



Universidad Nacional de Río Negro

INGENIERÍA ELECTRÓNICA

**Diseño, Implementación y Control de Posición Angular
mediante Rueda de Reacción de un Giróscopo**

Proyecto Final Integrador

Alumno: Franco Dal Farra

Director: Ing. Mg. Marcelo Javier Castelao Caruana

Codirector: Ing. Juan Pablo Pierini

San Carlos de Bariloche

2025

Índice

1. Resumen	3
2. Introducción	4
2.1. Motivación	5
2.2. Objetivos	5
2.3. Trabajo desarrollado	6
3. Descripción del sistema	7
3.1. Esquema experimental	7
3.2. Descripción del Hardware	10
3.2.1. Tarjeta de desarrollo	10
3.2.2. Unidad de medición inercial	11
3.2.3. Encoder magnético	12
3.2.4. Motor eléctrico	13
3.2.5. Driver puente H	14
3.2.6. Medición de corriente	15
3.3. Descripción del Software	17
3.3.1. Firmware ESP8266	18
3.3.2. Programa Python	20
3.3.3. Caracterización del tiempo de muestreo	21
4. Identificación y validación	23
4.1. Modelado matemático	23
4.2. Identificación del actuador	27
4.3. Identificación de planta	30
4.3.1. Velocidad angular del cubo	30
4.3.2. Corriente	33
4.3.3. Velocidad angular del disco	35
4.4. Validación del modelo	36
4.5. Identificación del sistema por espacio de estados	42
5. Diseño del control e implementación	45
5.1. Control mediante lazo simple	45
5.1.1. Controlador de adelanto de fase	45
5.1.2. Controlador proporcional integrativo derivativo	49
5.2. Control mediante doble lazo	52
5.2.1. Lazo interno de velocidad angular del disco	52
5.2.2. Lazo interno de corriente	55
5.3. Control moderno	59
6. Resultados obtenidos	65
6.1. Controlador de adelanto de fase	65
6.2. Controlador proporcional integrativo derivativo	67
6.3. Controlador de lazo interno de velocidad angular del disco	69
6.4. Controlador de lazo interno de corriente	71

6.5. Controlador moderno	73
7. Conclusiones	75
8. Anexos	78
8.1. Esquema de conexiones	78
8.2. Calibración de sensores y análisis del error	79
8.3. Código	86
8.4. Manual de usuario	94
Referencias	97

1. Resumen

El siguiente informe trata sobre el diseño y la implementación de un simulador de control de posición angular mediante ruedas de reacción. El trabajo fue realizado en el marco del Proyecto Final Integrador para la carrera de Ingeniería Electrónica en la Universidad Nacional de Río Negro, el cual a su vez es una continuación del trabajo realizado en una Práctica Profesional Supervisada en conjunto con el Centro Atómico Bariloche (CAB).

La motivación del trabajo tiene relación con el control de actitud de satélites mediante ruedas de reacción, sistema utilizado para mantener fija una orientación con respecto a otro elemento o marco de referencia fuera del marco de referencia del satélite. El esquema experimental intenta representar la dinámica del problema de control de actitud utilizando una rueda de reacción como actuador. La finalidad es lograr un dispositivo que pueda utilizarse académicamente para diseñar e implementar distintos algoritmos de control en la práctica.

La primera etapa del trabajo consistió en el diseño de la maqueta experimental en conjunto con los sensores, actuadores y módulos que conforman el sistema completo, tales como controladores, etapa de alimentación, etapa de comunicación, etcétera. Se logró el ensamblado físico del sistema con validación de funcionamiento mediante pruebas unitarias en cada parte.

La segunda etapa consistió en la caracterización e identificación de los distintos parámetros de la planta, para la realización de un modelo matemático que la represente y validación del mismo. Se utilizaron diferentes técnicas de identificación experimental y análisis de datos, buscando una correlación entre los resultados esperados y las mediciones obtenidas.

La última etapa consistió en el diseño e implementación de distintos esquemas de control, con el fin de estudiar las restricciones de la planta y proponer un control robusto y estable que satisfaga los requerimientos dados. A partir de las mediciones y los resultados obtenidos se reflexionó y se elaboraron conclusiones.

Se alcanzó un prototipo funcional en el cual se implementaron cinco algoritmos de control, diseñados a partir del modelo identificado, obteniendo resultados menores a 5 segundos en tiempo de establecimiento y 5% de error de estado estacionario.

2. Introducción

El control de actitud es un término que refiere a las acciones tomadas para mantener un control de la posición angular relativa de un cuerpo con respecto a otro cuerpo o un marco referencial externo. Dicha acción es de suma importancia para el manejo de aeronaves y particularmente para el control de naves no tripuladas, como pueden ser satélites o drones. Además, el control de actitud cobra mayor relevancia a partir de la necesidad de orientar al dispositivo hacia un punto determinado del espacio para poder establecer comunicación con otros satélites, con una antena de comunicación o para lograr mayor eficiencia en la carga de baterías mediante paneles solares.

Para poder realizar el control de actitud es necesario tomar mediciones sobre la posición angular relativa al objeto o al marco referencial con el que se lo desea comparar. Por otro lado, es necesario realizar acciones de posicionamiento acordes a la medición tomada y a la posición esperada. Existen diversas maneras de realizar la medición del movimiento angular como los son las unidades de medición inercial, giroscopios, acelerómetros, sensores de horizonte, sensores de campo magnético, rastreadores de estrellas, entre otros. A su vez, existen diversos mecanismos que se utilizan para realizar el posicionamiento, como propulsores, ruedas de reacción, imanes de torsión magnética, entre otros [9].

En este proyecto se trabajó en un prototipo experimental que detecta la posición angular relativa mediante acelerómetro y giróscopo, y utiliza una rueda de reacción como actuador. En la figura 1 se puede ver un esquema de sistema de control de posición con rueda de reacción [5]:

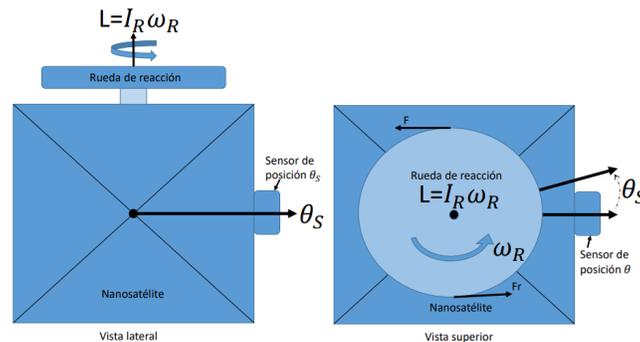


Figura 1: Esquema de sistema con rueda de reacción [5]

Se decidió orientar el trabajo a la medición y control en un único eje de rotación, siendo elegido el ángulo de guiñada entre los tres ángulos del sistema de ángulos de navegación o Tait-Bryan, como los vistos en la figura 2.

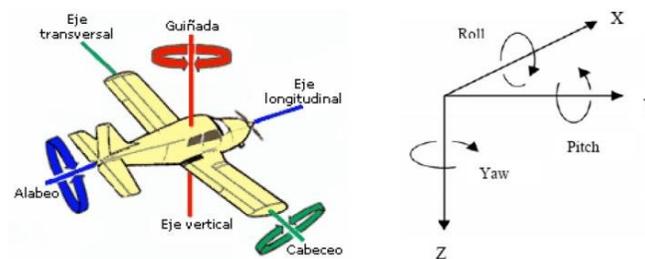


Figura 2: Ángulos de Tait-Bryan [1]

2.1. Motivación

Los sistemas de rueda de reacción funcionan con el principio de la conservación de momento angular, y presentan varias ventajas para ser utilizados en nanosatélites. Generan un torque elevado relativo a su tamaño reducido y su poco peso, y permiten lograr una elevada precisión de orientación. Además, consumen poco combustible comparados con los sistemas basados en propulsores [4] [19]. Se pueden encontrar en gran cantidad de nanosatélites en órbita, y algunos ejemplos comerciales son las ruedas de reacción cyber2 de Wittstein [7] o las ruedas de reacción Sinclair Interplanetary de Rocket Lab [23] (figura 3).



Figura 3: Rueda de reacción comercial [23]

Un paso importante para la puesta en marcha de los sistemas de control de actitud es su prueba y simulación en condiciones de laboratorio, ya que el coste de realizar pruebas con un prototipo en órbita son muy elevados. Debido a esto es que se fabrican simuladores que recrean las condiciones del sistema para comprobar el funcionamiento de las partes críticas del mismo. Algunos ejemplos de trabajos similares son los realizados por A. N. Castellon-López[5], F. W. Long[19], J. Gerber[13], C. Grassin[6], o M. A. Samuels[24].

Además, este proyecto tiene la motivación de generar un dispositivo que se pueda utilizar como soporte académico para poner en práctica distintos algoritmos de control que sean vistos de manera teórica. Dicho dispositivo quedará como recurso para ser utilizado en la materia Laboratorio de Control, de las carreras de Ingeniería del Instituto Balseiro (IB).

2.2. Objetivos

El objetivo principal del proyecto es obtener un dispositivo capaz de controlar la orientación, únicamente en el ángulo de la guiñada, siendo capaz de establecer y mantener una posición indicada.

El enfoque del proyecto consiste en realizar el dispositivo utilizando electrónica digital y sistemas modulares comerciales, para enfocarse en el ensamble y funcionamiento general del sistema. Además, se espera poder realizar el manejo del dispositivo a través de un ordenador externo, para lo cual se requiere una comunicación constante. Los requerimientos definidos para el prototipo son:

1. Requerimientos referidos al control:
 - 1.1 Establecimiento de un punto de referencia en la posición angular.
 - 1.2 Tiempo de muestreo menor a 50ms.
 - 1.3 Tiempo de establecimiento menor a 10s.
2. Requerimientos referidos al prototipo:
 - 2.1 Comunicación inalámbrica.
 - 2.2 Alimentación por baterías.

Para lograr ese objetivo principal se dividió el trabajo y se establecieron objetivos parciales, los cuales son:

- Diseño e implementación del sistema: Obtener un sistema funcional que permita tomar las mediciones, accionar el actuador y realizar pruebas a lazo abierto y cerrado.
- Modelado de la planta obtenida: Elaborar un modelo analítico de la planta a partir de analizar las ecuaciones físicas que la representan.
- Identificación y validación de los parámetros medidos: Obtener un modelo parametrizado que represente la planta con un nivel de fiabilidad que permita realizar simulaciones y diseñar los lazos de control.
- Propuesta y comparación de distintos esquemas de control: Diseñar e implementar como mínimo tres esquemas diferentes de control, utilizando el mismo actuador pero modificando el algoritmo empleado. Establecer y mantener cambios de posición con tiempos de establecimiento menor a 10 segundos.

2.3. Trabajo desarrollado

El presente Proyecto Final Integrador es una extensión del trabajo realizado en el marco de una Práctica Profesional Supervisada (PPS), en la cual se propuso la implementación de un prototipo de simulación para el control de orientación mediante rueda de reacción. Se utilizaron materiales provistos por el Centro Atómico Bariloche para el armado del mismo.

El trabajo realizado se dividió en tres grandes etapas. La primer etapa consistió en el armado del dispositivo. El desarrollo realizado abarcó el diseño e implementación del sistema electrónico, colocación del actuador, montaje de los sensores, colocación de la alimentación, articulación del sistema de comunicación, distribución de los elementos dentro de la maqueta y prueba del sistema a lazo abierto.

La segunda etapa consistió en la propuesta de un modelo matemático que represente la planta construida, medición y toma de parámetros de la planta a partir de distintas experiencias de laboratorio y validación del modelo matemático propuesto. Las experiencias que se realizaron consistieron en someter a la planta a señales de entrada de tipo escalón, chirp y secuencia aleatoria binaria. Se obtuvo un modelo en representación de funciones transferencias y un modelo en representación de variables de estado.

La última etapa consistió en la propuesta de distintos esquemas de control con el fin de ser capaces de controlar la planta dentro de un rango que cumpla con los requerimientos propuestos. Se implementaron algoritmos de lazo simple, lazo interno y externo y de control moderno, y se analizaron los resultados para estudiar la forma más eficaz de controlar la planta considerando sus restricciones.

3. Descripción del sistema

En el siguiente capítulo se hace una descripción del sistema implementado, haciendo detalle en los componentes de Hardware utilizados y en el esquema utilizado en el Software desarrollado.

3.1. Esquema experimental

La maqueta experimental diseñada consiste en un cubo plástico de 10x10x10cm dentro del cual se ubican todos los componentes electrónicos del sistema. El cubo se encuentra sobre una placa de madera, y a través de un eje rotativo encastrado en la cara inferior se logra la rotación en el plano X-Y con eje de rotación Z. Sobre el cubo se encuentra un motor eléctrico de corriente continua con una rueda de inercia de bronce de 50mm de diámetro, 4mm de espesor, y 30 gramos de masa. Además, se le agregaron dos pesos de 70 gramos, uno en cada costado, para aumentar la relación entre el momento de inercia del cubo y el momento de inercia de la rueda de reacción.

En la figura 4 se ve el prototipo armado y en la figura 5 se ve un detalle del eje sobre el cual rota el cubo:

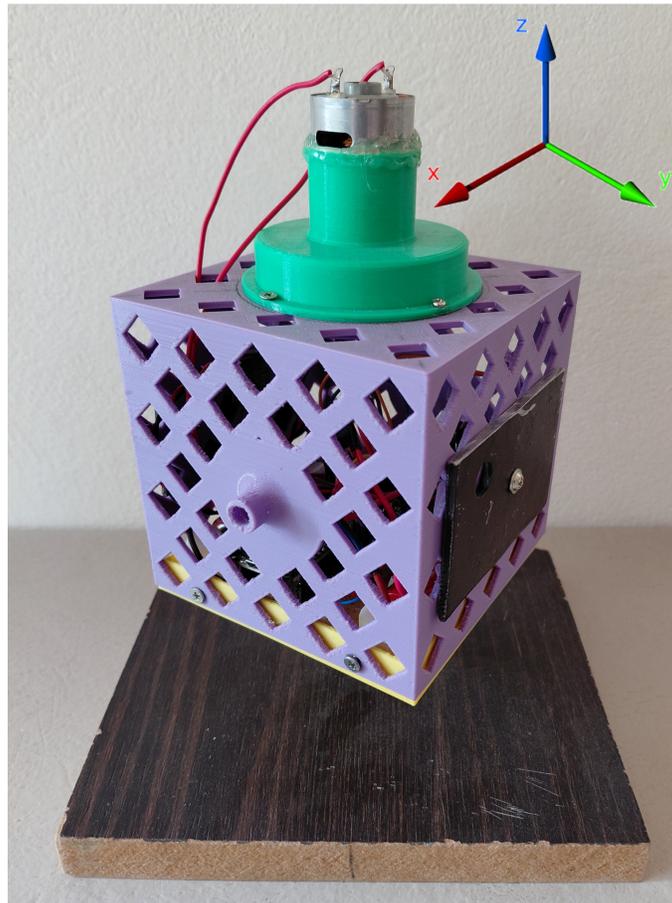


Figura 4: Prototipo final

El funcionamiento se basa en la conservación de momento angular. Al realizar una rotación de la rueda de inercia el cubo rotará con velocidad angular en sentido contrario, de manera que el total de momento angular se conserve. En ausencia de torques externos, condiciones de rozamiento nulo y velocidad angular inicial nula tanto de la rueda de reacción como del cubo el valor de reposo

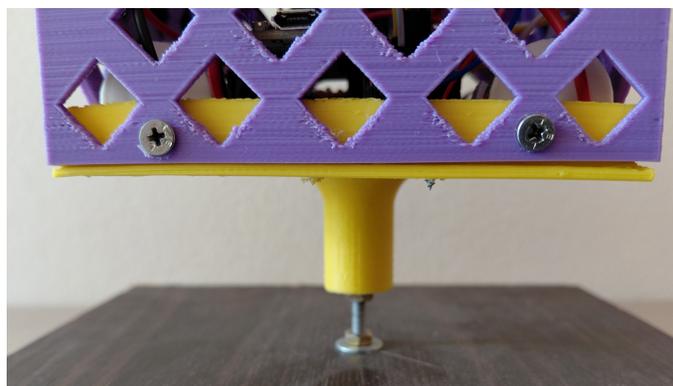


Figura 5: Detalle de eje de rotación

de la rueda de reacción es cero. Se busca trabajar cercano a ese punto de equilibrio, y controlar la orientación del cuerpo de la maqueta a partir de modificar la velocidad angular de la rueda de reacción. Debido a que el momento de inercia de la rueda de reacción es mucho menor que el del cubo, es necesario una menor velocidad angular en la maqueta para compensar el cambio en el momento angular de la rueda, lo que permite lograr gran precisión en el control de la posición angular de la maqueta. Se ampliará más sobre el modelo matemático del sistema en la siguiente sección.

Para el control del motor eléctrico se utiliza un módulo de puente H, el cual permite establecer el sentido de giro y diferentes tensiones de salida a partir de utilizar modulación por ancho de pulsos (PWM). La placa elegida para tal fin es L9110S que posee dos puentes H integrados en un mismo módulo. La velocidad angular del motor se mide con un sensor encoder magnético AS5600, el cual se basa en el efecto Hall para determinar la variación del campo magnético producido por un imán de neodimio colocado en el elemento rotativo. Dicho sensor permite la medición de posición angular y la posterior obtención de la velocidad angular de manera indirecta, sin que haya contacto entre el sensor y el elemento a medir. La medición que se logra corresponde a la velocidad angular relativa entre el disco y el cubo, es decir la velocidad angular del disco con respecto a un sistema de referencia situado en el cubo, la cual es diferente a la velocidad angular del disco para un observador externo a ese marco de referencia.

Para la medición de velocidad y posición angular del cubo, se utiliza una unidad de medición inercial (IMU) MPU6050 constituida por un acelerómetro de 3 ejes y un giróscopo de 3 ejes. Este dispositivo permite medir la velocidad, aceleración y orientación de un sistema. Los acelerómetros realizan la medición de aceleración absoluta a la cual el sensor se ve afectado, y permiten establecer el ángulo que forma el sensor con la vertical a partir de la medición de la gravedad. No tienen deriva a largo plazo, pero son muy sensibles a pequeños movimientos y al ruido, por lo que no son fiables a corto plazo [29]. Los giróscopos detectan la velocidad angular mediante la medición de la aceleración causada por efecto Coriolis en el elemento rotado. Funcionan bien para medición de movimientos cortos o bruscos, obteniendo el ángulo de orientación mediante la integración con respecto al tiempo. Sin embargo, debido a la naturaleza del proceso de integración, se acumulan los errores y el ruido de medición, por lo que no son fiables a largo plazo y se genera deriva [29]. Mediante la combinación de sensores de ambos tipos es posible obtener una medición de orientación angular fiable a corto y largo plazo. La placa elegida es la HW-290, que implementa dentro del mismo módulo la electrónica para el manejo y comunicación mediante protocolo I2C de un MPU6050, a la vez que un magnetómetro de tres ejes QMC5883L y un barómetro de presión absoluta BMP180.

Para la medición de corriente se utiliza una resistencia de bajo valor en serie con el motor eléctrico,

mediante la cual se puede obtener indirectamente el valor de corriente. La caída de tensión en la resistencia será directamente proporcional a la corriente que circule por ella, y por lo tanto a la corriente que circule por el motor. Se utiliza un circuito con amplificadores operacionales para la amplificación y el acondicionamiento de la señal.

Para la obtención de las mediciones y manejo de la variable de control se utiliza una tarjeta de desarrollo NodeMCU v2 que posee un System on Chip (SoC) ESP8266 y la electrónica necesaria para su programación y uso. El ESP8266 posee un módulo de comunicación inalámbrica que incluye protocolo WiFi IEEE802.11, por lo que es posible obtener los datos de los sensores mediante las entradas analógicas y digitales, enviarlos a un ordenador externo por medio de una red local, recibir la variable de control desde el ordenador y realizar la acción a través de las salidas.

Por último, la alimentación del sistema se realiza con baterías de litio recargables 18650, de manera que el sistema queda independiente de conexión externa con cables para su uso. Además, se agrega un puntero láser en una salida digital para utilizarlo como referencia visual en las mediciones de posición angular.

En la figura 6 se muestra un diagrama en bloques del sistema propuesto.

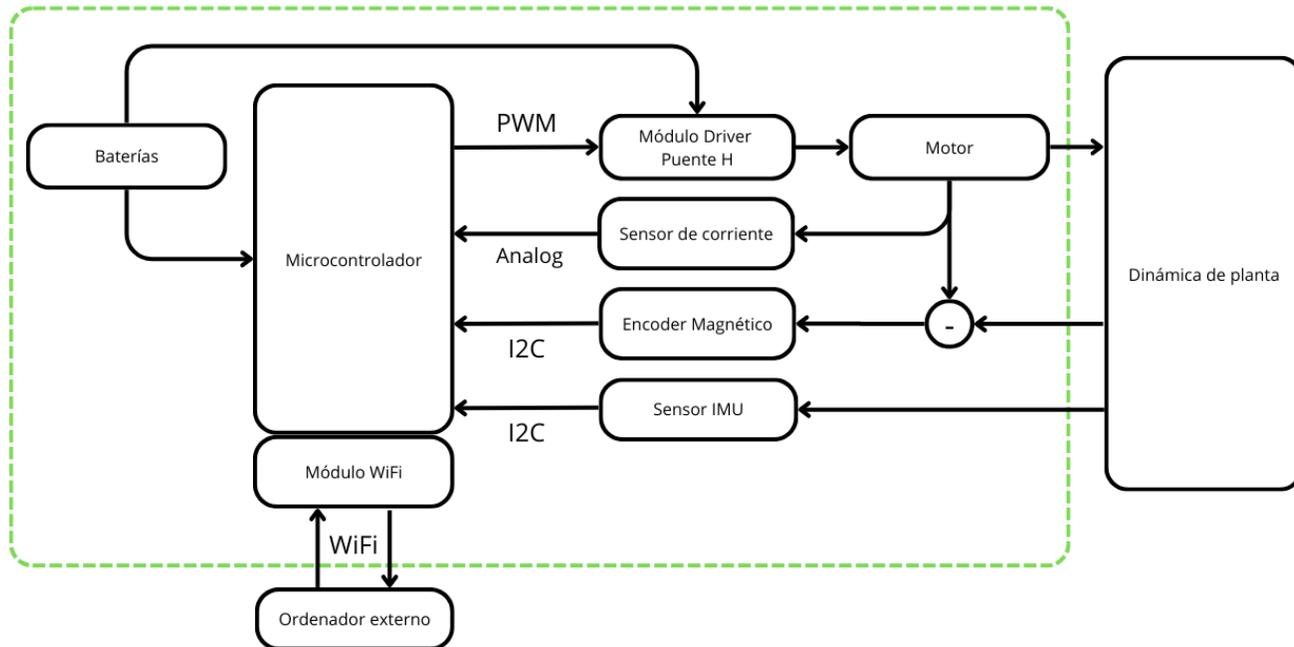


Figura 6: Diagrama del sistema

3.2. Descripción del Hardware

A continuación, se hace una descripción detallada de todos los componentes utilizados en el proyecto.

3.2.1. Tarjeta de desarrollo

El ESP8266 es un System on Chip (SoC) que implementa dentro de un mismo integrado tanto una unidad de procesamiento RISC de 32 bits con sus puertos de entrada y salida como un módulo de comunicación WiFi con capacidades de realizar comunicación en protocolo 802.11 a 2.4-2.5G, es decir que puede conectarse en las redes domésticas.

El ESP8266 es posible obtenerlo dentro de un módulo integrado, conocido como System on Module (SoM), dentro del cual se implementa el chip ESP8266 con la electrónica discreta necesaria para su programación y utilización, y una antena de radiofrecuencia para la frecuencia de operación de WiFi. El módulo más usual es el ESP12E, que además otorga acceso a todos los puertos de entrada/salida del chip. La electrónica necesaria incluye una memoria Flash, debido a que el chip solamente integra una memoria RAM. Esto implica que el canal de comunicación SPI y los pines GPIO correspondientes quedan permanentemente ocupados con la comunicación a la memoria, ya que allí se guarda el programa de usuario [8].

En este proyecto se utiliza la tarjeta de desarrollo NodeMCU v2 (figura 7), que posee un SoM ESP12E, reguladores de tensión, puerto serie microUSB y pines para entradas y salidas.

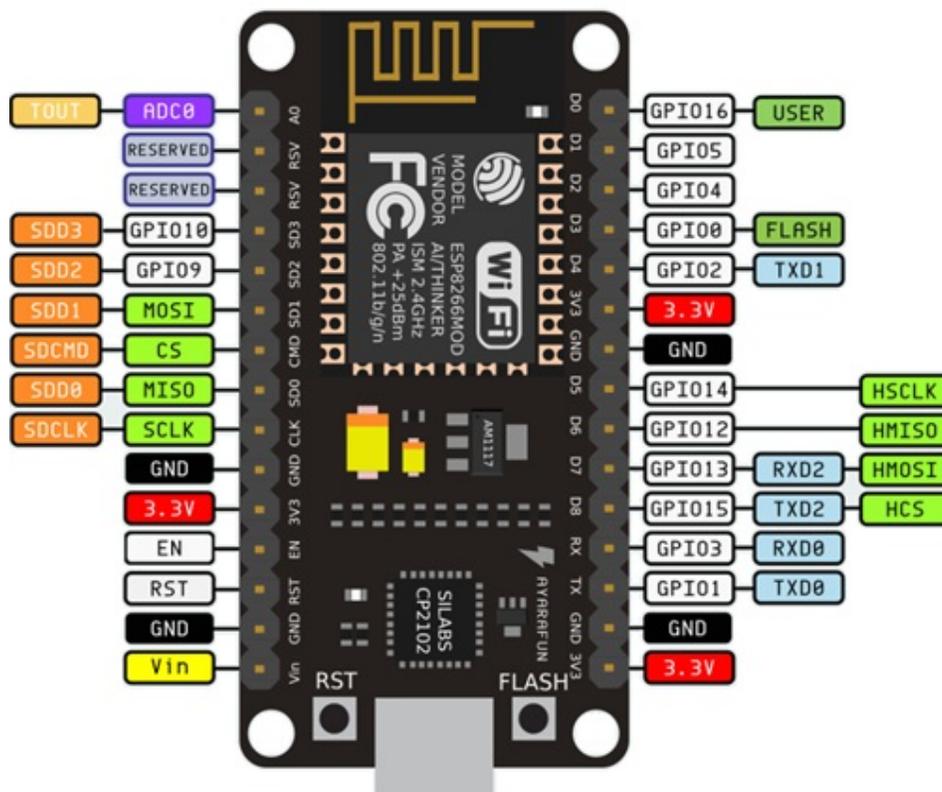


Figura 7: Pinout de la tarjeta de desarrollo NodeMCU [22]

Las características de la placa son:

- Alimentación externa 5 a 10v
- Tensión lógica (entradas/salidas) 3.3v
- Intensidad de corriente máxima por pin 12mA
- Puerto serie microUSB
- Wi-Fi Direct (P2P), soft-AP
- Stack de Protocolo TCP/IP integrado
- Potencia de salida de +19.5dBm en modo 802.11b
- CPU: Tensilica Xtensa LX3 (32 bit)
- Frecuencia de Reloj: 80MHz/160MHz
- Memoria RAM de instrucciones: 32KB
- Memoria RAM de datos: 96KB
- Memoria Flash Externa: 4MB
- 1 pin analógico con ADC 10 bits (0-3.3v)
- 17 GPIOs (6 usadas por la Flash, 2 por el puerto serie y 3 usados parcialmente durante le encendido y reset)

3.2.2. Unidad de medición inercial

El módulo utilizado en el proyecto es el HW-290 (figura 8), que incluye tres sensores dentro de una misma placa: una IMU MPU6050, un magnetómetro QMC5883L y un barómetro BMP180. Dentro del módulo se implementa la electrónica necesaria para el manejo de los sensores y ofrece una interfaz para comunicación mediante protocolo de comunicación I2C. Inicialmente se propuso la utilización del sensor magnetómetro para agregar la medición de campo magnético de la Tierra como referencia, pero luego de varias mediciones se decidió la no utilización del mismo debido a las interferencias introducidas por el actuador.



Figura 8: Módulo HW-290 utilizado

La unidad de medición inercial MPU6050 es un dispositivo que combina dentro del mismo integrado un giróscopo de 3 ejes y un acelerómetro de 3 ejes. Por eso se dice que posee 6 grados de libertad (DoF).

El acelerómetro está construido internamente con un sistema micro electromecánico (MEMS) en el cual una masa está suspendida con resortes en los tres ejes. En función de la aceleración recibida dicha masa aplicará una fuerza en los resortes, la cual puede ser medida para determinar la aceleración. En reposo el acelerómetro siempre detectará la aceleración de la gravedad. El giróscopo, por otro lado, utiliza un MEMS que consiste en brazos vibrantes sobre los tres planos del espacio sostenidos por un soporte central. El funcionamiento se basa en el efecto Coriolis, el cual afecta a los elementos que se encuentran en un sistema en rotación. Para un observador dentro del sistema no inercial en rotación, los elementos que se encuentran en movimiento previo a la rotación tienden a continuar en movimiento durante la rotación, produciendo una fuerza ficticia. Los brazos vibrantes del sensor giróscopo ejercen una fuerza proporcional a la velocidad de rotación sobre el brazo central, para poder continuar con su vibración. Esto implica que para poder tener el giróscopo en funcionamiento existe un consumo de corriente constante para la generación del clock de vibración.

Las mediciones del acelerómetro pueden ser integradas para obtener la velocidad de traslación e integradas nuevamente para obtener la posición, mientras que las mediciones del giróscopo pueden ser integradas para obtener la posición angular o derivadas para obtener la aceleración angular. El MPU6050 implementa internamente una interfaz de comunicación I2C, de manera que se puede acceder a los registros en donde se guarda la última medición realizada de cada sensor.

Existen dos maneras de tomar las mediciones del sensor MPU6050, tomando el dato crudo o tomando el dato procesado. El dato crudo es la velocidad angular o la aceleración lineal en cualquiera de los tres ejes del espacio tridimensional. El dato procesado, por otro lado, utiliza la funcionalidad del procesador interno que posee la IMU. Dicho procesador se denomina Digital Motion Processor (DMP) y es una tecnología patentada por Invensense que realiza cálculos de fusión de sensores dentro de la IMU para obtener la orientación angular. La medición se coloca dentro de un buffer de registro en modo de cuaternión, que se puede leer a través del bus I2C. La utilización de la fusión de sensores por hardware que provee el integrado permite que el dato obtenido sea de orientación angular previamente filtrado y procesado, y reduce el cálculo necesario en el microprocesador [21]. El cuaternión obtenido se procesa para convertirlo en un dato de ángulos de navegación (yaw, pitch, roll), de los cuales el ángulo que nos interesa controlar en el proyecto es el de yaw (deriva, guiñada o dirección).

3.2.3. Encoder magnético

El sensor AS5600 (figura 9) es un sensor magnético de posición angular de 12 bits de resolución, con salida analógica o digital por PWM, y comunicación I2C. Se encuentra en un módulo que permite alimentación de 3.3v. Su funcionamiento se basa en medir el vector de campo magnético de un imán imantado de manera axial, como se puede ver en la figura 10.

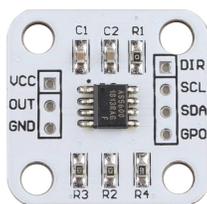


Figura 9: Módulo AS5600 utilizado

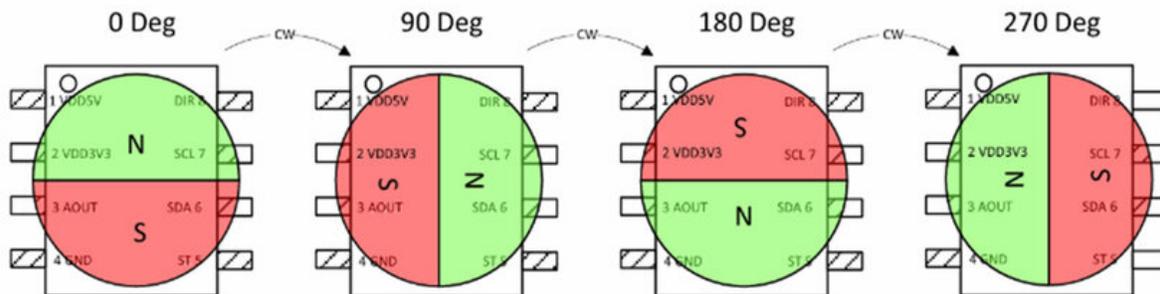


Figura 10: Rotación del campo magnético de un imán polarizado axialmente [3]

El sensor funciona correctamente si se encuentra a una distancia entre 0.5 y 3mm con respecto a un imán de neodimio polarizado axialmente, para lo cual se colocó el imán en el disco de reacción y se fijó el módulo del sensor en la caja plástica cumpliendo la distancia requerida.

La medición obtenida de este sensor es de posición angular, por la cual es necesario derivarla para obtener el dato de velocidad angular que se requiere. Esto se realiza tomando una medición y comparando la diferencia entre la medición actual y la anterior, relativo al tiempo transcurrido. Para poder realizar el cálculo de manera correcta es necesario tomar al menos dos mediciones antes de que el objeto a medir realice una rotación de más de 180 grados, de manera que se evite el aliasing.

3.2.4. Motor eléctrico

El motor utilizado es un motor eléctrico de escobillas, modelo RS-365SH-16120 (figura 11), con rango de operación entre 6 y 24v. En el eje del motor se colocó la rueda de reacción de bronce (figura 12), y el motor se colocó en un soporte plástico que se fija al cubo mediante tornillos. Como se mencionó anteriormente, la distancia entre el imán de neodimio y el sensor encoder AS5600 tiene que ser cercana a 3mm. Para ello se pegó el imán en el eje de rotación de la rueda de reacción y se colocó el sensor de manera que quede enfrente y con la distancia necesaria.



Figura 11: Detalle del motor utilizado



Figura 12: Detalle de la rueda de reacción

3.2.5. Driver puente H

Para el manejo de la corriente del motor se utilizó el módulo driver de motores L9110S (figura 13), que consiste en un módulo con dos puentes H L9110, de los cuales solamente se utilizó uno de los canales. El módulo permite la colocación de una alimentación externa y el control mediante señales de entrada de menor tensión. Al colocar una de las entradas en valor cero y la otra entrada con alguna tensión regulada por PWM el puente H envía tensión al motor en un sentido de giro, regulada por PWM pero utilizando de referencia la alimentación externa, la cual es en general mayor que la tensión utilizada para el control. Además, esto permite utilizar intensidad de corriente mayor que la máxima permitida en las GPIO del microcontrolador utilizado. En la figura 14 se ve un esquema de circuito tipo puente H.

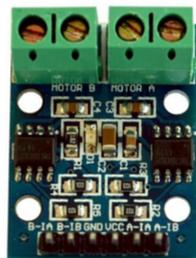


Figura 13: Módulo L9110S utilizado

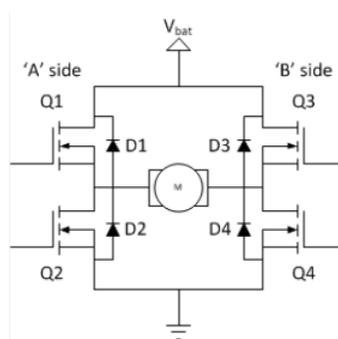


Figura 14: Esquema de un circuito de puente H

3.2.6. Medición de corriente

Con el propósito de implementar una medición de corriente del motor en el sistema, se diseñó e implementó una placa para acondicionar una señal analógica que pueda ser medida con el ADC del ESP8266. Se propuso utilizar una resistencia de bajo valor en serie con el motor, sobre la cual al medir la tensión se puede calcular el valor de corriente que circula en el circuito. Se utilizan resistencias que no impliquen una carga significativa en el circuito de manera que no se vea afectado con la inclusión de esta medición. El circuito esquemático de la placa implementada se encuentra en la imagen 15.

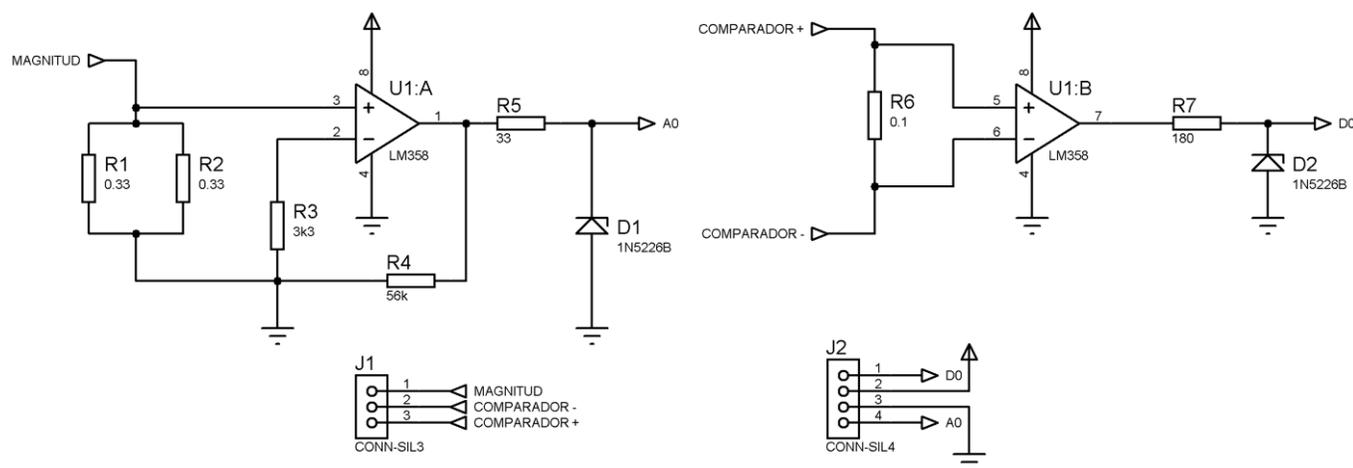


Figura 15: Circuito esquemático para medición de corriente

La placa NodeMCU tiene un único ADC de 10 bits de resolución y rango de 0-3.3v, al que se accede a través de su pin A0. Debido a que no soporta tensiones positivas mayores a 3.3v, ni tensiones negativas, se decidió utilizar la entrada analógica para medir la magnitud de la corriente eléctrica en el motor, y una entrada digital para medir el sentido de la misma. Esto se logra utilizando una resistencia de bajo valor en serie en la tierra del puente H, midiendo la corriente que circula en todo el módulo. La corriente siempre circulará en el mismo sentido, sin importar el sentido de giro del motor, con lo que la tensión medida en ese punto será siempre positiva, evitando la posibilidad de ingresar una tensión negativa en el ADC. Para medir el sentido de la corriente se coloca otra resistencia en serie con el motor, en la cual la corriente circulará hacia un lado o hacia el otro. La tensión entre los bornes de esa resistencia se colocará en la entrada de un comparador, que saturará en tensión máxima o mínima según cuál de los dos puntos esté en mayor tensión. Midiendo la salida del comparador con una entrada digital se puede identificar el signo de la corriente.

