



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



---

# Diseño, implementación y control de un prototipo de Ball & Beam

---

Autor: Guillermo Jover Ibáñez

Dirigido por: Vicente Fermín Casanova Calvo

Titulación: Máster en Ingeniería Mecatrónica.

Fecha: Valencia, Septiembre, 2019

## Resumen

Este Trabajo Final de Máster, consiste en el diseño, implementación, control y montaje de un prototipo de Ball & Beam para el departamento de control, para poder servir de modelo educativo entre los demás estudiantes.

Para ello, en el diseño se han tomado ciertas decisiones del tipo de modelo teniendo en cuenta los materiales que teníamos disponibles. Antes de realizar todo el montaje se ha decidido empleando el actual diseño simulándolo y aplicándole un sistema de control.

Al realizar la simulación se ha tratado de simular con las condiciones y variables reales del sistema real para obtener algo más cercano a la realidad.

Una vez realizadas las simulaciones se ha procedido con la implementación real y montaje tanto de la parte hardware como del software.

Finalmente se han aplicado sistemas de control al sistema real ajustándolo hasta que el sistema estaba en su máximo.

## Summary

This Final Master's Project consists in the design, implementation, control and assembly of a Ball & Beam prototype for the control department, in order to serve as an educational model among the other students.

For this, in the design certain decisions of the type of model have been taken taking into account the materials we had available. Before carrying out the entire assembly, it was decided using the current design simulating it and applying a control system. When performing the simulation, we tried to simulate with the real conditions and variables of the real system to obtain something closer to reality. Once the simulations have been carried out, the actual implementation and assembly of both the hardware and software part has been carried out. Finally, control systems have been applied to the real system by adjusting it until the system was at its maximum.

## Dedicatoria

Este proyecto se lo dedico a todas las personas que me han apoyado a lo largo de los años desde mis familiares más cercanos a amigos, compañeros, profesores, tutores séase en mis ideas como en mis decisiones.

También quiero dar las gracias a aquellas personas que no han participado activamente, pero sus consejos con respecto al proyecto me han ayudado mucho y por ello les doy las gracias a todos ellos.

Sobre todo, quiero dar las gracias a Polymer Char por la oportunidad que me han brindado y ha hecho posible parte del desarrollo del trabajo.

# Índice

Resumen.....	2
Summary .....	3
Dedicatoria.....	4
Índice de Ilustraciones.....	6
1. Introducción .....	8
1.1. Objetivo.....	9
2. Modelo Teórico.....	10
2.1. Modelando por el método de Newton .....	10
2.2. Representación del Espacio de estados. ....	11
2.3. Parámetros del sistema. ....	12
3. Simulación.....	13
3.1. Modelado del sistema.....	13
3.2. Control de la bola .....	15
3.3. Resultados obtenidos.....	16
4. Implementación física .....	26
4.1. Montaje .....	26
4.1.1. Componentes mecánicos.....	26
4.1.2. Componentes electrónicos.....	27
4.2. Simulación previa de la pieza conectada al motor y la viga.....	29
4.3. Funcionamiento y Caracterización del sensor de posición.....	31
4.4. Control empleado .....	36
4.5. Resultados obtenidos.....	38
5. Conclusiones .....	41
6. Presupuesto .....	42
6.1. Coste material.....	42
6.2. Coste laboral.....	43
7. Pliego de Condiciones.....	44
7.1. Condiciones generales.....	44
7.1.1. Objeto.....	44
7.1.2. Normativa y Reglamento Vigente.....	44
7.1.3. Pliego de condiciones facultativas .....	44
7.1.3.1. Programación de Trabajos.....	45
7.1.3.2. Ejecución Defectuosa y Modificaciones .....	45
7.1.3.3. Recepción Definitiva.....	45
7.1.4. Descripción General del Montaje.....	45

7.1.5.	Pliego de Condiciones Económicas .....	45
7.1.5.1.	Fianzas.....	45
7.1.5.2.	Composición de Precios .....	45
7.1.5.3.	Mejoras.....	45
7.2.	Condiciones Particulares .....	46
7.2.1.	Condiciones de los materiales .....	46
7.2.1.1.	Cableado Eléctrico .....	46
7.2.1.2.	Cables de Alimentación .....	46
7.2.2.	Condiciones de Funcionamiento del sistema .....	46
7.2.2.1.	Limitaciones por hardware.....	46
7.2.2.2.	Limitaciones por Software .....	46
7.2.3.	Condiciones de la ejecución.....	46
7.2.3.1.	Condiciones de la instalación.....	46
7.2.3.2.	Condiciones de Garantía .....	46
7.2.4.	Prueba de Servicio.....	47
8.	Bibliografía .....	48
9.	Anexos.....	49
9.1.	Código Arduino.....	49
9.2.	Planos .....	53

## Índice de Ilustraciones

Ilustración 1:	Descomposición del diagrama de bloques.....	8
Ilustración 2:	Ensamblaje Bola y Viga.....	13
Ilustración 3:	Modelo Real Bola y Viga.....	14
Ilustración 4:	Modelo Básico. ....	14
Ilustración 5:	Modelo real. ....	15
Ilustración 6:	Sistema del control del Ball & Beam.....	16
Ilustración 7:	Seguimiento simulación 1.....	16
Ilustración 8:	Acción de Control de la simulación 1.....	17
Ilustración 9:	Seguimiento Simulación 2.....	17
Ilustración 10:	Acción de control simulación 2.....	18
Ilustración 11:	Seguimiento Simulación 3.....	19
Ilustración 12:	Acción de control simulación 3.....	19
Ilustración 13:	Seguimiento Simulación 4.....	20
Ilustración 14:	Acción de control simulación 4.....	20
Ilustración 15:	Seguimiento Simulación 5.....	21
Ilustración 16:	Acción de control simulación 5.....	22
Ilustración 17:	Seguimiento Simulación 6.....	22
Ilustración 18:	Acción de control simulación 6.....	23

Ilustración 19: Seguimiento Simulación 7. ....	23
Ilustración 20: Acción de control simulación 7.....	24
Ilustración 21: Seguimiento Simulación 8. ....	24
Ilustración 22: Acción de control simulación 8.....	25
Ilustración 23: Perfil en L de aluminio.....	26
Ilustración 24: Pieza Eje motor.....	26
Ilustración 25: Motor Pololu 37D 70:1. ....	27
Ilustración 26: Sensor de posición, traza de grafito.....	27
Ilustración 27: Tarjeta Arduino DUE.....	28
Ilustración 28: Portapilas 12V DC.....	28
Ilustración 29: Equipo que emplearemos en la caracterización del sensor de posición. ....	31
Ilustración 30: Método de medición y caracterización del sensor. ....	32
Ilustración 31: Prototipo Bola y Viga. ....	36
Ilustración 32: Flujograma Arduino.....	37
Ilustración 33: Bola en 0 (Origen).....	38
Ilustración 34: Bola desplazándose.....	39
Ilustración 35: Bola desplazándose II. ....	40

# 1. Introducción

El sistema Bola y Viga (más conocido comúnmente como Ball & Beam), es uno de los problemas más conocidos en los sistemas de control.

Consiste principalmente en el control de posición de una esfera colocada sobre una barra o viga, en la que se variara el ángulo de inclinación de esta con el fin de estabilizar en una posición la bola. Por ello en este sistema se considerará el ángulo del eje del motor como una entrada y la posición de la bola como salida.

Este es un sistema con un grado de libertad ya que solo controlaremos el ángulo de inclinación de la viga. Por ello emplearemos un motor de continua al que se le aplicara un control PID, Para determinar la posición de la bola emplearemos el principio de un potenciómetro analógico para capturar su posición.

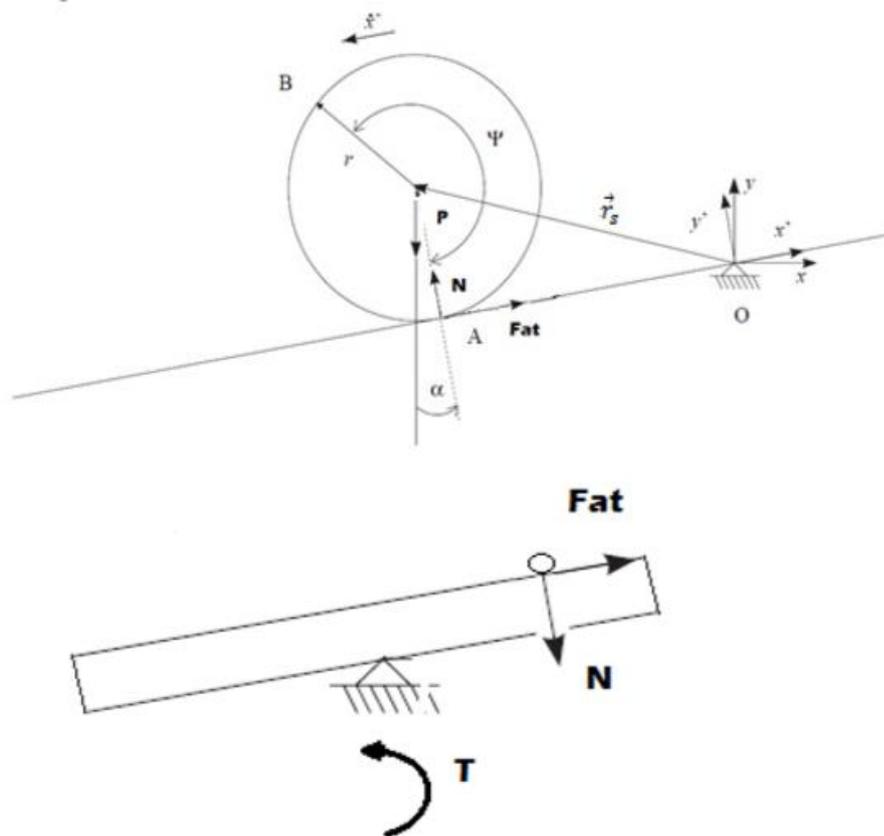


Ilustración 1: Decomposición del diagrama de bloques.

Primeramente, se obtendrá y analizará el modelo teórico del sistema con el cual trabajaremos y usaremos sus parámetros como referencia, y una vez obtenido comenzaremos con las simulaciones del sistema en Matlab/Simulink, en esta sección analizaremos paso por paso el comportamiento del sistema con respecto a distintas señales de referencia y diferencias con respecto a sistemas más rápidos y reales. Una vez obtenidos resultados concluyentes se procederán con la sección de implementación real y montaje con la que se irán realizando pruebas que contrastaran con respecto a los resultados simulados y se comentarán las posibles mejoras.

## 1.1. Objetivo

Este proyecto consistirá principalmente en el diseño de un modelo de simulación del sistema Ball & Beam, la implementación de un control PID, la caracterización del sensor de posición y finalmente la construcción del prototipo incluyendo las pruebas efectuadas a este.

Primeramente, simularemos el sistema con el que vamos a trabajar para distintas referencias de movimiento del motor y por tanto como reacciona la bola a tales, estas simulaciones se realizaran para observar la diferencias en el sistema al emplear distintas frecuencias o una mayor zona muerta.

El objetivo final será que, en el sistema real, cuando la bola se mueva lejos del centro la viga trate de reorientarla hacia el centro.

## 2. Modelo Teórico

Para el modelo matemático del sistema, emplearemos el método de Newton, el cual mediante separando los distintos cuerpos, aplicaremos un equilibrio de fuerzas en cada uno de ellos.

### 2.1. Modelando por el método de Newton

En un marco de referencia no inercial, la ley de Newton deberá modificarse adecuadamente, utilizando por tanto el concepto de derivado total (O derivado covariante) el cual explica las variaciones en el propio marco de referencia y el efecto en el movimiento total.

La matriz de transformación de coordenadas entre el marco de referencia fijo del haz y el marco de inercia es:

$$R = \begin{bmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{bmatrix}$$

La matriz de rotación relacionando ambos márgenes (expresada en el margen de la viga) está dada por  $\Omega = R - 1\dot{R}$ , Su expresión en forma matricial es la siguiente:

$$\Omega = \begin{bmatrix} 0 & -\dot{\alpha} \\ \dot{\alpha} & 0 \end{bmatrix}$$

Esta matriz representa la variación del marco móvil expresada en la base del marco móvil. La derivada covariante (total) se puede escribir como:

$$D_t = \frac{\partial}{\partial t} + \vec{\Omega}x = \frac{\partial}{\partial t} + \Omega$$

Donde  $\vec{\Omega}$  es el vector equivalente a  $\Omega$ . La ley de Newton para el cuerpo rotatorio se puede escribir simplemente como:

$$\vec{T} = D_t \cdot \vec{H}$$

y la ecuación de Newton para movimiento lineal es:

$$\vec{F} = mD_t D_t \vec{S}' = m \left( \frac{\partial}{\partial t} + \vec{\Omega}x \right) \left( \frac{\partial}{\partial t} + \vec{\Omega}x \right) \vec{S}'$$

donde  $S'$  es la posición del centro de masa de la pelota en el marco de referencia móvil (viga). Esta es la segunda ley de Newton covariante, que es válida para cualquier marco de referencia. Después de alguna manipulación, la segunda ley de Newton se puede escribir como:

$$m\ddot{S}' = F - m\Omega^2 S' - m\dot{\Omega}S' - 2m\Omega\dot{S}'$$

donde  $m$  es la masa de la bola y  $F$  es la fuerza total aplicada en la bola (expresada en el marco de referencia de la viga). La matriz  $\Omega = R - 1\dot{R}$  representa una velocidad diferente, que es la velocidad angular del marco no inercial expresado en el marco inercial. El cambio entre el marco no inercial y el inercial (o viceversa) es una transformación de similitud, también conocida como transformación adjunta. Considere ahora la descomposición del sistema en dos cuerpos independientes (en contacto por fricción y fuerzas normales), como se ha mostrado anteriormente en la imagen de descomposición del sistema.

Aplicando la segunda ley de Newton en el baricentro de bolas (en el marco de referencia móvil), tenemos la ecuación:

$$m\dot{S}' = [F_{at} - mg \cdot \sin\alpha \quad N - mg \cdot \cos\alpha]^t - m \cdot \begin{bmatrix} 0 & -\dot{\alpha} \\ \dot{\alpha} & 0 \end{bmatrix} \cdot \begin{bmatrix} 0 & -\dot{\alpha} \\ \dot{\alpha} & 0 \end{bmatrix} \cdot S' - m \cdot \begin{bmatrix} 0 & -\ddot{\alpha} \\ \ddot{\alpha} & 0 \end{bmatrix} S' - 2m \cdot \begin{bmatrix} 0 & -\dot{\alpha} \\ \dot{\alpha} & 0 \end{bmatrix} \cdot S'$$

En cualquier situación, la posición relativa del baricentro de la pelota es siempre constante y  $y' = r$ , de modo que la  $S'$  puede escribirse como:

$$S' = [x' \quad r]^t, \dot{S}' = [\dot{x}' \quad 0]^t, \ddot{S}' = [\ddot{x}' \quad 0]^t$$

Así que tenemos finalmente:

$$[m\ddot{x}, 0]^t = [F_{at} - mg \cdot \sin\alpha, N - mg \cdot \cos\alpha]^t + [m\dot{\alpha}^2 x', m\dot{\alpha}^2 r]^t - [-m\ddot{\alpha} r, m\ddot{\alpha} x']^t - 2m \cdot [0, \ddot{\alpha} x']^t$$

lo que implica que la fuerza normal debe ser:

$$N = m \cdot (g \cdot \cos\alpha - r \cdot \dot{\alpha}^2 + \ddot{\alpha} \cdot x' + 2\dot{x}' \dot{\alpha})$$

Y el movimiento del centro de masa es:

$$m\ddot{x}' = F_{at} - mg \cdot \sin\alpha + m\dot{\alpha}^2 x' + m\ddot{\alpha} r$$

Para determinar el movimiento rotacional relativo de la bola, descrito por el ángulo  $\Psi$ , tenemos que aplicar la segunda Ley de Newton rotacional, en el segundo marco de referencia no inercial de la bola, que se expresa como:

$$\vec{T} = \frac{d\vec{H}}{dt} = \frac{\partial \vec{H}}{\partial t} + (\vec{\Omega} + \vec{\Omega}') \cdot \vec{H}$$

donde  $\vec{H}$  es el momento angular,  $\vec{T}$  es el par total aplicado en el sistema, y  $\vec{\Omega} + \vec{\Omega}'$  es el vector de velocidad angular para la pelota, que está relacionado con la matriz  $\Omega + \Omega'$  por un isomorfismo de Algebra de Lie. De hecho, es posible representar todas las cantidades de vectores como matrices antisimétricas, con el producto del vector sustituido por el Corchete de Lie. Como los vectores  $\vec{H}$  y  $\vec{\Omega}$  tienen las mismas direcciones, lo que significa que su producto vectorial es nulo, la ecuación se reduce a:

$$\vec{T} = \frac{\partial \vec{H}}{\partial t} = I_b \cdot (\ddot{\alpha} + \ddot{\Psi}) = r \cdot F_{at}$$

Considerando ahora el par aplicado por el motor en la viga y la fuerza normal de contacto que se opone al movimiento, tenemos:

$$I_W \cdot \ddot{\alpha} = lu(t) - x'N - bl\dot{\alpha} - k_1 l\alpha - k_2 l\alpha^3$$

en el que  $u(t)$  es el control de entrada y todos los demás pares se oponen al movimiento, donde  $k_1$  y  $k_2$  son el coeficiente elástico de la correa de goma (que transmiten la fuerza a la viga),  $b$  es el coeficiente de fricción viscosa de la viga y  $I_W$  es el momento de inercia de la viga.

