

Trabajo Fin de Grado  
Grado en Ingeniería de Tecnologías Industriales

Calibración y Control de una Plataforma de Stewart  
para el Análisis de Sensores de Orientación

Autor: Antonio Ramón  
Belderrain Domínguez

Tutores: José Luis Escalona  
Franco

Pedro Urda Gómez

Dpto. Ingeniería Mecánica y Fabricación  
Escuela Técnica Superior de Ingeniería  
Universidad de Sevilla

Sevilla, 2023





Trabajo Fin de Grado  
Grado en Ingeniería de Tecnologías Industriales

# **Calibración y Control de una Plataforma de Stewart para el Análisis de Sensores de Orientación**

Autor:

Antonio Ramón Belderrain Domínguez

Tutores:

José Luis Escalona Franco

Catedrático de Universidad

Pedro Urda Gómez

Profesor Ayudante Doctor

Dpto. de Ingeniería Mecánica y Fabricación

Escuela Técnica Superior de Ingeniería

Universidad de Sevilla

Sevilla, 2023



Trabajo Fin de Grado: Calibración y Control de una Plataforma de Stewart para el Análisis de Sensores de Orientación

Autor: Antonio Ramón Belderrain Domínguez

Tutores: José Luis Escalona Franco  
Pedro Urda Gómez

El tribunal nombrado para juzgar el Proyecto arriba indicado, compuesto por los siguientes miembros:

Presidente:

Vocales:

Secretario:

Acuerdan otorgarle la calificación de:

Sevilla, 2023

El Secretario del Tribunal



*A mi familia*

*A mis amigos*

*A mis profesores*





# Agradecimientos

---

Quiero comenzar agradeciendo a todas las personas que han estado a mi lado hasta hoy en día y me han hecho sentir su apoyo.

De la misma forma, agradecer a todos los profesores que han compartido su conocimiento conmigo y que, a la vez, me han enseñado valores muy importantes para la vida.

Me gustaría agradecer especialmente a mi familia, por estar siempre conmigo, por educarme, por llevarme por el camino correcto y por facilitarme las oportunidades que me han hecho llegar aquí.

A mis tutores, José Luis Escalona y Pedro Urda, quienes han estado a mi disposición para cualquier situación y me han enseñado muchísimo.

*Antonio Ramón Belderrain Domínguez*

*Sevilla, 2023*



**E**n este trabajo se busca realizar la puesta en marcha de una plataforma Stewart y, entonces, tratar de optimizar un filtro de Kalman para la estimación de la inclinación de una bicicleta para así resolver correctamente la cinemática de la misma durante las prácticas de la asignatura Cinemática y Dinámica de Máquinas y determinar si las mediciones dinámicas del inclinómetro empleado durante las mismas son correctas.

Para poder llevar esto a cabo, este trabajo se centra principalmente en la puesta en marcha de la plataforma, desde el entendimiento de sus sistemas de movimientos y el ajuste de sus parámetros de funcionamiento hasta la adaptación de un modelo teórico para que se corresponda con la plataforma real y su empleo para la verificación de los movimientos.

A la vista de los resultados durante el desarrollo del trabajo, la optimización del filtro de Kalman y la verificación de las mediciones dinámicas del inclinómetro, quedan para realizarse en un futuro, tras mejorar el sistema de control de la plataforma siguiendo alguna de las líneas futuras que se plantean.



# Abstract

---

This work seeks to implement a Stewart Platform and, then, try to optimize a Kalman Filter to estimate the inclination of a bicycle in order to correctly solve its kinematics during the practices of the Kinematics and Dynamics of Machines subject and determine if the dynamic measurements of the inclinometer used during them are correct.

In order to carry this out, this work focuses mainly on the implementation of the platform, from the understanding of its movement systems and the adjustment of its operating parameters to the modification of a theoretical model of the platform so it corresponds to the real platform and then, use it to verify movements.

In view of the results during the development of the work, the optimization of the Kalman Filter and the verification of the dynamic measurements of the inclinometer remain to be carried out in the future, after improving the control system of the platform following some of the future lines that are proposed.



<b>Agradecimientos</b>	<b>ix</b>
<b>Resumen</b>	<b>xi</b>
<b>Abstract</b>	<b>xiii</b>
<b>Índice</b>	<b>xv</b>
<b>Índice de Tablas</b>	<b>xvii</b>
<b>Índice de Figuras</b>	<b>xix</b>
<b>Notación</b>	<b>xxi</b>
<b>1 Introducción</b>	<b>1</b>
<b>2 Equipos, instrumentación y utillaje</b>	<b>5</b>
2.1 Descripción de las prácticas de Cinemática y Dinámica de Máquinas	5
2.2 Plataforma Stewart	5
2.3 Modelo de la plataforma	7
2.4 Sensores	12
2.4.1 Unidad de medición inercial.	12
2.4.2 Inclinómetro	13
2.5 Dispositivo de embebido	13
2.6 Utillaje	14
2.7 Estimación de los órdenes de magnitud de frecuencia y amplitud del ángulo de inclinación de una bicicleta	15
<b>3 Verificación de posiciones y movimientos</b>	<b>17</b>
3.1 Procedimiento de verificación previo al diseño del utillaje	17
3.2 Procedimiento de verificación de movimiento	18
<b>4 Caracterización del inclinómetro</b>	<b>29</b>
<b>5 Conclusiones</b>	<b>31</b>
5.1 Posibles causas de la baja precisión de la plataforma Stewart	31
5.2 Líneas de investigación futuras	31
<b>Referencias</b>	<b>35</b>
<b>Anexo A Códigos de Matlab</b>	<b>37</b>
A.1. Archivo para controlar la plataforma con un joystick mediante movimientos punto a punto "joystick_pto_pto.m"	37
A.2. Archivo para controlar la plataforma con un joystick mediante control por trayectoria "joystick_trayectoria.m"	39
A.3. Archivo para controlar la plataforma controlando cada actuador "control_actuadores.m"	41
A.4. Archivo para realizar la caracterización del inclinómetro moviendo la plataforma según un seno "ensayo_seno.m"	44
A.5. Archivo para realizar las gráficas de la calibración de la sección 3.2 "graficas_calibracion.m"	46
A.6. API proporcionada por el fabricante "StewartPro_Scenario.m"	47
A.7. Archivo para realizar la calibración "calibracioninicial.m"	48

<b>Anexo B</b>	<b>Códigos del modelo de Simulon</b>	<b>51</b>
B.1.	<i>Función que define las restricciones cinemáticas de la plataforma Stewart "AnimaStewart.m"</i>	51
B.2.	<i>Función que define las restricciones de movilidad de la plataforma Stewart "RestrUsuario.m"</i>	55
B.3.	<i>Función que define las primeras derivadas temporales de las restricciones de movilidad de la plataforma Stewart "dtRestr.m"</i>	56
B.4.	<i>Función que define las segundas derivadas temporales de las restricciones de movilidad de la plataforma Stewart "DtdtRestr.m"</i>	57
B.5.	<i>Función que define la estimación inicial de las coordenadas de la plataforma Stewart "EstimacionInicial.m"</i>	58
B.6.	<i>Función que calcula la longitud de cada barra de la plataforma Stewart "LongitudBarras.m"</i>	59
B.7.	<i>Código que resuelve la cinemática de la plataforma Stewart "SimulacionCinematicaPlatStew.m"</i>	60
<b>Anexo C</b>	<b>Plano del utillaje</b>	<b>63</b>



# ÍNDICE DE TABLAS

---

Tabla 2-1 Características de la plataforma Stewart según el fabricante	6
Tabla 2-2 Correspondencia entre sólidos y elementos de la plataforma	9
Tabla 2-3 Datos adquiridos por la IMU	13
Tabla 3-1 Resultados del procedimiento previo de calibración	18
Tabla 3-2 Error relativo en posicionamiento	18
Tabla 3-3 Puntos de calibración	19
Tabla 3-4 Longitud medida del vástago de cada actuador	20
Tabla 3-5 Longitud medida de los vástagos en el punto de referencia	21
Tabla 3-6 Desplazamiento medido de cada vástago	21
Tabla 3-7 Longitud teórica de los actuadores en el punto de referencia	22
Tabla 3-8 Longitud teórica de los actuadores	22
Tabla 3-9 Desplazamiento teórico de los actuadores	23
Tabla 3-10 Relación desplazamiento medido entre desplazamiento teórico	24



# ÍNDICE DE FIGURAS

---

Figura 1-1 Máquina de ensayos de neumáticos de Eric Gough	1
Figura 1-2 Primer simulador de conducción de Mercedes-Benz en 1985	2
Figura 2-1 Modelo simplificado de la plataforma Stewart	8
Figura 2-2 Numeración de los actuadores de la plataforma Stewart	8
Figura 2-3 Ángulos entre las juntas cardan de la base fija	9
Figura 2-4 Ángulos entre las juntas cardan en la plataforma móvil	9
Figura 2-5 IMU 3DM-GX4-25	12
Figura 2-6 Inclinómetro NS-15/DPG2-RUG	13
Figura 2-7 Dispositivo de embebido Ni myRIO	14
Figura 2-8 Utillaje	14
Figura 2-9 Inclinación de una bicicleta durante un <i>sprint</i>	15
Figura 3-1 Utillaje provisional	17
Figura 3-2 Ejemplo de cómo se han realizado las mediciones	20
Figura 3-3 Relación entre los desplazamientos medidos y los teóricos cuando aumenta el ángulo de cabeceo	26
Figura 3-4 Relación entre los desplazamientos medidos y los teóricos cuando aumenta el ángulo de alabeo	26
Figura 3-5 Relación entre los desplazamientos medidos y los teóricos cuando aumenta la coordenada $X$	27
Figura 3-6 Relación entre los desplazamientos medidos y los teóricos cuando aumenta la coordenada $Y$	27
Figura 3-7 Relación entre los desplazamientos medidos y los teóricos cuando aumenta la coordenada $Z$	28
Figura 5-1 Modelo de la plataforma Stewart con sensores láser	32



# Notación

$\cos$	Función cos
$\sin$	Función seno
FDM	Modelado por deposición de fundido
IMU	Inertial measurement unit
AHRS	Attitude and Heading Reference Systems
g.d.l.	Grados de libertad
GPS	Global Positioning System
Control PI	Controlador proporcional-integrador
$\mathbf{q}^i$	Vector de coordenadas del sólido $i$
$\dot{\mathbf{q}}$	Primera derivada temporal del vector de coordenadas
$\ddot{\mathbf{q}}$	Segunda derivada temporal del vector de coordenadas
$\mathbf{r}^i$	Vector de posición del sólido $i$
$\boldsymbol{\theta}^i$	Parámetros del Euler del sólido $i$
$R_f$	Radio de la circunferencia en la que se encuentran los actuadores en la base fija de la plataforma Stewart
$R_m$	Radio de la circunferencia en la que se encuentran los actuadores en la plataforma móvil de la plataforma Stewart
$\alpha_f$	Ángulo entre las parejas de actuadores más próximos entre sí de la base fija de la base fija
$\beta_f$	Ángulo entre las parejas de actuadores más lejanos entre sí de la base fija de la base fija
$\alpha_m$	Ángulo entre las parejas de actuadores más próximos entre sí de la base fija de la plataforma móvil
$\beta_m$	Ángulo entre las parejas de actuadores más lejanos entre sí de la base fija de la plataforma móvil
$\varphi_f$	Ángulo entre el sexto actuador sobre la base fija y el eje $X$ del sistema de referencia de la base fija
$\varphi_m$	Ángulo entre el primer actuador en la unión con la plataforma móvil y el eje $X$ del sistema de referencia de la plataforma móvil
$z_b$	Altura de las juntas cardan respecto a la base fija de la plataforma Stewart
$z_m$	Altura de las juntas cardan respecto a la plataforma móvil de la plataforma Stewart
$\bar{\mathbf{u}}_p^i$	Vector en coordenadas locales del sólido $i$ que localiza el punto P
$\bar{\mathbf{v}}^i$	Vector en coordenadas locales del sólido $i$ que está alineado con el eje de giro que permite un par cilíndrico
$\mathbf{A}^i$	Matriz de giro del sólido $i$
$\mathbf{C}(\mathbf{q}, t)$	Vector de restricciones
$\mathbf{D}$	Jacobiano de las restricciones
$\mathbf{E}$	Derivada temporal de las restricciones

$\dot{D}$	Derivada temporal del jacobiano de las restricciones
$\ddot{E}$	Segunda derivada temporal de las restricciones
$A^{-1}$	Inversa de la matriz $A$
$A^T$	Traspuesta de la matriz $A$
$n^\circ$	Número
$r_{PR}$	Vector entre los puntos P y R
$n$	Vector normal a un plano

# 1 INTRODUCCIÓN

---

*Cualquiera que deje de aprender es viejo, ya sea a los veinte u ochenta. Cualquiera persona que sigue aprendiendo se mantiene joven. La cosa más grande en la vida es mantener la mente joven.*

*- Henry Ford -*

Este trabajo surge por dos motivos, el primero de ellos para buscar solución a un problema ocurrido durante la realización de las prácticas de la asignatura Cinemática y Dinámica de Máquinas del tercer curso de este grado. En estas prácticas se emplea un filtro de Kalman para la estimación de la inclinación de la bicicleta a partir de las mediciones de una IMU, donde las covarianzas son estimadas como consideraciones generales. Esto provoca que, los resultados del ángulo medido por un inclinómetro y los obtenidos mediante el filtro de Kalman a partir de los datos proporcionados por una IMU difieran entre sí. Con este trabajo se busca optimizar dicho filtro de Kalman para obtener unos mejores resultados y verificar si las mediciones dinámicas del inclinómetro son correctas.

El otro de los motivos por los que se realiza este trabajo es la adquisición por parte del laboratorio de Ingeniería Mecánica de esta escuela de una plataforma Stewart. Dado que la misma estaba sin usar, había que empezar desde el principio, es decir, comprender su funcionamiento, adaptar su sistema de funcionamiento para que pueda ser usada en futuras investigaciones, así como verificar si sus movimientos se corresponden con los ordenados.

Con todo esto, es interesante comenzar este trabajo hablando sobre las plataformas Stewart y sus aplicaciones en la industria. Las plataformas Stewart son consideradas como un robot paralelo que generalmente presenta seis barras articuladas, aprovechando la rigidez conseguida gracias a estas seis barras logra movimientos precisos en seis grados de libertad. Este tipo de robot paralelo fue desarrollado por Eric Gough en 1954 para realizar ensayos de neumáticos [1], Figura 1-1, pero fue Stewart quien publicó un artículo en 1965 para el Colegio de Ingenieros Mecánicos de Reino Unido sobre la plataforma. Desde entonces, fue conocida como la plataforma Stewart y se ha empleado en una gran variedad de campos y aplicaciones como la industria aeroespacial [2] y automovilística [3], Figura 1-2, la cirugía, la calibración de sensores o incluso en la industria del entretenimiento en atracciones temáticas y simuladores de realidad virtual [4].



Figura 1-1 Máquina de ensayos de neumáticos de Eric Gough

La plataforma Stewart consta de una base fija y una plataforma móvil, con diámetros que van desde centímetros hasta varios metros, ambas conectadas entre sí por seis brazos articulados mediante diferentes tipos de uniones según la aplicación. Del mismo modo, según la aplicación los actuadores varían. Es por su versatilidad que son tan usadas en aplicaciones muy diversas.



Figura 1-2 Primer simulador de conducción de Mercedes-Benz en 1985

Por otro lado, el filtro de Kalman es un algoritmo desarrollado por Rudolf E. Kalman en 1960 [5]. Este algoritmo tiene como objetivo estimar las variables de estado de un sistema que varía con el tiempo a partir de mediciones indirectas. Las variables de estado del sistema son variables físicas del mismo como por ejemplo aceleración, velocidad o posición del sólido rígido que se está moviendo. Está basado principalmente en dos pasos, predicción y corrección, pues las mediciones indirectas presentan ruido que provoca cierta incertidumbre y, las variables de estado pueden depender de otras variables no medidas, por lo que se debe de corregir la primera estimación. Es por ello que, para optimizar la predicción hay que filtrar la señal para eliminar el ruido de las mediciones para poder así obtener una buena predicción. Para realizar la predicción, el algoritmo emplea un modelo teórico del sistema y los datos del instante anterior para calcular una primera estimación del estado actual. Para la corrección, emplea la medición actual y la primera predicción para realizar una nueva predicción que minimice el error cuadrático medio [6].

El filtro de Kalman está presente en muchas aplicaciones tecnológicas, pero destaca su uso en el campo de la localización satelital, la navegación, tanto marítima como aérea, así como, en el control de vehículos. Dentro de las aplicaciones más importantes se encuentran:

- Durante el programa Apollo se empleó el filtro de Kalman para la estimación de la trayectoria y la orientación de las naves.
- El sistema GPS lo utiliza para combinar las medidas de los satélites con otros sensores como acelerómetros, giróscopos o magnetómetros para tener una estimación de la posición más precisa.
- Los sistemas de reconocimiento facial lo emplean para el rastreo de los rasgos faciales de una persona en una secuencia de imágenes y lograr adaptarse ante cambios de posición, expresión o de iluminación.
- En los sistemas de control de estabilidad de los vehículos terrestres se utiliza para estimar la velocidad y dirección de cada rueda para poder aplicar la fuerza de frenado necesaria y así evitar el derrape.

Durante las prácticas de Cinemática y Dinámica de Máquinas se emplea un filtro de Kalman en Matlab desarrollado por Jouni Hartikainen, Arno Solin, y Simo Särkkä, del departamento de Ingeniería Biomédica y Ciencia Computacional de la Universidad de Ciencia y Tecnología de Aalto [7]. Por lo tanto, es este el filtro que se busca optimizar mediante máxima verosimilitud para la correcta realización de las prácticas. Para poder mejorar los resultados durante este trabajo se empleará la plataforma de Stewart para determinar la relación que tienen entre sí las mediciones que son objeto de entrada al filtro, es decir, las mediciones a partir de las cuales se



determina el estado del sistema. En este caso, el estado a determinar del sistema es la inclinación de una bicicleta, mientras que, las mediciones realizadas por una IMU son la velocidad angular de alabeo y la aceleración lateral, siendo también variable de entrada el período de adquisición de datos.



# 2 EQUIPOS, INSTRUMENTACIÓN Y UTILLAJE

---

*Perseverancia, secreto de todos los triunfos.*

*- Victor Hugo -*

En este capítulo se lleva a cabo algo fundamental en cualquier investigación, una descripción de los equipos, instrumentación y utillaje empleados durante todo el desarrollo de la misma. En este caso, al surgir la idea de este trabajo ante un problema durante las prácticas de la asignatura Cinemática y Dinámica de Máquinas de tercer curso de este grado, se emplean los mismos sensores que en dichas prácticas. Por otro lado, es importante conocer las características de la plataforma Stewart con la que se trabaja, así como, el utillaje empleado para la colocación de los sensores en la plataforma. Además, se hace una descripción tanto del modelo de la plataforma Stewart como de las prácticas de Cinemática y Dinámica de Máquinas y un análisis de las frecuencias típicas de oscilación del ángulo de inclinación en las bicicletas.

## 2.1 Descripción de las prácticas de Cinemática y Dinámica de Máquinas

Es importante realizar una descripción de la bicicleta, la disposición de los sensores empleados, así como de las prácticas de la asignatura Cinemática y Dinámica de Máquinas [8] pues este trabajo surge para solucionar un problema ocurrido durante la realización de las mismas. En estas prácticas, se realiza una simulación cinemática y dinámica inversa del movimiento de una bicicleta a partir de medidas experimentales [9], pero este trabajo se centra en la cinemática, pues es ahí donde en primera instancia surge el problema. Siguiendo con la cinemática, se emplean coordenadas de referencia a partir del centro de gravedad del sistema bicicleta-ciclista y las mediciones de los sensores para imponer las restricciones de movilidad en el ángulo del manillar, de la rueda trasera y de la inclinación.

