

DISEÑO E IMPLEMENTACIÓN DE UN CONTROLADOR DE VUELO PARA UN
VEHÍCULO AÉREO AUTÓNOMO NO TRIPULADO (UAV) PARROT MAMBO

JAVIER MAURICIO ZARAZA ESPINOSA
EDGAR JULIAN FARID BUITRAGO GALVAN

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FÍSICOMECÁNICAS
ESCUELA DE INGENIERÍA MECÁNICA
BUCARAMANGA

2023

DISEÑO E IMPLEMENTACIÓN DE UN CONTROLADOR DE VUELO PARA UN
VEHÍCULO AÉREO AUTÓNOMO NO TRIPULADO (UAV) PARROT MAMBO

JAVIER MAURICIO ZARAZA ESPINOSA
EDGAR JULIAN FARID BUITRAGO GALVAN

Trabajo de grado para optar al título de Ingeniero Mecánico

Director

Jose Jorge Carreño Zagarra
PhD. Ingeniería Electrónica

Codirectores

Helio Sneyder Esteban Villegas
MSc. Ingeniería Mecánica
Diana Katheryn Poveda Rodríguez
MSc. Ingeniería Electrónica

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FÍSICOMECÁNICAS
ESCUELA DE INGENIERÍA MECÁNICA
BUCARAMANGA

2023

DEDICATORIA

Dedico los frutos de este proyecto a mi familia. Especialmente, a mis padres Javier y Alexandra, quienes estuvieron a mi lado en los momentos difíciles. Su apoyo inquebrantable me enseñó a enfrentar los desafíos sin perder la calma ni rendirme en el proceso.

También quiero dedicarle este trabajo a mi abuela Rosa, quien me acompañó durante todo este el camino, asegurándose de mi bienestar en todo momento. Su confianza en mí me motivaba cada día y cada noche.

Y, finalmente, a mis amigos, a los que creyeron en mí, con su actitud lograron que tomará más impulso, convirtiendo cada obstáculo en una oportunidad de crecimiento y cada logro en un motivo para celebrar juntos.

Javier Mauricio Zaraza Espinosa

DEDICATORIA

A mis padres, Orlando y Alba, quienes han sido mi fuente inagotable de apoyo y guía a lo largo de mi vida. El sacrificio y la confianza que depositaron en mí han sido el motor detrás de mis éxitos.

A mi hermano Cristian, mi eterno compañero y mejor amigo. Tu compañía ha sido un valioso apoyo.

A mis queridos hermanos y tíos, cuya presencia y aliento constante han dejado una influencia significativa en cada etapa de mi travesía.

A mi querida abuela Graciela, cuya sabiduría y cariño han sido una fuente de fortaleza.

A mis colegas, ingenieros mecánicos: Wilman, Mauricio, Camila, Andrés, Javier y Juan quienes han dejado una marca en mi vida. Deseo que los años venideros sean tan gratificantes para nosotros como los que hemos compartido.

Edgar Julian Farid Buitrago Galvan

AGRADECIMIENTOS

Quiero expresar mi sincero agradecimiento a las personas que desempeñaron un papel crucial en la realización de este proyecto de grado. En primer lugar, mi más profundo agradecimiento va dirigido a mi compañero de tesis Julian y Codirector de Tesis, Helio, cuya orientación experta, dedicación y sabios consejos fueron esenciales para llevar a cabo esta investigación con éxito. Agradezco también a todos mis Profesores y Profesoras, cuyas enseñanzas y experiencia contribuyeron de manera significativa a mi crecimiento académico.

Expreso mi gratitud a la Universidad Industrial de Santander por brindarme acceso a sus recursos y por acogerme en su comunidad académica. Durante mi tiempo en la universidad, esta institución desempeñó un papel fundamental en mi crecimiento tanto a nivel profesional como personal.

No puedo olvidar a mi Familia, especialmente a mis padres y abuelas, cuyo amor incondicional, paciencia y apoyo constante me han impulsado a superar desafíos. Su fe en mí ha sido un motor inagotable de motivación. Por último, mis Amigos han sido un faro de luz en los momentos de estudio y presión, brindándome risas, ánimo y apoyo constante. Cada uno de ustedes ha dejado una huella significativa en este logro, y les estoy profundamente agradecido por su contribución a mi éxito académico.

Javier Mauricio Zaraza Espinosa

AGRADECIMIENTOS

A Leonidas Vásquez Chaparro, distinguido docente que me brindó consejo y ayuda en mi trayecto académico.

A mi compañero de tesis Mauricio y codirector Helio por sus contribuciones esenciales en la realización de este proyecto.

A cada familiar, profesor, amigo y ser querido que haya compartido este viaje a mi lado. Sus enseñanzas, apoyo y compañía han sido un regalo en mi camino académico y personal.

A la vida, por todas las experiencias maravillosas que he tenido el privilegio de vivir. Cada momento ha sido una lección invaluable y una fuente de crecimiento en mi camino.

Edgar Julian Farid Buitrago Galvan

CONTENIDO

	pág.
INTRODUCCIÓN	18
1. OBJETIVOS	20
1.1 OBJETIVO GENERAL	20
1.2 OBJETIVOS ESPECÍFICOS	20
2. ESTADO DEL ARTE	21
2.1 HISTORIA DE LOS AVANCES EN EL CONTROL APLICADOS A UAV's.....	21
2.2 PRECEDENTES DE TRABAJOS CON EL MINIDRON PARROT MAMBO	24
3. MARCO TEÓRICO	28
3.1 MODELADO DINÁMICO DEL CUADRICÓPTERO	28
3.1.1 Orientación del Cuadricóptero	30
3.1.2 Modelo Aerodinámico para configuración en X.....	35
3.2 MODELO DE UN PENDULO BARRA CON ECUACIÓN DE EULER-LAGRANGE .	42
3.3 CONTROL PID.....	44
3.4 CONTROL ROBUSTO LQG.....	46
3.4.1 Control óptimo LQR.....	46
3.4.2 Filtros Kalman	49
3.5 ÍNDICES DE DESEMPEÑO	50
3.5.1 (ISE) Criterio de la integral del error al cuadrado.....	51
3.5.2 (ITSE) Criterio de la integral del error cuadrado multiplicado por el tiempo	51
3.5.3 (IAE) Criterio de la integral del valor absoluto del error	51
3.5.4 (ITAE) Criterio de la integral del valor absoluto del error multiplicado por el tiempo	52
4. METODOLOGÍA	53
4.1 IDENTIFICACIÓN Y VALIDACIÓN DE LOS PARÁMETROS DEL MODELADO DINÁMICO DEL PARROT MAMBO	54
4.1.1 Masa y distancia entre los motores y el centro de gravedad	55

4.1.2 Inercias	55
4.1.3 Coeficiente de empuje.....	57
4.1.4 Coeficiente de torque	58
4.2 ARQUITECTURA DE CONTROL EN CASCADA.....	59
4.3 SELECCIÓN DEL ENFOQUE DE CONTROL ROBUSTO	60
4.4 DISEÑO DEL CONTROLADOR LQG	61
4.4.1 Representación del espacio de estado.....	61
4.4.2 Linealización del modelo	62
4.4.3 Filtros Kalman	64
4.4.4 Cálculo de K con la Ponderación de Q y R	65
5. IMPLEMENTACIÓN.....	67
5.1 HARDWARE Y SOPORTE PARROT MINIDRONES DE SIMULINK.....	67
5.1.1 Minidron Parrot mambo	67
5.1.2 Soporte de Parrot minidrones de Simulink	72
5.1.3 Descripción del modelo de simulación	75
5.1.3.1 Command	76
5.1.3.2 Sensor.....	77
5.1.3.3 Environment.....	79
5.1.3.4 Airframe	80
5.1.3.5 Visualization	81
5.1.3.6 Flight Control System	81
5.2 ESTIMACIÓN DE LOS PARÁMETROS	83
5.2.1 Estimación de inercias	83
5.2.2 Estimaciones de coeficientes de empuje y torque	85
5.3 IMPLEMENTACIÓN DEL CONTROLADOR LQG	89
5.3.1 Sensor data group.....	90
5.3.2 Estimator	90
5.3.3 Controller	94
5.3.3.1 Yaw	94
5.3.3.2 XY to reference orientation.....	95

5.3.3.3 Attitude	95
5.3.3.4 Altitude	96
5.3.3.5 Controlmixer	97
5.3.3.6 ThrustsToMotorCommands	97
5.4 PRUEBAS EXPERIMENTALES	98
5.4.1 Escenario de pruebas	98
5.4.2 Pruebas de seguimiento de trayectoria	101
5.4.3 Cálculo índices de desempeño	102
RESULTADOS.....	103
6.1 MODELADO E IDENTIFICACIÓN DE PARÁMETROS DEL SISTEMA DINÁMICO	103
6.1.1 Momento de inercia del dron en X, I_{xx}	103
6.1.2 Momento de inercia del dron en Y, I_{yy}	104
6.1.3 Momento de inercia del dron en Z, I_{zz}	104
6.1.4 Coeficiente de empuje.....	105
6.1.5 Coeficiente de torque	107
6.1.6 Modelo Propuesto	107
6.2 DESEMPEÑO DEL CONTROLADOR SIMULADO VS EXPERIMENTAL	108
CONCLUSIONES	125
RECOMENDACIONES	127
BIBLIOGRAFÍA.....	128
ANEXOS.....	132

LISTA DE CUADROS

pág.

Cuadro 1. Parámetros Soporte Parrot minidrones de Simulink	54
Cuadro 2. Valores de equilibrio de los estados y entradas	63
Cuadro 3. Inercia teórica y experimental de la barra	85
Cuadro 4. Inercia I_{xx} del Soporte Parrot minidrones de Simulink y experimental	103
Cuadro 5. Inercia I_{yy} del Soporte Parrot minidrones de Simulink y experimental	104
Cuadro 6. Inercia I_{zz} del Soporte Parrot minidrones de Simulink y experimental	105
Cuadro 7. Comparación entre las rpm experimentales y teóricas motor posterior derecho	106
Cuadro 8. Comparación entre las rpm experimentales y teóricas motor posterior izquierdo	106
Cuadro 9. % Error producido del coeficiente de empuje de Simulink y el propuesto. ...	106
Cuadro 10. % Error producido del coeficiente de torque de Simulink y el propuesto. ...	107
Cuadro 11. Parámetros propuestos	108
Cuadro 12. PID Real vs PID Simulado en trayectoria cuadrada	115
Cuadro 13. LQG Real vs LQG Simulado en trayectoria cuadrada	115
Cuadro 14. PID Simulado vs LQG Simulado en trayectoria cuadrada	116
Cuadro 15. PID Real vs LQG Real en trayectoria cuadrada	116
Cuadro 16. PID Real vs PID Simulado en trayectoria circular	123
Cuadro 17. LQG Real vs LQG Simulado en trayectoria circular	123
Cuadro 18. PID Simulado vs LQG Simulado en trayectoria circular	123
Cuadro 19. PID Real vs LQG Real en trayectoria circular	124

LISTA DE FIGURAS

	pág.
Figura 1. Parrot AR Dron	22
Figura 2. DJI Phantom 4	23
Figura 3. Simulador 3D Simulink Support Package for Parrot Minidrones	25
Figura 4. DCL del dron Parrot Mambo y sistemas de referencia.	29
Figura 5. Rotación de un cuerpo libre.	32
Figura 6. Esquema péndulo barra	42
Figura 7. Estructura del controlador PID	45
Figura 8. Sistema de control LQR	49
Figura 9. Ejemplo filtro Kalman	50
Figura 10. Estructura metodológica.....	53
Figura 11. Esquema del sistema péndulo dron	57
Figura 12. Diagrama general del control del dron.....	60
Figura 13. Diagrama general de la aplicación de los filtros Kalman.....	65
Figura 14. Controlador LQR para Attitude	66
Figura 15. Dron Parrot Mambo y accesorios.	67
Figura 16. Motor A "Anti-Clock" coreless.....	68
Figura 17. Motor C "Anti-Clock" coreless.	69
Figura 18. Parrot Mambo SIP6 Linux Motherboard con 800MHz ARM A9.	69
Figura 19. MPU6050 Acelerómetro y giroscopio.	70
Figura 20. Parte inferior del dron Parrot mambo.	71
Figura 21. Hardware Setup.	74
Figura 22. Modelo asbquadcopter.....	76
Figura 23. Subsistema Command.	77
Figura 24. Subsistema Sensors.	78
Figura 25. Sensors (Dynamics).	78
Figura 26. Subsistema Environment.	79

Figura 27. Constantes Environment.	79
Figura 28. Subsistema Airframe.	80
Figura 29. Modelo lineal dron.	80
Figura 30. Subsistema Visualization.	81
Figura 31. Subsistema Flight Control System.	82
Figura 32. Montaje del sistema péndulo barra dron para el eje x.	83
Figura 33. Montaje del sistema péndulo barra dron para el eje y.	83
Figura 34. Montaje del sistema péndulo barra dron para el eje z.	84
Figura 35. Tacómetro DT-2234B.	85
Figura 36. Montaje medición RPM.	86
Figura 37. CAD Adaptador de base.	87
Figura 38. Celda de carga 1kg.	87
Figura 39. Modulo Transmisor HX711.	87
Figura 40. Calibración celda de carga.	88
Figura 41. Montaje de torque.	88
Figura 42. Montaje de medición de torque.	89
Figura 43. Interior del subsistema Flight Control System.	89
Figura 44. Subgrupo Sensor data group.	90
Figura 45. Subgrupo Estimator.	91
Figura 46. Filtro Kalman en la estimación de velocidades X y Y.	92
Figura 47. Filtro Kalman en la estimación de posición y velocidad en Z.	93
Figura 48. Filtro Kalman en la estimación de posición en X y Y.	93
Figura 49. Subgrupo Controller.	94
Figura 50. Bloque Yaw.	95
Figura 51. Bloque XY to reference orientation.	95
Figura 52. Bloque Attitude con controlador LQR.	96
Figura 53. Bloque Attitude con controlador PID.	96
Figura 54. Bloque Altitude.	97
Figura 55. Bloque Controlmixer.	97
Figura 56. Bloque ThrustsToMotorCommands.	98

Figura 57. Patrón de suelo 1	99
Figura 58. Patrón de suelo 2	99
Figura 59. Patrón de suelo 3	100
Figura 60. Patrón de suelo 4	100
Figura 61. Trayectoria Circular Diámetro 1 m.....	101
Figura 62. Trayectoria Cuadrada Lado 1 m.....	102
Figura 63. Resultado del Grey Box para el cálculo de la inercia I_{xx}	103
Figura 64. Resultado del Grey Box para el cálculo de la inercia I_{yy}	104
Figura 65. Resultado del Grey Box para el cálculo de la inercia I_{zz}	104
Figura 66. Sentido de giro de los motores.....	105
Figura 67. Vista isométrica y superior de la trayectoria cuadrada simulada LQG	108
Figura 68. Datos posición de vuelo de la trayectoria cuadrada simulada LQG	109
Figura 69. Datos orientación de vuelo de la trayectoria cuadrada simulada LQG.....	109
Figura 70. Vista isométrica y superior de la trayectoria cuadrada experimental LQG ..	110
Figura 71. Datos posición de vuelo de la trayectoria cuadrada experimental LQG	110
Figura 72. Datos orientación de vuelo de la trayectoria cuadrada experimental LQG ..	111
Figura 73. Vista isométrica y superior de la trayectoria cuadrada simulada PID.....	112
Figura 74. Datos posición de vuelo de la trayectoria cuadrada simulada PID	112
Figura 75. Datos orientación de vuelo de la trayectoria cuadrada simulada PID	113
Figura 76. Vista isométrica y superior de la trayectoria cuadrada experimental PID ...	113
Figura 77. Datos posición de vuelo de la trayectoria cuadrada experimental PID	114
Figura 78. Datos orientación de vuelo de la trayectoria cuadrada experimental PID ...	114
Figura 79. Vista isométrica y superior de la trayectoria circular simulada LQG	117
Figura 80. Datos posición de vuelo de la trayectoria circular simulada LQG	117
Figura 81. Datos orientación de vuelo de la trayectoria circular simulada LQG	118
Figura 82. Vista isométrica y superior de la trayectoria circular experimental LQG	118
Figura 83. Datos posición de vuelo de la trayectoria circular experimental LQG	119
Figura 84. Datos orientación de vuelo de la trayectoria circular experimental LQG	119
Figura 85. Vista isométrica y superior de la trayectoria circular simulada PID	120
Figura 86. Datos posición de vuelo de la trayectoria circular simulada PID	120

Figura 87. Datos orientación de vuelo de la trayectoria circular simulada PID	121
Figura 88. Vista isométrica y superior de la trayectoria circular experimental PID	121
Figura 89. Datos posición de vuelo de la trayectoria circular experimental PID	122
Figura 90. Datos orientación de vuelo de la trayectoria circular experimental PID	122

LISTA DE ANEXOS

	pág.
Anexo A. Linealización y cálculo de K LQR.....	132
Anexo B. Grey Box estimación de parámetros.	134
Anexo C. Gráficas y cálculo de índices de desempeño.	136
Anexo D. Código balanza.....	148

RESUMEN

TÍTULO: DISEÑO E IMPLEMENTACIÓN DE UN CONTROLADOR DE VUELO PARA UN VEHÍCULO AÉREO AUTÓNOMO NO TRIPULADO (UAV) PARROT MAMBO*

AUTOR: JAVIER MAURICIO ZARAZA ESPINOSA, EDGAR JULIAN FARID BUITRAGO GALVAN. **

PALABRAS CLAVE: CONTROL ROBUSTO, LQG, LQR, FILTROS KALMAN, PID, UAV, SIMULINK, MATLAB, PARROT MAMBO.

DESCRIPCIÓN:

En el presente trabajo de investigación se diseña y pone a prueba un sistema de control robusto de vuelo para el dron Parrot Mambo, tanto en un entorno simulado como real. Este trabajo emplea la teoría de control LQG e implementa el soporte Parrot minidrones de Simulink. En los últimos años, se ha despertado un gran interés en la investigación con sistemas aéreos no tripulados, esto se debe a su notable capacidad para maniobrar y múltiples campos de aplicación. Por esta razón, surge el reto de establecer un control confiable y preciso que garantice operaciones exitosas en las diferentes aplicaciones de los drones.

El objetivo principal de este proyecto de investigación es desarrollar un sistema de control para el dron Parrot Mambo. Esto implica una serie de pasos clave, que incluyen la caracterización de los parámetros del sistema dinámico con un método mixto (experimental y Grey Box). Luego, se diseña un controlador basado en la teoría de control robusto LQG para posteriormente implementarlo y comparar el rendimiento del controlador con trayectorias cuadradas y circulares, tanto en simulaciones como en condiciones reales.

De acuerdo con el análisis realizado al sistema de control robusto se concluyó que este trabajo presenta un avance con el establecimiento de un control más confiable y preciso aplicado a un dron real. Además, este trabajo investigativo reestableció los parámetros fundamentales del dron Parrot Mambo.

*Trabajo de grado

** Facultad de Ingeniería Físico-mecánicas. Escuela de Ingeniería Mecánica. Director: Phd. Jose Jorge Carreño Zagarra. Codirectores: Msc. Helio Sneyder Esteban Villegas, Msc. Diana Katheryn Poveda Rodríguez.

ABSTRACT

TITLE: DESIGN AND IMPLEMENTATION OF A FLIGHT CONTROLLER FOR AN AUTONOMOUS UNMANNED AERIAL VEHICLE (UAV) PARROT MAMBO. *

AUTHOR: JAVIER MAURICIO ZARAZA ESPINOSA, EDGAR JULIAN FARID BUITRAGO GALVAN. **

KEY WORDS: ROBUST CONTROL, LQG, LQR, KALMAN FILTERS, PID, UAV, SIMULINK, MATLAB, PARROT MAMBO.

DESCRIPTION:

In this research work, a robust flight control system for the Parrot Mambo drone is designed and tested in both a simulated and real environment. This work employs LQG control theory and implements the Parrot minidrones support of Simulink. In recent years, there has been a great interest in research with unmanned aerial systems, this is due to their remarkable maneuverability and multiple fields of application. For this reason, the challenge arises to establish reliable and accurate control to ensure successful operations in the different applications of drones.

The main objective of this research project is to develop a control system for the Parrot Mambo drone. This involves a series of key steps, including the characterization of the dynamic system parameters with a mixed method (experimental and Grey Box). Then, a controller based on LQG robust control theory is designed and subsequently implemented to compare the controller performance with square and circular trajectories, both in simulations and in real conditions.

According to the analysis conducted on the robust control system, it was concluded that this work represents progress with the establishment of a more reliable and precise control applied to a real drone. Furthermore, this research work redefined the fundamental parameters of the Parrot Mambo drone.

* Bachelor's Degree Thesis

** Faculty of Physicomechanical Engineering. School of Mechanical Engineering. Director: Phd. Jose Jorge Carreño Zagarra. Co-directors: Msc. Helio Sneyder Esteban Villegas, Msc. Diana Katheryn Poveda Rodríguez.

INTRODUCCIÓN

El desarrollo de vehículos aéreos no tripulados (UAV) ha experimentado un rápido avance impulsado por el desarrollo tecnológico. Son ampliamente empleados en diversas aplicaciones, tales como labores de búsqueda y rescate en áreas afectadas por desastres naturales, vigilancia y monitoreo, mediciones topográficas y geodésicas, observaciones meteorológicas y climatológicas, así como también en el ámbito del transporte de mercancías. Los drones se han vuelto cada vez más populares en diversas industrias, como la agricultura, la construcción, la cinematografía y la inspección de infraestructuras. Estos robots voladores ofrecen una amplia gama de aplicaciones y beneficios, pero todavía existen desafíos en términos de rendimiento y estabilidad. En este contexto, surge la necesidad de establecer un control confiable y preciso que garantice operaciones exitosas en las diversas aplicaciones de los UAV.

Los drones multirrotores han despertado un gran interés debido a sus ventajas, como el despegue y aterrizaje vertical, la capacidad de mantener una posición estacionaria en el aire y su libertad de movimiento, sin embargo, lograr un control preciso en estos drones implica tener en cuenta que el modelo dinámico es no lineal y se deben implementar técnicas de control robusto para que así se pueda cumplir con las diversas aplicaciones del mercado, es por eso que los objetivos de este proyecto de investigación son caracterizar, desarrollar un sistema de control robusto de vuelo, implementarlo y validar su desempeño.

Para lograr este objetivo, se llevará a cabo una investigación y análisis de los componentes y características del dron Parrot Mambo. Se realizarán pruebas de vuelo y se recolectarán datos para la posterior modelización y simulación del sistema. A partir de esta información, se diseñará un controlador LQG que permita mejorar la estabilidad y precisión del dron en diversas condiciones de vuelo. Una vez diseñado el controlador, se procederá a su implementación en el dron y se realizarán pruebas de validación para evaluar su desempeño. Estas pruebas se llevarán a cabo en diferentes escenarios y se

compararán los resultados obtenidos con los parámetros de referencia establecidos previamente. De esta manera, y mediante el uso de índices de desempeño se podrá determinar la eficacia y eficiencia del controlador LQG implementado y la comparación con su contraparte PID.

1. OBJETIVOS

1.1 OBJETIVO GENERAL

Desarrollar un sistema de control robusto de vuelo para un Dron Parrot Mambo empleando principios de diseño de software basado en modelos con el fin de contribuir con la misión y visión de la escuela de ingeniería mecánica en la investigación, desarrollo y transferencia de tecnologías.

1.2 OBJETIVOS ESPECÍFICOS

- Modelar e identificar los parámetros del sistema dinámico de un vehículo aéreo autónomo no tripulado (UAV) Parrot Mambo para su control de vuelo.
- Diseñar el controlador basado en la teoría de control robusto para el seguimiento de trayectorias del dron.
- Implementar el sistema de control de vuelo en el UAV Parrot Mambo.
- Comparar el desempeño del controlador simulado en Matlab/Simulink y de forma experimental.

2. ESTADO DEL ARTE

En este capítulo, se abordará la historia de los avances en el control aplicado a UAV Cuadcopter, así como los precedentes de trabajos relacionados con el minidron Parrot Mambo. Se explorarán los hitos y desarrollos significativos en el campo del control de drones. Además, se revisarán trabajos previos que han utilizado el minidron Parrot Mambo, examinando las contribuciones y aprendizajes obtenidos. Este análisis de la historia y los precedentes sentará las bases para comprender el contexto y la relevancia de nuestro propio estudio en el campo del control de drones.

2.1 HISTORIA DE LOS AVANCES EN EL CONTROL APLICADOS A UAV's

Los cuadricópteros, también conocidos como drones cuadricópteros, son vehículos aéreos no tripulados (VANT) que están equipados con cuatro rotores principales para la sustentación y el control. Estos dispositivos han experimentado avances significativos en los últimos años y han encontrado una amplia variedad de aplicaciones en diferentes campos.

Los primeros artículos públicos específicamente de drones de tipo cuadricópteros datan de la década de los años 80's. En ellos se llevaron a cabo investigaciones centrándose en el comportamiento de un cuadricóptero con una carga suspendida y vientos cruzados¹, además, también se trataron estudios de cómo la turbulencia atmosférica afectaba a un dirigible de carga pesada de tipo cuadricóptero². En estas investigaciones se obtuvieron

¹ NAGABHUSHAN, B. L. y TOMLINSON, N. P. Dynamics and Control of a Heavy Lift Airship Hovering in a Turbulent Cross Wind. En: Journal of Aircraft [en línea]. Octubre, 1982. vol. 19, no. 10 [consultado 11 septiembre 2023], p. 826-830. ISSN 1533-3868. Disponible en: <https://doi.org/10.2514/3.61564>

² TISCHLER, Mark B. y JEX, Henry R. Effects of atmospheric turbulence on a quadrotor heavy lift airship. En: Journal of Aircraft [en línea]. Diciembre, 1983. vol. 20, no. 12 [consultado 11 septiembre 2023], p. 1050-1057. ISSN 1533-3868. Disponible en: <https://doi.org/10.2514/3.48211>

resultados que destacan la importancia de cómo el vehículo y la carga se afectan mutuamente cuando enfrentan el viento y las órdenes de control.

Según la base de datos de Scopus, después de esta década, se publicaron algunos artículos sobre el tema, pero no fue hasta mediados de los años 2000's que la investigación en torno a los drones cuadricópteros experimentó un rápido crecimiento y se disparó en términos de interés y desarrollo.

En 2006, se otorgaron los primeros permisos comerciales de drones por parte de la FAA (Administración Federal de Aviación) en Estados Unidos. En esta época también se adelantaron la interconexión de tecnologías. La empresa francesa Parrot lanzó en el año 2010 el "Parrot AR Dron", un dron que se podía controlar mediante Wi-Fi y un teléfono inteligente³. Tecnología que ha sido adoptada hasta la actualidad por su gran versatilidad y la facilidad que ofrece a los usuarios para controlar este tipo de dispositivos.

Figura 1. Parrot AR Dron



Fuente: IFIXIT [En línea]. [Recuperado en 2023]. Disponible en: <https://es.ifixit.com/Desmontaje/Parrot+AR.Drone+Teardown/3984>

³ OLIVER, Estefania. Te mostramos la historia de los drones con lujos y detalles. En: Digitaltrends [En línea]. 2018. [Consultado 08 junio 2023]. Disponible en: <https://es.digitaltrends.com/drones/la-historia-de-los-drones/>

En 2016, DJI, uno de los principales fabricantes de drones del mercado, lanzó el Phantom 4, que introdujo una característica revolucionaria: la visión inteligente basada en computadora y la tecnología de aprendizaje automático. Esta innovación permitió al dron evitar obstáculos de manera inteligente y realizar un seguimiento y capturar imágenes de personas, animales u objetos, en lugar de simplemente seguir una señal de GPS. Este avance significativo representó un antes y un después importante tanto para la fotografía con drones como para los drones de consumo en general⁴. Uno de los mayores problemas de los drones hasta ese momento era la dificultad para controlarlos, se requería de un piloto con cierto nivel de experiencia para manejar de manera confiable uno de estos dispositivos. La probabilidad de estrellarse o perder el control ahora se ve reducida gracias a estas características de prevención de obstáculos, proporcionando de esta manera un mayor acceso a un público que no cuenta con tanta experiencia.

Figura 2. DJI Phantom 4



Fuente: DJI [En línea]. [Recuperado en 2023]. Disponible en: <https://www.dji.com/global/phantom-4>

En la actualidad, se estima que el mercado global de drones tiene un valor de alrededor de 100 mil millones de dólares. Esta industria se puede clasificar en tres categorías principales según sus objetivos. En primer lugar, los drones utilizados para fines de defensa y militares representan aproximadamente el 70% del consumo total. En segundo lugar, los drones enfocados en el ámbito comercial ocupan alrededor del 13% del

⁴ Ibid., p.20.

mercado. Por último, los drones destinados al consumo personal constituyen el 17% restante⁵. Ahora es muy común que los drones tengan incorporados gran cantidad de sensores con la finalidad de establecer técnicas de control que garanticen vuelos confiables, seguros, precisos y eficientes para todo tipo de industrias que requieran de sus habilidades para tareas determinadas.

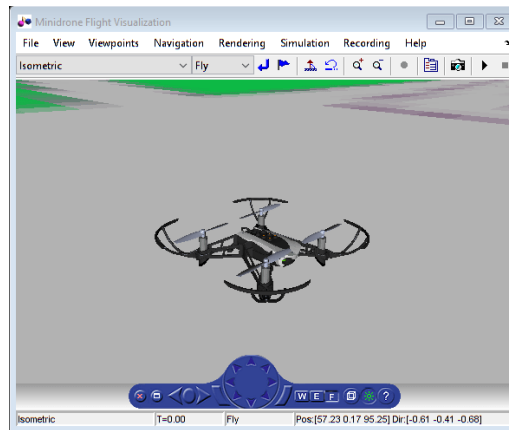
2.2 PRECEDENTES DE TRABAJOS CON EL MINIDRON PARROT MAMBO

El Parrot Mambo se lanzó al mercado en septiembre del año 2016. Este resultó ser muy atractivo para aplicar diseño de controladores a nivel académico debido a sus diversos sensores integrados y el paquete de soporte para Simulink que salió tiempo después. El dron contaba con sensores como acelerómetros, giroscopios, sensores de presión y sensores ultrasónicos, que permitían capturar datos precisos del entorno y del propio dron durante el vuelo. Además, el paquete de soporte para Simulink brindaba a los estudiantes la capacidad de utilizar Matlab como herramienta de diseño de controladores y análisis de datos. Matlab es ampliamente utilizado en el campo de la ingeniería y proporciona una amplia gama de herramientas y funciones para el diseño y la simulación de controladores.

Con la combinación de los sensores del Parrot Mambo y el soporte de Simulink los usuarios podían diseñar y probar algoritmos de control avanzados para el dron, lo que resultaba especialmente útil para propósitos educativos y de investigación en el campo de la robótica y la ingeniería de control.

⁵ Goldman Sachs. Drones: Reporting for Work. En: The verge [Sitio Web]. [Consultado 08 junio 2023]. Disponible en: <https://www.theverge.com/sponsored/goldman-sachs-drones/mobile>

Figura 3. Simulador 3D Simulink Support Package for Parrot Minidrones



Fuente: MATLAB [En línea]. [Recuperado en 2023]. Disponible en: <https://la.mathworks.com/help/supportpkg/parrot/ref/path-planning-keyboard-example.html>

ROA, Marcos. DISEÑO DEL CONTROL DE NAVEGACIÓN PARA UN DRON PARROT MAMBO [en línea]. Grado en Ingeniería de Tecnología Industrial. Madrid: Universidad Pontificia Comillas. 2019. p.56 – 65. [Consultado 08 junio 2023]. Disponible en: <https://repositorio.comillas.edu/xmlui/handle/11531/32123>

En este documento se determina la experiencia de uso del Parrot Mambo desde el entorno Matlab. Valoran la importancia del sensor de flujo óptico para la navegación de interiores, pero, señala las limitaciones de este mismo sensor. Se concluye que para una buena implementación de cualquier controlador que haga uso de este sensor, se debe tener una superficie con marcas de tal manera que el dron pueda posicionarse. Recalcan la gran tarea que fue adaptar la programación al entorno universitario y que fue todo un reto descifrar el funcionamiento del paquete de soporte en Simulink.

NOORDIN, Aminurrashid; BASRI, Mohd; MOHAMED, Zaharuddin. Simulation and experimental study on PID control of a quadrotor MAV with perturbation [en línea]. Grado en Ingeniería de Tecnología Industrial. Malaysia: Universidad Técnica Malaysia Melaka. Escuela de ingeniería eléctrica. Facultad de ingeniería eléctrica y electrónica. 2019. p.4

– 6. [Consultado 08 junio 2023]. Disponible en: <https://beei.org/index.php/EEI/article/view/2158>

El artículo presenta a detalle el modelo matemático por el cual se puede definir el comportamiento dinámico del dron, además extraen las variables representativas del Parrot Mambo como la masa, el coeficiente de empuje y arrastre de los motores y las inercias necesarias para terminar de definir el sistema planteado. En él se trabaja con sistemas de control PID y la respuesta que estos tienen frente a perturbaciones. Se precisa de forma sencilla la habilidad del dron para maniobrar exitosamente frente a pequeñas perturbaciones y se aconsejan otro tipo de técnicas que se podrían implementar para obtener un controlador más robusto.

GARCIA, Daniel. Aplicación de Matlab/Simulink al posicionamiento y control de drones en interiores [en línea]. Máster universitario en Ingeniería Industrial. Madrid: Universidad De Alcalá. Escuela Politécnica Superior. 2020. p.66 – 68. [Consultado 08 junio 2023]. Disponible en: <https://ebuah.uah.es/dspace/handle/10017/44709>

En esta investigación se establece una visión más profunda del sensor de la cámara y el procesamiento de datos por parte de este sensor para el control en interiores. Implementan por primera vez filtros Kalman para la corrección de la información proporcionada por la cámara y así evitar los errores acumulativos asociados al sensor que se encuentra inmerso en un sistema inercial. Además, se habla de la desconexión parcial de los resultados obtenidos mediante las simulaciones y los resultados experimentales. Describen la necesidad de un fuerte ajuste para obtener un controlador funcional.

OKASHA, Mohamed; KRALEV, Jordan; ISLAM, Maidul. Design and Experimental Comparison of PID, LQR, and MPC Stabilizing Controllers for Parrot Mambo Mini-Drone [en línea]. Emiratos Árabes Unidos: Universidad de los Emiratos Árabes Unidos.

Departamento de ingeniería Aeroespacial y mecánica. 2022. p.11 – 19. [Consultado 08 junio 2023]. Disponible en: <https://www.mdpi.com/2226-4310/9/6/298>

Compara tres métodos diferentes de control en el seguimiento de una trayectoria. Se expone el desempeño de cada uno de estos, concluyendo que todos ofrecen un sistema estable y que bajo la técnica de modelo predictivo de control se obtiene una mayor robustez. Además, se muestra la formulación de cada uno de los controladores y el proceso de diseño de estos, así como también las decisiones y el razonamiento detrás de ellas para la determinación de las ganancias al momento de sintonizar los sistemas. Todas las comparaciones son presentadas en gráficos muy dinámicos y explícitos que facilitan una buena interpretación de los resultados.

NOORDIN, Aminurrashid; BASRI, Mohd; MOHAMED, Zaharuddin. Real-Time Implementation of an Adaptive PID Controller for the Quadrotor MAV Embedded Flight Control System. Universidad Técnica Malaysia Melaka. Escuela de ingeniería eléctrica. Facultad de ingeniería eléctrica y electrónica. 2023. p.15 – 22. [Consultado 08 junio 2023]. Disponible en: <https://www.mdpi.com/2226-4310/10/1/59>

Se discute la adición de un compensador fuzzy a un controlador PID adaptativo de tal manera que se obtenga una mayor robustez frente a perturbaciones. Se encuentra que mejora el comportamiento un 6% frente al PID y al PID adaptativo, pero que en consecuencia aumenta el consumo de la batería en su implementación. Para el diseño de los controladores hacen uso de un modelo reducido que no parece tener mayor incidencia sobre el control que ejercen y los resultados que obtienen.

3. MARCO TEÓRICO

En este capítulo se exploran varios temas fundamentales para el desarrollo del proyecto de investigación. Se aborda el modelado dinámico del cuadricóptero, que es esencial para comprender su comportamiento y características de vuelo. También se exponen las técnicas de control robusto, como el control LQG (Linear Quadratic Gaussian) y el control óptimo LQR (Linear Quadratic Regulator), que permiten diseñar controladores eficientes y estables para el cuadricóptero. Además, se abordan los filtros de Kalman, que son herramientas importantes para la estimación y combinación de datos en sistemas con incertidumbre. Por último, se exponen los índices de desempeño para evaluar y cuantificar el rendimiento de los sistemas de control. Estos temas proporcionan los fundamentos teóricos necesarios para el desarrollo y análisis del controlador del cuadricóptero en el presente proyecto.

