UNIVERSIDAD MIGUEL HERNÁNDEZ DE ELCHE
ESCUELA POLITÉCNICA SUPERIOR DE ELCHE
GRADO EN INGENIERÍA ELECTRÓNICA Y AUTOMÁTICA INDUSTRIAL

"MODELADO Y CONTROL DE UNA PLATAFORMA STEWART DE 3 GRADOS DE LIBERTAD"

TRABAJO FIN DE GRADO

Junio - 2026

AUTOR: David Botella Urios
DIRECTORES: Carlos Pérez Vidal
Luis Miguel Jiménez García

AGRADECIMIENTOS

A mi hermano Antonio por su inestimable apoyo.

RESUMEN

El presente trabajo estudia el modelado, simulación y control de una plataforma paralela de tres grados de libertad destinada a estabilizar una bola sobre su superficie. El sistema se basa en una arquitectura mecánica 3-RRS y se desarrolla íntegramente en un entorno de simulación mediante MATLAB/Simulink.

En primer lugar, se obtiene un modelo matemático simplificado del conjunto bola-plataforma en espacio de estados, adecuado para el diseño de controladores en torno al punto de equilibrio. De forma complementaria, se desarrolla un modelo CAD de la plataforma y se exporta a Simscape Multibody, permitiendo comparar ambos enfoques y validar el modelo analítico para pequeñas inclinaciones.

Posteriormente, se modela el sistema de medida, considerando tanto un sensor ideal como un sensor visual simulado con ruido, retardo, cuantización y frecuencia de muestreo limitada. Además, se incorpora un filtro de Kalman para estimar el vector completo de estados a partir de medidas parciales de posición.

Finalmente, se diseñan y evalúan dos estrategias de control: un controlador PD con derivada filtrada y un controlador por realimentación de estados mediante LQR, extendido posteriormente con observador de Kalman. Los resultados muestran que ambas estrategias permiten estabilizar la bola en los escenarios de simulación considerados, destacando el compromiso entre simplicidad de implementación, dependencia del modelo, esfuerzo de control y sensibilidad frente al sistema de medida.

ABSTRACT

This work studies the modeling, simulation and control of a three-degree-of-freedom parallel platform designed to stabilize a ball on its surface. The system is based on a 3-RRS mechanical architecture and is developed entirely in a simulation environment using MATLAB/Simulink.

First, a simplified mathematical model of the ball-platform system is obtained in state-space form, suitable for controller design around the equilibrium point. In addition, a CAD model of the platform is developed and exported to Simscape Multibody, allowing both approaches to be compared and the analytical model to be validated for small inclinations.

Subsequently, the measurement system is modeled, considering both an ideal sensor and a simulated visual sensor with noise, delay, quantization and limited sampling frequency. A Kalman filter is also incorporated to estimate the complete state vector from partial position measurements.

Finally, two control strategies are designed and evaluated: a PD controller with filtered derivative action and a state-feedback controller based on LQR, subsequently extended with a Kalman observer. The results show that both strategies are able to stabilize the ball in the considered simulation scenarios, highlighting the trade-off between implementation simplicity, model dependence, control effort and sensitivity to the measurement system.

ÍNDICE

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

1. INTRODUCCIÓN

El presente Trabajo Fin de Grado aborda un desafío clásico en la ingeniería de control: la estabilización de un objeto sobre una superficie móvil. Específicamente, se desarrolla el **modelado, la simulación y el control de una plataforma paralela de tres grados de libertad (3-DOF)** con una configuración mecánica **3-RRS**, cuya finalidad es mantener una bola en una posición de referencia. Este trabajo integra áreas fundamentales de la ingeniería electrónica como la cinemática de mecanismos, la teoría de control moderno y la visión artificial.

1.1. Motivación

La robótica desempeña en la actualidad un papel fundamental en los procesos de modernización e innovación industrial, constituyendo una herramienta clave para el aumento de la productividad, la mejora de la calidad de los productos y la automatización de procesos complejos. Los manipuladores robóticos permiten sustituir al operario humano en tareas repetitivas, operaciones desarrolladas en entornos peligrosos o tóxicos, procesos de alta precisión y aplicaciones que requieren velocidades de actuación superiores a las alcanzables manualmente [1], [2]. Como consecuencia, el desarrollo de nuevas arquitecturas robóticas y estrategias de control constituye una de las principales líneas de investigación dentro del ámbito de la automatización y la mecatrónica.

Dentro de este contexto, los **mecanismos paralelos** han adquirido una relevancia creciente en aplicaciones industriales y de investigación debido a sus elevadas prestaciones mecánicas, entre las que destacan su alta rigidez estructural, su precisión posicional y su favorable relación entre capacidad de carga y masa móvil en comparación con arquitecturas seriales equivalentes [1]. Estas características los convierten en una solución especialmente atractiva para aplicaciones que demandan movimientos rápidos, precisos y dinámicamente exigentes.

No obstante, el aprovechamiento de las ventajas mecánicas de este tipo de arquitecturas conlleva importantes desafíos desde el punto de vista del modelado y el control. La elevada complejidad cinemática de los manipuladores paralelos, unida al fuerte acoplamiento entre

sus variables de estado y a la presencia de no linealidades inherentes a su geometría, convierte el diseño de estrategias de control robustas en una tarea significativamente más compleja que en sistemas robóticos convencionales [1].

Paralelamente, la expansión de la robótica hacia aplicaciones más complejas y dinámicas ha impulsado la necesidad de integrar sistemas sensoriales avanzados capaces de proporcionar información del entorno en tiempo real. Entre las distintas tecnologías disponibles, los sensores visuales han adquirido una importancia creciente debido a su capacidad para ofrecer una descripción rica y no intrusiva del entorno de trabajo. Su utilización permite implementar **estrategias de control visual** en las que la información extraída de una o varias cámaras se emplea como realimentación para modificar dinámicamente la acción de control del sistema.

La incorporación de visión artificial en sistemas de control introduce, sin embargo, nuevas problemáticas asociadas al procesamiento de la información sensorial, la presencia de ruido de medida, retardos temporales, cuantización y limitaciones de resolución. Estas no idealidades deben ser consideradas adecuadamente durante la fase de diseño para garantizar el correcto funcionamiento del sistema en condiciones realistas.

Dentro de este marco, los sistemas de estabilización de bola sobre superficie inclinable (*ball and plate systems*) representan un banco de pruebas clásico para el estudio y validación de estrategias avanzadas de control [3]. Su interés reside en que combinan un comportamiento dinámico inherentemente inestable con un carácter multivariable y no lineal, constituyendo un problema de referencia en la literatura de control avanzado.

En este contexto, el presente Trabajo Fin de Grado propone el desarrollo de una **plataforma paralela de tres grados de libertad destinada a la estabilización dinámica de una bola sobre su superficie mediante el control de la orientación de la plataforma**. El sistema combina la complejidad mecánica propia de un manipulador paralelo con la problemática dinámica de un sistema subactuado e inestable, incorporando además un sistema de visión artificial simulado como sensor principal de realimentación.

La combinación de estos elementos convierte al sistema propuesto en un escenario de estudio especialmente adecuado para la aplicación y comparación de técnicas modernas de modelado, simulación y control, constituyendo una plataforma de elevado interés tanto

desde el punto de vista académico como tecnológico.

1.2. Justificación del trabajo

El desarrollo experimental de sistemas mecatrónicos complejos requiere habitualmente un elevado esfuerzo económico y temporal derivado de la fabricación de prototipos físicos, la adquisición de instrumentación específica y la iteración sobre diseños mecánicos y electrónicos. Estas limitaciones hacen especialmente atractiva la utilización de herramientas de prototipado virtual y simulación avanzada durante las etapas iniciales de diseño y validación.

No obstante, más allá de las ventajas económicas y logísticas, la simulación constituye una herramienta de especial relevancia en sistemas de elevada complejidad dinámica como el considerado en este trabajo. La interacción entre la dinámica no lineal de la bola, la cinemática acoplada de la plataforma paralela y la integración de un sistema de visión artificial como sensor de realimentación da lugar a un sistema multivariable altamente sensible a perturbaciones, retardos e incertidumbres paramétricas, cuyo análisis experimental resultaría considerablemente más complejo en una fase temprana de desarrollo.

La creación de un **modelo virtual de alta fidelidad** permite, por tanto, abordar de manera controlada el estudio de este tipo de sistemas, ofreciendo ventajas significativas tales como la validación temprana de hipótesis de diseño mecánico, la evaluación reproducible de múltiples estrategias de control, la introducción sistemática de perturbaciones y no idealidades, y el acceso completo a variables internas del sistema difícilmente medibles en un entorno experimental real.

Asimismo, el empleo de modelos virtuales posibilita analizar de forma segura el comportamiento del sistema bajo condiciones extremas o configuraciones potencialmente inestables, facilitando la identificación de limitaciones estructurales, singularidades cinemáticas y restricciones operativas antes de una eventual implementación física.

En este contexto, el presente trabajo adopta una filosofía de **prototipado virtual**, combinando un modelo analítico simplificado del sistema con un modelo multicuerpo no lineal de alta fidelidad implementado en Simulink. Este doble enfoque permite, por una parte,

disponer de modelos matemáticos adecuados para el diseño analítico de controladores y, por otra, validar dichos modelos frente a una representación física más realista del sistema.

En consecuencia, la utilización de un entorno de simulación integral no solo se justifica como una alternativa eficiente al prototipado físico, sino como una metodología técnicamente adecuada para el análisis, validación y optimización del sistema propuesto en las distintas fases de su desarrollo.

1.3. Objetivos del trabajo

El objetivo principal del presente Trabajo Fin de Grado es desarrollar un entorno integral de modelado, simulación y control de una plataforma paralela de tres grados de libertad orientada a la estabilización dinámica de una bola sobre su superficie, empleando para ello herramientas de modelado analítico, diseño CAD y simulación en entorno MATLAB/Simulink.

Con el fin de alcanzar dicho objetivo general, se establecen los siguientes objetivos específicos:

1. **Diseñar y dimensionar geométricamente** una plataforma paralela basada en una arquitectura 3-RRS adecuada para la aplicación de estabilización propuesta, definiendo los parámetros geométricos fundamentales del mecanismo y verificando su viabilidad cinemática.

2. **Obtener el modelo cinemático inverso** del sistema, estableciendo la relación matemática entre la orientación deseada de la plataforma y las variables articulares de los actuadores.

3. **Desarrollar un modelo dinámico simplificado** del sistema bola-plataforma que permita formular el problema de control en un marco analítico adecuado, obteniendo una representación en espacio de estados válida en torno al punto de equilibrio.

4. **Diseñar un modelo tridimensional CAD** del sistema mecánico y exportarlo al entorno Simscape Multibody para la construcción de una planta multicuerpo que permita contrastar el modelo analítico desarrollado.

5. **Validar el modelo analítico** mediante comparación con la respuesta obtenida en la planta multicuerpo implementada en Simscape Multibody, analizando el rango de validez de las hipótesis de linealización y pequeñas inclinaciones.

6. **Modelar el sistema de medida** mediante un sensor ideal y un sensor visual simulado, incorporando efectos no ideales tales como ruido de medida, retardo temporal, cuantización y resolución limitada.

7. **Diseñar e implementar estrategias de control clásicas y modernas** para la estabilización de la bola sobre la superficie móvil, incluyendo un controlador PD con derivada filtrada y un controlador por realimentación de estados mediante LQR.

8. **Incorporar técnicas de estimación de estados** mediante un filtro de Kalman, con el fin de reconstruir el vector de estados a partir de las medidas proporcionadas por el sensor visual simulado.

9. **Evaluar el comportamiento de los controladores implementados** en términos de estabilidad, tiempo de establecimiento, error final, esfuerzo de control y sensibilidad frente a las no idealidades del sistema de medida.

1.4. Alcance y Limitaciones

El presente trabajo **se desarrolla íntegramente en un entorno de simulación**, no constituyendo la implementación física del sistema un objetivo prioritario dentro de su alcance actual. No obstante, la metodología empleada y el nivel de detalle del modelado realizado permitirán que los resultados obtenidos puedan servir como base para una futura materialización experimental del prototipo.

Con el propósito de acotar adecuadamente el problema de estudio y centrar el análisis en la estabilización dinámica de la bola sobre la superficie móvil, se adoptan **una serie de hipótesis simplificadoras** durante el modelado del sistema.

En primer lugar, aunque la arquitectura mecánica seleccionada dispone de tres grados de libertad —correspondientes a dos rotaciones y una traslación vertical de la plataforma—, el movimiento vertical se mantendrá constante durante la operación nominal del sistema.

En consecuencia, el problema de control se reduce al gobierno de las inclinaciones de la plataforma respecto a sus ejes principales, al ser éstas las variables determinantes en la dinámica de la bola sobre la superficie.

Asimismo, el sistema de visión artificial se modelará mediante una representación matemática equivalente, simulando su comportamiento como sensor de medida de posición sin recurrir a técnicas reales de adquisición y procesamiento de imagen. No obstante, se incorporarán al modelo los principales efectos no ideales asociados a este tipo de sensores, con el objetivo de aproximar el comportamiento del sistema a condiciones realistas de operación.

En relación con el contacto entre la bola y la plataforma, se asumirán inicialmente condiciones ideales de rodadura sin deslizamiento, despreciándose fenómenos de segundo orden tales como deformaciones locales, pérdidas de adherencia, irregularidades superficiales o efectos aerodinámicos sobre la bola, salvo en aquellos escenarios específicos en los que dichos efectos sean introducidos deliberadamente como perturbaciones para el análisis de robustez.

Adicionalmente, se considerará comportamiento rígido de los elementos estructurales del mecanismo, despreciándose deformaciones elásticas, holguras mecánicas y fenómenos vibratorios de alta frecuencia que, si bien pueden aparecer en una implementación física real, quedan fuera del alcance del presente estudio.

Estas hipótesis permiten reducir la complejidad del problema abordado manteniendo un nivel de realismo suficiente para el análisis, diseño y validación de las estrategias de control propuestas, garantizando al mismo tiempo la viabilidad técnica y temporal del desarrollo del trabajo.

1.5. Estructura del Proyecto

El presente documento se organiza en diez capítulos principales, cuya estructura se describe a continuación:

- **Capítulo 1. Introducción:** presenta el trabajo, exponiendo la motivación, la justifi-

cación, los objetivos planteados y el alcance general del proyecto.

- **Capítulo 2. Estado del arte:** Ofrece una revisión bibliográfica, analizando los conceptos fundamentales relacionados con la robótica paralela, los sistemas de estabilización tipo bola-superficie, el modelado de sistemas mecánicos, las estrategias de control o el empleo de sensores visuales en lazo cerrado.

- **Capítulo 3. Materiales y métodos:** presenta las herramientas software empleadas durante el desarrollo del trabajo y la metodología seguida para el diseño, modelado, simulación y control del sistema. Además, describe el funcionamiento global de la plataforma, su arquitectura mecánica basada en una configuración 3-RRS y las principales variables de interés para el modelado y control.

- **Capítulo 4. Modelado matemático:** desarrolla el modelo matemático del sistema. En él se obtiene la cinemática inversa de la plataforma, se plantean las hipótesis de modelado y se formula el modelo dinámico simplificado de la planta en espacio de estados, incluyendo el bloque de actuación equivalente empleado posteriormente en las simulaciones.

- **Capítulo 5. Modelado CAD y exportación a Simscape Multibody:** aborda el modelado CAD de la plataforma y su exportación al entorno Simscape Multibody para la obtención de la planta multicuerpo no lineal.

- **Capítulo 6. Validación y análisis comparativo de modelos:** analiza la coherencia entre el modelo matemático en espacio de estados y el modelo multicuerpo de Simscape Multibody, evaluando la respuesta de ambos ante distintas referencias de inclinación.

- **Capítulo 7. Modelado del sistema de medida y realimentación:** estudia primero un sensor ideal de posición y, posteriormente, un sensor visual simulado que incorpora no idealidades como ruido, retardo, cuantización y frecuencia de muestreo limitada. Además, se introduce el filtro de Kalman como estimador de estados.

- **Capítulo 8. Controlador PID:** desarrolla el diseño de un controlador PD con derivada filtrada. Se plantea una estrategia descentralizada por ejes, se realiza la sintonización del controlador y se evalúa su comportamiento tanto con sensor ideal como con sensor visual simulado.

- **Capítulo 9. Control por realimentación de estados:** presenta el control por realimentación de estados. En primer lugar, se diseña una matriz de realimentación mediante LQR suponiendo disponible el estado completo. Posteriormente, se incorpora el sensor visual simulado y el observador de Kalman, analizando el comportamiento del sistema en una situación más próxima a una posible implementación física.

- **Capítulo 10. Conclusiones y líneas de trabajo futuras:** recoge las conclusiones generales del trabajo, resume los principales resultados obtenidos y plantea posibles líneas futuras de desarrollo, entre ellas la mejora de la sintonización de los controladores, la implementación física del prototipo y la ampliación del sistema hacia tareas de seguimiento de trayectorias.

Finalmente, los Anexos incluyen el código fuente empleado durante el desarrollo del proyecto en el entorno MATLAB/Simulink.

2. ESTADO DEL ARTE

2.1. Introducción a los sistemas robóticos

La robótica constituye un campo multidisciplinar que integra conocimientos de mecánica, electrónica, informática y teoría de control con el objetivo de diseñar sistemas capaces de ejecutar tareas de forma autónoma o semiautónoma. En el ámbito de la ingeniería, un robot puede definirse como un sistema electromecánico programable capaz de interactuar con su entorno mediante sensores y actuadores, con el fin de realizar una tarea específica de manera repetitiva, precisa y adaptable [4].

Desde un punto de vista funcional, los robots se caracterizan por la presencia de tres elementos fundamentales: un sistema de percepción, encargado de adquirir información del entorno; un sistema de procesamiento, donde se toman decisiones en función de dicha información; y un sistema de actuación, responsable de ejecutar las acciones físicas necesarias. Esta estructura permite diferenciar a los robots de otros sistemas automatizados más simples.

En este sentido, resulta importante distinguir entre una máquina automática y un robot. Una máquina automática opera siguiendo una secuencia predefinida de acciones, generalmente rígida y poco adaptable a cambios en el entorno. Por el contrario, un robot incorpora capacidades de sensado y toma de decisiones que le permiten modificar su comportamiento en función de las condiciones externas, lo que le confiere un mayor grado de flexibilidad y autonomía. Esta capacidad de adaptación es especialmente relevante en entornos no estructurados o dinámicos, donde las condiciones de operación no pueden preverse completamente.

Los robots pueden clasificarse atendiendo a distintos criterios, entre los que destacan su estructura mecánica, su tipo de aplicación o su nivel de autonomía. Desde el punto de vista estructural, una de las clasificaciones más relevantes distingue entre **robots seriales y robots paralelos**. Esta distinción resulta de especial interés en el contexto del presente trabajo, ya que condiciona de manera directa tanto el comportamiento cinemático como las estrategias de control aplicables.

2.2. Robots seriales y paralelos

Dependiendo de la disposición de sus eslabones y articulaciones, así como en la forma en que se transmite el movimiento desde los actuadores hasta el efector final, los robots manipuladores pueden clasificarse en dos grandes categorías: robots seriales y robots paralelos [1].

2.2.1. Robots seriales

Los robots seriales (véase fig 2.1) están formados por una **cadena cinemática abierta** en la que cada eslabón se conecta con el siguiente mediante una articulación, de forma que existe un único camino desde la base hasta el efector final. Este tipo de configuración es el más extendido en aplicaciones industriales, siendo característico de manipuladores antropomórficos ampliamente utilizados en tareas de ensamblaje, soldadura o manipulación de materiales.

Figura 2.1: Robot serial comercial [5].

Desde el punto de vista cinemático, cada articulación añade un grado de libertad al sistema, lo que permite alcanzar una elevada flexibilidad y un amplio espacio de trabajo. Sin embargo, esta arquitectura presenta ciertas limitaciones inherentes. En particular, la acumulación de errores a lo largo de la cadena cinemática puede reducir la precisión del efector final, y la estructura en voladizo disminuye la rigidez global del sistema, especialmente en configuraciones extendidas.

Entre las principales ventajas de los robots seriales destacan su **gran espacio de trabajo**, su **elevada maniobrabilidad** y la relativa simplicidad de su modelado y control. Como inconvenientes, presentan **menor rigidez estructural**, menor capacidad de carga en relación con su peso y una mayor sensibilidad a errores y perturbaciones.

2.2.2. Robots paralelos

En contraste, los robots paralelos están formados por varias **cadenas cinemáticas cerradas** que conectan la base con la plataforma móvil. En este caso, el efector final está soportado simultáneamente por múltiples ramas, lo que da lugar a una estructura mecánica más rígida y compacta.

Esta configuración permite distribuir las cargas entre varias cadenas, aumentando significativamente la **rigidez estructural** y la **precisión del sistema**. Además, la masa de los elementos móviles suele ser menor, ya que los actuadores pueden situarse en la base, lo que mejora el comportamiento dinámico y permite alcanzar mayores aceleraciones.

No obstante, los robots paralelos presentan también ciertas desventajas. Su espacio de trabajo suele ser más reducido en comparación con los robots seriales, y la presencia de múltiples lazos cinemáticos introduce una mayor complejidad en el análisis, especialmente en la resolución de la cinemática directa. Asimismo, pueden aparecer singularidades en determinadas configuraciones que limitan su operatividad.

2.2.3. Comparación y selección de arquitectura

La elección entre una arquitectura serial o paralela depende en gran medida de los requisitos de la aplicación. Mientras que los robots seriales resultan adecuados para tareas que requieren gran alcance y flexibilidad, los robots paralelos son especialmente adecuados en aplicaciones donde se requiere alta precisión, elevada rigidez y buenas prestaciones dinámicas [1].

En el contexto del presente trabajo, estas características hacen que los manipuladores paralelos resulten especialmente atractivos para el problema planteado. En particular, la

necesidad de controlar con precisión la orientación de una plataforma para estabilizar una bola sobre su superficie favorece el uso de arquitecturas con alta rigidez y respuesta dinámica rápida.

Por este motivo, el sistema desarrollado en **este trabajo se basa en una plataforma paralela** de tres grados de libertad, cuya arquitectura se analizará con mayor detalle en la siguiente sección.

2.3. Arquitectura de robots paralelos

Los robots paralelos pueden presentar una gran variedad de configuraciones mecánicas en función del número de grados de libertad requeridos, la disposición geométrica de sus cadenas cinemáticas y el tipo de articulaciones que componen cada una de ellas.

2.3.1. Tipos de articulaciones

Las cadenas cinemáticas están formadas por eslabones conectados mediante distintos tipos de articulaciones, cada una de las cuales restringe o permite determinados movimientos relativos. Los distintos tipos de articulaciones se muestran en la figura 2.2, siendo las más comunes:

- **Articulación revoluta (R):** permite un movimiento de rotación alrededor de un eje fijo, proporcionando un único grado de libertad. Es ampliamente utilizada en mecanismos rotacionales.
- **Articulación prismática (P):** permite un desplazamiento lineal a lo largo de un eje, siendo habitual en actuadores lineales y sistemas de posicionamiento.
- **Articulación esférica (S):** permite tres grados de libertad de rotación, equivalentes a una rótula. Se emplea frecuentemente en la conexión entre las cadenas cinemáticas y la plataforma móvil.
- **Articulación cilíndrica (C):** permite un movimiento combinado de rotación y traslación a lo largo de un mismo eje, proporcionando dos grados de libertad.

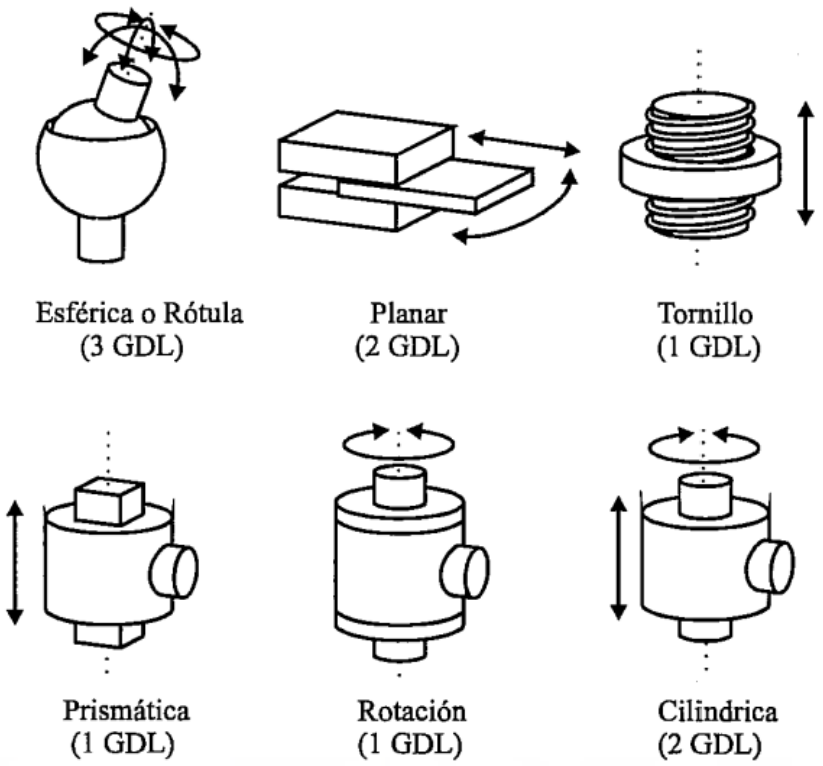


Figura 2.2: Distintos tipos de articulaciones para robots [6].

La combinación de estas articulaciones en cada cadena da lugar a diferentes configuraciones, que suelen representarse mediante una notación simbólica. Por ejemplo, una arquitectura **3-RRS** está formada por tres cadenas cinemáticas, cada una de ellas compuesta por dos articulaciones revolutas seguidas de una articulación esférica.

Por supuesto, esta no es la única configuración posible, existiendo otras arquitecturas alternativas como los mecanismos de tipo **3-RPS**, en los que se combinan articulaciones revolutas, prismáticas y esféricas, también estudiados en la literatura [7].

2.3.2. Grados de libertad en robots paralelos

El número de grados de libertad (DOF, *Degrees of Freedom*) de un robot paralelo determina la capacidad de movimiento de su plataforma móvil. En general, un cuerpo rígido en el espacio tridimensional posee seis grados de libertad: tres traslaciones y tres rotaciones. Sin embargo, muchas aplicaciones requieren únicamente un subconjunto de estos movimientos.

En este sentido, es posible clasificar los robots paralelos en función de los grados de libertad de su plataforma:

- **Plataformas de 1 DOF:** empleadas en sistemas simples como el *ball and beam*, donde únicamente se controla la inclinación en un eje.
- **Plataformas de 2 DOF:** permiten controlar la orientación en dos ejes, como en los sistemas *ball and plate* tradicionales.
- **Plataformas de 3 DOF:** pueden proporcionar dos rotaciones y una traslación, o tres traslaciones puras, dependiendo de la arquitectura. Este tipo de plataformas representa un compromiso entre complejidad y capacidad de control. Un ejemplo representativo es el **robot Delta** (figura 2.3), utilizado en aplicaciones de alta velocidad para tareas de manipulación.

Figura 2.3: Robot delta industrial con 3-DOF [5].

- **Plataformas de 6 DOF:** permiten el control completo de la posición y orientación en el espacio. La **plataforma Stewart** (figura 2.4) constituye el ejemplo más conocido, ampliamente utilizada en simuladores de movimiento y sistemas de posicionamiento de alta precisión.

La elección del número de grados de libertad depende de los requisitos de la aplicación, siendo habitual limitar el movimiento a aquellos grados estrictamente necesarios para simplificar el diseño y el control del sistema.

Figura 2.4: Sistema de movimiento de 6-DOF basado en una plataforma Stewart industrial [8].

2.3.3. Consideraciones sobre diseño y aplicación

El diseño de un robot paralelo implica un compromiso entre varios factores, entre los que destacan el espacio de trabajo, la rigidez estructural, la complejidad cinemática y la facilidad de control. En general, arquitecturas con mayor número de grados de libertad ofrecen mayor versatilidad, pero también incrementan la complejidad del sistema.

En este contexto, conviene precisar que, aunque la denominación *plataforma Stewart* suele asociarse de forma estricta a plataformas paralelas de seis grados de libertad, en esta memoria se utiliza de forma amplia para situar el sistema dentro de la familia de mecanismos paralelos de plataforma móvil. La configuración concreta desarrollada corresponde a una **arquitectura de tipo 3-RRS que proporciona tres grados de libertad** a la plataforma móvil: dos rotaciones (ϕ, θ) y una traslación vertical z. En este trabajo se emplean principalmente los grados de libertad asociados a la orientación de la plataforma, lo que permite simplificar el problema de control manteniendo un comportamiento dinámico suficientemente rico para el estudio propuesto.

2.4. Sistemas de estabilización tipo bola-superficie

Los sistemas de estabilización de bola sobre superficie inclinable constituyen un problema clásico en la ingeniería de control. El objetivo fundamental consiste en controlar la

posición de una bola que se desplaza libremente sobre una superficie plana mediante la modificación de la **inclinación de dicha superficie**. Para ello, se actúa sobre uno o varios grados de libertad que permiten variar la orientación del plano, generando componentes de la fuerza gravitatoria que inducen el movimiento de la bola.

2.4.1. Clasificación según grados de libertad

En función del número de grados de libertad, visto en la sección 2.3.2, los sistemas pueden clasificarse en:

- **Sistemas de 1 DOF (*Ball and Beam*, Fig. 2.5a):** en estos sistemas la bola se desplaza a lo largo de una única dirección, controlándose la inclinación de una viga. Constituyen el caso más simple y se utilizan frecuentemente como introducción al control de sistemas inestables.
- **Sistemas de 2 DOF (*Ball and Plate* clásico, Fig. 2.5b):** permiten controlar la posición de la bola en dos dimensiones mediante la inclinación de una plataforma en dos ejes. Este tipo de sistemas presenta un comportamiento multivariable (MIMO) y una mayor complejidad de control.
- **Sistemas avanzados de múltiples grados de libertad:** en configuraciones más complejas, la plataforma puede estar soportada por mecanismos paralelos que permiten controlar no solo la orientación, sino también la posición en el espacio. Estas arquitecturas introducen una mayor complejidad mecánica y cinemática, pero ofrecen mayores prestaciones dinámicas y precisión.

(a) Sistema *Ball and Beam*. (b) Sistema *Ball and Plate*.

Figura 2.5: Comparativa de sistemas de estabilización de bola según grados de libertad. Imágenes adaptadas de [8].

2.4.2. Problemática de control

El interés de los sistemas *Ball and Plate* radica en la combinación de diversas características que los convierten en un problema de control especialmente exigente. En primer lugar, presentan una **inestabilidad inherente**, ya que, en ausencia de control, la bola tiende a desplazarse hasta abandonar la superficie. Asimismo, se trata de un **sistema no lineal**, cuya dinámica depende de funciones trigonométricas de los ángulos de inclinación de la plataforma.

Por otro lado, en configuraciones de dos o más grados de libertad, el sistema presenta un **comportamiento multivariable** con variables **fuertemente acopladas**, lo que incrementa la complejidad del diseño de controladores. A ello se añade su carácter **subactuado**, dado que la posición de la bola no se controla de forma directa, sino mediante la inclinación de la superficie. Finalmente, el sistema es altamente **sensible a perturbaciones** y condiciones iniciales, de modo que pequeñas variaciones pueden dar lugar a desviaciones significativas en la trayectoria de la bola.

Estas características hacen que los sistemas *Ball and Plate* constituyan un banco de pruebas idóneo para la evaluación y comparación de distintas estrategias de control, tanto clásicas como modernas.

En el ámbito académico, numerosos trabajos han abordado versiones simplificadas de este problema, como el sistema *Ball and Beam*, empleado habitualmente como introducción al control de sistemas inestables [9]. Asimismo, existen múltiples Trabajos Fin de Grado/Máster centrados en plataformas *Ball and Plate* de dos grados de libertad, en los que se analizan distintas estrategias de control e implementación [10], [11], [12], [13], [14], [15].

En contraste, el análisis de plataformas paralelas de mayor complejidad, como las configuraciones de tres grados de libertad basadas en arquitecturas tipo 3-RRS, ha sido menos elegido por los estudiantes para elaborar su tesis, siendo mayor la complejidad cinemática y dinámica del sistema [16].

2.5. Modelado de sistemas mecánicos

El modelado matemático de sistemas mecánicos constituye una etapa fundamental en el análisis y diseño de sistemas de control, ya que permite describir de forma cuantitativa la relación entre las variables de entrada, estado y salida del sistema.

2.5.1. Modelado cinemático

El **modelado cinemático** de robots paralelos se centra en la relación geométrica entre las variables articulares y la posición y orientación de la plataforma móvil. A diferencia de los robots seriales, donde la cinemática directa se obtiene de forma relativamente sencilla, en los robots paralelos este problema resulta considerablemente más complejo debido a la presencia de múltiples cadenas cinemáticas cerradas.

En este sentido, se distinguen dos problemas fundamentales:

- **Cinemática directa:** consiste en determinar la posición y orientación de la plataforma a partir de las variables articulares. En robots paralelos, este problema suele implicar la resolución de sistemas no lineales de ecuaciones, pudiendo presentar múltiples soluciones o incluso configuraciones singulares.
- **Cinemática inversa:** consiste en determinar las variables articulares necesarias para alcanzar una posición y orientación deseadas de la plataforma. En aplicaciones como la considerada en este trabajo, la cinemática inversa adquiere un papel fundamental, ya que permite transformar las referencias de orientación de la plataforma en señales de control para los actuadores.

El problema cinemático inverso es, en general, más sencillo de resolver que el directo [1]. Si bien existen artículos científicos que tratan de facilitar la resolución de la cinemática directa para el manipulador paralelo con 3-RRS [17], la mayoría de investigaciones utilizan el modelo inverso [3], [18], [19], [20] que será el utilizado en el presente proyecto.

2.5.2. Modelado dinámico

El **modelado dinámico** describe la evolución temporal del sistema bajo la acción de fuerzas y momentos. En el caso de los sistemas *Ball and Plate*, la dinámica está gobernada por la interacción entre la bola y la superficie inclinada, así como por la dinámica de la propia plataforma.

La dinámica de la bola puede modelarse considerando condiciones de rodadura sin deslizamiento, lo que conduce a ecuaciones de movimiento no lineales en las que la aceleración depende de la inclinación de la superficie. Por su parte, la dinámica de la plataforma depende de la arquitectura mecánica empleada y de las características de los actuadores.

En robots paralelos, la obtención de un modelo dinámico completo resulta compleja debido al acoplamiento entre las cadenas cinemáticas y a la presencia de restricciones geométricas. Por este motivo, en muchos trabajos se opta por utilizar modelos simplificados o aproximaciones que capturan los efectos dominantes del sistema.

2.5.3. Acoplamiento de variables y no linealidad

Una de las principales dificultades en el modelado de estos sistemas es la presencia de variables **fuertemente acopladas**. En el caso del sistema propuesto, el movimiento de la bola en cada dirección depende simultáneamente de la inclinación en ambos ejes, lo que da lugar a un **sistema multivariable (MIMO)**.

Asimismo, las ecuaciones que describen el comportamiento del sistema son inherentemente **no lineales**, debido a la presencia de funciones trigonométricas y a la dependencia de las fuerzas con la orientación de la plataforma. Este carácter no lineal dificulta el análisis y el diseño de controladores, especialmente cuando se requieren prestaciones en un amplio rango de operación.

2.5.4. Simplificación y linealización del sistema

En la práctica, es habitual introducir ciertas **hipótesis simplificadoras** para reducir la complejidad del modelo, tales como la consideración de pequeños ángulos de inclinación, la ausencia de deslizamiento entre la bola y la superficie, o el comportamiento rígido de los elementos mecánicos.

De igual forma, es habitual recurrir a **técnicas de linealización** del modelo dinámico con el objetivo de facilitar el diseño de controladores. Este proceso consiste en aproximar el comportamiento del sistema mediante un modelo lineal válido en torno a un punto de operación, generalmente asociado a pequeñas inclinaciones de la plataforma.

Estas simplificaciones permiten obtener modelos manejables desde el punto de vista analítico, manteniendo al mismo tiempo una representación suficientemente precisa del comportamiento del sistema para el diseño y validación de estrategias de control. Se usarán para obtener representaciones en espacio de estados que resultan especialmente adecuadas para el diseño de controladores modernos; teniendo en consideración que el uso de estas aproximaciones **introduce limitaciones**, ya que el modelo lineal deja de ser válido para desviaciones grandes respecto al punto de operación.

2.5.5. Implementación en entornos de simulación

El uso de herramientas de simulación como MATLAB/Simulink constituye una práctica ampliamente extendida en el modelado y análisis de sistemas dinámicos. En estos entornos, es posible implementar tanto modelos matemáticos analíticos como modelos físicos basados en simulación multicuerpo.

En particular, el entorno **Simscape Multibody** permite modelar sistemas mecánicos complejos a partir de su geometría y propiedades físicas, proporcionando una representación no lineal de alta fidelidad. Esta aproximación resulta especialmente útil en el caso de robots paralelos, donde la obtención de modelos analíticos completos puede resultar inviable.

La combinación de modelos analíticos simplificados y modelos multicuerpo permite aprovechar las ventajas de ambos enfoques: por un lado, la facilidad de diseño de controladores

basada en modelos lineales y, por otro, la validación del comportamiento del sistema en condiciones más realistas.

2.6. Estrategias de control

El diseño de estrategias de control para sistemas mecánicos, como los robots paralelos o los sistemas de estabilización de bola, constituye un aspecto fundamental para garantizar el cumplimiento de los objetivos de estabilidad, precisión y robustez.

Para el sistema de tres grados de libertad con articulaciones RRS, considerado en este proyecto, caracterizado por su comportamiento no lineal, su naturaleza multivariable y la presencia de perturbaciones e incertidumbres; resulta necesario evaluar distintas alternativas de control con el fin de determinar la solución más adecuada.

2.6.1. Control clásico

El control clásico, y en particular el controlador proporcional-integral-derivativo (**PID**), constituye una de las estrategias más ampliamente utilizadas en aplicaciones industriales debido a su simplicidad, robustez y facilidad de implementación.

El controlador PID permite regular sistemas dinámicos a partir de la combinación de tres acciones de control: proporcional, integral y derivativa. Su popularidad radica en que, pese a su estructura relativamente simple, es capaz de proporcionar un comportamiento satisfactorio en una gran variedad de sistemas.

La sintonización de los parámetros del controlador PID puede realizarse mediante distintos métodos, entre los que destacan:

- Técnicas basadas en el **lugar de las raíces**
- Métodos en el dominio frecuencial, como el **diagrama de Bode**
- Reglas empíricas, como el método de **Ziegler–Nichols**

- Herramientas de ajuste automático (*autotuning*) disponibles en entornos como MATLAB

En este contexto, MATLAB proporciona herramientas específicas para el diseño y ajuste de controladores clásicos, como **Control System Designer**, orientada principalmente a sistemas de una sola entrada y una sola salida (SISO).

No obstante, aunque los controladores PID pueden extenderse a sistemas multivariables mediante estructuras descentralizadas, su aplicación en sistemas fuertemente acoplados puede presentar limitaciones en términos de rendimiento y robustez.

2.6.2. Control en espacio de estados

El control en espacio de estados constituye una alternativa más general para el tratamiento de sistemas dinámicos complejos, especialmente aquellos con múltiples entradas y salidas (**sistemas MIMO**, *Multiple Input Multiple Output*).

En este enfoque, el sistema se describe mediante un conjunto de ecuaciones diferenciales que relacionan el vector de estado, las entradas y las salidas del sistema. Esta representación permite modelar de forma explícita el acoplamiento entre variables y facilita el diseño de controladores multivariables.

El uso de modelos en espacio de estados resulta especialmente adecuado en sistemas como el considerado en este trabajo, donde la dinámica presenta un fuerte acoplamiento entre las variables y donde es necesario actuar de forma coordinada sobre múltiples actuadores.

En el entorno MATLAB, herramientas como **Control System Tuner** permiten diseñar y ajustar controladores en sistemas MIMO de forma sistemática, facilitando la incorporación de especificaciones de diseño y la evaluación del comportamiento del sistema.

2.6.3. Métodos de control avanzado

Además de las técnicas clásicas y del control en espacio de estados, la literatura recoge un amplio conjunto de metodologías de control avanzado aplicadas a sistemas no lineales y

multivariables.

Entre estas técnicas se incluyen, entre otras:

- **Control predictivo basado en modelo (MPC)**
- **Control difuso (Fuzzy Control)**
- **Control óptimo**
- **Control robusto**
- **Control basado en modos deslizantes (SMC)**

Estas estrategias permiten abordar de forma más eficaz problemas asociados a no linealidades, restricciones, incertidumbres y perturbaciones externas, aunque generalmente a costa de una mayor complejidad computacional y de implementación.

2.6.4. Comparación de los métodos de control

En la literatura reciente, diversos trabajos han abordado la implementación y comparación de estrategias de control en sistemas *Ball and Plate* de 2-DOF, tanto en entornos simulados como experimentales. En particular, algunos trabajos académicos a nivel de grado han analizado el comportamiento de distintos controladores —tales como PID y técnicas avanzadas— sobre plataformas de dos grados de libertad, proporcionando resultados comparativos en términos de estabilidad y precisión [13], [14].

Asimismo, existen estudios publicados en la literatura científica que abordan el control de plataformas paralelas de características similares a la analizada en este trabajo (3-RRS) evaluando distintas estrategias de control sobre modelos no lineales [21], [22]. No obstante, estos trabajos suelen centrarse en enfoques específicos o presentan un alcance limitado en cuanto a modelado y análisis comparativo.

Estas contribuciones ponen de manifiesto el interés existente en el problema, así como la necesidad de seguir explorando metodologías de modelado y control en sistemas más complejos, como el considerado en este trabajo.

2.7. Sistemas de visión artificial en control

La incorporación de sistemas de visión artificial en lazo de control ha experimentado un notable crecimiento en las últimas décadas, impulsada por la necesidad de dotar a los sistemas robóticos de una mayor capacidad de percepción e interacción con entornos dinámicos y no estructurados. A diferencia de los sensores tradicionales, los sistemas de visión permiten obtener una descripción rica del entorno de forma no intrusiva, proporcionando información espacial relevante para el control.

En este contexto, el término **control visual** hace referencia a aquellas estrategias de control en las que la información utilizada para la realimentación del sistema se obtiene a partir de imágenes captadas por una o varias cámaras. Esta información se procesa para estimar variables de interés, como la posición o velocidad de un objeto, que son posteriormente utilizadas para generar las acciones de control.

2.7.1. Sensores empleados

En sistemas de estabilización tipo *Ball and Plate*, la estimación de la posición de la bola puede realizarse mediante distintos tipos de sensores, siendo los más habituales los sistemas de visión artificial y los dispositivos táctiles.

El uso de **cámaras** constituye una de las soluciones más extendidas en este tipo de aplicaciones, permitiendo obtener la posición de la bola mediante técnicas de procesamiento de imagen. Diversos trabajos han adoptado este enfoque, empleando sistemas de visión para la estimación de la posición en tiempo real [11], [15], [20]. Su principal ventaja radica en la capacidad de capturar información bidimensional completa del sistema de forma no intrusiva, lo que permite una mayor flexibilidad en el diseño del sistema y su aplicación en entornos dinámicos.

Por otro lado, existen implementaciones basadas en **pantallas táctiles**, en las que la posición de la bola se obtiene directamente a partir del punto de contacto con la superficie. Este enfoque ha sido utilizado en diversos trabajos académicos [10], [12], [13], [22], simplificando el problema de percepción a costa de limitar la generalidad del sistema y su

aplicabilidad a escenarios más realistas.

La elección del tipo de sensor depende de factores como la precisión requerida, la complejidad del sistema y el entorno de aplicación. En particular, el uso de cámaras resulta especialmente adecuado cuando se pretende reproducir condiciones cercanas a aplicaciones reales de robótica y control visual.

2.7.2. Algoritmos de procesamiento en visión artificial

La estimación de la posición de objetos mediante sistemas de visión artificial se basa en el uso de algoritmos de procesamiento de imagen, cuya complejidad puede variar en función de los requisitos de precisión y de las condiciones del entorno. En el contexto de sistemas *Ball and Plate*, los métodos más habituales incluyen:

- **Detección por color:** consiste en identificar el objeto de interés a partir de su color característico, utilizando transformaciones en el espacio de color (por ejemplo, RGB o HSV) y técnicas de umbralización. Este enfoque ha sido empleado en distintos trabajos por su sencillez y eficiencia computacional, lo que lo hace especialmente adecuado para aplicaciones en tiempo real [20].
- **Detección por formas:** se basa en la identificación de características geométricas del objeto, como su forma circular en el caso de una bola. Técnicas como la transformada de Hough permiten detectar circunferencias en imágenes, proporcionando una estimación robusta de la posición incluso en presencia de variaciones en las condiciones de iluminación. Este enfoque ha sido utilizado en sistemas basados en visión para el seguimiento de la bola [11].
- **Métodos basados en aprendizaje automático:** incluyen el uso de técnicas más avanzadas como redes neuronales o algoritmos de visión profunda (*deep learning*), capaces de identificar y seguir objetos en entornos complejos. Si bien existen artículos científicos que lo abordan [17], su implementación suele requerir mayores recursos computacionales y conjuntos de datos de entrenamiento.

2.7.3. Modelado del sistema de visión

En entornos de simulación, los algoritmos mencionados pueden modelarse de forma simplificada mediante bloques matemáticos equivalentes, reproduciendo su comportamiento sin necesidad de implementar el procesamiento completo de imagen. Esto permite evaluar el impacto de distintas estrategias de percepción en el lazo de control de forma flexible y controlada.

Dicho modelo puede incorporar distintos efectos característicos de los sistemas reales, tales como el **ruido de medida**, habitualmente representado mediante distribuciones gaussianas, el **retardo temporal** asociado a los procesos de adquisición y procesamiento de la imagen, así como los efectos derivados de la **cuantización** y las limitaciones de **resolución** espacial propias del sensor.

La inclusión de estos fenómenos permite evaluar el comportamiento del sistema de control en condiciones más realistas, aproximando el entorno de simulación a un posible escenario de implementación física.

2.8. Niveles de implementación

El desarrollo de sistemas de control puede abordarse desde distintos niveles de implementación, que abarcan desde la simulación completamente virtual hasta la implementación física del sistema. La elección del nivel adecuado depende de factores como la complejidad del sistema, los recursos disponibles y los objetivos del estudio.

En términos generales, pueden distinguirse tres niveles principales de implementación:

2.8.1. Simulación pura

La **simulación pura** consiste en el desarrollo completo del sistema dentro de un entorno virtual, sin la existencia de un prototipo físico real. En este enfoque, tanto la planta como los sensores y actuadores son modelados mediante representaciones matemáticas o físicas en entornos de simulación como MATLAB/Simulink.

Este nivel de implementación permite un control total sobre las variables del sistema, facilitando la validación de modelos, la evaluación de distintas estrategias de control y el análisis del comportamiento bajo condiciones ideales o perturbadas. Además, ofrece una elevada flexibilidad para modificar parámetros y explorar distintos escenarios sin incurrir en costes adicionales.

No obstante, la simulación pura presenta como principal limitación la dependencia de la fidelidad del modelo, pudiendo existir discrepancias entre el comportamiento simulado y el sistema real.

2.8.2. Sistemas híbridos

En un nivel intermedio se encuentran los **sistemas híbridos**, en los que coexisten elementos reales y simulados. En muchas implementaciones prácticas de sistemas *ball and plate*, este enfoque consiste en disponer de una **planta física real**, mientras que el algoritmo de control se ejecuta en un ordenador externo, utilizando herramientas como MATLAB o Python.

En este tipo de configuraciones, el ordenador se encarga del cálculo de la acción de control a partir de las medidas del sistema, mientras que dispositivos intermedios, como placas de desarrollo (por ejemplo, Arduino u otros microcontroladores), actúan como interfaz de potencia para el accionamiento de los actuadores.

Este enfoque permite combinar la flexibilidad del desarrollo en entorno software con el comportamiento real de la planta, facilitando la validación experimental de los controladores sin necesidad de implementar sistemas embebidos completos desde el inicio.

En variantes más avanzadas, este tipo de arquitecturas puede evolucionar hacia esquemas de tipo **Hardware-in-the-Loop (HIL)**, en los que determinados subsistemas se simulan en tiempo real mientras otros se implementan físicamente.

2.8.3. Implementación física y sistemas embebidos

El nivel más completo corresponde a la **implementación física**, en la que tanto la planta como el sistema de control, los sensores y los actuadores operan en un entorno real.

En este contexto, es habitual el uso de **sistemas embebidos**, en los que el algoritmo de control se ejecuta directamente sobre hardware dedicado, como microcontroladores o sistemas de procesamiento en tiempo real. Este enfoque permite reducir la latencia, mejorar la eficiencia del sistema y lograr una integración completa del control dentro del dispositivo.

Sin embargo, la implementación embebida introduce nuevas dificultades, tales como limitaciones de capacidad de cálculo, restricciones de memoria, gestión del tiempo real y mayor complejidad en el desarrollo del software.

Además, este nivel de implementación implica mayores costes económicos, tiempos de desarrollo más elevados y menor flexibilidad para modificar el sistema una vez construido.

2.8.4. Justificación del enfoque adoptado

En el contexto del presente trabajo, se opta por un enfoque basado en **simulación pura**, dado que el objetivo principal es el modelado, análisis y diseño de estrategias de control para un sistema mecánico complejo.

Este enfoque permite desarrollar tanto modelos analíticos como modelos multicuerpo de alta fidelidad, evaluar distintas estrategias de control en condiciones reproducibles e introducir de forma controlada perturbaciones y no idealidades propias de sistemas reales. Asimismo, la metodología empleada facilita una posible transición futura hacia una implementación física, al disponer de una base sólida de modelado y validación.

2.9. Limitaciones del estado del arte y aportación del trabajo

El análisis de la literatura existente pone de manifiesto el notable interés que ha suscitado el problema de la estabilización de bola sobre superficies inclinables, así como el desarrollo de plataformas paralelas para aplicaciones de control preciso. No obstante, dicho análisis también revela una serie de limitaciones recurrentes en los trabajos previos que justifican el planteamiento del presente Trabajo de Fin de Grado.

En primer lugar, una gran parte de los trabajos existentes, especialmente en el ámbito académico a nivel de grado, se centran en sistemas *Ball and Plate* de **dos grados de libertad**, en los que la plataforma se acciona mediante mecanismos relativamente simples. Aunque estos sistemas resultan adecuados como banco de pruebas inicial, presentan una complejidad mecánica y cinemática limitada en comparación con arquitecturas paralelas de mayor sofisticación.

Por otro lado, en la literatura científica pueden encontrarse estudios que abordan el control de plataformas paralelas más complejas, incluyendo configuraciones de **tres grados de libertad** similares a la considerada en este trabajo. Sin embargo, estos trabajos suelen centrarse en aspectos específicos, como el diseño de un controlador concreto o el análisis dinámico del sistema, sin abordar de forma integrada el modelado completo, la simulación de alta fidelidad y la comparación sistemática entre distintas estrategias de control.

Asimismo, muchos de los trabajos existentes basan su validación en implementaciones físicas, lo que, si bien aporta realismo, introduce **limitaciones importantes** en términos de flexibilidad experimental. En particular, la modificación de parámetros del sistema, la introducción de perturbaciones controladas o la evaluación de múltiples estrategias de control pueden resultar costosas y difíciles de implementar en un entorno físico real.

En este contexto, el presente trabajo propone un enfoque basado en **simulación pura de alta fidelidad**, que permite superar en gran medida estas limitaciones. La utilización de entornos como MATLAB/Simulink y Simscape Multibody posibilita la construcción de una planta no lineal detallada, en la que se consideran propiedades físicas realistas como masas, inercias y efectos gravitatorios.

Una de las principales aportaciones del trabajo reside en la **combinación de modelos analíticos y modelos multicuerpo**, lo que permite disponer de una doble representación del sistema. Por un lado, el modelo matemático simplificado facilita el diseño de controladores y el análisis teórico; por otro, el modelo CAD exportado a Simscape proporciona una representación física más realista del comportamiento del sistema. La comparación entre ambos modelos permite validar la consistencia del modelado y obtener una planta de referencia bien definida.

Además, el uso de un modelo CAD detallado introduce una importante ventaja en términos de **escalabilidad y proyección futura del sistema**. Al disponer de un diseño geométrico completo, el sistema puede ser potencialmente fabricado mediante técnicas de prototipado rápido, como la **impresión 3D**, facilitando una eventual transición hacia una implementación física real. De este modo, el trabajo no se limita a un estudio teórico, sino que establece las bases para un desarrollo experimental posterior.

3. MATERIALES Y MÉTODOS

3.1. Introducción

En este capítulo se presentan los materiales y la metodología empleados para el desarrollo del presente trabajo. Dado que el proyecto se ha realizado íntegramente en un **entorno de simulación**, los materiales utilizados no corresponden a componentes físicos reales, sino a herramientas software destinadas al diseño mecánico, el modelado matemático, la simulación dinámica y el diseño de estrategias de control.

El capítulo se estructura en tres bloques principales. En primer lugar, se describen las herramientas empleadas durante el desarrollo del proyecto, indicando el papel que ha desempeñado cada una de ellas dentro del flujo de trabajo. En segundo lugar, se expone la metodología seguida, organizada como una secuencia ordenada de etapas que abarcan desde el diseño conceptual de la plataforma hasta la validación de los modelos y controladores implementados. Finalmente, se describe el sistema objeto de estudio, incluyendo su funcionamiento global, la arquitectura mecánica de la plataforma paralela y las principales variables consideradas para el modelado y control.

Este planteamiento permite contextualizar el desarrollo del sistema antes de abordar en detalle los modelos matemáticos, la implementación multicuerpo, el sistema de medida y las estrategias de control que se presentan en los capítulos posteriores.

3.2. Materiales empleados

En la Tabla 3.1 se recogen las principales herramientas software empleadas durante el desarrollo del trabajo, indicando su versión y el uso principal que han tenido dentro del proyecto.

Tabla 3.1: Herramientas software empleadas en el desarrollo del trabajo.

Herramienta	**Versión/Modelo**	**Uso principal**
Autodesk Inventor	2027	Diseño CAD tridimensional de la plataforma
Autodesk AutoCAD	2026	Bocetos, esquemas y diagramas auxiliares
MATLAB	R2025b	Cálculo matemático, scripts y análisis de resultados
Simulink	R2025b	Simulación e integración del sistema completo
Simscape Multibody	R2025b	Modelado multicuerpo no lineal de la plataforma
Contact Forces Library	R2025b	Modelado del contacto bola-plataforma
Control System Toolbox	R2025b	Análisis y diseño de sistemas de control
TeXstudio	4.9.3	Redacción y maquetación de la memoria mediante LaTeX
Obsidian	v1.8.10	Toma de notas y organización de contenidos
ChatGPT	GPT-5.5	Apoyo en redacción, revisión y scripts

Las herramientas de Autodesk se han empleado principalmente durante las fases de diseño geométrico y representación mecánica del sistema. En particular, **Autodesk Inventor** ha permitido desarrollar el modelo CAD tridimensional de la plataforma, mientras que **Autodesk AutoCAD** se ha utilizado como apoyo para la elaboración de bocetos, esquemas y diagramas auxiliares. Estas herramientas han facilitado la definición inicial de la geometría del mecanismo y la posterior construcción del modelo empleado en simulación.

El entorno principal de cálculo y simulación ha sido **MATLAB/Simulink**. MATLAB se ha utilizado para la definición de parámetros, el desarrollo de modelos matemáticos, la implementación de scripts auxiliares, el diseño de controladores y el análisis de resultados. Por su parte, Simulink ha permitido integrar de forma modular la planta, los sensores, los estimadores, los controladores y las perturbaciones consideradas, proporcionando un entorno adecuado para el estudio del comportamiento dinámico del sistema en lazo abierto y en lazo cerrado.

Dentro del entorno MATLAB/Simulink se han utilizado librerías y bloques específicos de especial relevancia para el desarrollo del trabajo. **Simscape Multibody** se ha empleado para construir el modelo multicuerpo no lineal de la plataforma paralela, representando el

sistema mediante cuerpos rígidos, articulaciones y restricciones mecánicas. Asimismo, se han utilizado bloques de contacto, como **Simscape Multibody Contact Forces Library** [23], para modelar la interacción entre la bola y la superficie de la plataforma. Por otro lado, **Control System Toolbox** ha servido como apoyo para el análisis de sistemas dinámicos y el diseño de estrategias de control clásicas y modernas.

Las herramientas de Autodesk y MATLAB/Simulink, así como sus correspondientes librerías, se han utilizado mediante licencias educativas, empleadas exclusivamente con fines académicos y no comerciales, en el marco del desarrollo del presente Trabajo Fin de Grado.

Además de las herramientas técnicas anteriores, se han empleado recursos auxiliares para la organización y elaboración de la memoria. TeXstudio se ha utilizado para la redacción y maquetación del documento final mediante **LaTeX**, permitiendo estructurar ecuaciones, figuras, tablas y referencias bibliográficas. Obsidian se ha empleado como herramienta de toma de notas, organización de ideas y estructuración preliminar de contenidos.

Finalmente, se han utilizado herramientas de **inteligencia artificial** como apoyo auxiliar en tareas de redacción, revisión ortográfica y estilística, conversión de notas preliminares a formato LaTeX y ayuda puntual en la generación o depuración de scripts. En todos los casos, el contenido técnico, los resultados obtenidos y las decisiones de diseño han sido **revisados, adaptados y validados por el autor.**

3.3. Metodología de desarrollo

La metodología seguida para el desarrollo del presente trabajo se estructura en una secuencia ordenada de etapas que abarcan el diseño mecánico, el modelado matemático, la simulación física, el modelado del sistema de medida y el diseño de estrategias de control. Este planteamiento permite avanzar de forma progresiva desde una descripción conceptual del sistema hasta su implementación completa en un entorno de simulación.

