlador de acción derivativa.

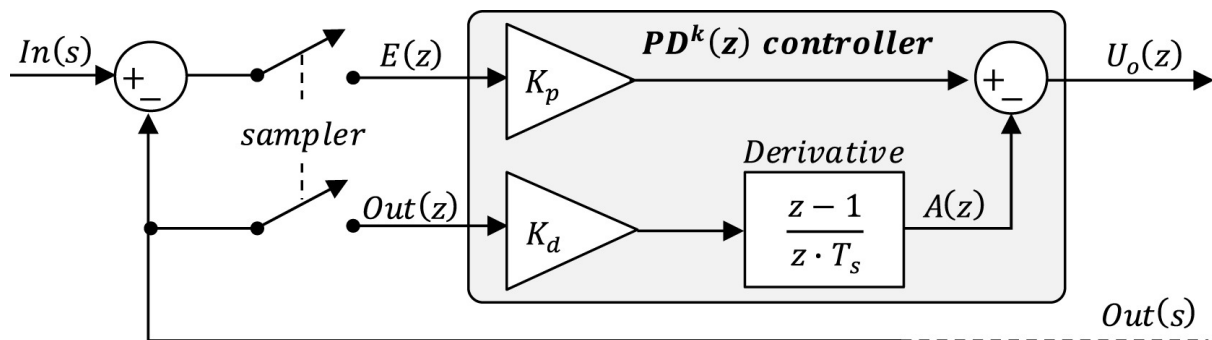


Ilustración 39: Diagrama de bloques "Anti derivative kick"

En nuestro caso al utilizar un controlador PID, la obtención de la ecuación en diferencias parte de la siguiente ecuación del diagrama de bloques que le corresponde:

$$U_o(z) = U_o^1(z) - U_o^2(z) = PI(z) \cdot E(z) - K_d \cdot D(z) \cdot OUT(z)$$

$$D(z) = \frac{z - 1}{z \cdot T_s}$$

Cuando se aplica el "anti derivative kick" la señal de control tiene dos términos, y la ecuación en diferencias de un controlador PID que debe implementarse en el microcontrolador es la siguiente:

$$u_o^1(k) = a \cdot e(k) + b \cdot e(k - 1) + u_o^1(k - 1)$$

$$u_o^2(k) = a' \cdot (out(k) - out(k - 1))$$

$$u_o(k) = u_o^1(k) - u_o^2(k)$$

$$a' = \frac{K_d}{T_s}$$

Derivative filter

El ruido se puede filtrar agregando un polo de primer orden en la acción derivativa a la frecuencia deseada "N". Asumiendo que la señal de "feedback" es solo la salida del sistema, la frecuencia debería ajustarse a la frecuencia natural del modelo matemático de la planta.

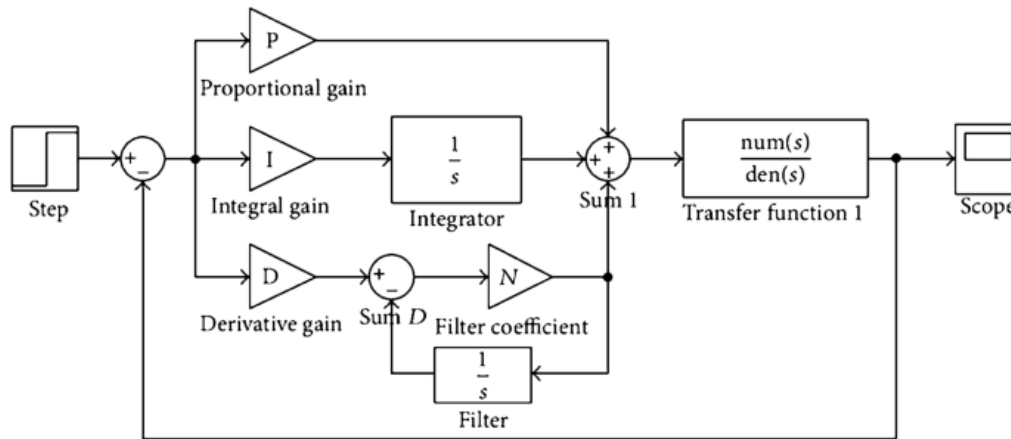


Ilustración 40: Diagrama de bloques PID con "Derivative filter"

La función de transferencia típica del controlador PID tras añadir un polo de primer orden en la acción derivativa en el plano "s" es la siguiente:

$$PID^f(s) = K_p + \frac{K_i}{s} + K_d \cdot \frac{s \cdot N}{s + N}$$

Realizando las transformaciones mostradas en el PID(s) convencional sobre la nueva función de transferencia PID(s) con filtro en la acción derivativa se obtiene la siguiente función de transferencia discreta:

$$PID^f(z) = \frac{a \cdot z^2 + b \cdot z + c}{z^2 + d \cdot z + f}$$

$$a = \frac{2 \cdot K_p \cdot (1 + N \cdot T_s) + K_i \cdot T_s \cdot (1 + N \cdot T_s) + 2 \cdot K_d \cdot N}{2 \cdot N \cdot T_s + 1}$$

$$b = \frac{-2 \cdot K_p \cdot (2 + N \cdot T_s) + K_i \cdot N \cdot T_s^2 - 4 \cdot K_d \cdot N}{2 \cdot N \cdot T_s + 1}$$

$$c = \frac{2 \cdot K_p - K_i \cdot T_s + 2 \cdot K_d \cdot N}{2 \cdot N \cdot T_s + 1}$$

$$d = \frac{-(2 + N \cdot T_s)}{N \cdot T_s + 1}$$

$$f = \frac{1}{N \cdot T_s + 1}$$

Repitiendo el procedimiento realizado en el controlador PID convencional se obtiene la función de transferencia en “z” negativas:

$$\frac{U_o(z)}{E(z)} = \frac{a + b \cdot z^{-1} + c \cdot z^{-2}}{1 + d \cdot z^{-1} + f \cdot z^{-2}}$$

Finalmente realizando el producto cruzado y la transformada “z” inversa se obtiene la siguiente ecuación en diferencias:

$$u_o(k) = a \cdot e(k) + b \cdot e(k - 1) + c \cdot e(k - 2) - d \cdot u_o(k - 1) - f \cdot u_o(k - 2)$$

Hacer uso combinado del filtro derivativo y del “anti derivative kick” conlleva una serie de cambios en la ecuación en diferencias. Partiendo del ejemplo del diagrama de bloques con “anti derivative kick” se obtiene la siguiente ecuación del sistema:

$$U_o(z) = U_o^1(z) - U_o^2(z) = PI(z) \cdot E(z) - K_d \cdot D^f(z) \cdot OUT(z)$$

$$D^f(z) = \frac{(z - 1) \cdot N}{z \cdot (N \cdot T_s + 1) - 1}$$

Realizando las correspondientes transformaciones mostradas anteriormente se obtiene la ecuación en diferencias que combina las dos estrategias que mejoran la señal de la acción derivativa:

$$u_o^1(k) = a \cdot e(k) + b \cdot e(k - 1) + u_o^1(k - 1)$$

$$u_o^2(k) = \frac{a'}{b'} \cdot (out(k) - out(k - 1)) + \frac{1}{b'} \cdot u_o^2(k - 1)$$

$$u_o(k) = u_o^1(k) - u_o^2(k)$$

$$a' = N \cdot K_d$$

$$b' = N \cdot T_s + 1$$

Anti wind-up

Generalmente, es seleccionado el compensador PID para regular las variables de los procesos de una planta industrial. Sin embargo, si el proyecto de este controlador PID no lleva en consideración las limitaciones físicas del proceso, podremos llegar a tener un comportamiento no deseado de la variable controlada.

En un lazo de control, muchas veces por las condiciones deseadas de desempeño, es muy probable alcanzar los límites de los actuadores. Cuando esto sucede el sistema se satura y no puede añadir más energía al proceso. Este comportamiento se mantiene hasta que la variable controlada se aproxime del a la señal de referencia y el sistema entre nuevamente dentro del rango de funcionamiento.

Si en nuestro lazo de control usamos un controlador con acción integral, durante la saturación del sistema, la integración del error continua y se va acumulando. Esto se traduce en que el término integral puede volverse exageradamente grande, o sea se sobrecarga.

La consecuencia de esta sobrecarga de la acción integral, es que cuando el sistema entra de nuevo en dentro del rango de funcionamiento (sale de las limitaciones físicas del proceso) se debe esperar a que la integral se descargue para comenzar a reaccionar. Esto conlleva a tener períodos transitorios muy largos.

El “anti wind-up” es un mecanismo de control que evita que la acción integral se sobrecargue cuando el sistema se encuentra saturado. Esto es importante porque cuando el sistema entra de nuevo dentro del rango de funcionamiento del actuador, permite que el controlador pueda actuar inmediatamente, sin necesidad de esperar a que todo el error acumulado se descargue.

El “anti wind-up” tipo “back-calculation” es el que se ha seleccionado para nuestro sistema. Consiste en obtener la diferencia de saturación del sistema y multiplicarlo por una ganancia para luego hacer la diferencia con la ganancia de la parte integral del controlador, de este modo la señal del término integral se ajusta dinámicamente evitando un crecimiento fuera de lo normal.

La siguiente ecuación en diferencias es la que se debe implementar en el microcontrolador para conseguir dicha función:

$$u_o^a(k) = u_o(k) - \frac{K_w T_s}{2} (w(k) + w(k-1))$$

$$w(k) = u_o(k) - u(k)$$

Diagrama de bloques de un controlador en el que se ha implementado el “anti wind-up” de descarga de acción integral.

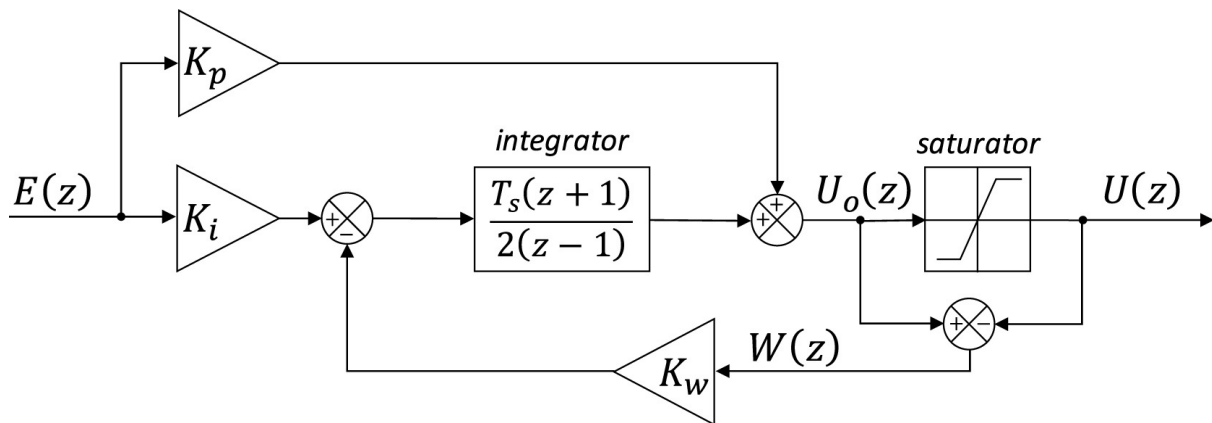


Ilustración 41: Diagrama de bloques del "anti wind-up"

En nuestro sistema se hace uso combinado del “anti wind-up”, “anti derivative kick” y “derivative filter”, entonces la ecuación en diferencias de un controlador PID implementada en el microcontrolador resulta ser:

$$u_o^3(k) = \frac{K_w \cdot T_s}{2} \cdot (w(k) + w(k-1)) + u_o^3(k-1)$$

$$u_o^a(k) = u_o(k) - u_o^3(k)$$

En artículos relacionados con el estudio de este método se propone una ecuación para la obtener el valor de la constante de descarga de acción integral, y se corresponde con la siguiente expresión para un controlador PID.

$$K_w = \frac{1}{\sqrt{K_d/K_i}}$$

Diseño del modelo matemático

Para diseñar e implementar con éxito un controlador que gobierne un sistema de forma satisfactoria, es favorable disponer de un modelo matemático que reproduzca fielmente su comportamiento. Esto facilitará en tiempo y precisión su controlabilidad al usarse técnicas más sencillas y fiables, y por otro lado hará que el sistema se estabilice sin complicaciones.

