



Universidad de Buenos Aires

FACULTAD DE INGENIERÍA

LABORATORIO DE CONTROL (86.22)

Trabajo Practico N°3 - Grupo 4

2°C 2024

Alumno: Milidoni, Cristian Julian

Padrón: 107647

Alumno: Nobili, Nicolas

Padrón: 107540

Septiembre 2024

Contents

1	Introducción	2
2	Planta	3
3	Observador	4
3.1	Diseño	4
3.2	Mediciones	5
4	Controlador	7
4.1	Controlador por estados simple	7
4.1.1	Diseño	7
4.1.2	Precompensación: Matriz de Feedforward	8
4.1.3	Mediciones y Simulaciones	8
4.2	Controlador por Estados con Acción Integral	10
4.2.1	Diseño	10
4.2.2	Mediciones y Simulaciones	11
5	Conclusión	14

1 Introducción

En el trabajo anterior se identificó y modeló una planta compuesta por un Péndulo de Furuta como se muestra en la Fig. 1, y se diseñaron controladores del tipo P, PI y PD para su control. El objetivo de este trabajo práctico es, partiendo de lo desarrollado anteriormente, mejorar el desempeño del sistema mediante la implementación de un esquema de control por realimentación de estados. Este enfoque permitirá estabilizar el péndulo en torno a su punto de equilibrio estable y, además, garantizar un seguimiento robusto de referencias externas para el ángulo del brazo ϕ .



Figure 1: Planta a identificar y controlar.

2 Planta

La planta a controlar consiste en un péndulo que pivota sobre un brazo acoplado a un servomotor, conocida comúnmente como Péndulo de Furuta. En la Fig. 1 se mostró la planta real implementada mientras que en la Fig. 2 se muestra un diagrama ilustrativo de la misma.

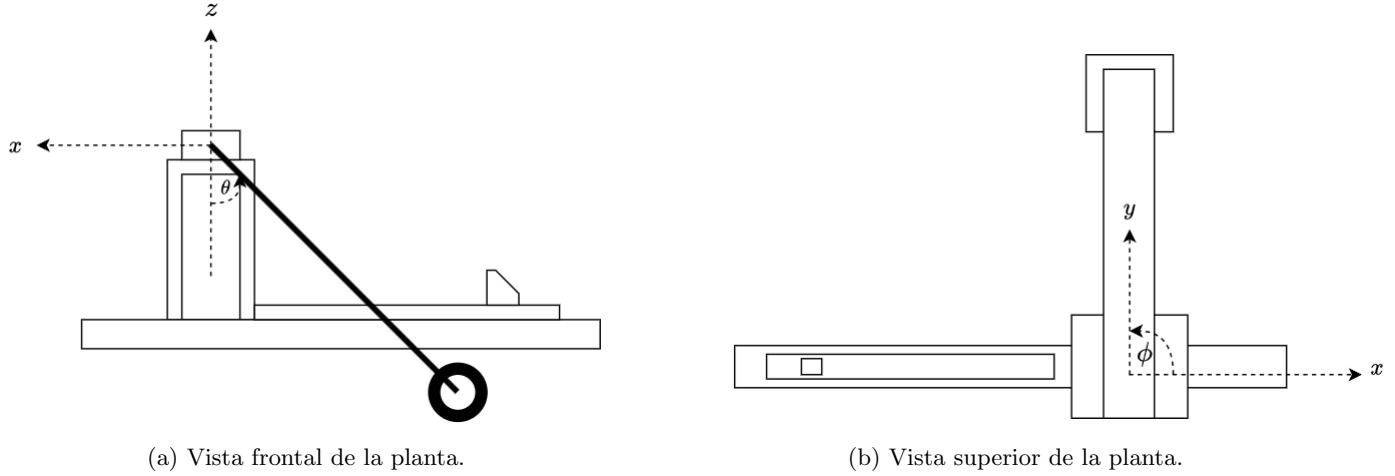


Figure 2: Diagramas de la planta: vistas frontal y superior.

El objetivo es diseñar un controlador mediante realimentación de estados para estabilizar la planta. Las variables de interés corresponden a las variables de estado de la planta, junto con la acción de control, descritas de la siguiente manera:

- θ : Ángulo del péndulo medido en sentido antihorario.
- $\dot{\theta}$: Velocidad angular del péndulo en sentido antihorario.
- ϕ : Ángulo del brazo del servomotor medido en sentido antihorario.
- $\dot{\phi}$: Velocidad angular del brazo del servomotor en sentido antihorario.
- u : Señal de control enviada al servomotor, que actúa como entrada del sistema. Esta señal corresponde al ángulo de referencia ϕ_{ref} para el brazo del servomotor, medido en radianes.

A partir de la identificación realizada en el trabajo práctico anterior, se obtuvo la siguiente representación del sistema en variables de estado, la cual será utilizada para implementar la realimentación de estados:

$$\begin{bmatrix} \dot{\theta} \\ \ddot{\theta} \\ \dot{\phi} \\ \ddot{\phi} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -61.8768 & -0.3136 & 222.1301 & 26.9333 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -272.0799 & -38.9897 \end{bmatrix} \cdot \begin{bmatrix} \theta \\ \dot{\theta} \\ \phi \\ \dot{\phi} \end{bmatrix} + \begin{bmatrix} 0 \\ -222.1488 \\ 0 \\ 240.9646 \end{bmatrix} \cdot u$$

$$y = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} \theta \\ \dot{\theta} \\ \phi \\ \dot{\phi} \end{bmatrix}$$

Cabe aclarar que a diferencia del trabajo práctico anterior, la variable de estado ϕ también es una salida del sistema.

