



**Universidad**  
Zaragoza

# Trabajo Fin de Grado

Desarrollo e implementación de un sistema de control posición en 2D de una plataforma de nanoposicionamiento

Autor

**Guillermo González Calvo**

Directores

Lucía Candela Díaz Pérez

José Antonio Albajez García

Escuela de Ingeniería y Arquitectura

2019

# Resumen

El objetivo de este trabajo es el de diseñar e implantar un sistema de control 2D en una plataforma de nanoposicionamiento, denominada como NanoPla. Para ello, este trabajo se ha basado en estudios anteriores y tras haber explicado el funcionamiento de la plataforma se han realizado varias subtarefas para lograr implementar el control. Se ha realizado un ajuste del sistema láser que se emplea para realimentar la posición, el cual es clave para que el control funcione. Tras esto, en el trabajo se muestra el proceso seguido para la obtención de un modelado del sistema adecuado, que permitió obtener las funciones de transferencia que rigen el movimiento de la plataforma. Estas funciones, posteriormente, se emplearon en la labor del diseño de los controladores PID necesarios para el control de la plataforma. Una vez diseñados e implementados los PID, se analizan los resultados obtenidos y se valida el control de la plataforma mediante el estudio de las respuestas de la plataforma ante diversas entradas y mediante el análisis del ruido que genera en el eje Z el control diseñado. Además, para facilitar el estudio y análisis de estos resultados se realizó una interfaz gráfica orientada al control con PID en el entorno de Simulink. Para finalizar, se obtuvieron las conclusiones del proyecto realizado.

## Contenido

<b>1. INTRODUCCIÓN .....</b>	<b>1</b>
1.1 CONTEXTO Y JUSTIFICACIÓN DEL PROYECTO.....	1
1.2 OBJETIVO Y ALCANCE .....	2
<b>2. PLATAFORMA DE NANOPOSICIONAMIENTO .....</b>	<b>3</b>
2.1 ESTRUCTURA DE LA PLATAFORMA.....	3
2.2 FUNCIONAMIENTO DE LOS MOTORES HALBACH .....	5
2.3 SISTEMA DE MEDIDA DE LA POSICIÓN .....	9
2.4 TARJETAS TEXAS INSTRUMENTS .....	11
2.5 MONTAJE EXPERIMENTAL .....	12
<b>3. AJUSTE DEL SISTEMA DE MEDIDA DE LA POSICIÓN.....</b>	<b>14</b>
<b>4. ESTRATEGIA DE CONTROL .....</b>	<b>16</b>
4.1 ANTECEDENTES DEL CONTROL 2D Y ESTRATEGIA APLICADA .....	16
4.2 MODELADO DEL SISTEMA.....	18
4.2.1 <i>Funciones de transferencia para los desplazamientos en X e Y.....</i>	<i>18</i>
4.2.1.1 Desplazamiento en el eje X .....	20
4.2.1.2 Desplazamiento en el eje Y.....	26
4.2.2 <i>Obtención de la función de transferencia del giro.....</i>	<i>29</i>
4.3 DISEÑO E IMPLEMENTACIÓN DE LOS PID .....	31
4.3.1 <i>Ajuste de los PID.....</i>	<i>31</i>
4.4 RESULTADOS EXPERIMENTALES .....	35
4.4.1 <i>Respuestas a movimientos a velocidad constante.....</i>	<i>35</i>
4.4.2 <i>Cambio de <math>F_z</math> y de velocidad.....</i>	<i>44</i>
4.4.3 <i>Estudio del ruido en el eje z a través de sensor confocal .....</i>	<i>47</i>
<b>5. INTERFAZ PARA TRATAMIENTO DE DATOS DEL CONTROL PID.....</b>	<b>52</b>
5.1 OBJETIVO Y HERRAMIENTA EMPLEADA .....	52
5.2 DESCRIPCIÓN DEL PROGRAMA .....	53
5.2.1 <i>Inicio.....</i>	<i>54</i>
5.2.2 <i>Entradas.....</i>	<i>55</i>
5.2.3 <i>Salidas.....</i>	<i>57</i>
<b>6. CONCLUSIONES.....</b>	<b>59</b>
<b>7. BIBLIOGRAFÍA .....</b>	<b>60</b>
<b>ANEXO 1: PROGRAMA INTERFAZ COMPLETO.....</b>	<b>61</b>

# 1. Introducción

En este apartado introductorio se explica el marco en el que se engloba este Trabajo Fin de Grado y los aspectos generales que se tratan a lo largo de este, definiendo tanto los objetivos marcados como el alcance del proyecto.

## 1.1 Contexto y justificación del proyecto

Este trabajo se engloba dentro de una línea de investigación que se está llevando a cabo en el Departamento de Ingeniería de Diseño y Fabricación de la Universidad de Zaragoza, más concretamente dentro del área de Ingeniería de los Procesos de Fabricación.

El proyecto que se está llevando a cabo consiste en el desarrollo de una plataforma de nanoposicionamiento en dos dimensiones cuyos fines últimos son metrológicos y en el campo de la nanotecnología, campo en auge en los últimos años. La primera aplicación para la que se quiere emplear la plataforma es para alojar y posicionar un microscopio de fuerzas atómicas (AFM), un instrumento que se emplea para caracterizar la superficie de distintas probetas o pequeñas muestras.

Este trabajo parte de los resultados obtenidos en trabajos anteriores (1) y (2), donde se conseguía el control de la posición de uno de los motores de la plataforma, es decir un control en 1D, y continúa en esa línea para ampliar ese control a los cuatro motores de la plataforma, es decir un control de la posición en 2D.

A lo largo del presente trabajo se hará referencia a dos programas de control de Simulink, que se emplearon a la hora de realizar los distintos experimentos; la realización de esos programas no se llevó a cabo durante este trabajo, sino que fueron realizados previamente en el contexto de esta línea de investigación de la Universidad. Los programas en cuestión son el control en bucle abierto y cerrado del sistema. Si bien es cierto que se realizaron pequeños cambios y ajustes en estos programas, en especial para la creación de la interfaz para el programa en bucle cerrado, al no cambiar la funcionalidad del programa, estos programas no se explican durante esta memoria.

Para la realización del trabajo, se parte de la estructura de la plataforma desmontada, ya que, para un mayor conocimiento de esta, se empezó con su montaje y ajuste, incluyendo el cableado de las tarjetas a los motores, el ajuste de los cojinetes de aire o el ajuste de los láseres. Todas las partes de la plataforma nombradas aquí, se explican más adelante en detalle.

## 1.2 Objetivo y alcance

El objetivo de este trabajo, como se indica en el título, es el desarrollo e implementación de un sistema de control para el movimiento en 2D de una plataforma de nanoposicionamiento. Este objetivo a su vez se divide en varias tareas.

- **Ajuste del sistema de medida:** al partir de la base de que la plataforma estaba desmontada, una de las tareas a realizar al principio es la de ajustar el sistema de medida, que está formado por un sistema de interferómetros láser.
- **Modelado y estudio del sistema:** se modelará y estudiará el sistema objeto de este proyecto para así poder diseñar y ajustar el control en dos dimensiones. Para ello se determinarán las funciones de transferencia que rigen el movimiento de la plataforma y se simularán para estudiar su comportamiento.
- **Diseño de los controladores PID y análisis de los resultados:** a partir de la información obtenida en el estudio del sistema se diseñarán unos PID adecuados para el posicionamiento de la plataforma y se analizarán sus resultados.
- **Estudio del ruido en el eje z de los PID mediante un sensor confocal:** se estudiará cómo afecta la implantación de los PID al ruido que se produce en el eje Z en la plataforma.
- **Diseño de una interfaz gráfica:** para poder manejar de manera más sencilla el programa de Simulink que controla la plataforma en bucle cerrado con los controladores PID y para visualizar los resultados de manera más sencilla, se procederá al diseño de una interfaz gráfica en el entorno de MatLab gracias a su herramienta GUIDE.

## 2. Plataforma de nanoposicionamiento

En el presente trabajo se va llevar a cabo el estudio de una plataforma de nanoposicionamiento, también llamada NanoPla, que es una plataforma que tiene un rango de trabajo de 50 mm x 50 mm y que permite el posicionamiento a escala micrométrica, cuyo uso radica en un gran número de aplicaciones, como pueden ser el uso de un microscopio de fuerzas atómicas, de un sensor confocal o incluso para litografía.

A lo largo de este apartado se definirán los elementos que componen esta plataforma y todos los elementos que se emplean para conseguir el control del posicionamiento de esta plataforma, y que es el objetivo principal de este trabajo.

### 2.1 Estructura de la plataforma

La plataforma consta de 3 partes, dos de ellas son fijas, las bases superior e inferior, y la otra, la que se encuentra en medio, es la parte móvil. Además, cuenta con tres cojinetes de aire, que son los que hacen que la parte móvil de la plataforma pueda levitar y por tanto desplazarse sin rozamiento.

El desplazamiento se consigue gracias a la acción de 4 motores lineales Halbach que están colocados de manera simétrica en la plataforma y con sus estatores invertidos, tal que el estator queda arriba, ensamblado a la parte superior fija de la plataforma, y la matriz de imanes Halbach se encuentra en la parte móvil. Cada motor produce, cuando fluye la corriente por su bobinado, dos fuerzas, una horizontal, responsable del movimiento de la plataforma en X e Y, y otra vertical, que gracias al diseño de la plataforma ayuda a los cojinetes de aire en la tarea de hacer levitar la plataforma. Sin embargo, en vez de fuentes de corriente, se han empleado fuentes de tensión para este proyecto, debido a que la oferta en el mercado es muchísimo mayor y con menores costes.

Además, la plataforma cuenta con un sistema de medición por interferómetros láser 2D para la lectura de la posición de la plataforma, y el cual está conformado por tres cabezales que se encuentran integrados en la parte inferior de la plataforma, y por dos espejos que se encuentran en la parte móvil.

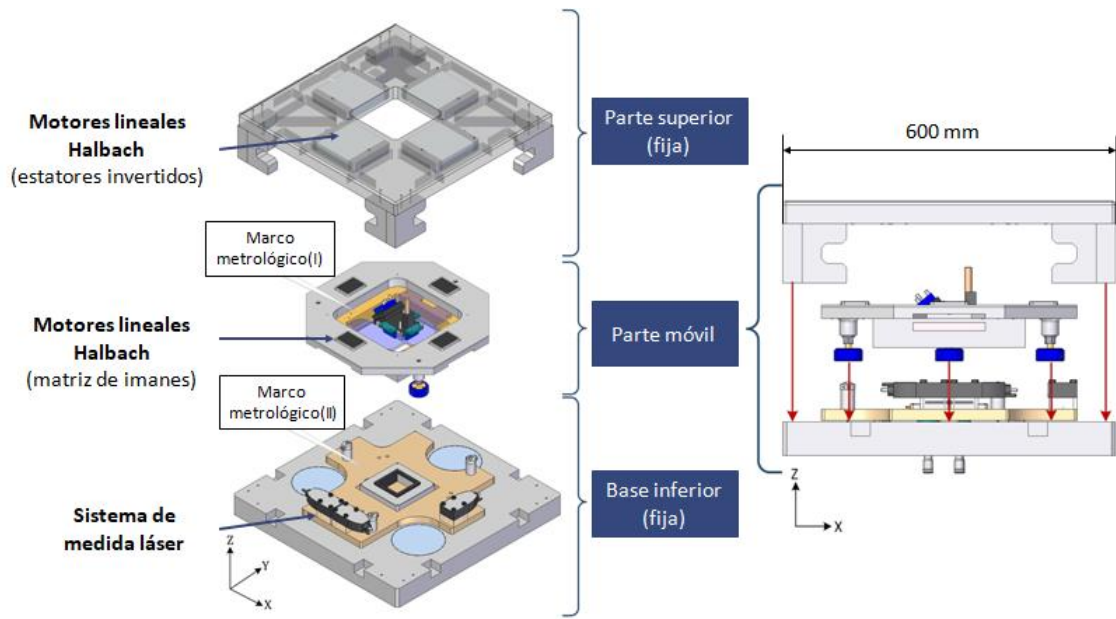


Figura 1: Estructura de la plataforma. Fuente: (1).

## 2.2 Funcionamiento de los motores Halbach

Como se menciona en (3), los 4 motores lineales Halbach empleados fueron desarrollados expresamente para su aplicación en este proyecto en la Universidad de North Carolina en Charlotte, EE.UU, debido a que los motores que hay en el mercado no cumplían con las necesidades del proyecto. La razón es que los motores que se podían adquirir llevaban una guía que impedía el movimiento ortogonal, y para este proyecto se decidió proceder con motores sin guía.

Este tipo de motores son lineales de imanes permanentes; al estar dispuestos el rotor y estator sobre una superficie plana se produce una fuerza lineal longitudinal en vez del característico par de los motores rotativos. También generan levitación magnética: la matriz de imanes de tipo Halbach, que se coloca en paralelo a los bobinados, tiene una distribución especial, de tal manera que aumenta el flujo magnético en uno de los lados y lo nulifica en el otro, es decir solo existe campo magnético en uno de los lados. En la figura 2 se puede apreciar la diferencia entre una matriz de imanes típica y una Halbach.

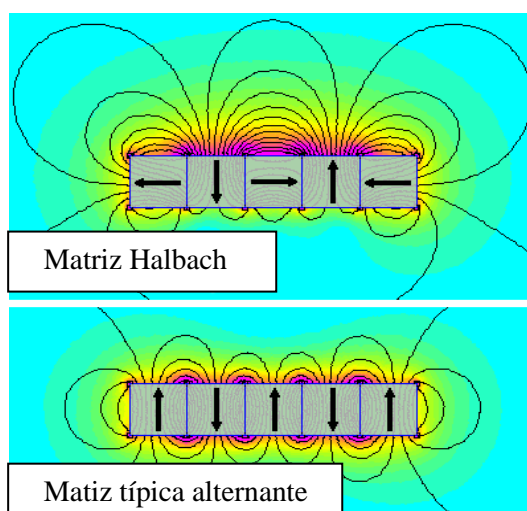


Figura 2: Campo magnético generado por matriz alternante vs matriz Halbach. Fuente: (4)

Esta característica de los motores permite tener un campo mayor en un uno de los lados y al no haber campo en el otro, no afecta al resto de componentes electrónicos, lo cual supone una ventaja. En el motor seleccionado entonces, el flujo magnético se encuentra entre el estator y los imanes y cuando la corriente continua fluye por los devanados senoidales y trifásicos del motor, se produce la interacción entre la corriente y el campo magnético produciéndose las dos fuerzas ortogonales antes mencionadas, como se puede ver en la figura 3.



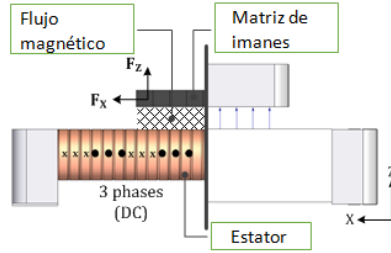


Figura 3: Sección del motor con las fuerzas vertical y horizontal representadas. Fuente: (1)

La magnitud de estas fuerzas depende de la posición relativa entre los devanados y los imanes del motor, además de las intensidades que fluyen por los devanados. La ecuación [1], vista en (1), es la ley del motor y representa la relación que se acaba de mencionar.

$$\begin{bmatrix} F_x \\ F_z \end{bmatrix} = A \begin{bmatrix} \cos(kx_0 + \varphi) & \cos(kx_0 - \frac{2\pi}{3} + \varphi) & \cos(kx_0 + \frac{2\pi}{3} + \varphi) \\ \sin(kx_0 + \varphi) & \sin(kx_0 - \frac{2\pi}{3} + \varphi) & \sin(kx_0 + \frac{2\pi}{3} + \varphi) \end{bmatrix} \begin{bmatrix} I_a \\ I_b \\ I_c \end{bmatrix} \quad [1]$$

En la ecuación [1], A y k son parámetros constantes del motor determinados experimentalmente en (1),  $I_a$ ,  $I_b$  e  $I_c$  son las intensidades de fase que circulan por los devanados,  $F_x$  y  $F_z$  las fuerzas ortogonales que se generan en el motor y  $x_0$  es la posición relativa entre el estator y la matriz de imanes.

En las figuras 4 y 5 se ha representado cada una de las componentes de la fuerza horizontal ( $F_x$ ) y de la fuerza vertical ( $F_z$ ) y la total. Se ve claramente como varían con forma senoidal ambas fuerzas con la posición. Se aprecia además, que ambas,  $F_x$  y  $F_z$ , están desfasadas  $\pi/2$ .

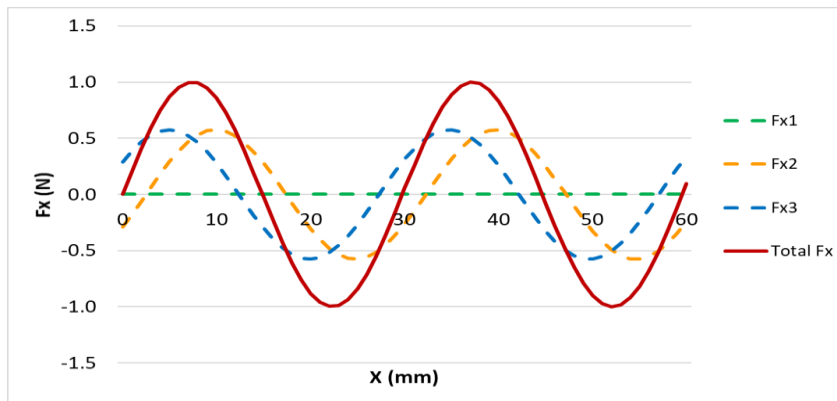


Figura 4: Fuerza horizontal a lo largo de su eje para  $I_a = 0$  A,  $I_b = -0.3593$  A, and  $I_c = 0.3593$  A. Fuente: (3)

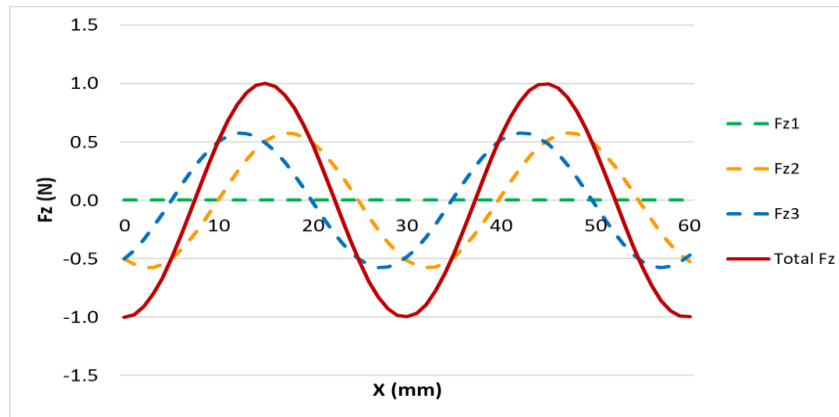


Figura 5: Fuerza vertical a lo largo del eje X para  $I_a = 0$  A,  $I_b = -0.3593$  A, and  $I_c = 0.3593$  A. Fuente: (3)

Sin embargo, para poder calcular todas las intensidades de fase es necesaria la ley de la conmutación, ecuación [3] y se define como la inversa de la ley del motor, ecuación [1]. Para poder calcular las intensidades a partir de unas fuerzas cualesquiera, es necesaria, además, otra ecuación, ya que existen 3 incógnitas y solo 2 ecuaciones. Esta viene dada gracias a la conexión en estrella de las fases del motor y es que la suma de las intensidades de fase es 0, como se ve en la ecuación [2].

$$I_a + I_b + I_c = 0 \quad [2]$$

$$\begin{bmatrix} I_a \\ I_b \\ I_c \end{bmatrix} = \frac{2}{3A} \begin{bmatrix} \cos kx_0 & \sin kx_0 \\ \cos(kx_0 - \frac{2\pi}{3}) & \sin(kx_0 - \frac{2\pi}{3}) \\ \cos(kx_0 + \frac{2\pi}{3}) & \sin(kx_0 + \frac{2\pi}{3}) \end{bmatrix} \begin{bmatrix} F_x \\ F_z \end{bmatrix} \quad [3]$$

La ecuación [3] por tanto, nace de la combinación de las ecuaciones [1] y [2].

En las figuras 4 y 5, además, se puede observar que existen una serie de puntos de equilibrio, cuando la fuerza se hace 0, que son de vital importancia a la hora de diseñar el control, ya que cuando la plataforma llegara a la posición deseada debería permanecer inmóvil, con  $F_x=0$ . Sin embargo, se ha observado en (3) que solo los puntos de equilibrio con pendiente negativa son estables, los otros serían inestables. De esta manera, cuando el motor se encontrase en un punto de equilibrio estable y sufriese pequeñas perturbaciones, volvería al punto de equilibrio. Por tanto, interesa trabajar en la zona lineal que esta próxima a los puntos de equilibrio estable del sistema. Coincide, además, que en estos puntos la fuerza vertical es máxima y positiva, por tanto, contribuiría a la levitación de la plataforma, ya que una fuerza positiva es de atracción entre los imanes y el estator.

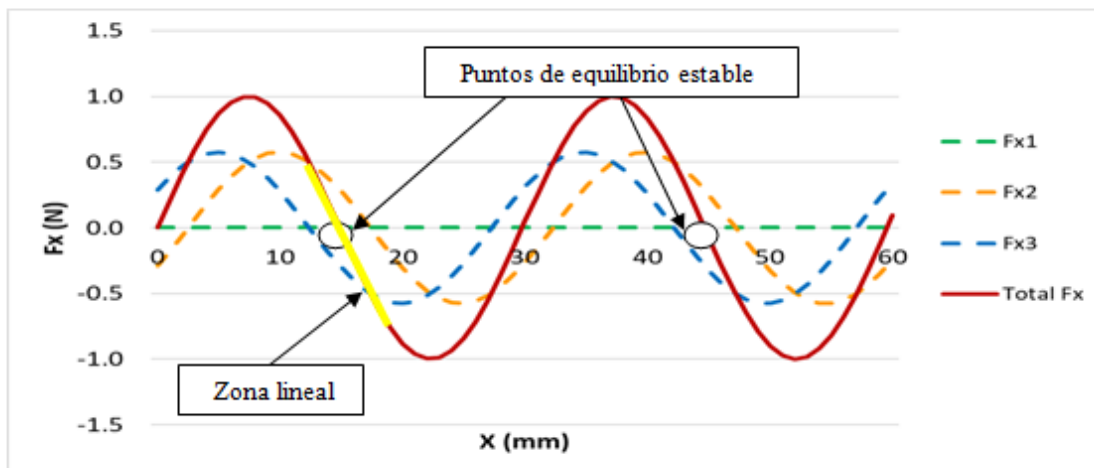


Figura 6: Puntos de equilibrio estable y zona lineal. Fuente: (3)

## 2.3 Sistema de medida de la posición

Al haber planteado una estrategia de control en bucle cerrado con controladores PID, como se detallará más adelante, es necesario realimentar la posición real de la plataforma al controlador de tal manera que pueda calcular la acción requerida a partir del error. Para ello, se utilizarán interferómetros láser RLE 10 de Renishaw.