El modelo matemático de un sistema dinámico se puede definir como el conjunto de ecuaciones, generalmente diferenciales, que representan la dinámica del sistema con bastante precisión. Este modelo matemático no será único, dependiendo del sistema del que se trate y de las circunstancias específicas, un modelo será más conveniente que otros.

Por ejemplo, en problemas de control óptimo, es provechoso usar representaciones en el espacio de estados. En cambio, para los análisis de la respuesta transitoria o en frecuencia de sistemas lineales, una función de transferencia puede ser más conveniente.

En la obtención de un modelo matemático se debe establecer un equilibrio entre la simplicidad del mismo y la precisión de los resultados del análisis. Teniendo en cuenta esta serie de consideraciones, nuestro sistema de control simplificado equivale al siguiente diagrama de bloques:

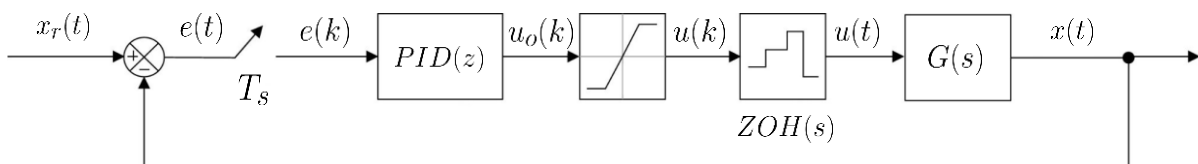


Ilustración 42: Diagrama de bloques simplificado del sistema

El modelo matemático del sistema corresponde al bloque "G(s)", en realidad este bloque es el producto de varios bloques en serie (componentes eléctricos y electrónicos, partes mecánicas, servomotor, perfil de aluminio, esfera...). En busca de ese equilibrio entre simplicidad y precisión, algunos de estos bloques se ignoran considerando los de más relevancia en el sistema. En nuestro caso nos centramos en obtener la función de transferencia que relaciona el giro del eje del servomotor con la posición de la esfera.

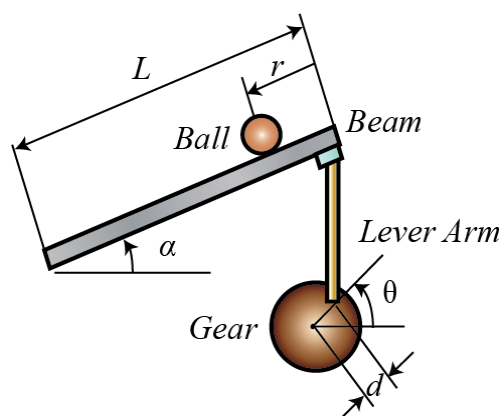


Ilustración 43: Parámetros geométricos del sistema "Beam and Ball";
<https://ctms.engin.umich.edu/CTMS/index.php?example=BallBeam§ion=SystemModeling>

La elección de la formulación lagrangiana para desarrollar las ecuaciones de movimiento, se fundamenta en las ventajas que introduce frente a las ecuaciones de Newton. Por definición, la lagrangiana de un sistema es la diferencia entre las energías cinética y potencial, siendo por tanto un escalar. Así, mientras que con el procedimiento de Newton se coloca el énfasis sobre el agente exterior que actúa sobre el cuerpo (la fuerza), con el de Lagrange se manejan magnitudes asociadas al cuerpo (energías cinética y potencial).

Este hecho es especialmente importante por diversos motivos. En primer lugar, hace que la lagrangiana de un sistema sea invariante ante los cambios de coordenadas. En segundo lugar, no es necesario establecer de forma explícita las fuerzas que actúan sobre el cuerpo, cosa que a veces ni siquiera es posible. Entonces el modelo del sistema, se consigue al aplicar la ecuación de Lagrange. Debido a que el actuador del sistema es un servomotor, no existe elasticidad, y por lo tanto no existe energía potencial en el sistema. Siendo así, el lagrangiano está compuesto por energía cinética y se expresa como:

$$U = \frac{1}{2} \cdot m \cdot v + \frac{1}{2} \cdot I_b \cdot \omega^2 + \frac{1}{2} \cdot I_a \cdot \alpha^2$$

Donde m es la masa de la esfera, " v " es la velocidad de traslación de la esfera, " ω " es la velocidad angular de la esfera, " α " es el ángulo del carril pivotante con respecto al eje horizontal, " I_a " es el momento de inercia del carril, " I_b " es el momento de inercia de la esfera.

Como los sensores del sistema miden la posición de la esfera " x ", es necesario representar el modelo del sistema en términos de estas dos variables, por lo que ahora se encuentran las relaciones entre " r ", " ω " y " x ". De eso se puede observar que la distancia recorrida se encuentra expresada por:

$$x = r \cdot \varphi$$

Donde " φ " es el ángulo rotacional de la esfera con respecto al eje del carril pivotante y r es el radio de la esfera. El ángulo total de la esfera es la suma de su ángulo rotacional y el ángulo de rotación del carril " α ". La velocidad angular de la esfera está dada por:

$$\omega = \varphi + \alpha = \frac{\dot{x}}{r} + \alpha$$

La velocidad de traslación de la esfera está dada por:

$$v = \sqrt{(\dot{x})^2 + (x \cdot \dot{\alpha})^2}$$

Al utilizar las ecuaciones para la traslación y rotación en el lagrangiano, se obtiene:

$$U = \frac{1}{2} \cdot \left(m \cdot ((\dot{x})^2 + (x \cdot \dot{\alpha})^2) + I_b \cdot \left(\frac{\dot{x}}{r} + \alpha \right)^2 + I_a \cdot \alpha^2 \right)$$

$$\frac{d}{dt} \cdot \left(\frac{d}{dx} \cdot U \right) - \frac{d}{dx} \cdot U = m \cdot g \cdot \sin(\alpha)$$

$$\left(m + \frac{I_b}{r^2}\right) \cdot \ddot{x} + \left(\frac{I_b}{r^2}\right) \cdot \ddot{\alpha} - m \cdot x \cdot \dot{\alpha}^2 = m \cdot g \cdot \sin(\alpha)$$

A partir del modelo no lineal de la planta se hacen cero las derivadas con respecto al tiempo de forma que:

$$0 = m \cdot g \cdot \sin(\alpha)$$

La solución de la ecuación anterior es $n\pi$ con $n = 0, 1, 2, \dots$. Y debido a que el carril pivotante solo rota dentro del intervalo $[-30, 30]$ grados, se toma como solución el ángulo para $n = 0$. Además, se puede observar que la solución no depende de la posición de la bola. A partir del punto de equilibrio y linealizando alrededor de los puntos cercanos a este, se tiene:

$$\alpha = 0 \leftrightarrow \sin(\alpha) = 0$$

Por lo que las derivadas de orden superior de esta variable se hacen despreciables, obteniendo la siguiente ecuación diferencial:

$$\ddot{x} = \frac{m \cdot g \cdot \alpha}{m + \frac{I_b}{r^2}}$$

Es necesario aplicar la ecuación que relaciona el ángulo del carril pivotante “ α ” respecto al ángulo del servomotor “ θ ”, siendo “ d ” el brazo de palanca del servomotor y “ L ” el brazo de palanca del perfil. Esta ecuación se puede aproximar como lineal y se especifica como:

$$\alpha = \frac{\theta \cdot d}{L}$$

Aplicando la transformada de Laplace con condiciones iniciales nulas se obtiene:

$$\frac{X(s)}{\theta(s)} = \frac{m \cdot g \cdot d}{L \left(m + \frac{I_b}{r^2}\right) \cdot s^2}$$

Sustituyendo los valores se obtiene la siguiente función de transferencia continua:

$$m = 0.002 \text{ [Kg]}$$

$$g = 9.806 \text{ [m/seg}^2\text{]}$$

$$d = 0.050 \text{ [m]}$$

$$L = 0.075 \text{ [m]}$$

$$r = 0.030 \text{ [m]}$$

$$I_b = 6.498 \cdot e^{-7} \text{ [Kg} \cdot \text{m}^2\text{]}$$

$$G(s) = \frac{X(s)}{\theta(s)} = \frac{0.0009806}{0.00021 \cdot s^2}$$

Nuestro sistema utiliza un servomotor que necesita recibir un pulso cada 0.02 segundos para funcionar, entonces, es interesante comprobar si el periodo de muestreo del controlador también puede funcionar a 0.02 segundos.

Para conocer la viabilidad de esta decisión es necesario comprobar si se cumple el teorema de Shannon. Este teorema dice que la frecuencia de muestreo seleccionada debe ser mayor o igual al doble del componente de frecuencia máxima de la señal en tiempo continuo (s).

$$\omega_s = \frac{2 \cdot \pi}{T_s} \geq 2 \cdot \omega_m$$

$$T_s = 0.02 \text{ [s]}$$

Esto significa que la componente de máxima frecuencia “ ω_s ” del modelo matemático del sistema debe ser inferior a 157.08 rad/s.

$$\omega_s \leq 157.08 \text{ [rad/s]}$$

Generando en Matlab un diagrama de bode de la función de transferencia en el plano “s” se puede determinar dicha frecuencia cuando la magnitud es igual a 0 decibelios.

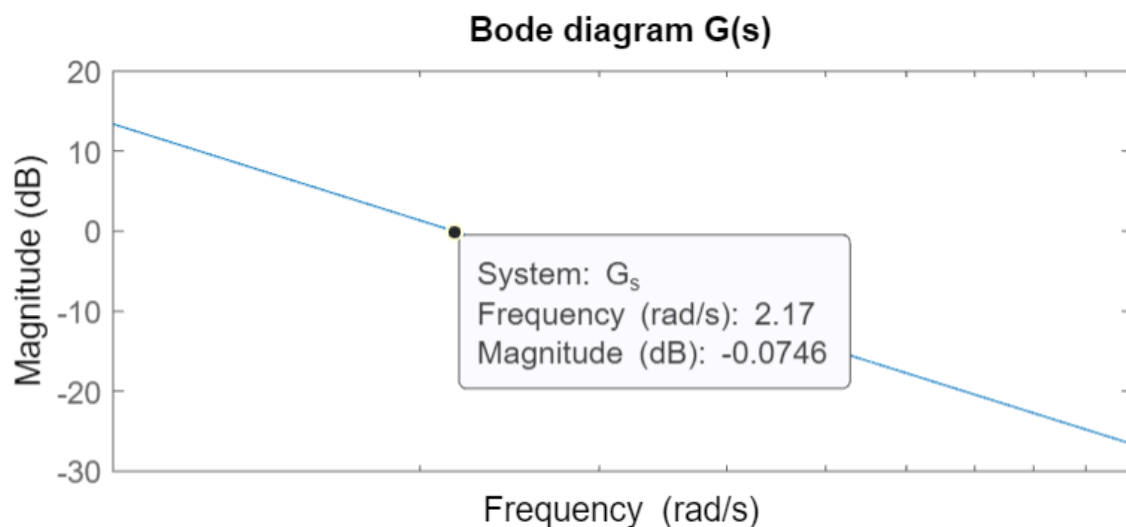


Ilustración 44: Gráfica de la magnitud del diagrama de Bode G(s)

De la gráfica de magnitud del diagrama de bode se obtiene que:

$$\omega_m \approx 2 \text{ [rad/s]}$$

En consecuencia, el tiempo de muestreo elegido es válido para el control de nuestro sistema según el teorema de Shannon.