3 Observador

Para implementar una realimentación por estados en un sistema, es indispensable conocer todos los estados del mismo. Sin embargo, en la práctica, no siempre es posible medir directamente todas las variables de estado. En estos casos, se recurre al uso de un **observador**, que es un sistema dinámico diseñado específicamente para estimar los estados internos del sistema a partir de sus entradas y salidas.

3.1 Diseño

En este trabajo, utilizaremos un **observador de Luenberger**, que es una herramienta comúnmente empleada en sistemas lineales y tiempo invariante. Para garantizar que sea posible estimar los estados internos utilizando este observador, el sistema debe ser **observable**. Esto implica que la matriz de observabilidad del sistema debe tener rango completo, es decir, igual al número de estados del sistema.

La matriz de observabilidad se define como:

$$\mathcal{O} = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{n-1} \end{bmatrix}$$

Donde:

- C : Matriz de salida del sistema.
- A : Matriz de dinámica del sistema.
- n : Número de estados del sistema.

Para nuestra planta, se verifica que es **observable**, ya que la matriz de observabilidad tiene rango completo. Esto asegura que las salidas del sistema contienen suficiente información para reconstruir los estados. En particular, esto ocurre porque todos los estados afectan a la salida del sistema, ya sea de manera directa (a través de la matriz C) o indirecta, al influir en la dinámica del sistema a través de la matriz A . Gracias a esta propiedad, es posible diseñar un observador que estime todos los estados del sistema.

Las ecuaciones del **observador de Luenberger discreto** que utilizaremos son las siguientes:

$$\begin{aligned}\hat{x}(k+1) &= A_d \hat{x}(k) + B_d u(k) + L(y(k) - \hat{y}(k)) \\ \hat{y}(k) &= C \hat{x}(k) + D u(k)\end{aligned}$$

Donde:

- $\hat{x}(k)$: Es la estimación del vector de estado $x(k)$.
- $y(k)$: Es la salida medida del sistema.
- $\hat{y}(k)$: Es la salida estimada del sistema basada en $\hat{x}(k)$.
- L : Es la matriz de ganancia del observador, que determina la rapidez con la que el observador corrige el error entre $y(k)$ y $\hat{y}(k)$.
- A, B, C, D : Son las matrices del modelo del sistema original.

En estas ecuaciones, el término $L(y(k) - \hat{y}(k))$ actúa como una retroalimentación que corrige la estimación de los estados $\hat{x}(k)$ en función del error de estimación $(y(k) - \hat{y}(k))$. La matriz L debe ser diseñada de manera que el error de estimación $e(k) = x(k) - \hat{x}(k)$ converja a cero de forma estable.

La dinámica del error de estimación $e(k)$ está determinada por el sistema descrito por la matriz:

$$A_e = A_d - LC_d$$

Para diseñar la matriz de ganancia del observador L , se pueden aplicar conceptos de realimentación de estados, seleccionando los polos deseados para la matriz A_e de modo que el observador tenga el comportamiento dinámico esperado. En MATLAB, esto se puede realizar utilizando el comando `PLACE()`, que permite calcular L para ubicar los polos en las posiciones especificadas. La elección de los polos del observador debe ser tal que estos sean mas rápidos que los del sistema original y el sistema a lazo cerrado controlado de forma de garantizar que la estimación del estado sea lo suficientemente rápida para seguir la dinámica del sistema.

Los polos continuos elegidos para el observador, ordenados según la variable de estado observada $(\theta, \dot{\theta}, \phi, \dot{\phi})$, son los siguientes:

$$p_{\text{obs}}^c = \{-30, -30, -10, -10\}$$

Se seleccionaron polos rápidos para observar las variables relacionadas con el péndulo, como el ángulo θ y su velocidad angular $\dot{\theta}$. Esta elección permite capturar de manera precisa la dinámica rápida, especialmente en situaciones donde las oscilaciones son pronunciadas o se requieren correcciones inmediatas. Al priorizar una respuesta más ágil en estas variables, se asegura que el sistema sea capaz de reaccionar rápidamente a perturbaciones o desviaciones con respecto al equilibrio.

Por otro lado, se eligieron polos más lentos que los de la planta para observar las variables asociadas al movimiento del brazo. Esta decisión se fundamenta en que la dinámica del servomotor es significativamente más rápida en comparación con la velocidad objetivo del control. En otras palabras, el péndulo se controlará a un ritmo más lento que el movimiento del servomotor.

Además, emplear polos más lentos para el ángulo ϕ ayuda a filtrar el ruido proveniente del potenciómetro, evitando que dicho ruido afecte las observaciones de las variables de estado y asegurando observaciones más estables y confiables.

