Trabajo Práctico: Helicóptero vertical de un solo eje.

Elías Álvarez
Carrera de Ing. Electrónica
Universidad Católica Nuestra Señora de la Asunción
Asunción, Paraguay
Email: elias.alvarez@universidadcatolica.edu.py

Tania Romero
Carrera de Ing. Electrónica
Universidad Católica Nuestra Señora de la Asunción
Asunción, Paraguay
Email: tania.romero@universidadcatolica.edu.py

***Resumen*—En este trabajo práctico se estudia el control de altura de una planta experimental basada en un sistema de propulsión vertical mediante un motor brushless controlado electrónicamente. La planta consiste en un cuerpo móvil guiado mecánicamente en el eje vertical, cuya posición es medida mediante un sensor de distancia láser y regulada a través de una señal PWM aplicada a un controlador electrónico de velocidad (ESC).**

A partir del modelado físico y matemático del sistema, se desarrollan e implementan distintas estrategias de control vistas a lo largo de la materia Automatizaciones. Se analizan controladores clásicos, tales como el PID y métodos de diseño basados en el lugar de las raíces, diagramas de Bode, síntesis directa y ubicación arbitraria de polos. Asimismo, se aborda el control en el espacio de estados, incorporando control integral para sistemas de seguimiento y el uso de estimadores de estado tanto de predicción como de actualización.

Finalmente, se diseñan y evalúan controladores óptimos que incluyen integrador y estimadores basados en el filtro de Kalman, culminando en la implementación de un regulador LQG. Los distintos enfoques de control son comparados en términos de desempeño dinámico, estabilidad y robustez frente a perturbaciones y variaciones paramétricas, utilizando resultados obtenidos mediante ensayos experimentales sobre la planta real.

***Index Terms*—control de altura, sistemas dinámicos, identificación de sistemas, control PID, control en espacio de estados, LQG, PSoC, TFMini, ESC, PWM**

I. Introducción

El control de sistemas propulsados verticalmente constituye un problema clásico dentro de la ingeniería de control, particularmente cuando la dinámica presenta inestabilidad inherente, no linealidades y limitaciones físicas severas. En este trabajo se aborda la regulación y seguimiento de referencia de una planta experimental de tercer orden, caracterizada por un comportamiento inestable en lazo abierto, presencia de polos complejos dominantes, un integrador y ceros que afectan significativamente la respuesta dinámica del sistema.

Más allá del modelo teórico, la planta real introduce múltiples no idealidades: saturaciones estrictas del actuador (1100–1700 µs por razones de seguridad térmica y estructural), fricción variable debida a imperfecciones mecánicas en los rieles de guiado, variación del punto de operación asociada a la descarga progresiva de la batería, dinámica propia del motor y del ESC, y ruido de medición proveniente del sensor láser utilizado. Estas condiciones convierten el problema en un banco de pruebas exigente para evaluar la robustez y aplicabilidad práctica de distintas estrategias de control.

Un aspecto central del trabajo fue la obtención de un modelo lineal representativo del sistema alrededor del punto de operación de hover, incluyendo la identificación experimental del esfuerzo de equilibrio y la caracterización del ruido de medición y proceso. La calidad de este modelo resultó determinante para el desempeño de los métodos de control modernos, particularmente aquellos basados en espacio de estados y estimación óptima.

Sobre esta misma planta se implementaron y compararon distintas filosofías de diseño, desde enfoques clásicos hasta control óptimo con estimación de estados, evaluando su desempeño no sólo en simulación sino también mediante validación experimental directa. Esta comparación permite analizar en condiciones reales las ventajas, limitaciones y requerimientos de modelado de cada metodología.

El objetivo del trabajo no se limita a la obtención de un regulador funcional, sino a estudiar la relación entre complejidad del método, dependencia del modelo y desempeño obtenido en un sistema físico con restricciones reales.

II. Planteamiento del problema

Los sistemas de regulación de altura basados en propulsión vertical constituyen un problema particularmente exigente desde el punto de vista del control automático, debido a la combinación de inestabilidad inherente, dinámica no lineal y restricciones físicas severas del actuador. La planta experimental considerada en este trabajo presenta comportamiento inestable en lazo abierto, dinámica de tercer orden con integrador y ceros que condicionan la respuesta transitoria, incluyendo la presencia de un cero en el semiplano derecho que impone limitaciones fundamentales en el desempeño alcanzable.

Desde el punto de vista físico, el sistema consiste en un cuerpo móvil guiado en el eje vertical cuyo equilibrio depende del balance entre la fuerza gravitatoria y el empuje generado por un motor brushless accionado mediante un ESC. Sin embargo, el comportamiento real se ve influenciado por múltiples no idealidades: saturaciones estrictas del esfuerzo de control por razones de seguridad térmica (1100–1700 µs), variaciones del punto de operación asociadas a la descarga de la batería, fricción no uniforme en los rieles de guiado, dinámica propia del conjunto motor–ESC y ruido de medición proveniente del sensor láser de distancia.

Se realizaron desarrollos físico–matemáticos iniciales con el objetivo de modelar el sistema a partir de principios dinámicos fundamentales. No obstante, la complejidad aerodinámica del empuje, junto con la dinámica interna del motor y del ESC, dificultaron la obtención de un modelo analítico suficientemente representativo, lo que motivó la adopción de técnicas de identificación experimental alrededor del punto de operación de hover.

El desafío central radica entonces en obtener un modelo lineal que capture adecuadamente la dinámica dominante del sistema y permita diseñar controladores implementables en tiempo real sobre hardware embebido. El firmware desarrollado integra múltiples estrategias de control dentro del mismo sistema, permitiendo la modificación dinámica de coeficientes, la adquisición de datos en lazo abierto y cerrado, y la transmisión en tiempo real de las señales de esfuerzo y altura mediante comunicación UART para su análisis posterior.

En este contexto, el problema abordado no se limita a la estabilización del sistema, sino que implica analizar la relación entre calidad del modelado, complejidad de la estrategia de control y desempeño experimental obtenido sobre una planta física con restricciones reales.

III. Objetivos

III-A. *Objetivo general*

Desarrollar, analizar e implementar un sistema de control de altura para una planta experimental basada en un motor brushless, aplicando y comparando distintas estrategias de control automático estudiadas en la materia Automatizaciones, a partir del modelado físico y matemático del sistema y su validación experimental sobre la planta real.

III-B. *Objetivos específicos*

- Caracterizar físicamente la planta experimental y describir su comportamiento dinámico a partir de sus componentes mecánicos, eléctricos y de sensado.
- Desarrollar un modelo físico y matemático del sistema que represente adecuadamente la dinámica vertical de la planta y sirva como base para el diseño de controladores.
- Identificar experimentalmente los parámetros relevantes del modelo, considerando las no idealidades propias del sistema real.
- Diseñar e implementar controladores clásicos de altura, incluyendo el controlador PID y métodos de diseño basados en el lugar de las raíces, diagramas de Bode, síntesis directa y ubicación arbitraria de polos.
- Diseñar controladores en el espacio de estados para sistemas de seguimiento, incorporando control integral para la eliminación del error estacionario.
- Implementar y evaluar estimadores de estado tanto de tipo predictivo como de actualización, analizando su influencia en el desempeño del sistema.
- Diseñar y aplicar estrategias de control óptimo, incluyendo el uso de integrador y estimadores basados en el filtro de Kalman, culminando en la implementación de un regulador LQG.
- Comparar el desempeño de las distintas estrategias de control implementadas en términos de estabilidad, respuesta transitoria, error en régimen permanente y robustez frente a perturbaciones y variaciones paramétricas.
- Validar experimentalmente los resultados obtenidos mediante ensayos sobre la planta real, contrastando el comportamiento observado con las predicciones del modelo.

IV. Caracterización física de la planta

IV-A. *Descripción general de la planta*

La planta desarrollada corresponde a un sistema mecatrónico de **un grado de libertad**, cuyo movimiento está restringido a la **dirección vertical**. El principio de funcionamiento se basa en la generación de empuje aerodinámico mediante un **motor brushless con hélice**, controlado electrónicamente, que permite regular la altura de un cuerpo móvil guiado mecánicamente.

El sistema fue concebido como una planta experimental para el análisis y diseño de estrategias de control en altura, incorporando sensado directo de posición y actuadores eléctricos de rápida respuesta.

IV-B. *Estructura física de la planta*

La planta experimental fue diseñada y construida específicamente para el desarrollo de las prácticas de control de altura previstas en el presente trabajo. El diseño de la estructura respondió a la necesidad de disponer de una altura útil suficiente para la correcta evaluación de las distintas estrategias de control, manteniendo al mismo tiempo un esquema constructivo simple, robusto y de fácil implementación. Durante el desarrollo del proyecto se atravesaron distintas etapas de diseño, las cuales se describen con mayor detalle en el Apéndice D.

Inicialmente, la planta fue concebida con una altura total de aproximadamente 80 cm. Sin embargo, dicha dimensión resultó insuficiente para la realización de todas las prácticas propuestas. En consecuencia, y por recomendación del profesor, se decidió extender la estructura hasta alcanzar una altura total de 165 cm, permitiendo una altura útil de movimiento del cuerpo móvil de aproximadamente 134 cm.

La estructura se apoya sobre una base de madera de dimensiones 50×45 cm y un espesor aproximado de 2 cm, la cual proporciona estabilidad al conjunto. En la parte superior se dispone de un techo de madera de 50×50 cm y un espesor de 0,6 cm, que actúa como elemento de cierre y soporte estructural. La base y el techo se encuentran unidos mediante tres columnas verticales de madera, cuya función principal es aportar rigidez al conjunto y limitar las deformaciones de la estructura.

El **movimiento vertical** del sistema se guía mediante tres rieles metálicos dispuestos en paralelo, construidos a partir de vigas de metal de aproximadamente 0,6 cm de diámetro. Estas vigas presentan deformaciones inherentes al material y a su longitud, por lo que las columnas de madera cumplen un rol fundamental en evitar que dichas deformaciones se acentúen durante el funcionamiento del sistema.

El **cuerpo móvil** se desplaza a lo largo de los rieles mediante piezas impresas en 3D de tipo abrazadera. Estas abrazaderas poseen un diámetro aproximado de 1,5 cm, superior al de las

Figura 1. Vista general de la planta física.

vigas, con el objetivo de permitir cierto grado de libertad angular y evitar perturbaciones en el movimiento vertical causadas por las deformaciones de los rieles. Las abrazaderas se unen al cuerpo móvil mediante un sistema de tornillos, arandelas y tuercas, funcionando como articulaciones tipo "muñeca", que facilitan el guiado sin generar atascamientos.

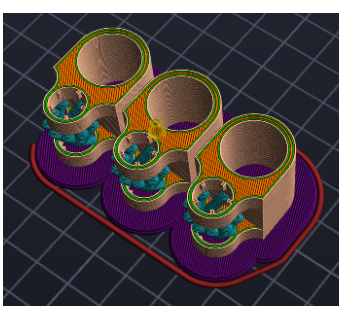


Figura 2. Diseño 3D final de la abrazadera con agarre tipo "muñeca".

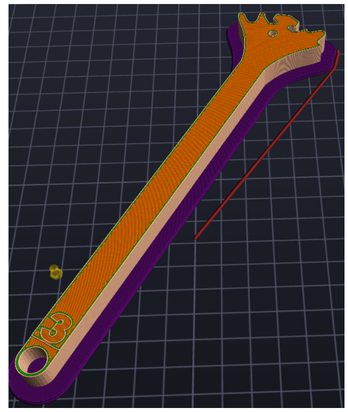


Figura 3. Diseño 3D final del brazo de unión del cuerpo móvil.

El cuerpo móvil está compuesto por dos **soportes principales**: un soporte superior que constituye la base de montaje del motor brushless y un soporte inferior que sostiene el conjunto estructural. Ambos soportes se encuentran unidos mediante tres brazos impresos en 3D, de aproximadamente 1 cm de altura, 1,3 cm de ancho y 23 cm de longitud. La unión de los distintos componentes del cuerpo móvil se realiza mediante roscas de aproximadamente 3 cm de diámetro y tuercas, evitando el uso de tornillería adicional. El conjunto incluye además un soporte para la batería, en cuya base se encuentra montado el ESC, equipado con dos disipadores térmicos laterales. La masa total del cuerpo móvil es de aproximadamente 360 g.

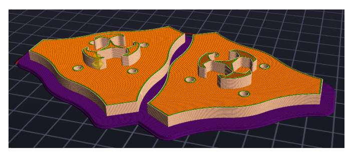


Figura 4. Diseño 3D final del soportes principales del cuerpo móvil (soporte superior e inferior).

Con el fin de proteger el sistema ante caídas y movimientos bruscos, se incorporaron **elementos de seguridad adicionales**. En la parte inferior de la estructura se dispuso papel burbuja enrollado alrededor de los rieles, actuando como amortiguación ante impactos. Asimismo, se añadieron tubos de PVC tanto en la base como en la parte superior de la estructura, los cuales funcionan como topes mecánicos que limitan el recorrido del cuerpo móvil y evitan colisiones con la base o el techo. Adicionalmente, se incorporó una cuerda de seguridad destinada a restringir levantamientos excesivos durante las prácticas, reduciendo el riesgo de movimientos abruptos.

Finalmente, se añadió una cinta métrica a lo largo de la estructura con el objetivo de **facilitar la visualización** del desplazamiento vertical y permitir una referencia directa de la altura durante el funcionamiento del sistema.

El diseño de la estructura **priorizó** la rigidez del cuerpo móvil, de modo que pueda soportar tanto su propio peso como eventuales caídas desde alturas cercanas a un metro. Asimismo, se buscó una solución de fácil construcción, utilizando materiales de adquisición accesible en el contexto local y adaptándose a los recursos disponibles y al tiempo de desarrollo del trabajo. La visibilidad del movimiento y el correcto funcionamiento mecánico del sistema fueron considerados aspectos clave para su utilización como planta experimental en este trabajo práctico.

IV-C. Cuerpo móvil

- **Masa total móvil:**

$$m = 0.360 \text{ kg}$$

(incluye motor, hélice, soporte, cableado y elementos solidarios al movimiento).

- **Tipo de movimiento:** traslación puramente vertical.
- **Altura inicial típica:** aproximadamente 12.5 cm.
- **Altura máxima disponible en la estructura:** aproximadamente 134 cm.

La ausencia de contrapesos implica que el sistema depende exclusivamente del empuje generado por la hélice para vencer la fuerza gravitatoria y los efectos de rozamiento.

IV-D. Sistema de actuación (propulsión)

El sistema de actuación está compuesto por:

- **Motor:** brushless A2212/5T, 2450 KV [1].

Figura 5. Motor

- **Hélice:**
  - Diámetro: 25 cm.

Figura 6. Hélice

- **Controlador electrónico (ESC) [2].:**
  - Corriente continua: 40 A.
  - Corriente máxima de corta duración: 55 A.
- **Batería:** LiPo 3S, [3].

Figura 7. ESC 40A

- Tensión inicial típica: 12.5 V.
- La tensión disminuye de forma apreciable durante la operación, dependiendo del régimen de empuje y la duración de la práctica.

Figura 8. Bateria LiPo 3S Ovonic

El empuje generado por el sistema depende fuertemente del comando aplicado, de la hélice y del voltaje instantáneo de la batería, lo que introduce una **no linealidad significativa** en la planta.

IV-E. Señal de control

- **Tipo de señal:** PWM tipo servo.
- **Frecuencia:** 50 Hz.
- **Rango:** 1000 μs – 2000 μs.

Esta señal actúa como la **entrada manipulada** del sistema, regulando indirectamente el empuje generado por el motor y la hélice a través del ESC.

IV-F. Sistema de sensado

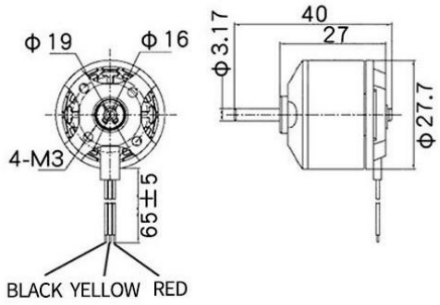
- **Sensor de altura:** TFMini Plus [4].
- **Variable medida:** posición vertical del cuerpo móvil $z(t)$.
- **Frecuencia de lectura:** configurable, típicamente en el rango de 1 Hz hasta 1000 Hz, según la configuración utilizada durante las distintas prácticas.

La señal de medición presenta efectos de **ruido**, **cuantización** y **latencia**, propios del sistema de sensado y del procesamiento digital, los cuales deben ser considerados tanto en el diseño del sistema de control como en el tratamiento de la señal medida.

IV-G. Variables del sistema

- **Entrada del sistema:**

$$u(t) = \text{PWM} \in [1000, 2000] \ \mu s$$



MOTOR PERFORMANCE DATA（性能参数）：

MODEL	KV (rpm/V)	Voltage (V)	Prop	Load Currrnt (A)	Pull (g)	Power (W)	Efficiency (g/W)	Lipo Cell	Weight (g)Approx
	930		1060	9.8	660	109	6.1		60
	1000		1047	15.6	885	173	5.1	3S	61
A2212	1400	11.1	9050	19.0	910	210	4.3		61
	2200		6030	21.5	732	239	3.1	2-3S	62
	2450		6*3	25.2	815	280	2.9		62

Figura 9. Características del motor A2212/5T, 2450KV [1]

Figura 10. Sensor óptico de distancia TFMini Plus.

- **Salida del sistema:**

$$y(t) = z(t)$$

- **Disturbios principales:**
  - Variaciones del voltaje de la batería durante la operación.
  - Rozamiento mecánico en las guías.
  - Perturbaciones aerodinámicas externas.
  - Vibraciones estructurales.

IV-H. *Limitaciones físicas y no idealidades*

La planta presenta las siguientes características no ideales:

- **Saturación del actuador**, limitada por el rango de PWM y la corriente máxima del ESC.
- **Dinámica no instantánea del empuje**, asociada a la respuesta del ESC, del motor y de la hélice.
- **Variabilidad paramétrica**, principalmente debida a la caída de tensión de la batería bajo carga.

V. Diagrama del Sistema

La **Figura 11** muestra el diagrama de bloques del sistema de control de altura en lazo cerrado. La referencia de altura z_{ref} es comparada con la señal de salida medida $z(t)$, generando el error $e(t)$, el cual es procesado por el controlador.

El controlador representa el algoritmo de control implementado, pudiendo corresponder a un controlador estudiado en el semestre. A partir del error, el controlador genera la señal de control $u(t)$, que constituye el esfuerzo aplicado al sistema.

La señal $u(t)$ actúa sobre el actuador, conformado por el motor brushless y la hélice, el cual convierte la señal de control en empuje mecánico. Dicho empuje excita la planta física, cuya dinámica vertical determina la altura real del sistema $y(t)$.

La altura es medida mediante un sensor TFMini Plus, que proporciona la señal $z(t)$ utilizada para cerrar el lazo de control. En el diagrama se indican explícitamente las fuentes de ruido asociadas tanto al controlador como al sensor, reflejando las perturbaciones y no idealidades presentes en el sistema real.

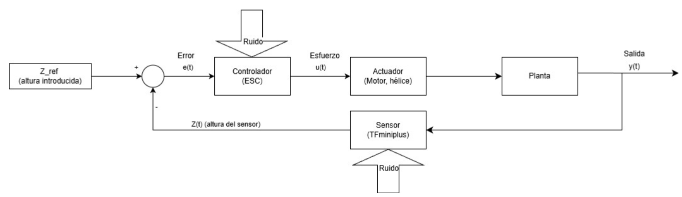


Figura 11. Diagrama de bloques de Control.

La **Figura 12** muestra un diagrama de bloques que representa la implementación física y funcional completa del sistema de control de altura. En dicho diagrama se identifican claramente los siguientes subsistemas:

- la cadena de energía (`batería` → `ESC` → `motor`),
- la cadena de control (`MATLAB` → `PSoC` → `PWM`),
- la planta física correspondiente al sistema bajo control,
- y la cadena de medición y realimentación (`planta` → `sensor` → `PSoC` → `MATLAB`).