## 2.2. Representación del Espacio de estados.

La representación del espacio de estado del sistema es: definir las variables de estado como

$$x_1 = x', x_2 = \dot{x}', x_3 = \alpha, x_4 = \dot{\alpha}, x_5 = \Psi, x_6 = \dot{\Psi}$$

y escribiendo la fuerza normal que no depende de las aceleraciones, tenemos:

$$N = \frac{1}{1 + mx_1^2/I_w} \cdot [mg \cdot \text{Cos}x_3 - mrx_4^2 + \frac{ml}{I_w}x_1u(t) + 2x_2x_4 - \frac{blm}{I_w}x_1x_4 - \frac{k_1lm}{I_w}x_1x_3 - \frac{k_2lm}{I_w}x_1x_3^3]$$

y la representación del espacio de estados es:

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= \frac{1}{m}F_{at} - g\text{Sin}x_3 + x_1x_4^2 + r\left(\frac{lu(t) - x_1N - blx_4 - k_1lx_3 - k_2lx_3^3}{I_w}\right) \\ \dot{x}_3 &= x_4 \\ \dot{x}_4 &= \frac{lu(t) - x_1N - blx_4 - k_1lx_3 - k_2lx_3^3}{I_w} \\ \dot{x}_5 &= x_6 \\ \dot{x}_6 &= \frac{r}{I_b}F_{at} - \left(\frac{lu(t) - x_1N - blx_4 - k_1lx_3 - k_2lx_3^3}{I_w}\right) \end{aligned}$$

### 2.3. Parámetros del sistema.

Finalmente, los parámetros del Ball & Beam que utilizaremos para el modelo teórico son los siguientes:

$\alpha$	Coordenada del Angulo de la Viga.
L	Longitud de la Viga.
m	Masa de la bola.
R	Radio de la bola.
J	Momento de inercia de la bola.
G	Aceleración gravitacional.
x	Posición de la bola.

Sin embargo, cabe destacar que ya que emplearemos Matlab/Simulink muchos de los parámetros aquí mostrados se calcularán automáticamente, por lo que este modelo teórico no se empleará en el proyecto.

### 3. Simulación

Para la simulación del comportamiento del sistema Bola y Viga, emplearemos Simscape Multibody un plugin de Matlab Simulink, el cual nos proporcionara el entorno de simulación con respecto al tipo de piezas que tenemos y su forma de esta forma podremos efectuar la simulación de la manera más precisa con respecto a nuestro sistema. Este entorno es capaz de resolver las ecuaciones de movimiento para un sistema complejo y nos permite la importación de los archivos Step y todo lo referente a su geometría y condiciones físicas.

A diferencia de otros programas, con Simscape multibody también podemos trabajar en tiempo real por lo que somos capaces de ver un sistema más cercano al real. Lo cual favorece en gran medida nuestra aplicación.

A parte, también emplearemos la librería de Contact Forces, para dar más realismo al sistema y establecer las fuerzas de contacto presentes tanto en el choque inicial y el rozamiento durante el movimiento.

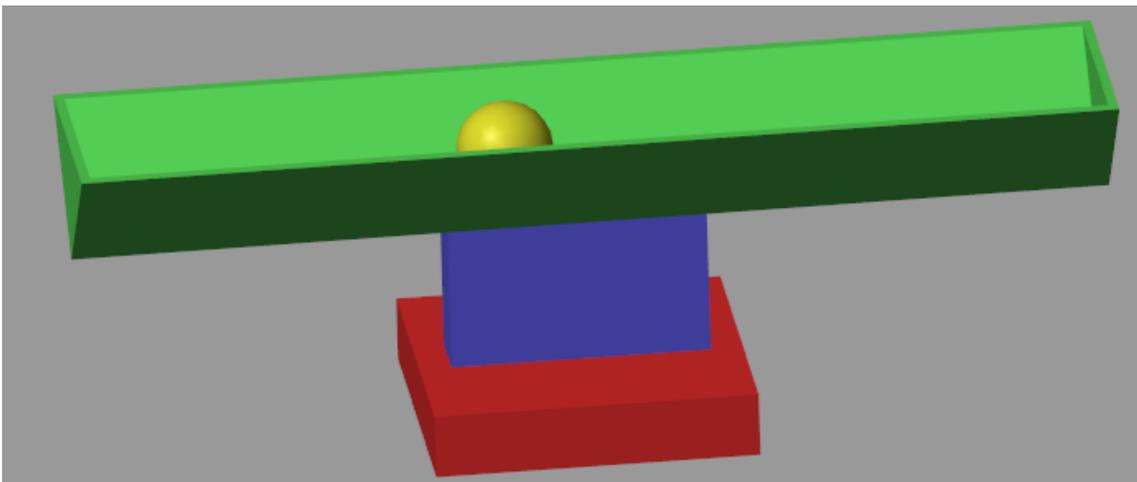
#### 3.1. Modelado del sistema

Primero, antes de nada, comenzaremos con el diseño de las piezas, ya que muchas de ellas tienen bastantes detalles no podremos diseñarlas desde el propio Multiscape multibody de Matlab, por ello, emplearemos SolidWorks para el diseño de las piezas y luego la importación a formato Step debido a los requerimientos de Matlab/Simulink. Para piezas más simples como es el caso de la bola se puede emplear Simscape multibody para su diseño.

El conjunto Bola y Viga está principalmente compuesto de:

- Viga.
- Eje Motor.
- Motor/Base.
- Bola.

A continuación, mostramos el primer ensamblaje de Bola y Viga que se ha simulado, seguido del ensamblaje con respecto al sistema real que también se ha simulado.



*Ilustración 2: Ensamblaje Bola y Viga.*

Evidentemente utilizar este modelo de simulación puede servir como base, pero requeriríamos simular el sistema de acuerdo al que vamos a emplear en la implementación real. Por ello la siguiente imagen muestra el sistema real que también vamos a simular.

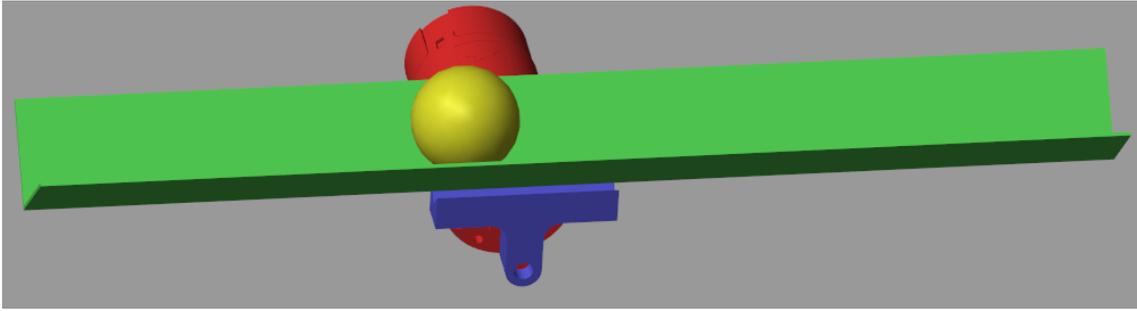


Ilustración 3: Modelo Real Bola y Viga.

El modelo real contiene menos piezas que el básico que contribuirán al ahorro del coste computacional de la simulación.

El ensamblaje funciona de la siguiente manera: el motor anclado al suelo produce una rotación en la pieza eje para regular el ángulo de la viga, el cual afectara a la posición de la bola.

Evidentemente este mecanismo es bastante simple, No obstante, hemos decidido simular el sistema dos veces: una con un modelo básico y otra con un modelo más cercano a la realidad para comprobar su comportamiento.

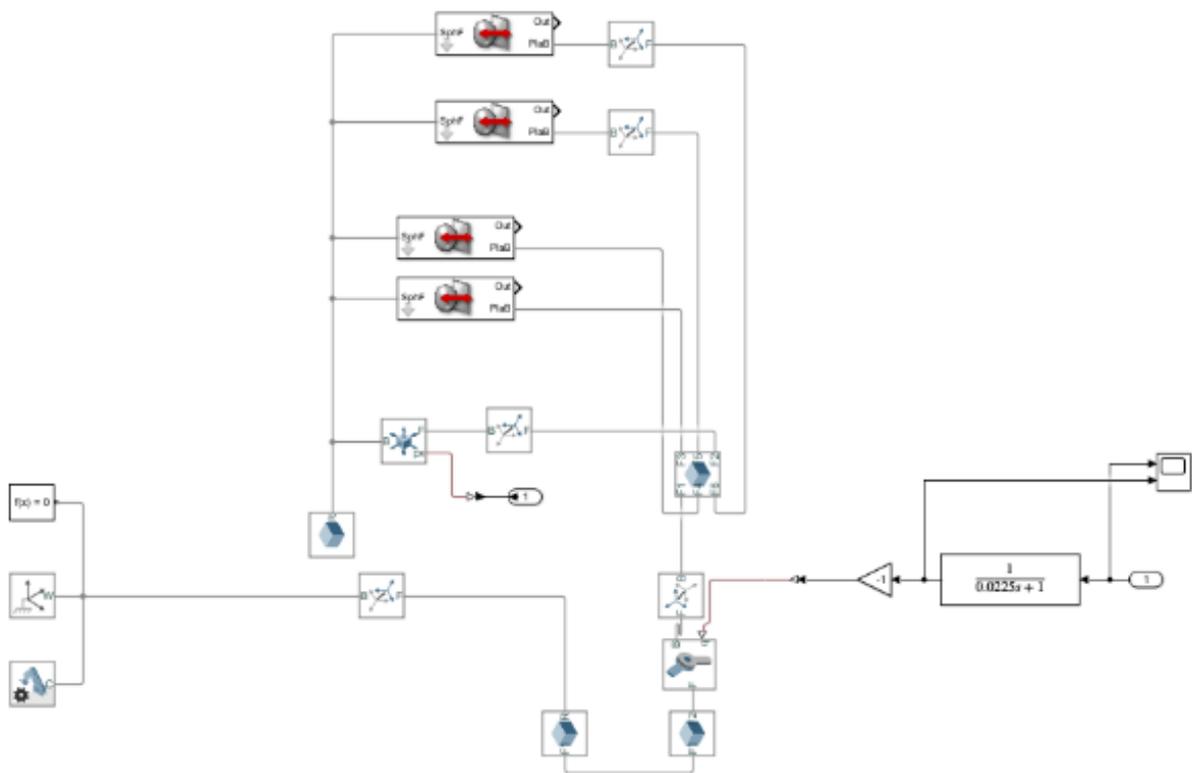


Ilustración 4: Modelo Básico.

Como podemos apreciar en este sistema claramente tardara mas en renderizar ya que aparte de tener mas piezas, existen mas interacciones del sistema con la bola.

Y finalmente mostramos el bloque del sistema de simulación con el que luego se usara como base para el sistema real.

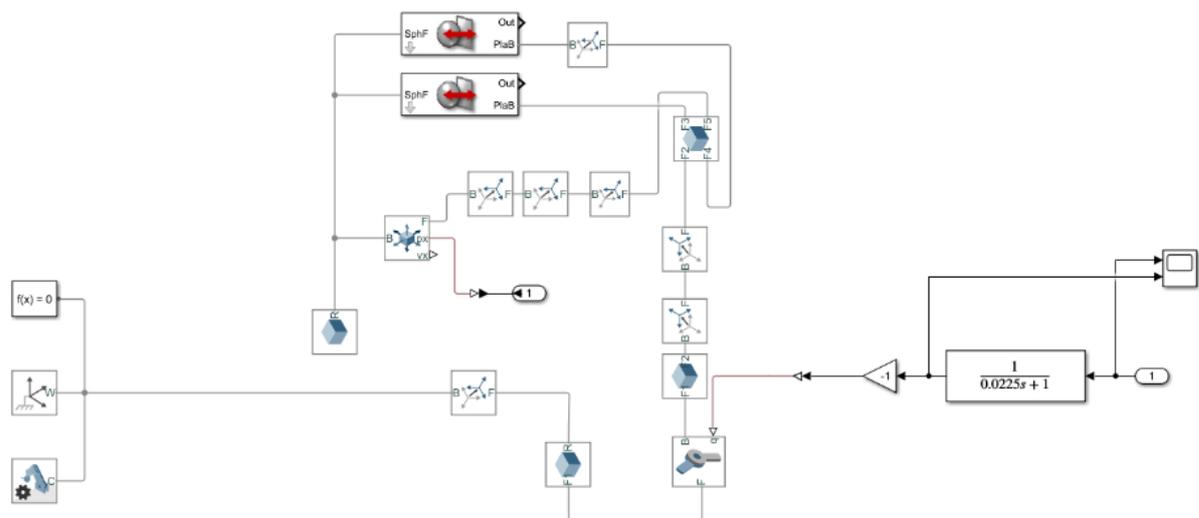


Ilustración 5: Modelo real.

Ambos sistemas tienen una cosa en común y esa es el bloque que simula la componente dinámica del motor y con el que alimentamos el eje de revolución. Esta componente está principalmente compuesta de un regulador PD que se encarga de ajustar los parámetros para que el sistema de control de posición se comporte como uno de primer orden y pueda realizar la simulación con el respectivo control de posición. Para prevenir las sobreoscilaciones si las hubiera se podría ajustar la acción integral del regulador.

Como se puede observar el sistema sería prácticamente el mismo tanto para el primer modelo como el segundo. También podemos observar las herramientas empleadas en el mecanismo de simulación. Ya que nuestro mecanismo tiene un grado de libertad solo hemos empleado una junta de revolución para que se pueda mover en el eje y una junta cartesiana para la unión de la bola con respecto a la viga empleada.

También podemos observar que se ha introducido para ambos modelos la constante del motor para hacer esta simulación más realista.

Una vez finalizado el modelo de SimScape que vamos a emplear pasaremos a la siguiente etapa.

### 3.2. Control de la bola

Primeramente, consistirá en controlar la viga para conseguir un buen control. Para ello se tratará de controlar los ángulos de la plataforma para poder posicionar la bola en las posiciones deseadas, para ello emplearemos un regulador PID para el control de la bola.

El sistema previamente diseñado irá en un subsistema al cual le introduciremos la referencia como entrada y nos dará como salida la posición de la bola.

Después le añadiremos una referencia con la forma de onda que deseamos para ver si el control sigue la trayectoria designada por la referencia, la cual observaremos a través de los scopes del sistema.

El sistema de control está representado en la siguiente imagen.

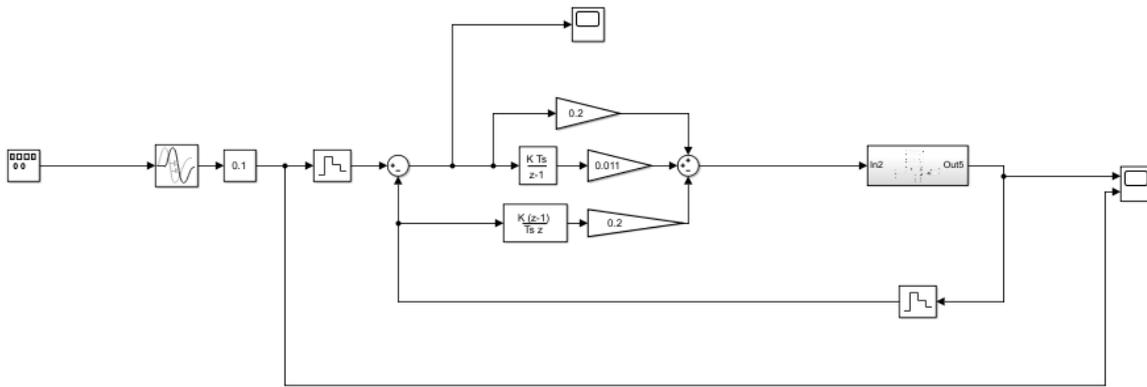


Ilustración 6: Sistema del control del Ball & Beam.