Para obtener las mediciones experimentales durante estas prácticas, se emplean los siguientes sensores: IMU e inclinómetro colocados en el cuadro de la bicicleta, un *encoder* en la rueda trasera y otro en el manillar [8], todos ellos con una frecuencia de adquisición de 100 Hz. Para este trabajo solo son de interés los datos recopilados por la IMU y el inclinómetro, pues el filtro de Kalman que se emplea en las prácticas obtiene la inclinación a partir de los datos de la IMU, pero los resultados difieren de las mediciones realizadas por el inclinómetro. Señalar también, que las mediciones de velocidad y aceleración, son filtradas con un filtro de Butterworth paso bajo de 4º orden con frecuencia de corte 5 Hz.

## 2.2 Plataforma Stewart

Se trabaja con una plataforma Stewart comercial, fabricada y vendida por ACROME. En este caso, es el modelo ACROME Stewart Pro, un modelo que tiene diferentes configuraciones según las peticiones del cliente. La plataforma que se encuentra en el laboratorio, consta de juntas cardan colocadas sobre rodamientos para la conexión entre los actuadores lineales y las bases, fija y móvil, de la plataforma, presentando, según el fabricante [10], las características que aparecen en la Tabla 2-1.

Tabla 2-1 Características de la plataforma Stewart según el fabricante

Características	
Repetibilidad	50 $\mu\text{m}$ y 0,5°
Velocidad de los actuadores	10 mm/s
Rango de funcionamiento en X e Y	-100÷100 mm
Rango de funcionamiento en Z	385÷480 mm
Rango de funcionamiento de alabeo, cabeceo y guiñada	-20°÷20°
Rango de funcionamiento de los actuadores	0÷100 mm
Carga máxima	50 kg

Por otro lado, la plataforma cuenta con su sistema de distribución y su propia tarjeta de alimentación. Igualmente, cuenta con una unidad de microprocesador que contiene un giróscopo y un acelerómetro, siendo el modelo MPU-6050 Six-Axis. Para controlar la plataforma, esta trae un dispositivo de embebido de National Instruments, Ni myRIO-1900, que controla los actuadores lineales a través de la tarjeta de alimentación. La conexión con el ordenador se realiza por USB, aunque presenta también la posibilidad de conexión Wi-Fi, pero esta conexión se reserva para los sensores instalados sobre la plataforma.

La plataforma puede ser actuada de tres formas distintas. La primera es un control punto a punto, donde no se conoce ni la trayectoria ni el tiempo en el que va a realizar el movimiento. La segunda, es mediante un control por trayectoria, donde se interpola un polinomio de tercer grado para realizar el movimiento en un tiempo indicado. Para trabajar con estas formas, debe de activarse previamente la *Kinematic Solution* y encender los motores. Por último, la plataforma puede ser controlada actuando sobre cada actuador lineal, de uno en uno, activando previamente el control de los motores. Es por ello que, para el propósito de este trabajo, la opción más interesante es el control de los actuadores. El código para el control del movimiento de la plataforma según los actuadores puede verse en el Anexo A.3. Esta era la primera idea, pues permite un control más preciso de la plataforma, pero las limitaciones de la misma, hacen que el movimiento no sea natural ni suave, pues los actuadores se actúan de uno en uno y el tiempo entre el movimiento de cada actuador es lo suficientemente grande como para que no se ejecute el movimiento de la plataforma en el tiempo deseado. Es por ello que, se decide trabajar con el control por trayectoria, pues se obtiene una mejor respuesta de la plataforma.

La plataforma tiene un sistema de control PI, al cual se le pueden modificar el valor de las constantes  $K_p$  y  $T_i$ . Esto puede realizarse desde su propia interfaz gráfica de usuario, *Stewart Pro Application*, desde donde se pueden cambiar varios parámetros de funcionamiento, tanto las constantes del controlador, una constante que tiene en cuenta la compensación por fricción respecto al modelo ideal, la zona de banda muerta del controlador sobre los actuadores lineales, así como, limitar las extensiones máximas y mínimas de los actuadores. Todo esto puede ser modificado mediante órdenes desde Matlab o con Python.

Desde la misma interfaz de usuario puede iniciarse la plataforma y mandar órdenes de movimiento de cualquiera de las tres formas. Esto está muy limitado en cuanto a utilidad más allá de situar la plataforma en determinadas posiciones, pues no permite comunicarle otra orden de movimiento hasta una vez finalizada la orden anterior.

Igualmente, la plataforma puede ser controlada con Python o Matlab, siendo ambas opciones más interesantes pues mediante un bucle pueden mandarse órdenes de movimiento seguidamente y, conseguir así, un movimiento preciso y constante. También, pueden ser modificados los parámetros de funcionamiento de esta forma. Es por ello que este trabajo se desarrolla con Matlab.

En una primera fase para entender el funcionamiento de la plataforma se comienza trabajando sobre la API proporcionada por el fabricante, Anexo A.6, mediante la cual, se inicia la plataforma y se realizan movimientos punto a punto. Gracias a esto, se ve que el sistema de comunicación con la máquina está basado en un mensaje con un prefijo el cual es la longitud del propio mensaje.

Tras esto, se pasa a controlar la plataforma con un joystick y con movimiento tanto punto a punto como por trayectoria. Esto se realiza con un script de Matlab que lee en tiempo real los inputs del joystick y mantiene una comunicación constante con la plataforma. El caso del control punto a punto puede verse en el Anexo A.1 y, el control por trayectoria en el Anexo A.2. Para ello se estiman previamente unas velocidades máximas para cada componente sin tener en cuenta las aceleraciones pues el objetivo de controlar la plataforma con el joystick es

entender mejor sus movimientos y su comportamiento ante diferentes inputs. El joystick genera una señal que va desde menos uno hasta uno, por lo que, se calcula la velocidad instantánea como la señal del joystick por la velocidad máxima. Con esto, la posición queda como la posición en el instante anterior más la velocidad por un incremento de tiempo que se ha considerado igual a  $0,1$  segundos. Para que sea un control continuo de la plataforma, se realiza un bucle en el cual se realiza la lectura del joystick, se calculan velocidad y posición y, se escribe la orden de movimiento con el formato “*move\_platform\_traj,x,y,z,roll,pitch,yaw,t*” donde  $t$  es el tiempo en el que la plataforma realiza el movimiento, es por ello que, toma el mismo valor que el incremento de tiempo empleado para calcular la posición. Por precaución, dentro del bucle se determina si la orden de movimiento se sale del rango de funcionamiento de la plataforma y, en caso de que se salga, se le da el valor máximo, evitando así que la plataforma responda con un error. De nuevo, por simplificar el procedimiento, la duración del control continuo es fijada antes de ejecutar el script de Matlab. Una vez que se termina el bucle, se lleva a la plataforma a un punto de posición y orientación conocidos, se apagan los motores y se finaliza la conexión.

Con este trabajo previo de conocer la plataforma, se aprecia que la plataforma permite almacenar una serie de órdenes de movimientos que se ejecutarán consecutivamente, esto permite trabajar de dos formas muy interesantes, la primera de ellas consiste en emplear un mismo bucle para escribir y mandar las órdenes de movimiento al mismo tiempo, mientras que, esto puede realizarse de forma separada, escribiendo previamente todas las órdenes de movimiento y posteriormente enviarlas a la plataforma. Por otro lado, es posible solicitar respuesta a la plataforma ante las órdenes, pero si se hace así, la nueva orden se manda una vez es recibida la respuesta, por lo que el sistema se vuelve más lento.

Empleando de nuevo el script del control por trayectoria con joystick, se ajustan los parámetros del control del sistema. En este caso, se va aumentando progresivamente el valor de la ganancia,  $K_p$ , hasta que el sistema oscila a velocidad máxima, en este caso  $K_p = 2,8$ . Una vez determinado el valor de  $K_p$  se comienza a variar el valor de  $T_i$  pero el sistema se vuelve rápidamente inestable, con oscilaciones incluso sin moverse, y se llega a la conclusión de dejar su valor por defecto, siendo este cero. Por lo que queda un control proporcional.

Posteriormente, se emplea otro script de Matlab, el cual puede verse en el Anexo A.4, que envía a la plataforma senos de diferentes frecuencias y amplitudes establecidas previamente, a uno de los ángulos de la plataforma. Gracias a esto, en primer lugar, se aprecia que el sistema responde más rápido ante movimientos en los que solo varía una coordenada, por otro lado, cuando aumenta la frecuencia, para el valor de la ganancia determinado anteriormente, el sistema se vuelve inestable. Por ello, para este caso, se trabaja con el valor por defecto,  $K_p = 0,12$ .

## 2.3 Modelo de la plataforma

Para poder comparar movimientos y resultados obtenidos con la plataforma, se trabaja con un modelo teórico de la misma. Este modelo es una adaptación del modelo desarrollado por el profesor Joaquín Ojeda Granja en Simulon, un programa de simulación cinemática y dinámica desarrollado en Matlab por varios profesores del departamento de Ingeniería Mecánica y Fabricación de esta escuela. Para poder adaptar el modelo a la geometría de la máquina, se han medido las posiciones de los actuadores respecto al centro de la misma, así como los ángulos que forman. Este modelo está desarrollado con parámetros de Euler, por lo que cada sólido tiene 7 coordenadas. Por lo tanto, se tienen 14 sólidos, es decir, la base fija, la plataforma móvil y 12 barras que forman los 6 actuadores. Los 14 sólidos dan lugar a 98 coordenadas.

$$\mathbf{q}^i = [r_x^i \ r_y^i \ r_z^i \ \theta_0^i \ \theta_1^i \ \theta_2^i \ \theta_3^i]^T = [\mathbf{r}^i \ \boldsymbol{\theta}^i]^T$$

Por otro lado, los actuadores se modelan como dos barras unidas por un par cilíndrico y las uniones con la base y la plataforma como juntas cardan. Cada par cilíndrico restringe 4 coordenadas, al igual que cada junta cardan. En el sistema se encuentran 12 juntas cardan y 6 pares cilíndricos. Del mismo modo, los parámetros de Euler llevan consigo una restricción, deben ser unitarios. Por lo tanto, considerando que todas las componentes de la base fija son nulas, quedan los siguientes grados de libertad:

$$g = 98 - 12 * 4 - 6 * 4 - 14 - 6 = 6 \text{ g. d. l.}$$

Estos seis grados de libertad son los movimientos de la plataforma móvil. Para poder resolver la cinemática de la plataforma, estos seis grados de libertad están condicionados a las restricciones de movilidad, que mediante código se definen en el Anexo B.2. Sus derivadas temporales primera y segunda se definen en los Anexos B.3

y B.4, respectivamente.

Para realizar la adaptación del modelo a la plataforma con la que se trabaja se han modificado las restricciones cinemáticas de las juntas cardan, así como de la estimación inicial que necesita el modelo para poder resolver correctamente las ecuaciones. En la Figura 2-1 puede verse el modelo de la plataforma con ejes y con los sólidos numerados. Para que las coordenadas de los sólidos correspondan con los de la plataforma con la que se trabaja, se define el sistema de ejes tal que el eje  $X$  pase por el punto medio entre los actuadores 1 y 6, los cuales pueden verse en la Figura 2-2, y quede el eje  $Z$  vertical.

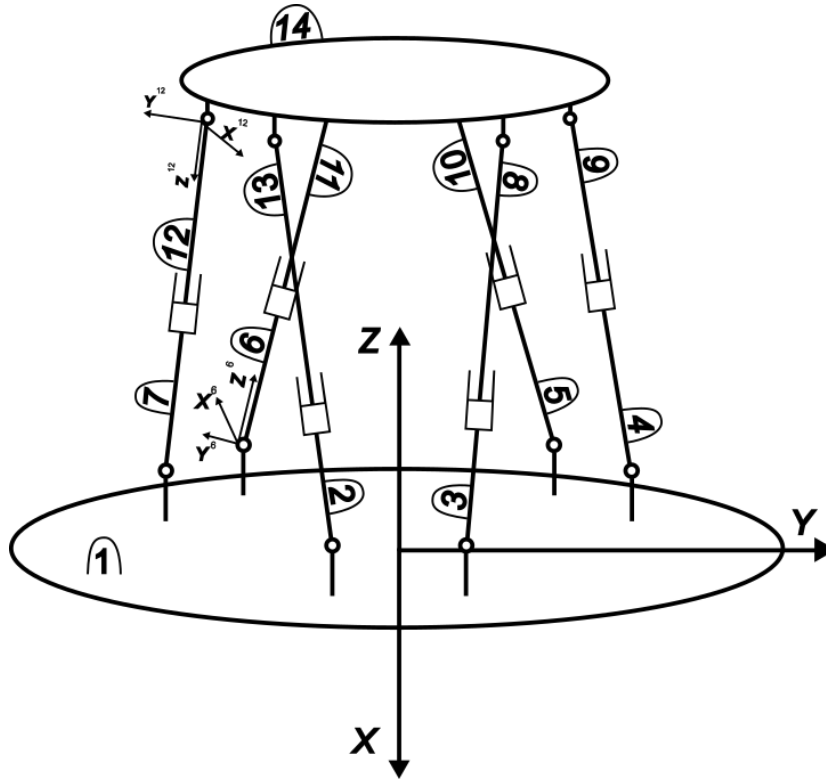


Figura 2-1 Modelo simplificado de la plataforma Stewart

La disposición de los actuadores de la plataforma se aprecia en la Figura 2-2. El centro de las juntas cardan que conectan con la base se encuentran a un radio,  $R_f = 170 \text{ mm}$ , donde  $f$  hace referencia a fija. Mientras que para el caso de las juntas que conectan con la plataforma móvil, estas se encuentran a un radio,  $R_m = 130 \text{ mm}$ , donde  $m$  hace referencia a móvil.



Figura 2-2 Numeración de los actuadores de la plataforma Stewart

Del mismo modo, en la base fija, el ángulo que hay entre dos actuadores que se encuentran más próximos, es decir, 6-1, 2-3 y 4-5, es  $\alpha_f = 30^\circ$ . Para las parejas de actuadores que están más alejadas, es decir, 1-2, 3-4 y 5-6, es  $\beta_f = 90^\circ$ .

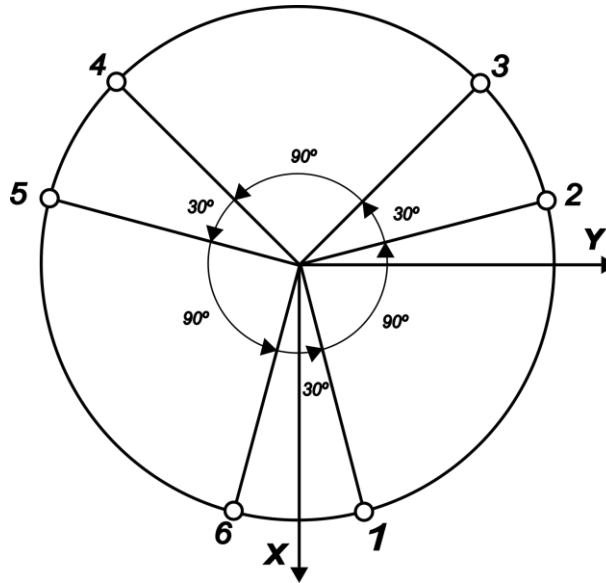


Figura 2-3 Ángulos entre las juntas cardan de la base fija

En el caso de la plataforma móvil, esta se encuentra girada  $32,5^\circ$  alrededor de la vertical con respecto a la base fija. Además, los ángulos entre los actuadores más próximos que en esta son 1-2, 3-4 y 5-6, es  $\alpha_m = 25^\circ$ . Por lo que, para los actuadores más separados, es  $\beta_m = 95^\circ$ .

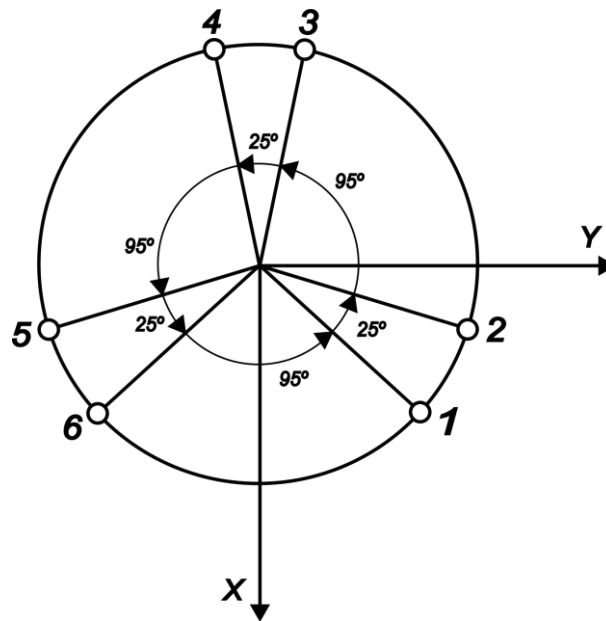


Figura 2-4 Ángulos entre las juntas cardan en la plataforma móvil

También, deben de definirse qué sólido es cada elemento de la plataforma. Esto queda recogido en la Tabla 2-2.

Tabla 2-2 Correspondencia entre sólidos y elementos de la plataforma

Sólido número	Elemento de la plataforma
1	Base fija
2	Parte inferior del actuador número 6
3	Parte inferior del actuador número 1

Sólido número	Elemento de la plataforma
4	Parte inferior del actuador número 2
5	Parte inferior del actuador número 3
6	Parte inferior del actuador número 4
7	Parte inferior del actuador número 5
8	Parte superior del actuador número 1
9	Parte superior del actuador número 2
10	Parte superior del actuador número 3
11	Parte superior del actuador número 4
12	Parte superior del actuador número 5
13	Parte superior del actuador número 6
14	Plataforma móvil

Con estas consideraciones, ya están localizados todos los puntos de interés del sistema, por lo que se puede comenzar con las restricciones, cuyo código puede verse en el Anexo B.1. En este caso, se comienza con las restricciones asociadas a las juntas cardan de la base fija. Es importante empezar definiendo un ángulo  $\varphi_f = -15^\circ$ , que es el ángulo que forma la junta cardan del actuador número seis y el eje  $X$ . Esto se hace así porque el sólido 2 es la parte inferior del actuador 6. Se definen las restricciones para la junta cardan entre los sólidos 1 y 2, para el resto simplemente habría que cambiar el ángulo, sumando al ángulo de la restricción anterior,  $\alpha_f$  si son dos juntas próximas entre sí (6-1, 2-3, 4-5) o bien  $\beta_f$  si son dos juntas que no son cercanas entre sí (1-2, 3-4, 5-6). Igualmente, hay que definir la altura a la que se encuentra el centro de las juntas cardan sobre la base fija,  $z_b = 53,05 \text{ mm}$ .