A fin de saber qué amplificación de la señal era necesaria para aprovechar al máximo la resolución del ADC se realizó una medición de la amplitud máxima del esquema propuesto. Se midió la resistencia del bobinado del motor obteniendo un valor de 18Ω , con lo cual se propuso utilizar dos resistencias de 0.33Ω en paralelo, logrando una resistencia equivalente de 0.165Ω (menor al 1\$), las cuales se colocaron en serie con el puente H. Se colocó una señal cuadrada de 1Hz y amplitud de tensión 7.2v en el motor, y se midió con un osciloscopio la tensión resultante en la resistencia. En la figura 16 se puede ver una captura de los resultados obtenidos:

Se detectó una amplitud de pico máxima de aproximadamente 150mV, y luego una bajada progresiva en la amplitud de dicha tensión. Se puede ver además que la corriente medida es siempre en valor absoluto, ya que durante el experimento el motor constantemente cambió su dirección, pero el valor de tensión medido siempre fue positivo. En base a esta medición se propuso utilizar una

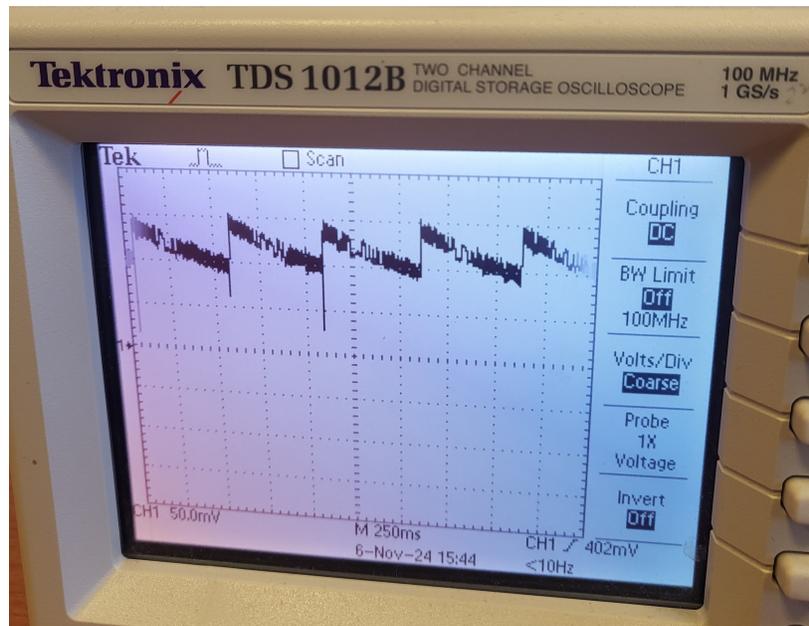


Figura 16: Resultados del experimento de medición de corriente

amplificación menor a 22, de modo que con una señal de 150mV no se llegue a 3.3v. Se diseñó un amplificador no inversor utilizando resistencias de 56kΩ y 3.3kΩ con ganancia:

$$G = 1 + \frac{56k}{3,3k} \cong 18$$

Se utilizó un amplificador operacional LM358P, que contiene dos operacionales dentro del mismo encapsulado, con el cual se implementaron ambos circuitos. La alimentación del LM358P se realizó utilizando la tensión Vin de las baterías. Debido a esto se colocaron diodos Zener de 3.3v protectores para controlar que la tensión máxima aplicada en las entradas del ESP8266 no supere ese valor. Se implementó el circuito en una plaqueta multiperforada, colocando todos los componentes y los pines para la integración del módulo en el sistema, como se puede ver en la imagen 17. Se realizaron mediciones del circuito construido y se determinó que la ganancia real lograda en el amplificador es de 15.

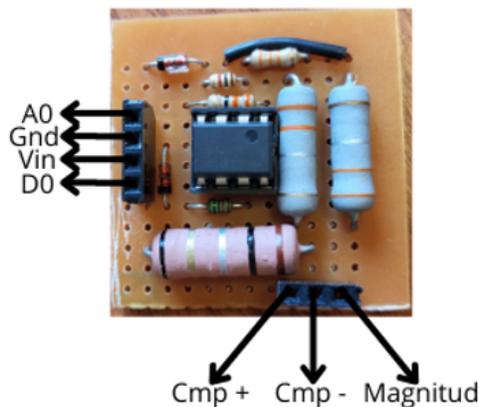


Figura 17: Placa construida

3.3. Descripción del Software

El software utilizado en el proyecto consiste en un firmware programado en el ESP8266 y un programa en lenguaje Python que se ejecuta en el ordenador externo y se comunica con el microcontrolador. El firmware se encarga de tomar las mediciones de los sensores, enviarlas mediante comunicación por red WiFi, recibir el comando de control y tomar la acción correspondiente con el actuador. El programa o script de Python recibe las mediciones que envía el microcontrolador, calcula la acción de control, la envía, guarda los datos para realizar un análisis posterior y controla el cumplimiento del tiempo de muestreo entre cada paso. En la figura 18 se muestra un diagrama de flujo del esquema de software propuesto y del camino que recorren los datos desde que son leídos en los sensores hasta que se activa el actuador.

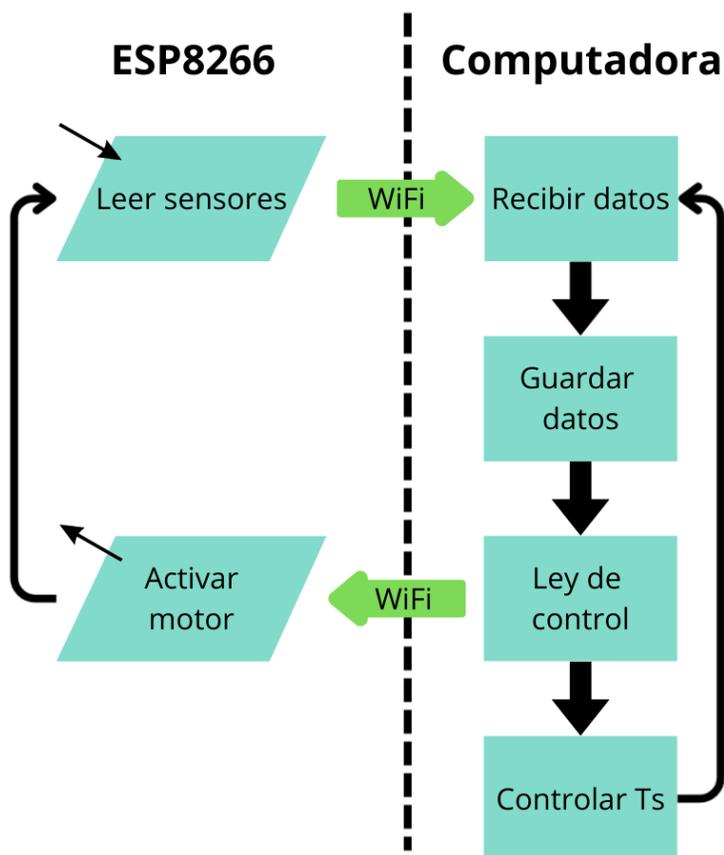


Figura 18: Esquema de la comunicación y el flujo de datos

La comunicación por red WiFi se realiza utilizando el protocolo UDP (User Datagram Protocol) el cual funciona mediante la creación de sockets, que son aplicaciones con punto de acceso a la red. Cada socket dentro de la red tiene una dirección particular de puerto, y los paquetes de datos enviados contienen una cabecera con la dirección del puerto que envía y la dirección del puerto que recibe, seguido por la longitud de mensaje, un checksum y la carga útil del mensaje. Debido a esto es necesario definir los mismos números de puerto tanto en el firmware del ESP8266 como en la aplicación Python con la que se realice la comunicación. El protocolo UDP es un protocolo de envío de paquetes de datos sin confirmación de recepción, por lo que es menos fiable que el protocolo TCP pero permite generar una comunicación directa y es eficiente en cuanto al manejo de los recursos de red.

3.3.1. Firmware ESP8266

El firmware se programó en Arduino IDE utilizando el lenguaje de programación de Arduino, el cual es un derivado del C++. Para poder realizar la programación en la placa es necesario descargar el paquete de librerías de ESP8266 a través del gestor de tarjetas de Arduino IDE.

Se dividió al programa en 4 archivos: el archivo principal que se denomina “Firmware-giroscopio.ino” en el cual se encuentran las funciones de setup() y loop() que definen el comportamiento del microcontrolador; y tres ficheros denominados “Sensores.hpp” en el cual se encuentran las funciones para el manejo de todos los módulos de sensores utilizados, “WiFi-UDP.hpp” en el cual se encuentran todas las funciones para la conexión con la red local de WiFi y la comunicación por protocolo UDP, y “config.h” en el cual se encuentra la configuración específica de la red local que se va a utilizar y la dirección IP fija que va a poseer el dispositivo. En la imagen 19 se puede ver una captura de pantalla del entorno de programación Arduino IDE y la división del programa en subarchivos.



```
void setup() {
  /*Definicion de pines*/
  pinMode(BIA, OUTPUT);
  pinMode(BIB, OUTPUT);
  digitalWrite(BIA, 0);
  digitalWrite(BIB, 0);
  analogWriteRange(1023); //Rango PWM
  analogWriteFreq(40000); //Frecuencia PWM
  pinMode(LASER, OUTPUT);
  digitalWrite(LASER, 0);
  pinMode(D_INPUT, INPUT);

  Wire.begin();
  Wire.setClock(50000);
  //Serial.begin(115200);
  delay(200);
  inicializarMPU6050();
  inicializarAS5600();

  //ConnectWiFi_AP(true);
  ConnectWiFi_STA(true);
  ConnectUDP();
  Wire.setClock(400000); // 400kHz I2C clock
  yaw=0;
  data=0;
}
```

Figura 19: Proyecto en el entorno Arduino IDE

Para la lectura de los sensores MPU6050 y AS5600 se utiliza protocolo I2C. La comunicación I2C es un protocolo de comunicación serial por bus digital, que se define mediante una arquitectura de dos cables denominados SCL y SDA, y una relación maestro-esclavo entre los participantes de la comunicación.

Dentro de un bus I2C se pueden conectar un maestro y hasta 127 esclavos, siendo el maestro quien arbitra los accesos al bus de los distintos dispositivos. El maestro es quién controla el reloj de sincronización con la frecuencia de operación, a través del cable SCL (Serial Clock), y todos los esclavos están conectados al mismo bus de manera que reciben la misma señal de clock. Además, todos

los dispositivos se conectan al mismo bus de datos SDA (Serial Data), y para poder diferenciarse cada dispositivo tiene una dirección de I2C única de 7 bits. Para el correcto funcionamiento del sistema se deben tener resistencias de pull-up en ambos buses, ya que se utiliza la norma de activo bajo y además los dispositivos tienen salida de tipo tristate para poder colocarse en modo alta impedancia cuando no es su turno dentro de la comunicación. De esta manera se asegura que la comunicación en cada momento sea entre el maestro y solo uno de los esclavos. La comunicación se inicia con el envío de una señal de inicio por medio del maestro, seguido con la dirección del esclavo y el modo de operación (lectura de datos del esclavo o escritura en los registros del esclavo). Luego de la confirmación de recepción se envía la dirección del registro que se desea leer o escribir, y por último se realiza la acción, finalizando la comunicación con una señal de fin de transmisión. Los bytes se envían de manera MSB, es decir el bit más significativo primero.

El ESP8266 no posee un módulo de comunicación I2C por hardware, por lo que la implementación se realiza de forma virtual. Esto permite elegir potencialmente cualquier combinación de pines para la comunicación. El clock interno permite una frecuencia de operación de 100kHz el pin elegido como SCL.

Para el manejo de la IMU MPU6050 se utilizó la librería “MPU6050” de Jeff Rowberg [16]. Los datos de los sensores tienen un rango entre -32768 y 32767, y la escala que representan dependen del rango elegido para cada sensor. El acelerómetro tiene las opciones de escala 2/4/8/16 (g), es decir que para un rango de escala elegido de 2g el valor máximo representará 2g y el mínimo -2g. Para el giróscopo las opciones son 250/500/1000/2000 (°/s), es decir que para un rango de escala elegido de 250 el valor máximo representará 250°/s y el mínimo -250°/s. Si se elige un rango mayor se puede medir valores más altos de aceleración o de velocidad angular, pero se pierde definición en el paso. Se eligió escala de 2g para el acelerómetro y de 500°/s para el giróscopo. Además, las mediciones que se utilizan corresponden a las mediciones preprocesadas internamente por el sensor, es decir las mediciones de los registros de DMP.

Para el manejo del sensor AS5600 se utilizó la librería “AS5600” de Rob Tillaart [15], que provee funciones para la comunicación y para el cálculo de la velocidad angular a partir de tomar las mediciones de posición.

Para la medición de corriente se utiliza la única entrada analógica, que tiene resolución de 10 bits. Para convertir el valor leído en un valor correspondiente a corriente en mA se multiplica la medición por el siguiente factor:

$$F_{ADC_to_mA} = \frac{3300mV}{1023 * 15 * 0,165\Omega} \cong 1,3$$

Por último, para el manejo del motor se utilizan las salidas con modulación por ancho de pulsos (PWM). El ESP8266 no implementa la modulación PWM por hardware, sino que la implementa por software. Para poder controlar la magnitud y el sentido de giro del motor se deben utilizar dos salidas PWM, y se definió la frecuencia de trabajo para las mismas en 400kHz.

En la imagen 20 se puede ver un diagrama de flujo del bucle principal que recorre el firmware. En el mismo se toman los datos de los sensores y se revisa la llegada de una nueva orden de control, la cual se aplica en el actuador. Cabe destacar que no se controla el cumplimiento del tiempo de muestreo en el microcontrolador, tomando todas las mediciones disponibles dentro del lapso entre la llegada de dos órdenes consecutivas. Esto es debido a que si se estableciera un tiempo de muestreo estricto en ambos extremos de la comunicación habría que realizar una sincronización, y no se ganaría ningún beneficio al realizarlo. Los registros de los sensores se actualizan con una frecuencia de 100Hz y la velocidad de comunicación del bus I2C se estableció a 400kHz (Fast mode), de manera que es posible realizar entre una y dos mediciones antes de enviar los datos vía WiFi, enviando siempre el

dato más actual que se posea. Además, la comunicación WiFi puede consumir hasta 240mA, por lo que se decidió mantener el envío de datos en la mínima medida posible, solo haciéndolo como respuesta a una acción recibida. Esto implica que es necesario enviar una acción para poder empezar la comunicación, y que el tiempo de muestreo y la duración total de la comunicación y el control de la planta lo establece la aplicación del ordenador externo.

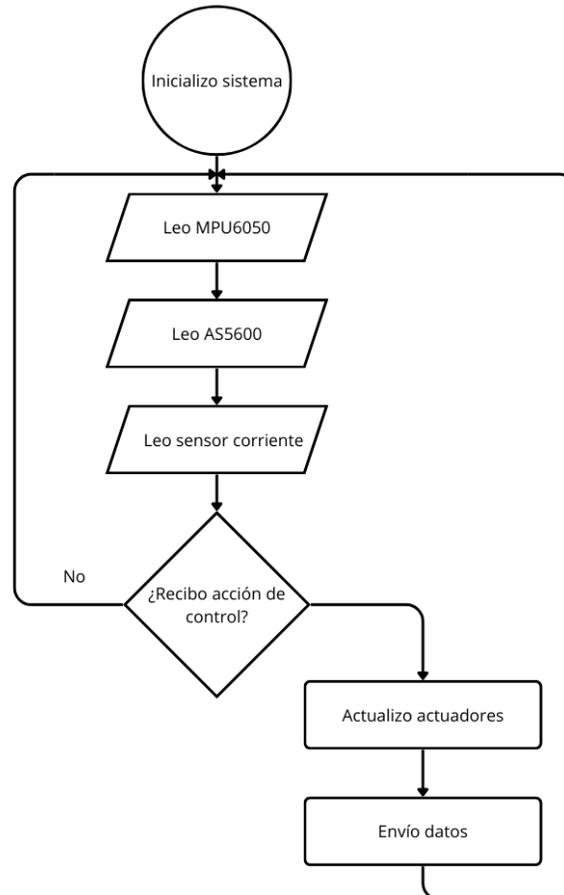


Figura 20: Diagrama de flujo del firmware en ESP8266

3.3.2. Programa Python

El programa que se implementó en lenguaje Python se encarga de recibir los datos tomados por el ESP8266, calcular la acción de control en base a la orientación deseada, los datos actuales y el esquema de control propuesto, enviar la acción de control y establecer el tiempo entre cada iteración. Además, guarda los datos en un archivo de tipo *.dat* para realizar un análisis de los resultados y provee un medio de interacción para el usuario a través de la consola. Se utiliza la librería *socket* para el manejo de la comunicación por red y la librería *time* para el control del tiempo de muestreo.

La secuencia del programa consiste en establecer un tiempo de duración total del experimento, e inicializar una matriz para guardar todos los datos de las mediciones durante el tiempo transcurrido, además de establecer el tiempo de muestreo T_s . Al dar inicio al programa se envía una acción inicial de control de valor 0, y se entra en un ciclo repetitivo que se ejecuta durante todo el tiempo de duración del experimento. Dentro del ciclo se toma el tiempo inicial, se leen los datos en el buffer del socket de comunicación, se calcula la ley de control (la cual se linealiza y se escala para entrar

en los límites definidos por el microcontrolador), se envía la variable de control y se toma el tiempo final. Si el tiempo transcurrido dentro de la iteración del ciclo es menor que el tiempo de muestreo entonces se espera durante el tiempo restante antes de comenzar la siguiente iteración. En la imagen 21 se puede ver el diagrama de flujo de la aplicación.

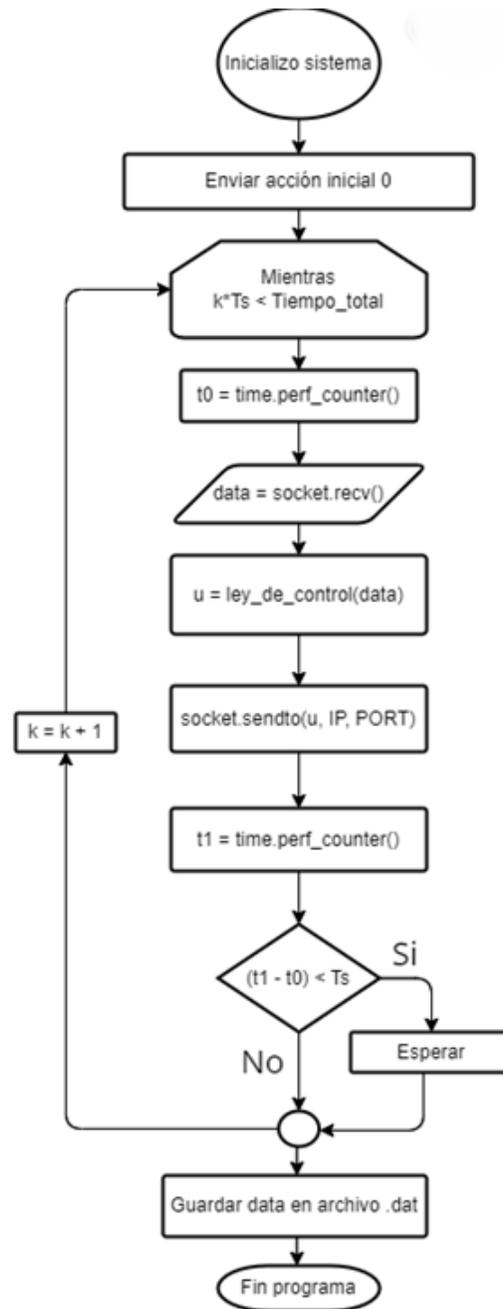


Figura 21: Diagrama de flujo del script de Python

3.3.3. Caracterización del tiempo de muestreo

Para validar el software utilizado se realizó una caracterización del tiempo de muestreo, debido a que es necesario poder definir el tiempo de muestreo mínimo que se puede utilizar en el sistema

de manera que se garantice el cumplimiento del mismo. Las principales limitaciones están dadas por la velocidad de la comunicación UDP, la tasa de muestreo interno de los sensores utilizados y la velocidad de la comunicación I2C. Además, se establece un grado de preferencia alto al proceso de ejecución del programa Python, para asegurar la prioridad dentro del sistema operativo del ordenador. Se presupone que la velocidad de cálculo del algoritmo de control, el guardado de los datos y la interpretación de la acción de control son operaciones que no conllevan un retardo considerable dentro del sistema.

Se realizó un experimento en el cual se propusieron distintos tiempos de muestreo requerido y se guardaron los datos del tiempo final medido luego de realizar cada iteración del ciclo. Se realizaron únicamente las acciones que conllevan el mayor retardo de ejecución, es decir tomar todas las mediciones y realizar la comunicación bidireccional, pero enviando una acción de control de valor 0 en todos los casos. El experimento se llevó a cabo durante 10000 iteraciones para cada caso. En la figura 22 se puede ver un gráfico de dispersión para los diferentes tiempos requeridos:

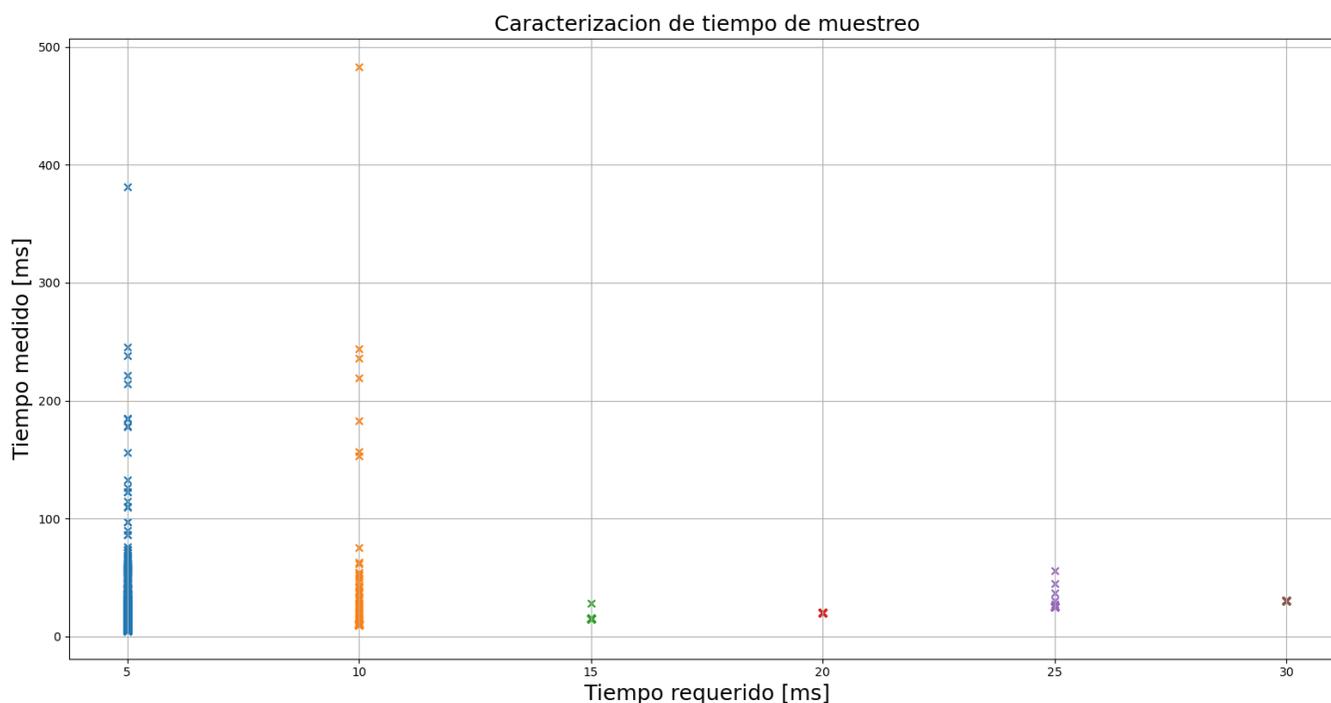


Figura 22: Resultados del experimento

Los resultados se resumen en la siguiente tabla:

Tiempo requerido	Media estadística	Desvío estándar
5	10.129	10.365
10	10.244	6.870
15	15.002	0.128
20	20.001	0.005
25	25.008	0.389
30	30.002	0.006

En base a estos resultados se determinó que el tiempo de muestreo será de 20ms, con el cual se espera cumplir con los requerimientos necesarios de frecuencia para el control de la planta.

4. Identificación y validación

El modelo matemático del dispositivo, también denominado modelo dinámico, es el conjunto de ecuaciones diferenciales que sirven para describir el comportamiento dinámico del sistema, siendo el primer paso para facilitar el diseño de un control realimentado. El modelo analítico se puede obtener a partir de desarrollar las ecuaciones determinadas por los principios físicos del sistema, mientras que un modelo experimental puede obtenerse a partir de medir la respuesta del sistema a distintas excitaciones de entrada [12].

En el siguiente capítulo se desarrolla el modelo matemático de la planta con el fin de encontrar las funciones de transferencia de manera analítica y su representación por espacio de estados. Luego se realiza un reconocimiento del actuador para la eliminación de sus comportamientos no lineales en la mayor medida posible.

Se continúa con una identificación empírica de los parámetros de la planta utilizando la metodología propuesta en [18], en donde se propone elegir un posible modelo a identificar, definir un experimento del cual se tomarán los datos, definir la metodología y el criterio de ajuste de los datos con el modelo, y definir un método de validación del modelo obtenido. En primera instancia se realiza la identificación a partir de someter la entrada a excitaciones de tipo escalón, la cual es una señal que permite analizar el comportamiento del sistema frente a un cambio brusco en la entrada seguido de un régimen estacionario, abarcando comportamientos de frecuencias altas y frecuencias bajas. Por otro lado, la validación de los parámetros obtenidos se efectúa a partir de mediciones de tipo chirp y secuencia aleatoria binaria, las cuales son señales con mayor energía frecuencial en un rango de frecuencias de interés, elegidas a partir de las frecuencias dominantes identificadas con el método de los escalones. Consecutivamente, se desarrolla una identificación por espacio de estados utilizando un modelo de caja gris y mediciones de tipo chirp, mientras que la validación en este caso se efectúa comparando la respuesta al escalón lograda por el modelo con la respuesta al escalón medida en los experimentos realizados previamente.

4.1. Modelado matemático

La variable de control que se utiliza en el proyecto es la tensión aplicada en el motor, que denominaremos $u(t)$, la cual causa una corriente $i(t)$ en el motor, que produce un torque en el eje del motor y la rueda de reacción, produciendo a su vez una reacción igual y contraria en el cubo debido a la conservación del momento angular.

Para iniciar el estudio del modelo matemático se tomaron las ecuaciones de partida que describen estos fenómenos, y se las desarrolló de manera que se obtengan las funciones transferencia que describan al dispositivo como un sistema lineal e invariante en el tiempo (SLIT). En la figura 23 se puede ver un esquema conceptual que ayuda a entender cómo es la disposición de los elementos y cómo se relacionan los sistemas físico y eléctrico.

Según la ley de las tensiones de Kirchhoff, la tensión aplicada $u(t)$ será igual a la suma de la caída de tensión producida por la resistencia equivalente del motor R , más la caída de tensión producida por la inductancia equivalente del motor L , más la tensión producida por la fuerza contraelectromotriz según el factor de constante eléctrica K_V . Esa relación se retrata en la siguiente ecuación:

$$u = Ri + Li + K_V(\dot{\theta}_D - \dot{\theta}_C) \quad (1)$$

Debido a que las ecuaciones se escriben para un observador externo, ubicado fuera del marco de referencia del cubo, es necesario aclarar que la fuerza contraelectromotriz depende de la velocidad

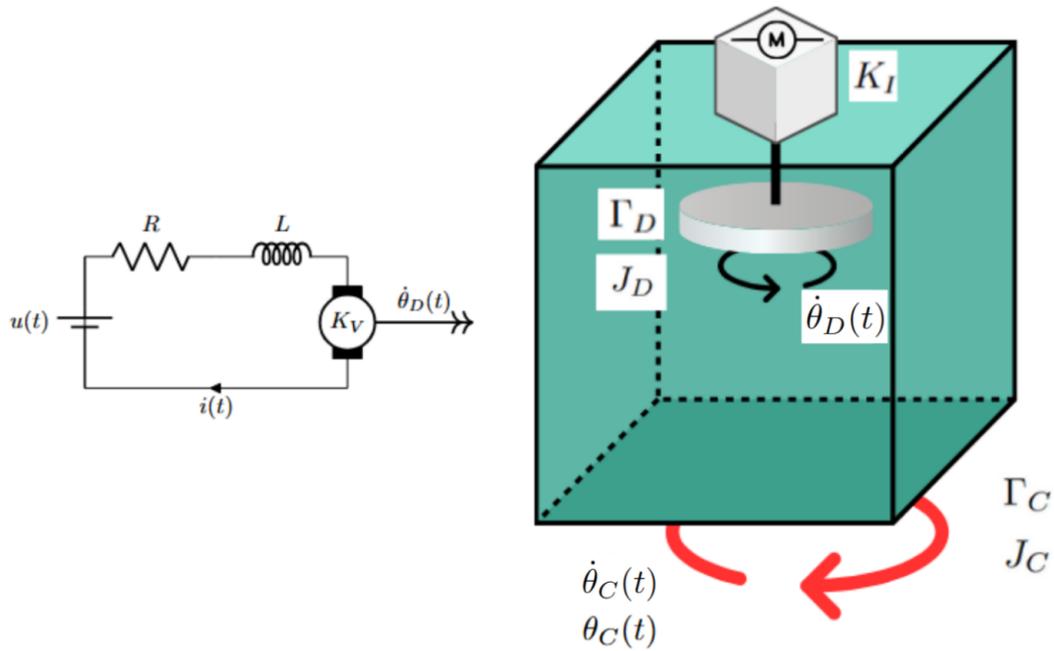


Figura 23: Esquema conceptual del gir6scopo

angular del motor $\dot{\theta}_D(t)$ vista desde dentro del cubo. Por lo tanto se debe considerar la diferencia entre la velocidad angular del disco y la velocidad angular del cubo $\dot{\theta}_C(t)$. Sin embargo, si se toma la hip6tesis de que la velocidad del disco es mucho mayor que la velocidad del cubo, se puede reducir la ecuaci6n a:

$$u = Ri + Li\dot{ } + K_V\dot{\theta}_D \tag{2}$$

Por otro lado, el torque aplicado por el motor T_M es directamente proporcional a la corriente que circula por el mismo, seg6n el factor de proporcionalidad K_I . Esto se representa con la siguiente ecuaci6n:

$$T_M = K_I i \tag{3}$$

Si se aplican las leyes de movimiento de Newton en sistemas rotacionales, el torque aplicado por el motor menos el torque producido debido a las fricciones del disco $T_{Friccion}^D$ ser6n iguales al torque aplicado sobre el disco con respecto al cubo T_C^D . Eso se resume en la siguiente ecuaci6n:

$$T_M - T_{Friccion}^D = T_C^D \tag{4}$$

En vista de que se est6 buscando el modelo din6mico lineal del sistema, se toma la hip6tesis de que la fuerza de rozamiento es din6mica, es decir que no se considera la fuerza de rozamiento est6tica. El torque producido por la fuerza de rozamiento din6mica depende de una constante de rozamiento Γ_D y de la velocidad angular del disco. Debido a que la velocidad angular del disco tiene que ser con respecto al cubo, nuevamente es necesario considerar la diferencia entre ambas:

$$T_{Friccion}^D = \Gamma_D(\dot{\theta}_D - \dot{\theta}_C) \tag{5}$$

Del mismo modo, si se considera la hip6tesis anterior se puede reducir a:

$$T_{Friccion}^D = \Gamma_D\dot{\theta}_D \tag{6}$$

Además, el torque aplicado en el disco es igual al momento de inercia del disco J_D por la aceleración angular del disco $\ddot{\theta}_D(t)$, es decir:

$$T_C^D = J_D \ddot{\theta}_D \quad (7)$$

Al reemplazar las equivalencias 3, 6 y 7 en 4 se obtiene:

$$K_I i - \Gamma_D \dot{\theta}_D = J_D \ddot{\theta}_D \quad (8)$$

En ausencia de torques externos, el torque del cubo con respecto al disco T_D^C será equivalente al torque del disco con respecto al cubo, pero con signo opuesto. Esto se debe a la conservación de momento angular. Luego, si se considera el torque producido por el rozamiento del cubo $T_{Friction}^C$ se obtiene el torque aplicado sobre el cubo T_C , de manera homóloga al caso anterior:

$$T_D^C - T_{Friction}^C = T_C \quad (9)$$

El torque producido por el rozamiento depende de una constante de rozamiento dinámica Γ_C y de la velocidad angular del cubo, mientras que el torque aplicado al cubo es igual a la inercia del cubo J_C por su aceleración angular $\ddot{\theta}_C(t)$. Al reemplazar las equivalencias en la ecuación 9 se obtiene:

$$-J_D \ddot{\theta}_D - \Gamma_C \dot{\theta}_C = J_C \ddot{\theta}_C \quad (10)$$

Las ecuaciones 2, 8 y 10 son las ecuaciones iniciales que representan la dinámica de los fenómenos físicos de la planta. A partir de esas ecuaciones se hace un desarrollo matemático para obtener las funciones transferencia de interés, las cuales serán la transferencia de la corriente en función de la tensión $\frac{I}{U}$, la velocidad angular del disco en función de la tensión $\frac{\omega_D}{U}$ y la velocidad angular del cubo en función de la velocidad angular del disco $\frac{\omega_C}{\omega_D}$, donde además se hace uso de la notación $\dot{\theta} = \omega$. El objetivo principal no es obtener los valores de los parámetros, sino lograr una idea acerca del orden de los polinomios que forman la función, para saber cuántos polos y ceros esperar en la identificación empírica de la planta.

Para hallar $\frac{I}{U}$ se comienza por aplicar la transformada de Laplace, definida como $\mathcal{L}\{f(t)\} = F(s) = \int_0^\infty f(t)e^{-st} dt$, en la ecuación 8:

$$K_I I(s) - \Gamma_D \omega_D = J_D s \omega_D \quad (11)$$

Reordenando los términos se obtiene:

$$\omega_D = I(s) \frac{K_I}{J_D s + \Gamma_D} \quad (12)$$

Luego se aplica la Transformada de Laplace en la ecuación 2 y se reordenan los términos:

$$U(s) = (R + Ls)I(s) + K_V \omega_D(s) \quad (13)$$

Reemplazando 12 en 13:

$$U(s) = (R + Ls)I(s) + \frac{K_V K_I}{J_D s + \Gamma_D} I(s) \quad (14)$$

Aplicando denominador común forzado:

$$U(s) = \frac{(R + Ls)(J_D s + \Gamma_D) + K_V K_I}{J_D s + \Gamma_D} I(s) \quad (15)$$

Se encuentra la función transferencia deseada, la cual podemos ver que tiene un cero y dos polos:

$$\frac{I(s)}{U(s)} = \frac{J_D s + \Gamma_D}{(R + Ls)(J_D s + \Gamma_D) + K_V K_I} \quad (16)$$

Para hallar la transferencia $\frac{\omega_D}{U}$ se reordena la ecuación 12:

$$I(s) = \omega_D \frac{J_D s}{K_I} + \omega_D \frac{\Gamma_D}{K_I} \quad (17)$$

Y se reemplaza en la ecuación 13:

$$U(s) = (R + Ls)\omega_D \frac{J_D s}{K_I} + \omega_D \frac{\Gamma_D}{K_I} + K_V \omega_D(s) \quad (18)$$

Distribuyendo los términos:

$$U(s) = \left(L \frac{J_D}{K_I} s^2 + (R \frac{\Gamma_D}{K_I} + L \frac{J_D}{K_I}) s + (R \frac{\Gamma_D}{K_I} + K_V) \right) \omega_D(s) \quad (19)$$

Se puede obtener la función transferencia, la cual vemos que tiene dos polos y ningún cero:

$$\frac{\omega_D(s)}{U(s)} = \frac{1}{L \frac{J_D}{K_I} s^2 + (R \frac{\Gamma_D}{K_I} + L \frac{J_D}{K_I}) s + (R \frac{\Gamma_D}{K_I} + K_V)} \quad (20)$$

Por último, para hallar la transferencia $\frac{\omega_C}{\omega_D}$ se aplica la Transformada de Laplace en la ecuación 10:

$$-J_D s \omega_D(s) - \Gamma_C \omega_C(s) = J_C s \omega_C(s) \quad (21)$$

Reordenando la ecuación 21:

$$-J_D s \omega_D(s) = (J_C s + \Gamma_C) \omega_C(s) \quad (22)$$

Finalmente se halla la función transferencia, la cual vemos que tiene un polo y un cero en el origen, además de tener ganancia negativa:

$$\frac{\omega_C(s)}{\omega_D(s)} = \frac{-J_D s}{(J_C s + \Gamma_C)} \quad (23)$$

Para finalizar el modelo del sistema se debe incluir una integración en tiempo, que en dominio de Laplace se representa con $\frac{1}{s}$, de la velocidad angular del cubo a fin de hallar la posición angular. En la figura 24 se puede ver un diagrama en bloques que representa al sistema.

La función de transferencia $\frac{\omega_C}{U}$ se halla mediante la multiplicación de 20 y 23:

$$\frac{\omega_C(s)}{U(s)} = \frac{-J_D s}{(J_C s + \Gamma_C) \left(L \frac{J_D}{K_I} s^2 + (R \frac{\Gamma_D}{K_I} + L \frac{J_D}{K_I}) s + (R \frac{\Gamma_D}{K_I} + K_V) \right)} \quad (24)$$

La función transferencia total del sistema es:

$$\frac{\Theta_C(s)}{U(s)} = \frac{-J_D}{(J_C s + \Gamma_C) \left(L \frac{J_D}{K_I} s^2 + (R \frac{\Gamma_D}{K_I} + L \frac{J_D}{K_I}) s + (R \frac{\Gamma_D}{K_I} + K_V) \right)} \quad (25)$$

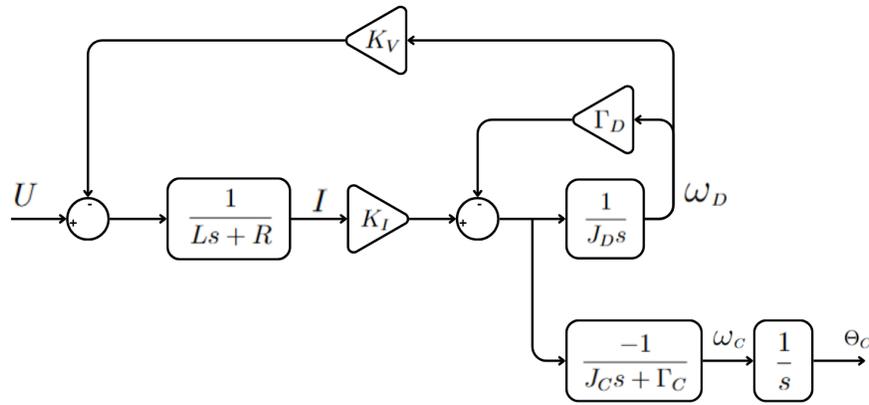


Figura 24: Diagrama de bloques de la planta

La transferencia total del sistema es una función con tres polos y ningún cero, al haber una cancelación cero-polo dada por el último integrador existe una reducción del orden total del sistema. Si se realizaran mediciones de la posición angular del cubo no se podría apreciar esa dinámica interna, ya que se vería solamente la respuesta característica correspondiente a un sistema de tres polos. Para poder observarla y corroborar su existencia, y debido a que la relación entre la velocidad y la posición angular del cubo es conocida y no necesita medirse, se determinó realizar la identificación de planta tomando mediciones de la corriente, de la velocidad angular del cubo y de la velocidad angular del disco.

