Universidad de Buenos Aires
Facultad de Ingeniería
Año 2025 - 2do cuatrimestre

Taller de Automatización y Control (TA135)

Trabajo Práctico N^{o}2

Gabriel José Vigo Abad 107980
gvigo@fi.uba.ar
Manuel Hirsch 110221
mhirsch@fi.uba.ar

Índice

1. Introducción **2**

2. Descripción física y dinámica **2**
2.1. Dinámica del brazo actuado o principal 2
2.2. Dinámica del péndulo 3

3. Calibración del potenciómetro **4**
3.1. Método de calibración 4

4. Ajuste de límites PWM del servomotor **5**

5. Identificación de la planta **5**
5.1. Obtención de datos experimentales 5
5.2. Modelo caja gris 6
5.3. Adquisición de datos en Arduino 6
5.4. Procesamiento y ajuste del modelo en MATLAB 6
5.5. Planta estimada 7

6. Control proporcional **9**
6.1. Planteo del control proporcional 9
6.2. Análisis mediante Root Locus 9
6.3. Simulación y validación experimental 11
6.4. Conclusiones del control proporcional 11

7. Control proporcional–derivativo (PD) **11**
7.1. Criterio de diseño 12
7.2. Implementación experimental 12
7.3. Observaciones finales 13

8. Control proporcional–integral (PI) **13**
8.1. Criterio de diseño 13
8.2. Implementación experimental 13
8.3. Análisis del fenómeno y compensación 13
8.4. Resultados y conclusiones 14

9. Conclusión **14**

1. Introducción

La planta utilizada en el presente trabajo práctico corresponde a un péndulo rotativo o *Rotary Inverted Pendulum* (RIP), también conocido como *péndulo de Furuta*. Este sistema se caracteriza por presentar una base giratoria accionada por un servomotor, sobre la cual se monta un brazo que permite el movimiento angular del péndulo en el plano vertical. En este caso, la base es controlada mediante un servomotor MG996R alimentado con 5V, mientras que el ángulo del brazo y del péndulo son medidos respectivamente por un potenciómetro lineal de 10 kΩ y una IMU MPU6050. Vale la pena aclarar que el control se realizará mediante un Arduino UNO, haciendo uso de todas las librerias necesarias para tal fin.

El conjunto se ensambla utilizando piezas impresas en 3D, rulemanes, y una estructura modular que permite una configuración tanto de péndulo normal (estable) como invertido (inestable). Para el presente trabajo, se analiza el **péndulo normal** Figura 1, es decir, con la barra colgando hacia abajo en su posición de equilibrio estable.

Este trabajo tiene puntualmente el objetivo de identificar la planta pertinente, y de realizar el control de la misma mediante varios tipos de controladores, analizando sus efectos.

2. Descripción física y dinámica

El sistema puede considerarse compuesto por dos cuerpos rígidos: el brazo actuado y el péndulo. El brazo posee un ángulo ϕ respecto al eje horizontal, comandado por el servomotor, mientras que el péndulo posee un ángulo θ medido desde la vertical. La acción de control se ejerce a través del torque generado por el servo, el cual se transmite al brazo y afecta indirectamente la posición del péndulo.

El objetivo es comprender el origen físico de cada término presente en las ecuaciones de movimiento y cómo estos representan los acoplamientos entre el brazo actuado y el péndulo.

Definimos las siguientes variables y parámetros del sistema:

- ϕ: ángulo del brazo actuado o principal, medido en el plano horizontal respecto a una referencia fija.
- θ: ángulo del péndulo medido desde la vertical (posición estable hacia abajo).
- u: par motor aplicado por el servomotor al eje del brazo.
- J_b: momento de inercia del brazo respecto de su eje de rotación.
- J_p: momento de inercia del péndulo respecto de su bisagra.
- m: masa total del péndulo.
- L_b: longitud efectiva del brazo hasta la bisagra del péndulo.
- l_p: distancia desde la bisagra del péndulo hasta su centro de masa.
- g: aceleración de la gravedad.

2.1. Dinámica del brazo actuado o principal

Aplicando el teorema del momento angular sobre el eje del servo, la suma de momentos es igual a la derivada temporal del momento angular del conjunto brazo+péndulo visto desde dicho eje.

$$\sum M_{\text{servo}} = \frac{d}{dt} H_{\text{brazo+péndulo}}. \tag{1}$$

Los momentos relevantes son:

1. Inercia del brazo: genera un momento $J_b \ddot{\phi}$.
2. Carga del péndulo en el extremo del brazo: su centro de masa se mueve con la rotación del brazo, aportando $m L_b^2 \ddot{\phi}$.
3. Acople inercial con el péndulo: al rotar el péndulo, su centro de masa se desplaza respecto al eje del brazo, generando un momento adicional $m L_b l_p \cos\theta \, \ddot{\theta}$.

4. Efecto centrífugo: el movimiento angular del péndulo produce un término $-\,mL_b l_p \sin\theta\,\dot{\theta}^2$.

El balance total de momentos sobre el eje del brazo queda entonces:

$$\boxed{(J_b + mL_b^2)\,\ddot{\phi} \;+\; mL_b l_p\,\cos(\theta)\,\ddot{\theta} \;-\; mL_b l_p\,\sin(\theta)\,\dot{\theta}^2 \;=\; u.} \tag{2}$$

- $(J_b + mL_b^2)\ddot{\phi}$ representa la inercia combinada del brazo y del péndulo "visto" como masa puntual en su extremo.
- $mL_b l_p \cos\theta\,\ddot{\theta}$ es el acople dinámico entre la aceleración del péndulo y la del brazo.
- $-mL_b l_p \sin\theta\,\dot{\theta}^2$ es el término centrífugo que tiende a abrir el ángulo cuando el péndulo gira rápidamente.