Realizando la transformada Z para un periodo de muestreo de 0.02 segundos se obtiene la siguiente función de transferencia discreta:

$$G(z) = \frac{0.0009339 \cdot (z + 1)}{(z - 1) \cdot (z - 1)}$$

Resultados simulados

En esta parte se realizan las simulaciones de la función de transferencia discreta de nuestro sistema. Se utiliza la aplicación de Matlab “PID Tuner” para obtener los parámetros del controlador según la respuesta deseada y “Simulink” para representar la respuesta de nuestro sistema según la entrada deseada.

Iniciamos las simulaciones con la función de transferencia teórica con un periodo de muestreo de 0.02 segundos:

$$G(z) = \frac{0.0009339 \cdot (z + 1)}{(z - 1) \cdot (z - 1)}$$

$$T_s = 0.02 \text{ [s]}$$

Se ajusta la respuesta del sistema de manera que se establezca en un intervalo de 2.5 a 3 segundos aproximadamente ya que no deseamos que haya un sobre pico superior al 20% debido a que existe una zona limitada en el carril pivotante en la que los sensores no funcionan correctamente.

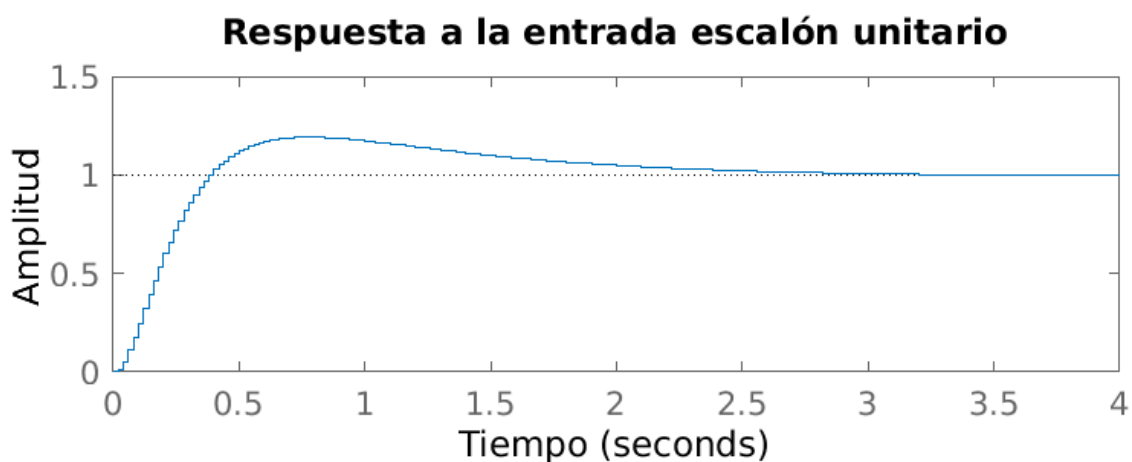


Ilustración 45: Gráfica respuesta a escalón unitario del sistema en lazo cerrado

En la siguiente tabla se muestran los parámetros obtenidos del controlador y la respuesta:

Kp	1.0127	Rise time	0.26 s
Kd	0.5201	Settling time	2.52 s
Ki	0.4873	Overshoot	19.1%
Tf	0.0397	Peak	1.19

Tabla 1: Parámetros del controlador PIDF a entrada escalón unitario

El diseño del PID implementa el filtro derivativo en el que el parámetro “Tr” es la inversa de la frecuencia del filtro en consecuencia obtenemos:

$$N = \frac{1}{T_f} \approx 25 \text{ (rad/seg)}$$

Los valores de las constantes del controlador PID se introducen en las variables del código del microcontrolador y se realizan los ensayos con los 3 tipos de señales de referencia utilizados habitualmente (escalón, seno y rampa).

Durante los ensayos registramos los datos de las señales de referencia, error, control y salida del sistema, utilizando la interfaz gráfica para realizar la comparativa con los resultados obtenidos con la aplicación de Matlab “Simulink”.

Los datos se organizan en matrices de 2 columnas y desde el espacio de trabajo de Matlab se importan directamente en el diagrama de bloques que simula el PID, también se ajustan los parámetros de las constantes obtenidas anteriormente, el saturador, anti wind-up y se añade un bloque que representa la holgura del mecanismo que actúa como “Death zone”.

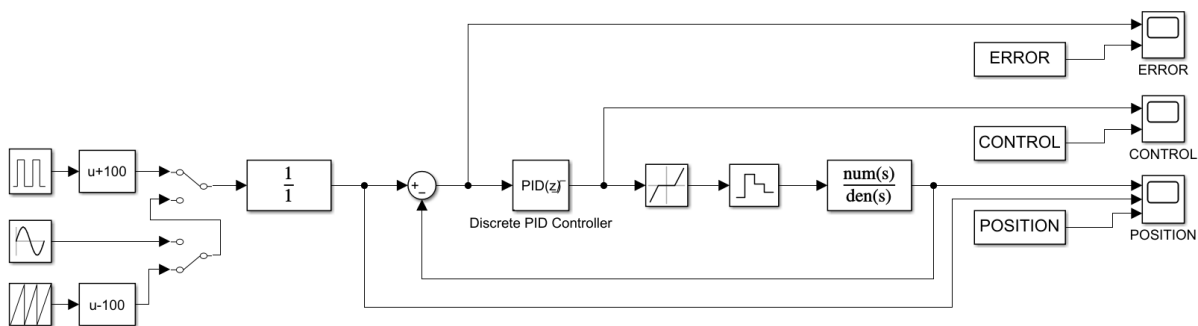


Ilustración 46: Diagrama de bloques completo del sistema en Simulink

El rango de actuación del bloque “Death Zone” se ha determinado de forma experimental, pues no es constante, depende de la holgura que ha ido incrementando en el reductor del servomotor durante los ensayos. En consecuencia, atendiendo a la resolución mínima del servomotor y la que realmente se obtiene, se ha determinado un intervalo de aproximadamente -1 a 1 grados respecto la posición instantánea del servomotor en los que el sistema no actúa.

Los valores del saturador introducidos en el bloque del PID discreto son de -47 a 52 grados que se ajustan al estudio mecánico realizando para mantener el carril pivotante por debajo de un ángulo máximo de rotación de 30 grados positivos y negativos.

Utilizando los bloques generadores de señales y “offset” se simulan las señales de referencia que utilizamos en el microcontrolador para realizar los ensayos.

El valor de la constante del “anti wind-up” se incluye en el bloque del PID discreto. En este caso conociendo los valores de las constantes teóricas y atendiendo a la ecuación mencionada anteriormente se obtiene:

$$K_w = 0.9679 \approx 1$$

Entrada escalón

Se puede comprobar gráficamente que las señales son parecidas, cabe destacar la importancia de la “Death zone” que afecta a la estabilidad de la señal simulada y real, aunque en este caso parece que afecta a la señal real de forma asimétrica, además la señal real se ve afectada también por la imperfección de la superficie de la esfera, la fricción y el ruido permanente en la detección de su posición.

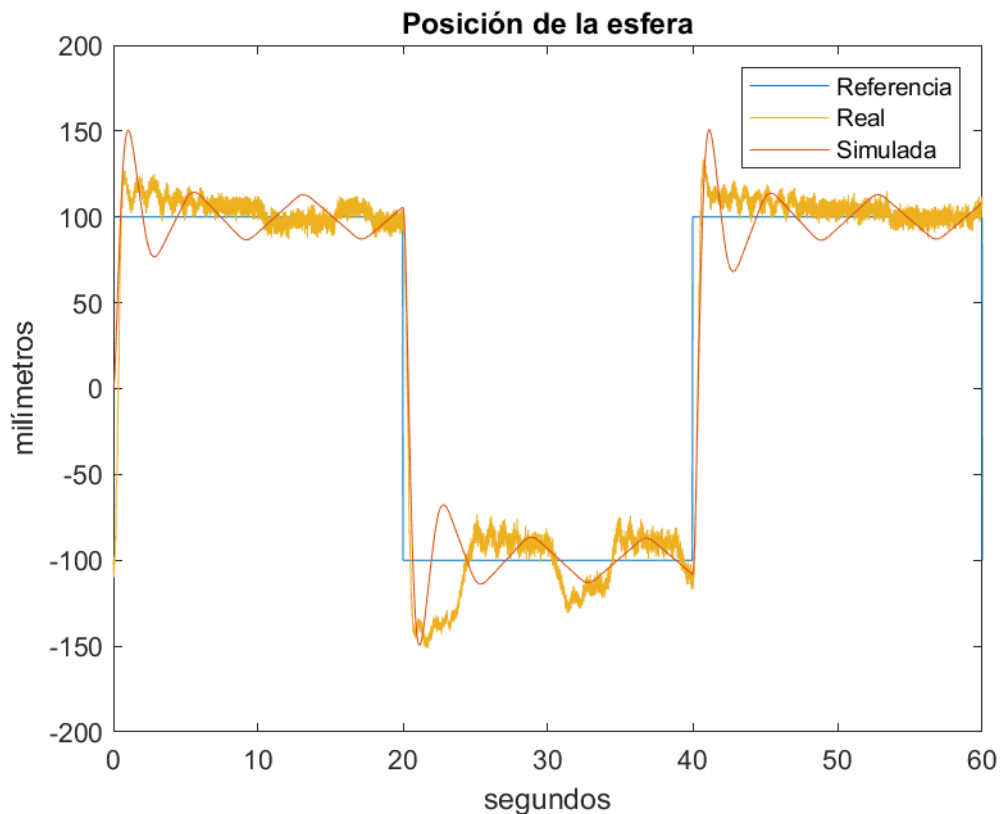
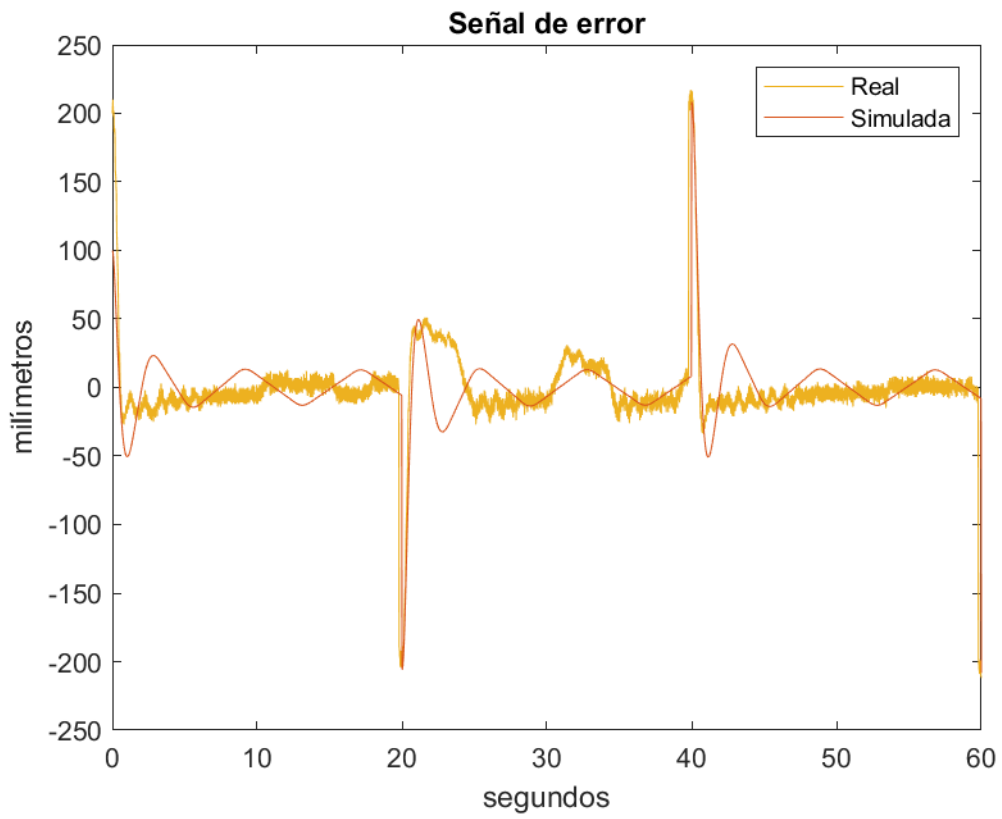
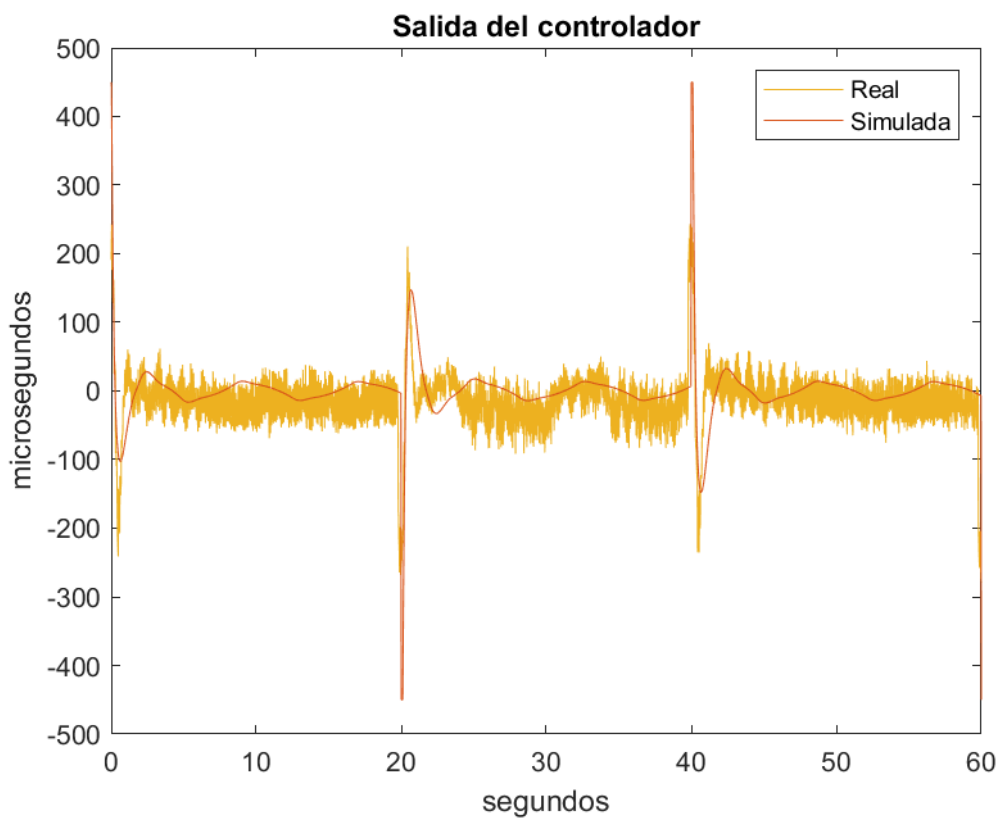