Los interferómetros láser aplican la metodología de la interferometría, que emplea la interferencia entre ondas, generalmente de luz, para medir distancias con gran precisión. Mediante dos haces de luz (normalmente, un haz desdoblado en dos), se forma un patrón de interferencia donde se superponen los dos haces, pudiéndose detectar pequeños cambios en las diferencias de las trayectorias ópticas, que sería la distancia recorrida de estos haces ya que estas producen cambios notables en el patrón de interferencia.

Un ejemplo de cómo funcionara y como se describe en (5), sería el que se ve en la figura 7, sería un caso de interferometría lineal, que es el que se aplica en este trabajo. Consistiría en que el haz láser sale del cabezal láser dividiéndose en dos rayos (que se reflejan y transmiten) en el desdoblador del haz polarizado. Los haces se reflejan de vuelta desde los dos retroreflectores, uno de los cuales está fijo y el otro se mueve, y se vuelven a combinar en el desdoblador del haz antes de llegar al detector. El uso de retroreflectores asegura que los rayos que vuelven desde los brazos de referencia y medición son paralelos cuando se combinan en el desdoblador del haz. Los haces combinados llegan al detector, donde interfieren entre ellos, de forma constructiva o destructiva.

El procesamiento de señales óptico del detector, permite observar la interferencia de los dos haces. El desplazamiento del brazo de medición, provoca el cambio en la fase relativa de los dos haces. Este ciclo de interferencia destructiva o constructiva hace que la intensidad de la luz de los haces combinados sufra una variación cíclica. Cada vez que se mueve el brazo de medición/retroreflector la mitad de longitud de onda del láser, se genera un ciclo de variación de intensidad de oscuro a iluminado.

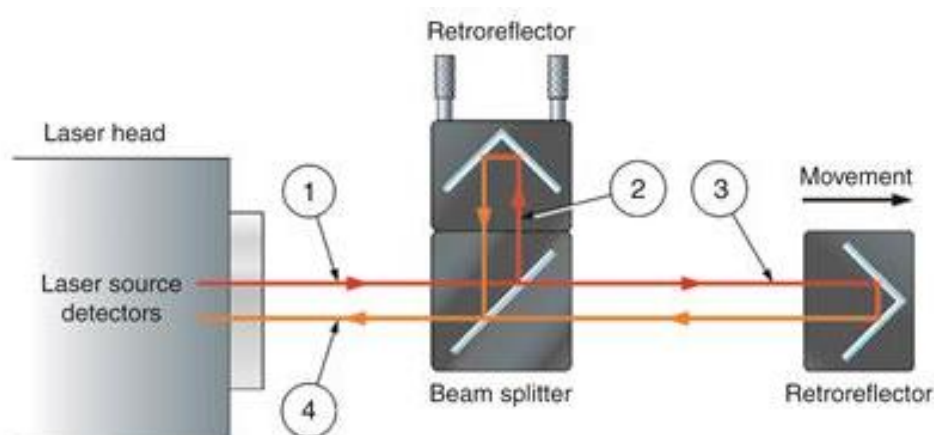


Figura 7: Interferómetro lineal. Fuente: (5)

Se dispone de tres interferómetros láser, uno para medir el desplazamiento de la plataforma en el eje X, y los otros dos se colocan en el eje Y, tanto para medir el desplazamiento en este eje,

como para medir el giro de la plataforma aplicando trigonometría, siendo conocida la distancia entre ambos interferómetros.

Un esquema general de dónde se colocan los interferómetros láser y los espejos para reflejar el haz láser se pueden apreciar en la figura 8.

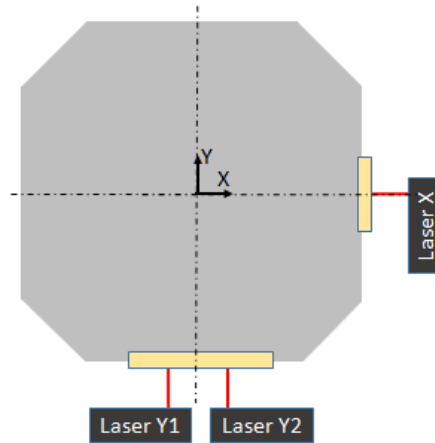


Figura 8: Esquema de la planta de la parte móvil de la plataforma, espejos y láseres. Fuente: Adaptado de (2)

En cuanto a los interferómetros empleados en el presente trabajo, existen algunas diferencias con respecto al esquema mostrado en la figura 7. Para empezar tanto el desdoblador como el retroreflector fijo se encuentran dentro del cabezal, y los espejos del retroreflector móvil no son cúbicos, sino planos, esto es debido a que el desplazamiento se produce perpendicular al haz del láser. Debido a esto, fue necesario el ajuste de los interferómetros láser y cuyo proceso se explicará en más detalle más adelante. Lo más importante para un buen ajuste de los láseres es conseguir que el haz láser sea completamente perpendicular a los espejos donde se tienen que reflejar, para así conseguir una buena calidad en la lectura, de manera que quede como en las figuras 8 y 9.

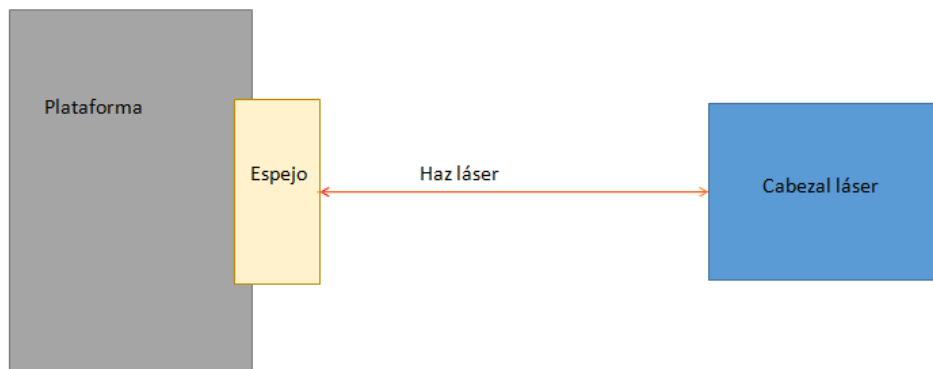


Figura 9: Esquema del sistema láser. Fuente: Elaboración propia.

## 2.4 Tarjetas Texas Instruments

Para completar los elementos que colaboran en el funcionamiento de la plataforma, es necesario hablar de las cuatro tarjetas de Texas Instruments. Estas tarjetas son las encargadas de conectar el ordenador a los motores y de generar y suministrar a cada uno de ellos las tensiones de fase necesarias gracias a la modulación por ancho de pulso (PWM). Estas tarjetas están conectadas al ordenador mediante puerto USB, desde donde se manda su programa para que funcionen, a través del programa Code Composer Studio. Además, las tarjetas están conectadas a los motores a través de las fases, y los neutros están conectados en estrella para cumplir la condición antes mencionada.

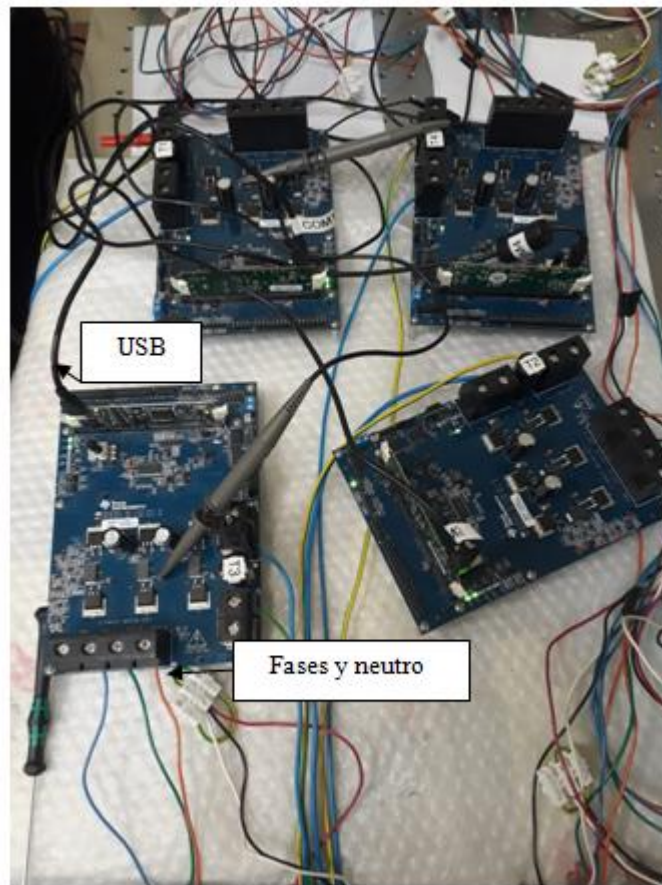


Figura 10: Tarjetas Texas Instruments. Fuente: Elaboración propia.

## 2.5 Montaje experimental

El montaje experimental se encuentra en el laboratorio de metrología de la Universidad de Zaragoza y consta de varias partes, que se pueden apreciar en la figura 11.

Además de la plataforma, ya explicada, hay un ordenador desde el cual se ejecuta MatLab y Simulink, con los distintos programas para controlar la plataforma. El siguiente elemento serían las tarjetas de Texas Instruments, que se encargan de generar los voltajes necesarios a través de modulación por ancho de pulsos (PWM) para los motores, se cuenta con 4 tarjetas, una para cada motor. Se cuenta, también, con un osciloscopio para ver la señal que emiten las tarjetas y comprobar que funcionan de manera correcta.

Se pueden apreciar, en la parte derecha superior de la figura 11, la válvula de aire y los tubos por los cuales llegan el aire y el vacío a los cojinetes para que puedan hacer que la plataforma se eleve y levite.

Dos de los interferómetros láser también se pueden ver, además, en la parte inferior se pueden apreciar sus unidades de control. A la izquierda hay tres botones o pulsadores que hacen que los láseres se referencien de nuevo y la lectura se reinicie tomando como referencia la posición en ese momento, y a la derecha los interpoladores, que, como se describirá más adelante, indican la precisión de la lectura de los láseres gracias al color de la luz que emiten.

Por último, en la parte inferior derecha de la figura 11 se puede observar la fuente de alimentación que se emplea para los láseres.

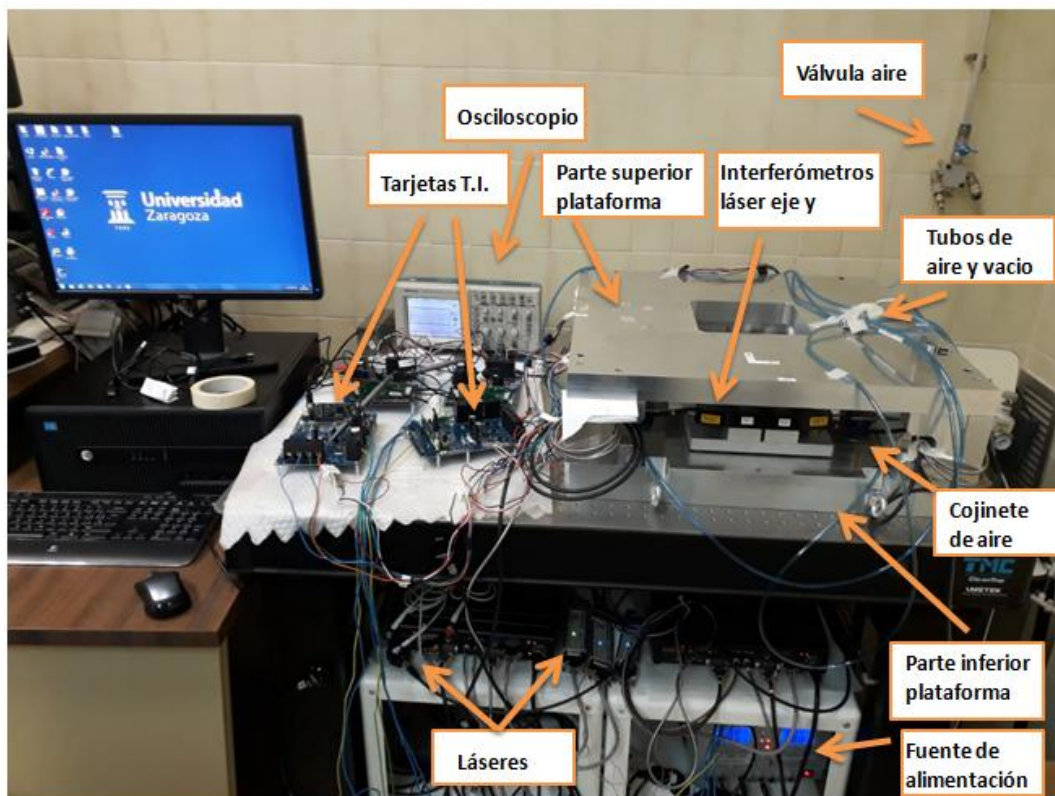


Figura 11: Montaje experimental en el laboratorio. Fuente: Elaboración propia.

Además, en la figura 12 se ha representado un esquema de cómo funcionaría todo este montaje, las tarjetas recibirían la consigna de las tensiones requeridas y mediante PWM producirían las tensiones, que llegarían a cada uno de los motores y a su vez, estos, producirían cada uno tanto la fuerza vertical como horizontal que provocaría un desplazamiento, el cual sería leído por los láseres y enviado al ordenador.

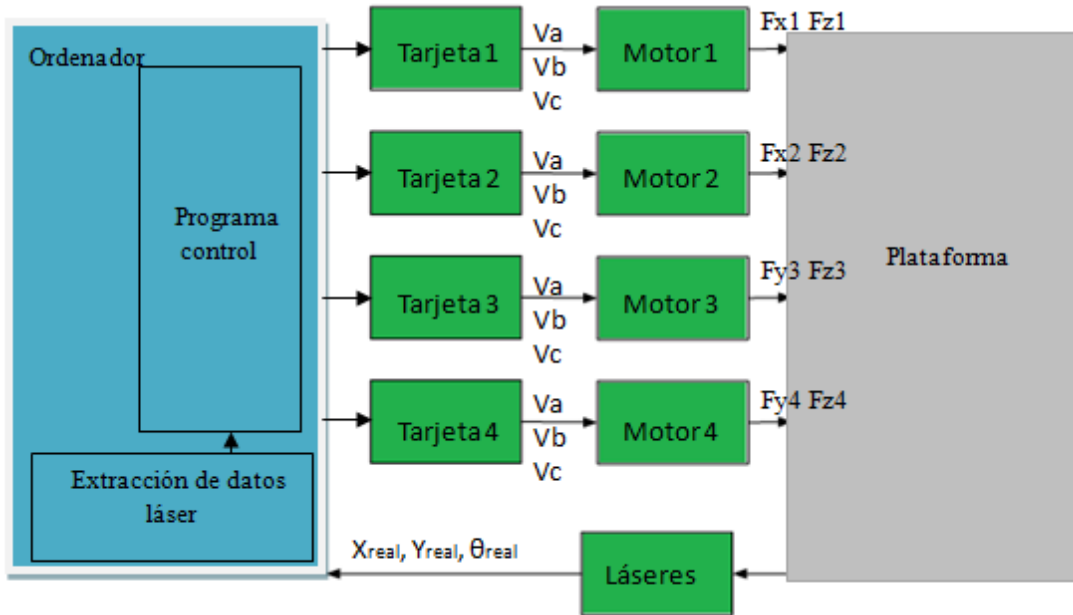


Figura 12. Esquema del montaje del control. Fuente: Elaboración propia.



### 3. Ajuste del sistema de medida de la posición

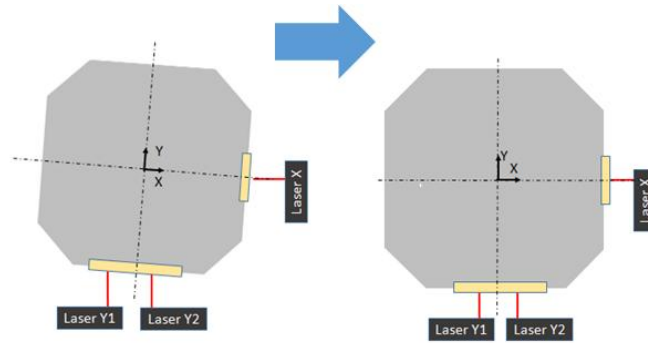
En este apartado se aborda el ajuste de los interferómetros láser. Este ajuste es debido a que los espejos encargados de reflejar el haz láser son planos y por tanto deben colocarse perpendicularmente tanto en Y como en X para asegurar el retorno de los láseres hasta las cabezas lectoras, por lo que se convierte en una tarea ardua. Por lo tanto, una vez que los interferómetros láser están colocados, para una mayor calidad en la lectura de estos es necesario su ajuste.

Para saber la calidad con la que están leyendo, los láseres cuentan con unas luces y un código de colores asociado, que se pueden ver en los interpoladores, y que indican la calidad de la lectura, donde azul y verde indican calidad alta, amarillo media, rojo baja y rojo parpadeante cuando el láser está desalineado y no puede leer. El ajuste, por tanto, trata de conseguir que todos los láseres lean con luz azul o verde y para ello consta de dos pasos.

El primero, se realiza de manera física, sobre el propio interferómetro, variando su inclinación, mediante una manivela, de tal manera que el haz de luz que emite el interferómetro se vea reflejado en el espejo que lleva la plataforma correctamente. Además, se ajustaron las alturas de los cojinetes de aire para corregir la inclinación de la plataforma. Con el primer paso, se consigue que los láseres estén perpendiculares a los espejos, sin embargo, es necesario un segundo paso para que el giro de la plataforma sea nulo fijando la posición inicial de la plataforma móvil.

En este segundo paso, lo que se ajusta es la posición inicial y el giro de la plataforma móvil con el aire encendido y sistema funcionando en bucle abierto, es decir en vez de mover el interferómetro, ahora se moverían los espejos que se mueven solidariamente con la plataforma. El objetivo de este paso es ajustar la posición inicial de tal manera que ambos haces queden perfectamente alineados y perpendiculares con los espejos, como se ve en la figura 13, para de esta manera definir el giro como nulo en la posición inicial. Si la plataforma girara provocaría un desalineamiento en los láseres y, dependiendo de su magnitud, podría provocar la pérdida de la señal y que dejaran de leer. Los láseres, según sus especificaciones pueden leer con errores angulares de hasta  $1,2 \times 10^{-4}$  radianes.

Para realizar el ajuste es necesario hacer uso de Simulink y del programa en bucle abierto que controla la posición inicial de cada motor. El proceso de ajuste es sencillo pero largo, ya que hay que ajustar lentamente la posición de cada motor mientras se observa como varía la calidad de lectura del láser, es decir la indicación de los LED del interpolador. Con esto, al final lo que se consigue es un alineamiento adecuado entre el interferómetro y el espejo de la plataforma mediante la variación de la posición de cada motor. Además, esta posición determinada es la que se usará en el resto de experimentos.



**Figura 13. Ajuste de la posición inicial de la plataforma para conseguir giro inicial nulo. Fuente: Adaptado de (2)**

Este sistema láser, además, será el que límite la velocidad del sistema, ya que con todos los láseres leyendo a la vez, los tres, el tiempo entre lecturas se encontraba entre 0,16 s y 0,19 s, lo que limitará la velocidad del PID al no poder leer más rápido.

# 4. Estrategia de control

## 4.1 Antecedentes del control 2D y estrategia aplicada

Tras haber explicado el funcionamiento de la plataforma, en este apartado se definirá la estrategia de control que se va a seguir y que condicionará el trabajo.

Previamente, es necesario explicar que un control en bucle abierto de la plataforma no funcionaría. Teóricamente, la plataforma al enviarle mediante el programa la posición deseada debería llegar a esta exactamente, gracias a lo explicado anteriormente de los puntos de equilibrio, comportándose como un servosistema. Sin embargo, al existir errores debidos a que no se generan directamente las intensidades, sino que se generan las tensiones de fase, o también debido a la existencia de cierto ruido en las intensidades, la plataforma nunca llega a la posición requerida con exactitud. Es por esto que un control en bucle cerrado con realimentación de la posición es necesario.

Además, mencionar que la estrategia de control empleada parte de dos trabajos previos (1) (2), en las cuales se llevó a cabo con éxito el control de uno de los motores de la plataforma y además se planteó y realizó una estrategia de control en 2D y que es la que sigue en este trabajo.

La estrategia de control empleada, hace uso de 3 controladores PID, uno para controlar la posición de la plataforma en el eje X, otra para controlar la posición del eje Y, y el último sería el encargado de que el giro de la plataforma sea nulo, necesario para que los láseres no dejen de leer o evitar que lean con menor calidad. Las siguientes ecuaciones, vistas en (2), serán las que rijan el comportamiento de cada uno de los motores. Las fuerzas que se calculan son las horizontales y son las encargadas del movimiento plano, la otra componente de fuerzas que genera cada motor, la  $F_z$ , es constante y se ajusta a 2 N, en principio, para este trabajo.

$$F_{m1} = F_x/2 + T_z / (4 * R) \quad [4]$$

$$F_{m2} = F_x/2 - T_z / (4 * R) \quad [5]$$

$$F_{m3} = F_y/2 - T_z / (4 * R) \quad [6]$$

$$F_{m4} = F_y/2 + T_z / (4 * R) \quad [7]$$

La fuerza horizontal de cada motor, por tanto, depende de la mitad de la fuerza horizontal de su eje (la que genera el PID de X o Y) y del par motor (que genera el PID de giro) dividido entre el radio multiplicado por 4, donde el radio es la distancia entre dos motores opuestos partida por dos. En la figura 14, se puede apreciar un esquema de cómo sería el control a implementar para la plataforma en Simulink.

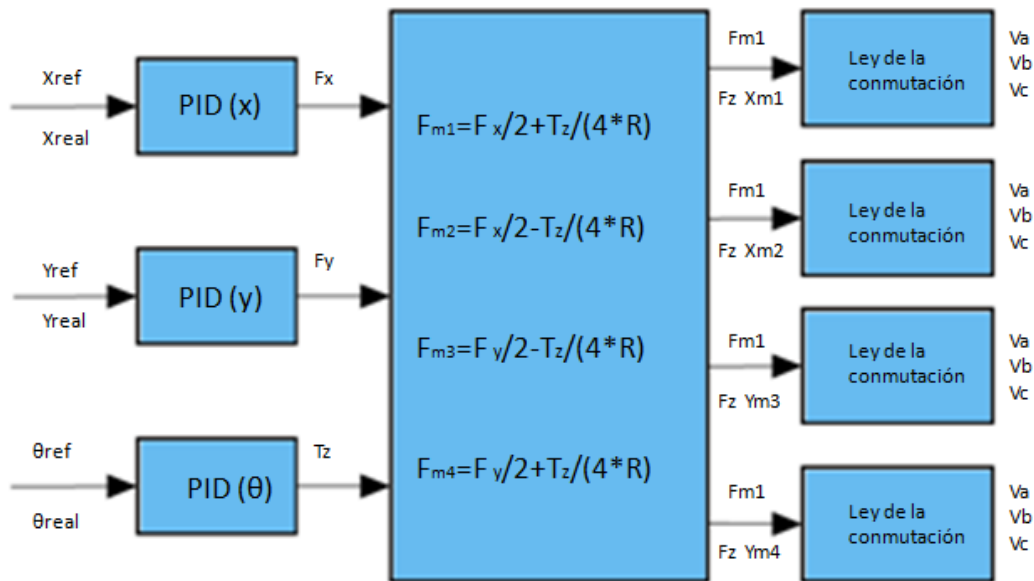


Figura 14. Esquema de control para Simulink con los 3 PID. Fuente: (2).