V-A. *Flujo de energía (parte electro–energética)*

La batería utilizada es una LiPo de tres celdas (3S), con tensión nominal de $11{,}1\,\text{V}$. Cada celda presenta una tensión máxima de $4{,}2\,\text{V}$, por lo que la batería completamente cargada alcanza:

$$V_{\text{max}} = 12{,}6\,\text{V}$$

Con el objetivo de preservar la integridad química de la batería y evitar degradación prematura, el rango operativo se restringe aproximadamente a:

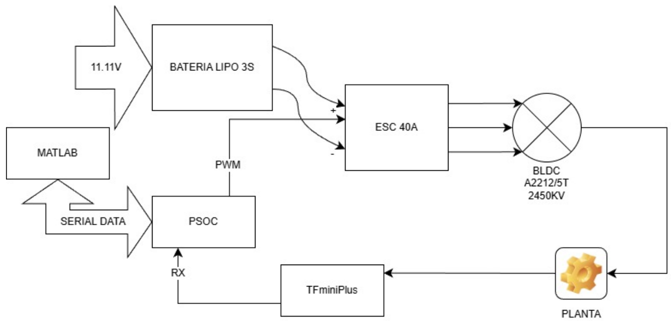


Figura 12. Diagrama de bloques, implementación física.

$$V \in [11{,}5,\ 12{,}5]\,\mathrm{V}$$

Evitar descargas por debajo de $11{,}5\,\mathrm{V}$ resulta fundamental, ya que tensiones inferiores pueden producir daño irreversible en las celdas LiPo.

La batería alimenta directamente al controlador electrónico de velocidad (ESC) de $40\,\mathrm{A}$, el cual convierte la tensión continua en señales trifásicas moduladas para accionar el motor brushless.

El ESC incorpora disipadores térmicos integrados sobre los dispositivos de potencia (MOSFETs), cuya función es evacuar el calor generado durante la conmutación y conducción de corriente. Dado que el sistema opera en un régimen de potencia relativamente elevado para el tamaño del conjunto, la gestión térmica resulta crítica para evitar sobrecalentamientos que puedan producir fallas o reducción de eficiencia.

La presencia de disipadores mejora la transferencia térmica hacia el ambiente, aumentando la confiabilidad del sistema durante los intervalos de operación de aproximadamente $5\,\mathrm{min}$ continuos.

A medida que la batería se descarga, se observa una disminución progresiva de la tensión disponible, lo que impacta directamente en la capacidad de generación de empuje del motor. Este fenómeno es perceptible incluso en aplicaciones aeronáuticas (como drones), donde la pérdida de tensión se traduce en menor capacidad de sustentación.

En la práctica experimental, el tiempo de operación continua es del orden de $5\,\mathrm{min}$ como máximo. Esto se debe a que la batería utilizada posee una capacidad limitada para el nivel de potencia demandado por el sistema.

El tiempo típico de recarga completa es aproximadamente $1{,}5\,\mathrm{h}$, lo que introduce una restricción significativa en la repetibilidad de los ensayos experimentales, constituyendo un factor limitante en el desarrollo de la práctica.

En esta etapa del sistema no se realiza acción de control propiamente dicha, sino únicamente conversión y transferencia de potencia hacia el actuador.

V-B. *Flujo de control (parte de mando)*

El PSoC actúa como el controlador embebido del sistema. A partir de los algoritmos de control implementados (PID, control en espacio de estados, LQG, entre otros), el PSoC genera una señal PWM de tipo servo que es enviada al ESC.

Esta señal PWM representa la variable de control $u(t)$ del sistema y determina el nivel de empuje aplicado al motor. Para asegurar una correcta referencia eléctrica y el funcionamiento adecuado del sistema, el PSoC y el ESC comparten una conexión de tierra común (GND).

El sistema MATLAB se comunica con el PSoC mediante una interfaz de datos seriales, lo que permite:

- enviar referencias de altura,
- modificar parámetros de control,
- recibir y visualizar datos del sistema en tiempo real.

En este esquema, MATLAB cumple una función de supervisión, ajuste y análisis experimental, mientras que el PSoC ejecuta el control en tiempo real.

V-C. *Medición y realimentación (cierre del lazo)*

La planta física, al desplazarse verticalmente, genera una altura real que es medida mediante el sensor de distancia láser TFMini Plus. Este sensor entrega la medición de altura al PSoC a través de su interfaz de recepción (RX).

La señal medida es utilizada por el PSoC para calcular el error entre la referencia y la salida real del sistema, cerrando de esta manera el lazo de control de altura. Adicionalmente, los datos medidos pueden ser enviados a MATLAB para su visualización, almacenamiento y análisis experimental.

VI. Modelado físico del sistema

VI-A. *Variables y convenciones*

Se definen a continuación las variables y convenciones utilizadas para el modelado del sistema:

- Eje vertical z [m], definido positivo hacia arriba.
- Masa móvil:
$$m = 0.360\ \text{kg}$$
- Aceleración de la gravedad:
$$g = 9.81\ \text{m/s}^2$$
- Peso del cuerpo móvil:
$$mg = 3.924\ \text{N}$$
- Entrada del sistema: señal PWM tipo servo a 50 Hz,
$$u \in [1000, 2000]\ \mu s$$

Si bien el rango eléctrico nominal del protocolo PWM se encuentra entre 1000 y 2000 μs, durante la operación en vuelo la señal se restringe intencionalmente al intervalo

$$u \in [1100, 1700]\ \mu s$$

Esta limitación surge de criterios de confiabilidad experimental. En ensayos previos realizados con ESCs de menor capacidad nominal (30 A), se observaron fallas térmicas al operar en valores superiores a aproximadamente 1600 μs bajo carga sostenida.

Dado que no se realizaron mediciones directas de corriente ni caracterizaciones térmicas detalladas del conjunto motor–ESC–hélice, se adoptó un margen de seguridad conservador que evita la operación prolongada en regímenes de alta demanda energética.

Asimismo, el límite inferior de 1100 μs se fija con el objetivo de evitar regiones cercanas a la detención del motor, donde pueden presentarse comportamientos fuertemente no lineales y pérdida abrupta de sustentación.

En consecuencia, el modelo identificado y las estrategias de control desarrolladas se consideran válidos únicamente dentro de este rango operativo seguro.

- Normalización de la entrada:

$$\hat{u} = \mathrm{sat}(u - u_0,\, 1100 - u_0,\, 1700 - u_0)$$

Siendo u_0 el valor de PWM necesario para generar un empuje equivalente al peso del sistema, es decir, la condición de equilibrio vertical estacionario.

- Salida medida:

$$y = z$$

correspondiente a la altura del cuerpo móvil medida mediante el sensor TFMini Plus, cuya resolución efectiva se encuentra en el orden de centímetros.

VI-A1. Modelo físico simplificado: Desde un punto de vista físico, el movimiento vertical del cuerpo móvil puede describirse, en primera aproximación, mediante:

$$\dot{z} = v$$

$$m\dot{v} = T - mg$$

Este modelo corresponde al caso ideal sin disipación, en el cual la dinámica posición–empuje presenta una estructura de doble integración.

En una aproximación más realista, puede incorporarse el efecto de rozamiento de las guías mediante un término viscoso proporcional a la velocidad:

$$m\dot{v} = T - mg - b\,v$$

donde b [N s/m] representa un coeficiente equivalente de fricción viscosa.

La inclusión de este término modifica la estructura ideal de doble integrador, convirtiéndola en un sistema con integración amortiguada. Desde el punto de vista de función de transferencia continua, el doble polo en el origen deja de ser estrictamente doble, introduciéndose un amortiguamiento mecánico que desplaza uno de los polos hacia el semiplano izquierdo.

Por lo tanto, la presencia de un único integrador dominante en el modelo simplificado resulta coherente con la física del sistema cuando se consideran pérdidas mecánicas.

VI-A2. Modelo dinámico equivalente del actuador (BLDC): A efectos de análisis lineal, el conjunto ESC–motor BLDC puede aproximarse mediante un modelo promedio equivalente al de un motor DC de imanes permanentes:

$$v(t) = R\,i(t) + L\,\dot{i}(t) + K_e\,\omega(t)$$

$$J\,\dot{\omega}(t) = K_t\,i(t) - B\,\omega(t) - \tau_L(t)$$

Este modelo introduce una dinámica electromecánica adicional respecto del modelo puramente mecánico de la masa móvil, justificando la aparición de un polo adicional en la relación entrada–salida.

En muchos casos, la constante de tiempo eléctrica $\tau_e = L/R$ es considerablemente menor que la mecánica, permitiendo despreciar L y obtener una dinámica dominante de segundo orden asociada al actuador.

VI-A3. Función de transferencia continua equivalente: La representación continua obtenida a partir del modelo identificado puede expresarse inicialmente como:

$$G(s) = \frac{-0.12107\,(s-14)(s+10.62)}{(s+0.0002797)\,(s^2+5.61s+14.02)}$$

El polo ubicado en $s = -0.0002797$ corresponde a una dinámica extremadamente lenta respecto de las restantes constantes de tiempo del sistema.

Considerando que:

- la resolución del sensor de altura se encuentra en el orden de centímetros,
- no se dispone de mediciones de alta precisión submilimétrica,
- el identificador polinomial puede ajustar polos muy cercanos al origen para capturar pequeñas tendencias de deriva,

se interpreta que dicho polo próximo a cero no representa una dinámica física real dominante, sino un posible efecto de sobreajuste (*overfitting*) del procedimiento de identificación.

En consecuencia, y en coherencia con el modelo físico que contempla amortiguamiento mecánico, se adopta la siguiente simplificación:

$$G(s) = \frac{-0.12107\,(s-14)(s+10.62)}{s\,(s^2+5.61s+14.02)}$$

En esta forma:

- El polo en el origen representa el carácter integrador dominante de la posición vertical.
- El segundo orden $s^2 + 5.61s + 14.02$ modela la dinámica electromecánica del actuador.
- Los ceros se interpretan como parámetros de ajuste que capturan efectos agregados del actuador, discretización y linealización alrededor del punto de operación.

VI-A4. Obtención del modelo mediante identificación en MATLAB: La función de transferencia utilizada en este trabajo no se obtuvo exclusivamente a partir de un modelo físico teórico, sino mediante un proceso de **identificación experimental**.

En particular, se empleó la herramienta *System Identification Toolbox* de MATLAB, utilizando registros experimentales previamente adquiridos del sistema en lazo abierto (señal de mando PWM y altura medida por el sensor). Dichos datos fueron importados al entorno de identificación y organizados en un objeto `iddata`, definiendo la entrada como $u(t)$ y la salida como $y(t) = z(t)$, con el tiempo de muestreo correspondiente al sistema de adquisición.

Sobre ese conjunto de datos se ajustó un modelo lineal SISO en tiempo continuo, seleccionando una estructura paramétrica compatible con la dinámica esperada del sistema, y estimando sus parámetros mediante los algoritmos de optimización incluidos en MATLAB.

Finalmente, el modelo resultante fue validado comparando su respuesta con los datos medidos (ajuste temporal y análisis de residuales), y se adoptó como representación equivalente continua la función de transferencia $G(s)$ presentada en la sección siguiente.

VI-A5. Validez del modelo: El modelo adoptado constituye una aproximación coherente con la física del sistema y adecuada para el diseño de control dentro del rango operativo $[1100, 1700]\ \mu s$.

Fuera de dicho intervalo, el sistema presenta comportamientos no lineales significativos (aerodinámica, fricción no lineal, saturaciones y posibles limitaciones térmicas del actuador) que no son capturados por el modelo lineal simplificado.

VII. Métodos Clásicos de Control

VII-A. Introducción

Los métodos clásicos de control se fundamentan en el análisis de sistemas lineales mediante funciones de transferencia y herramientas del dominio de la frecuencia y del plano complejo.

En este enfoque, la dinámica del sistema se describe como:

$$G(s) = \frac{Y(s)}{U(s)} \quad \text{o en tiempo discreto} \quad G(z) = \frac{Y(z)}{U(z)}$$

El diseño del controlador se realiza modificando la función de transferencia del lazo abierto:

$$L(s) = C(s)G(s) \quad \text{o} \quad L(z) = C(z)G(z)$$

con el objetivo de garantizar:

- Estabilidad en lazo cerrado.
- Desempeño transitorio adecuado.
- Error estacionario reducido.
- Robustez frente a incertidumbre.

Los métodos clásicos utilizados en este trabajo incluyen:

- Controlador PID.
- Diseño por Lugar de Raíces.
- Diseño mediante Respuesta en Frecuencia (Bode).
- Síntesis Directa (Truxal–Ragazzini).

VII-B. Modelo de la Planta

El diseño clásico parte de la función de transferencia identificada:

$$G(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + b_3 z^{-3}}{1 + a_1 z^{-1} + a_2 z^{-2} + a_3 z^{-3}}$$

Tiempo de muestreo, depende de la experiencia.

VII-C. Fundamentos Matemáticos del Control Clásico

VII-C1. Estabilidad en Lazo Cerrado: El sistema en lazo cerrado se expresa como:

$$G_{cl}(z) = \frac{C(z)G(z)}{1 + C(z)G(z)}$$

La estabilidad discreta requiere que:

$$|z_i| < 1 \quad \forall i$$

donde z_i son los polos de $G_{cl}(z)$.

VII-C2. Lugar de Raíces: Los polos del lazo cerrado satisfacen:

$$1 + C(z)G(z) = 0$$

El Lugar de Raíces describe la trayectoria de estos polos al variar la ganancia del controlador.

VII-C3. Márgenes de Estabilidad: A partir del diagrama de Bode del lazo abierto $L(z)$ se definen:

- Margen de ganancia.
- Margen de fase.
- Frecuencia de cruce de ganancia.

Estos parámetros cuantifican la estabilidad relativa y robustez del sistema.

VII-D. Controlador PID

Durante las etapas iniciales de diseño se intentó sintonizar un controlador PID utilizando la herramienta `PID Tuner` de MATLAB. Sin embargo, los resultados obtenidos no fueron satisfactorios para la planta bajo estudio, principalmente debido a la complejidad de la dinámica identificada, la presencia de retardos efectivos asociados al sistema de actuación (ESC–motor) y las limitaciones físicas del actuador. En particular, el desempeño obtenido presentaba respuestas lentas o esfuerzos de control excesivos, incompatibles con la implementación experimental.

Ante esta situación, se optó por utilizar una formulación de controlador PID basada en el método propuesto por Åström, el cual permite un mayor control sobre la estructura del controlador y sobre el compromiso entre rapidez, amortiguamiento y esfuerzo de control. Este enfoque resultó más adecuado para la planta identificada y permitió obtener respuestas dinámicas satisfactorias en simulación y en la práctica.

VII-D1. Formulación del PID de Åström: La estructura del controlador PID de Åström se implementa en forma discreta y separa explícitamente las acciones proporcional, integral y derivativa. El término proporcional se define como:

$$P(t) = K\big(b\,u_c(t) - y(t)\big)$$

donde K es la ganancia proporcional y b es un parámetro que determina qué fracción de la referencia se introduce en la acción proporcional, permitiendo reducir el sobreimpulso ante cambios bruscos de referencia.

La acción derivativa se implementa mediante un filtro de primer orden, cuya ecuación en tiempo discreto es:

$$D(kh) = \frac{T_d}{T_d + Nh}\,D(kh-h) - \frac{KT_dN}{T_d + Nh}\big(y(kh) - y(kh-h)\big)$$

donde T_d es la constante de tiempo derivativa, N es el parámetro que limita el ancho de banda del término derivativo, y h es el período de muestreo.

La acción integral se describe mediante:

$$I(kh + h) = I(kh) + \frac{Kh}{T_i}\,e(kh)$$

donde T_i es la constante de tiempo integral y $e(kh) = u_c(kh) - y(kh)$ es el error de control.

La señal de control total se obtiene como la suma de los tres términos:

$$u(kh) = P(kh) + I(kh) + D(kh)$$

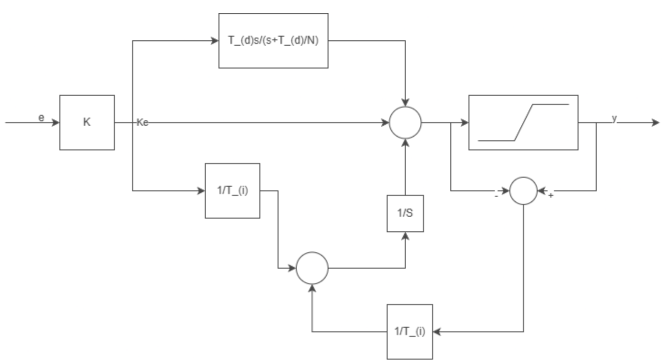


Figura 13. Diagrama de bloques, Astrom - Antiwindup [5]. Aplicada a la planta.

VII-D2. Criterios de sintonización: La sintonización de los parámetros del controlador se realizó de forma iterativa, utilizando simulaciones y observando tanto la respuesta del sistema como el esfuerzo de control. Los criterios adoptados fueron los siguientes:

- **Ganancia proporcional K:** se incrementó progresivamente hasta aproximar el sistema al umbral de inestabilidad, con el objetivo de obtener un transitorio rápido y una respuesta ágil.
- **Acción integral T_i:** se incorporó posteriormente para eliminar el error en régimen permanente, ajustando su valor de manera que el tiempo de establecimiento resultara razonable sin introducir oscilaciones significativas.
- **Acción derivativa T_d:** se añadió con el fin de reducir el sobreimpulso y mejorar el amortiguamiento del sistema.
- **Parámetro N:** se ajustó para limitar la amplificación de ruido del término derivativo, probando distintos valores hasta observar una señal excesivamente sensible al ruido de medición.

Durante todo el proceso de sintonización se monitoreó cuidadosamente el esfuerzo de control. Como condición de diseño, se impuso que la variación de la señal PWM no superara aproximadamente $10\,\mu s$ por centímetro de incremento en la altura, garantizando de esta manera que el actuador no entrara en saturación ni se expusiera la planta a condiciones potencialmente dañinas.

VII-D3. Resultados: El controlador PID basado en el método de Åström permitió obtener un comportamiento dinámico estable, con un compromiso adecuado entre rapidez de respuesta, amortiguamiento y esfuerzo de control. En comparación con los resultados obtenidos mediante el `PID Tuner`, esta metodología ofreció mayor flexibilidad y un mejor ajuste a las particularidades de la planta identificada, resultando adecuada para su implementación experimental (Figuras 14 y 15).

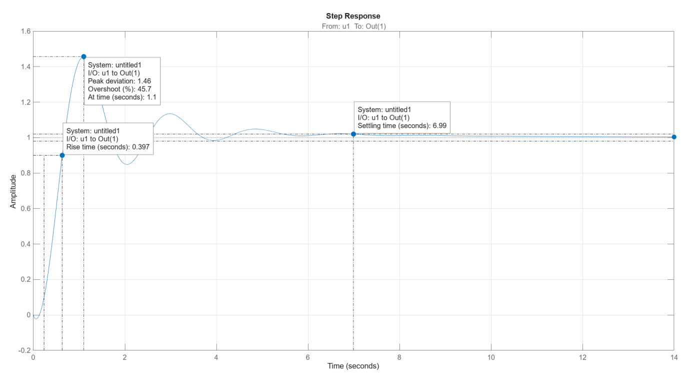


Figura 14. La respuesta al escalón con el controlador PID.