Ilustración 47: Simulación de la posición a entrada escalón



Il·lustración 48: Simulación el error a entrada escalón



Il·lustración 49: Simulación de la salida del controlador a entrada escalón

Entrada senoidal

Se puede comprobar gráficamente que para esta señal de referencia la presencia de ruido ha sido muy desafortunada, generando irregularidades en el desplazamiento de la esfera, que se incrementan sumando los efectos generados por la “Death zone” del servomotor. Las imperfecciones en superficie de la esfera producen esta discontinuidad en su desplazamiento. A pesar de ello la esfera consigue seguir la señal de referencia con unas oscilaciones importantes.

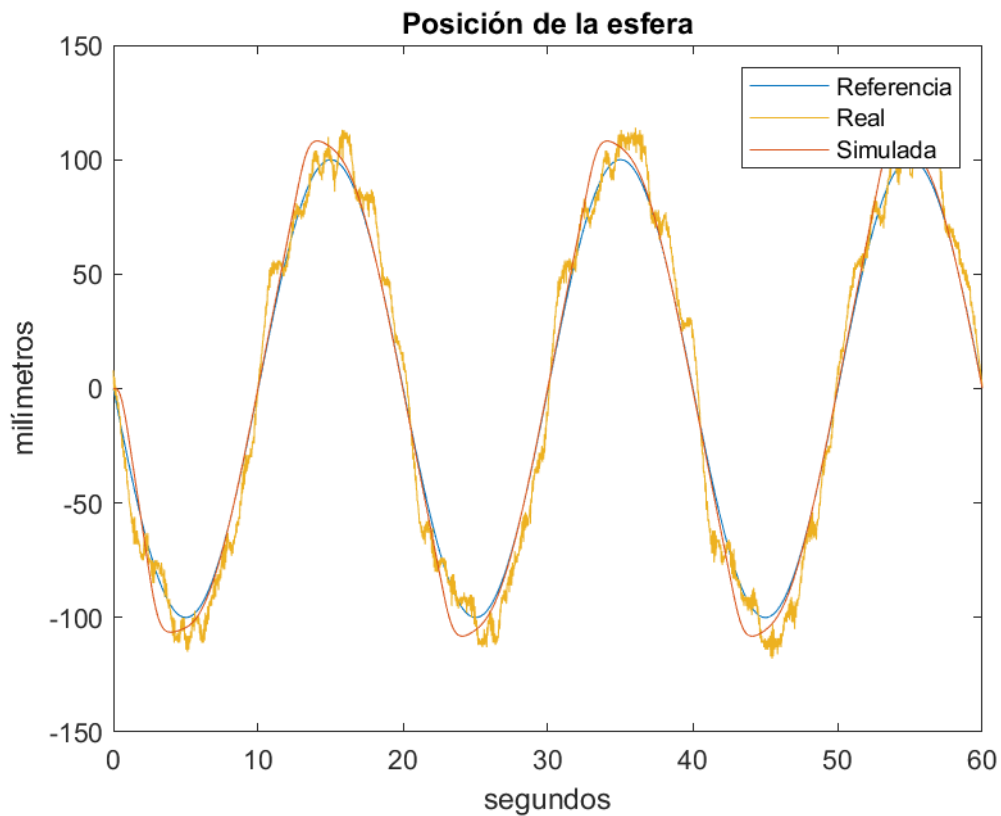


Ilustración 50: Simulación de la posición a entrada senoidal

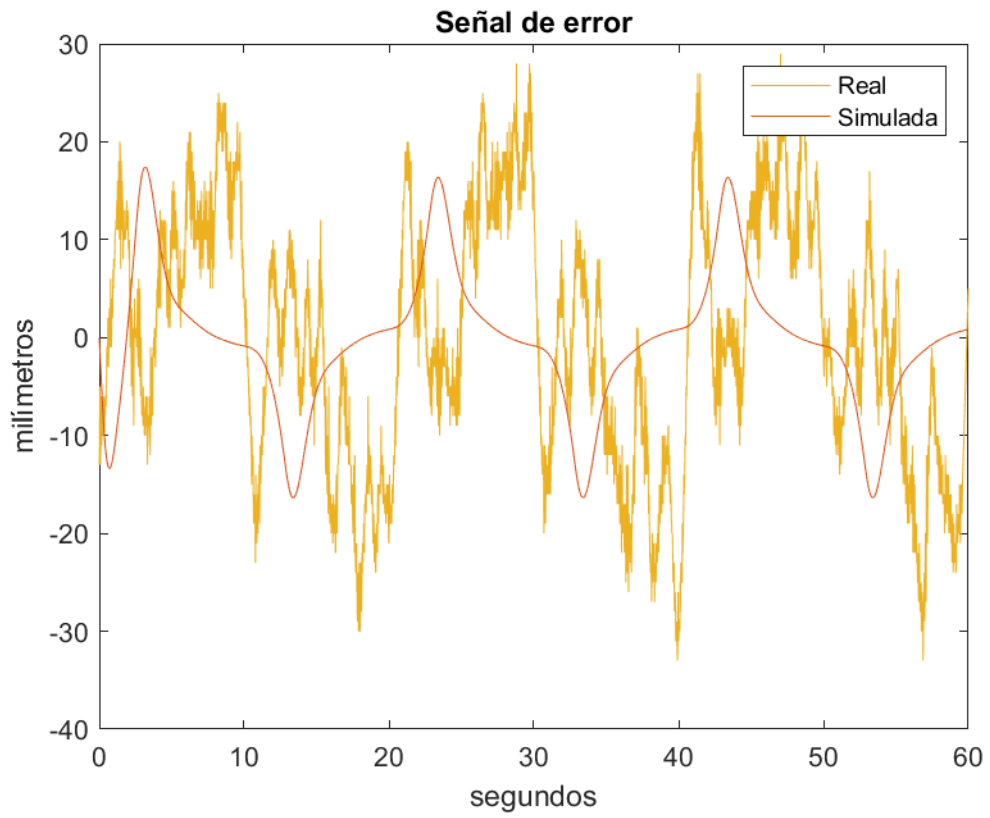


Ilustración 51: Simulación el error a entrada senoidal

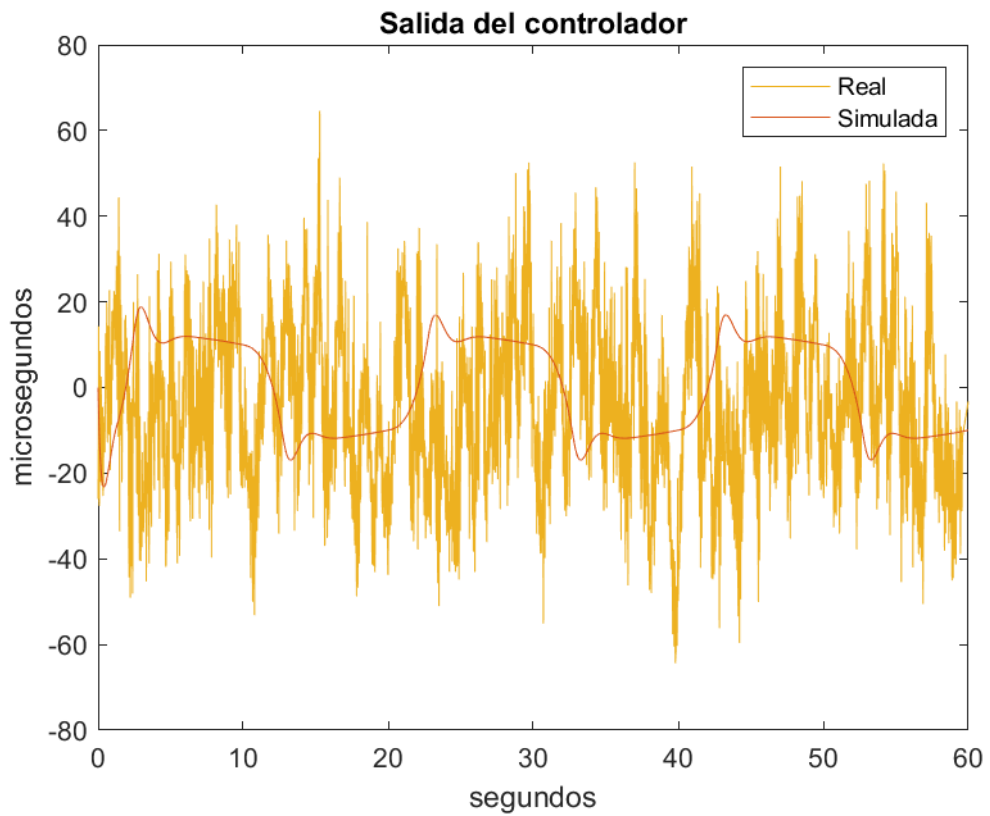


Ilustración 52: Simulación de la salida del controlador a entrada senoidal

Entrada rampa

Se puede comprobar gráficamente cómo las señales tienen cierta similitud, cabe destacar la importancia de la “Death zone”, que afecta a la estabilidad de la señal simulada y real, además la señal real se ve afectada también por la imperfección de la superficie de la esfera, la fricción y el ruido permanente en la detección de su posición.