El proceso de desarrollo adoptado se resume en las siguientes fases:

1. **Diseño conceptual y dimensionamiento geométrico** de la plataforma paralela,

definiendo la arquitectura mecánica 3-RRS y los parámetros fundamentales del mecanismo.

2. **Desarrollo de los modelos matemáticos** necesarios para describir la cinemática y la dinámica del sistema, obteniendo formulaciones simplificadas aptas para el diseño analítico de controladores.

3. **Diseño CAD tridimensional** de la plataforma mecánica a partir de la geometría previamente definida.

4. **Exportación del modelo CAD** al entorno Simscape Multibody, con objeto de construir una planta multicuerpo no lineal que permita representar de forma más realista el comportamiento físico del sistema.

5. **Validación del modelo matemático** mediante la comparación de su respuesta con la obtenida en el modelo multicuerpo, evaluando la coherencia entre ambos enfoques bajo distintas referencias de inclinación.

6. **Modelado del sistema de medida**, considerando inicialmente un sensor ideal y, posteriormente, un sensor visual simulado con no idealidades asociadas a ruido, retardo, cuantización y frecuencia de muestreo limitada.

7. **Incorporación de técnicas de estimación de estados** mediante un filtro de Kalman, con el fin de reconstruir el vector completo de estados a partir de medidas parciales de posición.

8. **Diseño e implementación de estrategias de control**, incluyendo técnicas clásicas y modernas aplicadas al sistema modelado.

9. **Integración global del sistema** en el entorno MATLAB/Simulink, unificando la planta, los sensores, los estimadores, los controladores y las perturbaciones dentro de un modelo completo de simulación.

10. **Validación y análisis comparativo de resultados**, evaluando el comportamiento del sistema bajo distintos escenarios operativos, condiciones iniciales, perturbaciones y referencias.

3.4. Descripción del sistema

Una vez definidos los recursos software empleados y la metodología seguida durante el desarrollo del trabajo, en este apartado se describe el sistema objeto de estudio. Esta descripción permite establecer una visión global de la plataforma bola-superficie y sirve como punto de partida para el posterior desarrollo del modelado, la simulación y el control del sistema.

3.4.1. Funcionamiento global del sistema

El sistema desarrollado en el presente trabajo tiene como objetivo la estabilización y el seguimiento de trayectorias de una bola sobre una superficie móvil, constituyendo una aplicación del clásico problema *Ball and Plate*. Para ello, se emplea una **plataforma paralela de tres grados de libertad** cuya orientación puede modificarse con el fin de controlar la posición de la bola sobre su superficie.

El principio de funcionamiento del sistema se basa en la **modificación controlada de la inclinación de la plataforma**, generando componentes de la fuerza gravitatoria que inducen el movimiento de la bola en la dirección deseada. De este modo, mediante la actuación coordinada sobre los grados de libertad rotacionales de la plataforma, es posible regular la posición de la bola o forzarla a seguir trayectorias predefinidas sobre el plano.

Desde el punto de vista del control, el sistema se estructura como un **lazo cerrado** de realimentación, como el mostrado en la figura 3.1. En este esquema, una *señal de referencia* define la posición o trayectoria deseada de la bola sobre la superficie. El *controlador* procesa el error entre esta referencia y la posición actual de la bola medida por el *sensor*, calculando la inclinación necesaria que debe adoptar el plato. Estas consignas de inclinación no pueden aplicarse directamente a los motores. Por tanto, se emplea un bloque de *cinemática inversa* que traduce la inclinación deseada en los ángulos específicos que deben adoptar cada uno de los tres servomotores. Como resultado, se modifica la orientación de la plataforma y la bola se desplaza sobre ella, siendo su posición medida nuevamente por el sistema de percepción, cerrando así el lazo de control.

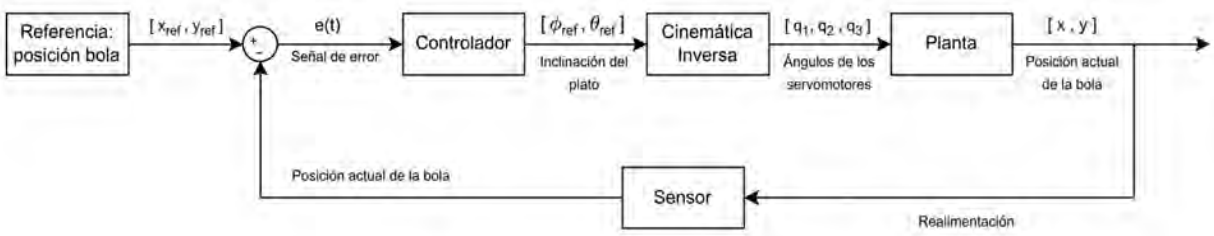


Figura 3.1: Diagrama de bloques general del lazo cerrado de control.

La implementación del sistema se realiza íntegramente en un **entorno de simulación**, en el que tanto la planta como los sensores y actuadores son modelados mediante herramientas de MATLAB/Simulink y Simscape Multibody. Este enfoque permite analizar el comportamiento del sistema en condiciones controladas, evaluar distintas estrategias de control y estudiar la influencia de perturbaciones y no idealidades.

El sistema de percepción se basa en un modelo de visión artificial simulado que proporciona una estimación de la posición de la bola en el plano. Esta información constituye la señal de realimentación del sistema, permitiendo corregir de forma continua las desviaciones respecto a la referencia. La interacción entre el sistema mecánico, el sistema de percepción y el controlador da lugar a un sistema dinámico multivariable no lineal cuyo comportamiento depende de la correcta coordinación de todos sus elementos.

La planta del sistema se modela mediante dos enfoques complementarios: un modelo matemático analítico, que permite el diseño de los controladores, y un modelo físico basado en simulación multicuerpo (mediante la herramienta de modelado CAD *Inventor*) exportado a MATLAB/Simulink mediante la librería Simscape Multibody, que proporciona una representación más realista del comportamiento del sistema.

El desarrollo de este modelo CAD se concibe con un nivel de detalle suficiente como para permitir una posible **implementación física**, mediante técnicas de prototipado rápido como la **impresión 3D**.

3.4.2. Descripción de la plataforma

La plataforma desarrollada en el presente trabajo (véase figura 3.2) se basa en una arquitectura de tipo 3-RRS, perteneciente a la familia de manipuladores paralelos. Tal y como

se ha descrito en el capítulo anterior, este tipo de configuración está compuesta por tres cadenas cinemáticas independientes que conectan una base fija con una plataforma móvil, cada una de ellas formada por tres articulaciones:

1. **Articulación de Revolución Activa (R):** Correspondiente al eje del actuador anclado a la base fija.
2. **Articulación de Revolución Pasiva (R):** La unión intermedia que permite el giro entre la manivela (eslabón conductor) y la biela (eslabón conducido).
3. **Articulación Esférica (S):** Unión tipo rótula que conecta el extremo de la biela con la plataforma móvil, permitiendo la rotación en los tres ejes espaciales.

Figura 3.2: Modelo CAD de la plataforma.

Las tres cadenas están dispuestas simétricamente a **120**° respecto al centro de la base, lo que permite una distribución uniforme de las fuerzas y un control equilibrado de la orientación del plato.

En cuanto a los grados de libertad, el mecanismo seleccionado presenta **tres grados de libertad**, correspondientes a dos rotaciones y una traslación vertical de la plataforma. No obstante, para la aplicación considerada en este trabajo, el movimiento vertical se **mantiene constante** durante la operación del sistema, empleándose únicamente los grados de libertad asociados a la inclinación de la plataforma. Esta simplificación permite centrar el problema en el control de la posición de la bola sobre el plano, reduciendo la complejidad del sistema sin comprometer la validez del estudio.

El accionamiento del sistema se realiza mediante **tres servomotores**, cada uno de los cuales actúa sobre una de las cadenas cinemáticas. Estos actuadores permiten modificar la orientación de la plataforma móvil de forma coordinada, generando las inclinaciones necesarias para controlar el movimiento de la bola. Si bien los servomotores constituyen una solución adecuada por su precisión y facilidad de control, existen alternativas como motores de corriente continua o motores paso a paso que podrían emplearse en una eventual implementación física del sistema.

Por otro lado, el sistema de percepción se basa en una **cámara** situada en posición superior, orientada perpendicularmente al plano de la plataforma, con el objetivo de obtener una vista cenital del sistema. En el contexto de este trabajo, dicho sistema se implementa mediante un modelo de visión artificial simulado en el entorno MATLAB/Simulink, encargado de procesar la imagen y proporcionar las coordenadas cartesianas (x, y) de la bola en tiempo real.

En conjunto, la plataforma descrita constituye la base física sobre la que se implementa el sistema de control, integrando los elementos mecánicos, de actuación y de percepción necesarios para el desarrollo del problema de estabilización planteado.

3.4.3. Variables del sistema

Con el fin de describir el comportamiento del sistema y establecer la relación entre el controlador, la plataforma y la bola, se definen las principales variables empleadas en el modelado y control. Las variables de salida corresponden a la posición de la bola sobre la superficie de la plataforma, representada mediante las coordenadas x e y. Estas magnitudes

constituyen el objetivo principal del control, ya que se pretende estabilizar la bola en una posición de referencia o, en futuras extensiones, hacerla seguir trayectorias determinadas.

El controlador actúa sobre el sistema generando referencias de inclinación para la plataforma, representadas por los ángulos ϕ_{ref} y θ_{ref}. Estas referencias indican la orientación deseada del plato respecto a sus ejes principales y permiten modificar indirectamente la dinámica de la bola. A partir de dichas referencias, el bloque de cinemática inversa calcula las variables articulares (q_1, q_2, q_3), asociadas a los actuadores de las tres cadenas cinemáticas de la plataforma paralela.

En el marco del modelado en **espacio de estados**, ampliamente utilizado en ingeniería de control [24], se consideran tanto las variables asociadas a la bola como las correspondientes a la orientación de la plataforma. De este modo, el estado del sistema incluye la posición y velocidad de la bola en el plano, x, $\dot{x}$, y e $\dot{y}$, junto con las inclinaciones de la plataforma y sus derivadas temporales, ϕ, $\dot{\phi}$, θ y $\dot{\theta}$. Esta representación permite describir de forma conjunta la evolución de la bola y la dinámica de orientación del plato, sirviendo como base para el diseño de las estrategias de control desarrolladas en capítulos posteriores.

En conjunto, el sistema puede interpretarse como una cadena de relaciones en la que el controlador genera referencias de orientación $(\phi_{ref}, \theta_{ref})$, la cinemática inversa las transforma en variables articulares (q_1, q_2, q_3), y la orientación resultante de la plataforma condiciona el movimiento de la bola (x, y).

4. MODELADO MATEMÁTICO

4.1. Estrategias de modelado del sistema

Con el objetivo de abordar el estudio del sistema desde una perspectiva completa, en este trabajo se adoptan dos enfoques de modelado complementarios, cada uno de ellos orientado a un propósito específico dentro del desarrollo del proyecto.

Por un lado, en este capítulo se plantea un **modelo analítico simplificado**, formulado en el espacio de estados, cuyo objetivo principal es facilitar el análisis dinámico del sistema y el diseño de controladores. Este modelo se basa en una serie de hipótesis simplificadoras, como la linealización para pequeñas inclinaciones y el desacoplo entre ejes, lo que permite obtener una representación matemática compacta y tratable.

Por otro lado, en el próximo capítulo se desarrollará un **modelo multicuerpo** en el entorno Simscape Multibody, en el cual se representa de forma explícita la geometría del manipulador paralelo, así como las propiedades físicas de sus componentes. Este modelo permite capturar el comportamiento no lineal del sistema y validar las hipótesis adoptadas en el modelo analítico.

Una de las principales diferencias entre ambos enfoques radica en el nivel de detalle con el que se representa la actuación sobre la plataforma. En el modelo multicuerpo, las inclinaciones deseadas del plato deben transformarse en variables articulares mediante la **cinemática inversa** del manipulador, ya que el sistema se excita a través de los actuadores reales. En cambio, en el modelo analítico en espacio de estados, la acción de los actuadores y la estructura mecánica del manipulador se agrupan en forma de **pares equivalentes** aplicados directamente sobre la plataforma, sin considerar explícitamente las coordenadas articulares.

Esta diferencia implica que la cinemática inversa, si bien resulta fundamental para el modelado físico del sistema, no se incorpora de forma explícita en el modelo dinámico lineal, donde su efecto queda implícitamente recogido en las variables de entrada del sistema.

En consecuencia, ambos modelos deben entenderse como representaciones complementarias del mismo sistema: el modelo analítico proporciona una herramienta adecuada para el diseño de controladores, mientras que el modelo multicuerpo permite evaluar la validez de las simplificaciones adoptadas y analizar el comportamiento del sistema en condiciones más realistas.

4.1.1. Objetivos del modelado matemático

El propósito principal de este capítulo es obtener un conjunto de expresiones analíticas que permitan predecir el comportamiento del sistema ante diferentes acciones de control. Para ello, el estudio se centra, en primer lugar, en resolver la **cinemática inversa** de la plataforma con el fin de determinar la posición de los actuadores necesaria para alcanzar una orientación específica del plato. El algoritmo de cinemática inversa será utilizado en el modelo en Simscape Multibody del próximo capítulo.

Asimismo, se busca caracterizar la **dinámica de la planta** estableciendo la relación entre la posición de la bola y los ángulos de inclinación de la plataforma para, finalmente, obtener un **modelo linealizado en el espacio de estados** que sirva como fundamento técnico en el diseño de las estrategias de control.

Por último se verá cómo implementar dicho modelo linealizado mediante MATLAB/Simulink.

4.1.2. Hipótesis adoptadas

Con el fin de obtener un modelo analítico tratable, se adoptan las siguientes hipótesis simplificadoras, ampliamente empleadas en la literatura para el modelado de sistemas tipo *Ball and Plate* [17], [21]:

- **Cuerpos rígidos:** Tanto la bola como la plataforma se consideran cuerpos perfectamente rígidos, despreciando posibles deformaciones elásticas durante el contacto.
- **Rodadura pura sin deslizamiento:** Se asume que el coeficiente de fricción estática es suficiente para asegurar que la bola rueda sobre la superficie en todo momento sin deslizar.

- **Contacto puntual:** La interacción entre la bola y el plato se produce en un único punto, despreciando la resistencia a la rodadura por deformación.
- **Simetría y homogeneidad:** La bola se considera una esfera perfecta de masa distribuida uniformemente.
- **Restricción de altura:** Se asume que la plataforma mantiene su centro de rotación en una cota Z constante, centrando el estudio en los giros de *Pitch* (ϕ) y *Roll* (θ).
- **Actuación equivalente simplificada:** Se considera que el sistema de actuación puede representarse mediante un bloque equivalente capaz de generar los pares necesarios para que la plataforma siga las referencias de orientación con una dinámica rápida frente al movimiento de la bola. No se modelan de forma detallada la dinámica interna de los actuadores, la electrónica de potencia, las saturaciones, las holguras ni los efectos de fricción.

4.1.3. Alcance del modelo

El modelo matemático desarrollado en este capítulo se centra en el comportamiento del sistema en las cercanías del **punto de equilibrio** (plataforma horizontal y bola en reposo).

Aunque el sistema real es intrínsecamente no lineal debido a las funciones trigonométricas de la cinemática y la dinámica de rotación, el alcance de este trabajo se limita a la **linealización para ángulos pequeños**. Esta aproximación es válida para el rango de operación previsto ($\pm 10^\circ$, valor típico en aplicaciones de este tipo), permitiendo el uso de herramientas de control lineal. No obstante, el modelo analítico aquí obtenido será validado en capítulos posteriores frente al entorno de simulación multicuerpo en Simscape, el cual sí contempla las no linealidades completas del sistema.

4.2. Modelado cinemático de la plataforma

El modelado cinemático del mecanismo tiene como objetivo establecer la relación entre las variables articulares de los actuadores (q_1, q_2, q_3) y la orientación de la plataforma móvil (ϕ_{ref}, θ_{ref}) proporcionada por el controlador. En el caso de manipuladores paralelos,

este análisis resulta especialmente relevante debido al fuerte acoplamiento geométrico existente entre las distintas cadenas cinemáticas.

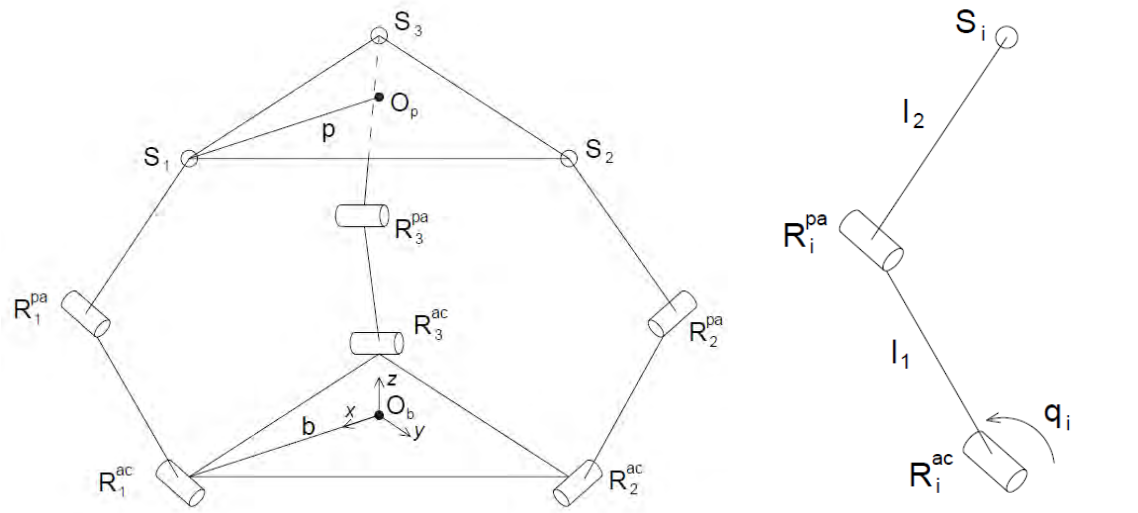
Como ya se estudió en la sección 2.5.1, hay dos formas de abordar el modelado cinemático: la directa y la inversa. El presente proyecto se centrará únicamente en el **método inverso**, si bien existen diversas metodologías en la literatura académica para desarrollarlo. Algunas aproximaciones recientes proponen el uso de herramientas matemáticas avanzadas como el Álgebra Geométrica Conformada (*Conformal Geometric Algebra*), la cual permite una representación unificada de las rotaciones y traslaciones mediante rotores y motores [19]. Otros enfoques se centran en el análisis diferencial, desarrollando no solo las posiciones, sino también las ecuaciones de restricción para velocidades y aceleraciones de los actuadores [18].

No obstante, este tipo de metodologías, si bien resultan potentes y generalizables, implican un mayor nivel de abstracción matemática y un coste computacional superior, lo que puede dificultar su implementación en aplicaciones prácticas de control en tiempo real o en entornos de simulación con requisitos de simplicidad y claridad estructural.

Por este motivo, en el presente trabajo se opta por un enfoque basado en un **análisis geométrico-vectorial**, sustentado en relaciones trigonométricas clásicas y en la formulación de ecuaciones de cierre geométrico para cada cadena cinemática. Este método permite obtener una solución explícita de la cinemática inversa con un equilibrio adecuado entre precisión, interpretabilidad y facilidad de implementación, resultando especialmente adecuado para su integración en el esquema de control del sistema.

4.2.1. Descripción geométrica del mecanismo

Tal como se ha descrito en el capítulo anterior, la plataforma empleada es una variante de la plataforma Stewart con tres grados de libertad y articulaciones **RRS**, esquematizada en la figura 4.1a, compuesta por tres cadenas cinemáticas idénticas que conectan la base fija con la plataforma móvil. Cada una de estas tres cadenas ($i = 1, 2, 3$) está formada por una articulación revoluta activa (R_i^{ac}), una segunda articulación revoluta pasiva (R_i^{pa}) y una articulación esférica (S_i) que permite la conexión con la plataforma.



(a) Estructura general. (b) Cadena cinemática individual.

Figura 4.1: Geometría del manipulador paralelo con 3-RRS.

Mientras que la base está fija y se encuentra a la altura más baja del robot, la plataforma es móvil y puede proporcionar un grado de libertad de traslación y dos de rotación. Sin embargo, para nuestro estudio, fijaremos la altura $z = z_0$. Es decir, la plataforma no sube ni baja globalmente, solo se inclina. Así pues, trabajamos únicamente con los dos grados de libertad de orientación, definidos mediante las referencias ϕ_{ref} y θ_{ref}. De acuerdo con el convenio adoptado en este trabajo, ϕ_{ref} se asocia a la inclinación que afecta a la dinámica de la bola en el eje X, mientras que θ_{ref} se asocia a la inclinación correspondiente al eje Y.

Los puntos de anclaje en la base y la plataforma se reparten simétricamente, con una configuración de triángulo equilátero, distribuidos a 120°. Los motores que proporcionan la inclinación se situarán en cada punto de unión de la base (R_i^{ac}), y la distancia entre el centro de la misma (O_b) con cada punto se llamará b. Por otro lado, nombramos p a la distancia entre el centro de la plataforma (O_p) hasta cada punto de unión (S_i).

Al derivar el modelo cinemático inverso, podemos pensar en centrarnos en una única pata al ser todas ellas idénticas (véase figura 4.1b). Llamamos l_1 a la longitud del brazo activo, l_2 a la longitud del brazo pasivo y q_i será el ángulo de cada motor i.

Estudiada la geometría del mecanismo, ya podemos abordar el problema de la cinemática inversa.

4.2.2. Cinemática inversa

Como se muestra en la figura 4.2, partimos de la **inclinación deseada de la plataforma** y queremos obtener los **tres ángulos de los motores**.

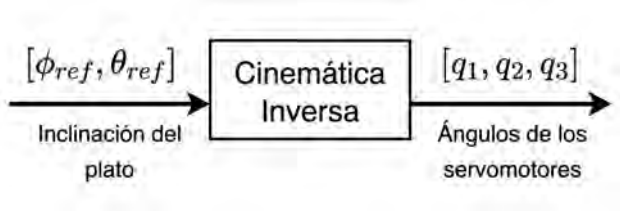


Figura 4.2: Bloque de la cinemática inversa en el esquema de control.

Para ángulos pequeños, la altura del punto de unión de la pata i en la plataforma cambia aproximadamente así:

$$h_i = z_0 + p(\cos\alpha_i\phi_{ref} + \sin\alpha_i\theta_{ref}) \tag{4.1}$$

Siendo:

$$\alpha_1 = 0, \quad \alpha_2 = \frac{2\pi}{3}, \quad \alpha_3 = \frac{4\pi}{3}$$

Esta aproximación es coherente con el rango de operación del sistema, ya que el control se diseñará en torno a la posición de equilibrio, donde las inclinaciones de la plataforma son reducidas. La ecuación (4.1) traduce la inclinación de la plataforma a desplazamiento vertical local del punto superior (S_i) de cada pata.

Definimos la distancia horizontal efectiva entre el anclaje de la base y el de la plataforma como:

$$d = b - p \tag{4.2}$$

Esto también es una simplificación razonable para el trabajo cerca de la posición horizontal.

El extremo del brazo activo, en el plano de la pata, está en:

$$C_i = \begin{bmatrix} l_1 \cos q_i \\ l_1 \sin q_i \end{bmatrix} \tag{4.3}$$

Y el punto de la plataforma que queremos alcanzar es en:

$$P_i = \begin{bmatrix} d \\ h_i \end{bmatrix} \tag{4.4}$$

Como el segundo brazo tiene longitud fija l_2, la distancia entre C_i y P_i debe ser exactamente l_2:

$$\|P_i - C_i\| = l_2 \tag{4.5}$$

La ecuación (4.5) se denomina **ecuación de cierre de lazo** y representa la condición geométrica necesaria para que el extremo del brazo activo y el punto de la plataforma estén separados exactamente una distancia l_2, garantizando el cierre de la cadena cinemática. Sustituyendo (4.3) y (4.4) en (4.5) hallamos:

$$(d - l_1 \cos q_i)^2 + (h_i - l_1 \sin q_i)^2 = l_2^2 \tag{4.6}$$

Esta es la ecuación fundamental de la cinemática inversa simplificada, la cual surge directamente de la geometría de una única pata. Indica que el brazo activo que mueve el motor debe colocarse justo en una posición tal que el brazo pasivo, de longitud fija, llegue al punto de la plataforma que se pretende alcanzar.

Desarrollando la expresión (4.6):

$$d^2 - 2dl_1 \cos q_i + l_1^2 \cos^2 q_i + h_i^2 - 2h_i l_1 \sin q_i + l_1^2 \sin^2 q_i = l_2^2$$
$$d^2 + h_i^2 + l_1^2(\cos^2 q_i + \sin^2 q_i) - 2dl_1 \cos q_i - 2h_i l_1 \sin q_i = l_2^2$$

Considerando la identidad trigonométrica fundamental $\cos^2 q_i + \sin^2 q_i = 1$, la expresión se simplifica como:

$$d^2 + h_i^2 + l_1^2 - l_2^2 - 2dl_1 \cos q_i - 2h_i l_1 \sin q_i = 0 \tag{4.7}$$

Para facilitar la resolución, definimos los siguientes coeficientes constantes para cada pata i:

$$\begin{aligned} A_i &= -2dl_1 \\ B_i &= -2h_i l_1 \\ C_i &= d^2 + h_i^2 + l_1^2 - l_2^2 \end{aligned} \tag{4.8}$$

Agrupando términos y reordenando la expresión, se obtiene:

$$A_i \cos q_i + B_i \sin q_i + C_i = 0 \tag{4.9}$$

Esta ecuación es del tipo $A\cos q + B\sin q + C = 0$ y puede resolverse analíticamente. La forma más práctica es emplear la sustitución de la tangente del semiángulo, resultando en la siguiente expresión para el cálculo de los ángulos de los servomotores:

$$q_i = 2\arctan\left(\frac{-B_i \pm \sqrt{A_i^2 + B_i^2 - C_i^2}}{C_i - A_i}\right) \tag{4.10}$$

De hecho, la expresión obtenida es consistente con las formulaciones reportadas en la literatura, como las presentadas por Moosavian et al. [17] y Tetik et al. [25], donde la cinemática inversa del manipulador 3-RRS conduce a ecuaciones de naturaleza trigonométrica equivalente.

El signo ± indica que, teóricamente, existen dos configuraciones geométricas posibles por cada cadena cinemática (denominadas habitualmente como *elbow up* y *elbow down*). En este trabajo se seleccionará la configuración compatible con la geometría física del prototipo y los límites articulares de los servomotores.

La expresión obtenida permite calcular directamente los ángulos articulares a partir de la inclinación deseada de la plataforma, constituyendo un bloque necesario para la creación del modelo multicuerpo que se verá en el capítulo 5.

No obstante, conviene señalar que las inclinaciones de referencia de la plataforma no coinciden necesariamente con las inclinaciones reales del sistema en todo instante, ya

que la plataforma presenta una dinámica propia. Por ello, en las secciones siguientes se distinguirá entre las variables de referencia ϕ_{ref} y θ_{ref}, generadas por el controlador, y las variables reales ϕ y θ, asociadas al movimiento efectivo del plato.

4.3. Modelado dinámico de la planta

4.3.1. Planteamiento del problema dinámico

Una vez establecida la relación cinemática entre las variables articulares y la orientación de la plataforma, el siguiente paso consiste en modelar el comportamiento dinámico del sistema, con el fin de describir la evolución temporal de la posición de la bola en función de las inclinaciones impuestas al plato.

El modelado dinámico de sistemas robóticos puede abordarse fundamentalmente mediante dos formulaciones: Newton–Euler y Euler–Lagrange. La primera se basa en la aplicación directa de las ecuaciones de equilibrio de fuerzas y momentos, mientras que la segunda se fundamenta en el principio de d'Alembert y en el principio de los trabajos virtuales, permitiendo expresar la dinámica del sistema en términos de energías y coordenadas generalizadas.

En este trabajo se ha optado por la formulación de **Euler–Lagrange**, siguiendo la metodología descrita por Spong et al. en *Robot Modeling and Control* [2]. Este enfoque permite obtener las ecuaciones de movimiento de manera sistemática a partir de una formulación energética, evitando la necesidad de modelar explícitamente las fuerzas de reacción internas. Esta característica resulta especialmente ventajosa en sistemas con cadenas cinemáticas cerradas, como la estructura paralela 3-RRS considerada.

Para el modelado dinámico de la planta, indicada en el esquema de control de la figura 3.1, el sistema completo puede descomponerse en distintos subsistemas, entre los que destacan la dinámica de la bola, la dinámica de la plataforma, la dinámica de los actuadores y la dinámica de los eslabones del manipulador.

No obstante, desde el punto de vista del control de la posición de la bola, las contribuciones dominantes corresponden a la dinámica de la bola y de la plataforma, siendo el resto de

efectos de menor influencia o susceptibles de ser modelados como perturbaciones o dinámicas internas de orden superior.

En consecuencia, en este trabajo se adopta un modelo simplificado en el que se consideran explícitamente la dinámica de la bola y la orientación de la plataforma, mientras que la dinámica de los actuadores se representa mediante un bloque de actuación equivalente. Este bloque genera los pares necesarios para el seguimiento de las referencias de orientación sin modelar de forma detallada la dinámica interna de los actuadores. El esquema adoptado se muestra en la figura 4.3 y permite obtener una representación más manejable sin perder las características esenciales del sistema para el diseño de controladores. También se tendrán en cuenta las hipótesis simplificadoras comentadas en el apartado 4.1.2.

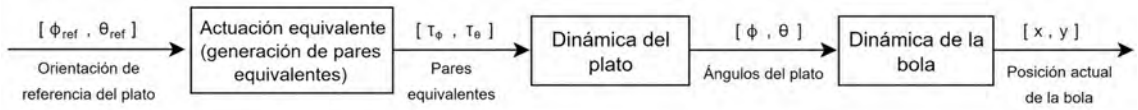


Figura 4.3: Modelado simplificado de la planta.

En el próximo capítulo se desarrollará un modelo completo del sistema implementado en Simscape Multibody, el cual permite validar el modelo simplificado y tener en cuenta de forma implícita la dinámica de los actuadores y de la estructura mecánica.

4.3.2. Dinámica de la bola

Consideremos el movimiento de la bola en una dirección s sobre un plano inclinado un ángulo α. La bola posee una masa m, un radio r y un momento de inercia J respecto a su centro. Definimos la coordenada generalizada como $q = s$.

Energía Cinética y Potencial

La energía cinética total (T) es la suma de los componentes traslacional y rotacional. Bajo la condición de rodadura sin deslizamiento ($\dot{s} = r\omega$), se expresa como:

$$T = \frac{1}{2}m\dot{s}^2 + \frac{1}{2}J\left(\frac{\dot{s}}{r}\right)^2 = \frac{1}{2}\left(m + \frac{J}{r^2}\right)\dot{s}^2 \tag{4.11}$$

La energía potencial gravitatoria (V), considerando la altura $h = s\sin\alpha$ y tomando como

referencia el centro de la plataforma, es:

$$V = mgs \sin \alpha \tag{4.12}$$

Lagrangiano y Ecuación de Lagrange

El Lagrangiano del sistema se define como $\mathcal{L} = T - V$:

$$\mathcal{L} = \frac{1}{2}\left(m + \frac{J}{r^2}\right)\dot{s}^2 - mgs \sin \alpha \tag{4.13}$$

Aplicando la ecuación de Euler-Lagrange $\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{s}}\right) - \frac{\partial \mathcal{L}}{\partial s} = 0$, se obtiene:

$$\left(m + \frac{J}{r^2}\right)\ddot{s} + mg \sin \alpha = 0$$

Considerando el sentido positivo de s en la dirección de descenso de la bola, la aceleración resultante es:

$$\ddot{s} = \frac{g \sin \alpha}{1 + \frac{J}{mr^2}}$$

Para una esfera maciza homogénea[1] ($J = \frac{2}{5}mr^2$), la ecuación se simplifica a:

$$\ddot{s} = \frac{5}{7} g \sin \alpha \tag{4.14}$$

Modelo Linealizado en el Espacio de Estados

En el sistema considerado, la inclinación de la plataforma se describe mediante los ángulos ϕ y θ. En particular, ϕ representa la rotación del plato alrededor del eje Y_p, mientras que θ representa la rotación alrededor del eje X_p. Con este convenio, la inclinación ϕ genera la componente de aceleración de la bola en la dirección X, mientras que la inclinación θ genera la componente de aceleración en la dirección Y. Aplicando la hipótesis de pequeñas inclinaciones ($\sin \alpha \approx \alpha$) a la ecuación 4.14, el sistema puede modelarse como

[1] Muchos sistemas *Ball and Plate* utilizan una bola de ping-pong, la cual se puede aproximar a una cáscara esférica delgada ($J = \frac{2}{3}mr^2$).

dos subsistemas desacoplados:

$$\ddot{x} \approx \frac{5}{7} g\, \phi, \quad \ddot{y} \approx \frac{5}{7} g\, \theta \tag{4.15}$$

Definiendo el vector de estado como $\mathbf{x_b} = [x, \dot{x}, y, \dot{y}]^T$ y el vector de entrada como $\mathbf{u_b} = [\phi, \theta]^T$, el modelo en espacio de estados queda definido por:

$$\dot{\mathbf{x}}_b = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \mathbf{x_b} + \begin{bmatrix} 0 & 0 \\ \frac{5}{7}g & 0 \\ 0 & 0 \\ 0 & \frac{5}{7}g \end{bmatrix} \mathbf{u_b}$$

La salida del sistema, correspondiente a las posiciones medidas, es:

$$\mathbf{y_b} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \mathbf{x_b}$$

4.3.3. Dinámica del plato

Consideremos ahora la dinámica de la plataforma móvil, modelada como un cuerpo rígido capaz de rotar alrededor de dos ejes ortogonales, asociados a los ángulos de inclinación ϕ y θ. Se definen como coordenadas generalizadas:

$$q_1 = \phi, \qquad q_2 = \theta$$

En la figura 4.4 se representa de forma esquemática el sistema de referencia asociado a la plataforma móvil, con origen en O_p, así como los dos grados de libertad de orientación considerados en el modelo, denotados por ϕ y θ.

Energía Cinética y Potencial

Suponiendo que la plataforma se comporta como un sólido rígido y que su centro de masas permanece a altura constante, la energía cinética total (T) viene dada únicamente por su

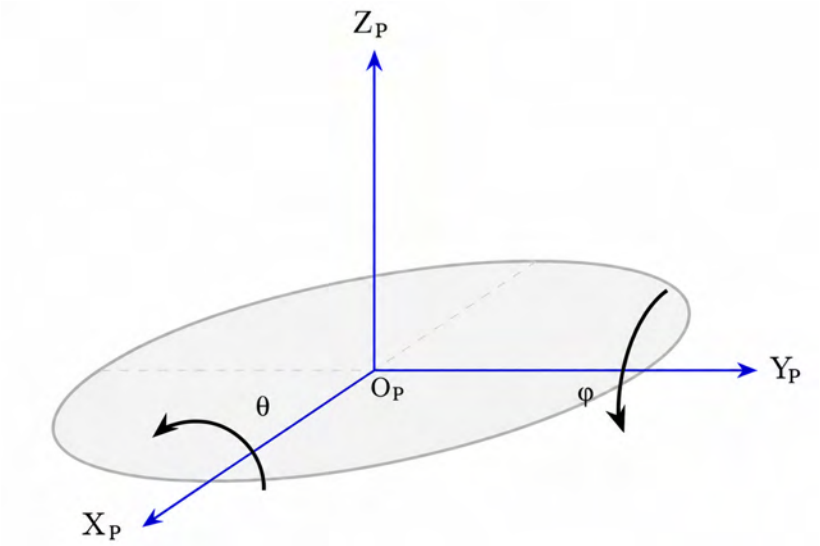


Figura 4.4: Representación esquemática del sistema de referencia de la plataforma y de las inclinaciones ϕ y θ.

energía cinética rotacional:

$$T = \frac{1}{2} J_\phi \dot{\phi}^2 + \frac{1}{2} J_\theta \dot{\theta}^2 \tag{4.16}$$

donde J_ϕ y J_θ son los momentos de inercia equivalentes de la plataforma respecto a los ejes de rotación considerados.

Dado que se asume que el centro de masas del plato no experimenta desplazamiento vertical apreciable durante su funcionamiento nominal, la energía potencial gravitatoria (V) puede considerarse constante, por lo que:

$$V = 0 \tag{4.17}$$

Lagrangiano y Ecuaciones de Lagrange

El Lagrangiano del sistema se define como:

$$\mathcal{L} = T - V = \frac{1}{2} J_\phi \dot{\phi}^2 + \frac{1}{2} J_\theta \dot{\theta}^2 \tag{4.18}$$

Para tener en cuenta las pérdidas mecánicas del sistema, se introducen pares disipativos equivalentes de tipo viscoso:

$$Q_\phi = \tau_\phi - b_\phi \dot{\phi}, \qquad Q_\theta = \tau_\theta - b_\theta \dot{\theta} \tag{4.19}$$

donde τ_ϕ y τ_θ representan los pares equivalentes aplicados sobre la plataforma, mientras que b_ϕ y b_θ son los coeficientes de amortiguamiento viscoso en cada eje.

Aplicando la ecuación de Euler-Lagrange con fuerzas generalizadas,

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{q}_i}\right) - \frac{\partial \mathcal{L}}{\partial q_i} = Q_i \tag{4.20}$$

se obtienen las ecuaciones de movimiento de la plataforma:

$$\begin{aligned} J_\phi \ddot{\phi} + b_\phi \dot{\phi} &= \tau_\phi \\ J_\theta \ddot{\theta} + b_\theta \dot{\theta} &= \tau_\theta \end{aligned} \tag{4.21}$$

Modelo Lineal en el Espacio de Estados

Definiendo el vector de estado como $x_p = [\phi, \dot{\phi}, \theta, \dot{\theta}]^T$ y el vector de entrada como $u_p = [\tau_\phi, \tau_\theta]^T$, el modelo en espacio de estados queda:

$$\dot{\mathbf{x}}_p = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -\frac{b_\phi}{J_\phi} & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -\frac{b_\theta}{J_\theta} \end{bmatrix} \mathbf{x}_p + \begin{bmatrix} 0 & 0 \\ \frac{1}{J_\phi} & 0 \\ 0 & 0 \\ 0 & \frac{1}{J_\theta} \end{bmatrix} \mathbf{u}_p$$

La salida del sistema, correspondiente a las inclinaciones de la plataforma, es:

$$\mathbf{y}_p = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \mathbf{x}_p$$

4.3.4. Modelo conjunto plato-bola en el espacio de estados

Considerando de forma conjunta la dinámica rotacional de la plataforma y la dinámica de la bola, el sistema completo puede modelarse como una cascada en la que la inclinación del plato actúa como entrada de la dinámica de la bola; es decir, las variables ϕ y θ determinan directamente la aceleración de la bola sobre cada eje. Bajo las hipótesis de pequeñas inclinaciones y rodadura sin deslizamiento, es posible obtener un modelo lineal

en espacio de estados que describe el comportamiento del sistema en torno a la posición de equilibrio.

Se define el vector de estado como:

$$\mathbf{x} = \begin{bmatrix} x & \dot{x} & y & \dot{y} & \phi & \dot{\phi} & \theta & \dot{\theta} \end{bmatrix}^T$$

donde x e y representan la posición de la bola sobre la plataforma, $\dot{x}$ y $\dot{y}$ sus velocidades, y ϕ, θ junto con sus derivadas representan la orientación de la plataforma y sus velocidades angulares.

El vector de entrada está formado por los pares equivalentes aplicados sobre la plataforma:

$$\mathbf{u} = \begin{bmatrix} \tau_\phi & \tau_\theta \end{bmatrix}^T$$

En el vector de salida se incluyen todas las variables de estado, con el fin de facilitar el análisis y la integración con el modelo en Simulink:

$$\mathbf{y} = \begin{bmatrix} x & \dot{x} & y & \dot{y} & \phi & \dot{\phi} & \theta & \dot{\theta} \end{bmatrix}^T$$

A partir de las ecuaciones obtenidas previamente para la dinámica conjunta, el sistema puede escribirse, siguiendo la formulación clásica [26], en forma de estado como:

$$\begin{aligned} \dot{\mathbf{x}} &= A\mathbf{x} + B\mathbf{u} \\ \mathbf{y} &= C\mathbf{x} + D\mathbf{u} \end{aligned} \tag{4.22}$$

donde las matrices del sistema vienen dadas por:

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & K_b & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & K_b & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -\frac{b_\phi}{J_\phi} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\frac{b_\theta}{J_\theta} \end{bmatrix} \tag{4.23}$$

Con $K_b = \frac{5}{7}g$ para una esfera maciza y homogénea.

$$B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ \frac{1}{J_\phi} & 0 \\ 0 & 0 \\ 0 & \frac{1}{J_\theta} \end{bmatrix} \tag{4.24}$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.25}$$

$$D = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \tag{4.26}$$

Este modelo describe la interacción entre la dinámica de la plataforma y la de la bola, donde los pares aplicados generan una inclinación del plato, y dicha inclinación provoca el movimiento de la bola sobre la superficie.

No obstante, es importante destacar que el modelo obtenido constituye una **aproximación desacoplada** del sistema. En particular, se asume que el movimiento de la bola en el eje X depende únicamente de la inclinación ϕ, mientras que el movimiento en el eje Y depende exclusivamente de la inclinación θ. En consecuencia, no se incluyen términos de acoplamiento entre ambos ejes, es decir, no aparecen dependencias cruzadas.

Aunque el sistema físico real es inherentemente **multivariable y no lineal**, la hipótesis de pequeñas inclinaciones, junto con la condición de rodadura sin deslizamiento, permite aproximar la dinámica de la bola mediante dos subsistemas independientes en los ejes X e Y. Esta simplificación resulta especialmente útil para el diseño inicial de controladores, al reducir significativamente la complejidad del modelo sin perder las características dinámicas dominantes del sistema en torno al punto de equilibrio.

4.3.5. Algoritmo de actuación equivalente

Tal como se ha indicado en las secciones anteriores, el modelo dinámico en espacio de estados se formula en términos de los pares equivalentes aplicados sobre la plataforma, representados por las variables τ_ϕ y τ_θ. Sin embargo, desde el punto de vista del control, las magnitudes de interés no son directamente estos pares, sino las inclinaciones de referencia de la plataforma, definidas por los ángulos ϕ_{ref} y θ_{ref}.

Con el fin de establecer una conexión entre ambas representaciones, se introduce un **modelo de actuación equivalente**, cuya función es generar los pares necesarios para que la plataforma siga las referencias de orientación impuestas por el controlador.

Este modelo no representa un actuador físico concreto, sino una abstracción que agrupa el efecto conjunto de los servomotores, la cinemática del manipulador y la transmisión mecánica. Bajo esta hipótesis, el sistema de actuación se representa mediante un bloque equivalente que aproxima el seguimiento de las referencias de orientación mediante una dinámica simplificada de segundo orden. Por tanto, no se asume una respuesta instantánea de la plataforma, sino una actuación suficientemente rápida respecto al movimiento de la bola, despreciándose la dinámica interna de los actuadores, la electrónica de potencia, las saturaciones, las holguras y los efectos de fricción.

De forma general, la relación entre las variables de referencia y los pares aplicados puede expresarse como:

$$\tau_\phi = f(\phi_{\text{ref}}, \phi, \dot{\phi}), \quad \tau_\theta = f(\theta_{\text{ref}}, \theta, \dot{\theta}) \tag{4.27}$$

donde la función $f(\cdot)$ representa una ley de actuación encargada de generar los pares equivalentes necesarios en función del error de orientación y de la dinámica del sistema.

Esta formulación permite separar el análisis dinámico simplificado de la complejidad cinemática del manipulador, evitando introducir explícitamente las variables articulares (q_1, q_2, q_3) en el modelo en espacio de estados.

Cabe destacar que esta aproximación resulta especialmente adecuada en el contexto del diseño de controladores, ya que permite trabajar con un modelo lineal y de orden reducido. No obstante, implica que la dinámica real de los actuadores y las restricciones geométricas del mecanismo no se consideran de forma explícita, siendo necesario recurrir a modelos más completos, como el desarrollado en Simscape Multibody, para validar el comportamiento del sistema en condiciones más realistas.

4.4. Implementación del modelo en MATLAB y Simulink

Una vez obtenido el modelo matemático del sistema, resulta necesario trasladar las expresiones analíticas desarrolladas a un entorno de simulación que permita su análisis numérico y el diseño de estrategias de control. En este trabajo, dicha implementación se realiza empleando MATLAB y Simulink, herramientas ampliamente utilizadas en el ámbito del control automático.

Como se muestra en la figura 4.5, la implementación computacional del sistema se aborda diferenciando claramente dos bloques fundamentales: la actuación equivalente de los motores y el modelo dinámico conjunto plato-bola en el espacio de estados.

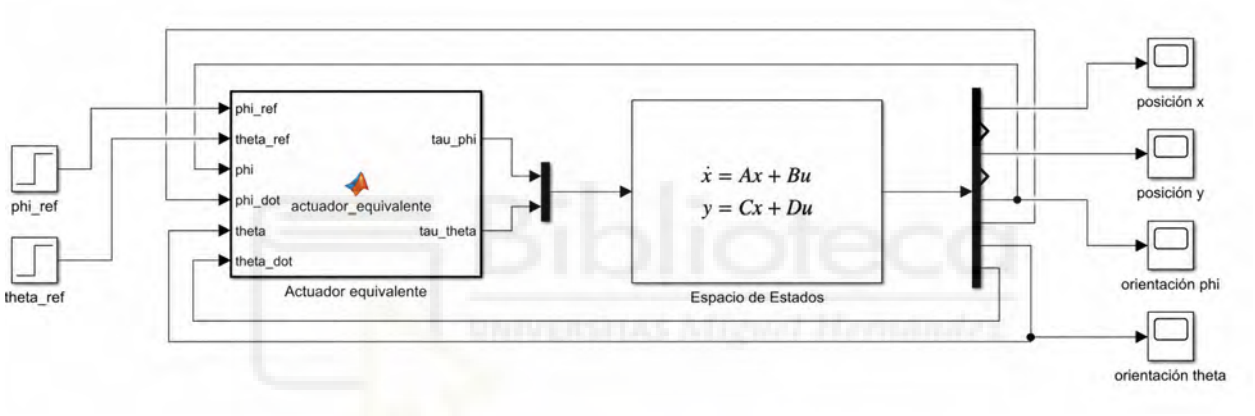


Figura 4.5: Esquema de prueba en Simulink del modelo de Espacio de Estados.

4.4.1. Implementación del bloque de actuación equivalente

El modelo de actuación equivalente descrito en la sección anterior se implementa en Simulink mediante un bloque de tipo *MATLAB Function*, en el cual se define una ley de control encargada de generar los pares equivalentes necesarios para que la plataforma siga las inclinaciones de referencia.

La implementación concreta de esta función se incluye en el Anexo A.3, donde se detalla el código empleado. Dicho bloque recibe como entradas las referencias de orientación (ϕ_{ref}, θ_{ref}), así como las variables reales del sistema (ϕ, $\dot{\phi}$, θ, $\dot{\theta}$), y proporciona como salida los pares equivalentes (τ_ϕ, τ_θ) que actúan sobre el modelo dinámico en espacio de estados.

Desde el punto de vista estructural, este bloque puede interpretarse como un **servo de**

orientación simplificado, cuya misión es imponer una dinámica deseada sobre la plataforma. Para ello, se define una ley de control de tipo proporcional-derivativa (PD), aplicada de forma independiente sobre cada uno de los ejes:

$$\tau_\phi = K_{p\phi}(\phi_{\text{ref}} - \phi) - K_{d\phi}\dot{\phi} \tag{4.28}$$

$$\tau_\theta = K_{p\theta}(\theta_{\text{ref}} - \theta) - K_{d\theta}\dot{\theta} \tag{4.29}$$

La inclusión de la realimentación de las variables ϕ y $\dot{\phi}$ (así como sus equivalentes en el eje θ) resulta imprescindible, ya que permite cerrar el lazo de control sobre la orientación de la plataforma. Sin esta realimentación, el sistema no podría corregir desviaciones respecto a la referencia ni compensar su propia dinámica, lo que impediría alcanzar el comportamiento deseado.

Las ganancias del controlador se seleccionan de forma que el sistema resultante presente una dinámica de segundo orden con características prefijadas. En particular, se define una frecuencia natural deseada ω_n y un coeficiente de amortiguamiento ζ, a partir de los cuales se calculan las ganancias como:

$$K_p = J\omega_n^2, \quad K_d = 2\zeta J\omega_n - b \tag{4.30}$$

donde J representa el momento de inercia del plato y b el coeficiente de amortiguamiento viscoso.

Esta elección permite que la dinámica cerrada del sistema en cada eje adopte la forma estándar de un sistema de segundo orden:

$$\ddot{\phi} + 2\zeta\omega_n\dot{\phi} + \omega_n^2\phi = \omega_n^2\phi_{\text{ref}} \tag{4.31}$$

y de forma análoga para el eje θ.

En este trabajo se ha seleccionado un valor de $\zeta = 1$, correspondiente a un **amortiguamiento crítico**, con el objetivo de obtener una respuesta rápida sin sobreoscilaciones. Esta

elección resulta especialmente adecuada en sistemas donde se desea evitar movimientos bruscos de la bola sobre la plataforma, ya que las oscilaciones en la orientación se traducen directamente en aceleraciones no deseadas.

Por otro lado, la frecuencia natural se fija en $\omega_n = 20\,\text{rad/s}$. Este valor se escoge de forma suficientemente elevada para que la dinámica del actuador equivalente sea significativamente más rápida que la dinámica de la bola, sin considerar una respuesta instantánea de la plataforma.

Con el fin de ilustrar este comportamiento, en la figura 4.6 se muestra la respuesta de la orientación ϕ ante una referencia escalón unitaria ϕ_{ref}. Puede observarse que la orientación del plato no alcanza la referencia de forma instantánea, sino mediante una respuesta transitoria rápida, coherente con la dinámica de segundo orden impuesta por el bloque de actuación equivalente.

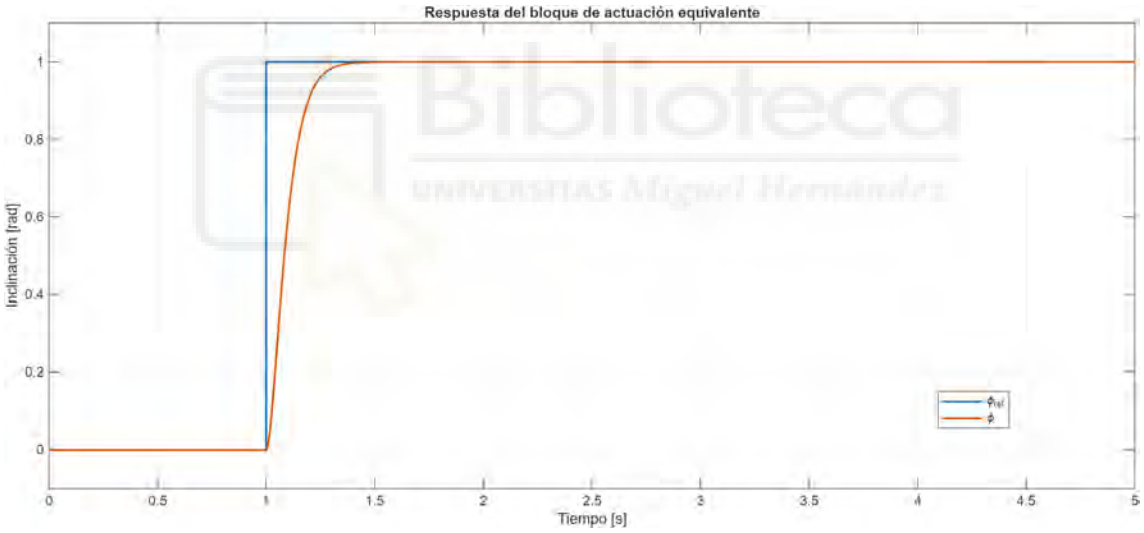


Figura 4.6: Comparación entre la referencia de inclinación ϕ_{ref} y la respuesta ϕ del bloque de actuación equivalente ante un escalón unitario.

En conjunto, esta formulación permite representar el comportamiento de los actuadores mediante una dinámica controlada y bien definida, sin necesidad de modelar explícitamente los servomotores ni la transmisión mecánica del sistema, manteniendo así la simplicidad del modelo en espacio de estados.

4.4.2. Implementación del modelo dinámico en espacio de estados

El modelo dinámico conjunto del sistema, obtenido en la sección 4.3.4, se implementa en MATLAB mediante la definición explícita de las matrices del sistema (A, B, C, D), tal como se recoge en el Anexo A.1.

A partir de estas matrices, se construye el modelo en espacio de estados utilizando la función `ss` de MATLAB, lo que permite representar el sistema en forma matricial y facilitar su análisis dinámico.

En el entorno Simulink, este modelo se integra empleando el bloque *State-Space*, en el cual se introducen directamente las matrices del sistema. Para ello, los valores numéricos de los parámetros dinámicos (véase tabla 5.3) se definen en un script de inicialización incluido en el Anexo A.2, el cual construye el sistema mediante la función `state_space_completa`, extrae las matrices A, B, C y D, y las exporta al *Workspace* de MATLAB para su utilización directa en el bloque *State-Space* de Simulink.

De este modo, se obtiene una representación compacta y eficiente de la dinámica del sistema completo, incluyendo tanto la evolución de la plataforma como el movimiento de la bola.

5. MODELADO CAD Y EXPORTACIÓN A SIMSCAPE MULTIBODY

5.1. Introducción

Una vez desarrollado el marco teórico y analítico en los capítulos precedentes, resulta imperativo verificar la fidelidad de las expresiones obtenidas antes de proceder al diseño de las estrategias de control. En este capítulo se aborda **la elaboración de un modelo físico virtual del sistema**, empleando herramientas de diseño CAD (Autodesk Inventor) y el entorno de simulación Simscape Multibody, una extensión de Simulink.

Para ello, se han desarrollado dos modelos CAD de la plataforma. Por un lado, un **modelo completo**, concebido con criterios de fabricación y ensamblaje, que representa de forma detallada el sistema real. Por otro lado, un **modelo simplificado**, en el que se han eliminado aquellos elementos geométricos no relevantes desde el punto de vista dinámico, con el objetivo de reducir la complejidad computacional y facilitar su integración en el entorno de simulación.

El uso de Simscape Multibody permite representar de forma más fiel la geometría, las masas y las inercias del sistema, así como introducir efectos no lineales que no han sido considerados en el modelo matemático simplificado. De este modo, es posible comparar la respuesta del modelo multicuerpo con la del modelo matemático desarrollado en el capítulo anterior, evaluando la validez de las hipótesis adoptadas.

5.2. Diseño CAD de la plataforma

5.2.1. Modelo CAD completo

Con el objetivo de disponer de una representación lo más fiel posible del sistema real, se ha desarrollado un modelo CAD completo de la plataforma empleando la herramienta de **modelado paramétrico** Autodesk Inventor. Este modelo ha sido concebido teniendo en cuenta criterios de fabricación y ensamblaje, lo que permite obtener una visión realista del dispositivo y evaluar su viabilidad constructiva e identificar posibles restricciones de

movimiento o colisiones espaciales.

La Figura 5.1 presenta todos los elementos principales del manipulador paralelo 3-RRS, tales como la base, la plataforma móvil, los brazos activos y pasivos, así como los puntos de unión mediante articulaciones. Asimismo, se han incorporado detalles geométricos adicionales, como soportes, fijaciones y elementos estructurales, que, si bien no son estrictamente necesarios para el modelado dinámico, resultan relevantes desde el punto de vista mecánico.

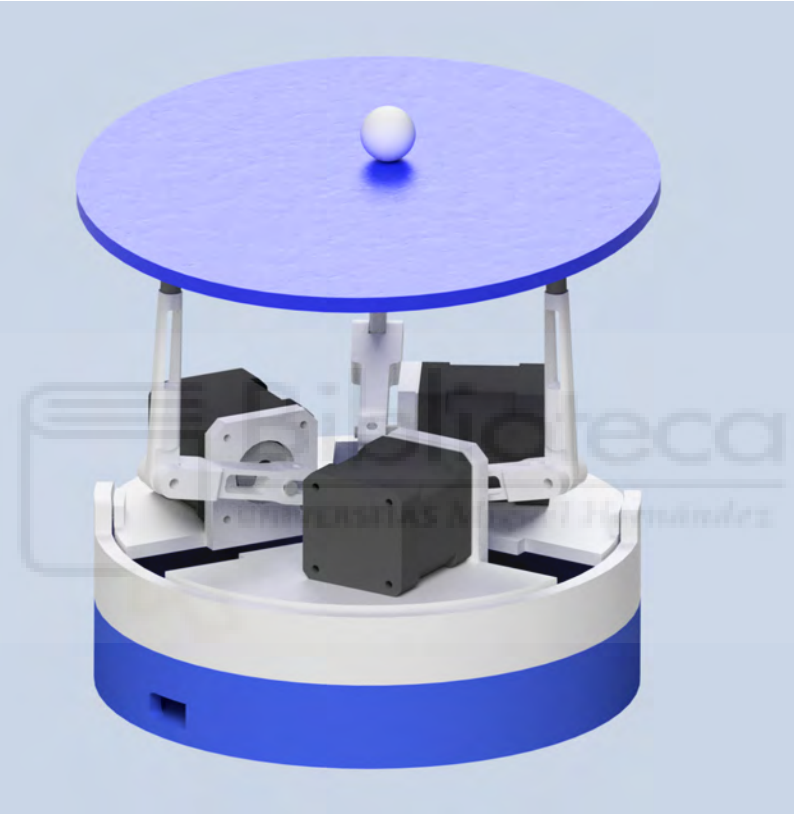


Figura 5.1: Vista general del modelo CAD completo de la plataforma.