A continuación, proseguiremos con el control de la bola. Para ello, implementaremos en el sistema una realimentación obtenida del valor de la posición de la bola en la simulación. De esta forma el sistema reaccionara a la posición de la bola y tratara de colocar la bola en las posiciones deseadas de acuerdo con las señales de referencia que le enviamos.

### 3.3. Resultados obtenidos

Una vez realizadas las simulaciones con respecto a nuestro sistema comentaremos los resultados obtenidos. Dada la poca diferencia entre el sistema inicial de simulación y la final hemos decidido comentar unicamente los resultados finales ya que son los que mas se aproximan con el modelo real.

Para los resultados la señal azul mostrara la referencia y la señal roja sera la posición de la bola.

Para empezar hemos realizado la simulación del sistema con una frecuencia de 0,05 Hz en forma senoidal y cuadrada. Los resultados se muestran a continuación. Hemos empleado en los valores del regulador PID de la bola los siguientes valores.

$K_p$	$K_i$	$K_d$
0.2	0.011	0.2

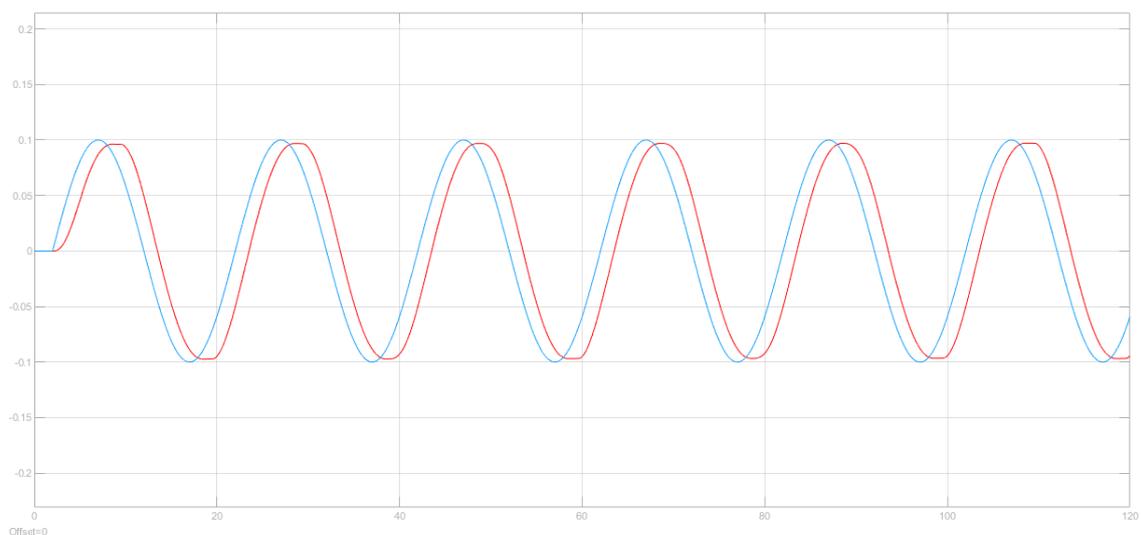
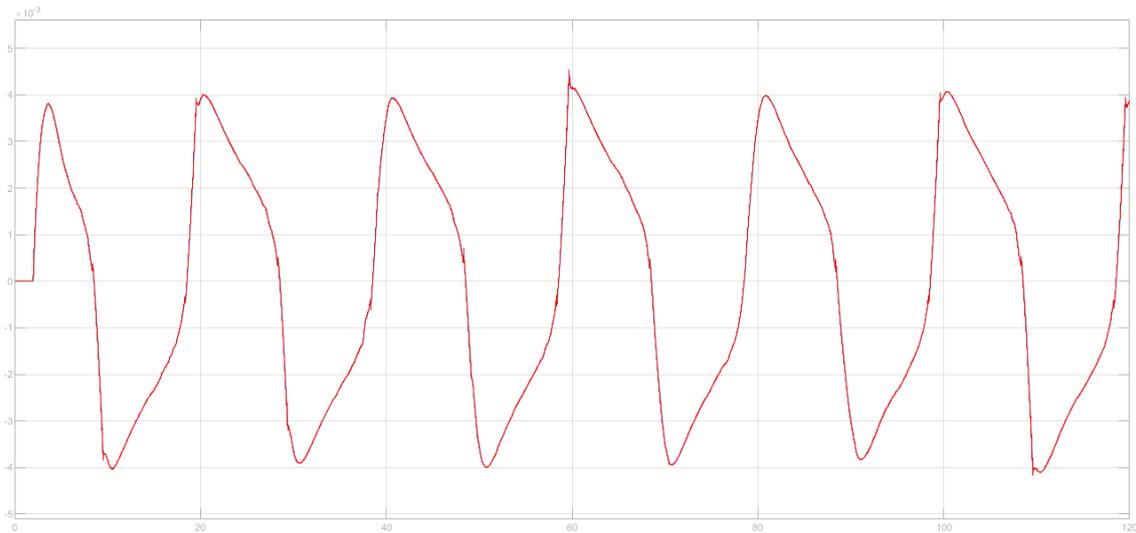


Ilustración 7: Seguimiento simulación 1.

<b>Simulación 1</b>	<b>Frecuencia (Hz)</b>	<b>Amplitud</b>	<b>Fase (rad)</b>
<i>Referencia</i>	0,05	0,1	0
<i>Posición Bola</i>	0,05	0,097	1,91

Podemos observar que en la simulación existe cierto desfase que es típico de bastantes motores de continua debido a sus características y su transitorio.

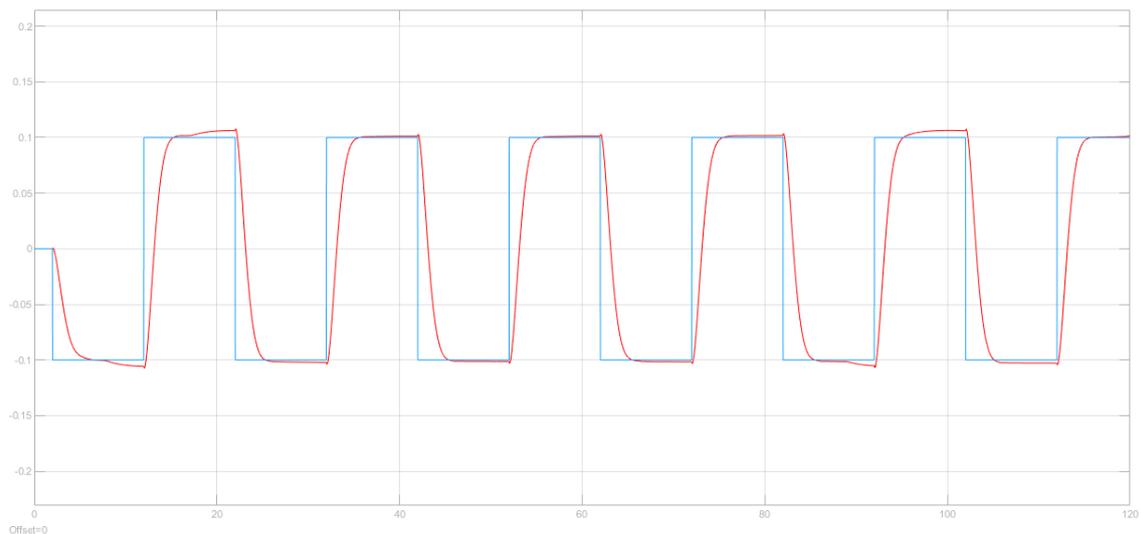
Para la simulación también podemos observar la acción de control del motor en la siguiente imagen.



*Ilustración 8: Acción de Control de la simulación 1.*

Como podemos observar el sistema no tiene grandes dificultades para seguir señales senoidales.

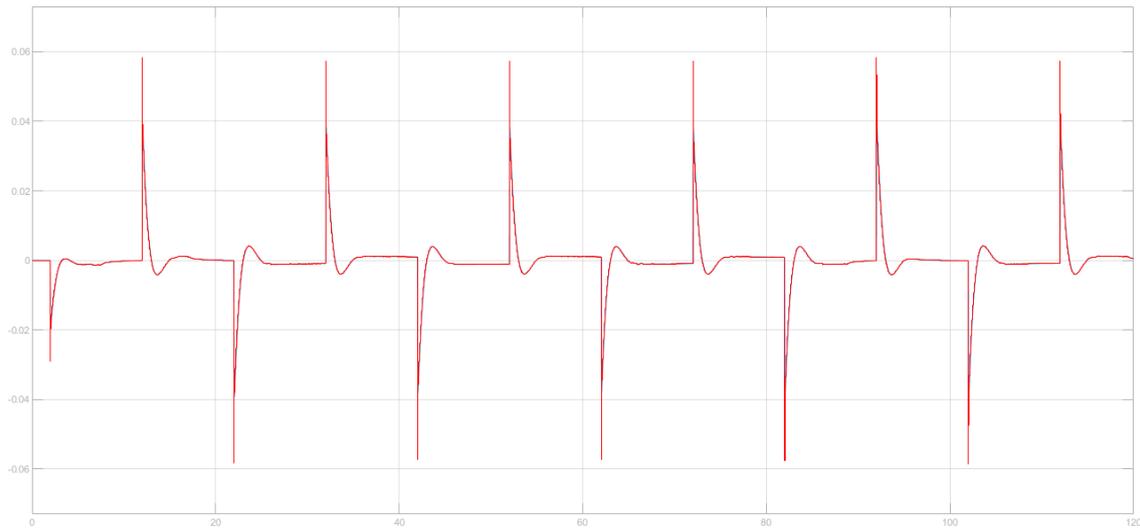
Seguidamente simularemos en este caso el comportamiento del sistema con respecto a una señal cuadrada con las mismas condiciones que la anterior simulación.



*Ilustración 9: Seguimiento Simulación 2.*

<i>Simulación 2</i>	<b>Frecuencia (Hz)</b>	<b>Amplitud</b>	<b>Tiempo de establecimiento (98%)</b>
<i>Referencia</i>	0,05	0,1	-
<i>Posición Bola</i>	0,05	0,104	3,49

Como se puede apreciar en la simulación con forma de onda senoidal el sistema parece tener un retardo el cual es comun ya que los motores no suelen actuar al instante y siempre suele haber algún tipo de delay. No obstante vemos que el sistema sigue la trayectoria de forma perfecta tanto en forma senoidal como de onda cuadrada. Seguidamente analizaremos la acción de control del motor.



*Ilustración 10: Acción de control simulación 2.*

A pesar de que no se pueda ver con claridad, el pico de la acción de control generada por el motor cuando realiza el control de posición es bastante mayor a la generada por la referencia. Esto se debe a las características de las señales cuadradas en nuestro sistema y motor, ya que le es mas sencillo seguir una referencia senoidal a una cuadrada.

Una vez realizadas, probaremos a añadirle al sistema una zona muerta para simular las condiciones del motor de manera mas real y analizaremos los resultados del mismo. Para ello emplearemos el mismo regulador que en el sistema anterior y con las mismas frecuencias. Aparte seleccionaremos de zona muerta desde -0,01 a 0,01.

Primeramente volveremos a simular con la referencia senoidal, la señal con zona muerta estara representada de color naranja y podremos compararla con las otras dos señales, los resultados se muestran a continuación.

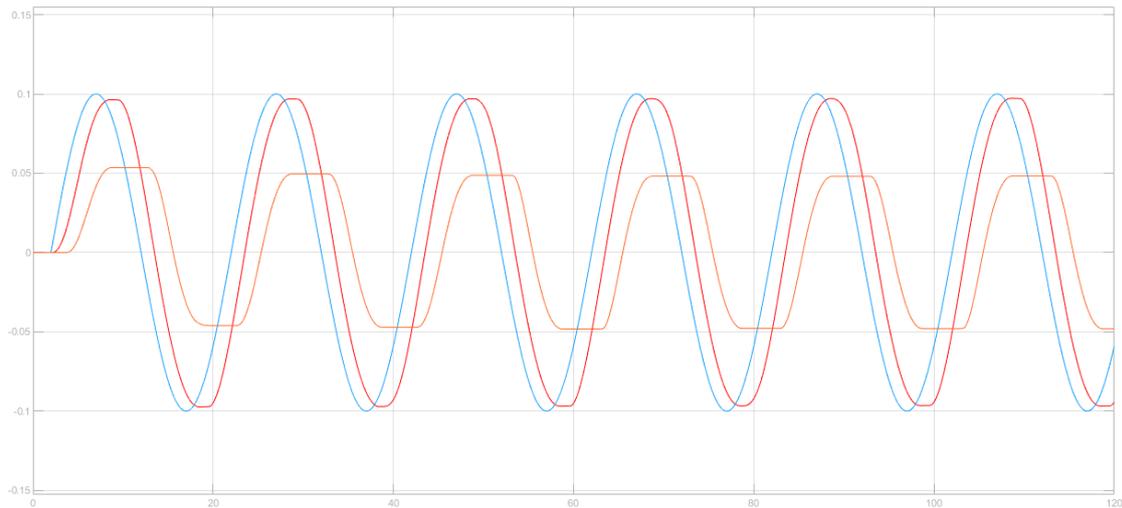


Ilustración 11: Seguimiento Simulación 3.

<b>Simulación 3</b>	<b>Frecuencia (Hz)</b>	<b>Amplitud</b>	<b>Fase (rad)</b>
Referencia	0,05	0,1	0
Posición Bola	0,05	0,097	1,91
Posición Bola Zm	0,05	0,04837	3,47

Podemos observar que el sistema, al aplicarle zona muerta la amplitud disminuye considerablemente y se genera aun mas desfase, también podemos observar que los picos de la señal con zona muerta se aplanan generando por tanto una zona recta.



Ilustración 12: Acción de control simulación 3.

Podemos observar que a diferencia de la acción de control de la simulación 1 esta contiene tramos que se sitúan en el 0 debido a la zona muerta del motor.

A continuación, proseguiremos con la señal cuadrada.

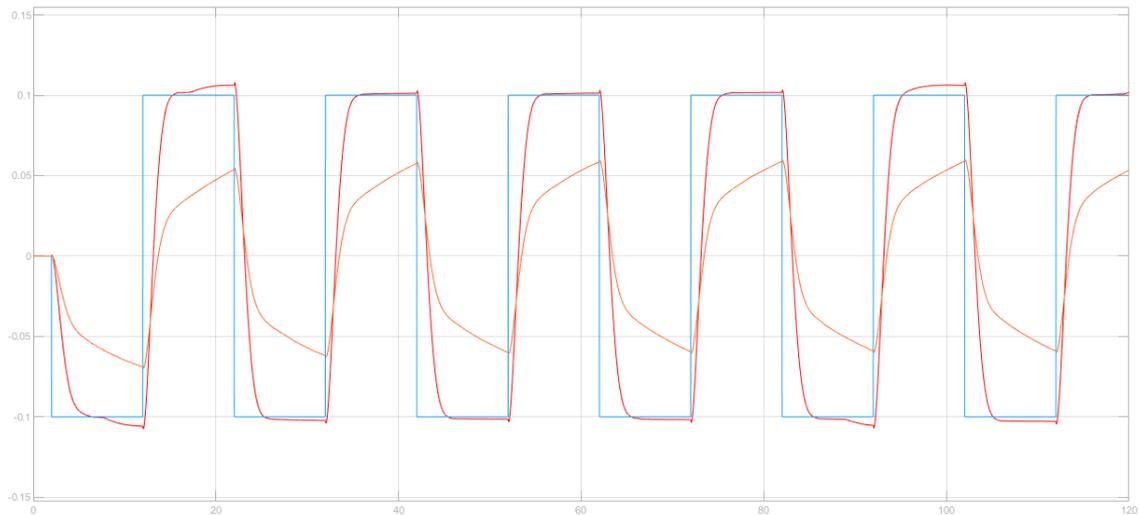


Ilustración 13: Seguimiento Simulación 4.

<b>Simulación 4</b>	<b>Frecuencia (Hz)</b>	<b>Amplitud</b>	<b>Tiempo de establecimiento (98%)</b>
<i>Referencia</i>	0,05	0,1	-
<i>Posición Bola</i>	0,05	0,104	3,49
<i>Posición Bola Zm</i>	0,05	0,061	9,64

La señal obtenida por parte de la onda cuadrada nos da a entender que al existir una zona muerta el sistema tardara considerablemente mas en llegar a establecerse y tambien afectara a la propia amplitud de la señal.



Ilustración 14: Acción de control simulación 4.

Como podemos observar en esta acción de control los tramos se separan mas del 0 con respecto a la acción de control de la simulación 2, lo que nos muestra que si se aplica una zona muerta en una señal cuadrada en nuestro sistema en vez de pasar por 0 como en la senoidal se alejara un poco del 0.