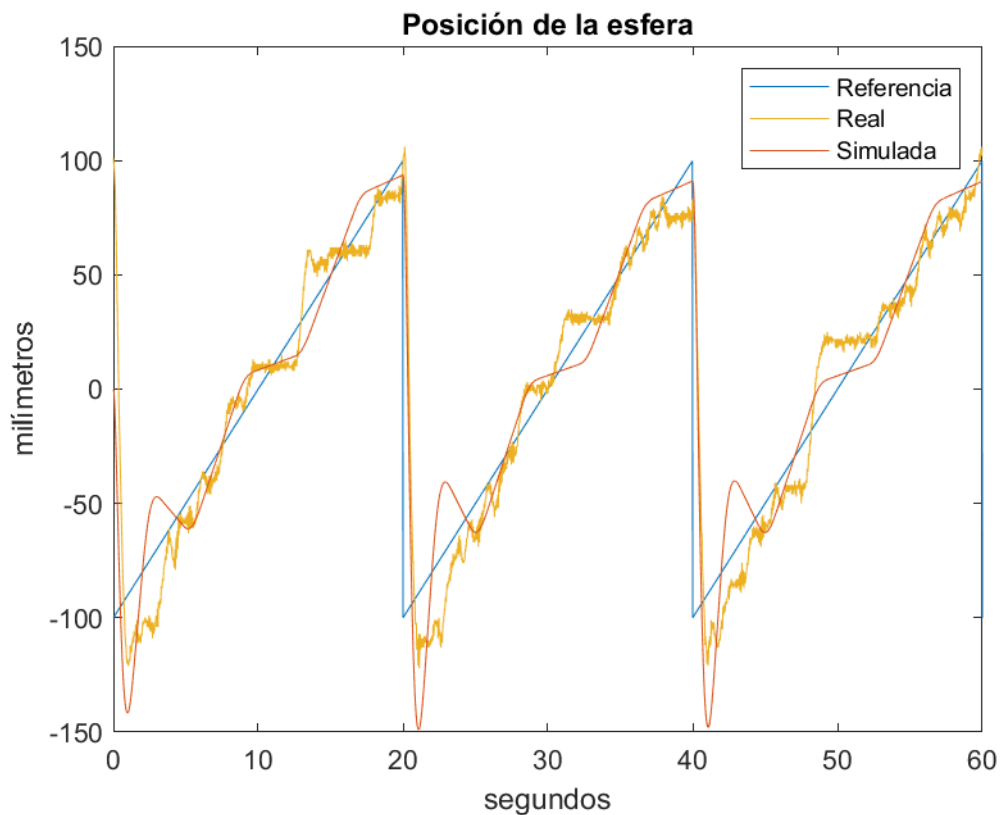


Ilustración 53: Simulación de la posición a entrada rampa

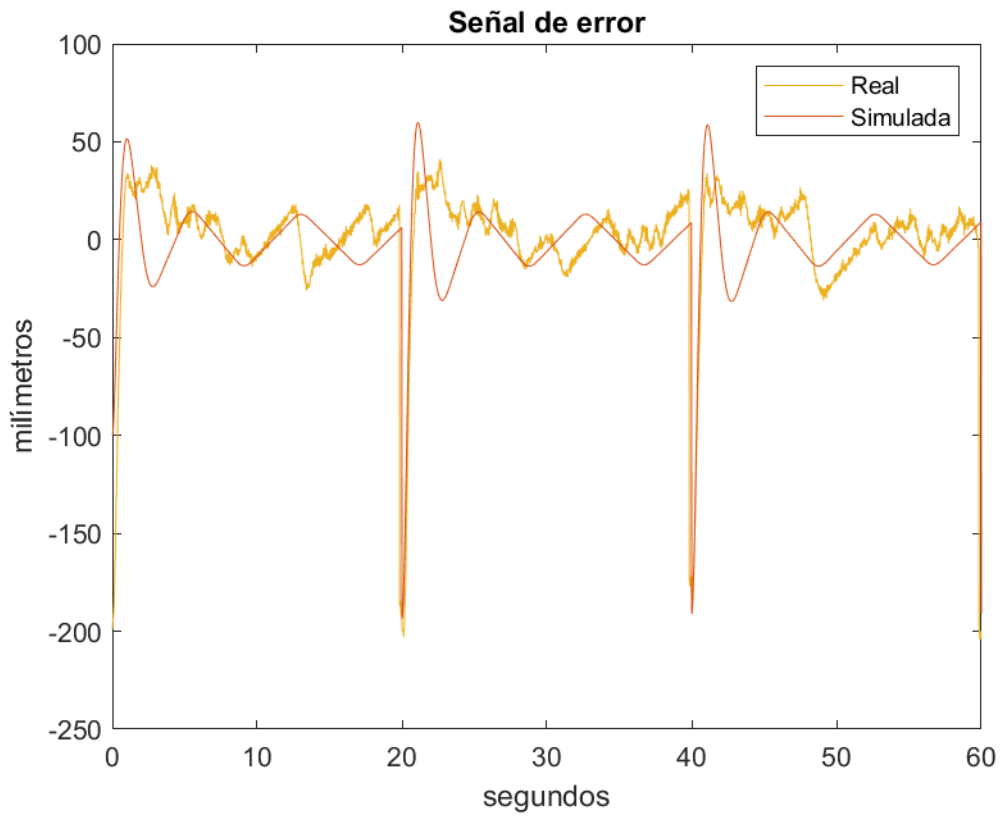


Ilustración 54: Simulación el error a entrada rampa



Ilustración 55: Simulación de la salida del controlador a entrada rampa

Programación

Interfaz gráfica

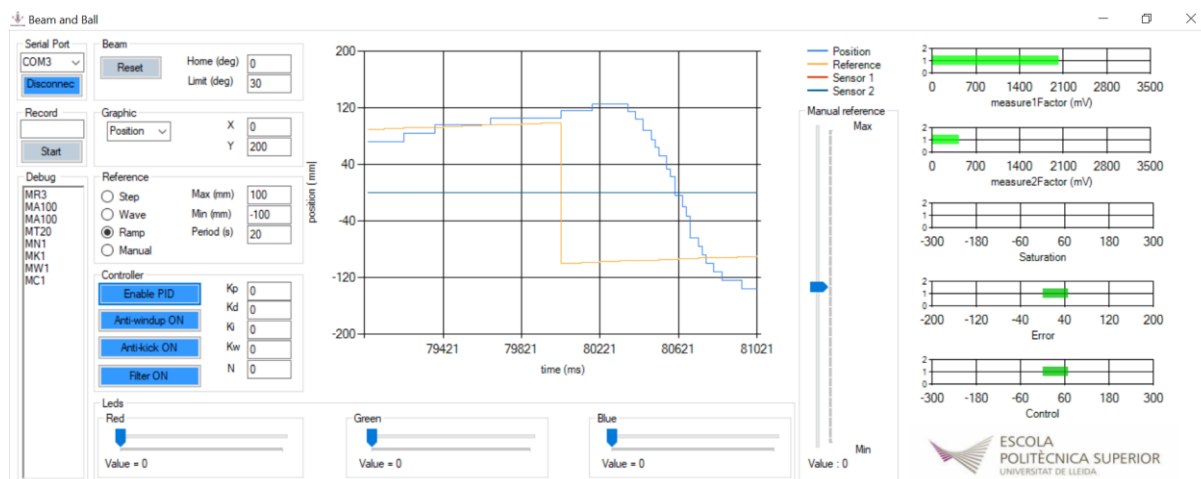


Ilustración 56: Diseño de la interfaz gráfica en funcionamiento

Para realizar este proyecto es imprescindible disponer de una plataforma de comunicación con el sistema intuitiva y versátil, se requiere para realizar ensayos, calibraciones y extraer información. Para ello se ha desarrollado una aplicación utilizando el software Visual Studio mediante lenguaje C#.

Esta aplicación permite visualizar a tiempo real el estado del sistema, modificar las variables del microcontrolador para realizar ajustes y registrar los datos que recibe a través del puerto serie.

Se pueden clasificar las funciones implementadas en grupos, siendo los siguientes:

- Comunicación con el puerto serie: Permite establecer comunicación mediante un cable USB con la placa de evaluación Arduino Due u otras. Detecta todos los puertos disponibles y permite seleccionarlos.
- Registro de datos: La aplicación recibe cadenas de datos, las separa mediante un carácter específico que en nuestro caso es el espacio y las almacena en variables. Podemos dar nombre a un fichero y comenzar a registrar los datos o parar a nuestra voluntad. El resultado es un archivo de texto con columnas de datos que posteriormente se pueden procesar para generar simulaciones, calibraciones o análisis experimental de la planta del sistema.
- Ajustes del carril pivotante: Permite realizar ajustes del ángulo de rotación del carril pivotante para calibrar la posición inicial de los ensayos, también se puede limitar el ángulo máximo de rotación el mismo lo que resulta equivalente a saturar la señal de salida del controlador. También se puede realizar un reinicio de la posición del perfil en cualquier instante en caso de malfuncionamiento del sistema o caída de la esfera.

- Selección de gráficas: Podemos seleccionar el tipo de datos que se muestran en la gráfica principal, pudiendo visualizar la posición de la esfera, la señal de referencia y las señales de los sensores de proximidad.
- Selección de señal de referencia: La aplicación permite seleccionar entre 4 tipos de señales de referencia que están implementados en el microcontrolador, podemos elegir entre una señal de tipo escalón, senoide, rampa y manual. Además, se puede elegir el periodo de la señal y los límites superiores e inferiores siendo los valores máximos 150 y -150 milímetros debido al margen de sobre pico del controlador que sitúa la posición de la esfera entre las posiciones máximas de 180 y -180 milímetros.
- Ajustes del controlador PID: En este grupo de opciones se pueden activar o desactivar el controlador PID y sus complementos (anti wind-up, anti derivative kick y derivative filter), también se pueden modificar los valores de las constantes del controlador y la frecuencia de corte del filtro.
- Ajustes de la tira de LED: La tira de LED puede cambiar de color e intensidad moviendo los deslizadores que hay en la aplicación.
- Señal de referencia manual: El deslizador vertical que hay en la aplicación permite generar una señal manual si la opción está activa previamente con los mismos límites mencionados anteriormente.

Los datos que recibe la aplicación se manifiestan en una serie de gráficas en las que se pueden ver las siguientes variables:

- Posición de la esfera en milímetros.
- Señales analógicas de los sensores de proximidad en milivoltios.
- Señal de error del sistema en milímetros.
- Señal de salida del controlador en microsegundos.
- Señal de saturación del controlador en microsegundos.

Debido a que este sistema tiene un periodo de muestreo bastante rápido se ha establecido un “baud-rate” de comunicación de 250000 bits por segundo entre la aplicación y el microcontrolador para asegurar una velocidad de transmisión de datos a tiempo real.

Programación de la tira de LED

La tira de LED se utiliza para mostrar la posición de la esfera y crear una posición de referencia a la vez. El fabricante “Adafruit” proporciona la librería “Adafruit_DotStar.h” específica para el control de la tira mediante el uso de un microcontrolador.

El uso de esta librería es compatible con las demás funciones de control utilizadas por el sistema y permite la comunicación por “SPI” aunque no es imprescindible para realizar el control. En este caso se decide realizar el control de la tira de LED de la forma más sencilla utilizando dos salidas digitales con funciones de reloj y transmisión de datos respectivamente.