El modelo desarrollado presenta una geometría detallada que permite visualizar la disposición espacial de los distintos componentes, así como las relaciones cinemáticas entre ellos. En particular, se ha prestado especial atención a la definición de las articulaciones (véase Figura 5.2) y a la correcta ubicación de los puntos de anclaje, de modo que la configuración del modelo sea coherente con el planteamiento teórico descrito en el capítulo anterior.

Además, el modelo CAD permite asignar materiales a los distintos componentes, lo que posibilita la estimación de propiedades físicas como masas y momentos de inercia.

Figura 5.2: Detalle de la cadena cinemática RRS.

Estas magnitudes resultan fundamentales para la posterior definición de los parámetros dinámicos del sistema, asegurando la **correlación directa** entre el modelo geométrico y el modelo matemático implementado en MATLAB y Simulink.

No obstante, debido al elevado nivel de detalle geométrico, este modelo presenta una alta complejidad computacional, lo que dificulta su uso directo en entornos de simulación dinámica como Simscape Multibody. Por este motivo, en las secciones siguientes se introduce un modelo simplificado, específicamente diseñado para su empleo en simulación.

5.2.2. Modelo CAD simplificado

Con el fin de facilitar la simulación dinámica del sistema, se ha desarrollado un modelo CAD simplificado a partir del modelo completo descrito en la sección anterior. Este modelo, mostrado en la figura 5.3, ha sido diseñado específicamente para su integración en Simscape Multibody, priorizando la eficiencia computacional frente al nivel de detalle geométrico.

El proceso de simplificación ha consistido en eliminar aquellos elementos geométricos que no influyen significativamente en la dinámica del sistema, tales como detalles estructurales, fijaciones, redondeos o componentes de carácter puramente estético. Asimismo, se han sustituido geometrías complejas por formas más simples, manteniendo la estructura general del manipulador y la disposición espacial de sus componentes.

Figura 5.3: Modelo CAD simplificado de la plataforma empleado para simulación.

En este modelo se conservan los elementos esenciales del manipulador, tales como la base, la plataforma móvil y las tres cadenas cinemáticas, garantizando la correcta representación de la geometría espacial del sistema. No obstante, al tratarse de una simplificación geométrica, las propiedades dinámicas derivadas del modelo CAD, como las masas o los momentos de inercia, pueden diferir de las del modelo completo.

Por este motivo, se establece una correspondencia entre los parámetros geométricos y dinámicos empleados en el modelo matemático —tales como l_1, l_2, b, p, z_0, J_ϕ y J_θ— y las características físicas del modelo CAD simplificado. De este modo, se garantiza que la configuración geométrica y el comportamiento dinámico del sistema en simulación sean coherentes con las hipótesis adoptadas en el capítulo anterior.

La simplificación del modelo permite reducir de forma significativa la carga computacional asociada a la simulación, mejorando la estabilidad numérica y disminuyendo los tiempos de cálculo. Esto resulta especialmente relevante en simulaciones donde se incorporan fenómenos adicionales, como el contacto entre la bola y la plataforma o la acción de control en lazo cerrado.

5.2.3. Comparativa y justificación del modelo simplificado

Tras la exposición de ambos diseños, resulta necesario establecer una comparativa técnica que justifique el empleo del modelo simplificado para las tareas de **validación cinemática**

y dinámica. Mientras que el modelo completo es una herramienta de ingeniería orientada al diseño físico y la fabricación, el modelo simplificado actúa como un nexo directo con las ecuaciones desarrolladas en el capítulo 4, facilitando su implementación en el entorno de simulación.

En la tabla 5.1 se resumen las diferencias principales entre ambas representaciones:

Tabla 5.1: Comparativa técnica entre el modelo CAD completo y el modelo simplificado.

Aspecto	**Modelo Completo**	**Modelo Simplificado**
Geometría	Realista y detallada (incluye elementos constructivos).	Esencialista (basada en sólidos primitivos).
Coste Computacional	Elevado (alto número de restricciones y contactos complejos).	Bajo (optimizado para una integración numérica rápida).
Precisión Dinámica	Alta (considera hasta el menor detalle estructural).	Suficiente para el rango de operación.
Objetivo Primario	Verificación de montaje, colisiones y estética.	Validación del modelo matemático y diseño de control.

Cabe destacar que el modelo matemático desarrollado en el capítulo anterior se basa en hipótesis simplificadoras, como la linealización para pequeñas inclinaciones y el desacoplo entre ejes. En este contexto, el uso de un modelo CAD altamente detallado no aporta una mejora significativa en la validación del modelo, mientras que sí introduce un incremento innecesario de la complejidad computacional.

Por el contrario, el modelo simplificado permite establecer una correspondencia directa con los parámetros geométricos y dinámicos utilizados en MATLAB y Simulink, facilitando la comparación entre el comportamiento del modelo analítico y el modelo de simulación.

Por tanto, se concluye que el modelo simplificado constituye una herramienta adecuada para la validación del modelo dinámico, permitiendo reproducir de forma fiel el comportamiento del sistema en las condiciones de operación consideradas, con un coste computacional reducido.

5.3. Parámetros geométricos y dinámicos del sistema

Una vez definidos los modelos virtuales, es necesario establecer los valores numéricos de los parámetros que rigen el comportamiento del sistema. La precisión en la determinación de estas magnitudes es crítica, ya que son empleadas tanto en las funciones de cinemática inversa en MATLAB como en la formulación del modelo en espacio de estados y en la implementación en Simscape Multibody.

5.3.1. Parámetros geométricos

Los parámetros geométricos describen la **configuración espacial** del manipulador 3-RRS y son empleados tanto en la formulación de la cinemática inversa como en la implementación del modelo dinámico. En la Tabla 5.2 se detallan las longitudes de los eslabones y las distancias características definidas en el sistema.

Tabla 5.2: Parámetros geométricos del sistema.

Parámetro	Descripción	Valor	Unidad
l_1	Longitud del brazo activo (manivela)	0,06	m
l_2	Longitud del brazo pasivo (biela)	0,12	m
b	Radio de la base fija (distancia del centro al eje del motor)	0,12	m
p	Radio de la plataforma móvil (distancia del centro a la rótula)	0,08	m
z_0	Altura nominal de reposo del plato	0,14	m

5.3.2. Parámetros dinámicos

Además de la geometría del sistema, el modelo dinámico requiere la definición de ciertas magnitudes físicas que describen el comportamiento inercial de la plataforma y de la bola. En particular, se consideran los parámetros establecidos en la Tabla 5.3.

Tanto las masa como los momentos de inercia del plato (J_ϕ, J_θ) han sido calculados a partir del modelo CAD simplificado en Autodesk Inventor, tras asignar como material el plástico ABS, con una densidad de 1,060 g/cm^3. La elección de dicho material se basa

Tabla 5.3: Parámetros dinámicos del sistema.

Parámetro	Descripción	Valor	Unidad
g	Gravedad	9,81	m/s^2
m_{bola}	Masa de la bola	0,01	kg
r_{bola}	Radio de la bola	0,005	m
J_b	Momento de inercia de la bola	1×10^{-7}	$kg{\cdot}m^2$
J_ϕ, J_θ	Momentos de inercia del plato (ejes X e Y)	$1{,}3413 \times 10^{-3}$	$kg{\cdot}m^2$
b_ϕ, b_θ	Coeficientes de amortiguamiento	0,01	N·m·s/rad

en la similitud con materiales comúnmente empleados en fabricación mediante impresión 3D. Es importante destacar que se asume simetría en el plato, por lo que $J_\phi \approx J_\theta$, lo que simplifica el posterior diseño de los controladores desacoplados.

Por otra parte, los coeficientes de amortiguamiento (b_ϕ, b_θ) no son proporcionados directamente por el modelo CAD, por lo que se han estimado para representar de forma aproximada las pérdidas por fricción en las articulaciones.

5.4. Exportación a Simscape Multibody

Una vez definido el modelo CAD simplificado y establecidos los parámetros físicos del sistema, el siguiente paso consiste en su integración en el entorno de simulación **Simscape Multibody**.

5.4.1. Introducción a Simscape Multibody

Simscape Multibody (anteriormente conocido como **SimMechanics**) es una librería especializada dentro del entorno de Simulink que permite modelar, simular y analizar sistemas mecánicos tridimensionales de forma eficiente. Esta herramienta adopta un enfoque de **modelado físico**, mediante el cual los sistemas se representan a partir de bloques que simbolizan cuerpos rígidos, articulaciones, restricciones y elementos de interacción.

Una de las principales ventajas de este entorno radica en que permite describir el comportamiento dinámico del sistema sin necesidad de formular explícitamente las ecuaciones

diferenciales de movimiento ni de implementar algoritmos numéricos de resolución [27]. En su lugar, el modelado se realiza de forma gráfica mediante bloques interconectados en Simulink, mientras que Simscape se encarga internamente de generar y resolver automáticamente las ecuaciones dinámicas del sistema a partir de la información geométrica, inercial y de las conexiones entre sus componentes.

Este enfoque permite obtener una representación dinámica fiel del manipulador 3-RRS basada en sus propiedades físicas de geometría, masa e inercia; incorporando de forma natural efectos no lineales y acoplamientos dinámicos que no han sido considerados en el modelo analítico en espacio de estados. De este modo, Simscape Multibody actúa como un nexo de unión entre el diseño CAD y la validación analítica.

5.4.2. Flujo de trabajo: De Autodesk Inventor a Simscape

La forma más directa de exportar el modelo desde Autodesk Inventor consiste en utilizar el plugin *Simscape Multibody Link*, que permite generar un archivo en formato XML que contiene la geometría, propiedades físicas y restricciones del ensamblaje. Este archivo puede ser posteriormente importado en MATLAB mediante el comando *smimport*, que traduce automáticamente dicha información en un modelo de bloques en Simulink que reproduce la estructura del sistema [20].

No obstante, debido a la incompatibilidad de este plugin con versiones recientes de Inventor [28], el proceso de integración se ha realizado mediante una **reconstrucción manual y paramétrica**.

Para ello, cada componente del sistema se ha exportado individualmente en formato *.stp* (estándar para el intercambio de modelos CAD tridimensionales) e incorporado mediante bloques **File Solid**, que permite importar la geometría de cada pieza junto con sus propiedades de masa e inercia. En la figura 5.4 se muestra la integración del plato.

En particular, se deben definir los sistemas de referencia (*frames*) que describen la posición y orientación relativa entre los distintos elementos. Esto se realiza mediante bloques **Rigid Transform** o directamente en la configuración de los bloques *File Solid*, asegurando la coherencia entre los ejes del modelo CAD y los utilizados en el modelo matemático.

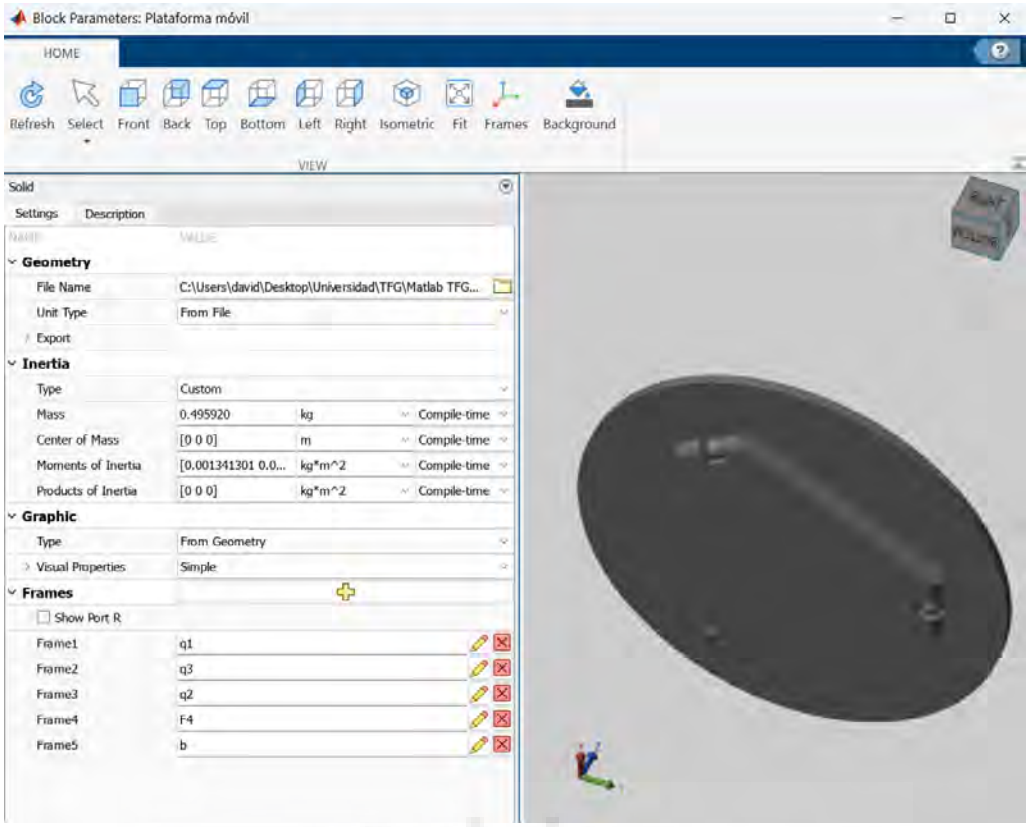


Figura 5.4: Integración del plato exportado desde Inventor.

Asimismo, es necesario incorporar las uniones (*joints*) que definen las restricciones cinemáticas del manipulador. Cada una de las tres ramas del mecanismo 3-RRS se modela mediante dos uniones de tipo revoluta (**Revolute Joint**) y una unión esférica (**Spherical Joint**), reproduciendo así la estructura física del sistema. Para asegurar la convergencia entre el modelo físico y el analítico, se han configurado los parámetros de **Internal Mechanics** en las articulaciones, asignando coeficientes de amortiguamiento viscoso consistentes con los del modelo de estados. Asimismo, se han definido las condiciones iniciales de posición y velocidad en los bloques de junta para garantizar que la simulación parta de un estado de equilibrio estático, evitando discontinuidades numéricas en el instante inicial.

En este modelo multicuerpo, la actuación del sistema se realiza a nivel de variables articulares. Por este motivo, las inclinaciones de referencia de la plataforma deben transformarse previamente en ángulos de los actuadores mediante la cinemática inversa del manipulador, desarrollada en el capítulo anterior. Dichos ángulos se introducen como entradas del modelo a través de las articulaciones activas, implementadas mediante fuentes de movimiento angular.

La interacción entre la bola y la plataforma se modela mediante el bloque **Sphere to Plane Force** de la librería *Simscape Multibody Contact Forces Library* [23], que permite simular contactos tridimensionales entre cuerpos. Este modelo emplea un enfoque de **fuerza de penalización** (*penalty method*), donde se calculan fuerzas de reacción normales y de fricción basadas en la penetración geométrica, la rigidez (*stiffness*) y el amortiguamiento (*damping*) del contacto.

La bola se modela directamente en Simscape mediante el bloque **Spherical Solid**, y se le asignan sus grados de libertad mediante una unión de tipo **6-DOF Joint**, lo que permite describir su movimiento libre en el espacio.

Para la obtención de las variables de interés, se incorporan distintos sensores al modelo. En particular, se emplean bloques **Transform Sensor** para medir la altura y orientación de la plataforma, así como la posición de la bola. La orientación se obtiene mediante la descomposición de la rotación en ángulos de Euler (ϕ, θ, ψ). Estas magnitudes se convierten en señales de Simulink mediante bloques **PS-Simulink Converter**, permitiendo su uso en el análisis y comparación de resultados.

Finalmente, el modelo incluye bloques esenciales para la simulación, como:

- **World Frame**, que define el sistema de referencia global.
- **Mechanism Configuration**, que permite establecer condiciones como la gravedad.
- **Solver Configuration**, necesario para la resolución numérica del sistema.

El esquema completo del modelo multicuerpo implementado en Simscape se muestra en la figura 5.5.

5.4.3. Ventajas de la simulación en Simscape Multibody

El uso de Simscape Multibody presenta diversas ventajas frente al modelo analítico en espacio de estados. En primer lugar, permite modificar de forma directa parámetros físicos del sistema, tales como:

Figura 5.5: Modelo multicuerpo implementado mediante Simscape Multibody.

- Masas e inercias de los componentes.
- Propiedades de las uniones, incluyendo rigidez, amortiguamiento y límites.
- Parámetros de la bola, como masa, radio o inercia.
- Coeficientes de fricción en el contacto bola-plataforma.

Asimismo, facilita la incorporación de sensores que permiten monitorizar en tiempo real variables como posición, velocidad, aceleración o esfuerzos en las articulaciones, proporcionando una visión más completa del comportamiento del sistema.

Otra ventaja fundamental es la eliminación de las hipótesis simplificadoras. A diferencia del modelo en espacio de estados, Simscape captura **fenómenos no lineales** y **acoplamientos dinámicos reales**, permitiendo validar las simplificaciones introducidas en la sección 4.1.2.

Finalmente, la simulación puede visualizarse en 3D mediante el entorno *Multibody Explorer* (véase figura 5.6), lo que proporciona una representación gráfica del movimiento del sistema y facilita la detección de colisiones o comportamientos anómalos que no son evidentes en las gráficas de señales.

5.4.4. Desventajas de la simulación en Simscape Multibody

A pesar de las ventajas que ofrece el modelado multicuerpo en Simscape Multibody, es importante considerar una serie de limitaciones asociadas a este enfoque.

En primer lugar, el modelo presenta un **mayor coste computacional** en comparación con el modelo analítico en espacio de estados, debido a la naturaleza no lineal del sistema y a la necesidad de resolver numéricamente un conjunto complejo de ecuaciones dinámicas. Esto puede dificultar su uso en aplicaciones que requieran simulaciones rápidas o en tiempo real.

Asimismo, la **definición del modelo requiere un mayor esfuerzo de implementación**, ya que es necesario configurar manualmente elementos como sistemas de referencia, uniones

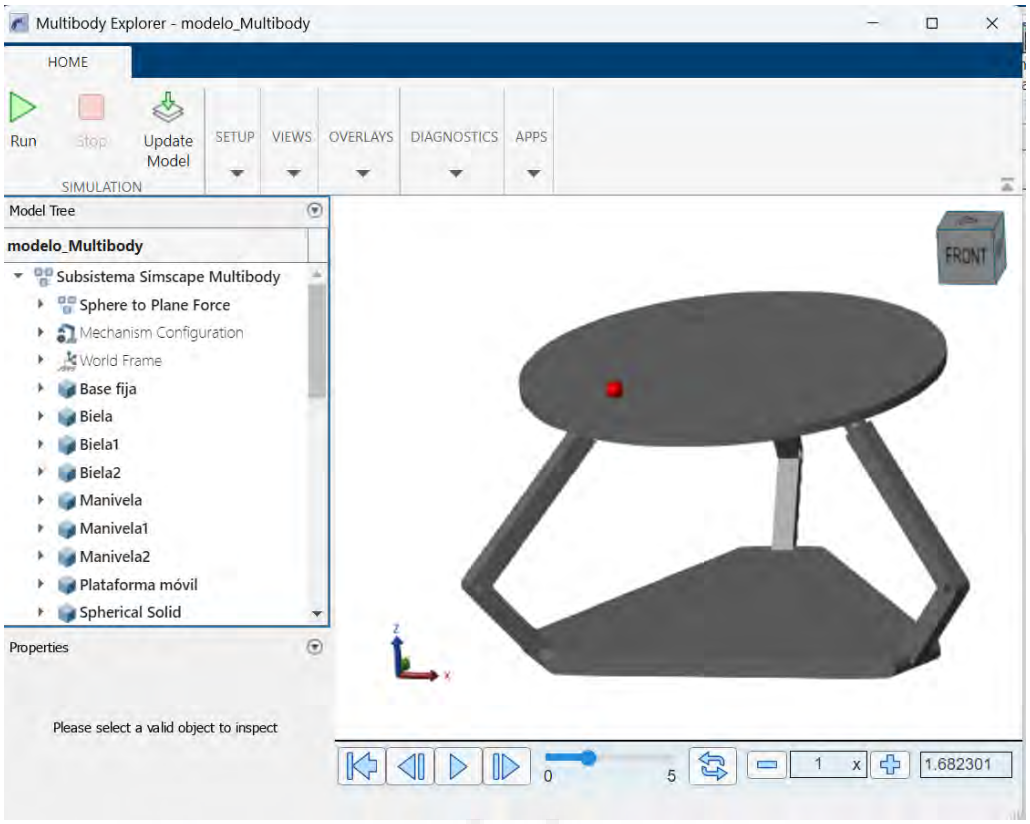


Figura 5.6: Representación 3D del movimiento del robot paralelo en *Multibody Explorer*.

o condiciones de contacto, especialmente cuando la exportación automática desde CAD no es viable.

Por otro lado, el comportamiento del sistema depende en gran medida de la correcta selección de parámetros, en particular en el modelado de contactos y fricción. Estos modelos suelen basarse en aproximaciones numéricas que pueden introducir desviaciones respecto al comportamiento físico real si no se ajustan adecuadamente.

Finalmente, aunque Simscape Multibody permite una representación física detallada del sistema, su estructura resulta **menos adecuada para el diseño directo de controladores**, en comparación con modelos analíticos linealizados, que facilitan el análisis y la síntesis de estrategias de control.

En consecuencia, el uso de modelos multicuerpo debe entenderse como una herramienta complementaria al modelado analítico, orientada principalmente a la validación y al análisis detallado del comportamiento del sistema. En este sentido, la validez del modelo en espacio de estados visto anteriormente se evaluará en el próximo capítulo mediante su comparación con este modelo multicuerpo desarrollado en Simscape.

5.5. Implementación del modelo multicuerpo en MATLAB/Simulink

Con el objetivo de facilitar la claridad del esquema de simulación y su posterior utilización en el análisis del sistema, el modelo multicuerpo desarrollado en Simscape se encapsula en un *subsistema*. Esta estrategia permite abstraer la complejidad del modelo físico, representándolo como un único bloque con entradas y salidas bien definidas.

De este modo, el modelo multicuerpo puede integrarse de forma sencilla en el entorno Simulink, permitiendo evaluar su comportamiento dinámico frente a distintas referencias de orientación, tal como se muestra en la figura 5.7. Cabe destacar que, a diferencia del modelo analítico en espacio de estados, en este caso no se recurre al bloque de actuación equivalente descrito en la sección 4.4.1, ya que la dinámica del sistema incluye explícitamente la estructura mecánica del manipulador paralelo.

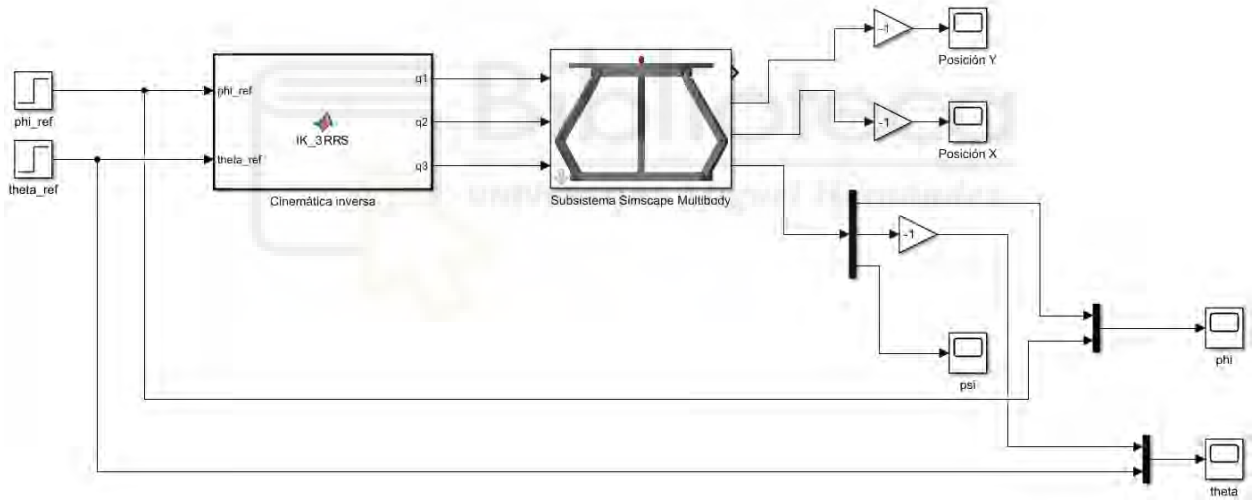


Figura 5.7: Esquema de prueba en Simulink del modelo Multicuerpo.

5.5.1. Estructuración del modelo como subsistema

El modelo generado en Simscape Multibody está compuesto por un gran número de bloques interconectados que representan cuerpos rígidos, uniones y fuerzas. Para mejorar la legibilidad del diagrama de Simulink, dicho modelo se agrupa en un subsistema, de forma que desde el exterior se comporte como un bloque funcional.

Este subsistema define claramente las siguientes interfaces:

- **Entradas:** ángulos articulares de los actuadores (q_1, q_2, q_3).
- **Salidas:** variables de interés, que incluyen la posición de la bola en el plano (x, y) y la orientación real de la plataforma (ϕ, θ).

Esta estructuración permite tratar el modelo multicuerpo como una **planta no lineal**, facilitando su integración con el resto del sistema.

5.5.2. Integración de la cinemática inversa

La cinemática inversa, desarrollada en la sección 4.2, se implementa en Simulink mediante el bloque *MATLAB Function* que permite calcular los ángulos articulares de los actuadores a partir de las inclinaciones de referencia de la plataforma, ϕ_{ref} y θ_{ref}.

Este bloque implementa la función que recibe como entradas las inclinaciones deseadas y los parámetros geométricos del manipulador, devolviendo como salida el vector de ángulos de los servomotores. La implementación completa se incluye en el Anexo A.4.

Tal como se mostraba en la ecuación 4.10, la resolución analítica de la cinemática inversa conduce a dos posibles soluciones para cada articulación, asociadas al signo ± presente en la expresión final. Estas soluciones corresponden a dos configuraciones geométricas distintas de cada cadena cinemática, comúnmente denominadas *elbow-up* y *elbow-down* (véase figura 5.8). En la implementación en MATLAB, esta ambigüedad se gestiona mediante el parámetro `elbowSign`, el cual toma valores ±1 y permite seleccionar la rama de solución deseada de forma explícita.

Cabe destacar que la elección incorrecta de la rama puede dar lugar a configuraciones no alcanzables o inconsistentes con la orientación real de los eslabones.

5.5.3. Consideraciones de implementación

Para garantizar la coherencia entre el modelo multicuerpo y el modelo en espacio de estados, se han tenido en cuenta diversas consideraciones:

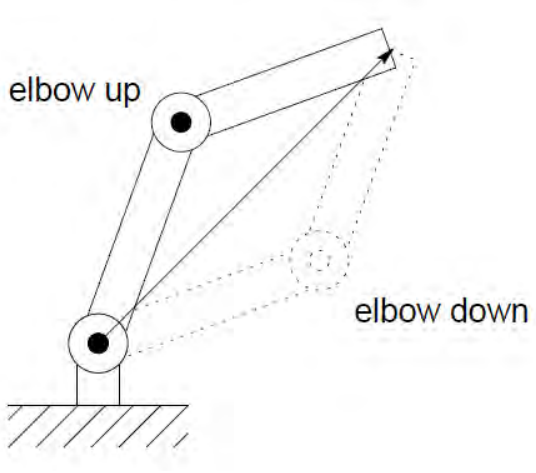


Figura 5.8: Distintas soluciones para las articulaciones. Imagen adaptada de [2].

- **Correspondencia de ejes:** Se han incorporado bloques de ganancia unitaria negativa o transformaciones de señal para asegurar que los ejes X e Y de Simscape coincidan con el convenio de signos del modelo en espacio de estados.
- **Unidades físicas:** Todas las magnitudes se expresan en unidades del Sistema Internacional (SI), garantizando la coherencia dimensional del modelo.
- **Conversión de señales:** Se emplean bloques de tipo *Simulink-PS Converter* y *PS-Simulink Converter* para adaptar las señales entre el dominio físico de Simscape y el entorno de Simulink.
- **Naturaleza no lineal del modelo:** El modelo multicuerpo presenta un comportamiento no lineal, lo que implica que su respuesta puede diferir del modelo linealizado, especialmente para grandes inclinaciones o condiciones alejadas del punto de equilibrio.

5.5.4. Interpretación del modelo resultante

El sistema completo implementado en Simulink puede interpretarse como una planta no lineal excitada a través de sus variables articulares, en la que la cinemática inversa actúa como elemento de transformación entre el espacio de control (orientación de la plataforma) y el espacio físico del manipulador (ángulos de los actuadores).

No obstante, debido a su elevada complejidad computacional y a la naturaleza no lineal del

modelo, el modelo multicuerpo no se empleará directamente para el diseño de controladores, tarea que se abordará utilizando el modelo lineal en espacio de estados desarrollado en el capítulo 4.

En su lugar, el modelo en Simscape se empleará en el capítulo siguiente como herramienta de **validación**, permitiendo contrastar el comportamiento del modelo analítico con una representación no lineal más fiel del sistema, así como evaluar el impacto de las simplificaciones introducidas en su formulación.

6. VALIDACIÓN Y ANÁLISIS COMPARATIVO DE MODELOS

El modelo en espacio de estados desarrollado en el capítulo 4 constituye la base para el diseño de los controladores del sistema. No obstante, dicho modelo se ha obtenido a partir de una serie de hipótesis simplificadoras, tales como la linealización en torno a pequeñas inclinaciones o el desacoplo entre ejes, que pueden introducir desviaciones respecto al comportamiento real del sistema.

Con el fin de evaluar la validez de este modelo, en el presente capítulo se lleva a cabo su comparación con el modelo multicuerpo desarrollado en Simscape Multibody en el capítulo 5. Este último proporciona una representación más completa del sistema, al considerar de forma explícita su geometría, las propiedades inerciales y las interacciones físicas, incluyendo efectos no lineales y de contacto.

El objetivo principal de este análisis es **determinar el grado de fidelidad del modelo analítico**, identificando sus limitaciones y cuantificando las discrepancias existentes entre ambos enfoques. De este modo, se pretende justificar el uso del modelo en espacio de estados para el diseño de controladores, así como delimitar su rango de validez.

6.1. Validación del algoritmo de cinemática inversa

Antes de proceder a la comparación dinámica entre modelos, resulta necesario verificar la validez del algoritmo de cinemática inversa desarrollado en la sección 4.2. Este algoritmo constituye el nexo entre el espacio de control (orientación de la plataforma) y el espacio físico del manipulador (ángulos articulares), por lo que su correcta implementación es fundamental para garantizar la coherencia del modelo multicuerpo.

Para ello, se propone un procedimiento de validación basado en dos enfoques complementarios: la verificación geométrica mediante el modelo CAD y la comprobación directa en el entorno de simulación Simscape.

En primer lugar, se procede al cálculo analítico de los ángulos articulares utilizando la función **inverse_kinematics_3rrs** (anexo A.4). Se definen los parámetros geométricos

establecidos en la sección 5.3.1 y se selecciona una postura de referencia con la plataforma paralela a la base ($\theta = \phi = 0$) a una altura nominal de trabajo de $z = 140mm$, empleando la configuración *elbow-up* ($elbowSign = 1$):

```
>> params.l1 = 0.06;
>> params.l2 = 0.12;
>> params.b = 0.12;
>> params.p = 0.08;
>> params.z0 = 0.14;
>> inverse_kinematics_3rrs(0,0,params,1)

ans =
2.2257
2.2257
2.2257

>> rad2deg(2.2257)

ans =
127.5232
```

Como se observa en el *Workspace*, los tres ángulos calculados para esta configuración resultan ser idénticos, con un valor de $127,52°$.

6.1.1. Verificación mediante entorno CAD

Se realiza una validación geométrica utilizando el modelo CAD simplificado (apartado 5.2.2). El procedimiento consiste en imponer en Autodesk Inventor un plano de trabajo paralelo a la base fija con un desfase de 140 mm. Se aplican restricciones de coincidencia entre el centro de la plataforma móvil y dicho plano para fijar su posición y orientación ($\phi = \theta = 0$).

Posteriormente, se miden los ángulos de las articulaciones activas mediante la herramienta de inspección del software. Como se aprecia en la Figura 6.1, el valor medido es de 127,53°. La elevada concordancia con el valor calculado (127,52°), con una desviación de apenas

el 0,01 %, ratifica la exactitud del algoritmo cinemático y la correcta correspondencia con el modelo físico.

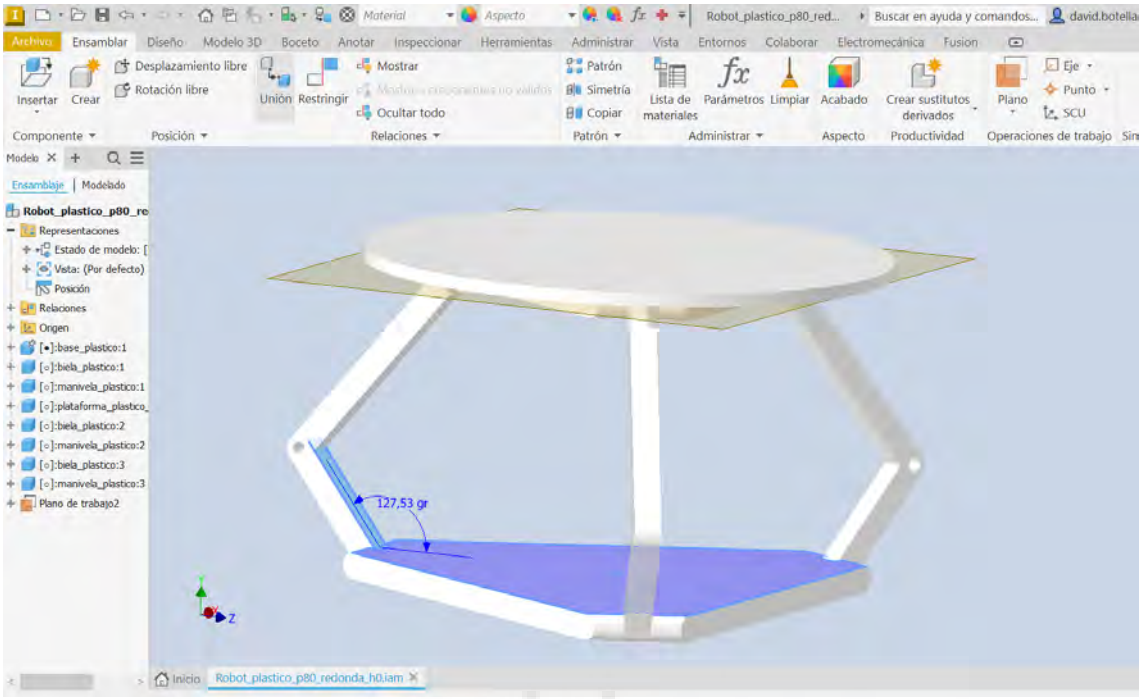


Figura 6.1: Medición de los ángulos de las articulaciones activas en Inventor.

6.1.2. Verificación mediante Simscape Multibody

De forma complementaria, se realiza una segunda validación del algoritmo en el entorno de Simscape Multibody, aprovechando el modelo multicuerpo desarrollado en el capítulo anterior.

Para ello, se emplean los bloques de tipo **Transform Sensor** establecidos en el esquema 5.5 para medir la posición y orientación de la plataforma, así como su altura respecto a la base. Al excitar el sistema con diferentes consignas de orientación (ϕ, θ), el sensor registra en tiempo real la altura y los ángulos de inclinación resultantes en el espacio 3D. La comparación entre las referencias de entrada y las medidas del sensor (visualizadas mediante bloques *Scope*) permite ratificar la correcta integración de la cinemática inversa en el modelo multicuerpo.

En conclusión, los resultados obtenidos mediante ambos procedimientos permiten validar el algoritmo de cinemática inversa en el rango de operación considerado, garantizando la

correcta transformación entre las variables de orientación de la plataforma y los ángulos articulares del manipulador.

6.2. Configuración del entorno de comparación

Para llevar a cabo una comparación válida entre el modelo en espacio de estados y el modelo multicuerpo desarrollado en Simscape Multibody, resulta imprescindible establecer un entorno de simulación coherente que garantice que ambos modelos son evaluados bajo condiciones equivalentes. Este enfoque, comúnmente referido como una comparación "**en igualdad de condiciones**", permite atribuir las diferencias observadas exclusivamente a la naturaleza de los modelos y a las hipótesis adoptadas en su formulación.

6.2.1. Consistencia paramétrica

El primer paso consiste en asegurar la coherencia entre los parámetros utilizados en ambos modelos. En particular, se han igualado las siguientes magnitudes:

- **Propiedades físicas del sistema:**
  - Aceleración de la gravedad g.
  - Momentos de inercia de la plataforma J_ϕ, J_θ, así como coeficientes de amortiguamiento viscoso b_ϕ, b_θ.
  - Masa, inercia y dimensiones de la bola m_{bola}, J_b, r_{bola}.
- **Parámetros geométricos:**
  - Longitudes de los eslabones del manipulador (l_1, l_2).
  - Posición de los puntos de anclaje respecto al centro de la base y la plataforma (b, p).
  - Altura nominal de trabajo z_0.

Estos parámetros, definidos en la sección 5.3, se han implementado en el entorno de Simscape mediante los bloques correspondientes (principalmente *File Solid* y configuraciones de masa e inercia), garantizando que ambos modelos representan el mismo sistema físico.

Por otro lado, existen ciertos parámetros que únicamente están presentes en el modelo multicuerpo, como los coeficientes de amortiguamiento de las articulaciones o los parámetros del modelo de contacto bola-plataforma (rigidez, fricción estática, fricción dinámica, etc.). Estos han sido ajustados de forma razonable con el fin de evitar comportamientos no físicos, si bien su influencia será analizada en la sección de discusión de resultados.

Añadir por último que, con el objetivo de garantizar una comparación coherente entre ambos modelos, en las simulaciones del modelo multicuerpo se considera una superficie de contacto **suficientemente extensa** para evitar la caída de la bola durante los ensayos. Esta decisión se adopta debido a que el modelo en espacio de estados no contempla los límites físicos del plato, por lo que la posición de la bola puede crecer de forma no acotada en lazo abierto. De este modo, se evita que la comparación quede condicionada por la pérdida de contacto o por la salida de la bola de la superficie, fenómenos que no están representados en el modelo analítico.

6.2.2. Esquema de simulación en Simulink

Una vez asegurada la consistencia paramétrica, se define el esquema de simulación en Simulink que permite llevar a cabo la comparación entre ambos modelos de forma directa.

Para ello, en la figura 6.2 se implementa una arquitectura en paralelo en la que el modelo en espacio de estados y el modelo multicuerpo en Simscape Multibody son excitados a partir de las mismas referencias de orientación de la plataforma (ϕ_{ref}, θ_{ref}), si bien adaptadas a sus respectivas variables de entrada.

En la rama correspondiente al modelo analítico, las referencias de orientación se introducen en el bloque de actuación equivalente desarrollado en la sección 4.4.1, que genera los pares equivalentes (τ_ϕ, τ_θ) necesarios para excitar el modelo en espacio de estados. Por otro lado, en la rama del modelo multicuerpo, dichas referencias se transforman mediante el bloque de cinemática inversa en los ángulos articulares (q_1, q_2, q_3), que actúan directamente sobre el subsistema de Simscape.

Las variables de salida de ambos modelos se almacenan mediante bloques de tipo **To Workspace**, configurados en formato *Structure with time*, lo que permite su posterior

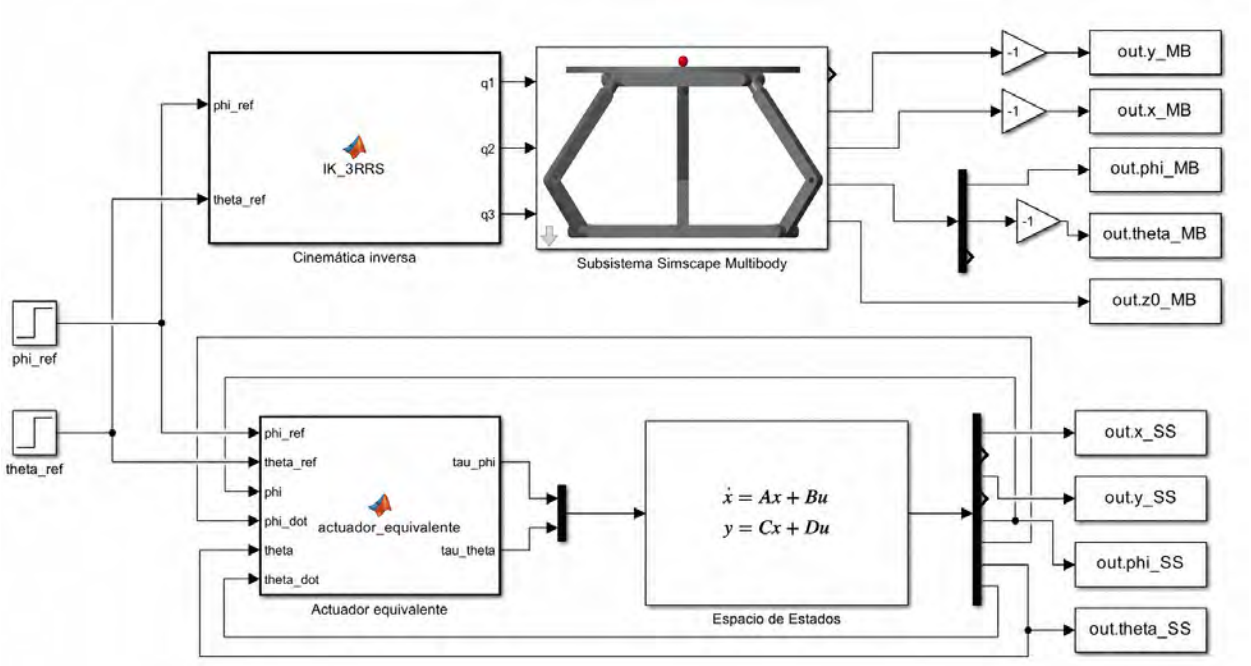


Figura 6.2: Esquema Simulink para la comparación de los modelos.

tratamiento en MATLAB. A partir de estos datos, se generan las gráficas comparativas y se realiza el análisis cuantitativo de las diferencias entre modelos. Aquellas variables que terminan con *MB* (*Multi-Body*) hacen referencia a las variables de salida del modelo multicuerpo, mientras que las finalizadas en *SS* (*State-Space*) se refieren al modelo de espacio de estados.

Las magnitudes seleccionadas para la comparación son:

- La posición de la bola en el plano (x, y).
- La orientación de la plataforma (ϕ, θ).

Adicionalmente, con el fin de evaluar el cumplimiento de una de las hipótesis fundamentales del modelo analítico, se analiza la desviación de la altura de la plataforma respecto a su valor nominal z_0, definiendo la variable:

$$\Delta z(t) = z_{MB}(t) - z_0$$

donde $z_{MB}(t)$ corresponde a la altura medida en el modelo multicuerpo. En el modelo en espacio de estados, dicha magnitud se considera constante por hipótesis ($z_0 = 0{,}14\,\text{m}$),

por lo que cualquier desviación observada en Simscape permite cuantificar el efecto de las no idealidades del sistema.

Finalmente, debido a la naturaleza no lineal del modelo multicuerpo y a la presencia de contactos, se emplea un *solver* de paso variable, como *ode15s*, que permite gestionar adecuadamente las posibles rigideces numéricas del sistema y garantizar la estabilidad de la simulación.

6.2.3. Definición de estímulos de referencia

Con el fin de evaluar el comportamiento de ambos modelos en diferentes condiciones de operación, se definen distintos tipos de entradas de referencia:

- **Entradas tipo escalón (Step):** Se aplican variaciones bruscas en las referencias de orientación (ϕ_{ref}, θ_{ref}) para analizar la respuesta transitoria del sistema, evaluando parámetros como el tiempo de establecimiento o la estabilidad.
- **Combinaciones de inclinaciones:** Se consideran excitaciones simultáneas en ambos ejes para estudiar el grado de acoplamiento entre las dinámicas en (x, y).
- **Pruebas de inclinación elevada:** Se introducen referencias de mayor amplitud con el objetivo de analizar el comportamiento fuera del rango de validez de la linealización, identificando las limitaciones del modelo en espacio de estados.

En todos los casos, se priorizan inicialmente pequeñas inclinaciones, donde se espera una mayor concordancia entre modelos, extendiendo posteriormente el análisis a condiciones más exigentes que permitan evidenciar las diferencias derivadas de las hipótesis simplificadoras.

6.3. Resultados de simulación

Una vez definido el entorno de comparación y establecidos los estímulos de entrada, se procede a analizar la respuesta dinámica de ambos modelos. Las simulaciones se han

realizado en el entorno de Simulink, almacenando las variables de interés en MATLAB para su posterior representación gráfica y análisis. En el Anexo A.5 se muestra el *script* utilizado para graficar las respuestas del sistema ante distintas entradas.

El objetivo de esta sección es comparar el comportamiento del modelo en espacio de estados y del modelo multicuerpo en Simscape Multibody, evaluando tanto la concordancia entre sus respuestas como las posibles discrepancias derivadas de las hipótesis simplificadoras.

6.3.1. Comparación de la orientación de la plataforma

En primer lugar, se analiza la evolución temporal de la orientación de la plataforma en ambos modelos, considerando como variables de interés los ángulos $\phi(t)$ y $\theta(t)$.

Para ello, se aplica una entrada tipo escalón en una de las componentes de la orientación (por ejemplo, ϕ_{ref}), manteniendo la otra constante, con el objetivo de evaluar la respuesta transitoria del sistema.

Los resultados de la figura 6.3 muestran que ambos modelos alcanzan valores próximos a la orientación de referencia en régimen permanente ($10^\circ \approx 0{,}175 rad$). No obstante, se aprecia una diferencia clara durante el transitorio inicial: la respuesta del modelo en espacio de estados es suave, mientras que la del modelo multicuerpo presenta una variación prácticamente instantánea.

Para pequeñas inclinaciones, como se observa en la figura 6.4, la concordancia en régimen permanente entre ambos modelos es elevada, lo que resulta coherente con la hipótesis de linealización adoptada en el modelo analítico.

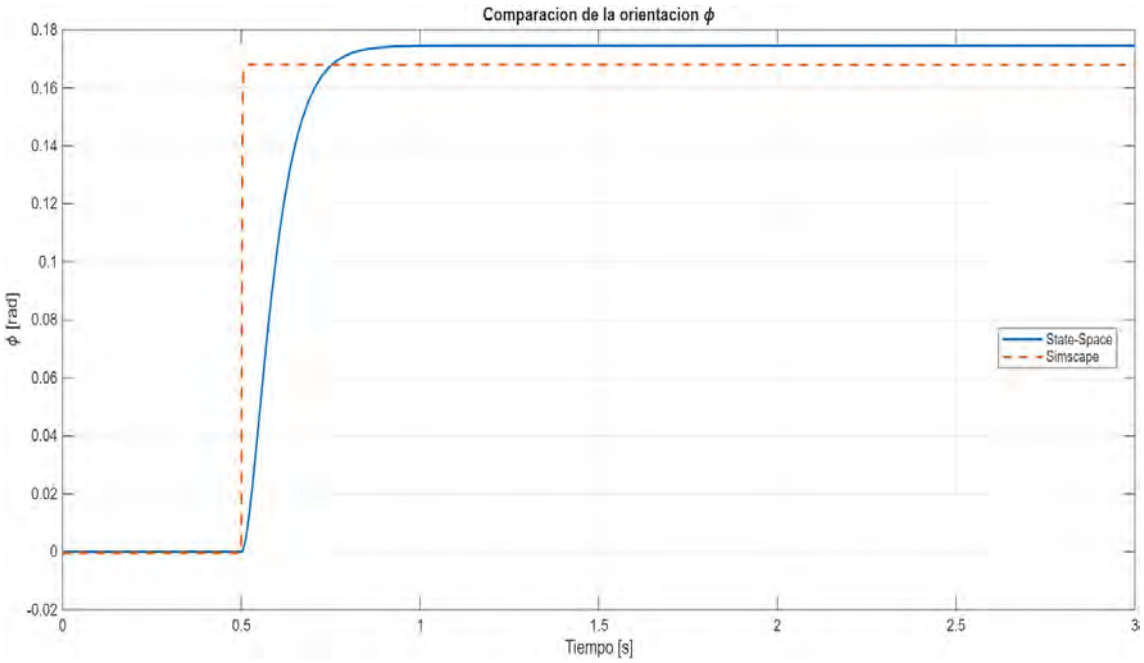


Figura 6.3: Comparación de la orientación $\phi(t)$ para $\phi_{\text{ref}} = 10°$ y $\theta_{\text{ref}} = 0°$.

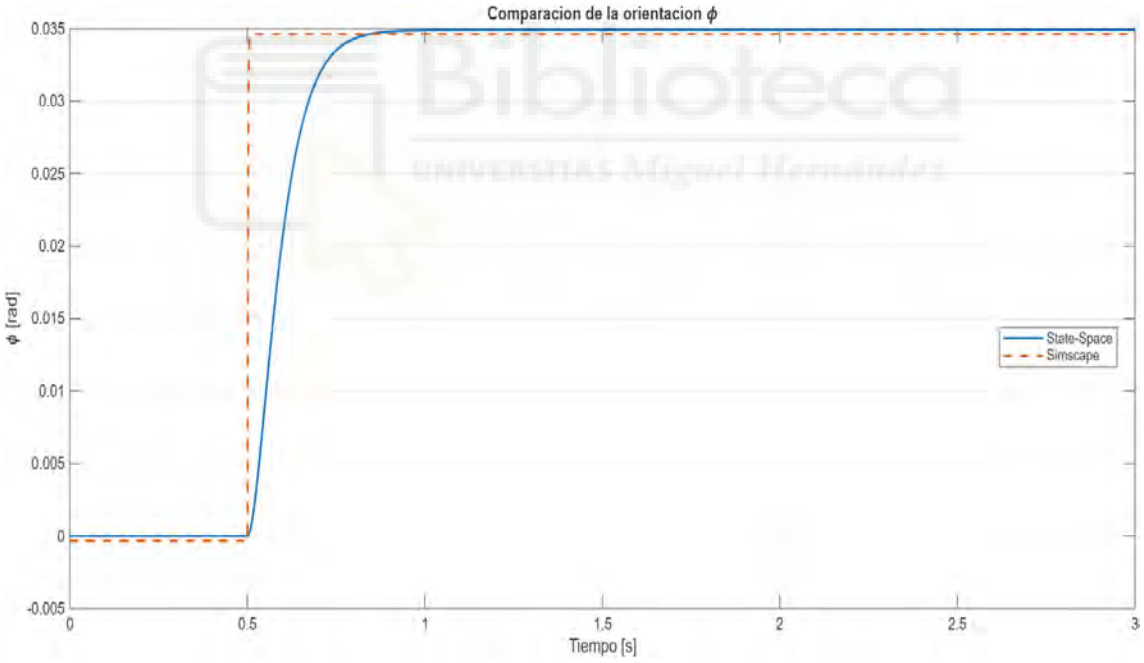
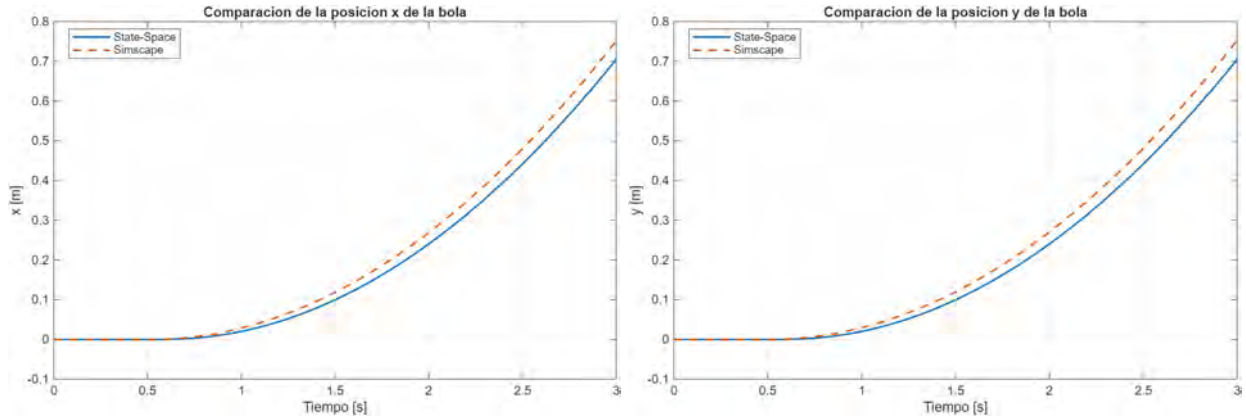


Figura 6.4: Comparación de la orientación $\phi(t)$ para $\phi_{\text{ref}} = 2°$ y $\theta_{\text{ref}} = 0°$.

6.3.2. Comparación de la posición de la bola

A continuación, se compara la evolución de la posición de la bola en el plano, representada por las variables $x(t)$ e $y(t)$.

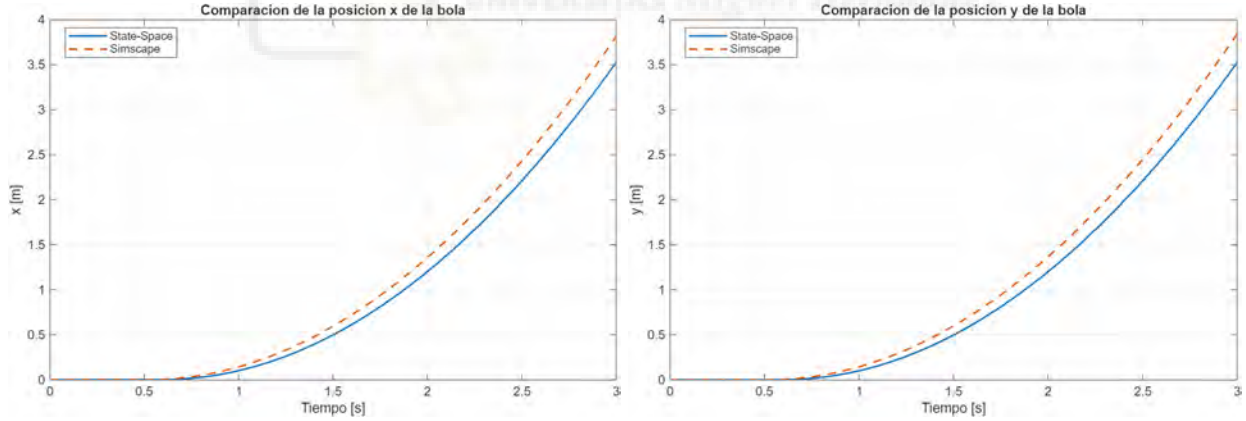
Los resultados evidencian una tendencia general similar en ambos modelos, especialmente en el caso de pequeñas inclinaciones (figura 6.5), donde las trayectorias $x(t)$ e $y(t)$ presentan una evolución muy próxima.



(a) Eje X. (b) Eje Y.

Figura 6.5: Comparación de la posición $x(t)$ e $y(t)$ para $\phi_{\text{ref}} = 2°$ y $\theta_{\text{ref}} = 2°$.

Sin embargo, a medida que aumenta la amplitud de las referencias (figura 6.6), las diferencias entre ambas respuestas se hacen más apreciables, aunque se mantiene la tendencia general del movimiento de la bola.



(a) Eje X. (b) Eje Y.

Figura 6.6: Comparación de la posición $x(t)$ e $y(t)$ para $\phi_{\text{ref}} = 10°$ y $\theta_{\text{ref}} = 10°$.

Estas diferencias ponen de manifiesto las limitaciones del modelo analítico, especialmente fuera del rango de validez de la aproximación lineal.

6.3.3. Análisis de la altura de la plataforma

Con el fin de evaluar el cumplimiento de una de las hipótesis fundamentales del modelo en espacio de estados, se analiza la evolución de la altura de la plataforma en el modelo multicuerpo.

En el modelo analítico, la altura se considera constante ($z = z_0$), mientras que en Simscape se mide directamente mediante sensores, obteniendo la variable $z_{MB}(t)$. Para cuantificar la diferencia entre modelos, se representa la desviación:

$$\Delta z(t) = z_{MB}(t) - z_0$$

Los resultados muestran que, si bien la altura de la plataforma permanece aproximadamente constante, se observan pequeñas variaciones en torno al valor nominal. En la figura 6.7 se muestra dicha desviación al proporcionar un escalón de 2° para ambas orientaciones de referencia.

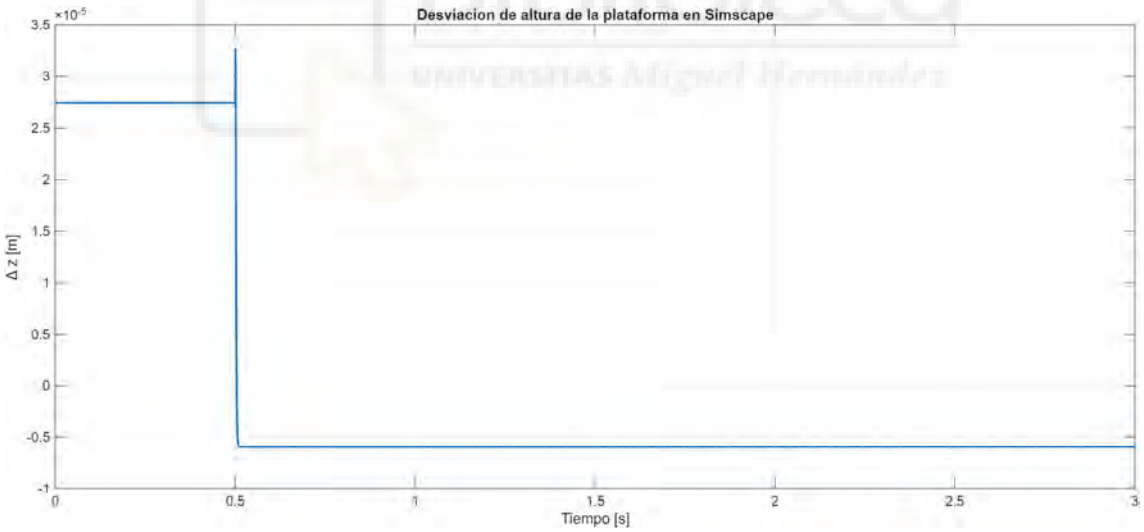


Figura 6.7: Desviación de la altura $\Delta z(t)$ en Simscape para $\phi_{\text{ref}} = 2°$ y $\theta_{\text{ref}} = 2°$.