La ganancia del observador de Luenberger utilizando matlab es:

$$L = \begin{bmatrix} 0.3450 & 0.2478 \\ 1.6387 & 3.1239 \\ 0 & 0.0502 \\ 0 & -1.5472 \end{bmatrix}$$

3.2 Mediciones

Se implemento el observador planteado en arduino y se compararon los estados estimados con los medidos a partir de los sensores de la planta. Además de las mediciones de las salidas θ y ϕ , se midio la velocidad angular $\dot{\theta}$ a partir del giroscopo de la IMU y se estimo la velocidad angular del brazo $\dot{\phi}$ a partir de diferencias finitas para comparar con el valor observado. En la Fig. 3 y Fig. 4 se muestran los resultados de las variables de estado observadas comparadas con las medidas.

El análisis del gráfico muestra que el observador diseñado cumple con el funcionamiento esperado, proporcionando estimaciones precisas de los estados del sistema. Además, también se observo que, ante un fallo en la IMU que genera un pico inesperado en la medición de la velocidad, el observador no se ve afectado significativamente. Esto se debe a que el fallo es de alta frecuencia, y el diseño del observador atenúa este tipo de perturbaciones, ignorando el error en la medición y manteniendo una estimación estable y confiable de los estados. Este comportamiento demuestra la robustez del observador frente a errores de medición transitorios.

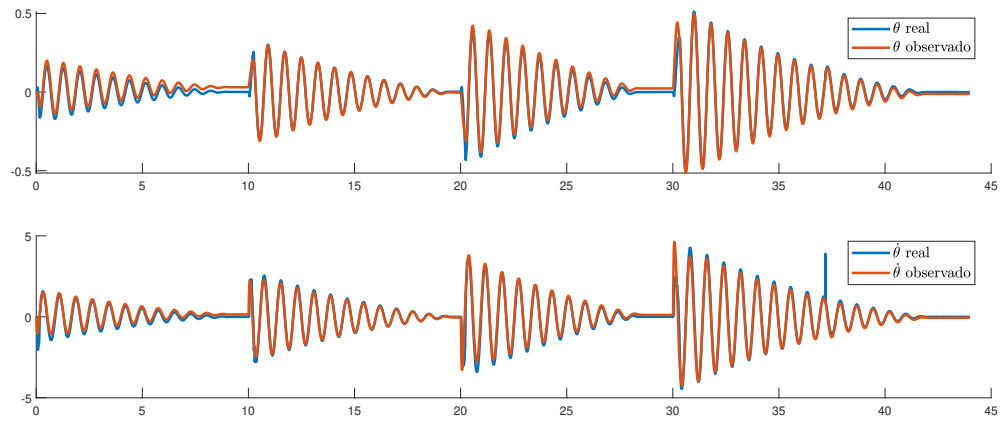


Figure 3: Comparación entre la medición y la estimación del ángulo del péndulo θ y la velocidad angular $\dot{\theta}$.

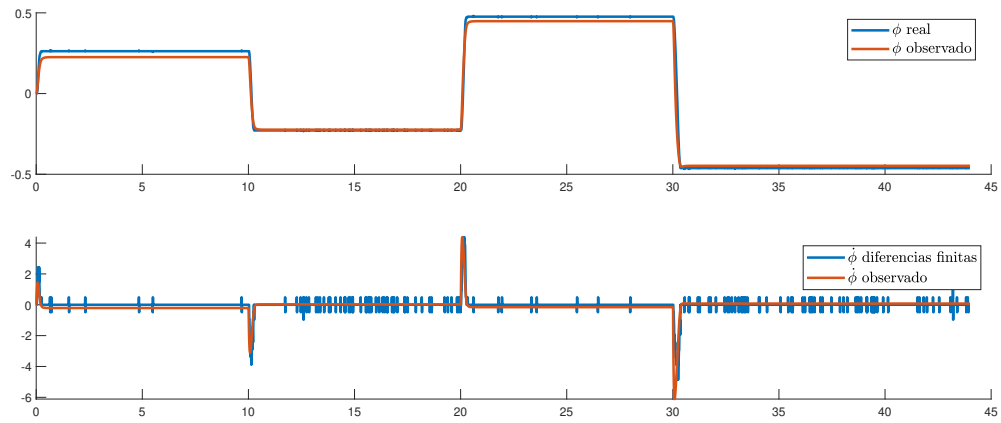


Figure 4: Comparación entre la medición y la estimación del ángulo del brazo ϕ y la velocidad angular $\dot{\phi}$.

4 Controlador

En esta sección se diseñaron dos controladores por realimentación de estados utilizando las estimaciones de los estados proporcionadas por el observador.

En primer lugar, se desarrolló un controlador por realimentación de estados simple, cuyo objetivo era estabilizar el sistema en torno al punto de equilibrio. A este controlador se le incorporó posteriormente una precompensación mediante la matriz de feedforward F , con el fin de dotar al sistema de la capacidad de seguir referencias de manera más efectiva.

En segundo lugar, se reformuló el controlador para incluir una acción integral, lo que permitió mejorar el seguimiento de referencias de tipo escalón y, además, implementar un mecanismo de rechazo de perturbaciones. Esta adición garantizó una mayor precisión en el seguimiento de las señales de referencia y una mayor robustez frente a las variaciones no modeladas o perturbaciones externas en el sistema.

4.1 Controlador por estados simple

4.1.1 Diseño

Se diseñó un controlador por realimentación de estados para estabilizar el sistema en torno al punto de equilibrio, en la Fig.5 se muestra el diagrama en bloques de la realimentación. Antes de implementar esta técnica, es esencial garantizar que el sistema sea **completamente controlable**, lo que asegura que todos los estados del sistema puedan ser influenciados por las entradas de control.