La representación del sistema por variables de estado está dada por:

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t) \\ y(t) = Cx(t) + Du(t) \end{cases} \quad (26)$$

El vector de estados elegidos es:

$$x = \begin{bmatrix} i \\ \omega_D \\ \omega_C \\ \theta_C \end{bmatrix} \quad (27)$$

Con lo cual el desarrollo de la representación en variables de estado, a partir de las ecuaciones 2, 8 y 10 es:

$$\begin{bmatrix} \dot{i} \\ \dot{\omega}_D \\ \dot{\omega}_C \\ \dot{\theta}_C \end{bmatrix} = \begin{bmatrix} -\frac{R}{L} & -\frac{K_V}{L} & 0 & 0 \\ \frac{K_I}{J_D} & -\frac{\Gamma_D}{J_D} & 0 & 0 \\ -\frac{K_I}{J_C} & \frac{\Gamma_D}{J_C} & -\frac{\Gamma_C}{J_C} & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} i \\ \omega_D \\ \omega_C \\ \theta_C \end{bmatrix} + \begin{bmatrix} \frac{1}{L} \\ 0 \\ 0 \\ 0 \end{bmatrix} u \quad (28)$$

$$y = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} i \\ \omega_D \\ \omega_C \\ \theta_C \end{bmatrix} + 0u$$

4.2. Identificación del actuador

Como primer paso de la identificación de planta se decidió realizar un reconocimiento de la dinámica del actuador, y realizar una linealización de su respuesta, puesto que la respuesta lineal del

motor es una de las hipótesis tomadas en el modelo. Para poder realizar el control es necesario que el actuador opere de forma lineal, de modo que sea consistente con un modelo de sistema lineal e invariante en el tiempo.

Se comenzó realizando mediciones del motor para obtener un estimativo de la velocidad terminal del motor según distintos valores de la variable de control enviada. La variable de control se normalizó entre -1 y 1, donde -1 equivale a la máxima tensión negativa que se le puede enviar al motor, 1 equivale a la máxima tensión positiva y 0 equivale a apagar el motor por completo. El experimento realizado consistió en colocar un escalón de tensión sobre el motor a los 2 segundos durante un tiempo total de 40 segundos de medición, para poder analizar la curva de respuesta y la velocidad final en distintos valores de variable de control enviada. En la imagen 25 se grafican los resultados. Durante el experimento se mantuvo al cubo fijo en una posición impidiendo su rotación, para medir solamente la velocidad angular del disco.

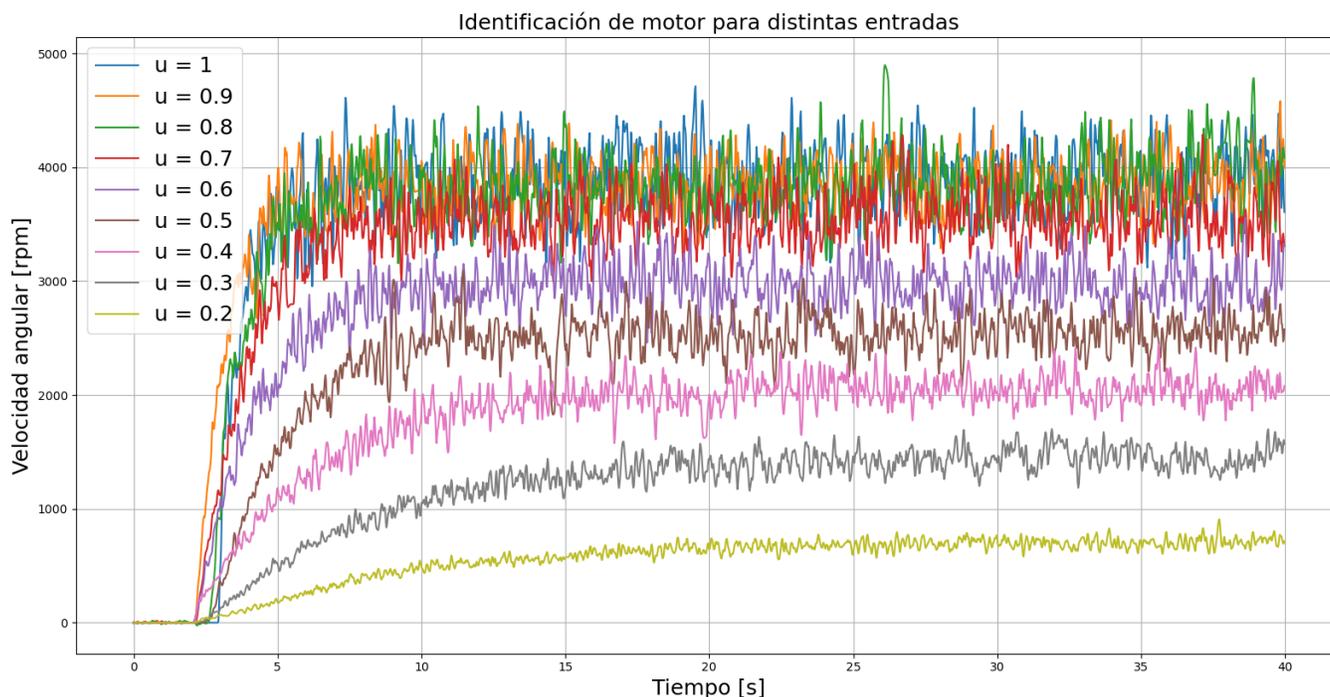


Figura 25: Respuesta temporal del motor a entradas de tipo escalón

Cabe destacar que las mediciones de velocidad angular tomadas por el encoder magnético se filtraron utilizando un filtro digital de tipo Butterworth de orden 2 y frecuencia de corte 2 rad/seg. La frecuencia se eligió luego de hacer una primera medición en la que se detectó ruido y se observó una dinámica aproximada con constante de tiempo de 2seg, lo cual es una frecuencia de 0,5 rad/seg. Utilizando un filtro con frecuencia de corte 4 veces mayor a la frecuencia observada se asegura que no se oculte la dinámica de interés. En la imagen 26 se puede observar una comparación entre la medición sin filtrar y la medición filtrada con distintas frecuencias de corte, donde se ve una respuesta similar en cuanto a la atenuación del ruido en filtros con menor frecuencia de corte, pero un retardo más marcado, siendo el de 2rad/seg el más eficiente.

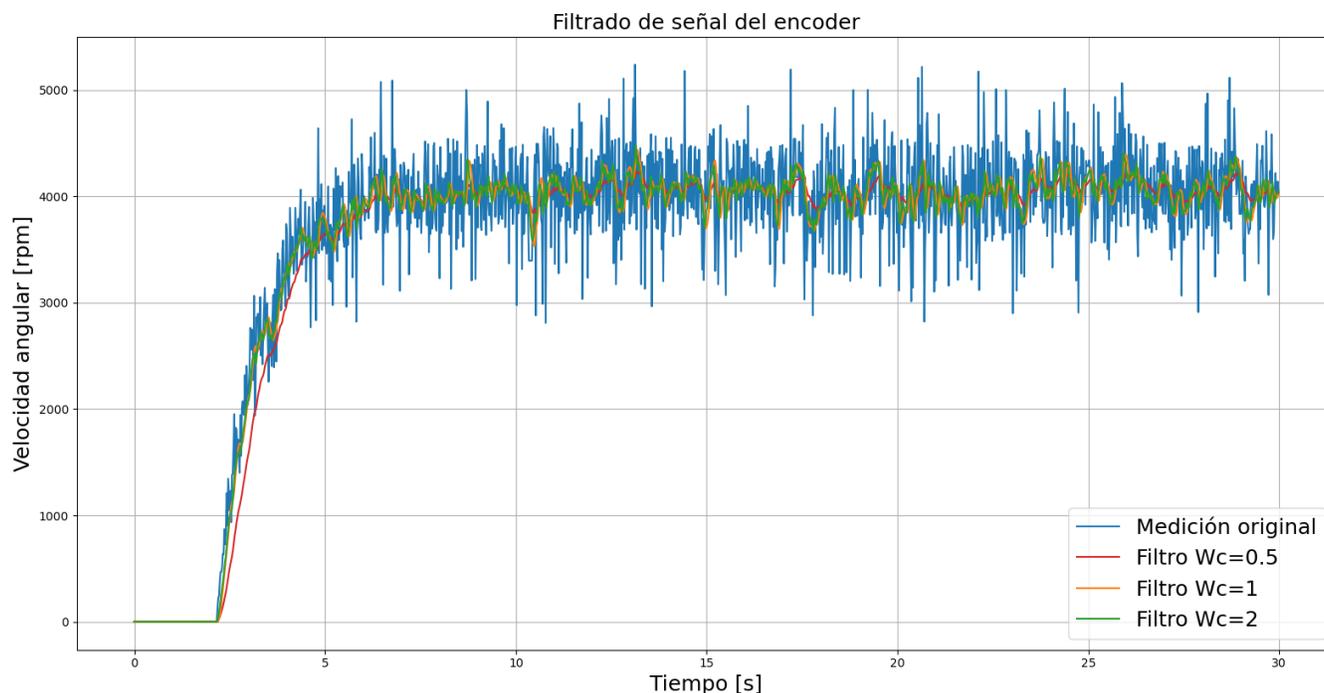


Figura 26: Análisis de filtros para señal de encoder

Utilizando un valor de 4000rpm como supuesto para la velocidad lograda con una entrada de valor 1, se procedió a realizar un gráfico de dispersión en el cual se puede ver la velocidad final lograda en comparación con el valor de la entrada, en ambos casos normalizado entre -1 y 1. Se detectó que para valores de entrada menores a 0.2 el motor no podía comenzar a girar, debido a que no vencía el rozamiento estático. Para evitar ese comportamiento no lineal se decidió que se buscaría la menor tensión posible a la que se comience a ver una respuesta de movimiento y se linealice a partir de ese punto, priorizando evitar la aparición de una “banda muerta” pero sin dejar de observar que se genera una discontinuidad en el paso por cero. En la figura 27 se observa el gráfico de dispersión, donde además se dibujaron dos líneas rectas que se propusieron como el objetivo ideal de funcionamiento.

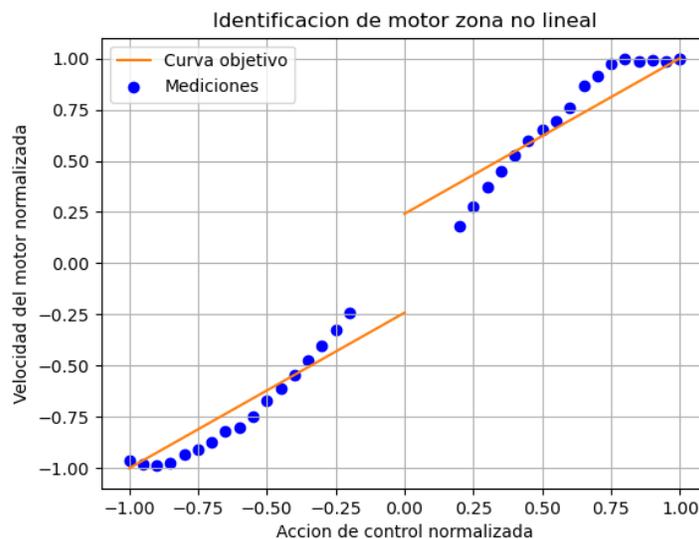


Figura 27: Mediciones para identificación de alinealidad

A partir del objetivo ideal de funcionamiento y de las mediciones obtenidas, se realizó una proyección para determinar qué valor de entrada se debe enviar para obtener realmente el valor final de velocidad deseado. Se construyó una tabla de consulta, o *look-up table*, a partir de la cual se genera una predistorsión a lazo abierto de la señal enviada. De esta manera se linealiza la acción de control enviada, estableciendo una respuesta real más cercana al caso ideal. Los datos fueron guardados de manera permanente en un archivo denominado “LookUp_Table_Linear.dat”, el cual es necesario mantener en la misma carpeta en donde se encuentra el programa de Python para que se ejecute correctamente. En la figura 28 se observa un gráfico de dispersión en el cual se muestra la acción enviada previo a la predistorsión comparado con la velocidad final obtenida luego de haberla aplicado.

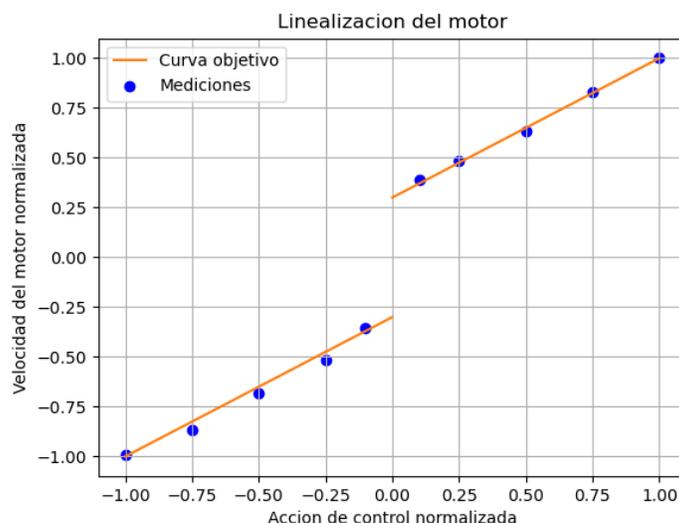


Figura 28: Mediciones para comprobación de esquema propuesto

4.3. Identificación de planta

La intención de la identificación de planta es obtener un modelo matemático simplificado que contenga las dinámicas dominantes de la planta, que se pueda utilizar para diseño, simulación y comparación de los resultados. Para ello, se busca obtener los parámetros de los ceros, polos y ganancia de las funciones transferencia, es decir funciones con la siguiente estructura:

$$G(s) = K \frac{\prod_i (s - z_i)}{\prod_j (s - p_j)}$$

4.3.1. Velocidad angular del cubo

Para comenzar la identificación se realizaron mediciones utilizando como variable de entrada la acción de control u y midiendo como variable de salida la velocidad angular del cubo ω_C . De esta manera la función transferencia que se busca hallar será $\frac{\omega_C}{U}$, vista en 24, a la cuál luego se le incluirá un integrador, ya que la variable objetivo a controlar es la posición angular del cubo. La identificación se realizó sobre la velocidad angular del cubo debido a que la medición de posición angular puede tener errores incluidos en la integración, por lo que se decidió evitar ese paso. Además, al proponer una función objetivo con un derivador puro, es decir con un cero en el origen, se puede perder

la información luego del integrador debido a una cancelación cero-polo. Al ser una medición de la respuesta de la planta, la velocidad angular es suficiente para poder representarla de buena manera.

El experimento realizado consistió en enviar distintos escalones de tensión durante un tiempo de 40 segundos, comenzando a los 2 segundos, y luego enviar tensión cero y continuar la medición durante 40 segundos más para ver la dinámica del frenado. En la figura 29 se grafican los resultados de un experimento con escalón de tensión de valor 70 %. Se observa una dinámica inicial con un crecimiento en la velocidad angular, y luego un decaimiento de la misma hasta no haber velocidad en el cubo. En ese momento, al mantener constante la tensión aplicada al motor ya no se produce ningún cambio en la velocidad angular del cubo.

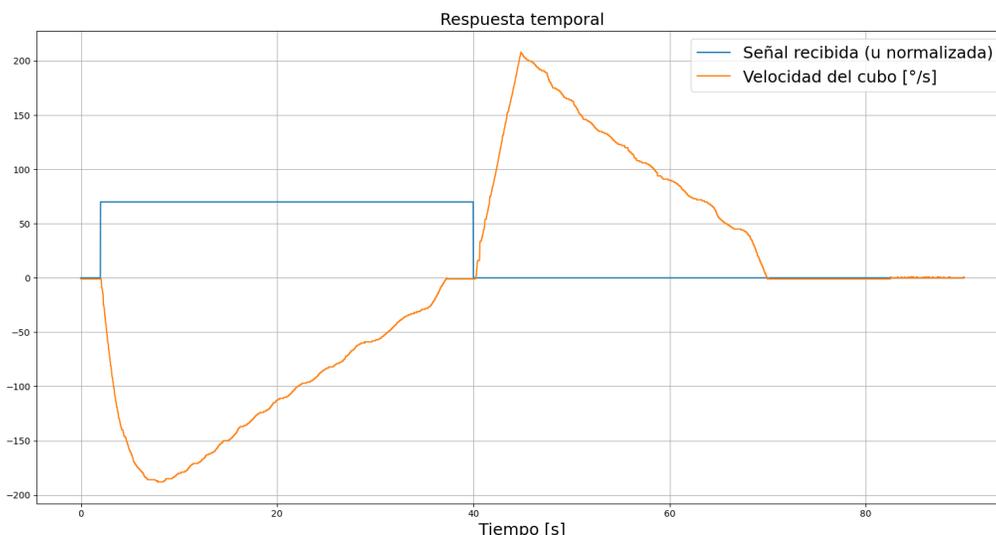


Figura 29: Gráfico temporal de respuesta a entrada escalón de $u=70\%$

La dinámica de apagado vemos que es muy diferente a la dinámica de encendido, ya que al momento que el motor deja de aplicar torque el cubo comienza a girar hacia el lado contrario, con un cambio en la velocidad angular muy marcado, y luego de llegar a un máximo desciende de velocidad hasta detenerse por completo. Por inspección visual podemos ver que las curvas de comportamiento son distintas, por lo cual se puede decir que existe una no linealidad que no se incluyó en el modelo.

Además, se puede ver que la hipótesis de que la velocidad angular del cubo es mucho menor que la velocidad angular del disco, tomada en 2 y en 6, es válida ya que la velocidad lograda por el disco fue del orden de los 3500-4000rpm mientras que la velocidad lograda por el cubo es del orden de 150-200°/seg, es decir 25-30rpm.

En la figura 30 se pueden ver los resultados del experimento para diferentes valores del escalón de tensión de entrada. Se determinó, mediante inspección visual, que la dinámica de la planta si bien es muy similar en todos los casos, presenta diferentes constantes de tiempo según la amplitud de la señal de entrada. Esto significa que la respuesta temporal depende de la magnitud de la entrada, e implica que el modelo de sistema lineal e invariante en el tiempo no puede representar dicho comportamiento. Se consideró que la mejor manera de representar el comportamiento es mediante una estimación estadística de los datos, realizando un ajuste de curva que busque el menor error cuadrático para los diferentes casos. La curva a ajustar tiene que ser igual a la transformada inversa de Laplace de la respuesta al escalón de una función como la vista en 24, es decir una función con

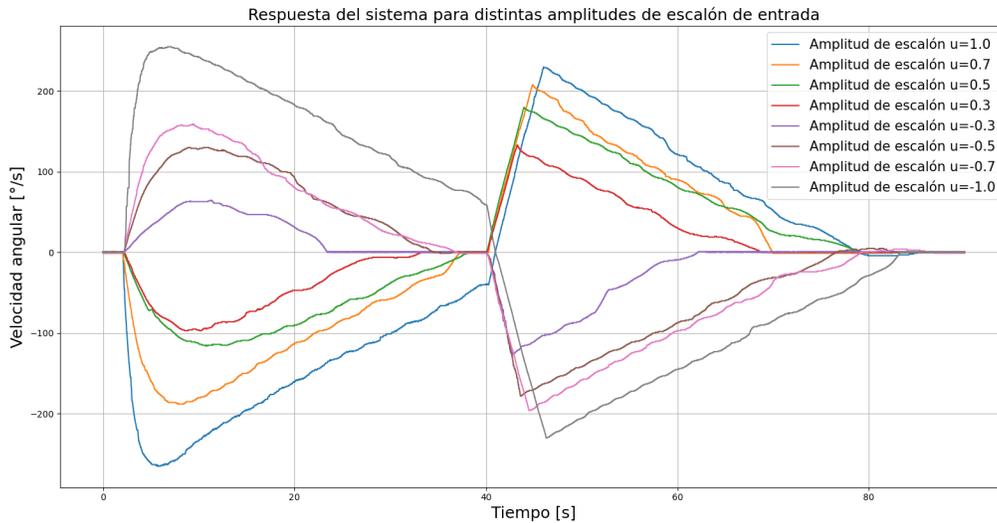


Figura 30: Comparación de gráficos temporales de distintos experimentos

tres polos:

$$\mathcal{L}^{-1}\left(\frac{Ks}{(s+a1)(s+a2)(s+a3)}\frac{1}{s}\right) = K\left(\frac{e^{-a1t}}{(a1-a2)(a1-a3)} - \frac{e^{-a2t}}{(a1-a2)(a2-a3)} - \frac{e^{-a3t}}{(a1-a3)(a3-a2)}\right) \tag{29}$$

Se utilizaron datos obtenidos en diversas mediciones, en las cuales se introdujo una entrada de tipo escalón de igual duración temporal y distinta amplitud. Se normalizó la escala para obtener amplitudes de salida similares entre todas las mediciones, además de mantener un signo positivo en todos los datos. Sólo se utilizaron los datos de los 40 segundos correspondientes al periodo durante el cual se realizó la excitación con la entrada de tipo escalón, es decir que la dinámica no lineal correspondiente al momento en que se deja de aplicar corriente no se modelará. Utilizando la herramienta *Curve Fitting Tool* de MatLab [20], se realizó el ajuste a una curva temporal de la forma obtenida en 29, logrando los siguientes resultados, los cuales se grafican en la figura 31:

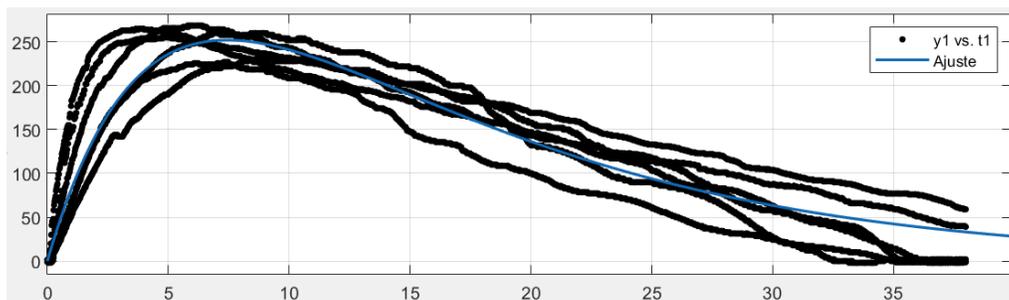


Figura 31: Gráfico de ajuste de función con tres polos

Variable	Valor obtenido	Intervalo de confianza 95 %
K	$2,633 \times 10^6$	$(-2,782 \times 10^9, 2,782 \times 10^9)$
a1	0.2028	(0.1953,0.2104)
a2	0.08344	(0.08125,0.08563)
a3	$2,769 \times 10^4$	$(-2,925 \times 10^7, 2,925 \times 10^7)$

Error cuadrático medio	29.0678
Coficiente de determinación	0.8687

Se observa que uno de los polos obtenidos tiene una dinámica mucho más rápida que la dinámica de los dos restantes. Se toma la hipótesis de que ese polo se corresponde con la dinámica de corriente, la cuál es mucho más rápida que las dinámicas de la inercia del cubo y de la rueda de reacción. Esto es un problema debido a que la frecuencia de muestreo utilizada es de 20ms, es decir 50Hz, mientras que el polo obtenido tiene una frecuencia en el orden de los 4400Hz ($27690rad/s * \frac{1rev}{2\pi rad}$). No se puede decir que la identificación se realizó con confianza, debido a que la frecuencia de muestreo no es lo suficientemente alta como para tomar mediciones representativas de la alta frecuencia de la planta.

Se propuso utilizar un modelo simplificado, de manera similar a como se realizó anteriormente para identificar la dinámica del actuador, tomando en este caso dos polos:

$$\mathcal{L}^{-1}\left(\frac{Ks}{(s+a1)(s+a2)}\frac{1}{s}\right) = K\left(\frac{e^{-a2t}}{(a1-a2)} - \frac{e^{-a1t}}{(a1-a2)}\right) \tag{30}$$

Realizando el ajuste de curva según la función hallada en 30, se obtuvieron los siguientes resultados, los cuales se grafican en la figura 32:

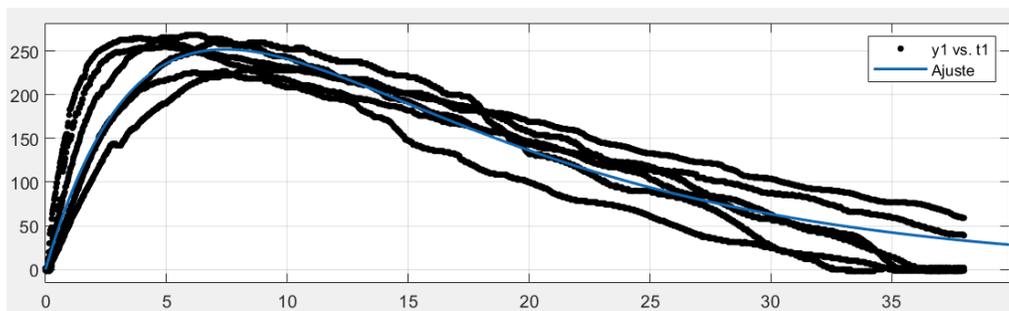


Figura 32: Gráfico de ajuste de función con dos polos

Variable	Valor obtenido	Intervalo de confianza 95 %	Error cuadrático medio	29.0663
K	95.11	(93.91,96.3)	Coefficiente de determinación	0.8687
a1	0.2028	(0.1968,0.2088)		
a2	0.08344	(0.08146,0.08542)		

Las dinámicas lentas mantuvieron el mismo orden de magnitud, mientras que la ganancia muestra una similitud con la ganancia obtenida anteriormente, según se ve a continuación:

$$\frac{2633000}{(s+0,083)(s+0,203)(s+27690)} \simeq \frac{95}{(s+0,083)(s+0,203)\left(\frac{s}{27690}+1\right)}$$

Además, el error se mantiene similar y el gráfico visual es idéntico, por lo que al eliminar un polo de la función objetivo no se está eliminando ninguna dinámica apreciable para nuestra frecuencia de observación. El modelo hallado, luego de agregar el signo negativo, es el siguiente:

$$\frac{\omega_C}{U} = \frac{-95s}{(s+0,083)(s+0,203)} \tag{31}$$

4.3.2. Corriente

Para corroborar la hipótesis tomada al simplificar el modelo se analizaron las mediciones de la corriente en la respuesta al escalón. En la figura 33 se puede ver la respuesta de la corriente a una

entrada de escalón de amplitud 70 %, análoga a la vista en la figura 29. Se observa un pico marcado de corriente en el principio, y luego un suave decaimiento hasta una corriente de régimen estacionario.

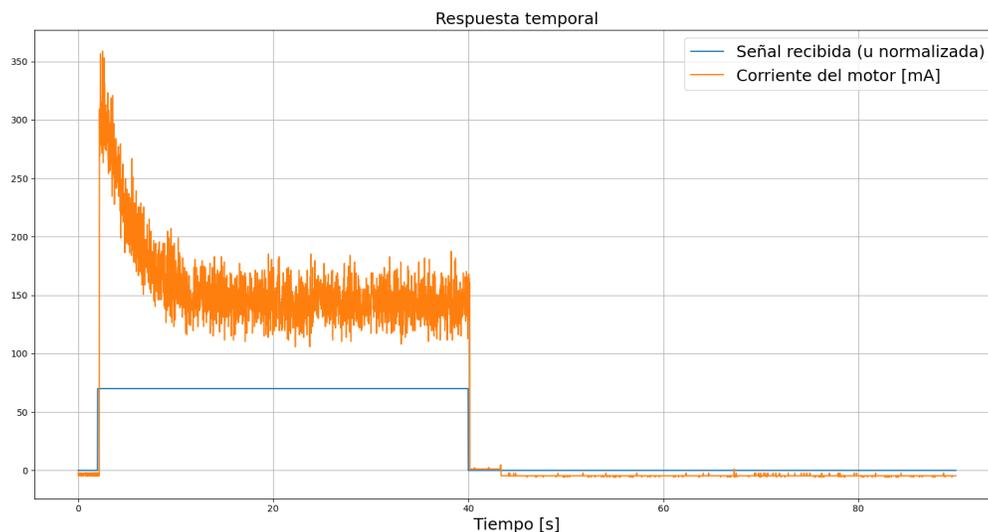


Figura 33: Gráfico temporal de respuesta de la corriente del motor

Esto se corresponde con lo esperado intuitivamente: un alto consumo de corriente en el momento del encendido del motor y luego una corriente de equilibrio cuando el motor llega a su velocidad máxima.

En la figura 34 se grafica la respuesta temporal de la corriente para diferentes valores del escalón de tensión de entrada. La respuesta es muy similar en todos los casos.

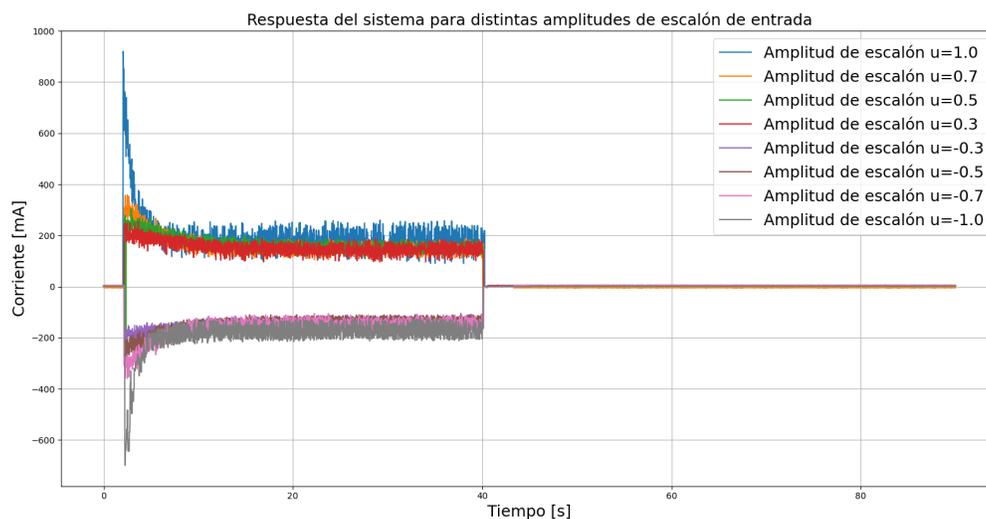


Figura 34: Comparación de gráficos temporales de distintos experimentos en la medición de corriente

Se propuso realizar un ajuste de curva utilizando las mismas herramientas que en el caso anterior, para una curva con la siguiente estructura, según 16:

$$\mathcal{L}^{-1}\left(\frac{K\left(\frac{s}{b_1} + 1\right)}{\left(\frac{s}{a_1} + 1\right)\left(\frac{s}{a_2} + 1\right)} * \frac{1}{s}\right) = K\left(\frac{a_1(a_2 - b_1)e^{-a_2t}}{b_1(a_1 - a_2)} - \frac{a_2(a_1 - b_1)e^{-a_1t}}{b_1(a_1 - a_2)} + 1\right) \quad (32)$$

Se obtuvieron los siguientes resultados para la función hallada en 32, los cuales se grafican en la figura 35:

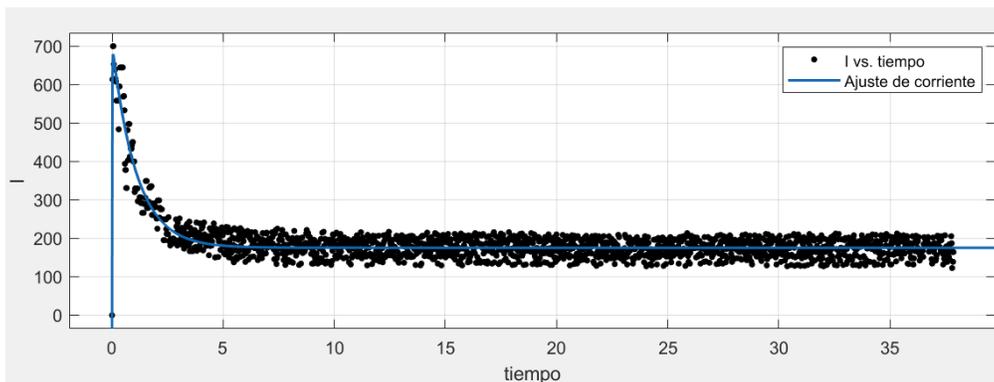


Figura 35: Gráfico de ajuste de función con dos polos y un cero para respuesta de corriente

Variable	Valor obtenido	Intervalo de confianza 95 %
K	174.7	(173.5, 176)
a1	0.9174	(0.8819, 0.9529)
a2	109.7	(77.16, 142.3)
b1	0.2283	(0.2213, 0.2353)

Error cuadrático medio	25.59
Coefficiente de determinación	0.8511

Según los datos obtenidos, el modelo hallado para la transferencia de la corriente es:

$$\frac{I}{U} = \frac{174,7(\frac{s}{0,228} + 1)}{(\frac{s}{109,7} + 1)(\frac{s}{0,917} + 1)} \tag{33}$$

Este modelo sirve para demostrar la validez de las ecuaciones matemáticas propuestas y la existencia de un polo de alta frecuencia en el bloque de corriente. Se observa que el polo dominante, de baja frecuencia, es 100 veces más lento que el polo de alta frecuencia. Dicho polo se encuentra dentro de los límites de la frecuencia de máxima medible sin incurrir en errores de aliasing, en este caso de $157rad/seg = 25Hz$, determinada por el teorema de Nyquist-Shannon. Sin embargo, debido a la degradación de las señales, en el caso de la dinámica de la velocidad angular del cubo el polo hallado se encontraba muy por encima de dicha frecuencia.

Además, se podría considerar para los fines prácticos que la respuesta es inmediata, ya que el pico de corriente se detecta en la muestra inmediato posterior a cuando se aplicó el escalón de tensión, lo cual elimina dicha dinámica para un observador que no es lo suficientemente rápido.

4.3.3. Velocidad angular del disco

Por último, se aprovecharon las mediciones realizadas para realizar una aproximación de la respuesta de la velocidad angular del disco frente al escalón de tensión. Si bien la función transferencia $\frac{\omega_D}{U}$ hallada en 20 es una función con dos polos, mediante inspección visual se puede ver que la respuesta al escalón observada en la figura 25 se asemeja a una función de primer orden, es decir de un solo polo. Esto puede deberse a que los polos tienen frecuencias con distintos ordenes de magnitud, y en la práctica solo es visible el efecto del polo dominante, es decir el de frecuencia más baja. Se decidió realizar una identificación de la transferencia simplificada, es decir proponiendo una función

de primer orden. La respuesta al escalón de una función de primer orden tiene la siguiente expresión, donde K es la ganancia y τ es la constante de tiempo:

$$y(t) = K(1 - e^{-\frac{t}{\tau}}) \quad (34)$$

Se propuso realizar una adaptación numérica de una curva con la forma vista en la ecuación 34, donde los parámetros libres de ganancia y constante de tiempo se ajustan a los datos de modo que se obtenga el menor error cuadrático medio. Se utilizaron mediciones para distintos valores de escalón de entrada, normalizándolos mediante una división por dicho valor enviado, de manera que se adquirió una conglomeración de puntos alrededor de la ganancia del sistema. Utilizando herramientas numéricas de Python se obtuvo como resultado $K \cong 3660$ y $\tau \cong 2,188$, lo cual se puede ver en la figura 36.

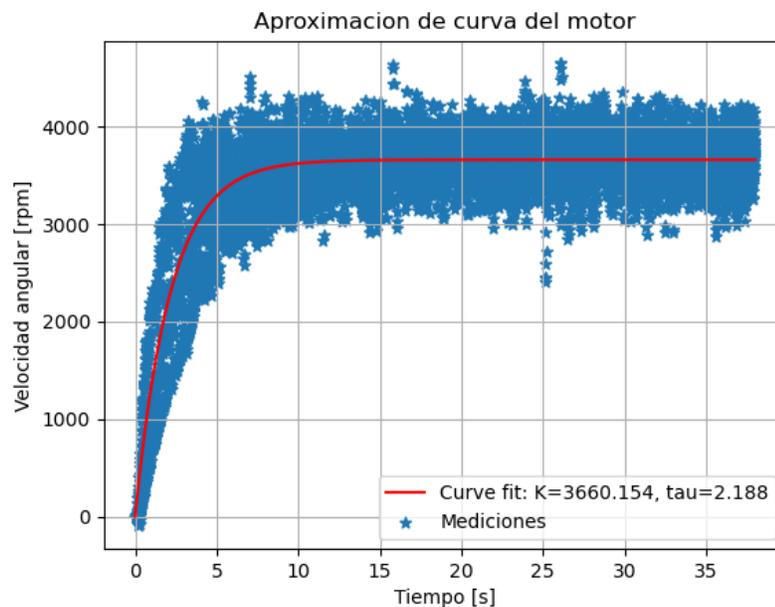


Figura 36: Ajuste de curva para respuesta del motor

Con lo cual la función transferencia resulta igual a:

$$\frac{\omega_D}{U} = \frac{3660}{(2,188s + 1)} = \frac{1672}{(s + 0,457)} \quad (35)$$

4.4. Validación del modelo

La validación de un modelo matemático identificado responde a las preguntas de si el modelo concuerda suficientemente bien con los datos observados, si sirve para el propósito que se necesita y si describe correctamente el sistema real [18]. En el caso del modelo identificado en este proyecto, el propósito es tener una base para comenzar el diseño de los lazos de control, por lo que no es necesario que la predicción del modelo sea muy rigurosa. Se optó por realizar una comparación de consistencia entre la salida modelada y la salida medida para señales de entrada de un nuevo conjunto de mediciones, realizando una comparación meramente visual para validar el modelo, con análisis del porcentaje de ajuste de los datos a partir del error cuadrático medio normalizado (NMRSE).

Se propuso realizar un experimento utilizando una señal de tipo chirp para comprobar la validez del modelo. La señal de tipo chirp es una señal senoidal de frecuencia variable, cuyos parámetros son: duración, frecuencia inicial, frecuencia final, método de barrido y amplitud. Como frecuencia mínima se eligió 0.01 rad/seg y como frecuencia máxima se eligió 0.8 rad/seg, utilizando un barrido lineal. El tiempo total del experimento fue de 10 minutos, por lo que las frecuencias se repartieron linealmente durante ese periodo. La amplitud de trabajo se eligió entre 1 y -1 de señal de control normalizada, es decir entre el máximo y el mínimo. En la figura 37 se puede ver la señal utilizada. Es una señal rica en frecuencias dentro de un rango de interés, por lo que mediante el análisis de la respuesta de la planta a una señal de este tipo se realiza un análisis más profundo en cuanto a la respuesta frecuencial.