En el instante de aplicación del escalón ($t = 0{,}5s$) se observa un pequeño pico transitorio en la desviación de altura. Este efecto se atribuye al reajuste instantáneo de la configuración del mecanismo multicuerpo ante el cambio brusco de consigna articular y a la resolución numérica del *solver* en Simscape. No obstante, su magnitud es del orden de 10^{-5} m, por lo que resulta despreciable frente a la altura nominal de la plataforma y no compromete la hipótesis de altura aproximadamente constante.

6.3.4. Análisis del error entre modelos

Con el objetivo de cuantificar las diferencias entre ambos enfoques, se calcula el error entre las variables de posición de la bola:

$$e_x(t) = x_{SS}(t) - x_{MB}(t), \quad e_y(t) = y_{SS}(t) - y_{MB}(t)$$

El error entre modelos permanece reducido para pequeñas inclinaciones, como se observa en la figura 6.8. Al aumentar la amplitud de las referencias, figura 6.9, el error crece de forma más acusada, especialmente conforme avanza el tiempo de simulación.

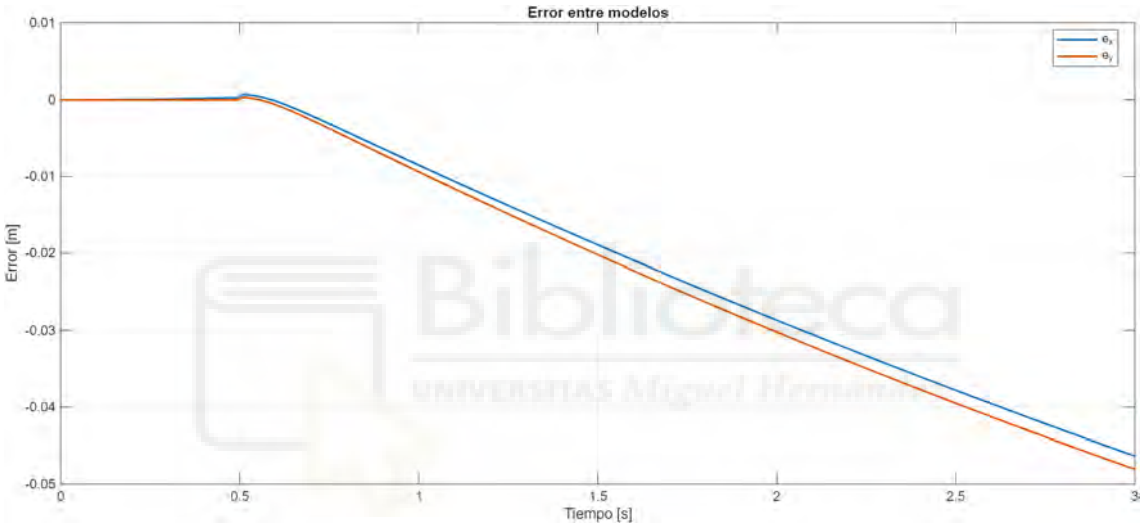


Figura 6.8: Comparación del error $e_x(t)$ e $e_y(t)$ para $\phi_{\text{ref}} = 2°$ y $\theta_{\text{ref}} = 2°$.

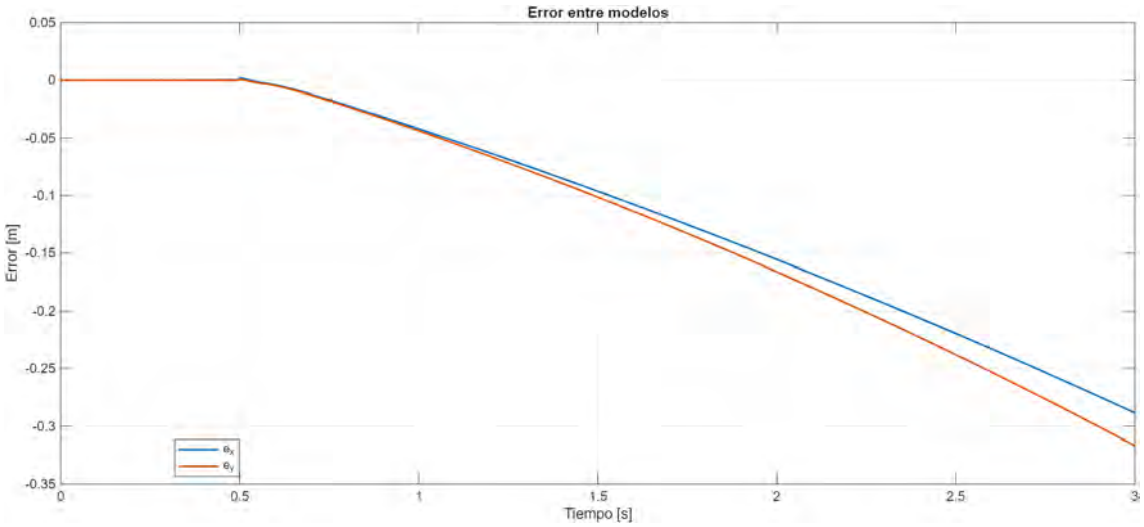


Figura 6.9: Comparación del error $e_x(t)$ e $e_y(t)$ para $\phi_{\text{ref}} = 10°$ y $\theta_{\text{ref}} = 10°$.

Cabe destacar que, aunque el sistema presenta una simetría nominal debido a la distribución

de los tres actuadores a 120°, los errores en los ejes X e Y no son exactamente iguales. Esta diferencia puede deberse a pequeñas asimetrías introducidas por la implementación multicuerpo, la definición de los sistemas de referencia, la extracción de las variables medidas y el modelo de contacto. En cualquier caso, las discrepancias observadas son reducidas y no comprometen la validez del modelo analítico en el rango de operación considerado.

6.4. Discusión de resultados

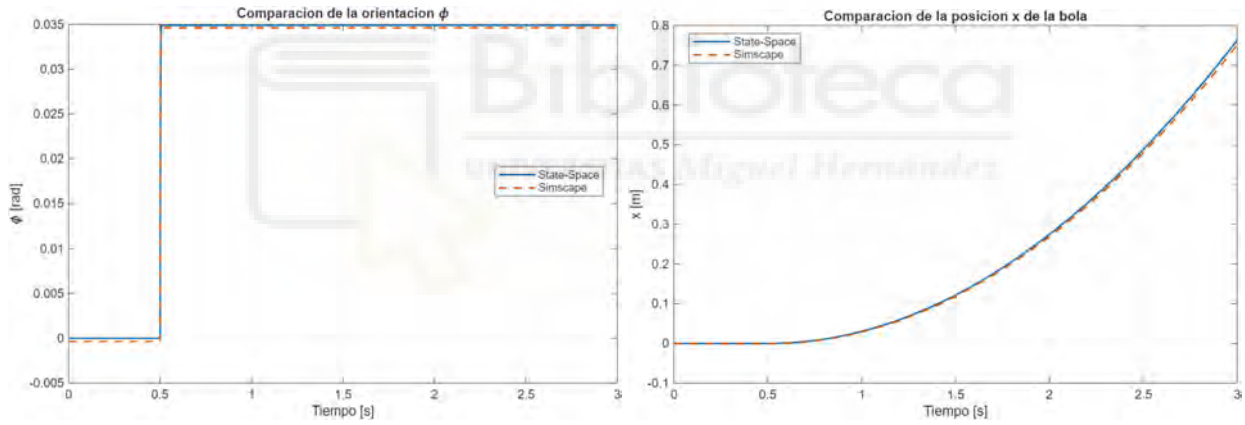
A partir de las simulaciones realizadas, se observa que el modelo en espacio de estados y el modelo multicuerpo desarrollado en Simscape Multibody presentan una **buena concordancia para inclinaciones pequeñas** de la plataforma. En particular, cuando las referencias de orientación se mantienen próximas al punto de equilibrio, ambos modelos predicen una evolución similar de la posición de la bola en los ejes X e Y. Este resultado confirma que el modelo lineal desarrollado en el capítulo 4 reproduce de forma adecuada la dinámica dominante del sistema dentro del rango de validez de las hipótesis adoptadas.

No obstante, existen diferencias entre ambos enfoques que deben tenerse en cuenta. Una de las más relevantes está asociada al modo de actuación empleado en cada modelo. En el modelo multicuerpo, las articulaciones activas se configuran mediante *Motion Actuation = Provided by Input*, por lo que los ángulos articulares calculados mediante la cinemática inversa se imponen directamente sobre las uniones del mecanismo. En consecuencia, la orientación de la plataforma medida en Simscape responde de forma **prácticamente instantánea** a la consigna articular aplicada.

Por el contrario, en el modelo en espacio de estados la orientación de la plataforma no cambia instantáneamente, sino que evoluciona de forma progresiva debido a la presencia del bloque de actuación equivalente descrito en la sección 4.4.1. Dicho bloque introduce una **dinámica de segundo orden** entre las referencias de inclinación y la orientación real del plato, por lo que la respuesta queda determinada por la frecuencia natural ω_n y el coeficiente de amortiguamiento ζ seleccionados. Esta diferencia en la actuación explica parte de la discrepancia inicial observada entre las curvas de orientación de ambos modelos, especialmente durante el transitorio.

Con el objetivo de analizar la influencia de esta dinámica de actuación en la comparación entre modelos, se ha realizado un ensayo complementario aumentando la frecuencia natural del bloque de actuación equivalente hasta $\omega_n = 1000\,\text{rad/s}$. Bajo esta condición, la respuesta de orientación del modelo en espacio de estados se aproxima a un seguimiento prácticamente instantáneo de la referencia, de forma más similar al comportamiento del modelo multicuerpo. Como se observa en la figura 6.10a, la diferencia entre la orientación $\phi(t)$ de ambos modelos se reduce de forma apreciable.

Esta mejora en la concordancia de la orientación se refleja también en la respuesta de la bola. En la figura 6.10b se muestra la posición $x(t)$ para el mismo ensayo, observándose una mayor similitud entre la respuesta del modelo en espacio de estados y la del modelo multicuerpo. Este resultado indica que una parte de las diferencias observadas en la comparación principal no se debe únicamente a las simplificaciones dinámicas del modelo analítico, sino también al distinto modo de actuación considerado en cada enfoque.



(a) Orientación $\phi(t)$. (b) Posición $x(t)$.

Figura 6.10: Comparación entre modelos para una frecuencia natural elevada del bloque de actuación equivalente ($\omega_n = 1000\,\text{rad/s}$).

No obstante, este valor elevado de ω_n se emplea únicamente con fines de análisis comparativo, para aproximar el comportamiento del bloque de actuación equivalente a una actuación casi instantánea. En las simulaciones de control desarrolladas en los capítulos posteriores se mantiene el valor $\omega_n = 20\,\text{rad/s}$, al considerarse una representación más realista de una actuación rápida pero no idealmente instantánea.

Por otra parte, cuando se incrementa la amplitud de las referencias de inclinación, la correlación entre ambos modelos empeora de forma progresiva. Este comportamiento es

coherente con la formulación del modelo en espacio de estados, basada en la hipótesis de pequeñas inclinaciones y en la linealización de las ecuaciones dinámicas alrededor del punto de equilibrio. Al aumentar la amplitud de las referencias, las aproximaciones lineales dejan de representar con la misma precisión el comportamiento real del sistema, por lo que aparecen discrepancias más significativas respecto al modelo multicuerpo.

Además, el modelo de Simscape Multibody incorpora fenómenos que no se encuentran representados de forma explícita en el modelo analítico, como los acoplamientos geométricos entre ejes, las restricciones cinemáticas del mecanismo 3-RRS, las pequeñas variaciones de altura de la plataforma y la interacción de contacto entre la bola y el plato. Estos efectos, aunque reducidos para pequeñas inclinaciones, adquieren mayor relevancia cuando el sistema se aleja del punto de operación nominal.

En relación con la altura de la plataforma, los resultados muestran que esta permanece aproximadamente constante en torno al valor nominal z_0, aunque se observan pequeñas desviaciones en el modelo multicuerpo. Dichas variaciones pueden ser consecuencia de la geometría tridimensional del mecanismo, de las restricciones impuestas por las articulaciones y de las tolerancias asociadas a la resolución numérica del *solver*. No obstante, su magnitud es reducida, por lo que la hipótesis de altura constante adoptada en el modelo analítico puede considerarse válida dentro del rango de operación estudiado.

Asimismo, debe destacarse que las simulaciones realizadas corresponden a un análisis en **lazo abierto**. Por ello, ante una inclinación constante del plato, la bola experimenta una aceleración sostenida y su posición crece de forma no acotada. En estas condiciones, pequeñas diferencias en la aceleración se integran con el tiempo y pueden dar lugar a errores crecientes en la posición. Este comportamiento pone de manifiesto la naturaleza inestable [2] del sistema y justifica la necesidad de diseñar estrategias de control en lazo cerrado.

En conjunto, los resultados obtenidos permiten concluir que el modelo en espacio de estados **reproduce adecuadamente la dinámica principal del sistema para pequeñas**

[2]De forma estricta, la bola situada en reposo sobre una plataforma perfectamente horizontal constituye una situación de equilibrio. No obstante, dicho equilibrio no es asintóticamente estable, ya que el sistema no presenta capacidad de retorno ante perturbaciones. Esta interpretación es coherente con la presencia de polos en el origen en el modelo lineal, como se observa en capítulos posteriores en la representación del lugar de las raíces de la figura 8.6.

inclinaciones y condiciones próximas al equilibrio, por lo que resulta apropiado para el diseño inicial de controladores. Por su parte, el modelo multicuerpo implementado en Simscape Multibody permite contrastar la validez de las hipótesis adoptadas e identificar las limitaciones del modelo analítico cuando se consideran efectos geométricos, no linealidades, contacto y modos de actuación más cercanos a una implementación física.

7. MODELADO DEL SISTEMA DE MEDIDA Y REALIMENTACIÓN

7.1. Introducción

En los capítulos anteriores se ha desarrollado y validado el modelo dinámico del sistema bola-plataforma, considerando tanto una formulación analítica en espacio de estados como un modelo multicuerpo implementado en *Simscape Multibody*. A partir de la comparación entre ambos enfoques, se ha determinado que el **modelo en espacio de estados** constituye la herramienta más adecuada para el diseño inicial de los controladores, debido a su menor complejidad computacional y a su validez en torno al punto de operación considerado. De este modo, el modelo analítico se emplea como base para el diseño de control, mientras que el modelo multicuerpo queda como banco de pruebas y referencia de validación física.

Una vez definido el modelo de la planta, es necesario establecer cómo se obtiene la información requerida para cerrar el **lazo de control**. En un sistema real, el controlador no dispone de todas las variables internas de la planta de forma directa, sino únicamente de aquellas magnitudes que pueden ser capturadas mediante sensores, lo que condiciona la estructura de realimentación empleada [24]. En el caso de una plataforma tipo *ball and plate*, las variables críticas para la estabilización son las coordenadas de posición de la bola sobre la superficie, expresadas como (x, y). A partir de esta medida, el controlador debe calcular las inclinaciones de referencia de la plataforma, que posteriormente se transforman en acciones equivalentes sobre el modelo dinámico mediante el bloque de actuación equivalente.

Al tratarse de un entorno íntegramente simulado, es posible acceder directamente a las salidas ideales del modelo de espacio de estados. En esta configuración, el vector de salida considerado coincide con el vector de estado de la planta:

$$\mathbf{y} = \mathbf{x} = \begin{bmatrix} x & \dot{x} & y & \dot{y} & \phi & \dot{\phi} & \theta & \dot{\theta} \end{bmatrix}^T \tag{7.1}$$

Esta disponibilidad completa de variables facilita el diseño inicial del sistema de control, especialmente en el caso de estrategias basadas en **realimentación de estados**, ya que

permite emplear directamente tanto las posiciones como las velocidades de la bola y de la plataforma.

Sin embargo, esta situación representa una idealización del sistema físico. En una implementación experimental, la medida de la posición de la bola se podría realizar mediante un sistema de visión artificial, por ejemplo una cámara cenital situada sobre la plataforma (entre otras posibilidades, como se indicó en la sección 2.7.1 del estado de arte). Este sensor proporcionaría únicamente las coordenadas (x, y), pero no mediría de forma directa las velocidades $(\dot{x}, \dot{y})$. Además, la señal visual estaría afectada por no idealidades propias del proceso de adquisición y procesamiento de imagen, tales como **ruido de medida**, **retardo temporal**, **frecuencia de muestreo limitada**, **cuantización** y **resolución finita** [29].

Por este motivo, en este capítulo se aborda el modelado del **sistema de medida** y su integración dentro de la estructura de realimentación. En primer lugar, se definen las variables medidas y su relación con el vector de estados de la planta. A continuación, se introduce un **sensor ideal de posición**, utilizado como caso base para el ajuste inicial de los controladores. Posteriormente, se plantea un **sensor visual simulado**, en el que se incorporan de forma progresiva las principales no idealidades asociadas a una cámara real. Finalmente, se analiza la necesidad de filtrar o estimar las señales medidas, prestando especial atención al **filtro de Kalman** como estimador de estados capaz de reconstruir las velocidades no medidas a partir de medidas de posición ruidosas.

Este capítulo establece, por tanto, el puente necesario entre el modelo dinámico desarrollado en capítulos anteriores y la implementación del control en lazo cerrado. La inclusión del sistema de medida permite distinguir entre un primer nivel de simulación ideal, orientado al diseño inicial de controladores, y un segundo nivel más realista, destinado a evaluar la influencia de las limitaciones sensoriales sobre el comportamiento global del sistema.

7.2. Variables medidas y estructura de realimentación

Una vez planteada **la necesidad de modelar el sistema de medida**, resulta necesario definir qué variables se consideran disponibles para la realimentación y cómo se relacionan

con el vector de estados de la planta (ec. 7.1).

Desde el punto de vista del sistema de medida, se distingue entre dos grupos de variables. Por un lado, las variables asociadas a la bola, principalmente x e y, que constituyen la salida principal del sensor visual. Por otro lado, las variables asociadas a la plataforma, ϕ y θ, junto con sus derivadas, que describen el estado de orientación del plato. Esta distinción permite separar conceptualmente la medida de la posición de la bola y la información relativa a la actuación de la plataforma.

En el presente trabajo, el sensor visual simulado se modela como un **sensor de posición bidimensional**. Por tanto, su salida se define como:

$$\mathbf{y}_m = \begin{bmatrix} x_m \\ y_m \end{bmatrix} \tag{7.2}$$

donde x_m e y_m representan las coordenadas medidas de la bola sobre la superficie. La relación entre el vector de estados de la planta y la salida del sensor puede expresarse mediante una **matriz de selección** C_m:

$$\mathbf{y}_m = C_m \mathbf{x}$$

siendo:

$$C_m = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{7.3}$$

Esta matriz selecciona únicamente las componentes de posición de la bola dentro del vector de estado completo. En consecuencia, las velocidades $\dot{x}$ e $\dot{y}$ no forman parte de la salida directa del sensor, sino que deberán obtenerse mediante estimación si la estrategia de control utilizada requiere dichas variables.

Para representar el comportamiento del sensor visual de forma **más realista**, se considera un modelo de medida que incorpora retardo y ruido aditivo. Esta expresión puede interpretarse como una extensión del modelo habitual de medida en espacio de estados, en el

que la salida medida se obtiene a partir de una matriz de observación y se ve afectada por ruido de medida [30]. En el caso de un sensor visual, además, puede existir un retardo asociado al tiempo de exposición, adquisición, procesamiento y transmisión de la imagen [29]. Así, la ecuación de salida del sensor queda definida como:

$$\mathbf{y}_m(t) = C_m\mathbf{x}(t - T_d) + \mathbf{v}(t) \tag{7.4}$$

donde T_d representa el retardo equivalente asociado al proceso de adquisición y procesamiento de la imagen, y $\mathbf{v}(t)$ representa el ruido de medida. Esta expresión constituye una formulación compacta del modelo de medida no ideal y servirá como base para introducir en secciones posteriores el ruido, el retardo, el muestreo y la cuantización.

La estructura general de realimentación queda organizada en tres bloques principales: planta, sistema de medida y controlador. La planta recibe como entradas los pares equivalentes τ_ϕ y τ_θ, generados por el bloque de actuación equivalente a partir de las referencias de orientación. A la salida de la planta, el sistema de medida selecciona las variables correspondientes a la posición de la bola y, en función del nivel de modelado considerado, puede introducir las no idealidades propias del sensor visual simulado. Finalmente, las señales medidas o estimadas se realimentan al controlador para cerrar el lazo.

Esta organización modular permite utilizar distintas configuraciones de medida sin modificar el modelo de planta. En primer lugar, puede emplearse un sensor ideal de posición, adecuado para verificar el comportamiento básico del lazo cerrado. Posteriormente, el mismo subsistema puede ampliarse mediante la incorporación de efectos no ideales y, si resulta necesario, mediante un estimador de estados que proporcione las variables no medidas directamente.

7.3. Sensor ideal de posición

Como primer nivel de modelado del sistema de medida, se considera un **sensor ideal de posición**. Esta aproximación permite cerrar el lazo de control sin introducir inicialmente errores asociados al proceso de medida, de forma que el comportamiento obtenido dependa

únicamente de la dinámica de la planta y del controlador diseñado.

Bajo esta hipótesis, la posición de la bola sobre la plataforma se considera disponible de forma exacta, continua e instantánea. Por tanto, las coordenadas medidas coinciden directamente con las coordenadas reales proporcionadas por el modelo dinámico:

$$x_m(t) = x(t)$$

$$y_m(t) = y(t)$$

o, de forma vectorial:

$$\mathbf{y}_m(t) = \begin{bmatrix} x_m(t) \\ y_m(t) \end{bmatrix} = \begin{bmatrix} x(t) \\ y(t) \end{bmatrix}$$

De acuerdo con la matriz de selección C_m definida en el apartado anterior, esta medida ideal puede escribirse como:

$$\mathbf{y}_m(t) = C_m \mathbf{x}(t) \tag{7.5}$$

En este caso, el sensor no introduce **ruido**, **retardo**, **cuantización** ni limitaciones de **resolución**. La señal de realimentación empleada por el controlador se corresponde, por tanto, con la posición real de la bola sobre la plataforma.

A partir de esta medida, el **error de posición** utilizado por el controlador se define como:

$$\mathbf{e}(t) = \mathbf{r}(t) - \mathbf{y}_m(t)$$

donde $\mathbf{r}(t)$ representa la referencia de posición de la bola:

$$\mathbf{r}(t) = \begin{bmatrix} x_{ref}(t) \\ y_{ref}(t) \end{bmatrix}$$

Para el caso de estabilización en el centro de la plataforma, la referencia se fija como:

$$\mathbf{r}(t) = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

mientras que para ensayos de seguimiento podría definirse una trayectoria variable en el tiempo. De este modo, el controlador calcula la acción de control a partir de la diferencia entre la posición deseada y la posición medida de la bola.

La estructura de realimentación con sensor ideal se muestra conceptualmente en la figura 7.1. En ella, la planta proporciona el vector de estado completo, mientras que el sensor ideal entrega únicamente la posición de la bola, sin modificar ni degradar la señal medida.

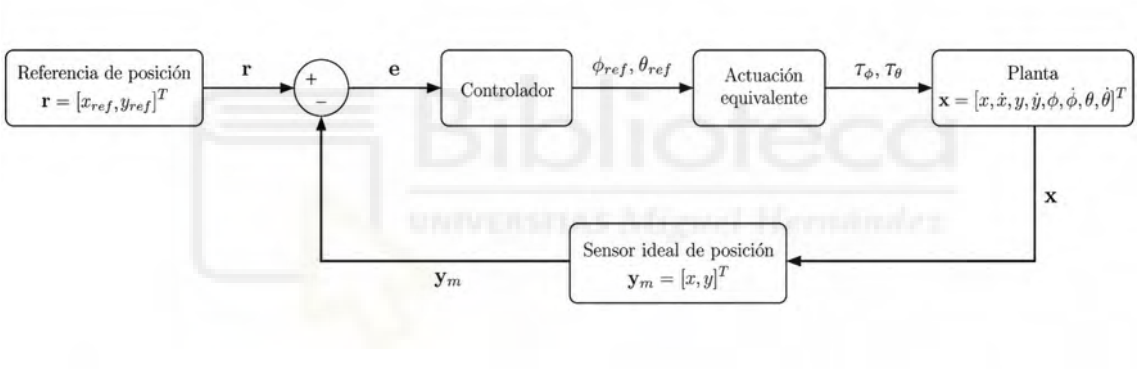


Figura 7.1: Estructura de realimentación empleando un sensor ideal de posición.

Desde el punto de vista de **Simulink**, este sensor puede implementarse mediante un bloque de selección de señales, como un *Selector*, un *Demux* o un subsistema equivalente. La entrada del bloque es el vector de salida de la planta y la salida está formada únicamente por las componentes x e y. Al no añadirse bloques de ruido, retardo, discretización ni cuantización, la medida se comporta como una realimentación continua e ideal.

El uso de este sensor resulta útil como **caso base** para el diseño inicial de los controladores. Al eliminar las no idealidades del sistema de medida, se puede verificar si la estrategia de control es capaz de estabilizar la planta en condiciones nominales. Una vez comprobado este comportamiento, el modelo puede ampliarse mediante la incorporación del sensor visual simulado descrito en el apartado siguiente.

7.4. Sensor visual simulado

Una vez definido el sensor ideal de posición, se introduce un segundo nivel de modelado orientado a representar de forma más realista el comportamiento de un sistema de medida basado en visión artificial. En este caso, no se pretende implementar una cámara física ni desarrollar un algoritmo completo de procesamiento de imagen, sino modelar la señal de medida que proporcionaría dicho sistema una vez estimada la posición de la bola.

Por tanto, el **sensor visual simulado** se plantea como un subsistema que recibe como entrada la posición real de la bola sobre la plataforma y proporciona como salida una medida degradada por las principales limitaciones asociadas a un sistema de adquisición visual. Entre estas limitaciones se consideran la conversión entre coordenadas de imagen y coordenadas físicas, el ruido de medida, la frecuencia de muestreo finita, el retardo temporal y la cuantización debida a la resolución limitada de la cámara [29].

Partiendo de la ecuación 7.4, el modelo de medida no ideal se formula ahora en tiempo discreto, de acuerdo con la naturaleza de una cámara digital. Para ello, se evalúa la señal de la planta en los instantes $t = kT_s$ de acuerdo a la siguiente expresión:

$$\mathbf{y}_m[k] = Q(C_m\mathbf{x}(kT_s - T_d) + \mathbf{v}[k]) \tag{7.6}$$

donde $\mathbf{y}_m[k]$ representa la medida discreta entregada por el sensor visual, C_m es la matriz de selección definida anteriormente, T_s es el periodo de muestreo de la cámara, T_d es el retardo equivalente del sistema de adquisición y procesamiento, $\mathbf{v}[k]$ representa el ruido de medida y $Q(\cdot)$ representa el efecto de cuantización asociado a la resolución finita del sensor.

Esta formulación permite introducir progresivamente las distintas no idealidades del sistema de visión y analizar su influencia sobre el comportamiento del lazo cerrado.

7.4.1. Conversión de coordenadas imagen-plataforma

En un sistema de visión real, la posición de la bola se obtiene inicialmente en coordenadas de imagen, normalmente expresadas en píxeles. Si se considera una cámara cenital fija situada sobre la plataforma, la posición detectada de la bola puede representarse mediante las coordenadas (u, v), donde u y v corresponden a los ejes horizontal y vertical de la imagen.

Para que esta información pueda ser utilizada por el controlador, es necesario transformarla a coordenadas físicas sobre la plataforma, expresadas en metros. Bajo las hipótesis de cámara cenital, plano de trabajo horizontal respecto al plano de imagen, ausencia de distorsión óptica significativa y escala conocida, esta conversión puede aproximarse mediante una transformación lineal:

$$x = s_x(u - u_0)$$

$$y = s_y(v - v_0)$$

donde (u_0, v_0) representa el centro de la imagen, asociado al origen de coordenadas de la plataforma, y s_x, s_y son los factores de escala que relacionan píxeles y metros en cada dirección.

Si la cámara tiene una resolución de $N_x \times N_y$ píxeles y observa una superficie útil de dimensiones $L_x \times L_y$, los factores de escala pueden aproximarse como:

$$s_x = \frac{L_x}{N_x}, \quad s_y = \frac{L_y}{N_y}$$

De este modo, una posición detectada en píxeles puede transformarse en una posición física equivalente sobre la plataforma. En el presente trabajo, al no implementarse un procesamiento real de imagen, esta conversión se modela de forma inversa: se parte de la posición física real (x, y) proporcionada por la planta y se introducen sobre ella las limitaciones equivalentes que tendría una cámara real.

Por tanto, el bloque de conversión imagen-plataforma no representa una etapa de visión artificial completa, sino una abstracción matemática del proceso de medida. Esta simplificación es coherente con el enfoque de simulación adoptado, ya que permite estudiar el efecto de las limitaciones sensoriales sin incorporar la complejidad adicional de algoritmos de detección (véase apartado 2.7.2), segmentación o calibración de cámara.

7.4.2. Ruido de medida

La medida obtenida mediante un sistema de visión artificial no es perfectamente exacta. Incluso en condiciones controladas, la estimación de la posición de la bola puede verse afectada por errores debidos a iluminación, resolución de imagen, segmentación, detección del centroide, vibraciones, reflejos o pequeñas incertidumbres en la calibración.

Para representar este comportamiento, se introduce un término de **ruido de medida** añadido a cada coordenada de posición. El modelo puede expresarse como:

$$x_n(t) = x(t) + n_x(t)$$

$$y_n(t) = y(t) + n_y(t)$$

donde $n_x(t)$ y $n_y(t)$ representan las perturbaciones de medida en cada eje. En una primera aproximación, se consideran señales aleatorias de media nula y distribución gaussiana [30]:

$$n_x(t) \sim \mathcal{N}(0, \sigma_x^2), \quad n_y(t) \sim \mathcal{N}(0, \sigma_y^2)$$

donde σ_x y σ_y son las desviaciones típicas del ruido de medida en las direcciones x e y, respectivamente. En forma vectorial, la medida afectada por ruido puede escribirse como:

$$\mathbf{y}_n(t) = C_m\mathbf{x}(t) + \mathbf{v}(t)$$

con:

$$\mathbf{v}(t) = \begin{bmatrix} n_x(t) \\ n_y(t) \end{bmatrix}$$

La inclusión de este ruido permite evaluar la sensibilidad del controlador frente a errores de medida. Además, justifica la necesidad de emplear técnicas de filtrado o estimación, especialmente cuando se requiera obtener velocidades a partir de señales de posición ruidosas.

Desde el punto de vista de **Simulink**, este efecto puede implementarse mediante un bloque *Random Number* sumado a las señales x e y. La amplitud del ruido debe seleccionarse de forma coherente con la precisión esperada del sistema de visión simulado.

7.4.3. Retardo y frecuencia de muestreo

A diferencia del sensor ideal, un sistema visual real no proporciona medidas continuas en el tiempo. La cámara adquiere imágenes a una determinada frecuencia de muestreo, normalmente expresada en fotogramas por segundo (fps). Por tanto, la posición de la bola solo se actualiza en instantes discretos separados por el periodo de muestreo:

$$T_s = \frac{1}{f_s}$$

donde f_s es la frecuencia de adquisición de la cámara. Entre dos medidas consecutivas, el valor entregado al controlador puede considerarse constante.

Además del muestreo, debe considerarse la existencia de un **retardo temporal** entre el instante en que la bola se encuentra en una determinada posición y el instante en que dicha medida está disponible para el controlador. Este retardo puede deberse al tiempo de exposición de la cámara, adquisición de imagen, procesamiento, conversión de coordenadas y comunicación con el sistema de control.

El efecto del retardo puede representarse como:

$$\mathbf{y}_d(t) = \mathbf{y}_n(t - T_d)$$

donde T_d es el retardo equivalente total del sistema de medida. En el caso discreto, si el retardo es múltiplo del periodo de muestreo, puede expresarse como:

$$T_d = N_d T_s$$

siendo N_d el número de muestras de retardo.

La presencia de retardo es especialmente relevante en sistemas inestables como el *ball and plate*, ya que el controlador actúa sobre información atrasada respecto al estado real de la bola. Un retardo excesivo puede degradar la respuesta dinámica, aumentar el sobreimpulso o incluso comprometer la estabilidad del lazo cerrado. Por este motivo, su inclusión en el modelo de sensor visual simulado resulta importante para evaluar condiciones más próximas a una implementación experimental.

En **Simulink**, la frecuencia de muestreo puede modelarse mediante un bloque *Zero-Order Hold*, mientras que el retardo puede representarse mediante un bloque *Transport Delay* en tiempo continuo o *Integer Delay* en tiempo discreto.

7.4.4. Cuantización y resolución

Otra limitación propia de un sistema de visión artificial es la **resolución finita** de la cámara. La imagen está formada por un número limitado de píxeles, por lo que la posición de la bola no puede determinarse con precisión arbitraria. Aunque los algoritmos de procesamiento pueden estimar posiciones subpíxel, en una primera aproximación resulta razonable modelar la medida como una señal cuantizada.

Si la superficie visible de la plataforma tiene dimensiones L_x y L_y, y la cámara dispone de una resolución de N_x píxeles en horizontal y N_y píxeles en vertical, la resolución espacial aproximada viene dada por:

$$\Delta x = \frac{L_x}{N_x}, \quad \Delta y = \frac{L_y}{N_y}$$

Estos valores representan el incremento mínimo de posición que puede distinguirse en cada eje. Por tanto, la medida cuantizada puede expresarse como:

$$x_q = \Delta x \cdot \text{round}\left(\frac{x}{\Delta x}\right)$$
$$y_q = \Delta y \cdot \text{round}\left(\frac{y}{\Delta y}\right)$$

En forma vectorial, este efecto se representa mediante el operador de cuantización $Q(\cdot)$:

$$\mathbf{y}_q = Q(\mathbf{y}_d)$$

donde $\mathbf{y}_d$ es la señal medida tras considerar ruido, muestreo y retardo.

La cuantización introduce un error de medida acotado, cuyo valor máximo depende del intervalo de cuantización. En cada eje, dicho error puede aproximarse como:

$$|e_{q,x}| \leq \frac{\Delta x}{2}, \quad |e_{q,y}| \leq \frac{\Delta y}{2}$$

Este efecto suele ser pequeño cuando la resolución de la cámara es elevada respecto al tamaño de la plataforma, pero puede adquirir importancia si se trabaja con cámaras de baja resolución o si se requiere una precisión elevada en la posición de la bola.

En **Simulink**, la cuantización puede implementarse mediante un bloque *Quantizer*, definiendo como intervalo de cuantización los valores Δx y Δy. De este modo, el modelo de medida reproduce la pérdida de resolución asociada a la conversión de la posición física continua en una medida digital discreta.

En conjunto, el sensor visual simulado queda formado por una cadena de bloques como el mostrado en la figura 7.2, que introduce progresivamente las principales limitaciones

de una cámara real.

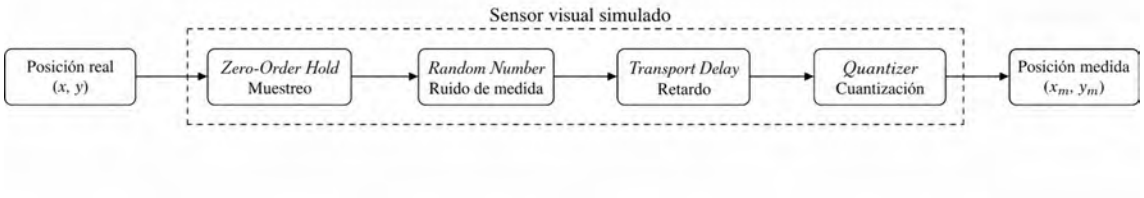


Figura 7.2: Diagrama de bloques del sensor visual simulado.

Esta estructura permite comparar la respuesta del sistema bajo medida ideal y bajo condiciones de medida más realistas, manteniendo al mismo tiempo un modelo suficientemente sencillo para su integración en el entorno de control desarrollado en Simulink.

7.5. Filtrado de la señal medida

La incorporación de un sensor visual simulado permite representar de forma más realista la señal de posición disponible para el controlador. Sin embargo, al introducir ruido, muestreo, retardo y cuantización, la medida deja de coincidir exactamente con la posición real de la bola. Esto puede afectar al comportamiento del lazo cerrado, especialmente si el controlador emplea derivadas de la señal o requiere información del estado completo.

En particular, el sensor visual considerado proporciona directamente únicamente las coordenadas de posición de la bola (ec. 7.2), pero no mide de forma directa las velocidades $\dot{x}$ e $\dot{y}$. Esta limitación resulta relevante, ya que algunas estrategias de control, especialmente las basadas en **realimentación de estados**, pueden requerir el conocimiento de dichas velocidades. Una solución simple consistiría en obtenerlas mediante derivación numérica de las posiciones medidas. No obstante, esta alternativa resulta poco recomendable cuando la señal está afectada por ruido, ya que la derivación tiende a amplificar las componentes de alta frecuencia y puede introducir oscilaciones no deseadas en la acción de control [31].

Por este motivo, resulta conveniente considerar técnicas de **filtrado o estimación de estados** que permitan mejorar la calidad de la señal empleada en la realimentación. En este trabajo se consideran dos enfoques:

1. **Filtro paso bajo:** Una solución sencilla para suavizar la medida de posición.

2. **Filtro de Kalman:** Una alternativa más completa, capaz no solo de reducir el efecto del ruido, sino también de estimar variables no medidas directamente, como las velocidades de la bola.

7.5.1. Filtro paso bajo

El filtro paso bajo constituye una de las técnicas más sencillas para atenuar el ruido de alta frecuencia presente en una señal medida. Su objetivo es permitir el paso de las componentes lentas de la señal, asociadas al movimiento real de la bola, y reducir las variaciones rápidas producidas por el ruido de medida.

Una forma habitual de representar un filtro paso bajo de primer orden en tiempo continuo es mediante la función de transferencia:

$$H(s) = \frac{1}{\tau_f s + 1} \tag{7.7}$$

donde τ_f es la constante de tiempo del filtro. Cuanto **mayor** sea el valor de τ_f, mayor será el suavizado de la señal, aunque también aumentará el retardo introducido por el filtro. Por el contrario, un valor **pequeño** de τ_f permite una respuesta más rápida, pero reduce la capacidad de atenuación del ruido.

Aplicado a las coordenadas medidas de la bola, el filtrado puede expresarse como:

$$x_f(t) = H(s)x_m(t)$$

$$y_f(t) = H(s)y_m(t)$$

donde x_f e y_f representan las señales de posición filtradas. Estas señales pueden emplearse posteriormente para calcular el error de control:

$$\mathbf{e}(t) = \mathbf{r}(t) - \begin{bmatrix} x_f(t) \\ y_f(t) \end{bmatrix}$$

En una **implementación discreta**, más coherente con el comportamiento de una cámara digital, el filtro paso bajo puede implementarse mediante una estructura recursiva de primer orden. Este tipo de filtro se corresponde con los denominados *single pole recursive filters*, ampliamente empleados en procesado digital de señales por su sencillez computacional [31]:

$$x_f[k] = \alpha x_f[k-1] + (1-\alpha)x_m[k]$$

$$y_f[k] = \alpha y_f[k-1] + (1-\alpha)y_m[k]$$

donde α es un parámetro comprendido entre 0 y 1. Valores de α próximos a 1 producen un filtrado más intenso, mientras que valores más bajos permiten que la señal filtrada siga más rápidamente a la medida.

Aunque el filtro paso bajo permite reducir el ruido de medida, presenta una limitación importante: **introduce desfase o retardo adicional** en la señal. En sistemas dinámicos inestables, como el sistema bola-plataforma, este retardo puede afectar negativamente a la estabilidad y al tiempo de respuesta del lazo cerrado. Por ello, la elección de la constante de tiempo del filtro debe realizarse buscando un compromiso entre suavizado de la señal y rapidez de respuesta.

Además, el filtro paso bajo no resuelve por sí mismo el problema de la estimación de velocidades. Si se desea obtener $\dot{x}$ e $\dot{y}$, sería necesario derivar posteriormente la señal filtrada o emplear una estrategia adicional de estimación. Por este motivo, aunque resulta útil como primera aproximación, puede ser insuficiente para controladores que requieran información completa del estado.

Desde el punto de vista de **Simulink**, este filtrado puede implementarse mediante un bloque de función de transferencia de primer orden, un filtro discreto o un subsistema que implemente directamente la ecuación recursiva anterior.

7.5.2. Filtro de Kalman

El filtro de Kalman se plantea como una alternativa más adecuada cuando el objetivo no es únicamente suavizar la medida, sino también **estimar variables del estado** que no son medidas directamente.

A diferencia de un filtro paso bajo convencional, el filtro de Kalman utiliza un modelo dinámico del sistema para predecir la evolución del estado y corrige dicha predicción mediante las medidas disponibles. De esta forma, combina la información proporcionada por el modelo de espacio de estados con la información procedente del sensor visual.

Considerando el modelo lineal de la planta:

$$\dot{\mathbf{x}}(t) = A\mathbf{x}(t) + B\mathbf{u}(t)$$

$$\mathbf{y}_m(t) = C_m\mathbf{x}(t) + \mathbf{v}(t)$$

donde $\mathbf{u}(t)$ representa el vector de entradas de la planta y $\mathbf{v}(t)$ el ruido de medida, el filtro de Kalman permite obtener una estimación del vector de estado:

$$\hat{\mathbf{x}} = \begin{bmatrix} \hat{x} & \hat{\dot{x}} & \hat{y} & \hat{\dot{y}} & \hat{\phi} & \hat{\dot{\phi}} & \hat{\theta} & \hat{\dot{\theta}} \end{bmatrix}^T \tag{7.8}$$

En el caso del sensor visual considerado, la matriz de medida C_m (definida en la ecuación 7.3) selecciona únicamente las coordenadas de posición de la bola, que son las que recibe el filtro; y, a partir de ellas y del modelo dinámico, estima también las velocidades $\hat{\dot{x}}$ e $\hat{\dot{y}}$. Esta característica resulta especialmente útil para evitar el uso de derivadas numéricas directas sobre señales ruidosas.

Siguiendo la formulación del filtro discreto de Kalman descrita por Pérez Vidal en el contexto de la estimación de posición, velocidad y aceleración de objetos mediante información visual [27], el modelo empleado por el estimador se expresa como:

$$\begin{aligned} \mathbf{x}[k+1] &= A_d\mathbf{x}[k] + B_d\mathbf{u}[k] + \mathbf{w}[k] \\ \mathbf{y}_m[k] &= C_m\mathbf{x}[k] + \mathbf{v}[k] \end{aligned} \tag{7.9}$$

donde A_d y B_d son las matrices discretizadas del modelo de la planta desarrollado en el capítulo 4.3.4, $\mathbf{w}[k]$ representa el ruido de proceso y $\mathbf{v}[k]$ el ruido de medida. Estos términos se caracterizan mediante sus matrices de covarianza:

$$Q_k = E\{\mathbf{w}[k]\mathbf{w}^T[k]\} \tag{7.10}$$

$$R_k = E\{\mathbf{v}[k]\mathbf{v}^T[k]\} \tag{7.11}$$

La matriz Q_k representa la incertidumbre asociada al modelo dinámico, mientras que R_k representa la incertidumbre de las medidas proporcionadas por el sensor. La elección de estas matrices determina el comportamiento del estimador:

- Si se asigna una **covarianza de medida elevada**, el filtro confiará más en el modelo que en la señal medida.
- Si se considera que la **medida es muy precisa**, el estimador seguirá más de cerca las mediciones del sensor.

En el presente trabajo, el filtro de Kalman se plantea principalmente como **estimador de estados**. Su función permite reconstruir una estimación coherente del estado de la bola a partir de medidas parciales. De este modo, si el controlador requiere información de velocidad, puede emplearse:

$$\hat{\mathbf{x}}_{bola} = \begin{bmatrix} \hat{x} \\ \hat{\dot{x}} \\ \hat{y} \\ \hat{\dot{y}} \end{bmatrix} \tag{7.12}$$

en lugar de recurrir a la derivación numérica directa, que sería más sensible al ruido:

$$\dot{x} \approx \frac{x_m[k] - x_m[k-1]}{T_s} \quad , \quad \dot{y} \approx \frac{y_m[k] - y_m[k-1]}{T_s}$$

Desde el punto de vista de **Simulink**, el filtro de Kalman puede implementarse mediante el bloque correspondiente disponible en el entorno de control. Se requiere definir el modelo en espacio de estados (A_d, B_d, C_m y D_m), así como las covarianzas Q_k y R_k. En este caso, la matriz D_m puede considerarse nula, ya que la cámara no mide directamente el efecto instantáneo de los pares de entrada sobre la posición de la bola:

$$D_m = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \tag{7.13}$$

Por tanto, la incorporación del filtro de Kalman resulta directa desde el punto de vista estructural, al estar ya disponible el modelo de espacio de estados de la planta. La principal dificultad reside en la selección adecuada de las matrices de covarianza Q_k y R_k, que deberán ajustarse en función del nivel de ruido considerado y de la confianza asignada al modelo.

7.6. Integración del sensor en Simulink

Una vez definidos los modelos de medida descritos en los apartados anteriores, se procede a su implementación en el entorno Simulink. En esta fase del trabajo todavía no se ha cerrado el lazo de control. Por tanto, la integración del sensor se realiza inicialmente en **lazo abierto**, utilizando el modelo de espacio de estados como generador de la posición real de la bola.

La estructura general de la prueba implementada puede resumirse como se muestra en la figura 7.3.

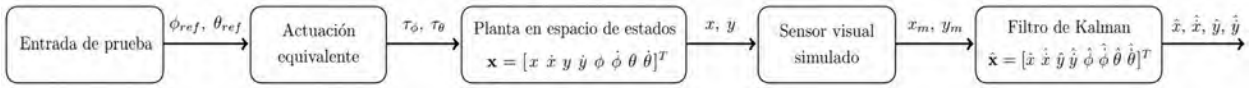


Figura 7.3: Esquema de la prueba en lazo abierto empleada para validar el sensor visual simulado y el filtro de Kalman.

7.6.1. Estructura general del sensor simulado

El sensor visual simulado se ha implementado de forma modular, de manera que pueda modificarse fácilmente su nivel de realismo sin alterar el modelo de la planta. En primer lugar, se extraen las señales x e y de la salida del modelo de espacio de estados. Posteriormente, a dichas señales se les aplica varias etapas de procesado: muestreo, ruido de medida, retardo y cuantización; de acuerdo a la figura 7.2 ya vista.

En **Simulink**, esta cadena se implementa mediante los bloques *Zero-Order Hold*, *Random Number*, *Transport Delay* y *Quantizer* [32]. Se utilizan dos ramas independientes, una para el eje X y otra para el eje Y, con el fin de poder introducir ruido independiente en cada coordenada. Una vez procesadas ambas señales, estas se agrupan mediante un bloque *Mux* para formar el vector de medida:

$$\mathbf{y}_m = \begin{bmatrix} x_m \\ y_m \end{bmatrix}$$

La **Figura 7.4** muestra el esquema completo implementado en Simulink para la prueba en lazo abierto del sensor visual simulado y del filtro de Kalman.

Esta organización permite comparar directamente tres tipos de señales: la salida ideal del modelo de espacio de estados, la señal medida por el sensor visual simulado y la señal estimada por el filtro de Kalman.

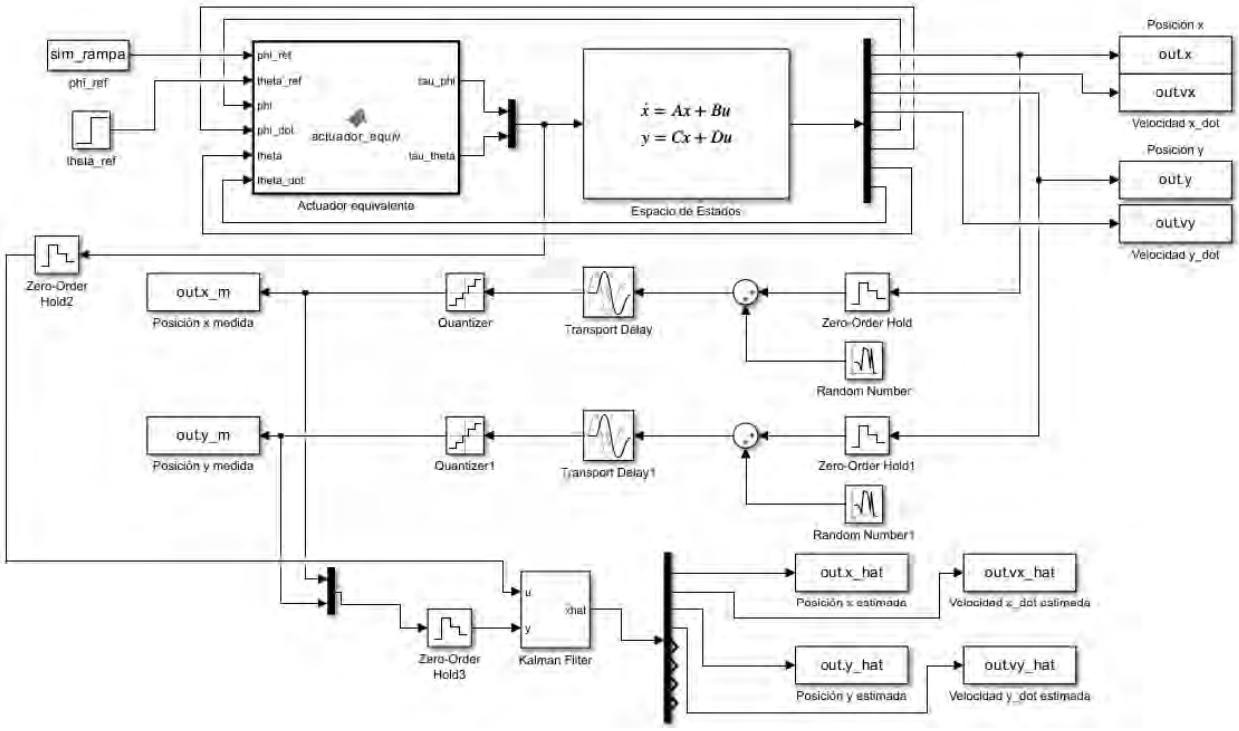


Figura 7.4: Integración del sensor simulado y del filtro de Kalman en Simulink.

7.6.2. Configuración de los bloques del sensor visual simulado

Los parámetros del sensor visual simulado se han seleccionado de forma que representen una cámara convencional, sin introducir unas condiciones excesivamente restrictivas. En concreto, se ha considerado una cámara con una frecuencia de adquisición de **60 fps**, lo que equivale a un periodo de muestreo:

$$T_s = \frac{1}{60} \approx 0{,}0167\,\text{s}$$

Este valor se configura en los bloques *Zero-Order Hold* utilizados para representar el muestreo de la cámara. Además, se emplea el mismo tiempo de muestreo en los bloques de ruido aleatorio y en el filtro de Kalman, garantizando así la coherencia temporal entre la medida y el estimador.

El ruido de medida se modela mediante bloques *Random Number*, uno por cada coordenada. Se considera una señal aleatoria discreta de media nula y desviación típica:

$$\sigma = 0{,}0005\,\text{m} \quad (0{,}5\,\text{mm})$$

Por tanto, la varianza configurada en los bloques de ruido es:

$$\sigma^2 = 2{,}5 \cdot 10^{-7}$$

Para evitar que el ruido introducido en ambos ejes sea idéntico, se emplean semillas (*seeds*) distintas en los generadores aleatorios de X e Y. La semilla permite reproducir la misma secuencia aleatoria en simulaciones sucesivas, lo que facilita la comparación entre distintos ensayos.

El retardo asociado al proceso de adquisición y procesamiento de imagen se representa mediante un bloque *Transport Delay*. Se ha considerado un retardo equivalente:

$$T_d = 0{,}02\,\text{s}$$

Este valor representa una latencia moderada, asociada al tiempo de exposición, procesamiento de imagen y comunicación de la medida.

Por último, la cuantización se introduce mediante el bloque *Quantizer*. Considerando un radio de plataforma de 120 mm y una resolución de cámara suficientemente alta, se ha empleado un intervalo de cuantización:

$$\Delta = 5 \cdot 10^{-4}\,\text{m} \quad (0{,}5\,\text{mm})$$

Este valor permite representar la resolución finita de la medida sin introducir una degradación excesiva de la señal.

La Tabla 7.1 resume los valores empleados en la configuración del sensor visual simulado.

Tabla 7.1: Parámetros empleados en el sensor visual simulado.

Bloque	**Parámetro**	**Valor**
Zero-Order Hold	Tiempo de muestreo	$T_s = 1/60\,\text{s}$
Random Number	Media	0
Random Number	Varianza	$2{,}5 \cdot 10^{-7}$
Random Number	Tiempo de muestreo	$T_s = 1/60\,\text{s}$
Transport Delay	Retardo	$T_d = 0{,}02\,\text{s}$
Transport Delay	Salida inicial	0
Quantizer	Intervalo de cuantización	$\Delta = 5 \cdot 10^{-4}\,\text{m}$

7.6.3. Integración del filtro de Kalman

A la salida del sensor visual simulado se incorpora un filtro de Kalman con el objetivo de estimar las variables de estado no medidas directamente.

El modelo empleado por el filtro se obtiene a partir del modelo lineal en espacio de estados desarrollado previamente. Puesto que el sensor visual trabaja en tiempo discreto, el modelo continuo se discretiza con el periodo de muestreo (T_s) de la cámara.

El modelo discreto utilizado por el estimador queda definido como se indicó en las ecuaciones 7.9, con $\mathbf{u}[k]$ representando las entradas conocidas aplicadas a la planta:

$$\mathbf{u}[k] = \begin{bmatrix} \tau_\phi[k] \\ \tau_\theta[k] \end{bmatrix}$$

, C_m es la matriz de medida (ec. 7.3) asociada al sensor visual y la matriz D_m se considera nula.

Para la configuración del filtro se emplean las matrices A_d, B_d, C_m y D_m, junto con las matrices de covarianza del ruido de proceso y de medida. La covarianza de medida se define a partir de la desviación típica del ruido introducido en el sensor [30]:

$$R_k = \begin{bmatrix} \sigma^2 & 0 \\ 0 & \sigma^2 \end{bmatrix}$$

mientras que la covarianza del ruido de proceso se selecciona de forma diagonal, asignando una incertidumbre ligeramente superior a las componentes de velocidad:

$$Q_k = \text{diag}\left(10^{-8}, 10^{-6}, 10^{-8}, 10^{-6}, 10^{-8}, 10^{-6}, 10^{-8}, 10^{-6}\right)$$

La matriz de covarianza cruzada entre ruido de proceso y ruido de medida se toma nula:

$$N_k = 0$$

En **Simulink** se implementa mediante el bloque *Kalman Filter* [33]. Para su configuración se emplea un script en MATLAB, adjuntado en el Anexo A.6. El filtro proporciona como

salida una estimación del vector de estado completo:

$$\hat{\mathbf{x}} = \begin{bmatrix} \hat{x} & \hat{\dot{x}} & \hat{y} & \hat{\dot{y}} & \hat{\phi} & \hat{\dot{\phi}} & \hat{\theta} & \hat{\dot{\theta}} \end{bmatrix}^T$$

En la validación del sistema de medida se emplean principalmente las cuatro primeras componentes, correspondientes a la posición y velocidad estimadas de la bola.

7.6.4. Pruebas en lazo abierto

Con el objetivo de comprobar el funcionamiento del subsistema de medida y del estimador, se realizan varias simulaciones en lazo abierto. En estas pruebas no se emplea todavía un controlador, sino que se aplican directamente señales de referencia a la orientación de la plataforma. Estas referencias se transforman en pares equivalentes mediante el bloque de actuación equivalente y se introducen en el modelo de espacio de estados.

Durante las simulaciones se almacenan en el *workspace* las siguientes señales:

- $x, y, \dot{x}, \dot{y}$ correspondientes a la salida ideal del modelo de espacio de estados.
- x_m, y_m correspondientes a la posición medida por el sensor visual simulado.
- $\hat{x}, \hat{y}, \hat{\dot{x}}, \hat{\dot{y}}$ correspondientes a la salida del filtro de Kalman.

A partir de estas señales se realizan comparaciones temporales entre la posición real, la posición medida y la posición estimada:

$$x(t), \quad x_m(t), \quad \hat{x}(t)$$

$$y(t), \quad y_m(t), \quad \hat{y}(t)$$

También se comparan las velocidades reales del modelo de espacio de estados con las velocidades estimadas por el filtro:

$$\dot{x}(t), \quad \hat{\dot{x}}(t)$$

$$\dot{y}(t), \quad \hat{\dot{y}}(t)$$

Se han considerado los mismos estímulos de referencia usados para comparar los modelos del espacio de estados y de Simscape Multibody (véase sección 6.2.3), es decir, distintos escalones cuya amplitud representa la orientación de la plataforma.