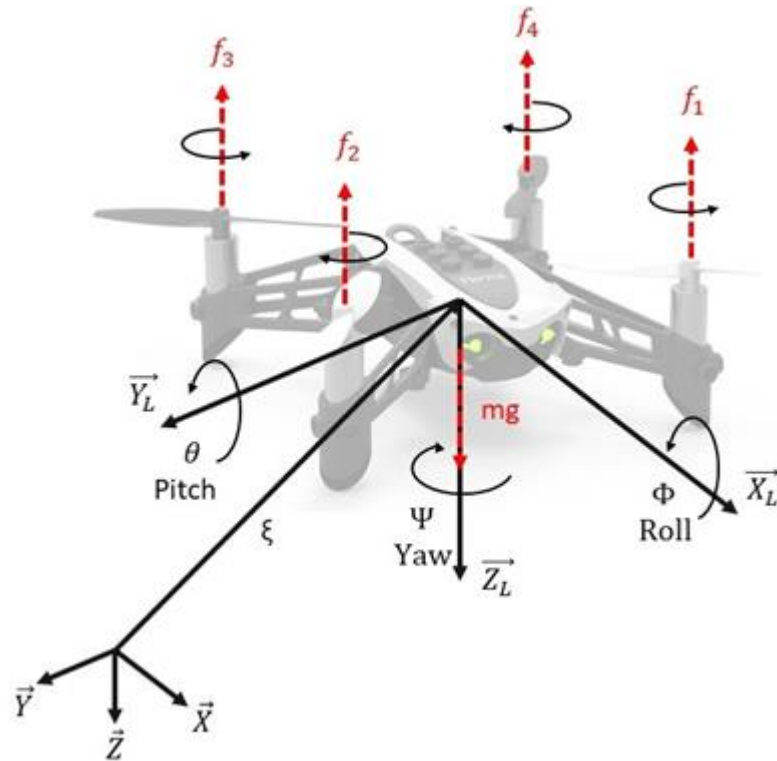
3.1 MODELADO DINÁMICO DEL CUADRICÓPTERO

Un cuadricóptero es un tipo de vehículo aéreo no tripulado (UAV) que se caracteriza por tener cuatro rotores dispuestos en una configuración de X o de cruz. En nuestro caso el Parrot Mambo posee una distribución en X. Cada rotor puede variar su velocidad y dirección de giro de manera independiente, lo que permite controlar el movimiento del cuadricóptero en los tres ejes espaciales.

En términos de grados de libertad, un cuadricóptero tiene seis grados de libertad: tres de traslación (movimiento en X, Y y Z) y tres de rotación (giro alrededor de los ejes X, Y y Z) que representan la orientación en términos de “*Roll*”, “*Pitch*” y “*Yaw*”. Estos grados de libertad son controlados por medio de los cuatro rotores, que generan una fuerza y un momento de torsión que permiten al cuadricóptero moverse y rotar en el aire. El cuadricóptero tiene un sistema de coordenadas ligado a él y con origen en su centro de masa, que se utiliza para describir su posición y orientación $\vec{B} = [\vec{X}_L, \vec{Y}_L, \vec{Z}_L]$. Por otro

lado, existe otro sistema de coordenadas, el sistema de referencia inercial, que es un sistema de coordenadas fijo en el espacio y es utilizado como referencia para describir la posición y orientación de otros objetos en movimiento. En el caso del cuadricóptero, se utiliza el sistema de coordenadas de referencia inercial $\vec{I} = [\vec{X}, \vec{Y}, \vec{Z}]$ como referencia para describir su posición y orientación.

Figura 4. DCL del dron Parrot Mambo y sistemas de referencia.



Fuente: Adaptación de fotografía del dron Parrot Mambo de Mundo dron [En línea]. [Recuperado en 2023]. Disponible en: <https://mundodrone.es/parrot-mambo/>

Para obtener el modelo de posición y orientación del cuadricóptero, se deben realizar mediciones de los ángulos de orientación y las coordenadas de posición en los sistemas de coordenadas del cuadricóptero y del sistema de coordenadas de referencia inercial. A partir de estas mediciones se pueden obtener las ecuaciones que describen la posición y orientación del cuadricóptero en términos de las coordenadas y ángulos medidos.

Partiendo de las consideraciones mencionadas anteriormente, tenemos que contemplar el cuadricóptero como un cuerpo rígido en el espacio que está sujeto a una fuerza principal para el movimiento, y tres momentos provocados por desequilibrios en las fuerzas de empuje de los rotores. Cada rotor produce fuerzas que contribuyen a los movimientos "*Roll*", "*Pitch*" y "*Yaw*".

El primer paso para desarrollar un control efectivo es modelar adecuadamente el sistema dinámico. El modelo dinámico del cuadricóptero se explica mediante la formulación matemática de Newton-Euler⁶, que se basa en la física newtoniana y la relación entre sistemas inerciales mediante ángulos de Euler.

El movimiento de *balanceo* o "*Roll*" (ángulo ϕ) se logra mediante un desequilibrio entre las fuerzas conjuntas f_1, f_4 y f_2, f_3 . Para el movimiento de *cabeceo* o "*Pitch*" (ángulo θ), se utiliza el desequilibrio entre las fuerzas conjuntas f_1, f_2 y f_3, f_4 . Finalmente, el último movimiento angular es de *guiñada* o "*Yaw*" (ángulo ψ) que se produce mediante un desequilibrio entre pares de fuerzas opuestas proporcionales a las fuerzas de empuje. El desplazamiento perpendicular al plano de los rotores (empuje) se obtiene mediante la suma de las cuatro fuerzas f_1, f_2, f_3 y f_4 .

3.1.1 Orientación del Cuadricóptero

Cuando se trabaja con objetos en movimiento, se necesita una referencia constante y estable para poder medir y predecir su movimiento. Un sistema de referencia inercial proporciona un marco de referencia para medir cualquier cambio en el movimiento, por lo tanto, se necesita establecer una orientación del dron para así poder estimar la posición y orientación del vehículo con respecto a un sistema de coordenadas de referencia inercial. Los sistemas de coordenadas se pueden ver en la Figura 4. $\vec{B} = [\vec{X}_L, \vec{Y}_L, \vec{Z}_L]$ es

⁶ PAIVA, Ernesto. MODELADO Y CONTROL DE UN CUADRICÓPTERO [en línea]. Máster en ingeniería mecánico-eléctrica con mención en automática y optimización. Piura: Universidad de Piura. 2016. p.7. [Consultado 08 junio 2023]. Disponible en: <https://pirhua.udep.edu.pe/handle/11042/2514>

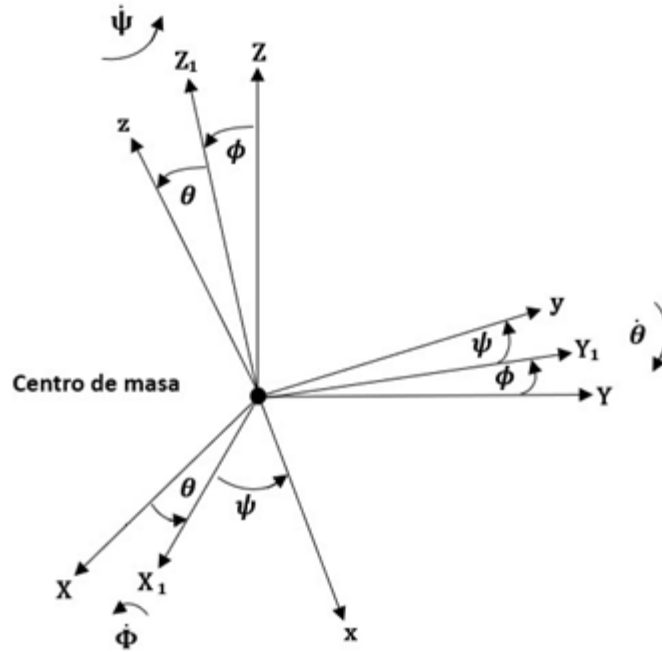
el sistema de coordenadas móvil que se encuentra ubicado en el centro de masa del cuadricóptero y $\vec{I} = [\vec{X}, \vec{Y}, \vec{Z}]$ es el sistema de coordenadas inercial fijado como marco de referencia en tierra. Establecido estos sistemas de coordenadas se necesita la representación de las rotaciones necesarias para llevar el sistema de coordenadas del dron desde su posición actual a otra posición en el espacio. Esto se hace usando la parametrización de ángulos de Euler, la orientación del dron en el espacio viene dada por una rotación R de $\vec{B}$ a $\vec{I}$, donde $R \in SO3$ es la matriz de rotación⁷.

En el contexto de un dron, los ángulos de Euler son importantes porque permiten describir la orientación del dron en el espacio en relación con un sistema de coordenadas de referencia. Esta información es fundamental para controlar el movimiento del dron y llevar a cabo tareas específicas, como el mantenimiento de una altitud constante, la estabilización durante vuelos en condiciones de viento o la navegación precisa en espacios reducidos.

En este trabajo de investigación se emplea la convención de Euler XYZ (giro alrededor de X , Y' , Z''), la cual es frecuentemente usada en la ingeniería aeroespacial y se conoce como ángulos de Tait-Bryan o ángulos Cardano. Los ángulos de Tait-Bryan se componen de tres ángulos que describen la rotación de un cuerpo rígido como resultado de rotaciones sucesivas en torno a los tres ejes $\vec{X}_L$, $\vec{Y}_L$ y $\vec{Z}_L$. En la Figura 5. Se ilustra la rotación de un cuerpo rígido mediante rotaciones sucesivas y se establecen los ángulos de rotación, los cuales son importantes para definir la matriz de rotación ortonormal R .

⁷ Ibid., p. 9.

Figura 5. Rotación de un cuerpo libre.



De acuerdo con la convención, la rotación de un cuerpo rígido en el espacio mediante tres rotaciones sucesivas⁸ es:

Rotación según $\vec{X}$ de ϕ :

El primer giro se realiza alrededor del eje $\vec{X}_L$, giro correspondiente al ángulo de “Roll” o de balanceo, ϕ .

$$\begin{bmatrix} X_1 \\ Y_1 \\ Z_1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi & \cos \phi \end{bmatrix} \begin{bmatrix} X_L \\ Y_L \\ Z_L \end{bmatrix} \quad (1)$$

Rotación según $\vec{Y}$ de θ :

⁸ BRESCIANI, Tommaso. Modelling, Identification and Control of a Quadrotor Helicopter [en línea]. Máster en control automático. Lund: Lund University. 2008. p.13. [Consultado 08 junio 2023]. Disponible en: <https://lup.lub.lu.se/luur/download?func=downloadFile&recordId=8847641&fileId=8859343>

La segunda rotación se realiza para dejar el eje $\vec{Z}_L$ en su posición final. Esto se realiza con un giro alrededor del eje $\vec{Y}$ a partir del nuevo eje $\vec{Y}_L$, con el ángulo “Pitch” o ángulo de cabeceo, θ .

$$\begin{bmatrix} X_2 \\ Y_2 \\ Z_2 \end{bmatrix} = \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix} \begin{bmatrix} X_1 \\ Y_1 \\ Z_1 \end{bmatrix} \quad (2)$$

Rotación según $\vec{Z}$ de Ψ :

La tercera rotación corresponde al giro del ángulo de *guinada* o “yaw”, Ψ , alrededor del eje $\vec{Z}$ a partir del nuevo eje $\vec{Z}_L$ para llevar el cuerpo rígido (cuadricóptero) a su posición final.

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \cos \Psi & -\sin \Psi & 0 \\ \sin \Psi & \cos \Psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix} \quad (3)$$

A partir de las rotaciones, se establecen las matrices de rotación que expresan la posición del cuerpo rígido girando en torno a cada eje de la siguiente manera:

$$R(x, \Phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \Phi & -\sin \Phi \\ 0 & \sin \Phi & \cos \Phi \end{bmatrix} \quad (4)$$

$$R(y, \theta) = \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix} \quad (5)$$

$$R(z, \Psi) = \begin{bmatrix} \cos \Psi & -\sin \Psi & 0 \\ \sin \Psi & \cos \Psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (6)$$

La matriz de rotación completa de un cuerpo rígido de B respecto a I , llamada *Matriz Coseno Directa*⁹, está dada por:

⁹ Bouabdallah, Samir; Noth, A. y Siegwart, Roland. PID vs LQ Control Techniques Applied to an Indoor Micro Quadrotor. En: International Conference on Intelligent Robots and Systems. Lausanne. 2004. vol.3. No. 3. p. 2.

$$\mathbf{R}_I = \mathbf{R}(z, \Psi) \cdot \mathbf{R}(y, \theta) \cdot \mathbf{R}(x, \Phi) \quad (7)$$

$$\mathbf{R}_I = \begin{bmatrix} \cos \Psi & -\sin \Psi & 0 \\ \sin \Psi & \cos \Psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \Phi & -\sin \Phi \\ 0 & \sin \Phi & \cos \Phi \end{bmatrix} \quad (8)$$

$$\mathbf{R}_I = \begin{bmatrix} \cos \Psi \cos \theta & \cos \Psi \sin \theta \sin \Phi - \sin \Psi \cos \Phi & \cos \Psi \sin \theta \cos \Phi + \sin \Psi \sin \Phi \\ \sin \Psi \cos \theta & \sin \Psi \sin \theta \sin \Phi + \cos \Psi \cos \Phi & \sin \Psi \sin \theta \cos \Phi - \cos \Psi \sin \Phi \\ -\sin \theta & \cos \theta \sin \Phi & \cos \theta \cos \Phi \end{bmatrix} \quad (9)$$

Debido a la propiedad de ortonormalidad, la matriz de rotación en el sistema de coordenadas $\mathbf{B}$ se expresa como la traspuesta de $\mathbf{R}_I$

$$\mathbf{R}_B = \begin{bmatrix} \cos \Psi \cos \theta & \sin \Psi \cos \theta & -\sin \theta \\ \cos \Psi \sin \theta \sin \Phi - \sin \Psi \cos \Phi & \sin \Psi \sin \theta \sin \Phi + \cos \Psi \cos \Phi & \cos \theta \sin \Phi \\ \cos \Psi \sin \theta \cos \Phi + \sin \Psi \sin \Phi & \sin \Psi \sin \theta \cos \Phi - \cos \Psi \sin \Phi & \cos \theta \cos \Phi \end{bmatrix} \quad (10)$$

A partir de la matriz de rotación (10) generada por las tres rotaciones sucesivas, la ortonormalidad y la derivada de la matriz ortonormal con una cierta matriz antisimétrica, se puede obtener las ecuaciones cinemáticas de rotación del vehículo que establecen las relaciones entre las velocidades angulares¹⁰.

Siendo $\omega = (p, q, r)^T$ el vector de velocidades angulares respecto al sistema $\mathbf{B}$ fijado al cuerpo rígido y $\theta = (\phi, \theta, \psi)^T$ el vector de ángulos de Euler. La relación entre $\dot{\theta}$ y ω es:

$$\dot{\theta} = \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \mathbf{T} \omega \quad (11)$$

¹⁰ OKASHA, Op., Cit, p. 3.

$$\mathbf{T} = \begin{bmatrix} 1 & \sin \Phi \tan \theta & \cos \Phi \tan \theta \\ 0 & \cos \Phi & -\sin \Phi \\ 0 & \sin \Phi \sec \theta & \cos \Phi \sec \theta \end{bmatrix} \quad (12)$$

Donde la variable $\mathbf{T}$ es llamada matriz de Euler y relaciona las velocidades angulares en el sistema $\mathbf{B}$ (sistema fijado al cuerpo) y la variación en el tiempo de los ángulos de Tait-Bryan (sistema inercial)¹¹.

$$\begin{bmatrix} \dot{\Phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin \Phi \tan \theta & \cos \Phi \tan \theta \\ 0 & \cos \Phi & -\sin \Phi \\ 0 & \sin \Phi \sec \theta & \cos \Phi \sec \theta \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad (13)$$

3.1.2 Modelo Aerodinámico para configuración en X

El dron Parrot Mambo presenta una configuración en equis “X”. El modelo dinámico se describe mediante las ecuaciones de Newton Euler. La cinemática se expresa con la siguiente expresión ya que es un cuerpo rígido de 6 GDL¹².

$$\dot{\xi} = J_{\theta} v \quad (14)$$

El vector ξ describe la posición lineal y angular respecto al sistema de coordenadas inercial.

$$\xi = \begin{bmatrix} T \\ \theta \end{bmatrix} = \begin{bmatrix} X \\ Y \\ Z \\ \Phi \\ \theta \\ \psi \end{bmatrix} \quad (15)$$

Por otro lado, el vector v describe la velocidad lineal y angular respecto al sistema de coordenadas fijas al dron.

¹¹ PAIVA, Op., Cit., p. 10.

¹² BRESCIANI, Op., Cit., p. 12.

$$v = \begin{bmatrix} V \\ \omega \end{bmatrix} = \begin{bmatrix} u \\ v \\ w \\ p \\ q \\ r \end{bmatrix} \quad (16)$$

La matriz de conversión J_Θ .

$$J_\Theta = \begin{bmatrix} \mathbf{R} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{T} \end{bmatrix} \quad (17)$$

La matriz de transferencia¹³ se define de acuerdo con:

$$T = \begin{bmatrix} \dot{\Phi} \\ \dot{\Theta} \\ \dot{\Psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin \Phi \tan \theta & \cos \Phi \tan \theta \\ 0 & \cos \Phi & -\sin \Phi \\ 0 & \sin \Phi \sec \theta & \cos \Phi \sec \theta \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad (18)$$

Cabe resaltar que las velocidades angulares en el sistema del dron (p, q, r) son distintas las derivadas de los ángulos de Tait-Bryan $(\dot{\Phi}, \dot{\Theta}, \dot{\Psi})$, y la relación entre estas se denota de la siguiente forma:

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\sin \theta \\ 0 & \cos \Phi & -\sin \Phi \cos \theta \\ 0 & -\sin \Phi & \cos \Phi \cos \theta \end{bmatrix} \begin{bmatrix} \dot{\Phi} \\ \dot{\Theta} \\ \dot{\Psi} \end{bmatrix} \quad (19)$$

Mediante la formulación Newton-Euler¹⁴, la dinámica no lineal bajo fuerzas y torques a un centro de masa expresado en su sistema de coordenadas se puede expresar así:

¹³ BRESCIANI, Op., Cit., p. 13.

¹⁴ RAFFO, Guilherme. Modelado y control de un helicóptero Quadrotor. [en línea]. Máster en automática, robótica y telemática. Sevilla: Universidad de Sevilla. 2007. p.15-16. [Consultado 08 junio 2023]. Disponible en: https://biblus.us.es/bibing/proyectos/abreproy/70017/descargar_fichero/Tesis_Master_GuilhermeRaffo.pdf

$$\begin{bmatrix} mI_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & I \end{bmatrix} \begin{bmatrix} \dot{V} \\ \dot{\omega} \end{bmatrix} + \begin{bmatrix} \omega \times mV \\ \omega \times I \omega \end{bmatrix} = \begin{bmatrix} F_B \\ \tau_B \end{bmatrix} \quad (20)$$

El termino $I_{3 \times 3}$ es la matriz de identidad, V y ω son los vectores de velocidad lineal y velocidad angular del cuerpo respecto a $\vec{B}$, m es la masa completa del cuerpo e I es la matriz diagonal de inercia con respecto a $\vec{B}$.

$$I = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \quad (21)$$

Debido a que el centro de masa coincide con el sistema de coordenadas, se puede decir que la expresión $\omega \times mV$ es igual a cero y queda de la siguiente forma:

$$\begin{bmatrix} mI_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & I \end{bmatrix} \begin{bmatrix} \dot{V} \\ \dot{\omega} \end{bmatrix} + \begin{bmatrix} 0 \\ \omega \times I \omega \end{bmatrix} = \begin{bmatrix} F_B \\ \tau_B \end{bmatrix} \quad (22)$$

De acuerdo con Bresciani¹⁵, las ecuaciones de movimiento para un cuerpo rígido son las que se describen a continuación:

$$\dot{r} = [\dot{X} \quad \dot{Y} \quad \dot{Z}]^T = v \quad (23)$$

$$m\dot{v} = RF_B \quad (24)$$

$$\dot{v} = [\ddot{X} \quad \ddot{Y} \quad \ddot{Z}]^T \quad (25)$$

Al igual que:

$$I\dot{\omega} = -\omega \times I \omega + \tau_B \quad (26)$$

$$\dot{\omega} = [\dot{p} \quad \dot{q} \quad \dot{r}]^T \quad (27)$$

¹⁵ BRESCIANI, Op., Cit., p. 19.

Las fuerzas externas aplicadas como el peso, las fuerzas aerodinámicas y el empuje son F_B , y los pares externos de los motores son τ_B , ambos pertenecen a $\vec{B}$. De acuerdo Raffo¹⁶ se establecen las siguientes ecuaciones:

$$R F_B = -m g e_3 + R_{e_3} \left(b \sum_{i=1}^4 \Omega_i^2 \right) \quad (28)$$

$$\tau_B = - \sum_{i=1}^4 J_R (\omega x e_3) \Omega_i + \tau_a \quad (29)$$

Donde J_R es el momento de inercia del rotor respecto a su eje, g es la gravedad, Ω_i es la velocidad angular del rotor, y e_3 es una componente básica. El efecto giroscópico solo ocurre cuando hay un desbalance en la sumatoria de las velocidades de los rotores.

Los pares τ_a aplicados en los tres ejes del cuadricóptero, vienen expresados como:

$$\tau_a = \begin{bmatrix} l(f_4 - F_2) \\ l(f_3 - F_1) \\ \sum_{i=1}^4 \tau_{Mi} \end{bmatrix} = \begin{bmatrix} lb(\Omega_4^2 - \Omega_2^2) \\ lb(\Omega_3^2 - \Omega_1^2) \\ d(\Omega_2^2 + \Omega_4^2 - \Omega_1^2 - \Omega_3^2) \end{bmatrix} = \begin{bmatrix} lU_2 \\ lU_3 \\ U_4 \end{bmatrix} \quad (30)$$

Donde l es la distancia entre los motores y el centro de gravedad, las constantes b y d son los coeficientes de empuje y torque respectivamente¹⁷.

La fuerza U_1 es la encargada del movimiento del dron en el eje z, y también se conoce como *throttle*. f_i es la fuerza de empuje producida por cada motor.

¹⁶ RAFFO, Op., Cit., p. 16-18.

¹⁷ PAIVA, Op., Cit., p. 13-15.

$$U_1 = \sum_{i=1}^4 f_i = \sum_{i=1}^4 b\Omega_i^2 \quad (31)$$

U_2 y U_3 son las fuerzas que crean el movimiento de “Roll” y “Pitch”. Por ultimo U_4 es el torque que produce el movimiento de “Yaw”, todas estas aparecen como desequilibrios entre las velocidades de cada uno de los rotores.

$$\begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix} = \begin{bmatrix} b(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \\ b(\Omega_4^2 + \Omega_3^2 - \Omega_1^2 - \Omega_2^2) \\ b(\Omega_2^2 + \Omega_3^2 - \Omega_1^2 - \Omega_4^2) \\ d(\Omega_1^2 + \Omega_3^2 - \Omega_2^2 - \Omega_4^2) \end{bmatrix} \quad (32)$$

$$\Omega_1^2 = \frac{1}{4b} U_1 + \frac{1}{4b} U_2 - \frac{1}{4b} U_3 - \frac{1}{4d} U_4 \quad (33)$$

$$\Omega_2^2 = \frac{1}{4b} U_1 - \frac{1}{4b} U_2 - \frac{1}{4b} U_3 + \frac{1}{4d} U_4 \quad (34)$$

$$\Omega_3^2 = \frac{1}{4b} U_1 - \frac{1}{4b} U_2 + \frac{1}{4b} U_3 - \frac{1}{4d} U_4 \quad (35)$$

$$\Omega_4^2 = \frac{1}{4b} U_1 + \frac{1}{4b} U_2 - \frac{1}{4b} U_3 - \frac{1}{4d} U_4 \quad (36)$$

Determinadas todas las expresiones anteriores, desde la (23) hasta la (36) se remplacea en (22) y se obtiene el modelo matemático no lineal del cuadricóptero¹⁸:

$$\ddot{X} = (\sin \Psi \sin \Phi + \cos \Psi \sin \theta \cos \Phi) \frac{-U_1}{m} \quad (37)$$

$$\ddot{Y} = (-\cos \Psi \sin \Phi + \sin \Psi \sin \theta \cos \Phi) \frac{-U_1}{m} \quad (38)$$

$$\ddot{Z} = g + (\cos \theta \cos \Phi) \frac{-U_1}{m} \quad (39)$$

¹⁸ PAIVA, Op., Cit., p. 17-18.

$$\dot{\Phi} = p + q \sin \Phi \tan \theta + r \cos \Phi \tan \theta \quad (40)$$

$$\dot{\theta} = q \cos \Phi - r \sin \Phi \quad (41)$$

$$\dot{\Psi} = q \sin \Phi \sec \theta + r \cos \Phi \sec \theta \quad (42)$$

$$\dot{p} = \frac{I_{YY} - I_{ZZ}}{I_{XX}} qr - \frac{J_{TP}}{I_{XX}} q \Omega + \frac{l U_2}{I_{XX}} \quad (43)$$

$$\dot{q} = \frac{I_{ZZ} - I_{XX}}{I_{YY}} pr + \frac{J_{TP}}{I_{YY}} p \Omega + \frac{l U_3}{I_{YY}} \quad (44)$$

$$\dot{r} = \frac{I_{XX} - I_{YY}}{I_{ZZ}} pq + \frac{U_4}{I_{ZZ}} \quad (45)$$

Donde Ω es la sumatoria de todas las velocidades angulares de los motores teniendo en cuenta el sentido de giro.

$$\Omega = \Omega_1 - \Omega_2 + \Omega_3 - \Omega_4 \quad (46)$$

En los movimientos lineales del dron a lo largo de los ejes el valor de omega es despreciable, por lo tanto, el término que está multiplicado por Ω también se puede despreciar.

Cuando los ángulos ϕ , θ y ψ son pequeños, las ecuaciones (40), (41) y (42) correspondientes a las velocidades angulares ($\dot{\phi}$, $\dot{\theta}$, $\dot{\psi}$) respecto a los ejes del sistema inercial pueden aproximarse a las velocidades angulares (p , q , r) de los ejes del sistema de referencia fijo al cuadricóptero. Esto es ampliamente utilizado en el control de

estabilización de vuelo, ya que en estas aplicaciones los ángulos son relativamente pequeños¹⁹.

$$p = \dot{\Phi} \quad (47)$$

$$q = \dot{\theta} \quad (48)$$

$$r = \dot{\Psi} \quad (49)$$

Por lo tanto, las ecuaciones utilizadas para el control del cuadricóptero son las siguientes:

$$\ddot{X} = (\sin \Psi \sin \Phi + \cos \Psi \sin \theta \cos \Phi) \frac{-U_1}{m} \quad (50)$$

$$\ddot{Y} = (-\cos \Psi \sin \Phi + \sin \Psi \sin \theta \cos \Phi) \frac{-U_1}{m} \quad (51)$$

$$\ddot{Z} = g + (\cos \theta \cos \Phi) \frac{-U_1}{m} \quad (52)$$

$$\ddot{\Phi} = \dot{p} = \frac{I_{YY} - I_{ZZ}}{I_{XX}} qr + \frac{lU_2}{I_{XX}} \quad (53)$$

$$\ddot{\theta} = \dot{q} = \frac{I_{ZZ} - I_{XX}}{I_{YY}} pr + \frac{lU_3}{I_{YY}} \quad (54)$$

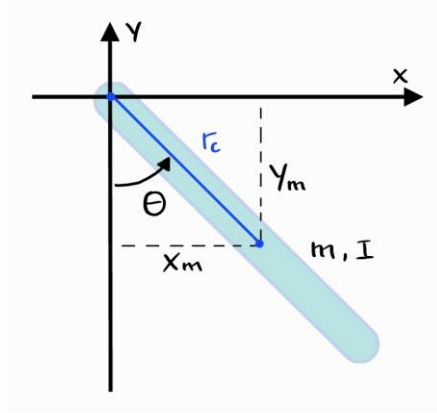
$$\ddot{\Psi} = \dot{r} = \frac{I_{XX} - I_{YY}}{I_{ZZ}} pq + \frac{U_4}{I_{ZZ}} \quad (55)$$

¹⁹ PAIVA, Op., Cit., p. 11.

3.2 MODELO DE UN PENDULO BARRA CON ECUACIÓN DE EULER-LAGRANGE

El modelo de un péndulo barra se realiza con el propósito de calcular las inercias del dron mediante el teorema de ejes paralelos.

Figura 6. Esquema péndulo barra



Inicialmente se define la posición del centro de masa como se observa en la Figura 6 y se describe en las ecuaciones (78) y (79).

$$X_m = r_c \sin(\theta) \quad (56)$$

$$Y_m = -r_c \cos(\theta) \quad (57)$$

Debido a que solo hay movimiento rotacional sobre el eje Z solo se considera esta velocidad angular.

$$\omega_x = \omega_y = 0 \quad (58)$$

$$\omega_z = \dot{\theta} \quad (59)$$

Tensor de inercia:

$$I = \begin{bmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{yx} & I_{yy} & I_{yz} \\ I_{zx} & I_{zy} & I_{zz} \end{bmatrix} \quad (60)$$

La velocidad en el centro de masa se calcula de manera vectorial por la ecuación (61).

$$V_c = \omega \times r \quad (61)$$

$$V_c = \begin{bmatrix} 0 \\ 0 \\ \dot{\theta} \end{bmatrix} \times \begin{bmatrix} r_c \sin(\theta) \\ -r_c \cos(\theta) \\ 0 \end{bmatrix} \quad (62)$$

$$V_c = \begin{bmatrix} \dot{\theta} r_c \cos(\theta) \\ \dot{\theta} r_c \sin(\theta) \\ 0 \end{bmatrix} \quad (63)$$

La energía cinética de la velocidad lineal y angular viene dada por las ecuaciones (65) y (67).

$$K_v = \frac{1}{2} m \|V_c\|^2 = \frac{1}{2} m \begin{pmatrix} \dot{\theta} r_c \cos(\theta) \\ \dot{\theta} r_c \sin(\theta) \\ 0 \end{pmatrix}^T \begin{pmatrix} \dot{\theta} r_c \cos(\theta) \\ \dot{\theta} r_c \sin(\theta) \\ 0 \end{pmatrix} \quad (64)$$

$$K_v = \frac{1}{2} m \dot{\theta}^2 r_c^2 \quad (65)$$

$$K_w = \frac{1}{2} \omega' I \omega = \frac{1}{2} \begin{pmatrix} 0 \\ 0 \\ \dot{\theta} \end{pmatrix}^T \begin{bmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{yx} & I_{yy} & I_{yz} \\ I_{zx} & I_{zy} & I_{zz} \end{bmatrix} \begin{pmatrix} 0 \\ 0 \\ \dot{\theta} \end{pmatrix} \quad (66)$$

$$K_v = \frac{1}{2} I_{zz} \dot{\theta}^2 \quad (67)$$

La energía potencial viene dada por la ecuación (68).

$$U = -mgr_c \cos(\theta) \quad (68)$$

Para aplicar la ecuación generalizada de Euler-Lagrange ecuación (70), se debe calcular el Lagrangiano, el cual es la suma de toda la energía del sistema como se observa en la ecuación (69).

$$\mathcal{L} = \frac{1}{2} (mr_c^2 + I_{zz}) \dot{\theta}^2 + mgr_c \cos(\theta) \quad (69)$$

$$\frac{d}{dt} \frac{\partial \mathcal{L}}{\partial \dot{q}} - \frac{\partial \mathcal{L}}{\partial q} + \frac{\partial R}{\partial \dot{q}} = \tau \quad (70)$$

Desarrollando los términos de la expresión (70) tendríamos lo siguiente:

$$\frac{\partial \mathcal{L}}{\partial \dot{q}} = \frac{\partial}{\partial \dot{\theta}} \left(\frac{1}{2} (mr_c^2 + I_{zz}) \dot{\theta}^2 + mgr_c \cos(\theta) \right) = (mr_c^2 + I_{zz}) \dot{\theta} \quad (71)$$

$$\frac{d}{dt} \frac{\partial \mathcal{L}}{\partial \dot{q}} = (mr_c^2 + I_{zz})\ddot{\theta} \quad (72)$$

$$\frac{\partial \mathcal{L}}{\partial q} = \frac{\partial}{\partial \theta} \left(\frac{1}{2} (mr_c^2 + I_{zz})\dot{\theta}^2 + mgr_c \cos(\theta) \right) = -mgr_c \sin(\theta) \quad (73)$$

$$R = \frac{1}{2} b \dot{\theta}^2 \quad \frac{\partial R}{\partial \dot{q}} = b \dot{\theta} \quad (74)$$

Dando, así como resultado el modelo dinámico del sistema denotado por la ecuación (75).

$$(mr_c^2 + I_{zz})\ddot{\theta} + mgr_c \sin(\theta) + b \dot{\theta} = \tau \quad (75)$$

3.3 CONTROL PID

Un controlador PID es un tipo de controlador utilizado en sistemas de control automático para mantener o regular una variable de proceso específica dentro de un rango deseado. Los controladores PID son ampliamente utilizados en una variedad de aplicaciones industriales y de automatización debido a su capacidad para ajustar la salida del sistema en función de las diferencias entre la variable de proceso medida y el valor deseado (setpoint). El libro de Ogata señala que más de la mitad de los controladores industriales utilizan esquemas de control PID o PID modificados²⁰. Aquí hay una breve descripción de los tres componentes principales de un controlador PID:

Proporcional (P): La acción proporcional del controlador es directamente proporcional al error actual entre la variable de proceso medida y el setpoint. Cuanto mayor sea el error, mayor será la corrección proporcional aplicada. Este componente ayuda a reducir el error presente en el sistema.

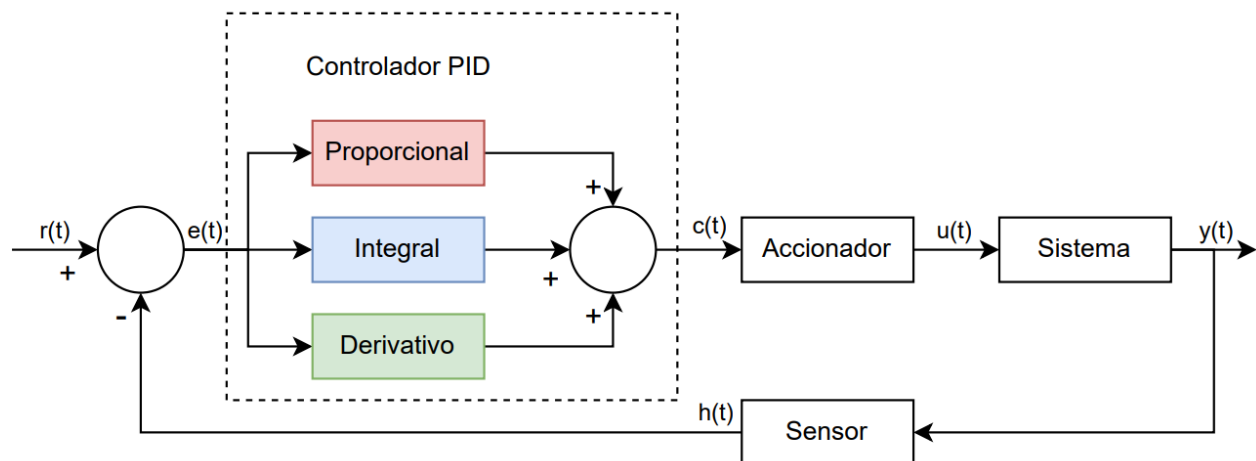
²⁰ OGATA, Katsuhiko. Ingeniería de control moderna. 5 ed. Madrid: Pearson education, 2010. p. 567. ISBN 848322660X

Integral (I): La acción integral del controlador se basa en la acumulación de errores pasados a lo largo del tiempo. Ayuda a eliminar el error constante o de estado estacionario que podría persistir debido a factores como la fricción, la inercia o las perturbaciones persistentes. El componente integral se encarga de eliminar este error acumulado.

Derivativo (D): La acción derivativa del controlador se basa en la tasa de cambio del error en el tiempo. Ayuda a anticipar las tendencias futuras y contrarrestarlas antes de que el error alcance un valor significativo. Esto es particularmente útil para prevenir oscilaciones o fluctuaciones excesivas en el sistema.

La combinación de estos tres componentes en un controlador PID permite controlar una amplia gama de sistemas y procesos, desde sistemas de calefacción y refrigeración hasta procesos industriales complejos. Ajustar los parámetros del controlador PID correctamente es esencial para lograr un rendimiento óptimo y mantener la estabilidad del sistema.

Figura 7. Estructura del controlador PID



Por otro lado, un controlador PD es un tipo de controlador más simplificado que utiliza la parte "Proporcional" y "Derivativo". El controlador PD se utiliza para controlar una variable

de proceso en función de la diferencia entre el valor deseado (setpoint) y el valor real medido (proceso). Aunque el controlador PD no aborda el error de estado estacionario por sí solo, es eficaz para mejorar la respuesta transitoria y la estabilidad de muchos sistemas.

El controlador PD se utiliza en una variedad de aplicaciones, como control de velocidad de motores, control de temperatura y control de posición en sistemas automatizados. La elección entre un controlador PD o un controlador PID depende de las características específicas del sistema y de los objetivos de control, ya que cada uno tiene sus ventajas y desventajas en diferentes situaciones.

3.4 CONTROL ROBUSTO LQG

Control LQG o control cuadrático gaussiano es la combinación entre un regulador LQR y el diseño de un estimador de estados que involucre el uso de un filtro Kalman. El control LQR funciona muy bien bajo un ambiente determinístico donde las mediciones realizadas por los sensores no se ven alteradas por ruido. En la realidad todos los procesos funcionan en ambientes ruidosos y las mediciones son de una naturaleza no determinística, allí es donde aparecen los filtros Kalman para mitigar el ruido.

3.4.1 Control óptimo LQR

LQR o regulador cuadrático lineal es una estrategia de control que consiste en encontrar una ley de control lineal que pondere los errores en las variables de estado y las entradas del control. También se conoce como control óptimo, ya que con su implementación se puede minimizar el costo bajo algunos compromisos impuestos por el diseñador. En otras palabras, el LQR nos dota de la capacidad para elegir el tipo de estrategia con la cual podemos seguir una referencia, dándonos así la opción de elegir entre una rápida respuesta o seguir una referencia con el menor coste energético posible.