Se comienza definiendo el vector en coordenadas locales de cada sólido que indica la posición de la junta cardan:

$$\bar{\mathbf{u}}_p^1 = [R_f \cos(\varphi_f) \quad R_f \sin(\varphi_f) \quad z_b]^T$$

$$\bar{\mathbf{u}}_p^2 = [0 \ 0 \ 0]^T$$

Se continúa definiendo un vector en coordenadas locales para cada sólido, los cuales en coordenadas globales se mantienen perpendiculares entre sí.

$$\bar{\mathbf{s}}^1 = [-\sin(\varphi_f) \quad \cos(\varphi_f) \quad 0]^T$$

$$\bar{\mathbf{s}}^2 = [1 \ 0 \ 0]^T$$

Por lo tanto, las restricciones por la junta cardan quedan:

$$\mathbf{C}_{car}^{1-2} = \begin{bmatrix} \mathbf{q}^1(1) + \mathbf{A}^1 \bar{\mathbf{u}}_p^1 - \mathbf{q}^2(1) - \mathbf{A}^2 \bar{\mathbf{u}}_p^2 \\ \mathbf{q}^1(2) + \mathbf{A}^1 \bar{\mathbf{u}}_p^1 - \mathbf{q}^2(2) - \mathbf{A}^2 \bar{\mathbf{u}}_p^2 \\ \mathbf{q}^1(3) + \mathbf{A}^1 \bar{\mathbf{u}}_p^1 - \mathbf{q}^2(3) - \mathbf{A}^2 \bar{\mathbf{u}}_p^2 \\ (\mathbf{A}^1 \bar{\mathbf{s}}^1)^T (\mathbf{A}^2 \bar{\mathbf{s}}^2) \end{bmatrix} = \mathbf{0}$$

Donde las matrices de giro cuando se emplean parámetros de Euler tienen la siguiente forma [11]:

$$\mathbf{A}^i = \begin{bmatrix} 1 - 2(\theta_2^i)^2 - 2(\theta_3^i)^2 & 2(\theta_1^i \theta_2^i - \theta_0^i \theta_3^i) & 2(\theta_1^i \theta_3^i + \theta_0^i \theta_2^i) \\ 2(\theta_1^i \theta_2^i + \theta_0^i \theta_3^i) & 1 - 2(\theta_1^i)^2 - 2(\theta_3^i)^2 & 2(\theta_2^i \theta_3^i - \theta_0^i \theta_1^i) \\ 2(\theta_1^i \theta_3^i - \theta_0^i \theta_2^i) & 2(\theta_2^i \theta_3^i + \theta_0^i \theta_1^i) & 1 - 2(\theta_1^i)^2 - 2(\theta_2^i)^2 \end{bmatrix}$$

Para las juntas cardan de la plataforma móvil, las restricciones son similares. En este caso, las juntas se encuentran por debajo de la plataforma móvil, por lo tanto, la altura de las juntas con respecto a la plataforma es  $z_m = -51,90 \text{ mm}$ . Se muestra el caso de las restricciones para la junta cardan entre los sólidos 14 (plataforma móvil) y 8 (parte superior del actuador n° 1). En este caso, como la plataforma está girada  $32,5^\circ$  alrededor de la



vertical, se define el ángulo  $\varphi_m = 47,5^\circ$ . Este es el ángulo que forma la junta cardan del actuador número 1 con el eje  $X$  de la plataforma. De la misma forma que en el caso anterior, para las restricciones del resto de juntas cardan, habría que variar el ángulo, sumando al ángulo de la restricción anterior,  $\alpha_m$  si son dos juntas próximas entre sí (1-2, 3-4, 5-6) o bien  $\beta_m$  si son dos juntas que no son cercanas entre sí (2-3, 4-5, 6-1).

$$\begin{aligned}\bar{\mathbf{u}}_p^{14} &= [R_m \cos(\varphi_m) \ R_m \sin(\varphi_m) \ z_m]^T \\ \bar{\mathbf{u}}_p^8 &= [0 \ 0 \ 0]^T \\ \bar{\mathbf{s}}^{14} &= [-\text{sen}(\varphi_m) \ \cos(\varphi_m) \ 0]^T \\ \bar{\mathbf{s}}^8 &= [1 \ 0 \ 0]^T \\ \mathbf{C}_{car}^{14-8} &= \begin{bmatrix} \mathbf{q}^{14}(1) + \mathbf{A}^{14}\bar{\mathbf{u}}_p^{14} - \mathbf{q}^8(1) - \mathbf{A}^8\bar{\mathbf{u}}_p^8 \\ \mathbf{q}^{14}(2) + \mathbf{A}^{14}\bar{\mathbf{u}}_p^{14} - \mathbf{q}^8(2) - \mathbf{A}^8\bar{\mathbf{u}}_p^8 \\ \mathbf{q}^{14}(3) + \mathbf{A}^{14}\bar{\mathbf{u}}_p^{14} - \mathbf{q}^8(3) - \mathbf{A}^8\bar{\mathbf{u}}_p^8 \\ (\mathbf{A}^{14}\bar{\mathbf{s}}^{14})^T (\mathbf{A}^8\bar{\mathbf{s}}^8) \end{bmatrix} = \mathbf{0}\end{aligned}$$

Por otro lado, los pares cilíndricos se encuentran entre los dos sólidos que forman parte de un mismo actuador. Se muestra el caso del par cilíndrico entre los sólidos 3 y 8, los cuales conforman el actuador número 1. En primer lugar, se definen los vectores que representan las componentes del unitario alineado con el eje de giro del par cilíndrico en coordenadas locales para cada sólido.

$$\begin{aligned}\bar{\mathbf{v}}^3 &= [0 \ 0 \ 1]^T \\ \bar{\mathbf{v}}^8 &= [0 \ 0 \ 1]^T\end{aligned}$$

Por otro lado, se definen los vectores que representan la posición de dos puntos pertenecientes a cada sólido, P y Q, y a la línea de deslizamiento.

$$\begin{aligned}\bar{\mathbf{u}}_P^3 &= [0 \ 0 \ 0]^T \\ \bar{\mathbf{u}}_Q^8 &= [0 \ 0 \ 0]^T\end{aligned}$$

Para establecer las restricciones hay que imponer que los vectores  $\bar{\mathbf{v}}^3$  y  $\bar{\mathbf{v}}^8$  sean paralelos y que el vector que conecta los dos puntos pertenecientes a los sólidos sea también paralelo a estos. Para imponer el paralelismo entre los dos primeros vectores, se definen dos nuevos vectores que sean perpendiculares al primero:

$$\begin{aligned}(\mathbf{v}_1^3)^T \mathbf{v}^3 &= 0 \\ (\mathbf{v}_2^3)^T \mathbf{v}^3 &= 0\end{aligned}$$

Con esto, las restricciones quedan [11]:

$$\mathbf{C}_{cil}^{3-8} = \begin{bmatrix} (\mathbf{A}^3\bar{\mathbf{v}}_1^3)^T (\mathbf{A}^8\bar{\mathbf{v}}^8) \\ (\mathbf{A}^3\bar{\mathbf{v}}_2^3)^T (\mathbf{A}^8\bar{\mathbf{v}}^8) \\ (\mathbf{A}^3\bar{\mathbf{v}}^3)^T (\mathbf{r}^3 + \mathbf{A}^3\bar{\mathbf{u}}_P^3 - \mathbf{r}^8 + \mathbf{A}^8\bar{\mathbf{u}}_Q^8) \\ (\mathbf{A}^3\bar{\mathbf{v}}^3)^T (\mathbf{r}^3 + \mathbf{A}^3\bar{\mathbf{u}}_P^3 - \mathbf{r}^8 + \mathbf{A}^8\bar{\mathbf{u}}_Q^8) \end{bmatrix} = \mathbf{0}$$

Para el resto de parejas de sólidos unidos por pares cilíndricos, es decir, los sólidos 4-9, 5-10, 6-11, 7-12 y 2-13, se sigue el mismo procedimiento.

Por otro lado, la restricción asociada a que los parámetros de Euler son unitarios puede escribirse [11]:

$$(\boldsymbol{\theta}^i)^T \boldsymbol{\theta}^i - 1 = 0$$

Con esto, las restricciones del sistema quedan:

$$\mathbf{C}(\mathbf{q}, t) = \begin{bmatrix} \mathbf{C}^{pares}(\mathbf{q}) \\ \mathbf{C}^{Euler}(\mathbf{q}) \\ \mathbf{C}^{mov}(\mathbf{q}, t) \end{bmatrix} = \mathbf{0}$$

Para la resolución cinemática, se comienza con una estimación inicial de las coordenadas,  $\mathbf{q}_0(t_0)$ , con el código de Matlab del Anexo B.5. En esta estimación, se considera que la plataforma móvil se encuentra en el punto de referencia  $\mathbf{r}^{14}(t_0) = [0 \ 0 \ 433]^T \text{ mm}$ , dado que ese es el punto medio del campo de funcionamiento de la

plataforma real. A partir de ahí, considerando que el resto de sólidos están unidos a las plataformas, se tiene la estimación inicial al completo.

Tras esto, el primer paso es resolver el problema de posición. Para ello se hace uso de la función *fsolve* de Matlab, a la cual se le aportan las condiciones iniciales y las restricciones cinemáticas. Esto puede verse en el Anexo B.7.

Una vez resuelto el problema de posición, se pasa a resolver el problema de velocidad de la forma:

$$\dot{\mathbf{q}}(t_i) = -\mathbf{D}^{-1}\mathbf{E}(\mathbf{q}(t_i), t_i)$$

Donde  $\mathbf{D}$  es el jacobiano de las restricciones y  $\mathbf{E}$  es la derivada temporal de las mismas.

Con esto, se resuelve el problema de aceleraciones de la forma:

$$\ddot{\mathbf{q}}(t_i) = -\mathbf{D}^{-1}[\dot{\mathbf{D}}(\mathbf{q}(t_i), \dot{\mathbf{q}}(t_i)) + \dot{\mathbf{E}}(\mathbf{q}(t_i), t_i)]$$

Por último, para continuar con la resolución cinemática, la estimación inicial para el instante siguiente se considera la posición en el instante resuelto, esto es:

$$\mathbf{q}_0(t_{i+1}) = \mathbf{q}(t_i)$$

## 2.4 Sensores

Para la adquisición de los datos se emplean los mismos sensores que se emplearon en las prácticas de Cinemática y Dinámica de Máquinas [8]. Estos sensores son una unidad de medición inercial, IMU, y un inclinómetro, conectados ambos a un sistema de embebido de National Instruments, Ni myRIO-1900, adicional al que viene con la propia plataforma.

### 2.4.1 Unidad de medición inercial.

Durante el desarrollo de este trabajo se ha empleado una unidad de medición inercial de la empresa LORD MicroStrain Sensing System. Este tipo de sensores son de uso habitual en la industria en múltiples aplicaciones. En concreto, se ha trabajado con el modelo 3DM-GX4-25.



Figura 2-5 IMU 3DM-GX4-25

Este sensor es un sistema de referencia de inclinación y rumbo (AHRS) en miniatura, el cual emplea microsensores inerciales, integrando giróscopos multiejes, acelerómetros y magnetómetros, así como, realiza lecturas de presión y temperatura, para corregir las mediciones [12]. Estas mediciones se realizan sobre un sistema de referencia ortogonal. Además, combina los sensores y realiza un procesamiento de las mediciones mediante un Filtro de Kalman Adaptativo. Por otro lado, el sensor proporciona las incertidumbres de las mediciones. Los datos proporcionados por el sensor que son de interés para la realización del trabajo quedan recogidos en la Tabla 2-3.

Tabla 2-3 Datos adquiridos por la IMU

Medición	Unidades	Descripción
Aceleraciones	Fuerza gravitacional (g)	Aceleración en tres ejes
Velocidad angular	Radianes por segundo	Velocidad angular en tres ejes

La comunicación se realiza mediante una conexión por puerto serie, a un dispositivo de embebido, Ni myRIO-1900, y, desde este dispositivo al ordenador mediante conexión Wi-Fi.

### 2.4.2 Inclinómetro

El inclinómetro empleado es el modelo NS-15/DPG2-RUG, de la empresa TE Connectivity. Este inclinómetro está pensado para estar colocado en horizontal, y es así como se emplea durante el desarrollo de este trabajo. Es capaz de medir los ángulos según sus ejes  $X$  e  $Y$  en un rango de  $\pm 15^\circ$ , con una resolución de  $0,001^\circ$ . Además, para condiciones ambientales desde  $0^\circ\text{C}$  hasta  $50^\circ\text{C}$  tiene una exactitud de  $0,2^\circ$  [13]. La comunicación se realiza a través del mismo dispositivo de embebido al que se conecta la IMU.



Figura 2-6 Inclinómetro NS-15/DPG2-RUG

### 2.5 Dispositivo de embebido

Este es el dispositivo al cual se conectan los sensores para enviar la información al ordenador mediante conexión Wi-Fi. Este es el modelo Ni myRIO-1900 de National Instruments. Tiene conexiones de entrada y salida a ambos lados del dispositivo, con conectores MXP y SMP, además, presenta entradas y salidas tanto analógicas como digitales, LEDs, un pulsador, un acelerómetro interno, un FPGA Xilinx y un procesador dual-core ARM Cortex-A9 junto con la conexión Wi-Fi [14].



Figura 2-7 Dispositivo de embebido Ni myRIO

Este dispositivo envía los datos recopilados por los sensores mediante conexión Wi-Fi al ordenador y son tratados en un programa de LabVIEW 2017 myRIO. Con este programa, se tratan los datos y son mostrados por pantalla, además de poder registrarlos en archivo de texto sin formato. Gracias a esto, quedan registrados los datos obtenidos durante la experimentación.

Igualmente, otro dispositivo Ni myRIO-1900 es el empleado para establecer la comunicación entre la plataforma Stewart y el ordenador. Las órdenes de movimiento se mandan a este dispositivo el cual tiene un controlador que se encarga de procesar estas órdenes y hacer la distribución de energía en la tarjeta de alimentación para cada actuador.

## 2.6 Utillaje

El utillaje ha sido diseñado para poder colocar los sensores con la misma disposición en la que se encontraban durante la realización de las prácticas de la asignatura Cinemática y Dinámica de Máquinas. Se realiza un diseño simple pero efectivo, el cual permite fijarlo a la plataforma mediante 4 tornillos M4x50 mm y 4 tuercas, consiguiendo gracias a la disposición de las 4 roscas de la plataforma que el utillaje sea autocentrante y queden los sensores alineados con el sistema de ejes de la plataforma. Además, para poder fijar los dos sensores con tuercas, se diseñan dos plataformas elevadas respecto a la base. Para colocar la IMU se emplean 2 tornillos M3x20 mm con 2 tuercas y, para el inclinómetro 3 tornillos M4x25 mm con 3 tuercas. Por otro lado, las dimensiones de los sensores han sido extraídas de los manuales de los mismos [13] y [12].

El utillaje ha sido diseñado en Solid Edge 2022 para fabricado por fabricación aditiva, mediante FDM. Dado que se fija tanto el utillaje como los sensores con tuercas, los agujeros se han sobredimensionado 4 milímetros con respecto a las dimensiones de los manuales de los sensores y el diámetro medido de los agujeros de la plataforma, para así facilitar el montaje, pues la fijación se consigue gracias al apriete de las tuercas. El plano del modelo se encuentra en el Anexo C. Puede apreciarse el resultado y la disposición de los sensores en la Figura 2-8.

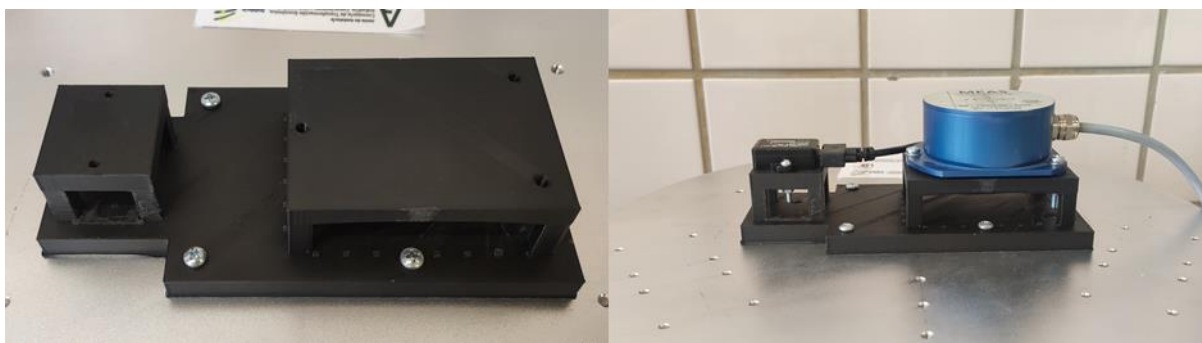


Figura 2-8 Utillaje

## 2.7 Estimación de los órdenes de magnitud de frecuencia y amplitud del ángulo de inclinación de una bicicleta

Es clave conocer las frecuencias típicas que se alcanzan en la inclinación durante el movimiento de una bicicleta para poder realizar correctamente este trabajo. Es fundamental distinguir en primera instancia entre trayectos en ciudad o carretera y trayectos campo a través, ya que, el terreno en estos últimos suele ser muy irregular, por lo que se alcanzan grandes inclinaciones. Para corresponder con las prácticas de Cinemática y Dinámica de Máquinas, este análisis se centra en trayectos en ciudad o carretera, donde la influencia del terreno es menor. Por otro lado, es importante diferenciar entre ciclistas profesionales y usuarios que tienen la bicicleta como afición o la utilizan para desplazarse. Igualmente, dentro de ambos grupos se pueden distinguir dos situaciones, *sprint* y velocidad constante.

En el caso de los ciclistas profesionales, es lógico que las frecuencias que alcanza la oscilación de la inclinación cuando realizan un *sprint* son mayores que en el caso de aficionados. Siguiendo con el *sprint*, a simple vista se observan altas amplitudes y frecuencias algo mayores que a velocidad constante. Para realizar una estimación de los órdenes de magnitud, se han consultado vídeos de ciclistas profesionales en competición. En cuanto a las frecuencias, tomando de los diferentes vídeos intervalos de 10 segundos se observan entre 16 y 20 ciclos completos, por lo que el orden de las frecuencias en *sprint* es de unos 2 Hz. Para estimar la amplitud, se ha tomado una altura del cuadro aproximada de 800 milímetros, pues es un valor intermedio entre las diferentes configuraciones que hay en el mercado.



Figura 2-9 Inclinación de una bicicleta durante un *sprint*

En este caso, el análisis se centra en la bicicleta que presenta un mayor ángulo de alabeo. Con las consideraciones anteriores y sabiendo que el ancho de las líneas de carreteras es de 100 milímetros, la proyección del cuadro de la bicicleta sobre el asfalto, se encuentra aproximadamente a 200 milímetros del apoyo de la rueda. Por ello, puede estimarse la amplitud del alabeo como:

$$\Phi \sim \text{atan}\left(\frac{200}{800}\right) = 14.04^\circ$$

Las estimaciones a velocidad aproximadamente constante se han realizado de la misma forma. En estos casos y considerando trayectos con poca pendiente, la amplitud es del orden de  $2^\circ$  y las frecuencias suelen ser de alrededor de 1 Hz. Siendo en este caso los valores muy similares entre profesionales y aficionados.

Por otro lado, hay que considerar que, cuando un ciclista toma una curva, inclina la bicicleta, por lo que el ángulo de alabeo aumenta en gran medida.



# 3 VERIFICACIÓN DE POSICIONES Y MOVIMIENTOS

*La cosa más hermosa que podemos experimentar es el misterio. Es la fuente de toda arte y toda ciencia.*

*- Albert Einstein -*

Es importante verificar que las órdenes de movimiento dan como resultado el movimiento deseado. Para ello se lleva a cabo un procedimiento de verificación de movimientos que se basa en medir la longitud de los actuadores de la plataforma tras cada movimiento y comparar el desplazamiento con el teórico según el modelo de Simulon.

Previamente al diseño del utillaje, trabajando con un utillaje provisional, se realiza un ensayo de verificación, el cual consiste en emplear el inclinómetro, pues tiene una gran precisión en medidas estáticas, para verificar si el sistema ha llegado a la posición angular deseada.

## 3.1 Procedimiento de verificación previo al diseño del utillaje

Este fue el primer ensayo que se realizó tras haber ajustado los parámetros de funcionamiento de la plataforma Stewart. Este ensayo tenía dos objetivos, verificar los movimientos y sacar ideas para el diseño del utillaje. El utillaje que se empleó para este ensayo era provisional, consistiendo de una placa plana, sobre la que se encontraban los sensores, y dos piezas que hacían la función de apoyos sobre la plataforma móvil, verificando mediante el empleo de un nivel que la placa plana estuviera paralela a la plataforma móvil. La unión tanto entre la placa y las piezas y, las piezas con la plataforma, se realizó con cinta de doble cara, quedando el utillaje como puede verse en la Figura 3-1.