También se ha aplicado una secuencia de rampas en la referencia de orientación ϕ_{ref} (figura 7.5). Esta entrada permite generar una trayectoria más variable de la bola sobre el eje X de la plataforma y comprobar si el filtro de Kalman es capaz de seguir correctamente la evolución de la posición y la velocidad en condiciones dinámicas más exigentes. La orientación θ_{ref} se ha mantenido nula.

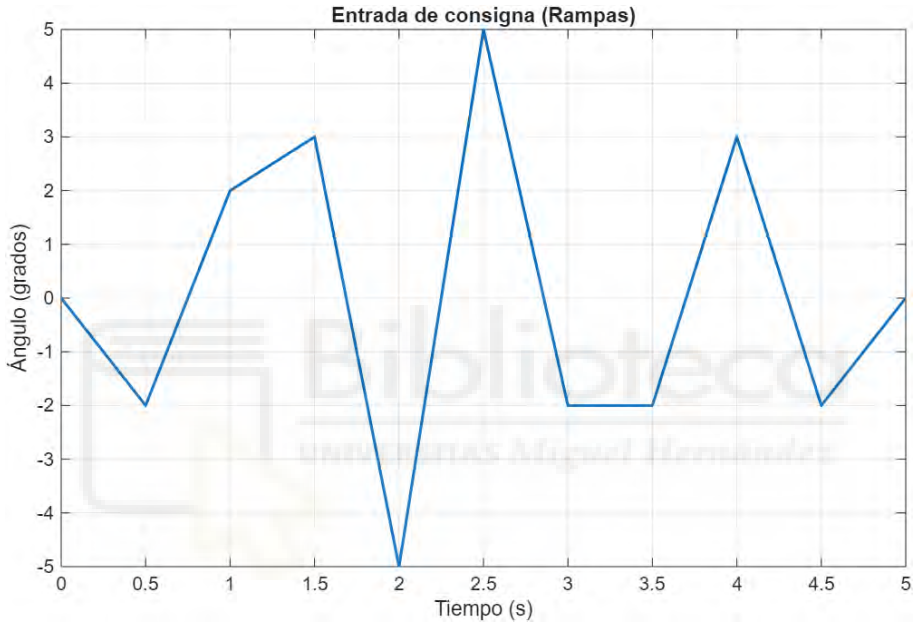


Figura 7.5: Entrada de referencia conformada por una secuencia de rampas.

Las figuras 7.6 y 7.7 muestran la comparación de la posición de la bola en el eje X e Y, respectivamente.

Por otra parte, la comparación entre las velocidades del modelo de espacio de estados y la estimada por el filtro se estudian en la figura 7.8 (eje X) y en la figura 7.9 (eje Y).

Con el fin de cuantificar la precisión del estimador, en la figura 7.10 se muestra la evolución temporal del error entre las variables reales proporcionadas por el modelo de espacio de estados y las variables estimadas por el filtro de Kalman. Además, para cada variable se incluye el valor RMSE[3] correspondiente.

[3]El RMSE (*Root Mean Square Error*), o raíz del error cuadrático medio, es una métrica escalar que resume la diferencia media entre una señal real y su estimación. Se calcula como $RMSE = \sqrt{\frac{1}{N}\sum_{k=1}^{N}(z[k] - \hat{z}[k])^2}$.

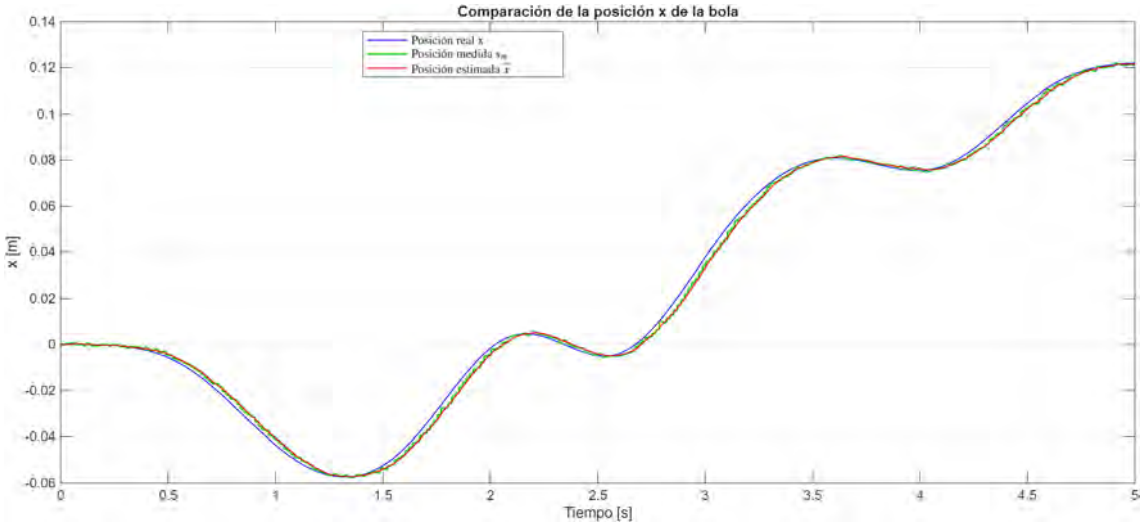


Figura 7.6: Comparación de la posición x de la bola.

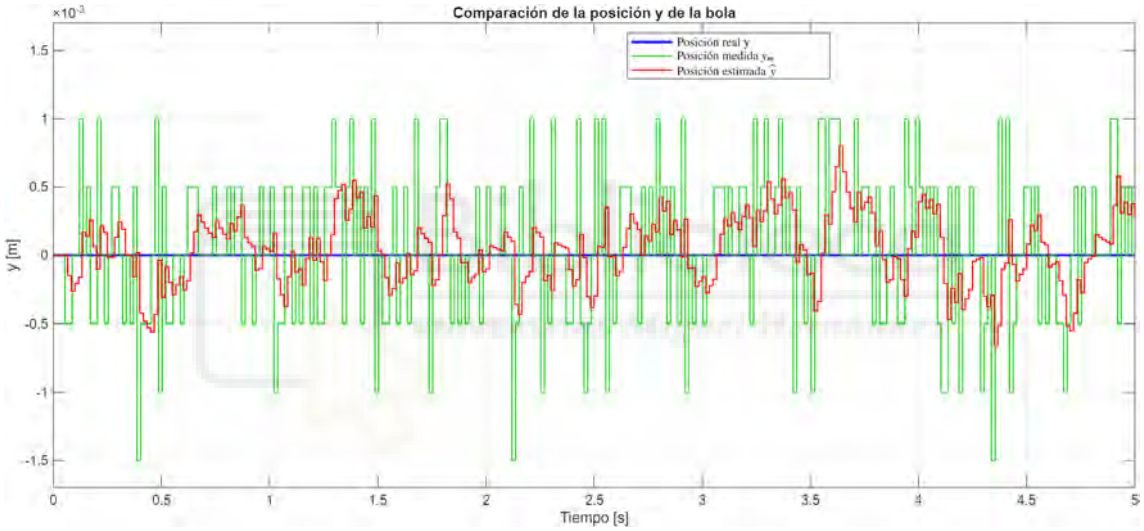


Figura 7.7: Comparación de la posición y de la bola.

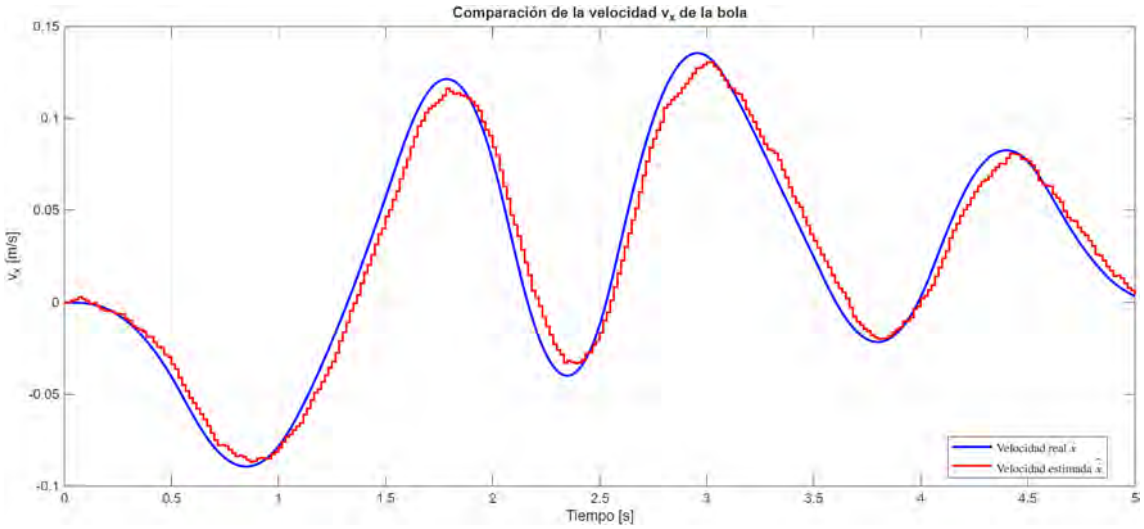


Figura 7.8: Comparación de la velocidad $\dot{x}$ de la bola.

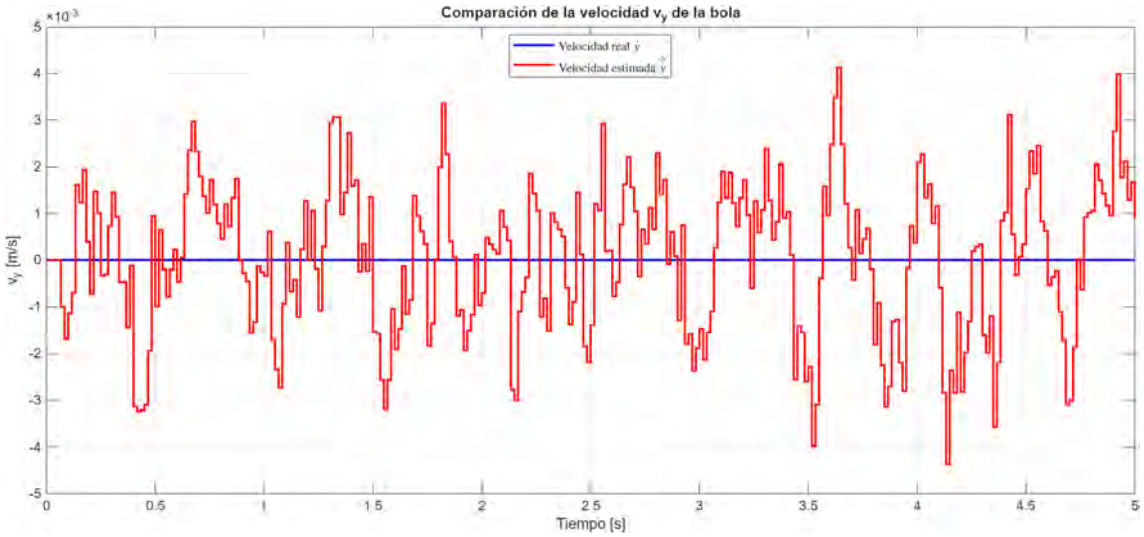


Figura 7.9: Comparación de la velocidad $\dot{y}$ de la bola.

En el Anexo A.7 se adjunta el código en MATLAB para proporcionar las gráficas mostradas en el presente apartado.

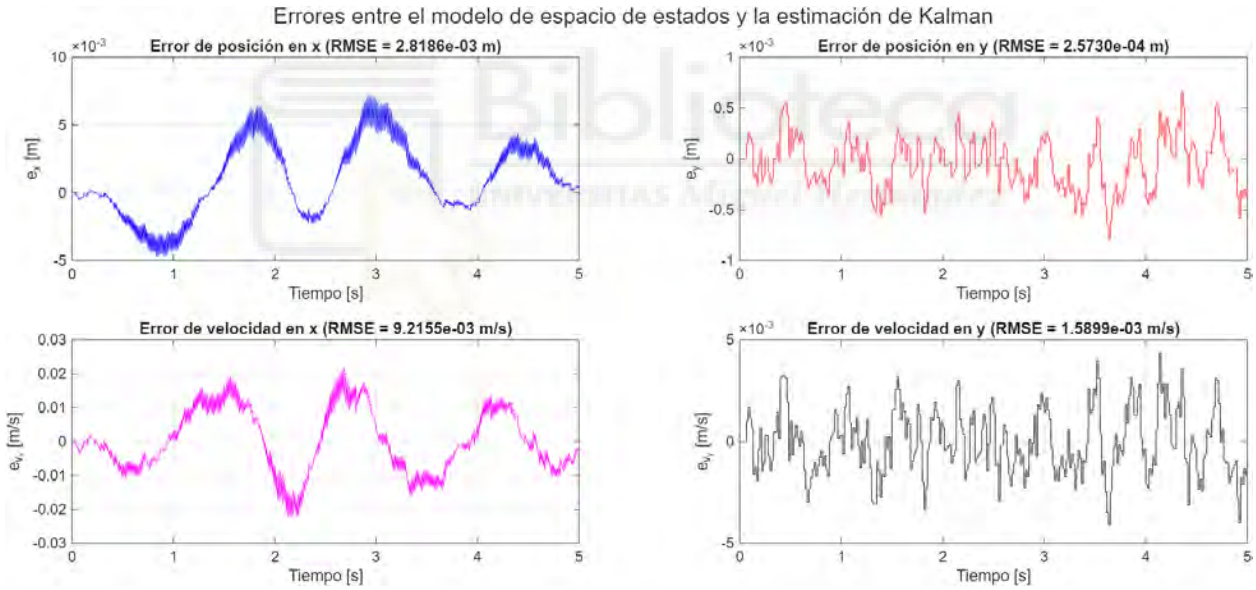


Figura 7.10: Errores entre las variables reales del modelo de espacio de estados y las variables estimadas por el filtro de Kalman.

7.6.5. Análisis de resultados

A partir de los resultados obtenidos se comprueba que el subsistema de medida implementado reproduce correctamente el comportamiento esperado de una cámara simulada. En la comparación de la posición en el eje X, figura 7.6, se observa que la posición medida x_m sigue de forma próxima a la posición real x, aunque incorpora las pequeñas variaciones

asociadas al muestreo, al ruido de medida, al retardo y a la cuantización. La estimación proporcionada por el filtro de Kalman, $\hat{x}$, mantiene una evolución muy cercana a la señal real, suavizando parcialmente las perturbaciones introducidas por el sensor.

La respuesta en el eje Y, figura 7.7, resulta especialmente útil para comprobar el desacoplo entre ejes. Dado que durante esta prueba se mantiene $\theta_{ref} = 0$, no se introduce una excitación directa sobre dicho eje. En consecuencia, la posición real y permanece prácticamente nula. Las pequeñas variaciones observadas en la medida y_m proceden principalmente del ruido y de la cuantización del sensor visual simulado, mientras que la señal estimada $\hat{y}$ se mantiene próxima al valor real. Esto indica que la excitación aplicada en ϕ_{ref} afecta únicamente al eje X, sin generar un acoplamiento significativo sobre el eje Y en el modelo empleado.

En cuanto a la estimación de velocidades, la figura 7.8 muestra que $\hat{\dot{x}}$ reproduce adecuadamente la evolución de $\dot{x}$. Se aprecia un ligero suavizado y un pequeño desfase, propios de un estimador discreto que trabaja con medidas afectadas por no idealidades. Aun así, la señal estimada conserva los cambios de tendencia y de amplitud asociados a la secuencia de rampas, por lo que resulta válida como aproximación de la velocidad real de la bola.

Por su parte, la figura 7.9 muestra que la velocidad real $\dot{y}$ permanece cercana a cero, de forma coherente con la ausencia de excitación en θ_{ref}. La estimación $\hat{\dot{y}}$ presenta pequeñas oscilaciones alrededor de cero, con una amplitud reducida.

Los **errores** mostrados en la figura 7.10 permiten cuantificar estas observaciones. En el eje X, donde se concentra el movimiento de la bola, se obtienen los mayores errores, con un RMSE de $2{,}8186 \cdot 10^{-3}$ m en posición y $9{,}2155 \cdot 10^{-3}$ m/s en velocidad. En el eje Y, al no existir excitación directa, los errores son menores: $2{,}5730 \cdot 10^{-4}$ m para la posición y $1{,}5899 \cdot 10^{-3}$ m/s para la velocidad. Esta diferencia entre ejes es coherente con el hecho de que la excitación se aplica únicamente sobre ϕ_{ref}, produciendo un movimiento dominante en la dirección X.

En conjunto, los resultados **validan la integración** del sensor visual simulado y del filtro de Kalman en Simulink. La medida simulada reproduce de forma sencilla las limitaciones esperables de una cámara, mientras que el estimador permite reconstruir de forma coherente tanto la posición como la velocidad de la bola. Esta última característica resulta

especialmente relevante para el posterior diseño de controladores por realimentación de estados, donde será necesario disponer de variables que no serían medidas directamente por un sistema de visión artificial.

8. CONTROLADOR PID

8.1. Introducción

Una vez desarrollado el modelo matemático del sistema bola-plataforma y definido el sistema de medida necesario para cerrar el lazo de realimentación, el siguiente paso consiste en diseñar una estrategia de control capaz de estabilizar la bola en el centro de la plataforma. En este capítulo se aborda el diseño, ajuste e implementación de una estrategia de control basada en la estructura PID clásica, aplicada al sistema propuesto.

El **controlador PID** constituye una de las estrategias de control clásico más extendidas tanto en el ámbito industrial como académico, debido a su simplicidad estructural, facilidad de implementación y capacidad para proporcionar un comportamiento satisfactorio en una amplia variedad de sistemas dinámicos. Por este motivo, se emplea en este trabajo como primera estrategia de control, estableciendo una solución de referencia sobre la que analizar el comportamiento del sistema en lazo cerrado. No obstante, como se justificará durante la sintonización, la acción integral no resulta conveniente para la planta considerada, por lo que el controlador finalmente implementado corresponde a un **PD con derivada filtrada**, equivalente a un PID con ganancia integral nula.

En una primera fase, las simulaciones se realizan empleando el sensor ideal de posición definido en el capítulo anterior. De este modo, las variables medidas coinciden directamente con las variables reales del modelo, permitiendo analizar el comportamiento del controlador sin introducir inicialmente efectos asociados al sistema de medida. Posteriormente, el mismo controlador se valida empleando el sensor visual simulado, que incorpora no idealidades; permitiendo evaluar la influencia de un sistema de medida más realista sobre el comportamiento del lazo cerrado.

Finalmente, se presentan y analizan los resultados obtenidos para los distintos escenarios de medida considerados y se extraen las principales conclusiones del capítulo.

8.2. Planteamiento y estructura del control PID

En este apartado se define la estructura general del PID empleado para controlar la posición de la bola. Para ello, se parte del esquema general de lazo cerrado presentado en la figura 8.1, en el que la posición medida de la bola se compara con una referencia deseada y el controlador genera las inclinaciones necesarias para corregir el error de posición.

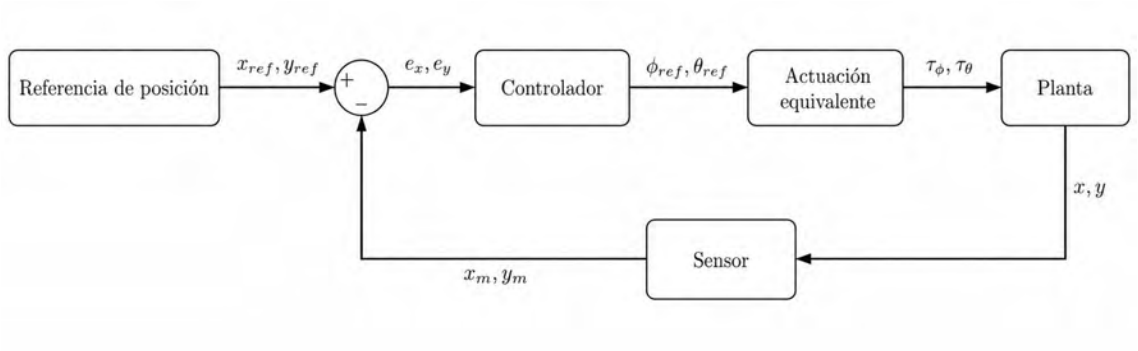


Figura 8.1: Esquema general de control del sistema bola-plataforma.

La planta empleada para el diseño inicial del controlador es el **modelo en espacio de estados** desarrollado en el capítulo 4. Esta elección se justifica por el menor coste computacional de dicho modelo y por su validez en torno al punto de equilibrio, especialmente para pequeñas inclinaciones de la plataforma y desplazamientos moderados de la bola.

A diferencia de una estrategia de control aplicada directamente sobre los pares equivalentes del modelo dinámico, el controlador diseñado en este capítulo actúa sobre las **referencias de orientación** de la plataforma. De este modo, el problema de control se plantea en términos de la relación entre la posición de la bola y la inclinación del plato, manteniendo el **bloque de actuación equivalente** como etapa intermedia entre el controlador y la planta.

Esta estructura permite separar el diseño del controlador de la generación de los pares equivalentes, manteniendo una arquitectura modular coherente con los modelos desarrollados en capítulos anteriores. Además, facilita la posterior comparación entre diferentes estrategias de control, ya que todas ellas pueden generar referencias de orientación para la plataforma y utilizar el mismo bloque de actuación equivalente.

8.2.1. Objetivo de control y estructura general del lazo

El objetivo principal del controlador PID es **estabilizar la bola en el centro de la plataforma**. Para ello, se define como referencia de posición el origen del sistema de coordenadas asociado al plato, de forma que:

$$x_{ref} = 0$$
$$y_{ref} = 0$$

donde x_{ref} e y_{ref} representan las coordenadas deseadas de la bola sobre la superficie de la plataforma.

La posición real de la bola viene dada por las variables x e y, obtenidas a la salida de la planta. Sin embargo, en un sistema de control en lazo cerrado, el controlador no utiliza directamente estas variables reales, sino las señales proporcionadas por el sistema de medida. Por este motivo, se definen x_m e y_m como las posiciones medidas de la bola.

A partir de estas señales, los errores de posición se calculan como:

$$\begin{aligned} e_x(t) &= x_{ref}(t) - x_m(t) \\ e_y(t) &= y_{ref}(t) - y_m(t) \end{aligned} \tag{8.1}$$

Estos errores representan la desviación de la bola respecto a la posición deseada y constituyen las señales de entrada del controlador PID. Cuando la bola se encuentra exactamente en el centro de la plataforma, las señales medidas coinciden con las referencias y, por tanto, los errores de posición son nulos.

A partir de los errores $e_x(t)$ y $e_y(t)$, el controlador PID calcula las referencias de inclinación de la plataforma $\phi_{ref}(t)$ y $\theta_{ref}(t)$ y, a través del bloque de actuación equivalente, se generan los pares equivalentes $\tau_\phi(t), \tau_\theta(t)$.

Dichos pares son finalmente aplicados a la planta, cuya salida proporciona la evolución temporal de la posición de la bola. El sensor toma esta información y genera las señales medidas $x_m(t)$ e $y_m(t)$, que se emplean de nuevo para calcular el error de posición, cerrando así el lazo de control.

De forma resumida, el flujo de señales puede expresarse como:

$$(x_{ref}, y_{ref}) \rightarrow (e_x, e_y) \rightarrow \text{PID} \rightarrow (\phi_{ref}, \theta_{ref}) \rightarrow \text{Actuación equivalente} \rightarrow (\tau_\phi, \tau_\theta) \rightarrow \text{Planta}$$

En el caso de las simulaciones con sensor ideal, las variables medidas coinciden directamente con las variables reales de la planta:

$$x_m(t) = x(t)$$
$$y_m(t) = y(t)$$

Por el contrario, cuando se emplea el sensor visual simulado, las señales x_m e y_m incluyen las **no idealidades** asociadas al proceso de medida, tales como ruido, retardo, cuantización y frecuencia de muestreo limitada. Esta distinción permite analizar en primer lugar el comportamiento ideal del controlador y, posteriormente, evaluar su respuesta ante condiciones de medida más realistas.

8.2.2. Control independiente por ejes y desacoplamiento aproximado

El sistema bola-plataforma presenta una naturaleza multivariable, ya que el movimiento de la bola sobre el plato depende de la inclinación de la superficie en dos direcciones. Además, la arquitectura paralela de la plataforma introduce acoplamientos entre las variables de orientación y las acciones de los actuadores.

No obstante, bajo la hipótesis de pequeñas inclinaciones y considerando el funcionamiento del sistema alrededor del punto de equilibrio, el comportamiento dinámico puede aproximarse mediante **dos subsistemas desacoplados**, asociados respectivamente a los movimientos de la bola en los ejes X e Y. Esta aproximación permite simplificar el diseño del controlador y tratar inicialmente el sistema como dos problemas de control independientes.

De este modo, como se muestra en la figura 8.2, se plantea una estrategia de control descentralizada formada por **dos controladores PID**, uno para cada eje de movimiento de la bola. El primer controlador actúa sobre el error de posición en el eje X, mientras que el segundo actúa sobre el error de posición en el eje Y. Las salidas de ambos controladores

se interpretan como referencias de inclinación de la plataforma.

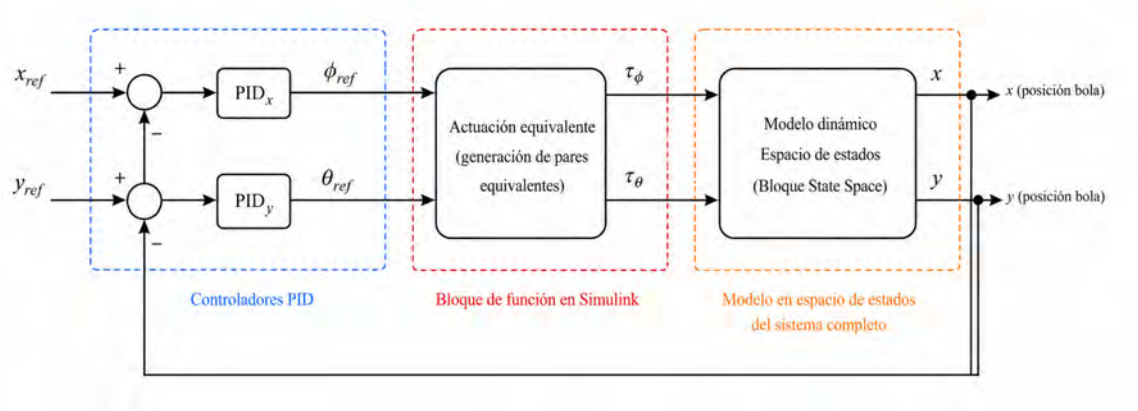


Figura 8.2: Esquema de control para un PID por eje.

Dado que la plataforma se considera geométricamente simétrica y el sistema se linealiza alrededor del punto central de operación, se asume que la dinámica en ambos ejes es equivalente. En consecuencia, en los próximos apartados se empleará el mismo conjunto de parámetros PID para ambos controladores.

8.2.3. Obtención de funciones de transferencia equivalentes

Con el objetivo de facilitar la sintonización de los controladores PID, resulta conveniente obtener funciones de transferencia equivalentes para cada uno de los subsistemas desacoplados. Para ello, se parte del modelo lineal en espacio de estados desarrollado en el Capítulo 4, cuya forma general viene dada por:

$$\begin{aligned} \dot{\mathbf{x}}(t) &= A\mathbf{x}(t) + B\mathbf{u}(t) \\ y(t) &= C\mathbf{x}(t) + D\mathbf{u}(t) \end{aligned} \tag{8.2}$$

donde $\mathbf{x}(t)$ representa el vector de estados, $\mathbf{u}(t)$ el vector de entradas, $y(t)$ el vector de salidas y A, B, C y D son las matrices que definen el sistema linealizado (véase 4.3.4).

La representación en espacio de estados resulta adecuada para describir el comportamiento multivariable completo del sistema. Sin embargo, para el diseño inicial de controladores PID es útil obtener una representación mediante funciones de transferencia, ya que esta permite aplicar técnicas clásicas de análisis y sintonización en sistemas SISO [26]. La conversión se realiza en MATLAB mediante el comando:

```
[num, den] = ss2tf(A, B, C, D, iu)
```

donde *iu* indica el índice de la entrada considerada. Este procedimiento permite obtener las funciones de transferencia asociadas a cada entrada del modelo y seleccionar los canales principales que relacionan los pares equivalentes con la posición de la bola.

No obstante, en la estructura de control propuesta en la figura 8.1, el PID no genera directamente los pares equivalentes τ_ϕ y τ_θ, sino las referencias de inclinación de la plataforma, ϕ_{ref} y θ_{ref}. Por este motivo, resulta más adecuado emplear una función de transferencia equivalente que incluya tanto el bloque de actuación equivalente como la planta. De este modo, para el eje X se considera la relación:

$$G_{eq,x}(s) = \frac{X(s)}{\Phi_{ref}(s)}$$

y, de forma análoga, para el eje Y:

$$G_{eq,y}(s) = \frac{Y(s)}{\Theta_{ref}(s)}$$

Estas funciones representan la dinámica vista por el controlador PID. El esquema equivalente empleado para la sintonización se muestra en la figura 8.3.

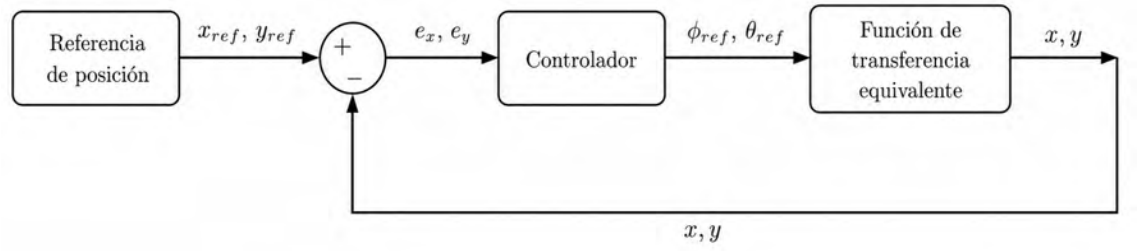


Figura 8.3: Esquema SISO equivalente empleado para la sintonización del controlador PID.

Debido a la simetría y a la linealización, se obtiene la misma función de transferencia para ambos ejes: $G_{eq}(s) = G_{eq,x}(s) = G_{eq,y}(s)$. El cálculo se realiza en MATLAB y el

procedimiento detallado se recoge en el Anexo A.8, obteniendo:

$$G_{eq}(s) = \frac{2803s^3 + 6{,}269 \cdot 10^4 s^2 + 4{,}674 \cdot 10^5 s + 1{,}162 \cdot 10^6}{s^7 + 62{,}37s^6 + 1461s^5 + 1{,}603 \cdot 10^4 s^4 + 8{,}328 \cdot 10^4 s^3 + 1{,}658 \cdot 10^5 s^2} \quad (8.3)$$

Tras ello, se procede a validar dicha función en Simulink, comparando su respuesta con la del modelo conjunto del bloque de actuación equivalente y espacio de estados (figura 8.4).

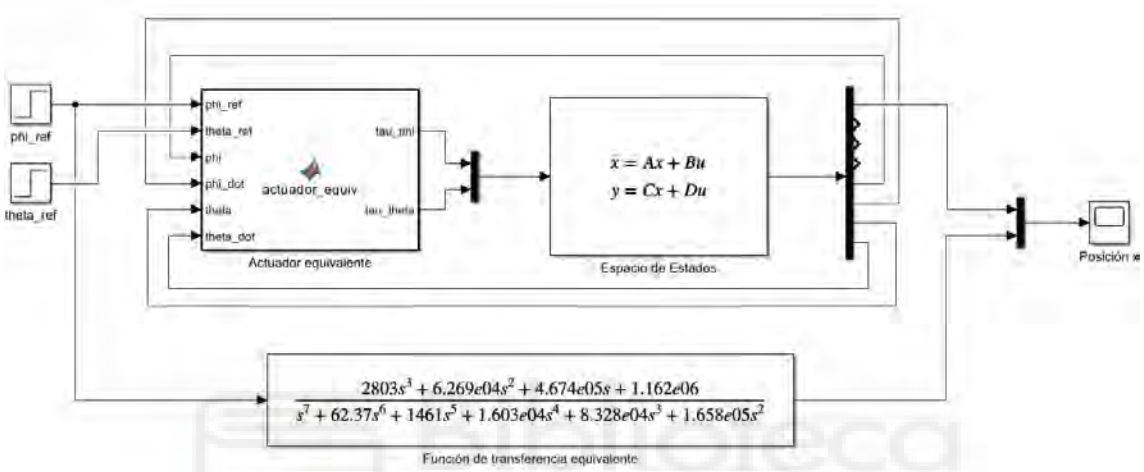


Figura 8.4: Validación de la función de transferencia equivalente en Simulink.

Esta función de transferencia será la empleada como modelo SISO para la sintonización inicial del controlador PID. Posteriormente, el controlador obtenido se integra en el modelo completo en espacio de estados para comprobar su comportamiento conjunto.

8.3. Ley de control PID

El controlador proporcional-integral-derivativo, o controlador PID, es una de las estructuras de control clásico más utilizadas debido a su sencillez, robustez y facilidad de implementación. Su funcionamiento se basa en generar una acción de control a partir del error entre la referencia deseada y la salida medida del sistema. Esta acción se obtiene combinando tres términos: una acción proporcional, una acción integral y una acción derivativa [26].

De forma general, la ley de control PID en tiempo continuo puede expresarse como:

$$u(t) = K_p e(t) + K_i \int_0^t e(\lambda) d\lambda + K_d \frac{de(t)}{dt} \tag{8.4}$$

donde $e(t)$ representa el error de control, $u(t)$ la señal generada por el controlador, y K_p, K_i y K_d son las ganancias proporcional, integral y derivativa, respectivamente.

La **acción proporcional** genera una señal de control directamente proporcional al error instantáneo. Su principal efecto es aumentar la rapidez de respuesta del sistema, ya que cuanto mayor sea la desviación respecto a la referencia, mayor será la acción correctora aplicada. No obstante, un valor excesivo de K_p puede producir oscilaciones o incluso comprometer la estabilidad del sistema.

La **acción integral** actúa sobre el error acumulado en el tiempo. Su finalidad principal es reducir o eliminar el error en régimen permanente, especialmente ante perturbaciones constantes o pequeños desajustes del modelo. Sin embargo, una acción integral demasiado elevada puede ralentizar la respuesta transitoria y aumentar la sobreoscilación, por lo que su ajuste debe realizarse cuidadosamente.

La **acción derivativa** depende de la variación temporal del error. Esta componente introduce un efecto anticipativo, ya que permite reaccionar no solo al valor actual del error, sino también a su tendencia de evolución. En consecuencia, puede mejorar el amortiguamiento de la respuesta y reducir las oscilaciones alrededor del punto de equilibrio.

En el sistema considerado, el controlador PID se aplica de forma independiente a cada eje, empleando como entradas los errores de posición $e_x(t)$ y $e_y(t)$. Así, las acciones de control generadas pueden escribirse como:

$$u_x(t) = K_p e_x(t) + K_i \int_0^t e_x(\lambda) d\lambda + K_d \frac{de_x(t)}{dt}$$
$$u_y(t) = K_p e_y(t) + K_i \int_0^t e_y(\lambda) d\lambda + K_d \frac{de_y(t)}{dt}$$

donde se ha considerado el mismo conjunto de ganancias para ambos ejes, de acuerdo con la hipótesis de simetría de la plataforma planteada en el apartado anterior. Estas señales de control se interpretan posteriormente como las referencias de inclinación necesarias

para corregir la posición de la bola sobre el plato, teniendo en cuenta la correspondencia de signos y ejes definida en la implementación del modelo.

Un aspecto importante en la implementación práctica del PID es el tratamiento de la acción derivativa. En este tipo de controlador, dicha acción requiere calcular la variación temporal del error de posición. Si la medida procede de un sensor visual ruidoso, una derivación directa puede amplificar el ruido y degradar la acción de control; por ello, la derivada suele implementarse de forma filtrada. En cambio, en un controlador por **realimentación de estados**, las velocidades $\dot{x}$ e $\dot{y}$ forman parte explícitamente del vector de estado requerido, por lo que resulta más natural emplear un estimador como el **filtro de Kalman** cuando dichas variables no están disponibles directamente.

Por este motivo, en este capítulo el filtro de Kalman no se emplea como elemento principal del controlador PID. Su uso queda reservado para estrategias de control basadas en espacio de estados, donde la estimación de variables no medidas directamente resulta necesaria para reconstruir el vector de estado completo.

8.4. Sintonización de parámetros

La sintonización del controlador PID consiste en determinar los valores de las ganancias K_p, K_i y K_d que permiten obtener una respuesta estable y adecuada del sistema en lazo cerrado. En este trabajo, la sintonización se realiza a partir de la función de transferencia equivalente obtenida para uno de los ejes del sistema, aprovechando la simetría geométrica de la plataforma y la equivalencia dinámica asumida entre los ejes X e Y.

Aunque existen distintos métodos clásicos para la sintonización de controladores PID (véase la sección 2.6.1), en este trabajo se adopta una **metodología asistida mediante MATLAB**. Este enfoque permite obtener una primera estimación de los parámetros del controlador a partir del modelo SISO desacoplado y realizar posteriormente un ajuste iterativo en simulación atendiendo a la respuesta temporal y frecuencial del sistema.

8.4.1. Metodología de sintonización empleada

Como se ha indicado, para el diseño inicial del controlador se emplea la función de transferencia equivalente correspondiente a uno de los ejes de la planta, definida en la ecuación 8.3. A partir de esta, se obtiene un controlador PID en el dominio de Laplace.

En su forma ideal, el controlador PID puede expresarse como:

$$C(s) = K_p + \frac{K_i}{s} + K_d s \tag{8.5}$$

donde K_p, K_i y K_d representan, respectivamente, las ganancias proporcional, integral y derivativa. La acción derivativa mejora el amortiguamiento de la respuesta, pero su implementación ideal puede amplificar componentes de alta frecuencia presentes en la señal de medida. Por este motivo, en la práctica suele sustituirse por una acción derivativa filtrada, cuya estructura puede interpretarse como una red de adelanto de fase al introducir un cero y un polo real en la función de transferencia del controlador [34].

Así pues, en este trabajo se emplea una acción derivativa filtrada, de forma que el controlador implementado adopta la expresión:

$$C(s) = K_p + \frac{K_i}{s} + K_d \frac{Ns}{s + N} \tag{8.6}$$

donde N es el coeficiente de filtrado derivativo. Este término permite limitar la ganancia de la acción derivativa a altas frecuencias, reduciendo así la sensibilidad del controlador frente al ruido de medida, especialmente cuando se emplea el sensor visual simulado.

Para obtener una primera aproximación de las ganancias se emplea la herramienta de diseño clásico disponible en MATLAB conocida como **Control System Designer**. Esta herramienta permite ajustar el controlador a partir de la función de transferencia equivalente y analizar la respuesta en lazo cerrado, facilitando la selección inicial de los parámetros del PID.

Una vez obtenida esta primera sintonización, los parámetros se refinan mediante simulaciones en Simulink sobre el modelo completo en espacio de estados. De este modo, aunque el diseño inicial se realiza sobre una aproximación SISO desacoplada, la validación se

lleva a cabo sobre la planta completa, teniendo en cuenta la dinámica conjunta del sistema.

8.4.2. Criterios de ajuste

El ajuste final de los parámetros del controlador PID se realiza atendiendo a criterios clásicos de respuesta temporal, habitualmente empleados en el diseño de sistemas de control. Entre estos criterios destacan el tiempo de establecimiento, la sobreoscilación máxima y el error en régimen permanente, que permiten evaluar la rapidez, el amortiguamiento y la precisión de la respuesta del sistema [24], [26], [34].

En la figura 8.5 se muestran de forma esquemática algunos de los parámetros característicos de la respuesta temporal de un sistema subamortiguado. Aunque estas magnitudes se definen habitualmente a partir de la respuesta al escalón de un sistema de segundo orden, en este trabajo se emplean como referencia práctica para evaluar el comportamiento del sistema bola-plataforma en lazo cerrado.

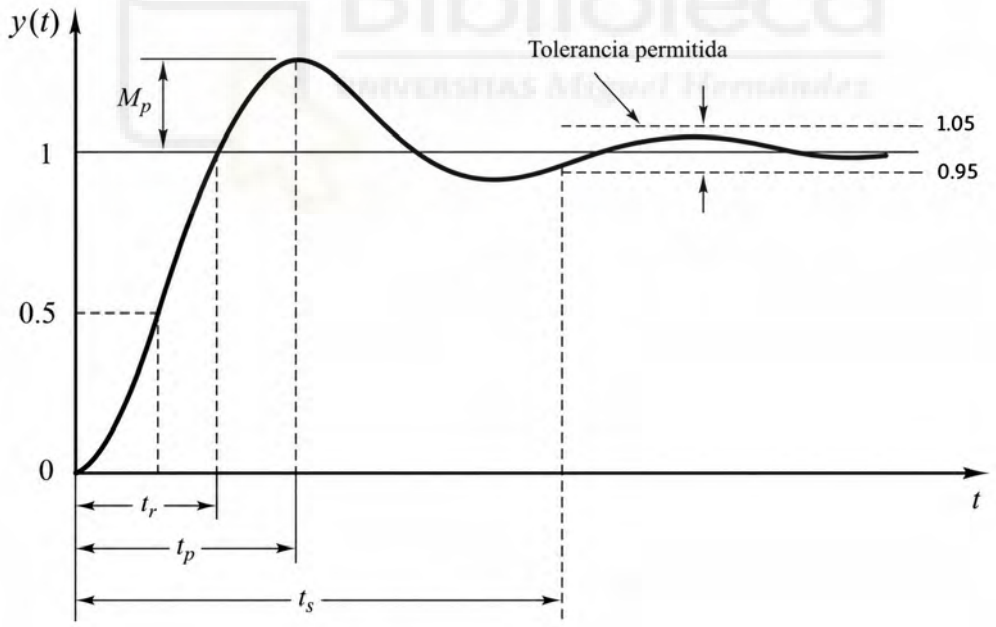


Figura 8.5: Parámetros característicos de la respuesta temporal de un sistema subamortiguado.

En el caso considerado, el objetivo principal no es el seguimiento de una referencia variable, sino la estabilización de la bola en el centro de la plataforma. Por tanto, durante las simulaciones la referencia de posición se mantiene constante en el origen del plato, mientras que la bola parte de una condición inicial desplazada respecto al centro. De este

modo, el análisis se centra en la capacidad del controlador para devolver la bola al punto de equilibrio.

En este contexto, el **tiempo de establecimiento** se define como el tiempo necesario para que la posición de la bola entre y permanezca dentro de una banda de tolerancia alrededor del centro de la plataforma. En este trabajo se considera una banda de tolerancia de:

$$\varepsilon = 0{,}005 \text{ m}$$

por lo que el sistema se considera establecido cuando se cumple:

$$\begin{aligned} |x(t)| &\leq \varepsilon \\ |y(t)| &\leq \varepsilon \end{aligned} \tag{8.7}$$

y ambas coordenadas permanecen dentro de dicha banda hasta el final de la simulación.

La **sobreoscilación máxima** se interpreta como el desplazamiento de la bola al cruzar el punto de referencia y alejarse hacia el lado opuesto del plato. Este criterio permite evaluar el grado de amortiguamiento de la respuesta y detectar comportamientos excesivamente oscilatorios. Una sobreoscilación elevada puede indicar un ajuste demasiado agresivo del controlador o una acción derivativa insuficiente.

El **error estacionario** se evalúa a partir de la posición final de la bola respecto al centro de la plataforma. Dado que la referencia de posición es nula, el objetivo es que:

$$x_{ss} \approx 0$$
$$y_{ss} \approx 0$$

o, de forma equivalente, que los errores de posición tiendan a cero en régimen permanente.

Además de la evolución de la posición de la bola, durante la sintonización también se analizan las referencias de inclinación generadas por el controlador: $\phi_{ref}(t), \theta_{ref}(t)$. Estas señales deben mantenerse dentro de un rango compatible con la hipótesis de pequeñas inclinaciones empleada en la linealización del modelo. Si el controlador genera referencias demasiado elevadas o cambios bruscos en la orientación de la plataforma, la respuesta obtenida puede dejar de ser representativa del comportamiento previsto por el modelo

linealizado.

Por tanto, el ajuste del controlador no se realiza únicamente buscando minimizar el error de posición, sino también garantizando una respuesta suficientemente amortiguada, sin oscilaciones sostenidas y con referencias de inclinación físicamente razonables para la plataforma.

8.4.3. Parámetros finales del controlador

Durante la sintonización se comprobó que la incorporación de acción integral no resultaba conveniente para la planta equivalente obtenida. La función de transferencia de la ecuación 8.3 presenta dos polos en el origen, asociados al carácter integrador de la dinámica entre la inclinación de la plataforma y la posición de la bola. En la figura 8.6 se muestra la situación de los polos y los ceros de la planta equivalente mediante el **lugar de las raíces**.

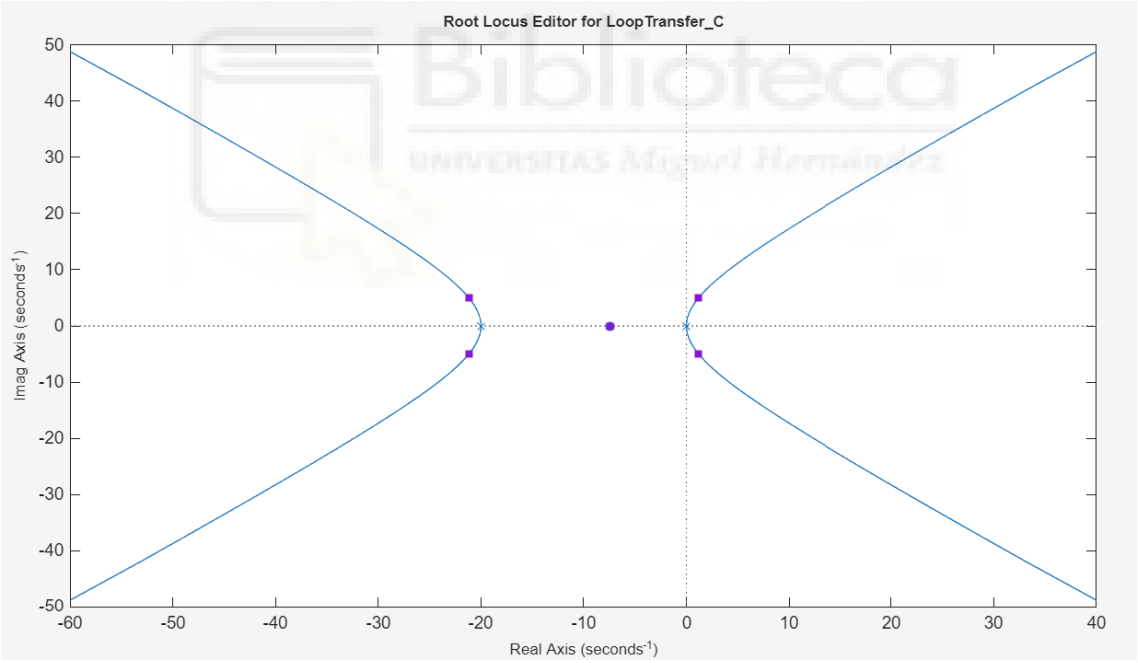


Figura 8.6: Representación del lugar de las raíces de la planta equivalente.

Al añadir un término integral externo, el controlador introduce un nuevo polo en el origen, lo que desplaza desfavorablemente el lugar de las raíces y compromete la estabilidad del sistema. Por este motivo, se opta por emplear un controlador PD con acción derivativa filtrada, equivalente a un controlador PID con $K_i = 0$.

Como se muestra en la figura 8.7, la herramienta **Control System Designer** facilita el

diseño del controlador mediante gráficas interactivas de la respuesta temporal, el lugar de las raíces y los diagramas de Bode. Esta herramienta se emplea como apoyo para ajustar los parámetros del controlador y analizar la estabilidad del sistema en lazo cerrado.

Figura 8.7: Configuración del controlador mediante Control System Designer.

Tras el proceso de sintonización y ajuste iterativo, se obtiene el conjunto final de parámetros empleados en las simulaciones posteriores. Estos valores se recogen en la tabla 8.1.

Parámetro	**Valor**
K_p	0.32
K_i	0
K_d	0.384
N	20
Saturación angular máxima	$\pm 10°$

Tabla 8.1: Parámetros del controlador PD con derivada filtrada.

Sustituyendo dichos valores en la ecuación 8.6, el controlador implementado queda definido como:

$$C(s) = 0{,}32 + 0{,}384\frac{20s}{s+20} \tag{8.8}$$

El controlador resultante se emplea inicialmente en las simulaciones con sensor ideal, con el objetivo de comprobar su capacidad para estabilizar la bola sin la influencia de no idealidades de medida. Posteriormente, se mantiene la misma sintonización y se valida el comportamiento del sistema empleando el sensor visual simulado. De esta forma, puede

analizarse la degradación de la respuesta debida al ruido, al retardo, a la cuantización y a la frecuencia de muestreo limitada, sin modificar los parámetros del controlador.

8.5. Simulaciones con sensor ideal

Una vez definidos los parámetros del controlador, se realiza una primera validación en simulación empleando un sensor ideal de posición. En esta configuración, las variables medidas coinciden directamente con las variables reales de la planta. Es decir, el lazo de control se cierra mediante realimentación negativa directa de la posición de la bola.

La figura 8.8 muestra el esquema implementado en Simulink para la validación inicial del controlador con sensor ideal. En él, las referencias de posición se fijan en el centro de la plataforma, mientras que las posiciones $x(t)$ e $y(t)$ obtenidas a la salida del modelo en espacio de estados se realimentan directamente para calcular los errores de posición. Cada error se introduce en un controlador PD con derivada filtrada, cuyas salidas corresponden a las referencias de inclinación ϕ_{ref} y θ_{ref}.

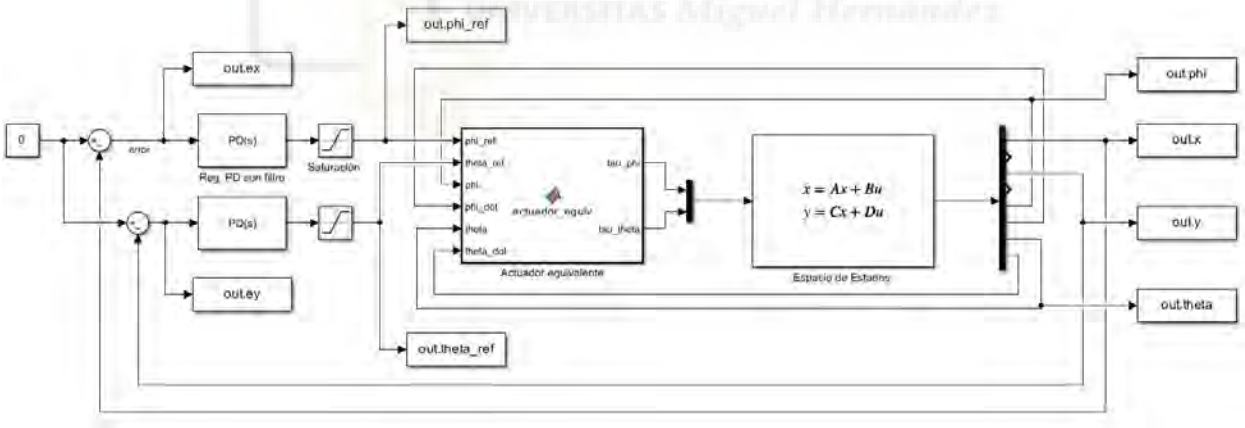


Figura 8.8: Esquema de simulación del controlador PD con sensor ideal.

Con el objetivo de evitar inclinaciones excesivas de la plataforma y mantener la validez de la aproximación lineal empleada en el modelo, las referencias de orientación generadas por el controlador se limitan mediante bloques de **saturación**, en concordancia con los valores establecidos en la tabla 8.1.

Las referencias saturadas se introducen posteriormente en el bloque de actuación equivalente, que genera los pares equivalentes τ_ϕ y τ_θ. Dichos pares constituyen las entradas

del modelo en espacio de estados, cuya salida proporciona la evolución temporal de las variables del sistema.

8.5.1. Escenarios de prueba

Para evaluar la capacidad del controlador de estabilizar la bola en el centro de la plataforma, se definen diferentes condiciones iniciales de simulación. En todos los casos, la referencia de posición se mantiene constante en el origen del plato: $x_{ref}, y_{ref} = 0$.

Por tanto, los ensayos realizados no representan un seguimiento de referencia, sino la recuperación del equilibrio a partir de distintas posiciones y velocidades iniciales de la bola. Las condiciones iniciales del modelo en espacio de estados se definen como:

$$\mathbf{x}(0) = [x_0, \dot{x}_0, y_0, \dot{y}_0, \phi_0, \dot{\phi}_0, \theta_0, \dot{\theta}_0]^T \tag{8.9}$$

En los ensayos considerados, la plataforma parte inicialmente horizontal y en reposo, por lo que:

$$\phi_0 = 0, \qquad \dot{\phi}_0 = 0$$
$$\theta_0 = 0, \qquad \dot{\theta}_0 = 0$$

De esta forma, únicamente se modifican la posición y la velocidad inicial de la bola. La tabla 8.2 recoge las condiciones iniciales empleadas en las simulaciones.

Ensayo	x_0 **[m]**	$\dot{x}_0$ **[m/s]**	y_0 **[m]**	$\dot{y}_0$ **[m/s]**	**Descripción**
1	0.03	0	0	0	Desplazamiento inicial en el eje X
2	0	0	0.03	0	Desplazamiento inicial en el eje Y
3	0.03	0	-0.03	0	Desplazamiento inicial diagonal
4	0	0.1	0	0	Velocidad inicial en el eje X
5	0.02	0.08	-0.02	-0.05	Posición y velocidad inicial diagonal
6	0.08	0.15	-0.06	0.10	Condición desfavorable cercana al borde

Tabla 8.2: Condiciones iniciales empleadas en los ensayos de simulación.

Los tres primeros ensayos evalúan la recuperación del equilibrio sin velocidad inicial. Los ensayos 4 y 5 introducen velocidades iniciales no nulas, y el ensayo 6 representa una condición crítica cerca del borde. Dado que el radio de la plataforma es de 0.12 m, en el sexto ensayo se selecciona una posición inicial situada a una distancia de 0.10 m del

centro:

$$r_0 = \sqrt{x_0^2 + y_0^2} = \sqrt{0{,}08^2 + (-0{,}06)^2} = 0{,}10\,\text{m}$$

En **Simulink**, estas condiciones se introducen en el bloque de espacio de estados. Por ejemplo, el vector de condiciones iniciales para el ensayo 6 es: `[0.08; 0.15; -0.06; 0.10; 0; 0; 0; 0]`.

8.5.2. Variables analizadas y criterios de evaluación

Durante las simulaciones con sensor ideal se registran las principales variables del sistema con el objetivo de evaluar la respuesta del controlador en cada uno de los escenarios definidos en el apartado anterior. En particular, se analizan:

- Las posiciones de la bola: $x(t), y(t)$.
- Los errores de posición a la entrada del controlador: $e_x(t), e_y(t)$.
- Las referencias de inclinación generadas por el controlador tras la saturación: $\phi_{ref}(t), \theta_{ref}(t)$.
- Las inclinaciones reales de la plataforma obtenidas a la salida del modelo en espacio de estados: $\phi(t), \theta(t)$.

El análisis de la posición de la bola permite comprobar si el controlador es capaz de devolver el sistema al punto de equilibrio desde las distintas condiciones iniciales consideradas. Dado que la referencia se mantiene fija en el centro del plato, el objetivo es que ambas coordenadas $x(t)$ e $y(t)$ converjan hacia cero.

Para cuantificar este comportamiento, se emplea la banda de tolerancia definida en el apartado 8.4.2. El sistema se considera establecido cuando ambas coordenadas entran y permanecen dentro de dicha banda hasta el final de la simulación.

Además de la posición de la bola, se estudian las referencias de inclinación proporcionadas por el controlador. Estas señales permiten evaluar el esfuerzo de control exigido a la plataforma y comprobar si se alcanzan los límites de saturación impuestos. Valores

cercanos a estos límites indican que el controlador está solicitando la máxima inclinación permitida para corregir la desviación de la bola. Aunque una saturación puntual puede ser admisible, una saturación mantenida en el tiempo podría indicar que la condición inicial es demasiado exigente o que el controlador requiere una acción excesiva.

También se analizan las inclinaciones reales de la plataforma, $\phi(t)$ y $\theta(t)$, con el fin de verificar el seguimiento de las referencias generadas por el controlador y comprobar que la orientación del plato se mantiene dentro de un rango compatible con la hipótesis de pequeñas inclinaciones empleada en el modelado linealizado.

Finalmente, para resumir de forma cuantitativa los resultados de cada ensayo, se calculan los indicadores que posteriormente se recogen en la tabla de resultados: el tiempo de establecimiento, los valores máximos de posición, las inclinaciones máximas de referencia y reales, y los valores finales de posición. Estos parámetros permiten comparar de forma homogénea los distintos escenarios de prueba y valorar la capacidad del controlador para estabilizar la bola en condiciones iniciales cada vez más exigentes.

8.5.3. Análisis de resultados con sensor ideal

Una vez definidos los escenarios de prueba, se realizan las simulaciones empleando el esquema con sensor ideal de posición. Las simulaciones se llevan a cabo con un tiempo total de **cinco segundos**, suficiente para observar tanto el régimen transitorio como la permanencia de la bola dentro de la banda de tolerancia indicada en el apartado anterior.

Con el fin de resumir los resultados obtenidos en los distintos escenarios se proponen las tablas 8.3 y 8.4. La primera recoge los indicadores asociados a la posición de la bola, mientras que la segunda resume las referencias de inclinación generadas por el controlador y las orientaciones máximas alcanzadas por la plataforma.