El proceso es el siguiente, cada PID recibiría, por un lado, la posición a la cual se desea desplazar la plataforma, el de X recibiría la coordenada en X, el de Y la de Y, y en el del giro, la referencia, que siempre será 0, al desear un giro nulo. Además, los PID reciben lo posición real de la plataforma, en X y en Y, y el giro y con ello calcularían la acción requerida a partir del error. De esta manera los PID calculan las fuerzas en X y en Y, y el par necesario. Así, con las ecuaciones [4], [5], [6] y [7], se calcula la fuerza que le llegaría a cada motor. A los motores 1 y 2, responsables del movimiento en el eje X, les llegaría la mitad de la fuerza del eje X, para intentar que se desplacen de manera simétrica, y el par motor, que trata de compensar si algún motor va más rápido que el otro sumando o restando fuerza según se necesite.

Así, la componente horizontal ya estaría calculada, si a esto se le suma la posición inicial de cada motor y la fuerza vertical, que son conocidas, se podrían calcular los voltajes necesarios para controlar la plataforma a través de las tarjetas de Texas Instruments, que generarían los voltajes necesarios mediante PWM.

## 4.2 Modelado del sistema

En este apartado se explica el proceso de obtención de las funciones de transferencia para su posterior uso en el desarrollo del control de la plataforma. Para ello, en primer lugar, se explicará el proceso seguido para obtener un modelo matemático aproximado para los desplazamientos en el eje X e Y, ya que al ser muy similares se sigue el mismo proceso, y por último se explicará cómo se determinó la función de transferencia para el giro. Además, para realizar el modelado del sistema se ha empleado la información presentada en (6), (7) y (8).

### 4.2.1 Funciones de transferencia para los desplazamientos en X e Y

Para poder obtener la función de transferencia de ambos movimientos, se parte de la ecuación ya obtenida en (3), para un motor lineal en una dirección y en una guía lineal. Esta ecuación también se puede aplicar para este caso debido a que no hay más fuerzas actuando sobre la plataforma móvil, a parte de las generadas por los motores, a que ambos motores son muy similares y que aunque ahora no exista una guía como tal, los motores opuestos actúan con esa función. Sin embargo, en este caso al operar con dos motores, la ecuación queda de esta manera:

$$\frac{X(s)}{X_{eq}(s)} = \frac{2k/m}{s^2 + \frac{b}{m}s + \frac{2k}{m}} \quad [8]$$

Esta función de transferencia, que caracteriza a ambos desplazamientos, incluye la masa de la plataforma (m), la fricción viscosa (b) y la constante (k), que relaciona la fuerza con la posición, multiplicada por dos por ser dos motores los encargados del movimiento en cada eje. Se puede apreciar que la función de transferencia es de segundo orden sin ceros y además, los valores de k y m se pueden determinar de manera teórica, lo cual es de gran ayuda para la determinación de los valores finales.

Para poder obtener un modelo que aproxime bien cada movimiento, se procedió a realizar ensayos para ver como respondía el sistema y posteriormente analizar esa respuesta con el fin de obtener una función de transferencia.

La manera en la que estos experimentos se llevaron a cabo fue mediante la aplicación de un escalón de 1 mm al sistema en bucle abierto manteniendo la fuerza en el eje Z en 2 N. Para realizar este escalón, lo que hizo fue cambiar el punto de equilibrio del sistema, cambiando la distribución de intensidades, de esta manera se crea un nuevo punto de equilibrio en la posición deseada y la plataforma se desplazará hasta este.

El sistema se comporta como un muelle, es decir la plataforma para cada combinación de intensidades tiene un punto de equilibrio distinto y al igual que un muelle, si no se produce una fuerza demasiado fuerte, este siempre tenderá a volver al equilibrio. Con esto, lo que se quiere indicar es que la plataforma siempre va ir al punto de equilibrio, por lo tanto al analizar los resultados hay que considerar que no existe error de posición en los desplazamientos puesto que la plataforma siempre se va a quedar en el punto de equilibrio. La diferencia entonces que surja entre la referencia, es decir, entre el valor de posicionamiento deseado, y el real será debido al

ruido y el error que existe en las intensidades generadas y a las perturbaciones externas, que provocan este error de posición, pero que sería inadecuado tener en cuenta a la hora de determinar una función de transferencia que represente fielmente el desplazamiento de cada eje. El cálculo de la constante del “muelle” ( $k$ ) se realizó de manera teórica en base a esto, mediante el cambio del punto de equilibrio sin alejarse de la zona lineal, donde  $k$  es la pendiente, de manera que al cambiar el punto de equilibrio habiendo aplicado otra fuerza horizontal, la plataforma se desplaza. Con ese desplazamiento y la fuerza se puede calcular la constante  $k$  gracias a la fórmula  $F=k \cdot x$ , similar a la del muelle. Este cálculo dependía de la fuerza vertical que se considerase. Para distintas fuerzas horizontales la  $k$  es constante, siempre que no se salga de la zona lineal, ya que el desplazamiento cambia proporcionalmente, por lo que en la siguiente tabla se puede ver como varía  $k$  con la fuerza vertical.

$F_z$ (N)	$k$ (N/m)
<b>1</b>	213
<b>2</b>	426
<b>3</b>	639

**Tabla 1: Relación  $F_z$  con  $k$ . Fuente: Elaboración propia.**

Se escogió aplicar un escalón de 1 mm debido a que es un valor suficientemente grande como para obtener una cantidad de datos adecuada y a la vez suficientemente pequeño como para no salirse de la zona lineal en la que interesa trabajar, ya que si se sale de esta el sistema dejaría de ser un segundo orden y todo lo que se realiza en este trabajo está enfocado a este segundo orden. Además, toda la tarea del control sería mucho más compleja, ver figura 6.

Para cada desplazamiento se realizaron 15 experimentos y debido a que los resultados obtenidos diferían levemente, aunque no significativamente, se realizó una media de las características más importantes que definen la respuesta de un sistema de segundo orden, como son el tiempo de respuesta y la sobreoscilación, y se procedieron a realizar los cálculos con estas medias.

### 4.2.1.1 Desplazamiento en el eje X

Como se ha indicado antes, se realizaron 15 experimentos con un escalón de valor 1 mm, y las características más importantes que arrojaron dichos experimentos fueron que el sistema era un sistema subamortiguado, con una sobreoscilación (SO) media del 40% y un tiempo de establecimiento o de respuesta de 1,4s.

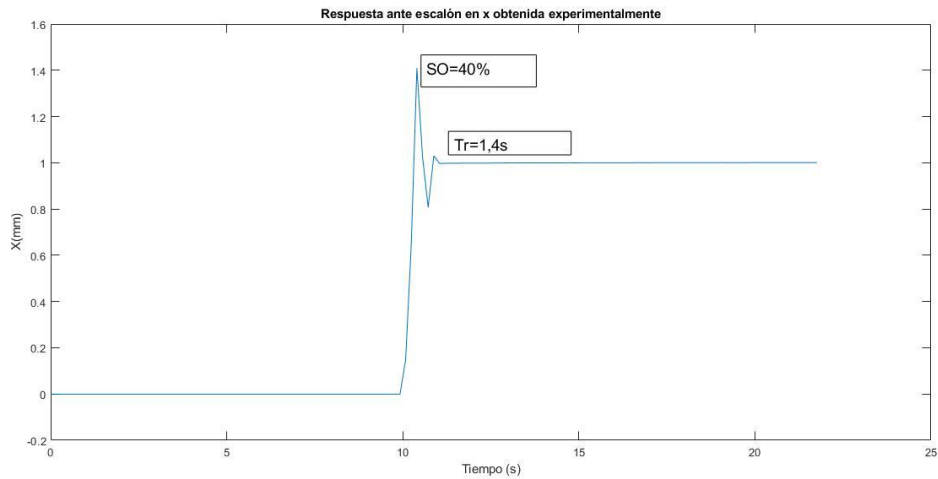


Figura 15: Resposta ante escalón experimental obtenida. Fuente: Elaboración propia.

Los resultados se analizaron de dos maneras diferentes, calculando la función de transferencia directamente a partir de la SO, el tiempo de respuesta del sistema y los parámetros conocidos ( $k$  y  $m$ ) y también a partir la herramienta de Matlab llamada *System Identification Toolbox*.

### Cálculo directo

Para el cálculo directo de la función se partió del valor de  $k$  teórico para  $F_z=2$  N, siendo el resultado de  $k=426$  N/m, y del de  $m$  que se conocía de antemano que su valor era 13,2 Kg por lo que analizando la sobreoscilación media (40%) y el tiempo de respuesta medio (1,4s) de los experimentos, se calcula la primera aproximación del modelo.

Partiendo de la ecuación de un sistema de segundo orden, con las fórmulas de la SO y del tiempo de respuesta, se determinaron los valores del factor de amortiguamiento y de la frecuencia natural del sistema.

Función de transferencia de segundo orden:

$$\frac{Y(s)}{X(s)} = \frac{\omega_n^2}{s^2 + 2\xi\omega_n s + \omega_n^2} \quad [9]$$

Formulas de la sobreoscilación y del tiempo de respuesta o de establecimiento:

$$SO=100*\exp\left(\frac{-\xi\pi}{\sqrt{1-\xi^2}}\right) \quad [10]$$

$$t_r=\frac{\pi}{\xi\omega_n} \quad [11]$$

Empleando estas fórmulas y con los valores antes mencionados los resultados para  $\xi$  y  $\omega_n$  son 0,28 y 8,03 rad/s respectivamente y entonces la función de transferencia calculada de manera directa es la siguiente:

$$\frac{X(s)}{X_{eq}(s)} = \frac{64,54}{s^2 + 4,5s + 64,54} \quad [12]$$

En la figura 16, se puede observar cómo responde la función de transferencia obtenida ante un escalón de 1 mm, viendo que las características antes dichas (SO y tiempo de respuesta) se cumplen.

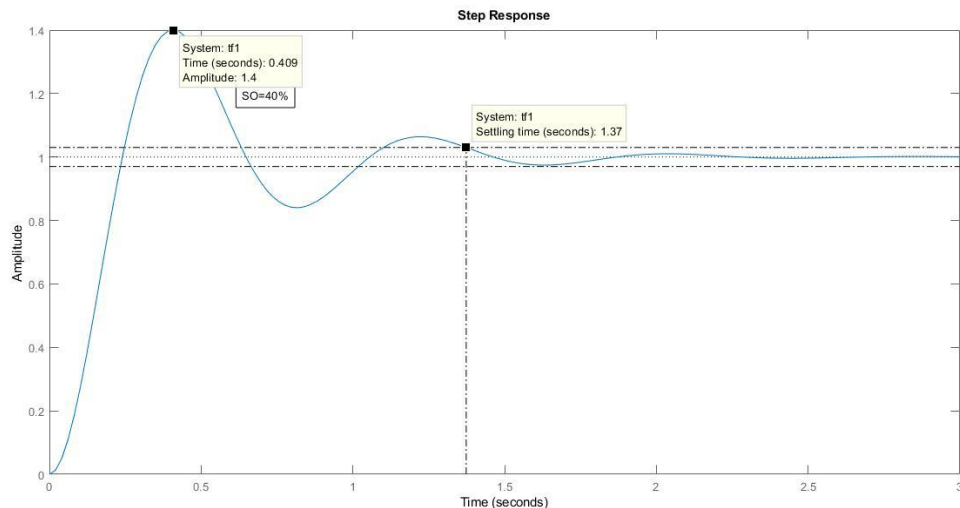


Figura 16: Respuesta ante escalón del modelo obtenido. Fuente: Elaboración propia.

## Cálculo con la herramienta de Matlab

Para comprobar la exactitud de este cálculo, se determinó también la función de transferencia mediante la herramienta de Matlab antes mencionada.

*System Identification Toolbox* es una herramienta que permite construir un modelo matemático a partir de datos experimentales de entrada y salida de un sistema. Permite usar tiempo discreto o tiempo continuo y además, puede obtener directamente la función de transferencia de un sistema a partir de los datos mencionados anteriormente. Esta herramienta se empleará en ambos desplazamientos para corroborar los cálculos realizados.

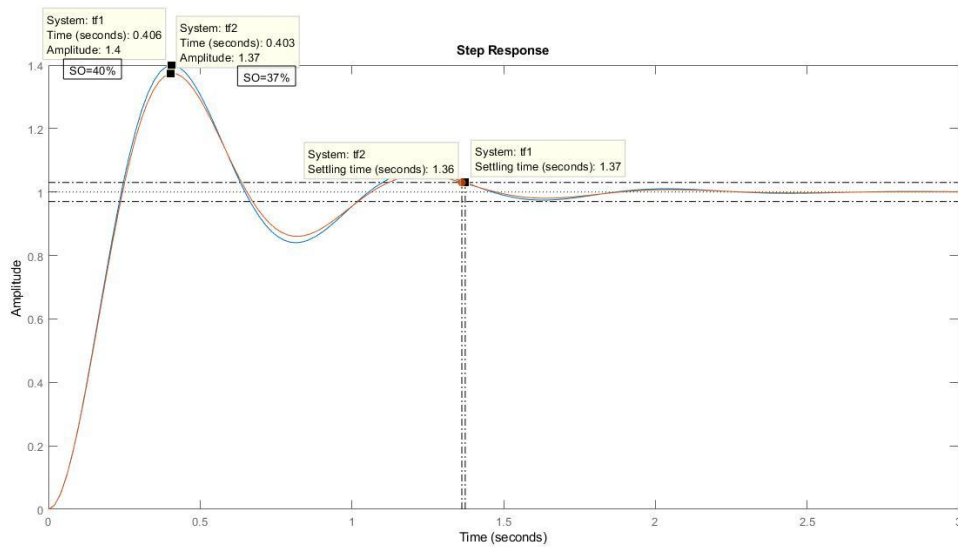
Para ello se introdujeron los datos tanto de entrada (escalón de 1 mm) como de salida, obtenidos en el experimento que más se aproximaba a la media, y el tiempo de muestreo empleado, esto es el tiempo que hay entre lecturas del láser, usando un láser solo al mismo



tiempo, (0,092 s). Esta herramienta, además, permite ajustar de antemano el número de ceros y polos del sistema, lo cual es perfecto ya que en este caso es conocido, ningún cero y dos polos. Con todo esto, el resultado obtenido determinó una función de transferencia con un 95% de precisión, es decir que la curva obtenida con la herramienta se aproximaba a la obtenida experimentalmente en un 95%. La función de transferencia obtenida entonces con la herramienta de Matlab es la siguiente:

$$\frac{X(s)}{X_{eq}(s)} = \frac{65,7}{s^2 + 4,487s + 65,8} [13]$$

Como se puede apreciar ambas funciones son muy parecidas, además en la figura 17 se pueden observar las dos respuestas al escalón de ambas funciones de transferencia obtenidas, y como cabía esperar ambas son muy similares.



**Figura 17: Comparación de las respuestas a escalón de los dos modelos obtenidos. Fuente: Elaboración propia.**

Por último, se comparan las gráficas de la respuesta obtenida experimentalmente del sistema a un escalón de 1 mm, con la obtenida a partir de la herramienta de Matlab para el mismo escalón, y como se puede observar ambas gráficas son muy parecidas y lo más importante, que los parámetros de SO y  $t_r$  son casi idénticos. En rojo estaría representada la respuesta obtenida experimentalmente del sistema y en azul la obtenida simulando la respuesta al escalón de la función de transferencia obtenida.

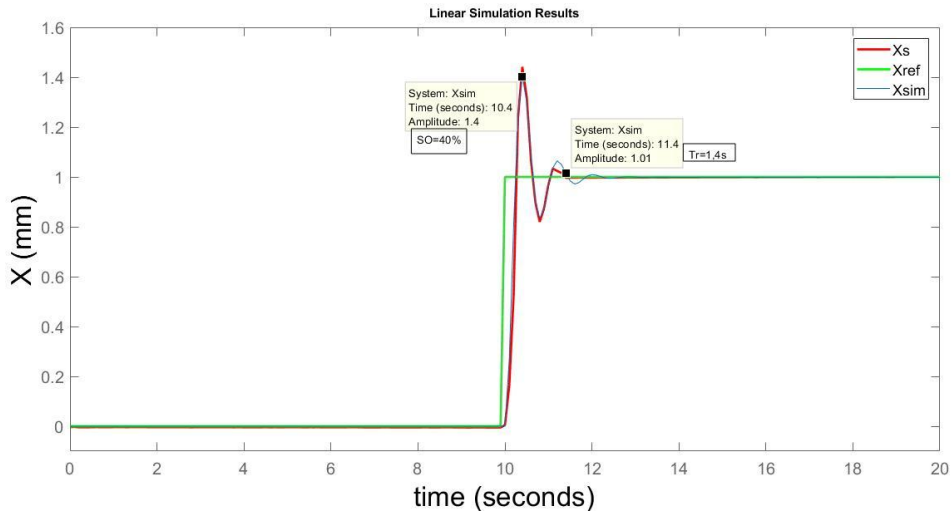


Figura 18: Comparación entre respuesta ante escalón obtenida de manera experimental y la del modelo obtenido. Fuente: Elaboración propia.

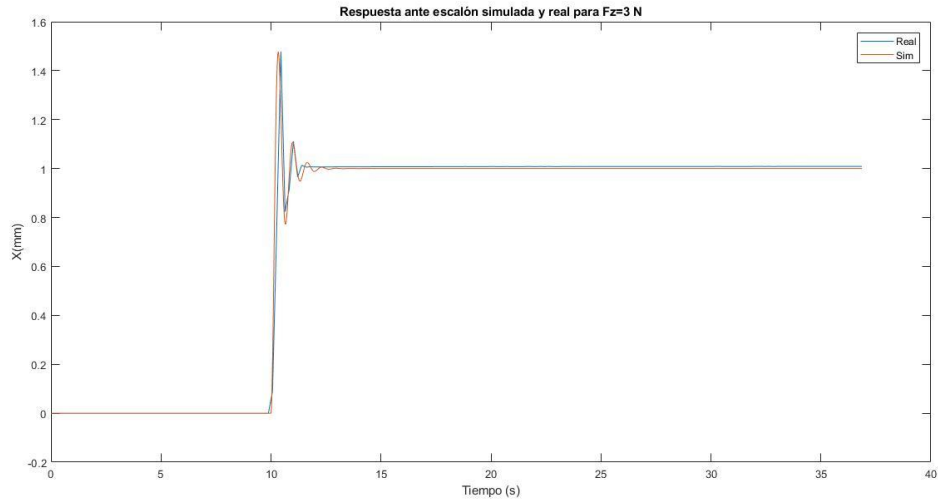
## Comprobación de la función de transferencia del desplazamiento en X

Para comprobar que la función de transferencia obtenida es un modelo matemático adecuado para el desplazamiento en X de la plataforma, se realizaron dos comprobaciones.

La primera de estas comprobaciones se basaba en variar el valor de la Fz aplicada al sistema y ver como afectaba a la respuesta del sistema ante un escalón unitario de nuevo. Esto trataba de demostrar el significado físico de la función de transferencia, es decir, si se variaba el valor de k, los únicos valores que se deberían ver afectados de la función de transferencia serían el numerador y el término independiente del denominador. Como además, k es un valor que se puede determinar de manera teórica, sin cometer demasiado error, se ensayaron las respuestas al escalón unitario para los valores de  $F_z=3\text{ N}$  y  $F_z=1\text{ N}$ .

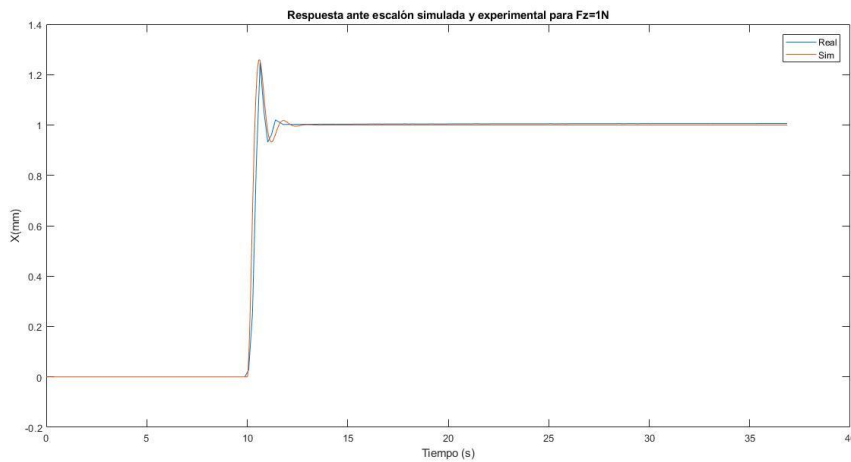
Tras determinar los valores teóricos de k, para  $F_z=3\text{ N}$  su valor sería de  $639\text{ N/m}$  y para  $F_z=1\text{ N}$  sería de  $213\text{ N/m}$ , los valores para el numerador y el término independiente del denominador, ecuación 8, pasaron a ser 96 y 32 respectivamente. Una vez calculados estos términos, y manteniendo constante el término b/m de la función de transferencia para el desplazamiento en X, se simuló su respuesta al escalón unitario y se comparó con la obtenida de manera experimental.

En la figura 19 se puede observar en rojo la respuesta obtenida de manera experimental y en azul la respuesta simulada, las respuestas son muy similares siendo los parámetros como la SO y el tiempo de respuesta casi idénticos.



**Figura 19: Comparativa entre respuesta obtenida experimentalmente y con el modelo ante escalón con  $F_z=3\text{N}$ . Fuente: Elaboración propia.**

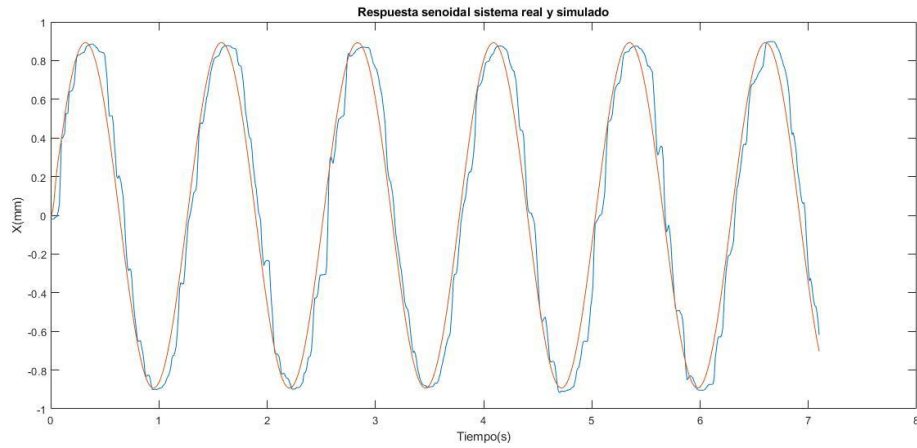
Para  $F_z=1$  se realizó el mismo proceso, viendo los resultados en la figura 20, y se pudo apreciar como la sobreoscilación disminuía bastante y eso se veía reflejado tanto en la respuesta experimental como en la simulada. Además, ambas gráficas son muy similares pudiéndose ver entonces idóneamente el efecto que tiene la  $k$  sobre la función de transferencia y demostrando así su sentido físico.