Como podemos apreciar en los resultados, existen diferencias visibles con respecto a la zona muerta de -0.01 a 0.01 con el periodo de muestreo de  $T= 0,001$  s. Esto se observa viendo

principalmente que en la señal con zona muerta (naranja) tiene un mayor error de posición que afecta a los máximos que tiene la señal con el mismo regulador, para compensarlo sería necesario que la acción integral aumente para que el efecto de la zona muerta fuera inferior.

A pesar de ello la señal que probablemente emplearíamos si consideráramos que va a existir zona muerta en el sistema emplearíamos señales senoidales que se deforman considerablemente menos que las señales cuadradas.

Finalmente incrementaremos la frecuencia de la señal de entrada para comprobar el efecto que tiene sobre el sistema, al aumentarla comprobamos que primero debemos volver a modificar nuestro regulador en especial la  $K_p$  ya que el sistema no puede seguir la trayectoria del movimiento por ello una vez modificado este valor podemos observar las diferencias con respecto a la primera simulación. Para cada señal se muestra la configuración del regulador PID que se va a emplear.

	$K_p$	$K_i$	$K_d$
Señal senoidal	0.65	0.011	0.2
Señal cuadrada	0.55	0.11	0.2

Los resultados mostrados son con respecto a una frecuencia de 0.5 Hz.

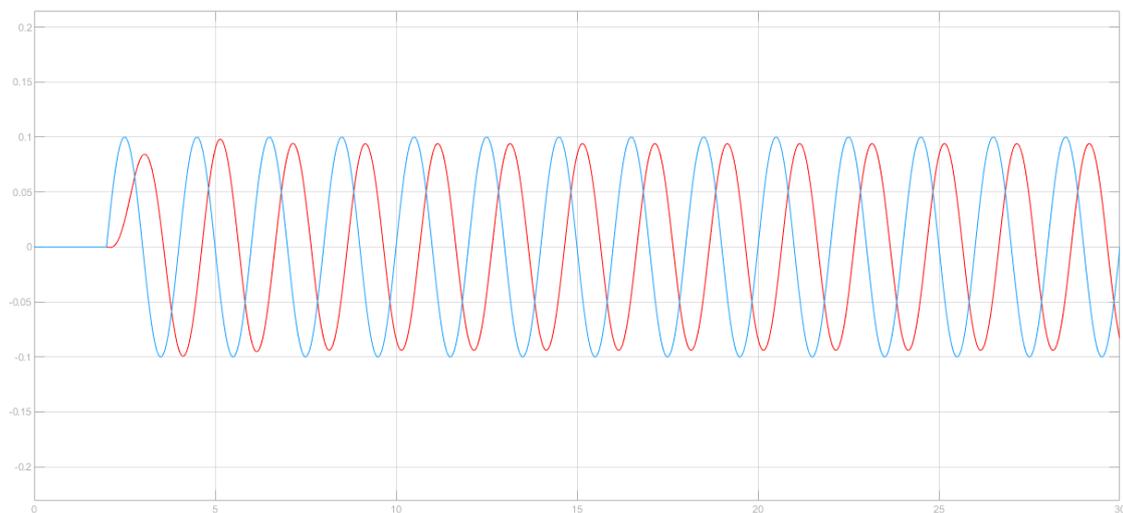


Ilustración 15: Seguimiento Simulación 5.

<b>Simulación 5</b>	<b>Frecuencia (Hz)</b>	<b>Amplitud</b>	<b>Fase (rad)</b>
Referencia	0,5	0,1	0
Posición Bola	0,5	0,094	0,655

Podemos observar de los resultados que al haber aumentado la frecuencia del sistema, ha afectado un poco a la amplitud de la señal pero ha reducido considerablemente el desfase existente. Es decir que en el caso real debería actuar con mayor rapidez.

Proseguimos con la acción control de la simulación.



Ilustración 16: Acción de control simulación 5.

Observando la acción de control vemos que el sistema es capaz de seguir la forma senoidal sin ningun problema.

A continuación simularemos de nueva la señal cuadrada con las mismas condiciones.

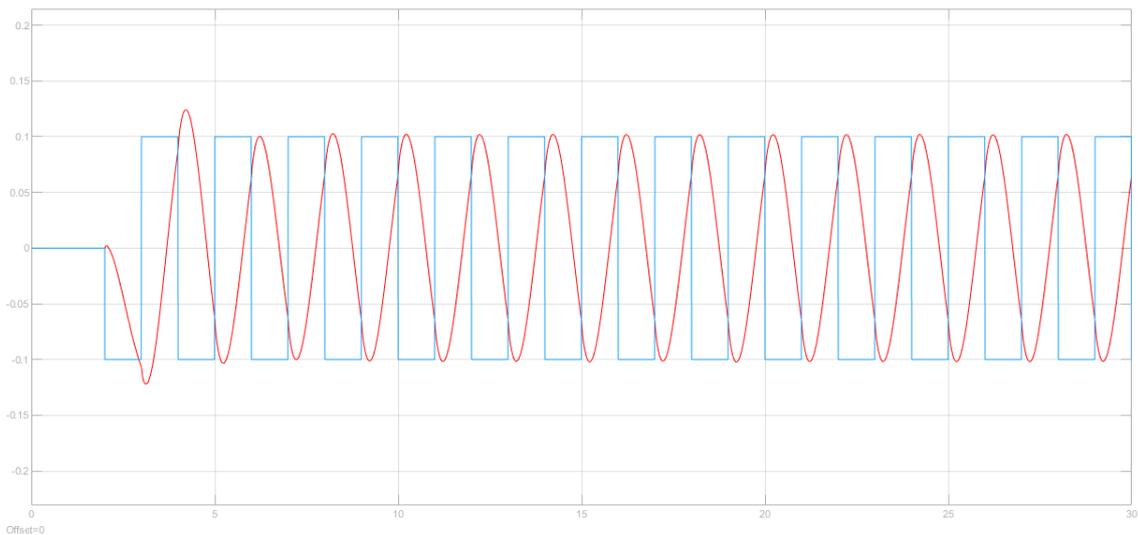


Ilustración 17: Seguimiento Simulación 6.

<b>Simulación 6</b>	<b>Frecuencia (Hz)</b>	<b>Amplitud</b>	<b>Tiempo de establecimiento (98%)</b>
Referencia	0,5	0,1	-
Posición Bola	0,5	0,1	No llega a establecerse.

Primero podemos apreciar en la onda cuadrada que le cuesta tanto que no llega a establecerse aunque llegue a seguir la trayectoria, es decir, no le da tiempo a detener la bola en los tramos rectos y por tanto continua a la siguiente posición. Analizaremos por tanto la acción de control de esta simulación para ver por que no llega a establecerse.

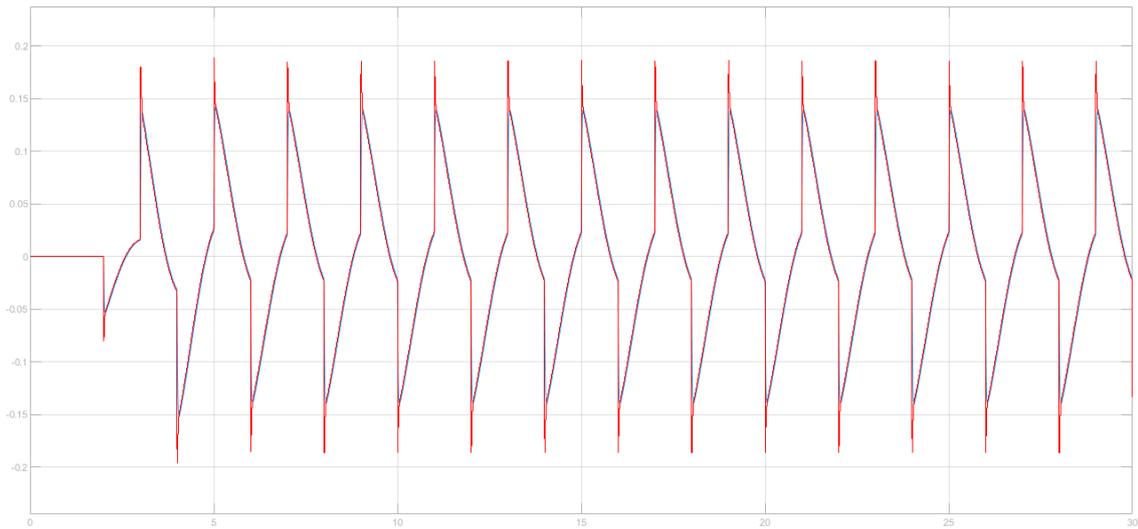


Ilustración 18: Acción de control simulación 6.

Como podemos observar en la acción de control el pico de la posición de la bola se sitúa bastante por encima de la señal de referencia demostrando que, ante una señal de referencia cuadrada no es la más adecuada para nuestro sistema.

Por ello, debido a que la onda senoidal sigue comportándose de la misma forma con un respectivo aumento de la frecuencia de la señal se podrá emplear ya que será capaz de seguir mejor una referencia senoidal a una cuadrada.

Finalmente, hemos tratado de simular el sistema con una mayor zona muerta para ver el efecto que causaría y este sería el resultado de la simulación para la señal senoidal.

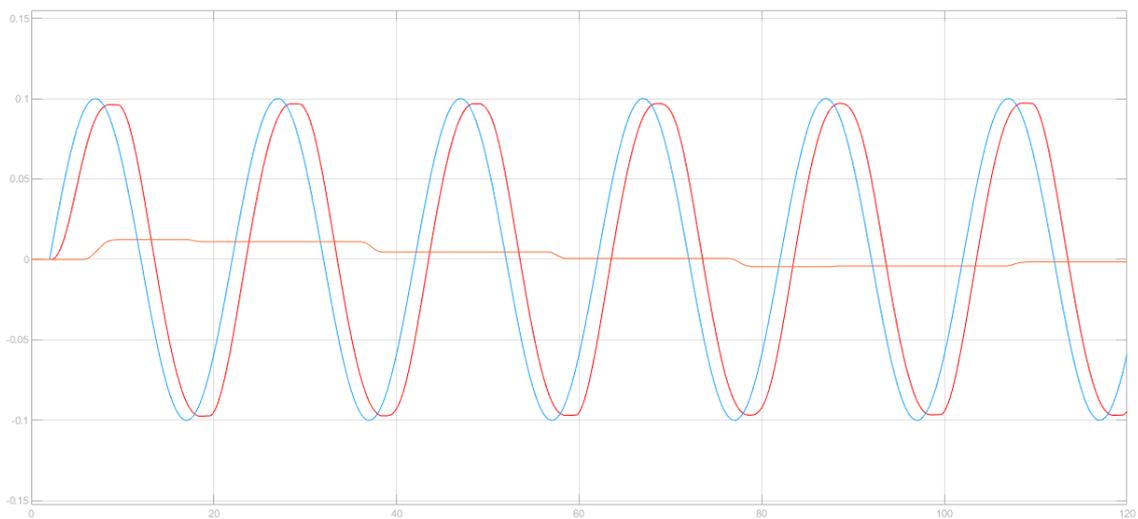


Ilustración 19: Seguimiento Simulación 7.

<b>Simulación 7</b>	<b>Frecuencia (Hz)</b>	<b>Amplitud</b>	<b>Fase (rad)</b>
<i>Referencia</i>	0,05	0,1	0
<i>Posición Bola</i>	0,05	0,097	1,91
<i>Posición Bola Zm</i>	0,05	0,012	Desconocido

Viendo los resultados, podemos asumir que con una mayor zona muerta las señales suelen volverse más difíciles de controlar, en este caso vemos que la amplitud ha disminuido considerablemente y somos incapaces de determinar el desfase empleando el regulador que hemos empleado con la zona muerta de -0.01 a 0.01.

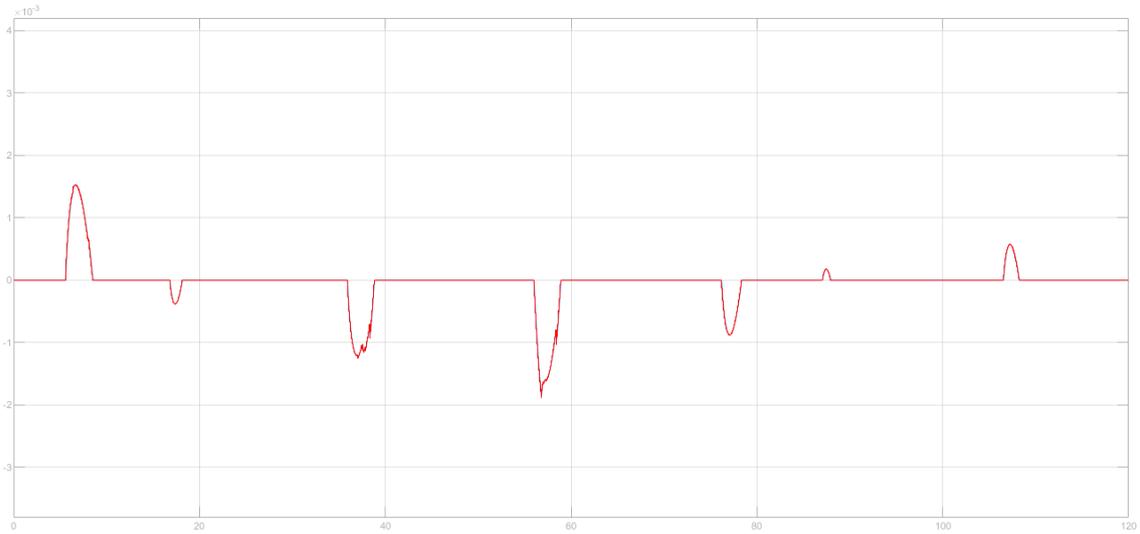


Ilustración 20: Acción de control simulación 7.

Podemos apreciar que el sistema no es capaz de seguir la referencia con una zona muerta demasiado alta, a pesar de ser una señal senoidal.

A continuación, proseguimos con la señal cuadrada para comprobar su comportamiento.

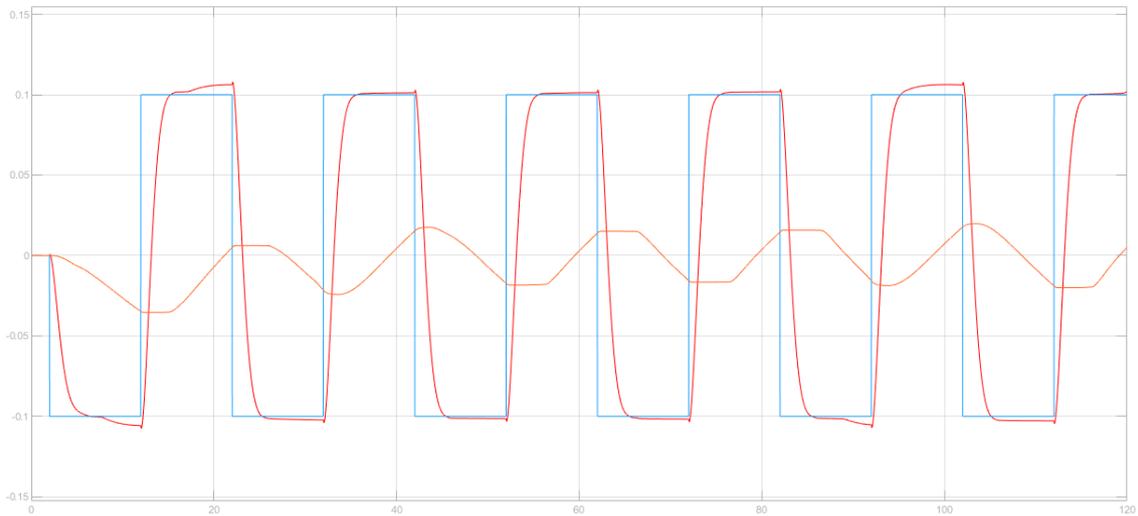
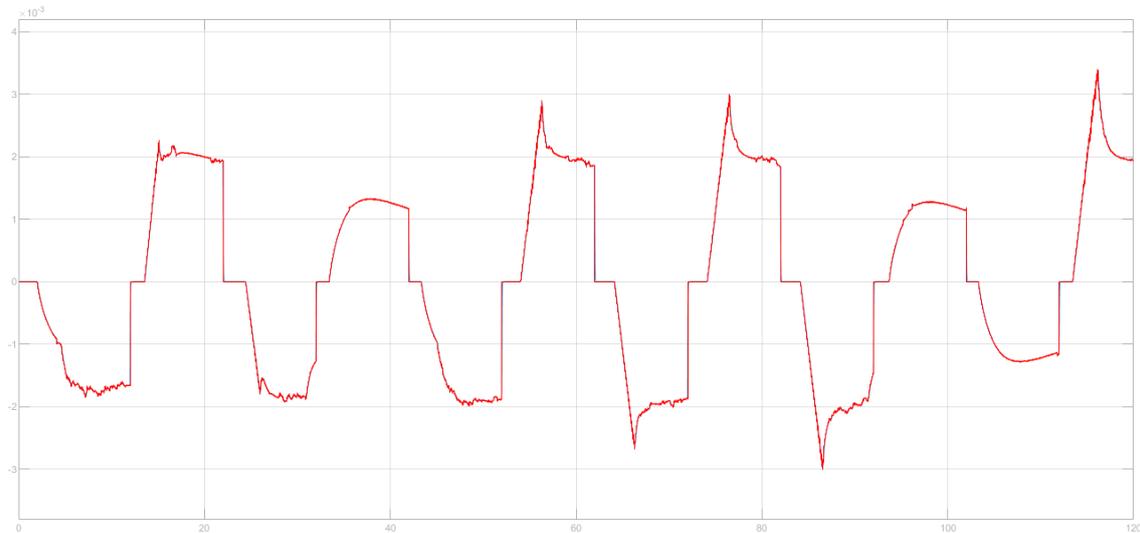


Ilustración 21: Seguimiento Simulación 8.