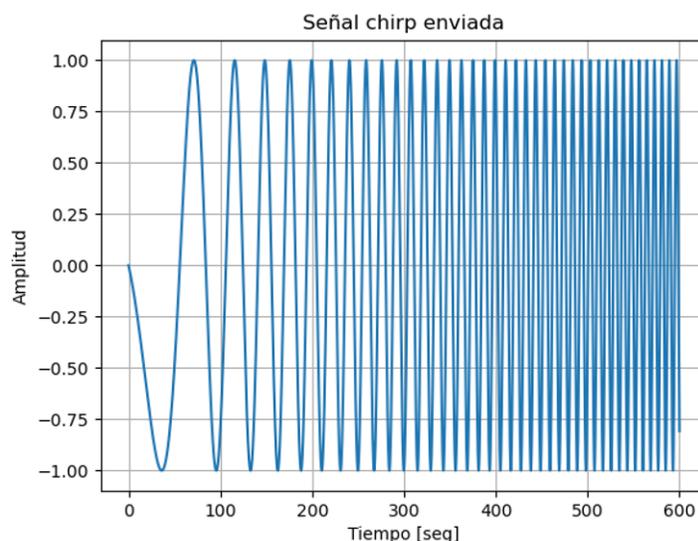


Figura 37: Señal de tipo chirp utilizada

La mínima resolución frecuencial que se puede lograr utilizando la ventana temporal elegida es de $1/600\text{seg} = 0,00167\text{Hz} = 0,01\text{rad/seg}$. Por otro lado, la máxima frecuencia que se puede reconstruir está dada por la frecuencia de muestreo, que en este caso es de 50Hz, por lo que la frecuencia máxima es de $25\text{Hz} = 157\text{ rad/seg}$. Es decir que la ventana de análisis utilizando técnicas de Transformada Rápida de Fourier (FFT) será entre 0,01 y 157 rad/seg. Sin embargo, nuestras frecuencias de interés son entre 0.2 y 0.08 rad/seg, por lo que la validación se observará principalmente en ese rango.

Una vez tomadas las mediciones, se realizó la FFT en la señal enviada y en la señal medida, obteniendo la transferencia de manera experimental al realizar la división de la Transformada de la salida sobre la Transformada de la entrada. Los puntos de medición disponibles son $600\text{seg}/0,02\text{seg} = 30000$, por lo que se usó una transformada de $2^{15} = 32678$ puntos, para poder aprovechar mejor las capacidades del algoritmo de FFT.

El módulo de la respuesta en frecuencia de la señal Chirp enviada es, idealmente, distinta de cero entre la frecuencia mínima (0.02 rad/seg) y la frecuencia máxima (0.8 rad/seg), y de amplitud cero en el resto de las frecuencias. Esto significa que al realizar la división entre la Transformada de Fourier de la señal medida y la Transformada de Fourier de la señal de entrada chirp fuera del rango de frecuencias de la chirp, se estará realizando una división por cero, o muy cercana a cero, introduciendo gran error numérico. En la figura 38 se puede ver el módulo del espectro de frecuencia de la señal utilizada, el cual se ve que es de amplitud muy baja fuera del rango de frecuencias de interés.

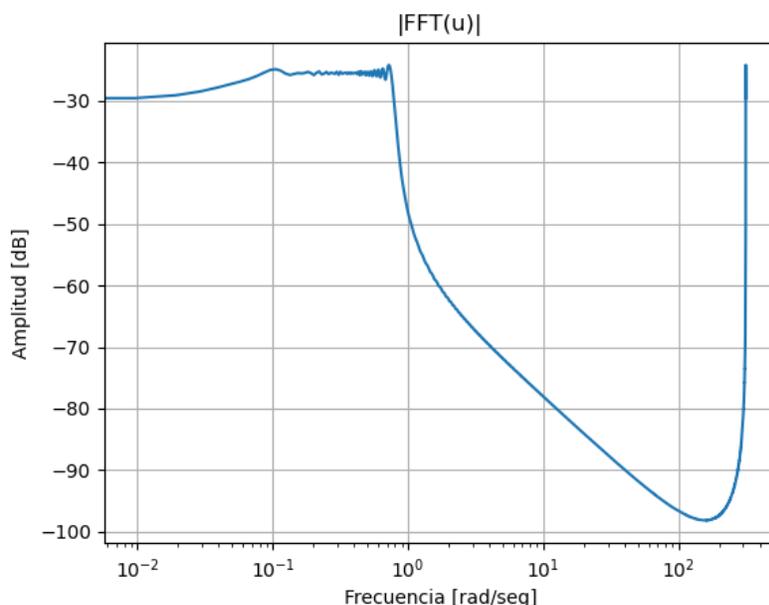


Figura 38: Amplitud del espectro frecuencial de la señal chirp utilizada

Para poder comparar los resultados medidos con los resultados esperados se realizó un modelo lineal de simulación utilizando la función transferencia hallada anteriormente. Luego, utilizando ese modelo, se halló la magnitud y el ángulo de la respuesta en frecuencia utilizando la herramienta bode del módulo *scipy.signal* [25]. Además, se realizó una simulación lineal de la respuesta utilizando la herramienta *lsim*, también del módulo *scipy.signal*. A dicha respuesta temporal se le calculó la FFT, y se realizó el mismo proceso de aproximación de diagrama de Bode que con los datos de la respuesta temporal medida en el experimento. De esta manera se puede comparar los resultados medidos con los resultados ideales, es decir el diagrama de Bode del modelo, y los resultados esperables del experimento, es decir la respuesta simulada. En la figura 39 se pueden ver los gráficos obtenidos de magnitud y fase.

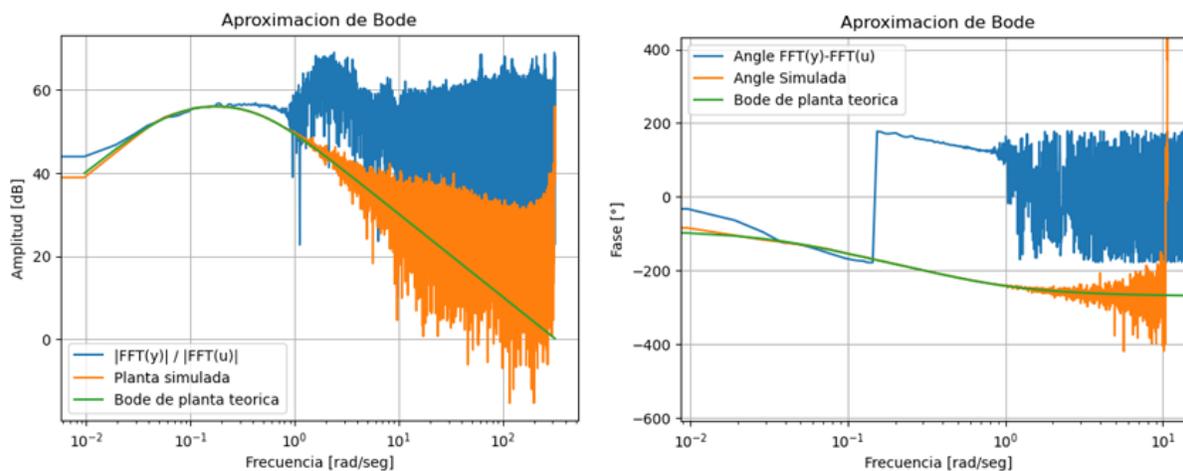


Figura 39: Resultados frecuenciales obtenidos en el experimento con señal chirp

Se puede ver que dentro de las frecuencias de interés tanto la magnitud como la fase de la señal

medida se asemejan a los valores esperados según el modelo propuesto. El salto que se observa en el gráfico de fase corresponde a un cambio de 360° , es decir que el ángulo es equivalente.

se analizó la respuesta temporal medida con respecto a la respuesta temporal simulada, tomando el porcentaje de ajuste a partir del error cuadrático medio normalizado, de la siguiente manera, donde y son los datos de la medición mientras que \hat{y} es la simulación del modelo:

$$Ajuste \% = 100 \left(1 - \frac{\|y - \hat{y}\|}{\|y - \text{mean}(y)\|} \right) \quad (36)$$

Se obtuvo un porcentaje de ajuste de 74.356%, como se observa en la figura 40. Si bien hay una pequeña diferencia de amortiguamiento, la dinámica es muy similar.

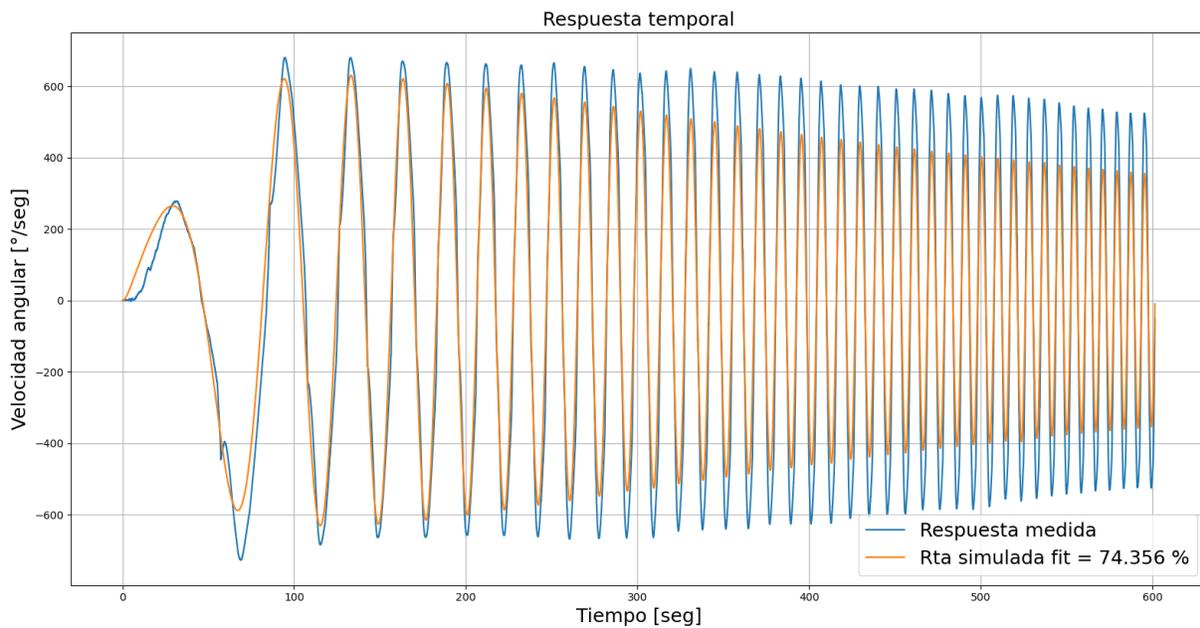


Figura 40: Resultados temporales obtenidos en el experimento con señal chirp

Para completar la validación del modelo se propuso realizar una medición utilizando como entrada una señal de tipo secuencia binaria aleatoria (RBS). Este formato de señal, al igual que la señal tipo chirp, introduce una gran cantidad de frecuencias y excita principalmente en la zona de las frecuencias de interés. Sin embargo, tiene la particularidad de que permite afectar mejor a las frecuencias utilizando una menor ventana de tiempo. La intención es generar una señal con gran densidad espectral de potencia en las frecuencias de interés de la planta. En la figura 41 se puede ver un ejemplo de una realización de señal de secuencia aleatoria binaria como la utilizada en el experimento.



Figura 41: Señal de tipo RBS utilizada

Los parámetros de la señal RBS son la duración mínima de pulso (T_c), la duración total del ensayo (T_t), la probabilidad de cambio (p) y la amplitud de pulso (ΔP). La duración mínima de pulso se define en función de la dinámica más rápida esperada del sistema (τ_{min}) de manera que se excite una zona de frecuencias cercanas. En este caso la frecuencia máxima es de 0.2rad/seg, por lo que se utilizó una duración mínima de pulso $T_c = 1seg$. La duración total del ensayo, por otro lado, depende principalmente de la constante de tiempo más lenta del sistema (τ_{max}), de manera que se pueda observar hasta las frecuencias cercanas a la misma. Además, se relaciona el tiempo total del ensayo y con la duración mínima de pulso a través del número de elementos total del ensayo. En este caso la frecuencia mínima es de 0.08rad/seg, con lo cual la constante de tiempo es 12.5seg, y se utilizó duración total de 45seg, siendo más de tres veces mayor. Por lo tanto, la cantidad de elementos del experimento es $n = 45$.

La probabilidad de cambio se define según cuál sea la frecuencia de mayor interés. Si la probabilidad de cambio es baja es más probable que se produzcan pulsos de unos o ceros consecutivos da mayor duración, por lo que la energía se concentrará en las bajas frecuencias, mientras que si la probabilidad de cambio es alta los cambios serán más frecuentes y la energía se concentrará en las altas frecuencias. Se definió utilizar una probabilidad $p=0.5$ para obtener una distribución pareja de la energía en la zona de las frecuencias de interés. Por último, la amplitud de pulso depende de la zona de trabajo que se busca caracterizar, siendo elegida una amplitud de 0.8 y -0.8 para el valor de entrada u , de manera de no llegar a la saturación, pero sí excitar al sistema en la zona de trabajo que deseamos.

Se realizó el mismo procedimiento que en el caso anterior, es decir, se tomó la FFT de la señal medida sobre la FFT de la señal de entrada, y se comparó contra la amplitud y fase teóricos de la planta y contra la amplitud y fase de la FFT de una respuesta simulada sobre la FFT de la entrada. Los resultados se grafican en la figura 42. Se debe considerar que al realizar un experimento de menor duración se pierde resolución en las frecuencias más bajas. Sin embargo, la finalidad de este experimento no fue identificar los parámetros sino validar los obtenidos anteriormente.

En la figura 43 se puede ver un gráfico temporal de la respuesta simulada y la respuesta medida. Se observa un comportamiento similar en ambos casos, con un porcentaje de ajuste de 68.954 %.

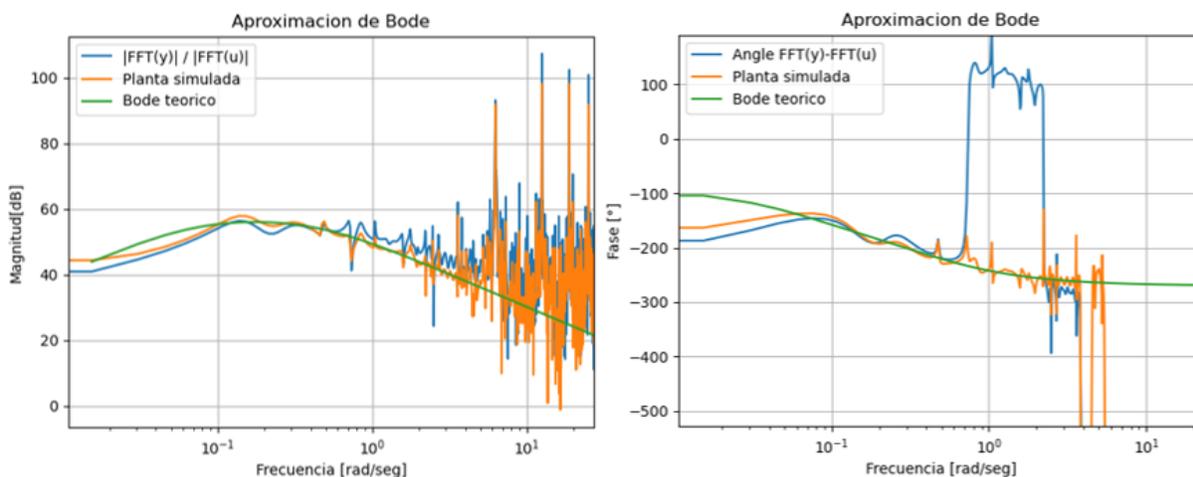


Figura 42: Resultados frecuenciales obtenidos en el experimento con señal RBS

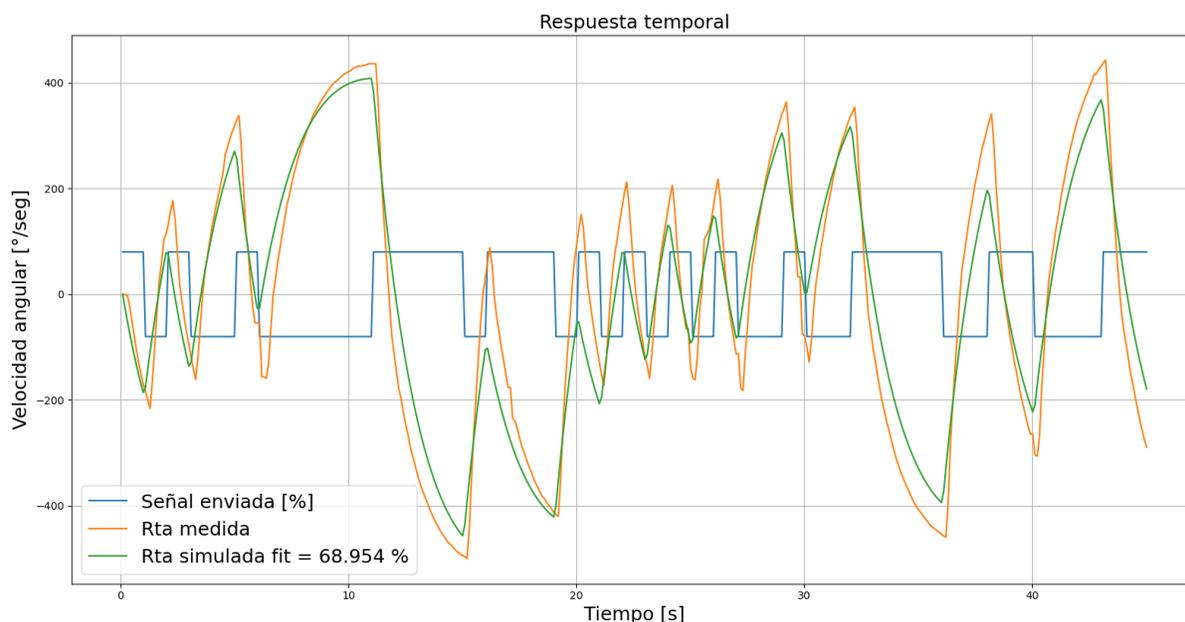


Figura 43: Resultados temporales obtenidos en el experimento con señal RBS

Se concluyó que el modelo hallado es válido, con un cierto error que se considera despreciable debido a la robustez que se logra al cerrar el lazo de control. Es decir que cuando el lazo de control se cierre, la sensibilidad del sistema realimentado frente a algún cambio en los parámetros de la planta será muy chica, por lo que la dinámica no se modificará en gran medida. La sensibilidad se define como la relación entre la tasa de cambio de la función de lazo cerrado a partir de un cambio en algún parámetro del modelo [12], y se expresa mediante la siguiente fórmula:

$$T_{cl} = \frac{DG}{(1 + DG)}$$

$$\delta T_{cl} = \frac{dT_{cl}}{dG} \delta G$$

$$S_G^{T_{cl}} = \text{Sensibilidad de } T_{cl} \text{ con respecto a } G$$

$$S_G^{T_{cl}} = \frac{\left(\frac{\delta T_{cl}}{T_{cl}}\right)}{\left(\frac{\delta G}{G}\right)} = \frac{dT_{cl} G}{dG T_{cl}} = \frac{1}{1 + DG}$$

4.5. Identificación del sistema por espacio de estados

La identificación del sistema en su descripción por espacio de estados se realizó utilizando un nuevo conjunto de mediciones. Sin embargo, se utilizó la información obtenida previamente para diseñar el experimento y para validar el modelo hallado. Se propuso realizar la identificación utilizando un modelo de caja gris, en el cual la estructura del modelo se mantiene fija en algunos parámetros y solamente habrá libertad en otros, de manera que la representación en espacio de estados sea lo más similar a la hallada analíticamente en 28, la cual recordamos a continuación:

$$\begin{bmatrix} \dot{i} \\ \dot{\omega}_D \\ \dot{\omega}_C \\ \dot{\theta}_C \end{bmatrix} = \begin{bmatrix} -\frac{R}{L} & -\frac{K_V}{L} & 0 & 0 \\ \frac{K_I}{J_D} & -\frac{\Gamma_D}{J_D} & 0 & 0 \\ -\frac{K_I}{J_C} & \frac{\Gamma_D}{J_C} & -\frac{\Gamma_C}{J_C} & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} i \\ \omega_D \\ \omega_C \\ \theta_C \end{bmatrix} + \begin{bmatrix} \frac{1}{L} \\ 0 \\ 0 \\ 0 \end{bmatrix} u \quad (37)$$

$$y = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} i \\ \omega_D \\ \omega_C \\ \theta_C \end{bmatrix} + 0u$$

La señal de entrada utilizada en el experimento fue una señal de tipo chirp de frecuencia inicial 1rad/seg y frecuencia final 0.02rad/seg con barrido en escala logarítmica. Dicha señal se eligió debido a que tiene gran variedad de frecuencias y su barrido logarítmico permite que se haga énfasis de manera similar tanto en las altas frecuencias como en las bajas frecuencias. En la figura 44 se ve un ejemplo de la misma.

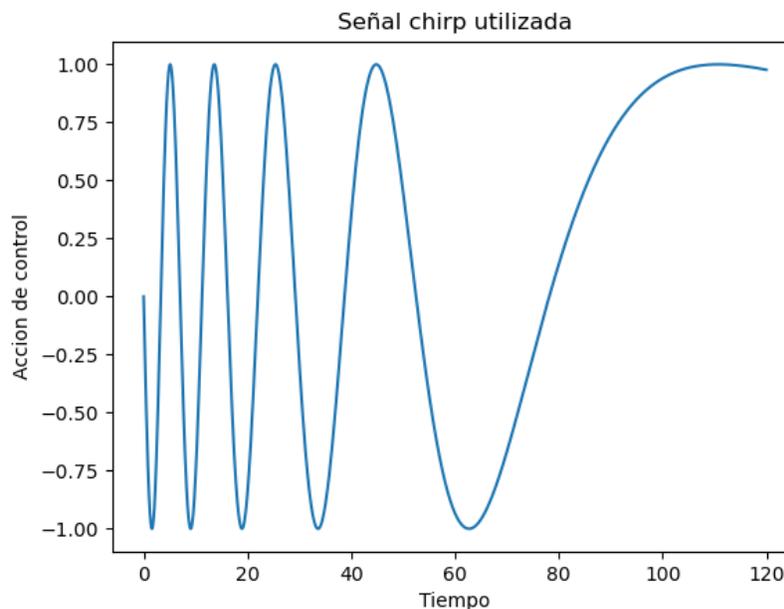


Figura 44: Señal utilizada para identificación de espacio de estados

El modelo de caja gris propuesto tiene la estructura vista en 28. De la misma manera que se hizo anteriormente, se decidió reducir el sistema en un estado, quitando el estado de posición angular del cubo θ_C , ya que el mismo corresponde simplemente a la integral de la velocidad angular del cubo ω_C . Esto se realiza para reducir las variables a calcular, y por ende la complejidad computacional. Se utilizaron restricciones estrictas para los elementos de valor cero de las matrices A y B, y restricciones de valor máximo o valor mínimo para los elementos restantes. Además, se quitó la libertad al elemento $a_{3,2}$, ya que su valor es igual a $\frac{a_{2,2}a_{3,1}}{a_{2,1}}$, y al respetar esa igualdad se mantiene la relación de proporcionalidad entre los parámetros del modelo y se asegura la existencia del cero en el origen para la transferencia de $\frac{\omega_C}{U}$ y $\frac{\Theta_C}{U}$. Por último, la matriz C se definió como matriz identidad, ya que las mediciones son exactamente los estados internos del sistema, y la matriz D se definió como 0 debido a que no hay término de feed-forward. En consecuencia, el modelo propuesto tiene la siguiente estructura:

$$\begin{aligned}
 A &= \begin{bmatrix} < 0 & < 0 & 0 \\ > 0 & < 0 & 0 \\ < 0 & \frac{a_{2,2}a_{3,1}}{a_{2,1}} & < 0 \end{bmatrix} \\
 B &= \begin{bmatrix} > 0 \\ 0 \\ 0 \end{bmatrix} \\
 C &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \\
 D &= 0
 \end{aligned} \tag{38}$$

Utilizando la herramienta *System Identification Toolbox* de MatLab[20], se obtuvieron los siguientes resultados:

$$\begin{aligned}
 A &= \begin{bmatrix} -107,4 & -7,839 & 0 \\ 4,375 & -0,2694 & 0 \\ -0,3682 & 0,02267 & -0,02523 \end{bmatrix} \\
 B &= \begin{bmatrix} 54698 \\ 0 \\ 0 \end{bmatrix}
 \end{aligned} \tag{39}$$

Estado	Porcentaje de ajuste a los datos	Error cuadrático medio	120.8
I	65.6 %	Error final de predicción	864.5
ω_D	86.22 %		
ω_C	86.28 %		

En la figura 45 se muestra una comparación entre las mediciones y la simulación del modelo para la señal de entrada utilizada.

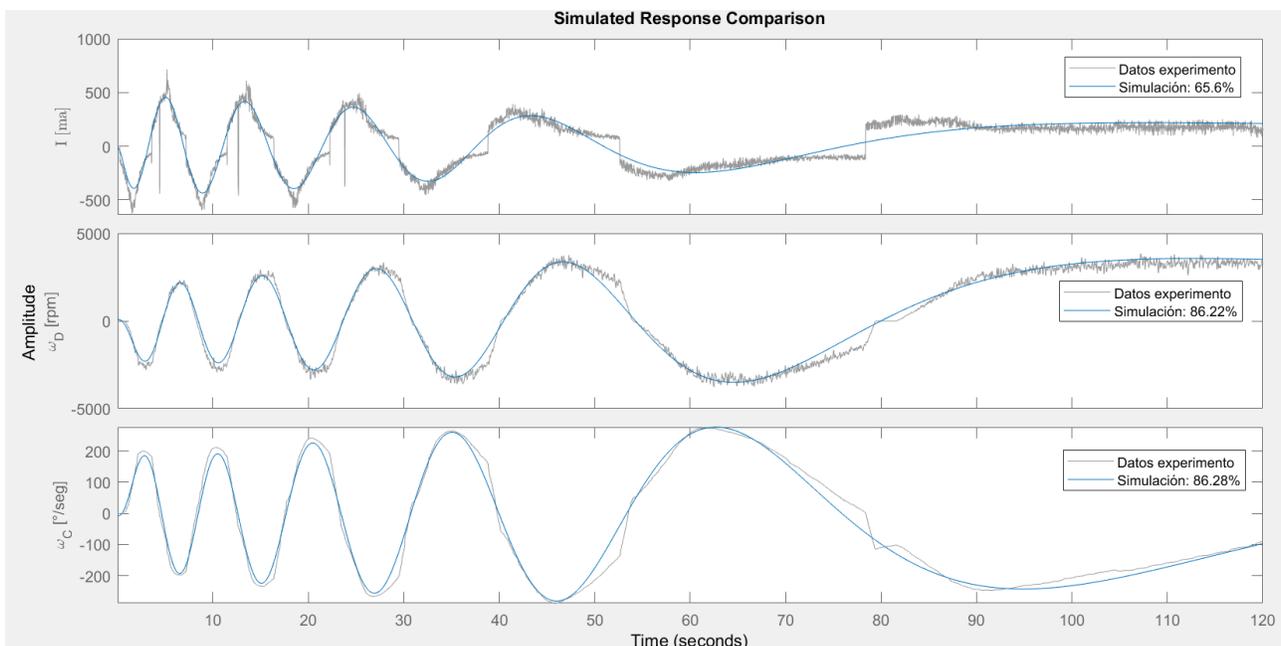


Figura 45: Comparación del modelo de estados

La validación de este modelo se realizó a partir de observar la respuesta al escalón simulada, la cual se muestra en la figura 46. Se puede comprobar que la respuesta simulada en las tres variables de estado es muy similar a la obtenida en los experimentos previos, mostrados en las figuras 32, 35 y 36, considerando los tiempos característicos de las curvas y los valores finales obtenidos.

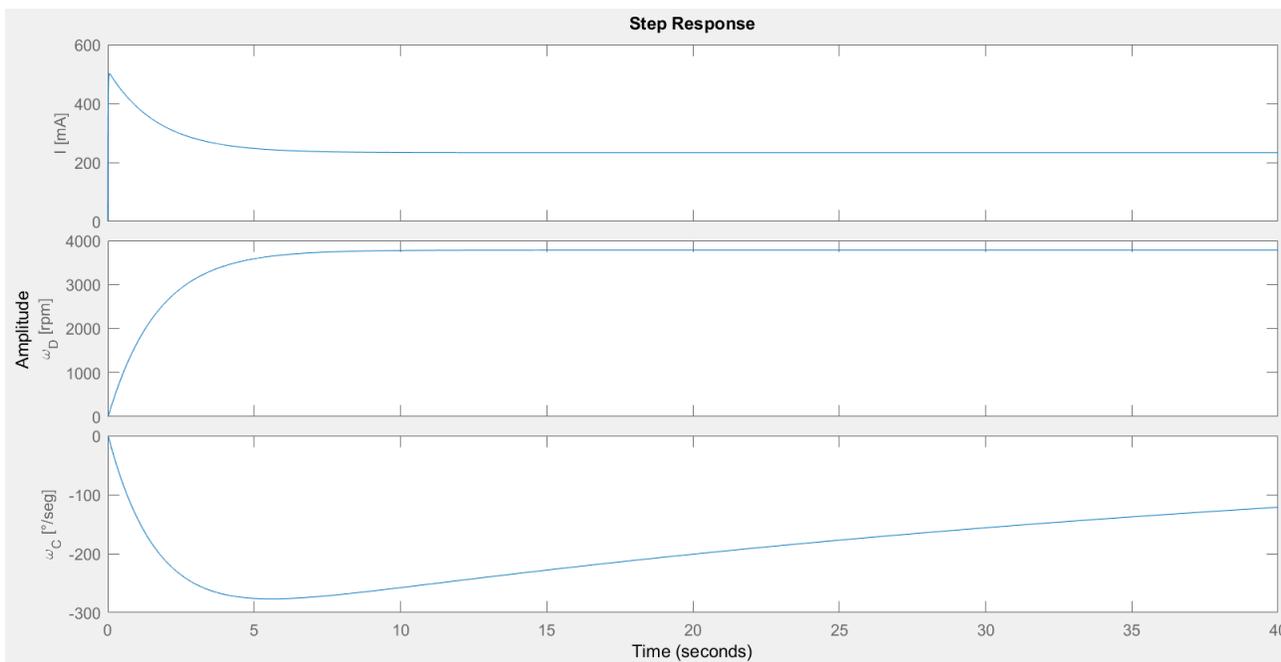


Figura 46: Simulación de la respuesta al escalón del modelo de estados

5. Diseño del control e implementación

En la siguiente sección se detallará el proceso de diseño realizado. Se comenzará por un análisis del problema a partir de técnicas de control clásico y propuestas de lazo de control simple, luego se pasará a propuestas de control mediante lazo interno y lazo externo, y por último se hará una propuesta de control realizando el análisis en función de los métodos de control moderno o por espacio de estados.

Se define utilizar como lineamiento para el diseño una frecuencia de corte deseada de 2.5rad/seg y un factor de amortiguamiento mayor a 0.5, buscando mantener los parámetros de los controladores dentro de un rango que sea comparable entre todos. Con dicha frecuencia de corte y factor de amortiguamiento se espera obtener un tiempo de establecimiento de aproximadamente 3seg, menor al tiempo requerido de 10seg.

5.1. Control mediante lazo simple

Recordando la función transferencia de la planta obtenida experimentalmente en 31, la respuesta de la posición angular del cubo en función de la entrada de control se puede representar de la siguiente manera:

$$\frac{\Theta_C}{U} = \frac{-95}{(s + 0,083)(s + 0,203)} \quad (40)$$

Se tiene por tanto una función transferencia con dos polos en baja frecuencia, y ganancia negativa. Se propone un esquema de control como el visto en la figura 47, en donde el error E es igual a la diferencia entre la referencia R y la salida Y . El controlador D tiene una función transferencia $D(s) = \frac{U(s)}{E(s)}$ que se debe hallar, a partir de la cual se calcula la acción de control U que se aplica en la planta G .

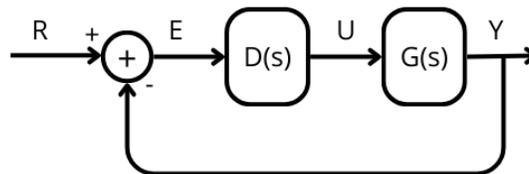


Figura 47: Esquema de control genérico para lazo externo

5.1.1. Controlador de adelanto de fase

La estabilidad de un sistema BIBO estable se define cuando para una entrada acotada en amplitud la salida del sistema también es acotada en amplitud. Dicha estabilidad se puede analizar en función de los polos de su función transferencia, de manera que si hay al menos un polo con parte real no negativa el sistema es inestable. El diseño del control se comenzó a partir de un análisis simple de estabilidad en el control proporcional. El sistema a lazo abierto hallado en 40 no posee polos en el semiplano derecho, por ende es estable y permite realizar las pruebas y mediciones que se propusieron anteriormente. Como se desea respetar la nomenclatura $E = R - Y$ de realimentación negativa, y debido a que la planta tiene ganancia de valor negativo, se optó por buscar una ganancia negativa para la función del controlador D , necesaria para lograr una realimentación negativa que asegure la estabilidad. Utilizando un gráfico del lugar de las raíces se pueden ver en dónde quedarán los polos

del sistema en lazo cerrado al agregar una ganancia. Según el gráfico de la figura 48 se observa que el sistema será estable para cualquier ganancia negativa.

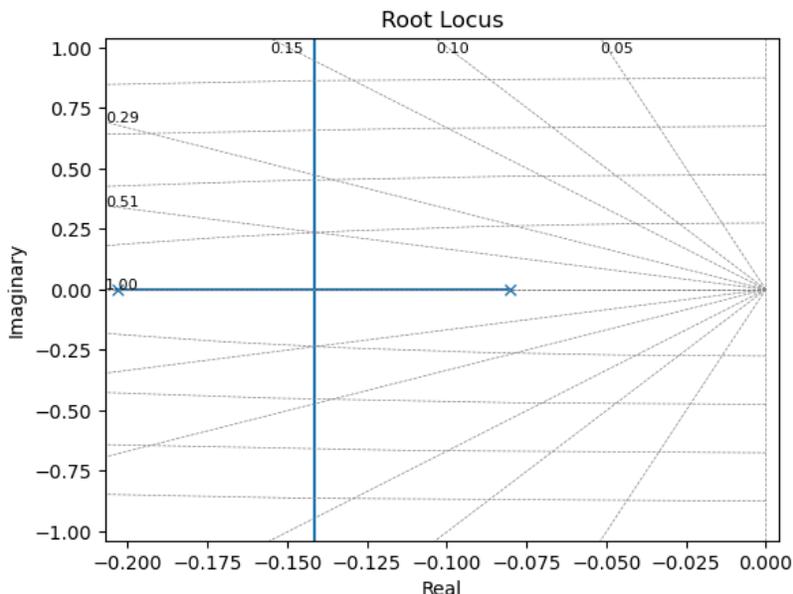


Figura 48: Gráfico de lugar de las raíces para ganancia negativa

Sin embargo, la estabilidad a lazo cerrado también puede estudiarse en función de la posibilidad de que la realimentación sea de ganancia 1 y de fase -180° . Esto quiere decir que la salida del sistema en lazo abierto será exactamente igual a la entrada pero con signo opuesto, y al cerrar el lazo con realimentación negativa se reintroduce la misma señal de entrada. Esto genera el caso en que para un referencia de valor nulo el sistema se realimente a sí mismo generando un bucle. El Margen de Fase determina cuánto cambio de fase puede soportar el sistema a lazo cerrado antes de volverse inestable, es decir, cuánta diferencia de fase hay en el sistema cuando la ganancia vale 1 para llegar a -180° . Cualquier cambio externo o interno del sistema que modifique la fase pero no la ganancia hará que el sistema se inestabilice.

De manera práctica, el Margen de Fase se encuentra en el momento que la magnitud en dB cruza el eje de 0dB, en la frecuencia conocida como frecuencia de corte. En la figura 49 se muestra un diagrama de Bode de amplitud y fase de la función de lazo con ganancias negativas. Se puede ver que el Margen de Fase (para una ganancia de valor -1) es muy pequeño. Para agrandar el Margen de Fase se debe colocar una ganancia de magnitud menor a 1, lo que disminuiría la frecuencia de corte llevandola a frecuencias en donde la fase es menos cercana a -180° . De esta manera se puede ver que, si bien la información vista en la figura 48 indica que el lazo puede cerrarse con un control proporcional, el sistema a lazo cerrado quedará muy cerca de ser inestable, y probablemente tenga dificultades para converger.

Por otro lado, al disminuir la frecuencia de corte se modifican la frecuencia fundamental del sistema en lazo cerrado y su factor de amortiguamiento. La función de lazo cerrado simplificada de

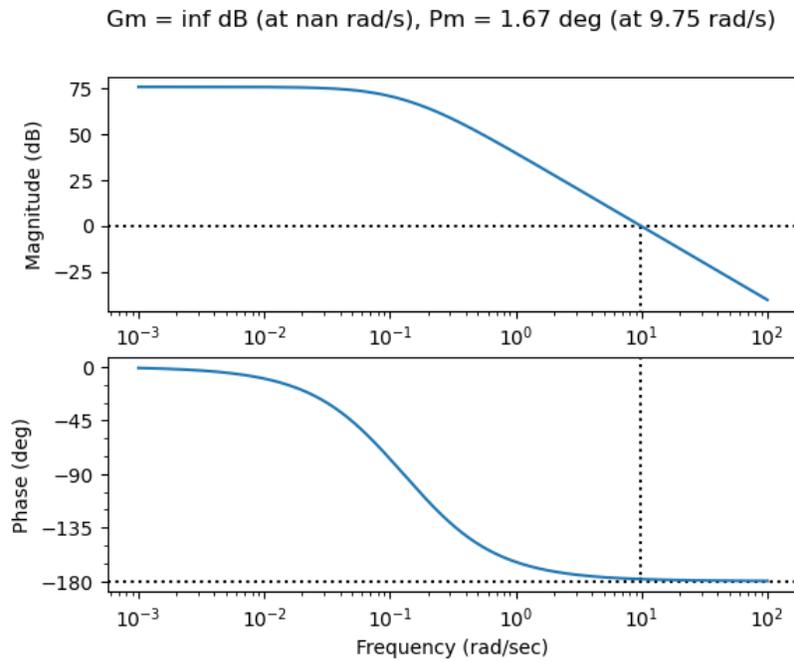


Figura 49: Diagrama de Bode de planta

segundo orden se define de la siguiente manera, según [12]:

$$H(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}$$

$$0 \leq \zeta \leq 1$$

$$\omega_n > 0$$

Donde ω_n es la frecuencia natural o fundamental del sistema y ζ es el factor de amortiguamiento. Ambos parámetros definen las características dinámicas que tendrá el sistema, como tiempo de establecimiento, tiempo de subida, valor del sobrepico, entre otros. La frecuencia de corte se relaciona con la frecuencia fundamental y el factor de amortiguamiento según:

$$\omega_c = \omega_n \sqrt{(1 - 2\zeta^2) + \sqrt{4\zeta^4 - 4\zeta^2 + 2}}$$

$$\omega_c \cong \omega_n \tag{41}$$

Además, el factor de amortiguamiento se relaciona con el Margen de Fase de la siguiente manera:

$$MF = \arctan \left(\frac{2\zeta}{\sqrt{\sqrt{1 + 4\zeta^4} - 2\zeta^2}} \right) \cong 100\zeta \tag{42}$$

Se propuso diseñar un controlador que cumpla con una frecuencia de corte de 2.5rad/seg y un Margen de Fase de 50° en dicha frecuencia. Para ello se propuso utilizar una ganancia que defina la frecuencia de corte, y luego mejorar el Margen de Fase mediante el agregado de una red de adelanto de fase. En la figura 50 se observa el diagrama de Bode del sistema con una ganancia de $K = \frac{-1}{15}$, lo cual da la frecuencia de corte deseada y un Margen de Fase de 6° en ese punto.