2.2. Dinámica del péndulo

El balance de momentos sobre la bisagra del péndulo considera:

- Su propia inercia $(J_p + ml_p^2)\ddot{\theta}$.
- El acople con la aceleración del brazo $mL_b l_p \cos\theta\,\ddot{\phi}$.
- El momento gravitatorio restaurador $mgl_p \sin\theta$.

De este modo, la ecuación de equilibrio dinámico del péndulo resulta:

$$\boxed{(J_p + ml_p^2)\,\ddot{\theta} \;+\; mL_b l_p\,\cos\theta\,\ddot{\phi} \;+\; mgl_p\,\sin\theta \;=\; 0.} \tag{3}$$

- $(J_p + ml_p^2)\ddot{\theta}$ es la inercia total del péndulo respecto a su bisagra.
- $mL_b l_p \cos\theta\,\ddot{\phi}$ indica cómo la aceleración del brazo "sacude" al péndulo.
- $mgl_p \sin\theta$ representa el torque gravitatorio que busca restablecer el equilibrio vertical.

Combinando ambos balances obtenemos el sistema no lineal que describe el movimiento acoplado del brazo y el péndulo:

$$\begin{aligned}(J_b + mL_b^2)\,\ddot{\phi} + mL_b l_p\,\cos\theta\,\ddot{\theta} - mL_b l_p\,\sin\theta\,\dot{\theta}^2 &= u,\\ (J_p + ml_p^2)\,\ddot{\theta} + mL_b l_p\,\cos\theta\,\ddot{\phi} + mgl_p\,\sin\theta &= 0.\end{aligned}$$

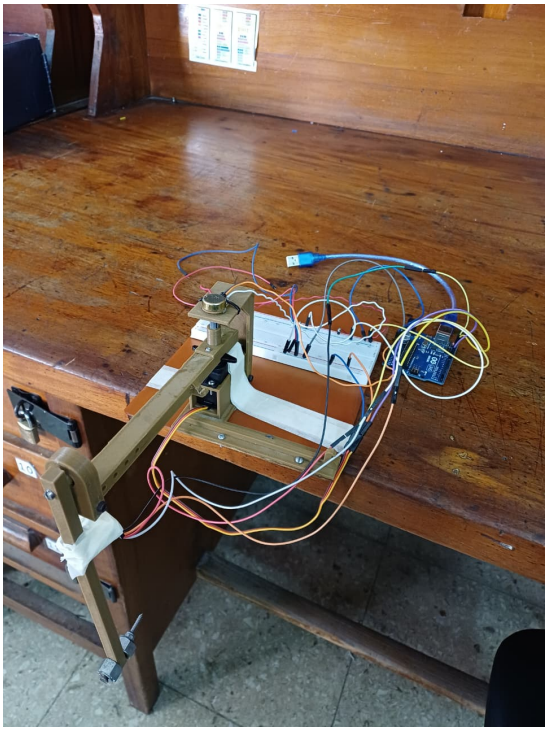

Figura 1: Péndulo rotativo en configuración normal.

A partir de este modelo acoplado, es posible unificar las ecuaciones en una única ecuación diferencial que relacione directamente la salida del sistema, el ángulo del péndulo $\theta(t)$, con la entrada de control aplicada al servomotor $u(t)$.

Dado que el conjunto brazo–péndulo–actuador constituye una dinámica de cuarto orden, la ecuación resultante puede expresarse de manera general como:

$$a_4\,\dddot{\theta}(t) + a_3\,\ddot{\theta}(t) + a_2\,\ddot{\theta}(t) + a_1\,\dot{\theta}(t) + a_0\,\theta(t) = b_3\,\ddot{u}(t) + b_2\,\ddot{u}(t) + b_1\,\dot{u}(t) + b_0\,u(t), \tag{4}$$

donde los coeficientes a_i y b_i representan los parámetros efectivos del sistema una vez incorporadas las inerciales del brazo y del péndulo, el acople entre ambos y la dinámica propia del actuador.
Esta forma general resulta conveniente para la etapa de identificación experimental, ya que permite ajustar los coeficientes directamente a partir de los datos medidos de entrada y salida, sin necesidad de resolver analíticamente las ecuaciones acopladas, que en este caso, son bastante engorrosas.

3. Calibración del potenciómetro

Antes de comenzar con el modelado y la identificación de la planta, fue necesario calibrar el potenciómetro que mide el ángulo del brazo principal ϕ. Este elemento, de tipo lineal de 10 kΩ, se encuentra acoplado mecánicamente al eje del servo mediante una perilla impresa en 3D, y entrega una tensión analógica proporcional al ángulo mecánico.

3.1. Método de calibración

La calibración consistió en registrar pares de valores (ADC, ϕ), donde:

- ADC: valor digital leído por el conversor analógico–digital (0–1023) del Arduino.
- ϕ: ángulo físico del brazo medido manualmente en grados con respecto a la posición central.

Los datos fueron obtenidos girando el brazo desde un extremo a otro de su recorrido útil (aproximadamente ±90°), y registrando la lectura correspondiente del potenciómetro para distintos puntos intermedios.

A partir de la relación experimental, se ajustó una función lineal del tipo:

$$\phi(\text{ADC}) = A_{\text{cal}} \cdot \text{ADC} + B_{\text{cal}}, \tag{5}$$

donde A_{cal} representa la pendiente (grados por unidad ADC) y B_{cal} el offset angular correspondiente al punto medio.

El ajuste lineal mostró una correlación prácticamente perfecta, confirmando el comportamiento lineal del potenciómetro en el rango de interés. En la implementación final se emplearon los coeficientes:

$$A_{\text{cal}} = -0{,}261627 \qquad B_{\text{cal}} = 251{,}16192$$

permitiendo obtener la conversión directa:

$$\phi[^\circ] = -0{,}261627 \cdot \text{ADC} + 251{,}16192.$$

De esta forma, la señal analógica del potenciómetro se transforma en un ángulo absoluto que sirve tanto para el control del servo como para la identificación de la dinámica de la planta.