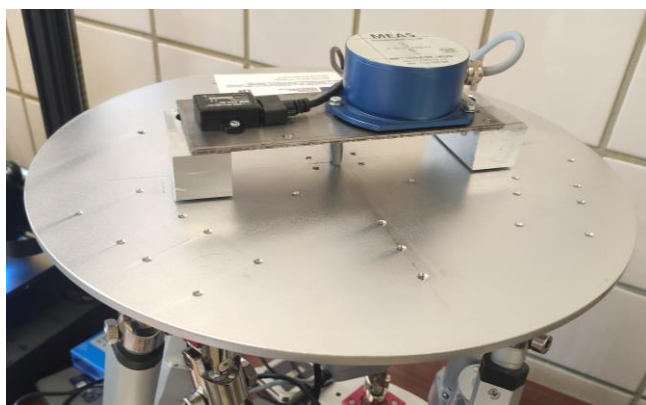


Figura 3-1 Utillaje provisional

Este procedimiento se centra exclusivamente en los ángulos de alabeo y cabeceo, pues con la configuración del utillaje son los que puede medir el inclinómetro. En este procedimiento se aprovecha la gran precisión del inclinómetro en medidas estáticas para comparar con la posición ordenada. En la Tabla 3-1 aparecen tanto las posiciones angulares, siempre alrededor del punto de referencia (0,0,433) milímetros, como las mediciones del inclinómetro.

Tabla 3-1 Resultados del procedimiento previo de calibración

<i>Roll</i> (°) deseado	<i>Pitch</i> (°) deseado	<i>Roll</i> (°) medido	<i>Pitch</i> (°) medido
-10,00	0,00	-9,62	-0,31
-8,00	0,00	-7,38	-0,29
-4,00	0,00	-3,62	-0,25
0,00	0,00	0,21	-0,24
4,00	0,00	4,2	-0,24
8,00	0,00	8,36	-0,19
10,00	0,00	10,31	-0,15
0,00	-10,00	0,11	-10,24
0,00	-8,00	0,13	-8,25
0,00	-4,00	0,18	-4,19
0,00	4,00	0,17	3,71
0,00	8,00	0,21	7,61
0,00	10,00	0,22	9,24

Se observa que no se consiguen las posiciones deseadas. En la Tabla 3-2, se puede ver el error relativo de posicionamiento del sistema para el alabeo y el cabeceo.

Tabla 3-2 Error relativo en posicionamiento

<i>Roll</i> (°) deseado	Error relativo (%)	<i>Pitch</i> (°) deseado	Error relativo (%)
-10,00	3,80	-10,00	2,40
-8,00	7,75	-8,00	3,13
-4,00	9,50	-4,00	4,75
4,00	5,50	4,00	7,25
8,00	4,50	8,00	4,88
10,00	3,10	10,00	7,60

Se aprecia que el error máximo es del orden del 10% con un error medio del 5,35%. Esto implica que los movimientos carezcan de la precisión requerida para lograr el alcance y los objetivos definidos en un principio, en la sección 3.2 se comprueba si es cuestión del utillaje provisional o de la plataforma.

Por otro lado, se comprobó con este ensayo el correcto alineamiento de los sensores con respecto a los ejes de la plataforma Stewart, algo que se empleó para el diseño del utillaje.

### 3.2 Procedimiento de verificación de movimiento

Este procedimiento está basado en la comparación de las variaciones en las longitudes de los actuadores tras realizar un movimiento respecto a un punto de referencia u origen, comparando entonces los resultados medidos durante los ensayos frente a los obtenidos a partir del modelo teórico. Se define el punto de referencia respecto al sistema global como el (0,0,433,0,0,0) en milímetros y grados. Los puntos de calibración se muestran en la Tabla 3-3, en milímetros y grados. Para poder realizar una comparación entre los resultados medidos frente a los teóricos, se calculan los desplazamientos relativos del extremo del vástago como la diferencia entre la longitud en ese punto de calibración frente a la longitud en el punto de referencia, tanto para los valores experimentales como para los teóricos. Esto se tiene que hacer así pues experimentalmente se miden las longitudes de los vástagos y del modelo teórico se obtienen las distancias entre los centros de las juntas cardan.



En este análisis no se ha tenido en cuenta la guiñada pues este movimiento no es de interés para este trabajo ya que no es medido por el inclinómetro.

Tabla 3-3 Puntos de calibración

Puntos de calibración	$X$ (mm)	$Y$ (mm)	$Z$ (mm)	$Roll$ (°)	$Pitch$ (°)	$Yaw$ (°)
1	0	0	433	0	1	0
2	0	0	433	0	2	0
3	0	0	433	0	4	0
4	0	0	433	0	6	0
5	0	0	433	0	8	0
6	0	0	433	0	10	0
7	0	0	433	0	12	0
8	0	0	433	0	14	0
9	0	0	433	1	0	0
10	0	0	433	2	0	0
11	0	0	433	4	0	0
12	0	0	433	6	0	0
13	0	0	433	8	0	0
14	0	0	433	10	0	0
15	0	0	433	12	0	0
16	0	0	433	14	0	0
17	10	0	433	0	0	0
18	25	0	433	0	0	0
19	50	0	433	0	0	0
20	75	0	433	0	0	0
21	90	0	433	0	0	0
22	0	10	433	0	0	0
23	0	25	433	0	0	0
24	0	50	433	0	0	0
25	0	75	433	0	0	0
26	0	90	433	0	0	0
27	0	0	390	0	0	0
28	0	0	400	0	0	0
29	0	0	420	0	0	0
30	0	0	440	0	0	0
31	0	0	450	0	0	0
32	0	0	475	0	0	0

Para llevar a cabo este procedimiento se emplea un script de Matlab dividido en tres bloques que puede verse en el Anexo A.7. El primero de ellos consiste en cargar una serie de parámetros, establecer la comunicación con la plataforma, encender motores, comprobar su funcionamiento, establecer los parámetros de operación de la máquina y, llevar la plataforma móvil al punto de referencia. Tras esto, el segundo bloque consiste en la ejecución de una orden de movimiento simple para llevar a la plataforma a los diferentes puntos de calibración.

Una vez aquí, se mantiene a la plataforma en ese punto y se realizan las mediciones de las longitudes de los vástagos de cada uno de los seis actuadores, siguiendo la numeración de la Figura 2-3. Las mediciones se realizan con un pie de rey, tomando como referencia el final de la cubierta del actuador y la conexión con la junta cardan como puede verse en la Figura 3-2.



Figura 3-2 Ejemplo de cómo se han realizado las mediciones

Con las medidas realizadas, se modifica el segundo bloque para llevar a la plataforma al siguiente punto de calibración y repetir el proceso hasta completar todos los puntos de calibración. Con esto, se pasa al último bloque el cual consiste en llevar la plataforma a un punto de cota cercana al punto más bajo, en este caso se lleva a  $z = 390 \text{ mm}$ , apagar motores y finalizar la conexión con la plataforma. Los resultados de las mediciones en milímetros se presentan en la Tabla 3-4.

Tabla 3-4 Longitud medida del vástago de cada actuador

Puntos de calibración	1 (mm)	2 (mm)	3 (mm)	4 (mm)	5 (mm)	6 (mm)
1	59,72	56,25	58,70	58,80	55,30	56,21
2	58,61	55,37	60,91	61,07	54,48	54,99
3	55,89	53,42	65,56	65,66	52,91	52,75
4	53,58	51,88	70,08	70,18	51,43	50,82
5	51,13	50,46	74,80	74,91	49,91	48,62
6	48,81	48,84	79,36	79,55	48,85	46,65
7	47,16	47,31	84,04	84,08	46,72	45,12
8	45,33	46,02	88,87	89,00	46,42	43,40
9	62,83	59,12	56,70	56,34	54,00	55,81
10	64,73	61,33	57,00	56,14	52,00	53,75
11	68,43	65,34	57,54	55,82	47,90	50,17
12	72,09	69,18	58,24	55,60	44,18	46,80
13	75,80	74,21	59,01	55,41	40,69	43,30

Puntos de calibración	1 (mm)	2 (mm)	3 (mm)	4 (mm)	5 (mm)	6 (mm)
14	79,73	78,38	59,83	55,26	37,14	40,10
15	83,50	83,09	60,69	55,36	33,95	36,83
16	87,36	87,60	61,72	55,44	30,80	33,75
17	58,85	59,81	56,41	56,50	58,56	55,18
18	56,11	64,21	56,89	56,96	63,15	52,38
19	53,24	73,50	59,11	59,12	72,08	49,22
20	52,50	83,98	63,41	63,51	82,73	48,44
21	53,34	90,77	66,94	66,97	89,42	48,50
22	62,61	56,10	54,07	59,57	57,52	56,02
23	65,60	55,11	50,39	64,50	59,80	54,22
24	71,80	54,96	46,92	73,96	65,70	53,28
25	81,30	56,53	44,60	85,13	73,08	54,19
26	85,92	58,31	44,47	92,43	78,36	55,61
27	19,70	14,68	14,22	14,50	13,55	15,10
28	29,50	24,71	24,14	24,37	23,38	24,96
29	49,36	44,43	43,95	43,91	42,97	44,45
30	68,00	64,12	63,50	63,50	63,04	64,24
31	78,72	74,08	73,30	73,32	72,70	74,07
32	103,39	98,53	98,00	98,15	97,09	98,77

Con las mediciones ya realizadas, se procede a calcular la variación de la longitud del vástago respecto al punto de referencia como la diferencia entre las longitudes medidas de los actuadores para cada punto de calibración y las longitudes en el punto de referencia, las cuales se muestran en milímetros en la Tabla 3-5.

Tabla 3-5 Longitud medida de los vástagos en el punto de referencia

Punto de referencia	1 (mm)	2 (mm)	3 (mm)	4 (mm)	5 (mm)	6 (mm)
	61,00	57,12	56,46	56,55	56,10	57,45

Los desplazamientos, medidos en milímetros, de cada vástago se presentan en la Tabla 3-6. En ella puede verse como la mayoría de los desplazamientos presentan valores similares por parejas. Por ejemplo, en los primeros seis puntos de calibración, los desplazamientos son similares en las parejas de actuadores 1-6, 2-5 y 3-4.

Tabla 3-6 Desplazamiento medido de cada vástago

Puntos de calibración	1 (mm)	2 (mm)	3 (mm)	4 (mm)	5 (mm)	6 (mm)
1	-1,28	-0,87	2,24	2,25	-0,80	-1,24
2	-2,39	-1,75	4,45	4,52	-1,62	-2,46
3	-5,11	-3,70	9,10	9,11	-3,19	-4,70
4	-7,42	-5,24	13,62	13,63	-4,67	-6,63
5	-9,87	-6,66	18,34	18,36	-6,19	-8,83
6	-12,19	-8,28	22,90	23,00	-7,25	-10,80
7	-13,84	-9,81	27,58	27,53	-9,38	-12,33
8	-15,67	-11,10	32,41	32,45	-9,68	-14,05

Puntos de calibración	1 (mm)	2 (mm)	3 (mm)	4 (mm)	5 (mm)	6 (mm)
9	1,83	2,00	0,24	-0,21	-2,10	-1,64
10	3,73	4,21	0,54	-0,41	-4,10	-3,70
11	7,43	8,22	1,08	-0,73	-8,20	-7,28
12	11,09	12,06	1,78	-0,95	-11,92	-10,65
13	14,80	17,09	2,55	-1,14	-15,41	-14,15
14	18,73	21,26	3,37	-1,29	-18,96	-17,35
15	22,50	25,97	4,23	-1,19	-22,15	-20,62
16	26,36	30,48	5,26	-1,11	-25,30	-23,70
17	-2,15	2,69	-0,05	-0,05	2,46	-2,27
18	-4,89	7,09	0,43	0,41	7,05	-5,07
19	-7,76	16,38	2,65	2,57	15,98	-8,23
20	-8,50	26,86	6,95	6,96	26,63	-9,01
21	-7,66	33,65	10,48	10,42	33,32	-8,95
22	1,61	-1,02	-2,39	3,02	1,42	-1,43
23	4,60	-2,01	-6,07	7,95	3,70	-3,03
24	10,80	-2,16	-9,54	17,41	9,60	-4,17
25	20,30	-0,59	-11,86	28,58	16,98	-3,26
26	24,92	1,19	-11,99	35,88	22,26	-1,84
27	-41,30	-42,44	-42,24	-42,05	-42,55	-42,35
28	-31,50	-32,41	-32,32	-32,18	-32,72	-32,49
29	-11,64	-12,69	-12,51	-12,64	-13,13	-13,00
30	7,00	7,00	7,04	6,95	6,94	6,79
31	17,72	16,96	16,84	16,77	16,60	16,62
32	42,39	41,41	41,54	41,60	40,99	41,32

Por otro lado, se lleva a cabo un procedimiento similar con el modelo teórico de la plataforma en Simulon. Mediante las restricciones de movilidad, Anexo B.2, se lleva a la plataforma a los puntos de calibración. Una vez aquí, se resuelve la cinemática de la plataforma con el código del Anexo B.7. Dentro de este código, se llama a la función “*LongitudBarras*”, la cual puede verse en el Anexo B.6, que calcula y muestra por pantalla la longitud de cada actuador. La longitud, en milímetros, de los actuadores en el punto de referencia según el modelo teórico puede verse en la Tabla 3-7.

Tabla 3-7 Longitud teórica de los actuadores en el punto de referencia

Punto de referencia	1 (mm)	2 (mm)	3 (mm)	4 (mm)	5 (mm)	6 (mm)
	340,79	340,79	340,79	340,79	340,79	340,79

La longitud en milímetros de cada actuador según el modelo teórico en los puntos de calibración se muestra en la Tabla 3-8.

Tabla 3-8 Longitud teórica de los actuadores

Puntos de calibración	1 (mm)	2 (mm)	3 (mm)	4 (mm)	5 (mm)	6 (mm)
1	339,53	339,92	342,95	342,95	339,92	339,53

Puntos de calibración	1 (mm)	2 (mm)	3 (mm)	4 (mm)	5 (mm)	6 (mm)
2	338,30	339,06	345,13	345,12	339,07	338,30
3	335,92	337,39	349,52	349,52	337,40	335,91
4	333,65	335,79	353,98	353,97	335,79	333,64
5	331,49	334,24	358,48	358,48	334,25	331,49
6	329,47	332,77	363,02	363,02	332,77	329,46
7	327,57	331,36	367,60	367,59	331,36	327,56
8	325,80	330,03	372,20	372,20	330,03	325,80
9	342,55	342,78	341,03	340,57	338,83	339,05
10	344,31	344,79	341,29	340,38	336,89	337,32
11	347,88	348,88	341,87	340,04	333,09	333,91
12	351,48	353,05	342,52	339,79	329,41	330,55
13	355,11	357,29	343,26	339,63	325,85	327,27
14	358,78	361,60	344,07	339,55	322,42	324,05
15	362,46	365,97	344,96	339,55	319,13	320,92
16	366,17	370,39	345,93	339,64	315,99	317,88
17	338,69	343,37	340,74	340,74	343,37	338,69
18	336,07	347,73	341,22	341,22	347,73	336,07
19	333,17	356,30	343,47	343,46	356,30	333,17
20	332,13	366,37	347,50	347,50	366,37	332,12
21	332,40	373,09	350,76	350,76	373,09	332,40
22	342,46	339,76	338,23	343,63	342,12	339,41
23	345,48	338,75	334,90	348,38	344,64	337,89
24	351,89	338,55	330,81	357,55	350,23	336,83
25	359,92	340,19	328,56	368,20	357,49	337,62
26	365,48	342,05	328,13	375,25	362,60	338,98
27	299,63	299,62	299,63	299,62	299,63	299,62
28	309,16	309,15	309,16	309,15	309,16	309,15
29	328,30	328,30	328,30	328,30	328,30	328,30
30	347,54	347,54	347,54	347,54	347,54	347,54
31	357,19	357,18	357,19	357,18	357,19	357,18
32	381,39	381,39	381,39	381,39	381,39	381,39

Con los datos de la Tabla 3-8 y de la Tabla 3-7, pueden calcularse los desplazamientos de cada actuador, los cuales se presentan en la Tabla 3-9.

Tabla 3-9 Desplazamiento teórico de los actuadores

Puntos de calibración	1 (mm)	2 (mm)	3 (mm)	4 (mm)	5 (mm)	6 (mm)
1	-1,26	-0,87	2,16	2,16	-0,87	-1,26
2	-2,49	-1,73	4,34	4,33	-1,72	-2,49

Puntos de calibración	1 (mm)	2 (mm)	3 (mm)	4 (mm)	5 (mm)	6 (mm)
3	-4,87	-3,40	8,73	8,72	-3,39	-4,88
4	-7,14	-5,00	13,19	13,18	-5,00	-7,15
5	-9,30	-6,55	17,69	17,69	-6,54	-9,30
6	-11,32	-8,02	22,23	22,23	-8,02	-11,33
7	-13,22	-9,43	26,81	26,80	-9,43	-13,23
8	-14,99	-10,76	31,41	31,41	-10,76	-14,99
9	1,76	1,99	0,24	-0,22	-1,96	-1,74
10	3,52	4,00	0,50	-0,41	-3,90	-3,47
11	7,09	8,09	1,08	-0,75	-7,70	-6,88
12	10,69	12,26	1,73	-1,00	-11,38	-10,24
13	14,32	16,50	2,47	-1,16	-14,94	-13,52
14	17,99	20,81	3,28	-1,24	-18,37	-16,74
15	21,67	25,18	4,17	-1,24	-21,66	-19,87
16	25,38	29,60	5,14	-1,15	-24,80	-22,91
17	-2,10	2,58	-0,05	-0,05	2,58	-2,10
18	-4,72	6,94	0,43	0,43	6,94	-4,72
19	-7,62	15,51	2,68	2,67	15,51	-7,62
20	-8,66	25,58	6,71	6,71	25,58	-8,67
21	-8,39	32,30	9,97	9,97	32,30	-8,39
22	1,67	-1,03	-2,56	2,84	1,33	-1,38
23	4,69	-2,04	-5,89	7,59	3,85	-2,90
24	11,10	-2,24	-9,98	16,76	9,44	-3,96
25	19,13	-0,60	-12,23	27,41	16,70	-3,17
26	24,69	1,26	-12,66	34,46	21,81	-1,81
27	-41,16	-41,17	-41,16	-41,17	-41,16	-41,17
28	-31,63	-31,64	-31,63	-31,64	-31,63	-31,64
29	-12,49	-12,49	-12,49	-12,49	-12,49	-12,49
30	6,75	6,75	6,75	6,75	6,75	6,75
31	16,40	16,39	16,40	16,39	16,40	16,39
32	40,60	40,60	40,60	40,60	40,60	40,60

Puede observarse como los resultados de los desplazamientos relativos obtenidos con las mediciones experimentales, Tabla 3-6, no son muy similares en todos los puntos a los teóricos, Tabla 3-9. Para facilitar la comparación se calcula el cociente entre los desplazamientos medidos y los calculados con el modelo teórico. Los resultados se muestran tanto en la Tabla 3-10 como gráficamente. A la hora de representar gráficamente, se ha considerado interesante representar cómo varía este cociente con respecto a cada coordenada. Por lo tanto, los resultados se presentan gráficamente en la Figura 3-3, Figura 3-4, Figura 3-5, Figura 3-6 y Figura 3-7.