Tabla 8.3: Resultados de posición obtenidos con sensor ideal.

Ensayo	t_s [s]	**máx** $\lvert x \rvert$ [m]	**máx** $\lvert y \rvert$ [m]	x_f [m]	y_f [m]
1	1.435	0.0300	0.0000	-0.00005	0.00000
2	1.435	0.0000	0.0300	0.00000	-0.00005
3	1.435	0.0300	0.0300	-0.00005	-0.00005
4	2.256	0.0325	0.0000	0.00015	0.00000
5	1.779	0.0319	0.0256	0.00008	-0.00004
6	2.682	0.1021	0.0600	0.00026	0.00019

Tabla 8.4: Referencias de inclinación y orientación máxima de la plataforma con sensor ideal.

Ensayo	**máx** $\lvert \phi_{ref} \rvert$ [°]	**máx** $\lvert \theta_{ref} \rvert$ [°]	**máx** $\lvert \phi \rvert$ [°]	**máx** $\lvert \theta \rvert$ [°]
1	10.00	0.00	3.71	0.00
2	0.00	10.00	0.00	3.71
3	10.00	10.00	3.71	3.71
4	2.28	0.00	2.10	0.00
5	9.17	9.17	3.32	3.02
6	10.00	10.00	7.34	4.90

El tiempo de establecimiento t_s se calcula como el instante a partir del cual ambas coordenadas de la bola permanecen dentro de la banda de tolerancia hasta el final de la simulación. Los valores máx $|x(t)|$ y máx $|y(t)|$ permiten comprobar el desplazamiento máximo alcanzado por la bola durante la respuesta, especialmente en los ensayos con velocidad inicial. Por su parte, las inclinaciones máximas de referencia y reales permiten evaluar el esfuerzo exigido al controlador y verificar si la plataforma trabaja dentro del rango admisible de pequeñas inclinaciones.

Los valores x_f e y_f corresponden a la posición final de la bola al terminar la simulación. Estos valores permiten evaluar el error estacionario alcanzado en cada ensayo. Dado que la referencia se fija en el centro de la plataforma, un controlador adecuado debe proporcionar valores finales próximos a cero:

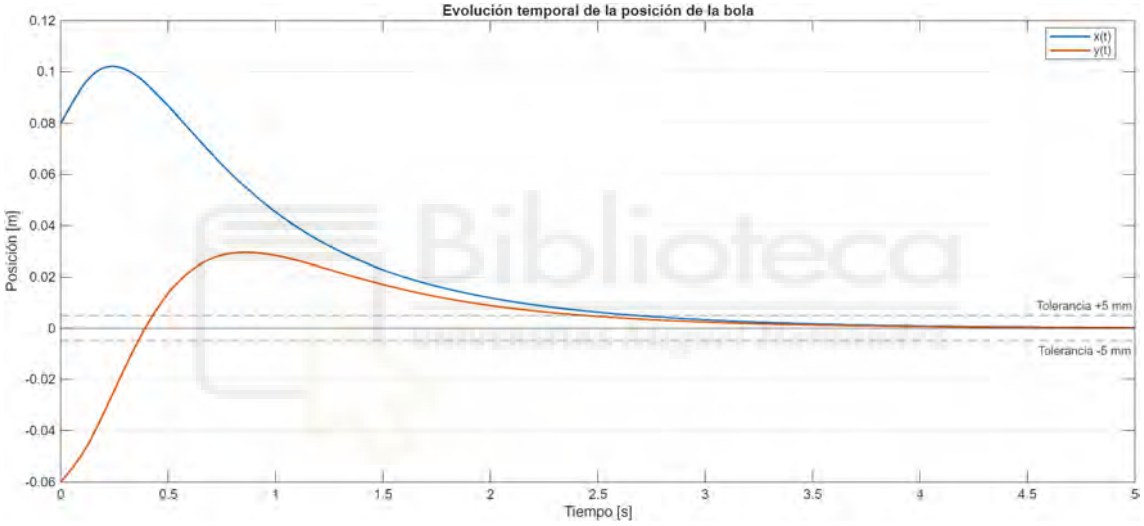
$$x_f \approx 0$$

$$y_f \approx 0$$

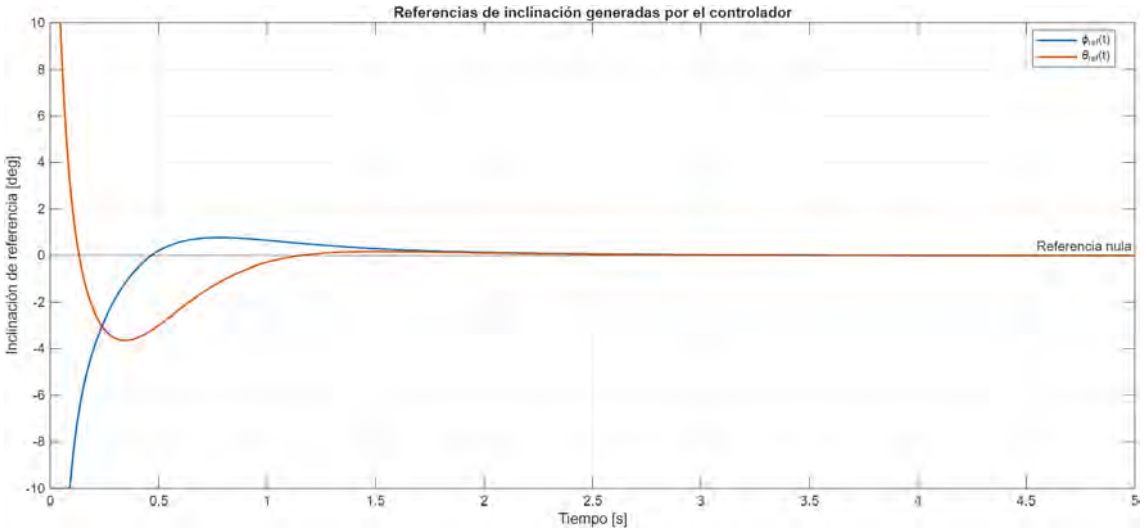
A partir de los resultados obtenidos, se comprueba que el controlador diseñado es capaz de estabilizar la bola en el centro de la plataforma para todas las condiciones iniciales consideradas. En todos los ensayos, las posiciones $x(t)$ e $y(t)$ convergen hacia el origen y

alcanzan la banda de establecimiento antes de los 3 s. Este comportamiento confirma que la sintonización del controlador PD con derivada filtrada resulta adecuada para el modelo en espacio de estados cuando se emplea realimentación ideal de posición.

La figura 8.9 muestra la respuesta temporal obtenida para el **ensayo 6**, correspondiente a la condición inicial más desfavorable considerada. En este caso, la bola parte desde una posición cercana al borde del plato y con velocidad inicial distinta de cero. Como se observa en la figura, la coordenada $x(t)$ alcanza inicialmente un valor máximo próximo a 0,102 metros, mientras que la coordenada $y(t)$ cambia de signo durante el transitorio. No obstante, ambas posiciones convergen hacia el origen y entran dentro de la banda de tolerancia de ±5 mm antes de los 3 segundos.



(a) Posición de la bola.



(b) Referencias de inclinación generadas por el controlador.

Figura 8.9: Respuesta temporal del sistema con sensor ideal para el ensayo 6.

En cuanto a las referencias de inclinación, se aprecia que el controlador alcanza inicialmente los límites de saturación angular de $\pm 10°$, debido a la severidad de la condición inicial. Sin embargo, esta saturación se produce únicamente durante el transitorio inicial, y las referencias $\phi_{ref}(t)$ y $\theta_{ref}(t)$ tienden posteriormente a cero sin presentar oscilaciones sostenidas.

8.6. Validación con sensor visual simulado

Una vez validado el controlador empleando realimentación ideal de posición, se analiza su comportamiento al sustituir el sensor ideal por el sensor visual simulado desarrollado en el capítulo 7.4. Esta segunda validación permite estudiar la influencia de las **no idealidades** asociadas al proceso de medida sobre el comportamiento del lazo cerrado.

En este caso, el controlador no recibe directamente las posiciones reales $x(t)$ e $y(t)$ de la bola, sino las posiciones $x_m(t)$, $y_m(t)$ medidas por el sensor. Estas señales incorporan los efectos de muestreo, ruido de medida, retardo temporal y cuantización definidos previamente; y, a partir de ellas, se calculan los errores de entrada al controlador señalados en la ec. 8.1.

La figura 8.10 muestra el esquema de Simulink empleado para esta validación. Como puede observarse, la estructura de control es equivalente a la utilizada con sensor ideal, pero incorporando el bloque de sensor visual simulado en la rama de realimentación. De este modo, la planta proporciona las posiciones reales de la bola, mientras que el controlador trabaja únicamente con las señales medidas.

Los parámetros del sensor visual simulado son los mismos que los definidos en la tabla 7.1 y se mantienen los mismos escenarios de prueba y las mismas condiciones iniciales que en el apartado anterior. Asimismo, no se modifica la sintonización del controlador. De esta forma, las diferencias observadas en la respuesta pueden atribuirse al efecto del sistema de medida no ideal.

Durante las simulaciones se analizan las mismas variables que en el caso con sensor ideal. Para mejorar la legibilidad, los resultados se dividen en dos tablas. La tabla 8.5 recoge los indicadores asociados a la posición de la bola, mientras que la tabla 8.6 resume

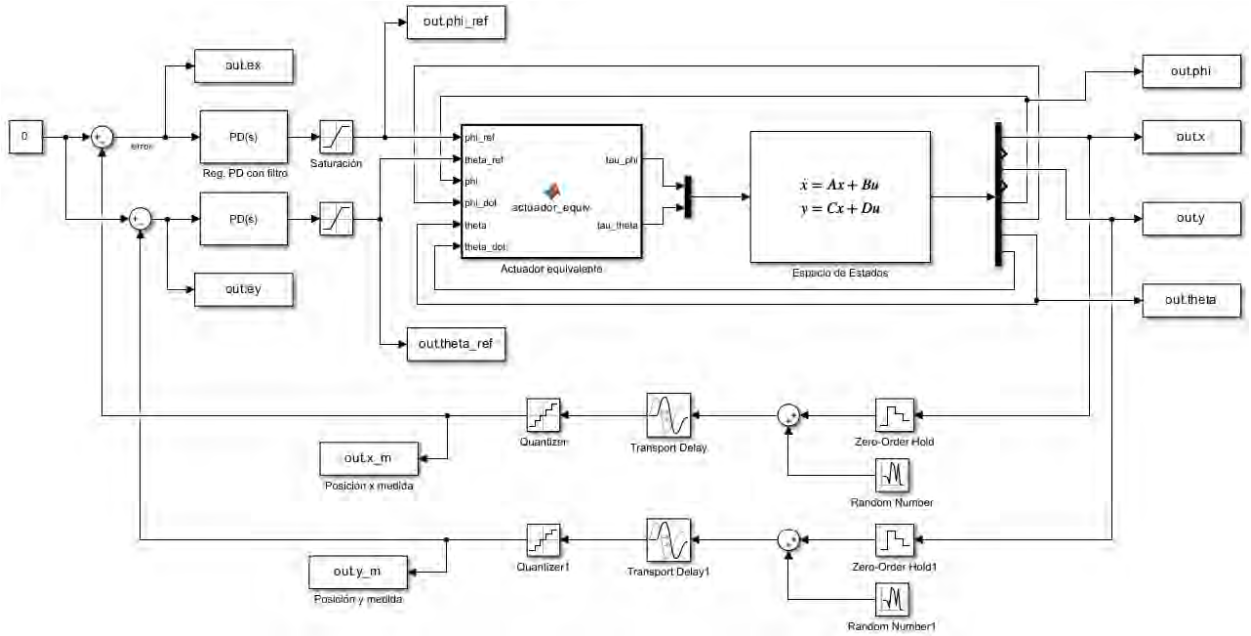


Figura 8.10: Esquema de simulación del controlador PD con sensor visual simulado.

las referencias de inclinación generadas por el controlador y las orientaciones máximas alcanzadas por la plataforma.

Tabla 8.5: Resultados de posición obtenidos con sensor visual simulado.

Ensayo	t_s [s]	**máx** $\lvert x \rvert$ [m]	**máx** $\lvert y \rvert$ [m]	x_f [m]	y_f [m]
1	1.430	0.0300	0.0003	-0.00018	0.00005
2	1.461	0.0002	0.0300	-0.00013	0.00005
3	1.434	0.0300	0.0300	-0.00019	0.00012
4	2.186	0.0345	0.0003	0.00007	0.00005
5	1.689	0.0335	0.0266	0.00009	0.00007
6	2.652	0.1051	0.0600	0.00013	0.00030

Tabla 8.6: Referencias de inclinación y orientación máxima de la plataforma con sensor visual simulado.

Ensayo	**máx** $\lvert \phi_{ref} \rvert$ [°]	**máx** $\lvert \theta_{ref} \rvert$ [°]	**máx** $\lvert \phi \rvert$ [°]	**máx** $\lvert \theta \rvert$ [°]
1	10.00	0.73	3.74	0.09
2	0.81	10.00	0.13	3.72
3	10.00	10.00	3.74	3.76
4	2.83	0.73	2.15	0.09
5	9.40	9.17	3.28	2.99
6	10.00	10.00	7.37	4.95

Los resultados obtenidos muestran que el controlador mantiene la capacidad de estabilizar la bola aun cuando la realimentación procede del sensor visual simulado. En todos los

ensayos, las posiciones reales $x(t)$ e $y(t)$ entran en la banda de tolerancia antes de los 3 segundos. La principal diferencia respecto al sensor ideal se observa en las referencias de inclinación, que presentan un mayor rizado debido a las no idealidades de medida.

En una implementación física, este rizado debería tenerse en cuenta, ya que podría generar pequeñas correcciones continuas en los actuadores. Para reducir este efecto podrían incorporarse estrategias adicionales, como un filtrado paso bajo de la medida, una reducción del coeficiente de filtrado derivativo N, una zona muerta alrededor del centro o una limitación de la velocidad de cambio de las referencias angulares.

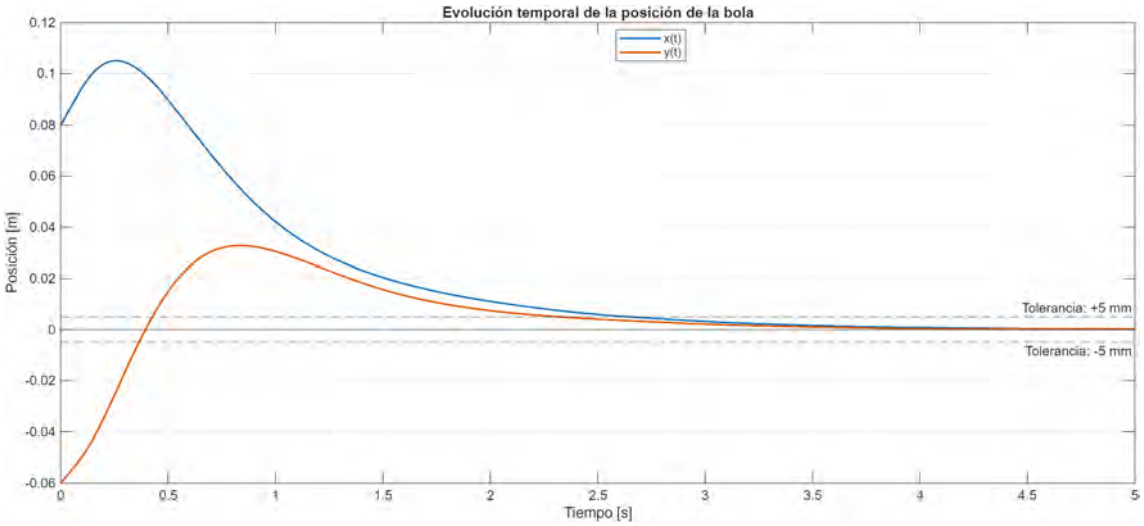
En los ensayos 1, 2 y 4 aparece una pequeña actuación en el eje no excitado inicialmente. Este efecto se debe a que el controlador trabaja con señales medidas afectadas por ruido, cuantización y muestreo, generando pequeñas correcciones aunque la posición real en dicho eje sea prácticamente nula. No obstante, la amplitud de estas variaciones es reducida y no compromete la estabilidad del sistema.

El ensayo 6 vuelve a constituir el caso más exigente. En este escenario, la bola alcanza un desplazamiento máximo de 0,1051 m en el eje X, permaneciendo todavía dentro del radio de la plataforma. Aunque las referencias de inclinación alcanzan inicialmente los límites de saturación, la bola converge finalmente hacia el centro, con un tiempo de establecimiento de 2,652 s. La figura 8.11 muestra la respuesta temporal correspondiente a este ensayo.

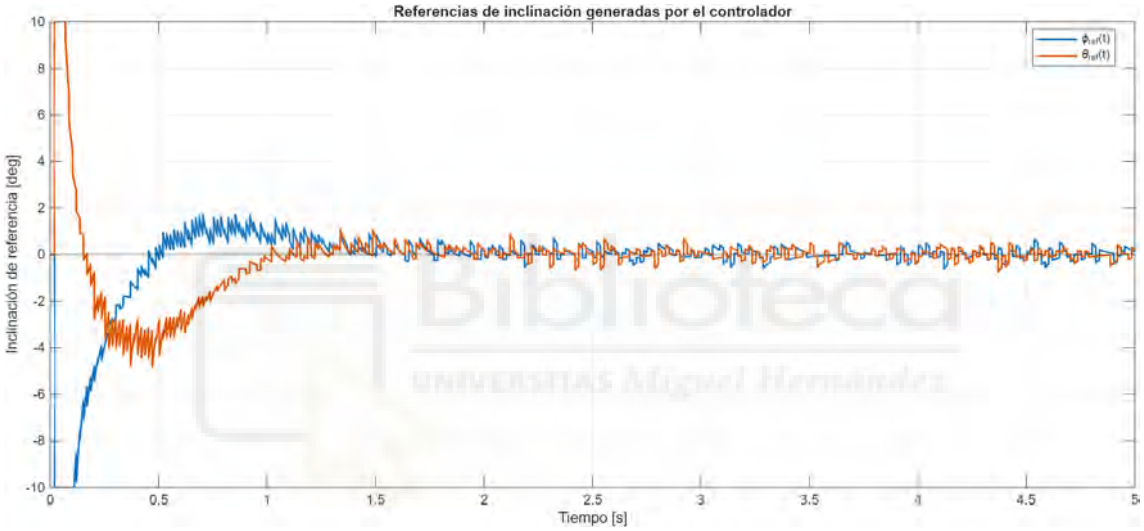
Por otra parte, la comparación entre la posición real de la bola $x(t)$ y la medida $x_m(t)$ permite apreciar el efecto combinado del muestreo, retardo, ruido y cuantización. Para visualizar este efecto se propone la figura 8.12.

A pesar de estas no idealidades, la señal medida conserva la información suficiente para que el controlador estabilice la bola en el centro de la plataforma.

En conjunto, los resultados muestran que el controlador PD con derivada filtrada mantiene un comportamiento satisfactorio al trabajar con un sensor más realista. No obstante, en comparación con el sensor ideal, se observa un aumento del rizado en las referencias de inclinación y una ligera degradación de la suavidad de la respuesta. Estos efectos justifican la importancia de considerar las no idealidades del sistema de medida durante la validación del controlador.



(a) Posición de la bola.



(b) Referencias de inclinación generadas por el controlador.

Figura 8.11: Respuesta temporal del sistema con sensor visual simulado para el ensayo 6.

Adicionalmente, se realizaron ensayos complementarios de rechazo frente a **perturbaciones externas** mediante la introducción de señales tipo pulso sobre la posición de la bola. Para ello, se empleó el bloque *Pulse Generator* de Simulink, aplicado de forma aditiva sobre la coordenada perturbada. Esta configuración permite representar una desviación brusca y de duración limitada de la bola respecto a su posición nominal, evitando introducir un desplazamiento permanente en la señal de realimentación.

Durante estos ensayos se modificaron distintos parámetros de la perturbación, como su amplitud, duración e instante de aplicación, con el fin de analizar de forma cualitativa la capacidad del controlador para recuperar la posición de equilibrio tras una desviación externa. Los resultados obtenidos mostraron que el sistema era capaz de compensar dichas

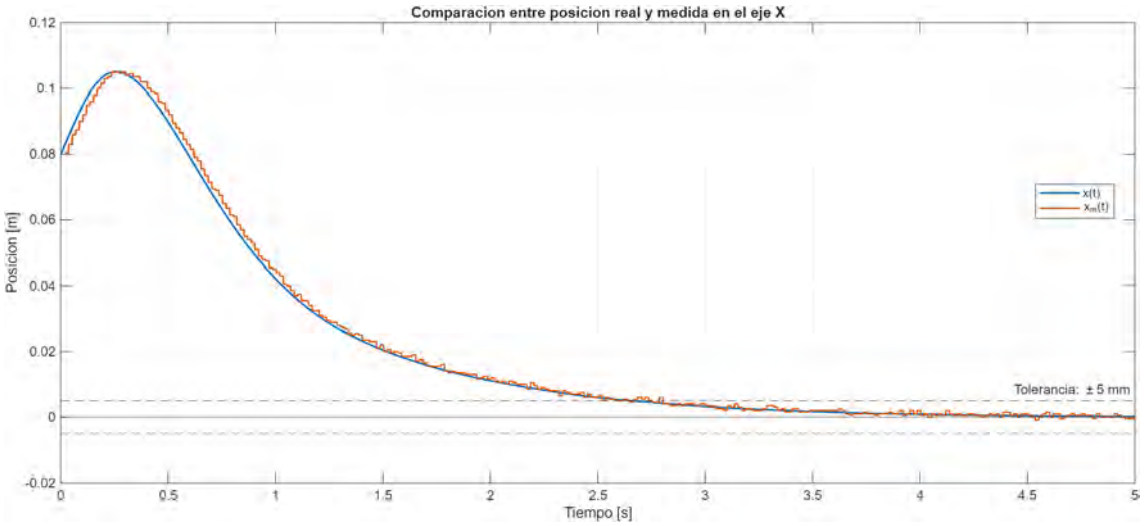


Figura 8.12: Comparación entre la posición real $x(t)$ y la posición medida $x_m(t)$ para el ensayo 6.

perturbaciones y reconducir la bola hacia la referencia una vez desaparecía la acción externa.

8.7. Conclusiones del capítulo

En este capítulo se ha diseñado e implementado una primera estrategia de control clásico para la estabilización de la bola sobre la plataforma. Partiendo del modelo linealizado en espacio de estados, se ha obtenido una función de transferencia equivalente que incluye tanto la dinámica de la planta como el bloque de actuación equivalente, permitiendo realizar la sintonización del controlador sobre un modelo SISO representativo de cada eje.

Durante el proceso de ajuste se comprobó que la incorporación de acción integral no resultaba conveniente, debido a la presencia de polos en el origen en la planta equivalente. Por este motivo, se seleccionó finalmente un **controlador PD con acción derivativa filtrada**.

Se han realizado simulaciones tanto con un sensor ideal como con el sensor visual simulado desarrollado en el capítulo 7 y, en conjunto, los resultados confirman que el controlador implementado constituye una solución sencilla y eficaz para estabilizar la bola en torno al punto de equilibrio. Sin embargo, también se evidencian algunas limitaciones propias del control clásico descentralizado, especialmente relacionadas con la sensibilidad frente

a las no idealidades de medida y con el tratamiento aproximado del sistema como dos ejes desacoplados.

Estas limitaciones motivan el estudio de estrategias de control más avanzadas. En particular, el siguiente capítulo aborda el diseño de un controlador por **realimentación de estados**, capaz de aprovechar de forma explícita la representación multivariable del sistema y de incorporar estimación de estados cuando algunas variables, como las velocidades de la bola, no se encuentran disponibles directamente mediante el sistema de medida.

9. CONTROL POR REALIMENTACIÓN DE ESTADOS

9.1. Introducción

En el capítulo anterior se diseñó un controlador PD con derivada filtrada para estabilizar la bola en el centro de la plataforma. Dicha estrategia se basaba en una estructura de control descentralizada, empleando un regulador independiente para cada eje de movimiento. De este modo, el problema multivariable original se aproximaba mediante dos lazos de control desacoplados, de tipo Entrada Única-Salida Única (SISO), uno asociado al desplazamiento de la bola en el eje X y otro asociado al desplazamiento en el eje Y.

Esta aproximación resulta adecuada bajo las hipótesis adoptadas en el modelado lineal del sistema, especialmente para pequeñas inclinaciones de la plataforma y operación en torno al punto de equilibrio. Sin embargo, el sistema real presenta una naturaleza inherentemente **multivariable**, ya que el movimiento de la bola, la orientación del plato y la actuación de la plataforma forman parte de una dinámica conjunta. Además, aunque el modelo empleado para el diseño se haya linealizado y simplificado, el sistema físico original es no lineal y presenta **acoplamientos** derivados tanto de la dinámica de la bola como de la arquitectura paralela del mecanismo 3-RRS.

El presente capítulo introduce una transición hacia las técnicas de control moderno mediante la teoría del **espacio de estados**. A diferencia del esquema clásico descentralizado, la realimentación de estados trata al sistema de una manera integrada y multivariable (MIMO), explotando de forma directa las matrices dinámicas A y B desarrolladas en el Capítulo 4. De esta forma, la acción de control se calcula a partir del vector de estados completo, y no únicamente a partir del error de posición de cada eje.

El objetivo primordial de este esquema continúa siendo el problema de **regulación al origen**, definido formalmente mediante:

$$x_{ref} = 0, \qquad y_{ref} = 0 \tag{9.1}$$

Por tanto, en este trabajo no se aborda el problema de seguimiento de trayectorias, sino únicamente el problema de regulación alrededor del punto de equilibrio. Esta decisión es

coherente con el alcance del proyecto, centrado en validar el modelado, la simulación y la estabilización del sistema ante distintas condiciones iniciales.

La estrategia desarrollada en este capítulo se divide en dos etapas. En primer lugar, se diseña un controlador por realimentación de estados suponiendo que el vector completo de estados $\mathbf{x}(t)$ está disponible de forma directa para el cálculo de la ley de control. Esta etapa ideal permite evaluar el comportamiento nominal del controlador sin introducir errores de estimación ni no idealidades asociadas al sistema de medida.

En segundo lugar, se sustituye el estado real por el estado estimado $\hat{\mathbf{x}}(t)$ mediante el filtro de Kalman desarrollado en el Capítulo 7. Esto plantea un escenario más realista y cercano a una posible implementación física, en el cual únicamente se dispone de las medidas de posición procedentes del sensor visual simulado, afectadas por ruido, retardo, cuantización y frecuencia de muestreo limitada.

9.2. Fundamentos del control por realimentación de estados

El modelo lineal del sistema puede expresarse mediante la representación general en espacio de estados:

$$\begin{aligned} \dot{\mathbf{x}}(t) &= A\mathbf{x}(t) + B\mathbf{u}(t) \\ \mathbf{y}(t) &= C\mathbf{x}(t) + D\mathbf{u}(t) \end{aligned} \tag{9.2}$$

donde $\mathbf{x}(t)$ es el vector de estados del sistema, $\mathbf{u}(t)$ es el vector de entradas de control y $\mathbf{y}(t)$ es el vector de salidas medidas. En el caso del modelo empleado en este trabajo, el vector de estados se define como:

$$\mathbf{x}(t) = \begin{bmatrix} x(t) & \dot{x}(t) & y(t) & \dot{y}(t) & \phi(t) & \dot{\phi}(t) & \theta(t) & \dot{\theta}(t) \end{bmatrix}^T \tag{9.3}$$

Las variables $x(t)$ e $y(t)$ representan la posición de la bola sobre la plataforma, mientras que $\phi(t)$ y $\theta(t)$ representan las inclinaciones del plato. Sus respectivas derivadas temporales completan el vector de estados. En una primera aproximación se supondrá que todos los

estados se **encuentran disponibles** para su realimentación. Posteriormente, se introducirá el concepto de observador de estados, con el objetivo de reconstruir el vector completo de estados a partir de un conjunto limitado de medidas.

La realimentación de estados consiste en aplicar una ley de control lineal de la forma [24], [26], [34]:

$$\mathbf{u}(t) = -K\mathbf{x}(t) \tag{9.4}$$

donde K es la matriz de ganancias de realimentación. Sustituyendo esta ley de control en la ecuación 9.2 se obtiene el sistema en lazo cerrado:

$$\begin{aligned} \dot{\mathbf{x}}(t) &= A\mathbf{x}(t) - BK\mathbf{x}(t) \\ \dot{\mathbf{x}}(t) &= (A - BK)\mathbf{x}(t) \end{aligned} \tag{9.5}$$

Por tanto, la dinámica del sistema realimentado queda determinada por los autovalores de la matriz $A - BK$. El diseño de la matriz K permite modificar la posición de los polos del sistema en lazo cerrado [26], [35] y, en consecuencia, ajustar características de la respuesta temporal como la rapidez, el amortiguamiento o el esfuerzo de control requerido.

En este trabajo, al tratarse de un problema de regulación al origen, el vector de estados actúa directamente como vector de error. Es decir, dado que la referencia deseada es nula, se cumple:

$$\mathbf{e}(t) = \mathbf{x}_{ref} - \mathbf{x}(t) = -\mathbf{x}(t) \tag{9.6}$$

De este modo, la ley de control puede interpretarse como una acción proporcional sobre el error de estado. La matriz K pondera la contribución de cada variable del sistema para generar la señal de control adecuada.

Conviene diferenciar conceptualmente la estrategia de regulación al origen frente al seguimiento de referencias no nulas (control de trayectorias). En un problema de regulación

como el planteado, el objetivo es extinguir cualquier desviación del estado respecto a cero. Si se desease ampliar las capacidades del sistema para seguir referencias de posición variables en el tiempo ($x_{ref}(t) \neq 0$ e $y_{ref}(t) \neq 0$), la estructura básica de realimentación de estados requeriría modificaciones adicionales, tales como el uso de un precompensador de referencia o la incorporación de acción integral mediante un servosistema aumentado [26], [35].

La Figura 9.1 ilustra el diagrama de bloques general de un servosistema aumentado diseñado para el seguimiento de referencias. En este esquema, se introduce un estado adicional $\xi(t)$, definido como la integral del error de salida, y una ganancia k_I asociada a dicha acción integral, lo que permite eliminar el error en régimen permanente ante entradas de tipo escalón. Investigaciones previas en la literatura científica han analizado la implementación de control de trayectorias en plataformas 3-RRS mediante técnicas de control avanzadas, como control difuso [21] o estrategias basadas en LQR [22]. En el presente trabajo, no obstante, se empleará la estructura de realimentación de estados sin acción integral, al centrarse el estudio en la regulación al origen.

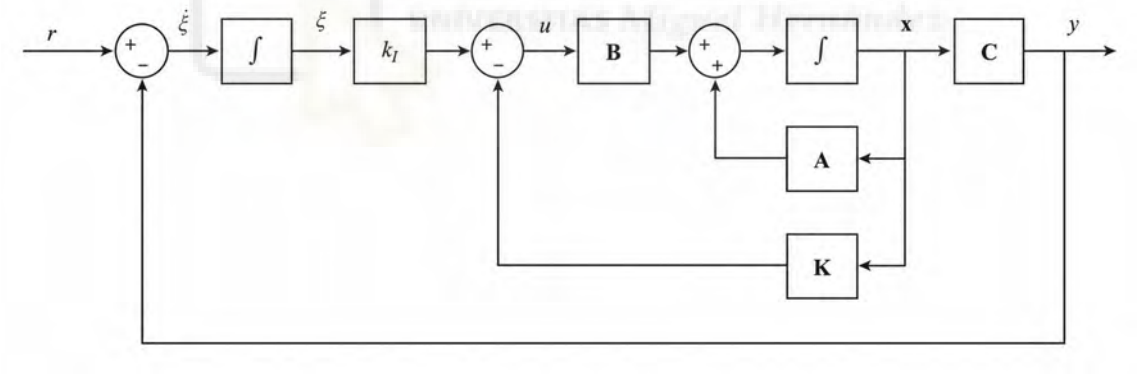


Figura 9.1: Diagrama de bloques general de un servosistema aumentado.

Existen distintos métodos para obtener la matriz de realimentación K. En sistemas SISO, una alternativa clásica consiste en la asignación directa de polos mediante la fórmula de Ackermann [24]. Para sistemas multivariables, como el considerado en este trabajo, puede emplearse la función `place` de MATLAB, que permite ubicar los polos de lazo cerrado cuando el sistema es controlable. Sin embargo, este procedimiento requiere seleccionar explícitamente todos los polos deseados, lo cual puede resultar poco sistemático en sistemas de orden elevado [26].

Por este motivo, en la sección 9.4 se abordará un enfoque basado en el regulador lineal cuadrático, conocido como LQR.

9.3. Análisis de controlabilidad y observabilidad

Antes de diseñar la matriz de realimentación K, es necesario comprobar que el modelo lineal empleado cumple las propiedades estructurales necesarias para aplicar técnicas de control en espacio de estados [24], [26]. En particular, se estudia la **controlabilidad** del sistema, con el fin de verificar si las entradas disponibles permiten gobernar la evolución del vector de estados, y la **observabilidad**, con el objetivo de comprobar si dicho vector puede reconstruirse a partir de las salidas medidas.

Estas propiedades resultan especialmente importantes en el presente trabajo. Por una parte, la controlabilidad condiciona la posibilidad de estabilizar la bola mediante la actuación sobre la orientación de la plataforma. Por otra parte, la observabilidad justifica el uso posterior de un **observador de estados**, como el filtro de Kalman desarrollado en el capítulo 7, para estimar aquellas variables que no se miden directamente.

El análisis se ha realizado mediante un script específico de MATLAB, incluido en el Anexo A.9. Dicho script parte de las matrices A, B, C y D definidas previamente para el modelo lineal del sistema, y calcula los rangos de las matrices de controlabilidad, controlabilidad de salida y observabilidad. Además, en el mismo script se construye posteriormente el modelo equivalente empleado para el diseño LQR, incorporando el efecto del bloque de actuación equivalente (véase sección 4.4.1).

9.3.1. Controlabilidad del modelo

Un sistema lineal e invariante en el tiempo definido por el par (A, B) es controlable si, mediante la aplicación de un vector de entradas admisibles $\mathbf{u}(t)$, es posible llevar el estado desde cualquier condición inicial hasta cualquier estado final en un tiempo finito. Esta propiedad puede comprobarse mediante la matriz de controlabilidad:

$$C = \begin{bmatrix} B & AB & A^2B & \cdots & A^{n-1}B \end{bmatrix} \tag{9.7}$$

donde n es el orden del sistema. El sistema será completamente controlable si dicha matriz tiene rango completo [24], [35]:

$$\text{rank}(C) = n \tag{9.8}$$

En el caso del modelo empleado, el vector de estados (ec. 9.3) está formado por ocho variables, por lo que el orden del sistema es $n = 8$.

La comprobación se realiza en MATLAB mediante la función `ctrb`, tal y como se muestra en el Anexo A.9:

$$C = \texttt{ctrb}(A, B)$$

El resultado obtenido es:

$$\text{rank}(C) = 8$$

Dado que el rango de la matriz de controlabilidad coincide con el número de estados del sistema, se concluye que el modelo lineal es completamente controlable.

Este resultado indica que las entradas de control disponibles permiten actuar sobre todos los modos dinámicos del sistema. Aunque las entradas no modifican directamente la posición de la bola, sí actúan sobre la orientación de la plataforma y, a través de ella, sobre la aceleración de la bola. Por tanto, las variables $x(t)$, $y(t)$ y sus derivadas pueden regularse de forma indirecta mediante la inclinación del plato.

9.3.2. Controlabilidad de la salida

Además de la controlabilidad completa del estado, resulta conveniente analizar si las salidas principales del sistema pueden ser controladas mediante las entradas disponibles. Este análisis se conoce como controlabilidad de la salida y permite estudiar si la acción de control puede influir sobre las variables seleccionadas como salidas de interés [35]. En este trabajo, las variables de mayor interés son las posiciones de la bola sobre la plataforma:

$$\mathbf{y}_{pos}(t) = \begin{bmatrix} x(t) \\ y(t) \end{bmatrix}$$

Por ello, se define una matriz C_{pos} que selecciona únicamente las posiciones $x(t)$ e $y(t)$ del vector de estados:

$$C_{pos} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{9.9}$$

La matriz de controlabilidad de salida se construye como:

$$C_y = \begin{bmatrix} D_{pos} & C_{pos}B & C_{pos}AB & C_{pos}A^2B & \cdots & C_{pos}A^{n-1}B \end{bmatrix}$$

donde D_{pos} es nula en este caso, al no existir transmisión directa entre la entrada y las salidas de posición. La inclusión de este término permite mantener la forma general de la matriz de controlabilidad de salida.

Para que las salidas consideradas sean controlables, el rango de C_y debe coincidir con el número de salidas analizadas:

$$\text{rank}(C_y) = p$$

donde, en este caso $p = 2$. El cálculo implementado en el Anexo A.9 proporciona:

$$\text{rank}(C_y) = 2$$

Por tanto, se concluye que las salidas de posición $x(t)$ e $y(t)$ son controlables. Este resultado es coherente con el comportamiento físico del sistema: las entradas de control modifican la inclinación de la plataforma y permiten generar las aceleraciones necesarias para desplazar la bola hacia el punto de equilibrio.

9.3.3. Observabilidad del sistema

La observabilidad determina si es posible reconstruir el estado interno del sistema a partir de las salidas medidas y de la evolución conocida de las entradas [34], [35]. Esta propiedad es fundamental para la segunda parte del capítulo, en la que se empleará un filtro de Kalman para estimar el vector completo de estados a partir de las medidas proporcionadas por el sensor visual simulado.

El sistema definido por el par (A, C) es observable si la matriz de observabilidad:

$$\mathcal{O} = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{n-1} \end{bmatrix} \tag{9.10}$$

tiene rango completo:

$$\text{rank}(\mathcal{O}) = n$$

En este trabajo, el análisis de observabilidad se realiza considerando una situación más realista que la de estado completo. En concreto, se supone que el sistema de medida proporciona únicamente la posición de la bola en los ejes X e Y, de forma coherente con el sensor visual simulado:

$$\mathbf{y}_{med}(t) = \begin{bmatrix} x_m(t) \\ y_m(t) \end{bmatrix} \tag{9.11}$$

Por tanto, la matriz de salida C_{med} empleada para este análisis coincide con C_{pos} (ec. 9.9). La comprobación se realiza en MATLAB mediante la función `obsv`, tal y como se recoge en el Anexo A.9:

$$\mathcal{O} = \texttt{obsv}(A, C_{med})$$

El resultado obtenido es:

$$\text{rank}(\mathcal{O}) = 8$$

Dado que el rango de la matriz de observabilidad coincide con el número de estados del sistema, se concluye que el modelo es completamente observable [34], [35] a partir de las medidas de posición $x_m(t)$ e $y_m(t)$.

Este resultado indica que, aunque el sensor visual simulado proporcione únicamente la posición de la bola, el resto de variables del estado puede reconstruirse mediante un observador, siempre que el modelo dinámico represente adecuadamente el comportamiento del sistema. En particular, pueden estimarse las velocidades de la bola, las inclinaciones de la plataforma y sus velocidades angulares.

Finalmente, dado que en el diseño del controlador se mantiene el bloque de actuación equivalente, en el Anexo A.9 también se comprueba la controlabilidad del modelo equivalente definido por las matrices A_{eq} y B_{eq}. Dicho modelo será empleado en la sección siguiente para el diseño de la matriz K mediante LQR. El resultado obtenido es:

$$\text{rank}(C_{eq}) = 8$$

Por tanto, la inclusión de la dinámica de actuación equivalente no compromete la contro-

labilidad del sistema, lo que permite continuar con el diseño del regulador sobre el modelo equivalente planta-actuador.

9.4. Diseño de la matriz de realimentación mediante LQR

Una vez comprobadas la controlabilidad y observabilidad del modelo, se procede al diseño de la matriz de realimentación K. A diferencia del servosistema aumentado mostrado en la Figura 9.1, en este trabajo se adopta una estructura de realimentación de estados sin acción integral, al centrarse el problema en la regulación al origen. La Figura 9.2 muestra el esquema general correspondiente, en el que la señal de control se obtiene a partir de la realimentación del vector de estados mediante la matriz K.

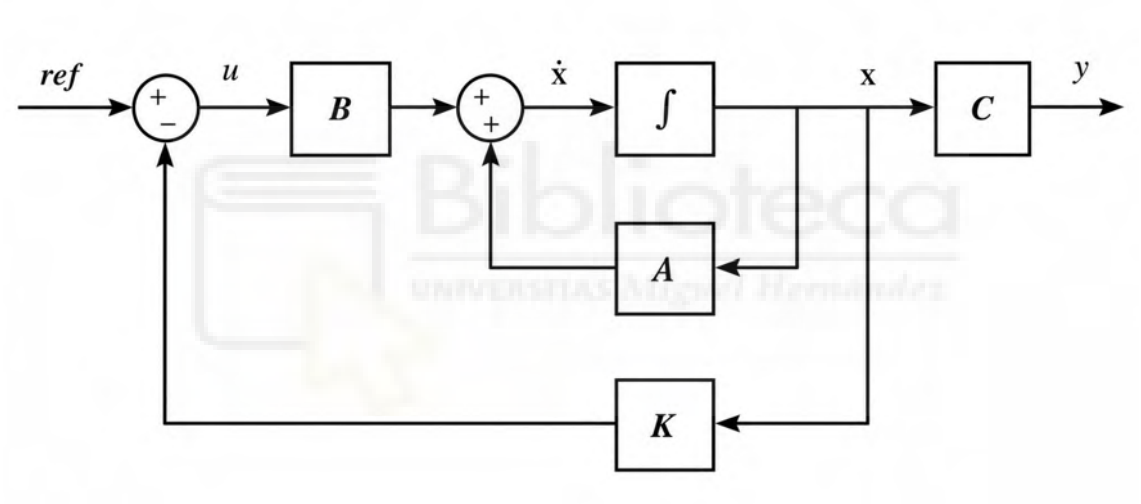


Figura 9.2: Diagrama de bloques de un sistema con realimentación.

Tal y como se ha indicado anteriormente, existen distintas metodologías para obtener dicha matriz. En este trabajo se opta por emplear el regulador lineal cuadrático, conocido como LQR (*Linear Quadratic Regulator*). Esta técnica permite obtener la matriz de realimentación de estados a partir de la minimización de un índice de coste cuadrático, estableciendo un compromiso entre la rapidez de respuesta del sistema y el esfuerzo de control requerido.

El procedimiento completo de construcción del modelo equivalente, selección de matrices de ponderación y cálculo de la ganancia K se recoge en el script del Anexo A.9.

9.4.1. Justificación del método LQR

El regulador LQR busca una ley de control lineal, tal como se estudió en la ecuación 9.4, que minimice el funcional de coste:

$$J = \int_0^\infty \left[\mathbf{x}^T(t)Q\mathbf{x}(t) + \mathbf{u}^T(t)R\mathbf{u}(t)\right] dt \tag{9.12}$$

donde Q es una matriz semidefinida positiva que pondera la penalización de las variables de estado, y R es una matriz definida positiva que pondera el esfuerzo de control . La matriz Q permite dar mayor o menor importancia a determinadas variables del sistema, mientras que R permite limitar la agresividad de la señal de control [26].

Desde el punto de vista físico, una mayor penalización sobre las posiciones de la bola $x(t)$ e $y(t)$ tiende a generar una corrección más rápida de la desviación respecto al centro de la plataforma. Sin embargo, esto suele implicar referencias de inclinación más elevadas y, por tanto, un mayor esfuerzo de actuación. Por el contrario, aumentar los valores de la matriz R penaliza más intensamente la acción de control, dando lugar a movimientos más suaves de la plataforma, aunque generalmente con una respuesta más lenta.

Esta metodología resulta especialmente adecuada para el sistema estudiado, ya que permite ponderar simultáneamente el error de estado y el esfuerzo de control. A diferencia de una asignación manual de polos, el LQR proporciona una forma sistemática de obtener la matriz K, evitando la selección directa de todos los polos del sistema en lazo cerrado.

Además, el método se adapta de forma natural al carácter multivariable del modelo. En lugar de diseñar un controlador independiente para cada eje, como ocurría en el controlador PD del capítulo anterior, la matriz K se obtiene considerando el conjunto completo de estados y entradas. Por tanto, la acción de control puede tener en cuenta simultáneamente la posición y velocidad de la bola, así como la orientación y velocidad angular del plato.

9.4.2. Construcción del modelo equivalente con actuación equivalente

En el modelo lineal obtenido en el apartado 4.3.4, las entradas del sistema se corresponden con los pares aplicados sobre la plataforma en los ejes de inclinación, de forma que la ecuación de estado puede expresarse como:

$$\dot{\mathbf{x}}(t) = A\mathbf{x}(t) + B\boldsymbol{\tau}(t) \tag{9.13}$$

donde:

$$\boldsymbol{\tau}(t) = \begin{bmatrix} \tau_\phi(t) \\ \tau_\theta(t) \end{bmatrix}$$

representa el vector de pares equivalentes aplicados al plato.

Sin embargo, esta formulación no coincide directamente con la estructura de control implementada en Simulink. En la práctica, el regulador no genera como salida dichos pares, sino unas referencias de inclinación para la plataforma, es decir:

$$\mathbf{u}_{ref}(t) = \begin{bmatrix} \phi_{ref}(t) \\ \theta_{ref}(t) \end{bmatrix} \tag{9.14}$$

Por tanto, entre la ley de control LQR y la planta en espacio de estados debe incorporarse una etapa previa de actuación equivalente, encargada de transformar las referencias angulares en los pares equivalentes realmente aplicados al sistema.

La Figura 9.3 muestra de forma conceptual la estructura adoptada. En ella puede observarse que la realimentación de estados, a través de la ganancia K, genera las referencias de inclinación de la plataforma. Dichas referencias pasan por un bloque de saturación, que limita el valor máximo admisible de la inclinación, y posteriormente se introducen en el actuador equivalente. Finalmente, este subsistema convierte las referencias angulares ϕ_{ref} y θ_{ref} en los pares equivalentes τ_ϕ y τ_θ, que son las entradas efectivas del bloque de espacio de estados.

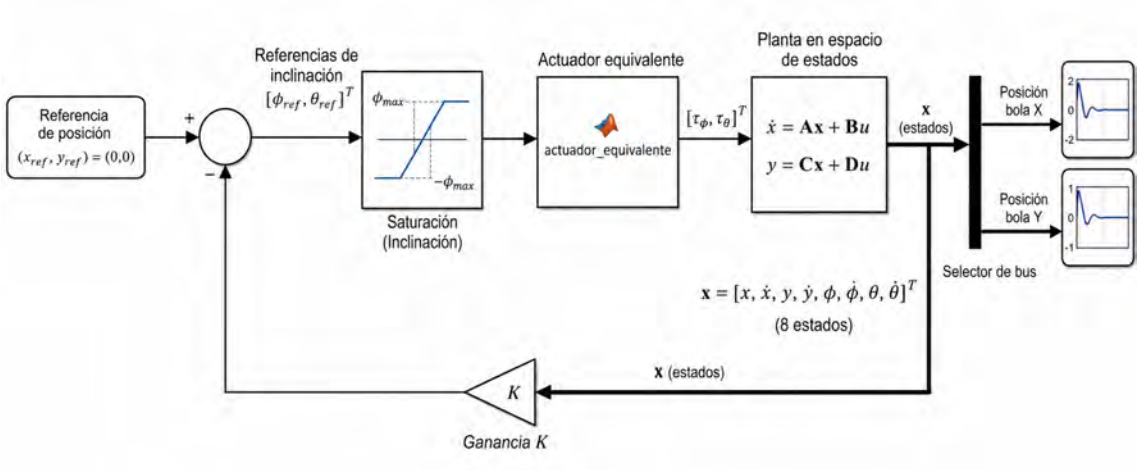


Figura 9.3: Estructura conceptual del controlador LQR con bloque de actuación equivalente.

El bloque de actuación equivalente utilizado en este trabajo implementa una dinámica de segundo orden para cada eje de inclinación mediante una ley proporcional-derivativa interna (véase Anexo A.3). Para el eje ϕ, el par equivalente generado viene dado por:

$$\tau_\phi(t) = K_{p,\phi}\big(\phi_{ref}(t) - \phi(t)\big) - K_{d,\phi}\dot{\phi}(t)$$

y, de forma análoga, para el eje θ:

$$\tau_\theta(t) = K_{p,\theta}\big(\theta_{ref}(t) - \theta(t)\big) - K_{d,\theta}\dot{\theta}(t)$$

Las ganancias del actuador equivalente se seleccionan imponiendo una dinámica de segundo orden con frecuencia natural ω_n y coeficiente de amortiguamiento ζ. De este modo:

$$K_{p,\phi} = J_\phi \omega_n^2, \qquad K_{d,\phi} = 2\zeta J_\phi \omega_n - b_\phi$$

$$K_{p,\theta} = J_\theta \omega_n^2, \qquad K_{d,\theta} = 2\zeta J_\theta \omega_n - b_\theta$$

donde J_ϕ y J_θ son los momentos de inercia equivalentes del plato respecto a cada eje, y b_ϕ, b_θ representan los coeficientes de amortiguamiento viscoso incluidos en la planta. La resta del término b en la ganancia derivativa se justifica porque dicho amortiguamiento ya

forma parte del modelo dinámico original.

Desarrollando las ecuaciones anteriores, el vector de pares puede escribirse como:

$$\boldsymbol{\tau}(t) = G\mathbf{u}_{ref}(t) - F\mathbf{x}(t) \tag{9.15}$$

donde $\mathbf{x}(t)$ es el vector de estados del sistema, y las matrices F y G toman la forma:

$$F = \begin{bmatrix} 0 & 0 & 0 & 0 & K_{p,\phi} & K_{d,\phi} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & K_{p,\theta} & K_{d,\theta} \end{bmatrix}$$

$$G = \begin{bmatrix} K_{p,\phi} & 0 \\ 0 & K_{p,\theta} \end{bmatrix}$$

Sustituyendo esta relación en el modelo original de la planta (ec. 9.13) se obtiene:

$$\dot{\mathbf{x}}(t) = A\mathbf{x}(t) + B\left(G\mathbf{u}_{ref}(t) - F\mathbf{x}(t)\right)$$

$$\dot{\mathbf{x}}(t) = (A - BF)\,\mathbf{x}(t) + BG\mathbf{u}_{ref}(t) \tag{9.16}$$

Por tanto, el sistema equivalente visto desde el controlador queda descrito por las matrices:

$$A_{eq} = A - BF \tag{9.17}$$

$$B_{eq} = BG \tag{9.18}$$

Este nuevo modelo conserva el mismo vector de estados que la planta original, pero sustituye como entradas los pares mecánicos por las referencias angulares ϕ_{ref} y θ_{ref}. En consecuencia, la ley de control adoptada en este trabajo pasa a expresarse como:

$$\begin{bmatrix} \phi_{ref}(t) \\ \theta_{ref}(t) \end{bmatrix} = -K\mathbf{x}(t)$$

en lugar de actuar directamente sobre τ_ϕ y τ_θ.

La construcción de las matrices F, G, A_{eq} y B_{eq}, así como la verificación de la controlabilidad del sistema equivalente, se implementan en MATLAB mediante el script incluido en el Anexo A.9. Tal y como se comprobó en la sección 9.3, el modelo equivalente conserva la propiedad de controlabilidad, obteniéndose:

$$\text{rank}(C_{eq}) = 8$$

Por tanto, la inclusión del bloque de actuación equivalente no compromete la posibilidad de controlar todos los modos dinámicos del sistema, sino que permite reformular el problema de manera más coherente con la arquitectura de control realmente empleada.

9.4.3. Selección de las matrices Q y R

La selección de las matrices Q y R determina el comportamiento del controlador LQR. En este trabajo se emplea una metodología basada en la regla de Bryson, utilizada habitualmente como criterio inicial de sintonización en diseños LQR [29]. Esta regla consiste en ponderar cada variable según el valor máximo admisible que se considera aceptable durante la operación del sistema. La documentación de MATLAB para la función `lqr` también recoge este criterio como una forma práctica de definir valores iniciales para las matrices Q y R, si bien señala que normalmente constituye un punto de partida para un proceso posterior de ajuste de la respuesta en lazo cerrado [36].

De forma general, para una variable de estado x_i, el peso correspondiente se define como:

$$Q_{ii} = \frac{1}{x_{i,max}^2}$$

mientras que, para una entrada de control u_j, se define:

$$R_{jj} = \frac{1}{u_{j,max}^2}$$

Aplicando este criterio al vector de estados del sistema, la matriz Q se construye como:

$$Q = \text{diag}\left(\frac{1}{x_{max}^2}, \frac{1}{\dot{x}_{max}^2}, \frac{1}{y_{max}^2}, \frac{1}{\dot{y}_{max}^2}, \frac{1}{\phi_{max}^2}, \frac{1}{\dot{\phi}_{max}^2}, \frac{1}{\theta_{max}^2}, \frac{1}{\dot{\theta}_{max}^2}\right) \tag{9.19}$$

Dado que la señal de control generada por el LQR está formada por las referencias angulares, la matriz R se define como:

$$R = \text{diag}\left(\frac{1}{\phi_{ref,max}^2}, \frac{1}{\theta_{ref,max}^2}\right) \tag{9.20}$$

Los valores máximos considerados en el diseño son los empleados en el script del Anexo A.9 y se recogen en la Tabla 9.1.

Tabla 9.1: Valores máximos admisibles empleados para la selección de Q y R.

Variable	Valor máximo admisible	Unidad
x_{max}	0.03	m
$\dot{x}_{max}$	0.10	m/s
y_{max}	0.03	m
$\dot{y}_{max}$	0.10	m/s
ϕ_{max}	5	°
$\dot{\phi}_{max}$	1.0	rad/s
θ_{max}	5	°
$\dot{\theta}_{max}$	1.0	rad/s
$\phi_{ref,max}$	10	°
$\theta_{ref,max}$	10	°

En el script de MATLAB, los valores angulares se introducen en radianes, aunque en la tabla se expresan en grados para facilitar su interpretación física. La elección de x_{max} e y_{max} está relacionada con la desviación máxima admisible de la bola respecto al centro de la plataforma. Por su parte, los límites de ϕ_{max}, θ_{max}, $\phi_{ref,max}$ y $\theta_{ref,max}$ se seleccionan de forma coherente con las restricciones de inclinación empleadas en los capítulos anteriores.

El uso de una referencia angular máxima mayor que la inclinación máxima deseada del plato permite al controlador solicitar correcciones suficientemente rápidas, mientras que la penalización de R evita que dichas referencias alcancen valores excesivos. Esta selección establece un compromiso entre rapidez de estabilización y suavidad de la actuación.

9.4.4. Ganancia obtenida y polos de lazo cerrado

Una vez definidas las matrices Q y R, la matriz de realimentación se obtiene en MATLAB mediante:

$$K = \texttt{lqr}(A_{eq}, B_{eq}, Q, R) \tag{9.21}$$

Este cálculo se implementa en el script del Anexo A.9. La matriz K obtenida tiene dimensión 2×8, ya que el sistema dispone de dos entradas de control y ocho estados:

$$K = \begin{bmatrix} k_{11} & k_{12} & k_{13} & k_{14} & k_{15} & k_{16} & k_{17} & k_{18} \\ k_{21} & k_{22} & k_{23} & k_{24} & k_{25} & k_{26} & k_{27} & k_{28} \end{bmatrix}$$

La primera fila de K se emplea para calcular la referencia ϕ_{ref}, mientras que la segunda fila se emplea para calcular la referencia θ_{ref}:

$$\phi_{ref}(t) = -\begin{bmatrix} k_{11} & k_{12} & \cdots & k_{18} \end{bmatrix} \mathbf{x}(t)$$

$$\theta_{ref}(t) = -\begin{bmatrix} k_{21} & k_{22} & \cdots & k_{28} \end{bmatrix} \mathbf{x}(t)$$

Sustituyendo los valores numéricos obtenidos en MATLAB:

$$K = \begin{bmatrix} 5{,}8178 & 3{,}0742 & -0{,}0000 & 0{,}0000 & 2{,}8570 & 0{,}1340 & -0{,}0000 & 0{,}0000 \\ 0{,}0000 & 0{,}0000 & 5{,}8178 & 3{,}0742 & 0{,}0000 & 0{,}0000 & 2{,}8570 & 0{,}1340 \end{bmatrix} \tag{9.22}$$

Los polos del sistema equivalente en lazo cerrado se calculan como los autovalores de:

$$A_{lc} = A_{eq} - B_{eq}K$$

es decir:

$$\lambda_{lc} = \text{eig}(A_{eq} - B_{eq}K)$$

El resultado obtenido es:

$$\lambda_{lc} = \begin{bmatrix} -4{,}4174 + 1{,}2586i \\ -4{,}4174 - 1{,}2586i \\ -74{,}3639 + 0{,}0000i \\ -74{,}3639 + 0{,}0000i \\ -10{,}3936 + 0{,}0000i \\ -10{,}3936 + 0{,}0000i \\ -4{,}4174 + 1{,}2586i \\ -4{,}4174 - 1{,}2586i \end{bmatrix} \tag{9.23}$$

Todos los polos presentan parte real negativa, por lo que el sistema equivalente en lazo cerrado es estable. La localización de estos polos determina la rapidez y el amortiguamiento de la respuesta temporal. En particular, los polos más alejados del eje imaginario se asocian a modos rápidos, mientras que los polos dominantes, más próximos al eje imaginario, condicionan principalmente el tiempo de establecimiento del sistema.