4. Ajuste de límites PWM del servomotor

En el código de control se establecieron los siguientes valores para definir los límites de operación del servomotor:

```
const int PWM_MIN = 700;
const int PWM_MAX = 2300;
const int SERVO_CENTER_DEG = 90;
```

Estos parámetros determinan la conversión entre el ancho del pulso PWM (en microsegundos) y el ángulo mecánico del eje del servo. En particular:

- `PWM_MIN` y `PWM_MAX` establecen los pulsos mínimos y máximos enviados, que corresponden aproximadamente a los extremos de recorrido angular.
- `SERVO_CENTER_DEG` define el punto medio de operación, que se asume en 90°.

En esta implementación los valores se adoptaron de manera arbitraria, dentro del rango típico aceptado para el servo MG996R (700–2300 µs). Sin embargo, para lograr una calibración más precisa y simétrica respecto del centro real del servo, podría determinarse experimentalmente el pulso correspondiente al ángulo de 90° y ajustar los límites de forma equidistante a partir de ese valor, evitando saturaciones o topes mecánicos.
En resumen, los límites empleados son adecuados para una primera puesta en marcha, pero podrían optimizarse mediante una medición directa del punto central y del rango seguro de movimiento del brazo.

5. Identificación de la planta

Una vez calibrado el sistema de medición, se procedió a la etapa de identificación. Dado que el modelo teórico resultante del análisis dinámico posee parámetros difíciles de determinar con precisión (momento de inercia, fricción, masas efectivas), se optó por una estrategia de **modelado caja gris** (*grey-box modeling*), combinando conocimiento físico y datos experimentales.

5.1. Obtención de datos experimentales

Las señales de entrada y salida fueron adquiridas mediante el entorno Arduino–Matlab. Se aplicó al brazo principal una excitación angular de referencia de 45° en torno a su punto de equilibrio. El ángulo del brazo $\phi(t)$ se obtuvo a partir del potenciómetro calibrado, mientras que el ángulo del péndulo $\theta(t)$ fue medido mediante la IMU MPU6050. Para reducir ruido y compensar el desfase entre el acelerómetro y el giróscopo, se implementó un *filtro complementario*, definido como:

$$\theta_{\text{filt}}(t) = \alpha\big[\theta_{\text{gyro}}(t)\big] + (1-\alpha)\big[\theta_{\text{acc}}(t)\big] = 0{,}98 \cdot \big[\theta_{\text{gyro}}(t)\big] + 0{,}02 \cdot \big[\theta_{\text{acc}}(t)\big], \tag{6}$$

con $\alpha = 0{,}98$, proporcionando una lectura suave y estable.

5.2. Modelo caja gris

Con el fin de obtener un modelo dinámico representativo de la planta real, se realizaron mediciones experimentales empleando un montaje basado en Arduino y MATLAB. El objetivo fue estimar los coeficientes del modelo lineal de cuarto orden propuesto, que describe la relación entre la señal de control aplicada al servomotor y el ángulo del péndulo medido mediante la IMU:

$$a_4\,\ddddot{\theta}(t) + a_3\,\dddot{\theta}(t) + a_2\,\ddot{\theta}(t) + a_1\,\dot{\theta}(t) + a_0\,\theta(t) = b_3\,\dddot{u}(t) + b_2\,\ddot{u}(t) + b_1\,\dot{u}(t) + b_0\,u(t), \tag{7}$$

la cual representa un sistema lineal de cuarto orden, cuyos parámetros a_i y b_i se identifican a partir de los datos medidos.

5.3. Adquisición de datos en Arduino

Para la adquisición experimental se desarrolló un programa en Arduino que ejecuta una señal de escalón angular sobre el brazo principal, mientras registra el ángulo de respuesta del péndulo.

- La señal de entrada $u(t)$ se generó mediante un cambio de referencia de 45° sobre el servo centrado en 90°.
- El ángulo del brazo ϕ se obtuvo mediante un potenciómetro calibrado según los coeficientes experimentales A_{cal} y B_{cal}.
- El ángulo del péndulo θ se midió con el sensor MPU6050, aplicando el filtro complementario para combinar la información del acelerómetro y el giróscopo.

El código se estructuró para enviar por puerto serie, a 115200 baudios, las tres variables registradas: tiempo, entrada y salida (t_s, ϕ_{ref}, θ_{filt}). Cada registro se transmitió en formato CSV para su posterior procesamiento en MATLAB.

Este método permite capturar las respuestas dinámicas del sistema ante excitaciones simples, garantizando una tasa de muestreo de 100 Hz (delay de 10 ms). Se hace a esta frecuencia para tener una resolución suficiente en tiempo para luego hacer el control a 50 Hz.

5.4. Procesamiento y ajuste del modelo en MATLAB

Los datos enviados por Arduino se recibieron y procesaron en MATLAB mediante comunicación serial. El script correspondiente realiza los siguientes pasos:

1. Configuración del puerto y lectura de los datos durante 15 s.
2. Recorte del segmento de interés (a partir de los 5 s) para eliminar el transitorio inicial que lleva el brazo al centro.
3. Construcción de un objeto de identificación `iddata(y,u,Ts)` con periodo de muestreo $T_s = 0{,}01$ s. `iddata(y,u,Ts)` es una estructura de datos especializada de MATLAB que se usa para almacenar de forma organizada las señales de entrada y salida de un sistema dinámico, junto con el tiempo de muestreo. Su propósito es que todos los algoritmos de identificación (en este caso, `tfest`) trabajen con un mismo formato estándar.
4. Estimación de una función de transferencia de 4 polos y 3 ceros mediante `tfest`. La función `tfest` estima una función de transferencia continua o discreta a partir de los datos experimentales contenidos en el objeto `iddata`. El procedimiento puede resumirse en las siguientes etapas:

   a) Lectura de los datos. Se extraen las señales de entrada $u(t)$, salida $y(t)$ y el tiempo de muestreo T_s. Si los datos son discretos, el algoritmo puede interpolarlos internamente para operar en el dominio continuo, salvo que se indique lo contrario.