Para verificar la controlabilidad, se evalúa el **criterio de controlabilidad**. Un sistema lineal en espacio de estados descrito por (A, B) es completamente controlable si la matriz de controlabilidad:

$$\mathcal{C} = [B \quad AB \quad A^2B \quad \dots \quad A^{n-1}B]$$

tiene rango completo, es decir, si $\text{rank}(\mathcal{C}) = n$, donde n es el número de estados del sistema. En la práctica, esta verificación se realiza utilizando herramientas como MATLAB. El comando `ctrb()` permite calcular \mathcal{C} , y `rank()` comprueba su rango. Para el sistema estudiado, se confirmó que \mathcal{C} tiene rango completo, lo que garantiza que el sistema es controlable.

La ecuación de estado del sistema a lazo cerrado se obtiene al sustituir la ley de control $u = K\mathbf{x}$, donde K es la matriz de realimentación de estados. Esto resulta en:

$$\dot{\mathbf{x}} = (A + BK)\mathbf{x}.$$

La matriz K se calcula seleccionando la ubicación deseada de los polos a lazo cerrado, utilizando el comando `place()` de MATLAB. Para garantizar la estabilidad del sistema, los polos deben ubicarse en el semiplano izquierdo (para sistemas continuos) o dentro del círculo unitario (para sistemas discretos). La ubicación de estos polos determina las características dinámicas del sistema, como la rapidez de respuesta y el amortiguamiento.

Con el objetivo de diseñar un controlador que ofrezca una respuesta rápida sin saturar la acción de control, se seleccionaron los polos a lazo cerrado en tiempo continuo por su facilidad de interpretación y ajuste. Posteriormente, se calcularon sus equivalentes discretos teniendo en cuenta el método de discretización empleado.

Los polos a lazo cerrado seleccionados en tiempo continuo son:

$$p_c = [-5 - 7.8646i \quad -5 + 7.8646i \quad -10 \quad -10],$$

y sus equivalentes discretos, obtenidos mediante la discretización con **zero-order hold** (retención de orden cero), son:

$$p_d = [0.9483 - 0.0747i \quad 0.9483 + 0.0747i \quad 0.9048 \quad 0.9048].$$

A partir de estos polos discretizados, se utilizó el comando `place()` de MATLAB para calcular la matriz de realimentación de estados K . La matriz obtenida es:

$$K = [0.7159 \quad -0.0530 \quad 0.4258 \quad -0.0347].$$

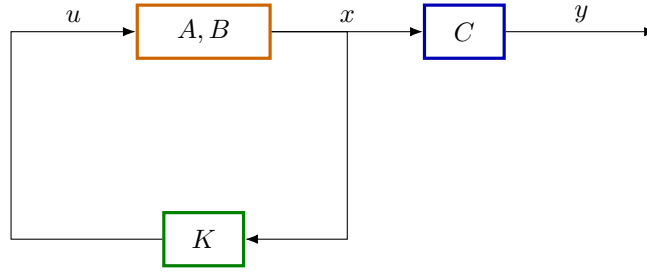


Figure 5: Diagrama en bloques de realimentación de estados

4.1.2 Precompensación: Matriz de Feedforward

Una vez calculada la matriz de realimentación de estados K , se añadió la capacidad de seguimiento simultáneo de referencias para ϕ y θ mediante una precompensación utilizando la matriz de feedforward F . Esta matriz se calcula de la siguiente forma:

$$F = \frac{1}{C(I - (A_d + B_d K))^{-1} B_d},$$

donde C , A_d , B_d , y K corresponden a los parámetros y matrices del sistema discreto y el controlador diseñado. En este caso particular, el valor obtenido para F es:

$$F = \frac{1}{\begin{bmatrix} 0 & 2.0528 \end{bmatrix}}.$$

Esta solución permite integrar de manera sencilla el seguimiento de referencias al controlador diseñado. Sin embargo, presenta ciertas limitaciones: es altamente sensible a las variaciones en los parámetros de la planta y al modelo utilizado para su diseño. Además, no compensa perturbaciones que afecten directamente a la salida del sistema.

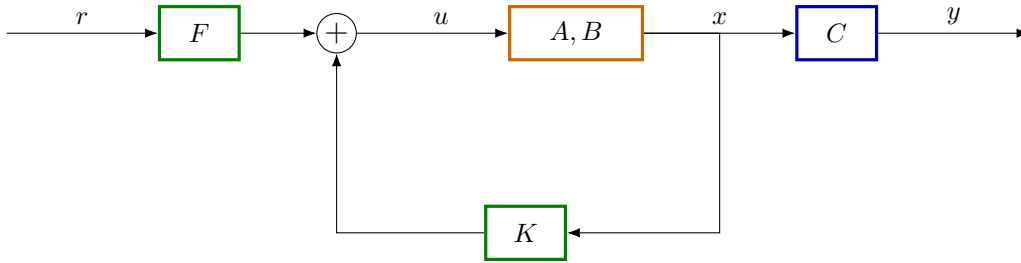


Figure 6: Diagrama en bloques de realimentación de estados con precompensación para seguimiento de referencias

4.1.3 Mediciones y Simulaciones

En primer lugar, en esta sección, se evaluó el desempeño del controlador diseñado para estabilizar la planta en torno al punto de equilibrio. El análisis incluyó tanto simulaciones como mediciones experimentales, con el objetivo de verificar su efectividad frente a perturbaciones externas.