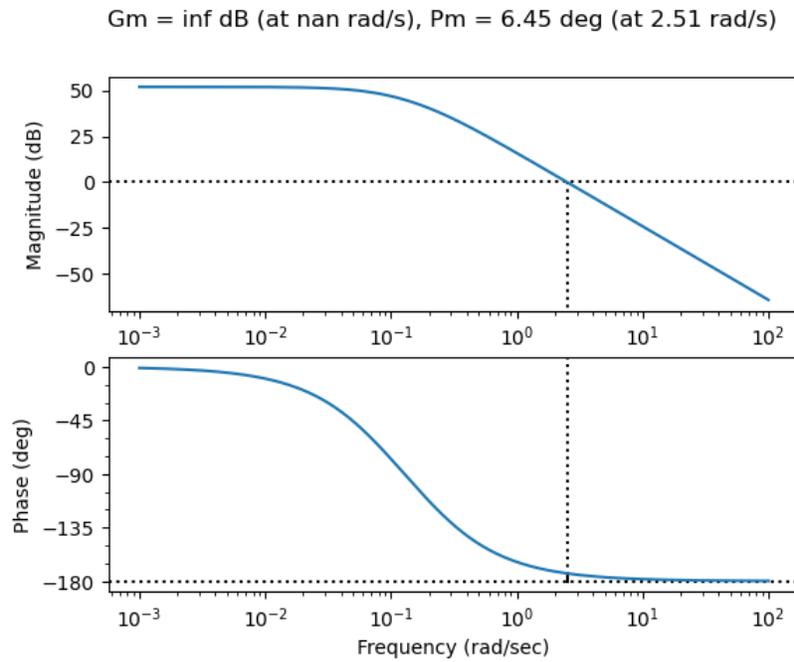


Figura 50: Diagrama de Bode de planta con ganancia $K=-1/15$

Luego, la red de adelanto de fase para un agregado de 45° se calcula mediante:

$$\begin{aligned}
 K &= -\frac{1}{15} & \Phi &= 45^\circ & \alpha &= \frac{(1 - \sin \Phi)}{(1 + \sin \Phi)} \\
 \omega_c &= 2,5 \text{ rad/seg} & T_d &= \frac{1}{\sqrt{\alpha} \omega_c} \\
 D_{lead}(s) &= K \frac{(1 + T_d s)}{1 + \alpha T_d s} \sqrt{\alpha} & & & & (43)
 \end{aligned}$$

La función transferencia que se halló para el controlador de adelanto de fase es:

$$D_{lead}(s) = -0,161 \frac{(s + 1,0352)}{(s + 6,035)} = \frac{-0,02667s - 0,02761}{0,1657s + 1} \quad (44)$$

En la figura 51 se puede ver el diagrama de Bode del sistema con el agregado de fase propuesto.

Para la implementación numérica, se realizó la conversión a variable discreta mediante el método de Tustin, el cual aproxima la relación entre la variable de la Transformada de Laplace (s) y la variable de la Transformada Zeta (z) a partir de una aproximación por cociente de polinomios de su relación exponencial inherente:

$$z \approx \frac{1 + sT/2}{1 - sT/2} \quad (45)$$

En donde T es el tiempo de muestreo de las señales, es decir 0.02 seg. A partir de aplicar dicha aproximación, se obtuvo la siguiente función en Transformada Zeta:

$$D_{lead}(z) = \frac{U}{E} = \frac{-0,1534 + 0,1502z^{-1}}{1 - 0,8862z^{-1}} \quad (46)$$

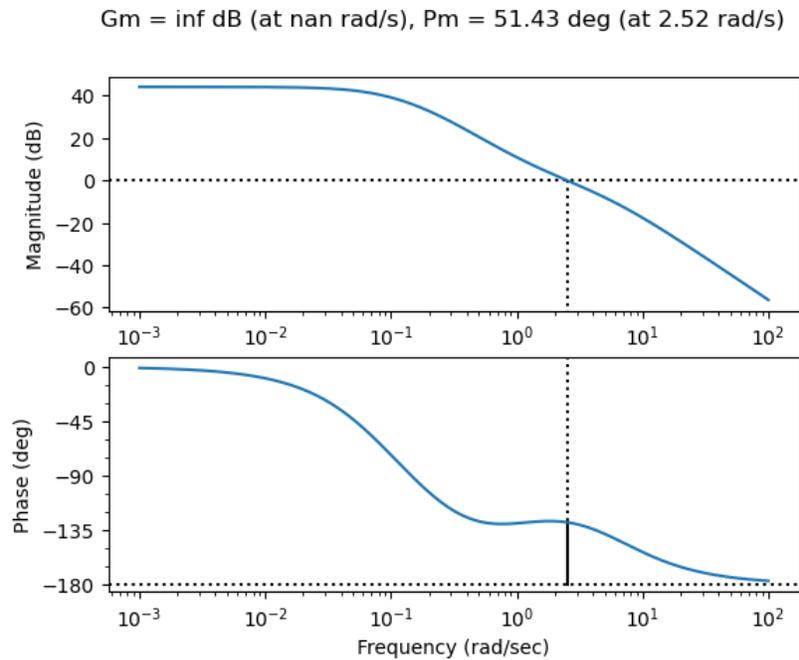


Figura 51: Diagrama de Bode de planta y controlador de adelanto de fase

A partir de los coeficientes de la función de controlador $D(z)$ y de las propiedades de la Transformada Zeta inversa, se obtiene el algoritmo de control que se aplica para obtener la variable de control u en cada instante de tiempo k . De manera genérica:

$$D(z) = \frac{U}{E} = \frac{a_0 + a_1 z^{-1}}{1 + b_1 z^{-1}} \quad (47)$$

$$u[k] = a_0 e[k] + a_1 e[k - 1] - b_1 u[k - 1] \quad (48)$$

Y para el caso del controlador de adelanto de fase:

$$u[k] = -0,1534e[k] + 0,1502e[k - 1] + 0,8862u[k - 1] \quad (49)$$

5.1.2. Controlador proporcional integrativo derivativo

Bajo otra perspectiva se propuso analizar el desempeño del sistema en estado estacionario. La ganancia del sistema para una entrada constante ($DG(0)$) influye en el error de estado estacionario. El sistema a lazo cerrado propuesto anteriormente es un sistema de tipo 0, es decir que no tiene ningún integrador, por lo que no será capaz de seguir referencias de tipo escalón sin error de estado estacionario. Dicho error es igual a:

$$e_{ss} = \frac{1}{1 + K_p} = \frac{1}{1 + DG(0)} \quad (50)$$

Es decir que cuanto mayor sea la ganancia para frecuencia cero, menor será el error de estado estacionario, con lo cual al disminuir la ganancia se logra un mejor Margen de Fase pero un peor error e_{ss} . En la figura 51 se puede ver que la ganancia en bajas frecuencias para el controlador de adelanto de fase tiende a ser superior a los 40dB. Al calcular K_p se obtiene 160, con lo cual el error e_{ss} es menor al 0.6 %.

Sin embargo, debido a que se desea controlar el sistema frente a referencias de posición de tipo escalón, es deseable que el sistema sea de tipo 1, es decir que no presente error de estado estacionario frente a referencias de escalón y sí lo presente frente a referencias de tipo rampa. Para ello es necesario agregar un integrador puro a la función de lazo, lo cual puede generar inestabilidad. En la figura 52 se puede ver un gráfico del lugar de las raíces para ganancias negativas con un integrador, y se observa que al agregar un nuevo polo existe la posibilidad de que al cerrar el lazo hayan polos con parte real positiva, lo que genera inestabilidad en el sistema. La menor ganancia para la cual el sistema es estable está dentro del orden de 5×10^{-5} , y al ser un valor tan pequeño se pueden introducir errores numéricos que lo modifiquen y le quiten robustez al sistema. Es decir que utilizando solamente un integrador el sistema tiene riesgo de volverse inestable.

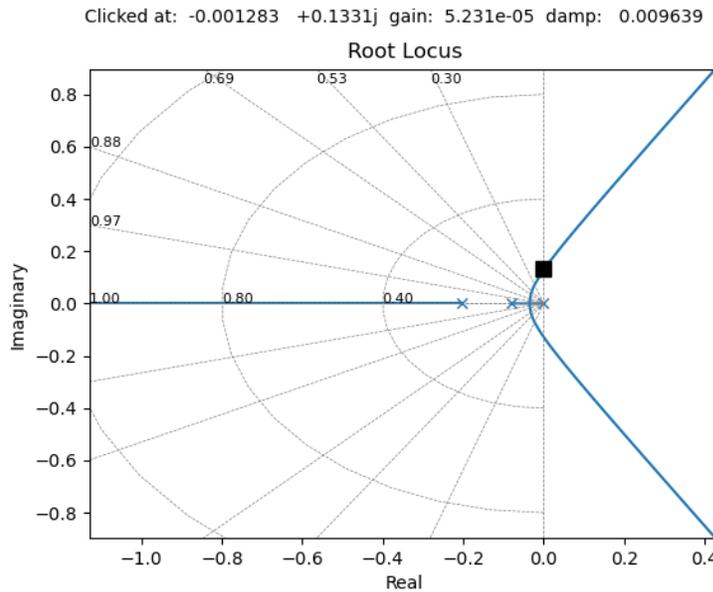


Figura 52: Gráfico de lugar de las raíces para sistema con integrador puro

Se propuso realizar un segundo esquema de control de tipo Proporcional Integrativo Derivativo (PID), como los propuestos en [28]. Con la acción integrativa se logra agregar el integrador puro que convierte al sistema en un sistema de tipo 1, mientras que la acción derivativa agrega una mayor respuesta del sistema frente a cambios bruscos de referencia, y la acción proporcional le da estabilidad general al lazo ayudando a eliminar los problemas inherentes tanto a la acción integral como a la acción derivativa. Se propuso un controlador con la siguiente estructura:

$$D_{PID}(s) = K_P \left(1 + \frac{1}{T_I} \frac{1}{s} + T_D s \right) \tag{51}$$

Cuya relación con el dominio temporal es:

$$u(t) = K_P \left(e(t) + \frac{1}{T_I} \int_0^t e(t) dt + T_D \dot{e}(t) \right) \tag{52}$$

En la figura 53 se observa un gráfico del lugar de las raíces para la ganancia K_P . Los valores se T_I y T_D fueron 20 y 0.05 respectivamente, obtenidos experimentalmente de manera que el gráfico del lugar de las raíces permita elegir polos cercanos a la frecuencia de 2.5rad/seg.

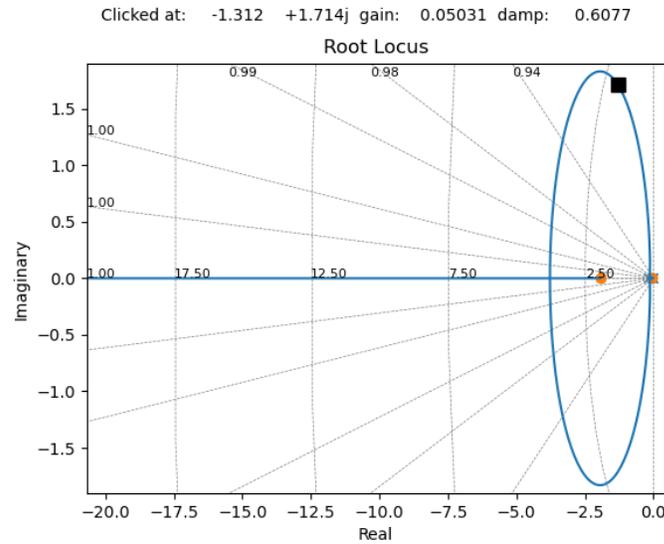


Figura 53: Gráfico de lugar de las raíces para sistema con control PID

Se propuso un valor de $K_P = -0,05$ con lo cual se obtuvo la siguiente función:

$$u(t) = -0,05 \left(e(t) + 0,05 \int_0^t e(t)dt + 0,5\dot{e}(t) \right) \tag{53}$$

En la figura 54 se observa un diagrama de Bode del sistema con el controlador PID propuesto. Se puede ver que la frecuencia de corte es de 2.8rad/seg y el Margen de Fase es de 60°, similar a lo obtenido con el controlador de adelanto de fase pero con ganancia infinita en frecuencia cero.

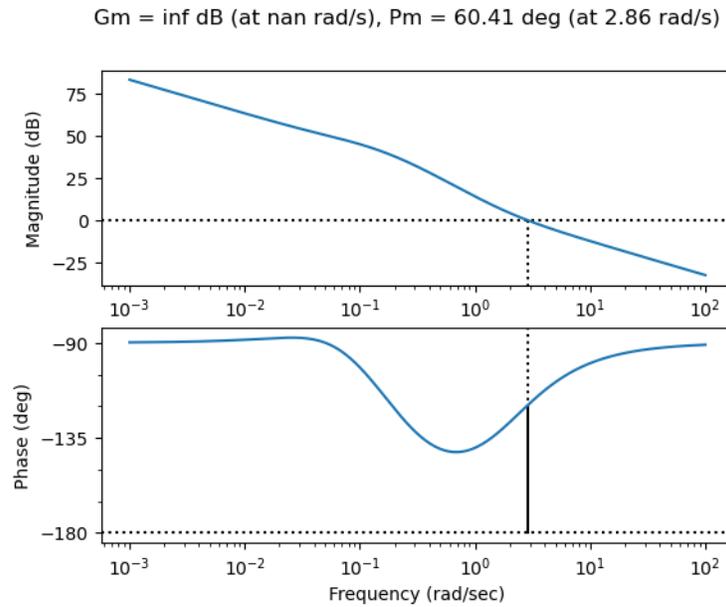


Figura 54: Diagrama de Bode de planta y controlador PID

En este caso para la conversión a tiempo discreto se utilizó la aproximación de la derivada por

diferencia finita:

$$\dot{x}(t) \approx \frac{x[k] - x[k-1]}{T_s} \quad (54)$$

Con lo cual el algoritmo para la implementación numérica es:

$$u[k] = \left(K_P + \frac{K_P T_s}{T_I} + \frac{K_P T_D}{T_s}\right)e[k] - \left(K_P + 2\frac{K_P T_D}{T_s}\right)e[k-1] + \frac{K_P T_D}{T_s}e[k-2] + u[k-1] \quad (55)$$

$$u[k] = -1,32e[k] + 2,5e[k-1] - 1,25e[k-2] + u[k-1]$$

5.2. Control mediante doble lazo

Se propone en este caso un control a partir de cerrar un lazo interno de alguna de las variables que se poseen en medición, y luego cerrar un lazo externo de la posición angular del cubo, de manera que se controlen las dinámicas internas de la planta.

Los controladores propuestos tendrán un esquema doble, en el cual para el diseño se analiza primero la sección del lazo interno, calculando un controlador D_i que cierre el lazo de control. A partir de tener el lazo interno cerrado se realiza un análisis de la planta con esa nueva dinámica controlada, y se calcula el controlador externo D_e , cuya salida será la referencia que toma el controlador D_i . No se incluyó dentro del alcance del diseño limitaciones a la escala máxima admisible de la variable del lazo interno.

Como criterio de diseño se considera que el lazo interno debe converger al menos 5 veces más rápido que el lazo externo, es decir que su ω_c sea 5 veces mayor. Además, se debe tener en cuenta que el diseño se realiza en el plano continuo, mientras que la implementación se realiza en el plano discreto. Para que la frecuencia de muestreo elegida sea efectiva en el seguimiento del algoritmo de control diseñado para un sistema continuo se toma la relación $\omega_c < F_s/25$. Es decir que en este caso: $\omega_c < 12,56\text{rad/seg}$, y tomando esa relación en el lazo interno se obtiene la restricción para el lazo externo de $\omega_c < 2,512\text{rad/seg}$.

Al proponer frecuencias de trabajo en lazo cerrado más cercanas a la frecuencia de muestreo se comienza a apreciar la acción de la discretización como un retardo. Los retardos temporales, al trasladarse al dominio de Laplace, se ven como una función exponencial del tipo: e^{-sT} donde T es el tiempo del retardo. Su efecto en el sistema consiste en un agregado de fase a medida que aumenta la frecuencia.

Para diseñar el lazo interno se considerará un retardo de medio tiempo de muestreo, es decir 0.01seg, a partir de su aproximación de Padé de primer orden:

$$e^{-sT} \approx \frac{1 - \frac{sT}{2}}{1 + \frac{sT}{2}}$$

5.2.1. Lazo interno de velocidad angular del disco

Como primer caso se plantea controlar la velocidad angular del disco en el lazo interno, siguiendo el esquema que se ve en la figura 55. A partir de tener un control en la velocidad que tenga una convergencia más veloz que el lazo externo se puede asegurar que se llegará al estado de equilibrio con el motor en velocidad cero en el estacionario.

Se realizó un análisis de la función de transferencia $\frac{\omega_D}{U}$ obtenida experimentalmente en 35, a partir del diagrama de Bode mostrado en la figura 56, en el cual se incluye el retardo mencionado.

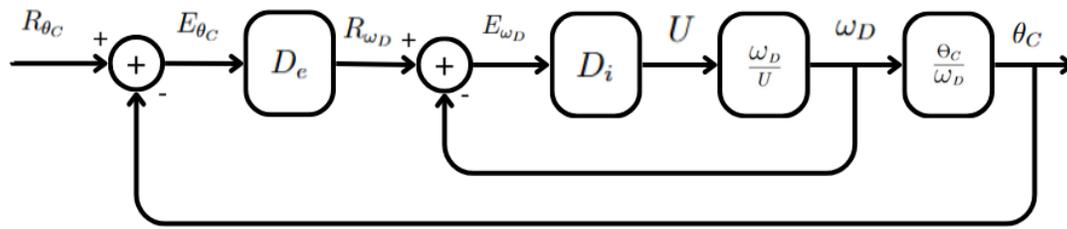


Figura 55: Esquema de control para lazo interno de velocidad angular del disco

Gm = -18.42 dB (at 200.46 rad/s), Pm = -76.34 deg (at 1672.00 rad/s)

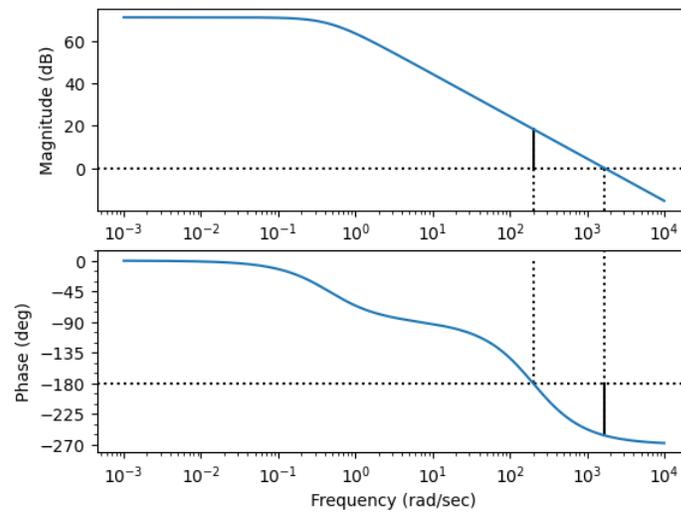


Figura 56: Diagrama de Bode para transferencia de velocidad angular del disco

La función transferencia posee un único polo, por lo que la fase de su respuesta en frecuencia sólo cambia de 0° a 90° , pero se ve cómo influye la acción del retardo a partir de los 40-50rad/seg. A partir de ver el diagrama se concluye que se puede lograr un buen controlador utilizando un control proporcional, con ganancia que lleve la frecuencia de corte a la zona deseada. En la figura 57 se ve el resultado de aplicar una ganancia de valor 0.0075, con la que se logra una frecuencia de corte de 12.5rad/seg y un Margen de Fase de 85° .

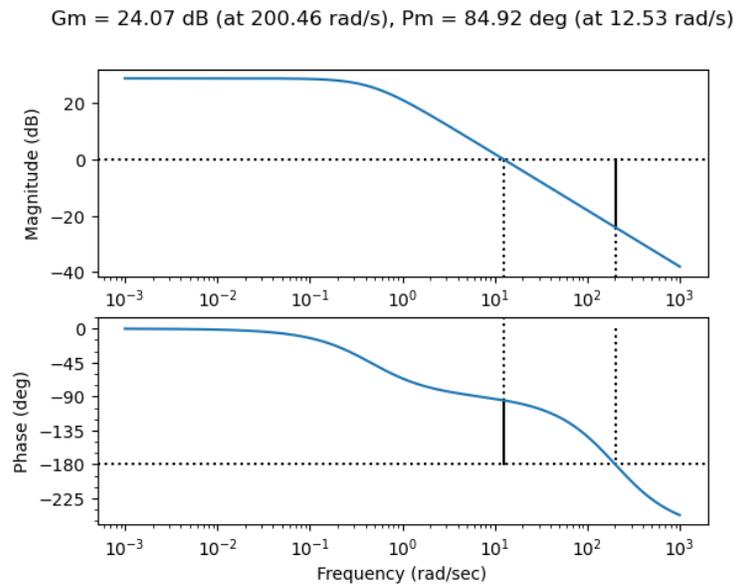


Figura 57: Diagrama de Bode para transferencia de velocidad angular del disco con ganancia

Es decir:

$$D_i = 0,0075 \tag{56}$$

Con ese lazo de control cerrado, se procedió a analizar la transferencia de la planta resultante, cuyo diagrama de Bode se puede ver en la figura 58. Al existir un signo negativo en la transferencia $\frac{\Theta_C}{\omega_D}$ la ganancia debe también ser de valor negativo.

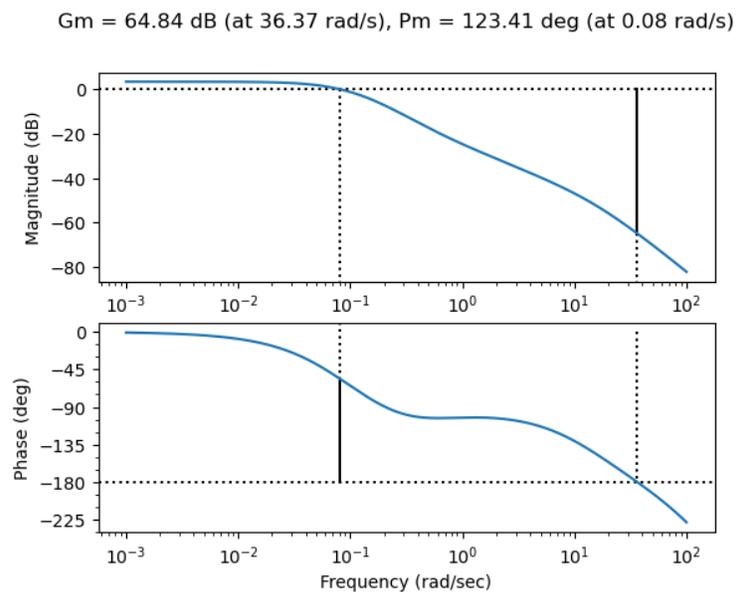


Figura 58: Diagrama de Bode para transferencia de posición con lazo interno de velocidad angular del disco

Se ve que aumentando la ganancia se puede llegar a una frecuencia de corte del rango deseado, mientras que el Margen de Fase será mayor a 50° hasta los 10rad/seg. Se propone una ganancia de

valor -45, con la que se logra una frecuencia de corte de 2.5rad/seg y Margen de Fase 75°, como se ve en la figura 59.

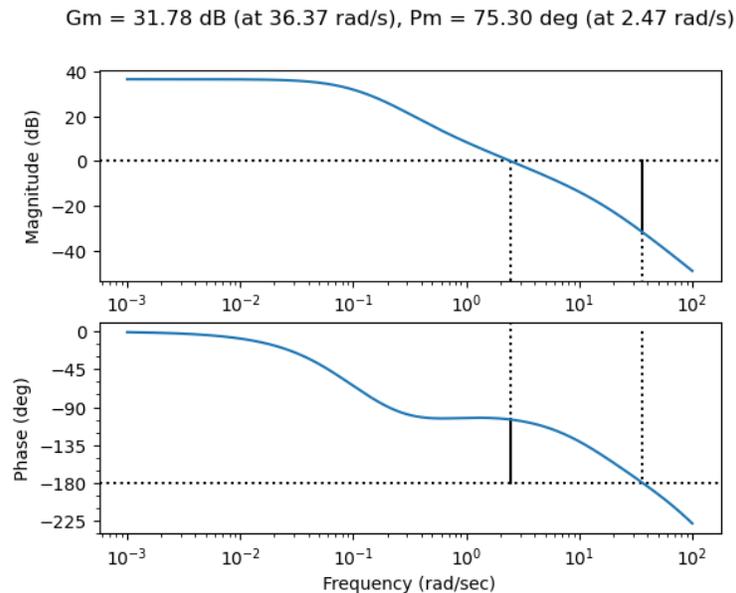


Figura 59: Diagrama de Bode para transferencia posición con ganancia y lazo interno de velocidad angular del disco

Es decir:

$$D_e = -45 \tag{57}$$

5.2.2. Lazo interno de corriente

A continuación, se plantea un esquema de control en donde se cierra un lazo interno que controle la corriente que circula en el motor, como el esquema mostrado en la figura 60. Se espera que al colocar realimentación en la corriente se tenga un control directo sobre el torque que se aplica con el motor, debido a su relación de proporcionalidad mostrada en 3. De esta manera se espera tener un rechazo frente a las perturbaciones de tipo torque que se pueden introducir al sistema, por ejemplo a modo de fuerzas de rozamiento no modeladas. Sin embargo, se debe considerar que no se puede rechazar perturbaciones de torque constante debido a que el actuador saturará en una velocidad angular máxima.

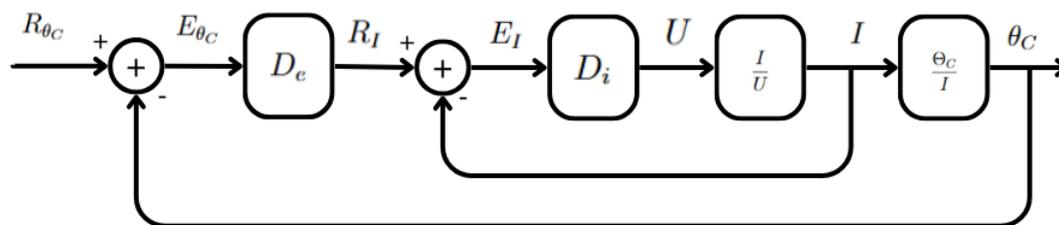


Figura 60: Esquema de control para lazo interno de corriente

Se realizó un análisis de la función de transferencia $\frac{I}{U}$ obtenida experimentalmente en 33, a partir del diagrama de Bode mostrado en la figura 61, en el cual se incluye el retardo mencionado.

Gm = -46.38 dB (at 289.68 rad/s), Pm = -89.55 deg (at 64530.97 rad/s)

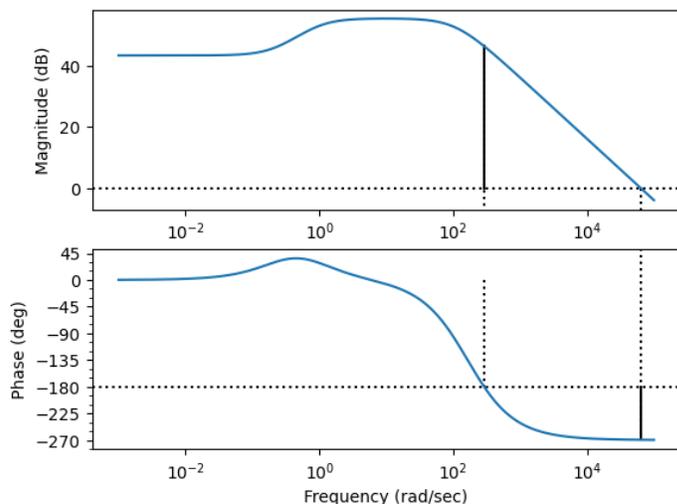


Figura 61: Diagrama de Bode para transferencia de corriente

Se observa que para una realimentación de tipo proporcional la frecuencia de corte será siempre muy alta. No se puede lograr una frecuencia de corte menor con buena performance en baja frecuencia debido a que en las bajas frecuencias la ganancia no tiene pendiente descendente, dado que la función tiene un polo y un cero cuyas ganancias se cancelan mutuamente. Solamente luego del polo de frecuencia 109rad/seg comienza a haber pendiente negativa de 20dB por década. Se propone un controlador integrador, que agregue un polo en el origen y ayude a dar la pendiente que la función no tiene por sí sola, y luego ajustar la ganancia para tener la frecuencia de corte deseada. En la figura 62 se puede ver el diagrama de Bode con el agregado del integrador y una ganancia de 0.0125, lo cual logra un frecuencia de corte de 12.5rad/seg con 80° de Margen de Fase.

Gm = 19.79 dB (at 93.10 rad/s), Pm = 79.54 deg (at 12.43 rad/s)

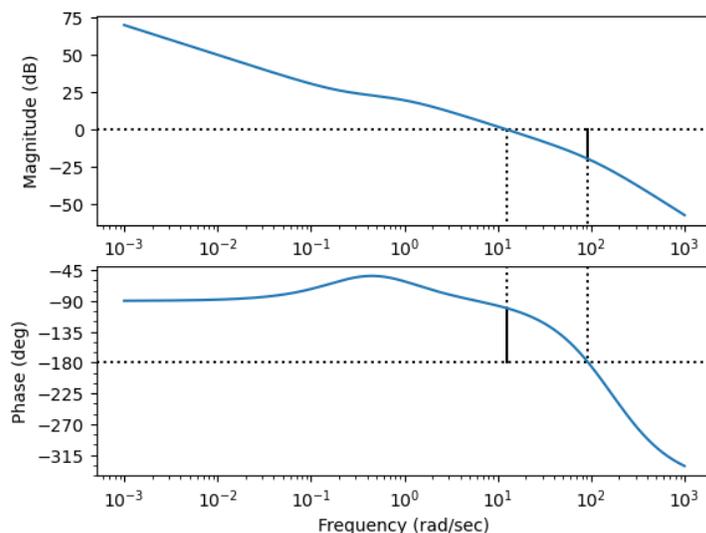


Figura 62: Diagrama de Bode para transferencia de corriente con integrador y ganancia

Con lo cual el controlador interno es:

$$D_i(s) = \frac{0,0125}{s} \quad (58)$$

Aplicando el método de Tustin visto en 45 para el cambio a Transformada Zeta:

$$D_i(z) = \frac{0,000125 + 0,000125z^{-1}}{1 - z^{-1}} \quad (59)$$

Con lo cual el controlador interno en tiempo discreto resulta:

$$u[k] = 0,000125e_i[k] + 0,000125e_i[k - 1] + u[k - 1] \quad (60)$$

Luego se analizó la planta resultante para el lazo de control externo, a partir del diagrama de Bode que se observa en la figura 63, utilizando ganancia negativa.

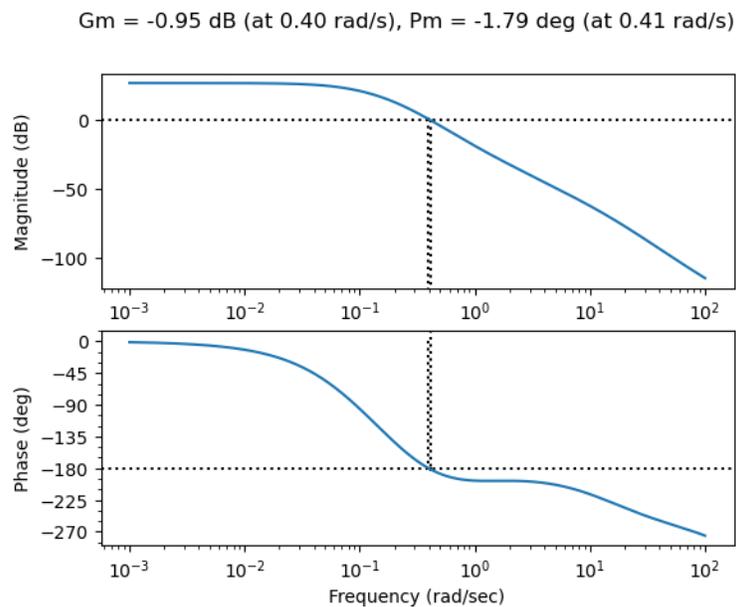


Figura 63: Diagrama de Bode para transferencia de posición con lazo interno de corriente

Se observa que se debe agregar una ganancia $|K| > 1$ para buscar una frecuencia de corte mayor. Con el objetivo de lograr una frecuencia entre 2 y 2.5 rad/seg se colocó una ganancia $K = -50$, logrando un diagrama como se ve en la figura 64.

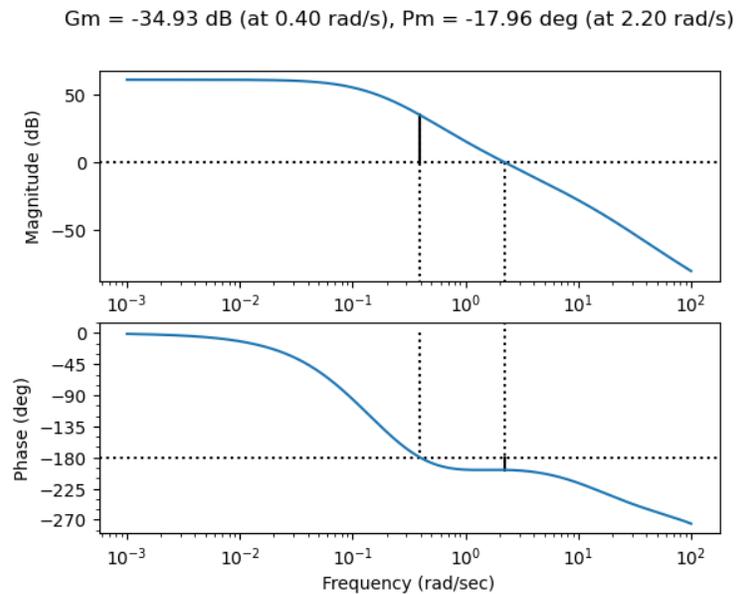


Figura 64: Diagrama de Bode para transferencia de posición con lazo interno de corriente y ganancia

Para mejorar el Margen de Fase resultante se propone colocar una red de adelanto de fase, agregando 80° en la frecuencia de corte lograda. El resultado se observa en la figura 65.

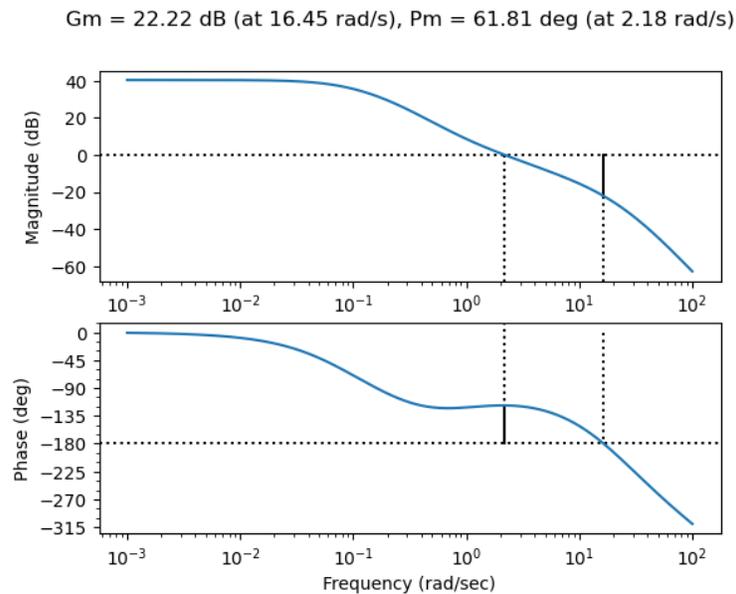


Figura 65: Diagrama de Bode para transferencia de posición con lazo interno de corriente y red de adelanto de fase

La función transferencia del controlador del lazo externo es:

$$D_e(s) = -571,53 \frac{(s + 0,1924)}{(s + 25,1445)} = \frac{(-22,73s - 4,374)}{(0,03977s + 1)} \tag{61}$$

Aplicando el método de Tustin para el cambio a Transformada Zeta:

$$D_e(z) = \frac{(-457,5 + 455,8z^{-1})}{(1 - 0,5981z^{-1})} \tag{62}$$

Con lo cual el algoritmo temporal para la referencia de corriente es:

$$r_i[k] = -457,5e_e[k] + 455,8e_e[k - 1] + 0,5981r_i[k - 1] \tag{63}$$

5.3. Control moderno

El esquema de control moderno se diseña siguiendo una serie de pasos individuales que se combinan para generar el controlador que realimenta el sistema. En primer lugar se diseña una ley de control que sitúa los polos del sistema en lazo cerrado en posiciones determinadas, elegidas para que tengan un desempeño dinámico satisfactorio, y luego se diseña un estimador u observador que calcula el vector de estados estimado a partir de las mediciones. El estimador y la ley de control se combinan para crear un regulador que modifique la dinámica del sistema en lazo cerrado a partir de las mediciones de la salida, es decir un sistema cuyas entradas son las salidas de la planta, y cuya salida es la entrada de la planta. Por último se agrega una nueva entrada de referencia al regulador, creando un controlador que sigue la referencia con la dinámica de la planta a lazo cerrado[12]. En la figura 66 se observa un diagrama en bloques del sistema controlador y la relación interna de sus partes.