Tabla 3-10 Relación desplazamiento medido entre desplazamiento teórico

Puntos de calibración	1	2	3	4	5	6
1	1,02	1,00	1,04	1,04	0,92	0,98

Puntos de calibración	1	2	3	4	5	6
2	0,96	1,01	1,03	1,04	0,94	0,99
3	1,05	1,09	1,04	1,04	0,94	0,96
4	1,04	1,05	1,03	1,03	0,93	0,93
5	1,06	1,02	1,04	1,04	0,95	0,95
6	1,08	1,03	1,03	1,03	0,90	0,95
7	1,05	1,04	1,03	1,03	0,99	0,93
8	1,05	1,03	1,03	1,03	0,90	0,94
9	1,04	1,01	1,00	0,95	1,07	0,94
10	1,06	1,05	1,08	1,00	1,05	1,07
11	1,05	1,02	1,00	0,97	1,06	1,06
12	1,04	0,98	1,03	0,95	1,05	1,04
13	1,03	1,04	1,03	0,98	1,03	1,05
14	1,04	1,02	1,03	1,04	1,03	1,04
15	1,04	1,03	1,01	0,96	1,02	1,04
16	1,04	1,03	1,02	0,97	1,02	1,03
17	1,02	1,04	1,00	1,00	0,95	1,08
18	1,04	1,02	1,00	0,95	1,02	1,07
19	1,02	1,06	0,99	0,96	1,03	1,08
20	0,98	1,05	1,04	1,04	1,04	1,04
21	0,91	1,04	1,05	1,05	1,03	1,07
22	0,96	0,99	0,93	1,06	1,07	1,04
23	0,98	0,99	1,03	1,05	0,96	1,04
24	0,97	0,96	0,96	1,04	1,02	1,05
25	1,06	0,98	0,97	1,04	1,02	1,03
26	1,01	0,94	0,95	1,04	1,02	1,02
27	1,00	1,03	1,03	1,02	1,03	1,03
28	1,00	1,02	1,02	1,02	1,03	1,03
29	0,93	1,02	1,00	1,01	1,05	1,04
30	1,04	1,04	1,04	1,03	1,03	1,01
31	1,08	1,03	1,03	1,02	1,01	1,01
32	1,04	1,02	1,02	1,02	1,01	1,02

Como puede verse, el comportamiento de los actuadores presenta un error relativo de hasta un 10%, con una media de 3,67%, tal y como se esperaba tras los resultados de la sección 3.1. Esto hace que los movimientos no sean tan precisos como se necesita para el objetivo de este trabajo.

Por otro lado, gracias a representar este cociente frente a cada coordenada, es posible ver el comportamiento de la plataforma según varía cada coordenada. Esto se vuelve más importante cuando sobre la plataforma hay una carga, pues al alejarse del centro aparecen momentos que antes no estaban, aunque no es objeto de este trabajo por lo que no se ensaya con cargas sobre la plataforma.

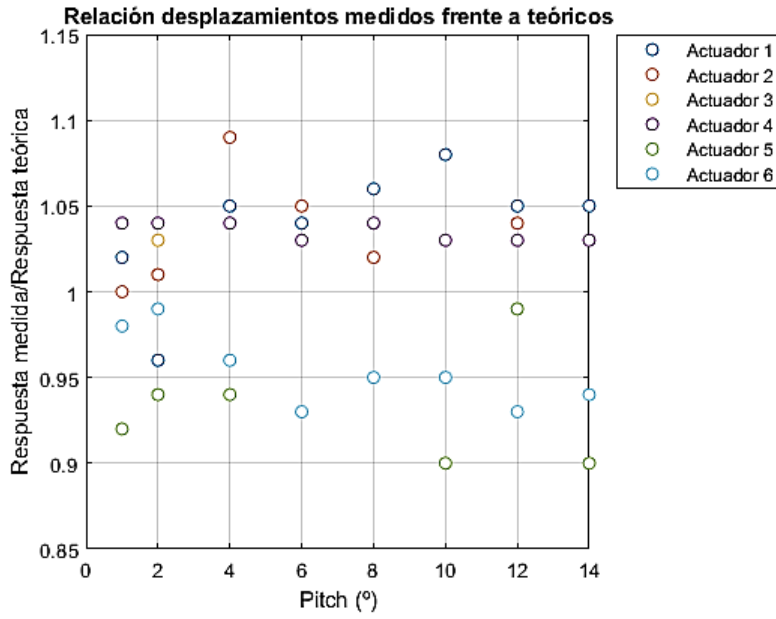


Figura 3-3 Relación entre los desplazamientos medidos y los teóricos cuando aumenta el ángulo de cabeceo

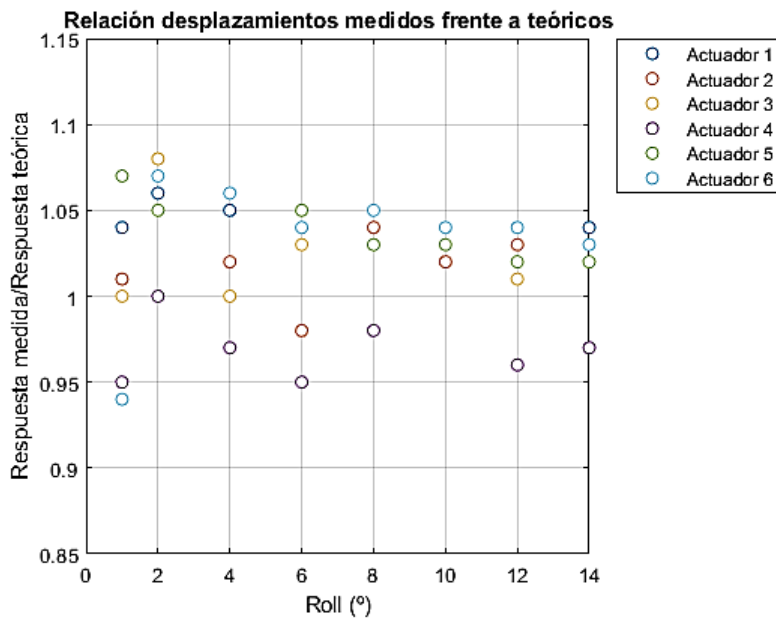


Figura 3-4 Relación entre los desplazamientos medidos y los teóricos cuando aumenta el ángulo de alabeo



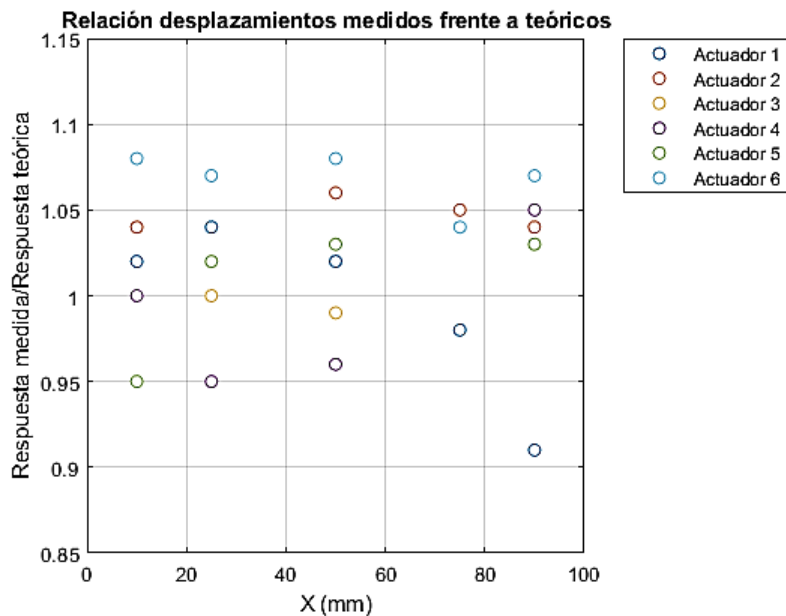


Figura 3-5 Relación entre los desplazamientos medidos y los teóricos cuando aumenta la coordenada X

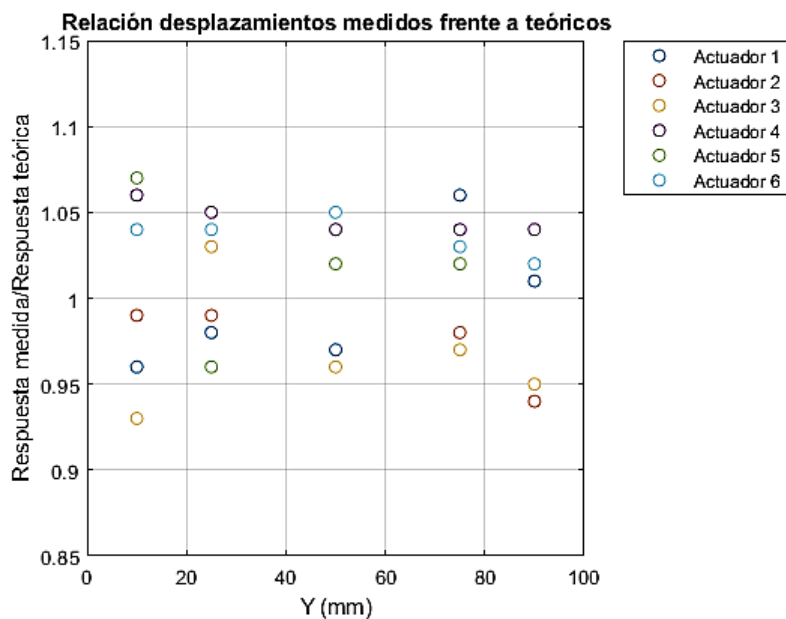


Figura 3-6 Relación entre los desplazamientos medidos y los teóricos cuando aumenta la coordenada Y

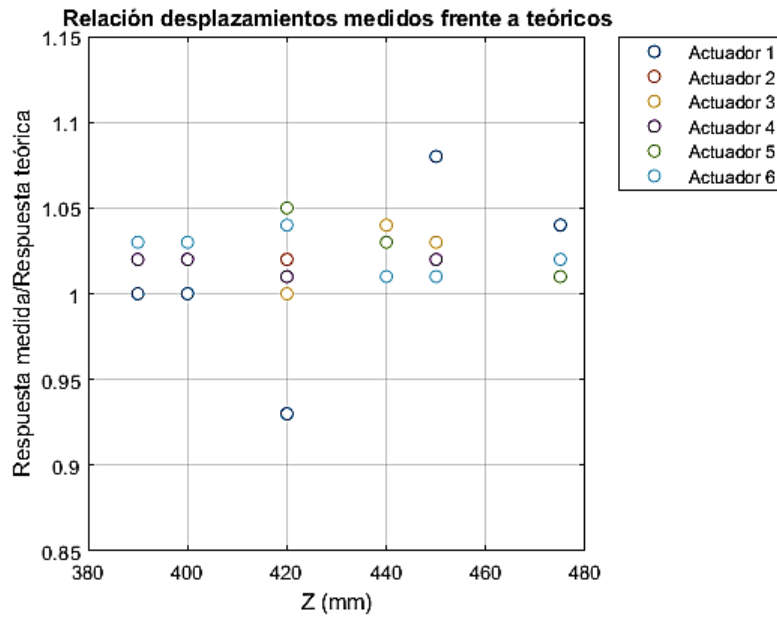


Figura 3-7 Relación entre los desplazamientos medidos y los teóricos cuando aumenta la coordenada Z

Observando los resultados se puede apreciar que este cociente presenta un comportamiento que varía durante todo el rango de funcionamiento de cada variable. Igualmente, el comportamiento de cada actuador difiere entre movimientos para diferentes variables. De nuevo, se ve como los valores máximos y mínimos de este cociente son respectivamente 1,1 y 0,9. También se aprecia que, en la mayoría de los casos, el desplazamiento de los actuadores medido es mayor que el teórico, por lo que generalmente se obtienen desplazamientos mayores que los deseados.

Por otro lado, no se observa una tendencia general en las desviaciones de los movimientos que permitiese corregir las órdenes de movimientos de forma sistemática para lograr así llegar a los puntos deseados, pues para variaciones de una misma coordenada el comportamiento de cada actuador difiere en cada punto de calibración, sin presentar una tendencia clara.

# 4 CARACTERIZACIÓN DEL INCLINÓMETRO

---

*El conocimiento es poder en sí mismo.*

*- Francis Bacon -*

En este capítulo se trata de llevar a cabo la caracterización de las mediciones dinámicas del inclinómetro. Para ello, se deben de separar lo máximo posible sus dos mediciones, es decir, cuando varía el ángulo alrededor de  $X$ , el ángulo alrededor de  $Y$  debe permanecer constante para así separar sus influencias. Para llevar a cabo esto, se obliga a la plataforma a moverse según una función seno exclusivamente en uno de los ángulos.

Para una correcta caracterización del inclinómetro es importante analizar diferentes frecuencias y amplitudes. Para ello, se debe ir aumentando la frecuencia hasta que las mediciones que realiza el inclinómetro no se correspondan con los ángulos reales, dado que el trabajo parte con la duda de si el inclinómetro realiza correctamente las mediciones dinámicas y repetir estos ensayos para diferentes amplitudes.

Como ya se ha mencionado, se analizan por separados los dos ángulos que el inclinómetro puede medir. Se comienza por el ángulo alrededor de  $X$ . Para ello se controla la plataforma con un control de trayectoria, obligándola a variar el ángulo que forma alrededor del eje  $X$  siguiendo una función seno. Esto se realiza con el script de Matlab del Anexo A.4.

Este script se divide en tres bloques, el primero de ellos, es el bloque en el cual se establece la comunicación con la plataforma, se comprueba el estado de los motores, se fijan sus parámetros de funcionamiento y la plataforma móvil es llevada al punto de referencia. En el segundo de los bloques, se establecen la frecuencia y la amplitud del seno, el cual es mandando como orden de movimiento en el ángulo de alabeo. Para ello, mediante un bucle, se van cargando las órdenes de movimiento a la plataforma por control de trayectoria, de forma que, la orden de movimiento es “*move\_platform\_traj,0,0,433, $\varphi$ ,0,0,0.01*”, donde  $\varphi$  es el valor del seno por la amplitud en ese instante. Es durante este bloque, haciendo uso del programa de LabVIEW usado durante las prácticas de Cinemática y Dinámica de Máquinas del que se han eliminado el resto de sensores, cuando se graban los datos registrados por el inclinómetro para posteriormente ser tratados. El último bloque consiste en llevar a la plataforma a su punto más bajo posible, desconectar los motores y finalizar la comunicación.

Este procedimiento se debe repetir para diferentes frecuencias y amplitudes para así tener varios registros. El análisis consiste en la comparación para diferentes frecuencias de la relación entre la amplitud del *input* y la amplitud obtenida a partir del registro del inclinómetro, así como de la relación entre las frecuencias teóricas y calculada a partir del registro.

Para calcular la frecuencia a partir del registro, se representa gráficamente y se cuenta el número de cortes con el eje del tiempo. Con esto pueden contarse el número de ciclos totales como el número de puntos de corte entre dos, y dado que la duración de la señal sinusoidal, está fijada, puede calcularse fácilmente la frecuencia medida.

Para los ángulos alrededor del eje  $Y$ , el procedimiento es similar, pero en este caso se obliga a la plataforma a variar dicho ángulo según una función seno, por lo que la orden de movimiento queda “*move\_platform\_traj,0,0,433,0, $\varphi$ ,0,0.01*”, donde  $\varphi$  es el valor del seno por la amplitud en ese instante.

Es durante la realización estos ensayos cuando la plataforma empieza a fallar, por lo que los datos adquiridos hasta el momento del fallo (mediciones alrededor del eje  $X$  con una amplitud de  $2^\circ$  y frecuencias desde 0,2 Hz hasta 1 Hz en intervalos de 0,2 Hz), no tienen la fiabilidad para realizar su análisis. El primer fallo que se produjo fue que, durante el movimiento de la plataforma siguiendo la función seno, a los aproximadamente 7 segundos, los actuadores 1, 4 y 5 sufrían bloqueos, por lo que los movimientos no se correspondían con lo deseado. Este fallo, ocurría repetidamente durante todos los ensayos siguientes al primero en el que falló. Por ello, se volvió a

restaurar la configuración de la plataforma, así como, se volvieron a instalar todos los *drivers* de la misma.

Dado que el sistema seguía fallando con estos ensayos, se probó a volver a controlar la plataforma con el *joystick* pues, así no había fallado hasta el momento. Fue entonces, cuando en uno de los movimientos que requería que el actuador 1 estuviese aproximadamente en su máxima extensión, sufrió un bloqueo y dejó de moverse mientras que el resto sí lo hacían. Para poder determinar si el fallo era del motor o de los controladores, se excitó el motor con una batería externa y sí se movió el actuador. Por otro lado, consultando desde la interfaz de usuario de la plataforma la extensión de cada actuador, se veía que claramente la extensión que el sistema detectaba del actuador 1 no se correspondía con la real.

# 5 CONCLUSIONES

---

*El placer más noble es el júbilo de comprender.*

*- Leonardo Da Vinci -*

Como ha podido observarse a lo largo de este trabajo, las limitaciones de la plataforma Stewart con la que se ha trabajado han impedido lograr las expectativas y el alcance definido previo al comienzo de este trabajo. En primer lugar, la baja precisión de la plataforma daba lugar a que los movimientos no se correspondiesen con los reales. Otra limitación de la plataforma, era su velocidad de respuesta, ya que, durante algunos ensayos, movimientos ordenados para realizarse en un tiempo determinado, se realizaban en un tiempo entorno a un 10% mayor. Por otro lado, los fallos en los actuadores, que sufrían bloqueos a los segundos de ejecución de movimientos algo más rápidos, lo que ocurrió durante los ensayos de caracterización del inclinómetro al aumentar la frecuencia del seno, han acabado provocado que los movimientos no fueran los esperados. Todo esto ha hecho que este trabajo finalice con las limitaciones de la misma y no haya sido posible optimizar el filtro de Kalman para la estimación del ángulo de inclinación de una bicicleta a partir de las mediciones de la IMU ni determinar si las mediciones dinámicas del inclinómetro son correctas y, poder resolver correctamente la cinemática de la bicicleta en las prácticas de Cinemática y Dinámica de Máquinas.

## 5.1 Posibles causas de la baja precisión de la plataforma Stewart

En primer lugar, señalar que las juntas cardan están colocadas sobre unos rodamientos, esto introduce un grado de libertad que no aporta nada al movimiento de la plataforma. Esto puede llevar al desalineamiento de los actuadores, provocando así que los movimientos de la plataforma no sean precisos en esos instantes.

Por otro lado, el sistema de control de la plataforma, combina las mediciones de una IMU y un inclinómetro con las lecturas de los *encoders* de los motores de los actuadores para determinar la posición de la plataforma en cada instante. Por lo que es posible que, algún fallo en estos sensores haya provocado una pérdida de precisión en los movimientos, ya que, como se ha señalado en el capítulo 4, tras el fallo en la plataforma, la lectura realizada por los sistemas de la plataforma de la extensión del actuador nº 1, no se correspondía con la real.

Por último, el control de los movimientos por trayectorias, interpola la trayectoria entre el punto actual y el punto de la orden de movimiento en un polinomio de tercer grado, siendo esta interpolación desconocida para el usuario. Por ello, esto junto a la baja precisión obtenida invita a plantearse que la interpolación tenga un error relativo del mismo orden que los obtenidos en la sección 3.2.

Por otro lado, la baja repetibilidad en orientación de la plataforma según el fabricante, da lugar a variaciones de los ángulos que no son las correctas incluso para movimientos que no requieren variación de ese ángulo.

Quizás, el gran error relativo que presenta la plataforma llegando al 10% en algunos casos, pueda deberse a una combinación de las causas anteriores. Esto puede ser así, pues hay movimientos que presentan errores relativos bajos mientras en otras ocasiones se alcanza el error máximo del 10%, por lo que, cada una de las causas pueden aportar error y combinarse estos errores en algunos instantes dando lugar a los errores máximos que se han visto durante el desarrollo del trabajo.