Para llevar a cabo este análisis, se aplicó una perturbación tipo impulso a la planta y se registró la respuesta dinámica de todas las variables de estado. Las variables consideradas incluyen las posiciones angulares θ y ϕ , así como sus respectivas velocidades angulares $\dot{\theta}$ y $\dot{\phi}$. Estas mediciones permiten evaluar no solo la capacidad del controlador para retornar al equilibrio, sino también la rapidez y calidad de la respuesta en términos de tiempo de asentamiento, sobreimpulso y oscilaciones.

Los resultados obtenidos para la respuesta de las variables asociadas a θ , $\dot{\theta}$, ϕ y $\dot{\phi}$ se presentan en la Fig. 7. En todos los casos, se puede observar el efecto de la perturbación inicial, seguido de un retorno al estado de equilibrio gracias a la acción del controlador.

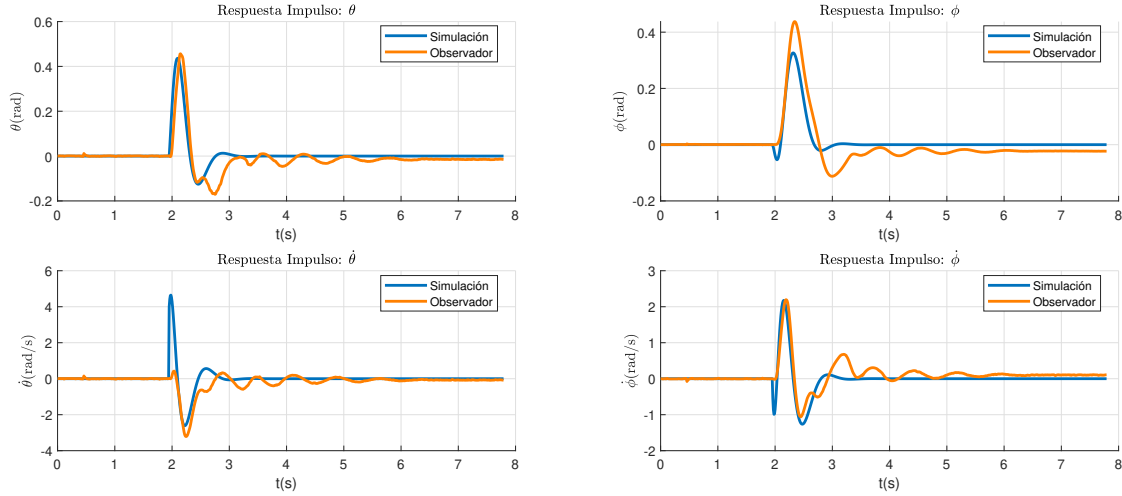


Figure 7: Respuesta ante perturbación tipo impulso del controlador diseñado.

Los gráficos evidencian que el controlador diseñado logra estabilizar el sistema de manera efectiva tras la perturbación aplicada al ángulo del péndulo. Además, se observa una notable coherencia entre el comportamiento del sistema simulado y el de la planta controlada en las pruebas experimentales, lo que valida cierta precisión del modelo de la planta y la implementación del controlador en condiciones reales.

En segundo lugar, se simuló y midió la respuesta del sistema ante una secuencia de comandos tipo escalón en ϕ manteniendo la referencia de θ en 0, con el objetivo de verificar la capacidad del sistema precompensado para seguir referencias. Los resultados se presentan en las Fig. 8 y Fig. 9, donde se observa el desempeño del sistema en términos del seguimiento de las referencias en las variables θ y ϕ . En ambas figuras, se comparan los valores obtenidos mediante simulación y los observados con el sistema implementado.

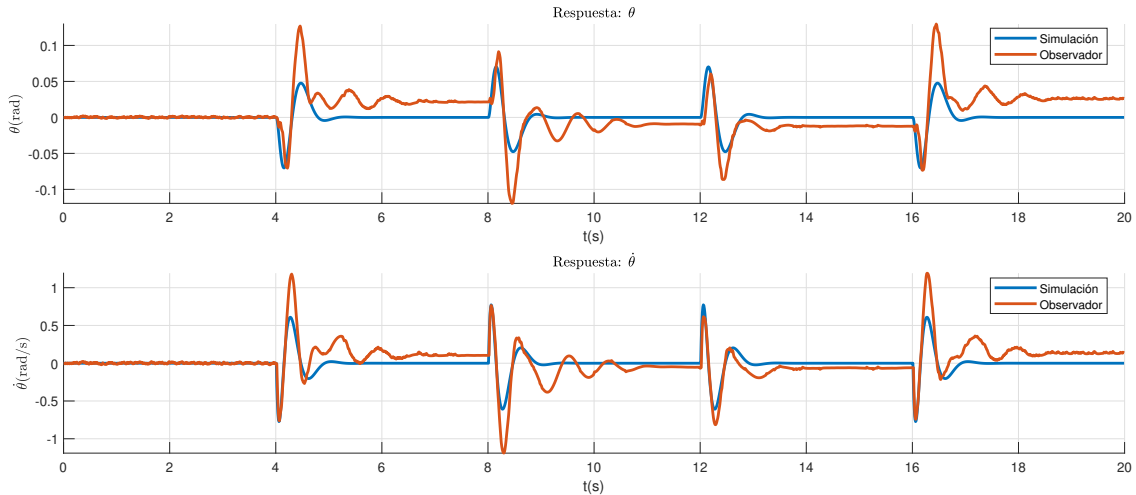


Figure 8: Respuesta ante secuencia de escalones de θ y $\dot{\theta}$ del sistema controlado y precompensado.