VII-D4. Práctica: Los parámetros implementados del controlador PID fueron:

$$K_p = 2.5, \quad T_i = 5, \quad T_d = 0.1, \quad N = 3$$

En la simulación del modelo lineal (figura 14) se obtuvo:

$$\%OS_{\text{sim}} \approx 46\,\%, \quad t_r^{\text{sim}} = 0.397 \text{ s}$$

En la implementación experimental (figura 16) se observaron los siguientes valores de sobreimpulso según la altura de referencia:

- Para 57 cm: $\%OS = 28.07\,\%$
- Para 78 cm: $\%OS = 11.54\,\%$
- Para 90 cm: $\%OS \approx 0\,\%$

El tiempo de subida experimental fue:

$$t_r^{\text{exp}} = 0.393 \text{ s}$$

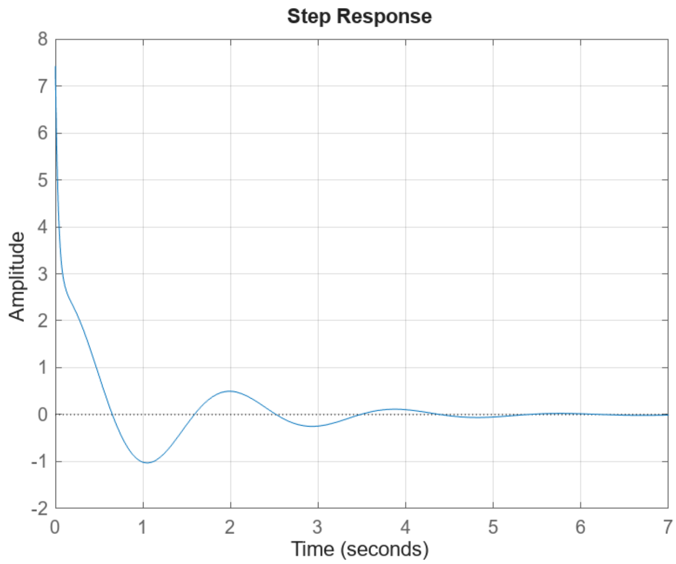


Figura 15. El esfuerzo del con el controlador PID.

Se observa que el tiempo de subida coincide prácticamente con el predicho por el modelo, mientras que el sobreimpulso disminuye progresivamente a medida que aumenta la altura de operación.

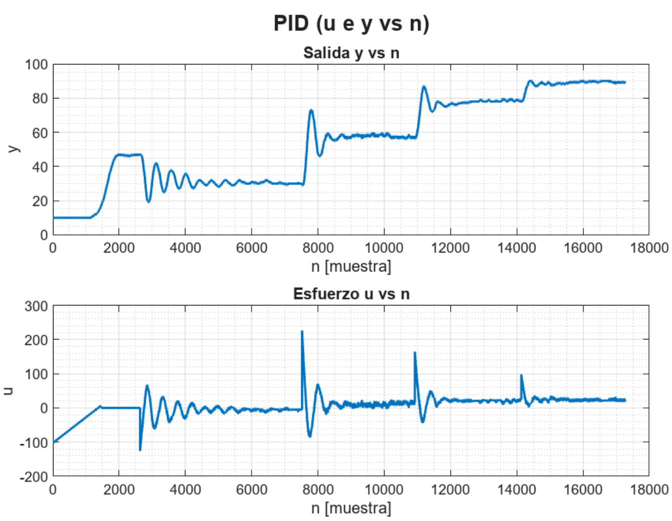


Figura 16. Implementación práctica del controlador PID.

VII-E. *Diseño por Lugar de Raíces*

Se analiza la ecuación característica:

$$1 + C(z)G(z) = 0$$

Dado que la planta identificada presenta polos ubicados fuera del círculo unitario, el sistema en lazo abierto resulta inestable en el dominio discreto. En consecuencia, cualquier diseño de control debe garantizar que los polos del lazo cerrado queden estrictamente dentro del círculo unitario para asegurar estabilidad interna.

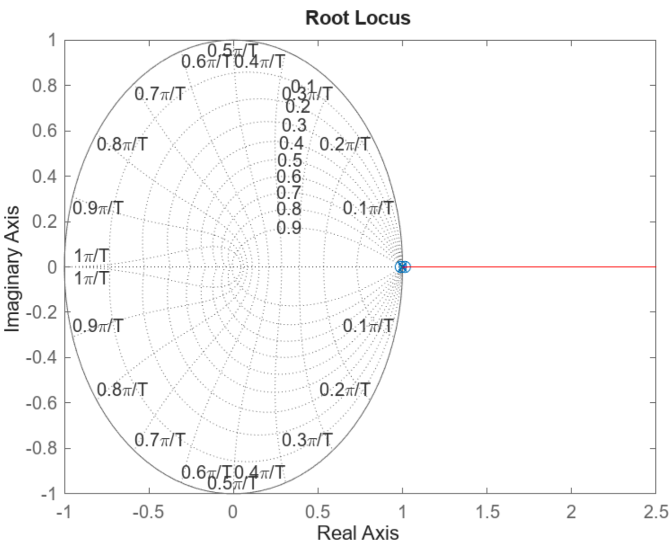


Figura 17. Ubicación de los polos de la planta sin compensar en el plano z. Se observa que al menos uno de ellos se encuentra fuera del círculo unitario, lo que implica inestabilidad discreta.

El objetivo del diseño consistió en modificar la dinámica del sistema mediante compensación, de modo que:

- todos los polos del lazo cerrado queden dentro del círculo unitario,
- se logre un compromiso adecuado entre rapidez de respuesta y amortiguamiento,
- el esfuerzo de control permanezca dentro de límites físicamente realizables.

VII-E1. Elección de la estructura del compensador: Para estabilizar el sistema se adoptó una estructura de tipo **lag–lead** (atraso–adelanto). Esta configuración permite actuar simultáneamente sobre la estabilidad relativa y el desempeño en régimen permanente.

El término *lead* (adelanto) se empleó para aumentar el margen de fase y desplazar los polos dominantes del lazo cerrado hacia regiones del plano z asociadas con mayor amortiguamiento y mejor desempeño transitorio. Por otro lado, el término *lag* (atraso) permitió ajustar la ganancia en bajas frecuencias, mejorando el comportamiento estacionario sin comprometer significativamente la estabilidad.

La adecuada ubicación de ceros permitió modificar la geometría del lugar de raíces, atrayendo las ramas hacia la región estable del plano discreto, mientras que los polos adicionales modelaron el compromiso dinámico requerido.

VII-E2. Determinación de la ganancia K: Una vez definida la estructura del compensador, se analizó el lugar de raíces del sistema compensado. La determinación manual de la ganancia K resultó particularmente sensible, ya que pequeños incrementos en su valor provocaban que las trayectorias de los polos abandonaran el círculo unitario antes de satisfacer las especificaciones dinámicas deseadas.

Esta sensibilidad está directamente relacionada con la naturaleza inestable de la planta y con la fuerte dependencia de la ubicación de los polos del lazo cerrado respecto a la ganancia del compensador.

El compensador finalmente adoptado fue:

$$C(z) = -0.0173\,\frac{(z - 1.0140)(z - 0.5)}{(z - 0.9522)(z - 0.9894)} \tag{1}$$

VII-E3. Ajuste mediante `Optimization-Based Tuning`: Con el fin de sistematizar el proceso de ajuste y garantizar el cumplimiento simultáneo de múltiples especificaciones (estabilidad, rapidez y esfuerzo de control), se utilizó la herramienta `Optimization-Based Tuning` de MATLAB.

Este enfoque permitió:

- definir directamente especificaciones temporales (tiempo de establecimiento, sobreimpulso, etc.),
- ajustar automáticamente los parámetros del compensador,
- verificar la estabilidad del sistema en el dominio discreto.

El resultado fue un compensador lag–lead cuyos parámetros fueron obtenidos mediante optimización numérica, asegurando que los polos del lazo cerrado se mantengan dentro del círculo unitario y que el desempeño temporal cumpla con los objetivos establecidos para la planta experimental.

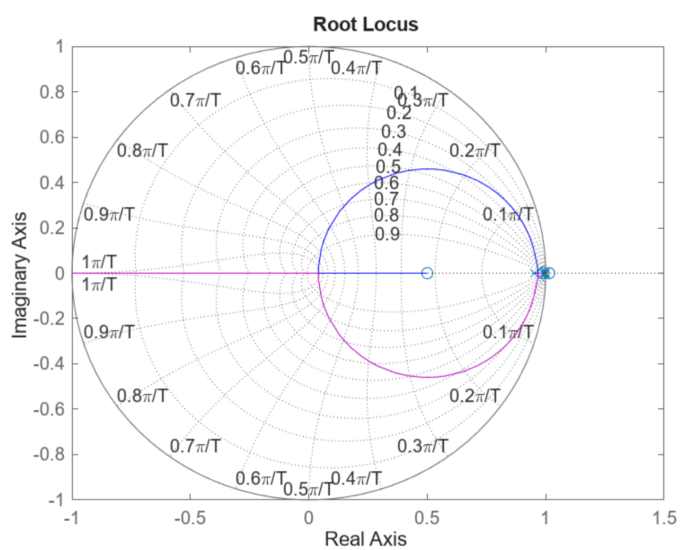


Figura 18. Ubicación de los polos del sistema compensado en el plano z. Se verifica que todos ellos se encuentran dentro del círculo unitario, garantizando estabilidad discreta.

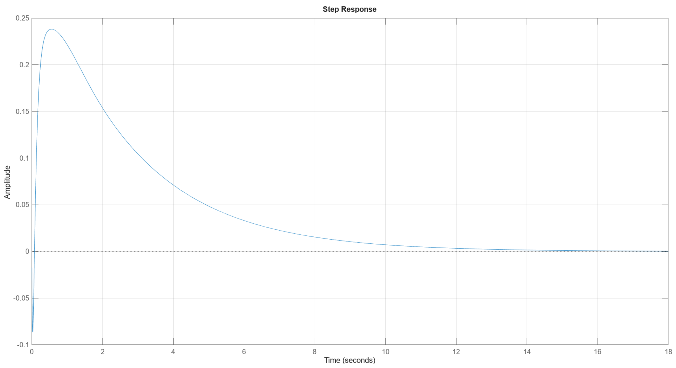


Figura 19. Esfuerzo de control en lazo cerrado con el compensador diseñado mediante Lugar de Raíces.

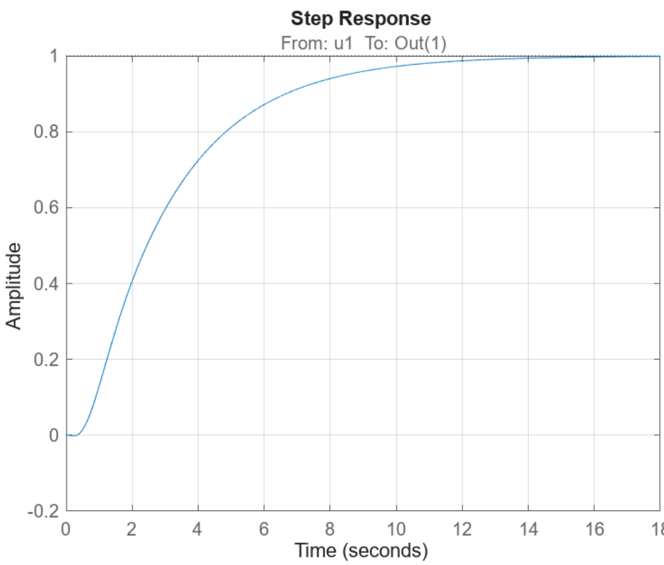


Figura 20. Respuesta temporal del sistema en lazo cerrado con el compensador diseñado.

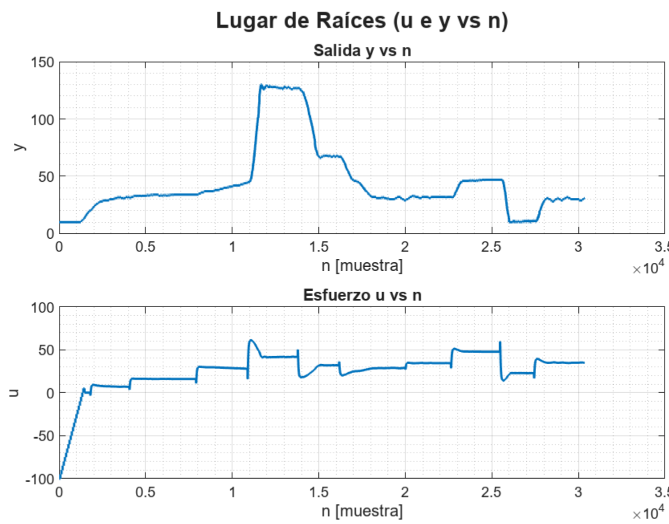


Figura 21. Esfuerzo y altura de la implementación práctica de lugar de raices.

VII-E4. Práctica: En la implementación experimental se obtuvo un tiempo de subida de:

$$t_r^{\text{exp}} = 1.068 \text{ s}$$

Mientras que en la simulación del modelo lineal el tiempo de subida observado fue aproximadamente:

$$t_r^{\text{sim}} \approx 5 \text{ s}$$

En ambos casos no se registró sobreimpulso, lo que indica un comportamiento sobreamortiguado del sistema.

Sin embargo, se observa una diferencia significativa en la rapidez de la respuesta, siendo el sistema real considerablemente más rápido que el modelo simulado.

Dado que el sistema presentaba estabilidad intrínseca, se ajustó la ganancia del controlador con el objetivo de limitar el esfuerzo de control y evitar saturaciones excesivas.

VII-F. *Diseño por Respuesta en Frecuencia*

Se analiza el lazo abierto:

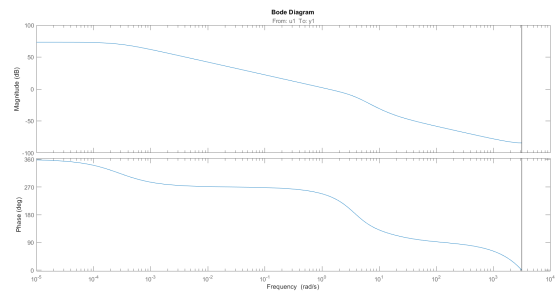


Figura 22. Respuesta en frecuencia del lazo abierto del sistema identificado sin compensación.

Para el diseño del controlador basado en el método de respuesta en frecuencia se utilizó directamente el modelo discreto de la planta obtenido mediante identificación experimental, mostrado en la Fig. 22. En particular, se trabajó con la función de transferencia discreta $G(z)$ dada en (**??**), la cual representa el comportamiento dinámico del sistema en el rango de operación considerado.

El modelo identificado fue introducido en el entorno de diseño de MATLAB mediante la herramienta `controlSystemDesigner`, lo que permitió analizar la respuesta en frecuencia del sistema y realizar el diseño del controlador de manera interactiva. A partir de esta herramienta se obtuvieron los diagramas de Bode del lazo abierto, así como los márgenes de ganancia y de fase asociados.

VII-F1. Análisis del sistema sin compensar: En la Fig. 23, correspondiente al sistema sin compensación, se observa el diagrama de Bode del lazo abierto conformado únicamente por la planta identificada. A partir de dicho análisis se determinan los márgenes de estabilidad iniciales, los cuales permiten evaluar la estabilidad relativa y la robustez del sistema frente a variaciones paramétricas.

El sistema presenta un margen de fase positivo, lo que indica estabilidad en lazo cerrado para valores reducidos de ganancia, aunque con un compromiso limitado en términos de rapidez de respuesta y amortiguamiento. Asimismo, la pendiente del módulo en la región de cruce de ganancia evidencia la presencia de múltiples polos dominantes, coherentes con la dinámica de orden superior identificada y asociada al conjunto motor–ESC–hélice.

VII-F2. Diseño del compensador: Con base en el análisis previo, se procedió al diseño de un compensador con el objetivo de mejorar el desempeño dinámico del sistema, manteniendo márgenes de estabilidad adecuados. El diseño se realizó ajustando la estructura y los parámetros del controlador directamente sobre el diagrama de Bode del lazo abierto, utilizando la herramienta gráfica provista por MATLAB.

Los criterios de diseño considerados fueron:

- incrementar el margen de fase para mejorar el amortiguamiento del sistema,
- fijar una frecuencia de cruce que permita un compromiso adecuado entre rapidez de respuesta y robustez,
- limitar la amplificación de ruido a altas frecuencias y el esfuerzo de control.

VII-F3. Análisis del sistema compensado: La Fig. 24 muestra el diagrama de Bode del lazo abierto una vez incorporado el controlador diseñado. Se observa un aumento del margen de fase y un ajuste controlado del margen de ganancia, lo cual indica una mejora en la estabilidad relativa del sistema.

Los valores obtenidos a partir de los diagramas de Bode son los siguientes:

- **Sistema sin compensación (Fig. 23):**
  - Margen de ganancia: $13\,\mathrm{dB}$,
  - Margen de fase: $61{,}700°$.
- **Sistema compensado (Fig. 24):**
  - Margen de ganancia: $10.6\,\mathrm{dB}$,
  - Margen de fase: $52{,}900°$.

La validación del diseño se realizó mediante el análisis de la respuesta temporal en lazo cerrado. La respuesta al escalón del sistema compensado presenta un comportamiento estable, con un sobreimpulso moderado y un tiempo de establecimiento acorde a los objetivos del trabajo. Asimismo, la señal de control se mantiene dentro de valores aceptables, evitando saturaciones prolongadas del actuador.

En este caso, la compensación se realizó mediante un controlador puramente proporcional, definido como:

$$C(z) = K_p, \qquad K_p = 1.308$$

Por lo tanto, el lazo abierto queda dado por $L(z) = K_p\, G(z)$. La acción del controlador proporcional consiste en escalar la magnitud de la respuesta en frecuencia del lazo abierto, desplazando la frecuencia de cruce y, en consecuencia, modificando los márgenes de ganancia y de fase observados. No se introducen polos ni ceros adicionales, por lo que no se realiza una compensación dinámica de fase; la mejora del desempeño se logra exclusivamente mediante el ajuste de la ganancia.

La incorporación de polos adicionales introduciría retardos en la respuesta y una reducción del margen de fase, mientras que la adición de ceros podría generar un adelanto de fase a costa de un incremento significativo del esfuerzo de control. Dado que en la simulación no se dispone de una estimación fiable del esfuerzo del actuador, no resulta posible ponderar adecuadamente estos efectos. Por este motivo, se descartó la inclusión de polos, ceros o acción integral en esta etapa, con el fin de preservar la integridad del sistema físico durante la implementación experimental.

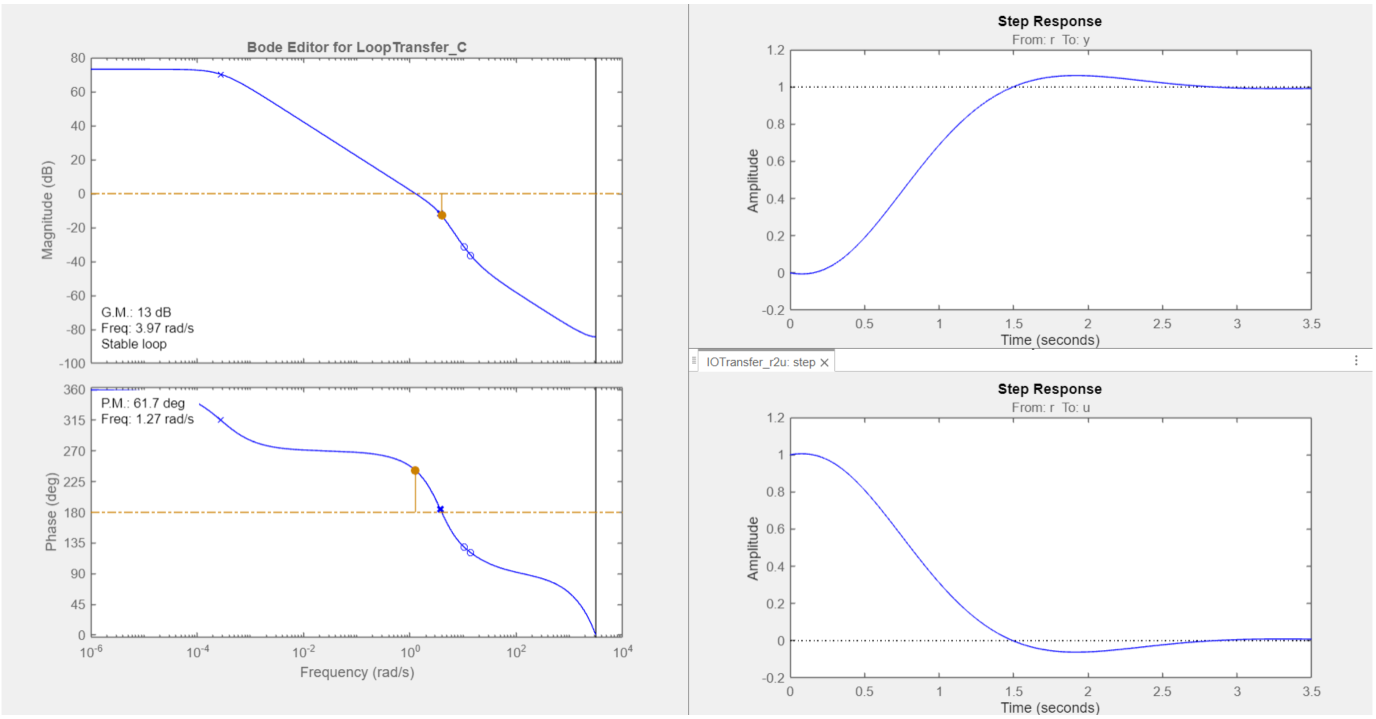


Figura 23. Diagrama de Bode y respuestas temporales del sistema con compensación.