   b) Definición de la estructura del modelo. Se asume una función de transferencia del tipo:

$$P(s) = \frac{b_0 s^{n_z} + b_1 s^{n_z-1} + \cdots + b_{n_z}}{s^{n_p} + a_1 s^{n_p-1} + \cdots + a_{n_p}}, \tag{8}$$

   donde n_z es el número de ceros y n_p el número de polos especificados por el usuario. El algoritmo busca los coeficientes a_i y b_i que mejor ajustan el modelo a los datos medidos.

c) Simulación de la salida del sistema. Con una estimación inicial de los parámetros (obtenida por mínimos cuadrados o predicción instrumental), se calcula la respuesta simulada $\hat{y}(t)$ que produciría el modelo ante la entrada $u(t)$.

d) Minimización del error de predicción. Se calcula el error $e(t) = y(t) - \hat{y}(t)$ y se ajustan los coeficientes para minimizar la suma cuadrática:

$$J = \sum_t e(t)^2. \tag{9}$$

Este proceso iterativo se basa en algoritmos de optimización no lineal, como el método de Levenberg–Marquardt, hasta alcanzar la mejor correspondencia entre datos medidos y simulados.

e) Generación del modelo resultante. Finalmente, `tfest` devuelve un objeto de tipo `idtf`, que contiene:

- Los coeficientes del numerador y denominador de $P(s)$.
- El tipo de modelo (continuo o discreto).
- Información del porcentaje de ajuste al conjunto de datos (*Fit to estimation data*).
- Los polos, ceros y funciones de respuesta temporal y frecuencial del modelo estimado.

5. Validación del modelo con la función `compare`, que contrasta la respuesta simulada con la experimental. La figura resultante permite evaluar visualmente la calidad del ajuste y verificar si el modelo obtenido reproduce fielmente la dinámica observada.

5.5. Planta estimada

El modelo identificado permitió obtener la función de transferencia $\frac{\Theta(s)}{U(s)}$ que aproxima la dinámica experimental de la mejor manera, como se puede ver en la Ec. 10.

$$\frac{\Theta(s)}{U(s)} = \frac{-107{,}2 * s^3 - 157{,}7 * s^2 - 1413 * s - 107{,}5}{s^4 + 117{,}6 * s^3 + 494{,}9 * s^2 + 1{,}296e4 * s + 1{,}808e4} \tag{10}$$

En la Figura 2, se pueden observar las mediciones de la respuesta al impulso de θ a lo largo del tiempo, comparado con la respuesta estimada utilizando el modelo de la planta obtenida previamente. Como podemos ver en la Figura, se tiene un $\approx 88\,\%$ de precisión, pudiéndose concluir una estimación exitosa, sobretodo observando que se tiene un muy alto nivel de correlación en las primeras 2 oscilaciones.

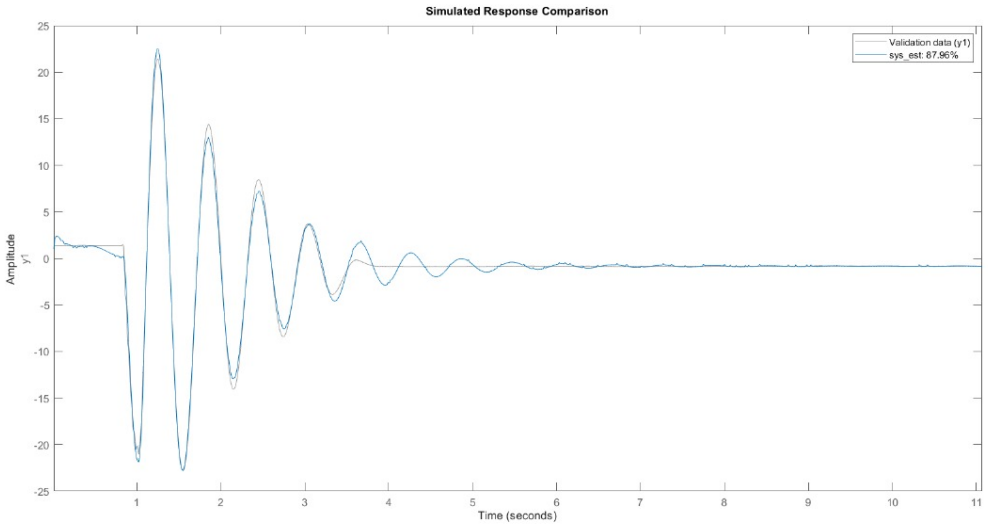

Figura 2: Respuesta al escalón del sistema real y del estimado.

Además de la identificación del sistema, podemos obtener los parámetros característicos de la respuesta de la planta, como son el tiempo de establecimiento t_s, rise-time t_r y el *Overshoot* o sobrepico inicial.

Para el primero de estos, usando la convención del $\pm 5\,\%$, se obtiene un $t_s \approx 5,5$s, para el rise-time se obtiene $t_r \approx 0{,}3$s y un *Overshoot* del $22\,\%$. Se remarca que en este caso, sin ningún tipo de controlador, se tienen aproximadamente 7 oscilaciones.

A continuación se realizó el diagrama de Bode de la planta obtenida en la identificación (Figura 3), para visualizar su comportamiento en frecuencia.