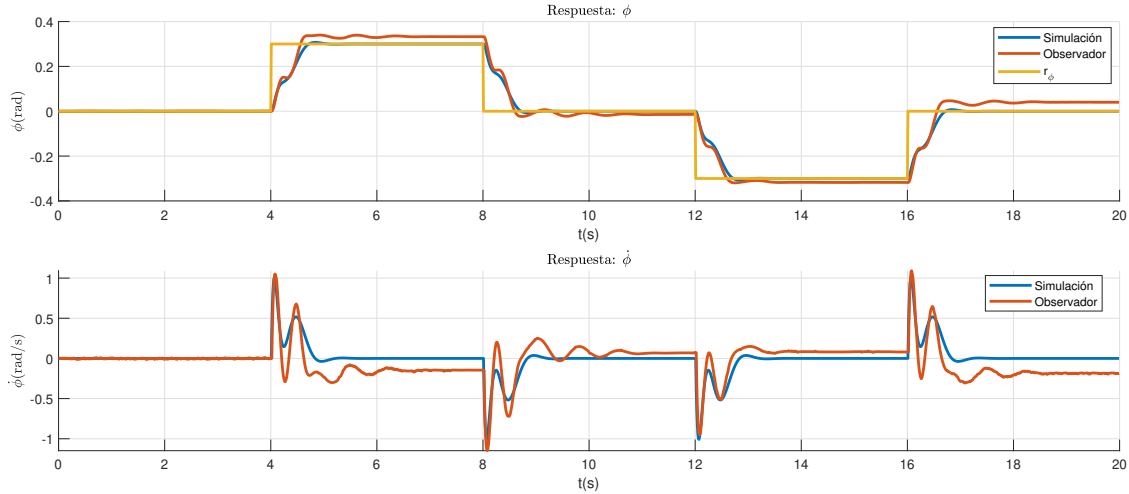


Figure 9: Respuesta ante secuencia de escalones de ϕ y $\dot{\phi}$ del sistema controlado y precompensado.

Se observa que la precompensación no logra eliminar el error en estado estacionario durante el seguimiento de referencia. Esto se debe a la ausencia de acción integral en el lazo de control y al modelo del servomotor, que también presenta error en estado estacionario al aplicarle referencias. Además, utilizar una precompensación no trae consigo las ventajas de una acción integral como el rechazo de perturbaciones. No obstante, la precompensación representa una solución rápida y fácil de implementar cuando no se desea agregar acción integral al controlador, evitando así una mayor complejidad en su diseño.

4.2 Controlador por Estados con Acción Integral

4.2.1 Diseño

Se rediseñó el controlador por estados agregando una acción integral para lograr un seguimiento robusto de la referencia de ϕ . Esta solución permite alcanzar un error nulo en estado estacionario para referencias tipo escalón y, además, rechazar perturbaciones del mismo tipo. La implementación comienza añadiendo al sistema tantas variables como referencias se deseen seguir; en este caso, únicamente la variable ϕ . A este estado aumentado se le denomina q , definido de manera que su derivada sea el error entre la salida y la referencia correspondiente:

$$\dot{q} = \phi - r_\phi.$$

A continuación, se reescribe el sistema original incorporando la nueva variable de estado aumentado:

$$\dot{\mathbf{x}}(t) = \begin{bmatrix} \dot{\mathbf{x}}(t) \\ \dot{q}(t) \end{bmatrix} = \begin{bmatrix} A & 0 \\ -C & 0 \end{bmatrix} \cdot \begin{bmatrix} \mathbf{x}(t) \\ q(t) \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} \cdot u(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \cdot r(t).$$

Una vez encontrada la dinámica del sistema en tiempo continuo, se discretiza utilizando la técnica de **Zero-Order Hold** con un tiempo de muestreo $t_s = 10$ ms. Posteriormente, se realiza la realimentación por estados. En la Fig.10 se presenta el diagrama de simulación correspondiente, que ilustra su implementación.