**Figura 20: Comparativa entre respuesta obtenida experimentalmente y con el modelo ante escalón con  $F_z=2\text{N}$ . Fuente: Elaboración propia.**

La segunda comprobación que se realizó fue introduciendo una entrada senoidal de amplitud  $0,89\text{ mm}$  tanto al sistema real como al teórico del desplazamiento en  $X$  para ver la respuesta de ambos sistemas.

En la figura 21 se puede observar como ambas respuestas son casi idénticas, en rojo se puede ver la respuesta del sistema simulada, con mucho menos ruido, como cabía esperar de una simulación, mientras que en azul se ve representada la respuesta de la plataforma real.



**Figura 21 Comparativa entre respuesta obtenida experimentalmente y con el modelo ante entrada senoidal. Fuente: Elaboración propia.**

Con todo lo visto se puede confirmar que la función de transferencia obtenida es un modelo adecuado para representar el desplazamiento en  $X$  de la plataforma.

### 4.2.1.2 Desplazamiento en el eje Y

De igual manera que para el eje X se procedió para el movimiento en Y. Para este caso, sin embargo, la media de la sobreoscilación fue del 35%, por lo que el cálculo de la función de transferencia de manera directa varía un poco en el término de primer orden del denominador, el cual depende del factor de amortiguamiento para una función de transferencia de segundo orden. La función queda entonces:

$$\frac{Y(s)}{Y_{eq}(s)} = \frac{64,54}{s^2 + 4,8s + 64,54} [14]$$

A continuación, a partir de un experimento medio, como antes, se determinó con la herramienta de Matlab la función de transferencia, en este caso la precisión de los resultados fue menor, del 93%, y se observa como difiere un poco más que la calculada para el eje X.

$$\frac{Y(s)}{Y_{eq}(s)} = \frac{69,8}{s^2 + 4,76s + 69,8} [15]$$

Para este caso, la función de transferencia elegida va a ser la calculada a través de Matlab, ya que la calculada de manera directa puede implicar errores en el cálculo teórico de la k. Además, aunque teóricamente las dos funciones deberían ser iguales, tanto para X como para Y, la realidad es que los motores no son idénticos y la plataforma tampoco es realmente simétrica, dos lados contiguos cuentan con los espejos empleados por los interferómetros láser, por lo que cabía esperar ligeras diferencias, como es el caso, en sus funciones de transferencia.

A continuación, se muestran las respuestas ante un escalón de valor la unidad tanto del sistema como de la función de transferencia elegida para el eje Y.

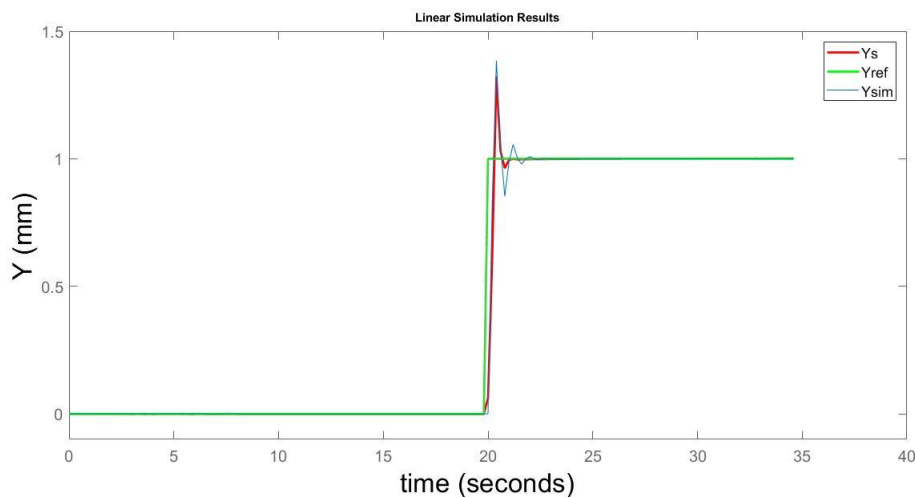
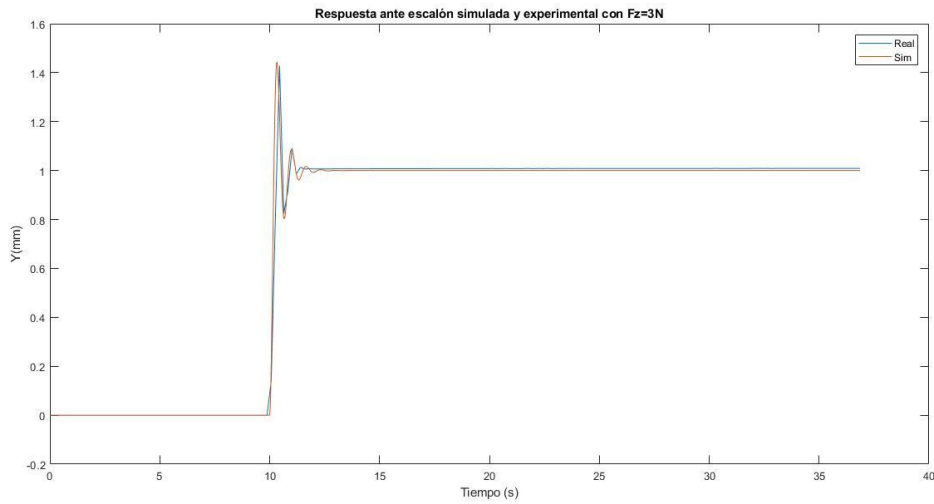
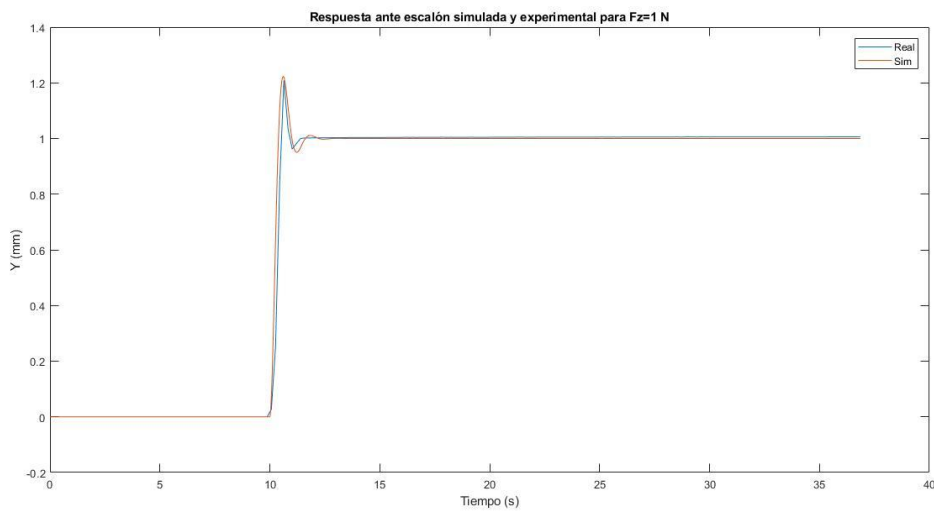


Figura 22: Comparación entre respuesta ante escalón obtenida de manera experimental y la del modelo obtenido. Fuente: Elaboración propia.

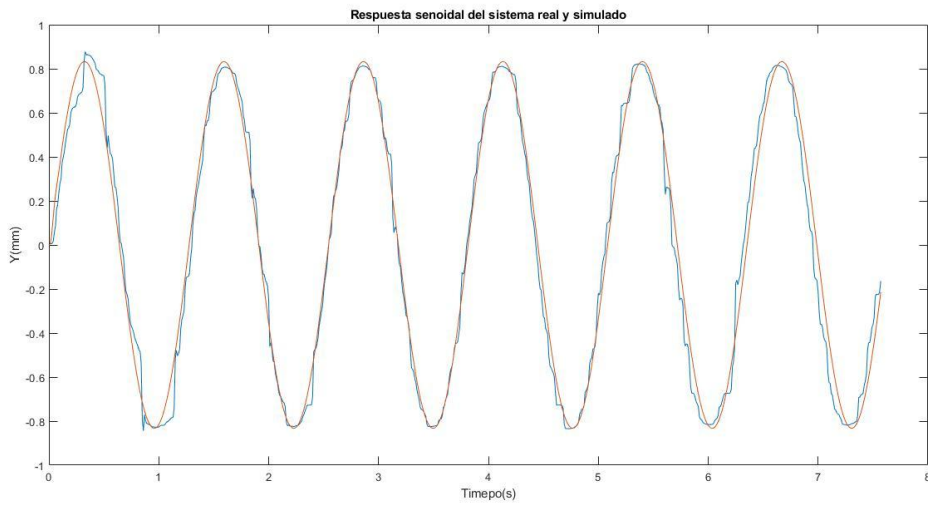
De igual manera que para el desplazamiento en el eje X, se comprobó la idoneidad de la función de transferencia determinada para el desplazamiento en Y de la plataforma. Se realizaron las mismas comprobaciones, variando la fuerza vertical e introduciendo una entrada senoidal en vez del escalón para ver como respondía el sistema. Las siguientes gráficas indican los resultados obtenidos y análogamente al eje X, se da por válida la función obtenida.



**Figura 23: Comparativa entre respuesta obtenida experimentalmente y con el modelo ante escalón con  $F_z=3\text{N}$ . Fuente: Elaboración propia.**



**Figura 24: Comparativa entre respuesta obtenida experimentalmente y con el modelo ante escalón con  $F_z=1\text{N}$ . Fuente: Elaboración propia.**



**Figura 25: Comparativa entre respuesta obtenida experimentalmente y con el modelo ante entrada senoidal.**  
**Fuente: Elaboración propia.**

## 4.2.2 Obtención de la función de transferencia del giro

Para obtener un modelo matemático adecuado se procedió de manera similar, se volvió a usar el control en bucle abierto. Sin embargo, ahora se introducirá el mismo desplazamiento en los dos ejes, pero para cada par de motores en sentido opuesto, tratando de que la plataforma gire. Es decir, los cuatro contribuyen al giro como se puede ver en la figura 26.

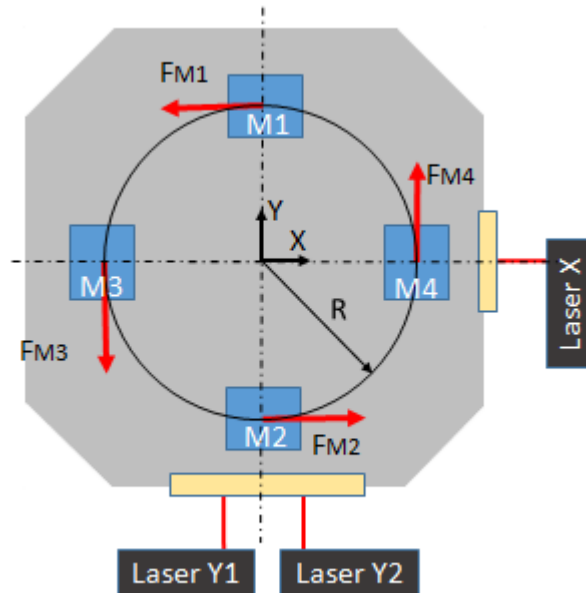


Figura 26: Esquema empleado para girar la plataforma. Fuente: (2).

Además, para el cálculo de la función se tendrá en cuenta cómo responde la plataforma desde que esta girada el máximo número de radianes sin que afecte a la calidad de la lectura hasta 0, debido a que ese es el objetivo posteriormente del PID. Por último a tener en cuenta, aunque la función de transferencia teórica del giro también sea conocida con anterioridad, en este caso simplemente se calculará mediante la herramienta de Matlab. La función de transferencia teórica para el giro tiene la siguiente forma.

$$\frac{\theta_z(s)}{\theta_{z,eq}(s)} = \frac{4KR^2/I_z}{s^2 + \frac{b}{I_z}s + \frac{4KR^2}{I_z}} [16]$$

De acuerdo al fabricante de los interferómetros láser, el giro máximo permitido sin que afecte a la calidad de la medición de estos es de  $1,2 \times 10^{-4}$  radianes, sin embargo, experimentalmente se ha determinado que este valor es un poco más elevado, y, por lo tanto, para los experimentos se han introducido a la plataforma escalones de entre  $1,8 \times 10^{-4}$  y  $2 \times 10^{-4}$  para así poder tener una señal de salida más clara al reducirse ligeramente el ruido.

De esta manera, y al igual que anteriormente, se realizaron 15 experimentos aplicando un escalón en radianes como el descrito anteriormente y posteriormente se devolvía la plataforma a 0, es decir a giro nulo. De este último escalón es del cual se analizaron los datos. Tras introducir los datos del experimento que más se aproximaba a la media en la herramienta de Matlab, la función de transferencia que determinó fue la siguiente.



$$\frac{\theta(s)}{\theta_{eq}(s)} = \frac{321,3}{s^2 + 140,2s + 321,3} [17]$$

A diferencia de las anteriores funciones determinadas, y aunque sea igualmente un sistema de segundo grado, es un sistema sobreamortiguado, sin sobreoscilación. En la figura 27 se representan las respuestas al escalón del sistema simulado y del sistema real. Se observa que la exactitud es menor que en las anteriores funciones. Sin embargo, teniendo en cuenta que los valores del escalón introducido son del orden de  $10^{-4}$  radianes y por ello la presencia de ruido afecta mucho a los datos obtenidos se considera una función adecuada para modelar el giro de la plataforma.

En la siguiente figura se puede la comparación entre las respuestas de la función calculada y la determinada experimentalmente. En la respuesta del sistema teórico, en color rojo, se puede apreciar incluso que, al ser valores tan pequeños, ciertas vibraciones cuando el sistema llega al valor estacionario. Esto aporta mayor similitud entre ambas respuestas, además de que, como se ve en la imagen, ambas respuestas son muy parecidas.

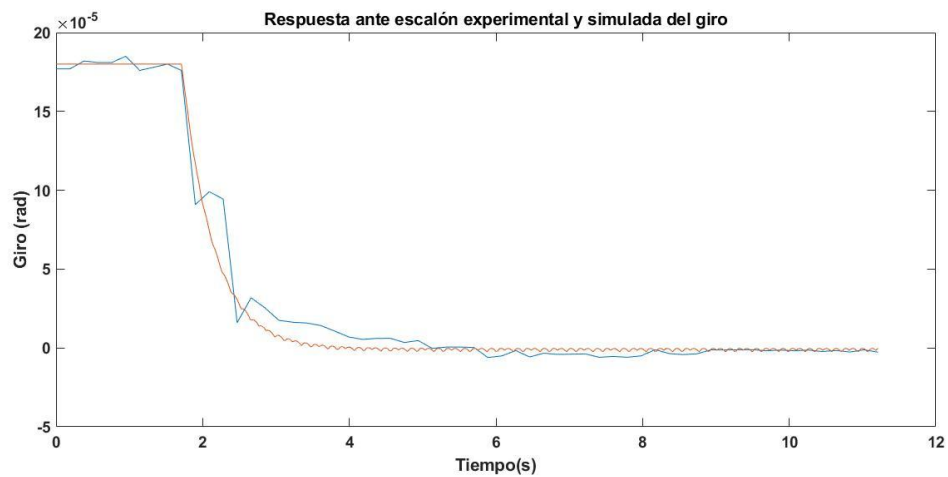


Figura 27: Comparativa entre respuestas ante escalón del sistema real y del modelo obtenido. Fuente: Elaboración propia.

## 4.3 Diseño e implementación de los PID

Para el diseño de los controladores PID y su posterior implementación se siguió un proceso que contó con varias etapas. En primer lugar, se estudió mediante simulación como afectaban distintos controladores PID sobre las plantas modeladas en el apartado anterior. Estos efectos de los controladores se estudiaron individualmente, es decir, se estudió el efecto de un controlador PID sobre la planta modelada para el desplazamiento en X, el de otro PID para la planta del desplazamiento Y, y por último otro para la del giro, sin interactuar unos con otros.

Tras estas primeras simulaciones, se probó cómo funcionaban estos controladores sobre la plataforma. Los objetivos para los PID eran sobretodo dos, eliminar lo máximo posible la sobreoscilación que producía en bucle abierto y que era bastante significativa tanto para los desplazamientos en el eje X como para los del eje Y, y por otro asegurarse de que la plataforma llegaba siempre a la posición deseada en un tiempo aceptable y con el menor giro posible. Tras realizar un ajuste fino, se consiguieron controladores que realizaban la función requerida, tanto para el control a velocidad constante con rampa como para el que usaba escalones, como se verá más adelante. Después de su diseño, se analizó como respondía ante distintas entradas,  $F_z$  y velocidades. Por último, se estudió el ruido que generaba este controlador sobre el eje Z a través de un sensor confocal.

### 4.3.1 Ajuste de los PID

Para el diseño de los PID, en primer lugar, se han de obtener las funciones de transferencia en función de la fuerza, no de la posición como se han obtenido. Esto es debido a que en el sistema real en los PID entra el error como una posición, sin embargo, lo que sale del PID es una fuerza. Para ello, simplemente se despeja el término  $2*k$  en las ecuaciones de los desplazamientos y el término  $4*k*R^2$  en la del giro quedando como en la figura 28. De esta manera quedan las funciones de transferencia de la siguiente manera.

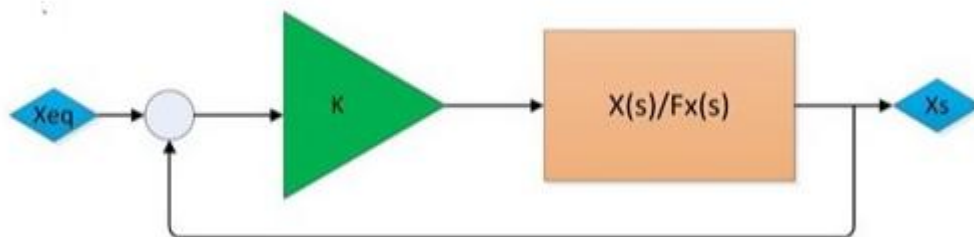


Figura 28: Esquema con función de transferencia en función de la fuerza. Fuente: (3).

$$\frac{X(s)}{F_x(s)} = \frac{1}{13.2s^2 + 61.11s} \quad [18]$$

$$\frac{Y(s)}{F_y(s)} = \frac{1}{13.2s^2 + 62.823s} \quad [19]$$

$$\frac{\theta(s)}{T_z(s)} = \frac{1}{0,21s^2 + 29,6s} \quad [20]$$

Una vez obtenidas las funciones, se pasó a la simulación del sistema con el PID, se empleó un PID discreto con el tiempo de muestreo real. En la siguiente figura se puede ver un esquema de la simulación realizada en Simulink, donde se puede ver de izquierda a derecha, la entrada en escalón, la realimentación al ser control en bucle cerrado, una ganancia de 1000 para asemejarse a la realidad, ya que en los PID reales les entraría el error en mm y saldría en N, el PID en z, la función de transferencia obtenida en s ( no había diferencia entre simular con la función de transferencia en s o en z, discretizada) y ya por último ,una ganancia para verlo en mm como el real, aunque esta ganancia no afecta realmente a la simulación, y un *scope* para visualizar resultados.

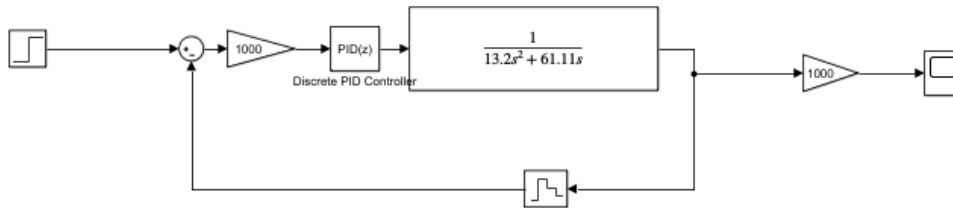


Figura 29: Esquema empleado para simular el sistema en Simulink. Fuente: Elaboración propia.

Se obtuvo un PID para cada función de transferencia que cumplía con las especificaciones requeridas. En las siguientes figuras se puede ver las respuestas al escalón con los PID diseñados.

Para el eje X: PID: P=0.104 I=0,0451 D=0,02

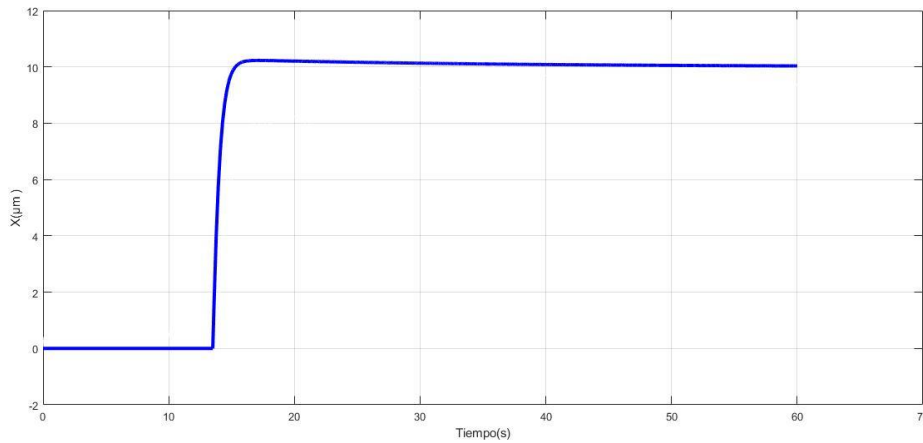


Figura 30: Respuesta teórica con el PID diseñado para X. Fuente: Elaboración propia.

Para el eje Y: PID: P=0,110 I=0,04 D= 0,02

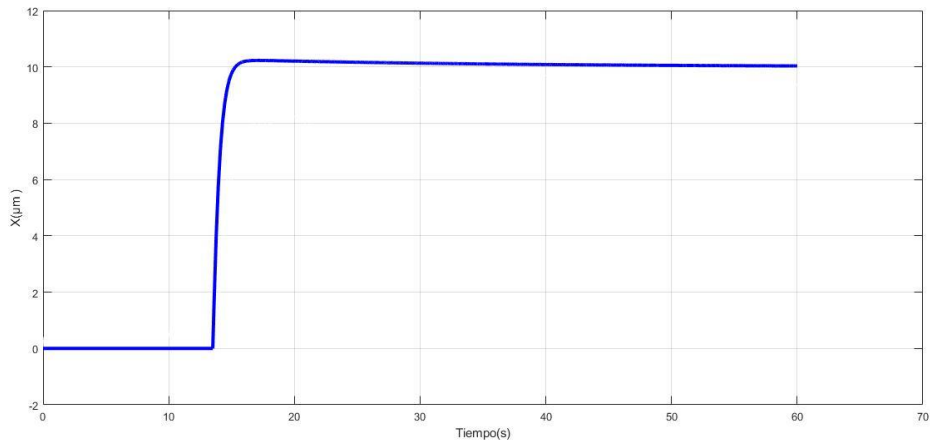


Figura 31: Respuesta teórica con el PID diseñado para Y. Fuente: Elaboración propia.