Para hacer uso de este controlador se necesita de un modelo lineal del sistema a controlar. Ya que la mayor parte de los sistemas no poseen esta linealidad, se debe entonces realizar una linealización alrededor de un punto de operación. A continuación, se presenta el análisis matemático:

Primero se parte de la función de costo (76). En esta función lo que se busca es minimizar el error del sistema desde un tiempo inicial hasta un tiempo final.

$$J = \int_{t_0}^{t_f} e(t) dt \quad (76)$$

Como el LQR es un control optimo cuadrático, entonces se hace uso del error al cuadrado como se muestra en (77).

$$J = \int_{t_0}^{t_f} e(t)^2 dt \quad (77)$$

Debido a que el error no solo depende de los estados o salidas sino también de las entradas, una mejor expresión sería la ecuación (78).

$$J = \int_{t_0}^{t_f} (e(t)^2 - p u(t)^2) dt \quad (78)$$

Ahora, para relacionar la función de costo con nuestro sistema dinámico se plantean la ecuación de estado.

$$\dot{x} = Ax(t) + Bu(t) \quad (79)$$

Donde,

$$u(t) = -Kx(t) \quad (80)$$

Al definir la ecuación de costo en términos de espacio de estados obtenemos la ecuación (81), el mínimo índice de comportamiento²¹.

$$J = \int_{t_0}^{t_f} (x^T Q x + u^T R u) dt \quad (81)$$

Q y R son los factores de ponderación, es decir que podemos elegir una matriz Q de tal manera que se prioricen los estados deseados, o la matriz R de tal forma que se priorice la conservación de energía. Q y R deben ser matrices reales, simétricas, constantes y definidas positivas.

Para solucionar el problema de optimización se aplica la ecuación algebraica de Riccati (82). Que además puede solucionarse a través del comando $[K, P, E] = lqr(A, B, Q, R)$ en MATLAB²².

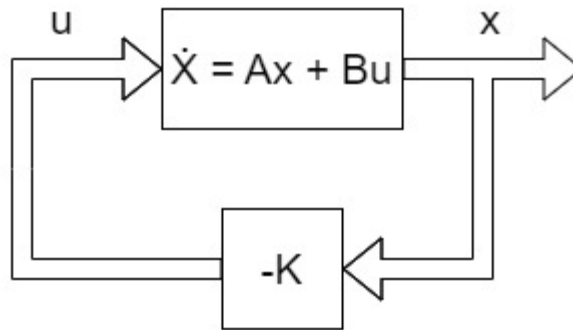
$$PA + A^T P - PBR^{-1}B^T P + Q = 0 \quad (82)$$

Para implementar la ley de control ya definida, se puede representar esquemáticamente de la siguiente forma:

²¹ Ibid., p.798.

²² Ibid., p.798.

Figura 8. Sistema de control LQR

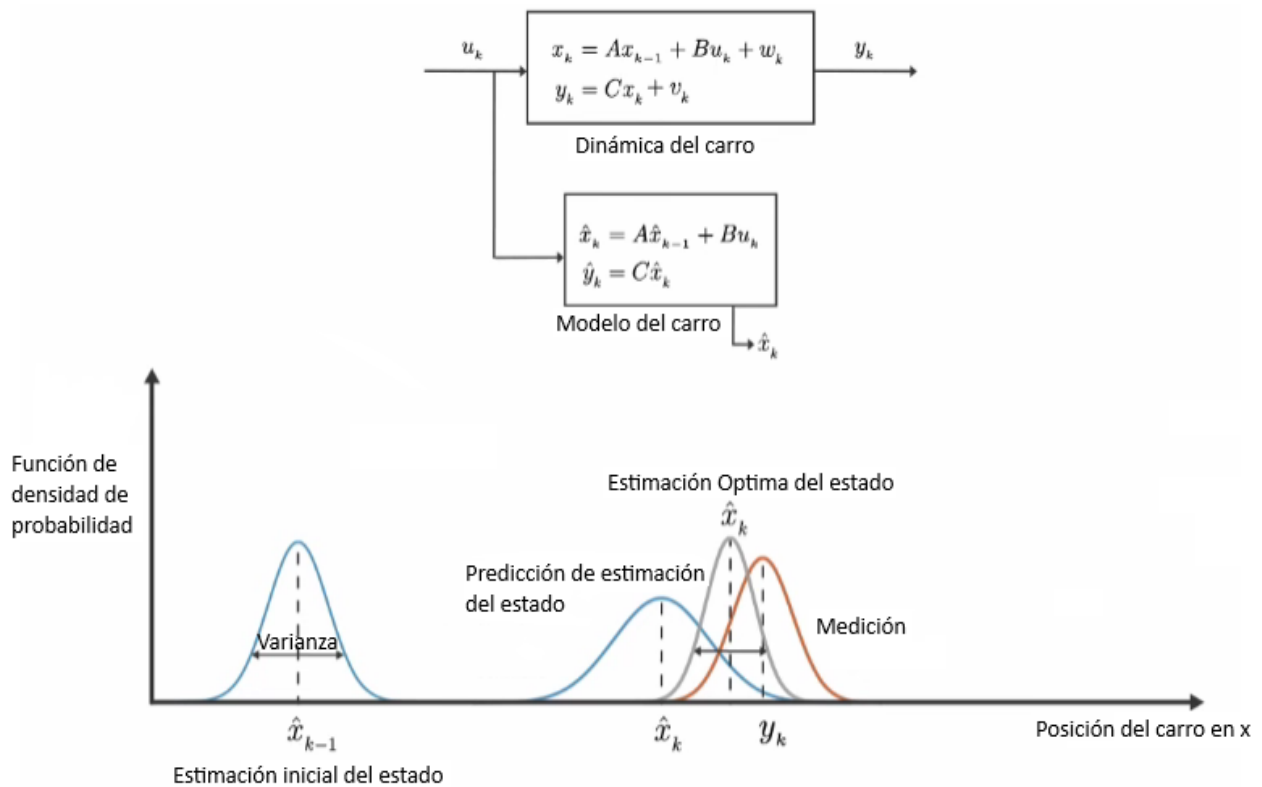


Fuente: OGATA, Katsuhiko. Ingeniería de control moderna, 5 ed. Madrid: Pearson education, 2010. p.793

3.4.2 Filtros Kalman

El filtro Kalman es un algoritmo de estimación óptima. Un uso común es estimar los estados de un sistema cuando estos no pueden ser medidos directamente y cuando hay mediciones de otros sensores, pero estos están sujetos a ruido. En otras palabras, a partir de unas entradas y salidas de nuestra planta y el modelo matemático del sistema, se puede no solo estimar un estado como un observador, sino que también se asegura de encontrar la estimación con la menor varianza posible, que normalmente es causada por el ruido en los sensores, algo que sucede siempre en la vida real tal como se ve en la Figura 9.

Figura 9. Ejemplo filtro Kalman



3.5 ÍNDICES DE DESEMPEÑO

Los índices de desempeño en control son medidas cuantitativas utilizadas para evaluar la eficiencia y efectividad de un sistema de control. Proporcionan información sobre cómo se comporta el sistema en términos de su capacidad para seguir una referencia o para rechazar perturbaciones. Para evaluar que tan optimo es un sistema de control se aplican estos índices y se elige aquel que los minimice.

Existen diferentes tipos de criterios comúnmente utilizados en el control, y cada uno de ellos se enfoca en aspectos específicos del comportamiento del sistema. A continuación, se presentan los índices de desempeño basados en la integral del error.

3.5.1 (ISE) Criterio de la integral del error al cuadrado

Este criterio ayudará a diferenciar entre sistemas con amortiguamiento excesivo y sistemas con amortiguamiento insuficiente²³. Al integrar el cuadrado del error, se le da mayor importancia a los errores grandes y persistentes, lo que puede ser útil para sistemas en los que se desea reducir al mínimo tanto los errores instantáneos como los errores en estado estacionario. Sin embargo, el índice ISE no tiene en cuenta otros aspectos del comportamiento del sistema, como el tiempo de respuesta o el sobre pico.

$$ISE = \int_0^T e(t)^2 dt \quad (83)$$

3.5.2 (ITSE) Criterio de la integral del error cuadrado multiplicado por el tiempo

Cuando se pondera el cuadrado del error por el tiempo, se otorga más importancia a los errores que se mantienen durante períodos prolongados, lo que resulta beneficioso para sistemas en los que se busca minimizar tanto los errores instantáneos como los errores en estado estacionario, teniendo en cuenta además el tiempo requerido para lograr la respuesta deseada.

$$ITSE = \int_0^T te(t)^2 dt \quad (84)$$

3.5.3 (IAE) Criterio de la integral del valor absoluto del error

Con el IAE se penalizan tanto los errores grandes como los pequeños, lo que puede ser útil para sistemas en los que se desea minimizar la magnitud total del error a lo largo del tiempo.

$$IAE = \int_0^T |e(t)| dt \quad (85)$$

²³ DORF, Richard. Modern Control Systems. 13 ed. Boston: Editorial Pearson, 2016. p. 316. ISBN 0134407628

3.5.4 (ITAE) Criterio de la integral del valor absoluto del error multiplicado por el tiempo

La multiplicación del valor absoluto del error por el tiempo proporciona una mayor importancia a los errores que persisten. Esta consideración resulta valiosa en sistemas donde se busca minimizar tanto los errores instantáneos como los errores en estado estacionario, teniendo en cuenta además el tiempo necesario para lograr la respuesta deseada. El índice de desempeño ITAE ofrece la mejor selectividad entre los índices de desempeño, lo que significa que el valor mínimo de la integral se puede identificar claramente al variar los parámetros del sistema²⁴.

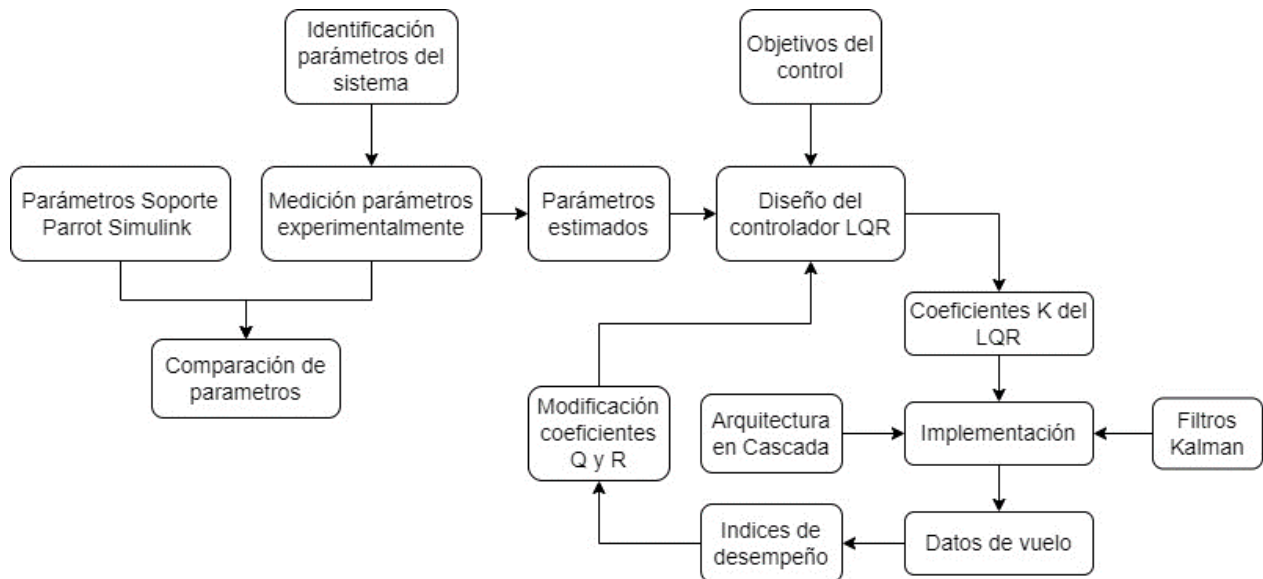
$$ITAE = \int_0^T t|e(t)|dt \quad (86)$$

²⁴ Ibid., p.316.

4. METODOLOGÍA

La metodología del proyecto se estructuró a través del diagrama presentado en la Figura 10. En primer lugar, se identificaron los parámetros presentes en el modelo matemático del dron que eran medibles. A continuación, se llevó a cabo la medición experimental de estos parámetros y se realizó una comparación con los valores preestablecidos por el soporte Parrot. Con base en los parámetros estimados y los objetivos de control, se procedió al diseño inicial de un controlador LQR que proporcionó coeficientes K . Estos coeficientes se implementaron en el control de vuelo del dron utilizando una arquitectura en cascada y filtros Kalman. Posteriormente, después de realizar pruebas, se recopilaron los datos de vuelo y se utilizaron, junto con los índices de desempeño, para ajustar los coeficientes Q y R . Esto marco el inicio del proceso iterativo para el desarrollo un controlador LQG más eficiente.

Figura 10. Estructura metodológica



4.1 IDENTIFICACIÓN Y VALIDACIÓN DE LOS PARÁMETROS DEL MODELADO DINÁMICO DEL PARROT MAMBO

En la sección 3.1 se desarrolló el modelado dinámico del UAV Parrot Mambo teniendo en cuenta su configuración física en “X” y cantidad de rotores. Los parámetros del modelo dinámico hacen referencia a las constantes intrínsecas del UAV que para este caso de estudio son la masa, las inercias, coeficiente de empuje, coeficiente de torque y la distancia entre los motores al centro de gravedad. Los parámetros mencionados se encuentran en las ecuaciones (50) a (55). Estos parámetros son importantes para describir las características y respuestas del dron en diversas situaciones. Un modelo minucioso y completo incorporará una amplia variedad de estos parámetros, lo que permitirá prever con gran precisión cómo el dron reaccionará ante diferentes entradas y contextos de vuelo.

En los trabajos desarrollados con el UAV Parrot Mambo (sección 2.2) los valores usados para los parámetros fueron obtenidos por el Soporte Parrot.

Cuadro 1. Parámetros Soporte Parrot minidrones de Simulink

PARÁMETRO	VALOR	UNIDADES
m	0.063	kg
I_{xx}	5.829E-05	kgm^2
I_{yy}	7.169E-05	kgm^2
I_{zz}	0.0001	kgm^2
b	4.72E-08	Ns^2
d	1.139E-10	Nms^2
l	0.0624	m

4.1.1 Masa y distancia entre los motores y el centro de gravedad

La masa del dron se determinó con una báscula de laboratorio. El valor registrado de la masa total del dron incluida las protecciones de hélice son de 0.072 kg. El valor proporcionado de 0.063 kg por el soporte Parrot es sin protecciones de hélice.

La distancia entre los motores y el centro de gravedad se determinó con una regla graduada, el valor 0.062 m coincidió con el proporcionado por el soporte Parrot.

4.1.2 Inercias

Para determinar experimentalmente las inercias en todos los ejes del dron se realizó un montaje de péndulo barra en el laboratorio. La lectura del ángulo del péndulo con respecto a la vertical se hizo con un potenciómetro magnético para así poder obtener un voltaje y convertirlo en ángulo respecto al tiempo. La conversión de la lectura del voltaje se hizo con Simulink. En base al modelo de un péndulo barra con la ecuación de Euler-Lagrange desarrollado en la sección 3.2 y la estimación de parámetros de Grey Box con Matlab se obtuvieron las inercias experimentalmente.

Un modelo de estimación de Grey Box es una herramienta que combina el conocimiento teórico de un sistema con datos observados para realizar predicciones o estimaciones. Esta herramienta integra ecuaciones teóricas con datos reales para ajustar parámetros y refinar la comprensión del sistema. Matlab cuenta con un ejemplo llamado *Represent Nonlinear Dynamics Using Matlab File for Grey-Box Estimation*, allí se detalla cómo realizar las estimaciones de parámetros de acuerdo con las ecuaciones dinámicas de un sistema no lineal con sus datos de entradas y salidas.

Para el cálculo de las inercias se desarrollaron los siguientes pasos:

1. Montaje y calibración del sistema péndulo barra.
2. Obtención de los ángulos en función del tiempo de las oscilaciones realizadas por la barra.

3. Montaje del dron en el extremo inferior de la barra.
4. Obtención de los ángulos en función del tiempo de las oscilaciones realizadas por el sistema de péndulo barra dron.
5. Formulación de las ecuaciones teóricas en representación de estados para Grey Box.

Con la ecuación (75) obtenida teóricamente en la sección 3.2 se despejó la aceleración angular:

$$\ddot{\theta} = \frac{\tau - mgr_c \sin(\theta) - b\dot{\theta}}{mr_c^2 + I_{zz}} \quad (87)$$

Para la representación de estados se estableció la expresión:

$$\dot{x} = f(x, u) \quad (88)$$

Donde el vector de estados está expresado como:

$$X = [\theta \ \dot{\theta}]^T \quad (89)$$

El vector de entradas:

$$u = [\tau] \quad (90)$$

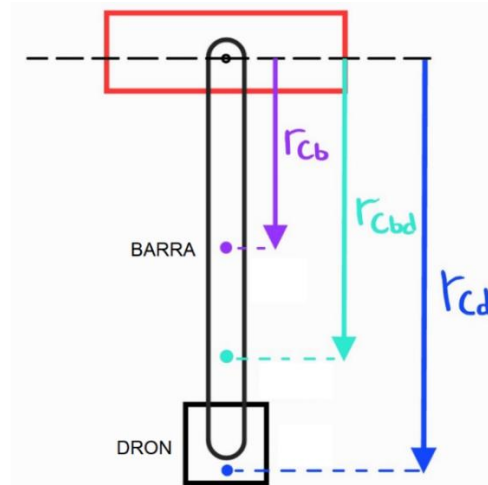
Y las ecuaciones:

$$x_1 = \theta ; x_2 = \dot{\theta} \quad (91)$$

$$\dot{x} = F(x, u) = \begin{Bmatrix} x_2 \\ \tau - mgr_c \sin(x_1) - bx_2 \\ mr_c^2 + I \end{Bmatrix} \quad (92)$$

6. Estimación de las inercias del sistema péndulo dron con Grey Box.
7. Cálculo de las inercias sobre cada eje del dron con el teorema de los ejes paralelos.

Figura 11. Esquema del sistema péndulo dron



Donde:

$$m_{dron} = 0.072 \text{ kg}$$

$$m_{barra} = 0.076 \text{ kg}$$

$$r_{cb} = 0.19 \text{ m}$$

$$r_{cbd} = 0.2825 \text{ m}$$

$$r_{cd} = 0.38 \text{ m}$$

$$I_{barra\ dron} + m_{barra\ dron} \cdot r_{centroide\ barra\ dron} = I_{dron} + m_{dron} \cdot r_{centroide\ dron} + I_{barra} + m_{barra} \cdot r_{centroide\ barra} \quad (93)$$

4.1.3 Coeficiente de empuje

El coeficiente de empuje es la capacidad del dron para generar una fuerza que lo mantenga en el aire. Este depende en gran medida de la forma de las hélices, el perfil aerodinámico y el ángulo de ataque. En cada dron este coeficiente es diferente, por lo tanto se requiere de su identificación y validación con el coeficiente estimado que ofrece el soporte Parrot.

Para el cálculo del coeficiente de empuje se desarrollaron los siguientes pasos:

1. Montaje de base.

2. Medición y verificación de las rpm de cada motor.
3. Calibración celda de carga.
4. Medición de fuerza ejercida.
5. Cálculo del coeficiente de empuje.
6. Comparación con el parámetro proporcionado por el soporte Parrot.

El cálculo del coeficiente de empuje se realizó con la ecuación (95).

$$U1 = mg \quad (94)$$

$$b = \frac{U1}{\left(2 \left(RPM_{MPI} \frac{\pi}{30}\right)^2 + 2 \left(RPM_{MPD} \frac{\pi}{30}\right)^2\right)} \quad (95)$$

4.1.4 Coeficiente de torque

El coeficiente de torque es la medida de la fuerza rotacional que actúa sobre el cuerpo del dron debido a la acción de sus hélices o rotores. Cuando los rotores giran, generan fuerzas de empuje que pueden causar un giro no deseado. El coeficiente de torque es una medida de esta tendencia a girar.

Para el calculo del coeficiente de torque se desarrollaron los siguientes pasos:

- Adecuación de base.
- Montaje mecanismo de sujeción dron.
- Medición de fuerza ejercida.
- Cálculo del coeficiente de torque.
- Comparación con el parámetro proporcionado por el soporte Parrot.

El cálculo del coeficiente de torque se realizó con la ecuación (97).

$$U4 = F * D \quad (96)$$

$$d = \frac{U4}{\left(2 \left(RPM_{MPD} \frac{\pi}{30}\right)^2\right)} \quad (97)$$

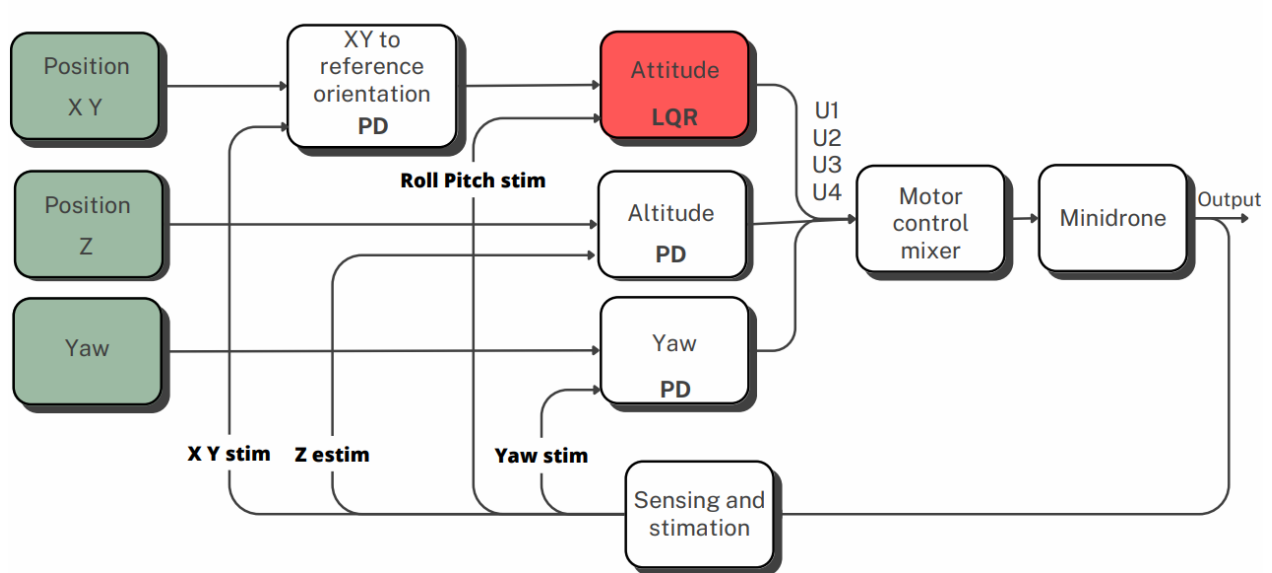
4.2 ARQUITECTURA DE CONTROL EN CASCADA

Para este proyecto, se ha seguido la implementación de una estructura de control conocida como "Control en Cascada" establecida por el soporte Parrot para minidrones de Simulink. Esta arquitectura de control se caracteriza por incorporar dos controladores interconectados, donde la salida del controlador primario sirve como referencia para el controlador secundario. La salida del controlador secundario, a su vez, influye en el proceso mismo.

Esta estrategia no solo mejora las capacidades dinámicas del sistema de control, sino que también reduce el impacto de diversas perturbaciones. El funcionamiento del control en cascada implica el uso de mediciones de variables internas para detectar rápidamente las perturbaciones y comenzar a aplicar correcciones.

Estos factores llevaron a la elección de implementar una estructura de control en cascada en el modelo del quadrotor. En esta estrategia, el controlador primario es representado por el bloque denominado *XY to reference orientation*, cuya función principal es la conversión de las señales correspondientes a X y Y en ángulos "*Pitch*" y "*Roll*", que luego son utilizados como referencias. Estos valores de referencia, junto con las señales que retroalimentan los ángulos estimados en tiempo real, son transmitidos al controlador secundario, identificado como *attitude*. La implementación en cascada se puede observar en la Figura 12.

Figura 12. Diagrama general del control del dron



La principal ventaja identificada en el uso del control en cascada radica en su capacidad para actuar con rapidez ante cualquier cambio en las medidas de los ángulos que puedan alterar la orientación del dron. El controlador secundario responde de manera ágil eliminando cualquier perturbación antes de que esta tenga la oportunidad de influir en las mediciones de posición y, como resultado, en el seguimiento del vehículo.

4.3 SELECCIÓN DEL ENFOQUE DE CONTROL ROBUSTO

El Control Lineal Cuadrático Gaussiano (LQG) presenta varias ventajas distintivas sobre otros métodos de control robusto. En primer lugar, el LQG persigue un rendimiento óptimo al minimizar tanto la variabilidad del estado del sistema como el esfuerzo de control, resultando en un desempeño globalmente eficiente. En segundo lugar, este enfoque permite lograr un equilibrio flexible entre la precisión en el seguimiento de referencia y la capacidad para atenuar perturbaciones, lo que lo convierte en una elección adecuada para sistemas que requieren respuestas rápidas y precisas en presencia de incertidumbres. El control robusto se diseñó para la actitud (*Attitude*) del dron ya que esta parte contiene los principales estados para garantizar un vuelo estable y preciso,

permitiendo ajustes y maniobras controladas de su orientación en el espacio tridimensional.

Enfocar el control robusto en la *Attitude* de un dron es esencial debido a la alta influencia que esta tiene en la estabilidad, maniobrabilidad y precisión del vuelo. La *Attitude* es definida por los ángulos de Euler que describen la orientación del dron. Por otro lado, determina los comandos de control y enfrenta las perturbaciones del entorno. Dado que los drones a menudo operan en condiciones cambiantes y desafiantes, como vientos cruzados o turbulencias, el control robusto de la *Attitude* permite mantener una orientación precisa con alta resistencia a perturbaciones y ruido en los sensores. Este enfoque robusto asegura que el dron pueda mantener su orientación y estabilidad incluso en situaciones imprevistas, garantizando un vuelo seguro y controlado en una variedad de escenarios operativos.

4.4 DISEÑO DEL CONTROLADOR LQG

4.4.1 Representación del espacio de estado

Para aplicar técnicas avanzadas de control y diseño es importante primero establecer el espacio de estados del modelo dinámico del cuadricóptero. Al utilizar el espacio de estados, podemos describir el sistema en términos de variables de estado, entradas y salidas, lo que nos permite analizar su estabilidad, respuesta transitoria y capacidad de control.

$$\dot{x} = f(x, u) \quad (98)$$

Donde el vector de estados está expresado como:

$$X = [\phi \ \theta \ \psi \ \dot{\phi} \ \dot{\theta} \ \dot{\psi} \ Z \ X \ Y \ \dot{Z} \ \dot{X} \ \dot{Y}]^T \quad (99)$$

El vector de entradas:

$$U = [U_1 \ U_2 \ U_3 \ U_4]^T \quad (100)$$

Y las ecuaciones:

$$x_1 = \phi ; x_2 = \theta ; x_3 = \psi ; x_4 = \dot{\phi} ; x_5 = \dot{\theta} ; x_6 = \dot{\psi} ; \quad (101)$$

$$x_7 = Z ; x_8 = X ; x_9 = Y ; x_{10} = \dot{Z} ; x_{11} = \dot{X} ; x_{12} = \dot{Y} ;$$

$$\dot{x} = F(x, u) = \left\{ \begin{array}{c} x_4 \\ x_5 \\ x_6 \\ x_5 x_6 \frac{I_{YY} - I_{ZZ}}{I_{XX}} + \frac{lU_2}{I_{XX}} \\ x_4 x_6 \frac{I_{ZZ} - I_{XX}}{I_{YY}} + \frac{lU_3}{I_{YY}} \\ x_4 x_5 \frac{I_{XX} - I_{YY}}{I_{ZZ}} + \frac{U_4}{I_{ZZ}} \\ x_{10} \\ x_{11} \\ x_{12} \\ g + (\cos x_1 \cos x_3) \frac{-U_1}{m} \\ (\cos x_1 \sin x_2 \cos x_3 + \sin x_1 \sin x_3) \frac{-U_1}{m} \\ (\cos x_1 \sin x_2 \sin x_3 - \sin x_1 \cos x_3) \frac{-U_1}{m} \end{array} \right\} \quad (102)$$

4.4.2 Linealización del modelo

La linealización del modelo dinámico del cuadricóptero es esencial para poder emplear el control LQG (Linear Quadratic Gaussian). Dado que el modelo no es lineal, la linealización nos permite aproximar su comportamiento alrededor de un punto de operación específico.

El punto de operación se ubica en cero para simplificar el análisis y el diseño del controlador. Al igualar el punto de operación a cero, estamos seleccionando un estado de equilibrio para que el sistema se encuentre en reposo o en un estado estable. Los valores para este punto de operación se encuentran en el Cuadro 2.

Cuadro 2. Valores de equilibrio de los estados y entradas

Estados	P. Equilibrio
$x_1, x_2, x_3, x_4, x_5, x_6, x_{10}, x_{11}, x_{12}$	0
x_7, x_8, x_9	Cualquier valor
Entradas	P. Equilibrio
U_2, U_3, U_4	0
U_1	$m * g$

La linealización de un sistema no lineal requiere el uso del Jacobiano. Este concepto se implementó escribiendo las ecuaciones del sistema y utilizando la función Jacobiano de Matlab. Esta función se utiliza para calcular las derivadas parciales de las ecuaciones en un punto de operación específico, lo que nos permite aproximar localmente el sistema no lineal por medio de un sistema lineal. Al utilizar la función Jacobiano de Matlab, se simplifica el proceso de obtener las derivadas parciales, con esto se facilita el proceso de análisis y diseño de controlador. En el Anexo A. Se encuentra el código donde se empleó esta función con el punto de operación del Cuadro 2. Los valores de x_7, x_8 y x_9 correspondiente a la ubicación del dron en el espacio y se tomaron como cero para mayor simplificación.

$$A = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -9.81 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 9.81 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (103)$$

$$B = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 303.6541 & 0 & 0 \\ 0 & 0 & 275.4237 & 0 \\ 0 & 0 & 0 & 3934.2 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -13.8889 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (104)$$

Considerando los ángulos “*Roll*”, “*Pitch*” y “*Yaw*” como salidas:

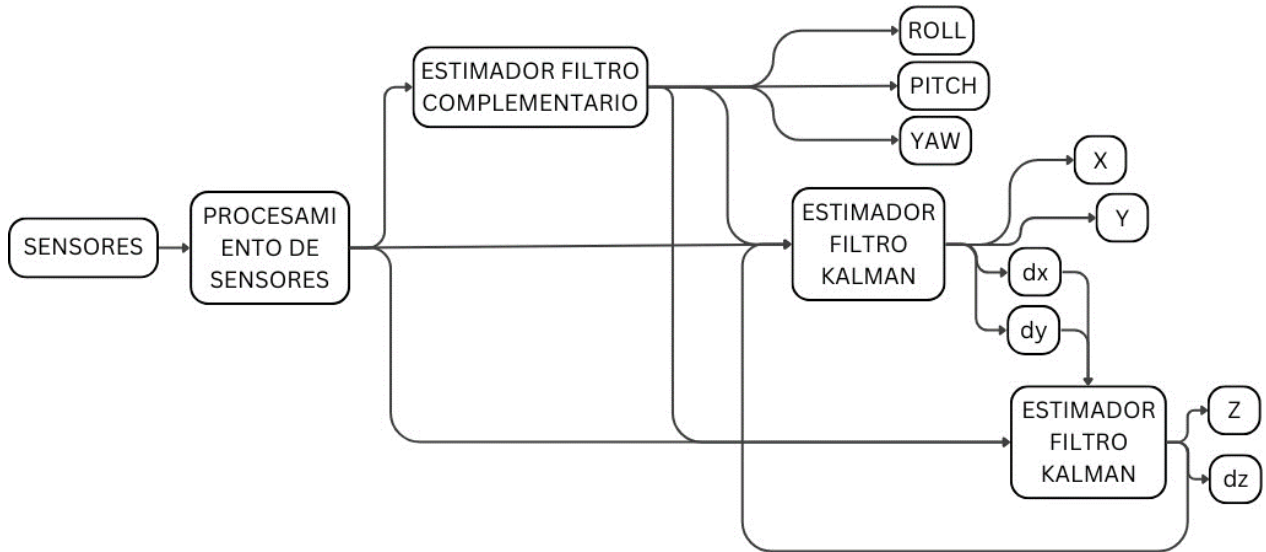
$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (105)$$

$$D = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (106)$$

4.4.3 Filtros Kalman

Dado que el dron está equipado con sensores que capturan la aceleración lineal en los ejes, la velocidad angular y la altitud del dron, es posible emplear estimadores con filtros Kalman para lograr una precisa estimación de la posición y la velocidad lineal. Por otro lado, como el giroscopio proporciona directamente información sobre la velocidad angular y el acelerómetro puede proporcionar la estimación de la orientación en relación con la gravedad, es factible estimar la orientación del dron mediante un filtro complementario sin requerir la aplicación Kalman. En la Figura 13 se puede observar la organización de los filtros Kalman.

Figura 13. Diagrama general de la aplicación de los filtros Kalman

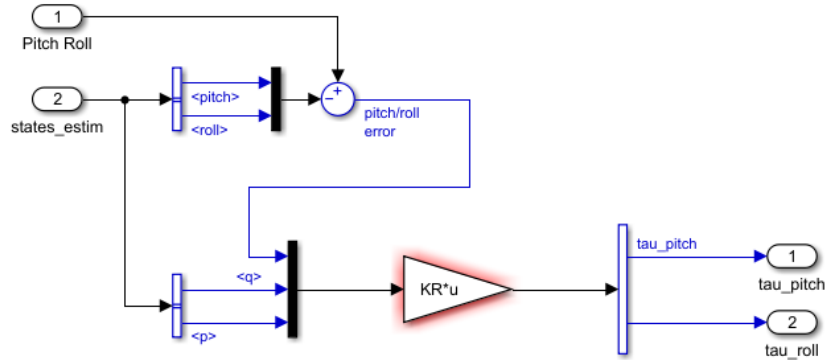


4.4.4 Cálculo de K con la Ponderación de Q y R

En la sección 3.4.1 se contextualizaron los términos Q y R para la solución de problemas de control óptimo lineal cuadrático, estos términos están relacionados con la función de costo y la penalización de las variables de estado y control en un sistema dinámico.

Con el modelo linealizado de la sección 4.4.2 y el código del Anexo A se obtiene la matriz de retroalimentación de estados K con los valores de Q y R asignados. La matriz K obtenida es de 4×12 ya que contiene las ganancias de todas las retroalimentaciones de estados. Como el control robusto va a estar enfocado en los movimientos x y y que se relacionan con los movimientos de ϕ y θ se reduce la matriz K a 2×4 .

Figura 14. Controlador LQR para Attitude



La matriz de ponderación de estados Q tiene elementos solo en la diagonal principal y se denota como:

$$Q = \text{diag}(Q_\phi, Q_\theta, Q_\psi, Q_{\dot{\phi}}, Q_{\dot{\theta}}, Q_{\dot{\psi}}, Q_Z, Q_X, Q_Y, Q_{\dot{Z}}, Q_{\dot{X}}, Q_{\dot{Y}}) \quad (107)$$

Donde $Q_\phi = 3.39$, $Q_\theta = 3.35$, $Q_{\dot{\phi}} = 0.01$, $Q_{\dot{\theta}} = 0.01$, $Q_{X,Y,\dot{X},\dot{Y}} = 0.0001$ y $Q_{\psi,\dot{\psi},Z,\dot{Z}} = 1$. Los valores de $Q_{\psi,\dot{\psi},Z,\dot{Z}}$ pueden tener cualquier valor ya que no tienen influencia sobre los ángulos ϕ y θ .

La matriz de ponderación de control R tiene elementos solo en la diagonal principal y se ponderaron los valores como:

$$R = \text{diag}(200 * [1 \ 1 \ 1 \ 1]) \quad (108)$$

Con estos valores de Q y R la matriz de realimentación de estados K queda:

$$K = \begin{bmatrix} 0.0128 & 0 & -0.0030 & 0 \\ 0 & 0.0128 & 0 & -0.0030 \end{bmatrix} \quad (109)$$

Los valores asignados a las matrices Q y R se escogieron evaluando la matriz resultante K con los índices de desempeño vistos en la sección 3.5. Las evaluaciones se llevaron a cabo realizando una trayectoria cuadrada de 1 metro de longitud a una altura de 0.6 metros y una trayectoria circular con diámetro de 1 metro a la misma altura.

5. IMPLEMENTACIÓN

5.1 HARDWARE Y SOPORTE PARROT MINIDRONES DE SIMULINK

El soporte Parrot Minidrones de Simulink es compatible con los minidrones Parrot Rolling Spider y Parrot Mambo. Para el diseño e implementación del controlador se usó el minidron Parrot Mambo. El Soporte de Simulink es fundamental para el diseño y desarrollo de algoritmos de control de vuelo, así como para acceder a los sensores incorporados en el dron. Gracias a la combinación de estas tecnologías, se puede implementar de manera efectiva el controlador y realizar pruebas de vuelo en un entorno simulado y real.

5.1.1 Minidron Parrot mambo

El minidron Parrot Mambo es un UAV con dimensiones de 18x18 mm (medida desde las protecciones de hélices) y peso de 72 g con batería y protecciones. La configuración del dron es de tipo X Quadcoptero. Este modelo de dron es considerado de bajo costo y es producido por la empresa francesa Parrot.

Figura 15. Dron Parrot Mambo y accesorios.



Fuente: Miguel Gonzáles de Filmora wondershare [En línea]. [Recuperado en 2023].
Disponible en:
<https://filmora.wondershare.es/drones/parrot-mambo-complete-review.html>

El diseño de este dron se centra en el mercado de los aficionados, con una batería de litio y un chasis resistente, aunque no es tan robusto como otros modelos. Incluye adaptadores como una garra y un cañón. En las pruebas de vuelo, el dron tiene un buen desempeño con excelentes comandos de despegue y aterrizaje. Su rango de control es limitado a menos de 70 metros y no cuenta con un extensor de señal²⁵. El dron Parrot Mambo está compuesto principalmente por:

2 motores A "Anti-Clock" estos motores giran en el sentido contrario a las agujas del reloj, se encuentran situados en el lado derecho de la sección frontal del dron y en la sección posterior en el lado izquierdo²⁶.