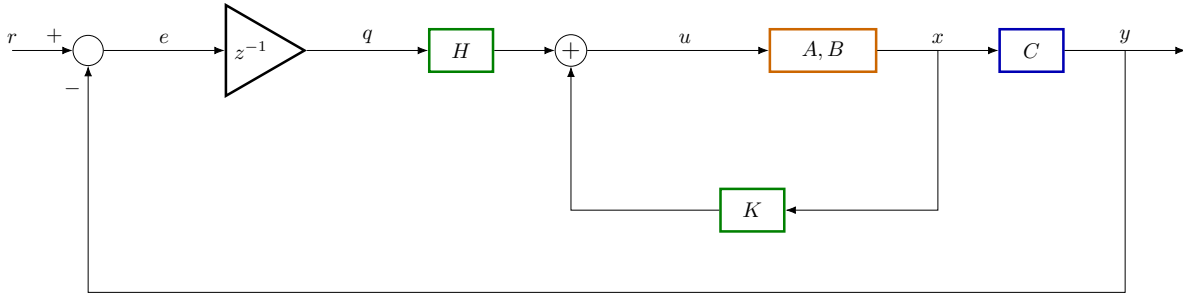


Figure 10: Diagrama en bloques de realimentación de estados con acción integral para seguimiento de referencias

El sistema discretizado realimentado queda expresado como:

$$\dot{\mathbf{x}}(k+1) = \left(\begin{bmatrix} A_d & 0 \\ -C_d \cdot t_s & I \end{bmatrix} + \begin{bmatrix} B_d \\ 0 \end{bmatrix} \cdot [K \ H] \right) \cdot \begin{bmatrix} \mathbf{x}(k) \\ q(k) \end{bmatrix} + \begin{bmatrix} 0 \\ t_s \end{bmatrix} \cdot r(k),$$

donde K corresponde a la submatriz de realimentación de estados $\mathbf{x}(k)$, y H a la del estado aumentado $q(k)$. Así, la entrada al sistema se define como:

$$u(k) = [K \ H] \cdot \begin{bmatrix} \mathbf{x}(k) \\ q(k) \end{bmatrix}.$$

La matriz de realimentación de estados se calculó utilizando el comando `PLACE()` de MATLAB, indicando los polos a lazo cerrado deseados en el orden correspondiente a la variable de estado a la que se le quiere asignar los polos. En cuanto a estos últimos, para su selección se utilizó como base los del controlador anterior y se los modificó de forma tal de obtener un tiempo de establecimiento de 1.5 segundos o menor y un sobre-pico menor al 15 %. Los polos a lazo cerrado de tiempo continuo utilizados para la realimentación son:

$$p_c = [-6 - 5i \quad -6 + 5i \quad -8 \quad -8 \quad -15]$$

y sus equivalente discretos obtenidos a partir de la ecuación del **zero-order hold**:

$$p_d = [0.9406 - 0.0471i \quad 0.9406 + 0.0471i \quad 0.9231 \quad 0.9231 \quad 0.8607]$$

Las submatrices de realimentación calculadas que garantizan las ubicaciones deseadas de los polos son las siguientes:

$$K = [0.6924 \quad -0.1675 \quad -0.6269 \quad -0.1681] \quad \wedge \quad H = 2.9629$$

4.2.2 Mediciones y Simulaciones

Al igual que en el caso del controlador anterior, se evaluó el comportamiento del controlador con acción integral diseñado para estabilizar la planta en torno al punto de equilibrio y seguir referencias tipo escalón de ϕ . Para ello, se simuló un impulso aplicando un golpe al péndulo, seguido del envío de una secuencia de referencias en forma de escalón r_ϕ . En la Fig. 11 se presentan los resultados obtenidos frente al impulso, mientras que en las Fig. 12 y Fig. 13 se muestran los resultados de la respuesta ante una secuencia de escalones, superponiendo en un mismo gráfico los datos simulados y los obtenidos de la planta real.

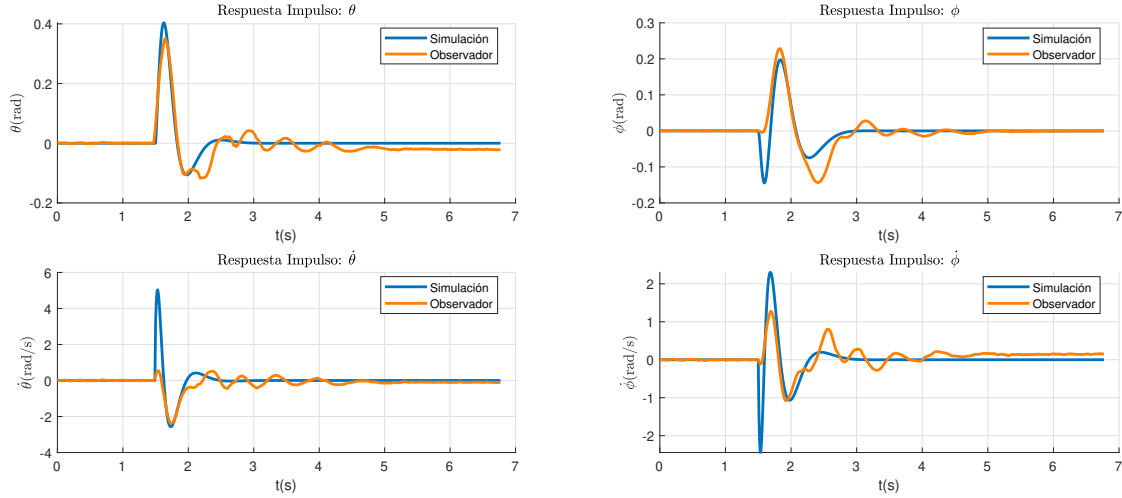


Figure 11: Respuesta ante perturbación tipo impulso del controlador diseñado.

En primer lugar, se observa que el comportamiento de la planta controlada en el sistema real es consistente con las simulaciones realizadas. Los resultados muestran que el controlador diseñado no solo estabiliza la planta alrededor del punto de equilibrio, sino que también garantiza un seguimiento robusto de la referencia.