Ambas son muy similares debido a que ambas funciones de transferencia son casi idénticas

Para el giro: PID: P=100 I=62 D= 0,1

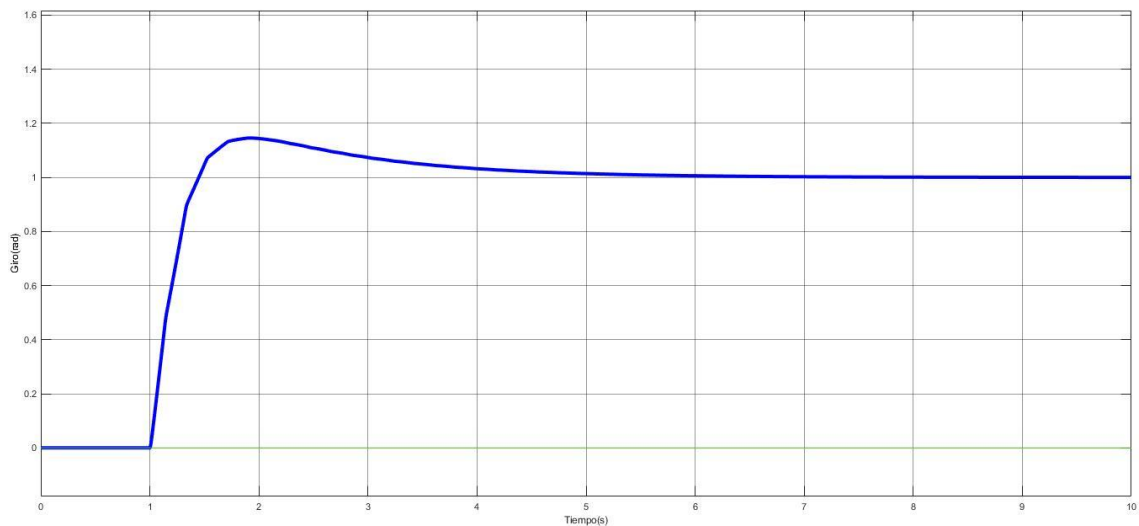


Figura 32: Respuesta teórica con el PID diseñado para el giro. Fuente: Elaboración propia.

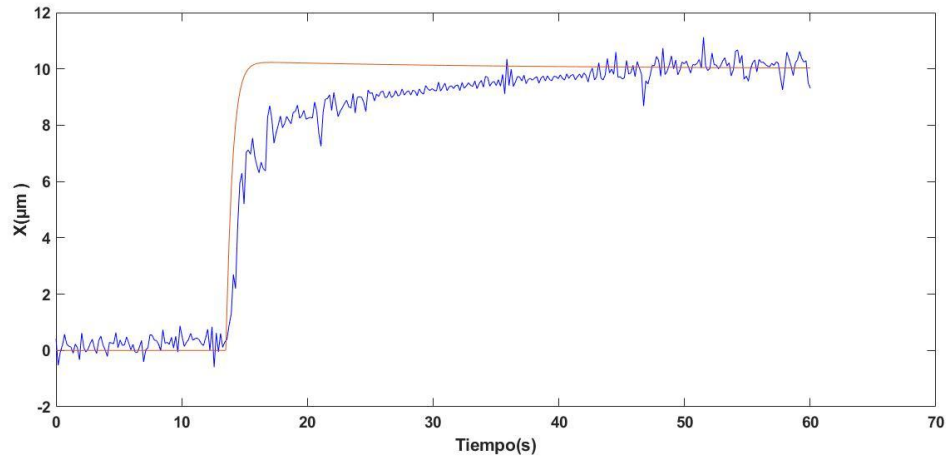
La respuesta de este sistema sería subamortiguada debido a que se busca mayor rapidez en el control del giro y de esta manera llega al valor final con mayor rapidez.

Tras haber diseñado unos PID, se pasó a la práctica y se trató de implementar estos PID en la plataforma, sin embargo al tener un efecto unos controladores sobre otros y al ser las funciones de transferencia simplemente unas aproximaciones, los PID quedaron finalmente.

	P	I	D
<b>X</b>	3.1	2.1	0.02
<b>Y</b>	3.5	2.5	0.02
<b>Giro</b>	120	60	0.01

Tabla 2: Valores finales de PID. Fuente: Elaboración propia.

Con estos PID la plataforma podía realizar escalones y movimientos a velocidad constante de manera adecuada. A continuación, se representa la respuesta ante un escalón y su comparación con la obtenida de manera teórica. Se ha representado un escalón para el eje X, para el eje Y la respuesta sería muy similar.



**Figura 33: Comparativa entre respuesta real y teórica. Fuente: Elaboración propia.**

## 4.4 Resultados experimentales

En este apartado se expondrán los resultados obtenidos empleando el control con PID de la plataforma. En primer lugar, se muestran los resultados de la respuesta de la plataforma ante distintas entradas a velocidad constante. A continuación, se va un poco más allá en el control y se analiza si es capaz de trabajar en todo el rango (50 mm x 50 mm), además de mostrar diferentes formas para mostrar el buen funcionamiento. Por último, se estudia el ruido que genera el control PID en el eje Z a través de un confocal.

### 4.4.1 Respuestas a movimientos a velocidad constante

Para probar la efectividad del control, se realizaron varios experimentos a velocidad constante, en primer lugar, se introdujo al sistema, tanto en el eje X como en el Y, una entrada de 10  $\mu\text{m}$  y otra de 100  $\mu\text{m}$ . Los resultados se pueden ver en las siguientes figuras.

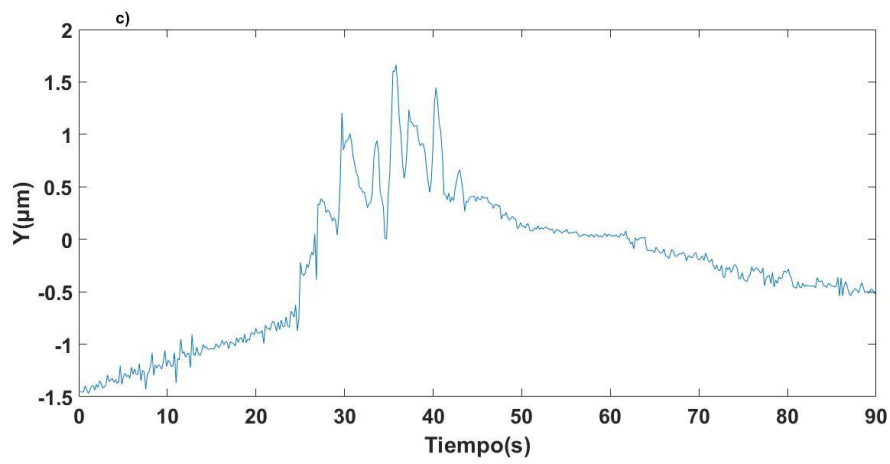
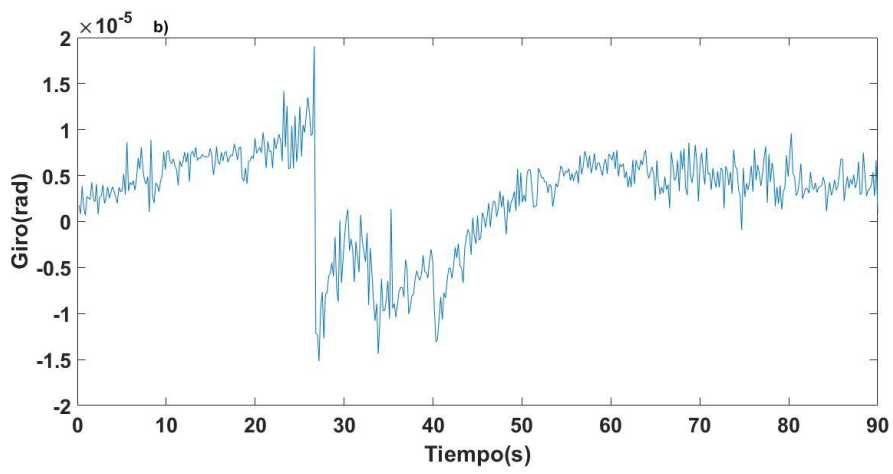
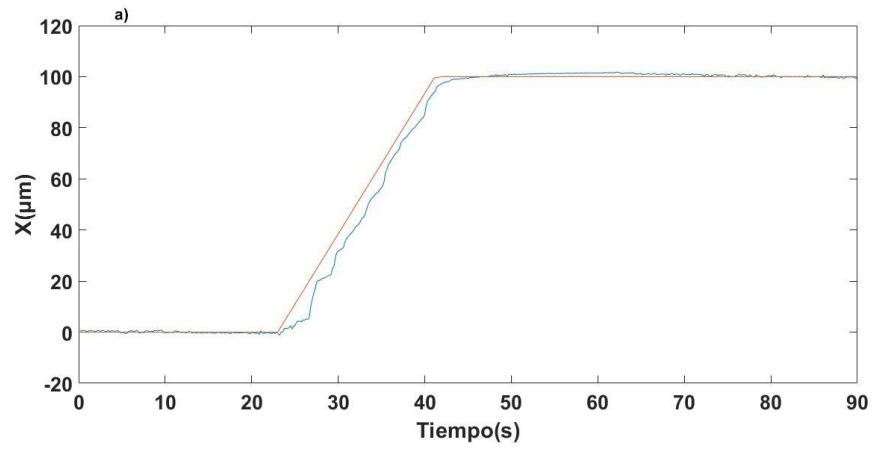


Figura 34: a) Respuesta movimiento de 100  $\mu\text{m}$  en X b) Giro c) Desplazamiento en Y. Fuente: Elaboración propia.

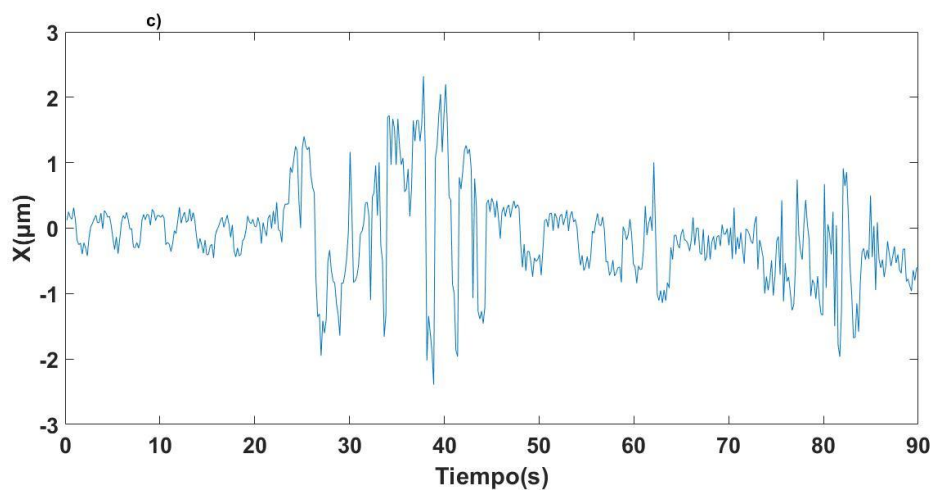
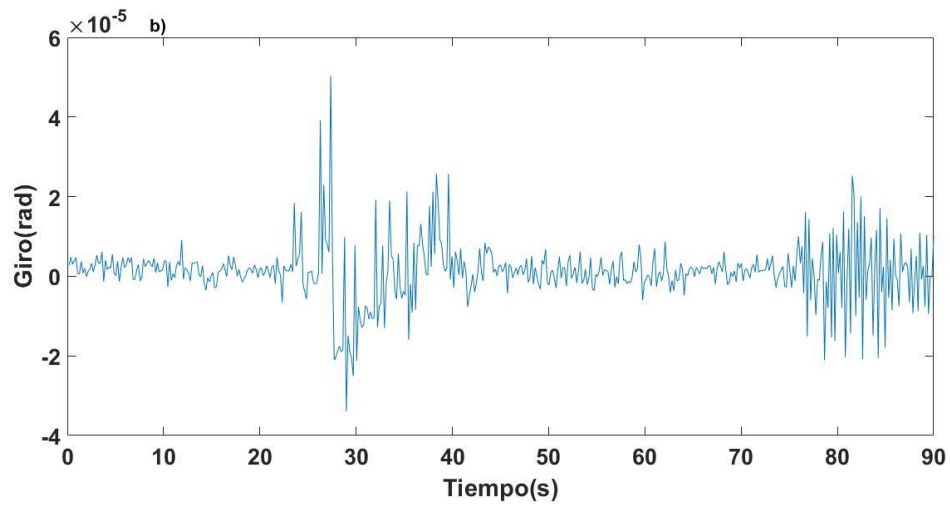
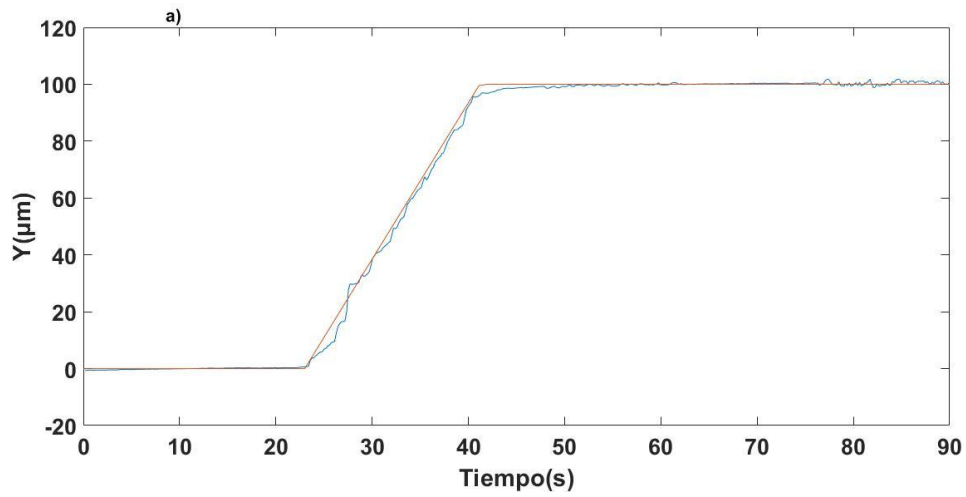


Figura 35: a) Respuesta movimiento de 100  $\mu\text{m}$  en X b) Giro c) Desplazamiento en Y. Fuente: Elaboración propia.



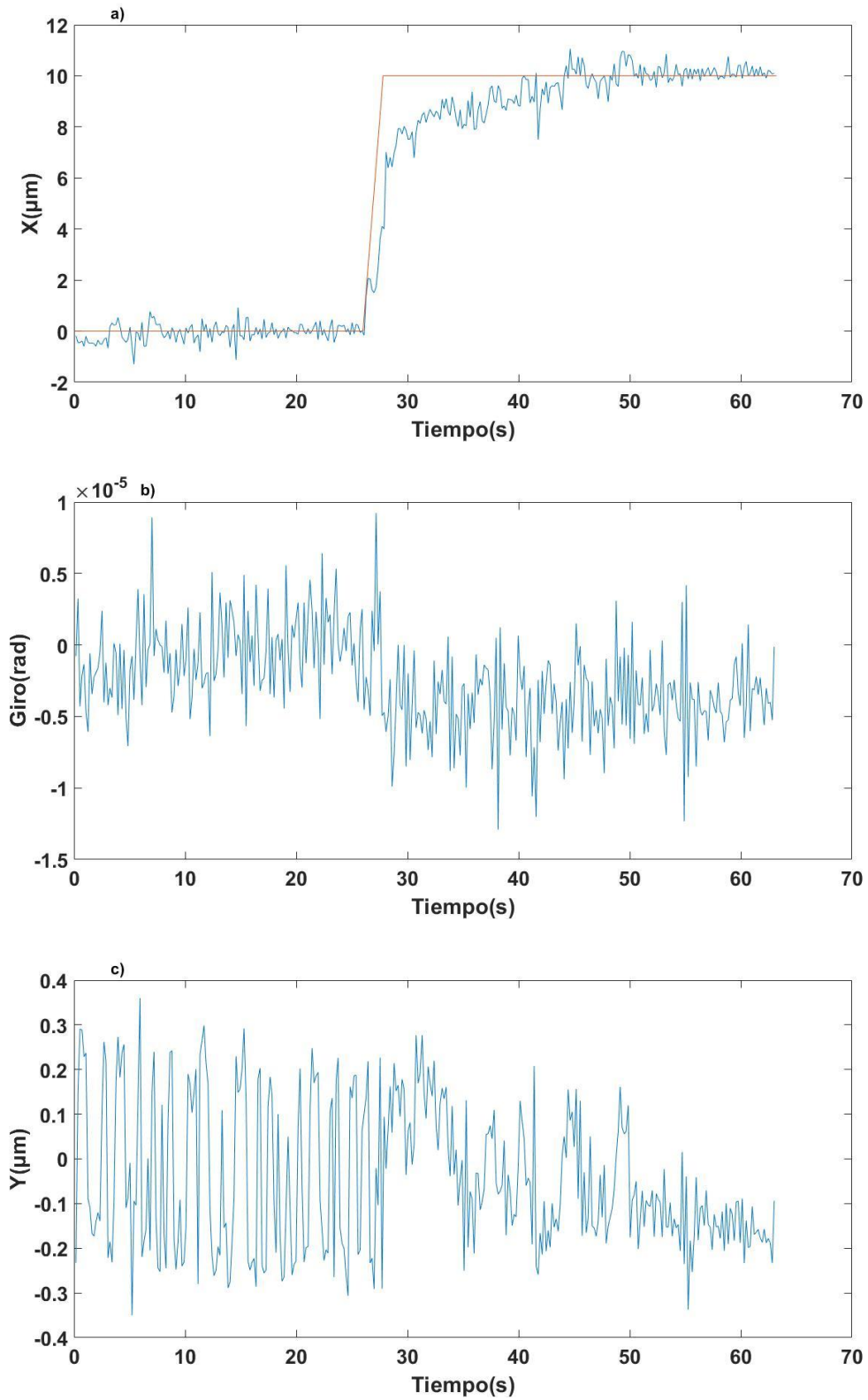
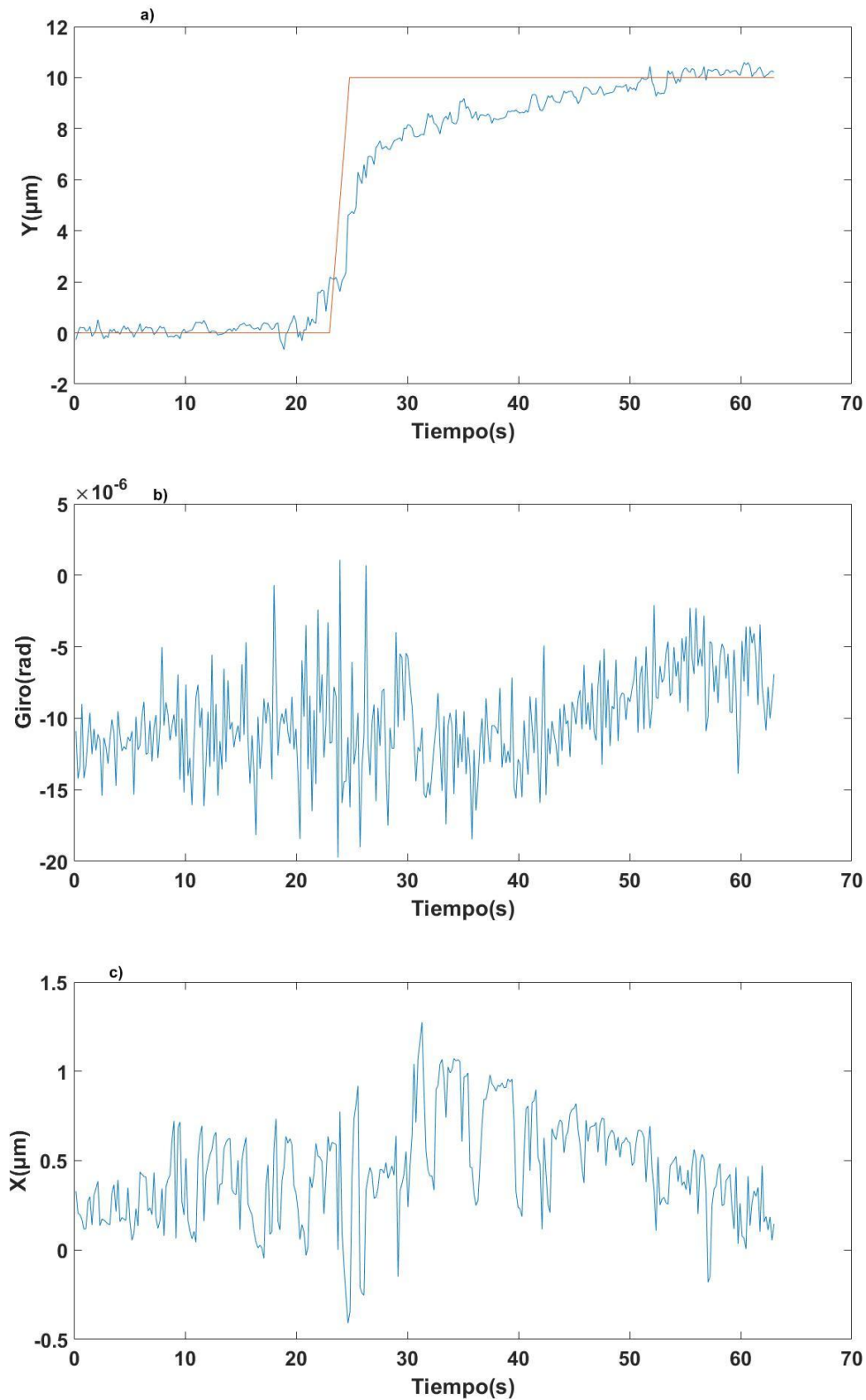


Figura 36: a) Respuesta movimiento de 10  $\mu\text{m}$  en X b) Giro c) Desplazamiento en Y. Fuente: Elaboración propia.



**Figura 37: a) Respuesta movimiento de 10  $\mu\text{m}$  en Y b) Giro c) Desplazamiento en X. Fuente: Elaboración propia.**

Como se puede ver, los objetivos planteados de reducir la sobreoscilación del sistema y de llegar al punto exacto requerido se cumplen. Además, se puede observar el desplazamiento en el eje contrario y el giro para respuesta, y se puede ver como apenas existe desplazamiento o giro, que es otro de los objetivos del control.

Posteriormente, se realizaron en el sistema movimientos consecutivos de 100  $\mu\text{m}$  para observar como respondía el sistema. En las siguientes figuras se puede observar el resultado.