Figura 16. Motor A "Anti-Clock" coreless.



Fuente: Ro-botica [En línea]. [Recuperado en 2023]. Disponible en: <https://www.ro-botica.com/producto-Print/Motor-a-para-Parrot-Mambo/>.

2 motores C "Clock" estos motores giran en el sentido de las agujas del reloj, se encuentran situados en el lado izquierdo de la sección frontal del dron y en la sección posterior en el lado derecho²⁷.

²⁵ GONZÁLES, Miguel. Análisis completo de Parrot Mambo. [Sitio web]. [Consultado 09 junio 2023]. Disponible en: <https://filmora.wondershare.es/drones/parrot-mambo-complete-review.html>

²⁶ RO-BOTICA. Motor A para Parrot Mambo. [Sitio web]. [Consultado 09 junio 2023]. Disponible en: <https://www.ro-botica.com/producto-Print/Motor-a-para-Parrot-Mambo/>

²⁷ RO-BOTICA. Motor C para Parrot Mambo. [Sitio web]. [Consultado 09 junio 2023]. Disponible en: <https://www.ro-botica.com/producto-Print/Motor-c-para-Parrot-Mambo/>

Figura 17. Motor C "Anti-Clock" coreless.



Fuente: Ro-botica [En línea]. [Recuperado en 2023]. Disponible en: <https://www.ro-botica.com/producto-Print/Motor-c-para-Parrot-Mambo/>.

Ambos motores son de tipo Coreless y con unas dimensiones de 8,5 x 20 mm. Difieren en su configuración interna y el color de los cables de conexión. A diferencia de los motores convencionales, los motores coreless no tienen un núcleo ferromagnético en el rotor. En su lugar, utilizan un diseño sin núcleo, lo que los hace más ligeros y eficientes. Estos motores se caracterizan por su alta velocidad de respuesta, bajo nivel de ruido y un rendimiento suave y preciso. Son especialmente populares en aplicaciones donde se requiere un peso reducido y una alta eficiencia energética²⁸.

La **placa base** que se encarga de controlar la estabilización y el piloto automático de manera integrada. Está equipada con el chipset Parrot SIP6, que incluye un procesador Arm A9 de 800 MHz. Además, la placa base contiene a todos los sensores y cuenta con una capacidad de 1 GB de RAM y 256 MB de DDR²⁹.

Figura 18. Parrot Mambo SIP6 Linux Motherboard con 800MHz ARM A9.



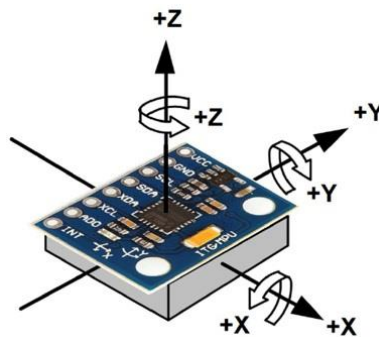
²⁸ INEED-MOTOR. ¿Qué es el motor sin núcleo? [Sitio web]. [Consultado 09 junio 2023]. Disponible en: <https://www.ineed-motor.com/news/what-is-coreless-motor-49234777.html>

²⁹ B&H. Parrot Mambo SIP6 Linux Motherboard con 800MHz ARM A9. [Sitio web]. [Consultado 09 junio 2023]. Disponible en: https://www.bhphotovideo.com/c/product/1274651-REG/parrot_pf070237_mambo_sip6_linux_motherboard.html

Fuente: B&H [En línea]. [Recuperado en 2023]. Disponible en: https://www.bhphotovideo.com/c/product/1274651-REG/parrot_pf070237_mambo_sip6_linux_motherboard.html.

Sensor IMU de referencia MPU-6050 fabricado por InvenSense³⁰. Este sensor cuenta con un sistema de detección de movimiento de 3 ejes. En su interior, se encuentra un procesador de movimiento digital (DMP) que se comunica con un acelerómetro de tres ejes y un giroscopio de tres ejes. Este sensor es usado para la estabilidad del dron.

Figura 19. MPU6050 Acelerómetro y giroscopio.



Fuente: NAYLAMP MECHATRONICS [En línea]. [Recuperado en 2023]. Disponible en: https://naylampmechatronics.com/blog/45_tutorial-mpu6050-acelerometro-y-giroscopio.html

Sensor de ultrasonido o **SONAR** (Sound Navigation and Ranging) que se emplea para proporcionar estabilidad vertical al dron. En este proyecto se hace uso de este sensor para calcular la distancia entre el dron y el suelo. Estos sensores funcionan midiendo el tiempo que tarda una onda de ultrasonido en viajar de ida y vuelta desde el sensor,

³⁰ CEPPI, Paolo. MODEL-BASED DESIGN OF A LINE-TRACKING ALGORITHM FOR A LOW-COST MINI DRONE THROUGH VISION-BASED CONTROL [En línea]. Maestría en Ciencias en Ingeniería Eléctrica e Informática. Chicago: Universidad de Illinois. 2020. p.35. [Consultado 08 junio 2023]. Disponible en: <https://webthesis.biblio.polito.it/16018/1/tesi.pdf>

calculando así la distancia. Actúan como generadores de señal y transductores. La altitud máxima que puede estimar es de aproximadamente 4 metros³¹.

La **Cámara** de 0.3 MP a 60 FPS ubicada en la parte inferior. Utiliza una técnica llamada flujo óptico para detectar los cambios de posición entre fotogramas y calcular la velocidad. Proporciona estabilidad horizontal (plano XY). El flujo óptico es el patrón de movimiento aparente de objetos y superficies en una escena debido al movimiento relativo entre el observador (cámara) y la escena.

Figura 20. Parte inferior del dron Parrot mambo.



Sensor de presión para detectar los cambios de altitud. Al medir la presión atmosférica, este sensor proporciona datos sobre los incrementos en la altitud del dron, permitiendo un control más preciso durante el vuelo. Este sensor y el sonar obtienen la información necesaria para ajustar la altitud de manera adecuada.

³¹ DRONE MANYA. Revisión del quadcopter Parrot Mambo. [Sitio web]. [Consultado 09 junio 2023]. Disponible en: <https://dronemanya.com/es/post/revision-del-quadcopter-parrot-mambo#i-8>

Batería LiPo de 660 mAh que proporciona 8 minutos de autonomía con accesorios y 10 minutos de autonomía sin accesorios. Carga en 30 minutos con un cargador de 2,1 Amperios³².

5.1.2 Soporte de Parrot minidrones de Simulink

El Soporte de Parrot minidrones de Simulink es un entorno de programación visual, que trabaja sobre el entorno de programación Matlab. Simulink cuenta con paquetes de soporte con diferentes herramientas de programación y control, una de ella es el soporte de Parrot minidrones, que sirve para diseñar, simular e implementar controladores en los minidrones Parrot Rolling Spider y Parrot Mambo. Este soporte de Simulink usa diagramas de la biblioteca de bloques para el diseño basado en modelos, admite la simulación del sistema, contiene generación automática de código, prueba y verificación.

El paquete de soporte de Simulink para minidrones Parrot permite diseñar algoritmos de control de vuelo y acceder a los sensores a bordo. Con Simulink Coder, es posible registrar datos de vuelo y generar código C a partir de modelos de Simulink. Además, se proporcionan herramientas complementarias, como Aerospace Blockset, para modelar ecuaciones de movimiento y simular el comportamiento de la aeronave en diferentes condiciones de vuelo. El paquete de soporte brinda la flexibilidad de mejorar los controladores de vuelo existentes o desarrollar nuevos algoritmos desde cero³³.

Para realizar simulaciones es necesario contar con una serie de complementos instalados. Estos complementos proporcionan herramientas y funcionalidades adicionales que amplían las capacidades de Simulink. A continuación, se mencionan los complementos que se deben tener instalados:

³² MANUALPDF. Manual Parrot mambo. [En línea]. 2016. p.34. [Consultado 09 junio 2023]. Disponible en: <https://www.manualpdf.es/parrot/mambo-fly/manual?p=2>

³³ MATHWORKS. Parrot Minidrones Support from Simulink. [Sitio web]. [Consultado 09 junio 2023]. Disponible en: <https://la.mathworks.com/hardware-support/parrot-minidrones.html>.

- Aerospace Blockset: Proporciona bloques y herramientas para modelar y simular sistemas aeroespaciales, como aeronaves y satélites.
- Optimization Toolbox: Permite realizar optimización de parámetros y diseño de control mediante técnicas de optimización.
- Simulink Control Design: Ofrece herramientas para el diseño y análisis de controladores, incluyendo técnicas de sintonización automática.
- Signal Processing Toolbox: Contiene algoritmos y herramientas para el procesamiento de señales, como filtrado, análisis espectral y transformadas.
- Computer Vision Toolbox: Proporciona algoritmos y herramientas para el procesamiento de imágenes y visión por computadora, permitiendo el análisis de datos visuales en las simulaciones.
- Simulink 3D Animation: Permite la visualización en 3D de los modelos y simulaciones en Simulink, creando entornos virtuales interactivos.

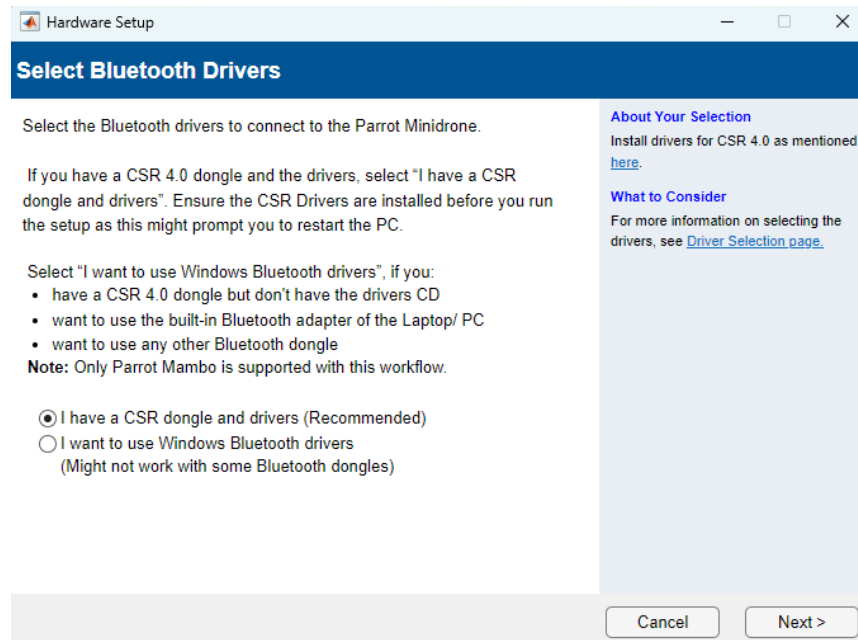
Además, para la implementación de carga de datos al dron, MathWorks desarrolló un firmware compatible para que se implemente el modelo creado en Simulink directamente en la placa del dron. Esto permite ejecutar el modelo en el dron mediante una conexión Bluetooth, brindando una solución efectiva en la integración de software y hardware. Para realizar esta conexión es necesario tener:

- Dron Parrot Mambo.
- Cable micro-USB.
- Bluetooth USB Dongle 4.0 con Drivers instalados (Este dispositivo es utilizado en caso de que la conexión Bluetooth de la computadora no logre establecer el enlace).
- El paquete de soporte de Simulink de Add-ons: “Simulink Support Package for Parrot Minidrones”.

A continuación, se detallan los pasos necesarios para establecer la conexión entre el PC y el dron:

Desde la ventana principal de Matlab se selecciona “Manage Add-Ons”, luego se abre el paquete de soporte “Simulink Support Package for Parrot Minidrones” y se inicia el procedimiento de conexión haciendo clic en el ícono de ajustes. Aparecerá la ventana de la Figura 21.

Figura 21. Hardware Setup.



Fuente: Simulink Support Package for Parrot Minidrones.

El paquete de soporte recomienda usar el Bluetooth CSR 4.0 dongle con sus drivers previamente instalados. Para este trabajo se estableció primero la conexión sin el uso del CSR 4.0 dongle y la conexión fue exitosa, luego para otras pruebas establecimos conexión con otro PC, este equipo si requirió el uso del CSR 4.0 ya que no eran compatibles.

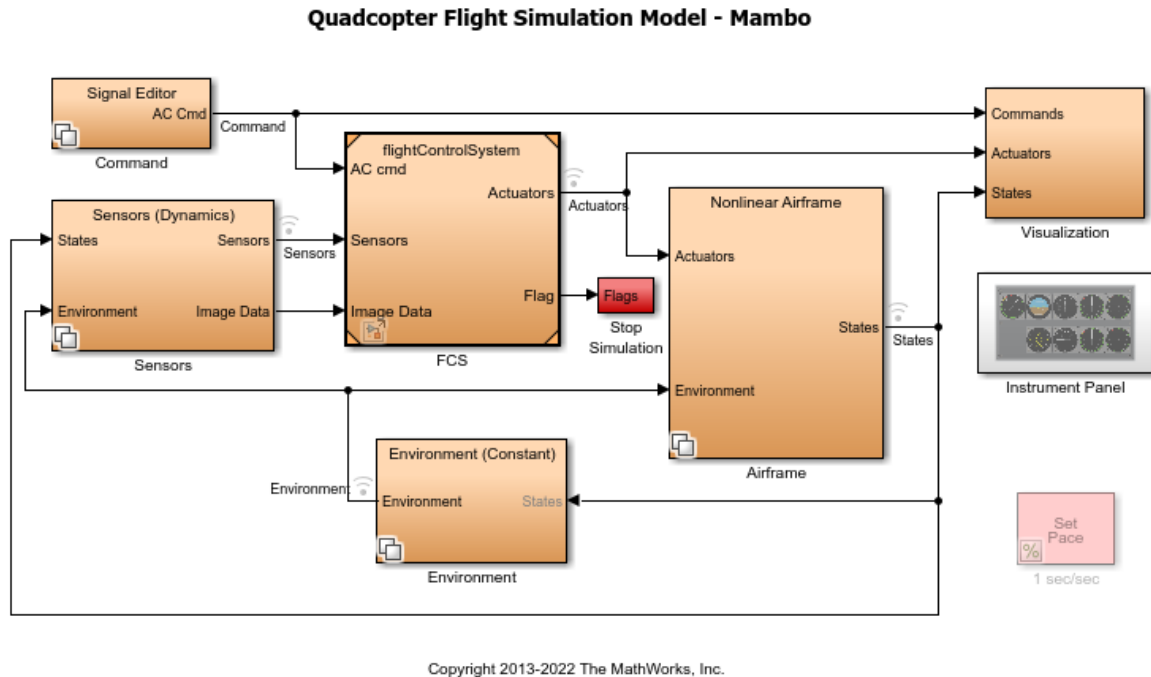
Luego de escoger el tipo de Bluetooth se conecta el dron con el cable USB y al cabo de unos segundos el PC realiza el reconocimiento del dron. El archivo firmware se carga en la memoria del dron en tres pasos: carga, desconexión USB y espera de instalación. Estos pasos están descritos sucesivamente al dar clic en “Siguiente”.

Posteriormente activado el Bluetooth, se selecciona “Unirse a una red de área personal” y se escoge el dispositivo “Mambo”. Una vez conectado, se hace clic en “Siguiente” y luego en “Finalizar” en la ventana abierta en el paso de instalación del firmware. Este procedimiento solo debe realizarse una vez. Si queremos conectar el dron en otros momentos, se debe solo realizar la conexión por la opción del Bluetooth “Unirse a una red de área personal”. Cualquier inconveniente presentado durante la instalación del firmware y conexión vía Bluetooth, el soporte de Simulink ofrece ayudas en todos los pasos.

5.1.3 Descripción del modelo de simulación

El proyecto se desarrolló dentro del entorno del modelo *asbquadcopter*, el cual forma parte de la competencia de minidrones de MathWorks. La competencia organizada por MathWorks, empleó el Soporte Parrot minidrones de Simulink para el desarrollo de los proyectos con el objetivo fomentar la innovación y la creatividad al utilizar las capacidades de los drones en combinación con las herramientas y tecnologías proporcionadas por MathWorks. Los participantes tuvieron la oportunidad de mostrar sus habilidades y conocimientos al diseñar y desarrollar proyectos utilizando los drones Parrot Mambo y las soluciones de MathWorks.

Figura 22. Modelo asbquadcopter.



FUENTE: Simulink.

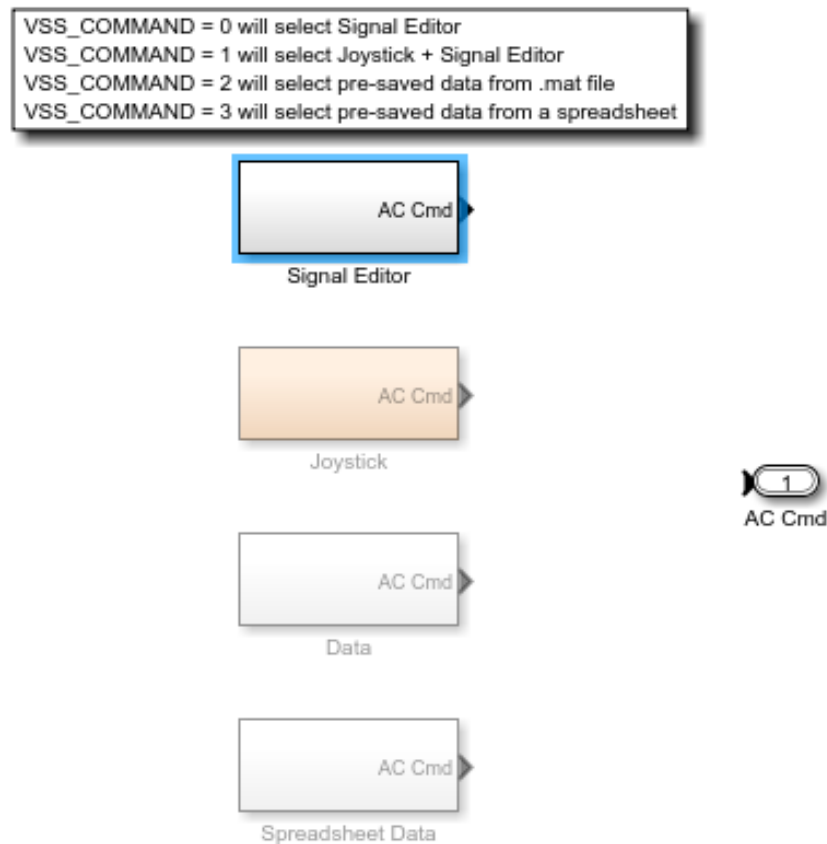
El modelo original *asbquadcopter* se compone de 6 subsistemas interconectados, como se muestra en la Figura 22. Estos subsistemas trabajan en conjunto para permitir una simulación de vuelo. El subsistema que contiene la estructura de control es el *FCS* (*Flight control system*) y es el subsistema que se carga al dron. A continuación, se describe cada subsistema del modelo:

5.1.3.1 Command

En este subsistema se genera la trayectoria de 4 formas distintas como se ve en la Figura 23. *Signal editor* emplea señales de posiciones XYZ (*position*) o ángulos “*Roll*”, “*Pitch*” y “*Yaw*” modificables con un bloque de Simulink. La siguiente opción de *Joystick* es para un enviar señales “*Roll*”, “*Pitch*”, “*Yaw*” y “*throttle*” mediante un mando. La tercera opción de *Data*, usa un archivo .mat para enviar señales *position/attitude*. Finalmente, *Spreadsheet Data* también envía señales *position/attitude* pero con una hoja de datos .xlsx. Sin embargo, en el presente trabajo no se utiliza este bloque, ya que para realizar

la prueba de vuelo con el dron Parrot Mambo, es necesario que las señales de referencia estén dentro del subsistema que se carga en el dron, específicamente en el *FCS (Flight Control System)*.

Figura 23. Subsistema Command.

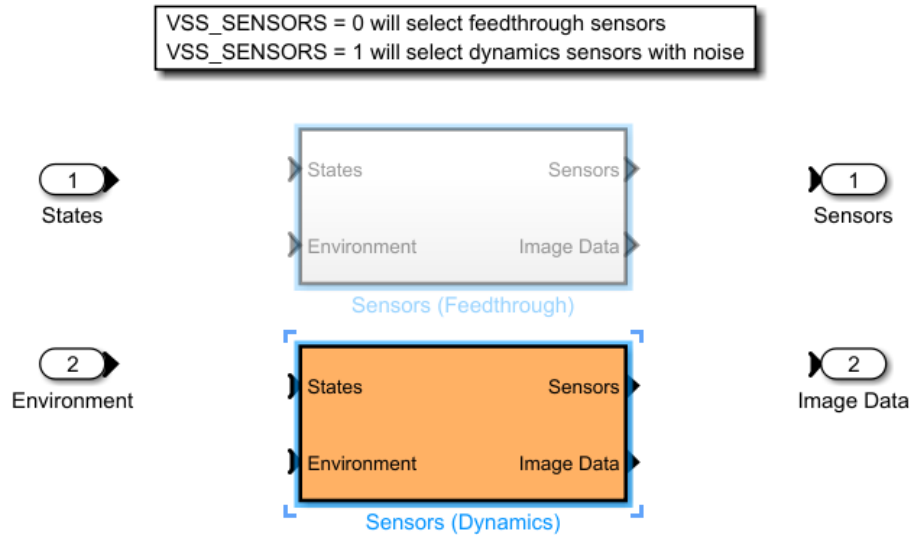


FUENTE: Simulink – Quadcopter Project.

5.1.3.2 Sensor

En este subsistema entran los estados y los datos del ambiente para que los sensores de presión, IMU y cámara puedan estimar las variables físicas y entregarlas al subsistema *FCS*. Se selecciona la segunda opción de sensores dinámicos con ruido para optar por una simulación real.

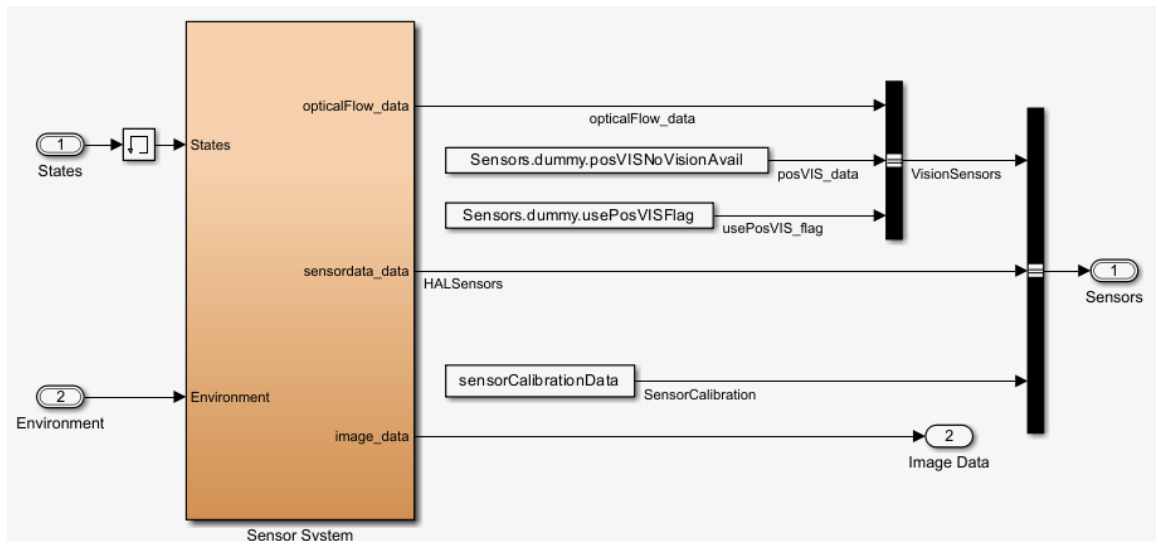
Figura 24. Subsistema Sensors.



FUENTE: Simulink – Quadcopter Project.

Los datos de los estados del sistema y los datos del medio ambiente son tomados por el bloque *Sensor system* para ser procesados por los sensores y luego entregar aceleraciones XYZ, velocidades angulares e imágenes a un *Bus creator* para simplificar la conexión de señales entre bloques y ayudar a mejorar la legibilidad y la comprensión del modelo.

Figura 25. Sensors (Dynamics).

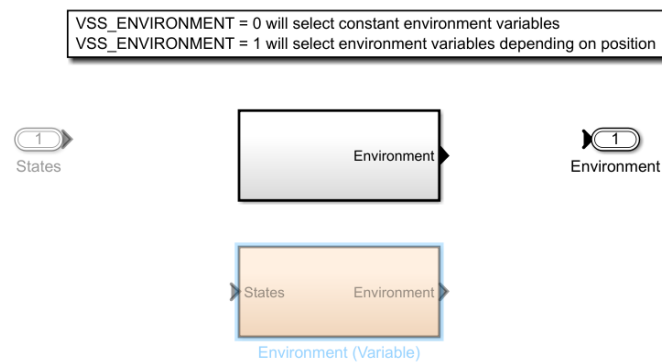


FUENTE: Simulink – Quadcopter Project.

5.1.3.3 Environment

Dentro de este bloque se encuentra la información de las variables con respecto al ambiente que genera la simulación, se puede elegir entre mantener unos valores constantes o que varíen.

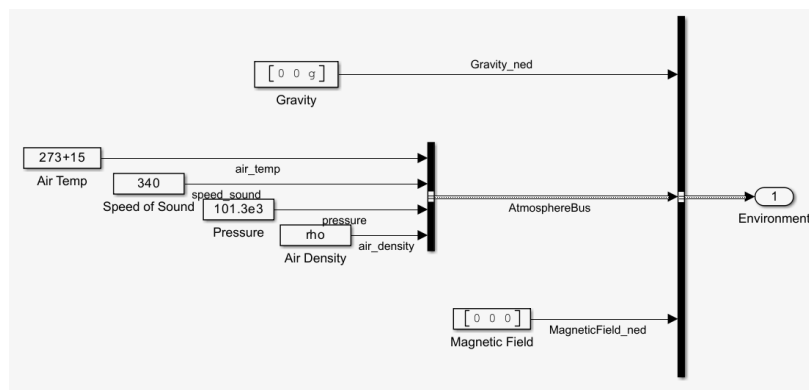
Figura 26. Subsistema Environment.



FUENTE: Simulink – Quadcopter Project.

En nuestro caso optamos por valores constantes durante las simulaciones para una mayor simplicidad y un menor gasto computacional.

Figura 27. Constantes Environment.

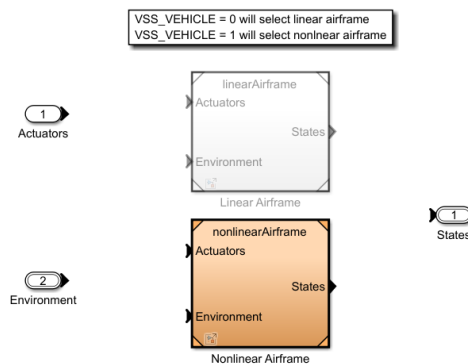


FUENTE: Simulink – Quadcopter Project.

5.1.3.4 Airframe

En este bloque se encuentra el modelo del dron, las entradas son las señales de los actuadores y la información proveniente del ambiente, por otro lado, en la salida se encuentran los estados resultantes.

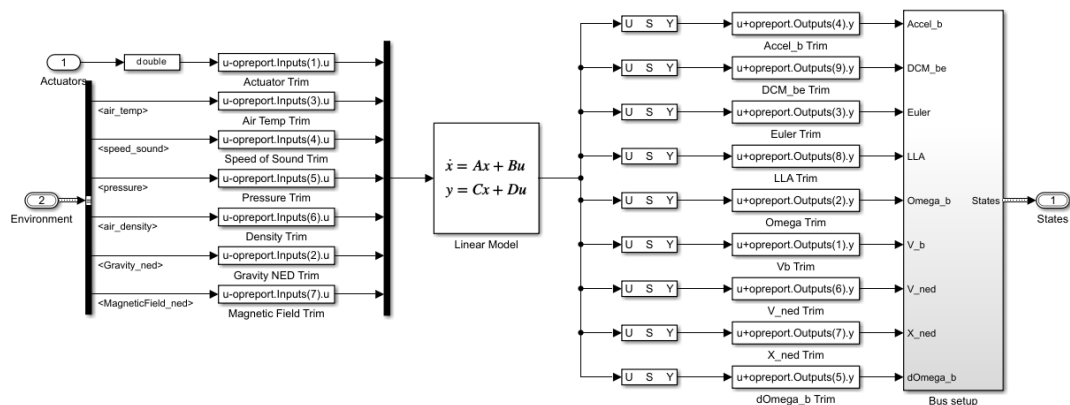
Figura 28. Subsistema Airframe.



FUENTE: Simulink – Quadcopter Project.

Se nos presenta la opción de elegir entre un modelo lineal y uno no lineal. El sistema lineal es el que usamos para las simulaciones de vuelo, así como también para la sintonización de los controladores.

Figura 29. Modelo lineal dron



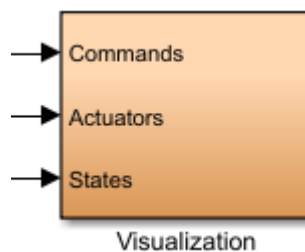
FUENTE: Simulink – Quadcopter Project.

El modelo lineal es representado con un bloque que contiene las matrices del espacio de estados. Además, mediante el uso de la herramienta “*trimLinearizeOpPoint*” del Simulink Control Design se obtiene el sistema no lineal.

5.1.3.5 Visualization

El subsistema *Visualization* traza la señal y controla el visualizador 3D mientras se ejecuta la simulación. Este subsistema usa Simulink 3D Animation. Permite crear entornos virtuales interactivos para visualizar y simular modelos dinámicos. Con esta herramienta, los usuarios pueden explorar y analizar el comportamiento del sistema mediante animaciones en tiempo real.

Figura 30. Subsistema Visualization.

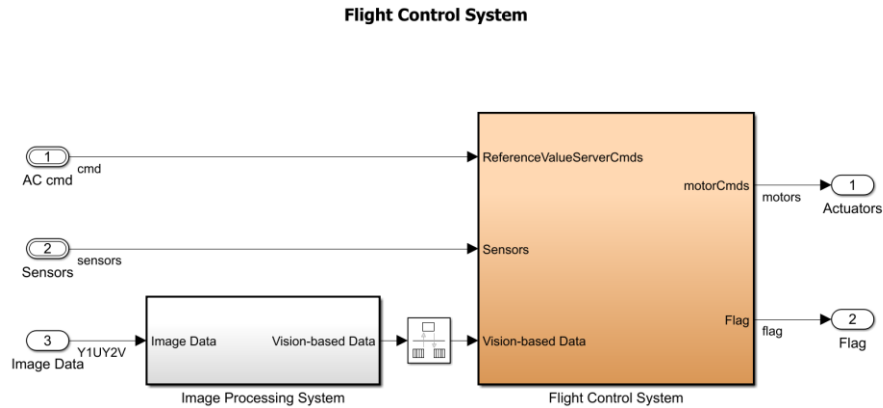


FUENTE: Simulink – Quadcopter Project.

5.1.3.6 Flight Control System

El subsistema de control de vuelo (*Flight control system*) es el bloque que se codifica y se carga en el dron. En él se encuentra principalmente el controlador y la estimación de estados, aquí se reciben las señales del comando y de los sensores (incluyendo el procesador de imágenes). De este bloque salen las instrucciones para los actuadores y las banderas rojas (parada de emergencia).

Figura 31. Subsistema Flight Control System



FUENTE: Simulink – Quadcopter Project.

Nuestro foco principal se dirige hacia este subsistema en particular. Aquí es donde desplegaremos los bloques y aplicaremos la teoría de control LQG (Control Lineal Cuadrático con Estimación de Kalman).

Para el procesamiento de imágenes se usa el bloque *Color Space Conversion* de Simulink, este bloque se convierte en una herramienta esencial en el sistema del dron al permitir la conversión de las imágenes RGB capturadas por su cámara al espacio de color YUV. Esta conversión resulta importante, ya que el espacio de color YUV separa la información de luminancia (Y) de la crominancia (U y V), lo que facilita el procesamiento de imágenes. El dron utiliza los datos en el espacio de color YUV y demás sensores para estimar su posición y velocidad con mayor precisión.

Dado que este subsistema es el que se carga al dron, es crucial que en él se encuentre un bloque encargado de recopilar todos los datos de vuelo relevantes. Este bloque adquiere un papel de vital importancia al capturar cada aspecto del vuelo, generando un conjunto de datos esenciales para posteriormente hacer su respectivo análisis. Este bloque se llama *Logging* y servirá tanto para obtener los datos de vuelo simulados como reales.

5.2 ESTIMACIÓN DE LOS PARÁMETROS

5.2.1 Estimación de inercias

Como lo descrito en la sección 4.1.2. Se realizó el montaje, la calibración y la obtención de los ángulos en función del tiempo de las oscilaciones realizadas por el péndulo barra y el sistema de péndulo barra dron. En las figuras 32, 33 y 34 se observan los montajes.

Figura 32. Montaje del sistema péndulo barra dron para el eje x.



Figura 33. Montaje del sistema péndulo barra dron para el eje y.



Figura 34. Montaje del sistema péndulo barra dron para el eje z.



Datos

- Masa del sistema péndulo dron, $m = 0.148\text{kg}$
- Distancia del centroide del sistema péndulo dron al centro de rotación, calculada con Solidworks, $r_c = 0.2825\text{m}$
- Gravedad, $g = 9.81\text{ m/s}^2$

La inercia del sistema péndulo dron I [kgm^2] se halló mediante un proceso de estimación de parámetros al igual que el coeficiente de amortiguamiento b [kgm^2/srad], esto se hizo introduciendo un torque externo $\tau = 0$ como dato de entrada ya que el péndulo se soltó libremente a un ángulo de entre $\sim 40^\circ$ y $\sim 50^\circ$. Este procedimiento se realizó para los 3 ejes del dron rotando y alineándolos. Para cada cálculo de inercia se realizó el experimento 5 veces. El código de la estimación de parámetros con la herramienta Grey Box Toolbox de Matlab y se encuentra en el Anexo B.

La inercia hallada fue $I_{\text{barra dron}}$ con respecto a su centroide, por lo tanto, se tuvo que hallar I_{barra} con el mismo método explicado anteriormente para así aplicar el teorema de los ejes paralelos y calcular la inercia del dron.

Cuadro 3. Inercia teórica y experimental de la barra

I BARRA	kg*m ²	ERROR
I TEÓRICO	0.000898	11%
I EXP.	0.00079545	

Conocidas las masas, las distancias de los centros de rotación (con ayuda de SolidWorks), y calculadas las inercias $I_{barra\ dron}$ y I_{barra} con Grey Box, se obtuvo I_{dron} de la ecuación (93) para cada eje.

5.2.2 Estimaciones de coeficientes de empuje y torque

Inicialmente se comprobó la precisión de los sensores que miden las rpm de cada motor, para esto se uso un tacómetro digital DT-2234B (Figura 35). El tacómetro se ubicó sobre la hélice y en el lado opuesto se ubicó la cinta reflectiva como se ve en la Figura 36. De esta forma, cada vez que la helice cubria la cinta reflectiva se cortaba la señal que emitia el tacómetro y se calculaban las rpm promedio. Para conocer el valor de las rpm se dividió en dos ya que la helice pasaba dos veces por cada ciclo.

Figura 35. Tacómetro DT-2234B



Figura 36. Montaje medición RPM



Al extraer los datos del *log* proporcionados por los sensores en el dron, los rpm de cada motor venían dados en términos de amplitud, por lo que se necesitó aplicar la conversión que aparece en la ecuación (110), extraída del soporte Parrot.

$$RPM_{TEÓRICA} = \frac{30}{\pi} \sqrt{A * 13841} \quad (110)$$

Para conocer el coeficiente de sustentación se requirió establecer la fuerza que ejercen los motores del dron cuando este intenta elevarse. Para esta tarea se diseñó un adaptador de base que sostuviese una celda de carga a una distancia prudente sobre el aire como se aprecia en la Figura 37. Se seleccionó una celda de carga de 1Kg (Figura 38) ya que el dron no ejerce fuerzas muy grandes y se utilizó un módulo de transmisor HX711 (Figura 39) para establecer la comunicación con el Arduino. El código de Arduino usado para esta prueba se encuentra en el Anexo D.