Este desempeño se refleja en la obtención de error nulo en estado estacionario, atribuible a la acción integral incorporada en el diseño. En comparación con la precompensación, esta solución resulta significativamente más efectiva.

Finalmente, se confirma que el controlador cumple con los requisitos de diseño especificados, tanto en términos de tiempo de establecimiento como de sobre-pico.

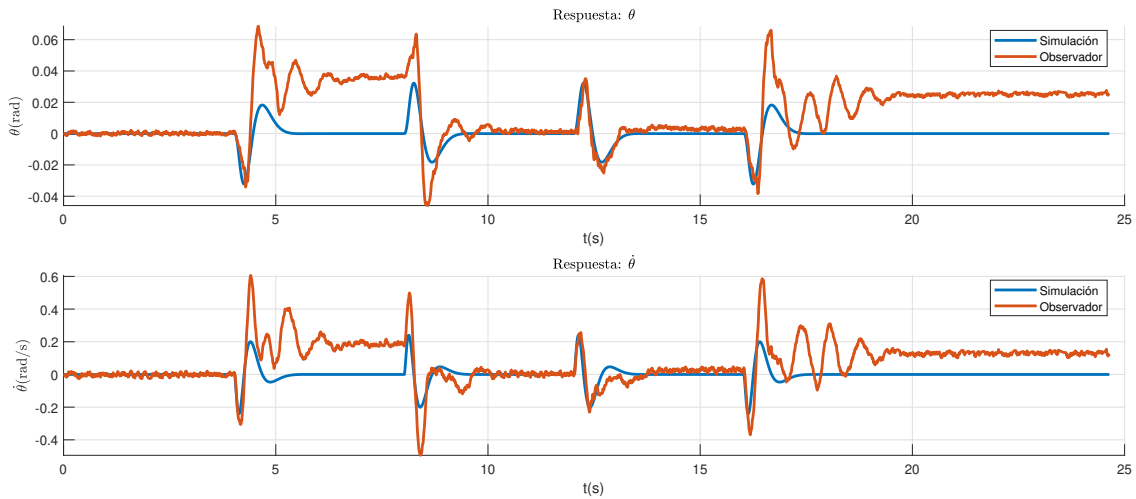


Figure 12: Respuesta ante secuencia de escalones de θ y $\dot{\theta}$ del sistema controlado y precompensado.

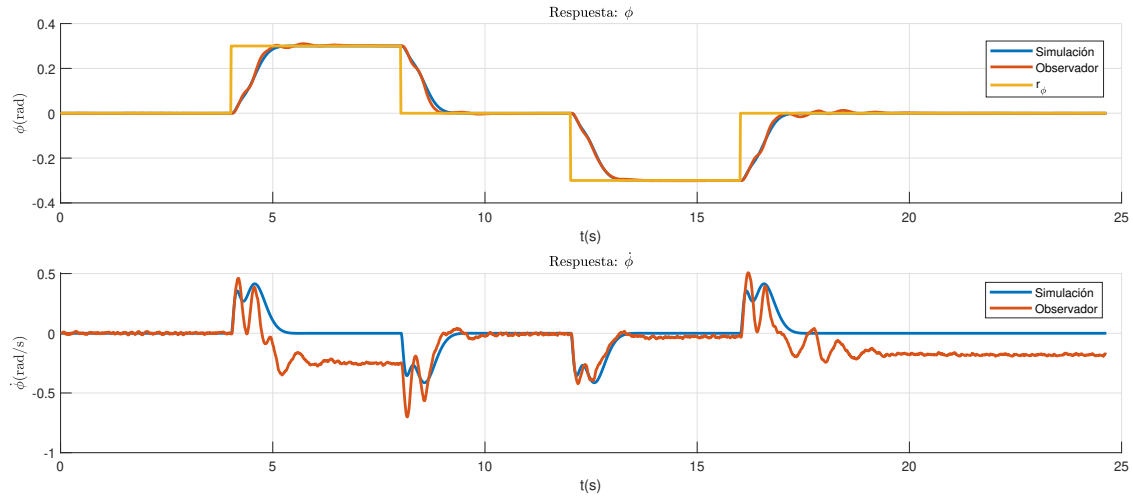


Figure 13: Respuesta ante secuencia de escalones de ϕ y $\dot{\phi}$ del sistema controlado y precompensado.

5 Conclusión

En este trabajo se logró implementar un observador con un desempeño satisfactorio para la estimación de las variables de estado del sistema. Estas estimaciones permitieron desarrollar y probar dos esquemas de control exitosos: uno basado en realimentación de estados con precompensación y otro con acción integral.

El controlador sin acción integral, complementado con precompensación, permitió un seguimiento adecuado de referencias, pero con limitaciones dado el modelo imperfecto de la planta. Por otro lado, el controlador con acción integral demostró ser superior en términos de seguimiento robusto de referencias y rechazo de perturbaciones, logrando un mejor desempeño en aplicaciones donde estas características son críticas.

Ambos controladores validaron el modelo y las técnicas de diseño utilizadas, destacando el potencial de los esquemas de control por realimentación de estados para abordar problemas de estabilización y seguimiento en sistemas complejos y con múltiples salidas como el Péndulo de Furuta.