<b>Simulación 8</b>	<b>Frecuencia (Hz)</b>	<b>Amplitud</b>	<b>Tiempo de establecimiento (98%)</b>
Referencia	0,05	0,1	-
Posición Bola	0,05	0,104	3,49
Posición Bola Zm	0,05	0,018	No llega a establecerse dentro de la referencia

Como podemos observar, tanto para la señal senoidal como la cuadrada como en este caso obtenemos una señal que no deseáramos para nuestro sistema ya que es incapaz de seguir la referencia.



*Ilustración 22: Acción de control simulación 8.*

Podemos ver que en la acción de control el comportamiento es inadecuado para nuestros requerimientos.

Podemos concluir que ante una mayor zona muerta nuestro sistema se vería gravemente afectado dando lugar a comportamientos que impedirían cierta precisión en la aplicación.

No obstante vemos que el sistema en las simulaciones responde adecuadamente a las señales de referencia senoidales cuando no hay o hay poca zona muerta, lo cual es bastante satisfactorio.

## 4. Implementación física

Esta sección tiene como objeto la implementación real del sistema Bola y Viga, previamente diseñado en Matlab/Simulink.

### 4.1. Montaje

Ahora procederemos con la explicación de cada uno de los materiales que emplearemos en la construcción, así como de las medidas que hemos empleado para aproximarnos de la mejor manera al sistema simulado.

#### 4.1.1. Componentes mecánicos

##### - Viga

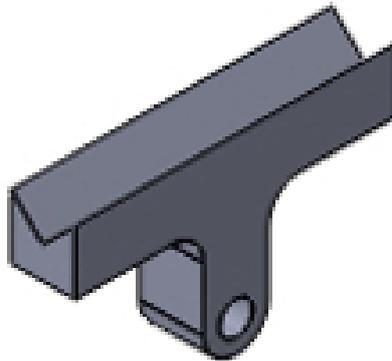
Para la Viga, obtendremos un perfil en L desde el cual trabajaremos. El perfil estará compuesto de aluminio y será de longitud de 30 cm para darle espacio a la bola para desplazarse y poder conducir electricidad, ya que esta pieza estará en contacto con la bola y el potenciómetro.



*Ilustración 23: Perfil en L de aluminio.*

##### - Pieza Eje motor

Esta pieza será la misma que la modelada por SolidWorks, por tanto, será necesaria realizar una impresión en 3D. Como extra caracterizaremos esta pieza mecánicamente con el fin de conocer sus propiedades mecánicas y saber sus límites estructurales. Esta pieza será la encargada de sostener la Viga y por tanto deberá estar correctamente fijada a esta, para ello emplearemos un adhesivo epoxi de 2 componentes para que esta sección este bien fija.



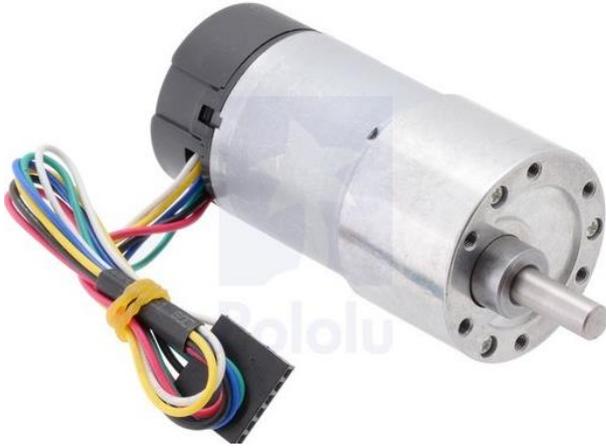
*Ilustración 24: Pieza Eje motor.*

#### 4.1.2. Componentes electrónicos

- Motor

El motor que emplearemos para esta aplicación será un Pololu 37Dx70L mm, de 12V DC, con la caja de cambios de metal 70:1 y un encoder con una resolución de 64 CPR.

Sus propiedades mecánicas serán de masa de 204g con un par de 14 Kg/cm.



*Ilustración 25: Motor Pololu 37D 70:1.*

- Sensor de posición/ Potenciómetro

En este proyecto trataremos de emplear un potenciómetro como sensor analógico de posición.

Primero se intentó obtener una pista impresa para actuar como potenciómetro y por ello tratamos de ponernos en contacto con algunos fabricantes, sin éxito.

No obstante, descubrimos que se podía emplear una traza de grafito (dibujada gracias a un lápiz B2 en papel) como sensor debido a sus propiedades. Por ello procedimos con la caracterización del sensor para ver si se podía emplear.

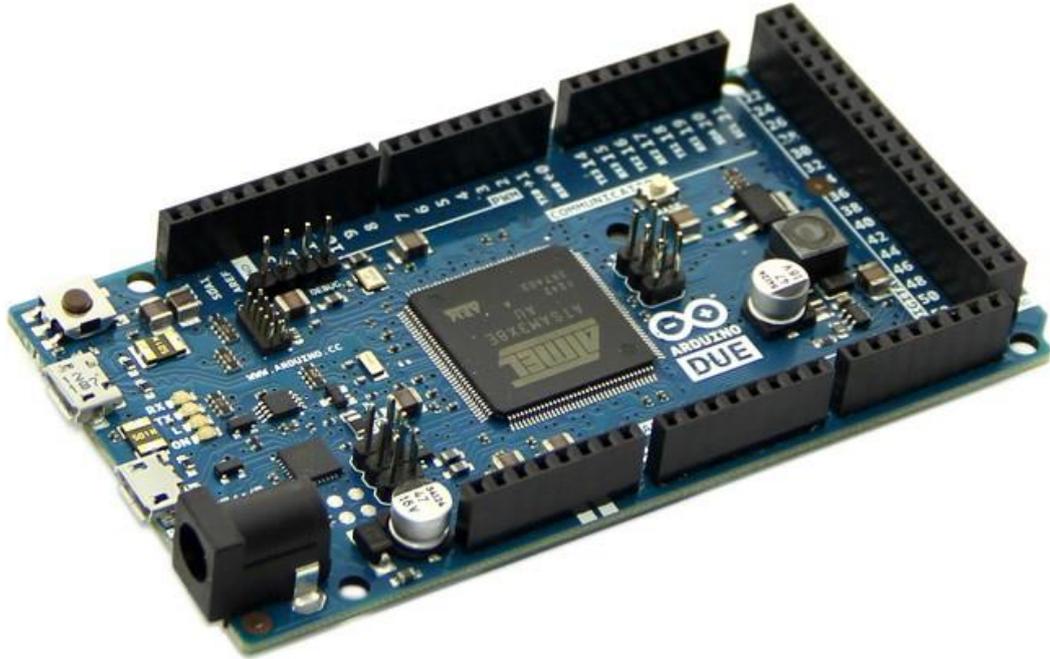


*Ilustración 26: Sensor de posición, traza de grafito.*

- Tarjeta Arduino Due

Esta tarjeta se seleccionó debido al motor que empleábamos para sincronizar tanto el encoder como darle la suficiente potencia para que este ejecutara todo el movimiento de la mejor manera. A parte la tarjeta Arduino DUE tiene una mayor velocidad de procesamiento lo cual para sistemas

de control es imprescindible, en especial si se requiere control en tiempo real. (84 MHz la DUE a diferencia de otros modelos que tienen 16 MHz).



*Ilustración 27: Tarjeta Arduino DUE.*

- Batería Externa

Para el funcionamiento del motor será preciso de una fuente de alimentación externa que pueda alimentar sobre los 12V DC.



*Ilustración 28: Portapilas 12V DC*

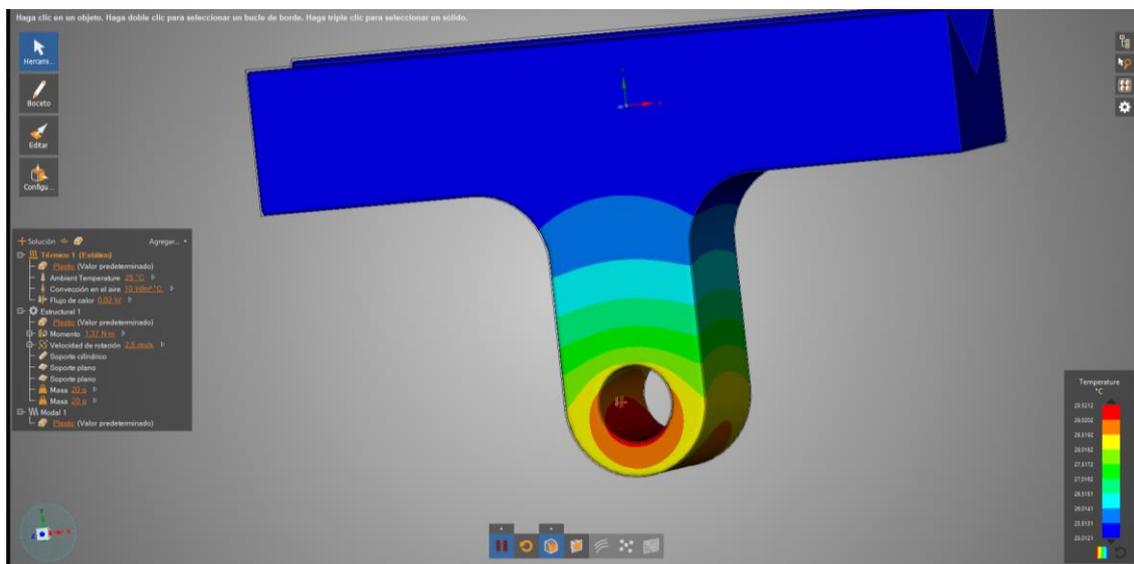
## 4.2. Simulación previa de la pieza conectada al motor y la viga

A pesar de trabajar a baja tensión, es importante conocer el comportamiento de las piezas que están sometidas a largos esfuerzos. Ya que de esta forma no solo podremos conocer su comportamiento, sino también su duración y con qué debemos ser cautos.

Para ello emplearemos ANSYS Discovery Live el cual nos permitirá simular y comprobar el estado de nuestra pieza. Realizaremos 3 tipos de simulaciones:

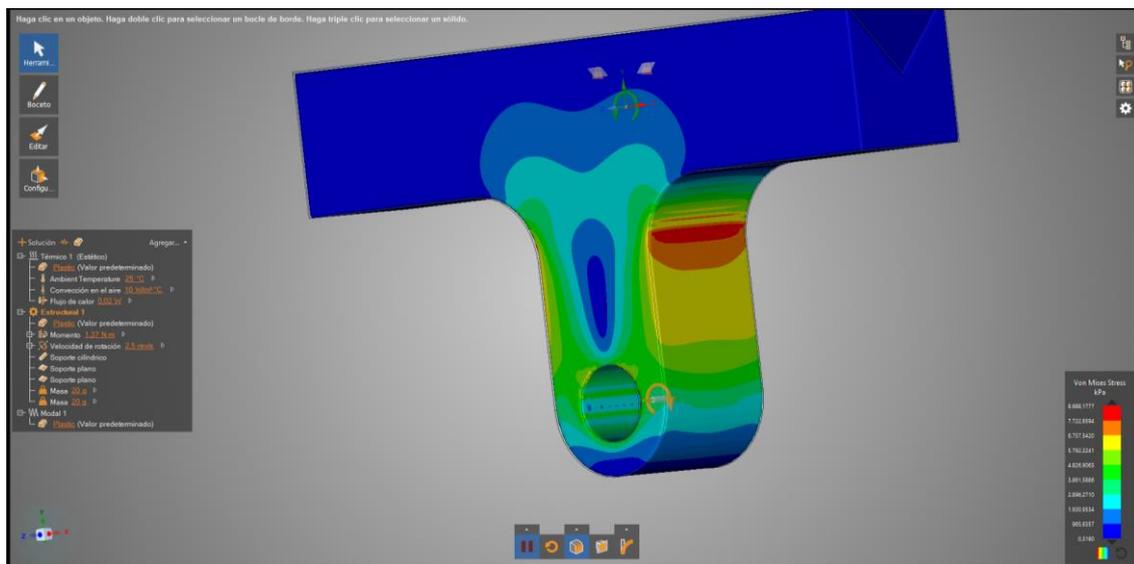
- Temperatura

Para esta simulación emplearemos un 1% de flujo de calor del valor de potencia del motor transferido por calor a la pieza debido a que no es una conducción perfecta y que el calor generado por el motor se transmite alrededor del eje



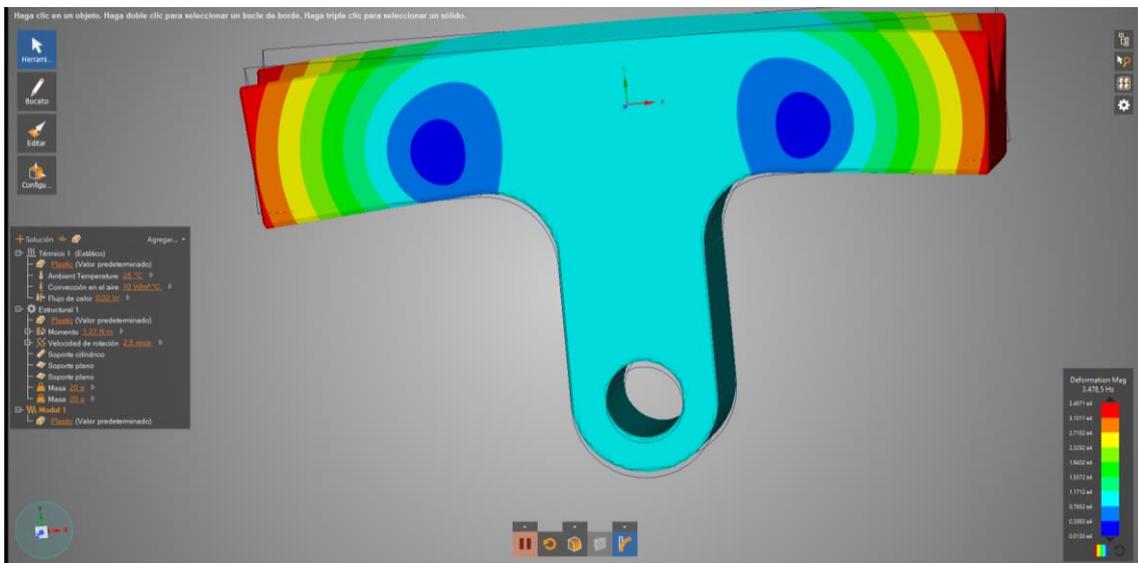
- Estructural

En esta simulación emplearemos las condiciones mecánicas del sistema y las propiedades del motor para así definir correctamente sus valores.



- Modal

Con esta simulación podemos observar donde se realizará la deformación y que magnitud es necesario para que ocurra en Hz.



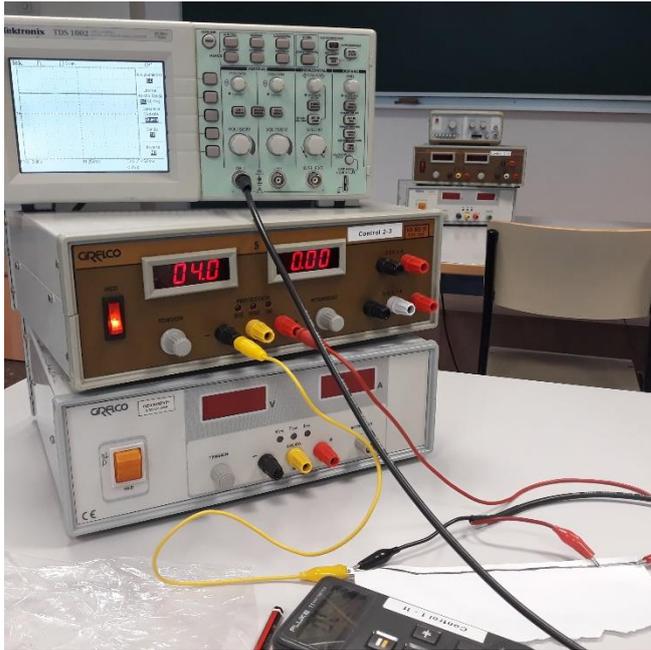
Visto los resultados será imprescindible emplear un tipo de fijación epoxi para la prevención de posibles deformaciones.