Figura 37. CAD Adaptador de base

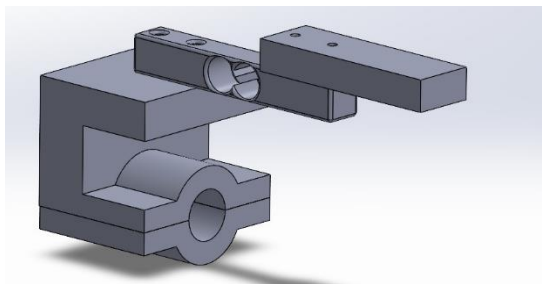


Figura 38. Celda de carga 1kg

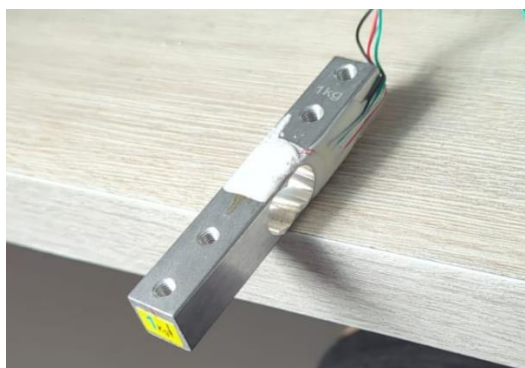
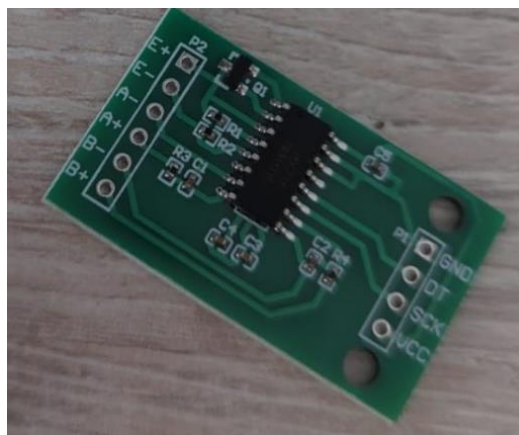


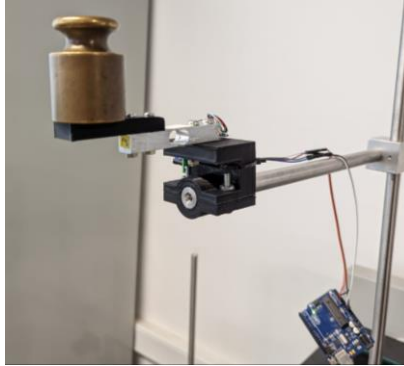
Figura 39. Modulo Transmisor HX711



Al instalar la celda de carga y antes de cada prueba era necesario calibrar la balanza, en este caso la calibración se realizó con una pesa de 500g como se observa en la Figura

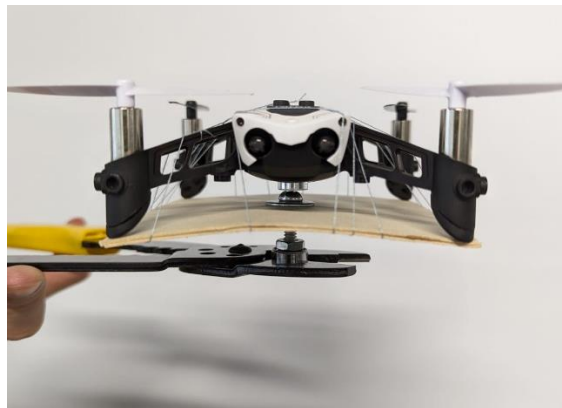
40. Posteriormente, se amarró el dron por medio de abrazaderas plásticas a la base diseñada (Figura 36) y se realizaron las pruebas de carga a diferentes amplitudes.

Figura 40. Calibración celda de carga



Para conocer el coeficiente de torque se requirió establecer la fuerza que ejercen los motores del dron cuando este intenta girar en “Yaw”. En este caso se usó el mismo mecanismo diseñado para el coeficiente de sustentación girado 90° tal como se observa en la Figura 42. El dron se sujetó por hilo a una lámina cuadrada de cartón paja de 7 cm, en el centro se encontraba un tornillo que tenía en su extremo inferior un rodamiento para permitir que el dron girara libremente como se muestra en la Figura 41. La razón de usar cartón paja e hilo para el montaje fue reducir el impacto de la masa sobre el experimento ya que estos son livianos.

Figura 41. Montaje de torque



Para el cálculo de la fuerza se activaron los dos motores que giran en la misma dirección y con un extremo del cuadro se hacía contacto con la celda de carga para así medir la fuerza generada. Tal como se representa en la Figura 42.

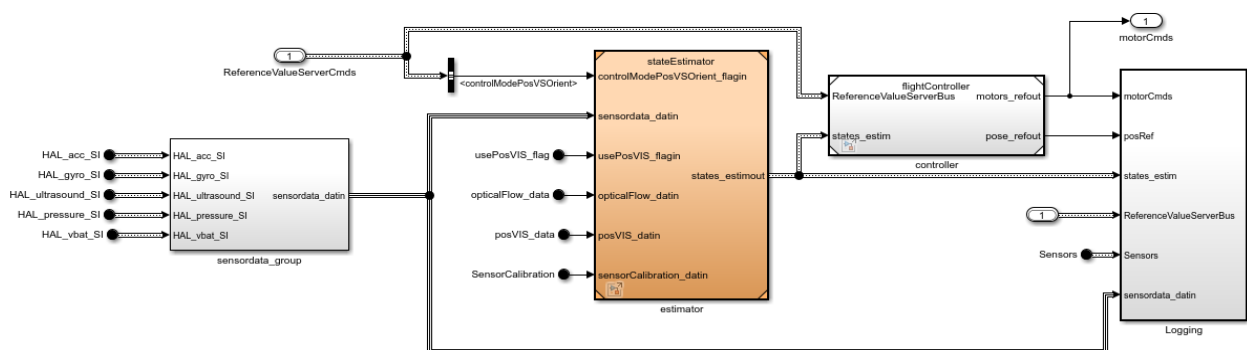
Figura 42. Montaje de medición de torque



5.3 IMPLEMENTACIÓN DEL CONTROLADOR LQG

Siguiendo la arquitectura establecida por el soporte Parrot para minidrones de Simulink de la sección 4.2 y aplicando las técnicas descritas en la sección 4.4, se procede a mostrar el controlador LQG diseñado en Simulink. Este controlador se encuentra ubicado en el subsistema *Flight control system*. En el interior de este subsistema se encuentran 3 subgrupos además del bloque *Logging* que servirá para guardar los datos de vuelo.

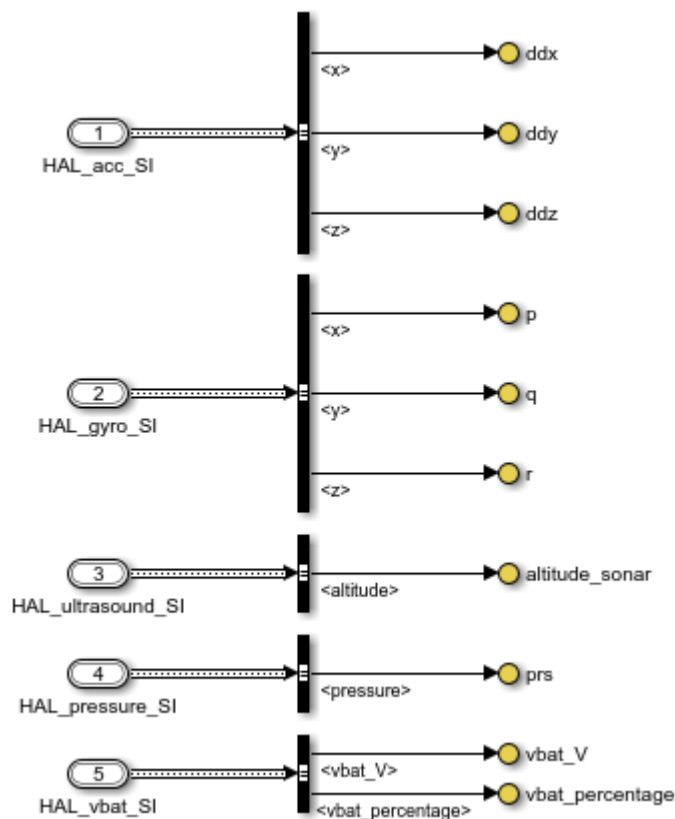
Figura 43. Interior del subsistema Flight Control System



5.3.1 Sensor data group

El subgrupo de *Sensor data group* contiene las entradas de los sensores para separar y tomar los datos necesarios. Los datos se encuentran organizados en la Figura 44 en orden de aceleraciones lineales, velocidades angulares, altitud, presión atmosférica, voltaje de la batería y su porcentaje. Estos datos de los sensores son enviados al subgrupo *Estimator*.

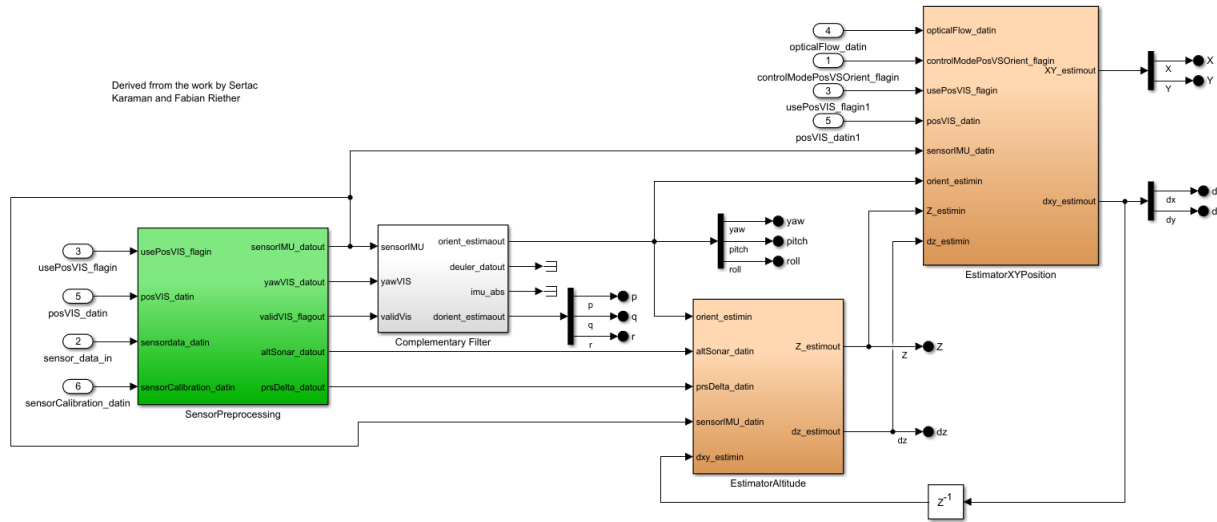
Figura 44. Subgrupo Sensor data group



5.3.2 Estimator

Se encarga de calcular y estimar la posición y velocidad del dron en tiempo real. Utiliza datos provenientes de los sensores a bordo, como el IMU (Unidad de Medición Inercial), sensor sonar y cámara. Estos sensores junto con algoritmos y filtros Kalman determinan con precisión la ubicación y la velocidad del dron mientras este se encuentra en vuelo.

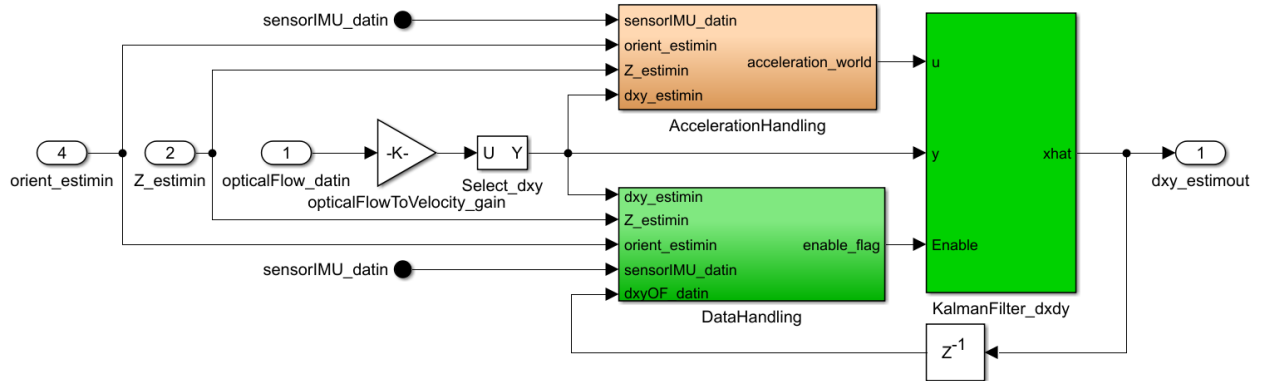
Figura 45. Subgrupo Estimator



Los estados de velocidades y posiciones son estimados mediante el uso de filtros Kalman. En las figuras 46, 47 y 48 se observa como por medio de las aceleraciones provistas por el IMU y con la información de sensores como los sensores de sonar y la cámara se logra extraer las velocidades y las posiciones respectivas. Un punto importante es que para estimar las posiciones de X y Y primero aparece un filtro Kalman en la estimación de sus velocidades. Estos filtros fueron desarrollados por Matlab.

Las matrices A, B, C, D son las que establecen el modelo de cada mini sistema. Las matrices G, H, Q, R y N son las que establecen las características del ruido presente en los datos. Sus valores ya fueron previamente estudiados y definidos por el soporte Parrot y por lo tanto se respetan estos valores. Estos fueron seleccionados porque es la desviación de ruido que se tiene en el sistema.

Figura 46. Filtro Kalman en la estimación de velocidades X y Y



En este caso en particular, el sistema que describen las matrices A, B, C y D en la ecuación (111). es un sistema simplificado en donde lo que entra es lo mismo que sale. Solo que la salida está integrada. El valor de 0.005 es el tiempo de muestreo de los sensores. La matriz G y H es la versión discreta del sistema y es un parámetro que requiere el bloque Filtro Kalman de Simulink. La matriz Q es la varianza del proceso y tiene un valor de 0.09, R es la varianza del ruido del sensor y tiene un valor de 5 y, por último, N es igual a 0 ya que no existe ninguna correlación entre el ruido del proceso y el ruido de la medición. Estos valores los estableció el soporte Parrot de Matlab estudiando los fenómenos de ruido de los sensores en cada caso particular.

$$A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad B = \begin{bmatrix} 0.005 & 0 \\ 0 & 0.005 \end{bmatrix}$$

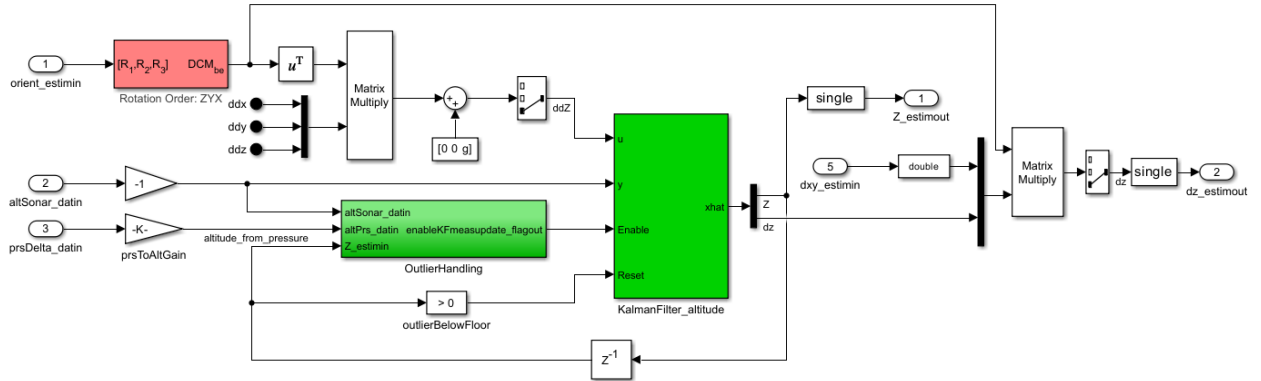
$$C = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad D = 0$$

(111)

$$G = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad H = 0 \quad Q = \begin{bmatrix} 0.09 & 0 \\ 0 & 0.09 \end{bmatrix}$$

$$R = \begin{bmatrix} 5 & 0 \\ 0 & 5 \end{bmatrix} \quad N = 0$$

Figura 47. Filtro Kalman en la estimación de posición y velocidad en Z



$$A = \begin{bmatrix} 1 & 0.005 \\ 0 & 1 \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ 0.005 \end{bmatrix}$$

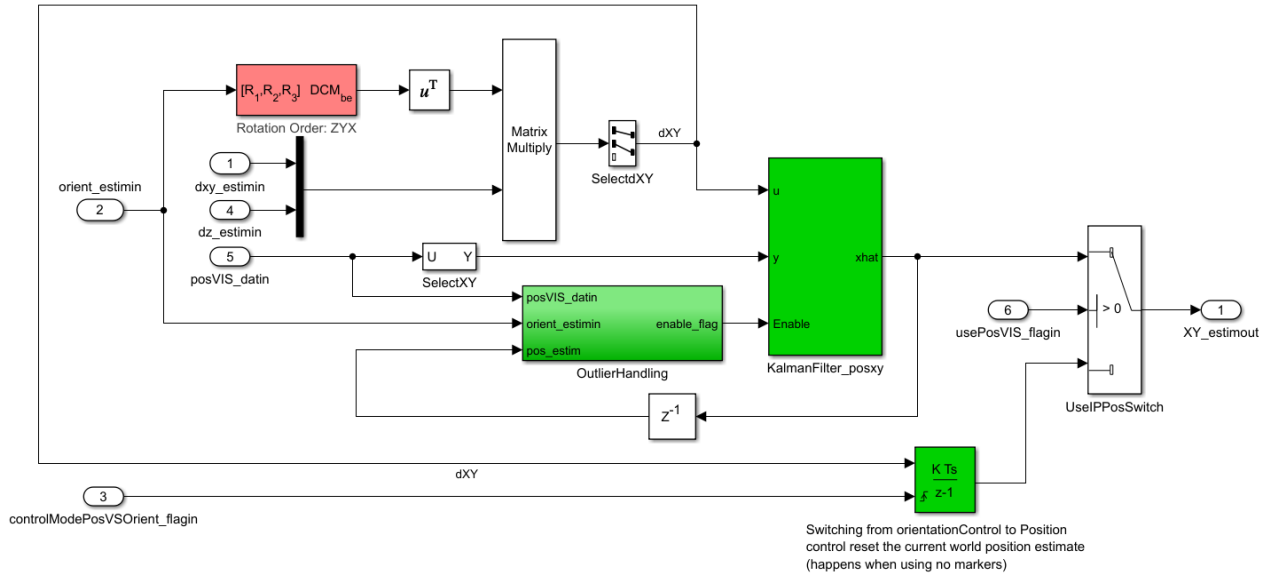
$$C = \begin{bmatrix} 1 & 0 \end{bmatrix} \quad D = 0$$

$$G = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad H = 0 \quad Q = 0.0005$$

$$R = 0.1 \quad N = 0$$

(112)

Figura 48. Filtro Kalman en la estimación de posición en X y Y



$$A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad B = \begin{bmatrix} 0.005 & 0 \\ 0 & 0.005 \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad D = 0$$

(113)

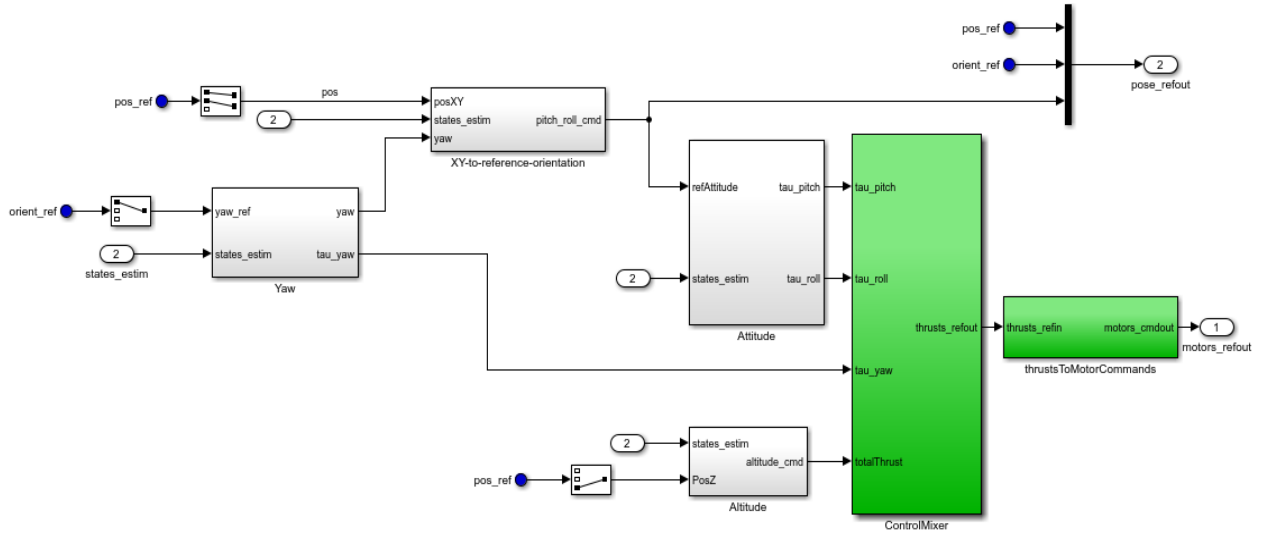
$$G = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix} \quad H = 0 \quad Q = \begin{bmatrix} 0.001 & 0 \\ 0 & 0.001 \end{bmatrix}$$

$$R = \begin{bmatrix} 0.3 & 0 \\ 0 & 0.3 \end{bmatrix} \quad N = 0$$

5.3.3 Controller

En el subgrupo *Controller* se encuentran 6 bloques que desempeñan funciones vitales para el control de vuelo del dron.

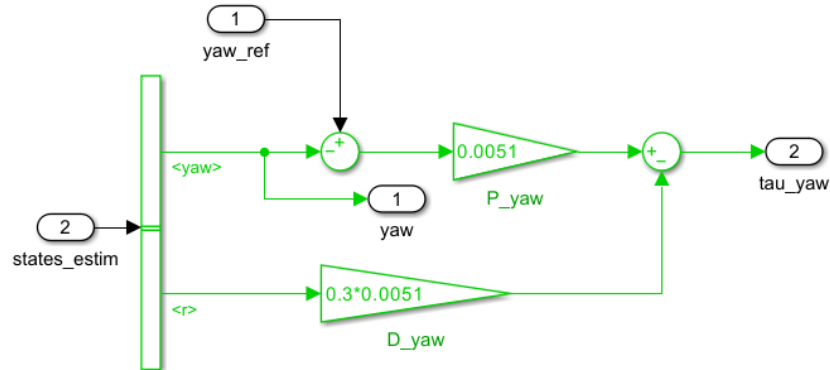
Figura 49. Subgrupo Controller



5.3.3.1 Yaw

Este bloque contiene un controlador PD para el ángulo “yaw”. El controlador PD recibe la señal de referencia introducida y lo convierte en U_4 .

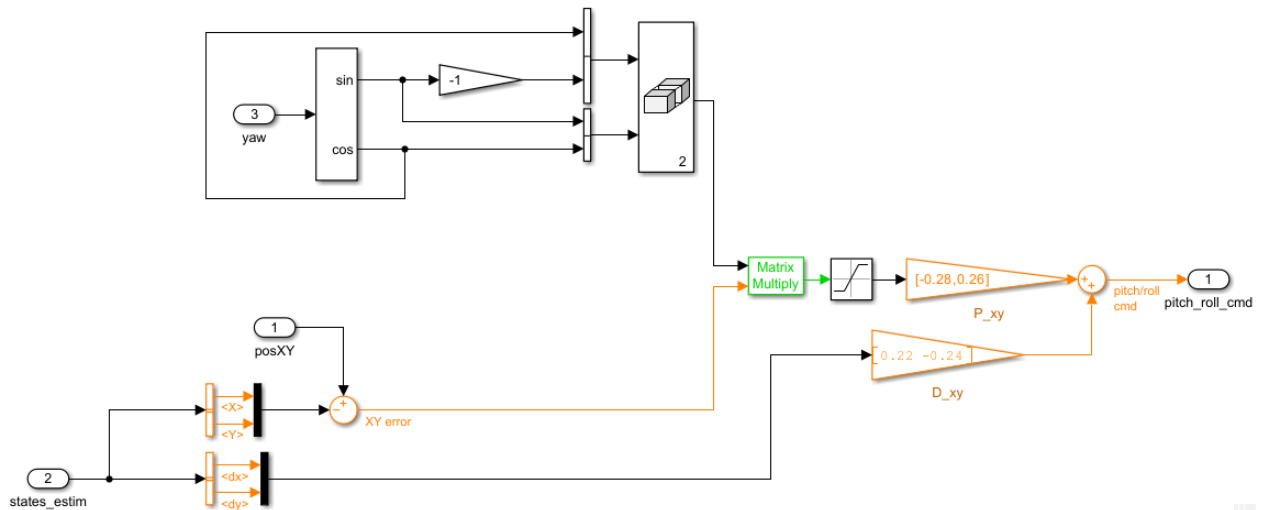
Figura 50. Bloque Yaw.



5.3.3.2 XY to reference orientation

Este bloque convierte la señal XY a referencia de ángulo *pitch/roll* con un controlador PD. Debido a que los ejes de referencia fijadas al dron rotan de acuerdo con “yaw”, se debe tener en cuenta la rotación con una matriz de concatenación.

Figura 51. Bloque XY to reference orientation

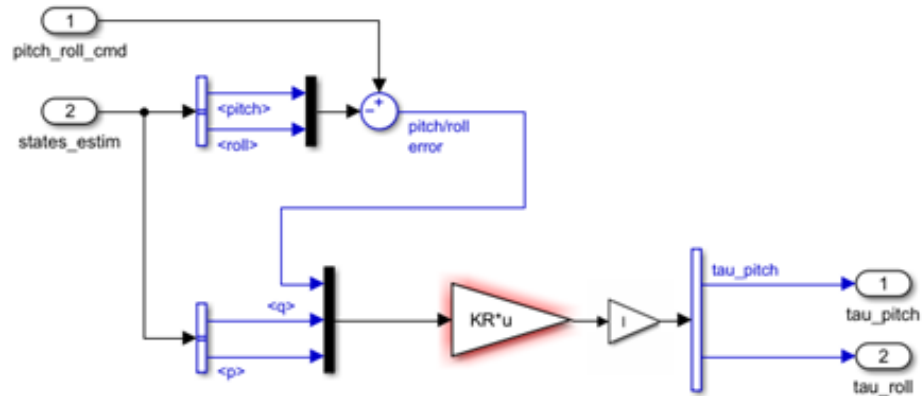


5.3.3.3 Attitude

Para crear el giro de *pitch/roll* se deben generar las fuerzas U_2 y U_3 que multiplicadas por I se convierten en torque (que es lo que se necesita como entrada al Controlmixer). La

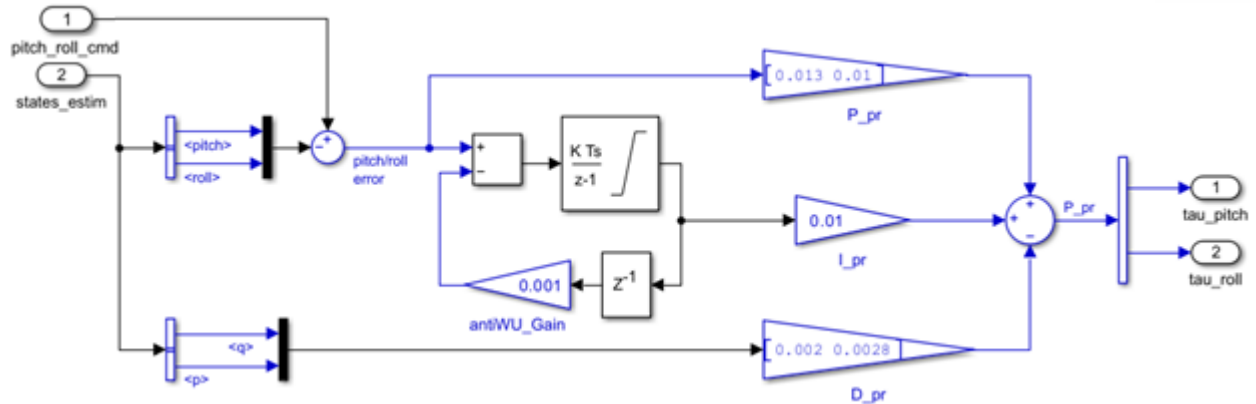
señal de *roll/pitch* generada en el bloque *XY to reference orientation* entra como *pitch_roll_cmd*. En este bloque se implementó el controlador LQR diseñado.

Figura 52. Bloque Attitude con controlador LQR



El soporte Parrot cuenta con un controlador PID ya sintonizado para la *attitude* del dron. La estructura del controlador y las ganancias se encuentran en la Figura 53. Este controlador también se implementó en conjunto con los filtros Kalman y demás controladores PD del subgrupo *Controller* para su posterior evaluación frente al LQR.

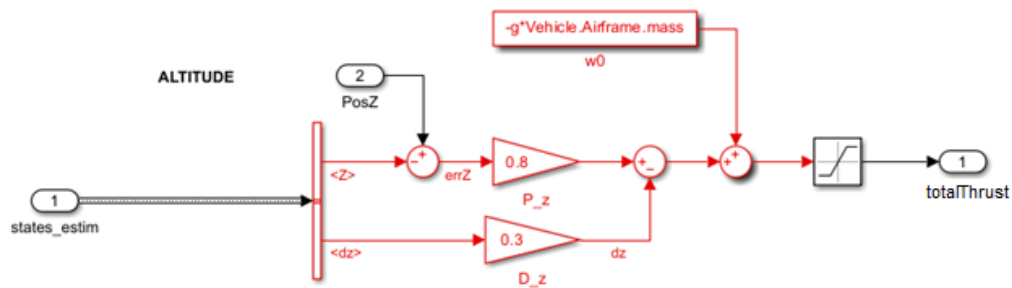
Figura 53. Bloque Attitude con controlador PID



5.3.3.4 Altitude

Este bloque contiene un controlador PD para la altitud del dron. El controlador recibe la señal de referencia introducida y lo convierte en U_I (totalThrust).

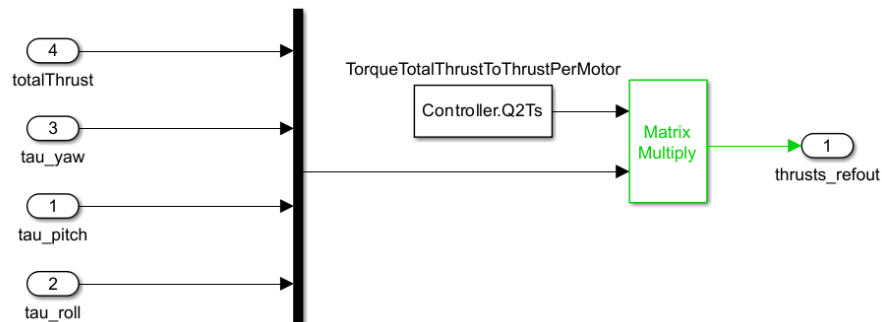
Figura 54. Bloque Altitude



5.3.3.5 Controlmixer

Las señales de torque deben pasar por una conversión de empuje y combinarse. Esta conversión fue proporcionada por el soporte Parrot.

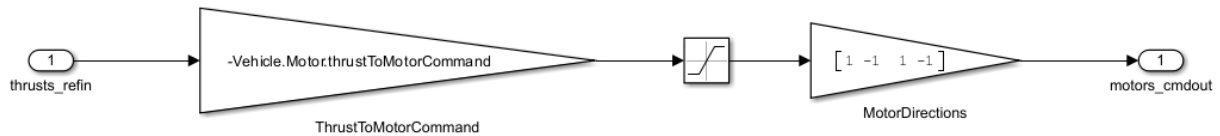
Figura 55. Bloque Controlmixer



5.3.3.6 ThrustsToMotorCommands

En el interior de este bloque se encuentra el proceso de traducción de empuje en señales de control específicas para cada motor con el fin de lograr el movimiento deseado. Las revoluciones máximas por minuto y el sentido de giro de cada motor están dadas por un saturador y un vector.

Figura 56. Bloque ThrustsToMotorCommands



5.4 PRUEBAS EXPERIMENTALES

5.4.1 Escenario de pruebas

En el escenario de pruebas se emplearon varios tipos de sabanas extendidas en el suelo, a través de un proceso de evaluación se seleccionó la sábana con mejor rendimiento en términos de capacidad de ubicación y seguimiento de la trayectoria. La elección del tipo de sábana adquiere una relevancia significativa en el control del dron, ya que este utiliza su cámara inferior para detectar y reconocer los patrones presentes. La precisión y la confiabilidad en la detección de patrones juegan un papel fundamental en la capacidad del dron para mantener su posición y ejecutar las trayectorias de manera efectiva, asegurando así un vuelo controlado y preciso en el entorno designado.

El patrón de suelo en la Figura 57 (con dimensiones 2m x 2.2m) tiene papeles blancos de referencia debido a que en otros trabajos realizados con el dron Parrot Mambo emplearon este tipo de patrón y colores. De este obtuvimos un comportamiento poco fiable ya que en múltiples ocasiones se perdía la adquisición de imágenes por parte de la cámara.

Figura 57. Patrón de suelo 1



El patrón de suelo de la Figura 58 con dimensiones 2.5m x 2.4m fue el peor ya que el dron no podía mantener su posición y se caía. A demás, desorientaba la cámara del dron a tal punto que este empezaba a realizar movimientos erráticos para intentar compensar las lecturas erróneas.

Figura 58. Patrón de suelo 2



Los mejores patrones fueron los de las Figura 59Figura 60Figura 60. En estos dos el dron se desempeñó de manera muy similar. Se escogió el suelo 3 ya que tenía una mayor área.

Figura 59. Patrón de suelo 3



Figura 60. Patrón de suelo 4



5.4.2 Pruebas de seguimiento de trayectoria

Llevamos a cabo pruebas de trayectoria específicas para evaluar el desempeño de los controladores PID y LQG. Estas pruebas incluyeron la ejecución de trayectorias tanto circulares como cuadradas. La elección de estos patrones de vuelo se realizó estratégicamente para abordar aspectos clave del control y la navegación del dron.

La trayectoria circular (Figura 61) se utilizó para evaluar la capacidad del dron de mantener una trayectoria precisa y estable durante giros continuos. Estas pruebas permitieron verificar la eficacia de nuestro controlador en situaciones que requieren una maniobrabilidad suave y precisa.

Por otro lado, la trayectoria cuadrada (Figura 62) se implementó con el propósito de evaluar la precisión en cambios bruscos de dirección y la capacidad del dron para seguir una ruta predeterminada de manera exacta. Estas pruebas nos proporcionaron información valiosa sobre la capacidad de nuestro controlador para realizar maniobras rápidas y mantener una trayectoria precisa. En conjunto, estas pruebas de trayectoria fueron fundamentales para afinar y perfeccionar el controlador del dron.

Figura 61. Trayectoria Circular Diámetro 1 m

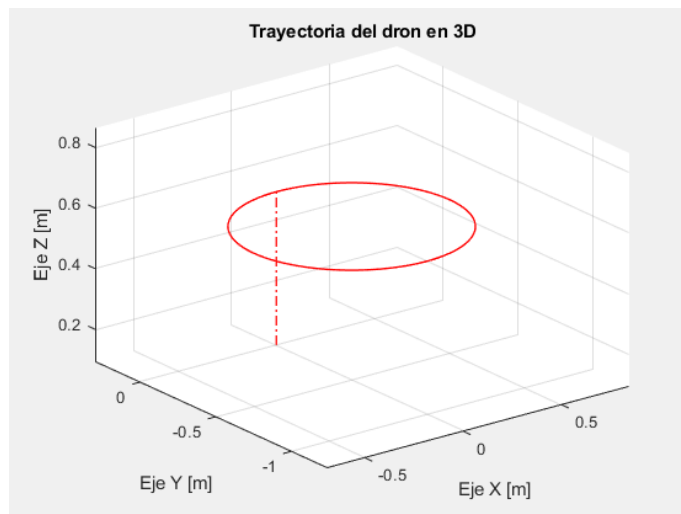
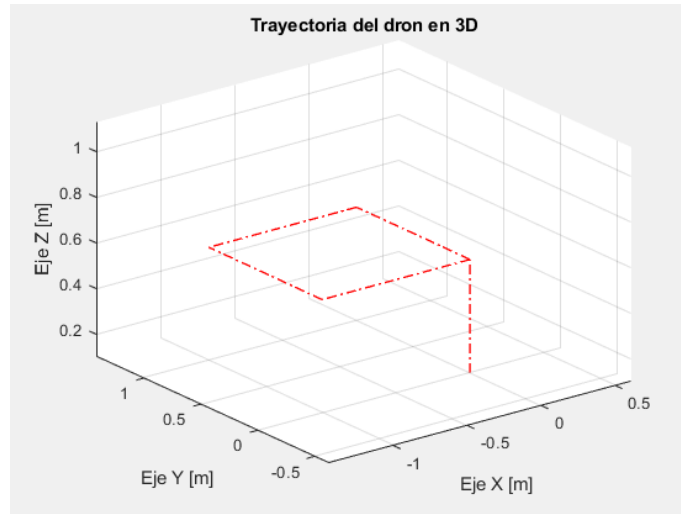


Figura 62. Trayectoria Cuadrada Lado 1 m



5.4.3 Cálculo índices de desempeño

Con la información de vuelo recopilada de las trayectorias circulares y cuadradas se evaluó la exactitud utilizando todos los índices de desempeño mencionados en la sección 3.5. A partir de los resultados obtenidos en estos índices, se llevó a cabo un proceso iterativo con el objetivo de determinar el coeficiente K óptimo que maximizara el rendimiento deseado.

Los índices de desempeño fueron calculados utilizando Matlab. El código desarrollado se encuentra disponible en el Anexo C, donde se pueden visualizar tanto las funciones como las fórmulas utilizadas. Además, en este mismo anexo, se incluye el código de las gráficas de los datos de vuelo recopilados para un análisis completo y visual.

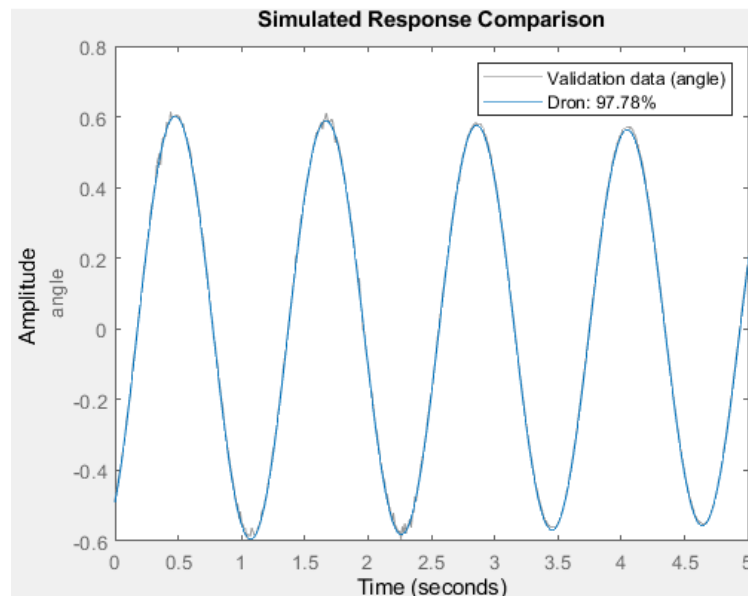
RESULTADOS

6.1 MODELADO E IDENTIFICACIÓN DE PARÁMETROS DEL SISTEMA DINÁMICO

La base fundamental para comprender y analizar el comportamiento de un UAV radica en los parámetros de su modelo dinámico. A continuación, se presentan los valores de los parámetros fundamentales del dron, los cuales fueron descritos y determinados experimentalmente en las secciones 4.1 y 5.1.