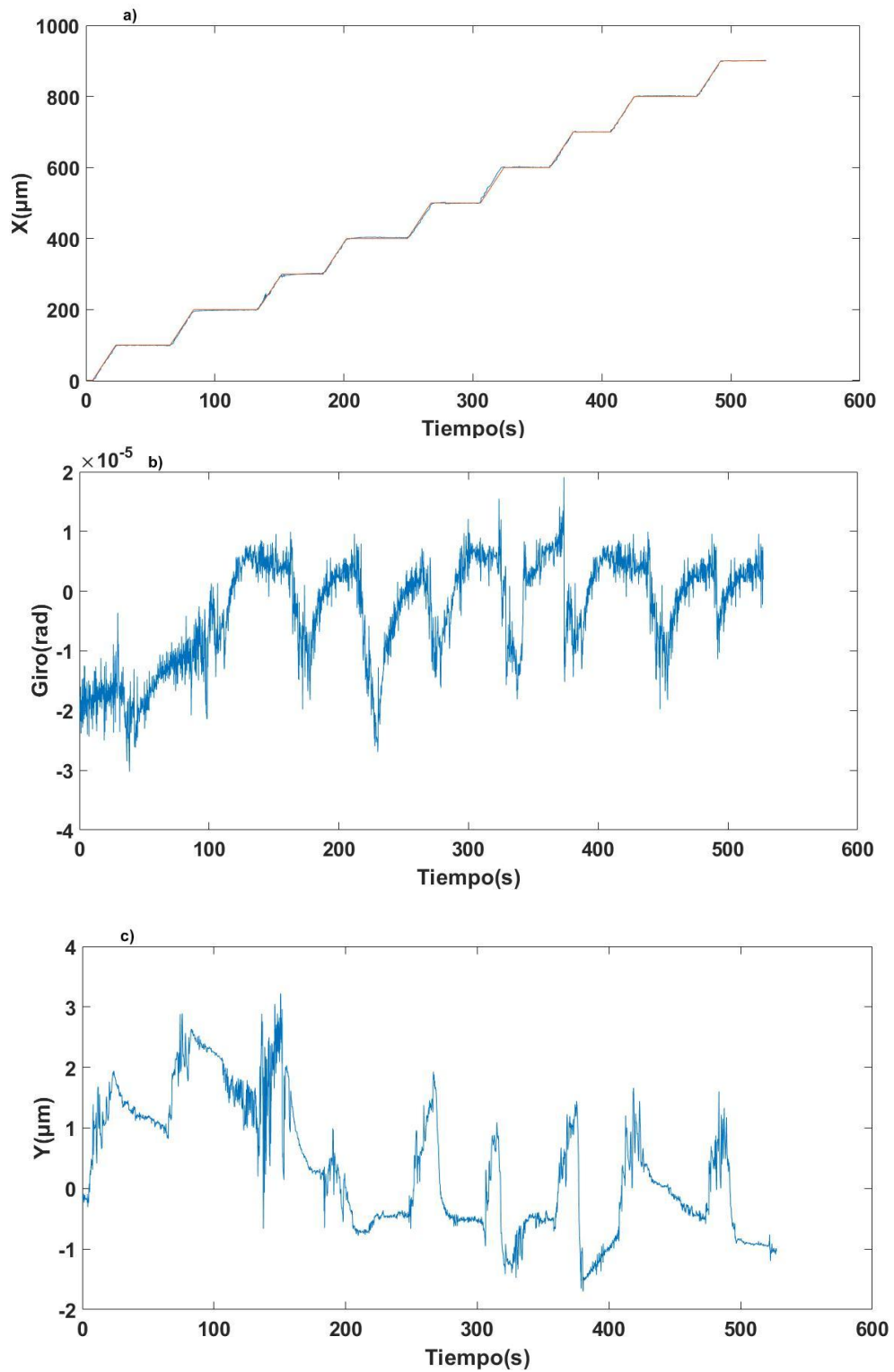


Figura 38:a) Desplazamientos consecutivos en el eje X de 100  $\mu\text{m}$  b) Giro c) Desplazamiento en Y. Fuente: Elaboración propia.

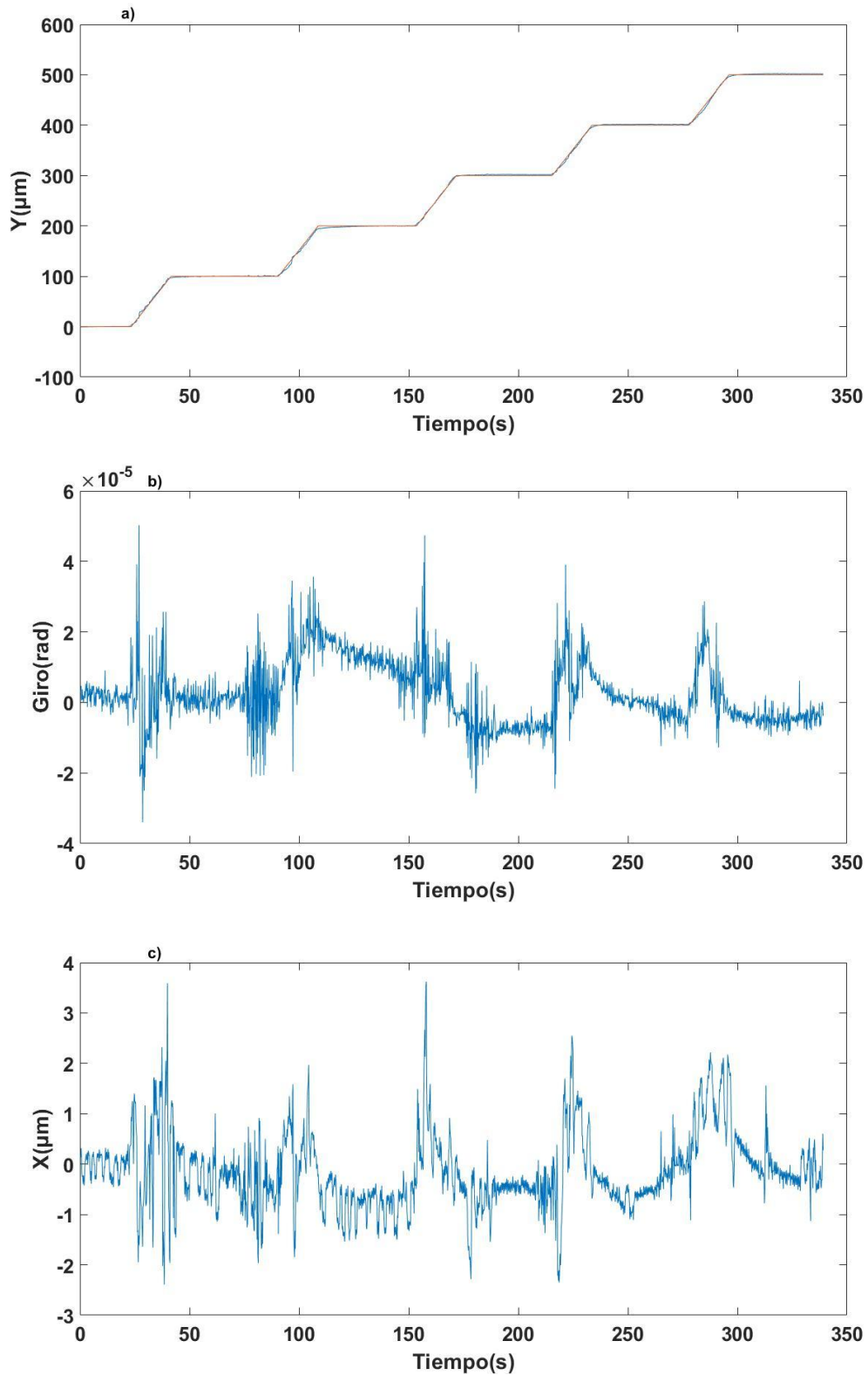


Figura 39:a) Desplazamientos consecutivos en el eje Y de 100 μm b) Giro c) Desplazamiento en X. Fuente: Elaboración propia.

También se realizaron movimientos más largos para comprobar que la plataforma funciona para todo el rango de trabajo (50x50mm). Se envió la plataforma con velocidad constante hasta el límite de la zona de trabajo que serían 25 mm ( al trabajar de 25 mm hasta -25mm) destacando que para todo el recorrido los láseres no bajaran su precisión, estando la luz siempre azul o verde, y las respuestas fueron las siguientes

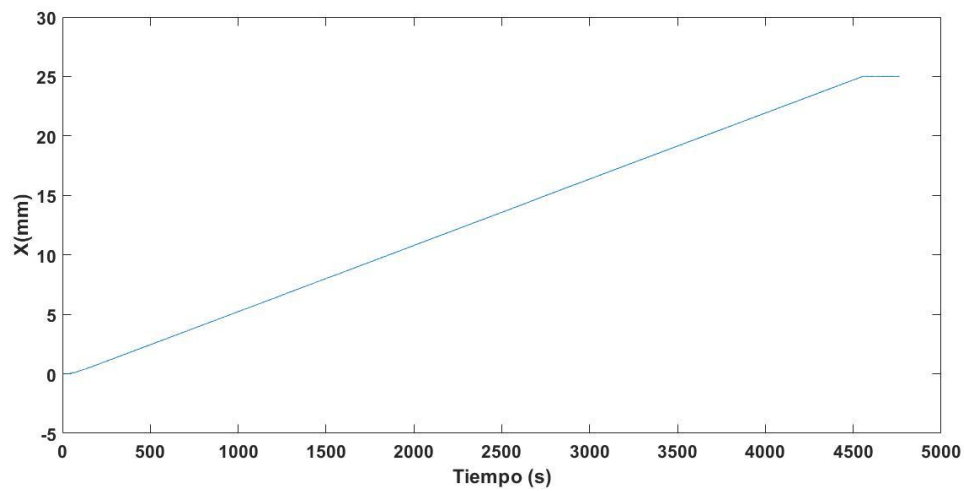


Figura 40: Movimiento largo de la plataforma en X, hasta 25mm. Fuente: Elaboración propia.

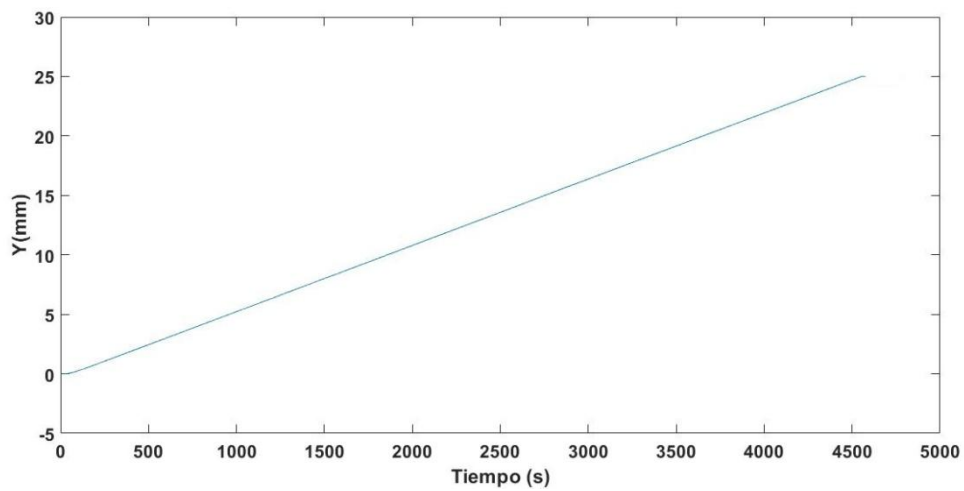


Figura 41: Movimiento largo de la plataforma en Y, hasta 25mm. Fuente: Elaboración propia.

De esta manera se comprueba que la plataforma alcanza los extremos de la zona de trabajo.

También se realizaron formas como un círculo o una espiral para demostrar el buen funcionamiento del control, como se puede ver en las siguientes figuras.

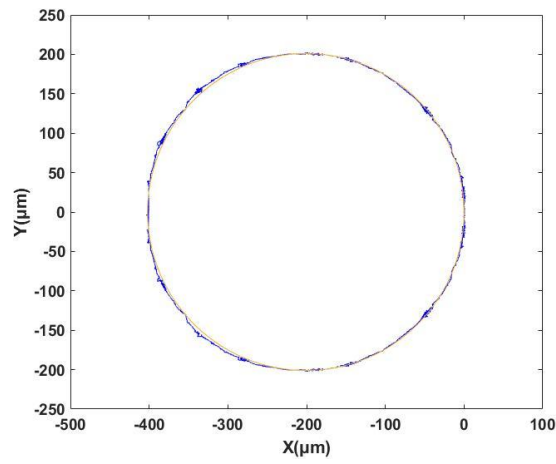


Figura 42: Movimiento a velocidad constante circular. Fuente: Elaboración propia.

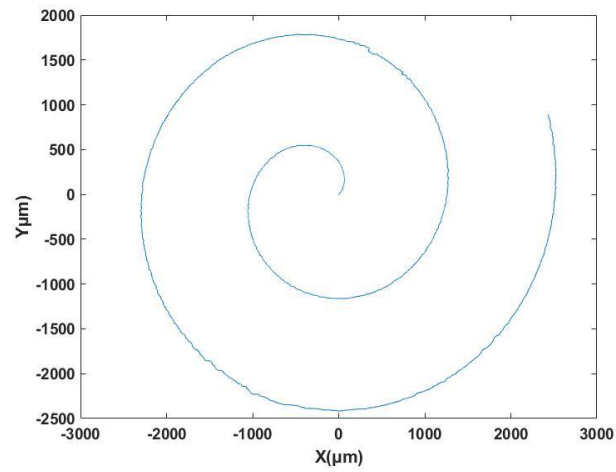


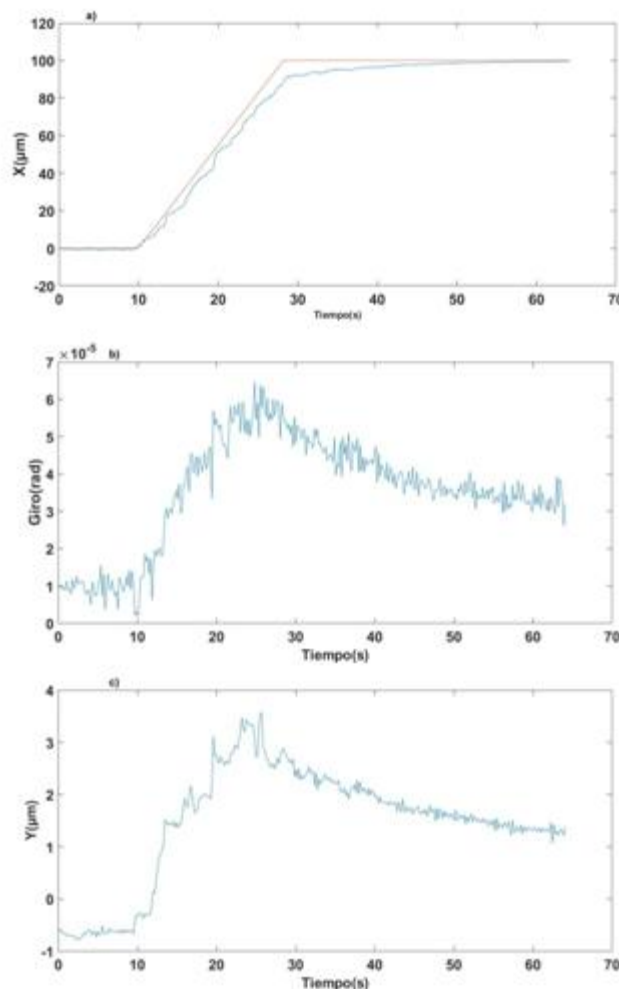
Figura 43: Movimiento a velocidad constante en forma espiral. Fuente: Elaboración propia.

## 4.4.2 Cambio de $F_z$ y de velocidad

En este apartado se realiza el estudio de funcionamiento del PID cuando se cambia la fuerza vertical del sistema, para  $F_z=1$  N y  $F_z=3$  N, para comprobar si el control funciona y cómo afecta al sistema.

Además, se variará la velocidad a la que se realizan los experimentos para tratar de obtener hasta que velocidades el sistema responde bien.

En primer lugar, se comprobó el funcionamiento del sistema con  $F_z=3$  N, el resultado fue satisfactorio ya que el PID funcionaba y la plataforma se desplazaba de manera similar a como lo hace con  $F_z=2$  N. Esto podía ser esperable si se comparan las respuestas ante un escalón en bucle abierto del sistema, ya que ambas son muy similares, como se vio en el apartado 4.2. En la figura 44 se puede ver una respuesta del sistema ante un movimiento a velocidad constante de  $100 \mu\text{m}$ .



**Figura 44:** a) Respuesta ante desplazamiento a velocidad constante de  $100 \mu\text{m}$  en X para  $F_z=3$  N b) Giro c) Desplazamiento Y. Fuente: Elaboración propia

Sí que es cierto que el giro y el desplazamiento en el otro eje aumentan un poco aunque no demasiado como para considerar que no funciona el PID. Sin embargo, para  $F_z=1$  N, el PID diseñado no funciona, esto podría deberse a la diferencia que existe entre ambas respuestas ( $F_z=$

1 N y  $F_z = 2$  N) al escalón en bucle abierto. Se podría volver a ajustar el PID para obtener un mejor resultado con esta fuerza.

También se probó el control a varias velocidades, la velocidad a la que realizaron el resto e pruebas y para la que fue diseñada el PID es de  $5.55 \mu\text{m/s}$ , y se observó que funcionaba de manera adecuada hasta una velocidad de  $16.66 \mu\text{m/s}$ .

Para una velocidad de  $8.33 \mu\text{m/s}$ :

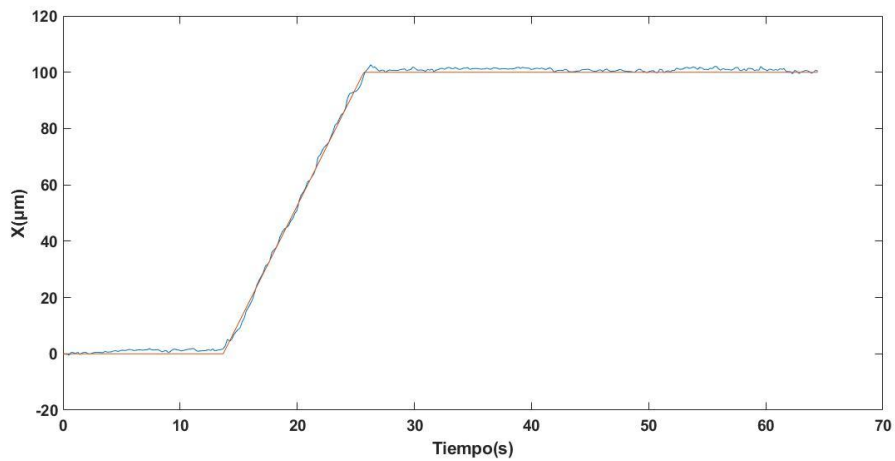


Figura 45: Respuesta del sistema a velocidad constante de  $8.33 \mu\text{m/s}$ . Fuente: Elaboración propia.

Para una velocidad de  $13.88 \mu\text{m/s}$ :

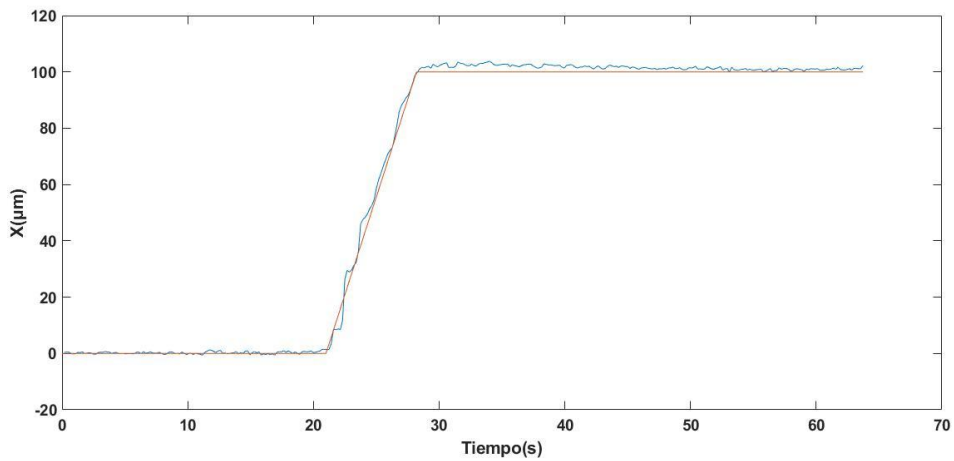
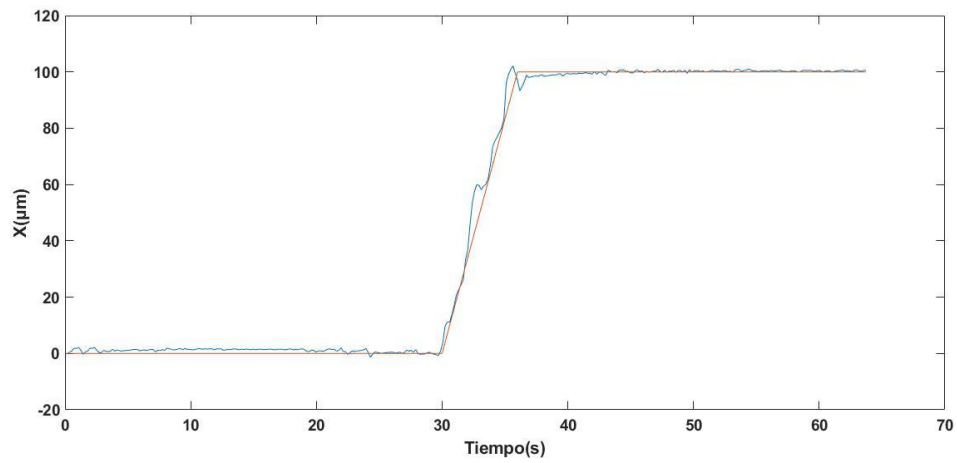


Figura 46: Respuesta del sistema a velocidad constante de  $13.88 \mu\text{m/s}$ . Fuente: Elaboración propia.



Para una velocidad de  $16.66 \mu\text{m/s}$ :



**Figura 47:** Respuesta del sistema a velocidad constante de  $16.66 \mu\text{m/s}$ . Fuente: Elaboración propia.

Se puede ver como ya se dan ciertos picos en la respuesta y a partir de esa velocidad el sistema comienza a comportarse peor.

### 4.4.3 Estudio del ruido en el eje Z a través de sensor confocal

En este último apartado, se estudió el ruido que generaba el controlador PID diseñado en el eje Z, para ello se hizo uso de un sensor confocal que permite medir la distancia en el eje Z. Este ruido, viene dado por los cojinetes de aire que hacen que la plataforma levite y por la fuerza vertical que ayuda a esta levitación

Para el análisis del ruido que generaba el PID, en primer lugar se estudió el ruido que se generaba sin controlador y con todo apagado (cojinetes de aire y motores), es decir el ruido del propio instrumento. Posteriormente, con el control del sistema en bucle abierto, para ver el ruido que generan estos cojinetes de aire y la fuerza vertical de los motores, y finalmente con el control en bucle cerrado quieto, para ver el ruido que genera el control en estático, y en movimiento, para ver el ruido del control al desplazarse. De esta manera, se pueden determinar qué efectos son los debidos al controlador diseñado.