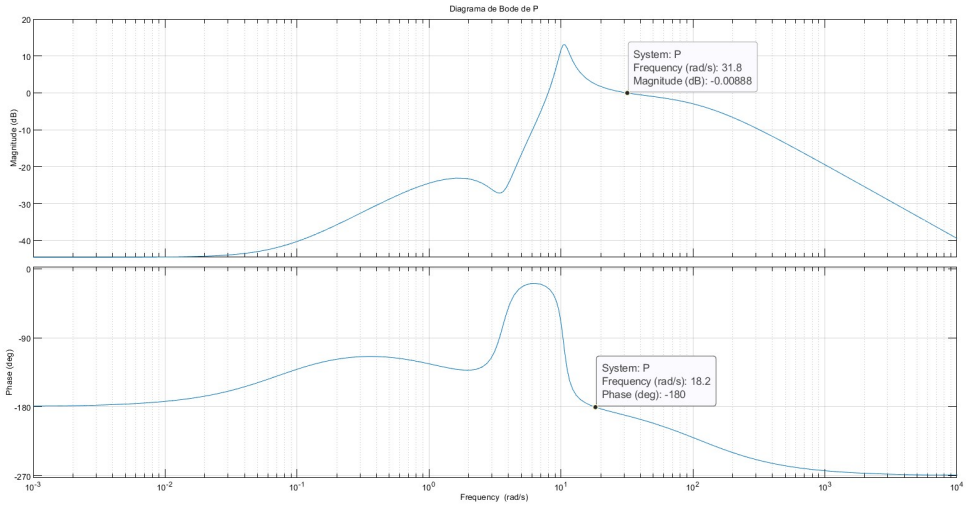

Figura 3: Diagrama de Bode de la planta P estimada.

Se denota que cuando $\omega = 18,1 \frac{\text{rad}}{\text{s}}$ la planta se vuelve inestable, frecuencia a la cual la ganancia de la misma equivale a $|P(s)| = 2,15$ dB. Es decir, es aproximadamente a una ganancia de 0.75 que la planta se vuelve inestable.

Lamentablemente, más adelante en el proceso de comparación entre simulación y realidad, la planta estimada no respondía de la misma manera frente a perturbaciones de salida que la real. Es por esto que se modificó la expresión de la transferencia de manera que se correspondan las señales simuladas con las capturadas en la realidad. Esto se realizó moviendo las singularidades para emparejar frecuencias de oscilación y ganancias. La transferencia entonces pasó a ser la siguiente:

$$P(s) = -90 \cdot \frac{s \cdot (s+8)}{(s+114)(s+1{,}44)(s^2+s+56{,}5)} \tag{11}$$

De aquí en más, se utiliza esta nueva transferencia ya que es la que mejor se ajusta antes perturbaciones de salida con la realidad.

El diagrama de Bode se observa en la Figura 4

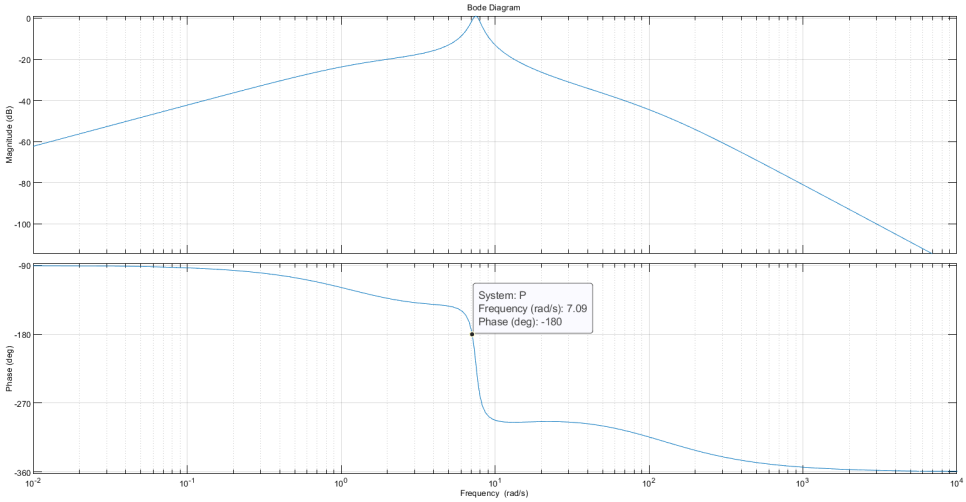

Figura 4: Diagrama de Bode de la planta P estimada luego de ajustarla manualmente.

6. Control proporcional

Una vez obtenida la transferencia identificada $P(s)$, se procedió al diseño e implementación de un controlador proporcional, cuyo objetivo fue estabilizar y amortiguar la respuesta angular del péndulo en torno a su posición de equilibrio. El control proporcional tiene una transferencia del tipo

$$C(s) = K_p \tag{12}$$

6.1. Planteo del control proporcional

El controlador proporcional se implementa como una ganancia estática K_p que amplifica el error entre la posición deseada del péndulo y su posición medida. Matemáticamente se expresa como:

$$u(t) = K_p \left(\theta_{\text{ref}} - \theta(t)\right) = K_p\, e(t), \tag{13}$$

donde $e(t)$ representa el error instantáneo de posición.

El objetivo del ajuste de K_p consiste en lograr que el sistema sea estable y que responda con un compromiso adecuado entre rapidez y amortiguamiento. Para ello se analizó el *Root Locus* del sistema, a fin de determinar los valores de ganancia que mantienen la estabilidad del lazo cerrado.

Posteriormente se seleccionó un valor de K_p que reduzca el tiempo de establecimiento y limite el sobreimpulso (*Overshoot*) del péndulo, garantizando al mismo tiempo un esfuerzo de control moderado que no sature al actuador.

6.2. Análisis mediante Root Locus

Como recientemente se dijo, para seleccionar un valor adecuado de K_p, se realizó el análisis de Root Locus sobre la planta identificada $P(s)$ empleando Matlab, como se puede ver en las figuras 5 y figura 6. Este método permitió visualizar la trayectoria de los polos de lazo cerrado a medida que K_p aumenta, identificando la región de ganancia que garantiza estabilidad y respuesta deseada.