### 4.3. Funcionamiento y Caracterización del sensor de posición

Como se ha comentado anteriormente, para el sensor de posición/Potenciómetro emplearemos una traza de grafito dibujada en una cartulina, por ello la cortaremos de acuerdo con la longitud de la viga.

Debido al tipo de sensor rudimentario que vamos a emplear será necesario caracterizarlo correctamente. Para ello emplearemos una fuente de alimentación y una sonda de osciloscopio. El objetivo será la obtención de la relación Voltaje diferencial – distancia.

Para ello realizaremos varias tomas de datos y graficaremos cada una de ellas y la media de ellas, realizaremos la prueba a 5,2V DC de alimentación. El equipo que emplearemos será el siguiente:



*Ilustración 29: Equipo que emplearemos en la caracterización del sensor de posición.*

La imagen mostrada a continuación muestra cómo se realizarán las medidas del sensor.

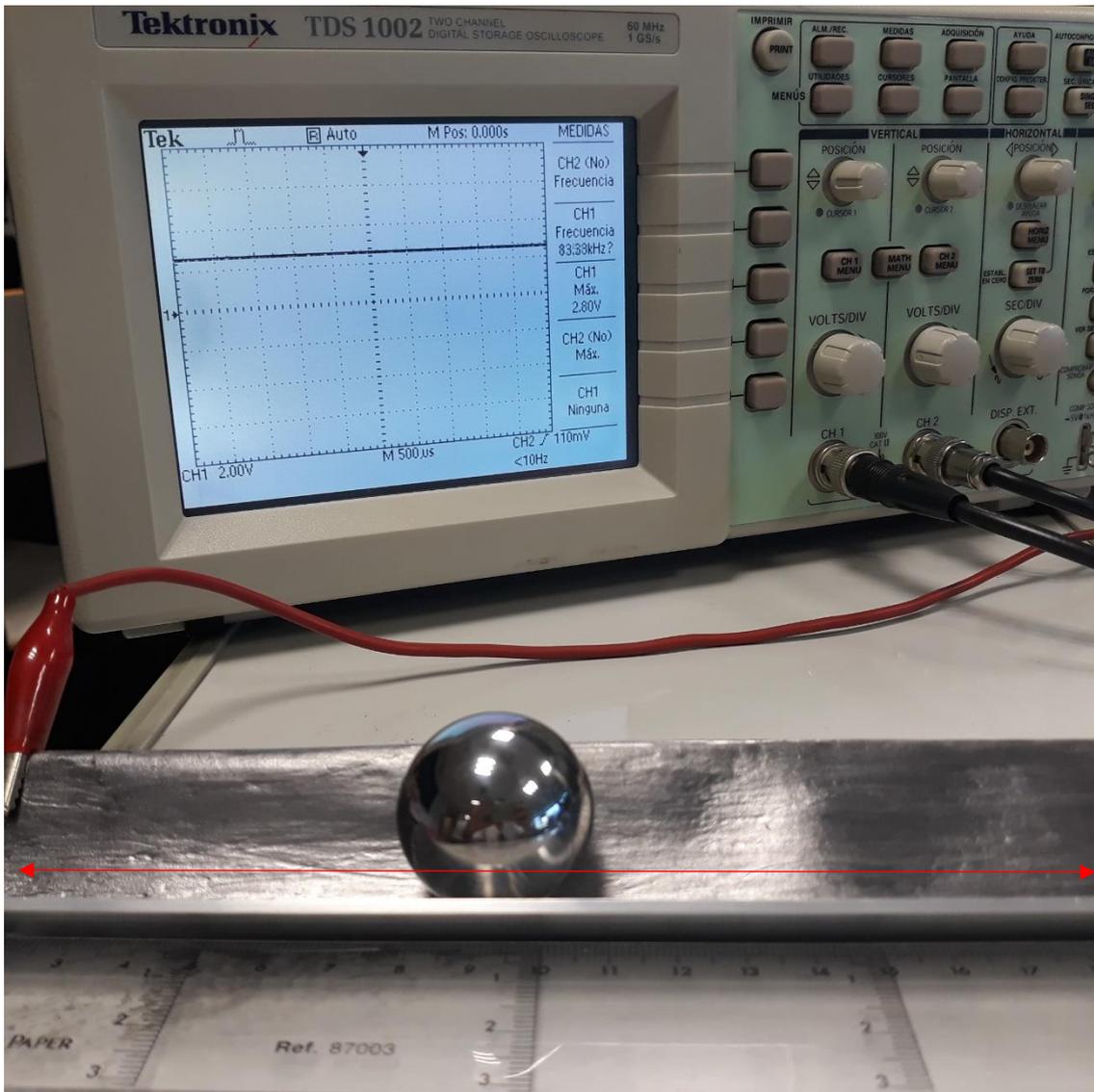


Ilustración 30: Método de medición y caracterización del sensor.

Como se puede observar en la imagen iremos desplazando la bola a través de la viga y la cara del sensor, midiendo su posición y el voltaje que se lee en el osciloscopio.

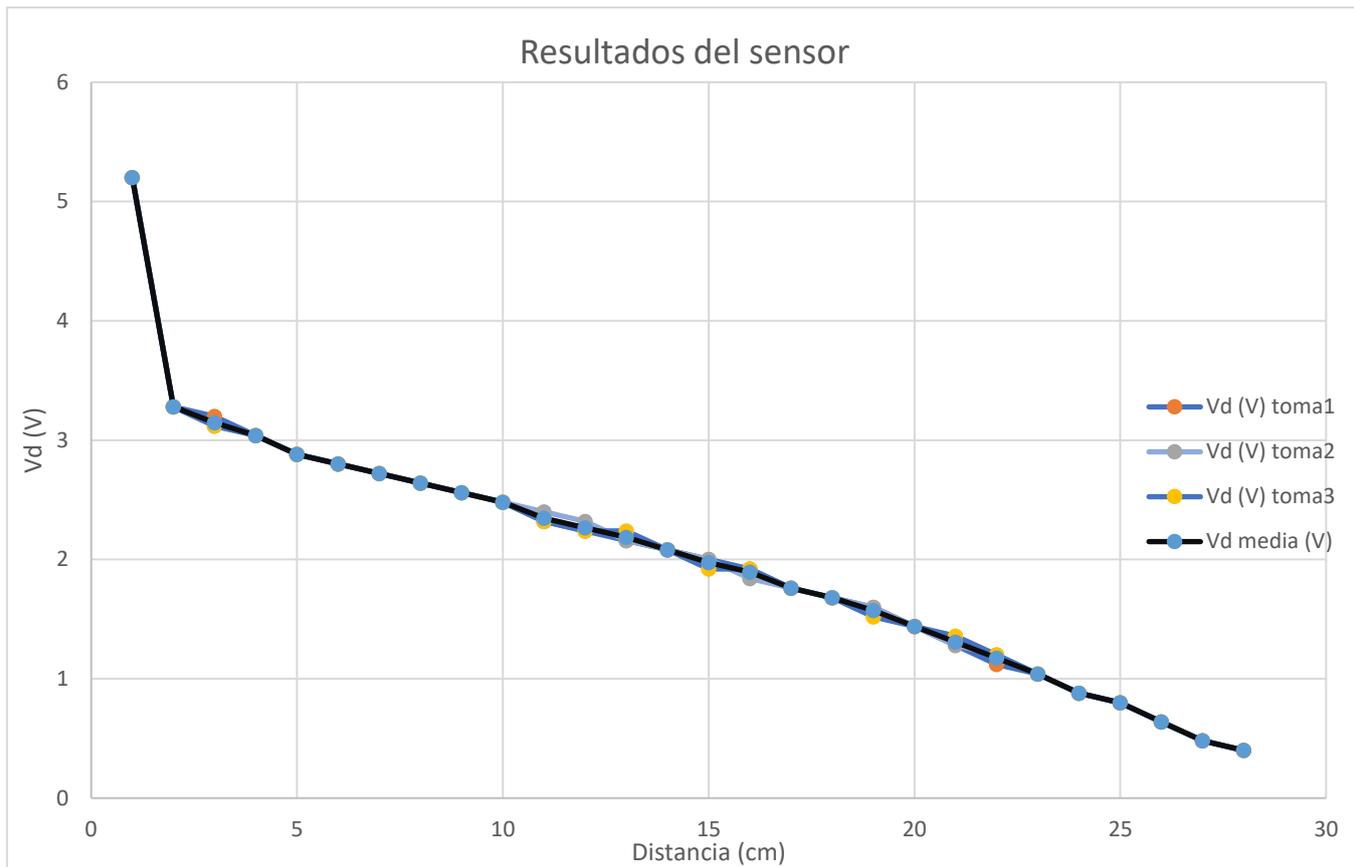
Debido a la posición de los terminales de entrada algunas posiciones no se pueden leer y por tanto aparecerán en blanco en los resultados mostrados a continuación:

<b>Distancia (cm)</b>	<b>Vd (V) toma1</b>	<b>Vd (V) toma2</b>	<b>Vd (V) toma3</b>	<b>Vd media (V)</b>
0	-	-	-	-
1	5,2	5,2	5,2	5,2
2	3,28	3,28	3,28	3,28
3	3,2	3,12	3,12	3,146666667
4	3,04	3,04	3,04	3,04
5	2,88	2,88	2,88	2,88
6	2,8	2,8	2,8	2,8
7	2,72	2,72	2,72	2,72
8	2,64	2,64	2,64	2,64
9	2,56	2,56	2,56	2,56
10	2,48	2,48	2,48	2,48
11	2,32	2,4	2,32	2,346666667
12	2,24	2,32	2,24	2,266666667
13	2,16	2,16	2,24	2,186666667
14	2,08	2,08	2,08	2,08
15	2	2	1,92	1,973333333
16	1,92	1,84	1,92	1,893333333
17	1,76	1,76	1,76	1,76
18	1,68	1,68	1,68	1,68
19	1,6	1,6	1,52	1,573333333
20	1,44	1,44	1,44	1,44
21	1,28	1,28	1,36	1,306666667
22	1,12	1,2	1,2	1,173333333
23	1,04	1,04	1,04	1,04
24	0,88	0,88	0,88	0,88
25	0,8	0,8	0,8	0,8
26	0,64	0,64	0,64	0,64
27	0,48	0,48	0,48	0,48
28	0,4	0,4	0,4	0,4
29	-	-	-	-
30	-	-	-	-

Como se puede observar dentro de los resultados existe cierta conducción, sin embargo, deberemos comprobar que la otra cara no conduzca para que no se produzca un mal contacto entre el papel y el aluminio. Por ello realizaremos de nuevo las tres medidas, pero esta vez con el papel como conductor.

<b>Cara Blanca</b>	<b>Vd toma1 (V)</b>	<b>Vd toma2 (V)</b>	<b>Vd toma3 (V)</b>
0	0,08	0,08	0,08
1	0,08	0,08	0,08
2	0,08	0,08	0,08
3	0,08	0,08	0,08
4	0,08	0,08	0,08
5	0,08	0,08	0,08
6	0,08	0,08	0,08
7	0,08	0,08	0,08
8	0,08	0,08	0,08
9	0,08	0,08	0,08
10	0,08	0,08	0,08
11	0,08	0,08	0,08
12	0,08	0,08	0,08
13	0,08	0,08	0,08
14	0,08	0,08	0,08
15	0,08	0,08	0,08
16	0,08	0,08	0,08
17	0,08	0,08	0,08
18	0,08	0,08	0,08
19	0,08	0,08	0,08
20	0,08	0,08	0,08
21	0,08	0,08	0,08
22	0,08	0,08	0,08
23	0,08	0,08	0,08
24	0,08	0,08	0,08
25	0,08	0,08	0,08
26	0,08	0,08	0,08
27	0,08	0,08	0,08
28	0,08	0,08	0,08
29	0,08	0,08	0,08
30	0,08	0,08	0,08

Como se puede apreciar en los resultados apenas existe conducción en la otra cara, la lectura mostrada corresponde al ruido del propio osciloscopio, por lo que podemos asumir que se puede emplear el sensor. Finalmente mostramos la curva graficada de los datos captados en las primeras tomas para ver si efectivamente se debería usar.



Gráfica 1: Gráfica de la caracterización del sensor.

En la gráfica como podemos observar el sensor se comporta de acuerdo con lo que precisamos y por ello le damos el visto bueno para su uso. Esto se debe a que en la gráfica observamos un comportamiento lineal a medida que nos vamos desplazando por la viga lo cual da a entender que este sensor es capaz de cubrir las necesidades que estamos buscando, las cuales es emplear la diferencia de voltaje para determinar la posición. También podemos observar que apenas existe diferencia entre las distintas tomas dándonos a entender que a pesar de realizar nuevas medidas los valores no cambiarán lo cual muestra cierta precisión.

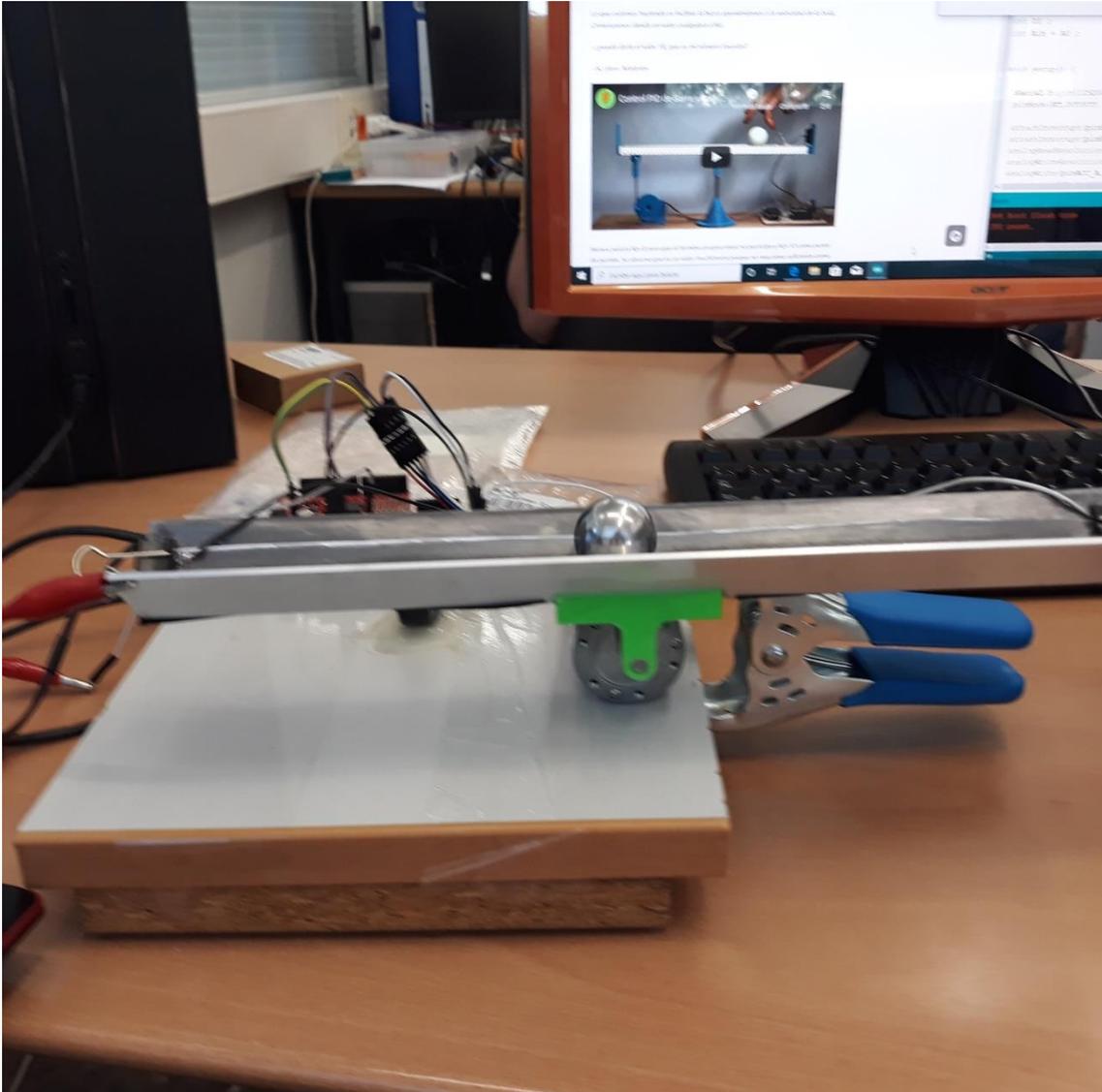
El sensor funcionará de acuerdo con lo siguiente: se le inducirá al tramo del potenciómetro 3,3V DC, debido a que para el sistema de Arduino es capaz de leer mejor en esta lectura que en la de 5V DC. Una vez hecho esto, nos aseguraremos de que el tramo realice contacto con la bola y conectaremos una entrada analógica a la placa de aluminio que realiza el contacto con la bola para así obtener nuestro sensor.