6.1.1 Momento de inercia del dron en X, I_{xx}

Figura 63. Resultado del Grey Box para el cálculo de la inercia I_{xx}

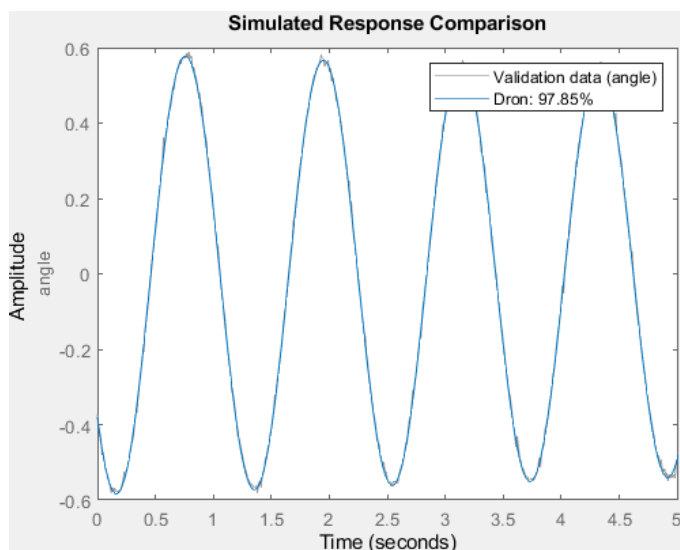


Cuadro 4. Inercia I_{xx} del Soporte Parrot minidrones de Simulink y experimental

	$\text{kg}\cdot\text{m}^2$	ERROR
I_{xx} SUPPORT	0.000058286	65%
I_{xx} EXP.	0.000166944	

6.1.2 Momento de inercia del dron en Y, I_{yy}

Figura 64. Resultado del Grey Box para el cálculo de la inercia I_{yy}

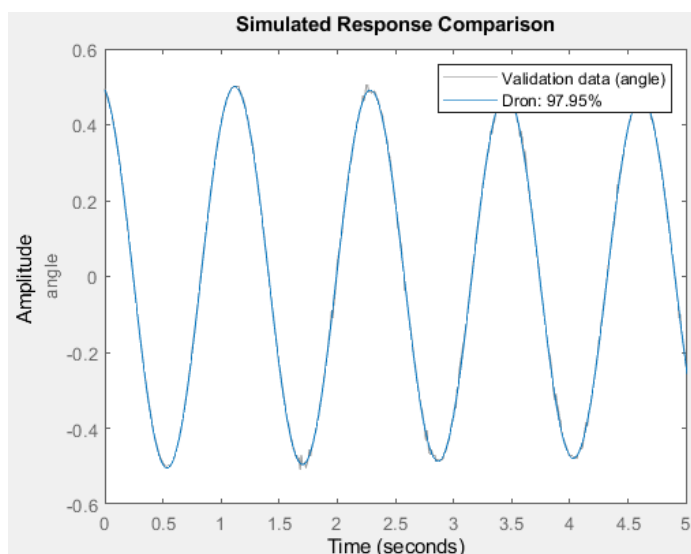


Cuadro 5. Inercia I_{yy} del Soporte Parrot minidrones de Simulink y experimental

	$\text{kg}\cdot\text{m}^2$	ERROR
I_{yy} SUPPORT	0.000071691	58%
I_{yy} EXP.	0.00017119	

6.1.3 Momento de inercia del dron en Z, I_{zz}

Figura 65. Resultado del Grey Box para el cálculo de la inercia I_{zz}



Cuadro 6. Inercia I_{zz} del Soporte Parrot minidrones de Simulink y experimental

	$\text{kg}\cdot\text{m}^2$	ERROR
I_{zz} SUPPORT	0.000100000	61%
I_{zz} EXP.	0.00025418	

La calidad de las inercias proporcionadas por el soporte Parrot presenta ciertos márgenes de error en relación con las inercias halladas. En particular, se identificaron errores de un 65% en la inercia en el eje X, un 58% en el eje Y y un 61% en el eje Z.

6.1.4 Coeficiente de empuje

La comparación entre las rpm experimentales y teóricas se aprecian en el Cuadro 7 para el motor posterior derecho y en el Cuadro 8 para el motor posterior izquierdo. El factor más influyente en su funcionamiento es la dirección en la que giran, y por lo tanto podemos decir que el motor izquierdo posterior es igual al derecho frontal y el motor derecho posterior es igual al izquierdo frontal como se observa en la Figura 66.

Figura 66. Sentido de giro de los motores



Cuadro 7. Comparación entre las rpm experimentales y teóricas motor posterior derecho

MOTOR POSTERIOR DERECHO			
AMPLITUD	EXPERIMENTAL [RPM]	TEÓRICA [RPM]	%ERROR
400	22984.5	22469.07	-2%
300	18961	19458.79	3%
200	13757	15888.03	15%
100	9048.5	11234.54	24%

Cuadro 8. Comparación entre las rpm experimentales y teóricas motor posterior izquierdo

MOTOR POSTERIOR IZQUIERDO			
AMPLITUD	EXPERIMENTAL [RPM]	TEÓRICA [RPM]	%ERROR
400	23569.5	22469.07	-5%
300	19595.5	19458.79	-1%
200	14013	15888.03	13%
100	9029.5	11234.54	24%

En ambos cuadros se aprecia como la precisión en los rpm se ve considerablemente reducida cuando la amplitud se encuentra en un rango por debajo de 200. En nuestro caso particular, cada prueba de vuelo los comandos de rpm siempre se mantuvieron en un rango de amplitud superior a 200 por lo que podemos decir que la aproximación teórica es fiable.

Cuadro 9. % Error producido del coeficiente de empuje de Simulink y el propuesto.

AMPLITUD	MASA [Kg]	FUERZA [N]	b [Ns ²]	b[Ns ²] SIMULINK	%ERROR SIMULINK	b[Ns ²] PROPUESTO	%ERROR PROPUESTO
400	0.101	0.99081	4.16822E-08	4.72E-08	13%	3.99E-08	-4%
300	0.066	0.64746	3.97048E-08		19%		0%
200	0.034	0.33354	3.94368E-08		20%		1%
100	0.016	0.15696	4.37956E-08		8%		-9%

Como se puede observar en el Cuadro 9 los errores llegan hasta un 20%. A partir de estos datos se propone un coeficiente de $3.99 \times 10^8 \text{ N s}^2$ que minimiza estos errores en el rango de trabajo normal del Parrot.

6.1.5 Coeficiente de torque

Como se puede observar en el Cuadro 10 los errores llegan hasta un 71%. A partir de estos datos se propone un coeficiente de $3.29 \times 10^8 \text{ Nms}^2$ que minimiza estos errores en el rango de trabajo normal del Parrot.

Cuadro 10. % Error producido del coeficiente de torque de Simulink y el propuesto.

AMPLI -TUD	MASA [g]	FUERZA [N]	TORQUE [N*m]	d[Nms ²]	d [Nms ²] SIMULINK	%ERROR SIMULINK	d [Nms ²] PROPUES	%ERROR PROPUESTO
400	5.886	0.05774	0.0040421	3.489E-10	1.139 E-10	-67%	3.29 E-10	-6%
300	3.600	0.03532	0.0024721	3.135E-10		-64%		5%
200	1.968	0.01931	0.0013515	3.256E-10		-65%		1%
100	1.034	0.01014	0.0007099	3.953E-10		-71%		-17%

6.1.6 Modelo Propuesto

Concluido el proceso de identificación y validación de todos los parámetros, procedemos a presentar las ecuaciones del modelo dinámico del dron desarrolladas en la sección 3.1 con sus parámetros propuestos. Con este modelo se obtiene una representación completa e intrínseca de su dinámica operativa.

$$\ddot{X} = (\sin \Psi \sin \Phi + \cos \Psi \sin \theta \cos \Phi) \frac{-U_1}{m} \quad (114)$$

$$\ddot{Y} = (-\cos \Psi \sin \Phi + \sin \Psi \sin \theta \cos \Phi) \frac{-U_1}{m} \quad (115)$$

$$\ddot{Z} = g + (\cos \theta \cos \Phi) \frac{-U_1}{m} \quad (116)$$

$$\ddot{\Phi} = \dot{p} = \frac{I_{YY} - I_{ZZ}}{I_{XX}} qr + \frac{lU_2}{I_{XX}} \quad (117)$$

$$\ddot{\theta} = \dot{q} = \frac{I_{ZZ} - I_{XX}}{I_{YY}} pr + \frac{lU_3}{I_{YY}} \quad (118)$$

$$\ddot{\psi} = \dot{r} = \frac{I_{XX} - I_{YY}}{I_{ZZ}} pq + \frac{U_4}{I_{ZZ}} \quad (119)$$

Cuadro 11. Parámetros propuestos

PARÁMETRO	VALOR	UNIDADES
m	0.072	kg
I_{XX}	0.000166944	kgm^2
I_{YY}	0.00017119	kgm^2
I_{ZZ}	0.00025418	kgm^2
b	3.99E-08	Ns^2
d	3.29E-10	Nms^2
l	0.0624	m

6.2 DESEMPEÑO DEL CONTROLADOR SIMULADO VS EXPERIMENTAL

Realizada la implementación de las pruebas tanto en el entorno de simulación en Simulink como en la vida real, a continuación, se presentan los datos de vuelo recopilados y los índices de desempeño obtenidos.

Trayectoria cuadrada simulada con LQG:

Figura 67. Vista isométrica y superior de la trayectoria cuadrada simulada LQG

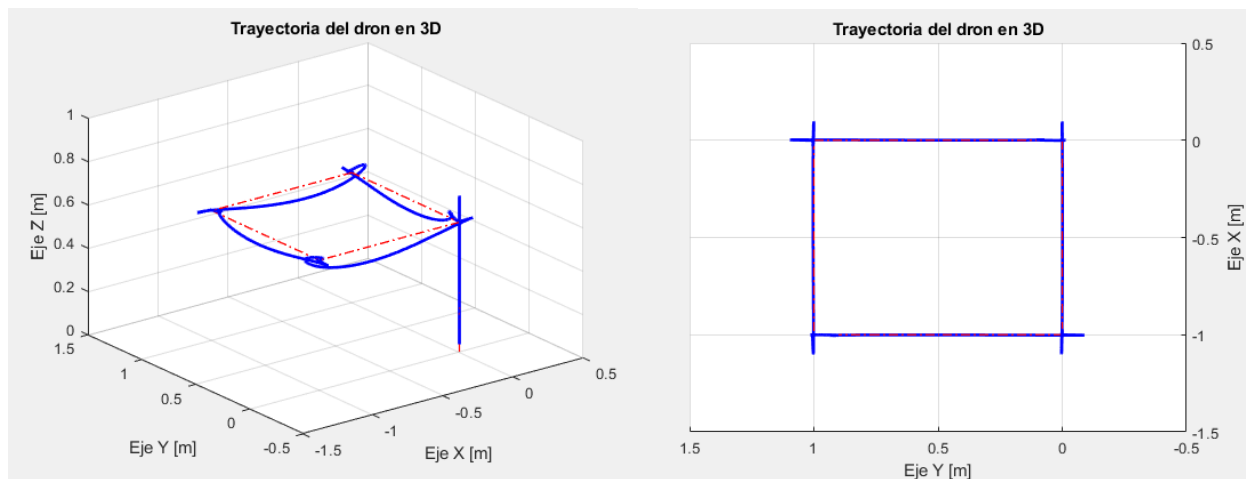


Figura 68. Datos posición de vuelo de la trayectoria cuadrada simulada LQG

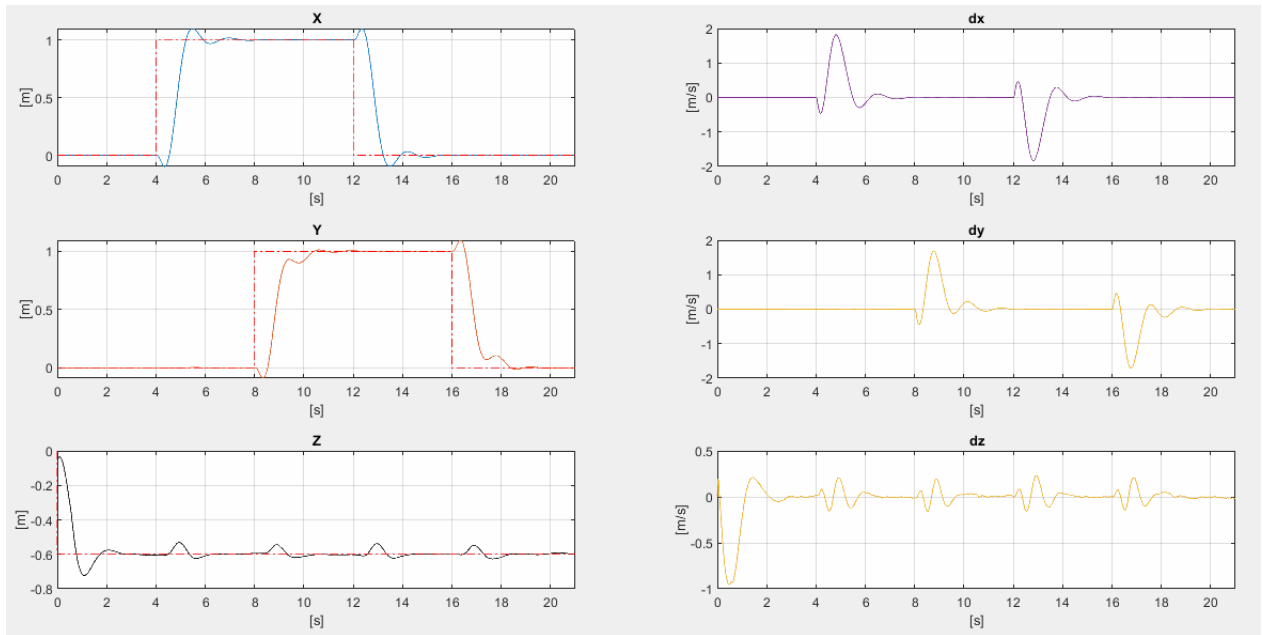
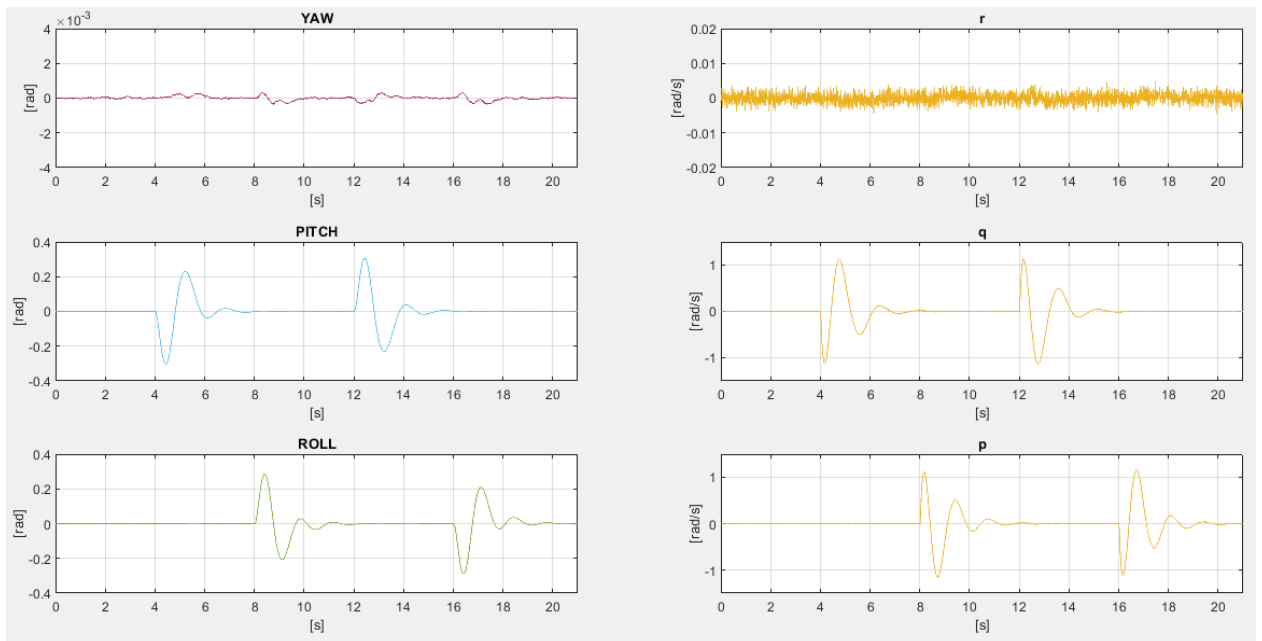


Figura 69. Datos orientación de vuelo de la trayectoria cuadrada simulada LQG



Trayectoria cuadrada experimental con LQG:

Figura 70. Vista isométrica y superior de la trayectoria cuadrada experimental LQG

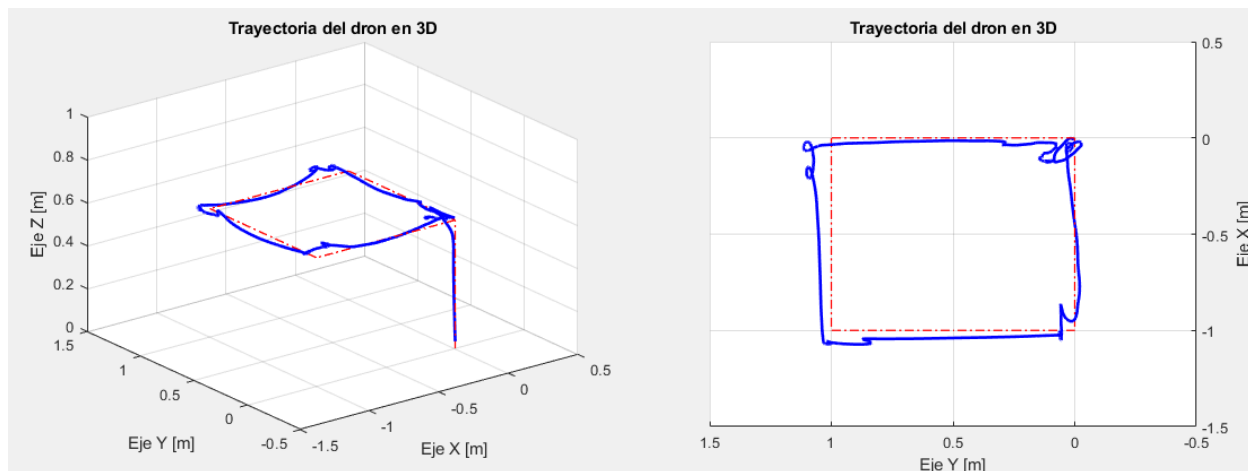


Figura 71. Datos posición de vuelo de la trayectoria cuadrada experimental LQG

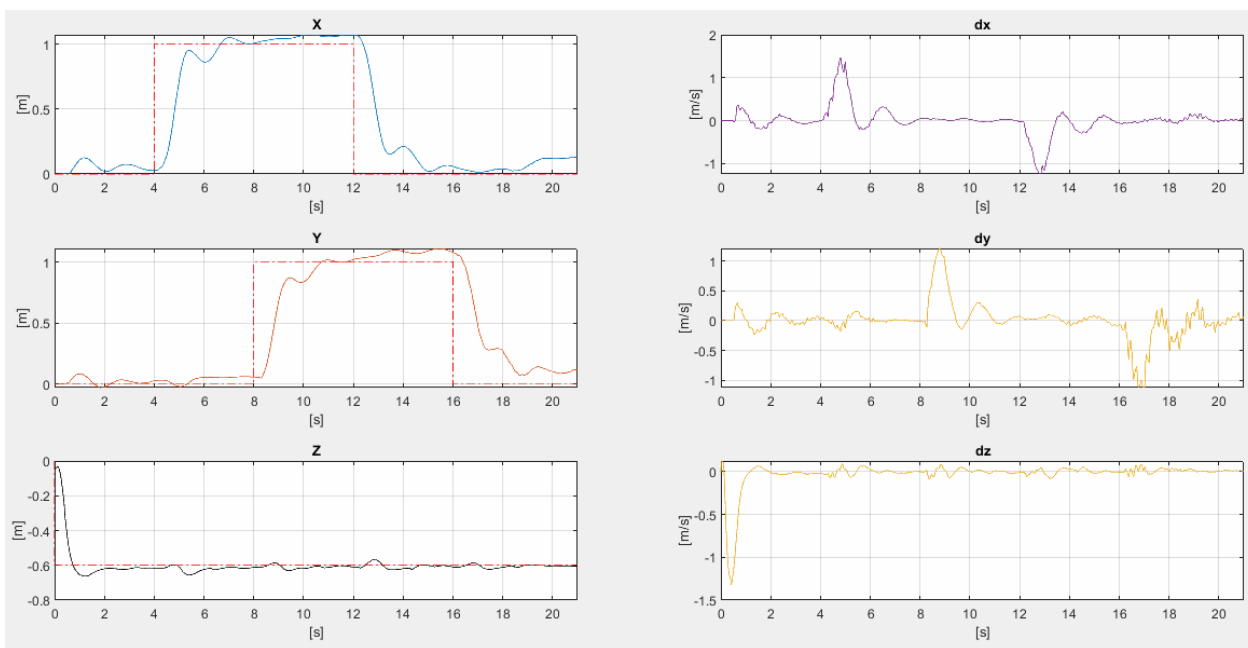
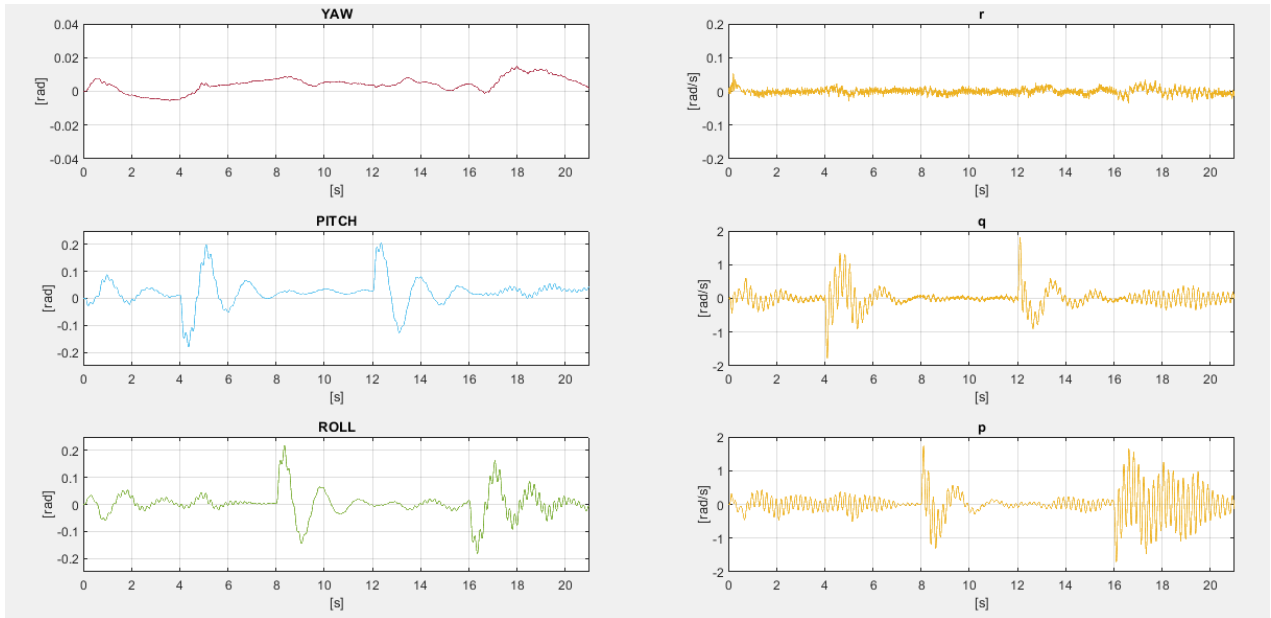


Figura 72. Datos orientación de vuelo de la trayectoria cuadrada experimental LQG



Al ver la Figura 70 se observa como el controlador LQG logra mantener la trayectoria de manera bastante precisa. En los cambios bruscos de dirección, es decir, en las esquinas se ven ángulos rectos bastante limpios sin perder demasiado la referencia. Esta información es corroborada con la Figura 71 en donde las gráficas de las coordenadas X, Y muestran un buen comportamiento. No presenta sobre picos pronunciados y la señal se estabiliza a los ~2.5 segundos.

Trayectoria cuadrada simulada con PID:

Figura 73. Vista isométrica y superior de la trayectoria cuadrada simulada PID

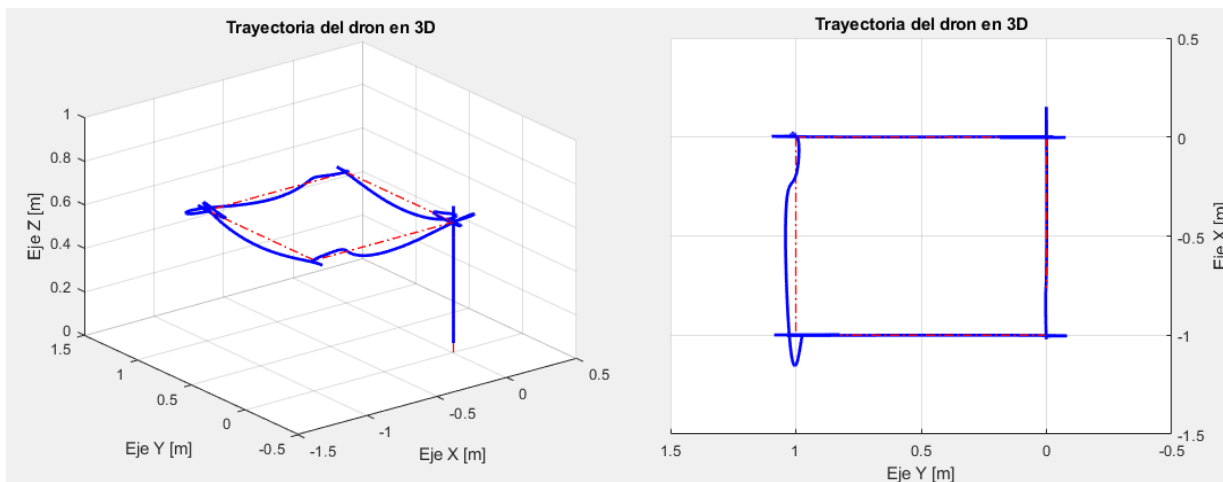


Figura 74. Datos posición de vuelo de la trayectoria cuadrada simulada PID

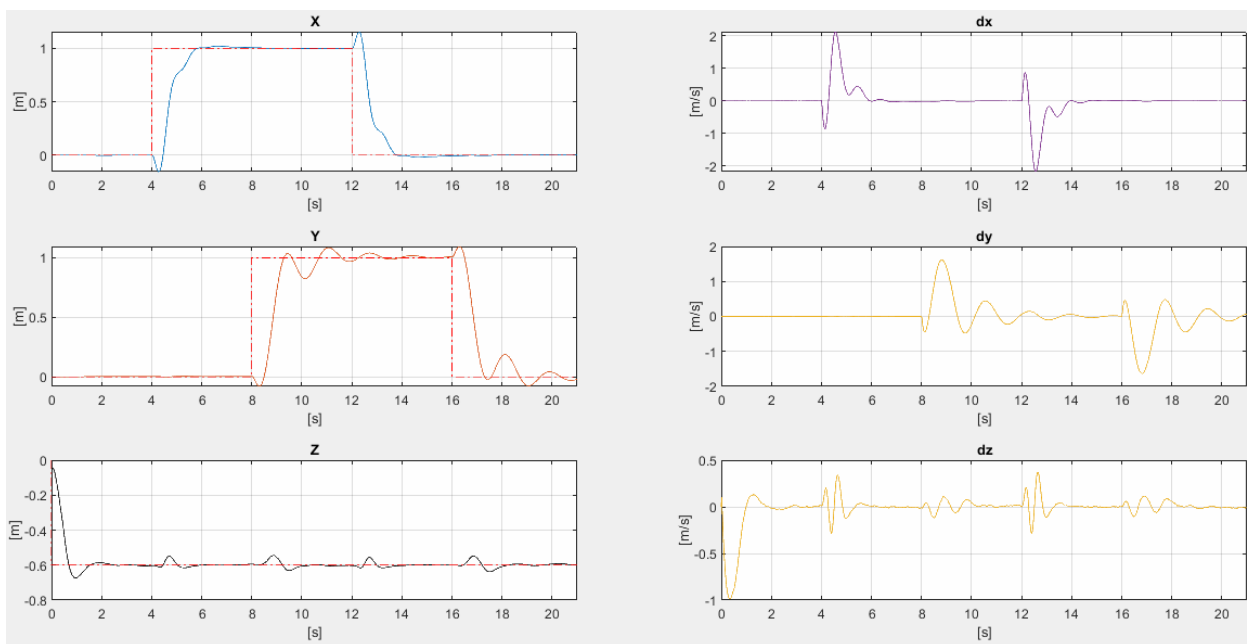
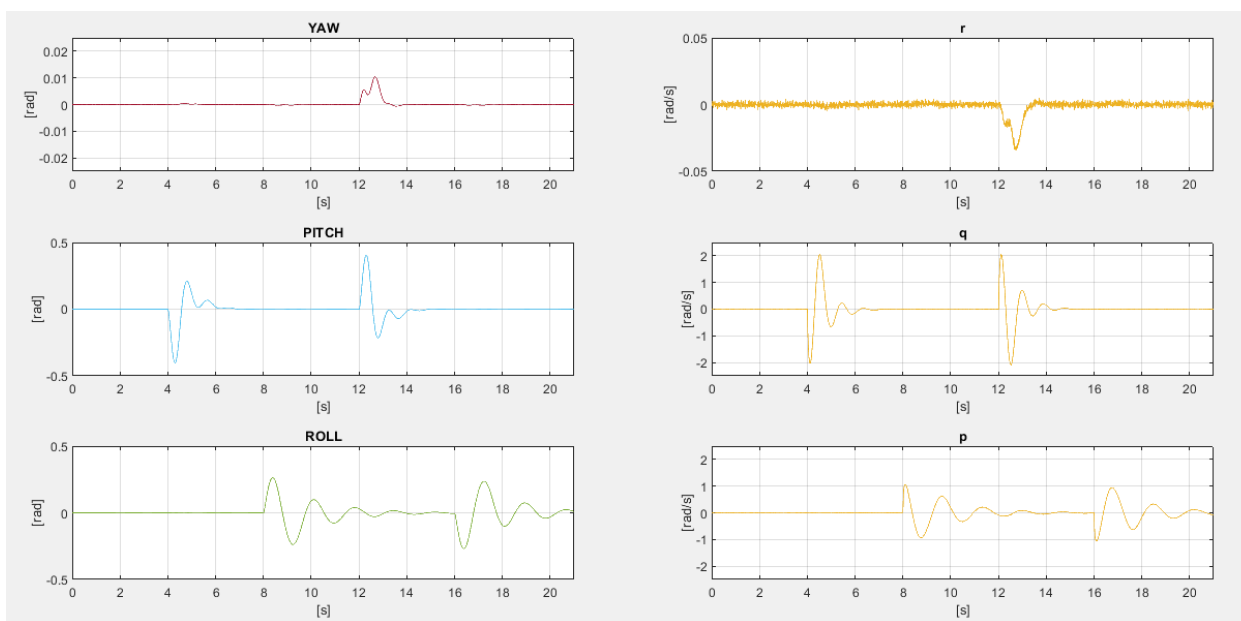


Figura 75. Datos orientación de vuelo de la trayectoria cuadrada simulada PID



Trayectoria cuadrada experimental con PID:

Figura 76. Vista isométrica y superior de la trayectoria cuadrada experimental PID

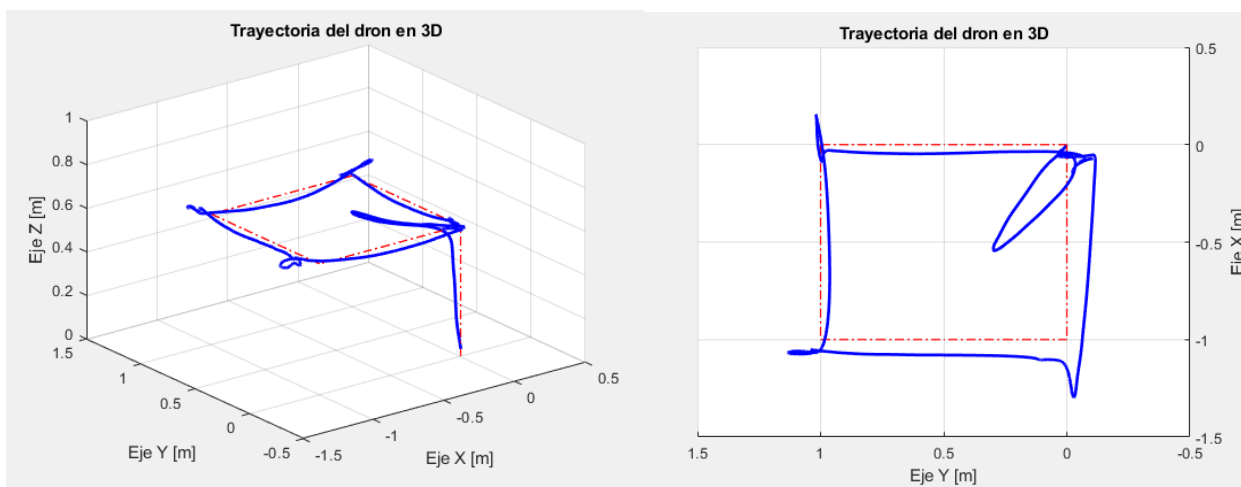


Figura 77. Datos posición de vuelo de la trayectoria cuadrada experimental PID

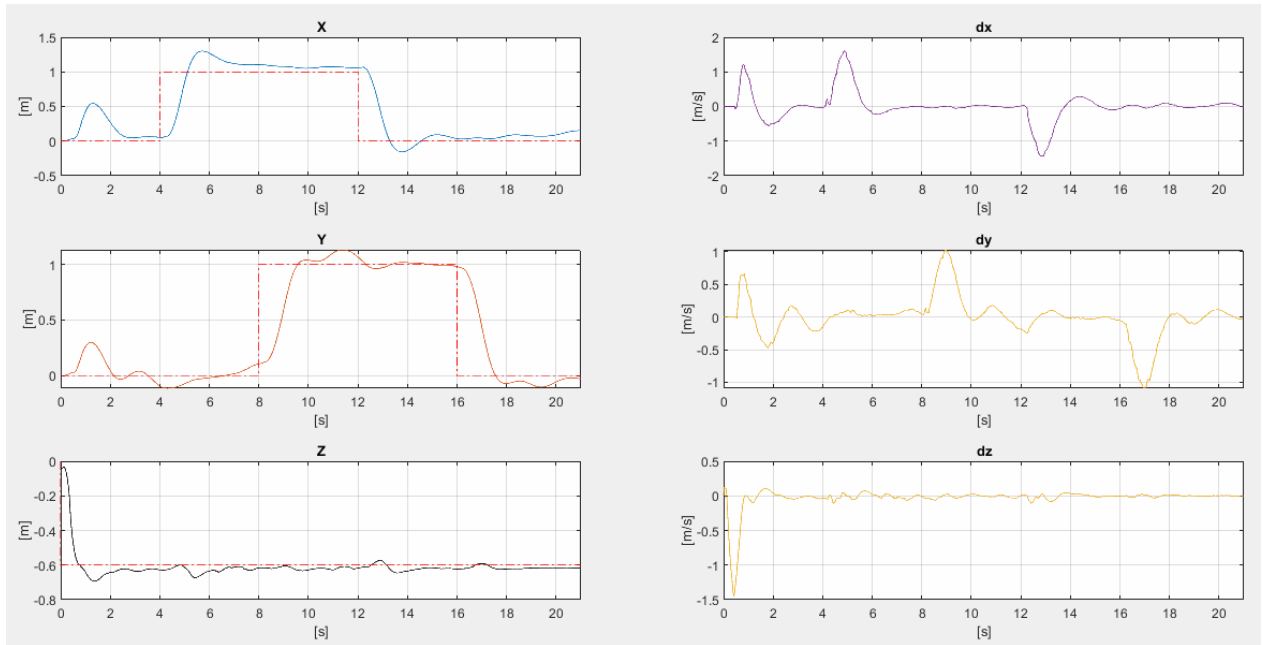
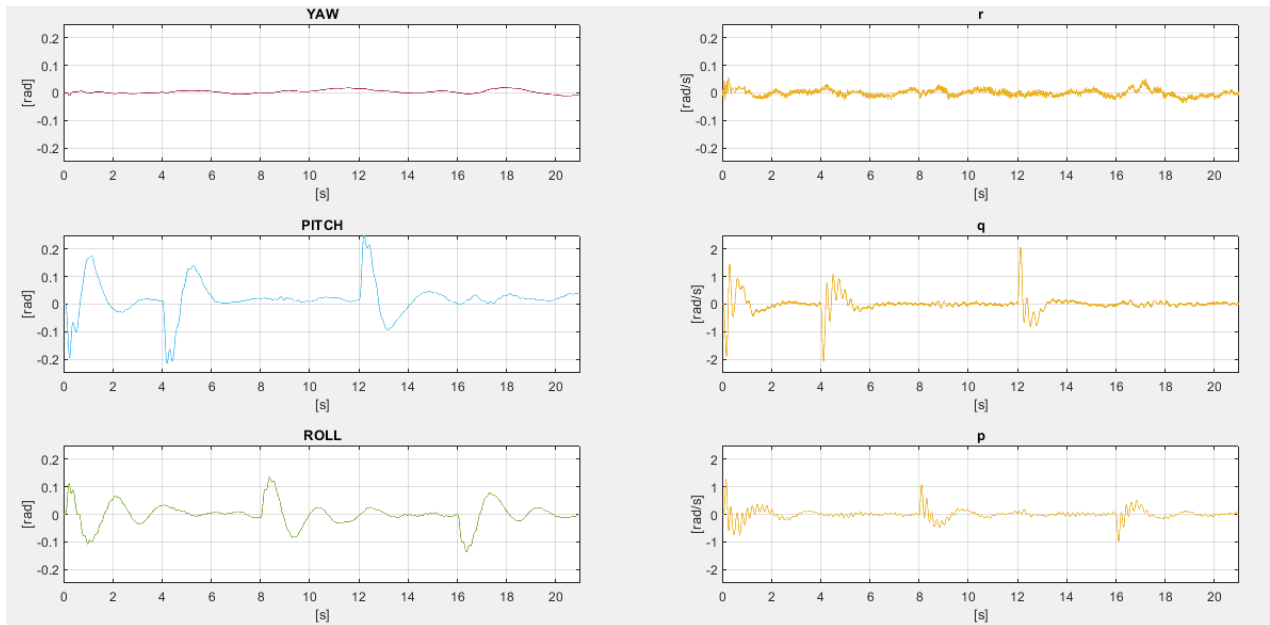


Figura 78. Datos orientación de vuelo de la trayectoria cuadrada experimental PID



Al ver la Figura 76 se observa como el controlador PID mantiene la trayectoria del dron con menor precisión que la del LQG. En los cambios bruscos de dirección, es decir, en

las esquinas se ve un comportamiento alargado de la trayectoria con una corrección deficiente de los sobre picos de señal. Sin embargo, se puede ver una trayectoria más suavizada. Esta información es corroborada con la Figura 77 en donde las gráficas de las coordenadas X, Y muestran picos más pronunciados.