Se realizaron 7 experimentos para cada caso con un mismo patrón cerámico de grado 0, y se empleó el sensor confocal a una frecuencia de 500 Hz durante 100 segundos para cada experimento, por lo que en cada experimento se obtuvieron 50000 datos. Para el estudio de estos datos, se procedió al cálculo de la desviación estándar y también de los valores máximos y mínimos. A continuación, se exponen los resultados para cada caso. Además, los experimentos se realizaron a una distancia del patrón de unos 2000  $\mu\text{m}$  (distancia óptima a la que trabaja el confocal). Sin embargo, para una mayor facilidad en la visualización de los resultados, la referencia se ha bajado escalando la gráfica a valores más pequeños. Además, el confocal trabajó con una intensidad de entre el 65% y 70%, de la máxima, la cual es suficiente para realizar medidas precisas según el fabricante y sin que llegue a saturar en ningún momento.

Para el primer caso, en el que las tarjetas no estaban conectadas y la plataforma estaba completamente estática, es decir los motores estaban apagados y los cojinetes de aire también, se calculó que los experimentos tenían una desviación estándar de entre 0,115  $\mu\text{m}$  y 0,162  $\mu\text{m}$  y una diferencia entre los valores máximo y mínimo de entre 0,854  $\mu\text{m}$  y 1,099  $\mu\text{m}$  (ver tabla 3). Se ha representado una gráfica con los datos de uno de los experimentos, además de representar el promedio y los límites superior e inferior calculados con la desviación estándar.

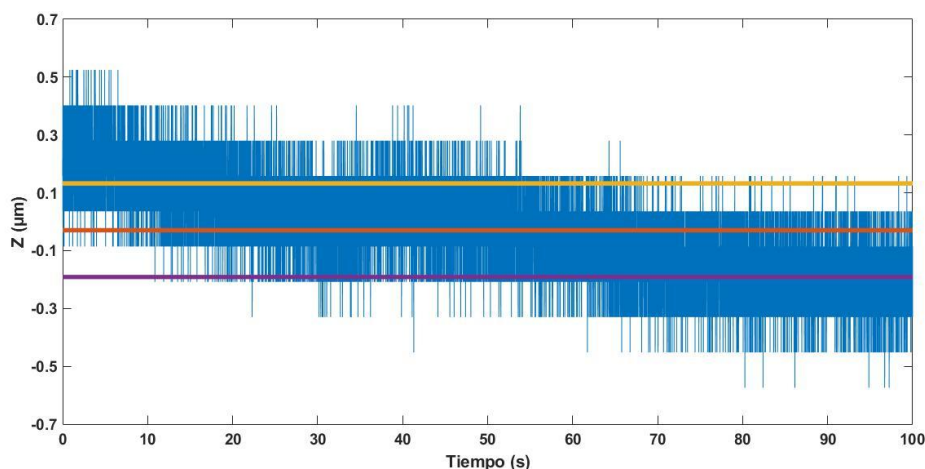


Figura 48: Ruido eje Z con todo apagado. Fuente: Elaboración propia.

Experimento	Desviación estándar $\sigma$ ( $\mu\text{m}$ )	Max-Min ( $\mu\text{m}$ )
1	0,109	0,854
2	0,115	0,854
3	0,162	1,099
4	0,135	0,89
5	0,146	0,933
6	0,122	0,866
7	0,161	1,034

Tabla 3: Experimentos con confocal con todo apagado. Fuente: Elaboración propia.

Para el segundo caso, con el control de la plataforma en bucle abierto, se volvió a realizar el mismo proceso, resultando una desviación estándar entre 0,141  $\mu\text{m}$  y 0,202  $\mu\text{m}$ , y una diferencia entre los valores máximos y mínimos de entre 0,977  $\mu\text{m}$  y 1,465  $\mu\text{m}$  (ver tabla 4). Se ha vuelto a representar un experimento. La inclinación que se ve tanto en esta gráfica como la anterior puede estar debida a una deriva térmica ya que ambos experimentos se realizaron consecutivamente.

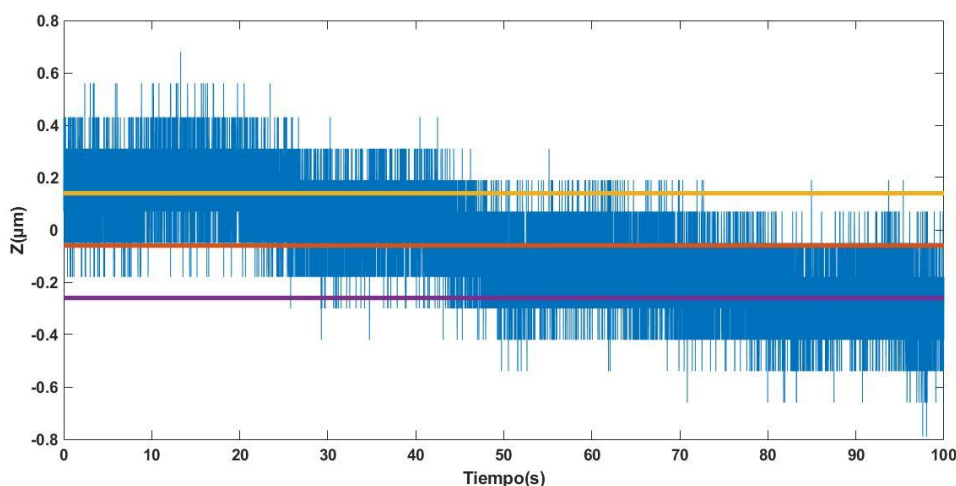


Figura 49: Ruido eje Z con sistema en bucle abierto. Fuente: Elaboración propia.

Experimento	Desviación estándar $\sigma$ ( $\mu\text{m}$ )	Max-Min ( $\mu\text{m}$ )
1	0,165	1,109
2	0,141	0,977
3	0,187	1,059
4	0,184	1,29
5	0,193	1,322
6	0,202	1,465
7	0,161	1,034

Tabla 4: Experimentos con confocal con sistema en bucle abierto. Fuente: Elaboración propia.

El tercer caso sería con el control con los PID diseñados en bucle cerrado, controlando la posición de la plataforma, pero con esta en la misma posición durante todos los experimentos. Para este caso las desviaciones estándar de todos los experimentos estuvieron comprendidas entre valores muy similares a los del control en bucle abierto, estando todos los valores entre 0,13  $\mu\text{m}$  y 0,2  $\mu\text{m}$ , y todos los valores de la diferencia entre el máximo y el mínimo valor de

cada experimento entre 1,1  $\mu\text{m}$  y 1,4  $\mu\text{m}$  (ver tabla 5). De esta manera se puede afirmar que estando la plataforma quieta el control no aporta ruido extra en el eje Z, lo cual es muy positivo.

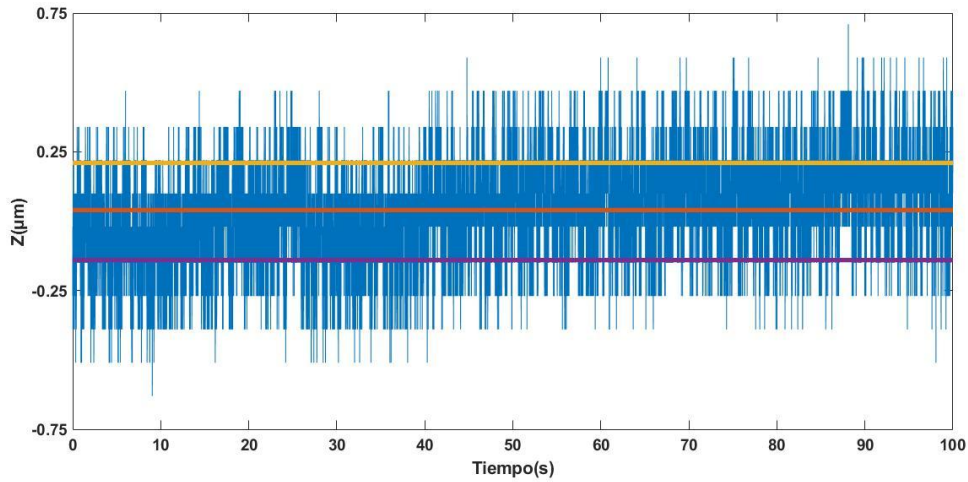


Figura 50: Ruido eje Z con sistema en bucle cerrado con control. Fuente: Elaboración propia.

Experimento	Desviación estándar $\sigma$ ( $\mu\text{m}$ )	Max-Min ( $\mu\text{m}$ )
1	0,161	1,099
2	0,171	1,343
3	0,137	1,099
4	0,189	1,343
5	0,201	1,478
6	0,13	1,100
7	0,156	1,221

Tabla 5: Experimentos con confocal con sistema en bucle cerrado con control. Fuente: Elaboración propia.

El último caso, sería el de control en bucle cerrado con el control PID, pero en este caso la plataforma no estaría estática sino que se realizarían 4 desplazamientos de 0,1  $\mu\text{m}$  durante cada experimento, yendo hasta 0,2  $\mu\text{m}$  y regresando. Los resultados para este caso sí que difirieron respecto a los anteriores, y los valores de la desviación estándar de cada experimento oscilaban entre 0,442  $\mu\text{m}$  y 0,476  $\mu\text{m}$ , y los de la diferencia del valor máximo y mínimo entre 4,123  $\mu\text{m}$  y 4,517  $\mu\text{m}$  (ver tabla 6). Para este caso todos los valores han aumentado, por lo que el ruido es un poco mayor cuando la plataforma se desplaza. Sin embargo, este aumento no es significativo ya que además al producirse desplazamiento plano, es posible que este aumento del ruido no sea solo debido al control, sino que también puede haberse dado debido a pequeños cambios que pueden existir en la superficie de la muestra empleada para los experimentos. A continuación, puede verse representada la gráfica para un experimento.

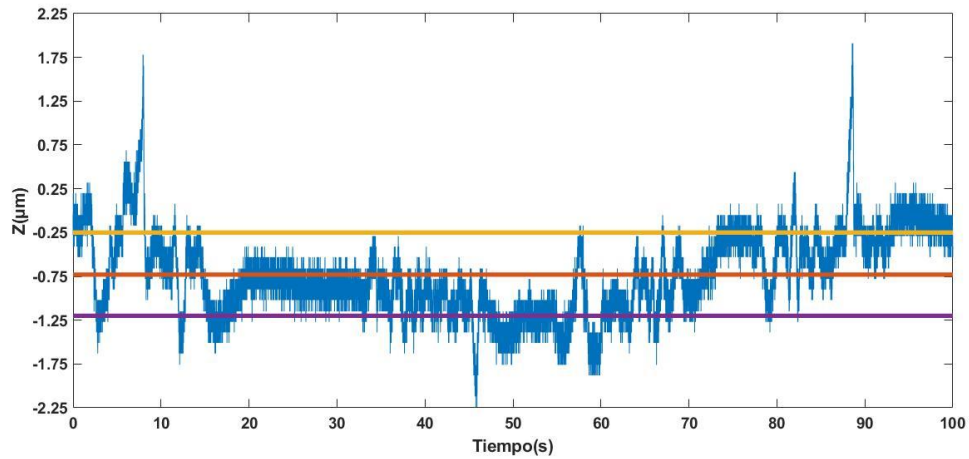


Figura 51: Ruido eje Z con sistema en bucle cerrado con control y en movimiento. Fuente: Elaboración propia.

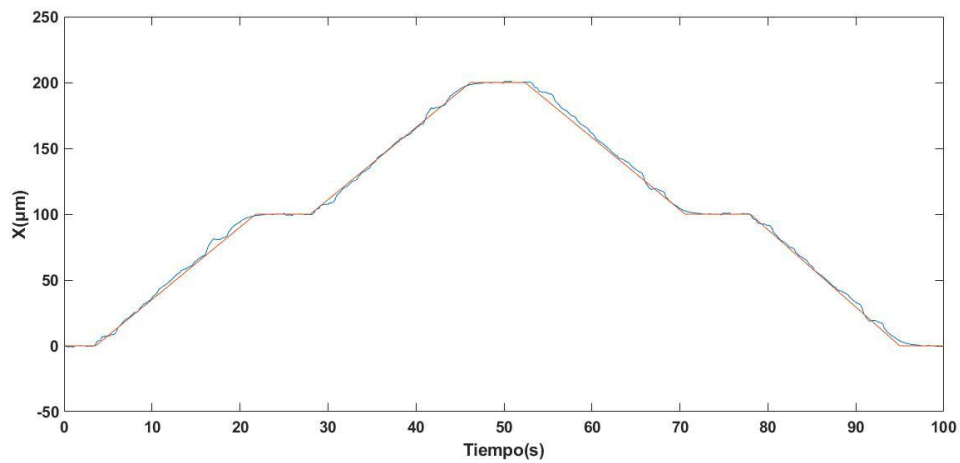
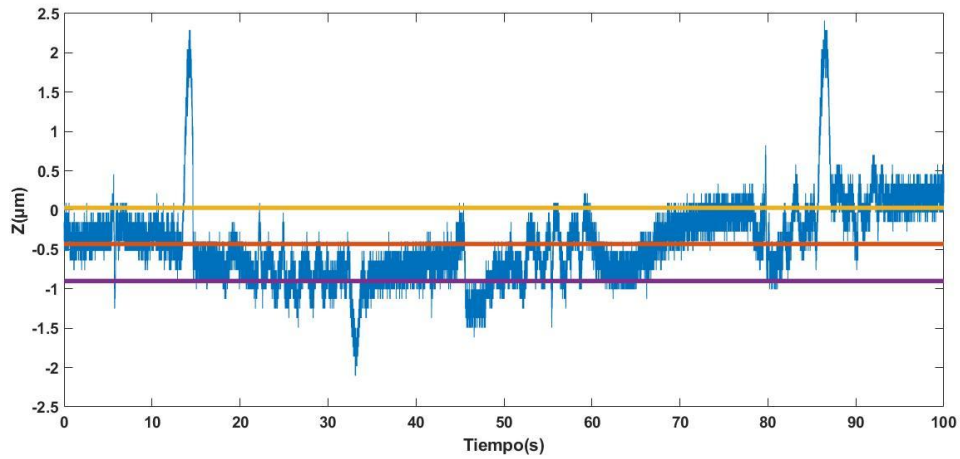


Figura 52: Trayectoria durante estudio con confocal. Fuente: Elaboración propia.

Experimento	Desviación estándar $\sigma$ ( $\mu\text{m}$ )	Max-Min ( $\mu\text{m}$ )
1	0,463	4,517
2	0,476	4,150
3	0,442	4,123
4	0,448	4,245
5	0,457	4,489
6	0,467	4,356
7	0,471	4,502

Tabla 6: Experimentos con confocal con sistema en bucle cerrado con control y en movimiento. Fuente: Elaboración propia.

En la gráfica podemos ver cierta simetría en los picos más altos de la gráfica, al ir y al volver con la plataforma, por lo que la propia probeta empleada puede ser responsable de este aumento en el ruido en el eje Z. Otro ejemplo se puede ver en la figura 53, que se realizó con el mismo recorrido.



**Figura 53: Ruido eje Z con sistema en bucle cerrado con control y en movimiento. Fuente: Elaboración propia.**

La conclusión del estudio del ruido indica por tanto, que el controlador diseñado no aporta ruido extra en el eje Z cuando la plataforma se mantiene estática y el ruido que se aporta cuando se desplaza no es suficientemente alto como para considerarlo significativo, ya que además este puede ser producido por la superficie del propio patrón empleado en los experimentos al no ser totalmente lisa.

# 5. Interfaz para tratamiento de datos del control PID

## 5.1 Objetivo y herramienta empleada

Se desarrolló una interfaz gráfica para el programa de Simulink del control en bucle cerrado con PID, para obtener así mayor simplicidad a la hora de ejecutar el programa y una mejora en la visualización de los resultados obtenidos en los experimentos realizados con el control PID. De esta manera, la interfaz debía ser capaz de extraer gráficas y visualizar distintos datos en tiempo real además de poder controlar la plataforma.

Esta labor se llevó a cabo empleando la herramienta de Matlab GUIDE, que permite la creación de una GUI (Graphical User Interface), (Interfaz Gráfica de Usuario). Esta herramienta genera dos archivos un \*.fig, el cual es la propia interfaz, lo que ve el usuario. Ese archivo, mediante el editor que proporciona la herramienta GUIDE, y que se puede ver en la figura 54, se edita de tal manera que los distintos cuadros de texto, botones, pulsadores, barras... necesarios, quedan como el usuario desee. Tras esta parte más visual en la que el usuario elige como ha de quedar la GUI, la herramienta GUIDE genera el programa correspondiente a la interfaz en el otro archivo, que sería el \*.m.

En ese archivo, además, será donde el propio usuario programe la función de cada elemento que se ha agregado a la interfaz, a través de funciones que genera automáticamente la herramienta para cada elemento, y que se ejecutarán posteriormente cuando el usuario interactúe con ellos.

Para la realización de la interfaz se hizo uso de los siguientes trabajos (9) y (10)

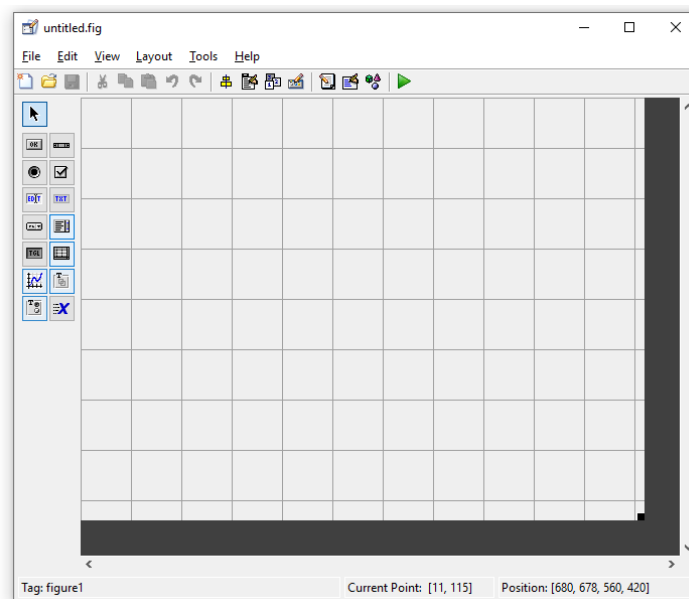


Figura 54: Editor de la herramienta GUIDE. Fuente: MatLab.

## 5.2 Descripción del programa

Existen varias maneras de programar una GUI, en este caso y al ser necesaria la comunicación entre la interfaz y el programa de Simulink, se ha realizado una programación totalmente orientada a ese programa, es decir que la interfaz diseñada solo funcionaría para ese programa y para los distintos bloques que contiene. Si se quisiera extender la misma interfaz a otro programa parecido, habría que cambiar ligeramente el código.

La interfaz en cuestión debía contener lo necesario para poner en marcha y parar el programa de Simulink, activar y mandar a la posición inicial la propia plataforma, poder cambiar la posición de la plataforma y su velocidad. También debía ser capaz de leer los datos de las posiciones de cada motor y la media entre la posición de estos, y mostrarlos en tiempo real, además de poder graficar estos datos.

Con toda esta información de partida, se procedió al diseño de la GUI, la cual se puede ver en la siguiente imagen.

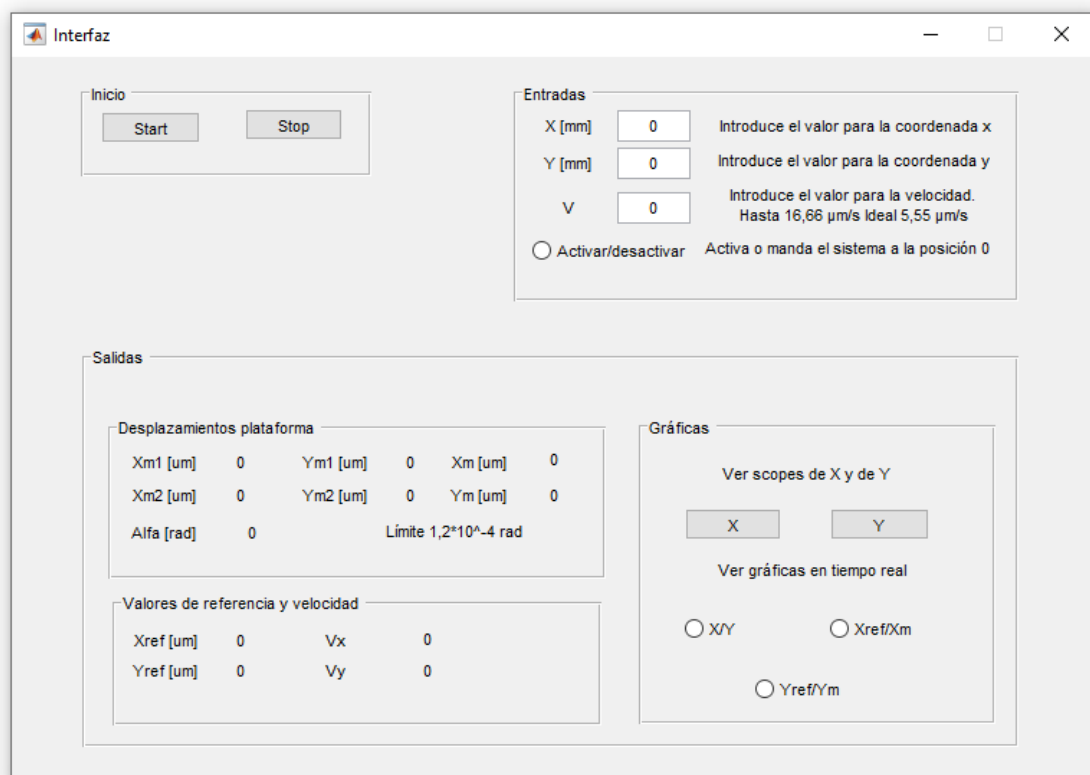


Figura 55: Interfaz gráfica diseñada. Fuente: Elaboración propia.

La interfaz se divide en tres partes, una orientada al inicio del programa, otra a las entradas del sistema y otra a las salidas o datos que se extraen del sistema. En los siguientes apartados, se hablará en detalle de cada una de estas partes, explicando su función y su programación.



## 5.2.1 Inicio

La parte de inicio comprende dos pulsadores, uno para lanzar y otro para parar el programa. En la figura 56 se ha resaltado la parte del inicio.

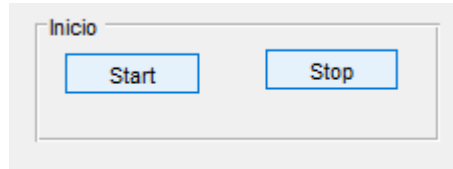


Figura 56: Parte de inicio de la interfaz. Fuente: Elaboración propia.

La programación de esta parte resulta sencilla, son dos pulsadores y cuando el usuario los acciona, la función mostrada abajo se ejecuta para poner en marcha el sistema o para pararlo. Mediante el comando *set\_param* se ajusta el parámetro de la simulación para que la ponga en marcha (start) o la pare (stop) según el botón pulsado.

```
function Start_Callback( ~, ~, ~)

set_param
('MandarDatos_Laser_0_01_ControlFuerzas_OrdenadoGuille',
'SimulationCommand', 'start');

function Stop_Callback( ~, ~, ~)

set_param
('MandarDatos_Laser_0_01_ControlFuerzas_OrdenadoGuille',
'SimulationCommand', 'stop');
```

## 5.2.2 Entradas

Otra de las partes es la correspondiente a las entradas del sistema, donde se introducen los datos para la posición de la plataforma (en X e Y), de la velocidad y también un botón extra para activar y mandar el sistema a la posición inicial. En la siguiente imagen se ve más en detalle esta parte, que además, cuenta con descripciones para cada una de las entradas de las que dispone.