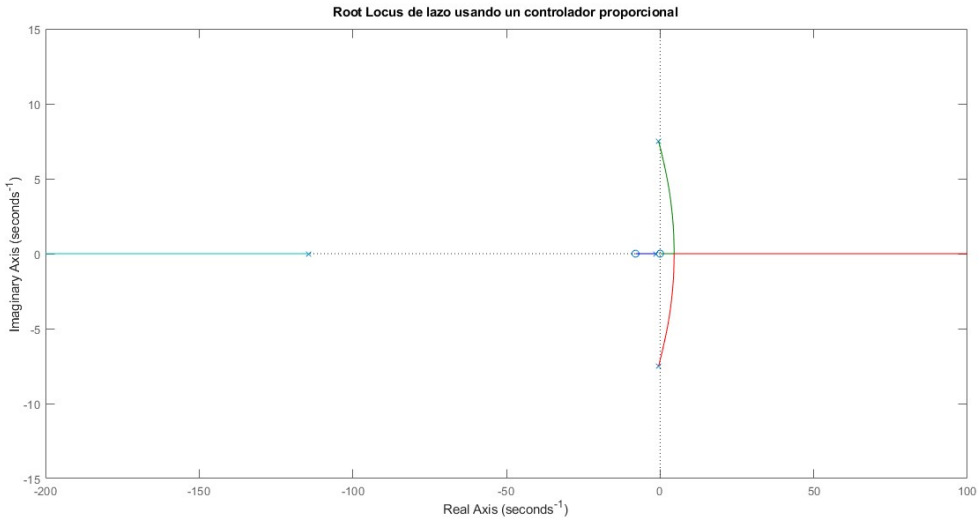

Figura 5: Root Locus completo de P.

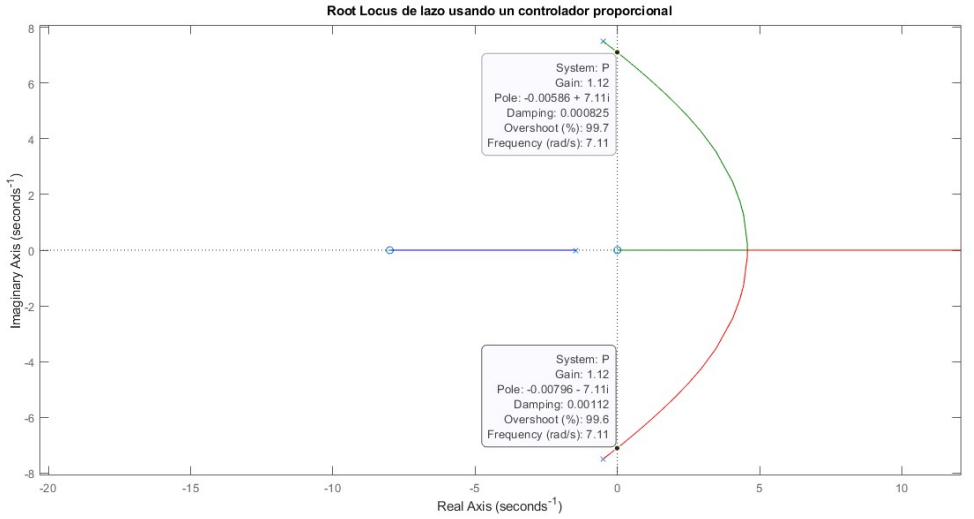

Figura 6: Root Locus para los polos complejos conjugados de P.

A partir del diagrama, se observó que:

- Para valores bajos de K_p, el sistema presenta polos (reales y complejos) negativos (estables) y lejanos al eje imaginario, con una respuesta lenta pero estable.
- Al aumentar K_p, los polos se desplazan hacia la derecha (mejorando la rapidez), pero acercandose cada vez más al eje imaginadio, lo que incrementa el sobreimpulso.
- Más allá de un umbral de ganancia crítica $K_{p,\text{crit}} = 0{,}75$, el sistema tiende a la inestabilidad, con polos que cruzan el eje imaginario.

El rango de ganancia estable se determinó experimentalmente dentro del intervalo:

$$0 < K_p < 1{,}13$$

y se seleccionó el valor que ofrecía el mejor compromiso entre rapidez y amortiguamiento. En nuestro caso, el valor adoptado fue:

$$K_p = K_{p,\text{opt}} = 0{,}7$$

Se denota que a tal frecuencia, el polo real posee una frecuencia equivalente a 1.56 rad/s, la que corresponde a la envolvente del subamortiguado. Vemos que dicho valor tiene sentido ya que se extingue en 3 segundos aproximadamente.
Es de gran importancia aclarar que, como en nuestra identificación no se contaba con un cero en el semiplano derecho, conocido por realizar una inversión de fase, se tuvo que realizar un cambio de signo en cada una de las constantes de los diversos controladores para garantizar el correcto funcionamiento del control de la planta.

6.3. Simulación y validación experimental

Una vez elegido el control proporcional se cargó el código a arduino y se le aplicó un pequeño golpe al péndulo, lo cual se considera una perturbación de salida. Es con estos resultados preliminares que decidimos hacer el ajuste manual de la planta mencionado previamente al final de la sección de identificación. Con este cambio ya hecho, se recolectaron los datos del comportamiento real con un script de Matlab y se graficó dicha respuesta junto con la respuesta al impulso de la planta teórica también calculada en Matlab. En la figura 7 se observa esta comparación. En la imagen se observa una buena concordancia entre la simulación y la realidad, salvando las diferencias de desfasaje, atribuibles a que el impulso simulado es inmediato, mientras que el real toma más tiempo. La planta real deja de oscilar más rápido que la simulación debido a la fricción que presentan los componentes físicos. Este detalle se repite en los otros tipos de controladores también.