Cuadro 12. PID Real vs PID Simulado en trayectoria cuadrada

PID VS PID SIMULADO									
Trayectoria	Control	ISE		ITSE		IAE		ITAE	
		X	Y	X	Y	X	Y	X	Y
Cuadrada	PID	1.796	1.454	31.650	30.960	3.710	2.770	14.277	17.739
	PID Simulado	1.475	1.652	15.765	28.793	1.832	2.234	12.317	20.620

Cuando se contrastan los indicadores de rendimiento entre el controlador PID real y el controlador PID Simulado en una trayectoria cuadrada (Cuadro 12), se nota que en lo que respecta al ISE y al ITAE, el controlador PID es mejor en la coordenada Y, mientras que, en todos los demás índices, el controlador PID Simulado muestra un mejor desempeño.

Cuadro 13. LQG Real vs LQG Simulado en trayectoria cuadrada

LQG VS LQG SIMULADO									
Trayectoria	Control	ISE		ITSE		IAE		ITAE	
		X	Y	X	Y	X	Y	X	Y
Cuadrada	LQG	1.564	1.674	29.252	41.593	2.930	3.153	14.237	22.247
	LQG Simulado	1.658	1.648	16.649	25.392	1.941	2.017	13.939	20.487

Cuando se evalúan los índices de desempeño entre el sistema de control LQG real y el sistema de control LQG Simulado en una trayectoria cuadrada (Cuadro 13), se aprecia que en términos de ISE, el LQG real supera al simulado en la coordenada X. Sin embargo, en todos los demás índices, el sistema LQG Simulado exhibe un mejor rendimiento.

Cuadro 14. PID Simulado vs LQG Simulado en trayectoria cuadrada

PID SIMULADO VS LQG SIMULADO									
Trayectoria	Control	ISE		ITSE		IAE		ITAE	
		X	Y	X	Y	X	Y	X	Y
Cuadrada	PID Simulado	1.475	1.652	15.765	28.793	1.832	2.234	12.317	20.620
	LQG Simulado	1.658	1.648	16.649	25.392	1.941	2.017	13.939	20.487

Al analizar los índices de rendimiento entre el PID Simulado y el LQG Simulado (Cuadro 14), se puede observar que el PID muestra un mejor desempeño en la coordenada X, mientras que el LQG muestra consistentemente un mejor desempeño en la coordenada Y.

Cuadro 15. PID Real vs LQG Real en trayectoria cuadrada

PID VS LQG									
Trayectoria	Control	ISE		ITSE		IAE		ITAE	
		X	Y	X	Y	X	Y	X	Y
Cuadrada	PID	1.796	1.454	31.650	30.960	3.710	2.770	14.277	17.739
	LQG	1.564	1.674	29.252	41.593	2.930	3.153	14.237	22.247

Al comparar los índices de desempeño entre el PID y el LQG (Cuadro 15), se ve que el PID invariablemente exhibe un mejor rendimiento en la coordenada Y, mientras que el LQG constantemente demuestra un mejor desempeño en la coordenada X.

Trayectoria circular simulada con LQG:

Figura 79. Vista isométrica y superior de la trayectoria circular simulada LQG

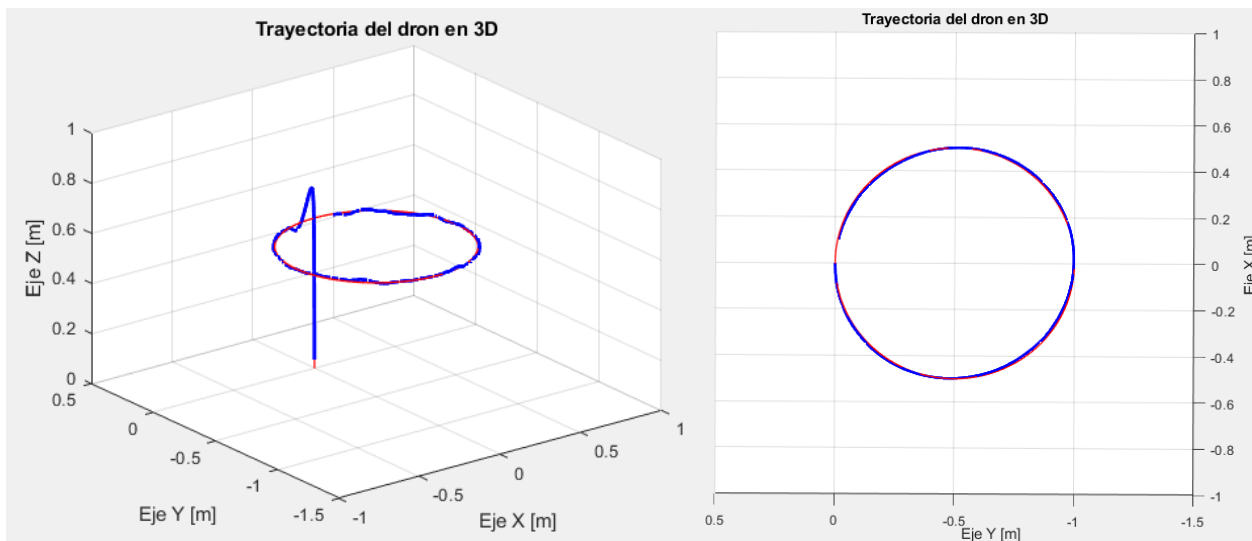


Figura 80. Datos posición de vuelo de la trayectoria circular simulada LQG

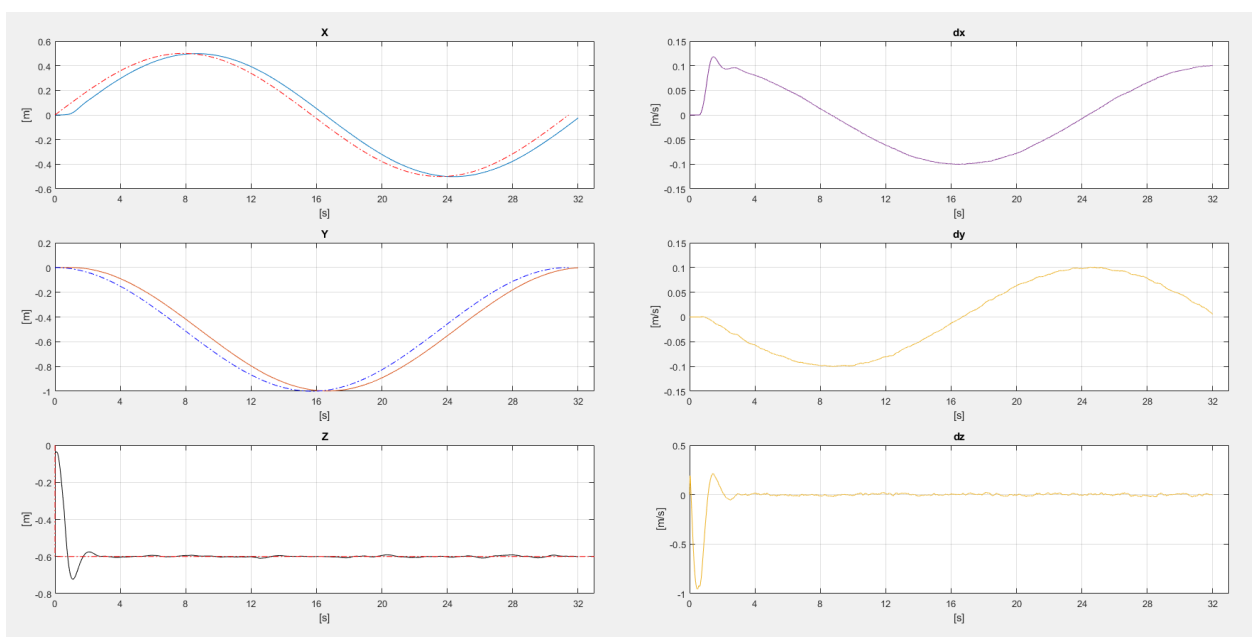
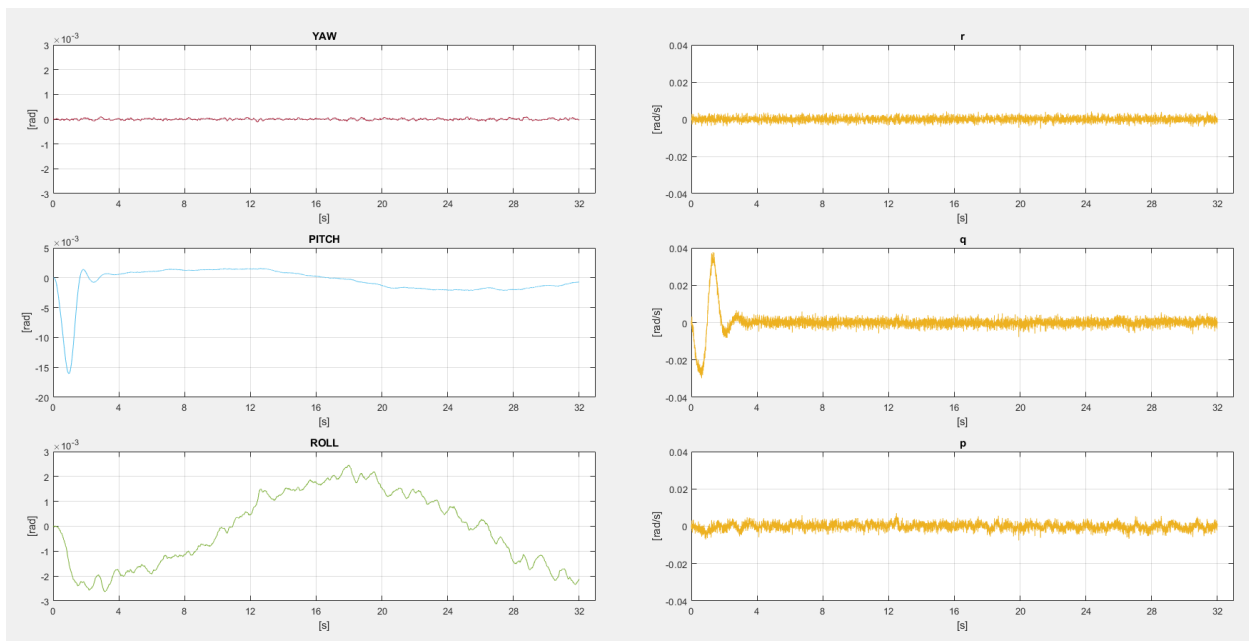


Figura 81. Datos orientación de vuelo de la trayectoria circular simulada LQG



Trayectoria circular experimental con LQG:

Figura 82. Vista isométrica y superior de la trayectoria circular experimental LQG

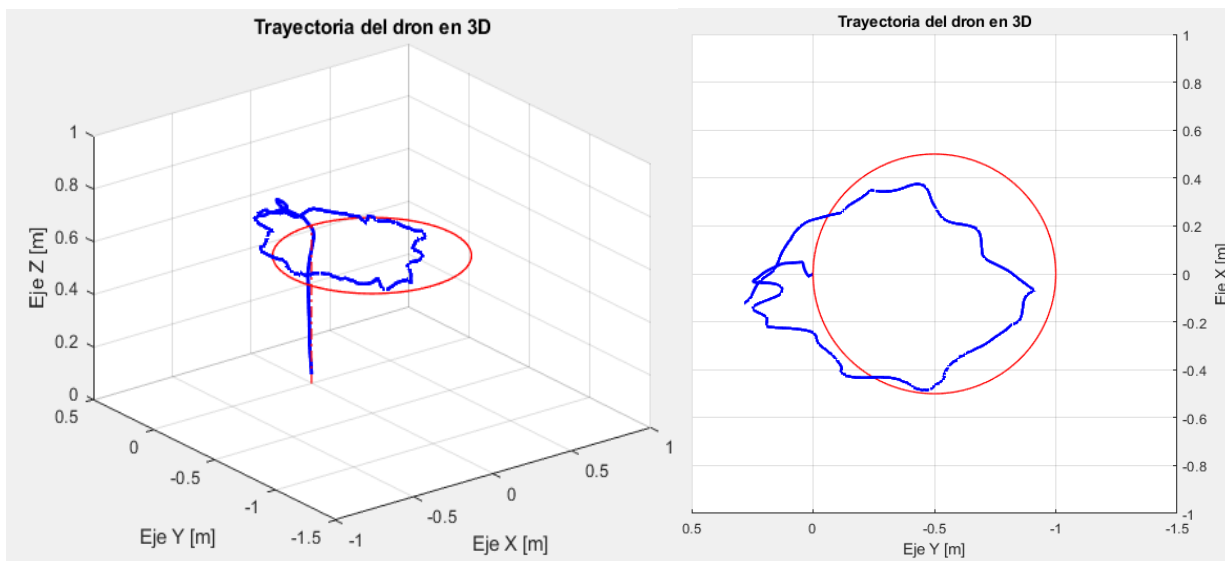


Figura 83. Datos posición de vuelo de la trayectoria circular experimental LQG

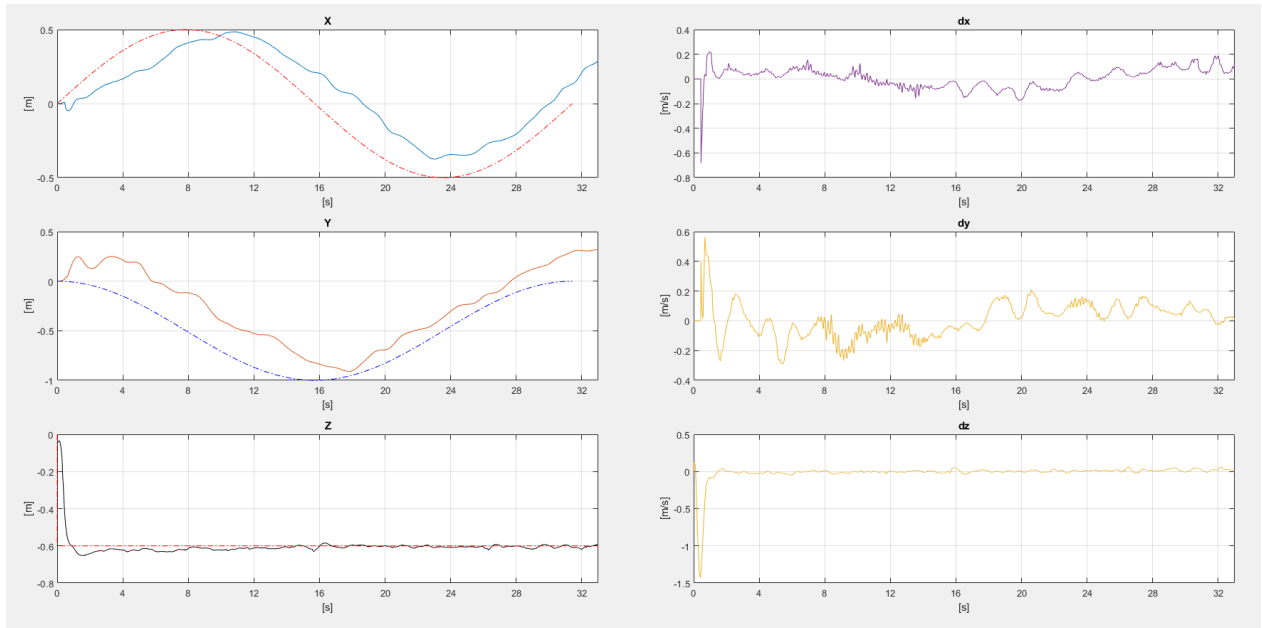
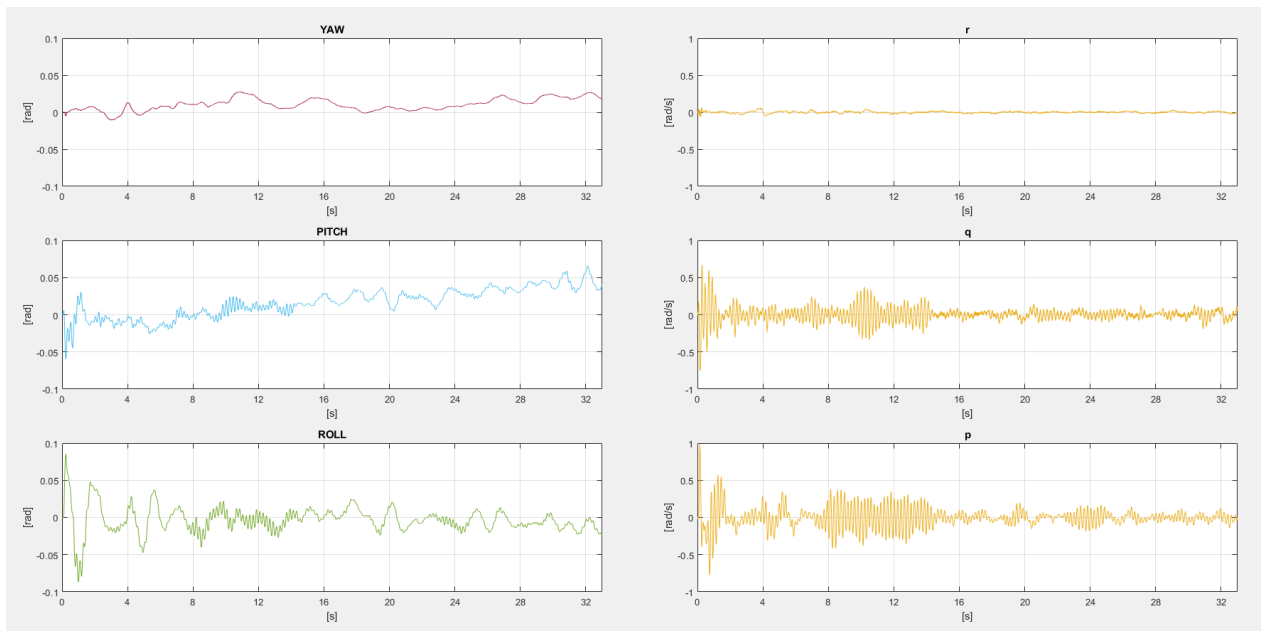


Figura 84. Datos orientación de vuelo de la trayectoria circular experimental LQG



En la trayectoria real del controlador LQG (Figura 82) se presenta un seguimiento de trayectoria muy poco preciso y bastante irregular. Este comportamiento aparece en

consecuencia de un error estacionario que se ve presente en las coordenadas X y Y de la Figura 83. Algo a resaltar es que en la gráfica de “Pitch” el ángulo no se estabiliza en cero por lo que no corrige completamente su trayectoria.

Trayectoria circular simulada con PID:

Figura 85. Vista isométrica y superior de la trayectoria circular simulada PID

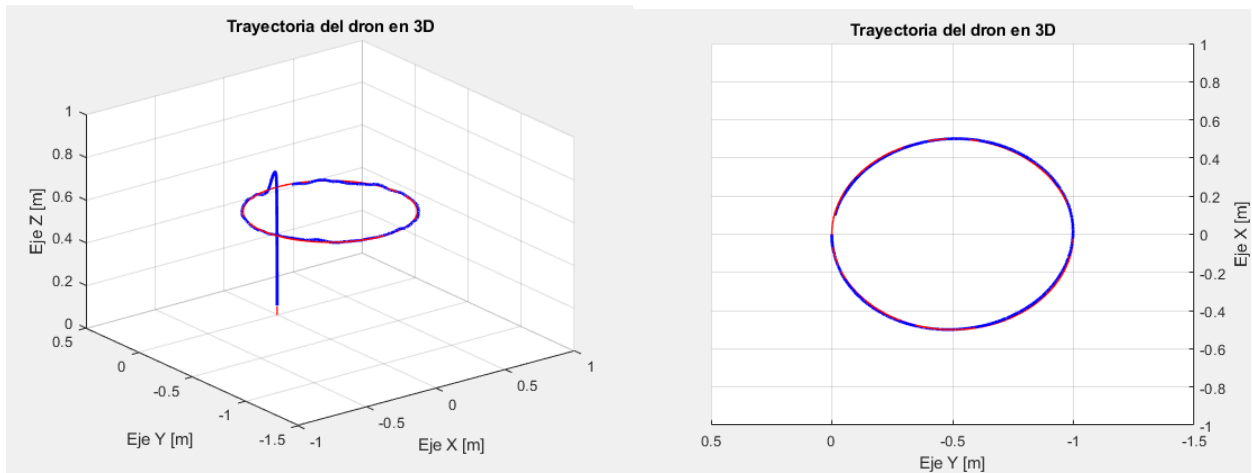


Figura 86. Datos posición de vuelo de la trayectoria circular simulada PID

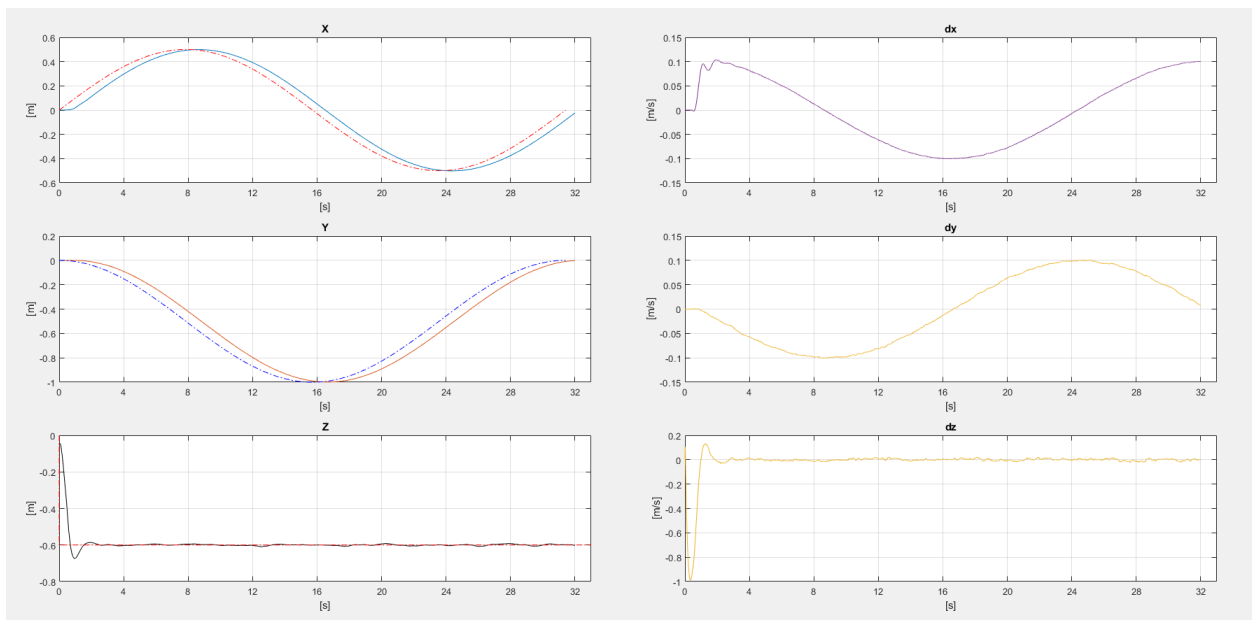
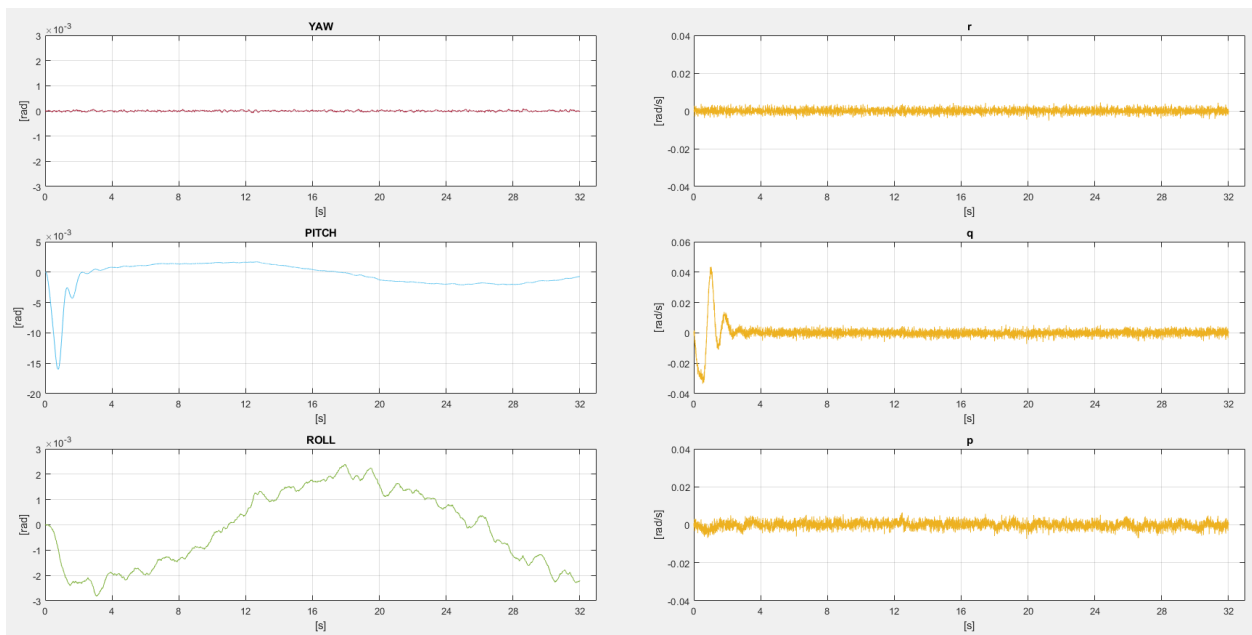


Figura 87. Datos orientación de vuelo de la trayectoria circular simulada PID



Trayectoria circular experimental con PID:

Figura 88. Vista isométrica y superior de la trayectoria circular experimental PID

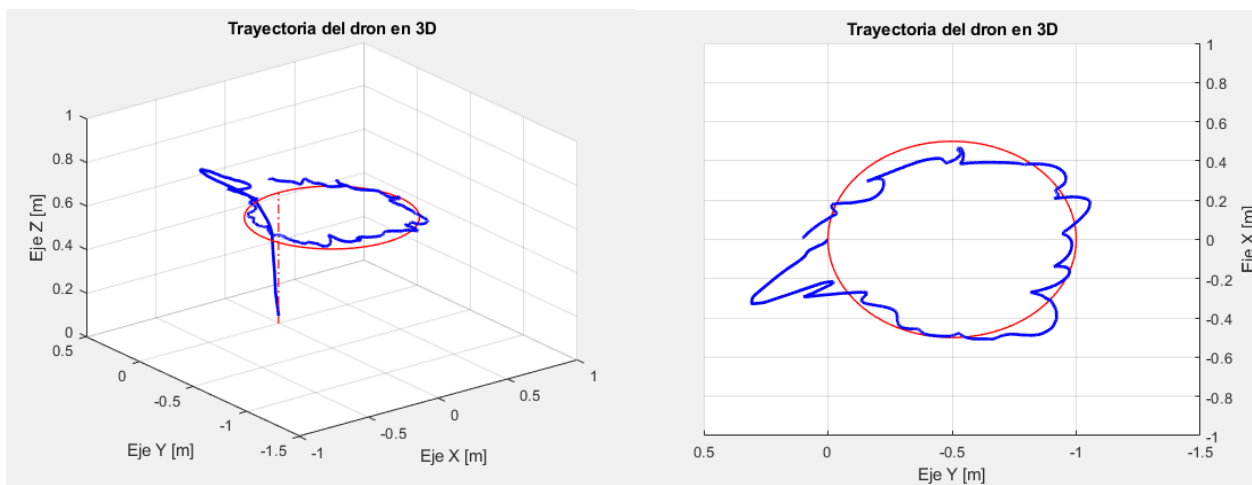


Figura 89. Datos posición de vuelo de la trayectoria circular experimental PID

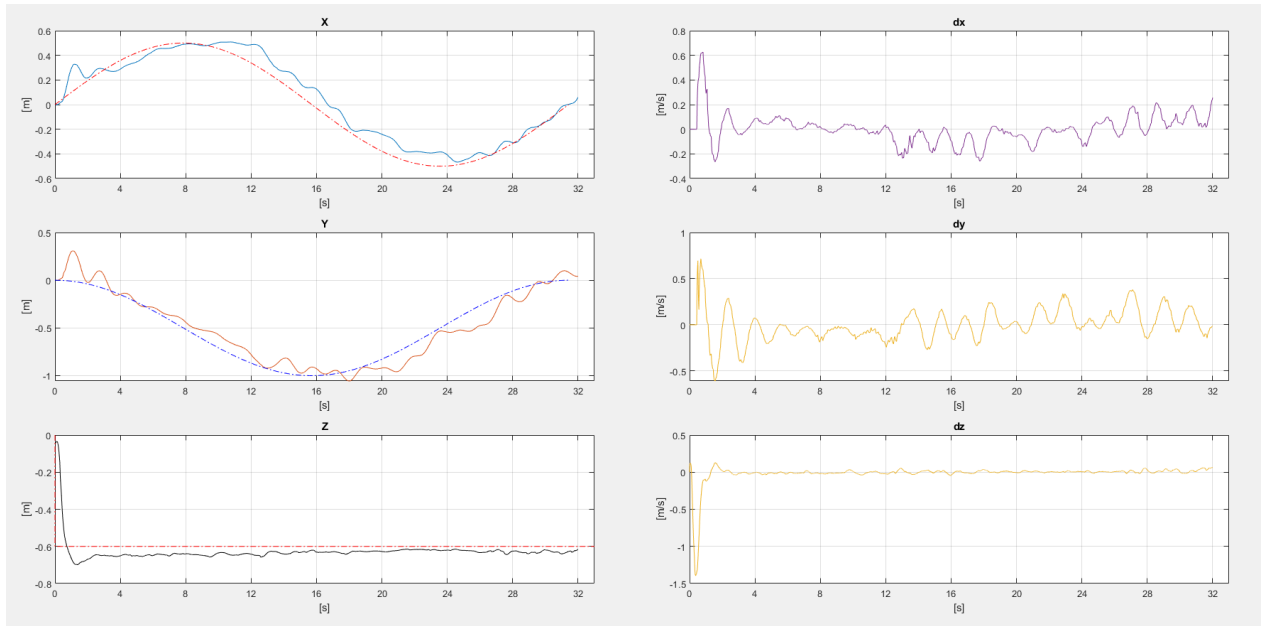
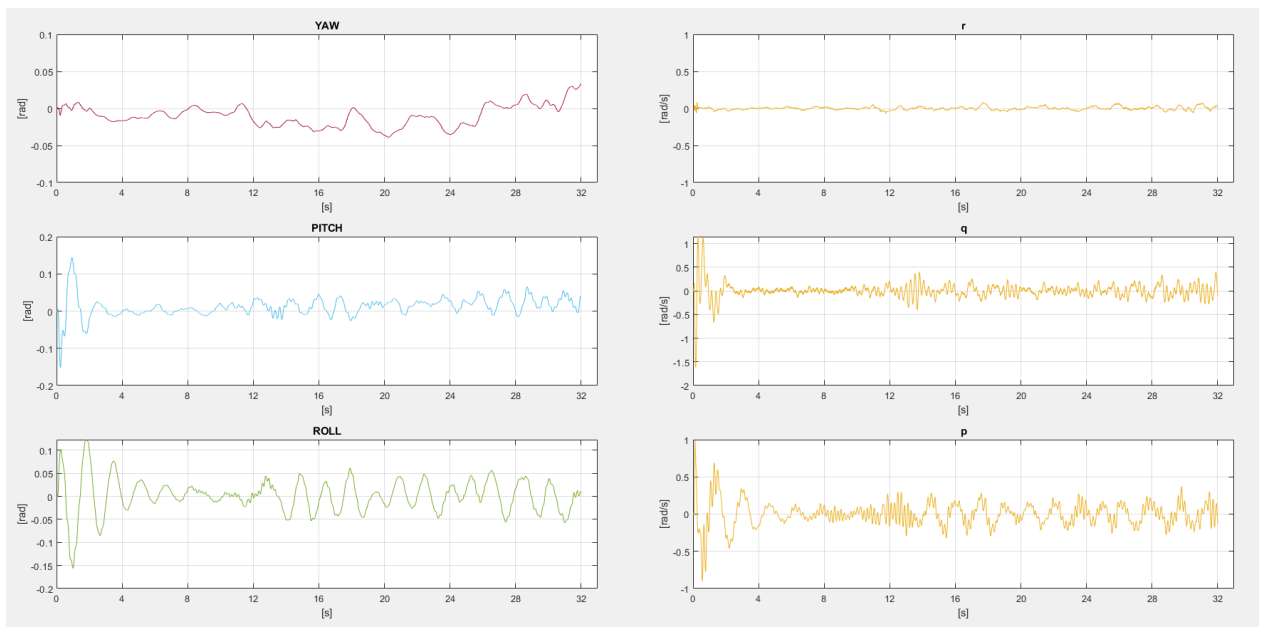


Figura 90. Datos orientación de vuelo de la trayectoria circular experimental PID



En las figuras de esta trayectoria circular experimental con PID, se observa que el dron sigue la señal de referencia de manera fluctuante a lo largo de la onda sinusoidal, sin embargo, sigue con buena precisión la señal de referencia. También es notable que, en

los primeros segundos de la trayectoria, el dron experimenta una desviación, la cual se corrige de manera ágil en un corto período de tiempo.

Cuadro 16. PID Real vs PID Simulado en trayectoria circular

PID VS PID SIMULADO									
Trayectoria	Control	ISE		ITSE		IAE		ITAE	
		X	Y	X	Y	X	Y	X	Y
Circular	PID	0.226	0.392	32.333	47.329	2.166	2.844	3.239	6.427
	PID Simulado	0.104	0.142	26.490	30.688	1.634	1.901	1.693	2.284

Cuadro 17. LQG Real vs LQG Simulado en trayectoria circular

LQG VS LQG SIMULADO									
Trayectoria	Control	ISE		ITSE		IAE		ITAE	
		X	Y	X	Y	X	Y	X	Y
Circular	LQG	0.860	2.008	80.966	100.830	4.877	7.245	14.332	24.227
	LQG Simulado	0.107	0.146	26.838	31.263	1.655	1.932	1.739	2.364

En los Cuadro 16 y Cuadro 17 las simulaciones demostraron mejor comportamiento en todos los índices de desempeño que su contraparte real. Algo a resaltar es que el índice ITSE es en el que mayor se aprecia esta diferencia. Esto se debe a que esta toma en cuenta el error acumulado en el tiempo.

Cuadro 18. PID Simulado vs LQG Simulado en trayectoria circular

PID SIMULADO VS LQG SIMULADO									
Trayectoria	Control								

Cuadro 19. PID Real vs LQG Real en trayectoria circular

PID VS LQG									
Trayectoria	Control	ISE		ITSE		IAE		ITAE	
		X	Y	X	Y	X	Y	X	Y
Circular	PID	0.226	0.392	32.333	47.329	2.166	2.844	3.239	6.427
	LQG	0.860	2.008	80.966	100.830	4.877	7.245	14.332	24.227

Entre el PID Real y el LQG real (Cuadro 19) se ve una gran diferencia en índices como el ITSE y el ITAE, esto se debe a que ambos penalizan fuertemente el error acumulado en el tiempo. En este caso el PID muestra un mejor desempeño que el LQG real.

Comparando las pruebas LQG reales de la trayectoria cuadrada y circular (Figura 71 y Figura 83) se observa que esta segunda trayectoria presenta un error estacionario de posición. Este comportamiento observado en la circular y no en la cuadrada posiblemente sea resultado de factores externos como corrientes de aire u otras perturbaciones ambientales que pueden afectar la trayectoria del dron y causar un error de offset.