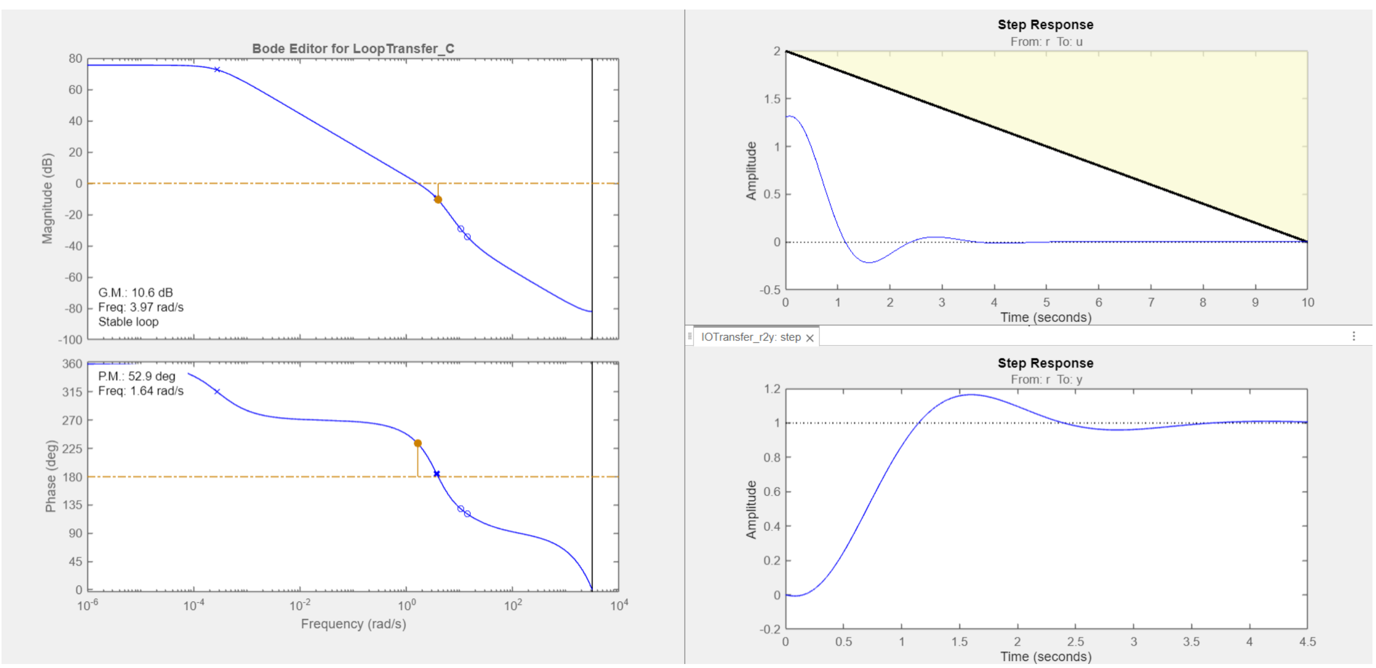


Figura 24. Diagrama de Bode y respuestas temporales del sistema con compensación.

VII-G. *Práctica*

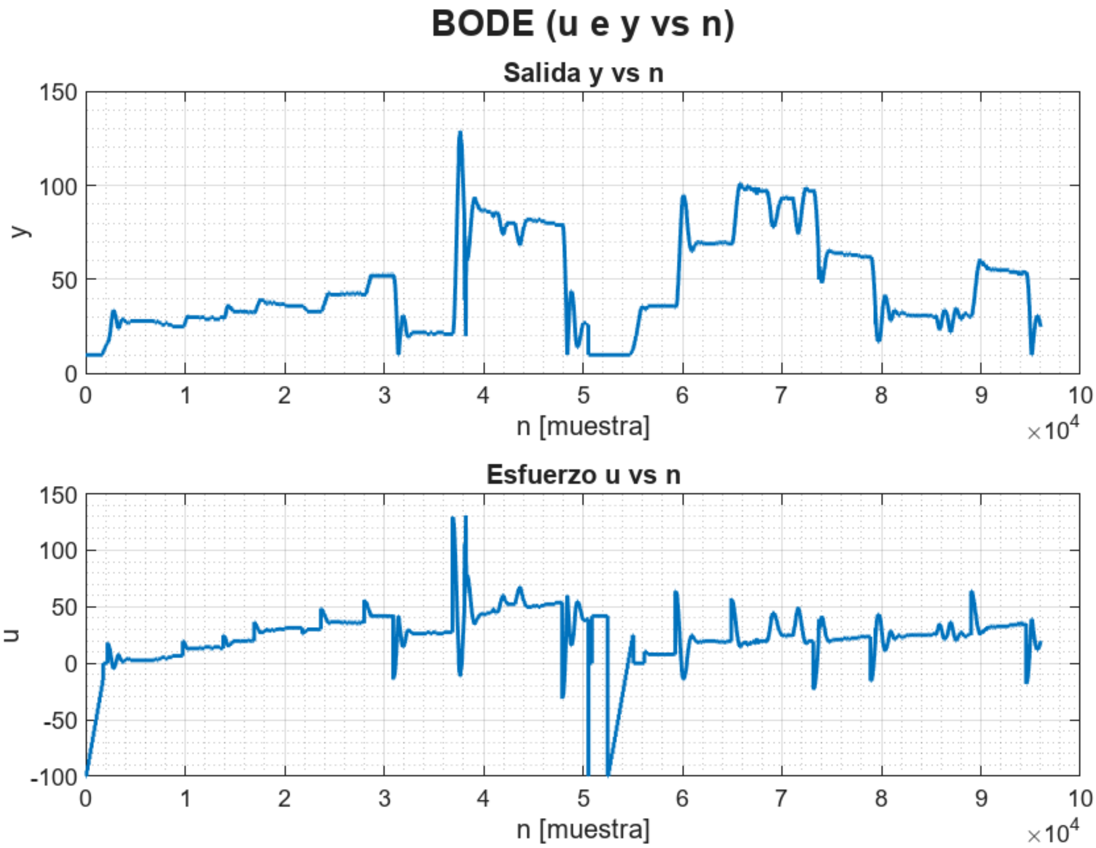


Figura 25. Figura de la respuesta del bode, altura y esfuerzo en n muestras

Compensador usado:

$$C_{Bode} = 1.3082$$

Tiempo de muestreo:

$$T_s = 0.0001$$

El sobreimpulso simulado con el compensador es de aproximadamente $20\,\%$. En la implementación experimental, el primer levantamiento presenta un sobreimpulso máximo de $55.55\,\%$. Sin embargo, en un levantamiento posterior el sobreimpulso disminuye a aproximadamente $37\,\%$, y tiende a reducirse adicionalmente a medida que aumenta la altura de operación.

Esta variabilidad entre levantamientos sugiere una dependencia del comportamiento transitorio con el punto de operación y las condiciones iniciales (por ejemplo, arranque desde reposo, dinámica del actuador y no linealidades del empuje). En particular, durante el primer levantamiento se observa un esfuerzo de control mayor al inicio, que luego decrece conforme el sistema se aproxima al equilibrio dinámico, reduciendo la acción requerida para sostener la altura.

VII-H. Síntesis Directa

VII-H1. Síntesis directa (Truxal–Ragazzini): Se parte del modelo discreto identificado de la planta:

$$G_{ZAS}(z) = \frac{-0.0001205\, z^{-1} + 0.0002415\, z^{-2} - 0.0001209\, z^{-3}}{1 - 2.994\, z^{-1} + 2.989\, z^{-2} - 0.9944\, z^{-3}}. \tag{2}$$

El método de Truxal–Ragazzini consiste en especificar explícitamente una dinámica deseada en lazo cerrado $G_{cl}(z)$ y obtener el controlador a partir de la relación:

$$G_{cl}(z) = \frac{C(z)G_{ZAS}(z)}{1 + C(z)G_{ZAS}(z)} \quad \Longrightarrow \quad C(z) = \frac{1}{G_{ZAS}(z)} \frac{G_{cl}(z)}{1 - G_{cl}(z)}. \tag{3}$$

Este procedimiento implica una inversión explícita del modelo de la planta, por lo que el diseño depende fuertemente de la exactitud del modelo identificado.

VII-H1a. Método 1: respuesta deadbeat: Como primera aproximación se adoptó una dinámica deseada del tipo *deadbeat*, definida por:

$$G_{cl}(z) = z^{-1}. \tag{4}$$

Esta elección implica que la salida alcance el valor deseado en un único período de muestreo, anulando el error en el menor tiempo posible.

Reemplazando en (3) se obtiene:

$$C_1(z) = \frac{1}{G_{ZAS}(z)} \frac{z^{-1}}{1 - z^{-1}} = \frac{1}{G_{ZAS}(z)} \frac{1}{z - 1}. \tag{5}$$

Se observa que el controlador resultante contiene explícitamente la inversa de la planta y un polo adicional en $z = 1$, lo que anticipa posibles problemas de magnitud del esfuerzo de control.

VII-H2. Método 2: Deadbeat ripple-free: Como alternativa se evaluó la variante *ripple-free*, cuyo controlador obtenido es:

$$C_2(z) = \frac{10.43038\, z^2 - 2.7635\, z + 3.3344}{-0.33662\, z^2 - 0.66338\, z + 1}. \tag{6}$$

Multiplicando numerador y denominador por (-1) y normalizando el coeficiente líder del denominador, se obtiene:

$$C_2(z) = \frac{-30.9856\, z^2 + 8.2096\, z - 9.9055}{z^2 + 1.9707\, z - 2.9707}. \tag{7}$$

Para implementación digital resulta conveniente expresarlo en términos de z^{-1}:

$$C_2(z) = \frac{-30.9856 + 8.2096\, z^{-1} - 9.9055\, z^{-2}}{1 + 1.9707\, z^{-1} - 2.9707\, z^{-2}}. \tag{8}$$

VII-H3. Resultados y Limitaciones Prácticas: Las simulaciones mostraron que ambos controladores demandan esfuerzos de control extremadamente elevados, alcanzando valores del orden de:

$$|u_{\text{máx}}| \sim 10^{29},$$

lo cual excede ampliamente las capacidades del actuador físico.

En la implementación real, la señal de control corresponde a una señal PWM tipo servo a 50 Hz acotada en el rango:

$$u \in [1000, 2000]\ \mu s. \tag{9}$$

La magnitud desproporcionada del esfuerzo se explica por:

- La inversión explícita del modelo $G_{ZAS}(z)$.
- La presencia de polos cercanos a $z = 1$ en la planta.
- Alta sensibilidad a pequeñas incertidumbres del modelo.
- Cancelaciones exactas requeridas por el diseño.

En particular, la inversión de dinámicas cercanas al borde del círculo unitario produce amplificaciones significativas en la señal de control, haciendo que el diseño sea extremadamente sensible a variaciones como cambios en la tensión de batería, fricción, efectos aerodinámicos y dinámica no modelada del conjunto ESC–motor–hélice.

Por estas razones, si bien la síntesis directa resulta valiosa desde el punto de vista conceptual y didáctico, no se considera viable para implementación experimental en la planta real.

En consecuencia, para la etapa práctica se priorizan estrategias de menor orden y mayor robustez, que contemplen explícitamente las limitaciones del actuador y la saturación de la señal de control. Ecuación general:

$$C(z) = \frac{1}{G(z)} \frac{G_{cl}(z)}{1 - G_{cl}(z)}$$

Dinámica deseada:

$$G_{cl}(z) = \textbf{[Completar]}$$

Observaciones:

- Sensibilidad al modelo.
- Elevado esfuerzo de control.
- Limitaciones prácticas.

VII-I. Comparación entre Métodos Clásicos

- PID: **[Completar fortalezas y debilidades]**
- Lugar de Raíces: **[Completar]**
- Bode: **[Completar]**
- Síntesis Directa: **[Completar]**

Observación general:

[Completar: limitaciones estructurales del enfoque clásico]

VIII. Métodos Modernos de Control

VIII-A. *Introducción*

A diferencia de los métodos clásicos basados en funciones de transferencia y análisis en frecuencia, los métodos modernos de control se fundamentan en la representación en espacio de estados del sistema dinámico.

En este enfoque, la dinámica se describe mediante:

$$x_{k+1} = Ax_k + Bu_k \tag{10}$$

$$y_k = Cx_k + Du_k \tag{11}$$

donde $x_k \in \mathbb{R}^n$ es el vector de estados, u_k la entrada de control y y_k la salida medida.

Este formalismo permite:

- Diseñar realimentación directa de estados.
- Ubicar polos del sistema de manera sistemática.
- Formular problemas de control óptimo.
- Incorporar estimadores de estado.

VIII-A1. Modelo en espacio de estados continuo: Las matrices del modelo continuo en representación estado–espacio son:

$$F = \begin{bmatrix} -5.6102 & -3.5055 & -0.0314 \\ 4.0000 & 0 & 0 \\ 0 & 0.0312 & 0 \end{bmatrix}$$

$$G = \begin{bmatrix} 16 \\ 0 \\ 0 \end{bmatrix}$$

$$H = \begin{bmatrix} -0.0076 & 0.0064 & 8.9975 \end{bmatrix}$$

$$J = \begin{bmatrix} 0 \end{bmatrix}$$

VIII-A2. Modelo en Espacio de Estados: El modelo discreto obtenido:

$$A = e^{FT_s}, \qquad B = F^{-1}\left(e^{FT_s} - I\right)G, \qquad C = H,$$
$$D = J$$

Tiempo de muestreo depende de la práctica

VIII-A3. Análisis de Controlabilidad y Observabilidad: Matriz de controlabilidad:

$$\mathcal{C} = [G\ FG\ F^2G\ \ldots F^{n-1}G]$$

$$\text{rank}(\mathcal{C}) = \textbf{[3]}$$

Matriz de observabilidad:

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

$$\text{rank}(\mathcal{O}) = \textbf{[3]}$$

Conclusión estructural:

Dado que el rango de ambas matrices coincide con el orden del sistema, se concluye que el modelo es completamente controlable y completamente observable.

Desde el punto de vista físico, esto implica que:

- Existe una combinación adecuada de la señal de entrada que permite influenciar todos los estados internos del sistema.
- La salida medida contiene información suficiente para reconstruir completamente el vector de estados mediante un observador.

En consecuencia, el modelo identificado resulta estructuralmente apto para el diseño de control por realimentación de estados, ubicación de polos, LQR y estimación de estados mediante observador de Luenberger o filtro de Kalman.

No obstante, la validez práctica de dicha síntesis en tiempo discreto depende de que el tiempo de muestreo T_s sea suficientemente pequeño para capturar la dinámica relevante del sistema, evitando aliasing. En particular, se asume que las componentes significativas de la señal de salida y de las perturbaciones se encuentran por debajo de la frecuencia de Nyquist $f_N = \frac{1}{2T_s}$, y que el acondicionamiento analógico (filtrado anti-alias) es consistente con esta hipótesis.

VIII-B. *Ubicación Arbitraria de Polos*

VIII-C. *Realimentación de Estados y Estimación*

VIII-C1. Realimentación de Estados: Considerando el modelo discreto del sistema:

$$x_{k+1} = Ax_k + Bu_k$$

$$y_k = Cx_k$$

se propone una ley de control por realimentación de estados:

$$u_k = -Kx_k$$

lo que conduce a la dinámica en lazo cerrado:

$$x_{k+1} = (A - BK)x_k$$

El diseño por ubicación arbitraria de polos consiste en determinar la matriz de ganancia K tal que:

$$\lambda(A - BK) = \{p_1, p_2, \ldots, p_n\}$$

siendo $|p_i| < 1$ condición necesaria para estabilidad discreta.

La determinación de K puede realizarse mediante el método de Ackermann o utilizando la función `place()` de MATLAB, siempre que el sistema sea completamente controlable.

VIII-C2. Estimador de Estados: En situaciones donde no todos los estados son medibles, se introduce un estimador para reconstruir el vector de estados a partir de la entrada y la salida medida.

El estimador discreto se define como:

$$\hat{x}_{k+1} = A\hat{x}_k + Bu_k + L(y_k - \hat{y}_k)$$

donde:

$$\hat{y}_k = C\hat{x}_k$$

La dinámica del error de estimación:

$$e_k = x_k - \hat{x}_k$$

queda gobernada por:

$$e_{k+1} = (A - LC)e_k$$

Por lo tanto, la convergencia del estimador depende de la ubicación de los autovalores de la matriz $A - LC$, que pueden ser fijados arbitrariamente siempre que el sistema sea observable:

$$\lambda(A - LC) = \{p_{obs,1}, \dots, p_{obs,n}\}$$

VIII-C3. Principio de Separación: Cuando se combinan realimentación de estados y estimación, la ley de control adopta la forma:

$$u_k = -K\hat{x}_k$$

y la dinámica total del sistema presenta autovalores dados por la unión de los polos del controlador y los polos del estimador:

$$\lambda_{\text{total}} = \lambda(A - BK) \cup \lambda(A - LC)$$

Este resultado, conocido como principio de separación, permite diseñar independientemente el controlador y el estimador.

VIII-D. Control Integrador

VIII-D1. Ubicación arbitraria de polos con Integrador: En esta sección se implementa el procedimiento propuesto en Ogata (Ec. 6.19) para incorporar acción integral al sistema en espacio de estados, utilizando únicamente un integrador externo y realimentación de estados estimados.

La planta discreta utilizada (orden $n = 3$) se obtuvo mediante discretización por ZOH con período de muestreo:

$$T_s = 0.02 \text{ s} \qquad (50 \text{ Hz})$$

VIII-D1a. Sistema aumentado con integrador: Para eliminar el error en régimen permanente frente a referencias constantes, se define el estado integral:

$$v_{k+1} = v_k + (r_k - y_k)$$

donde $y_k = Cx_k$.

Siguiendo el desarrollo de Ogata, se construye el sistema aumentado:

$$\hat{A} = \begin{bmatrix} A & B \\ 0 & 0 \end{bmatrix}, \qquad \hat{B} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

y se ubican los polos deseados del sistema aumentado:

$$p_{\text{ctrl}} = \{0.95 \pm 0.15j,\ 0.98\}, \qquad p_i = 0.96$$

De esta forma se obtienen las ganancias equivalentes:

$$K_2 = \begin{bmatrix} 0.7820721 & -0.5678709 & 0.3779423 \end{bmatrix}$$

$$K_1 = 0.1469358$$

donde:

- K_2 actúa sobre el estado estimado $\hat{x}_k$. - K_1 actúa sobre el estado integral v_k.