La señal analógica que obtenemos la convertiremos a un valor de entre -1 y 1 para así poder facilitar un poco el procesamiento del sistema.

Finalmente comenzaremos con el montaje del prototipo y pasaremos al control que se empleará en el mismo.

#### 4.4. Control empleado

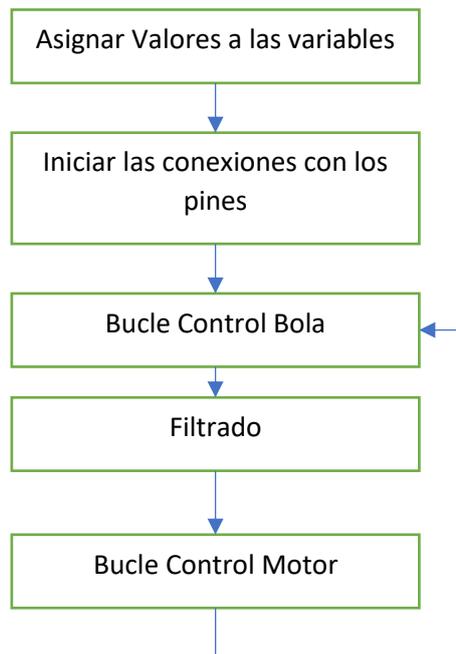
El control que emplearemos se efectuara primeramente en Arduino. Se dividirá en 2 secciones: Control de Referencia y Control del Motor. En ambos casos emplearemos un PID.



*Ilustración 31: Prototipo Bola y Viga.*

Antes de aplicar el control de referencia la señal obtenida por el potenciómetro deberá ser filtrada debido a la cantidad de ruido que esta porta. Una vez filtrada estará lista para ser utilizada como la referencia.

El diagrama de flujo del sistema es el mostrado a continuación:



*Ilustración 32: Flujograma Arduino.*

Como podemos comprobar primero se reasignarán la variables y se establecerán las conexiones y después se iniciara el bucle de control de la bola el cual se encargara de recoger y corregir la posición de la bola para ello filtrara la entrada del sensor con la fórmula de filtrado preparada, finalmente pasara a la zona del bucle de control del motor el cual empleara la salida obtenida del control de la bola para corregirla gracias al motor.

Primero se realizará el control del motor, ya que el primer paso será el de ajustar el motor todo lo posible para tratar de observar el movimiento simulado. Una vez que se realice el máximo ajuste se procederá con la referencia hasta que el sistema sea capaz de devolver la bola a su posición original.

#### 4.5. Resultados obtenidos

Tras la implementación real del sistema, llega el análisis de los resultados obtenidos para las distintas pruebas, para ello hemos tratado de mantener la relación de las pruebas entre la simulación y el sistema real.

La existencia de la “Zona muerta” causada por la estructura del propio motor provocan la aparición de un error bastante notable respecto a la simulación realizada. Por ello se ha empleado la acción integral dentro del bucle de control para reducir este efecto sin llegar a provocar demasiada inestabilidad.

La bola reconoce en algunos casos el punto 0 y suele no iniciar correcciones en este punto ya que la referencia del sistema es la misma. La siguiente imagen es la prueba de ello. La posición de la bola está representada de color azul, la referencia de color rojo y la posición del motor en verde.

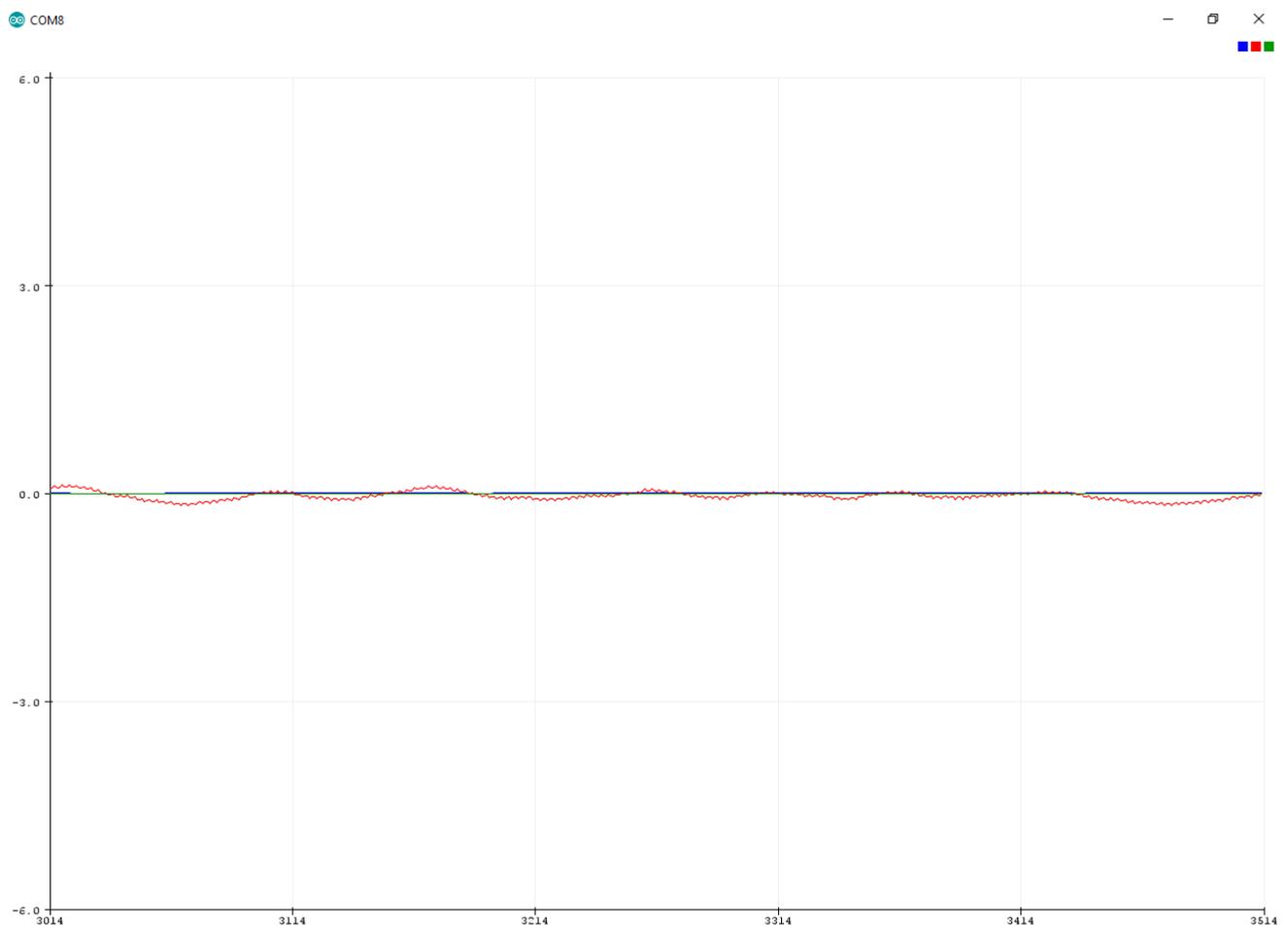


Ilustración 33: Bola en 0 (Origen).

Como podemos observar la referencia apenas se llega a mover y la posición del motor esta fija en el 0.

Dentro de los resultados, la bola trata de seguir la referencia marcada por el sistema, pero debido en gran parte a la zona muerta del motor y el tipo de motor. Por ello en algunas pruebas ha llegado a tardar en exceso en detenerse. También puede ser debido al sensor que estamos empleando.

Mostramos en la siguiente imagen el comportamiento del sistema cuando la bola se desplaza.

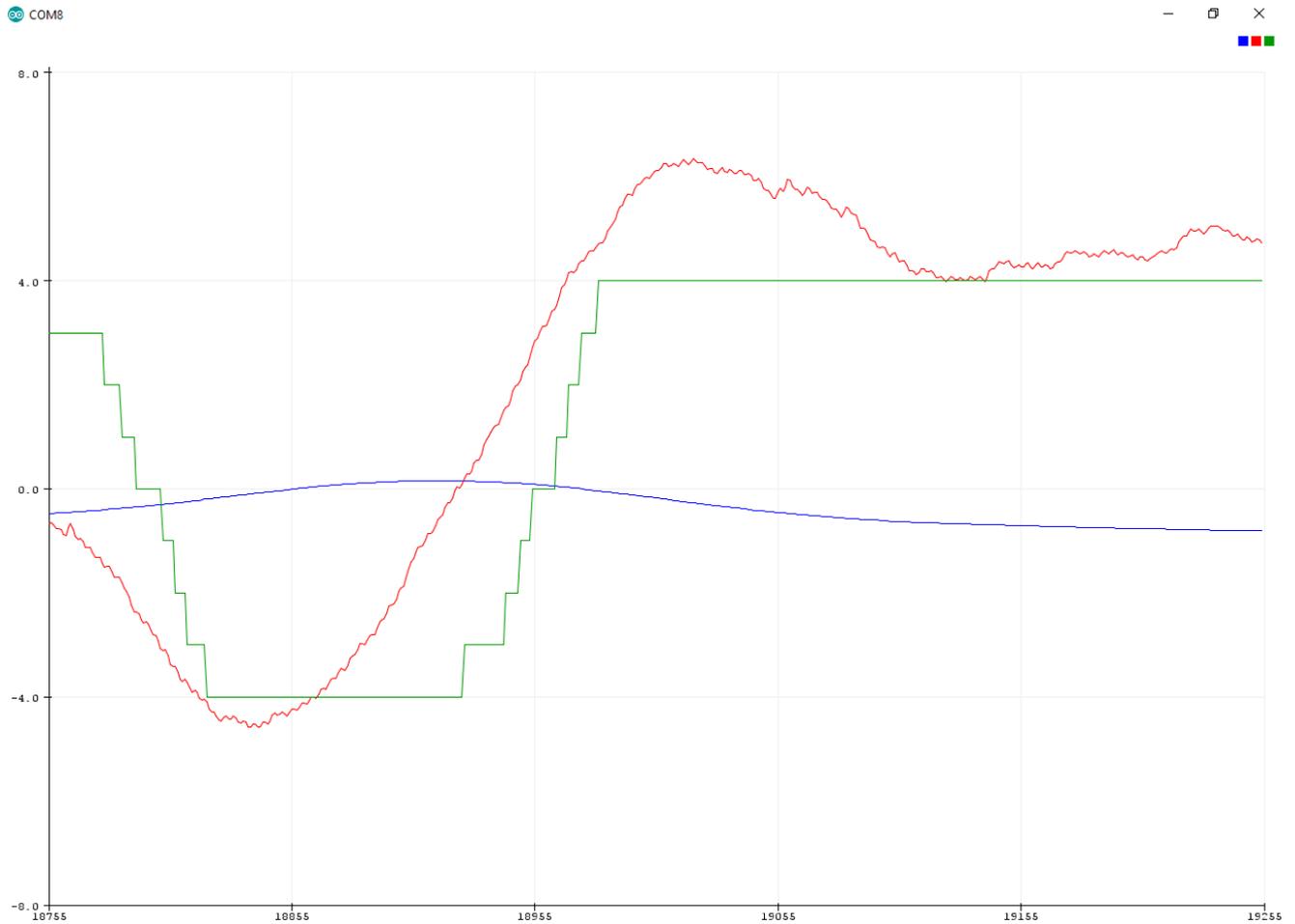
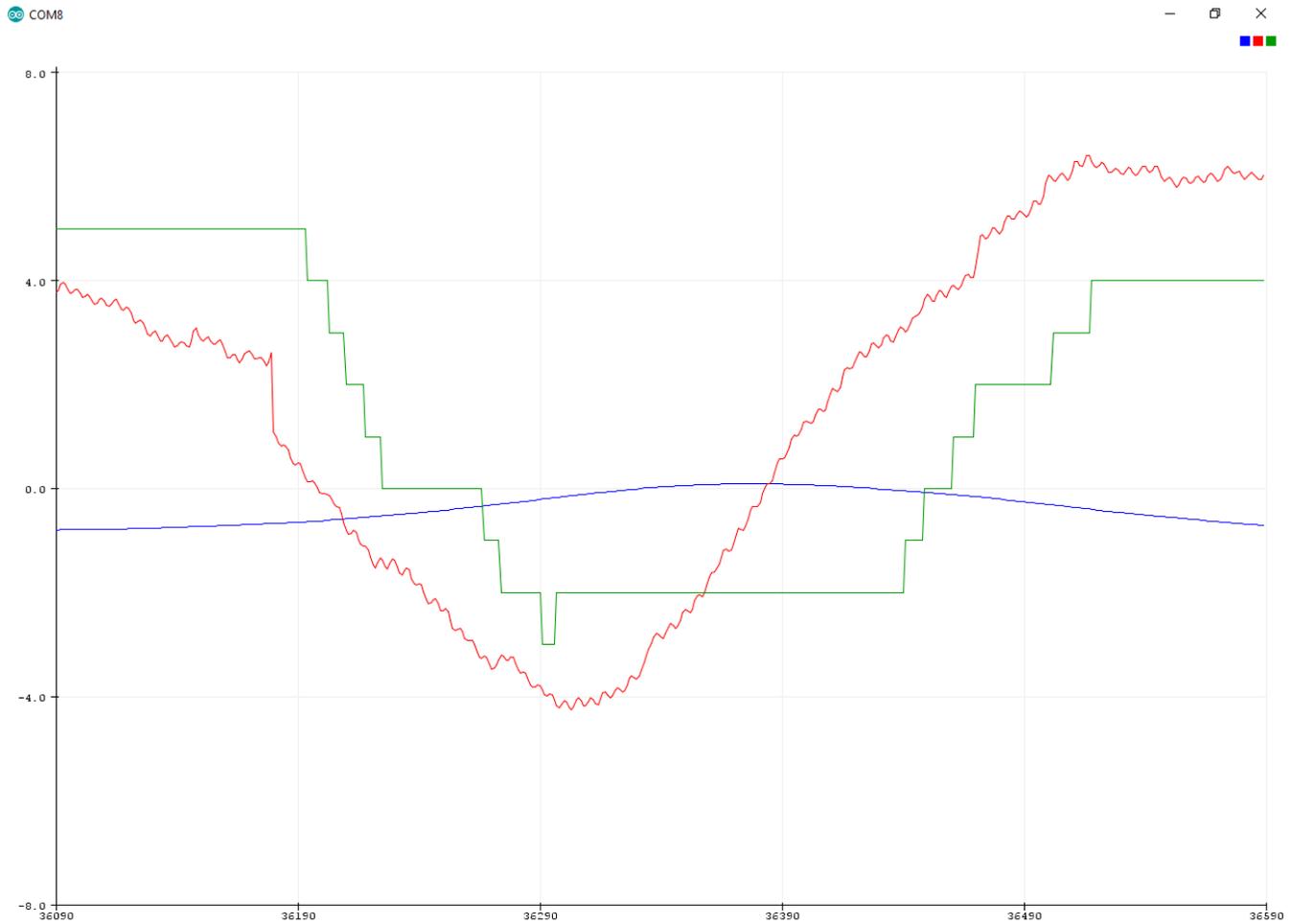


Ilustración 34: Bola desplazándose.

Como podemos apreciar a pesar de que la bola tarda cierto tiempo en corregir su posición, el sistema actúa de manera correcta haciendo que la posición del motor siga la referencia tal y como se muestra.



*Ilustración 35: Bola desplazándose II.*

No obstante, el sistema es capaz de seguir la trayectoria y detecta la dirección en la cual el motor debe angular. Aunque le falta un poco más de precisión en la estabilización.

Sustituyendo el tipo de motor, a uno con un transitorio más corto o una zona muerta más pequeña sería posible ajustarlo con mayor precisión para realizar el movimiento con más exactitud.

## 5. Conclusiones

El trabajo en cuestión muestra los pasos a seguir para la implementación del prototipo del sistema Bola y Viga, como también la implementación del control mediante los distintos tipos de software.

El principal objetivo de este prototipo es que, al desplazar la Bola, el motor trate de volverla a posicionar en el origen (es decir en la posición central), así como reaccionar ante posibles perturbaciones. En el prototipo la bola trataba de llegar a la posición y en algún momento llegaba a la posición, pero o no se detenía o tardaba una considerable cantidad de tiempo en detenerse. Y como hemos podido observar debido a la estructura de nuestro motor la zona muerta actuaba en mayor parte en nuestra contra.