En las trayectorias de las pruebas cuadradas y circulares de PID reales (Figura 76 y Figura 88) se observan cómo en los primeros segundos de la trayectoria el dron presenta una desviación, pero luego se corrige rápidamente.

En general se observan en las figuras de trayectorias de vuelo cuadrada y circular del entorno simulado que el seguimiento del dron es muy preciso. Al entrar en detalle con las figuras de datos de vuelo de estas, se evidencia que los estados estimados de X, Y y Z siguen muy bien las señales de referencia de tipo escalón (trayectoria cuadrada) y sinusoidales (trayectoria circular). En las figuras de trayectorias de vuelo cuadrada y circular del entorno real se observan que el seguimiento del dron presenta pequeñas y moderadas desviaciones. Al entrar en detalle con las figuras de datos de vuelo de estas, se evidencia que los estados estimados de X, Y y Z se van corrigiendo y tratan de seguir las señales de referencia con desviaciones aceptables.

CONCLUSIONES

- Se exploró la posibilidad de aplicar un enfoque de estimación de parámetros utilizando el modelo dinámico desarrollado y el método empleado en Grey box de Matlab, pero los resultados no fueron concluyentes ni satisfactorios. Este resultado subraya la complejidad inherente de la tarea de estimación de parámetros en sistemas de vuelo, donde la incertidumbre desempeña un papel crucial. Para abordar con éxito este método, se hace evidente la necesidad de emplear métodos de estimación de parámetros más avanzados y modelos dinámicos más complejos. Estos modelos deben tener en cuenta la geometría y la naturaleza estocástica del ruido, que es una variable que influye significativamente en las mediciones y perturbaciones en el entorno real de vuelo. Para solucionar la estimación de los parámetros del modelo dinámico de manera eficaz, se optó por hallarlos por un método mixto (experimental y Grey Box).
- La combinación de conocimiento teórico y datos experimentales con Grey box, demostró ser una herramienta capaz de estimar las características dinámicas del sistema como lo es las inercias del dron con una precisión del $\sim 98\%$. Con estas nuevas inercias se ajustaron los parámetros fundamentales y se mejoró la precisión del modelo dinámico del dron. La precisión de las inercias ofrecidas por el soporte Parrot muestra ciertos niveles de discrepancia en comparación con las inercias calculadas. Concretamente, se observaron desviaciones del 65% en el eje X, del 58% en el eje Y y del 61% en el eje Z. En cuanto a los coeficientes de empuje y de torque, su diferencia oscila entre 8% a 20 % y 64% a 71% respectivamente.
- Las pruebas simuladas no fueron semejantes a las reales. Los tiempos de establecimiento y precisión del seguimiento de la señal de referencia tuvieron notables diferencias. No obstante, la simulación desempeñó un papel fundamental al permitirnos poner a prueba y ajustar las ganancias de los controladores antes

de su implementación en la prueba real. Estas simulaciones no solo nos proporcionaron ahorros significativos en tiempo de sintonización, sino que también contribuyó a un proceso de desarrollo más eficiente y efectivo de las trayectorias.

- El controlador LQG demostró ser eficiente en cambios bruscos de trayectoria, mostrando una capacidad excepcional para mantener la estabilidad y la precisión. En la trayectoria circular demostró ser muy poco preciso ya que se ve desviado de la referencia. Por otro lado, el controlador PID sobresalió en trayectorias suavizadas como la circular, destacando su habilidad para lograr un seguimiento preciso y con algunas oscilaciones notables.
- El patrón de superficie tiene una alta influencia en el control del dron, ya que estados como la posición en X y Y dependen de la cámara y su procesador de imagen para estimar estos valores. Es un factor tan relevante, que sin una superficie adecuada y compatible ningún controlador funcionaría.

RECOMENDACIONES

- Para mejorar aún más el proceso de pruebas y validación del controlador del dron, sugerimos buscar patrones de vuelo compatibles con áreas más extensas. La elección de un espacio de pruebas más grande permitirá realizar evaluaciones más precisas y exhaustivas, ya que se podrán ejecutar trayectorias considerablemente más amplias. Esto proporcionará una representación más realista de las condiciones del mundo real y permitirá evaluar el rendimiento del dron en escenarios más desafiantes y variados. Además, un espacio de pruebas más grande brindará la oportunidad de simular situaciones de vuelo prolongadas, lo que es esencial para garantizar la estabilidad y eficiencia del controlador en misiones de mayor duración.
- Se recomienda explorar la implementación del enfoque "L1 Adaptive Augmentation for Geometric Tracking Control of Quadrotors". Este método avanzado de control adaptativo es diseñado específicamente para quadrotores, y ofrece notables ventajas en términos de precisión y capacidad de seguimiento de trayectorias de acuerdo con múltiples trabajos consultados. Al utilizar conceptos de control adaptativo L1, este enfoque permite que el dron ajuste dinámicamente sus parámetros de control en tiempo real, lo que es especialmente beneficioso para mantener trayectorias precisas en entornos variables. La incorporación de esta metodología puede llevar a mejoras significativas en la capacidad de respuesta del dron, su estabilidad y su resistencia a perturbaciones, lo que resulta fundamental para tareas que requieren un seguimiento de trayectorias de alta precisión, como la inspección de infraestructuras o la cartografía aérea.

BIBLIOGRAFÍA

Bouabdallah, Samir; Noth, A. y Siegwart, Roland. PID vs LQ Control Techniques Applied to an Indoor Micro Quadrotor. En: International Conference on Intelligent Robots and Systems. Lausanne. 2004. vol.3. No. 3. p. 2.

BRESCIANI, Tommaso. Modelling, Identification and Control of a Quadrotor Helicopter [en línea]. Máster en control automático. Lund: Lund University. 2008. p.13. [Consultado 08 junio 2023]. Disponible en: <https://lup.lub.lu.se/luur/download?func=downloadFile&recordId=8847641&fileId=8859343>

B&H. Parrot Mambo SIP6 Linux Motherboard con 800MHz ARM A9. [Sitio web]. [Consultado 09 junio 2023]. Disponible en: https://www.bhphotovideo.com/c/product/1274651-REG/parrot_pf070237_mambo_sip6_linux_motherboard.html

CEPPI, Paolo. MODEL-BASED DESIGN OF A LINE-TRACKING ALGORITHM FOR A LOW-COST MINI DRONE THROUGH VISION-BASED CONTROL [En línea]. Maestría en Ciencias en Ingeniería Eléctrica e Informática. Chicago: Universidad de Illinois. 2020. p.35. [Consultado 08 junio 2023]. Disponible en: <https://webthesis.biblio.polito.it/16018/1/tesi.pdf>

DORF, Richard. Modern Control Systems. 13 ed. Boston: Editorial Pearson, 2016. p. 316. ISBN 0134407628

DRONE MANYA. Revisión del quadcopter Parrot Mambo. [Sitio web]. [Consultado 09 junio 2023]. Disponible en: <https://dronemanya.com/es/post/revision-del-quadcopter-parrot-mambo#i-8>

GARCIA, Daniel. Aplicación de Matlab/Simulink al posicionamiento y control de drones en interiores [en línea]. Máster universitario en Ingeniería Industrial. Madrid: Universidad De Alcalá. Escuela Politécnica Superior. 2020. p.66 – 68. [Consultado 08 junio 2023]. Disponible en: <https://ebuah.uah.es/dspace/handle/10017/44709>

GONZÁLES, Miguel. Análisis completo de Parrot Mambo. En: Filmora [Sitio web]. [Consultado 09 junio 2023]. Disponible en: <https://filmora.wondershare.es/drones/parrot-mambo-complete-review.html>

Goldman Sachs. Drones: Reporting for Work. [Sitio web]. [Consultado 08 junio 2023]. Disponible en: <https://www.theverge.com/sponsored/goldman-sachs-drones/mobile>

INEED-MOTOR. ¿Qué es el motor sin núcleo? [Sitio web]. [Consultado 09 junio 2023]. Disponible en: <https://www.ineed-motor.com/news/what-is-coreless-motor-49234777.html>

MANUALPDF. Manual Parrot mambo. [En línea]. 2016. p.34. [Consultado 09 junio 2023]. Disponible en: <https://www.manualpdf.es/parrot/mambo-fly/manual?p=2>

MATHWORKS. Parrot Minidrones Support from Simulink. [Sitio web]. [Consultado 09 junio 2023]. Disponible en: <https://la.mathworks.com/hardware-support/parrot-minidrones.html>.

NAGABHUSHAN, B. L. y TOMLINSON, N. P. Dynamics and Control of a Heavy Lift Airship Hovering in a Turbulent Cross Wind. En: Journal of Aircraft [en línea]. Octubre, 1982. vol. 19, no. 10 [consultado 11 septiembre 2023], p. 826-830. ISSN 1533-3868. Disponible en: <https://doi.org/10.2514/3.61564>

NOORDIN, Aminurrashid; BASRI, Mohd; MOHAMED, Zaharuddin. Real-Time Implementation of an Adaptive PID Controller for the Quadrotor MAV Embedded Flight

Control System. Universidad Técnica Malaysia Melaka. Escuela de ingeniería eléctrica. Facultad de ingeniería eléctrica y electrónica. 2023. p.15 – 22. [Consultado 08 junio 2023]. Disponible en: <https://www.mdpi.com/2226-4310/10/1/59>

NOORDIN, Aminurrashid; BASRI, Mohd; MOHAMED, Zaharuddin. Simulation and experimental study on PID control of a quadrotor MAV with perturbation [en línea]. Grado en Ingeniería de Tecnología Industrial. Malaysia: Universidad Técnica Malaysia Melaka. Escuela de ingeniería eléctrica. Facultad de ingeniería eléctrica y electrónica. 2019. p.4 – 6. [Consultado 08 junio 2023]. Disponible en: <https://beei.org/index.php/EEl/article/view/2158>

OGATA, Katsuhiko. Ingeniería de control moderna. 5 ed. Madrid: Pearson education, 2010. p.798. ISBN 848322660X

OKASHA, Mohamed; KRALEV, Jordan; ISLAM, Maidul. Design and Experimental Comparison of PID, LQR, and MPC Stabilizing Controllers for Parrot Mambo Mini-Drone [en línea]. Emiratos Árabes Unidos: Universidad de los Emiratos Árabes Unidos. Departamento de ingeniería Aeroespacial y mecánica. 2022. p.11 – 19. [Consultado 08 junio 2023]. Disponible en: <https://www.mdpi.com/2226-4310/9/6/298>

OLIVER, Estefania. Te mostramos la historia de los drones con lujos y detalles. En: Digitaltrends [En línea]. 2018. [Consultado 08 junio 2023]. Disponible en: <https://es.digitaltrends.com/drones/la-historia-de-los-drones/>

PAIVA, Ernesto. MODELADO Y CONTROL DE UN CUADRICÓPTERO [en línea]. Máster en ingeniería mecánico-eléctrica con mención en automática y optimización. Piura: Universidad de Piura. 2016. p.7. [Consultado 08 junio 2023]. Disponible en: <https://pirhua.udep.edu.pe/handle/11042/2514>

RAFFO, Guilherme. Modelado y control de un helicóptero Quadrotor. [en línea]. Máster en automática, robótica y telemática. Sevilla: Universidad de Sevilla. 2007. p.15-16. [Consultado 08 junio 2023]. Disponible en: https://biblus.us.es/bibing/proyectos/abreproy/70017/descargar_fichero/Tesis_Master_GuilhermeRaffo.pdf

RO-BOTICA. Motor A para Parrot Mambo. [Sitio web]. [Consultado 09 junio 2023]. Disponible en: <https://www.ro-botica.com/producto-Print/Motor-a-para-Parrot-Mambo/>

RO-BOTICA. Motor C para Parrot Mambo. [Sitio web]. [Consultado 09 junio 2023]. Disponible en: <https://www.ro-botica.com/producto-Print/Motor-c-para-Parrot-Mambo/>

ROA, Marcos. DISEÑO DEL CONTROL DE NAVEGACIÓN PARA UN DRON PARROT MAMBO [en línea]. Grado en Ingeniería de Tecnología Industrial. Madrid: Universidad Pontificia Comillas. 2019. p.56 – 65. [Consultado 08 junio 2023]. Disponible en: <https://repositorio.comillas.edu/xmlui/handle/11531/32123>

TISCHLER, Mark B. y JEX, Henry R. Effects of atmospheric turbulence on a quadrotor heavy lift airship. En: Journal of Aircraft [en línea]. Diciembre, 1983. vol. 20, no. 12 [consultado 11 septiembre 2023], p. 1050-1057. ISSN 1533-3868. Disponible en: <https://doi.org/10.2514/3.48211>

ANEXOS

Anexo A. Linealización y cálculo de K LQR

```
syms x1 x2 x3 x4 x5 x6 x7 x8 x9 x10 x11 x12 U1 U2 U3 U4 Omega

Ix=0.000166944;
Iy=0.00017119;
Iz=0.00025418;

g=9.81;

m=0.072;
l=0.0624;

b=0.0000000399;
d=3.29*10^(-10);

%ESTADOS
eq4 = x5*x6*((Iy-Iz)/Ix)+l/Ix*(U2); %roll_dot
eq5 = x4*x6*((Iz-Ix)/Iy)+l/Iy*(U3); %pitch_dot
eq6 = x4*x5*((Ix-Iy)/Iz)+l/Iz*(U4); %yaw_dot
eq10 = g+(cos(x1)*cos(x2))*(1/m)*(-U1); %z_dot
eq11 = (cos(x1)*sin(x2)*cos(x3)+sin(x1)*sin(x3))*(1/m)*(-U1); %x_dot
eq12 = (cos(x1)*sin(x2)*sin(x3)-sin(x1)*cos(x3))*(1/m)*(-U1); %y_dot
eq1 = x4; %roll
eq2 = x5; %pitch
eq3 = x6; %yaw
eq7 = x10; %z
eq8 = x11; %x
eq9 = x12; %y

%PARA HALLAR MATRICES A Y B
AA = jacobian([eq1;eq2;eq3;eq4;eq5;eq6;eq7;eq8;eq9;eq10;eq11;eq12],[x1; x2; x3; x4;
x5; x6; x7; x8; x9; x10; x11; x12]);

BB = jacobian([eq1;eq2;eq3;eq4;eq5;eq6;eq7;eq8;eq9;eq10;eq11;eq12],[U1; U2; U3; U4]);

A_evaluada = subs(AA, [U1 U2 U3 U4 x1 x2 x3 x4 x5 x6], [0.7063 0 0 0 0 0 0 0 0 0]);
B_evaluada = subs(BB, [U1 U2 U3 U4 x1 x2 x3 x4 x5 x6], [0.7063 0 0 0 0 0 0 0 0 0]);

A= double(A_evaluada);
B= double(B_evaluada);

M = [B, A*B, (A^2)*B,(A^3)*B,(A^4)*B,(A^5)*B];
RM=rank(M);

% K DE LQR
q1 = 3.39;    q2 = 3.35;
```

```

q4 = 0.01;    q5 = 0.01;
q8 = 0.0001;  q9 = 0.0001;
q11 = 0.0001; q12 = 0.0001;

Q = diag([q1 q2 1 q4 q5 1 1 q8 q9 1 q11 q12]);

% q1(phi) - q2(theta) -q3(yaw)- q4(phidot) - q5(thetadot)- q6(yawdot) - q7(Z) - q8(X)
% - q9(Y)- q10(Zdot) - q11(Xdot) - q12(Ydot)

R = diag(200*[1 1 1 1]);
Ts1 = 0.02;
KR = lqrd(A,B,Q,R,Ts1);

KR = 0.1*[KR(3,2) KR(3,1) KR(3,5) KR(3,4)
          KR(2,2) KR(2,1) KR(2,5) KR(2,4)]

```

Anexo B. Grey Box estimación de parámetros.

```
close all
clear all
clc

load('C:\Users\POECSMART\Desktop\Id\Ix\y4.mat');
y=y4(2500:3001,1)*(pi()/180);
u = zeros(502,1);

%2
z = iddata(y, u, 0.01, 'Name', 'Dron');

%3
z.InputName = {"tao"};
z.InputUnit = {'N*m'};
z.OutputName = {'angle'};
z.OutputUnit = {'rad'};
z.Tstart = 0;
z.TimeUnit = 's';

%4
figure('Name', [z.Name ': tao -> angle output']);
plot(z(:, 1, 1));

%6
FileName      = 'dronCP_mI';           % File describing the model structure.
Order         = [1 1 2];               % Model orders [ny nu nx].
Parameters    = [0.001 0.0001]; % Initial parameters. I,
InitialStates = [0; 0];                % initial states.
Ts            = 0;                     % Time-continuous system.
nlgr = idnlgrey(FileName, Order, Parameters, InitialStates, Ts, 'Name', 'Dron');

%7
set(nlgr, 'InputName', {'tao'}, 'InputUnit', {'N*m'}, ...
        'OutputName', {'angle'}, ...
        'OutputUnit', {'rad'}, ...
        'TimeUnit', 's');

%8
nlgr = setinit(nlgr, 'Name', {'angle' 'angle_dot'});
nlgr = setinit(nlgr, 'Unit', {'rad' 'rad/s'});
nlgr = setpar(nlgr, 'Name', {'I' 'b'});
nlgr = setpar(nlgr, 'Unit', {'kg*m^2' 'kg*m^2/s*rad'});

%9
nlgr.SimulationOptions.AbsTol = 1e-10;
nlgr.SimulationOptions.RelTol = 1e-9;
compare(z, nlgr);

%10
nlgr = setinit(nlgr, 'Fixed', {false false}); % Estimate the initial states.
opt = nlgreyestOptions('Display', 'on');
```

```

opt.SearchOptions.MaxIterations = 100;
nlgr = nlgreyest(z, nlgr, opt);

nlgr.Report
fprintf('\n\nThe search termination condition:\n')
nlgr.Report.Termination

compare(z, nlgr);

present(nlgr);

%EN OTRA SCRIP PREVIAMENTE GUARDADO:

function [dx, y] = dronCP_mI(t, x, u, I,b, varargin)

y = [x(1)];

dx =[x(2);
      (u(1)-b*x(2)-0.148*9.81*0.2825*sin(x(1)))/(0.148*((0.2825)^2)+I)
    ];
end

```

Anexo C. Gráficas y cálculo de índices de desempeño.

```
clear
clc
close all

load('C:\Users\POECSMART\Desktop\Nueva carpeta (2)\RSdata56_PID_S.mat');

load('C:\Users\POECSMART\Desktop\xy.mat');

%motor

x_motor = rt_motor.time(:,1);
y_motor_1 = rt_motor.signals.values(:,1);
y_motor_2 = rt_motor.signals.values(:,2);
y_motor_3 = rt_motor.signals.values(:,3);
y_motor_4 = rt_motor.signals.values(:,4);

figure(1)
tiledlayout(2,2)

nexttile
plot(x_motor,y_motor_1,'Color',[0 0.4470 0.7410])
title('motor_1_U1')
grid

nexttile
plot(x_motor,y_motor_2,'Color',[0.8500 0.3250 0.0980])
title('motor_2_U4')
grid

nexttile
plot(x_motor,y_motor_3,'Color',[0 0 0])
title('motor_3_U3')
grid

nexttile
plot(x_motor,y_motor_4,'Color',[0.6350 0.0780 0.1840])
title('motor_4_U2')
grid

% Estim

x_estim = rt_estim.time(:,1);
y_estim_1 = rt_estim.signals.values(:,1);
y_estim_2 = rt_estim.signals.values(:,2);
y_estim_3 = rt_estim.signals.values(:,3);
y_estim_4 = rt_estim.signals.values(:,4);
y_estim_5 = rt_estim.signals.values(:,5);
y_estim_6 = rt_estim.signals.values(:,6);
y_estim_7 = rt_estim.signals.values(:,7);
```

```

y_estim_8 = rt_estim.signals.values(:,8);
y_estim_9 = rt_estim.signals.values(:,9);
y_estim_10 = rt_estim.signals.values(:,10);
y_estim_11 = rt_estim.signals.values(:,11);
y_estim_12 = rt_estim.signals.values(:,12);

figure(2)
tiledlayout(4,3)

nexttile
xr = [0,4,4,12,12,21];
yr = [0,0,1,1,0,0];
plot(x_estim,y_estim_1,'Color',[0 0.4470 0.7410])
hold on
plot(xr,yr,'r-.')
xticks(0:2:22)
xlim([0 21])
title('X')
xlabel('[s]')
ylabel('[m]')
grid

nexttile
xry = [0,8,8,16,16,21];
yry = [0,0,1,1,0,0];
plot(x_estim,y_estim_2,'Color',[0.8500 0.3250 0.0980])
hold on
plot(xry,yry,'r-.')
xticks(0:2:22)
xlim([0 21])
title('Y')
xlabel('[s]')
ylabel('[m]')
grid

nexttile
xrz = [0,0,21];
yrz = [0,-0.6,-0.6];
plot(x_estim,y_estim_3,'Color',[0 0 0])
hold on
plot(xrz,yrz,'r-.')
xticks(0:2:22)
xlim([0 21])
title('Z')
xlabel('[s]')
ylabel('[m]')
grid

nexttile

plot(x_estim,y_estim_4,'Color',[0.6350 0.0780 0.1840])
hold on
title('YAW')

```

```

xticks(0:2:22)
xlim([0 21])
xlabel('s')
ylabel('rad')
grid

nexttile
plot(x_estim,y_estim_5,'Color',[0.3010 0.7450 0.9330])
title('PITCH')
xticks(0:2:22)
xlim([0 21])
xlabel('s')
ylabel('rad')
grid

nexttile
plot(x_estim,y_estim_6,'Color',[0.4660 0.6740 0.1880])
title('ROLL')
xticks(0:2:22)
xlim([0 21])
xlabel('s')
ylabel('rad')
grid

nexttile
plot(x_estim,y_estim_7,'Color',[0.4940 0.1840 0.5560])
title('dx')
xticks(0:2:22)
xlim([0 21])
xlabel('s')
ylabel('m/s')
grid

nexttile
plot(x_estim,y_estim_8,'Color',[0.9290 0.6940 0.1250])
title('dy')
xticks(0:2:22)
xlim([0 21])
xlabel('s')
ylabel('m/s')
grid

nexttile
plot(x_estim,y_estim_9,'Color',[0.9290 0.6940 0.1250])
title('dz')
xticks(0:2:22)
xlim([0 21])
xlabel('s')
ylabel('m/s')
grid

nexttile
plot(x_estim,y_estim_10,'Color',[0.9290 0.6940 0.1250])

```

```

title('p')
xticks(0:2:22)
xlim([0 21])
xlabel('s')
ylabel('rad/s')
grid

nexttile
plot(x_estim,y_estim_11,'Color',[0.9290 0.6940 0.1250])
title('q')
xticks(0:2:22)
xlim([0 21])
xlabel('s')
ylabel('rad/s')
grid

nexttile
plot(x_estim,y_estim_12,'Color',[0.9290 0.6940 0.1250])
title('r')
xticks(0:2:22)
xlim([0 21])
xlabel('s')
ylabel('rad/s')
grid

% Posref

L=length(rt_posref.time);
j=rt_posref.time(L,1);

x_posref = rt_posref.time(:,1);
% y_posref_1 = cos(x_posref); %REFERENCIA PARA CIRCULAR
% y_posref_2 = sin(x_posref); %REFERENCIA PARA CIRCULAR
%y_posref_1 = zeros(1,9); %REFERENCIA PARA HOVER 0.8m
%y_posref_2 = zeros(1,9); %REFERENCIA PARA HOVER 0.8m
%y_posref_3 =0:0.1:0.8; %REFERENCIA PARA HOVER 0.8m
y_posref_1 = rt_posref.signals.values(:,1);
y_posref_2 = rt_posref.signals.values(:,2);
y_posref_3 = rt_posref.signals.values(:,3);
y_posref_4 = rt_posref.signals.values(:,4);
y_posref_5 = rt_posref.signals.values(:,5);
y_posref_6 = rt_posref.signals.values(:,6);
y_posref_7 = rt_posref.signals.values(:,7); %PITCH
y_posref_8 = rt_posref.signals.values(:,8); %ROLL

figure(3)
tiledlayout(4,2)

nexttile
plot(x_posref,y_posref_1,'Color',[0 0.4470 0.7410])
title('posref_1')
grid

```

```

nexttile
plot(x_posref,y_posref_2, 'Color',[0.8500 0.3250 0.0980])
title('posref_2')
grid

nexttile
plot(x_posref,y_posref_3, 'Color',[0 0 0])
title('posref_3')
grid

nexttile
plot(x_posref,y_posref_4, 'Color',[0.6350 0.0780 0.1840])
title('posref_4')
grid

nexttile
plot(x_posref,y_posref_5, 'Color',[0.3010 0.7450 0.9330])
title('posref_5')
grid

nexttile
plot(x_posref,y_posref_6, 'Color',[0.4660 0.6740 0.1880])
title('posref_6')
grid

nexttile
plot(x_posref,y_posref_7, 'Color',[0.4940 0.1840 0.5560])
title('posref_7')
grid

nexttile
plot(x_posref,y_posref_8, 'Color',[0.9290 0.6940 0.1250])
title('posref_8')
grid

%calib

x_calib = rt_calib.time(:,1);
y_calib_1 = rt_calib.signals.values(:,1);
y_calib_2 = rt_calib.signals.values(:,2);
y_calib_3 = rt_calib.signals.values(:,3);
y_calib_4 = rt_calib.signals.values(:,4);
y_calib_5 = rt_calib.signals.values(:,5);
y_calib_6 = rt_calib.signals.values(:,6);
y_calib_7 = rt_calib.signals.values(:,7);
y_calib_8 = rt_calib.signals.values(:,8);

figure(4)
tiledlayout(4,2)

nexttile
plot(x_calib,y_calib_1, 'Color',[0 0.4470 0.7410])
title('calib_1')

```

```

grid

nexttile
plot(x_calib,y_calib_2, 'Color',[0.8500 0.3250 0.0980])
title('calib_2')
grid

nexttile
plot(x_calib,y_calib_3, 'Color',[0 0 0])
title('calib_3')
grid

nexttile
plot(x_calib,y_calib_4, 'Color',[0.6350 0.0780 0.1840])
title('calib_4')
grid

nexttile
plot(x_calib,y_calib_5, 'Color',[0.3010 0.7450 0.9330])
title('calib_5')
grid

nexttile
plot(x_calib,y_calib_6, 'Color',[0.4660 0.6740 0.1880])
title('calib_6')
grid

nexttile
plot(x_calib,y_calib_7, 'Color',[0.4940 0.1840 0.5560])
title('calib_7')
grid

nexttile
plot(x_calib,y_calib_8, 'Color',[0.9290 0.6940 0.1250])
title('calib_8')
grid

%cmd

x_cmd = rt_cmd.time(:,1);
y_cmd_1 = rt_cmd.signals.values(:,1);
y_cmd_2 = rt_cmd.signals.values(:,2);
y_cmd_3 = rt_cmd.signals.values(:,3);
y_cmd_4 = rt_cmd.signals.values(:,4);
y_cmd_5 = rt_cmd.signals.values(:,5);
y_cmd_6 = rt_cmd.signals.values(:,6);
y_cmd_7 = rt_cmd.signals.values(:,7);
y_cmd_8 = rt_cmd.signals.values(:,8);
y_cmd_9 = rt_cmd.signals.values(:,9);

figure(5)
tiledlayout(3,3)

```

```

nexttile
plot(x_cmd,y_cmd_1,'Color',[0 0.4470 0.7410])
title('controlModePosVSorient')
grid

nexttile
plot(x_cmd,y_cmd_2,'Color',[0.8500 0.3250 0.0980])
title('pos ref_1')
grid

nexttile
plot(x_cmd,y_cmd_3,'Color',[0 0 0])
title('pos ref_2')
grid

nexttile
plot(x_cmd,y_cmd_4,'Color',[0.6350 0.0780 0.1840])
title('pos ref_3')
grid

nexttile
plot(x_cmd,y_cmd_5,'Color',[0.3010 0.7450 0.9330])
title('takeoff flag')
grid

nexttile
plot(x_cmd,y_cmd_6,'Color',[0.4660 0.6740 0.1880])
title('orient ref_1')
grid

nexttile
plot(x_cmd,y_cmd_7,'Color',[0.4940 0.1840 0.5560])
title('orient ref_2')
grid

nexttile
plot(x_cmd,y_cmd_8,'Color',[0.9290 0.6940 0.1250])
title('orient ref_3')
grid

nexttile
plot(x_cmd,y_cmd_9,'Color',[0.4660 0.6740 0.1880])
title('live time ticks')
grid

%optical

x_optical = rt_optical.time(:,1);
y_optical_1 = rt_optical.signals.values(:,1);
y_optical_2 = rt_optical.signals.values(:,2);
y_optical_3 = rt_optical.signals.values(:,3);
y_optical_4 = rt_optical.signals.values(:,4);
y_optical_5 = rt_optical.signals.values(:,5);

```

```

y_optical_6 = rt_optical.signals.values(:,6);
y_optical_7 = rt_optical.signals.values(:,7);
y_optical_8 = rt_optical.signals.values(:,8);

figure(6)
tiledlayout(4,2)

nexttile
plot(x_optical,y_optical_1,'Color',[0 0.4470 0.7410])
title('usePosVIS flag')
grid

nexttile
plot(x_optical,y_optical_2,'Color',[0.8500 0.3250 0.0980])
title('opticalFlow data_1')
grid

nexttile
plot(x_optical,y_optical_3,'Color',[0 0 0])
title('opticalFlow data_2')
grid

nexttile
plot(x_optical,y_optical_4,'Color',[0.6350 0.0780 0.1840])
title('opticalFlow data_3')
grid

nexttile
plot(x_optical,y_optical_5,'Color',[0.3010 0.7450 0.9330])
title('posVIS data_1')
grid

nexttile
plot(x_optical,y_optical_6,'Color',[0.4660 0.6740 0.1880])
title('posVIS data_2')
grid

nexttile
plot(x_optical,y_optical_7,'Color',[0.4940 0.1840 0.5560])
title('posVIS data_3')
grid

nexttile
plot(x_optical,y_optical_8,'Color',[0.9290 0.6940 0.1250])
title('posVIS data_4')
grid

%sensor

x_sensor = rt_sensor.time(:,1);
y_sensor_1 = rt_sensor.signals.values(:,1);
y_sensor_2 = rt_sensor.signals.values(:,2);
y_sensor_3 = rt_sensor.signals.values(:,3);

```

```

y_sensor_4 = rt_sensor.signals.values(:,4);
y_sensor_5 = rt_sensor.signals.values(:,5);
y_sensor_6 = rt_sensor.signals.values(:,6);
y_sensor_7 = rt_sensor.signals.values(:,7);
y_sensor_8 = rt_sensor.signals.values(:,8);
y_sensor_9 = rt_sensor.signals.values(:,9);
y_sensor_10 = rt_sensor.signals.values(:,10);

figure(7);
tiledlayout(5,2)

nexttile
plot(x_sensor,y_sensor_1, 'Color',[0 0.4470 0.7410])
title('ddx')
grid

nexttile
plot(x_sensor,y_sensor_2, 'Color',[0.8500 0.3250 0.0980])
title('ddy')
grid

nexttile
plot(x_sensor,y_sensor_3, 'Color',[0 0 0])
title('ddz')
grid

nexttile
plot(x_sensor,y_sensor_4, 'Color',[0.6350 0.0780 0.1840])
title('p')
grid

nexttile
plot(x_sensor,y_sensor_5, 'Color',[0.3010 0.7450 0.9330])
title('q')
grid

nexttile
plot(x_sensor,y_sensor_6, 'Color',[0.4660 0.6740 0.1880])
title('r')
grid

nexttile
plot(x_sensor,y_sensor_7, 'Color',[0.4940 0.1840 0.5560])
title('altitude sonar')
grid

nexttile
plot(x_sensor,y_sensor_8, 'Color',[0.9290 0.6940 0.1250])
title('prs')
grid

nexttile
plot(x_sensor,y_sensor_9, 'Color',[0.4940 0.1840 0.5560])

```

```

title('vbat_V')
grid

nexttile
plot(x_sensor,y_sensor_10,'Color',[0.9290 0.6940 0.1250])
title('vbat_percentage')
grid

%yout

x_yout = rt_yout.time(:,1);
y_yout_1 = rt_yout.signals(1).values(:,1);
y_yout_2 = rt_yout.signals(1).values(:,2);
y_yout_3 = rt_yout.signals(1).values(:,3);
y_yout_4 = rt_yout.signals(1).values(:,4);

figure(8);
tiledlayout(4,1)

nexttile
plot(x_yout,y_yout_1,'Color',[0 0.4470 0.7410])
title('Actuators yout_1')
grid

nexttile
plot(x_yout,y_yout_2,'Color',[0.8500 0.3250 0.0980])
title('Actuators yout_2')
grid

nexttile
plot(x_yout,y_yout_3,'Color',[0 0 0])
title('Actuators yout_3')
grid

nexttile
plot(x_yout,y_yout_4,'Color',[0.6350 0.0780 0.1840])
title('Actuators yout_4')
grid

% Graficar la trayectoria en 3D
figure(9);

% PARA CIRCUNFERENCIA
% plot3(0.4*sin(x_posref), 0.4*cos(x_posref)-0.4 , 0.8*ones(1,L), 'r-', 'LineWidth',
2);
% axis([-1.5 1.5 -1.5 1.5 0 1.5]);
% hold on
plot3(0:-0.1:-1 , zeros(1,11) , 0.6*ones(1,11), 'r-.', 'LineWidth', 1);
axis([-1.5 1.5 -1.5 1.5 0 1.5]);
hold on
plot3(-1*ones(1,11) , 0:
```

```

hold on

plot3(0:-0.1:-1 , 1*ones(1,11) , 0.6*ones(1,11), 'r-.', 'LineWidth', 1);
axis([-1.5 1.5 -1.5 1.5 0 1.5]);
hold on
plot3(zeros(1,11) , 0:0.1:1 , 0.6*ones(1,11), 'r-.', 'LineWidth', 1);
axis([-1.5 1.5 -1.5 1.5 0 1.5]);
hold on

plot3(zeros(1,7) , zeros(1,7) , 0:0.1:0.6, 'r-.', 'LineWidth', 1);
axis([-1.5 1.5 -1.5 1.5 0 1.5]);
hold on
% plot3(-y_estim_1, y_estim_2, -y_estim_3, 'b-', 'LineWidth', 2);
% axis([-1.5 1.5 -1.5 1.5 0 1.5]);
% hold off

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

i=1;
while i<length(x_sensor)
    plot3(-y_estim_1(1:i), y_estim_2(1:i), -y_estim_3(1:i), 'b-', 'LineWidth', 2);
    hold on
    axis([-1.5 0.5 -0.5 1.5 0 1]);
    % xlim([-1.5 1.5]);
    % ylim([-1.5 1.5]);
    % zlim([0 1.5])
    i=i+10;
    pause(0.005)
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% PARA HOVER

% plot3(y_posref_1, y_posref_2, y_posref_3, 'r-', 'LineWidth', 2);
% axis([-1.5 1.5 -1.5 1.5 0 1.5]);
% hold on
% plot3(y_estim_1, y_estim_2, -y_estim_3, 'b-', '
```

```

y_posref_2=posrefbien.signals.values(1:4200,2);

y_estim_1 = rt_estim.signals.values(1:4200,1);
y_estim_2 = rt_estim.signals.values(1:4200,2);

t=0.005:0.005:21;
ti=t';

X_error = abs(y_posref_1 - y_estim_1);
Y_error = abs(y_posref_2 - y_estim_2);

ISE_X = trapz(t,X_error.^2);
ISE_Y = trapz(t,Y_error.^2);

ITSE_X = trapz(ti,ti.*X_error);
ITSE_Y = trapz(ti,ti.*Y_error);

IAE_X = trapz(t,X_error);
IAE_Y = trapz(t,Y_error);

ITAE_X = trapz(ti,ti.*X_error.^2);
ITAE_Y = trapz(ti,ti.*Y_error.^2);

disp(['El índice ISE_X es: ' num2str(ISE_X)]);
disp(['El índice ISE_Y es: ' num2str(ISE_Y)]);
disp(['El índice ITSE_X es: ' num2str(ITSE_X)]);
disp(['El índice ITSE_Y es: ' num2str(ITSE_Y)]);
disp(['El índice IAE_X es: ' num2str(IAE_X)]);
disp(['El índice IAE_Y es: ' num2str(IAE_Y)]);
disp(['El índice ITAE_X es: ' num2str(ITAE_X)]);
disp(['El índice ITAE_Y es: ' num2str(ITAE_Y)]);

```

Anexo D. Código balanza

%Para la calibración

```
#include "HX711.h"
const int DOUT=A1;
const int CLK=A0;

HX711 balanza;
void setup() {
  Serial.begin(9600);
  balanza.begin(DOUT, CLK);
  Serial.print("Lectura del valor del ADC:t");
  Serial.println(balanza.read());
  Serial.println("No ponga ningún objeto sobre la balanza");
  Serial.println("Destarando...");
  balanza.set_scale(); //La escala por defecto es 1
  balanza.tare(20); //El peso actual es considerado Tara.
  Serial.println("Coloque un peso conocido:");
}

void loop() {
  Serial.print("Valor de lectura: t");
  Serial.println(balanza.get_value(10),0);
  delay(100);
}
```

%Para la medición

```
#include "HX711.h"

const int DOUT=A1;
const int CLK=A0;

HX711 balanza;

void setup() {
  Serial.begin(9600);
  balanza.begin(DOUT, CLK);
  Serial.print("Lectura del valor del ADC: ");
  Serial.println(balanza.read());
  Serial.println("No ponga ningun objeto sobre la balanza");
  Serial.println("Destarando...");
  Serial.println("...");
  balanza.set_scale(439430.25); // Establecemos la escala
  balanza.tare(20); //El peso actual es considerado Tara.

  Serial.println("Listo para pesar");
}

void loop() {
  Serial.print("
```

```
Serial.println(" kg");  
delay(500);  
}
```