La ley de control implementada es:

$$u_k = K_1 v_k - K_2 \hat{x}_k$$

Los polos del sistema aumentado cerrado fueron correctamente ubicados (según `place`) en:

$$p_{K_{21}} = 13$$

(lo que confirma la correcta asignación interna en MATLAB).

VIII-D1b. Diseño del observador: Se diseñaron dos variantes de observador con los mismos polos deseados:

$$p_{\text{obs}} = \{0.8 \pm 0.25j,\ 0.9\}$$

VIII-D1c. Observador predictor: Dinámica del error:

$$A - L_{\text{pred}} C$$

Ganancia obtenida:

$$L_{\text{pred}} = \begin{bmatrix} 167.3535 \\ 253.6822 \\ 86.0310 \end{bmatrix}$$

$$p_{L_{\text{pred}}} = 12$$

VIII-D1d. Observador actual: Dinámica del error:

$$A - L_{\text{actual}} CA$$

Ganancia obtenida:

$$L_{\text{actual}} = \begin{bmatrix} 126.8411 \\ 172.0621 \\ 45.1301 \end{bmatrix}$$

$$p_{L_{\text{actual}}} = 13$$

VIII-D1e. Estructura completa del sistema aumentado cerrado: Para analizar explícitamente la dinámica del sistema completo (planta + integrador + control), el script construye el modelo aumentado:

$$A_{\text{aug}} = \begin{bmatrix} A & B \\ K_2 - K_2 A - K_1 CA & 1 - K_2 B - K_1 CB \end{bmatrix}$$

$$B_{\text{aug}} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ K_1 \end{bmatrix}$$

$$C_{\text{aug}} = \begin{bmatrix} C & 0 \end{bmatrix}$$

Este sistema representa la dinámica cerrada resultante de aplicar la ley de control integral derivada mediante Ogata 6.19.

Además, para comparar la dinámica del error de estimación, se construye el sistema:

$$A_{\text{obs}} = A - L_{\text{actual}} CA$$

y se superponen ambos mapas de polos mediante `pzmap`.

VIII-D2. Resultados de simulación: Se evaluó el comportamiento del sistema para una referencia escalón de 25 unidades, comparando:

- Observador predictor
- Observador actual
- Esfuerzo de control
- Evolución del estado integral

Las figuras correspondientes se muestran a continuación:

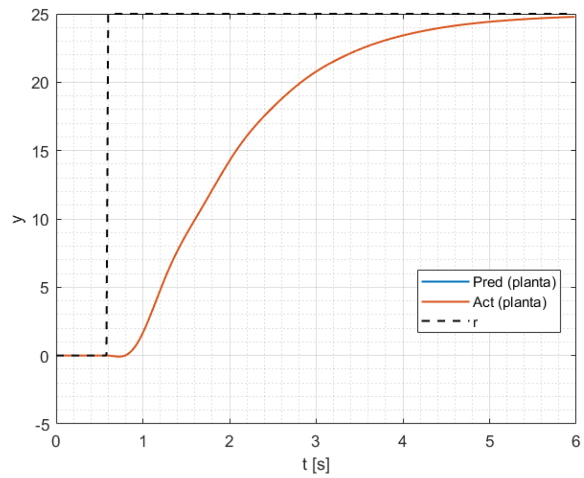


Figura 26. Respuesta temporal $y(t)$ con integrador — comparación predictor vs actual.

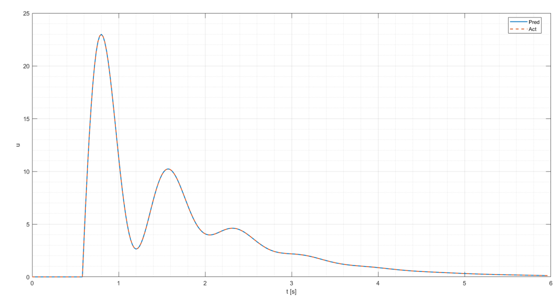


Figura 27. Esfuerzo de control $u(t)$ con integrador.

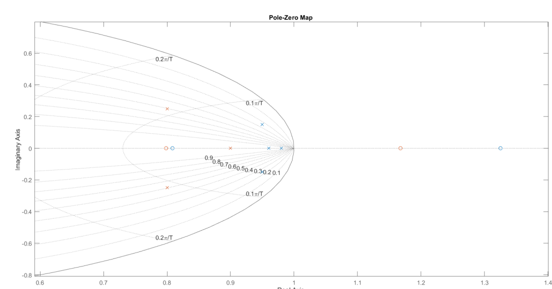


Figura 28. Mapa de polos: planta, sistema aumentado y dinámica del observador.

VIII-D3. Resultados experimentales (implementacion practica): Ademas de las simulaciones, se realizaron ensayos sobre la planta real implementando el controlador con accion integral y las dos variantes de observador (predictor y actual). Las siguientes figuras muestran los resultados practicos registrados para una referencia tipo escalon, utilizando la misma configuracion de ganancias reportada previamente.

VIII-D4. Discusión: El método de Ogata 6.19 permite incorporar acción integral sin modificar la estructura original

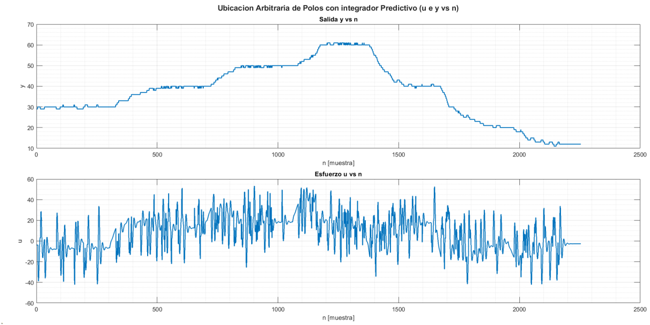


Figura 29. Resultado experimental con observador predictor (implementacion practica).

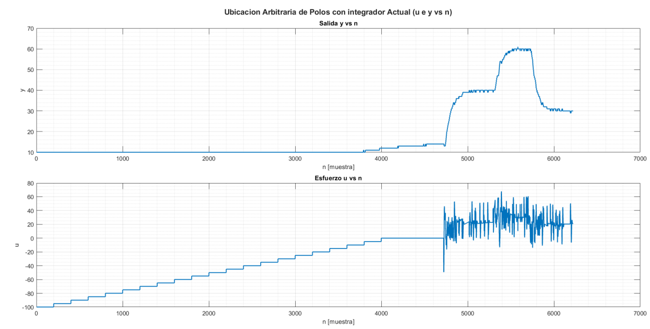


Figura 30. Resultado experimental con observador actual (implementacion practica).

de la planta, utilizando un estado adicional y realimentación adecuada.

Se observa que:

- El integrador elimina el error en régimen permanente.
- La dinámica del observador es más rápida que la del sistema controlado.
- Ambos esquemas (predictor y actual) resultan estables, con polos dentro del círculo unidad.
- El sistema aumentado conserva estabilidad global.

El código completo utilizado para esta sección se incluye en el Apéndice C.

VIII-E. Control Óptimo (LQR)

Se diseñó un regulador óptimo discreto de tipo LQR a partir del modelo en espacio de estados discretizado mediante retención de orden cero (ZOH) con tiempo de muestreo:

$$T_s = 0.01 \text{ s}$$

El objetivo del diseño es minimizar el funcional de costo cuadrático infinito:

$$J = \sum_{k=0}^{\infty} (x_k^T Q x_k + u_k^T R u_k)$$

donde $Q \succeq 0$ penaliza la energía de los estados y $R \succ 0$ penaliza el esfuerzo de control.

VIII-E0a. Matrices de ponderación: Se adoptó una sintonización práctica basada en escalas típicas del experimento: un cambio de referencia del orden de $\Delta y \approx 20$ y una restricción del mando aproximadamente $|u| \lesssim 300$ (en unidades del actuador). El diseño penaliza principalmente la salida mediante $C^T C$, agregando un término pequeño para asegurar buena condición numérica:

$$Q = w_y (C^T C) + 10^{-8} I_n, \qquad R = \frac{w_u}{u_{\text{máx}}}$$

con $w_y = 20$, $w_u = 500$ y $u_{\text{máx}} = 300$. Para el modelo discretizado utilizado, las matrices resultantes fueron:

$$Q = \begin{bmatrix} 6.8188 \times 10^{-3} & -6.9883 \times 10^{-3} & 7.0544 \times 10^{-3} \\ -6.9883 \times 10^{-3} & 7.1620 \times 10^{-3} & -7.2298 \times 10^{-3} \\ 7.0544 \times 10^{-3} & -7.2298 \times 10^{-3} & 7.2982 \times 10^{-3} \end{bmatrix}$$

$$R = 1.6667$$

VIII-E0b. Ecuación de Riccati discreta y ganancia óptima: La solución del problema se obtiene resolviendo la ecuación de Riccati discreta:

$$P = A^T P A - A^T P B (R + B^T P B)^{-1} B^T P A + Q$$

La ganancia óptima resulta:

$$K = (R + B^T P B)^{-1} B^T P A$$

Para el modelo discretizado se obtuvo:

$$K = \begin{bmatrix} 0.5119095 & -0.4937470 & 0.4765571 \end{bmatrix}$$

VIII-E0c. Polos del sistema en lazo cerrado: La estabilidad se verifica mediante los polos del sistema:

$$A_{\text{cl}} = A - BK$$

Los polos obtenidos fueron:

$$\lambda(A - BK) = \begin{matrix} 0.9728458 + 0.0327300\, j \\ 0.9728458 - 0.0327300\, j \\ 0.9663930 \end{matrix}$$

Se observa que todos los polos se encuentran dentro del círculo unidad, garantizando estabilidad discreta.

VIII-E0d. Referencia al apéndice de implementación: El código completo utilizado para:

- cargar y discretizar la planta,
- definir Q y R mediante parámetros *knobs* (w_y, w_u),
- calcular K y verificar controlabilidad,
- diseñar observadores (predictor y actual) por ubicación de polos,
- calcular el prefiltro N_{bar} para seguimiento,
- y simular el lazo con/sin saturación, con/sin ruido, y en doble precisión y `float32`,

se incluye en el Apéndice A.

VIII-E1. Simulación del LQR con observador, saturación y ruido: Con el fin de aproximar el comportamiento del sistema real y evitar conclusiones optimistas, se implementó un entorno de simulación que contempla:

- **Observador de estados** en dos variantes:
  - *Predictor*: corrige usando y_k.
  - *Actual*: corrige usando y_{k+1}.
- **Prefiltro de referencia** N_{bar} para mejorar el seguimiento de referencia en lazo cerrado (estructura $u_k = N_{\mathrm{bar}} r_k - K\hat{x}_k$).
- **Saturación** del actuador con límite $\pm u_{\mathrm{máx}}$ para representar las restricciones del PWM/ESC.
- **Ruido** configurable para emular incertidumbres del experimento:
  - Ruido de medición: $y_{\mathrm{meas}} = y_{\mathrm{true}} + v_y$, con $v_y \sim \mathcal{N}(0, \sigma_y^2)$.
  - Ruido del actuador (jitter) y cuantización: $u_{\mathrm{app}} = \mathrm{sat}\left(\mathrm{quant}(u_{\mathrm{cmd}} + v_u)\right)$.
  - Ruido de proceso: $x_{k+1} = Ax_k + Bu_k + w_k$ (configurable como prueba de estrés).
- **Comparación numérica** entre simulación en doble precisión y `float32` para anticipar efectos de implementación embebida.

En particular, para evitar comparaciones sesgadas, las secuencias de ruido se pre-generaron y se reutilizaron de forma idéntica tanto en doble precisión como en `float32`.

VIII-E1a. Observadores y ubicación de polos: Los observadores se diseñaron mediante ubicación arbitraria de polos en el plano-z, definiendo polos deseados:

$$p_{\mathrm{obs}} = \{0.8 \pm 0.25j,\ 0.9\}$$

y obteniendo:

$$K_{e,\mathrm{pred}} = \mathrm{place}(A^T, C^T, p_{\mathrm{obs}})^T \qquad K_{e,\mathrm{act}} = \mathrm{place}(A^T, (CA)^T, p_{\mathrm{obs}})^T$$

VIII-E1b. Prefiltro N_{bar}: Para mejorar el seguimiento de referencia (SISO), se calculó un prefiltro N_{bar} tal que la ganancia estática equivalente sea adecuada, utilizando la rutina auxiliar `refi`:

$$u_k = N_{\mathrm{bar}}\, r_k - K\, \hat{x}_k$$

VIII-E1c. Diagnósticos para evitar autoengaño: Se incluyeron verificaciones directas sobre el ruido inyectado:

$$e_y(k) = y_{\mathrm{meas}}(k) - y_{\mathrm{true}}(k)$$

y se reportaron métricas como RMS(e_y) y conteo de saturaciones en u, para confirmar que las perturbaciones y límites realmente están actuando sobre el lazo.

VIII-E1d. Referencia al apéndice: El script completo de simulación (incluyendo funciones locales `sim_obs_loop`, `plot_zplane`, `refi` y el *fallback* `dlqr_iter_nolic`) se incluye en el Apéndice A.

VIII-F. Filtro de Kalman

VIII-F1. Modelo con Ruido: Para incorporar incertidumbre y modelar explícitamente la presencia de perturbaciones no modeladas y ruido del sensor, se adopta la siguiente representación estocástica discreta:

$$x_{k+1} = Ax_k + Bu_k + w_k$$

$$y_k = Cx_k + v_k$$

donde:

- $w_k \sim \mathcal{N}(0, Q)$ representa el ruido de proceso, asociado a dinámica no modelada, perturbaciones aerodinámicas y simplificaciones del modelo identificado.
- $v_k \sim \mathcal{N}(0, R)$ representa el ruido de medición, proveniente del sensor láser de distancia.

VIII-F1a. Estimación de R (ruido de medición): La varianza del ruido de medición se obtuvo mediante ensayos empíricos, midiendo la dispersión de la señal del sensor con la planta en reposo. Siendo σ_v la desviación estándar medida (en cm), se adopta:

$$R = \sigma_v^2$$

En los ensayos realizados se obtuvo:

$$\sigma_v = 2.043 \text{ cm} \qquad \Longrightarrow \qquad R = 4.174 \text{ cm}^2$$

VIII-F1b. Parametrización y sintonización de Q (ruido de proceso): La matriz de covarianza del ruido de proceso se parametrizó como:

$$Q = q\, I_n$$

donde q es un escalar positivo ajustable e I_n es la matriz identidad de dimensión n. El valor de q se determinó mediante consistencia estadística de la innovación normalizada (ver Apéndice A).

El valor óptimo obtenido fue:

$$q = 50.8022 \times 10^{-3} \qquad \Longrightarrow \qquad Q = q\, I_n$$

VIII-F2. Filtro de Kalman en régimen permanente: Se utilizó un estimador de Kalman discreto en su variante *current estimator* (`kalman(...,'current')`). La ganancia en régimen permanente L se obtiene a partir de la solución estacionaria P de la ecuación de Riccati discreta:

$$P = APA^T - APC^T \left(CPC^T + R\right)^{-1} CPA^T + Q$$

y la ganancia queda:

$$L = APC^T \left(CPC^T + R\right)^{-1}$$

La ganancia obtenida en MATLAB para el modelo discretizado fue:

$$L = \begin{bmatrix} 97.4968 \\ 192.3014 \\ 94.7820 \end{bmatrix}$$

VIII-F2a. Polos del observador: La estabilidad del observador se verifica a partir de los polos del sistema estimador. Para el *current estimator*, la dinámica del error queda determinada por $A - LCA$, por lo que se reportan los polos:

$$\lambda(A - LCA) = \begin{matrix} 0.9730274 + 0.0308455\,j \\ 0.9730274 - 0.0308455\,j \\ 0.9694430 \end{matrix}$$

VIII-F3. Control integral y realimentación de estados (LQI): Con el objetivo de eliminar el error en régimen permanente ante referencias tipo escalón, se incorporó un integrador de error. Definiendo el estado integral ξ_k:

$$\xi_{k+1} = \xi_k + (r_k - y_k)$$

Se diseñó una ley de control tipo LQI:

$$u_k = -K_x\, \hat{x}_k + K_i\, \xi_k$$

donde $\hat{x}_k$ proviene del estimador de Kalman.

Los valores obtenidos mediante `dlqr` (con pesos heurísticos) fueron:

$$K_x = \begin{bmatrix} 16.5343 & -15.8046 & 15.1271 \end{bmatrix}$$

$$K_i = 3.2067$$

VIII-F3a. Polos de la planta y del lazo cerrado: Los polos de la planta discretizada (sin control) fueron:

$$\lambda(A) = \begin{matrix} 0.9999972 \\ 0.9720409 + 0.0241158\,j \\ 0.9720409 - 0.0241158\,j \end{matrix}$$

Los polos del lazo cerrado del sistema aumentado (planta + integrador + control) fueron:

$$\lambda(A_{\text{cl}}) = \begin{matrix} 0.9735939 + 0.0318308\,j \\ 0.9735939 - 0.0318308\,j \\ 0.9671966 \\ 0 \end{matrix}$$

VIII-F4. Sistema aumentado usado para `pzmap`: Además del cálculo teórico de polos del lazo cerrado, en el script de validación se armó explícitamente un *sistema aumentado discreto* para analizar la estabilidad en el plano-z con `pzmap`. Este sistema corresponde a la dinámica conjunta de planta e integrador, con la ley de control LQI aplicada.

En el código se definieron las matrices del sistema aumentado como:

$$A_{\text{aug}} = \begin{bmatrix} A & B \\ K_x - K_x A - K_i C A & 1 - K_x B - K_i C B \end{bmatrix}$$

$$B_{\text{aug}} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ K_i \end{bmatrix}$$

$$C_{\text{aug}} = \begin{bmatrix} C & 0 \end{bmatrix} \qquad D_{\text{aug}} = 0$$

y se construyó el modelo:

```
sysDaug = ss(Aaug,Baug,Caug,0,Ts).
```

Finalmente, se utilizó `pzmap(sysDaug)` para visualizar los polos del sistema aumentado en el plano-z y verificar que la dinámica resultante permanezca estable (polos dentro del círculo unidad) en el caso nominal.

VIII-F4a. Nota: Esta construcción se incluye tal cual se implementa en el script, con el objetivo de replicar el análisis práctico realizado y obtener directamente el mapa de polos asociado a la estructura planta + integrador + control.

VIII-F5. Nota práctica sobre saturación: En la implementación práctica se aplica saturación al mando u_k para respetar los límites del actuador (PWM). La presencia de saturación puede degradar el cumplimiento exacto de la dinámica diseñada y producir integrador acumulado, por lo que en firmware se complementa con estrategias de anti-windup cuando corresponde.

VIII-F6. Resultados con ruido y validación: Para evaluar el desempeño se simuló el sistema incluyendo tanto ruido de medición (como dispersión del sensor) como ruido de proceso (dinámica no modelada). Se comparó:

- Respuesta de salida con ruido, y seguimiento de referencia.
- Esfuerzo de control requerido en presencia de ruido.
- Estabilidad del sistema mediante mapa de polos y ceros.

Las siguientes figuras muestran los resultados obtenidos:

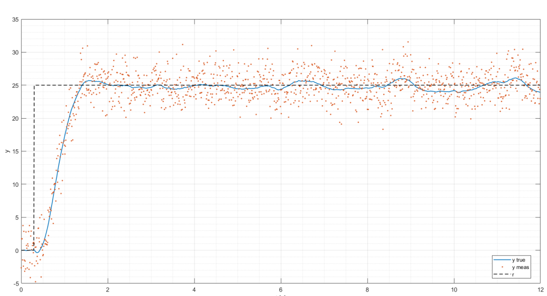


Figura 31. Respuesta temporal de la salida en presencia de ruido (medición y proceso) y comparación con la referencia.

El código completo utilizado para calcular L, K_x, K_i, los polos de la planta, los polos del observador y los polos del lazo cerrado se incluye en el Apéndice B.