La Figura 9.4 muestra la localización de los polos del sistema en lazo cerrado en el plano complejo. La aparición de polos repetidos se debe a la simetría del modelo respecto a los ejes X e Y. Bajo las hipótesis de pequeñas inclinaciones y operación cerca del punto de equilibrio, las dinámicas $x - \phi$ e $y - \theta$ resultan equivalentes. Además, se han empleado los mismos parámetros físicos y las mismas ponderaciones LQR para ambos ejes, por lo que el sistema presenta modos de lazo cerrado duplicados. Esta misma simetría se observa en la matriz K de la ecuación 9.22, cuyos términos cruzados son nulos o numéricamente despreciables.

La ubicación de los polos del sistema en lazo cerrado en el semiplano izquierdo del plano complejo confirma que la matriz K obtenida estabiliza el sistema alrededor del punto de equilibrio.

9.5. Implementación en Simulink sin observador

Una vez diseñada la matriz de realimentación K mediante el regulador LQR, se procede a su implementación en Simulink. En esta primera etapa se considera una **situación ideal**

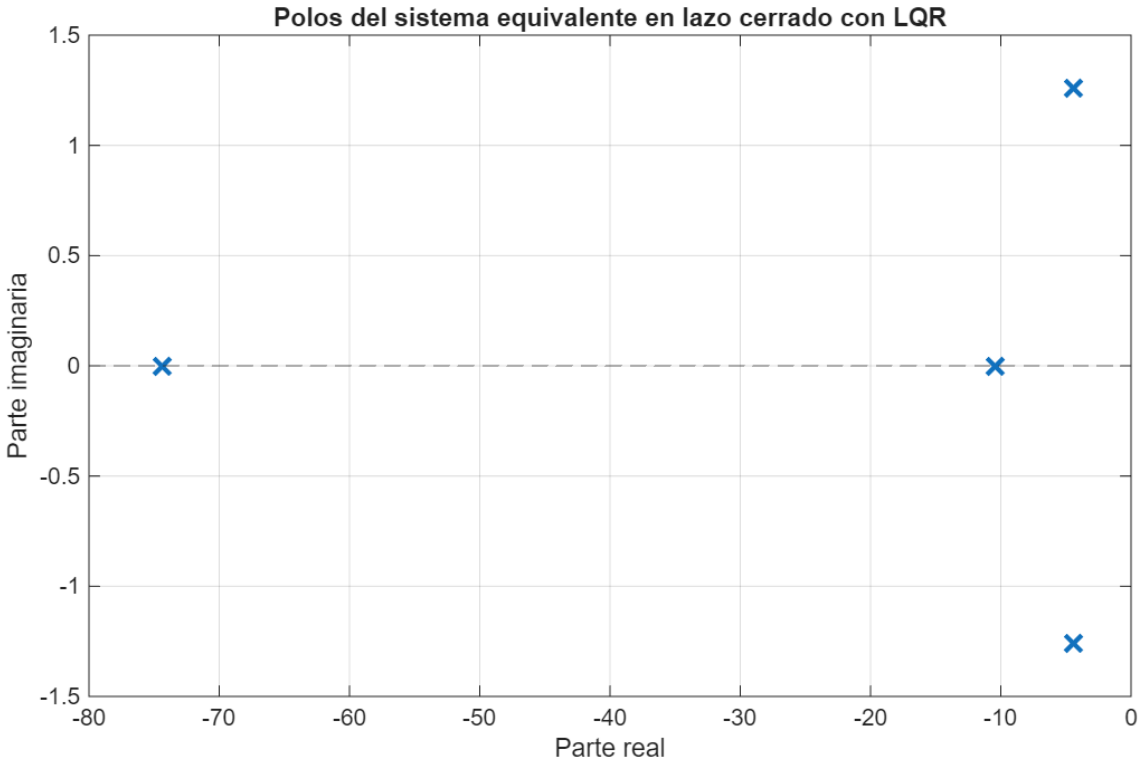


Figura 9.4: Polos del sistema equivalente en lazo cerrado con controlador LQR.

en la que el vector completo de estados se encuentra disponible directamente a la salida del bloque de espacio de estados. Por tanto, no se incorpora todavía ningún observador ni sistema de medida no ideal, sino que se emplea la información interna exacta de la planta para calcular la acción de control.

9.5.1. Esquema de simulación y condiciones de ensayo

La implementación del controlador LQR sin observador se muestra en la Figura 9.5. En él puede observarse la estructura completa del lazo cerrado, formada principalmente por la realimentación del estado completo, el bloque de saturación, el actuador equivalente y la planta lineal en espacio de estados.

El vector de estados se extrae directamente de la salida del bloque de espacio de estados y se realimenta mediante la matriz K. La señal generada por el controlador se limita mediante un bloque de saturación antes de aplicarse al actuador equivalente, de forma que las referencias de inclinación se mantienen dentro de los límites admisibles ($\pm 10°$).

La estructura implementada permite registrar las principales variables de interés durante

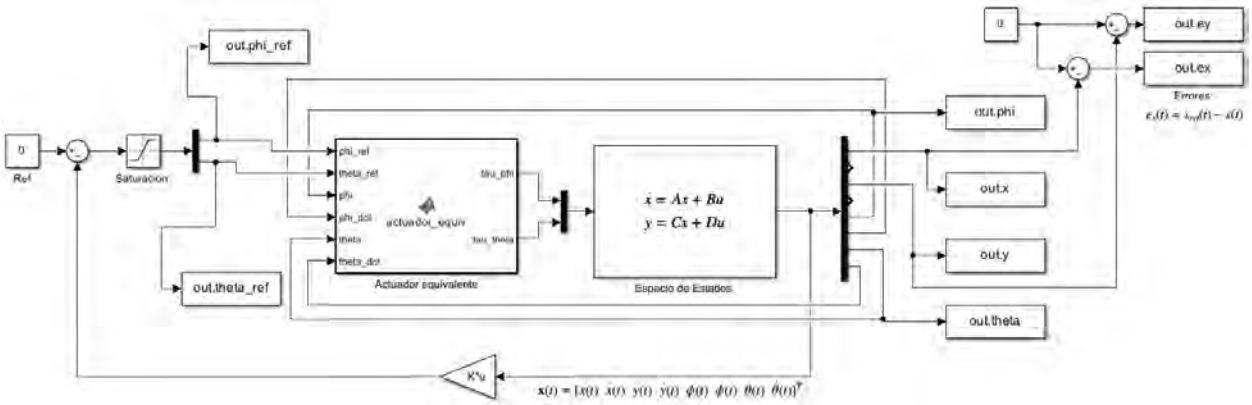


Figura 9.5: Esquema de Simulink del controlador LQR con realimentación del estado completo y sin observador.

la simulación: las posiciones de la bola $x(t)$ e $y(t)$, las inclinaciones de la plataforma $\phi(t)$ y $\theta(t)$, las referencias de inclinación generadas por el controlador $\phi_{ref}(t)$ y $\theta_{ref}(t)$ y los errores de posición respecto al origen. Estos errores se calculan como:

$$\begin{aligned} e_x(t) &= x_{ref}(t) - x(t) \\ e_y(t) &= y_{ref}(t) - y(t) \end{aligned} \tag{9.24}$$

Dado que el objetivo del capítulo es la regulación al origen, las referencias de posición se mantienen constantes e iguales a cero, por lo que:

$$\begin{aligned} e_x(t) &= -x(t) \\ e_y(t) &= -y(t) \end{aligned} \tag{9.25}$$

No obstante, debe destacarse que estos errores de posición se utilizan principalmente como magnitudes de evaluación. A diferencia del controlador PD, el LQR no calcula la acción de control únicamente a partir del error de posición, sino a partir del vector completo de estados. Por tanto, la comparación entre ambos controladores debe realizarse mediante variables físicas comunes, como el tiempo de establecimiento, el error final o la inclinación máxima solicitada.

Para evaluar el comportamiento del controlador se emplean los mismos escenarios de prueba definidos en la Tabla 8.2 del capítulo 8. De este modo, la comparación posterior con el controlador PD se realiza bajo las mismas condiciones iniciales. En todos los ensayos

se mantiene la referencia de posición nula, y se analiza la capacidad del controlador para devolver la bola al centro de la plataforma.

A partir de las variables registradas se obtienen los indicadores de comportamiento utilizados en el análisis:

$$t_s, \quad x_{max}, \quad y_{max}, \quad \phi_{ref,max}, \quad \theta_{ref,max}, \quad \phi_{max}, \quad \theta_{max}, \quad x_f, \quad y_f$$

donde t_s representa el tiempo de establecimiento del sistema, x_{max} e y_{max} son los desplazamientos máximos alcanzados por la bola durante la simulación, y $\phi_{ref,max}$, $\theta_{ref,max}$, ϕ_{max} y θ_{max} permiten evaluar tanto las referencias de inclinación máximas solicitadas por el controlador como la orientación real alcanzada por la plataforma. Finalmente, x_f e y_f representan las posiciones de la bola en el instante final de la simulación, es decir, tras 5 segundos.

El tiempo de establecimiento se calcula empleando la misma banda de tolerancia considerada en el capítulo anterior (ecuación 8.7), lo que permite comparar de forma directa la respuesta del controlador LQR con la obtenida mediante el controlador PD. Al tratarse de un problema de regulación al origen, las posiciones finales x_f e y_f pueden interpretarse directamente como una medida del error residual del sistema y estarán muy próximas a cero.

9.5.2. Resultados de simulación sin observador

Las simulaciones realizadas muestran que el controlador LQR con realimentación del estado completo es capaz de estabilizar la bola en el origen para todos los escenarios de prueba considerados. Al no incluirse todavía el sensor visual simulado ni el observador de Kalman, el controlador trabaja directamente con el vector de estados real proporcionado por la planta. Por tanto, los resultados obtenidos en este apartado, recogidos en las Tablas 9.2 y 9.3, representan el comportamiento nominal del regulador diseñado.

En primer lugar, se observa que el sistema alcanza la banda de tolerancia en todos los ensayos, con tiempos de establecimiento comprendidos entre 0.796 s y 1.256 s. Los

Tabla 9.2: Resultados de posición obtenidos con controlador LQR sin observador.

Ensayo	t_s [s]	**máx** $\lvert x \rvert$ [m]	**máx** $\lvert y \rvert$ [m]	x_f [m]	y_f [m]
1	0.803	0.0300	0.0000	$3{,}13 \cdot 10^{-12}$	≈ 0
2	0.803	0.0000	0.0300	≈ 0	$3{,}13 \cdot 10^{-12}$
3	0.803	0.0300	0.0300	$3{,}13 \cdot 10^{-12}$	$-3{,}13 \cdot 10^{-12}$
4	0.796	0.0154	0.0000	$-7{,}06 \cdot 10^{-12}$	≈ 0
5	0.911	0.0297	0.0252	$-4{,}01 \cdot 10^{-12}$	$1{,}43 \cdot 10^{-12}$
6	1.256	0.1014	0.0600	$-3{,}73 \cdot 10^{-11}$	$-1{,}30 \cdot 10^{-11}$

Tabla 9.3: Referencias de inclinación y orientación máxima de la plataforma con controlador LQR sin observador.

Ensayo	**máx** $\lvert \phi_{ref} \rvert$ [°]	**máx** $\lvert \theta_{ref} \rvert$ [°]	**máx** $\lvert \phi \rvert$ [°]	**máx** $\lvert \theta \rvert$ [°]
1	10.00	0.00	1.94	0.00
2	0.00	10.00	0.00	1.94
3	10.00	10.00	1.94	1.94
4	10.00	0.00	3.94	0.00
5	10.00	10.00	4.44	3.25
6	10.00	2.39	9.24	1.32

ensayos 1, 2 y 3 presentan el mismo tiempo de establecimiento, $t_s = 0{,}803$ s, lo cual resulta coherente con la simetría del modelo respecto a los ejes X e Y. En estos casos, las condiciones iniciales afectan de forma equivalente a uno o ambos ejes, y el controlador genera respuestas prácticamente idénticas en cada dirección.

El ensayo 4 presenta un tiempo de establecimiento ligeramente inferior, $t_s = 0{,}796$ s, pese a alcanzar una inclinación real máxima mayor que en los primeros ensayos. Esto se debe a que la condición inicial y la acción de control asociada producen una corrección más rápida de la desviación inicial, aunque con una demanda de orientación más elevada sobre la plataforma.

En el ensayo 5, el tiempo de establecimiento aumenta hasta 0.911 s. En este caso se excitan simultáneamente ambos ejes, con valores máximos de posición $x_{max} = 0{,}0297$ m e $y_{max} = 0{,}0252$ m. Las referencias de inclinación alcanzan el límite impuesto de 10° en ambos ejes, y las inclinaciones reales máximas de la plataforma son 4,44° y 3,25°, respectivamente. Esto indica que el bloque de saturación limita la actuación solicitada por el controlador, pero el sistema mantiene una respuesta estable y convergente.

El ensayo 6 constituye el caso más exigente, con los mayores desplazamientos máximos

registrados, x_{max} = 0,1014 m e y_{max} = 0,0600 m. En este ensayo el tiempo de establecimiento aumenta hasta 1.256 s, como consecuencia de la mayor desviación inicial de la bola y de la proximidad a los límites de operación de la plataforma. En la Figura 9.6(a) puede observarse que la posición de la bola converge progresivamente hacia el origen, sin oscilaciones sostenidas ni divergencias. La respuesta resulta coherente con la ubicación de los polos dominantes obtenidos en el apartado anterior, los cuales presentaban parte real negativa y un amortiguamiento elevado.

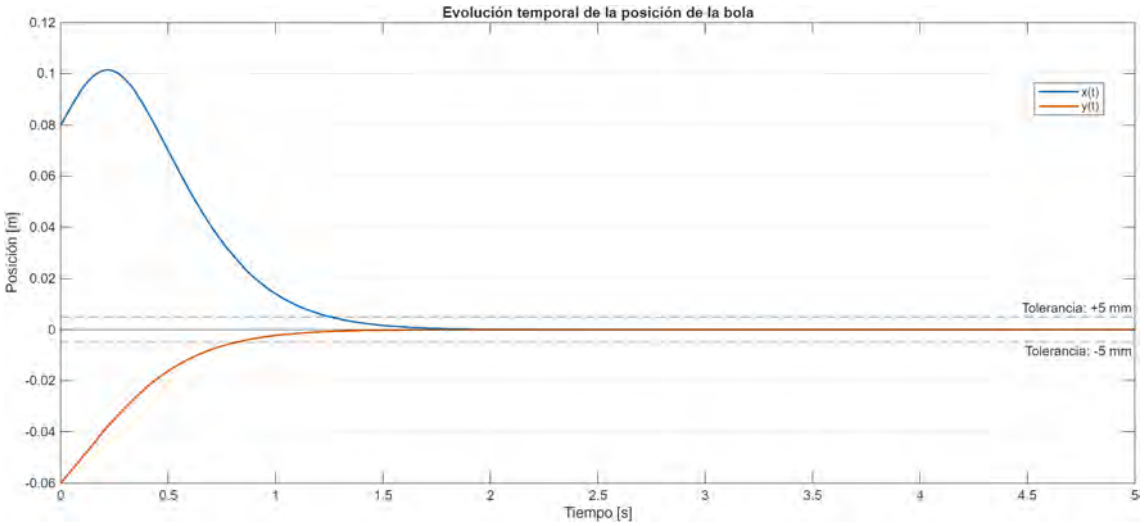
Respecto a la acción de control, la referencia de inclinación en el eje ϕ alcanza el límite de saturación de 10°, mientras que la referencia θ_{ref} permanece en 2,39°. Los resultados se pueden apreciar en la Figura 9.6(b). La inclinación real máxima en ϕ llega a 9,24°, valor cercano al límite físico considerado, mientras que la inclinación en θ se mantiene en 1,32°. Este resultado muestra que el controlador concentra la acción correctora principalmente en el eje más exigido, manteniendo una actuación más moderada en el otro eje.

Los valores finales x_f e y_f son del orden de 10^{-12} – 10^{-11} m, por lo que pueden considerarse nulos a efectos prácticos. Estas pequeñas desviaciones no representan un error estacionario apreciable, sino residuos numéricos asociados a la precisión del integrador y al cálculo computacional. En consecuencia, puede afirmarse que el controlador LQR elimina prácticamente por completo el error final en las condiciones ideales de simulación consideradas.

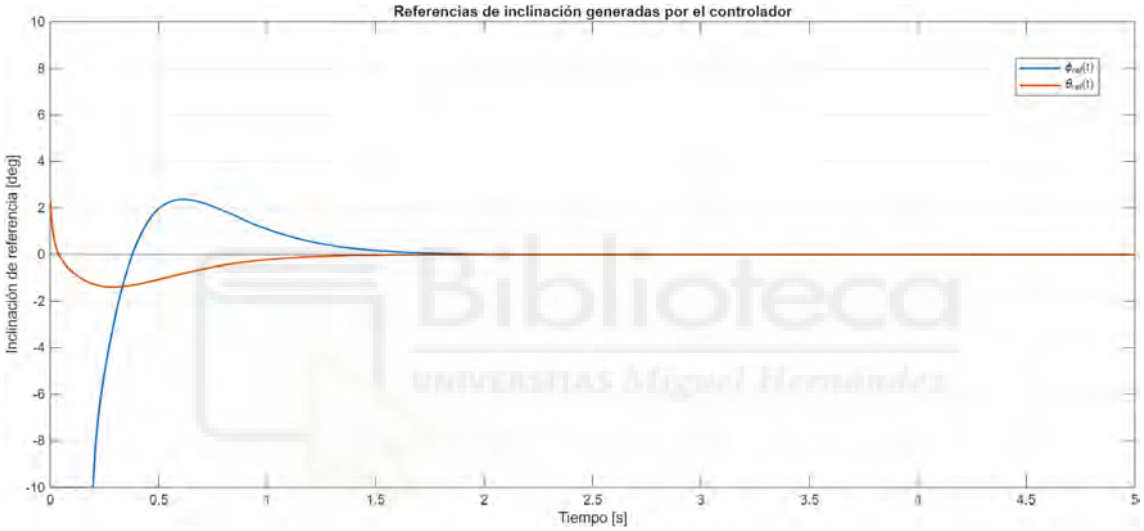
También se aprecia que las referencias de inclinación alcanzan el límite de 10° en varios ensayos. Esto indica que, para las condiciones iniciales más exigentes, el controlador solicita inicialmente la máxima inclinación permitida con el objetivo de devolver la bola al origen lo más rápidamente posible. Sin embargo, la orientación real de la plataforma no coincide necesariamente con dicha referencia máxima, debido a la dinámica introducida por el bloque de actuación equivalente. Este comportamiento es especialmente claro en los ensayos 1, 2 y 3, donde la referencia alcanza 10°, pero la inclinación real máxima se mantiene en torno a 1,94°.

La respuesta obtenida está directamente condicionada por la elección de las matrices de ponderación Q y R, estudiadas anteriormente en el apartado 9.4.1.

En este sentido, los resultados obtenidos muestran el compromiso impuesto por la selección



(a) Posición de la bola.



(b) Referencias de inclinación generadas por el controlador.

Figura 9.6: Respuesta temporal del sistema sin observador para el ensayo 6.

de dichas matrices: el controlador logra estabilizar la bola en tiempos reducidos, pero en los ensayos más exigentes llega a solicitar la inclinación máxima permitida. Esto confirma la importancia de seleccionar las ponderaciones no solo buscando rapidez, sino también teniendo en cuenta las restricciones físicas de la plataforma y la dinámica del actuador equivalente.

En conjunto, los resultados confirman que la ganancia K diseñada mediante LQR estabiliza correctamente el modelo equivalente cuando se dispone del estado completo. La respuesta obtenida es estable, amortiguada y con error final prácticamente nulo. Además, la simetría observada entre los ensayos asociados a los ejes X e Y es coherente con la estructura de la matriz K y con la duplicidad de polos obtenida en el apartado anterior.

9.6. Simulaciones con sensor visual simulado y observador de Kalman

En el apartado anterior se ha evaluado el controlador LQR suponiendo disponible el vector completo de estados de la planta. Esta hipótesis permite analizar el comportamiento nominal del regulador, pero no representa una situación realista de implementación, ya que el sistema de medida considerado en este trabajo proporciona únicamente información de la posición de la bola sobre la plataforma.

Por este motivo, en esta sección se incorpora el sensor visual simulado desarrollado en el capítulo 7, junto con el filtro de Kalman empleado como observador de estados. De este modo, la realimentación deja de realizarse con el estado real $\mathbf{x}(t)$ y pasa a utilizar el estado estimado $\hat{\mathbf{x}}(t)$, obtenido a partir de las medidas de posición $x_m(t)$ e $y_m(t)$ y del modelo dinámico del sistema.

Esta configuración permite evaluar el comportamiento del controlador en un escenario más cercano a una posible implementación física, incorporando las no idealidades del sistema de visión, tales como ruido de medida, retardo, cuantización y frecuencia de muestreo limitada.

9.6.1. Realimentación del estado estimado

En una implementación real, no todas las variables del vector de estados pueden medirse directamente. En particular, el sensor visual simulado proporciona únicamente las coordenadas de la bola sobre la plataforma:

$$\mathbf{y}_m(t) = \begin{bmatrix} x_m(t) \\ y_m(t) \end{bmatrix}$$

Estas medidas no coinciden exactamente con las posiciones reales $x(t)$ e $y(t)$, ya que incorporan las no idealidades descritas en la sección 7.4. Además, el controlador LQR requiere el vector completo de estados (ec. 9.3) para calcular la acción de control.

Por tanto, resulta necesario incorporar un observador de estados capaz de reconstruir las variables no medidas a partir de las medidas disponibles y del modelo dinámico del sistema. La Figura 9.7 incorpora el observador al diagrama de bloques general mostrado anteriormente en la Figura 9.2. En este trabajo se emplea el filtro de Kalman introducido en el apartado 7.5.2, que proporciona una estimación del vector completo de estados:

$$\hat{\mathbf{x}}(t) = \begin{bmatrix} \hat{x}(t) & \hat{\dot{x}}(t) & \hat{y}(t) & \hat{\dot{y}}(t) & \hat{\phi}(t) & \hat{\dot{\phi}}(t) & \hat{\theta}(t) & \hat{\dot{\theta}}(t) \end{bmatrix}^T \tag{9.26}$$

La ley de control implementada deja entonces de depender del estado real y pasa a calcularse a partir del estado estimado:

$$\begin{bmatrix} \phi_{ref}(t) \\ \theta_{ref}(t) \end{bmatrix} = -K\hat{\mathbf{x}}(t)$$

De esta forma, el controlador no actúa directamente sobre las medidas ruidosas del sensor visual, sino sobre una estimación filtrada y completa del estado del sistema.

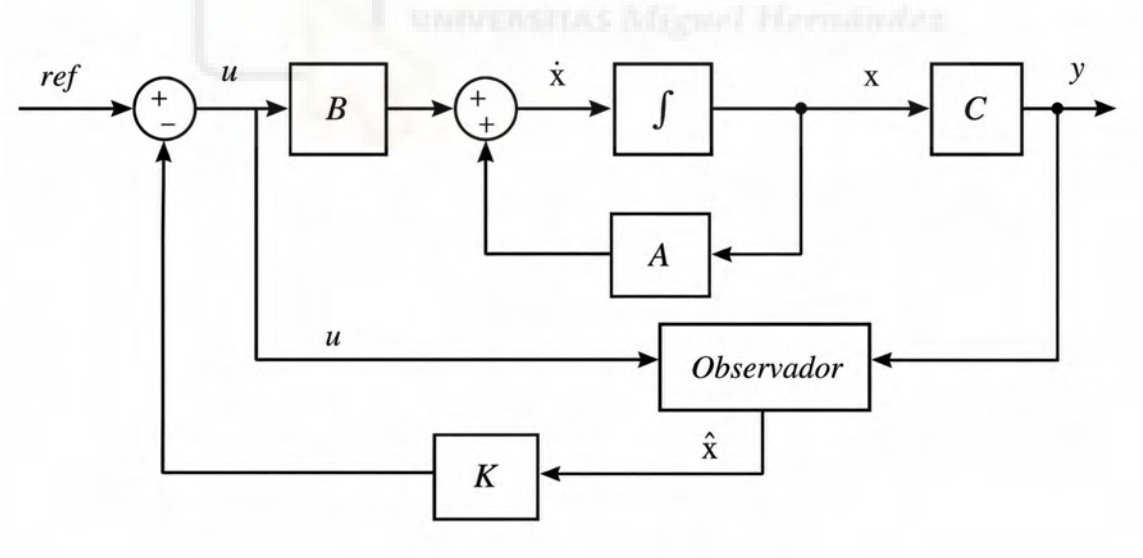


Figura 9.7: Diagrama de bloques de un sistema con realimentación y observador.

No obstante, el uso de un observador introduce una nueva **dependencia** respecto al modelo matemático empleado. Si el modelo utilizado por el filtro no representa adecuadamente la dinámica real de la planta, o si las covarianzas de ruido no están correctamente ajustadas, la estimación de estados puede degradarse y afectar al comportamiento del lazo cerrado.

9.6.2. Principio de separación y relación con LQG

La combinación de un regulador LQR con un filtro de Kalman como observador de estados se conoce habitualmente como control LQG (*Linear Quadratic Gaussian*). Esta estructura combina un diseño óptimo de control, basado en la minimización de un coste cuadrático, con una estimación óptima del estado bajo hipótesis lineales y ruido gaussiano.

En este trabajo, la denominación LQG puede emplearse para describir la estructura formada por el controlador LQR y el filtro de Kalman. No obstante, se mantiene la obtención separada de ambos elementos, ya que el regulador se ha diseñado en el presente capítulo y el observador fue introducido previamente en el capítulo 7.

La validez de esta combinación se apoya en el principio de separación. Según este principio, bajo las hipótesis de linealidad y observabilidad, el diseño de la ganancia de control K y el diseño del observador pueden realizarse de forma independiente [24], [29]. Por una parte, la matriz K se obtiene a partir del modelo equivalente (A_{eq}, B_{eq}), considerando como entrada las referencias de inclinación de la plataforma. Por otra parte, el filtro de Kalman se diseña a partir del modelo de la planta y de la matriz de medida asociada al sensor visual.

En el apartado 9.3 se comprobó que el sistema es observable a partir de las medidas de posición $x_m(t)$ e $y_m(t)$. Esta propiedad justificaba el uso del filtro de Kalman para estimar el vector completo de estados. Asimismo, también se verificó la controlabilidad del modelo equivalente empleado para el diseño del LQR, lo que permite aplicar la realimentación de estados sobre dicho modelo.

En consecuencia, el controlador y el observador pueden integrarse en el mismo lazo de control sin necesidad de rediseñar la matriz K. El controlador sigue empleando la ganancia obtenida, mostrada en la ecuación 9.22, pero sustituye el estado real por el estado estimado:

$$\mathbf{x}(t) \longrightarrow \hat{\mathbf{x}}(t)$$

Por tanto, la matriz dinámica del sistema controlado con observador no se analiza únicamente a partir de $A_{eq} - B_{eq}K$, como en el caso ideal, sino que debe considerarse también

la dinámica propia del estimador. Aun así, el principio de separación permite diseñar ambas partes de forma independiente y validar posteriormente el comportamiento global mediante simulación.

Debe destacarse que el filtro de Kalman fue inicialmente validado en lazo abierto en la sección 7.6.4. Al incorporarlo ahora en lazo cerrado, su estructura matemática no cambia, pero sí lo hace el régimen de funcionamiento del sistema. La planta ya no evoluciona libremente, sino bajo la acción del controlador LQR. Por ello, resulta necesario comprobar mediante simulación que la estimación sigue siendo adecuada cuando el observador forma parte del lazo de control.

9.6.3. Implementación en Simulink

La Figura 9.8 muestra el esquema implementado en Simulink para la simulación del controlador LQR con sensor visual simulado y observador de Kalman.

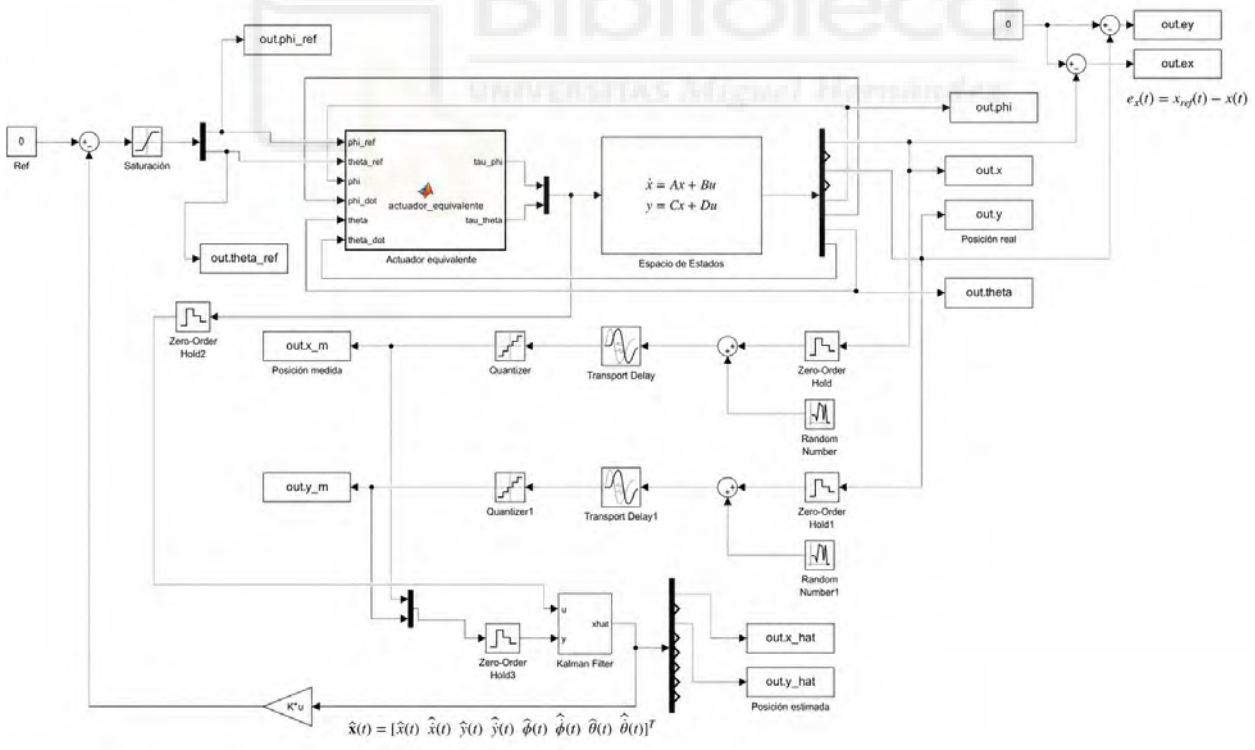


Figura 9.8: Esquema de Simulink del controlador LQR con sensor visual simulado y observador de Kalman.

En este esquema, la planta genera el estado real del sistema, pero dicho estado no se emplea directamente para el cálculo de la acción de control. En su lugar, las posiciones

reales de la bola se introducen en el bloque del sensor visual simulado, que genera las medidas $x_m(t)$ e $y_m(t)$. Estas señales incluyen las no idealidades, por lo que representan una medida más realista de la posición de la bola.

Las medidas proporcionadas por el sensor visual se introducen en el filtro de Kalman, junto con el modelo dinámico del sistema y las entradas aplicadas. El filtro estima el vector completo de estados $\hat{\mathbf{x}}(t)$, que posteriormente se realimenta mediante la matriz K. La señal generada por el controlador corresponde de nuevo a las referencias de inclinación (ec. 9.14), limitadas mediante el bloque de saturación y aplicadas al bloque de actuación equivalente, que calcula los pares equivalentes τ_ϕ y τ_θ introducidos en la planta de espacio de estados.

La diferencia fundamental respecto al esquema del apartado 9.5 es que el controlador ya no recibe el estado real, sino el estado estimado por el observador. El estado real se mantiene únicamente para el análisis de resultados y para comparar la respuesta de la planta con la estimación proporcionada por el filtro de Kalman.

Durante las simulaciones se registran las mismas variables utilizadas en el caso sin observador:

$$x(t), \quad y(t), \quad \phi(t), \quad \theta(t), \quad \phi_{ref}(t), \quad \theta_{ref}(t)$$

así como las medidas del sensor visual y las estimaciones del filtro, que se emplearán para analizar la calidad del sistema de medida y del observador:

$$x_m(t), \quad y_m(t), \quad \hat{x}(t), \quad \hat{y}(t)$$

Por último, se calculan los errores de estimación:

$$\begin{aligned} e_{\hat{x}}(t) &= x(t) - \hat{x}(t) \\ e_{\hat{y}}(t) &= y(t) - \hat{y}(t) \end{aligned} \tag{9.27}$$

Estos errores permiten cuantificar la diferencia entre las posiciones reales a la salida del espacio de estados y las posiciones estimadas por el observador en lazo cerrado.

Al igual que en el apartado anterior, las simulaciones se realizan empleando los mismos escenarios de prueba definidos en la Tabla 8.2 del capítulo 8. De este modo, los resultados con observador pueden compararse directamente con el caso ideal de estado completo.

9.6.4. Resultados de simulación con observador

Una vez integrado el sensor visual simulado y el observador de Kalman en el lazo de control, se repiten los ensayos definidos anteriormente para evaluar el comportamiento del controlador LQR en una situación más realista.

Los resultados obtenidos se resumen en la Tabla 9.4 y la Tabla 9.5. En ellas se observa que el sistema consigue estabilizar la bola en todos los ensayos, con tiempos de establecimiento comprendidos entre 1.400 s y 2.756 s.

Tabla 9.4: Resultados de posición obtenidos con controlador LQR y observador de Kalman.

Ensayo	t_s [s]	**máx** $\lvert x \rvert$ [m]	**máx** $\lvert y \rvert$ [m]	x_f [m]	y_f [m]
1	1.400	0.0300	0.0004	0.00014690	0.00010971
2	1.410	0.0003	0.0300	-0.00012246	0.00039640
3	1.403	0.0300	0.0300	0.00014690	-0.00004823
4	2.405	0.0293	0.0004	-0.00036093	0.00010971
5	2.051	0.0320	0.0263	-0.00015141	0.00023029
6	2.756	0.1064	0.0600	-0.00024149	-0.00038746

Tabla 9.5: Referencias de inclinación y orientación máxima de la plataforma con controlador LQR y observador de Kalman.

Ensayo	**máx** $\lvert \phi_{ref} \rvert$ [°]	**máx** $\lvert \theta_{ref} \rvert$ [°]	**máx** $\lvert \phi \rvert$ [°]	**máx** $\lvert \theta \rvert$ [°]
1	10.00	0.82	6.06	0.14
2	0.83	10.00	0.19	6.04
3	10.00	10.00	6.06	6.01
4	3.77	0.82	3.25	0.14
5	10.00	9.86	5.23	4.71
6	10.00	10.00	9.77	8.09

En primer lugar, puede apreciarse que los ensayos 1 y 2 presentan un comportamiento prácticamente simétrico. En el ensayo 1 la desviación principal se produce en el eje X, alcanzándose x_{max} = 0,0300 m, mientras que el desplazamiento en Y se mantiene muy reducido, con y_{max} = 0,0004 m. En el ensayo 2 ocurre lo contrario: el desplazamiento dominante aparece en el eje Y, con y_{max} = 0,0300 m, mientras que x_{max} = 0,0003 m.

Esta simetría también se refleja en las referencias de inclinación: en el primer caso se alcanza $\phi_{ref,max} = 10°$, mientras que en el segundo se alcanza $\theta_{ref,max} = 10°$. Este comportamiento es coherente con la simetría del modelo lineal y con la estructura de la matriz K obtenida mediante LQR.

El ensayo 3, en el que se excitan ambos ejes de forma simultánea, mantiene un tiempo de establecimiento similar al de los ensayos 1 y 2, con $t_s = 1{,}403$ s. Además, las referencias de inclinación alcanzan el límite de saturación en ambos ejes:

$$\phi_{ref,max} = 10°, \qquad \theta_{ref,max} = 10°$$

mientras que las inclinaciones reales máximas se sitúan en torno a 6°. Esto indica que, aunque el controlador solicita inicialmente la máxima inclinación permitida, la dinámica del actuador equivalente suaviza la respuesta real de la plataforma.

El ensayo 4 presenta un tiempo de establecimiento mayor, $t_s = 2{,}405$ s, a pesar de que el desplazamiento máximo de la bola es inferior al de otros ensayos. En este caso, la referencia máxima de inclinación en el eje principal es de 3,77°, significativamente menor que el límite de saturación. Por tanto, la respuesta resulta menos agresiva y el retorno al origen se produce de forma más lenta. Este resultado pone de manifiesto el compromiso característico del diseño LQR entre rapidez de respuesta y esfuerzo de control.

El ensayo 5 vuelve a excitar ambos ejes y presenta un tiempo de establecimiento de 2.051 s. Las referencias de inclinación alcanzan valores elevados en ambos ejes, especialmente $\phi_{ref,max} = 10°$ y $\theta_{ref,max} = 9{,}86°$, aunque las inclinaciones reales máximas permanecen por debajo de dichos valores. Esto confirma nuevamente que el bloque de actuación equivalente introduce una dinámica que impide que la plataforma siga de manera instantánea las referencias generadas por el controlador.

El ensayo 6 constituye el caso más exigente. Se alcanzan los mayores desplazamientos de la bola:

$$x_{max} = 0{,}1064 \text{ m}, \qquad y_{max} = 0{,}0600 \text{ m}$$

y el mayor tiempo de establecimiento: $t_s = 2{,}756$ s. También se obtienen las mayores

inclinaciones reales de la plataforma:

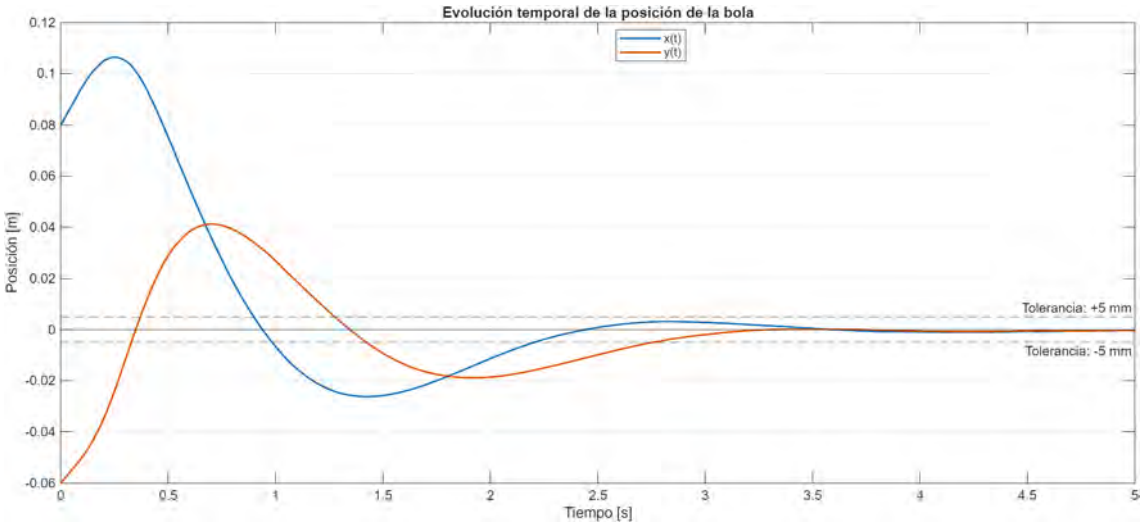
$$\phi_{max} = 9{,}77^\circ, \qquad \theta_{max} = 8{,}09^\circ$$

valores próximos al límite de 10°, pero sin superarlo. Este resultado indica que el controlador aprovecha prácticamente todo el margen de actuación disponible para estabilizar la bola en el escenario más desfavorable.

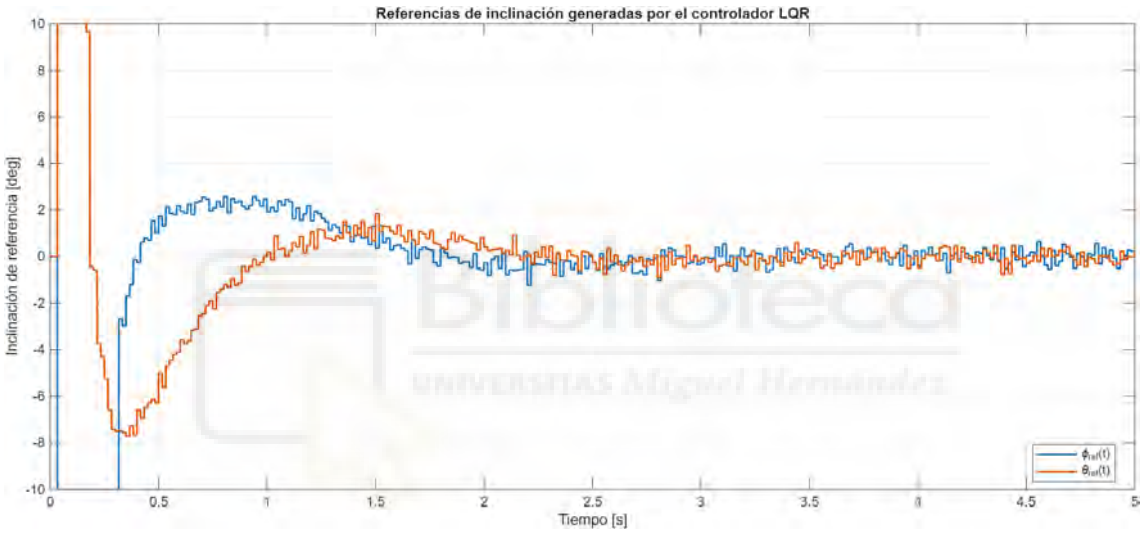
La Figura 9.9(a) muestra la evolución temporal de las posiciones de la bola para el ensayo 6. Se observa que $x(t)$ e $y(t)$ presentan un transitorio inicial de mayor amplitud, seguido de una convergencia progresiva hacia el origen. Ambas señales entran finalmente en la banda de tolerancia de ±5 mm, lo que coincide con el tiempo de establecimiento recogido en la tabla.

La Figura 9.9(b) representa las referencias de inclinación generadas por el controlador. En los primeros instantes de la simulación, ambas referencias alcanzan la saturación de 10°, lo que confirma que el controlador solicita la máxima actuación disponible para corregir la desviación inicial. Posteriormente, las referencias disminuyen y oscilan en torno a cero con pequeñas variaciones. Estas irregularidades son consecuencia de que la realimentación se realiza a partir del estado estimado, el cual se obtiene a partir de medidas afectadas por ruido, cuantización, retardo y frecuencia de muestreo limitada.

A pesar de estas pequeñas oscilaciones, la orientación real de la plataforma permanece dentro de los límites admisibles.



(a) Posición de la bola.



(b) Referencias de inclinación generadas por el controlador.

Figura 9.9: Respuesta temporal del sistema con observador para el ensayo 6.

La Figura 9.10 muestra la comparación entre la posición real $x(t)$, la posición medida $x_m(t)$ y la posición estimada $\hat{x}(t)$ para el ensayo 6. La señal medida sigue la tendencia de la posición real, pero presenta pequeñas discontinuidades e irregularidades debidas a las no idealidades del sensor visual simulado. Por su parte, la estimación proporcionada por el filtro de Kalman sigue adecuadamente la evolución de la señal real, suavizando parte del ruido de medida y manteniendo un desfase reducido.

Finalmente, la Figura 9.11 representa los errores de estimación del filtro de Kalman, estudiados en la ecuación 9.27.

Durante el transitorio inicial aparecen los errores de mayor amplitud, debido a que el

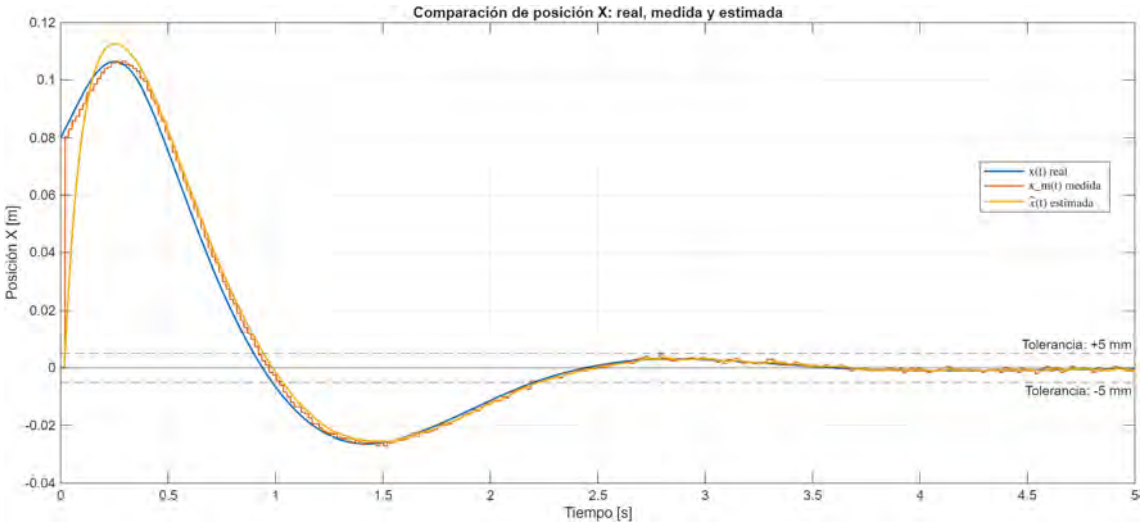


Figura 9.10: Comparación entre posición real, medida y estimada en el eje X para el ensayo 6.

filtro parte de una estimación inicial nula y debe corregirla a partir de las primeras medidas del sensor visual. Sin embargo, estos errores disminuyen progresivamente y tienden a valores próximos a cero conforme la bola se aproxima al punto de equilibrio. Este comportamiento confirma que el observador proporciona una estimación adecuada del estado en lazo cerrado.

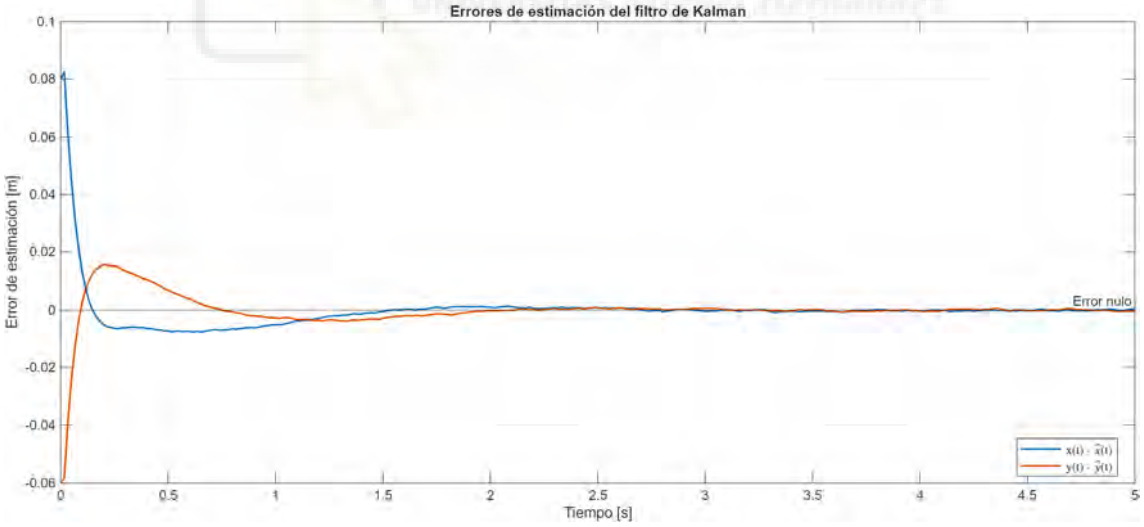


Figura 9.11: Errores de estimación del filtro de Kalman en lazo cerrado para el ensayo 6.

En comparación con la simulación sin observador de la sección 9.5, se aprecia un aumento general de los tiempos de establecimiento. En el caso ideal, el controlador disponía directamente del vector completo de estados, sin ruido, retardo ni cuantización. En cambio, en esta configuración el controlador actúa sobre el estado estimado $\hat{\mathbf{x}}(t)$, construido a partir de medidas visuales no ideales. Por ello, la respuesta es más lenta y presenta pequeñas

oscilaciones en las referencias de inclinación.

Por ejemplo, en el ensayo 6 el tiempo de establecimiento aumenta respecto al caso sin observador, pasando de un valor del orden de 1.256 s a 2.756 s. Además, los valores finales x_f e y_f, que en el caso ideal eran prácticamente nulos a nivel numérico, pasan ahora a ser del orden de 10^{-4} m. No obstante, estos errores finales siguen siendo inferiores a 0.5 mm, por lo que permanecen ampliamente dentro de la banda de tolerancia establecida. Esta magnitud resulta coherente con la configuración del sensor visual simulado, ya que coincide con el orden de resolución espacial o intervalo de cuantización $\Delta = 5 \cdot 10^{-4}$ m definido en el capítulo 7.6.2. Por tanto, las pequeñas desviaciones finales observadas no deben atribuirse a un error estacionario significativo del controlador, sino principalmente a las no idealidades introducidas por el sistema de medida y al proceso de estimación del estado.

En conjunto, los resultados muestran que la incorporación del sensor visual simulado y del observador de Kalman degrada la respuesta respecto al caso ideal, como era esperable, pero no compromete la estabilidad del sistema. El controlador LQR sigue siendo capaz de estabilizar la bola en el origen para todos los escenarios de prueba, y el filtro de Kalman proporciona una estimación suficientemente precisa para cerrar el lazo de control utilizando únicamente medidas de posición.

De forma análoga al caso del controlador PD, la estrategia LQR con observador de Kalman también fue evaluada frente a **perturbaciones puntuales** introducidas mediante el bloque *Pulse Generator*. La respuesta obtenida confirmó la capacidad del controlador para recuperar la posición de equilibrio tras desviaciones externas de corta duración.

9.7. Análisis crítico de las estrategias de control

A partir de los resultados obtenidos en los capítulos 8 y 9, puede realizarse una valoración conjunta de las estrategias de control estudiadas. No obstante, esta comparación debe interpretarse con cautela, ya que las arquitecturas consideradas no son completamente equivalentes desde el punto de vista de la medida y del procesamiento de señales.

En el caso ideal, el controlador LQR con realimentación del estado completo presenta

una respuesta muy favorable, con tiempos de establecimiento reducidos y error final prácticamente nulo. Este comportamiento es coherente con el hecho de que el controlador dispone directamente de todas las variables del vector de estados y puede calcular la acción de control a partir de una descripción completa del sistema. Sin embargo, al incorporar el sensor visual simulado y sustituir el estado real por el estado estimado mediante el filtro de Kalman, la respuesta se degrada respecto al caso ideal. Esta degradación es esperable, ya que el controlador deja de trabajar con información exacta y continua del sistema, pasando a depender de medidas discretas, ruidosas, cuantizadas y retardadas.

Por el contrario, el controlador PD presenta una estructura más directa. Su acción de control se calcula a partir del error de posición y de una acción derivativa filtrada, por lo que está más ligado a las variables que proporciona directamente el sensor visual. Esto puede explicar que, en las simulaciones con sensor visual simulado, la degradación respecto al caso ideal sea menor que en el caso del LQR con observador. En otras palabras, el LQR aprovecha mejor la información del modelo cuando el estado completo está disponible, pero su rendimiento en una situación más realista queda condicionado por la calidad de la estimación de estados.

Respecto al **tiempo de establecimiento**, el LQR sin observador ofrece mejores resultados que el controlador PD en condiciones ideales. No obstante, cuando se introduce el sensor visual simulado y el filtro de Kalman, los tiempos de establecimiento aumentan. Este incremento se debe a la dinámica adicional del observador, a la discretización del sistema de medida y a las no idealidades del sensor. Por tanto, la ventaja observada en el caso ideal no se mantiene necesariamente cuando el lazo se cierra a partir del estado estimado.

En cuanto al **error final**, ambas estrategias consiguen estabilizar la bola dentro de la banda de tolerancia establecida. En el caso ideal, el LQR obtiene errores finales prácticamente nulos a nivel numérico. Con sensor visual simulado, los valores finales pasan a ser del orden de 10^{-4} m, una magnitud coherente con la resolución espacial y la cuantización del sistema de visión. Por tanto, estas pequeñas desviaciones no deben interpretarse como errores estacionarios significativos del controlador, sino como una consecuencia de las limitaciones del sistema de medida.

En relación con el **esfuerzo de control**, el LQR permite introducir de forma explícita una

penalización de la acción de control mediante la matriz R. Esta es una ventaja importante desde el punto de vista metodológico, ya que permite ajustar el compromiso entre rapidez de respuesta y esfuerzo de actuación. Sin embargo, los resultados obtenidos con la sintonización empleada muestran que el LQR con observador puede llegar a solicitar inclinaciones más elevadas que el controlador PD en determinados ensayos. Esto indica que las matrices Q y R, ajustadas inicialmente para el caso ideal con estado completo, no tienen por qué proporcionar el mejor compromiso cuando el controlador trabaja con estados estimados. Una **mejora futura** consistiría en reajustar dichas matrices considerando directamente el sensor visual y el observador de Kalman en el lazo.

En lo relativo a la **sensibilidad al sensor visual**, el controlador PD puede verse afectado por el ruido de medida, especialmente en la acción derivativa. Por ello, requiere filtrado para evitar que pequeñas variaciones de la medida generen oscilaciones en la señal de control. No obstante, su estructura es sencilla y depende de un conjunto reducido de variables. En cambio, el LQR necesita disponer del vector completo de estados, por lo que requiere un observador cuando solo se miden las posiciones de la bola. El filtro de Kalman permite estimar las variables no medidas y evita la derivación numérica directa de señales ruidosas, pero introduce una dependencia adicional respecto al modelo matemático, las covarianzas de ruido y la frecuencia de muestreo del sensor.

La Tabla 9.6 resume las principales ventajas e inconvenientes de ambas estrategias de control analizadas.

En conjunto, los resultados muestran que el controlador PD constituye una solución sencilla y eficaz para la estabilización de la bola, especialmente cuando se trabaja con medidas de posición procedentes del sensor visual. Por su parte, la estrategia LQR/LQG proporciona un marco de diseño más general y sistemático, especialmente atractivo desde el punto de vista del control moderno, pero su rendimiento práctico depende en mayor medida de la calidad del modelo y del observador.

Tabla 9.6: Comparación cualitativa entre el controlador PD y la estrategia LQR/LQG.

Estrategia	Ventajas	Inconvenientes
Control PD	- Estructura sencilla e intuitiva. - Implementación directa mediante diagramas de bloques. - Bajo coste computacional. - Puede ajustarse de forma empírica sin requerir un modelo completo de la planta. - Ofrece una respuesta robusta con el sensor visual simulado en el sistema estudiado.	- Diseño descentralizado por ejes que ignora los acoplamientos interaxiales. - No aprovecha explícitamente el modelo dinámico multivariable completo. - La acción derivativa es sensible al ruido de medida, requiriendo filtros que añaden retardo. - El compromiso entre rapidez y esfuerzo de control se ajusta de forma menos sistemática.
Control LQR/LQG	- Diseño sistemático basado en el modelo en espacio de estados. - Permite ponderar estados y esfuerzo de control mediante Q y R. - Trata de forma natural sistemas multivariables y acoplados. - El filtro de Kalman permite estimar estados no medidos de forma sistemática bajo las hipótesis lineales y gaussianas adoptadas. - Evita la derivación numérica directa de señales de posición ruidosas.	- Mayor dependencia de la precisión del modelo matemático de la planta. - Requiere un observador de estados si no se dispone del estado completo. - Mayor complejidad matemática y coste computacional en tiempo real. - La respuesta global depende críticamente de la sintonización de cuatro matrices (Q, R, Q_k y R_k). - Puede degradarse si la estimación de estados introduce retardo o errores transitorios.

Por tanto, no puede concluirse de forma absoluta que una estrategia sea superior a la otra en todos los escenarios. El LQR presenta mejores prestaciones en el caso ideal con estado completo, mientras que el PD muestra una mayor simplicidad y una menor dependencia de la estimación de estados en las simulaciones con sensor visual. Esta conclusión resulta especialmente relevante de cara a una posible **implementación física**, donde la elección del controlador no dependería únicamente de la respuesta nominal, sino también de la complejidad de implementación, la capacidad de cómputo disponible y la calidad del sistema de medida.

9.8. Conclusiones del capítulo

En este capítulo se ha desarrollado una estrategia de control basada en realimentación de estados para la estabilización de la bola sobre la plataforma. A diferencia del controlador

PD estudiado en el capítulo anterior, esta metodología utiliza directamente la representación en espacio de estados del sistema, permitiendo formular el problema de control desde una perspectiva multivariable.

En primer lugar, se ha comprobado que el modelo lineal empleado es controlable y observable, tanto desde el punto de vista del estado como desde las salidas principales de posición. Estos resultados justifican el diseño de una matriz de realimentación K y el uso posterior de un observador para estimar las variables no medidas directamente.

La matriz de realimentación se ha obtenido mediante el método LQR, empleando un modelo equivalente que incorpora el bloque de actuación equivalente. Esta formulación permite que el controlador genere referencias de inclinación ϕ_{ref} y θ_{ref}, coherentes con la implementación realizada en Simulink, en lugar de actuar directamente sobre los pares de la plataforma.

Las simulaciones sin observador han mostrado que, cuando el vector completo de estados se encuentra disponible, el controlador LQR estabiliza la bola de forma rápida y con error final prácticamente nulo. Este resultado confirma la validez del diseño en condiciones ideales y pone de manifiesto el potencial de la realimentación de estados cuando se dispone de información completa del sistema.

Posteriormente, se ha incorporado el sensor visual simulado y el filtro de Kalman como observador de estados. En esta configuración, el controlador deja de utilizar el estado real y pasa a trabajar con el estado estimado $\hat{\mathbf{x}}(t)$. Los resultados muestran que el sistema sigue siendo estable y converge al entorno del origen, aunque con una degradación de la respuesta respecto al caso ideal. Esta degradación es coherente con la presencia de ruido, retardo, cuantización, muestreo discreto y errores transitorios de estimación.