## 5.2 Líneas de investigación futuras

Dado que el trabajo concluye debido a la baja precisión de la plataforma, el principal trabajo a llevar a cabo en

un futuro debe de consistir en primera instancia en recalibrarla para tratar de aumentar su precisión. Para ello, lo primero es desarrollar un sistema capaz de determinar la posición y la orientación de la plataforma móvil con precisión en cada instante. Para poder lograrlo existen diversas maneras de conseguirlo como pueden ser el empleo de sensores láser que incidan en todo momento sobre la plataforma móvil o bien mediante sensores que determinen la extensión de cada actuador en cada instante.

Empleando este tipo de sensores junto con el modelo teórico de la plataforma, la posición queda determinada en todo instante. Por lo tanto, sería de interés desarrollar un controlador para el sistema combinando estos sensores con el modelo teórico.

Por otro lado, al desarrollar un controlador nuevo, se conocerían sus características, ya que, trabajando con el del fabricante no es posible, por lo que, por ejemplo, como se ha mencionado en la sección 5.1, no se conoce cómo realiza la interpolación de las trayectorias. Igualmente, al desarrollar el propio controlador, sería posible mejorar el sistema para controlar los movimientos a través de las extensiones o retracciones de los actuadores, pues con el sistema del fabricante, los movimientos generados con este tipo de control, no eran suaves y acababan dando lugar a posiciones intermedias entre las deseadas que provocaban que el movimiento no fuera el ordenado.

Para ver la viabilidad del empleo de sensores láser, se estudia a continuación el sistema de ecuaciones para determinar la posición y orientación de la plataforma. Pues para el caso de realizar mediciones de la extensión de los actuadores es fácil de ver que se puede obtener la posición del centro de la plataforma en todo instante.

En primer lugar, señalar que se sigue el modelo de la sección 2.3, al que se le añaden los sensores láseres como puede verse en la Figura 5-1.

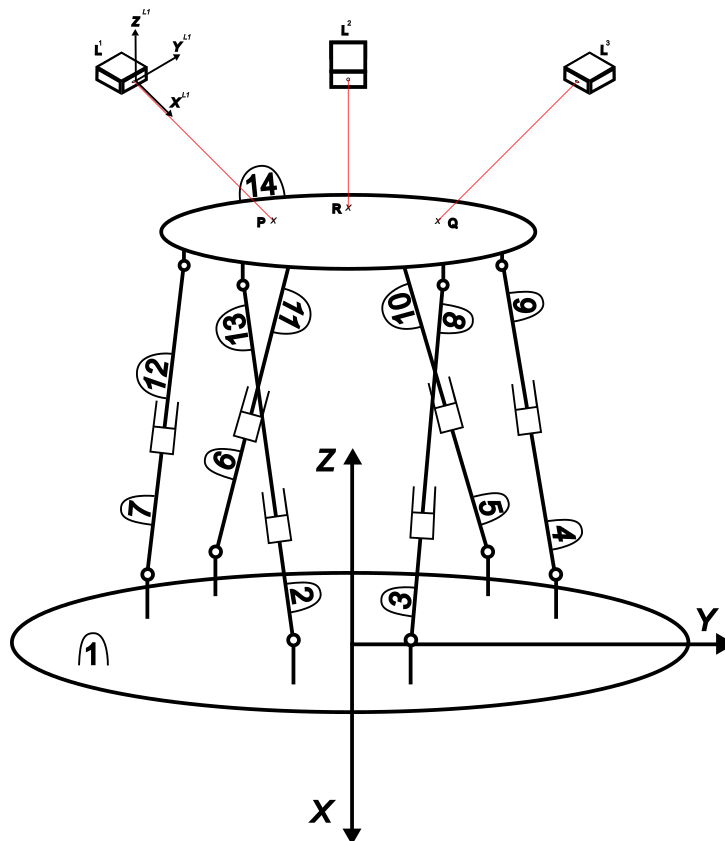


Figura 5-1 Modelo de la plataforma Stewart con sensores láser

La posición de los sensores láseres en el sistema de referencia global, es decir, en el sistema de referencia asociado a la base fija de la plataforma, es conocida y constante en todo instante y, se denotan como:  $r^{L1}$ ,  $r^{L2}$  y  $r^{L3}$  haciendo referencia a cada uno de los tres sensores láser. Por otro lado, el sistema de referencia local de cada sensor láser se puede apreciar en la Figura 5-1, por lo que, el vector en coordenadas locales de cada láser que da la posición del punto sobre el que inciden es:

$$\begin{aligned}\bar{\mathbf{u}}_P^{L1} &= [l_1 \quad 0 \quad 0]^T \\ \bar{\mathbf{u}}_R^{L2} &= [l_2 \quad 0 \quad 0]^T \\ \bar{\mathbf{u}}_Q^{L3} &= [l_3 \quad 0 \quad 0]^T\end{aligned}$$

Donde, P, R y Q son los tres puntos de incidencia de los sensores láser sobre el plano de la plataforma móvil y  $l_1$ ,  $l_2$  y  $l_3$  son las distancias medidas por cada sensor láser.

Con ello, se puede obtener la posición de los tres puntos P, R y Q en coordenadas globales de la siguiente forma:

$$\begin{aligned}\mathbf{r}_P &= \mathbf{r}^{L1} + \mathbf{A}^{L1}\bar{\mathbf{u}}_P^{L1} \\ \mathbf{r}_R &= \mathbf{r}^{L2} + \mathbf{A}^{L2}\bar{\mathbf{u}}_R^{L2} \\ \mathbf{r}_Q &= \mathbf{r}^{L3} + \mathbf{A}^{L3}\bar{\mathbf{u}}_Q^{L3}\end{aligned}$$

Conocida la posición de estos tres puntos y sabiendo que pertenecen al plano de la plataforma móvil, para determinar el vector normal a dicho plano,  $\mathbf{n}$ , se sigue el siguiente procedimiento:

$$\begin{aligned}\mathbf{r}_{PR} &= \mathbf{r}_R - \mathbf{r}_P \\ \mathbf{r}_{PQ} &= \mathbf{r}_Q - \mathbf{r}_P \\ \mathbf{n} &= \mathbf{r}_{PQ} \times \mathbf{r}_{PR}\end{aligned}$$

Una vez conocido el vector perpendicular al plano de la plataforma móvil,  $\mathbf{n}$ , para obtener las coordenadas del centro de la plataforma, se trata de imponer que las coordenadas de las conexiones de los seis actuadores con la plataforma móvil,  $\mathbf{r}_{Ai}$  con  $i$  desde 1 hasta 6, pertenezcan al plano, es decir que sean perpendiculares al vector  $\mathbf{n}$ . Para ello, se localizan los seis actuadores a partir del sistema de referencia local de la plataforma móvil (sólido 14 según el modelo de la sección 2.3), siendo  $\bar{\mathbf{u}}_{Ai}^{14}$  con  $i$  desde 1 hasta 6 el vector en coordenadas locales que indica la posición de la unión de cada actuador con la plataforma móvil que se definen:

$$\begin{aligned}\mathbf{r}_{Ai} &= \mathbf{r}_{14} + \mathbf{A}^{14}\bar{\mathbf{u}}_{Ai}^{14} \\ \bar{\mathbf{u}}_{A1}^{14} &= [R_m \cos(\varphi_m) \quad R_m \sin(\varphi_m) \quad 0]^T \\ \bar{\mathbf{u}}_{A2}^{14} &= [R_m \cos(\varphi_m + \alpha_m) \quad R_m \sin(\varphi_m + \alpha_m) \quad 0]^T \\ \bar{\mathbf{u}}_{A3}^{14} &= [R_m \cos(\varphi_m + \alpha_m + \beta_m) \quad R_m \sin(\varphi_m + \alpha_m + \beta_m) \quad 0]^T \\ \bar{\mathbf{u}}_{A4}^{14} &= [R_m \cos(\varphi_m + 2\alpha_m + \beta_m) \quad R_m \sin(\varphi_m + 2\alpha_m + \beta_m) \quad 0]^T \\ \bar{\mathbf{u}}_{A5}^{14} &= [R_m \cos(\varphi_m + 2\alpha_m + 2\beta_m) \quad R_m \sin(\varphi_m + 2\alpha_m + 2\beta_m) \quad 0]^T \\ \bar{\mathbf{u}}_{A6}^{14} &= [R_m \cos(\varphi_m + 3\alpha_m + 2\beta_m) \quad R_m \sin(\varphi_m + 3\alpha_m + 2\beta_m) \quad 0]^T\end{aligned}$$

Por lo tanto, imponiendo que los seis vectores  $\mathbf{r}_{Ai}$  sean perpendiculares al vector  $\mathbf{n}$ , se obtienen seis ecuaciones con siete incógnitas, pero añadiendo la restricción asociada los parámetros de Euler, es decir, que  $\boldsymbol{\theta}^{14}$  tiene que ser unitario se consigue un sistema de siete ecuaciones y siete incógnitas que son las siete coordenadas de la plataforma móvil. Para que tenga solución el sistema de ecuaciones, los haces de luz de los sensores no pueden ser paralelos.

Por otro lado, tras realizar el procedimiento de recalibración de la plataforma y el desarrollo de un controlador mejor, sería interesante fabricar una plataforma Stewart con mejores prestaciones que la empleada en este trabajo. Empleando unos actuadores con mayor capacidad de carga y velocidad de extensión, colocados entre dos plataformas de mayores dimensiones, podría emplearse dicha plataforma en una gran variedad de aplicaciones como se vio en el capítulo 1.





# REFERENCIAS

---

- [1] R. Aracil, R. J. Saltaré, J. M. Sabater, y O. Reinoso, «ROBOTS PARALELOS: MÁQUINAS CON UN PASADO PARA UNA ROBÓTICA DEL FUTURO», *Revista Iberoamericana de Automática e Informática Industrial*, vol. 3, pp. 16-28, 2006.
- [2] A. Tabor, «What is the Vertical Motion Simulator? - “Flying” Before You Can Fly | NASA», 2023. <https://www.nasa.gov/ames/vms> (accedido 21 de junio de 2023).
- [3] H. Mohn, «Moving-base driving simulator: Virtual reality on tracks | Mercedes-Benz Group > Company > Magazine > Technology & Innovation», 2021. <https://group.mercedes-benz.com/company/magazine/technology-innovation/100-things-driving-simulator.html> (accedido 21 de junio de 2023).
- [4] Acrome myCONTROL Team, «The Use of the Stewart Platform (aka Hexapod) in Real Life and Industry», 2022. <https://acrome.net/post/the-use-of-the-stewart-platform-in-real-life> (accedido 30 de junio de 2023).
- [5] A. Becker, «El Filtro de Kalman», 2023. [https://www.kalmanfilter.net/ES/default\\_es.aspx](https://www.kalmanfilter.net/ES/default_es.aspx) (accedido 30 de junio de 2023).
- [6] J. González-Carbajal, P. Urda, S. Muñoz, y J. L. Escalona, «Estimation of the trajectory and attitude of railway vehicles using inertial sensors with application to track geometry measurement», mar. 2022, doi: 10.1080/00423114.2023.2203865.
- [7] J. Hartikainen, A. Solin, y S. Särkkä, «Optimal Filtering with Kalman Filters and Smoothers a Manual for the Matlab toolbox EKF/UKF», 2011.
- [8] J. L. Escalona, «PRIMERA PRÁCTICA DE CINEMÁTICA Y DINÁMICA DE MÁQUINAS, MEDIDA EXPERIMENTAL Y PROCESADO DE DATOS DEL MOVIMIENTO DE UNA BICICLETA», 2023.
- [9] J. L. Escalona, «SIMULACIÓN CINEMÁTICA Y DINÁMICA INVERSA DEL MOVIMIENTO DE UNA BICICLETA A PARTIR DE MEDIDAS EXPERIMENTALES», 2022.
- [10] Acrome myCONTROL Team, «PROGRAMMING MANUAL STEWART PRO PLATFORM», 2021. [En línea]. Disponible en: [www.acrome.net](http://www.acrome.net)
- [11] J. L. Escalona, «DINÁMICA DE SISTEMAS MULTICUERPO», 2022.
- [12] «3DM-GX4-25™ Attitude Heading Reference System (AHRS)», 2015. [En línea]. Disponible en: <http://www.microstrain.com>
- [13] «Instruction Manual DPG-Series Inclinometer Instruction Manual DPG-Series Inclinometer Instruction Manual DPG-Series Inclinometer». [En línea]. Disponible en: [www.te.com](http://www.te.com)
- [14] «NI myRIO-1900 User Guide and Specifications - National Instruments; NI myRIO-1900 User Guide and Specifications - National Instruments».



# Anexo A CÓDIGOS DE MATLAB

---

A continuación, se muestran los códigos empleados durante este trabajo.

## A.1. Archivo para controlar la plataforma con un joystick mediante movimientos punto a punto “*joystick\_pto\_pto.m*”

```
clc
clear all
instrreset

% PARÁMETROS
vmax_x=10; %mm/s
vmax_y=10; %mm/s
vmax_z=10; %mm/s
x_0=0; %mm
y_0=0; %mm
z_0=433; %mm
r=0; %deg
p=0; %deg
yaw=0; %deg
dt=0.1; %s

% Crea un objeto de entrada del mando
joy = vrjoystick(1);

% ESTABLECE LA COMUNICACIÓN
t = tcpip('172.22.11.2',6500,'NetworkRole','Client','Timeout',2);
u = udp('172.22.11.1','LocalPort',6600);
fopen(t);
fopen(u);
pre_A=(1:1);
i=1;

% SE INICIA LA PLATAFORMA
B=("motor_check");
B=[B "kinematics,True"];
B=[B "enable,False"];
B=[B "enable,True"];
B=[B "kinematics,True"];
B=[B "move_platform,0,0,433,0,0,0"];
while (size(B,2)>i-1)
    fwrite(t, strlength(B(i)));
    fwrite(t,B(i));
    i=i+1;
end
%%
comma=",";

t_f = 100; % tiempo en segundos
t_start = tic; % tiempo de inicio
```

```

while toc(t_start) < t_f
    [a, b, p] = read(joy); % leer estado del joystick
    v_x=a(1,1)*vmax_x;
    v_y=-a(1,2)*vmax_y; % el signo - es por como reconoce el input
    v_z=a(1,3)*vmax_z;
    x=x_0+v_x*dt;
    y=y_0+v_y*dt;
    z=z_0+v_z*dt;
    if x>100
        x=100;
    end
    if x<-100
        x=-100;
    end
    if y>100
        y=100;
    end
    if y<-100
        y=-100;
    end
    if z>479
        z=479;
    end
    if z<385
        z=385;
    end
    X=num2str(x);
    Y=num2str(y);
    Z=num2str(z);
    R=num2str(r);
    P=num2str(p);
    YAW=num2str(yaw);
    MOV="move_platform,";
    MOV=strcat(MOV,X,comma);
    MOV=strcat(MOV,Y,comma);
    MOV=strcat(MOV,Z,comma);
    MOV=strcat(MOV,R,comma);
    MOV=strcat(MOV,P,comma);
    MOV=strcat(MOV,YAW); % se crea el string de movimiento por
    coordenadas
    B=[B MOV];
    fwrite(t,strlength(B(i)));
    fwrite(t,B(i));
    i=i+1;
    x_0=x;
    y_0=y;
    z_0=z;
end
pause(3) % se esperan 3 segundos para llevar la plataforma al punto de
referencia y desconectarla
B=("move_platform,0,0,433,0,0,0");
B=[B "enable,False"];
i=1;
while(size(B,2)>i-1)
    fwrite(t,strlength(B(i)));
    fwrite(t,B(i));

```

```

        pause(3);
        i=i+1;
end
disp("Connection is turned off")
fclose(u);
delete(u);
fclose(t);
delete(t);

```

## A.2. Archivo para controlar la plataforma con un joystick mediante control por trayectoria "joystick\_trayectoria.m"

```

clc
clear all
instrreset

% PARÁMETROS
vmax_x=10; % mm/s
vmax_y=10; % mm/s
vmax_z=10; % mm/s
w=3; % deg/s
x_0=0; % mm
y_0=0; % mm
z_0=433; % mm
r=0; % deg
p=0; % deg
yaw=0; % deg
dt=0.1; % s

% CREA UN OBJETO DE ENTRADA DE MANDO
joy = vrjoystick(1);

% ESTABLECE LA COMUNICACIÓN
t = tcpip('172.22.11.2',6500,'NetworkRole','Client','Timeout',2);
u = udp('172.22.11.1','LocalPort',6600);
fopen(t);
fopen(u);
pre_A=(1:1);
i=1;

% SE INICIA LA PLATAFORMA
B=("motor_check");
B=[B "SetPI,2.8,0"];
B=[B "enable,True"];
B=[B "kinematics,True"];
B=[B "move_platform,0,0,433,0,0,0"];
while(size(B,2)>i-1)
    fwrite(t,strlength(B(i)));
    fwrite(t,B(i));
    i=i+1;
end
%%
comma=","; % generamos el string de , para usarlo en la orden de
movimiento
T=num2str(dt);

```

```

YAW=num2str(yaw);
t_f = 100; % duración en segundos
t_start = tic; % tiempo de inicio
while toc(t_start) < t_f
    [a] = read(joy); % leer estado del joystick
    x=x_0+dt*a(1,1)*vmax_x;
    y=y_0+dt*(-a(1,2)*vmax_y); % el signo - es por como reconoce el
input
    z=z_0+dt*a(1,3)*vmax_z;
    r=r_0+w*a(1,4)*dt;
    p=p_0+w*a(1,5)*dt;
    if abs(x)>100
        x=(x/abs(x))*100;
    end
    if abs(y)>100
        y=(y/abs(y))*100;
    end
    if z>480
        z=480;
    end
    if z<385
        z=385;
    end
    if abs(r)>20
        r=(r/abs(r))*20;
    end
    if abs(p)>20
        p=(p/abs(p))*20;
    end
    X=num2str(x); % se crea el string de las coordenadas
    Y=num2str(y);
    Z=num2str(z);
    R=num2str(r);
    P=num2str(p);
    MOV="move_platform_traj,";

MOV=strcat(MOV,X,comma,Y,comma,Z,comma,R,comma,P,comma,YAW,comma,T); %
se genera la orden de movimiento
    fwrite(t,strlength(MOV));
    fwrite(t,MOV);
    r_0=r;
    p_0=p;
    x_0=x;
    y_0=y;
    z_0=z;
end

pause(3) % se esperan 3 segundos para llevar la plataforma al punto de
referencia y desconectarla
B=("move_platform,0,0,433,0,0,0");
B=[B "enable,False"];
i=1;
while(size(B,2)>i-1)
    fwrite(t,strlength(B(i)));
    fwrite(t,B(i));
    pause(3);
end

```

```
        i=i+1;
end
disp("Connection is turned off")
fclose(u);
delete(u);
fclose(t);
delete(t);
```