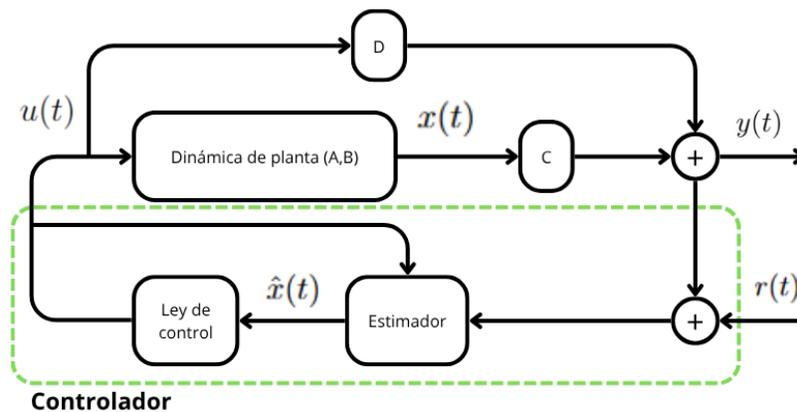


Figura 66: Diagrama esquemático de un controlador de espacio de estados

A partir de la representación por espacio de estados hallada en 39, se agrega el estado de posición angular del cubo θ_C , quedando de la forma:

$$\begin{bmatrix} \dot{i} \\ \dot{\omega}_D \\ \dot{\omega}_C \\ \dot{\theta}_C \end{bmatrix} = \begin{bmatrix} -107,4 & -7,839 & 0 & 0 \\ 4,375 & -0,2694 & 0 & 0 \\ -0,3682 & 0,02267 & -0,02523 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} i \\ \omega_D \\ \omega_C \\ \theta_C \end{bmatrix} + \begin{bmatrix} 54698 \\ 0 \\ 0 \\ 0 \end{bmatrix} u \tag{64}$$

$$y = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} i \\ \omega_D \\ \omega_C \\ \theta_C \end{bmatrix} + 0u$$

Para comenzar el diseño se realizó un análisis de la controlabilidad del sistema en función de los estados elegidos. La controlabilidad es una propiedad inherente de los sistemas dinámicos representados por variable de estado que determina la posibilidad de llevar el sistema de un estado inicial a un estado final a través de sus entradas, en un tiempo finito. Es una función de los estados internos del sistema, por lo que no se la puede determinar a partir de la función transferencia.

Si la matriz de controlabilidad de un sistema es no singular, es decir que tiene determinante distinto de cero y se puede invertir, entonces el sistema es de controlabilidad total, lo cual significa que se pueden controlar todos los estados a través de las entradas definidas. La matriz de controlabilidad se define para un sistema de n-estados como:

$$\mathcal{C} = [B \quad AB \quad A^2B \quad \dots \quad A^{n-1}B] \quad (65)$$

Por el contrario, si la matriz de controlabilidad no es inversible, o está mal condicionada, entonces el sistema no es de controlabilidad total. Esto es equivalente a decir que la matriz de controlabilidad tiene rango menor a n. Se realizó un análisis de la matriz de controlabilidad con el sistema propuesto y se obtuvo:

$$\text{rango}(\mathcal{C}(A, B)) = 3 \quad (66)$$

Con lo cual el sistema no es de controlabilidad total, lo cual es producto de un estado redundante en la representación utilizada. Esto puede deberse a la cancelación cero-polo que existe en la función transferencia, la cual ya se había detectado al calcular las funciones transferencia de manera analítica.

A partir de un sistema no controlable, se puede realizar un cambio de variable de estado con matriz de transformación T , para pasar a una nueva representación del sistema que sea controlable, de la forma:

$$\begin{aligned} \bar{x} &= T^{-1}x \\ \bar{A} &= T^{-1}AT \\ \bar{B} &= T^{-1}B \\ \bar{C} &= CT \end{aligned} \quad (67)$$

Donde las matrices \bar{A} y \bar{B} se dividen en una parte controlable y una parte no controlable, de la siguiente manera:

$$\begin{aligned} \bar{A} &= \begin{bmatrix} A_c & A_{1,2} \\ 0 & A_{nc} \end{bmatrix} \\ \bar{B} &= \begin{bmatrix} B_c \\ 0 \end{bmatrix} \end{aligned} \quad (68)$$

De modo que si el rango de la matriz de controlabilidad $\mathcal{C}(A, B)$ es menor que el número de estados n, para la nueva matriz de controlabilidad $\mathcal{C}(A_c, B_c)$ el rango será igual, y se podrá hacer la reducción de estados descomponiendo al sistema en sus partes controlable y no controlable[10][11], como se muestra en la figura 67.

Al realizar la transformación para la representación de estados controlable, se obtuvo un nuevo

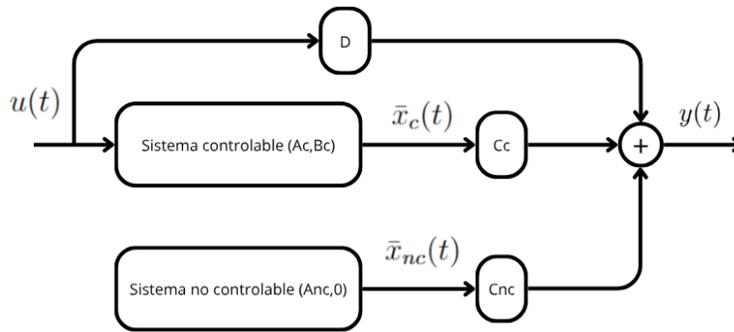


Figura 67: Descomposición en parte controlable y no controlable

modelo, en el cual el nuevo vector de estados se denomina \bar{x} :

$$\begin{aligned} \dot{\bar{x}} &= \begin{bmatrix} -0,02505 & 0,08389 & 5,326 \times 10^{-16} \\ -0,0005167 & -0,2696 & 4,39 \\ -0,01652 & -7,812 & -107,4 \end{bmatrix} \begin{bmatrix} \bar{x}_1 \\ \bar{x}_2 \\ \bar{x}_3 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 54698 \end{bmatrix} u \\ y &= \begin{bmatrix} 0 & 0 & 1 \\ 0,002107 & 0,9965 & 0 \\ 0,02504 & -0,08386 & 0 \\ -0,9997 & 0 & 0 \end{bmatrix} \begin{bmatrix} \bar{x}_1 \\ \bar{x}_2 \\ \bar{x}_3 \end{bmatrix} + 0u \end{aligned} \tag{69}$$

Una cancelación cero-polo implica que puede haber una función transferencia estable pero un sistema internamente inestable, debido a lo cual se define a un sistema estabilizable si su parte no controlable es estable. En la transformación de estados realizada se pudo comprobar que la matriz de cambio de estados de la parte no controlable A_{nc} posee un polo en el origen, es decir que es condicionalmente estable.

Por otro lado, se realizó un análisis de la observabilidad del sistema. La observabilidad es una propiedad que depende de las salidas y de la matriz de estados, y se define como la posibilidad de determinar el estado de un sistema a partir de lecturas en las salidas en un tiempo acotado.

De manera análoga a la matriz de controlabilidad existe la matriz de observabilidad, definida como:

$$\mathcal{O} = \begin{bmatrix} C \\ CA \\ CA^2 \\ \dots \\ CA^{n-1} \end{bmatrix} \tag{70}$$

Si dicha matriz es no singular, o equivalentemente de rango total n , el sistema es observable. Para el nuevo vector de estados reducido se obtuvo:

$$rango(\mathcal{O}(A_c, C_c)) = 3 \tag{71}$$

Con lo cual se determinó que el sistema es observable a partir de las mediciones disponibles, y se procedió con el diseño de la realimentación para esta nueva representación de estados.

La ley de control determina el valor de la entrada u a partir de una combinación lineal de los estados, de la forma:

$$u = -Kx = - [K_1 \quad K_2 \quad K_3] \begin{bmatrix} \bar{x}_1 \\ \bar{x}_2 \\ \bar{x}_3 \end{bmatrix} \tag{72}$$

La matriz de ganancias K se calcula a partir de la ubicación deseada de los polos a lazo cerrado, utilizando el método conocido como fórmula de Ackermann[12]. En un sistema de controlabilidad total los polos pueden ser elegidos de forma arbitraria, siendo válida cualquier ubicación posible. Esto implica mucha libertad en la decisión y a la vez mucho margen de mejora para diseñar un controlador que haga el esfuerzo óptimo a la hora de aplicar la acción de control.

Una de las técnicas más efectiva y más utilizada para elegir la ubicación de los polos es el regulador lineal cuadrático óptimo (LQR), en el cual se busca minimizar un funcional de costo de la forma:

$$J = \int_0^{\infty} (x^T Q x + u^T R u) dt \quad (73)$$

Donde las matrices Q y R determinan el peso que tienen el error de seguimiento en los estados y el costo en las entradas, respectivamente. Para el diseño se utilizó la regla de Bryson, en donde las matrices se eligen diagonales y se toma como punto de partida:

$$Q_{i,i} = 1/\text{máximo valor aceptable de } [x_i^2]$$

$$R_{i,i} = 1/\text{máximo valor aceptable de } [u_i^2]$$

Se realizó un proceso iterativo, obteniendo la colocación de polos óptima en las siguientes ubicaciones:

$$[-2,51 + 0,33j \quad -2,51 - 0,33j \quad -177,2]$$

Las matrices Q y R utilizadas en el algoritmo de LQR que lograron la colocación de polos elegida fueron respectivamente:

$$Q = \begin{bmatrix} 1,6 \times 10^{-3} & 0 & 0 \\ 0 & 3,33 \times 10^{-6} & 0 \\ 0 & 0 & 3,33 \times 10^{-6} \end{bmatrix} \quad R = 0,5$$

La frecuencia de los polos dominantes ω_d es cercana a 2.5rad/seg y el sistema está sobreamortiguado, de forma similar a los controladores diseñados previamente utilizando métodos de control clásico. La matriz de ganancias K que logra esta ubicación de polos es:

$$K = [0,0555 \quad 0,0034 \quad 0,0014] \quad (74)$$

El diseño de la ley de control se realiza suponiendo que se poseen todas las variables de estado como información para la realimentación. A fin de garantizar que se cumpla se debe diseñar un observador de orden completo que estime los estados \hat{x} con mínimo error, de forma que se pueda reemplazar en la ley de control quedando:

$$u = -K\hat{x} \quad (75)$$

Al tener la información de la planta (A,B) y la información de la entrada $u(t)$, se puede utilizar el modelo dinámico de la planta para crear un estimador de estados de lazo abierto. Sin embargo, pueden existir errores debido a condiciones iniciales no nulas, perturbaciones externas, o diferencias entre el modelo y la planta real. Debido a eso se propone realizar un estimador realimentado en el cual se controle la dinámica relacionada con el error de estimación $\tilde{x} \triangleq x - \hat{x}$. En la figura 68 se muestra un diagrama interno del estimador realimentado (asumiendo la matriz de feed-forward D igual a cero), en el cual la matriz de ganancias L realiza la realimentación como una combinación lineal del error en la salida \tilde{y} .

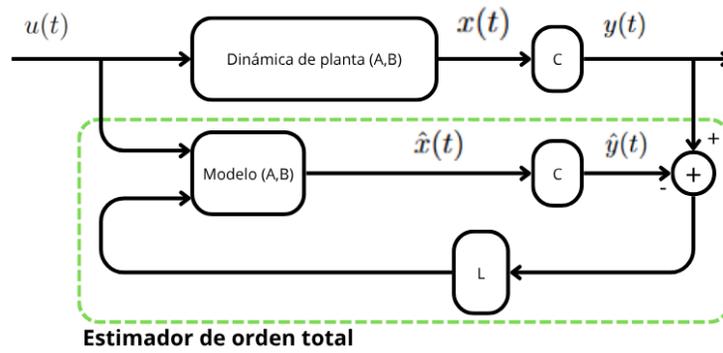


Figura 68: Diagrama en bloques del estimador realimentado

Para que la dinámica del error de estimación converja más rápidamente que la dinámica de la planta se suele elegir polos entre 5 y 10 veces más rápidos que los polos a lazo cerrado del sistema. En este caso se colocaron los polos en:

$$[-5 \quad -10 \quad -100]$$

Para lo cual la matriz L resulta:

$$L = \begin{bmatrix} 0,0000 & 0,0941 & 0,1175 & -4,9734 \\ 4,3904 & 9,6962 & -0,8161 & 0,0005 \\ -7,4074 & -7,7843 & 0,6547 & 0,0165 \end{bmatrix} \tag{76}$$

El sistema regulador, combinando la ley de control y el estimador, es un sistema de la forma:

$$\begin{aligned} \dot{\hat{x}} &= (A - BK - LC)\hat{x} + Ly \\ u &= -K\hat{x} \end{aligned} \tag{77}$$

Dicho regulador se sitúa en el lazo de realimentación. Luego, se desea agregar la referencia como una entrada más del sistema, lo cual se hace con coeficientes de feed-forward de la siguiente manera:

$$\begin{aligned} \dot{\hat{x}} &= (A - BK - LC)\hat{x} + Ly + Mr \\ u &= -K\hat{x} + \bar{N}r \end{aligned} \tag{78}$$

En la figura 69 se observa un diagrama en bloques del agregado de la referencia.

Se desea que en estado estacionario la salida sea igual que la referencia, y además que los cambios en la referencia no modifiquen el error de estimación. Los coeficientes \bar{N} y M se calculan con las siguientes fórmulas:

$$\begin{aligned} \begin{bmatrix} N_x \\ N_u \end{bmatrix} &= \begin{bmatrix} A & B \\ C & D \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 0 \\ \dots \\ 1 \end{bmatrix} \\ \bar{N} &= N_u + KN_x \\ M &= B\bar{N} \end{aligned} \tag{79}$$

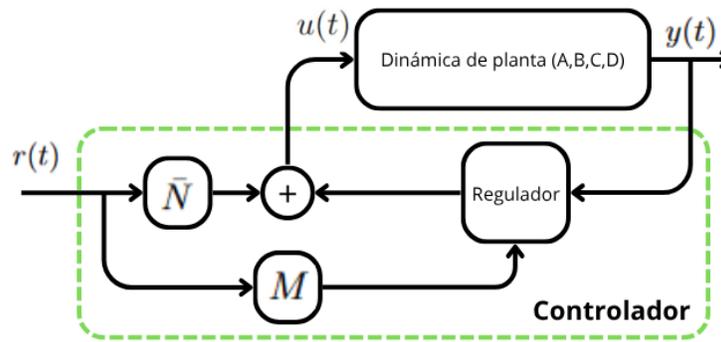


Figura 69: Diagrama en bloques del controlador (regulador y referencia)

Como paso final, se realiza el armado de un sistema discreto tipo máquina de estados, que calcule el siguiente estado y la acción de control en base al estado actual y a las mediciones, de la siguiente manera:

$$\begin{bmatrix} \hat{x}_1[k+1] \\ \hat{x}_2[k+1] \\ \hat{x}_3[k+1] \end{bmatrix} = A_k \begin{bmatrix} \hat{x}_1[k] \\ \hat{x}_2[k] \\ \hat{x}_3[k] \end{bmatrix} + B_k \begin{bmatrix} i[k] \\ \omega_D[k] \\ \omega_C[k] \\ \theta_C[k] \\ r[k] \end{bmatrix} \quad (80)$$

$$u[k] = C_k \begin{bmatrix} \hat{x}_1[k] \\ \hat{x}_2[k] \\ \hat{x}_3[k] \end{bmatrix} + D_k \begin{bmatrix} i[k] \\ \omega_D[k] \\ \omega_C[k] \\ \theta_C[k] \\ r[k] \end{bmatrix}$$

Realizando la transformación a variable discreta del sistema controlador, utilizando el método de Tustin, se obtienen los valores para las matrices:

$$A_k = \begin{bmatrix} 0,9048 & -3,225 \times 10^{-17} & -3,312 \times 10^{-19} \\ 1,531 \times 10^{-16} & 0,8182 & -5,882 \times 10^{-18} \\ -21,05 & -1,221 & -0,2715 \end{bmatrix}$$

$$B_k = \begin{bmatrix} 9,655 \times 10^{-18} & 0,001792 & 0,002239 & -0,09473 & 1,92 \times 10^{-17} \\ 0,07983 & 0,1763 & -0,01484 & 9,392 \times 10^{-6} & 1,821 \times 10^{-16} \\ -0,1076 & -0,1949 & -0,01001 & 1,047 & -22,55 \end{bmatrix} \quad (81)$$

$$C_k = [-0,03848 \quad -0,002232 \quad -0,0004963]$$

$$D_k = [-6,125 \times 10^{-5} \quad -0,000214 \quad -3,026 \times 10^{-5} \quad 0,001914 \quad -0,04123]$$

6. Resultados obtenidos

Se realizó la implementación de los controladores en el entorno de programación Python, utilizando el programa descrito anteriormente y modificando únicamente lo relativo al algoritmo de control, según cada caso de controlador propuesto. Para realizar un estudio del comportamiento de los controladores se propone analizar la respuesta a un escalón en la referencia de 20° luego de un segundo. Se analizará individualmente cada controlador tratando de indentificar las características principales en la respuesta obtenida de manera que se pueda hacer un análisis cualitativo del desempeño de cada uno y una comparación entre todos ellos. Como variables de análisis se utilizarán el tiempo de crecimiento r_t definido como el tiempo que tarda el sistema en llegar al 90 % de su valor estacionario, tiempo de establecimiento s_t definido como el tiempo que tarda en llegar al 98 % de su valor estacionario, error de estado estacionario e_{ss} y sobrepico.

Además, para cada controlador se realizará una simulación de la respuesta a un escalón de las mismas características, para poder comprobar la diferencia entre el modelo y su respuesta simulada con la implementación empírica, con las dificultades que presenta y las posibles desviaciones del comportamiento ideal esperado. En la figura 70 se observa un ejemplo del simulador implementado en SimuLink [26], donde se utilizó el modelo obtenido experimentalmente en 35, 31 y 33. Se agregaron bloques no lineales que representan la saturación de la acción de control y la discontinuidad que se generó en el cruce por cero para evitar la banda muerta.

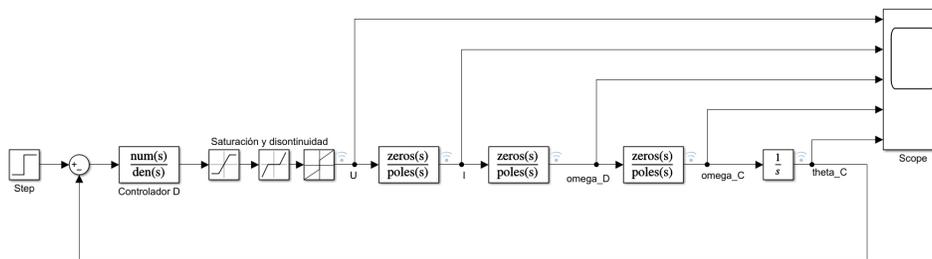


Figura 70: Simulador realizado en SimuLink para control de lazo simple

Se observará la respuesta en todas las variables del sistema de las cuales se tiene medición, es decir la acción de control U , la corriente I , la velocidad angular del disco ω_D , la velocidad angular del cubo ω_C y la posición angular del cubo θ_C .

6.1. Controlador de adelanto de fase

Para el controlador de adelanto de fase, cuyo algoritmo fue descrito en 49, se obtuvieron los resultados que se observan en la figura 71.

En la simulación se ve una respuesta de θ_C con tiempo de establecimiento de 4 segundos con error despreciable de estado estacionario, y tiempo de crecimiento de 1 segundo. Se aprecian distorsiones en la forma de la curva con respecto a un sistema SLIT debidas al modelo no lineal utilizado.

La respuesta medida, por otro lado, tiene un tiempo de crecimiento de 2seg y tiempo de establecimiento de 2.5seg pero no logra llegar a la posición deseada, manteniendo un error de estado estacionario de 17.5 %. El comportamiento de las variables intermedias es muy similar en todos los casos, aunque se puede ver una menor velocidad angular que la esperada tanto en el disco como en el cubo. La velocidad angular del motor resulta nula en el estado estacionario, con lo cual no hay consumo de energía excesivo.

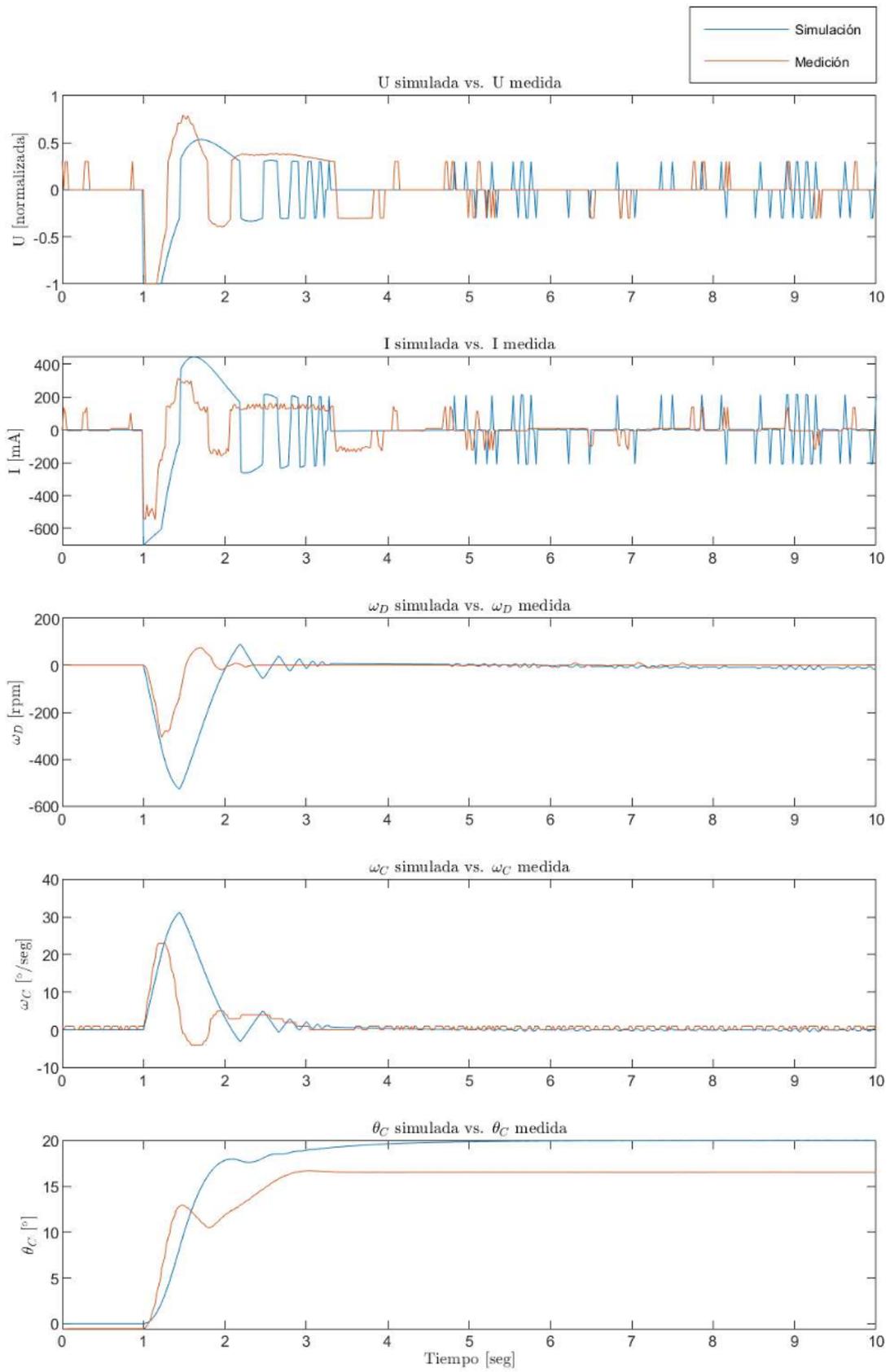


Figura 71: Resultados del controlador de adelanto de fase

6.2. Controlador proporcional integrativo derivativo

En la figura 72, por otro lado, se ven los resultados del controlador de tipo proporcional integrativo derivativo de lazo simple. La implementación se hizo a partir del algoritmo propuesto en 55, mientras que la simulación se realizó a partir del sistema mostrado en la figura 70 modificando el bloque correspondiente al controlador.

Se puede observar que, a diferencia de lo esperado, en la simulación no se logró un error cero en estado estacionario, debido a las alinealidades introducidas, siendo este de 2.5 %. La respuesta medida tampoco logra alcanzar el valor de referencia, pero el error que se alcanza es de 7.5 %, menor que en el caso anterior pero mayor que en la simulación. Sin embargo, el controlador continúa exigiendo acción de control, a partir de aplicar una tensión mínima que acciona al motor.

La acción de control solamente llegó a la saturación en el momento que inicia el escalón en la referencia, mientras que el resto del tiempo se mantuvo en la zona linealizada. Sin embargo, se mantuvo mucho tiempo en el límite inferior de esa zona, lo cual puede haber producido que el motor responda con una dinámica más lenta que la modelada. Esto se debe a que el modelo se realizó tomando un promedio estadístico de su respuesta frente a escalones de distinta amplitud, pero se había visto en la figura 25 que para tensiones menores la respuesta es más lenta.

Además, se observa que sobre el final del experimento la velocidad angular del disco continuó en aumento pero la velocidad angular del cubo se mantuvo en cero. Esto se puede deber a rozamientos en el cubo que dificulten su giro, y al ser una acción de control de amplitud tan baja, el cubo no puede superar esa fuerza. En el estado estacionario se obtiene un estado no deseado en el cual el motor está en funcionamiento y consumiendo energía pero su acción no se ve reflejada en el cubo. Esto se debe a la cancelación cero-polo que existe en la planta, la cual no se puede controlar a partir de un lazo externo. Esta restricción de la implementación es la que lleva a proponer controladores más complejos que consideren más variables.

La dinámica de la respuesta de θ_C en simulación fue más rápida que en el caso anterior, con un tiempo de crecimiento de 1.5seg y llegando al establecimiento en 2.5seg. Sin embargo, la respuesta medida fue más lenta, con un tiempo de crecimiento de 4seg y de establecimiento de 6seg, en el cual además no se lleva a cero la velocidad del motor. La diferencia entre la respuesta simulada y la respuesta medida se hace más evidente a medida que avanza el tiempo del experimento, debido a que en la simulación se tiene un valor de velocidad angular del motor cercana a cero, mientras que en la realidad sucedió que el motor nunca bajó su velocidad sino que fue en aumento.

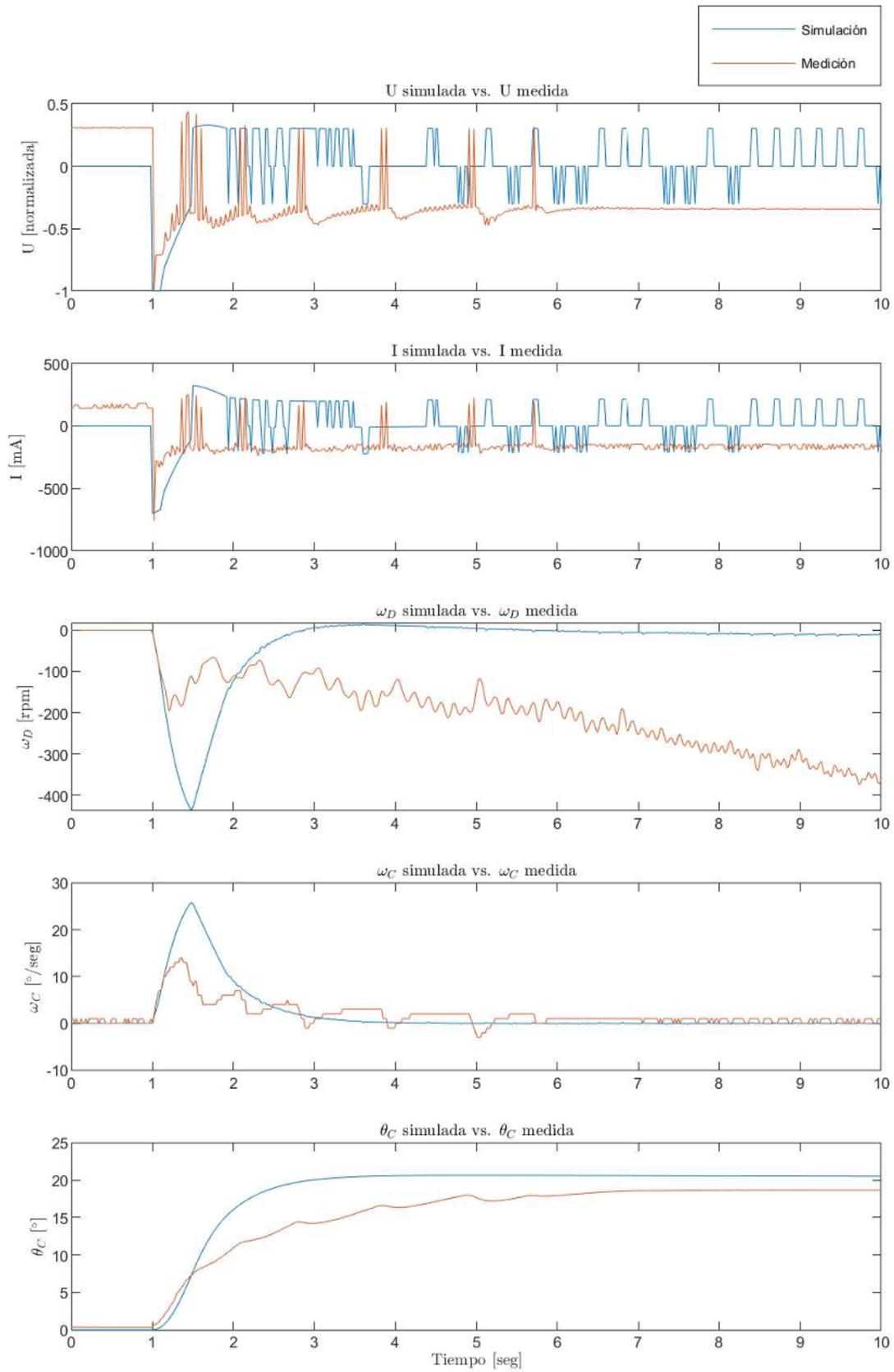


Figura 72: Resultados del controlador PID

6.3. Controlador de lazo interno de velocidad angular del disco

En la figura 74 se pueden ver los resultados del controlador de lazo interno de velocidad angular del disco y lazo externo de posición angular del cubo, implementado a partir de 56 y 57, comparado contra la simulación realizada a partir del sistema mostrado en la figura 73.

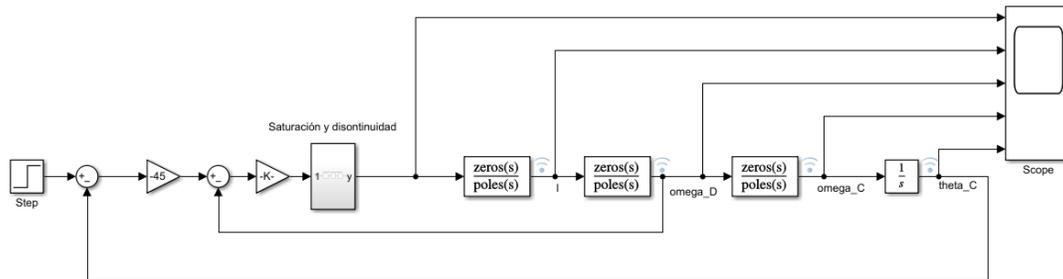


Figura 73: Simulador para controlador de lazo interno de velocidad angular del disco

La respuesta medida tiene un comportamiento con mayores oscilaciones en la velocidad angular tanto del disco como del cubo, y se esperaría ver un sobrepico en la respuesta de la posición angular. Sin embargo, se ve que la oscilación resultante en la posición es de amplitud menor que la posición final lograda. La acción de control aplicada fue de gran amplitud en tiempos breves, logrando estabilizar luego el disco con velocidad angular nula. El tiempo de crecimiento es de 0.7seg y el establecimiento se logra en 2seg con 4 % de error.

Esto difiere de lo visto en la simulación, en donde las velocidades angulares del disco y del cubo tienen un crecimiento rápido seguido de un decrecimiento amortiguado sin oscilaciones, y la respuesta vista en la posición angular resulta apenas subamortiguada con un sobrepico de 2.5 %. El tiempo de crecimiento es de 2.5 % y el tiempo de establecimiento es de 4seg, llegando al estado estacionario con error despreciable.

el tiempo de respuesta de la velocidad angular del cubo frente a los cambios de velocidad angular del disco resultan más rápidos en la implementación, y el lazo interno de velocidad angular del disco se comporta con mayor velocidad de establecimiento que el simulado y con menor amortiguamiento. Esta diferencia es la que se observa luego en la respuesta de la posición angular del cubo.

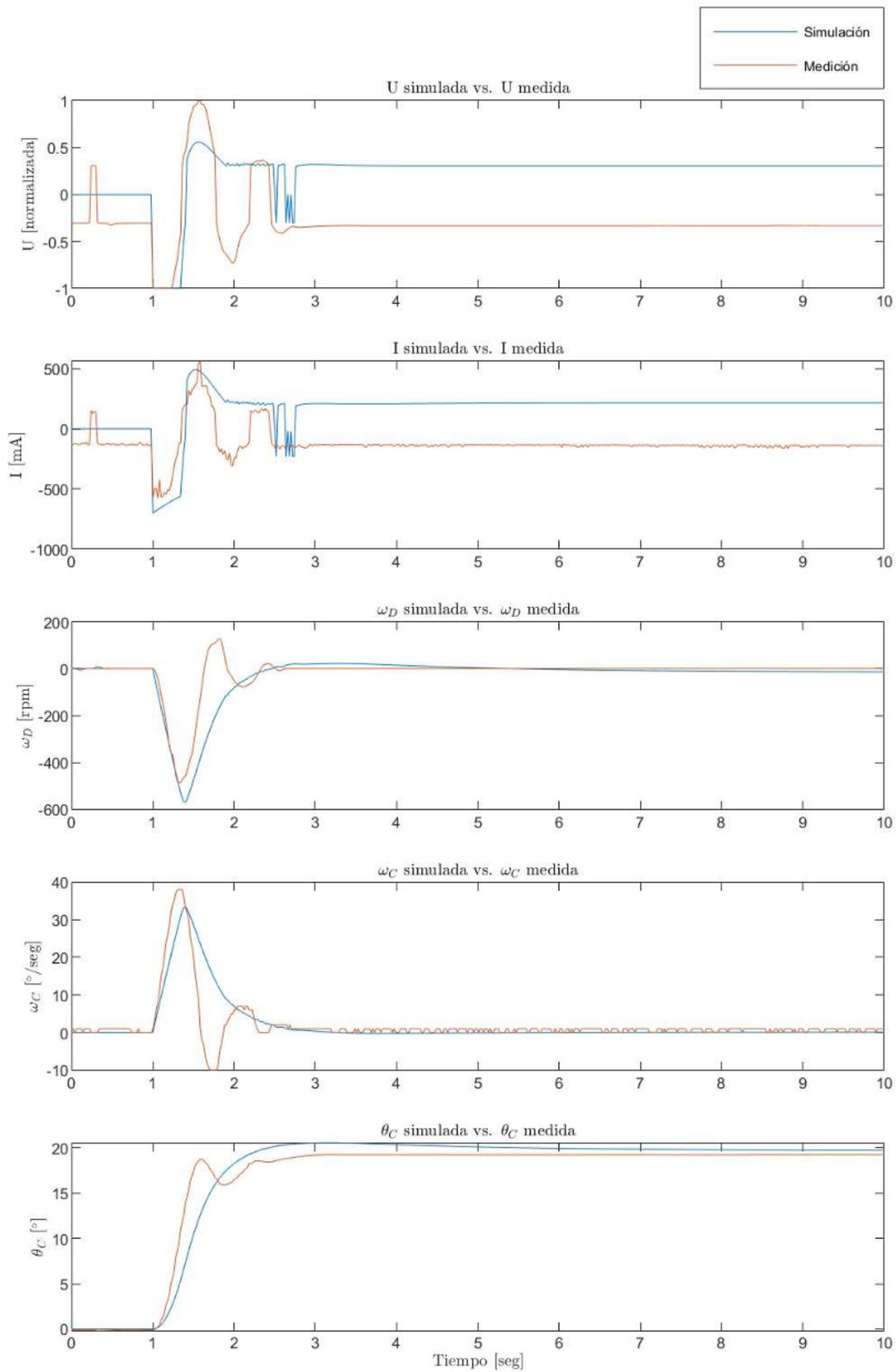


Figura 74: Resultados del controlador de lazo interno de velocidad angular del disco

6.4. Controlador de lazo interno de corriente

En la figura 76 se observan los resultados del controlador de lazo interno de corriente, implementado a partir de 60 y 63, comparado contra la simulación realizada a partir del sistema mostrado en la figura 75.

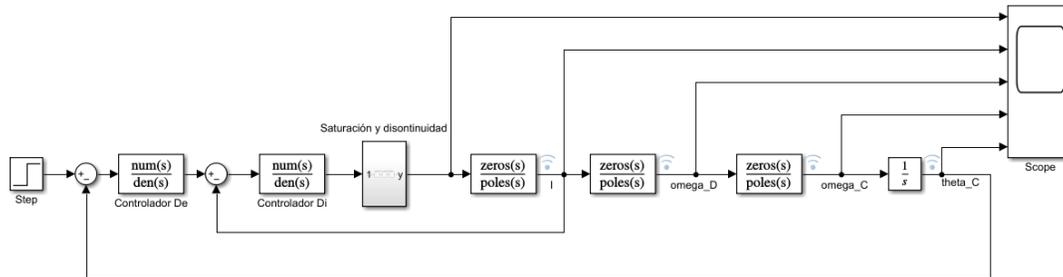


Figura 75: Simulador para controlador de lazo interno de corriente

En este caso la respuesta simulada presenta un sobrepico marcado en la posición angular de 7.5 %, y tiempos de crecimiento y establecimiento largos, 1.5 y 4 segundos respectivamente. Los tiempos largos provienen del controlador integral que hay en el lazo interno, ya que las integraciones presentan respuestas prolongadas. La respuesta medida es nuevamente de mayor velocidad y comportamiento más oscilatorio que la respuesta simulada, pero con menor amplitud en los picos de velocidad angular del disco y velocidad angular del cubo. El tiempo de crecimiento medido es de 0.7seg y el de establecimiento es de 2seg, mientras que no hay sobrepico en la posición angular debido a que el comportamiento oscilatorio del inicio no sobrepasa el valor final.

La posición final no resulta ser la esperada, existiendo un error de estado estacionario en la posición medida de 5 %, mientras que en la simulación el error es despreciable. Además, se puede ver una corriente no nula circulando en el circuito del motor aún cuando no hay velocidad en el mismo. Es decir que se encuentra en la zona de banda muerta, y la corriente circula por las bobinas sin que esto se manifieste en una acción de giro. Esto es un estado no deseado similar al que también se manifiesta en el caso del controlador de lazo interno de velocidad angular del cubo. En ambos controladores se logra estabilizar el disco en velocidad final nula pero con consumo de corriente en el circuito del motor. Se puede ver en el caso de lazo interno de ω_D (figura 74) que la simulación mantiene una acción de control y una corriente no nulas hasta el final del experimento, por lo que ese comportamiento en el experimento medido no está fuera de las posibilidades. Sin embargo, en el caso del lazo interno de corriente se esperaría que el control no presente ese consumo innecesario.