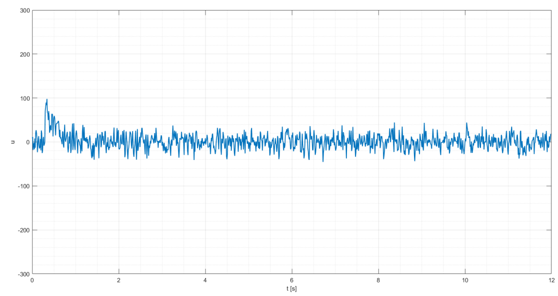


Figura 32. Esfuerzo de control (señal de mando) en presencia de ruido.

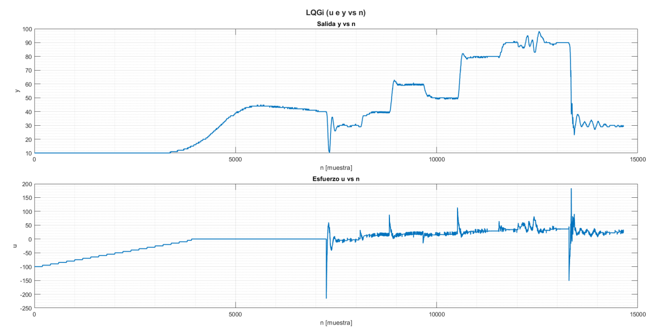


Figura 33. Resultado experimental obtenido con la implementación práctica sobre la planta real.

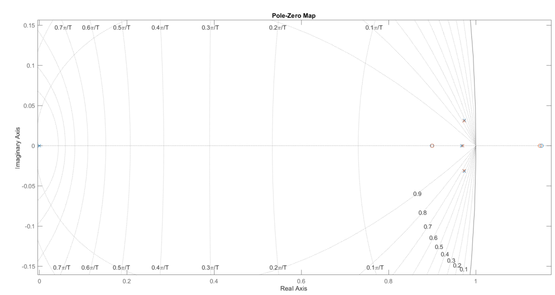


Figura 34. Mapa de polos y ceros. En naranja se muestran los polos del observador y en azul los polos asociados a la planta/sistema.

IX. Conclusiones

Apéndice A
Parámetros del controlador

Apéndice B
Códigos MATLAB

Apéndice C
Espacio de estado con integrador)

En este apendice se incluye el script completo utilizado para implementar el metodo de Ogata 6.19 con un integrador externo (accion integral) y dos variantes de observador en tiempo discreto: predictor y actual. El codigo calcula las ganancias K_1 y K_2 mediante `place` sobre el sistema aumentado, disena los observadores con los polos p_{obs}, y genera las simulaciones y graficos asociados.

```
%% ========================
%  OGATA 6.19 − SOLO INTEGRADOR
%  + Observador PREDICTOR y ACTUAL
%  (Planta discreta desde .mat, orden 3)
%% ========================
close all; clear; clc

%% ========================
% 1) CARGA + DISCRETIZACION
%% ========================
S = load('planta␣(1).mat');

if isfield(S,'plantaC')
plantaC = S.plantaC;
elseif isfield(S,'sysC')
plantaC = S.sysC;
else
error('No␣encuentro␣"plantaC"␣ni␣"sysC"␣dentro␣de␣planta␣(1).mat');
end

Ts   = 1/50;                % ajusta si queres
sysD = c2d(plantaC, Ts, 'zoh');
[A,B,C,D] = ssdata(ss(sysD));

n = size(A,1);
if n ~= 3
error('Esta␣plantilla␣asume␣planta␣de␣orden␣3.␣n=%d', n);
end

% Asegurar SISO (una salida)
if size(C,1) ~= 1
C = C(1,:);
end

fprintf('Ts=%.9f␣|␣n=%d\n', Ts, n);

%% ========================
% 2) POLOS DESEADOS (FIJOS)
%% ========================
p_ctrl = [0.95 + 0.15i, 0.95 − 0.15i, 0.98];    % control
p_obs  = [0.8 + 0.25i, 0.8 − 0.25i, 0.9];
%p_obs  = [0.4 + 0.25i, 0.4 − 0.25i, 0.6];       % observador
p_i    = 0.96;                                   % polo integrador

%% ========================
% 3) OGATA 6.19 (K1 y K2) TAL CUAL
%% ========================
m = 1;

Ahat = [A B; zeros(m,n+m)];
Bhat = [zeros(n,m); eye(m)];
Chat = [C zeros(1,m)];

polos_i = [p_ctrl p_i];           % 3 polos control + 1 integrador
[Khat,pKhat]    = place(Ahat, Bhat, polos_i);

Aux  = [A−eye(size(A))  B;
C∗A               C∗B];

K2K1 = (Khat + [zeros(1,n) eye(m)]) / Aux;
K2   = K2K1(1,1:n);                % sobre xhat (1 x3)
K1   = K2K1(1,n+1:end);            % sobre v   (1 x1)

fprintf('\nK2␣=␣'); disp(K2);
fprintf('K1␣=␣'); disp(K1); fprintf('\tpK21␣=␣'); disp(pKhat);

%% ========================
% 4) OBSERVADORES (PRED / ACT) − mismo p_obs
%% ========================
[L_pred, pL_pred]     = place(A', C',     p_obs); L_pred   = L_pred.';     % predictor: A − L∗C
[L_actual, pL_actual] = place(A', (C∗A)', p_obs); L_actual = L_actual.';   % actual:    A − L∗C∗A

fprintf('L_pred␣=␣'); disp(L_pred); fprintf('\tpL_pred␣=␣'); disp(pL_pred);
fprintf('\nL_actual␣=␣'); disp(L_actual); fprintf('\tpL_actual␣=␣'); disp(pL_actual);

%% ========================
% 5) SIMULACION (SOLO CON INTEGRADOR)
%% ========================
N  = 300;
t  = 0:Ts:(N−1)∗Ts;

r = ones(1,N)∗25;
r(1:30) = 0;

% ruido en medicion (igual que tu estilo)
w = [zeros(1,75), 0.01∗ones(1,N−75)];

% −−−−− PREDICTIVO CON integrador −−−−−

X_p    = zeros(n,N);       % estado real
Xh_p   = zeros(n,N);       % estado estimado
V_p    = zeros(1,N);       % integrador
U_p    = zeros(1,N);       % control

% −−−−− ACTUAL CON integrador −−−−−
X_a    = zeros(n,N);
Xh_a   = zeros(n,N);
```

```
V_a     = zeros(1,N);
U_a     = zeros(1,N);

for k = 1:N-1
rk = r(k);

%% ===== 1) PREDICTIVO + integrador (MISMA FORMA) =====
y_p           = C*X_p(:,k);
V_p(k+1)      = V_p(k) + (rk - y_p);     % integrador real
u_p           = K1*V_p(k+1) - K2*Xh_p(:,k);
X_p(:,k+1)    = A*X_p(:,k) + B*u_p; % + w(k);
Xh_p(:,k+1)  = A*Xh_p(:,k) + B*u_p + L_pred*(y_p - C*Xh_p(:,k));
U_p(k)        = u_p;

%% ===== 2) ACTUAL + integrador (MISMA FORMA) =====
y_a           = C*X_a(:,k);
V_a(k+1)      = V_a(k) + (rk - y_a);
u_a           = K1*V_a(k+1) - K2*Xh_a(:,k);
X_a(:,k+1)    = A*X_a(:,k) + B*u_a; % + w(k);
y_next        = C*X_a(:,k+1); % + w(k);     % misma muestra de ruido
z_a           = A*Xh_a(:,k) + B*u_a;
Xh_a(:,k+1)  = z_a + L_actual*(y_next - C*z_a);
U_a(k)        = u_a;
end

%% ========================
% 6) PLOTS RAPIDOS
%% ========================
figure('Name','y(t) - integrador + predictor vs actual');
plot(t, (C*X_p).', t, (C*X_a).', t, r, 'k--','LineWidth',1.2);
grid on; grid minor;
xlabel('t [s]'); ylabel('y');
legend('Pred (planta)','Act (planta)','r','Location','best');

figure('Name','u(t) - integrador + predictor vs actual');
plot(t(1:end-1), U_p(1:end-1), 'LineWidth',1.4); hold on;
plot(t(1:end-1), U_a(1:end-1), '--', 'LineWidth',1.4);
grid on; grid minor;
xlabel('t [s]'); ylabel('u');
legend('Pred','Act','Location','best');

figure('Name','v(t) - integrador (estado integral)');
plot(t, V_p, 'LineWidth',1.4); hold on;
plot(t, V_a, '--', 'LineWidth',1.4);
grid on; grid minor;
xlabel('t [s]'); ylabel('v');
legend('Pred','Act','Location','best');

%% ========================
% 7) MAPA DE POLOS (resumen)
%% ========================
figure('Name','Z-plane - planta, aumentado e info obs','Position',[100 100 1000 400]);

subplot(1,3,1); zgrid; hold on; grid on; box on;
p_planta = eig(A);
plot(real(p_planta), imag(p_planta), 'ko','MarkerSize',8,'LineWidth',1.4);
title('Planta (A)'); xlabel('Re\{z\}'); ylabel('Im\{z\}');

subplot(1,3,2); zgrid; hold on; grid on; box on;
p_aug     = eig(Ahat);
p_aug_cl = eig(Ahat - Bhat*Khat);
plot(real(p_aug),    imag(p_aug),    'm^','MarkerSize',8,'LineWidth',1.4);
plot(real(p_aug_cl), imag(p_aug_cl), 'c*','MarkerSize',10,'LineWidth',1.6);
title('Aumentado: Ahat y Ahat-BhatKhat'); xlabel('Re\{z\}'); ylabel('Im\{z\}');

subplot(1,3,3); zgrid; hold on; grid on; box on;
p_obs_pred = eig(A - L_pred*C);
p_obs_act  = eig(A - L_actual*C*A);
plot(real(p_obs_pred), imag(p_obs_pred), 'bs','MarkerSize',9,'LineWidth',1.5);
plot(real(p_obs_act),  imag(p_obs_act),  'rx','MarkerSize',9,'LineWidth',1.5);
title('Obs error'); xlabel('Re\{z\}'); ylabel('Im\{z\}');
legend('A-LC (pred)','A-LCA (act)','Location','best');

sgtitle('Ogata 6.19 - Solo integrador + predictor/actual');

Aaug = [A,B;...
K2-K2*A-K1*C*A,1-K2*B-K1*C*B];
Baug = [0;0;0;K1];
Caug    = [C,0];
D = 0;
sysDaug = ss(Aaug,Baug,Caug,D,Ts);
sysObs = ss((A-L_actual*C*A),B,C,D,Ts);
figure; pzmap(sysDaug); grid on; zgrid; hold on; pzmap(sysObs);
```

Listing 1. OGATA 6.19 - SOLO INTEGRADOR + Observador PREDICTOR y ACTUAL (planta orden 3).

APÉNDICE

En este apendice se incluye el script MATLAB utilizado para: (i) cargar y discretizar la planta identificada, (ii) definir las ponderaciones Q y R del LQR, (iii) calcular la ganancia optima K (via `dlqr`/`dare`), (iv) disenar observadores (predictor y actual) por ubicacion arbitraria de polos, (v) calcular el prefiltro N_{bar} para seguimiento de referencia, y (vi) simular el lazo con/sin saturacion y con/sin ruido (incluyendo comparacion double vs `float32`).

```matlab
close all; clear; clc

%% ========================
% 1) CARGA + DISCRETIZACION
%% ========================
S = load('planta (1).mat');

if isfield(S,'plantaC')
plantaC = S.plantaC;
elseif isfield(S,'sysC')
plantaC = S.sysC;
else
error('No encuentro "plantaC" ni "sysC" dentro de planta (1).mat');
end

Ts   = 1/100;                % sample time
sysD = c2d(plantaC, Ts, 'zoh');
[A,B,C,D] = ssdata(ss(sysD));
n = size(A,1);

fprintf('Ts=%.9f | n=%d\n', Ts, n);
disp('A='); disp(A); disp('B='); disp(B); disp('C='); disp(C); disp('D='); disp(D);

%% ========================
% 2) PARAMETROS + PESOS (LQR)
%% ========================
% Observador (z-plane)
p_obs  = [0.8 + 0.25i, 0.8 - 0.25i, 0.9];

% Objetivo practico: step ~20 y u limitado a +/-300
r_step = 20;
u_max  = 300;

% knobs
wy = 20;          % subir => mas seguimiento (mas agresivo)
wu = 500;         % subir => menos esfuerzo (mas timido)

% Q y R coherentes
Q = wy*(C'*C) + 1e-8*eye(n);
R = wu/(u_max);

%% ========================
% 3) GANANCIAS: LQR + OBSERVADOR + Nbar
%% ========================
rc = rank(ctrb(A,B));
if rc < n
error('El par (A,B) NO es controlable (rank=%d < n=%d).', rc, n);
end

% --- LQR discreto (con fallback) ---
if exist('dlqr','file') == 2
[K, P, e_cl] = dlqr(A, B, Q, R);
elseif exist('dare','file') == 2
[P,~,~] = dare(A,B,Q,R);
K = (R + B'*P*B)\(B'*P*A);
e_cl = eig(A - B*K);
else
[P, K, e_cl, info] = dlqr_iter_nolic(A,B,Q,R);
fprintf('DLQR sin toolbox: iters=%d, err=%.3e\n', info.iters, info.err);
end

fprintf('LQR: eig(A-BK) = \n'); disp(e_cl.');

% --- Observador (place) ---
Ke_pred = place(A', C', p_obs).';           % predictor: A - Ke*C
Ke_act  = place(A', (C*A)', p_obs).';       % "actual": A - Ke*C*A

% --- Nbar (SISO) ---
if size(C,1) ~= 1
error('Tu C no es SISO (tiene %d salidas). Elegi una fila de C.', size(C,1));
end
[~,~,Nbar] = refi(A, B, C, K);
fprintf('Nbar=%.6g\n', Nbar);

%% ========================
% 4) SIMULACION: SIN RUIDO vs CON RUIDO (double y single)
%% ========================
N         = 200;
ulim_sat = u_max;
ulim_inf = Inf;

% referencia
r = zeros(1,N);
r(2:end) = r_step;

% --------- RUIDO (config) ----------

rng(1);                 % repetible

noise.enable   = true;
noise.sigma_y  = 2;     % ruido de medicion
noise.sigma_u  = 2.0;   % jitter actuador
noise.q_u      = 1.0;   % cuantizacion u
```

```
    noise.sigma_w  = 0.0;   % ruido de proceso

    noise.vy = noise.sigma_y * randn(1,N);
    noise.vu = noise.sigma_u * randn(1,N);
    noise.wx = noise.sigma_w * randn(n,N);

    noise_off = noise;
    noise_off.enable = false;
    noise_off.vy = zeros(1,N);
    noise_off.vu = zeros(1,N);
    noise_off.wx = zeros(n,N);

    % --- IDEAL (sin sat, sin ruido) ---
    out_pred_ideal = sim_obs_loop(A,B,C,D,K,Nbar,Ke_pred,Ke_act,r,Ts,ulim_inf,false,false,noise_off);
    out_act_ideal  = sim_obs_loop(A,B,C,D,K,Nbar,Ke_pred,Ke_act,r,Ts,ulim_inf,true ,false,noise_off);

    % --- REAL (sat, sin ruido) ---
    out_pred_sat_clean = sim_obs_loop(A,B,C,D,K,Nbar,Ke_pred,Ke_act,r,Ts,ulim_sat,false,false,noise_off);
    out_act_sat_clean  = sim_obs_loop(A,B,C,D,K,Nbar,Ke_pred,Ke_act,r,Ts,ulim_sat,true ,false,noise_off);

    % --- REAL (sat, con ruido) ---
    out_pred_sat_noise = sim_obs_loop(A,B,C,D,K,Nbar,Ke_pred,Ke_act,r,Ts,ulim_sat,false,false,noise);
    out_act_sat_noise  = sim_obs_loop(A,B,C,D,K,Nbar,Ke_pred,Ke_act,r,Ts,ulim_sat,true ,false,noise);

    % --- REAL (sat, con ruido) en FLOAT32 ---
    out_pred_sat_noise_f = sim_obs_loop(A,B,C,D,K,Nbar,Ke_pred,Ke_act,r,Ts,ulim_sat,false,true,noise);
    out_act_sat_noise_f  = sim_obs_loop(A,B,C,D,K,Nbar,Ke_pred,Ke_act,r,Ts,ulim_sat,true ,true,noise);

    t = (0:N-1)*Ts;

    %% ========================
    % 5) DIAGNOSTICOS
    %% ========================
    eyp = out_pred_sat_noise.y_meas - out_pred_sat_noise.y_true;
    eya = out_act_sat_noise.y_meas  - out_act_sat_noise.y_true;

    fprintf('\n--- CHECK RUIDO ---\n');
    fprintf('sigma_y=%.3g | RMS(y_meas-y_true) Pred=%.3g Act=%.3g\n', noise.sigma_y, rms(eyp), rms(eya));
    fprintf('sigma_u=%.3g | q_u=%g\n', noise.sigma_u, noise.q_u);

    fprintf('\n--- SATURACION ---\n');
    fprintf('Pred clean: max|u|=%.2f sat=%d\n', max(abs(out_pred_sat_clean.u)), sum(abs(out_pred_sat_clean.u) >= ulim_sat-1e-9));
    fprintf('Pred noise: max|u|=%.2f sat=%d\n', max(abs(out_pred_sat_noise.u)), sum(abs(out_pred_sat_noise.u) >= ulim_sat-1e-9));
    fprintf('Act  clean: max|u|=%.2f sat=%d\n', max(abs(out_act_sat_clean.u)),  sum(abs(out_act_sat_clean.u)  >= ulim_sat-1e-9));
    fprintf('Act  noise: max|u|=%.2f sat=%d\n', max(abs(out_act_sat_noise.u)),  sum(abs(out_act_sat_noise.u)  >= ulim_sat-1e-9));

    %% ========================
    % 6) PLOTS
    %% ========================
    figure('Name','Predictor: y (clean vs noise) [sat]');
    plot(t, out_pred_sat_clean.y_true, 'LineWidth',1.6); hold on;
    plot(t, out_pred_sat_noise.y_meas, '.-');
    plot(t, r, 'k--','LineWidth',1.2);
    grid on; xlabel('t [s]'); ylabel('y');
    legend('y_true (clean)','y_meas (noise)','r','Location','best');

    figure('Name','Predictor: u (clean vs noise) [sat]');
    plot(t, out_pred_sat_clean.u, 'LineWidth',1.6); hold on;
    plot(t, out_pred_sat_noise.u, '.-');
    yline(+ulim_sat,'k--'); yline(-ulim_sat,'k--');
    grid on; xlabel('t [s]'); ylabel('u');

    figure('Name','Actual: y (clean vs noise) [sat]');
    plot(t, out_act_sat_clean.y_true, 'LineWidth',1.6); hold on;
    plot(t, out_act_sat_noise.y_meas, '.-');
    plot(t, r, 'k--','LineWidth',1.2);
    grid on; xlabel('t [s]'); ylabel('y');

    figure('Name','Actual: u (clean vs noise) [sat]');
    plot(t, out_act_sat_clean.u, 'LineWidth',1.6); hold on;
    plot(t, out_act_sat_noise.u, '.-');
    yline(+ulim_sat,'k--'); yline(-ulim_sat,'k--');
    grid on; xlabel('t [s]'); ylabel('u');

    figure('Name','Ruido de medicion (y_meas - y_true)');
    plot(t, eyp, t, eya);
    grid on; xlabel('t [s]'); ylabel('error de medicion');
    legend('Pred','Act','Location','best');

    figure('Name','Float32 - Double (y_meas) [sat+noise]');
```

```
    plot(t, out_pred_sat_noise.y_meas - out_pred_sat_noise_f.y_meas, ...
    t, out_act_sat_noise.y_meas  - out_act_sat_noise_f.y_meas);
    grid on; xlabel('t [s]'); ylabel('double - float32');
    legend('Pred','Act','Location','best');

    %% ========================
    % 7) Z-PLANE (polos planta, CL, obs)
    %% ========================
    p_ol = eig(A);
    p_cl = eig(A - B*K);
    p_op = eig(A - Ke_pred*C);
    p_oa = eig(A - Ke_act*C*A);

    try
    z_plant = tzero(ss(A,B,C,D,Ts));
    catch
    z_plant = [];
    end

    plot_zplane('Z-plane: Predictor', p_ol, p_cl, p_op, z_plant);
    plot_zplane('Z-plane: Actual'   , p_ol, p_cl, p_oa, z_plant);

    %% ==========================================================================

    % ==================== FUNCIONES LOCALES =====================

    function out = sim_obs_loop(A,B,C,D,K,Nbar,Ke_pred,Ke_act,r,Ts,ulim,use_actual,use_single,noise)
    if nargin < 15 || isempty(noise)
    noise.enable=false; noise.vy=0; noise.vu=0; noise.wx=0; noise.q_u=Inf;
    end

    if use_single
    A=single(A); B=single(B); C=single(C); D=single(D);
    K=single(K); Nbar=single(Nbar);
    Ke_pred=single(Ke_pred); Ke_act=single(Ke_act);
    r=single(r); Ts=single(Ts); ulim=single(ulim);
    end

    n = size(A,1); N = numel(r);

    x  = zeros(n,N,'like',A);
    xh = zeros(n,N,'like',A);

    y_true = zeros(1,N,'like',A);
    y_meas = zeros(1,N,'like',A);

    u_cmd  = zeros(1,N,'like',A);
    u_app  = zeros(1,N,'like',A);

    vy = zeros(1,N,'like',A);
    vu = zeros(1,N,'like',A);
    wx = zeros(n,N,'like',A);

    if isfield(noise,'enable') && noise.enable
    vy = cast(noise.vy,'like',A);
    vu = cast(noise.vu,'like',A);
    wx = cast(noise.wx,'like',A);
    end

    q_u = Inf;
    if isfield(noise,'q_u'), q_u = noise.q_u; end
    q_u = cast(q_u,'like',A);

    y_true(1) = C*x(:,1) + D*0;
    y_meas(1) = y_true(1) + vy(1);

    for k=1:N-1
    u_unsat  = Nbar*r(k) - K*xh(:,k);
    u_cmd(k) = sat(u_unsat, ulim);

    u_app(k) = u_cmd(k) + vu(k);
    if isfinite(double(q_u))
    u_app(k) = round(u_app(k)/q_u)*q_u;
    end
    u_app(k) = sat(u_app(k), ulim);

    x(:,k+1) = A*x(:,k) + B*u_app(k) + wx(:,k);

    y_true(k)   = C*x(:,k)   + D*u_app(k);
    y_true(k+1) = C*x(:,k+1) + D*u_app(k);

    y_meas(k)   = y_true(k)   + vy(k);
    y_meas(k+1) = y_true(k+1) + vy(k+1);

    if ~use_actual
    yhat_k = C*xh(:,k) + D*u_app(k);
    xh(:,k+1) = A*xh(:,k) + B*u_app(k) + Ke_pred*( y_meas(k) - yhat_k );
    else
    z   = A*xh(:,k) + B*u_app(k);
    yzh = C*z + D*u_app(k);
    xh(:,k+1) = z + Ke_act*( y_meas(k+1) - yzh );
    end
    end

    u_cmd(N) = sat(Nbar*r(N) - K*xh(:,N), ulim);
    u_app(N) = u_cmd(N) + vu(N);
    if isfinite(double(q_u))
    u_app(N) = round(u_app(N)/q_u)*q_u;
    end
    u_app(N) = sat(u_app(N), ulim);

    y_true(N) = C*x(:,N) + D*u_app(N);
```

```
    y_meas(N) = y_true(N) + vy(N);

    out.x = x; out.xh = xh;
    out.y_true = y_true;
    out.y_meas = y_meas;
    out.u_cmd  = u_cmd;
    out.u_app  = u_app;

    out.y = y_meas;
    out.u = u_app;
    end

    function y = sat(u,lim)
    if isinf(lim)
    y = u;
    else
    y = min(max(u, -lim), lim);
    end
    end

    function plot_zplane(figName, p_ol, p_cl, p_obs, z_plant)
    figure('Name',figName,'NumberTitle','off');
    hold on; grid on; grid minor; axis equal;
    title('Z-plane');
    xlabel('Re{z}'); ylabel('Im{z}');

    th = linspace(0,2*pi,400);
    plot(cos(th), sin(th), 'k:');  % unit circle

    plot(real(p_ol),  imag(p_ol),  'o', 'LineWidth', 1.5);
    plot(real(p_cl),  imag(p_cl),  'x', 'LineWidth', 1.8);
    plot(real(p_obs), imag(p_obs), '^', 'LineWidth', 1.8);

    if ~isempty(z_plant)
    plot(real(z_plant), imag(z_plant), 's', 'LineWidth', 1.5);
    legend('unit␣circle','poles␣plant␣(A)','poles␣CL␣(A-BK)','poles␣obs','zeros␣plant','Location','bestoutside');
    else
    legend('unit␣circle','poles␣plant␣(A)','poles␣CL␣(A-BK)','poles␣obs','Location','bestoutside');
    end

    xlim([-1.2 1.2]); ylim([-1.2 1.2]);
    end

    function [Nx,Nu,Nbar] = refi(phi,gam,Hr,K)
    I=eye(size(phi));
    [m,n]=size(Hr);
    np=inv([phi-I gam;Hr zeros(m)])*([zeros(n,m);eye(m)]);
    Nx=np(1:n,:);
    Nu=np(n+1:n+m,:);
    Nbar=Nu+K*Nx;
    end

    function [P, K, e_cl, info] = dlqr_iter_nolic(A,B,Q,R)
    maxit = 5000;
    tol   = 1e-10;

    P = Q;
    err = Inf;

    for it = 1:maxit
    G = R + B'*P*B;
    Ktmp = G \ (B'*P*A);
    Pn = A'*P*A - A'*P*B*Ktmp + Q;

    err = norm(Pn - P, 'fro');
    P = Pn;

    if err < tol
    break;
    end
    end

    K = (R + B'*P*B) \ (B'*P*A);
    e_cl = eig(A - B*K);

    info.iters = it;
    info.err   = err;
    info.converged = (err < tol);

    if ~info.converged
    warning('Riccati␣iterativa␣NO␣convergio␣(err=%.3e).', err);
    end
    end
```