### A.3. Archivo para controlar la plataforma controlando cada actuador "control\_actuadores.m"

```
clc
clear all
% Se cargan las posiciones de los actuadores según el modelo de simulon
load ('control.mat')

% Barras pegadas a la plataforma fija
x02 = q(8, :);
y02 = q(9, :);
z02 = q(10, :);
x03 = q(15, :);
y03 = q(16, :);
z03 = q(17, :);
x04 = q(22, :);
y04 = q(23, :);
z04 = q(24, :);
x05 = q(29, :);
y05 = q(30, :);
z05 = q(31, :);
x06 = q(36, :);
y06 = q(37, :);
z06 = q(38, :);
x07 = q(43, :);
y07 = q(44, :);
z07 = q(45, :);
% Barras pegadas a la plataforma móvil.
x08 = q(50, :);
y08 = q(51, :);
z08 = q(52, :);
x09 = q(57, :);
y09 = q(58, :);
z09 = q(59, :);
x10 = q(64, :);
y10 = q(65, :);
z10 = q(66, :);
x11 = q(71, :);
y11 = q(72, :);
z11 = q(73, :);
x12 = q(78, :);
y12 = q(79, :);
z12 = q(80, :);
x13 = q(85, :);
y13 = q(86, :);
z13 = q(87, :);
```

```
% Longitudes de las barras
i=1;
x1=x08-x03;
y1=y08-y03;
z1=z08-z03;
r1=(x1.^2+y1.^2+z1.^2).^0.5;
for j=1:length(q)
    l(i,j)=389.32-10*r1(j);
end

i=i+1;
x2=x09-x04;
y2=y09-y04;
z2=z09-z04;
r2=(x2.^2+y2.^2+z2.^2).^0.5;
for j=1:length(q)
    l(i,j)=392.36-10*r2(j);
end

i=i+1;
x3=x10-x05;
y3=y10-y05;
z3=z10-z05;
r3=(x3.^2+y3.^2+z3.^2).^0.5;
for j=1:length(q)
    l(i,j)=389.32-10*r3(j);
end

i=i+1;
x4=x11-x06;
y4=y11-y06;
z4=z11-z06;
r4=(x4.^2+y4.^2+z4.^2).^0.5;
for j=1:length(q)
    l(i,j)=392.36-10*r4(j);
end

i=i+1;
x5=x12-x07;
y5=y12-y07;
z5=z12-z07;
r5=(x5.^2+y5.^2+z5.^2).^0.5;
for j=1:length(q)
    l(i,j)=389.32-10*r5(j);
end

i=i+1;
x6=x13-x02;
y6=y13-y02;
z6=z13-z02;
r6=(x6.^2+y6.^2+z6.^2).^0.5;
for j=1:length(q)
    l(i,j)=392.36-10*r6(j);
end
```



```

%% Inicio de la plataforma
% ESTABLECE LA COMUNICACIÓN
t = tcpip('172.22.11.2',6500,'NetworkRole','Client','Timeout',2);
fopen(t);
pre_A=(1:1);
i=1;

% SE INICIA LA PLATAFORMA
B=("ResetError");
B=[B "motor_check"];
B=[B "SetFrictionCompensator,0.3"];
B=[B "SetPI,2,0.5"];
B=[B "enable,True"];
B=[B "motors,True"];
B=[B "kinematics,True"];
while(size(B,2)>i-1)
    fwrite(t,strlength(B(i)));
    fwrite(t,B(i));
    i=i+1;
        while (1)
            A=pre_A;
            try
                G = fread(t, 1);
                A = fread(t, G);
            catch
                continue;
            end
            if (size(A,1)~=size(pre_A,1))
                break
            end
        end
        end
    disp("Response Message")
    str = convertCharsToStrings(char(A))
    pre_str=str;

end
pause(3)
B=("move_platform,0,0,433,0,0,0");
fwrite(t,strlength(B));
fwrite(t,B);
%% Se ejecuta el movimiento controlando los actuadores
comma=",";
i=1;
while i <= length(q)
    for j=1:6
        mot=num2str(j-1);
        desp=num2str(l(j,i));
        MOV="move_motor,";
        MOV=strcat(MOV,mot,comma);
        MOV=strcat(MOV,desp);
        fwrite(t,strlength(MOV));
        fwrite(t,MOV);
    end
    i=i+1;
end

%% Se desconecta la plataforma

```

```

pause(10)
B= ("move_platform,0,0,390,0,0,0");
B=[B "enable,False"];
i=1;
while(size(B,2)>i-1)
    fwrite(t,strlength(B(i)));
    fwrite(t,B(i));
    pause(3);
    i=i+1;
end
disp("Connection is turned off")
fclose(t);
delete(t);

```

#### A.4. Archivo para realizar la caracterización del inclinómetro moviendo la plataforma según un seno “ensayo\_seno.m”

```

clc
clear all
instrreset

% ESTABLECE LA COMUNICACIÓN
t = tcpip('172.22.11.2',6500,'NetworkRole','Client','Timeout',2);
fopen(t);
pre_A=(1:1);
i=1;

% SE INICIA LA PLATAFORMA
B= ("motor_check");
B=[B "ResetError"];
B=[B "SetFrictionCompensator,0.3"];
B=[B "SetPI,0.12,0"];
B=[B "enable,True"];
B=[B "motors,True"];
B=[B "kinematics,True"];

while(size(B,2)>i-1)
    fwrite(t,strlength(B(i)));
    fwrite(t,B(i));
    i=i+1;
    while (1)
        A=pre_A;
        try
            G = fread(t, 1);
            A = fread(t, G);
        catch
            continue;
        end
        if (size(A,1)~=size(pre_A,1))
            break
        end
    end
    disp("Response Message")
    str = convertCharsToStrings(char(A))
end

```

```

pause(3)
B=("move_platform,0,0,433,0,0,0");
fwrite(t, strlen(B));
fwrite(t, B);
%%
comma=",";
F=0.8; % Hz
A=2; % Amplitud en grados
Tf=1/F;
w=2*pi*F;
tw = 0:0.01:20;
phi=round(A*sin(w*tw),2); % se redondea a 2 decimales, pues la
plataforma funciona mejor así

i=1;
figure(1); plot(tw,phi); grid on
while i <=length(phi)
    R=num2str(phi(i));
    MOV=strcat("move_platform_traj,0,0,433,",R,",0,0,0.01");
    fwrite(t, strlen(MOV));
    fwrite(t, MOV);
    i=i+1;
    % El bucle para leer la respuesta de la plataforma está comentado
para
    % evitar que se ralentice la comunicación
    % while (1)
    %     A=pre_A;
    %     try
    %         G = fread(t, 1);
    %         A = fread(t, G);
    %     catch
    %         continue;
    %     end
    %     if (size(A,1)~=size(pre_A,1))
    %         break
    %     end
    % end
    % disp("Response Message")
    % str = convertCharsToStrings(char(A))
end
pause(15)
% Se lleva a la plataforma al punto de referencia
B=("move_platform,0,0,433,0,0,0");
fwrite(t, strlen(B));
fwrite(t, B);

%% Se desconecta la plataforma
B=("move_platform,0,0,390,0,0,0");
B=[B "enable,False"];
i=1;
while(size(B,2)>i-1)
    fwrite(t, strlen(B(i)));
    fwrite(t, B(i));
    pause(3);
    i=i+1;

```

```

end
disp("Connection is turned off")
fclose(t);
delete(t);

```

### A.5. Archivo para realizar las gráficas de la calibración de la sección 3.2 "graficas\_calibracion.m"

```

clear
clc
%%%%%%%%%%
%%% CARGA DATOS %%%
%%%%%%%%%%

DATOS =
readtable('DATOS_CALIBRACION.xlsx','Sheet','Hojal','HeaderLines',1);
DATOS = table2array(DATOS);
x=DATOS(:,1);
y=DATOS(:,2);
z=DATOS(:,3);
roll=DATOS(:,4);
pitch=DATOS(:,5);
yaw=DATOS(:,6);

act_1=round(DATOS(:,8),2);
act_2=round(DATOS(:,9),2);
act_3=round(DATOS(:,10),2);
act_4=round(DATOS(:,11),2);
act_5=round(DATOS(:,12),2);
act_6=round(DATOS(:,13),2);

figure(1)
plot(pitch(1:8,1),act_1(1:8,1),'o',pitch(1:8,1),act_2(1:8,1),'o',pitch(1:8,1),act_3(1:8,1),'o',pitch(1:8,1),act_4(1:8,1),'o',pitch(1:8,1),act_5(1:8,1),'o',pitch(1:8,1),act_6(1:8,1),'o','MarkerFaceColor','w')
xlabel('Pitch (°)')
ylabel('Respuesta medida/Respuesta teórica')
title('Relación desplazamientos medidos frente a teóricos')
legend('Actuador 1','Actuador 2','Actuador 3','Actuador 4','Actuador 5','Actuador 6','Location','northeastoutside')
ylim([0.85 1.15])
grid on

figure(2)
plot(roll(9:16,1),act_1(9:16,1),'o',roll(9:16,1),act_2(9:16,1),'o',roll(9:16,1),act_3(9:16,1),'o',roll(9:16,1),act_4(9:16,1),'o',roll(9:16,1),act_5(9:16,1),'o',roll(9:16,1),act_6(9:16,1),'o','MarkerFaceColor','w')
xlabel('Roll (°)')
ylabel('Respuesta medida/Respuesta teórica')
title('Relación desplazamientos medidos frente a teóricos')
legend('Actuador 1','Actuador 2','Actuador 3','Actuador 4','Actuador 5','Actuador 6','Location','northeastoutside')
ylim([0.85 1.15])
grid on

```

```

figure(3)
plot(x(17:21,1),act_1(17:21,1),'o',x(17:21,1),act_2(17:21,1),'o',x(17:21,1),act_3(17:21,1),'o',x(17:21,1),act_4(17:21,1),'o',x(17:21,1),act_5(17:21,1),'o',x(17:21,1),act_6(17:21,1),'o','MarkerFaceColor','w')
xlabel('x (mm)')
ylabel('Respuesta medida/Respuesta teórica')
title('Relación desplazamientos medidos frente a teóricos')
legend('Actuador 1','Actuador 2','Actuador 3','Actuador 4','Actuador 5','Actuador 6','Location','northeastoutside')
ylim([0.85 1.15])
grid on

figure(4)
plot(y(22:26,1),act_1(22:26,1),'o',y(22:26,1),act_2(22:26,1),'o',y(22:26,1),act_3(22:26,1),'o',y(22:26,1),act_4(22:26,1),'o',y(22:26,1),act_5(22:26,1),'o',y(22:26,1),act_6(22:26,1),'o','MarkerFaceColor','w')
xlabel('y (mm)')
ylabel('Respuesta medida/Respuesta teórica')
title('Relación desplazamientos medidos frente a teóricos')
legend('Actuador 1','Actuador 2','Actuador 3','Actuador 4','Actuador 5','Actuador 6','Location','northeastoutside')
ylim([0.85 1.15])
grid on

figure(5)
plot(z(27:32,1),act_1(27:32,1),'o',z(27:32,1),act_2(27:32,1),'o',z(27:32,1),act_3(27:32,1),'o',z(27:32,1),act_4(27:32,1),'o',z(27:32,1),act_5(27:32,1),'o',z(27:32,1),act_6(27:32,1),'o','MarkerFaceColor','w')
xlabel('z (mm)')
ylabel('Respuesta medida/Respuesta teórica')
title('Relación desplazamientos medidos frente a teóricos')
legend('Actuador 1','Actuador 2','Actuador 3','Actuador 4','Actuador 5','Actuador 6','Location','northeastoutside')
ylim([0.85 1.15])
grid on

```

## A.6. API proporcionada por el fabricante “StewartPro\_Scenario.m”

```

clc
clear all
instrreset
t = tcpip('172.22.11.2',6500,'NetworkRole','Client','Timeout',2);
u = udp('172.22.11.1','LocalPort',6600);
fopen(t);
fopen(u);
pre_A=(1:1);
i=1;

B=("motor_check");
B=["kinematics,True"];
B=[B "enable,False"];
B=[B "enable,True"];
B=[B "kinematics,True"];
%-----Example-----

```

```

B=[B "move_platform,0,0,450,0,0,0"];
%-----

pre_str="";
while(size(B,2)>i-1)
    fwrite(t,strlen(B(i)));
    fwrite(t,B(i));
    while (1)
        A=pre_A;
        try
            G = fread(t, 1);
            A = fread(t, G);
        catch
            continue;
        end
        if (size(A,1)~=size(pre_A,1))
            break
        end
    end
    disp("Response Message")
    str = convertCharsToStrings(char(A))
    i=i+1;
    pre_str=str;
    try
        A = fread(u, 1);
        A = fread(u, A);
        disp("Feedback Message")
        str = convertCharsToStrings(char(A))
    catch ME
        disp(ME)
    end
end
disp("Connection is turned off")
fclose(u);
delete(u);
fclose(t);
delete(t);

```

## A.7. Archivo para realizar la calibración “*calibracioninicial.m*”

```

%% INICIO
clc
clear all
instrreset

% ESTABLECE LA COMUNICACIÓN
t = tcpip('172.22.11.2',6500,'NetworkRole','Client','Timeout',2);
fopen(t);
pre_A=(1:1);
i=1;

% SE INICIA LA PLATAFORMA
B("ResetError");
B=[B "motor_check"];

```

```
B=[B "SetFrictionCompensator,0.24"];
B=[B "SetPI,0.12,0"];
B=[B "enable,True"];
B=[B "kinematics,True"];
while (size(B,2)>i-1)
    fwrite(t, strlen(B(i)));
    fwrite(t, B(i));
    i=i+1;
    while (1)
        A=pre_A;
        try
            G = fread(t, 1);
            A = fread(t, G);
        catch
            continue;
        end
        if (size(A,1)~=size(pre_A,1))
            break
        end
        end
        disp("Response Message")
        str = convertCharsToStrings(char(A))
        pre_str=str;

end
pause(3)
B=("move_platform,0,0,433,0,0,0");
fwrite(t, strlen(B));
fwrite(t, B);
%% MOVIMIENTO
MOV="move_platform_traj,0,0,433,10,0,0,2"; % se coloca la plataforma
en el punto de calibración
fwrite(t, strlen(MOV));
fwrite(t, MOV);
%% Desconectar plataforma
B=("move_platform,0,0,390,0,0,0");
B=[B "enable,False"];
i=1;
while (size(B,2)>i-1)
    fwrite(t, strlen(B(i)));
    fwrite(t, B(i));
    pause(3);
    i=i+1;
end
disp("Connection is turned off")
fclose(t);
delete(t);
```





# Anexo B CÓDIGOS DEL MODELO DE SIMULON

En las siguientes páginas se muestran como quedan los códigos que han sido modificados o creados para adaptar el modelo de Simulon a la plataforma Stewart con la que se trabaja.

## B.1. Función que define las restricciones cinemáticas de la plataforma Stewart “AnimaStewart.m”

```
function MBody = PlataformaStewart

MBody.Dim = 3;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% DESCRIPCION DEL MECANISMO %%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% Montaje de la estructura del mecanismo.
MBody.nombre = 'StewartPlatform';
%% Datos enteros básicos del mecanismo
MBody.nb = 14; %Número de sólidos
MBody.nr = 0; %Numero de pares de revolucion%%par entre
solidos 1 y 2
MBody.np = 0;
MBody.ncar = 12; %Número de pares cardan
MBody.nc = 6; %Número de pares cilíndricos
MBody.ne = 0; %Numero de pares esfericos%%par entre solidos
2 y 3
MBody.nl = 0;
MBody.npl = 0; %Restricciones punto a linea
MBody.nrus = 6; %Restricciones definindas por el usuario,
incluidas las de movilidad
%% Vector de estructuras de sólidos
MBody.Solidos(1).nombre = 'Plataforma fija';
MBody.Solidos(2).nombre = 'Barra 1';
MBody.Solidos(3).nombre = 'Barra 2';
MBody.Solidos(4).nombre = 'Barra 3';
MBody.Solidos(5).nombre = 'Barra 4';
MBody.Solidos(6).nombre = 'Barra 5';
MBody.Solidos(7).nombre = 'Barra 6';
MBody.Solidos(8).nombre = 'Barra 7';
MBody.Solidos(9).nombre = 'Barra 8';
MBody.Solidos(10).nombre = 'Barra 9';
MBody.Solidos(11).nombre = 'Barra 10';
MBody.Solidos(12).nombre = 'Barra 11';
MBody.Solidos(13).nombre = 'Barra 12';
MBody.Solidos(14).nombre = 'Plataforma móvil';

% Coordenadas, generalizadas
MBody.ncoord = 7*MBody.nb; %98 coordenadas 7*14=98
% N° Total de restricciones
MBody.nrestr = 6+4*MBody.nc+4*MBody.ncar+MBody.nrus+MBody.nb; %Esta
última
```

```

%añade 1 ec por barra debido a la normalización de los parámetros de
Euler
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Dirección y sentido de la gravedad %%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

MBody.g = [0 0 -9.81]';

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Coordenadas independientes %%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

MBody.Ind = [92 93 94 95 96 97]';

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%% Selección de coordenadas dependientes (las no independientes) %%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%

es_ind = 0;
ii = 1;

for i = 1:MBody.ncoord
    for j = 1:length(MBody.Ind)
        if i == MBody.Ind(j)
            es_ind = 1;
        end
    end

    if es_ind == 0
        MBody.Dep(ii,1) = i;
        ii = ii+1;
    end

    es_ind = 0;
end

%% Vector de estructuras de pares cardan
%% Pares cardan de la plataforma fija
Rf=170; % mm
ang_o=-pi/12; % El eje X pasa por el punto medio entre las juntas 1 y
2
zb=(16+43.15-6.1); % mm
MBody.Pares.Car(1).I = 1;
MBody.Pares.Car(1).J = 2;
MBody.Pares.Car(1).uPi = [Rf*cos(ang_o) Rf*sin(ang_o) zb]';
MBody.Pares.Car(1).uPj = [0 0 0]';
MBody.Pares.Car(1).si = [-sin(ang_o) cos(pi/6+ang_o) 0]';
MBody.Pares.Car(1).sj = [1 0 0]';
MBody.Pares.Car(2).I = 1;
MBody.Pares.Car(2).J = 3;
MBody.Pares.Car(2).uPi = [Rf*cos(pi/6+ang_o) Rf*sin(pi/6+ang_o) zb ]';
MBody.Pares.Car(2).uPj = [0 0 0]';
MBody.Pares.Car(2).si = [-sin(pi/6+ang_o) cos(pi/6+ang_o) 0]';
MBody.Pares.Car(2).sj = [1 0 0]';

```

```

MBody.Pares.Car(3).I = 1;
MBody.Pares.Car(3).J = 4;
MBody.Pares.Car(3).uPi = [Rf*cos(2*pi/3+ang_o) Rf*sin(2*pi/3+ang_o) zb
]';
MBody.Pares.Car(3).uPj = [0 0 0]';
MBody.Pares.Car(3).si = [-sin(2*pi/3+ang_o) cos(2*pi/3+ang_o) 0]';
MBody.Pares.Car(3).sj = [1 0 0]';
MBody.Pares.Car(4).I = 1;
MBody.Pares.Car(4).J = 5;
MBody.Pares.Car(4).uPi = [Rf*cos(5*pi/6+ang_o) Rf*sin(5*pi/6+ang_o) zb
]';
MBody.Pares.Car(4).uPj = [0 0 0]';
MBody.Pares.Car(4).si = [-sin(5*pi/6+ang_o) cos(5*pi/6+ang_o) 0]';
MBody.Pares.Car(4).sj = [1 0 0]';
MBody.Pares.Car(5).I = 1;
MBody.Pares.Car(5).J = 6;
MBody.Pares.Car(5).uPi = [Rf*cos(4*pi/3+ang_o) Rf*sin(4*pi/3+ang_o) zb
]';
MBody.Pares.Car(5).uPj = [0 0 0]';
MBody.Pares.Car(5).si = [-sin(4*pi/3+ang_o) cos(4*pi/3+ang_o) 0]';
MBody.Pares.Car(5).sj = [1 0 0]';
MBody.Pares.Car(6).I = 1;
MBody.Pares.Car(6).J = 7;
MBody.Pares.Car(6).uPi = [Rf*cos(3*pi/2+ang_o) Rf*sin(3*pi/2+ang_o) zb
]';
MBody.Pares.Car(6).uPj = [0 0 0]';
MBody.Pares.Car(6).si = [-sin(3*pi/2+ang_o) cos(3*pi/2+ang_o) 0]';
MBody.Pares.Car(6).sj = [1 0 0]';
% %Pares Cardan de la plataforma móvil. Girada 32.5° alrededor de la
vertical.
gir=47.5*pi/180;
Rm=130; %mm
zm=-51.9; %mm
% 25 grados entre dos Cardan sucesivos (1-2,3-4,5-6)

MBody.Pares.Car(7).I = 14;
MBody.Pares.Car(7).J = 8;
MBody.Pares.Car(7).uPi = [Rm*cos(gir) Rm*sin(gir) zm]';
MBody.Pares.Car(7).uPj = [0 0 0]';
MBody.Pares.Car(7).si = [-sin(gir) cos(gir) 0]';
MBody.Pares.Car(7).sj = [1 0 0]';
MBody.Pares.Car(8).I = 14;
MBody.Pares.Car(8).J = 9;
MBody.Pares.Car(8).uPi = [Rm*cos(25*pi/180+gir) Rm*sin(25*pi/180+gir)
zm]';
MBody.Pares.Car(8).uPj = [0 0 0]';
MBody.Pares.Car(8).si = [-sin(25*pi/180+gir) cos(25*pi/180+gir) 0]';
MBody.Pares.Car(8).sj = [1 0 0]';
MBody.Pares.Car(9).I = 14;
MBody.Pares.Car(9).J = 10;
MBody.Pares.Car(9).uPi = [Rm*cos(2*pi/3+gir) Rm*sin(2*pi/3+gir) zm ]';
MBody.Pares.Car(9).uPj = [0 0 0]';
MBody.Pares.Car(9).si = [-sin(2*pi/3+gir) cos(2*pi/3+gir) 0]';
MBody.Pares.Car(9).sj = [1 0 0]';
MBody.Pares.Car(10).I = 14;
MBody.Pares.Car(10).J = 11;

```

```

MBody.Pares.Car(10).uPi          =          [Rm*cos(25*pi/180+2*pi/3+gir)
Rm*sin(25*pi/180+2*pi/3+gir) zm ]';
MBody.Pares.Car(10).uPj = [0 0 0]';
MBody.Pares.Car(10).si          =          [-sin(25*pi/180+2*pi/3+gir)
cos(25*pi/180+2*pi/3+gir) 0]';
MBody.Pares.Car(10).sj = [1 0 0]';
MBody.Pares.Car(11).I = 14;
MBody.Pares.Car(11).J = 12;
MBody.Pares.Car(11).uPi = [Rm*cos(4*pi/3+gir) Rm*sin(4*pi/3+gir) zm]';
MBody.Pares.Car(11).uPj = [0 0 0]';
MBody.Pares.Car(11).si = [-sin(4*pi/3+gir) cos(4*pi/3+gir) 0]';
MBody.Pares.Car(11).sj = [1 0 0]';
MBody.Pares.Car(12).I = 14;
MBody.Pares.Car(12).J = 13;
MBody.Pares.Car(12).uPi          =          [Rm*cos(25*pi/180+4*pi/3+gir)
Rm*sin(25*pi/180+4*pi/3+gir) zm ]';
MBody.Pares.Car(12).uPj = [0 0 0]';
MBody.Pares.Car(12).si          =          [-sin(25*pi/180+4*pi/3+gir)
cos(25*pi/180+4*pi/3+gir) 0]';
MBody.Pares.Car(12).sj = [1 0 0]';

```

```

%% Vector de estructuras de pares cilíndricos.