La programación se inicia con la definición de una función propia de la librería para determinar el número de LED que se van controlar, las rutas de las salidas digitales que se van a utilizar y finalmente el tipo de sistema de colores que se desea.

```
#define NUMPIXELS          72
#define DATAPIN            43
#define CLOCKPIN           41
Adafruit_DotStar strip(NUMPIXELS, DATAPIN, CLOCKPIN, DOTSTAR_BRG);
```

Luego es necesario inicializar en “void setup()” las salidas digitales que se van a utilizar para el control y los chips internos de la tira:

```
pinMode(DATAPIN, OUTPUT);
digitalWrite(DATAPIN, LOW);
pinMode(CLOCKPIN, OUTPUT);
digitalWrite(CLOCKPIN, LOW);
strip.begin();
```

La función de la tira consiste en encender 3 puntos de un color que se corresponden con la posición real de la esfera y un punto que indica la posición deseada. Para ello se dispone de 2 variables “dist” y “ref”, que almacenan valores en milímetros. La tira dispone de 72 LED por lo que es necesario realizar una calibración que relacione la distancia en milímetros con las unidades de LED a lo largo de la tira. Con las medidas de la tira adaptadas al origen de la posición del carril pivotante se obtiene el siguiente gráfico con la correspondiente recta de calibración:

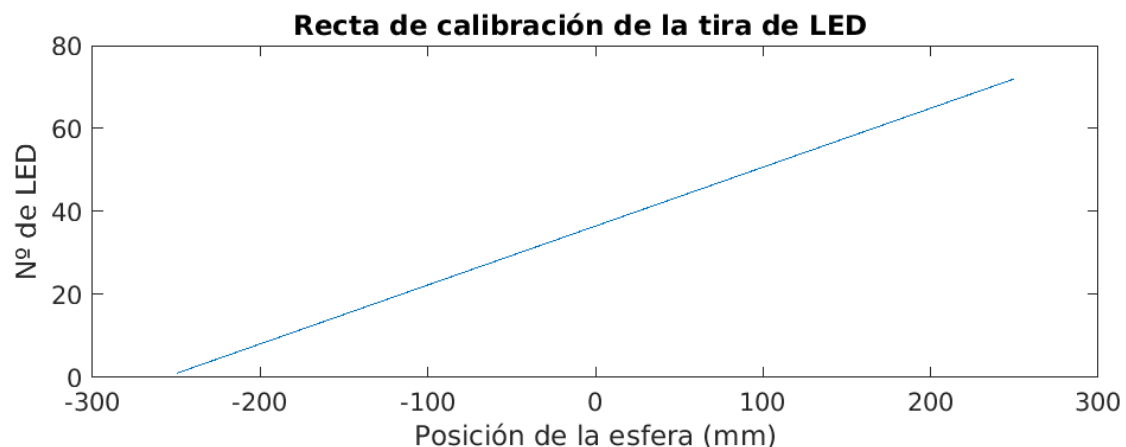


Ilustración 57: Recta de calibración de la tira de LED

$$n = \frac{dist}{7} + 36$$

Finalmente para encender cada LED de manera independiente se utiliza el siguiente fragmento de código en el “void loop()”. Este código apaga todo e inmediatamente asigna un color RGB a cada led asignada mediante una función propia de la librería. Si se ha seleccionado la opción de encender la tira de LED muestra los colores y posición, en caso contrario permanece apagada.

```
strip.clear();
strip.setPixelColor(dist / 7 + 35, R, G, B);
strip.setPixelColor(dist / 7 + 36, R, G, B);
strip.setPixelColor(dist / 7 + 37, R, G, B);
strip.setPixelColor(ref / 7 + 36, R, G, B);
if ( led == true ){
    strip.show();
}
```

Calibración de los sensores de proximidad

El sistema utiliza dos sensores de proximidad analógicos para estimar la posición de la esfera, cada sensor se ubica en un extremo del carril pivotante mediante un soporte. Es necesario realizar una calibración de los dos sensores para poder relacionar los valores que proporciona la salida analógica con la distancia de la esfera.

Se trabaja con el modelo de sensor GP2Y0A41SK0F de la marca Sharp. Es una unidad de sensor de medición de distancia con un rango de medición de 4 a 30 cm. El sensor de medición incluye un PSD (detector sensible a la posición), un IR-LED (diodo emisor de infrarrojos) y un circuito de procesamiento de señales.

El "datasheet" del sensor proporciona un gráfico que relaciona la salida analógica del sensor con la distancia detectada. Utilizando la información del gráfico se ha obtenido una función matemática, y en los ensayos se ha podido comprobar que no se ajusta lo suficiente a la posición real de la esfera.

En consecuencia, se inicia un procedimiento de calibración manual para obtener una función matemática que se ajuste más a la posición real de la esfera. Se realizan una serie de registros en múltiples posiciones del carril pivotante, en este caso se han realizado mediciones para cada centímetro tomando como origen el centro de rotación del carril, como resultado se han obtenido una serie de ficheros de texto que almacenan datos de la salida analógica dentro del intervalo [-180, 180] milímetros.

La señal analógica es muy sensible al ruido y se puede observar al realizar una gráfica en "Matlab" de cualquier fichero de texto obtenido, por lo que se toma un valor medio de todas las muestras de cada fichero de texto y se almacenan en un nuevo vector.

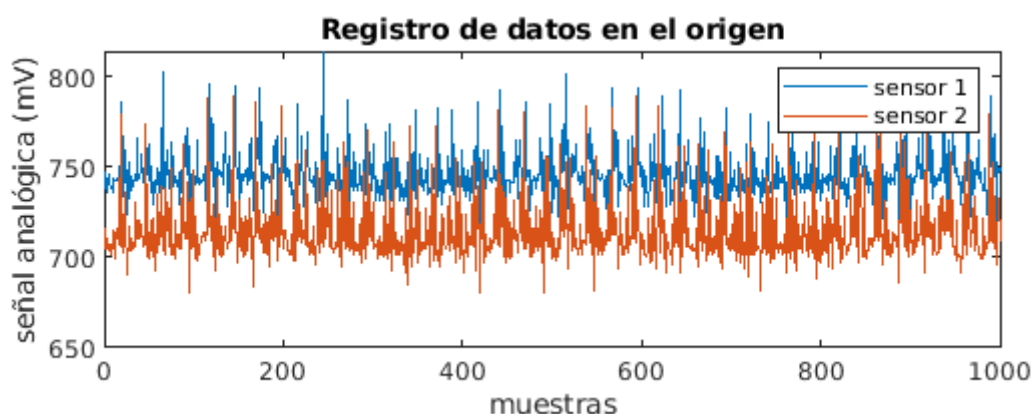


Ilustración 58: Salidas de los sensores de proximidad a posición 0 milímetros

Otro aspecto destacable es la importancia de realizar los registros de forma simultánea en ambos sensores, para evitar fallos de sincronización que producen inestabilidad o escalones en los valores obtenidos de la posición.

Utilizando una herramienta de aproximación de funciones en “Matlab”, con los valores medios de los registros se obtienen los siguientes resultados:

- Señal analógica del sensor 1:

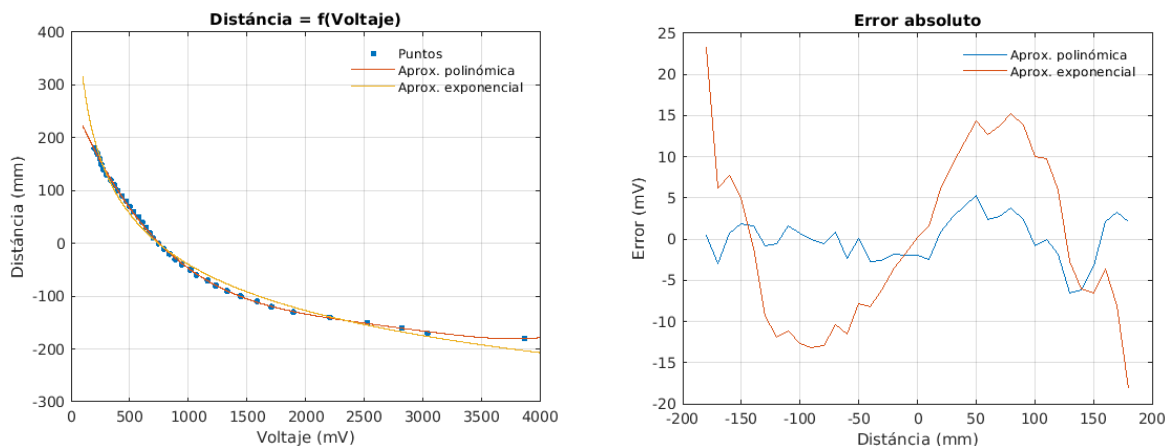


Ilustración 59: Izquierda: Aproximación matemática de la nube de puntos del sensor 1; Derecha: Error cometido

- Señal analógica del sensor 2:

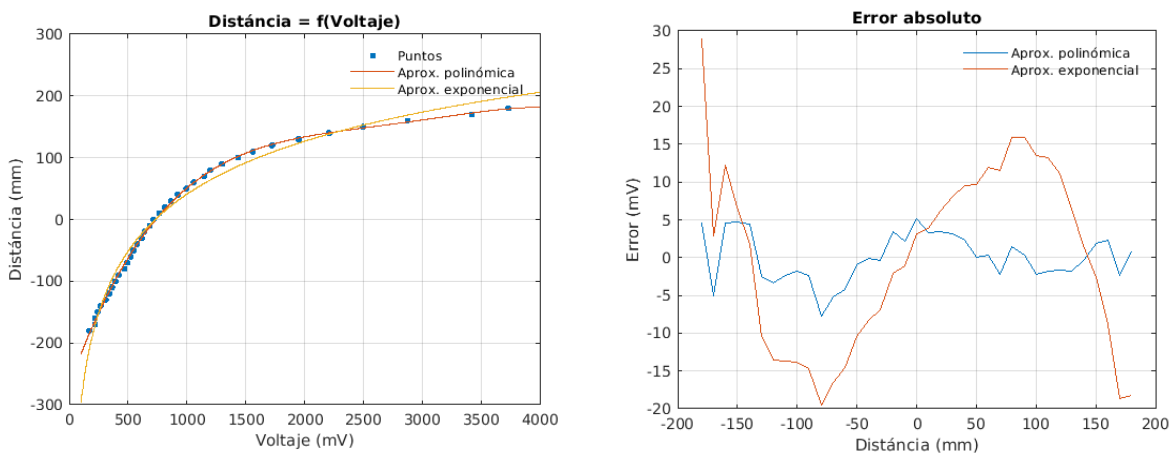


Ilustración 60: Izquierda: Aproximación matemática de la nube de puntos del sensor 2; Derecha: Error cometido

Se observa como una aproximación polinómica ofrece mejores resultados que la aproximación exponencial, dando lugar a un error muy bajo, que apenas tendrá influencia en la realidad ya que el propio ruido del sistema supera la variación del error.

Observando las gráficas del error absoluto, se observa que la precisión del sensor empeora cuando la esfera se aleja más allá del origen del carril pivotante. Por lo que el funcionamiento de los sensores será alternativo cuando el cuerpo esférico se encuentre dentro de la zona de mejor resolución y funcionarán de forma simultánea en la zona próxima al origen.

Las funciones polinómicas obtenidas son las siguientes para cada sensor:

- Función polinómica del sensor 1:

$$dist1 = 6.0133 \cdot e^{-12} \cdot x^4 - 6.5961 \cdot e^{-8} \cdot x^3 + 2.7266 \cdot e^{-4} \cdot x^2 - 0.5331 \cdot x + 2.7336 \cdot e^2$$

- Función polinómica del sensor 2:

$$dist2 = -5.4908 \cdot e^{-12} \cdot x^4 - 6.3016 \cdot e^{-8} \cdot x^3 + 2.6856 \cdot e^{-4} \cdot x^2 - 0.5298 \cdot x - 2.6808 \cdot e^2$$

Los resultados de estas funciones se almacenan en una variable llamada “dist” en el microcontrolador, para el intervalo de posición de -30 a -180 milímetros adquiere los valores de la variable “dist1”, para el intervalo de posición de 30 a 180 milímetros adquiere los valores de la variable “dist2”.

En el intervalo de -30 a 30 milímetros adquiere el valor del promedio lineal de las dos variables en función de la posición. Se configura así para evitar escalones en la señal de los sensores al cambiar de intervalo.