Listing 2. LQR + Observador (predictor/actual) + Nbar + Simulacion con saturacion y ruido

En este apéndice se presenta el script utilizado para sintonizar la covarianza del ruido de proceso del filtro de Kalman manteniendo fija la covarianza de medición.

Se parametriza la matriz de proceso como $Q = qI_n$ y se fija $R = \sigma_v^2$, donde σ_v se obtuvo a partir de mediciones empíricas del sensor (planta en reposo). El escalar q se selecciona mediante un criterio de consistencia estadística sobre la innovación normalizada.

La innovación se define como $\nu_k = y_k - \hat{y}_{k|k-1}$ y su varianza teórica estacionaria como $S = CPC^T + R$, donde P es la solución estacionaria de Riccati. Se utiliza:

$$\eta_k = \frac{\nu_k}{\sqrt{S}}$$

y se escoge el valor q que aproxima $\mathrm{var}(\eta_k) \approx 1$.

A. *Script de sintonización (RQ_tuning_fixedR.m)*

A continuación se incluye el código completo empleado para el barrido de q en escala logarítmica, el cálculo de la ganancia estacionaria con `dlqe` y el guardado de los parámetros resultantes en `RQ_tuning_fixedR.mat`.

```matlab
%% ======================================
% Tuning de Q = q*I para Kalman (dlqe) con R FIJO (sigma_v ~ 2..3 cm)
% usando datos reales (dato1,dato2) desde iddata.
%
% - Planta: BJ continuo en planta (1).mat -> G = B/F
% - Datos : datos.mat con dato1,dato2 (iddata) con y en cm, u en du_us
% - Ts_target: 0.01 s (decimación exacta si Ts original es divisor)
%
% Objetivo: var(eta) ~ 1 donde eta = nu / sqrt(S), S = C P C' + R
% (criterio consistente para elegir q cuando el modelo no es perfecto)
%% ======================================
close all; clear; clc

%% ===== rutas =====
mat_planta = 'planta (1).mat';
mat_datos  = 'D:\GitHub\auto\TPF-HelicopteroVertical\Matlab\PruebasEmpiricas\datos.mat';

Ts_target = 0.01;                    % 100 Hz

%% ======================================
% 1) CARGAR PLANTA Y DISCRETIZAR
%% ======================================
S = load(mat_planta);

if isfield(S,'plantaC')
plantaC = S.plantaC;
elseif isfield(S,'sysC')
plantaC = S.sysC;
else
error('No encuentro "plantaC" ni "sysC" dentro de %s', mat_planta);
end

% Planta determinista desde BJ: G = B/F
G  = tf(plantaC.B, plantaC.F);
Gd = c2d(ss(G), Ts_target, 'zoh');
[A,B,C,D] = ssdata(Gd);

n = size(A,1);
if size(B,2) ~= 1
error('SISO requerido. size(B,2)=%d', size(B,2));
end
if size(C,1) ~= 1
C = C(1,:);
D = D(1,:);
end
if isempty(D), D = 0; end
D = double(D);

fprintf('Planta discretizada: n=%d, Ts_target=%.6g\n', n, Ts_target);

%% ======================================
% 2) CARGAR DATOS (iddata) Y ARMAR u,y (concatenados)
%% ======================================
DD = load(mat_datos);
use_names = {'dato1','dato2'};

u_all = [];
y_all = [];

for i=1:numel(use_names)
nm = use_names{i};
if ~isfield(DD,nm)
error('No existe %s en %s', nm, mat_datos);
end

zi = DD.(nm);
if ~isa(zi,'iddata')
error('%s no es iddata.', nm);
end

Ts_i = zi.Ts;
if isempty(Ts_i) || Ts_i <= 0
error('%s: iddata sin Ts válido.', nm);
end

% señales crudas
yraw = zi.OutputData(:);
uraw = zi.InputData(:);

% limpiar NaN/Inf (sin isfinite por compat)
m = ~isnan(yraw) & ~isinf(yraw) & ~isnan(uraw) & ~isinf(uraw);
yraw = yraw(m);
uraw = uraw(m);

% decimación exacta a Ts_target
if abs(Ts_i - Ts_target) > 1e-12
ratio = Ts_target / Ts_i;
if abs(ratio - round(ratio)) > 1e-12
error('%s: Ts_target/Ts_i no entero. Ts_i=%.17g Ts_target=%.17g', nm, Ts_i, Ts_target);
end
M = round(ratio);
fprintf('%s: decimación M=%d (Ts %.17g -> %.17g)\n', nm, M, Ts_i, Ts_target);
yraw = yraw(1:M:end);
uraw = uraw(1:M:end);
else
fprintf('%s: Ts ya coincide (%.17g)\n', nm, Ts_i);
end

% (opcional) detrend: sacá media para no pelear con offsets
yraw = yraw - mean(yraw);
uraw = uraw - mean(uraw);

fprintf('%s: N=%d\n', nm, numel(yraw));

% concatenar
y_all = [y_all; yraw(:)]; %#ok<AGROW>
u_all = [u_all; uraw(:)]; %#ok<AGROW>
end

y = y_all;
u = u_all;

```

```matlab
fprintf('TOTAL: N=%d muestras\n', numel(y));

%% =======================================
% 3) FIJAR R (ruido de medición) y TUNEAR q en Q=q*I
%% =======================================
sigma_v = 2.043227851;      % cm  (poné 2..3 según tu medición)
R = sigma_v^2;

q_grid = logspace(-12, 2, 120);

best_q = NaN; best_err = Inf;
stats_q = zeros(numel(q_grid),1);
stats_var_eta = zeros(numel(q_grid),1);
stats_var_nu  = zeros(numel(q_grid),1);

for i=1:numel(q_grid)
q = q_grid(i);
Q = q*eye(n);

% Kalman (w entra a estados)
Gk = eye(n);
[L,P,~] = dlqe(A,Gk,C,Q,R);

% Sinnov estacionario
Sinnov = C*P*C' + R;
if Sinnov < 1e-12, Sinnov = 1e-12; end

xhat = zeros(n,1);
nu   = zeros(numel(y),1);

for k=1:numel(y)
ypred = C*xhat + D*u(k);
nu(k) = y(k) - ypred;

% delayed estimator estilo firmware
xhat = xhat + L*nu(k);
xhat = A*xhat + B*u(k);
end

eta = nu / sqrt(Sinnov);

var_eta = var(eta,1);
var_nu  = var(nu,1);

stats_q(i) = q;
stats_var_eta(i) = var_eta;
stats_var_nu(i)  = var_nu;

err = abs(log(var_eta));    % objetivo var_eta ~ 1
if err < best_err
best_err = err;
best_q = q;
best_L = L;
best_P = P;
best_S = Sinnov;
end
end

fprintf('\n=== Resultado tuning Q con R fijo ===\n');
fprintf('sigma_v = %.3f cm => R= %.3f cm^2\n', sigma_v, R);
fprintf('q_best  = %.3e\n', best_q);
fprintf('Sinnov  = %.3f  => sigma_pred_innov = %.3f cm\n', best_S, sqrt(best_S));
fprintf('L_best  =\n'); disp(best_L);

%% =======================================
% 4) PLOTS
%% =======================================
figure('Name','Tuning Q: var(eta)');
semilogx(stats_q, stats_var_eta,'LineWidth',1.2); grid on; grid minor;
xlabel('q'); ylabel('var(\eta)'); title('Objetivo: var(\eta) \approx 1');

figure('Name','Tuning Q: sigma(nu)');
semilogx(stats_q, sqrt(stats_var_nu),'LineWidth',1.2); grid on; grid minor;
xlabel('q'); ylabel('sigma(\nu) [cm]'); title('Innovación (predicción) vs q');

%% =======================================
% 5) Guardar
%%
Q_best = best_q*eye(n);
L_best = best_L;
save('RQ_tuning_fixedR.mat','sigma_v','R','best_q','Q_best','L_best','best_P','best_S','A','B','C','D','Ts_target');
disp('Guardado: RQ_tuning_fixedR.mat');
```

B. *Parámetros exportados para el diseño LQG*

El script guarda el archivo `RQ_tuning_fixedR.mat` con los parámetros utilizados posteriormente en el diseño del estimador y controlador:

- `sigma_v`: desviación estándar empírica del ruido de medición.
- `R`: covarianza del ruido de medición ($R = \sigma_v^2$).
- `best_q`: escalar óptimo de la parametrización del ruido de proceso.
- `Q_best`: matriz final de ruido de proceso ($Q = qI_n$).
- `L_best`: ganancia estacionaria calculada con `dlqe`.
- `best_P` y `best_S`: covarianza estacionaria y varianza de innovación.
- `A,B,C,D,Ts_target`: modelo discretizado usado en la sintonización.

En este apéndice se incluye el script utilizado para obtener los parámetros numéricos del esquema LQG/LQI empleado en el trabajo: ganancia del filtro de Kalman en régimen permanente L, ganancias de realimentación de estados K_x y del integrador K_i, junto con la verificación de polos del observador $\lambda(A-LC)$ y la simulación en presencia de ruido de proceso y medición.

El script fue construido con el objetivo de ser *firmware friendly*, es decir, replicar la lógica de estimación y control que luego se implementa en el microcontrolador:

- **Estimador tipo *current***: se calcula L con `kalman(...,'current')`.
- **Ruido de proceso y medición**: se inyecta w_k en la dinámica del estado y v_k en la medición.
- **Control con integrador**: se emplea una acción integral ξ_k para asegurar error estacionario nulo.

C. Estructura del diseño

C0a. Planta discretizada: A partir del modelo continuo identificado (`planta (1).mat`) se discretiza con ZOH a $T_s = 0.01$ s (100 Hz), obteniendo el sistema discreto:

$$x_{k+1} = Ax_k + Bu_k, \qquad y_k = Cx_k + Du_k$$

C0b. Covarianzas de ruido: Las covarianzas utilizadas por el estimador se toman del archivo `RQ_tuning_fixedR.mat` (Apéndice A):

$$Q_n = Q_{\text{best}}, \qquad R_n = R$$

Adicionalmente, para la simulación se utilizan las desviaciones estándar: σ_v (medición) y σ_w (proceso), e inyección de ruido blanco gaussiano.

C0c. Filtro de Kalman (current estimator): Se construye un sistema extendido para el cálculo de Kalman, incorporando explícitamente el canal de ruido de proceso w:

$$x_{k+1} = Ax_k + Bu_k + Gw_k, \qquad y_k = Cx_k + Du_k + Hw_k + v_k$$

En este trabajo se asumió:

$$G = I_n, \qquad H = 0$$

y se obtiene la ganancia L en régimen permanente (y sus polos):

$$L = L_k, \qquad \lambda(A - LCA)$$

C0d. Control LQI con integrador: Se incorpora un integrador escalar ξ_k sobre el error:

$$\xi_{k+1} = \xi_k + (r_k - y_k)$$

y se adopta la ley de control:

$$u_k = -K_x \hat{x}_k + K_i \xi_k$$

Las ganancias se obtienen mediante `dlqr` aplicado al sistema aumentado, usando pesos heurísticos basados en escalas prácticas: paso deseado $\Delta y \approx 20$ y esfuerzo máximo $|u| \lesssim 300$.