Finalmente, el análisis crítico realizado permite concluir que la estrategia LQR/LQG proporciona un marco de diseño más sistemático y general que el controlador PD, pero también presenta una mayor dependencia del modelo matemático y del sistema de estimación. En cambio, el controlador PD destaca por su sencillez y por su menor complejidad de implementación, aunque no aprovecha de forma explícita el modelo multivariable completo.

10. CONCLUSIONES Y LÍNEAS DE TRABAJO FUTURAS

10.1. Conclusiones generales del trabajo

El presente Trabajo Fin de Grado ha abordado el modelado, simulación y control de una plataforma tipo Stewart de tres grados de libertad cuyo objetivo es la estabilización de una bola sobre su superficie. El desarrollo se ha realizado íntegramente en un **entorno de simulación**, empleando principalmente MATLAB/Simulink como herramienta de modelado, implementación y análisis.

Dado el enfoque de simulación adoptado, uno de los aspectos fundamentales del trabajo ha sido la obtención de un modelo que represente adecuadamente la dinámica del sistema real. Para ello, se han estudiado dos metodologías complementarias: un modelo matemático simplificado en **espacio de estados** y un **modelo multicuerpo** implementado mediante Simscape Multibody. Puesto que todo modelo constituye una representación aproximada de la realidad, resultaba necesario comprobar si las hipótesis simplificadoras del modelo matemático lineal permitían describir con suficiente fidelidad el comportamiento de la plataforma en el rango de operación considerado.

Con este objetivo, en el capítulo 6 se compararon ambos modelos, excitándolos simultáneamente mediante una serie de entradas de referencia diseñadas para evaluar su respuesta dinámica. Los resultados obtenidos mostraron que, para pequeñas inclinaciones y operación en torno al punto de equilibrio, las respuestas del modelo en espacio de estados y del modelo multicuerpo eran suficientemente similares. De este modo, se concluyó que el modelo en espacio de estados era adecuado para el desarrollo de las estrategias de control planteadas en los capítulos posteriores.

Otra diferencia clave entre un entorno simulado ideal y una posible plataforma real reside en el tratamiento de la información proporcionada por los sensores. En sistemas *Ball and Plate* implementados mediante una cámara digital, la medida de la posición de la bola puede verse afectada por no idealidades asociadas al proceso de adquisición y tratamiento de imagen. Con el fin de aproximar el modelo a una situación más realista, en el capítulo 7 se incorporaron dichas no idealidades mediante la implementación de un **sensor visual**

simulado en Simulink.

Asimismo, aunque en un entorno de simulación es posible acceder directamente a todas las variables de interés, en una implementación física el número de sensores disponibles suele ser limitado. En este trabajo se considera que el sistema de medida proporciona únicamente la posición de la bola sobre la plataforma. Por ello, se ha incorporado un **filtro de Kalman** como estimador de estados, permitiendo reconstruir el vector completo de estados a partir de un conjunto reducido de medidas. Esta realimentación no ideal se ha utilizado posteriormente para analizar de forma más realista el comportamiento de las estrategias de control propuestas.

En concreto, se han estudiado dos enfoques de control principales. En primer lugar, se ha diseñado un **controlador PD** con derivada filtrada, implementado de forma descentralizada mediante un regulador independiente para cada eje. Esta estrategia ha demostrado ser sencilla, eficaz y adecuada bajo las hipótesis de desacoplamiento aproximado del modelo lineal. En segundo lugar, se ha desarrollado una estrategia de control por **realimentación de estados mediante LQR**, aprovechando la representación interna del sistema y permitiendo un diseño más sistemático desde el punto de vista del control moderno.

Los resultados obtenidos muestran que ambas estrategias son capaces de estabilizar la bola en torno al origen para los escenarios de prueba considerados. En condiciones ideales, la realimentación de estados mediante LQR ofrece una respuesta rápida y con error final prácticamente nulo, al disponer directamente del vector completo de estados. Sin embargo, al incorporar el sensor visual simulado y el observador de Kalman, la respuesta se degrada respecto al caso ideal, debido a la presencia de ruido, retardo, cuantización, muestreo discreto y errores transitorios de estimación.

El análisis conjunto de los controladores permite concluir que **no existe una estrategia claramente superior** en todos los escenarios. El controlador PD destaca por su simplicidad, bajo coste de implementación y menor dependencia del modelo matemático completo. Por el contrario, el controlador LQR/LQG proporciona un marco de diseño más general y sistemático, especialmente adecuado para sistemas multivariables, aunque su rendimiento práctico depende en mayor medida de la calidad del modelo, del sistema de medida y de la estimación de estados.

En conjunto, el trabajo realizado demuestra la viabilidad de emplear un entorno de simulación integral para estudiar el comportamiento de una plataforma paralela aplicada a un problema de estabilización tipo *Ball and Plate*. La combinación de modelado analítico, simulación multicuerpo, sensores virtuales y estrategias de control ha permitido analizar el sistema de forma progresiva, desde condiciones ideales hasta escenarios más próximos a una posible implementación física.

10.2. Cumplimiento de objetivos

Una vez presentadas las conclusiones generales del trabajo, se revisa a continuación el grado de cumplimiento de los objetivos definidos inicialmente en la sección 1.3. Esta revisión permite comprobar que el desarrollo realizado se ajusta al alcance planteado al inicio del proyecto.

En relación con el **diseño mecánico**, se ha definido una plataforma paralela basada en una arquitectura 3-RRS, estableciendo sus parámetros geométricos principales y obteniendo el modelo de cinemática inversa necesario para relacionar las inclinaciones deseadas del plato con las variables articulares de los actuadores. Con ello se cumplen los objetivos asociados al diseño conceptual de la plataforma y a la verificación cinemática del mecanismo.

Respecto al **modelado del sistema**, se ha obtenido una representación dinámica simplificada del conjunto bola-plataforma en espacio de estados. Además, se ha desarrollado un modelo CAD de la plataforma y se ha exportado a Simscape Multibody, permitiendo comparar el modelo analítico con una representación multicuerpo más detallada. Los resultados de dicha comparación han permitido validar el uso del modelo linealizado dentro del rango de operación considerado, cumpliéndose así los objetivos relativos al modelado matemático, al prototipado virtual y a la validación entre modelos.

En cuanto al **sistema de medida**, se han implementado tanto un sensor ideal como un sensor visual simulado. Este último incorpora las principales no idealidades asociadas a una posible cámara real, incluyendo ruido, retardo, cuantización y frecuencia de muestreo finita. Asimismo, se ha integrado un filtro de Kalman para estimar el vector completo de estados a partir de medidas parciales de posición, satisfaciendo los objetivos relacionados

con la realimentación no ideal y la estimación de estados.

Finalmente, se han diseñado, implementado y evaluado distintas **estrategias de control**, abarcando tanto enfoques clásicos, mediante un controlador PD con derivada filtrada, como técnicas de control moderno basadas en la realimentación de estados. Esta última estrategia se ha ampliado posteriormente mediante la incorporación de un observador de Kalman, con el fin de estimar las variables no medidas directamente. Ambos enfoques se han analizado bajo distintos escenarios de simulación, considerando indicadores como el tiempo de establecimiento, el error final, el esfuerzo de control y la sensibilidad frente a las no idealidades del sistema de medida.

Por tanto, puede afirmarse que los **objetivos principales del trabajo** se han cumplido de forma satisfactoria. Se ha logrado estabilizar la bola en el centro de la superficie de la plataforma para los escenarios de simulación considerados, desarrollándose todo el estudio en un entorno de simulación basado en MATLAB/Simulink.

10.3. Líneas futuras de desarrollo

A partir del trabajo realizado, pueden plantearse distintas líneas futuras de desarrollo orientadas a ampliar el alcance del proyecto. Estas líneas pueden agruparse en dos bloques principales: por una parte, aquellas que mantienen el enfoque íntegramente basado en simulación y, por otra, aquellas orientadas a una futura implementación física de la plataforma.

En primer lugar, dentro del **entorno de simulación,** una extensión natural del trabajo consistiría en abordar el problema de seguimiento de trayectorias. En el presente proyecto el objetivo de control se ha limitado a la regulación de la bola en torno al origen, con referencias nulas de posición. Para ampliar las capacidades del sistema, podría diseñarse una estructura de control con precompensador de referencia, acción integral o servosistema aumentado, permitiendo que la bola siguiera trayectorias predefinidas sobre la superficie de la plataforma (véase sección 9.2). Esta línea requeriría analizar nuevamente las restricciones de actuación, la influencia del sensor visual y el comportamiento del sistema ante referencias variables en el tiempo.

Otra posible línea de investigación dentro del entorno simulado sería el estudio de **otras estrategias de control**. Además de los controladores PD y LQR/LQG desarrollados en este trabajo, podrían analizarse técnicas como control difuso, control robusto, control predictivo basado en modelo o control no lineal. Estas estrategias permitirían evaluar si es posible mejorar la respuesta del sistema ante no linealidades, incertidumbres paramétricas, saturaciones de actuación o perturbaciones externas.

También resultaría de interés reducir progresivamente algunas de las **hipótesis simplificadoras** adoptadas durante el modelado matemático. En particular, podrían incorporarse efectos como rozamientos no lineales, deslizamiento entre la bola y la plataforma, holguras mecánicas, saturaciones reales de los actuadores, flexibilidad estructural o variaciones de la altura de la plataforma. De este modo, se obtendría un modelo más próximo al comportamiento real del sistema, permitiendo estudiar con mayor precisión los límites de validez del modelo lineal empleado para el diseño de controladores.

En segundo lugar, una línea futura de especial relevancia sería la **implementación física del prototipo**. Para ello sería necesario fabricar la estructura mecánica de la plataforma, por ejemplo mediante impresión 3D o mecanizado de componentes, seleccionar los servomotores adecuados e integrar la electrónica de potencia y control necesaria. Esta etapa permitiría validar experimentalmente la cinemática inversa, comprobar la respuesta real del mecanismo y analizar efectos no considerados en simulación, como errores de montaje, tolerancias de fabricación, rozamientos, vibraciones o deformaciones estructurales.

La implementación física requeriría además sustituir el sensor visual simulado por un **sistema de visión real**. Esto implicaría seleccionar una cámara adecuada, implementar algoritmos de detección de la bola y realizar la conversión entre coordenadas de imagen y coordenadas físicas de la plataforma. A las no idealidades ya modeladas en simulación habría que añadir otros efectos propios del entorno real, tales como variaciones de iluminación, reflejos, oclusiones parciales, distorsión óptica y errores de calibración de la cámara.

Asimismo, sería necesario llevar a cabo una **calibración experimental** del sistema completo. Esta calibración incluiría la identificación de parámetros geométricos reales, la relación entre ángulos de servomotor e inclinación efectiva del plato, la conversión entre

píxeles y milímetros, la localización del origen de coordenadas y la estimación de retardos reales del sistema de adquisición y procesamiento. Estos procedimientos serían esenciales para reducir discrepancias entre el modelo utilizado en simulación y el comportamiento del prototipo físico.

Finalmente, de cara a una implementación real, resultaría imprescindible estudiar la **discretización de los controladores**. En el presente trabajo se ha desarrollado el control principalmente en un entorno de simulación continuo o mixto. Sin embargo, en un prototipo físico los algoritmos deberían ejecutarse en un microcontrolador, PLC o sistema embebido con un periodo de muestreo definido. Por tanto, sería necesario analizar la influencia de la discretización, los retardos computacionales, la frecuencia de actualización y las limitaciones de cálculo sobre la estabilidad y las prestaciones del sistema.

En conjunto, estas líneas futuras permitirían avanzar desde el entorno de simulación desarrollado en este trabajo hacia una plataforma experimental completa. Igualmente, ofrecerían la posibilidad de profundizar en el estudio del sistema mediante modelos más realistas, estrategias de control más avanzadas y una caracterización más precisa de los procesos de medida y actuación.

A. CÓDIGO FUENTE

A.1. Implementación del espacio de estados

A continuación, se muestra la implementación del modelo dinámico lineal conjunto plato-bola en el espacio de estados, obtenido en la sección 4.3.4.

```
function sys = state_space_completa(params)
% STATE_SPACE_BALL_PLATE
% Construye el modelo lineal conjunto plato-bola en espacio de
   estados.
% Modelo linealizado bajo pequeñas inclinaciones y rodadura sin
   deslizamiento.
% Se asume desacoplo entre ejes X e Y.
%
% Estado:
%   x = [x; x_dot; y; y_dot; phi; phi_dot; theta; theta_dot]
%
% Entrada:
%   u = [tau_phi; tau_theta]
%
% Salida:
%   y = [x; x_dot; y; y_dot; phi; phi_dot; theta; theta_dot]
%
% Parámetros requeridos:
%   params.g
%   params.Jphi
%   params.Jtheta
%   params.bphi
%   params.btheta

arguments
    params struct
end

```

```
% Validación de parámetros
requiredFields = {'g','Jphi','Jtheta','bphi','btheta'};

for k = 1:numel(requiredFields)
    assert(isfield(params,requiredFields{k}), ...
    'Falta el parámetro "%s" en la estructura params.',
       requiredFields{k});
end

g      = params.g;
Jphi   = params.Jphi;
Jtheta = params.Jtheta;
bphi   = params.bphi;
btheta = params.btheta;

assert(Jphi > 0 && Jtheta > 0, 'Las inercias deben ser
   positivas.');

% Constante de acoplamiento bola-plataforma
% Esfera maciza homogénea con rodadura sin deslizamiento
Kb = (5/7) * g;

% Matriz A
% - Estados 1-4: bola
% - Estados 5-8: plataforma
A = [ ...
0   1   0   0   0              0       0              0;
0   0   0   0   Kb             0       0              0;
0   0   0   1   0              0       0              0;
0   0   0   0   0              0       Kb             0;
0   0   0   0   0              1       0              0;
0   0   0   0   0      -bphi/Jphi      0              0;
0   0   0   0   0              0       0              1;
0   0   0   0   0              0       0      -btheta/Jtheta
```

```
];

% Matriz B
B = [ ...
0           0;
0           0;
0           0;
0           0;
0           0;
1/Jphi      0;
0           0;
0           1/Jtheta
];

% Matriz C: se sacan todos los estados
C = eye(8);

% Matriz D
D = zeros(8,2);

% Construcción del sistema
sys = ss(A,B,C,D);

% Nombres descriptivos
sys.StateName =
   {'x','x_dot','y','y_dot','phi','phi_dot','theta','theta_dot'};
sys.InputName = {'tau_phi','tau_theta'};
sys.OutputName =
   {'x','x_dot','y','y_dot','phi','phi_dot','theta','theta_dot'};

end
```

Listing 1: Implementación del modelo conjunto plato-bola en espacio de estados.

A.2. Script de inicialización de las matrices del espacio de estados

Seguidamente se adjunta un script en Matlab para almacenar las variables en el *Workspace* y poder usarlas en el bloque del *State-Space* en el diagrama de Simulink.

```
% INIT_STATE_SPACE_COMPLETA
% Script de inicialización del modelo en espacio de estados.
% Define los parámetros dinámicos, construye el sistema y
   exporta las matrices A, B, C y D al workspace para su
   uso en Simulink.

clear; clc;

%% Parámetros dinámicos
params.g      = 9.81;         % gravedad [m/s^2]
params.Jphi   = 1.3413e-3;    % inercia eje phi [kg*m^2]
params.Jtheta = 1.3413e-3;    % inercia eje theta [kg*m^2]
params.bphi   = 1e-2;         % amortiguamiento eje phi
   [N*m*s/rad]
params.btheta = 1e-2;         % amortiguamiento eje theta
   [N*m*s/rad]

%% Construcción del modelo
sys_ball_plate = state_space_completa(params);

%% Extracción de matrices
[A,B,C,D] = ssdata(sys_ball_plate);

%% Variables disponibles para Simulink
assignin('base','A',A);
assignin('base','B',B);
assignin('base','C',C);
assignin('base','D',D);
assignin('base','sys_ball_plate',sys_ball_plate);
assignin('base','params',params);
```

```
27
28      disp('Modelo en espacio de estados inicializado
            correctamente.');
```

Listing 2: Script de inicialización del modelo en espacio de estados.

A.3. Implementación del algoritmo de la actuación equivalente

En este apartado se muestra el contenido del bloque *MATLAB Function* que define la actuación simplificada de los motores en Simulink.

```
1  function [tau_phi, tau_theta] = actuador_equivalente(phi_ref,
       theta_ref, phi, phi_dot, theta, theta_dot)
2  % ACTUADOR_EQUIVALENTE
3  % Modelo de actuación equivalente para la plataforma.
4  %
5  % Esta función implementa un servo de orientación que genera
       los pares equivalentes necesarios (tau_phi, tau_theta) para
       que la plataforma siga las referencias de inclinación
       (phi_ref, theta_ref).
6  %
7  % ENTRADAS:
8  % phi_ref   -> referencia de inclinación asociada a la dinámica
       en X [rad]
9  % theta_ref -> referencia de inclinación asociada a la dinámica
       en Y [rad]
10 % phi       -> inclinación actual del plato para dinámica en X
       [rad]
11 % phi_dot   -> velocidad angular asociada a phi [rad/s]
12 % theta     -> inclinación actual del plato para dinámica en Y
       [rad]
13 % theta_dot -> velocidad angular asociada a theta [rad/s]
14 %
15 % SALIDAS:
16 % tau_phi   -> par equivalente asociado al ángulo phi [N.m]
```

```
% tau_theta -> par equivalente asociado al ángulo theta [N.m]

%
% El modelo aproxima el seguimiento de las referencias de
   orientación mediante una dinámica de segundo orden, sin
   modelar de forma explícita la dinámica interna de los
   actuadores ni sus saturaciones.

% --------------------------------
% PARÁMETROS DINÁMICOS DEL PLATO
% --------------------------------
% Momentos de inercia del plato respecto a los ejes X e Y
% Representan la resistencia del sistema a cambios de velocidad
   angular
Jphi   = 1.3413e-3;
Jtheta = 1.3413e-3;

% Coeficientes de amortiguamiento viscoso
% Modelan pérdidas por fricción en las articulaciones y
   estructura
bphi   = 1e-2;
btheta = 1e-2;

% --------------------------------
% PARÁMETROS DEL SERVO EQUIVALENTE
% --------------------------------
% Frecuencia natural deseada del sistema en lazo cerrado
% Determina la rapidez de respuesta del actuador
wn = 20;        % [rad/s]

% Coeficiente de amortiguamiento
% zeta = 1 corresponde a amortiguamiento crítico (sin
   sobreoscilación)
zeta = 1;
```

```
45
46 % -------------------------------
47 % CÁLCULO DE GANANCIAS
48 % -------------------------------
49 % Las ganancias se obtienen imponiendo una dinámica de segundo
   orden:
50 %   phi_ddot + 2*zeta*wn*phi_dot + wn^2*phi = wn^2*phi_ref
51 %
52 % Esto permite que el sistema siga la referencia con una
   dinámica conocida
53
54 % Ganancias para el eje phi
55 Kp_phi = Jphi * wn^2;                  % Término proporcional
   (rigidez)
56 Kd_phi = 2*zeta*Jphi*wn - bphi;        % Término derivativo
   (amortiguamiento)
57
58 % Ganancias para el eje theta
59 Kp_theta = Jtheta * wn^2;
60 Kd_theta = 2*zeta*Jtheta*wn - btheta;
61
62 % -------------------------------
63 % CÁLCULO DEL ERROR
64 % -------------------------------
65 % Error de orientación entre referencia y estado actual
66 e_phi   = phi_ref   - phi;
67 e_theta = theta_ref - theta;
68
69 % -------------------------------
70 % LEY DE CONTROL (PD)
71 % -------------------------------
72 % Se implementa un control proporcional-derivativo equivalente:
73 %
74 % tau = Kp*error - Kd*velocidad
```

```
%
% - El término proporcional corrige el error de posición
% - El término derivativo introduce amortiguamiento y
   estabilidad

tau_phi   = Kp_phi   * e_phi   - Kd_phi   * phi_dot;
tau_theta = Kp_theta * e_theta - Kd_theta * theta_dot;

end
```

Listing 3: Función de la actuación equivalente de los motores.

A.4. Función de cinemática inversa

A continuación se muestra la implementación de la cinemática inversa, donde se calcula el vector de ángulos articulares a partir de las inclinaciones de referencia de la plataforma; tal como se desarrolla en la sección 4.2.

```
function q = inverse_kinematics_3rrs(phi_ref, theta_ref,
   params, elbowSign)
% INVERSE_KINEMATICS_3RRS
% Cinemática inversa simplificada de una plataforma 3-RRS.
% Válida cerca de la configuración nominal y para pequeñas
   inclinaciones.
% Calcula los ángulos articulares q1, q2 y q3 de un manipulador
   3-RRS a partir de la inclinación deseada de la plataforma.
%
% Entradas:
% phi_ref - Inclinación deseada asociada a la dinámica en X
   [rad]
% theta_ref - Inclinación deseada asociada a la dinámica en Y
   [rad]
% params - Estructura con parámetros geométricos:
% params.l1 -> longitud del brazo activo [m]
% params.l2 -> longitud del brazo pasivo [m]
```

```
% params.b -> distancia del centro de la base al anclaje [m]
% params.p -> distancia del centro de la plataforma al anclaje
    [m]
% params.z0 -> altura nominal de la plataforma [m]
% elbowSign - +1 para una rama de solución, -1 para la otra
%
% Salida:
% q - Vector columna [q1; q2; q3] en rad

arguments
phi_ref   (1,1) double
theta_ref (1,1) double
params    struct
elbowSign (1,1) double {mustBeMember(elbowSign,[-1,1])} = 1
end

% Parámetros geométricos
l1 = params.l1;
l2 = params.l2;
b  = params.b;
p  = params.p;
z0 = params.z0;

% Ángulos de distribución de las patas
alpha = [0; 2*pi/3; 4*pi/3];

% Distancia horizontal efectiva
d = b - p;

% Tolerancia numérica
tol = 1e-10;

% Inicialización
q = zeros(3,1);
```

```
46
47 for i = 1:3
48
49 % Altura local del punto de unión de la plataforma
50 hi = z0 + p * (cos(alpha(i))*phi_ref + sin(alpha(i))*theta_ref);
51
52 % Coeficientes
53 Ai = -2 * d  * l1;
54 Bi = -2 * hi * l1;
55 Ci = d^2 + hi^2 + l1^2 - l2^2;
56
57 % Discriminante
58 Delta = Ai^2 + Bi^2 - Ci^2;
59
60 if Delta < -tol
61 error('No existe solución real para la pata %d. Orientación
      fuera del espacio de trabajo.', i);
62 end
63
64 Delta = max(Delta,0);
65
66 den = Ci - Ai;
67
68 if abs(den) < tol
69 error('Denominador casi nulo en la pata %d. Posible
      singularidad.', i);
70 end
71
72 % Solución
73 q(i) = 2 * atan((-Bi + elbowSign * sqrt(Delta)) / den);
74 end
75 end
```

Listing 4: Implementación en MATLAB de la cinemática inversa del manipulador 3-RRS.

A.5. Script para la comparación del modelo Espacio de estados vs Simscape Multibody

A continuación se muestra el script en MATLAB desarrollado para graficar las respuestas del sistema mostrado en la figura 6.2, ante distintas configuraciones de las orientaciones de referencia (modificando los parámetros de los bloques *Step*):

```
%% Comparacion modelo State-Space vs Simscape
clc; close all;

z0 = 0.14;   % altura nominal [m]

%% Extraer señales desde SimulationOutput
x_SS     = getSignal(out.x_SS);
y_SS     = getSignal(out.y_SS);
phi_SS   = getSignal(out.phi_SS);
theta_SS = getSignal(out.theta_SS);

x_MB     = getSignal(out.x_MB);
y_MB     = getSignal(out.y_MB);
phi_MB   = getSignal(out.phi_MB);
theta_MB = getSignal(out.theta_MB);
z_MB     = getSignal(out.z0_MB);

%% Usar tiempo del modelo State-Space como referencia (es
   igual para ambos: 3 seg)
t = x_SS.t;

% Interpolar Simscape al mismo vector temporal
% Recomendable pues uso un solver de paso variable (ode15s)
   y cada modelo genera puntos en espacio de tiempo
   distintos
x_MB_i     = interp1(x_MB.t,     x_MB.y,     t, 'linear',
   'extrap');
```

```
y_MB_i     = interp1(y_MB.t,     y_MB.y,     t, 'linear', 'extrap');
phi_MB_i   = interp1(phi_MB.t,   phi_MB.y,   t, 'linear', 'extrap');
theta_MB_i = interp1(theta_MB.t, theta_MB.y, t, 'linear', 'extrap');
z_MB_i     = interp1(z_MB.t,     z_MB.y,     t, 'linear', 'extrap');

%% Errores
e_x = x_SS.y - x_MB_i;
e_y = y_SS.y - y_MB_i;

RMSE_x = sqrt(mean(e_x.^2)); %Root Mean Square Error (Error Cuadrático Medio Radicular)
RMSE_y = sqrt(mean(e_y.^2));

Emax_x = max(abs(e_x));
Emax_y = max(abs(e_y));

fprintf('RMSE x = %.6f m\n', RMSE_x);
fprintf('RMSE y = %.6f m\n', RMSE_y);
fprintf('Error max x = %.6f m\n', Emax_x);
fprintf('Error max y = %.6f m\n', Emax_y);

%% Figura 1: orientacion de la plataforma
figure;
plot(t, phi_SS.y, 'LineWidth', 1.5); hold on;
plot(t, phi_MB_i, '--', 'LineWidth', 1.5);
grid on;
xlabel('Tiempo [s]');
ylabel('\phi [rad]');
legend('State-Space','Simscape','Location','best');
title('Comparacion de la orientacion \phi');
```

```
53
54      figure;
55      plot(t, theta_SS.y, 'LineWidth', 1.5); hold on;
56      plot(t, theta_MB_i, '--', 'LineWidth', 1.5);
57      grid on;
58      xlabel('Tiempo [s]');
59      ylabel('\theta [rad]');
60      legend('State-Space','Simscape','Location','best');
61      title('Comparacion de la orientacion \theta');
62
63      %% Figura 2: posicion de la bola
64      figure;
65      plot(t, x_SS.y, 'LineWidth', 1.5); hold on;
66      plot(t, x_MB_i, '--', 'LineWidth', 1.5);
67      grid on;
68      xlabel('Tiempo [s]');
69      ylabel('x [m]');
70      legend('State-Space','Simscape','Location','best');
71      title('Comparacion de la posicion x de la bola');
72
73      figure;
74      plot(t, y_SS.y, 'LineWidth', 1.5); hold on;
75      plot(t, y_MB_i, '--', 'LineWidth', 1.5);
76      grid on;
77      xlabel('Tiempo [s]');
78      ylabel('y [m]');
79      legend('State-Space','Simscape','Location','best');
80      title('Comparacion de la posicion y de la bola');
81
82      %% Figura 3: errores de posicion
83      figure;
84      plot(t, e_x, 'LineWidth', 1.5); hold on;
85      plot(t, e_y, 'LineWidth', 1.5);
86      grid on;
```

```
87      xlabel('Tiempo [s]');
88      ylabel('Error [m]');
89      legend('e_x','e_y','Location','best');
90      title('Error entre modelos');
91
92      %% Figura 4: desviacion de altura
93      delta_z = z_MB_i - z0;
94      figure;
95      plot(t, delta_z, 'LineWidth', 1.5);
96      grid on;
97      xlabel('Tiempo [s]');
98      ylabel('\Delta z [m]');
99      title('Desviacion de altura de la plataforma en Simscape');
100
101     %% Funcion auxiliar de comprobacion
102     function sig = getSignal(s)
103     if isa(s,'timeseries')
104     sig.t = s.Time(:);
105     sig.y = squeeze(s.Data);
106     sig.y = sig.y(:);
107     elseif isstruct(s)
108     sig.t = s.time(:);
109     sig.y = squeeze(s.signals.values);
110     sig.y = sig.y(:);
111     else
112     error('Formato de señal no reconocido. Usa Timeseries o
            Structure with time.');
113     end
114     end
```

Listing 5: Script de comparación entre el modelo en espacio de estados y el modelo multicuerpo en Simscape.

A.6. Configuración del filtro de Kalman

En este anexo se recoge el script empleado para inicializar los parámetros del sensor visual simulado y del filtro de Kalman utilizado como estimador de estados en el capítulo 7.5.2.

```
%% Parametros del sensor
Ts_sensor = 1/60; % Camara simulada de 60 fps
sigma_sensor = 0.0005; % Desviacion tipica del ruido: 0.5 mm

%% Modelo de medida para Kalman
C_k = [1 0 0 0 0 0 0 0;
0 0 1 0 0 0 0 0];
D_k = zeros(2,2);

% Modelo continuo de la planta visto desde el sensor
% Las matrices A y B son las proporcionadas por el espacio
   de estados
sys_k_cont = ss(A,B,C_k,D_k);

% Discretizacion del modelo con el periodo de muestreo del
   sensor
sys_k = c2d(sys_k_cont,Ts_sensor,'zoh');

Ad_k = sys_k.A;
Bd_k = sys_k.B;
Cd_k = sys_k.C;
Dd_k = sys_k.D;

%% Covarianzas
Q_k = diag([1e-8 1e-6 1e-8 1e-6 1e-8 1e-6 1e-8 1e-6]);
R_k = diag([sigma_sensor^2 sigma_sensor^2]);
N_k = zeros(8,2);

%% Condiciones iniciales del estimador
% Indica con que valor inicial empieza a trabajar el filtro
```

```
    de Kalman antes de recibir y procesar las primeras
    medidas del sensor. Más información en el libro de
    D.Simon [30].

x0_k = zeros(8,1); % estimacion inicial del espacio de
    estados

P0_k = diag([1e-6 1e-4 1e-6 1e-4 1e-6 1e-4 1e-6 1e-4]); %
    covarianza inicial: incertidumbre asociada al estado
    inicial anterior
```

Listing 6: Configuración del filtro de Kalman en MATLAB.

A.7. Script de postprocesado de señales del sensor simulado

En este anexo se recoge el script empleado para generar las gráficas de comparación entre las señales reales del modelo de espacio de estados, las señales medidas por el sensor visual simulado y las señales estimadas mediante el filtro de Kalman. Este código se utiliza para obtener las figuras incluidas en la sección 7.6.4.

```
%% ============================================================
%  POSTPROCESADO DEL SENSOR SIMULADO Y FILTRO DE KALMAN
%  Representación de señales discretas con retención de
    orden cero
%
%  Variables disponibles en el workspace:
%  out.x, out.y, out.vx, out.vy
%  out.x_m, out.y_m
%  out.x_hat, out.y_hat
%  out.vx_hat, out.vy_hat
% ============================================================
close all
clc
%% ========================
```

```
14  % 1. EXTRACCIÓN Y SINCRONIZACIÓN DE DATOS
15  % ========================
16  % Tiempo de referencia: salida real del modelo de espacio
        de estados
17  t = out.x.Time;
18  % Señales reales del modelo de espacio de estados
19  x  = squeeze(out.x.Data);
20  y  = squeeze(out.y.Data);
21  vx = squeeze(out.vx.Data);
22  vy = squeeze(out.vy.Data);
23  % Señales medidas
24  t_xm = out.x_m.Time;
25  t_ym = out.y_m.Time;
26  x_m_raw = squeeze(out.x_m.Data);
27  y_m_raw = squeeze(out.y_m.Data);
28  % Señales estimadas por el filtro de Kalman
29  t_xhat  = out.x_hat.Time;
30  t_yhat  = out.y_hat.Time;
31  t_vxhat = out.vx_hat.Time;
32  t_vyhat = out.vy_hat.Time;
33  x_hat_raw  = squeeze(out.x_hat.Data);
34  y_hat_raw  = squeeze(out.y_hat.Data);
35  vx_hat_raw = squeeze(out.vx_hat.Data);
36  vy_hat_raw = squeeze(out.vy_hat.Data);
37  % Asegurar vectores columna
38  t = t(:);
39  x  = squeeze(x);   x  = x(:);
40  y  = squeeze(y);   y  = y(:);
41  vx = squeeze(vx);  vx = vx(:);
42  vy = squeeze(vy);  vy = vy(:);
43  t_xm = t_xm(:);
44  t_ym = t_ym(:);
45  t_xhat  = t_xhat(:);
46  t_yhat  = t_yhat(:);
```

```
47      t_vxhat = t_vxhat(:);
48      t_vyhat = t_vyhat(:);
49      x_m_raw    = x_m_raw(:);
50      y_m_raw    = y_m_raw(:);
51      x_hat_raw  = x_hat_raw(:);
52      y_hat_raw  = y_hat_raw(:);
53      vx_hat_raw = vx_hat_raw(:);
54      vy_hat_raw = vy_hat_raw(:);
55      % Sincronización al tiempo continuo de referencia usando
           retención de orden cero. Esto representa cómo vería el
           controlador las señales discretas entre instantes de
           muestreo.
56      x_m    = interp1(t_xm,    x_m_raw,    t, 'previous',
           'extrap');
57      y_m    = interp1(t_ym,    y_m_raw,    t, 'previous',
           'extrap');
58      x_hat  = interp1(t_xhat,  x_hat_raw,  t, 'previous',
           'extrap');
59      y_hat  = interp1(t_yhat,  y_hat_raw,  t, 'previous',
           'extrap');
60      vx_hat = interp1(t_vxhat, vx_hat_raw, t, 'previous',
           'extrap');
61      vy_hat = interp1(t_vyhat, vy_hat_raw, t, 'previous',
           'extrap');
62      x_m    = x_m(:);
63      y_m    = y_m(:);
64      x_hat  = x_hat(:);
65      y_hat  = y_hat(:);
66      vx_hat = vx_hat(:);
67      vy_hat = vy_hat(:);
68      %% =========================
69      % 2. ERRORES
70      % =========================
71      % Error entre la salida ideal del espacio de estados y la
```

```
    estimación
% discreta mantenida por el filtro de Kalman
e_x  = x  - x_hat;
e_y  = y  - y_hat;
e_vx = vx - vx_hat;
e_vy = vy - vy_hat;
% Métricas RMSE
rmse_x  = sqrt(mean(e_x.^2));
rmse_y  = sqrt(mean(e_y.^2));
rmse_vx = sqrt(mean(e_vx.^2));
rmse_vy = sqrt(mean(e_vy.^2));
%% =========================
% 3. POSICIÓN EN X
% =========================
figure('Name','Comparación de la posición x','Color','w');
plot(t, x, 'b', 'LineWidth', 0.8); hold on;
stairs(t_xm, x_m_raw, 'Color', [0.12 0.80 0.12],
    'LineWidth', 1.3);
plot(t_xhat, x_hat_raw, 'r', 'LineWidth', 0.8);
grid on;
xlabel('Tiempo [s]');
ylabel('x [m]');
title('Comparación de la posición x de la bola');
legend('Posición real x', ...
'Posición medida $x_m$', ...
'Posición estimada $\hat{x}$', ...
'Interpreter','latex', ...
'Location','best');
%% =========================
% 4. POSICIÓN EN Y
% =========================
figure('Name','Comparación de la posición y','Color','w');
plot(t, y, 'b', 'LineWidth', 2); hold on;
stairs(t_ym, y_m_raw, 'Color', [0.12 0.80 0.12],
```

```
        'LineWidth', 1);
104     stairs(t_yhat, y_hat_raw, 'r', 'LineWidth', 1);
105     grid on;
106     xlabel('Tiempo [s]');
107     ylabel('y [m]');
108     title('Comparación de la posición y de la bola');
109     legend('Posición real y', ...
110     'Posición medida $y_m$', ...
111     'Posición estimada $\hat{y}$', ...
112     'Interpreter','latex', ...
113     'Location','best');
114     ylim([-0.0017 0.0017]);
115     %% =========================
116     % 5. VELOCIDAD EN X
117     % =========================
118     figure('Name','Comparación de la velocidad
        v_x','Color','w');
119     plot(t, vx, 'b', 'LineWidth', 1.8); hold on;
120     stairs(t_vxhat, vx_hat_raw, 'r', 'LineWidth', 1.5);
121     grid on;
122     xlabel('Tiempo [s]');
123     ylabel('v_x [m/s]');
124     title('Comparación de la velocidad v_x de la bola');
125     legend('Velocidad real $\dot{x}$', ...
126     'Velocidad estimada $\hat{\dot{x}}$', ...
127     'Interpreter','latex', ...
128     'Location','best');
129     %% =========================
130     % 6. VELOCIDAD EN Y
131     % =========================
132     figure('Name','Comparación de la velocidad
        v_y','Color','w');
133     plot(t, vy, 'b', 'LineWidth', 1.8); hold on;
134     stairs(t_vyhat, vy_hat_raw, 'r', 'LineWidth', 1.5);
```

```
135     grid on;
136     xlabel('Tiempo [s]');
137     ylabel('v_y [m/s]');
138     title('Comparación de la velocidad v_y de la bola');
139     legend('Velocidad real $\dot{y}$', ...
140     'Velocidad estimada $\hat{\dot{y}}$', ...
141     'Interpreter','latex', ...
142     'Location','best');
143     %% =========================
144     % 7. ERRORES DE POSICIÓN Y VELOCIDAD
145     % =========================
146     figure('Name','Errores de estimación','Color','w');
147     subplot(2,2,1)
148     plot(t, e_x, 'b', 'LineWidth', 0.5);
149     grid on;
150     xlabel('Tiempo [s]');
151     ylabel('e_x [m]');
152     title(sprintf('Error de posición en x (RMSE = %.4e m)',
            rmse_x));
153     subplot(2,2,2)
154     plot(t, e_y, 'r', 'LineWidth', 0.5);
155     grid on;
156     xlabel('Tiempo [s]');
157     ylabel('e_y [m]');
158     title(sprintf('Error de posición en y (RMSE = %.4e m)',
            rmse_y));
159     subplot(2,2,3)
160     plot(t, e_vx, 'm', 'LineWidth', 0.5);
161     grid on;
162     xlabel('Tiempo [s]');
163     ylabel('e_{v_x} [m/s]');
164     title(sprintf('Error de velocidad en x (RMSE = %.4e m/s)',
            rmse_vx));
165     subplot(2,2,4)
```

```
166     plot(t, e_vy, 'k', 'LineWidth', 0.5);
167     grid on;
168     xlabel('Tiempo [s]');
169     ylabel('e_{v_y} [m/s]');
170     title(sprintf('Error de velocidad en y (RMSE = %.4e m/s)',
            rmse_vy));
171     sgtitle('Errores entre el modelo de espacio de estados y la
            estimación de Kalman');
172     %% =========================
173     % 8. MENSAJES EN CONSOLA
174     % =========================
175     fprintf('\n===== MÉTRICAS DE ERROR =====\n');
176     fprintf('RMSE posición x   = %.6e m\n',   rmse_x);
177     fprintf('RMSE posición y   = %.6e m\n',   rmse_y);
178     fprintf('RMSE velocidad vx = %.6e m/s\n', rmse_vx);
179     fprintf('RMSE velocidad vy = %.6e m/s\n', rmse_vy);
180     fprintf('============================\n\n');
```

Listing 7: Postprocesado de datos del sensor y filtro de Kalman en MATLAB.

A.8. Funciones de transferencia asociadas al espacio de estados

Se presenta el *script* en MATLAB que permite obtener las funciones de transferencia equivalentes del modelo conjunto planta-actuación equivalente empleadas, posteriormente, para la sintonización del controlador PID.

```
1     %% Obtención de la función de transferencia equivalente
          planta + actuación simplificada
2     % Este script obtiene las funciones de transferencia
          equivalentes empleadas para la sintonización del
          controlador PID. Se parte del modelo lineal en espacio
          de estados y se incorpora el efecto del bloque de
          actuación equivalente.
3     %
```

```
%% Conversión del modelo en espacio de estados a funciones
   de transferencia
% Entrada 1: tau_phi
[num_phi, den_phi] = ss2tf(A, B, C, D, 1);

% Entrada 2: tau_theta
[num_theta, den_theta] = ss2tf(A, B, C, D, 2);

% Función desde tau_phi hasta x
G_x_tau = tf(num_phi(1,:), den_phi);

% Función desde tau_theta hasta y
G_y_tau = tf(num_theta(3,:), den_theta);

% Función desde tau_phi hasta phi
G_phi_tau = tf(num_phi(5,:), den_phi);

% Función desde tau_theta hasta theta
G_theta_tau = tf(num_theta(7,:), den_theta);

%% Parámetros del actuador equivalente
% Momentos de inercia
Jphi = 1.3413e-3;
Jtheta = 1.3413e-3;

% Amortiguamiento viscoso
bphi = 1e-2;
btheta = 1e-2;

% Parámetros de la dinámica deseada del servo equivalente
wn = 20; % Frecuencia natural [rad/s]
zeta = 1; % Amortiguamiento crítico

% Ganancias del actuador equivalente
```

```
Kp_phi = Jphi * wn^2;
Kd_phi = 2*zeta*Jphi*wn - bphi;
Kp_theta = Jtheta * wn^2;
Kd_theta = 2*zeta*Jtheta*wn - btheta;

%% Obtención de las funciones de transferencia equivalentes
s = tf('s');

% Función equivalente eje X: X(s)/Phi_ref(s)
G_eq_x = minreal( ...
(Kp_phi * G_x_tau) / ...
(1 + (Kp_phi + Kd_phi*s) * G_phi_tau) ...
)

% Función equivalente eje Y: Y(s)/Theta_ref(s)
G_eq_y = minreal( ...
(Kp_theta * G_y_tau) / ...
(1 + (Kp_theta + Kd_theta*s) * G_theta_tau) ...
)

%% Comprobación de equivalencia entre ejes
G_diff = minreal(G_eq_x - G_eq_y);
tol = 1e-8;
[num_diff, ~] = tfdata(G_diff, 'v');

if all(abs(num_diff) < tol)
disp('Resultado: G_eq_x y G_eq_y son iguales.');
else
disp('Resultado: G_eq_x y G_eq_y son diferentes.');
disp('Diferencia entre ambas funciones:');
G_diff
end
```

Listing 8: Obtención de la función de transferencia equivalente planta + actuación

A.9. Script para el análisis y diseño del control en espacio de estados

Este apartado contiene el script de MATLAB utilizado para verificar las propiedades estructurales del sistema linealizado y calcular la ganancia LQR asociada al índice de coste cuadrático definido. En primer lugar, el código evalúa la controlabilidad del estado y de la salida (posiciones de la bola), así como la observabilidad del sistema a partir de las medidas del sensor visual. En segundo lugar, aplica la regla de Bryson para construir las matrices de ponderación **Q** y **R**, y calcula mediante el comando `lqr` la matriz de ganancia óptima **K** y los polos en lazo cerrado correspondientes.

```
%% CONTROL EN ESPACIO DE ESTADOS
% Parto de las matrices A,B,C,D (almacenadas en el workspace);
   definidas en el anexo A.1 e inicializadas mediante los
   parámetros geométricos y dinámicos de mi sistema en el anexo
   A.2
clc;

%% Dimensiones del sistema
n = size(A,1);    % Numero de estados
m = size(B,2);    % Numero de entradas

fprintf('Numero de estados: %d\n', n);
fprintf('Numero de entradas: %d\n\n', m);

%% 1. Controlabilidad del estado
Co = ctrb(A,B);
rank_Co = rank(Co);

fprintf('===== Controlabilidad del estado =====\n');
fprintf('Rango de la matriz de controlabilidad: %d\n', rank_Co);
fprintf('Numero de estados del sistema: %d\n', n);

if rank_Co == n
fprintf('Resultado: El sistema es completamente controlable en
```

```
    estado.\n\n');
else
fprintf('Resultado: El sistema NO es completamente controlable
    en estado.\n');
fprintf('Estados no controlables: %d\n\n', n - rank_Co);
end

%% 2. Controlabilidad de la salida
% Definicion de las salidas de interes.
% En este caso se consideran como salidas las posiciones de la
    bola:
% y_pos = [x; y]
Cpos = [1 0 0 0 0 0 0 0;
0 0 1 0 0 0 0 0];

Dpos = zeros(2,m);

p = size(Cpos,1);   % Numero de salidas consideradas

% Construccion de la matriz de controlabilidad de salida:
% Cy = [D, C B, C A B, C A^2 B, ..., C A^(n-1) B]
% Si D = 0, el termino D no afecta al rango, pero se incluye
    por generalidad.
Cy = Dpos;

for k = 0:n-1
Cy = [Cy, Cpos*(A^k)*B];
end

rank_Cy = rank(Cy);

fprintf('===== Controlabilidad de la salida =====\n');
fprintf('Rango de la matriz de controlabilidad de salida:
    %d\n', rank_Cy);
```

```
51 fprintf('Numero de salidas consideradas: %d\n', p);
52
53 if rank_Cy == p
54 fprintf('Resultado: Las salidas consideradas son
      controlables.\n\n');
55 else
56 fprintf('Resultado: Las salidas consideradas NO son
      completamente controlables.\n');
57 fprintf('Salidas no controlables: %d\n\n', p - rank_Cy);
58 end
59
60 %% 3. Observabilidad del sistema
61 % Para analizar la observabilidad se considera que el sistema
      de medida proporciona únicamente la posicion de la bola:
62 %
63 % y_med = [x_m; y_m]
64 %
65 % Por tanto, la matriz de salida considerada es equivalente a
      Cpos.
66 Cmed = Cpos;
67
68 Ob = obsv(A,Cmed);
69 rank_Ob = rank(Ob);
70
71 fprintf('===== Observabilidad del sistema =====\n');
72 fprintf('Rango de la matriz de observabilidad: %d\n', rank_Ob);
73 fprintf('Numero de estados del sistema: %d\n', n);
74
75 if rank_Ob == n
76 fprintf('Resultado: El sistema es completamente observable a
      partir de x_m e y_m.\n\n');
77 else
78 fprintf('Resultado: El sistema NO es completamente observable a
      partir de x_m e y_m.\n');
```

```
fprintf('Estados no observables: %d\n\n', n - rank_Ob);
end

%% Diseño LQR para realimentación de estados basada en
   referencias de inclinación
% La salida del controlador será:
% u_ref = [phi_ref; theta_ref]
%
% El bloque de actuación equivalente convierte:
% [phi_ref; theta_ref] -> [tau_phi; tau_theta]

%% Parámetros del actuador equivalente
Jphi   = 1.3413e-3;
Jtheta = 1.3413e-3;
bphi   = 1e-2;
btheta = 1e-2;

wn   = 20;
zeta = 1;

Kp_phi = Jphi * wn^2;
Kd_phi = 2*zeta*Jphi*wn - bphi; % Se resta el amortiguamiento
   viscoso b porque dicho término ya está incluido en el modelo
   dinámico de la planta.

Kp_theta = Jtheta * wn^2;
Kd_theta = 2*zeta*Jtheta*wn - btheta;

%% Matrices equivalentes del actuador
% Vector de estados:
% x = [x, x_dot, y, y_dot, phi, phi_dot, theta, theta_dot]^T

F = [0 0 0 0 Kp_phi   Kd_phi   0          0;
0 0 0 0 0         0         Kp_theta   Kd_theta];
```

```
110
111 G = [Kp_phi    0;
112 0          Kp_theta];
113
114 %% Planta equivalente: actuación simplificada + modelo en
    espacio de estados
115 A_eq = A - B*F;
116 B_eq = B*G;
117
118 %% Comprobación de controlabilidad del sistema equivalente
119 Co_eq = ctrb(A_eq, B_eq);
120 rank_Co_eq = rank(Co_eq);
121 n = size(A_eq,1);
122
123 fprintf('===== Controlabilidad del sistema equivalente
    =====\n');
124 fprintf('Rango: %d / %d\n', rank_Co_eq, n);
125
126 if rank_Co_eq == n
127 fprintf('El sistema equivalente es controlable.\n\n');
128 else
129 fprintf('El sistema equivalente NO es completamente
    controlable.\n\n');
130 end
131
132 %% Diseño LQR mediante regla de Bryson
133 x_max       = 0.03;           % [m]
134 xdot_max    = 0.10;           % [m/s]
135 y_max       = 0.03;           % [m]
136 ydot_max    = 0.10;           % [m/s]
137
138 phi_max     = 5*pi/180;      % [rad]
139 phid_max    = 1.0;            % [rad/s]
140 theta_max   = 5*pi/180;      % [rad]
```

```
141 thetad_max = 1.0;            % [rad/s]
142
143 % Ahora la entrada del LQR son referencias angulares
144 u_ref_max = 10*pi/180;       % [rad]
145
146 Q_lqr = diag([ ...
147 1/x_max^2, ...
148 1/xdot_max^2, ...
149 1/y_max^2, ...
150 1/ydot_max^2, ...
151 1/phi_max^2, ...
152 1/phid_max^2, ...
153 1/theta_max^2, ...
154 1/thetad_max^2]);
155
156 R_lqr = diag([ ...
157 1/u_ref_max^2, ...
158 1/u_ref_max^2]);
159
160 K = lqr(A_eq, B_eq, Q_lqr, R_lqr);
161
162 eig_cl = eig(A_eq - B_eq*K);
163
164 disp('Matriz K de realimentacion para generar referencias
       angulares:');
165 disp(K);
166 disp('Polos del sistema equivalente en lazo cerrado:');
167 disp(eig_cl);
```

Listing 9: Análisis de controlabilidad, observabilidad y sintonización LQR.

BIBLIOGRAFÍA

[1] L.-W. Tsai, *Robot Analysis: The Mechanics of Serial and Parallel Manipulators*. Wiley, 1999, ISBN: 978-0-471-32593-2.

[2] M. W. Spong, S. Hutchinson y M. Vidyasagar, *Robot Modeling and Control*, 2.ª ed. Wiley, 2020, ISBN: 978-1-119-52399-4.

[3] M. A. Rastin, E. Talebzadeh, S. A. A. Moosavian y M. Alaeddin, «Trajectory tracking and obstacle avoidance of a ball and plate system using fuzzy theory,» en *13th Iranian Conference on Fuzzy Systems (IFSC)*, 2013, págs. 1-5. DOI: 10.1109/IFSC.2013.6675631

[4] J. J. Craig, *Introduction to Robotics: Mechanics and Control*, 3.ª ed. Pearson, Prentice Hall, 2005, ISBN: 978-0-13-123629-5.

[5] ABB Group, *IRB 360 | ABB*, Accedido: 13-04-2026. dirección: https://www.abb.com/global/en/areas/robotics/products

[6] A. Barrientos, A. Cruz, L. Peñín y C. Balaguer, *Fundamentos de Robótica*. McGraw-Hill, 2007, ISBN: 978-84-481-5636-7.

[7] N. Ruiz-Hidalgo, A. Blanco-Ortega, A. Abúndez-Pliego, J. Ocampo, W. Alcocer y M. Arias-Montiel, «DINÁMICA Y CONTROL DE UN ROBOT PARALELO 3-RPS,» *Pistas Educativas*, vol. 39, págs. 518-542, 2017.

[8] Acrome Robotics, *Productos comerciales de Acrome Robotics*, Accedido: 13-04-2026. dirección: https://acrome.net/

[9] J. Gómez Eguizábal, «Sistema de control barra-bola para la docencia de automática,» Trabajo Fin de Grado, Universidad de Cantabria, 2023. dirección: https://hdl.handle.net/10902/29505

[10] A. Sentís Fàbregas, «Prototip d'un sistema ball & plate amb control PID controlat per Arduino,» Trabajo Fin de Grado, Universitat Rovira i Virgili (URV), 2021. dirección: https://hdl.handle.net/20.500.11797/TFG4722

[11] J. F. Quesada Marañón, «Prototipo de Ball & Plate para control con retroalimentación visual y CUDA,» Trabajo Fin de Grado, Universidad de Sevilla, 2020. dirección: https://hdl.handle.net/11441/102088

[12] C. García Martín, «Diseño, construcción y control de una planta Ball and Plate,» Trabajo Fin de Máster, Universidad de Jaén, 2018. dirección: https://hdl.handle.net/10953.1/14304

[13] S. X. Arroyo Gallardo, «Diseño, implementación y comparación de los algoritmos de control SMC, Fuzzy y FSMC, de un sistema de balance ball & plate (esfera & plato) para el control de posición y seguimiento de camino de una esfera,» Trabajo Fin de Máster, Escuela Politécnica Nacional de Quito, 2019. dirección: http://bibdigital.epn.edu.ec/handle/15000/21359

[14] F. Strand y M. Wahlund, «Control of the ball and plate system : A comparative study of controllers,» KTH, School of Industrial Engineering y Management (ITM), 2022, pág. 140.

[15] J. Alsamarji, «Design and Control of a Ball and Plate Didactic Device,» Accedido: 13-04-2026, Trabajo Fin de Máster, Université Libre de Bruxelles, 2019. dirección: https://github.com/jihadsamarji/Ball-Plate-Control

[16] H. Tetik, «Modelling and control of a 3-RRS parallel manipulator,» Trabajo Fin de Máster, Izmir Institute of Technology (Turkey), 2016. dirección: https://hdl.handle.net/11147/2831

[17] S. A. A. Moosavian, O. Mahdizadeh, S. Hallajian, S. Ashrafi, M. Jamali y V. Akbari, «Forward Kinematics Development of a 3-RRS Parallel Manipulator for Real-Time

Control,» en *10th RSI International Conference on Robotics and Mechatronics (ICRoM)*, 2022, págs. 459-465. DOI: 10.1109/ICRoM57054.2022.10025318

[18] J. Li, J. Wang, W. Chou, Y. Zhang, T. Wang y Q. Zhang, «Inverse kinematics and dynamics of the 3-RRS parallel platform,» en *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No.01CH37164)*, 2001, 2506-2511 vol.3. DOI: 10.1109/ROBOT.2001.932999

[19] O. Carbajal-Espinosa, F. Izar-Bonilla, M. Díaz-Rodríguez y E. Bayro-Corrochano, «Inverse kinematics of a 3 DOF parallel manipulator: A conformal geometric algebra approach,» en *IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, 2016, págs. 766-771. DOI: 10.1109/HUMANOIDS.2016.7803360

[20] R. Martínez-González, J. Hernández-Reyes y G. Sánchez-Vázquez, «OBTAINING INVERSE KINEMATICS EQUATIONS FOR A PLANAR BALL-PLATE ROBOT,» *International Journal of Information Technology, Control and Automation*, vol. 15, págs. 01-18, 2025. DOI: 10.5121/ijitca.2022.15401

[21] R. K. Pour, H. Khajvand y S. A. A. Moosavian, «Fuzzy logic trajectory tracking control of a 3-RRS ball and plate parallel manipulator,» en *4th International Conference on Robotics and Mechatronics (ICROM)*, 2016, págs. 343-348. DOI: 10.1109/ICRoM.2016.7886762

[22] H.-H. Pham, D.-H. Doan, H.-N. Vo, V.-T. Dang, H.-Q.-B. Nguyen, T.-V. Le, T.-T.-Q. Truong y T.-D. Tran, «Ball Balancing on 3-RRS Parallel Manipulator Using PD and LQR Control,» *Review of Mechatronics*, págs. 38-44, 2023, Accedido: 2026-04-22. dirección: https://rm.reviste.ubbcluj.ro/wp-content/uploads/2024/02/RM_2023_2_Pag_38_Pham_HH.pdf

[23] S. Miller, *Simscape Multibody Contact Forces Library*, Accedido: 03-05-2026. dirección: https://es.mathworks.com/matlabcentral/fileexchange/47417-simscape-multibody-contact-forces-library

[24] R. C. Dorf, *Sistemas de control moderno*, 10.ª ed. Pearson Prentice Hall, 2005, ISBN: 978-84-205-4401-4.

[25] H. Tetik, R. Kalla, G. Kiper y S. Bandyopadhyay, «Position Kinematics of a 3-RRS Parallel Manipulator,» vol. 569, 2016, págs. 65-72, ISBN: 978-3-319-33713-5. DOI: 10.1007/978-3-319-33714-2_8

[26] K. Ogata, *Ingeniería de control moderna*, 5.ª ed. Prentice Hall, 2003, ISBN: 978-84-8322-660-5.

[27] C. Pérez Vidal, *Control Sensorial de Sistemas Robóticos*, 1st ed. Universidad Miguel Hernández, 2017, ISBN: 978-84-16024-49-0.

[28] The MathWorks, Inc., *Install the Simscape Multibody Link Plugin*, Accedido: 03-05-2026. dirección: https://es.mathworks.com/help/smlink/ug/installing-and-linking-simmechanics-link-software.html

[29] G. F. Franklin, J. D. Powell y M. Workman, *Digital Control of Dynamic Systems*, 3.ª ed. Addison-Wesley, 1997, ISBN: 978-0-201-82054-6.

[30] D. Simon, *Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches*. Wiley-Interscience, 2006, ISBN: 978-0-471-70858-2.

[31] S. W. Smith, *The Scientist and Engineer's Guide to Digital Signal Processing*, 2.ª ed. California Technical Publishing, 1999, ISBN: 978-0-9660176-4-9.

[32] The MathWorks, Inc., *Simulink Block Library*, Accedido: 07-05-2026. dirección: https://www.mathworks.com/help/simulink/block-libraries.html

[33] The MathWorks, Inc., *Kalman Filter: Estimate states of discrete-time or continuous-time linear system*, Accedido: 07-05-2026. dirección: https://www.mathworks.com/help/ident/ref/kalmanfilter.html

[34] N. S. Nise, *Sistemas de control para ingeniería*. Compañía Editorial Continental, 2006, ISBN: 978-970-24-0254-1.

[35] S. Domínguez, P. Campoy, J. M. Sebastían y A. Jiménez, *Control en el espacio de estado*, 2.ª ed. Prentice-Hall, 2006, ISBN: 978-84-8322-297-3.

[36] The MathWorks, Inc., *lqr: Linear-Quadratic Regulator (LQR) design*, Accedido: 15-05-2026. dirección: `https://es.mathworks.com/help/control/ref/lti.lqr.html`