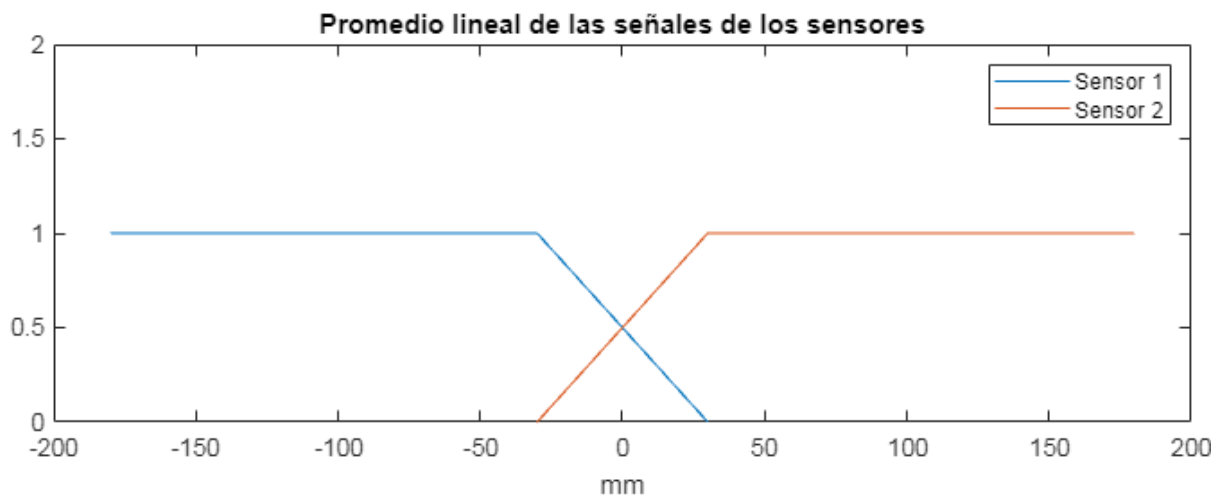


Ilustración 61: Gráfico del promedio lineal de las señales de los sensores

$$dist = dist1 \cdot \left(\frac{-dist1}{60} + 0.5 \right) + dist2 \cdot \left(\frac{dist2}{60} + 0.5 \right)$$

Control del ángulo de rotación del servomotor

El servomotor utiliza una señal digital para el control del ángulo de rotación del eje, la señal es de tipo PWM con una frecuencia específica para el servomotor. Según los parámetros del fabricante, ajustando el “duty cycle” de la señal PWM, dentro del intervalo [750, 2250] microsegundos, el ángulo de rotación del eje del servomotor aumenta de forma lineal en el intervalo de [0,180] grados.

Debido a la construcción de nuestro sistema, el servomotor parte del reposo en una posición de 90 grados, por lo que en la programación parte del reposo a los 1500 microsegundos de “duty cycle” de la señal PWM.

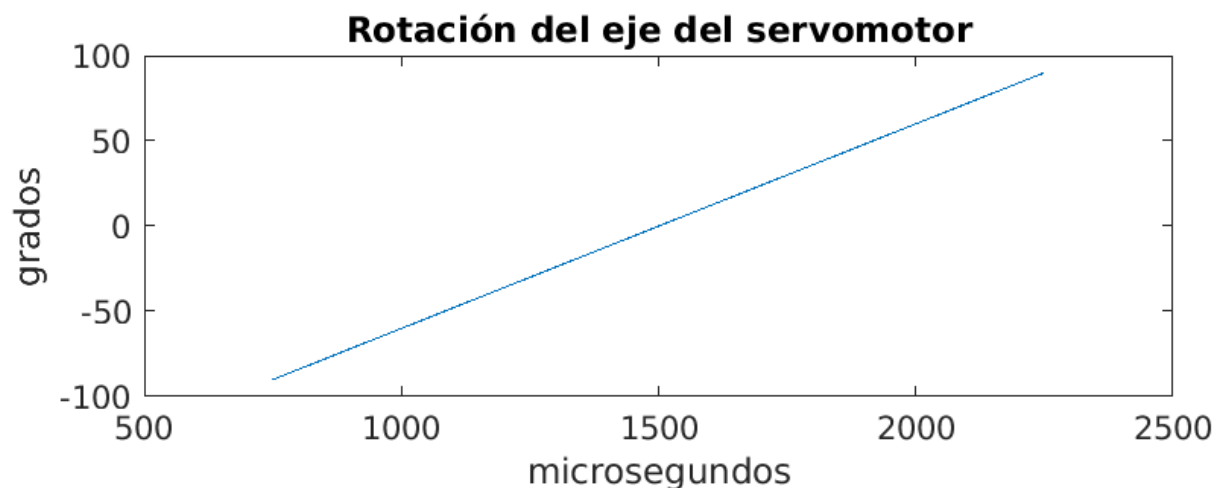


Ilustración 62: Recta de calibración del servomotor

$$\theta = 0.12 \cdot \mu s - 180$$

Para realizar el control del servomotor existe una librería compatible con el microcontrolador que utilizamos llamada “servo.h”. Desafortunadamente, el uso de esta librería no es compatible con el uso de algunos “timers”, que nos interesan para otros elementos de control, ya que usa los mismos “callback” que la librería “DueTimer.h”.

Para controlar el servomotor, primero hay que declarar la función “myservo” junto al resto de variables y luego se inicializa en el “void setup()” asignando el pin digital que se va a utilizar para el control del servomotor.

```
#define servoPin          39
Servo myservo;

void setup() {
  myservo.attach(servoPin);
}
```

El control del servomotor se realiza en el “void loop()” dentro de un condicional que evalúa si el tiempo transcurrido desde la última ejecución no supera los 0.02 segundos. En caso afirmativo, una función específica para el control del servomotor ejecuta el movimiento hasta alcanzar la posición correspondiente. La función contiene 2 variables, una que ajusta la posición inicial del carril pivotante y otra que es la salida del PID que controla la posición.

```
void loop() {  
  if (millis()>=time_0+20){  
    time_0 = millis();  
    calcStates();  
    err();  
    cPID();  
    myservo.writeMicroseconds(reposo-u0);  
  }  
}
```


Controlador

El “firmware” del controlador se encarga de integrar y sincronizar el funcionamiento de los sensores de proximidad, servomotor, tira de LED e interfaz gráfica.

Consiste en un código “C” que se implementa en el microcontrolador de la placa de evaluación “Arduino Due”, y se divide en 2 partes (cálculo del error y controlador PID).

El cálculo del error del sistema consiste en una función que compara la posición de referencia (“ref”) con la posición real de la esfera (“dist”), y la almacena en una variable llamada “error”.

Debido a las imperfecciones de construcción de la esfera, la holgura del mecanismo y la sensibilidad de los sensores al ruido, es muy difícil que el controlador se estabilice en torno a un valor nulo de error. Para solventar este inconveniente, se puede añadir una ligera tolerancia que anula el error cuando la esfera se encuentra a una distancia inferior a 5 milímetros de la posición de referencia, y además transcurre un tiempo de 500 milisegundos. Además, se ha implementado de forma opcional una variación en la frecuencia de filtrado de la acción derivativa, que depende de la distancia de la esfera respecto la posición de referencia.

```
void err(){
  error = (ref - dist);
  if(0){
    if(abs(error) < 5){
      if(cont_0){
        time_2 = millis();
        cont_0 = false;
        N = 10;
      }
      else if(millis()>time_2+500){
        error = 0;
        N = 5;
      }
    }
    else{
      cont_0 = true;
      N = 25;
    }
  }
}
```

El controlador PID es una función que contiene el controlador básico y todos sus accesorios para mejorar la respuesta.

Utilizando unas variables auxiliares es posible obtener un PID con todas las combinaciones posibles del “anti-derivativekick”, “derivative filter” y “anti-windup” (“dick”, “dftr” y “awu” respectivamente).

En la misma función se ha implementado el saturador de la señal de control, en el que una variable establece el ángulo máximo de rotación del carril pivotante.

```
void cPID(){
    wu = u - u0; // Calculo de la saturación

    if(dftr && !dkick){ // PID con derivative filter
        u = (kp*(1+N*Ts) + ki*Ts/2*(1+N*Ts) + kd*N)/(1+N*Ts) * error +
            (-kp*(2+N*Ts) + ki*N*pow(Ts,2)/2 - 2*N*kd)/(1+N*Ts) * error_1 +
            (kp + kd*N - ki*Ts/2)/(1+N*Ts) * error_2 -
            (-N*Ts-2)/(1+ N*Ts) * u_1 - 1/(1+N*Ts) * u_2;
    }

    if(!dftr && !dkick){ // PID sin accesorios
        u = u_1 + (kp + ki*Ts/2 + kd/Ts)*error -
            (kp - ki*Ts/2 + 2*kd/Ts)*error_1 + (kd/Ts) * error_2;
    }

    if(!dftr && dkick){ // PID con anti derivative kick
        u1 = (2*kp + ki*Ts)/2 * error + (ki*Ts - 2*kp)/2 * error_1 + u1_1;
        u2 = kd/Ts * (dist-distAnt);
        u = u1 - u2;
    }

    if(dftr && dkick){ // PID con ambos
        u1 = (2*kp + ki*Ts)/2 * error + (ki*Ts - 2*kp)/2 * error_1 + u1_1;
        u2 = N*kd/(N*Ts+1)*(dist-distAnt) + 1/(N*Ts+1)* u2_1;
        u = u1 - u2;
    }

    if(awu)    u = u - kw*Ts/2 * (wu + wu_1); // Anti wind-up

    u0 = u;
    if(u0 > limit ) u0 = limit;
    if(u0 <-limit ) u0 = -limit;

    u2_1 = u2;
    u1_1 = u1;
    distAnt = dist;
    u_2 = u_1;
    u_1 = u;
    error_2 = error_1;
    error_1 = error;
    wu_1 = wu;
}
```

Discusión de resultados

A partir del modelo matemático teórico del sistema, se ha obtenido una serie de parámetros de diseño del controlador PID, que permiten el funcionamiento del sistema. Al comparar las simulaciones con el registro de datos real, se ve como hay algunas diferencias que mejoran con un ajuste manual, esto es debido a que nuestro modelo matemático teórico no es suficiente preciso para simular el sistema real.

Hay elementos que participan en el desempeño del control de la esfera, que no se han tenido en consideración, y que tienen su modelo matemático (servomotor, holguras, masas, inercias).

Este suceso crea la necesidad de determinar un modelo matemático experimental del sistema, que es el que más se ajusta a la realidad. Para obtener este modelo es necesario realizar ensayos en los que se hace oscilar la esfera sobre el carril pivotante durante un tiempo, registrando los datos de entrada al servomotor y la posición de la esfera.

Luego, hay que identificar el modelo matemático utilizando Matlab mediante la aplicación "System identification toolbox", a pesar de los intentos, no se han obtenido resultados concluyentes, y no se ha conseguido obtener un modelo matemático que se ajuste a la realidad, proporcionando una serie de parámetros de diseño del controlador PID que carecen de sentido.

Utilizando la estimación mediante modelo de función de transferencia se ha conseguido una aproximación máxima de 48.6%.