```

```

MBody.Pares.Cilin(1).I = 3;
MBody.Pares.Cilin(1).J = 8;
MBody.Pares.Cilin(1).vi = [0 0 1]';
MBody.Pares.Cilin(1).vj = [0 0 1]';
MBody.Pares.Cilin(1).ui = [0 0 0]';
MBody.Pares.Cilin(1).uj = [0 0 0]';
MBody.Pares.Cilin(2).I = 4;
MBody.Pares.Cilin(2).J = 9;
MBody.Pares.Cilin(2).vi = [0 0 1]';
MBody.Pares.Cilin(2).vj = [0 0 1]';
MBody.Pares.Cilin(2).ui = [0 0 0]';
MBody.Pares.Cilin(2).uj = [0 0 0]';
MBody.Pares.Cilin(3).I = 5;
MBody.Pares.Cilin(3).J = 10;
MBody.Pares.Cilin(3).vi = [0 0 1]';
MBody.Pares.Cilin(3).vj = [0 0 1]';
MBody.Pares.Cilin(3).ui = [0 0 0]';
MBody.Pares.Cilin(3).uj = [0 0 0]';
MBody.Pares.Cilin(4).I = 6;
MBody.Pares.Cilin(4).J = 11;
MBody.Pares.Cilin(4).vi = [0 0 1]';
MBody.Pares.Cilin(4).vj = [0 0 1]';
MBody.Pares.Cilin(4).ui = [0 0 0]';
MBody.Pares.Cilin(4).uj = [0 0 0]';
MBody.Pares.Cilin(5).I = 7;
MBody.Pares.Cilin(5).J = 12;
MBody.Pares.Cilin(5).vi = [0 0 1]';
MBody.Pares.Cilin(5).vj = [0 0 1]';
MBody.Pares.Cilin(5).ui = [0 0 0]';
MBody.Pares.Cilin(5).uj = [0 0 0]';
MBody.Pares.Cilin(6).I = 2;
MBody.Pares.Cilin(6).J = 13;
MBody.Pares.Cilin(6).vi = [0 0 1]';
MBody.Pares.Cilin(6).vj = [0 0 1]';

```

```

MBody.Pares.Cilin(6).ui = [0 0 0]';
MBody.Pares.Cilin(6).uj = [0 0 0]';

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% OTROS PARAMETROS DEL MECANISMO %%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

MBody.Param.w = 2*pi; % Velocidad angular

```

## B.2. Función que define las restricciones de movilidad de la plataforma Stewart “*RestrUsuario.m*”

```

function Cus = RestrUsuario(q,t)

global MBody

w = MBody.Param.w;
MBody.Param.A=10*pi/180/2; %rad (Se divide entre dos para alabeo,
cabeceo o guiñada)
A=MBody.Param.A;

% % Calibración
% Cus(1,1) = q(92);
% Cus(2,1) = q(93);
% Cus(3,1) = q(94)-475;
% Cus(4,1) = q(95)-1;%-cos(14*pi/180/2);
% Cus(5,1) = q(96);
% Cus(6,1) = q(97);%-sin(14*pi/180/2);

% Alabeo
Cus(1,1) = q(92);
Cus(2,1) = q(93);
Cus(3,1) = q(94)-(433);
Cus(4,1) = q(95)-cos(A*sin(w*t));
Cus(5,1) = q(96)-sin(A*sin(w*t));
Cus(6,1) = q(97);

% %Cabeceo
% Cus(1,1) = q(92);
% Cus(2,1) = q(93);
% Cus(3,1) = q(94)-433;
% Cus(4,1) = q(95)-cos(pi/18/2*sin(w*t));
% Cus(5,1) = q(96);
% Cus(6,1) = q(97)-sin(pi/18/2*sin(w*t));

% %Guiñada
% Cus(1,1) = q(92);
% Cus(2,1) = q(93);
% Cus(3,1) = q(94)-43.3;
% Cus(4,1) = q(95)-cos(pi/18/2*sin(w*t));
% Cus(5,1) = q(96);
% Cus(6,1) = q(98)-sin(pi/18/2*sin(w*t));

```

### B.3. Función que define las primeras derivadas temporales de las restricciones de movilidad de la plataforma Stewart “dtRestr.m”

```
function Ct = dtRestr(q,t)

global MBody

if nargin == 1
    t = MBody.t;
end

if nargin == 0
    t = MBody.t;
    q = MBody.q;
end

w=MBody.Param.w;%VELOCIDAD ANGULAR
A=MBody.Param.A;
Ct(MBody.nrestr-6,1) = 0;

% % Calibración
% Ct(MBody.nrestr-5,1)= 0; %Primera ecuación
% Ct(MBody.nrestr-4,1)= 0; %Segunda ecuación
% Ct(MBody.nrestr-3,1)= 0; %Tercera ecuación
% Ct(MBody.nrestr-2,1)= 0; %Cuarta ecuación
% Ct(MBody.nrestr-1,1)= 0; %Quinta ecuación
% Ct(MBody.nrestr,1)= 0; %Sexta ecuación

% Alabeo
Ct(MBody.nrestr-5,1)= 0; %Primera ecuación
Ct(MBody.nrestr-4,1)= 0; %Segunda ecuación
Ct(MBody.nrestr-3,1)= 0; %Tercera ecuación
Ct(MBody.nrestr-2,1)= sin(A*sin(w*t))*A*cos(w*t)*w; %Cuarta ecuación
Ct(MBody.nrestr-1,1)= -cos(A*sin(w*t))*A*cos(w*t)*w; %Quinta ecuación
Ct(MBody.nrestr,1)= 0; %Sexta ecuación

% % % Cabeceo
% Ct(MBody.nrestr-5,1)= 0; %Primera ecuación
% Ct(MBody.nrestr-4,1)= 0; %Segunda ecuación
% Ct(MBody.nrestr-3,1)= 0; %Tercera ecuación
% Ct(MBody.nrestr-2,1)= sin(A*sin(w*t))*A*cos(w*t)*w; %Cuarta ecuación
% Ct(MBody.nrestr-1,1)= 0; %Quinta ecuación
% Ct(MBody.nrestr,1)= -cos(A*sin(w*t))*A*cos(w*t)*w; %Sexta ecuación
%

% % Guiñada
% Ct(MBody.nrestr-5,1)= 0; %Primera ecuación
% Ct(MBody.nrestr-4,1)= 0; %Segunda ecuación
% Ct(MBody.nrestr-3,1)= 0; %Tercera ecuación
% Ct(MBody.nrestr-2,1)= sin(A*sin(w*t))*A*cos(w*t)*w; %Cuarta ecuación
% Ct(MBody.nrestr-1,1)= 0; %Quinta ecuación
% Ct(MBody.nrestr,1)= -cos(A*sin(w*t))*A*cos(w*t)*w; %Sexta ecuación
```

#### B.4. Función que define las segundas derivadas temporales de las restricciones de movilidad de la plataforma Stewart “DtdtRestr.m”

```

function DcT = DtdtRestr(q,v,t)

global MBody

if nargin == 1
    t = MBody.t;
end

if nargin == 0
    t = MBody.t;
    q = MBody.q;
end

DcT(MBody.nrestr-6,1) = 0;

A=MBody.Param.A;
w=MBody.Param.w;%VELOCIDAD ANGULAR

% % Calibración
% DcT(MBody.nrestr-5,1)= 0; %Primera ecuación
% DcT(MBody.nrestr-4,1)= 0; %Segunda ecuación
% DcT(MBody.nrestr-3,1)= 0; %Tercera ecuación
% DcT(MBody.nrestr-2,1)= 0; %Cuarta ecuación
% DcT(MBody.nrestr-1,1)= 0; %Quinta ecuación
% DcT(MBody.nrestr,1)= 0; %Sexta ecuación

% % % Cabeceo
% DcT(MBody.nrestr-5,1)= 0; %Primera ecuación
% DcT(MBody.nrestr-4,1)= 0; %Segunda ecuación
% DcT(MBody.nrestr-3,1)= 0; %Tercera ecuación
%DcT(MBody.nrestr-2,1)=w*A*(cos(A*sin(w*t))*w*A*cos(w*t)^2-
w*sin(w*t)*sin(A*sin(w*t))); %Cuarta ecuación
% DcT(MBody.nrestr-1,1)= 0; %Quinta ecuación
%DcT(MBody.nrestr,1)=w*A*(sin(A*sin(w*t))*w*A*cos(w*t)^2+w*sin(w*t)*c
os(A*sin(w*t))); %Sexta ecuación

%%Alabeo
DcT(MBody.nrestr-5,1)= 0; %Primera ecuación
DcT(MBody.nrestr-4,1)= 0; %Segunda ecuación
DcT(MBody.nrestr-3,1)= 0; %Tercera ecuación
DcT(MBody.nrestr-2,1)=
w*A*(cos(A*sin(w*t))*w*A*cos(w*t)^2-
w*sin(w*t)*sin(A*sin(w*t))); %Cuarta ecuación
DcT(MBody.nrestr-1,1)=
w*A*(sin(A*sin(w*t))*w*A*cos(w*t)^2+w*sin(w*t)*cos(A*sin(w*t)));
%Quinta ecuación
DcT(MBody.nrestr,1)= 0; %Sexta ecuación

% % Guiñada
%DcT(MBody.nrestr-5,1)= 0; %Primera ecuación
%DcT(MBody.nrestr-4,1)= 0; %Segunda ecuación

```

```

%Dct(MBody.nrestr-3,1)= 0; %Tercera ecuación
%Dct(MBody.nrestr-2,1)=w*A*(cos(A*sin(w*t))*w*A*cos(w*t)^2-
w*sin(w*t)*sin(A*sin(w*t))); %Cuarta ecuación
%Dct(MBody.nrestr-1,1)= 0; %Quinta ecuación
%Dct(MBody.nrestr,1)=5wA*(sin(A*sin(w*t))*w*A*cos(w*t)^2+w*sin(w*t)*c
os(A*sin(w*t))); %Sexta ecuación

```

### B.5. Función que define la estimación inicial de las coordenadas de la plataforma Stewart “EstimacionInicial.m”

% Función que calcula una estimación inicial decente para empezar a resolver el mecanismo.

```

function q0=EstimacionInicial(q)
%% Plataforma fija (Barra fija)
q(1:3)=0;
q(4)=1;
q(5:7)=0;
%% Barras pegadas a la plataforma fija.
vx=0;
vy=0;
vz=1;
Rf=170;
ang_o=-pi/12;
zb=(16+43.15-6.1); % mm
k=2;
for i=[ang_o pi/6+ang_o 2*pi/3+ang_o 5*pi/6+ang_o 4*pi/3+ang_o
3*pi/2+ang_o]
q(7*k-6:7*k)=[Rf*cos(i) Rf*sin(i) zb cos(i/2) vx*sin(i/2) vy*sin(i/2)
vz*sin(i/2)];
k=k+1;
end
%% Barras pegadas a la plataforma móvil.
vx=0;
vy=0;
vz=1;
Rm=130;
zm=-51.9; %mm
k=8; %La primera barra pegada a la plataforma movil es el sólido 8. Ya
% viene del bucle anterior.
gir=47.5*pi/180;
for i=[0+gir 25*pi/180+gir 2*pi/3+gir 2*pi/3+25*pi/180+gir 4*pi/3+gir
4*pi/3+25*pi/180+gir]
q(7*k-6:7*k)=[Rm*cos(i) Rm*sin(i) 43.3+zm cos(i/2) vx*sin(i/2)
vy*sin(i/2) vz*sin(i/2)];
k=k+1;
end

%% Plataforma móvil.
vx=0;
vy=0;
vz=1;
i=0; %El Sistema de Referencia no está girado respecto a la fija
q(7*k-6:7*k)=[0 0 43.3 cos(i/2) vx*sin(i/2) vy*sin(i/2) vz*sin(i/2)];
%% Lo metemos todo en el vector q0
q0=q;

```



```
end
```

## B.6. Función que calcula la longitud de cada barra de la plataforma Stewart “LongitudBarras.m”

```
function l=LongitudBarras(q)
%% Definición de los puntos notables del mecanismo.
% Barras pegadas a la plataforma fija
x02 = q(8, :);
y02 = q(9, :);
z02 = q(10, :);
x03 = q(15, :);
y03 = q(16, :);
z03 = q(17, :);
x04 = q(22, :);
y04 = q(23, :);
z04 = q(24, :);
x05 = q(29, :);
y05 = q(30, :);
z05 = q(31, :);
x06 = q(36, :);
y06 = q(37, :);
z06 = q(38, :);
x07 = q(43, :);
y07 = q(44, :);
z07 = q(45, :);
% Barras pegadas a la plataforma móvil.
x08 = q(50, :);
y08 = q(51, :);
z08 = q(52, :);
x09 = q(57, :);
y09 = q(58, :);
z09 = q(59, :);
x10 = q(64, :);
y10 = q(65, :);
z10 = q(66, :);
x11 = q(71, :);
y11 = q(72, :);
z11 = q(73, :);
x12 = q(78, :);
y12 = q(79, :);
z12 = q(80, :);
x13 = q(85, :);
y13 = q(86, :);
z13 = q(87, :);

%% Barras
i=1;
x1=x08-x03;
y1=y08-y03;
z1=z08-z03;
r1=(x1.^2+y1.^2+z1.^2).^0.5;
l(i)=r1(end);

i=i+1;
```

```

x2=x09-x04;
y2=y09-y04;
z2=z09-z04;
r2=(x2.^2+y2.^2+z2.^2).^0.5;
l(i)=r2(end);

i=i+1;
x3=x10-x05;
y3=y10-y05;
z3=z10-z05;
r3=(x3.^2+y3.^2+z3.^2).^0.5;
l(i)= r3(end);

i=i+1;
x4=x11-x06;
y4=y11-y06;
z4=z11-z06;
r4=(x4.^2+y4.^2+z4.^2).^0.5;
l(i)=r4(end);

i=i+1;
x5=x12-x07;
y5=y12-y07;
z5=z12-z07;
r5=(x5.^2+y5.^2+z5.^2).^0.5;
l(i)=r5(end);

i=i+1;
x6=x13-x02;
y6=y13-y02;
z6=z13-z02;
r6=(x6.^2+y6.^2+z6.^2).^0.5;
l(i)=r6(end);

for i=1:6
    lon=l(i);
    fprintf('l%d = %.2f mm \n',i,lon)
end

```

## B.7. Código que resuelve la cinemática de la plataforma Stewart “*SimulacionCinematicaPlatStew.m*”

```

clc
clear all
global MBody

MBody = PlataformaStewart;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% SIMULACIÓN CINEMÁTICA %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

w=MBody.Param.w;%VELOCIDAD ANGULAR

tspan= 0:0.01:2; %TIEMPO DE INTEGRACIÓN

```

```

%Estimacion inicial
q0=zeros(98,1);
q0=EstimacionInicial';
q00=q0;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% BUCLE DE SOLUCIÓN DEL PROBLEMA DE POSICIÓN %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

for i=1:length(tspan)
    MBody.t = tspan(i);
    %PROBLEMA DE POSICION
    [q(:,i),fval(:,i),exitflag(i),output(i)]=
fsolve(@Restricciones,q0,optimset('Display','on'));
    MBody.q=q(:,i);
    Cq = Jacobiano(q(:,i));
    Ct = dtRestr;
    %CALCULO DE VELOCIDADES
    v(:,i) = -Cq\Ct;
    DCq = DtJacobiano(q(:,i),v(:,i));
    DCt = DtdtRestr;
    %CALCULO DE ACELERACIONES
    a(:,i) = -Cq\ (DCq*v(:,i)+DCt);
    q0 = q(:,i);
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% POSTPROCESO DE RESULTADOS %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% p=1;
% for k=1:MBody.nb
% figure(p)
% plot(tspan,v(1+7*(k-1),:),'g')
% hold on
% plot(tspan,v(2+7*(k-1),:),'b')
% plot(tspan,v(3+7*(k-1),:),'r')
% xlabel('t(s)')
% ylabel('v(mm/s)')
% title(['Velocidades solido ', num2str(k)])
% legend('Eje x','Eje y','Eje z')
% % filename=['Vel_' num2str(k) '.png'];
% % saveas([p],filename)
% p=p+1;
% end
% p=1;
% for k=1:MBody.nb
% figure(p)
% plot(tspan,a(1+7*(k-1),:),'g')
% hold on
% plot(tspan,a(2+7*(k-1),:),'b')
% plot(tspan,a(3+7*(k-1),:),'r')
% xlabel('t(s)')
% ylabel('a(mm/s^2)')
% title(['Aceleraciones solido ', num2str(k)])
% legend('Eje x','Eje y','Eje z')
% % filename=['Ac_' num2str(k) '.png'];

```

```
% % saveas ([p],filename)
% p=p+1;
% end

% ANIMACIÓN
figure
AnimaStewart(q,tspan)

% filename='control.mat'; % Para almacenar las coordenadas y controlar
la plataforma con los actuadores
% save(filename)
l= LongitudBarras(q); % Llamada a la función que da las longitudes de
% las barras
```

# Anexo C PLANO DEL UTILLAJE