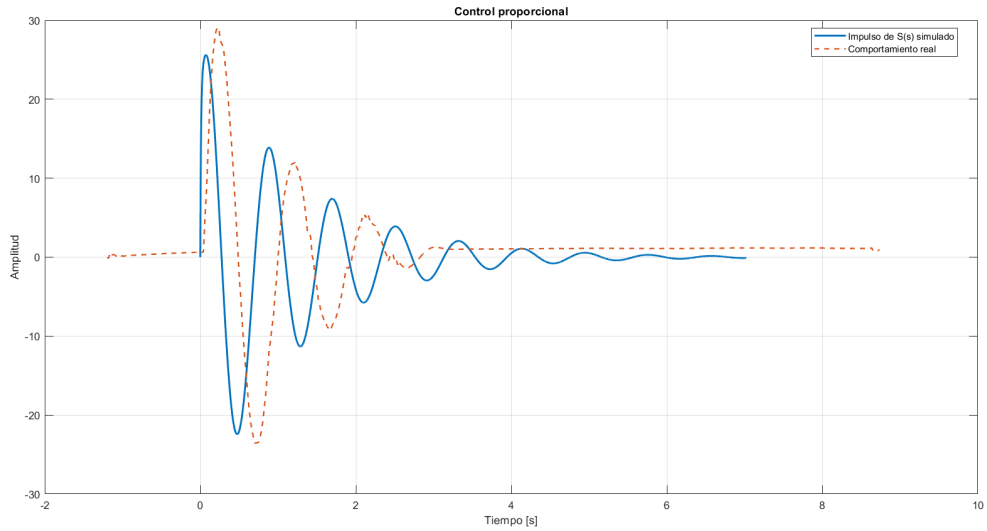

Figura 7: Respuesta temporal del sistema ante un control proporcional

6.4. Conclusiones del control proporcional

El análisis del lugar de las raíces permitió establecer un criterio sistemático para ajustar la ganancia proporcional, logrando un equilibrio entre estabilidad y desempeño. Si bien el controlador proporcional es suficiente para estabilizar el sistema en torno a su punto de equilibrio, se observó un pequeño error en régimen permanente ante perturbaciones lentas, lo que motiva la incorporación de una acción integral en la siguiente etapa.

7. Control proporcional–derivativo (PD)

En esta etapa se buscó mejorar el desempeño dinámico logrado con el control proporcional, reduciendo el tiempo de establecimiento y el sobreimpulso sin comprometer la estabilidad. Para ello se diseñó un **controlador proporcional–derivativo (PD)** de la forma:

$$C(s) = K_p + K_d s, \tag{14}$$

donde K_p corresponde a la ganancia proporcional previamente obtenida y K_d introduce una acción derivativa que anticipa los cambios en la variable controlada $\theta(t)$, generando un efecto de amortiguamiento adicional sobre el sistema.

7.1. Criterio de diseño

Según las consignas del trabajo práctico, el objetivo fue estabilizar la planta lo más rápido posible, sin saturar la acción de control en ningún momento. Por lo tanto, se inicializó K_d en un 10 % de K_p ya que las convenciones dictan que tiene que ser menor el primero que el segundo. A partir de este punto inicial se fue iterando hasta encontrar un valor que no vuelva al sistema inestable y que corrija la posición del péndulo lo más rápido posible.

Del análisis se determinó que valores pequeños de K_d apenas modifican la velocidad de respuesta, mientras que valores excesivos inducen oscilaciones y ruido en la medida de $\dot{\theta}$, producto de la amplificación de altas frecuencias. El valor óptimo adoptado fue:

$$K_d = 0{,}03,$$

ya que proporcionó un amortiguamiento adecuado y una respuesta transitoria más rápida sin exceder los límites del servo ni producir vibraciones en el péndulo.

7.2. Implementación experimental

El controlador PD fue implementado en el Arduino en forma discreta bilineal mediante las siguientes cuentas:

$$\dot{y}_f[k] = c_0\,\theta[k] + c_1\,\theta[k-1] - d_1\,\dot{y}_f[k-1]$$

$$u[k] = -K_p\,(\mathrm{REF} - \theta[k]) - K_d\,\dot{y}_f[k]$$

$$c_0 = \frac{2/T_s}{1 + 2\tau_f/T_s}, \quad c_1 = \frac{-2/T_s}{1 + 2\tau_f/T_s}, \quad d_1 = \frac{1 - 2\tau_f/T_s}{1 + 2\tau_f/T_s}$$

El experimento mostró una respuesta más rápida que con el control proporcional puro. El sistema alcanzó la posición de equilibrio en menor tiempo, con un sobreimpulso reducido y sin saturar la señal PWM del servo. Esta respuesta se observa en la figura 8. Cabe recalcar que se ven diferencias por la naturaleza real de la planta versus la perfeccción de la simulaación. Sin embargo la forma de onda y las oscilaciones coinciden bastante.

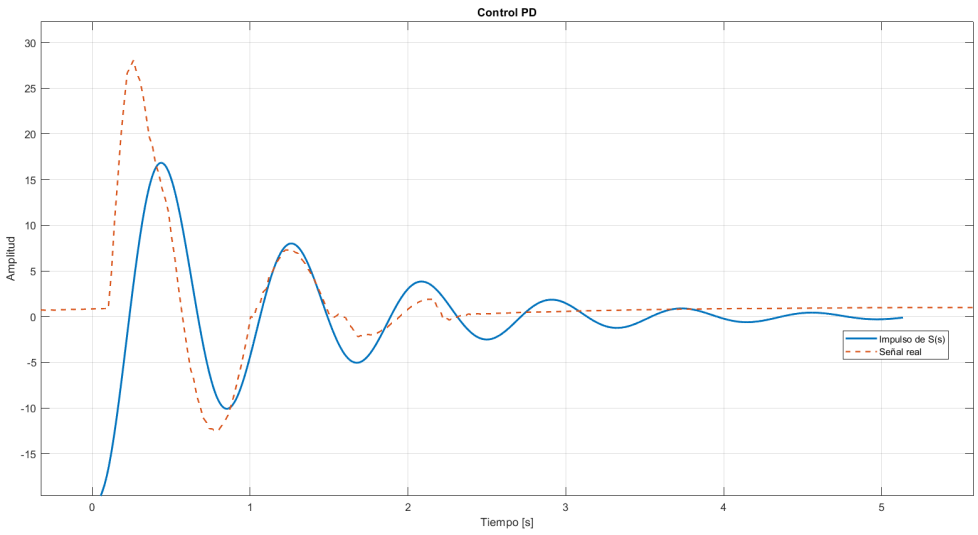

Figura 8: Respuesta del sistema utilizando un controlador PD.