Entradas		
X [mm]	<input type="text" value="0"/>	Introduce el valor para la coordenada x
Y [mm]	<input type="text" value="0"/>	Introduce el valor para la coordenada y
V	<input type="text" value="0"/>	Introduce el valor para la velocidad. Hasta 16,66 $\mu\text{m/s}$ Ideal 5,55 $\mu\text{m/s}$
<input type="radio"/>	Activar/desactivar	Activa o manda el sistema a la posición 0

Figura 57: Parte de las entradas de la interfaz. Fuente: Elaboración propia.

El código para esta parte sería el siguiente. En cuanto a cambiar la posición de la velocidad o de la posición del sistema, serían tres funciones pero con la misma programación, solo cambiaría el bloque al cual modifica la función. Su programación sería la siguiente.

```
function editx_Callback(hObject, ~, ~)

x=0
x=get(hObject, 'String')
set_param('MandarDatos_Laser_0_01_ControlFuerzas_OrdenadoGuille/X', 'value', num2str(x))
```

En la función se define una variable local y se le asigna el valor introducido por el usuario como un *string* cuando este presiona *enter*, para posteriormente, asignar ese valor con *set\_param* al bloque correspondiente y cambiándolo de *string* a número con *num2str*.

En cuanto al botón que activa o devuelve a la posición inicial a la plataforma, su código sería el siguiente.

```
function radiobutton2_Callback(hObject, ~, ~)

b=0;
b=get(hObject, 'Value');

if b==0

set_param('MandarDatos_Laser_0_01_ControlFuerzas_OrdenadoGuille/
Reiniciar1', 'value', num2str(b));

set_param('MandarDatos_Laser_0_01_ControlFuerzas_OrdenadoGuille/
Reiniciar2', 'value', num2str(b));

else

set_param('MandarDatos_Laser_0_01_ControlFuerzas_OrdenadoGuille/
Reiniciar1', 'value', num2str(b));

set_param('MandarDatos_Laser_0_01_ControlFuerzas_OrdenadoGuille/
Reiniciar2', 'value', num2str(b));
end
```

Lo que se hace es definir una variable local, b, para asignarle un valor de 0 o 1 dependiendo de si el botón está sin pulsar o pulsado respectivamente. Posteriormente, según el valor asignado, es decir si el botón está pulsado o no, se ejecuta un código u otro, que simplemente cambia el bloque en Simulink para iniciar o reiniciar la posición.

En este botón, además, es donde se programa el bucle que muestra en tiempo real los datos de la plataforma pero se explica más adelante junto a las salidas que proporciona la interfaz.

## 5.2.3 Salidas

La última parte de la interfaz es la dedicada a las salidas del sistema, en ella por un lado se pueden ver en tiempo real los datos de la posición de cada uno de los motores, su media, el giro de la plataforma, la referencia del sistema, que sería el valor introducido por el usuario, y por último la velocidad en cada eje, para así poder ver con más facilidad como se desplaza la plataforma y la posición de cada motor. Además, la interfaz permite ver gráficas en tiempo real para observar cómo responde el sistema ante determinadas entradas y entrar directamente a los *scopes* de Simulink. La parte de la interfaz de las salidas sería la mostrada en la figura 58.

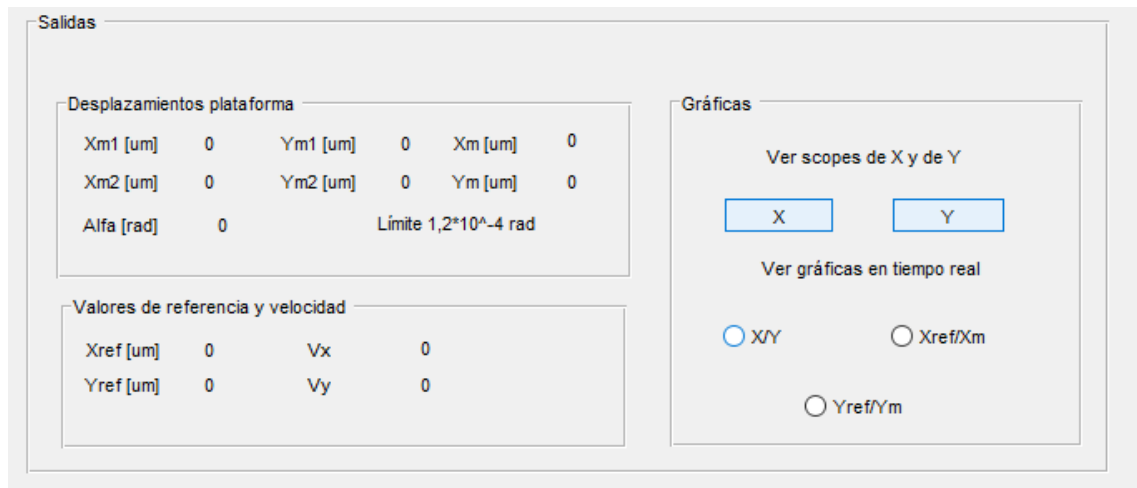


Figura 58: Parte de las salidas de la interfaz. Fuente: Elaboración propia.

Para poder mostrar los datos en tiempo real se crea un bucle que se repite cada 0,19 segundos, valor que no ralentiza el programa, y se muestran los datos cada vez que transcurre ese intervalo. La programación estaría contenida en la función antes mostrada para el botón de activar o desactivar la plataforma, y mostraría los datos de los desplazamientos plataforma y valores de referencia y velocidad. Abajo se muestra el programa.

```
while get(hObject, 'Value')

    xml=get_param('MandarDatos_Laser_0_01_ControlFuerzas_OrdenadoGuille/G1', 'RuntimeObject');
    xmlb=num2str(xml.OutputPort(1).Data);
    set(handles.valorxml, 'string', xmlb);

    pause(0.19);

end
```

El programa consiste en, tras haberse inicializado el bucle, coger de Simulink el dato deseado y almacenarlo en una variable a través de *get\_param*. Posteriormente se lee ese valor en la interfaz sacando el dato a través del puerto de salida que tiene Simulink disponible (*OutputPort(1)*).

Solo se ha escrito el código, en este apartado, para un valor ya que para el resto sería la misma programación, simplemente cambiando las variables y los bloques de los que se obtienen esos datos. Ver Anexo 1 para programación completa.

En cuanto a los gráficos para acceder directamente a los *scopes*, se han añadido dos pulsadores para acceder a los desplazamientos de X y de Y. La programación sería la siguiente.

```
function X_Callback(~, ~, ~)

set_param
('MandarDatos_Laser_0_01_ControlFuerzas_OrdenadoGuille/Xref, Xm1
, Xm2, Xmedia', 'open', 'on');

function Y_Callback(~, ~, ~)
set_param
('MandarDatos_Laser_0_01_ControlFuerzas_OrdenadoGuille/Yref, Ym4,
Ym3, Ymedia', 'open', 'on');
```

Por último, la interfaz también permite graficar en tiempo real el desplazamiento en X y de Y medio de ambos motores respecto al tiempo y uno respecto del otro. Esto permite ver en tiempo real las respuestas más importantes y con las que se evalúa si el control es óptimo, aparte de poder representar las figuras que se verán más adelante.

Para su programación se crearon 3 vectores para contener los datos de la posición y el tiempo, y que cuando se presiona el botón, se grafican los datos. A continuación, se muestra la programación de uno de los botones, el que gráfica el desplazamiento en X respecto al de Y, el resto serían iguales solo cambiando el vector que emplearían.

```
function XY_Callback(hObject, eventdata, handles)

global a;
a=0;
a=get(hObject, 'Value')
```

El funcionamiento es el siguiente, en cada botón se crea una variable global que vale 1 cuando se pulsa el botón y 0 cuando se pulsa de nuevo. De esta manera en el bucle principal se crea una condición (un *if*) que solo se cumple cuando este botón esta pulsado, es decir la variable global es 1. Cuando esta condición se cumple se representa la gráfica en tiempo real. Esta sería la condición creada en el bucle grande, antes explicado. Ver Anexo 1 para ver programa completo.

```
if a==1
figure(1)
labelx('Xm(um) ')
labely('Ym(um) ')
plot(X, Y)
drawnow
end
```

Esto se repetiría 3 veces para los 3 gráficos en tiempo real que se pueden ver.

## 6. Conclusiones

El objetivo previamente marcado para este trabajo de implementar un control en 2D para la NanoPla se ha cumplido con éxito, sin embargo para una conclusión global hay que comprobar que se han cumplido cada una de las tareas necesarias que se han realizado para la consecución del objetivo.

En primer lugar, se ha realizado una explicación de cómo funciona la plataforma y de cada una de sus partes y de las que entran en juego en el control de esta. Posteriormente, se ha realizado un ajuste del sistema de medida láser, el cual consta de 3 interferómetros láser, y con el cual se ha conseguido colocar perpendicularmente los haces láser y los espejos, y definir el giro inicial de la plataforma como nulo, logrando así una precisión de la lectura de los láseres alta.

Una vez ajustados los láseres, se ha procedido a explicar la estrategia de control que se iba a emplear y que condiciona el resto del trabajo. Para conseguir unos PID adecuados, se comenzó obteniendo un modelo que representara de manera precisa los tres movimientos de la plataforma en 2D, los desplazamientos en cada eje y su giro. Para ello, se hizo uso de una herramienta de MatLab y de los conocimientos de funciones de transferencia calculando estas funciones directamente. Gracias a estas herramientas se determinaron tres modelos que representaban de manera fiel el movimiento de la plataforma, además de demostrar el sentido físico de las funciones de transferencia de los desplazamientos al comprobar cómo se modificaba la respuesta del sistema tanto en la realidad como en la simulación cuando se cambiaba la fuerza vertical del sistema.

Una vez obtenidas las funciones de transferencia de cada movimiento, se procedió a la simulación de unos PID para estas de manera individual. Tras obtenerlos, se probaron sobre el sistema real, y tras un ajuste fino se consiguió que funcionase el control. El motivo del que los controladores no fueran exactos es, por un lado que las funciones de transferencia obtenidas son solo una aproximación, y que los controladores tienen efecto unos sobre otros (se interfieren mutuamente) generando diferencias entre la simulación y la realidad.

Tras obtener los PID, se procedió a su validación mediante una serie de movimientos a velocidad constante en ambos ejes para demostrar que el control funcionaba: 10  $\mu\text{m}$ , 100  $\mu\text{m}$ , 100  $\mu\text{m}$  de manera consecutiva, 25 mm para comprobar el funcionamiento en la zona de trabajo, un círculo y una espiral. Además, se realizó un estudio del ruido en el eje Z para comprobar que los tres controladores PID diseñados no generaban mucho ruido. Para ello en primer lugar se estudió el ruido que generaba ya de por sí la plataforma sin el control para compararlo posteriormente. El resultado fue satisfactorio ya que el control no generaba demasiado ruido en el eje Z.

Por último, la interfaz gráfica necesaria para facilitar la obtención de datos de la plataforma se diseñó de manera exitosa, consiguiendo que esta pudiera mover y obtener los datos requeridos del movimiento de la plataforma, además de poder graficar estos datos en tiempo real sin ralentizar el programa.

## 7. Bibliografía

1. *One-Dimensional Control System for a Linear Motor of a Two-Dimensional Nanopositioning Stage Using Commercial Control Hardware*. **Lucía Candela Díaz Pérez, Marta Torralba Gracia, José Antonio Albajez García, José Antonio Yagüe Fabra**. 2018, Micromachines.
2. *2D positioning control strategy for a nanopositioning stage* . **L.C. Díaz-Pérez, M. Torralba, J.A. Albajez, J.A. Yagüe-Fabra**. Bilbao : Euspen, 2019.
3. *Vector Control Strategy for Halbach Linear Motor Implemented in a Commercial Control Hardware*. **Lucía C. Díaz-Pérez, José A. Albajez , Marta Torralba and José A. Yagüe-Fabra**. s.l. : Electronics, 2018.
4. Kjmagnetics. [En línea] K&J Magnetics, Inc. [Citado el: 20 de Agosto de 2019.] <https://www.kjmagnetics.com/blog.asp?p=halbach-arrays>.
5. Renishaw. [En línea] Renishaw. [Citado el: 22 de Agosto de 2019.] <https://www.renishaw.es/es/explicacion-de-la-interferometria--7854>.
6. **Dobson, Roberto Cárdenas**. U-cursos. [En línea] [Citado el: 25 de Agosto de 2019.] [https://www.u-cursos.cl/ingenieria/2014/2/EL4004/1/material\\_docente/bajar?id\\_material=962199](https://www.u-cursos.cl/ingenieria/2014/2/EL4004/1/material_docente/bajar?id_material=962199).
7. **Keesman, Karel J**. *System Identification*. Londres : Springer, 2011. 978-0-85729-521-7.
8. **Ogata, Katsuhiko**. *Ingeniería de control moderna 5ed*. Madrid : PEARSON EDUCACIÓN, S.A, 2010. 978-84-8322-660-5.
9. **Morales, Santiago Galán**. *Interfaz Gráfica Para La Simulación De Modelos*. Leganés : s.n., 2011.
10. **Guerrero, Diego Orlando Barragán**. MANUAL GUIDE EN MATLAB. [En línea] [Citado el: 10 de septiembre de 2019.] [matpic.com](http://matpic.com).

# Anexo 1: Programa interfaz completo

% Esta parte del programa la crea directamente la herramienta empleada, no hace falta modificarla (tres primeras funciones), sirven para inicializar la interfaz

```
function varargout = Interfaz(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @Interfaz_OpeningFcn, ...
                  'gui_OutputFcn',  @Interfaz_OutputFcn, ...
                  'gui_LayoutFcn',  [], ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State,
varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

function Interfaz_OpeningFcn(hObject, eventdata, handles,
varargin)

guidata(hObject, handles);

function varargout = Interfaz_OutputFcn(hObject, eventdata,
handles)
varargout{1} = handles.output;

% A partir de aquí empieza la programación realizada en este
trabajo

% Parte del inicio de la interfaz

% Se ejecuta al presionar el botón Start
function Start_Callback( ~, ~, ~)

set_param
('MandarDatos_Laser_0_01_ControlFuerzas_OrdenadoGuille',
'SimulationCommand', 'start'); % Comando para iniciar el
programa

% Se ejecuta al presionar el botón Stop
function Stop_Callback( ~, ~, ~)
```



```
set_param('MandarDatos_Laser_0_01_ControlFuerzas_OrdenadoGuille'  
, 'SimulationCommand', 'stop'); % Comando para parar el programa
```

```
% Parte de las entradas de la interfaz
```

```
function editx_Callback(hObject, ~, ~)  
x=0 % se inicializa variable x
```

```
x=get(hObject,'String') % Se obtiene parámetro escrito en la  
interfaz(hObject) y se almacena en x
```

```
set_param('MandarDatos_Laser_0_01_ControlFuerzas_OrdenadoGuille/  
X','value',num2str(x)) % Se manda el parámetro obtenido al  
bloque del programa correspondiente, se manda como número
```

```
function editx_CreateFcn(hObject, ~, ~)% Crea el cuadro de texto
```

```
if ispc && isequal(get(hObject,'BackgroundColor'),
```

```
get(0,'defaultUicontrolBackgroundColor'))  
    set(hObject,'BackgroundColor','white');  
end
```

```
function edity_Callback(hObject, ~, ~) %Función similar a  
editx_Callback
```

```
y=0  
y=get(hObject,'String')  
set_param('MandarDatos_Laser_0_01_ControlFuerzas_OrdenadoGuille/  
Y','value',num2str(y))
```

```
function edity_CreateFcn(hObject, ~, ~) %Función similar a  
editx_CreateFcn
```

```
if ispc && isequal(get(hObject,'BackgroundColor'),  
get(0,'defaultUicontrolBackgroundColor'))  
    set(hObject,'BackgroundColor','white');  
end
```

```
function editvelocidad_Callback(hObject, ~, ~)%Función similar a  
editx_Callback
```

```
v=0  
v=get(hObject,'String')
```

```

set_param('MandarDatos_Laser_0_01_ControlFuerzas_OrdenadoGuille/
V','value',num2str(v))

function editvelocidad_CreateFcn(hObject, ~, ~) %Función similar
a editx_CreateFcn

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(

% Se ejecuta cuando se presiona el botón Activar/desactivar
function radiobutton2_Callback(hObject, ~, ~)

b=0; %Se inicializa la variable b
b=get(hObject,'Value'); % Se toma el valor del botón, es decir
0si está sin pulsar y 1 si está pulsado

global X
global T
global Y
global i % Se crean 4 variables globales, se emplearán para
graficar más adelante

X=[]
Y=[]
T=[] % Se inicializan como vectores donde se almacenarán los
datos a graficar
global a
global c
global d
c=0
d=0
a=0 % Se crean otras tres variables, sirven para poder graficar
en tiempo real cuando se cumpla la condición, es decir que el
botón este pulsado

i=1 % Variable para ir almacenando en los vectores

if b==0 %Si el botón no está presionado

set_param('MandarDatos_Laser_0_01_ControlFuerzas_OrdenadoGuille/
Reiniciar1','value',num2str(b));

set_param('MandarDatos_Laser_0_01_ControlFuerzas_OrdenadoGuille/
Reiniciar2','value',num2str(b)); % Se mantienen a 0 los bloques
requeridos

else % Si está pulsado

```

```

set_param('MandarDatos_Laser_0_01_ControlFuerzas_OrdenadoGuille/
Reiniciar1','value',num2str(b));

set_param('MandarDatos_Laser_0_01_ControlFuerzas_OrdenadoGuille/
Reiniciar2','value',num2str(b)); % Se ponen a 1 los bloques
requeridos y el sistema se activa

while get(hObject,'Value') % Se crea un bucle cuando el sistema
está activo, se emplea para mostrar los datos en tiempo real

t=get_param('MandarDatos_Laser_0_01_ControlFuerzas_OrdenadoGuille/
GT','RuntimeObject'); %Se almacena el tiempo

xm1=get_param('MandarDatos_Laser_0_01_ControlFuerzas_OrdenadoGui
lle/G1','RuntimeObject'); % Se toma el valor del bloque de
Simulink
xm1b=num2str(xm1.OutputPort(1).Data); %Se pasa a la interfaz a
través del Puerto de salida que existe
set(handles.valorxm1,'string',xm1b); % Se muestra en el texto
creado

%De esta manera se muestran en tiempo real los valores deseados
y se ven en la parte de las salidas

%El resto de la programación para los demás datos es idéntica,
solo cambia el bloque del que se coge el dato y el texto en el
que se muestra

xm2=get_param('MandarDatos_Laser_0_01_ControlFuerzas_OrdenadoGui
lle/G2','RuntimeObject');
xm2b=num2str(xm2.OutputPort(1).Data);
set(handles.valorxm2,'string',xm2b);

ym1=get_param('MandarDatos_Laser_0_01_ControlFuerzas_OrdenadoGui
lle/G3','RuntimeObject');
ym1b=num2str(ym1.OutputPort(1).Data);
set(handles.valorym1,'string',ym1b);

ym2=get_param('MandarDatos_Laser_0_01_ControlFuerzas_OrdenadoGui
lle/G4','RuntimeObject');
ym2b=num2str(ym2.OutputPort(1).Data);
set(handles.valorym2,'string',ym2b);

xref=get_param('MandarDatos_Laser_0_01_ControlFuerzas_OrdenadoGu
ille/G5','RuntimeObject');
xrefb=num2str(xref.OutputPort(1).Data);
set(handles.valorxref,'string',xrefb);

xm=get_param('MandarDatos_Laser_0_01_ControlFuerzas_OrdenadoGuil
le/G6','RuntimeObject');
xmb=num2str(xm.OutputPort(1).Data);
set(handles.valorxm,'string',xmb);

```

```

yref=get_param('MandarDatos_Laser_0_01_ControlFuerzas_OrdenadoGuille/G7','RuntimeObject');
yrefb=num2str(yref.OutputPort(1).Data);
set(handles.valoryref,'string',yrefb);

ym=get_param('MandarDatos_Laser_0_01_ControlFuerzas_OrdenadoGuille/G8','RuntimeObject');
ymb=num2str(ym.OutputPort(1).Data);
set(handles.valorym,'string',ymb);

al=get_param('MandarDatos_Laser_0_01_ControlFuerzas_OrdenadoGuille/G9','RuntimeObject');
alb=num2str(al.OutputPort(1).Data);
set(handles.valoralfa,'string',alb);

vx=get_param('MandarDatos_Laser_0_01_ControlFuerzas_OrdenadoGuille/G10','RuntimeObject');
vxb=num2str(vx.OutputPort(1).Data);
set(handles.velocidadx,'string',vxb);

vy=get_param('MandarDatos_Laser_0_01_ControlFuerzas_OrdenadoGuille/G11','RuntimeObject');
vyb=num2str(vy.OutputPort(1).Data);
set(handles.velocidady,'string',vyb);

X(i)=xm.OutputPort(1).Data;
T(i)=t.OutputPort(1).Data;
Y(i)=ym.OutputPort(1).Data; %Se almacenan lo datos en cada vector
pause(0.19); %Se pausa el bucle

if a==1 % Cuando el botón está activado se gráfica con los datos almacenados y con el drawnow se van actualizando los datos viéndolo en tiempo real.
figure(1)
labelx('Xm(um)')
labely('Ym(um)')
plot(X,Y)
drawnow
end

if c==1
figure(2)
labelx('Tiempo(s)')
labely('Xm(um)')
plot(T,X)
drawnow
end

if d==1
figure(3)
labelx('Tiempo(s)')
labely('Ym(um)')
plot(T,Y)

```

```

drawnow
end

i=i+1; % se suma uno para almacenar los datos en el vector
correctamente
end
end

% Parte de las salidas de la interfaz, gráficas

% Se ejecuta cuando se presiona el botón X
function X_Callback(hObject, eventdata, handles)

set_param('MandarDatos_Laser_0_01_ControlFuerzas_OrdenadoGuille/
Xref, Xm1 , Xm2, Xmedia', 'open', 'on'); % Abre el scope con las
posiciones del eje X

% Se ejecuta cuando se presiona el botón Y
function Y_Callback(hObject, eventdata, handles)

set_param('MandarDatos_Laser_0_01_ControlFuerzas_OrdenadoGuille/
Yref, Ym4, Ym3, Ymedia', 'open', 'on'); % Abre el scope con las
posiciones del eje Y

% Se ejecuta al presionar el botón X/Y
function X/Y_Callback(hObject, eventdata, handles)

global a;
a=0;
a=get(hObject, 'Value'); %Se almacena 1 en a si el botón está
pulsado, 0 si no lo está

% Se ejecuta al presionar el botón Xref/Xm
function Xref/Xm_Callback(hObject, eventdata, handles)

global c;
c=0;
c=get(hObject, 'Value');

```

```
% Se ejecuta al presionar el botón Yref/Ym
function pushbutton10_Callback(hObject, eventdata, handles)

global d;
d=0;
d=get(hObject, 'Value');
```