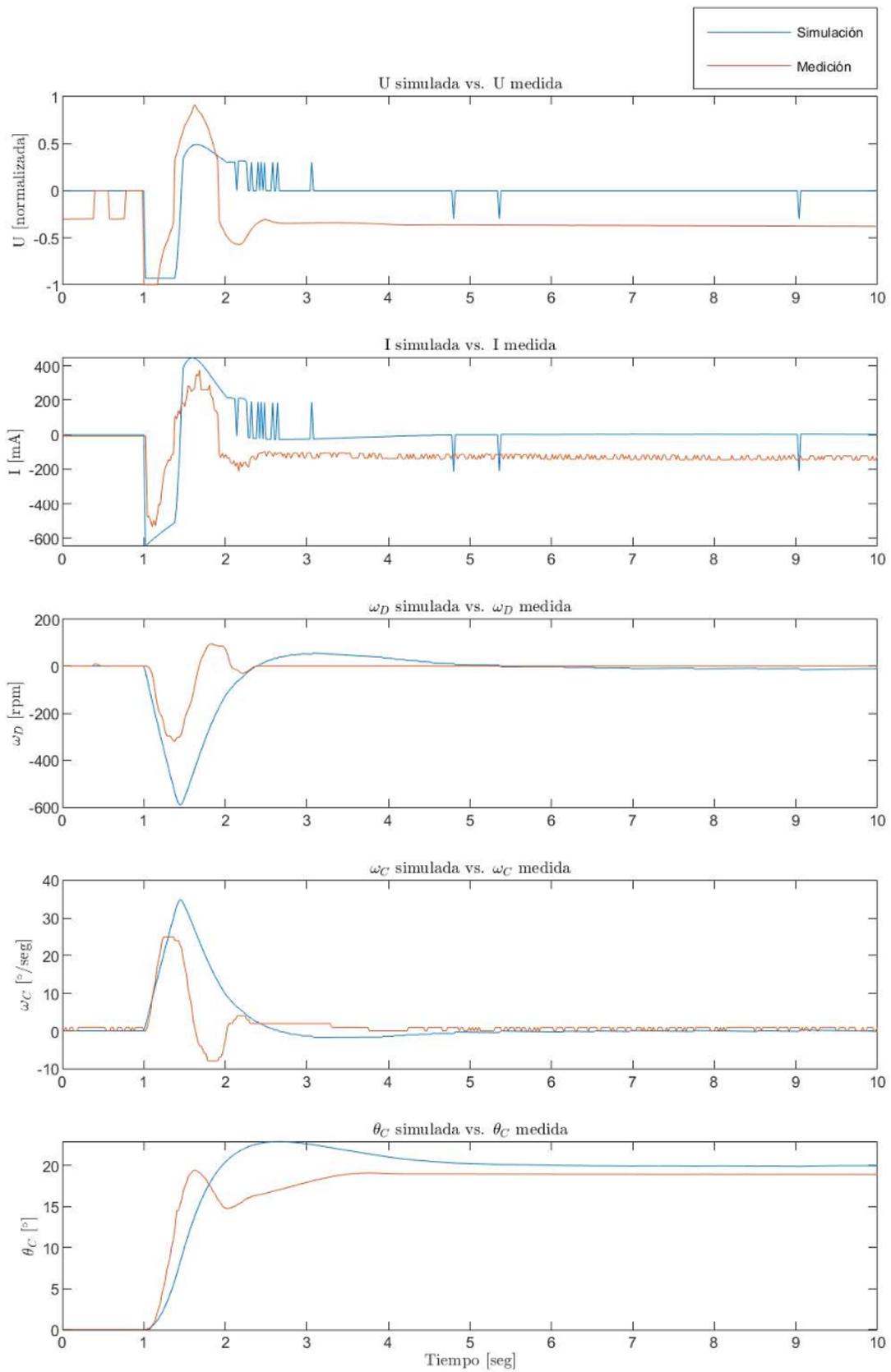


Figura 76: Resultados del controlador de lazo interno de corriente

6.5. Controlador moderno

En la figura 78 se observan los resultados del controlador moderno implementado según 80 y 81, comparados contra la simulación hecha a partir del esquema mostrado en la figura 77. Se grafica además el valor del estimador de estados implementado, para comprobar su dinámica.

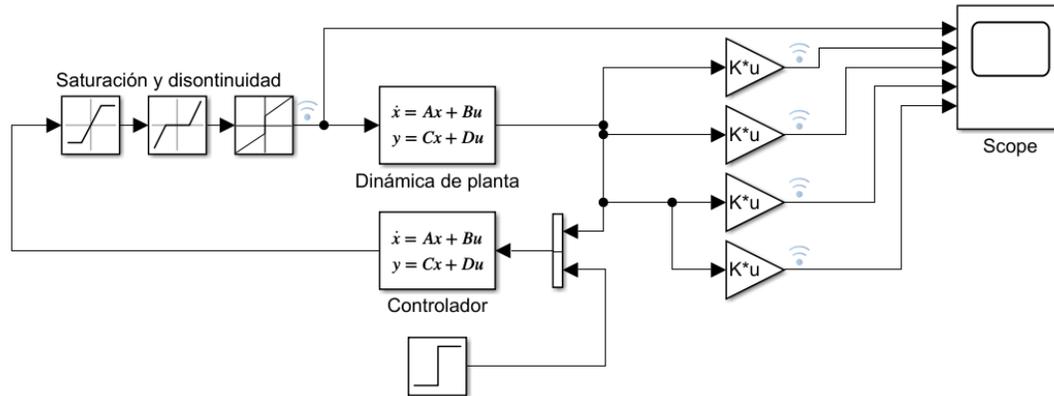


Figura 77: Simulador para controlador moderno

Se observa que la estimación de corriente y de posición del cubo es correcta, pero tanto en la velocidad angular del disco como en la velocidad angular del cubo existe un error de estado estacionario en la estimación. Sin embargo, la dinámica de cambio se estima correctamente en todos los casos.

La respuesta temporal medida es rápida, con un tiempo de crecimiento de 0.8seg y un tiempo de establecimiento de 2seg. El error de estado estacionario es de 5%, y hubo un sobrepico inicial de 1%. Hay una diferencia considerable entre la respuesta medida y la respuesta simulada, ya que esta última está sobreamortiguada y tiene un tiempo de crecimiento de 2seg y un tiempo de establecimiento de 3.5seg con error de estado estacionario de 1%.

La acción de control aplicada con este controlador no llegó a la saturación, siendo el único de los esquemas de control que se mantiene dentro de la zona de trabajo. Esto se debe a las limitaciones impuestas en el diseño de la ley de control mediante el funcional de costos del controlador LQR.

Por otro lado, no se mantiene la acción de control en valor cero pasado el tiempo de establecimiento, se mantiene una corriente constante que no logra hacer girar el motor y por lo tanto no logra terminar de corregir el error de estado estacionario. Además, el estimador utiliza la información de la acción de control no nula para calcular los estados estimados de la planta, con lo cual estima que la acción de control se verá reflejada en una velocidad angular del motor no nula, provocando esa diferencia entre los estados estimados y los estados reales. Para el caso de la estimación del estado de posición angular del cubo se toma más confianza de las mediciones que del modelo, por lo que la estimación no presenta grandes diferencias con el valor real.

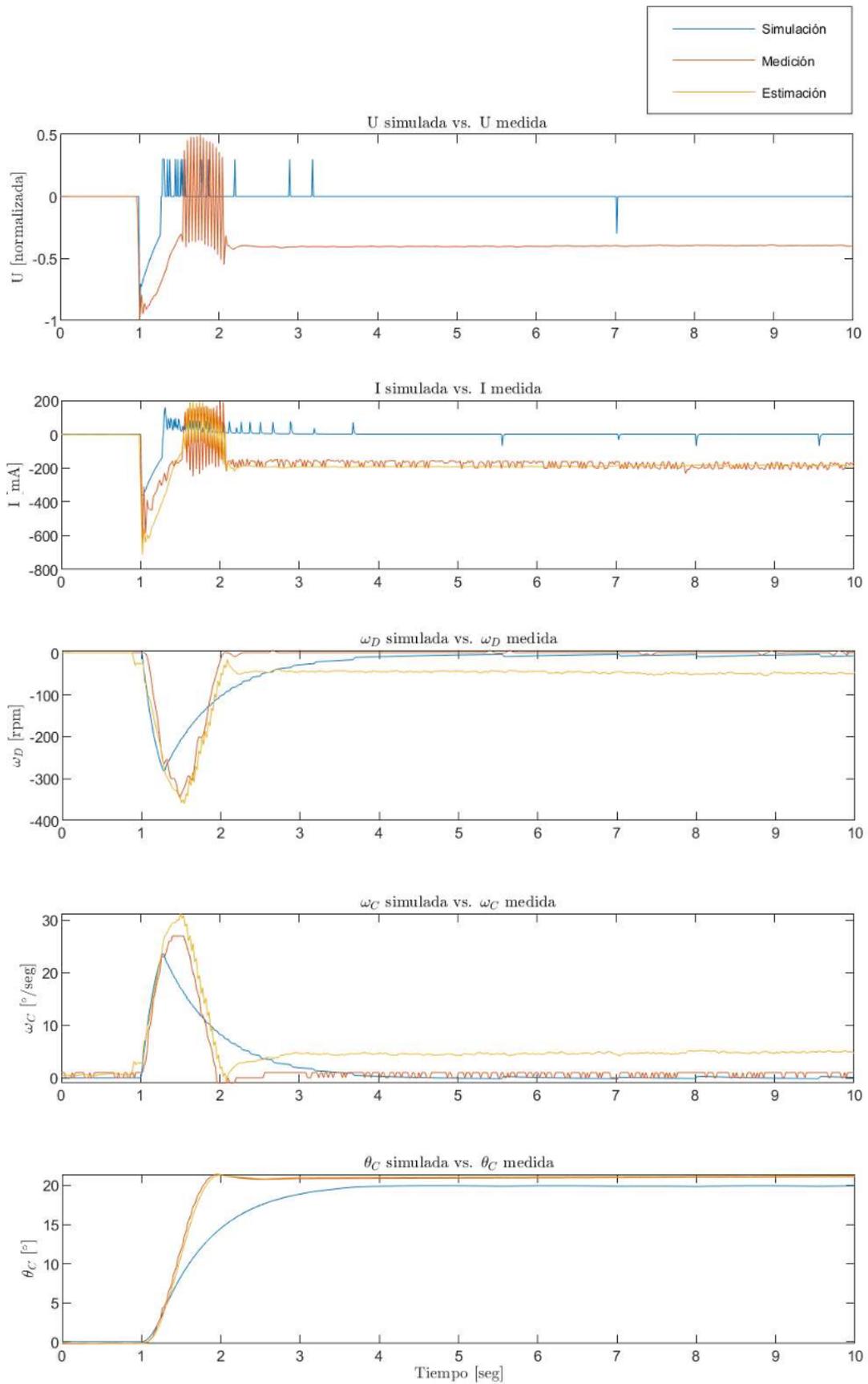


Figura 78: Resultados del controlador moderno

7. Conclusiones

Se diseñó y construyó un sistema de control de posición angular mediante rueda de reacción que permite la implementación y estudio de distintos esquemas de control. Se realizó un análisis matemático de la planta y diferentes mediciones para la identificación de la misma, obteniendo distintos modelos matemáticos que la representan (funciones transferencia y espacio de estados). Se controló el establecimiento de posición de referencia realizando el diseño y la implementación de controladores clásicos utilizando una o dos mediciones de realimentación, y de un controlador moderno utilizando todas las mediciones disponibles para la realimentación.

En la siguiente tabla se condensan los resultados obtenidos con los distintos controladores, tanto de las simulaciones como de la implementación.

Controlador	Tiempo de crecimiento r_t	Tiempo de establecimiento s_t	Sobrepico	Error estacionario e_{ss}	Corriente final	Velocidad del motor final
Adelanto de fase simulado	1seg	4seg	No	Despreciable	Nula	Nula
Adelanto de fase implementado	2seg	2.5seg	No	17.5 %	Nula	Nula
PID simulado	1.5seg	2.5seg	No	2.5 %	Nula	Nula
PID implementado	4seg	6seg	No	7.5 %	No nula	No nula
Lazo interno ω_D simulado	1.5seg	4seg	Si 2.5 %	Despreciable	No nula	Nula
Lazo interno ω_D implementado	0.7seg	2seg	No	4 %	No nula	Nula
Lazo interno i simulado	1seg	4seg	Si 7.5 %	Despreciable	Nula	Nula
Lazo interno i implementado	0.7seg	3seg	Si 2 %	5 %	No nula	Nula
Moderno simulado	2seg	3.5seg	No	1 %	Nula	Nula
Moderno implementado	0.8seg	2seg	Si 1 %	5 %	No nula	Nula

Los controladores implementados cumplieron con el requisito de un tiempo de establecimiento menor a 10seg, pero en ningún caso se logró controlar la posición sin error de estado estacionario. Los controladores más lentos fueron el controlador PID y el controlador de lazo interno de corriente. Ambos controladores incluyen integradores dentro del lazo de control, por lo que se obtienen tiempos de respuesta más lentos. Con respecto a los sobrepicos, en la mayoría de los casos no hubo sobrepico en la implementación, y en los controladores que sí hubo fueron de baja amplitud (máximo 2% en el caso del lazo interno de corriente).

La respuesta medida fue más rápida que la respuesta simulada en casi todos los casos, salvo en el caso del controlador PID. Esto se puede deber a una diferencia de ganancia entre la planta real y el modelo identificado, por lo que al cerrar los lazos de control implementados se logra una mayor frecuencia de corte que la diseñada, resultando en tiempos más rápidos. La diferencia entre la respuesta implementada y la respuesta simulada fue muy notoria en todos los parámetros, siendo las respuestas implementadas de mayor rapidez, menor sobrepico y mayor error de estado estacionario. Se evidencia que el modelo fue de utilidad para tener una guía a la hora de comenzar los diseños de control pero no debe ser tomado en cuenta para el análisis de los resultados, debido a la brecha entre lo teórico y lo práctico.

Además, un problema que se generalizó para la mayoría de los controladores fue el cruce por cero del actuador, ya que ocurrió en varias situaciones que la acción de control aplicada no se veía reflejada en la planta. En el estado estacionario de los controladores simulados la acción de control final se estabiliza cerca del cero, por lo que no se suministra corriente, con la salvedad del controlador de lazo interno de velocidad angular del disco. Por otro lado, en el estado estacionario de los controladores implementados la acción de control se estabiliza en un valor distinto de cero, suministrando corriente para terminar de corregir el error de estado estacionario, con la salvedad del controlador de adelanto de fase. Sin embargo, el motor no se mueve debido a la fricción estática, que presenta un comportamiento no lineal en el cual se requiere mayor tensión para comenzar el movimiento que para mantenerlo. Si bien se generó una rutina de predistorsión mediante una tabla de consulta con la intención de evitar este comportamiento, el motor resultó no estar del todo bien identificado y en muchos casos la acción de control estuvo dentro de la banda muerta.

El controlador que mejor desempeño obtuvo en la implementación fue el controlador moderno, ya que fue más rápido y preciso que los controladores clásicos, y no saturó la acción de control. Los controladores de doble lazo son ambos de rapidez similar pero la curva de respuesta está más degradada y la acción de control se saturó en ambos casos. En los controladores de lazo simple el tiempo de crecimiento fue más lento y el error de estado estacionario fue mayor. Es un resultado esperado, ya que en el controlador moderno se utiliza un mayor número de mediciones para controlar todos los parámetros de la planta, y además el diseño se realizó siguiendo criterios de control óptimo.

Por otra parte, el control realizado no es robusto frente a perturbaciones, por lo que los controladores implementados no son capaces de contrarrestar un torque externo que modifique el punto de operación. Para ello es necesario diseñar un esquema de control que incluya las perturbaciones dentro del modelo para poder rechazarlas satisfactoriamente.

Como lecciones aprendidas en la construcción de este prototipo se puede concluir que un motor de escobillas no es el actuador más adecuado para el sistema, debido a que posee un gran rozamiento estático y no es muy preciso en cuanto a la velocidad lograda en función de la tensión aplicada. Posibles opciones para su reemplazo son: motor paso a paso, motor brushless o motor de alta precisión. Además, la identificación realizada se basó en aplicarle escalones de tensión a la planta saturando el actuador y obteniendo grandes valores de velocidad y posición angular del cubo, cuando los controladores implementados apenas llegaron a la saturación en los primeros instantes y la zona de trabajo fue de apenas algunos grados. Esto quiere decir que la identificación realizada no fue la más congruente con la zona de trabajo elegida.

Teniendo como base la información obtenida en este proyecto se puede elegir un actuador que tenga una mejor respuesta en la zona de trabajo, y realizar la identificación utilizando señales de entrada que sean más representativas de la misma. El diseño físico del esquema experimental juega un rol muy importante y debe ser considerado con mayor detenimiento, siendo la relación entre los momentos de inercia del cuerpo de la maqueta y la rueda de reacción de gran importancia para

definir la zona de trabajo del actuador.

Por otro lado, el diseño de la maqueta como un cubo cerrado presenta el inconveniente de que la electrónica es poco accesible, y en una etapa de prototipo y desarrollo resulta una incomodidad. Se podría realizar la maqueta con una distribución diferente de manera que no se coloquen los componentes dentro de un ambiente cerrado, y no se tengan limitaciones de espacio. Esto permitiría realizar cambios y mediciones con mayor facilidad. Además, la carga de las baterías no está implementada dentro del sistema, y se deben cambiar las baterías cada vez que se agotan, lo cual puede ser mejorado agregando un módulo de carga. Posibles opciones para carga de baterías de litio de 3.7v son los circuitos integrados TP4056 o CN3791.

Adicionalmente, la utilización de electrónica modular fue una ventaja a corto plazo ya que permitió el armado del sistema sin la necesidad de diseñar una placa, pero los módulos utilizados no son muy confiables. Las mediciones presentan ruido, y el sistema en general tiene muchos factores de falla debido a las conexiones mediante cables. Además, se decidió no colocar un regulador de tensión debido a que los reguladores lineales provocan una caída de tensión y se esperaba contar con la máxima tensión disponible en las dos baterías, mientras que un regulador por conmutación provoca una fuente de ruido adicional. Esta decisión resultó no ser la más adecuada ya que en varias ocasiones el sistema no funcionó correctamente debido a caídas de tensión, que se podrían remediar mediante la colocación de un regulador.

Por otro lado, el esquema propuesto realiza el cierre del lazo de control en un ordenador externo, lo cual contribuye a una mayor versatilidad a la hora de modificar el algoritmo de control y realizar pruebas, ya que no se debe reprogramar el microcontrolador en cada caso. Sin embargo, esto es poco robusto ya que se agrega una nueva variable relacionada con la comunicación inalámbrica, y cabe la posibilidad de que no se cumpla el tiempo de muestreo debido a bloqueos o retardos en la comunicación. Si se deseara un sistema más robusto se debería cerrar el lazo directamente en el microcontrolador, utilizando la comunicación para dar las órdenes de inicio y fin de los experimentos y para obtener los datos. A su vez, existe la posibilidad de realizar la conexión cableada utilizando un conector con mecanismo de antirretorcido.

Como trabajo a futuro se propone un rediseño de la planta utilizando un nuevo actuador y electrónica más robusta, realizando un circuito impreso que integre todas las etapas del sistema, mejorando la medición de los sensores mediante filtros discretos y la estabilidad de la tensión de alimentación mediante la colocación de reguladores. También se propone realizar diseños de control más avanzados, siendo las posibilidades control robusto con rechazo a perturbaciones o control no lineal utilizando un modelo no lineal del sistema, y realizar una ampliación del sistema para que el control de posición angular sea en los tres ángulos de posición, creando un sistema de control de actitud completo.

8. Anexos

8.1. Esquema de conexiones

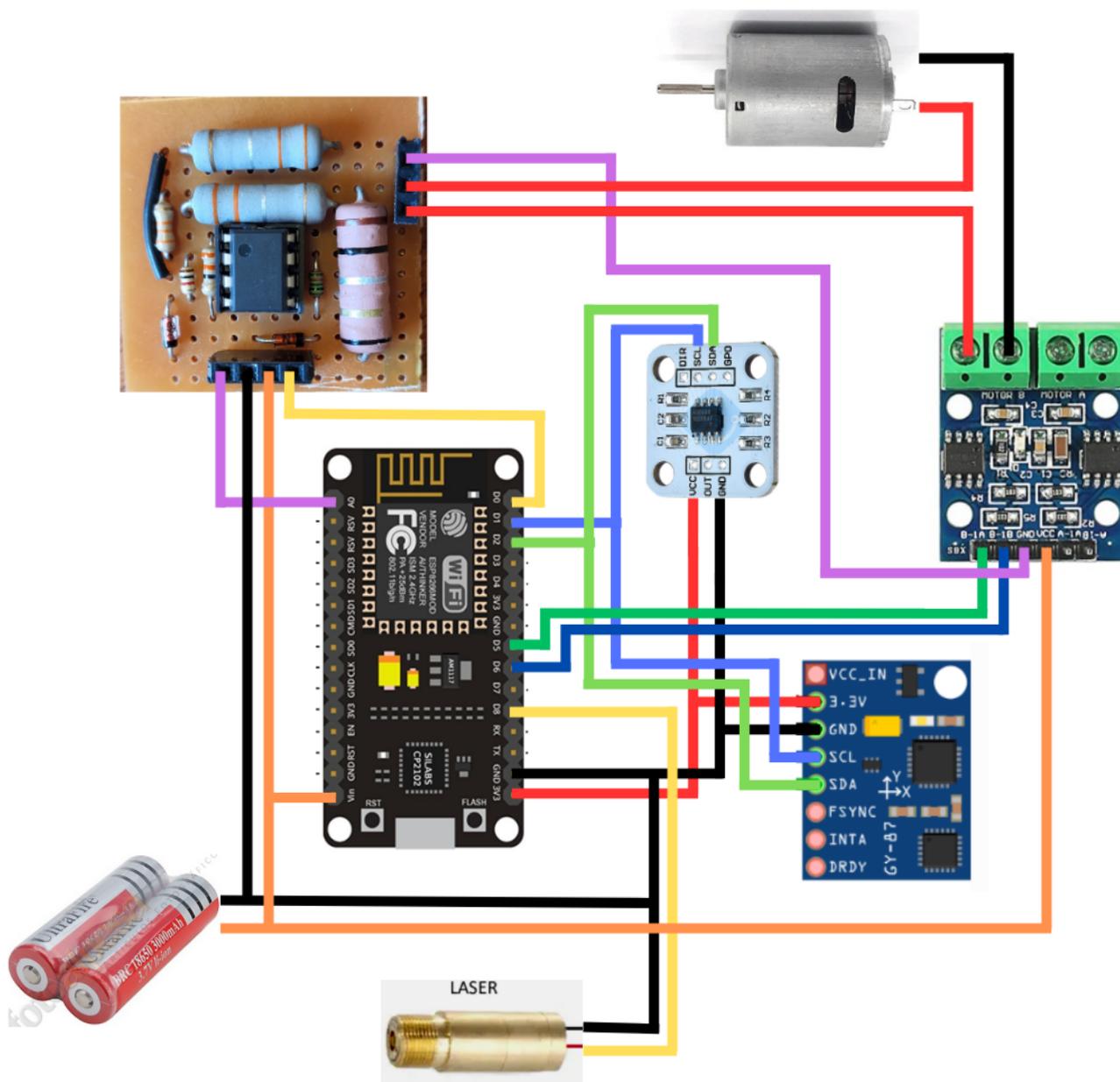


Figura 79: Esquema de conexiones

8.2. Calibración de sensores y análisis del error

Para utilizar el sensor MPU6050 por primera vez es necesario realizarle una calibración en la cual se obtengan los valores de offset que poseen tanto el acelerómetro como el giróscopo en sus tres ejes. En reposo, el sensor debería medir un valor de 0 en todos los ejes salvo en el eje z del acelerómetro, que debería medir 16384 (equivalente a 1g con sensibilidad $\pm 2g$) [2]. Utilizando el algoritmo recomendado por el fabricante, se obtuvieron los siguientes valores de offset:

Acelerómetro eje X	-3890
Acelerómetro eje Y	538
Acelerómetro eje Z	540
Giróscopo eje X	52
Giróscopo eje Y	63
Giróscopo eje Z	63

Con dichos offsets programados en los registros correspondientes del sensor se puede asegurar que no se detecte movimiento cuando el dispositivo se encuentre quieto. Se deben reprogramar cada vez que se encienda el sistema, es decir que se colocan en el bloque inicial del firmware. El dispositivo cuenta con algoritmos internos para realizar la fusión de las mediciones de los dos sensores que lo componen.

Sin embargo, se presenta el problema de que con la información de los sensores acelerómetro y giróscopo disponibles el ángulo de guiñada no se puede obtener con certeza. Los ángulos de alabeo y cabeceo se obtienen mediante el cálculo del vector de gravedad a partir de las mediciones del acelerómetro, siendo éste una referencia fija que enclava al sistema. Las mediciones del giróscopo sirven para seguir los cambios rápidos de orientación en dichos ejes, mientras que las mediciones del vector de gravedad sirven para mantener el sistema de referencia a largo plazo. En el caso del ángulo de guiñada, por otro lado, el vector de gravedad no es de utilidad y la medición obtenida a partir del algoritmo de DMP, si bien posee un filtrado previo, tiene una deriva constante. Dicha deriva se produce debido a que la medición se obtiene únicamente de la integración de la velocidad angular del giróscopo y proviene de un error de condición inicial no nulo que se suma en cada paso de tiempo de la integración.

En la imagen 80 se puede observar la deriva introducida en la lectura del dato de orientación angular durante una medición en la que se mantuvo quieto al dispositivo durante un tiempo, luego se lo rotó manualmente aproximadamente 2.5° y se lo mantuvo en esa nueva posición. El error introducido llegó a ser de más de 1° . Sin embargo, esto es solo la demostración de un caso particular y no implica que la pendiente de deriva sea siempre la medida en este experimento, ya que el error de la integración se reinicia cada vez que se vuelve a encender el dispositivo. De lo contrario, bastaría con tomar la pendiente y luego restarla al ángulo medido para obtener el ángulo real.

Se propuso mejorar la medición obtenida por la unidad de medición inercial siguiendo un esquema de fusión de sensores como los propuestos en [29], utilizando la medición del campo magnético para enclavar el ángulo de orientación del cubo en base a una referencia fija. Se consideró la propuesta de utilizar el sensor magnetómetro QMC5883L, el cual se encuentra en la misma placa que la unidad de medición inercial MPU6050, y sobre el mismo bus de comunicación.

El magnetómetro QMC5883L es un sensor que permite medir campos magnéticos con una precisión de ± 2 gauss o ± 8 gauss, con resolución de 16 bits. La medición de campo magnético sirve para detectar el campo magnético terrestre, el cual tiene una orientación determinada. Calculando el ángulo de orientación del campo magnético se puede determinar el ángulo de orientación del sensor, es decir que sirve para medir la orientación absoluta con respecto al norte magnético de la Tierra.

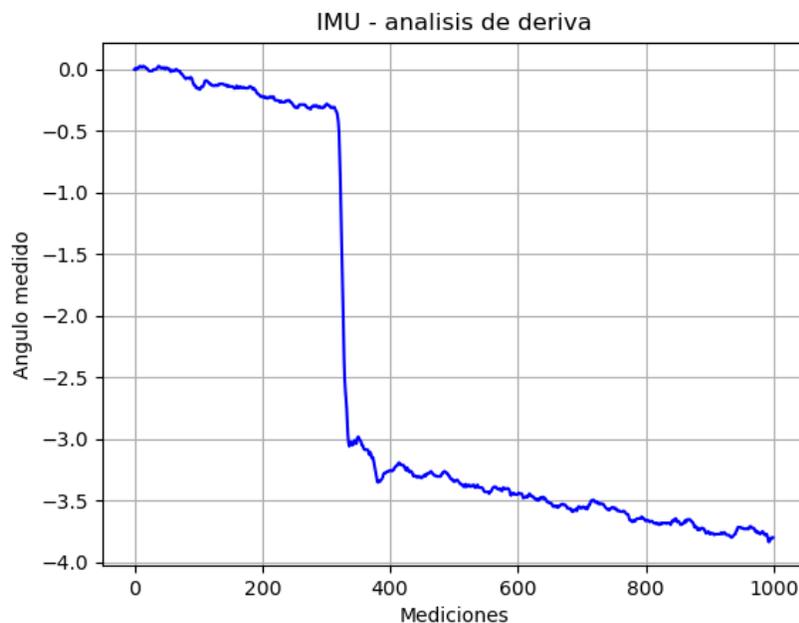


Figura 80: Ejemplo de deriva en el ángulo de la IMU lego de 1000 muestras

El manejo del sensor se realizó utilizando la librería “QMC5883L.h” de Douglas Thain [17]. Dentro del módulo HW-290, el sensor QMC5883L se encuentra conectado al bus I2C auxiliar de la IMU MPU6050, en donde puede utilizarse como esclavo en una comunicación que la IMU obre como maestro. Esto permite a la IMU tomar las mediciones de magnetómetro y colocarlas en el mismo buffer de comunicación que las mediciones propias, de manera que estén disponibles para que el microcontrolador las tome a la vez. Sin embargo, realizar las mediciones de esa manera no presenta ninguna ventaja, y tiene el problema de que la IMU debe obrar como intermediario entre el microcontrolador y el magnetómetro. Para poder habilitar una comunicación directa se debe establecer el modo Bypass en la IMU, de lo contrario no se podría acceder al sensor magnetómetro a través del bus I2C.

El magnetómetro se debe calibrar debido a que es susceptible a perturbaciones del campo magnético existentes en el espacio de trabajo donde se realizan las mediciones. Tales perturbaciones pueden ser de dos tipos, de hierros duros o de hierros blandos. Las perturbaciones de hierros duros son perturbaciones de campo magnético debido a la presencia de campos magnéticos externos que se suman o restan al campo magnético de la Tierra agregando un offset a la medición. Las perturbaciones de hierros blandos, por otro lado, son debido a la presencia de materiales ferromagnéticos cercanos, los cuales producen cambios en la magnitud y dirección del campo magnético debido a su alta permeabilidad magnética, produciendo una deformación espacial en el campo medido [14].

Si se grafica el campo magnético en función de la posición, se debería tener un máximo cuando el campo magnético está alineado con la dirección del sensor y en el sentido del campo, mínimo cuando está alineado en la misma dirección pero en sentido opuesto, y cero cuando está de forma perpendicular. Una medición de campo magnético sobre un plano espacial, si se rota al sensor en todas las direcciones posibles sobre el mismo plano, produce idealmente un gráfico en el que la relación entre el campo magnético en una dirección y el campo magnético en la dirección ortogonal es una circunferencia centrada en el origen. De esta manera, calculando el arco tangente del campo medido se puede establecer el ángulo de orientación sobre ese plano. Sin embargo, al agregar perturbaciones el gráfico que se obtiene puede ser de una elipse (debido a las deformaciones del campo de los hierros blandos) que no esté centrada en el origen (debido al offset agregado por los hierros duros). En la

figura 81 se puede ver una comparación entre casos de mediciones sin distorsión, mediciones con distorsión de hierros duros y mediciones con distorsión de hierros blandos y duros, tomada de [27]. Es indispensable que las mediciones de campo magnético se asemejen a una circunferencia centrada en el origen, ya que el cálculo del ángulo de orientación se hace a partir de tomar el arco tangente de la coordenada medida.

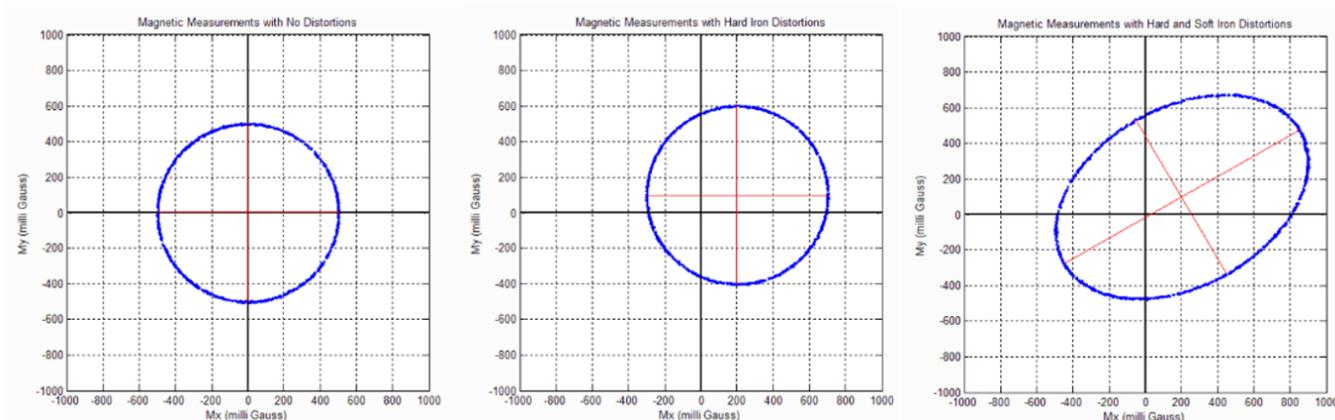


Figura 81: Ejemplos de mediciones con distorsión [27]

Para poder calibrar el magnetómetro es necesario tomar mediciones con el dispositivo montado en el lugar físico en el que se va a utilizar, debido a que los campos magnéticos pueden modificarse. Debido a esto, la calibración se realizó una vez que el prototipo se encontraba armado, con todos los módulos colocados y utilizando la comunicación inalámbrica y la alimentación por baterías. Se realizó un experimento que consistió en tomar mediciones crudas del magnetómetro en sus ejes X e Y, mientras se giraba manualmente al dispositivo de manera que se tomen todos los ángulos posibles, inicialmente tomando las mediciones con el motor apagado. En la figura 82 se puede ver el resultado de la medición, en donde se aprecia el corrimiento del origen debido a los hierros duros, y una leve deformación debido a los hierros blandos.

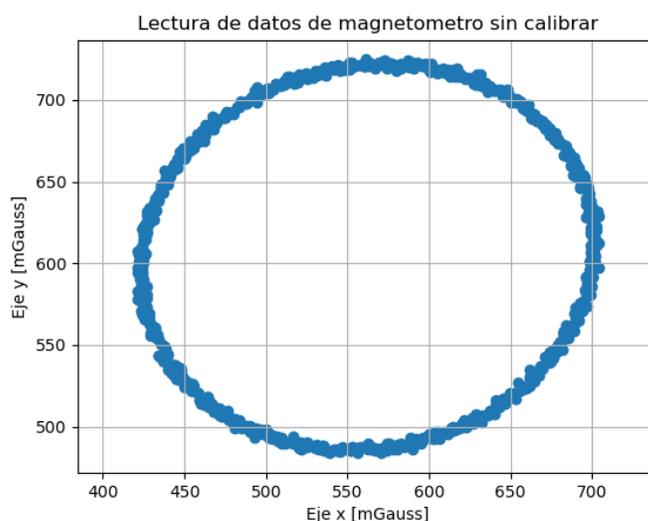


Figura 82: Medición inicial de magnetómetro

Se realizó un algoritmo para determinar la media y el factor de deformación tanto en el eje x

como en el eje y, aplicando la corrección en un nuevo experimento de medición. En la figura 83 se puede ver el resultado de la misma, en donde se eliminaron las deformaciones.

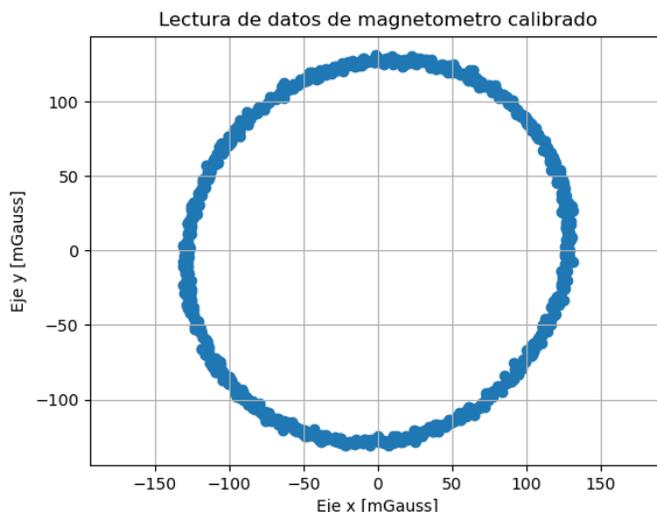


Figura 83: Datos de magnetómetro calibrados

Con esa etapa concluida, se decidió analizar el comportamiento del sensor en un caso más real de funcionamiento. Para ello se repitió el experimento de medición pero manteniendo el motor encendido durante todo el experimento. En la figura 84 se pueden ver los resultados, en donde se grafican las mediciones del nuevo experimento en verde, sobre las mediciones del experimento anterior para poder hacer una comparativa visual.

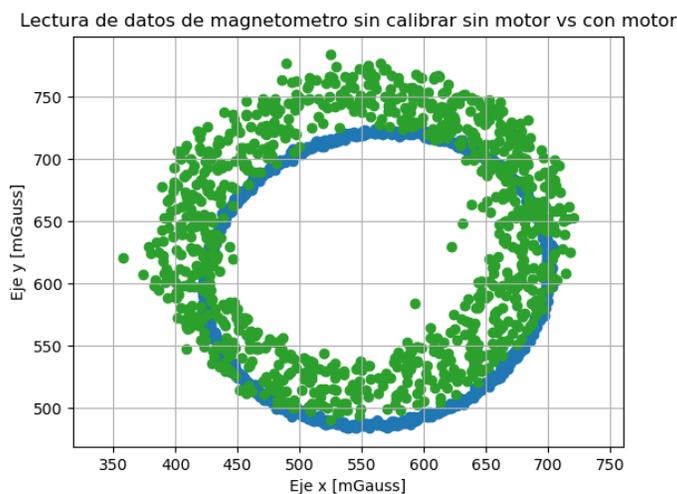


Figura 84: Comparación entre mediciones con y sin motor encendido

Se observó que las mediciones mostraban un desplazamiento del punto central, es decir que con el motor encendido se sumó un valor medio diferente al que se había tomado con los experimentos sin encender el motor, resultando en un error en la exactitud. Además, se observó una dispersión muy marcada de los datos, con lo cual la medición tomada resulta muy precisa. Esto puede deberse al ruido magnético que se produce en los inductores del motor al circular corriente por los mismos, y a los imanes permanentes del motor que cambian su posición relativa al sensor.

Se decidió rehacer la calibración tomando mediciones con el motor encendido como datos de partida, de manera que se encuentre el nuevo valor medio y se puedan centrar las mediciones. Los datos calibrados nuevamente pueden verse en la figura 85.