7.3. Observaciones finales

El agregado de la acción derivativa demostró su efectividad para compensar las oscilaciones propias del sistema, funcionando como una forma de *amortiguamiento artificial*. Sin embargo, se observó que un valor excesivo de K_d provoca oscilaciones en el control, efecto que se buscaba evitar.

8. Control proporcional–integral (PI)

Con el objetivo de eliminar el error en régimen permanente observado con el controlador proporcional, se incorporó una acción integral, obteniendo un controlador proporcional–integral (PI) de la forma:

$$C(s) = K_p + \frac{K_i}{s}, \tag{15}$$

donde K_p es la ganancia proporcional obtenida en la etapa anterior y K_i representa la ganancia integral, encargada de acumular el error en el tiempo y corregir desplazamientos lentos del péndulo respecto a su posición de equilibrio.

8.1. Criterio de diseño

El criterio de diseño adoptado fue:

- **Eliminación del error estático:** ajustar K_i de modo que el sistema responda ante perturbaciones o desequilibrios manteniendo $\theta \to 0$ en régimen permanente.
- **Mantenimiento de la estabilidad:** dado que la acción integral introduce un polo en el origen, se buscó evitar que este desplazara los polos dominantes hacia el semiplano derecho. Para ello se utilizaron valores pequeños de K_i inicialmente y se verificó la estabilidad mediante la prueba en la planta. Se determinó finalmente un valor de

  $$K_i = 0{,}3$$

  que cumplía con creces los objetivos propuestos.

8.2. Implementación experimental

El controlador se implementó en el Arduino en su forma discreta bilineal, de acuerdo con las indicaciones del TP, mediante la ecuación en diferencias:

$$\begin{aligned} e[k] &= \mathrm{REF} - \theta[k] \\ I[k] &= I[k-1] + K_i\, T_s\, e[k] \\ u[k] &= -K_p\, e[k] - I[k] \end{aligned}$$

Durante la experiencia, se observó que al dejar el sistema un tiempo prolongado sin perturbaciones, el péndulo comenzaba a oscilar lentamente. Este comportamiento se debe a que la acción integral continúa acumulando pequeñas desviaciones causadas por ruido y desbalance mecánico, generando una señal de control neta distinta de cero.

8.3. Análisis del fenómeno y compensación

El desplazamiento espontáneo observado tiene origen en la **deriva del integrador**, provocada por pequeñas imperfecciones en el sensor (offset del giroscopio, error del filtro complementario, fricción asimétrica, etc.). Para mitigar este efecto, se propusieron las siguientes alternativas:

1. Implementar una **zona muerta** (deadband) alrededor del error nulo, de modo que errores menores a un umbral no sean integrados.
2. Introducir un **filtro de primer orden** sobre la acción integral o un término de *anti–windup* para limitar el crecimiento de la integral ante errores persistentes.

8.4. Resultados y conclusiones

El controlador PI logró eliminar el error estacionario ante pequeñas perturbaciones o inclinaciones, mejorando la precisión del equilibrio con respecto al controlador unicamente proporcional, tal y como se puede ver en la Figura 9. Sin embargo, la sensibilidad del integrador ante el ruido lo vuelve menos robusto que el PD frente a perturbaciones rápidas. En consecuencia, se considera que el PI es adecuado para mejorar la exactitud de la posición en condiciones de bajo ruido, mientras que el PD ofrece un mejor desempeño frente a movimientos bruscos o impactos externos.

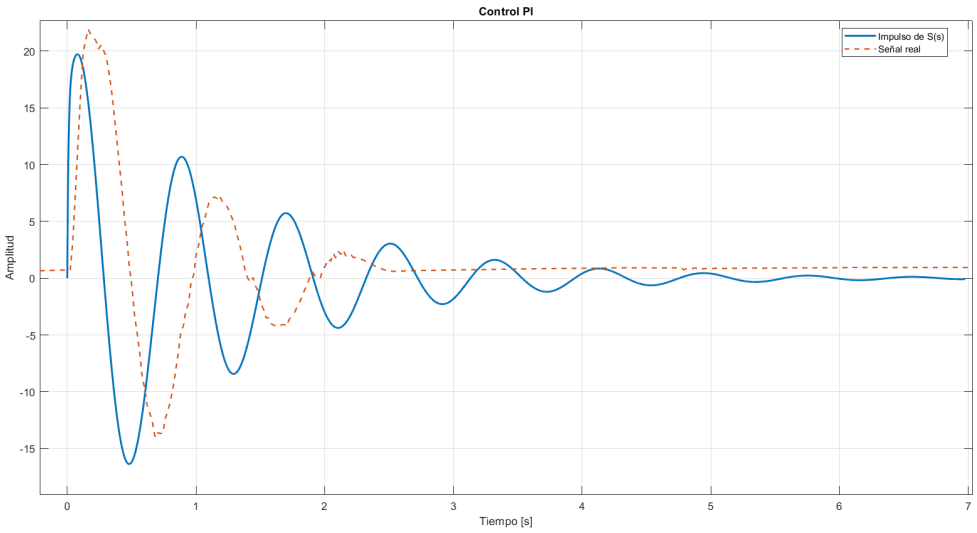

Figura 9: Respuesta del sistema utilizando un controlador PI.

9. Conclusión

A lo largo de este acercamiento a la planta, se observó como es clave encontrar una relación de compromiso entre una variedad de variables para llegar a un controlador simple pero efectivo.

Se presentaron varios problemas durante la implementación que sirvieron para afianzar los conocimientos y producir un mejor resultado. Entre algunos de ellos, el más relevante fue que la planta estimada no respondía de la misma manera a las perturbaciones que la real, con lo cual fue necesario una modificación de la transferencia para que cumpliera tal fin.

Creemos que a priori el mejor control para esta planta es un PD, ya que realiza el control más rápido que sus contrapartes P y PI.