No obstante, el sistema era capaz de seguir la trayectoria y era capaz de corregir la trayectoria de bola para reconducirla al origen como se han observado en los resultados.

Finalmente podemos concluir que la gran parte de los objetivos propuestos se han logrado cumplir. Ciertamente que no se han obtenido con la precisión necesaria, sin embargo, hemos obtenido unos resultados únicos que resaltan que el control realizado ha sido el mejor posible para el sistema.

Si se deseara continuar con este sistema se recomendaría el uso de un servomotor con mayor precisión y una menor zona muerta, o bien un motor con un transitorio más corto lo cual también sería una opción válida.

## 6. Presupuesto

### 6.1. Coste material

El coste total del prototipo queda comentado en la siguiente tabla, así como el de todos los materiales empleados, también se ha decidido poner el precio de las licencias de los programas que se han empleado, en el caso de Ansys Discovery live se ha empleado una versión gratuita que hemos obtenido desde la página de Ansys.

<i>Material</i>	<b>Unidades</b>	<b>Precio (€)</b>	<b>Importe (€)</b>
<i>Perfil en L</i>	1	4,79	4,79
<i>Motor Pololu 37D 70:1</i>	1	36,0	36,0
<i>Tarjeta Arduino DUE</i>	1	34,5	34,5
<i>Cartulina Blanca</i>	1	0,2	0,2
<i>Mezcla Epoxi</i>	1	4,0	4,0
<i>Portapilas 12V</i>	1	2,6	2,6
<i>Lápiz B2</i>	1	0,4	0,4
<i>Gato de agarre</i>	1	3,2	3,2
<i>Cable USB</i>	1	1,75	1,75
<i>Cables Macho/Macho</i>	2	0,13	0,26
<i>Cables Cocodrilo</i>	1	0,10	0,10
<b>Licencias</b>			
<i>Licencia Matlab perpetual</i>	1	2000	2000
<i>Licencia Solidworks profesional</i>	1	8250	8250
<i>Licencia Ansys Discovery</i>	1	0	0
<b>Total material:</b>			87,8
<b>Total + licencias:</b>			10337,8

## 6.2. Coste laboral

En base al convenio colectivo nacional de empresas de ingeniería y oficinas de estudios técnicos publicado en el BOE, se establece el coste/hora de un ingeniero de en torno a 22 € a la hora.

<i>Descripción</i>	<b>Tiempo (h)</b>
<i>Estudio y análisis</i>	40
<i>Diseño</i>	60
<i>Programación</i>	60
<i>Montaje</i>	10
<i>Simulación, ensayos y depuración de errores</i>	90
<b>Total</b>	<b>260</b>

Se ha considerado el puesto de Ingeniero Técnico Industrial debido al tipo de proyecto y el arco que abarca.

<i>Descripción</i>	<b>Tiempo (h)</b>	<b>Precio (€)</b>	<b>Total (€)</b>
<i>Ingeniero Técnico Industrial</i>	260	22	5720

Finalmente, la tabla a continuación nos muestra el coste total del proyecto.

<i>Descripción</i>	<b>Total (€)</b>
<i>Coste material</i>	87,8
<i>Coste Licencias</i>	10250
<i>Coste laboral</i>	5720
<b>Coste Total</b>	<b>16057,80</b>

En resumen, el coste total del proyecto se estima en unos 16.057,80 € incluyendo las licencias de los programas. Si estas licencias ya han sido proveídas por algún otro departamento el coste del proyecto se estimaría en 5807,80 €.

## 7. Pliego de Condiciones

### 7.1. Condiciones generales

#### 7.1.1. Objeto

El presente documento tiene como objetivo la definición total de las condiciones básicas del proyecto “Diseño e implementación del sistema Ball & Beam mediante Simscape Multibody y Arduino”, donde se describen los conceptos técnicos, facultativos, económicos y legales que afectan al proyecto.

Se trata de un proyecto de investigación y, por tanto, solamente abarcará aspectos relacionados con el diseño, construcción, programación y puesta en marcha de los distintos dispositivos.

Todo lo que afecte a la ejecución del proyecto deberá seguir las normas y condiciones técnicas establecidas en este documento, en lo que se incluye:

- Normativa legal que afecta a los materiales empleados y la forma de usarlos, los procesos a seguir y las personas que estén sujetas a la constitución del proyecto.
- Seguridad del proyecto. Todos los aspectos para tener en cuenta de modo que no produzca daños a las personas o instalaciones en el momento de ejecución.
- Descripción de los materiales empleados en la ejecución del proyecto, incluyendo características técnicas de los mismos.
- Descripción detallada de todos los procesos de ejecución, ya que afectará a la calidad final del montaje.

#### 7.1.2. Normativa y Reglamento Vigente

Se describen las condiciones y normas de carácter general de obligado cumplimiento en la ejecución del proyecto. Las normas de carácter particular se detallan en posteriores apartados.

En lo que refiere a los elementos, componentes o aparatos incluidos en la instalación electrónica o eléctrica utilizados para el desarrollo del proyecto se seguirán las normas establecidas por el

“REBT”, por estar conectados a la red de suministro eléctrico de 220V, así como las normas UNE y DIN que se describen a continuación.

- BT-19 – “Instalaciones interiores o receptoras. Prescripciones generales”
- BT-22 – “Protección contra sobrecorrientes”
- BT-24 – “Protección contra los contactos directos e indirectos”
- UNE 20315-1-1:2004 – “Bases de toma de corriente y clavijas para usos domésticos y análogos”.

#### 7.1.3. Pliego de condiciones facultativas

El cliente tiene la obligación de conocer:

- Las leyes.
- El proyecto en su totalidad.

### 7.1.3.1.Programación de Trabajos

#### 7.1.3.2.Ejecución Defectuosa y Modificaciones

Recae sobre el cliente la responsabilidad de revisar el correcto funcionamiento del programa suministrado e informar al responsable de aquellos aspectos que no funcionen de la manera apropiada.

En el caso de que el cliente quisiera realizar alguna modificación, el sobrecoste provocado deberá ser asumido por el mismo, ya que se considerarán cambios prescindibles que no afectan al normal funcionamiento del proyecto.

En caso de modificaciones, las tarifas a aplicar serán las mismas que para el resto del proyecto, siguiendo las normas publicadas en el BOE. Si se detectasen errores debidos a una mala programación, el responsable del proyecto deberá asumir los costes provocados para subsanarlos.

#### 7.1.3.3.Recepción Definitiva

Una vez finalizado el plazo de ejecución del proyecto se hará efectiva la entrega del mismo al usuario final.

### 7.1.4. Descripción General del Montaje

En este apartado se detalla de forma esquemática los pasos a seguir para la ejecución del proyecto.

- Diseño del mecanismo.
- Simulación del mecanismo.
- Adquisición del material de montaje
- Caracterización del sensor de posición
- Construcción del prototipo.
- Programación del microcontrolador
- Control de calidad.

### 7.1.5. Pliego de Condiciones Económicas

A continuación, se detallan las relaciones económicas que deben regir entre el cliente y el proyectista.

- Fianzas.
- Composición de precios.
- Mejoras.

#### 7.1.5.1.Fianzas

Las condiciones de pago del proyecto son las descritas a continuación: El comprador deberá depositar una fianza del 20% del valor total del proyecto, descrito en el apartado **Presupuesto**, como modo de garantía.

En el caso de que se produjesen retrasos en el abono de la fianza, ésta aumentaría en un 5% semanal. El resto del valor del proyecto deberá ser abonado una vez finalizado y previa comprobación del correcto funcionamiento.

#### 7.1.5.2.Composición de Precios

Los precios aplicados al proyecto están en consonancia con los dispositivos generales acordados en los convenios en los que tiene ámbito.

#### 7.1.5.3.Mejoras

En el caso de que el cliente realice mejoras, éstas supondrán un aumento en el presupuesto total estipulado con anterioridad.

## 7.2. Condiciones Particulares

### 7.2.1. Condiciones de los materiales

#### 7.2.1.1. Cableado Eléctrico

Todo el cableado deberá instalarse en un lugar en el cual esté protegido frente a la intemperie y, por tanto, no será necesario ningún tipo de protección adicional.

Por otra parte, según obliga la norma se debe ofrecer una protección mínima frente a descargar y, por tanto, los cables estarán protegidos mediante material aislante de PVC o similar.

#### 7.2.1.2. Cables de Alimentación

Para la alimentación del microcontrolador se empleará tanto el cable de alimentación al PC proporcionado por el fabricante como una batería externa, cuyos cables también son proporcionados por el mismo.

### 7.2.2. Condiciones de Funcionamiento del sistema

En este apartado se detallan los requisitos mínimos que se deben cumplir tanto en el apartado de hardware como en software para garantizar un correcto funcionamiento.

#### 7.2.2.1. Limitaciones por hardware

- Procesador Intel Core i7-4700HQ, 2,4GHz o superior.
- Memoria RAM 8GB o superior.
- Espacio libre en el disco duro de aproximadamente 20 GB (Matlab y SolidWorks) y 15 GB para ANSYS Discovery Live.
- Periféricos necesarios.
- Tarjeta gráfica NVIDIA GeForce GTX 860M.

#### 7.2.2.2. Limitaciones por Software

Los requisitos correspondientes al software son los siguientes:

- Windows 10 64 bits o superior.
- Matlab 2017B.
- SolidWorks 2017-2018.
- Arduino o Arduino IDE.

### 7.2.3. Condiciones de la ejecución.

#### 7.2.3.1. Condiciones de la instalación

Los programas serán instalados por el director del proyecto, ya que el promotor (Cliente) es posible que no tenga los conocimientos necesarios para su correcta instalación.

#### 7.2.3.2. Condiciones de Garantía

En caso de pérdida de alguno de los ficheros necesarios para la activación del producto o bien generales, el director del proyecto deberá dar su completa autorización para poder efectuar una nueva copia de esos ficheros extraviados.

Si se produjera durante los 2 primeros años desde la adjudicación del proyecto, el responsable de este se compromete a suministrar otra copia en un plazo máximo de 10 días hábiles desde la notificación de la pérdida.

El período de la garantía es de 2 años. Una vez transcurrido este período el proyectista no se responsabiliza de los posibles defectos surgidos tanto en el apartado del hardware como en el software.

#### 7.2.4. Prueba de Servicio

Será necesaria la inspección de los distintos componentes que componen el proyecto para descartar que se encuentren en mal estado previamente a la realización del montaje. Además, se comprobará tanto el correcto estado de las conexiones físicas como el buen funcionamiento del software instalado y programado

## 8. Bibliografía

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.

## 9. Anexos

### 9.1. Código Arduino

```
/*
 * B_B_Programa: Programa Ball & Beam. Autor: Guillermo Jover
 Ibáñez
 * Versión: 05/09/2019.
 */
#define RT 53

#define CW 1

#define CCW 0

#define pinDIR_A 2
#define pinACC_A 3
#define pinENC_A_chA 40
#define pinENC_A_chB 41

#define MINU -1.0
#define MAXU 1.0

#define MINP -180
#define MAXP 180

#define ZMP 0.3
#define ZMN -0.3 #define
SAT 1.0

long t1 = 0 ; long t2 = 0 ; int
T = 5 ; float Ts = T / 1000.0 ;
float C2D = 360.0 / 4480.0 ;
float t = 0.0 ;

long quad = 0 ; float Pos_analog = 0.0, P_B = 0.0; float KpM = 0.007,
KiM = 0.01, KdM = 0.0005 ; //kdMm buena 0.
0005, kp buena=0.007,ki buena=0.01 float KpB = 4.1, KiB = 1.9, KdB =
4.33; //Kp buena 4.1,Ki buena=1.9, Kd buena=4.33 float Rint = 10,
Rext = 45; float ref = 0.0, refB = 0.0, per = 1.0 ; float err = 0.0,
err_ant = 0.0, errB = 0.0, err_antB = 0.0 ; float pos = 0.0, acc =
0.0 , accB = 0.0; float Int = 0.0, IntB = 0.0 ; int cnt = 0 ;
```

```

float medida = 0.0, medidaF = 0.0 ; float uk = 0.0, uk1 = 0.0, uk2 =
0.0, yk1 = 0.0, yk2 = 0.0 ;

int D1 ;

int Ain = A0 ; // Entrada
analógica 0

void setup() {

    Serial.begin(115200) ; pinMode(RT, OUTPUT) ; for (int ii = 1;
ii <= 10; ii++) { digitalWrite(RT, HIGH) ; delay(50) ;
digitalWrite(RT, LOW) ; delay(50) ;

    }

    attachInterrupt(pinENC_A_chA, encoder_A_chA, CHANGE) ;
attachInterrupt(pinENC_A_chB, encoder_A_chB, CHANGE) ;
analogReadResolution(8) ; // Entrada
analógica 8 bits de lectura y escritura
analogWriteResolution(8) ; analogWrite(pinACC_A, 0) ;
pinMode(pinDIR_A, OUTPUT ) ;

    delay(5000) ;

}

void loop() { /*Control
Bucle Bola */

    D1 = analogRead(Ain) ; // Lectura de la posición de la bola en
el sensor. Pos_analog = float((2.0 * D1
/ 256.0) - 1.0); medida = Pos_analog;

    Filtrado(4); //usamos el filtrado maximo P_B = medidaF; errB
= refB - P_B; if (abs(P_B * 15) > Rint && abs(P_B * 15) < Rext)
{ //Colocamos cuando empezara a actuar la acción integral.

        IntB = IntB + errB * Ts;

    } else {
IntB = 0; }
    accB = errB * KpB + ((errB - err_antB) / Ts) * KdB + IntB * KiB;
err_antB = errB;

/*Control Bucle Motor */ ref = 0.0 * sin(2 * 3.14 * 0.1 * t) ; ref
= accB; pos = quad * C2D ; pos = (((int)pos * 10 + 1800) % 3600) -

```

```

1800) / 10.0 ;   err = ref - pos ;   Int = Int + err * Ts ;   acc =
KpM * err + KiM * Int + KdM * (err - err_ant) / Ts ;   err_ant = err
;
  Serial.print(P_B) ; Serial.print("\t"); Serial.print(ref) ;

Serial.print("\t") ; Serial.println(pos) ;
//Configuración de la zona muerta   if (acc >
0.1 * ZMP && acc < ZMP) {   acc = ZMP ;

  } ; if (acc < 0.1 * ZMN && acc > ZMN) {   acc
= ZMN ;

  }   if (acc > 0.0 && acc <= 0.1 * ZMP) {
acc = 0.0 ;
  } ; if (acc < 0.0 && acc >= 0.1 * ZMN) {   acc
= 0.0 ;

  }   if (acc > MAXU) {
acc = MAXU ;   } ; if (acc
< MINU) {   acc = MINU
;
  }

  if (acc > 0) {   digitalWrite(pinDIR_A,
CW) ;

  } else {   digitalWrite(pinDIR_A,
CCW) ;

  }   analogWrite(pinACC_A, (byte)(abs(acc) * 255)) ;

  t2 = millis() ;   if (t2 - t1
> T) {   digitalWrite(RT,
HIGH) ;

  }   else   {
digitalWrite(RT, LOW) ;

  }   while (t2 - t1 < T)
{   t2 = millis() ;
  }   t1 = millis() ;   t
= t + T / 1000.0 ;

}

// Función de filtrado de la entrada del sensor void
Filtrado(int ff) {   uk2 = uk1 ; uk1 = uk ; uk = medida
;   switch (ff) {   case 1: // wn=0.25

      medidaF = 0.09763 * uk + 0.1953 * uk1 + 0.09763 * uk2 +
0.9428 * yk1 - 0.3333 * yk2 ;   break ;   case 2: // wn=0.1
medidaF = 0.0201 * uk + 0.0402 * uk1 + 0.0201 * uk2 + 1.

```

```

5610 * yk1 - 0.6414 * yk2 ;          break ;          case 3: // wn=0.025
medidaF = 0.0015 * uk + 0.0029 * uk1 + 0.0015 * uk2 + 1.
8890 * yk1 - 0.8949 * yk2 ;
break ;          case 4: // wn=0.01

          medidaF = 0.2414e-3 * uk + 0.4827e-3 * uk1 + 0.2414e-3
* uk2 + 1.9556 * yk1 - 0.9565 * yk2 ;

          break ;          default:
medidaF = medida ;          break ;
}  yk2 = yk1 ; yk1 = medidaF ;
}

//funciones del encoder void
encoder_A_chA () {

    if (digitalRead(pinENC_A_chA) == digitalRead(pinENC_A_chB))
{
    quad-
-;  } else
{
quad++;
    }
}

void encoder_A_chB () {          if (digitalRead(pinENC_A_chA) ==
digitalRead(pinENC_A_chB))

{
quad++;  }
else {
quad--;
    }
}
}

```

9.2. Planos