D. Script completo (LQG_servo_firmware_friendly.m)

A continuación se incluye el script completo utilizado para calcular L, K_x, K_i, reportar polos y simular en presencia de ruido.

```
        %% =========================
        %  LQG SERVO "firmware friendly"
        %  (kalman + lqi + lqgtrack)
        %  + estimador CURRENT (default en discreto)
        %  + ruido de proceso y medición (sigma_w, sigma_v)
        %% =========================
        close all; clear; clc

        %% =========================
        % 1) CARGA + DISCRETIZACIÓN
        %% =========================
        S = load('planta (1).mat');

        if isfield(S,'plantaC')
        plantaC = S.plantaC;
        elseif isfield(S,'sysC')
        plantaC = S.sysC;
        else
        error('No encuentro "plantaC" ni "sysC" dentro de planta (1).mat');
        end

        Ts   = 1/100;                  % 100 Hz
        sysD = c2d(plantaC, Ts, 'zoh');
        [A,B,C,D] = ssdata(ss(sysD));

        n = size(A,1);
        if size(C,1) ~= 1
        C = C(1,:);
        D = D(1,:);
        end
        if size(B,2) ~= 1
        error('Este script asume SISO (1 entrada). size(B,2)=%d', size(B,2));
        end

        fprintf('Ts=%.9f | n=%d\n', Ts, n);

        %% =========================
        % 2) RUIDOS (sigma -> covarianzas)
        %% =========================
        S = load("RQ_tuning_fixedR.mat");

        Qn = S.Q_best;      % cov(w) : nxn
        Rn = S.R;           % cov(v) : 1x1
        sigma_v = S.sigma_v;
        sigma_w = S.best_q;
        Nn = zeros(n,1);    % cov(wv') asumimos 0

        %% =========================
        % 3) KALMAN: sys debe incluir el canal de ruido w
        %% =========================
        G = eye(n);          % w entra a todos los estados
        H = zeros(1,n);     % y NO depende directamente de w (H=0)

        sysK = ss(A, [B G], C, [D H], Ts);  % inputs: [u ; w]

        % Para tu MATLAB: TYPE va como ÚLTIMO argumento.
        % 'current' es el default, pero lo dejamos explícito.
        [kest, Lk, Pk, Mx, Z, My] = kalman(sysK, Qn, Rn, Nn, 'current');

        fprintf('\nLk (kalman) = '); disp(Lk);
        fprintf('poles(A-LC) = '); disp(eig(A - Lk*C).');
```

```matlab
    %% ========================
    % 4) LQI: K = [Kx Ki] para u = -Kx*xhat - Ki*xi
    %% ========================
    m = 1;                      % integrador escalar
    Ahat = [A B; zeros(m,n+m)];
    Bhat = [zeros(n,m); eye(m)];
    Chat = [C zeros(1,m)];

    % --- interpretación física: "step ~20" y "u no pase ~300" ---
    y_step = 20;
    u_max  = 300;

    % knobs
    wy = 20;                    % subí => seguís más (más agresivo)
    wu = 50;                    % subí => penalizás más u (más tímido)
    wv = 0.00000000000001;      % subí => penalizás integrador

    % OJO: acá tu “escala física” es discutible. Esto es heurístico.
    qy = wy*(1/(y_step));
    ru = wu*(1/(u_max));
    qv = wv*(1/(y_step));

    Qa = blkdiag(qy*(C'*C) + 1e-8*eye(n), qv);
    R  = ru;

    [Khat,Pa,ecl] = dlqr(Ahat, Bhat, Qa, R);

    Aux  = [A-eye(size(A))  B;
    C*A              C*B];

    K2K1 = (Khat + [zeros(1,n) eye(m)]) / Aux;

    Kx = K2K1(1,1:n);
    Ki = K2K1(1,n+1);

    fprintf('\nKx (lqi) = '); disp(Kx);
    fprintf('Ki (lqi) = '); disp(Ki);

    %% ========================
    % 6) SIMULACIÓN (tu estilo) con ruido de proceso+medición
    %% ========================
    N  = 1200;
    t  = 0:Ts:(N-1)*Ts;

    r = ones(1,N)*25;
    r(1:30) = 0;

    umin = -u_max; umax = u_max;

    X  = zeros(n,N);    % estados reales
    Xh = zeros(n,N);    % estados estimados (los que usás en tu control)
    V  = zeros(1,N);    % integrador
    U  = zeros(1,N);    % control

    Y_true = zeros(1,N);
    Y_meas = zeros(1,N);

    for kidx = 1:N-1
    rk = r(kidx);

    % medición y[k]
    y_true_k = C*X(:,kidx);
    v_k      = sigma_v*randn;
    y_meas_k = y_true_k + v_k;

    Y_true(kidx) = y_true_k;
    Y_meas(kidx) = y_meas_k;

    % integrador xi[k+1] = xi[k] + (r-y)
    V(kidx+1) = V(kidx) + (rk - y_meas_k);

    % control: u = -Kx*xhat - Ki*xi
    u_unsat = -Kx*Xh(:,kidx) + Ki*V(kidx+1);
    u_k     = min(max(u_unsat, umin), umax);
    U(kidx) = u_k;

    % planta con ruido de proceso: x[k+1] = A x + B u + w
    w_k = sigma_w*randn(n,1);
    X(:,kidx+1) = A*X(:,kidx) + B*u_k + w_k;

    % estimador (usa Lk)
    Xh(:,kidx+1) = A*Xh(:,kidx) + B*u_k + ...
    Lk*(y_meas_k - (C*Xh(:,kidx) + D*u_k));
    end

    % último sample para plots
    Y_true(N) = C*X(:,N);
    Y_meas(N) = Y_true(N) + sigma_v*randn;
    U(N)      = U(N-1);

    %% ========================
    % 7) PLOTS
    %% ========================
    figure('Name','y(t) -- LQG + integrador + ruido');
    plot(t, Y_true, 'LineWidth',1.2); hold on;
    plot(t, Y_meas, '.', 'LineWidth',1.0);
    plot(t, r, 'k--','LineWidth',1.1);
    grid on; grid minor;
    xlabel('t [s]'); ylabel('y');
    legend('y_true','y_meas','r','Location','best');

    figure('Name','u(t) -- saturado');
    plot(t, U, 'LineWidth',1.4);
    grid on; grid minor;
    xlabel('t [s]'); ylabel('u');
    ylim([umin umax]);

    figure('Name','xi(t) -- integrador');
    plot(t, V, 'LineWidth',1.4);
```

```
        grid on; grid minor;
        xlabel('t␣[s]'); ylabel('\xi');

        Aaug = [A,B;...
        Kx-Kx*A-Ki*C*A,1-Kx*B-Ki*C*B];
        Baug = [0;0;0;Ki];
        Caug = [C,0];
        D = 0;
        sysDaug = ss(Aaug,Baug,Caug,D,Ts);

        figure; pzmap(sysDaug); grid on; zgrid;
```

Durante el desarrollo del trabajo práctico, la estructura física de la planta atravesó distintas etapas de diseño, las cuales permitieron identificar limitaciones mecánicas y realizar mejoras progresivas hasta alcanzar la configuración final utilizada en las prácticas experimentales. En este apéndice se describe la primera etapa de diseño de la estructura y se destacan las principales diferencias respecto de la versión final.

E. Primera etapa de diseño

La primera versión de la estructura fue concebida con una altura total aproximada de 80 cm, utilizando la misma base y el mismo techo de madera que se mantienen en el diseño final. Debido a las dimensiones de estos elementos, la altura útil de movimiento del cuerpo móvil en esta etapa era de aproximadamente 72 cm.

En esta configuración inicial, el diseño mecánico del cuerpo móvil era diferente al actual, presentando dimensiones ligeramente mayores. El sistema no contaba con elementos de seguridad adicionales, tales como topes mecánicos, amortiguación ante caídas ni cuerda de seguridad, dado que el recorrido vertical era considerablemente menor y el riesgo asociado a caídas desde grandes alturas resultaba limitado.

El guiado del cuerpo móvil se realizaba mediante vigas metálicas rectas y rígidas, las cuales no presentaban deformaciones apreciables. Debido a esta rigidez estructural, no fue necesario incorporar articulaciones pasivas tipo “muñeca” en las abrazaderas, ni estructuras auxiliares de madera para limitar deformaciones. En esta etapa, el contacto entre el cuerpo móvil y los rieles generaba fricción apreciable, la cual se manifestaba de forma consistente durante el movimiento vertical.

Cabe destacar que este comportamiento friccional, observable en la primera versión de la estructura, no se presenta de la misma manera en el diseño final. La incorporación de vigas metálicas de mayor longitud, junto con las deformaciones inherentes a las mismas y la inclusión de articulaciones pasivas en las abrazaderas, redujo significativamente la fricción directa entre el cuerpo móvil y los rieles, modificando así las características mecánicas del sistema.

En las figuras siguientes se presentan imágenes correspondientes a las primeras versiones de las piezas impresas en 3D utilizadas en esta etapa inicial del diseño, las cuales difieren de las empleadas en la configuración final de la planta.

En la primera etapa de diseño, el soporte superior del cuerpo móvil presentaba una altura aproximada de 3 cm y un diámetro de 4 cm. Las secciones sobresalientes destinadas al acople de

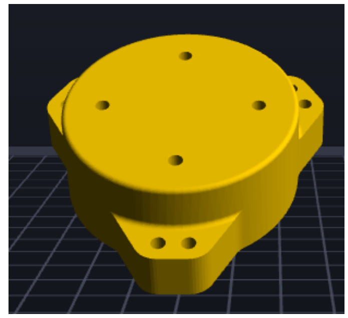


Figura 35. Primer diseño del soporte superior del motor.

los brazos contaban con una altura de aproximadamente $1{,}5$ cm. Dicho soporte incluía orificios dimensionados específicamente para el montaje del motor brushless, con un diámetro de 3 mm, acorde al patrón de fijación del mismo.

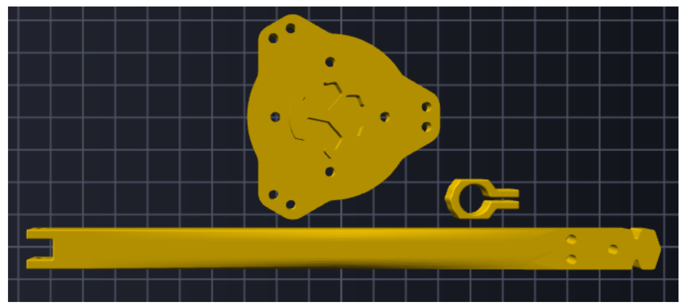


Figura 36. Primeros diseños del soporte inferior y de los brazos estructurales.

El soporte inferior del cuerpo móvil correspondía a una geometría espejo del soporte superior. En este componente se realizaba el encastre de los brazos estructurales, los cuales, en esta etapa inicial, presentaban dimensiones menores en comparación con el diseño final. Cada brazo tenía dimensiones aproximadas de $1\,\text{cm} \times 1\,\text{cm} \times 20\,\text{cm}$.

El sistema de agarre de los brazos difería del implementado en la versión final de la planta. Inicialmente, el agarre no contaba con movilidad angular, aunque permitía un ajuste manual respecto a la posición de la viga metálica, lo que condicionaba el guiado del cuerpo móvil y su interacción con los rieles.

F. Segunda etapa de diseño

En la segunda etapa de diseño, el componente que presentó mayores modificaciones fue el brazo estructural. A partir de la experiencia obtenida en la etapa inicial, se introdujeron variaciones geométricas en el diseño del brazo, incorporando curvaturas con el objetivo de mejorar el encastre y la interacción con la estructura de guiado.

En esta versión, el brazo y el sistema de agarre fueron integrados en una única pieza impresa en 3D, eliminando la separación entre ambos componentes. Dado que en esta etapa las vigas metálicas utilizadas como rieles presentaban una geometría recta y una rigidez suficiente, no fue necesaria la incorporación de articulaciones pasivas tipo “muñeca”. En consecuencia, el guiado del cuerpo móvil se realizaba mediante un agarre rígido, sin movilidad angular.

Cabe destacar que, durante esta etapa, el diseño del cuerpo móvil se mantuvo sin modificaciones significativas respecto a la versión anterior. Las mejoras se concentraron exclusivamente en el diseño de los brazos y del sistema de agarre, manteniendo constante la geometría general del conjunto móvil.

Figura 37. Segundo diseño de brazo.

G. *Tercera etapa de diseño*

En la tercera etapa de diseño se introdujo una modificación significativa en la estructura general de la planta, extendiendo su altura máxima hasta aproximadamente 160 cm. Esta ampliación respondió a la necesidad de disponer de un mayor recorrido vertical para la realización de las prácticas de control, lo cual implicó nuevas exigencias mecánicas sobre el conjunto estructural y el cuerpo móvil.

Como consecuencia del aumento de altura de la estructura, el diseño de los brazos del cuerpo móvil volvió a ser modificado. En esta etapa, los brazos fueron rediseñados con mayor grosor y mayor altura, con el objetivo de incrementar su rigidez y capacidad de carga. Asimismo, se incorporaron aberturas longitudinales en los brazos, destinadas a permitir la inserción de elementos metálicos, con el fin de reforzar la estructura y mejorar su resistencia mecánica frente a esfuerzos y vibraciones.

El cuerpo móvil mantuvo su configuración general respecto a las etapas anteriores; sin embargo, el rediseño de los brazos resultó fundamental para adaptar el conjunto a las nuevas condiciones estructurales impuestas por la mayor altura de la planta.

Figura 38. Tercer diseño de brazos.

H. *Cuarta etapa de diseño: configuración final*

La cuarta etapa de diseño corresponde a la configuración final de la estructura y del cuerpo móvil utilizada en las prácticas experimentales del trabajo. En esta etapa se introdujeron modificaciones orientadas principalmente a mejorar el guiado mecánico del cuerpo móvil y a reducir la masa total del conjunto.

Debido a que las vigas metálicas empleadas como rieles presentan deformaciones asociadas a su longitud, se incorporaron articulaciones pasivas tipo "muñeca" en el sistema de guiado. Estas articulaciones permiten un movimiento angular relativo entre el cuerpo móvil y los rieles, mejorando el desplazamiento vertical y evitando atascamientos o esfuerzos indeseados durante el recorrido.

Con el objetivo de reducir la masa del cuerpo móvil, se redimensionaron los soportes principales. El soporte superior, que en versiones anteriores presentaba una altura de 3 cm, fue reducido a aproximadamente $1{,}5$ cm, mientras que el soporte inferior pasó de 2 cm a $0{,}5$ cm. A pesar de esta reducción dimensional, se conservó el sistema de encastre tanto en el soporte superior como en el inferior, asegurando la rigidez estructural del conjunto.

Adicionalmente, se incorporó un soporte específico para la batería, integrado al cuerpo móvil. En dicho soporte se colocaron almohadillas internas con el fin de proteger la batería frente a vibraciones e impactos durante el funcionamiento del sistema.

Estas modificaciones permitieron obtener un diseño final más liviano, adaptable a las deformaciones estructurales de los rieles y adecuado para la implementación de las distintas estrategias de control desarrolladas en el presente trabajo.

Durante el desarrollo experimental del trabajo se presentaron fallas en los controladores electrónicos de velocidad (ESC) utilizados en las primeras etapas de prueba del sistema. En este apéndice se describen los incidentes observados, junto con el análisis de las posibles causas y las medidas adoptadas posteriormente.

I. *Primer incidente: ESC de 30 A con alimentación externa*

En una primera instancia, se utilizó un ESC de 30 A alimentado mediante una batería para automóviles, con el objetivo de verificar el funcionamiento básico del sistema de propulsión. La conexión entre la fuente de alimentación y el ESC se realizó utilizando un cable unifilar de cobre de considerable longitud.

Figura 39. Tercer diseño de brazos.

Durante las pruebas iniciales, el sistema logró generar empuje y el cuerpo móvil llegó a elevarse. Sin embargo, tras un período de funcionamiento, el ESC comenzó a emitir una secuencia de señales acústicas consistente en cuatro pitidos cortos seguidos de un pitido largo. En ese momento no se contaba con una interpretación clara del significado de dicha señalización.

Posteriormente, mediante la consulta de documentación y experiencias previas, se determinó que dicha secuencia de pitidos está asociada a condiciones de protección del ESC, tales como sobrecorriente o sobretemperatura. Esta hipótesis se vio reforzada por el hecho de que los cables unifilares utilizados para la alimentación se calentaron excesivamente y llegaron a derretirse, indicando una circulación de corriente elevada y pérdidas resistivas significativas.

J. Segundo incidente: reinicios y falla del ESC de 30 A

En una segunda etapa de pruebas con el mismo ESC de 30 A, se reemplazaron los cables de alimentación por conductores adecuados para altas corrientes, conectando el ESC directamente a la batería utilizada en la planta. En esta configuración, el sistema no lograba elevarse de forma sostenida y el ESC emitía una secuencia de sonidos correspondiente a un reinicio del controlador.

Con el fin de descartar un problema en la señal de control, se analizó la señal PWM generada por el PSoC mediante un osciloscopio, verificándose que la misma presentaba una forma adecuada y estable, sin perturbaciones significativas. En consecuencia, se descartó que la falla estuviera asociada a errores en la generación de la señal de control.

Ante la hipótesis de una posible caída de tensión en la alimentación del ESC durante los transitorios de corriente, se incorporaron capacitores de desacople en la línea de alimentación. Tras esta modificación, el sistema logró generar empuje y elevarse durante breves instantes. No obstante, luego de un corto período de funcionamiento, se produjo la falla definitiva del ESC, observándose la quema de un MOSFET correspondiente a una de las fases del motor.

K. Análisis y consideraciones

A partir de los incidentes descritos, se identificaron como causas probables la combinación de sobrecorriente, exigencias térmicas elevadas y condiciones de alimentación no ideales durante las primeras pruebas. La utilización de una fuente de alimentación inadecuada, conductores con alta resistencia y la ausencia inicial de medidas de protección contribuyeron a someter al ESC a esfuerzos superiores a sus límites operativos.

Estos eventos pusieron de manifiesto la importancia de considerar cuidadosamente los aspectos de potencia, disipación térmica y protección eléctrica en sistemas de propulsión basados en motores brushless, incluso en etapas preliminares de prueba.

Las lecciones aprendidas a partir de estas fallas motivaron la adopción de controladores de mayor capacidad de corriente, mejoras en el cableado de alimentación y la implementación de estrategias de operación más conservadoras, las cuales permitieron continuar con el desarrollo experimental del trabajo de manera segura y confiable.

L. Tercer incidente: falla del ESC de 30 A durante operación con batería LiPo

En un tercer incidente, se utilizó un ESC de 30 A alimentado mediante una batería LiPo para drones. Durante esta prueba, se incrementó la señal PWM hasta aproximadamente $1500\,\mu s$, logrando que el sistema generara empuje suficiente para elevar el cuerpo móvil hasta la parte superior de la estructura.

Al intentar detener el movimiento, se adoptó un procedimiento no óptimo, consistente en bloquear mecánicamente la hélice con el fin de evitar una colisión con el techo de la estructura. Esta acción provocó el trabado de la hélice durante el funcionamiento del motor, lo cual generó un incremento abrupto de la corriente demandada. Como consecuencia, el ESC sufrió una falla catastrófica, produciéndose la quema de múltiples componentes internos y la pérdida total del controlador.

Este incidente permitió identificar el riesgo asociado al bloqueo mecánico del rotor en sistemas de propulsión brushless, dado que dicha condición conduce a corrientes elevadas que superan rápidamente la capacidad de los dispositivos de conmutación del ESC.

M. Cuarto incidente: falla del ESC de 30 A por sobrecorriente

En un cuarto incidente, se realizaron pruebas controladas con un nuevo ESC de 30 A, con el objetivo de determinar el valor máximo de PWM que el sistema podía soportar de manera segura. Durante esta prueba, el valor de PWM se incrementó progresivamente hasta alcanzar aproximadamente $1600\,\mu s$.

En estas condiciones, el ESC volvió a presentar una falla similar a la observada en el segundo incidente, registrándose la quema de un MOSFET correspondiente a una de las fases del motor. Este comportamiento reforzó la hipótesis de que el controlador se encontraba operando cerca de sus límites de corriente, incluso sin que se produjera un bloqueo mecánico del rotor.

N. *Medidas adoptadas*

La repetición de fallas en controladores de 30 A, tanto bajo condiciones transitorias como en operación sostenida, llevó a concluir que dicho margen de corriente resultaba insuficiente para el motor utilizado y las exigencias mecánicas de la planta. Asimismo, se consideró la posible influencia de algoritmos internos del ESC y de su calidad de construcción, los cuales podrían limitar su capacidad de manejo de sobrecorrientes.

En función de estas observaciones, se decidió sobredimensionar el sistema de actuación mediante la adquisición de un ESC de 40 A. Esta decisión permitió operar el motor con un mayor margen de seguridad, evitando la necesidad de reducir aún más la masa del cuerpo móvil y mejorando la confiabilidad del sistema durante las prácticas experimentales.

REFERENCIAS

[1] Dongguan E-S Motor Co., Ltd., "E-s motor official website," https://cdn.robotshop.com/rbm/a00a7635-653b-4220-aac9-b0c23c5c5e2c/5/520795f9-301f-4e66-a0a3-019811d1f78b/222c434c_a2212-brushless-motor.pdf, 2026, accessed: Feb. 2026.

[2] NEX Robotics Pvt, Ltd, "40a bldc esc," https://www.scribd.com/document/280924512/ESC-Datasheet, 2015, accessed: Feb. 2026.

[3] AMAZON, "Lipo 3s ovonic air," https://www.amazon.com/-/es/OVONIC-3s-Conector-Helic%C3%B3ptero-Quadcopter/dp/B07MS8QF3K, 1996-2026, accessed: Feb. 2026.

[4] Benewake Co., Ltd., *TFmini Plus Single-Point LiDAR Range Sensor Datasheet*, 2019, rev. A02. [Online]. Available: https://cdn.sparkfun.com/assets/2/b/0/3/8/TFmini_Plus-01-A02-Datasheet_EN.pdf

[5] Lic. Piero Codas Morselli, *Evaluación de Algoritmos de Control Digital Industrial*, 2005. [Online]. Available: https://drive.google.com/file/d/1TIa3A4172OXRvjMa7e-VbKXvCpIqimyW/view