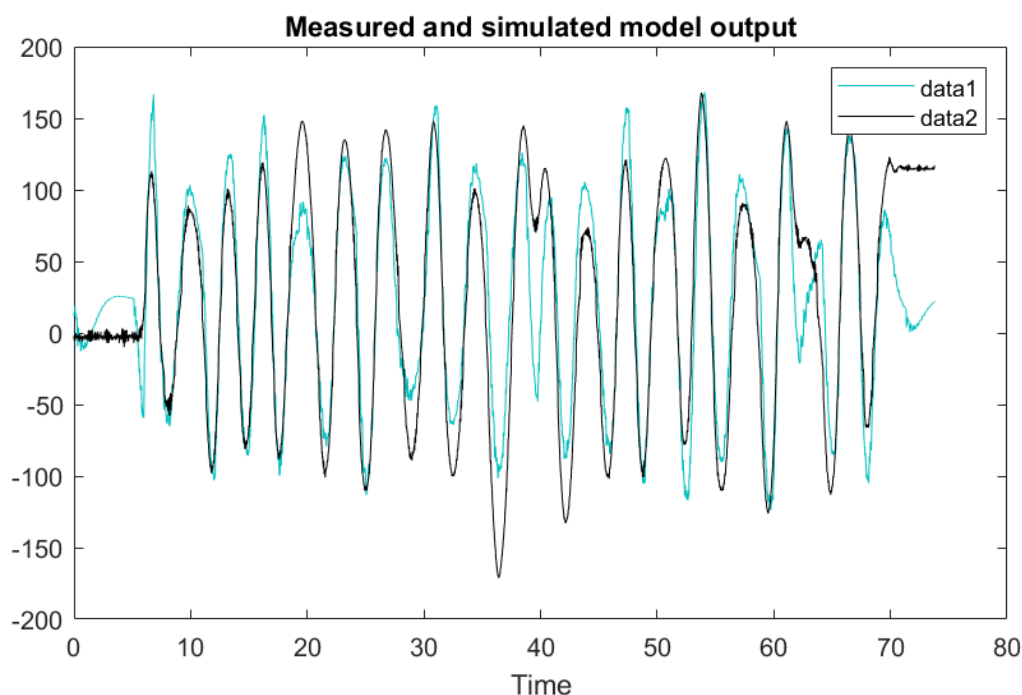


Ilustración 63: Comparación respuesta aproximada con Matlab (data1) con la respuesta real (data2)

Utilizando la estimación mediante modelo de espacio de estados se ha conseguido una aproximación máxima de 37.4%.

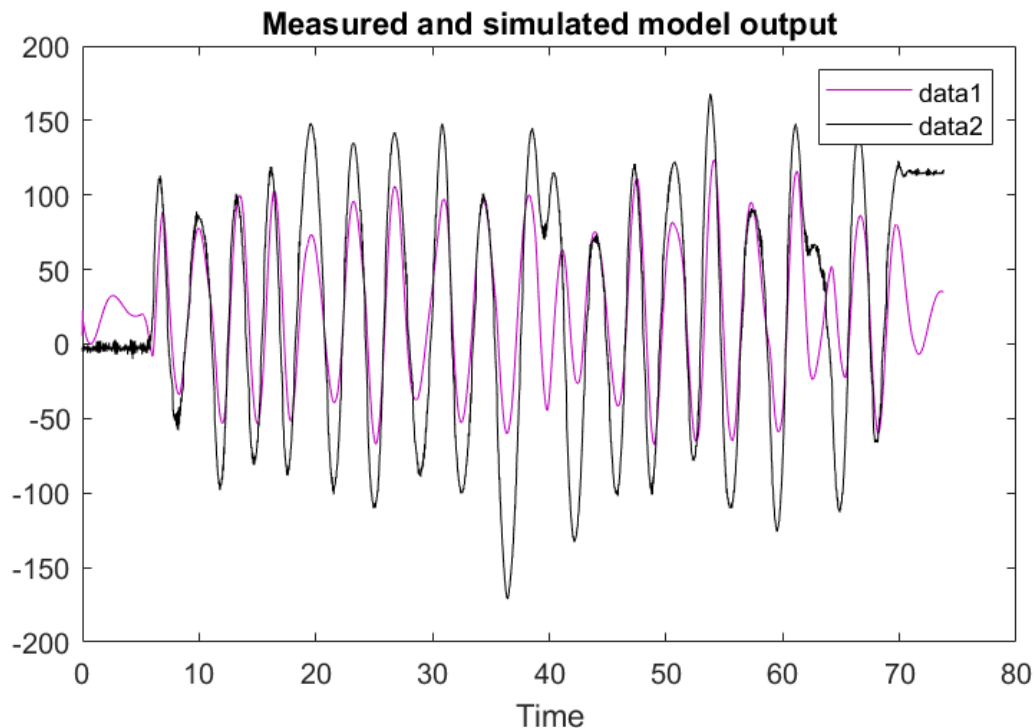


Ilustración 64: Comparación respuesta aproximada con Matlab (data1) con la respuesta real (data2)

Por ahora, la mejor opción para el diseño del controlador, es el modelo matemático teórico, realizar algún ajuste manual de los parámetros de diseño.

Por otra parte, el ruido afecta a la acción derivativa de tal manera que produce vibraciones en el carril pivotante, durante los desplazamientos a baja velocidad de la esfera o cuando se detiene. Para paliar este efecto, se ha optado por variar la frecuencia de corte del filtro en función del valor de la señal de error, permitiendo que el controlador reciba las frecuencias necesarias para el control de la posición de la esfera, en función de la distancia respecto de la señal de referencia.

Otra medida que se ha utilizado para paliar los efectos del ruido, consiste en promediar la señal de los sensores analógicos, en este caso el promedio se realiza con pocas muestras debido a las limitaciones del microcontrolador. Añadir más funciones al código del microcontrolador para procesar la señal, disminuye la capacidad de realizar todas las instrucciones dentro del periodo de muestreo de 0.02 segundos, afectando a la comunicación con la aplicación y control del servomotor.

La tira de LED tiene el mismo problema, cuando se utiliza de forma simultánea con el resto elementos del sistema, el microcontrolador no es capaz de realizar todas las instrucciones dentro de los 0.02 segundos, por lo que ha sido necesaria su desactivación para realizar los ensayos.

Conclusiones

La realización de este proyecto, ha permitido construir la base de un sistema de control discreto para uso docente. Por lo que servirá para futuras mejoras en el diseño de este sistema.

El desarrollo de una interfaz gráfica facilita el control del sistema, y en consecuencia el aprendizaje, permitiendo al estudiante visualizar y modificar las variables del sistema a tiempo real.

El ruido en los componentes electrónicos analógicos y las holguras detectadas en el mecanismo, han influido en el desarrollo del sistema de control, evitando que se pueda obtener un modelo matemático experimental que se aproxime con máxima fidelidad a la realidad, y generando un poco imprecisión en el control de la posición de la esfera.

El diseño mecánico requiere como mínimo un ligero rediseño, es necesario solventar la pequeña asimetría en la rotación del carril pivotante respecto la rotación del balancín del servomotor. También se considera la posibilidad de aumentar la resolución del sistema de control, ya que se necesita muy poca rotación del carril pivotante para el control de la esfera.

Los sensores de proximidad analógicos utilizados en nuestro sistema no ofrecen suficiente precisión. La calidad de la detección mejora al ubicar una superficie plana enfrente del sensor, esto indica que una superficie curva produce algunas perturbaciones en el receptor del sensor. Tras realizar alguna prueba experimental, se entiende que la calidad de detección del sensor se ve afectada por el color, textura y tamaño de la superficie esférica. Es conveniente considerar una modificación del tipo de sensor para mejorar la calidad de la señal.

Las imperfecciones en la superficie de la esfera y el desplazamiento de su centro de masas respecto al origen geométrico de la misma han dificultado el control, provocando pequeñas irregularidades en su desplazamiento a baja velocidad.

Tras realizar algunos ajustes manuales en los ensayos, el prototipo de nuestro sistema funciona bastante bien, considerando la calidad de los componentes de bajo coste, el sistema de control PID obtiene un resultado aceptable para el control de la esfera.

Trabajo futuro

Mejoras mecánicas

Nuestro sistema no requiere tanta rapidez en la respuesta del controlador, por lo que sería conveniente un rediseño del mecanismo para aumentar la resolución de la señal de control en detrimento del ángulo máximo de rotación del carril pivotante. Una solución válida sería modificar la ubicación del servomotor desplazándolo a un extremo del carril, asumiendo un nuevo rediseño de los elementos estructurales que implica una peor ergonomía del sistema

Otro aspecto mejorable de nuestro sistema sería eliminar la ligera asimetría que existe en la relación de grados de rotación entre el carril pivotante y el balancín del servomotor. Para ello se propone que la barra de conexión entre el balancín del servomotor y el carril, sea perpendicular al eje horizontal en la posición de reposo.

Durante los ensayos, se ha observado que la baja calidad de construcción del servomotor, ha originado un exceso de holgura en los engranajes del reductor, generando deficiencias de control del carril pivotante para ángulos muy pequeños, afectando principalmente a la acción integradora del controlador PID. Se propone sustituir este elemento por uno de mayor calidad de construcción. Esta modificación implicaría un rediseño del soporte del servomotor y de la base del mecanismo.

Mejoras electrónicas

El ruido en la señal de los sensores de proximidad analógicos, es excesivo para un control preciso de nuestro sistema, por lo que es conveniente sustituir los sensores actuales por unos de salida digital. Esta solución mejoraría la eficiencia de la programación de los sensores de proximidad utilizando un código y protocolo de comunicación más sencillo y optimizado. Además, se eliminarían los capacitadores que se utilizan como filtro de la señal analógica, ya que generan un retardo en la señal.

Mejoras modelo matemático

Se han realizado numerosos intentos para obtener un modelo matemático experimental de la planta. Desafortunadamente, los resultados no han sido exitosos para poder demostrar la teoría con la realidad. Se entiende que la falta de precisión del mecanismo, la holgura del servomotor, el ruido de los sensores de proximidad y las imperfecciones de la esfera, han influido en la calidad de los datos registrados que se utilizan para realizar la aproximación del modelo matemático experimental. Se propone implementar las mejoras mencionadas para realizar nuevos intentos de simulación del modelo matemático experimental.

Referencias bibliográficas

Didactic platform for DC motor speed and position control in Z-plane. Tomàs Pallejà Cabré, Albert Saiz Vela, Marcel Tresanchez Ribes, Javier Moreno Blanc, Jose Ribó Pablo, Francisco Clariá Sancho. Lleida, Febrero 2021.

Diseño, implementación y control de un prototipo de Ball & Beam. Guillermo Jover Ibáñez y Vicente Fermín Casanova Calvo. Valencia, Septiembre 2019.

Modeling and Analysis of a Ball and Beam System Including Impacts and Dry Friction. Conference Paper. August 2014.

Modeling and Control of the Ball and Beam Process. Marta Virseda. Department of Automatic Control, Lund Institute of technology. March 2004.

El Sistema Barra-Esfera (Ball & Beam) en un Laboratorio de Robótica Luis Ángel Blas Sánchez, Oscar Martínez Fuentes, Juan Javier Montesinos García, César Ulises Solís Cervantes y Alejandro J Malo Tamayo. Control Automático, Cinvestav. México, D.F. Abril 2015.

Desarrollo de un sistema Ball and Beam, para implementar estrategias de control mediante Labview, Oscar Alirio Obando Correa, Hector Romero Brand y Luis Ángel Silva. Bolivia, 2010.

Diseño y control de un sistema Ball and Beam con realimentación visual con Raspberry-Pi. Anabel Araujo Puerta, Carlos Vivas Venegas e Ignacio Alvarado Aldea. Sevilla, 2018.

Control PID de Barra y Bola con Arduino. Ángel Espeso. Agosto 2015.

Control de un sistema Bola Balancín. Maestre Rodríguez, Héctor Manuel y Francisco Javier García Ruiz. Valladolid 2019.

Plataformas de bajo coste para experimentación en control. Daniel Sánchez Haro y Sebastián Tornil Sin. Mayo 2017.

<https://ctms.engin.umich.edu/CTMS/index.php?example=BallBeam§ion=SystemModeling>

<http://bibing.us.es/proyectos/abreproy/3953/fichero/Memoria+PFC%252F04-CAPITULO+3.pdf>

Ingeniería electromecánica, sistemas de control automático, Ball & Beam. William Rubio. Universidad Antonio Nariño.

Diseño, construcción y estabilización de un sistema Bola-Balancín. B.A. López, N.C. Castillo, L.A. Cabrera, L. Vite y R. Villafuerte. México, 2018.