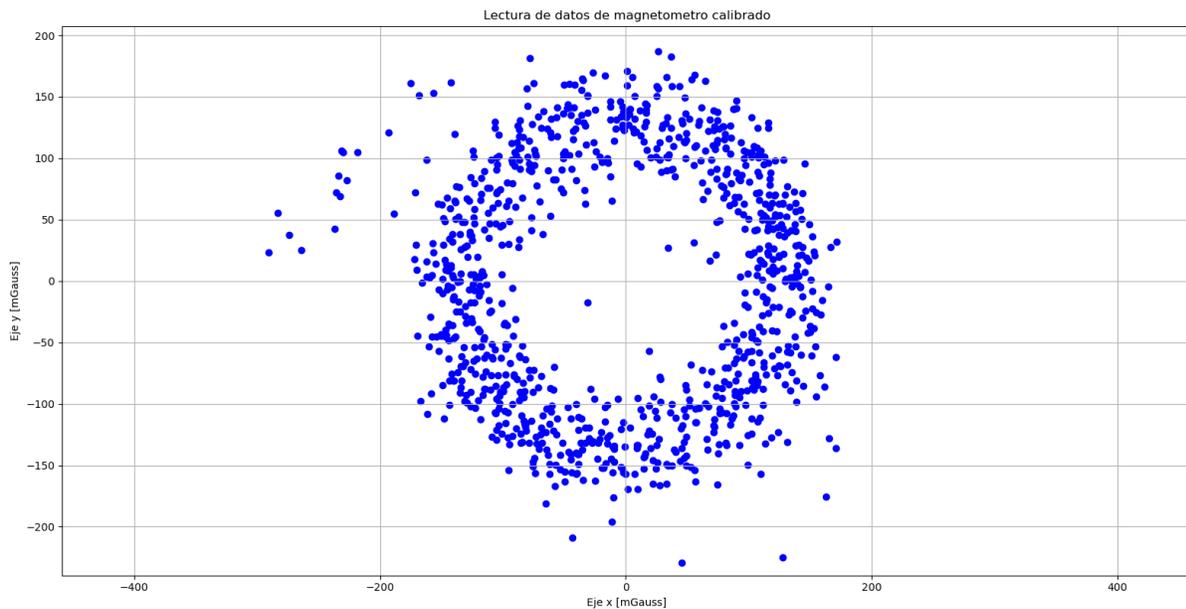


Figura 85: Datos de magnetómetro calibrado con el motor encendido

Para analizar el efecto que produce la dispersión de los datos crudos en la medición de posición angular se realizó el cálculo del ángulo a partir de los datos, utilizando la siguiente fórmula:

$$\theta_{mag} = \tan^{-1} \left(\frac{y}{x} \right)$$

Se realizó un experimento de medición en el que se encendió el motor pero se mantuvo fijo al cubo, para que la posición no cambie debido a la acción del torque de la rueda de reacción. Se mantuvo al cubo apuntando en un ángulo 0, luego se lo movió manualmente hasta 2.5° y por último hasta 5°. En la figura 86 se puede ver el resultado temporal de la experiencia, comparando las mediciones tomadas por el magnetómetro y por la unidad de medición inercial.

Se puede ver que la magnitud del error es muy significativa comparada con los valores de ángulo que se intenta medir. La medición de la IMU es libre de ruido y presenta una lectura más estable. Para intentar minimizar el error se propuso implementar un filtro de tipo pasabajos de frecuencia de corte 2rad/seg, ya que se analizó previamente que la planta posee una frecuencia mínima de aproximadamente 0.5rad/seg. En la figura 87 se puede ver el filtro aplicado sobre la medición, con un resultado que no fue satisfactorio.

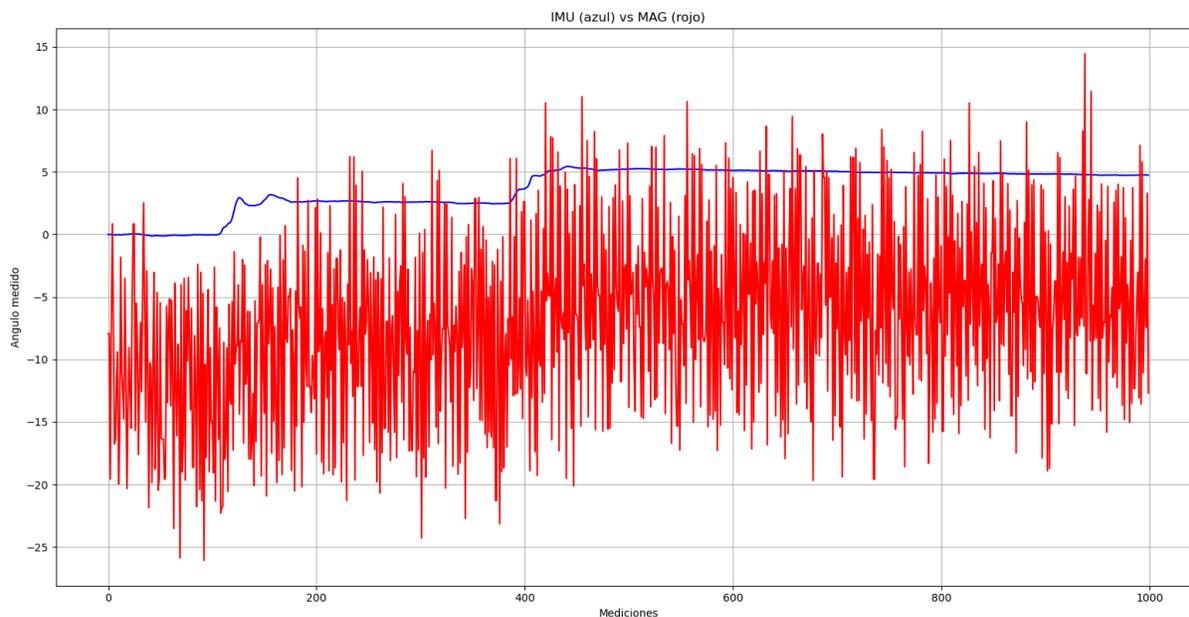


Figura 86: Comparación entre valor de ángulo obtenido con los sensores

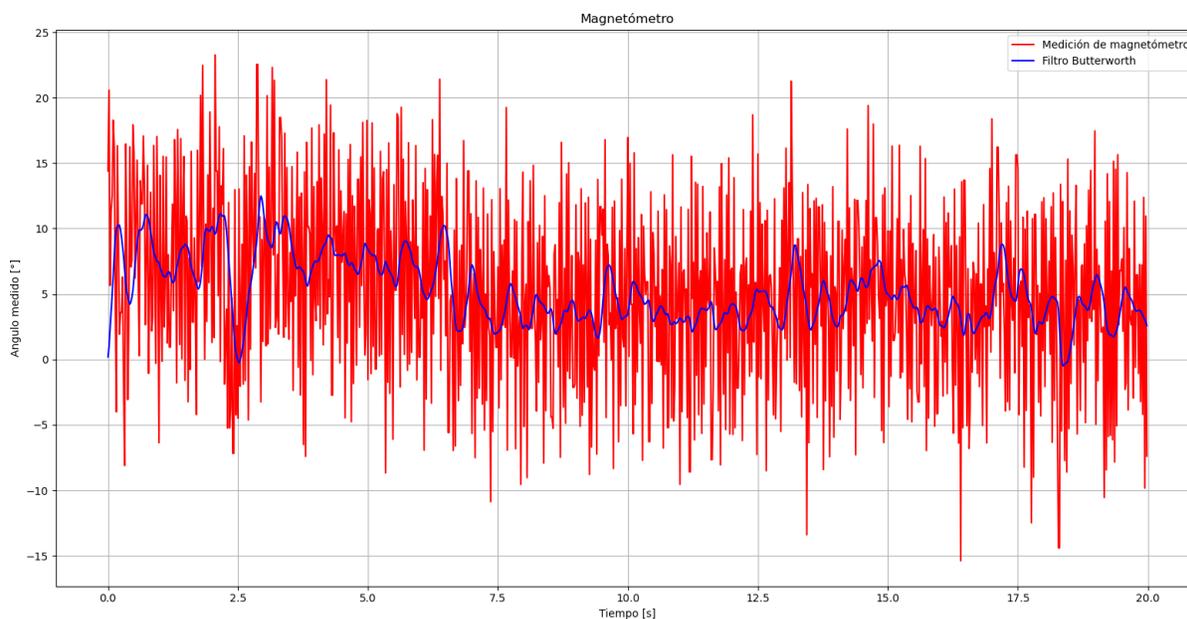


Figura 87: Aplicación de filtro sobre medición de ángulo

En base al trabajo desarrollado se tomaron las siguientes conclusiones:

- La medición de ángulo utilizando los datos que provee la unidad de medición inercial es una medición limpia y presenta una lectura estable, pero no es confiable debido a la deriva. La medición de ángulo provista por el magnetómetro, por otro lado, no es limpia y presenta mucha variación en los valores que se obtienen, con errores muy grandes relativos al valor medido.
- Se esperaba contar con la ventaja de la confiabilidad a largo plazo en la lectura provista por el magnetómetro. Sin embargo, esto no se puede garantizar debido a que la calibración del sensor

es muy sensible a los cambios en los campos magnéticos de los elementos circundantes, como lo son los imanes del motor y el imán de neodimio colocado para la lectura de velocidad angular de la rueda de reacción. Al cambiar la posición del sensor con respecto a esos imanes se cambia el punto de la calibración de hierros duros. Además, no se puede asegurar que la calibración hecha en el laboratorio sea válida si se utiliza el dispositivo en otro ambiente, como puede ser otro laboratorio situado en otro edificio. Es decir que la calibración realizada no es robusta.

- En el proyecto se utiliza la unidad de medición inercial para simular un caso posible de aplicación del control por rueda de reacción, utilizando componentes sencillos y económicos. El error de deriva es un error inherente a la medición utilizando sensor giróscopo, pero el objetivo del proyecto no es centrarse en la corrección de dicho error sino centrarse en el control de la planta.
- El tiempo objetivo de los experimentos de control utilizando el prototipo se estima en alrededor de un minuto. En lapsos de tiempo tan breves se consideró que el error introducido por el magnetómetro, incluso luego de filtrarlo, es mucho mayor que el introducido por la unidad de medición inercial.
- Al no haber podido solucionar el problema utilizando la tecnología disponible, se consideró que el alcance del proyecto no contemplaba un análisis exhaustivo de las técnicas existentes de medición angular o de filtrado de señales para poder solucionar el error de otro modo y se decidió continuar con el esquema del proyecto descartando la medición con magnetómetro.

8.3. Código

Firmware Arduino: config.h

```

1 #include <ESP8266WiFi.h>
2
3 const char* ssid      = "redLAN";
4 const char* password = "clave";
5 const char* hostname = "ESP8266_1";
6 IPAddress ip(192,168,1,200);
7 IPAddress gateway(192,168,1, 1);
8 IPAddress subnet(255, 255, 255, 0);

```

WiFi_UDP.hpp

```

1 #include <ESP8266WiFi.h>
2 #include <WiFiUDP.h>
3 #include "config.h"
4
5 //Declaración de variables
6 WiFiUDP UDP;
7 unsigned int localPort = 8888;
8 unsigned int remotePort = 8889;
9 char packetBuffer[UDP_TX_PACKET_MAX_SIZE];
10 int data = 0;
11
12 void ConnectWiFi_STA(bool useStaticIP = false){
13     WiFi.mode(WIFI_STA);
14     if(useStaticIP) WiFi.config(ip, gateway, subnet);
15     WiFi.begin(ssid, password);
16     while (WiFi.status() != WL_CONNECTED){
17         delay(100);
18     }
19 }
20
21 void ConnectUDP(){
22     while (UDP.begin(localPort) != 1){
23         delay(1000);
24     }
25 }
26
27 void ProcessPacket(String response){
28     data = response.toInt();
29 }
30
31 void SendUDP_Packet(String content){
32     UDP.beginPacket(UDP.remoteIP(), remotePort);
33     UDP.write(content.c_str());
34     UDP.endPacket();
35 }
36
37 bool GetUDP_Packet() {
38     int packetSize = UDP.parsePacket();
39     if (packetSize != 0){

```

```

40     UDP.read(packetBuffer, UDP_TX_PACKET_MAX_SIZE);
41     ProcessPacket((char*)packetBuffer);
42     for (int i = 0; i < packetSize; i++){
43         packetBuffer[i]=' '; //limpio el buffer
44     }
45     return true;
46 }
47 else
48     return false;
49 }

```

Sensores.hpp

```

1  #include <Wire.h>
2  #include <math.h>
3  #include <MPU6050_6Axis_MotionApps20.h>
4  #include <I2Cdev.h>
5  #include <AS5600.h>
6
7  //Definicion de pines
8  #define D_INPUT 16// DO = GPIO16
9  #define A_INPUT A0// ADC0=A0
10
11 #define TO_MA (3.3*1000)/(1023*17*0.165) //Conversion de cuentas a mA
12
13 //Factores de calibracion para IMU
14 #define OFFSET_ACCEL_X -3890
15 #define OFFSET_ACCEL_Y 538
16 #define OFFSET_ACCEL_Z 540
17 #define OFFSET_GYRO_X 52
18 #define OFFSET_GYRO_Y 63
19 #define OFFSET_GYRO_Z 63
20
21 #ifndef M_PI
22 #define M_PI 3.14159265358979323846264338327950288
23 #endif
24
25 //Definicion de objetos
26 AS5600 as5600;
27 MPU6050 mpu;
28
29 // MPU control/status vars
30 bool dmpReady = false; // set true if DMP init was successful
31 uint8_t mpuIntStatus; // holds actual interrupt status byte from MPU
32 uint8_t devStatus; // return status after each device operation(0 = success)
33 uint16_t packetSize; // expected DMP packet size (default is 42 bytes)
34 uint16_t fifoCount; // count of all bytes currently in FIFO
35 uint8_t fifoBuffer[64]; // FIFO storage buffer
36
37 //Variables para las mediciones
38 float yaw;
39 int16_t gyro[3];
40 int motor_speed;
41 float mA;
42

```

```

43 void inicializarMPU6050(){
44     mpu.initialize();
45     mpu.setI2CBypassEnabled(false);
46     devStatus = mpu.dmpInitialize();
47     // Valores de calibracion
48     mpu.setXAccelOffset(OFFSET_ACCEL_X);
49     mpu.setYAccelOffset(OFFSET_ACCEL_Y);
50     mpu.setZAccelOffset(OFFSET_ACCEL_Z);
51     mpu.setXGyroOffset(OFFSET_GYRO_X);
52     mpu.setYGyroOffset(OFFSET_GYRO_Y);
53     mpu.setZGyroOffset(OFFSET_GYRO_Z);
54     // Activar DMP
55     if (devStatus == 0){
56         mpu.setDMPEnabled(true);
57         dmpReady = true;
58         // get expected DMP packet size for later comparison
59         packetSize = mpu.dmpGetFIFOPacketSize();
60     }
61     else{
62         dmpReady = false;// ERROR!
63     }
64 }
65
66 void leerDMPFIFO(){
67     if (mpu.dmpGetCurrentFIFOPacket(fifoBuffer)){
68         mpu.dmpGetYaw(&yaw, fifoBuffer);
69         mpu.dmpGetGyro(gyro, fifoBuffer); //xyz
70     }
71 }
72
73 void inicializarAS5600(){
74     as5600.begin();
75     as5600.setDirection(AS5600_CLOCK_WISE);
76     as5600.setSlowFilter(0);
77     as5600.setFastFilter(4);
78 }
79
80 void leerAS5600(){
81     motor_speed = as5600.getAngularSpeed(AS5600_MODE_RPM);
82 }
83
84 void leerCorriente(){
85     mag = analogRead(A_INPUT);
86     mA = mag * TO_MA - 43.7; //le saco el offset
87     if (!digitalRead(D_INPUT)){
88         mA = -mA;
89     }
90 }

```

Firmware_giroscoPIO.ino

```

1 #include "Sensores.hpp"
2 #include "WiFi_UDP.hpp"
3
4 #define BIA 12 //D5 = GPI014

```

```
5 #define BIB 14 //D6 = GPIO12
6 #define LASER 15 //D8 = GPIO15
7
8 void sendPacket(){
9     String message = String(yaw * 180 / M_PI, 3); //Posicion en grados, 3 decimales
10    message += ",";
11    message += String(motor_speed); //velocidad en RPM
12    message += ",";
13    message += String(gyro[2]); //Gyro eje Z en grados/seg
14    message += ",";
15    message += String(mA, 3); //Corriente en mA, 3 decimales
16    SendUDP_Packet(message);
17 }
18
19 void setup(){
20     /*Definicion de pines*/
21     pinMode(BIA, OUTPUT);
22     pinMode(BIB, OUTPUT);
23     digitalWrite(BIA, 0);
24     digitalWrite(BIB, 0);
25     analogWriteRange(1023); //Rango PWM
26     analogWriteFreq(40000); //Frecuencia PWM
27     pinMode(LASER, OUTPUT);
28     digitalWrite(LASER,0);
29     pinMode(D_INPUT, INPUT);
30     /*Inicialización de sensores comunicación*/
31     Wire.begin();
32     Wire.setClock(50000);
33     delay(200);
34     inicializarMPU6050();
35     inicializarAS5600();
36     ConnectWiFi_STA(true);
37     ConnectUDP();
38     Wire.setClock(400000); //400kHz I2C clock
39     yaw = 0;
40     data = 0;
41 }
42
43 void loop(){
44     if (!dmpReady){
45         return; // Si fallo al iniciar, parar programa
46     }
47     leerAS5600();
48     delayMicroseconds(100);
49     leerDMPFIFO();
50     leerAS5600();
51     leerCorriente();
52     if (GetUDP_Packet()){
53         sendPacket();
54         if (data <= 1023 && data > 0){
55             analogWrite(BIA, data);
56             digitalWrite(BIB, 0);
57         }
58         else if (data >= -1023 && data < 0){
59             analogWrite(BIB, -data);
60             digitalWrite(BIA, 0);
61         }
62     }
```

```

61     }
62     else if (data == 2000){ //2000 para prender el laser
63         digitalWrite(LASER, 1);
64     }
65     else if (data == 2001){ //2001 para apagar el laser
66         digitalWrite(LASER, 0);
67     }
68     else if (data == 0) {
69         digitalWrite(BIA, 0);
70         digitalWrite(BIB, 0);
71     }
72     else { //apaga todo
73         digitalWrite(BIA, 0);
74         digitalWrite(BIB, 0);
75         digitalWrite(LASER, 0);
76     }
77 }
78 }

```

Aplicación Python: Control_cubo_giroscopo.py

```

1 import time
2 import socket
3 import numpy as np
4 from scipy import signal
5 from bisect import bisect_left
6 import os
7 import psutil
8
9 """Establecimiento de prioridad de proceso"""
10 process = psutil.Process(os.getpid())
11 if os.name == 'nt':#Windows
12     process.nice(psutil.HIGH_PRIORITY_CLASS)
13 elif os.name == 'posix':#Linux/MacOS
14     process.nice(-20)
15
16 """Linealización de la acción del motor"""
17 file = open('LookUp_Table_Linear.dat', 'r')
18 lista = np.array(file.readlines())
19 i = 0
20 N = 34
21 lup_table = np.zeros((2, N))
22 for linea in lista:
23     uu = lista[i].split(" ")
24     for k in range(0, N):
25         lup_table[i][k] = float(uu[k])
26     i += 1
27 file.close()
28 def lookup(x, xs, ys):
29     if x <= xs[0]: return ys[0]
30     if x >= xs[-1]: return ys[-1]
31     if x < 0.001 and x > -0.001 : return 0
32     i = bisect_left(xs, x)
33     k = (x - xs[i - 1]) / (xs[i] - xs[i - 1])

```

```

34     y = k * (ys[i] - ys[i - 1]) + ys[i - 1]
35     return y
36 def linealizar(x):
37     if x < -0.001:#Mitad negativa
38         return lookup(x * 0.7 - 0.3, lup_table[1, 0:17], lup_table[0, 0:17])
39     elif x > 0.001:#Mitad positiva
40         return lookup(x * 0.7 + 0.3, lup_table[1, 17:34], lup_table[0, 17:34])
41     else:
42         return 0
43
44 """Inicializacion de la comunicacion"""
45 UDP_IP = '192.168.1.200'
46 UDP_PORT_RECV = 8889
47 UDP_PORT_SEND = 8888
48 sock_recv = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
49 sock_recv.bind(('', UDP_PORT_RECV))
50 sock_recv.settimeout(0.0199) #timeout por si no le llega el mensaje
51 mensaje_recv = "0,0,0,0,0"
52 mensaje_send = str(2000) #prender laser
53 sock_send = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
54 sock_send.sendto(bytes(mensaje_send, "utf-8"), (UDP_IP, UDP_PORT_SEND))
55 try:
56     sock_recv.recv(1024)
57 except:
58     pass
59
60 """Definicion de parametros de la experiencia"""
61 Dt = 20 #Segundos total
62 file1 = 'temp.dat' #Archivo en el que se guardan los datos
63 Ts = 0.02 #Tiempo de muestreo
64 N = int(Dt / Ts)
65 N2 = 6 #Tiempo | Control | Ang_imu | Vel_imu | RPM_motor | Corriente
66 Datos = np.zeros((N, N2)) #Matriz para guardar los datos
67
68 """Filtro para la medicion del encoder"""
69 cutoff = 2
70 normal_cutoff = cutoff / (0.5 / Ts)
71 b, a = signal.butter(2, normal_cutoff, analog = False, output = 'ba')
72 zi = signal.lfilter_zi(b, a) * 0
73 rpm_unfilt = np.zeros(N)
74
75 """Calculo del offset inicial"""
76 k = 0
77 media_imu = 0
78 N_media = 250 #Para la media inicial 5 segundos
79 input("Coloque el laser en la posicion inicial y presione enter")
80 while k < N_media:
81     ti = time.perf_counter()
82     sock_send.sendto(bytes(mensaje_send, "utf-8"), (UDP_IP, UDP_PORT_SEND))
83     try:
84         data = sock_recv.recv(1024)
85         mensaje_recv = data.decode('utf-8')
86     except:
87         pass
88     ang_imu, rpm, gyro, ma = mensaje_recv.split(",")
89     media_imu += float(ang_imu)

```

```

90     k = k + 1
91     while (time.perf_counter() - ti) < Ts:
92         pass
93 media_imu /= N_media
94
95 """Experimento loop principal"""
96 k = 0
97 ang_prev = 0
98 vueltas = 0
99 set_point = 0
100 u = 0
101 error_k1 = 0
102 u_k1 = 0
103 input("Para empezar presione enter") #Para sincronizar la comunicacion
104 sock_send.sendto(bytes(mensaje_send, "utf-8"), (UDP_IP, UDP_PORT_SEND))
105 try:
106     data = sock_recv.recv(1024)
107     mensaje_rcv = data.decode('utf-8')
108 except:
109     pass
110 t_inicial = time.perf_counter() #Tiempo inicial del experimento
111 while k < N:
112     #Tiempo inicial de la iteracion
113     ti = time.perf_counter()
114     #Lectura de datos
115     ang_imu, rpm, gyro, ma = mensaje_rcv.split(",")
116     #Filtro para encoder
117     rpm_unfilt[k] = int(rpm)
118     if (k > len(a) - 1):
119         filtered, zi = signal.lfilter(b, a, rpm_unfilt[k - (len(a) - 1) : k + 1], axis = -1, zi =
120             ↪ zi)
121         filt = filtered[len(a) - 1]
122     else:
123         filt = int(rpm)
124     #Llevo la cuenta del angulo absoluto
125     ang = (float(ang_imu) - media_imu) + vueltas * 360
126     if (ang - ang_prev) > 180:
127         ang -= 360
128         vueltas -= 1
129     elif (ang - ang_prev) < -180:
130         ang += 360
131         vueltas += 1
132     #Set point y algoritmo de control
133     if (k*Ts) >= 1:
134         set_point = 20
135     error = set_point - ang
136     u = - 0.1534 * error + 0.1502 * error_k1 + 0.8862 * u_k1
137     #Mantengo variables para el siguiente paso
138     ang_prev = ang
139     error_k1 = error
140     u_k1 = u
141     #Restriccion y linealizacion de u
142     if u > 1.0:
143         u = 1.0
144     elif u < -1.0:
145         u = -1.0

```

```
145 u_lineal = linealizar(u)
146 #Envio mensaje
147 mensaje_send = str(int(u_lineal * 1023))
148 sock_send.sendto(bytes(mensaje_send, "utf-8"), (UDP_IP, UDP_PORT_SEND))
149 #Recibo mensaje
150 try:
151     data = sock_recv.recv(1024)
152     mensaje_recv = data.decode('utf-8')
153 except:
154     pass
155 #Guardado de datos
156 Datos[k][0] = time.perf_counter() - t_inicial #Tiempo
157 Datos[k][1] = float(u) #Variable u usada en ESP
158 Datos[k][2] = ang #Angulo absoluto de la imu grados
159 Datos[k][3] = -int(gyro) #Angulos/seg del cubo
160 Datos[k][4] = filt #Revoluciones filtradas RPM
161 Datos[k][5] = float(ma) #Corriente del motor mA
162 #Establecimiento del tiempo de muestreo
163 k = k + 1
164 while (time.perf_counter() - ti) < Ts:
165     pass
166
167 """Finalización del programa"""
168 sock_send.sendto(bytes('1900', "utf-8"), (UDP_IP, UDP_PORT_SEND))#apaga laser y motores
169 header = "Tiempo | Control recibido | Ang_imu | Vel_imu | RPM | Corriente | Señal utilizada | N =
→ " + str(N) + " TS = " + str(Ts)
170 np.savetxt(file1, Datos, fmt = '%f', header = header) #guardo en archivo
171 sock_recv.close()
172 sock_send.close()
```

8.4. Manual de usuario

A continuación, se presenta un manual de usuario para poder utilizar el proyecto “Giróscopo controlado por rueda de reacción” de manera académica en clases universitarias relacionadas con el estudio de control e implementación. Particularmente se propone para clases de Laboratorio de Control de las carreras de Ingeniería del Instituto Balseiro.

Programación de ESP8266:

Para poder utilizar el dispositivo en una nueva red LAN se le deberá programar tanto el nombre de la red como la clave. Para ello se debe abrir el proyecto “Firmware_giroscopio” dentro del software Arduino IDE, y modificar los campos correspondientes en el fichero “config.h”. Dentro de ese mismo archivo es posible modificar la dirección IP que se le asignará al dispositivo. No es recomendable modificar otras características del firmware, aunque es posible.

Para programar la tarjeta de desarrollo se utiliza el software Arduino IDE, utilizando el paquete de ESP8266. Para descargar el mismo se debe seguir el siguiente procedimiento: Dentro del IDE, seleccionar *Archivo*→*Preferencias*, y dentro de la casilla “*Gestor de URLs Adicionales de Tarjetas*” agregar: http://arduino.esp8266.com/stable/package_esp8266com_index.json.

Esto permite que luego en el gestor de tarjetas se pueda encontrar la opción para descargar todo el paquete de ESP8266 para Arduino, lo cual se hace seleccionando *Herramientas*→*placa*→*Gestor de tarjetas* y luego instalar el paquete “*esp8266 by ESP8266 Community*”. Una vez hecho esto seleccionar *Herramientas*→*placa*→*NodeMCU 1.0 ESP12E Module*.

Una vez hecho el paso anterior, se conecta el dispositivo utilizando un cable microUSB, y se carga el nuevo firmware. Tal como se ve en la figura 88, no es necesario desarmar el dispositivo para poder acceder a la placa y reprogramarla.

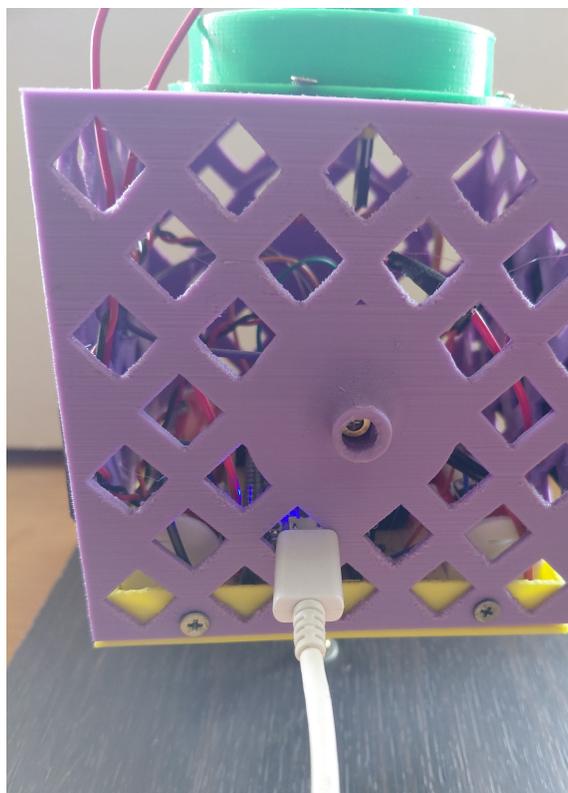


Figura 88: Conexión con el dispositivo para reprogramación de firmware

Utilización de programa Python:

El programa realizado en Python para la comunicación y el control del dispositivo se llama "Control_cubo_giroscopo.py". Al abrir el mismo en un entorno de trabajo de Python, se pueden modificar los parámetros de la experiencia que se desea realizar y de la ley de control propuesta. No es recomendable modificar el algoritmo de la aplicación.

Entre las líneas 50 a 58 se encuentran los parámetros de la experiencia que se pueden modificar, los cuales son el tiempo total del experimento en segundos y las especificaciones del archivo en donde se guardan los datos. Luego, entre las líneas 122 a 130, dentro del loop principal, se encuentra el establecimiento del punto de trabajo y el algoritmo de control. Allí se podrá definir el formato del experimento que se desea realizar al establecer el punto de trabajo deseado para el dispositivo, y se podrá utilizar cualquier algoritmo que se haya desarrollado. A modo de ejemplo se encuentra el algoritmo del controlador de adelanto de fase propuesto en este proyecto.

Para utilizar el dispositivo se debe encender la llave para darle alimentación, y esperar al menos 30 segundos para asegurar que se establezca la comunicación con la red LAN y que se inicialicen correctamente los sensores. Luego se puede ejecutar el programa en Python, el cual inicialmente envía una señal de mando que enciende el led láser, de manera que se puede apuntar el frente del dispositivo hacia algún punto de referencia que se desee. Se le pedirá por consola que mantenga el dispositivo quieto y presione la tecla 'Enter'. El dispositivo se tomará 5 segundos para medir el ángulo y establecerlo como su ángulo de referencia. Finalmente, se le pedirá que presione nuevamente la tecla 'Enter' para ingresar en el loop principal que realizará el experimento con los parámetros que se le hayan configurado.

Cambio de baterías:

Para cambiar las baterías del dispositivo se deben quitar los tornillos que están en el borde inferior del cubo, como se muestra en la figura 89. Son cuatro tornillos, dos en el frente y dos en el dorso. Luego, retirar la tapa hacia arriba 3cm e inclinarla hacia el frente (la cara con el puntero láser) de manera que se pueda acceder a los portapilas. Para apoyar la tapa es recomendable hacerlo sobre una caja que tenga la misma altura que la base giratoria del cubo, de manera que los cables que están unidos tanto a la tapa como a la base no soporten tensiones durante el proceso, como se muestra en la figura 90.

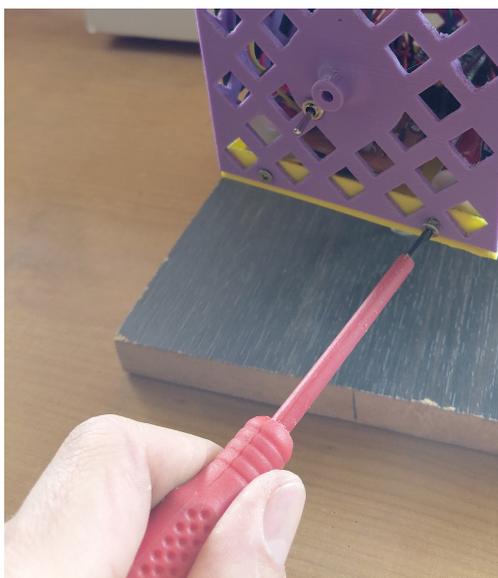


Figura 89: Detalle del quitado de los tornillos

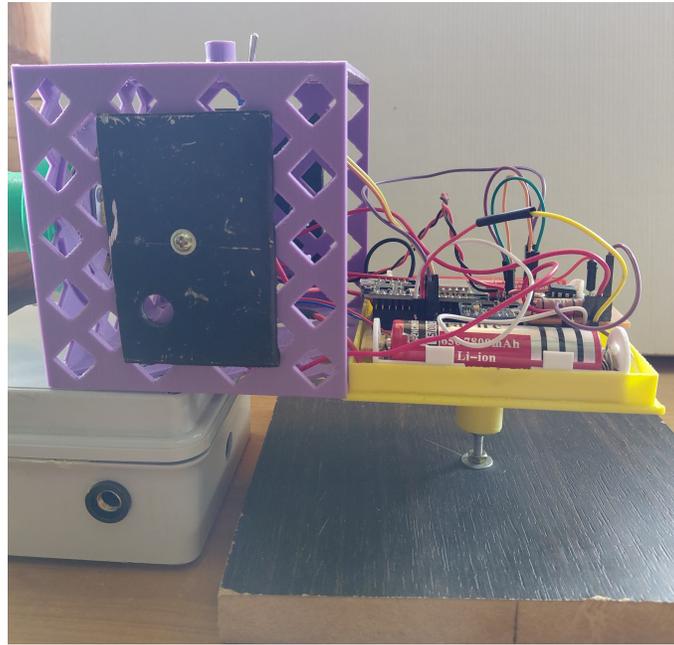


Figura 90: Ejemplo de soporte para la tapa

El dispositivo utiliza dos baterías recargables de litio 18650, que tienen 4.2v de tensión cuando se encuentran en el nivel máximo de carga, 3.7v de tensión nominal de trabajo y <3.5v cuando se encuentran descargadas. En la figura 91 se ve la disposición de las mismas, ubicadas a los lados en portapilas individuales.

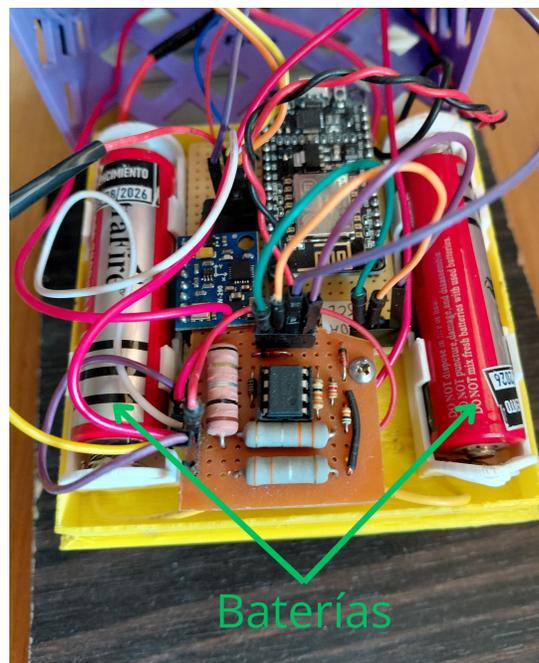


Figura 91: Ubicación de las baterías en el dispositivo

Referencias

- [1] A. Corrales A. Candelas. Giroscopios en el sistema gypsygyro-18. *Publicación interna No. 4., Grupo de Automática , Robótica y Visión Artificial, Universidad de Alicante*, 2007.
- [2] *Invensense MPU Hardware Offset Registers Application Note*, 2014.
- [3] asm OSRAM AS5600 position sensor <https://ams-osram.com/products/sensor-solutions/position-sensors/ams-as5600-position-sensor>.
- [4] R. M. Blenden. Regenerative power optimal reaction wheel attitude control. Master's thesis, University of Colorado, 2011.
- [5] Alan N. Castellon-López. Plataforma experimental que emula la dinamica de orientación de nanosatélites. *Publicación Semestral Páidi Vol.12 No. Especial 4*, 2024.
- [6] Charles Grassin, Reaction Wheel Attitude Control <https://charleslabs.fr/en/project-Reaction+Wheel+Attitude+Control>.
- [7] CyberMotor WITTESTEIN Rueda de reacción Cyber®2 para satélites pico y nano <https://cyber-motor.wittenstein.de/en-en/reaction-wheel-satellite/>.
- [8] ESP8266 Datasheet https://www.espressif.com/sites/default/files/documentation/0a-esp8266ex_datasheet_en.pdf.
- [9] Vincent Francois-Lavet. Study of passive and active attitude control systems for the oufti nanosatellites. Master's thesis, University of Liège Faculty of Applied Sciences, 2010.
- [10] Bernard Friedland. *CONTROL SYSTEM DESIGN An Introduction to State-Space Methods*. McGraw-Hill, 1986.
- [11] M.E. Salgado G.C. Goodwin, S.F. Graebe. *CONTROL SYSTEM DESIGN*. Prentice Hall,, 2000.
- [12] Abbas Emami-Naeini Gene F. Franklin, J. David Powell. *Feedback control of dynamic systems 7th Ed.* Pearson, 2014.
- [13] Jako Gerber. A 3-axis attitude control system hardware design for a cubesat. Master's thesis, University of Stellenbosch, 2014.
- [14] Github Kris Winer Simple and Effective Magnetometer Calibration <https://github.com/kriswiner/MPU6050/wiki/Simple-and-Effective-Magnetometer-Calibration>.
- [15] Github Rob Tillaart AS5600 <https://github.com/RobTillaart/AS5600>.
- [16] Github Jeff Rowberg MPU6050 <https://github.com/jrowberg/i2cdevlib/tree/master/Arduino/MPU6050>.
- [17] Github Douglas Thain QMC5883L <https://github.com/dthain/QMC5883L>.
- [18] Lennart Ljung. *System Identification Theory for the User Second Edition*. PRENTICE HALL PTR, 1999.

- [19] Fredric William Long. Design and testing of a nanosatellite simulator reaction wheel attitude control system. Master's thesis, Utah State University, 2014.
- [20] MATLAB Documentation <https://la.mathworks.com/help/matlab/index.html>.
- [21] Invensense MPU6050 <https://invensense.tdk.com/products/motion-tracking/6-axis/mpu-6050/>.
- [22] NayLamp Mechatronics <https://naylampmechatronics.com/blog>.
- [23] Rocket Lab Ruedas de Reacción <https://www.rocketlabusa.com/space-systems/satellite-components/reaction-wheels/>.
- [24] Marina A. Samuels. The design and testing of a three-degree-of-freedom small satellite simulator using a linear controller with feedback linearization and trajectory generation. Master's thesis, Utah State University, 2014.
- [25] Python SciPy documentation <https://docs.scipy.org/doc/scipy/>.
- [26] MatLab SimuLink <https://la.mathworks.com/products/simulink.html>.
- [27] VectorNav Educational Material <https://www.vectornav.com/resources/inertial-navigation-primer/specifications-and-error-budgets/specs-hsicalibration>.
- [28] ZAYAS GATO Francisco QUINTIÁN PARDO Héctor JOVE PÉREZ Esteban CASTELEIRO ROCA José Luis y CALVO ROLLE José Luis. *Diseño de controladores PID*. Servizo de Publicacións - Universidade da Coruña, 2020.
- [29] Milton David Pujos Yanzapanta. *Diseño e implementación de una unidad de medición inercial IMU embebida en base a un sistema microcontrolado*. PhD thesis, Escuela Politécnica Nacional-Ecuador, 2016.